

Człowiek - najlepsza inwestycja



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Przewodnik Metodyczny

dla nauczycieli informatyki szkół zawodowych
Część pierwsza SQL i PHP

Opracowanie teoretyczne i metodyczne
Elżbieta Perzycka, Janusz Olczak

Opracowanie scenariuszy lekcji
Michał Galas – moduł SQL
Bartosz Komin – moduł PHP



Spis treści

SPIS TREŚCI	1
WSTĘP	2
METODY KSZTAŁCENIA	3
PRZYGOTOWANIE NAUCZYCIELA INFORATYKI	5
ANALIZA PODSTAWY PROGRAMOWEJ KSZTAŁCENIA ZAWODOWEGO	8
MODUŁ SZKOLENIOWY SQL	20
LEKCJE	27
LEKCJA 1 - BAZY DANYCH, PODSTAWOWE POJĘCIA	27
LEKCJA 3 - ZŁOŻONE ZAPYTANIA, GRUPOWANIE, AGREGACJA DANYCH.....	57
LEKCJA 4 – DML	73
LEKCJA 5 - DDL	83
LEKCJA 6 - DCL, UPRAWNIENIA, UŻYTKOWNICY.....	99
LEKCJA 7 - WIDOKI I FUNKCJE.....	115
LEKCJA 8 - PROCEDURY TRIGGERY	130
LEKCJA 9 - INDEKSY, CONSTRAINTY	144
LEKCJA 10 - TRANSAKcje, ZATWIERDZANIE ZMIAN, OBSŁUGA BŁĘDÓW, WPROWADZENIE DO PROJEKTOWANIA BAZY DANYCH	164
LWKcja 11 - PROJEKT INTERDYSCYPLINARNY - KSIĄŻKA TELEADRESOWA	178
MODUŁ SZKOLENIOWY PHP	183
LEKCJE	192
LEKCJA 1 - BEZPIECZEŃSTWO APLIKACJI	192
LEKCJA 2 - ARCHITEKTURA USŁUG SIECIOWYCH	214
LEKCJA 3 - PROTOKÓŁ HTTP.....	232
LEKCJA 4 - MODYFIKACJA NAGŁÓWKÓW HTTP	248
LEKCJA 5 – COOKIE - INFORMACJE O UŻYTKOWNIKU.....	269
LEKCJA 6 - SESJE - WSPÓŁDZIELENIE INFORMACJI	282
LEKCJA 7 - POCZTA - SYSTEMY SMTP	293
LEKCJA 8 - MIME - ROZSZERZENIA MULTIMEDIALNE	313
LEKCJA 9 - UML 2.1 (DIAGRAMY CZYNNOŚCI, PROJEKTOWANIE ZŁOŻONYCH APLIKACJI)	328
LEKCJA 10 - DOBRE PRAKTYKI PROGRAMOWANIA W PHP	346
LEKCJA 11 - PROJEKT INTERDYSCYPLINARNY - KSIĄŻKA TELEADRESOWA	359
BIBLIOGRAFIA	368

Wstęp

Szkoła, a w niej edukacja przechodzi transformację nie tylko spowodowaną legislacyjnymi zmianami, ale także alternatywnym do obecnego zrozumieniem procesu kształcenia. Z jednej strony jest to dążenie do rewitalizacji i kultywowania tradycyjnych wartości, które mieszczą się w szeroko rozumianym nurcie humanizmu, z drugiej strony podporządkowanie się w coraz większym stopniu narastającej fali technologicznej. Zmiany wynikają także z oczekiwań jakie stawia się absolwentom szkoły zawodowej, w szczególności: umiejętnościom rozwiązywania problemów i podejmowania decyzji; krytycznym i twórczym myśleniem; umiejętnościami współpracy i negocjacji; intelektualną ciekawością; wyszukiwaniem, selekcjonowaniem, porządkowaniem i ocenianiem informacji; wykorzystywaniem wiedzy w nowych sytuacjach; integrowaniem technologii z kształceniem i własnym rozwojem (Sysło, 2010).

Tradycyjny model edukacji oparty na behawiorystycznym podejściu do uczenia się ustępuje miejsca refleksji krytycznej, na której wyrósł konstruktywizm. Jego idea nie jest nowa. Odnajdziemy ją w pracach J.Piageta, L.Wygotskiego, J.Brunera czy J.Deweya. Potrzeba poznania tkwi w każdym człowieku i to ona motywuje nas do działania. Instrumentalne podejście do edukacji sprawdza się w wyjaśnianiu prostych opartych na przekazie i pamięci zadań, jednak nie wystarcza, kiedy potrzebujemy uzasadnienia i wyjaśnienia zróżnicowanych zadań w złożonych sytuacjach życia codziennego. Podejście krytyczno-refleksyjne traktuje ucznia i nauczyciela nie tylko jako układ tworzący wiedzę, ale jako aktywne, samodzielnie wybierające i przetwarzające informacje podmioty edukacji. W takim układzie nauczyciel i uczeń podejmują samodzielnie decyzje, realizują indywidualne strategie nauczania i uczenia się. Współistnienie w edukacji informatycznej alternatywnego podejścia do wprowadzania pojęć i rozwijania umiejętności informatycznych jest na dzień dzisiejszy komplementarnym podejściem do kształcenia.

Propozycją wychodzącą naprzeciw oczekiwaniom rynku pracy i efektywnej edukacji jest niniejszy Przewodnik Metodyczny. Zamieszczone w nim scenariusze lekcji odpowiadają założeniom Podstawy Programowej Kształcenia w Zawodzie. Treści przewodnika uwzględniają zintegrowane i skorelowane treści kształcenia ogólnego i zawodowego – informatycznego w tym doskonalenie kompetencji kluczowych nabytych w procesie kształcenia ogólnego, z uwzględnieniem niższych etapów edukacyjnych. W szczególności:

- celom kształcenia zawodowego, których realizacja doprowadzi do przygotowania uczniów do życia w warunkach współczesnego świata, wykonywania pracy zawodowej i aktywnego funkcjonowania na zmieniającym się technologicznym rynku pracy.

- zadaniom szkoły, warunkowanym zmianami zachodzącymi w otoczeniu gospodarczo-społecznym, na które wpływają w szczególności: idea gospodarki opartej na wiedzy, globalizacja procesów gospodarczych i społecznych, rosnący udział handlu międzynarodowym, mobilność

geograficzna i zawodowa, nowe techniki i technologie, a także wzrost oczekiwań pracodawców w zakresie poziomu wiedzy i umiejętności pracowników – techników - informatyków.

Autorzy przewodnika pochylili się tutaj nad koncepcją sfery aktualnego i najbliższego rozwoju zaproponowaną przez L. Wygotskiego. Odpowiedni poziom wiedzy ogólnej powiązanej z wiedzą zawodową - informatyczną przyczyni się do podniesienia poziomu umiejętności zawodowych absolwentów szkół kształcących w zawodzie technik - informatyk, a tym samym zapewni im możliwość sprostania wyzwaniom zmieniającego się rynku pracy.

W przewodniku uwzględniono możliwości dostosowania treści z zakresu tworzenia aplikacji internetowych na przykładach czterech języków programowania (Java, JavaScript, SQL, PHP) do poziomu rozwojowego uczących się. Zagadnienia tematyczne zestawiono w bloki problemowe, jako propozycje zajęć lekcyjnych. Wybór lekcji na kolejne zajęcia jest warunkowany wyborem nauczyciela i może być podyktowany rozpoznaniem poziomu zrozumienia treści przez ucznia i rozpoczynać się od dowolnej lekcji. Kolejność lekcji nie jest obowiązkowa. To nauczyciel decyduje jaki problem będzie analizował z uczniami na kolejnym spotkaniu.

To za sprawą znajomości różnych języków programowania uczniowie będą tworzyć własne zasoby – edukacyjne, kulturalne, informacyjne, artystyczne, graficzne, rozrywkowe. Będą konstruować swój świat – siebie.

Metody kształcenia

Czy metody stosowane dziś w dydaktyce informatyki są skuteczne? Jak wprowadzać ucznia w świat języków programowania, aby był on dla niego przyjazny i rozpoznawalny?

W zakresie kształcenia informatycznego możemy wyróżnić dwie grupy zagadnień:

I grupa – zagadnienia ogólnokształcące, które mogą być realizowane za pomocą metod przekazu (pogadanka, pokaz, referat ucznia, dyskusja, itp.)

II grupa - zagadnienia specjalistyczne wymagające bezpośredniej pracy z komputerem, realizowane przy zastosowaniu metod aktywnych (np. dyskusja, praca w grupach) koncentrujących się na uczniu (np. samodzielne wykonywanie zadań problemowych, ćwiczeń, tworzenie, projektowane).

Niezależnie jednak od tego jaki zakres zagadnień realizuje nauczyciel z uczniami na zajęciach informatycznych nie może zapominać o zasadach metodycznego przygotowania.

Po pierwsze bardzo ważne jest, aby proces kształcenia przebiegał na poziomie dającym uczniom poczucie uczestnictwa w wartościowych dla nich zajęciach. Będzie to możliwe kiedy poziom kształcenia zostanie dostosowany do poziomu rozwoju poznawczego ucznia (aktualnych umiejętności i wcześniejszych osiągnięć), tempa przyswajania nowych informacji, postaw wobec uczenia się i oczekiwań wobec lekcji.

Po drugie język informatyki składa się z wielu trudnych terminów. Możemy spotkać się także z sytuacją ukrytego żargonu. Dla przykładu, menu w życiu codziennym oznacza zupełnie co innego niż w informatyce. Uczniowie mogą nie znać języka używanego przez nauczyciela i odwrotnie, uczniowie mogą stosować terminy odmienne niż nauczyciel, ale znaczeniowo takie same. Nauczyciele mają

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
skłonność do budowania bardziej skomplikowanych zdań niż ich uczniowie (Pety, s.47-48). Na zajęciach informatycznych wskazane jest komunikowanie dostosowane do obu stron procesu kształcenia.

Po trzecie z różnych powodów uczniowie mogą nie być zadowoleni z lekcji. To wynika z różnych dotychczasowych doświadczeń szkolnych. Warto zastanowić się nad zmotywowaniem uczniów do uczenia się. Jak to zrobić? Należy rozpocząć od odpowiedzi na pytanie, czy byli zmotywowani, gdy przyszli na pierwsze zajęcia? Czy motywacja słabła podczas kolejnych zajęć? Ważne są rozmowy z całą klasą jak również z pojedynczymi uczniami o tym, co jest dla nich ważne i dlaczego uczą się języków programowania. Znając problemy swoich uczniów nauczyciel może odnaleźć pomysł na poprawienie ich motywacji zmienić postawy do uczenia się (Perzycka, 2008).

Informatyka dostarcza jakościowo innych warunków niż pozostałe przedmioty kształcenia. Po rozpoznaniu potrzeb i oczekiwań uczniów oraz treści podstawy programowej kształcenia ogólnego nauczyciel realizuje lekcje z uwzględnieniem reguł wprowadzania do języków programowania. Reguły odnoszą się zarówno do nauczyciela jak i do ucznia. Stymulacja najbardziej naturalnych warunków pracy – jest problem do rozwiązania. Trzeba to zrobić szybko i dobrze. Każdy pracuje samodzielnie i nie przeszkadza innym. O efekcie decyduje czas spędzony przy komputerze. Lekcje informatyki to nie czytelnia, można przygotować się w domu. W domu należy przeczytać zadanie i zastanowić się jak je wykonać i gdzie szukać potrzebnych informacji. Przy korzystaniu z instrukcji najważniejsze jest rozeznanie, gdzie znajduje się opis danej metody rozwiązania problemu, czy funkcji programu. Instrukcje czyta się jak encyklopedię. Może to zrobić uczeń lub przekazać nauczyciel. Przy pierwszym pytaniu ucznia nauczyciel pokaże jak znaleźć odpowiedź czy objaśnienie w instrukcji, przy drugim – odeśle do instrukcji, przy trzecim – może wprowadzić w błąd! Kto szybciej rozwiąże postawione problemy będzie miał szansę pogłębić zrealizowany materiał i dostanie nowe zadanie.

Po rozwiązaniu problemów następuje czas na zadania sprawdzające. Każdy dostaje nowe problemy. Nauczyciel wystawia ocenę za wykonane zadanie wcześniej informując ucznia o kryteriach oceniania. Na sprawdzianach z poszczególnych języków programowania można korzystać z dowolnych źródeł. Sprawdziany mają wykazać, czy uczniowie nauczyli się samodzielnie rozwiązywać problemy za pomocą poznanego materiału na kolejnych lekcjach. Oceniany będzie poziom i czas rozwiązania.

W początkowym okresie nauki języków programowania uczeń stoi przed bardzo trudnym zadaniem. Musi bowiem równocześnie: zrozumieć co do niego mówi nauczyciel, nauczyć się interpretować sytuację na ekranie i nauczyć się wydawać polecenia systemowi i programom (Bougon, Weick, Binkhorst, s. 606-639).

Ważnym czynnikiem mającym znaczenie w procesie kształcenia jest problem wielozadaniowości, występujący u młodzieży w toku ich uczenia się. Dlatego zarówno ilość treści jak i sposób ich przyswajania mają decydujący wpływ na rozwój poznawczy.

Podczas pracy nad przygotowaniem przewodników dla nauczycieli skoncentrowano się na takim doborze zadań, aby uwzględniały zarówno percepcję ucznia, jak i stopień trudności treści kształcenia. W ich opracowaniu przyjęto zasadę stopniowania trudności według poziomów przetwarzania informacji, które można rozumieć jako zakres i intensywność przetwarzania informacji, zgodnie z teorią F.I.M.Craika, R.S.Lockharta.

W kontekście poziomów przetwarzania informacji proponujemy stopniowanie trudności uczenia się, poprzez dobór odpowiednich zadań, ćwiczeń i problemów do rozwiązania.

Poziom pierwszy to percepcja sensoryczna. Zastosowane metody podające: wykład i ćwiczenie krok po kroku pozwalają na przekazanie treści, ale nie dają gwarancji trwałego zapamiętania. Metody podające wymagają częstego powtarzania w celu zapamiętania treści – wyuczenia się materiału. Rezultaty są podatne na wszelkiego rodzaju zakłócenia.

Poziom drugi to semantyczny odbiór przekazywanych treści. Zastosowane metody zadaniowe polegające na uzupełnieniach, dokończeniach pozwalają na trwalsze zapamiętanie treści, ale wciąż wymagają powtórzeń. Uczeń rozpoznaje znaczenie poprzez interpretacje odbieranych danych i informacji w formie treści zadań, ćwiczeń.

Poziom trzeci to aktywizowanie wiedzy już posiadanej. Zastosowanie metod problemowych pozwoli wykonać zadanie na zasadzie skojarzeń podczas reorganizacji wiedzy, którą zdobył uczeń na poprzednich poziomach przetwarzania informacji. Im wyższy poziom przetwarzania informacji tym trwalsze zapamiętywanie i głębsze zrozumienie treści, a tym samym skuteczniejsze uczenie się.

Nauczyciel może rozpocząć zajęcia od metody kursowej, polegającej na kolejnym omawianiu poszczególnych funkcji języka programowania i ćwiczeniach z ich wykorzystaniem. Takie rozwiązanie jest przydatne dla osób, które będą zawodowo wykorzystywały konkretny program. W warunkach szkoły sprawia, to wrażenie, że uczy się rzeczy zupełnie później nieprzydatnych, dlatego w dalszych zajęciach wskazane jest wiązanie zadań z codziennymi sytuacjami żywymi/zawodowymi. W większości przypadków uczeń działa sam na sam z komputerem. Po stronie nauczyciela zatem stoi wyzwanie w kierunku precyzji formułowania zadań i odpowiedzialność za ich realizację. Chodzi przede wszystkim o to by uczeń po zapoznaniu się z problemem sam potrafił go rozwiązać. Programowanie to proces twórczy i może prowadzić do innowacyjnych rozwiązań i do odkryć. W takim układzie uczeń staje się odkrywcą, a języki programowania komunikują jego odkrycia.

Przygotowanie nauczyciela inforatyki

Poszukując wytycznych dotyczących przygotowania nauczyciela informatyki do prowadzenia zajęć w zawodowej szkole ponadgimnazjalnej, w dostępnej literaturze przedmiotu odnajdziemy standardy przygotowania nauczyciela informatyki, które zostały opisane w dokumencie z roku 2003, zatytułowanym Standardy przygotowania nauczycieli w zakresie technologii informacyjnej i informatyki przygotowanym przez Radę ds. Edukacji Informatycznej i Medialnej działającej przy Ministerstwie Edukacji Narodowej, pod kierunkiem M.M.Systry (<http://syslo.nq.pl/Edukacja/Dokumenty/Standardy-przygotowania-nauczycieli>).

Nauczyciel informatyki zdefiniowany został jako nauczyciel przygotowany do prowadzenia zajęć z przedmiotu informatyka w zakresie rozszerzonym w szkole ponadgimnazjalnej, które mogą

kończyć się egzaminem maturalnym. Jego kompetencje są rozszerzeniem kompetencji nauczyciela technologii informacyjnej o zagadnienia z informatyki, jako dziedziny naukowej.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Szczegółowe, postulowane kompetencje nauczyciela informatyki zostały opisane w sześciu blokach zagadnieniowych. Są one następujące:

1. Profesjonalne przygotowanie w zakresie informatyki

Nauczyciel wzbogaca kompetencje wymienione w punkcie 1, aby posiadać wiedzę, umiejętności i doświadczenie, odpowiednie do zakresu i poziomu wiedzy informatycznej, niezbędnej do nauczania informatyki, jako dyscypliny naukowej na poziomie szkoły ponadgimnazjalnej. W szczególności:

1.1. Zna podstawy informatyki w zakresie: historii, struktury dziedziny informatyka i jej matematycznych podstaw (elementy teorii mnogości i logiki) oraz powiązań z innymi dziedzinami nauki.

1.2. Zna zaawansowane metody wykorzystywania oraz programowania narzędzi informatycznych takich, jak: edytor tekstu, arkusz kalkulacyjny, system zarządzania bazą danych, wybrane programy edukacyjne.

1.3. Ma praktyczną znajomość przynajmniej dwóch języków programowania wyższego poziomu, w tym języka algorytmicznego i języka do tworzenia prezentacji w sieci, w zakresie projektowania programów oraz testowania i weryfikowania ich poprawności.

1.4. Zna podstawowe metody algorytmicznego rozwiązywania problemów: techniki algorytmiczne, podstawowe algorytmy, algorytmy rozwiązywania klasycznych problemów, własności algorytmów, oraz podstawowe metody analizy algorytmów i ich własności: poprawności, skończoności, złożoności obliczeniowej i efektywności praktycznej.

1.5. Zna podstawowe struktury danych, stosuje je w powiązaniu z wybranymi algorytmami oraz w realizacji algorytmów w wybranym języku programowania.

1.6. Zna budowę typowej lokalnej sieci komputerowej oraz sieci Internet, w tym zasady funkcjonowania architektury klient-serwer.

1.7. Ma osobiste doświadczenie, nabyte w pracowni komputerowej, w zakresie:

- pracy w środowisku różnych systemów operacyjnych i systemów sieciowych;
- administrowania siecią komputerową na potrzeby swoich zajęć;
- praktycznej znajomości zagadnień wymienionych w punktach 5.1.2 - 5.1.6;
- realizacji zespołowych projektów programistycznych.

2. Warsztat pracy nauczyciela informatyki: sprzęt i oprogramowanie

Nauczyciel wzbogaca kompetencje wymienione w punkcie 2 o znajomość kryteriów doboru, sposobów instalowania i utrzymywania w sprawności sprzętu i oprogramowania, niezbędnego do prowadzenia zajęć z informatyki w szkole ponadgimnazjalnej, znajdującego się na wyposażeniu szkolnej pracowni komputerowej.

2.1. Dostosowuje konfigurację sprzętu i oprogramowania do potrzeb swoich zajęć z informatyki. Planuje dalszy ich rozwój zaplecza technicznego swoich zajęć. Radzi sobie z prostymi awariami w pracowni komputerowej.

2.2. Zapewnia przestrzeganie nakazów prawnych i norm etycznych w korzystaniu z TI w pracowni szkolnej. Zapewnia i chroni bezpieczeństwo danych oraz prywatność innych osób.

2.3. Stosuje pełne możliwości: edytorów (tekstu, grafiki i muzyki), systemów składu i arkusza kalkulacyjnego.

2.4. Projektuje i tworzy bazy danych (w tym relacyjne) i korzysta z systemów zarządzania bazą danych.

2.5. Wykorzystuje pełne możliwości narzędzi sieciowych do: poszukiwania, gromadzenia i przetwarzania informacji, komunikacji elektronicznej, tworzenia stron i serwisów internetowych, prezentacji multimedialnych oraz udostępniania ich w sieci.

2.6. Stosuje pełne możliwości środowiska, umożliwiające integrowanie: edytorów tekstu, grafiki i muzyki, arkusza kalkulacyjnego, systemu zarządzania bazą danych, programów komunikacyjnych, usług internetowych i innych mediów.

2.7. Korzysta z oprogramowania edukacyjnego i systemów autorskich do tworzenia mediów dydaktycznych. Tworzy własne pomoce z wykorzystaniem narzędzi informatycznych, np. języka programowania.

2.8. Określa działania i środki wspomagające swój profesjonalny rozwój w zakresie informatyki i TI, uwzględnia w tym zachodzące zmiany technologiczne.

3. Metodyka nauczania informatyki i jej zastosowań

Nauczyciel wzbogaca kompetencje wymienione w punkcie 3 o umiejętności nauczania informatyki jako dziedziny akademickiej, przynajmniej w zakresie wymienionym w punktach 3.1 i 3.2. W szczególności:

3.1. Określa zakres nauczania informatyki w szkole ponadgimnazjalnej na podstawie obowiązującej podstawy programowej tego przedmiotu i programów jego nauczania oraz programów studiów informatycznych.

3.2. Dostosowuje program nauczania i metody nauczania do zmian zachodzących w informatyce, w technologii informatycznej i w technologii informacyjnej.

3.3. Opracowuje i stosuje praktycznie metody nauczania pojęć i kształtowania umiejętności informatycznych, w tym również w pracowni komputerowej, związanych z:

- posługiwaniem się systemami komputerowymi i sieciowymi oraz technologią informacyjną,
- korzystaniem z oprogramowania użytkowego i edukacyjnego oraz języków programowania,
- docieraniem do lokalnych i rozproszonych źródeł informacji, korzystaniem z nich oraz przetwarzaniem i prezentowaniem informacji w różnej postaci.

3.4. Opracowuje i stosuje praktycznie metody: modelowania problemów i ich rozwiązywania, przedstawiania rozwiązań w postaci algorytmicznej oraz otrzymywania rozwiązań za pomocą wybranych narzędzi informatycznych, w tym również za pomocą programów własnych.

3.5. Zna podstawowe zastosowania informatyki w innych dziedzinach kształcenia i stosuje praktycznie nauczanie informatyki zintegrowane z innymi przedmiotami; współpracuje w tym zakresie z nauczycielami innych przedmiotów.

3.6. Projektuje i stosuje w nauczaniu zespołowe metody pracy, w tym zespołowy projekt programistyczny.

3.7. Doradza uczniom w zakresie wyboru drogi dalszego kształcenia się w szkole wyższej na specjalności informatycznej.

3.8. Projektuje i rozwija metody oceniania osiągnięć uczniów w zdobywaniu umiejętności i kompetencji informatycznych.

4. Aspekty humanistyczne, etyczno-prawne i społeczne w dostępie do TI i w korzystaniu z tej technologii

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Nauczyciel jest świadomy, że TI może powodować (w tym również w szkole) powstawanie wielu kwestii prawnych, etycznych i społecznych, a także zagrożeń w tych sferach. Dbą o przestrzeganie norm prawnych i etycznych oraz zasad równouprawnienia w dostępie do komputerów i technologii informacyjnej oraz w posługiwaniu się nią przez uczniów. Przestrzega i wpaja uczniom normy współżycia w kształtującym się społeczeństwie informacyjnym. Wszystkie te kwestie potrafi przedstawić i uzasadnić uczniom. W szczególności:

4.1. Przestrzega norm prawnych i etycznych w korzystaniu ze źródeł informacji oraz w posługiwaniu się informacją w swojej pracy i w pracy z uczniami. Wyjaśnia pochodzenie i zasadność stosowania tych norm. Odróżnia i wyjaśnia różnice między korzystaniem z cudzej własności intelektualnej z powołaniem się na autora a plagiatem. Szanuje prywatność innych użytkowników TI - w tym również uczniów - i chroni ich dane oraz zasoby.

4.2. Uwzględnia w nauczaniu humanistyczne, etyczno-prawne i społeczne aspekty stosowania informatyki przez uczniów, w szkole i poza nią, w tym również w celach osobistych

4.3. Jest świadomy różnic kulturowych tkwiących w informacji, sposobie jej przedstawiania oraz udostępniania. Dbą o zagwarantowanie uczniom równych praw dostępu do komputerów, TI oraz informacji, bez względu na pochodzenie społeczne i kulturowe, płeć, zamożność i wcześniejsze przygotowanie.

4.4. Zna podstawowe zasady etyki w korzystaniu z mediów. Stosuje je w świadomym i krytycznym odbiorze komunikatów medialnych.

4.5. Zna zagrożenia (w tym etyczne i prawne) wynikające z niewłaściwego posługiwania się komputerami oraz korzystania z nieodpowiedniego dla uczniów oprogramowania i źródeł informacji. Potrafi skutecznie przeciwdziałać tym zagrożeniom, chroniąc przed nimi uczniów oraz kształtując ich świat pozytywnych wartości.

4.6. Zna zagrożenia psychiczne i fizyczne dla zdrowia, wynikające z nadmiernego lub niewłaściwego korzystania ze środków TI. Wystrzega się tych zagrożeń, uprzedza o nich uczniów i chroni ich przed nimi.

4.7. Jest świadomy wpływu na demokrację swobodnego dostępu do informacji oraz nieskrępowanej komunikacji. Dostrzega zagrożenia przez to powodowane, związane z globalizacją procesów społecznych.

4.8. Jest świadomy wpływu TI na zachowania społeczne. Potrafi przedstawić trendy w rozwoju techniki i technologii informatycznej i informacyjnej, oraz konsekwencje tego rozwoju dla życia, w tym zawodowego, jednostki i całych społeczności w tworzącym się społeczeństwie informacyjnym. Zna i przedstawia oficjalne wskazania gremiów krajowych i międzynarodowych, odnoszące się do kierunków zmian powodowanych, rozwojem informatyki i TI w skali państwa, Europy i świata.

Analiza podstawy programowej kształcenia zawodowego

Cele kształcenia w zawodzie

Absolwent szkoły kształcącej w zawodzie technik informatyk powinien być przygotowany do wykonywania następujących zadań zawodowych:

- 1) montowania oraz eksploatacji komputera i urządzeń peryferyjnych;

- 2) projektowania i wykonywania lokalnych sieci komputerowych, administrowania tymi sieciami;
- 3) projektowania baz danych i administrowania bazami danych;
- 4) tworzenia stron www i aplikacji internetowych, administrowania tymi stronami i aplikacjami.

Efekty kształcenia

Do wykonywania wyżej wymienionych zadań zawodowych niezbędne jest osiągnięcie zakładanych efektów kształcenia, na które składają się:

- efekty kształcenia wspólne dla wszystkich zawodów;

(BHP). Bezpieczeństwo i higiena pracy

Uczeń:

- 1) rozróżnia pojęcia związane z bezpieczeństwem i higieną pracy, ochroną przeciwpożarową, ochroną środowiska i ergonomią;
- 2) rozróżnia zadania i uprawnienia instytucji oraz służb działających w zakresie ochrony pracy i ochrony środowiska w Polsce;
- 3) określa prawa i obowiązki pracownika oraz pracodawcy w zakresie bezpieczeństwa i higieny pracy;
- 4) przewiduje zagrożenia dla zdrowia i życia człowieka oraz mienia i środowiska związane z wykonywaniem zadań zawodowych;
- 5) określa zagrożenia związane z występowaniem szkodliwych czynników w środowisku pracy;
- 6) określa skutki oddziaływania czynników szkodliwych na organizm człowieka;
- 7) organizuje stanowisko pracy zgodnie z obowiązującymi wymaganiami ergonomii, przepisami bezpieczeństwa i higieny pracy, ochrony przeciwpożarowej i ochrony środowiska;
- 8) stosuje środki ochrony indywidualnej i zbiorowej podczas wykonywania zadań zawodowych;
- 9) przestrzega zasad bezpieczeństwa i higieny pracy oraz stosuje przepisy prawa dotyczące ochrony przeciwpożarowej i ochrony środowiska;

- 10) udziela pierwszej pomocy poszkodowanym w wypadkach przy pracy oraz w stanach zagrożenia zdrowia i życia.

(PDG). Podejmowanie i prowadzenie działalności gospodarczej

Uczeń:

- 1) stosuje pojęcia z obszaru funkcjonowania gospodarki rynkowej;



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

- 2) stosuje przepisy prawa pracy, przepisy prawa dotyczące ochrony danych osobowych oraz przepisy prawa podatkowego i prawa autorskiego;
- 3) stosuje przepisy prawa dotyczące prowadzenia działalności gospodarczej;
- 4) rozróżnia przedsiębiorstwa i instytucje występujące w branży i powiązania między nimi;
- 5) analizuje działania prowadzone przez przedsiębiorstwa funkcjonujące w branży;
- 6) inicjuje wspólne przedsięwzięcia z różnymi przedsiębiorstwami z branży;
- 7) przygotowuje dokumentację niezbędną do uruchomienia i prowadzenia działalności gospodarczej;
- 8) prowadzi korespondencję związaną z prowadzeniem działalności gospodarczej;
- 9) obsługuje urządzenia biurowe oraz stosuje programy komputerowe wspomagające prowadzenie działalności gospodarczej;
- 10) planuje i podejmuje działania marketingowe prowadzonej działalności gospodarczej;
- 11) optymalizuje koszty i przychody prowadzonej działalności gospodarczej.

(JOZ). Język obcy ukierunkowany zawodowo

Uczeń:

- 1) posługuje się zasobem środków językowych (leksykalnych, gramatycznych, ortograficznych oraz fonetycznych), umożliwiającą realizację zadań zawodowych;
- 2) interpretuje wypowiedzi dotyczące wykonywania typowych czynności zawodowych artykułowane powoli i wyraźnie, w standardowej odmianie języka;
- 3) analizuje i interpretuje krótkie teksty pisemne dotyczące wykonywania typowych czynności zawodowych;
- 4) formułuje krótkie i zrozumiałe wypowiedzi oraz teksty pisemne umożliwiające komunikowanie się w środowisku pracy;
- 5) korzysta z obcojęzycznych źródeł informacji.

(KPS). Kompetencje personalne i społeczne

Uczeń:

- 1) przestrzega zasad kultury i etyki;
- 2) jest kreatywny i konsekwentny w realizacji zadań;
- 3) przewiduje skutki podejmowanych działań;
- 4) jest otwarty na zmiany;
- 5) potrafi radzić sobie ze stresem;
- 6) aktualizuje wiedzę i doskonali umiejętności zawodowe;

- 7) przestrzega tajemnicy zawodowej;
- 8) potrafi ponosić odpowiedzialność za podejmowane działania;
- 9) potrafi negocjować warunki porozumień;
- 10) współpracuje w zespole.

(OMZ). Organizacja pracy małych zespołów (wyłącznie dla zawodów nauczanych na poziomie technika)

Uczeń:

- 1) planuje pracę zespołu w celu wykonania przydzielonych zadań;
 - 2) dobiera osoby do wykonania przydzielonych zadań;
 - 3) kieruje wykonaniem przydzielonych zadań;
 - 4) ocenia jakość wykonania przydzielonych zadań;
 - 5) wprowadza rozwiązania techniczne i organizacyjne wpływające na poprawę warunków i jakość pracy;
 - 6) komunikuje się ze współpracownikami.
- efekty kształcenia wspólne dla zawodów w ramach obszaru elektryczno-elektronicznego, stanowiące podbudowę do kształcenia w zawodzie lub grupie zawodów PKZ(E.b);

PKZ(E.b) Umiejętności stanowiące podbudowę do kształcenia w zawodach: technik informatyk, technik tyfloinformatyk, technik teleinformatyk

Uczeń:

- 1) rozpoznaje symbole graficzne i oznaczenia podzespołów systemu komputerowego;
- 2) dobiera elementy i konfiguracje systemu komputerowego;
- 3) dobiera oprogramowanie użytkowe do realizacji określonych zadań;
- 4) stosuje zabezpieczenia sprzętu komputerowego i systemu operacyjnego;
- 5) rozróżnia parametry sprzętu komputerowego;
- 6) charakteryzuje informatyczne systemy komputerowe;
- 7) określa funkcje systemu operacyjnego;

- 8) posługuje się terminologią dotyczącą lokalnych sieci komputerowych;
- 9) charakteryzuje urządzenia sieciowe;
- 10) charakteryzuje rodzaje oprogramowania użytkowego;
- 11) korzysta z publikacji elektronicznych;
- 12) przestrzega zasad zarządzania projektem w trakcie organizacji i planowania pracy;

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

13) stosuje programy komputerowe wspomagające wykonywanie zadań.

- efekty kształcenia właściwe dla kwalifikacji wyodrębnionych w zawodzie technik informatyk opisane w części II:

E.12. Montaż i eksploatacja komputerów osobistych oraz urządzeń peryferyjnych

1. Przygotowanie stanowiska komputerowego do pracy

Uczeń:

- 1) stosuje systemy liczbowe używane w technice komputerowej;
- 2) wymienia funkcje i przestrzega zasad działania poszczególnych elementów jednostki centralnej komputera;
- 3) dobiera urządzenia techniki komputerowej do określonych warunków technicznych;
- 4) montuje komputer osobisty z podzespołów;
- 5) modernizuje i rekonfiguruje komputery osobiste;
- 6) planuje przebieg prac związanych z przygotowaniem komputera osobistego do pracy;
- 7) instaluje i aktualizuje systemy operacyjne i aplikacje;
- 8) stosuje polecenia systemów operacyjnych do zarządzania systemem;
- 9) instaluje i konfiguruje sterowniki urządzeń;
- 10) konfiguruje ustawienia personalne użytkownika w systemie operacyjnym;
- 11) stosuje oprogramowanie narzędziowe systemu operacyjnego;
- 12) stosuje oprogramowanie zabezpieczające;
- 13) odczytuje dokumentację techniczną informatycznych systemów komputerowych;
- 14) opracowuje wskazania do użytkowania systemu operacyjnego;
- 15) sporządza cenniki i kosztorysy stanowisk komputerowych;
- 16) opracowuje dokumentację techniczną stanowiska komputerowego;
- 17) stosuje przepisy prawa autorskiego w zakresie dotyczącym systemów informatycznych;
- 18) rozpoznaje rodzaje licencji oprogramowania komputerowego;
- 19) stosuje przepisy prawa dotyczące certyfikacji CE i recyklingu.

2. Użytkowanie urządzeń peryferyjnych komputera osobistego

Uczeń:

- 1) wyjaśnia zasadę działania interfejsów komputera osobistego;

- 2) wyjaśnia zasadę działania urządzeń peryferyjnych komputera osobistego;
- 3) przygotowuje urządzenia peryferyjne komputera osobistego do pracy;
- 4) stosuje przepisy prawa dotyczące gospodarki odpadami niebezpiecznymi;
- 5) dobiera i wymienia materiały eksploatacyjne urządzeń peryferyjnych komputera osobistego;
- 6) wykonuje konserwację urządzeń peryferyjnych komputera osobistego;
- 7) instaluje sterowniki urządzeń peryferyjnych komputera osobistego;
- 8) konfiguruje urządzenia peryferyjne komputera osobistego.

3. Naprawa komputera osobistego

Uczeń:

- 1) posługuje się narzędziami do naprawy sprzętu komputerowego;
- 2) określa kody błędów uruchamiania komputera osobistego;
- 3) lokalizuje oraz usuwa uszkodzenia sprzętowe podzespołów komputera osobistego;
- 4) lokalizuje oraz usuwa usterki systemu operacyjnego i aplikacji;
- 5) lokalizuje uszkodzenia urządzeń peryferyjnych komputera osobistego;
- 6) sporządza harmonogram prac związanych z lokalizacją i usuwaniem usterek komputera osobistego;
- 7) dobiera oprogramowanie diagnostyczne i monitorujące pracę komputera osobistego;
- 8) odzyskuje z komputera osobistego dane użytkownika;
- 9) tworzy kopie bezpieczeństwa danych;
- 10) formułuje wskazania dla użytkownika po wykonaniu naprawy komputera osobistego;
- 11) sporządza kosztorys naprawy komputera osobistego.

E.13. Projektowanie lokalnych sieci komputerowych i administrowanie sieciami

1. Projektowanie i wykonywanie lokalnej sieci komputerowej

Uczeń:

- 1) rozpoznaje topologie lokalnych sieci komputerowych;
- 2) rozpoznaje i stosuje normy dotyczące okablowania strukturalnego;
- 3) rozpoznaje protokoły sieci lokalnych i protokoły dostępu do sieci rozległej;
- 4) rozpoznaje urządzenia sieciowe na podstawie opisu, symboli graficznych i wyglądu;
- 5) określa funkcje komputerowego systemu sieciowego;

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

- 6) wykonuje projekt lokalnej sieci komputerowej;
- 7) dobiera elementy komputerowej sieci strukturalnej, urządzenia i oprogramowanie sieciowe;
- 8) sporządza kosztorys projektowanej sieci komputerowej;
- 9) dobiera medium do budowy lokalnej sieci komputerowej;
- 10) dobiera przyrządy i urządzenia do montażu okablowania strukturalnego;
- 11) montuje okablowanie sieciowe;
- 12) wykonuje pomiary okablowania strukturalnego;
- 13) opisuje i analizuje klasy adresów IP;
- 14) projektuje strukturę adresów IP w sieci;
- 15) wykonuje pomiary i testy sieci logicznej;
- 16) opracowuje dokumentację powykonawczą lokalnej sieci komputerowej.

2. Konfigurowanie urządzeń sieciowych

Uczeń:

- 1) modernizuje i rekonfiguruje serwery;
- 2) konfiguruje przełączniki lokalnych sieci komputerowych;
- 3) konfiguruje sieci wirtualne w lokalnych sieciach komputerowych;
- 4) konfiguruje rutery i urządzenia zabezpieczające typu zaporę sieciową (ang. firewall);
- 5) konfiguruje urządzenia dostępu do lokalnej sieci komputerowej bezprzewodowej;
- 6) konfiguruje urządzenia telefonii internetowej;
- 7) dobiera i stosuje narzędzia diagnostyczne;
- 8) tworzy sieci wirtualne za pomocą połączeń internetowych;
- 9) monitoruje pracę urządzeń lokalnych sieci komputerowych.

3. Administrowanie sieciami systemami operacyjnymi

Uczeń:

- 1) instaluje sieciowe systemy operacyjne;
- 2) konfiguruje interfejsy sieciowe;
- 3) udostępnia zasoby lokalnej sieci komputerowej;
- 4) charakteryzuje usługi serwerowe;
- 5) określa funkcje profili użytkowników i zasady grup użytkowników;

- 6) zarządza kontami użytkowników i grup użytkowników systemu operacyjnego lub komputera;
- 7) konfiguruje usługi katalogowe lokalnej sieci komputerowej;
- 8) zarządza centralnie stacjami roboczymi;
- 9) rozpoznaje protokoły aplikacyjne;
- 10) monitoruje działania użytkowników lokalnej sieci komputerowej;
- 11) modernizuje lokalną sieć komputerową;
- 12) przestrzega zasad udostępniania i ochrony zasobów sieciowych;
- 13) wyjaśnia zasady działania protokołów lokalnej sieci komputerowej;
- 14) konfiguruje usługi odpowiedzialne za adresację hostów (adresację IP), system nazw, routing, zabezpieczenie przed wszelkiego rodzaju atakami z sieci (firewall);
- 15) podłącza lokalną sieć komputerową do Internetu;
- 16) konfiguruje usługi serwerów internetowych;
- 17) określa rodzaje awarii lub wadliwego działania lokalnej sieci komputerowej;
- 18) lokalizuje i usuwa przyczyny wadliwego działania systemów sieciowych;
- 19) zabezpiecza komputery przed zawirusowaniem, niekontrolowanym przepływem informacji oraz utratą danych.

E.14. Tworzenie aplikacji internetowych i baz danych oraz administrowanie bazami

1. Tworzenie stron internetowych

Uczeń:

- 1) posługuje się hipertekstowymi językami znaczników;
- 2) tworzy strony internetowe za pomocą hipertekstowych języków znaczników;
- 3) tworzy kaskadowe arkusze stylów (CSS);
- 4) wykorzystuje kaskadowe arkusze stylów (CSS) do opisu formy prezentacji strony internetowej;
- 5) rozpoznaje funkcje edytorów spełniających założenia WYSIWYG;

- 6) tworzy strony internetowe za pomocą edytorów spełniających założenia WYSIWYG;
- 7) projektuje strukturę witryny internetowej;
- 8) wykonuje strony internetowe zgodnie z projektami;
- 9) stosuje reguły walidacji stron internetowych;
- 10) testuje i publikuje witryny internetowe;
- 11) stosuje różne modele barw;



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

- 12) przestrzega zasad cyfrowego zapisu obrazu;
- 13) wykonuje projekt graficzny witryny internetowej;
- 14) tworzy grafikę statyczną i animacje jako elementy stron internetowych;
- 15) zmienia atrybuty obiektów graficznych i modyfikuje obiekty graficzne;
- 16) przetwarza i przygotowuje elementy graficzne, obraz i dźwięk do publikacji w Internecie;
- 17) przestrzega zasad komputerowego przetwarzania obrazu i dźwięku.

2. Tworzenie baz danych i administrowanie bazami danych

Uczeń:

- 1) korzysta z funkcji strukturalnego języka zapytań;
- 2) posługuje się strukturalnym językiem zapytań do obsługi baz danych;
- 3) projektuje i tworzy relacyjne bazy danych;
- 4) importuje dane do bazy danych;
- 5) tworzy formularze, zapytania i raporty do przetwarzania danych;
- 6) instaluje systemy baz danych i systemy zarządzania bazami danych;
- 7) modyfikuje i rozbudowuje struktury baz danych;
- 8) dobiera sposoby ustawiania zabezpieczeń dostępu do danych;
- 9) zarządza bazą danych i jej bezpieczeństwem;
- 10) określa uprawnienia poszczególnych użytkowników i zabezpieczenia dla nich;
- 11) udostępnia zasoby bazy danych w sieci;
- 12) zarządza kopiami zapasowymi baz danych i ich odzyskiwaniem;
- 13) kontroluje spójność baz danych;
- 14) dokonuje naprawy baz danych.

3. Tworzenie aplikacji internetowych

Uczeń:

- 1) korzysta z wbudowanych typów danych;
- 2) tworzy własne typy danych;
- 3) przestrzega zasad programowania;
- 4) stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania;
- 5) tworzy własne funkcje, procedury, obiekty, metody wybranych języków programowania;
- 6) wykorzystuje środowisko programistyczne: edytor, kompilator i debugger;

- 7) kompiluje i uruchamia kody źródłowe;
- 8) wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera;
- 9) stosuje skrypty wykonywane po stronie klienta przy tworzeniu aplikacji internetowych;
- 10) wykorzystuje frameworki do tworzenia własnych aplikacji;
- 11) pobiera dane aplikacji i przechowuje je w bazach danych;
- 12) testuje tworzoną aplikację i modyfikuje jej kod źródłowy;
- 13) dokumentuje tworzoną aplikację;
- 14) zamieszcza opracowane aplikacje w Internecie;
- 15) zabezpiecza dostęp do tworzonych aplikacji.

Warunki realizacji kształcenia w zawodzie

Szkoła podejmująca kształcenie w zawodzie technik informatyk powinna posiadać następujące pomieszczenia dydaktyczne:

- pracownię urządzeń techniki komputerowej, wyposażoną w: stanowiska komputerowe (jedno stanowisko dla jednego ucznia); podzespoły umożliwiające montaż komputera osobistego; dodatkowe elementy komputera osobistego umożliwiające jego rekonfigurację; oprogramowanie do wirtualizacji; różne systemy operacyjne stacji roboczej; oprogramowanie narzędziowe, diagnostyczne i zabezpieczające; drukarkę laserową, atramentową, igłową; skaner, ploter, tablicę interaktywną, palmtop PDA (Personal Digital Assistant), tablet, projektor multimedialny, klawiaturę i mysz bezprzewodową, czytnik kart podpisu elektronicznego; adapter Bluetooth; stół monterski z matą i opaską antystatyczną; zestaw urządzeń monterskich; podłączenie do sieci lokalnej z dostępem do Internetu;
- pracownię lokalnych sieci komputerowych, wyposażoną w: stanowiska komputerowe (jedno stanowisko dla jednego ucznia); szafę dystrybucyjną 19" z wyposażeniem, połączoną korytkową instalacją okablowania strukturalnego z czterema punktami elektryczno-logicznymi; serwer stelażowy z kontrolerem pamięci masowej; zasilacz awaryjny z zasilaniem; napęd taśmowy do archiwizacji; komputer typu notebook z obsługą lokalnej sieci bezprzewodowej; przełącznik zarządzany z obsługą lokalnych sieci wirtualnych i portami zasilania przez Ethernet; koncentrator xDSL z obsługą protokołu PPP; ruter z modemem xDSL, z portem Ethernet i obsługą protokołu

PPP, oprogramowanie typu firewall z obsługą wirtualnych sieci prywatnych; punkt dostępu do lokalnej sieci bezprzewodowej z różnego typu antenami zewnętrznymi i portem zasilania przez Ethernet; telefon internetowy; tester okablowania; reflektometr; różne sieciowe systemy operacyjne przeznaczone dla serwera; oprogramowanie do wirtualizacji; oprogramowanie komputerowego wspomaganie projektowania (Computer Aided Design) z biblioteką elementów sieci lokalnej; oprogramowanie do monitorowania pracy sieci; stół monterski z matą i opaską antystatyczną; zestaw narzędzi monterskich; podłączenie do sieci lokalnej z dostępem do Internetu;

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

- pracownię sieciowych systemów operacyjnych, wyposażoną w: stanowiska komputerowe (jedno stanowisko dla jednego ucznia); laptop lub notebook dla nauczyciela z oprogramowaniem do wirtualizacji; różne serwerowe systemy operacyjne z usługami katalogowymi i internetowymi; serwerowe oprogramowanie typu firewall; oprogramowanie do analizy protokołów sieciowych; oprogramowanie do monitorowania pracy sieci; podłączenie do sieci lokalnej z dostępem do Internetu;
- pracownię aplikacji internetowych, wyposażoną w: stanowiska komputerowe (jedno stanowisko dla jednego ucznia); laptop lub notebook dla nauczyciela z oprogramowaniem do wirtualizacji; edytor WYSIWYG stron internetowych z możliwością edycji hipertekstowego języka znaczników i kaskadowych arkuszy stylów, z możliwością walidacji strony; oprogramowanie do tworzenia grafiki i animacji, obróbki materiałów audio i wideo; oprogramowanie serwera relacyjnej bazy danych z programami narzędziowymi; oprogramowanie umożliwiające tworzenie aplikacji internetowych po stronie serwera i klienta w wybranych językach programowania; podłączenie do sieci lokalnej z dostępem do Internetu; dostęp do serwera umożliwiającego publikację stron www i aplikacji internetowych; dostęp do portalu wspierającego pracę grupową, komunikację, publikację wiadomości i materiałów.

W szkole prowadzącej kształcenie w zawodzie technik informatyk językiem obcym ukierunkowanym zawodowo jest język angielski.

Kształcenie praktyczne może odbywać się w: pracowniach szkolnych, placówkach kształcenia ustawicznego, placówkach kształcenia praktycznego oraz podmiotach stanowiących potencjalne miejsca zatrudnienia absolwentów szkół kształcących w zawodzie.

Szkoła organizuje praktyki zawodowe w podmiocie zapewniającym rzeczywiste warunki pracy właściwe dla nauczanego zawodu w wymiarze 4 tygodni (160 godzin).

Minimalna liczba godzin kształcenia zawodowego

Efekty kształcenia wspólne dla wszystkich zawodów oraz efekty kształcenia wspólne dla zawodów w ramach obszaru elektryczno-elektronicznego stanowiące podbudowę do kształcenia w zawodzie lub grupie zawodów - 270 godz.

E.12. Montaż i eksploatacja komputerów osobistych oraz urządzeń peryferyjnych -360 godz.

E.13. Projektowanie lokalnych sieci komputerowych i administrowanie sieciami - 300 godz.

E.14. Tworzenie aplikacji internetowych i baz danych oraz administrowanie bazami - 420 godz.

W szkole liczbę godzin kształcenia zawodowego należy dostosować do wymiaru godzin określonego w przepisach w sprawie ramowych planów nauczania w szkołach publicznych, przewidzianego dla kształcenia zawodowego, zachowując, z wyjątkiem szkoły policealnej dla dorosłych, minimalną liczbę godzin wskazanych w tabeli odpowiednio dla efektów kształcenia: wspólnych dla wszystkich zawodów i wspólnych dla zawodów w ramach obszaru kształcenia stanowiących podbudowę do kształcenia w zawodzie lub grupie zawodów oraz właściwych dla kwalifikacji wyodrębnionych w zawodzie.

Możliwości uzyskiwania dodatkowych kwalifikacji w zawodach w ramach obszaru kształcenia określonego w klasyfikacji zawodów szkolnictwa zawodowego

Absolwent szkoły kształcącej w zawodzie technik informatyk po potwierdzeniu kwalifikacji

E.12. Montaż i eksploatacja komputerów osobistych oraz urządzeń peryferyjnych,

E.13. Projektowanie lokalnych sieci komputerowych i administrowanie sieciami

E.14. Tworzenie aplikacji internetowych i baz danych oraz administrowanie bazami może uzyskać dyplom potwierdzający kwalifikacje w zawodzie technik teleinformatyk, po potwierdzeniu dodatkowo kwalifikacji

E.15. Uruchamianie oraz utrzymanie terminali i przyłączy abonenckich

E.16. Montaż i eksploatacja sieci rozległych.

Moduł szkoleniowy

SQL

Wprowadzenie

Język SQL ("Structure Query Language") to strukturalny język zapytań służącym do budowania systemów obsługi relacyjnych baz danych (RDBMS) na dowolnej platformie sprzętowej. Obecnie jest on standardowym językiem służącym do obsługi komercyjnych baz danych.

SQL został opracowany w latach 70. w firmie IBM. Szybko stał się standardem w komunikacji z serwerami baz danych. Wiele współczesnych systemów relacyjnych baz danych używa do komunikacji z użytkownikiem SQL, dlatego potocznie mówi się, że korzystanie z relacyjnych baz danych to korzystanie z SQL-a.

Pionierską firmą, która rozpoczęła dystrybucję komercyjnych baz danych, był Oracle. Dalsze wprowadzanie SQL-a, w produktach innych firm, wiązało się nierozłącznie z wprowadzaniem modyfikacji pierwotnego języka.

Pierwotną nazwą języka miał być SEQUEL, jednakże okazało się, że nazwa ta była już zastrzeżona przez brytyjską wytwórnię lotniczą Hawker Siddeley.

Język SQL umożliwia zadawanie pytań do baz danych oraz pozwala na zmienianie struktury baz danych. W opracowaniu tym omawiamy tylko ten fragment języka który służy do zadawania pytań.

Systemy informatyczne od zawsze budowane były z myślą o przetwarzaniu elementarnych informacji, które odpowiednio zamodelowane formułowały odpowiedzi na pytania naukowców, analityków, przedsiębiorców biznesu. Bazy danych są zatem nieodłącznym elementem każdego systemu informatycznego. Żadna, nawet najbardziej użyteczna i wysokowydajna technologia, nie odniesie sukcesu, jeżeli nie będzie dostarczać informacji i wskazówek dla końcowego użytkownika. Dlatego znajomość języka manipulacji na rekordach baz danych – SQL kreuje się na pozycję obowiązkową w umiejętnościach każdej osoby, związanej z branżą IT.

W poszczególnych częściach kursu szczegółowo zostaną opisane podstawowe elementy składni języka zapytań SQL. Zapoznanie się z ich treścią pozwoli na swobodną pracę z tymże językiem w ramach dowolnego systemu zarządzania bazą danych.

Cele

Poznanie zagadnień z zakresu relacyjnych baz danych w przykładowym profesjonalnym środowisku relacyjnego systemu zarządzania bazą danych, jakim jest system produkcyjny Microsoft SQL Server 2008. Praktyczne zapoznanie się ze środowiskiem. Przystwojenie dedykowanej terminologii w języku polskim i angielskim. Nabycie umiejętności manipulacji zarówno bazami danych, ich strukturami, jak i przechowywanymi w nich danymi za pomocą języka SQL.

Opis sposobu realizacji celów

10 półtoragodzinnych lekcji składających się z przypomnienia wiedzy z poprzedniej lekcji, przedstawienia materiału wraz z przykładami, ćwiczeń praktycznych. Po zakończeniu całego cyklu - przeprowadzenie egzaminu sprawdzającego wiedzę.

Dodatkowo wprowadzono lekcję 11 zawierającą część projektu interdyscyplinarnego łączącego 4 dziedziny wiedzy z zakresu informatyki (SQL, PHP, JavaScript oraz JAVA) w zakresie zaprojektowania bazy danych oraz obiektów służących jej obsłudze.

Treści kształcenia

Treść kursu została podzielona na 10 bloków tematycznych – po jednym do każdej lekcji:

- Bazy danych, podstawowe pojęcia (tabela, dana, wiersz, relacja, klucz podstawowy, klucz obcy, sortowanie, zapytanie, system zarządzania bazą danych), interakcje z bazą danych (Język SQL, aplikacja, formularz raport)
- Języki programowania serwerów SQL, środowisko programowania w języku SQL, wprowadzenie do języka SQL
- Złożone zapytania, grupowanie, agregacja danych
- DDL
- DML
- DCL, uprawnienia, użytkownicy
- Widoki i Funkcje
- Procedury i wyzwalacze (triggers)
- Indeksy i ograniczenia (constraints)
- Transakcje, zatwierdzanie zmian, obsługa błędów, normalizacja bazy danych
- Projekt interdyscyplinarny – baza teleadresowa kolegów i koleżanek ze szkoły

Opis założonych osiągnięć ucznia

Uczeń po odbyciu kursu pozna podstawy teoretyczne związane z bazami danych, pozna techniki projektowania baz danych, nauczy się praktycznego programowania baz danych w środowisku Microsoft SQL Serwer 2008.

Korelacja z treściami podstawy programowej

Cele kształcenia	Treść kształcenia	Podstawa programowa
Uczeń poznaje podstawowe typy danych	Lekcja 5	Uczeń korzysta z wbudowanych typów danych
Uczeń tworzy procedury, funkcje, triggerzy oraz constrainty	Lekcje 8 i 9	Uczeń tworzy własne typy danych
Uczeń poznaje transakcje bazodanowe Uczeń potrafi obsługiwać błędy Uczeń prawidłowo wykonuje zapytania SQL Uczeń potrafi modyfikować dane	Lekcje: 3, 4, 5, 6, 7, 8, 9	Uczeń przestrzega zasad programowania



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Cele kształcenia	Treść kształcenia	Podstawa programowa
Uczeń potrafi modyfikować struktury danych Uczeń tworzy i modyfikuje obiekty bazy danych		
Uczeń tworzy widoki, procedury, funkcje, triggery oraz constrainty	Lekcje 8 i 9	Uczeń stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania
Uczeń tworzy widoki, procedury, funkcje, triggery oraz constrainty	Lekcje 8 i 9	Uczeń tworzy własne funkcje, procedury, obiekty, metody wybranych języków programowania
Uczeń potrafi zainstalować oraz poprawnie skonfigurować środowisko pracy	Lekcje: 2	Uczeń wykorzystuje środowisko programistyczne: edytor, kompilator, debugger
Uczeń projektuje bazy danych dla aplikacji (w tym webowych)	Lekcje: 10	Uczeń wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera
Uczeń poznaje mechanizmy tworzenia i modyfikacji danych	Wszystkie lekcje modułu	Uczeń pobiera dane aplikacji i przechowuje je w bazie danych
Uczeń potrafi testować zapytania Uczeń poznaje instrukcje modyfikujące obiekty bazodanowe	Lekcje: 2, 3, 4, 5, 6, 7, 8, 9	Uczeń testuje tworzoną aplikację i modyfikuje jej kod źródłowy
Uczeń potrafi komentować kod	Lekcje: 3	Uczeń dokumentuje tworzoną aplikację
Uczeń poznaje podstawy projektowania baz danych Uczeń przenosi bazy danych	Lekcje: 1,2, 10	Uczeń zamieszcza opracowane aplikacje w Internecie
Uczeń poznaje zagadnienia dotyczące bezpieczeństwa, transakcji oraz obsługi błędów	Lekcje: 10	Uczeń zabezpiecza dostęp do tworzonych aplikacji

Sposoby osiągnięcia celów

Po odbyciu kursu uczeń powinien wykonać projekt/zadanie, które zmotywuje go do pracy indywidualnej ze środowiskiem programistycznym i bazą danych. Zadanie utrwali zdobytą na kursie wiedzę i zmusi do wykorzystania zdobytej wiedzy teoretycznej w praktyce. Zadanie powinno być sformułowane w sposób otwarty, tak aby każdy z uczniów mógł wybrać coś dla niego interesującego.

Powinna też zostać dostarczona lista możliwych tematów projektu do wyboru - dla osób, które nie będą miały pomysłu na projekt.

Przykładowe tematy projektu:

- Projekt bazy danych placówki medycznej, z możliwością zapisów na badania (główne tabele: pacjent, lekarz, badanie, wizyta). Opcjonalnie: procedura składowana służąca do zapisów na badanie
- Projekt bazy danych działu kadr w firmie (główne tabele: pracownik, dział, wypłaty). Opcjonalnie: procedura składowana służąca wyliczenia pensji na podstawie ilości przepracowanych godzin
- Projekt bazy danych rezerwacji biletów w kinie (główne tabele: sala, film, seans, bilet). Opcjonalnie: procedura składowana służąca rezerwacji biletów

Lista czynności w zależności od roli

Wykonywane czynności są takie same dla każdej z lekcji, jedynie materiał którego dotyczą jest różny. Może się zdarzyć iż w przypadku barku materiału w danej lekcji dla danej czynności, czynność nie powinna być wykonana.

Czynności nauczyciela	Czynności ucznia
Prezentowanie przygotowanego materiału.	Sporządzanie notatek z prezentowanego materiału
Zadawanie pytań kontrolnych odnośnie już pokazanego materiału	Udzielanie odpowiedzi na pytania nauczyciela
Zadawanie pytań odnośnie materiału, który dopiero będzie pokazany, jeśli możliwe jest uzyskanie odpowiedzi w drodze dedukcji	
Objaśnianie ćwiczeń lekcyjnym, pilnowanie czasu ich wykonania przez uczniów, udzielanie wskazówek	Rozwiązywanie zadań lekcyjnych
Objaśnienie zadań na koniec lekcji, udzielanie wskazówek podczas rozwiązywania, zlecenie dokończenia zadań jako pracę domową	Odrabianie zadań domowych
Sprawdzanie ćwiczeń lekcyjnych i zadań domowych: należy zwrócić uwagę na poprawne formatowanie kodu (wcięcia) poprawne użycie konstrukcji języka dokumentowanie kodu komentarzami otrzymywanie poprawnych rezultatów	
Zapisywanie uwag odnośnie materiałów szkoleniowych i sugestie ewentualnych korekt.	

Propozycje kryteriów oceny i metod sprawdzania osiągnięć ucznia

Opis założonych osiągnięć ucznia – przykłady wymagań na poszczególne oceny szkolne

Ocenę **dopuszczającą (2)** otrzymuje uczeń, który potrafi skonstruować proste, niezłożone zapytania SQL: SELECT, INSERT i UPDATE, zna podstawowe pojęcia związane z bazami danych i silnikami baz danych, potrafi własnymi słowami wyjaśnić podstawowe instrukcje języka.

Ocenę **dostateczną (3)** otrzymuje uczeń, który spełnia wymagania oceny dopuszczającej oraz potrafi stworzyć zapytania SQL zawierające kilka warunków. Zna podstawowe struktury danych w języku SQL. Wykonuje ćwiczenia z niewielką pomocą nauczyciela. Potrafi wskazać podstawowe zastosowania baz danych.

Ocenę **dobrą (4)** otrzymuje uczeń, który spełnia wymagania na ocenę dostateczną oraz potrafi stworzyć skrypt złożony z kilku zapytań. Potrafi łączyć tabele za pomocą instrukcji JOIN. Uczeń powinien być w stanie stworzyć własne obiekty baz danych takie jak widoki i funkcje. Na lekcjach wykonuje samodzielnie zadane ćwiczenia. Umiejętnie korzysta z mechanizmów pozwalających na dbanie o jednorodność i spójność danych (np. klucze i ograniczenia).

Ocenę **bardzo dobrą (5)** otrzymuje uczeń, który spełnia wymagania na ocenę dobrą oraz bardzo dobrze zna teorię omawianą na zajęciach, w trakcie wykonywania ćwiczeń wykazuje inicjatywę, potrafi wskazać kilka rozwiązań zadania, tworzy skrypty języka z wykorzystaniem dobrych praktyk programistycznych. Uczeń potrafi zaprojektować bazę danych poprawną pod względem normalizacji, relacji i ograniczeń kolumn.

Ocenę **celującą (6)** otrzymuje uczeń, którego wiedza wykracza poza omawiany zakres. Np. potrafi stworzyć skomplikowaną procedurę składowaną zawierającą kursor. Tworzy zaawansowane obiekty baz danych takie jak procedury, trigger, funkcje skalarne i tablicowe.

Metody sprawdzania osiągnięć ucznia

Ocena zaliczenia przedmiotu powinna być składową dwóch ocen: testu oraz projektu (zaprojektowanej bazy danych).

Kryteria oceny testu:

- 0-7 poprawnych odpowiedzi - 1
- 8-9 poprawnych odpowiedzi - 2
- 10-11 poprawnych odpowiedzi - 3
- 12-13 poprawnych odpowiedzi - 4
- 14-15 poprawnych odpowiedzi - 5

Kryteria oceny projektu:

- Zaprojektowana baza jest logicznie poprawna, występują relacje, typy danych - 3
- W zaprojektowanej bazie widać próbę normalizacji, zaznaczone są klucze podstawowe i obce, występuje logiczna konwencja nazewnictwa kolumn i tabel - 4
- Poza poprawnym projektem bazy danych uczeń napisał poprawną procedurę składowaną - 5

Test końcowy sprawdzający wiedzę

15 zadań na 30 minutowy test sprawdzający wiedzę. Pogrubioną czcionką zaznaczono prawidłowe odpowiedzi.

- Argumentem funkcji MAX() może być kolumna, w której znajdują się dane typu:
 - Liczby naturalne
 - Ciągi znaków
 - Data
 - Typ money
- Które z wymienionych funkcji agregujących są zdefiniowane w języku SQL?
 - AVG()
 - MODE()
 - MEDIANA()
 - SUMA()
- W bazie danych znajdują się tabele:
Film (id, tytuł, rok, id_rezyser, producent)
Rezyser (id, imię, nazwisko, rok_urodzenia, narodowosc)
W wyniku wykonania zapytania:
SELECT tytuł, rok FROM Film WHERE (rok != 1984 OR rok != 2011) AND
id_rezyser = (SELECT id FROM Rezyser WHERE rok_urodzenia >= 1949 OR
rok_urodzenia <= 1951)
wyświetlone zostaną:
 - Wszystkie filmy (tytuł i rok produkcji) z tabeli Film.
 - Wszystkie filmy (tytuł i rok produkcji), które powstały w innych latach niż 1984 i 2011, oraz których reżyserowie urodzili się pomiędzy rokiem 1949 i 1951.
 - Wszystkie filmy (tytuł i rok produkcji), które powstały w innych latach niż 1984 i 2011, oraz których reżyserowie urodzili się pomiędzy rokiem 1949 i 1951, uszeregowane alfabetycznie.
 - Żadne filmy nie zostaną wyświetlone.
- Za pomocą polecenia JOIN możemy łączyć:
 - Różne tabele ze sobą
 - Tabelę z widokiem
 - Tabelę z indeksem
 - Tabelę z samą sobą
- Najczęściej używanym zapytaniem w tabeli tbl jest:
SELECT id, a, b FROM tbl WHERE a = @a AND b = @b;
Jaki indeks powinien zostać założony na tej tabeli
 - CREATE INDEX tbl_idx ON a (tbl);
 - CREATE INDEX tbl_idx ON b (tbl);
 - CREATE INDEX tbl_idx ON tbl (a, b);
 - CREATE INDEX tbl_idx ON tbl (a);

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

6. Którym poleceniem można zamienić "Kowalski" na "Nowak" w kolumnie „Nazwisko” w tabeli Osoby?
 - a. `MODIFY Osoby SET Nazwisko= 'Nowak' WHERE Nazwisko ='Kowalski'`
 - b. `UPDATE Osoby SET Nazwisko ='Kowalski' INTO Nazwisko ='Nowak'`
 - c. `MODIFY Osoby SET Nazwisko ='Kowalski' INTO Nazwisko ='Nowak'`
 - d. `UPDATE Osoby SET Nazwisko ='Nowak' WHERE Nazwisko ='Kowalski'`

7. Za pomocą SQL, dodaj do tabeli Osoby wiersz, gdzie nazwiskiem będzie "Olsen"?
 - a. `INSERT ('Olsen') INTO Osoby (Nazwisko)`
 - b. `INSERT INTO Osoby (Nazwisko) VALUES ('Olsen')`
 - c. `INSERT 'Olsen' INTO Osoby (Nazwisko)`
 - d. `INSERT INTO Osoby ('Olsen') INTO Nazwisko`

8. Jakie zapytanie zwróci wszystkie rekordy z tabeli Osoby, których nazwisko zaczyna się na literę "a"?
 - a. `SELECT * FROM Osoby WHERE Nazwisko='a'`
 - b. `SELECT * FROM Osoby WHERE Nazwisko ='%a%'`
 - c. `SELECT * FROM Osoby WHERE Nazwisko LIKE 'a%'`
 - d. `SELECT * FROM Osoby WHERE Nazwisko LIKE '%a'`

9. Klauzula TOP
 - a. Musi być umieszczona na końcu zapytania.
 - b. Służy do wyboru rekordów o największej wartości we wskazanej kolumnie.
 - c. Jest niedostępna w systemie MSSQL.
 - d. Pozwala ograniczyć liczbę wyświetlanych wyników

10. Złączenia tabel w języku SQL
 - a. Są realizowane za pomocą instrukcji MERGE
 - b. Powodują trwałe zmiany w strukturze bazy danych
 - c. Można wykonywać na dowolnej liczbie tabel (większej niż 2)
 - d. Są realizowane za pomocą operatora „ ; ” (średnik)

11. W której postaci normalnej znajduje się poniższa tabela

Imię	Nazwisko	Adres	Pesel
Jan	Kowalski	Ul. Mała 11, 60-003 Poznań	82111112345
Joanna	Kowalska	Ul. Długa 9a, 00-103 Warszawa	78010112345

- a. Tabela nie jest znormalizowana
- b. W pierwszej postaci normalnej
- c. W drugiej postaci normalnej
- d. W trzeciej postaci normalnej

12. UPDATE jest częścią języka
- DDL
 - DML
 - DQL
 - Żadnego z powyższych
13. Która z poniższych definicji widoku jest prawidłowa?
- CREATE VIEW Widok As SELECT imie, nazwisko, pesel FROM pracownicy
 - CREATE VIEW Widok SELECT * FROM pracownicy
 - CREATE VIEW Widok FROM SELECT * FROM pracownicy
 - CREATE VIEW Widok As imie, nazwisko, pesel FROM pracownicy
14. Do czego służy polecenie:
- ```
SELECT imie, nazwisko, pesel INTO pracownicy FROM klienci
```
- Zapytanie wyświetla imię, nazwisko i pesel z tabeli pracownicy
  - Za pytanie wyświetla imię, nazwisko i pesel z tabeli klienci
  - Tworzy tabelę klienci
  - Zapytanie używane przy tworzeniu widoków – na bazie tabeli klienci tworzy widok pracownicy
15. Które rozszerzenie języka SQL wykorzystywane jest w Microsoft SQL Server
- PL SQL
  - PLPsg SQL
  - T SQL
  - MS SQL

## Lekcje

### Lekcja 1 - Bazy danych, podstawowe pojęcia.

**(tabela, dana, wiersz, relacja, klucz podstawowy, klucz obcy, sortowanie, zapytanie, system zarządzania bazą danych), interakcje z bazą danych (Język SQL, aplikacja, formularz raport)**

#### Cel lekcji

Celem lekcji jest wyjaśnienie, czym jest baza danych. Nauczyciel zaprezentuje przykłady bazy danych w życiu codziennym (książka telefoniczna, system rezerwacji biletów lotniczych, system aukcji elektronicznych) – tak, aby uczniowie zdali sobie sprawę, że bazy danych są obecne w ich życiu. Starając się zainteresować tematem nauczyciel zapyta, w jakich bazach danych znajdują się dane uczniów (eWUŚ, PESEL, NFZ, elektroniczny dzienniczek ucznia).

W lekcji zdefiniowane zostaną podstawowe pojęcia dotyczące baz danych: baza danych, system zarządzania bazą danych, tabela, dana, wiersz, relacja, klucz podstawowy, klucz obcy, sortowanie, zapytanie. W ramach przykładów zaprezentowane zostaną proste schematy bazy danych, np. książki telefonicznej. Na ich przykładzie nauczyciel zademonstruje zdefiniowane pojęcia.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## 1.2. Treść - slajdy z opisem

Slajd 1

SQL jest najpopularniejszym językiem programowania baz danych. W tej lekcji wyjaśnimy sobie czym jest baza danych oraz objaśnimy podstawowe pojęcia dotyczące baz danych.

Slajd 2

Pytania do uczniów: jakie znacie przykłady baz danych? W jakich bazach znajdują się Wasze dane?

Slajd 3

W czasie kiedy komputery nie były jeszcze znane tak powszechnie jak dzisiaj, wszystkie informacje były gromadzone na papierze. Pracownicy firmy musieli ręcznie organizować dane. Zapisywać je, wyszukiwać, aktualizować itd. Dzisiaj, oprócz operacji, które muszą wykonać ręcznie, pozostałe wykonuje za pomocą funkcji, które dostarcza relacyjny system bazy danych.



Slajd 4

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

### Definicja bazy danych

- zbiór informacji wraz z możliwością łatwego dostępu oraz ich zmiany

DAILY Lekcja 1 - Bazy danych, podstawowe pojęcia

Najogólniej rzecz biorąc, „baza danych” to zbiór informacji wraz z możliwością łatwego dostępu oraz ich zmiany (tj. modyfikacją, dodawaniem nowych i usuwaniem starych) z poziomu aplikacji z niej korzystającej.

Często używamy sformułowania „Używam bazy danych”, co dokładniej oznacza - „korzystam ze zbioru informacji, który łatwo odczytywać i zmieniać”.

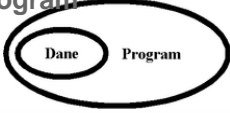
Slajd 5

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

### Elektroniczne bazy danych

#### Pliki w programie

- Dane dostępne tylko dla programu
- Problem z wieloma operacjami jednocześnie
- Brak możliwości skorzystania z danych przez inny program



DAILY Lekcja 1 - Bazy danych, podstawowe pojęcia

Pytanie do uczniów: czy plik tekstowy z nazwiskami i numerem PESEL jest bazą danych?

Plik tekstowy można uznać za bazę danych jeśli odczytanie i modyfikację zapisanych w nim informacji w konkretnym zastosowaniu uznamy za łatwe. Przykładowo niech program wyświetla krótki komunikat, który powinien mieć możliwość dowolnej zmiany. Zastosowanie wtedy zwykłego pliku z tekstem jest jak najbardziej dobrym rozwiązaniem i można go nazwać „bazą danych”.

Rysunek na slajdzie przedstawia sytuację, gdzie program korzysta z pliku tekstowego, który jest jego bazą. Wiąże się to jednak z ograniczeniami (wymienione na slajdzie).

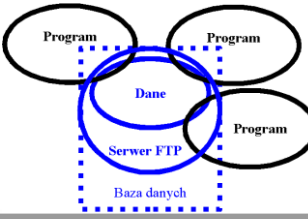
Slajd 6

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

### Elektroniczne bazy danych

#### Pliki dostępne dla wielu programów

- Pliki na serwerze np. FTP
- Wiele aplikacji korzystających z plików
- Zarządzaniem dostępem



DAILY Lekcja 1 - Bazy danych, podstawowe pojęcia

W takim razie czy plik na serwerze (np. FTP), z którego mogą korzystać różne programy jest już „prawdziwą bazą danych”? Mogą z niego korzystać różne programy, lub praktycznie każdy, kto ma dostęp do serwera.

W przedstawionej sytuacji nie ma mechanizmu zapewniającego zarządzania dostępem do danych.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 7

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## System Zarządzania Bazą Danych

- „Coś” pomiędzy danymi a programem

Dane + SZBD = Baza danych

DAILY Lekcja 1 - Bazy danych, podstawowe pojęcia

Brakującym elementem jest System Zarządzania Bazą Danych, który wraz z danymi stanowi BAZĘ DANYCH

Slajd 8

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## System Zarządzania Bazą Danych

- (Database Management System, DBMS) – oprogramowanie bądź system informatyczny służący do zarządzania bazą danych.

DAILY Lekcja 1 - Bazy danych, podstawowe pojęcia

Definicja SZDB

Slajd 9

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## System Zarządzania Bazą Danych

- łatwy dostęp do danych (dane są zorganizowane w struktury)
- jednoczesny dostęp do danych przez wielu użytkowników (architektura klient-serwer, transakcje)
- zmianę istniejących danych (dodawanie, usuwanie, edycja)
- kontrolę dostępu do danych (autoryzacja, ograniczenie dostępu)
- bezpieczeństwo danych (archiwizacja, logowanie)

DAILY Lekcja 1 - Bazy danych, podstawowe pojęcia


Cechy, określające system zarządzania bazą danych

Slajd 10

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Systemy Zarządzania Bazą Danych

- Przykłady:
  - Oracle
  - IBM DB2
  - Microsoft SQL Server
  - Firebird
  - MySQL
  - PostgreSQL
  - Sybase



DAILY GROUP Lekcja 1 - Bazy danych, podstawowe pojęcia

Pytanie do uczniów: czy znacie przykłady SZBD?

Slajd 11

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Relacyjny System Bazy Danych

| id_działu | nazwa            | adres                  |
|-----------|------------------|------------------------|
| 1         | Marketing        | Warszawa, Słoneczna 3  |
| 2         | Księgowość       | Warszawa, Słoneczna 3  |
| 3         | Administracja IT | Warszawa, Słoneczna 3a |
| 4         | Programiści      | Bydgoszcz, Kwiatowa 16 |

| p_id | szeft_projektu | nazwa              | termin_oddania |
|------|----------------|--------------------|----------------|
| it1  | 2              | Upgrade sprzętu    | 10.01.2009     |
| p1   | 5              | Program magazynowy | 1.08.2009      |
| p2   | 8              | Statistica         | 20.02.2009     |

| id | imie   | nazwisko   | id_działu | pensja | projekt |
|----|--------|------------|-----------|--------|---------|
| 1  | Jan    | Kowalski   | 2         | 2600   |         |
| 2  | Adam   | Nowak      | 3         | 3100   | it1     |
| 3  | Halina | Szańska    | 4         | 1800   |         |
| 4  | Anna   | Ryś        | 3         | 3000   |         |
| 5  | Piotr  | Lis        | 4         | 2000   | p1      |
| 6  | Paweł  | Lis        | 4         | 1800   | p2      |
| 7  | Jan    | Nowikowski | 1         | 2200   | it1     |
| 8  | Adam   | Kot        | 4         | 1900   | p2      |

DAILY GROUP Lekcja 1 - Bazy danych, podstawowe pojęcia

Relacyjny system baz danych przechowuje wszystkie dane w tabelach. Każda tabela zawiera dane na konkretny temat, np. dane o klientach, pracownikach, towarach itp. System bazy danych zarządza tymi informacjami, pozwala m.in. na szybsze ich wyszukiwanie i zorganizowanie.



Slajd 12

**Relacyjny System Bazy Danych**

- W bazach relacyjnych wiele tablic danych może współpracować ze sobą
- Wszystkie dane w bazie relacyjnej przedstawiane są w formie dwuwymiarowych tabel
- Wszystkie wartości danych oparte są na typach danych
- Możliwe jest porównywanie wartości z różnych kolumn, tabel
- Wszystkie operacje wykonywane są w oparciu o algebrę relacji
- Przez brak możliwości identyfikacji wiersza przez jego pozycję niezbędna jest jedna lub więcej kolumn niepowtarzalnych

Lekcja 1 - Bazy danych, podstawowe pojęcia

W bazach relacyjnych wiele tablic danych pozostaje w relacjach ze sobą (są między sobą powiązane). Bazy relacyjne posiadają wewnętrzne języki programowania, wykorzystujące zwykle SQL do operowania na danych, za pomocą których tworzone są zaawansowane funkcje obsługi danych. Relacyjne bazy danych (jak również przeznaczony dla nich standard SQL) oparte są na kilku prostych zasadach:

1. Wszystkie wartości danych oparte są na typach danych.
2. Wszystkie dane w bazie relacyjnej przedstawiane są w formie dwuwymiarowych tabel (w matematycznym żargonie noszących nazwę „relacji”). Każda tabela zawiera zero lub więcej wierszy (w tymże żargonie – „krotki”) i jedną lub więcej kolumn („atrybutów”). Na każdy wiersz składają się jednakowo ułożone kolumny wypełnione wartościami, które z kolei w każdym wierszu mogą być inne.
3. Po wprowadzeniu danych do bazy, możliwe jest porównywanie wartości z różnych kolumn, zazwyczaj również z różnych tabel, i scalanie wierszy, gdy pochodzące z nich wartości są zgodne. Umożliwia to wiązanie danych i wykonywanie stosunkowo złożonych operacji w granicach całej bazy danych.
4. Wszystkie operacje wykonywane są w oparciu o algebrę relacji, bez względu na położenie wiersza tabeli. Nie można więc zapytać o wiersze, gdzie (x=3) bez wiersza pierwszego, trzeciego i piątego. Wiersze w relacyjnej bazie danych przechowywane są w porządku zupełnie dowolnym – nie musi on odzwierciedlać ani kolejności ich wprowadzania, ani kolejności ich przechowywania.
5. Z braku możliwości identyfikacji wiersza przez jego pozycję pojawia się potrzeba obecności jednej lub więcej kolumn niepowtarzalnych w granicach całej tabeli, pozwalających odnaleźć konkretny wiersz. Kolumny te określa się

jako „klucz podstawowy” (ang. primary key) tabeli.

Slajd 13

  **SQL**  
Structured Query Language

### Podstawowe pojęcia

- Tabela

Pracownicy

| id | imie   | nazwisko   | id_dzialu | pensja | projekt |
|----|--------|------------|-----------|--------|---------|
| 1  | Jan    | Kowalski   | 2         | 2600   |         |
| 2  | Adam   | Nowak      | 3         | 3100   | it1     |
| 3  | Halina | Szańska    | 4         | 1800   |         |
| 4  | Anna   | Ryś        | 3         | 3000   |         |
| 5  | Piotr  | Lis        | 4         | 2000   | p1      |
| 6  | Paweł  | Lis        | 4         | 1800   | p2      |
| 7  | Jan    | Nowikowski | 1         | 2200   | it1     |
| 8  | Adam   | Kot        | 4         | 1900   | p2      |

 Lekcja 1 - Bazy danych, podstawowe pojęcia

Tabela składa się z wierszy i kolumn. Wiersze w tabeli są przechowywane w porządku zależnym od implementacji bazy danych. Dla każdego wiersza każda z kolumn posiada jedno pole z wartością. Wszystkie wartości w kolumnie są tego samego typu.

Slajd 14

  **SQL**  
Structured Query Language

### Podstawowe pojęcia

- Dana

Pracownicy

| id | imie   | nazwisko   | id_dzialu | pensja | projekt |
|----|--------|------------|-----------|--------|---------|
| 1  | Jan    | Kowalski   | 2         | 2600   |         |
| 2  | Adam   | Nowak      | 3         | 3100   | it1     |
| 3  | Halina | Szańska    | 4         | 1800   |         |
| 4  | Anna   | Ryś        | 3         | 3000   |         |
| 5  | Piotr  | Lis        | 4         | 2000   | p1      |
| 6  | Paweł  | Lis        | 4         | 1800   | p2      |
| 7  | Jan    | Nowikowski | 1         | 2200   | it1     |
| 8  | Adam   | Kot        | 4         | 1900   | p2      |

 Lekcja 1 - Bazy danych, podstawowe pojęcia

Dana to pojedynczy wpis w tabeli – jak komórka w MS Excel. Często stosowanym pojęciem jest atrybut.

Slajd 15

  **SQL**  
Structured Query Language

### Podstawowe pojęcia

- Wiersz

Pracownicy

| id | imie   | nazwisko   | id_dzialu | pensja | projekt |
|----|--------|------------|-----------|--------|---------|
| 1  | Jan    | Kowalski   | 2         | 2600   |         |
| 2  | Adam   | Nowak      | 3         | 3100   | it1     |
| 3  | Halina | Szańska    | 4         | 1800   |         |
| 4  | Anna   | Ryś        | 3         | 3000   |         |
| 5  | Piotr  | Lis        | 4         | 2000   | p1      |
| 6  | Paweł  | Lis        | 4         | 1800   | p2      |
| 7  | Jan    | Nowikowski | 1         | 2200   | it1     |
| 8  | Adam   | Kot        | 4         | 1900   | p2      |

 Lekcja 1 - Bazy danych, podstawowe pojęcia

Pojedynczy wiersz tabeli nazywany jest rekordem (lub encją) i stanowi najczęściej zbiór danych o pojedynczym obiekcie (ew. grupie obiektów).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 16

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Podstawowe pojęcia

- Kolumna

Pracownicy

| id | imie   | nazwisko   | id_dzialu | pensja | projekt |
|----|--------|------------|-----------|--------|---------|
| 1  | Jan    | Kowalski   | 2         | 2600   |         |
| 2  | Adam   | Nowak      | 3         | 3100   | it1     |
| 3  | Halina | Szańska    | 4         | 1800   |         |
| 4  | Anna   | Ryś        | 3         | 3000   |         |
| 5  | Piotr  | Lis        | 4         | 2000   | p1      |
| 6  | Paweł  | Lis        | 4         | 1800   | p2      |
| 7  | Jan    | Nowikowski | 1         | 2200   | it1     |
| 8  | Adam   | Kot        | 4         | 1900   | p2      |

DAILY GROUP Lekcja 1 - Bazy danych, podstawowe pojęcia

W relacyjnym modelu baz danych i podobnych, kolumny stanowią zwykle atrybuty jakiegoś obiektu (np. wielkość, grubość, tytuł, nazwisko) i stąd dane zawarte w kolumnach mają najczęściej jeden określony typ. Dodatkowo w bazach obsługiwanych przez język SQL kolumnom nadawane są nazwy, także poza etapem projektowym i nazwy te są unikatowe w obrębie jednej tabeli.

Slajd 17

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Podstawowe pojęcia

- Relacja

T\_Odpowiedz

- pytanie\_id
- numer
- fala\_id
- tekst
- punkty
- emotikon

Referenced objects

- T\_Fala (id)
- T\_Pytanie (id)

T\_Fala

- id
- nazwa
- typfali\_id
- reklamacje\_aktywne
- media\_aktywne
- kom\_wyniki
- kom\_ranking
- kom\_ankieta
- kom\_reklamacje
- kom\_media

Referenced objects

T\_Pytanie

- id
- fala\_id
- numer\_pytania
- tekst
- obszar\_id
- kolejnosc
- poziom
- czy\_punktowane

Referenced objects


- T\_Fala (id)
- T\_Pytanie (id)

DAILY GROUP Lekcja 1 - Bazy danych, podstawowe pojęcia

Każda relacja (prezentowana w postaci np. tabeli) posiada unikatową nazwę, nagłówek i zawartość. Nagłówek relacji to zbiór atrybutów, gdzie atrybut jest parą nazwa\_atributu: nazwa\_typu,

zawartość natomiast jest zbiorem krotek (reprezentowanych najczęściej w postaci wiersza w tabeli). W związku z tym, że nagłówek jest zbiorem atrybutów nie jest ważna ich kolejność. Atrybuty zazwyczaj utożsamiane są z kolumnami tabeli. Każda krotka (wiersz) wyznacza zależność pomiędzy danymi w poszczególnych komórkach (np. osoba o danym numerze PESEL posiada podane nazwisko i imię oraz adres).

Slajd 18





**SQL**  
Structured Query Language

---

## Podstawowe pojęcia

- Rodzaje relacji

| Zamówienia : Tabela |            |               |
|---------------------|------------|---------------|
| ID zamówienia       | ID klienta | ID pracownika |
| 10000               | FRANS      | 6             |
| 10001               | MEREP      | 8             |
| 10002               | FOLKO      | 3             |


Klucz podstawowy z tabeli "Zamówienia"  
 Klucz podstawowy z tabeli "Produkty"

W jednym zamówieniu może być wiele produktów...

| Opisy zamówień : Tabela |             |               |  |
|-------------------------|-------------|---------------|--|
| ID zamówienia           | ID produktu | Cena jednostk |  |
| 10001                   | 25          | 14,00 zł      |  |
| 10001                   | 40          | 12,80 zł      |  |
| 10001                   | 59          | 38,50 zł      |  |
| 10002                   | 25          | 14,00 zł      |  |

...i każdy produkt może się pojawić w wielu zamówieniach.

| Produkty : Tabela |                         |                  |
|-------------------|-------------------------|------------------|
| ID produktu       | Nazwa produktu          | Cena jednostkowa |
| 25                | NuNuCa Nuß-Nougat Creme | 14,00 zł         |
| 26                | Gumbär Gummibärchen     | 31,23 zł         |



Lekcja 1 - Bazy danych, podstawowe pojęcia

### 1. Relacja jeden-do-jednego

W relacji **jeden-do-jednego** każdy rekord w tabeli A może mieć tylko jeden dopasowany rekord z tabeli B, i tak samo każdy rekord w tabeli B może mieć tylko jeden dopasowany rekord z tabeli A. Ten typ relacji spotyka się rzadko, ponieważ większość informacji powiązanych w ten sposób byłoby zawartych w jednej tabeli. Relacji **jeden-do-jednego** można używać do podziału tabeli z wieloma polami, do odizolowania części tabeli ze względów bezpieczeństwa, albo do przechowania informacji odnoszącej się tylko do podzbioru tabeli głównej.

### 2. Relacja jeden-do-wielu

Relacja jeden-do-wielu jest najbardziej powszechnym typem relacji.

W relacji **jeden-do-wielu** rekord w tabeli A może mieć wiele dopasowanych do niego rekordów z tabeli B, ale rekord w tabeli B ma tylko jeden dopasowany rekord w tabeli A.

### 3. Relacja wiele-do-wielu

W relacji **wiele-do-wielu**, rekord w tabeli A może mieć wiele dopasowanych do niego rekordów z tabeli B i tak samo rekord w tabeli B może mieć wiele dopasowanych do niego rekordów z tabeli A. Jest to możliwe tylko przez zdefiniowanie trzeciej tabeli (nazywanej tabelą łącząca), której klucz



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

podstawowy składa się z dwóch pól z kluczy obcych z tabel A i B. Relacja **wiele-do-wielu** jest w istocie dwiema relacjami jeden-do-wielu z trzecią tabelą. Na przykład, tabele "Zamówienia" i "Produkty" są powiązane relacją **wiele-do-wielu** zdefiniowaną przez utworzenie dwóch relacji jeden-do-wielu z tabelą "Opisy zamówień".

Slajd 19



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Podstawowe pojęcia

- Klucz podstawowy(główny)- PK

**T\_Odpowiedz**

pytanie\_id  
numer  
fala\_id  
tekst  
punkty  
emotikon

Referenced objects

- T\_Fala (id)
- T\_Pytanie (id)

**T\_Fala**

id  
nazwa  
typfali\_id  
reklamacje\_aktywne  
media\_aktywne  
kom\_wyniki  
kom\_ranking  
kom\_ankieta  
kom\_reklamacje  
kom\_media

Referenced objects

**T\_Pytanie**

id  
fala\_id  
numer\_pytania  
tekst  
obszar\_id  
kolejnosc  
poziom  
czy\_punktowane

Referenced objects

- T\_Fala (id)
- T\_Pytanie (id)



Lekcja 1 - Bazy danych, podstawowe pojęcia

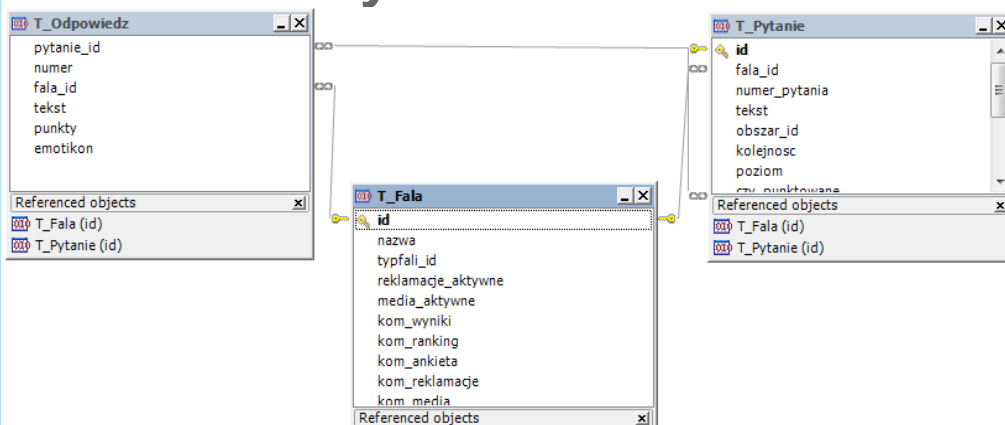
Każda tabela posiada tzw. klucz główny (primary key). Klucz ten jest unikatowym identyfikatorem w relacji i może być kombinacją kilku kolumn, często jednak obejmuje jedną kolumnę (jeden atrybut). Klucz ma za zadanie jednoznacznie identyfikować każdą krotkę (wiersz) – wartości w wyznaczonych kolumnach są jako zestaw niepowtarzalne w danej tabeli.





## Podstawowe pojęcia


- Klucz obcy - FK




Innym rodzajem klucza jest tzw. klucz obcy (foreign key). Jest to zbiór atrybutów jednej tabeli (relacji) wskazujący wartości klucza kandydującego innej tabeli. Służy do wskazywania zależności pomiędzy danymi składowanymi w różnych tabelach. Klucze w modelu relacyjnym służą m.in. do sprawdzania spójności danych w bazie. Głównie dotyczy to kluczy obcych, na które nałożony jest wymóg, że w tabeli wskazywanej musi istnieć wartość klucza wskazującego.



Slajd 21



**B2E**  
BUSINESS TO EDUCATION



**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

---


## Podstawowe pojęcia

- **Sortowanie**

| Lp. | Nazwisko | Imię     | Liczba książek |
|-----|----------|----------|----------------|
| 1   | Pasek    | Karolina | 24             |
| 2   | Nowak    | Zbigniew | 21             |
| 3   | Różga    | Adam     | 19             |
| 4   | Opal     | Ewa      | 14             |
| 5   | Kowalski | Paweł    | 10             |
| 6   | Nowak    | Zuzanna  | 8              |
| 7   | Nowak    | Anna     | 5              |

➔

| Lp. | Nazwisko | Imię     | Liczba książek |
|-----|----------|----------|----------------|
| 5   | Kowalski | Paweł    | 10             |
| 7   | Nowak    | Anna     | 5              |
| 2   | Nowak    | Zbigniew | 21             |
| 6   | Nowak    | Zuzanna  | 8              |
| 4   | Opal     | Ewa      | 14             |
| 1   | Pasek    | Karolina | 24             |
| 3   | Różga    | Adam     | 19             |



Lekcja 1 - Bazy danych, podstawowe pojęcia

Dodatkowym elementem modelu relacyjnego jest zbiór operacji służących do przeszukiwania i manipulacji danymi. Przykładem operacji jest sortowanie danych.



SQL  
Structured Query Language

## Podstawowe pojęcia

- Zapytanie



Czy mogę prosić o sporządzenie listy wszystkich kluczowych klientów?

**DAILY GROUP** Lekcja 1 - Bazy danych, podstawowe pojęcia

Najbardziej typowym zastosowaniem bazy danych jest zapytanie o wiersze spełniające konkretne kryteria, np. lista kluczowych klientów z tabeli klienci.



SQL  
Structured Query Language

## Interakcje z bazą danych

- Język SQL
  - Jedyny sposób interakcji z bazą danych
  - Język deklaratywny
  - Ustandaryzowany
    - Różni producenci stosują ten sam standard

```
select nazwisko, etat, placa
from pracownicy
where idzesp=30
and etat='kierownik'
```

**DAILY GROUP** Lekcja 1 - Bazy danych, podstawowe pojęcia

Jakakolwiek interakcja aplikacji z bazą danych odbywa się za pomocą języka SQL. Jest to jedyny sposób komunikowania się aplikacji z bazą danych.

SQL jest językiem deklaratywnym (posługując się nim specyfikujemy tylko co chcemy otrzymać). Nie specyfikujemy sposobu (algorytmu) w jaki ma być zrealizowane zadanie. Przykładem polecenia SQL może być zapytanie do bazy danych poszukujące informacje o klientach banku ze Szczecina, którzy w ciągu ostatniego miesiąca wypłacili z bankomatu łącznie powyżej 8000 PLN. W tym zapytaniu specyfikujemy tylko jakie dane nas interesują. Sposób ich wyszukania jest automatycznie dobierany przez SZBD.

SQL jest językiem ustandaryzowanym. Jego standardyzacją zajmuje się specjalny międzynarodowy komitet, w skład którego wchodzi przedstawiciele największych producentów SZBD (IBM, Microsoft, Oracle). Dotychczas opracowano trzy standardy języka SQL, kolejno rozszerzające jego



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

funkcjonalność. Standardy te to: SQL-92, SQL-99, SQL-2003.

Przykład prostego polecenia SQL będącego zapytaniem do bazy danych przedstawiono na slajdzie. Zapytanie to wyszukuje pracowników (nazwisko, etat, płaca) zatrudnionych w zespole o numerze 30 na etacie kierownika.

Slajd 24

**SQL**  
Structured Query Language

## Interakcje z bazą danych

- **Aplikacje**
  - **formularze**
    - elektroniczne formularze z polami, listami,
    - elementami wyboru
    - umożliwiają wstawianie, modyfikowanie, usuwanie,
    - wyszukiwanie danych
  - **raporty**
    - umożliwiają prezentowanie zawartości bazy danych (teksty, wykresy, grafika)

DAILY GROUP Lekcja 1 - Bazy danych, podstawowe pojęcia

Język SQL jest narzędziem dostępu do bazy danych stosowanym głównie przez projektantów aplikacji, projektantów baz danych i administratorów baz danych. Standardowym sposobem korzystania z bazy danych przez użytkowników końcowych są aplikacje. Należy jednak pamiętać, że na poziomie programistycznym aplikacje również komunikują się z bazą danych za pomocą poleceń SQL.

Ze względu na funkcjonalność, wyróżnia się dwa rodzaje aplikacji, tj. formularze i raporty. Aplikację pierwszego rodzaju należy postrzegać jako elektroniczny formularz (z polami, listami, elementami wyboru) wypełniany przez użytkownika. Formularze umożliwiają pełną obsługę danych, tj. wstawianie, modyfikowanie, usuwanie i wyszukiwanie.

Raporty umożliwiają wyłącznie odczytywanie danych z bazy i prezentowanie ich w różnej postaci, głównie tekstu lub wykresu.

Slajd 25

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

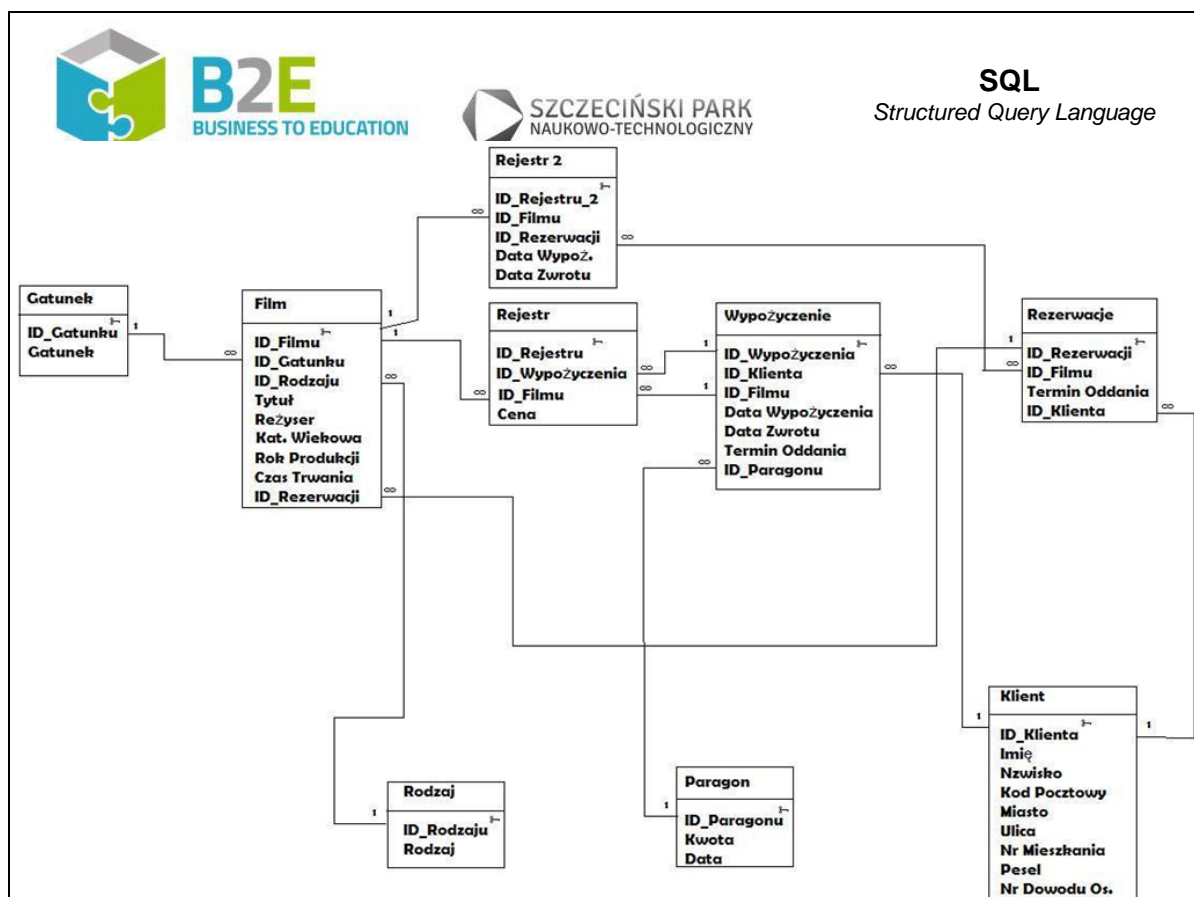
## Ćwiczenie

- Baza danych wypożyczalni filmów DVD/VHS

DAILY GROUP Lekcja 1 - Bazy danych, podstawowe pojęcia

Próba zaprojektowania bazy danych wypożyczalni filmów razem z uczniami, użycie pojęć zdefiniowanych w pierwszej części lekcji.

Slajd 26



Przykład rozwiązania ćwiczenia



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 27

Slide 27 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenie' (Exercise). The exercise text asks for a list of databases that do not exist in the current world. The footer includes the DAILY GROUP logo and the text 'Lekcja 1 - Bazy danych, podstawowe pojęcia'.

Pytanie do uczniów o podanie najważniejszych ich zdaniem baz danych w dzisiejszej rzeczywistości. Przykłady odpowiedzi:

- Rachunki bankowe
- Mapy – nawigacje samochodowe
- Bazy urzędu skarbowego
- Bazy NFZ – np. eWUŚ
- Baza stron WWW – np. Google

Slajd 28

Slide 28 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenie' (Exercise). The exercise text asks for a list of databases that contain their own data. The footer includes the DAILY GROUP logo and the text 'Lekcja 1 - Bazy danych, podstawowe pojęcia'.

Pytanie do uczniów o podanie baz danych, zawierających ich dane. Przykłady odpowiedzi:

- Rachunki bankowe
- Mapy – nawigacje samochodowe
- Baza uczniów szkoły
- NFZ
- Baza Urzędu Miasta

Slajd 29

Slide 29 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenie' (Exercise). The exercise text asks for the smallest unit in a database. The footer includes the DAILY GROUP logo and the text 'Lekcja 1 - Bazy danych, podstawowe pojęcia'.

Odpowiedź prawidłowa: Dana





**SQL**  
Structured Query Language

---

## Ćwiczenie

- **Jaka Twoim zdaniem baza danych jest największa na świecie?**
  - Biblioteka Kongresu
  - World Data Centre for Climate
  - YouTube
  - ChoicePoint
  - Google
  - CIA
  - China Mobile



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI

Człowiek - najlepsza inwestycja



UNIA EUROPEJSKA  
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



Lekcja 1 - Bazy danych, podstawowe pojęcia

### Biblioteka Kongresu

To chyba akurat nikogo nie dziwi. Biblioteka Kongresu przechowuje 130 milionów dokumentów, wśród których można znaleźć autentycznie wszystko – od archiwalnych książek kucharskich aż po egzemplarze gazet z czasów kolonialnych. Gdyby wycisnąć z nich wszystkie literki, same słowa zajęłyby 20 terabajtów. BC rozrasta się o 10 000 egzemplarzy dziennie, a do przechowywania tego wszystkiego potrzeba pólki o łącznej długości aż 530 mil. Pięć milionów dokumentów jest już dostępnych w wersji cyfrowej.

### World Data Centre for Climate

Którym zarządzają Meteorologiczny Instytut Maxa Plancka oraz Niemieckie Centrum Komputerowej Analizy Klimatu. WDCC chwali się tym, że posiada bazę danych o objętości ponad 220 terabajtów, do których można dostać się przez internet oraz dodatkowo 110 terabajtów symulacji klimatycznych, a także sześć petabajtów przechowywanych na taśmach magnetycznych.

### YouTube

Przy powstawaniu serwisu, przewidywano że nigdy nie będzie on wymagał więcej niż 45 terabajtów. Ze względu na to, że filmiki wrzucane przez użytkowników różnią się od siebie długością i rozmiarem, ciężko oszacować rozmiar bazy danych YT dzisiaj, ale "magiczne" 45 terabajtów zostało z pewnością już dawno przekroczone. Co godzinę na YouTube pojawia się 60 godzin wideo, oglądane są na nim cztery miliardy filmów dziennie, co daje łącznie ponad trzy miliardy godzin oglądania na miesiąc. Robi wrażenie.

### ChoicePoint

Jedna z dziwniejszych firm, jakie kiedykolwiek istniały. ChoicePoint, zlokalizowana niedaleko

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Atlanty, zajmowała się zbieraniem danych osobowych, które potem sprzedawała tym, którzy oferowali najwyższą cenę. Udało jej się wejść w posiadanie 17 miliardów dokumentów na temat 250 milionów osób – gdyby je wydrukować sięgnęłyby z Ziemi do Księżyca i z powrotem 77 razy. Dane zbierane przez ChoicePoint zawierały wszystko na temat Amerykanów – rejestry karanych, adresy, numery telefonów, wydruki z kart kredytowych i tak dalej, i tak dalej. Niestety okazało się przy okazji, że wyciekały z firmy, a do tego wielokrotnie zdarzało się, że podawane przez CP informacje okazywały się nie do końca poprawne, co potrafiło doprowadzić do tego, że niektóre osoby traciły pracę, kiedy okazywało się na przykład, że wbrew temu, co mówiły, były jednak karane.

### **Google**

To też chyba nikogo tutaj nie dziwi. Jednak dane o dokładnych rozmiarach baz danych Google`a nie są znane, bo firma ich nie ujawnia. To co jednak wiadomo, to to, że centra danych korporacji z Mountain View pożarły w 2010 roku kilowatogodzin czyli połowę tego, co produkuje w ciągu roku słynna Zapora Hoovera – tama wysoka na ponad 220 metrów i długa na blisko 400. I że każdego dnia Google analizuje 24 petabajty danych. Żeby je wydrukować na papierze, trzeba by ściąć ponad 1,2 miliona drzew.

### **CIA**

Tutaj małe zaskoczenie, aczkolwiek w sumie to żadne zaskoczenie – bazy danych CIA są uznawane za jedne z największych na świecie, choć nikt nie ma zielonego pojęcia, ile informacji się w nich znajduje. A wszystko to dlatego, że większość z nich jest utajnionych.

### **China Mobile**

Tutaj też dokładnie nie wiadomo, niełatwo wyciąga się takie informacje od Chińczyków. Ale swego czasu jedną z największych baz danych miał znany operator telefonii komórkowej Sprint, który z 53 milionami użytkowników posiadał bazę danych zawierającą blisko trzy miliardy rekordów. A China Mobile oferuje swoje usługi 70% Chińczyków, mając tym samym 655 milionów klientów. Poczujcie cenną różnicę.

## **Ćwiczenia**

Ćwiczenie zostało przedstawione na slajdzie 25: Próba wspólnego zaprojektowania bazy danych wypożyczalni filmów DVD, wykorzystanie pojęć przedstawionych w czasie lekcji oraz dodatkowo na slajdach 26-30.

## **Opis założonych osiągnięć ucznia**

Po tej lekcji uczniowie poznają definicję bazy danych, podstawowe pojęcia z nią związane. Będą potrafili wskazać przykłady baz danych.

## **Lekcja 2 - Języki programowania serwerów SQL, środowisko programowania w języku SQL, wprowadzenie do języka SQL**

### **Cel lekcji**

Celem lekcji jest poznanie różnych rozszerzeń języka SQL, stosowanych przez producentów systemów zarządzania bazą danych. Uczniowie zostaną się ze środowiskiem Microsoft SQL Management Studio Express. Zostaną poinstruowani w jaki sposób można pobrać i zainstalować środowisko.



W drugiej części lekcji uczniowie poznają podstawowe kwerendy (zapytania) SQL, a w ramach ćwiczeń – spróbują praktycznie wykorzystać wiedzę.

## Treść - slajdy z opisem

Slajd 1

**SQL**  
Structured Query Language

# SQL

## Structured Query Language

Lekcja 2

Języki programowania SQL,  
środowisko programowania,  
wprowadzenie do języka SQL

Celownik - najlepsza inwestycja

KAPITAŁ LUDZKI

PROJEKT WSPÓLFINANSOWANY PRZEZ UNIĘ EUROPEJSKĄ W RAMACH EUROPEJSKIEGO FUNDUSZU SPOŁECZNEGO

**DAILY GROUP** SQL Structured Query Language Lekcja 2

SQL jest najpopularniejszym językiem programowania baz danych. W tej lekcji poznamy różne rodzaje języka SQL, oraz poznamy podstawy tego języka.

Slajd 2

**SQL**  
Structured Query Language

## Przypomnienie

- Baza danych
- System Zarządzania Bazą Danych
- Tabela, wiersz, kolumna
- Relacje

**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Slajd 3

**SQL**  
Structured Query Language

## Języki programowania SQL

- SQL jest powszechnie stosowanym językiem programowania baz danych. Istnieją różne rozszerzenia tego języka stosowane przez głównych producentów SZDB: SQL Server, Oracle MySQL, PostgreSQL, DB2, Informix, itd.:
  - PL/SQL jest językiem proceduralnym używanym przez Oracle
  - PL/pgSQL jest językiem proceduralnym używanym przez PostgreSQL
  - TSQL (transakcyjny SQL), rozszerzenie języka SQL umożliwiające tworzenie pętli, instrukcji warunkowych i zmiennych, stworzone przez Sysbase i wykorzystywane przez Microsoft in SQL Server.

**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Wyjaśnienie, że pomimo wspólnego standardu istnieje wiele różnych rozszerzeń. Podstawowa składnia wszystkich rozszerzeń języka jest taka sama, jednak istnieje wiele różnic. W tym kursie zajmiemy się językiem SQL w ujęciu rozszerzenia TSQL, promowanym przez Microsoft.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4

**Środowisko programowania**

- Microsoft SQL Server Management Studio
- Wersja bezpłatna – Express
- Toad for SQL Server Freeware

Microsoft SQL Server Management Studio Express

Toad World  
Collaboration. Education. Expertise.

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Środowiska programowania TSQL. Microsoft oferuje wraz z bazą danych SQL Server narzędzie Microsoft SQL Server Management Studio (często używa się skrótu MSSMS). Możliwe jest pobranie wersji bezpłatnej oznaczonej przez producenta hasłem „Express”. Poza oficjalnym narzędziem dostępne są również narzędzia innych firm, np. Toad (również darmowe).

Slajd 5

**MS SQL Management Studio Express**

- Do pobrania ze strony Microsoft
  - <http://www.microsoft.com/en-us/download/details.aspx?id=29062>

Microsoft® SQL Server® 2012 Express

Quick links  
 Overview  
 System requirements  
 Instructions  
 Additional information

Looking for support?  
 Visit the Microsoft Support site now >

Microsoft® SQL Server® 2012 Express is a powerful and reliable free data management system that delivers a rich and reliable data store for lightweight Web Sites and desktop applications.

Quick details  
 Version: 11.0.2100.80 Date published: 5/14/2012  
 Change language: English

Files in this download  
 The links in this section correspond to files available for this download. Download the files appropriate for you.

| File name                              | Size     | DOWNLOAD |
|----------------------------------------|----------|----------|
| ENU\i64\SQLSERVER_64_ENU.exe           | 132.1 MB | DOWNLOAD |
| ENU\i64\SQLSERVER_64_ENU.exe           | 1.3 GB   | DOWNLOAD |
| ENU\i64\SQLSERVER_64_ENU.exe           | 689.9 MB | DOWNLOAD |
| ENU\i64\SqlLocalDB.MSI                 | 33.0 MB  | DOWNLOAD |
| ENU\i64\SQLManagementStudio_64_ENU.exe | 600.2 MB | DOWNLOAD |

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

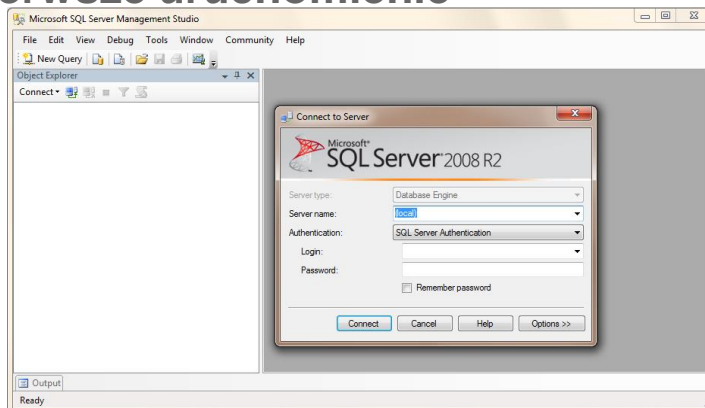
Management Studio wraz z bazą danych można pobrać ze strony Microsoftu. Adres podany jest na slajdzie. Warto zwrócić uwagę aby pobrać wersję SZDB wraz z MS SQL SMS: ENU\i64\SQLSERVER\_64\_ENU.exe:

„Express with Tools (with LocalDB) Includes the database engine and SQL Server Management Studio Express)

This package contains everything needed to install and configure SQL Server as a database server. Choose either LocalDB or Express depending on your needs above.”

## MS SQL Management Studio Express

- Pierwsze uruchomienie




Po instalacji oprogramowania należy się zalogować do bazy danych: wybieramy serwer lokalny, podajemy użytkownika i hasło, które zostało zdefiniowane w procesie instalacji.


Domyślnym użytkownikiem administracyjnym jest sa (super administrator).



Slajd 7



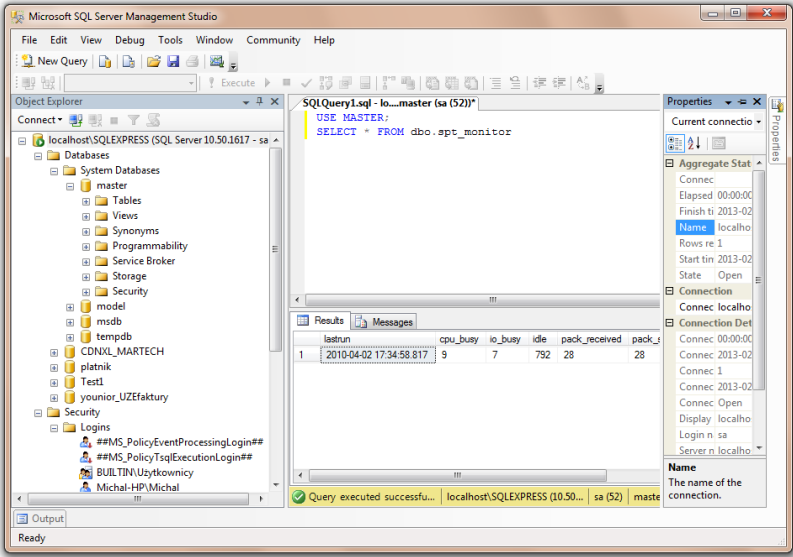
**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language


## MS SQL Management Studio Express




**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Opis elementów programu MS SQL SMS. Okno programu składa się z sekcji obiektów bazy danych po lewej stronie, rekacji roboczej (wykorzystywanej na przykład do wpisywania zapytań TSQL) na środku, poniżej – okno rezultatów zapytań TSQL oraz okna właściwości po prawej stronie.

Slajd 8



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Wprowadzenie do języka SQL

- **SQL to język dostępu do bazy danych**
- **Grupy poleceń języka:**
  - DDL - język definiowania danych (Data Definition Language)
  - DML - Język manipulowania danych (Data Manipulation Language)
  - DCL - Język sterowania danych (Data Control Language)
  - Język zapytań (Query Language)

**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

- Język definiowania danych (Data Definition Language - DDL), który umożliwia definiowanie struktury danych zawartych w bazie.
- Język manipulowania danych (Data Manipulation Language - DML), który umożliwia wypełnienie, modyfikowanie i usuwanie danych z bazy.
- Język sterowania danych (Data Control Language), który umożliwia sterowanie transakcjami tj. akceptacja lub wycofanie.
- Język zapytań (Query Language), który umożliwia pobieranie informacji z bazy za pośrednictwem określonych zapytań, warunków.

Slajd 9

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Wprowadzenie do języka SQL

- Polecenie SQL może być zapisane:
  - w jednym bądź wielu wierszach
  - dużymi lub małymi literami
- Polecenia powinny być zakończone średnikiem

```
SELECT * FROM pracownicy;

select * from PRACOWNICY;
```

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

W SQL występuje dowolność wielkich i małych liter. W ramach ujednoczenia rozszerzeń języków SQL, Microsoft dopuszcza stosowanie średnika na końcu polecenia (w PL/SQL jest wymagany).

Slajd 10

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Wyświetlanie elementów (projekcja)

- Wybór wartości z tabeli, np.

```
SELECT imie, nazwisko, pesel FROM pracownicy;
```

- Operatory arytmetyczne, np. +, -, \*, /

```
SELECT nazwisko, placa_zasadnicza*3, premia + 300 FROM pracownicy;
```

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Przykład zapytania SELECT: pobranie imion i nazwisk wszystkich pracowników. W poleceniu tłumaczonym z j. angielskiego Wybierz imię, nazwisko, pesel z tabeli pracownicy, występują nazwy kolumn (atrybuty) oraz nazwa tabeli (pracownicy).

W wyrażeniach SELECT można korzystać z operacji arytmetycznych takich jak dodawanie, odejmowanie, mnożenie i dzielenie.

Slajd 11

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Aliasy

- Alternatywna nazwa atrybutu
- Można stosować słowo AS

```
SELECT nazwisko,
 placa_zasadnicza*1.2 AS nowa_placa,
 premia + 300 premia_swiateczna
FROM pracownicy;
```

|   | nazwisko | nowa_placa | premia_swiateczna |
|---|----------|------------|-------------------|
| 1 | Kowalski | 2760.000   | 550.00            |
| 2 | Nowak    | 3240.000   | 400.00            |

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

W celu wyświetlenia kolumn o nazwach innych niż kolumny tabeli można stosować aliasy. Alias wpisujemy po nazwie kolumny lub po słowie kluczowym AS.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**

Structured Query Language

## NULL

- Wartość pusta w tabeli
- Wynikiem jakiegokolwiek operacji z NULL jest NULL

```
SELECT * FROM pracownicy;
```

|   | nazwisko   | imie    | placa_zasadnicza | premia | pesel       | stanowisko  |
|---|------------|---------|------------------|--------|-------------|-------------|
| 1 | Kowalski   | Jan     | 2300.00          | 250.00 | 82091104357 | Manager     |
| 2 | Nowak      | Karol   | 2700.00          | 100.00 | 80010123987 | Specjalista |
| 3 | Przepiórka | Marzena | 2700.00          | NULL   | 89121203456 | Specjalista |

```
SELECT nazwisko,
placa_zasadnicza*1.2 AS nowa_placa,
premia + 300) premia_swiateczna
FROM pracownicy;
```


|   | nazwisko   | nowa_placa | premia_swiateczna |
|---|------------|------------|-------------------|
| 1 | Kowalski   | 2760.000   | 550.00            |
| 2 | Nowak      | 3240.000   | 400.00            |
| 3 | Przepiórka | 3240.000   | NULL              |




Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

W bazach danych występuje wartość pusta NULL. Nie jest to 0, czy pusty ciąg znaków – tylko wartość nieokreślona. Wynikiem dodawania, odejmowania, mnożenia czy dzielenia wartości NULL jest również wartość NULL, co zaprezentowano na przykładzie.

Ciekawostka: w Oracle w przeciwieństwie do MSSQL NULL traktowany jest jak pusty string.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language


---

## ISNULL

- ISNULL(*kolumna, wartość\_w\_przypadku\_NULL*)
- Możliwość zmiany wartości pustej na inną, np.

```
SELECT nazwisko,
placa_zasadnicza*1.2 AS nowa_placa,
ISNULL(premia,0) + 300 premia_swiateczna
FROM pracownicy;
```

|   | nazwisko   | nowa_placa | premia_swiateczna |
|---|------------|------------|-------------------|
| 1 | Kowalski   | 2760.000   | 550.00            |
| 2 | Nowak      | 3240.000   | 400.00            |
| 3 | Przepiórka | 3240.000   | 300.00            |



Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Możliwością uniknięcia problemów z kolumnami zawierającymi wartość NULL jest polecenie ISNULL, które ma dwa parametry. Pierwszym jest nazwa kolumny, drugim wartość jaka ma być podstawiona w przypadku gdy kolumna zawiera wartość NULL.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Porządkowanie elementów

- Klauzula ORDER BY
- Słowa kluczowe DESC i ASC
- ORDER BY występuje zawsze na końcu zapytania

```
SELECT imie, nazwisko, pesel FROM pracownicy
ORDER BY nazwisko DESC, imie ASC;
```



Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Sortowanie wyników uzyskuje się przy wykorzystaniu klauzuli ORDER BY. Sortowanie rosnące (ascending) jest domyślne. Po nazwie kolumny można użyć słów kluczowych ASC i DESC w celu uzyskania oczekiwanej kolejności.

Slajd 15

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Eliminowanie duplikatów

- Polecenie DISTINCT

```
SELECT DISTINCT stanowisko FROM pracownicy;
```

Results Messages

| stanowisko    |
|---------------|
| 1 Manager     |
| 2 Specjalista |

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Słowo kluczowe DISTINCT musi występować zaraz po słowie kluczowym SELECT.

SELECT stanowisko FROM pracownicy

Takie zapytanie wyświetli wszystkie stanowiska obejmowane wśród pracowników. Jeżeli pojawią się dwa takie same stanowiska, tylko jedno zostanie wyświetlone. Słowo DISTINCT eliminuje wiersze, które posiadają duplikaty we wszystkich kolumnach wyspecyfikowanych w wyrażeniu SELECT. Tylko jedno słowo DISTINCT może zostać użyte w całym zapytaniu SELECT.

Slajd 16

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Selekcja wybranych danych

- Klauzula WHERE
- Ogólna składnia:
- SELECT kolumna1, kolumna2, kolumna3  
FROM tabela  
WHERE kolumna operator wartość
- Operatory: =, !=, >, >=, <, <=

```
SELECT imie, nazwisko, pesel FROM pracownicy
WHERE placa_zasadnicza > 2000
```

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Używając klauzuli WHERE możemy zawęzić zwracane wartości do tylko tych które spełniają wybrane kryteria. Np. kiedy chcemy znaleźć pracowników zarabiających więcej niż 2000zł.



Slajd 17

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

## Selekcja wybranych danych

- Operatory **BETWEEN ... AND ...**

```
SELECT imie, nazwisko, pesel FROM pracownicy
WHERE placa_zasadnicza BETWEEN 2000 AND 3000
```
- **IN**

```
SELECT imie, nazwisko, pesel FROM pracownicy
WHERE stanowisko IN ('Specjalista', 'Manager')
```

**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Operator BETWEEN AND umożliwia wybór wartości z danego zakresu, operator IN wybór wartości zawierających się w zdefiniowanym zbiorze

Slajd 18

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

## Selekcja wybranych danych

- **LIKE**

```
SELECT imie, nazwisko, pesel FROM pracownicy
WHERE nazwisko LIKE 'M%';

SELECT imie, nazwisko, pesel FROM pracownicy
WHERE nazwisko LIKE '%ski';
```
- **IS NULL**

```
SELECT imie, nazwisko, pesel FROM pracownicy
WHERE premia IS NULL;
```

**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Operator LIKE umożliwia wybór wartości które zawierają pewien ciąg znaków, np. zaczynają się od litery „M”, lub kończą na „ski”.

Slajd 19

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

## Selekcja wybranych danych

- Negacja operatorów SQL
  - NOT BETWEEN ... AND ...
  - NOT IN
  - NOT LIKE
  - IS NOT NULL

```
SELECT imie, nazwisko, pesel FROM pracownicy
WHERE stanowisko NOT IN ('Specjalista', 'Manager');
```

**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Negowanie warunków jest możliwe za pomocą słowa kluczowego NOT. NOT występuje przed operatorem. Wyjątkiem jest operator IS NULL, gdzie negator występuje w środku: IS NOT NULL.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 20

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Selekcja wybranych danych

- Łączenie warunków
  - AND
  - OR

```
SELECT imie, nazwisko, pesel FROM pracownicy
WHERE stanowisko NOT IN ('Specjalista', 'Manager')
AND placa_zasadnicza > 2000;
```

DAILY Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Operatory logiczne mogą być stosowane jednocześnie w tej samej klauzuli WHERE. Służą do tego operatory logiczne AND i OR. AND posiada wyższy priorytet niż OR, a zmiana priorytetu jest możliwa za pomocą nawiasów.

Slajd 21

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Ćwiczenia

|   | nazwisko    | imie      | placa_zasadnicza | premia | pesel       | stanowisko  |
|---|-------------|-----------|------------------|--------|-------------|-------------|
| 1 | Kowalski    | Jan       | 2300.00          | 250.00 | 82091104357 | Manager     |
| 2 | Nowak       | Karol     | 2700.00          | 100.00 | 80010123987 | Specjalista |
| 3 | Przepiórka  | Marzena   | 2700.00          | NULL   | 89121203456 | Specjalista |
| 4 | Burzych     | Paweł     | 1900.00          | 500.00 | 78032309123 | Specjalista |
| 5 | Makłowicz   | Marek     | 2000.00          | NULL   | 54013112345 | Specjalista |
| 6 | Naramowicka | Magdalena | 2100.00          | 200.00 | 77121312098 | Specjalista |
| 7 | Witos       | Jacek     | 3000.00          | 500.00 | 69100967234 | Manager     |

DAILY Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

**Ćwiczenie 1**

Napisz zapytanie zwracające wszystkich Managerów

**Ćwiczenie 2**

Napisz zapytanie zwracające wszystkich specjalistów, których łączna płaca jest wyższa niż 2300zł

**Ćwiczenie 3**

Napisz zapytanie wyświetlające wszystkie kobiety wśród pracowników (końcówka imienia „a”)

**Ćwiczenie 4**

Napisz zapytanie wyświetlające pracowników, których roczna płaca zasadnicza jest większa niż 25.000zł

**Ćwiczenie 5**

Napisz zapytanie wyświetlające pracowników, którzy nie otrzymują premii

**Ćwiczenie 6**

Napisz zapytanie wyświetlające pracowników, których premia jest pomiędzy 200 a 300zł

**Ćwiczenie 7**

Napisz zapytanie wyświetlające łączny roczny dochód pracowników

**Ćwiczenie 8**

Napisz zapytanie wyświetlające pracowników urodzonych w latach 80-tych

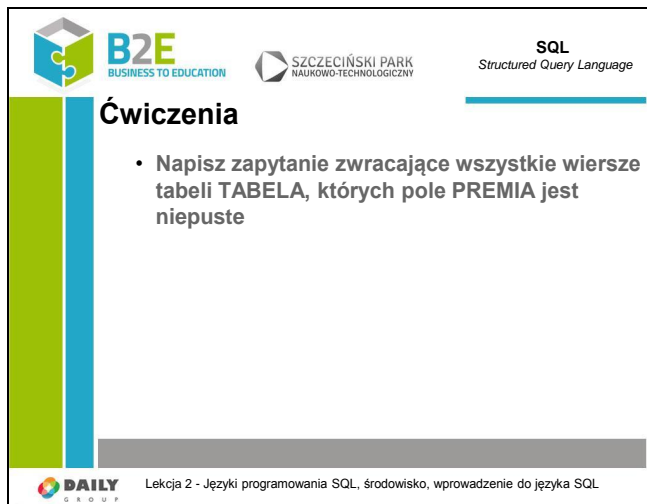
### Ćwiczenie 9

Napisz zapytanie wyświetlające specjalistów urodzonych w latach 80-tych, zarabiających więcej niż 2.700zł

### Ćwiczenie 10

Napisz zapytanie wyświetlające pracowników, których imię i nazwisko zaczyna się na literę M

Slajd 22



Slide 22 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). Below the header, the title 'Ćwiczenia' is followed by a single bullet point: 'Napisz zapytanie zwracające wszystkie wiersze tabeli TABELA, których pole PREMIA jest niepuste'. At the bottom, there is a footer with the DAILY GROUP logo and the text 'Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL'.

```
SELECT * FROM TABELA
WHERE PREMIA IS NOT NULL
```

Slajd 23



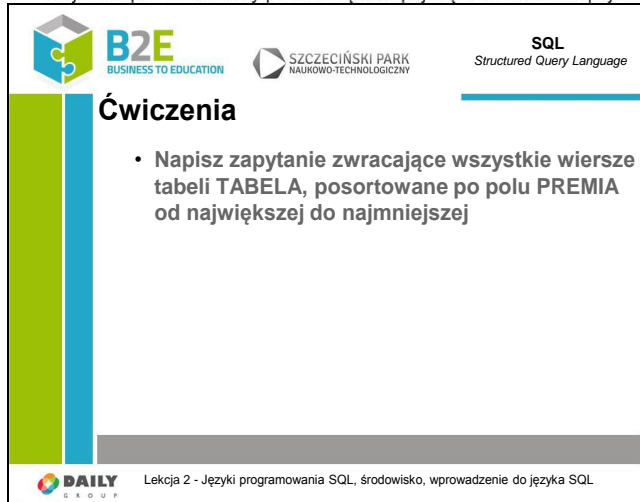
Slide 23 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). Below the header, the title 'Ćwiczenia' is followed by a single bullet point: 'Napisz zapytanie zwracające wszystkie wiersze tabeli TABELA, których pole PREMIA jest większe od 1000 i mniejsze od 2000'. At the bottom, there is a footer with the DAILY GROUP logo and the text 'Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL'.

```
SELECT * FROM TABELA
WHERE PREMIA IS BETWEEN
1000 AND 2000
```



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 24



SQL  
Structured Query Language

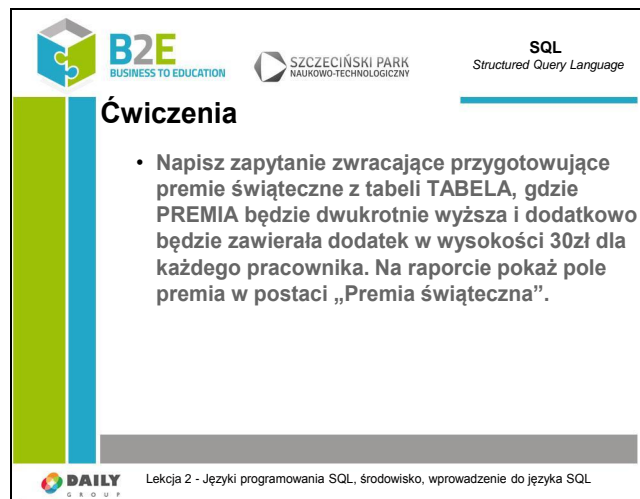
### Ćwiczenia

- Napisz zapytanie zwracające wszystkie wiersze tabeli TABELA, posortowane po polu PREMIA od największej do najmniejszej

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

```
SELECT * FROM TABELA
ORDER BY PREMIA DESC
```

Slajd 25



SQL  
Structured Query Language

### Ćwiczenia

- Napisz zapytanie zwracające przygotowane premie świąteczne z tabeli TABELA, gdzie PREMIA będzie dwukrotnie wyższa i dodatkowo będzie zawierała dodatek w wysokości 30zł dla każdego pracownika. Na raporcie pokaż pole premia w postaci „Premia świąteczna”.

DAILY GROUP Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

```
SELECT (PREMIA*2)+30 AS
[Premia świąteczna] FROM
TABELA
```



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Podsumowanie

|          |                                                         |
|----------|---------------------------------------------------------|
| SELECT   | Wybiera listę kolumn                                    |
| alias    | Można stosować tylko do kolumn i wyrażeń (nie do *)     |
| *        | Oznacza wszystkie kolumny                               |
| DISTINCT | Eliminuje duplikaty ze zbioru wynikowego                |
| FROM t   | Określa relację, z której odczytujemy dane              |
| WHERE    | Określa warunki wyboru wierszy, zawiera wartości kolumn |
| AND/OR   | Łączy warunki w klauzuli WHERE                          |
| ()       | Pozwala na zmianę priorytetu operatorów                 |
| ORDER BY | Służy do określenia kryterium sortowania                |
| ASC      | Rosnący porządek sortowania                             |
| DESC     | Malejący porządek sortowania                            |

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA  
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



**DAILY**  
GROUP

Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

Tabela podsumowująca poznane wyrażenia

### Ćwiczenia

Ćwiczenie zostało przedstawione na slajdzie 21. Uczniowie w ramach ćwiczenia będą musieli napisać zapytania SQL z tabel przygotowanych przez szkoleniowca. Dodatkowe ćwiczenia przedstawiono na slajdach 22-25.

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą wiedzieli jak pobrać i zainstalować środowisko MS SMS. Dowiedzą się jak konstruować proste zapytania SQL, polegające na wyświetlaniu informacji z bazy, sortowaniu, definiowaniu podstawowych warunków zapytania.

### Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

#### Cel lekcji

Celem lekcji jest zaznajomienie uczniów z bardziej skomplikowanymi zapytaniem SQL niż w poprzedniej lekcji. Uczniowie dowiedzą się, jakie są sposoby łączenia danych z wielu tabel. W drugiej części lekcji przedstawione zostaną zagadnienia związane z funkcjami agregującymi oraz grupowaniem danych. Dzięki ćwiczeniom praktycznym, uczniowie praktycznie wykorzystają zdobytą w czasie lekcji wiedzę.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Treść - slajdy z opisem

Slajd 1

SQL  
Structured Query Language

# SQL

## Structured Query Language

Lekcja 3  
Złożone zapytania,  
grupowanie, agregacja danych

Celownik - najlepsza inwestycja

KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPOJNOŚCI

UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

DAILY GROUP SQL Structured Query Language Lekcja 3

Na poprzedniej lekcji zajmowaliśmy się prostymi zapytaniami do bazy danych, teraz przejdziemy do bardziej złożonych zapytań, nauczymy się łączyć tabele, grupować dane. Dowiemy się jak wybierać wartości minimalne i maksymalne, nauczymy się sumować.

Slajd 2

SQL  
Structured Query Language

## Przypomnienie

- Polecenie SELECT
- WHERE
- ORDER BY
- DISTINCT
- Alias

DAILY GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Powtórzenie materiału z poprzedniej lekcji.

Slajd 3

SQL  
Structured Query Language

## Ograniczanie wyników

```
SELECT TOP 3 imie, nazwisko, pesel FROM pracownicy
```



- Polecenie TOP:
  - Używana zaraz po SELECT
  - Wartość liczbowa, np. TOP 3
  - Wartość procentowa, np. TOP 10 PERCENT

DAILY GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Ograniczanie wyników w SQL realizowane jest słowem TOP. Możliwe jest ograniczenie ilościowe i procentowe.

Używanie TOP bez klauzuli ORDER BY jest mocno niezalecane z uwagi na możliwą przypadkowość wyniku!

Ciekawostka: Od MSSQL 2012 są też bardziej praktyczna metody tj. OFFSET (ilość rekordów do pominięcia) i FETCH ilość rekordów do pobrania.

**SQL**  
Structured Query Language

## Komentarze w SQL


- „--” – pojedynczy wiersz
- „/\*” i „\*/” – cała sekcja

```

SELECT TOP 3
 imie,
 --nazwisko,
 pesel
FROM pracownicy



SELECT TOP 3
 /*imie,
 nazwisko, */
 pesel
FROM pracownicy

```



Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Język SQL umożliwia stosowanie komentarzy. W celu komentowania jednego wiersza stosuje się dwa minusy. Można również zakomentować całą sekcję – zaczynamy komentarz od ukośnika i gwiazdki, kończymy komentarz gwiazdką i ukośnikiem.


**SQL**  
Structured Query Language

## Łączenie tabel

| dbo.pracownicy |           |                  |        |             |             |         |
|----------------|-----------|------------------|--------|-------------|-------------|---------|
| nazwisko       | imie      | placa_zasadnicza | premia | pesel       | stanowisko  | miejsce |
| Kowalski       | Jan       | 2300,00          | 250,00 | 82091104357 | Manager     | 1       |
| Nowak          | Karol     | 2700,00          | 100,00 | 80010123987 | Specjalista | 1       |
| Przepiórka     | Marzena   | 2700,00          | NULL   | 89121203456 | Specjalista | 3       |
| Burzych        | Paweł     | 1900,00          | 500,00 | 78032309123 | Specjalista | 2       |
| Makłowicz      | Marek     | 2000,00          | NULL   | 54013112345 | Specjalista | 2       |
| Naramowicka    | Magdalena | 2100,00          | 200,00 | 77121312098 | Specjalista | 1       |
| Witos          | Jacek     | 3000,00          | 500,00 | 69100967234 | Manager     | 3       |
| * NULL         | NULL      | NULL             | NULL   | NULL        | NULL        | NULL    |

| dbo.miejsca |             |       |          |        |                |
|-------------|-------------|-------|----------|--------|----------------|
| nr_miejsca  | ulica       | numer | miasto   | kod    | telefon        |
| 1           | Mała        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2           | Wąska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3           | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |



Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Na slajdzie widzimy dwie tabele – znaną z poprzednich lekcji „pracownicy” oraz „miejsca” możemy sobie wyobrazić potrzebę połączenia tych tabel, np. w celu przygotowania zestawienia – pracownicy wraz z miejscem zamieszkania.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6



**B2E**  
BUSINESS TO EDUCATION



**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Łączenie tabel

| dbo.pracownicy |           |                  |        |             |             |        |
|----------------|-----------|------------------|--------|-------------|-------------|--------|
| id             | imie      | placa_zasadnicza | premia | pesel       | stanowisko  | miesto |
| 1              | Jan       | 2300,00          | 250,00 | 82091104357 | Manager     | 13     |
| 2              | Karol     | 2700,00          | 300,00 | 80010123987 | Specjalista | 1      |
| 3              | Marzena   | 2700,00          | NEEL   | 89121203456 | Specjalista | 3      |
| 4              | Pawel     | 1900,00          | 500,00 | 78032309123 | Specjalista | 2      |
| 5              | Marek     | 2000,00          | NEEL   | 54013112345 | Specjalista | 2      |
| 6              | Magdalena | 2100,00          | 200,00 | 77121312098 | Specjalista | 1      |
| 7              | Jacek     | 3000,00          | 500,00 | 69100967234 | Manager     | 3      |
| 8              | NEEL      | NEEL             | NEEL   | NEEL        | NEEL        | NEEL   |

| dbo.miejsca |             |       |          |        |                |
|-------------|-------------|-------|----------|--------|----------------|
| id          | ulica       | numer | miasto   | kod    | telefon        |
| 1           | Mala        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2           | Waska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3           | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |

| id | imie      | nazwisko    | pesel       | ulica       | numer | miasto   |
|----|-----------|-------------|-------------|-------------|-------|----------|
| 1  | Jan       | Kowalski    | 82091104357 | Mala        | 13    | Poznań   |
| 2  | Karol     | Nowak       | 80010123987 | Mala        | 13    | Poznań   |
| 3  | Marzena   | Przepiórka  | 89121203456 | Grunwaldzka | 34    | Warszawa |
| 4  | Pawel     | Burzych     | 78032309123 | Waska       | 4/12A | Kraków   |
| 5  | Marek     | Makłowicz   | 54013112345 | Waska       | 4/12A | Kraków   |
| 6  | Magdalena | Naramowicka | 77121312098 | Mala        | 13    | Poznań   |
| 7  | Jacek     | Witos       | 69100967234 | Grunwaldzka | 34    | Warszawa |



Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Często w praktyce musimy połączyć wyniki z kilku tabel. Wyobraźmy sobie zadanie – przygotować raport pracowników wraz z ich miejscem pracy.





## Złączenia JOIN oraz CROSS JOIN

```
SELECT imie, nazwisko, pesel, ulica, numer, miasto
FROM pracownicy, miejsca
SELECT * FROM pracownicy CROSS JOIN miejsca
```

- **Zbiór wszystkich możliwych kombinacji rekordów z obu tabel**

|    |           | nazwisko  | pesel       | ulica       | numer  | miasto   |
|----|-----------|-----------|-------------|-------------|--------|----------|
| 1  | Jan       | Kowalski  | 82091104357 | Mała        | 13     | Poznań   |
| 2  | Karol     | Nowak     | 80010123987 | Mała        | 13     | Poznań   |
| 3  | Marzena   | Przepi... | 89121203456 | Mała        | 13     | Poznań   |
| 4  | Paweł     | Burzych   | 78032309123 | Mała        | 13     | Poznań   |
| 5  | Marek     | Maklo...  | 54013112345 | Mała        | 13     | Poznań   |
| 6  | Magdalena | Naram...  | 77121312098 | Mała        | 13     | Poznań   |
| 7  | Jacek     | Witos     | 69100967234 | Mała        | 13     | Poznań   |
| 8  | Jan       | Kowalski  | 82091104357 | Wąska       | 4/1... | Kraków   |
| 9  | Karol     | Nowak     | 80010123987 | Wąska       | 4/1... | Kraków   |
| 10 | Marzena   | Przepi... | 89121203456 | Wąska       | 4/1... | Kraków   |
| 11 | Paweł     | Burzych   | 78032309123 | Wąska       | 4/1... | Kraków   |
| 12 | Marek     | Maklo...  | 54013112345 | Wąska       | 4/1... | Kraków   |
| 13 | Magdalena | Naram...  | 77121312098 | Wąska       | 4/1... | Kraków   |
| 14 | Jacek     | Witos     | 69100967234 | Wąska       | 4/1... | Kraków   |
| 15 | Jan       | Kowalski  | 82091104357 | Grunwaldzka | 34     | Warszawa |
| 16 | Karol     | Nowak     | 80010123987 | Grunwaldzka | 34     | Warszawa |
| 17 | Marzena   | Przepi... | 89121203456 | Grunwaldzka | 34     | Warszawa |
| 18 | Paweł     | Burzych   | 78032309123 | Grunwaldzka | 34     | Warszawa |
| 19 | Marek     | Maklo...  | 54013112345 | Grunwaldzka | 34     | Warszawa |



JOIN (lub CROSS JOIN) może być stosowany w różnych wersjach

Wynikiem złączenia naturalnego jest zbiór wierszy łączonych tabel; dla tych wierszy wartości kolumn określonych jako warunek złączenia są takie same. Ponieważ w relacyjnych bazach danych informacje są podzielone pomiędzy tabele zawierające dane o obiektach jednego typu, złączenie naturalne jest najczęściej wykorzystywanym (i domyślnym) złączeniem obiektów.

W przykładowej bazie danych informacje o pracownikach i ich miejscach pracy przechowywane są w powiązanych ze sobą tabelach. Dlatego aby wyświetlić pracowników i ich miejsca pracy musimy skorzystać z obu tabel. Przy czym z reguły nie chodzi nam o uzyskanie poniższego wyniku - który wyświetla wszystkie możliwe kombinacje danych.





**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Złączenie INNER JOIN

```
SELECT imie, nazwisko, pesel, ulica, numer, miasto
FROM pracownicy INNER JOIN miejsca
ON pracownicy.miejsce = miejsca.nr_miejsca
```

- **INNER JOIN ... ON ...**

|   | imie      | nazwisko    | pesel       | ulica       | numer | miasto   |
|---|-----------|-------------|-------------|-------------|-------|----------|
| 1 | Jan       | Kowalski    | 82091104357 | Mała        | 13    | Poznań   |
| 2 | Karol     | Nowak       | 80010123987 | Mała        | 13    | Poznań   |
| 3 | Marzena   | Przepiórka  | 89121203456 | Grunwaldzka | 34    | Warszawa |
| 4 | Paweł     | Burzych     | 78032309123 | Wąska       | 4/12A | Kraków   |
| 5 | Marek     | Makłowicz   | 54013112345 | Wąska       | 4/12A | Kraków   |
| 6 | Magdalena | Naramowicka | 77121312098 | Mała        | 13    | Poznań   |
| 7 | Jacek     | Witos       | 69100967234 | Grunwaldzka | 34    | Warszawa |

Klauzula ON jest sposobem na podanie warunku złączenia, czyli wskazania wspólnych kolumn łączonych tabel.



Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Poprawnie napisana instrukcja powinna zwrócić nam tylko miejsca do których przypisany jest konkretny pracownik. Żeby to osiągnąć, musimy określić warunek złączenia, czyli poinformować serwer baz danych, co łączy zapisane w obu tabelach dane. W tym przypadku jest to identyfikator miejsca pracy — zwróć uwagę, że kolumna *miejsce* występuje w obu tabelach, czyli na podstawie tego identyfikatora jesteśmy w stanie sensownie połączyć informacje o pracownikach i miejscach.

INNER JOIN ... ON .. łączy tabele zgodnie ze zdefiniowanym warunkiem.

JOIN (z ang. łączyć) – występuje zawsze w części po FROM w zapytaniu SQL. Tłumacząc powyższą kwerendę (zapytanie):

Wyświetl imię, nazwisko, pesel, ulica, numer, miasto z tabeli pracownicy i połącz z tabelą miejsca, gdzie miejsce w tabeli pracownicy jest takie samo jak nr\_miejsca w tabeli miejsca.



## Złączenia zewnętrzne

- **INNER JOIN** pomija wyniki, które nie mają odpowiedników obu tabelach

```
SELECT imie, nazwisko, pesel, ulica, numer, miasto |
FROM pracownicy INNER JOIN miejsca
ON pracownicy.nr_miejsca = miejsca.nr_miejsca
```

|   | imie      | nazwisko    | pesel       | ulica       | numer | miasto   |
|---|-----------|-------------|-------------|-------------|-------|----------|
| 1 | Jan       | Kowalski    | 82091104357 | Mała        | 13    | Poznań   |
| 2 | Karol     | Nowak       | 80010123987 | Mała        | 13    | Poznań   |
| 3 | Paweł     | Burzych     | 78032309123 | Wąska       | 4/12A | Kraków   |
| 4 | Marek     | Makłowicz   | 54013112345 | Wąska       | 4/12A | Kraków   |
| 5 | Magdalena | Naramowicka | 77121312098 | Mała        | 13    | Poznań   |
| 6 | Jacek     | Witos       | 69100967234 | Grunwaldzka | 34    | Warszawa |

Złączenie naturalne eliminuje z wyniku niepasujące (niespełniające warunku złączenia) wiersze. To dobrze, bo w innym przypadku otrzymalibyśmy zawierający mnóstwo powtórzeń i niepotrzebnych danych iloczyn kartezjański. Ale z drugiej strony ten sam warunek złączenia usunął z wyniku rekordy niemające odpowiedników w łączonej tabeli. Czyli wynik instrukcji wcale nie musi zawierać danych wszystkich naszych pracowników.



Slajd 10

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Złączenia zewnętrzne

- **OUTER JOIN nie pomija wyników**

**OUTER JOIN:**

- LEFT OUTER JOIN – nie pomija wyników z lewej tabeli
- RIGHT OUTER JOIN – nie pomija wyników z prawej tabeli
- OUTER JOIN – nie pomija wyników

```
SELECT imie, nazwisko, pesel, ulica, numer, miasto
FROM pracownicy LEFT OUTER JOIN miejsca
ON pracownicy.nr_miejsca = miejsca.nr_miejsca
```

|   | imie        | nazwisko    | pesel       | ulica       | numer | miasto   |
|---|-------------|-------------|-------------|-------------|-------|----------|
| 1 | Jan         | Kowalski    | 82091104357 | Mala        | 13    | Poznań   |
| 2 | Karol       | Nowak       | 80010123987 | Mala        | 13    | Poznań   |
| 3 | Marzena     | Przepiórka  | 89121203456 | NULL        | NULL  | NULL     |
| 4 | Pawel       | Burzych     | 78032309123 | Waska       | 4/12A | Kraków   |
| 5 | Marek       | Makłowicz   | 54013112345 | Waska       | 4/12A | Kraków   |
| 6 | Magdalena   | Naramowicka | 77121312098 | Mala        | 13    | Poznań   |
| 7 | Jacek       | Witos       | 69100967234 | Grunwaldzka | 34    | Warszawa |
| 8 | Maksymilian | Markowski   | 67121209878 | NULL        | NULL  | NULL     |

Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Czasami chcielibyśmy uzyskać komplet danych z jednej tabeli, nawet jeżeli nie są one powiązane z danymi w innych tabelach. Umożliwia nam to złączenie zewnętrzne. Wynikiem lewo- lub prawostronnego złączenia zewnętrznego jest zbiór wierszy łączonych tabel, dla których wartości kolumn określonych jako warunek złączenia są takie same; zbiór ten uzupełniony jest pozostałymi wierszami z lewej lub prawej łączonej tabeli. Nieistniejące wartości reprezentowane są w wyniku złączenia przez wartość NULL.

OUTER JOIN:

LEFT OUTER JOIN – nie pomija wyników z lewej tabeli

RIGHT OUTER JOIN – nie pomija wyników z prawej tabeli

OUTER JOIN – nie pomija wyników z żadnej tabeli

Slajd 11

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language


## Złączenia zewnętrzne

Złączenia zewnętrzne stosowane są do wyświetlania kompletnych informacji o wszystkich obiektach danego typu, nawet jeżeli nie istnieją powiązane z nimi obiekty innego typu.

Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Podsumowanie OUTER JOIN

Slajd 12



**SQL**  
Structured Query Language

### Złączenie tabeli z nią samą

- Złączenie tabeli z nią samą jest jedną z technik języka SQL, odpowiadającą użyciu zmiennych w proceduralnych językach programowania.


**Ważne zasady:**

- Trzeba utworzyć różne aliasy dla łączonej tabeli i w ramach zapytania konsekwentnie odwoływać się do aliasów, a nie do nazwy tabeli.
- Każdy rekord, w którym wartości atrybutu złączenia będą sobie równe, zostanie dodany do wyniku złączenia, co spowoduje powstanie duplikatów rekordów.

**DAILY GROUP** Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Złączenie tabeli z nią samą wykonywane jest w taki sam sposób, jak omawiane do tej pory złączenia różnych tabel. Chociaż serwery bazodanowe nie tworzą kopii złączonej tabeli, to wszystkie operacje przeprowadzane są tak, jakby dotyczyły dwóch identycznych tabel. Złączenie tabeli z nią samą stosujemy, kiedy chcemy wybrać rekordy z tabeli na podstawie wspólnych wartości atrybutów rekordów tej samej tabeli.

Slajd 13



**SQL**  
Structured Query Language

### Złączenie tabeli z nią samą

```
SELECT p.imie, p.nazwisko, p.pesel, m.imie, m.nazwisko
FROM pracownicy p INNER JOIN pracownicy m
ON p.przełożony = m.nr_pracownika
```


|   | imie        | nazwisko    | pesel       | imie  | nazwisko |
|---|-------------|-------------|-------------|-------|----------|
| 1 | Jan         | Kowalski    | 82091104357 | Jacek | Witos    |
| 2 | Karol       | Nowak       | 80010123987 | Jan   | Kowalski |
| 3 | Marzena     | Przepiórka  | 89121203456 | Jan   | Kowalski |
| 4 | Paweł       | Burzych     | 78032309123 | Jacek | Witos    |
| 5 | Marek       | Makłowicz   | 54013112345 | Jacek | Witos    |
| 6 | Magdalena   | Naramowicka | 77121312098 | Jan   | Kowalski |
| 7 | Jacek       | Witos       | 69100967234 | Jacek | Witos    |
| 8 | Maksymilian | Markowski   | 67121209878 | Jacek | Witos    |

**DAILY GROUP** Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych


Złączenia tabeli z samą sobą są często wykorzystywane do rekurencyjnego odczytania danych, na przykład informacji o podwładnych (podwładny osoby X może być przełożonym osoby Y, która z kolei może być przełożonym osoby Z). W testowej bazie danych nie ma zapisanych takich zależności. Przykład jest czysto szkoleniowy.



Slajd 14



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Unie


- UNION
  - Umożliwia połączenie dwóch tabel
  - Nazwy kolumn muszą być takie same (można użyć alias)

**Klienci**

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Mikołaj  | Wiśniewski | 82040112389 |
| 2          | Marcin   | Kaniewski  | 73122990123 |

**Pracownicy**

| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia | pesel       |
|---------------|-------------|-------------|------------------|--------|-------------|
| 1             | Kowalski    | Jan         | 2300.00          | 250.00 | 82091104357 |
| 2             | Nowak       | Karol       | 2700.00          | 100.00 | 80010123987 |
| 3             | Przepiórka  | Marzena     | 2700.00          | NULL   | 89121203456 |
| 4             | Burzych     | Paweł       | 1900.00          | 500.00 | 78032309123 |
| 5             | Makłowicz   | Marek       | 2000.00          | NULL   | 54013112345 |
| 6             | Naramowicka | Magdalena   | 2100.00          | 200.00 | 77121312098 |
| 7             | Witos       | Jacek       | 3000.00          | 500.00 | 69100967234 |
| 8             | Markowski   | Maksymilian | 2000.00          | NULL   | 67121209878 |




Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych


Unie umożliwiają łączenie tabel. Jeśli chcemy wyświetlić wszystkie osoby bez względu czy są to klienci czy pracownicy – możemy użyć właśnie polecenia UNION (warunkiem są takie same nazwy kolumn).

Ciekawostka: działania na zbiorach sortują wyniki i usuwają duplikaty. Wyjątkiem jest UNION ALL

Slajd 15



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Unie

```
SELECT nazwisko ,imie, pesel FROM pracownicy
UNION
SELECT nazwisko ,imie, pesel FROM klienci;
```

|    | nazwisko    | imie        | pesel       |
|----|-------------|-------------|-------------|
| 1  | Burzych     | Paweł       | 78032309123 |
| 2  | Kowalski    | Jan         | 82091104357 |
| 3  | Makłowicz   | Marek       | 54013112345 |
| 4  | Marcin      | Kaniewski   | 73122990123 |
| 5  | Markowski   | Maksymilian | 67121209878 |
| 6  | Mikołaj     | Wiśniewski  | 82040112389 |
| 7  | Naramowicka | Magdalena   | 77121312098 |
| 8  | Nowak       | Karol       | 80010123987 |
| 9  | Przepiórka  | Marzena     | 89121203456 |
| 10 | Witos       | Jacek       | 69100967234 |



Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Powyższy przykład pokazuje jak w SQL połączyć dwie tabele razem. Polecenie UNION eliminuje duplikaty, a co za tym idzie sortuje wynik połączenia obu tabel. Rezultat będzie posortowany po nazwisku, a w dalszej kolejności po imieniu i numerze pesel

Slajd 16

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

### Wyłączenie

- EXCEPT – wyświetlenie tylko kolumn z jednej tabeli, które nie znajdują się w drugiej
- Przykład, pokaż wszystkich pracowników, którzy nie są klientami:

```
SELECT nazwisko , imie, pesel FROM pracownicy
EXCEPT
SELECT nazwisko , imie, pesel FROM klienci;
```

DAILY GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Jeśli chcemy wyszukać wartości tabeli A, które nie znajdują się w tabeli B – korzystamy z polecenia EXCEPT. Konstrukcja jest identyczna jak przy UNION.

Slajd 17

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

### Funkcje agregujące

- COUNT - czyli zliczanie ilości zwróconych wierszy z zapytania,
- SUM - wynik z dodawania wartości, ze wszystkich elementów występujących w zapytaniu (każda kolumna osobno),
- AVG - wartość średnia (arytmetyczna), ze wszystkich wartości występujących w zapytaniu (również każda kolumna osobno),
- MIN - wartość najmniejsza ze wszystkich występujących w kolumnie,
- MAX - wartość największa ze wszystkich dostępnych w kolumnie.

DAILY GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Wprowadzenie do funkcji agregujących. Każda z nich pochodzi z j. angielskiego – w ten sposób można łatwo zapamiętać.

Z wyjątkiem funkcji COUNT(\*), wszystkie funkcje agregujące wykonują krok eliminacji NULL tak, że wartości NULL nie są uwzględniane w ostatecznym wyniku obliczeń.

Slajd 18

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

### Grupowanie GROUP BY

- GROUP BY:
  - Umożliwia grupowanie wyników względem warunku, np.
    - Pokaż ilość pracowników w podziale na miejsca pracy

```
SELECT COUNT(*), nr_miejsca FROM pracownicy GROUP BY nr_miejsca;
```

DAILY GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

SQL jest językiem zapytań zorientowanym na przetwarzanie zbiorów, a nie pojedynczych rekordów, dlatego jego integralnym składnikiem są polecenia grupujące dane. Właśnie te polecenia, wraz z funkcjami sumującymi, stosowane są w praktyce do uzyskiwania odpowiedzi na złożone zapytania, dotyczące zawartości bazy danych.

Przykład ilustruje użycie klauzuli GROUP BY.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 19

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## COUNT

- Zliczanie liczby rekordów
- Przykład liczba pracowników w poszczególnych miastach

```
SELECT miasto, COUNT(imie)
FROM pracownicy INNER JOIN miejsca
ON pracownicy.nr_miejscas = miejsca.nr_miejscas GROUP BY miasto;
```

|   | miasto   | (No column name) |
|---|----------|------------------|
| 1 | Kraków   | 2                |
| 2 | Poznań   | 3                |
| 3 | Warszawa | 1                |

**DAILY** Lektja 3 - Złożone zapytania, grupowanie, agregacja danych

Funkcja COUNT liczy wystąpienia bazy danych w tym przypadku zliczamy dla każdego z miast – ilu pracowników jest zatrudnionych w konkretnej lokalizacji. W klauzuli *group by* wskazujemy po czym mają być grupowane dane.

Slajd 20

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## SUM

- SUM – sumuje wartości liczbowe, np. suma pensji w poszczególnych miastach

```
SELECT miasto, SUM(placa_zasadnicza)
FROM pracownicy INNER JOIN miejsca
ON pracownicy.nr_miejscas = miejsca.nr_miejscas GROUP BY miasto;
```

|   | miasto   | (No column name) |
|---|----------|------------------|
| 1 | Kraków   | 3900.00          |
| 2 | Poznań   | 7100.00          |
| 3 | Warszawa | 3000.00          |

**DAILY** Lektja 3 - Złożone zapytania, grupowanie, agregacja danych

Przykład wykorzystania funkcji SUM

Slajd 21

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## AVG

- AVG – wylicza średnią
- Średnia łączna płaca w różnych miastach:

```
SELECT miasto, AVG(placa_zasadnicza + ISNULL(premia,0))
FROM pracownicy INNER JOIN miejsca
ON pracownicy.nr_miejscas = miejsca.nr_miejscas GROUP BY miasto;
```

|   | miasto   | (No column name) |
|---|----------|------------------|
| 1 | Kraków   | 2200.000000      |
| 2 | Poznań   | 2550.000000      |
| 3 | Warszawa | 3500.000000      |

**DAILY** Lektja 3 - Złożone zapytania, grupowanie, agregacja danych



**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## MIN i MAX

- MIN – to wartość minimalna, a MAX wartość maksymalna

```
SELECT miasto, MIN(placa_zasadnicza + ISNULL(premia,0)) as Minimum,
MAX(placa_zasadnicza + ISNULL(premia,0)) as Maksimum
FROM pracownicy INNER JOIN miejsca
ON pracownicy.nr_miejsca = miejsca.nr_miejsca GROUP BY miasto;
```

|   | miasto   | Minimum | Maksimum |
|---|----------|---------|----------|
| 1 | Kraków   | 2000.00 | 2400.00  |
| 2 | Poznań   | 2300.00 | 2800.00  |
| 3 | Warszawa | 3500.00 | 3500.00  |

**DAILY** GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Grupowanie

- HAVING – umożliwia stosowanie warunku na podstawie funkcji agregującej
- Przykład – wyświetl miasta z pensją średnią większą niż 2500

```
SELECT miasto, AVG(placa_zasadnicza + ISNULL(premia,0)) as Średnia
FROM pracownicy INNER JOIN miejsca
ON pracownicy.nr_miejsca = miejsca.nr_miejsca GROUP BY miasto
having AVG(placa_zasadnicza + ISNULL(premia,0)) > 2500;
```

|   | miasto   | Średnia     |
|---|----------|-------------|
| 1 | Poznań   | 2550.000000 |
| 2 | Warszawa | 3500.000000 |

**DAILY** GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Klauzula HAVING służy do ograniczania zbioru wierszy, zwracanych w wyniku wykonania klauzuli GROUP BY. Między HAVING a GROUP BY występuje zależność podobna do tej, która wiąże klauzulę WHERE z poleceniem SELECT. W przeciwieństwie do klauzuli WHERE, która operuje na wierszach tabel podawanych w zapytaniu, HAVING działa na zbiorze wierszy wynikowych

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Podzapytania

- Mniej wydajne niż JOIN czy UNION
- Używane gdy nie ma innej możliwości
- SELECT w SELECT:

```
SELECT nazwisko ,imie, pesel FROM pracownicy
WHERE nr_miejsca = (SELECT nr_miejsca FROM miejsca WHERE miasto='Kraków');
```

|   | nr_miejsca | nr_miejsca | nr_miejsca  |
|---|------------|------------|-------------|
| 1 | Burzych    | Paweł      | 78032309123 |
| 2 | Makłowicz  | Marek      | 54013112345 |

**DAILY** GROUP Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Podzapytanie jest zapytaniem umieszczonym wewnątrz innego zapytania, tzw. zapytania zewnętrznego. Podzapytania najczęściej umieszcza się w warunkach w klauzulach WHERE i HAVING zapytania zewnętrznego, niektóre SZBD dopuszczają również stosowanie podzapytań w klauzulach SELECT i FROM.

Podzapytania stosuje się w ostateczności, gdy nie można zastosować połączenia typu UNION lub JOIN.



Slajd 25

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Podzapytania skorelowane

- W zapytaniach zawierających podzapytania skorelowane, zapytanie wewnętrzne jest realizowane oddzielnie dla każdego wiersza z zapytania zewnętrznego.

```
SELECT nazwisko, imie, pesel,
(
SELECT miejsce M FROM miejsca
WHERE M.nr_miejsca = P.nr_miejsca -- faktyczna korelacja
) miejsce
FROM pracownicy P
```

**DAILY GROUP** Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

W zapytaniach zawierających podzapytania skorelowane, zapytanie wewnętrzne jest realizowane oddzielnie dla każdego wiersza z zapytania zewnętrznego.

Slajd 26

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Ćwiczenia

### Pracownicy

| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia | pesel       | stanowisko          | nr_miejsca | przełożony |
|---------------|-------------|-------------|------------------|--------|-------------|---------------------|------------|------------|
| 1             | Kowalski    | Jan         | 2300,00          | 250,00 | 82091104357 | Manager             | 1          | 7          |
| 2             | Nowak       | Karol       | 2700,00          | 100,00 | 80010123987 | Specjalista         | 1          | 1          |
| 3             | Przepiórka  | Marzena     | 2700,00          | NULL   | 89121203456 | Specjalista         | NULL       | 1          |
| 4             | Burzych     | Paweł       | 1900,00          | 500,00 | 78032309123 | Specjalista         | 2          | 7          |
| 5             | Makłowicz   | Marek       | 2000,00          | NULL   | 54013112345 | Specjalista         | 2          | 7          |
| 6             | Naramowicka | Magdalena   | 2100,00          | 200,00 | 77121312098 | Specjalista         | 1          | 1          |
| 7             | Witos       | Jacek       | 3000,00          | 500,00 | 69100967234 | Manager             | 3          | 7          |
| 8             | Markowski   | Maksymilian | 2000,00          | NULL   | 67121209878 | Młodszy specjalista | NULL       | 7          |

### Klienci

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Mikołaj  | Wiśniewski | 82040112389 |
| 2          | Marcin   | Kaniewski  | 73122990123 |

### Miejsca

| nr_miejsca | ulica       | numer | miasto   | kod    | telefon        |
|------------|-------------|-------|----------|--------|----------------|
| 1          | Mała        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2          | Wąska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3          | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |

**DAILY GROUP** Lekcja 2 - Języki programowania SQL, środowisko, wprowadzenie do języka SQL

### Ćwiczenie 1

Pokaż w jakich miejscowościach pracują pracownicy

### Ćwiczenie 2

Pokaż pracowników z miejscowości Warszawa

### Ćwiczenie 3

Pokaż wszystkie osoby (klienci i pracownicy)

### Ćwiczenie 4

Pokaż pracowników i ich przełożonych

### Ćwiczenie 5

Pokaż najniższą pensję całej firmie

### Ćwiczenie 6

Pokaż średnią pensję w podziale na miejscowości

### Ćwiczenie 7

Pokaż miejscowość, gdzie pracuje najwięcej pracowników

### Ćwiczenie 8

Pokaż ilość pracowników w każdej miejscowości

### Ćwiczenie 9

Pokaż średnią pensję specjalistów i managerów

### Ćwiczenie 10

Pokaż najwyższą pensję średnią premie w podziale na stanowiska

Slajd 27



Slide 27 content includes logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and DAILY GROUP. The title is 'SQL Structured Query Language'. The exercise title is 'Ćwiczenia' and the task is: 'Napisz zapytanie zwracające 3 uczniów z tabeli TBL\_UCZEN, których wartość kolumny IMIE to Anna'. The footer text is 'Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych'.

```
SELECT TOP 3 * FROM
TBL_UCZEN WHERE
IMIE=„Anna”
```

Slajd 28



Slide 28 content includes logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and DAILY GROUP. The title is 'SQL Structured Query Language'. The exercise title is 'Ćwiczenia' and the task is: 'Napisz zapytanie zwracające z tabeli TBL\_UCZEN średnią ocen (kolumna OCENA) wszystkich uczniów klasy 3A (kolumna KLASA)'. The footer text is 'Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych'.

```
SELECT AVG(OCENA) FROM
TBL_UCZEN WHERE KLASA=„3A”
```



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 29

  **SQL**  
Structured Query Language



### Ćwiczenia

- Napisz zapytanie zwracające z tabeli TBL\_UCZEN ilość uczniów klasy 3A (kolumna KLASA)

 Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych


```
SELECT COUNT(*) FROM
TBL_UCZEN WHERE KLASA='3A'
```

Slajd 30

  **SQL**  
Structured Query Language

### Ćwiczenia

- Napisz zapytanie zwracające średnią ocen (kolumna OCENA) uczniów każdej klasy (kolumna KLASA) z tabeli TBL\_UCZEN

 Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

```
SELECT AVG(OCENA) FROM
TBL_UCZEN GROUP BY
KLASA='3A'
```





**SQL**  
*Structured Query Language*

## Podsumowanie

|                       |                                                                  |
|-----------------------|------------------------------------------------------------------|
| JOIN                  | Łączy dane z tabel                                               |
| INNER JOIN ... ON ... | Pokazuje tylko wyniki, które są łączy wspólna wartość            |
| LEFT/RIGHT OUTER JOIN | Nie wyklucza wartości                                            |
| GROUPING              | Grupuje wyniki po zadanym warunku                                |
| HAVING                | Pozwala na użycie funkcji agregujących w warunku WHERE           |
| UNION                 | łączy tabele                                                     |
| EXCEPT                | Pokazuje wartości z jednej tabeli, które nie występują w drugiej |
| MIN i MAX             | Pozwala wyświetlić wartość maksymalną i minimalną                |
| COUNT                 | Zlicza wiersze                                                   |
| SUM                   | Sumuje wartości                                                  |
| AVG                   | Liczy wartość średnią                                            |




Człowiek - najlepsza inwestycja

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


Lekcja 3 - Złożone zapytania, grupowanie, agregacja danych

Tabela podsumowująca poznane wyrażenia

### Ćwiczenia

Ćwiczenia zostały przedstawione na slajdach 26-30. Uczniowie w ramach ćwiczenia będą musieli napisać zapytania SQL na bazie tabel przygotowanych przez szkoleniowca.

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili budować złożone zapytania SQL, łączyć wiele tabel ze sobą na różne sposoby, grupować dane w tabelach, wyliczać średnie, wartości minimalne i maksymalne, liczyć wiersze w tabeli. Uczniowie powinni być w stanie budować zapytania złożone z podzapytań.

## Lekcja 4 – DML

### Cel lekcji

Celem lekcji jest wyjaśnienie języka DML, szczegółowe zapoznanie z podstawowymi poleceniami tego języka: INSERT, UPDATE oraz DELETE.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Treść - slajdy z opisem



Slajd 1

DML (Data Manipulation Language) służy do wykonywania operacji na danych – do ich umieszczania w bazie, kasowania, przeglądania oraz dokonywania zmian. Najważniejsze polecenia z tego zbioru to: INSERT, UPDATE oraz DELETE

Slajd 2

Slajd 3

Część osób do DML zalicza także zapytania typu SELECT, jednak przeglądanie danych nie do końca można zaliczyć jako manipulacja danymi – w tej kwestii informatycy nie są zgodni.

SQL  
Structured Query Language



## INSERT

- Polecenie INSERT służy do wstawiania danych do bazy danych
- Ogólna składnia INSERT to:  
INSERT INTO tabela VALUES (wartość\_kolumny1, wartość\_kolumny2, ...)



Lekcja 4 - DML (Data Manipulation Language)

Do wstawiania wierszy do tabeli służy polecenie INSERT. Najprostszą wersję tego polecenia przedstawiono na przykładzie. Polecenie zaczyna się od słów kluczowych INSERT INTO po którym podaje się nazwę tabeli, słowo kluczowe VALUES i w nawiasie listę wartości oddzielonych przecinkami. Wartości na tej liście odpowiadają kolejnym atrybutom relacji (w kolejności w jakiej atrybuty te zostały zdefiniowane). Wpisami na liście wartości mogą być: słowa kluczowe NULL i DEFAULT oraz konkretne wartości atrybutów i ewentualnie podzapytania zwracające jedną wartość. Jeżeli zamiast konkretnej wartości poda się NULL, wówczas nowy wiersz będzie miał pustą wartość atrybutu odpowiadającego pozycji na której wpisano NULL. Podanie DEFAULT spowoduje, że w odpowiedniej kolumnie zostanie zapisana jego wartość domyślna.


SQL  
Structured Query Language

## INSERT - przykład

```
INSERT INTO klienci VALUES (1, 'Wiśniewski', 'Mikołaj', '82040112389');
INSERT INTO klienci VALUES (2, 'Kaniewski', 'Marcin', '73122990123');
```

```
SELECT * FROM klienci
```

| nr_klienta | nazwisko   | imie    | pesel       |
|------------|------------|---------|-------------|
| 1          | Wiśniewski | Mikołaj | 82040112389 |
| 2          | Kaniewski  | Marcin  | 73122990123 |



Lekcja 4 - DML (Data Manipulation Language)

Przykład umieszczania danych w bazie. Za pomocą dwóch wierszy polecenia INSERT wstawiamy do bazy danych numer klienta, nazwisko, imię oraz PESEL klienta. Poniżej sprawdzamy zawartość bazy – upewniamy się że wartości zostały odpowiednio umieszczone w bazie danych.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6

**BZE** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

### INSERT – wartości puste

- INSERT umożliwia wstawienie wiersza bez uzupełniania wszystkich wartości
- Sposób 1:
 

```
INSERT INTO klienci VALUES (3, 'Galas', 'Jan', NULL);
```
- Sposób 2:
 

```
INSERT INTO klienci(nr_klienta, nazwisko, imie) VALUES (4, 'Zimna', 'Maria');
```

DAILY GROUP | Lekcja 4 - DML (Data Manipulation Language)

Umieszczanie pustych wartości w wierszach można zrealizować w dwojaki sposób: albo używany wartości pustej NULL w miejscach gdzie nie znamy wartości, albo wymieniamy po nazwie tabeli wartości które zamierzamy uzupełnić. Oba sposoby dają identyczny efekt

Slajd 7

**BZE** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

### INSERT – wartość domyślna

- Do wstawienia wartości domyślnej służy DEFAULT
- Przykład:
 

```
INSERT INTO klienci VALUES (5, 'Wiśniewska', 'Joanna', DEFAULT)
```

| nr_klienta | nazwisko   | imie    | pesel       |
|------------|------------|---------|-------------|
| 1          | Wiśniewski | Mikołaj | 82040112389 |
| 2          | Kaniewski  | Marcin  | 73122990123 |
| 3          | Galas      | Jan     | NULL        |
| 4          | Zimna      | Maria   | NULL        |
| 5          | Wiśniewska | Joanna  | NULL        |

DAILY GROUP | Lekcja 4 - DML (Data Manipulation Language)

Kolumny tabeli posiadają często wartość domyślną. Użycie polecenia DEFAULT w poleceniu SQL umożliwia wstawienie wartości domyślnej. W przykładzie użyto DEFAULT dla kolumny pesel, której wartością domyślną jest wartość pusta - NULL.

Slajd 8

**BZE** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

### INSERT z SELECT

- Wstawianie wielu wierszy z innej tabeli
- INSERT INTO tabela SELECT ... FROM tabela 2

```
INSERT INTO klienci(nazwisko, imie, pesel)
SELECT imie, nazwisko, pesel FROM pracownicy;
```

DAILY GROUP | Lekcja 4 - DML (Data Manipulation Language)

Wstawianie wielu wierszy pojedynczo jest bardzo czasochłonne. W języku SQL możemy wykorzystać zawartość innych tabel do uzupełnienia tabeli. W przykładzie użyto danych z tabeli pracownicy w celu wstawienia danych do tabeli klienci.



**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## UPDATE

- UPDATE służy do aktualizacji istniejących w tabeli wierszy
- Składnia polecenia:  
UPDATE nazwa\_tabeli SET kolumna = nowa\_wartość WHERE kolumna2=wartość

**DAILY** GROUP

Lekcja 4 - DML (Data Manipulation Language)

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## UPDATE – przykład 1

- Przykład:  
Zaktualizuj premie managerów i ustaw na 500zł

```
UPDATE pracownicy SET premia = 500 WHERE stanowisko = 'Manager'
```

| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia | pesel       | stanowisko          | nr_miejsca | przełożony |
|---------------|-------------|-------------|------------------|--------|-------------|---------------------|------------|------------|
| 1             | Kowalski    | Jan         | 2300.00          | 500.00 | 82091104357 | Manager             | 1          | 7          |
| 2             | Nowak       | Karol       | 2700.00          | 100.00 | 80010123987 | Specjalista         | 1          | 1          |
| 3             | Przepiórka  | Marzena     | 2700.00          | NULL   | 89121203456 | Specjalista         | NULL       | 1          |
| 4             | Burzych     | Paweł       | 1900.00          | 500.00 | 78032309123 | Specjalista         | 2          | 7          |
| 5             | Makłowicz   | Marek       | 2000.00          | NULL   | 54013112345 | Specjalista         | 2          | 7          |
| 6             | Naramowicka | Magdalena   | 2100.00          | 200.00 | 77121312098 | Specjalista         | 1          | 1          |
| 7             | Witos       | Jacek       | 3000.00          | 500.00 | 69100967234 | Manager             | 3          | 7          |
| 8             | Markowski   | Maksymilian | 2000.00          | NULL   | 67121209878 | Młodszy specjalista | NULL       | 7          |

**DAILY** GROUP

Lekcja 4 - DML (Data Manipulation Language)

Polecenie zmienia premię na 500 dla wszystkich rekordów, gdzie stanowisko to Manager



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 11

**B2E** BUSINESS TO EDUCATION **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** **SQL** Structured Query Language

### UPDATE – przykład 2

- Przykład:  
Powiększ wynagrodzenie Kowalskiego i Nowaka o 20%

```
UPDATE pracownicy SET placa_zasadnicza = placa_zasadnicza*1.2
WHERE nazwisko = 'Kowalski' OR nazwisko = 'Nowak'

UPDATE pracownicy SET placa_zasadnicza = placa_zasadnicza*1.2
WHERE nazwisko IN ('Kowalski', 'Nowak')
```

**DAILY** GROUP Lekcja 4 - DML (Data Manipulation Language)

Oba polecenia wykonują taką samą operację, tj. powiększają kwotę płacy zasadniczej o 20% (mnożą płacę przez 1,2) w rekordach gdzie nazwisko to Nowak lub Kowalsk

Slajd 12

**B2E** BUSINESS TO EDUCATION **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** **SQL** Structured Query Language

### UPDATE – przykład 3

- Przykład:  
Zmień nazwisko pracownika z Naramowicka na Naramowicka-Przybylska

```
UPDATE pracownicy SET nazwisko = 'Naramowicka-Przybylska'
WHERE nazwisko = 'Naramowicka'
```

**DAILY** GROUP Lekcja 4 - DML (Data Manipulation Language)

Przykład aktualizuje pole nazwisko gdzie nazwisko to Naramowicka

Slajd 13

**B2E** BUSINESS TO EDUCATION **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** **SQL** Structured Query Language

### UPDATE - ostrzeżenie

**UWAGA!!!!**

- Proszę pamiętać, aby w zapytaniu UPDATE definiować odpowiednie wartości dla WHERE.
- W przypadku braku zdefiniowanego warunku WHERE w tabeli zostaną zmienione **WSZYSTKIE** wiersze!

**DAILY** GROUP Lekcja 4 - DML (Data Manipulation Language)

Bardzo ważna uwaga: wiele początkujących osób zapomina dodać klauzulę WHERE do polecenia UPDATE, skutkuje to zastąpieniem wartości we wszystkich wierszach

Slajd 14

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language

### UPDATE - ostrzeżenie

```
UPDATE klienci SET nazwisko = 'Kowalski';
```

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Kowalski | Mikołaj    | 82040112389 |
| 2          | Kowalski | Marcin     | 73122990123 |
| 3          | Kowalski | Jan        | NULL        |
| 4          | Kowalski | Maia       | NULL        |
| 5          | Kowalski | Joanna     | NULL        |
| 6          | Kowalski | Kowalski   | 82091104357 |
| 7          | Kowalski | Nowak      | 80010123987 |
| 8          | Kowalski | Przepiórka | 89121203456 |
| 9          | Kowalski | Burzych    | 78032309123 |
| 10         | Kowalski | Makłowi... | 54013112345 |
| 11         | Kowalski | Naramo...  | 77121312098 |
| 12         | Kowalski | Witos      | 69100967234 |

**DAILY** Lekcja 4 - DML (Data Manipulation Language)

Bardzo ważna uwaga: wiele początkujących osób zapomina dodać klauzulę WHERE do polecenia UPDATE, skutkuje to zastąpieniem wartości we wszystkich wierszach

Slajd 15

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language

### DELETE

- DELETE używane jest do usuwania istniejących wierszy z tabeli.
- Składnia:  
`DELETE FROM nazwa_tabeli WHERE nazwa_kolumny=wartosc;`

**DAILY** Lekcja 4 - DML (Data Manipulation Language)

Poznaliśmy sposób umieszczania danych w tabeli, ich modyfikacji – teraz czas na usuwania danych. DELETE to ostatnie z wyrażeń DML. Podobnie jak w przypadku UPDATE, należy zwrócić szczególną uwagę na warunek WHERE, tak aby nie skasować wszystkich danych z tabeli.

Slajd 16

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language

### DELETE - przykład

- Usuamy z bazy pracownika o nazwisku Nowak

```
DELETE FROM pracownicy WHERE nazwisko = 'Nowak';
```

(1 row(s) affected)

**DAILY** Lekcja 4 - DML (Data Manipulation Language)

Przykład usuwania rekordu – poniżej widzimy odpowiedź systemu. Widzimy, że został usunięty jeden wiersz.



Slajd 17

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## DELETE z podzapytaniem

- Usuwamy z bazy wszystkich pracowników z najwyższą pensją

```
DELETE FROM pracownicy WHERE placa_zasadnicza =
(SELECT MAX(placa_zasadnicza) FROM pracownicy)
```

**DAILY GROUP** Lekcja 4 - DML (Data Manipulation Language)

W warunku WHERE w poleceniu DELETE możemy zagnieździć podzapytanie – podobnie jak w SELECT.

Slajd 18

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Ćwiczenia

**Pracownicy**

| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia | pesel       | stanowisko          | nr_miejsca | przelozony |
|---------------|-------------|-------------|------------------|--------|-------------|---------------------|------------|------------|
| 1             | Kowalski    | Jan         | 2300,00          | 250,00 | 82091104357 | Manager             | 1          | 7          |
| 2             | Nowak       | Karol       | 2700,00          | 100,00 | 80010123987 | Specjalista         | 1          | 1          |
| 3             | Przepiórka  | Marzena     | 2700,00          | NULL   | 89121203456 | Specjalista         | NULL       | 1          |
| 4             | Burzych     | Paweł       | 1900,00          | 500,00 | 78032309123 | Specjalista         | 2          | 7          |
| 5             | Makłowicz   | Marek       | 2000,00          | NULL   | 54013112345 | Specjalista         | 2          | 7          |
| 6             | Naramowicka | Magdalena   | 2100,00          | 200,00 | 77121312098 | Specjalista         | 1          | 1          |
| 7             | Witos       | Jacek       | 3000,00          | 500,00 | 69100967234 | Manager             | 3          | 7          |
| 8             | Markowski   | Maksymilian | 2000,00          | NULL   | 67121209878 | Młodszy specjalista | NULL       | 7          |

**Klienci**

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Mikołaj  | Wiśniewski | 82040112389 |
| 2          | Marcin   | Kaniewski  | 73122990123 |

**Miejsca**

| nr_miejsca | ulica       | numer | miasto   | kod    | telefon        |
|------------|-------------|-------|----------|--------|----------------|
| 1          | Mala        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2          | Waska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3          | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |

**DAILY GROUP** Lekcja 4 - DML (Data Manipulation Language)

### Ćwiczenie 1

Wstaw dwie miejscowości do tabeli Miejsca

### Ćwiczenie 2

Dodaj po dwóch pracowników do każdej z wstawionych w ćw. 1 lokalizacji

### Ćwiczenie 3

Podnieś wszystkim premie o 30zł, pamiętaj o osobach, które dotychczas nie miały premii

### Ćwiczenie 4

Zatrudniamy naszych klientów, wstaw do tabeli pracownicy osoby z tabeli klienci

### Ćwiczenie 5

Zmień miejsce zatrudnienia osób pracujących w Warszawie – przenosimy biuro do Szczecina

### Ćwiczenie 6

Rezygnujemy z biura w Warszawie – usuń lokalizację z tabeli miejsca

### Ćwiczenie 7

Zwiększ pensje zasadnicze pracowników ze Szczecina o 30%

### Ćwiczenie 8

Osoby, które dotychczas nie miały przypisanej lokalizacji zostają przypisane do Szczecina – uaktualnij bazę danych

### Ćwiczenie 9

Znaleźliśmy nowych klientów – wprowadź kilka dodatkowych osób

### Ćwiczenie 10

Zaktualizuj nazwisko pani Marzeny – po ślubie zmieniła nazwisko na Nowak

Slajd 19



Slide 19 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main title is 'Ćwiczenia'. A bullet point reads: 'Napisz komendę usuwającą wszystkich uczniów (tabela TBL\_UCZEN) z klasy 4A (kolumna KLASA)'. The footer includes the DAILY GROUP logo and the text 'Lekcja 4 - DML (Data Manipulation Language)'.

```
DELETE FROM TBL_UCZEN
WHERE KLASA = „4A”
```

Slajd 20





Slide 20 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main title is 'Ćwiczenia'. A bullet point reads: 'Rozpoczynamy nowy rok szkolny - napisz komendę zmieniającą w tabelu TBL\_UCZEN nazwę klasy (kolumna KLASA) wszystkich uczniów z klasy 1A na klasę 2A'. The footer includes the DAILY GROUP logo and the text 'Lekcja 4 - DML (Data Manipulation Language)'.

```
UPDATE TBL_UCZEN SET KLASA
= „2A” WHERE KLASA = „1A”
```




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 21

  **SQL**  
Structured Query Language



### Ćwiczenia

- Napisz komendę dodającą Jana Kowalskiego do klasy 1A – tabela TBL\_UCZEN ma kolumny:
  - IMIE
  - NAZWISKO
  - KLASA

 Lekcja 4 - DML (Data Manipulation Language)


```
INSERT INTO TBL_UCZEN(IMIE,
NAZWISKO, KLASA) VALUES
(„Jan”, „Kowalski”, „1A”)
```

Slajd 22

  **SQL**  
Structured Query Language

### Ćwiczenia



- Napisz komendę wstawiającą do tabeli z poprzedniego ćwiczenia uczniów: Maria Nowak, Jakub Wiśniewski bez przypisanej klasy.

 Lekcja 4 - DML (Data Manipulation Language)

```
INSERT INTO TBL_UCZEN(IMIE,
NAZWISKO, KLASA) VALUES
(„Maria”, „Nowak”, NULL)
```


```
INSERT INTO TBL_UCZEN(IMIE,
NAZWISKO, KLASA) VALUES
(„Jakub”, „Wiśniewski”, NULL)
```

Slajd 23

  **SQL**  
Structured Query Language

### Ćwiczenia

- Napisz komendę aktualizującą uczennicę Marię Nowak – przyporządkującą ją do klasy 3A

 Lekcja 4 - DML (Data Manipulation Language)

```
UPDATE TBL_UCZEN SET KLASA
= „1A” WHERE IMIE = „Maria”
AND NAZWISKO=„Nowak”
```

**SQL**  
Structured Query Language

## Podsumowanie

|        |                             |
|--------|-----------------------------|
| INSERT | Wstawia wiersze do tabeli   |
| UPDATE | Aktualizuje wartość wierszy |
| DELETE | Usuwa wiersze z tabeli      |

KAPITAŁ LUDZKI  
INICJATYWA EUROPEJSKA  
FUNDUSZ SPÓŁNOŚCI

CELNIK - najlepsza inwestycja

INIA EUROPEJSKA  
INICJATYWA EUROPEJSKA  
FUNDUSZ SPÓŁNOŚCI

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**DAILY**  
GROUP

Lekcja 4 - DML (Data Manipulation Language)

Tabela podsumowująca poznane wyrażenia

## Ćwiczenia

Ćwiczenie zostało przedstawione na slajdach 19-23. Uczniowie napiszą instrukcje modyfikujące dane w bazie danych.

## Opis założonych osiągnięć ucznia

We wcześniejszych lekcjach uczniowie zapoznali się z poleceniem SELECT, potrafią już odczytywać dane z bazy. W tej lekcji nauczyli się modyfikować dane – wstawiać nowe rekordy, aktualizować i modyfikować istniejące.

## Lekcja 5 - DDL

### Cel lekcji

Celem lekcji jest zapoznanie się z językiem DDL. W tej części kursu uczniowie zapoznają się z mechanizmami tworzącymi tabele w bazie danych. Przed przystąpieniem do definiowania tabel, uczniowie zostaną zapoznani z typami danych w języku SQL.

### Treść - slajdy z opisem

Slajd 1

**SQL**  
Structured Query Language

# SQL

## Structured Query Language

### Lekcja 5

### DDL

### Data Definition Language

KAPITAŁ LUDZKI  
INICJATYWA EUROPEJSKA  
FUNDUSZ SPÓŁNOŚCI

CELNIK - najlepsza inwestycja

INIA EUROPEJSKA  
INICJATYWA EUROPEJSKA  
FUNDUSZ SPÓŁNOŚCI

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**DAILY**  
GROUP

SQL Structured Query Language Lekcja 5

Kolejna dziedzina języka SQL – DDL – Data Definition Language, służy do definiowania obiektów bazy danych. Nauczyliśmy się wyświetlać dane z bazy (SELECT) oraz je modyfikować (DML: INSERT, UPDATE, DELETE). Przyszedł czas na tworzenie, modyfikację i usuwanie baz danych, tabel i innych obiektów.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 2

**Przypomnienie**

- INSERT
- UPDATE
- DELETE

Lekcja 5 - DDL (Data Definition Language)

UPDATE służy do aktualizacji istniejących danych w tabeli.

INSERT używane jest do dodawania nowych wierszy do tabeli. INSERT pozwala na dodawanie nowych wierszy na dwa sposoby. W jednym z nich definiujemy nazwy kolumn, zaś w drugim pomijamy nazwy kolumn wprowadzając same wartości. W przypadku drugiego sposobu, musimy wziąć pod uwagę kolejność kolumn w tabeli.

DELETE używane jest do usuwania istniejących wierszy z tabeli.

Slajd 3

**Data Definition Language**

Dzięki DDL (Data Definition Language) można operować na strukturach, w których dane są przechowywane – czyli np. dodawać, zmieniać i kasować tabele lub bazy. Najważniejsze polecenia tej grupy to:



- **CREATE** (np. CREATE TABLE, CREATE DATABASE, ...) – utworzenie struktury (bazy, tabeli, indeksu itp.),
- **DROP** (np. DROP TABLE, DROP DATABASE, ...) – usunięcie struktury,
- **ALTER** (np. ALTER TABLE ADD COLUMN ...) – zmiana struktury (dodanie kolumny do tabeli, zmiana typu danych w kolumnie tabeli).

Lekcja 5 - DDL (Data Definition Language)

Wyjaśnienie, że pomimo wspólnego standardu istnieje wiele różnych rozszerzeń. Podstawowa składnia wszystkich rozszerzeń języka jest taka sama, jednak istnieje wiele różnic. W tym kursie zajmiemy się językiem SQL w ujęciu rozszerzenia TSQL, promowanym przez Microsoft.




Slajd 4



SQL  
Structured Query Language

## Typy danych w SQL



- Typy numeryczne
- Data i czas
- Typ znakowy
- Typy binarne
- Typy przestrzenne



Lekcja 5 - DDL (Data Definition Language)

Zanim będziemy mogli stworzyć tabelę musimy poznać typy danych (w tym kolumn), które możemy użyć w języku SQL. Większość z przedstawionych typów dotyczy wszystkich rozszerzeń języka SQL. Jednak warto pamiętać, że w tym kursie skupiamy się na języku TSQL

Slajd 5




SQL  
Structured Query Language

## Typy danych w SQL

- Typy numeryczne

| Typ        | Opis                                                                                                                   |
|------------|------------------------------------------------------------------------------------------------------------------------|
| Bigint     | 8-bajtowy typ numeryczny z zakresem: $-2^{63}$ (-9,223,372,036,854,775,808) to $2^{63}-1$ (9,223,372,036,854,775,807). |
| Numeric    | Wielkość typu zależy od precyzji. 1-9 – 5 bajtów, 10-19 – 9 bajtów, 20-28 – 13 bajtów, 29-38 – 17 bajtów.              |
| bit        | Wartości: 0, 1 lub NULL.                                                                                               |
| smallint   | 2-bajtowy typ numeryczny z zakresem: $-2^{15}$ (-32,768) do $2^{15}-1$ (32,767).                                       |
| decimal    | Wielkość typu zależy od precyzji. 1-9 – 5 bajtów, 10-19 – 9 bajtów, 20-28 – 13 bajtów, 29-38 – 17 bajtów.              |
| smallmoney | 4-bajtowy typ do zapisu walut z zakresem: - 214,748.3648 do 214,748.3647.                                              |





Lekcja 5 - DDL (Data Definition Language)

Do wartości numerycznych zalicza się typy przedstawione w tabeli.



Slajd 6






**SQL**  
Structured Query Language

## Typy danych w SQL


- Typy numeryczne

| Typ     | Opis                                                                                                                                                                              |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int     | 4-bajtowy typ numeryczny z zakresem: $-2^{31}$ (-2,147,483,648) do $2^{31}-1$ (2,147,483,647).                                                                                    |
| tinyint | 1-bajtowy typ numeryczny z zakresem: 0 – 255.                                                                                                                                     |
| money   | 8-bajtowy typ do zapisu walut z zakresem: -922,337,203,685,477.5808 do 922,337,203,685,477.5807.                                                                                  |
| float   | Typ zmiennoprzecinkowy z zakresem: - 1.79E+308 to -2.23E-308, 0 and 2.23E-308 to 1.79E+308. Ilość bajtów zależy od precyzji. 1-24 – 4 bajty (7 cyfr), 25-53 – 8 bajtów (15 cyfr). |
| real    | 4-bajtowy typ zmiennoprzecinkowy z zakresem: - 3.40E + 38 to -1.18E - 38, 0 and 1.18E - 38 to 3.40E + 38. Typ real jest 24-cyfrowym typem float.                                  |




Lekcja 5 - DDL (Data Definition Language)

Zanim będziemy mogli stworzyć tabelę musimy poznać typy danych (w tym kolumn), które możemy użyć w języku SQL. Większość z przedstawionych typów dotyczy wszystkich rozszerzeń języka SQL. Jednak warto pamiętać, że w tym kursie skupiamy się na języku TSQL



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


**SQL**  
Structured Query Language

---

## Typy danych w SQL


- **Data i czas**

| Typ            | Opis                                                                                                                                                                                                                    |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| date           | 10-znakowy typ zapisu daty z precyzją: 10.0. Domyślnym formatem jest: YYYY-MM-DD, natomiast domyślna wartość to 1900-01-01                                                                                              |
| datetimeoffset | 8-bajtowy typ do zapisu daty z zakresem: 1 stycznia 1 roku n.e. do 31 grudnia roku 9999, liczone wg kalendarza gregoriańskiego oraz czas 24-godzinny z dokładnością do 100 ns. Uwzględnia przesunięcie strefy czasowej. |
| datetime2      | 8-bajtowy typ do zapisu daty z zakresem: 1 stycznia 1 roku n.e. do 31 grudnia 9999 oraz czas 24-godzinny z dokładnością do 100 ns.                                                                                      |
| smalldatetime  | 4-bajtowy typ do zapisu daty z zakresem 1900-01-01 do 2076-06-06 z dokładnością do 1 minuty.                                                                                                                            |
| datetime       | 8-bajtowy typ do zapisu daty z zakresem 1 stycznia 1753 do 31 grudnia 9999 z dokładnością do 0.00333 sekund (zaokrągła milisekundy do .000, .003, lub .007)                                                             |
| time           | 3 – 5-bajtowy typ do zapisu czasu z dokładnością do 100 ns. Ilość bajtów zależy od skali precyzji.                                                                                                                      |




Lekcja 5 - DDL (Data Definition Language)

Dane zapisywane jako data i czas powinny zostać zaprezentowane za pomocą typów przedstawionych w tabeli.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


**SQL**  
Structured Query Language

---

## Typy danych w SQL

- **Typ znakowy**

| Typ           | Opis.                                                                    |
|---------------|--------------------------------------------------------------------------|
| char          | Stało-znakowy typ o wielkości 1 – 8000 bajtów.                           |
| varchar(max)  | Zmienne-znakowy typ o długości max. 2 <sup>30</sup> – 1 bajtów.          |
| varchar       | Zmienne-znakowy typ o długości 1 – 8000 bajtów.                          |
| nchar         | Stało-znakowy typ Unicode o wielkości 1 – 4000 bajtów.                   |
| nvarchar      | Zmienne-znakowy typ Unicode o wielkości 1 – 4000 bajtów.                 |
| nvarchar(max) | Zmienne-znakowy typ Unicode o wielkości max. 2 <sup>30</sup> – 1 bajtów. |



Lekcja 5 - DDL (Data Definition Language)

Do wartości znakowych zalicza się typy przedstawione w tabeli



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 9

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language

## Typy danych w SQL

- Typy binarne

| Typ       | Opis.                                                                                   |
|-----------|-----------------------------------------------------------------------------------------|
| binary    | Przechowywany najczęściej jako stały strumień bajtów typ o wielkości 1 – 8000 bajtów.   |
| varbinary | Przechowywany najczęściej jako zmienny strumień bajtów typ o wielkości 1 – 8000 bajtów. |

**DAILY GROUP** Lekcja 5 - DDL (Data Definition Language)

Do wartości binarnych zalicza typy przedstawione w tabeli

Slajd 10

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language


## Typy danych w SQL

- Typy przestrzenne


| Typ       | Opis.                                                                                                                                                    |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| geography | Typ do przechowywania danych geograficznych, zaimplementowany w .NET CLR. Wykorzystywany jest głównie do zapisu pozycji GPS. Uwzględnia krzywiznę Ziemi. |
| geometry  | Typ do przechowywania typów geometrycznych (w szczególności figur), zaimplementowany w .NET CLR.                                                         |

**DAILY GROUP** Lekcja 5 - DDL (Data Definition Language)

SQL Server oferuje typy do reprezentowania obiektów geometrycznych i geograficznych



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


**SQL**  
Structured Query Language

---

## Typy danych w SQL


- **Pozostałe typy**

| Typ              | Opis                                                                                                                     |
|------------------|--------------------------------------------------------------------------------------------------------------------------|
| cursor           | Typ danych dla zmiennych parametrów wyjściowych procedury, które zawierają doniesienie do kursora.                       |
| hierarchid       | Typ o zmiennej długości danych, służący do reprezentowania pozycji danej informacji w hierarchii.                        |
| sql_variant      | Typ danych, przechowujący różne wartości typów obsługiwanych przez SQL Server. Jest odpowiednikiem typu var w języku C#. |
| table            | Typ tabelaryczny.                                                                                                        |
| uniqueidentifier | Typ umożliwiający automatyczne generowanie unikalnych liczb binarnych w bazie danych.                                    |
| xml              | Typ XML-owy.                                                                                                             |



Lekcja 5 - DDL (Data Definition Language)

Pozostałe typy zostały przedstawione w tabeli



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Typy danych w SQL

- **Zapamiętaj, że:**
  - SQL Server 2008+ oferuje, oprócz typów standardowych, typy do zapisu danych geograficznych oraz figur geometrycznych, często wykorzystywanych w systemach GIS.
  - Typ real to tak naprawdę float(24).
  - Do zapisu daty i czasu powinno używać się typów, jakie oferuje SQL Server 2008 R2, czyli: time, datetime2, datetimeoffset.



Lekcja 5 - DDL (Data Definition Language)



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## CREATE DATABASE

- Służy do tworzenia bazy danych
  - Składnia:

```
CREATE DATABASE nazwa_bazy
```

```
CREATE DATABASE Test;
```

```
USE Test;|
```

DAILY GROUP Lekcja 5 - DDL (Data Definition Language)

Pamiętajmy, że SZBD może zawierać wiele baz. Często używamy skrótu myślowego i nazywamy bazą danych SZBD, faktycznie system zarządzania bazą danych zawiera wiele baz danych.

USE wykorzystywany jest do wyboru bazy danych, w kontekście której zamierzamy pracować.

Slajd 14

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## CREATE TABLE

- Służy do tworzenia tabel
  - Składnia:

```
CREATE TABLE nazwa_tabeli
(nazwa_kolumny1 typ_danych,
nazwa_kolumny2 typ_danych,
nazwa_kolumny3 typ_danych,
...
);
```

DAILY GROUP Lekcja 5 - DDL (Data Definition Language)

CREATE służy do tworzenia nowych tabel, w których później możemy przechowywać różnego rodzaju dane. Tworząc nową tabelę musimy podać jej nazwę, nazwę kolumny/kolumn oraz typ danych danej kolumny. Można też podawać dodatkowe informacje, podczas tworzenia tabeli takie jak np. kodowanie znaków itd.

Slajd 15

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## CREATE TABLE

- Przykład 1

```
CREATE TABLE klienci1(
nr_klienta int,
nazwisko nvarchar(50),
imie nvarchar(50),
pesel nchar(11)
);
```

| nr_klienta | nazwisko | imie | pesel |
|------------|----------|------|-------|
|------------|----------|------|-------|

DAILY GROUP Lekcja 5 - DDL (Data Definition Language)

Przykład CREATE tworzący tabelę klientów. Tworząc tabelę trzeba zwrócić uwagę na to, jaki typ danych będziemy przechowywać, aby był on najlepiej dopasowany. Jeżeli przechowujemy wyłącznie liczby, to nie ma potrzeby tworzenia kolumny typu tekstowego itd.

Slajd 16

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language

### CREATE TABLE

- **Przykład 2 – klucz i autonumerowanie**

```
CREATE TABLE klienci2(
 nr_klienta int primary key IDENTITY(1,1),
 nazwisko nvarchar(50),
 imie nvarchar(50),
 pesel nchar(11)
);
```

- **IDENTITY(1,1) – pole autonumerowane – zaczynamy od wartości 1 i zwiększamy przy każdym insercie o 1.**

**DAILY** GROUP | Lekcja 5 - DDL (Data Definition Language)

Najczęściej chcemy posiadać kolumnę w tabeli, która będzie przechowywała unikalne wartości automatycznie numerowane. Zastosowanie tego jest bardzo proste. Dzięki takiemu rozwiązaniu każdy wiersz w tabeli posiada unikalny "identyfikator". W tym celu możemy wykorzystać **IDENTITY** oraz definicję klucza głównego.

Ciekawostka: Z punktu widzenia programisty, **IDENTITY** jako sposób na klucz główny (PK), jest straszną bolączką ORMów (mapowań relacyjno obiektowych).

Slajd 17

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language

### CREATE TABLE

- **Przykład 3 – wartości domyślnie, możliwość wartości puste**

```
CREATE TABLE klienci3(
 nr_klienta int primary key IDENTITY(1,1) NOT NULL,
 nazwisko nvarchar(50) NULL,
 imie nvarchar(50) NOT NULL DEFAULT 'Jan',
 pesel nchar(11) NULL
);
```

- **DEFAULT – wartość domyślna**
- **NULL/NOT NULL – czy możliwa jest wartość pusta**

**DAILY** GROUP | Lekcja 5 - DDL (Data Definition Language)

Tworząc tabelę mamy wpływ na to czy dana kolumna będzie dopuszczała wartości puste, bądź nie. Służy do tego wyrażenie **NULL** lub **NOT NULL** po typie zmiennej.

Dodatkowo możemy kolumnie nadać wartość domyślną – polecenie **DEFAULT**.

Slajd 18

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY | **SQL** Structured Query Language

### DROP DATABASE

- **Służy do usuwania bazy danych**

```
CREATE DATABASE Test2;



DROP DATABASE Test2;
```

**DAILY** GROUP | Lekcja 5 - DDL (Data Definition Language)

**DROP DATABASE** usuwa całą bazę danych. Warto zwrócić uwagę, że operacja jest nieodwracalna i należy z niej korzystać ze szczególną ostrożnością.



Slajd 19

**SQL**  
Structured Query Language

## DROP TABLE

- Służy do usuwania tabeli



```
CREATE TABLE klienci4(
 nr_klienta int primary key IDENTITY(1,1) NOT NULL,
 nazwisko nvarchar(50) NULL,
 imie nvarchar(50) NOT NULL DEFAULT 'Jan',
 pesel nchar(11) NULL
);

DROP TABLE klienci4;
```

**DAILY GROUP** Lekcja 5 - DDL (Data Definition Language)

DROP TABLE usuwa tabelę. Warto zwrócić uwagę, że operacja jest nieodwracalna i należy z niej korzystać ze szczególną ostrożnością.

Slajd 20

**SQL**  
Structured Query Language

## ALTER DATABASE

- Służy do dokonywania zmian bazy danych

```
ALTER DATABASE database_name
SET
{
 { <optionspec> [,...n] [WITH <termination>] }
}

<optionspec> ::=
{
 <auto_option>
 | <change_tracking_option>
 | <cursor_option>
 | <database_mirroring_option>
 | <date_correlation_optimization_option>
 | <db_encryption_option>
 | <db_state_option>
 | <db_update_option>
 | <db_user_access_option>
 | <external_access_option>
 | <parameterization_option>
 | <recovery_option>
 | <service_broker_option>
 | <snapshot_option>
 | <sql_option>
}

<auto_option> ::=
```

**DAILY GROUP** Lekcja 5 - DDL (Data Definition Language)

ALTER służy głównie służy do modyfikacji istniejących elementów w bazie. Początkujący użytkownicy będą głównie korzystali z polecenia **ALTER TABLE**, które będzie odnosiło się do tabel. Poleceniem tym możemy modyfikować tabele między innymi dodając i usuwając kolumny, zmieniając kolumny oraz typy kolumn.

W TSQL posiadamy inne typy zapytań **ALTER**, np. ALTER DATABASE (odnosi się do bazy danych), ALTER FUNCTION (odnosi się do funkcji), ALTER USER (odnosi się do użytkowników), ALTER VIEW (odnosi się do widoków), ALTER PROCEDURE (odnosi się do procedur). Na slajdzie przedstawiono możliwe opcje ALTER DATABASE. Najważniejsze z nich to zmiana kodowania, szyfrowania, opcji cursorów.

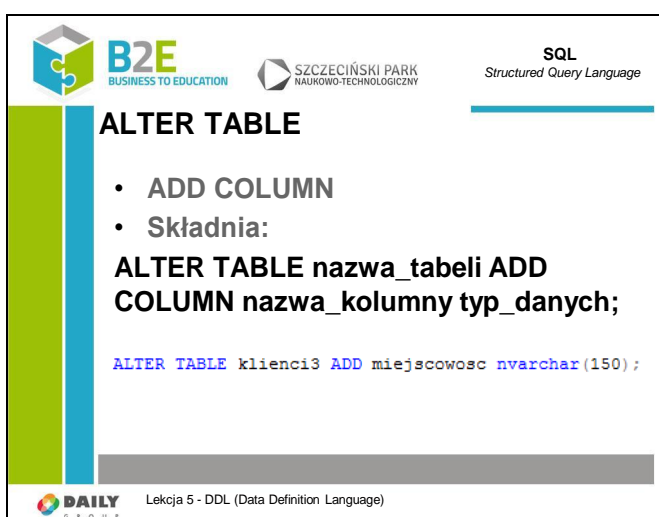


Slajd 21



Slide 21 features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main title is "ALTER TABLE". The content includes a bulleted list: "Służy do modyfikacji tabeli" and "Podstawowa składnia ALTER TABLE: ALTER TABLE nazwa\_tabeli [ADD|DROP|ALTER|..] nazwa\_kolumny ...". The footer contains the DAILY GROUP logo and the text "Lekcja 5 - DDL (Data Definition Language)".

Slajd 22



Slide 22 features the same header as slide 21. The main title is "ALTER TABLE". The content includes a bulleted list: "ADD COLUMN" and "Składnia: ALTER TABLE nazwa\_tabeli ADD COLUMN nazwa\_kolumny typ\_danych;". Below this, a code example is shown: `ALTER TABLE klienci3 ADD miejscowosc nvarchar(150);`. The footer contains the DAILY GROUP logo and the text "Lekcja 5 - DDL (Data Definition Language)".

Przykład ALTER TABLE dodający kolumnę do istniejącej tabeli.

Slajd 23



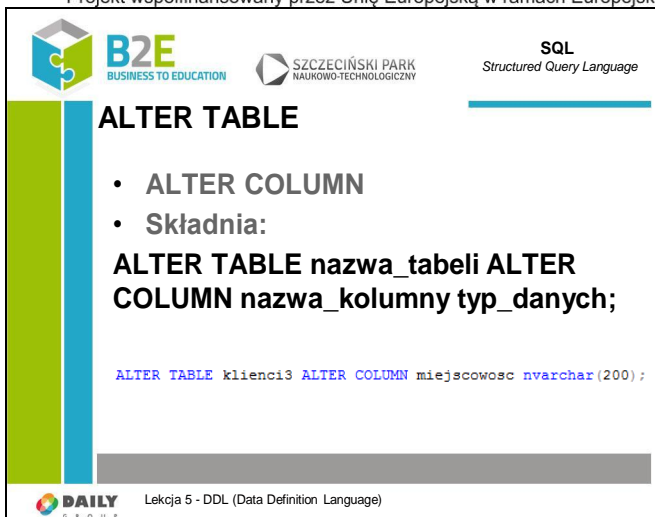
Slide 23 features the same header as slide 21. The main title is "ALTER TABLE". The content includes a bulleted list: "DROP COLUMN" and "Składnia: ALTER TABLE nazwa\_tabeli DROP COLUMN nazwa\_kolumny typ\_danych;". Below this, a code example is shown: `ALTER TABLE klienci3 DROP COLUMN miejscowosc;`. Another bulleted list item is "UWAGA!" followed by "Wrz z usunięciem kolumny zostaną usunięte wszystkie dane zapisane w kolumnie.". The footer contains the DAILY GROUP logo and the text "Lekcja 5 - DDL (Data Definition Language)".

Przykład ALTER TABLE usuwający kolumnę z istniejącej tabeli.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 24



SQL  
Structured Query Language

## ALTER TABLE

- ALTER COLUMN
- Składnia:

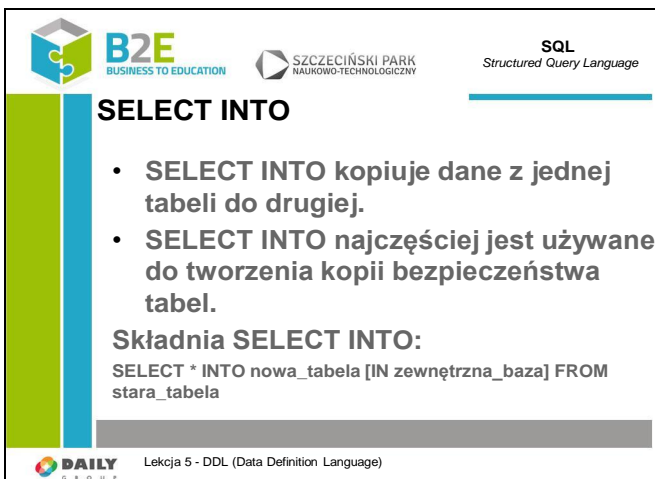
**ALTER TABLE nazwa\_tabeli ALTER COLUMN nazwa\_kolumny typ\_danych;**

```
ALTER TABLE klienci3 ALTER COLUMN miejscowosc nvarchar(200);
```

DAILY GROUP Lekcja 5 - DDL (Data Definition Language)

Przykład ALTER TABLE modyfikujący kolumnę w istniejącej tabeli.

Slajd 25



SQL  
Structured Query Language

## SELECT INTO

- SELECT INTO kopiuje dane z jednej tabeli do drugiej.
- SELECT INTO najczęściej jest używane do tworzenia kopii bezpieczeństwa tabel.

**Składnia SELECT INTO:**

```
SELECT * INTO nowa_tabela [IN zewnetrzna_baza] FROM stara_tabela
```

DAILY GROUP Lekcja 5 - DDL (Data Definition Language)

SELECT INTO – szybka metoda kopiowania tabel.



## SELECT INTO

```
SELECT * INTO pracownicy_kopia FROM pracownicy
```


```
SELECT * FROM pracownicy_kopia;
```

| nr_pracownika | nazwisko               | imie        | placa_zasadnicza | premia | pesel       | stanowisko          | nr_miejsca | przelozony |
|---------------|------------------------|-------------|------------------|--------|-------------|---------------------|------------|------------|
| 1             | Kowalski               | Jan         | 3312.00          | 500.00 | 82091104357 | Manager             | 1          | 7          |
| 3             | Przepiórka             | Marzena     | 2700.00          | NULL   | 89121203456 | Specjalista         | NULL       | 1          |
| 4             | Burzych                | Paweł       | 1900.00          | 500.00 | 78032309123 | Specjalista         | 2          | 7          |
| 5             | Makłowicz              | Marek       | 2000.00          | NULL   | 54013112345 | Specjalista         | 2          | 7          |
| 6             | Naramowicka-Przybylska | Magdalena   | 2100.00          | 200.00 | 77121312098 | Specjalista         | 1          | 1          |
| 7             | Witos                  | Jacek       | 3000.00          | 500.00 | 69100967234 | Manager             | 3          | 7          |
| 8             | Markowski              | Maksymilian | 2000.00          | NULL   | 67121209878 | Młodszy specjalista | NULL       | 7          |


Przykład SELECT INTO kopiujący dane z tabeli pracownicy.



Slajd 27



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Ćwiczenia

### Pracownicy

| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia      | pesel       | stanowisko          | nr_miejsca  | przelozony |
|---------------|-------------|-------------|------------------|-------------|-------------|---------------------|-------------|------------|
| 1             | Kowalski    | Jan         | 2300,00          | 250,00      | 82091104357 | Manager             | 1           | 7          |
| 2             | Nowak       | Karol       | 2700,00          | 100,00      | 80010123987 | Specjalista         | 1           | 1          |
| 3             | Przeziórka  | Marzena     | 2700,00          | <i>NULL</i> | 89121203456 | Specjalista         | <i>NULL</i> | 1          |
| 4             | Burzych     | Paweł       | 1900,00          | 500,00      | 78032309123 | Specjalista         | 2           | 7          |
| 5             | Makłowicz   | Marek       | 2000,00          | <i>NULL</i> | 54013112345 | Specjalista         | 2           | 7          |
| 6             | Naramowicka | Magdalena   | 2100,00          | 200,00      | 77121312098 | Specjalista         | 1           | 1          |
| 7             | Witos       | Jacek       | 3000,00          | 500,00      | 69100967234 | Manager             | 3           | 7          |
| 8             | Markowski   | Maksymilian | 2000,00          | <i>NULL</i> | 67121209878 | Młodszy specjalista | <i>NULL</i> | 7          |


### Klienci

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Mikołaj  | Wiśniewski | 82040112389 |
| 2          | Marcin   | Kaniewski  | 73122990123 |

### Miejsca

| nr_miejsca | ulica       | numer | miasto   | kod    | telefon        |
|------------|-------------|-------|----------|--------|----------------|
| 1          | Maja        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2          | Wąska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3          | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |



Lekcja 5 - DDL (Data Definition Language)

### Ćwiczenie 1

Utwórz nową bazę danych o nazwie Testowa\_Inicjały

### Ćwiczenie 2

Utwórz nowe tabele zgodnie ze slajdem

### Ćwiczenie 3

Usuń tabelę miejsca

### Ćwiczenie 4

Zmodyfikuj tabelę klienci – dodaj kolumnę wartość zamówienia (typ decimal z dwoma miejscami po przecinku)

### Ćwiczenie 5

Zmodyfikuj kolumnę wartość zamówienia – zmień typ na money z domyślną wartością 0

### Ćwiczenie 6

Skopiuj tabelę klienci do tabeli kontrahenci

### Ćwiczenie 7

Usuń kolumnę wartość zamówienia w tabeli kontrahenci

### Ćwiczenie 8

Usuń bazę testowa

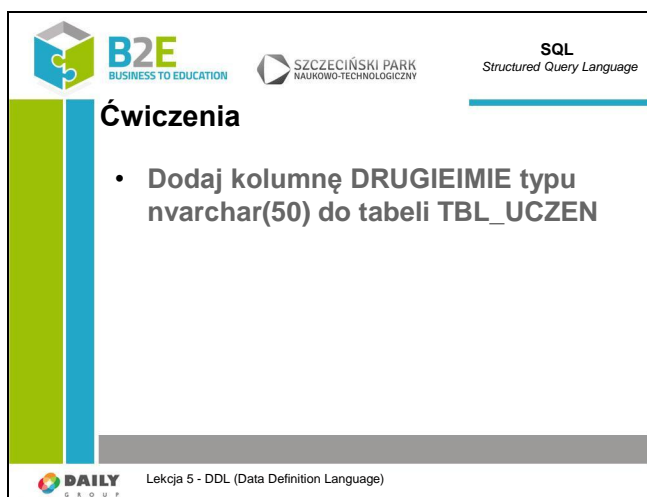
Slajd 28



Slide 28 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to 'Utwórz kopię zapasową tabeli TBL\_UCZEN' (Create a backup of the TBL\_UCZEN table). The footer includes the DAILY GROUP logo and the text 'Lekcja 5 - DDL (Data Definition Language)'.

```
SELECT * INTO
TBL_UCZEN_BCKP FROM
TBL_UCZEN
```

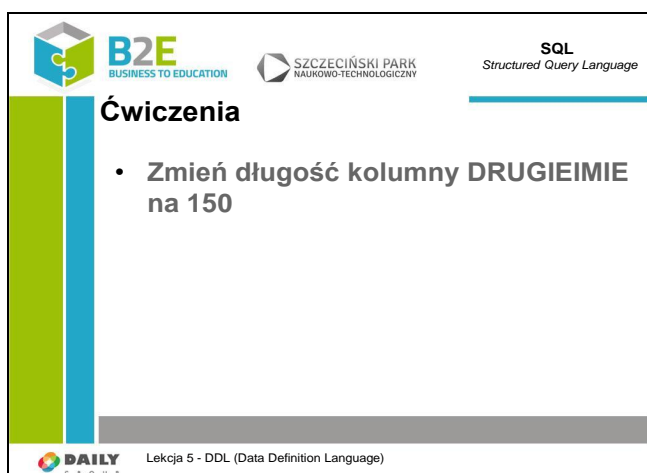
Slajd 29



Slide 29 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to 'Dodaj kolumnę DRUGIEIMIE typu nvarchar(50) do tabeli TBL\_UCZEN' (Add a column named DRUGIEIMIE of type nvarchar(50) to the TBL\_UCZEN table). The footer includes the DAILY GROUP logo and the text 'Lekcja 5 - DDL (Data Definition Language)'.

```
ALTER TABLE TBL_UCZEN ADD
COLUMN DRUGIEIMIE
nvarchar(50);
```

Slajd 30



Slide 30 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to 'Zmień długość kolumny DRUGIEIMIE na 150' (Change the length of the DRUGIEIMIE column to 150). The footer includes the DAILY GROUP logo and the text 'Lekcja 5 - DDL (Data Definition Language)'.

```
ALTER TABLE TBL_UCZEN ALTER
COLUMN DRUGIEIMIE
nvarchar(150);
```



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 31

**SQL**  
Structured Query Language

## Ćwiczenia

- Utwórz bazę danych Szkoła

**DAILY** Lektcja 5 - DDL (Data Definition Language)

CREATE DATABASE SZKOŁA

Slajd 32

**SQL**  
Structured Query Language

## Ćwiczenia

- Usuń bazę danych Szkoła

**DAILY** Lektcja 5 - DDL (Data Definition Language)

DROP DATABASE SZKOŁA

Slajd 33

**SQL**  
Structured Query Language

## Podsumowanie

### Najważniejsze polecenia DDL

|             |                                    |
|-------------|------------------------------------|
| CREATE      | Tworzy nowe obiekty w bazie danych |
| DROP        | Usuwa obiekty                      |
| ALTER       | Modyfikuje obiekty                 |
| SELECT INTO | Tworzy kopię tabeli i kopiuje dane |

**DAILY** Lektcja 5 - DDL (Data Definition Language)

Tabela podsumowująca poznane wyrażenia

## Ćwiczenia

Ćwiczenie zostało przedstawione na slajdach 27-32. Uczniowie przyswoją wiedzę dotyczącą tworzenia i modyfikacji tabel w bazie danych.

## Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą wiedzieć jak tworzyć i modyfikuje tabele w bazie danych.

## Lekcja 6 - DCL, uprawnienia, użytkownicy

### Cel lekcji

Celem lekcji jest zapoznanie uczniów z pojęciem użytkownika i loginu w bazie danych. Uczniowie dowiedzą się jak przyznawać i odbierać uprawnienia do obiektów bazodanowych użytkownikom i grupom użytkowników.

### Treść - slajdy z opisem

Slajd 1

Slide 1 is a title slide for the lesson. It features the B2E logo (Business to Education) and the Szczeciński Park Naukowo-Technologiczny logo. The main title is 'SQL Structured Query Language' and the subtitle is 'Lekcja 6 DCL, uprawnienia, użytkownicy'. It also includes logos for 'KAPITAŁ LUDZKI' and 'UNIA EUROPEJSKA FUNDUSZ ROZWOJU REGIONALNEGO'. The footer contains the DAILY GROUP logo and the text 'SQL Structured Query Language Lekcja 5'.

DCL (Data Control Language) – to język wykorzystywany do nadawania uprawnień użytkownikom i grupom do obiektów baz danych. W tej lekcji przyjrzymy się uprawnieniom, użytkownikom i nauczymy się tworzyć polecenia SQL do sterowania użytkownikami i uprawnieniami.

Slajd 2

Slide 2 is a reminder slide for the lesson. It features the B2E logo and the Szczeciński Park Naukowo-Technologiczny logo. The main title is 'Przypomnienie' and the subtitle is 'Lekcja 6 - DCL, uprawnienia, użytkownicy'. It lists the following SQL commands: CREATE, DROP, ALTER, and SELECT INTO. The footer contains the DAILY GROUP logo and the text 'Lekcja 6 - DCL, uprawnienia, użytkownicy'.

Przypomnienie materiału z poprzedniej lekcji



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 3

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Data Control Language

DCL (Data Control Language) ma zastosowanie do nadawania uprawnień do obiektów bazodanowych. Najważniejsze polecenia w tej grupie to:

- GRANT - służące do nadawania uprawnień do pojedynczych obiektów lub globalnie konkretnemu użytkownikowi
- REVOKE – służące do odbierania wskazanych uprawnień konkretnemu użytkownikowi (np. REVOKE ALL PRIVILEGES ON EMPLOYEE FROM PIOTR - odebranie użytkownikowi wszystkich praw do tabeli EMPLOYEE).
- DENY.

**DAILY**  
GROUP

Lekcja 6 - DCL, uprawnienia, użytkownicy

Definicja DCL

Slajd 4

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Użytkownicy i loginy

- Login – jest używany do połączenia z SZBD
- Użytkownik dotyczy konkretnej bazy danych
- Loginy i użytkowników łączy się ze sobą za pomocą tak zwanego mapowania

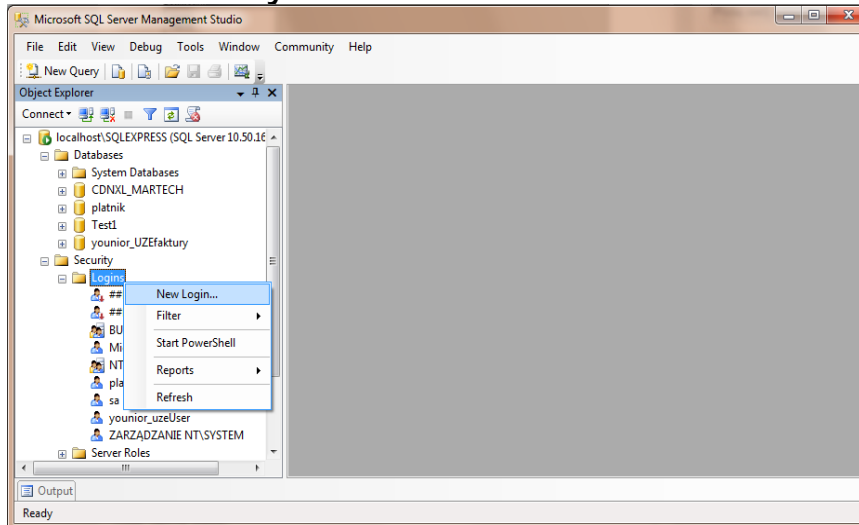
**DAILY**  
GROUP

Lekcja 6 - DCL, uprawnienia, użytkownicy

Definicja loginu i użytkownika. Login to obiekt funkcjonujący w kontekście SZDB, a użytkownik w kontekście bazy danych. Na kolejnych slajdach przedstawiono tworzenie loginu i użytkownika do niego za pomocą MS SQL Management Studio oraz za pomocą TSQL

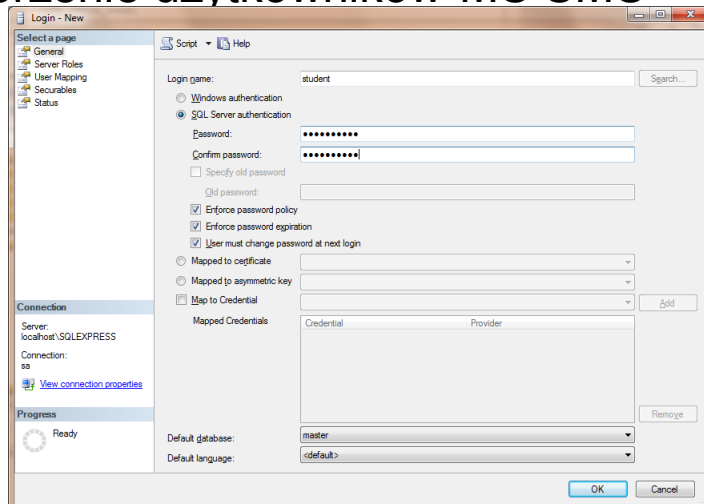


## Tworzenie użytkowników MS SMS



Sekcja z loginami znajduje się w drzewie SZBD w sekcji Security -> Logins


## Tworzenie użytkowników MS SMS



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego  
Po wybraniu New login pojawia nam się okno. W zakładce General wprowadzamy nazwę, wybieramy tryb logowania. Możliwa jest autentykacja Windows lub SQL. W przypadku Windows logować się z systemu windows, w którym użytkownik jest zalogowany. Ta opcja kierowana jest głównie do użytkowników domenowych. Jeśli chcemy umożliwić logowanie się użytkownikom z różnych systemów operacyjnych wybieramy opcję SQL Server authentication.

Sekcje Default Database oraz Default language są widoczne zarówno dla loginu z autoryzacją SQL jak i Windows. W Default Database określamy domyślną bazę danych dla loginu.

Slajd 7



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Tworzenie użytkowników MS SMS

- W przypadku wybrania autoryzacji SQL, należy wprowadzić hasło dla loginu oraz poniżej wybrać jedną lub kilka z trzech opcji:
  - Enforce password Policy
  - Enforce password expiration
  - User must change password at next logon



Lekcja 6 - DCL, uprawnienia, użytkownicy

W przypadku wybrania autoryzacji SQL, należy wprowadzić hasło dla loginu oraz poniżej wybrać jedną lub kilka z trzech opcji:

- Enforce password Policy – hasło nie może zawierać w sobie części nazwy loginu i nie może być krótsze niż 7 znaków oraz powinno zawierać cyfry, duże i małe litery oraz znaki nie alfanumeryczne. Informacje pobierane są z polityki grupowej.
- Enforce password expiration – hasło wygasa po przekroczeniu wartości określonej w polityce grupowej.
- User must change password at next logon – wymaga zmiany hasła użytkownika przy kolejnym logowaniu.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Tworzenie użytkowników MS SMS

- Na stronie **Server Roles** wybieramy role serwerowe, przypisane dla tego loginu:
  - bulkadmin
  - dbcreator
  - diskadmin
  - processadmin
  - securityadmin
  - serveradmin
  - setupadmin
  - sysadmin

 **DAILY**  
GROUP

Lekcja 6 - DCL, uprawnienia, użytkownicy

Na stronie **Server Roles** wybieramy role serwerowe, przypisane dla tego loginu:

bulkadmin – zezwala na operację masowego wstawiania danych (BULK INSERT),

dbcreator – zezwala na tworzenie, usuwanie, modyfikację bazy danych oraz dodawanie do niej nowych członków (CREATE DATABASE),

diskadmin – zezwala na zarządzanie plikami .mdf i .ldf (ALTER),

processadmin – zezwala na kontrolę procesów (ALTER ANY CONNECTION oraz ALTER SERVER STATE),


securityadmin – zezwala na zarządzanie loginami i uprawnieniami (ALTER ANY LOGIN),

serveradmin – zezwala na konfigurację całego serwera (ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN),


setupadmin – zezwala na zarządzanie serwerami połączonymi (ALTER ANY LINKED SERVER),

sysadmin – zezwala na pełną kontrolę nad bazami danych (CONTROL SERVER with GRANT).





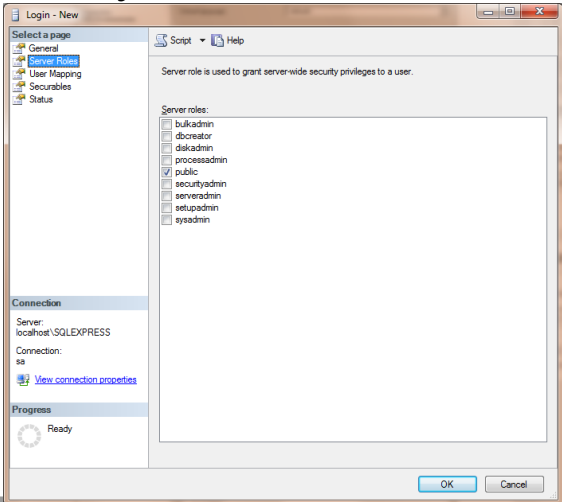
**B2E**  
BUSINESS TO EDUCATION




**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Tworzenie użytkowników MS SMS





Lekcja 6 - DCL, uprawnienia, użytkownicy

Na stronie **Server Roles** wybieramy role serwerowe, przypisane dla tego loginu:

bulkadmin – zezwala na operację masowego wstawiania danych (BULK INSERT),

dbcreator – zezwala na tworzenie, usuwanie, modyfikację bazy danych oraz dodawanie do niej nowych członków (CREATE DATABASE),

diskadmin – zezwala na zarządzanie plikami .mdf i .ldf (ALTER),

processadmin – zezwala na kontrolę procesów (ALTER ANY CONNECTION oraz ALTER SERVER STATE),

securityadmin – zezwala na zarządzanie loginami i uprawnieniami (ALTER ANY LOGIN),

serveradmin – zezwala na konfigurację całego serwera (ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN),

setupadmin – zezwala na zarządzanie serwerami połączonymi (ALTER ANY LINKED SERVER),

sysadmin – zezwala na pełną kontrolę nad bazami danych (CONTROL SERVER with GRANT).



SQL  
Structured Query Language

## Tworzenie użytkowników MS SMS

- User Mapping:
  - db\_accessadmin
  - db\_backupoperator
  - db\_datareader
  - db\_datawriter
  - db\_ddladmin
  - db\_denydatareader
  - db\_denydatawriter
  - db\_owner
  - db\_securityadmin
  - public

 Lekcja 6 - DCL, uprawnienia, użytkownicy

Na stronie **User Mapping** zaznaczamy, do której bazy użytkownik będzie posiadał uprawnienia (**Users mapped to this login**), a następnie w sekcji **Database role membership for** : nazwa\_bazy zaznaczamy role, które chcemy nadać:

db\_accessadmin – zezwala na dodawanie i usuwanie kont,

db\_backupoperator – zezwala na wykonywanie kopii zapasowych,

db\_datareader – zezwala na odczyt baz danych,

db\_datawriter – zezwala na zapisywanie i modyfikację baz danych,

db\_ddladmin – zezwala na modyfikację i usuwanie obiektów baz danych,

db\_denydatareader – nie zezwala na odczyt baz danych,

db\_denydatawriter – nie zezwala na zapisywanie i modyfikację baz danych,

db\_owner – zezwala na pełną kontrolę nad bazą danych,

db\_securityadmin – zezwala na zarządzanie uprawnieniami oraz rolami baz danych,

public – rola domyślna, zapewniająca minimum uprawnień.

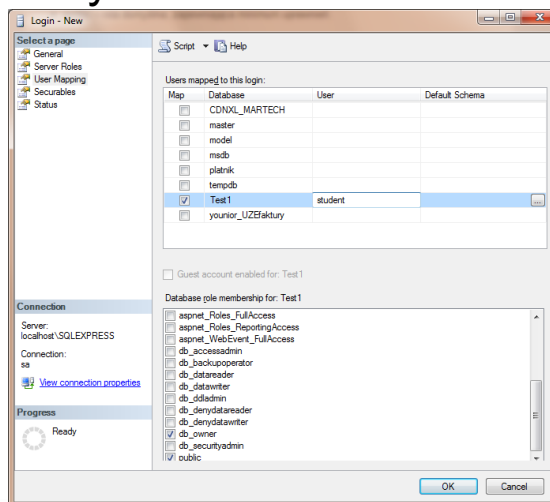


**B2E**  
BUSINESS TO EDUCATION



**SQL**  
Structured Query Language

## Tworzenie użytkowników MS SMS



Lekcja 6 - DCL, uprawnienia, użytkownicy

Na stronie **User Mapping** zaznaczamy, do której bazy użytkownik będzie posiadał uprawnienia (**Users mapped to this login**), a następnie w sekcji **Database role membership for** : nazwa\_bazy zaznaczamy role, które chcemy nadać:

db\_accessadmin – zezwala na dodawanie i usuwanie kont,

db\_backupoperator – zezwala na wykonywanie kopii zapasowych,

db\_datareader – zezwala na odczyt baz danych,

db\_datawriter – zezwala na zapisywanie i modyfikację baz danych,

db\_ddladmin – zezwala na modyfikację i usuwanie obiektów baz danych,

db\_denydatareader – nie zezwala na odczyt baz danych,

db\_denydatawriter – nie zezwala na zapisywanie i modyfikację baz danych,

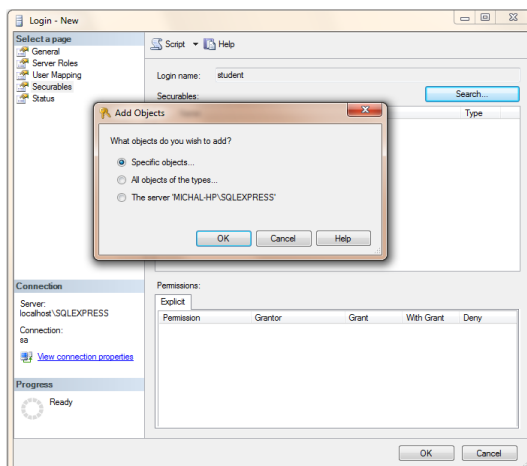
db\_owner – zezwala na pełną kontrolę nad bazą danych,

db\_securityadmin – zezwala na zarządzanie uprawnieniami oraz rolami baz danych,

public – rola domyślna, zapewniająca minimum uprawnień.



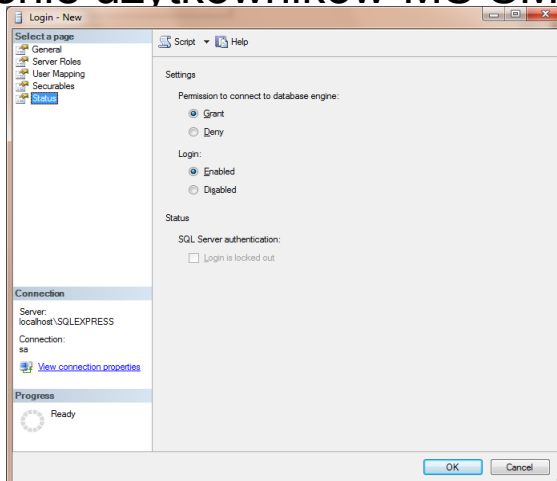
## Tworzenie użytkowników MS SMS



Strona **Securables** służy do przypisywania uprawnień do obiektów zabezpieczanych dla tego loginu.




## Tworzenie użytkowników MS SMS




Strona **Status** służy do nadawania lub odejmowania uprawnień dla loginu do łączenia się z bazą danych (**Permission to connect to Database engine**) oraz blokowania konta.



Slajd 14



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Tworzenie użytkowników TSQL

- Taki sam efekt uzyskamy za pomocą polecenia:

```
USE [master]
GO
CREATE LOGIN [student] WITH PASSWORD=N'Pa$$w0rd' MUST_CHANGE, DEFAULT_DATABASE=[master], CHECK_EXPIRATION=ON, CHECK_POLICY=ON
GO
EXEC master..sp_addsrvrolemember @loginame = N'student', @rolename = N'sysadmin'
GO
USE [Test1]
GO
CREATE USER [student] FOR LOGIN [student]
GO
USE [Test1]
GO
EXEC sp_addrolemember N'db_datareader', N'student'
GO
USE [Test1]
GO
EXEC sp_addrolemember N'db_datawriter', N'student'
GO
USE [student]
GO
EXEC sp_addrolemember N'db_owner', N'student'
GO
```





**DAILY**  
GROUP

Lekcja 6 - DCL, uprawnienia, użytkownicy

Tworzenie nowego loginu za pomocą TSQL i nadawanie odpowiednich uprawnień.






**SQL**  
*Structured Query Language*

## Podstawowe zasady bezpieczeństwa

- **Należy:**
  - Przyznawać użytkownikom uprawnienia tylko takie jakie potrzebują do swojej pracy
  - Monitorować przyznane uprawnienia
  - Pełne prawa do bazy powinien mieć jedynie użytkownik będący właścicielem bazy danych
- **Nie należy:**
  - Przypisywać użytkownikom wszystkich uprawnień, aby rozwiązać pewien problem.
  - Pozwalać zwykłym użytkownikom na tworzenie baz danych lub obiektów w bazach

 Lekcja 6 - DCL, uprawnienia, użytkownicy


Najważniejsze zasady, o których należy pamiętać podczas przydzielania uprawnień

**SQL**  
*Structured Query Language*

## Prawa dostępu

- Dla każdej tabeli lub widoku (relacji) można określić prawa dostępu do danych:
  - ALL – pozwala na wszystkie operacje na danych relacji
  - SELECT – pozwala użytkownikowi na odczytywanie danych relacji
  - DELETE – pozwala na usuwanie rekordów z relacji
  - INSERT – pozwala na wstawianie nowych rekordów do tabel lub widoków
  - UPDATE – pozwala na zmianę danych w relacji
  - EXECUTE – pozwala na wykonywaniu procedur zapamiętanych

 Lekcja 6 - DCL, uprawnienia, użytkownicy

Na slajdzie znajduje się zestawienie wszystkich praw dostępu, jakie można zastosować dla tabel i widoków.



Slajd 17

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

## Prawa dostępu

- **Uwaga** – w MS SQL Server prawa SELECT i UPDATE mogą być przyznawane do wybranych kolumn.
- Domyślnie, nadanie użytkownikowi praw dostępu nie umożliwia mu przekazywanie tych praw innym użytkownikom. Dopiero dołączenie opcji WITH GRANT OPTION pozwoli mu na dalsze przekazywanie posiadanych praw.
- Prawa dostępu przechowywane są w jednej z tabel systemowych.

**DAILY GROUP** Lekcja 6 - DCL, uprawnienia, użytkownicy

Slajd 18

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

## GRANT

- Do przyznawania praw dostępu utworzonym wcześniej rolam bądź użytkownikom służy polecenie GRANT.
- Przykładowa składnia:

```
GRANT nazwy_praw_dostępu
ON nazwa_tabeli_lub_widoku TO
nazwa_rol_i_lub_uzytkownika
GRANT ALL
ON klienci TO kierownik
WITH GRANT OPTION
```

**DAILY GROUP** Lekcja 6 - DCL, uprawnienia, użytkownicy

Definicja i przykład użycia polecenia GRANT

Slajd 19

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

## REVOKE

- Polecenie REVOKE zabiera uprawnienia, które zostały wcześniej przyznane.
- Przykładowa składnia:

```
REVOKE nazwy_praw_dostępu
ON nazwa_tabeli_lub_widoku TO
nazwa_rol_i_lub_uzytkownika
REVOKE ALL
ON klienci TO kierownik
```

**DAILY GROUP** Lekcja 6 - DCL, uprawnienia, użytkownicy

Polecenie REVOKE – służące od odbierania praw do obiektów

Slajd 20

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## DENY

- Zabrania wykonywania operacji, jest silniejsze niż grant
- Nie można wykonywać czynności. Uprawnienie nie może zostać zmienione w wyniku członkostwa w roli.
- Przykładowa składnia:  
DENY nazwy\_praw\_dostępu  
TO nazwa\_rol\_i\_lub\_uzytkownika

```
DENY SELECT ON klienci TO kierownik;
```

DAILY GROUP Lekcja 6 - DCL, uprawnienia, użytkownicy

Inaczej niż w przypadku polecenia REVOKE, polecenie DENY bezpośrednio zabiera uprawnienie polecenia. Przykładowo, jeżeli Jan jest członkiem roli bazy danych, a rola ta ma uprawnienie CREATE TABLE, Jan może również tworzyć tabele. Jednak, jeżeli Jan powinien mieć zabronione tworzenie tabel, mimo tego, że jako członek roli posiada to uprawnienie, można zabronić Janowi wykonywania tego polecenia. Dlatego, Jan nie będzie mógł uruchomić wyrażenia CREATE TABLE, pomimo tego, że normalnie rola przydzieliła mu to prawo.

Slajd 21

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Przykłady

- aby przyznać użytkownikowi Jan uprawnienie do tworzenia widoku w bazie, należy uruchomić:  

```
GRANT CREATE VIEW TO Jan
```
- aby cofnąć uprawnienie do tworzenia widoków i tabel dla użytkowników Jana i Anny należy uruchomić:  

```
REVOKE CREATE TABLE, CREATE VIEW FROM Anna, Jan
```
- aby przyznać Joe wszystkie uprawnienia w bazie danych należy uruchomić:  

```
GRANT ALL TO Jan
```

DAILY GROUP Lekcja 6 - DCL, uprawnienia, użytkownicy

Przegląd kilku przykładów jest najlepszym sposobem na zrozumienie działania omówionych poleceń.



Slajd 22

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main title is 'Ćwiczenia' (Exercises). At the bottom, there is a logo for DAILY GROUP and the text 'Lekcja 6 - DCL, uprawnienia, użytkownicy'.

### Ćwiczenie 1

Utwórz nową bazę danych o nazwie Testowa\_Inicjały

### Ćwiczenie 2

Utwórz użytkownika Nazwisko\_studenta1 z poziomu MS SMS z domyślną bazą Testowa\_Inicjały

### Ćwiczenie 3

Utwórz użytkownika Nazwisko\_studenta2 z poziomu TSQL z domyślną bazą Testowa\_Inicjały

### Ćwiczenie 4

Wykorzystaj procedury:

sp\_password

sp\_defaultdb

sp\_defaultlanguage

sp\_helplogins

sp\_droplogin

dla użytkownika Nazwisko\_studenta2

### Ćwiczenie 5

Wykorzystując procedurę *sp\_changedbowner* zmień właściciela bazy danych Testowa\_Inicjały na Nazwisko\_studenta1

### Ćwiczenie 6

Wykorzystując polecenia GRANT, REVOKE i DENY ustaw wybrane 3 uprawnienia (CREATE DATABASE, CREATE TABLE, CREATE PROCEDURE, CREATE DEFAULT, CREATE RULE, CREATE VIEW, CREATE FUNCTION, BACKUP DATABASE I BACKUP LOG) wybranym użytkownikom lub rolam (w razie potrzeby utworzyć dodatkowych użytkowników lub role)

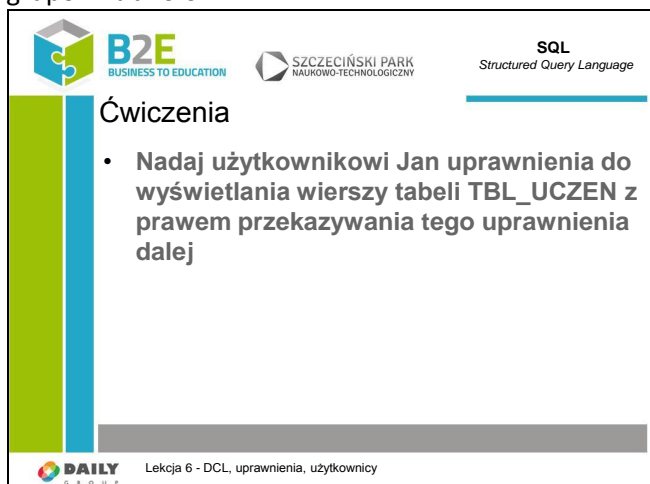
### Ćwiczenie 7

Wykorzystując polecenia GRANT, REVOKE i DENY ustaw wybrane uprawnienia (SELECT, INSERT, UPDATE, DELETE, EXECUTE, REFERENCES) dla wybranych obiektów bazy *Test* poszczególnym użytkownikom, grupom lub rolam

### Ćwiczenie 8

Zobrazuj na przykładach skutki przyznania lub zabronienia poszczególnym użytkownikom, grupom lub rolam

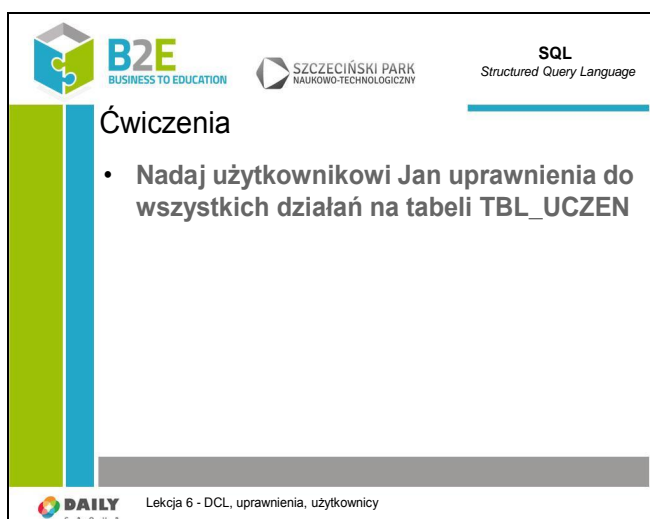
Slajd 23



Slide 23 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia'. A single bullet point reads: 'Nadaj użytkownikowi Jan uprawnienia do wyświetlania wierszy tabeli TBL\_UCZEN z prawem przekazywania tego uprawnienia dalej'. The footer includes the DAILY GROUP logo and the text 'Lekcja 6 - DCL, uprawnienia, użytkownicy'.

```
GRANT SELECT ON TBL_UCZEN
TO Jan WITH GRANT OPTION;
```

Slajd 24



Slide 24 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia'. A single bullet point reads: 'Nadaj użytkownikowi Jan uprawnienia do wszystkich działań na tabeli TBL\_UCZEN'. The footer includes the DAILY GROUP logo and the text 'Lekcja 6 - DCL, uprawnienia, użytkownicy'.

```
GRANT ALL ON TBL_UCZEN TO
Jan;
```



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 25

REVOKE ALL ON TBL\_UCZEN TO Jan;

Slajd 26

DENY ALL ON TBL\_UCZEN TO Jan CASCADE;

Slajd 27

EXEC sp\_helpuser

**SQL**  
Structured Query Language

## Podsumowanie

|        |                      |
|--------|----------------------|
| GRANT  | nadawanie uprawnień  |
| DENY   | odmawianie uprawnień |
| REVOKE | odbieranie uprawnień |

Kapitał Ludzki  
Szczytowa Inwestycja

Unia Europejska  
Europejski Fundusz Społeczny

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**DAILY** Lektora 6 - DCL, uprawnienia, użytkownicy

Tabela podsumowująca poznane wyrażenia

### Ćwiczenia

Ćwiczenie zostało przedstawione na slajdach 22-27. Uczniowie utworzą nową bazę danych, użytkowników oraz tabelę, a następnie dokonają szeregu modyfikacji uprawnień tych obiektów.

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie nauczą się świadomie przydzielać prawa do obiektów bazodanowych, poznają najlepsze praktyki, a następnie w czasie ćwiczeń praktycznie wykorzystają zdobytą wiedzę.

## Lektora 7 - Widoki i Funkcje

### Cel lekcji

Celem lekcji jest przedstawienie dwóch typów obiektów w bazie danych: widoków oraz funkcji. Uczniowie po odbyciu lekcji powinni być w stanie samodzielnie zaprogramować proste widoki i funkcje użytkownika.

### Treść - slajdy z opisem

Slajd 1

**SQL**  
Structured Query Language

# SQL

## Structured Query Language

### Lektora 7: widoki i funkcje

Kapitał Ludzki  
Szczytowa Inwestycja

Unia Europejska  
Europejski Fundusz Społeczny

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**DAILY** SQL Structured Query Language Lektora 7

W tej lekcji poznamy inne niż tabele obiekty bazy danych: widoki oraz funkcje.

Widoki umożliwiają „zapisanie” skomplikowanych zapytań do późniejszego – szybkiego wykorzystania. Funkcje to obiekty, które już poznaliśmy. Na poprzednich lekcjach używaliśmy funkcji systemowych MIN, MAX, SUM, AVG. SQL Serwer umożliwia definicję własnych funkcji użytkownika – w tej lekcji dowiemy się w jaki sposób.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 2

**Przypomnienie**

- Login
- Użytkownik
- GRANT
- REVOKE
- DENY

Przypomnienie materiału z poprzedniej lekcji.


Slajd 3

**Widoki**

- Widok (lub perspektywa) to tabela wirtualna określona przez zapytanie SQL-owe.
- Zapytanie ma własną nazwę i jest przechowywane w bazie danych. Zmiana danych w tabeli bazowej na ogół ma wpływ na dane w widoku zbudowanym na bazie tej tabeli.
- Perspektywy dają duże możliwości odnośnie przetwarzania danych.

Widoki to tabele wirtualne. Wykorzystywane są w sytuacji, gdy np. mamy tabelę posiadającą 10 kolumn, a na potrzebę aplikacji chcemy prezentować tylko część z nich.



**SQL**  
Structured Query Language



---

## Widoki

- Przy tworzeniu widoków możliwe jest:
  - złączenie, dowolnie wybranych tabel,
  - wykonanie operacji selekcji, projekcji i sortowania,
  - wykonanie instrukcje grupowania,
  - użycie podzapytania.

**DAILY GROUP** Lekcja 7 – widoki i funkcje

Slajd wyjaśnia jakie operacje są możliwe przy tworzeniu widoków.

**SQL**  
Structured Query Language

---

## Widoki

- Standardowa składnia definicji instrukcji tworzenia widoku:
 

```
CREATE VIEW <nazwa widoku> [(<nazwa kol> [, <nazwa kol> ...])
AS
(SELECT <instrukcja>
[WITH [CASCADED][LOCAL] CHECK OPTION]);
```

**DAILY GROUP** Lekcja 7 – widoki i funkcje

Definicja widoku jest podobna do wszystkich definicji obiektów w bazie danych. Create, rodzaj obiektu, nazwy kolumn, słówko AS (jako) i SELECT, który chcemy zapisać w widoku.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Widoki

- Listy kolumn używa się gdy:
  - jakikolwiek dwie kolumny mają identyczne nazwy,
  - kolumny zawierają wartości wyliczalne,
  - występują połączone kolumny o różnych nazwach.

**DAILY GROUP** Lekcja 7 – widoki i funkcje

W widokach możemy zapisać `SELECT * FROM table` lub `SELECT col1, col2, ... FROM table`. Slajd wyjaśnia kiedy unikać gwiazdki.

Slajd 7

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Widoki

- Usuwanie widoków:
  - Podobnie jak w przypadku tabel, do kasowania widoków stosuje się DROP:  
`DROP VIEW <nazwa perspektywy>`

**DAILY GROUP** Lekcja 7 – widoki i funkcje

Usuwanie widoków – tak jak wszystkich obiektów w bazie danych – dokonywane jest instrukcją DROP

Slajd 8

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Widoki - przykład

```
CREATE VIEW oaklanders
AS
SELECT au_fname, au_lname, title
FROM authors, titles, titleauthors
WHERE authors.au_id = titleauthors.au_id
AND titles.title.id = titleauthors.au_id
AND city = 'Oakland';
```

**DAILY GROUP** Lekcja 7 – widoki i funkcje

Przykład prezentuje widok, który łączy dwie tabele i zawęży zwracane wartości tylko do autorów pochodzących z miasta Oakland.




**SQL**  
Structured Query Language

---

## Widoki - przykład

```
CREATE VIEW currentinfo (PUB, TYPE, INCOME, AVG_PRICE, AVG_SALES)
AS
 SELECT pub_id, type, sum(price*ytd_sales), avg(price), avg(ytd_sales)
 FROM titles
 GROUP BY pub_id, type;
```

**DAILY** GROUP    Lekcja 7 – widoki i funkcje

Przykład prezentuje widok, który dokonuje obliczeń na kolumnach tabeli titles.




**SQL**  
Structured Query Language

---

## Widoki - przykład

```
CREATE VIEW cities (Author, Authorcity, Pub, Pubcity)
AS
 SELECT au_lname, authors.city, pub_name, publishers.city
 FROM authors, publishers
 WHERE authors.city = publishers.city;
```

**DAILY** GROUP    Lekcja 7 – widoki i funkcje

Przykład prezentuje widok, który łączy dwie tabele i zawęża zwracane wartości tylko do autorów pochodzących z miasta Oakland.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 11

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Widoki modyfikowalne

- Dane w tablicach bazowych mogą być modyfikowane (wstawiane, usuwane) z poziomu widoku tylko przy spełnieniu określonych wymagań:
  - zmiany muszą być określone jednoznacznie.
  - widok oparty jest na jednej tablicy bazowej;
  - w definicji zapytania nie występują funkcje agregujące

DAILY GROUP Lekcja 7 – widoki i funkcje

Dane w tablicach bazowych mogą być modyfikowane (wstawiane, usuwane) z poziomu widoku tylko przy spełnieniu określonych wymagań:

- zmiany muszą być określone jednoznacznie.
- perspektywa oparta jest na jednej tablicy bazowej; odniesienia są tylko do kolumn tej tablicy zawiera tylko jedno zapytanie (bez UNION, EXCEPT, INTERSECT),
- w definicji zapytania nie występują funkcje agregujące (nie ma GROUP BY, HAVING, DISTINCT).

Slajd 12

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Więź CHECK w widokach

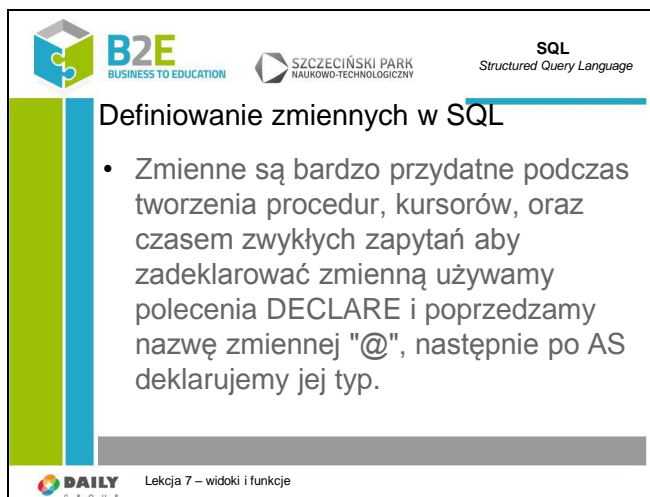
- Jeśli w zapytaniu tworzącym widok jest klauzula WHERE, więź CHECK uniemożliwi takie dodanie/ zmodyfikowanie danych do widoku, które naruszałoby tę klauzulę

```
CREATE VIEW nazwa
AS SELECT treść_zapytania.select WHERE warunek
[WITH [CASCADED | LOCAL] CHECK OPTION];
```

DAILY GROUP Lekcja 7 – widoki i funkcje

CASCADED i LOCAL mają znaczenie jeśli widok tworzony jest na podstawie innych widoków. Jeżeli wybierzemy CASCADED, klauzule WHERE "podwidoków" też są sprawdzane, nawet jeśli nie nałożono na nie jawnie więzu (CONSTRAINT – wyjaśniony w kolejnych lekcjach) CHECK.

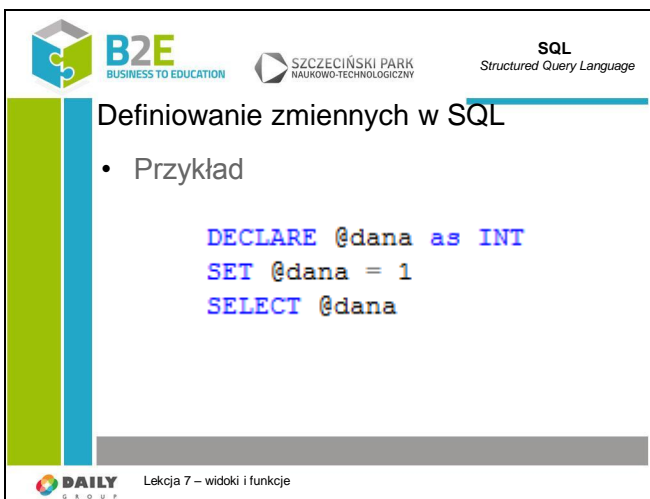
Slajd 13



Slide 13 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main title is "Definiowanie zmiennych w SQL". A bullet point explains that variables are useful for creating procedures, cursors, and queries, and that the DECLARE statement is used to declare a variable, with the @ symbol followed by the variable name and its type. The footer includes the DAILY GROUP logo and the text "Lekcja 7 – widoki i funkcje".

Typy zmiennych mogą być dokładnie takie same jak typy tabel, które omawialiśmy na poprzednich lekcjach

Slajd 14



Slide 14 content: The slide features the same header as slide 13. The main title is "Definiowanie zmiennych w SQL". A bullet point labeled "Przykład" is followed by a code block showing the SQL statements: DECLARE @dana as INT, SET @dana = 1, and SELECT @dana. The footer includes the DAILY GROUP logo and the text "Lekcja 7 – widoki i funkcje".

Definicja zmiennej @dana, przypisanie do niej wartości 1, a następnie wyświetlenie wartości za pomocą znanego nam polecenia SELECT. Zmiennej możemy użyć nie tylko w selekcje ale również w warunkach.



Slajd 15

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Definiowanie zmiennych w SQL

- Przykład

```

DECLARE @dana as INT
SET @dana = 3000
Select imie, nazwisko FROM pracownicy WHERE placa_zasadnicza >= @dana

```

| Results |       | Messages |  |
|---------|-------|----------|--|
|         | imie  | nazwisko |  |
| 1       | Jan   | Kowalski |  |
| 2       | Jacek | Witos    |  |

Lekcja 7 – widoki i funkcje

Pamiętacie naszą tabelę pracownicy? Zdefiniowanie warunku za pomocą zmiennej @dana

Slajd 16

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY



**SQL**  
Structured Query Language

## Zmienne tablicowe

- Zmienne tablicowe to tabele danych przechowywane w pamięci (zapis do nich jest bardzo szybki).
- Zmienia deklarujemy w ten sam sposób co zwykłą zmienną DECLARE @dana następnie TABLE i w nawiasach nazwy kolumn i ich typy)

Lekcja 7 – widoki i funkcje

Definicja zmiennych tablicowych





**SQL**  
Structured Query Language

## Zmienne tablicowe

- Przykład

```
DECLARE @dana TABLE (koszty int, miesiac char(10))
INSERT INTO @dana (koszty, miesiac)
SELECT Koszty, miesiac FROM @dana
---kolejny selekt---
SELECT * FROM @dana
```

 Lekcja 7 – widoki i funkcje

Na początku deklarujemy zmienną tablicową @dana, jej kolumny i ich typy, przypadku zmiennej skalarnej nie możemy użyć SET ale możemy umieścić dane w tabeli przy użyciu INSERT INTO wyniku z poniższego zapytania. Zwróćcie szczególną uwagę na zapis ---kolejny selekt--- jest to działanie celowe, aby zademonstrować jak wprowadza się komentarze w t-sql "--" oznacza komentarz lub /\* wykomentowany tekst \*/.

Ostatni SELECT powoduje wyświetlenie zawartości tabeli @dana. Uwaga po zakończeniu ostatniego select wartość zmiennej jest wymazywana.




**SQL**  
Structured Query Language

## Zmienne systemowe

- Zmienne systemowe oznaczone są dwoma znakami "@". Często używane zmienne systemowe:
  - @@ERROR** - numer ostatniego błędu
  - @@FETCH\_STATUS** - czy kursor pobrał wiersz (0 gdy pobrał)
  - @@IDENTITY** - zawiera ostatnio wygenerowaną wartość IDENTITY (przydatne, gdy chcemy użyć ID wstawionego przez INSERT wiersza)
  - @@ROWCOUNT** - zwraca liczbę wierszy, na których operowała ostatnia instrukcja SQL
  - @@VERSION** - zwraca informację o wersji SQL Serwera

 Lekcja 7 – widoki i funkcje

Zmienne systemowe – to zdefiniowane już w SQL serwer dostępne w każdej bazie danych.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 19

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

### Podział funkcji

- Funkcje systemowe, np.
  - GETDATE()
  - CAST()
  - ROUND()
  - SIN()
  - HOSTNAME()
  - ISNULL()
- Funkcje użytkownika

```
SELECT GETDATE (), HOST_NAME ()
```

| (No column name)        | (No column name) |
|-------------------------|------------------|
| 2013-02-08 18:23:38.903 | MICHAL-HP        |

**DAILY GROUP** Lekcja 7 – widoki i funkcje

Funkcje dzielimy na funkcje systemowe – dostępne w każdej instalacji MSSQL oraz funkcje użytkownika, które zostały napisane i zachowane w konkretnej bazie danych.

Slajd 20

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

### Funkcje systemowe - przykłady

- CONVERT(typ\_danych, wyrażenie [, styl]) - dokonuje konwersji typów danych (styl jest używany przy konwersji do daty, typów walutowych itp.)
- GETDATE() - zwraca aktualną datę systemową
- LEFT(napis, ile\_znaków) - zwraca określoną liczbę znaków napisu rozpoczynając od lewej
- LEN(napis) - zwraca długość napisu
- REPLACE(napis, wzorzec, napis\_do\_zamiany) - wyszukuje i zamienia fragment napisu
- RIGHT(napis, ile\_znaków) - zwraca określoną liczbę znaków napisu rozpoczynając od prawej
- SUBSTRING(napis, od, do) - zwraca określoną część napisu

**DAILY GROUP** Lekcja 7 – widoki i funkcje

Przykłady najczęściej wykorzystywanych funkcji systemowych.

Slajd 21

**B2E** BUSINESS TO EDUCATION | **SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY** | **SQL** Structured Query Language

### Tworzenie funkcji użytkownika

- Podział funkcji użytkownika:
  - Funkcje skalarne (zwracają wartość)
  - Funkcje tablicowe (zwracają tabelę)

**DAILY GROUP** Lekcja 7 – widoki i funkcje

Zajmijmy się teraz tworzeniem funkcji w t-sql. Możemy stworzyć funkcje użytkownika które zwracają skalarnie lub tablicowe wartości. Funkcji możemy używać w widokach, w innych funkcjach, w aplikacjach oraz procedurach.





**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

---

## Funkcje skalarne

- **Składnia:**

```
CREATE FUNCTION function_name
(@parameter_name parameter_data_type
 [= default] [READONLY] }
 [,...n]
)
RETURNS return_data_type
[AS]
BEGIN
 function_body
RETURN scalar_expression
END
```



DAILY GROUP

Lekcja 7 – widoki i funkcje

Ogólna składnia tworzenia funkcji wygląda następująco. Funkcja przyjmuje parametry (jeden lub wiele) oraz zwraca wartość (funkcja skalarna zwraca jedną wartość). Podczas definicji funkcji musimy określić ilość, typ i nazwy parametrów oraz typ zwracanej wartości.

Zamiast function body – wprowadza się treść funkcji.



Slajd 23

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Funkcje skalarne

```

CREATE FUNCTION tylko_data (@DATE smalldatetime)
RETURNS smalldatetime
AS
BEGIN
declare @return as smalldatetime

 set @return = (convert(smalldatetime,convert(nvarchar(11),left(@DATE,11),120),120))

return @return
END

SELECT dbo.tylko_data('2012-12-12 12:12:12')

```

| Results             | Messages |
|---------------------|----------|
| (No column name)    |          |
| 2012-12-12 00:00:00 |          |

Lekcja 7 – widoki i funkcje

Powyższy przykład demonstruje w jaki sposób stworzyć funkcje tylko data, czyli datę bez godziny i minuty.

Slajd 24

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Funkcje tablicowe

- Funkcje o wartościach tabelarycznych (ang. table-valued function) – są nazywane również sparametryzowanymi widokami, zwracają jako wartość tablicę rekordów, na przykład:

Lekcja 7 – widoki i funkcje

Definicja funkcji tablicowych, funkcje te często są nazywane widokami parametryzowanymi.



## Funkcje tablicowe

```
create table dane (lp int identity(1,1), nazwa nvarchar(30), wartosc int default 0)
insert into dane (nazwa) values ('a'), ('b'), ('c'), ('d'), ('e')
insert into dane (nazwa, wartosc) values ('1',1), ('2',1), ('3',1), ('4',1), ('5',1)
go

CREATE FUNCTION dbo.FunkcjaTablicowa(@wartosc bit)
RETURNS TABLE
AS
RETURN select * from (select lp, nazwa, wartosc from dane where wartosc=@wartosc) as subk
go

select * from dbo.FunkcjaTablicowa(1)

drop function dbo.FunkcjaTablicowa
drop table dane|
```



Przykład demonstruje użycie funkcji w celu otrzymania zapytania z tabeli dane z wartością jako warunkiem i parametrem funkcji.



## Ćwiczenia

### Pracownicy

| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia | pesel       | stanowisko          | nr_miejsca | przelozony |
|---------------|-------------|-------------|------------------|--------|-------------|---------------------|------------|------------|
| 1             | Kowalski    | Jan         | 2300,00          | 250,00 | 82091104357 | Manager             | 1          | 7          |
| 2             | Nowak       | Karol       | 2700,00          | 100,00 | 80010123987 | Specjalista         | 1          | 1          |
| 3             | Przepiórka  | Marzena     | 2700,00          | NULL   | 89121203456 | Specjalista         | NULL       | 1          |
| 4             | Burzych     | Paweł       | 1900,00          | 500,00 | 78032309123 | Specjalista         | 2          | 7          |
| 5             | Makłowicz   | Marek       | 2000,00          | NULL   | 54013112345 | Specjalista         | 2          | 7          |
| 6             | Naramowicka | Magdalena   | 2100,00          | 200,00 | 77121312098 | Specjalista         | 1          | 1          |
| 7             | Witos       | Jacek       | 3000,00          | 500,00 | 69100967234 | Manager             | 3          | 7          |
| 8             | Markowski   | Maksymilian | 2000,00          | NULL   | 67121209878 | Młodszy specjalista | NULL       | 7          |

### Klienci

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Mikołaj  | Wiśniewski | 82040112389 |
| 2          | Marcin   | Kaniewski  | 73122990123 |

### Miejsca

| nr_miejsca | ulica       | numer | miasto   | kod    | telefon        |
|------------|-------------|-------|----------|--------|----------------|
| 1          | Mała        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2          | Wąska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3          | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |





Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

### Ćwiczenie 1

Utwórz widok zwracający pracowników i miejsca pracy

### Ćwiczenie 2

Utwórz widok zwracający zatrudnione kobiety. Zaktualizuj pensje zasadniczą o 20% poprzez wykorzystanie UPDATE na widoku. Sprawdź czy dane zmieniły się w tabeli.

### Ćwiczenie 3

Utwórz zmienną tabelaryczną z takimi samymi kolumnami jak klienci. Wpisz kilka przykładowych wartości do zmiennej. Dodaj wartości ze zmiennej do tabeli klienci

### Ćwiczenie 4

Utwórz funkcję zwracającą średnią płacę, na podstawie lokalizacji (miejsca pracy)

### Ćwiczenie 5

Utwórz funkcję zwracającą premię na podstawie numeru pracownika

### Ćwiczenie 6

Utwórz funkcję tabelaryczną zwracającą imię, nazwisko i pesel pracownika na podstawie miejsca zatrudnienia

### Ćwiczenie 7

Utwórz funkcję zwracającą adres (w formie tabeli) na podstawie numeru pracownika

### Ćwiczenie 8

Zmień funkcję z ćwiczenia 7, tak aby zwracała wartość skalarną

### Ćwiczenie 9

Usuń utworzone widoki i funkcje

Slajd 27

The slide contains the following elements:

- Logos: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, SQL Structured Query Language, and DAILY C.A.F.E.
- Section header: **Ćwiczenia**
- Exercise list:
  - Wyświetl bieżącą datę
- Footer: Lekcja 7 – widoki i funkcje

SELECT GETDATE()

Slajd 28



Slide 28 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point reads: 'Wyświetl ostatni kod błędu' (Display the last error code). The footer includes the DAILY GROUP logo and the text 'Lekcja 7 – widoki i funkcje' (Lesson 7 – views and functions).

```
SELECT @@ERROR
```

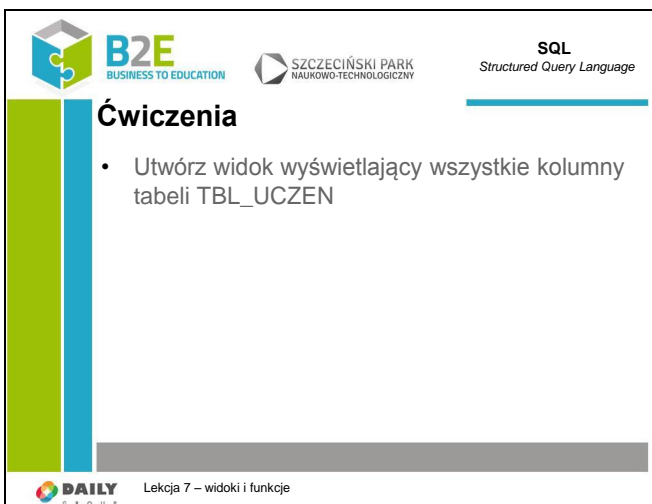
Slajd 29



Slide 29 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point reads: 'Zadeklaruj zmienną liczbową ILE i przypisz jej wartość 5' (Declare a numeric variable ILE and assign it the value 5). The footer includes the DAILY GROUP logo and the text 'Lekcja 7 – widoki i funkcje' (Lesson 7 – views and functions).

```
DECLARE @ILE int
SET @ILE=5
```

Slajd 30



Slide 30 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point reads: 'Utwórz widok wyświetlający wszystkie kolumny tabeli TBL\_UCZEN' (Create a view displaying all columns of the TBL\_UCZEN table). The footer includes the DAILY GROUP logo and the text 'Lekcja 7 – widoki i funkcje' (Lesson 7 – views and functions).

```
CREATE VIEW mojewidok
AS
SELECT * FROM TBL_UCZEN
```



Slajd 31

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Ćwiczenia

- Utwórz funkcję zwracającą liczbę uczniów w klasie. Funkcja przyjmuje nazwę klasy w parametrze

DAILY Lektora 7 – widoki i funkcje

```
CREATE FUNCTION
dbo.liczbaUczniow (@klasa
nvarchar(50))
RETURNS int
AS
BEGIN
DECLARE @ile int SELECT
@ile = COUNT(*) FROM
TBL_UCZEN WHERE
KLASA=@klasa
RETURN @ile
END
```

Slajd 32

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Podsumowanie

|                                      |                                                                                         |
|--------------------------------------|-----------------------------------------------------------------------------------------|
| Widok                                | tabela wirtualna określona przez zapytanie SQL-owe                                      |
| Funkcje o wartościach skalarnych     | zwracają jako wynik pojedynczą wartość                                                  |
| Funkcje o wartościach tabelarycznych | są nazywane również sparametryzowanymi widokami, zwracają jako wartość tablicę rekordów |

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

DAILY Lektora 7 – widoki i funkcje

Tabela podsumowująca poznane wyrażenia

### Ćwiczenia

Ćwiczenie zostało przedstawione na slajdach 26-31, zadaniem uczniów jest utworzenie widoków i funkcji na podstawie zdobytej wiedzy.

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili samodzielnie zaprogramować widoki i funkcje w bazie danych.

## Lekcja 8 - Procedury Triggery

### Cel lekcji

Celem lekcji prezentacja dwóch kolejnych typów obiektów baz danych: procedur oraz wyzwalaczy (triggerów). Szkoleniowiec zademonstruje kilka przykładów wykorzystania procedur i triggerów.

## Treść - slajdy z opisem

### Slajd 1

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The main title is 'SQL Structured Query Language' in a large, bold font. Below it, the subtitle reads 'Lekcja 8: procedury i triggerzy'. At the bottom, there are logos for 'KAPITAŁ LUDZKI' (Human Capital) and the European Union, along with the text 'Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego' and the 'DAILY GROUP' logo.

W tej lekcji poznamy kolejne dwa typy obiektów baz danych: procedury składowane i wyzwalacze(triggerzy).

### Slajd 2

The slide features a header with logos for B2E, Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The main title is 'Przypomnienie' (Reminder). Below it, there is a bulleted list of topics covered in the previous lesson: 'Widok', 'Funkcja o wartościach skalarnych', and 'Funkcja o wartościach tabelarycznych'. At the bottom, there is the 'DAILY GROUP' logo and the text 'Lekcja 8 – procedury i triggerzy'.

Podsumowanie poprzedniej lekcji.

### Slajd 3

The slide features a header with logos for B2E, Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The main title is 'Procedura SQL'. Below it, there is a bulleted list of characteristics of stored procedures: 'Procedury składowane to prekompilowane wyrażenia języka SQL przechowywane na serwerze bazodanowym.', 'Mogą być definiowane z parametrami wejściowymi i wyjściowymi.', and 'Stosuje się je do wykonywania powtarzających się logicznie takich samych operacji na (bazie) danych, nie wymagających ingerencji ze strony użytkownika.' At the bottom, there is the 'DAILY GROUP' logo and the text 'Lekcja 8 – procedury i triggerzy'.

Procedura składowana, to nic innego jak specjalna struktura w bazie danych, która przypomina klasyczne programowanie. Każda z tworzonych przez nas procedur, może zawierać parametry i zwracać określone wartości, może również wykonywać pewne operacje.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Procedura SQL

- Składnia ogólna:  
CREATE PROC[EDURE] nazwa\_procedury lista\_parametrów AS instrukcje
- Parametry podajemy zawsze ze znakiem "@", następnie typ danych i rozdzielamy parametry przecinkami.
- Możemy podać wartość domyślną oraz określić, że jest to parametr wyjściowy OUTPUT
- Definicja parametru procedury ma postać:  
@nazwa\_parametru typdanych [=wartość\_domyślna] [OUTPUT]
- ALTER PROCEDURE pozwala powtórnie wprowadzić tekst procedury.

DAILY GROUP Lekcja 8 – procedury i triggerzy

Procedura składowana, to nic innego jak specjalna struktura w bazie danych, która przypomina klasyczne programowanie. Każda z tworzonych przez nas procedur, może zawierać parametry i zwracać określone wartości, może również wykonywać pewne operacje.

Slajd 5

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Procedura SQL

- Zalety używania procedur składowanych:
  - Różne aplikacje korzystające z tej samej bazy danych korzystają z tej samej procedury — mniejsze ryzyko popełnienia błędu.
  - Mniejsze koszty uruchomienia i konserwacji.
  - Z punktu widzenia wydajności
    - procedura wykonywana jest przez wolniejszy język, ale na szybszym serwerze,
    - znaczne zmniejszenie kosztu przesyłu danych.

DAILY GROUP Lekcja 8 – procedury i triggerzy

Zalety stosowania procedur składowanych przedstawiono na slajdzie.

Slajd 6

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language


## Wybrane procedury systemowe

- **Sp\_tables** - tabele bazy danych
- **Sp\_help** nazwa\_obiektu - informacje na temat obiektu (np. tabeli, widoku, procedury)
- **Sp\_helptext** nazwa\_obiektu - tekst obiektu (np. procedury)
- **Sp\_helpdb** nazwa\_bazy - informacje na temat bazy danych
- **Sp\_helpindex** nazwa\_tabeli - indeksy założone na tabeli
- **Sp\_helpconstraint** nazwa\_tabeli - więzy spójności na tabeli
- **Sp\_spaceused** nazwa\_obiektu - ilość miejsca zajętego przez obiekt


DAILY GROUP Lekcja 8 – procedury i triggerzy

Zalety stosowania procedur składowanych przedstawiono na slajdzie.





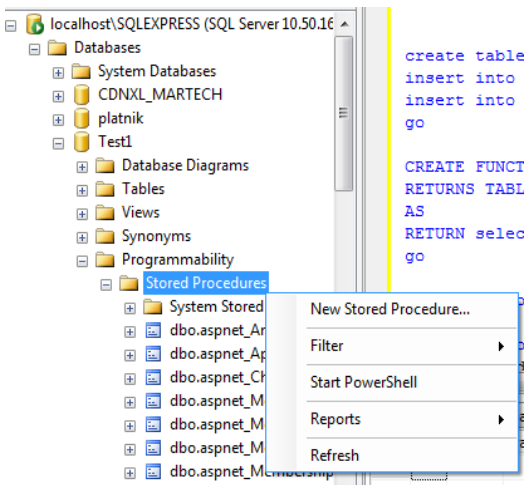
**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Tworzenie procedury w MS SMS





Lekcja 8 – procedury i triggery

Zarządzanie procedurami składowanymi jest bardzo proste, bo możemy to robić przy pomocy Microsoft SQL Server Management Studio. Tam w Object Explorerze znajdują się wszystkie bazy danych, a jeśli rozwinieśmy odpowiednio drzewo, to znajdziemy również gałąź o nazwie Programmability, a w niej Stored Procedures. W tym właśnie miejscu, trzymane są wszystkie procedury przeznaczone dla konkretnej bazy danych. Idąc głębiej, możemy uzyskać dostęp do sporej kolekcji procedur systemowych. Aby utworzyć nową procedurę, wystarczy kliknąć prawym przyciskiem myszy na katalogu Stored procedures i z menu kontekstowego wybrać opcję New stored procedure.



Slajd 8

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

---

## Procedura SQL

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE <Procedure_Name, sysname, ProcedureName>
 <@Param1, sysname, @p1> <Datatype_For_Param1, , int> = <Default_Value_For_Param1, , 0>,
 <@Param2, sysname, @p2> <Datatype_For_Param2, , int> = <Default_Value_For_Param2, , 0>
AS
BEGIN
 SET NOCOUNT ON;
 SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
END
GO

```

Lekcja 8 – procedury i triggerzy

Wybranie opcji „New stored procedure” w MS SMS, spowoduje utworzenie pustego szablonu procedury, który powinien wyglądać mniej więcej tak.

Slajd 9

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

---

## Procedura SQL

- Najprostszy przykład

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE HelloWorld
 @Name varchar(50)
AS
BEGIN
 SELECT 'Witaj ' + @Name + '!'
END
GO

```

Lekcja 8 – procedury i triggerzy

Bardzo prosty przykład procedury składowanej. W szkielecie z poprzedniego slajdu wprowadziliśmy parametr wejściowy @Name typu varchar(50), procedura wykonuje polecenie SELECT i wyświetli Witaj + parametr + wykrzyknik.

Slajd 10

The screenshot shows a SQL execution environment. At the top, there are logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The main title is "Procedura SQL". A bullet point indicates "Wywołanie procedury". Below this, the SQL command `exec HelloWorld 'Tomek'` is shown. The results window displays a single row with the value "Witaj Tomek!". The footer includes the DAILY GROUP logo and the text "Lekcja 8 – procedury i triggerzy".

Wywołanie procedury z poprzedniego slajdu. Procedurę wywołujemy zawsze za pomocą polecenia EXEC (z ang. execute).

Slajd 11

The screenshot shows a SQL script defining a stored procedure. At the top, there are logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The main title is "Procedura SQL". A bullet point indicates "Analiza kodu". The SQL code is as follows:  
`SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE PROCEDURE HelloWorld  
 @Name varchar(50)  
AS  
BEGIN  
 SELECT 'Witaj ' + @Name + '!!'  
END  
GO`  
The footer includes the DAILY GROUP logo and the text "Lekcja 8 – procedury i triggerzy".

Przeanalizujemy teraz poszczególne polecenia w naszej procedurze:

W linii 1 oraz 3, określamy ustawienia Servera SQL. Pierwszy zapis mówi o zasadach porównywania wartości *NULL*, natomiast drugi zapis oznacza sposób używania cudzysłowów.

Dopiero w linii 5, zaczynam właściwy kod procedury. Aby stworzyć procedurę, wykorzystujemy standardowe polecenie *CREATE*. Następnie, mówimy jaki obiekt chcemy utworzyć i jaką nazwę ma przyjąć. Kolejnym krokiem, będzie określenie parametrów. Choć są one opcjonalne, jednak mogą okazać się bardzo przydatne. Np. w tym przypadku określamy imię użytkownika. Każdy parametr musi zaczynać się od znaku @. Musimy podać także jego typ oraz ewentualnie długość. Parametry (jeśli jest ich więcej niż jeden), rozdzielamy przecinkiem.

Po utworzeniu nagłówka procedury, w linii 7, stosujemy słowo kluczowe *AS*, które pozwala nam na



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

wprowadzenie właściwego ciała procedury. Treść procedury, umieszczamy pomiędzy kolejnymi słowami kluczowymi - *BEGIN* oraz *END*.

Kluczowym poleceniem naszej procedury, jest zawarte w linii 10 „SELECT 'Witaj ' + @Name + '!'”, która wykonuje *SELECT* z naszym imieniem. Procedura ta nie odwołuje się do żadnej tabeli z bazy danych, jedynie wyświetla powitanie na podstawie podanego parametru.

Slajd 12



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Procedury mogą uzyskać dostęp do tabel

```

CREATE PROCEDURE WyzerujPremie
 @numer_pracownika int
AS
BEGIN
 UPDATE pracownicy SET premia=0 WHERE nr_pracownika = @numer_pracownika
END
GO

```

```
exec WyzerujPremie 1|
```

Messages

(1 row(s) affected)



Lekcja 8 – procedury i triggerzy

Procedury składowane w SQL mają dostęp do tabel w bazie danych. Na powyższym przykładzie zademonstrowano procedurę składowaną ustawiającą premię pracownika na 0. Jeśli operację zerowania premii przeprowadzamy często – warto stworzyć procedurę, gdzie podając numer pracownika kasujemy premię. Dzięki temu nie musimy każdorazowo używać odpowiedniego polecenia UPDATE.

Poniżej przykład wywołania procedury wraz z odpowiedzią serwera – jeden wiersz został zaktualizowany.

Slajd 13

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Podstawowe instrukcje

- RETURN w procedurze

```
CREATE PROCEDURE Ilu_pracownikow AS
BEGIN
DECLARE @ile INT;
SELECT @ile = COUNT(*) FROM pracownicy;
RETURN @ile;
END;
```

DAILY GROUP Lekcja 8 – procedury i triggerzy

Procedura zliczy wszystkie wiersze z tabeli pracownicy i zwróci wartość poprzez RETURN. Stosuje się ją zwykle do przekazania stanu obliczeń wywołania procedury, np. czy i jaki wystąpił błąd.

Slajd 14

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Podstawowe instrukcje

- PRINT

```
DECLARE @Imie VARCHAR(9), @Nazwisko VARCHAR(20)
SELECT @nazwisko = Nazwisko, @imie=imie FROM pracownicy
WHERE nr_pracownika=1

PRINT 'Imię='+@imie+', Nazwisko='+@Nazwisko
```

Messages  
Imię=Jan, Nazwisko=Kowalski

DAILY GROUP Lekcja 8 – procedury i triggerzy

Do debuggowania (śledzenia poszczególnych kroków fragmentu kodu), najczęściej używa się instrukcji PRINT, która w konsoli umożliwia wyświetlenie (wydruk) zmiennej lub ciągu znaków.

Slajd 15

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY SQL Structured Query Language

## Podstawowe instrukcje

- Instrukcja warunkowa

IF warunek  
instrukcja lub blok  
[ELSE  
instrukcja lub blok]

```
IF @pensja > 0
BEGIN
INSERT INTO pracownicy (nazwisko, placa_zasadnicza, nr_miejsca)
VALUES (@nazwisko, @pensja, @miejsce);
PRINT 'Wstawiono pracownika';
END
ELSE
PRINT 'Niepoprawna pensja';
```

DAILY GROUP Lekcja 8 – procedury i triggerzy

Przykład warunku IF-ELSE. Jeśli pensja jest większa niż zero – wykonywany jest blok instrukcji zawarty pomiędzy BEGIN oraz END, w przeciwnym razie (ELSE) wykonywany jest PRINT.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 16

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Podstawowe instrukcje

- Instrukcja iteracji

WHILE warunek  
instrukcja lub blok

```
WHILE (SELECT AVG(placa_zasadnicza) FROM pracownicy) < 2000
BEGIN
UPDATE pracownicy SET placa_zasadnicza = placa_zasadnicza * 1.2;
IF (SELECT MAX(placa_zasadnicza) FROM pracownicy) > 4000 BREAK;
ELSE CONTINUE;
END;
SELECT * FROM pracownicy;
```

DAILY GROUP Lekcja 8 – procedury i triggerzy

Przykład warunku iteracji - WHILE. Istnieje możliwość wyjścia z pętli używając instrukcji BREAK. Instrukcja CONTINUE powoduje, że reszta instrukcji w pętli jest ignorowana i następuje wykonanie kolejnej iteracji.

Slajd 17

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### SELECT w procedurze

- Procedura oprócz parametrów wejściowych i wyjściowych może wykonywać jedno lub kilka zapytań SELECT

```
CREATE PROC PokazTabele AS
BEGIN
SELECT * FROM pracownicy;
SELECT * FROM klienci;
END;
GO
EXEC PokazTabele;
```

DAILY GROUP Lekcja 8 – procedury i triggerzy

Przykład procedury wywołującej dwa zapytania typu SELECT. Rezultatem działania tej procedury będzie wyświetlenie zawartości obu tabel: klienci i pracownicy.

Slajd 18

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

### Triggerzy (wyzwalacze)

- Predefiniowane operacje na danych wyzwalane zdarzeniami.
- Definiowane przez użytkownika i uruchamiane automatycznie.
- Inne nazwy: wyzwalacze, procedury wyzwalane.

DAILY GROUP Lekcja 8 – procedury i triggerzy

Definicja triggerów

Slajd 19

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Triggery (wyzwalacze)

- Triggery uruchamiane są zdarzeniami:
  - INSERT
  - UPDATE
  - DELETE
- Triggery mogą być uruchamiane:
  - INSTEAD OF
  - AFTER
- Triggery mogą być uruchamiane:
  - FOR EACH ROW
  - FOR EACH STATEMENT

**DAILY GROUP** Lekcja 8 – procedury i triggery

Wyzwalacze uruchamiane są zdarzeniami: INSERT, UPDATE, DELETE.

Wyzwalacze mogą być uruchamiane: INSTEAD OF, AFTER, czyli odpowiednio zamiast lub po modyfikacji danych.

Wyzwalacze mogą być uruchamiane: FOR EACH ROW, FOR EACH STATEMENT, czyli dla każdego wiersza, którego dotyczy działanie lub raz dla całej instrukcji.

Slajd 20

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Triggery (wyzwalacze)

- Składnia:  
CREATE TRIGGER nazwa\_wyzwalacza  
ON tabela  
FOR [INSERT|UPDATE|DELETE]  
AS instrukcje
- Po słowie FOR (równoważnie AFTER) możemy napisać jedną, dwie lub wszystkie trzy nazwy (oddzielone przecinkami).
- ALTER TRIGGER pozwala powtórnie wprowadzić zmodyfikowany tekst wyzwalacza.

**DAILY GROUP** Lekcja 8 – procedury i triggery

Składnia wyzwalacza zawiera nazwę triggera, tabelę, na której trigger będzie działał oraz warunek wystąpienia:

INSERT – trigger zostanie uruchomiony w czasie wstawiania danych do tabeli

UPDATE – trigger zostanie uruchomiony w czasie aktualizacji danych w tabeli

DELETE – trigger zostanie uruchomiony w czasie usuwania danych z tabeli



Slajd 21

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Triggery – przykład

- Uniemożliwienie usuwania danych
 

```
CREATE TRIGGER wyzw1 ON pracownicy
FOR DELETE AS
BEGIN PRINT 'Nie możesz usuwać danych z tej tabeli';
SELECT * FROM pracownicy;
ROLLBACK;
END;
```
- Próba usunięcia
 

```
DELETE FROM pracownicy;
```

Results Messages

Nie możesz usuwać danych z tej tabeli

(0 row(s) affected)

Msg 3609, Level 16, State 1, Line 1  
The transaction ended in the trigger. The batch has been aborted.

Lekcja 8 – procedury i triggery

Przykład demonstrujący utworzenie triggera, który zabezpiecza nas przed omyłkowym usunięciem danych z tabeli pracownicy. Po zdefiniowaniu triggera, próbujemy usunąć dane.

Slajd 22

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Usuwanie triggerów i procedur

- Usuwanie triggerów
 


```
DROP TRIGGER nazwa_triggera
```
- Usuwanie procedur składowanych
 


```
DROP PROCEDURE nazwa_procedury
DROP PROC nazwa_procedury
```

Lekcja 8 – procedury i triggery

Usuwanie obiektów typu trigger i procedura składowana. Proszę zwrócić uwagę, że SQL (T-SQL) wyrażenia PROC i PROCEDURE są równoważne.







**SQL**  
Structured Query Language

## Ćwiczenia

### Pracownicy


| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia | pesel       | stanowisko          | nr_miejsca | przelozony |
|---------------|-------------|-------------|------------------|--------|-------------|---------------------|------------|------------|
| 1             | Kowalski    | Jan         | 2300,00          | 250,00 | 82091104357 | Manager             | 1          | 7          |
| 2             | Nowak       | Karol       | 2700,00          | 100,00 | 80010123987 | Specjalista         | 1          | 1          |
| 3             | Przepiórka  | Marzena     | 2700,00          | NULL   | 89121203456 | Specjalista         | NULL       | 1          |
| 4             | Burzych     | Paweł       | 1900,00          | 500,00 | 78032309123 | Specjalista         | 2          | 7          |
| 5             | Makłowicz   | Marek       | 2000,00          | NULL   | 54013112345 | Specjalista         | 2          | 7          |
| 6             | Naramowicka | Magdalena   | 2100,00          | 200,00 | 77121312098 | Specjalista         | 1          | 1          |
| 7             | Witos       | Jacek       | 3000,00          | 500,00 | 69100967234 | Manager             | 3          | 7          |
| 8             | Markowski   | Maksymilian | 2000,00          | NULL   | 67121209878 | Młodszy specjalista | NULL       | 7          |

### Klienci

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Mikołaj  | Wiśniewski | 82040112389 |
| 2          | Marcin   | Kaniewski  | 73122990123 |

### Miejsca

| nr_miejsca | ulica       | numer | miasto   | kod    | telefon        |
|------------|-------------|-------|----------|--------|----------------|
| 1          | Mała        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2          | Wąska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3          | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |



Lekcja 8 – procedury i triggerzy

**Ćwiczenie 1**

Trigger uniemożliwiający usuwanie danych z tabeli miejsca.

**Ćwiczenie 2**

Utwórz trigger, który w utworzy kopię danych kasowanych z tabeli klienci w tabeli klienci\_kopia z identyczną strukturą jak klienci i dodatkową kolumną data usunięcia.

**Ćwiczenie 3**

Utwórz trigger, który w czasie wstawiania nowych osób do tabeli pracownicy, automatycznie doda ich do tabeli klienci.

**Ćwiczenie 4**

Napisz procedurę składowaną, która wylicza planowany budżet na pensje i premie na najbliższe 12 miesięcy.


**Ćwiczenie 5**


Napisz procedurę składowaną, do wykorzystania przez aplikację drukującą koperty, która zwraca w tabeli imię, nazwisko i adres zatrudnienia pracownika w jednej kolumnie.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 24

 **B2E**  
BUSINESS TO EDUCATION

 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

### Ćwiczenia

- Wywołaj procedurę OBLICZ

 **DAILY**  
GROUP

Lekcja 8 – procedury i triggerzy

EXEC OBLICZ

Slajd 25

 **B2E**  
BUSINESS TO EDUCATION

 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

### Ćwiczenia

- Napisz procedurę, która zwraca liczbę z parametru podniesioną do kwadratu

 **DAILY**  
GROUP

Lekcja 8 – procedury i triggerzy

CREATE PROCEDURE potega  
@liczba int

AS

BEGIN

SELECT @liczba \*  
@liczba

END

Slajd 26

 **B2E**  
BUSINESS TO EDUCATION

 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

### Ćwiczenia

- Usuń procedurę z poprzedniego ćwiczenia

 **DAILY**  
GROUP

Lekcja 8 – procedury i triggerzy

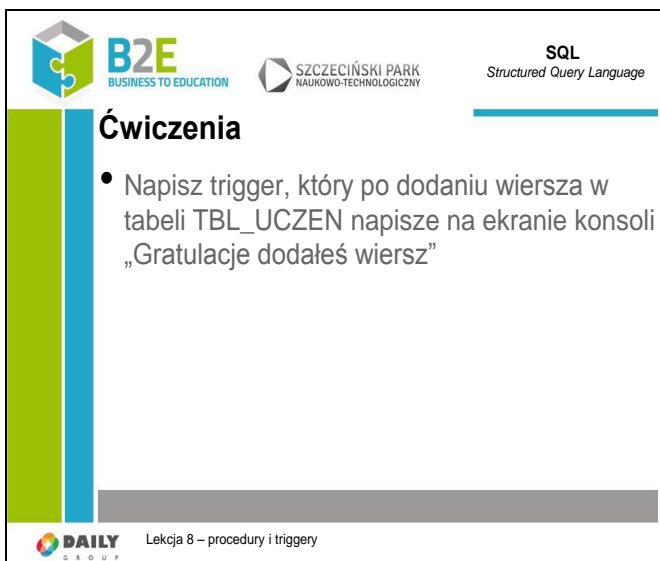
DROP PROCEDURE potega

Slajd 27

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to write a trigger that displays the content of a deleted row from the TBL\_UCZEN table. The footer includes the DAILY GROUP logo and the text 'Lekcja 8 – procedury i triggerzy'.

```
CREATE TRIGGER ttbl_uczen
ON tbl_uczen
AFTER DELETE
AS
BEGIN
SELECT * FROM DELETED
END
```

Slajd 28

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to write a trigger that prints 'Gratulacje dodałeś wiersz' (Congratulations, you added a row) to the console when a new row is added to the TBL\_UCZEN table. The footer includes the DAILY GROUP logo and the text 'Lekcja 8 – procedury i triggerzy'.

```
CREATE TRIGGER ttbl_uczen1
ON tbl_uczen
AFTER INSERT
AS
BEGIN
PRINT „Gratulacje dodałeś
wiersz”
END
```



Slajd 29



**B2E**  
BUSINESS TO EDUCATION



**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Podsumowanie

|                      |                                                                           |
|----------------------|---------------------------------------------------------------------------|
| Procedura składowana | Prekompilowane wyrażenia języka SQL przechowywane na serwerze bazodanowym |
| Wyzwalacz            | Predefiniowane operacje na danych wyzwalane zdarzeniami                   |
| Warunek IF           | Instrukcja warunkowa wykorzystywana w funkcjach, procedurach i triggerach |
| Pętla WHILE          | Instrukcja iteracji wykorzystywana w funkcjach, procedurach i triggerach  |



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI

Człowiek - najlepsza inwestycja



UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



Lekcja 8 – procedury i triggerery

Tabela podsumowująca poznane wyrażenia

### Ćwiczenia

Ćwiczenie zostało przedstawione na slajdach 23-28. Uczniowie wykorzystają zdobytą wiedzę przy tworzeniu procedur i triggerów.

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili samodzielnie implementować procedury i triggerery w języku TSQL.

## Lekcja 9 - Indeksy, constrainty

### Cel lekcji

Celem lekcji jest poznanie struktury optymalizującej bazę danych jaką jest indeks. W drugiej części lekcji uczniowie poznają ograniczenia bazy – constraint, służące zachowaniu integralności danych.

## Treść - slajdy z opisem

Slajd 1

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The main title is 'SQL Structured Query Language' in a large, bold font. Below it, the subtitle reads 'Lekcja 9: indeksy, constrainty'. At the bottom, there are logos for 'KAPITAŁ LUDZKI' (Human Capital) and 'UNA EUROPEJSKA' (European Union), along with the text 'Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego'. The footer includes the 'DAILY GROUP' logo and 'SQL Structured Query Language Lekcja 9'.

W trakcie tej lekcji poznamy strukturę optymalizującą bazę danych jaką jest indeks. W drugiej części lekcji poznamy ograniczenie bazy – constraint, służące zachowaniu integralności danych.

Slajd 2

The slide has the same header as slide 1. The main heading is 'Przypomnienie' (Reminder). Below it is a bulleted list of SQL constructs: 'Procedura składowana' (Stored procedure), 'Wyzwalacz' (Trigger), 'Warunek IF' (IF condition), and 'Pętla WHILE' (WHILE loop). The footer contains the 'DAILY GROUP' logo and 'Lekcja 9 – indeksy, constrainty'.

Przypomnienie pojęć z poprzednich zajęć

Slajd 3

The slide features the same header. The main heading is 'Indeksy' (Indexes). The text explains: 'Indeks jest specjalną strukturą danych wprowadzoną w celu zwiększenia prędkości wykonywania operacji na tabeli. Indeks w bazie danych jest odpowiednikiem spisu treści w książce.' (An index is a special data structure introduced to increase the speed of operations on tables. An index in a database is the equivalent of a table of contents in a book.) The footer includes the 'DAILY GROUP' logo and 'Lekcja 9 – indeksy, constrainty'.

Po co kartkować całą książkę dla znalezienia jednej interesującej nas informacji, jeśli możemy zajrzeć do spisu treści i na jego podstawie odnaleźć stronę, na której znajduje się to, czego szukamy. Zaoszczędzimy w ten sposób cenny czas choćby dlatego, że spis treści jest zwykle zorganizowany w sposób alfabetyczny, co znacznie upraszcza wyszukanie frazy, która nas interesuje. Indeks w bazie danych wykorzystuje się przy



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

zapytaniach typu DQL (SELECT), które mają na celu wyszukiwanie odpowiednich wartości w bazie danych. Podczas realizowania zapytania optymalizator (Serwer SQL) najpierw przeszukuje indeks, który jest uporządkowany, a następnie na podstawie indeksu odczytuje odpowiednie rekordy. Indeks posiada strukturę logiczną i fizyczną niezależną od tabeli, do jakiej się odwołuje. Posiada również własną przestrzeń dyskową oraz jest automatycznie utrzymywany przez system zarządzania bazą danych.

Slajd 4



**Indeksy**

- Tworzenie indeksów

```
CREATE INDEX nazwa_indeksu ON tabela (nazwa_indeksu);
```

- lub

```
ALTER TABLE nazwa_tabeli ADD INDEX (nazwa_indeksu);
```

Lekcja 9 – indeksy, constrainty

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

SQL  
Structured Query Language

## Indeksy – do czego służą?

- Zwiększają wydajność
- Zapisują gdzie znajduje się dana wartość
- Umożliwiają wyszukiwanie w tabelach bez potrzeby sprawdzania zawartości od początku do końca

**DAILY** GROUP Lekcja 9 – indeksy, constrainty

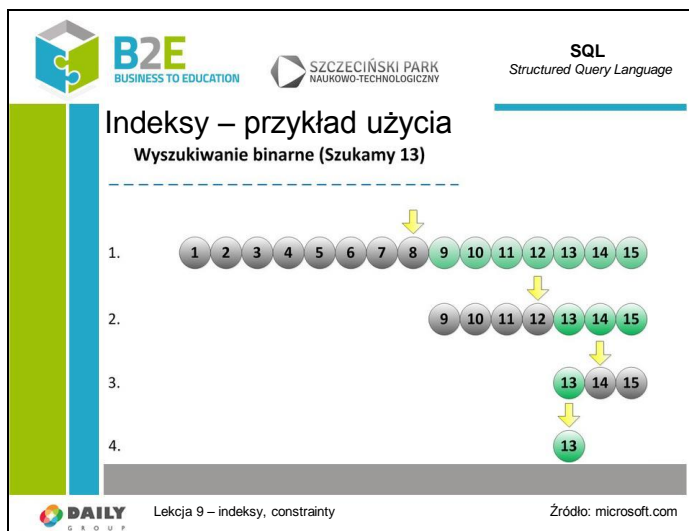
Bez indeksów SQL serwer oferuje nam dostęp do danych – są one dostępne. Jednak przy większych zbiorach (niż nasze tabele klienci, miejsca, pracownicy), musimy zwrócić uwagę na wydajność bazy danych. W tym momencie powinniśmy zastanowić się nad indeksami, gdyż dzięki nim szybciej odnajdujemy dane w tabeli. Podczas wykonywania polecenia typu DQL (SELECT) serwer bazy danych musi wykonać bardzo dużo operacji – wybieranie danych, sortowanie wyników itd. SQL Server może pracować z bardzo dużą ilością rekordów, a często z dziesiątkami milionów rekordów w tabeli, dlatego niezwykle ważną rolę odgrywa w nim optymalizacja wszelkiego rodzaju wyszukiwania oraz sortowania.

Rekordy ułożone są w tabeli w takiej kolejności, w jakiej zostały dodane\*, co oznacza, że jeśli spróbujemy wyszukać w bazie danych np. sklepu ceny danego produktu, to SQL Server musi za każdym razem na nowo przetrząsać od początku do końca wszystko, co tam się znajduje. Kiedy produktów będzie parę tysięcy, może to potrwać dosyć długo. W takiej sytuacji pomocą służą nam indeksy. Indeks nałożony na pole jest niczym innym jak kopią jego zawartości, tyle że posortowaną i odpowiednio ułożoną.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

\*Taka sytuacja występuje domyślnie, w przypadku gdy PK jest indeksem klastrowanym. Można to zmienić wskazując inny indeks klastrowany niż PK.

Slajd 6



Mamy tysiąc wartości posortowanych rosnąco, a nas interesują rekordy z zakresu od 600 do 700. Jeśli rekordy są posortowane rosnąco, SQL Server może zacząć od środkowego rekordu tego zbioru i sprawdzić, czy znaleziona wartość jest mniejsza, czy większa od podanego zakresu. Naturalnie, jeśli wartość środkowego elementu jest mniejsza od podanego zakresu, przeszukuje tylko późniejsze rekordy (również przez wyszukiwanie połówkowe w nowym zbiorze, stanowiącym połowę zbioru poprzedniego), jeśli wartość tego elementu jest większa od podanego zakresu, SQL Server szuka jedynie w rekordach wcześniejszych. Jeśli z kolei poszukiwana wartość mieści się w podanym zakresie, porusza się w obu kierunkach, dopóki nie wypadnie poza założony zakres. W sytuacji, gdy nie byłoby posortowanej tablicy, musiałby najpierw albo ją posortować, albo przejrzeć wszystkie wartości w celu wyodrębnienia tych, które go interesują. Slajd przedstawia przykład binarnego wyszukiwania elementu numer 13 w zbiorze



uporządkowanym 15-  
elementowym.

Dzięki posortowaniu rekordów SQL Server na wstępie odrzucił całe mnóstwo rekordów, które nie pasują do naszego zapytania. Dodatkowo mamy zrealizowaną automatycznie funkcję sortowania, bez uruchamiania jakiegokolwiek funkcji sortującej.

Źródło: microsoft.com

Slajd 7

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Indeksy – kiedy stosować

- Indeks optymalizuje zapytania = stosować wszędzie?
- Nie:
  - Indeks zajmuje miejsce w bazie danych
  - Indeks optymalizuje odczyt, pogarsza czas zapisu
- Stosujemy indeksy na polach, które najczęściej wykorzystujemy w warunku WHERE
- EXPLAIN – ułatwia podejmowanie decyzji gdzie stosować indeks

**DAILY**  
GROUP

Lekcja 9 – indeksy, constrainty

Niestety, nie możemy stosować indeksów wszędzie gdzie tylko możliwe. To dlatego, że ceną za zwiększenia wydajności, jakie oferuje nam indeks, jest zwiększenie rozmiarów bazy, gdyż indeks również potrzebuje trochę miejsca. Przestrzeń dyskowa jest z kolei kluczowym aspektem podczas projektowania bazy danych. A zatem już na początku została obalona postawiona teza o popieraniu rozrzutności podczas indeksowania danych. Indeksy zazwyczaj zajmują tyle samo miejsca co dane i jest to sytuacja zupełnie naturalna. Łatwiej jest zaakceptować wydatki na przestrzeń dyskową niż akceptować czas oczekiwania na zapytanie zwiększony z sekund do kilku godzin.

Należy się zastanowić, na których polach indeks będzie nam potrzebny, a na których jest zwyczajnie zbędny. W tym momencie pomocne będzie środowisko SQL Server, które udzieli nam diagnostycznej informacji o wykonywaniu zapytania, na podstawie której przekonamy



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego  
się, czy wykorzystał on nasze indeksy. Wystarczy zapytanie SELECT poprzedzić słowem EXPLAIN, a system nam podpowie.

O wyborze indeksu decyduje optymalizator kwerend określający, które (ewentualnie) indeksy będą najbardziej użyteczne.

Warto również w tym punkcie określić, czym jest selektywność indeksu. Jest to nic innego, jak parametr określający, czy indeks na określonych kolumnach może być przydatny. Wartość tego parametru wyliczamy ze wzoru:

$$S = U/W$$

gdzie:

- S – selektywność;
- U – liczba unikalnych wartości dla kolumny;  
W – liczba wszystkich wierszy w kolumnie.

Jeśli wartość selektywności indeksu wynosi mniej niż 85%, SQL Server raczej nie będzie z niego korzystał.

Indeksy, z wyjątkiem grupujących, w pewnych sytuacjach mogą wydłużać czas operacji wstawiania i modyfikowania danych.

Indeks grupujący powinien być tworzony dla kolumny, według której użytkownicy często sortują dane odczytywane z tabeli lub dla kolumn przechowujących wartości, na podstawie których zwracane są zbiory danych.



## Indeksy – podział

- Ze wzgl. na liczbę wskazań indeksu :

- Indeks gęsty
- Indeks rzadki, przykład:



punktu widzenia liczby wskazań indeksu do pliku danych rozróżnia się:

- Indeks gęsty (dense) – zawiera wpis dla każdej wartości klucza wyszukiwania, czyli dla każdego rekordu.
- Indeks rzadki (sparse) – posiada wpis jedynie dla niektórych wartości wyszukiwania (np. bloków).



Slajd 9



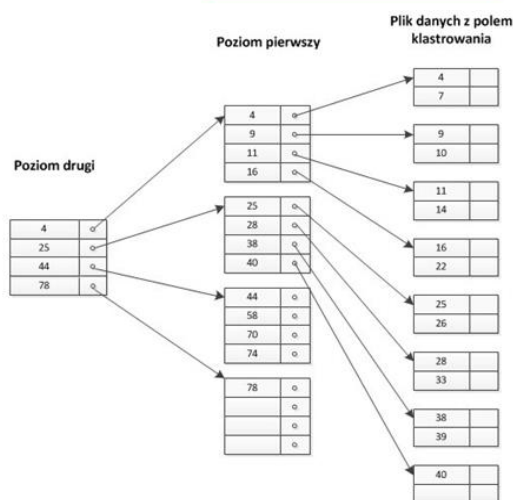
**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Indeksy – podział

- Ze względu na liczbę poziomów
  - Indeksy jednopoziomowe
  - Indeksy wielopoziomowe



Lekcja 9 – indeksy, constrainty

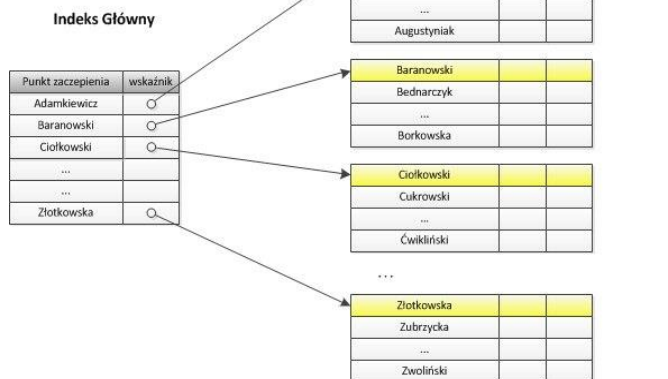
Na slajdzie przykład indeksu wielopoziomowego.

Indeksy wielopoziomowe – dla pierwszego poziomu tworzymy indeks podstawowy i nazywamy go indeksem drugiego poziomu. Analogicznie dla poziomu drugiego, gdzie tworzy się indeks poziomu trzeciego.



## Indeks główny

- Primary key (PK)



**Indeks główny** (Primary index) – zwany także podstawowym, jest założony na kluczu podstawowym pliku uporządkowanego i zawiera jeden klucz dla każdego bloku dyskowego. Pierwszy z rekordów danego bloku nazywamy rekordem zaczepienia lub rekordem kotwiczącym. Należy on do grupy indeksów rzadkich.

PK jest indeksem klastrowanym (domyślnie). Wskazanie innego indeksu klastrowanego (można jeden) powoduje zmianę uporządkowania struktury pliku zgodnie ze wskazanym indeksem klastrowanym.



Slajd 11

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Indeks zgrupowany

- Clustered index

**Plik indeksu**

| Wartość pola klastrowania | wskaźnik |
|---------------------------|----------|
| 1                         | ○        |
| 2                         | ○        |
| 3                         | ○        |
| ...                       |          |
| 6                         | ○        |

**Plik danych z polem klastrowania**

|     |  |  |
|-----|--|--|
| 1   |  |  |
| 1   |  |  |
| 2   |  |  |
| 2   |  |  |
|     |  |  |
| 2   |  |  |
| 3   |  |  |
| 3   |  |  |
| 3   |  |  |
|     |  |  |
| 3   |  |  |
| 4   |  |  |
| 4   |  |  |
| 4   |  |  |
|     |  |  |
| 5   |  |  |
| 5   |  |  |
| ... |  |  |
| 6   |  |  |

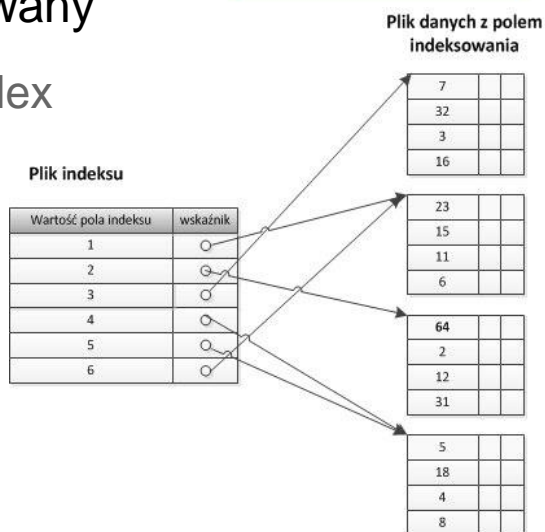
Lekcja 9 – indeksy, constrainty

**Indeks zgrupowany** (Clustered index) – jest założony na atrybucie niebędącym kluczem podstawowym pliku uporządkowanego (nieunikatowym) porządkującym pliku uporządkowanego. Indeks zawiera jeden klucz dla każdej wartości atrybutu. Posiada dwa pola – pierwsze ma ten sam typ co pole klastrowania, drugie – wskaźnik. Indeks zawiera wpis do każdej odrębnej wartości klastrowania oraz wskaźnik na pierwszy blok, do którego ona należy. Należy do grupy indeksów rzadkich.



## Indeks niezgrupowany

- Nunclustered index





**Indeks niezgrupowany** (Nunclustered index) – jest zakładany na pole, które ma unikatowe wartości w każdym rekordzie lub które nie jest polem klucza i posiada powtarzające się wartości. Plik indeksu niezgrupowanego jest uporządkowany i posiada dwa pola – jedno jest tego typu co wybrane pole niebędące polem uporządkowania pliku (pole indeksujące), drugie – wskaźnikiem na blok lub rekord. Dla jednego pliku może być wiele indeksów drugorzędnych. Należy do grupy indeksów zagęszczonych. Wskaźniki we wpisach indeksu są wskaźnikami na bloki.

Jest on zakładany na atrybucie indeksowym pliku danych, który nie jest atrybutem porządkującym tego pliku. Każdy rekord pliku danych posiada swój odpowiednik w rekordzie indeksu. Stąd indeks wtórny jest indeksem gęstym. Rekord indeksu wtórnego składa się z dwóch pól – wartości pola indeksowego i wskaźnika albo do rekordu albo do bloku danych zawierającego ten rekord.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


Slajd 13

  **SQL**  
Structured Query Language



## Indeksy – przykłady

- Tworzenie indeksów przy tworzeniu tabeli:

```
CREATE TABLE nazwa_tabeli (
 kolumna1 INT,
 kolumna2 INT,
 UNIQUE indeks_unikalny (kolumna1),
 INDEX indeks_zwykly (kolumna2)|
)
```

 Lekcja 9 – indeksy, constrainty


Slajd 14

  **SQL**  
Structured Query Language

## Indeksy – przykłady

- Tworzenie indeksów w istniejącej tabeli:

```
ALTER TABLE nazwa_tabeli
ADD UNIQUE indeks_unikalny (kolumna1),
ADD INDEX indeks_zwykly (kolumna2)
```

 Lekcja 9 – indeksy, constrainty

Slajd 15

  **SQL**  
Structured Query Language

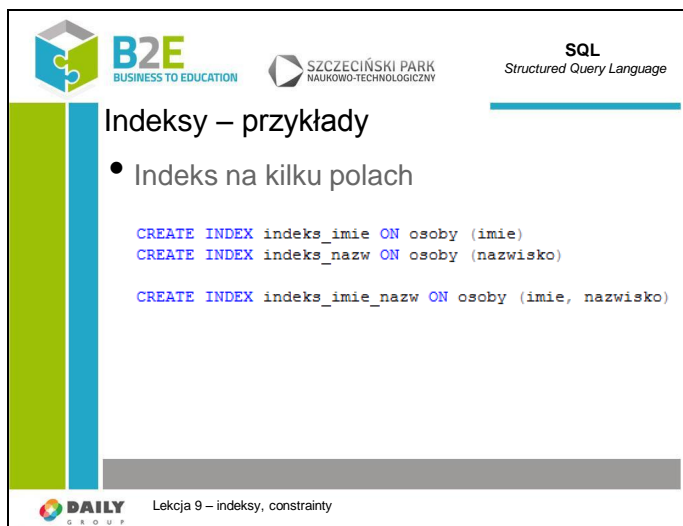
## Indeksy – przykłady

- Możemy też utworzyć sam indeks za pomocą CREATE INDEX

```
CREATE UNIQUE INDEX indeks_unikalny ON nazwa_tabeli (kolumna1)
CREATE INDEX indeks_zwykly ON nazwa_tabeli (kolumna2)
```

 Lekcja 9 – indeksy, constrainty





**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Indeksy – przykłady

- Indeks na kilku polach

```
CREATE INDEX indeks_imie ON osoby (imie)
CREATE INDEX indeks_nazw ON osoby (nazwisko)

CREATE INDEX indeks_imie_nazw ON osoby (imie, nazwisko)
```

**DAILY GROUP** Lekcja 9 – indeksy, constrainty

Dwa powyższe przykłady mogą wydawać się podobne, jednak występuje pomiędzy nimi zasadnicza różnica. Rozważmy, jak zostanie zinterpretowany przez system SQL Server poniższy przykład:

```
SELECT * FROM osoby
WHERE imie='Jan' AND
nazwisko='Kowalski'
```

Jeśli tabela posiada dwa różne indeksy, każdy na pojedynczej kolumnie, baza danych wykona to zapytanie w następujących krokach:

- wyszuka wszystkie rekordy, gdzie występuje imie = Jan;
- wyszuka wszystkie rekordy, gdzie występuje nazwisko = Kowalski;
- obliczy część wspólną zbiorów rekordów z pierwszego i drugiego kroku, i zwróci ją jako wynik zapytania.

W drugim przypadku, tzn. kiedy indeks jest nałożony na kolumnach (imię, nazwisko), baza danych może wyszukać potrzebne dane w jednym kroku. Sprawdzi równocześnie wartości w polach „imię” i „nazwisko”.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 17



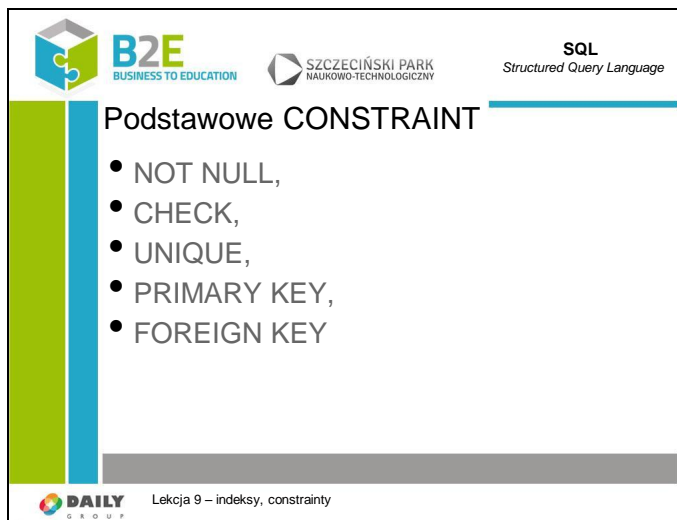
SQL  
Structured Query Language

## Klauzula CONSTRAINT

- o CONSTRAINT = więź, ograniczenie
- o Ograniczenie podobne do indeksu, chociaż może również służyć do ustanowienia relacji z inną tabelą.
- o Za pomocą klauzuli CONSTRAINT w instrukcjach ALTER TABLE i CREATE TABLE można tworzyć i usuwać ograniczenia.
- o Istnieją dwa typy klauzul CONSTRAINT:
  - ograniczenie dla jednego pola,
  - ograniczenie dla większej liczby pól.

DAILY GROUP Lekcja 9 – indeksy, constrainty

Slajd 18



SQL  
Structured Query Language


## Podstawowe CONSTRAINT


- NOT NULL,
- CHECK,
- UNIQUE,
- PRIMARY KEY,
- FOREIGN KEY

DAILY GROUP Lekcja 9 – indeksy, constrainty

W trakcie wcześniejszych lekcji omówiliśmy już NULL/NOT NULL oraz PRIMARY KEY, na kolejnych slajdach przyjrzymy się pozostałym.

Slajd 19

 **B2E**  
BUSINESS TO EDUCATION

 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language


## CHECK

- Umożliwia sprawdzenie (walidację) wartości przed zapisem do tabeli w bazie danych, np.

```
ALTER TABLE dbo.pracownicy WITH CHECK
ADD CONSTRAINT CK_placa CHECK ((placa_zasadnicza>0));
```

```
INSERT INTO dbo.pracownicy(nr_pracownika, placa_zasadnicza) VALUES (10,0);
```


Message  
Msg 547, Level 16, State 2, Line 1  
The INSERT statement conflicted with the CHECK constraint "CK\_placa". The conflict occurred in database "Dev11", table "dbo.pracownicy", column "placa\_zasadnicza". The statement has been terminated.


 **DAILY**  
G R O U P      Lekcja 9 – indeksy, constrainty

Wykonanie operacji z przykładowo uniemożliwi wstawianie nowych wierszy do tabeli pracownicy, lub modyfikację istniejących zmieniających płacę zasadniczą na wartość mniejszą lub równą 0.

Uwaga – w warunku można użyć także funkcję użytkownika lub funkcję systemową!

Slajd 20

 **B2E**  
BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## UNIQUE

- Umożliwia sprawdzenie (walidację) wartości przed zapisem do tabeli w bazie danych pod kątem niepowtarzalności

```
ALTER TABLE pracownicy
ADD UNIQUE (pesel)
```

 **DAILY**  
G R O U P      Lekcja 9 – indeksy, constrainty

Wykonanie operacji z przykładowo uniemożliwi wstawianie nowych wierszy do tabeli pracownicy, lub modyfikację istniejących zmieniających płacę zasadniczą na wartość mniejszą lub równą 0.



Slajd 21

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## FOREGIN KEY

- Klucz obcy - dodatkowa kolumna lub zbiór kolumn w danej tabeli z wartościami, stanowiącymi klucz główny w innej tabeli

Lekcja 9 – indeksy, constrainty

Przykład tabeli pracownicy, gdzie nr\_miejsca jest kluczem obcym do kolumny nr\_miejsca w tabeli miejsca.

Slajd 22

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## FOREGIN KEY - tworzenie

```

CREATE TABLE Nazwa_tabeli
(
 kolumna1 TYP,
 =
 kolumnaN TYP,

 CONSTRAINT nazwa_klucza_obcego
 FOREIGN KEY (nazwa_kolumny_z_tworzonej_tabeli)
 REFERENCES tabela_referencyjna (kolumna_tabeli_referencyjnej)
)

```

Lekcja 9 – indeksy, constrainty

T-SQL pozwala na oznaczenie danej kolumny, bądź zbioru kolumn, jako klucza obcego. Jedną z metod jest zdefiniowanie tzw. funkcji CONSTRAINT podczas tworzenia tabeli. Składnia przykładowego zapytania została przedstawiona na slajdzie.

Jeżeli zapytanie zostanie wykonane bezbłędnie, w SQL Server Management Studio, w lokalizacji Object Explorer->Baza Danych->Nazwa Tabeli->Keys, utworzony klucz obcy powinien funkcjonować włącznie z podaną nazwą.






**SQL**  
Structured Query Language

## FOREIGN KEY

- Najważniejsze informacje
  - Klucz obcy stanowi kolumnę, bądź zbiór kolumn, będących kluczem głównym w innej tabeli.
  - Łączenie tabel może odbywać się poprzez zapytanie z wykorzystaniem INNER/OUTER JOIN.
  - Klucze obce tworzy się w T-SQL, dodając odpowiednie wartości w klauzuli CONSTRAINT FOREIGN KEY.

**DAILY GROUP** Lekcja 9 – indeksy, constrainty

Podsumowanie informacji o kluczach obcych.

**SQL**  
Structured Query Language

## Ćwiczenia

**Pracownicy**

| nr_pracownika | nazwisko    | imie        | placa_zasadnicza | premia | pesel       | stanowisko          | nr_miejsca | przełożony |
|---------------|-------------|-------------|------------------|--------|-------------|---------------------|------------|------------|
| 1             | Kowalski    | Jan         | 2300,00          | 250,00 | 82091104357 | Manager             | 1          | 7          |
| 2             | Nowak       | Karol       | 2700,00          | 100,00 | 80010123987 | Specjalista         | 1          | 1          |
| 3             | Przepiórka  | Marzena     | 2700,00          | NULL   | 89121203456 | Specjalista         | NULL       | 1          |
| 4             | Burzych     | Paweł       | 1900,00          | 500,00 | 78032309123 | Specjalista         | 2          | 7          |
| 5             | Makłowicz   | Marek       | 2000,00          | NULL   | 54013112345 | Specjalista         | 2          | 7          |
| 6             | Naramowicka | Magdalena   | 2100,00          | 200,00 | 77121312098 | Specjalista         | 1          | 1          |
| 7             | Witos       | Jacek       | 3000,00          | 500,00 | 69100967234 | Manager             | 3          | 7          |
| 8             | Markowski   | Maksymilian | 2000,00          | NULL   | 67121209878 | Młodszy specjalista | NULL       | 7          |

**Klienci**

| nr_klienta | nazwisko | imie       | pesel       |
|------------|----------|------------|-------------|
| 1          | Mikołaj  | Wiśniewski | 82040112389 |
| 2          | Marcin   | Kaniewski  | 73122990123 |

**Miejsca**

| nr_miejsca | ulica       | numer | miasto   | kod    | telefon        |
|------------|-------------|-------|----------|--------|----------------|
| 1          | Mała        | 13    | Poznań   | 60-002 | (61) 123 09 89 |
| 2          | Wąska       | 4/12A | Kraków   | 30-012 | (12) 234 23 23 |
| 3          | Grunwaldzka | 34    | Warszawa | 00-123 | (22) 876 62 87 |

**DAILY GROUP** Lekcja 9 – indeksy, constrainty

**Ćwiczenie 1**

Utwórz klucze główne w trzech tabelach

**Ćwiczenie 2**

Utwórz klucz obcy w tabeli pracownicy – kolumna nr\_miejsca

**Ćwiczenie 3**

Dodaj unikalność pola pesel

**Ćwiczenie 4**

Wprowadź mechanizm zabezpieczający w tabeli pracownicy przed dodaniem wiersza z premią mniejszą niż 0

**Ćwiczenie 5**

Jak zoptymalizować zapytania do tabeli klientów zakładając, że zazwyczaj wyszukiwani są po nazwisku? Zaproponuj skrypt wprowadzający zmianę.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

### Ćwiczenie 6

Jakie indeksy należy założyć na tabeli pracownicy, najczęstsze zapytania to wyszukiwaniu po parze kolumn imię i nazwisko oraz po numerze pesel

### Ćwiczenie 7

Napisz skrypt definiujący tabelę pracownicy, gdzie pola imię, nazwisko, pesel i stanowisko nie mogą być puste, nr\_pracownika jest PK, nr\_miejsca FK, pesel musi być unikalny, a płaca większa niż 1600zł

### Ćwiczenie 8

Napisz funkcję sprawdzającą czy wartość nie jest większa od płacy minimalnej (1600zł) podłącz funkcję jako CHECK CONSTRAINT do kolumny płaca zasadnicza w tabeli pracownicy

Slajd 25

**Ćwiczenia**

- Utwórz klucz główny w tabeli TBL\_UCZEN na polu id\_ucznia

```
ALTER TABLE TBL_UCZEN
ADD PRIMARY KEY (id_ucznia)
```

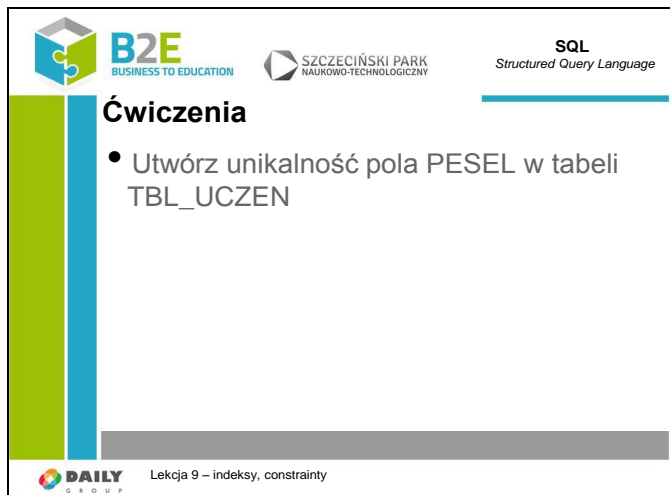
Slajd 26

**Ćwiczenia**

- Utwórz indeks o nazwie i\_tbl\_uczen na tabeli TBL\_UCZEN na polu id\_ucznia

```
CREATE INDEX i_tbl_uczen
ON TBL_UCZEN
```

Slajd 27



Slide 27 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to 'Utwórz unikalność pola PESEL w tabeli TBL\_UCZEN' (Create uniqueness for the PESEL field in the TBL\_UCZEN table). The footer includes the DAILY GROUP logo and the text 'Lekcja 9 – indeksy, constrainty'.

```
ALTER TABLE TBL_UCZEN
ADD UNIQUE (PESEL)
```

Slajd 28



Slide 28 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to 'Dodaj sprawdzanie (walidację) pola IMIE w tabeli TBL\_UCZEN. Pole IMIE nie może być krótsze niż 3 znaki' (Add validation for the IMIE field in the TBL\_UCZEN table. The IMIE field cannot be shorter than 3 characters). The footer includes the DAILY GROUP logo and the text 'Lekcja 9 – indeksy, constrainty'.

```
ALTER TABLE TBL_UCZEN
WITH CHECK ADD
CONSTRAINT CK_IMIE CHECK(
LEN(IMIE)>=3)
```

Slajd 29



Slide 29 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main heading is 'Ćwiczenia' (Exercises). A single bullet point instructs the user to 'Utwórz unikalny indeks na polu ID\_UCZNIA w tabeli TBL\_UCZEN' (Create a unique index on the ID\_UCZNIA field in the TBL\_UCZEN table). The footer includes the DAILY GROUP logo and the text 'Lekcja 9 – indeksy, constrainty'.

```
CREATE UNIQUE INDEX
indeks_unikalny ON
TBL_UCZEN (ID_UCZNIA)
```



Slajd 30

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**Podsumowanie**

|            |                                                   |
|------------|---------------------------------------------------|
| Indeks     | Struktura w bazie danych optymalizująca wydajność |
| Constraint | Ograniczenie, służące zachowaniu spójności danych |

Lekcja 9 – indeksy, constrainty

Tabela podsumowująca poznane wyrażenia

### Ćwiczenia

Ćwiczenie zostało przedstawione na slajdach 24-29. Zadaniem uczniów będzie praktyczna implementacja indeksów oraz constraintów.

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą mogli samodzielnie implementować tabele, z odpowiednimi ograniczeniami służącymi zachowaniu integralności bazy danych. Dodatkowo po zapoznaniu się z mechanizmem indeksów będą mogli podjąć próbę optymalizacji baz danych.

### Lekcja 10 - Transakcje, zatwierdzanie zmian, obsługa błędów, wprowadzenie do projektowania bazy danych

#### Cel lekcji

Celem lekcji jest wyjaśnienie transakcyjności bazy danych, wyjaśnienie istoty obsługi błędów oraz wprowadzenie do projektowania baz danych.



## Treść - slajdy z opisem

### Slajd 1

SQL  
Structured Query Language

# SQL

## Structured Query Language

Lektja 10:  
Transakcje, zatwierdzanie zmian, obsługa błędów, wprowadzenie do projektowania bazy danych

Celownik - najlepsza inwestycja

KAPITAŁ LUDZKI  
INICJATYWA WZROSTU I ZAMIESZKANIA

UNA EUROPEJSKA  
KONKURENCYJNOŚĆ I WYKSZTAŁCENIE

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

DAILY GROUP SQL Structured Query Language Lektja 10

Ostatnia lektja w kursie SQL dotyczyć będzie transakcyjności baz danych. Omówimy istotę obsługi błędów. Oras zajmujemy się zagadnieniami związanymi z normalizacją baz danych – jako podstawy do projektowania baz.

### Slajd 2

SQL  
Structured Query Language

## Przypomnienie


- INDEKS
- CONSTRAINT

DAILY GROUP Lektja 10 –Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych


Podsumowanie zagadnień z poprzedniej lektji



Slajd 3



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*


---

## Transakcje

- BEGIN TRAN - rozpoczęcie transakcji
- COMMIT TRAN - zatwierdzenie transakcji
- ROLLBACK TRAN - wycofanie transakcji

Domyślnie ustawiona jest opcja IMPLICIT\_TRANSACTIONS na OFF. Przy takim ustawieniu, jeśli nie zastosujemy BEGIN TRAN, system traktuje każdą instrukcję DML jako osobną transakcję i zatwierdza ją. Aby to wyłączyć, należy użyć (wtedy BEGIN TRAN przestaje być konieczne):


**SET IMPLICIT\_TRANSACTIONS ON**




Lekcja 10 –Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

Podstawowe komendy związane z transakcjami.

Slajd 4



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

---


## Transakcje

- Przykład

```


BEGIN TRANSACTION UsuwaniePracownika
 WITH MARK N'Usuwanie pracownika';
GO
USE Test1;
GO
DELETE FROM pracownicy WHERE nr_pracownika = 13;
GO
COMMIT TRANSACTION UsuwaniePracownika;
GO

```




Lekcja 10 –Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

WITH MARK – oznacza transakcje odpowiednim opisem (można go później odnaleźć w logu transakcji).



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Obsługa błędów


```

create table dane (id int, produkt nvarchar(30))

BEGIN TRANSACTION;-- początek transakcji
BEGIN TRY
 INSERT INTO dane (id,produkt) SELECT 1,'masio'
 SELECT 2/0
END TRY

BEGIN CATCH
 SELECT
 ERROR_NUMBER() AS ErrorNumber
 ,ERROR_SEVERITY() AS ErrorSeverity
 ,ERROR_STATE() AS ErrorState
 ,ERROR_PROCEDURE() AS ErrorProcedure
 ,ERROR_LINE() AS ErrorLine
 ,ERROR_MESSAGE() AS ErrorMessage;
 IF @@TRANCOUNT > 0
 ROLLBACK TRANSACTION;-- wycofywanie zmian
END CATCH;
IF @@TRANCOUNT > 0
 COMMIT TRANSACTION;-- akceptacja zmian
SELECT * FROM dane|


```




Lekcja 10 –Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

Tak jak w innych językach programowania, tak w SQLu powinniśmy zadbać o obsługę błędów – to znaczy jeśli coś pójdzie nie tak w trakcie wykonywania kodu, powinniśmy „obsłużyć” problem i odpowiednio zareagować. Na slajdzie przedstawiono przykład prawidłowej obsługi błędów w SQL.

Przykład przedstawia celowe spowodowanie błędu: dzielenie przez 0, jego obsługa polega na odczytaniu z systemu wszystkich możliwych danych i zaprezentowanie użytkownikowi.



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Obsługa błędów

- Kiedy stosować?
  - W transakcjach - po wykryciu błędu wycofujemy transakcję,
  - W procedurach składowanych - wykrywamy błędy powstałe głównie w wyniku niepoprawnych wartości parametrów przez użytkownika,
  - W triggerach - podobnie jak w transakcjach, po napotkaniu błędu wycofywana jest transakcja wywołująca trigger,
  - W blokach kodu SQL - wszelkie rozbudowane bloki kodu wymagają wykrywania błędów



Lekcja 8 – procedury i triggerzy

Kiedy obsługa błędów jest niezbędna.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 7

**Obsługa błędów**

- Jak wykrywać błędy?
  - IF...ELSE
  - @@ERROR
  - TRY...CATCH

SQL  
Structured Query Language

DAILY GROUP  
Lekcja 8 – procedury i triggerzy

Jak można wykrywać błędy? Metod na to jest wiele. Po pierwsze można zastosować instrukcje warunkowe, np. IF...ELSE, do sprawdzania wartości zmiennych jeszcze przed wystąpieniem błędu. Druga metoda to wykorzystanie istniejących w SZBD funkcji wykrywających błędy. W systemie Microsoft SQL

Server taką funkcją jest @@ERROR, która zwraca numer błędu napotkanego w ostatnio napotkanego błędu w bieżącej sesji.

Aktualnie istnieją także inne - lepsze metody wykrywania i obsługi błędów. Chodzi o strukturalną obsługę wyjątków. Jako wyjątek rozumiemy błąd, który wymaga obsługi, jedną z takich metod jest TRY...CATCH (TRY - próbuj, CATCH - przechwyć i obsłuż)

Slajd 8

**TRY CATCH**

- Składnia:
 

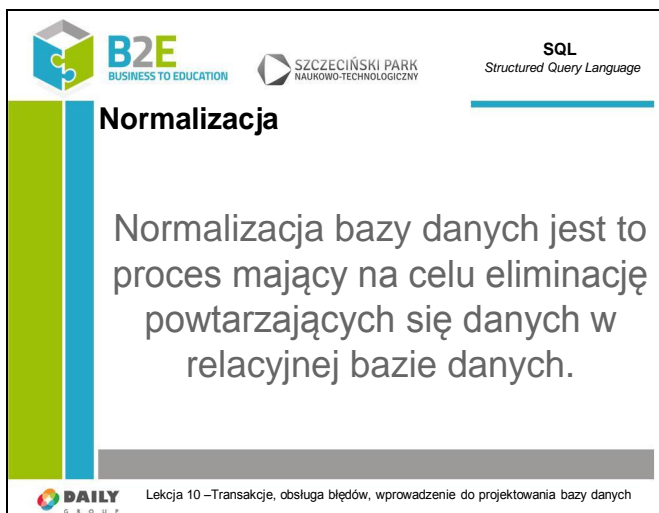
```
BEGIN TRY
 { sql_statement | statement_block }
END TRY
BEGIN CATCH
 [{ sql_statement | statement_block }]
END CATCH [;]
```
- W bloku try – kod stanowiący potencjalne ryzyko
- W bloku catch – obsługa ryzyka

SQL  
Structured Query Language

DAILY GROUP  
Lekcja 8 – procedury i triggerzy

Wyjaśnienie try – catch. Konstrukcja stosowana jest w większości obecnych języków programowania i jest jednym ze sposobów obsługi wyjątków. Jeśli w bloku TRY nastąpi błąd – nastąpi przejście do bloku CATCH. Jeśli nie będzie błędu – blok CATCH nigdy nie zostanie wywołany.

Slajd 9



Slide 9 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL (Structured Query Language). The main title is "Normalizacja". The text explains that normalization is a process to eliminate repeating data in a relational database. At the bottom, it identifies the slide as part of "Lekcja 10 - Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych" from "DAILY GROUP".

Główna idea polega na trzymaniu danych w jednym miejscu, a w razie potrzeby linkowania do danych. Taki sposób tworzenia bazy danych zwiększa bezpieczeństwo danych i zmniejsza ryzyko powstania niespójności (w szczególności problemów anomalii).

Istnieją sposoby ustalenia czy dany schemat bazy danych jest "znormalizowany", a jeżeli jest to jak bardzo. Jednym ze sposobów jest przyrównanie danej bazy do schematów zwanych postaciami normalnymi (ang. normal forms lub NF).

Slajd 10



Slide 10 content: The slide features the same header as slide 9. The main title is "1NF - Pierwsza postać normalna". The text states that the only condition is that every attribute in every record must be atomic (cannot be divided into smaller values). A key characteristic of a relation created according to the relational model is that all attributes are always normalized - they satisfy 1NF. At the bottom, it identifies the slide as part of "Lekcja 10 - Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych" from "DAILY GROUP".

Przykład:

Czy pole adres jest polem elementarnym czy nie? Jeśli wiemy, że w czasie operowania na bazie zawsze będziemy potrzebowali całego adresu, to pole adres możemy uznać za elementarne. Jeśli jednak dopuszczamy możliwość, że będziemy potrzebowali tylko samej miejscowości, to wtedy pole adres nie jest już elementarne i nie spełnia 1NF. Należy więc rozbić pole adres, np. na pola: ulica, miejscowość, kod pocztowy (czyli na pola elementarne).



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 11

Rekordy nie powinny zależeć od niczego innego tylko od klucza podstawowego tabeli (w razie potrzeby może to być klucz złożony). Rozważmy na przykład adres klienta w systemie księgowym. Obecność adresu konieczna jest w tabeli Klienci, ale również w tabelach Zamówienia, Wysyłka, Faktury, Należności i Inkaso. Zamiast przechowywać adres w postaci wpisu w każdej tabeli, przechowuje się go w jednym miejscu: albo w tabeli Klienci, albo w oddzielnej tabeli Adresy.

Slajd 12

Wartości rekordu, które nie są częścią jego klucza, nie należą do tabeli. Zazwyczaj, jeśli zawartość grupy pól odnosi się do więcej niż jednego rekordu tabeli, należy rozważyć umieszczenie tych pól w oddzielnej tabeli. Na przykład w tabeli Rekrutacja pracowników może znajdować się nazwa i adres uczelni, którą ukończył kandydat. Do korespondencji seryjnej potrzebna jest jednak kompletna lista uczelni. Jeśli informacje o uczelniach przechowywane są w tabeli Kandydaci, nie ma możliwości wyświetlenia listy uczelni bez aktualnych kandydatów. Utwórz oddzielną tabelę Uczelnie i połącz ją z tabelą Kandydaci za pomocą klucza z kodem uczelni. WYJĄTEK: Stosowanie reguł trzeciej postaci normalnej, chociaż teoretycznie wskazane, nie zawsze jest praktyczne. Chcąc wyeliminować wszystkie możliwe wewnętrzne zależności

pomiędzy polami tabeli Klienci, należałoby utworzyć oddzielne tabele dla miast, kodów pocztowych, przedstawicieli handlowych, klas klienta i innych czynników, które mogą być zduplikowane w wielu rekordach. Normalizacja oznacza teoretycznie poprawę wydajności. Jednak wiele mniejszych tabel może spowodować spadek wydajności lub brak możliwości otwarcia pliku i przekroczenie pojemności pamięci.

Slajd 13

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Ćwiczenie normalizacja

- Wyeliminuj pola, które nie zależą od klucza. (uzyskaj 1NF)

| NrStudenta | Doradca  | Pok-Dor | Klasa1 | Klasa2 | Klasa3 |
|------------|----------|---------|--------|--------|--------|
| 1022       | Nowak    | 412     | 101-07 | 143-01 | 159-02 |
| 4123       | Kowalski | 216     | 201-01 | 211-02 | 214-01 |

Lekcja 10 – Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

Spróbujmy na innej tabeli: jak powinna wyglądać ta tabela w 1NF?



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**

Structured Query Language

## Ćwiczenie normalizacja

- Pierwsza postać normalna: brak powtarzających się grup

| NrStudenta | Doradca  | Pok-Dor | NrKlasy |
|------------|----------|---------|---------|
| 1022       | Nowak    | 412     | 101-07  |
| 1022       | Nowak    | 412     | 143-01  |
| 1022       | Nowak    | 412     | 159-02  |
| 4123       | Kowalski | 216     | 201-01  |
| 4123       | Kowalski | 216     | 211-02  |
| 4123       | Kowalski | 216     | 214-01  |



Lekcja 10 – Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

Rozwiązaniem problemu jest relacja jeden-do-wielu, w której nie należy strony jeden i strony wielu umieszczać w tej samej tabeli. Zamiast tego, należy utworzyć inną tabelę w pierwszej postaci normalnej, eliminując powtarzające się grupy (NrKlasy), tak jak to przedstawiono na slajdzie.





## Ćwiczenie normalizacja

- Druga postać normalna: eliminowanie powtarzających się danych

| NrStudenta | Doradca  | Pok-Dor | NrStudenta | NrKlasy |
|------------|----------|---------|------------|---------|
| 1022       | Nowak    | 412     | 1022       | 101-07  |
| 4123       | Kowalski | 216     | 1022       | 143-01  |
|            |          |         | 1022       | 159-02  |
|            |          |         | 4123       | 201-01  |
|            |          |         | 4123       | 211-02  |
|            |          |         | 4123       | 214-01  |



W poprzedniej tabeli dla każdego pola NrStudenta istnieje wiele wartości w polach NrKlasy. Pole NrKlasy nie jest czynnościowo zależne od pola NrStudenta (klucz podstawowy), dlatego ta relacja nie znajduje się w drugiej postaci normalnej.

Drugą postacią normalną przedstawiono na podstawie następujących dwóch tabel: Pierwsza postać normalna: studenci i rejestracja.



Slajd 16

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Ćwiczenie normalizacja

- Trzecia postać normalna: eliminowanie danych, które nie zależą od klucza

| Studenci:  |          | Wydział: |       |         |
|------------|----------|----------|-------|---------|
| NrStudenta | Doradca  | Nazwa    | Pokój | Wydział |
| 1022       | Nowak    | Nowak    | 412   | 42      |
| 4123       | Kowalski | Kowalski | 216   | 42      |

Lekcja 10 – Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

W ostatnim przykładzie pole Pok-Dor (numer pokoju doradcy) jest czynnościowo zależne od atrybutu Doradca. Rozwiązaniem jest przeniesienie tego atrybutu z tabeli Studenci do tabeli Wydział, tak jak to przedstawiono na slajdzie.

Slajd 17

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Normalizacja plusy i minusy

Cel normalizacji:

- uniknięcie redundancji (tj. powtarzania się pól z identycznymi wartościami w różnych tabelach);
- wyeliminowanie niewygodnych relacji wieloznacznych;
- uniknięcie anomalii przy aktualizacji: modyfikacji, wstawianiu i usuwaniu;
- uniknięcie niespójności.

Koszt:

- Mnożenie liczby tabel – wydłużenie czasu dostępu do danych.

Poszukiwanie kompromisu (dwa współzawodniczące cele):

- Zapobieganie anomalii oraz zapewnienie rozsądnego czasu dostępu do danych.

Lekcja 10 – Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

Normalizacja ma swoje wady i zalety. W praktyce podczas projektowania baz danych najczęściej szukamy kompromisu.



## Ćwiczenia

Wprowadź obsługę błędów w procedurze (TRY-CATCH i RAISERROR):

```
CREATE PROCEDURE podzieli1
(
 @dzielna float,
 @dzielnik float
)
AS
BEGIN
 SELECT @dzielna / @dzielnik
END
```



Np.

```
CREATE PROCEDURE podzieli1
(
 @dzielna float,
 @dzielnik float
)
AS
IF @dzielnik = 0
RAISERROR('Dzielenie przez ZERO!',14,1)
ELSE
SELECT @dzielna / @dzielnik
END
```



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 19




**SQL**  
Structured Query Language

## Ćwiczenia



Przeprowadź normalizację tabeli:

| nazwa                      | imie prowadzacego | nazwisko prowadzacego | adres prowadzacego        |
|----------------------------|-------------------|-----------------------|---------------------------|
| język angielski dla dzieci | Katarzyna         | Nowak                 | ul. Zgierska 19 m. 6 Łódź |
| judo                       | Tomasz            | Woźniak               | ul. Motylowa 5 Łódź       |
| rytmika                    | Zofia             | Leśniak               | ul. Złota 14/17 Zgierz    |
| język francuski            | Katarzyna         | Nowak                 | ul. Zgierska 19 m. 6 Łódź |
| karate                     | Janusz            | Kot                   | Smardzew 8                |
| gimnastyka korekcyjna      | Tomasz            | Woźniak               | ul. Motylowa 5 Łódź       |
| ...                        |                   |                       |                           |



Lekcja 10 –Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

Slajd 20





**SQL**  
Structured Query Language

## Ćwiczenia

Wprowadź obsługę błędów w poniższej procedurze (w przypadku błędu zwróć -1):

```
CREATE PROCEDURE dbo.p_dziel_przez_0
AS
BEGIN
DECLARE @wynik int;
SELECT @wynik = 1/0;
RETURN @wynik;
END TRY
BEGIN CATCH
RETURN -1;
END CATCH
END
GO
```



Lekcja 10 –Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

```
CREATE PROCEDURE
dbo.p_dziel_przez_0
AS
BEGIN
BEGIN TRY
DECLARE @wynik int;
SELECT @wynik = 1/0;
RETURN @wynik;
END TRY
BEGIN CATCH
RETURN -1;
END CATCH
END
GO
```

Slajd 21




**SQL**  
Structured Query Language

## Ćwiczenia

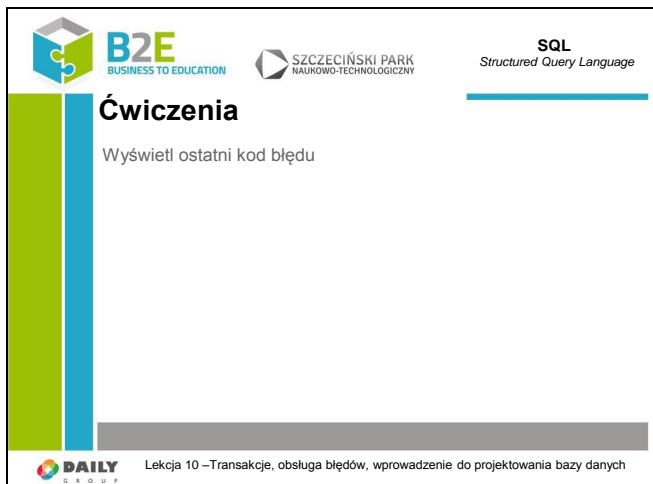
Zmodyfikuj procedurę z poprzedniego ćwiczenia, tak aby obok -1 zwróciła kod błędu



Lekcja 10 –Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

```
CREATE PROCEDURE
dbo.p_dziel_przez_0
AS
BEGIN
BEGIN TRY
DECLARE @wynik int;
SELECT @wynik = 1/0;
RETURN @wynik;
END TRY
BEGIN CATCH
SELECT -1, ERROR_NUMBER() AS
ErrorNumber
RETURN -1;
END CATCH
END
GO
```

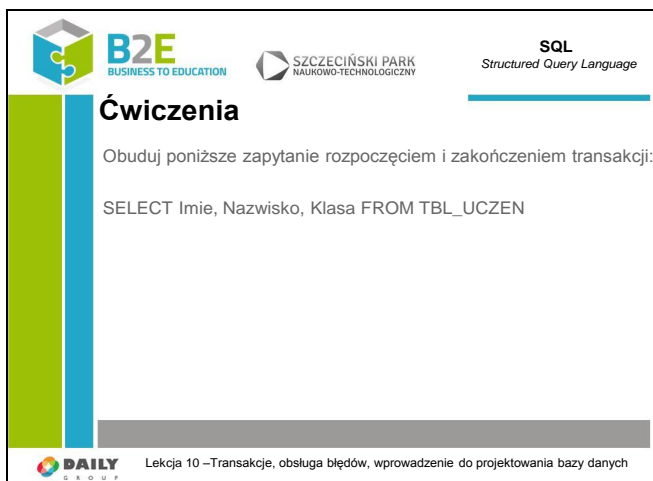
Slajd 22



Slide 22 content: B2E BUSINESS TO EDUCATION logo, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY logo, SQL Structured Query Language title, and exercise instructions: 'Wyświetl ostatni kod błędu'.

SELECT @@ERROR

Slajd 23



Slide 23 content: B2E BUSINESS TO EDUCATION logo, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY logo, SQL Structured Query Language title, and exercise instructions: 'Obuduj poniższe zapytanie rozpoczęciem i zakończeniem transakcji: SELECT Imie, Nazwisko, Klasa FROM TBL\_UCZEN'.

```
BEGIN TRANSACTION
MojaTransakcja
GO
SELECT Imie, Nazwisko, Klasa
FROM TBL_UCZEN
GO
COMMIT TRANSACTION
MojaTransakcja
GO
```

Slajd 24



Slide 24 content: B2E BUSINESS TO EDUCATION logo, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY logo, SQL Structured Query Language title, and exercise instructions: 'Wymień i scharakteryzuj postaci normalne'.



Slajd 25



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

## Podsumowanie

|                   |                                                                |
|-------------------|----------------------------------------------------------------|
| TRANSACTION       | Zestaw operacji w SQL, zamin następuje faktyczne zatwierdzenie |
| COMMIT            | Służy do zatwierdzania transakcji                              |
| ROLLBACK          | Służy co odrzucenia transakcji                                 |
| TRY ... CATCH ... | Zabezpieczenie kodu                                            |
| Normalizacja      | Eliminacja powtarzających się danych w relacyjnej bazie danych |
| 1NF               | Pierwsza postać normalna                                       |
| 2NF               | Druga postać normalna                                          |
| 3NF               | Trzecia postać normalna                                        |



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI

Człowiek - najlepsza inwestycja



UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



Lekcja 10 – Transakcje, obsługa błędów, wprowadzenie do projektowania bazy danych

Tabela podsumowująca poznane wyrażenia

### Ćwiczenia

Ćwiczenie zostało przedstawione na slajdach 18 - 24, będzie polegało na wprowadzeniu obsługi błędów w procedurze składowanej oraz na normalizacji przykładowej tabeli.

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą mogli podjąć próbę zaprojektowania bazy danych. Bloki kodu przez nich tworzone będą zawierały obsługę błędów, będą stosować transakcyjne podejście w programowaniu baz danych.

## Lekcja 11 – Projekt interdyscyplinarny - Książka teleadresowa

### Cel lekcji

Celem lekcji jest zaprojektowanie bazy danych do użycia w na stronie WWW oraz w aplikacji mobilnej z funkcjonalnością książki teleadresowej kolegów i koleżanek ze szkoły lub klasy. W tej lekcji uczniowie dowiedzą się jak praktycznie wykorzystać wiedzę zdobytą w tym module na potrzeby pierwszego projektu informatycznego składającego się z elementów wykorzystujących cztery języki programowania: SQL, PHP, JavaScript i JAVA.

## Treść - slajdy z opisem

Slajd 1

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The main title is 'SQL Structured Query Language' with the subtitle 'Lekcja 11: Interdyscyplinarny projekt: Książka teleadresowa kolegów/koleżanek ze szkoły'. Logos for 'KAPITAŁ LUDZKI' and 'UNIA EUROPEJSKA' are also present. The footer includes the DAILY GROUP logo and the text 'SQL Structured Query Language Lekcja 11'.

Proces powstawania książki teleadresowej kolegów/koleżanek ze szkoły. W module SQL pokażemy jak zaprojektować bazę danych, na PHP i JavaScript, jak stworzyć stronę WWW do dodawania kolegów i koleżanek z klasy do takiej książki, a w module JAVA nauczymy się, jak zaprogramować prostą aplikację na telefony z systemem operacyjnym Android do przeszukiwania danych o uczniach z różnych klas.

Slajd 2

The slide is titled 'Założenia' and lists the requirements for the application. It includes logos for B2E, Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The footer contains the DAILY GROUP logo and the text 'Lekcja 11 – Interdyscyplinarny projekt'.

Założenia naszej aplikacji – chcemy aby aplikacja umożliwiała prezentację listy klas w naszej szkole. Tak abyśmy mogli swobodnie przeglądać uczniów w każdej z klas. Po wybraniu szukanego ucznia chcielibyśmy zaprezentować użytkownikom naszej aplikacji (mobilnej lub webowej) podstawowe informacje dotyczące ucznia: imię, nazwisko, adres e-mail, numer telefonu oraz zdjęcie.

Slajd 3

The slide is titled 'Propozycja bazy danych' and lists the proposed database tables. It includes logos for B2E, Szczeciński Park Naukowo-Technologiczny, and SQL Structured Query Language. The footer contains the DAILY GROUP logo and the text 'Lekcja 11 – Interdyscyplinarny projekt'.

W tym celu potrzebne nam będą dwie tabele, które zgodnie z najlepszymi praktykami oznaczmy prefixem „tbl\_”: tbl\_uczen oraz tbl\_klasa.



Slajd 4



**B2E**  
BUSINESS TO EDUCATION



**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
*Structured Query Language*

## Propozycja bazy danych

| tbl_uczen |              |
|-----------|--------------|
| id_ucznia | klucz główny |
| imie      |              |
| nazwisko  |              |
| id_klasy  |              |
| tel       |              |
| email     |              |
| zdjecie   |              |

—

| tbl_klasa   |              |
|-------------|--------------|
| id_klasy    | klucz główny |
| nazwa_klasy |              |
| wychowawca  |              |
| zdjecie     |              |



Lekcja 11 – Interdyscyplinarny projekt

Tworzymy relacyjną bazę danych. Nawet przy dwóch tabelach taką relację należy utworzyć. Kluczem głównym tabeli tbl\_uczen będzie id\_ucznia, kluczem głównym tabeli tbl\_klasa – id\_klasy. Kluczem obcym tabeli tbl\_uczen będzie id\_klasy.





**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Bazy danych - skrypt

```

CREATE DATABASE SPNT

USE [SPNT]

CREATE TABLE [dbo].[tbl_klasa](
 [id_klasy] [int] IDENTITY(1,1) NOT NULL,
 [nazwa_klasy] [nvarchar](50) NOT NULL,
 [wychowawca] [nvarchar](150) NOT NULL,
 [zdjecie] [nchar](250) NULL,
 CONSTRAINT [PK_tbl_klasa] PRIMARY KEY CLUSTERED
 (
 [id_klasy] ASC
)
 ON [PRIMARY]

CREATE TABLE [dbo].[tbl_uczen](
 [id_ucznia] [int] IDENTITY(1,1) NOT NULL,
 [imie] [nvarchar](150) NOT NULL,
 [nazwisko] [nvarchar](150) NOT NULL,
 [id_klasy] [int] NOT NULL,
 [tel] [nchar](10) NULL,
 [email] [nvarchar](250) NULL,
 [zdjecie] [nvarchar](250) NULL,
 CONSTRAINT [PK_tbl_uczen] PRIMARY KEY CLUSTERED
 (
 [id_ucznia] ASC
)
 ON [PRIMARY]

ALTER TABLE [dbo].[tbl_uczen] WITH CHECK ADD CONSTRAINT [FK_tbl_uczen_tbl_klasa] FOREIGN KEY([id_klasy])
REFERENCES [dbo].[tbl_klasa] ([id_klasy])
GO


ALTER TABLE [dbo].[tbl_uczen] CHECK CONSTRAINT [FK_tbl_uczen_tbl_klasa]

```



Lekcja 11 – Interdyscyplinary projekt

Przygotujemy skrypt tworzący bazę danych oraz obie tabele, wraz z kluczami i relacją.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**SQL**  
Structured Query Language

---

## Widok

- Widok prezentujący uczniów wszystkich klas

```

CREATE VIEW [dbo].[v_uczniowie]
AS
SELECT dbo.tbl_uczen.id_ucznia, dbo.tbl_uczen.imie, dbo.tbl_uczen.nazwisko, dbo.tbl_klasa.nazwa_klasy,
dbo.tbl_klasa.wychowawca, dbo.tbl_uczen.tel, dbo.tbl_uczen.email, dbo.tbl_uczen.zdjecie
FROM dbo.tbl_klasa INNER JOIN dbo.tbl_uczen ON dbo.tbl_klasa.id_klasy = dbo.tbl_uczen.id_klasy

```





Lekcja 11 – Interdyscyplinary projekt

Na potrzeby czytelnej dla użytkownika prezentacji listy uczniów z przyporządkowaniem do klasy – utwórzmy właściwy widok.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 7

**SQL**  
Structured Query Language


## Procedura

- Procedura zapisująca nowego ucznia

```

CREATE PROCEDURE dodaj_ucznia
 @imie nvarchar(150),
 @nazwisko nvarchar(150),
 @tel nchar(10),
 @email nvarchar(150),
 @id_klasy int
AS
BEGIN
 INSERT INTO [dbo].[tbl_uczen] (imie, nazwisko, tel, email, id_klasy)
 VALUES (@imie, @nazwisko, @tel, @email, @id_klasy)
END
GO



```



Lekcja 11 – Interdyscyplinarny projekt

Dobrze jest nie wykonywać operacji zapisu z kodu PHP przez bezpośrednie napisanie kwerendy INSERT – utwórzmy więc procedurę do zapisu nowego ucznia.

Slajd 8

**SQL**  
Structured Query Language


## Funkcja

- Funkcja zwracająca ilość uczniów w klasie

```

CREATE FUNCTION dbo.f_ile_w_klasie
(
 @id_klasy int
)
RETURNS int
AS
BEGIN
 DECLARE @ile int
 SELECT @ile = COUNT(*) FROM tbl_uczen WHERE id_klasy = @id_klasy
 RETURN @ile
END
GO



```



Lekcja 11 – Interdyscyplinarny projekt

Być może jako programistom przyda nam się funkcja zwracająca ilość uczniów w klasie. Utwórzmy taką funkcję.


Slajd 9


**SQL**  
Structured Query Language

## Podsumowanie


- Gotowa prosta baza danych do gromadzenia informacji na temat uczniów i klas
- Widok wyświetlający w przyjemny dla użytkownika sposób informacje o uczniach
- Procedura do zapisu uczniów
- Funkcja do obliczania ilości uczniów



Człowiek - najlepsza inwestycja



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



Lekcja 11 – Interdyscyplinarny projekt

Podsumowanie utworzonych obiektów bazodanowych.

**Wprowadzenie**

Obecnie PHP jest jednym z najpopularniejszych języków programowania na świecie. W roku 1997 język PHP/FI obsługiwało około 50 tysięcy domen internetowych. Do roku 2005 udział PHP rósł. Później uległ stabilizacji na poziomie ok. 20 milionów domen (9% wszystkich). Obecnie jest dominującym językiem programowania i obejmuje około 75% domen internetowych.

Jest on rozwijany na zasadach licencji Open Source, więc każdy może mieć wgląd w jego kod źródłowy. Należy zaznaczyć, że PHP jest językiem skryptowym, więc aplikacje w nim napisane nie są kompilowane do postaci binarnej, lecz interpretowane przy każdym uruchomieniu.

PHP służy przede wszystkim do tworzenia dynamicznych stron WWW, czyli np. sklepów internetowych, systemów rejestracji użytkowników, statystyk itp. Zwyczajne, statyczne strony WWW to zwykle pliki tekstowe (najczęściej z rozszerzeniem \*.htm lub \*.html) zawierające instrukcje zapisane w języku HTML. Przeglądarka łącząc się z serwerem, pobiera zawartość takiego pliku, a następnie odpowiednio go interpretuje. W ten sposób na ekranie swoich komputerów możemy zobaczyć wszelkiego rodzaju interaktywne elementy, odnośniki, animacje, obrazki itp.

Możliwości PHP są bardzo duże, możemy np. napisać:

- sklepy internetowe;
- systemy ankiet;
- systemy quizów;
- fora dyskusyjne;
- systemy rejestracji użytkowników;
- statystyki odwiedzin strony;
- systemy dodawania aktualności;

Jednym z najpopularniejszych portali napisanych w PHP jest Allegro.

**Cele**

Poznanie zagadnień z zakresu języka PHP oraz protokołów sieciowych takich jak HTTP, SMTP oraz MIME. Następnie wprowadzenie do mechanizmów ciasteczek i sesji. Zdobyć wiedzy na temat podstawowych architektur aplikacji internetowych oraz bezpieczeństwa stron, bądź portali internetowych. Zobrazowanie jak radzić sobie z atakami na strony, przez wykorzystanie podstawowych metod zabezpieczeń. Kurs ma również na celu zachęcić uczestnika do dalszego rozwoju, poprzez wskazanie mu bibliotek, frameworków oraz narzędzi upłynniających i ułatwiających pracę z językiem PHP.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

### **Opis sposobu realizacji celów**

10 półtoragodzinnych lekcji składających się z przedstawienia materiału wraz z przykładami oraz ćwiczeniami praktycznymi. Po zakończeniu całego cyklu - przeprowadzenie egzaminu sprawdzającego wiedzę.

Dodatkowo wprowadzono lekcję 11 zawierającą część projektu interdyscyplinarnego łączącego 4 dziedziny wiedzy z zakresu informatyki (SQL, PHP, JavaScript oraz JAVA) w zakresie zaprojektowania strony internetowej, która komunikuje się z bazą danych oraz serwisu REST API do komunikacji z klientem na telefonie komórkowym.

### **Treści kształcenia**

Treść kursu została podzielona na 10 bloków tematycznych – po jednym do każdej lekcji:

1. Bezpieczeństwo aplikacji (Rodzaje ataków na portale internetowe, Testowanie poszczególnych zagrożeń, Zabezpieczanie aplikacji).
2. Architektura usług sieciowych (Klient – serwer, Peer to peer, Architektura hybrydowa);
3. Protokół http (budowa protokołu HTTP 1.1, nagłówki żądania i odpowiedzi, komunikacja http);
4. Modyfikacja nagłówków http (Lista nagłówków, Wykorzystanie funkcji header, Zastosowania w skryptach PHP);
5. Cookie - informacje o użytkowniku (Budowa i działanie cookie, Używanie cookie w aplikacjach, Definiowanie i obsługa z poziomu PHP, Bezpieczeństwo);
6. Sesje - współdzielenie informacji (Struktura i sposób działania, Kontrola sesji w PHP, Wykorzystanie sesji w aplikacjach internetowych);
7. Poczta - systemy SMTP (Systemy pocztowe w Internecie, Budowa protokołu SMTP, Komunikacja SMTP, Struktura nagłówków wiadomości, Obsługa poczty w aplikacjach PHP);
8. MIME - rozszerzenia multimedialne (Struktura nagłówków MIME, Kodowanie Base64 i Quoted-Printable, Wiadomości wieloczęściowe, Obsługa poczty multimedialnej w PHP);
9. UML 2.1 (Diagramy czynności, Projektowanie złożonych aplikacji);
10. Dobre praktyki programowania w PHP
11. Projekt interdyscyplinarny – baza teleadresowa kolegów i koleżanek ze szkoły

### **Opis założonych osiągnięć ucznia**

Uczeń po odbyciu kursu pozna protokoły sieciowe, mechanizm ciasteczek i sesji, nauczy się zabezpieczać swoje aplikacje, pozna podstawowe architektury aplikacji internetowych i nauczy się je projektować, będzie posiadał wytyczoną ścieżkę dalszego rozwoju.

### **Sposoby osiągania celów**

Podczas procesu kształcenia, uczeń powinien wykonywać ćwiczenia oraz przykłady zawarte w materiałach edukacyjnych. Po ukończeniu szkolenia, każdy uczestnik musi zaliczyć odpowiedni test, który pomoże zweryfikować jego świeżo nabytą wiedzę oraz określić ewentualne luki w wiedzy, który powinien uzupełnić.

### Lista czynności w zależności od roli

Wykonywane czynności są takie same dla każdej z lekcji, jedynie materiał którego dotyczą jest różny. Może się zdarzyć iż w przypadku braku materiału w danej lekcji dla danej czynności, czynność nie powinna być wykonana.

| Czynności nauczyciela                                                                                                                                                                                                            | Czynności ucznia                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Prezentowanie przygotowanego materiału.                                                                                                                                                                                          | Sporządzanie notatek z prezentowanego materiału |
| Zadawanie pytań kontrolnych odnośnie już pokazanego materiału                                                                                                                                                                    | Udzielanie odpowiedzi na pytania nauczyciela    |
| Zadawanie pytań odnośnie materiału, który dopiero będzie pokazany, jeśli możliwe jest uzyskanie odpowiedzi w drodze dedukcji                                                                                                     |                                                 |
| Objaśnianie ćwiczeń lekcyjnym, pilnowanie czasu ich wykonania przez uczniów, udzielanie wskazówek                                                                                                                                | Rozwiązywanie zadań lekcyjnych                  |
| Objaśnienie zadań na koniec lekcji, udzielanie wskazówek podczas rozwiązywania, zlecenie dokończenia zadań jako pracę domową                                                                                                     | Odrabianie zadań domowych                       |
| Sprawdzanie ćwiczeń lekcyjnych i zadań domowych:<br>należy zwrócić uwagę na poprawne formatowanie kodu (wcięcia)<br>poprawne użycie konstrukcji języka<br>dokumentowanie kodu komentarzami<br>otrzymywanie poprawnych rezultatów |                                                 |
| Zapisywanie uwag odnośnie materiałów szkoleniowych i sugestie ewentualnych korekt.                                                                                                                                               |                                                 |

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Szczegółowe cele kształcenia oraz treści nauczania w korelacji z treściami podstawy programowej

| Programowanie w języku PHP                                                                                                                                                     |                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cele kształcenia                                                                                                                                                               | Treści kształcenia                                                                                                                       | Podstawa programowa                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Uczeń zna zagadnienia związane z bezpieczeństwem stron i portali internetowych oraz umie radzić sobie z atakami na strony przez wykorzystanie podstawowych metod zabezpieczeń. | Lekcja 1: Bezpieczeństwo aplikacji (Rodzaje ataków na portale internetowe, Testowanie poszczególnych zagrożeń, Zabezpieczanie aplikacji) | Uczeń: <ul style="list-style-type: none"> <li>• Korzysta z wbudowanych typów danych</li> <li>• Przestrzega zasad programowania</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Tworzy własne funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Wykorzystuje środowisko programistyczne: edytor, kompilator i debugger</li> <li>• Kompiluje i uruchamia kody źródłowe</li> <li>• Wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera</li> <li>• Testuje tworzoną aplikację i modyfikuje jej kod źródłowy<br/>Zabezpiecza dostęp do tworzonych aplikacji</li> </ul> |
| Uczeń poznaje oraz umie rozróżnić podstawowe architektury aplikacji internetowych.                                                                                             | Lekcja 2: Architektura usług sieciowych (Klient – serwer, Peer to peer, Architektura hybrydowa)                                          | Uczeń: <ul style="list-style-type: none"> <li>• Przestrzega zasad programowania</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Uczeń poznaje zagadnienia z zakresu protokołów sieciowych (HTTP)                                                                                                               | Lekcja 3: Protokół http (budowa protokołu HTTP 1.1, nagłówki żądania i odpowiedzi, komunikacja http)                                     | Uczeń: <ul style="list-style-type: none"> <li>• Przestrzega zasad programowania</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Uczeń potrafi modyfikować nagłówki protokołu http.                                                                                                                             | Lekcja 4: Modyfikacja nagłówków http (Lista nagłówków, Wykorzystanie funkcji header, Zastosowania w skryptach PHP)                       | Uczeń: <ul style="list-style-type: none"> <li>• Korzysta z wbudowanych typów danych</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

|                                                                                                     |                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                     |                                                                                                                                                                     | <ul style="list-style-type: none"> <li>• Wykorzystuje środowisko programistyczne: edytor, kompilator i debugger</li> <li>• Kompiluje i uruchamia kody źródłowe</li> <li>• Wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera</li> <li>• Testuje tworzoną aplikację i modyfikuje jej kod źródłowy</li> </ul>                                                                                                                                                                                                                                                                                                         |
| <p>Uczeń potrafi tworzyć, modyfikować oraz usuwać ciasteczka oraz poznaje zasadę ich działania.</p> | <p>Lekcja 5: Cookie - informacje o użytkowniku (Budowa i działanie cookie, Używanie cookie w aplikacjach, Definiowanie i obsługa z poziomu PHP, Bezpieczeństwo)</p> | <p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Korzysta z wbudowanych typów danych</li> <li>• Przestrzega zasad programowania</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Tworzy własne funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Wykorzystuje środowisko programistyczne: edytor, kompilator i debugger</li> <li>• Kompiluje i uruchamia kody źródłowe</li> <li>• Wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera</li> <li>• Testuje tworzoną aplikację i modyfikuje jej kod źródłowy</li> </ul> |
| <p>Uczeń potrafi tworzyć, modyfikować oraz usuwać sesje PHP oraz poznaje zasadę ich działania.</p>  | <p>Lekcja 6: Sesje - współdzielenie informacji (Struktura i sposób działania, Kontrola sesji w PHP, Wykorzystanie sesji w aplikacjach internetowych)</p>            | <p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Korzysta z wbudowanych typów danych</li> <li>• Przestrzega zasad programowania</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Tworzy własne funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Wykorzystuje środowisko programistyczne: edytor, kompilator i debugger</li> <li>• Kompiluje i uruchamia kody źródłowe</li> <li>• Wykorzystuje języki programowania do tworzenia aplikacji internetowych</li> </ul>                                                                                                              |



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

|                                                                                                                                                |                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                |                                                                                                                                                                                   | <p>realizujących zadania po stronie serwera</p> <ul style="list-style-type: none"> <li>• Testuje tworzoną aplikację i modyfikuje jej kod źródłowy</li> <li>• Zabezpiecza dostęp do tworzonych aplikacji</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <p>Uczeń poznaje zagadnienia z zakresu protokołów sieciowych (SMTP). Potrafi wysłać wiadomość e-mail przy pomocy skryptu napisanego w PHP.</p> | <p>Lekcja 7: Poczta - systemy SMTP (Systemy pocztowe w Internecie, Budowa protokołu SMTP, Komunikacja SMTP, Struktura nagłówek wiadomości, Obsługa poczty w aplikacjach PHP)</p>  | <p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Korzysta z wbudowanych typów danych</li> <li>• Przestrzega zasad programowania</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Tworzy własne funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Wykorzystuje środowisko programistyczne: edytor, kompilator i debugger</li> <li>• Kompiluje i uruchamia kody źródłowe</li> <li>• Wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera</li> <li>• Testuje tworzoną aplikację i modyfikuje jej kod źródłowy</li> </ul> |
| <p>Uczeń poznaje zagadnienia z zakresu protokołów sieciowych (MIME)</p>                                                                        | <p>Lekcja 8: MIME - rozszerzenia multimedialne (Struktura nagłówek MIME, Kodowanie Base64 i Quoted-Printable, Wiadomości wieloczęściowe, Obsługa poczty multimedialnej w PHP)</p> | <p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Korzysta z wbudowanych typów danych</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Wykorzystuje środowisko programistyczne: edytor, kompilator i debugger</li> <li>• Kompiluje i uruchamia kody źródłowe</li> <li>• Wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera</li> <li>• Testuje tworzoną aplikację i modyfikuje jej kod źródłowy</li> </ul>                                                                                                                                         |
| <p>Uczeń poznaje architektury aplikacji; zdobywa umiejętności projektowania</p>                                                                | <p>Lekcja 9: UML 2.1 (Diagramy czynności, Projektowanie złożonych aplikacji)</p>                                                                                                  | <p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Przestrzega zasad programowania</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |



|                                                                                                                                                                                                                                                                      |                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| systemów informatycznych.                                                                                                                                                                                                                                            |                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Uczeń poznaje możliwości swojego przyszłego rozwoju, poprzez wskazanie mu bibliotek, frameworków oraz narzędzi upłynniających i ułatwiających pracę z językiem PHP. Zna narzędzia i rozwiązania stosowane przez programistów podczas ich codziennej pracy zawodowej. | Lekcja 10: Dobre praktyki programowania w PHP | Uczeń: <ul style="list-style-type: none"> <li>• Korzysta z wbudowanych typów danych</li> <li>• Przestrzega zasad programowania</li> <li>• Stosuje instrukcje, funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Tworzy własne funkcje, procedury, obiekty, metody wybranych języków programowania</li> <li>• Wykorzystuje środowisko programistyczne: edytor, kompilator i debugger</li> <li>• Kompiluje i uruchamia kody źródłowe</li> <li>• Wykorzystuje języki programowania do tworzenia aplikacji internetowych realizujących zadania po stronie serwera</li> <li>• Wykorzystuje frameworki do tworzenia własnych aplikacji</li> <li>• Testuje tworzoną aplikację i modyfikuje jej kod źródłowy</li> <li>• Zamieszcza opracowane aplikacje w internecie</li> </ul> |

### Propozycje kryteriów oceny i metod sprawdzania osiągnięć ucznia

Ocena zaliczenia przedmiotu powinna być składową dwóch ocen: testu oraz ocen cząstkowych za ćwiczenia :

Kryteria oceny testu:

- 0-7 poprawnych odpowiedzi - 1
- 8-9 poprawnych odpowiedzi - 2
- 10-11 poprawnych odpowiedzi - 3
- 12-13 poprawnych odpowiedzi - 4
- 14-15 poprawnych odpowiedzi - 5

Kryteria oceny ćwiczeń:

- Rozwiązania ćwiczeń spełniają podstawową funkcjonalność - 3
- Rozwiązania ćwiczeń spełniają podstawową funkcjonalność oraz działają poprawnie w każdym przypadku - 4
- Spełniają powyższe wymagania oraz są zabezpieczone na błędne dane – 5

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Opis założonych osiągnięć ucznia – przykłady wymagań na poszczególne oceny szkolne:

Oceną zaliczającą kurs będzie wynik testu sprawdzającego wiedzę z zakresu kursu, ocena pracy ucznia w trakcie semestru oraz przygotowanie prostego projektu (dla chętnych).

Ocenę dopuszczającą (2) otrzymuje uczeń, który potrafi podłączyć i uruchomić skrypt na stronie internetowej, zna podstawowe pojęcia związane z językiem PHP, potrafi własnymi słowami wyjaśnić podstawowe instrukcje języka.

Ocenę dostateczną (3) otrzymuje uczeń, który spełnia wymagania oceny dopuszczającej oraz potrafi stworzyć własne skrypty pozwalające na operowanie danymi i wyświetlanie ich na stronie WWW. Zna podstawowe struktury danych w języku PHP (zmienne i tablice). Wykonuje ćwiczenia z niewielką pomocą nauczyciela. Potrafi wskazać podstawowe zastosowania języka PHP w tworzeniu stron internetowych

Ocenę dobrą (4) otrzymuje uczeń, który spełnia wymagania na ocenę dostateczną oraz potrafi stworzyć skrypt i podłączyć skrypty reagujące na zdarzenia na stronie internetowej, zna i potrafi wykorzystać obiektowy model programowania w języku PHP. Na lekcjach wykonuje samodzielnie zadane ćwiczenia. Umiejętnie korzysta z mechanizmów pozwalających na modyfikację witryny internetowej.

Ocenę bardzo dobrą (5) otrzymuje uczeń, który spełnia wymagania na ocenę dobrą oraz bardzo dobrze zna teorię omawianą na zajęciach, w trakcie wykonywania ćwiczeń wykazuje inicjatywę, potrafi wskazać kilka rozwiązań zadania, tworzy skrypty języka z wykorzystaniem dobrych praktyk programistycznych. Wykorzystuje funkcje/klasę dostępne w bibliotekach PEAR/PECL.

Ocenę celującą (6) otrzymuje uczeń, którego wiedza wykracza poza omawiany zakres. Np. potrafi stworzyć aplikację/grę w języku PHP. Tworzy zaawansowane interaktywne witryny www, udziela się na forach i grupach dyskusyjnych poświęconych językowi PHP.

### Test końcowy sprawdzający wiedzę

15 zadań na 30 minutowy test sprawdzający wiedzę. Pogrubioną czcionką zaznaczono prawidłowe odpowiedzi.

1. Czym różni się usługa REST w stosunku od innych webservice'ów opartych o WSDL:
  - e. Brak stanowości zasobów**
  - f. Posiada opis formatu zapytań i odpowiedzi
  - g. Jest kompilowany
  - h. Wykorzystuje metody http**
  
2. Która metoda z niżej wymienionych nie jest metodą HTTP w wersji 1.1:
  - a. CONNECT**
  - b. PUT
  - c. OPTIONS
  - d. TRACE

3. Jeżeli w nagłówku HTTP chcemy określić główny język dokumentu jako polski, stosujemy składnie:
- Content-Language: en, pl**
  - Content-Language: pl, en
  - Content-Language: pl**
  - Content-Language: en, pl, de
4. Jak usunąć ciasteczko o nazwie 'hello' w języku php:
- setcookie("hello", "", time()-3600);**
  - deletecookie("hello");
  - unset("hello");
  - unsetcookie("hello");
5. Czy możemy ustawić zmienną w sesji przed wywołaniem funkcji session\_start:
- Tak, za pomocą tablicy \$\_SESSION;
  - Możemy, ale musimy użyć specjalnej funkcji;
  - Nie;**
6. Podaj numer domyślnego portu SMTP?  
Odp: Port o numerze 25.
7. Kodowania transportowe to:
- Base64**
  - Quoted-Printable**
  - ISO 8859-2
  - Utf-16
8. Do diagramów języka UML nie zaliczamy:
- Diagramu klas
  - Diagramu pakietów
  - Diagramu konstelacji**
  - Diagramu czynności
9. Rodzajem bezpośredniego ataku na portale internetowe lub ich serwery nie jest:
- PHP Injection
  - XSS Injection
  - SQL Injection
  - Phishing**
10. Szkieletem aplikacji języka PHP nie jest:
- Zend Framework
  - Symfony
  - Composer**



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

d. Yii

11. Do pisania jakich systemów służy język php:

- a. **Systemów działających po stronie serwera**
- b. Aplikacji desktopowych
- c. Systemów do obsługi serwera WWW
- d. Sterowników urządzeń

12. Mechanizm sesji przechowuje dane (nie licząc identyfikatora) po stronie:

- a. Klienta
- b. **Serwera**
- c. Serwera oraz klienta
- d. W dowolnej bazie danych, wybranej przez użytkownika
- e. Mechanizm sesji nie przechowuje żadnych danych

13. Który protokół z wymienionych poniżej, nie jest protokołem pocztowym:

- a. **FTP**
- b. SMTP
- c. POP3
- d. IMAP

14. Do czego służy narzędzie ping:

- a. Jest to system zarządzania treścią
- b. Generowania dokumentacji
- c. **Automatyzacji instalacji w języku php**
- d. Wysyłania poczty

15. Możliwą wartość typu w nagłówku Content-Type protokołu MIME nie jest:

- a. text
- b. application
- c. **graphic**
- d. audio

## Lekcje

### Lekcja 1 - Bezpieczeństwo aplikacji

**(Rodzaje ataków na portale internetowe, Testowanie poszczególnych zagrożeń, Zabezpieczanie aplikacji)**

## Cel lekcji

Celem lekcji jest poznanie rodzajów ataków na portale internetowe, opisanie możliwości testowania zagrożeń oraz podanie podstawowych technik do zabezpieczania aplikacji.

## Treść - slajdy z opisem

Slajd 1

Slide 1 is a title slide for a course. It features a green and blue vertical bar on the left. The top left contains the B2E logo (Business to Education) and the Szczeciński Park Naukowo-Technologiczny logo. The top right says 'PHP: Hypertext Preprocessor'. The center text reads 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany Lekcja 1 Bezpieczeństwo aplikacji'. At the bottom, there are logos for 'KAPITAŁ LUDZKI' and 'UNIA EUROPEJSKA' with the text 'Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego'. The DAILY GROUP logo is at the bottom left.

Słowa wstępu.

Slajd 2

Slide 2 shows an example of an attack payload. It features the same layout as slide 1. The title is 'Przykładowy atak'. The code is as follows:

```
<html>
<form>
Imie: <input type="text" name="firstname">

Nazwisko: <input type="text" name="lastname">
<input type="submit" value="Submit">
</form>
</html>
<?php
if (isset($_GET['firstname'])) {
 echo "Imie: " . $_GET['firstname'] . "
";
 echo "Nazwisko: " . $_GET['lastname'] . "
";
}
?>
```

The footer of the slide includes the DAILY GROUP logo and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego  
Wykonajmy prosty formularz HTML oraz fragment kodu PHP wyświetlający wprowadzone dane na stronie.

Slajd 3

PHP: Hypertext Preprocessor

### Przykładowy atak JS injection

Wprowadzamy zamiast imienia fragment kodu JavaScript:

```
<script>alert('Hacked');</script>
```

lub

```
<script>alert(document.cookie);</script>
```

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby atak zadziałał w pliku php.ini magic\_quotes\_gpc musi być wyłączony:  
magic\_quotes\_gpc = Off

Slajd 4

## Atak brute-force

- Najprostszy atak na systemy komputerowe.
- Wykorzystanie bazy danych
- Metody zapobiegawcze

Największą zaletą aplikacji internetowych jest ich powszechność i ogólnodostępność. Zaleta ta jednak może zamienić się w wadę, ponieważ, czym więcej osób ma do czegoś dostęp, tym większe ryzyko związane z atakami. Nowe technologie sieciowe, bądź te, które są jeszcze w fazie rozwoju (do których należy PHP), narażone są na luki we własnej implementacji, bądź ich niewłaściwe stosowanie. Jest wiele ataków na systemy komputerowe, jednak my zajmiemy się tylko częścią związaną z bezpośrednim dostępem do aplikacji, czyli atakami na portale internetowe. Podstawowymi typami ataków na portale są:

1. Przerwanie – polega na przerwaniu świadczenia usług przez serwer danej aplikacji np. DoS lub DDoS
2. Modyfikacja – wprowadzenie zmian w aplikacji przez osoby nieupoważnione do wykonania takowych zmian
3. Przechwycenie – uzyskanie kontroli nad aplikacją, bądź jej częścią lub dostępu do obszarów/zasobów, do których nie powinny mieć dostępu osoby nieuprawnione
4. Podrobienie – wprowadzenie fałszywych danych do aplikacji

Slajd 5



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is 'Atak brute-force'. Below the title is a bulleted list: 'Najprostszy atak na systemy komputerowe.', 'Wykorzystanie bazy danych', and 'Metody zapobiegawcze'. At the bottom left is the 'DAILY GROUP' logo, and at the bottom right is the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'. The slide has a decorative vertical bar on the left side, colored green and blue.

Brute – force to najprostszy rodzaj ataku na systemy komputerowe, w tym portale internetowe. Polega na znalezieniu formularza dostępu do aplikacji, konta użytkownika, bądź bazy danych, często też znalezieniu loginu i wpisywaniu wszystkich możliwych kombinacji haseł. W wersji ulepszonej z racji tego, że dla lepszego zapamiętania hasła ludzie stosują naturalne ciągi znaków jak wyrazy, korzysta się z zewnętrznej bazy danych haseł lub słowników np. imion.

Jest wiele metod zapobiegawczych tego typu atakom. Do najważniejszych zaliczamy w przypadku formularza logowania, który jest ogólnodostępny: losowe generatory haseł, określona z góry złożoność hasła, podpisywanie w aplikacji użytkownika inną nazwą, niż jego login (np. imieniem i nazwiskiem), blokowanie formularza dostępu po określonej ilości prób wpisania błędnego hasła (np. po trzech), systemy wymuszające zmianę hasła, co jakiś czas. W przypadku paneli administracyjnych oraz aplikacji do obsługi baz danych typu phpMyAdmin, oprócz wyżej wymienionych zabezpieczeń możemy głęboko ukryć link oraz przypisać określony adres ip dostępu do panelu, bądź zezwolić na dostęp tylko z hosta lokalnego, a nie z zewnątrz. Jest jeszcze wiele innych metod, na przykład autoryzacja innych adresów IP drogą mailową lub SMSesową.



B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Atak brute-force

```
password P@5sword! - failed
password P@5s#w@rd! - failed
password P@55word! - failed
password P@55w@rd! - success!
```

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Konsola przedstawiająca próby łamania hasła metodą brute-force – sprawdzanie wielu kombinacji haseł.

Screen ze strony <http://www.virtualizationadmin.com/articles-tutorials/terminal-services/security/brute-force-hacking-terminal-server-environments.html>

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Atak 'Głębokie ukrycie'

- Ukrywanie dokumentów/zasobów pod adresami, których nie jest w stanie wpisać/odgadnąć człowiek.
- Metody zapobiegawcze

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Głębokie ukrycie to termin dotyczący bezpieczeństwa dokumentów na portalach internetowych.

Umieszczamy dokument pod adresem, którego nie jest w stanie wpisać człowiek oraz nie indeksujemy go w wyszukiwarkach, w czym pomocny jest htmlowy `rel=„nofollow”` lub ustawienia pliku robots. Metoda ataku polega na znalezieniu niezabezpieczonej ścieżki i dotarciu do dokumentów.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Prostym, a mocnym zabezpieczeniem jest użycie pliku .htaccess i zabronienie dostępu osobom z zewnątrz lub bez autoryzacji po http. Możemy również zabezpieczyć dostęp z poziomu systemu operacyjnego np. dla systemów uniksowych, zmieniamy właściciela katalogu na serwer apache, który ma wszystkie uprawnienia i zabieramy wszelkie uprawnienia innym użytkownikom.

Slajd 8

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Account lockout

- Blokowanie kont w aplikacjach internetowych.
- Metody zapobiegawcze

DAILY

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak polega na blokowaniu kont w aplikacjach internetowych, dlatego używając wcześniej wymienionego zabezpieczenia, które blokuje dostęp do konta, warto zadbać o prostą i szybką możliwość odblokowania konta na przykład drogą mailową, bądź smsową, czy w przypadku banków i innych instytucji zaufania publicznego, telefoniczną lub za pomocą wydrukowanych wcześniej kodów.

Slajd 9

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Web parameter tampering

- Próba przekazania parametrów, które różnią się od tych dostępnych w aplikacji.
- Metody zapobiegawcze

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda polega na przekazywaniu metodą GET lub POST, parametrów które różnią się od tych dostępnych w aplikacji. Użytkownik, może mieć w ten sposób dostęp do funkcji lub danych, do których nie powinien go mieć. Warto więc zwrócić uwagę na to co przekazujemy ww. metodami oraz pomyśleć o szyfrowaniu parametrów, jak i ich walidowaniu. Warto też zwrócić uwagę na to w jaki sposób nasze strony komunikują się z poziomu JS -> PHP, jeżeli wykorzystujemy metody AJAX-owe oraz czy nasza aplikacja, nie wypisuje bezpośrednio tego co podajemy jako parametry, w przeciwnym wypadku możemy przekazać fragment kodu PHP, poprzez parametr i narobić szkód na portalu.

Slajd 10

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Path i information disclosure

- Próba uzyskania szczegółowych informacji o umiejscowieniu aplikacji, konfiguracji serwera itp.
- Metody zapobiegawcze

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jest to rodzaj ataku, który sam w sobie jest nieszkodliwy, aczkolwiek może przynieść wiele korzyści osobie, która próbuje się włamać na naszą stronę. Polega on na znalezieniu informacji o umiejscowieniu naszej aplikacji na serwerze, jego ustawień lub innych ważnych informacji. Informacje te można znaleźć przy błędach aplikacji oraz w obiektach konfiguracyjnych. Zapobiegać im można na wiele sposobów np. w środowisku produkcyjnym



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

wyłączamy wyświetlanie na ekran błędów aplikacji, piszemy własną obsługę błędów, debugujemy aplikację, tworzymy loadery do zasobów takich jak pliki js, obrazki, arkusze styli oraz innych dokumentów, które nie wyświetlają ścieżek do zasobów i informacji o błędach w przypadku nie istnienia zasobu.

Slajd 11

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Forced browsing

- Polega na spreparowaniu adresu url, w celu wykonania operacji, bądź uzyskania informacji, do których atakująca osoba nie powinna mieć dostępu
- Metody zapobiegawcze

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak polega na spreparowaniu adresu url, w celu wykonania operacji, bądź uzyskania informacji, do których atakująca osoba nie powinna mieć dostępu. Problem rozwiązuje odpowiednio zaimplementowany system autoryzacji lub odpowiednia architektura aplikacji np. MVC (z którego powinniśmy, korzystać w większości aplikacji). System autoryzacji powinien za każdym razem sprawdzać czy dana osoba, nie próbuje uzyskać dostępu do odpowiedniego miejsca. Aplikacja w architekturze MVC, ułatwia tę sprawę tak, że wszystko jest sterowane za pomocą odpowiedniego kontrolera, który jest dostępny tylko dla zalogowanych użytkowników.

Slajd 12

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Path traversal

- Polega na zmianie nazwy pliku, w adresie do zasobu, który zwróciła aplikacja
- Metody zapobiegawcze

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten polega na zmianie nazwy pliku, w adresie do zasobu, który zwróciła aplikacja. Zapobiegać możemy jemu na wiele sposobów, poprzez np. tworzenia osobnego katalogu (ścieżki) dla zasobu, który udostępniamy lub za pomocą szyfrowania nazw pliku, co utrudni identyfikację nazw pozostałych plików.

Slajd 13

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Unicode encoding

- Polega na przesłaniu za pomocą formularza zakodowanego kodu, który w przypadku wykonania, w którymś momencie aplikacji, może wywołać niepożądane efekty
- Metody zapobiegawcze

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten polega na przesłaniu za pomocą formularza zakodowanego kodu, który w przypadku wykonania, w którymś momencie aplikacji, może wywołać niepożądane efekty. Bronić się przed nim można przede wszystkim, odpowiednim przetwarzaniem formularza, nigdy nie pobieramy czystych danych z formularza i operujemy na nich. Zawsze trzeba je w odpowiedni sposób przetworzyć. Używajmy również walidacji treści pól formularza oraz określonej dziedziny znaków, która jest dostępna w odpowiednim miejscu np. Przy imieniu i nazwisku, nie potrzebne są cyfry oraz znaki inne, niż znaki alfabetu.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 14

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## PHP injection

- Metoda polega na wstrzyknięciu kodu PHP do aplikacji, a następnie poprzez odpowiednią manipulację aplikacją uruchomienie złośliwego kodu
- Metody zapobiegawcze

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda polega na wstrzyknięciu kodu PHP do aplikacji, a następnie poprzez odpowiednią manipulację aplikacją uruchomienie złośliwego kodu. W szczególności należy uważać przy uploadowaniu plików na serwer, zawsze trzeba walidować ich rozszerzenie oraz zawartość. Przy ich otwieraniu bezpośrednio w przeglądarce należy je przeformatować i odpowiednio wyświetlić np. jako tekst, aby przypadkiem ich nie wykonać.

Slajd 15

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## SQL injection

- Polega na wstrzyknięciu kodu zawierającego zapytanie SQL, które następnie wykona niepożądane działanie na bazie danych.
- Metody zapobiegawcze

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jest to jedna z najpopularniejszych metod ataku na serwisy internetowe, poprzez nieodpowiednie wyescapowanie treści z formularzy, bądź nieodpowiednie składanie poleceń sql. Istnieje możliwość przekazania do nich fragmentów zapytań jako parametrów, co może spowodować wykonanie nieodpowiednich akcji na bazie danych,

jak np. zmianę hasła administratora, modyfikacja parametrów konfiguracyjnych lub też usunięcie ważnych danych. Jak już wcześniej wspomniano, nie należy wstawiać czystych danych z formularzy do kodu naszej aplikacji. Zalecane jest również używanie odpowiednich obiektów, bibliotek lub adapterów, jako warstwę do komunikacji z bazą danych np. PHP Database Object w skrócie PDO, który świetnie nadaje się do użycia jako warstwa do obsługi bazy danych.

Slajd 16

Przerywnik – o co tutaj chodzi?

Zabawna implementacja ataku SQL injection. Użytkownik samochodu znając strukturę bazy danych systemu radarowego, próbuje skasować całą zawartość tabeli „TABLICE”

```
INSERT INTO Tablice
(numer_rejestracji,
mandat, kwota)
VALUES ('?', '?', '?')
```

```
INSERT INTO Tablice
(numer_rejestracji,
mandat, kwota)
VALUES ('ZU 0666', 0,
0); DROP DATABASE
TABLICE; -- '?',?)
```

Slajd 17

Atak polegający na umieszczeniu na stronie skryptu np. js, który będzie powodował niepożądane akcję przez użytkowników, np. wysyłanych ich dane logowania na inny serwer lub wykonywał inne złośliwe operacje. Stosuje się go głównie w stronach, które pozwalają użytkownikom na generowanie treści. Osadzenie skryptu w stronie HTML-owej i jego wykonanie, można wyeliminować z poziomu aplikacji poprzez np. stworzenie własnego



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

języku znaczników lub edytora graficznego do tworzenia treści, wyświetlanie treści generowanej z odpowiednim enkodowaniem lub wyszukiwaniem skryptów w zawartości treści generowanej przez użytkownika i jego wycinaniem.

Slajd 18

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### XST – Cross Site Tracing

- Wykorzystuję metodę HTTP Trace, poprzez obiekty ActiveX lub Flash. Gromadzi otrzymane w odpowiedzi nagłówki HTTP w tym nagłówek autoryzacji, który zawiera zakodowane przeważnie w base64 ciasteczko z loginem i hasłem, po czym wykorzystuje je do tworzenia własnych zapytań HTTP.
- Metody zapobiegawcze

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten wykorzystuję metodę HTTP Trace, poprzez obiekty ActiveX lub Flash. Gromadzi otrzymane w odpowiedzi nagłówki HTTP w tym nagłówek autoryzacji, który zawiera zakodowane przeważnie w base64 ciasteczko z loginem i hasłem, po czym wykorzystuje je do tworzenia własnych zapytań HTTP. Obecnie najlepszym rozwiązaniem jest nie wykorzystywanie tych technologii na stronach www. Wszystko co można zrobić we flashu, można zrobić za pomocą javascriptu i mechanizmów HTML5. A jeżeli musimy używać flasha to należy zabezpieczyć proces ładowania oraz ładować go asynchronicznie.



Slajd 19

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## XSRF – Cross Site Request Forgery

- Polega na wysłaniu żądań do serwera za pomocą osób trzecich, które są w nim autoryzowane.
- Metody zapobiegawcze

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda polega na wysłaniu żądań do serwera za pomocą osób trzecich, które są w nim autoryzowane. Osoba wykorzystuje uprawnienia osoby, pod którą się podszywa do wykonania akcji, wymagających zezwolenie tej osoby. Aby zapobiec tym atakom należy za każdym razem wylogować się z serwisu po zakończeniu korzystania z niego. Czas sesji serwera powinien być adekwatny do jej poufności. Żądania, które wymagają szczególnej ostrożności, jak np. przelewy, powinny wymagać dodatkowej autoryzacji np. via sms. Stosować dodatkowe ukryte dane przy autentykacji np. pseudolosową liczbę.

Slajd 20

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Click jacking

- Polega na przechwyceniu zdarzenia kliknięcia użytkownika oraz wykonania innej akcji niż oczekiwana przez niego w celu uzyskania poufnych informacji lub wykonania odpowiedniej akcji.
- Metody zapobiegawcze

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten polega na przechwyceniu zdarzenia kliknięcia użytkownika oraz wykonania innej akcji niż oczekiwana przez niego w celu uzyskania poufnych informacji lub wykonania odpowiedniej akcji, np. aktywacji mikrofonu, czy kamerki internetowej poprzez technologie flash. W dzisiejszych czasach wiele stron używa tzw. Likejacking'u, poprzez kliknięcia w obrazki lub inne komponenty strony, które w tajemniczy sposób lubimy, bądź udostępniamy stronę na facebooku. Zabezpieczyć się można poprzez wtyczki



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 21

**Session hijacking**

- Polega na przechwyceniu sesji serwera danej ofiary.
- Metody zapobiegawcze

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

w przeglądarce lub odpowiednią jej konfigurację.

Metoda ta polega na przechwyceniu sesji serwera danej ofiary. Możemy do tego celu użyć kodu javascript poprzez XSS, który zwróci nam ciasteczko z identyfikatorem sesji osoby zalogowanej do serwisu. Zabezpieczyć się można, korzystając z zapamiętywania adresu IP danej sesji, połączenia poprzez SSL lub enkrypcję danych przesyłanych w odpowiedziach i żądaniach. Można również zaimplementować własny mechanizm sesji, oparty np. o bazę danych.


Slajd 22

**Session fixation**


- Polega na wykorzystaniu identyfikatora sesji, uzyskanego np przy pomocy ataku session hijacking i użycie go poprzez wstawienie w odpowiednie miejsce (np adres URL czy parametr POST)
- Metody zapobiegawcze

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda ta jest metodą pochodną, poprzedniej. Po uzyskaniu identyfikatora sesji, należy znaleźć sposób na umieszczenie w odpowiednim miejscu identyfikatora sesji, czyli obejście wcześniej wspomnianych w poprzedniej metodzie zabezpieczeń, w sposób zależny od metody zabezpieczenia.



**B2E**  
BUSINESS TO EDUCATION



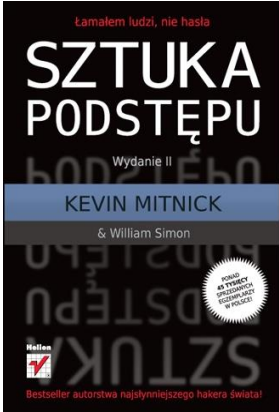
SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor


---

## Phishing

- Polega na podszyciu się pod zaufaną osobę, instytucję lub firmę, w celu wyłudzenia poufnych danych np. loginu i hasła użytkownika.
- Metody zapobiegawcze



Łamałem ludzi, nie hasła  
**SZTUKA  
PODSTĘPU**  
Wydanie II  
KEVIN MITNICK  
& William Simon  
PODROZUMIĆ  
AS FENOMEN  
OPERACJI  
W POLICJI  
Bestseller autorstwa najłynniejszego hakera świata!




**DAILY**  
GROUP


PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak polega na podszyciu się pod zaufaną osobę, instytucję lub firmę, w celu wyłudzenia poufnych danych np. loginu i hasła użytkownika. Osoba atakująca może np. podać się za firmę hostingową, po wcześniejszym sprawdzeniu dostawcy usług internetowych i poprzez odpowiedni schemat konwersacji, uzyskać odpowiednie dane. Popularność metoda ta zawdzięcza najłynniejszemu hakerowi w historii informatyki Kevinowi Mitnickowi, który jak twierdzi łamał ludzi, a nie hasła. Metoda ta bazuje na socjotechnice, czyli metodzie manipulacji osobami w kontakcie bezpośrednim. Zabezpieczenie przed nią jest oczywiste, nigdy nie wolno podawać żadnych danych poufnych osobom z zewnątrz, gdyż nawet jak same firmy często zaznaczają, nikt z obsługi o takie dane prosić nie będzie.





**B2E**  
BUSINESS TO EDUCATION

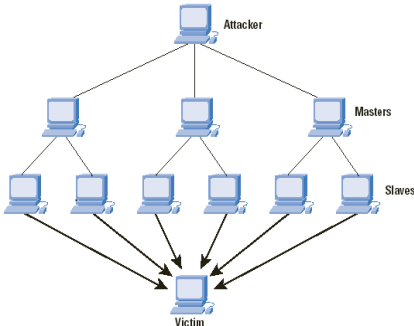


SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

## Ataki DoS i DDoS

- polegają na zablokowaniu serwera, zasypując go ogromną ilością żądań, najczęściej wykorzystuje się do tego odpowiednie oprogramowanie, np. boty lub oprogramowanie do testowania wydajności serwera.
- Metody zapobiegawcze



Źródło: www.cisco.com



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ataki DoS i DDoS, czyli odpowiednio Denial of Service i Disturbed Denial of Service, polegają na zablokowaniu serwera, zasypując go ogromną ilością żądań, najczęściej wykorzystuje się do tego odpowiednie oprogramowanie, np. boty lub oprogramowanie do testowania wydajności serwera. Serwer otrzymując ogromną liczbę żądań, nie jest w stanie przetwarzać kolejnych co powoduje wstrzymanie jego pracy lub po wyczerpaniu pamięci operacyjnej jego zawieszenie. Przerwa w pracy serwera w niektórych przypadkach, może spowodować ogromne straty finansowe. Nie ma skutecznej metody zapobiegania tej metodzie.

(grafika z [www.cisco.com](http://www.cisco.com))



## Testowanie zagrożeń


Jak testować bezpieczeństwo swojego portalu?

- Stosując metody automatyczne, np. programy analizujące znane luki (Burp site, WebScarab, w3af, Nikto Web Scanner)
- Próbując dokonać ataku samodzielnie, znając własny kod łatwiej jest przeprowadzić atak
- Śledząc nowości dotyczące znalezienia nowych luk przez społeczność zajmującą się bezpieczeństwem




Jest wiele narzędzi służących do testowania aplikacji internetowych, jedne są bardziej skuteczne, drugie mniej. Należy pamiętać o tym, że testują one tylko te ataki, które są im znane. Nie są one jednak w stanie przetestować wszystkich luk w aplikacjach. Do tego celu najlepszy jest człowiek, który jest istotą kreatywną i jest w stanie znajdować luki nikomu wcześniej nie znane oraz często publikować wynik swoich badań w internecie np. na swoim blogu. Dlatego najlepszą metodą testów jest połączenie trzech czynników: programy analizujące wszystkie znane luki, wiedza ludzi, którzy szukają nowych dziur w technologiach, aplikacjach oraz architekturach, jak i własny wkład, gdyż jako programiści, znamy najlepiej swoje aplikacje i rozumiemy ich potencjalne słabe punkty.





**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---


## Ćwiczenia praktyczne

Proszę wykonać formularz HTML zawierający pole z adresem e-mail oraz tabelę w bazie danych z kluczem głównym „id” (autoincrement int) oraz polem „adres”. Następnie przygotuj skrypt PHP, który będzie wyświetlał zawartość stworzonej tabeli, oraz umożliwił dodanie nowego adresu e-mail do bazy. Na tak przygotowanym materiale proszę wykonać atak SQL Injection tak aby:

- osoba atakująca skasowała tabele w bazie danych
- osoba atakująca podmieniła adres e-mail innego użytkownika o znanym adresie

Proponowane dodatkowe zadania związane z tym ćwiczeniem:

- Zabezpiecz skrypt przed atakiem poprzez dodanie walidacji pola formularza



DAILY  
GROUP



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Gotowe rozwiązania tego zadania znajdują się w plikach:

- skrypt.sql – skrypt tworzący tabelę w bazie danych
- sqlInjection.php – formularz HTML, połączenie do bazy danych, dodawanie adresu e-mail do bazy, wyświetlenie istniejących danych
- przykłady ataku SQL Injection – kasowanie tabeli, zmiana adresu dla istniejącego użytkownika

Do prawidłowego przetestowania tego kodu w pliku php.ini trzeba wyłączyć zabezpieczenie poprzez wyłączenie flagi `magic_quotes_gpc = Off`

Proszę upewnić się również, czy php posiada możliwość użycia `php_mssql.dll` potrzebnego do połączenia się z bazą danych (plik `php.ini`, linijka `extension=php_mssql.dll` powinna być odkomentowana).



PHP: Hypertext Preprocessor


## Ćwiczenia praktyczne

Proszę przygotować skrypt PHP, dzięki któremu możliwe będzie wykonanie ataku typu brute-force, polegające na złamaniu hasła zakodowanego algorytmem MD5. Zakodowane hasło to:

- 098f6bcd4621d373cade4e832627b4f6

Proponowane dodatkowe zadania związane z tym ćwiczeniem:




- zmień złożoność hasła i zbadaj jaki ma to wpływ na czas łamania hasła
- zmień sposób kodowania hasła np na SHA-1 i zbadaj czy takie samo hasło jest łamane w tym samym czasie dla 2 różnych algorytmów (np MD5 i SHA-1)



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Gotowe rozwiązania tego zadania znajdują się w plikach:

- brute.php – przy pomocy trudniejszej metody rekursji (funkcja wywołuje samą siebie)
- Brute1.php – przy pomocy łatwiejszej metody dla hasła 4 literowego




PHP: Hypertext Preprocessor

## Ćwiczenia praktyczne

Proszę wykonać formularz HTML zawierający dowolne pole, oraz fragment skryptu PHP odczytujący zawartość tego pola metodą GET. Następnie proszę zrealizować atak JS Injection poprzez wpisanie w pole formularza kodu JS, który wykona się w przeglądarce.

Proponowane dodatkowe zadania związane z tym ćwiczeniem:

- Zabezpiecz skrypt przed atakiem poprzez dodanie walidacji pola formularza





PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Gotowe rozwiązania tego zadania znajdują się w pliku:

- atak.php

Do prawidłowego przetestowania tego kodu w pliku php.ini trzeba wyłączyć zabezpieczenie poprzez wyłączenie flagi `magic_quotes_gpc = Off`




PHP: Hypertext Preprocessor

## Ćwiczenia praktyczne

Przy pomocy skryptu curlLoop.php wraz z kolegami z klasy, sróbuście wykonać atak DoS na komputer jednego z kolegów. Jeden komputer udostępnia dowolny zasób pod podanym adresem HTTP, pozostali uczniowie z wielu komputerów wykonują skrypt, który wielokrotnie w pętli próbuje ten zasób pobrać.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Gotowe rozwiązania tego zadania znajdują się w pliku:

- curlLoop.php

- 


Aktualnie skrypt wykonuje próbę pobrania zasobu w pętli 10 razy, można próbować manipulować tym parametrem aby bardziej dociążyć atakowany komputer.




PHP: Hypertext Preprocessor

## Dziękujemy za uwagę

Czwiek - najlepsza inwestycja




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Słowa na zakończenie.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Ćwiczenia

Ćwiczenia należy wykonać z każdym uczniem z osobna.

Ćwiczenie 1 Wymień i opisz trzy wybrane przez siebie ataki na portale internetowe.

Ćwiczenie 2 Opisz metody zapobiegania atakom na portale internetowe, dla wcześniej wymienionych rodzajów ataków.

Ćwiczenie 3 Utworzyć formularz zawierający jedno pola tekstowe oraz napisać skrypt, który dokona jego walidacji, do wyboru (przydziela nauczyciel):

- Pole z imieniem (tylko znaki alfabetu)
- Numer telefonu (cyfry spacje i znak +)
- Adres (tylko znaki alfanumeryczne)

## Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą posiadać wiedzę o popularnych atakach na portale internetowe, poznają podstawowe metody prewencji oraz będą widzieć jak testować bezpieczeństwo w swoich aplikacjach.

## Lekcja 2 - Architektura usług sieciowych

### (Klient – serwer, Peer to peer, Architektura hybrydowa)

### Cel lekcji

Celem lekcji jest wyjaśnienie podstawowych architektur aplikacji sieciowych: klient – serwer, p2p oraz architektury mieszanej. Nauczyciel opisuje i pokazuje ich schematy. Opis podstawowych architektur zorientowanych na usługi SOA. Opis jej najważniejszych implementacji SOAP, WebService, REST, DCOM, CORBA, WCF.

### Treść - slajdy z opisem

Slajd 1

Na lekcji zostaną omówione najczęściej spotykane architektury rozwiązań informatycznych, ich wady i zalety.

W kolejnej części omówione będą tematy SOA i najczęściej używane architektury usług sieciowych.

Slajd 2

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY PHP Hypertext Preprocessor

## Architektury rozwiązań informatycznych

- client – server
- peer to peer
- hybrid

DAILY GROUP Lekcja 2 – Architektura usług sieciowych

Wyróżniamy architektury:

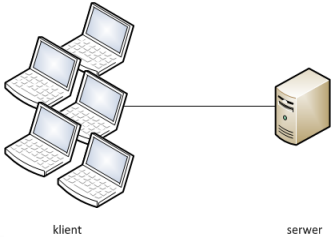
- klient - serwer - najbardziej popularna architektura oprogramowania w której klient (aplikacja kliencka) łączy się z serwerem (farmą serwerów),
- peer to peer - rozproszona architektura, której założeniem jest możliwość połączenia się wszystkich aplikacji między sobą
- hybrydowa - szerokie pojęcie np. architektura mieszana klient - serwer + peer to peer.

Slajd 3

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY PHP Hypertext Preprocessor

## Architektury rozwiązań informatycznych

- client – server



DAILY GROUP Lekcja 2 – Architektura usług sieciowych

Wyróżniamy architektury:

- klient - serwer - najbardziej popularna architektura oprogramowania w której klient (aplikacja kliencka) łączy się z serwerem (farmą serwerów),  
Przykładem jest serwer poczty elektronicznej (klient – np. outlook, serwer – exchange), serwery WWW (klient-przeglądarka, serwer-serwer WWW), serwery plików czy serwery aplikacji.

Stroną inicjującą komunikację jest klient.  
Stroną wykonującą usługę jest serwer.

W zależności od zastosowań wyróżnia się wiele typów architektury klient serwer:

- dwuwarstwowa (klient komunikuje się z serwerem np. serwer plików),
- trójwarstwowa (klient łączy się z serwerem



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

- aplikacji (warstwa druga) który przetwarza żądania, korzystając z baz danych (warstwa trzecia) lub innych zasobów odpowiada na żądanie,
- wielowarstwowa w której logika biznesowa została zaimplementowana na wielu poziomach.

Zalety: wspólne dane w jednym miejscu, zapewniona spójność i brak konfliktów wersji, łatwość zabezpieczenia danych

Wady: ograniczona wydajność (można stosować klastry wydajnościowe), awaria serwera powoduje niedostępność aplikacji/danych (stosuje się klastry niezawodnościowe)

Slajd 4

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Architektury rozwiązań informatycznych

- peer to peer
- P2P

Lekcja 2 – Architektura usług sieciowych

- peer to peer - rozproszona architektura której założeniem jest możliwość połączenia się wszystkich aplikacji między sobą
- P2P zakłada możliwość połączenia się wszystkich aplikacji między sobą. Każdy element architektury jest na tym samym równorzędym z innymi poziomami. Każdy element architektury jest zatem i klientem i serwerem. W przypadku odłączenia jednego elementu transmisja nadal jest prowadzona z wykorzystaniem zasobów innych elementów.

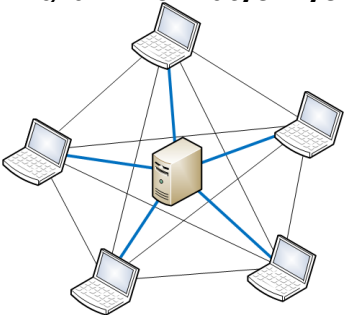
Wadą takiego rozwiązania jest konieczność przepytывania wszystkich elementów o potrzebne zasoby (np. pliki).

Slajd 5

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY PHP Hypertext Preprocessor

## Architektury rozwiązań informatycznych

- Hybrid
- P2P + server



DAILY GROUP Lekcja 2 – Architektura usług sieciowych

- hybrydowa - architektura mieszana klient-serwer + peer to peer.
- Architektura pozbawiona wady P2P, serwer pełni rolę bazy wiedzy i informacji na którym elemencie znajdują się określone zasoby. Klient nie pyta się wszystkich elementów architektury o potrzebne zasoby, zdobywa je odpytując serwer centralny. Ten w odpowiedzi kieruje klienta do właściwego hosta.

Slajd 6

B2E BUSINESS TO EDUCATION SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY PHP Hypertext Preprocessor

## Ćwiczenie

1. Narysuj i omów architekturę klient – serwer i podaj jej największe zalety/wady.
2. Czym różni się architektura peer to peer od hybrydowej?
3. Podaj przykłady architektury klient-serwer oraz P2P, które znasz z życia codziennego.

DAILY GROUP Lekcja 2 – Architektura usług sieciowych

Odpowiedzi:

- 1) zalety: wspólne dane w jednym miejscu, zapewniona spójność i brak konfliktów wersji, łatwość zabezpieczenia danych  
Wady: ograniczona wydajność (można stosować klastry wydajnościowe), awaria serwera powoduje niedostępność aplikacji/danych (stosuje się klastry niezawodnościowe),
- 2) w architekturze hybrydowej występuje serwer centralny, który pełni rolę bazy wiedzy i informacji na którym elemencie znajdują się określone zasoby
- 3) Klient-Server: Gadu-Gadu, przeglądarka+strona www, klient FTP + serwer FTP, klient poczty np. outlook lub Thunderbird + serwer pocztowy  
P2P: torrenty, Skype,



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 7

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP  
Hypertext Preprocessor

## Architektura usług sieciowych

- SOA – Service Oriented Architecture
- Usługa sieciowa

Aplikacja SOA

DAILY GROUP Lekcja 2 – Architektura usług sieciowych

SOA jest koncepcją tworzenia systemów informatycznych w której poszczególne moduły oprogramowania (aplikacje SOA) działają autonomicznie udostępniając zestaw usług do komunikacji.

Usługi implementują funkcjonalność potrzebną klientowi lub innej aplikacji SOA. Aplikacja SOA działa na zasadzie skrzynki, przykrywając szczegóły implementacji, zapewniając ściśle określoną funkcjonalność.

Usługą może być każdy, niezależnie działający program zapewniający poprzez interfejs dostęp do określonej funkcjonalności.

Slajd 8

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP  
Hypertext Preprocessor

## Architektura usług sieciowych

- SOA

Aplikacja SOA

DAILY GROUP Lekcja 2 – Architektura usług sieciowych

W szczególności, możliwe jest współdziałanie między sobą wielu aplikacji SOA (np. systemy

zintegrowane). Implementacja połączenia każdy z każdym jest bardzo skomplikowana i pracochłonna...

Slajd 9

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP Hypertext Preprocessor

## Architektura usług sieciowych

- **SOA BUS**

The diagram illustrates the SOA BUS architecture. At the top, a horizontal bar labeled 'SOA BUS' is connected by vertical lines to five separate boxes below, each labeled 'Aplikacja SOA'. Each application box contains a small icon representing a service interface. The entire architecture is set against a background with a green and blue vertical bar on the left and a grey horizontal bar at the bottom.

**DAILY** GROUP Lekcja 2 – Architektura usług sieciowych

...dlatego stosuje się rozwiązania typu ESB (Enterprise SOA Bus) zapewniające możliwość rejestracji usług każdej aplikacji SOA oraz definiowania procesów integracyjnych pomiędzy aplikacjami.

Slajd 10

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP Hypertext Preprocessor

## Implementacje SOA

- SOAP
- WebServices
- REST
- DCOM
- CORBA
- WCF

**DAILY** GROUP Lekcja 2 – Architektura usług sieciowych

Komunikacja między aplikacjami SOA i realizacja interfejsów może zostać wykonana na wiele sposobów. Przybliżymy następujące implementacje SOA:

SOAP  
WebService  
REST  
DCOM  
CORBA  
WCF



The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP Hypertext Preprocessor. The main title is 'Implementacje SOA'. Below it, 'SOAP' is listed with two bullet points: 'Protokół zdalnego dostępu' and 'Format na bazie XML'. A diagram on the right shows the SOAP-ENV structure: an outer 'Envelope' box containing an inner 'Header' box and a 'Body' box. The footer includes the DAILY GROUP logo and the text 'Lekcja 2 – Architektura usług sieciowych'.

SOAP – (Simple Object Acces Protocol) protokół służący do komunikacji między systemami wykorzystujący format XML. Dokument SOAP ma określoną strukturę i składa się z:




Koperty (envelope)

Nagłówka (header)

I ciała komunikatu (body) – w którym zawarto informację jaką usługę i metodę należy wywołać

Transmisję SOAP można realizować protokołami HTTP lub RPC

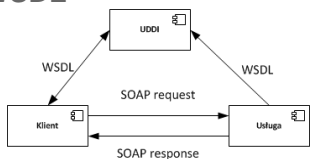







## Implementacje SOA

### WebServices

- Protokół HTTP/HTTPS
- Komunikat XML (SOAP)
- Opis interfejsu WSDL





 Lekcja 2 – Architektura usług sieciowych

Web service jest metodą komunikacji pomiędzy dwoma urządzeniami/aplikacjami w sieci z wykorzystaniem protokołu HTTP/HTTPS. Formatem komunikatu jest SOAP (Simple Object Access Protocol) – dokument XML o określonej strukturze


Jest wyposażona w mechanizm opisu usługi WSDL (Web Services Description Language). Pozwala on na opis usługi w taki sposób żeby aplikacja kliencka (klient) mógł poprawnie wczytać format zapytania i format odpowiedzi. Definicje poszczególnych usług i metod mogą być częścią usługi albo mogą zostać wyniesione do pośrednika (broker) UDDI (Universal Description Discovery and Integration).



Slajd 13



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

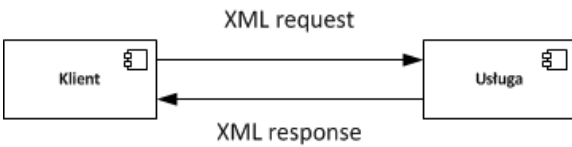
**PHP**  
Hypertext Preprocessor

---

## Implementacje SOA

### REST


- Protokół HTTP/HTTPS
- Standardowe operacje HTTP: GET, PUT, POST, DELETE
- Komunikat XML



```

graph LR
 Klient[Klient] -- XML request --> Usługa[Usługa]
 Usługa -- XML response --> Klient


```




Lekcja 2 – Architektura usług sieciowych

Odmianą Web service jest REST (Representational State Transfer). W odróżnieniu od WSDL w REST nie ma opisu formatu zapytania i odpowiedzi. Zamiast tego wykorzystywane są standardowe operacje HTTP jak GET, POST, PUT, DELETE pozwalające na realizację operacji CRUD (Create, Read, Update, Delete).

Założeniem REST jest brak stanowości zasobów, uproszczenie implementacji i zarządzania w porównaniu do WSDL.

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

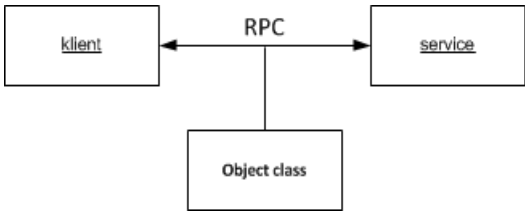
**PHP**  
Hypertext Preprocessor

---


## Implementacje SOA

### DCOM,

- Serializowany komunikat
- Microsoft
- Protokół RPC



```
graph LR; klient[klient] -- RPC --> service[service]; object_class[Object class];
```

Lekcja 2 – Architektura usług sieciowych

DCOM – (Distributed Component Object Model) - interfejs programistyczny zapewniający komunikację między aplikacjami na poziomie obiektów. Jest technologią firmy Microsoft przygotowaną jako konkurencję do CORBA. Technologia rozwinięta w .NET Framework stała się częścią COM+ powszechnie wykorzystywaną w Windows. Jest opatentowaną technologią firmy Microsoft.


Formatem komunikatu jest serializowany obiekt określonej klasy. Zatem w celu przygotowania integracji aplikacji ze sobą wymagana jest znajomość definicja obiektów jakie używane są po stronie usługi.

Protokół: RPC, później stosowano HTTP


Problemy z implementacją, komunikacją poprzez firewalle oraz ze zwalnianiem pamięci.



Slajd 15



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

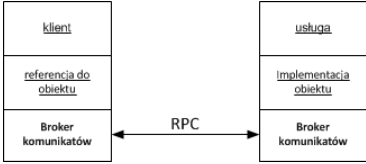
**PHP**  
Hypertext Preprocessor

---

## Implementacje SOA

### CORBA,


- Broker komunikatów
- Obiekt i mapowanie
- Protokół InterORB i jego odmiany



```

graph LR
 subgraph Klient
 K1[klient]
 K2[referencja do obiektu]
 K3[Broker komunikatów]
 end
 subgraph Usługa
 U1[usługa]
 U2[Implementacja obiektu]
 U3[Broker komunikatów]
 end
 K3 <--> |RPC| U3

```



Lekcja 2 – Architektura usług sieciowych

CORBA– (Common Object Request Broken Architecture) – standard zapewniający modułom oprogramowania napisanym w różnych językach, na różnych platformach do wzajemnej komunikacji. Standard został zdefiniowany przez OMG (Object Management Group).

CORBA wymaga zastosowania mechanizmu ORB (Object Request Broker) do wymiany komunikatów. Komunikaty muszą zostać prawidłowo zmapowane do wspólnego interfejsu (wprowadzony IDL – interface definition language) .

Protokół: InterORB, SSL InterORB, HTIOP – HyperText InterORB, Zipped IOP

Slajd 16



Slide 16 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP Hypertext Preprocessor. The main title is 'Implementacje SOA'. Below it, the text 'WCF,' is followed by a bulleted list: '• WSDL', '• Różne formy komunikacji', and '• Różne formaty komunikatu'. The footer includes the DAILY GROUP logo and the text 'Lekcja 2 – Architektura usług sieciowych'.

WCF – (Windows Communication Foundation) – nowoczesny zestaw narzędzi programistycznych na platformę Microsoft .NET pozwalających na budowę aplikacji SOA.

WCF mocno rozszerza paletę funkcjonalności SOA o np. dynamiczną adresację (routing), zapewnienie dostarczenia komunikatu oraz kontrolę uprawnień.

Usługa pozwala na wybór formy komunikacji np. SOAP HTTP, SOAP TCP, SOAP MQ.

Format komunikatu to uproszczony komunikat SOAP lub JSON (JavaScript Object Notation).

Slajd 17



Slide 17 content: The slide features the same header as slide 16. The main title is 'Ćwiczenia praktyczne - WSDL'. Below it, the text 'Przeanalizuj i omów plik HelloService.wsdl' is displayed. The footer includes the DAILY GROUP logo and the text 'Lekcja 2 – Architektura usług sieciowych'.

Opis na stronie: [http://www.tutorialspoint.com/wsdl/wsdl\\_example.htm](http://www.tutorialspoint.com/wsdl/wsdl_example.htm)



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 18

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Ćwiczenia

- Zaznacz implementacje SOA**
  - Peer2peer,
  - SOAP,
  - REST,
  - Klient-serwer,
  - PHP,
  - WebServices,
  - DCOM,
  - COBRA,
  - WCF,
  - Windows

**DAILY** GROUP Lekcja 2 – Architektura usług sieciowych

1. Gwiazdką  
zaznaczono prawidłowe  
odpowiedzi

peer2peer,  
SOAP\*,  
REST\*,  
klient-serwer,  
PHP,  
WebServices\*,  
DCOM\*,  
COBRA\*,  
WCF\*,  
Windows

Slajd 19

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor



## Ćwiczenia

### 2. Który diagram reprezentuje SOA BUS

**DAILY** GROUP Lekcja 2 – Architektura usług sieciowych

1. Odpowiedź prawidłowa to diagram w lewym górnym rogu.


Slajd 20



PHP  
Hypertext Preprocessor



## Pytania

1. Co to jest usługa sieciowa?
2. Rozwiń i omów skrót SOA.
3. Wymień 4 implementacje SOA.
4. Co to jest WSDL?
5. Omów strukturę komunikatu SOAP.
6. Na jakie formy komunikacji pozwala WCF?
7. Do czego stosuje się SOA BUS?



Lekcja 2 – Architektura usług sieciowych


Slajd 21



PHP  
Hypertext Preprocessor

## Ćwiczenia praktyczne

Proszę przygotować skrypt, który wywoła webserwis spod adresu <http://footballpool.dataaccess.eu/data/info.wso?wsdl> udostępniający informację na temat piłki nożnej. Przeanalizuj plik WSDL oraz w prawidłowy sposób wyświetl 5, 10 lub 20 najlepszych piłkarzy.



Lekcja 2 – Architektura usług sieciowych

Rozwiązanie w pliku test.php

Pomocny może być analizator plików wsdl, np <https://www.wsdl-analyzer.com/>



Slajd 22



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Ćwiczenia praktyczne

Proszę przygotować skrypt, który wywoła webserwis spod adresu <http://footballpool.dataaccess.eu/data/info.wso?wsdl> udostępniający informację na temat piłki nożnej. Przeanalizuj plik WSDL oraz w prawidłowy sposób wyświetl wszystkich piłkarzy, którzy otrzymali żółte lub czerwone kartki podczas gry na boisku.





DAILY  
GROUP

Lekcja 2 – Architektura usług sieciowych

Rozwiązanie w pliku test2.php

Pomocny może być analizator plików wsdl, np <https://www.wsdl-analyzer.com/>





**B2E**  
BUSINESS TO EDUCATION


SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Ćwiczenia praktyczne

Proszę przygotować skrypt, który wywoła webservice spod adresu <http://footballpool.dataaccess.eu/data/info.wso?wsdl> udostępniający informację na temat piłki nożnej. Przeanalizuj plik WSDL oraz stwórz formularz, który pobierał będzie informacje o ilości najlepszych piłkarzy do wyświetlenia. Wprowadź walidację, polegającą na ograniczeniu wprowadzanej wartości z zakresu od 1 do 20.



Lekcja 2 – Architektura usług sieciowych

Rozwiązanie w pliku index.php

Pomocny może być analizator plików wsdl, np <https://www.wsdl-analyzer.com/>



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Ćwiczenia praktyczne

Proszę przygotować skrypt, który wywoła webserwis spod adresu <http://footballpool.dataaccess.eu/data/info.wso?wsdl> udostępniający informację na temat piłki nożnej. Przeanalizuj plik WSDL oraz wybierz samodzielnie jakie informacje z podanego webserwisu chcesz pobrać. W odpowiedni sposób wywołaj wybraną metodę zwracając uwagę na listę parametrów, które ta metoda przyjmuje.



DAILY  
GROUP

Lekcja 2 – Architektura usług sieciowych

Pomocny może być analizator plików wsdl, np <https://www.wsdl-analyzer.com/>



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Ćwiczenia praktyczne

Przygotuj REST API bazując na frameworku SLIM (<http://www.slimframework.com/>). Stwórz dowolną tabelę w bazie danych, pobierz dane, oraz udostępnij przy pomocy stworzonego przez Ciebie API. (Np. Pobierz listę wszystkich adresów e-mail oraz udostępnij je w lokalizacji <http://localhost/api/email/>)



Lekcja 2 – Architektura usług sieciowych



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Ćwiczenia praktyczne

Przetestuj przygotowane we wcześniejszym ćwiczeniu API przy pomocy narzędzia „Advanced REST client” dostępnego w przeglądarce Chrome.



Advanced REST client  
Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA  
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



Lekcja 2 – Architektura usług sieciowych

Advanced REST client trzeba wcześniej w Chrome zainstalować. Dodatek ten nie jest domyślnie dostępny.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Ćwiczenia

Ćwiczenie zostało przedstawione na slajdzie 6, 17, 18, 19. Pytania na slajdzie 20.

## Opis założonych osiągnięć ucznia

Uczeń powinien znać i potrafić omówić (narysować) podstawowe architektury aplikacji sieciowych oraz posiadać wiedzę na temat podanych implementacji architektury SOA.

## Lekcja 3 - Protokół http

### (budowa protokołu HTTP 1.1, nagłówki żądania i odpowiedzi, komunikacja http)

### Cel lekcji

Celem lekcji jest zapoznanie ucznia z protokołem HTTP, a w szczególności jego wersji 1.1. Pokazanie schematu tworzenia żądań i rozumienia odpowiedzi od serwera. Poznanie metod protokołu HTTP oraz sposobu komunikacji pomiędzy klientem i serwerem. Zastosowanie podstawowych metod \$\_GET oraz \$\_POST w języku PHP.

### Treść - slajdy z opisem

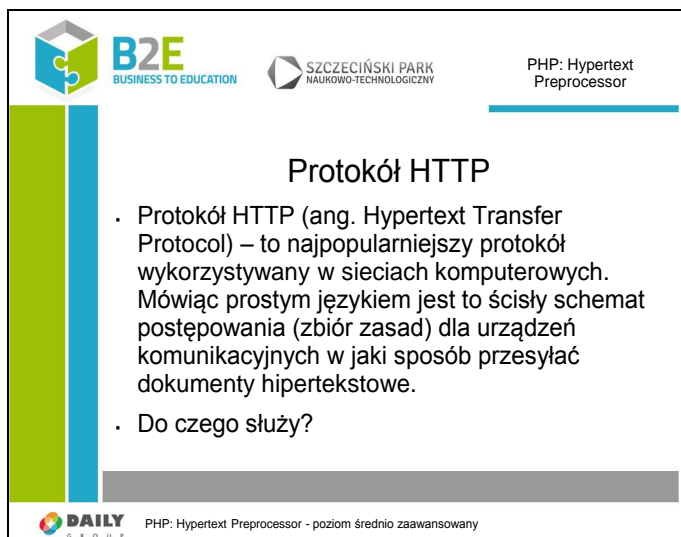
Slajd 1

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor  
- poziom średnio zaawansowany  
Lekcja 3  
Protokół http

Słowa wstępu.

## Slajd 2



Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

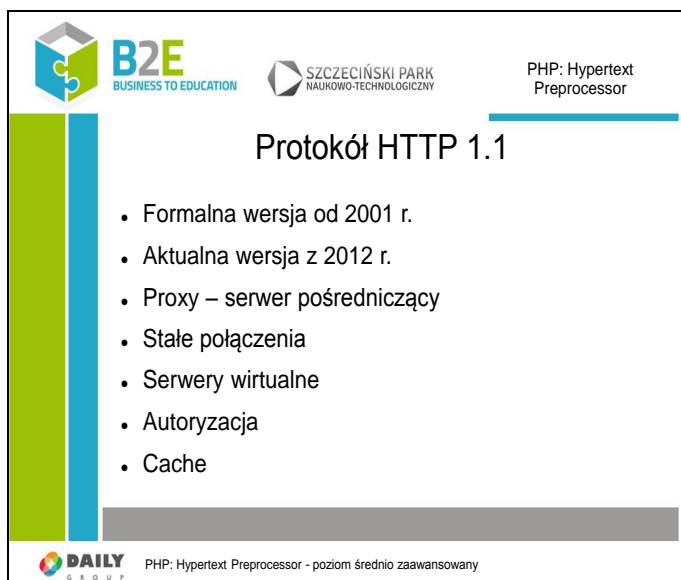
### Protokół HTTP

- Protokół HTTP (ang. Hypertext Transfer Protocol) – to najpopularniejszy protokół wykorzystywany w sieciach komputerowych. Mówiąc prostym językiem jest to ścisły schemat postępowania (zbiór zasad) dla urządzeń komunikacyjnych w jaki sposób przesyłać dokumenty hipertekstowe.
- Do czego służy?

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Protokół HTTP (ang. Hypertext Transfer Protocol) – to najpopularniejszy protokół wykorzystywany w sieciach komputerowych. Mówiąc prostym językiem jest to ścisły schemat postępowania (zbiór zasad) dla urządzeń komunikacyjnych w jaki sposób przesyłać dokumenty hipertekstowe. Rozpowszechniony dzięki sieci WWW (ang. World Wide Web), za jego pomocą odbywa się przesyłanie danych oraz komunikacja w większości aplikacji internetowych.

## Slajd 3



Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Protokół HTTP 1.1

- Formalna wersja od 2001 r.
- Aktualna wersja z 2012 r.
- Proxy – serwer pośredniczący
- Stałe połączenia
- Serwery wirtualne
- Autoryzacja
- Cache

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany




Protokół HTTP w wersji 1.1 powstał w 2001 r. Obecny standard pochodzi z grudnia 2012 r.

Obecna wersja wprowadziła szereg zmian w stosunku do poprzedniej. Możemy wykorzystać protokół jako protokół pośredniczący np. pomiędzy klientem, a innym protokołem np. SMTP albo FTP. Dzięki protokołowi TCP/IP najnowszy standard wspiera stałe połączenia z serwerem. Możliwa jest również obsługa serwerów wirtualnych. Stworzono mechanizm autoryzacji poprzez HTTP oraz mechanizm Cache'owania czyli przechowywania poprzednich wyników.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4

PHP: Hypertext Preprocessor




### Sposób działania

- Wykorzystywany rodzaj architektury
- Bezstanowość protokołu
- Mechanizmy sesji i ciasteczek

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

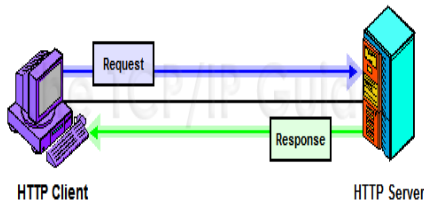
Protokół HTTP wykorzystuje architekturę klient – serwer, klient (w tym przypadku przeglądarka) wysyła żądanie do serwera, który wysyła odpowiedź w zależności od otrzymanej wcześniej treści. Standaryzują on formę zapytań oraz odpowiedzi. W związku z tym, że protokół nie zapamiętuje informacji o wcześniejszych transakcjach (transakcja = żądanie + odpowiedź), określa się go mianem protokołu bezstanowego. Brak możliwości przechowywania informacji jest swoistą przeszkodą, jeśli chodzi o komunikację w sieci (np. autentykacja), dlatego istnieją inne mechanizmy przechowujące informacje ('cookies' – po stronie klienta, 'session' po stronie serwera).

Slajd 5

PHP: Hypertext Preprocessor

### Podstawowy schemat działania protokołu HTTP





HTTP Client      HTTP Server

Request


Response

PHP: Hypertext Preprocessor - poziom średnio zaawansowany



 PHP: Hypertext Preprocessor

## Opis podstawowego schematu działania protokołu HTTP

1. Nawiązanie połączenia
2. Wystanie żądania HTTP
3. Otrzymanie odpowiedzi

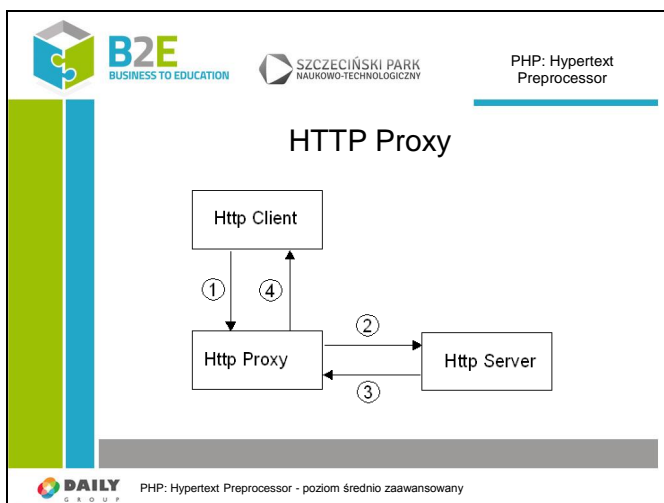

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

1. Nawiązanie połączenia – w pierwszej kolejności klient używając adresu serwera (URL lub IP), nawiązuje połączenie najczęściej za pomocą protokołu TCP/IP. Domyślny port 80.

2. Następnie klient wysyła żądanie HTTP do serwera, które jest ciągiem znaków ASCII zakończona znakami „CR LF”

3. Ostatnim krokiem komunikacji jest otrzymanie odpowiedzi od serwera w postaci ciągu znaków w określonym standardzie np. HTML

Istnieje również bezpieczna wersja protokołu (HTTPS), która działa na porcie 443, a jej transmisja jest szyfrowana.





Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 8

Logo: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Opis działania HTTP Proxy

1. Nawiązanie połączenia z serwerem Proxy i wysłanie żądania HTTP
2. Translacja adresu i przekazanie żądania
3. Przekazanie odpowiedzi do serwera proxy
4. Otrzymanie odpowiedzi

Logo: DAILY GROUP, PHP: Hypertext Preprocessor - poziom średnio zaawansowany

1. Nawiązanie połączenia – w pierwszej kolejności klient używając adresu serwera( URL lub IP), nawiązuje połączenie najczęściej za pomocą protokołu TCP/IP. Domyślny port 80. Następnie wysyła żądanie HTTP do serwera, które jest ciągiem znaków ASCII zakończona znakami „CR LF”

2. Translacja adresu zewnętrznego na adres wewnętrzny przez serwer Proxy. Przekazanie żądania HTTP z serwera Proxy do wewnętrznego serwera HTTP.

3. Wysłanie odpowiedzi przez serwer wewnętrzny do serwera proxy.

4. Ostatnim krokiem komunikacji jest otrzymanie odpowiedzi od serwera proxy w postaci ciągu znaków w określonym standardzie np. HTML

Slajd 9

Logo: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Przykładowe żądanie HTTP (metoda POST)



```
POST /enlighten/calais.asmx/Enlighten HTTP/1.1
Host: api.opencalais.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

licenseID=string&content=string¶msXML=string
```

Logo: DAILY GROUP, PHP: Hypertext Preprocessor - poziom średnio zaawansowany



Slajd 10




PHP: Hypertext Preprocessor

### Przykładowa odpowiedź HTTP (XML)



```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://clearforest.com/">string</string>
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Slajd 11



PHP: Hypertext Preprocessor



### Ćwiczenie

Wykonać skrypty get.php oraz post.php. Przeanalizować zwracaną zawartość.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Slajd 12



PHP: Hypertext Preprocessor

### Metody HTTP

- GET
- HEAD
- POST
- DELETE
- TRACE
- PUT
- OPTIONS



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metody służą do komunikacji z serwerem poprzez protokół HTTP. Najpopularniejsze metody wykorzystywane w skryptach PHP to POST oraz GET. Są one dostępne w podstawowych składnikach aplikacji jak tablice superglobalne, formularze oraz adres URI (ang. Uniform Resource Identifier). HTTP posiada jednak więcej metod, którymi możemy się w odpowiedni sposób posłużyć.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13

Metoda GET służy do pobierania zasobu wskazanego przez URI, może mieć postać warunkową jeśli w nagłówku występują pola warunkowe takie jak "If-Modified-Since,,

Na zdjęciu przedstawiono przykładowy adres z dwoma parametrami GET. Zmienna locale przyjmuje wartość en, natomiast zmienna app wartość forestkit.

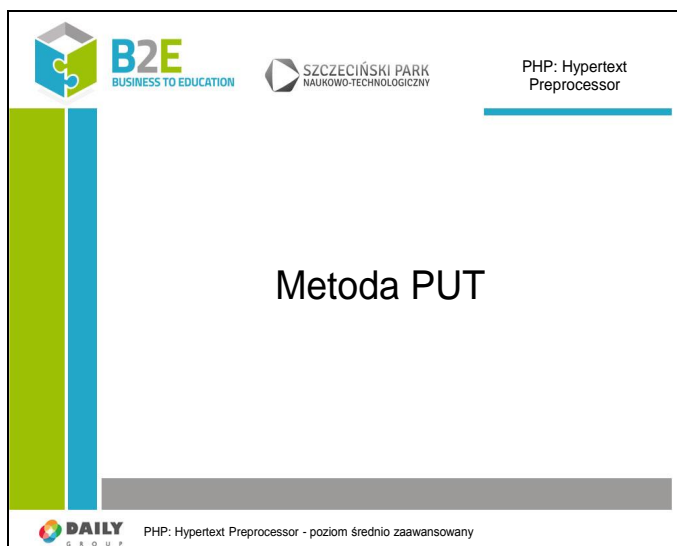
Metoda GET posiada ograniczenia dotyczące ilości użytych w adresie znaków. Ograniczenie to zależy od konfiguracji serwera oraz użytej przeglądarki. W przypadku wystąpienia zbyt długiego adresu, serwer powinien zwrócić błąd 414 (Request-URI Too Long).

Slajd 14

Metoda ta służy do przyjmowania danych przesłanych przez klienta np. podczas przetwarzania formularzy.

Metodę tę używamy również do przesyłania plików (przykład w materiałach dołączonych do lekcji, plik upload.html)

Slajd 15



Metoda PUT służy do przesyłania danych w postaci pliku od klienta do serwera

Slajd 16



Metoda DELETE wykorzystywana jest jako żądanie usunięcia zasobu z serwera, ze względów bezpieczeństwa poprzedzona jest autoryzacją



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 17

B2E  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

Metoda OPTIONS

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda OPTIONS służy do pobierania informacji o opcjach na danym kanale komunikacyjnym dla zarejestrowanego użytkownika

Slajd 18

B2E  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

Metoda HEAD

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda HEAD służy do pobierania informacji o zasobie, sprawdzania jego dostępności

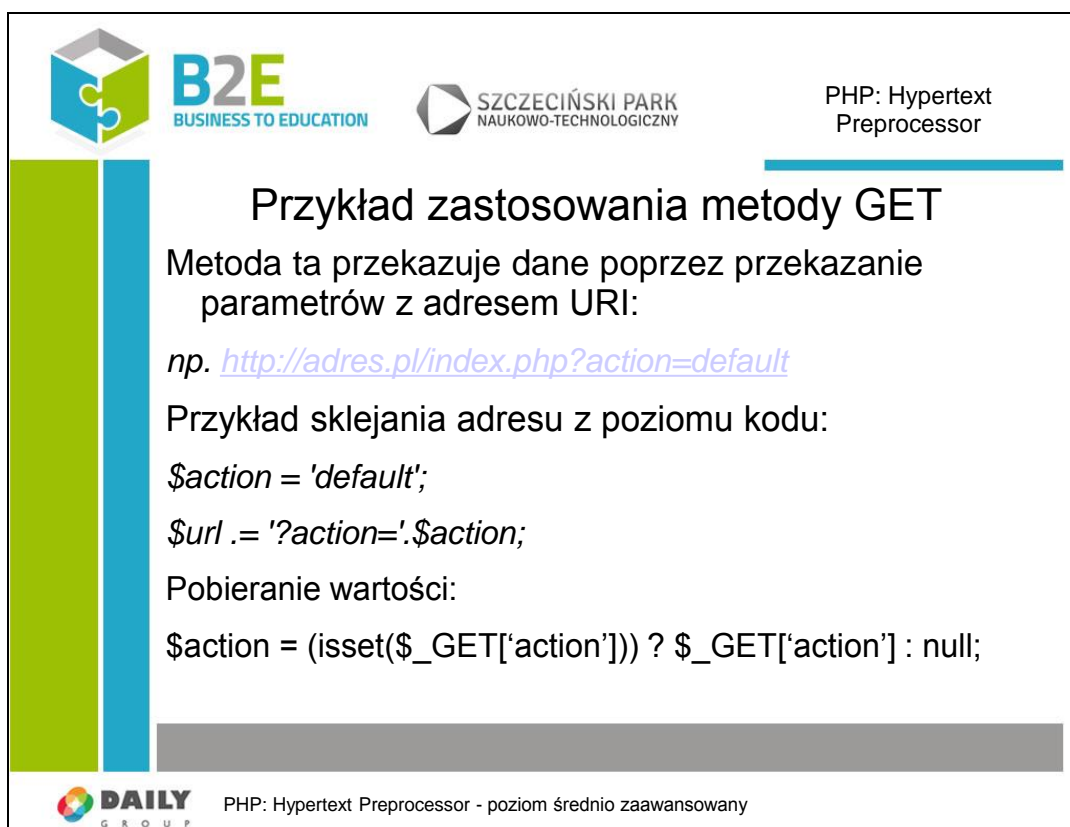
Slajd 19



Slide 19 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is 'Metoda TRACE'. The footer includes the DAILY GROUP logo and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Metoda TRACE służy do diagnostyki, analizy kanału komunikacyjnego.

Slajd 20



Slide 20 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is 'Przykład zastosowania metody GET'. The text describes the method and provides an example URL and PHP code snippets. The footer includes the DAILY GROUP logo and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

### Przykład zastosowania metody GET

Metoda ta przekazuje dane poprzez przekazanie parametrów z adresem URI:

np. <http://adres.pl/index.php?action=default>

Przykład sklejenia adresu z poziomu kodu:

```
$action = 'default';
$url .= '?action='.$action;
```

Pobieranie wartości:

```
$action = (isset($_GET['action'])) ? $_GET['action'] : null;
```

Podstawowy mechanizm w języku PHP, zastosowania żądań typu GET oraz jednocześnie najłatwiejsza metoda skomunikowania ze sobą skryptów na serwerze HTTP. Wykorzystujemy ją dodając parametry do adresu skryptu.

Pobieranie wartości przy pomocy kodu:



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


```
$action = (isset($_GET['action'])) ? $_GET['action'] : null;
```

jest równoznaczne z:


```
if(isset($_GET['action'])) {
$action = $_GET['action'];
}
else {
$action = null;
}
```

I polega na sprawdzeniu czy zmienna action istnieje w tablicy \$\_GET (czy istnieje w adresie). Jeśli tak – pobieramy jej wartość, jeśli nie – ustawiamy na null.

Slajd 21



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład zastosowania metody POST


Metoda najczęściej wykorzystywana przy przesyłaniu formularzy:

Setter:

```
<form action="adres_skryptu" method="post">
<input name="filed_name" type="text" />
</form>
```

Getter:

```
$form_data = $_POST['field_name']
```

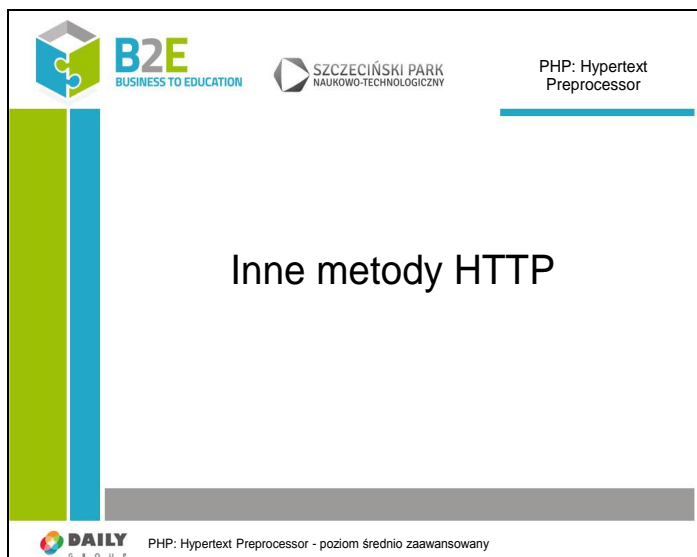


DAILY  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Drugi podstawowy mechanizm komunikacji dostępny w języku PHP, najczęściej stosowany do przetwarzania formularzy lub asynchronicznych zapytań HTTP, czyli technologii AJAX. W przeciwieństwie do metody GET, której limit długości adresu ustalany jest przez przeglądarkę klienta. Limit długości tablicy POST ustalany jest przez konfigurację serwera. Metoda POST służy również do przesyłania plików w języku PHP.

Slajd 22

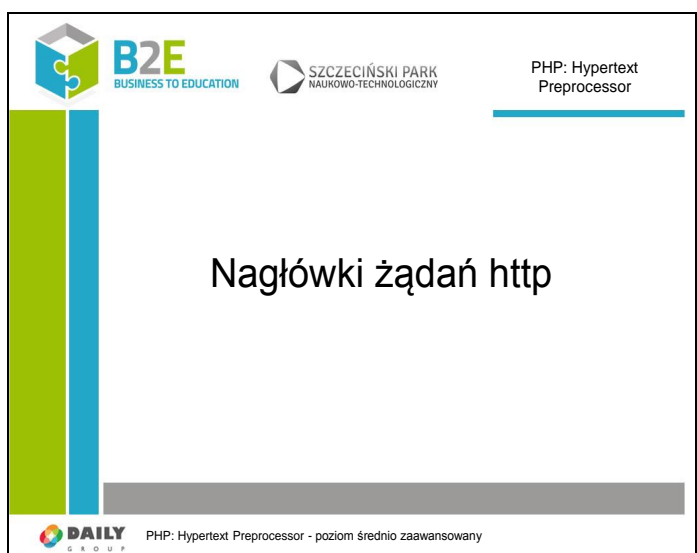


Poza standardowymi metodami jak GET i POST dostępnymi w samym języku PHP, możemy tworzyć własne zapytania HTTP korzystając przykładowo z rozszerzenia HTTP języka PHP lub zdefiniowanych przez nas obiektów, bądź funkcji. np. biblioteka cURL wspiera przesyłanie plików metodą PUT. Istnieją również architektury, które ułatwiają komunikację z serwerem, udostępniając odpowiednie API, jak architektura REST, SOAP czy Web Service.

Biblioteka cURL w domyślnej konfiguracji serwera jest wyłączona, aby ją uruchomić należy w pliku php.ini odkomentować linijkę: `extension=php_curl.dll`

Po wprowadzeniu zmiany należy zrestartować serwer apache.

Slajd 23



Żądanie HTTP realizuje się poprzez przesłanie odpowiednich nagłówków oraz parametrów do serwera, zakończonych znakami powrotu karetki oraz nowej linii. HTTP 1.1 pozwala na wysyłanie potokiem kilku żądań naraz. Nagłówki służą jako parametry żądania np. w jakim formacie wysyłamy żądanie i w jakim formacie odpowiedzi się spodziewamy.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 24

**Budowa żądania HTTP**

1. Nazwa metody oraz standard np. *GET / HTTP/1.1*
2. Nagłówki wymagane np. *Host: jakishost.pl*
3. Nagłówki opcjonalne np. *Accept: text/xml*
4. Dane np.
5. *CR + LF*

PHP: Hypertext Preprocessor

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

- W pierwszej kolejności wysyłamy nazwę metody, która w naszym przykładzie jest to metoda GET oraz nazwa standardu z jakim nasza prośba ma być przetwarzana.
- Następnie wysyłamy nagłówki wymagane przez standard w tym przypadku HOST, który jest wymagany ze względu na możliwość stworzenia Wirtualnych Hostów w obrębie jednego adresu IP.
- W dalszej części znajdują się nagłówki opcjonalne w przykładzie rodzaj akceptowalnych przez klienta danych.
- Jeżeli metoda przesyła dane np. POST umieszczamy je w tym właśnie miejscu.
- Kończymy żądanie znakiem powrotu karetki oraz znakiem przejścia do nowej linii.
- Pierwszym elementem jest kod odpowiedzi w tym przypadku wszystko jest prawidłowo, żądanie zostało zaakceptowane i zwrócone zostały dane.
- Nagłówki odpowiedzi w przykładzie czas serwera.
- Zakończenie nagłówków odpowiedzi.
- W tym miejscu przekazane są dane zwrócone z serwera HTTP.

Slajd 25


**Budowa odpowiedzi HTTP**

1. Kod odpowiedzi HTTP *HTTP/1.1 200 OK*
2. Nagłówki wymagane np. *Date: Thu, 20 Dec 2001 12:04:30 GMT*
3. *CR + LF*
4. Zwrócone dane


PHP: Hypertext Preprocessor

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany





**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---

## Ćwiczenia praktyczne


Przy pomocy cURLa pobierz dowolną stronę internetową, następnie zmodyfikuj nagłówek odpowiedzi i wyświetl całość jako tekst (text/plain).




**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

curl.php



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

---

## Ćwiczenia praktyczne

Stwórz skrypt, który wywoła z dowolnymi zmiennymi i wartościami w polu adresu, a następnie wyświetl całą tablicę \$\_GET.  
Przykładowy adres do wywołania:

- <http://localhost/Lekcja3/get.php?imie=Adam&nazwisko=slodowy>



**DAILY**  
GROUP



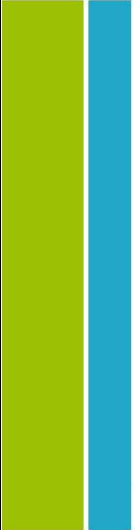
PHP: Hypertext Preprocessor - poziom średnio zaawansowany

get.php



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


Slajd 28



PHP: Hypertext Preprocessor

## Ćwiczenia praktyczne



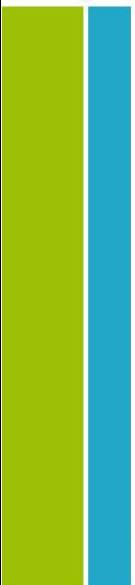
Stwórz skrypt, który zawiera formularz HTML z polami imienia oraz nazwiskiem. Prześlij formularz przy pomocy metody POST a następnie wyświetl zawartość odpowiednich elementów tablicy \$\_POST.



DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

post.php


Slajd 29



PHP: Hypertext Preprocessor

## Ćwiczenia praktyczne

Stwórz skrypt, który będzie wysyłał pliki na serwer i umieszczał je w folderze „upload”.



DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

upload.php

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Ćwiczenia praktyczne

Do stworzonego wcześniej skryptu wysyłającego pliki na serwer, dodaj ograniczenie polegające na możliwości wysłania:

- Tylko plików graficznych
- Plików nie większych niż 20000 bajtów
- Sprawdzanie plików dokonaj po rozszerzeniu (\*.jpg, \*.gif itd...) oraz mime-type pliku (image/gif, image/jpeg ...)

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

upload.php

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Dziękujemy za uwagę

Czaszek - najlepsza inwestycja

**KAPITAŁ LUDZKI**  
INWESTYCJA W PRZODKOWNIKÓW

**UNIA EUROPEJSKA**  
ROZWOJ I WYKONANIE

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

### Ćwiczenia

Wykonać ćwiczenia ze slajdu 11. Dodatkowo można zlecić uczniom napisanie przykładów użycia innych metod HTTP znajdujących się na wybranych slajdach.

### Opis założonych osiągnięć ucznia

Uczeń powinien poznać i umieć opowiedzieć o metodach protokołu HTTP, potrafił wyjaśnić na czym polega komunikacja pomiędzy klientem, a serwerem. Powinien umieć opowiedzieć z czego składa się żądanie oraz odpowiedź HTTP.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Lekcja 4 - Modyfikacja nagłówków http

### (Lista nagłówków, Wykorzystanie funkcji header, Zastosowania w skryptach PHP)

#### Cel lekcji

Celem lekcji jest nauczenie uczestników nagłówków http. Powinni umieć używać ich w języku php przy użyciu funkcji header.

#### Treść - slajdy z opisem



Slajd 1

Słowa wstępu.

Slajd 2

Nagłówki HTTP w języku PHP, wykorzystywane są do komunikacji pomiędzy serwerem, a stroną WWW.

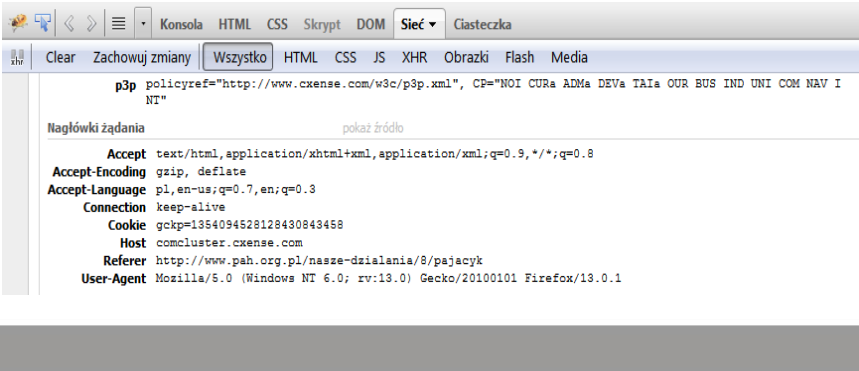
Do ich wysyłania najczęściej stosujemy funkcję 'header'. Używać ich może w różnych celach np. do autoryzacji HTTP, odświeżenia strony, bądź przekierowania przez serwer na inną stronę.

PHP: Hypertext Preprocessor

## Ćwiczenie

Otwórz dowolną stronę www i przeanalizuj nagłówki HTTP przy pomocy narzędzia firebug.



**Nagłówki żądania**

```

Accept text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding gzip, deflate
Accept-Language pl,en-us;q=0.7,en;q=0.3
Connection keep-alive
Cookie gckp=1354094528128430843458
Host comcluster.cxense.com
Referer http://www.pah.org.pl/nasze-dzialania/8/pajacyk
User-Agent Mozilla/5.0 (Windows NT 6.0; rv:13.0) Gecko/20100101 Firefox/13.0.1

```

**DAILY GROUP** PHP: Hypertext Preprocessor - poziom średnio zaawansowany




PHP: Hypertext Preprocessor

## Dostępne nagłówki

- . Standardowe
- . Niestandardowe

**DAILY GROUP** PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nagłówki HTTP w języku PHP, dzielimy na standardowe i niestandardowe. Standardowe to te, które znajdują się w specyfikacji HTTP w wersji 1.1, a niestandardowe to takie nagłówki, które są dostępne, lecz nie znajdują się w specyfikacji.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 5

Nagłówki HTTP w języku PHP, wykorzystywane są do komunikacji pomiędzy serwerem, a stroną WWW.

Do ich wysyłania najczęściej stosujemy funkcję 'header'. Używać ich może w różnych celach np. do autoryzacji HTTP, odświeżenia strony, bądź przekierowania przez serwer na inną stronę.

Slajd 6

Służy do określenia listy akceptowalnych przez przeglądarkę typów MIME dokumentu, oraz opcjonalnie hierarchii każdego typu. Liczby podane po ;q= powinny mieć wartości od 0 do 1, co 0.1. Jako separator miejsc dziesiętnych użyta musi być kropka. W przypadku braku zdefiniowanej wartości ;q= przyjmowana jest wartość: ;q=1 (równoznaczna z ;q=1.0).

Powyższy przykład ustala najwyższy priorytet typowi application/xhtml+xml. Jeżeli serwer nie posiada dokumentu w formacie application/xhtml+xml, to prześle plik w formacie application/xml – ogólnym dla podjęzyków XML, takich jak XHTML, SVG. Jeżeli na serwerze

nie ma również dokumentu w tym formacie, dane zostaną przesłane jako text/xml – czyli ogólne dane w formacie XML, w ostateczności, jeżeli żadnego z tych typów nie posiada, serwer wyśle dane w formacie text/plain, czyli zwykłego tekstu.

Typy MIME mogą być określone za pomocą symboli wieloznacznych np.: text/\*, \*/\*.

Slajd 7

**Nagłówek Accept-Charset**

**Accept-Charset:** utf-8, iso-8859-13;q=0.8

Określa preferowane przez przeglądarkę formaty kodowania. Wartości ;q= interpretowane są na takich samych zasadach, jak w przypadku typów MIME z nagłówka Accept.

Slajd 8

**Nagłówek Accept-Encoding**

**Accept-Encoding:** gzip;q=1.0, identity; q=0.5, \*;q=0

Ten nagłówek określa kodowanie, za pomocą którego zostanie przesłana zawartość. Stosowane zwykle do kompresji treści.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 9

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Nagłówek Accept-Language

**Accept-Language:** pl, en-us;q=0.7

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa w jakim języku użytkownik przeglądarki życzy sobie czytać strony.

Slajd 10

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Nagłówek Accept-Ranges

**Accept-Ranges:** bytes *lub* none

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ten nagłówek jest ustawiany przez przeglądarki i programy ułatwiające ściąganie plików. Określa czy klient potrafi (wartość bytes) lub nie (wartość none) odczytywać pliki przesyłane w częściach.

Slajd 11

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Nagłówek Allow

**Allow:** GET, POST, HEAD

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Lista metod na które serwer ma zezwolić.

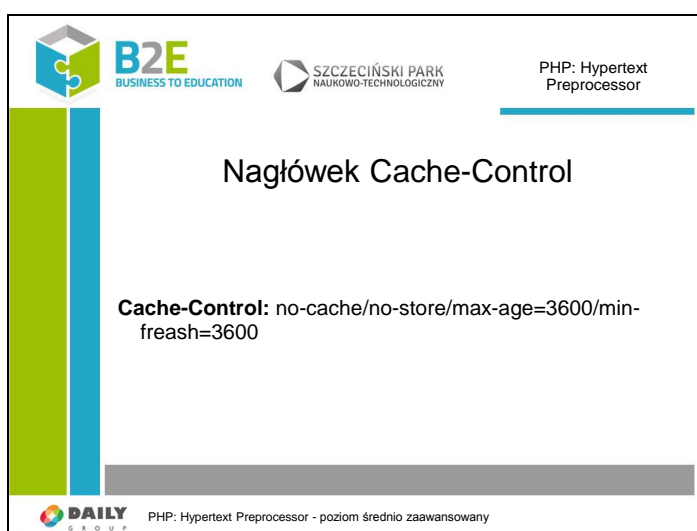


Slajd 12



Przesyła nagłówek do autoryzacji, najczęściej zakodowane login i hasło.

Slajd 13



Określa regułę według której przeglądarka może (lub nie może) przechowywać stronę w pamięci podręcznej. Dostępne wartości to:

- no-store – strona nie może być przechowywana.
- no-cache – strona może być przechowywana pod warunkiem każdorazowego sprawdzania jej aktualności. Te dwa nagłówki mają szczególne znaczenie dla serwerów proxy. Pierwszy nakazuje przesyłać dokument bezwzględnie, natomiast drugi pozwala na skorzystanie z kopii w pamięci podręcznej, jeżeli dokument źródłowy się nie zmienił.
- max-age=liczba\_sekund – oznacza, że można skorzystać z cache pod warunkiem iż nie jest przechowywane dłużej jak liczba\_sekund. Jeżeli jest starsze przeglądarka powinna ściągnąć nowszą wersję.
- min-fresh=liczba\_sekund – nakazuje przeglądarce ściągnąć dokument jeżeli jego wiek nie przekracza



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 14

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Connection

Connection: keep-alive/close

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

liczby\_sekund +  
aktualna\_data  
Nagłówek określa, czy  
zamknąć połączenie po  
zapytaniu, czy je  
utrzymywać.

Slajd 15

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Cookie

Cookie: ciastko1=wartosc1; ciastko2=wartosc2

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W tym nagłówku przesyłane  
są wszystkie ciasteczka  
ustawione w przeglądarce.

Slajd 16

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Content-Encoding

**Content-Encoding:** gzip

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa typ kodowania pliku.

Slajd 17

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Content-Language

**Content-Language:** en, pl

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Język dokumentu przesłanego przez serwer. Może być określony więcej niż jeden język. Wtedy język główny strony jest określony jako ostatni na tej liście.

Slajd 18

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Content-Length

**Content-Length:** 1000

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa rozmiar zawartości w bajtach



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 19

**Nagłówek Content-Range**

**Content-Range:** bytes 123-456/5000

PHP: Hypertext Preprocessor

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa zakres przesłanej części pliku w tym przypadku bajty od 123 do 456 z 5000(cały plik).

Slajd 20

**Nagłówek Content-Type**

**Content-Type:** application/xhtml+xml; charset=utf-8

PHP: Hypertext Preprocessor

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Tym nagłówkiem serwer informuje przeglądarkę, w jakim formacie i stronie kodowej wysłany jest dokument.

Slajd 21

**Nagłówek Date**

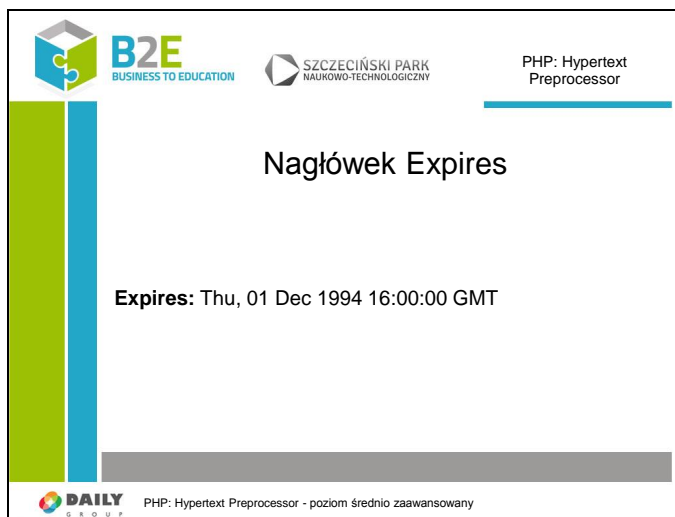
**Date:** Tue, 15 Nov 1994 08:12:31 GMT

PHP: Hypertext Preprocessor

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Data serwera.

Slajd 22



Slide 22 features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main content area displays the title "Nagłówek Expires" and the value "Expires: Thu, 01 Dec 1994 16:00:00 GMT". A footer contains the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Data powyżej której dokument będzie nieaktualny.

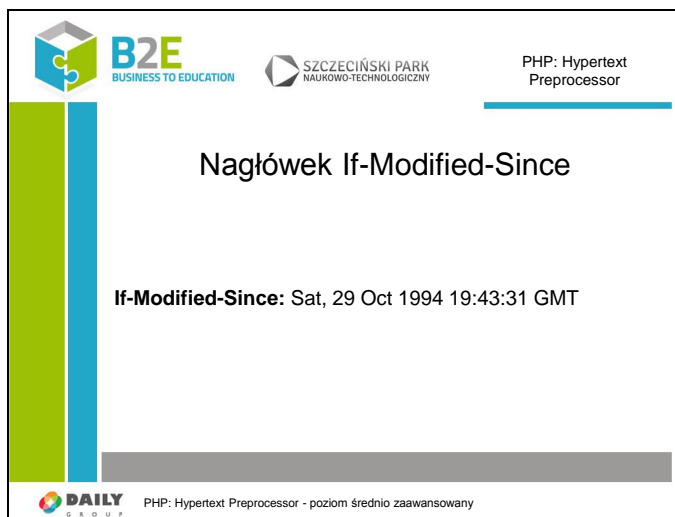
Slajd 23



Slide 23 features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main content area displays the title "Nagłówek Host" and the value "Host: www.w3.org". A footer contains the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Nagłówek określający z/dla jakiej domeny określony został URI (do serwera bowiem nie dociera nazwa domenowa, ale jedynie adres IP). Nagłówek wymagany w HTTP 1.1. Dzięki niemu można uruchomić kilka serwerów wirtualnych na jednym adresie IP.

Slajd 24



Slide 24 features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main content area displays the title "Nagłówek If-Modified-Since" and the value "If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT". A footer contains the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Nagłówek nakazuje serwerowi przesłać dokument tylko jeżeli został zmodyfikowany od danej daty. Jeżeli dokument się nie zmienił serwer powinien wysłać przeglądarce kod odpowiedzi 304.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 25

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main content area displays the text 'Nagłówek Last-Modified' and 'Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT'. A footer contains the DAILY GROUP logo and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Nagłówek serwera informujący o ostatniej aktualizacji dokumentu.

Slajd 26

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main content area displays the text 'Nagłówek Location' and 'Location: http://serwer/'. A footer contains the DAILY GROUP logo and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Wymusza przekierowanie na określony adres.

Dla kodu odpowiedzi 201 adres oznacza miejsce nowo utworzonego zasobu.

Kod 301 - oznacza przekierowanie stałe. Od tej pory przeglądarka powinna korzystać z nowego adresu.

Kod 304 - oznacza, że strona nie została zmodyfikowana od ostatniego zgłoszonego pobrania.

Kod 307 – oznacza czasowe przekierowani

Specyfikacja odpowiedzi HTTP:

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Slajd 27

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Range

**Range:** bytes=500-999

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa jaką część pliku oczekuje przeglądarka. Podawany w formacie bytes=pozycja\_początkowa-  
pozycja\_końcowa

Slajd 28

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Referer

**Referer:** http://serwer/katalog/plik.html

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa adres strony internetowej, z której użytkownik został przekierowany za pomocą odnośnika.

Przykład: film z YouTube został umieszczony na FaceBook'u, po kliknięciu w adres filmu serwer YouTube otrzymuje w nagłówku informację, że wejście na film odbyło się z serwisu FaceBook.

Slajd 29

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Refresh

**Refresh:** czas; url=adres

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ustawia automatyczne przekierowanie w przeglądarce na podany adres po określonym czasie



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 30

Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are at the top. The slide title is "Nagłówek Retry-After". The main content is "Retry-After: Fri, 31 Dec 1999 23:59:59 GMT". The footer includes the DAILY GROUP logo and "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Używany razem z kodem odpowiedzi 503 (Serwer zajęty). Określa czas po którym serwer będzie w stanie odpowiedzieć. Czas może być podany w formie konkretnej daty (jak w przykładzie) lub liczby sekund.

Slajd 31

Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are at the top. The slide title is "Nagłówek Server". The main content is "Server: Apache/2.0.50". The footer includes the DAILY GROUP logo and "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Nagłówek identyfikujący serwer i użyte w nim oprogramowanie.

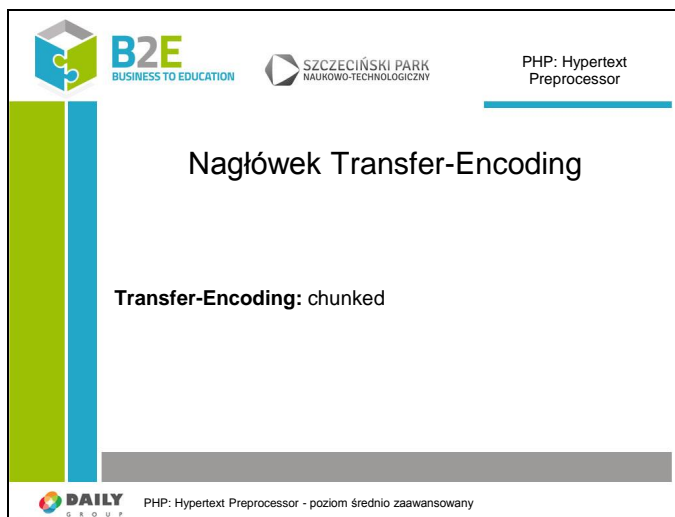
Slajd 32

Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are at the top. The slide title is "Nagłówek Set-Cookie". The main content is "Set-Cookie: ciastko1=wartosc1; ciastko2=wartosc2". The footer includes the DAILY GROUP logo and "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Nagłówek wysłany od serwera nakazuje przeglądarce ustawienie określonych ciasteczek.



Slajd 33



B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek Transfer-Encoding

**Transfer-Encoding: chunked**

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa w jaki sposób serwer przesłał zawartość dokumentu.

Dostępne metody to:

- chunked

Oznacza przesyłanie zawartości kawałkami. Każdy z nich powinien mieć postać: wielkość treść wielkość – jest to długość treści w bajtach ,zapisane w formie szesnastkowej. Przykład: 1a abcdefghijklmnopqrstuvwxyz 10 1234567890abcdef 0 Zero w ostatniej linijce określa zerowej długości kawałek i kończy transmisję.

- identity

Oznacza tekst nieskompresowany


- gzip

Oznacza dane skompresowane algorytmem ZIP

- compress

Dane skompresowane algorytmem LZW

Slajd 34



B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Nagłówek User-Agent

**User-Agent: Mozilla/5.0 (X11; U; Linux i686; pl; rv:1.8.0.1) Gecko/20060124 Firefox/1.5.0.1**

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Identyfikuje przeglądarkę, silnik html oraz System operacyjny.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 35

Slide 35 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is 'Nagłówek WWW-Authenticate'. Below the title, the text reads 'WWW-Authenticate: rodzaj autentyfikacji'. At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Przesyłany razem z kodem 401. Określa sposób w jaki ma zostać przeprowadzona identyfikacja użytkownika.

Slajd 36



Slide 36 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is 'Funkcja getallheaders() oraz apache\_response\_headers()'. Below the title, there is a list of two items: 'array **getallheaders** ( void )' and 'array **apache\_response\_headers** ( void )'. At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Funkcja 'getallheaders' pobiera wszystkie nagłówki żądania http i parsuje je do tablicy PHP, a funkcja 'apache\_response\_headers', parsuje wszystkie nagłówki odpowiedzi http.

Slajd 37

Slide 37 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is 'Funkcja getallheaders'. Below the title, there is a list of two items: 'array **getallheaders** ( void )' and 'array **apache\_response\_headers** ( void )'. Below the list, the text reads 'Wykonaj skrypty wyświetlające rezultaty ww funkcji.'. At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.


Wykonaj skrypty z lekcji 4, wyświetlające rezultaty funkcji getallheaders(); oraz apache\_response\_headers();

PHP: Hypertext Preprocessor



## Funkcja header

```
void header (string $string [, bool $replace = true [, int $http_response_code]])
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

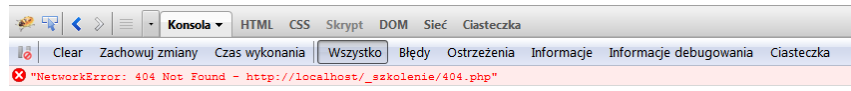

Funkcja 'header' służy do przesyłania nagłówka http dokumentu. Pierwszym argumentem jest treść nagłówka w postaci ciągu znaków PHP. Drugim argumentem jest argument opcjonalny, mówi o tym, czy funkcja ma zastąpić nagłówek tego samego typu przygotowany przez serwer. Kolejnym argumentem opcjonalnym jest narzucony serwerowi kod odpowiedzi.

PHP: Hypertext Preprocessor

## Przykłady zastosowania header()

```
<?php
 header("HTTP/1.0 404 Not Found");
?>
```


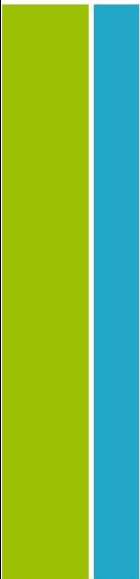
PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przy odpowiedniej konfiguracji serwera HTTP, język PHP może zwracać komunikaty o braku zasobów np. strony o podanym adresie. W tym wypadku znany wszystkim błąd 404.


Zrzut ekranu przedstawia narzędzie Firebug. Jeśli przeglądarka Firefox nie zawiera wspomnianego narzędzia, trzeba je pobrać oraz zainstalować (Firefox -> Dodatki -> Rozszerzenia).



Slajd 40



B2E  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykłady zastosowania header()

```
<?php
header("Location: http://przyklad.pl/");
exit();
?>
```


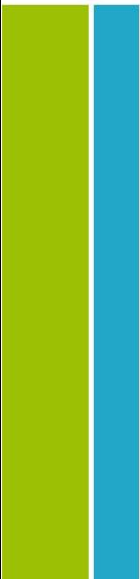


DAILY GROUP


PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Proste przekierowanie HTTP na podany adres.

Slajd 41



B2E  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykłady zastosowania header()



```
<?php
$file = 'przyklad.php';
header('Content-
disposition: attachment; filename=' . basename($file));
header('Content-type: text/plain');
readfile($file);
?>
```



DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Przykład dotyczy wykorzystania nagłówków do pobrania pliku typu PHP jako tekst.

PHP: Hypertext Preprocessor



## Przykłady zastosowania header()

```
header('Cache-Control: no-cache, no-store, max-age=0,
must-revalidate');
header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
header('Pragma: no-cache');
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Powyższy przykład spowoduje wyłączenie cache'u.

PHP: Hypertext Preprocessor

## Przykłady zastosowania header()

```
<?php
header('Content-Type: text/html; charset=utf-8');
echo 'ąęźźńń';
?>
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Powyższy przykład ustawia typ kodowanie pliku na utf-8.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego  
UWAGA: nagłówek informuje przeglądarkę, że będzie wysyłał treść w kodowaniu UTF-8.  
Aby polskie znaki diakrytyczne prawidłowo wyświetliły się w przeglądarce, plik zawierający polskie znaki również musi być zakodowany w UTF-8.

Slajd 44

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Przykład zastosowania kodu HTML w PHP - przekierowanie

```
<?php
echo "<meta http-equiv='refresh' content='0;url=http://przykład.pl'>";
?>
```

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przekierowanie lub odświeżenie strony wygenerowane przed kod php.



Po załadowaniu strony przeglądarka wykonuje kod HTML i przechodzi na adres <http://przykład.php>

Proszę zwrócić uwagę na fakt, iż jest to inna metoda niż:

```
header('Location: http://przykład.pl');
```

gdzie przeglądarka bez ładowania strony natychmiast wykonuje przekierowanie.


Slajd 45



PHP: Hypertext Preprocessor

## Ćwiczenia praktyczne



Wykonaj skrypt, który zwróci do przeglądarki odpowiedź o braku znalezienia strony (404 – Not Found). Sprawdź działanie skryptu poprzez analizę nagłówka zwróconej odpowiedzi HTTP przy pomocy dowolnego narzędzia (Firebug w Firefox, Narzędzie dla programistów – Chrome)



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

404.php


Slajd 46



PHP: Hypertext Preprocessor

## Ćwiczenia praktyczne



Wykonaj skrypt, który najpierw odczeka 10 sekund a następnie zmodyfikuje element nagłówka „Location” i przekieruje użytkownika na dowolny inny adres.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

location.php


Slajd 47



PHP: Hypertext Preprocessor

## Ćwiczenia praktyczne

Wykonaj skrypt, który wymusi na przeglądarce rozpoczęcie pobieranie zdefiniowanego przez Ciebie pliku oraz ukryje przed pobierającym oryginalną lokalizację pliku.





PHP: Hypertext Preprocessor - poziom średnio zaawansowany

downloadFile.php




Slajd 48

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

  PHP: Hypertext Preprocessor



## Ćwiczenia praktyczne

Wykonaj skrypt, który z poziomu PHP wyświetli nagłówki HTTP żądania oraz odpowiedzi.

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


get.php

Slajd 49

  PHP: Hypertext Preprocessor


## Ćwiczenia praktyczne

Stwórz plik, który będzie zawierał link do innego pliku. Po kliknięciu w link wyszukaj i sprawdź element nagłówka Referer żądania HTTP.



 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

referer1.php, referer.php

Slajd 50

  PHP: Hypertext Preprocessor

## Dziękujemy za uwagę

Człowiek - najlepsza inwestycja

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

## Ćwiczenia

Jako ćwiczenia należy przećwiczyć z uczniami przykłady podane na slajdach.



## Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą znali nagłówki protokołu HTTP oraz będą potrafili stosować je w swoich skryptach.

## Lekcja 5 – Cookie - informacje o użytkowniku

**(Budowa i działanie cookie, Używanie cookie w aplikacjach, Definiowanie i obsługa z poziomu PHP, Bezpieczeństwo)**

### Cel lekcji

Celem lekcji jest wyjaśnienie uczniom na czym polega mechanizm ciasteczek i jak stosować go w języku PHP. Opisano również wady i zalety stosowania tego mechanizmu oraz do czego należy go stosować.

### Treść - slajdy z opisem

Slajd 1

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor  
- poziom średnio zaawansowany  
Lekcja 5  
Cookie - informacje o użytkowniku

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI

UNIA EUROPEJSKA

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


DAILY GROUP


Słowa wstępu.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 2



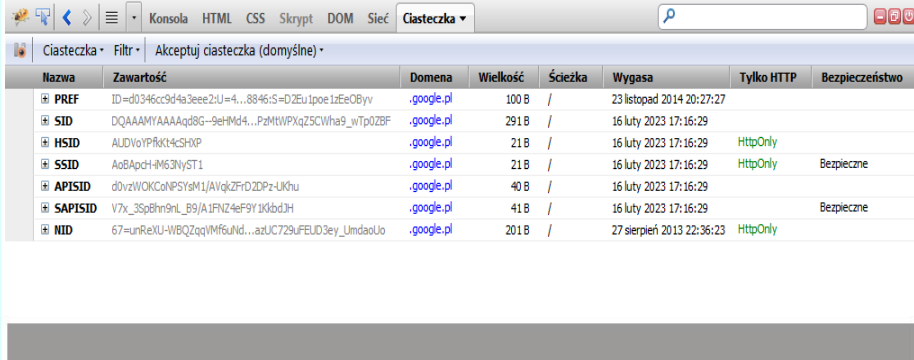


SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

## Cookies – ciasteczka

Wejdź na dowolną stronę i sprawdź ciasteczka dla danej witryny w narzędziu firebug





Nazwa	Zawartość	Domena	Wielkość	Ścieżka	Wygasa	Tylko HTTP	Bezpieczeństwo
PREF	ID=d0346cc9d4a3eee21u=4...8846-S=02Eu1poe1zEeOByv	.google.pl	100 B	/	23 listopad 2014 20:27:27		
SID	DQAAAMYAAAAAgd8G--9eHMD4...PaMlWpXqZ5CWha9_wTp0ZBF	.google.pl	291 B	/	16 luty 2023 17:16:29		
HSID	AUDVoYPfKl4cSHVP	.google.pl	21 B	/	16 luty 2023 17:16:29	HttpOnly	
SSID	As8ApdH-M63NyST1	.google.pl	21 B	/	16 luty 2023 17:16:29	HttpOnly	Bezpieczne
APISID	d0vzWOKCoNPSy9M1jAVqkzFDZDPz-Ukhu	.google.pl	40 B	/	16 luty 2023 17:16:29		
SAPISID	V7x_3Sp8m9nL_B9/A1FNZ4eF9Y1Kbd3H	.google.pl	41 B	/	16 luty 2023 17:16:29		Bezpieczne
NID	67=unReXU-WBQZqqIMf6uNd...adUC729uFUD03ey_UmdaoUo	.google.pl	201 B	/	27 sierpień 2013 22:36:23	HttpOnly	



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slajd 3






SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Cookies – ciasteczka

Wejdź na dowolną stronę i sprawdź ciasteczka dla danej witryny przy pomocy wybranego przez Ciebie narzędzia dla deweloperów jak Firebug, „Narzędzie dla programistów” Chrome, lub dowolnego innego.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Można tutaj użyć np. standardowo w budowanego w przeglądarkę Chrome „Narzędzia dla programistów” wchodząc w: Ustawienia->Narzędzia->Narzędzia dla programistów (kombinacja Ctrl+Shift+I)

Slajd 4

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Cookies - ciasteczka

- Historia
- Czym jest mechanizm ciasteczek?

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Mechanizm ciasteczek został wymyślony przez pracownika firmy Netscape Communications.

Ciasteczko to niewielka porcja danych wysyłana przez serwer WWW i zapisywana w danych przeglądarki po stronie klienta. Dostęp do nich ma tylko serwer, który je wysłał, czyli mówiąc prosto każda domena ma swoje własne ciasteczka po stronie klienta, o ile mechanizm jest włączony i ciasteczka dla konkretnego adresu są akceptowane. Obecnie wykorzystuje się je głównie do prowadzenia statystyk, bądź zliczania wyświetleń reklam.

Slajd 5

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor


## Cookies w PHP

- Gdzie można je znaleźć – tablica `$_COOKIE`?
- Jak z nich korzystać?


DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wszystkie ciasteczka dla danej domeny przechowywane są w superglobalnej tablicy `$_COOKIE`. PHP udostępnia kilka funkcji pomocniczych do mechanizmu ciasteczek np. `setcookie`.





**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---

## Set Cookie – Tworzymy ciasteczko

- `bool setcookie ( string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]] )`
- `bool setrawcookie ( string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false ]]]]] ] )`



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Ciasteczko ustawiamy (tworzymy) używając funkcji `setcookie` lub analogicznej funkcji `setrawcookie`, które różnią się tylko tym, że `setrawcookie` nie używa automatycznego enkodowania adresu url. Wymagany jest pierwszy parametr, czyli nazwa ciasteczka. Pozostałe parametry są opcjonalne, dlatego umieszczone w nawiasach kwadratowych. Drugim parametrem jest wartość ciasteczka. Kolejnym data do której ciasteczko jest ważne w formacie Unixowym, czyli daty i czasu skonwertowanego do liczby całkowitej. Czwartym parametrem jest ścieżka pod którą ciasteczka mają być dostępne np. `/`, pozwala na dostęp w dowolnej ścieżce. Piątym parametrem jest nazwa domeny, która ogranicza dostęp do ciasteczka dla domen zewnętrznych, najbardziej zewnętrzną domeną, która pozwala na dostęp dla całej witryny jest np. `nazwa_domeny.pl`. Wszystkie domeny wewnętrzne np. `www.nazwa_domeny.pl`, będą miały dostęp do ciasteczek. Następne parametry określają dostęp do ciasteczka ze względu na używany protokół. `Secure` mówi, że ciasteczko będzie tylko dostępne jeżeli ustawione jest połączenie HTTPS., a `httponly`, że jest dostępne tylko za pomocą protokołu HTTP. Funkcja zwraca `true` (prawdę), jeżeli ciasteczko zostało ustawione, `false` (fałsz) w przeciwnym wypadku.

Należy dodatkowo zwrócić uwagę na istotny fakt, iż próba ustawienia ciasteczka musi nastąpić przed wysłaniem jakiegokolwiek treści do przeglądarki. W przypadku nie spełnienia tego warunku otrzymamy ostrzeżenie:


**Warning:** Cannot modify header information - headers already sent.

Jak można wywnioskować z ostrzeżenia, ustawianie ciasteczek jest realizowane przez wysłanie stosownych nagłówków.

Przykład nieprawidłowego użycia `setCookie` dostępny jest w pliku: `wrong.php`



B2E  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Read Cookie – odczytujemy ciasteczko

- Do odczytu używamy tablicy oraz klucza(nazwy)  

```
if(isset($_COOKIE['cookie_name']))
 $cookie = $_COOKIE['cookie_name'];
```



DAILY  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Odczyt ciasteczek jest o wiele prostszy, niż ich tworzenie. Posługujemy się tablicą `$_COOKIE` oraz nazwą ciasteczka. Z racji możliwości kasowania ciasteczek przez użytkowników, należy się upewnić, że owe ciasteczko istnieje, możemy wykorzystać do tego funkcję `isset`.

Ze względu na fakt, iż skrypty php przetwarzane są po stronie serwera, a ciastka przechowywane są po stronie klienta (w przeglądarce) przy pierwszym żądaniu serwer otrzymuje stan ciasteczek z poprzedniego żądania, następnie wykonuje skrypt w którym ustawia ciastko. Wysyła odpowiedź HTTP z nowo ustawionym ciastkiem. Dopiero w drugim żądaniu otrzymuje ciastko ustawione w pierwszym żądaniu i może je przetworzyć.



Przykład: wyobraźmy sobie skrypt `cookie.php` który ustawia oraz odczytuje ciastko - gdy ono istnieje.

- Pierwsze przetworzenie skryptu serwer wyśle do przeglądarki informację o stworzeniu ciastka, ale zawartość ciastka nie wyświetli się
- Drugie przetworzenie skryptu – do serwera dociera ciastko utworzone w poprzednim wywołaniu, strona serwerowa przetwarza jego zawartość i generuje rezultat polegający na wyświetleniu zawartości ciastka, który przesyła do przeglądarki



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


Slajd 8



PHP: Hypertext Preprocessor

### Ćwiczenie – tworzymy ciasteczko

- Wykorzystując funkcję setcookie ustaw ciasteczko o nazwie 'name', zawierające wartość Twojego imienia oraz następnie sprawdź czy istnieje jeżeli tak, wyświetl komunikat: Hello, 'ciasteczko\_z\_imieniem!' (skrypt nazwij name.php) Jeżeli nie ma ciasteczka lub jest puste, wypisz tylko 'Hello'. Odśwież stronę dwa razy.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slajd 9



PHP: Hypertext Preprocessor

### Rozwiązanie i opis

```
<?php
if (!isset($_COOKIE['name']))
 setcookie('name', 'Stefan');
if (isset($_COOKIE['name']) and !empty($_COOKIE['name']))
 echo 'Hello, ' . $_COOKIE['name'] . '!';
else
 echo 'Hello!';
?>
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Rozwiązanie poprzedniego ćwiczenia.

W pierwszej kolejności sprawdzamy, czy ciasteczko istnieje, jeżeli nie to je tworzymy. Następnie jeżeli ciasteczko istnieje i nie jest puste wypisujemy 'Hello, imię', a jeżeli istnieje i jest puste wypisujemy tylko 'Hello'.

Slajd 10

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor



CSLACKER.COM

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Chwila rozluźnienia.

Slajd 11

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Overwrite Cookie – nadpisywanie ciasteczka

- Ponowne wywołanie funkcji setcookie, bez sprawdzenia, czy ciasteczko już istnieje

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Sprawdzenie istnienia ciasteczka za pomocą funkcji 'isset', jest rzeczą istotną, ponieważ jeżeli ustawimy ciasteczko o nazwie istniejącego ciasteczka, zostanie ono nadpisane nowym. Mechanizm ten jest wykorzystywany w praktyce, z tym, że dobrze jest rozumieć jak on działa i używać go świadomie, ze względu na możliwe problemy w aplikacjach.

Slajd 12

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Delete Cookie – usuwanie ciasteczka

- `setcookie("hello", "", time()-3600);`

DAILY GROUP

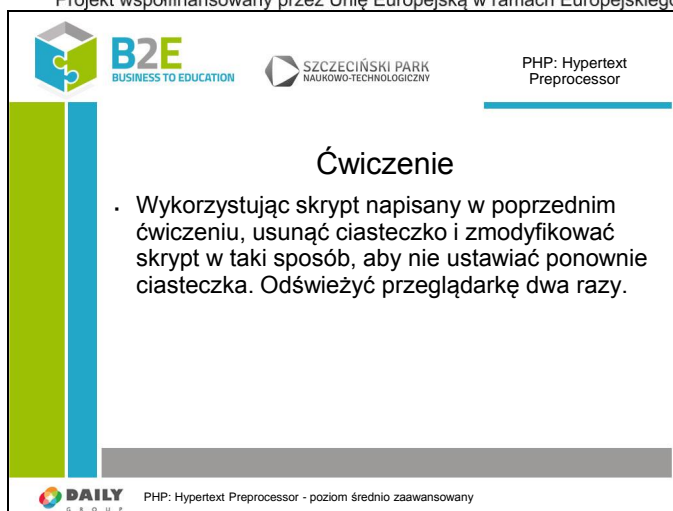
PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jedyną metodą usunięcia ciasteczka jest stworzenie ciasteczka, którego okres ważności się skończył. W powyższym przykładzie ustawiliśmy ważność ciasteczka na czas obecny minus 3600 sekund, czyli na godzinę wstecz.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13



Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor


### Ćwiczenie

- Wykorzystując skrypt napisany w poprzednim ćwiczeniu, usunąć ciasteczko i zmodyfikować skrypt w taki sposób, aby nie ustawiać ponownie ciasteczka. Odświeżyć przeglądarkę dwa razy.

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slajd 14



Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Rozwiązanie

```
<?php
 if (isset($_COOKIE['name']))
 setcookie('name', '', time() - 3600);
 if (isset($_COOKIE['name']) and !empty($_COOKIE['name']))
 echo 'Hello, ' . $_COOKIE['name'] . '!';
 else
 echo 'Hello!';
?>
```

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

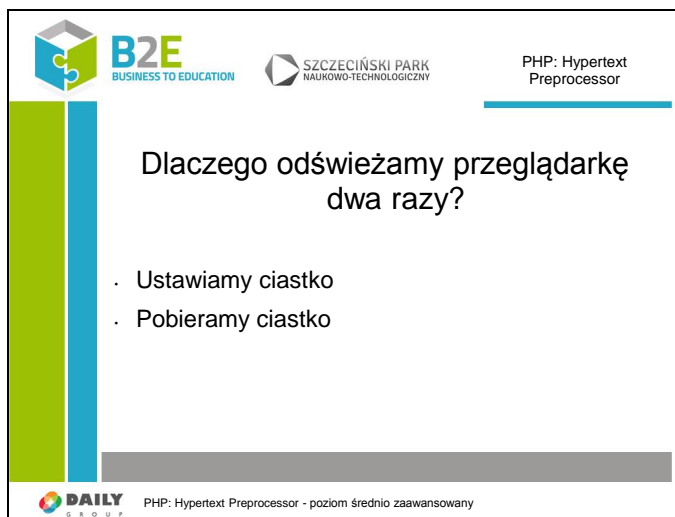
Rozwiązanie poprzedniego ćwiczenia.

W pierwszej kolejności sprawdzamy, czy ciasteczko istnieje, aby następnie ustawić jego datę ważności na godzinę wstecz.

Następnie jeżeli istnieje, wypisujemy 'Hello, ciastko', bądź jeżeli je usunęliśmy tylko 'Hello'.



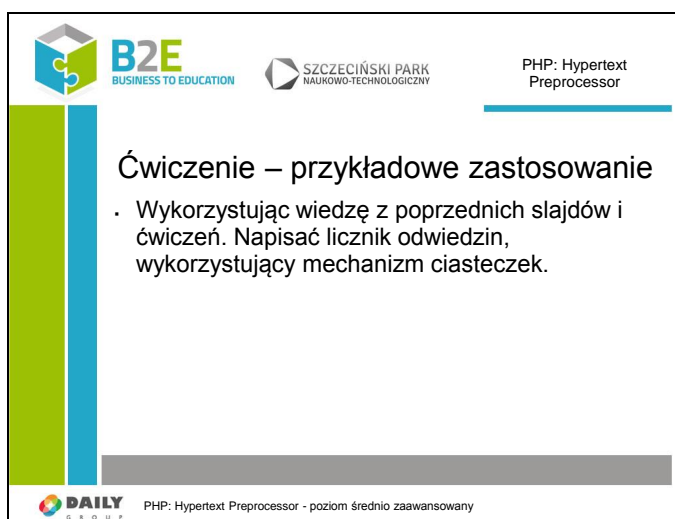
Slajd 15



Slide 15 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is "Dlaczego odświeżamy przeglądarkę dwa razy?". Below the title is a bulleted list: ". Ustawiamy ciastko" and ". Pobieramy ciastko". The footer includes the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Skrypty PHP wykonywane są po stronie serwera w wyniku zgłoszenia żądania HTTP w którym wysyłamy ciasteczka dla danego serwera. Przy pierwszym żądaniu serwer otrzymuje stan ciasteczek z poprzedniego żądania, następnie wykonuje skrypt w którym ustawia ciastko. Wysyła odpowiedź HTTP z nowo ustawionym ciastkiem. Dopiero w drugim żądaniu otrzymuje ciastko ustawione w pierwszym żądaniu i może je przetworzyć.

Slajd 16




Slide 16 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is "Ćwiczenie – przykładowe zastosowanie". Below the title is a bulleted list: ". Wykorzystując wiedzę z poprzednich slajdów i ćwiczeń. Napisać licznik odwiedzin, wykorzystujący mechanizm ciasteczek". The footer includes the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".


Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.



Slajd 17



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Rozwiązanie

```

<?php
 if (!isset($_COOKIE['counter']))
 setcookie('counter', 2, time() + 60 * 60 * 24 * 365);
 if (isset($_COOKIE['counter'])) {
 setcookie('counter', $_COOKIE['counter'] + 1, time() + 60 * 60 * 24 * 365);
 echo 'Odwiedzasz stronę ' . $_COOKIE['counter'] . ' raz!';
 } else {
 echo 'Odwiedzasz stronę 1 raz!';
 }
?>
```



DAILY  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Rozwiązanie poprzedniego ćwiczenia.

Jeżeli ciastko licznika nie istnieje ustawiamy je na dwa, ponieważ, zostanie wyświetlone dopiero przy następnej wizycie i wyświetlamy ciąg znaków z wyrażenia warunkowego po 'else'. W przypadku gdy ciastko jest już ustawione, wyświetlamy jego wartość i zwiększamy o jeden. Kolejność nie ma znaczenia, gdyż jak wcześniej wspomniano ustawione zostanie ono dopiero w odpowiedzi HTTP po stronie klienta, aby zostać wysłane przy kolejnym żądaniu HTTP.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Rozwiązanie podgląd

*Dla zrealizowanego zadania wyświetl zawartość ciasteczka w firebug'u.*




Nazwa	Zawartość	Domena	Wielkość	Ścieżka	Wygasa	Tylko HTTP	Bezpieczeństwo
PHPSESSID	5f8b9f72a3cd3a95a3810ff5c27e69b4	localhost	41 B	/	Sesja		
counter	3	localhost	8 B	/szkolenie/	Sesja		




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Rozwiązanie poprzedniego ćwiczenia.

Jeżeli ciastko licznika nie istnieje ustawiamy je na dwa, ponieważ, zostanie wyświetlone dopiero przy następnej wizycie i wyświetlamy ciąg znaków z wyrażenia warunkowego po 'else'. W przypadku gdy ciastko jest już ustawione, wyświetlamy jego wartość i zwiększamy o jeden. Kolejność nie ma znaczenia, gdyż jak wcześniej wspomniano ustawione zostanie ono dopiero w odpowiedzi HTTP po stronie klienta, aby zostać wysłane przy kolejnym żądaniu HTTP.



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Zalety ciasteczek

- Prywatność - są dostępne tylko dla użytkownika w jego przeglądarce i nikt z zewnątrz nie ma do nich dostępu
- Użytkownik ma na nie bezpośredni wpływ, co może być również wadą, ale w niektórych przypadkach jest zdecydowanie zaletą np. możliwość ich blokowania dla konkretnych stron



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przedyskutować wypunktowane kwestie.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 20

Logos: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Zalety ciasteczek

- Łatwość w ich obsłudze z poziomu języka PHP
- Ułatwienia dla użytkownika, np. nie musi wypełniać formularza, kolejny raz
- Możliwość tworzenia efektywnych kampanii reklamowych oraz prowadzenia statystyk (zaleta dla właściciela portalu)

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przedyskutować  
wypunktowane kwestie.

Slajd 21

Logos: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Wady ciasteczek

- Ciasteczka muszą być dołożone do każdego żądania HTTP
- Mają ograniczony rozmiar z góry przez przeglądarkę
- Bez użycia HTTPS są widoczne i niezaszyfrowane w przeglądarce
- Użytkownik może zmieniać dane w nich zawarte

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przedyskutować  
wypunktowane kwestie.

Slajd 22

Logos: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

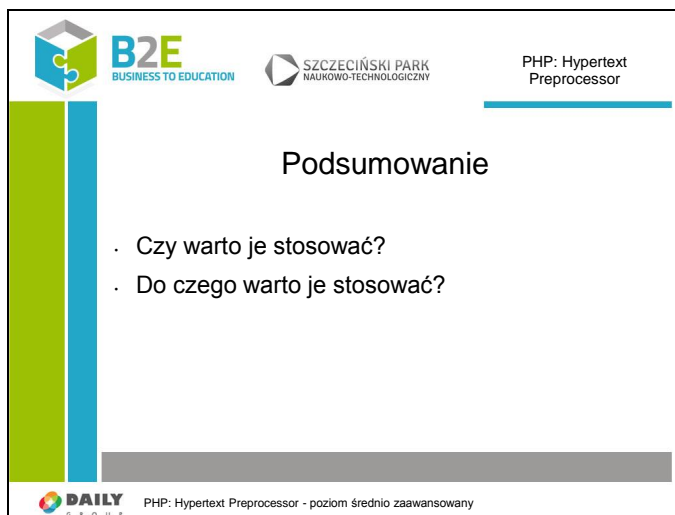
### Wady ciasteczek

- Użytkownik może je usunąć
- Użytkownik może zablokować ciasteczka
- Zależą od przeglądarki, czyli ich bezpieczeństwo zależy od konkretnej implementacji, czasami wadliwej, wykazano luki w niektórych przeglądarkach

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przedyskutować  
wypunktowane kwestie.

Slajd 23



Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

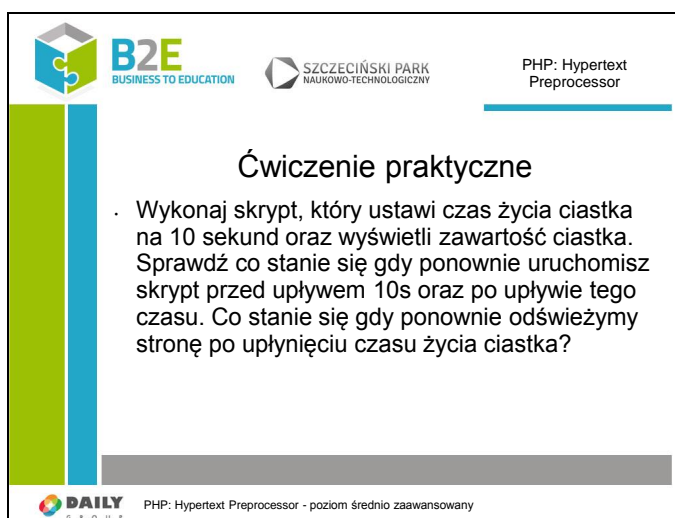
## Podsumowanie

- Czy warto je stosować?
- Do czego warto je stosować?

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ciasteczka ułatwiają życie zarówno użytkownikowi, programiście oraz właścicielowi portalu. W niektórych kwestiach warto je stosować np. w statystykach, autouzupełnianiu formularzy, kampaniach reklamowych czy jeżeli używamy szyfrowania np. SSL, również do autoryzacji.

Slajd 24



Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

## Ćwiczenie praktyczne

- Wykonaj skrypt, który ustawi czas życia ciastka na 10 sekund oraz wyświetli zawartość ciastka. Sprawdź co stanie się gdy ponownie uruchomisz skrypt przed upływem 10s oraz po upływie tego czasu. Co stanie się gdy ponownie odświeżymy stronę po upływie czasu życia ciastka?

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

cookieLifeTime.php

Slajd 25



Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

## Dziękujemy za uwagę

Celownik - najlepsza inwestycja

KAPITAŁ LUDZKI

UNIA EUROPEJSKA

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

## Ćwiczenia

Ćwiczenia zostały przedstawione na slajdach. Podczas prowadzenia zajęć należy wykonywać je razem z uczniami.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Opis założonych osiągnięć ucznia

Uczeń powinien osiągnąć wiedzę na temat mechanizmu ciasteczek, poznać schemat jego działania oraz potrafić korzystać z niego w swoich skryptach.

## Lekcja 6 - Sesje - współdzielenie informacji

### (Struktura i sposób działania, Kontrola sesji w PHP, Wykorzystanie sesji w aplikacjach internetowych)

#### Cel lekcji

Celem lekcji jest zaprezentowanie mechanizmu sesji, opis jego wykorzystania w skryptach i nauczenie uczestników kursu wykorzystywać powyższy mechanizm w ich skryptach.

#### Ćwiczenia

Ćwiczenia zostały przedstawione na slajdach. Podczas prowadzenia zajęć należy wykonywać je razem z uczniami.

#### Treść - slajdy z opisem

Slajd 1

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main content area contains the title 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany', the lesson number 'Lekcja 6', and the topic 'Sesje - współdzielenie informacji'. At the bottom, there are logos for 'KAPITAŁ LUDZKI' and 'UNIA EUROPEJSKA' with the text 'Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego'. The 'DAILY GROUP' logo is located in the bottom left corner.

Słowa wstępu.

## Slajd 2

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Session – czyli Sesje HTTP

- Czym jest Sesja HTTP?
- Na czym polega mechanizm Sesji internetowej

**DAILY** PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Sesja HTTP to mechanizm przechowywania danych po stronie serwera. Parametry sesji takie jak np. czas trwania sesji, znajdują się w pliku konfiguracyjnym serwera. W przeciwieństwie do mechanizmu ciasteczek znanego z poprzedniej lekcji. Sesje są niezależne od klienta, dzięki czemu dają programiście większą kontrolę nad danymi w nich przechowywanymi. Każda sesja posiada swój identyfikator, który jest unikatowy i jeden dla każdego klienta. Klientem w rozumieniu sesji jest np. przeglądarka, a nie adres komputera osoby z niej korzystającej. Konsekwencją tego jest tworzenie kilku sesji, dla jednego adresu komputera, w przypadku np. zalogowania się z różnych przeglądarek, czyli możliwość wielokrotnego logowania się na np. jedno konto, w przypadku nie zaimplementowanej możliwości jednokrotnego dostępu z poziomu jednego konta.

## Slajd 3

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Sesje w PHP

- Superglobalna tablica `$_SESSION`
- Otwieranie i zamykanie sesji

**DAILY** PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Sesje w przeciwieństwie do ciasteczek, są mechanizmem bardziej rozbudowanym. Podobnie jak ciasteczka posiadają tablicę superglobalną, dostępną w każdym miejscu aplikacji PHP. Dane z sesji są dostępne w skrypcie od razu po ich stworzeniu (ustawieniu), nie ma potrzeby odświeżania strony, żeby korzystać z sesji trzeba ją jednak najpierw otworzyć lub stworzyć, możemy ją również zamknąć wedle naszej potrzeby.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4

Aby zainicjalizować sesję, używamy funkcji 'session\_start', zwraca ona zawsze wartość true i nie przyjmuje żadnych argumentów. Aby pobrać stan sesji stosujemy funkcję 'session\_status', zwraca 0 jeżeli sesje są wyłączone po stronie serwera, 1 jeżeli sesja nie istnieje lub jest nieaktywna bądź, 2 jeżeli istnieje.

Slajd 5


Funkcja session\_id(), zwraca nam identyfikator obecnej sesji lub jeżeli zostanie identyfikator jako parametr, ustawia identyfikator sesji, w tym przypadku należy ją wywołać przed funkcją session\_start. Kolejna funkcja zastępuje obecny identyfikator sesji, nowym, a argument decyduje o tym, czy usunąć zmienianą sesję, czy nie.

Slajd 6

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.



Slajd 7



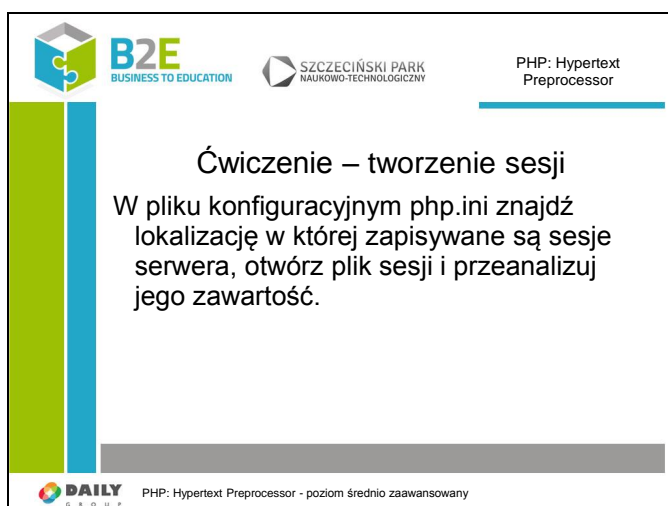
Slide 7 content: The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is "Rozwiązanie". Below it is a PHP code snippet: 

```
<?php
session_id('Ewa');
if (session_id() == '')
 session_start();
echo session_id();
?>
```

 The footer includes the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

W pierwszej kolejności ustalamy nazwę sesji na nasze imię. Następnie sprawdzamy czy sesja istnieje, jeżeli nie istnieje tworzymy nową lub startujemy wcześniej utworzoną. Na koniec wypisujemy identyfikator sesji.

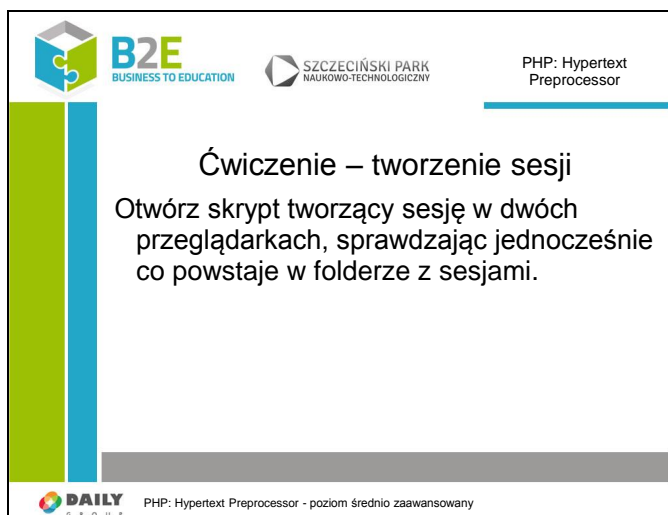
Slajd 8



Slide 8 content: The slide features a header with logos for B2E, Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is "Ćwiczenie – tworzenie sesji". Below it is a text instruction: "W pliku konfiguracyjnym php.ini znajdź lokalizację w której zapisywane są sesje serwera, otwórz plik sesji i przeanalizuj jego zawartość." The footer includes the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Wykonać ćwiczenie.

Slajd 9



Slide 9 content: The slide features a header with logos for B2E, Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main title is "Ćwiczenie – tworzenie sesji". Below it is a text instruction: "Otwórz skrypt tworzący sesję w dwóch przeglądarkach, sprawdzając jednocześnie co powstaje w folderze z sesjami." The footer includes the DAILY GROUP logo and the text "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

Wykonać ćwiczenie.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 10

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Dodawanie danych do sesji

- session\_register (stara metoda)
- \$\_SESSION['key'] = \$value;

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby dodać dane do wcześniej rozpoczętej sesji, należy użyć drugiej metody ze slajdu. Pierwsza metoda jest przestarzała i niepoprawna w najnowszej implementacji PHP, działa w wersji PHP < 5.4. Wartości zapisujemy operując bezpośrednio na superglobalnej tablicy sesji.

Slajd 11

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Nadpisywanie danych w sesji

- \$\_SESSION['key'] = \$value;

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jeżeli klucz w sesji istnieje, a my dodamy od niego nowe dane, zostanie on nadpisany. Tablica sesji działa dokładnie tak jak inne tablice w języku PHP.

Slajd 12

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Usuwanie bieżących danych sesji

- session\_destroy();

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Funkcja session\_destroy usuwa tylko bieżące dane skojarzone z sesją. Nie usuwa jednak sesji, której identyfikator przechowywany jest w ciasteczku sesji.

Slajd 13

Nazwa sesji

- string **session\_name** ([ string \$nazwa ] )

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nazwa sesji - obok identyfikatora - jest używana do identyfikacji sesji. Funkcja `session_name` podobnie jak funkcja `session_id`, służy albo do zwracania nazwy sesji albo do zmieniania jej nazwy. Podobnie jak `session_id` musi być wywołana przed funkcją `session_start`.

Slajd 14

Usuwanie sesji

```
if (isset($_COOKIE[session_name()])) {
 setcookie(session_name(), "", time()-42000, '/');
}
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby usunąć sesję całkowicie, należy usunąć ciasteczko z jej nazwą, które jest przechowywane po stronie użytkownika. Wpierw należy sprawdzić, czy takowe istnieje, a następnie należy je usunąć, metodą podaną w poprzedniej lekcji.

Slajd 15

Usuwanie danych z sesji

- `session_unset();`
- `unset($_SESSION['key']);`

PHP: Hypertext Preprocessor - poziom średnio zaawansowany



Aby usunąć wszystkie dane ustawione w sesji kiedykolwiek, należy zastosować funkcję `session_unset`.

Aby usunąć konkretną zmienną należy użyć drugiej konstrukcji.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


Slajd 16



PHP: Hypertext Preprocessor

### Ćwiczenie – tworzenie sesji


- Należy przygotować skrypt, który zapisze w sesji nasze imię oraz wyświetli 'Hello, nasze\_imie' z sesji. Następnie zapisze do poprzedniej zmiennej nazwisko i wyświetli 'Hello, nasze\_nazwisko'. Na końcu usunie całą sesję.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slajd 17



PHP: Hypertext Preprocessor

### Rozwiązanie cz.1



```
<?php
if (session_id() == '')
 session_start();
if (!isset($_SESSION['name']))
 $_SESSION['name'] = 'Ewa';
echo 'Hello, ' . $_SESSION['name'];
$_SESSION['name'] = 'Kowalska';
echo '
Hello, ' . $_SESSION['name'];
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności inicjalizujemy sesję. Następnie ustawiamy zmienną `$_SESSION['name']` na nasze imię oraz wyświetlamy ją.

Kolejnym krokiem jest nadpisanie zmiennej naszym nazwiskiem oraz wyświetlenie jej.


PHP: Hypertext Preprocessor

## Rozwiązanie cz.2

```



session_unset();
if (isset($_COOKIE[session_name()])) {
 setcookie(session_name(), '', time() -
42000, '/');
}
?>

```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Część końcową naszego skryptu rozpoczyna usunięcie zmiennych z sesji oraz zniszczenie ciasteczka z jej identyfikatorem.

PHP: Hypertext Preprocessor

### Ćwiczenie – tworzenie sesji

- Napisz skrypt, który będzie zliczał i wyświetlał wizyty na stronie. Dane musi przechowywać w tablicy sesji. Korzystając z tablicy \$\_GET oraz funkcji header do naszego licznika dodajmy link, który go zeruje.




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 20



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---

## Rozwiązanie


```
<?php
$url = (@$_SERVER["HTTPS"] == "on") ? "https://" : "http://";
if ($_SERVER["SERVER_PORT"] != "80") {
 $url .= $_SERVER["SERVER_NAME"] . ":" .
 $_SERVER["SERVER_PORT"] . $_SERVER["REQUEST_URI"];
} else {
 $url .= $_SERVER["SERVER_NAME"] . $_SERVER["REQUEST_URI"];
}
```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności potrzebujemy adres url naszej strony, będzie nam on potrzebny do zerowania licznika.

Slajd 21



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---

## Rozwiązanie

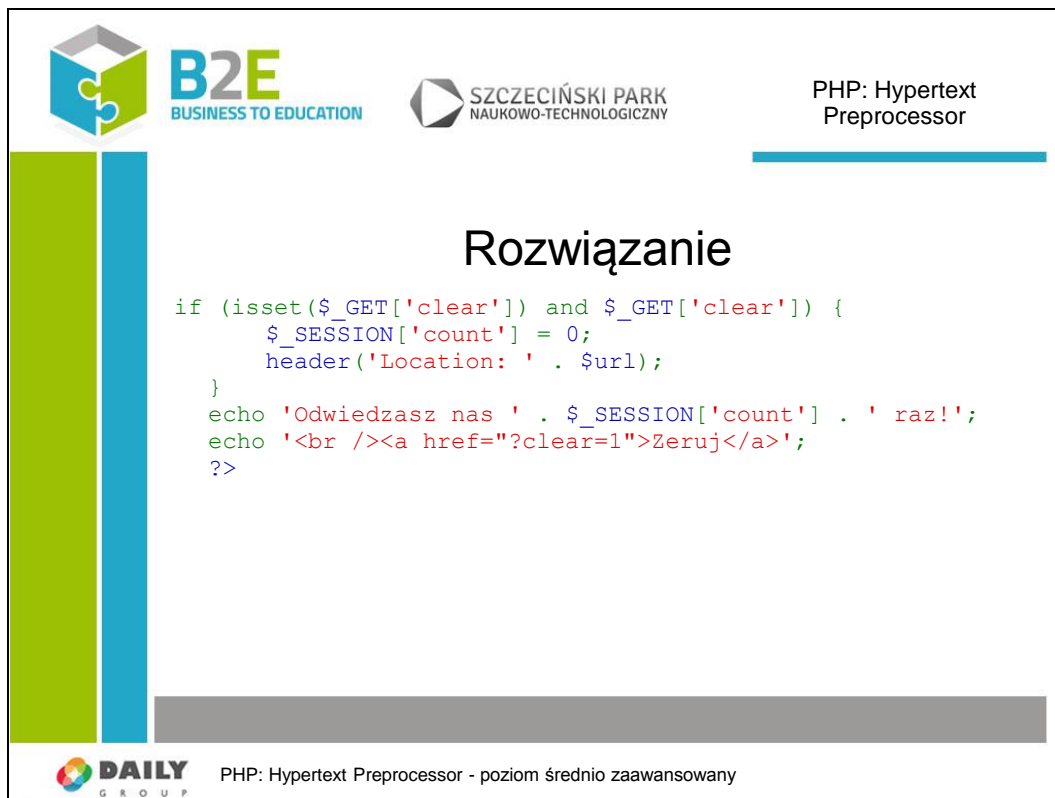
```
$url = str_replace('?clear=1', '', $url);
if (session_id() == '')
 session_start();
if (!isset($_SESSION['count']))
 $_SESSION['count'] = 0;
else
 $_SESSION['count'] = $_SESSION['count'] + 1;
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Następnie czyścimy z niego zmienne z tablicy \$\_GET, w przeciwnym wypadku zerowalibyśmy licznik za każdym razem. Kolejną rzeczą jest implementacja samego mechanizmu zliczającego i przechowującego dane w sesji.

Slajd 22



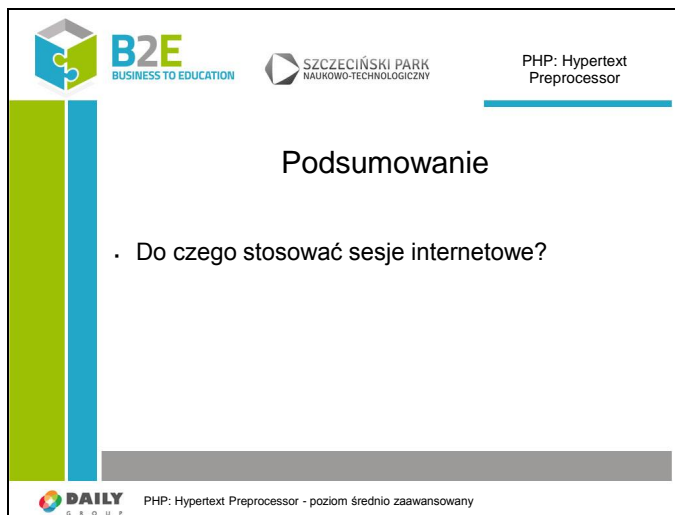
Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are visible in the top left. The title 'PHP: Hypertext Preprocessor' is in the top right. The main content is a code snippet:

```
if (isset($_GET['clear']) and $_GET['clear']) {
 $_SESSION['count'] = 0;
 header('Location: ' . $url);
}
echo 'Odwiedzasz nas ' . $_SESSION['count'] . ' raz!';
echo '
Zeruj';
?>
```

At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Na koniec potrzebna jest nam akcja, która w związku ze zmienną \$\_GET wyczyści nam licznik i odświeży stronę. Kolejną rzeczą jest wyświetlenie naszego licznika oraz linku do jego zerowania.

Slajd 23



Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are visible in the top left. The title 'PHP: Hypertext Preprocessor' is in the top right. The main content is a question:

- Do czego stosować sesje internetowe?



At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Sesje HTTP służą głównie do systemów logowania oraz przechowywania danych globalnych lub konfiguracyjnych. Są bezpieczniejsze od ciasteczek jeśli chodzi o przechowywanie danych ze względu na ich umiejscowienie (serwer). Opierają się jednak o system ciasteczek, co może spowodować brak możliwości dostania się do odpowiedniej sesji.




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 24

  PHP: Hypertext Preprocessor



### Ćwiczenie praktyczne

Używając sesji, przygotuj skrypt odpowiedzialny za logowanie użytkownika do systemu. Sprawdź login oraz hasło, w przypadku sukcesu zaloguj użytkownika i wyświetl jego login. Po wylogowaniu wyświetl ponownie formularz logowania.

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Login.php, logout.php

Slajd 25

  PHP: Hypertext Preprocessor



### Ćwiczenie praktyczne

Do stworzonego wcześniej systemu logowania dodaj sprawdzanie liczby nieudanych prób zalogowania (błędny login lub hasła). Po 5 nieudanych próbach zablokuj możliwość zalogowania do aplikacji.

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Login.php, logout.php

Slajd 26

  PHP: Hypertext Preprocessor

### Ćwiczenie praktyczne

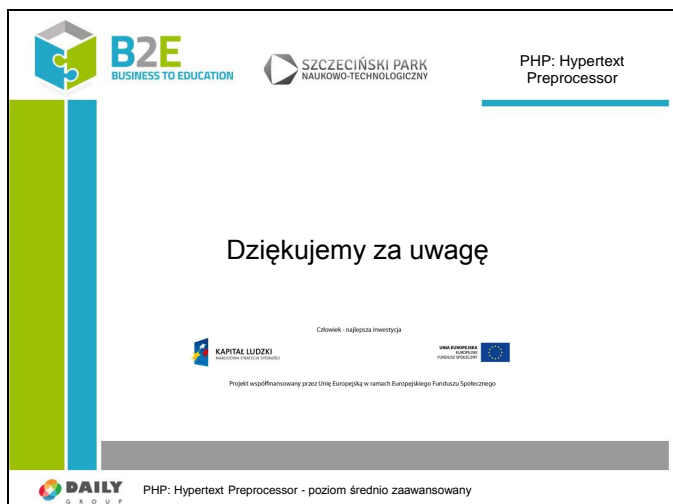
Do stworzonego wcześniej ćwiczenia, dodaj sprawdzanie czasu, który upłynie od momentu zablokowania użytkownika na skutek podania nieprawidłowego loginu lub hasła. Po upłygnięciu minuty odblokuj formularz logowania i pozwól użytkownikowi na ponowną próbę logowania.

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Login.php, logout.php



Slajd 27



### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą posiadać wiedzę o mechanizmie sesji oraz będą potrafili wykorzystać go w swoich skryptach, bądź aplikacjach internetowych.

### Lekcja 7 - Poczta - systemy SMTP

**(Systemy pocztowe w Internecie, Budowa protokołu SMTP, Komunikacja SMTP, Struktura nagłówek wiadomości, Obsługa poczty w aplikacjach PHP)**

### Cel lekcji

Celem lekcji jest zapoznanie z protokołem SMTP. Poznanie jego działania oraz nauczenie się, jak wykorzystać go w języku php, celem wysyłania poczty elektronicznej.

### Treść - slajdy z opisem

Slajd 1



Słowa wstępu



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 2

**BZE** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Simple Mail Transfer Protocol – czyli SMTP

- Co to jest?
- Historia
- Zastosowanie w języku PHP

**DAILY GROUP** PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Protokół SMTP (ang. Simple Mail Transfer Protocol) – to protokół komunikacyjny, który służy jako opis standardu przesyłania poczty elektronicznej. PO raz pierwszy został opisany w roku 1982 przez Jonathana B. Postela. Często jest wymieniany razem z protokołem POP3, który służy do pobierania poczty z serwera. SMTP jest wykorzystywany w języku PHP jako warstwa do obsługi poczty elektronicznej. Możemy z niego korzystać bezpośrednio lub za pomocą np. obiektu phpmailer. Istnieje również wiele innych komponentów wspomagających obsługę poczty elektronicznej na serwerach WWW.

Slajd 3

**BZE** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

## Systemy pocztowe w Internecie

- Gmail
- Hotmail
- Yahoo
- AOL

**DAILY GROUP** PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Do najpopularniejszych systemów pocztowych w internecie należą te wyżej wymienione. Są dostępne z poziomu przeglądarki najczęściej po protokole HTTPS. Działają najczęściej w oparciu o protokoły IMAP (następca POP3) oraz SMTP. W niektórych istnieje możliwość dostępu po zdefiniowanych przez nas protokołach.




PHP: Hypertext Preprocessor



## SMTP

- Domyślny port 25
- Wykorzystuje warstwę TCP/IP
- Adres user@domena.pl



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Protokół SMTP wykorzystuje warstwę transportową TCP, domyślnie po porcie 25. Adres pocztowy zdefiniowany jest jako część związana z domeną, czyli adres serwera na serwerze DNS (Domain Name System), który zwraca adres IP danej domeny oraz nazwę użytkownika systemu pocztowego na danym serwerze.


PHP: Hypertext Preprocessor

## Składnia wiadomości SMTP

```

S: 220 domena.pl ESMTP Postfix
C: HELO filip.domena.pl
S: 250 domena.pl
C: MAIL FROM:<ewa@domena.pl >
S: 250 2.1.0 Ok
C: RCPT TO:<ewa@domena.pl >
S: 250 2.1.5 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: This is the message body.
C: .
S: 250 2.0.0 Ok: queued as C2BEC5B0143
C: QUIT
S: 221 2.0.0 Bye

```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Składnia wiadomości SMTP wygląda całkiem podobnie do składni żądania HTTP. Można ją podzielić na część w której znajdują się nagłówki, część z treścią oraz część kończącą wiadomość. Treść zawsze musi być zakończona znakiem kropki (.) oraz znakami powrotu karetki (CR) i znakiem przejścia do nowej linii (LF).



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6

**Działanie protokołu SMTP cz.1**

1. Połączenie.
2. Potwierdzenie otwarcia sesji
3. Identyfikacja hosta
4. Identyfikacja nadawcy
5. Potwierdzenie gotowości na odebranie wiadomości e-mail

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności nawiązywane jest połączenie TCP hosta z serwerem, do którego ma zostać wysłana wiadomość (wysyłane są trzy wiadomości, pierwsza z hosta na serwer, druga to odpowiedź serwera, trzecia prośba o otwarcie sesji). Następnie serwer SMTP wysyła wiadomość, która potwierdza otwarcie sesji. Kolejną rzeczą jest identyfikacja hosta (wiadomość z komendą HELO, lub EHLO dla rozszerzonej wersji SMTP – ESMTP, Extended SMTP). Po zidentyfikowaniu hosta, klient SMTP wysyła wiadomość MAIL służącą do identyfikacji nadawcy. W dalszej części serwer potwierdza gotowość na odebranie maila (wysyła wiadomość zawierającą komendę 250 OK).

Slajd 7

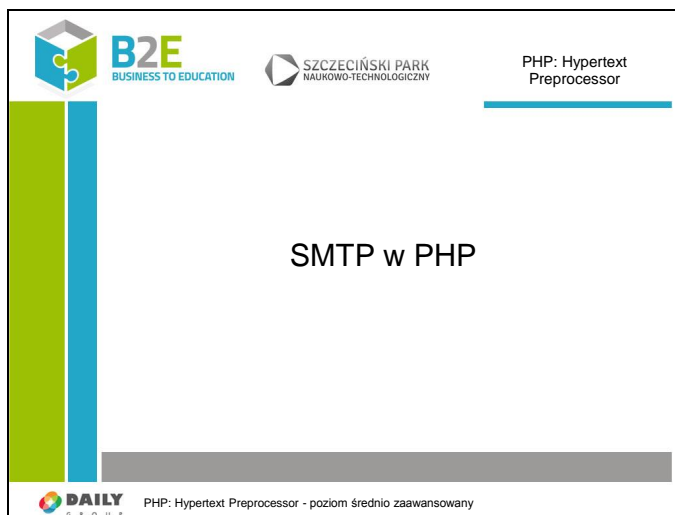
**Działanie protokołu SMTP cz.2**

6. Wysłanie informacji o odbiorcy
7. Poinformowanie o rozpoczęciu przesyłania danych
8. Przesłanie danych
9. Zakończenie sesji - QUIT

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Klient SMTP wysyła wiadomość z informacją o odbiorcy maila (RCPT TO). Serwer akceptuje dane odpowiadając komendą: 250 OK. Następnie klient SMTP wysyła wiadomość z komendą DATA <CRLF> informując o początku przesyłania danych maila. Serwer akceptuje komendę, odpowiadając komendą: 354. W końcu dane zostają przesłane, zakończone kropką i przejściem do nowej linii w formacie systemu Windows. Klient kończy sesję, wysyła wiadomość z komendą QUIT. Serwer potwierdza koniec sesji wysyłając wiadomość z kodem: 221.

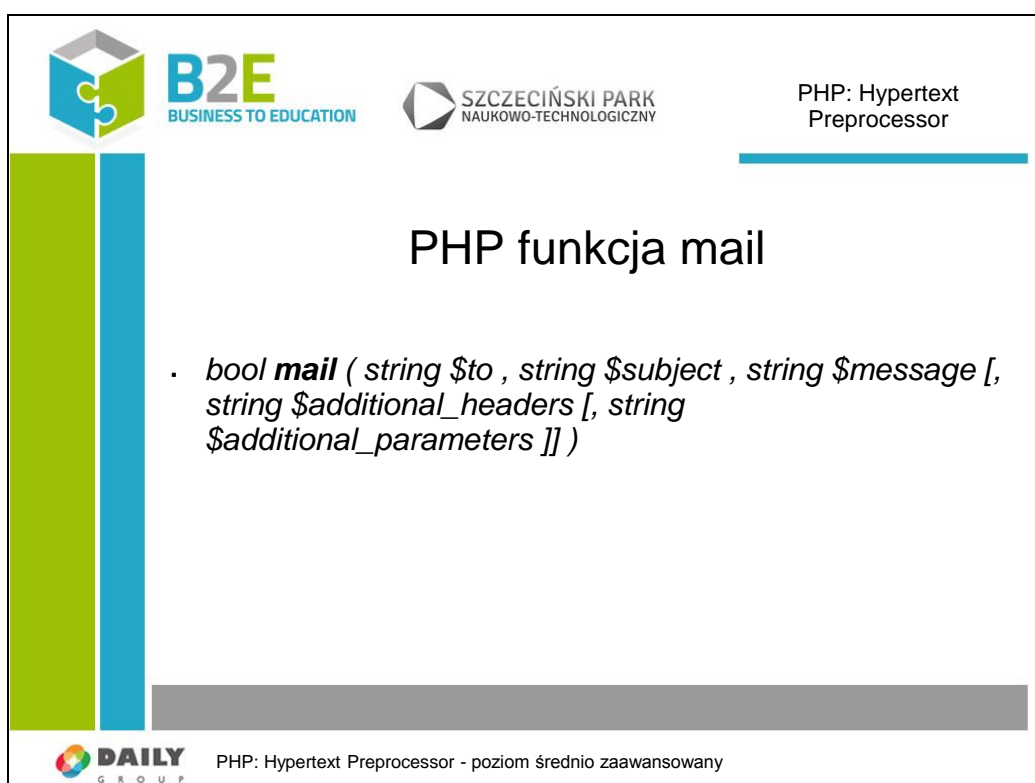
Slajd 8



SMTP w PHP

Aby PHP mogło wysyłać wiadomości via SMTP, musi mieć dostęp do serwera pocztowego albo do serwera, na którym jest zainstalowany i skonfigurowany protokół SMTP.

Slajd 9



PHP funkcja mail



- *bool mail ( string \$to , string \$subject , string \$message [, string \$additional\_headers [, string \$additional\_parameters ] ] )*

Do wysyłania wiadomości e-mail bezpośrednio z serwera, na którym SMTP jest skonfigurowane, można użyć funkcji mail. Funkcja zwraca wartość true, jeżeli wysyłanie wiadomości się powiodło, false w przeciwnym wypadku. Wymaga trzech argumentów: pierwszym jest adres e-mail osoby do której wysyłamy wiadomość, drugi to temat wiadomości, a trzeci jej treść. Posiada również dwa argumenty opcjonalne: pierwszy to dodatkowe nagłówki oraz drugi to dodatkowe parametry dla poleceń linii komend.




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 10

  PHP: Hypertext Preprocessor



## Ćwiczenie – wysyłania maila

- Używając skonfigurowanego serwera SMTP, wysłać maila na swoje konto. Użyć funkcji PHP `mail`.

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Wykonać ćwiczenie.

Slajd 11

  PHP: Hypertext Preprocessor



## Rozwiązanie ćwiczenia

```
<?php
 $email = 'adres@domena.pl';
 $subject = 'Temat maila';
 $content = 'Hello Filip';
 try{
 mail($email, $subject, $content)
 echo 'Wysłano wiadomość email!';
 } catch(Exception $e) {
 echo $e->message;
 }
?>
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności definiujemy adres pod który chcemy wysłać maila, treść maila oraz jego temat.


Następnie próbujemy wysłać maila, jeżeli nam się uda otrzymujemy komunikat o wysłaniu, w przeciwnym wypadku treść błędu, który mówi nam o tym dlaczego się nie powiodło.

PHP: Hypertext Preprocessor

## Przykład obiektu klienta SMTP cz. 1

```
<?php
 $cfg = new stdClass();
 $cfg ->SMTP->SmtServer = 'localhost';
 $cfg ->SMTP->SmtPort = '25';
 $cfg ->SMTP->SmtUser = 'me@localhost.com';
 $cfg ->SMTP->SmtPass = 'aaaaa';
?>
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przykład zobrazuje nam całą komunikację pomiędzy serwerem SMTP, a klientem i wyświetli w postaci tablicy. Został skompresowany na potrzeby zobrazowania niskopoziomowego mechanizmu komunikacji z serwerem oraz uszczuplony o mechanizmy zabezpieczające aplikacje. Nie należy korzystać z niego jako z gotowej aplikacji. W pierwszej kolejności stworzymy plik konfiguracyjny dla naszego obiektu umieszczamy go w osobnym pliku i wypełniamy odpowiednimi danymi., które będą nam potrzebne do wysłania maila. Użyta w przykładzie klasa StdClass jest podstawową klasą dostępną w PHP (podobnie jak Object w Javie). Jest obiektywnym odpowiednikiem tablicy, często używana do ustawiania parametrów konfiguracyjnych.

Np:

```
$book = new stdClass;
$book->title = "Harry Potter and the Prisoner of Azkaban";
$book->author = "J. K. Rowling";
$book->publisher = "Arthur A. Levine Books";
$book->amazon_link = "http://www.amazon.com/dp/0439136369/";
```

Następnie uzyskać można dostęp do odpowiednich pól poprzez: \$book->title  
Można również rzutować tablicę na obiekt StdClass, np:

```
$array = array(
 "title" => "Harry Potter and the Prisoner of Azkaban",
 "author" => "J. K. Rowling",
 "publisher" => "Arthur A. Levine Books",
 "amazon_link" => "http://www.amazon.com/dp/0439136369/"
);
$books = (object) $array;
Źródło:
```



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego  
<http://www.webmaster-source.com/2009/08/20/php-stdclass-storing-data-object-instead-array/>

Slajd 13



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---

## Przykład obiektu do klienta SMTP cz. 2

```
<?php
class MailForm{
 public static function getForm(){
 $form = '<form method="post" action=""><table>
<tr><td>To: </td><td><input type="text" name="to" /></td></tr>
<tr><td>From: </td><td><input type="text" name="from" /></td></tr>
<tr><td>Subject: </td><td><input type="text" name="sub" /></td></tr>
<tr><td>Message: </td><td><textarea
name="message"></textarea></td></tr>
<tr><td colspan="2"><input type="submit" value=" Send " /></td></tr>
</table></form>';
 return $form;
 }
}
?>
```



DAILY  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Potrzebujemy obiekt formularza, który będzie przetwarzany na wiadomości e-mail. Powyższy obiekt posiada jedną metodę, która zwraca nam formularz HTML w postaci ciągu znaków PHP. Umieszczamy go w osobnym pliku.






PHP: Hypertext Preprocessor



## Przykład obiektu klienta SMTP cz. 3

```
<?php
class SMTPClient
{
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Część pierwsza klasy

PHP: Hypertext Preprocessor

```
function setConfig($config) {
 $this->SmtpServer = $config->SMTP->SmtpServer;
 $this->SmtpUser = base64_encode ($config->SMTP->SmtpUser);
 $this->SmtpPass = base64_encode ($config->SMTP->SmtpPass);
 if ($config->SMTP->SmtpPort == ""){
 $this->PortSMTP = 25;
 } else {
 $this->PortSMTP = $config->SMTP->SmtpPort;
 }
}

function setFrom($from) {
 $this->from = $from;
}
```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część druga klasy, procedura, która ustawia odpowiedni parametry i dane, potrzebne do wysłania naszego maila.




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 16



**B2E**  
BUSINESS TO EDUCATION



**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

---

```
function setTo($to) {
 $this->to = $to;
}

function setSubject($subject) {
 $this->subject = $subject;
}


function setMailContent($body) {
 $this->body = $body;
}
```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część druga klasy, procedura, która ustawia odpowiedni parametry i dane, potrzebne do wysłania naszego maila.

Slajd 17



**B2E**  
BUSINESS TO EDUCATION




**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---



```
function SendMail () {
 $talk = array();
 $HTTP_HOST = $_SERVER['HTTP_HOST'];
 try {
 if($this->SmtServer == null || strlen($this->SmtServer) == 0){
 throw new Exception('SmtServer is missing, please use setConfig() method.');
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Część trzecia klasy. Próbujemy otworzyć połączenie z serwerem SMTP za pomocą funkcji `fsockopen`, a jeżeli nam się powiedzie zaczynamy się z nim komunikować. Funkcja `fwrite` wysyła komendy z parametrami, a funkcja `fgets` pobiera odpowiedź.

Slajd 18



PHP: Hypertext Preprocessor

```
if ($SMTPPIN = fsockopen ($this->SmtpServer, $this->PortSMTP, $errno,
 $errstr, 10)) {
 fwrite($SMTPPIN, "EHLO ".$HTTP_HOST."\r\n");
 $stalk["hello"] = fgets ($SMTPPIN, 1024);
 fwrite($SMTPPIN, "auth login\r\n");
 $stalk["res"]=fgets($SMTPPIN,1024);
 fwrite($SMTPPIN, $this->SmtpUser."\r\n");
 $stalk["user"]=fgets($SMTPPIN,1024);
 fwrite($SMTPPIN, $this->SmtpPass."\r\n");
 $stalk["pass"]=fgets($SMTPPIN,256);
 fwrite ($SMTPPIN, "MAIL FROM: <".$this->from.">\r\n");
 $stalk["From"] = fgets ($SMTPPIN, 1024);
 fwrite ($SMTPPIN, "RCPT TO: <".$this->to.">\r\n");
 $stalk["To"] = fgets ($SMTPPIN, 1024);
 fwrite($SMTPPIN, "DATA\r\n");
 $stalk["data"]=fgets($SMTPPIN,1024);
 fwrite($SMTPPIN, "To: <".$this->to.">\r\nFrom: <".$this->from.">\r\nSubject: ".$this->subject."\r\n\r\n\r\n".$this->body."\r\n.\r\n");
 $stalk["send"]=fgets($SMTPPIN,256);
 fwrite ($SMTPPIN, "QUIT\r\n");
 fclose($SMTPPIN);
}
```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część trzecia klasy. Próbujemy otworzyć połączenie z serwerem SMTP za pomocą funkcji `fsockopen`, a jeżeli nam się powiedzie zaczynamy się z nim komunikować. Funkcja `fwrite` wysyła komendy z parametrami, a funkcja `fgets` pobiera odpowiedź.




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 19



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

```

else {
 $talk["error"] = "$errstr ($errno) \n";
}
}
catch (Exception $e) {
 $talk["exception"] = $e->getMessage(). " \n";
}

return $talk;
}


```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część trzecia klasy. Próbujemy otworzyć połączenie z serwerem SMTP za pomocą funkcji fsockopen, a jeżeli nam się powiedzie zaczynamy się z nim komunikować. Funkcja fwrite wysyła komendy z parametrami, a funkcja fgets pobiera odpowiedź.

Slajd 20



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

### Przykład obiektu klienta SMTP cz. 4

```

<?php
require_once('config.php');
require_once('App.php');
$app = new App($cfg);
?>

```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Implementujemy główne wywołanie naszego klienta SMTP. W przykładzie można ten kod umieścić w pliku index.php w katalogu naszej aplikacji.

Importujemy plik konfiguracyjny oraz klasę naszej aplikacji, którą zaraz zaimplementujemy. Tworzymy jej egzemplarz i wywołujemy metodę inicjalizującą.



## Przykład obiektu klienta SMTP cz. 5


```
<?php
class App{
 private $config;
 private $form;
 private $data;
 public function __construct($config){
 $this->config = $config;
 $this->data = new stdClass();
 $this->init();
 }
}
```




Pierwsza część klasy. Rozpoczynamy implementację obiektu App. W przykładzie powyżej znajdują się zmienne wewnętrzne jak dane konfiguracyjne, formularz i dane do wysłania. Plik konfiguracyjny przesyłamy za pomocą konstruktora obiektu, który wywołany jest w momencie jego tworzenia. Funkcja aplikacji mogła być obiektem typu singleton, lecz ze względu na brak potrzeby, mogło by to tylko zaciemnić obraz przykładu.



Slajd 22



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

---

```

private function init(){
 $this->loadClasses(array('SMTPClient',
 'MailForm'));
 $this->form = MailForm::getForm();
 if($_SERVER["REQUEST_METHOD"] == "POST"){
 $this->getPostData();
 $this->sendMail();
 } else {
 $this->display();
 }
}


```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część druga implementacji klasy App. Funkcja inicjalizująca aplikację. W pierwszej kolejności ładuje odpowiednie klasy, przekazane w argumencie jako tablica. Następnie pobiera formularz potrzebny do wysłania maila. Metoda sprawdza czy zostało coś wysłane metodą POST, jeżeli tak to pobiera te dane, przetwarza je i wysyła maila. W przeciwnym wypadku wyświetla formularz.

Slajd 23



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

---

```

private function loadClasses($arr){
 foreach($arr as $class)
 require_once($class.'.php');
}
private function display(){
 echo $this->form;
}

```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część trzecia implementacji klasy App. Pierwsza metoda ładuje klasy podane w tablicy jako argumenty.

Druga metoda wyświetla formularz, wcześniej wczytany do obiektu.



```
private function sendMail(){

 $SMTPMail = new SMTPClient ($this->config->SMTP->
 >SmtpServer, $this->config->SMTP->SmtpPort, $this->
 >config->SMTP->SmtpUser, $this->config->SMTP->SmtpPass,
 $this->data->from, $this->data->to, $this->data->
 >subject, $this->data->body);

 $SMTPChat = $SMTPMail->SendMail();


 print_r($SMTPChat);

}
```


Część czwarta implementacji klasy App. Tworzymy egzemplarz wcześniej utworzonej klasy i wywołujemy jej metodę sendmail. Następnie wypisujemy tablicę z odpowiedziami od serwera SMTP.



Slajd 25



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

```

private function getPostData() {
 if($_SERVER["REQUEST_METHOD"] == "POST")
 {
 $this->data->to = $this->test_input($_POST['to']);
 $this->data->from = $this->test_input($_POST['from']);
 $this->data->subject = $this->test_input($_POST['sub']);
 $this->data->body = $this->test_input($_POST['message']);
 }
}

private function test_input($data) {
 $data = trim($data);
 $data = stripslashes($data);
 $data = htmlspecialchars($data);
 return $data;
}



```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ostatnia część trzecia implementacji klasy App. Funkcja przetwarzająca dane z formularza. Plik zapisujemy jako np. App.php. Nareszcie możemy zobaczyć efekt działania naszej aplikacji, pod warunkiem, że serwer SMTP jest poprawnie skonfigurowany oraz wprowadziliśmy poprawne dane w pliku konfiguracyjnym i formularzu.




PHP: Hypertext Preprocessor

## Przykład użycia PHPMailera cz. 1

```
<?php
require_once('../class.phpmailer.php');



$mail = new PHPMailer();

$body = file_get_contents('szablon_wiadomosci.html');
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


W pierwszej kolejności dołączamy klasę phpmailera do naszego skryptu, używamy `require_once` – jeśli plik był już wcześniej dołączony, nie dołączamy go ponownie. Następnie tworzymy egzemplarz klasy i pobieramy szablon html, bądź txt naszego maila w zależności od potrzeb.

PHP: Hypertext Preprocessor

## Przykład użycia PHPMailera cz. 2

```
$mail->IsSMTP();
$mail->Host = "mail.nazwa_domeny.pl";
$mail->SMTPAuth = true;
$mail->Port = 25;
$mail->Username = "mail.nazwa_domeny.pl";
$mail->Password = "haslo";
```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Ustalamy protokół na SMTP, włączamy autoryzację oraz konfigurujemy podstawowe parametry dla konta SMTP., takie jak host, port, nazwę użytkownika i hasło.



Slajd 28



B2E  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład użycia PHPMailera cz. 3

```
$mail->From('filip@nazwa_domeny.pl');
$mail->AddReplyTo(„ewa@nazwa_domeny.pl”, 'Ewa');
$mail->Subject = "Temat";
$mail->MsgHTML($body);
```




DAILY GROUP


PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W tym miejscu ustalamy adres nadawcy, adres pod który należy wysłać odpowiedź, temat wiadomości oraz treść wiadomości, w tym przypadku wiadomość w formacie HTML.

Slajd 29



B2E  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład użycia PHPMailera cz. 4

```
$address = "adres_odbiornicy";
// dodawanie adresów i załączników
$mail->AddAddress($address, "Ewa L");
$mail->AddAttachment("sciezka_do_zalacznika");
```

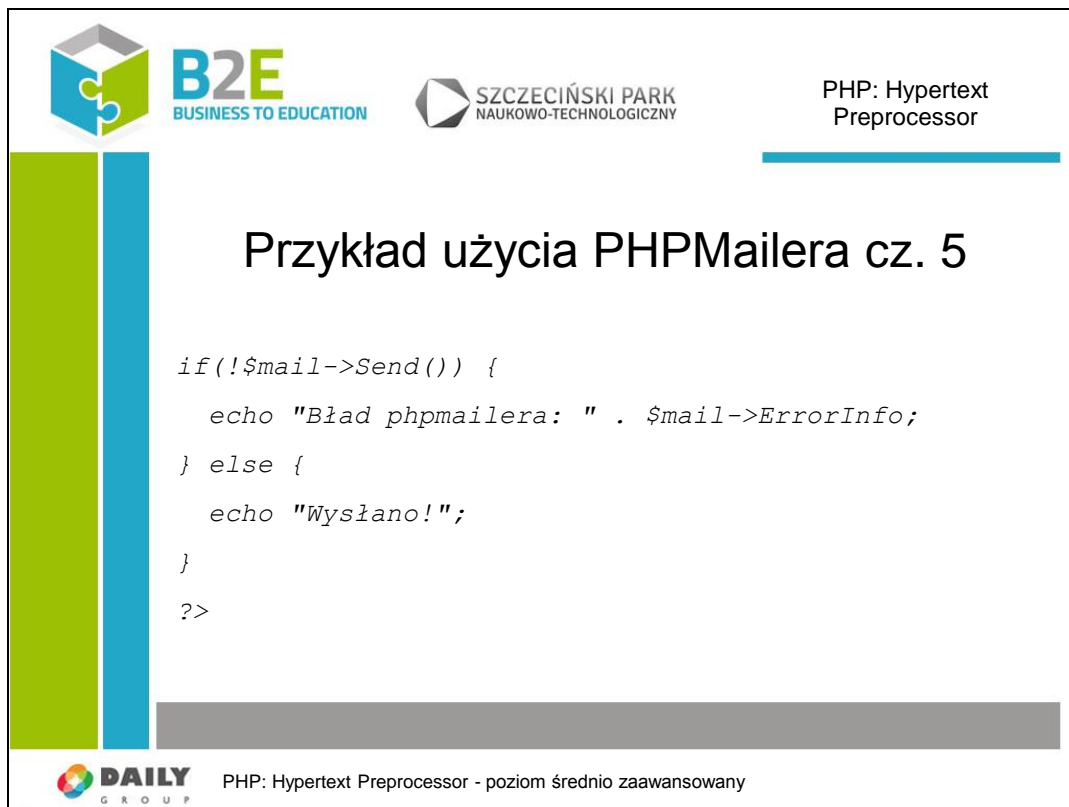


DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Możemy dodawać wiele adresów, bądź załączników do wiadomości e-mail. Należy jednak pamiętać, że wielkość załącznika jest konfigurowalna po stronie serwera, co może spowodować niemożliwość przyjęcia go przez serwer SMTP. Dodatkowo podczas tworzenia skryptu walidującego rozmiar załączanego pliku, należy zwrócić uwagę na fakt, że ze względu na specyfikę kodowania załączników w mailach (base64) jego rozmiar rośnie o ok. 30%.

Slajd 30

The slide features a header with logos for B2E (Business to Education), Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. A vertical bar on the left is split into green and blue sections. The main title is 'Przykład użycia PHPMailera cz. 5'. Below it is a PHP code snippet. At the bottom left is the DAILY GROUP logo, and at the bottom right is the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład użycia PHPMailera cz. 5

```
if(!$mail->Send()) {
 echo "Błąd phpmailera: " . $mail->ErrorInfo;
} else {
 echo "Wysłano!";
}
?>
```

**DAILY**  
GROUP

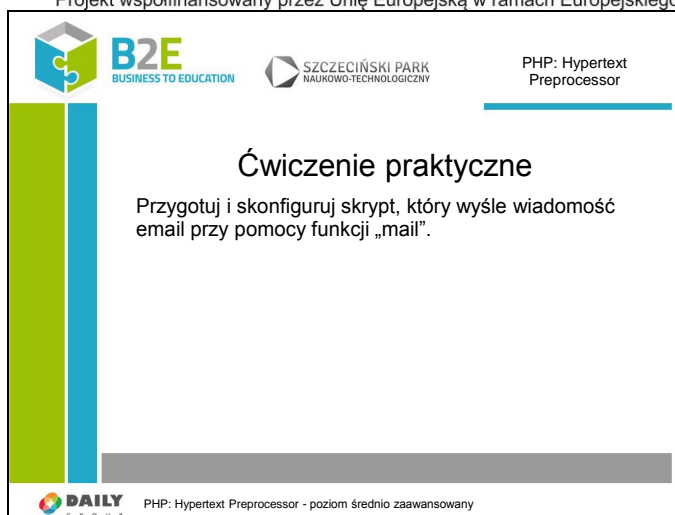
PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Po odpowiednim skonfigurowaniu naszego obiektu próbujemy wysłać wiadomość, w przypadku niepowodzenia otrzymuje błąd zdefiniowany w Klasie PHPMailer, w przeciwnym razie otrzymujemy komunikat o wysłaniu wiadomości.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

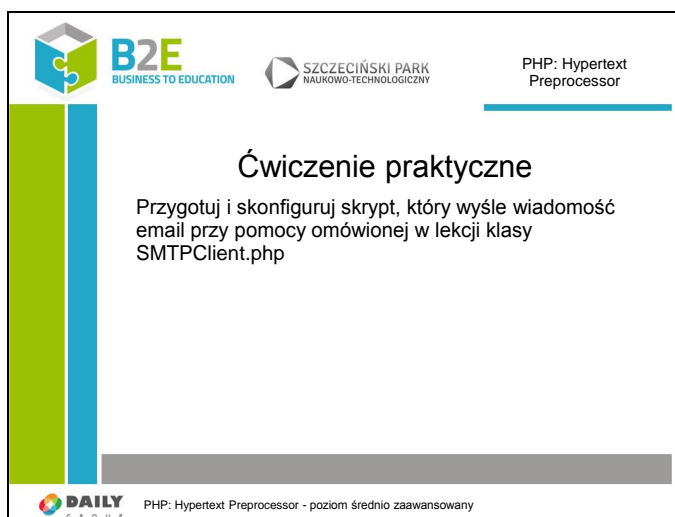
Slajd 31



Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are at the top. The slide title is 'PHP: Hypertext Preprocessor'. The main heading is 'Ćwiczenie praktyczne'. The text reads: 'Przygotuj i skonfiguruj skrypt, który wyśle wiadomość email przy pomocy funkcji „mail”'. At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

mail.php

Slajd 32



Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are at the top. The slide title is 'PHP: Hypertext Preprocessor'. The main heading is 'Ćwiczenie praktyczne'. The text reads: 'Przygotuj i skonfiguruj skrypt, który wyśle wiadomość email przy pomocy omówionej w lekcji klasy SMTPClient.php'. At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

smtp/index.php

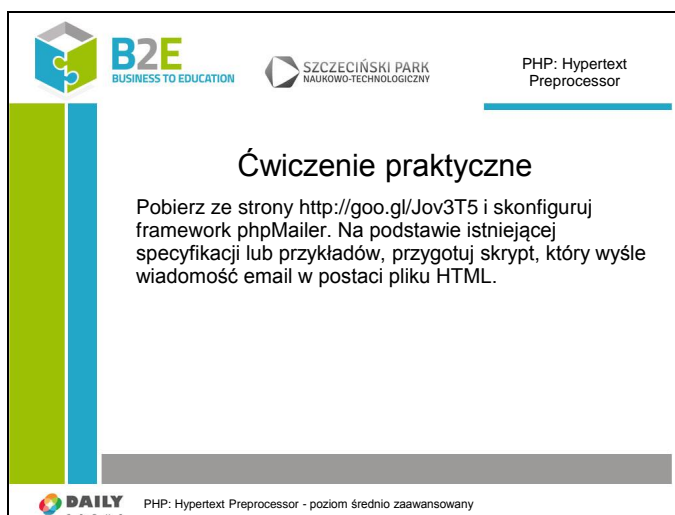
smtp/App.php

smtp/config.php

smtp/MailForm.php

smtp/SMTPClient.php

Slajd 33



Logo B2E BUSINESS TO EDUCATION and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY are at the top. The slide title is 'PHP: Hypertext Preprocessor'. The main heading is 'Ćwiczenie praktyczne'. The text reads: 'Pobierz ze strony <http://goo.gl/Jov3T5> i skonfiguruj framework phpMailer. Na podstawie istniejącej specyfikacji lub przykładów, przygotuj skrypt, który wyśle wiadomość email w postaci pliku HTML'. At the bottom, there is a logo for DAILY GROUP and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

phpMailer/index.php

phpMailer/contents.html

The slide features a white background with a green and blue vertical bar on the left. At the top left are logos for B2E (Business to Education) and Szczeciński Park Naukowo-Technologiczny. At the top right is the text 'PHP: Hypertext Preprocessor'. The center contains the text 'Dziękujemy za uwagę'. Below this are logos for 'KAPITAŁ LUDZKI' (Human Capital) and 'UNIA EUROPEJSKA' (European Union), with the text 'Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego'. At the bottom left is the 'DAILY GROUP' logo, and at the bottom right is the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

## Ćwiczenia

Przykłady i ćwiczenia zawarte są na slajdach tej lekcji.

## Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie nauczą się jak działa protokół SMTP oraz będą potrafili wykorzystać go w swoich skryptach. Nauczą się również jak zaimplementować obiektowo małą aplikację, w tym przypadku prostego klienta do wysyłania poczty elektronicznej.

## Lekcja 8 - MIME - rozszerzenia multimedialne

**(Struktura nagłówek MIME, Kodowanie Base64 i Quoted-Printable, Wiadomości wieloczęściowe, Obsługa poczty multimedialnej w PHP)**

## Cel lekcji

Celem lekcji jest przedstawienie protokołu MIME oraz jego rozszerzenia multimedialnego. Uczniowie poznają kodowania Base64 i Quoted-Printable. Nauczą się wysyłać pocztę multimedialną oraz poznają strukturę wiadomości wieloczęściowych.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Treść - slajdy z opisem

Slajd 1

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor  
- poziom średnio zaawansowany  
Lekcja 8  
MIME - rozszerzenia multimedialne

KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

DAILY GROUP

Słowa wstępu.

Slajd 2

MIME

- Co to jest?
- Wprowadzenie

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

MIME (ang. Multipurpose Internet Mail Extensions) jest to standard do przesyłania poczty elektronicznej, który definiuje budowę komunikatu. Składa się z nagłówek i treści. Standardowo mail wysyła treść jako tekst zakodowany w 7-bitowym kodzie ASCII. MIME posiada kodowanie 8-bitowe, dlatego zwiększa objętość maila.

Źródło zdjęcia:

[http://www.costelloentertainments.co.uk/3770/174/artists/street-performers---](http://www.costelloentertainments.co.uk/3770/174/artists/street-performers---walkabout-entertainers.aspx)

[walkabout-entertainers.aspx](http://www.costelloentertainments.co.uk/3770/174/artists/street-performers---walkabout-entertainers.aspx)

Slajd 3

Typ MIME

- Co to jest?
- Do czego służy?

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Typ MIME (nazwany od standardu MIME) – internet media type lub Content-Type (po nazwie nagłówek kilku protokołów, którego wartość jest tego typu) jest to dwuczęściowy identyfikator typu pliku w sieci, pierwsza część opisuje jego typ, a druga konkretny podtyp. Np. Image/gif. Image mówi, że jest to typ obrazkowy o podtypie gif, który jest zdefiniowany jako standard dla obrazów o tym rozszerzeniu. Początkowo

typ ten był używany w protokole SMTP, a później zaczęto z niego korzystać również przy protokole HTTP. Służy do określania standardu przesyłanego pliku w sieci Internet.

Slajd 4

Struktura nagłówków MIME

MIME-version: 1.0

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Struktura nagłówków MIME jest identyczna ze strukturą nagłówków HTTP. Powyższy nagłówek określa wersję stosowanego standardu, w tym przypadku wersja 1.0.

Slajd 5

Nagłówki MIME

Wymagane:

- MIME-version
- Content-Type

Opcjonalne:

- Content-Disposition
- Content-Transfer-Encoding
- Content-ID
- Content-Description

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W najnowszym standardzie wymagane nagłówki to wersja MIME oraz typ przesyłanej treści. Opcjonalne: Kodowanie MIME zawiera przeważnie jeden z dwóch parametrów base64 lub quoted-printable. Jest jeszcze jeden mniej popularny za to najprostsz: 8bit, nie wymagający żadnej konwersji. Nagłówek Content-ID wyznacza wiadomość właściwą, a Content-Description komentarz do wysyłanego komunikatu.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6

**Składnia nagłówka Content-Type**

1. Nazwa typu mediów
2. Nazwa podtypu mediów
3. Wymagane parametry (nie każdy typ tego wymaga)
4. Opcjonalne parametry (nie każdy typ tego wymaga)

PHP: Hypertext Preprocessor

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nagłówek Content-Type wymaga typu i podtypu mediów. Niektóre z nich wymagają parametrów, które podajemy w następnej kolejności. Na ostatnim miejscu umieszczamy parametry opcjonalne.

Slajd 7

**Możliwe wartości Content-Type cz. 1**


- . text
- . image
- . audio
- . video
- . application
- . multipart
- . message

PHP: Hypertext Preprocessor


DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część odpowiedzialna za typ, nagłówek Content-Type może przyjmować jedną z powyższych wartości. Czasami dopuszczalne są dwie ostatnie opcje multipart oraz message. Przez użycie typu multipart, MIME pozwala, by wiadomość posiadała wiele części, z których każda może mieć określony swój własny typ MIME.





**B2E**  
BUSINESS TO EDUCATION

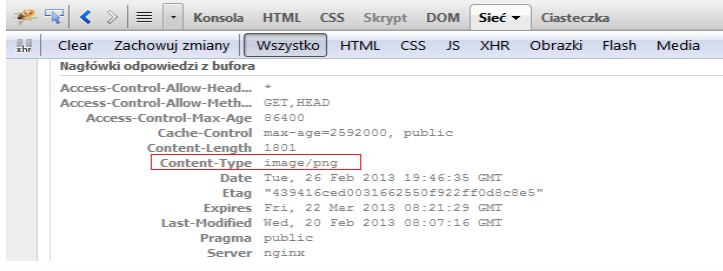



**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor


## Ćwiczenie

- Otwórz dowolną stronę www i przy pomocy narzędzia firebug znajdź content-type dla różnych typów plików, np. grafiki, plików tekstowych itd.






PHP: Hypertext Preprocessor - poziom średnio zaawansowany



**B2E**  
BUSINESS TO EDUCATION

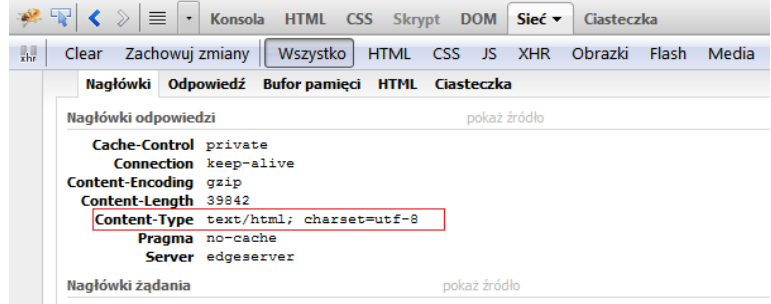



**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Ćwiczenie cd

### Przykład dla pliku tekstowego html:





PHP: Hypertext Preprocessor - poziom średnio zaawansowany



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 10

Slide 10 content: B2E logo, Szczeciński Park Naukowo-Technologiczny logo, PHP: Hypertext Preprocessor. Title: Możliwe wartości Content-Type cz. 2. List item: Podtypów MIME jest bardzo dużo, podobnie jak możliwych rozszerzeń plików, cały wykaz znajduje się w dokumentacji: <http://goo.gl/P5iCtw>. Footer: DAILY GROUP, PHP: Hypertext Preprocessor - poziom średnio zaawansowany.

Slajd 11

Slide 11 content: B2E logo, Szczeciński Park Naukowo-Technologiczny logo, PHP: Hypertext Preprocessor. Title: Możliwe wartości Content-Type cz. 3. List item: `charset="us-ascii"`. Footer: DAILY GROUP, PHP: Hypertext Preprocessor - poziom średnio zaawansowany.

Powyższy przykład obrazuje zastosowanie parametrów w nagłówku Content-Type, w tym przypadku jest to kodowanie znaków.

Slajd 12

Slide 12 content: B2E logo, Szczeciński Park Naukowo-Technologiczny logo, PHP: Hypertext Preprocessor. Title: Kodowanie Base64. List items: Co to jest?, Do czego służy?, Jak działa?. Footer: DAILY GROUP, PHP: Hypertext Preprocessor - poziom średnio zaawansowany.

Kodowanie base64 to rodzaj kodowania transportowego, oznacza to, że służy do bezpiecznego przesyłania danych. Zabezpiecza dane szczególnie ze względu na uszkodzenia spowodowane niemożnością przesyłania 8 bitowych danych przez starsze maszyny, które przesyłały znaki 7-bitowe. Kodowanie to dzieli dowolne 64 znaki na grupy po trzy bajty. Bajt posiada 8 bitów, dlatego grupa ma 24 bity. Co po podzieleniu na cztery skutkuje 6 bitowymi jednostkami. Na sześciu bitach możemy zapisać

64 różne znaki, czyli każda jednostka może przyjąć 64 wartości, stąd też nazwa kodowania. Wszystkim takim jednostkom przypisywane są znaki, na podstawie początkowo ustalonych liczb. Jeśli rozmiar wejściowego ciągu bajtów nie jest podzielny przez 3, stosowane jest dopełnienie, aż do jego podzielności przez 4.

Slajd 13

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Przykład kodowania Base64

- Hello World!

W kodowaniu Base64:

- SGVsbG8gV29ybGQh

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Powyższy przykład obrazuje wynik kodowania ciągu znaków przy pomocy kodowania Base64.

Slajd 14

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

### Zastosowania kodowania Base64


- Przesyłanie załączników binarnych w mailach
- Kodowanie haseł przy wysyłaniu protokołem SMTP

**DAILY** GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Przesyłanie załączników binarnych w mailach – np. grafiki, muzyki mp3 itp



Slajd 15



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---


## Ćwiczenie

Otwórz plik mail.eml zawierający przykładowy e-mail z programu pocztowego. Znajdź fragment w którym znajduje się zakodowany w Base64 załącznik.

```


--_004_4EBD81F4E859284EBCD02845B7EF01A2453FD6A0AM2PRD0311MB421
Content-Type: application/octet-stream; name="Week meeting 06022013.pdf"
Content-Description: Week meeting 06022013.pdf
Content-Disposition: attachment; filename="Week meeting 06022013.pdf";
size=85382; creation-date="Mon, 11 Feb 2013 06:13:22 GMT";
modification-date="Mon, 11 Feb 2013 06:13:22 GMT"
Content-Transfer-Encoding: base64

JVBERi0xLjQKJwEJLjz9MKMyAwIG9iago8PAovUHJvZHVjZSIgKFBER11YQ2hhbmdlIDQuMC4yMDEu
MCBcKFdpbmRvd3MgU2V2ZW4gVWx0aW1hdGUgeDY0IFN1cnZpY2UgUGFjayAxIFwQnVpbGQgNzYw
MVVpXCxpCi9DcmVhdG1vbkRhdGUgKEQ6MjAxMzAyMTEwMTEwMDEwMDEwMDEwMDEwMDEwMDEw
NSAwIG9iago8PAovVHlwZSAvWE9iamVjdAovU3VidHlwZSAvSW1hZ2UKL1dpZHRoIDk2OQovS2Vp
Z2h0IDYzMQovQml0c1BlckNvbXBvbmVudCA4Ci9Db2xvc1NwYWNlIC9EZXZpY2VSR0IKL0x1bmd0
aCA2ODgzMAovRmlsdGVyIFsvRmxdGVEZWNvZGUgL0RDVER1Y29kZV0KPCj4K3RyZWFTCnJa7P0H
VFtLti+ML0RFQUARkNwqIFkkZ1pEstgSJCdBRUBjyq2ioQqObRIRoIKTYWYSA5KTnnnOmw+mv2
Pufcvfc5997z3rv/8X3jjT+Ohay1Z1fVnFU152/OmlWN/4mfB16xKqooAkREAEBE+Afqwx854ML5
8yTnz10gISG5ePEC6SVg8ktkZJfoqa5SUjMzQF1YGZiYrRpd5rh+k5eViY1TjIv3jCwDCEQ0JG
XFD6tpCw4GkhRBcvXrxEdomOnJxO8AbTDcH/5R/8N+DKhTPux03ERDeBM1e1iK8Q4ZsACKGd54h+
```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slajd 16



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

---

## Kodowanie Quoted-Printable

- Co to jest?
- Wprowadzenie



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Quoted-Printable – jest kolejnym kodowaniem transportowym zgodnym ze standardem MIME. Polega ono na tym, że pozostawia wszystkie bajty mniejsze od 128 (7-bitów), które nie są znakami sterującymi ASCII (oprócz spacji), a resztę zamienia na zakodowany szesnastkowo napis ASCII poprzedzony znakiem równości. Znak równości też jest kodowany, w celu uniknięcia wieloznaczności jako =3D. Kodowanie to jest głównie wykorzystywane w poczcie elektronicznej do kodowania przesyłanych wiadomości, które zawierają znaki narodowe z poza ASCII. Nie nadaje się do przesyłania plików binarnych ze względu

na jego narzut około 225%,  
gdzie np. Base64 posiada tylko  
33% narzutu.

Slajd 17



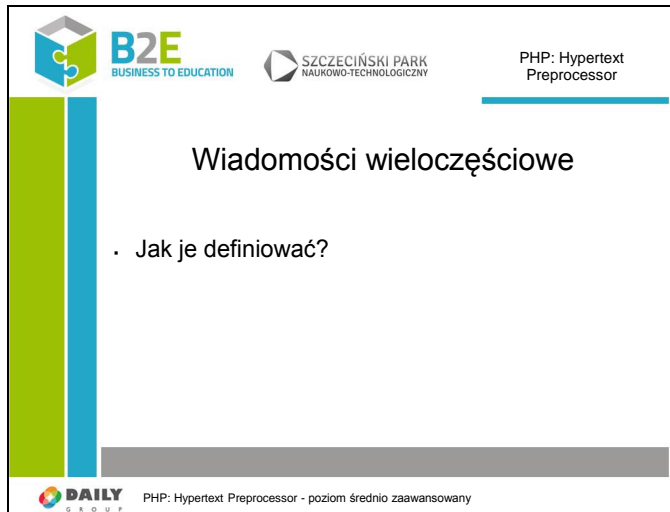
Przykład kodowania Quoted-Printable

- Hello= world!
- Hello=3D world!

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przykład zakodowania hello world w kodowaniu Quoted – Printable, znaki alfabetu oraz znak '!', mają kod ASCII poniżej 128, dlatego dla lepszego zobrazowania, że coś się dzieje, dodano znak równości.

Slajd 18



Wiadomości wieloczęściowe


- Jak je definiować?

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Wiadomości wieloczęściowe określone jako typ multipart w nagłówku Content-Type. Dodatkowo w tym samym nagłówku muszą posiadać parametr określający separator 'boundary', którego nie wolno umieszczać w treści wiadomości. Służy on do oddzielania od siebie wiadomości.



Slajd 19



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład wiadomości wieloczęściowej cz. 1


From: zuraw@domena.pl  
To: czapla@domena.pl Subject: =?iso-8859 -  
2?Q?Czaplo=2C\_czy\_um=F3wisz\_si=EA\_ze\_mn=B1=3F?=  
MIME Version: 1.0  
Content-Type: multipart/mixed;  
boundary="xxxToJestSeparator0000xxx"  
This is a message with multiple parts in MIME format.  
--xxxToJestSeparator0000xxx  
Content-Type: multipart/alternative;  
boundary="xxxToJestSeparatorZagniezdzony1111xxx"




DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Na początku określamy adresata, odbiorcę oraz wersję MIME z której skorzystamy. Typ wiadomości ustawiony jest na multipart oraz podtyp na mixed (mieszany), który określa, że części wiadomości mogą być różnych typów. W sekcji typu umieszczamy parametr boundary, który definiuje separator w tym przypadku główny, gdyż wiadomościach można zagnieżdżać również wiadomości.

Slajd 20



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład wiadomości wieloczęściowej cz. 2

--xxxToJestSeparatorZagniezdzony1111xxx  
Content-Type: text/plain; charset="iso-8859-2"  
Content-Transfer-Encoding: quoted-printable  
To jest tre=B6=E6 wiadomo=B6ci.  
--xxxToJestSeparatorZagniezdzony1111xxx  
Content-Type: text/html; charset="iso-8859-2"  
Content-Transfer-Encoding: quoted-printable



DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany




## Przykład wiadomości wieloczęściowej cz. 3


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD>
<META http-equiv=3DContent-Type content=3D"text/html;
charset=3Diso-8859-2"></HEAD>
<BODY>To jest tre=B6=E6
wiadomo=B6ci.</BODY></HTML>
--xxxToJestSeparatorZagniezdzony1111xxx--
```

Tutaj widzimy przykład zagnieżdżonej wiadomości typu HTML.



Slajd 22





SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

## Przykład wiadomości wieloczęściowej cz. 4

```

--xxxToJestSeparator0000xxx
Content-Type: image/gif; name="obrazek.gif"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="obrazek.gif"
PGh0bWw+CiAgPGhYVWQ+CiAgPC9oZWFKPgogIDxib2R5PgogICAAP
HA+VGhpcyBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0b
Ww+Cg==
--xxxToJestSeparator0000xxx--

```





PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jak widać w powyższym przykładzie po raz pierwszy użyto kodowania base64, zamiast quoted-printable. Użyto go do zakodowania obrazu typu gif.

Ćwiczenie – zdekoduj zawartość pliku decodeBase64.txt do postaci pliku graficznego gif.

Slajd 23





SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


PHP: Hypertext  
Preprocessor

## Przykład wysyłania wiadomości HTML w PHP cz. 1

```

<?php
$to = 'stefan@domena.pl' . ', ';
$to .= 'marian@domena.pl';
$subject = 'Birthday Reminders for August';

```





PHP: Hypertext Preprocessor - poziom średnio zaawansowany



Na początek definiujemy odbiorców oraz ustalamy temat maila, ponieważ do przesłania wiadomości multimedialnej w tym przypadku w formacie html, użyjemy funkcji mail w PHP.

Slajd 24




**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład wysłania wiadomości HTML w PHP cz. 2

```
$message = '
<html>
<head><title>Tytuł</title></head>
<body>
 <p>Hello world!</p>
</body>
</html>
';
```




PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Jako treść wiadomości definiujemy sobie kod w formacie htmla.



Slajd 25



B2E  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład wysyłania wiadomości HTML w PHP cz. 3

```
$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=utf8' .
"\r\n";
$headers .= 'From: Marek<marek@domena.pl>' . "\r\n";
```




DAILY  
GROUP


PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W nagłówkach ustalamy wersję MIME, wymagany w tym przypadku typ text/html z kodowaniem w utf8 oraz nadawcę maila.

Slajd 26



B2E  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Przykład wysyłania wiadomości HTML w PHP cz. 4

```
if (mail($to, $subject, $message, $headers))
 echo 'Wysłano wiadomość!';
else
 echo 'Wysyłanie wiadomości nie powiodło się!';
?>
```





DAILY  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Na końcu wysyłamy wiadomość.


Slajd 27



PHP: Hypertext Preprocessor

### Ćwiczenie praktyczne



Znajdź dowolny, nieduży plik graficzny GIF i przygotuj skrypt, który zakoduje grafikę do ciągu znaków Base64.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

base64Encode.php


Slajd 28



PHP: Hypertext Preprocessor

### Ćwiczenie praktyczne



Przygotuj skrypt, który zdekoduje podany w pliku decodeBase64.txt ciąg znaków do pliku graficznego GIF. Skrypt powinien wyświetlić grafikę w przeglądarce lub uruchomić pobieranie pliku.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

decodeToGif.php


Slajd 29



PHP: Hypertext Preprocessor

### Ćwiczenie praktyczne

Przygotuj skrypt, który zakoduje w Base64 dowolny plik binarny (grafika, muzyka) o rozmiarze ok 1MB. Wygenerowany w Base64 ciąg znaków zapisz do pliku TXT. Porównaj różnicę w rozmiarach obu plików.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

decodeToGif.php



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 30

Slide 30 content: The slide features a green and blue vertical bar on the left. At the top left are logos for BZE (BUSINESS TO EDUCATION) and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY. At the top right is the text 'PHP: Hypertext Preprocessor'. The main text in the center reads 'Dziękujemy za uwagę'. At the bottom, there are logos for 'KAPITAŁ LUDZKI' and 'UNIA EUROPEJSKA' with the text 'Człowiek - najlepsza inwestycja' and 'Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego'. The footer includes the 'DAILY GROUP' logo and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

## Ćwiczenia

Jako ćwiczenia do lekcji, należy wykonać razem z uczniami przykłady znajdujące się na slajdach.

## Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili wysyłać pocztę multimedialną np. w postaci plików html. Uczeń powinien poznać możliwe rozszerzenia plików, podstawowe nagłówki MIME oraz podstawowe kodowania transportowe. Umieć rozróżnić wiadomość wieloczęściową, od jednoczęściową.

## Lekcja 9 - UML 2.1 (Diagramy czynności, Projektowanie złożonych aplikacji)

### Cel lekcji

Celem lekcji jest zapoznanie ucznia z językiem do projektowania UML. Nauczenie go na czym polegają podstawowe diagramy oraz jak zaprojektować złożone aplikacje.

### Treść - slajdy z opisem

Slajd 1

Slide 1 content: The slide features a green and blue vertical bar on the left. At the top left are logos for BZE (BUSINESS TO EDUCATION) and SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY. At the top right is the text 'PHP Hypertext Preprocessor'. The main text in the center reads 'PHP Hypertext Preprocessor' in large letters, followed by 'Lekcja 9 UML 2.1'. At the bottom, there are logos for 'KAPITAŁ LUDZKI' and 'UNIA EUROPEJSKA' with the text 'Człowiek - najlepsza inwestycja' and 'Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego'. The footer includes the 'DAILY GROUP' logo and the text 'PHP Hypertext Preprocessor Lekcja 8 UML 2.1'.

UML jest notacją za pomocą której możliwe jest graficzne modelowanie oprogramowania. Notacja to sposób zapisu symboli, liter i znaków za pomocą której można wyrazić treść. Treścią którą będziemy omawiać jest budowa oprogramowania.

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP  
Hypertext Preprocessor

## Czym jest UML?

- Unified Modeling Language
- Modelowanie
- Historia UML

UNIFIED  
MODELING  
LANGUAGE

DAILY  
GROUP

Lekcja 8 - UML

Rozwinięciem skrótu UML jest Unified Modeling Language czyli... Zunifikowany Język Modelowania.

Czym jest modelowanie? Opis struktur, zachowań, interakcji, zjawisk, układu, budowy za pomocą narzędzi (notacji, diagramów..) pozwalających zobrazowanie budowy układu, sprawdzenie zachowania badanego przedmiotu. Przykładem modelu jest np. model układu słonecznego.

UML jest notacją za pomocą której możliwe jest zamodelowanie różnego rodzaju systemy. Bardzo często wykorzystywany jest do zamodelowania tworzonego oprogramowania. Pozwala na zaprojektowanie budowy oprogramowania z wykorzystaniem podejścia obiektowego, jego modułów, integracji z użytkownikiem, z systemami zewnętrznymi.

Twórcami notacji UML są G. Booch i I.Jacobson oraz J.Rumbaugh. Notacja powstała w latach 90 i jest rozwijana do dzisiaj – obecna wersja notacji to 2.2. UML jest standardem rozwijaną przez organizację OMG (Object Management Group).



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 3

**Perspektywy implementacyjne**

- przypadków użycia
- logiczna
- dynamiczna
- komponentów
- rozlokowania

przypadków użycia - definiuje funkcjonalność systemu, dynamiczna - prezentuje dynamikę systemu, w jaki sposób zachowuje się system, logiczna - dokumentuje statyczną część systemu komponentów - grupuje powyższe perspektywy w logicznie powiązane komponenty rozlokowania - specyfikuje sprzęt niezbędny do funkcjonowania określonych komponentów

Slajd 4

**Podstawowe rodzaje diagramów UML**

- Perspektywa przypadków użycia  
**Diagram użycia**  
Diagram pakietów
- Perspektywa logiczna  
**Diagram klas**  
Diagram obiektów  
Diagram pakietów
- Perspektywa dynamiczna  
**Diagram czynności**  
Diagram sekwencji  
Diagram pakietów
- Perspektywa rozlokowania  
**Diagram wdrożeniowy**

Aktualnie notacja składa się z około 16 rodzajów diagramów za pomocą których możliwe jest modelowanie statycznych i dynamicznych obszarów budowy i działania systemów. Diagramy pozwalają na zaprojektowanie budowy oprogramowania, baz danych, zaprojektowanie algorytmów, określenie modułów oprogramowania, rozmieszczenie modułów oprogramowania.


Diagramy można podzielić na statyczne - przedstawiające budowę systemu, oraz dynamiczne - przedstawiające np. komunikację pomiędzy elementami statycznymi


Na tej lekcji omówione zostaną podstawowe i najbardziej potrzebne rodzaje diagramów:

Diagram użycia

Diagram klas

Diagram czynności

 **B2E**  
BUSINESS TO EDUCATION

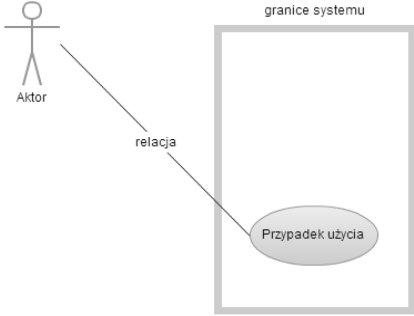
 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY


**PHP**  
Hypertext Preprocessor

## Diagram przypadków użycia

W skład diagramu wchodzi następujące elementy:

- aktor - wykonawca akcji, funkcji
- przypadek użycia - akcja, funkcja
- granice systemu
- relacje (związki)



 **DAILY**  
C R O U P

Lekcja 8 - UML

Diagram przypadków użycia (DPU) pozwala na zaprezentowanie funkcjonalności projektowanego systemu, jego aktorów oraz komunikację między aktorami a przypadkami użycia. Pozwala także na zaprezentowanie relacji pomiędzy aktorami i przypadkami użycia. Diagramy nazywane w języku ang. use case są stosowane do dokumentowania wymagań wobec projektowanego systemu. Projektant ma możliwość dekompozycji wymagania na podwymagania i w procesie iteracyjnego projektowania systemu na stopniowe ich uszczegóławianie. Odpowiednio opisany zestaw przypadków użycia może stanowić dokumentację wymagań która jest podstawą do projektowania i implementacji systemu a finalnie stanowi materiał wejściowy potrzebny przy testach aplikacji.

- Przypadek użycia - konkretna funkcjonalność, akcja którą system może wykonać w interakcji z aktorem. Przypadek użycia powinien mieć zwięzła i jednoznaczną nazwę np. zadzwoń

- Aktor - użytkownik systemu, klient, inny system korzystający z projektowanego przez nas systemu. Aktor określany jest rzeczownikiem. Aktor reprezentuje typ użytkownika systemu, użytkownik może korzystać z wielu przypadków użycia.

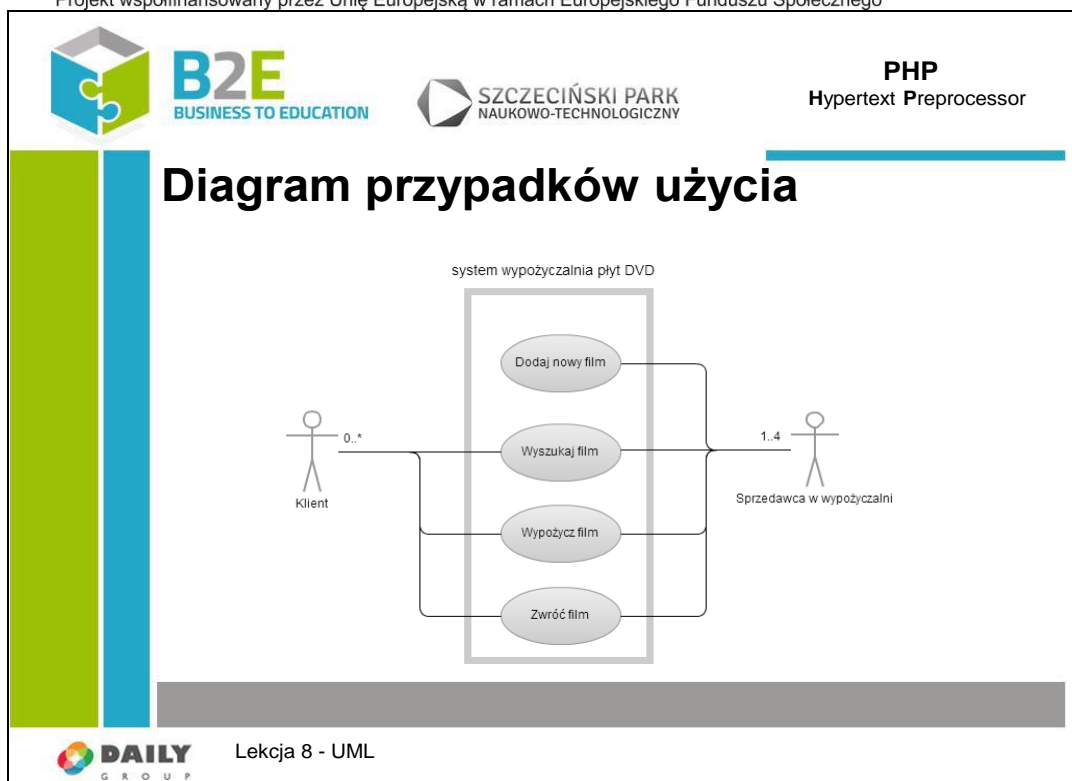
- Relacja - każdy aktor na diagramie przypadków użycia powinien być związany przynajmniej z jednym przypadkiem użycia. Każdy przypadek użycia powinien być użytkowany przez co najmniej jednego aktora.

- Granice systemu - prezentują granice projektowanych systemów (na diagramie mogą wystąpić 2 lub więcej systemów których przypadki użycia należy ze sobą powiązać)

Przykłady: Przycisk dzwoni w telefonie, start w pralce, włącz silnik w samochodzie, wyloguj na facebook-u itp.



Slajd 6




Na rysunku przedstawiono przykładowy diagram przypadków użycia systemu zarządzającego wypożyczalnią płyt DVD (np. osiedlowa wypożyczalnia). Projektant założył bardzo prostą funkcjonalność:

Klient ma możliwość wyszukiwania filmów, wypożyczenia filmu, zwrócenia filmu


Sprzedawca ma także dostęp wyszukiwania filmów ich wypożyczenia i zwracania, ale dodatkowo może dodać nową pozycję w wypożyczalni.

Należy zauważyć że funkcjonalności są bardzo ogólne i w dalszym etapie projektowania należy je uszczegółowić - np. funkcja wypożycz film powinna sprawdzać czy dana płyta jest dostępna (nie jest wypożyczona), przydadzą się szczegóły każdej z funkcji zwłaszcza pola służące np. do wyszukiwania, zakres danych wyświetlanych jako rezultat itp.





**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor


---

## Diagram przypadków użycia

Ćwiczenie:

Przygotuj szczegółowy diagram przypadków użycia dla biblioteki osiedlowej.


Zapewnij kompletność funkcjonalności związanej z obsługą wypożyczeń i zwrotów a także z zarządzaniem zbiorami bibliotecznymi i rezerwacjami.




Lekcja 8 - UML

Zadanie można wykonać na kartkach.

Opcjonalnie można użyć istniejących darmowych narzędzi jak: <http://www.umlet.com/>



**B2E**  
BUSINESS TO EDUCATION

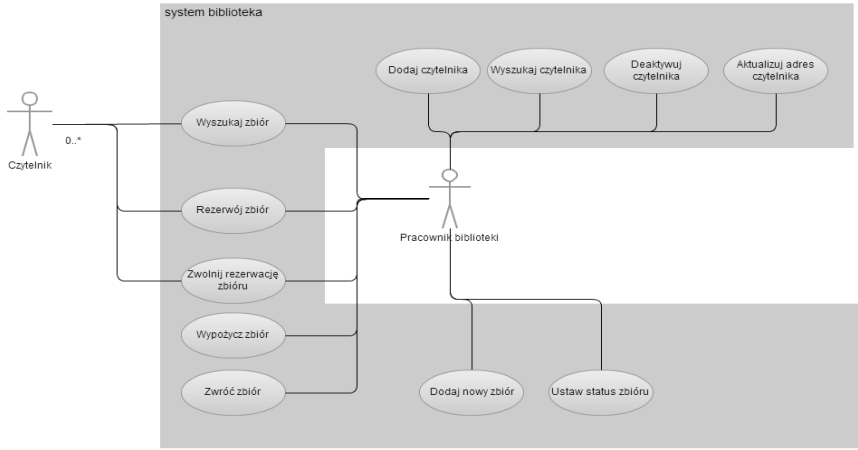


SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Diagram przypadków użycia




```

graph LR
 subgraph system_biblioteka [system biblioteka]
 direction TB
 UC1(Wyszukaj zbiór)
 UC2(Rezerwuj zbiór)
 UC3(Zwolnij rezerwację zbioru)
 UC4(Wypożycz zbiór)
 UC5(Zwroc zbiór)
 UC6(Dodaj czytelnika)
 UC7(Wyszukaj czytelnika)
 UC8(Deaktywuj czytelnika)
 UC9(Aktualizuj adres czytelnika)
 UC10(Dodaj nowy zbiór)
 UC11(Ustaw status zbioru)
 end

 C[Czytelnik] -- "0..*" --- UC1
 C --- UC2
 C --- UC3
 C --- UC4
 C --- UC5

 P[Pracownik biblioteki] --- UC6
 P --- UC7
 P --- UC8
 P --- UC9
 P --- UC10
 P --- UC11

```



Lekcja 8 - UML

Założono obsługę obszarów


Wyszukania i Rezerwacji (dostępne dla czytelnika)

Wypożyczenia i zwrotu – dla pracownika




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego  
Dodania zwrotu i edycji statusu zbioru  
Zarządzania użytkownikiem (dodanie, wyszukanie, deaktywację, aktualizację).

Slajd 9



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Diagram klas


Elementy klasy:

- Nazwa klasy
- Atrybuty - cechy obiektów
- Operacje - czynności wykonywane na atrybutach

Klient
- klient_id : int
- imie : string
- nazwisko : string
- aktywny : boolean
+ ustaw_aktywnosc(_aktywny : boolean);
+ ustaw_imie_nazwisko(_imie : string, _nazwisko: string);

Nosnik
- plyta_id: int
- cena : double
- tytul : string
- wypożyczona : boolean
+ wypożycz(klient_id : int)
+ zwróć()

Przykład



Lekcja 8 - UML

Diagram klas jest obok diagramów przypadków użycia najczęściej stosowanym diagramem w notacji UML.

Diagram klas najlepiej odwzorowuje programowanie obiektowe. Klasa w oprogramowaniu jest definicją obiektów. Obiekt zawiera wycinek modelowanej przez oprogramowanie rzeczywistości, jest czarną skrzynką która posiada atrybuty (dane) oraz metody (funkcje) które działają na atrybutach. Obiekt może udostępniać określone metody na zewnątrz dla innych obiektów - są to metody publiczne (w programowaniu obiektowym zadeklarowane jako public na diagramie oznaczone przez +). Za pomocą udostępnionych metod świat zewnętrzny może korzystać z funkcjonalności obiektu.

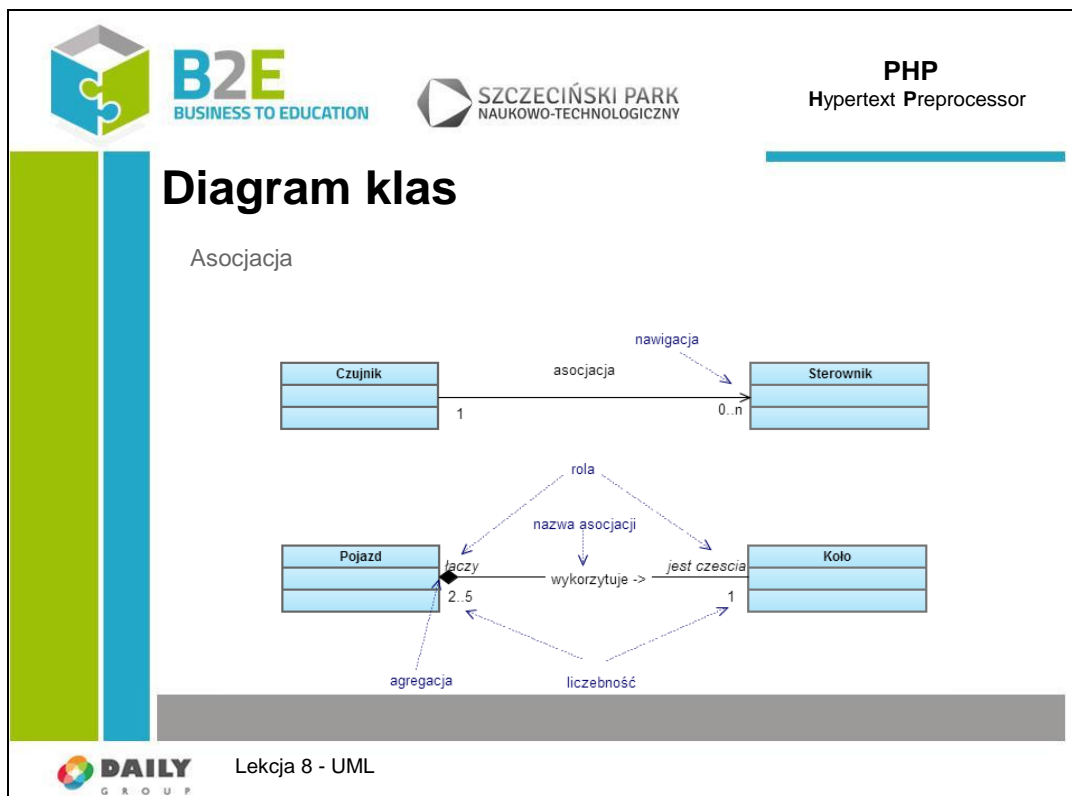
Widoczność atrybutów i metod oznaczana jest symbolami:

"-" - prywatne, niewidoczne na zewnątrz obiektu, dostępne tylko dla kodu obiektu

"+" - publiczne, widoczne na zewnątrz obiektu,

"#" - chronione, prywatne i dodatkowo widoczne dla klas dziedziczących

"~" - pakietowe, widoczne w pakiecie do którego należy klasa



Podobnie jak na diagramie przypadków użycia, diagram klas wykorzystuje związki (relacje) do obrazowania zależności między klasami. Pierwszą relacją jest asocjacja która reprezentuje dowolny związek między obiektami klasy. Asocjacja jest reprezentowana linią ciągłą i może mieć następujące cechy:

nazwę - podkreślająca istotę związku, posiadająca strzałkę prezentującą kierunek interpretacji (np. pojazd wykorzystuje koło),

rolę - definiująca powinność obiektu względem drugiego (np. pojazd łączy koła, koło jest częścią pojazdu),

nawigacja - precyzująca kierunek przepływu komunikatów (np. czujnik przekazuje sygnały do sterownika),

liczebność - definiuje zakres dopuszczalnej liczby obiektów występujących w związku (np. pojazd ma od 2 do 5 kół, koło ma 1 pojazd, czujnik jest podłączony do jednego sterownika, sterownik posiada od zera do n czujników)

agregacja - opisuje związek całość - część. Wyróżnia się


agregację silną (kompozycja) (zapętniony romb) oznaczającą, że po usunięciu agregatu (obiekty grupującego) usuwane są także wszystkie segmenty (obiekty składowe),

agregację częściową (pusty romb) oznaczającą, że po usunięciu agregatu segmenty nadal mogą istnieć.


Wykorzystywanie cech jest opcjonalne i zależy od przyjętego dla danego projektu szczegółowości dokumentacji.



Slajd 11



**B2E**  
BUSINESS TO EDUCATION

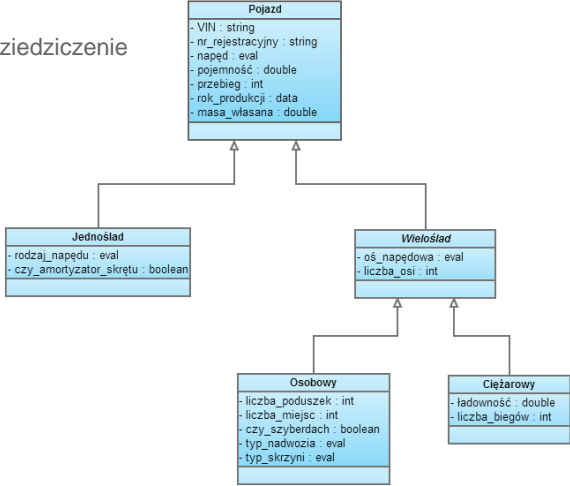



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Diagram klas

Generalizacja - hierarchia, dziedziczenie







Lekcja 8 - UML

Generalizacja jest związkiem reprezentującym dziedziczenie klas. Dziedziczenie skutkuje powstaniem hierarchii klas w której najbardziej ogólna klasa jest umieszczona na samym szczycie hierarchii (korzeń). Na samym dole występują klasy nazywane liśćmi. W hierarchii występują też klasy abstrakcyjne (nazwa pisana kursywą). Klasy abstrakcyjne nie mają swoich instancji obiektów, służą natomiast do zdefiniowania wspólnych atrybutów i metod które będą dziedziczone przez klasy potomne (np. Wieloślad).

Slajd 12



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor


## Diagram klas

Ćwiczenie:

Zaproponuj diagram klas dla systemu zarządzania biblioteką publiczną.

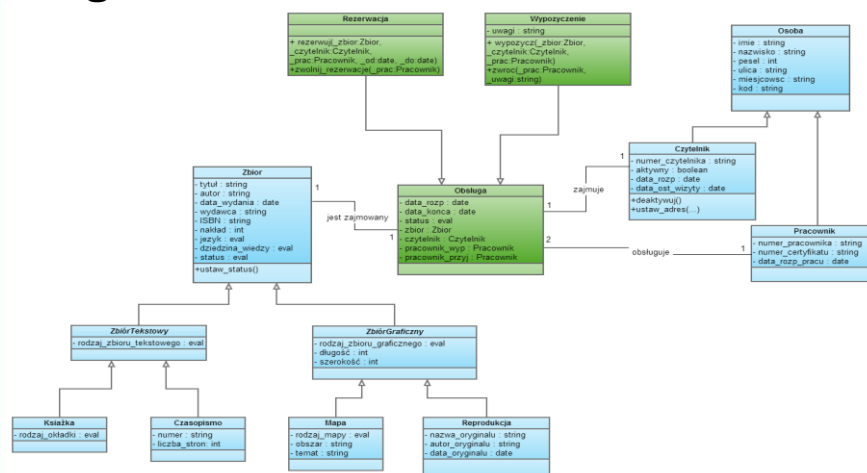
Przyjmij przynajmniej 3 poziomą hierarchię opisującą zbiory biblioteczne i min 2 poziomą hierarchię klas opisującą użytkowników i pracowników biblioteki.

Wykorzystaj wcześniej przygotowany diagram przypadków użycia w celu weryfikacji kompletności funkcjonalności oprogramowania.



Lekcja 8 - UML


## Diagram klas




W przykładowym rozwiązaniu zamodelowano tylko część struktury zbiorów bibliotecznych (tekstowe i graficzne – klasy abstrakcyjne). Rozróżniono osoby jako czytelników i pracowników biblioteki. Centralną klasą wiążącą jest klasa Obsługa która jest powiązana ze zbiorem który jest wypożyczany (założono że na 1 wypożyczenie 1 zbiór) z czytelnikiem który zbiór wypożycza/rezerwuje z pracownikiem który zbiór przekazuje i przyjmuje/dokonuje rezerwacji. Potomkowie klasy Obsługa (Wypożyczenie, Rezerwacja) implementują metody przedstawiono metody wypożycz i zwróć dla klasy albo rezerwuj lub zwolnij Wypożyczenie oraz ustaw\_status dla Zbiór i deaktywuj i ustaw\_adres dla czytelnika.



Slajd 14



**B2E**  
BUSINESS TO EDUCATION



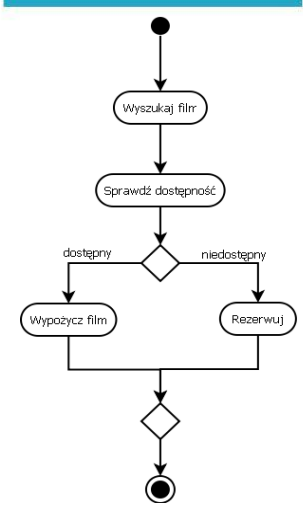
SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Diagram czynności

Diagramy czynności (aktywności)


- czynności
- akcje
- przepływ
- start, koniec



```

graph TD
 Start(()) --> Wyszukaj[Wyszukaj film]
 Wyszukaj --> Sprawdź[Sprawdź dostępność]
 Sprawdź --> Dec1{ }
 Dec1 -- dostępny --> Wypożycz[Wypożycz film]
 Dec1 -- niedostępny --> Rezerwuj[Rezerwuj]
 Wypożycz --> Dec2{ }
 Rezerwuj --> Dec2
 Dec2 --> End((()))


```




Lekcja 8 - UML

Diagram czynności służy do zobrazowania algorytmów, procesów biznesowych w projektowanym oprogramowaniu lub scenariuszy przypadków użycia. Składają się z następujących elementów:

- czynności - reprezentująca czynność złożoną, możliwą do dalszej dekompozycji na podczynności i akcje i uszczegółowienia, (np. wyszukaj film),
- akcji - elementarna jednostka zachowania, nie podlega dalszej dekompozycji, reprezentuje algorytm, metodę,
- przeptywów sterowania - strzałki łączące czynności, akcje i punkty startu i zakończenia,
- punkt startu - czarne kółko,
- punkt końca - okrąg z czarną kropką w środku



**B2E**  
BUSINESS TO EDUCATION

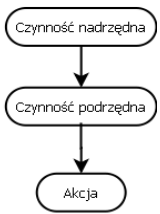


SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Diagram czynności

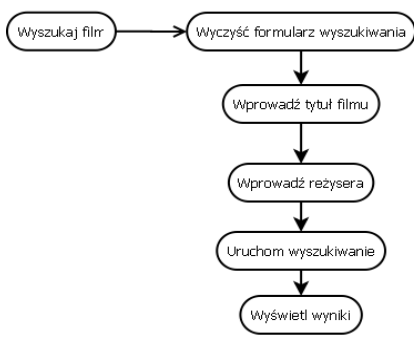
Dekompozycja czynności



```

graph TD
 A(Czynność nadrzędna) --> B(Czynność podrzędna)
 B --> C(Akcja)


```



```


graph TD
 A(Wyszukaj film) --> B(Wyczyść formularz wyszukiwania)
 B --> C(Wprowadź tytuł filmu)
 C --> D(Wprowadź reżysera)
 D --> E(Uruchom wyszukiwanie)
 E --> F(Wyświetl wyniki)

```




Lekcja 8 - UML

Czynności mogą być dekomponowane. Czynność nadrzędna po uszczegółowieniu może być procesem czynności podrzędnych albo sekwencją akcji. Przykładowo, czynność "Wyszukaj film" może składać się z sekwencji czynności.



**B2E**  
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

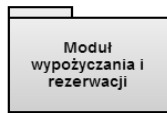
**PHP**  
Hypertext Preprocessor


## Diagram pakietów

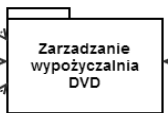
Diagram pakietów


- pakiet
- zależność
- zagnieżdżanie


















Lekcja 8 - UML

Diagram składa się z następujących elementów:  
 pakiet - element grupujący inne diagramy, (ramka otaczająca elementy innych diagramów), praktyka zaleca objęcie pakietem diagramów




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego  
zależność - łączy pakiety w logiczną strukturę, linia przerywana której kierunek wykorzystania pakietu,  
zagnieżdżanie - grupowanie innych pakietów.

Slajd 17



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

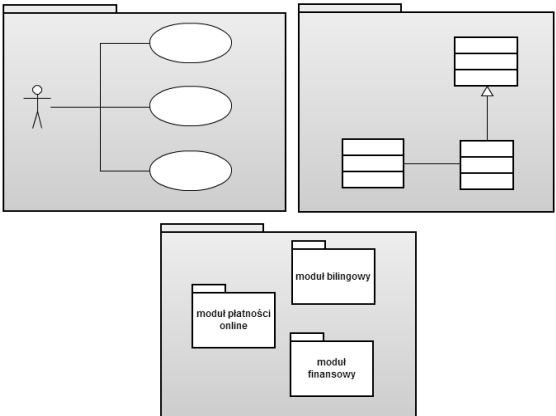
**PHP**  
Hypertext Preprocessor


---

## Diagram pakietów

Diagram pakietów

- zagnieżdżanie





Lekcja 8 - UML

Diagram pakietów pozwala na porządkowanie i grupowanie dokumentacji. Wszystkie diagramy mogą być organizowane w pakiety (np. pakiet grupujący przypadki użycia lub diagram klas). Pakiety mogą także grupować inne pakiety (np. pakiet dolny).





## Diagram pakietów

Ćwiczenie:

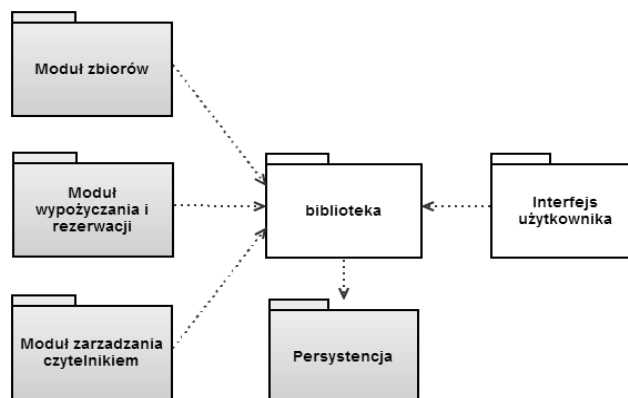
Zaproponuj podział oprogramowania do zarządzania biblioteką na pakiety.

Postaraj się pogrupować tematycznie klasy, przypadki użycia i diagramy czynności.

Następnie dokonaj zagnieżdżenia powstałych pakietów w pakiety nadrzędne - grupujące moduły między sobą np. zarządzanie czytelnikami, zarządzanie zbiorami bibliotecznymi.



## Diagram pakietów - rozwiązanie



Założono rozdzielanie modułów:

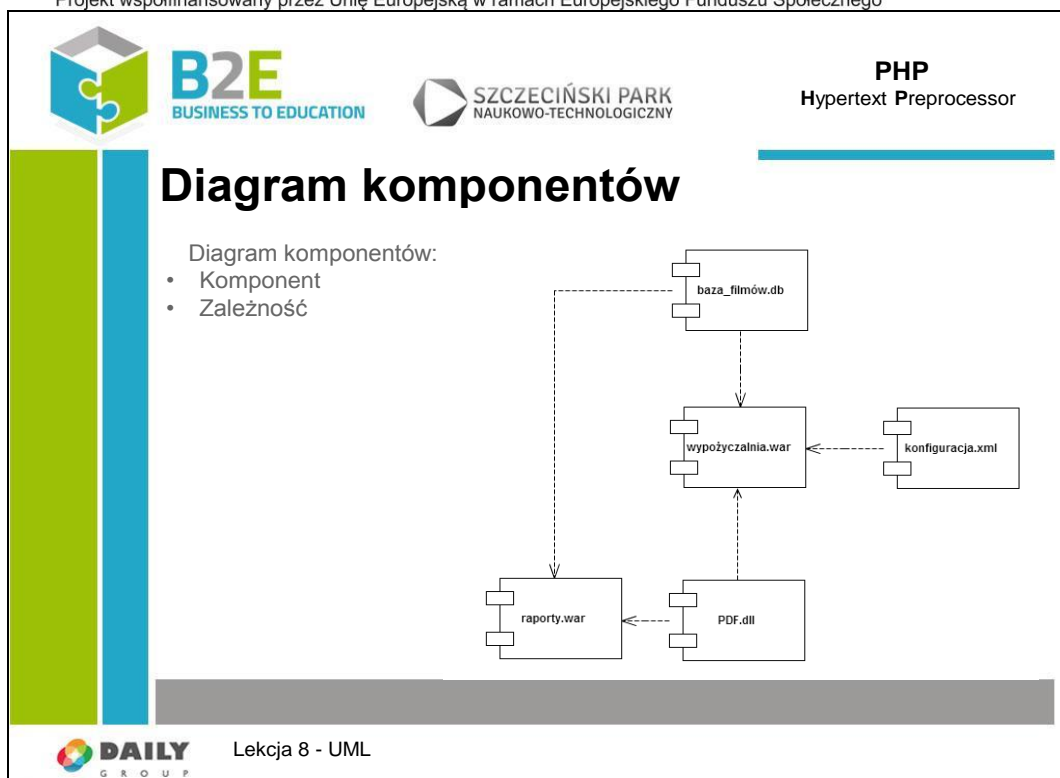
Zarządzania zbiorami

Wypożyczenia i rezerwacji

Zarządzania czytelnikiem

Zaprojektowano, że stan systemu będzie zapisywany w bazie danych (persystencja)

Założono prosty interfejs użytkownika (GUI) dla pracowników i czytelników

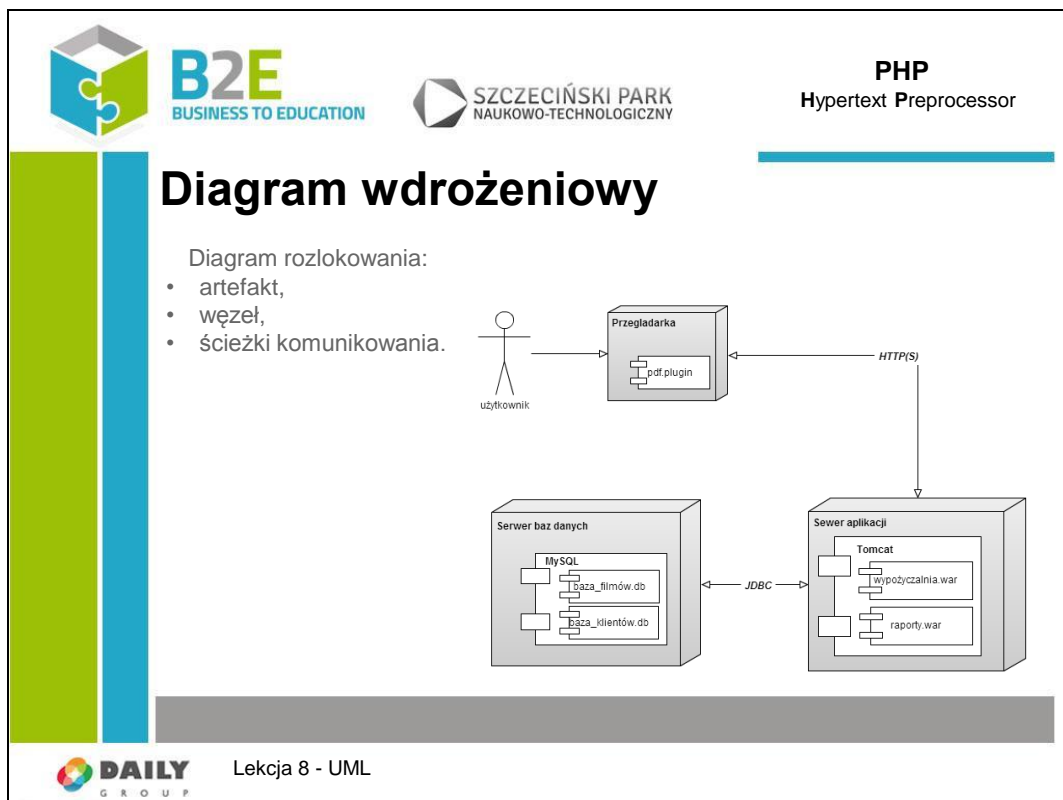


Lekcja 8 - UML

Komponentem jest izolowany, wymienny moduł oprogramowania stanowiący część systemu. Komponentami może być każdy element składowy systemu przykładowo: bazy danych (np. plik bazy danych, system zarządzania bazami danych), pliki bibliotek programowych (np. moduł do tworzenia plików PDF), moduły oprogramowania (np. wypożyczalnia, raporty), pliki (np. konfiguracyjne, graficzne).

Komponent jest reprezentowany przez symbol prostokąta z dwoma bolcami (symbolizuje interfejsu). W nowszej notacji UML dodano dodatkowe ikony komponentów pozwalających na definiowanie i nazywanie interfejsów.

Zależności komponentów są reprezentowane przez linię przerywaną z grotem. Kierunek wskazuje na komponent wykorzystujący.



W dokumentacji często występuje potrzeba określenia fizycznego rozmieszczenia artefaktów (sztucznie wytworzonych elementów, np. komponentów) oprogramowania. Artefaktem mogą być bazy danych (np. baza\_filmów.db), klasy, diagramy, biblioteki, komponenty (np. wypożyczalnia.war)

Węzłem jest fizyczny lub logiczny zasób przetwarzający na którym zainstalowano i skonfigurowano do działania artefakty (np. serwer baz danych, przeglądarka).

Ścieżki komunikowania reprezentują przepływ i kierunek przepływu informacji. Nazwa ścieżki może sugerować protokół komunikacyjny który projektuje się wykorzystać.



Slajd 22

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

## Diagram wdrożeniowy

Ćwiczenie:

Zaproponuj diagram wdrożeniowy dla oprogramowania zarządzającego biblioteką.

Postaraj się rozdzielić funkcje serwera aplikacji od serwera baz danych, umieść w węzłach odpowiednie komponenty.

Lekcja 8 - UML

Slajd 23

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

---

```

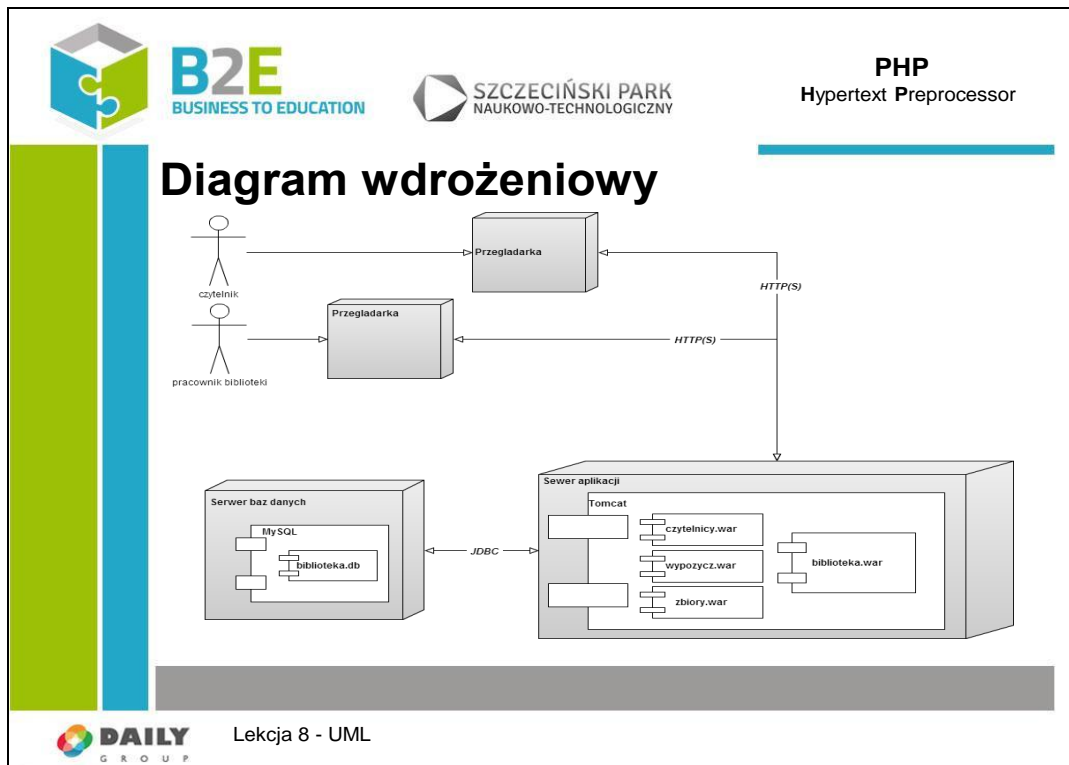
graph TD
 db[biblioteka.db] -.-> czytelnicy[czytelnicy.war]
 db -.-> wypożycz[wypożycz.war]
 db -.-> zbiory[zbiory.war]
 czytelnicy -.-> biblioteka[biblioteka.war]
 wypożycz -.-> biblioteka
 zbiory -.-> biblioteka
 konfiguracja[konfiguracja.xml] -.-> biblioteka

```

Lekcja 8 - UML

Założono realizację wspólnej bazy danych i rozdzielanie modułów funkcjonalnych wg pakietów

Całość zintegrowana interfejsem użytkownika biblioteka.war oraz moduł konfiguracji technicznej.



Założono instalację rozwiązania na dwóch serwerach – serwer baz danych oraz serwer aplikacyjny

Przyjęto zastosowanie darmowych rozwiązań MySQL i implementacji baz danych

Przyjęto zastosowanie serwera aplikacyjnego tomcat i uruchamianie rozwiązania z poziomu tego serwera

Założono że dostęp do systemu odbywa się wyłącznie przez przeglądarkę.



Slajd 25

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Ćwiczenia

1. Co to jest UML? Co oznacza skrót? Do czego służy?
2. Wymień min 5 diagramów UML. Jakie znasz perspektywy projektowania oprogramowania?
3. Jakim diagramem zobrazujesz wymagania na system informatyczny? Narysuj przykład diagramu i omów go.
4. Jakim diagramem zobrazujesz model obiektowy? Narysuj przykład diagramu prezentującego dziedziczenie i omów jego elementy.
5. Jakim diagramem zaprezentujesz algorytm lub proces który należy zaimplementować w systemie? Narysuj przykład diagramu i omów go.
6. Jakim diagramem zorganizujesz i pogrupujesz inne diagramy?
7. Jaki diagram służy do przedstawienia fizycznego rozmieszczenia elementów oprogramowania?
8. Do czego służy diagram komponentów?

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**DAILY**  
GROUP

Lekcja 8 - UML

## Ćwiczenia

Ćwiczenia dla tej jednostki lekcyjnej zostały umieszczone na ostatnim slajdzie.

## Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili samodzielnie zaprojektować aplikację w języku UML oraz stworzyć podstawowe diagramy aplikacji.

## Lekcja 10 - Dobre praktyki programowania w PHP

### Cel lekcji

Celem lekcji jest wprowadzenie ucznia w świat praktycznego podejścia do aplikacji oraz wytyczenie im dalszej drogi na rozwój w przyszłości. Pokazanie im dobrych praktyk oraz wielu przydatnych fragmentów kodu, szkieletów oraz narzędzi, które przydatne są każdemu programiście. Zaprezentowano również wzorce projektowe i obecnie najwyższy abstrakcyjnie paradygmat programowania, zwany programowaniem zorientowanym na aspekty.

## Treść - slajdy z opisem

Slajd 1

Slide 1 features a title slide with the following content:

- Logos: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, and PHP: Hypertext Preprocessor.
- Vertical bars: A green bar on the left and a blue bar on the right.
- Text: "PHP: Hypertext Preprocessor - poziom średnio zaawansowany", "Lekcja 10", and "Dobre praktyki programowania w PHP".
- Logos: "KAPITAŁ LUDZKI" (Human Capital) and "WALIOŚNOŚĆ EUROPEJSKIEGO FUNDUSZA SPOŁECZNEGO" (European Union Regional Development Fund).
- Text: "Celem: najlepsza inwestycja" and "Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego".
- Logo: DAILY GROUP at the bottom left.

Słowa wstępu

Slajd 2

Slide 2 features a content slide with the following content:

- Logos: B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, and PHP: Hypertext Preprocessor.
- Vertical bars: A green bar on the left and a blue bar on the right.
- Text: "Dobre praktyki programowania" and ". Wprowadzenie".
- Logo: DAILY GROUP at the bottom left.
- Text: "PHP: Hypertext Preprocessor - poziom średnio zaawansowany" at the bottom right.

PHP jest językiem, w którym łatwo rozwiązuje się większość problemów. Jego prostota niesie jednak ze sobą poważną konsekwencję, kod który zaimplementowaliśmy, a który wymaga zmian, jest niezrozumiały i wymaga przepisania od nowa, aby zawrzeć odpowiednie zmiany. Często też kod, który rozwiązuje podobny problem w innym projekcie, nie nadaje się do zastosowania. Należy zatem zadać sobie pytanie, czy powinniśmy po raz kolejny rozwiązywać ten sam problem, bądź też na nowo odkrywać koło, czyli pisać rzeczy, które mam w najlepszy i sprawdzony sposób zaimplementowane przez innych programistów.




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 3



**B2E**  
BUSINESS TO EDUCATION







SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Rozszerzenia PHP

- PEAR
- PECL
- Composer



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

PEAR jest to system dystrybucji open-source'owych rozszerzeń języka PHP. Powstał od roku 1999 i posiada wielką ilość zasobów, które ułatwiają nam pracę z językiem PHP. PECL to pierwotnie część PEAR, w przeciwieństwie do swojego rodzica, jest to repozytorium społeczności, a pakiety w nim zawarte, są pakietami nieoficjalnymi. Zanim napiszesz coś co już zostało napisane, zajrzyj do repozytorium, gdyż może to Ci pomóc zaoszczędzić mnóstwo czasu i dostarczyć źródło ze sprawdzonym rozwiązaniem.

Używając wielu frameworków oraz dodatków, do zarządzania zależnościami w projekcie przydatny może okazać się Composer (<https://getcomposer.org/>).

Composer jest genialnym systemem wspomagającym zarządzanie zależnościami aplikacji w PHP. Wystarczy jeżeli wylistujesz swoje zależności w pliku composer.json, a narzędzie to, przy użyciu kilku prostych poleceń, automatycznie ściągnie odpowiednie wersje bibliotek i ich zależności oraz odpowiednio skonfiguruje autoloadera. Za pomocą Composera można zainstalować wiele znanych bibliotek. Ich listę znajdziesz na stronie projektu Packagist - oficjalnego repozytorium bibliotek kompatybilnych z tym systemem.






PHP: Hypertext Preprocessor

---

## Frameworki

- Wprowadzenie
- ZF 
- Symfony 
- Slim



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Implementując aplikację internetową wykonujemy szereg operacji, które związane są z jej szkieletem oraz często powtarzalnymi rzeczami, jak autoryzacja czy systemy formularzy. Z pomocą przychodzą nam gotowe szkielety aplikacji takie jak Zend Framework czy Symfony. Jeżeli piszemy aplikację, która jest większa niż formularz kontaktowy, warto użyć gotowych rozwiązań. Szkielety te posiadają architekturę MVC oraz całą masę gotowych bibliotek np. pośredniczących w komunikacji z bazą danych, związanych z cache'owaniem, czy chociażby obsługą plików pdf. Wersje 2 pierwszych frameworków, są zaimplementowane w pełni obiektowo i wykorzystują wiele wzorców poprawnego programowania, z których można czerpać świetne lekcje. Na ich stronach znajdziemy mnóstwo materiałów związanych z ich użyciem, video-tutoriale oraz wyczerpującą dokumentację np. ZF około 1300 stron formatu A4. Oprócz wyżej wymienionych frameworków, jest jeszcze wiele innych jak np. Yii, cakePHP, codeigniter czy fuel. Warto poświęcić trochę czasu na eksplorację internetu w poszukiwaniu tych zasobów, ponieważ można zaoszczędzić masę pracy oraz otrzymać sporo profesjonalnie napisanego kodu.

Slim (<http://www.slimframework.com/>) jest natomiast przykładem frameworka, przy pomocy którego w szybki i prosty sposób będziemy mogli tworzyć własne API (np REST API dla aplikacji na Androidzie – Lekcja 11).



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 5

Jeżeli nie czujesz się jeszcze na siłach, aby korzystać z wyrafinowanych narzędzi, ponieważ ich ogrom Cię przeraża lub piszesz tylko małe aplikacje dla których szkoda tracić zasoby i wydajność, możesz korzystać z frameworków jako bibliotek lub ze strictly bibliotek, których jest wiele np. TCPDF do obsługi plików pdf, czy biblioteki SMARTY do obsługi szablonów. Użytecznym narzędziem okazuje się również Composer, czyli program do zarządzania pakietami PHP.

Slajd 6

PHP w wersji 5, zapewnia pełną obsługę programowania obiektowego, czyli najpopularniejszego obecnie programowania. Kontynuacją drogi ku wydzieleniu abstrakcji w językach programowania jest programowanie aspektowe. Implementuje się je na razie w sposób obiektowy, gdyż PHP nie zapewnia jeszcze wbudowanej obsługi aspektów. Polega ono na wydzieleniu abstrakcyjnych czynności na poziomie rozumowania człowieka np. system logów, i uniezależnienie ich od klas. Grupowanie takie zapewnia nam lepsze

możliwości projektowania i bardziej zrozumiałą implementację. Zobrazujmy nasz przykład, założmy że nasz zbiór klas przy wykonywaniu jakiś czynności, zapisuje te akcje w tzw. Logach. Czy nie łatwiej by było wyodrębnić ten mechanizm z klas i przenieść go do poziomu kontenera naszej aplikacji? Zyskujemy na tym wiele czasu, nie musimy implementować go za każdym razem. Poza tym wszelkie zmiany dokonujemy tylko raz. Co w przyszłości i przy częstych zmianach w aplikacji staje się uciążliwe.

Slajd 7

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Wzorce projektowe

- Wprowadzenie
- Do czego służą?

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Kolejną ważną rzeczą, która ułatwia życie programiście są wzorce projektowe. Są to rozwiązania problemów projektowych, które stosowane w odpowiednich przypadkach, potrafią zaoszczędzić nam wiele czasu i stworzyć łatwe do dalszego rozwijania środowisko, czy też uniknąć wielu błędów.

Slajd 8

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Wzorce projektowe ćwiczenie

Analiza przykładu wzorca projektowego singleton lub MVC.

Singleton implementation

Conventional implementation

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Celem wzorca projektowego singleton jest wprowadzenie ograniczenia możliwości tworzenia obiektów danej klasy do tylko jednej instancji. Tak stworzony obiekt dostępny jest globalnie dla całej aplikacji.

Wzorca singleton powinniśmy używać do implementacji:



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

- obiektów, przechowujących informacje o stanie aplikacji (np konfiguracji).
- obiektów które powinny istnieć przez cały czas trwania aplikacji (np logowanie zdarzeń z wielu miejsc w kodzie do jednego pliku Log)

Przykład znajduje się w materiałach ze skryptami PHP.

Zmiany w stosunku do skryptu nie używającego wzorca singleton:

- konstruktor został zmieniony z publicznego na prywatny (to chroni przed tworzeniem obiektu bez wywołania metody getInstance)
- zmienna \$dbh zmieniona z publicznej na prywatną (chroni to przed nadpisaniem zmiennej \$dbh )
- tablica db\_info zamieniona z publicznej na prywatną (z tego samego powodu co \$dbh)

Grafika pobrana ze strony: <http://yavkata.co.uk/weblog/php/php-singleton-design-pattern/>

Kolejnym bardzo popularnym wzorcem projektowym jest MVC (Model-Widok-Kontroler). Używając tego modelu, aplikacje zawierające interfejs użytkownika są zorganizowane w 3 struktury:

- Model – zawiera logikę aplikacji
- Widok – odpowiedzialny jest za wyświetlenie rezultatów modelu w interfejsie użytkownika
- Kontroler - przyjmuje dane wejściowe od użytkownika i na ich podstawie dostarcza oraz pobiera dane z/do modelu oraz odświeża widok

Slajd 9

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor



Dobre praktyki samego programowania

- <http://goo.gl/suWCFu>

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jeśli chodzi o samo pisanie w języku PHP, w internecie możemy znaleźć wiele stron poświęconych dobrym praktykom programowania. Warto polecić „php the right way”, gdyż jest ona podstawą wielu publikacji na temat dobrych praktyk w języku PHP.


PHP: Hypertext Preprocessor

## Dokumentowanie kodu

- Wprowadzenie
- phpDocumentor

```

/**
 * Displays a new message box, or reinitializes an existing message box, based on the config options passed in.
 *
 * @param config the message box config
 */
public static native void show(MessageBoxConfig config) /*-{
 $wnd.Ext.MessageBox.show(config.$com.gwtext.client.core.JsObject::jsObj);
}-*/;
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jeżeli chcemy używać naszego kodu wielokrotnie, a nie pracujemy samemu, bądź też chcemy udostępniać nasz kod dla innych programistów. Warto zadbać o dobre udokumentowanie kodu. Ułatwia ono i przyspiesza pracę nie tylko innym, ale też nam samym, kiedy po dłuższym okresie sięgamy po nasz wcześniej napisany kod. Polecanym tutaj narzędziem jest phpDocumentator, który tworzy dokumentację automatycznie na podstawie specjalnych komentarzy zamieszczanych w skrypcie.

PhpDocumentor jest narzędziem, dzięki któremu możliwe jest generowanie dokumentacji z kodu źródłowego PHP. Dzięki takiej dokumentacji, można dostarczyć odbiorcy więcej informacji związanych z funkcjonalnością osadzoną w kodzie źródłowym, a nie tylko to, co jest wykorzystywane w interfejsie użytkownika.

Dokumentacja taka jest bardzo przydatna w następujących sytuacjach:

- Gdy tworzymy zestaw aplikacji udostępniające API
- Gdy tworzymy aplikacje opartej na architekturze pluginowej, umożliwiającej tworzenie własnych dodatków

Więcej informacji dostępnych na stronie: <http://www.phpdoc.org/>



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 11

**BZE**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Kontrola wersji

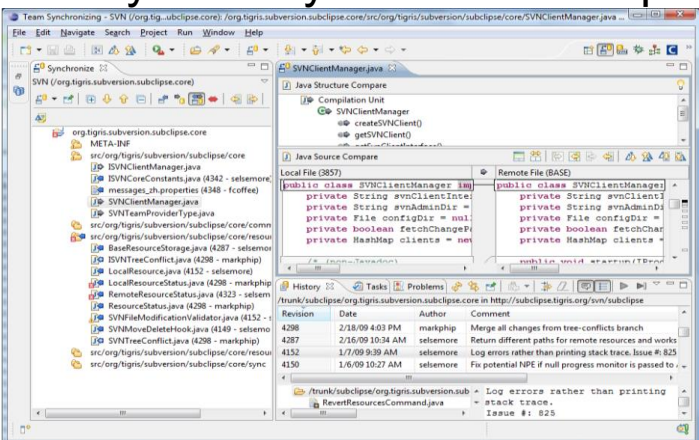
- Co to jest i do czego służy?
- SVN
- GIT

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Systemy kontroli wersji lub w skrócie systemy wersjonowania, są to programy, które przechowują historię naszej pracy i ułatwiają pracę w większej grupie programistów. Do najbardziej popularnych należą subversion (SVN) oraz git. Po napisaniu danego fragmentu aplikacji, możemy swobodnie wysłać ją na serwer i dołączyć do całości większej aplikacji, możemy także pobierać fragmenty napisane w przeszłości, czy przeglądać zmiany zrobione przez innych w naszym kodzie. Systemy te pozwalają także na tworzenie repozytoriów na serwerach i udostępnianie najnowszych wersji naszych bibliotek, czy też aplikacji innym.

Do komunikacji z serwerami zawierającymi usługi wersjonowania często używamy pluginów (np do ulubionego IDE) lub samodzielnych aplikacji jak np. TortoiseSVN.



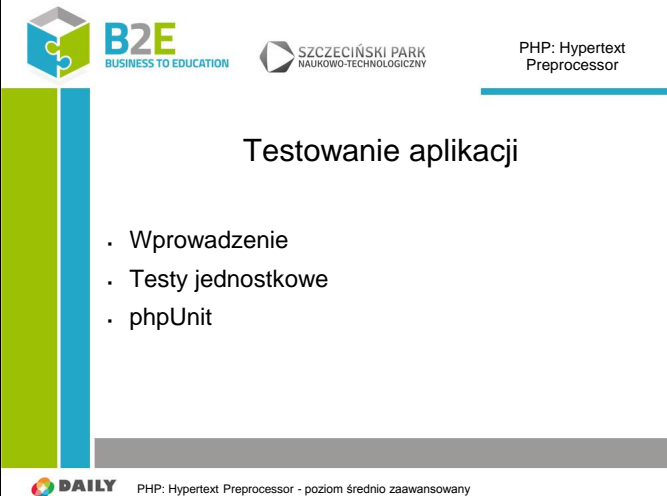
The screenshot shows the Eclipse IDE interface. On the left, the 'Synchronize' view displays a tree of SVN files for the project 'org.tigris.subversion.subclipse.core'. The 'SVNClientManager.java' file is selected. The main editor shows the code for 'SVNClientManager.java' with a 'Java Source Compare' window open, comparing the local file (3857) with the remote file (BASE). The code includes private fields like 'svnClientList', 'svnAdminDir', 'configDir', 'testChangeF', and 'clients', and methods 'createSVNClient()' and 'getSVNClient()'. Below the code, a 'History' view shows recent revisions with their dates, authors, and comments.

PHP: Hypertext Preprocessor

**DAILY GROUP** PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Na przykładzie widzimy zmiany przychodzące i wychodzące. Zmiany wychodzące to praca wykonana przez nas. Zmiany przychodzące to zmiany wykonane przez innych deweloperów. Z prawej strony widzimy konkretne zmiany dokonane na wybranym pliku.

(screen pobrany ze strony: <http://www.java-forums.org/blogs/eclipse/975-working-svn-eclipse.html>).



The slide features the B2E logo and the text 'SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY'. The main title is 'Testowanie aplikacji'. Below it, a bulleted list contains: 'Wprowadzenie', 'Testy jednostkowe', and 'phpUnit'. At the bottom, the 'DAILY GROUP' logo and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany' are visible.

PHP: Hypertext Preprocessor

**DAILY GROUP** PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Testowanie aplikacji to jedna z ważniejszych części programowania, mamy różne rodzaje testów np. integracyjne, wydajnościowe. Jeśli chodzi jednak o najczęściej przeprowadzane testy przez samych programistów, są to testy jednostkowe. Testy jednostkowe to testy poszczególnych metod, procedur, bądź obiektów. Z pomocą przy testowaniu przychodzi nam biblioteka automatyzująca testy jednostkowe phpUnit. Warto z niej korzystać, ponieważ zaoszczędzimy sporo pracy.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 14

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Automatyzacja instalacji

- Wprowadzenie
- Phing

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

PHP jest językiem skryptowym, stąd też nie posiada instalatorów znanych nam z systemów operacyjnych w formie pliku wykonywalnego. Są jednak narzędzia np. Phing, które pomogą nam stworzyć instalator aplikacji. Phing jest narzędziem niezależnym od platformy, wzorowanym na java-owym ANT-cie. Reguły budowania określa się w pliku xml. Po zakończonym procesie możemy zainstalować naszą aplikację, korzystając przy tym np. z rozszerzenia PEAR.

Slajd 15

**B2E**  
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext  
Preprocessor

## Gotowe systemy

- CMS
- Fora
- Blogi
- Sklepy
- Platformy e-learningowe

**DAILY**  
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Oprócz wcześniej wspomnianych mniejszych elementów jak biblioteki, przy tworzeniu nowej aplikacji, warto się zastanowić, czy jako podstawę do tworzenia, nie wykorzystać jakiegoś gotowego systemu, który jest udostępniany na otwartej licencji i rozwijany od wielu lat. Nasza część implementacji w tym momencie opiera się wtedy tylko na dopisaniu odpowiednich wtyczek, komponentów, bloków lub modułów, w zależności od wybranego systemu. Wśród systemów zarządzania treścią mamy ogromny wybór, polecić tu jednak można system Joomla, czy Drupal, które należą do najbardziej rozbudowanych. Jeśli chodzi



o fora internetowe napisane w języku PHP, szczególną popularnością cieszy się phpBB, z dużym wyborem darmowych szat graficznych i dodatków. Wordpress jest bezkonkurencyjnym systemem, jeśli chodzi o blogi internetowe, choć swój rozwój kieruje w stronę systemu CMS, pewnie z uwagi na prostotę i wygodę zarządzania tym systemem. Jeśli chodzi o darmowe sklepy wykonane w technologii PHP, z racji ogromnej ilości pluginów, warto polecić prestashop. Jeśli chodzi o platformy e-learningowe nie ma za dużego wyboru, zgodność ze standardami i wysoki poziom funkcjonalności, zapewnia tylko system Moodle.

Slajd 16

Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Instalacja CMS – Joomla!

- Zainstaluj system zarządzania treścią Joomla!
- Dokonaj zmian poprzez panel administracyjny.
- Zainstaluj dodatek.
- Znajdź, pobierz i zainstaluj darmowy szablon zmieniający wygląd strony.

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Dla uczniów zaawansowanych można zaproponować napisanie własnego modułu/komponentu instalowanego w Joomla!

Slajd 17

Logo B2E BUSINESS TO EDUCATION, SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY, PHP: Hypertext Preprocessor

### Ćwiczenie praktyczne

- Napisz własny moduł dla CMSa Joomla! Moduł powinien być instalowalny poprzez panel administracyjny Joomla! Moduł może wyświetlać dowolny komunikat, celem zadania jest poznanie struktury modułów.

DAILY GROUP PHP: Hypertext Preprocessor - poziom średnio zaawansowany



mod\_firstmodule/\*.\*



Slajd 18

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego


singleton2.php

PHP: Hypertext Preprocessor

### Ćwiczenie praktyczne

- Uruchom i przeanalizuj przykład zawarty w pliku singleton2.php. Przykład ten zawiera implementację książki, która jest singletonem, oraz 2 użytkowników, którzy mogą wykonać na książce pewne operacje. Co stanie się gdy użytkownik 2 spróbuje wypożyczyć książkę, która była wypożyczona wcześniej przez użytkownika 2?



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slajd 19




PHP: Hypertext Preprocessor

### Dziękujemy za uwagę

Człowiek - najlepsza inwestycja




Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

## Ćwiczenia

Ćwiczenie 1 Zapoznać się z dokumentacją wybranego przez siebie, nowego (nieznanego) narzędzia, frameworku lub biblioteki napisanej w języku php.

Ćwiczenie 2 Wykonać jeden lub więcej podstawowych przykładów (tutoriali) dostępnych na stronie wybranej technologii lub w jej dokumentacji.

## Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą mieli szerszy pogląd na temat praktycznego wykorzystania języka PHP w przyszłych projektach. Będą wiedzieli, czym są wzorce projektowe, jakie narzędzia i biblioteki warto poznać oraz jakie gotowe systemy proponuje im społeczność programistów języka PHP. Lekcja ma też na celu zachęcić ich do obrania własnej ścieżki rozwoju i dalszej pracy z językiem PHP.

## Lekcja 11 - Projekt Interdyscyplinarny - Książka teleadresowa

### Cel lekcji

Celem lekcji jest zaprojektowanie strony WWW oraz w aplikacji mobilnej z funkcjonalnością książki teleadresowej kolegów i koleżanek ze szkoły lub klasy. W tej lekcji uczniowie dowiedzą się jak praktycznie wykorzystać wiedzę zdobytą w tym module na potrzeby pierwszego projektu informatycznego składającego się z elementów wykorzystujących cztery języki programowania: SQL, PHP, JavaScript i JAVA.

### Treść - slajdy z opisem

Slajd 1

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP Hypertext Preprocessor

# PHP: Hypertext Preprocessor

Lekcja 11:  
Interdyscyplinarny projekt:  
Książka teleadresowa kolegów/koleżanek ze szkoły

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPONSORUJĄCA

UNIA EUROPEJSKA FUNDUSZ SPÓŁCZYNNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**DAILY GROUP** PHP Hypertext preprocessor Lekcja 11

Proces powstawania książki teleadresowej kolegów/koleżanek ze szkoły. W module SQL pokażemy jak zaprojektować bazę danych, na PHP i JavaScript, jak stworzyć stronę WWW do dodawania kolegów i koleżanek z klasy do takiej książki, a w module JAVA pokażemy, jak zaprogramować prostą aplikację na telefonach z systemem operacyjnym Android do przeszukiwania danych o uczniach z różnych klas.

Slajd 2

**B2E** BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP Hypertext Preprocessor

## Założenia

- Lista klas w szkole
- Lista uczniów w klasie
- REST API dla aplikacji na Androida
- Szczegółowe informacje na temat ucznia:
  - Imię
  - Nazwisko
  - Adres e-mail
  - Numer telefonu
  - Zdjęcie

**DAILY GROUP** Lekcja 11 – Interdyscyplinarny projekt

Założenia projektu.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 3

**Struktura projektu**

- Template (index.php)
- Content – odpowiednie podstrony projektu (home.php, klasy.php, uczniowie.php)
- Obsługa połączenia do bazy danych (db.class.php)
- Klasa do wykonywania operacji na bazie danych - odczyt, zapis, usuwanie (data.class.php)

DAILY GROUP Lekcja 11 – Interdyscyplinarny projekt

Struktura projektu.

Slajd 4

**Alfabetyczna struktura katalogów projektu**


- API – REST API dla aplikacji na Androida
- Content – kolejne podstrony aplikacji
- Css, images – style oraz grafika
- Js – skrypty Java Script
- Lang – pliki językowe
- Libs – zewnętrzne biblioteki
- Upload – folder ze zdjęciami wysłanymi z aplikacji
- Utils – klasy, funkcje oraz konfiguracja


DAILY GROUP Lekcja 11 – Interdyscyplinarny projekt

Struktura projektu.API – implementacja REST API dla androida, wykorzystuje SLIM framework

Wielojęzykowość aktualnie nie jest uruchomiona, ale stanowić może załączek dla dalszego rozwoju aplikacji i dodatkowych zadań dla ucznia. Plik index.php zawiera zakomentowane fragmenty kodu odpowiedzialne za sterowanie wielojęzykowością – wyświetlanie odpowiednich flag, załączanie odpowiednich plików językowych w zależności do wcześniejszego wyboru, przykładowe wyświetlenie zawartości pliku językowego (np menu) w zależności od wybranego języka. Aktualnie domyślnie dołączony jest polski plik językowy.

Slajd 5


 **B2E**  
BUSINESS TO EDUCATION

 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Cechy projektu

- Napisany jest w PHP z wykorzystaniem klas oraz zewnętrznej biblioteki Slim dla REST API
- Jest idealną bazą do testowania, rozwijania i nauki dla poszczególnych lekcji kursu
- Posiada przykłady walidacji, obsługi błędów poprzez wyjątki
- Operacje CRUD dla klas oraz uczniów


 **DAILY**  
GROUP


Lekcja 11 – Interdyscyplinarny projekt

CRUD - Create, read, update and delete

Slim - (<http://www.slimframework.com/>) jest przykładem frameworka, przy pomocy którego w szybki i prosty sposób będziemy mogli tworzyć własne API (np REST API dla aplikacji na Androidzie – Lekcja 11 Java)

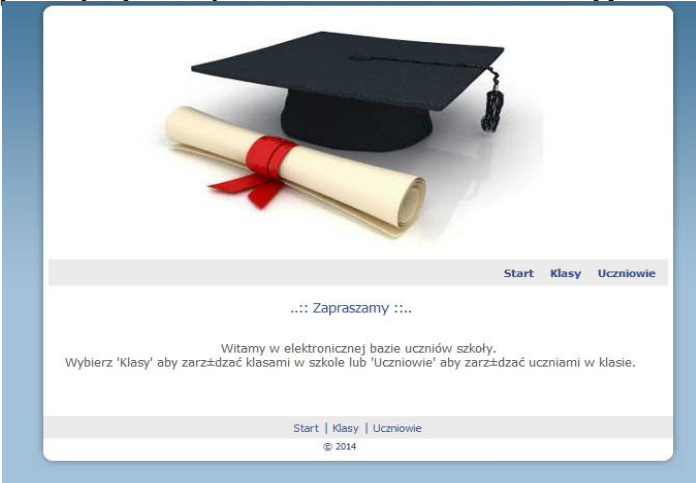
Slajd 6

 **B2E**  
BUSINESS TO EDUCATION

 SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – strona główna




Start | Klasy | Uczniowie

...: Zapraszamy ...:

Witamy w elektronicznej bazie uczniów szkoły.  
Wybierz 'Klasy' aby zarządzać klasami w szkole lub 'Uczniowie' aby zarządzać uczniami w klasie.

Start | Klasy | Uczniowie

© 2014

 **DAILY**  
GROUP

Lekcja 11 – Interdyscyplinarny projekt

Podstawowe komendy związane z transakcjami.



Slajd 7



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – widok klas

Klasy

ID	Nazwa klasy	Wychowawca	Edycja danych	Dodaj zdjęcie	Usuń
1	1A	Michał Gales		 (1)	
2	1B	Anna Komin		 (1)	
3	2C	Marcelina Jeż		 (1)	
4	3B	Jan Nowak		 (0)	
5	4A	Ewa Komin		 (0)	

Nazwa klasy:

Wychowawca:



Lekcja 11 – Interdyscyplinarny projekt

W podstawowym widoku klas wylistowana jest lista wszystkich klas w szkole. Wyświetlane jest ich:

- Id w bazie
- Nazwa klasy
- Wychowawca
- Przycisk do bezpośredniej edycji danej klasy
- Przycisk umożliwiający dodanie zdjęcia nauczyciela, oraz w nawiasie informacja czy zdjęcie już jest dodane (1) czy nie (0)
- Przycisk umożliwiający skasowanie klasy – jest to możliwe gdy w klasie nie ma żadnych uczniów

Poniżej znajduje się formularz umożliwiający dodanie nowej klasy.



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – edycja klasy

Edycja klasy:

Nazwa klasy:

Wychowawca:

**Klasy**

ID	Nazwa klasy	Wychowawca	Edycja danych	Dodaj zdjęcie	Usuń
1	1A	Michał Galas		 (1)	
2	1B	Anna Komin		 (1)	
3	2C	Marcelina Jez		 (1)	
4	3B	Jan Nowak		 (0)	
5	4A	Ewa Komin		 (0)	



Lekcja 11 – Interdyscyplinarny projekt

Formularz do edycji klasy, widoczny po wciśnięciu przycisku edycji klasy (symbol otwórka na kartce)



**B2E**  
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – edycja klasy, obsługa błędów

BŁĄD!  
Nie można zmienić nazwy tej klasy ponieważ inna klasa o podanej nazwie już istnieje

**Klasy**

ID	Nazwa klasy	Wychowawca	Edycja danych	Dodaj zdjęcie	Usuń
1	1A	Michał Galas		 (1)	
2	1B	Anna Komin		 (1)	
3	2C	Marcelina Jez		 (1)	
4	3B	Jan Nowak		 (0)	
5	4A	Ewa Komin		 (0)	



Lekcja 11 – Interdyscyplinarny projekt

Obsługa błędów – zabezpieczenie uniemożliwiające zmianę nazwy klasy, na klasę która już istnieje.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 10

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – widok klasy z uczniami

Start Klasy Uczniowie

Proszę wybrać klasę z listy:

1A - Michał Galas	Wyswietl
1A - Michał Galas	
1B - Anna Komin	
2C - Marcelina Jez	
3B - Jan Nowak	
4A - Ewa Komin	

Start | Klasy | Uczniowie

© 2014

Lekcja 11 – Interdyscyplinarny projekt

Widok klasy z uczniami – formularz do wyboru klasy, którą chcemy wyświetlić.

Slajd 11

**B2E**  
BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK**  
NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – widok klasy z uczniami

Klasa 1A, wychowawca: Michał Galas

ID	Imię	Nazwisko	Tel	Email	Edycja danych	Dodaj zdjęcie	Usuń	Szczegóły
1	Alojzy	Babek	534534543	alozjy@o2.pl	✎	👤 (1)	✖	⋮
2	Michał	Nowak	6543456	nowak@gmail.com	✎	👤 (1)	✖	⋮
3	Ewa	Nowak	23452345	ewa@gmail.com	✎	👤 (0)	✖	⋮

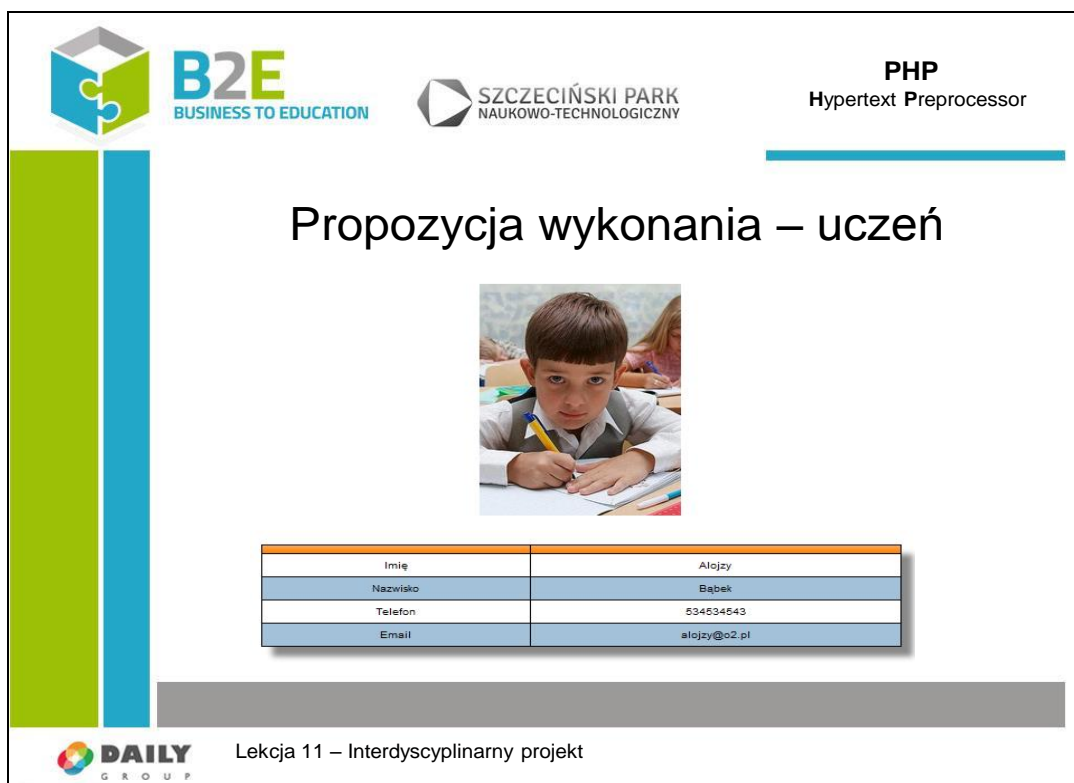
Lekcja 11 – Interdyscyplinarny projekt

Widok klasy z uczniami – po wybraniu klasy, widać informację o klasie, zdjęcie wychowawcy oraz listę uczniów w klasie.



Podobnie jak dla nauczyciela, dla konkretnych uczniów mamy dostępne opcje – edycję, dodanie zdjęcia, usunięcie oraz podgląd konkretnego ucznia.

Slajd 12




**B2E** BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – uczeń

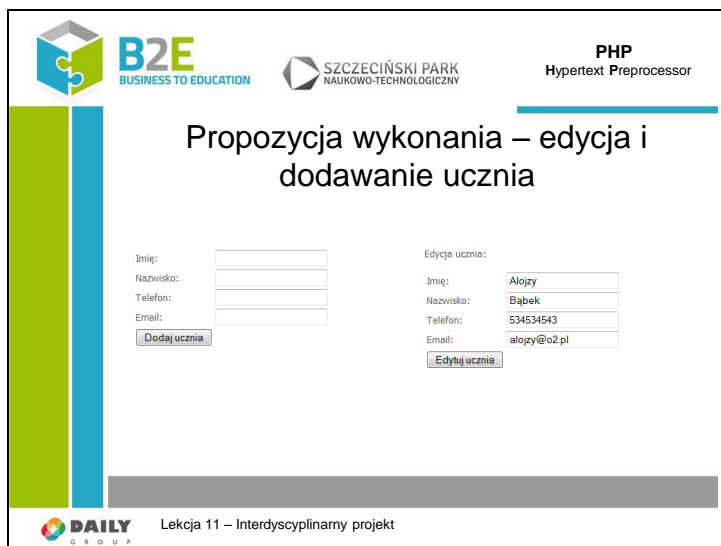


Imię	Alojzy
Nazwisko	Bąbek
Telefon	534534543
Email	alozjy@o2.pl

**DAILY** GROUP Lekcja 11 – Interdyscyplinarny projekt

Widok klasy z uczniami – po kliknięciu na „Szczegóły” dla konkretnego ucznia uruchamiamy podgląd pojedynczego ucznia.

Slajd 13



**B2E** BUSINESS TO EDUCATION

**SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY

**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – edycja i dodawanie ucznia

Imię:   
Nazwisko:   
Telefon:   
Email:

Edycja ucznia:

Imię:   
Nazwisko:   
Telefon:   
Email:


**DAILY** GROUP Lekcja 11 – Interdyscyplinarny projekt


Widok klasy z uczniami – edycja i dodawanie.



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 14






**PHP**  
Hypertext Preprocessor

## Propozycja wykonania – REST API

Do komunikacji serwera z aplikacją mobilną napisaną w Javie na telefony z Androidem wymagane jest API. Jest ona warstwą pośrednią pomiędzy klientem (smartphone) a bazą danych.


Stworzona propozycja umożliwia następujące operacje:

- /shool/class – wylistowanie klas w szkole
- /shool/class/:id – wylistowanie informacji o klasie oraz wszystkich jej uczniów, gdzie :id to identyfikator klasy w bazie
- /person/:id – wylistowanie informacji o uczniu, gdzie :id to identyfikator ucznia w bazie



Lekcja 11 – Interdyscyplinarny projekt

Slajd 15





**PHP**  
Hypertext Preprocessor

## REST API – przykładowa odpowiedź

Odpowiedź API po wywołaniu <http://localhost/rejestr/api/person/1>


```


{
 error: false
 _id: 1
 imie: "Alojzy"
 nazwisko: "Bąbek"
 tel: "534534543 "
 email: "alozjy@o2.pl"
 zdjecie: "http://localhost/rejestr/upload/uczen3.jpg"
}
```



Lekcja 11 – Interdyscyplinarny projekt

Slajd 16






**PHP**  
Hypertext Preprocessor

## Propozycje rozbudowy projektu w nawiązaniu do tematyki poszczególnych lekcji

Lekcja 1 - Bezpieczeństwo aplikacji – walidacja formularzy po stronie serwera




Lekcja 2 - Architektura usług sieciowych - rozbudowa REST API

Lekcja 3 - Protokół HTTP - implementacja dodatkowych metod REST API jak PUT, DELETE - wykorzystanie w praktyce



Lekcja 11 – Interdyscyplinarny projekt

Slajd 17




**Propozycje rozbudowy projektu w nawiązaniu do tematyki poszczególnych lekcji**

Lekcja 4 - Modyfikacja nagłówków HTTP - redirect przy logowaniu/wylogowaniu do aplikacji, modyfikacja strony kodowej wywołań do REST API




Lekcja 5 - Cookie - informacje o użytkowniku - śledzenie aktywności użytkownika rejestru uczniów

Lekcja 6 - Sesje - współdzielenie informacji - logowanie do aplikacji



Lekcja 11 – Interdyscyplinarny projekt

Slajd 18




**Propozycje rozbudowy projektu w nawiązaniu do tematyki poszczególnych lekcji**

Lekcja 7 - Poczta - systemy SMTP - dodanie możliwości wysyłania wiadomości e-mail z aplikacji (znamy adres email ucznia, mail wysyłany przy pomocy kodu z aplikacji, a nie otwarcie okienka aplikacji skojarzonej z pocztą)

Lekcja 8 - MIME - rozszerzenia multimedialne - wysyłanie zdjęcia




Lekcja 9 - UML - wyrysowanie projektu w UMLu

Lekcja 10 - Dobre praktyki programowania - biblioteki (mail, simple REST API), formatowanie kodu, komentarze



Lekcja 11 – Interdyscyplinarny projekt



Slajd 19




**Podsumowanie**

- Gotowa prosta aplikacja do gromadzenia informacji na temat uczniów i klas
- Idealna „baza” do testowania, wprowadzania poprawek i nauki na działającym przykładzie

Cibowik - najlepsza inwestycja



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



Lekcja 11 – Interdyscyplinarny projekt

Tabela podsumowująca

### Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie przygotowują dla siebie ich pierwszy projekt informatyczny – książkę teleadresowej kolegów i koleżanek z klasy i ze szkoły dostępną poprzez WWW oraz aplikację mobilną

## Bibliografia

Bougon M.K.E. , Weick D., Binkhorst D., Cognition in Organizations: An Analysis of the Utrecht Jazz Orchestra, *Administrative Science Quarterly* 22, 1997 [w:] baza cyfrowa ERIC.

Craik F.I.M., Lockhart R.S., Levels of processing: A Framework for Memory Research, *Jurnal of Verbal Learning and Verbal Behaviour*, 11, 1972, s. 671-684.

Czerepaniak – Walczak M., Perzycka E., (ed.) *Media and Trust – Theoretical and Practical Contexts*, Wydawnictwo Uniwersytetu Szczecińskiego, Szczecin, 2014.

Perzycka E., *Struktura i dynamika kompetencji informacyjnych nauczyciela w społeczeństwie sieciowym*, Wyd. Uniwersytet Szczeciński, Szczecin 2008.

Pety G., *Nowoczesne nauczanie. Praktyczne wskazówki i techniki dla nauczycieli, wykładowców i szkoleniowców*, Wydawnictwo GWP, Sopot 2010.

Syśło M., *Lekcja informatyki*, edunews.pl z dnia 06.02.2010.

*Thinking in Java (4th Edition)* Tłumaczenie: Przemysław Szeremiota na podstawie tłumaczenia Adriana Nowaka, Szymona Kobalczyka, Łukasza Fryza i Piotra Rajcy 2006.