

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2020-2024

Rok akademicki 2021/2022

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>bazy danych</i>
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	<i>Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>Informatyka</i>
Poziom studiów	<i>studia I stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok II semestr 3 i 4</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy inżynierski</i>
Język wykładowy	<i>język polski</i>
Koordinator	<i>dr inż. Piotr Grochowalski, dr inż. Dariusz Bober</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Piotr Grochowalski, dr inż. Dariusz Bober</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
3	30			30					5
4	15			30					4

1.2. Sposób realizacji zajęć

zajęcia częściowo w formie tradycyjnej, częściowo z wykorzystaniem metod i technik kształcenia na odległość

1.3. Forma zaliczenia przedmiotu /modułu (z toku)

zaliczenie z oceną po semestrze 3 i egzamin po semestrze 4.

2. WYMAGANIA WSTĘPNE

Rozumienie reguł przetwarzania podstawowych oraz złożonych typów danych w dowolnym języku proceduralnym.

Zaliczone przedmioty: podstawy programowania oraz programowanie obiektowe cz.1.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

3.1 Cele przedmiotu

C1	Dostarczenie wiedzy i wykształcenie umiejętności projektowania i eksploatacji systemów baz danych z wykorzystaniem języków strukturalnego i proceduralnego.
----	---

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Student zna podstawowe metody projektowania relacyjnych baz danych oraz wybrane postacie normalne. Zna polecenia języka SQL, sposób ich konstruowania oraz ich działanie i zastosowania. Student zna konstrukcje i elementy wybranego proceduralnego języka baz danych oraz ich działanie i zastosowania. Ma pogłębioną wiedzę o alternatywnych architekturach i modelach baz danych i konfrontuje te modele wobec modelu relacyjnego.	K_W07
EK_02	Potrafi projektować relacyjne bazy danych, przeprowadzać normalizację baz danych, konstruować polecenia języka SQL w celu przeprowadzenia pożądaných operacji. Potrafi programować w wybranym języku proceduralnym np. PL/SQL. Potrafi tworzyć aplikacje w wybranym języku programowania (np. desktopowe, internetowe lub na urządzenia mobilne), współpracujące z relacyjną bazą danych. Potrafi ocenić ryzyko wystąpienia problemów, spowodowanych przez błędy popełnione na etapie projektowania lub implementacji.	K_U07, K_U11
EK_03	Potrafi zrealizować przetwarzanie danych wewnątrz bazy danych wraz z implementacją w wybranym środowisku lub poza nią z wykorzystaniem odpowiednich bibliotek/narzędzi.	K_U12, K_U20

3.3 Treści programowe

A. Problematyka wykładu

Część pierwsza (semestr 3)
Wprowadzenie do systemów baz danych. Relacyjny model danych: struktury danych, ograniczenia wynikające z integralności danych, operacje.
Modelowanie pojęciowe: model związków-encji. Transformacja z modelu pojęciowego do relacyjnego. Normalizacja i denormalizacja schematu relacyjnej bazy danych.
Język opisu danych, omówienie składni języka DDL. Tworzenie, modyfikacja i destrukcja poszczególnych obiektów bazy danych. Słownik bazy danych.
Język manipulowania danymi, polecenia i składnia języka DML. Pojęcie transakcji. Zarządzanie transakcjami. Integralność danych, zarządzanie więzami integralności. Indeksy i optymalizacja

bazy danych.
Język SQL, projekcja, selekcja, grupowanie, sortowanie, połączenie, suma, iloczyn, różnica, podzapytania, zapytania skorelowane.
Wybrane funkcje języka SQL: numeryczne, znakowe, daty, konwersji, warunkowe.
Zagadnienia bezpieczeństwa informacji w systemach zarządzania bazami danych. Zarządzanie prawami dostępu do danych, poziomy uprawnień.
Część druga (semestr 4)
Proceduralny język bazy danych PL/SQL: definiowanie danych, instrukcje sterujące, obsługa kursorów, obsługa wyjątków.
Funkcje i procedury składowane w bazie danych. Sekwencje.
Aktywne bazy danych: procedury wyzwalane (ang. triggers).
Podstawy teoretyczne projektowania interfejsu użytkownika. Architektura interfejsu użytkownika.
Odwzorowanie struktur danych w graficznym interfejsie użytkownika. Rodzaje elementów graficznych.
Właściwości elementów graficznych i sposoby ich definiowania. Omówienie funkcjonalności aplikacji.
Omówienie dwu i trójwarstwowej architektury aplikacji, z przykładami implementacji z użyciem współczesnych języków programowania (np.: C#, Java, PHP).
Wprowadzenie do systemów nierelacyjnych baz danych przy uwzględnieniu rysu historycznego rozwoju baz danych, przeglądu kluczowych własności relacyjnych baz danych, z wyszczególnieniem tych ograniczeń, które determinują potrzebę opracowania modeli nierelacyjnych. Przegląd najpopularniejszych modeli nierelacyjnych.

B. Problematyka ćwiczeń laboratoryjnych

Część 1 (semestr 3)
Modelowanie pojęciowe: model związków-encji. Transformacja z modelu pojęciowego do relacyjnego. Normalizacja i denormalizacja schematu relacyjnej bazy danych.
Język opisu danych, omówienie składni języka DDL. Tworzenie, modyfikacja i destrukcja poszczególnych obiektów bazy danych. Język manipulowania danymi, polecenia i składnia języka DML. Pojęcie transakcji. Zarządzanie transakcjami. Integralność danych, zarządzanie więzami integralności. Indeksy i optymalizacja bazy danych.
Język SQL, projekcja, selekcja, grupowanie, sortowanie, połączenie, suma, iloczyn, różnica, podzapytania, zapytania skorelowane.
Wybrane funkcje języka SQL: numeryczne, znakowe, daty, konwersji, warunkowe.
Część 2 (semestr 4)
Proceduralny język bazy danych PL/SQL: definiowanie danych, instrukcje sterujące, obsługa kursorów, obsługa wyjątków.
Funkcje i procedury składowane w bazie danych. Sekwencje.
Aktywne bazy danych: procedury wyzwalane (ang. triggers).
Projekt aplikacji dwu- lub trójwarstwowej (warstwa bazy danych, model związków encji, implementacja w środowisku relacyjnym, procedury składowane realizujące scenariusze biznesowe; warstwa serwera aplikacji i GUI w dowolnym języku programowania)

3.4 Metody dydaktyczne

Wykład: wykład problemowy, wykład z prezentacją multimedialną.

Ćwiczenia: praca indywidualna przy komputerze, metoda projektów (projekt praktyczny).

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, lab)
EK_01	S1: Obserwacja w trakcie zajęć, weryfikacja wiedzy w postaci kolokwium/testu zaliczeniowego S2: Obserwacja w trakcie zajęć, Egzamin	wykład
EK_02	S1: Kolokwium S2: Projekt	laboratorium
EK_03	S2: Projekt	laboratorium

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

W semestrze pierwszym

Zaliczenie wykładu:

Następuje na podstawie pozytywnie ocenionej wiedzy z zakresu zagadnień z pierwszej części wykładu. Weryfikacja wiedzy może odbywać się w postaci np. kolokwium pisemnego, kolokwium testowego lub w innej formie. Negatywna ocena wiedzy studenta skutkuje brakiem zaliczenia wykładu.

Zaliczenie laboratorium:

Następuje na podstawie zaliczenia dwóch kolokwiów z:

1) Projektowania i normalizacji baz danych, podczas którego studenci rozwiązują zadania dotyczące tworzenia diagramu ERD na podstawie wymagań podanych w opisie wybranego wycinka rzeczywistości, na którym umieszczają odpowiedniki encji, atrybutów i związków pomiędzy encjami, następnie definiują istniejące (lub proponują nowe) reguły integralności. W dalszej części kolokwium na podstawie przedstawionych tabel/baz danych studenci muszą dokonać określenia obecnej postaci bazy danych lub udzielenia odpowiedzi czy spełniona jest konkretna zadana postać (np. BCNF), a także etapowego przeprowadzenia procesu normalizacji do 3 postaci normalnej.

Za kolokwium przyznawana jest ocena, w zależności proporcjonalnie od uzyskanej liczby punktów za całość. Ocena *dostateczny* wymaga uzyskania przynajmniej 50% punktów.

2) Języka SQL, w którym studenci na podstawie zadanych pytań w mowie potocznej potrafią napisać: zapytania SQL oparte o 1...n tabel źródłowych, a także zapytania zawierające złączenia tabel, grupujące wiersze, zawierające formuły wbudowane w system bazy danych w zakresie przetwarzania tekstu, konwersji formatu daty i wartości numerycznych, zapytania zagnieżdżone np. podzapytania skorelowane.

Oprócz zapytań SQL studenci formułują także polecenia DDL w zakresie implementacji modelu

ERD w bazie danych oraz w zakresie rekonfiguracji struktury baz danych, a także tworzące widoki (perspektywy).

Za kolokwium przyznawana jest ocena, w zależności proporcjonalnie od uzyskanej liczby punktów za całość. Ocena *dostateczny* wymaga uzyskania przynajmniej 50% punktów.

Efekt EK_02 jest uznany za zaliczony, gdy student z obu kolokwii otrzyma ocenę przynajmniej „dostateczny”.

W semestrze drugim

Zaliczenie wykładu następuje w oparciu o zaliczenie efektów EK_01 – EK_03 w ramach zaliczonego projektu i zaliczonego egzaminu.

Zaliczenie laboratorium następuje na podstawie zaliczenia efektów EK_02 i EK_03 poprzez realizację projektu aplikacji, komunikującej się z bazą danych oraz wykorzystującej język proceduralny na poziomie baz danych. Wymagania na poszczególłą ocenę:

dostateczny:

W ramach laboratorium studenci zostają podzieleni na zespoły. Studenci w ramach zespołów przy wykorzystaniu mechanizmów pracy zespołowej mają zaprojektować bazę danych w wybranym temacie, następnie zaimplementować ją w wybranym środowisku bazodanowym wspierającym proceduralny język bazy danych np.: PL/SQL. Baza danych powinna opisywać wybrany temat w postaci tabel połączonych między sobą różnymi typami powiązań. W zdefiniowanej bazie danych przy wykorzystaniu języka proceduralnego mają zostać zaimplementowane operacje CRUD dla każdej z tabel.

W ramach projektu ma zostać zaprogramowana aplikacja (desktopowa/internetowa/mobilna) mająca dostęp do bazy danych (do danych w niej zawartych), do tego celu może zostać wybrany dowolny język programowania (zaakceptowany przez prowadzącego) oraz dowolna technologia/biblioteka do komunikacji z bazą danych. W aplikacji ma zostać zaimplementowane przetwarzanie danych z bazy danych lub wywoływanie odpowiednich funkcjonalności znajdujących się wewnątrz bazy, w celu ich przetwarzania.

Po realizacji praktycznej projektu ma on zostać wyposażony w dokumentację przedstawiającą wszystkie aspekty procesu tworzenia oraz powstałą aplikację.

dobry:

Student spełnia kryterium oceny dostateczny, a ponadto powinien w ramach realizowanego projektu zawrzeć dodatkowe funkcjonalności po stronie bazy danych zaimplementowane przy użyciu języka proceduralnego w postaci procedur, funkcji, wyzwalaczy, itp.

bardzo dobry:

Student spełnia kryterium oceny dobry, a ponadto powinien w ramach realizowanego projektu zawrzeć dodatkowe funkcjonalności po stronie bazy danych zaimplementowane przy użyciu języka proceduralnego w postaci wykorzystania typów złożonych, kursorów, pakietów, itp.

Efekty EK_02 i EK_03 są uznane za zaliczone, gdy projekt wykonany przez studenta spełni wszystkie wymagania określone przynajmniej na ocenę „dostateczny”.

Zaliczenie egzaminu następuje poprzez uzyskanie oceny pozytywnej (ocena 3,0 i wyżej) z egzaminu pisemnego, który weryfikuje wiedzę z zakresu zagadnień z pierwszej i drugiej części przedmiotu.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	105
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	4
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	116
SUMA GODZIN	225
SUMARYCZNA LICZBA PUNKTÓW ECTS	9

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	---
zasady i formy odbywania praktyk	---

7. LITERATURA

<p>LITERATURA PODSTAWOWA:</p> <ol style="list-style-type: none"> 1. Jeffrey D. Ullman: Podstawowy wykład z systemów baz danych, Warszawa; Wydawnictwa Naukowo-Techniczne, 2001. 2. Dariusz Put: Bazy danych: pojęcia, projektowanie, podstawy SQL, Kraków; Wydaw. Uniwersytetu Ekonomicznego, 2007. 3. Krystyna Czapla: Bazy danych: podstawy projektowania i języka SQL, Gliwice; Helion, 2015. 4. Barbara Pękala: Bazy danych: teoria i praktyka, Wydawnictwo Uniwersytetu Rzeszowskiego, 2015.
<p>LITERATURA UZUPEŁNIAJĄCA:</p> <ol style="list-style-type: none"> 1. Andrzej Barczak, Jacek Florek, Tadeusz Sydoruk: Bazy danych. Akademia Podlaska. Wydział Nauk Ścisłych. Instytut Informatyki. Siedlce; Wydaw. AP, 2006. 2. Adam Pelikant: Bazy danych: pierwsze starcie. Gliwice; Wydaw. Helion, 2009. 3. Marcin Szeliga: ABC języka SQL. Gliwice; Wydaw. Helion, 2002. 4. Rafe Coburn, "SQL: dla każdego. Gliwice; Wydaw. Helion, 2001. 5. Judith S. Bowman, Sandra L. Emerson, Marcy Darnovsky: Podręcznik języka SQL. Warszawa; Wydawnictwa Naukowo-Techniczne, 2001. 6. Ryan K. Stephens i in.: SQL w 3 tygodnie. Warszawa; LT&P, 1999. 7. Michael J. Hernandez, John L. Viescas [tł. Piotr Nowakowski]: Zapytania SQL dla zwykłych śmiertelników: praktyka obróbki danych w języku SQL. Warszawa; Wydaw. Mikom, 2001

Akceptacja Kierownika Jednostki lub osoby upoważnionej