

# Bazy danych

## Wykład 3

**Temat:** Metodologia tworzenia baz danych

**Sławomir Świętoniowski**

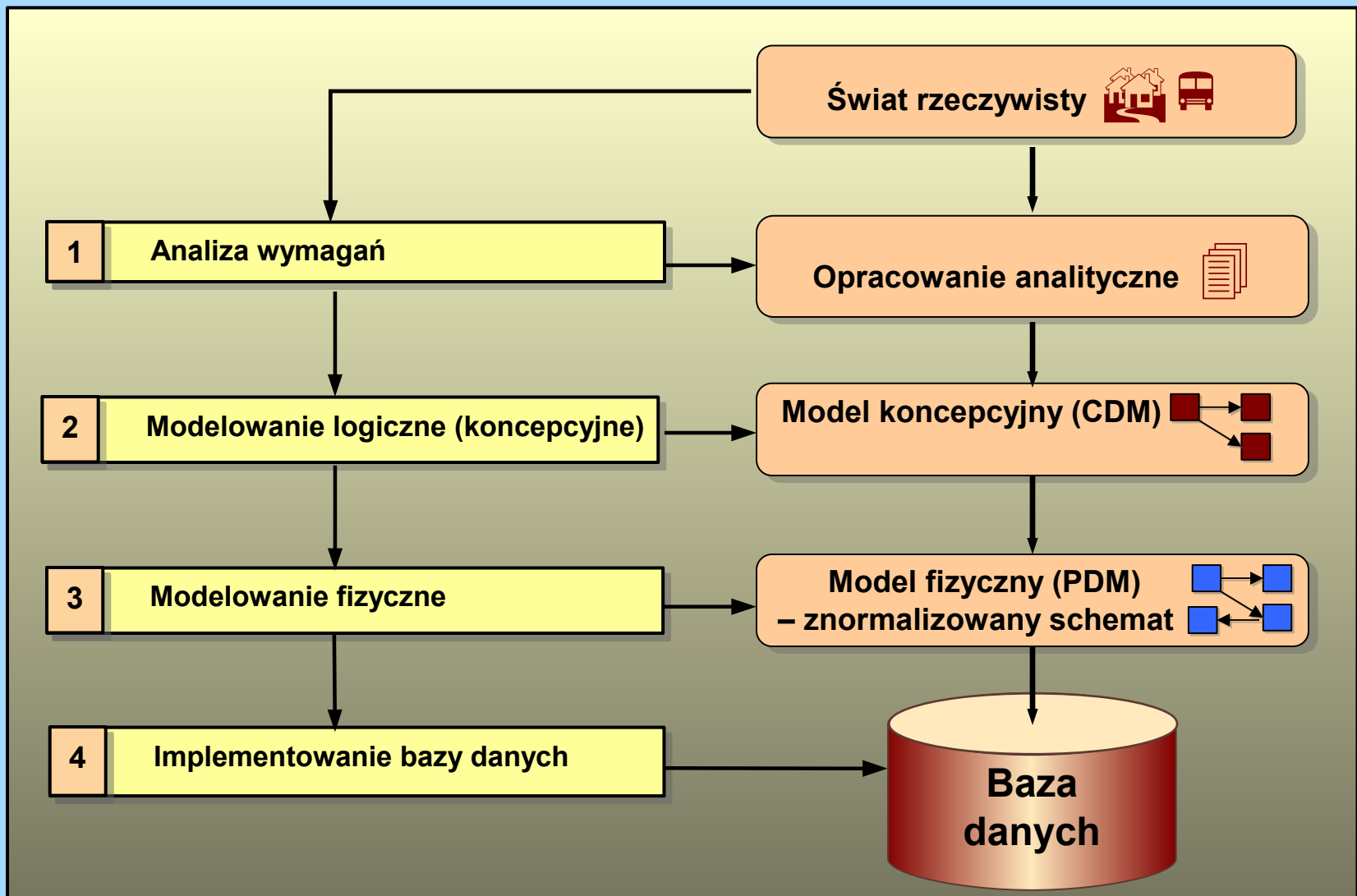
slawomir-swietoniowski@wp.pl

# Plan wykładu

---

1. Etapy tworzenia bazy danych.
2. Prosta metodologia tworzenia relacyjnych baz danych.
3. Przykład analizy dziedziny „Wyższa uczelnia techniczna”.

# Etapy tworzenia bazy danych



# 1. Analiza wymagań

---

- Rozpoznanie systemu i zachodzących w nim procesów, które mają być modelowane.
- Określenie celu, zakresu i funkcji systemu.
- Zidentyfikowanie głównych grup użytkowników i ich sposobu korzystania z systemu.
- Analiza korzyści z nowego systemu, w porównaniu z obecnym rozwiązaniem.

**Powstaje: opracowanie analityczne** – dokument zawierający opis tekstowy oraz schematy (np. diagramy modelu procesów biznesowych BPM), których dobór zależy od przyjętej metodologii.

## 2. Modelowanie logiczne

---

- Definiowanie obiektów, ich atrybutów oraz relacji między obiektami.
- Modelowanie zachowania systemu (bez implementacji).
- Na tym poziomie celowo pomija się uwarunkowania implementacyjne (np. system bazodanowy, język programowania, platformę systemową), aby móc się skoncentrować na właściwym przygotowaniu koncepcyjnego modelu systemu.

**Powstaje: logiczny model danych** (ang. *conceptual data model*, CDM)  
– opisujący strukturę bazy (obiekty, powiązania między nimi)  
w sposób niezależny od docelowego systemu DBMS.

# 3. Modelowanie fizyczne

---

- Tworzenie schematu bazy danych dla konkretnego systemu bazodanowego.
- Normalizacja schematu bazy – standardowo do 2–3 NF.
- Analiza integralności danych – opis więzów deklaratywnych i proceduralnych.
- Analiza ilościowa, użycia i transakcji – optymalizacja schematu bazy.
- Analiza zabezpieczeń i kontroli – projekt systemu bezpieczeństwa.
- Projekt struktur przechowywania (np. rozkładu plików bazy na dyskach).
- Na tym poziomie rozstrzyga się wszystkie istotne kwestie techniczne (m.in. platforma systemowa, system bazodanowy, architektura aplikacji klient-serwer, język programowania dla poszczególnych warstw systemu).

**Powstaje: fizyczny model danych** (ang. *physical data model*, PDM)  
– stanowiący projekt bazy dostosowany do konkretnego systemu DBMS (np. Oracle) w celu uzyskania jak największej wydajności.

# 4. Implementacja bazy danych

**Implementacja poniższych elementów** (ręczna lub automatyczna):

- schemat bazy danych: tabele i relacje;
- więzy integralności:
  - wewnętrzne – deklaratywne (ograniczenia PK, FK, CHECK, UNIQUE, DEFAULT, reguły, wartości domyślne);
  - dodatkowe – proceduralne (np. wyzwalacze – ang. *triggers*);
- obiekty kodu SQL, przechowywane na serwerze:
  - procedury przechowywane dla transakcji i operacji CRUD (INSERT, SELECT, UPDATE, DELETE);
  - widoki;
- indeksy – na podstawie struktury bazy danych i sposobu jej wykorzystania;
- system bezpieczeństwa – konta, role, użytkownicy i ich uprawnienia.

**Powstaje: baza danych** (ang. *database*, DB) – realizacja modelu fizycznego w konkretnym środowisku technicznym, gotowa do wdrożenia, optymalizacji i strojenia (ang. *tuning*).

# Plan wykładu

---

1. Etapy tworzenia bazy danych.

**2. Prosta metodologia tworzenia relacyjnych baz danych.**

3. Przykład analizy dziedziny „Wyższa uczelnia techniczna”.



# Prosta metodologia tworzenia relacyjnych baz danych

1. **Analiza dziedziny:** rozpoznanie systemu i zachodzących w nim procesów, które mają być modelowane.
2. **Projekt i implementacja schematu bazy danych (2–3NF):** tabele i relacje.
3. **Projekt i implementacja więzów integralności:**
  - **wewnętrznych – deklaratywnych** (ograniczenia PK, FK, CHECK, UNIQUE, DEFAULT, reguły, wartości domyślne);
  - **dodatkowych – proceduralnych** (wyzwalacze – ang. *triggers*).
4. **Tworzenie widoków i procedur przechowywanych dla operacji CRUD** – instrukcje INSERT, SELECT, UPDATE, DELETE; transakcje.
5. **Projekt i implementacja indeksów** – na podstawie struktury bazy danych i sposobu jej wykorzystania.
6. **Projekt i implementacja systemu bezpieczeństwa** – konta, role, użytkownicy i ich uprawnienia.
7. **Wdrożenie, konfiguracja zadań automatycznych i utrzymania bazy.**

# Prosta metodologia

## – zastosowanie

---

- Przedstawiona metodologia może być wykorzystana do:
  - szybkiego tworzenia baz danych o małej i średniej złożoności;
  - tworzenia warstwy bazodanowej w aplikacji klient-serwer (zwłaszcza dwuwarstwowych – baza danych + interfejs wraz z logiką biznesową);
  - tworzenia niewielkich aplikacji, w których nie jest wymagana rozbudowana dokumentacja;
  - budowy systemów w warunkach braku zaawansowanych narzędzi do projektowania (np. opartych na języku UML).
- Metodologia ta nie jest zalecana, jeśli:
  - tworzony jest system o dużej złożoności (np. aplikacja wielowarstwowa klient – serwer z dużą liczbą współpracujących ze sobą obiektów w poszczególnych warstwach lub z bardzo zaawansowanymi algorytmami);
  - dysponujemy dobrymi narzędziami do modelowania (np. PowerDesigner);
  - integrujemy lub rozbudowujemy wiele istniejących systemów baz danych (szczególnie heterogenicznych – z różnymi systemami DBMS).

# Plan wykładu

---

1. Etapy tworzenia bazy danych.
2. Prosta metodologia tworzenia relacyjnych baz danych.
- 3. Przykład analizy dziedziny „Wyższa uczelnia techniczna”.**

# Przykład dziedziny modelowania

## Wyższa uczelnia techniczna

Skrócony opis dziedziny.



- Uczelnia prowadzi studia inżynierskie z zakresu informatyki. Na program nauczania składają się przedmioty, przy czym każdy może mieć kilka form dydaktycznych (np. wykład, laboratorium).
- Do każdego przedmiotu przyporządkowywany jest co najwyżej jeden pracownik naukowo-dydaktyczny, który odpowiada za program nauczania i prowadzi nadzór merytoryczny nad jego wykonaniem.
- Uczelnia zatrudnia pracowników:
  - administracyjnych (pracują w dziekanacie, kwesturze itp.);
  - technicznych (np. konserwatorzy, administratorzy sieci i sprzętu komputerowego);
  - naukowo-dydaktycznych (prowadzą zajęcia ze studentami, piszą publikacje itp.).
- Każdy student ma przyporządkowywany niepowtarzalny numer indeksu z danego zakresu liczbowego.
- Student wpłaca uczelni miesięczne opłaty, uprawniające do studiowania.

# Przykład dziedziny modelowania

## Wyższa uczelnia techniczna

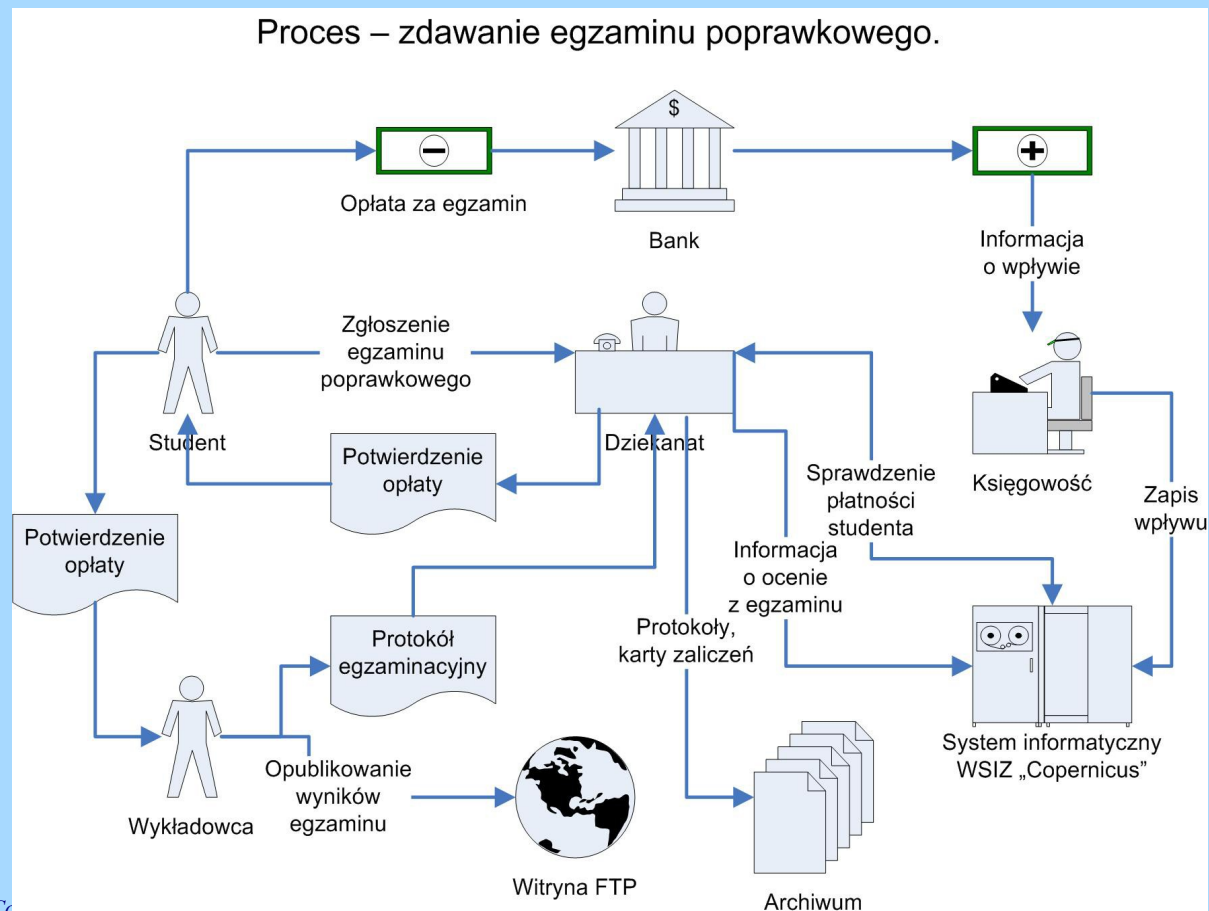
---

Skrócony opis dziedziny – ciąg dalszy.

- Studenci jednego roku są dzieleni na grupy, których nazwy są unikalne w skali jednego semestru. Jeden student może jednocześnie należeć do jednej grupy; grupa może zawierać od 15 do 30 studentów.
- Każda grupa studentów może mieć co najwyżej jednego studenta, który ją reprezentuje, nazywanego „starostą”.
- Przedmioty są realizowane przez określonych pracowników naukowo-dydaktycznych, w danym roku akademickim i semestrze, w danej grupie studentów lub indywidualnie (tylko przedmiot „Praca dyplomowa”).
- Student jest zapisywany na zajęcia z określonego przedmiotu – poprzez grupę (przedmioty grupowe) albo indywidualnie („Praca dyplomowa” lub inny przedmiot, realizowany w ramach indywidualnego toku studiów).
- Z każdego przedmiotu student w danym semestrze uzyskuje jedno lub więcej zaliczeń (1 – 3 terminy dla egzaminów i 1 – 2 terminy dla innych zaliczeń).

# Schemat działania systemu

Bardzo cennym uzupełnieniem słownego opisu działania systemu są diagramy (np. BPM, WFD), za pomocą których można czytelnie przedstawić np. obiekty, relacje i procesy zachodzące w danej organizacji (poniżej przykład WFD).



# Wyodrębnienie klas obiektów

Analizując słowny opis dziedziny, szukamy pojęć, obiektów, znaczeń, które dają się wyodrębnić z modelowanej rzeczywistości.

- **Przedmiot** (Kod, Nazwa, Semestr, ECTS);
- **Pracownik** (NrPracownika, Imie, Nazwisko, NIP, PESEL, Ulica, NumerDomu, NumerMieszkania, KodPocztowy, Miejscowosc, TelefonStac, TelefonKom, Email, Tytuł, Stanowisko, DataZatrudnienia, OkresZatrudnienia, TypZatrudnienia, Wynagrodzenie);
- **Student** (NrIndeksu, Nazwisko, Imie, NIP, PESEL, Ulica, NumerDomu, NumerMieszkania, KodPocztowy, Miejscowosc, TelefonStac, TelefonKom, Email, DataZapisania, DataWypisania);
- **Grupa** (NrGrupy, RokAkademicki);
- **Zajęcia** (FormaZajec, LiczbaGodzin, WymaganiaWstepne, Opis);
- **Zaliczenie** (DataZaliczenia, Ocena, RodzajZaliczenia, Termin);
- **Oплата** (DataWpłaty, Wpłacający, Miesiąc, Rok, Kwota).

# Powiązania między obiektami

---

- **Pracownik – Przedmiot** – dany pracownik prowadzi dany przedmiot w danej grupie, w określonym semestrze i roku akademickim.
- **Student – Opłata** – pozwala na ewidencjonowanie opłat studenta.
- **Student – Grupa** – wyraża przynależność studenta do grupy.
- **Student – Pracownik – Przedmiot** – student zapisuje się na dany przedmiot do określonego prowadzącego;
- **Student – Przedmiot – Zaliczenie** – po zapisaniu się na dany przedmiot, student uzyskuje z niego zaliczenie (może być kilka prób);



# Funkcje systemu – opis ogólny

---

- **Pracownik, Grupa, Przedmiot, Student, Opłata**  
– dodawanie, edycja, usuwanie.
- **Przypisanie studenta do grupy.**
- **Przypisanie pracownika do prowadzenia przedmiotu w określonej grupie studentów.**
- **Zapisanie studenta na przedmiot** do określonego prowadzącego (dotyczy przedmiotu „Praca dyplomowa”, który jest realizowany indywidualnie, a nie przez całą grupę).
- **Wpis zaliczenia studenta z danego przedmiotu.**

# Wybrane funkcje systemu

## – specyfikacja

---

- **Przypisanie studenta do grupy:**
  1. Jeśli nie jest zapisany, zapisanie studenta do bazy.
  2. Jeśli nie istnieje grupa, dodanie nowej grupy.
  3. Zapisanie studenta do grupy.
- **Przypisanie pracownika do prowadzenia przedmiotu w określonej grupie studentów:**
  1. Jeśli nie istnieje pracownik, zapisanie pracownika do bazy.
  2. Jeśli nie istnieje przedmiot, dodanie nowego przedmiotu.
  3. Jeśli nie istnieje grupa, dodanie nowej grupy.
  4. Przypisanie pracownika do danego przedmiotu.
- **Zapisanie studenta na przedmiot do określonego prowadzącego:**
  1. Jeśli żaden pracownik nie został przypisany do wybranego przedmiotu  
→ Proces: przypisanie pracownika do przedmiotu.
  2. Jeśli nie ma studenta w bazie, zapisanie studenta do bazy.
  3. Jeśli student zaliczył wymagane przedmioty z poprzedniego semestru i uiścił opłatę za nowy semestr, to zapisz studenta na przedmiot.
  4. W przeciwnym razie, odmowa zapisu.

# Grupy użytkowników

W systemie wyróżniamy 5 grup użytkowników.

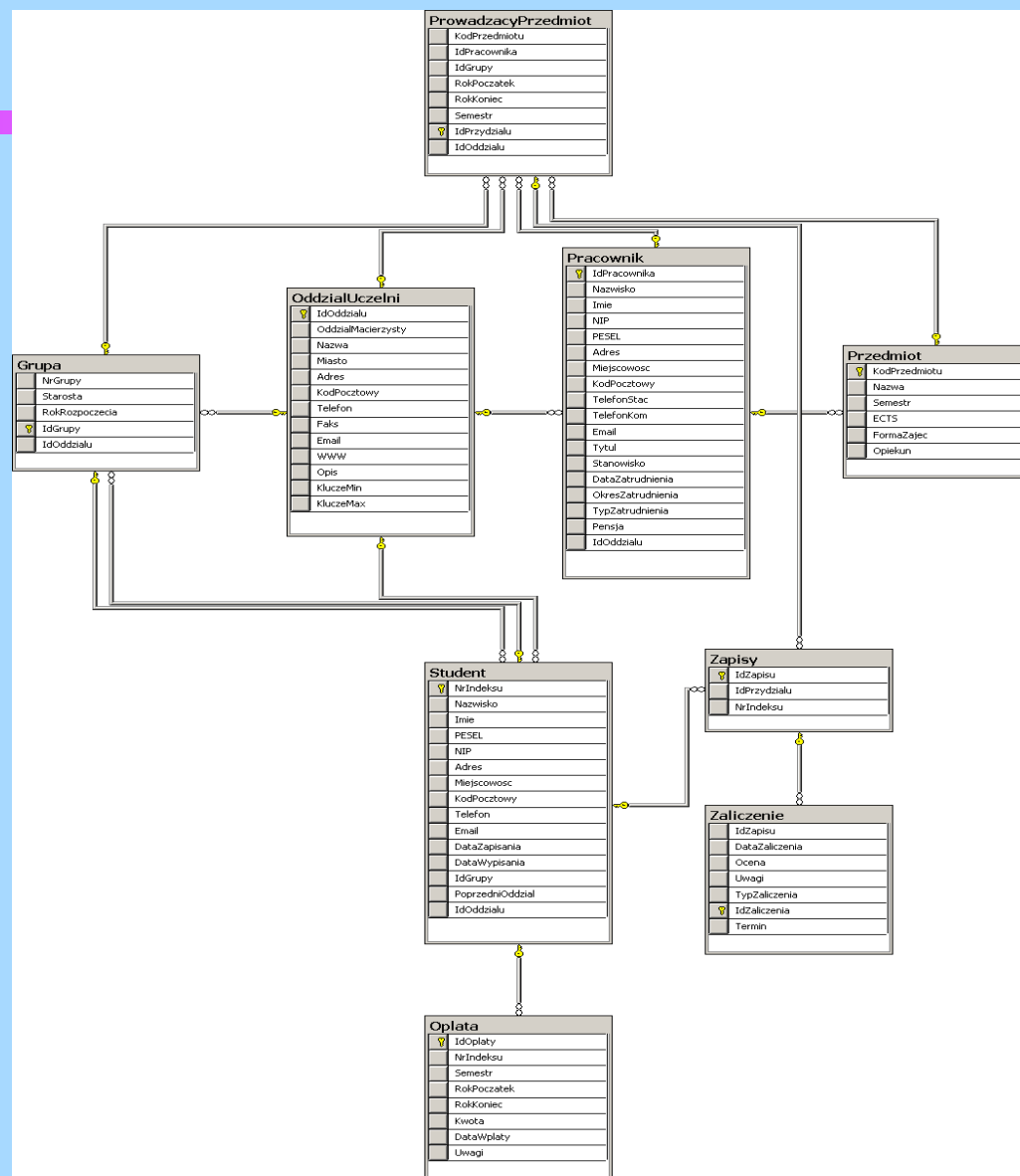
- **Administratorzy** – nieograniczone prawa operowania na danych i modyfikacji ustawień systemu. Do tej grupy należą administratorzy systemu informatycznego oraz kierownictwo uczelni.
- **Pracownicy naukowo-dydaktyczni** – mogą: przeglądać i częściowo edytować dane prowadzonych przez nich przedmiotów, zmieniać swoje dane osobowe i zarządzać zaliczeniami do momentu ich zatwierdzenia, odczytywać płatności za zaliczenia poprawkowe (tylko dla swoich studentów). Pozostałe obiekty – tylko do odczytu.
- **Pracownicy administracyjni** – mogą dodawać, odczytywać i modyfikować dane dotyczące pracowników, studentów, zajęć i opłat studentów. Mogą w pełni zarządzać grupami. Pozostałe obiekty – tylko do odczytu;
- **Studenci** – mogą przeglądać dane przedmiotów, grup i zajęć. Mogą także odczytać: ograniczone dane pracowników (tytuł, imię, nazwisko, stanowisko), swoje dane osobowe (+ prawo edycji telefonu i adresu e-mail), swoje zaliczenia i opłaty.
- **Goście** – na przykład nie zalogowani użytkownicy internetowi. Mogą wyłącznie czytać dane przedmiotów i ograniczone dane pracowników (tytuł, imię, nazwisko, stanowisko).

# Uprawnienia na poziomie obiektów

Poniżej przedstawiona jest specyfikacja uprawnień grup użytkowników na poziomie wyodrębnionych obiektów, w formie tzw. **macierzy CRUD** (Create, Read, Update, Delete). Gwiazdka (\*) wskazuje, że określone uprawnienia są ograniczone, zgodnie z podanym wcześniej opisem słownym.

	Administratorzy	PracownicyND	PracownicyADM	Studenci	Goscie
Przedmiot	CRUD	-RU*-	-R—	-R—	-R—
Pracownik	CRUD	-RU*-	CRU—	-R*-	-R*-
Student	CRUD	-R—	CRU—	-RU*-	—
Grupa	CRUD	-R—	CRUD	-R—	—
Zajecia	CRUD	-R—	CRU—	-R—	—
Zaliczenie	CRUD	CRUD*	-R—	-R*-	—
Oplata	CRUD	-R*-	CRU—	-R*-	—

# Schemat bazy



Zrzut ekranowy przedstawia bazę danych „Wyższa uczelnia techniczna”, wersja 1.1.

# Literatura

---

1. BEYNON-DAVIES P., *Systemy baz danych – nowe wydanie*, WNT, Warszawa 2003.
2. MICROSOFT, *Books On-Line* – dokumentacja systemu *MS SQL Server*, Microsoft Corp. 1988 – 2000.
3. PANKOWSKI T., *Podstawy baz danych*, Wydawnictwo Naukowe PWN, Warszawa, 1992.
4. RANKINS R., JENSEN P., BERTUCCI P., *Microsoft SQL Server 2000. Księga eksperta.*, HELION, Gliwice 2003 (książka dostępna w bibliotece WSIZ „Copernicus”).

# Bazy danych

## Wykład 5

Dziękuję za uwagę !

**Damian Dudek**