



Uniwersytet Gdański  
Wydział Matematyki, Fizyki i Informatyki  
Instytut Informatyki

# Wirtualna uczelnia

Damian Shroder

Projekt z przedmiotu bazy danych na kierunku informatyka profil ogólnoakademicki na Uniwersytecie Gdańskim.

Gdańsk  
26 maja 2020

## Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Opis projektu</b>	<b>2</b>
2.1	Potencjalne grupy użytkowników . . . . .	2
2.2	Wymagania funkcjonalne . . . . .	3
2.3	Wymagania niefunkcjonalne . . . . .	3
2.4	Diagram związków encji . . . . .	5
<b>3</b>	<b>Przykłady realizacji bazy danych</b>	<b>6</b>
3.1	Przykłady zawartości najważniejszych tabel . . . . .	6
3.2	Przykłady kilku zapytań i ich wyników . . . . .	8

# 1 Wprowadzenie

Baza danych przeznaczona jest dla uczelni. Znajdują się w niej takie informacje o wydziałach, kierunkach, studentach, klasach, nauczycielach, kursach, i wiele więcej.

Podstawowe pojęcia:

**Baza danych** - zorganizowany zbiór informacji zawierający jednolity rodzaj danych,

**Rekord** - pojedynczy wiersz w tabeli, który zawiera informacje dotyczące określonego elementu,

**Pole** - pojedyncza kolumna w tabeli, która określa rodzaj przechowywanych w niej informacji,

**Klucz podstawowy** - pojedyncze pole, którego wartości w danej tabeli są unikatowe, identyfikuje on poszczególne elementy,

**Klucz obcy** - pole, które jest kluczem podstawowym w innej tabeli,

**Kwerenda** - zapytania, które mają na celu powstanie tabel wirtualnych, są tworzone na chwilę, nie zapisują się,

**Formularz** - interfejs graficzny do wprowadzania danych do tabeli,

**Raport** - służy do wyświetlania danych z tabel i/lub kwerend w sposób przygotowany do wydruku,

**Model relacyjny** - istnieje kilka tabel, które są połączone relacjami,

**System baz danych** - skomputeryzowany system przechowywania danych, zorganizowany w pliku,

**Encja** - każdy przedmiot, zjawisko, stan, pojęcie, obiekt, który potrafimy odróżnić od innych obiektów,

**Atrybuty** - własności encji,

**Związek** - nazwana zależność pomiędzy podstawowymi zbiorami encji,

**Normalizacja** - proces projektowania baz danych, tak aby utworzyć zbiór tabel o odpowiedniej strukturze.

## 2 Opis projektu

Projekt powstał na zlecenie rady szkolnej Uniwersytetu Bolesławskiego. Jest to część większego projektu, który ma na celu zmodernizowanie obecnego systemu szkolnego. Zlecone zadania zostało wykonane zgodnie z najwyższymi normami.

### 2.1 Potencjalne grupy użytkowników

Potencjalni użytkownicy bazy danych to:

- **Administrator** - główny zarządca bazy danych, posiada pełen dostęp do bazy danych

- **Informatyk** - posiada większość uprawnień, dzięki którym nie trzeba za każdym razem wzywać głównego administratora
- **Nauczyciel** - posiada dostęp do bazy studentów, listy obecności, przedmiotów jakich naucza, kursów oraz edycji ocen
- **Student** - może sprawdzić przydział klas, swoje dane, liste przedmiotów, kursów, na które uczeszcza oraz wgląd w oceny

## 2.2 Wymagania funkcjonalne

Nasza baza danych będzie posiadała informacje o studentach, nauczycielach, ich tytule/stopniu, przedmiotach, kursach, ich poziomie, klasach, lekcjach i obecności na nich, wydziałach, kierunkach.

Dostępny jest proces złożenia pracy dyplomowej, dodania wielu adresów - także klas, kursów, przedmiotów, ocen - dla jednej osoby.

Wprowadzono zabezpieczenia m.in. do takich pól jak:

- Numer telefonu,
- Kod pocztowy,

Zautomatyzowano m.in. następujące pola:

- Email - wpisywany jest na podstawie pierwszej litery imienia oraz nazwiska,
- Data modyfikacji - jest aktualizowana przy dowolnej zmianie rekordu,
- Wynik w procesie składania pracy dyplomowej jest obliczany na podstawie pomniejszych ocen,
- Data obronienia pracy dyplomowej jest wpisywana automatycznie, jeżeli obrona jest pozytywna (zaliczona).

Dla ułatwienia wpisywania danych zastosowano formularze, w których można bezpośrednio wprowadzać studentów, nauczycieli, wystawiać oceny czy przypisywać przedmioty.

## 2.3 Wymagania niefunkcjonalne

Baza danych jest postawiona w chmurze [1]Azure, zaimportowana z [2]Microsoft Access.

Używany standard języka SQL: ANSI-89 Level 1.

**Zalety:**

- Pozwala na definiowanie parametrów przy definiowaniu kwerend,
- Dostarcza dodatkowych funkcji agregujących, jak StDev czy VarP,

**Wady:**

- Brak niektórych funkcji, np. SUM czy LIMIT,
- Składnia niektórych poleceń może się nie co różnić.

**Microsoft Acces** pozwala na:

- Projektowanie aplikacji w języku 4GL,
- Umożliwia automatyzację pracy,
- Posiada filtry do większości użytecznych formatów,
- Wchodzi w skład Microsoft Office.

Z kolei jego **wady:**

- Trudno oddzielić kod aplikacji od danych, brak możliwości kompilacji kodu,
- Ciężko, a czasami jest to nie możliwe, przenieść na inne platformy,

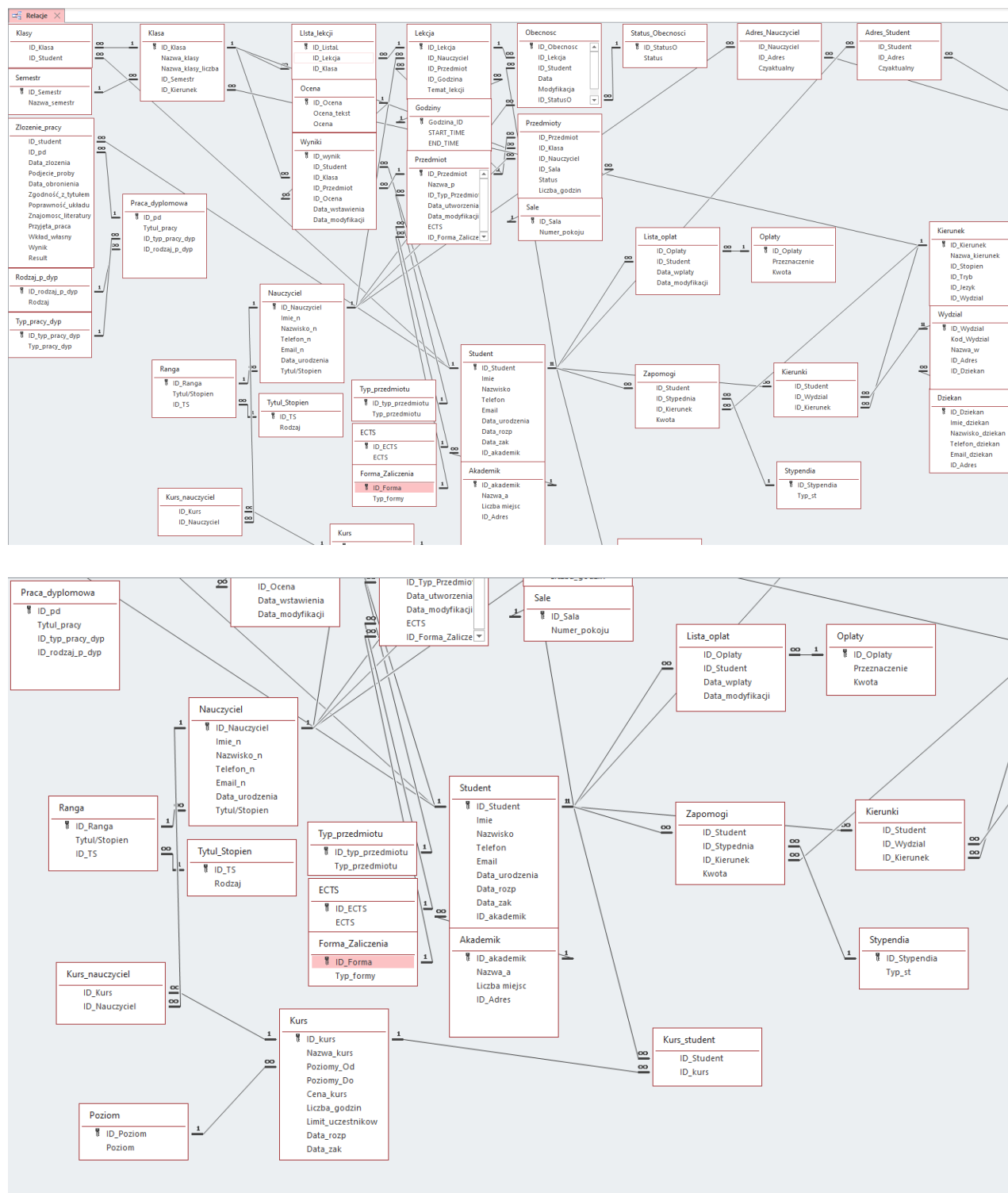
**Microsoft Azure** - zalety:

- Wysoka dostępność i mała awaryjność,
- Dobre zabezpieczenia,
- Wysoka skalowalność,
- Możliwość dostosowywania interfejsu,
- Wysoki stosunek ceny do jakości.

**Microsoft Azure** - minusy:

- Wymaga dużej ilości wiedzy i doświadczenia, by sprawnie się poruszać.

## 2.4 Diagram związków encji





EMAIL	varchar(255)	NO		None	yes	
DATA_URO	date	NO		None		
DATA_ROZP	date	NO		None		
DATA_ZAK	date	YES		None		
ID_AKADEMIK	int	YES	FOR	None		
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
Field	Type	Null	Key	Default	Extra	
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
ID_OBECNOSC	auto_inc	NO	PRI	None		
ID_LEKCJA	int	NO	FOR	None		
ID_STUDENT	int	NO	FOR	None		
DATA	data	NO		None		
MODYFIKACJA	data	NO		None		
ID_STATUSO	int	NO	FOR	None		
ID_NAUCZY	int	NO	FOR	None		
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
Field	Type	Null	Key	Default	Extra	
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
ID_WYDZIAL	auto_inc	NO	PRI	None		
KOD_WY	varchar(255)	NO		None		
NAZWA	varchar(255)	NO		None		
ID_ADRES	int	NO	FOR	None		
ID_DZIEKAN	int	NO	FOR	None		
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+



Field	Type	Null	Key	Default	Extra
ID_PRZEDMI	auto_inc	NO	PRI	None	
NAZWA	varchar(255)	NO		None	
ID_TYP_PRZ	int	NO	FOR	None	
TELEFON	varchar(255)	NO		None	
DATA_UTW	date	NO		None	
DATA_MOD	date	YES		None	
ECTS	int	NO	FOR	None	
ID_FORMA_Z	int	NO	FOR	None	

## 3.2 Przykłady kilku zapytań i ich wyników

Przykłady umieszczamy przy użyciu specjalnych narzędzi do wstawiania kodu, a dokładniej pakietu **lstlisting**.

```
01 | SELECT Student.Imie, Student.Nazwisko, Klasa.Nazwa_klasy, Klasa.  
    | ID_Semestr, Klasa.ID_Kierunek, Przedmiot.Nazwa_p, Przedmiot.  
    | ID_Typ_Przedmiotu  
02 | FROM Student INNER JOIN (Przedmiot INNER JOIN ((Klasa INNER JOIN Klasy  
    | ON Klasa.ID_Klasa = Klasy.ID_Klasa) INNER JOIN Przedmioty ON Klasa.  
    | ID_Klasa = Przedmioty.ID_Klasa) ON Przedmiot.ID_Przedmiot =  
    | Przedmioty.ID_Przedmiot) ON Student.ID_Student = Klasy.ID_Student  
03 | GROUP BY Student.ID_Student, Student.Imie, Student.Nazwisko, Klasa.  
    | ID_Klasa, Klasa.Nazwa_klasy, Klasa.ID_Semestr, Klasa.ID_Kierunek,  
    | Przedmiot.ID_Przedmiot, Przedmiot.Nazwa_p, Przedmiot.  
    | ID_Typ_Przedmiotu;  
04 |  
05 | SELECT Student.Imie, Student.Nazwisko, Klasa.Nazwa_klasy, Klasa.  
    | ID_Semestr, Klasa.ID_Kierunek, Avg(Ocena.Ocena) AS redniaOfOcena  
06 | FROM Student INNER JOIN (Przedmiot INNER JOIN (Ocena INNER JOIN (Klasa  
    | INNER JOIN Wyniki ON Klasa.ID_Klasa = Wyniki.ID_Klasa) ON Ocena.  
    | ID_Ocena = Wyniki.ID_Ocena) ON Przedmiot.ID_Przedmiot = Wyniki.  
    | ID_Przedmiot) ON Student.ID_Student = Wyniki.ID_Student  
07 | GROUP BY Student.ID_Student, Student.Imie, Student.Nazwisko, Klasa.  
    | ID_Klasa, Klasa.Nazwa_klasy, Klasa.ID_Semestr, Klasa.ID_Kierunek;
```

## Literatura

[1] Microsoft, *Azure*, 2017.

[2] ———, *Jezyk access sql: podstawowe pojecia, slownictwo i skladnia*, 2019.