



Uniwersytet Gdański  
Wydział Matematyki, Fizyki i Informatyki  
Instytut Informatyki

# Baza danych realizująca schemat systemu informatycznego obsługującego wirtualną uczelnię

Łukasz Ruzicki

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

Gdynia  
25 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 . . . . .	3
2.2	Wymagania funkcjonalne . . . . .	3
2.3	Wymagania niefunkcjonalne . . . . .	3
2.4	Diagram związków encji . . . . .	4
<b>3</b>	<b>Przykłady realizacji bazy danych</b>	<b>5</b>
3.1	Przykłady zawartości najważniejszych tabel . . . . .	6
3.2	Przykłady kilku zapytań i ich wyników . . . . .	6

# 1 Wprowadzenie

Baza danych została stworzona w celu organizowania danych na temat studentów, pracowników oraz zajęć na wirtualnej uczelni. Pozwala ona na dokładny zapis i odczyt postępu studentów, pracy pracowników oraz dostarcza podstawowe informacje na temat działania uczelni. Baza jest realizowana w relacyjnym modelu w celu zwiększenia elastyczności bazy danych przez wyeliminowanie nadmiarowości i niespójnej zależności oraz ochrony danych. Informacje w niej zawarte znajdują się w tabelach pomiędzy którymi zostały ustanowione relacje.

- Tabela - podstawowy obiekt bazy danych
- Wiersz tabeli nazywany jest krotką lub rekordem
- Pole to pojedyncza komórka tabeli
- Model relacyjny bazy danych - sposób przechowywania danych w wielu odrębnych, ale połączonych ze sobą tabelach
- Normalizacja to proces organizowania danych w bazie danych
- Klucz podstawowy - pojedyncze pole o unikalnej wartości, które identyfikuje dany rekord
- Klucz obcy - wykorzystuje się go do powiązania dwóch tabel Zazwyczaj jest on kluczem podstawowym innej tabeli
- Raport to prezentacja wybranych informacji z bazy danych
- Kolumna zawiera dane jednego określonego typu
- Relacje to zależności między tabkami, umożliwiające powiązanie ze sobą
- System zarządzania bazami danych to program komputerowy, który służy do przechowywania i modyfikowania danych, np. Microsoft Access
- Kwerenda to zapytanie umożliwiające wyświetlenie pól i rekordów tabeli według kryterium ustalonego przez użytkownika

# 2 Opis projektu

Uczelnia Wyższa w Sierakowicach zgłosiła zapotrzebowanie na bazę danych, która pomogłaby uporządkować dane związane z uczelnią. Pracownicy uczelni do tej pory gromadzili wszystkie dane bez użycia komputerów. Wpłynęło to negatywnie na efektywność pracy na uczelni, która zmagą się z ogromem dokumentów w wersji papierowej. Baza danych ma usprawnić uzyskiwanie informacji na temat studentów, ich osiągnięć, ocen oraz podziału na kierunki i grupy. Ponadto umożliwiać gromadzenie informacji o pracownikach i wykonywanych przez nich zajęć. Dodatkowo udzielać informacji na temat akademików i pomocy z której mogą korzystać studenci.

## 2.1 Potencjalne grupy użytkowników

Tu wypisujemy informacje o potencjalnych użytkownikach na przykład:

- Administrator – główny zarządca bazy danych, posiada pełen dostęp do bazy danych
- Pracownik dydaktyczny - posiada ograniczony dostęp do bazy danych, może wprowadzać dane na temat zajęć, które prowadzi oraz zapisywać oceny i obecności studentom znajdującym się w jego grupach.
- Pracownik administracji - posiada dostęp do danych na temat pracowników i studentów, otrzymywanej przez nich pomocy finansowej oraz informacji na temat akademików.
- Student - wgląd do historii swoich płatności, ocen, zajęć, osiągnięć, własnych danych oraz informacji na temat pomocy finansowej, którą może uzyskać i dostępnych miejsc w akademiku.
- Kandydat na studenta - wgląd do dostępnych kierunków i przedmiotów na studiach.

## 2.2 Wymagania funkcjonalne

- Wprowadzanie danych studentów, pracowników oraz informacji na temat uczelni.
- Bieżąca obsługa danych: dodawanie, aktualizacja oraz usuwanie.
- Weryfikacja użytkownika poprzez jego indywidualny numer i ocena jego pracy poprzez dopisywanie ocen.
- Tworzenie raportów na temat zebranych informacji z wybranego zakresu.
- Wyszukiwanie informacji na temat pracowników, studentów, grup dydaktycznych, możliwej pomocy finansowej, akademików i płatności.

Tutaj wypisujemy informacje o wymaganiach funkcjonalnych naszej bazy, a mianowicie jakie dane będzie przechowywać i do jakich zadań będzie użyta (to tylko przykłady można więcej)

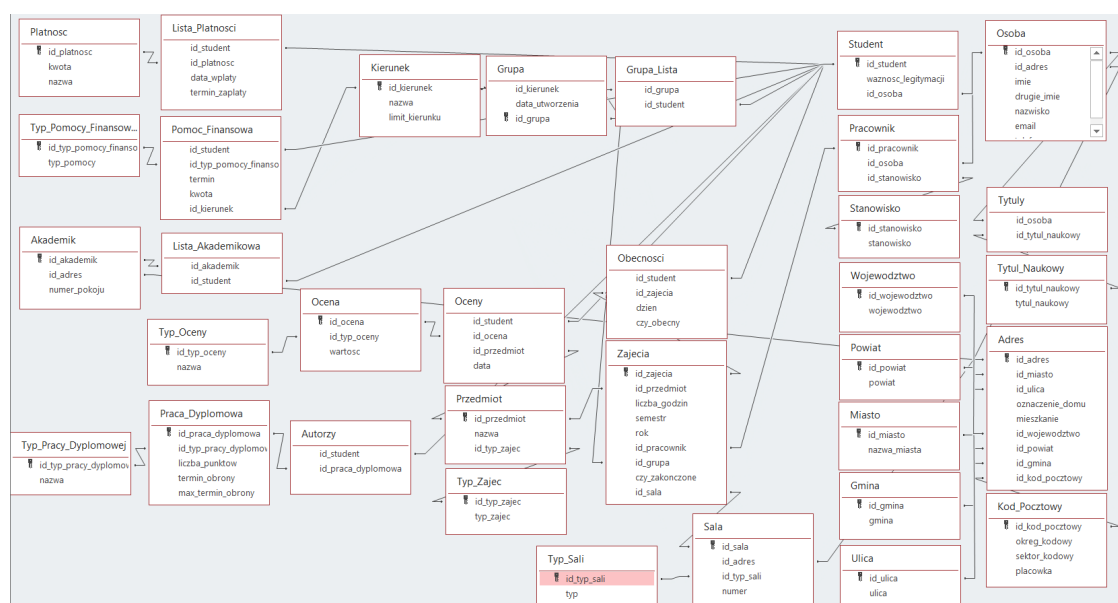
## 2.3 Wymagania niefunkcjonalne

- Użyty został obsługi relacyjnej bazy danych jest Microsoft Access.
- Bazy danych Access zapisywane są w pojedynczych plikach, dlatego jest to wygodne tylko w przypadku prostych zastosowań kosztem wydajności, wielodostępności oraz bezpieczeństwa przechowywanych danych.
- Synergię między Microsoft Accessem a phpMyAdmin zapewnił Connector/NET, który jest sterownikiem MySQL kompatybilnym z jego wszystkimi wersjami.

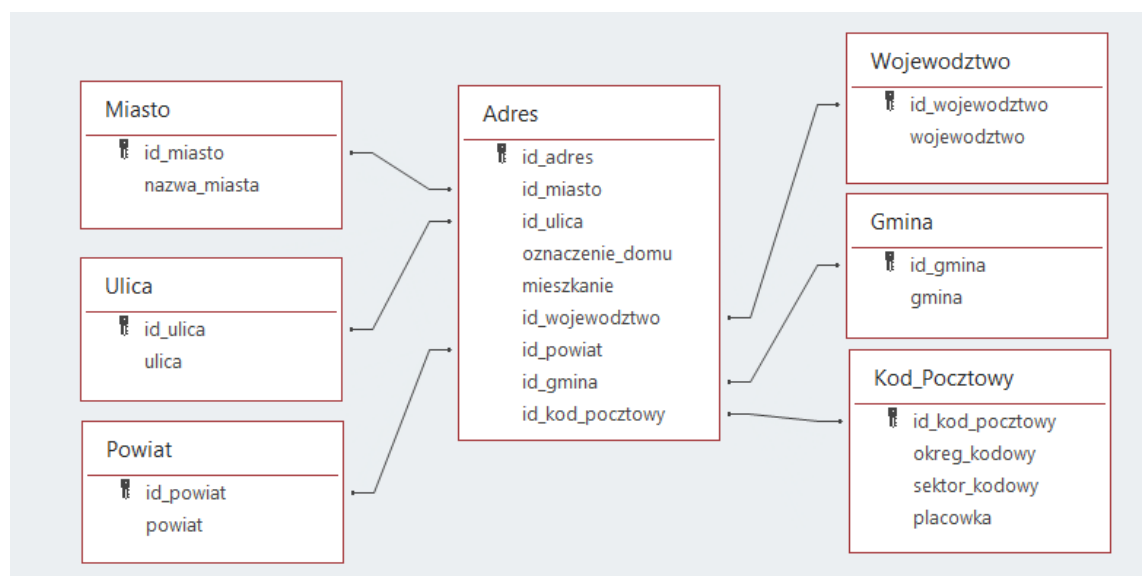
- PhpMyAdmin został głównym narzędziem do zarządzania bazą danych MySQL. Zaletą, prócz łatwości obsługi, jest to, że został wydany na licencji GNU i umożliwia między innymi tworzenie oraz usuwanie baz danych, dodawanie i usuwanie relacji oraz edycję struktur i zawartości.
- Baza danych zrealizowana jest w wolnodostępnym, otwarto źródłowym systemie zarządzania relacyjnymi bazami danych, MySQL 8.0.16.

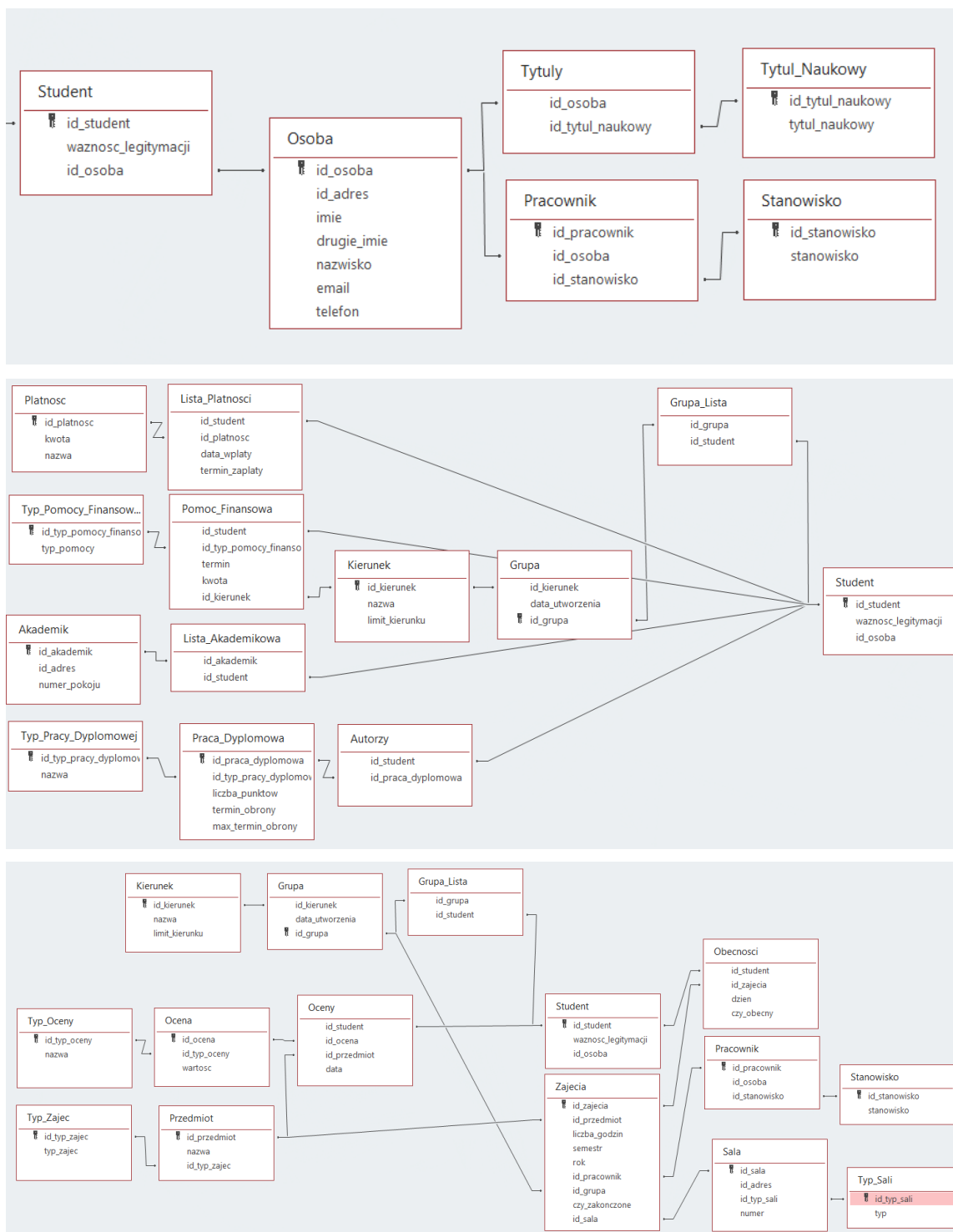
## 2.4 Diagram związków encji

Pełen diagram związków encji (poniżej zdjęcia jest podział na fragmenty):



Podzielony na elementy:





### 3 Przykłady realizacji bazy danych

Tabela Student jest jedną z kluczowych tabel. W jej zawartość wchodzi id-student, który jest numerem legitymacji studenckiej, ważność legitymacji oraz klucz obcy id-osoba, który odnosi się do tabeli zapisującej rekordy z wrażliwymi danymi osób. Zostało to wyodrębnione, aby uniknąć zbędnych powtórzeń, ponieważ te same dane używane są w przypadku tabeli Pracownik. Przykłady kluczowych tabel oraz realizacji zapytań.

### 3.1 Przykłady zawartości najważniejszych tabel

Student

Field	Type	Null	Key	Default	Extra
id_student	integer	NO	PRI	None	
id_osoba	integer	NO		None	
waznosc_legitymacji	date	NO		None	

Osoba

Field	Type	Null	Key	Default	Extra
id_osoba	integer	NO		None	
id_adres	integer	NO		None	
imie	varchar(255)	NO		None	
drugie_imie	varchar(255)	YES		NULL	
nazwisko	varchar(255)	NO		None	
email	varchar(255)	NO		None	
telefon	varchar(255)	NO		None	

### 3.2 Przykłady kilku zapytań i ich wyników

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

```
01 |
02 | SELECT
03 |     S.id_student, O.imie, O.nazwisko, SUM(P.kwota)
04 | FROM osoba      as O
05 |     JOIN Student as S
06 | ON O.id_osoba=S.id_osoba
07 |     JOIN lista_platnosci AS L
08 | ON S.id_student=L.id_student
09 |     JOIN platnosc as P
10 | ON L.id_platnosc=P.id_platnosc
11 |     GROUP BY S.id_osoba;
12 |
13 | +-----+-----+-----+-----+
14 | | id_student | imie      | nazwisko  | Suma(P.kwota) |
15 | +-----+-----+-----+-----+
16 | | 7          | Krzysztof | Krawczyk  | 1000           |
17 | | 38         | Magda     | RuteckaNO | 70             |
18 | | 46         | Wiktoria  | Machulska | 265            |
19 | +-----+-----+-----+-----+
```

```
01 | SELECT
02 |     P.id_praca_dyplomowa,liczba_punktow, COUNT(S.id_student)
03 | FROM   praca_dyplomowa AS P
04 |       JOIN   autorzy as A
05 | ON     P.id_praca_dyplomowa=A.id_praca_dyplomowa
06 |       JOIN   student AS S
07 | ON     A.id_student=S.id_student
08 |       JOIN   osoba AS O
09 | ON     S.id_osoba=O.id_osoba
10 |       GROUP BY P.liczba_punktow DESC LIMIT 3
11 |
12 | +-----+-----+-----+
13 | | id_praca_dyplomowa |   liczba_punktow   | COUNT(S.id_student) |
14 | +-----+-----+-----+
15 | |           4        |           87       |           3          |
16 | |           2        |           81       |           2          |
17 | |           1        |           75       |           2          |
18 | +-----+-----+-----+
```

```
01 | SELECT DISTINCT
02 |     kwota, P.id_platnosc
03 | FROM   lista_platnosci AS L
04 |       JOIN   platnosc AS P
05 | ON     L.id_platnosc=P.id_platnosc
06 |       GROUP BY kwota DESC
07 |
08 |
09 | +-----+-----+
10 | | kwota  | id_platnosc |
11 | +-----+-----+
12 | |   500  |           4  |
13 | |   100  |           1  |
14 | |    35  |           5  |
15 | |    30  |           2  |
16 | +-----+-----+
```

```
01 | SELECT
02 |     UPPER(nazwa),id_przedmiot,typ_zajec
03 | FROM   przedmiot AS P
04 |       JOIN   typ_zajec AS T
05 | ON     P.id_typ_zajec=T.id_typ_zajec
06 |       GROUP BY nazwa LIMIT 4
07 |
08 |
09 |
10 | +-----+-----+-----+
11 | | UPPER(nazwa)      | id_przedmiot      | typ_zajec          |
12 | +-----+-----+-----+
13 | |      ALGEBRA      |          10       | łwykad             |
14 | | ANALIZA MATEMATYCZNA |          4       | łwykad             |
15 | |      ANGIELSKI     |          7       | lektorat           |
16 | |      BAZY DANYCH   |         18       | laboratorium       |
17 | +-----+-----+-----+
```



```
01 | SELECT CONCAT (o.imie," ",o.nazwisko), id_student
02 |       FROM student AS S
03 | JOIN osoba AS O ON S.id_osoba = O.id_osoba
04 |       WHERE O.imie LIKE 'k%'
05 |
06 |
07 |
```

```
08 |
09 |      +-----+-----+
10 |      | CONCAT(O.imie," ",o.nazwisko) | id_student |
11 |      +-----+-----+
12 |      | Krzysztof Krawczyk           | 7         |
13 |      | Kewin ŁWodarczyk            | 30        |
14 |      | Karol Drewno                 | 33        |
15 |      | Kewin Sam                    | 35        |
      +-----+-----+
```