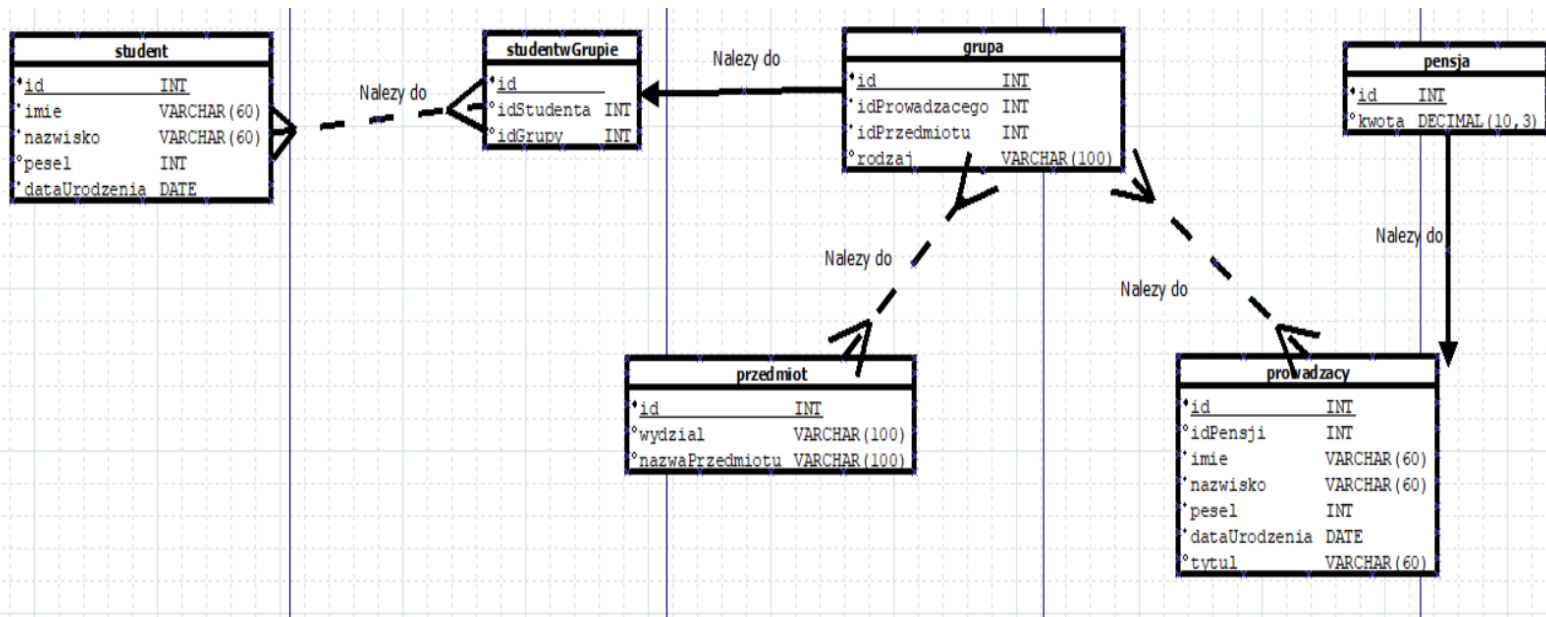


Baza danych by Kacper Wadecki: Grupy studenckie

Diagram związków encji:



Schemat bazy danych:

| studenci | grupy | studenciwGrupach | przedmioty | prowadzacy | pensje |
|--|--|--|--|---|--------------------------------|
| *id INT imie VARCHAR(60) nazwisko VARCHAR(60) pesel INT dataUrodzenia DATE | *id INT idProwadzacego INT idPrzedmiotu INT rodzaj VARCHAR(100) | *id INT idStudenta INT idGrupy INT | *id INT wydzial VARCHAR(100) nazwaPrzedmiotu VARCHAR(100) ECTS DECIMAL(5,2) | *id INT idPensji INT imie VARCHAR(60) nazwisko VARCHAR(60) pesel INT dataUrodzenia DATE tytul VARCHAR(60) | *id INT kwota DECIMAL(10,3) |

Tworzenie tabel:

```
CREATE TABLE studenci(id INT(6) AUTO_INCREMENT PRIMARY KEY, imie VARCHAR(60) NOT NULL, nazwisko VARCHAR(60) NOT NULL, pesel CHAR(11) NOT NULL, dataUrodzenia DATE NOT NULL);
```

```
CREATE TABLE grupy(id INT(6) AUTO_INCREMENT PRIMARY KEY, idProwadzacego INT(6), idPrzedmiotu INT(6), rodzaj VARCHAR(100) NOT NULL);
```

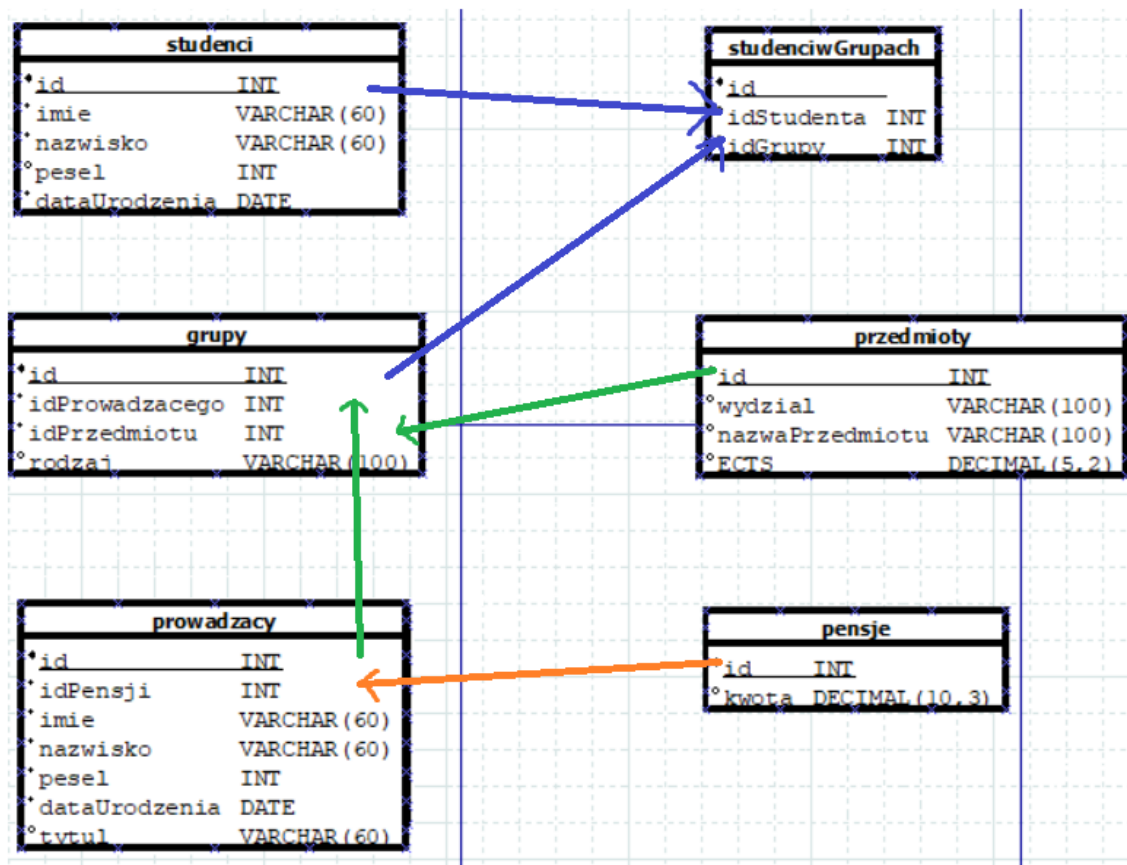
```
CREATE TABLE studenciwGrupach(id INT(6) AUTO_INCREMENT PRIMARY KEY, idStudenta INT(6), idGrupy INT(6));
```

```
CREATE TABLE przedmioty(id INT(6) AUTO_INCREMENT PRIMARY KEY, wydzial VARCHAR(100) NOT NULL, nazwaPrzedmiotu VARCHAR(100) NOT NULL);
```

```
CREATE TABLE prowadzacy(id INT(6) AUTO_INCREMENT PRIMARY KEY, idPensji INT(8), imie VARCHAR(60) NOT NULL, nazwisko VARCHAR(60) NOT NULL, pesel CHAR(11) NOT NULL, dataUrodzenia DATE NOT NULL, tytul VARCHAR(60) NOT NULL);
```

```
CREATE TABLE pensje(id INT(6) AUTO_INCREMENT PRIMARY KEY, kwota DEC(10,3));
```

Klucze obce:



```
ALTER TABLE studenciwGrupach ADD CONSTRAINT  
fk_studenciwGrupach_studenci FOREIGN KEY (idStudenta) REFERENCES  
studenci(id);
```

```
ALTER TABLE studenciwGrupach ADD CONSTRAINT  
fk_studenciwGrupach_grupy FOREIGN KEY (idGrupy) REFERENCES grupy(id);
```

```
ALTER TABLE grupy ADD CONSTRAINT fk_grupy_przedmioty FOREIGN KEY  
(idPrzedmiotu) REFERENCES przedmioty(id);
```

```
ALTER TABLE grupy ADD CONSTRAINT fk_grupy_przewadzacy FOREIGN KEY  
(idProwadzacego) REFERENCES przewadzacy(id);
```

```
ALTER TABLE przewadzacy ADD CONSTRAINT fk_przewadzacy_pensje FOREIGN  
KEY (idPensji) REFERENCES pensje(id);
```

Wstawianie wierszy:

```
INSERT INTO studenci(imie,nazwisko,pesel,dataUrodzenia)  
VALUES('imie','nazwisko',pesel,'dataUrodzenia');
```

```
INSERT INTO pensje(kwota) VALUES(kwota);
```

```
INSERT INTO prowadzacy(imie,nazwisko,pesel,dataUrodzenia,idPensji,tytul)  
VALUES('imie','nazwisko',pesel,'dataUrodzenia',idPensji,'tytul');
```

```
INSERT INTO przedmioty(wydzial,nazwaPrzedmiotu)  
VALUES('wydzial','nazwaPrzedmiotu');
```

```
INSERT INTO grupy (idProwadzacego,idPrzedmiotu,rodzaj)  
VALUES(idProwadzacego,idPrzedmiotu,'rodzaj');
```

```
INSERT INTO studenciwGrupach(idStudenta,idGrupy)  
VALUES(idStudenta,idGrupy);
```

Funkcjonalności:

Dodawanie grup:

```
INSERT INTO grupy (idProwadzacego,idPrzedmiotu,rodzaj)  
VALUES(idProwadzacego,idPrzedmiotu,'rodzaj');
```

Dodawanie studentów:

```
INSERT INTO studenci(imie,nazwisko,pesel,dataUrodzenia)  
VALUES('imie','nazwisko',pesel,'dataUrodzenia');
```

Dodawanie studentów do grup:

```
INSERT INTO studenciwGrupach(idStudenta,idGrupy)  
VALUES(idStudenta,idGrupy);
```

Sprawdzanie wszystkich studentów we wszystkich grupach:

```
SELECT * FROM KW1;
```

```
CREATE VIEW KW1 AS SELECT studenci.imie, studenci.nazwisko,  
nazwaPrzedmiotu,rodzaj FROM studenci INNER JOIN studenciwGrupach ON  
studenci.id=studenciwGrupach.idStudenta INNER JOIN grupy ON  
studenciwGrupach.idGrupy=grupy.id INNER JOIN przedmioty ON  
grupy.idPrzedmiotu=przedmioty.id ORDER BY nazwaPrzedmiotu;
```

Sprawdzanie, którzy studenci są w danej grupie:

SELECT * FROM KW2;

CREATE VIEW **KW2** AS SELECT studenci.imie, studenci.nazwisko, nazwaPrzedmiotu, rodzaj FROM studenci INNER JOIN studenciwGrupach ON studenci.id=studenciwGrupach.idStudenta INNER JOIN grupy ON studenciwGrupach.idGrupy=grupy.id INNER JOIN przedmioty ON grupy.idPrzedmiotu=przedmioty.id WHERE idGrupy=wybranaWartosc;

Sprawdzanie w jakich grupach jest dany student:

SELECT * FROM KW3;

CREATE VIEW **KW3** AS SELECT studenci.imie, studenci.nazwisko, nazwaPrzedmiotu, rodzaj FROM studenci INNER JOIN studenciwGrupach ON studenci.id=studenciwGrupach.idStudenta INNER JOIN grupy ON studenciwGrupach.idGrupy=grupy.id INNER JOIN przedmioty ON grupy.idPrzedmiotu=przedmioty.id WHERE idStudenta=wybranaWartosc;

Dane wszystkich studentów:

SELECT * FROM KW4;

CREATE VIEW **KW4** AS SELECT * FROM studenci;

Dane wszystkich grup:

SELECT * FROM KW5;

CREATE VIEW **KW5** AS SELECT DISTINCT grupy.id, nazwaPrzedmiotu, rodzaj, CONCAT(prowadzacy.imie, ' ', prowadzacy.nazwisko) AS prowadzacy FROM studenci INNER JOIN studenciwGrupach ON studenci.id=studenciwGrupach.idStudenta INNER JOIN grupy ON studenciwGrupach.idGrupy=grupy.id INNER JOIN przedmioty ON grupy.idPrzedmiotu=przedmioty.id INNER JOIN prowadzacy ON grupy.idProwadzacego=prowadzacy.id;

Liczebność grup:

SELECT * FROM KW6;

CREATE VIEW **KW6** AS SELECT nazwaPrzedmiotu, rodzaj, COUNT(studenci.id) AS liczebnosc FROM studenci INNER JOIN studenciwGrupach ON studenci.id=studenciwGrupach.idStudenta INNER JOIN grupy ON studenciwGrupach.idGrupy=grupy.id INNER JOIN przedmioty ON grupy.idPrzedmiotu=przedmioty.id GROUP BY grupy.id;

Prowadzący, którzy zarabiają więcej niż 3000 złotych:

SELECT * FROM KW7;

CREATE VIEW KW7 AS SELECT imie, nazwisko, kwota FROM prowadzacy INNER JOIN pensje ON prowadzacy.idPensji=pensje.id WHERE kwota>3000;

Prowadzący, którzy prowadzi zajęcia na wybranym wydziale:

SELECT * FROM KW8;

CREATE VIEW KW8 AS SELECT DISTINCT imie, nazwisko, wydzial FROM prowadzacy INNER JOIN grupy ON prowadzacy.id=grupy.idProwadzacego INNER JOIN przedmioty ON grupy.idPrzedmiotu=przedmioty.id WHERE wydzial='wybranyWydzial';

Studenci, którzy uczęszczają na zajęcia z danym prowadzącym:

SELECT * FROM KW9;

CREATE VIEW KW9 AS SELECT CONCAT(studenci.imie, ' ', studenci.nazwisko) AS Student, CONCAT(prowadzacy.imie, prowadzacy.nazwisko) AS prowadzacy FROM studenci INNER JOIN studenciwGrupach ON studenci.id=studenciwGrupach.idStudenta INNER JOIN grupy ON studenciwGrupach.idGrupy=grupy.id INNER JOIN prowadzacy ON grupy.idProwadzacego=prowadzacy.id WHERE prowadzacy.id=2;

Studenci, którzy uczęszczają na ćwiczenia:

SELECT * FROM KW10;

CREATE VIEW KW10 AS SELECT DISTINCT imie, nazwisko, rodzaj FROM studenci INNER JOIN studenciwGrupach ON studenci.id=studenciwGrupach.idStudenta INNER JOIN grupy ON studenciwGrupach.idStudenta=grupy.id WHERE rodzaj='cwiczenia';