

# Podstawy Baz Danych Projekt

*Marcin Wójcik Informatyka Stosowana 18/19, grupa 4*

Baza danych szkoły – MySQL

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

## Spis treści dokumentacji

1.	Diagramy .....	5
1.1.	Notacja Chena.....	5
1.2.	Notacja Barkera .....	6
1.3.	Notacja UML .....	7
1.4.	Model Relacyjny.....	8
2.	Założenia bazy.....	9
2.1.	Encja uczniowie.....	9
2.2.	Encja opłaty.....	9
2.3.	Encje zwykły_uczeń/przewodniczący/skarbnik.....	9
2.4.	Encja dziennik_ocen .....	9
2.5.	Encja Klasa.....	9
2.6.	Encja nauczyciel .....	10
2.7.	Encja wychowawca.....	10
2.8.	Encja pracownicy .....	10
2.9.	Encja zastępstwa.....	11
2.10.	Encja kadra.....	11
2.11.	Encja przedmiot.....	11
2.12.	Encja zajęcia.....	11
2.13.	Encja sala.....	12
3.	Procedury Składowane i Wyzwalacze.....	12
3.1.	Funkcja obliczająca dodatek_wychowawczy w zależności od wychowanych klas. Dwa triggerzy wywołują ją po UPDATE oraz po INSERT.....	12
3.2.	Procedury ustalają etat pracownika na podstawie wpisanej pensji. ....	13
3.3.	Gdy „chorobowe” w tabeli nauczyciel zostanie zmienione z 0 na 1, to id_nauczyciela i id_przedmiotu, którego naucza zostaje dodane do tabeli zastępstwa. ....	13
4.	Zapytania.....	14
4.1.	Spotkanie z rodzicami.....	14
4.1.1.	Lista uczniów wraz z wychowawcami (INNER JOIN) .....	14
4.1.1.	Lista dostępnych wychowawców .....	15
4.1.2.	Sale wychowawców (ANY) .....	15
4.1.3.	Uczniowie bez ocen (NOT EXISTS) .....	15
4.1.4.	Średnie uczniów z każdego przedmiotu (GROUP BY) .....	16
4.1.5.	Najlepsi uczniowie (HAVING, ANY) .....	18
4.1.6.	Uczniowie bez żadnej opłaty (NOT IN) .....	18

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

---

4.1.7.	Uczniowie bez opłaconego ubezpieczenia (ANY) .....	19
4.1.8.	Łączna suma opłat (SKORELOWANE) .....	19
4.1.9.	Opłaty za kino (INNER JOIN, GROUP BY) .....	20
4.1.10.	Skarbnicy (INNER JOIN, SORTOWANIE) .....	20
4.1.11.	Ile jest dziewczynek, a ilu chłopców (GROUP BY) .....	20
4.1.12.	Poszukiwania wandalą (LIKE) .....	21
4.2.	Kontrola z kuratorium.....	21
4.2.1.	Pensje nauczycieli (OUTER JOIN) .....	21
4.2.2.	Pensje nauczycieli wyższe niż 2700 zł (HAVING, GROUP BY, LIKE) .....	22
4.2.3.	Pokrewieństwa wśród nauczycieli (HAVING, GROUP BY).....	22
4.2.4.	Nepotyzm w szkole (NIESKORELOWANE, IN).....	22
4.2.5.	Średnie oceny z przedmiotów (INNER JOIN, GROUP BY) .....	22
4.2.6.	Urodzenia w miesiącu (REGEX) .....	23
5.	Dane w tabelach .....	24
5.1.	Pracownicy .....	24
5.2.	Nauczyciele .....	24
5.3.	Wychowawca .....	25
5.4.	Klasa.....	25
5.5.	Uczniowie .....	25
5.6.	Przewodniczący.....	25
5.7.	Skarbnik.....	25
5.8.	Zwykły uczeń .....	26
5.9.	Przedmiot .....	26
5.10.	Opłaty.....	26
5.11.	Sala .....	27
5.12.	Kadra .....	27
5.13.	Zajęcia .....	27
5.14.	Dziennik Ocen .....	28
5.15.	Zajęcia .....	28
6.	Użyte komendy .....	29
6.1.	Tabele.....	29
6.1.1.	Pracownicy.....	29
6.1.2.	Nauczyciele .....	29
6.1.3.	Wychowawca.....	29
6.1.4.	Klasa .....	29

---

# Dokumentacja

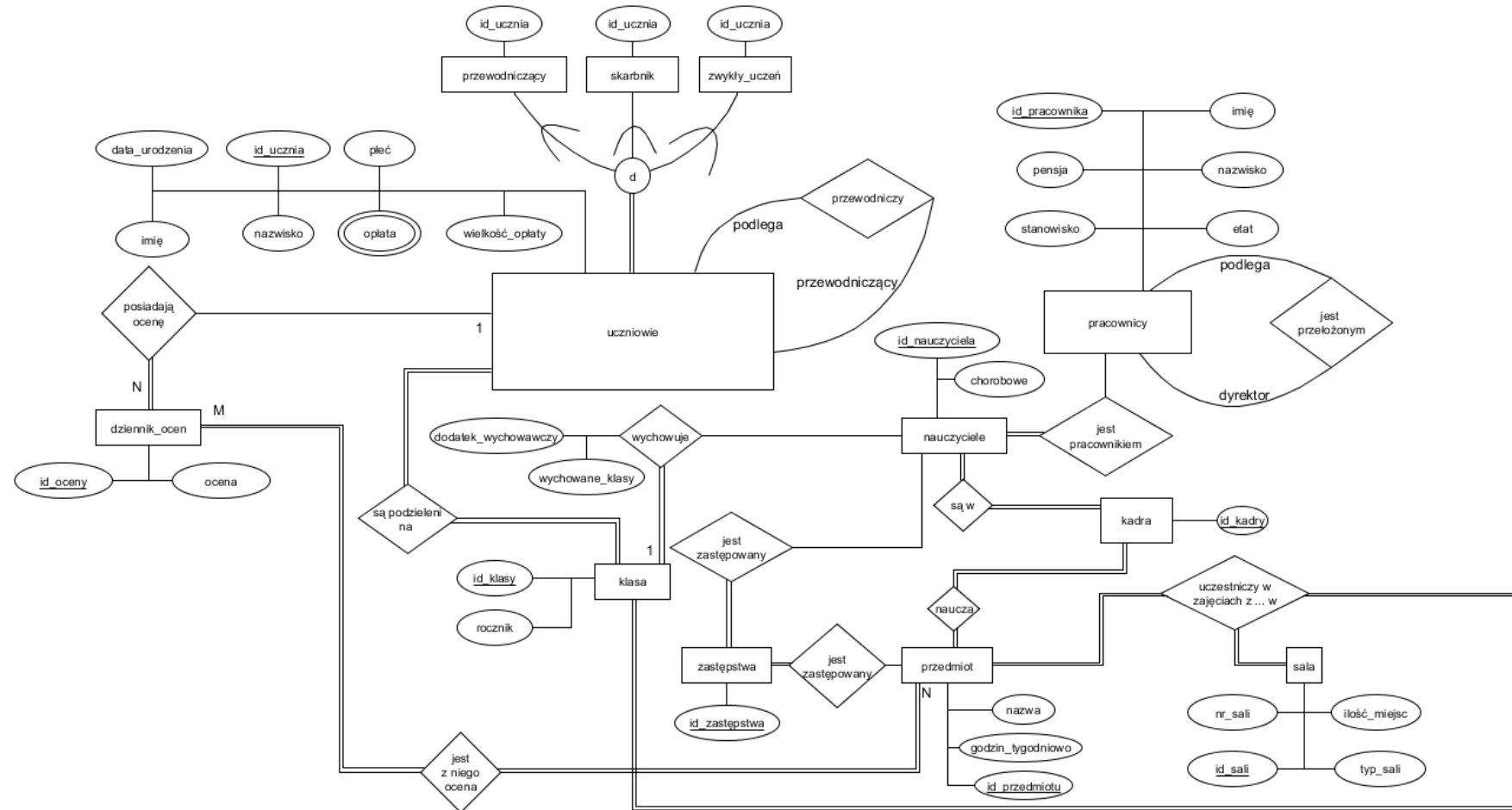
Marcin Wójcik Informatyka Stosowana 18/19

---

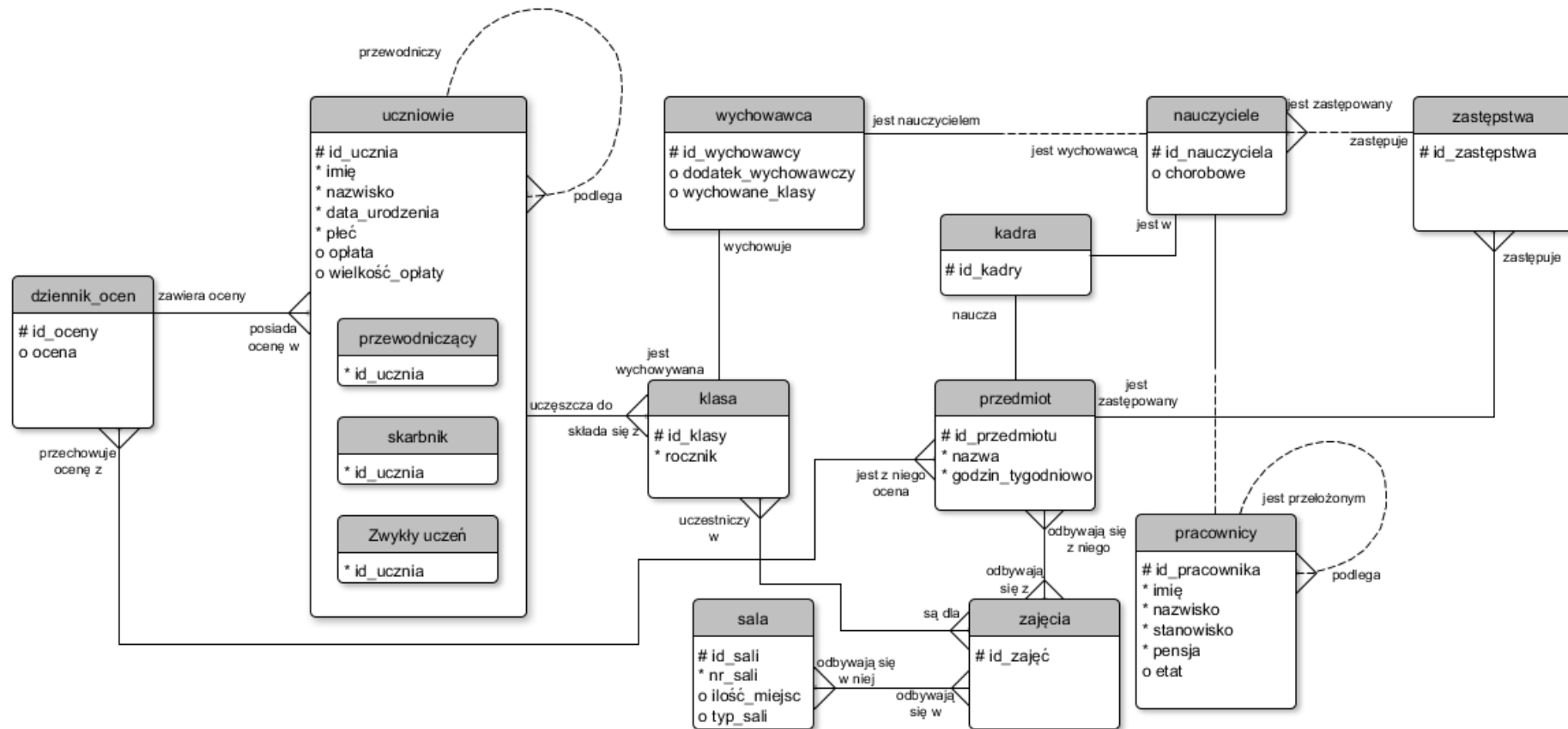
6.1.5.	Uczniowie.....	30
6.1.6.	Opłaty.....	30
6.1.7.	Zwykły_uczeń.....	30
6.1.8.	Przewodniczący .....	31
6.1.9.	Skarbnik.....	31
6.1.10.	Przedmiot.....	31
6.1.11.	Dziennik_ocen .....	31
6.1.12.	Zastępstwa.....	32
6.1.13.	Kadra .....	32
6.1.14.	Sala .....	32
6.1.15.	Zajęcia .....	33
6.2.	Insert .....	33
6.2.1.	Pracownicy.....	33
6.2.2.	Nauczyciele .....	34
6.2.3.	Wychowawca.....	34
6.2.4.	Klasa .....	34
6.2.5.	Uczniowie.....	34
6.2.6.	Zwykły_uczeń.....	35
6.2.7.	Skarbnik.....	35
6.2.8.	Przewodniczący .....	35
6.2.9.	Przedmiot.....	35
6.2.10.	Opłaty.....	35
6.2.11.	Sala .....	36
6.2.12.	Kadra .....	36
6.2.13.	Zajęcia .....	36
6.2.14.	Ocena .....	36
6.3.	Update.....	37
6.4.	Delete .....	37
6.5.	Select.....	37
6.6.	Inne.....	38
7.	Źródła .....	38

## 1. Diagramy

### 1.1. Notacja Chena

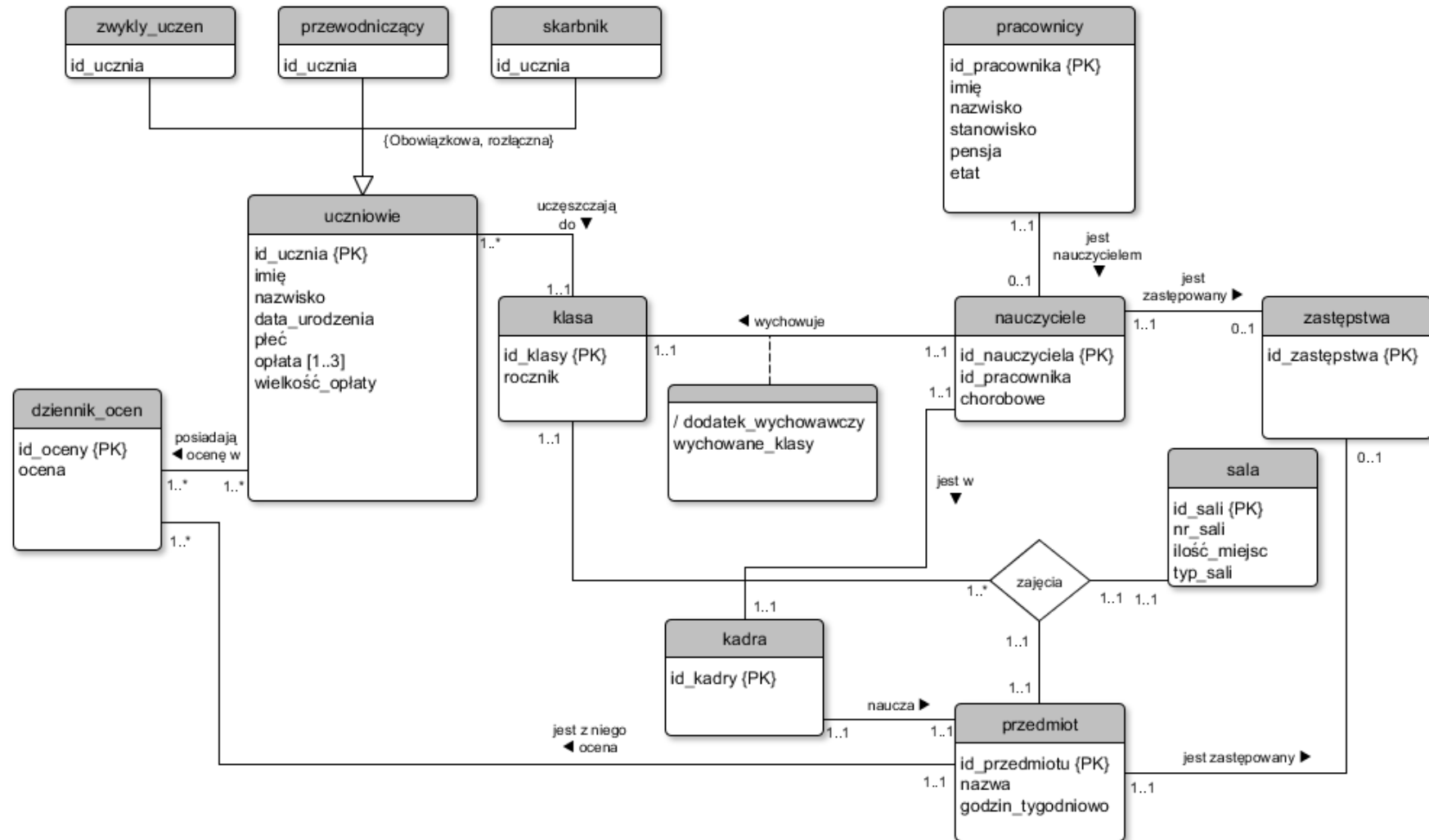


## 1.2. Notacja Barkera

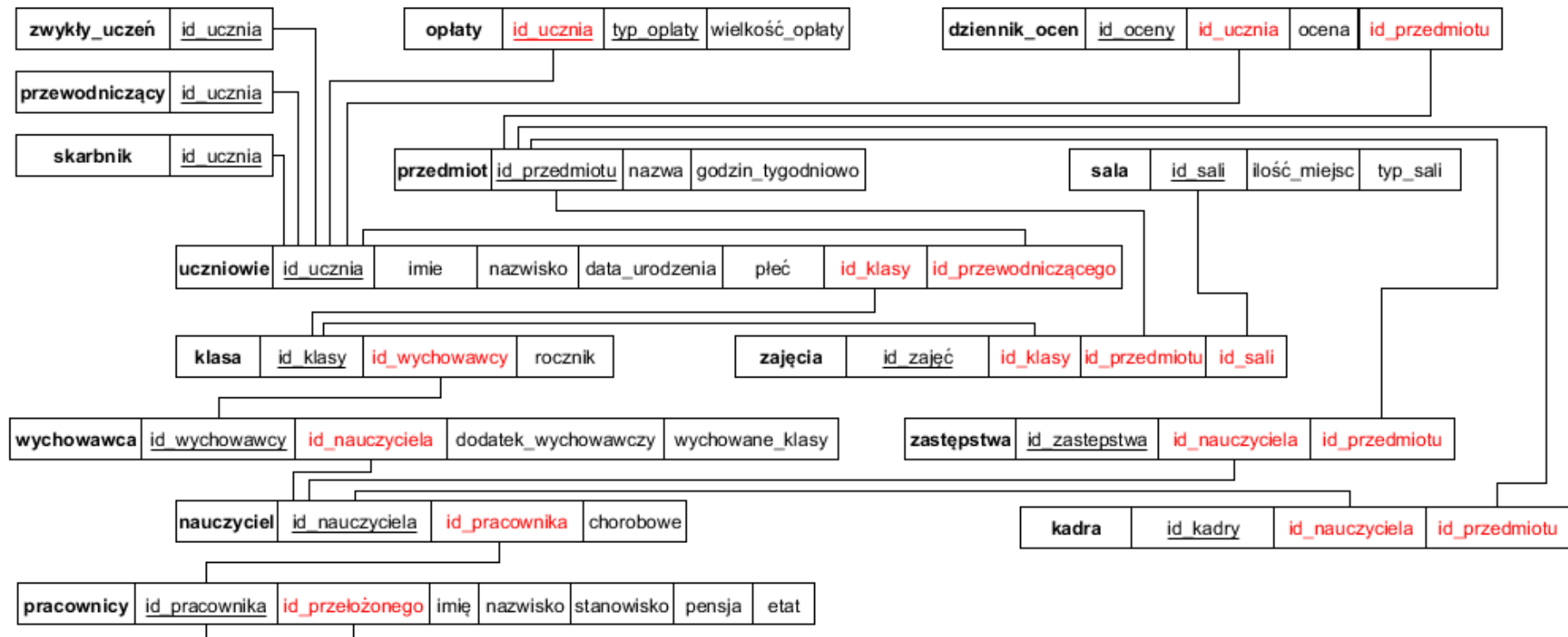


# Dokumentacja

### 1.3. Notacja UML



## 1.4. Model Relacyjny



- ❖ Atrybuty ze związku encji nauczyciel z atrybutami zostały zamienione na tabelę wychowawca.
- ❖ Relacja „zajęcia” ze związku ternarnego encji: klasa, sala, przedmiot, została zamieniona na encję zajęcia.
- ❖ Atrybut wielowartościowy „opłata” w encji „uczniowie” został zamieniony na encję „opłata”, atrybut „wielkość\_opłaty” została przeniesiona do encji „opłata”.
- ❖ Do encji, które są w związkach zostały dodane klucze obce.



## 2. Założenia bazy

### 2.1. Encja uczniowie

- **id\_ucznia** – klucz główny tabeli, unikalny dla każdego ucznia.
- **imię** – imię ucznia.
- **nazwisko** – nazwisko ucznia.
- **data\_urodzenia** – data urodzenia ucznia.
- **płeć** – płeć ucznia.
- **id\_klasy** – klucz obcy do tabeli klasa, przypisuje należność ucznia do danej klasy.
- **id\_przewodniczącego** – klucz obcy do tabeli uczniowie, związek unarny.
- **opłata** – atrybut wielowartościowy
- **wielkość\_opłaty** – atrybut określający wielkość opłaty uiszczonej przez ucznia w zależności od typu\_opłaty (atrybut wielowartościowy opłata).

- ❖ Wiele uczniów może należeć do jednej klasy.
- ❖ Każdy uczeń podlega pod przewodniczącego (klasy). Tylko przewodniczący nie podlega sam pod siebie.

### 2.2. Encja opłaty

*(atrybut wielowartościowy zamieniony na tabelę)*

- **id\_ucznia** – część klucza głównego, klucz obcy do tabeli uczniowie
- **typ\_opłaty** – część klucza głównego, wartość informująca za co uiszczona była opłata, np. ubezpieczenie.
- **wielkość\_opłaty** – wartość opłaty określana przez typ opłaty.

Każdy uczeń może opłacić wiele opłat.

Opłaty nie są obowiązkowe dla każdego ucznia.

Każdy typ opłaty ma inną wielkość opłaty.

### 2.3. Encje zwykły\_uczeń/przewodniczący/skarbnik

- **id\_ucznia** – klucz obcy do tabeli uczniowie.
- ❖ Jest to hierarchia encji określająca jakie role w klasie posiadają uczniowie.
- ❖ Każdy uczeń może posiadać tylko jedną „rolę”.
- ❖ Każdy uczeń musi posiadać rolę.

### 2.4. Encja dziennik\_ocen

- **id\_oceny** – klucz główny tabeli, unikalny dla każdej oceny.
- **id\_ucznia** – klucz obcy do tabeli uczniowie
- **id\_przedmiotu** – klucz obcy do tabeli przedmiot
- **ocena** – ocena otrzymana przez ucznia z danego przedmiotu
- ❖ Każdy uczeń może posiadać wiele ocen z jednego przedmiotu.

### 2.5. Encja Klasa

- **id\_klasy** – klucz główny tabeli, unikalny dla każdej klasy.

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

---

- **id\_wychowawcy** – klucz obcy do tabeli wychowawca.
  - **rocznik** – wartość określająca, która jest to rocznikowo klasa np. 1, 4, 7.
- ❖ Każda klasa może być wychowywana tylko przez jednego wychowawcę.
- ❖ Każda klasa musi mieć oznaczenie rocznika, wpływa on na przedmioty, których uczą się uczniowie.

id	Nazwa przedmiotu	Rocznik
1	Język Polski	1-2-3
2	Język Angielski	2-3
3	Język Niemiecki	3
4	Matematyka	1-2-3
5	Przyroda	1-2
6	Technika	2-3
7	WF	1-2-3
8	Godzina wychowawcza	1-2-3
9	Historia	1-2-3
10	Geografia	2-3
11	Plastyka	1
12	Muzyka	2-3
13	WOS	2-3
14	WDŻ	3
15	Informatyka	1-2-3

## 2.6. Encja nauczyciel

- **id\_nauczyciela** – klucz główny tabeli, unikalny dla każdego nauczyciela.
  - **id\_pracownika** – klucz obcy do tabeli pracownicy.
  - **chorobowe** – wartość opcjonalna.
- ❖ Jeżeli nauczyciel jest na chorobowym, to pojawia się w tabeli zastępstwa.
- ❖ Chorobowe jest wartością opcjonalną.
- ❖ Jeden nauczyciel jest tylko jednym pracownikiem.

## 2.7. Encja wychowawca

- **id\_wychowawcy** – klucz główny tabeli, unikalny dla każdego wychowawcy.
  - **id\_nauczyciela** – klucz obcy do tabeli nauczyciel.
  - **dodatek\_wychowawczy** – wartość informująca jaki dodatek do pensji posiada dany wychowawca.
  - **wychowane\_klasy** – liczba klas, które wychowywał dany wychowawca, im więcej klas, tym większy dodatek wychowawczy.
- ❖ Każdy nauczyciel może być tylko jednym wychowawcą.
- ❖ Każdy wychowawca wychowuje jedynie jedną klasę.
- ❖ Nauczyciel nie musi być wychowawcą, ale wychowawca musi być nauczycielem.
- ❖ Atrybut *dodatek\_wychowawczy* jest obliczany na podstawie ilości wychowanych klas, za każdą klasę należy się 100 zł dodatku.

## 2.8. Encja pracownicy

- **id\_pracownika** – klucz główny tabeli, unikalny dla każdego pracownika.

- **id\_przełożonego** – klucz obcy do tabeli pracownicy, związek unarny.
- **imię** – imię pracownika.
- **nazwisko** – nazwisko pracownika.
- **stanowisko** – stanowisko, jakie zajmuje pracownik, np. nauczyciel, sprzątac.
- **pensja** – wartość informująca, ile zarabia dany pracownik.
- **etat** – wartość informująca, na jakim etacie pracuje pracownik.
  
- ❖ Każdy pracownik (poza dyrektorem) ma swojego przełożonego.
- ❖ Jeżeli pracownik posiada stanowisko nauczyciel, to tylko jeden pracownik może być jednym nauczycielem i może uczyć tylko jednego przedmiotu.

## 2.9. Encja zastępstwa

- **id\_zastępstwa** – klucz główny tabeli, unikalny dla każdego zastępstwa.
- **id\_nauczyciela** – klucz obcy do tabeli nauczyciel.
- **id\_przedmiotu** – klucz obcy do tabeli przedmiot.
  
- ❖ Zastępstwo pojawia się, jeżeli nauczyciel jest na chorobowym.
- ❖ Każdemu zastępowanemu nauczycielowi odpowiada tylko jedno zastępstwo.

## 2.10. Encja kadra

- **id\_kadry** – klucz główny tabeli, unikalny dla każdego kadrowicza.
- **id\_nauczyciela** – klucz obcy do tabeli nauczyciele.
- **id\_przedmiotu** – klucz obcy do tabeli przedmiot.
  
- ❖ Encja umożliwia typ asocjacji wiele do wielu.
- ❖ Jeden przedmiot może być nauczany przez wielu nauczycieli, ale jeden nauczyciel może nauczać tylko jednego przedmiotu.

## 2.11. Encja przedmiot

- **id\_przedmiotu** – klucz główny tabeli, unikalny dla każdego przedmiotu.
- **nazwa** – nazwa przedmiotu.
- **godzin\_tygodniowo** – wartość określająca, ile godzin w tygodniu nauczany jest dany przedmiot.
  
- ❖ Przedmiot musi być nauczany przez przynajmniej jednego nauczyciela.
- ❖ Jeden nauczyciel może nauczać tylko jednego przedmiotu.

## 2.12. Encja zajęcia

- **id\_zajęc** – klucz główny tabeli, unikalny dla każdego zajęć.
- **id\_sali** – klucz obcy do tabeli sala.
- **id\_przedmiotu** – klucz obcy do tabeli przedmiot.
- **id\_klasy** – klucz obcy do tabeli klasa.
  
- ❖ Powstała ze związku ternarnego encji: sala, przedmiot, klasa.
- ❖ Z każdego przedmiotu mogą być prowadzone tylko jedno zajęcia.
- ❖ W każdej sali mogą być prowadzone tylko jedno zajęcia w danym momencie.
- ❖ Dla każdej klasy może odbywać się wiele zajęć.

## 2.13. Encja sala

- **id\_sali** – klucz główny tabeli, unikalny dla każdej sali.
  - **nr\_sali** – wartość informująca o numerze sali.
  - **ilość\_miejsc** – wartość informująca o ilości miejsc w sali.
  - **typ\_sali** – wartość informująca o tym, jakiego typu jest sala, co przybliży jej przeznaczenie w kwestii nauczania w niej danego przedmiotu.
- ❖ W jednej sali mogą odbywać się tylko jedno zajęcia.
- ❖ Dana klasa może mieć zajęcia w sali tylko wtedy, jeżeli starczy w niej miejsc.

## 3. Procedury Składowane i Wyzwalacze

---

3.1. Funkcja obliczająca dodatek\_wychowawczy w zależności od wychowanych klas. Dwa triggery wywołują ją po UPDATE oraz po INSERT.

---

```
DROP FUNCTION IF EXISTS dodatek_f;
DELIMITER $$
    CREATE FUNCTION dodatek_f(wychowane_klasy INT) RETURNS INT(11)
    BEGIN
        DECLARE dodatek INT;
        IF (wychowane_klasy = 1) THEN
            SET dodatek = 100;
        ELSEIF (wychowane_klasy = 2) THEN
            SET dodatek = 200;
        ELSEIF (wychowane_klasy = 3) THEN
            SET dodatek = 300;
        ELSEIF (wychowane_klasy = 4) THEN
            SET dodatek = 400;
        ELSEIF (wychowane_klasy >= 5) THEN
            SET dodatek = 500;
        END IF;
        RETURN dodatek;
    END $$
DELIMITER ;

DROP TRIGGER IF EXISTS dodatek_u;
DELIMITER $$
    CREATE TRIGGER dodatek_u
    BEFORE UPDATE ON wychowawca
    FOR EACH ROW BEGIN
        SET NEW.dodatek_wychowawczy = (SELECT dodatek_f(NEW.wychowane_klasy));
    END;
$$
DELIMITER ;

DROP TRIGGER IF EXISTS dodatek_i;
DELIMITER $$
    CREATE TRIGGER dodatek_i
    BEFORE INSERT ON wychowawca
    FOR EACH ROW BEGIN
        SET NEW.dodatek_wychowawczy = (SELECT dodatek_f(NEW.wychowane_klasy));
    END;
$$
DELIMITER ;
```

---

## 3.2. Procedury ustalają etat pracownika na podstawie wpisanej pensji.

---

```
DROP PROCEDURE IF EXISTS szkola.etat_pol;
DELIMITER $$
CREATE PROCEDURE etat_pol()
    UPDATE pracownicy
    SET etat = "pół"
    WHERE pensja < 1700

$$
DELIMITER ;

CALL etat_pol;

DROP PROCEDURE IF EXISTS szkola.etat_pelen;
DELIMITER $$
CREATE PROCEDURE etat_pelen()
    UPDATE pracownicy
    SET etat = "pełen"
    WHERE pensja >= 1700
$$
DELIMITER ;

CALL etat_pelen;
```

---

## 3.3. Gdy „chorobowe” w tabeli nauczyciel zostanie zmienione z 0 na 1, to id\_nauczyciela i id\_przedmiotu, którego naucza zostaje dodane do tabeli zastępstwa.

---

```
DROP TRIGGER IF EXISTS zastepstwo;
DELIMITER $$
CREATE TRIGGER zastepstwo
    AFTER UPDATE ON nauczyciele
    FOR EACH ROW BEGIN
    IF (NEW.chorobowe = 1) THEN
        INSERT INTO szkola.zastepstwa
            (id_zastepstwa, id_nauczyciela, id_przedmiotu)
        VALUES
            (default, OLD.id_nauczyciela, NULL);
    END IF;
END;
$$
DELIMITER ;

DELIMITER $$
CREATE PROCEDURE zastepstwa_p()

UPDATE szkola.zastepstwa
SET id_przedmiotu = (SELECT przedmiot.id_przedmiotu FROM przedmiot
    JOIN kadra ON kadra.id_przedmiotu = przedmiot.id_przedmiotu
    WHERE kadra.id_nauczyciela = zastepstwa.id_nauczyciela)

$$
DELIMITER ;

DROP TRIGGER IF EXISTS zastepstwo_p;
DELIMITER $$
CREATE TRIGGER zastepstwo_p
```

```
AFTER UPDATE ON nauczyciele
FOR EACH ROW BEGIN
    CALL zastepstwa_p();
END;
$$
DELIMITER ;
```

## 4. Zapytania

### 4.1. Spotkanie z rodzicami

Z okazji zakończenia pierwszego semestru, w szkole organizowane jest spotkanie z rodzicami, aby omówić kwestie ocen, opłat, uwag nauczycieli oraz niektóre sprawy organizacyjne. Z tej okazji należy najpierw przygotować odpowiednie dane z bazy danych szkoły.

#### 4.1.1. Lista uczniów wraz z wychowawcami (INNER JOIN)

Stwórz listę, która będzie zawierała imię, nazwisko i numer klasy ucznia oraz imię i nazwisko wychowawcy danej klasy. Listę posortuj alfabetycznie według nazwisk uczniów.

```
SELECT uczniowie.nazwisko AS 'nazwisko ucznia', uczniowie.imię AS 'imię ucznia',
klasa.id_klasy AS 'numer klasy', pracownicy.imię AS 'imię wychowawcy', pracownicy.nazwisko
AS 'nazwisko wychowawcy'
FROM pracownicy
INNER JOIN nauczyciele ON pracownicy.id_pracownika = nauczyciele.id_pracownika
JOIN wychowawca ON nauczyciele.id_nauczyciela = wychowawca.id_nauczyciela
JOIN klasa ON wychowawca.id_wychowawcy = klasa.id_wychowawcy
JOIN uczniowie ON klasa.id_klasy = uczniowie.id_klasy
ORDER BY uczniowie.id_klasy, uczniowie.nazwisko ASC
```

nazwisko ucznia	imię ucznia	numer klasy	imię wychowawcy	nazwisko wychowawcy
Daniel	Eryk	1	Juliusz	Mogiła
Garlicki	Paweł	1	Juliusz	Mogiła
Ibisz	Joanna	1	Juliusz	Mogiła
Kopytko	Zuzanna	1	Juliusz	Mogiła
Kowalski	Brajan	1	Juliusz	Mogiła
Majeranek	Marika	1	Juliusz	Mogiła
Sanitariusz	Mariusz	1	Juliusz	Mogiła
Stróżewski	Piotr	1	Juliusz	Mogiła
Baczyńska	Natalia	2	Jurek	Owsianka
Gierczak	Norbert	2	Jurek	Owsianka
Jurny	Kasia	2	Jurek	Owsianka
Marudny	Wojciech	2	Jurek	Owsianka
Nijaki	Patryk	2	Jurek	Owsianka
Skurzyński	Olgierd	2	Jurek	Owsianka
Włóczynoga	Hubert	2	Jurek	Owsianka
Zabłódzka	Martyna	2	Jurek	Owsianka
Bursztyński	Krystian	3	Matylda	Grzęda
Kemnitz	Sebastian	3	Matylda	Grzęda
Kieruj	Paulina	3	Matylda	Grzęda
Majewski	Mateusz	3	Matylda	Grzęda

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

Melkowski	Jakub	3	Matylda	Grzęda
Rubaj	Stanisław	3	Matylda	Grzęda
Szadowski	Łukasz	3	Matylda	Grzęda
Wójcik	Marcin	3	Matylda	Grzęda

## 4.1.2. Lista dostępnych wychowawców

Sprawdź, czy aby na pewno wychowawcy nie są na zwolnieniu chorobowym wyświetlając ich id nauczyciela, id wychowawcy, imię oraz nazwisko.

```
SELECT nauczyciele.id_nauczyciela, wychowawca.id_wychowawcy, pracownicy.imię,  
pracownicy.nazwisko  
FROM nauczyciele  
INNER JOIN pracownicy ON nauczyciele.id_pracownika = pracownicy.id_pracownika  
JOIN wychowawca ON nauczyciele.id_nauczyciela = wychowawca.id_nauczyciela  
WHERE nauczyciele.id_nauczyciela != ALL (SELECT id_nauczyciela FROM zastępstwa);
```

id_nauczyciela	id_wychowawcy	imię	nazwisko
5	1	Matylda	Grzęda
13	2	Juliusz	Mogiła
16	3	Jurek	Owsianka

## 4.1.3. Sale wychowawców (ANY)

Aby rodzice wiedzieli, w jakiej sali są spotkania, stwórz listę z imieniem i nazwiskiem wychowawców oraz sal, w których prowadzą oni zajęcia.

```
SELECT DISTINCT pracownicy.imię, pracownicy.nazwisko, sala.nr_sali  
FROM pracownicy  
INNER JOIN nauczyciele ON pracownicy.id_pracownika = nauczyciele.id_pracownika  
JOIN kadra ON nauczyciele.id_nauczyciela = kadra.id_nauczyciela  
JOIN przedmiot ON kadra.id_przedmiotu = przedmiot.id_przedmiotu  
JOIN zajęcia ON przedmiot.id_przedmiotu = zajęcia.id_przedmiotu  
JOIN sala ON zajęcia.id_sali = sala.id_sali  
WHERE nauczyciele.id_nauczyciela = ANY  
(  
  SELECT id_nauczyciela  
  FROM wychowawca  
)  
GROUP BY pracownicy.nazwisko;
```

imię	nazwisko	nr_sali
Matylda	Grzęda	5
Juliusz	Mogiła	1
Jurek	Owsianka	30

## 4.1.4. Uczniowie bez ocen (NOT EXISTS)

Na spotkaniu z rodzicami podawane będą oceny z wszystkich przedmiotów uzyskane w pierwszym semestrze. Sprawdź, podając id ucznia, nazwisko, imię oraz numer klasy, czy są uczniowie, którzy nie uzyskali żadnych ocen w poprzednim semestrze.

```
SELECT id_ucznia AS numer_ucznia, nazwisko, imię, id_klasy AS numer_klasy  
FROM uczniowie  
WHERE NOT EXISTS  
(  
  SELECT id_ucznia  
  FROM dziennik_ocen  
  WHERE uczniowie.id_ucznia = dziennik_ocen.id_ucznia  
);
```

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

numer_ucznia	nazwisko	imię	numer_klasy
6	Kieruj	Paulina	3
10	Daniel	Eryk	1
18	Jurny	Kasia	2
24	Włóczynoga	Hubert	2

## 4.1.5. Średnie uczniów z każdego przedmiotu (GROUP BY)

Jednym z celów spotkania z rodzicami jest poinformowanie ich o osiągnięciach w nauce ich pociech. Wyświetl imię i nazwisko ucznia, numer przedmiotu oraz średnią, jaką z niego uzyskał. Wyniki posortuj według nazwiska, rosnąco.

```
SELECT uczniowie.imię, uczniowie.nazwisko, przedmiot.nazwa, AVG(dziennik_ocen.ocena) AS 'średnia'
FROM uczniowie
INNER JOIN dziennik_ocen ON uczniowie.id_ucznia = dziennik_ocen.id_ucznia
INNER JOIN przedmiot ON dziennik_ocen.id_przedmiotu = przedmiot.id_przedmiotu
GROUP BY dziennik_ocen.id_ucznia, dziennik_ocen.id_przedmiotu
```

ORDER BY uczniowie.nazwisko ASC

imię	nazwisko	przedmiot	średnia
Natalia	Baczyńska	technika	3.0000
Natalia	Baczyńska	WOS	1.6667
Natalia	Baczyńska	przyroda	4.0000
Natalia	Baczyńska	WF	3.3333
Natalia	Baczyńska	język_angielski	2.6667
Natalia	Baczyńska	historia	5.0000
Natalia	Baczyńska	język_polski	3.6667
Natalia	Baczyńska	geografia	2.6667
Natalia	Baczyńska	matematyka	4.3333
Natalia	Baczyńska	informatyka	3.3333
Natalia	Baczyńska	muzyka	4.3333
Krystian	Bursztyński	język_angielski	4.0000
Krystian	Bursztyński	WF	2.3333
Krystian	Bursztyński	WDŻ	3.3333
Krystian	Bursztyński	historia	3.6667
Krystian	Bursztyński	język_polski	3.6667
Krystian	Bursztyński	technika	4.0000
Krystian	Bursztyński	WOS	2.6667
Krystian	Bursztyński	geografia	3.3333
Krystian	Bursztyński	matematyka	3.0000
Krystian	Bursztyński	język_niemiecki	4.3333
Krystian	Bursztyński	muzyka	3.6667
Krystian	Bursztyński	informatyka	3.0000
Paweł	Garlicki	historia	3.6667
Paweł	Garlicki	język_polski	4.0000
Paweł	Garlicki	matematyka	4.0000
Paweł	Garlicki	przyroda	4.3333
Paweł	Garlicki	plastyka	5.0000
Paweł	Garlicki	WF	3.3333
Paweł	Garlicki	informatyka	3.6667
Norbert	Gierczak	język_polski	3.3333
Norbert	Gierczak	muzyka	4.3333
Norbert	Gierczak	język_angielski	3.6667
Norbert	Gierczak	informatyka	5.0000
Norbert	Gierczak	WF	3.6667

Norbert	Gierczak	matematyka	4.0000
Norbert	Gierczak	geografia	3.6667
Norbert	Gierczak	historia	3.0000
Norbert	Gierczak	technika	2.6667
Norbert	Gierczak	przyroda	4.0000
Norbert	Gierczak	WOS	4.6667
Joanna	Ibisz	plastyka	5.0000
Joanna	Ibisz	matematyka	3.0000
Joanna	Ibisz	język_polski	5.0000
Joanna	Ibisz	historia	3.3333
Joanna	Ibisz	WF	4.0000
Joanna	Ibisz	przyroda	4.3333
Joanna	Ibisz	informatyka	4.0000
Sebastian	Kemnitz	matematyka	4.0000
Sebastian	Kemnitz	język_niemiecki	4.0000
Sebastian	Kemnitz	język_angielski	4.0000
Sebastian	Kemnitz	geografia	4.3333
Sebastian	Kemnitz	WDŻ	4.0000
Sebastian	Kemnitz	technika	3.6667
Sebastian	Kemnitz	historia	4.3333
Sebastian	Kemnitz	język_polski	4.0000
Sebastian	Kemnitz	informatyka	3.3333
Sebastian	Kemnitz	WOS	4.0000
Sebastian	Kemnitz	muzyka	4.3333
Sebastian	Kemnitz	WF	4.0000
Zuzanna	Kopytko	przyroda	4.3333
Zuzanna	Kopytko	język_polski	4.3333
Zuzanna	Kopytko	matematyka	3.3333
Zuzanna	Kopytko	plastyka	5.0000
Zuzanna	Kopytko	historia	4.3333
Zuzanna	Kopytko	WF	3.3333
Zuzanna	Kopytko	informatyka	3.6667
Brajan	Kowalski	plastyka	5.0000
Brajan	Kowalski	matematyka	4.3333
Brajan	Kowalski	przyroda	4.0000
Brajan	Kowalski	WF	4.6667



# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

Brajan	Kowalski	informatyka	4.0000
Brajan	Kowalski	język_polski	3.6667
Brajan	Kowalski	historia	3.6667
Marika	Majeranek	plastyka	5.0000
Marika	Majeranek	WF	2.6667
Marika	Majeranek	przyroda	4.6667
Marika	Majeranek	matematyka	2.0000
Marika	Majeranek	historia	3.3333
Marika	Majeranek	informatyka	2.6667
Marika	Majeranek	język_polski	3.0000
Mateusz	Majewski	WDŻ	2.0000
Mateusz	Majewski	matematyka	3.3333
Mateusz	Majewski	technika	4.3333
Mateusz	Majewski	język_niemiecki	4.0000
Mateusz	Majewski	muzyka	4.0000
Mateusz	Majewski	historia	3.3333
Mateusz	Majewski	geografia	4.6667
Mateusz	Majewski	WF	4.0000
Mateusz	Majewski	język_angielski	3.3333
Mateusz	Majewski	WOS	3.3333
Mateusz	Majewski	informatyka	2.6667
Mateusz	Majewski	język_polski	4.6667
Wojciech	Marudny	geografia	2.6667
Wojciech	Marudny	matematyka	3.6667
Wojciech	Marudny	język_angielski	4.6667
Wojciech	Marudny	technika	3.6667
Wojciech	Marudny	muzyka	3.3333
Wojciech	Marudny	WF	4.0000
Wojciech	Marudny	język_polski	3.6667
Wojciech	Marudny	historia	4.0000
Wojciech	Marudny	WOS	3.3333
Wojciech	Marudny	przyroda	4.3333
Wojciech	Marudny	informatyka	2.0000
Jakub	Melkowski	WOS	4.0000
Jakub	Melkowski	język_polski	4.6667
Jakub	Melkowski	informatyka	3.3333
Jakub	Melkowski	język_angielski	2.6667
Jakub	Melkowski	język_niemiecki	4.0000
Jakub	Melkowski	matematyka	4.6667
Jakub	Melkowski	WF	4.3333
Jakub	Melkowski	WDŻ	3.3333
Jakub	Melkowski	historia	4.0000
Jakub	Melkowski	technika	4.0000
Jakub	Melkowski	geografia	4.0000
Jakub	Melkowski	muzyka	3.3333
Patryk	Nijaki	matematyka	4.0000
Patryk	Nijaki	technika	4.0000
Patryk	Nijaki	muzyka	3.3333
Patryk	Nijaki	WF	3.0000
Patryk	Nijaki	WOS	2.3333
Patryk	Nijaki	geografia	5.0000
Patryk	Nijaki	historia	4.3333
Patryk	Nijaki	informatyka	3.3333
Patryk	Nijaki	przyroda	3.0000
Patryk	Nijaki	język_angielski	4.3333
Patryk	Nijaki	język_polski	3.3333
Stanisław	Rubaj	technika	4.0000

Stanisław	Rubaj	historia	3.6667
Stanisław	Rubaj	WOS	3.0000
Stanisław	Rubaj	język_niemiecki	4.3333
Stanisław	Rubaj	WF	4.0000
Stanisław	Rubaj	geografia	4.3333
Stanisław	Rubaj	informatyka	2.6667
Stanisław	Rubaj	język_angielski	3.6667
Stanisław	Rubaj	język_polski	3.3333
Stanisław	Rubaj	WDŻ	3.6667
Stanisław	Rubaj	matematyka	3.3333
Stanisław	Rubaj	muzyka	3.6667
Mariusz	Sanitariusz	informatyka	3.0000
Mariusz	Sanitariusz	matematyka	4.0000
Mariusz	Sanitariusz	przyroda	4.6667
Mariusz	Sanitariusz	język_polski	3.3333
Mariusz	Sanitariusz	plastyka	5.0000
Mariusz	Sanitariusz	WF	3.3333
Mariusz	Sanitariusz	historia	3.3333
Olgierd	Skurzyński	historia	4.6667
Olgierd	Skurzyński	język_polski	2.6667
Olgierd	Skurzyński	geografia	3.6667
Olgierd	Skurzyński	matematyka	3.6667
Olgierd	Skurzyński	WOS	3.3333
Olgierd	Skurzyński	język_angielski	4.0000
Olgierd	Skurzyński	przyroda	3.6667
Olgierd	Skurzyński	muzyka	3.6667
Olgierd	Skurzyński	informatyka	2.0000
Olgierd	Skurzyński	WF	3.0000
Olgierd	Skurzyński	technika	3.6667
Piotr	Stróżewski	historia	4.0000
Piotr	Stróżewski	język_polski	3.6667
Piotr	Stróżewski	informatyka	2.6667
Piotr	Stróżewski	przyroda	4.3333
Piotr	Stróżewski	matematyka	3.3333
Piotr	Stróżewski	plastyka	5.0000
Piotr	Stróżewski	WF	4.0000
Łukasz	Szadowski	muzyka	3.6667
Łukasz	Szadowski	WF	3.6667
Łukasz	Szadowski	język_niemiecki	4.3333
Łukasz	Szadowski	WDŻ	4.0000
Łukasz	Szadowski	matematyka	4.6667
Łukasz	Szadowski	język_polski	3.6667
Łukasz	Szadowski	technika	4.0000
Łukasz	Szadowski	historia	4.3333
Łukasz	Szadowski	geografia	5.0000
Łukasz	Szadowski	informatyka	2.6667
Łukasz	Szadowski	język_angielski	3.3333
Łukasz	Szadowski	WOS	3.6667
Marcin	Wójcik	matematyka	4.3333
Marcin	Wójcik	język_polski	4.6667
Marcin	Wójcik	WDŻ	4.6667
Marcin	Wójcik	geografia	4.0000
Marcin	Wójcik	informatyka	2.6667
Marcin	Wójcik	język_angielski	3.3333
Marcin	Wójcik	technika	4.3333
Marcin	Wójcik	WOS	4.6667
Marcin	Wójcik	WF	3.6667

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

Marcin	Wójcik	język_niemiecki	4.0000
Marcin	Wójcik	muzyka	4.0000
Marcin	Wójcik	historia	3.6667
Martyna	Zabłodzka	język_angielski	4.0000
Martyna	Zabłodzka	matematyka	3.6667
Martyna	Zabłodzka	przyroda	3.6667
Martyna	Zabłodzka	język_polski	4.0000

Martyna	Zabłodzka	muzyka	5.0000
Martyna	Zabłodzka	informatyka	5.0000
Martyna	Zabłodzka	WF	2.6667
Martyna	Zabłodzka	WOS	4.3333
Martyna	Zabłodzka	historia	4.0000
Martyna	Zabłodzka	technika	1.0000
Martyna	Zabłodzka	geografia	3.6667

## 4.1.6. Najlepsi uczniowie (HAVING, ANY)

Najlepszych uczniów należy nagrodzić za ich starania. Wyświetl wszystkich uczniów, którzy uzyskali średnią powyżej 4,5 z danego przedmiotu.

```
SELECT uczniowie.imię, uczniowie.nazwisko, przedmiot.nazwa AS 'przedmiot',  
AVG(dziennik_ocen.ocena) AS 'średnia'  
FROM uczniowie  
INNER JOIN dziennik_ocen ON uczniowie.id_ucznia = dziennik_ocen.id_ucznia  
INNER JOIN przedmiot ON dziennik_ocen.id_przedmiotu = przedmiot.id_przedmiotu  
GROUP BY dziennik_ocen.id_ucznia, dziennik_ocen.id_przedmiotu  
HAVING AVG(dziennik_ocen.ocena) >= 4.5  
ORDER BY uczniowie.nazwisko ASC;
```

imię	nazwisko	przedmiot	średnia
Natalia	Baczyńska	historia	5.0000
Paweł	Garlicki	plastyka	5.0000
Norbert	Gierczak	WOS	4.6667
Norbert	Gierczak	informatyka	5.0000
Joanna	Ibisz	język_polski	5.0000
Joanna	Ibisz	plastyka	5.0000
Zuzanna	Kopytko	plastyka	5.0000
Brajan	Kowalski	WF	4.6667
Brajan	Kowalski	plastyka	5.0000
Marika	Majeranek	przyroda	4.6667
Marika	Majeranek	plastyka	5.0000
Mateusz	Majewski	język_polski	4.6667
Mateusz	Majewski	geografia	4.6667
Wojciech	Marudny	język_angielski	4.6667

Jakub	Melkowski	język_polski	4.6667
Jakub	Melkowski	matematyka	4.6667
Patryk	Nijaki	geografia	5.0000
Mariusz	Sanitariusz	przyroda	4.6667
Mariusz	Sanitariusz	plastyka	5.0000
Olgiard	Skurzyński	historia	4.6667
Piotr	Stróżewski	plastyka	5.0000
Łukasz	Szadowski	matematyka	4.6667
Łukasz	Szadowski	geografia	5.0000
Marcin	Wójcik	język_polski	4.6667
Marcin	Wójcik	WOS	4.6667
Marcin	Wójcik	WDŻ	4.6667
Martyna	Zabłodzka	muzyka	5.0000
Martyna	Zabłodzka	informatyka	5.0000

## 4.1.7. Uczniowie bez żadnej opłaty (NOT IN)

Następną kwestią omawianą na spotkaniu z rodzicami będą opłaty. Sprawdź, czy jest taki uczeń, który nie dokonał żadnej wpłaty. Jeżeli taki uczeń istnieje, wyświetl jego numer, imię, nazwisko oraz numer klasy.

```
SELECT id_ucznia AS numer_ucznia, imię, nazwisko, id_klasy AS numer_klasy  
FROM uczniowie  
WHERE id_ucznia NOT IN  
(  
SELECT id_ucznia  
FROM opłaty  
);
```

numer_ucznia	imię	nazwisko	numer_klasy
13	Piotr	Stróżewski	1

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

## 4.1.8. Uczniowie bez opłaconego ubezpieczenia (ANY)

Najważniejszą opłatą jest ubezpieczenie. Sprawdź, czy są tacy uczniowie, którzy nie opłacili ubezpieczenia. Jeżeli są, to wyświetl ich imię, nazwisko, numer oraz numer klasy.

```
SELECT imię, nazwisko, id_ucznia AS numer_ucznia, id_klasy AS numer_klasy
FROM uczniowie
WHERE uczniowie.id_ucznia NOT IN
(
SELECT id_ucznia
FROM opłaty
WHERE typ_opłaty LIKE 'Ubezpieczenie'
);
```

imię	nazwisko	numer_ucznia	numer_klasy
Piotr	Stróżewski	13	1
Martyna	Zabłódzka	17	2

## 4.1.9. Łączna suma opłat (SKORELOWANE)

Sprawdź, ile pieniędzy wpłacił na opłaty każdy z uczniów, wyświetlając jego numer, imię i nazwisko oraz sumę wpłat. Dane posortuj według numeru ucznia rosnąco.

```
SELECT id_ucznia AS numer_ucznia, imię, nazwisko,
(SELECT SUM(wielkość_opłaty)
FROM szkoła.opłaty as O
WHERE O.id_ucznia = U.id_ucznia) AS SumaOpłat
FROM szkoła.uczniowie AS U
ORDER BY numer_ucznia ASC;
```

numer_ucznia	imię	nazwisko	SumaOpłat
1	Marcin	Wójcik	100
2	Sebastian	Kemnitz	100
3	Jakub	Melkowski	100
4	Stanisław	Rubaj	100
5	Łukasz	Szadowski	100
6	Paulina	Kieruj	100
7	Mateusz	Majewski	100
8	Krystian	Bursztyński	100
9	Mariusz	Sanitariusz	70
10	Eryk	Daniel	100
11	Zuzanna	Kopytko	100
12	Paweł	Garlicki	70
13	Piotr	Stróżewski	
14	Marika	Majeranek	100
15	Brajan	Kowalski	100
16	Joanna	Ibisz	100
17	Martyna	Zabłódzka	50
18	Kasia	Jurny	70
19	Olgierd	Skurzyński	70
20	Patryk	Nijaki	100
21	Norbert	Gierczak	100
22	Wojciech	Marudny	70
23	Natalia	Baczyńska	70
24	Hubert	Włóczynoga	70

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

## 4.1.10. Opłaty za kino (INNER JOIN, GROUP BY)

Na początku semestru odbędzie się wycieczka do kina. Policz, ilu uczniów z każdej klasy wpłaciło pieniądze na kino oraz ile wpłat jest ogółem.

```
SELECT COUNT(opłaty.id_ucznia) AS ilość_wpłat, uczniowie.id_klasy AS numer_klasy
FROM opłaty
INNER JOIN uczniowie ON opłaty.id_ucznia = uczniowie.id_ucznia
WHERE typ_opłaty LIKE 'Kino'
GROUP BY id_klasy
ORDER BY id_klasy ASC;
```

ilość_wpłat	numer_klasy
5	1
2	2
8	3

```
SELECT COUNT(id_ucznia) AS 'Ilość wszystkich wpłat na kino'
FROM opłaty
WHERE typ_opłaty LIKE 'Kino';
```

Ilość wszystkich wpłat na kino
15

## 4.1.11. Skarbnicy (INNER JOIN, SORTOWANIE)

Z okazji dnia kobiet organizowana będzie zbiórka pieniędzy na upominki dla wszystkich dziewczynek w klasach. Sprawdź najpierw, kto jest skarbnikiem w każdej klasie. Wyświetl imię, nazwisko oraz numer klasy każdego skarbnika.

```
SELECT imię, nazwisko, id_klasy AS numer_klasy
FROM uczniowie
INNER JOIN skarbnik ON uczniowie.id_ucznia = skarbnik.id_ucznia
ORDER BY id_klasy ASC;
```

imię	nazwisko	numer_klasy
Zuzanna	Kopytko	1
Martyna	Zabłodzka	2
Łukasz	Szadowski	3

## 4.1.12. Ile jest dziewczynek, a ilu chłopców (GROUP BY)

Oblicz, ilu jest chłopców, a ile dziewczynek w każdej klasie.

```
SELECT płeć, count(płeć) AS liczba, id_klasy
FROM uczniowie
GROUP BY płeć, id_klasy
ORDER BY id_klasy ASC;
```

płeć	liczba	id_klasy
Mężczyzna	5	1
Kobieta	3	1
Kobieta	3	2
Mężczyzna	5	2
Kobieta	1	3
Mężczyzna	7	3

## 4.1.13. Poszukiwania wandalą (LIKE)

Ostatnią kwestią do przedyskutowania jest kwestia brzydkiego rysunku, który pojawił się w jednej, że szkolnych toalet. Uczeń podpisał się pod rysunkiem inicjałami M.W. Wyszukaj wszystkich uczniów o inicjałach M.W oraz o inicjałach W.M, wyświetl imię, nazwisko oraz numer klasy.

```
SELECT imię, nazwisko, id_klasy AS numer_klasy
FROM uczniowie
WHERE imię LIKE 'M%' AND nazwisko LIKE 'W%' OR imię LIKE 'W%' AND nazwisko LIKE 'M%'
```

imię	nazwisko	numer_klasy
Marcin	Wójcik	3
Wojciech	Marudny	2

## 4.2. Kontrola z kuratorium

Niedługo do szkoły zawita kontrola z kuratorium. Przed wizytą należy przygotować odpowiednie dane potrzebne do statystyk. Szkoła będzie sprawdzana również pod kątem nepotyzmu.

### 4.2.1. Pensje nauczycieli (OUTER JOIN)

Pensje wszystkich nauczycieli powinny być jednakowe (pomijając dodatek wychowawczy), wyjątkiem są także nauczyciele zatrudniani z innych krajów. Wyświetl id każdego nauczyciela, imię, nazwisko oraz pensję, jaką on otrzymuje. Dane posortuj według pensji, malejąco.

```
SELECT nauczyciele.id_nauczyciela, pracownicy.imię, pracownicy.nazwisko, pracownicy.pensja
FROM szkoła.nauczyciele LEFT OUTER JOIN szkoła.pracownicy
ON nauczyciele.id_pracownika = pracownicy.id_pracownika
ORDER BY pracownicy.pensja DESC;
```

id_nauczyciela	imię	nazwisko	pensja
7	Konrad	Brzęczydło	3500
17	John	Smith	2900
6	Igor	Pelikan	2700
14	Krzysztof	Pieczarka	2700
15	Agata	Mysz	2700
8	Maria	Kwasigroch	2700
16	Jurek	Owsianka	2700
1	Marcin	Raczek	2700
9	Mateusz	Przeszkoda	2700
2	Adrianna	Hałas	2700
10	Julian	Tuniewim	2700
18	Włodzimierz	Tymianek	2700
3	Agata	Duda	2700
11	Beata	Worek	2700
19	Piotr	Luksja	2700
4	Marian	Kowal	2700
12	Apolonia	Kowalska	2700
20	Izabela	Luksja	2700
5	Matylda	Grzęda	2700
13	Juliusz	Mogiła	2700

## 4.2.2. Pensje nauczycieli wyższe niż 2700 zł (HAVING, GROUP BY, LIKE)

Dane te należy posortować. Wyświetl więc wszystkich nauczycieli, którzy zarabiają powyżej 2700 zł oraz czy są to zwykli nauczyciele, czy nauczyciele z zagranicy.

```
SELECT imię, nazwisko, pensja, stanowisko
FROM szkoła.pracownicy
GROUP BY pensja
HAVING pensja > 2700 AND stanowisko LIKE 'Nauczyciel%';
```

imię	nazwisko	pensja	stanowisko
Konrad	Brzęczydło	3500	Nauczyciel
John	Smith	2900	Nauczyciel_z_zagranicy

## 4.2.3. Pokrewieństwa wśród nauczycieli (HAVING, GROUP BY)

Szkoła musi zostać sprawdzona pod kątem nepotyzmu. Jeżeli w kwestiach pensji wystąpiły nieprawidłowości, należy sprawdzić, czy nazwiska pracowników się nie powtarzają. Jeżeli tak jest, wyświetl takie nazwiska oraz ich liczbę.

```
SELECT nazwisko, COUNT(nazwisko) AS 'ilość_nazwisk'
FROM pracownicy
GROUP BY nazwisko
HAVING ilość_nazwisk > 1;
```

nazwisko	ilość_nazwisk
Brzęczydło	2
Luksja	2

## 4.2.4. Nepotyzm w szkole (NIESKORELOWANE, IN)

Jeżeli nazwiska pracowników się powtarzają, to wyświetl imię, nazwisko, pensję oraz stanowisko takich pracowników.

```
SELECT imię, nazwisko, pensja, stanowisko
FROM pracownicy
WHERE nazwisko IN
(SELECT nazwisko
FROM pracownicy
GROUP BY nazwisko
HAVING COUNT(nazwisko) >= 2);
```

imię	nazwisko	pensja	stanowisko
Stefan	Brzęczydło	5000	Dyrektor
Konrad	Brzęczydło	3500	Nauczyciel
Piotr	Luksja	2700	Nauczyciel
Izabela	Luksja	2700	Nauczyciel

## 4.2.5. Średnie oceny z przedmiotów (INNER JOIN, GROUP BY)

Kuratorium zbiera dane do statystyk tworzonych przez Ministerstwo Edukacji Narodowej. Potrzebuje ono danych odnośnie tego, z jakim przedmiotem uczniowie radzą sobie najgorzej. Wyświetl średnią ocen z każdego przedmiotu oraz jego nazwę. Listę posortuj według średniej, malejąco.

```
SELECT AVG(dziennik_ocen.ocena) AS średnia_ocen, przedmiot.nazwa AS przedmiot
FROM dziennik_ocen
INNER JOIN przedmiot ON dziennik_ocen.id_przedmiotu = przedmiot.id_przedmiotu
GROUP BY dziennik_ocen.id_przedmiotu
ORDER BY średnia_ocen DESC;
```

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

średnia_ocen	przedmiot
5.0000	plastyka
4.1429	język_niemiecki
4.1026	przyroda
3.9231	geografia
3.8974	muzyka
3.8833	historia
3.8167	język_polski
3.7333	matematyka
3.6667	język_angielski
3.5714	WDŻ
3.5641	technika
3.5500	WF
3.4615	WOS
3.2333	informatyka

## 4.2.6. Urodzenia w miesiącu (REGEX)

Kuratorium zbiera dane do statystyk tworzonych przez Ministerstwo Edukacji Narodowej. Potrzebuje ono danych odnośnie tego, ile dzieci rodzi się w danym miesiącu. Używając wyrażeń regularnych policz, ilu uczniów urodziło się w danym miesiącu.

```
SELECT COUNT(id_ucznia) AS urodzenia_sty
FROM uczniowie
WHERE data_urodzenia REGEXP '.-01-.';
urodzenia_sty
2
```

```
SELECT COUNT(id_ucznia) AS urodzenia_lut
FROM uczniowie
WHERE data_urodzenia REGEXP '.-02-.';
urodzenia_lut
4
```

```
SELECT COUNT(id_ucznia) AS urodzenia_mar
FROM uczniowie
WHERE data_urodzenia REGEXP '.-03-.';
urodzenia_mar
2
```

```
SELECT COUNT(id_ucznia) AS urodzenia_kwi
FROM uczniowie
WHERE data_urodzenia REGEXP '.-04-.';
urodzenia_kwi
3
```

```
SELECT COUNT(id_ucznia) AS urodzenia_maj
FROM uczniowie
WHERE data_urodzenia REGEXP '.-05-.';
urodzenia_maj
2
```

```
SELECT COUNT(id_ucznia) AS urodzenia_cze
FROM uczniowie
WHERE data_urodzenia REGEXP '.-06-.';
urodzenia_cze
1
```

```
SELECT COUNT(id_ucznia) AS urodzenia_lip
FROM uczniowie
WHERE data_urodzenia REGEXP '.-07-.';
urodzenia_lip
3
```

```
SELECT COUNT(id_ucznia) AS urodzenia_sie
FROM uczniowie
WHERE data_urodzenia REGEXP '.-08-.';
urodzenia_sie
1
```

```
SELECT COUNT(id_ucznia) AS urodzenia_wrz
FROM uczniowie
WHERE data_urodzenia REGEXP '.-09-.';
urodzenia_wrz
0
```

```
SELECT COUNT(id_ucznia) AS urodzenia_paź
FROM uczniowie
WHERE data_urodzenia REGEXP '.-10-.';
urodzenia_paź
1
```

```
SELECT COUNT(id_ucznia) AS urodzenia_lis
FROM uczniowie
WHERE data_urodzenia REGEXP '.-11-.';
urodzenia_lis
3
```

```
SELECT COUNT(id_ucznia) AS urodzenia_gru
FROM uczniowie
WHERE data_urodzenia REGEXP '.-12-.';
urodzenia_gru
2
```

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

## 5. Dane w tabelach

### 5.1. Pracownicy

Id_pracownika	Id_przełożonego	Imię	Nazwisko	Stanowisko	Pensja	Etat
1		Stefan	Brzęczydło	Dyrektor	5000	pełen
2	1	Marcin	Raczek	Nauczyciel	2700	Pełen
3	1	Adrianna	Hałas	Nauczyciel	2700	Pełen
4	1	Agata	Duda	Nauczyciel	2700	Pełen
5	1	Marian	Kowal	Nauczyciel	2700	Pełen
6	1	Matylda	Grzęda	Nauczyciel	2700	Pełen
7	1	Igor	Pelikan	Nauczyciel	2700	Pełen
8	1	Konrad	Brzęczydło	Nauczyciel	3500	Pełen
9	1	Maria	Kwasigroch	Nauczyciel	2700	Pełen
10	1	Mateusz	Przeszkoda	Nauczyciel	2700	Pełen
11	1	Julian	Tuniewim	Nauczyciel	2700	Pełen
12	1	Beata	Worek	Nauczyciel	2700	Pełen
13	1	Apolonia	Kowalska	Nauczyciel	2700	Pełen
14	1	Juliusz	Mogiła	Nauczyciel	2700	Pełen
15	1	Krzysztof	Pieczarka	Nauczyciel	2700	Pełen
16	1	Agata	Mysz	Nauczyciel	2700	Pełen
17	1	Jurek	Owsianka	Nauczyciel	2700	Pełen
18	1	John	Smith	Nauczyciel_z zagranicy	2900	Pełen
19	1	Włodzimierz	Tymianek	Nauczyciel	2700	Pełen
20	1	Piotr	Luksja	Nauczyciel	2700	Pełen
21	1	Izabela	Luksja	Nauczyciel	2700	Pełen
22	1	Henryka	Awokado	Szef Kuchni	2300	Pełen
23	22	Stefan	Daniel	Kucharz	2000	Pełen
24	22	Judyta	Por	Kucharz	2000	Pełen
25	1	Józef	Kamień	Woźny	1500	Pełen
26	1	Henryk	Kaszanka	Woźny	1500	Pełen
27	1	Stefania	Huk	Sprzątaczk	1300	Pół
28	1	Helena	Cytrus	Sprzątaczk	1300	Pół
29	1	Alan	Modejko	Sprzątaczk	1300	Pół
30	1	Ewa	Czajka	Szatniarka	1000	Pół
31		Zdzisław	Gregorowicz	Ochroniarz	1600	Pół

### 5.2. Nauczyciele

Id_nauczyciela	Id_pracownika	chorobowe
1	2	0
2	3	1
3	4	1
4	5	0
5	6	0
6	7	1

7	8	0
8	9	0
9	10	0
10	11	1
11	12	0
12	13	0
13	14	0
14	15	0

15	16	0
16	17	0
17	18	0
18	19	0
19	20	0
20	21	0



# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

## 5.3. Wychowawca

Id_wychowawcy	Id_nauczyciela	Dodatek_wychowawczy	Wychowane_klasy
1	5	100	1
2	13	300	3
3	16	500	7

## 5.4. Klasa

Id_klasy	Id_wychowawcy	Rocznik
1	2	3
2	3	1
3	1	2

## 5.5. Uczniowie

Id_ucznia	Imię	Nazwisko	Data_urodzenia	Płeć	Id_klasy	Id_przewodniczącego
1	Marcin	Wójcik	1997-02-11	Mężczyzna	3	
2	Sebastian	Kemnitz	1997-02-22	Mężczyzna	3	1
3	Jakub	Melkowski	1997-07-31	Mężczyzna	3	1
4	Stanisław	Rubaj	1997-11-17	Mężczyzna	3	1
5	Łukasz	Szadowski	1997-03-13	Mężczyzna	3	1
6	Paulina	Kieruj	1997-04-01	Kobieta	3	1
7	Mateusz	Majewski	1997-08-10	Mężczyzna	3	1
8	Krystian	Bursztyński	1997-12-12	Mężczyzna	3	1
9	Mariusz	Sanitariusz	1999-05-11	Mężczyzna	1	12
10	Eryk	Daniel	1999-04-16	Mężczyzna	1	12
11	Zuzanna	Kopytko	1999-11-21	Kobieta	1	12
12	Paweł	Garlicki	1999-02-28	Mężczyzna	1	
13	Piotr	Stróżewski	1999-07-19	Mężczyzna	1	12
14	Marika	Majeranek	1999-10-02	Kobieta	1	12
15	Brajan	Kowalski	1999-05-01	Mężczyzna	1	12
16	Joanna	Ibisz	1999-06-26	Kobieta	1	12
17	Martyna	Zabłódzka	1998-07-27	Kobieta	2	20
18	Kasia	Jurny	1998-01-01	Kobieta	2	20
19	Olgiert	Skurzyński	1998-12-05	Mężczyzna	2	20
20	Patryk	Nijaki	1998-11-24	Mężczyzna	2	
21	Norbert	Gierczak	1998-01-01	Mężczyzna	2	20
22	Wojciech	Marudny	1998-03-30	Mężczyzna	2	20
23	Natalia	Baczyńska	1998-04-01	Kobieta	2	20
24	Hubert	Włóczynoga	1998-02-14	Mężczyzna	2	20

## 5.6. Przewodniczący

Id_przewodniczącego
1
12
20

## 5.7. Skarbnik

Id_skarbnika
5
11
17

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

## 5.8. Zwykły uczeń

Id_ucznia
2
3
4
6
7
8

9
10
13
14
15
16
18

19
21
22
23
24

## 5.9. Przedmiot

Id_przedmiotu	Nazwa	Godzin_tygodniowo
1	język_polski	12
2	język_angielski	4
3	język_niemiecki	2
4	matematyka	10
5	przyroda	2
6	technika	2
7	WF	3
8	godzina_wychowawcza	1
9	historia	2
10	geografia	2
11	plastyka	2
12	muzyka	2
13	WOS	4
14	WDŻ	2
15	informatyka	2

## 5.10. Opłaty

Id_ucznia	Typ_opłaty	Wielkość_opłaty
1	Ubezpieczenie	20
2	Ubezpieczenie	20
3	Ubezpieczenie	20
4	Ubezpieczenie	20
5	Ubezpieczenie	20
6	Ubezpieczenie	20
7	Ubezpieczenie	20
8	Ubezpieczenie	20
9	Ubezpieczenie	20
10	Ubezpieczenie	20
11	Ubezpieczenie	20
12	Ubezpieczenie	20
14	Ubezpieczenie	20
15	Ubezpieczenie	20
16	Ubezpieczenie	20
18	Ubezpieczenie	20
19	Ubezpieczenie	20
20	Ubezpieczenie	20

21	Ubezpieczenie	20
22	Ubezpieczenie	20
23	Ubezpieczenie	20
24	Ubezpieczenie	20
1	rada_rodziców	50
2	rada_rodziców	50
3	rada_rodziców	50
4	rada_rodziców	50
5	rada_rodziców	50
6	rada_rodziców	50
7	rada_rodziców	50
8	rada_rodziców	50
9	rada_rodziców	50
10	rada_rodziców	50
11	rada_rodziców	50
12	rada_rodziców	50
14	rada_rodziców	50
15	rada_rodziców	50
16	rada_rodziców	50

17	rada_rodziców	50
18	rada_rodziców	50
19	rada_rodziców	50
20	rada_rodziców	50
21	rada_rodziców	50
22	rada_rodziców	50
23	rada_rodziców	50
24	rada_rodziców	50
1	Kino	30
2	Kino	30
3	Kino	30
4	Kino	30
5	Kino	30
6	Kino	30
7	Kino	30
8	Kino	30
10	Kino	30
15	Kino	30
11	Kino	30

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

20	Kino	30
14	Kino	30

21	Kino	30
----	------	----

16	Kino	30
----	------	----

## 5.11. Sala

Id_sali	Nr_sali	Ilość_miejsc	Typ_sali
1	1	16	Matematyczna
2	5	9	Zwykła
3	6	10	Zwykła
4	8	16	Plastyczna
5	10	10	Zwykła
6	11	10	Biologiczna
7	12	16	Matematyczna
8	13	10	Biologiczna
9	15	12	Geograficzna
10	16	16	Plastyczna
11	18	13	Językowa
12	19	9	Zwykła
13	20	8	Informatyczna
14	24	14	Językowa
15	30	20	Sala_sportowa

## 5.12. Kadra

Id_kadry	Id_nauczyciela	Id_przedmiotu
1	1	2
2	2	7
3	3	8
4	4	3
5	5	1
6	6	11
7	7	15
8	8	10
9	9	9
10	10	1
11	11	14
12	12	13
13	13	4
14	14	3
15	15	5
16	16	7
17	17	4
18	18	6
19	19	7
20	20	12

## 5.13. Zajęcia

Id_zajęc	Id_przedmiotu	Id_sali	Id_klasy
1	1	2	2
2	4	1	2
3	5	6	2
4	7	15	2
5	8	12	2
6	9	3	2
7	11	4	2
8	15	13	2
9	1	2	3
10	2	11	3
11	4	14	3

12	5	1	3
13	6	7	3
14	7	15	3
15	8	12	3
16	9	3	3
17	10	9	3
18	12	10	3
19	13	5	3
20	15	13	3
21	1	2	1
22	2	11	1
23	3	4	1

24	4	1	1
25	6	6	1
26	7	7	1
27	8	12	1
28	9	3	1
29	10	9	1
30	12	10	1
31	13	5	1
32	14	8	1
33	15	13	1

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

## 5.14. Dziennik Ocen

id_oceny	ocena	id_ucznia	id_przedmiotu
1	5	1	1
2	4	1	1
3	5	1	1
4	4	2	1
5	3	2	1
6	5	2	1
7	5	3	1
8	4	3	1
9	5	3	1
10	4	4	1
11	2	4	1
12	4	4	1
13	4	5	1
14	3	5	1
15	4	5	1
16	5	7	1
17	5	7	1
18	4	7	1
19	3	8	1
20	5	8	1
21	3	8	1
22	4	9	1
23	3	9	1
24	3	9	1
25	4	11	1
26	4	11	1
27	5	11	1
28	5	12	1
29	4	12	1
30	3	12	1
31	4	13	1

32	3	13	1
33	4	13	1
34	3	14	1
35	4	14	1
36	2	14	1
37	4	15	1
38	3	15	1
39	4	15	1
40	6	16	1
41	5	16	1
42	4	16	1
43	4	17	1
44	5	17	1
45	3	17	1
46	3	19	1
47	4	19	1
48	1	19	1
49	5	20	1
50	2	20	1
51	3	20	1
52	2	21	1
53	5	21	1
54	3	21	1
55	4	22	1
56	3	22	1
57	4	22	1
58	4	23	1
59	4	23	1
60	3	23	1
61	3	1	2
62	3	1	2
63	4	1	2
64	2	2	2
65	5	2	2
66	5	2	2

67	2	3	2
68	4	3	2
69	2	3	2
70	5	4	2
71	3	4	2
72	3	4	2
73	5	5	2
74	3	5	2
75	2	5	2
76	3	7	2
77	4	7	2
78	3	7	2
79	3	8	2
80	4	8	2
81	5	8	2
82	4	17	2
83	5	17	2
84	3	17	2
85	4	19	2
86	5	19	2
...	...	...	...
586	5	20	15
587	2	20	15
588	3	20	15
589	5	21	15
590	5	21	15
591	5	21	15
592	1	22	15
593	3	22	15
594	2	22	15
595	1	23	15
596	4	23	15
597	5	23	15

## 5.15. Zajęcia

Id_zastępstwa	Id_nauczyciela	Id_przedmiotu
1	3	8
2	1	2

## 6. Użyte komendy

---

### 6.1. Tabele

---

#### 6.1.1. Pracownicy

---

```
CREATE TABLE IF NOT EXISTS szkola.pracownicy (  
  id_pracownika INT NOT NULL AUTO_INCREMENT,  
  id_przełożonego INT NULL,  
  imię VARCHAR(45) NOT NULL,  
  nazwisko VARCHAR(45) NOT NULL,  
  stanowisko VARCHAR(45) NOT NULL,  
  pensja INT NOT NULL,  
  etat VARCHAR(45) NOT NULL,  
  PRIMARY KEY (id_pracownika),  
  UNIQUE INDEX id_pracownika_UNIQUE (id_pracownika ASC) VISIBLE,  
  INDEX prac_prac_idx (id_przełożonego ASC) VISIBLE,  
  CONSTRAINT prac_prac  
    FOREIGN KEY (id_przełożonego)  
    REFERENCES szkola.pracownicy (id_pracownika)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

#### 6.1.2. Nauczyciele

---

```
CREATE TABLE IF NOT EXISTS szkola.nauczyciele (  
  id_nauczyciela INT NOT NULL AUTO_INCREMENT,  
  id_pracownika INT NOT NULL,  
  chorobowe TINYINT NULL,  
  PRIMARY KEY (id_nauczyciela),  
  UNIQUE INDEX id_nauczyciela_UNIQUE (id_nauczyciela ASC) VISIBLE,  
  INDEX nauc_prac_idx (id_pracownika ASC) VISIBLE,  
  CONSTRAINT nauc_prac  
    FOREIGN KEY (id_pracownika)  
    REFERENCES szkola.pracownicy (id_pracownika)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

#### 6.1.3. Wychowawca

---

```
CREATE TABLE IF NOT EXISTS szkola.wychowawca (  
  id_wychowawcy INT NOT NULL AUTO_INCREMENT,  
  id_nauczyciela INT NOT NULL,  
  dodatek_wychowawczy INT NULL,  
  wychowane_klasy INT NOT NULL,  
  PRIMARY KEY (id_wychowawcy),  
  UNIQUE INDEX id_wychowawcy_UNIQUE (id_wychowawcy ASC) VISIBLE,  
  INDEX wych_nauc_idx (id_nauczyciela ASC) VISIBLE,  
  CONSTRAINT wych_nauc  
    FOREIGN KEY (id_nauczyciela)  
    REFERENCES szkola.nauczyciele (id_nauczyciela)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

#### 6.1.4. Klasa

---

```
DROP TABLE IF EXISTS szkola.klasa ;
```

```
CREATE TABLE IF NOT EXISTS szkola.klasa (  
  id_klasy INT NOT NULL AUTO_INCREMENT,  
  id_wychowawcy INT NOT NULL,  
  rocznik INT NOT NULL,  
  PRIMARY KEY (id_klasy),  
  INDEX klas_wych_idx (id_wychowawcy ASC) VISIBLE,  
  CONSTRAINT klas_wych  
    FOREIGN KEY (id_wychowawcy)  
    REFERENCES szkola.wychowawca (id_wychowawcy)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

## 6.1.5. Uczniowie

---

```
CREATE TABLE IF NOT EXISTS szkola.uczniowie (  
  id_ucznia INT NOT NULL AUTO_INCREMENT,  
  imię VARCHAR(45) NOT NULL,  
  nazwisko VARCHAR(45) NOT NULL,  
  data_urodzenia DATE NOT NULL,  
  płeć VARCHAR(45) NOT NULL,  
  id_klasy INT NOT NULL,  
  id_przewodniczącego INT NULL,  
  PRIMARY KEY (id_ucznia),  
  UNIQUE INDEX id_ucznia_UNIQUE (id_ucznia ASC) VISIBLE,  
  INDEX uczn_klas_idx (id_klasy ASC) VISIBLE,  
  INDEX uczn_uczn_idx (id_przewodniczącego ASC) VISIBLE,  
  CONSTRAINT uczn_klas  
    FOREIGN KEY (id_klasy)  
    REFERENCES szkola.klasa (id_klasy)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT uczn_uczn  
    FOREIGN KEY (id_przewodniczącego)  
    REFERENCES szkola.uczniowie (id_ucznia)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

## 6.1.6. Opłaty

---

```
CREATE TABLE IF NOT EXISTS szkola.opłaty (  
  id_ucznia INT NOT NULL,  
  typ_opłaty VARCHAR(45) NOT NULL,  
  wielkość_opłaty INT NOT NULL,  
  PRIMARY KEY (id_opłaty),  
  UNIQUE INDEX id_opłaty_UNIQUE (id_opłaty ASC) VISIBLE,  
  INDEX opła_uczn_idx (id_ucznia ASC) VISIBLE,  
  CONSTRAINT opła_uczn  
    FOREIGN KEY (id_ucznia)  
    REFERENCES szkola.uczniowie (id_ucznia))  
ENGINE = InnoDB;
```

## 6.1.7. Zwyczajny\_uczeń

---

```
CREATE TABLE IF NOT EXISTS szkola.zwyczajny_uczeń (  
  id_ucznia INT NOT NULL,  
  INDEX zwyk_uczn_idx (id_ucznia ASC) VISIBLE,  
  CONSTRAINT zwyk_uczn  
    FOREIGN KEY (id_ucznia)  
    REFERENCES szkola.uczniowie (id_ucznia)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

ENGINE = InnoDB;

## 6.1.8. Przewodniczący

---

```
CREATE TABLE IF NOT EXISTS szkola.przewodniczący (  
  id_ucznia INT NOT NULL,  
  INDEX prze_uczn_idx (id_ucznia ASC) VISIBLE,  
  CONSTRAINT prze_uczn  
    FOREIGN KEY (id_ucznia)  
      REFERENCES szkola.uczniowie (id_ucznia)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

## 6.1.9. Skarbnik

---

```
CREATE TABLE IF NOT EXISTS szkola.skarbnik (  
  id_ucznia INT NOT NULL,  
  INDEX skar_uczn_idx (id_ucznia ASC) VISIBLE,  
  CONSTRAINT skar_uczn  
    FOREIGN KEY (id_ucznia)  
      REFERENCES szkola.uczniowie (id_ucznia)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

## 6.1.10. Przedmiot

---

```
CREATE TABLE IF NOT EXISTS szkola.przedmiot (  
  id_przedmiotu INT NOT NULL AUTO_INCREMENT,  
  nazwa VARCHAR(45) NOT NULL,  
  godzin_tygodniowo INT NOT NULL,  
  PRIMARY KEY (id_przedmiotu),  
  UNIQUE INDEX id_przedmiotu_UNIQUE (id_przedmiotu ASC) VISIBLE)  
ENGINE = InnoDB;
```

## 6.1.11. Dziennik\_ocen

---

```
CREATE TABLE IF NOT EXISTS szkola.dziennik_ocen (  
  id_oceny INT NOT NULL AUTO_INCREMENT,  
  id_ucznia INT NOT NULL,  
  id_przedmiotu INT NOT NULL,  
  PRIMARY KEY (id_oceny),  
  UNIQUE INDEX id_oceny_UNIQUE (id_oceny ASC) VISIBLE,  
  INDEX dzie_uczn_idx (id_ucznia ASC) VISIBLE,  
  INDEX dzie_prze_idx (id_przedmiotu ASC) VISIBLE,  
  CONSTRAINT dzie_uczn  
    FOREIGN KEY (id_ucznia)  
      REFERENCES szkola.uczniowie (id_ucznia)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT dzie_prze  
    FOREIGN KEY (id_przedmiotu)  
      REFERENCES szkola.przedmiot (id_przedmiotu)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

## 6.1.12. Zastępstwa

---

```
CREATE TABLE IF NOT EXISTS szkola.zastepstwa (  
  id_zastepstwa INT NOT NULL AUTO_INCREMENT,  
  id_nauczyciela INT NULL,  
  id_przedmiotu INT NULL,  
  id_zastepujacego INT NULL,  
  PRIMARY KEY (id_zastepstwa),  
  UNIQUE INDEX id_zastepstwa_UNIQUE (id_zastepstwa ASC) VISIBLE,  
  INDEX zast_nauc_idx (id_nauczyciela ASC) VISIBLE,  
  INDEX zast_przed_idx (id_przedmiotu ASC) VISIBLE,  
  INDEX zast_nauc_zast_idx (id_zastepujacego ASC) VISIBLE,  
  CONSTRAINT zast_nauc  
    FOREIGN KEY (id_nauczyciela)  
    REFERENCES szkola.nauczyciele (id_nauczyciela)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT zast_przed  
    FOREIGN KEY (id_przedmiotu)  
    REFERENCES szkola.przedmiot (id_przedmiotu)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT zast_nauc_zast  
    FOREIGN KEY (id_zastepujacego)  
    REFERENCES szkola.nauczyciele (id_nauczyciela)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

## 6.1.13. Kadra

---

```
CREATE TABLE IF NOT EXISTS szkola.kadra (  
  id_kadry INT NOT NULL AUTO_INCREMENT,  
  id_nauczyciela INT NOT NULL,  
  id_przedmiotu INT NOT NULL,  
  PRIMARY KEY (id_kadry),  
  UNIQUE INDEX id_kadry_UNIQUE (id_kadry ASC) VISIBLE,  
  INDEX kadr_nauc_idx (id_nauczyciela ASC) VISIBLE,  
  INDEX kadr_prze_idx (id_przedmiotu ASC) VISIBLE,  
  CONSTRAINT kadr_nauc  
    FOREIGN KEY (id_nauczyciela)  
    REFERENCES szkola.nauczyciele (id_nauczyciela)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT kadr_prze  
    FOREIGN KEY (id_przedmiotu)  
    REFERENCES szkola.przedmiot (id_przedmiotu)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

## 6.1.14. Sala

---

```
CREATE TABLE IF NOT EXISTS szkola.sala (  
  id_sali INT NOT NULL AUTO_INCREMENT,  
  nr_sali INT NOT NULL,  
  ilosc_miejsc INT NOT NULL,  
  typ_sali VARCHAR(45) NOT NULL,  
  PRIMARY KEY (id_sali),  
  UNIQUE INDEX id_sali_UNIQUE (id_sali ASC) VISIBLE)  
ENGINE = InnoDB;
```



## 6.1.15. Zajęcia

```
CREATE TABLE IF NOT EXISTS szkola.zajęcia (
  id_zajęc INT NOT NULL AUTO_INCREMENT,
  id_przedmiotu INT NOT NULL,
  id_sali INT NOT NULL,
  id_klasy INT NOT NULL,
  PRIMARY KEY (id_zajęc),
  UNIQUE INDEX id_zajęc_UNIQUE (id_zajęc ASC) VISIBLE,
  INDEX zaje_prze_idx (id_przedmiotu ASC) VISIBLE,
  INDEX zaje_sala_idx (id_sali ASC) VISIBLE,
  INDEX zaje_klas_idx (id_klasy ASC) VISIBLE,
  CONSTRAINT zaje_prze
    FOREIGN KEY (id_przedmiotu)
      REFERENCES szkola.przedmiot (id_przedmiotu)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT zaje_sala
    FOREIGN KEY (id_sali)
      REFERENCES szkola.sala (id_sali)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT zaje_klas
    FOREIGN KEY (id_klasy)
      REFERENCES szkola.klasa (id_klasy)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

## 6.2. Insert

### 6.2.1. Pracownicy

```
INSERT INTO szkola.pracownicy
```

```
(id_pracownika, id_przełożonego, imię, nazwisko, stanowisko, pensja, etat)
VALUES
```

1. (default, null, 'Stefan', 'Brzęczydło', 'Dyrektor', 5000, 'pełen'),
2. (default, 1, 'Marcin', 'Raczek', 'Nauczyciel', 2700, 'Pełen'),
3. (default, 1, 'Adrianna', 'Hałas', 'Nauczyciel', 2700, 'Pełen'),
4. (default, 1, 'Agata', 'Duda', 'Nauczyciel', 2700, 'Pełen'),
5. (default, 1, 'Marian', 'Kowal', 'Nauczyciel', 2700, 'Pełen'),
6. (default, 1, 'Matylda', 'Grzęda', 'Nauczyciel', 2700, 'Pełen'), #[wychowawca]
7. (default, 1, 'Igor', 'Pelikan', 'Nauczyciel', 2700, 'Pełen'),
8. (default, 1, 'Konrad', 'Brzęczydło', 'Nauczyciel', 2800, 'Pełen'),
9. (default, 1, 'Maria', 'Kwasigroch', 'Nauczyciel', 2700, 'Pełen'),
10. (default, 1, 'Mateusz', 'Przeszkoda', 'Nauczyciel', 2700, 'Pełen'),
11. (default, 1, 'Julian', 'Tuniewim', 'Nauczyciel', 2700, 'Pełen'),
12. (default, 1, 'Beata', 'Worek', 'Nauczyciel', 2700, 'Pełen'),
13. (default, 1, 'Apolonia', 'Kowalska', 'Nauczyciel', 2700, 'Pełen'),
14. (default, 1, 'Juliusz', 'Mogiła', 'Nauczyciel', 2700, 'Pełen'), #[wychowawca]
15. (default, 1, 'Krzysztof', 'Pieczarka', 'Nauczyciel', 2700, 'Pełen'),
16. (default, 1, 'Agata', 'Mysz', 'Nauczyciel', 2700, 'Pełen'),
17. (default, 1, 'Jurek', 'Owsianka', 'Nauczyciel', 2700, 'Pełen'), #[wychowawca]
18. (default, 1, 'John', 'Smith', 'Nauczyciel', 2900, 'Pełen'),
19. (default, 1, 'Włodzimierz', 'Tymianek', 'Nauczyciel', 2700, 'Pełen'),
20. (default, 1, 'Piotr', 'Luksja', 'Nauczyciel', 2700, 'Pełen'),
21. (default, 1, 'Izabela', 'Luksja', 'Nauczyciel', 2700, 'Pełen'),
22. (default, 1, 'Henryka', 'Awokado', 'Szef Kuchni', 2300, 'Pełen'),
23. (default, 22, 'Stefan', 'Daniel', 'Kucharz', 2000, 'Pełen'),
24. (default, 22, 'Judyta', 'Por', 'Kucharz', 2000, 'Pełen'),
25. (default, 1, 'Józef', 'Kamień', 'Woźny', 1500, 'Pełen'),

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

```
26. (default, 1, 'Henryk', 'Kaszanka', 'Woźny', 1500, 'Pełen'),
27. (default, 1, 'Stefania', 'Huk', 'Sprzątaczką', 1300, 'Pół'),
28. (default, 1, 'Helena', 'Cytrus', 'Sprzątaczką', 1300, 'Pół'),
29. (default, 1, 'Stefan', 'Brzęczydło', 'Dyrektor', 5000, 'Pełen'),
30. (default, 1, 'Alan', 'Modejko', 'Sprzątaczką', 1300, 'Pół'),
31. (default, 1, 'Ewa', 'Czajka', 'Szatniarka', 1000, 'Pół'),
32. (default, null, 'Zdzisław', 'Gregorowicz', 'Ochroniarz', 1600, 'Pół');
```

## 6.2.2. Nauczyciele

```
INSERT INTO szkola.nauczyciele
(id_nauczyciela, id_pracownika, chorobowe)
VALUES
(default, 2, 0),           (default, 9, 0),           (default, 16, 0),
(default, 3, 0),           (default, 10, 0),          (default, 17, 0),
(default, 4, 0),           (default, 11, 0),          (default, 18, 0),
(default, 5, 0),           (default, 12, 0),          (default, 19, 0),
(default, 6, 0),           (default, 13, 0),          (default, 20, 0),
(default, 7, 0),           (default, 14, 0),          (default, 21, 0);
(default, 8, 0),           (default, 15, 0),
```

## 6.2.3. Wychowawca

```
INSERT INTO szkola.wychowawca
(id_wychowawcy, id_nauczyciela, dodatek_wychowawczy, wychowane_klasy)
VALUES
(default, 5, null, 0),
(default, 13, null, 3),
(default, 16, null, 1);
```

## 6.2.4. Klasa

```
INSERT INTO szkola.klasa
(id_klasy, id_wychowawcy, rocznik)
VALUES
(default, 2, 3),
(default, 3, 1),
(default, 1, 2);
```

## 6.2.5. Uczniowie

```
INSERT INTO szkola.uczniowie
(id_ucznia, imię, nazwisko, data_urodzenia, płeć, id_klasy, id_przewodniczącego)
VALUES
1. (default, 'Marcin', 'Wójcik', '1997-02-11', 'Mężczyzna', 1, null),      #[przewodniczący]
2. (default, 'Sebastian', 'Kemnitz', '1997-02-22', 'Mężczyzna', 1, null),
3. (default, 'Jakub', 'Melkowski', '1997-07-31', 'Mężczyzna', 1, null),
4. (default, 'Stanisław', 'Rubaj', '1997-11-17', 'Mężczyzna', 1, null),
5. (default, 'Łukasz', 'Szadowski', '1997-03-13', 'Mężczyzna', 1, null),  #[skarbnik]
6. (default, 'Paulina', 'Kieruj', '1997-04-01', 'Kobieta', 1, null),
7. (default, 'Mateusz', 'Majewski', '1997-8-10', 'Mężczyzna', 1, null),
8. (default, 'Krystian', 'Bursztyński', '1997-12-12', 'Mężczyzna', 1, null),
9. (default, 'Mariusz', 'Sanitariusz', '1999-05-11', 'Mężczyzna', 2, null),
10. (default, 'Eryk', 'Daniel', '1999-04-16', 'Mężczyzna', 2, null),
11. (default, 'Zuzanna', 'Kopytko', '1999-11-21', 'Kobieta', 2, null),      #[skarbnik]
12. (default, 'Paweł', 'Garlicki', '1999-02-28', 'Mężczyzna', 2, null),    #[przewodniczący]
13. (default, 'Piotr', 'Stróżewski', '1999-07-19', 'Mężczyzna', 2, null),
14. (default, 'Marika', 'Majeranek', '1999-10-02', 'Kobieta', 2, null),
15. (default, 'Brajan', 'Kowalski', '1999-05-01', 'Mężczyzna', 2, null),
16. (default, 'Joanna', 'Ibisz', '1999-06-26', 'Kobieta', 2, null),
17. (default, 'Martyna', 'Zabłodźka', '1998-07-27', 'Kobieta', 3, null),  #[skarbnik]
```

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

```
18. (default, 'Kasia', 'Jurny', '1998-01-01', 'Kobieta', 3, null),
19. (default, 'Olgierd', 'Skurzyński', '1998-12-05', 'Mężczyzna', 3, null),
20. (default, 'Patryk', 'Nijaki', '1998-11-24', 'Mężczyzna', 3, null),      #[przewodniczący]
21. (default, 'Norbert', 'Gierczak', '1998-01-01', 'Mężczyzna', 3, null),
22. (default, 'Jakub', 'Byczyński', '1998-03-30', 'Mężczyzna', 3, null),
23. (default, 'Natalia', 'Baczyńska', '1998-04-01', 'Kobieta', 3, null),
24. (default, 'Hubert', 'Włóczynoga', '1998-02-14', 'Mężczyzna', 3, null);
```

## 6.2.6. Zwykły\_uczeń

```
INSERT INTO szkola.zwykły_uczeń
(id_ucznia)
VALUES
(2), (3), (4), (6), (7), (8),
(9), (10), (13), (14), (15), (16),
(18), (19), (21), (22), (23), (24);
```

## 6.2.7. Skarbnik

```
INSERT INTO szkola.skarbnik
(id_ucznia)
VALUES
(5), (11), (17);
```

## 6.2.8. Przewodniczący

```
INSERT INTO szkola.przewodniczący
(id_ucznia)
VALUES
(1), (12), (20);
```

## 6.2.10. Opłaty

```
INSERT INTO szkola.opłaty
(id_opłaty, id_ucznia, typ_opłaty, wielkość_opłaty)
VALUES
#<-- ubezpieczenie -->
(default, 1, 'Ubezpieczenie', 20),
(default, 2, 'Ubezpieczenie', 20),
(default, 3, 'Ubezpieczenie', 20),
(default, 4, 'Ubezpieczenie', 20),
(default, 5, 'Ubezpieczenie', 20),
(default, 6, 'Ubezpieczenie', 20),
(default, 7, 'Ubezpieczenie', 20),
(default, 8, 'Ubezpieczenie', 20),
(default, 9, 'Ubezpieczenie', 20),
(default, 10, 'Ubezpieczenie', 20),
(default, 11, 'Ubezpieczenie', 20),
(default, 12, 'Ubezpieczenie', 20),
(default, 14, 'Ubezpieczenie', 20),
(default, 15, 'Ubezpieczenie', 20),
(default, 16, 'Ubezpieczenie', 20),
(default, 18, 'Ubezpieczenie', 20),
(default, 19, 'Ubezpieczenie', 20),
(default, 20, 'Ubezpieczenie', 20),
(default, 21, 'Ubezpieczenie', 20),
(default, 22, 'Ubezpieczenie', 20),
(default, 23, 'Ubezpieczenie', 20),
(default, 24, 'Ubezpieczenie', 20),
#<-- rada_rodziców -->
(default, 1, 'rada_rodziców', 50),
(default, 2, 'rada_rodziców', 50),
```

## 6.2.9. Przedmiot

```
INSERT INTO szkola.przedmiot
(id_przedmiotu, nazwa, godzin_tygodniowo)
VALUES
(default, 'język_polski', 12),
(default, 'język_angielski', 4),
(default, 'język_niemiecki', 2),
(default, 'matematyka', 10),
(default, 'przyroda', 2),
(default, 'technika', 2),
(default, 'WF', 3),
(default, 'godzina_wychowawcza', 1),
(default, 'historia', 2),
(default, 'geografia', 2),
(default, 'plastyka', 2),
(default, 'muzyka', 2),
(default, 'WOS', 4),
(default, 'WDŻ', 2),
(default, 'informatyka', 2);
```

```
(default, 3, 'rada_rodziców', 50),
(default, 4, 'rada_rodziców', 50),
(default, 5, 'rada_rodziców', 50),
(default, 6, 'rada_rodziców', 50),
(default, 7, 'rada_rodziców', 50),
(default, 8, 'rada_rodziców', 50),
(default, 9, 'rada_rodziców', 50),
(default, 10, 'rada_rodziców', 50),
(default, 11, 'rada_rodziców', 50),
(default, 12, 'rada_rodziców', 50),
(default, 14, 'rada_rodziców', 50),
(default, 15, 'rada_rodziców', 50),
(default, 16, 'rada_rodziców', 50),
(default, 17, 'rada_rodziców', 50),
(default, 18, 'rada_rodziców', 50),
(default, 19, 'rada_rodziców', 50),
(default, 20, 'rada_rodziców', 50),
(default, 21, 'rada_rodziców', 50),
(default, 22, 'rada_rodziców', 50),
(default, 23, 'rada_rodziców', 50),
(default, 24, 'rada_rodziców', 50),
#<-- kino -->
(default, 1, 'Kino', 30),
(default, 2, 'Kino', 30),
(default, 3, 'Kino', 30),
(default, 4, 'Kino', 30),
```

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

```
(default, 5, 'Kino', 30),
(default, 6, 'Kino', 30),
(default, 7, 'Kino', 30),
(default, 8, 'Kino', 30),
(default, 10, 'Kino', 30),
(default, 15, 'Kino', 30),
```

## 6.2.11. Sala

```
INSERT INTO szkola.sala
(id_sali, nr_sali, ilość_miejsc, typ_sali)
VALUES
(default, 1, 16, 'Matematyczna'),
(default, 5, 9, 'Zwykła'),
(default, 6, 10, 'Zwykła'),
(default, 8, 16, 'Plastyczna'),
(default, 10, 10, 'Zwykła'),
(default, 11, 10, 'Biologiczna'),
(default, 12, 16, 'Matematyczna'),
(default, 13, 10, 'Biologiczna'),
(default, 1, 2),
(default, 2, 7),
(default, 3, 8),
(default, 4, 3),
(default, 5, 1),
(default, 6, 11),
(default, 7, 15),
(default, 8, 10),
(default, 9, 9),
(default, 10, 1),
```

```
(default, 11, 'Kino', 30),
(default, 20, 'Kino', 30),
(default, 14, 'Kino', 30),
(default, 21, 'Kino', 30),
(default, 16, 'Kino', 30);
```

```
(default, 15, 10, 'Geograficzna'),
(default, 16, 16, 'Plastyczna'),
(default, 18, 13, 'Językowa'),
(default, 19, 9, 'Zwykła'),
(default, 20, 8, 'Informatyczna'),
(default, 24, 14, 'Językowa'),
(default, 30, 20, 'Sala_sportowa');
```

## 6.2.12. Kadra

```
INSERT INTO szkola.kadra
(id_kadry, id_nauczyciela, id_przedmiotu)
VALUES
(default, 11, 14),
(default, 12, 13),
(default, 13, 4),
(default, 14, 16),
(default, 15, 5),
(default, 16, 7),
(default, 17, 4),
(default, 18, 6),
(default, 19, 16),
(default, 20, 12);
```

## 6.2.13. Zajęcia

```
INSERT INTO szkola.zajęcia
(id_zajęc, id_przedmiotu, id_sali, id_klasy)
VALUES
(default, 1, 2, 2),
(default, 4, 1, 2),
(default, 5, 6, 2),
(default, 7, 15, 2),
(default, 8, 12, 2),
(default, 9, 3, 2),
(default, 11, 4, 2),
(default, 15, 13, 2),
(default, 1, 2, 3),
(default, 2, 11, 3),
(default, 4, 14, 3),
(default, 5, 1, 3),
(default, 6, 7, 3),
(default, 7, 15, 3),
(default, 8, 12, 3),
(default, 9, 3, 3),
(default, 10, 9, 3),
(default, 12, 10, 3),
(default, 13, 5, 3),
(default, 15, 13, 3),
(default, 1, 2, 1),
(default, 2, 11, 1),
(default, 3, 4, 1),
(default, 4, 1, 1),
(default, 6, 6, 1),
(default, 7, 7, 1),
(default, 8, 12, 1),
(default, 9, 3, 1),
(default, 10, 9, 1),
(default, 12, 10, 1),
(default, 13, 5, 1),
(default, 14, 8, 1),
(default, 15, 13, 1);
```

## 6.2.14. Ocena

```
INSERT INTO szkola.ocena
(id_ucznia, id_przedmiotu, ocena)
VALUES
```

#p1	(2, 1, 2),	(4, 1, 4),	(7, 1, 5),	(8, 1, 1),	(11, 1, 3),	...	(21, 15, 2),	(23, 15, 3),
	(2, 1, 5),	(4, 1, 2),	(7, 1, 5),		(11, 1, 5),	...	(21, 15, 5),	(23, 15, 5),
(1, 1, 5),		(4, 1, 1),	(7, 1, 6),	(9, 1, 4),		...	(21, 15, 1),	(23, 15, 1);
(1, 1, 4),	(3, 1, 1),		(7, 1, 3),	(9, 1, 3),	(12, 1, 5),	...		
(1, 1, 3),	(3, 1, 4),	(5, 1, 6),		(9, 1, 2),	(12, 1, 4),	...	(22, 15, 1),	
	(3, 1, 5),	(5, 1, 2),	(8, 1, 2),		(12, 1, 3),	...	(22, 15, 5),	
(2, 1, 4),		(5, 1, 4),	(8, 1, 5),	(11, 1, 4),		...	(22, 15, 3),	

# Dokumentacja

Marcin Wójcik Informatyka Stosowana 18/19

---

## 6.3. Update

```
UPDATE szkola.wychowawca SET wychowane_klasy = '7' WHERE (id_wychowawcy = '3');
UPDATE szkola.wychowawca SET wychowane_klasy = '3' WHERE (id_wychowawcy = '2');
UPDATE szkola.wychowawca SET wychowane_klasy = '3' WHERE (id_wychowawcy = '2');

UPDATE szkola.kadra SET id_przedmiotu = '3' WHERE (id_kadry = '14');
UPDATE szkola.kadra SET id_przedmiotu = '7' WHERE (id_kadry = '19');

UPDATE szkola.pracownicy SET id_przełożonego = 22 WHERE stanowisko = 'Kucharz';

UPDATE szkola.uczniowie SET id_przewodniczącego = '1' WHERE (id_ucznia = '2');
UPDATE szkola.uczniowie SET id_przewodniczącego = '1' WHERE (id_ucznia = '3');
UPDATE szkola.uczniowie SET id_przewodniczącego = '1' WHERE (id_ucznia = '4');
UPDATE szkola.uczniowie SET id_przewodniczącego = '1' WHERE (id_ucznia = '5');
UPDATE szkola.uczniowie SET id_przewodniczącego = '1' WHERE (id_ucznia = '6');
UPDATE szkola.uczniowie SET id_przewodniczącego = '1' WHERE (id_ucznia = '7');
UPDATE szkola.uczniowie SET id_przewodniczącego = '1' WHERE (id_ucznia = '8');
UPDATE szkola.uczniowie SET id_przewodniczącego = '12' WHERE (id_ucznia = '9');
UPDATE szkola.uczniowie SET id_przewodniczącego = '12' WHERE (id_ucznia = '10');
UPDATE szkola.uczniowie SET id_przewodniczącego = '12' WHERE (id_ucznia = '11');
UPDATE szkola.uczniowie SET id_przewodniczącego = '12' WHERE (id_ucznia = '13');

UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '1');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '2');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '3');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '4');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '5');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '6');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '7');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '8');
UPDATE szkola.dziennik_ocen SET wzorowy = NULL WHERE (id_oceny = '9');
```

---

## 6.4. Delete

```
DELETE FROM szkola.wychowawca WHERE (id_wychowawcy = '4');
DELETE FROM szkola.przedmiot WHERE (id_przedmiotu = '16');
DELETE FROM szkola.ocena WHERE (id_przedmiotu = '1');
```

---

## 6.5. Select

```
SELECT przedmiot.id_przedmiotu FROM przedmiot
    JOIN kadra ON kadra.id_przedmiotu = przedmiot.id_przedmiotu
    WHERE kadra.id_nauczyciela LIKE zastępstwa.id_nauczyciela;
SELECT * FROM szkola.dziennik_ocen;
SELECT * FROM szkola.kadra;
SELECT * FROM szkola.klasa;
SELECT * FROM szkola.nauczyciele;
SELECT * FROM szkola.opłaty;
SELECT * FROM szkola.pracownicy;
SELECT * FROM szkola.przedmiot;
SELECT * FROM szkola.przewodniczący;
SELECT * FROM szkola.sala;
SELECT * FROM szkola.skarbnik;
SELECT * FROM szkola.uczniowie;
SELECT * FROM szkola.wychowawca;
SELECT * FROM szkola.zajęcia;
SELECT * FROM szkola.zastępstwa;
SELECT * FROM szkola.zwyczaj_uczeń;
```

## 6.6. Inne

---

```
ALTER TABLE szkola.zastepstwa
DROP FOREIGN KEY zast_nauc_zast;
ALTER TABLE szkola.zastepstwa
DROP COLUMN id_zastepujacego,
DROP INDEX zast_nauc_zast_idx ;
ALTER TABLE szkola.dziennik_ocen
CHANGE COLUMN ocena ocena VARCHAR(40) NOT NULL ;
ALTER TABLE szkola.dziennik_ocen
ADD COLUMN wzorowy TINYINT NULL AFTER ocena;
DROP TRIGGER IF EXISTS dodatek;

ALTER TABLE szkola.ocena
CHANGE COLUMN id_ucznia id_ucznia INT(11) NOT NULL FIRST,
DROP PRIMARY KEY,
ADD PRIMARY KEY (id_ucznia, ocena);
;

ALTER TABLE szkola.ocena
CHANGE COLUMN id_przedmiotu id_przedmiotu INT(11) NULL DEFAULT NULL AFTER id_ucznia;

ALTER TABLE szkola.ocena
CHANGE COLUMN id_przedmiotu id_przedmiotu INT(11) NOT NULL ,
DROP PRIMARY KEY,
ADD PRIMARY KEY (id_ucznia, ocena, id_przedmiotu);
;

ALTER TABLE szkola.dziennik_ocen
DROP FOREIGN KEY dzie_prze,
DROP FOREIGN KEY dzie_uczn;
ALTER TABLE szkola.dziennik_ocen
DROP COLUMN ocena,
CHANGE COLUMN id_ucznia id_ucznia INT(11) NULL ,
CHANGE COLUMN id_przedmiotu id_przedmiotu INT(11) NULL ;
ALTER TABLE szkola.dziennik_ocen
ADD CONSTRAINT dzie_prze
    FOREIGN KEY (id_przedmiotu)
    REFERENCES szkola.przedmiot (id_przedmiotu),
ADD CONSTRAINT dzie_uczn
    FOREIGN KEY (id_ucznia)
    REFERENCES szkola.uczniowie (id_ucznia);

ALTER TABLE szkola.ocena
ADD CONSTRAINT ocen_dzie
    FOREIGN KEY (id_ucznia)
    REFERENCES szkola.dziennik_ocen (id_ucznia)
ON DELETE NO ACTION
ON UPDATE NO ACTION;
```

## 7. Źródła

- Dokumentacja MySQL
- <https://stackoverflow.com/>
- Prezentacje z drugiego semestru podstaw baz danych
- [http://smurf.mimuw.edu.pl/external\\_slides/W3\\_Modelowanie\\_danych\\_Model\\_zwiazkow-encji/W3\\_Modelowanie\\_danych\\_Model\\_zwiazkow\\_encji.html](http://smurf.mimuw.edu.pl/external_slides/W3_Modelowanie_danych_Model_zwiazkow-encji/W3_Modelowanie_danych_Model_zwiazkow_encji.html)
- <https://www.sqlpedia.pl/podzapytania-skorelowane-sql/>
- <https://www.youtube.com/watch?v=rTuWRdOKYtU>