

Bazy Danych - Laboratorium 2

Michał Waluś

Baza danych używając

```
services:
  mariadb:
    image: mariadb:latest
    container_name: mariadb-container
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: aparaty
      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
```

Używanie bazy danych

```
# Up DB
docker compose up -d
# Down DB
docker compose down

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

Zad.1

Tworzenie bazy danych **aparaty**

```
CREATE DATABASE aparaty;
```

Tworzenie użytkownika z hasłem

```
CREATE USER '279695'@'%' IDENTIFIED BY 'walus95';
```

Nadanie uprawnień użytkownikowi

```
GRANT SELECT, INSERT, UPDATE ON aparaty.* TO '279695'@'%;  
FLUSH PRIVILEGES;
```

Zad.2

Wybieranie bazy danych

```
USE aparaty;
```

Tworzenie tabeli Matryca

```
CREATE TABLE Matryca (  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  przekatna DECIMAL(4,2) CHECK (przekatna >= 0),  
  rozdzielczosc DECIMAL(3,1) CHECK (rozdzielczosc >= 0),  
  typ VARCHAR(10)  
) AUTO_INCREMENT = 100;
```

Tworzenie tabeli Obiektyw

```
CREATE TABLE Obiektyw (  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  model VARCHAR(30),  
  minPrzeslona FLOAT CHECK (minPrzeslona >= 0),  
  maxPrzeslona FLOAT CHECK (maxPrzeslona > minPrzeslona),  
  CONSTRAINT max_lt_min CHECK (minPrzeslona < maxPrzeslona)  
) AUTO_INCREMENT = 1;
```

Tworzenie tabeli Producent

```
CREATE TABLE Producent (  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  nazwa VARCHAR(50) NOT NULL,  
  kraj VARCHAR(20) DEFAULT 'nieznany',
```

```
    adresKorespondencyjny VARCHAR(100) DEFAULT 'nieznany'
) AUTO_INCREMENT = 1;
```

Tworzenie tabeli **Aparat** - tworzymy jako ostatnią by pozostałe 3 były już stworzone i byśmy mogli z nich skorzystać

```
CREATE TABLE Aparat (
    model VARCHAR(30) PRIMARY KEY,
    producent INT,
    matryca INT,
    obiektyw INT,
    waga FLOAT,
    typ ENUM('kompaktowy', 'lustrzanka', 'profesjonalny', 'inny')
);
```

Dodajemy klucze obce

```
ALTER TABLE Aparat ADD CONSTRAINT FK_producent FOREIGN KEY (producent)
REFERENCES Producent (ID);
ALTER TABLE Aparat ADD CONSTRAINT FK_matryca FOREIGN KEY (matryca)
REFERENCES Matryca(ID);
ALTER TABLE Aparat ADD CONSTRAINT FK_obiektyw FOREIGN KEY (obiektyw)
REFERENCES Obiektyw(ID);
```

Zad.3

Połączenie z bazą danych jako użytkownik

```
docker exec -it mariadb-container mariadb -D aparaty -u 279695 -pwalus95
```

Dodawanie danych

```
INSERT INTO Producent(nazwa,kraj) VALUES ('Canon', 'Japonia');
INSERT INTO Producent(nazwa,kraj) VALUES ('Nikon', 'Japonia');
INSERT INTO Producent(nazwa,kraj) VALUES ('Sony', 'Japonia');
INSERT INTO Producent(nazwa,kraj) VALUES ('Panasonic', 'Japonia');
INSERT INTO Producent(nazwa,kraj) VALUES ('Olympus', 'Japonia');
INSERT INTO Producent(nazwa,kraj) VALUES ('Fujifilm', 'Japonia');
INSERT INTO Producent(nazwa,kraj) VALUES ('Huawei', 'Chiny');
INSERT INTO Producent(nazwa,kraj) VALUES ('Xiaomi', 'Chiny');
INSERT INTO Producent(nazwa,kraj) VALUES ('DJI', 'Chiny');
INSERT INTO Producent(nazwa,kraj) VALUES ('YI Technology', 'Chiny');
INSERT INTO Producent(nazwa,kraj) VALUES ('Zhiyun Technology', 'Chiny');
INSERT INTO Producent(nazwa,kraj) VALUES ('GoPro', 'USA');
INSERT INTO Producent(nazwa,kraj) VALUES ('Kodak', 'USA');
```

```
INSERT INTO Producent(nazwa,kraj) VALUES ('Leica', 'Niemcy');
INSERT INTO Producent(nazwa,kraj) VALUES ('Hasselblad', 'Szwecja');

INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.1, 20.1, 'CMOS');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.2, 18.3, 'CCD');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.5, 22.0, 'CMOS');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.3, 19.7, 'CCD');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.4, 21.5, 'CMOS');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.6, 23.2, 'CCD');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.7, 24.0, 'CMOS');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.8, 25.5, 'CCD');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (2.0, 26.8, 'CMOS');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (1.9, 28.3, 'CCD');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (2.1, 29.0, 'CMOS');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (2.2, 30.5, 'LIDAR');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (2.3, 32.1, 'CMOS');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (2.4, 34.0, 'CCD');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (2.5, 36.0, 'CMOS');

INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Canon EF 50mm f/1.8 STM', 1.8, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Nikon AF-S 50mm f/1.8G', 1.8, 16);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Sony FE 24-70mm f/2.8 GM', 2.8, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Tamron 70-200mm f/2.8 G2', 2.8, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Sigma 35mm f/1.4 DG HSM', 1.4, 16);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Canon RF 85mm f/1.2L USM', 1.2, 16);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Nikon Z 24-70mm f/2.8 S', 2.8, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Fujifilm XF 23mm f/1.4 R', 1.4, 16);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Olympus M.Zuiko 8 PRO', 2.8, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Leica Summilux-M 50mmH', 1.4, 16);
```

```
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Panasonic Lumix G 25mm', 1.7, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Tokina AT-X 11-16mm', 2.8, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Samyang 14mm f/2.8', 2.8, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Zeiss Loxia 35mm f/2', 2, 22);
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('Voigtlander 35mm', 1.4, 16);

INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Canon EOS 5D Mark II', 1, 100, 1, 800, 'lustrzanka');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Canon EOS 5D Mark III', 1, 101, 1, 800, 'lustrzanka');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Canon EOS 5D Mark IV', 1, 102, 1, 800, 'lustrzanka');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Canon EOS 5D Mark V', 1, 103, 1, 800, 'lustrzanka');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Nikon D850', 2, 101, 2, 915, 'lustrzanka');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Sony Alpha 7R IV', 3, 102, 3, 665, 'profesjonalny');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Panasonic Lumix GH5', 4, 103, 4, 725, 'profesjonalny');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Olympus OM-D E-M1 Mark III', 5, 104, 5, 500, 'profesjonalny');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Fujifilm X-T4', 6, 105, 6, 600, 'profesjonalny');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Huawei P40 Pro', 7, 106, 7, 209, 'kompaktowy');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Xiaomi Mi 10 Ultra', 8, 107, 8, 220, 'kompaktowy');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('DJI Osmo Pocket', 9, 108, 9, 116, 'kompaktowy');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('YI 4K Action Camera', 10, 109, 10, 94, 'kompaktowy');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Zhiyun Crane M2', 11, 110, 11, 500, 'inny');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('GoPro HERO10 Black', 12, 111, 12, 153, 'kompaktowy');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Kodak PIXPRO SP360 4K', 13, 112, 13, 196, 'kompaktowy');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Leica M10-R', 14, 113, 14, 660, 'lustrzanka');
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES ('Hasselblad X1D II 50C', 15, 114, 15, 725, 'profesjonalny');
```

Próba wstawienia niepoprawnych danych

```
INSERT INTO Producent(nazwa,kraj) VALUES (NULL, 'Szwecja');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (-2.5, 36.0,
```

```
'CMOS');  
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES  
( 'Voigtlander 35mm', 142, 16);  
INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES  
( 'Hasselblad', 15, 1000, 15, 725, 'profesjonalny');
```

Zad.4

Tworzenie procedurę tworzącą 100 losowych aparatów

```
DELIMITER $$  
  
CREATE PROCEDURE GenerujLosoweAparaty()  
BEGIN  
    DECLARE i INT DEFAULT 0;  
    DECLARE nazwa_modelu VARCHAR(255);  
    DECLARE losowy_producent INT;  
    DECLARE losowa_matryca INT;  
    DECLARE losowy_obiektyw INT;  
    DECLARE losowa_waga INT;  
    DECLARE losowy_typ ENUM('kompaktowy', 'lustrzanka', 'profesjonalny',  
'inny');  
  
    WHILE i < 100 DO  
        SET nazwa_modelu = CONCAT('Model ', i + 1);  
  
        SET losowy_producent = FLOOR(1 + (RAND() * 15));  
        SET losowa_matryca = FLOOR(100 + (RAND() * 15));  
        SET losowy_obiektyw = FLOOR(1 + (RAND() * 15));  
  
        SET losowa_waga = FLOOR(300 + (RAND() * 1200));  
        SET losowy_typ = ELT(FLOOR(1 + (RAND() * 4)), 'kompaktowy',  
'lustrzanka', 'profesjonalny', 'inny');  
  
        IF NOT EXISTS (SELECT 1 FROM Aparat WHERE model = nazwa_modelu)  
        THEN  
            INSERT INTO Aparat (model, producent, matryca, obiektyw, waga,  
            typ)  
            VALUES (nazwa_modelu, losowy_producent, losowa_matryca,  
            losowy_obiektyw, losowa_waga, losowy_typ);  
            END IF;  
  
            SET i = i + 1;  
        END WHILE;  
  
    END$$  
  
DELIMITER ;
```

Wykonujemy procedurę

```
CALL GenerujLosoweAparaty;
```

Użytkownik nie może wykonać procedury, bo nie ma uprawnień do **EXECUTE**.

Zad.5

Tworzenie funkcji

```
DELIMITER $$

CREATE OR REPLACE FUNCTION NajmniejszaPrzekatnaProducenta(id_producenta
INT) RETURNS VARCHAR(255)
BEGIN
    DECLARE nazwa_modelu VARCHAR(255);

    SELECT Aparat.model
    INTO nazwa_modelu
    FROM Aparat
    JOIN Matryca ON Aparat.matryca = Matryca.ID
    WHERE Aparat.producent = id_producenta
    ORDER BY Matryca.przekatna ASC
    LIMIT 1;

    RETURN nazwa_modelu;
END$$

DELIMITER ;
```

Wykonanie funkcji

```
SELECT NajmniejszaPrzekatnaProducenta(id);
```

Zad.6

Tworzenie trigger'a

```
DELIMITER $$

CREATE TRIGGER DodajProducentaAparatu
BEFORE INSERT ON Aparat
FOR EACH ROW
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Producent WHERE ID = NEW.producent) THEN
        INSERT INTO Producent (ID, nazwa) VALUES (NEW.producent, 'Nowy
Producent');
    END IF;
END IF;
```

```
END$$  
  
DELIMITER ;
```

Zad.7

```
DELIMITER $$  
  
CREATE OR REPLACE FUNCTION LiczbaModeli(matrycaID INT) RETURNS INT  
BEGIN  
    RETURN (SELECT COUNT(*) FROM Aparat WHERE matryca = matrycaID);  
END$$  
  
DELIMITER ;
```

Wykonanie funkcji

```
SELECT LiczbaModeli(id);
```

Zad.8

Tworzenie trigger'a

```
DELIMITER $$  
  
CREATE TRIGGER UsunNieuzyteMatryce  
AFTER DELETE ON Aparat  
FOR EACH ROW  
BEGIN  
    IF NOT EXISTS (SELECT 1 FROM Aparat WHERE matryca = OLD.matryca) THEN  
        DELETE FROM Matryca WHERE ID = OLD.matryca;  
    END IF;  
END$$  
  
DELIMITER ;
```

Zad.9

Tworzenie widoku

```
CREATE VIEW Lustrzanki AS  
SELECT Aparat.model, Aparat.waga, Producent.nazwa, Matryca.przekatna,  
Matryca.rozdzielczosc, Obiektow.minPrzeslona, Obiektow.maxPrzeslona  
FROM Aparat
```



```
JOIN Producent ON Aparat.producent = Producent.ID
JOIN Matryca ON Aparat.matryca = Matryca.ID
JOIN Obiektow ON Aparat.obiektow = Obiektow.ID
WHERE Aparat.typ = 'lustrzanka' AND Producent.kraj != 'Chiny';
```

Użytkownik nie może wykonać tego zadania, ponieważ nie ma uprawnień do tworzenia widoku.

Zad.10

Tworzenie widoku

```
CREATE VIEW ProducentAparatu AS
SELECT Aparat.model, Producent.nazwa, Producent.kraj
FROM Aparat
JOIN Producent ON Aparat.producent = Producent.ID;
```

Usuwanie producentów z Chin

```
DELETE FROM Aparat WHERE producent IN (SELECT ID FROM Producent WHERE kraj
= 'Chiny');
```

Ponieważ widok sam nie przechowuje danych, tylko kwerendę, która wykonujesz się przy każdym zapytaniu do widoku, dane usunięte z tabeli Aparat zostały również usunięte z widoku.

Zad.11

Dodanie kolumny

```
ALTER TABLE Producent ADD COLUMN liczbaModeli INT NOT NULL DEFAULT 0;
```

Dodanie danych do istniejących producentów

```
UPDATE Producent SET liczbaModeli = (SELECT COUNT(*) FROM Aparat WHERE
Aparat.producent = Producent.ID);
```

Tworzenie nowych trigger'ów (Można by zaktualizować trigger z zadania 8, ale wtedy użyta nazwa była by myłąca)

```
DELIMITER $$

CREATE TRIGGER WstawAparat
AFTER INSERT ON Aparat
```

```
FOR EACH ROW
BEGIN
    if NEW.producent IS NOT NULL THEN
        UPDATE Producent SET liczbaModeli = liczbaModeli + 1 WHERE ID =
NEW.producent;
    END IF;
END$$

CREATE TRIGGER ZaktualizujAparat
AFTER UPDATE ON Aparat
FOR EACH ROW
BEGIN
    IF OLD.producent IS NOT NULL THEN
        UPDATE Producent SET liczbaModeli = liczbaModeli - 1 WHERE ID =
OLD.producent;
    END IF;
    IF NEW.producent IS NOT NULL THEN
        UPDATE Producent SET liczbaModeli = liczbaModeli + 1 WHERE ID =
NEW.producent;
    END IF;
END$$

CREATE TRIGGER UsunAparat
AFTER DELETE ON Aparat
FOR EACH ROW
BEGIN
    IF OLD.producent IS NOT NULL THEN
        UPDATE Producent SET liczbaModeli = liczbaModeli - 1 WHERE ID =
OLD.producent;
    END IF;
END$$

DELIMITER ;
```

Użytkownik nie może utworzyć trigger'a, bo nie ma uprawnienie **CREATE TRIGGER**, ale już stworzony będzie działać niezależnie od tego, który użytkownik go wykona.