

# Laboratorium 2

## Zadanie 1

```
CREATE DATABASE `db-aparaty`  
    CHARACTER SET = 'utf8mb4';  
CREATE USER `268456`@`localhost`;  
SET PASSWORD FOR `268456`@`localhost` = PASSWORD('Lukasz456');  
GRANT SELECT, INSERT, UPDATE ON `db-aparaty`.*  
    TO `268456`@`localhost`;  
FLUSH PRIVILEGES;
```

## Zadanie 2

```
CREATE TABLE Matryca(  
    ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    przekatna decimal(4, 2) NOT NULL CHECK(przekatna > 0),  
    rozdzielczosc decimal(3, 1) NOT NULL CHECK(rozdzielczosc > 0),  
    typ varchar(10) NOT NULL  
);  
ALTER TABLE Matryca  
    AUTO_INCREMENT=100;  
CREATE TABLE Obiektyw(  
    ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    model varchar(30) NOT NULL,  
    minPrzeslona float NOT NULL CHECK(minPrzeslona > 0),  
    maxPrzeslona float NOT NULL CHECK(maxPrzeslona > minPrzeslona)  
);  
CREATE TABLE Producent(  
    ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nazwa varchar(50),  
    kraj varchar(20)  
);
```

```

CREATE TABLE Aparat(
    model varchar(30) NOT NULL PRIMARY KEY,
    producent int NOT NULL CHECK(producent > 0),
    matryca int NOT NULL CHECK(matryca > 0),
    obiekttyw int NOT NULL CHECK(obiekttyw > 0),
    typ enum('kompaktowy', 'lustrzanka', 'profesjonalny', 'inny') NOT NULL,
    FOREIGN KEY(producent)
        REFERENCES Producent(ID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
    FOREIGN KEY(matryca)
        REFERENCES Matryca(ID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
    FOREIGN KEY(obiekttyw)
        REFERENCES Obiekttyw(ID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

```

## Zadanie 3

```

INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES
    (43.3, 98, 'CMOS'),
    (27.5, 64, 'Exmor'),
    (27.1, 48, 'LiveMos'),
    (22.5, 43.7, 'X-Trans'),
    (16, 32, 'CCD'),
    (11, 98, 'BSI'),
    (10.5, 64, 'Foveon'),
    (9.5, 48, 'CMOS'),
    (8.93, 43.7, 'Exmor'),
    (7.2, 32, 'LiveMos'),
    (6.59, 98, 'X-Trans'),
    (43.3, 64, 'CCD'),
    (27.5, 48, 'BSI'),
    (27.1, 43.7, 'Foveon'),
    (22.5, 32, 'CMOS');

```

Dodatkowo przy próbie wprowadzenia np. takiej komendy:

```
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (43.3, -7, 'CMOS');
```

wyświetli się komunikat

ERROR 4025 (23000): CONSTRAINT `Matryca.rozdzielczosc` failed for `db-aparaty`.`Matryca`  
gdyż rozdzielczosc nie może być mniejsza niż 0.

```
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES
('Nikon 50mm AF-S', 1.8, 16),
('Sony 50mm OSS', 1.8, 22),
('CANON 16 mm STM', 2.8, 22),
('CANON 85mm STM', 2, 29),
('Nikon 40 mm DX', 2.8, 22),
('CANON 100mm USM', 2.8, 32),
('Sony 18-105 mm OSS', 4, 22),
('CANON 70-200 mm USM', 4, 32),
('Nikon 28-75 mm AF-S', 2.8, 22),
('Nikon 35 mm AF-S', 1.8, 22),
('CANON 75-300 mm EF', 5.6, 32),
('Sony 50 mm FE', 1.8, 22),
('Nikon 10-20 mm AF-P', 5.6, 22),
('Nikon 50 mm AF-S', 1.4, 16),
('Nikon 40mm AF-S', 2, 16);
```

Dodatkowo przy próbie wprowadzenia np. takiej komendy:

```
INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES ('NIKON', 16, 2);
```

wyświetli się komunikat

ERROR 4025 (23000): CONSTRAINT `Obiektyw.maxPrzeslona` failed for `db-aparaty`.`Obiektyw`  
gdyż maxPrzeslona nie może być mniejsza od minPrzeslona.

```
INSERT INTO Producent(nazwa, kraj) VALUES
('Canon', 'Chiny'),
('Casio', 'Japonia'),
('Kodak', 'Chiny'),
('Hasselblad', 'Szwecja'),
('Kiev', 'Ukraina'),
('Leica', 'Niemcy'),
('Lumix', 'Japonia'),
('Nikon', 'Japonia'),
('Panasonic', 'Japonia'),
('Pentax', 'Chiny'),
```

```
('Prakica', 'Niemcy'),
('Rollei', 'Niemcy'),
('Sony', 'Japonia'),
('Zenit', 'Chiny'),
('Zorkij', 'Chiny');
```

```
INSERT INTO Aparat(model, producent, matryca, obiektyw, typ) VALUES
(
    'Sony A7 IV ILCE7M4B',
    (SELECT ID FROM Producent WHERE nazwa = 'Sony'),
    (SELECT ID FROM Matryca WHERE przekatna = 43.3 AND rozdzielczosc = 98),
    (SELECT ID FROM Obiektyw WHERE model = 'Nikon 50mm AF-S'),
    'kompaktowy'
),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 5), ' EOS R + RF FHEA73'), 5,
103, 28, 'lustrzanka'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 4), ' EOS R6 DBBA4'), 4, 111, 30,
'profesjonalny'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 6), ' EOS R QWTA8348'), 6, 111,
18, 'inny'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 2), ' EOS R6 + VLLF9'), 2, 114,
17, 'kompaktowy'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 15), ' A7 III YFJSC64GA'), 15,
103, 26, 'lustrzanka'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 9), ' EOS R3 UTS67OJA'), 9, 107,
19, 'profesjonalny'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 11), ' EOS R5 BN7A'), 11, 103,
17, 'inny'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 11), ' EOS R4PONX'), 11, 105, 22,
'kompaktowy'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 12), ' EOS R3+ MLA'), 12, 112,
19, 'lustrzanka'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 13), ' A7 II NAG5B'), 13, 103,
23, 'profesjonalny'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 10), ' A5 QTY2'), 10, 103, 19,
'inny'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 2), ' A6 HNAK8'), 2, 107, 19,
'kompaktowy'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 9), ' P9 AGKN'), 9, 110, 29,
'lustrzanka'),
(CONCAT((SELECT nazwa FROM Producent WHERE ID = 8), ' P11 AGKAV'), 8, 108, 20,
'profesjonalny');
```

Dodatkowo przy próbie wprowadzenia np. takiej komendy:

```
INSERT INTO Aparat(model, producent, matryca, obiektow, typ) VALUES ('Testowa  
nazwa', 1, 2, 3, 'lustrzanka');
```

wyświetli się komunikat

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails  
(`db-aparaty`.`Aparat`, CONSTRAINT `Aparat\_ibfk\_2` FOREIGN KEY (`matryca`) REFERENCES  
`Matryca` (`ID`) ON UPDATE CASCADE)

gdyż wartość kolumny matryca jest powiązana z kluczem tabeli Matryca a podana wartość w niej nie istnieje.

## Zadanie 4

```
DELIMITER $$  
CREATE PROCEDURE generujKrotki()  
BEGIN  
    DECLARE i INT DEFAULT 0;  
    WHILE i < 100 DO  
        SET @p = FLOOR(1 + RAND() * 15);  
        SET @m = FLOOR(100 + RAND() * 15);  
        SET @o = FLOOR(16 + RAND() * 15);  
        SET @t = FLOOR(1 + RAND() * 4);  
        INSERT INTO Aparat(model, producent, matryca, obiektow, typ) VALUES(  
            CONCAT(  
                (SELECT nazwa FROM Producent WHERE ID = @p), ' M',  
                (SELECT COUNT(*) FROM Aparat AS a)  
            ),  
            @p, @m, @o, @t  
        );  
        SET i = i + 1;  
    END WHILE;  
END$$  
DELIMITER ;  
CALL generujKrotki();
```

Wcześniej stworzony użytkownik nie może stworzyć ani wykonać tej procedury.

## Zadanie 5

```
DELIMITER $$
CREATE PROCEDURE najwiekszaMatryca(id INT)
BEGIN
    SELECT model FROM Aparat JOIN Matryca ON Aparat.matryca = Matryca.ID WHERE
    Aparat.producent = id ORDER BY Matryca.przekatna DESC LIMIT 1;
END$$
DELIMITER ;
CALL najwiekszaMatryca(1);
```

Jako argument wstawiamy ID producenta którego aparat chcemy wyszukać

## Zadanie 6

```
DELIMITER $$
CREATE TRIGGER insert_any_aparat
BEFORE INSERT ON Aparat
FOR EACH ROW
BEGIN
    IF NOT EXISTS(SELECT * FROM Producent WHERE ID = NEW.producent) THEN
        INSERT INTO Producent(ID) VALUES(NEW.producent);
    END IF;
END$$
DELIMITER ;
```

Aby sprawdzić działanie wykonuję następujące polecenie, które nie zwraca błędu i zgodnie z poleceniem dodaje nowego producenta.

```
INSERT INTO Aparat VALUES('Model nowego producenta', 123, 100, 16, 4);
```

## Zadanie 7

```
CREATE FUNCTION ileAparatow(id INT)
RETURNS INT
RETURN (SELECT COUNT(*) FROM Aparat WHERE matryca = id);
SELECT ileAparatow(9);
```

## Zadanie 8

```
DELIMITER $$
CREATE TRIGGER delete_last_matryca
AFTER DELETE ON Aparat
FOR EACH ROW
BEGIN
    IF (SELECT COUNT(*) FROM Aparat WHERE matryca = OLD.matryca) = 0 THEN
        DELETE FROM Matryca WHERE ID = OLD.matryca;
    END IF;
END$$
DELIMITER ;
```

## Zadanie 9

```
CREATE VIEW lustrzanki AS
SELECT Aparat.model, Producent.nazwa, Matryca.przekatna, Matryca.rozdzielczosc,
    Obiektyw.minPrzeslona, Obiektyw.maxPrzeslona
FROM Aparat
JOIN Producent ON Aparat.producent = Producent.ID
JOIN Matryca ON Aparat.matryca = Matryca.ID
JOIN Obiektyw ON Aparat.obiektyw = Obiektyw.ID
WHERE Aparat.typ = 'lustrzanka' AND Producent.kraj <> 'Chiny';
```

Zadania nie może wykonać użytkownik z zadania 1.

## Zadanie 10

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

Sprawdź działanie widoku za pomocą następujących komend:

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

Po usunięciu wszystkich chińskich aparatów w stworzonym widoku również się usunęły.

## Zadanie 11

```
ALTER TABLE Producent ADD  
    liczbaModeli int NOT NULL DEFAULT 0;
```

Uzupełniłem tę kolumnę za pomocą następującej procedury:

```
DELIMITER $$  
  
CREATE PROCEDURE liczbaModeli()  
BEGIN  
    DECLARE i INT DEFAULT 1;  
    WHILE i < 16 DO  
        UPDATE Producent  
            SET liczbaModeli = (SELECT COUNT(*) FROM Aparat WHERE producent = i)  
            WHERE ID = i;  
        SET i = i + 1;  
    END WHILE;  
END$$  
  
DELIMITER ;  
  
CALL liczbaModeli();
```

I dodałem następujące triggery:

```
DELIMITER $$  
  
CREATE TRIGGER insert_aparat  
AFTER INSERT ON Aparat  
FOR EACH ROW  
BEGIN  
    UPDATE Producent  
        SET liczbaModeli = liczbaModeli + 1  
        WHERE ID = NEW.producent;  
END$$  
  
CREATE TRIGGER delete_aparat  
AFTER DELETE ON Aparat  
FOR EACH ROW  
BEGIN  
    UPDATE Producent  
        SET liczbaModeli = liczbaModeli - 1  
        WHERE ID = OLD.producent;  
END$$  
  
CREATE TRIGGER update_aparat  
AFTER UPDATE ON Aparat  
FOR EACH ROW  
BEGIN
```



```
IF OLD.producent <> NEW.producent THEN
    UPDATE Producent
        SET liczbaModeli = liczbaModeli - 1
        WHERE ID = OLD.producent;
    UPDATE Producent
        SET liczbaModeli = liczbaModeli + 1
        WHERE ID = NEW.producent;
END IF;
END$$
DELIMITER ;
```

Użytkownik z zadania 1 nie może ich stworzyć tych triggerów jednak wykonują się one gdy użytkownik doda lub edytuje wiersz tabeli Aparat.