

Paweł Smolnicki 283999

1

```
create database aparaty;

create user '283999' @'localhost' identified by 'smolnicki99';
GRANT select,
insert,
update on aparaty.* for '283999' @'localhost';
```

2

```
CREATE TABLE IF NOT EXISTS Producent (
    ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nazwa varchar(50) NOT NULL,
    kraj varchar(20) DEFAULT "nieznany",
    adresKorespondencyjny varchar(100) DEFAULT "nieznany"
);

CREATE TABLE IF NOT EXISTS Obiektyw (
    ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    model varchar(30),
    minPrzeslona float,
    maxPrzeslona float,
    CHECK(minPrzeslona < maxPrzeslona)
);

CREATE TABLE IF NOT EXISTS Matryca (
    ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    przekatna decimal(4,2),
    rozdzielczosc decimal(3,1),
    typ varchar(10)
);

ALTER TABLE Matryca AUTO_INCREMENT=100;

CREATE TABLE IF NOT EXISTS Aparat (
    model varchar(30) NOT NULL PRIMARY KEY,
    producent int,
    matryca int,
    obiektyw int,
    waga float NOT NULL DEFAULT 0,
    typ enum('kompaktowy', 'lustrzanka', 'profesjonalny', 'iny'),
    CONSTRAINT `FK_producent` FOREIGN KEY (producent) REFERENCES Producent(ID),
    CONSTRAINT `FK_matryca` FOREIGN KEY (matryca) REFERENCES Matryca(ID),
    CONSTRAINT `FK_obiektyw` FOREIGN KEY (obiektyw) REFERENCES Obiektyw(ID)
);

-- Wartości nieujemne
ALTER TABLE Obiektyw
    ADD CHECK (minPrzeslona >= 0),
    ADD CHECK (maxPrzeslona >= 0);

ALTER TABLE Matryca
    ADD CHECK (przekatna >= 0),
    ADD CHECK (rozdzielczosc >= 0);

ALTER TABLE Aparat
    ADD CHECK (waga >= 0);
```

3

```

INSERT INTO Producent(nazwa, kraj, adresKorespondencyjny) VALUES
('AlphaCorp','Polska','Warszawa 1'),
('BetaGear','Niemcy','Berlin 22'),
('DragonPhoto','Chiny','Shenzhen 5'),
('SunOptics','Chiny','Shanghai 8'),
('RedLeaf','Chiny','Beijing 9'),
('SkyVision','USA','NY 77'),
('NanoLens','Japonia','Tokyo 3'),
('Photonix','Chiny','Guangzhou 11'),
('EuroCam','Francja','Paris 2'),
('OptiMax','Chiny','Hangzhou 6'),
('LiteCam','Hiszpania','Madrid 4'),
('NordCam','Szwecja','Stockholm 10'),
('ClearShot','Czechy','Prague 12'),
('PrimeWorks','Włochy','Rome 13'),
('BrotFrame','Niemcy','Frankfurt 14');

INSERT INTO Obiektyw(model, minPrzeslona, maxPrzeslona) VALUES
('LensA',1.8,16), ('LensB',2.8,22), ('LensC',3.5,18), ('LensD',4,20), ('LensE',2,11),
('LensF',1.4,16), ('LensG',5.6,32), ('LensH',2.2,8), ('LensI',3.2,10), ('LensJ',2.5,12),
('LensK',4.5,22), ('LensL',6.3,36), ('LensM',3.8,14), ('LensN',1.2,16), ('LensO',7.1,42);

INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES
(1.00,12.0,'APS'), (2.50,24.0,'FF'), (1.60,16.0,'APS'), (1.40,10.0,'Micro'),
(2.70,30.0,'FF'), (1.20,8.0,'Micro'), (1.80,20.0,'APS'), (2.40,22.0,'FF'),
(1.30,9.0,'Micro'), (1.70,18.0,'APS'), (2.60,26.0,'FF'), (1.10,6.0,'Micro'),
(2.80,32.0,'FF'), (1.50,14.0,'APS'), (2.20,21.0,'FF');

INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ) VALUES
('ModelA1',1,100,1,450,'lustrzanka'),
('ModelB1',2,101,2,500,'lustrzanka'),
('ModelC1',3,102,3,300,'kompaktowy'),
('ModelD1',4,103,4,1200,'profesjonalny'),
('ModelE1',5,104,5,700,'lustrzanka'),
('ModelF1',6,105,6,550,'innny'),
('ModelG1',7,106,7,480,'lustrzanka'),
('ModelH1',8,107,8,320,'kompaktowy'),
('ModelI1',9,108,9,900,'profesjonalny'),
('ModelJ1',10,109,10,510,'lustrzanka'),
('ModelK1',11,110,11,600,'lustrzanka'),
('ModelL1',12,111,12,430,'kompaktowy'),
('ModelM1',13,112,13,800,'profesjonalny'),
('ModelN1',14,113,14,470,'lustrzanka'),
('ModelO1',15,114,15,490,'lustrzanka');

-- Niedziałające
INSERT INTO Producent(nazwa) VALUES (NULL);
INSERT INTO Aparat VALUES ('Model123', 1, 100, 1, -100, 'cos');
INSERT INTO Matryca(przekatna, rozdzielczosc, typ) VALUES (-1, -1, 'innny');

```

4

1: Żaden model się nie powtarza, bo nazwa dla każdego aparatu jest inna.

2: Nie, brakuje: CREATE ROUTINE

```

DELIMITER $$

CREATE PROCEDURE GenAparaty100()
BEGIN
    DECLARE i INT DEFAULT 1;
    DECLARE p INT;
    DECLARE m INT;
    DECLARE o INT;

    WHILE i <= 100 DO
        SELECT ID INTO p FROM Producent ORDER BY RAND() LIMIT 1;
        SELECT ID INTO o FROM Obiektyw ORDER BY RAND() LIMIT 1;
        SELECT ID INTO m FROM Matryca ORDER BY RAND() LIMIT 1;

        INSERT INTO Aparat(model, producent, matryca, obiektyw, waga, typ)
        VALUES (CONCAT('GenModel_', i), p, m, o, 400 + i, 'kompaktowy');

        SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

```

5

```

DELIMITER $$

CREATE FUNCTION MinMatrycaModel(producentId INT)
RETURNS VARCHAR(30)
DETERMINISTIC
BEGIN
    DECLARE vModel VARCHAR(30);
    SELECT a.model
    INTO vModel
    FROM Aparat a
    JOIN Matryca m ON a.matryca = m.ID
    WHERE a.producent = producentId
    ORDER BY m.przekatna ASC
    LIMIT 1;
    RETURN vModel;
END$$
DELIMITER ;

```

6

```

DELIMITER $$

CREATE TRIGGER AparatAddProducent
BEFORE INSERT ON Aparat
FOR EACH ROW
BEGIN
    IF NEW.producent IS NOT NULL
    AND NOT EXISTS (SELECT 1 FROM Producent WHERE ID = NEW.producent) THEN
        INSERT INTO Producent(ID, nazwa) VALUES (NEW.producent, CONCAT('AutoProd_', NEW.producent));
    END IF;
END$$
DELIMITER ;

```

7

```

DELIMITER $$

CREATE FUNCTION CountModelsByMatryca(matrycaId INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE c INT;
    SELECT COUNT(*) INTO c FROM Aparat WHERE matryca = matrycaId;
    RETURN c;
END$$
DELIMITER ;

```

8

```

DELIMITER $$

CREATE TRIGGER DeleteMatrycaWhenLast
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 ;

```

9

1: Nie, brakuje CREATE VIEW

```

CREATE OR REPLACE VIEW V_LustrzankiSpozaChin AS
SELECT a.model,
       a.waga,
       p.nazwa AS producent,
       m.przekatna,
       m.rozdzielczosc,
       o.minPrzeslona,
       o.maxPrzeslona
FROM Aparat a
JOIN Producent p ON a.producent = p.ID
JOIN Matryca m ON a.matryca = m.ID
JOIN Obiektyw o ON a.obiektyw = o.ID
WHERE a.typ = 'lustrzanka'
AND p.kraj != 'Chiny';

```

10

Odp: Tak, znikły aparaty wyprodukowane przez producentów z Chin (-5).

```

CREATE OR REPLACE VIEW V_ProducentKrajModel AS
SELECT
    p.nazwa as producent,
    p.kraj,
    a.model
FROM Aparat a
JOIN Producent p ON p.ID=a.producent;

DELETE FROM Aparat a
WHERE a.producent=(
    SELECT MIN(ID) FROM Producent p
    WHERE p.kraj='Chiny'
);

```

11

1: Nie, nie może być stworzony.

2: Tak, będzie działał.

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

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

DELIMITER $$

CREATE PROCEDURE IncrementProducentModelCount(prodID INT)
BEGIN
    Update Producent
        SET liczbaModeli = liczbaModeli + 1
    WHERE ID=prodID;
END$$
DELIMITER ;

DELIMITER $$

CREATE PROCEDURE DecrementProducentModelCount(prodID INT)
BEGIN
    Update Producent
        SET liczbaModeli = liczbaModeli - 1
    WHERE ID=prodID;
END$$
DELIMITER ;

DELIMITER $$

CREATE TRIGGER UpdateModelCountOnCreate
BEFORE INSERT ON Aparat
FOR EACH ROW
BEGIN
    IncrementProducentModelCount(NEW.producent);
END$$
DELIMITER ;

DELIMITER $$

CREATE TRIGGER UpdateModelCountonDelete
AFTER DELETE ON Aparat
FOR EACH ROW
BEGIN
    DecrementProducentModelCount(OLD.producent);
END$$
DELIMITER ;

DELIMITER $$

CREATE TRIGGER UpdateModelCountOnChange
AFTER UPDATE ON Aparat
FOR EACH ROW
BEGIN
    IF NEW.producent != OLD.producent THEN
        IncrementProducentModelCount(NEW.producent);
        DecrementProducentModelCount(OLD.producent);
    END
END$$
DELIMITER ;
```