

1.1. Stworzenia bazy danych:

```
01 | CREATE DATABASE IF NOT EXISTS `db-aparaty`;
```

1.2. Stworzenia uzytkownika:

```
01 | CREATE USER IF NOT EXISTS `268414`@`localhost`;
```

1.3. Ustawienie hasla dla uzytkownika:

```
01 | SET PASSWORD FOR `268414`@`localhost` = PASSWORD('karol414');
```

1.4. Ustawienie uprawnień dla uzytkownika:

```
01 | USE `db-aparaty`;  
02 | GRANT SELECT, INSERT, UPDATE ON `db-aparaty`.* TO `268414`@`localhost`;  
03 | FLUSH PRIVILEGES;
```

2.1. Stworzenia tabeli Producent:

```
01 | CREATE TABLE IF NOT EXISTS Producent(  
02 |     ID int AUTO_INCREMENT PRIMARY KEY,  
03 |     nazwa varchar(50),  
04 |     kraj varchar(20)  
05 | );
```

2.2. Stworzenia tabeli Obiektów:

```
01 | CREATE TABLE IF NOT EXISTS Obiektow(  
02 |     ID int AUTO_INCREMENT PRIMARY KEY,  
03 |     model varchar(30) NOT NULL,  
04 |     minPrzeslona float NOT NULL,  
05 |     maxPrzeslona float NOT NULL,  
06 |     CHECK(minPrzeslona >= 0),  
07 |     CHECK(maxPrzeslona >= 0),  
08 |     CHECK(minPrzeslona < maxPrzeslona)  
09 | );
```

2.3. Stworzenia tabeli Matryca:

```
01 | CREATE TABLE IF NOT EXISTS Matryca(  
02 |     ID int AUTO_INCREMENT PRIMARY KEY,  
03 |     przekatna decimal(4,2) NOT NULL,  
04 |     rozdzielczosc decimal(3,1) NOT NULL,  
05 |     typ varchar(10) NOT NULL,  
06 |     CHECK(przekatna >= 0),  
07 |     CHECK(rozdzielczosc >= 0)  
08 | );
```

2.4. Stworzenia tabeli Aparat:

```
01 | CREATE TABLE IF NOT EXISTS Aparat(  
02 |     model varchar(30) PRIMARY KEY,  
03 |     producent int NOT NULL,  
04 |     matryca int NOT NULL,  
05 |     obiektow int NOT NULL,  
06 |     typ ENUM('kompaktowy', 'lustrzanka', 'profesjonalny', 'inny') NOT NULL,  
07 |     FOREIGN KEY(producent) REFERENCES Producent(ID),  
08 |     FOREIGN KEY(matryca) REFERENCES Matryca(ID),  
09 |     FOREIGN KEY(obiektow) REFERENCES Obiektow(ID)  
10 | );
```

3.1. Zalogowanie sie jako nowo stworzony uzytkownik:

```
01 | exit;  
02 | mariadb -u 268414 -p
```

3.2.1. Dodanie poprawnych danych do tabeli Producent:

```
01 | INSERT INTO Producent (nazwa, kraj)  
02 |     VALUE ('Agfa', 'Belgium');  
03 | INSERT INTO Producent (nazwa, kraj)  
04 |     VALUE ('Akaso', 'China');  
05 | INSERT INTO Producent (nazwa, kraj)  
06 |     VALUE ('Canon', 'Japan');  
07 | INSERT INTO Producent (nazwa, kraj)  
08 |     VALUE ('Casio', 'Japan');  
09 | INSERT INTO Producent (nazwa, kraj)  
10 |     VALUE ('Foscam', 'China');  
11 | INSERT INTO Producent (nazwa, kraj)  
12 |     VALUE ('Insta360', 'China');  
13 | INSERT INTO Producent (nazwa, kraj)  
14 |     VALUE ('Kodak', 'USA');  
15 | INSERT INTO Producent (nazwa, kraj)  
16 |     VALUE ('Leica', 'Germany');  
17 | INSERT INTO Producent (nazwa, kraj)  
18 |     VALUE ('Nikolaus', 'China');  
19 | INSERT INTO Producent (nazwa, kraj)  
20 |     VALUE ('Nikon', 'Japan');  
21 | INSERT INTO Producent (nazwa, kraj)  
22 |     VALUE ('Panasonic', 'Japan');  
23 | INSERT INTO Producent (nazwa, kraj)  
24 |     VALUE ('Samsung', 'South Korea');  
25 | INSERT INTO Producent (nazwa, kraj)  
26 |     VALUE ('Sony', 'Japan');  
27 | INSERT INTO Producent (nazwa, kraj)  
28 |     VALUE ('Toshiba', 'Japan');  
29 | INSERT INTO Producent (nazwa, kraj)  
30 |     VALUE ('Xiaoyi', 'China');
```

3.2.2. Proba dodania niepoprawnych danych do tabeli Producent:

```
01 | INSERT INTO Producent (nazwa, kraj)  
02 |     VALUE ('Xiaoyi', 'Peoples Republic of China');
```

3.3.1. Dodanie poprawnych danych do tabeli Obiektyw:

```
01 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
02 |     VALUE ('Garth', 5.0, 15.0);
03 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
04 |     VALUE ('Eugene', 7.0, 12.0);
05 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
06 |     VALUE ('Michael', 10.0, 11.0);
07 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
08 |     VALUE ('Lucy', 1.0, 4.0);
09 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
10 |     VALUE ('Maisie', 9.0, 19.0);
11 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
12 |     VALUE ('Simon', 8.0, 11.0);
13 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
14 |     VALUE ('Roberta', 11.0, 23.0);
15 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
16 |     VALUE ('Tracy', 20.0, 33.0);
17 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
18 |     VALUE ('Jade', 2.0, 9.0);
19 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
20 |     VALUE ('Cecil', 3.0, 12.0);
21 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
22 |     VALUE ('Ned', 5.0, 22.0);
23 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
24 |     VALUE ('Nydia', 7.0, 9.0);
25 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
26 |     VALUE ('Hannibal', 9.0, 10.0);
27 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
28 |     VALUE ('Baz', 6.0, 13.0);
29 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
30 |     VALUE ('Dean', 2.0, 9.0);
```

3.3.2. Proba dodania niepoprawnych danych do tabeli Obiektyw:

```
01 | INSERT INTO Obiektyw (model, minPrzeslona, maxPrzeslona)
02 |     VALUE ('Eileen', 7.0, 5.0);
```

3.4.1. Dodanie poprawnych danych do tabeli Matryca:

```
01 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
02 |     VALUE (10.0, 12.0, 'Jehan');
03 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
04 |     VALUE (9.0, 14.0, 'Yvette');
05 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
06 |     VALUE (13.0, 14.0, 'Jehan');
07 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
08 |     VALUE (5.0, 16.0, 'Cerise');
09 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
10 |     VALUE (8.0, 11.0, 'Helewise');
11 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
12 |     VALUE (11.0, 11.0, 'Yvette');
13 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
14 |     VALUE (9.0, 15.0, 'Helewise');
15 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
16 |     VALUE (14.0, 15.0, 'Jehan');
17 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
18 |     VALUE (3.0, 13.0, 'Helewise');
19 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
20 |     VALUE (19.0, 13.0, 'Cerise');
21 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
22 |     VALUE (9.0, 12.0, 'Helewise');
23 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
24 |     VALUE (2.0, 16.0, 'Yvette');
25 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
26 |     VALUE (4.0, 13.0, 'Helewise');
27 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
28 |     VALUE (6.0, 14.0, 'Cerise');
29 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
30 |     VALUE (9.0, 15.0, 'Jehan');
```

3.4.2. Proba dodania niepoprawnych danych do tabeli Matryca:

```
01 | INSERT INTO Matryca (przekatna, rozdzielczosc, typ)
02 |     VALUE (-8.0, 15.0, 'Jehan');
```

3.5.1. Dodanie poprawnych danych do tabeli Aparat:

```
01 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
02 |     VALUE ('Jata', 1, 103, 7, 'kompaktowy');
03 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
04 |     VALUE ('Banou', 2, 105, 8, 'lustrzanka');
05 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
06 |     VALUE ('Muraty', 1, 101, 9, 'profesjonalny');
07 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
08 |     VALUE ('Jabali', 7, 101, 1, 'kompaktowy');
09 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
10 |     VALUE ('Lakicia', 10, 104, 6, 'lustrzanka');
11 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
12 |     VALUE ('Siwatu', 12, 109, 6, 'profesjonalny');
13 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
14 |     VALUE ('Winda', 11, 108, 13, 'inny');
15 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
16 |     VALUE ('Bina', 4, 105, 7, 'inny');
17 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
18 |     VALUE ('Mosi', 3, 104, 10, 'lustrzanka');
19 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
20 |     VALUE ('Kudio', 5, 102, 1, 'lustrzanka');
21 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
22 |     VALUE ('Dalila', 8, 103, 2, 'profesjonalny');
23 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
24 |     VALUE ('Jaleel', 7, 106, 4, 'kompaktowy');
25 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
26 |     VALUE ('Kapuki', 10, 104, 15, 'inny');
27 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
28 |     VALUE ('Hashaan', 15, 103, 11, 'kompaktowy');
29 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
30 |     VALUE ('Barika', 14, 101, 2, 'lustrzanka');
```

3.5.2. Proba dodania niepoprawnych danych do tabeli Aparat:

```
01 | INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
02 |     VALUE ('Khary', 5, 106, 7, 'the best');
```

4. Tworzenie procedury generującej 100 modeli Aparat:

```

01 | DELIMITER $$
02 | CREATE PROCEDURE IF NOT EXISTS Add100Aparats()
03 | BEGIN
04 |     DECLARE licznik int;
05 |     DECLARE losowaMatryca int;
06 |     DECLARE losowyObiektow int;
07 |     DECLARE losowyProducent int;
08 |     DECLARE losowyTyp varchar(20);
09 |     DECLARE justAdded int;
10 |
11 |     SET licznik = 0;
12 |     WHILE licznik < 100 DO
13 |         SELECT ID INTO losowaMatryca FROM Matryca ORDER BY RAND() LIMIT 1;
14 |         SELECT ID INTO losowyObiektow FROM Obiektow ORDER BY RAND() LIMIT 1;
15 |         SELECT ID INTO losowyProducent FROM Producent ORDER BY RAND() LIMIT 1;
16 |         SELECT (SELECT COUNT(*) FROM Aparat) INTO justAdded FROM Aparat;
17 |         SET losowyTyp = ELT(FLOOR(RAND()*4)+1,
18 |             'kompaktowy', 'lustrzanka', 'profesjonalny', 'inny');
19 |         INSERT INTO Aparat (model, producent, matryca, obiektow, typ)
20 |             VALUE(CONCAT('RandomModel-', licznik + justAdded,
21 |                 losowyProducent, losowaMatryca, losowyObiektow, losowyTyp);
22 |         SET licznik = licznik + 1;
23 |     END WHILE;
24 | END;$$
25 | DELIMITER ;

```

- Procedury nie można stworzyć ani wywołać z poziomu stworzonego użytkownika, ponieważ nie ma przyznanych odpowiednich uprawnień.
- Procedura stworzyła 100 różnych modeli aparatów.

5. Tworzenie funkcji zwracającej model aparatu producenta o podanym ID z największą przekatną matrycy:

```

01 | DELIMITER $$
02 | CREATE FUNCTION IF NOT EXISTS maxMatrix(producentID int) RETURNS varchar(30)
03 | BEGIN
04 |     DECLARE modelName varchar(30);
05 |
06 |     SELECT model INTO modelName
07 |     FROM Aparat JOIN Matryca ON matryca = Matryca.ID
08 |     WHERE producent = producentID ORDER BY przekatna DESC LIMIT 1;
09 |
10 |     RETURN modelName;
11 | END$$
12 | DELIMITER ;

```

6. Tworzenie triggera do wstawiania Aparatu o dowolnym Producencie:

```

01 | DELIMITER $$
02 | CREATE TRIGGER IF NOT EXISTS anyProducent BEFORE INSERT ON Aparat
03 | FOR EACH ROW
04 | BEGIN
05 |     IF (SELECT COUNT(*) FROM Producent WHERE ID = NEW.producent) = 0
06 |     THEN INSERT INTO Producent (ID) VALUE(NEW.producent);

```

```
07 |      END IF;  
08 | END$$  
09 | DELIMITER ;
```

7. Tworzenie funkcji do liczenia Aparatów o podanej Matrycy:

```

01 | DELIMITER $$
02 | CREATE FUNCTION IF NOT EXISTS countAparatsWithMatrix(matrixID int) RETURNS int
03 | BEGIN
04 |     DECLARE aparatsCount int;
05 |     SELECT COUNT(*) INTO aparatsCount FROM Aparat
06 |     WHERE matryca = matrixID;
07 |     RETURN aparatsCount;
08 | END$$
09 | DELIMITER ;

```

8. Tworzenie triggera do usuwania Matrycy gdy zaden Aparat jej nie wykorzystuje:

```

01 | DELIMITER $$
02 | CREATE TRIGGER IF NOT EXISTS removeUnusedMatrix AFTER DELETE ON Aparat
03 | FOR EACH ROW
04 | BEGIN
05 |     DELETE FROM Matryca
06 |     WHERE ID = OLD.matryca
07 |     AND (SELECT COUNT(*) FROM Aparat WHERE matryca = ID) = 0;
08 | END$$
09 | DELIMITER ;

```

9. Tworzenie widoku:

```

01 | CREATE VIEW IF NOT EXISTS AparatLustrzankaNieChinskiProducent
02 | AS SELECT Aparat.model, Producent.nazwa, Matryca.przekatna,
03 |     Obiektow.minPrzeslona, Obiektow.maxPrzeslona
04 | FROM Aparat
05 |     JOIN Matryca ON matryca = Matryca.ID
06 |     JOIN Obiektow ON obiektow = Obiektow.ID
07 |     JOIN Producent ON producent = Producent.ID
08 | WHERE Aparat.typ = 'lustrzanka' AND kraj <> 'China';

```

- Widoku nie można stworzyć z poziomu stworzonego użytkownika, ponieważ nie ma przyznanych odpowiednich uprawnień.

10.1. Tworzenie widoku:

```

01 | CREATE VIEW IF NOT EXISTS KrajeAparatow
02 | AS SELECT Aparat.model, Producent.kraj
03 | FROM Aparat JOIN Producent ON Aparat.producent = Producent.ID;

```

10.2 Wyświetlenie rekordów widoku:

```
01 | SELECT * FROM KrajeAparatow;
```

10.3 Usuwanie rekordów z których korzysta widok:

```

01 | DELETE FROM Aparat
02 | WHERE producent IN (SELECT ID FROM Producent WHERE kraj = 'China');

```

10.4 Wyświetlenie rekordów widoku:

```
01 | SELECT * FROM KrajeAparatow;
```

- Z widoku zostały usunięte rekordy w których krajem były 'China'

11.1. Dodawanie kolumny liczbaModeli do tabeli Producent:

```
01 | ALTER TABLE Producent ADD COLUMN IF NOT EXISTS liczbaModeli int NOT NULL;
```

11.2. Ustawienie wartosci kolumnie liczbaModeli:

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

11.3. Dodawanie trigera wywoływanego przy dodaniu Produktu:

```
01 | DELIMITER $$  
02 | CREATE TRIGGER IF NOT EXISTS AparatInserted AFTER INSERT ON Aparat  
03 | FOR EACH ROW  
04 | BEGIN  
05 |     UPDATE Producent SET liczbaModeli =  
06 |         (SELECT COUNT(*) FROM Aparat WHERE producent = Producent.ID);  
07 | END$$  
08 | DELIMITER ;
```

11.4. Dodawanie trigera wywoływanego przy usunięciu Produktu:

```
01 | DELIMITER $$  
02 | CREATE TRIGGER IF NOT EXISTS AparatDeleted AFTER DELETE ON Aparat  
03 | FOR EACH ROW  
04 | BEGIN  
05 |     UPDATE Producent SET liczbaModeli =  
06 |         (SELECT COUNT(*) FROM Aparat WHERE producent = Producent.ID);  
07 | END$$  
08 | DELIMITER ;
```

11.5. Dodawanie trigera wywoływanego przy zmianie Producenta Produktu:

```
01 | DELIMITER $$  
02 | CREATE TRIGGER IF NOT EXISTS AparatUpdated AFTER UPDATE ON Aparat  
03 | FOR EACH ROW  
04 | BEGIN  
05 |     UPDATE Producent SET liczbaModeli =  
06 |         (SELECT COUNT(*) FROM Aparat WHERE producent = Producent.ID);  
07 | END$$  
08 | DELIMITER ;
```

- Trigery nie mogą zostać utworzone przez utworzonego użytkownika, ponieważ nie posiada on uprawnień.
- Trigery będą wywoływane przy akcjach utworzonego użytkownika.