

1.1 CREATE DATABASE db_aparaty;

1.2 CREATE USER '268451'@localhost IDENTIFIED BY 'Maksymilian451';

1.3 GRANT GRANT SELECT, INSERT, ALTER ON *.* TO '268451'@localhost IDENTIFIED BY PASSWORD 'Maksymilian451';

2.1 CREATE TABLE `aparat` (

 `model` varchar(30) NOT NULL,

 `producent` int(11) NOT NULL,

 `matryca` int(11) NOT NULL,

 `obiekt` int(11) NOT NULL,

 `typ` enum('kompaktowy','lustrzanka','profesjonalny','inny') NOT NULL,

 PRIMARY KEY (`model`),

 KEY `producent` (`producent`),

 KEY `matryca` (`matryca`),

 KEY `obiekt` (`obiekt`),

 CONSTRAINT `aparat_ibfk_1` FOREIGN KEY (`producent`) REFERENCES `producent` (`ID`),

 CONSTRAINT `aparat_ibfk_2` FOREIGN KEY (`matryca`) REFERENCES `matryca` (`ID`),

 CONSTRAINT `aparat_ibfk_3` FOREIGN KEY (`obiekt`) REFERENCES `obiekt` (`ID`),

 CONSTRAINT `CONSTRAINT_1` CHECK (`matryca` >= 0 and `obiekt` >= 0 and `producent` >= 0)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

2.2 CREATE TABLE `matryca` (

 `ID` int(11) NOT NULL AUTO_INCREMENT,

 `przekatna` decimal(4,2) NOT NULL,

 `rozdzielczosc` decimal(3,1) NOT NULL,

 `typ` varchar(10) NOT NULL,

 PRIMARY KEY (`ID`),

 CONSTRAINT `CONSTRAINT_1` CHECK (`przekatna` >= 0 and `rozdzielczosc` >= 0)

) ENGINE=InnoDB AUTO_INCREMENT=115 DEFAULT CHARSET=utf8mb3;

2.3 CREATE TABLE `obiekt` (

 `ID` int(11) NOT NULL AUTO_INCREMENT,

 `model` varchar(30) NOT NULL,

 `minPrzeslona` float NOT NULL,

```

`maxPrzeslona` float NOT NULL,

PRIMARY KEY (`ID`),

CONSTRAINT `CONSTRAINT_1` CHECK (`minPrzeslona` >= 0 and `maxPrzeslona` >= 0 and
`minPrzeslona` < `maxPrzeslona`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

```

2.4 CREATE TABLE `producent` (

```

`ID` int(11) NOT NULL AUTO_INCREMENT,

`nazwa` varchar(50) DEFAULT NULL,

`kraj` varchar(20) DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8mb3;

```

3.1 INSERT INTO `matryca` VALUES

```

(100,28.21,26.1,'CMOS'),(101,28.26,24.2,'CMOS'),(102,42.82,24.2,'CMOS'),(103,43.13,33.0,'CMOS'),(
104,42.82,24.2,'CMOS'),(105,26.82,24.2,'CMOS'),(106,28.37,6.1,'CCD'),(107,43.27,24.1,'CMOS'),(108,
26.82,18.0,'CMOS'),(109,7.70,16.0,'CMOS'),(110,7.70,16.0,'CCD'),(111,7.54,8.0,'CCD'),(112,11.00,12.0
,'CMOS'),(113,28.40,16.3,'CMOS'),(114,11.00,12.0,'CMOS');

```

3.2 INSERT INTO `producent` VALUES (1,'DJI','Chiny'),(2,'Aigo','Chiny'),(3,'Seagull

```

Camera','Chiny'),(4,'Akaso','Chiny'),(5,'Insta360','Chiny'),(6,'Leica','Niemcy'),(7,'Canon','Japonia'),(8,'F
ujifilm','Japonia'),(9,'Nikon','Japonia'),(10,'Casio','Japonia'),(11,'Hasselblad','Szwecja'),(12,'Sony','Japo
nia'),(13,'Blackmagic','Australia'),(14,'Samsung','Korea Południowa'),(15,'Polaroid','USA');

```

3.3 INSERT INTO obiektyw VALUES (1, "Canon EF USM Fisheye", 4.0, 22.0), (2, "Canon EF-S IS STM", 5.6, 22.0), (3, "Canon EF-S USM", 3.5, 22.0), (4, "Sony AF DT D", 4.5, 22.0), (5, "Sony E OSS", 4.0, 22.0), (6, "Sony FE GM", 2.8, 22.0), (7, "Sony DT SSM", 2.8, 22.0), (8, "Sony AF DT", 3.5, 22.0), (9, "Leica D Vario-Elmarit", 2.8, 22.0), (10, "Leica Super-Elmarit-R", 2.8, 22.0), (11, "Leica Elmarit-M", 2.8, 16.0), (12, "Nikon AF-S Nikkor", 3.5, 22.0), (13, "Nikon 1 Nikkor VR", 4.5, 16.0), (14, "Carl Zeiss Distagon", 3.5, 22.0), (15, "Olympus ZUIKO DIGITAL ED", 4.0, 22.0);

4. Użytkownik może

```

DELIMITER //

```

```

CREATE PROCEDURE gen_aparaty ()

```

```

BEGIN

```

```

INSERT INTO aparat (model, producent, matryca, obiektyw, typ) SELECT CONCAT( p.ID, ".", m.ID, ".",
o.ID), p.ID, m.ID, o.ID, ELT(floor(RAND()*4)+1, 'kompaktowy', 'lustrzanka', 'profesjonalny', 'inny') AS
typ FROM producent AS p, matryca AS m, obiektyw AS o ORDER BY RAND() LIMIT 100;

```

```

END;

```

```

//

```

DELIMITER ;

5.

DELIMITER //

CREATE PROCEDURE biggestmodel (IN prodID INT)

BEGIN

SELECT model FROM aparat AS a JOIN matryca AS m ON a.matryca = m.ID WHERE producent = prodID HAVING MAX(przekatna);

END;

//

DELIMITER ;

6. CREATE TRIGGER no_producent BEFORE INSERT ON aparat FOR EACH ROW INSERT INTO producent (ID) VALUES (NEW.producent) ON DUPLICATE KEY UPDATE nazwa = nazwa, kraj = kraj, liczbaModeli = liczbaModeli;

7. CREATE FUNCTION ile_modeli_do_matrycy (id_matrycy INT) RETURNS INT RETURN (SELECT COUNT(*) AS ilosc FROM aparat GROUP BY matryca HAVING matryca = id_matrycy);

8. CREATE TRIGGER no_more_aparat_for_matryca AFTER DELETE ON aparat FOR EACH ROW DELETE FROM matryca WHERE ID = OLD.matryca AND 0 = (SELECT COUNT(*) FROM aparat GROUP BY matryca HAVING matryca = OLD.matryca);

9. nie może

CREATE VIEW not_chinese_dslr AS SELECT a.model, nazwa, przekatna, rozdzielczosc, minPrzeslona, maxPrzeslona FROM aparat AS a JOIN matryca AS m ON a.matryca = m.ID JOIN producent AS p ON a.producent = p.ID JOIN obiektyw AS o ON a.obiektyw = o.ID WHERE a.typ = "lustrzanka" AND p.kraj <> "Chiny";

10.1. CREATE VIEW zad10 AS SELECT model, nazwa, kraj FROM aparat AS a JOIN producent AS p ON a.producent = p.ID; - Nie może użytkownik widok się zaktualizować

11.1 ALTER TABLE producent ADD liczbaModeli INT;

11.2 INSERT INTO producent (id, nazwa, kraj, liczbaModeli) SELECT ID, nazwa, kraj, COUNT(*) AS liczbaModeli FROM aparat AS a JOIN producent AS p ON a.producent = p.ID GROUP BY producent ON DUPLICATE KEY UPDATE nazwa=VALUES(nazwa), kraj = VALUES(kraj), liczbaModeli=VALUES(liczbaModeli);

11.3 CREATE TRIGGER ins_aparat_up_prod AFTER INSERT ON aparat FOR EACH ROW INSERT INTO producent (id, nazwa, kraj, liczbaModeli) SELECT ID, nazwa, kraj, COUNT(*) AS liczbaModeli FROM aparat AS a JOIN producent AS p ON a.producent = p.ID GROUP BY producent ON DUPLICATE KEY UPDATE nazwa=VALUES(nazwa), kraj = VALUES(kraj), liczbaModeli=VALUES(liczbaModeli);

11.4 CREATE TRIGGER up_aparat_up_prod AFTER UPDATE ON aparat FOR EACH ROW INSERT INTO producent (id, nazwa, kraj, liczbaModeli) SELECT ID, nazwa, kraj, COUNT(*) AS liczbaModeli FROM aparat AS a JOIN producent AS p ON a.producent = p.ID GROUP BY producent ON DUPLICATE KEY UPDATE nazwa=VALUES(nazwa), kraj = VALUES(kraj), liczbaModeli=VALUES(liczbaModeli);

11.5 CREATE TRIGGER del_aparat_up_prod AFTER DELETE ON aparat FOR EACH ROW INSERT INTO producent (id, nazwa, kraj, liczbaModeli) SELECT ID, nazwa, kraj, COUNT(*) AS liczbaModeli FROM aparat AS a JOIN producent AS p ON a.producent = p.ID GROUP BY producent ON DUPLICATE KEY UPDATE nazwa=VALUES(nazwa), kraj = VALUES(kraj), liczbaModeli=VALUES(liczbaModeli);

11.odp nie może stworzyć bo nie ma update privilage, i nie będą dla niego działać