

Bazy Danych i Systemy Informacyjne

Laboratorium: sprawozdanie z listy 2

1. Wprowadzenie

1.1. Cel zadań

Polecenia z listy 2. sprawdzają i pozwalają na zapoznanie się z podstawami:

- tworzenia baz danych i dodawania tabel,
- generowania rekordów do tabel (sekwencyjnie lub pseudolosowo),
- tworzenia użytkowników i przyznawania im uprawnień,
- tworzenia i korzystania z procedur, funkcji i triggerów (wyzwalaczy),
- tworzenia widoków i ich własnościami.

1.2. Środowisko SQL wykorzystane w zadaniach

W rozwiązywaniu zadań wykorzystano darmowe zintegrowane środowisko programistyczne MySQL, w którego skład wchodzi:

- MySQL Workbench 8.0 – wizualne, programistyczne narzędzie do projektowania baz danych zawierające pole do wprowadzania zapytań w składni MySQL (wszystkie zadanie zostały rozwiązane z pomocą kwerend pisanych w języku SQL)
- MySQL Server – oprogramowanie służące za serwer bazodanowy, umożliwia połączenie programu MySQL Workbench z bazami danych

2. Opis rozwiązań poleceń

Uwaga – w rezultatach często są pokazane tylko fragmenty wypisanych rekordów ze względu na ich dużą liczbę.

2.1. Zadanie 1

2.1.1. Polecenie

Utwórz bazę danych db-aparaty. Utwórz użytkownika o nazwie będącej twoim numerem albumu. Ustaw dla niego hasło, będące konkatencją twojego imienia i trzech ostatnich cyfr numeru albumu. Nadaj utworzonemu użytkownikowi uprawnienia do selectowania, wstawiania i zmieniania danych w tabelach, jednak nie do tworzenia, usuwania i modyfikowania tabel w bazie. (1pkt).

2.1.2. Kwerendy niezbędne do realizacji

```
1 CREATE DATABASE IF NOT EXISTS `db-aparaty`;
```

```
1 • CREATE USER IF NOT EXISTS `261750` PASSWORD EXPIRE NEVER;
```

```
1 • SET PASSWORD FOR '261750' = 'Marek750';
```

```
1 • GRANT DELETE, INSERT, SELECT, UPDATE ON *.* TO `261750`;
```

```
1 • SELECT * FROM mysql.user;
```

(do sprawdzenia rezultatu)

2.1.3. Wynik

SCHEMAS

Filter objects

- db-aparaty
- sakila
- sys
- world

	Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Reload_priv	Shutdown_priv	Process_priv	File_priv	Grant_priv	References_priv	Index_priv	Alter_priv	Show_priv
▶	%	261750	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N
	localhost	mysql.infoschema	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	localhost	mysql.session	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N
	localhost	mysql.sys	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	localhost	root	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
•	mysql	mysql	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE

2.1.4. Opis rozwiązania

Na początku tworzę bazę danych *db-aparaty*. Korzystam z wyrażenia CREATE DATABASE (alternatywnie CREATE SCHEMA). IF NOT EXISTS pozwala na pozbycie się błędu w przypadku, gdyby taka baza już istniała (w przypadku tego zadania wprowadzone z przyzwyczajenia). Istotne tutaj jest wzięcie nazwy w ``; w przeciwnym razie wystąpiłby błąd składni – myślniki w nazwach są niewskazane.

Kolejna kwerenda tworzy użytkownika z użyciem wyrażenia CREATE USER (analogicznie jak przy tworzeniu bazy danych: jeśli nie istnieje, IF NOT EXISTS). Jego nazwa to numer albumu (skrótowy) oraz dodatkowo określiłem, że hasło nigdy nie wygaśnie (PASSWORD EXPIRE NEVER).

Ustawiam hasło (SET PASSWORD) dla (FOR) nowego użytkownika zgodnie z treścią zadania.

Poprzez słowo kluczowe GRANT przyznaję użytkownikowi prawa. Domyślnie nowy użytkownik nie ma żadnych praw (przynajmniej w MySQL), zatem wystarczy nadać mu potrzebne prawa do selectowania (SELECT), wstawiania rekordów (INSERT), usuwania rekordów (DELETE), modyfikowania istniejących rekordów (UPDATE) w tabeli. Innych praw użytkownik nie ma, więc nie może zmieniać struktury tabel, usuwać ich bądź tworzyć nowe – upewniam się w tym wyświetlając tabelę *mysql.user* przechowującą użytkowników z ich prawami.

2.2. Zadanie 2

2.2.1. Polecenie

Utwórz tabele o schematach:

(a) Aparat (model: varchar(30), producent: int, matryca: int, obiektyw: int, typ: enum('kompaktowy', 'lustrzanka', 'profesjonalny', 'inny'))

(b) Matryca (ID: int, przekatna: decimal(4,2), rozdzielczosc: decimal(3,1) typ: varchar(10))

(c) Obiektyw (ID: int, model: varchar(30), minPrzeslona: float, maxPrzeslona: float)

(d) Producent (ID: int, nazwa: varchar(50), kraj: varchar(20))

Zadbaj o to, by podkreślone atrybuty były kluczami głównymi, tam gdzie to możliwe zastosuj automatyczną inkrementację, w przypadku tabeli Matryca zacznij od *ID* o wartości 100. W tabeli Aparat zadbaj by odpowiednie identyfikatory były kluczami obcymi. Dodatkowo zadbaj, aby wprowadzane dane liczbowe nie miały wartości ujemnych a $minPrzeslona < maxPrzeslona$. Nazwa oraz kraj z siedzibą główną producenta mogą być puste w przypadku braku danych. (2pkt)

2.2.2. Kwerendy niezbędne do realizacji

```
1 CREATE TABLE IF NOT EXISTS Matryca (  
2     ID INT NOT NULL AUTO_INCREMENT,  
3     przekatna DECIMAL(4 , 2) NOT NULL,  
4     rozdzielczosc DECIMAL(3 , 1) NOT NULL,  
5     typ VARCHAR(10) NOT NULL,  
6     PRIMARY KEY (ID),  
7     CONSTRAINT nieujemne CHECK (przekatna >= 0 AND rozdzielczosc >= 0)  
8 )
```

```
9     AUTO_INCREMENT = 100;
```

```
1 CREATE TABLE IF NOT EXISTS Obiektyw (  
2     ID INT NOT NULL AUTO_INCREMENT,  
3     model VARCHAR(30) NOT NULL,  
4     minPrzeslona FLOAT NOT NULL,  
5     maxPrzeslona FLOAT NOT NULL,  
6     PRIMARY KEY (ID),  
7     CONSTRAINT nieujemne_obiektyw CHECK (minPrzeslona >= 0 AND maxPrzeslona >= 0),  
8     CONSTRAINT maxwiekszeodmin CHECK (minPrzeslona < maxPrzeslona)  
9 );
```

```
1 CREATE TABLE IF NOT EXISTS Producent (  
2     ID INT NOT NULL AUTO_INCREMENT,  
3     nazwa VARCHAR(50),  
4     kraj VARCHAR(20),  
5     PRIMARY KEY (ID) );
```

```

1 • CREATE TABLE IF NOT EXISTS Aparat (
2     model VARCHAR(30) NOT NULL,
3     producent INT NOT NULL,
4     matryca INT NOT NULL,
5     obiekttyw INT NOT NULL,
6     typ ENUM('kompaktowy', 'lustrzanka', 'profesjonalny', 'inny') NOT NULL,
7     PRIMARY KEY (model),
8     FOREIGN KEY (producent)
9         REFERENCES Producent (ID),
10    FOREIGN KEY (matryca)
11        REFERENCES Matryca (ID),
12    FOREIGN KEY (obiektyw)
13        REFERENCES Obiektyw (ID),
14    CONSTRAINT dodatnie_id CHECK (producent > 0 AND matryca > 0
15        AND obiekttyw > 0)
16 );

```

2.2.3. Rezultat

Info	Tables	Columns	Indexes	Triggers	Views	Stored Procedures	Functions	Grants	Events
Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data		
apar	InnoDB	10	Dynamic	15	1092	16.0 KiB			
matryca	InnoDB	10	Dynamic	15	1092	16.0 KiB			
obiektyw	InnoDB	10	Dynamic	15	1092	16.0 KiB			
producent	InnoDB	10	Dynamic	15	1092	16.0 KiB			

2.2.4. Opis rozwiązania

Tabelę z podpunktu a) muszę utworzyć jako ostatnią, ponieważ w niej znajdują się klucze obce odnoszące się do pozostałych tabel. Reszta kolejności jest bez znaczenia.

Korzystam ze składni CREATE TABLE IF NOT EXISTS aby stworzyć tablicę tylko w przypadku, jeśli nie istnieje.

W nawiasie zaczynam od wymienienia po przecinku kolumn zgodnych z treścią zadania. Do nazw dopisane są typy (np. VARCHAR(n) to ciąg znaków o długości n, DECIMAL (s,p) to liczba zmiennoprzecinkowa z maksymalnie s cyframi przed przecinkiem i p cyframi po przecinku, ENUM to typ wyliczeniowy, który może przyjąć tylko podane wartości) oraz, tam gdzie potrzebne, określenie, że kolumna nie przyjmuje wartości NULL a także automatyczne inkrementowanie liczbowych kluczy głównych.

Po określeniu kolumn zapisuję, które z nich są kluczami głównymi składnią PRIMARY KEY (nazwa) oraz kluczami obcymi składnią FOREIGN KEY (nazwa) REFERENCES Tablica_do_której_nawiązuje_klucz (nazwa_kolumny_do_której_nawiązuje_klucz).

Na samym końcu nawiasu okrągłego dodaję warunki (CONSTRAINT nazwa CHECK (warunek)), jakie muszą spełniać wartości dodawane do kolumn: wartości liczbowe nieujemne, maksymalna przesłona większa od minimalnej oraz wartości id dodawanych rekordów muszą być dodatnie (dla wygody, nie jest to konieczne teraz).

W tabeli matryca dodaję jeszcze poza nawiasem wartość początkową ID równą 100 (AUTO_INCREMENT = 100).

2.3. Zadanie 3

2.3.1. Polecenie

Połącz się z bazą przy pomocy konta użytkownika utworzonego w zadaniu 1. Dodaj do każdej z tabel po 15 rekordów (do tabeli Producenci koniecznie dodaj 5 pozycji z producentami, których kraj siedziby to Chiny). Dodatkowo dla każdej z tabel spróbuj wprowadzić rekordy naruszające ograniczenia z zadania 2. (1pkt)

2.3.2. Kwerendy niezbędne do realizacji

```
1 • INSERT INTO matryca (przekatna, rozdzielczosc, typ)
2   VALUES
3     (5.68, 5.0, '1/3.2"),
4     (5.68, 10.0, '1/3.2" HQ'),
5     (6.00, 8.5, '1/3.0"),
6     (6.86, 12.5, '1/2.6"),
7     (7.18, 10.0, '1/2.5"),
8     (7.18, 14.3, '1/2.5"+'),
9     (7.66, 15.0, '1/2.3"),
10    (11.00, 9.0, '2/3"),
11    (12.7, 18.0, 'Std 16mm'),
12    (15.86, 20.0, '1" Type'),
13    (21.60, 23.0, 'Micro 4/3"),
14    (43.20, 45.0, '35mm FF'),
15    (43.30, 56.7, '35mm FF+'),
16    (67.40, 70.9, 'Phase One'),
17    (87.91, 99.4, 'IMAX FF');

1 • INSERT INTO obiektyw (model, minPrzeslona, maxPrzeslona)
2   VALUES
3     ('Nikkor AF-P', 3.5, 5.6),
4     ('Sigma BCX', 2.85, 6.3),
5     ('Sigma A2', 2.5, 4.5),
6     ('Sigma A3-PwM', 3.7, 8.0),
7     ('Optica CG-X5', 3.4, 6.5),
8     ('Nikkor S68', 4.0, 5.6),
9     ('Fujinon XF', 2.8, 5.0),
10    ('Tokina AT-XD', 4.0, 5.7),
11    ('Fujinon XG', 3.5, 5.8),
12    ('Leica AQ', 3.0, 5.25),
13    ('Optica CG-X6', 3.0, 6.5),
14    ('Nikkor KIT', 4.0, 4.5),
15    ('Fujimax F2', 3.4, 7.9),
16    ('Agpha Alpha', 3.5, 5.6),
17    ('Billiga S5', 5.0, 5.9);
```

```
1 • INSERT INTO producent (nazwa, kraj)
2   VALUES
3     ('Aiigo', 'Chiny'),
4     ('Agpha', 'Niemcy'),
5     ('BingWang', 'Chiny'),
6     ('Cannon', 'Japonia'),
7     ('CasiJo', 'Japonia'),
8     ('DJJ', 'Chiny'),
9     ('FusCam', 'Chiny'),
10    ('Kokodak', 'Japonia'),
11    ('Lejka', 'Niemcy'),
12    ('Memento', 'Szwecja'),
13    ('PingZhang', 'Chiny'),
14    ('Państwowa Wytwórnia Optyki', 'Polska'),
15    ('Sams', 'Korea Południowa'),
16    ('ToShibaInu', 'Japonia'),
17    ('Weiss', 'Austria');

1 • INSERT INTO aparat
2   VALUES
3     ('AXBPro', 3, 105, 1, 'kompaktowy'),
4     ('EKVp3', 6, 112, 3, 'kompaktowy'),
5     ('MG53-U', 9, 114, 13, 'profesjonalny'),
6     ('GKR55', 2, 106, 1, 'lustrzanka'),
7     ('BRMOV5', 14, 113, 6, 'kompaktowy'),
8     ('56FCGBR', 6, 108, 12, 'inny'),
9     ('Go Quad X', 3, 105, 9, 'inny'),
10    ('CX-5-F', 7, 107, 7, 'profesjonalny'),
11    ('G3S', 14, 102, 8, 'kompaktowy'),
12    ('STSEWTB', 15, 111, 13, 'inny'),
13    ('EOEMB3', 10, 101, 1, 'profesjonalny'),
14    ('K4K3K2', 13, 102, 1, 'profesjonalny'),
15    ('MaxCap-3XCF', 7, 103, 8, 'profesjonalny'),
16    ('UltraLens G5', 11, 110, 6, 'lustrzanka'),
17    ('Basic Q2M', 1, 102, 1, 'kompaktowy');
```

Próby naruszenia ograniczeń:

```
1   INSERT INTO matryca (przekatna, rozdzielczosc, typ)
2   VALUES
3     (5.68, -5.0, 'test');

1   INSERT INTO obiektyw (model, minPrzeslona, maxPrzeslona)
2   VALUES
3     ('Nikkor AF-P', 9.5, 5.6);
```

```

1  INSERT INTO aparat
2  VALUES
3  ('AXBPro', -6, 105, -1, 'kompaktowy')j

```

```

1  •  INSERT INTO producent
2  VALUES
3  ('test', 'Aiigo', 'Chiny')

```

2.3.3. Rezultat

ID	przekatna	rozdzielczosc	typ
100	5.68	5.0	1/3.2"
101	5.68	10.0	1/3.2" HQ
102	6.00	8.5	1/3.0"
103	6.86	12.5	1/2.6"
104	7.18	10.0	1/2.5"
105	7.18	14.3	1/2.5"+
106	7.66	15.0	1/2.3"
107	11.00	9.0	2/3"
108	12.70	18.0	Std 16mm
109	15.86	20.0	1" Type
110	21.60	23.0	Micro 4/3"
111	43.20	45.0	35mm FF
112	43.30	56.7	35mm FF+
113	67.40	70.9	Phase One
114	87.91	99.4	IMAX FF
NULL	NULL	NULL	NULL

ID	model	minPrzeslona	maxPrzeslona
1	Nikkor AF-P	3.5	5.6
2	Sigma BCX	2.85	6.3
3	Sigma A2	2.5	4.5
4	Sigma A3-PWM	3.7	8
5	Optica CG-X5	3.4	6.5
6	Nikkor SG8	4	5.6
7	Fujinon XF	2.8	5
8	Tokina AT-XD	4	5.7
9	Fujinon XG	3.5	5.8
10	Leica AQ	3	5.25
11	Optica CG-X6	3	6.5
12	Nikkor KIT	4	4.5
13	Fujimax F2	3.4	7.9
14	Agpha Alpha	3.5	5.6
15	Billiga S5	5	5.9
NULL	NULL	NULL	NULL

ID	nazwa	kraj
1	Aiigo	Chiny
2	Agpha	Niemcy
3	BingWang	Chiny
4	Cannon	Japonia
5	Casijo	Japonia
6	DJJ	Chiny
7	FusCam	Chiny
8	Kokodak	Japonia
9	Lejka	Niemcy
10	Memento	Szwecja
11	PingZhang	Chiny
12	Państwowa Wytwórnia Optyki	Polska
13	Sams	Korea ...
14	ToShibaInu	Japonia
15	Weiss	Austria
NULL	NULL	NULL

model	producent	matryca	obiektyw	typ
56FCGBR	6	108	12	inny
AXBPro	3	105	1	kompaktowy
Basic Q2M	1	102	1	kompaktowy
BRMOV5	14	113	6	kompaktowy
CX-5-F	7	107	7	profesjonalny
EKVp3	6	112	3	kompaktowy
EOEM33	10	101	1	profesjonalny
G3S	14	102	8	kompaktowy
GKR55	2	106	1	lustrzanka
Go Quad X	3	105	9	inny
K4K3K2	13	102	1	profesjonalny
MaxCap-...	7	103	8	profesjonalny
MG53-U	9	114	13	profesjonalny
STSEWTB	15	111	13	inny
UltraLens ...	11	110	6	lustrzanka
NULL	NULL	NULL	NULL	NULL

Próby naruszenia ograniczeń:

✖	16	20:13:24	INSERT INTO matryca (przekatna, rozdzielczosc, typ) VALUES (5.68, -5.0, 'test')	Error Code: 3819. Check constraint 'nieujemne' is violated.
✖	18	20:17:29	INSERT INTO obiektyw (model, minPrzeslona, maxPrzeslona) VALUES ('Nikkor AF-P', 3.5, 5.6)	Error Code: 3819. Check constraint 'maxwiekszeodmin' is violated.
✖	23	20:21:36	INSERT INTO aparat VALUES ('AXBPro', -6, 105, -1, 'kompaktowy')	Error Code: 3819. Check constraint 'dodatnie_id' is violated.
✖	30	20:28:38	INSERT INTO producent VALUES ('test', 'Aiigo', 'Chiny')	Error Code: 1366. Incorrect integer value: 'test' for column 'ID' at row 1

2.3.4. Opis rozwiązania

Zadanie jest raczej żmudne w wykonaniu, korzystam ze składni INSERT INTO Tabela VALUES i wypisuję wszystkie rekordy w nawiasach oddzielonych przecinkami. Jeśli klucz jest inkrementowany automatycznie, to nie wprowadzam go sam i zaznaczam to po nazwie tabeli wymienionymi w nawiasie kolumnami, które są uzupełniane.

Próba naruszenia jakiegokolwiek z ograniczeń CONSTRAINT kończy się błędem. Wprowadzenie danych niezgodnych z formatem również kończy się błędem.

2.4. Zadanie 4

2.4.1. Polecenie

Napisz procedurę, która na podstawie danych z tabel Matryca, Obiektów oraz Producent wygeneruje 100 nowych modeli aparatów (o losowych lub sekwencyjnych nazwach) i uzupełni nimi tabelę Aparat. Czy żaden z modeli się nie powtarza? Czy procedura może być utworzona i wykonana przez użytkownika z zadania 1? (2pkt)

2.4.2. Kwerendy niezbędne do realizacji

```
1  DELIMITER $$
2  ● CREATE PROCEDURE generatorAparatow()
3  BEGIN
4      DECLARE i INT DEFAULT 100;
5      DECLARE r INT DEFAULT 0;
6      DECLARE s VARCHAR(20);
7      WHILE i > 0 DO
8          SET r = FLOOR(RAND() * 4);
9          IF(r = 0)
10             THEN SET s = 'profesjonalny';
11          ELSEIF(r = 1)
12             THEN SET s = 'lustrzanka';
13          ELSEIF(r = 2)
14             THEN SET s = 'kompaktowy';
15          ELSE
16             SET s = 'inny';
17          END IF;
18          INSERT INTO Aparat VALUES (
19              UPPER(SUBSTRING(MD5(RAND()) FROM 1 FOR 8)),
20              FLOOR(RAND() * 15 + 1),
21              FLOOR((RAND() * 15) + 100),
22              FLOOR(RAND() * 15 + 1),
23              s);
24          SET i = i - 1;
25      END WHILE;
26  END$$
27  DELIMITER ;

CALL generatorAparatow();
```

2.4.3. Rezultat

Zamiast 15 aparatów w tabeli znajduje się ich 115:

✓ 187 21:48:56 SELECT * FROM Aparat

115 row(s) returned

	model	producent	matryca	obiektyw	typ
▶	013F9D86	7	112	12	kompaktowy
	0D01B2C4	8	110	3	kompaktowy
	0D90F815	3	103	10	lustrzanka
	0E6620AC	15	112	5	inny
	0EB3F27E	14	109	5	inny
	11856AF6	5	101	9	lustrzanka
	130D0DF0	12	103	12	profesjonalny
	13EEAB79	10	105	11	profesjonalny
	15F44FBA	3	112	10	inny
	17F806F8	7	103	12	lustrzanka
	195AF5D0	4	101	15	profesjonalny
	1AB38CD0	7	105	9	kompaktowy
	1CDA20FF	12	101	4	kompaktowy
	21733E51	11	114	11	inny
	21C64DC0	15	104	11	kompaktowy
	230F261A	1	106	15	profesjonalny
	2727E028	12	106	13	kompaktowy

Próba utworzenia procedury i jej wywołania przez użytkownika z zadania 1 kończy się niepowodzeniem:

```

34 21:51:47 CREATE PROCEDURE generatorAparatow() BEGIN DECLARE i INT DEFAULT 100;... Error Code: 1044. Access denied for user '261750'@'%' to database 'db-aparaty'

35 21:52:36 CALL generatorAparatow() Error Code: 1370. execute command denied to user '261750'@'%' for routine 'db-apar...
```

2.4.4. Opis rozwiązania

Na początek musimy zmienić separator po to, aby rozróżnić zakończenie procedury od zakończenia jednej komendy w procedurze (DELIMITER \$\$). Tworzę procedurę generatorAparatow poprzez CREATE PROCEDURE nazwa(). Wnętrze procedury zaczyna się od słowa kluczowego BEGIN, zaś kończy się na END\$\$ w tym przypadku.

Deklaruję trzy zmienne poprzez słowo kluczowe DECLARE:

- i – liczba iteracji czy też liczba dodanych rekordów (domyślnie 100),
- r – zmienna do przechowywania wartości pseudolosowej,
- s – jeden z dostępnych w typie wyliczeniowym tabeli Aparat ciąg znaków

W pętli WHILE, której warunek ($i > 0$) wraz z dekrementacją i zapewnia wykonanie wnętrza i razy, przypisuję pseudolosową liczbę całkowitą należącą do zbioru $\{0,1,2,3\}$. Wykonuję to przez pomnożenie wyniku funkcji RAND() (która zwraca liczbę zmiennoprzecinkową z zakresu $(0,1)$) przez 4 i wzięcie podłogi z rezultatu (FLOOR()). [Podobny zabieg stosuję dalej, robiąc analogiczne przekształcenia.] W zależności od tego jaką wartość przyjmie zmienna r, do zmiennej s przypisuję odpowiedni ciąg znaków z typu wyliczeniowego. Korzystam przy tym ze składni instrukcji warunkowych

IF(warunek) THEN komendy ELSEIF(warunek) THEN komendy ELSE komendy END IF;

Następnie tworzę nowy rekord w tabeli Aparaty, gdzie:

- Nazwa jest tworzona z wykorzystaniem funkcji hashującej MD5(), która może przyjąć jako argument RAND() i zwrócić wtedy hash będący pseudolosowym ciągiem małych liter i cyfr. Rezultat jest przycinany do długości 8 z wykorzystaniem funkcji SUBSTRING() oraz dla estetyki małe litery są zamieniane na wielkie z użyciem funkcji UPPER().
- ID producenta i obiektywu jest całkowitą liczbą pseudolosową z zakresu $\{1,...,15\}$.

- ID matrycy jest całkowitą liczbą pseudolosową z zakresu {100,...,114}.
- Typ jest pobierany jest ze zmiennej s.

Na koniec zamykam pętlę while (END WHILE;), zamykam procedurę, i przywracam domyślny separator.

Żaden z rekordów się nie powtarza, ze względu na zastosowaną technikę generowania nazw modeli, szansa na to jest bardzo niewielka.

2.5. Zadanie 5

2.5.1. Polecenie

Wypisz Napisz funkcję lub procedurę, która dla podanego ID producenta, zwraca wyprodukowany przez niego model aparatu z największą przekątną matrycy. (2pkt)

2.5.2. Kwerendy niezbędne do realizacji

```
1  DELIMITER $$
2  • CREATE FUNCTION najwiekszaMatryca (id INT)
3    RETURNS VARCHAR(30)
4    DETERMINISTIC
5  BEGIN
6    DECLARE s VARCHAR(30);
7    SET s = (SELECT aparat.model FROM aparat INNER JOIN producent ON aparat.producent=producent.ID
8            INNER JOIN matryca ON aparat.matryca = matryca.ID
9            WHERE aparat.producent = id AND matryca.przekatna =
10               (SELECT MAX(matryca.przekatna) FROM aparat INNER JOIN producent ON aparat.producent=producent.ID
11                INNER JOIN matryca ON aparat.matryca = matryca.ID WHERE aparat.producent = id) LIMIT 1);
12    RETURN s;
13  END$$
14
15  DELIMITER ;
16
17  • SELECT najwiekszaMatryca (1);
```

2.5.3. Rezultat

najwiekszaMatryca (1)
B50BC1F6

Przekątne matryc aparatów wyświetlane malejąco:

	model	producent	matryca	obiektyw	typ	ID	przekatna	rozdzielczosc	typ
▶	21733E51	11	114	11	inny	114	87.91	99.4	IMAX FF
	6E04FFCE	7	114	10	inny	114	87.91	99.4	IMAX FF
	A82CEB5B	3	114	7	profesjonalny	114	87.91	99.4	IMAX FF
	AF94D5C7	7	114	7	lustrzanka	114	87.91	99.4	IMAX FF
	C3731F0E	5	114	14	lustrzanka	114	87.91	99.4	IMAX FF
	C7099AB7	14	114	2	kompaktowy	114	87.91	99.4	IMAX FF
	CAB15D53	9	114	6	kompaktowy	114	87.91	99.4	IMAX FF
	EAB0B757	6	114	13	lustrzanka	114	87.91	99.4	IMAX FF
	MG53-U	9	114	13	profesjonalny	114	87.91	99.4	IMAX FF
	301E37A2	3	113	15	kompaktowy	113	67.40	70.9	Phase One
	6D138469	5	113	12	kompaktowy	113	67.40	70.9	Phase One
	8C09CA68	3	113	2	inny	113	67.40	70.9	Phase One
	BRMOV5	14	113	6	kompaktowy	113	67.40	70.9	Phase One
	013F9D86	7	112	12	kompaktowy	112	43.30	56.7	35mm FF+
	0E6620AC	15	112	5	inny	112	43.30	56.7	35mm FF+
	15F44FBA	3	112	10	inny	112	43.30	56.7	35mm FF+
	685AB418	9	112	6	lustrzanka	112	43.30	56.7	35mm FF+
	6E796609	3	112	12	lustrzanka	112	43.30	56.7	35mm FF+
	87A56664	6	112	1	kompaktowy	112	43.30	56.7	35mm FF+
	B50BC1F6	1	112	3	inny	112	43.30	56.7	35mm FF+

2.5.4. Opis rozwiązania

Zmieniam separator na \$\$ by odróżnić zakończenie funkcji od zakończenia komendy. Słowami kluczowymi CREATE FUNCTION deklaruje funkcję, nadaje jej nazwę największaMatryca i w nawiasach zapisuję parametr przyjmowany przez funkcję jako id producenta (typu int). Następnie stwierdzam słowem kluczowym RETURNS, jaki typ zmiennej będzie zwracany przez funkcję (VARCHAR, ponieważ chcemy zwrócić nazwę modelu). Ponieważ funkcja jest deterministyczna (przy niezmiennionej tablicy i niezmiennym parametrze zwróci za każdym razem zwróci to samo), zapisuję słowo kluczowe DETERMINISTIC. Rozpaczynam wewnątrz funkcji słowem BEGIN

Wewnątrz funkcji deklaruje zmienną s typu VARCHAR, będzie ona przechowywać nazwę modelu i będzie zwracana przez funkcję. Ustawiam s jako wynik zapytania pomocniczego. Zapytanie pomocnicze pozyskuje z tabeli Aparat nazwę modelu aparatu o największej przekątnej matrycy dla danego producenta, stąd wymagane powiązania INNER JOIN z tabelą Matryca. Wynik zapytania jest ograniczany do jednej pozycji, ponieważ może być wiele aparatów danego producenta z taką samą matrycą – w takim przypadku wybrany zostaje domyślnie leksykograficznie pierwszy model.

Funkcję kończę przez END\$\$, przywracam delimiter na domyślny i wywołuję funkcję dla przykładowego ID producenta, używając SELECT nazwafunkcji(parametr); .

2.6. Zadanie 6

2.6.1. Polecenie

Napisz trigger, który pozwoli wstawić do tabeli Aparat model wyprodukowany przez dowolnego producenta, a jeśli takiego nie ma jeszcze w bazie danych, dodaje go do odpowiedniej tabeli. (2pkt)

2.6.2. Kwerendy niezbędne do realizacji

```
1 DELIMITER $$
2 • CREATE TRIGGER nowyProducent BEFORE INSERT ON aparat
3 FOR EACH ROW
4 BEGIN
5     IF new.producent NOT IN (SELECT ID FROM producent)
6     THEN INSERT INTO producent VALUES
7         (new.producent, 'Nieznany Producent', 'Nieznany Kraj');
8     END IF;
9 END$$
10 DELIMITER ;
```

2.6.3. Rezultat

Test dodania aparatu o id producenta spoza bazy:

```
1 INSERT INTO aparat VALUES
2 ("Dziwny aparat", 16, 101, 5, 'lustrzanka');
```

5 13:06:33 INSERT INTO aparat VALUES ("Dziwny aparat", 16, 101, 5, 'lustrzanka') 1 row(s) affected

Bez triggera wyskoczyłby błąd, ponieważ id producenta jest kluczem obcym w tabeli Aparat, musi zatem istnieć w tabeli Producent. I rzeczywiście powstał nowy producent:

```
1 • SELECT * FROM `db-aparaty`.producent;
```

ID	nazwa	kraj
7	FusCam	Chiny
8	Kodak	Japonia
9	Lajka	Niemcy
10	Memento	Szwecja
11	PingZhang	Chiny
12	Państwowa Wytwór...	Polska
13	Sams	Korea Południowa
14	ToShibaInu	Japonia
15	Weiss	Austria
16	Nieznany Producent	Nieznany Kraj
NULL	NULL	NULL

2.6.4. Opis rozwiązania

Tak jak przy procedurach i funkcjach, najpierw zmieniam delimiter na \$\$. Słowami kluczowymi CREATE TRIGGER nazwa deklaruję utworzenie triggera. Określam, kiedy wyzwalacz ma być uruchamiany – przed wstawieniem rekordu do tabeli Aparat (BEFORE INSERT ON nazwa_tabeli). Następnie stwierdzam, że ciało wyzwalacza ma działać dla każdego rekordu.

Wewnątrz triggera korzystam z instrukcji warunkowej (IF ... THEN END IF;). Jeśli ID producenta nowego rekordu nie znajduje się w bazie Producent, dodaję go wraz z nazwą producenta

„Nieznany producent” i krajem siedziby „Nieznany kraj” (równie dobrze mógłbym wprowadzić NULL).

Kończę ciąg komend wyzwacza tak samo jak przy procedurach – END\$\$\$. Zmieniam delimiter na domyślne ‘;’.

2.7. Zadanie 7

2.7.1. Polecenie

Napisz funkcję, która dla podanego ID matrycy, zwraca liczbę modeli aparatów z daną matrycą. (2pkt)

2.7.2. Kwerendy niezbędne do realizacji

```
1 DELIMITER $$
2 • CREATE FUNCTION liczbaAparatowZMatryca (id INT)
3 RETURNS INT
4 DETERMINISTIC
5 BEGIN
6     DECLARE s INT DEFAULT 0;
7     SET s = IFNULL((SELECT COUNT(*) FROM aparat INNER JOIN matryca ON aparat.matryca = matryca.ID WHERE aparat.matryca = id), 0);
8     RETURN s;
9 END$$
10
11 DELIMITER ;
```

```
1 • SELECT liczbaAparatowZMatryca (105);
```

2.7.3. Rezultat

	liczbaAparatowZMatryca (105)
▶	11

Co zgadza się z:

```
✓ 227 00:19:24 SELECT * FROM aparat INNER JOIN matryca ON aparat.matryca = matryca.ID WHERE aparat.matryca = 105 11 row(s) returned
```

2.7.4. Opis rozwiązania

Składnia niezbędna przy tworzeniu i wywołaniu funkcji została opisana już przy zadaniu 5. Ta funkcja zwraca typ INT, ponieważ jest to liczba modeli aparatów z daną matrycą. Wewnątrz funkcji deklaruję zmienną `s` typu `int`, która zostanie zwrócona i do której przypisuję odpowiednio albo wartość zliczoną przez `COUNT()` dla danego typu matrycy, albo 0 jeśli zapytanie pomocnicze zwróci NULL (brak rekordów dla danej matrycy). Pomaga w tym funkcja `IFNULL()`, która zwraca pierwszą wartość, gdy nie jest ona równa null lub drugą, gdy ta pierwsza jest równa null.

2.8. Zadanie 8

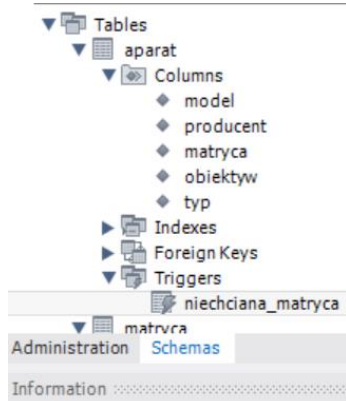
2.8.1. Polecenie

Napisz trigger usuwający z tabeli Matryca matrycę, dla której usunięty został ostatni aparat ją wykorzystujący. (2pkt)

2.8.2. Kwerendy niezbędne do realizacji

```
1 DELIMITER $$
2 • CREATE TRIGGER niehciana_matryca AFTER DELETE ON aparat
3 FOR EACH ROW
4 BEGIN
5     DELETE FROM matryca WHERE matryca.ID NOT IN
6     (SELECT DISTINCT A.ID FROM
7      (SELECT matryca.ID FROM matryca INNER JOIN aparat ON matryca.ID = aparat.matryca) AS A) AND matryca.ID > 0;
8 END$$
9 DELIMITER ;
```

2.8.3. Rezultat



Trigger: niehciana_matryca

Definition:

Event	DELETE
Timing	AFTER

2.8.4. Opis rozwiązania

Składnia potrzebna do stworzenia wyzwalacza (triggera) została opisana w zadaniu 6. Tutaj różnica polega na użyciu słów kluczowych AFTER DELETE, aby zaznaczyć, że trigger ma być uruchamiany po usunięciu rekordów z tabeli Aparat.

Wewnątrz triggera znajduje się jedno polecenie, zgodne z celem zdania. Usuwać z tabeli matryca te rekordy, które nie znajdują się w wyniku zapytania pomocniczego. Zapytanie pomocnicze zbiera unikalne wystąpienia ID matryc w tabeli Aparat, czyli po prostu pokazuje, które matryce są wykorzystywane przez jakikolwiek model aparatu.

2.9. Zadanie 9

2.9.1. Polecenie

Stwórz widok zawierający model aparatu, nazwę jego producenta, przekątną i rozdzielczość jego matrycy, minimalną oraz maksymalną wartość przesłony dla wszystkich aparatów typu *lustrzanka* wyprodukowanych przez producentów mających siedzibę poza Chinami. Czy zadanie może wykonać użytkownik z zadania 1? (2pkt)

2.9.2. Kwerendy niezbędne do realizacji

```
1 • CREATE VIEW LustrzankiNiechinskie AS
2 SELECT
3     aparat.model, producent.nazwa,
4     matryca.przekatna, matryca.rozdzielczosc,
5     obiektyw.minPrzeslona, obiektyw.maxPrzeslona
6 FROM aparat INNER JOIN matryca ON aparat.matryca = matryca.ID
7 INNER JOIN obiektyw ON aparat.obiektyw = obiektyw.ID
8 INNER JOIN producent ON aparat.producent = producent.ID
9 WHERE aparat.typ = 'lustrzanka' AND producent.kraj <> 'Chiny';
```

```
1 • SELECT * FROM `db-aparaty`.lustrzankiniechinskie;
```

2.9.3. Rezultat

Próba utworzenia widoku przez użytkownika z zadania 1. kończy się niepowodzeniem (nie ma on uprawnień do CREATE):

36 00:30:39 CREATE VIEW LustrzankiNiechinskie AS SELECT aparat.model, producent.nazwa, ... Error Code: 1142. CREATE VIEW command denied to user '261750'@'localhost' for t...

Tak wygląda fragment widoku utworzonego przez roota:

	model	nazwa	przekatna	rozdzielczosc	minPrzeslona	maxPrzeslona
	B22187F4	Agpha	21.60	23.0	3.4	6.5
	GKR55	Agpha	7.66	15.0	3.5	5.6
▶	506412BC	Cannon	5.68	5.0	2.8	5
	11856AF6	Casijo	5.68	10.0	3.5	5.8
	6D19FAB4	Casijo	6.00	8.5	4	4.5
	C3731F0E	Casijo	87.91	99.4	3.5	5.6
	CAED66AC	Casijo	43.30	56.7	2.85	6.3
	64896604	Kokodak	7.18	14.3	3.7	8
	685AB418	Lejka	43.30	56.7	4	5.6
	CC567BAB	Lejka	7.66	15.0	2.8	5
	2ED75FA6	Memento	7.18	10.0	3	6.5
	7CB3C926	Memento	6.00	8.5	3.5	5.6
	E7F222D9	Państwow...	43.20	45.0	2.8	5
	64356985	ToShibaInu	21.60	23.0	3.4	7.9

2.9.4. Opis rozwiązania

Do stworzenia widoku potrzebne jest sformułowanie CREATE VIEW nazwa AS. Po nim tworzy się normalne zapytanie SELECT z tym, że zostanie ono właśnie utrwalone w widoku. W zapytaniu łączę ze sobą wszystkie tabele po właściwych kluczach i wprowadzam warunek, że typ aparatu to lustrzanka i kraj różny od Chin. Zadania nie może wykonać użytkownik z zadania 1.

2.10. Zadanie 10

2.10.1. Polecenie

Stwórz widok zawierający nazwę i kraj producenta oraz model dla wszystkich aparatów. Następnie usuń z tabeli Aparat wszystkie modele producentów z Chin. Czy coś zmieniło się w widoku? (2pkt)

2.10.2. Kwerendy niezbędne do realizacji

```
1 • CREATE VIEW widok_zadanie10 AS
2   SELECT
3     producent.nazwa,
4     producent.kraj,
5     aparat.model
6   FROM aparat
7   INNER JOIN producent ON aparat.producent = producent.ID;

1 • SELECT * FROM widok_zadanie10;
```

Usuwanie z tabeli Aparat wszystkich modeli producentów z Chin:

```
1 • DELETE FROM aparat WHERE producent IN (SELECT ID FROM producent WHERE kraj = 'Chiny');
```

2.10.3. Rezultat

✓ 8 13:34:39 SELECT * FROM 'db-aparaty'.widok_zadanie10 116 row(s) returned

	nazwa	kraj	model
▶	Aiigo	Chiny	230F261A
	Aiigo	Chiny	73EA5AD7
	Aiigo	Chiny	976CEBA8
	Aiigo	Chiny	ACE07DE3
	Aiigo	Chiny	B50BC1F6
	Aiigo	Chiny	Basic Q2M
	Aiigo	Chiny	E502747C
	Agpha	Niemcy	3BDEBE4B
	Agpha	Niemcy	557E4958
	Agpha	Niemcy	5E69639C
	Agpha	Niemcy	69D27BC4
	Agpha	Niemcy	A0A5AEAD
	Agpha	Niemcy	ABB65310
	Agpha	Niemcy	B22187F4
	Agpha	Niemcy	B66C7303
	Agpha	Niemcy	C076CDA5
	Agpha	Niemcy	D673128F
	Agpha	Niemcy	GKR55

k_zadanie10 1 x

Usuwać chińskie aparaty:

✓ 5 13:48:05 DELETE FROM aparat WHERE producent IN (SELECT ID FROM producent WH... 42 row(s) affected

Po usunięciu chińskich aparatów z tabeli Aparat, znikają one również z widoku (choć nic nie było na samym widoku robione!):

✓ 6 13:49:37 SELECT * FROM 'db-aparaty'.widok_zadanie10 74 row(s) returned

	nazwa	kraj	model
►	Agpha	Niemcy	3BDEBE4B
	Agpha	Niemcy	557E4958
	Agpha	Niemcy	5E69639C
	Agpha	Niemcy	69D27BC4
	Agpha	Niemcy	A0A5AEAD
	Agpha	Niemcy	ABB65310
	Agpha	Niemcy	B22187F4
	Agpha	Niemcy	B66C7303
	Agpha	Niemcy	C076CDA5
	Agpha	Niemcy	D673128F
	Agpha	Niemcy	GKR55
	Cannon	Japonia	195AF5D0
	Cannon	Japonia	506412BC
	Cannon	Japonia	7D91B8C9
	Cannon	Japonia	86243893
	Cannon	Japonia	B8DB6EA6
	Cannon	Japonia	F6C38E14

2.10.4. Opis rozwiązania

Mając doświadczenie z zadania 9., utworzenie tego widoku jest raczej proste. Zawartość widoku można wyświetlić poprzez SELECT tak samo jak normalne tabele, wprowadzając nazwę widoku po słowie FROM. Następnie usuwam rekordy z tabeli Aparat, których producent dosłownie znajduje się w zbiorze producentów z Chin. Upewniam się, że polecenie zostało wykonane poprawnie i zauważam, że pomimo braku edycji widoku, jego zawartość została zaktualizowana i brakuje w nim chińskich aparatów.

2.11. Zadanie 11

2.11.1. Polecenie

Do tabeli **Producent** dodaj kolumnę *liczbaModeli* i uzupełnij ją na podstawie danych z tabeli **Aparat** (zadbaj, aby w kolumnie tej nie było wartości *null*). Napisz nowe lub uzupełnij stare triggera, tak by przy każdej zmianie (dodanie nowego modelu, usunięcie modelu, zmiana producenta) korygować odpowiednią wartość w nowej kolumnie. Czy taki trigger może zostać utworzony przez użytkownika z zadania 1? Czy będzie działał podczas operacji na tabelach wykonywanych przez tego użytkownika, jeśli trigger został stworzony przez roota? (2pkt)

2.11.2. Kwerendy niezbędne do realizacji

```
1 • ALTER TABLE producent ADD COLUMN liczbaModeli INT NOT NULL DEFAULT 0;

1 • UPDATE producent
2   SET liczbaModeli = IFNULL((SELECT COUNT(*) FROM aparat WHERE aparat.producent = producent.ID), 0);

1 DELIMITER $$
2 • CREATE TRIGGER liczbaModeli_delete AFTER DELETE ON aparat
3   FOR EACH ROW
4   BEGIN
5     UPDATE producent
6     SET liczbaModeli = IFNULL((SELECT COUNT(*) FROM aparat WHERE aparat.producent = producent.ID), 0);
7   END$$
8 DELIMITER ;

1 DELIMITER $$
2 • CREATE TRIGGER liczbaModeli_update AFTER UPDATE ON aparat
3   FOR EACH ROW
4   BEGIN
5     UPDATE producent
6     SET liczbaModeli = IFNULL((SELECT COUNT(*) FROM aparat WHERE aparat.producent = producent.ID), 0);
7   END$$
8 DELIMITER ;

1 DELIMITER $$
2 • CREATE TRIGGER liczbaModeli_insert AFTER INSERT ON aparat
3   FOR EACH ROW
4   BEGIN
5     UPDATE producent
6     SET liczbaModeli = IFNULL((SELECT COUNT(*) FROM aparat WHERE aparat.producent = producent.ID), 0);
7   END$$
8 DELIMITER ;
```

Sprawdzenie jednego z triggerów:

```
1 • SET SQL_SAFE_UPDATES = 0;
2 • INSERT INTO aparat VALUES
3   ('Aparat Testowy', 12, 100, 10, 'kompaktowy');
```

Jeśli chcemy zachować trigger z zadania 6. to trzeba go lekko zmodyfikować:

```
1 • DROP TRIGGER IF EXISTS nowyProducent;
```

```

1 DELIMITER $$
2 CREATE TRIGGER nowyProducent BEFORE INSERT ON aparat
3 FOR EACH ROW
4 BEGIN
5     IF new.producent NOT IN (SELECT ID FROM producent)
6     THEN INSERT INTO producent VALUES
7         (new.producent, 'Nieznany Producent', 'Nieznany Kraj', 0);
8     END IF;
9 END$$
10 DELIMITER ;

```

Sprawdzenie zachowania triggera z zadania 6. i triggera z tego zadania:

```

1 SET SQL_SAFE_UPDATES = 0;
2 INSERT INTO aparat VALUES
3 ('Nowy Aparat', 20, 100, 10, 'kompaktowy');

```

2.11.3. Rezultat

Tabela Producent po dodaniu kolumny:

	ID	nazwa	kraj	liczbaModeli
▶	1	Aiigo	Chiny	0
	2	Agpha	Niemcy	11
	3	BingWang	Chiny	0
	4	Cannon	Japonia	6
	5	Casijo	Japonia	8
	6	DJJ	Chiny	0
	7	FusCam	Chiny	0
	8	Kodak	Japonia	3
	9	Lejka	Niemcy	7
	10	Memento	Szwecja	8
	11	PingZhang	Chiny	0
	12	Państwowa Wytwórnia Optyki	Polska	10
	13	Sams	Korea ...	5
	14	ToShibaInu	Japonia	10
	15	Weiss	Austria	5
	16	Nieznany Producent	Niezna...	1
•	NULL	NULL	NULL	NULL

Aparatów chińskich nie ma ze względu na zadanie 10.

Próba utworzenia triggera przez użytkownika z zadania 1.:

```

❌ 1 14:26:15 CREATE TRIGGER liczbaModeli_delete AFTER DELETE ON aparat FOR EACH ... Error Code: 1419. You do not have the SUPER privilege and bir

```

Gdy tworzy root:

```

✅ 16 14:27:49 CREATE TRIGGER liczbaModeli_delete AFTER DELETE ON aparat FOR EA... 0 row(s) affected
✅ 17 14:28:41 CREATE TRIGGER liczbaModeli_update AFTER UPDATE ON aparat FOR EA... 0 row(s) affected
✅ 18 14:31:30 CREATE TRIGGER liczbaModeli_insert AFTER INSERT ON aparat FOR EAC... 0 row(s) affected

```

Sprawdzam czy triggerzy działają; ponieważ są praktycznie identyczne, zrobię to dodając nowy Aparat z pomocą użytkownika z zadania 1. Rezultat:

	ID	nazwa	kraj	liczbaModeli
▶	1	Aliigo	Chiny	0
	2	Agpha	Niemcy	11
	3	BingWang	Chiny	0
	4	Cannon	Japonia	6
	5	Casijo	Japonia	8
	6	DJJ	Chiny	0
	7	FusCam	Chiny	0
	8	Kodak	Japonia	3
	9	Lejka	Niemcy	7
	10	Memento	Szwecja	8
	11	PingZhang	Chiny	0
	12	Państwowa Wytwórnia Optyki	Polska	11
	13	Sams	Korea ...	5
	14	ToShibaInu	Japonia	10
	15	Weiss	Austria	5
	16	Nieznany Producent	Niezna...	1
•	NULL	NULL	NULL	NULL

Trigger działa, kolumna liczbaModeli została poprawnie zaktualizowana.

Co więcej, gdy dodamy aparat z nowym ID producenta, dzięki triggerowi z zadania 6. to ID zostanie dodane (Nieznany Producent) i następnie zostanie przypisana liczbaModeli równa 1:

2.11.4. Opis rozwiązania

Wykorzystana w tym zadaniu składnia była już omawiana poprzednio. Tworzę kolumnę liczbaModeli poprzez ALTER TABLE ADD COLUMN. Jako typ ustawiam INT oraz zaznaczam, żeby nie przyjmowała ona wartości null (NOT NULL). Domyślnie przyjmuje ona wartość 0.

Modyfikuję zawartość kolumny (UPDATE... SET...), wstawiając wartości zwrócone przez zapytanie pomocnicze, jeśli nie są one równe NULL (gdy są, z pomocą IFNULL() wstawiam 0). W zapytaniu pomocniczym zliczam rekordy z id producenta równym id danego wiersza w tabeli.

Następnie powstają trzy nowe triggerzy (z identycznym wnętrzem), wyzwalane: po usunięciu rekordów (AFTER DELETE), po dodaniu rekordów (AFTER INSERT) i po modyfikacji rekordów (AFTER UPDATE). Wewnątrz tych triggerów znajduje się polecenie zaktualizowania kolumny liczbaModeli w sposób opisany akapit wyżej. Triggerów nie może utworzyć użytkownik z zadania 1.

Sprawdzam działanie triggerów przy operacjach wykonywanych przez użytkownika z zadania pierwszego, aktualizują poprawnie kolumnę. Dodatkowo (nie było to wymagane w zadaniu) aktualizuję trigger z zadania 6. Uwzględniłam w nim nowo powstałą kolumnę.