



**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**ФАКУЛТЕТ КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ**

**Катедра „Компютърни системи“**

## **КУРСОВ ПРОЕКТ**

### **ПО БАЗИ ОТ ДАННИ**

**Студент : Стефан Стоилков**

**ФАК. № : 123221012**

**Група : 43**

<b>Тема № 25</b>
Напишете база данни за сайт за продажба на песни. Да се пази основна информация за композитора, изпълнителя и песента – жанр, стил, аранжиратор, продължителност. Една песен може да бъде написана от повече от един композитор и изпълнявана от повече от един изпълнител. Допълнете таблиците с необходима информация по ваш избор.

1. Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки за средата MySQL.
2. Напишете заявка, в която демонстрирате SELECT с ограничаващо условие по избор.
3. Напишете заявка, в която използвате агрегатна функция и GROUP BY по ваш избор.
4. Напишете заявка, в която демонстрирате INNER JOIN по ваш избор.
5. Напишете заявка, в която демонстрирате OUTER JOIN по ваш избор.
6. Напишете заявка, в която демонстрирате вложен SELECT по ваш избор.
7. Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функция.
8. Създайте тригер по ваш избор.
9. Създайте процедура, в която демонстрирате използване на курсор.

## **1. Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки за средата MySQL.**

Според условието на задачата, необходимо е да бъдат съхранени данни за основните обекти - композитори, изпълнители, песни, като за всяка песен трябва да знаем за нейния жанр, стил, аранжиратор и нейната продължителност.

Допълнително ще създадем още няколко таблици.

Започваме с таблицата 'userRole', в която ще пазим информация за ролите на потребителите в сайта за продажба на песни. Ролите ще са две (администратор и обикновен потребител).

За всеки потребител ще пазим информация в таблицата 'user', която ще включва колони за ID, име, фамилия, ЕГН, адрес, телефон и външен ключ към таблицата 'userRole', който ще определя дали потребителят е администратор или обикновен потребител.

Следваща таблица, която ще създадем е таблица свързана с поръчки – 'orders'.

В нея ще съхраняваме информацията за датата на поръчката, общата цена на поръчката, статуса на плащането и статуса на доставката. Ще имаме и едно поле, което е външен ключ към таблицата 'user', за да знаем коя поръчка на кой потребител принадлежи.

Потребителите могат да си създават плейлисти. В таблицата 'playlists' ще съхраняваме името на плейлистата и броя на песните в нея.

Тъй като една песен може да включва повече жанра, ще създадем отделна таблица 'genre'.

Следваща таблица необходима за реализирането на задачата ще бъде – 'albums'.

В нея ще пазим информация за името на албума, броя на песните в него, дължината на всички песни в минути и секунди, дата на издаване, музикален издател, цената на албума и един външен ключ сочещ към таблицата изпълнители – 'performer'.

Ще имаме и таблица за рецензии от страна на потребителите – 'reviews'. Полетата в нея ще ни дават информация за : рейтинг на албума между 0 и 10, добавен коментар към рецензията, дата на добавяне на рецензията, и два външни ключа към таблиците 'user' и 'albums'.

Създаваме таблица продажби - 'sales', която ще съдържа данните за продажбите на албумите на сайта. Ще пазим дата и цена на продажбата, и едно поле, което ще ни е външен ключ към таблицата поръчки – 'orders'.

Останалите таблици, които са ни нужни, са реализирани по между си чрез връзка 'много към много' (many-to-many relationship), и те са : 'performer\_song', 'performer\_genre', 'song\_genre', 'composer\_song', 'album\_genre', 'playlist\_album', 'playlist\_user', 'user\_album' и 'sale\_album'.

За да проектираме базата ще използваме модела ER-диаграма (Entity Relationship Diagram).



```

CREATE TABLE userRole (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL
);

CREATE TABLE user (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
username VARCHAR(255) NOT NULL,
password VARCHAR(255) NOT NULL,
egn CHAR(10) NOT NULL UNIQUE CONSTRAINT EGNuser
CHECK(CHAR_LENGTH(egn)=10),
address VARCHAR(255) NOT NULL,
phone VARCHAR(30) DEFAULT NULL,
userRole_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (userRole_id) REFERENCES userRole(id)
);

CREATE TABLE orders (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
order_date DATE NOT NULL,
price FLOAT NOT NULL,
payment_status VARCHAR(100) NOT NULL,
delivery_status VARCHAR(100) NOT NULL,
user_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (user_id) REFERENCES user(id)
);

CREATE TABLE playlists (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
songCount INT UNSIGNED NOT NULL
);

CREATE TABLE genre (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL
);

```

```

CREATE TABLE performer (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
egn CHAR(10) NOT NULL UNIQUE CONSTRAINT EGNperformer
CHECK(CHAR_LENGTH(egn)=10),
address VARCHAR(255) NOT NULL,
phone VARCHAR(30) DEFAULT NULL,
dateOfBirth DATE NOT NULL
);

CREATE TABLE albums (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
title VARCHAR(255) NOT NULL,
numberOfSongs INT NOT NULL,
length DECIMAL(6,2),
release_date DATE NOT NULL,
record_label VARCHAR(255) NOT NULL,
price FLOAT NOT NULL,
performer_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (performer_id) REFERENCES performer(id)
);

CREATE TABLE song (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
link VARCHAR(255) NOT NULL UNIQUE,
style VARCHAR(100) NOT NULL,
arrangement VARCHAR(100) NOT NULL,
duration INT UNSIGNED NOT NULL,
numberOfViews BIGINT UNSIGNED NOT NULL,
album_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (album_id) REFERENCES albums(id)
);

CREATE TABLE reviews (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
rating DECIMAL(10,2) NOT NULL,
comment TEXT NOT NULL,
review_date DATE NOT NULL,
user_id INT UNSIGNED NOT NULL,
album_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (user_id) REFERENCES user(id),
CONSTRAINT FOREIGN KEY (album_id) REFERENCES albums(id)
);

CREATE TABLE sales (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
sale_date DATE NOT NULL,
sale_price FLOAT NOT NULL,
orders_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (orders_id) REFERENCES orders(id)
);

```

```
CREATE TABLE performer_song (  
performer_id INT UNSIGNED NOT NULL,  
song_id INT UNSIGNED NOT NULL,  
CONSTRAINT FOREIGN KEY (performer_id) REFERENCES performer(id),  
CONSTRAINT FOREIGN KEY (song_id) REFERENCES song(id)  
);
```

```
CREATE TABLE performer_genre (  
performer_id INT UNSIGNED NOT NULL,  
genre_id INT UNSIGNED NOT NULL,  
CONSTRAINT FOREIGN KEY (performer_id) REFERENCES performer(id),  
CONSTRAINT FOREIGN KEY (genre_id) REFERENCES genre(id)  
);
```

```
CREATE TABLE song_genre (  
song_id INT UNSIGNED NOT NULL,  
genre_id INT UNSIGNED NOT NULL,  
CONSTRAINT FOREIGN KEY (song_id) REFERENCES song(id),  
CONSTRAINT FOREIGN KEY (genre_id) REFERENCES genre(id)  
);
```

```
CREATE TABLE composer_song (  
composer_id INT UNSIGNED NOT NULL,  
song_id INT UNSIGNED NOT NULL,  
CONSTRAINT FOREIGN KEY (composer_id) REFERENCES composer(id),  
CONSTRAINT FOREIGN KEY (song_id) REFERENCES song(id),  
);
```

```
CREATE TABLE album_genre (  
album_id INT UNSIGNED NOT NULL,  
genre_id INT UNSIGNED NOT NULL,  
CONSTRAINT FOREIGN KEY (album_id) REFERENCES albums(id),  
CONSTRAINT FOREIGN KEY (genre_id) REFERENCES genre(id)  
);
```

```
CREATE TABLE playlist_album (  
playlist_id INT UNSIGNED NOT NULL,  
album_id INT UNSIGNED NOT NULL,  
CONSTRAINT FOREIGN KEY (playlist_id) REFERENCES playlists(id),  
CONSTRAINT FOREIGN KEY (album_id) REFERENCES albums(id)  
);
```

```
CREATE TABLE playlist_user (
playlist_id INT UNSIGNED NOT NULL,
user_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (playlist_id) REFERENCES playlists(id),
CONSTRAINT FOREIGN KEY (user_id) REFERENCES user(id)
);
```

```
CREATE TABLE user_album (
user_id INT UNSIGNED NOT NULL,
album_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (user_id) REFERENCES user(id),
CONSTRAINT FOREIGN KEY (album_id) REFERENCES albums(id)
);
```

```
CREATE TABLE sale_album (
sale_id INT UNSIGNED NOT NULL,
album_id INT UNSIGNED NOT NULL,
CONSTRAINT FOREIGN KEY (sale_id) REFERENCES sales(id),
CONSTRAINT FOREIGN KEY (album_id) REFERENCES albums(id)
);
```

Добавяме и тестови данни в таблиците :

```
INSERT INTO composer (name, egn, nationality, phone, dateOfBirth)
VALUES ('Hanz Zimmer', '0912934563', 'German', '455-9009', '1957-09-12'),
('Ed Sheeran', '0342673901', 'British', '550-0673', '1991-02-17'),
('Toots Thielemans', '0067890123', 'German', NULL, '1922-04-29'),
('Manu Chao', '3450729113', 'French', '645-2480', '1961-06-21'),
('Freddie Mercury', '0608901230', 'British', NULL, '1946-09-05'),
('Alexandre Desplat', '6789012345', 'French', NULL, '1961-08-23'),
('Drake', '7890123050', 'American', '535-8903', '1986-10-24'),
('Joe Dassin', '8901934560', 'French', NULL, '1938-11-05'),
('Kanye West', '9012341638', 'American', '989-0909', '1977-06-08'),
('Charles Aznavour', '9897866540', 'French', NULL, '1924-05-22'),
('Philip Glass', '2146570901', 'American', NULL, '1937-01-31'),
('Elvis Presley', '0459789032', 'American', NULL, '1935-01-08');
```

```
INSERT INTO userRole
VALUES (NULL, 'Admin'),
       (NULL, 'Customer');
```

```
INSERT INTO user (username, password, egn, address, phone, userRole_id)
VALUES ('John Brown', 'H@rdT0Gu3ss!', '0123426739', '123 Main St, Anytown', '353-1333', 2),
       ('Jane Lee', '5tr0ngP@55w0rd', '2305673902', '21 Oak Ave, Otherville', NULL, 2),
       ('Oleg Smirnov', 'cat1@3!?5t', '3251789913', '34 Elm St, Somewhere', '757-1629', 2),
       ('Sara Ali', 'tY5nCj!1', '1567390224', '78 Pine Rd, Anytown', NULL, 2),
       ('Mike Durant', 'P@8rKm#7', '5378101237', '21 Maple Dr, Otherville', '252-4424', 2),
       ('Bob Wilson', 'Q8zHfL!7a', '9782032300', '21A Cedar Ave, Somewhere', NULL, 2),
       ('Ana Petkova', '8#vQfLj!', '1890223959', '32B Oak St, Anytown', '607-8780', 2),
       ('Robert James', 'M9pBcT!z', '8901257547', '87A Birch Ln, Otherville', NULL, 2),
       ('Ekaterina Valova', '1#jGHTk$', '9012245778', '133 Elm St, Somewhere', '989-3460', 2),
       ('Beatris Grivardis', 'F7nGkP!z', '0103459701', '454 Maple Ave, Anytown', '088-5035', 1);
```

```
INSERT INTO orders (order_date, price, payment_status, delivery_status, user_id)
VALUES ('2023-03-25', 50.00, 'paid', 'delivered', 1),
       ('2023-03-26', 75.00, 'paid', 'delivered', 2),
       ('2023-03-26', 120.00, 'paid', 'delivered', 2),
       ('2023-03-27', 200.00, 'paid', 'delivered', 3),
       ('2023-04-03', 150.00, 'paid', 'in transit', 4),
       ('2023-04-04', 300.00, 'paid', 'in transit', 5),
       ('2023-04-07', 400.00, 'pending', 'pending', 5),
       ('2023-04-07', 250.00, 'paid', 'in transit', 5),
       ('2023-04-07', 100.00, 'paid', 'in transit', 6),
       ('2023-04-10', 32.00, 'paid', 'delivered', 7),
       ('2023-04-10', 65.00, 'paid', 'delivered', 8),
       ('2023-04-12', 70.00, 'pending', 'not shipped', 9),
       ('2023-04-15', 500.00, 'pending', 'not shipped', 10);
```

```
INSERT INTO playlists (name, songCount)
VALUES ('Rock Anthems', 15),
       ('Hip Hop Bangers', 12),
       ('Jazz Classics', 10),
       ('Chill Vibes', 8),
       ('Party Mix', 20),
       ('Road Trip Tunes', 18),
       ('Pop Hits', 14),
       ('Workout playlist', 16),
       ('Oldies But Goodies', 10),
       ('Country Favorites', 12);
```



```
INSERT INTO genre
VALUES (NULL, 'Pop'),
      (NULL, 'Pop-punk'),
      (NULL, 'Hip-hop'),
      (NULL, 'Trip-hop'),
      (NULL, 'Jazz'),
      (NULL, 'Jazz fusion'),
      (NULL, 'Rock'),
      (NULL, 'Funk-rock');
```

```
INSERT INTO performer
VALUES (NULL, 'Drake', '9124656742', '111 Pine St San Francisco, CA 94111', '098-4265', '1986-10-24'),
      (NULL, 'Future', '9219309213', '456 Oak Ave Springfield, USA', '099-4372', '1990-02-16'),
      (NULL, 'Jimi Hendrix', '9029831092', '1111 Maple Ln Anytown, USA', '218-3890', '1942-11-27'),
      (NULL, 'Ed Sheeran', '2909719423', '9999 Cedar Dr Anytown, USA', '012-4567', '1991-02-17'),
      (NULL, 'Eric Clapton', '3053210991', '7879 Oak Ln Springfield, USA', NULL, '1945-03-30'),
      (NULL, 'Louis Armstrong', '2907980123', '2525 Pine Ln Springfield, USA', '128-4355', '1901-08-04'),
      (NULL, 'Lil Nas X', '9908978213', '2020 Pine Dr Anytown, USA', NULL, '1999-04-09'),
      (NULL, 'Michael Jackson', '8909234560', '2222 Oak Ave Anytown, USA', '099-4369', '1958-08-29'),
      (NULL, 'Miles Davis', '3202428290', '5555 Pine Ln Anytown, USA', '198-5141', '1926-05-26'),
      (NULL, 'Macklemore', '4545009877', '6767 Cedar Rd Anytown, USA', NULL, '1983-06-19'),
      (NULL, 'Kanye West', '9012341638', '2727 Oak St Springfield, USA', '108-4362', '1977-06-08'),
      (NULL, 'John Legend', '3402921045', '2424 Birch Ct Anytown, USA', '028-1285', '1978-12-28'),
      (NULL, 'Billie Holiday', '3003210981', '4433 Cedar St Miami, USA', NULL, '1915-07-17'),
      (NULL, 'Joseph Satriani', '3059210292', '853 Cedar Ave Springfield, USA', '201-4369', '1956-07-15'),
      (NULL, '21 Savage', '9359410219', '0929 Elm St Anytown, USA', '101-5468', '1992-10-22'),
      (NULL, 'Aggro Santoss', '0395210283', '8080 Maple Ln Anytown, USA', '109-2309', '1989-10-12'),
      (NULL, 'Common', '5059220290', '0880 Oak Ln Springfield, USA', '252-4359', '1972-03-13'),
      (NULL, 'Cardi B', '4059320291', '5608 Maple Blvd Los Angeles, USA', '522-3790', '1992-10-11'),
      (NULL, 'Ceza', '3038220281', '4884 Oak Ln Springfield, USA', '530-3438', '1976-12-31'),
      (NULL, 'Tim Dog', '4139221298', '729 Poplar Ave Anytown, USA', '601-4392', '1976-01-03');
```

```
INSERT INTO albums
VALUES (NULL, 'High Off Life', 21, 80.20, '2020-05-15', 'Republic Records', 250.00, 1),
      (NULL, 'Are You Experienced', 8, 42.39, '1969-10-22', 'Atlantic Records', 50.00, 3),
      (NULL, 'Divide', 11, 50.03, '2017-03-03', 'Asylum', 150.00, 4),
      (NULL, 'GEMINI', 16, 60.13, '2017-09-22', 'Bendo LLC', 600.00, 10),
      (NULL, 'The Orange Room', 15, 55.29, '2005-03-01', 'Sheeran Lock', 110.00, 4),
      (NULL, '7', 15, 43.37, '2021-09-17', 'Columbia Records', 120.00, 7),
      (NULL, 'Thriller', 9, 42.19, '1982-11-30', 'Epic Records', 100.00, 8),
      (NULL, 'Kind of Blue', 5, 45.52, '1959-08-17', 'Columbia Records', 400.00, 9),
      (NULL, 'Scorpion', 25, 89.04, '2018-06-29', 'Cash Money Records', 450.00, 1),
      (NULL, 'Future', 19, 66.14, '2017-02-17', 'Interscope Records', 300.00, 2),
      (NULL, 'Graduation', 13, 51.23, '2008-08-19', 'KonLive', 250.00, 11),
      (NULL, 'Wake Up', 13, 51.13, '2008-08-19', 'Domino Recording Company', 120.00, 12),
      (NULL, 'All or Nothing at All', 15, 32.48, '1938-08-19', 'Sony Music', 30.00, 13),
      (NULL, 'The Elephants of Mars', 10, 59.04, '1938-08-19', 'Universal Music Group', 150.00, 14),
      (NULL, 'Hello Dolly', 11, 40.24, '1938-08-19', 'Warner Music', 150.00, 6),
      (NULL, 'Journeyman', 12, 42.32, '1938-08-19', 'XL Recordings', 100.00, 5);
```

INSERT INTO song

VALUES (NULL, 'Shape Of You', 'https://www.youtube.com/watch?v=JGwWNGJdvx8', 'Pop', 'R&B', 263, 5900000000, 3),  
(NULL, 'Life Is Good', 'https://www.youtube.com/watch?v=l0U7SxXHkPY', 'Hip-hop', 'Trap', 237, 2100050978, 1),  
(NULL, 'Hard Times',  
'https://www.youtube.com/watch?v=uuB1d4Jn5H8&list=PLbGuJX1GQkodSan5SUi0k6i89hGTz9fjL', 'Rock', 'Blues Rock', 157, 345600000, 16),  
(NULL, 'Stronger', 'https://www.youtube.com/watch?v=PsO6ZnUZI0g', 'Hip-hop', 'Electronic', 246, 450908795, 11),  
(NULL, 'Thrift Shop', 'https://www.youtube.com/watch?v=QK8mJJJvaes', 'Hip-hop', 'Comedy Rap', 432, 345698098, 4),  
(NULL, 'Fire',  
'https://www.youtube.com/watch?v=cbG7HEEPE1o&list=PL1RRNXldMeRFm7CqtN44oav-TgaWT3yt\_', 'Hard rock', 'Electronic', 278, 4500989, 2),  
(NULL, 'Billie Jean', 'https://www.youtube.com/watch?v=Zi\_XLOBDo\_Y', 'Pop', 'Drum machines', 321, 123456432, 7),  
(NULL, 'Old Town Road', 'https://www.youtube.com/watch?v=w2Ov5jzm3j8', 'Pop', 'Electronic', 157, 345600000, 6),  
(NULL, 'Addicted',  
'https://www.youtube.com/watch?v=6YS03WG2t1Q&list=PLT3w2jU3eU6nF1HzjTtKYzvMhJIvfQ8me', 'Latin Pop', 'Piano', 320, 10500321, 5),  
(NULL, 'Sweet Child O Mine', 'https://www.youtube.com/watch?v=1w7OgIMMRc4', 'Hard rock', 'Electronic', 454, 1234000989, 16),  
(NULL, 'So What',  
'https://www.youtube.com/watch?v=ylXk1LBvIqU&list=PLrhkpF1bKMG\_D2sxTpxG63WGmIxyYGB-Jt', 'Pop rock', 'Electronic', 220, 415609909, 8),  
(NULL, 'Someday', 'https://www.youtube.com/watch?v=ayxLRvQOni0&list=PL0a-3E47mvaZsK839s\_pQSICTPnfKGGT9&index=5', 'Jazz', 'Swing', 640, 350000, 15),  
(NULL, 'Ten Million',  
'https://www.youtube.com/watch?v=24cCemVIIcY&list=PLCJjdqofvBsUJzotAzzXSxleSnjZpUCXp&index=10', 'Hip-hop', 'Electronic', 179, 1100000, 5),  
(NULL, 'In My Feelings', 'https://www.youtube.com/watch?v=DRS\_PpOrUZ4', 'Hip-Hop', 'Electronic', 217, 160000000, 1),  
(NULL, 'Stairway To Heaven', 'https://www.youtube.com/watch?v=QkF3oxziUI4', 'Rock', 'Electronic', 481, 56000000, 2),  
(NULL, 'All of Me', 'https://www.youtube.com/watch?v=450p7goxZqg', 'Pop', 'Electronic', 468, 2000000, 12),  
(NULL, 'Sahara', 'https://www.youtube.com/watch?v=q9HCGqByjak', 'Pop', 'Electronic', 679, 125000, 14),  
(NULL, 'Nonstop',  
'https://www.youtube.com/watch?v=XNpGNykVZ6U&list=PLnHe1KWzAptFf5UTOSuZWscI2EXz\_PHF&index=3', 'Hip-hop', 'Electronic', 389, 203200430, 9),  
(NULL, 'Zoom', 'https://www.youtube.com/watch?v=H-O\_mmDc4Xc', 'Hip-hop', 'Electronic', 325, 20200300, 10),  
(NULL, 'But Not For Me', 'https://www.youtube.com/watch?v=TUVdBKsgq2o', 'Jazz', 'Electronic', 232, 125000, 13);

```

INSERT INTO reviews (id, rating, comment, review_date, user_id, album_id)
VALUES (NULL, 8.5, 'Great album, loved it!', '2023-02-20', 1, 1),
       (NULL, 7.2, 'Not their best work, but still enjoyable', '2023-02-21', 2, 3),
       (NULL, 9.1, 'Incredible album, definitely a masterpiece', '2023-03-10', 3, 4),
       (NULL, 5.5, 'Not my cup of tea, but some may enjoy it', '2023-03-15', 4, 5),
       (NULL, 6.9, 'Solid album, nothing too groundbreaking', '2023-03-20', 5, 6),
       (NULL, 8.8, 'One of my favorite albums of all time', '2023-03-22', 6, 7),
       (NULL, 4.3, 'Disappointing album, expected more from this artist', '2023-03-25', 7, 8),
       (NULL, 9.5, 'Absolutely amazing album, highly recommend', '2023-04-05', 8, 9),
       (NULL, 7.8, 'Good album, but not as good as their previous work', '2023-04-10', 9, 10);

```

```

INSERT INTO sales (sale_date, sale_price, orders_id)
VALUES ('2023-04-01', 120.99, 1),
       ('2023-04-03', 35.99, 2),
       ('2023-04-05', 9.99, 3),
       ('2023-04-07', 140.99, 4),
       ('2023-04-09', 50.99, 5),
       ('2023-04-10', 52.99, 6),
       ('2023-04-10', 100.99, 5),
       ('2023-04-11', 89.09, 8),
       ('2023-04-11', 56.99, 5),
       ('2023-04-11', 30.20, 9);

```

```

INSERT INTO performer_song
VALUES (4, 1),
       (4, 9),
       (10, 6),
       (11, 6),
       (5, 7),
       (4, 3),
       (8, 4),
       (5, 5);

```

```

INSERT INTO performer_genre
VALUES (1, 3),
       (1, 4),
       (11, 4),
       (2, 4);

```

```
INSERT INTO song_genre
```

```
VALUES (5, 1),  
       (5, 2),  
       (8, 3),  
       (6, 3),  
       (10, 4),  
       (2, 3),  
       (7, 4),  
       (9, 5),  
       (4, 6);
```

```
INSERT INTO composer_song
```

```
VALUES (3, 9),  
       (3, 11),  
       (10, 2),  
       (11, 2),  
       (9, 1),  
       (8, 1);
```

```
INSERT INTO album_genre
```

```
VALUES (1, 3),  
       (1, 4),  
       (5, 6),  
       (6, 6);
```

```
INSERT INTO playlist_album
```

```
VALUES (2, 1),  
       (2, 13),  
       (1, 2),  
       (3, 2);
```

```
INSERT INTO playlist_user
```

```
VALUES (2, 3),  
       (2, 4),  
       (7, 5),  
       (8, 5);
```

```
INSERT INTO user_album
VALUES (10, 1),
      (10, 2),
      (1, 3),
      (3, 4),
      (8, 2);
```

```
INSERT INTO sale_album
VALUES (1, 1),
      (2, 2),
      (2, 7),
      (3, 4),
      (4, 5),
      (5, 3),
      (10, 8),
      (3, 8),
      (9, 9),
      (8, 9),
      (4, 9),
      (7, 9),
      (4, 8),
      (3, 8),
      (7, 8),
      (4, 5);
```

## 2. Напишете заявка, в която демонстрирате SELECT с ограничаващо условие по избор.

Ще изведем информация за всички немски композитори.

```
SELECT * FROM composer
WHERE nationality = 'German';
```

	id	name	egn	nationality	phone	dateOfBirth
▶	1	Hanz Zimmer	0912934563	German	455-9009	1957-09-12
	3	Toots Thielemans	0067890123	German	NULL	1922-04-29
*	NULL	NULL	NULL	NULL	NULL	NULL

**3. Напишете заявка, в която използвате агрегатна функция и GROUP BY по ваш избор.**

Ще изведем общата продажбена сума за всеки ден, в който има продажби.

```
SELECT sale_date, ROUND(SUM(sale_price), 2) AS totalSum
FROM sales
GROUP BY sale_date;
```

	sale_date	totalSum
►	2023-04-01	120.99
	2023-04-03	35.99
	2023-04-05	9.99
	2023-04-07	140.99
	2023-04-09	50.99
	2023-04-10	153.98
	2023-04-11	176.28

**4. Напишете заявка, в която демонстрирате INNER JOIN по ваш избор.**

Ще напишем заявка, с която ще изведем името на всеки потребител, коментара, който е оставил, и името на албума, на който е написан коментара. За да реализираме това условие свързваме таблиците : 'user', 'reviews' и 'albums'.

```
SELECT user.username, reviews.comment, albums.title AS albumTitle
FROM user
JOIN reviews ON user.id = reviews.user_id
JOIN albums ON reviews.album_id = albums.id;
```

	username	comment	albumTitle
►	John Brown	Great album, loved it!	High Off Life
	Jane Lee	Not their best work, but still enjoyable	Divide
	Oleg Smirnov	Incredible album, definitely a masterpiece	GEMINI
	Sara Ali	Not my cup of tea, but some may enjoy it	The Orange Room
	Mike Durant	Solid album, nothing too groundbreaking	7
	Bob Wilson	One of my favorite albums of all time	Thriller
	Ana Petkova	Disappointing album, expected more from this artist	Kind of Blue
	Robert James	Absolutely amazing album, highly recommend	Scorpion
	Ekaterina Valova	Good album, but not as good as their previous work	Future

## 5. Напишете заявка, в която демонстрирате OUTER JOIN по ваш избор.

Ще напишем заявка, която ще изведе информация за всички албуми в базата данни, включително имената на техните изпълнители, независимо дали изпълнителите имат собствен албум или не. Таблиците, които ще са ни необходими за това условие са : 'albums' и 'performer'. За да ги свържем, ще използваме RIGHT OUTER JOIN, с помощта на който ще получим в резултатната таблица и изпълнителите, които нямат собствени албуми.

```
SELECT albums.title AS albumName,
albums.numberOfSongs,
albums.length,
albums.release_date,
albums.record_label,
performer.name AS performerName
FROM albums
RIGHT OUTER JOIN performer
ON albums.performer_id = performer.id;
```

	albumName	numberOfSongs	length	release_date	record_label	performerName
►	High Off Life	21	80.20	2020-05-15	Republic Records	Drake
	Scorpion	25	89.04	2018-06-29	Cash Money Records	Drake
	Future	19	66.14	2017-02-17	Interscope Records	Future
	Are You Experienced	8	42.39	1969-10-22	Atlantic Records	Jimi Hendrix
	Divide	11	50.03	2017-03-03	Asylum	Ed Sheeran
	The Orange Room	15	55.29	2005-03-01	Sheeran Lock	Ed Sheeran
	Journeyman	12	42.32	1938-08-19	XL Recordings	Eric Clapton
	Hello Dolly	11	40.24	1938-08-19	Warner Music	Louis Armstrong
	7	15	43.37	2021-09-17	Columbia Records	Lil Nas X
	Thriller	9	42.19	1982-11-30	Epic Records	Michael Jackson
	Kind of Blue	5	45.52	1959-08-17	Columbia Records	Miles Davis
	GEMINI	16	60.13	2017-09-22	Bendo LLC	Macklemore
	Graduation	13	51.23	2008-08-19	KonLive	Kanye West
	Wake Up	13	51.13	2008-08-19	Domino Recording C...	John Legend
	All or Nothing at All	15	32.48	1938-08-19	Sony Music	Billie Holiday
	The Elephants of Mars	10	59.04	1938-08-19	Universal Music Group	Joseph Satriani
	NULL	NULL	NULL	NULL	NULL	21 Savage
	NULL	NULL	NULL	NULL	NULL	Aggro Santoss
	NULL	NULL	NULL	NULL	NULL	Common
	NULL	NULL	NULL	NULL	NULL	Cardi B
	NULL	NULL	NULL	NULL	NULL	Ceza
	NULL	NULL	NULL	NULL	NULL	Tim Dog

## 6. Напишете заявка, в която демонстрирате вложен SELECT по ваш избор.

Ще напишем заявка, с която ще изведем името на песента, и жанра на който тя принадлежи. Таблиците, които ще трябва да свържем за да реализираме това условие и съответно демонстрираме вложения SELECT са : 'song', 'genre' и 'song\_genre'.

```
SELECT song.name AS songName, genre.name AS genreName
FROM song JOIN genre
ON song.id IN (
    SELECT song_id
    FROM song_genre
    WHERE song_genre.genre_id = genre.id
);
```



	songName	genreName
►	Thrift Shop	Pop
	Thrift Shop	Pop-punk
	Old Town Road	Hip-hop
	Fire	Hip-hop
	Life Is Good	Hip-hop
	Sweet Child O Mine	Trip-hop
	Billie Jean	Trip-hop
	Addicted	Jazz
	Stronger	Jazz fusion

## 7. Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функция.

Ще напишем заявка, която ще извежда името на албума, броя на продажбите на албума и общата сума на продажбите, чиято стойност е над 100 лева. Резултатите ще подредим по общите стойности на продажбите в намаляващ ред и ще ги ограничим до първите три.

```
SELECT albums.title AS nameOfAlbum, COUNT(sale_album.sale_id) AS sales_count,
ROUND(SUM(sales.sale_price), 2) AS total_sales
FROM albums
JOIN sale_album ON albums.id = sale_album.album_id
JOIN sales ON sale_album.sale_id = sales.id
GROUP BY albums.id
HAVING total_sales > 100
ORDER BY total_sales DESC
LIMIT 3;
```

	nameOfAlbum	sales_count	total_sales
►	Scorpion	4	388.06
	Kind of Blue	5	292.16
	The Orange Room	2	281.98

## 8. Създайте тригер по ваш избор.

Ще създадем тригер, който да проверява броя на песните в албума и да увеличава цената автоматично, ако броят на песните надвишава определен лимит – 10.

```

DELIMITER //
CREATE TRIGGER increase_album_price
AFTER INSERT ON song
FOR EACH ROW
BEGIN
    DECLARE album_songs INT;
    DECLARE album_price FLOAT;
    SELECT numberOfSongs, price INTO album_songs,
    album_price FROM albums WHERE id = NEW.album_id;
    IF album_songs >= 10 THEN
        UPDATE albums
        SET price = album_price * 1.5 WHERE id = NEW.album_id;
    ELSEIF album_songs < 10 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Album must have at least 10 songs';
    ELSEIF album_songs IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid number of songs in album';
    END IF;
END//
DELIMITER ;

INSERT INTO song
VALUES (NULL, 'Shape Of You', 'https://www.youtube.com/watch?v=JGwWNGJdvx8', 'Pop', 'R&B',
263, 59000000000, 3),
(NULL, 'Zoom', 'https://www.youtube.com/watch?v=q9HCGqByjak', 'Hip-hop', 'Electronic',
325, 20200300, 10);

```

Първоначалната цена на албума :

	id	title	numberOfSongs	length	release_date	record_label	price	performer_id
▶	1	High Off Life	21	80.20	2020-05-15	Republic Records	250	1
	2	Are You Experienced	8	42.39	1969-10-22	Atlantic Records	50	3
	3	Divide	11	50.03	2017-03-03	Asylum	150	4
	4	GEMINI	16	60.13	2017-09-22	Bendo LLC	600	10
	5	The Orange Room	15	55.29	2005-03-01	Sheeran Lock	110	4
	6	7	15	43.37	2021-09-17	Columbia Records	120	7
	7	Thriller	9	42.19	1982-11-30	Epic Records	100	8
	8	Kind of Blue	5	45.52	1959-08-17	Columbia Records	400	9
	9	Scorpion	25	89.04	2018-06-29	Cash Money Records	450	1
	10	Future	19	66.14	2017-02-17	Interscope Records	300	2
	11	Graduation	13	51.23	2008-08-19	KonLive	250	11
	12	Wake Up	13	51.13	2008-08-19	Domino Recording C...	120	12
	13	All or Nothing at All	15	32.48	1938-08-19	Sony Music	30	13
	14	The Elephants of Mars	10	59.04	1938-08-19	Universal Music Group	150	14
	15	Hello Dolly	11	40.24	1938-08-19	Warner Music	150	6
	16	Journeyman	12	42.32	1938-08-19	XL Recordings	100	5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Цената на албума след проверката :

	id	title	numberOfSongs	length	release_date	record_label	price	performer_id
▶	1	High Off Life	21	80.20	2020-05-15	Republic Records	250	1
	2	Are You Experienced	8	42.39	1969-10-22	Atlantic Records	50	3
	3	Divide	11	50.03	2017-03-03	Asylum	225	4
	4	GEMINI	16	60.13	2017-09-22	Bendo LLC	600	10
	5	The Orange Room	15	55.29	2005-03-01	Sheeran Lock	110	4
	6	7	15	43.37	2021-09-17	Columbia Records	120	7
	7	Thriller	9	42.19	1982-11-30	Epic Records	100	8
	8	Kind of Blue	5	45.52	1959-08-17	Columbia Records	400	9
	9	Scorpion	25	89.04	2018-06-29	Cash Money Records	450	1
	10	Future	19	66.14	2017-02-17	Interscope Records	450	2
	11	Graduation	13	51.23	2008-08-19	KonLive	250	11
	12	Wake Up	13	51.13	2008-08-19	Domino Recording C...	120	12
	13	All or Nothing at All	15	32.48	1938-08-19	Sony Music	30	13
	14	The Elephants of Mars	10	59.04	1938-08-19	Universal Music Group	150	14
	15	Hello Dolly	11	40.24	1938-08-19	Warner Music	150	6
	16	Journeyman	12	42.32	1938-08-19	XL Recordings	100	5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ако добавим песен, която се намира в албум, който има по малко от 10 песни :

```
INSERT INTO song
VALUES (NULL, 'Fire',
'https://www.youtube.com/watch?v=cbG7HEEPEIo&list=PL1RRNXldMeRFm7CqtN44oav-TgaWT3yt_',
'Hard rock', 'Electronic', 278, 4500989, 2);
```

1139 16:28:54 INSERT INTO song VALUES (NULL, 'Shape Of You', 'https://www.youtube.com/watch?v=JGwvWNGJdvx8', 'Pop', 'R&B', 263, 590000000... Error Code: 1644. Album must have at least 10 songs

## 9. Създайте процедура, в която демонстрирате използване на курсор.

Ще създадем процедура, която създава плейлиста с песни от подаден жанр и извежда информация за всички песни от този жанр, които имат повече от 1 милион гледания, сортирани в намаляващ ред по продължителност, заедно с името на албума в който са песните.

Ще добавим и проверка дали подадения жанр съответства на жанровете в базата данни. Ако не съответства, ще изведем съобщение за грешка.

DELIMITER |

```
CREATE PROCEDURE createPlaylistByGenre(IN genreName VARCHAR(255))
BEGIN
    DECLARE numViewsThreshold INT DEFAULT 1000000;
    DECLARE tempSongName VARCHAR(255);
    DECLARE tempSongLink VARCHAR(255);
    DECLARE tempSongStyle VARCHAR(255);
    DECLARE tempSongArrangement VARCHAR(255);
    DECLARE tempSongDuration INT;
    DECLARE tempSongNumberOfViews INT;
    DECLARE tempAlbumName VARCHAR(255);
    DECLARE cursorFinished INT DEFAULT 0;
    DECLARE genreExists INT DEFAULT 0;
    DECLARE tempCursor CURSOR FOR
```

```
SELECT song.name, song.link, song.style, song.arrangement, song.duration, song.numberOfViews,
albums.title
```

```
FROM song
JOIN song_genre ON song.id = song_genre.song_id
JOIN genre ON song_genre.genre_id = genre.id
JOIN albums ON song.album_id = albums.id
WHERE genre.name = genreName AND song.numberOfViews > numViewsThreshold
ORDER BY song.duration DESC;
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET cursorFinished = 1;
```

```
SELECT COUNT(*) INTO genreExists FROM genre WHERE name = genreName;
```

```
IF genreExists = 0 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid genre';
```

```
ELSE
    DROP TABLE IF EXISTS tempPlaylist;
    CREATE TEMPORARY TABLE tempPlaylist (
        name VARCHAR(255) NOT NULL,
        link VARCHAR(255) NOT NULL,
        style VARCHAR(100) NOT NULL,
        arrangement VARCHAR(100) NOT NULL,
        duration INT UNSIGNED NOT NULL,
        numberOfViews BIGINT UNSIGNED NOT NULL,
        album_name VARCHAR(255) NOT NULL
    ) Engine = Memory;
```

```

OPEN tempCursor;
cursorLoop: LOOP
    FETCH tempCursor INTO tempSongName, tempSongLink, tempSongStyle,
tempSongArrangement, tempSongDuration, tempSongNumberOfViews, tempAlbumName;
    IF cursorFinished THEN
        LEAVE cursorLoop;
    END IF;

    INSERT INTO tempPlaylist(name, link, style, arrangement, duration, numberOfViews,
album_name)
        VALUES (tempSongName, tempSongLink, tempSongStyle, tempSongArrangement,
tempSongDuration, tempSongNumberOfViews, tempAlbumName);
    END LOOP;

CLOSE tempCursor;

SELECT * FROM tempPlaylist
ORDER BY duration DESC;

DROP TABLE tempPlaylist;
END IF;

END;
|

DELIMITER ;

CALL createPlaylistByGenre('Hip-hop');

```

	name	link	style	arrangement	duration	numberOfViews	album_name
►	Fire	<a href="https://www.youtube.com/watch?v=cbG7HEEPe1o&amp;list=PL1RRNXldMeRFm7Cqtn44oav-TgaWT3yt_">https://www.youtube.com/watch?v=cbG7HEEPe1o&amp;list=PL1RRNXldMeRFm7Cqtn44oav-TgaWT3yt_</a>	Hard rock	Electronic	278	4500989	Are You Experienced
	Life Is Good	<a href="https://www.youtube.com/watch?v=l0U7SxxHkPY">https://www.youtube.com/watch?v=l0U7SxxHkPY</a>	Hip-hop	Trap	237	2100050978	High Off Life
	Old Town Road	<a href="https://www.youtube.com/watch?v=w2Ov5jzm3j8">https://www.youtube.com/watch?v=w2Ov5jzm3j8</a>	Pop	Electronic	157	345600000	7

При грешно подаден жанр :

```
CALL createPlaylistByGenre('Folk');
```

3864 20:53:17 CALL createPlaylistByGenre('Folk')

Error Code: 1644, Invalid genre