

**КУРСОВА РАБОТА****ПО БАЗИ ОТ ДАННИ**

Студент: .....

ФАК. № ..... Група:.....

**Тема**

Напишете база данни за сайт за продажба на песни. Да се пази основна информация за композитора, изпълнителя и песента – жанр, стил, аранжировање, продължителност. Една песен може да бъде написана от повече от един композитор и изпълнявана от повече от един изпълнител.

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 и OUTER JOIN по ваш избор.
5. Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функция.
6. Създайте процедура, в която демонстрирате използване на курсор.

*Вашата работа трябва да включва: задание, ER-диаграма, CREATE TABLE заявки, всички останали заявки, решения на задачите от 2 до 5 и резултатите от тях.*

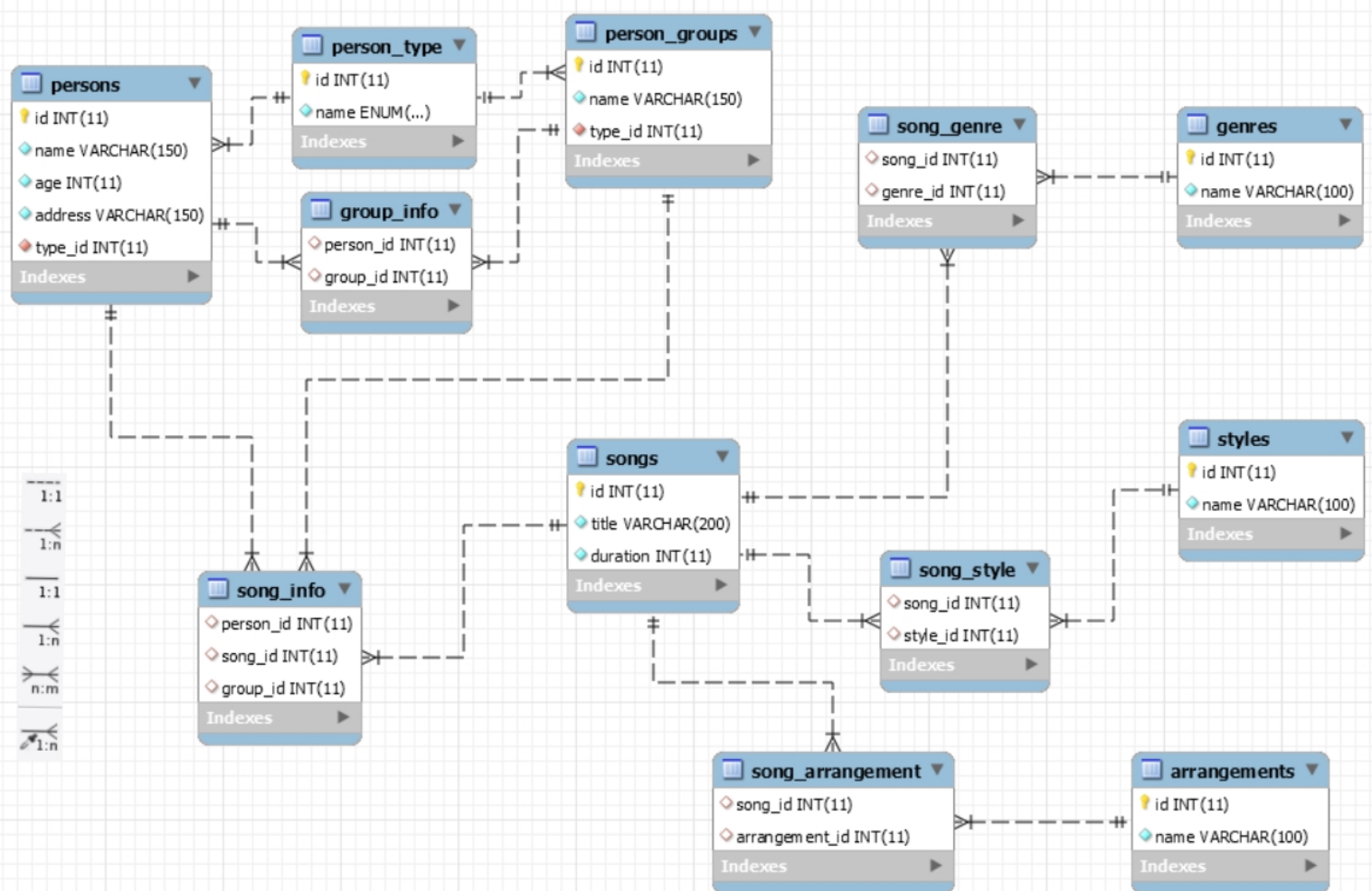
София,  
2017/2018Подпис на преподавателя:.....  
(ас. Петко Данов)

Подпис на студента:.....

# Решения

За решение на задачата ще използваме езика **MySQL**, който се изучава по време на лабораторните упражнения по дисциплината. За проектирането на базата в задача 1 ще използваме модела **ER**-диаграма (Entity Relationship Diagram).

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



## Създаваме таблица person\_type:

```
CREATE TABLE person_type (  
id INT AUTO_INCREMENT PRIMARY KEY,  
name ENUM('Composer', 'Artist', 'Group') NOT NULL UNIQUE  
);
```

### Вкарваме примерни данни за person\_type:

```
INSERT INTO person_type (name) VALUES ('Artist'); -- ID = 1
INSERT INTO person_type (name) VALUES ('Composer'); -- ID = 2
INSERT INTO person_type (name) VALUES ('Group'); -- ID = 3
```

```
mysql> SELECT * FROM person_type;
+----+-----+
| id | name      |
+----+-----+
| 2  | Composer  |
| 1  | Solo Artist |
| 3  | Group     |
+----+-----+
3 rows in set (0.00 sec)
```

### Създаваме таблица persons:

```
CREATE TABLE persons (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(150) NOT NULL,
age INT NOT NULL,
address VARCHAR(150) NOT NULL,
type_id INT NOT NULL,
CONSTRAINT FOREIGN KEY (type_id) REFERENCES person_type(id)
);
```

### Вкарваме примерни данни за persons:

```
/* INSERT заявки за добавяне на изпълнители */
```

```
INSERT INTO persons (name, age, address, type_id) VALUES ('50 Cent', 42, '50 Cent address', 1); -- ID = 1
INSERT INTO persons (name, age, address, type_id) VALUES ('Ice Cube', 48, 'Ice Cube address', 1); -- ID = 2
INSERT INTO persons (name, age, address, type_id) VALUES ('Taylor Swift', 28, 'Taylor Swift address', 1); -- ID = 3
INSERT INTO persons (name, age, address, type_id) VALUES ('Bon Jovi', 35, 'Bon Jovi address', 1); -- ID = 4
INSERT INTO persons (name, age, address, type_id) VALUES ('Eminem', 35, 'Eminem address', 1); -- ID = 5
INSERT INTO persons (name, age, address, type_id) VALUES ('Takeoff', 23, 'Takeoff address', 1); -- ID = 6
INSERT INTO persons (name, age, address, type_id) VALUES ('Offset', 26, 'Offset address', 1); -- ID = 7
INSERT INTO persons (name, age, address, type_id) VALUES ('Quavo', 27, 'Quavo address', 1); -- ID = 8
```

```
/* INSERT заявки за добавяне на композитори */
```

```
INSERT INTO persons (name, age, address, type_id) VALUES ('Lars Winther', 38, 'Lars Winther address', 2); -- ID = 9
```

```
INSERT INTO persons (name, age, address, type_id) VALUES ('Jacob Cooper', 28, 'Jacob Cooper address', 2); -- ID = 10
INSERT INTO persons (name, age, address, type_id) VALUES ('Daniel Hensel', 40, 'Jacob Cooper address', 2); -- ID = 11
```

```
mysql> SELECT * FROM persons;
```

id	name	age	address	type_id
1	50 Cent	42	50 Cent address	1
2	Ice Cube	48	Ice Cube address	1
3	Taylor Swift	28	Taylor Swift address	1
4	Bon Jovi	35	Bon Jovi address	1
5	Eminem	35	Eminem address	1
6	Takeoff	23	Takeoff address	1
7	Offset	26	Offset address	1
8	Quavo	27	Quavo address	1
9	Lars Winther	38	Lars Winther address	2
10	Jacob Cooper	28	Jacob Cooper address	2
11	Daniel Hensel	40	Daniel Hensel address	2

11 rows in set (0.00 sec)

### Създаваме таблица person\_groups:

```
CREATE TABLE person_groups (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(150) NOT NULL UNIQUE,
type_id INT NOT NULL,
CONSTRAINT FOREIGN KEY (type_id) REFERENCES person_type(id)
);
```

### Вкарваме примерни данни за person\_groups:

```
INSERT INTO person_groups (name, type_id) VALUES ('Migos', 3); -- ID = 1
```

```
mysql> SELECT * FROM person_groups;
```

id	name	type_id
1	Migos	3

1 row in set (0.00 sec)

### Създаваме таблица group\_info:

```
CREATE TABLE group_info (  
  person_id INT,  
  group_id INT,  
  CONSTRAINT FOREIGN KEY (person_id) REFERENCES persons(id),  
  CONSTRAINT FOREIGN KEY (group_id) REFERENCES person_groups(id)  
);
```

### Вкарваме примерни данни за group\_info:

```
INSERT INTO group_info (person_id, group_id) VALUES (6, 1);  
INSERT INTO group_info (person_id, group_id) VALUES (7, 1);  
INSERT INTO group_info (person_id, group_id) VALUES (8, 1);
```

```
mysql> SELECT * FROM group_info;  
+-----+-----+  
| person_id | group_id |  
+-----+-----+  
|          6 |         1 |  
|          7 |         1 |  
|          8 |         1 |  
+-----+-----+  
3 rows in set (0.00 sec)
```

### Създаваме таблица songs:

```
CREATE TABLE songs (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(200) NOT NULL,  
  duration INT NOT NULL  
);
```

### Вкарваме примерни данни за songs:

```
INSERT INTO songs (title, duration) VALUES ('Pilot', 182); -- ID = 1  
INSERT INTO songs (title, duration) VALUES ('Gangsta Rap Made Me Do It', 350); -- ID = 2  
INSERT INTO songs (title, duration) VALUES ('Animal Ambition', 200); -- ID = 3  
INSERT INTO songs (title, duration) VALUES ('Im The Man', 351); -- ID = 4  
INSERT INTO songs (title, duration) VALUES ('Look What You Made Me Do', 249); -- ID = 5  
INSERT INTO songs (title, duration) VALUES ('Its My Life', 220); -- ID = 6  
INSERT INTO songs (title, duration) VALUES ('Shake It Off', 300); -- ID = 7  
INSERT INTO songs (title, duration) VALUES ('Not Afraid', 290); -- ID = 8  
INSERT INTO songs (title, duration) VALUES ('In Your Head', 320); -- ID = 9  
INSERT INTO songs (title, duration) VALUES ('Talkin 2 Myself', 400); -- ID = 10
```

```

INSERT INTO songs (title, duration) VALUES ('Trapstar', 140); -- ID = 11
INSERT INTO songs (title, duration) VALUES ('Growth', 180); -- ID = 12
INSERT INTO songs (title, duration) VALUES ('Intruder', 134); -- ID = 13
INSERT INTO songs (title, duration) VALUES ('Slippery', 320); -- ID = 14
INSERT INTO songs (title, duration) VALUES ('Get Right Witcha', 274); -- ID = 15

```

```
mysql> SELECT * FROM songs;
```

id	title	duration
1	Pilot	182
2	Gangsta Rap Made Me Do It	350
3	Animal Ambition	200
4	Im The Man	351
5	Look What You Made Me Do	249
6	Its My Life	220
7	Shake It Off	300
8	Not Afraid	290
9	In Your Head	320
10	Talkin 2 Myself	400
11	Trapstar	140
12	Growth	180
13	Intruder	134
14	Slippery	320
15	Get Right Witcha	274

```
15 rows in set (0.00 sec)
```

### Създаваме таблица song\_info:

```

CREATE TABLE song_info (
  person_id INT,
  song_id INT,
  group_id INT,
  CONSTRAINT FOREIGN KEY (person_id) REFERENCES persons(id),
  CONSTRAINT FOREIGN KEY (song_id) REFERENCES songs(id),
  CONSTRAINT FOREIGN KEY (group_id) REFERENCES person_groups(id)
);

```

### Вкарваме примерни данни за song\_info:

```

/* INSERT заявки за свърване на песен към даден изпълнител | група */
INSERT INTO song_info (person_id, song_id) VALUES (1, 1);
INSERT INTO song_info (person_id, song_id) VALUES (1, 3);
INSERT INTO song_info (person_id, song_id) VALUES (1, 4);
INSERT INTO song_info (person_id, song_id) VALUES (2, 2);
INSERT INTO song_info (person_id, song_id) VALUES (3, 5);
INSERT INTO song_info (person_id, song_id) VALUES (3, 7);
INSERT INTO song_info (person_id, song_id) VALUES (4, 6);

```

```

INSERT INTO song_info (person_id, song_id) VALUES (5, 8);
INSERT INTO song_info (person_id, song_id) VALUES (5, 9);
INSERT INTO song_info (person_id, song_id) VALUES (5, 10);
INSERT INTO song_info (person_id, song_id) VALUES (6, 13);
INSERT INTO song_info (person_id, song_id) VALUES (7, 12);
INSERT INTO song_info (person_id, song_id) VALUES (8, 11);
INSERT INTO song_info (group_id, song_id) VALUES (1, 14);
INSERT INTO song_info (group_id, song_id) VALUES (1, 15);
/* INSERT заявки за свърване на песен към даден композитор */
INSERT INTO song_info (person_id, song_id) VALUES (9, 5);
INSERT INTO song_info (person_id, song_id) VALUES (9, 7);
INSERT INTO song_info (person_id, song_id) VALUES (10, 6);

```

```

mysql> SELECT * FROM song_info;
+-----+-----+-----+
| person_id | song_id | group_id |
+-----+-----+-----+
|          1 |          1 | NULL |
|          1 |          3 | NULL |
|          1 |          4 | NULL |
|          2 |          2 | NULL |
|          3 |          5 | NULL |
|          3 |          7 | NULL |
|          4 |          6 | NULL |
|          5 |          8 | NULL |
|          5 |          9 | NULL |
|          5 |         10 | NULL |
|          6 |         13 | NULL |
|          7 |         12 | NULL |
|          8 |         11 | NULL |
|        NULL |         14 |      1 |
|        NULL |         15 |      1 |
|          9 |          5 | NULL |
|          9 |          7 | NULL |
|         10 |          6 | NULL |
+-----+-----+-----+
18 rows in set (0.00 sec)

```

### Създаваме таблица arrangements:

```

CREATE TABLE arrangements (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL UNIQUE
);

```

### Вкарваме примерни данни за arrangements:

```

INSERT INTO arrangements (name) VALUES ('Electronic'); -- ID = 1
INSERT INTO arrangements (name) VALUES ('Jazz'); -- ID = 2

```



```
mysql> SELECT * FROM arrangements;
+-----+-----+
| id | name      |
+-----+-----+
| 1  | Electronic|
| 2  | Jazz      |
+-----+-----+
2 rows in set (0.00 sec)
```

### Създаваме таблица song\_arrangement:

```
CREATE TABLE song_arrangement (
  song_id INT,
  arrangement_id INT,
  CONSTRAINT FOREIGN KEY (song_id) REFERENCES songs(id),
  CONSTRAINT FOREIGN KEY (arrangement_id) REFERENCES arrangements(id)
);
```

### Вкарваме примерни данни за song\_arrangement:

```
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (1, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (2, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (3, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (4, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (5, 2);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (6, 2);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (7, 2);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (8, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (9, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (10, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (11, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (12, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (13, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (14, 1);
INSERT INTO song_arrangement (song_id, arrangement_id) VALUES (15, 1);
```



```
mysql> SELECT * FROM song_arrangement;
```

song_id	arrangement_id
1	1
2	1
3	1
4	1
5	2
6	2
7	2
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

```
15 rows in set (0.00 sec)
```

### Създаваме таблица genres:

```
CREATE TABLE genres (  
id INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(100) NOT NULL UNIQUE  
);
```

### Вкарваме примерни данни за genres:

```
INSERT INTO genres (name) VALUES ('Hip-Hop'); -- ID = 1  
INSERT INTO genres (name) VALUES ('Pop'); -- ID = 2  
INSERT INTO genres (name) VALUES ('Rock'); -- ID = 3
```

```
mysql> SELECT * FROM genres;
```

id	name
1	Hip-Hop
2	Pop
3	Rock

```
3 rows in set (0.00 sec)
```

### Създаваме таблица song\_genre:

```
CREATE TABLE song_genre (  
  song_id INT,  
  genre_id INT,  
  CONSTRAINT FOREIGN KEY (song_id) REFERENCES songs(id),  
  CONSTRAINT FOREIGN KEY (genre_id) REFERENCES genres(id)  
);
```

### Вкарваме примерни данни за song\_genre:

```
INSERT INTO song_genre (song_id, genre_id) VALUES (1, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (2, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (3, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (4, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (5, 2);  
INSERT INTO song_genre (song_id, genre_id) VALUES (6, 3);  
INSERT INTO song_genre (song_id, genre_id) VALUES (7, 2);  
INSERT INTO song_genre (song_id, genre_id) VALUES (8, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (9, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (10, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (11, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (12, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (13, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (14, 1);  
INSERT INTO song_genre (song_id, genre_id) VALUES (15, 1);
```

```
mysql> SELECT * FROM song_genre;
```

song_id	genre_id
1	1
2	1
3	1
4	1
5	2
6	3
7	2
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

```
15 rows in set (0.00 sec)
```

### Създаваме таблица styles:

```
CREATE TABLE styles (  
id INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(100) NOT NULL UNIQUE  
);
```

### Вкарваме примерни данни за styles:

```
INSERT INTO styles (name) VALUES ('Dance'); -- ID = 1  
INSERT INTO styles (name) VALUES ('Sad'); -- ID = 2  
INSERT INTO styles (name) VALUES ('Funky'); -- ID = 3  
INSERT INTO styles (name) VALUES ('Deep'); -- ID = 4
```

```
mysql> SELECT * FROM styles;  
+-----+-----+  
| id | name |  
+-----+-----+  
| 1 | Dance |  
| 4 | Deep |  
| 3 | Funky |  
| 2 | Sad |  
+-----+-----+  
4 rows in set (0.00 sec)
```

### Създаваме таблица song\_style:

```
CREATE TABLE song_style (  
song_id INT,  
style_id INT,  
CONSTRAINT FOREIGN KEY (song_id) REFERENCES songs(id),  
CONSTRAINT FOREIGN KEY (style_id) REFERENCES styles(id)  
);
```

### Вкарваме примерни данни за song\_style:

```
INSERT INTO song_style (song_id, style_id) VALUES (1, 1);  
INSERT INTO song_style (song_id, style_id) VALUES (2, 4);  
INSERT INTO song_style (song_id, style_id) VALUES (3, 1);  
INSERT INTO song_style (song_id, style_id) VALUES (4, 4);  
INSERT INTO song_style (song_id, style_id) VALUES (5, 1);  
INSERT INTO song_style (song_id, style_id) VALUES (6, 1);  
INSERT INTO song_style (song_id, style_id) VALUES (7, 3);  
INSERT INTO song_style (song_id, style_id) VALUES (8, 4);  
INSERT INTO song_style (song_id, style_id) VALUES (9, 4);  
INSERT INTO song_style (song_id, style_id) VALUES (10, 4);  
INSERT INTO song_style (song_id, style_id) VALUES (11, 1);  
INSERT INTO song_style (song_id, style_id) VALUES (12, 1);  
INSERT INTO song_style (song_id, style_id) VALUES (13, 1);
```

```
INSERT INTO song_style (song_id, style_id) VALUES (14, 3);
INSERT INTO song_style (song_id, style_id) VALUES (15, 1);
```

```
mysql> SELECT * FROM song_style;
```

song_id	style_id
1	1
2	4
3	1
4	4
5	1
6	1
7	3
8	4
9	4
10	4
11	1
12	1
13	1
14	3
15	1

```
15 rows in set (0.00 sec)
```

**Задача 2:** Заявката извежда имената на песните, продължителността им и годините на изпълнителя. Ограничаващото условие е изпълнителя да е с име "Eminem".

```
SELECT persons.name AS Artist, persons.age AS Age, songs.title AS Song, songs.duration AS Duration
FROM persons
JOIN songs ON persons.id IN (
    SELECT person_id
    FROM song_info
    WHERE song_id = songs.id)
WHERE persons.type_id = 1 AND persons.name LIKE '%Eminem%'
ORDER BY songs.title;
```

Artist	Age	Song	Duration
Eminem	35	In Your Head	320
Eminem	35	Not Afraid	290
Eminem	35	Talkin 2 Myself	400

```
3 rows in set (0.00 sec)
```

**Задача 3:** Заявката извежда имената на изпълнителите/групите, броя на песните на всеки един от тях и сумира продължителността им, ако броя на песните им е повече от една. Резултатите са групирани по имена на изпълнител/група и са подредени по максимален брой песни.

```
SELECT persons.name AS Artist, COUNT(songs.id) AS Songs, SUM(songs.duration) AS total_duration
FROM persons
JOIN songs ON persons.id IN (
    SELECT person_id
    FROM song_info
    WHERE song_id = songs.id)
WHERE persons.type_id = 1
GROUP by persons.name
HAVING songs > 1
UNION
SELECT person_groups.name AS Artist, COUNT(songs.id) AS Songs, SUM(songs.duration) AS
total_duration
FROM person_groups
JOIN songs ON person_groups.id IN (
    SELECT group_id
    FROM group_info
    WHERE group_id IN (
        SELECT group_id
        FROM song_info
        WHERE song_id = songs.id))
GROUP by person_groups.name
HAVING songs > 1
ORDER BY songs DESC;
```

Artist	Songs	total_duration
Eminem	3	1010
50 Cent	3	733
Taylor Swift	2	549
Migos	2	594

4 rows in set (0.00 sec)

**Задача 4:** Заявката извежда имена на артистите, композиторите и имената на песните, които те изпълняват/композират.

```
/* FOURTH TASK - INNER JOIN */
```

```
SELECT persons.name AS person_name, person_type.name AS type, songs.title AS songTitle
FROM persons
INNER JOIN song_info ON persons.id = song_info.person_id
INNER JOIN songs ON song_info.song_id = songs.id
INNER JOIN person_type ON persons.type_id = person_type.id
ORDER BY person_type.name DESC;
```

person_name	type	songTitle
50 Cent	Artist	Pilot
50 Cent	Artist	Animal Ambition
50 Cent	Artist	Im The Man
Ice Cube	Artist	Gangsta Rap Made Me Do It
Taylor Swift	Artist	Look What You Made Me Do
Taylor Swift	Artist	Shake It Off
Bon Jovi	Artist	Its My Life
Eminem	Artist	Not Afraid
Eminem	Artist	In Your Head
Eminem	Artist	Talkin 2 Myself
Takeoff	Artist	Intruder
Offset	Artist	Growth
Quavo	Artist	Trapstar
Lars Winther	Composer	Look What You Made Me Do
Lars Winther	Composer	Shake It Off
Jacob Cooper	Composer	Its My Life

16 rows in set (0.02 sec)

```
/* FOURTH TASK - LEFT OUTER JOIN */
```

```
SELECT persons.name AS person_name, person_type.name AS type, songs.title AS song_title
FROM persons
LEFT JOIN songs ON persons.id IN (
    SELECT person_id
    FROM song_info
    WHERE song_id = songs.id)
JOIN person_type ON persons.type_id = person_type.id;
```

person_name	type	song_title
Lars Winther	Composer	Look What You Made Me Do
Lars Winther	Composer	Shake It Off
Jacob Cooper	Composer	Its My Life
Daniel Hensel	Composer	NULL
50 Cent	Artist	Pilot
50 Cent	Artist	Animal Ambition
50 Cent	Artist	Im The Man
Ice Cube	Artist	Gangsta Rap Made Me Do It
Taylor Swift	Artist	Look What You Made Me Do
Taylor Swift	Artist	Shake It Off
Bon Jovi	Artist	Its My Life
Eminem	Artist	Not Afraid
Eminem	Artist	In Your Head
Eminem	Artist	Talkin 2 Myself
Takeoff	Artist	Intruder
Offset	Artist	Growth
Quavo	Artist	Trapstar

17 rows in set (0.00 sec)

**Задача 5:** Заявката извежда име на композитор и брой песни, които е композирал.

```

/* FIFTH TASK */
SELECT persons.name AS Composer, Count(songs.id) AS songs
FROM persons
LEFT JOIN song_info ON persons.id = song_info.person_id
LEFT JOIN songs ON song_info.song_id = songs.id
JOIN person_type ON persons.type_id = person_type.id
WHERE persons.type_id = 2
GROUP BY persons.name;

```

Composer	songs
Lars Winther	2
Jacob Cooper	1
Daniel Hensel	0

3 rows in set (0.00 sec)



**Задача 6:** В процедурата декларираме променливи. След това и курсорът, който взима име на изпълнител/композитор/група, име на песен и нейната продължителност. След това се създава празна таблица, в която пазим резултата. Итерираме през данните от курсора и ги записваме във временната таблица. При последната итерация запазваме състоянието на цикъла в таблицата. Взимаме информацията от временната таблица и я изтриваме.

```
/* SIX TASK */
-- USE song_seller;
DROP PROCEDURE IF EXISTS CursorTask;
DELIMITER |
CREATE PROCEDURE CursorTask()
BEGIN
    DECLARE finished INT;
    DECLARE lastId INT;
    DECLARE tempArtistName VARCHAR(200);
    DECLARE tempType VARCHAR(200);
    DECLARE tempSongName VARCHAR(200);
    DECLARE tempSongDuration INT;

    DECLARE SongsCursor CURSOR FOR
    SELECT persons.name AS Name, person_type.name AS type, songs.title AS title, songs.duration AS
    song_duration
    FROM persons
    JOIN songs ON persons.id IN (
        SELECT person_id
        FROM song_info
        WHERE song_id = songs.id)
    JOIN person_type ON persons.id
    WHERE person_type.id = persons.type_id
    UNION
    SELECT person_groups.name AS Name, person_type.name AS type, songs.title AS title,
    songs.duration AS song_duration
    FROM person_groups
    JOIN songs ON person_groups.id IN (
        SELECT group_id
        FROM group_info
        WHERE group_id IN (
            SELECT group_id
            FROM song_info
            WHERE song_id = songs.id))
    JOIN person_type ON person_groups.type_id
    WHERE person_type.id = person_groups.type_id;

    DECLARE CONTINUE handler FOR NOT FOUND SET finished = 1;
    SET finished = 0;
    set lastId = 0;
    DROP TABLE IF EXISTS tempSongsInfo;
    CREATE TEMPORARY TABLE tempSongsInfo(
```

```

        id INT PRIMARY KEY AUTO_INCREMENT,
        name VARCHAR(200),
        person_type VARCHAR(100),
        songTitle VARCHAR(200),
        songDuration INT,
        finishStatus INT NULL DEFAULT 0
    ) ENGINE = Memory;
    OPEN SongsCursor;
    songs_loop: WHILE (finished = 0)
    DO
    FETCH SongsCursor INTO tempArtistName, tempType, tempSongName, tempSongDuration;
        IF (finished = 1)
    THEN
        LEAVE songs_loop;
        END IF;
    SET lastId = lastId + 1;
    INSERT INTO tempSongsInfo VALUES (lastId, tempArtistName, tempType, tempSongName,
    tempSongDuration, 0);
    END WHILE;
    CLOSE SongsCursor;
    SET finished = 0;
    UPDATE tempSongsInfo SET finishStatus = 1 WHERE id = lastId;
    SELECT * FROM tempSongsInfo;
    DROP TABLE tempSongsInfo;
    END
    |
    DELIMITER ;

    CALL CursorTask();

```

```
mysql> CALL CursorTask();
```

id	name	person_type	songTitle	songDuration	finishStatus
1	Lars Winther	Composer	Look What You Made Me Do	249	0
2	Lars Winther	Composer	Shake It Off	300	0
3	Jacob Cooper	Composer	Its My Life	220	0
4	50 Cent	Solo Artist	Pilot	182	0
5	50 Cent	Solo Artist	Animal Ambition	200	0
6	50 Cent	Solo Artist	Im The Man	351	0
7	Ice Cube	Solo Artist	Gangsta Rap Made Me Do It	350	0
8	Taylor Swift	Solo Artist	Look What You Made Me Do	249	0
9	Taylor Swift	Solo Artist	Shake It Off	300	0
10	Bon Jovi	Solo Artist	Its My Life	220	0
11	Eminem	Solo Artist	Not Afraid	290	0
12	Eminem	Solo Artist	In Your Head	320	0
13	Eminem	Solo Artist	Talkin 2 Myself	400	0
14	Takeoff	Solo Artist	Intruder	134	0
15	Offset	Solo Artist	Growth	180	0
16	Quavo	Solo Artist	Trapstar	140	0
17	Migos	Group	Slippery	320	0
18	Migos	Group	Get Right Witcha	274	1

```

18 rows in set (0.01 sec)

Query OK, 0 rows affected (0.28 sec)

```