```
============================================================SET OPERATORS=======================================
1)Find all singers who have performed songs and also worked on albums:

mysql> (SELECT singer_name FROM singer WHERE singer_id IN (SELECT singer_id FROM songs))
    -> INTERSECT
    -> (SELECT singer_name FROM singer WHERE singer_id IN (SELECT singer_id FROM album));
+-----------------+
| singer_name     |
+-----------------+
| John Smith      |
| Emily Johnson   |
| Michael Brown   |
| Sophia Williams |
| David Lee       |
| Jennifer White  |
| Robert Johnson  |
| Emma Davis      |
| Olivia Martinez |
+-----------------+
9 rows in set (0.12 sec)

2)Find all singers who have not worked on albums:

mysql> (SELECT singer_name FROM singer)
    -> EXCEPT
    -> (SELECT singer_name FROM singer WHERE singer_id IN (SELECT DISTINCT singer_id FROM album));
Empty set (0.00 sec)

3)Find all composers who have composed songs but not worked on albums:

mysql> (SELECT composer_name FROM composer WHERE composer_id IN (SELECT composer_id FROM songs))
    -> EXCEPT
    -> (SELECT composer_name FROM composer WHERE composer_id IN (SELECT DISTINCT composer_id FROM album));
Empty set (0.04 sec)

4)Find all singers or composers who have worked on albums and are also in the "Rock" genre:
mysql> (SELECT singer_name FROM singer WHERE singer_id IN (SELECT DISTINCT singer_id FROM album)
        AND singer_id IN (SELECT singer_id FROM songs WHERE genre = 'Rock'))
    -> UNION
    -> (SELECT composer_name FROM composer WHERE composer_id IN (SELECT DISTINCT composer_id FROM album)
        AND composer_id IN (SELECT composer_id FROM songs WHERE genre = 'Rock'));
+-----------------+
| singer_name     |
+-----------------+
| Sophia Williams |
| David Lee       |
+-----------------+
============================================================STRING OPERATION============================
1)Find all male singers who are also composers:
mysql> SELECT * FROM singer WHERE gender = 'Male' AND singer_name IN (SELECT composer_name FROM composer);
+-----------+---------------+--------+------------+--------------+--------------+
| singer_id | singer_name   | gender | contact    | singer_scale | singer_genre |
+-----------+---------------+--------+------------+--------------+--------------+
|      1001 | John Smith    | Male   | 1234567890 | Tenor        | Pop          |
|      1003 | Michael Brown | Male   | 5551234567 | Baritone     | Jazz         |
|      1005 | David Lee     | Male   | 4445556666 | Bass         | Hip Hop      |
+-----------+---------------+--------+------------+--------------+--------------+
3 rows in set (0.01 sec)

2)Find all songs released after 2022-01-05 and before 2023-01-01:
mysql> SELECT * FROM songs WHERE release_date > '2022-01-05' AND release_date < '2023-01-01';
+---------+--------------+-----------+-----------+-------------+-----------+----------+-------------+
| song_id | release_date | song_name | genre     | no_of_plays | singer_id | album_id | composer_id |
+---------+--------------+-----------+-----------+-------------+-----------+----------+-------------+
|       6 | 2022-01-06   | Song 6    | Blues     |          90 |      1007 |        6 |        2006 |
|       7 | 2022-01-07   | Song 7    | Country   |         110 |      1008 |        7 |        2005 |
|       8 | 2022-01-08   | Song 8    | Classical |         130 |      1002 |        8 |        2010 |
|       9 | 2022-01-09   | Song 9    | R&B       |         170 |      1010 |        9 |        2009 |
|      10 | 2022-01-10   | Song 10   | Reggae    |         140 |      1005 |       10 |        2008 |
```

```
+---------+-------------+-----------+-----------+-------------+-----------+----------+-------------+
5 rows in set (0.00 sec)

3)Find all albums with more than 150 plays:
mysql> SELECT * FROM album WHERE no_of_play > 150;
+----------+-------------+------------+------------+-----------------+---------+-----------+-------------+
| album_id | release_date | album_name | no_of_play | record_label_id | song_id | singer_id | composer_id |
+----------+-------------+------------+------------+-----------------+---------+-----------+-------------+
|        3 | 2023-03-20   | Album 3    |        200 |            1013 |       3 |      1003 |        2003 |
|        4 | 2023-04-10   | Album 4    |        180 |            1006 |       4 |      1004 |        2004 |
|        5 | 2023-05-25   | Album 5    |        220 |            1007 |       5 |      1005 |        2005 |
|        6 | 2023-06-30   | Album 6    |        190 |            1008 |       6 |      1006 |        2006 |
|        7 | 2023-07-15   | Album 7    |        160 |            1009 |       7 |      1007 |        2007 |
|        9 | 2023-09-05   | Album 9    |        240 |            1011 |       9 |      1009 |        2009 |
|       10 | 2023-10-10   | Album 10   |        170 |            1012 |      10 |      1010 |        2010 |
+----------+-------------+------------+------------+-----------------+---------+-----------+-------------+
7 rows in set (0.00 sec)

4)Find all artists (singers and composers) who are not in the "Pop" genre:

mysql> (SELECT singer_id, singer_name FROM singer WHERE singer_id NOT IN (SELECT singer_id FROM songs
        WHERE genre = 'Pop'))
    -> UNION
    -> (SELECT composer_id , composer_name FROM composer WHERE composer_id NOT IN (SELECT composer_id FROM songs

         WHERE genre = 'Pop'));
+-----------+-----------------+
| singer_id | singer_name     |
+-----------+-----------------+
|      1001 | John Smith      |
|      1003 | Michael Brown   |
|      1004 | Sophia Williams |
|      1005 | David Lee       |
|      1006 | Jennifer White  |
|      1007 | Robert Johnson  |
|      1008 | Emma Davis      |
|      1009 | Matthew Wilson  |
|      1010 | Olivia Martinez |
|      2001 | Michael Johnson |
|      2003 | Michael Brown   |
|      2004 | Sophia Williams |
|      2005 | David Lee       |
|      2006 | John Smith      |
|      2007 | Emma Davis      |
|      2008 | Andrew Johnson  |
|      2009 | Olivia Adams    |
|      2010 | William Taylor  |
+-----------+-----------------+
18 rows in set (0.01 sec)

=======================================AGGREGATE FUNCTIONS (AVG,MIN,MAX,COUNT)================================
1)Count the total number of singers in the database:

mysql> SELECT COUNT(*) AS total_singers FROM singer;
+---------------+
| total_singers |
+---------------+
|            10 |
+---------------+
1 row in set (0.04 sec)

2)Find the minimum number of plays among all songs:

mysql> SELECT MIN(no_of_plays) AS min_plays FROM songs;
+-----------+
| min_plays |
+-----------+
|        80 |
+-----------+
```

1 row in set (0.00 sec)

3)Count the number of albums released by each record label:

```
mysql> SELECT record_label_id, COUNT(*) AS albums_count FROM album GROUP BY record_label_id;
+-----------------+--------------+
| record_label_id | albums_count |
+-----------------+--------------+
|            1004 |            1 |
|            1005 |            1 |
|            1006 |            1 |
|            1007 |            1 |
|            1008 |            1 |
|            1009 |            1 |
|            1010 |            1 |
|            1011 |            1 |
|            1012 |            1 |
|            1013 |            1 |
+-----------------+--------------+
10 rows in set (0.00 sec)
```

4)Calculate the average number of plays for each genre across all songs:

```
mysql> SELECT genre, AVG(no_of_plays) AS avg_plays FROM songs GROUP BY genre;
+------------+-----------+
| genre      | avg_plays |
+------------+-----------+
| Pop        |  100.0000 |
| Rock       |  150.0000 |
| Hip Hop    |  120.0000 |
| Electronic |   80.0000 |
| Jazz       |  200.0000 |
| Blues      |   90.0000 |
| Country    |  110.0000 |
| Classical  |  130.0000 |
| R&B        |  170.0000 |
| Reggae     |  140.0000 |
+------------+-----------+
10 rows in set (0.00 sec)
```

==========NESTED SUB QUERY==========

1)Find all songs sung by female singers:

```
mysql> SELECT * FROM songs WHERE singer_id IN (SELECT singer_id FROM singer WHERE gender = 'Female');
+---------+--------------+-----------+-----------+-------------+-----------+----------+-------------+
| song_id | release_date | song_name | genre     | no_of_plays | singer_id | album_id | composer_id |
+---------+--------------+-----------+-----------+-------------+-----------+----------+-------------+
|       1 | 2022-01-01   | Song 1    | Pop       |         100 |      1002 |        1 |        2002 |
|       8 | 2022-01-08   | Song 8    | Classical |         130 |      1002 |        8 |        2010 |
|       2 | 2022-01-02   | Song 2    | Rock      |         150 |      1004 |        2 |        2005 |
|       3 | 2022-01-03   | Song 3    | Hip Hop   |         120 |      1006 |        3 |        2001 |
|       7 | 2022-01-07   | Song 7    | Country   |         110 |      1008 |        7 |        2005 |
|       9 | 2022-01-09   | Song 9    | R&B       |         170 |      1010 |        9 |        2009 |
+---------+--------------+-----------+-----------+-------------+-----------+----------+-------------+
6 rows in set (0.00 sec)
```

2)Find all composers who have not composed any songs in the "Rock" genre:

```
mysql> SELECT * FROM composer WHERE composer_id NOT IN (SELECT composer_id FROM songs WHERE genre = 'Rock');
+-------------+-----------------+--------+------------+----------------+----------------+
| composer_id | composer_name   | gender | contact    | composer_scale | composer_genre |
+-------------+-----------------+--------+------------+----------------+----------------+
|        2001 | Michael Johnson | Male   | 1234567890 | Tenor          | Pop            |
|        2002 | Emily Smith     | Female | 9876543210 | Soprano        | Rock           |
|        2003 | Michael Brown   | Male   | 5551234567 | Baritone       | Jazz           |
|        2004 | Sophia Williams | Female | 1112223333 | Mezzo-soprano  | R&B            |
|        2006 | John Smith      | Male   | 9999999999 | Tenor          | Classical      |
|        2007 | Emma Davis      | Female | 8888888888 | Mezzo-soprano  | Folk           |
|        2008 | Andrew Johnson  | Male   | 5555555555 | Tenor          | Country        |
```

```
|        2009 | Olivia Adams    | Female | 6666666666 | Soprano        | Indie          |
|        2010 | William Taylor  | Male   | 7777777777 | Baritone       | Electronic     |
+-------------+-----------------+--------+------------+----------------+----------------+
9 rows in set (0.00 sec)
```

3)Find all songs with more than 100 plays performed by singers with a contact number ending in "8888":

```
mysql> SELECT * FROM songs WHERE no_of_plays > 100 AND singer_id IN (SELECT singer_id FROM singer
        WHERE contact LIKE '%8888');
+---------+--------------+-----------+-------+-------------+-----------+----------+-------------+
| song_id | release_date | song_name | genre | no_of_plays | singer_id | album_id | composer_id |
+---------+--------------+-----------+-------+-------------+-----------+----------+-------------+
|       9 | 2022-01-09   | Song 9    | R&B   |         170 |      1010 |        9 |        2009 |
+---------+--------------+-----------+-------+-------------+-----------+----------+-------------+
1 row in set (0.03 sec)
```

4)Find all singers who have worked on albums but have not composed any songs:

```
mysql> SELECT * FROM singer WHERE EXISTS (SELECT * FROM album WHERE singer.singer_id = album.singer_id)
    -> AND singer_id NOT IN (SELECT singer_id FROM songs);
+-----------+----------------+--------+------------+--------------+--------------+
| singer_id | singer_name    | gender | contact    | singer_scale | singer_genre |
+-----------+----------------+--------+------------+--------------+--------------+
|      1009 | Matthew Wilson | Male   | 6662228888 | Baritone     | Blues        |
+-----------+----------------+--------+------------+--------------+--------------+
1 row in set (0.00 sec)
```

==========JOIN OPERATOR==========

1)Retrieve composer names along with their corresponding genres from the songs table:

```
mysql> SELECT c.composer_name, sg.genre
    -> FROM composer c
    -> INNER JOIN songs sg ON c.composer_id = sg.composer_id;
+-----------------+------------+
| composer_name   | genre      |
+-----------------+------------+
| Emily Smith     | Pop        |
| David Lee       | Rock       |
| Michael Johnson | Hip Hop    |
| Michael Brown   | Electronic |
| Michael Johnson | Jazz       |
| John Smith      | Blues      |
| David Lee       | Country    |
| William Taylor  | Classical  |
| Olivia Adams    | R&B        |
| Andrew Johnson  | Reggae     |
+-----------------+------------+
10 rows in set (0.00 sec)
```

2)Retrieve singer names along with their corresponding genres from the songs table:

```
mysql> SELECT s.singer_name, sg.genre
    -> FROM singer s
    -> INNER JOIN songs sg ON s.singer_id = sg.singer_id;
+-----------------+------------+
| singer_name     | genre      |
+-----------------+------------+
| Emily Johnson   | Pop        |
| Sophia Williams | Rock       |
| Jennifer White  | Hip Hop    |
| Michael Brown   | Electronic |
| John Smith      | Jazz       |
| Robert Johnson  | Blues      |
| Emma Davis      | Country    |
| Emily Johnson   | Classical  |
| Olivia Martinez | R&B        |
| David Lee       | Reggae     |
+-----------------+------------+
10 rows in set (0.00 sec)
```

3)Retrieve singer names along with the total number of songs they have performed:

```
mysql> SELECT s.singer_name, COUNT(sg.song_id) AS total_songs
    -> FROM singer s
    -> LEFT JOIN songs sg ON s.singer_id = sg.singer_id
    -> GROUP BY s.singer_name;
+-----------------+-------------+
| singer_name     | total_songs |
+-----------------+-------------+
| John Smith      |           1 |
| Emily Johnson   |           2 |
| Michael Brown   |           1 |
| Sophia Williams |           1 |
| David Lee       |           1 |
| Jennifer White  |           1 |
| Robert Johnson  |           1 |
| Emma Davis      |           1 |
| Matthew Wilson  |           0 |
| Olivia Martinez |           1 |
+-----------------+-------------+
10 rows in set (0.00 sec)
```

4)Retrieve composer names along with the average number of plays for the songs they have composed:

```
mysql> SELECT c.composer_name, AVG(sg.no_of_plays) AS avg_plays
    -> FROM composer c
    -> INNER JOIN songs sg ON c.composer_id = sg.composer_id
    -> GROUP BY c.composer_name;
+-----------------+-----------+
| composer_name   | avg_plays |
+-----------------+-----------+
| Emily Smith     |  100.0000 |
| David Lee       |  130.0000 |
| Michael Johnson |  160.0000 |
| Michael Brown   |   80.0000 |
| John Smith      |   90.0000 |
| William Taylor  |  130.0000 |
| Olivia Adams    |  170.0000 |
| Andrew Johnson  |  140.0000 |
+-----------------+-----------+
8 rows in set (0.00 sec)
```

==========MODIFICATION IN RECORDS (Delete,Update & Insert)==========

1)Insert a new singer into the singer table:

```
mysql> INSERT INTO singer (singer_id, singer_name, gender, contact, singer_scale, singer_genre)
    -> VALUES (1011, 'Rachel Green', 'Female', '1234567890', 'Soprano', 'Pop');
Query OK, 1 row affected (0.05 sec)
```

2)Update the contact number of a specific singer:

```
mysql> UPDATE singer SET contact = '5555555555' WHERE singer_id = 1001;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

3)Delete an album from the album table based on album_id:

```
mysql> SET FOREIGN_KEY_CHECKS = 0;
mysql> DELETE FROM album WHERE album_id = 3;
mysql> SET FOREIGN_KEY_CHECKS = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> select* from album;
+----------+--------------+------------+------------+-----------------+---------+-----------+-------------+
| album_id | release_date | album_name | no_of_play | record_label_id | song_id | singer_id | composer_id |
+----------+--------------+------------+------------+-----------------+---------+-----------+-------------+
|        1 | 2023-01-01   | Album 1    |        100 |            1004 |       1 |      1001 |        2001 |
```

```
|        2 | 2023-02-15   | Album 2    |        150 |            1005 |       2 |      1002 |        2002 |
|        4 | 2023-04-10   | Album 4    |        180 |            1006 |       4 |      1004 |        2004 |
|        5 | 2023-05-25   | Album 5    |        220 |            1007 |       5 |      1005 |        2005 |
|        6 | 2023-06-30   | Album 6    |        190 |            1008 |       6 |      1006 |        2006 |
|        7 | 2023-07-15   | Album 7    |        160 |            1009 |       7 |      1007 |        2007 |
|        8 | 2023-08-20   | Album 8    |        130 |            1010 |       8 |      1008 |        2008 |
|        9 | 2023-09-05   | Album 9    |        240 |            1011 |       9 |      1009 |        2009 |
|       10 | 2023-10-10   | Album 10   |        170 |            1012 |      10 |      1010 |        2010 |
+----------+--------------+------------+------------+-----------------+---------+-----------+-------------+
```

4)Delete a specific song from the songs table:

```
mysql> SET FOREIGN_KEY_CHECKS = 0;
mysql> DELETE FROM songs WHERE song_id = 1;
mysql> SET FOREIGN_KEY_CHECKS = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> select*from songs;
+---------+--------------+-----------+------------+-------------+-----------+----------+-------------+
| song_id | release_date | song_name | genre      | no_of_plays | singer_id | album_id | composer_id |
+---------+--------------+-----------+------------+-------------+-----------+----------+-------------+
|       2 | 2022-01-02   | Song 2    | Rock       |         150 |      1004 |        2 |        2005 |
|       3 | 2022-01-03   | Song 3    | Hip Hop    |         120 |      1006 |        3 |        2001 |
|       4 | 2022-01-04   | Song 4    | Electronic |          80 |      1003 |        4 |        2003 |
|       5 | 2022-01-05   | Song 5    | Jazz       |         200 |      1001 |        5 |        2001 |
|       6 | 2022-01-06   | Song 6    | Blues      |          90 |      1007 |        6 |        2006 |
|       7 | 2022-01-07   | Song 7    | Country    |         110 |      1008 |        7 |        2005 |
|       8 | 2022-01-08   | Song 8    | Classical  |         130 |      1002 |        8 |        2010 |
|       9 | 2022-01-09   | Song 9    | R&B        |         170 |      1010 |        9 |        2009 |
|      10 | 2022-01-10   | Song 10   | Reggae     |         140 |      1005 |       10 |        2008 |
+---------+--------------+-----------+------------+-------------+-----------+----------+-------------+
```