

**MUSIC STREAMING PLATFORM PROJECT**

Name: Sujit Navadkar

Project: SQL Project

1. **Introduction**

Our project is database system of music streaming platform, so artists can share their music to their users and user can explore new music and artists as per there requirements. In this music database project, there are multiple tables are present in database. The database allows the storage and management of data related to artists, albums, songs, users and playlists on the platform.

There are 6 tables in database:

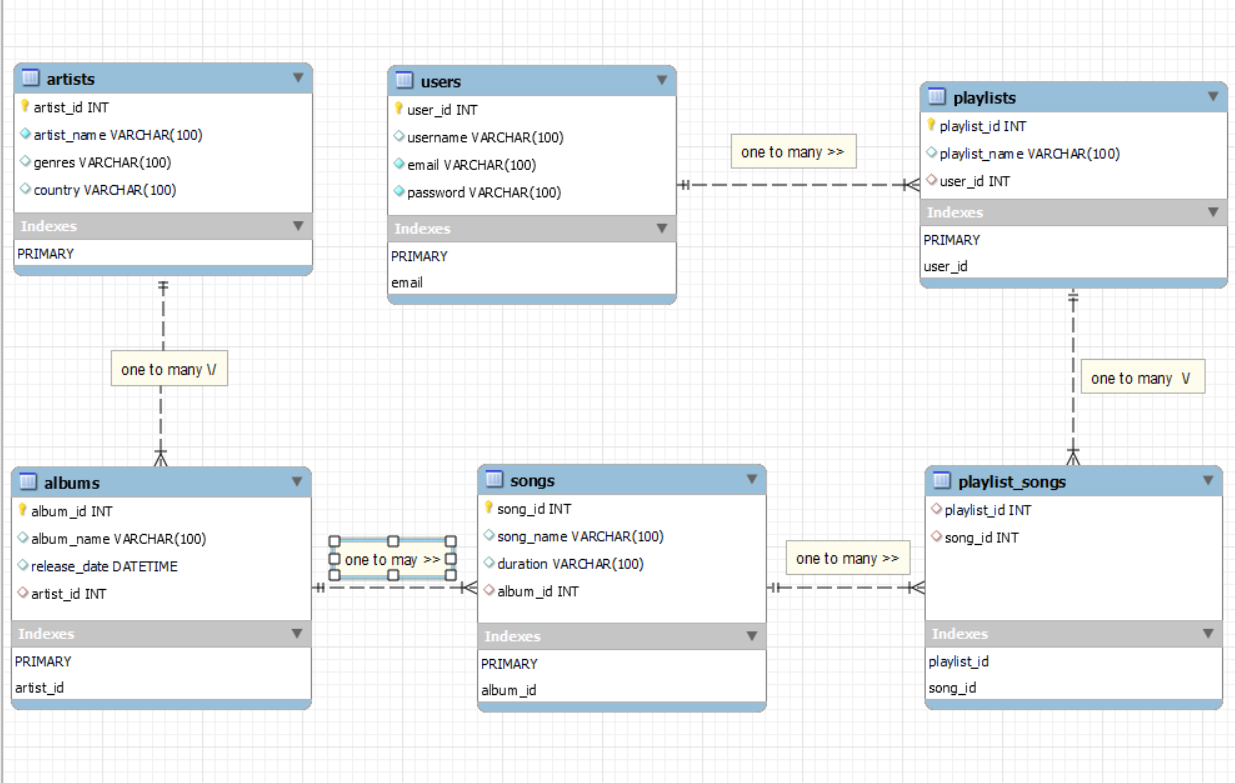
* Artists - This table stores data of artists
* Albums - This table stores data of albums released by artists
* Songs - This table stores data of songs from albums
* Users - This table stores data of users
* Playlists - This table stores playlists created by users
* Playlist\_Tracks – This tables link between playlist and songs

Tables are related using foreign key and cascading deletes are used to automatically remove related records.

There are some relations between tables:

* One artist can have multiple albums (One to Many)
* One album can have multiple songs (One to Many)
* One user can have multiple playlists (One to Many)
* One playlist can have multiple songs (One to Many)

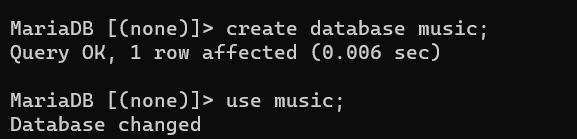
1. **ER Diagram**



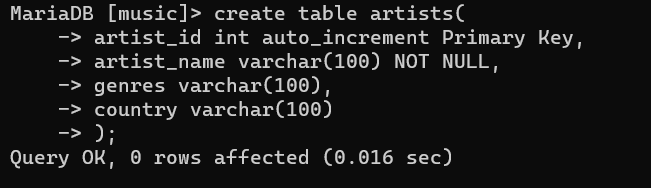
1. **Database Design**

* Database: Music
* Tables:

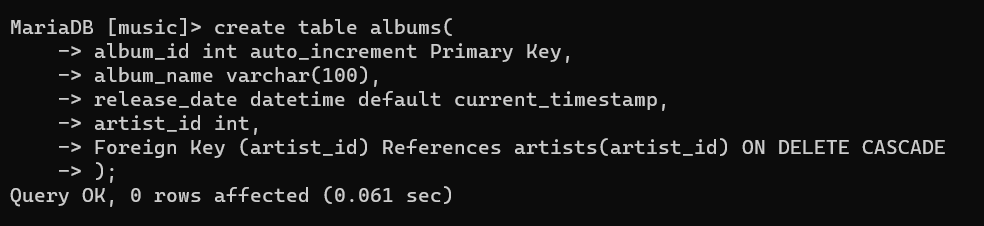
1. Artists
2. Albums
3. Songs
4. Users
5. Playlists
6. Playlist\_songs
7. **Create Database**



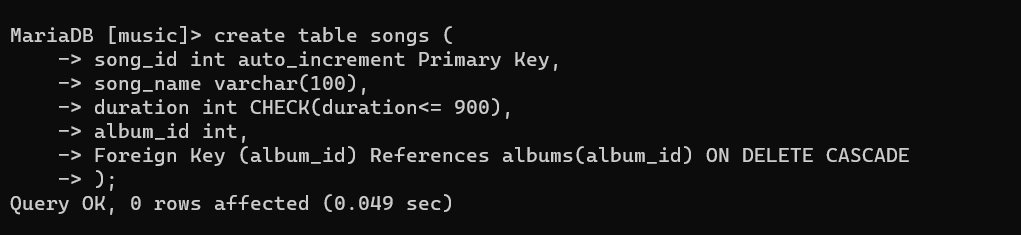
1. **Create Tables**
2. Artists



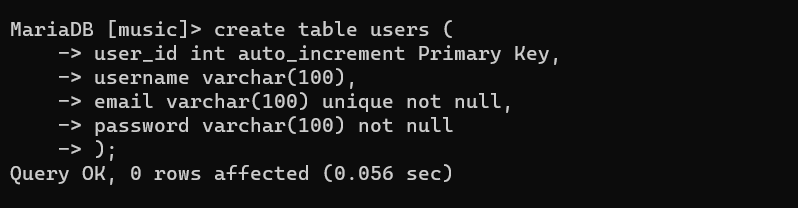
1. Albums



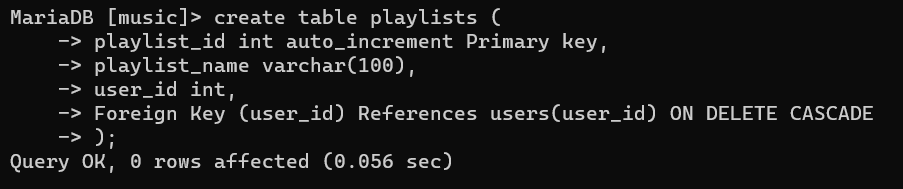
1. Songs



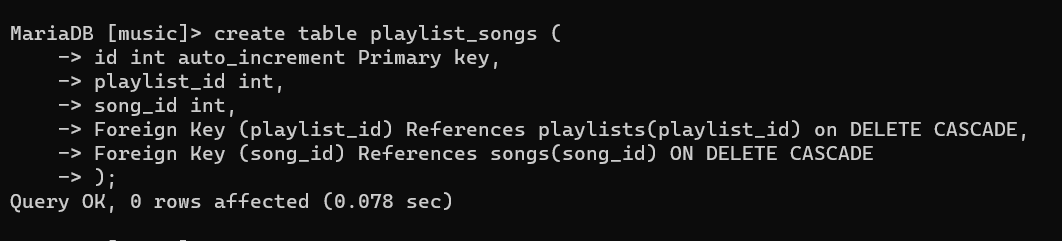
1. Users



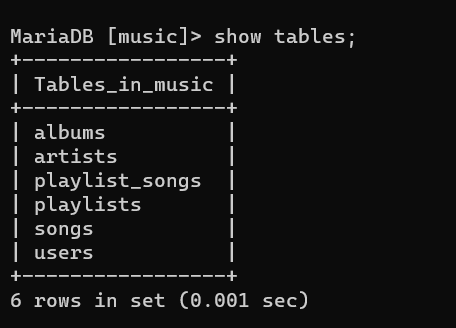
1. Playlists



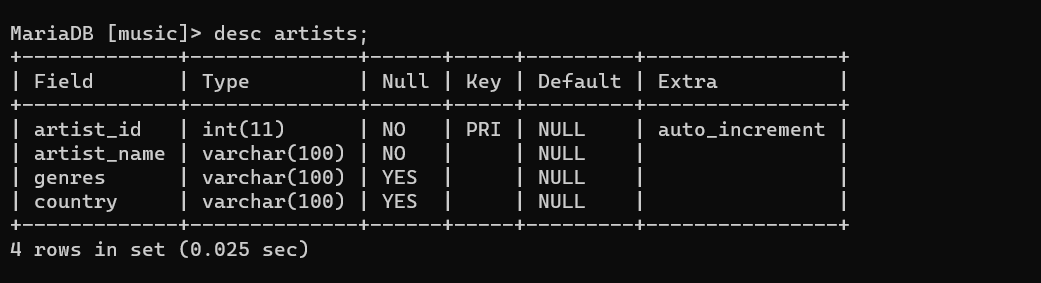
1. Playlist\_songs



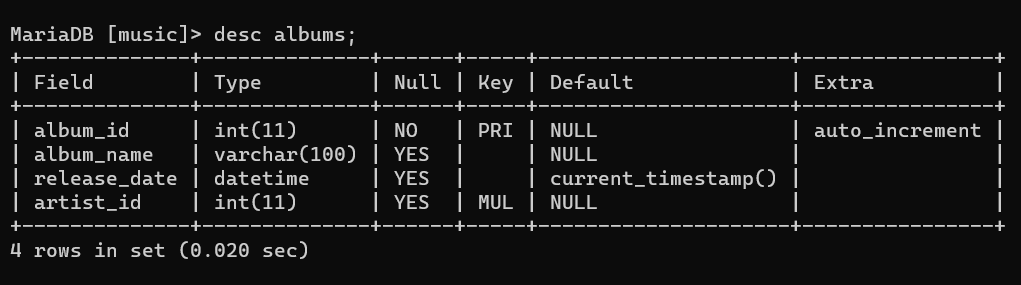
1. **Tables in database**



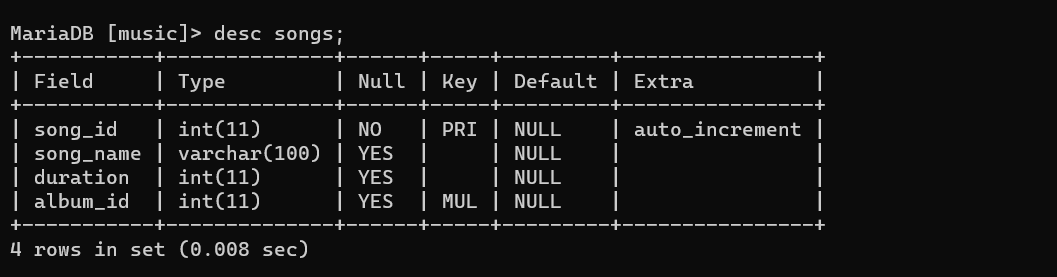
1. **Data Definition language (DDL)**
2. **Creating Tables**
3. Artists



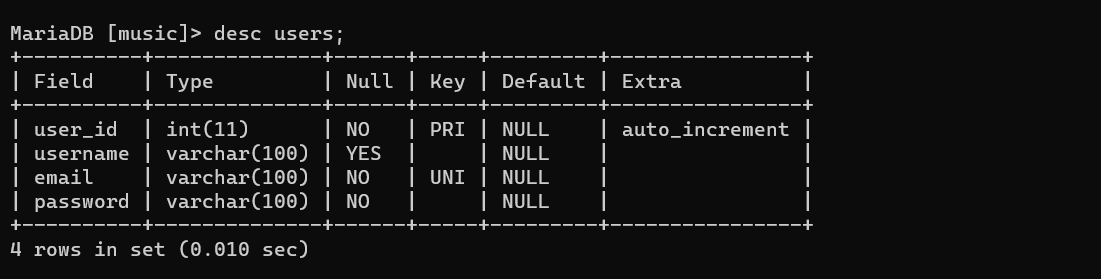
1. Albums



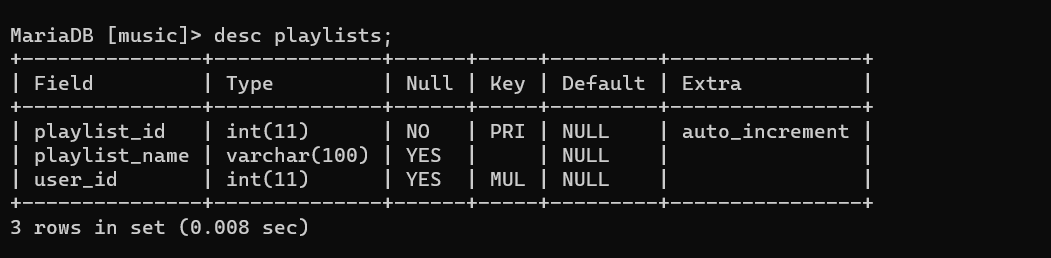
1. Songs



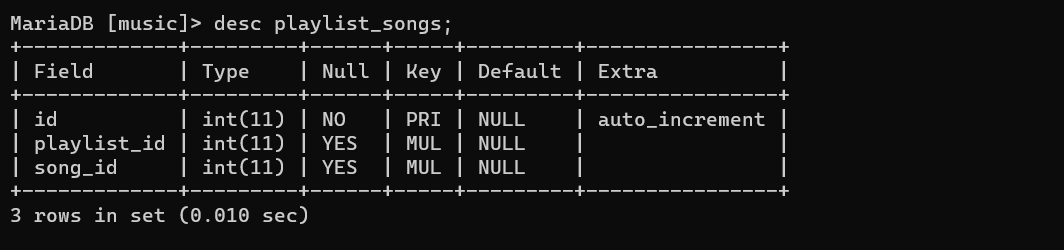
1. Users



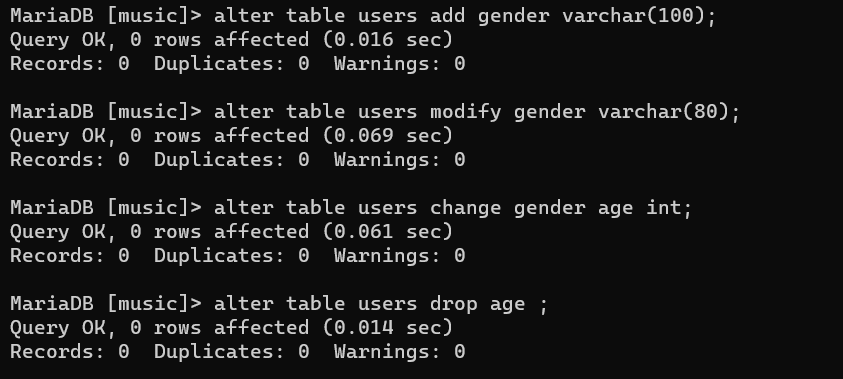
1. Playlists



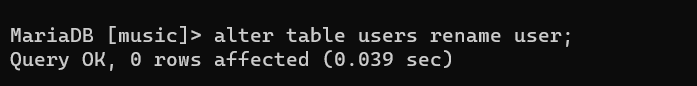
1. Playlist\_Songs



1. **Alter Table**
2. Alter table add column
3. Alter table modify column
4. Alter table rename column
5. Alter table drop column



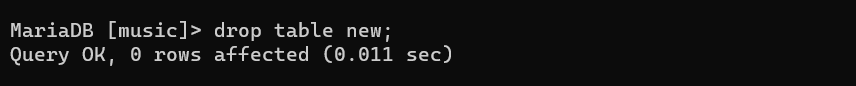
1. **Rename Table**



1. **Truncate Table**

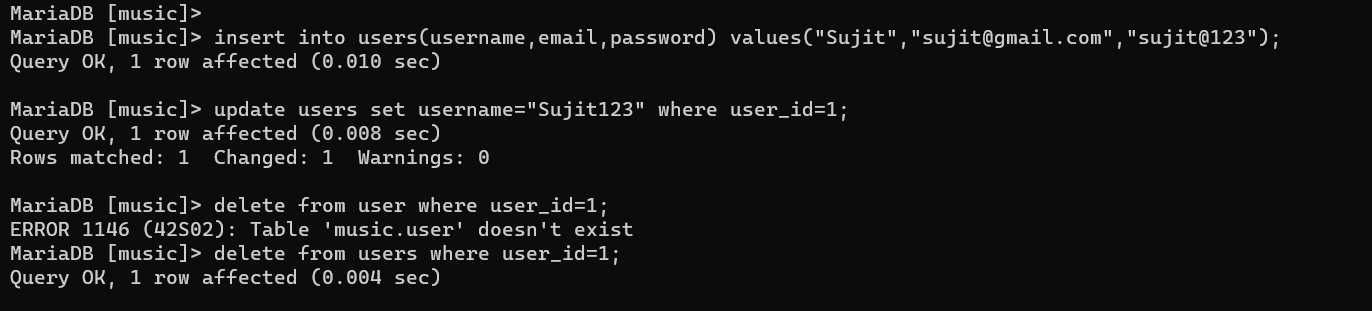


1. **Drop Table**



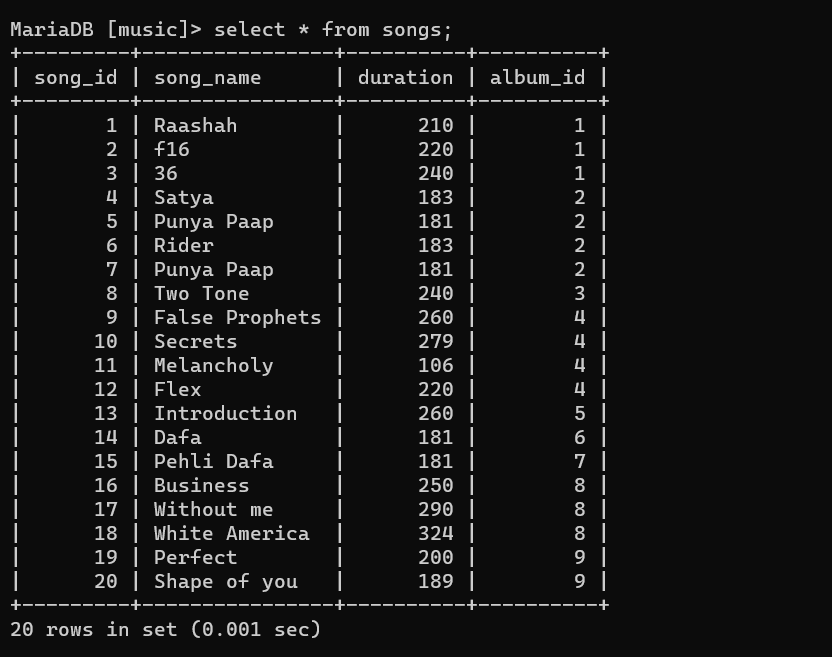
1. **Data Manipulation language (DML)**

* Insert, Update, Delete

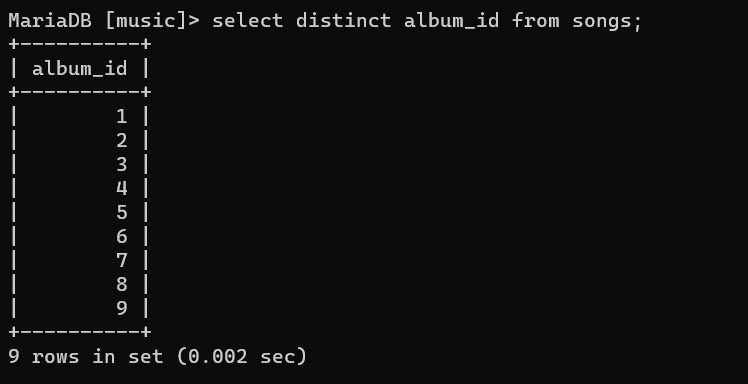


1. **Data query language (DQL)**
   1. Select Query

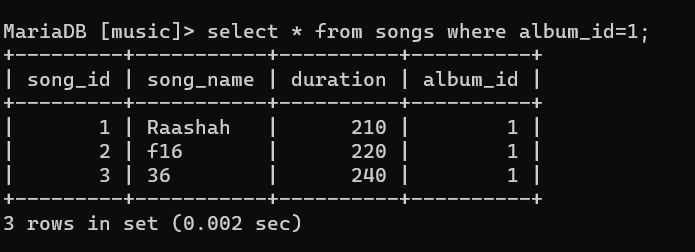




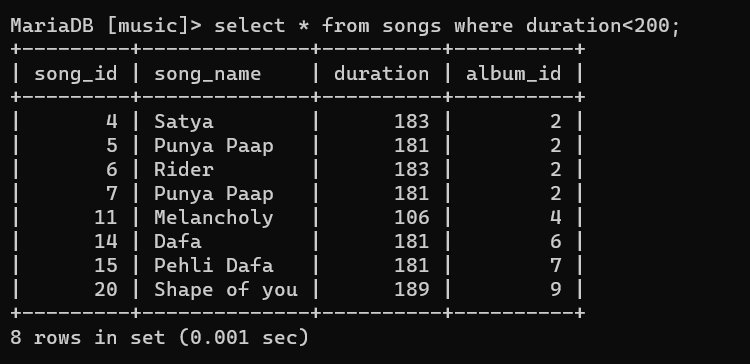
* 1. Distinct



* 1. Where clause
* = operator

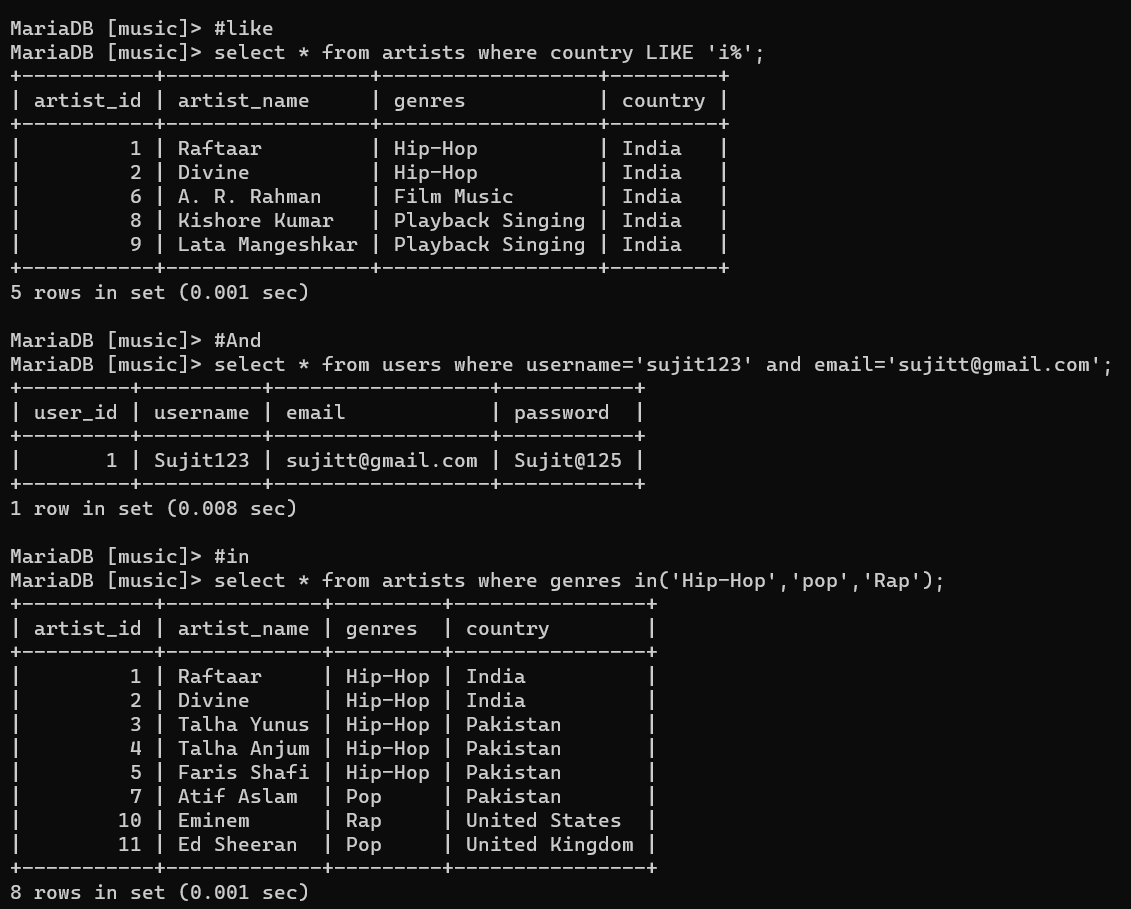


* < operator

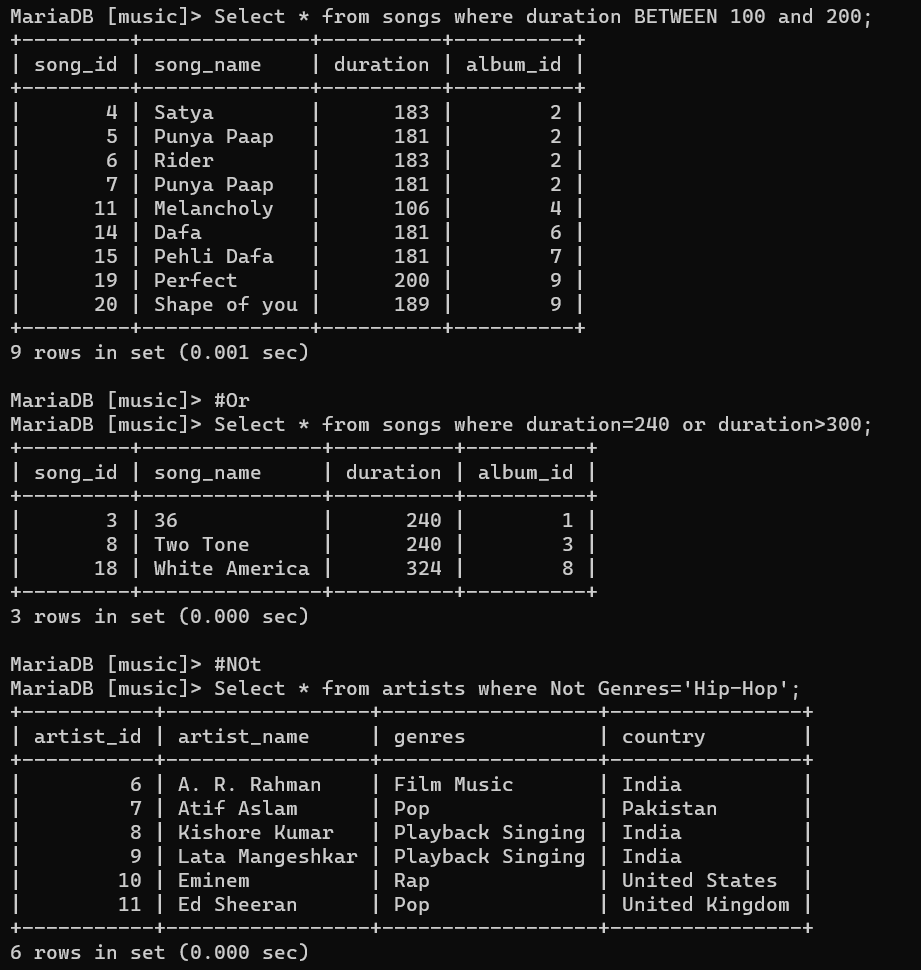


* 1. Logical Operator

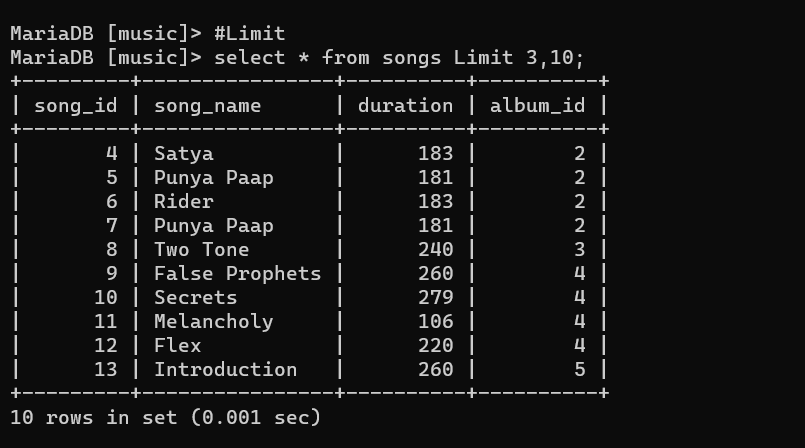
Like, And, In



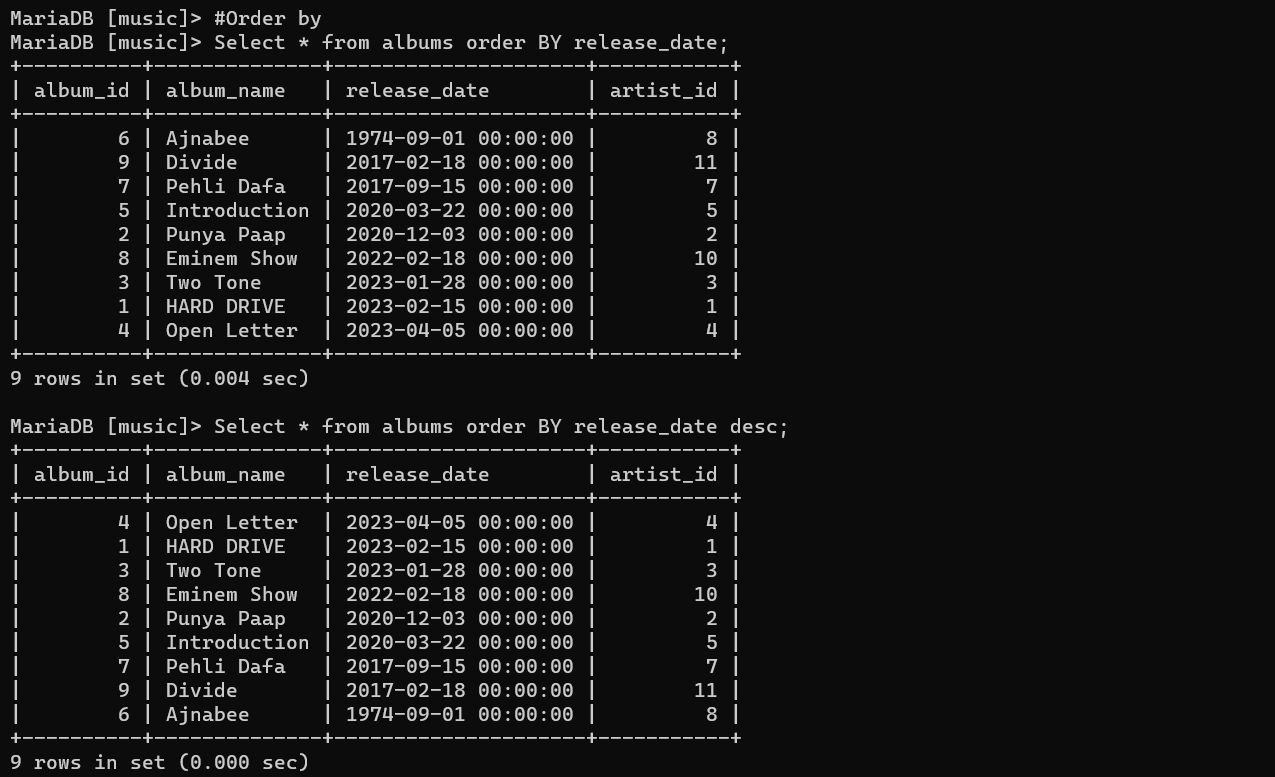
Between, Or, Not



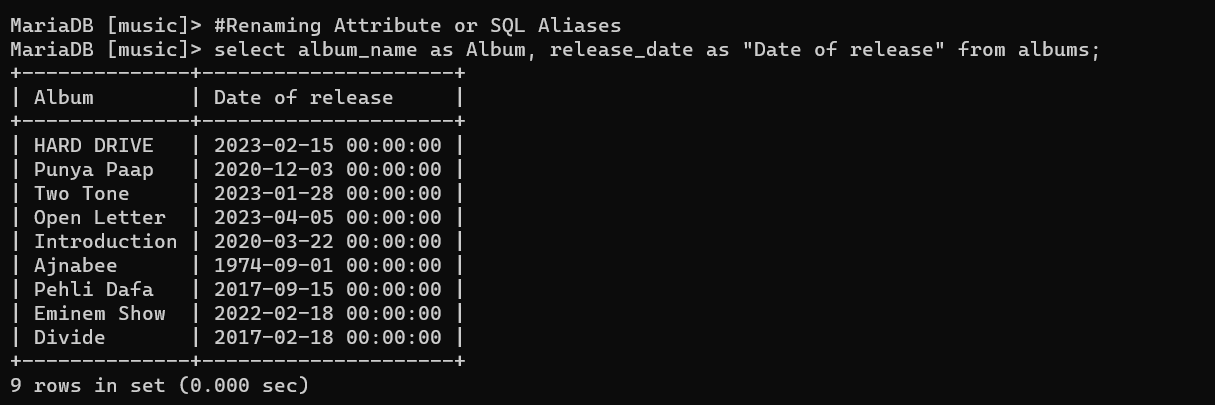
* 1. Limit



* 1. Order By

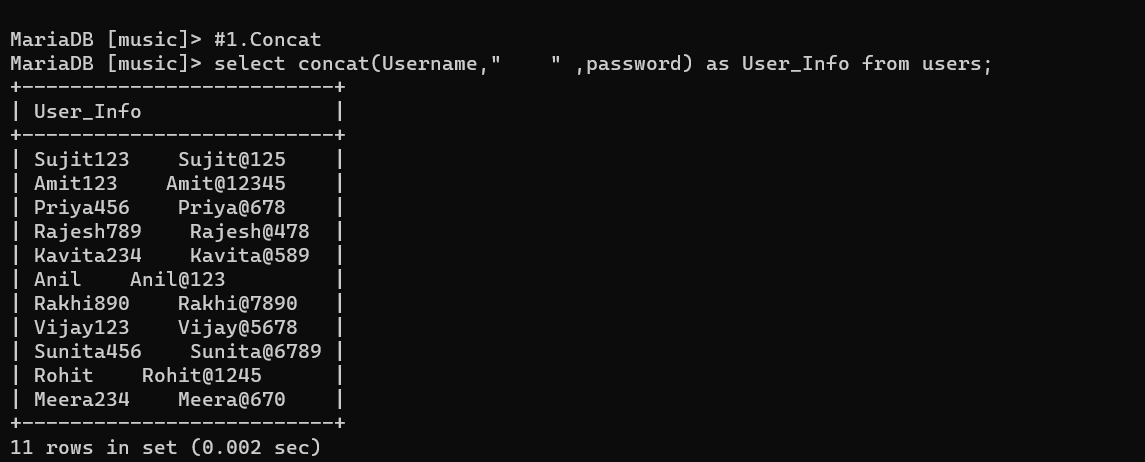


* 1. Renaming Attribute or SQL Aliases

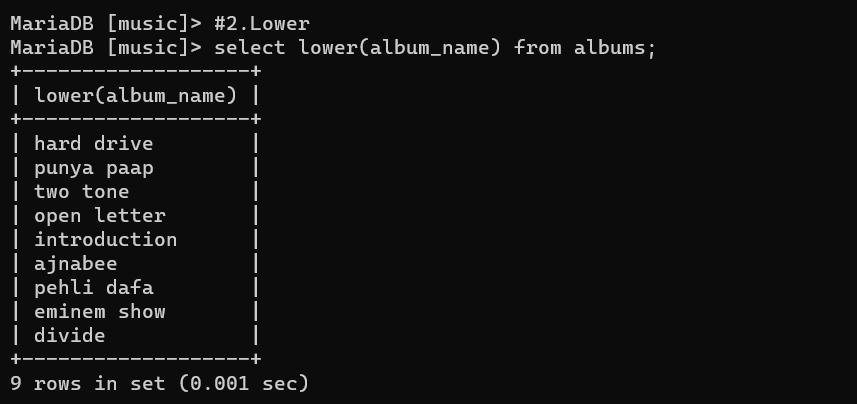


* 1. String()

1.Concat



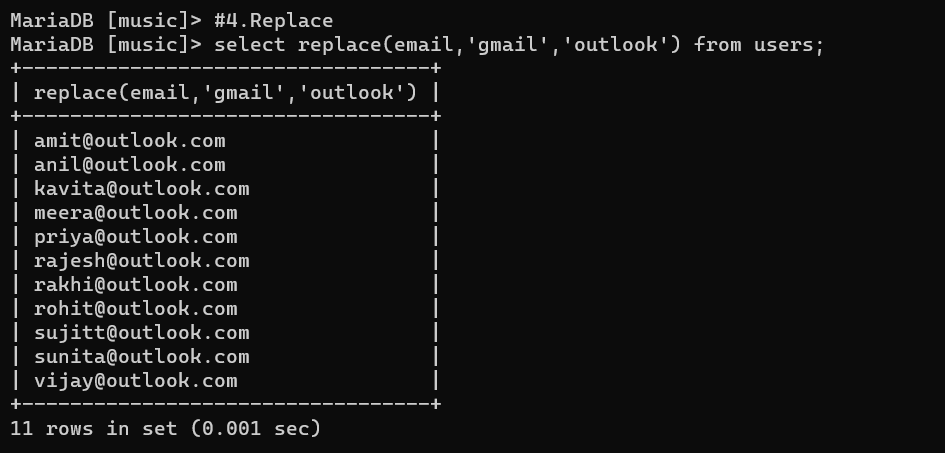
2.Lower

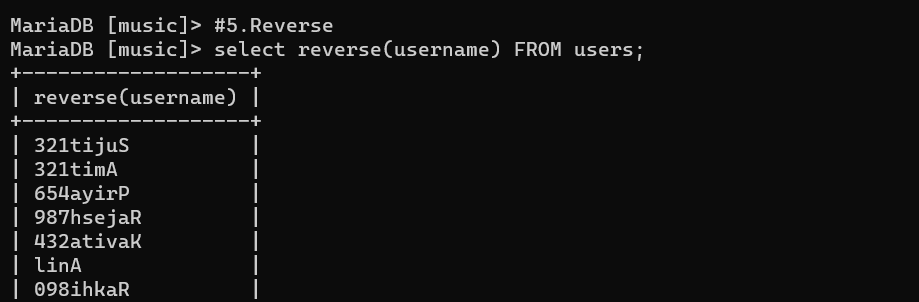


3.Upper

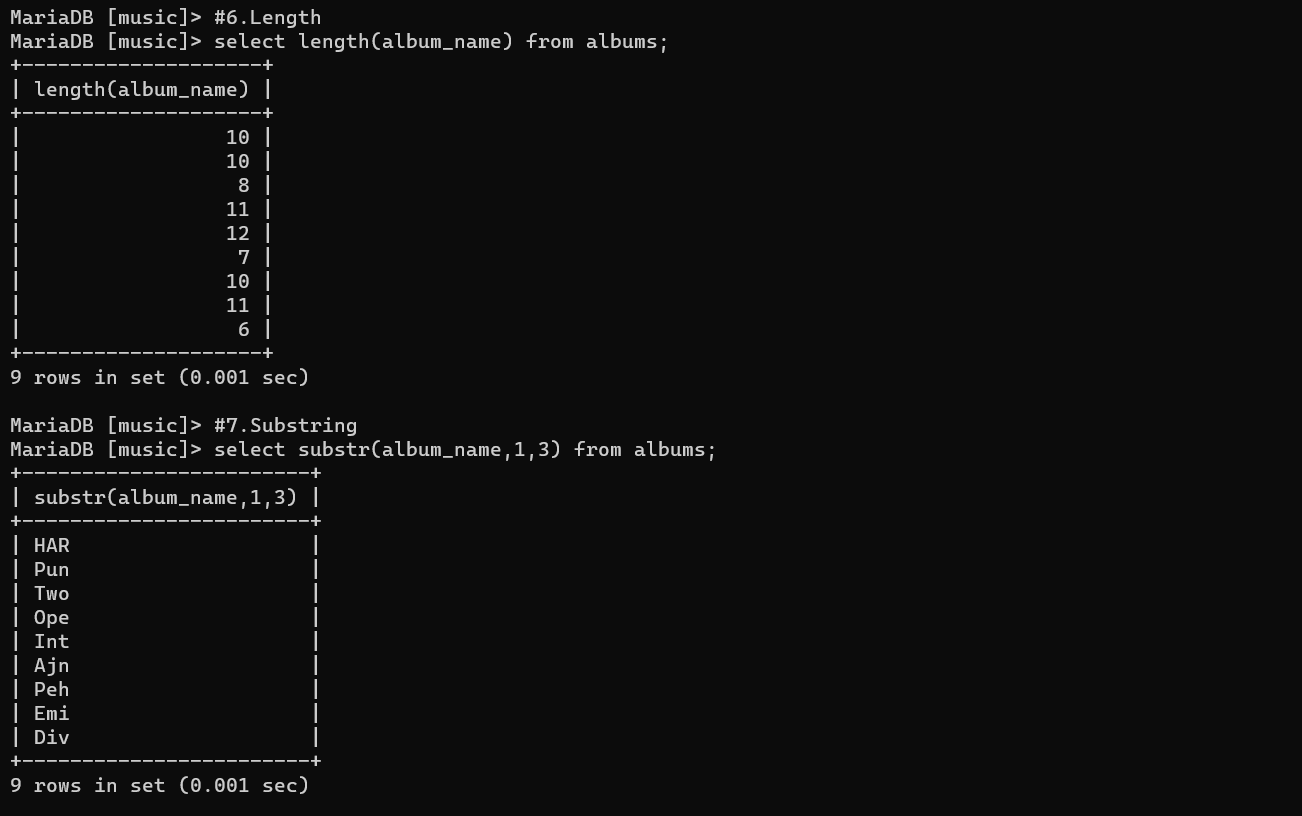


4.Replace



5.Reverse

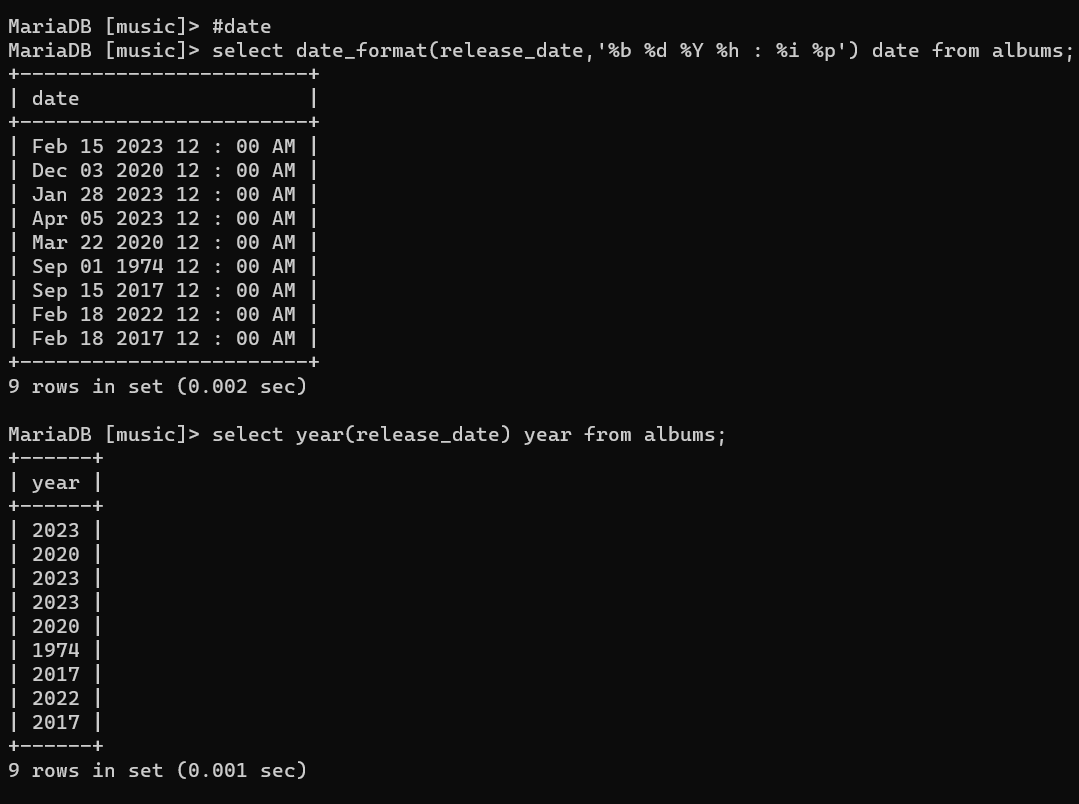
6.Length 7.Substring

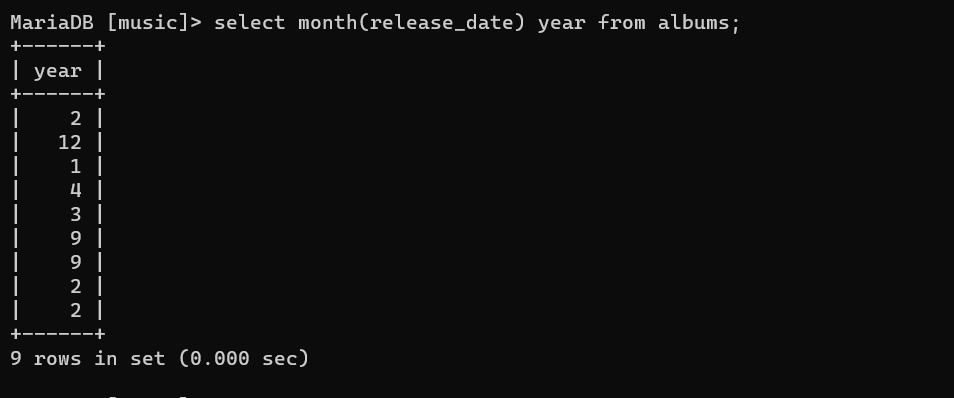


8.Trim



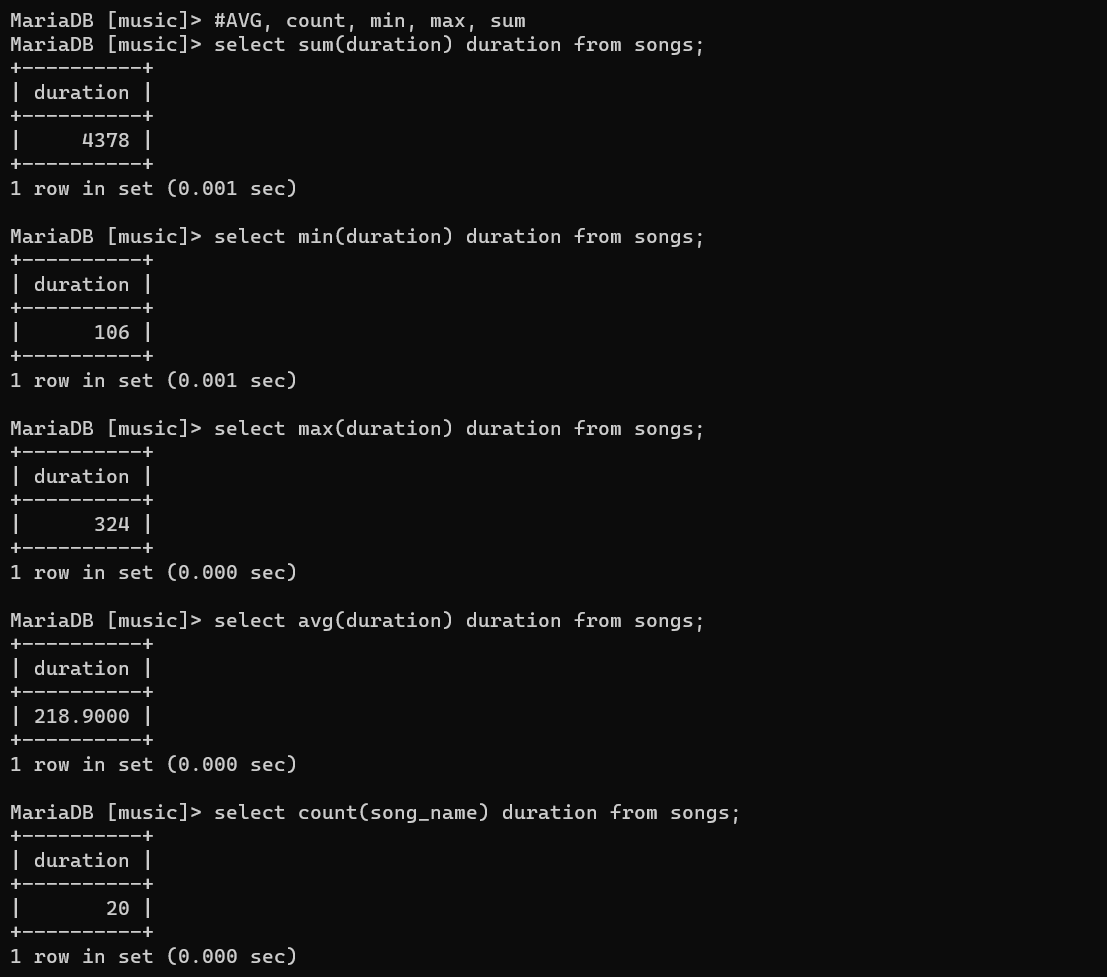
* 1. Date() : Date Format, Month(date), Year(date)



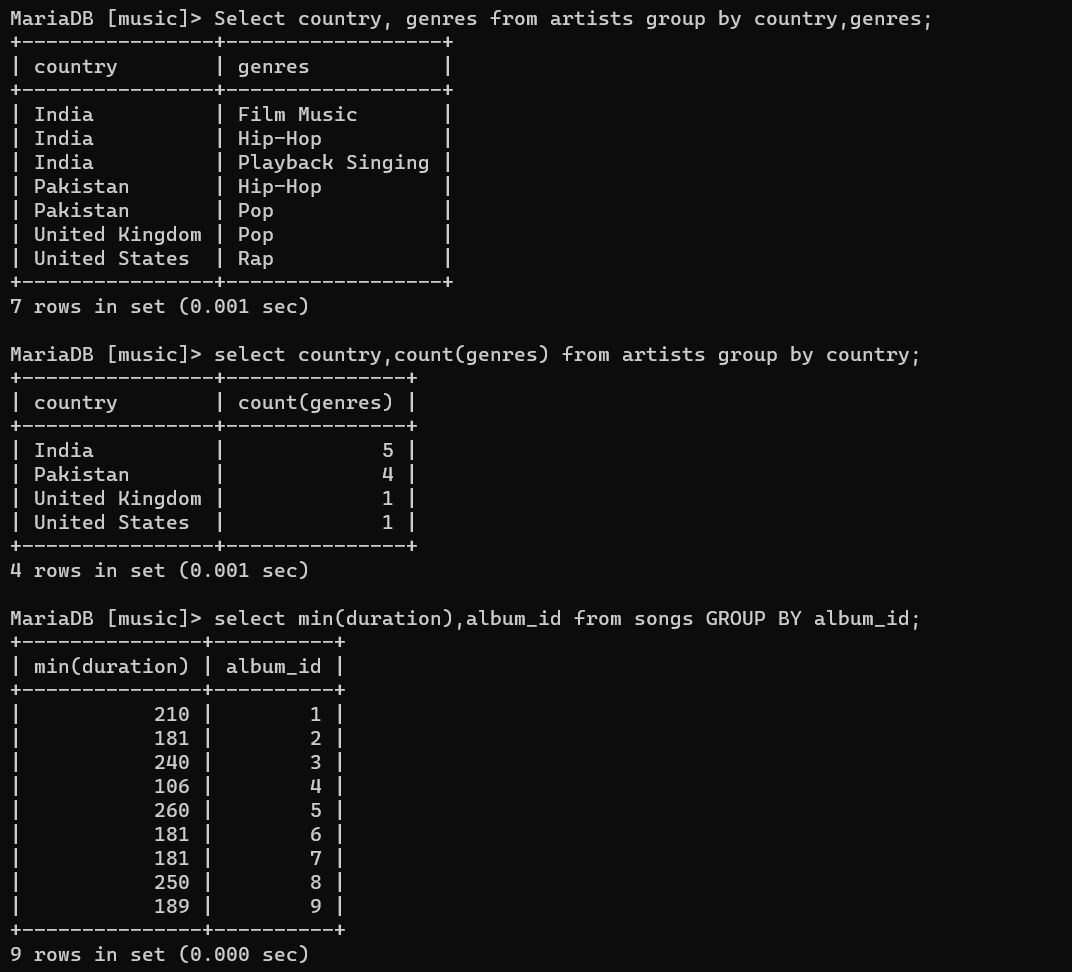


* 1. Aggregate Functions

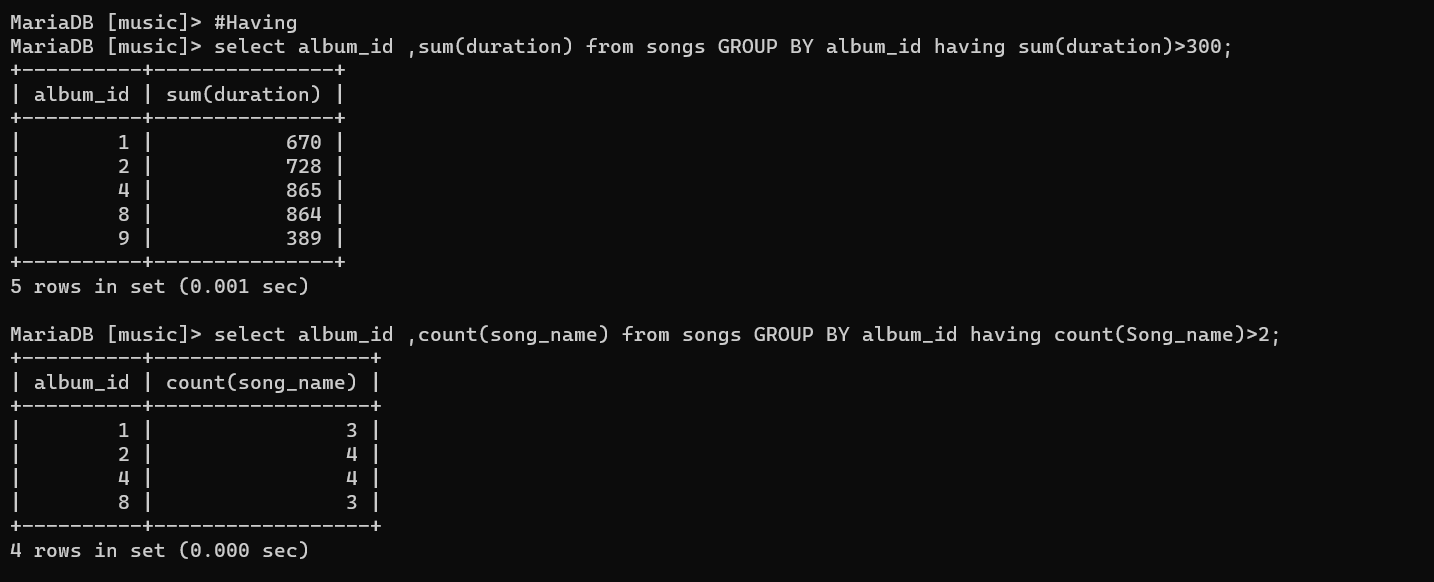
AVG, count, min, max, sum



* 1. Group by



* 1. Having

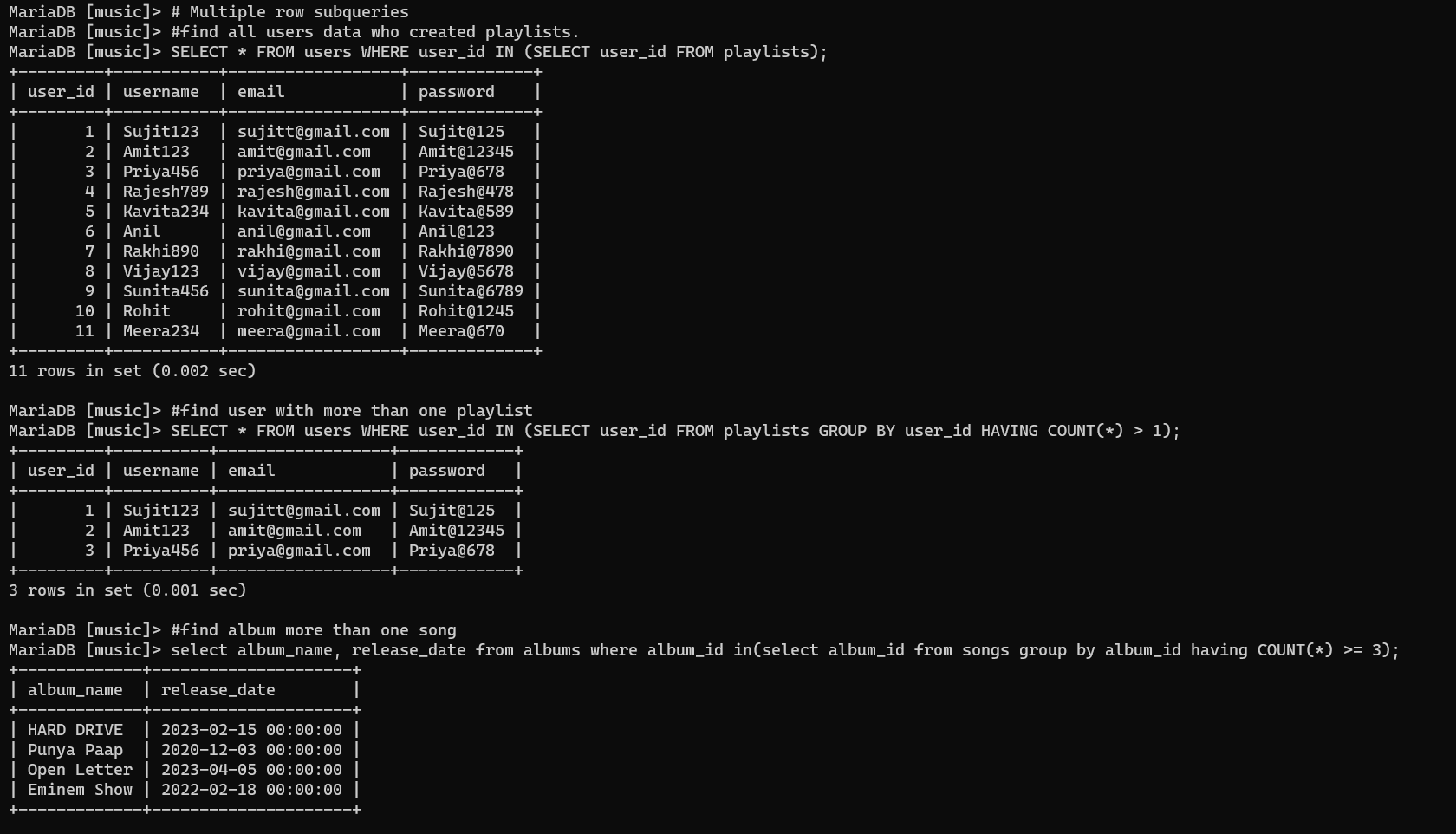


1. **Sub-Queries**

Single row subqueries



Multiple row subqueries

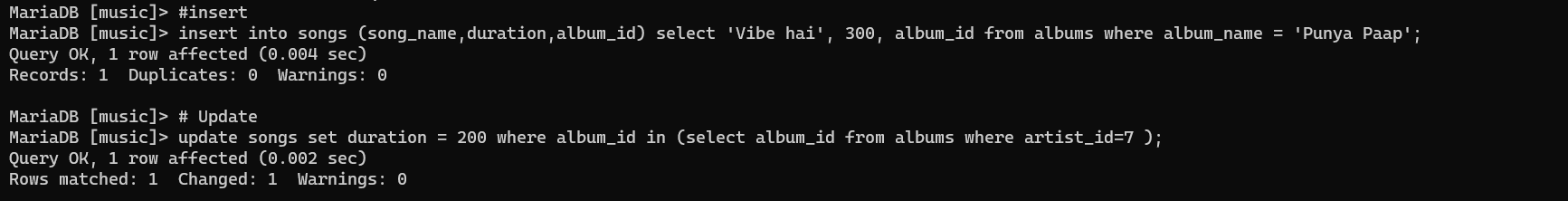


Multiple column subqueries



1. **DML with subqueries**

Insert Update

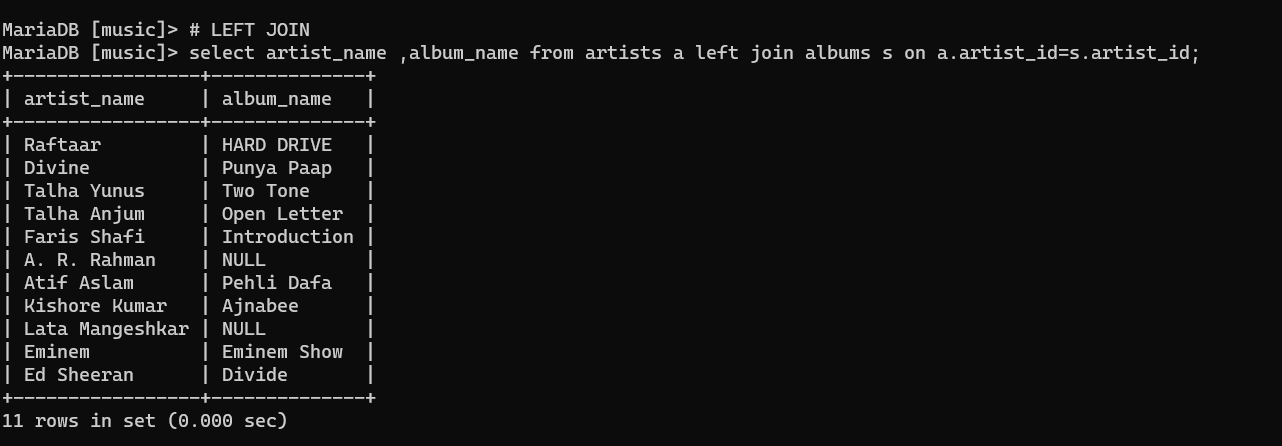


1. **Joins**

INNER JOIN



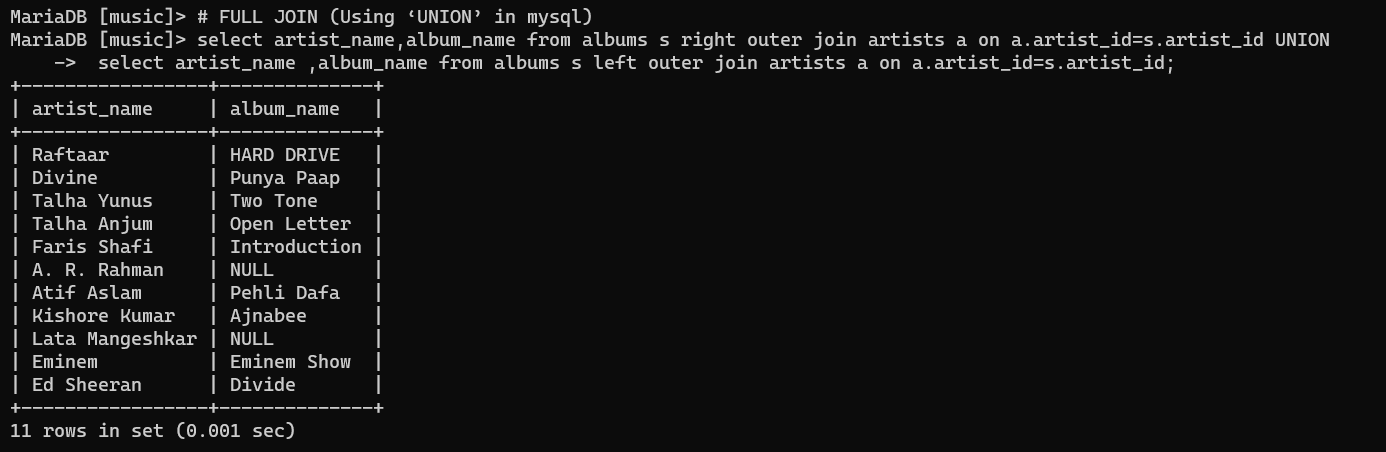
LEFT JOIN

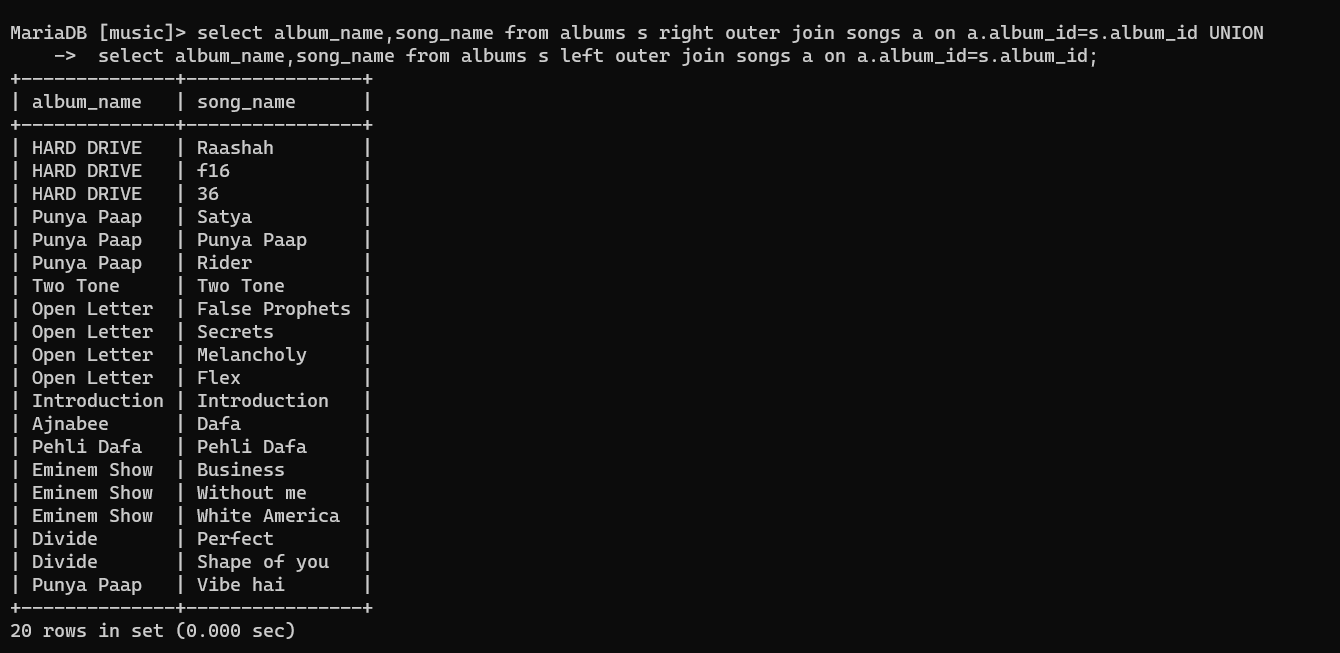


RIGHT JOIN

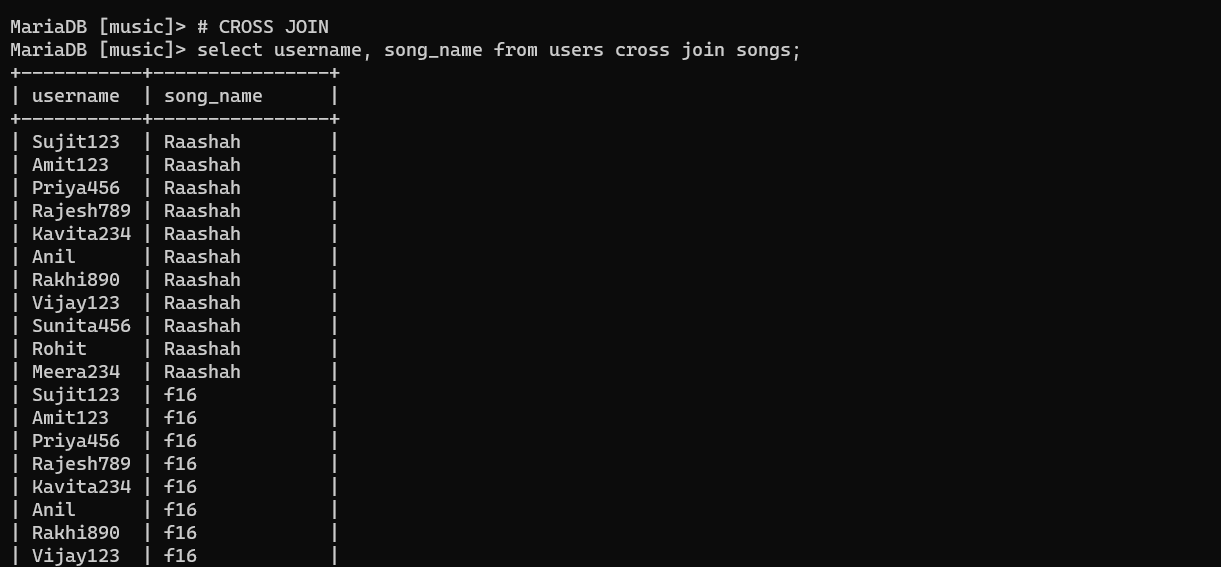


FULL JOIN (Using ‘UNION’ in mysql)

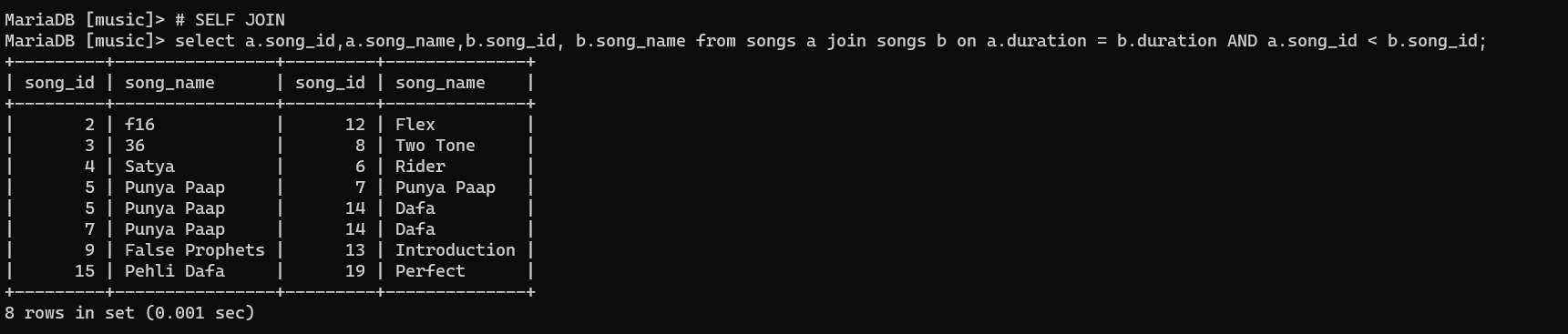




CROSS JOIN

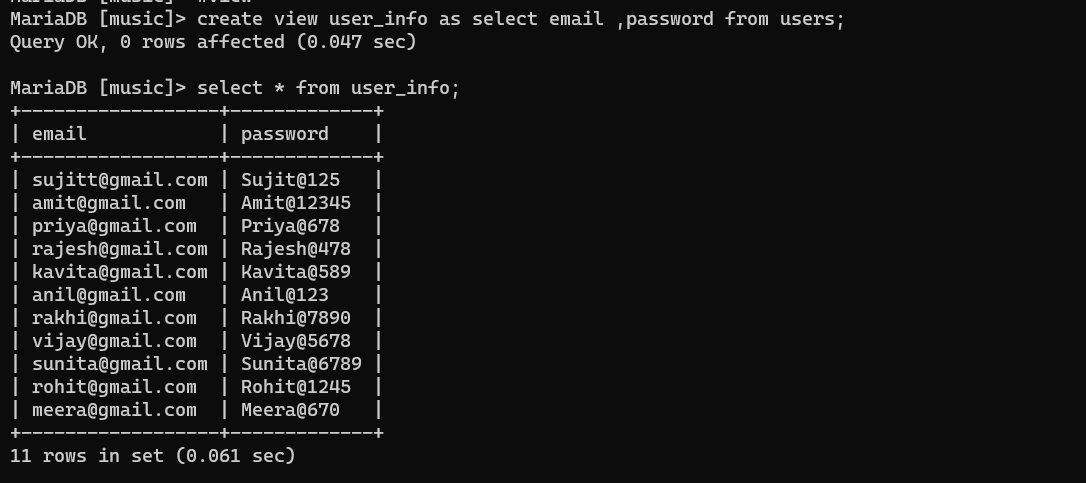


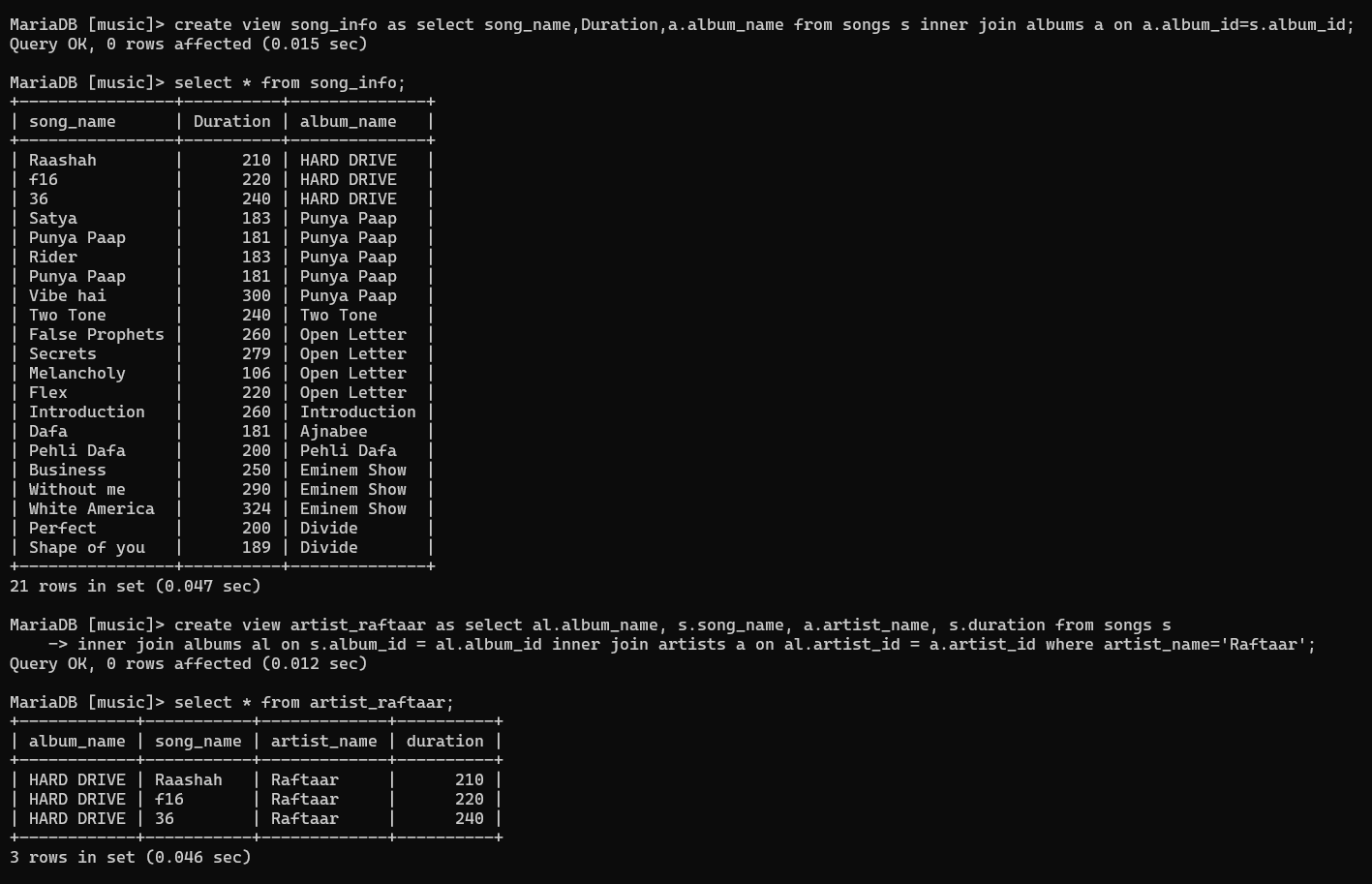
SELF JOIN



1. **Views**

Create view

****

****

Update ,Delete ,Drop

