DR AMBEDKAR INSTITUTE OF TECHNOLOGY

ISE - DBMS ACTIVITY REPORT

TOPIC: MUSIC DATA BASE

GROUP MEMBERES

ANVITH B.K	1DA19IS006
DARSHITH.S	1DA19IS010
HITESH.R	1DA19IS015
SUJAY.S	1DA19IS044
SURAJ.S	1DA19IS048

MUSIC DATABASE SYSTEM INTODUCTION

We don’t even need words to understand what music is saying. Every year I see a lot of different people from a lot of different countries attending festivals. When the music is being played, it seems everyone understands. Music database system keeps data of music that was added into database on basis of categories, language, artist, title, label, year of song which can later be searches by user on basis of any of the above info and it Automatically creates online library/collection of listened to music and generates recommendations.

MAIN FEATURES OF MUSIC DATABASE SYSTEM

- Creating playlists as an artist will show fans what songs and artists you enjoy listening to and be something fans can follow and interact with you through.
- Update your playlists often so you always stay active on follower’s feeds.
- Following and listening to your Discover Weekly playlists can help you discover new music to share it with your fans. You are an artists after all, you love discovering new music.
- Free full-length music download

Music classification in music data base system

One of the classical Music data bases is genre classification, which is categorizing music items into one of pre-defined genres such as classical, jazz, rock, etc. Mood classification, artist classification, and music tagging are also popular topics.

Recommender systems

Several recommender systems for music already exist, but surprisingly few are based upon MIR techniques, instead making use of similarity between users or laborious data compilation. Pandora, for example, uses experts to tag the music with particular qualities such as "female singer" or "strong bassline". Many other systems find users whose listening history is similar and suggests unheard music to the users from their respective collections. MIR techniques for similarity in music are now beginning to form part of such systems.

Music source separation and instrument recognition

Music source separation is about separating original signals from a mixture audio signal. Instrument recognition is about identifying the instruments involved in music. Various Music database systems have been developed that can separate music into its component tracks without access to the master copy. In this way e.g karaoke tracks can be created from normal music tracks, though the process is not yet perfect owing to vocals occupying some of the same frequency space as the other instruments.

Automatic music transcription

Automatic music transcription is the process of converting an audio recording into symbolic notation, such as a score or a MIDI file. This process involves several audio analysis tasks, which may include multi-pitch detection, onset detection, duration estimation, instrument identification, and the extraction of harmonic, rhythmic or melodic information. This task becomes more difficult with greater numbers of instruments and a greater polyphony level.

.

Entity Used

The database has 6 entities.

Namely:

- Record Label
- Song
- Albums
- Artist
- Musician
- Buyers

1. Record Label

The attributes used are Label Name, Song Name, Song ID, Artist ID, Album Name.

Primary Key used is Label Name.
Foreign Keys used are Song ID, Artist ID.

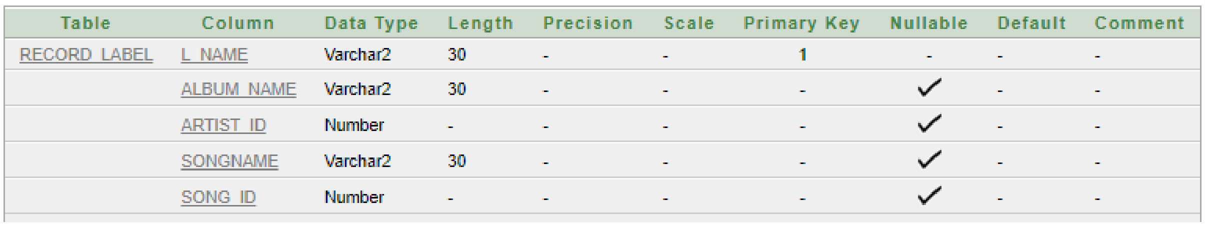
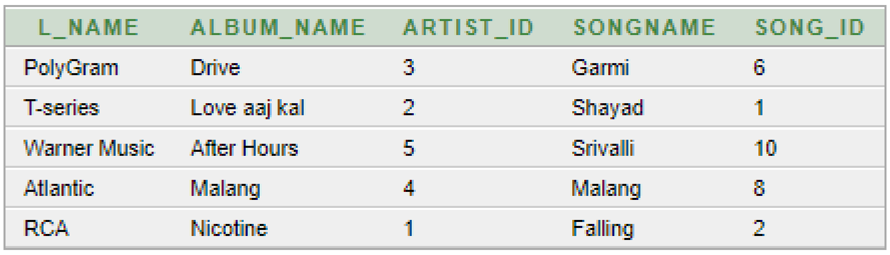


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RECORD_LABEL	L_NAME	Varchar2	30	-	-	1	-	-	-
	ALBUM_NAME	Varchar2	30	-	-	-	✓	-	-
	ARTIST_ID	Number	-	-	-	-	✓	-	-
	SONGNAME	Varchar2	30	-	-	-	✓	-	-
	SONG_ID	Number	-	-	-	-	✓	-	-



L_NAME	ALBUM_NAME	ARTIST_ID	SONGNAME	SONG_ID
PolyGram	Drive	3	Garmi	6
T-series	Love aaj kal	2	Shayad	1
Warner Music	After Hours	5	Srivalli	10
Atlantic	Malang	4	Malang	8
RCA	Nicotine	1	Falling	2

2. Song

The attributes used are Song ID, Song Name, Album Name, Artist Name, Artist ID, Musician ID.

Primary Key used is Song ID.
Foreign Keys used are Artist ID, Musician ID.

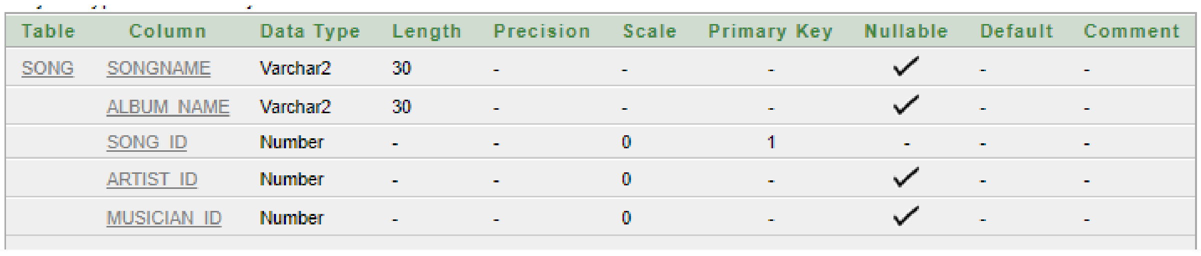
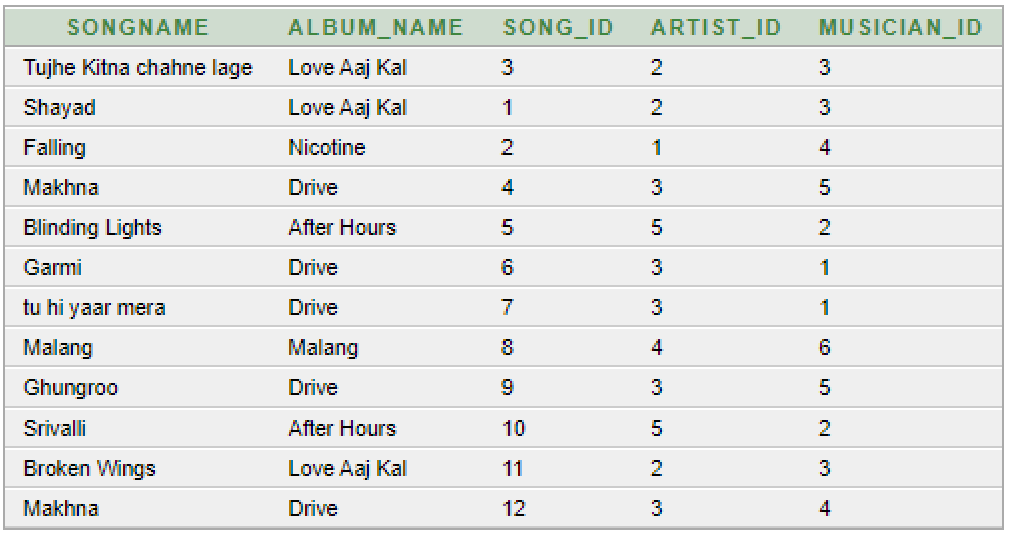


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SONG	SONGNAME	Varchar2	30	-	-	-	✓	-	-
	ALBUM_NAME	Varchar2	30	-	-	-	✓	-	-
	SONG_ID	Number	-	-	0	1	-	-	-
	ARTIST_ID	Number	-	-	0	-	✓	-	-
	MUSICIAN_ID	Number	-	-	0	-	✓	-	-



SONGNAME	ALBUM_NAME	SONG_ID	ARTIST_ID	MUSICIAN_ID
Tujhe Kitna chahne lage	Love Aaj Kal	3	2	3
Shayad	Love Aaj Kal	1	2	3
Falling	Nicotine	2	1	4
Makhna	Drive	4	3	5
Blinding Lights	After Hours	5	5	2
Garmi	Drive	6	3	1
tu hi yaar mera	Drive	7	3	1
Malang	Malang	8	4	6
Ghungroo	Drive	9	3	5
Srivalli	After Hours	10	5	2
Broken Wings	Love Aaj Kal	11	2	3
Makhna	Drive	12	3	4

3. Albums

The attributes used are Album ID, Album Name, Label Name, Artist ID, Release Date.

Primary Key used is Album ID.
Foreign Keys used are Artist ID, Label Name.

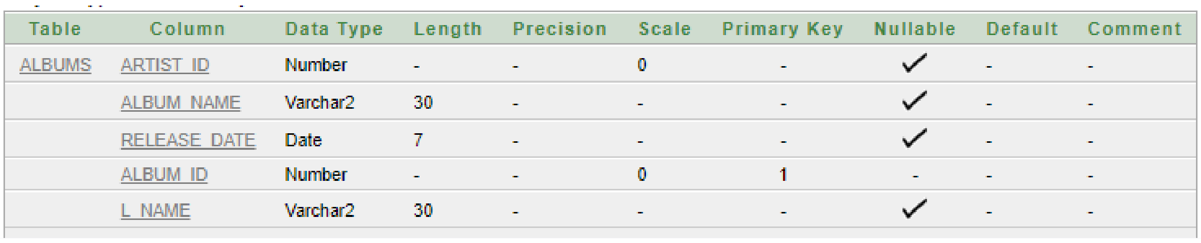
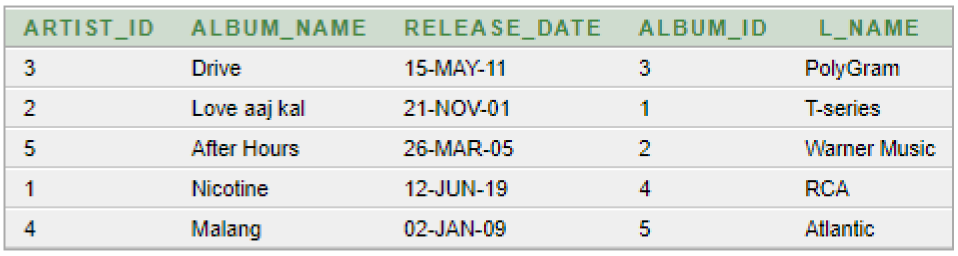


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ALBUMS	ARTIST_ID	Number	-	-	0	-	✓	-	-
	ALBUM_NAME	Varchar2	30	-	-	-	✓	-	-
	RELEASE_DATE	Date	7	-	-	-	✓	-	-
	ALBUM_ID	Number	-	-	0	1	-	-	-
	L_NAME	Varchar2	30	-	-	-	✓	-	-



ARTIST_ID	ALBUM_NAME	RELEASE_DATE	ALBUM_ID	L_NAME
3	Drive	15-MAY-11	3	PolyGram
2	Love aaj kal	21-NOV-01	1	T-series
5	After Hours	26-MAR-05	2	Warner Music
1	Nicotine	12-JUN-19	4	RCA
4	Malang	02-JAN-09	5	Atlantic

4. Artist

The attributes used are Artist ID, First Name, Middle Initial, Last Name.

Primary Key used is Artist ID.

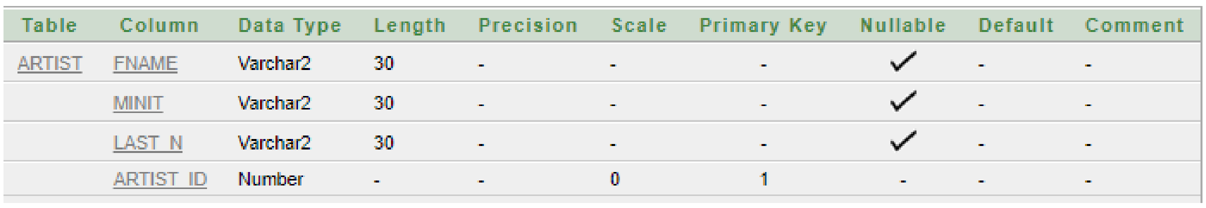
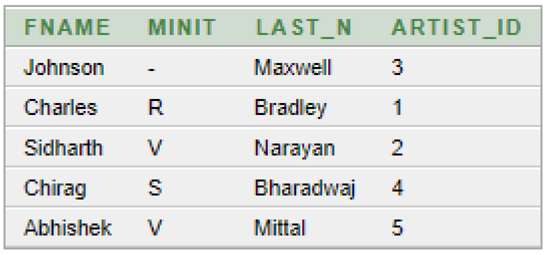


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ARTIST	FNAME	Varchar2	30	-	-	-	✓	-	-
	MINIT	Varchar2	30	-	-	-	✓	-	-
	LAST_N	Varchar2	30	-	-	-	✓	-	-
	ARTIST_ID	Number	-	-	0	1	-	-	-



FNAME	MINIT	LAST_N	ARTIST_ID
Johnson	-	Maxwell	3
Charles	R	Bradley	1
Sidharth	V	Narayan	2
Chirag	S	Bharadwaj	4
Abhishek	V	Mittal	5

5. Musician

The attributes used are Musician ID, First Name, Middle Initial, Last Name.

Primary Key used is Musician ID.

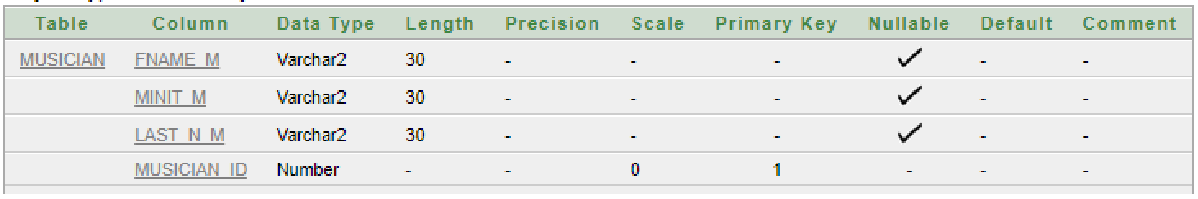
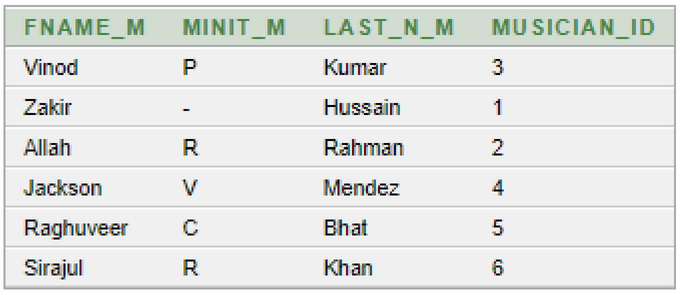


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MUSICIAN	FNAME_M	Varchar2	30	-	-	-	✓	-	-
	MINIT_M	Varchar2	30	-	-	-	✓	-	-
	LAST_N_M	Varchar2	30	-	-	-	✓	-	-
	MUSICIAN_ID	Number	-	-	0	1	-	-	-



FNAME_M	MINIT_M	LAST_N_M	MUSICIAN_ID
Vinod	P	Kumar	3
Zakir	-	Hussain	1
Allah	R	Rahman	2
Jackson	V	Mendez	4
Raghuveer	C	Bhat	5
Sirajul	R	Khan	6

6. Buyers

The attributes used are Bill Number, Album ID, Artist ID, Price Paid.

Primary Key used is Bill Number.
Foreign Keys used are Album ID, Artist ID.

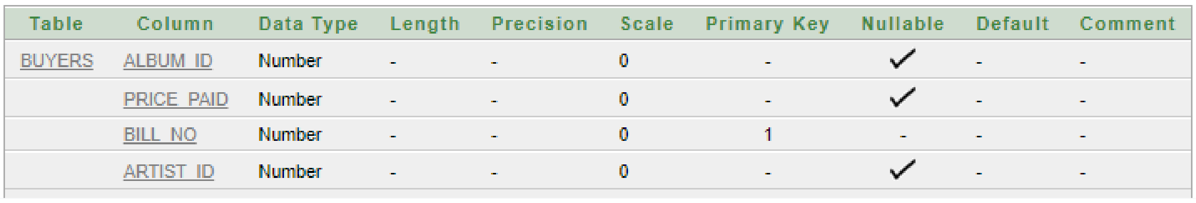
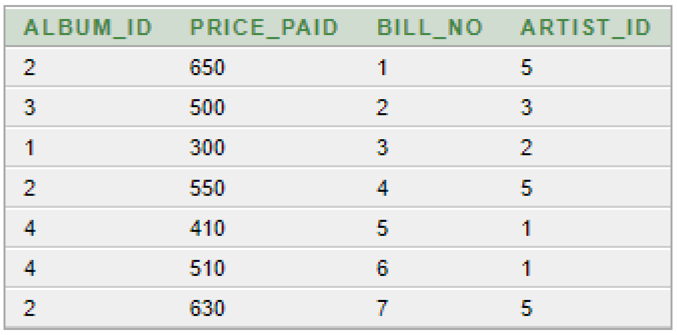
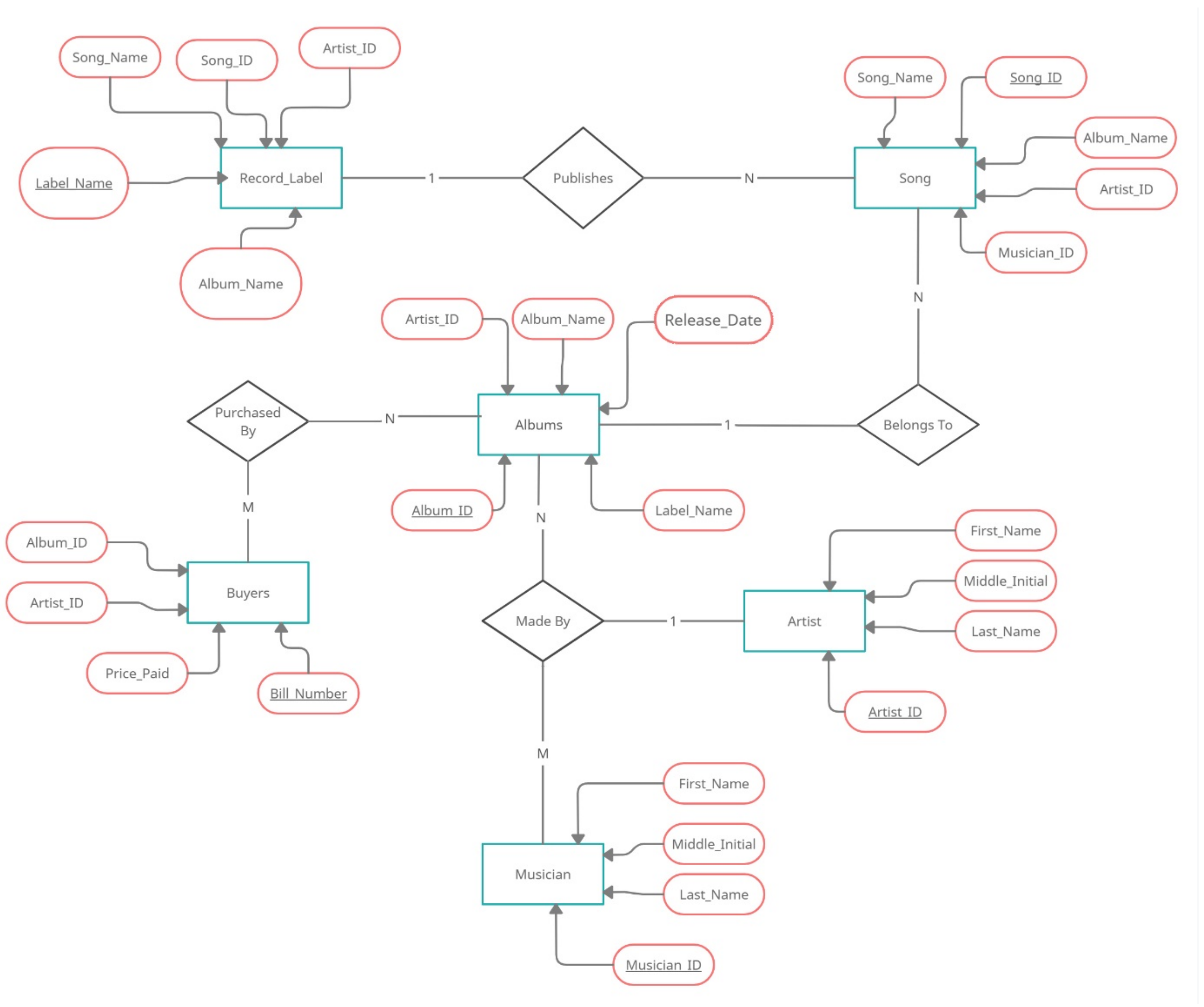


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BUYERS	ALBUM_ID	Number	-	-	0	-	✓	-	-
	PRICE_PAID	Number	-	-	0	-	✓	-	-
	BILL_NO	Number	-	-	0	1	-	-	-
	ARTIST_ID	Number	-	-	0	-	✓	-	-

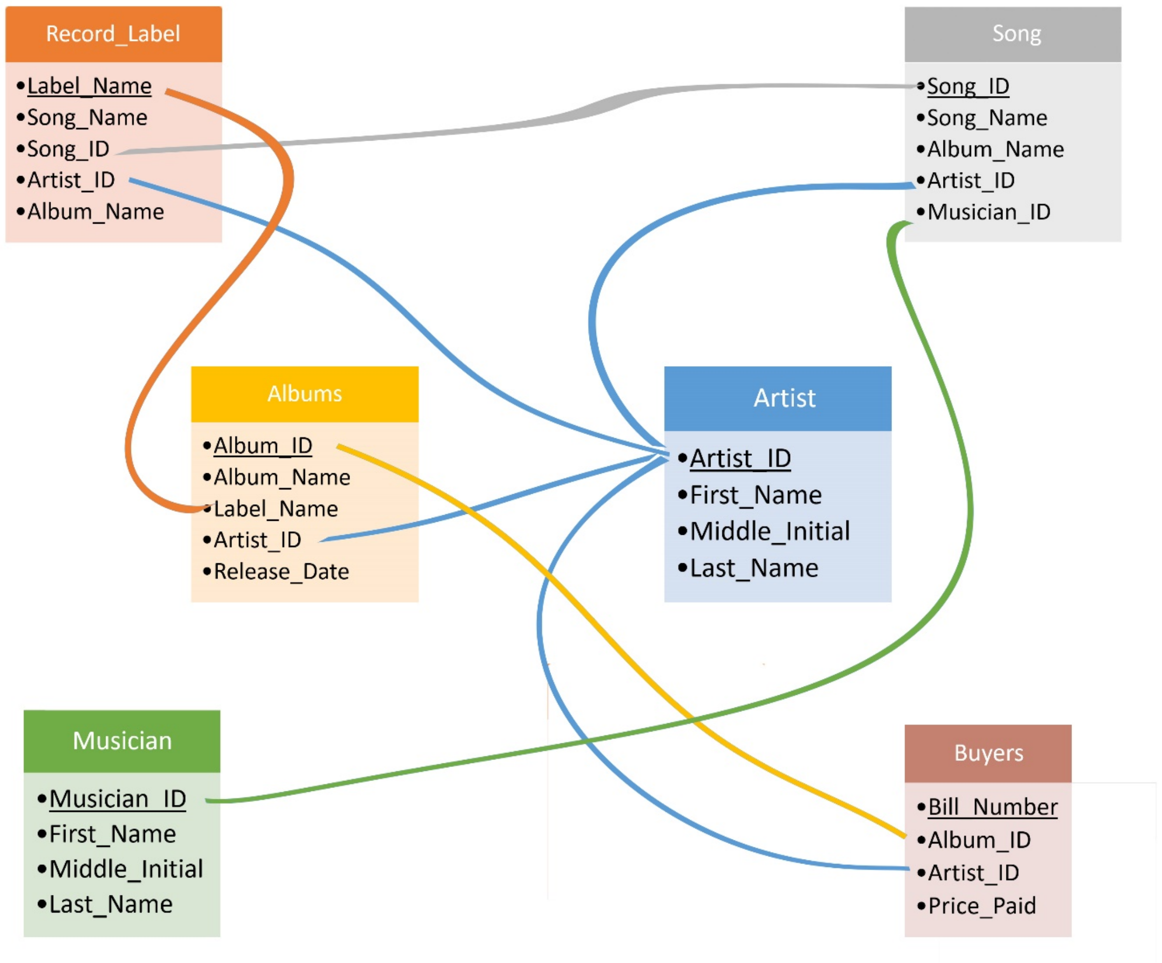


ALBUM_ID	PRICE_PAID	BILL_NO	ARTIST_ID
2	650	1	5
3	500	2	3
1	300	3	2
2	550	4	5
4	410	5	1
4	510	6	1
2	630	7	5

Entity Relationship Diagram:



Relational Schema Diagram:



Concepts Used In The Queries

1. The Create Statement:

Syntax:

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

Example:

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

2. The Create Constraints:

Syntax:

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ....
);
```

The constraints that can be used are:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified
- CREATE INDEX - Used to create and retrieve data from the database very quickly

Example:

```
CREATE TABLE Persons (
    PersonID int Primary Key,
    FirstName varchar(255),
);
```

Here we have used Primary key constraint and Foreign key constraint.

3. Insert Statement:

Syntax:

1) Specify both the column names and the values to be inserted:
```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

OR

2) If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table.
```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

Example:

1)
```
INSERT INTO Customers (CustomerName, ContactName, Address, City,
PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen
21', 'Stavanger', '4006', 'Norway');
```

2)
```
INSERT INTO Customers VALUES ('Cardinal', 'Tom B.
Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

4. The Select From Where Statement:

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, then:

Syntax:

```
SELECT * FROM table_name;
```

Example:

```
SELECT * FROM Customers
WHERE Country='Mexico';
```

5. The Aliases:

SQL aliases are used to give a table, or a column in a table, a temporary name.

An alias is created with the `AS` keyword.

Syntax:

```
SELECT column_name AS alias_name
FROM table_name1 tn1;
```

Example:

```
SELECT CustomerID AS ID, CustomerName AS Customer
FROM Customers;
```

6. The Group By Clause:

The GROUP BY statement groups rows that have the same values into summary rows.
The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Example:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

7. Nested Queries:

Query written inside a query is called nested query.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator
  (SELECT column_name
  FROM table_name
  WHERE condition);
```

Example:

```
SELECT ProductName
FROM Products
WHERE exists
  (SELECT ProductID
  FROM OrderDetails
  WHERE Quantity = 10);
```

8. SQL Operators

- ALL means that the condition will be true only if the operation is true for all values in the range.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
  (SELECT column_name
  FROM table_name
  WHERE condition);
```

Example:

```
SELECT ProductName
FROM Products
WHERE ProductID = ANY
  (SELECT ProductID
  FROM OrderDetails
  WHERE Quantity > 99);
```

- The `IN` operator allows you to specify multiple values in a `WHERE` clause.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

Example:

```
SELECT * FROM Customers
WHERE Country IN (SELECT Country FROM Suppliers);
```

- The `EXISTS` operator is used to test for the existence of any record in a subquery.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE condition);
```

Example:

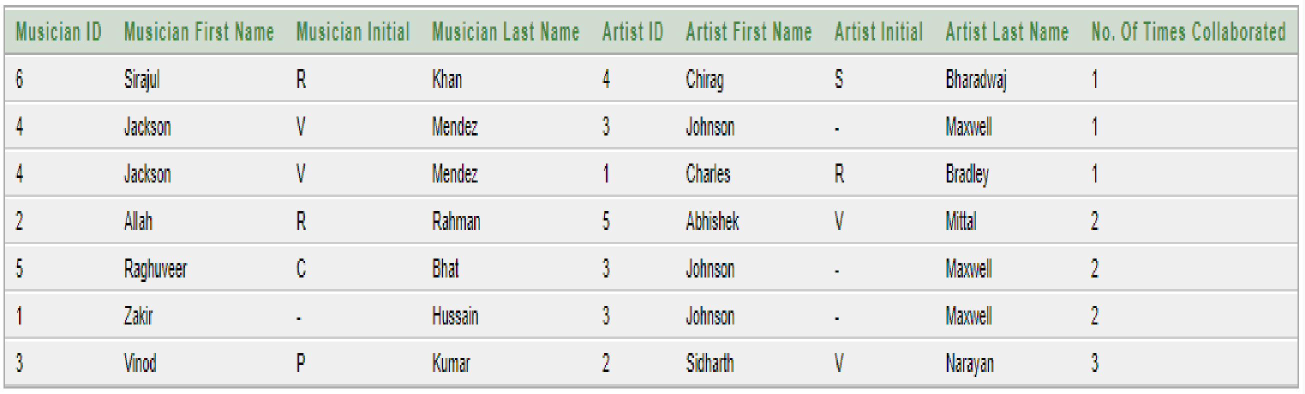
```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Products WHERE P
roducts.SupplierID = Suppliers.supplierID AND Price
< 20);
```

Given SQL Queries

1. Which artist has worked the maximum times with musician?

```
Select m.MUSICIAN_ID as "Musician ID", m.FNAME_M as "Musician First
Name", m.MINIT_M as "Musician Initial", m.LAST_N_M as "Musician Last
Name",  a.ARTIST_ID as "Artist ID", a.FNAME  as "Artist First Name",
a.MINIT  as "Artist Initial", a.LAST_N  as "Artist Last Name", count(*) as "No.
Of Times Collaborated"

From Musician m, Artist a, Song sWhere s.ARTIST_ID=a.ARTIST_ID and
s.MUSICIAN_ID= m.MUSICIAN_ID

Group By m.MUSICIAN_ID, m.FNAME_M, m.MINIT_M, m.LAST_N_M,
a.ARTIST_ID, a.FNAME, a.MINIT, a.LAST_N

Order By count(*)
```

Output:

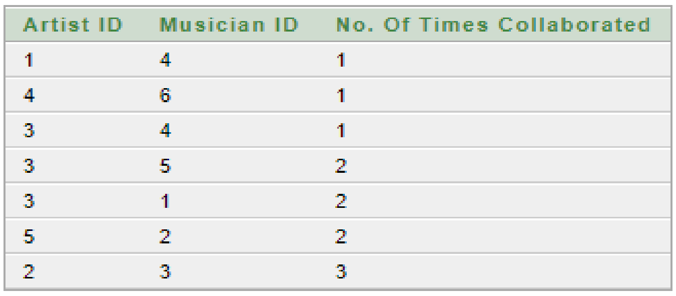


Musician ID	Musician First Name	Musician Initial	Musician Last Name	Artist ID	Artist First Name	Artist Initial	Artist Last Name	No. Of Times Collaborated
6	Sirajul	R	Khan	4	Chirag	S	Bharadvaj	1
4	Jackson	V	Mendez	3	Johnson	-	Maxwell	1
4	Jackson	V	Mendez	1	Charles	R	Bradley	1
2	Allah	R	Rahman	5	Abhishek	V	Mittal	2
5	Raghuveer	C	Bhat	3	Johnson	-	Maxwell	2
1	Zakir	-	Hussain	3	Johnson	-	Maxwell	2
3	Vinod	P	Kumar	2	Sidharth	V	Narayan	3

Summary version:

```
Select ARTIST_ID as "Artist ID", MUSICIAN_ID as "Musician ID", count(*) as
"No. Of Times Collaborated"

From song group by MUSICIAN_ID, ARTIST_ID

Order By count(*)
```

Output:

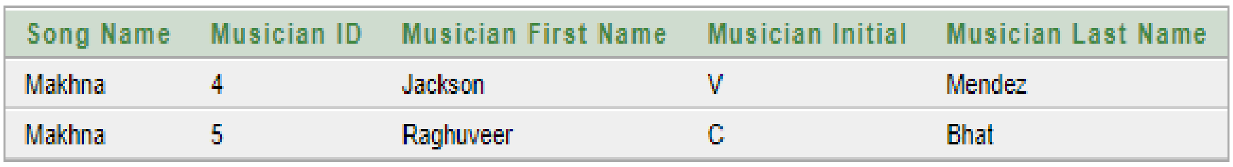


Artist ID	Musician ID	No. Of Times Collaborated
1	4	1
4	6	1
3	4	1
3	5	2
3	1	2
5	2	2
2	3	3

2. Given the details of musicians involved in composition of <Song Name> 'Makhna'.

```
Select s.SONGNAME as "Song Name", m.MUSICIAN_ID as "Musician ID",
m.FNAME_M as "Musician First Name", m.MINIT_M as "Musician Initial",
m.LAST_N_M as "Musician Last Name"

From Musician m, Song s

Where  s.MUSICIAN_ID= m.MUSICIAN_ID  and s.SONGNAME= 'Makhna'
```

Output:

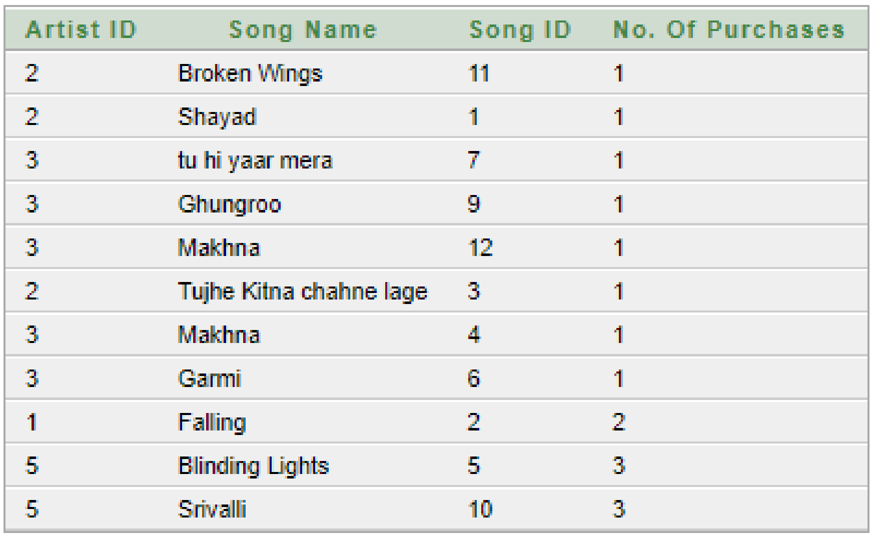


Song Name	Musician ID	Musician First Name	Musician Initial	Musician Last Name
Makhna	4	Jackson	V	Mendez
Makhna	5	Raghuveer	C	Bhat

3. Which song is least liked by the buyers?

```
Select b.ARTIST_ID as "Artist ID", s.SONGNAME as "Song Name",
s.Song_id as "Song ID", count(*) as "No. Of Purchases"
From Song s, Artist a, Buyers b
Where s.ARTIST_ID=a.ARTIST_ID and a.ARTIST_ID=b.ARTIST_ID
Group By b.ARTIST_ID, s.SONGNAME, s.Song_id
Order By count(*)
```

Output:

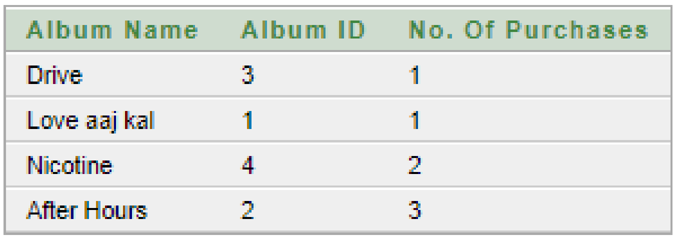


Artist ID	Song Name	Song ID	No. Of Purchases
2	Broken Wings	11	1
2	Shayad	1	1
3	tu hi yaar mera	7	1
3	Ghungroo	9	1
3	Makhna	12	1
2	Tujhe Kitna chahne lage	3	1
3	Makhna	4	1
3	Garmi	6	1
1	Falling	2	2
5	Blinding Lights	5	3
5	Srivalli	10	3

Which album is least liked by the buyers?

```
Select a.ALBUM_NAME as "Album Name", b.ALBUM_ID as "Album ID",
count(*) as "No. Of Purchases"

From Buyers b, Albums a

Where a.ALBUM_ID=b.ALBUM_ID

Group By a.ALBUM_NAME, b.ALBUM_ID

Order By count(*)
```

Output:

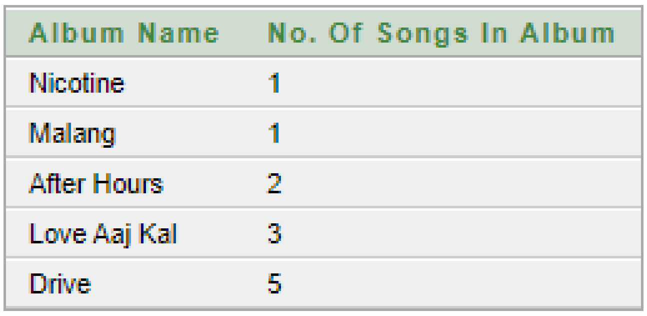


Album Name	Album ID	No. Of Purchases
Drive	3	1
Love aaj kal	1	1
Nicotine	4	2
After Hours	2	3

4. How many songs are there in each album?

```
Select ALBUM_NAME as "Album Name", count(*) as "No. Of Songs In
Album"
From Song
Group  By ALBUM_NAME
Order By count(*)
```

Output:

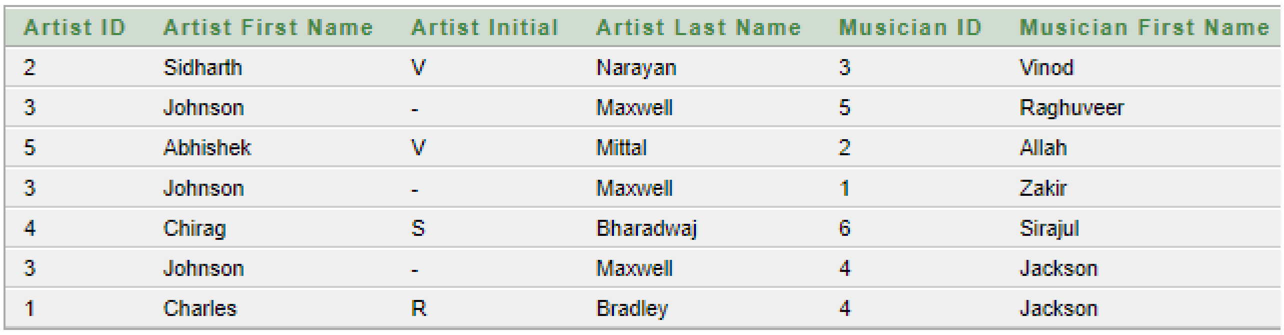
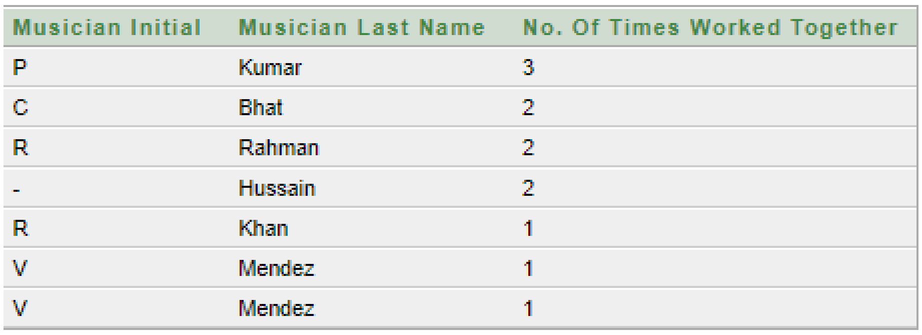


Album Name	No. Of Songs In Album
Nicotine	1
Malang	1
After Hours	2
Love Aaj Kal	3
Drive	5

Give the details of artist & musician with number of times they have worked together.

```
Select a.ARTIST_ID as "Artist ID", a.FNAME  as "Artist First Name", a.MINIT
as "Artist Initial", a.LAST_N  as "Artist Last Name", m.MUSICIAN_ID as
"Musician ID", m.FNAME_M as "Musician First Name", m.MINIT_M as
"Musician Initial", m.LAST_N_M as "Musician Last Name", count(*) as "No. Of
Times Done together"
From Musician m, Song s, Artist a
Where s.ARTIST_ID=a.ARTIST_ID and s.MUSICIAN_ID=m.MUSICIAN_ID
and exists
          (Select s.Song_id
          From Song s
          Where exists
                    (Select s.ARTIST_ID, s.Song_id
                     From Song s
                    Group By s.ARTIST_ID, s.Song_id)

                                             and exists

                    (Select s.Song_id, s.MUSICIAN_ID
                     From Song s
                     Group By s.Song_id, s.MUSICIAN_ID) )
Group By a.ARTIST_ID, a.FNAME, a.MINIT, a.LAST_N, m.MUSICIAN_ID,
m.FNAME_M, m.MINIT_M, m.LAST_N_M
 Order By count(*) desc
```

Output:

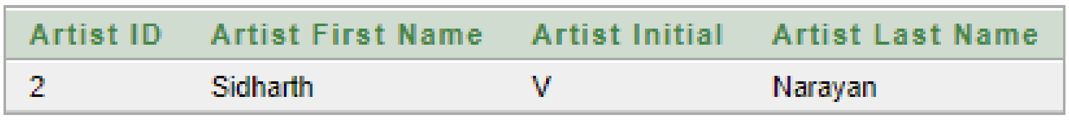



Artist ID	Artist First Name	Artist Initial	Artist Last Name	Musician ID	Musician First Name	Musician Initial	Musician Last Name	No. Of Times Worked Together
2	Sidharth	V	Narayan	3	Vinod	P	Kumar	3
3	Johnson	-	Maxwell	5	Raghuveer	C	Bhat	2
5	Abhishek	V	Mittal	2	Allah	R	Rahman	2
3	Johnson	-	Maxwell	1	Zakir	-	Hussain	2
4	Chirag	S	Bharadwaj	6	Sirajul	R	Khan	1
3	Johnson	-	Maxwell	4	Jackson	V	Mendez	1
1	Charles	R	Bradley	4	Jackson	V	Mendez	1

5. Give the details of artist who has worked in all songs of a <Musician Name> 'Vinod P Kumar'.

```
Select ARTIST_ID as "Artist ID", FNAME  as "Artist First Name", MINIT  as
"Artist Initial", LAST_N  as "Artist Last Name"
From Artist
Where ARTIST_ID = ALL

        (Select ARTIST_ID
         From Song
         Where MUSICIAN_ID IN

                (Select MUSICIAN_ID
                 From Musician
                 Where FNAME_M = 'Vinod' and MINIT_M = 'P' and LAST_N_M
                = 'Kumar')

        )
```

Output:



Artist ID	Artist First Name	Artist Initial	Artist Last Name
2	Sidharth	V	Narayan