Music Company Database

Bachelor of Technology
Computer Science and Engineering

Submitted By

ANURAG GANGULY (13000117125)
ANUPAM CHAKRABARTY (13000117125)
ANIKET DAS (13000117130)
ANANYA PAUL (13000117131)

June 2020

Techno Main Salt Lake
EM-4/1, Sector-V, Salt Lake
Kolkata-
700091 West
Bengal India

Table of Contents

Abstract **3**

Introduction **3**

Body **4**

i) EER Diagram 4

ii)Relational Model 4

iii) Table Creation 5

iv) Insertion 7

v) Queries 8

Normalisation **13**

Conclusion **14**

References **14**

Abstract

A database is a collection of data relevant to a particular enterprise. The computer performs various programming languages, such as C, C++, Java, Python, SQL to process large and complex data more effectively and quickly. That is the reason we find them valuable to organizations, particularly in the creation of system database management services.

SQL is a database computer language designed for the retrieval and management of data in a relational database**.** SQL stands for Structured Query Language.

We have prepared a music company database which enables a user to insert, update, and retrieve information about a song, it's artist, producer, composer etc.

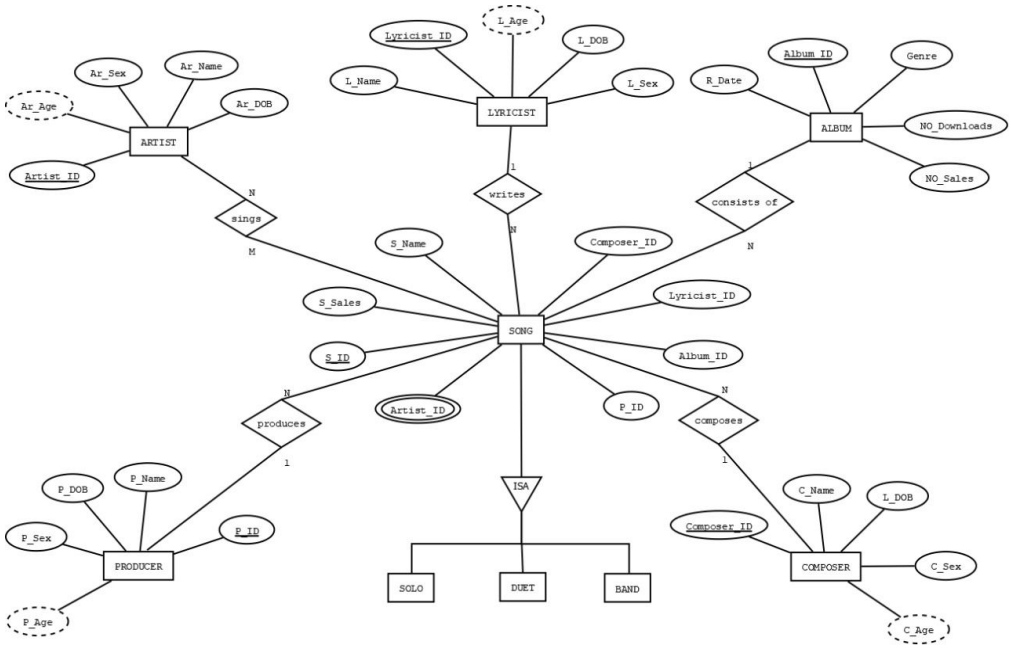
Introduction

A music company database consists of relevant data related to songs, artists, albums, composer and producer. It also consists of certain details with regards to the type of song, it's ID, number of sales etc. The names of the singer, producer, composer, their gender, the release date of an album—all these relevant information and details have been added. We have also created an EER (Extended Entity-Relationship) diagram and a relational model has been added as well.

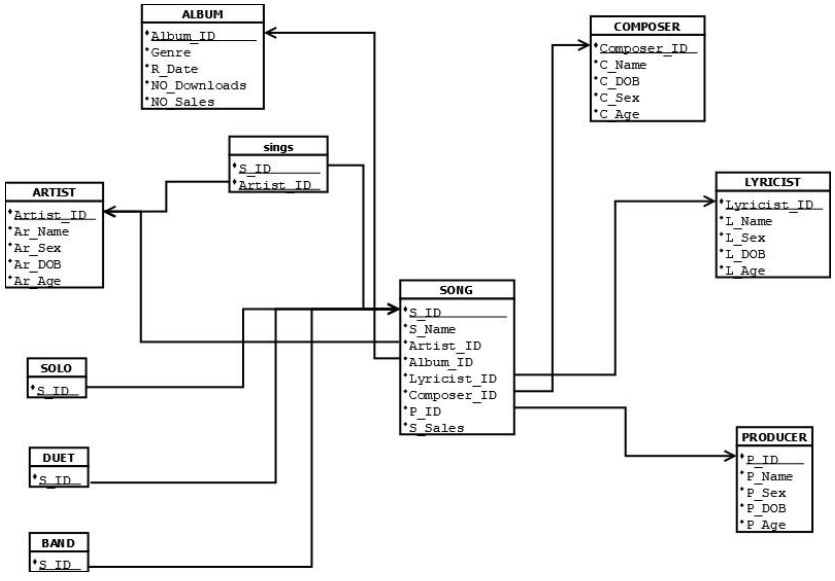
Some queries which consist of DDL, DML, DQL commands, Set, Aggregate operators and other functionalities have been added too. Thus, this database allows a user to operate within it and retrieve information accordingly.

Body

i) EER Diagram



ii)Relational Model



iii) Table Creation

```
create table ARTIST
(
   Artist_ID varchar(6) primary key check(Artist_ID like 'Ar%'),
   Ar_Name varchar(20) not null,
   Ar_Sex varchar(10) not null,
   Ar_DOB date not null,
   Ar_Age number(3) not null
);
```

```
create table ALBUM
(
   Album_ID varchar(6) primary key check(Album_ID like 'A%'),
   Genre varchar(20) not null,
   R_Date date,
   NO_Downloads number(6) not null,
   NO_Sales number(6) not null
);
```

```
create table COMPOSER
(
   Composer_ID varchar(6) primary key check(Composer_ID like 'C%'),
   C_Name varchar(20) not null,
   C_Sex varchar(10) not null,
   C_DOB date not null,
   C_Age number(3) not null
);
```

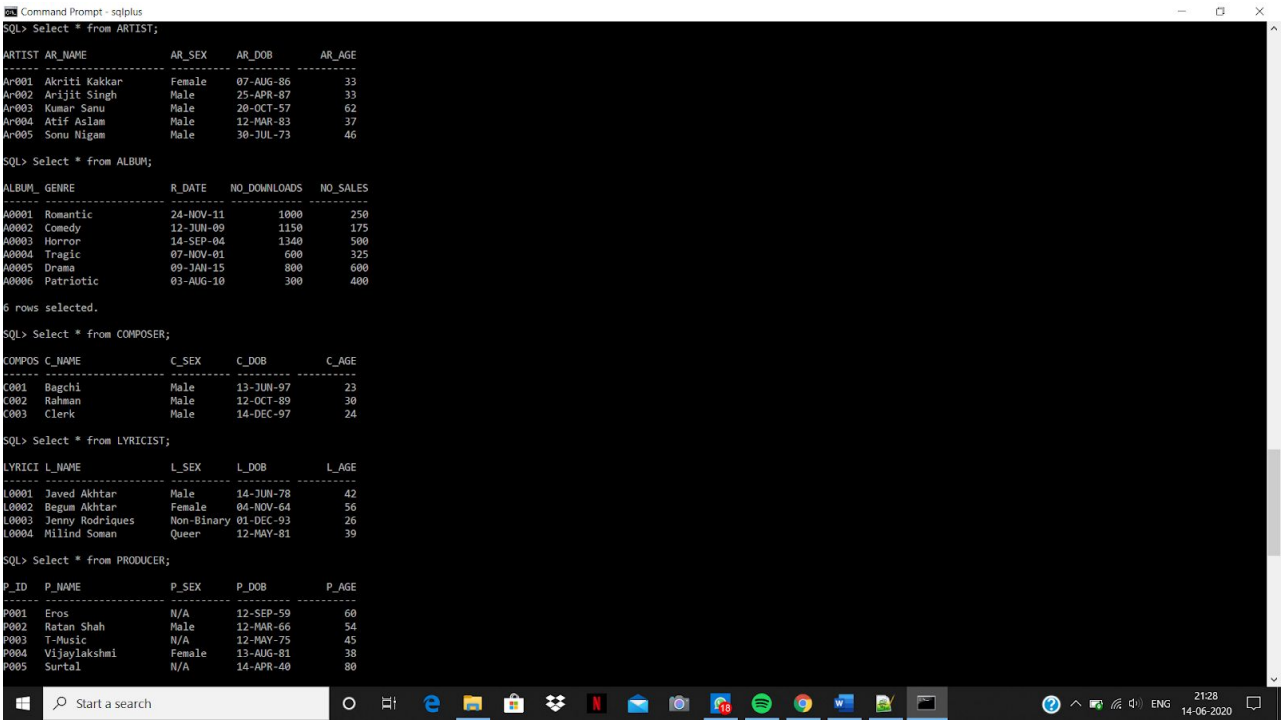
```
create table LYRICIST
(
   Lyricist_ID varchar(6) primary key check(Lyricist_ID like 'L%'),
   L_Name varchar(20) not null,
   L_Sex varchar(10) not null,
   L_DOB date not null,
   L_Age number(3) not null
);
```

```
create table PRODUCER
(
   P_ID varchar(6) primary key check(P_ID like 'P%'),
   P_Name varchar(20) not null,
   P_Sex varchar(10) not null,
   P_DOB date not null,
   P_Age number(3) not null
);
```

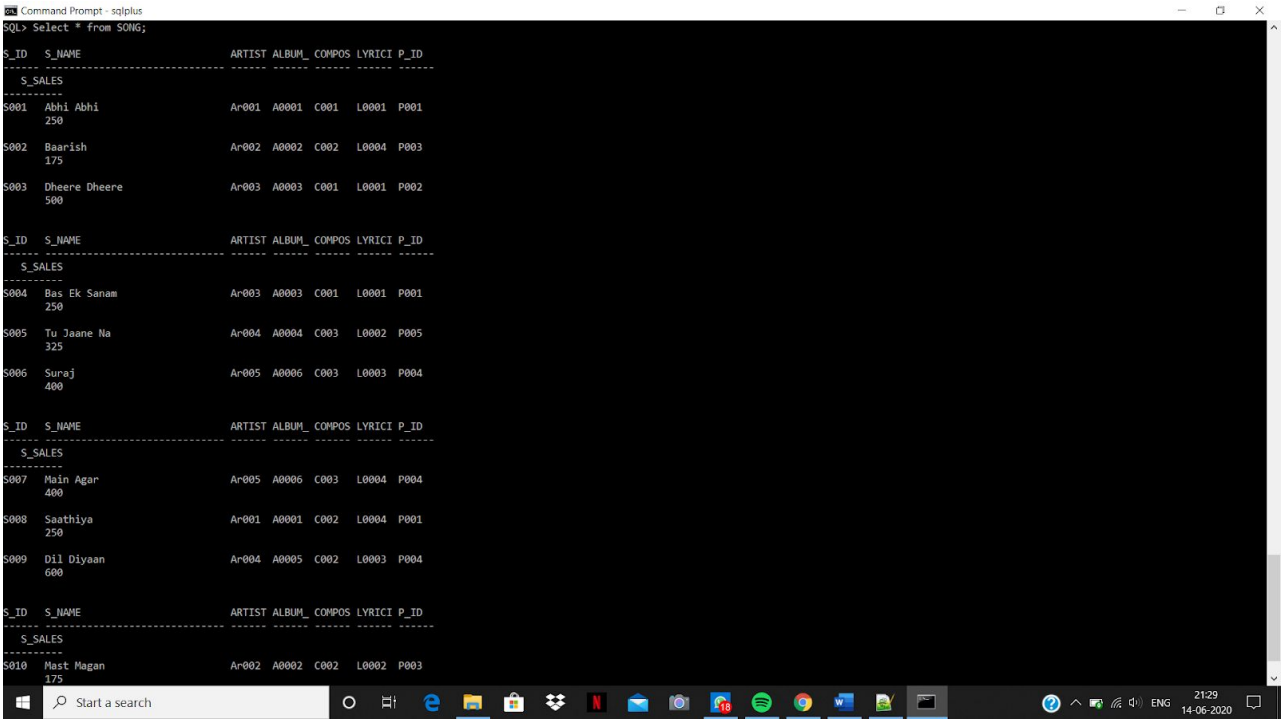
```
create table SONG
(
   S_ID varchar(6) primary key check(S_ID like 'S%'),
   S_Name varchar(30) not null,
   Artist_ID varchar(6) not null,
   Album_ID varchar(6) not null,
   Composer_ID varchar(6) not null,
   Lyricist_ID varchar(6) not null,
   P_ID varchar(6) not null,
   S_Sales number(6) not null,
   foreign key(Artist_ID) references ARTIST(Artist_ID) on delete cascade,
   foreign key(Composer_ID) references COMPOSER(Composer_ID) on delete cascade,
   foreign key(Lyricist_ID) references LYRICIST(Lyricist_ID) on delete cascade,
   foreign key(P_ID) references PRODUCER(P_ID) on delete cascade,
   foreign key(Album_ID) references ALBUM(Album_ID) on delete cascade
);
```

iv) Insertion

In this section, we are displaying each and every view after it is inserted with all the desired values



```
Command Prompt - sqlplus
SQL> Select * from ARTIST;

ARTIST AR_NAME              AR_SEX     AR_DOB       AR_AGE
------ -------------------- ---------- --------- ----------
Ar001  Akriti Kakkar        Female     07-AUG-86         33
Ar002  Arijit Singh         Male       25-APR-87         33
Ar003  Kumar Sanu           Male       20-OCT-57         62
Ar004  Atif Aslam           Male       12-MAR-83         37
Ar005  Sonu Nigam           Male       30-JUL-73         46

SQL> Select * from ALBUM;

ALBUM_ GENRE               R_DATE    NO_DOWNLOADS   NO_SALES
------ -------------------- --------- ------------ ----------
A0001  Romantic             24-NOV-11         1000        250
A0002  Comedy               12-JUN-09         1150        175
A0003  Horror               14-SEP-04         1340        500
A0004  Tragic               07-NOV-01          600        325
A0005  Drama                09-JAN-15          800        600
A0006  Patriotic            03-AUG-10          300        400

6 rows selected.

SQL> Select * from COMPOSER;

COMPOS C_NAME              C_SEX      C_DOB         C_AGE
------ -------------------- ---------- --------- ----------
C001   Bagchi               Male       13-JUN-97         23
C002   Rahman               Male       12-OCT-89         30
C003   Clerk                Male       14-DEC-97         24

SQL> Select * from LYRICIST;

LYRICI L_NAME              L_SEX      L_DOB         L_AGE
------ -------------------- ---------- --------- ----------
L0001  Javed Akhtar         Male       14-JUN-78         42
L0002  Begum Akhtar         Female     04-NOV-64         56
L0003  Jenny Rodriques      Non-Binary 01-DEC-93         26
L0004  Milind Soman         Queer      12-MAY-81         39

SQL> Select * from PRODUCER;

P_ID   P_NAME              P_SEX      P_DOB         P_AGE
------ -------------------- ---------- --------- ----------
P001   Eros                N/A        12-SEP-59         60
P002   Ratan Shah           Male       12-MAR-66         54
P003   T-Music              N/A        12-MAY-75         45
P004   Vijaylakshmi         Female     13-AUG-81         38
P005   Surtal               N/A        14-APR-40         80
```



```
Command Prompt - sqlplus
SQL> Select * from SONG;

S_ID   S_NAME                        ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S001   Abhi Abhi                     Ar001  A0001  C001   L0001  P001
       250

S002   Baarish                       Ar002  A0002  C002   L0004  P003
       175

S003   Dheere Dheere                 Ar003  A0003  C001   L0001  P002
       500


S_ID   S_NAME                        ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S004   Bas Ek Sanam                  Ar003  A0003  C001   L0001  P001
       250

S005   Tu Jaane Na                   Ar004  A0004  C003   L0002  P005
       325

S006   Suraj                         Ar005  A0006  C003   L0003  P004
       400


S_ID   S_NAME                        ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S007   Main Agar                     Ar005  A0006  C003   L0004  P004
       400

S008   Saathiya                      Ar001  A0001  C002   L0004  P001
       250

S009   Dil Diyaan                    Ar004  A0005  C002   L0003  P004
       600


S_ID   S_NAME                        ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S010   Mast Magan                    Ar002  A0002  C002   L0002  P003
       175
```

v) Queries

q) Find the names of artists who have recorded more than one song

```
SQL> select * from ARTIST
  2  where Artist_ID in (
  3  select Artist_ID from SONG
  4  group by Artist_ID
  5  having count(*)>1
  6  );

ARTIST AR_NAME              AR_SEX     AR_DOB        AR_AGE
------ -------------------- ---------- --------- ----------
Ar001  Akriti Kakkar        Female     07-AUG-86         33
Ar002  Arijit Singh         Male       25-APR-87         33
Ar003  Kumar Sanu           Male       20-OCT-57         62
Ar004  Atif Aslam           Male       12-MAR-83         37
Ar005  Sonu Nigam           Male       30-JUL-73         46
```

q) List the most populous sung by an artist

```
SQL> select * from SONG
  2  where S_Sales = (
  3      select max(S_Sales) from SONG
  4  );

S_ID   S_NAME                         ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S009   Dil Diyaan                     Ar004  A0005  C002   L0003  P004
       600
```

q) List the albums whose downloads count have exceeded 1000

```
SQL> select * from ALBUM
  2  where NO_Downloads > 1000;

ALBUM_ GENRE                R_DATE    NO_DOWNLOADS   NO_SALES
------ -------------------- --------- ------------ ----------
A0002  Comedy               12-JUN-09         1150        175
A0003  Horror               14-SEP-04         1340        500
```

q) List all the composers who have been born after 1980, and whose albums have been downloaded more than 500 times

```
SQL> select * from COMPOSER
  2  where Composer_ID in (
  3      select Composer_ID from COMPOSER
  4      natural join SONG
  5      natural join ALBUM
  6      where C_DOB > '31-DEC-1980' and NO_Downloads > 500
  7  );

COMPOS C_NAME              C_SEX      C_DOB          C_AGE
------ ------------------- ---------- --------- ----------
C001   Bagchi              Male       13-JUN-97         23
C002   Rahman              Male       12-OCT-89         30
C003   Clerk               Male       14-DEC-97         24
```

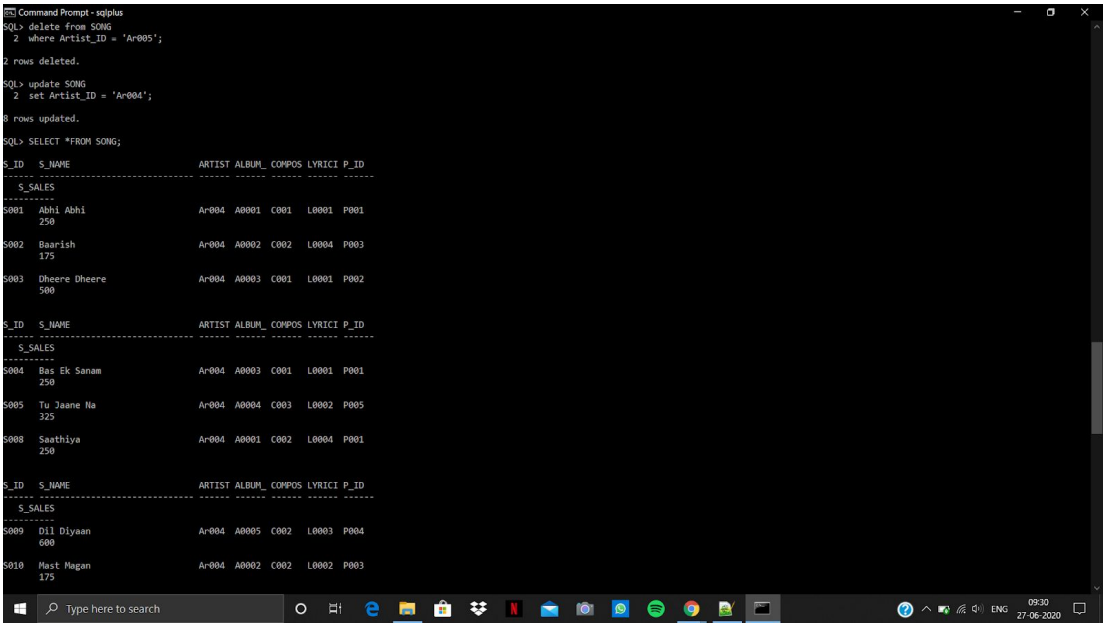
q) List all the songs produced by male producers whose sales have crossed 450

```
SQL> select * from PRODUCER
  2  where P_Sex = 'Male' and P_ID in (
  3      select P_ID from SONG
  4      where S_Sales > 450
  5  );

P_ID   P_NAME               P_SEX      P_DOB          P_AGE
------ -------------------- ---------- --------- ----------
P002   Ratan Shah           Male       12-MAR-66          54
```

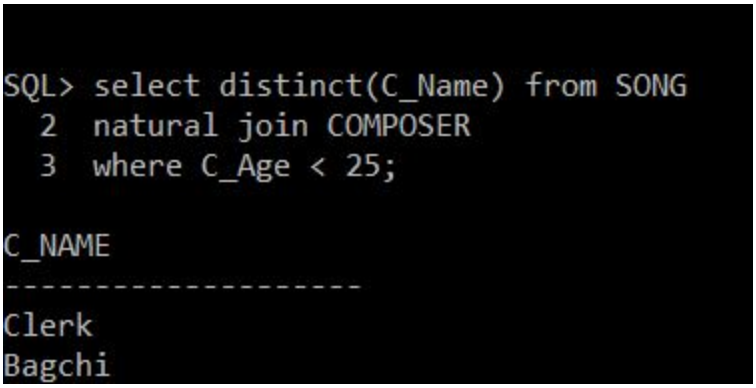
q) List all the patriotic songs recorded by multiple artists produced after 1990

```
SQL> select * from SONG
  2  where Album_ID in (
  3      select Album_ID from ALBUM
  4      where Genre = 'Patriotic'
  5  );

S_ID   S_NAME                         ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S006   Suraj                          Ar005  A0006  C003   L0003  P004
       400

S007   Main Agar                      Ar005  A0006  C003   L0004  P004
       400
```

q) Delete all the songs by "Sonu Nigam" , update the database with songs of Atif Aslam



q) Find out the names of all composers aging below 25 who have composed at most 2 songs



q) List all the songs produced under the banner "Eros", also sung by "Akriti Kakkar"

```
SQL> select * from SONG
  2  where P_ID = 'P001' and Artist_ID = 'Ar001';

S_ID   S_NAME                         ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S001   Abhi Abhi                      Ar001  A0001  C001   L0001  P001
       250

S008   Saathiya                       Ar001  A0001  C002   L0004  P001
       250
```

q) List all the songs sung by "Atif Aslam" under the banner "T-Music" after 2005

```
SQL> select * from SONG
  2  where Artist_ID = 'Ar004' and Album_ID in (
  3      select Album_ID from ALBUM
  4      where R_Date > '31-DEC-2005'
  5  );

S_ID   S_NAME                         ARTIST ALBUM_ COMPOS LYRICI P_ID
------ ------------------------------ ------ ------ ------ ------ ------
   S_SALES
----------
S009   Dil Diyaan                     Ar004  A0005  C002   L0003  P004
       600
```

q) List the total downloads' count

```
SQL> select sum(NO_Downloads) from ALBUM;

SUM(NO_DOWNLOADS)
-----------------
             5190
```

Normalisation

Normalization is the process of organizing the data in the database. **Normalization** is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies

Here are the most commonly used normal forms:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce & Codd normal form (BCNF)

Our database is normalised and it is in BCNF(Boyce Codd's Normal Form). Reasons are:-

1)Atomicity preserved, satisfying **1NF**

2)No non-prime attributes is dependent on the proper subset of the candidate key, satisfying **2NF**

3)Removal of transitive functional dependency and ensuring each of the attributes is directly dependent on the super key/candidate key, satisfying **3NF**

4)Since, all the attributes from each and every table are **strictly** dependent on the super key, thus database is in BCNF(**Stricter** form of 3NF)

Conclusion

Our music company database can be enhanced further to create a grand database which can hold huge entries of data ranging from different songs, artists produced under different production companies/independent producers. Our project is just a miniature reflection of our ideas which we would try to accomplish in the near future.

This project can be enhanced in its performance with the use of Embedded SQL. A C program can be interfaced with SQL queries using connection libraries. We aim to create an API to interact with database servers . We can use the concept of Bigtable - A fully managed, scalable NoSQL database service for large analytical and operational workloads.

Our future scope enlists multiple dimensions in magnifying the utility of our project.

References

We have prepared our project by taking help from various online sources. These have been mentioned below:

- https://www.w3schools.com/sql/
- https://beginnersbook.com/2015/04/dbms-tutorial/
- Jenny's lectures on DBMS(Youtube)
- Database system concepts by Henry F.Korth, Silberschatz Abhraham.
- NPTEL Lectures on DBMS

Apart from these, the lecture notes provided by our teacher have been of immense help. We are grateful to her for providing the necessary help and guiding us along the way.