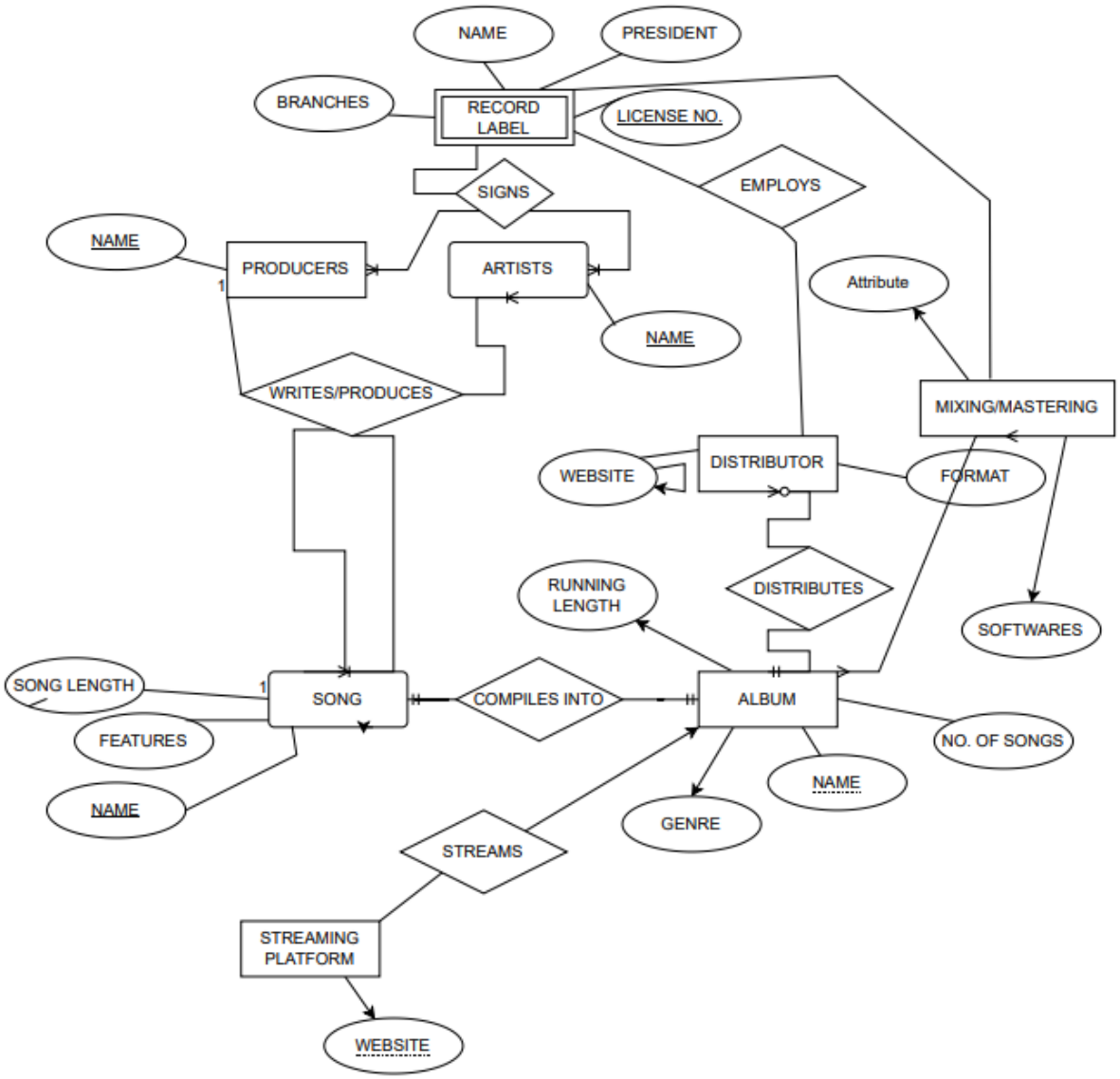
DBMS- Mini Project
RECORD LABEL DATABASE

Submitted By:
Name:VINESH S
SRN:PES1UG20CS505
V Semester Section I

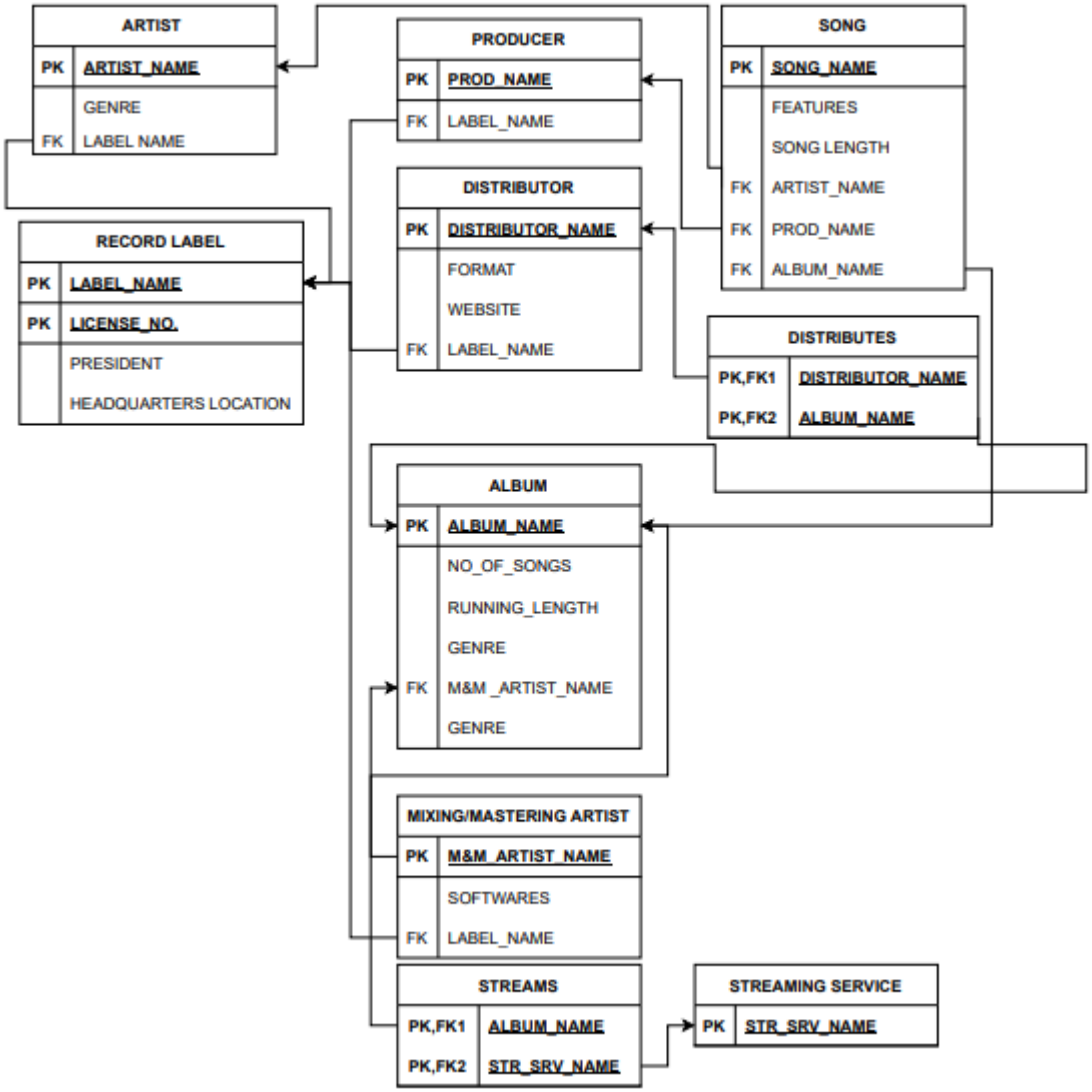
Short Description and Scope of the Project:

This database application aims on maintaining the records of one of the most fundamental foundations of today's music industry, the record labels. A record label provides support to various artist by helping them distribute their music, market it, give them a bigger audience by signing them under themselves. There are various nuances in making a record such as composing, producing mixing and mastering to deliver a final product. This final product has to be distributed in forms of streams and physical vinyl records by distributors. This application is a sample of such a database in action.

ER Diagram



Relational Schema



DDL statements - Building the database

```
CREATE TABLE `dbms_project`.`artist` (
  `Artist_name` CHAR NOT NULL,
  `Genre` VARCHAR(45) NULL,
  PRIMARY KEY (`Artist_name`))
ALTER TABLE `dbms_project`.`artist`
ADD COLUMN `label_name` VARCHAR(45) NULL AFTER `Genre`;
```

```
CREATE TABLE `dbms_project`.`record_label` (
  `Label_name` CHAR NULL,
  `license_no` INT NOT NULL,
  `President` VARCHAR(45) NULL,
  `Headquarters/location` VARCHAR(45) NULL,
  PRIMARY KEY (`Label_name`, `license_no`));
ALTER TABLE `dbms_project`.`record_label`
CHANGE COLUMN `Label_name` `Label_name` VARCHAR(45) NOT NULL ;

CREATE TABLE `dbms_project`.`producer` (
  `prod_name` VARCHAR(45) NULL,
  `label_name` VARCHAR(45) NULL,
  PRIMARY KEY (`prod_name`));

CREATE TABLE `dbms_project`.`song` (
  `song_name` VARCHAR(45) NULL,
  `feature` VARCHAR(45) NULL,
  `song_length` INT NULL,
  `artist_name` VARCHAR(45) NULL,
  `prod_name` VARCHAR(45) NULL,
  PRIMARY KEY (`song_name`));
ALTER TABLE `dbms_project`.`song`
ADD COLUMN `album_name` VARCHAR(45) NULL AFTER `prod_name`;


CREATE TABLE `dbms_project`.`album` (
  `album_name` VARCHAR(45) NULL,
  `no_of_songs` INT NULL,
  `running_length` INT NULL,
  `album_genre` VARCHAR(45) NULL,
  `m&m_artist_name` VARCHAR(45) NULL,
```

```
  PRIMARY KEY (`album_name`));



CREATE TABLE `dbms_project`.`distributor` (
  `distributor_name` VARCHAR(45) NOT NULL,
  `format` VARCHAR(45) NULL,
  `website` VARCHAR(45) NULL,
  `label_name` VARCHAR(45) NULL,
  PRIMARY KEY (`distributor_name`));

CREATE TABLE `dbms_project`.`mixing&mastering_artist` (
  `m&m_artist_name` VARCHAR(45) NOT NULL,
  `softwares` VARCHAR(45) NULL,
  `label_name` VARCHAR(45) NULL,
  PRIMARY KEY (`m&m_artist_name`));

CREATE TABLE `dbms_project`.`streaming_service` (
  `str_srv_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`str_srv_name`));

CREATE TABLE `dbms_project`.`distributes` (
  `distributor_name` VARCHAR(45) NOT NULL,
  `album_name` VARCHAR(45) NULL);

CREATE TABLE `dbms_project`.`streams` (
  `album_name` VARCHAR(45) NOT NULL,
  `str_srv_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`album_name`, `str_srv_name`));
```

POPULATING TABLES:

```
INSERT INTO `dbms_project`.`album` (`album_name`, `no_of_songs`, `running_length`, `album_genre`, `m&m_artist_name`) VALUES ('malibu', '12', '56', 'r&b', 'sam');

INSERT INTO `dbms_project`.`album` (`album_name`, `no_of_songs`, `running_length`, `album_genre`, `m&m_artist_name`) VALUES ('freenats', '13', '106', 'r&b', 'bill');

INSERT INTO `dbms_project`.`album` (`album_name`, `no_of_songs`, `running_length`, `album_genre`, `m&m_artist_name`) VALUES ('innebloom', '7', '34', 'electronic', 'nick');

INSERT INTO `dbms_project`.`album` (`album_name`, `no_of_songs`, `running_length`, `album_genre`, `m&m_artist_name`) VALUES ('oasis', '32', '143', 'rock', 'gur');

INSERT INTO `dbms_project`.`album` (`album_name`, `no_of_songs`, `running_length`, `album_genre`, `m&m_artist_name`) VALUES ('ventura ', '10', '45', 'r&b', 'sam');

INSERT INTO `dbms_project`.`album` (`album_name`, `no_of_songs`, `running_length`, `album_genre`, `m&m_artist_name`) VALUES ('compton', '20', '67', 'hip-hop', 'dre');

INSERT INTO `dbms_project`.`album` (`album_name`, `no_of_songs`, `running_length`, `album_genre`, `m&m_artist_name`) VALUES ('detox', '11', '55', 'hip-hop', 'dre');


INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`) VALUES ('bam', 'gunn', '346', 'paak');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('shiiii', 'roc', '311', 'gibbs', 'alchemist', 'oasis');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('cool', 'jay', '241', 'gibbs', 'hb', 'oasis');

INSERT INTO `dbms_project`.`song` (`song_name`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('hugo', '256', 'paak', 'doc', 'innebloom');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('ny', 'fredd', '275', 'west', 'doc', 'innebloom');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('louis', 'butch', '211', 'kdot', 'mb', 'detox');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('polo', 'kim', '123', 'kdot', 'hb', 'ye');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('taste', 'krs', '234', 'sir', 'alchemist', 'ye');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('gold', 'vinesh', '121', 'paak', 'mb', 'mailbu');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('hope', '', '345', 'sir', 'doc', 'malibu');
```

```
INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('flashy', 'masterp', '322', 'west', 'doc', 'ye');

INSERT INTO `dbms_project`.`song` (`song_name`, `feature`, `song_length`, `artist_name`, `prod_name`, `album_name`) VALUES ('mac', '', '420', 'west', 'alchemist', 'innebloom');

UPDATE `dbms_project`.`song` SET `prod_name` = 'alchemist', `album_name` = 'ye' WHERE (`song_name` = 'bam');



INSERT INTO `dbms_project`.`producer` (`prod_name`, `label_name`) VALUES ('alchemist', 'warner');

INSERT INTO `dbms_project`.`producer` (`prod_name`, `label_name`) VALUES ('hb', 'warner');

INSERT INTO `dbms_project`.`producer` (`prod_name`, `label_name`) VALUES ('mb', 'griselda');

INSERT INTO `dbms_project`.`producer` (`prod_name`, `label_name`) VALUES ('doc', 'aftermath');

INSERT INTO `dbms_project`.`producer` (`prod_name`, `label_name`) VALUES ('kanye', 'griselda');

INSERT INTO `dbms_project`.`producer` (`prod_name`, `label_name`) VALUES ('pharrell', 'ugm');


INSERT INTO `dbms_project`.`record_label` (`Label_name`, `license_no`, `President`, `Headquarters/location`) VALUES ('ugm', '34556', 'tony', 'new jersey');

INSERT INTO `dbms_project`.`record_label` (`Label_name`, `license_no`, `President`, `Headquarters/location`) VALUES ('griselda', '24886', 'paulie', 'new york');

INSERT INTO `dbms_project`.`record_label` (`Label_name`, `license_no`, `President`, `Headquarters/location`) VALUES ('aftermath', '34609', 'chrissy', 'berlin');

INSERT INTO `dbms_project`.`record_label` (`Label_name`, `license_no`, `President`, `Headquarters/location`) VALUES ('warner', '88923', 'junior', 'los angeles');

INSERT INTO `dbms_project`.`record_label` (`Label_name`, `license_no`, `President`, `Headquarters/location`) VALUES ('massappeal', '45629', 'bobby', 'bronx');


INSERT INTO `dbms_project`.`artist` (`Artist_name`, `Genre`, `label_name`) VALUES ('paak', 'r&b', 'warner');

INSERT INTO `dbms_project`.`artist` (`Artist_name`, `Genre`, `label_name`) VALUES ('gibbs', 'hip-hop', 'ugm');

INSERT INTO `dbms_project`.`artist` (`Artist_name`, `Genre`, `label_name`) VALUES ('kdot', 'hip-hop', 'ugm');

INSERT INTO `dbms_project`.`artist` (`Artist_name`, `Genre`, `label_name`) VALUES ('west', 'rock', 'aftermath');

INSERT INTO `dbms_project`.`artist` (`Artist_name`, `Genre`, `label_name`) VALUES ('sir', 'house', 'griselda');
```

```
INSERT INTO `dbms_project`.`artist` (`Artist_name`, `Genre`, `label_name`) VALUES ('nas', 'coke-rap', 'griselda');


INSERT INTO `dbms_project`.`streaming_service` (`str_srv_name`) VALUES ('spotify');

INSERT INTO `dbms_project`.`streaming_service` (`str_srv_name`) VALUES ('apple music');

INSERT INTO `dbms_project`.`streaming_service` (`str_srv_name`) VALUES ('pandora');

INSERT INTO `dbms_project`.`streaming_service` (`str_srv_name`) VALUES ('bandcamp');

INSERT INTO `dbms_project`.`streaming_service` (`str_srv_name`) VALUES ('soundcloud');


INSERT INTO `dbms_project`.`distributor` (`distributor_name`, `format`, `website`, `label_name`) VALUES ('complex', 'flac', 'complex.com', 'ugm');

INSERT INTO `dbms_project`.`distributor` (`distributor_name`, `format`, `website`, `label_name`) VALUES ('npr', 'alac', 'nprmusic.com', 'ugm');

INSERT INTO `dbms_project`.`distributor` (`distributor_name`, `format`, `website`, `label_name`) VALUES ('sopranos', 'alac', 'sopranos.to', 'griselda');

INSERT INTO `dbms_project`.`distributor` (`distributor_name`, `format`, `website`, `label_name`) VALUES ('dimeo', 'wav', 'dimeo.com', 'warner');

INSERT INTO `dbms_project`.`distributor` (`distributor_name`, `format`, `website`, `label_name`) VALUES ('shady', 'wav', 'shadyrecords.to', 'griselda');

INSERT INTO `dbms_project`.`distributor` (`distributor_name`, `format`, `website`, `label_name`) VALUES ('massappeal', 'wav', 'ma.com', 'griselda');

INSERT INTO `dbms_project`.`distributor` (`distributor_name`, `format`, `website`, `label_name`) VALUES ('stonesthrow', 'flac', 'stones.com', 'warner');


INSERT INTO `dbms_project`.`mixing&mastering_artist` (`m&m_artist_name`, `softwares`, `label_name`) VALUES ('sam', 'fruityloops', 'griselda');

INSERT INTO `dbms_project`.`mixing&mastering_artist` (`m&m_artist_name`, `softwares`, `label_name`) VALUES ('nick', 'cubase', 'griselda');

INSERT INTO `dbms_project`.`mixing&mastering_artist` (`m&m_artist_name`, `softwares`, `label_name`) VALUES ('gur', 'cubase', 'ugm');

INSERT INTO `dbms_project`.`mixing&mastering_artist` (`m&m_artist_name`, `softwares`, `label_name`) VALUES ('dre', 'yamaha', 'ugm');

INSERT INTO `dbms_project`.`mixing&mastering_artist` (`m&m_artist_name`, `softwares`, `label_name`) VALUES ('bill', 'steinberg', 'warner');

INSERT INTO `dbms_project`.`distributes` (`distributor_name`, `album_name`) VALUES ('complex', 'malibu');
```

INSERT INTO `dbms_project`.`distributes` (`distributor_name`, `album_name`) VALUES ('sopranos', 'detox');

INSERT INTO `dbms_project`.`distributes` (`distributor_name`, `album_name`) VALUES ('npr', 'ye');

INSERT INTO `dbms_project`.`distributes` (`distributor_name`, `album_name`) VALUES ('npr', 'ventura');

INSERT INTO `dbms_project`.`distributes` (`distributor_name`, `album_name`) VALUES ('complex', 'oasis');

FOREIGN KEY

ALTER TABLE `dbms_project`.`artist`

CHANGE COLUMN `Artist_name` `Artist_name` CHAR(45) NOT NULL ,

ADD INDEX `label_name_idx` (`label_name` ASC) VISIBLE;

alter table producer add foreign key(label_name) references record_label(label_name);

alter table artist add foreign key(label_name) references record_label(label_name);

alter table song add constraint fk1 FOREIGN KEY(prod_name) REFERENCES producer(prod_name);

alter table album add foreign key(mnm_artist_name) references mixingnmastering_artist(mnm_artist_name);

alter table mixingnmastering_artist add foreign key(label_name) references record_label(label_name);

alter table distributor add foreign key(label_name) references record_label(label_name) ON DELETE CASCADE;

alter table distributes add foreign key(distributor_name) references distributor(distributor_name) ON DELETE CASCADE;

alter table distributes add foreign key(album_name) references album(album_name);

alter table streams add foreign key(str_srv_name) references streaming_service(str_srv_name);

alter table streams add foreign key(album_name) references album(album_name);

Join Queries

1)Show the artist's name, the album's name and the mixing&mastering artist's name where they have the same genre.

```
> select artist.artist_name,album.album_name,album.mnm_artist_name
-> from album
-> inner join artist on album.album_genre=artist.genre;
```

```
MariaDB [dbms_project]> select artist.artist_name,album.album_name,album,mnm_artist_name

    -> from album
    -> inner join artist on album.album_genre=artist.genre;
ERROR 1054 (42S22): Unknown column 'album' in 'field list'
MariaDB [dbms_project]> select artist.artist_name,album.album_name,album.mnm_artist_name

    -> from album
    -> inner join artist on album.album_genre=artist.genre;
+-------------+------------+-----------------+
| artist_name | album_name | mnm_artist_name |
+-------------+------------+-----------------+
| gibbs       | detox      | dre             |
| kdot        | detox      | dre             |
| gibbs       | gunz       | dre             |
| kdot        | gunz       | dre             |
| paak        | malibu     | sam             |
| west        | oasis      | gur             |
| paak        | ventura    | sam             |
| paak        | ye         | bill            |
+-------------+------------+-----------------+
8 rows in set (0.003 sec)
```

2)Show the distributor names of producers who are affiliated to the same music label and the format of music they exported in.

-> select distributor.distributor_name,distributor.format,producer.prod_name

-> from distributor

-> left join producer on distributor.label_name=producer.label_name;

```
MariaDB [dbms_project]> select distributor.distributor_name,distributor.format,producer.
prod_name
    -> from distributor
    -> left join producer on distributor.label_name=producer.label_name;
+------------------+--------+-----------+
| distributor_name | format | prod_name |
+------------------+--------+-----------+
| complex          | flac   | pharrell  |
| npr              | alac   | pharrell  |
| sopranos         | alac   | kanye     |
| sopranos         | alac   | mb        |
+------------------+--------+-----------+
4 rows in set (0.001 sec)

MariaDB [dbms_project]>
```

3)Showcase the artist-producer collaborations of the same label and the genre of the music they have worked on.

->select artist.artist_name,artist.genre,producer.prod_name

-> from artist

-> right join producer on artist.label_name=producer.label_name;

```
MariaDB [dbms_project]> select artist.artist_name,artist.genre,producer.prod_name
    -> from artist
    -> right join producer on artist.label_name=producer.label_name;
+-------------+----------+-----------+
| artist_name | genre    | prod_name |
+-------------+----------+-----------+
| west        | rock     | doc       |
| nas         | coke-rap | kanye     |
| sir         | house    | kanye     |
| nas         | coke-rap | mb        |
| sir         | house    | mb        |
| gibbs       | hip-hop  | pharrell  |
| kdot        | hip-hop  | pharrell  |
| paak        | r&b      | alchemist |
| paak        | r&b      | hb        |
+-------------+----------+-----------+
9 rows in set (0.003 sec)
```

4)Show the songs and the album it is from, of an artist and the number of songs in the respective album.

->select song.song_name ,song.artist_name ,album.no_of_songs

-> from song

-> left join album on song.album_name=album.album_name;

```
MariaDB [dbms_project]> select song.song_name ,song.artist_name ,album.no_of_songs
    -> from song
    -> left join album on song.album_name=album.album_name;
+-----------+-------------+-------------+
| song_name | artist_name | no_of_songs |
+-----------+-------------+-------------+
| bam       | paak        |          32 |
| cool      | gibbs       |           7 |
| flashy    | west        |          13 |
| gold      | paak        |          32 |
| hope      | sir         |          32 |
| hugo      | paak        |          20 |
| louis     | kdot        |          13 |
| mac       | west        |          13 |
| ny        | west        |           7 |
| polo      | kdot        |          32 |
| shiiii    | gibbs       |          20 |
| taste     | sir         |          10 |
+-----------+-------------+-------------+
12 rows in set (0.000 sec)
```

AGGREGATE FUNCTIONS:

Show the number of songs with song length greater than 300 seconds

->select count(*)

-> from song

-> where song_length>300;

```
MariaDB [dbms_project]> select count(*)
    -> from song
    -> where song_length>300;
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.002 sec)
```

Show the total no_of_song's length of the artist name west

MariaDB [dbms_project]> select sum(song_length)

-> from song

-> where artist_name='west';

```
MariaDB [dbms_project]> select sum(song_length)
    -> from song
    -> where artist_name='west';
+------------------+
| sum(song_length) |
+------------------+
|             1017 |
+------------------+
1 row in set (0.010 sec)
```

Show the average number of songs in an R&B genre album

->select avg(no_of_songs)

-> from album

-> where album_genre='r&b';

```
MariaDB [dbms_project]> select avg(no_of_songs)
    -> from album
    -> where album_genre='r&b';
+------------------+
| avg(no_of_songs) |
+------------------+
|          11.6667 |
+------------------+
1 row in set (0.000 sec)
```

There are 12 songs, on an average ,for the R&B genre.

SET OPERATIONS:

Show all the artist's genres of the albums they have worked on

SELECT ALBUM_GENRE,artist_name FROM ALBUM

-> UNION

-> SELECT GENRE,artist_name from artists;

```
MariaDB [dbms_project]> SELECT ALBUM_GENRE,artist_name FROM ALBUM
    -> UNION
    -> SELECT GENRE,artist_name FROM ARTIST;
+-------------+-------------+
| ALBUM_GENRE | artist_name |
+-------------+-------------+
| NULL        |             |
| hip-hop     | paak        |
| hip-hop     | gibbs       |
| electronic  | paak        |
| r&b         | sir         |
| rock        | kdot        |
| r&b         | west        |
| r&b         | gibbs       |
| hip-hop     | kdot        |
| coke-rap    | nas         |
| r&b         | paak        |
| house       | sir         |
| rock        | west        |
+-------------+-------------+
```

2) Show the common albums in Volume1(table name: album) and Volume2(Table name:album2)

select * from album

-> intersect

-> select * from album2;

```
MariaDB [dbms_project]> select * from album
    -> intersect
    -> select * from album2;
+------------+-------------+----------------+-------------+-----------------+-------------+
| album_name | no_of_songs | running_length | album_genre | mnm_artist_name | artist_name |
+------------+-------------+----------------+-------------+-----------------+-------------+
| gunz       |          11 |             55 | hip-hop     | dre             | gibbs       |
| malibu     |          12 |             56 | r&b         | sam             | sir         |
| ventura    |          10 |             45 | r&b         | sam             | west        |
| ye         |          13 |            106 | r&b         | bill            | gibbs       |
+------------+-------------+----------------+-------------+-----------------+-------------+
4 rows in set (0.000 sec)
```

3)List out the albums that are in volume2 and volume1 wit the duplicates

MariaDB [dbms_project]> select * from album2

-> union all

-> select * from album;

```
MariaDB [dbms_project]> select * from album2
    -> union all
    -> select * from album;
+------------+-------------+----------------+-------------+-----------------+-------------+
| album_name | no_of_songs | running_length | album_genre | mnm_artist_name | artist_name |
+------------+-------------+----------------+-------------+-----------------+-------------+
| chronic    |          31 |             35 | gfunk       | dre             | sir         |
| freench    |          34 |             54 | hip-hop     | dre             | gibbs       |
| girl       |          12 |             56 | r&b         | sam             | sir         |
| graduation |          54 |             55 | electronic  | nick            | west        |
| gunz       |          11 |             55 | hip-hop     | dre             | gibbs       |
| malibu     |          12 |             56 | r&b         | sam             | sir         |
| quartz     |          44 |             67 | trance      | bill            | gibbs       |
| ventura    |          10 |             45 | r&b         | sam             | west        |
| ye         |          13 |            106 | r&b         | bill            | gibbs       |
|            |        NULL |           NULL | NULL        | NULL            |             |
| detox      |          20 |             67 | hip-hop     | dre             | paak        |
| gunz       |          11 |             55 | hip-hop     | dre             | gibbs       |
| innebloom  |           7 |             34 | electronic  | nick            | paak        |
| malibu     |          12 |             56 | r&b         | sam             | sir         |
| oasis      |          32 |            143 | rock        | gur             | kdot        |
| ventura    |          10 |             45 | r&b         | sam             | west        |
| ye         |          13 |            106 | r&b         | bill            | gibbs       |
+------------+-------------+----------------+-------------+-----------------+-------------+
```

STORED FUNCTION:

Create a stored function for classifying a song as “short”,”medium” and “long”:

CREATE FUNCTION howlong(song_length int)

-> returns varchar(20)

-> DETERMINISTIC

-> BEGIN

-> DECLARE thislong VARCHAR(20);

-> IF SONG_LENGTH>300 THEN

-> SET thislong='long';

-> ELSEIF (song_length<=300 AND song_length>=200) THEN

-> SET thislong='medium';

-> elseif song_length<200 THEN

-> SET thislong='short';

-> END IF;

-> RETURN (thislong);

-> END$$

```
MariaDB [(none)]> use dbms_project
Database changed
MariaDB [dbms_project]> select song_name,howlong(song_length)
    -> from song order by song_length;
+-----------+----------------------+
| song_name | howlong(song_length) |
+-----------+----------------------+
| vinersh   | short                |
| gold      | short                |
| polo      | short                |
| louis     | medium               |
| taste     | medium               |
| cool      | medium               |
| hugo      | medium               |
| ny        | medium               |
| shiiii    | long                 |
| flashy    | long                 |
| hope      | long                 |
| bam       | long                 |
| mac       | long                 |
+-----------+----------------------+
13 rows in set (0.006 sec)
```

STORED PROCEDURE:

Display the maximum number of songs in a volume of albums

DELIMITER &&

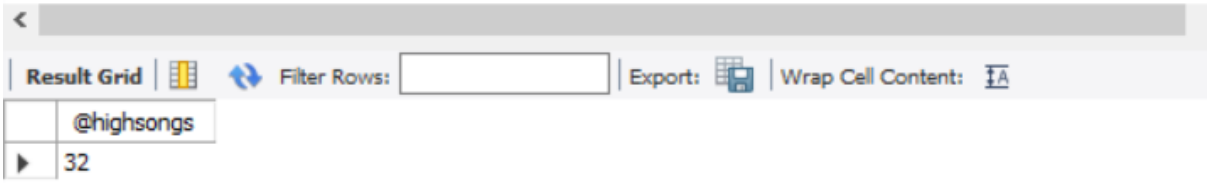
MariaDB [dbms_project]> CREATE PROCEDURE display_max_no_of_songs (OUT highsongs INT)

-> BEGIN

-> SELECT MAX(no_of_songs) INTO highsongs FROM album;

-> END &&

```
set @highsongs = 0;
call dbms_project.display_max_no_of_songs(@highsongs);
select @highsongs;
```



TRIGGER

Increment the number of songs field in album after a song has been added to the song table referencing to an album

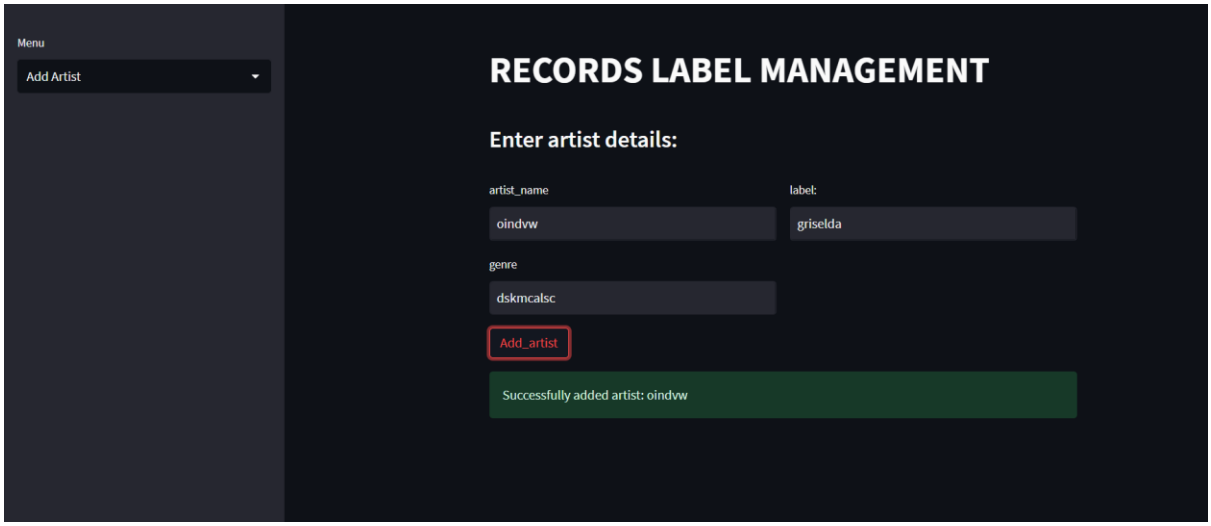
```
DROP TRIGGER IF EXISTS `dbms_project`.`song_AFTER_INSERT`;


DELIMITER $$

USE `dbms_project`$$

CREATE DEFINER = CURRENT_USER TRIGGER `dbms_project`.`song_AFTER_INSERT` AFTER INSERT
ON `song` FOR EACH ROW

BEGIN

  update album

  set no_of_songs=no_of_songs+1

  where album_name=NEW.album_name;

END$$

DELIMITER ;
```

SIMPLE FRONTEND IMPLEMENTATION

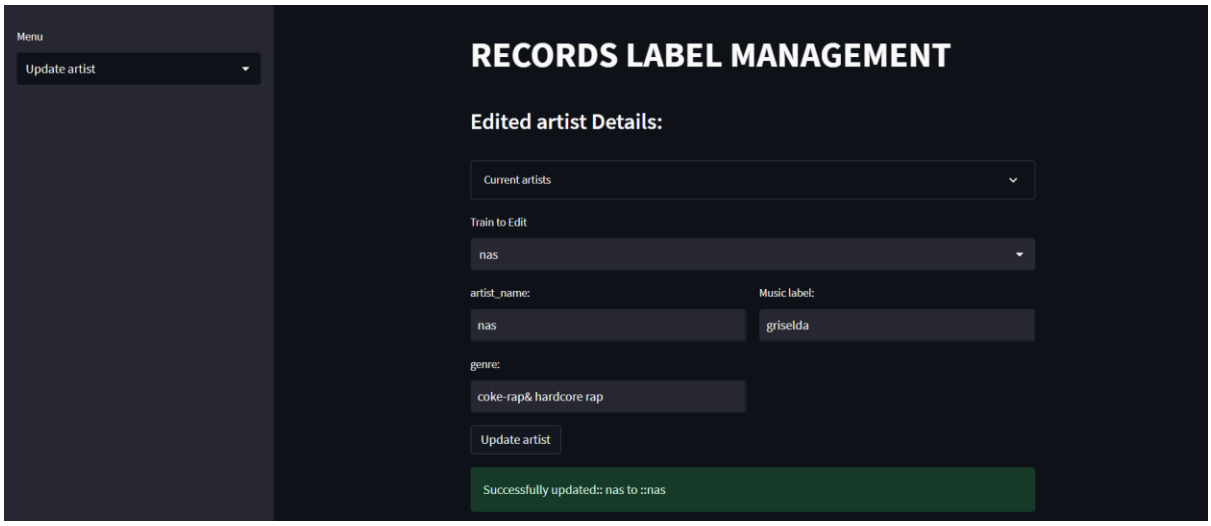
Streamlit has been used to display the various operations on the tables of the database

Provided below are the screenshots of the requirements of the project:

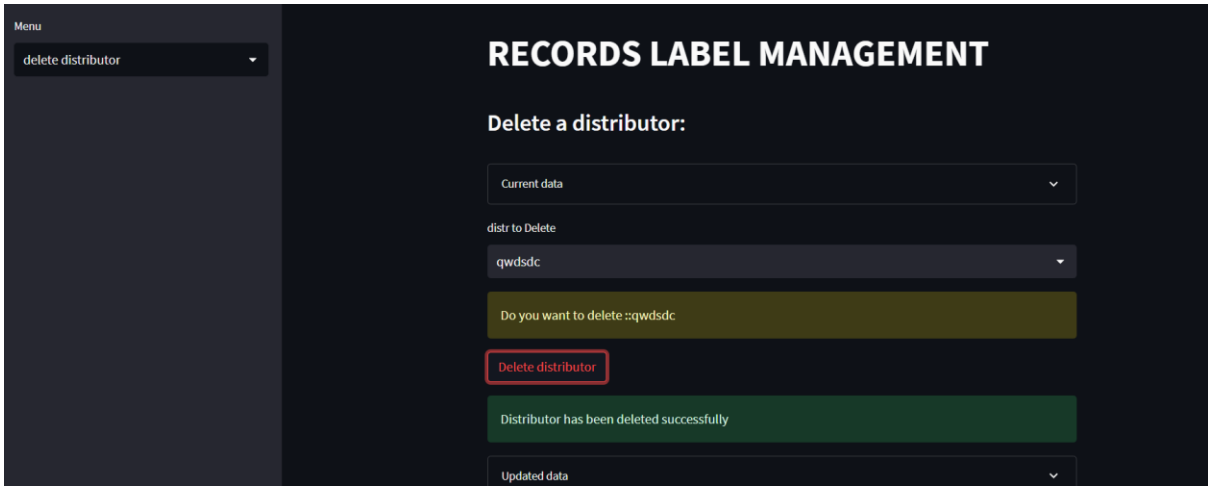
Addition of a tuple:



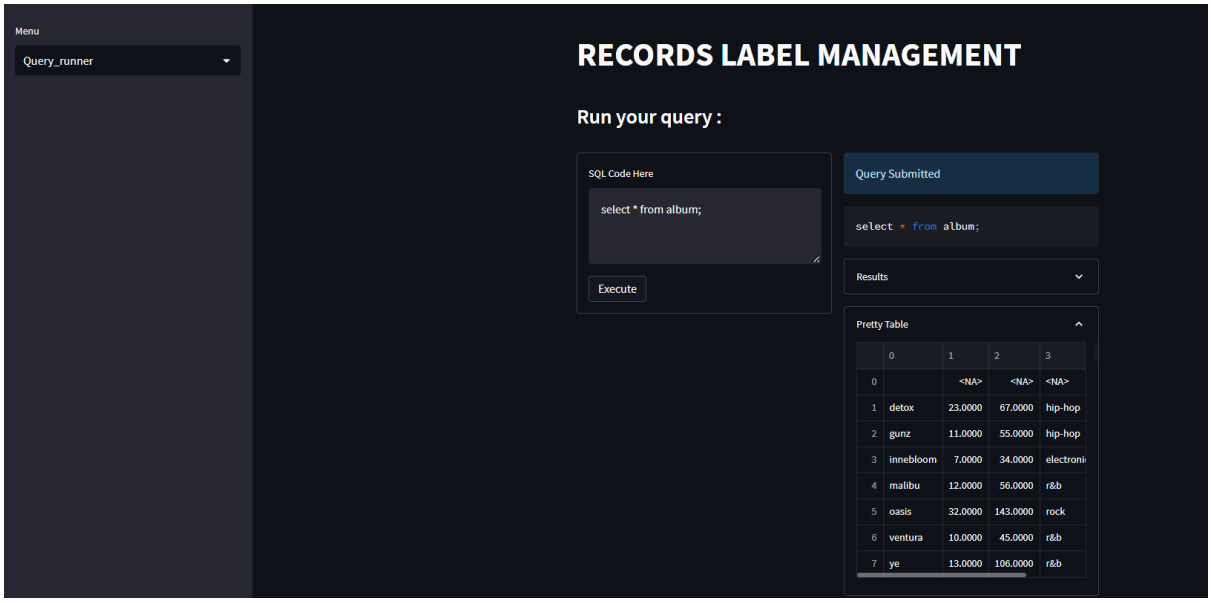
Updation of a tuple :



Deletion of a tuple:



Query runner:



MODIFICATION:

Create a stored procedure for showing the number of distributors an album has from the distributes table.

CREATE DEFINER=`root`@`localhost` PROCEDURE `distribute_no`()

BEGIN

select album_name, count(*) as num_distributors from distributes group by album_name;

END

```
1 •  call dbms_project.distribute_no();
2
```

Result Grid | Filter Rows: | Export:

album_name	num_distributors
detox	1
innebloom	1
malibu	2
oasis	1
ventura	1
ye	2