**INFO 5707**

**Group 12 Data Modelling for Information Professionals**

**Spring 2022**

IMDB SONG DATABASE BY:

PROJECT LEADER SADIA MAHBUB (10971223)

SAI SRIJA MUPPA

VAISHNAVI SIMHACHALAM(11525465)

# Project Overview:

The Music Database, which stands for Internet Music Database, is a database widely used by YouTube, Music, and other online sites. This database stores all data about audio and video songs and some music songs. The purpose and purpose of creating this database is to record detailed information about songs and series and make it available to users. This database allows registered users to submit new content and edit existing entries. The major focus of IMDB Song and music Database is to make content like live streams, music, and song along with summaries, themes, ratings, and reviews available for a viewer to watch. Using this database, a viewer can also gain information about the lyricist, singer, musician, composer, and actors who contributed to the song. And This database even suggests songs, music, series based on a viewer’s previous choice of content.

# Project Objectives:

This database is currently available for users like students and professors enrolled in universities. And users must have access to the university library to access this database. It is a platform that helps users find appropriate songs to consider their interests with information such as symbols, directors, prices, gender, language, distribution, and studio. Users are said to be interested in many different songs, including classical music, romantic songs, songs in any language, famous songs, songs and more. This database is a great resource for Google looking for people who want to see the search songs, because users use students and teachers using data recovery systems to choose specific songs for projects and their competitions like Comic-Con competitions and theaters. And since this database is accessible to everyone at the university, many users use it to hang out.

# Project Scope:

The database being created contains a list of song names such as romantic songs, classical songs, movie songs, international songs, song titles and categories. Main song, song ID. It includes song description, song audience, musician description, cost, composer, budget and more. All of the above information is used to develop the database in an organized way to make it easier for users to use and track research. Through this database, viewers can also get information about the singers, directors and producers who contributed to the songs.

# User Requirements:

**Singer:** Singer is the most important field because it helps the user to have a complete list of songs. No song is complete without a singer, so the singer is the most important person in creating the song. It makes it possible for ordinary users to watch the songs of their favorite singers.

**Language:** Language is a very important part of this database. Most users want to hear a language that understands audio or video songs. This makes the song a more immersive experience. This database contains a list of all languages in which the song resides.

**Rating:** Song writer: When a song is released to an audience, it is art, critiqued and appreciated by the audience. It helps the user to select a good song from the playlist.

**Song Titles:** Each singer releases an album with a name which could be anything, like what the song means or something that is in the song. (Exclaves story, written by Taylor Swift)

**Cost:** Cost is of 2 types; A. The amount used to make the song

B. The amount collected for the album.

**Studio:** It is the place where the singer has signed a contract to share her songs for a price. Ex: YG Entertainments

**Producer:** For making a song, an amount is spend and lot of production work is done for which a producer (“person”) he sponsors whatever is needed with a budget to get the outcome.

# Business Rules:

A song can have Multiple Singers, A Singer can work in many songs.

A song can have only single rating, A rating can be given to many songs.

A song can win zero or many price, A price can be given to one song.

A song can be made in multiple languages, A language can have many songs.

A song can have single set of cost, A cost can work for many songs.

A song can be produced by only one studio, A studio can work on one or many songs.

A singer can sing many songs,

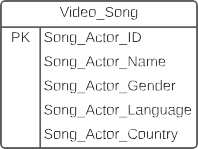
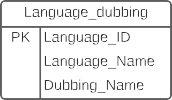
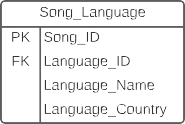
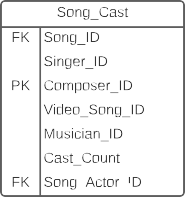
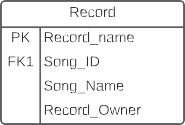
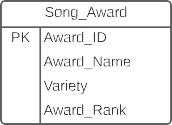
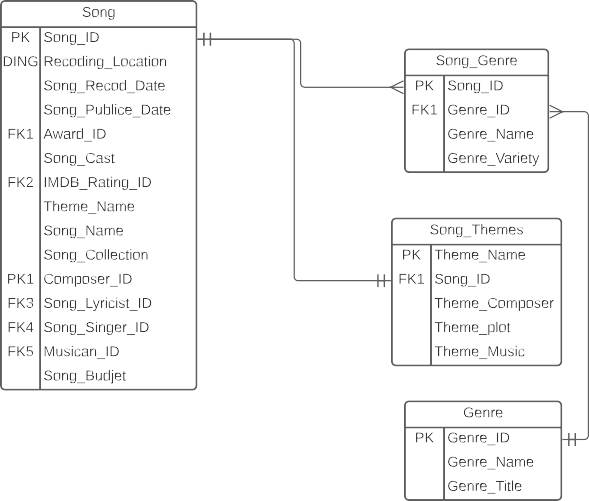
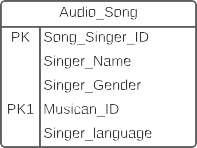
Multiple singers can sing multiple songs.

A song can have one or more titles, A title can have more songs.

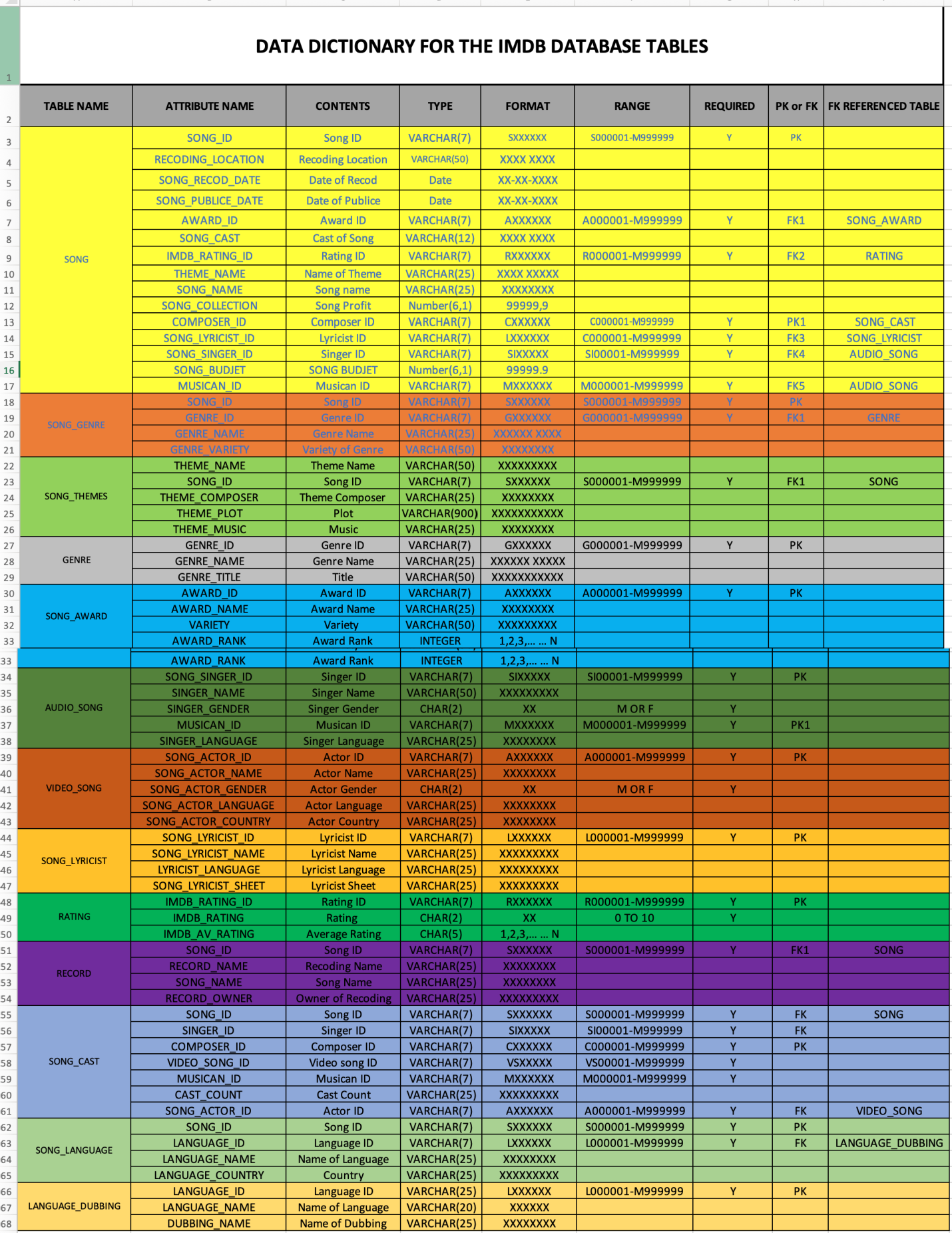
A song can be written one Lyricist,

A Lyricist can have one or more song

**ER Diagram:**



# Data Dictionary:



**PK:** Primary Key

**FK:** Forigen Key

**VARCHAR:** Variable character length data (1-1000 characters)

**CHAR:** Fixed character length data (1-200 characters)

**NUMBER:** Numeric data, integer number and numeric data we have 8 digits and decimal point**. DATE:** DATE is explained here month, years, and date its format MM-DD-YYYY

Here MM is month, DD is day and YYYY is years.

**Table Creation and Data Entry:**

**Table1:** songGenre

**Table Creation:**

**c**reate table songGenre(SNo Integer,songID integer,genreId integer, genreTypes varchar(25));

**Data Entry:**

insert into songGenre(SNo, songID,genreId,genreTypes)values(1,101,2,'comedy song’);

insert into songGenre(SNo, songID,genreId,genreTypes)values(2,102,4,'classical');

insert into songGenre(SNo, songID,genreId,genreTypes)values(3,103,2,'remix song');

insert into songGenre(SNo, songID,genreId,genreTypes)values(4,104,2,'sad song');

insert into songGenre(SNo, songID,genreId,genreTypes)values(5,105,6,'romantic song');

insert into songGenre(SNo, songID,genreId,genreTypes)values(6,106,2,'folk song');

**Displaying data:**

select \* from songGenre

**Output:**

**Table2:** Genre

**Table Creation:**

create table genre(genreID Integer, GenreName varchar(15));

**Data Entry:**

insert into genre(genreID,GenreName) values (1,'Comedy song');

insert into genre(genreID,GenreName) values (2,' classical songs ');

insert into genre(genreID,GenreName) values (3,'folk music');

insert into genre(genreID,GenreName) values (4,'sad song');

insert into genre(genreID,GenreName) values (5,'romantic song');

insert into genre(genreID,GenreName) values (6,'move song'');

insert into genre(genreID,GenreName) values (7, 'remix song');

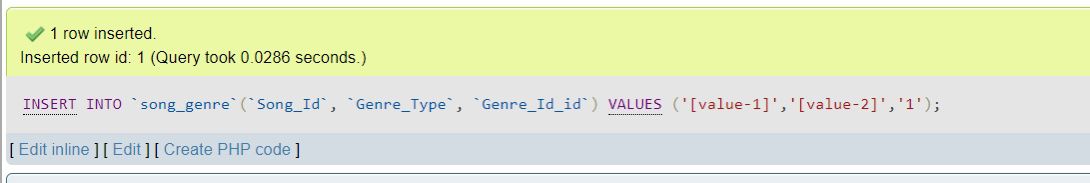
**Displaying data:**

select \* from genre;

Graphical user interface, application

Description automatically generated

INSERT INTO song\_genre(‘Song\_Id’, ‘Genre\_Type’, ‘Genre\_Id\_id’) VALUES (‘[value-1]’,’[value-2]’,’1);



SELECT \* FROM song\_genre

Graphical user interface, application

Description automatically generated

SELECT \* FROM song\_genre

Graphical user interface, text, application, email

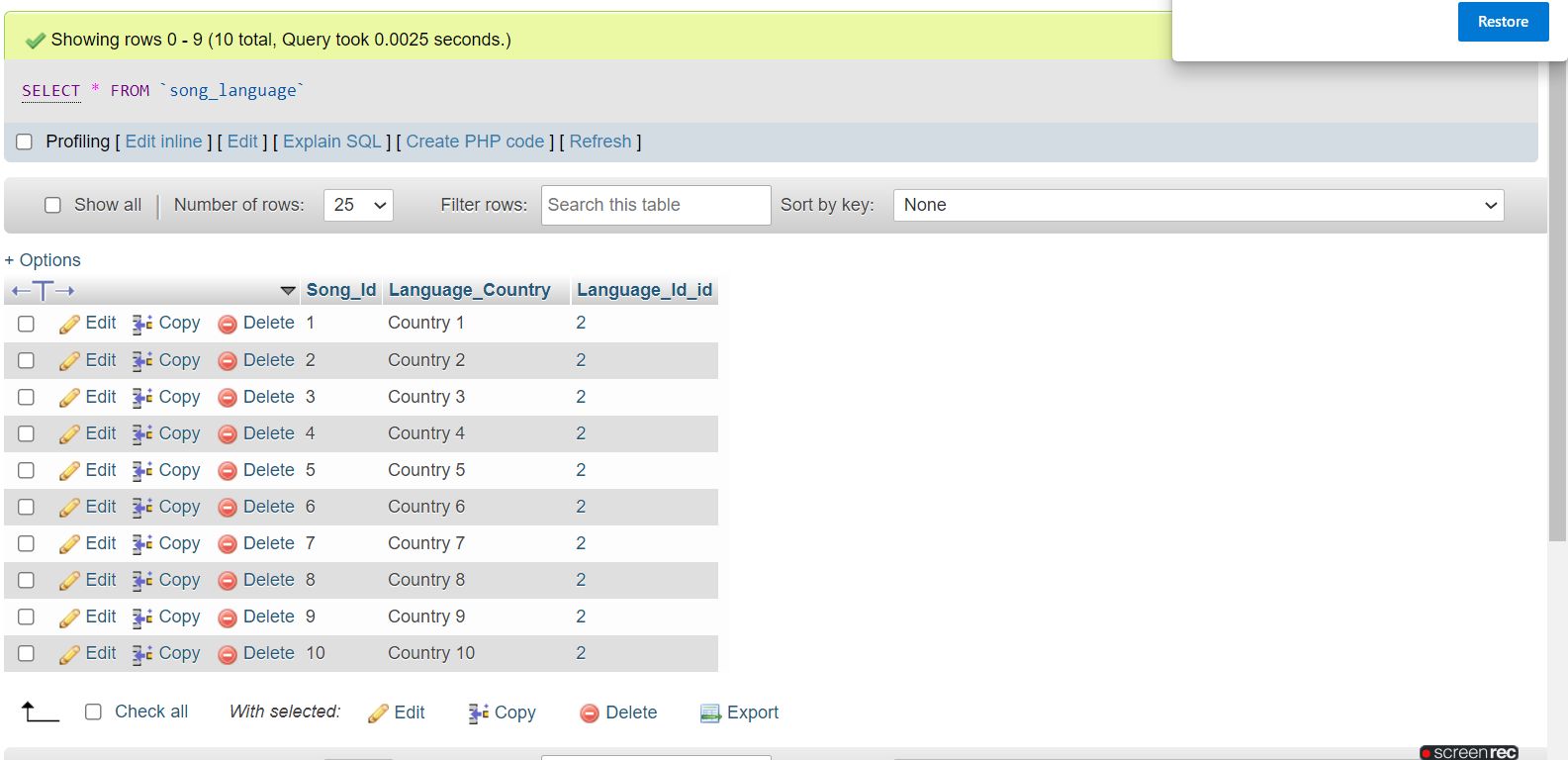
Description automatically generated

SELECT \* FROM song\_cast

Graphical user interface, application

Description automatically generated

SELECT \* FROM song\_language



SELECT \* FROM ratings

Graphical user interface, application

Description automatically generated

SELECT \* FROM genre

Graphical user interface, text, application

Description automatically generated

UPDATE genre SET genre\_Id=1, Genre\_Name=’Singing’ WHERE 1;

Graphical user interface, text, application

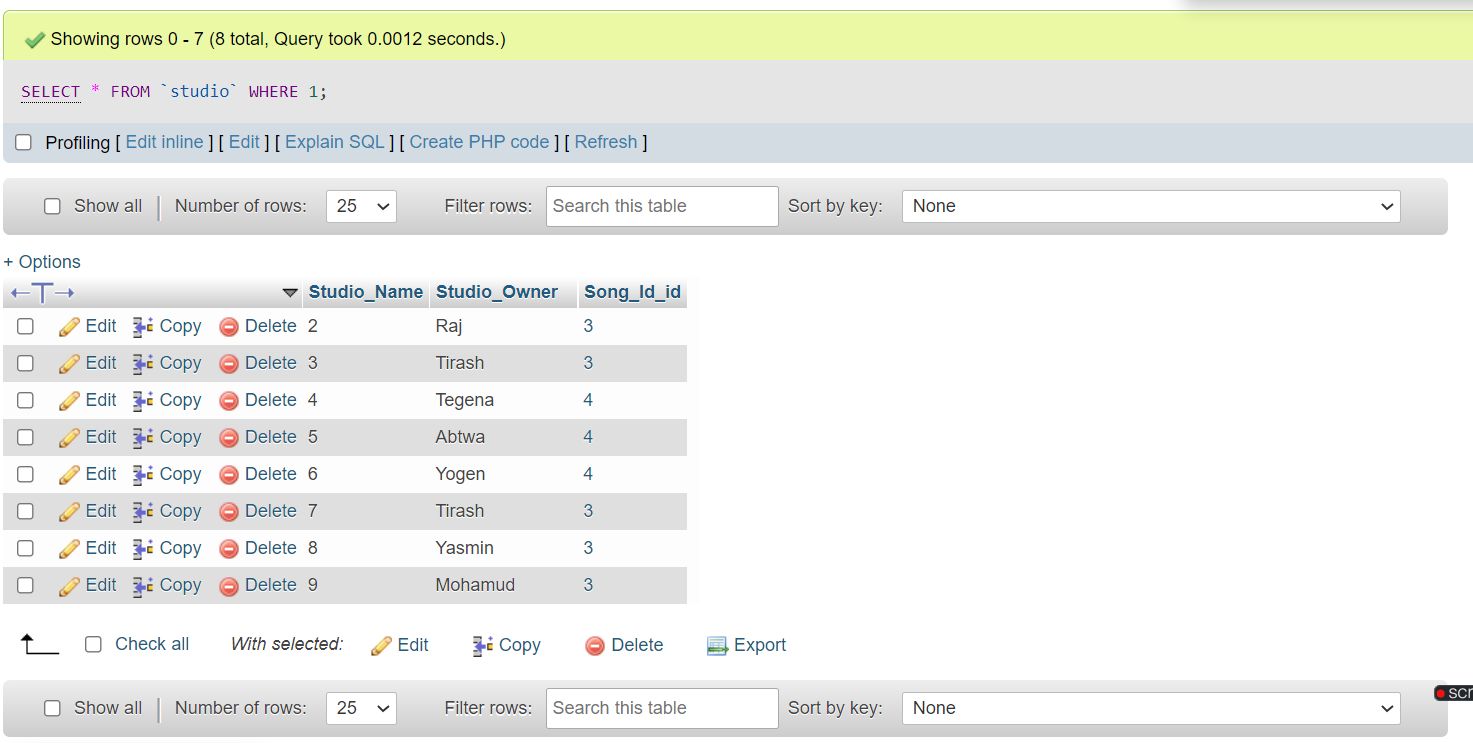
Description automatically generated

SELECT \* FROM studio

Graphical user interface, application

Description automatically generated

SELECT \*FROM studio WHERE 1;



UPDATE awards SET Award\_Id=’1’, Category=’Updated Award’ WHERE 1;



We can see all the genre and ratings, song\_cast, song\_genre, song\_language, Studio columns

Graphical user interface, application

Description automatically generated

INSERT INTO genre(‘Genre\_Id, ‘Genre\_Name’) VALUES (‘1’ , ‘Song of The songs’);

Inserting the genre with id and name in to the song of songs

Graphical user interface, text, application, chat or text message, website

Description automatically generated

SELCET \* FROM song\_genre

Graphical user interface, text, application, email

Description automatically generated

SELECT \* FROM ‘genre’ WHERE 1;

Selecting the genre with the value

Graphical user interface, text, application

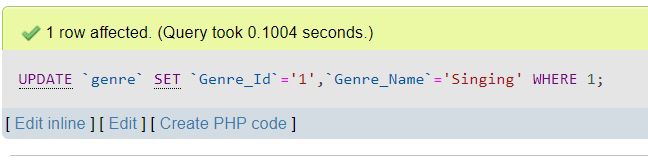
Description automatically generated

INSERT INTO ‘awards’( ‘Award\_Id’, ‘Category’) VALUES (‘1’,’First Category’);

Graphical user interface, application, website

Description automatically generated

UPDATE ‘genre’ SET ‘Genre\_Id’=’1’, ‘Genre\_Name’=’Singing ‘ WHERE 1;



INSERT INTO song\_genre (‘Song\_Id’, ‘Genre\_Type’, ‘Genre\_Id’) VALUES (‘[value-1]’,[value-2]’,’1’);

Graphical user interface, text, application

Description automatically generated

INSERT INTO ‘song’(‘Song\_Id’ , ‘Title’, ‘shootng\_location’, ‘Release\_date’, ‘IMDB\_Votes’, ‘IMDBRating\_Id\_id’) VALUES (‘[values-1]’, ‘[values-2]’,’[value-3]’,’[values-4]’,’[value-5]’,’1’,’1’);

Graphical user interface, application, website

Description automatically generated

SELECT \* FROM song WHERE 1;

Graphical user interface, text, application

Description automatically generated