**DBMS LAB**

BSCS *[4-A]*

**FINAL TERM PROJECT REPORT**



**Submitted By:**

ALI EHTISHAM

**Submitted to:**

**Lab In-charge:** Sir Shoaib Khan

**Course Teacher:** Ms. Sadia Aslam

**Movie Management System (Netflix)**

**Abstract:**

This report describes the design and implementation of a database for a streaming service (Netflix). The database stores information about users, media (e.g. movies, TV shows, documentaries etc.), and the relationships between them (e.g. which users have watched or downloaded which media, cast of the actor in a particular movie, tv show, documentaries and webs series , directors that have directed the movies). The database schema consists of several tables linked together through foreign keys, and is designed to be flexible and scalable to accommodate future growth.

**Introduction:**

Streaming services have become increasingly popular in recent years, with increased users subscribing to access a wide variety of media content. As the volume of data grows, it is important to have a robust and efficient database to store and manage the data. This project has been designed for this very purpose. The feature of the project resonate with those of the popular movie streaming network knows as Netflix.

**Problem Statement:**

The main challenge in designing the database was to handle the complexity and variety of data involved in a streaming service. The data includes information about users (e.g. username, email, payment type), media (e.g. name, genre, language), and the relationships between them (e.g. which users have watched or downloaded which media). Additionally, the database needed to be flexible enough to handle different types of media (e.g., movies, TV shows, documentaries) and support various queries and filters to allow users to browse and search for content.

**Objectives:**

The main objectives of this project were to:

* Design a database schema that can store and manage the data for a streaming service.
* Implement the database schema using a relational database management system (RDBMS).
* Test the database to ensure it is functional and efficient.
* Document the design and implementation of the database for future reference.

**Features:**

**Media metadata:** The database includes several tables that store information about media, such as the name, genre, language, and rating. This allows users to browse and search for media by these attributes.

**Movie/Shows metadata:** The relation between the tables of the movie/shows, actors, directors and their cast table allows you to search for acting actors in the movies/shows and their directors using only movie/show names

**User history:** The database includes several tables that store information about which users have watched or downloaded which media. This allows users to track their viewing history and resume watching content where they left off.

**Scalability:** The database was designed to be scalable to accommodate future growth in the number of users and media. This was achieved by using appropriate data types and indexing strategies, as well as implementing foreign keys to link the various tables together.

**Data integrity**: Foreign keys were used to enforce data integrity and prevent inconsistencies in the data. For example, a foreign key was used to link the "media" table to the "languages" table, ensuring that only valid language names can be stored in the "media" table same relations have been implement of other tables and their attributes

**Performance:** The database was optimized for performance through the use of indexes and other techniques to ensure fast query times. This is important for a streaming service, as users expect to be able to browse and search for media quickly and smoothly.

**In addition to the features mentioned above:**

* The database also has a C# .NET based Windows Form front-end. The front-end allows users to interact with the database through a user-friendly graphical interface. It provides various features such as the ability to search for and browse media and update user information.
* In total there are 5 forms pages to this front-end a main page for movies and individual pages for tv shows , documentaries and web series and a page for top picks .
* Each page allows you to insert, updated, delete, search and view data of the respective categories and search the actors and directors of respective categories my providing the movie/show names
* The frontend communicates with the database through a database access layer that implements various functions to retrieve and update data. This separation of concerns helps to improve the maintainability and scalability of the system.
* There is a top-ranking feature which display the top 3 movie/shows/web series and documentaries ranked using the ranking attribute defined in the tables.
* Overall, the combination of the database and frontend provides a comprehensive solution for a streaming service, enabling users to easily access and interact with a wide variety of media content.

**Methodology:**

* Data collection: The first step in the data collection process was to gather the necessary data from the relevant sources. That included data scraping data from relevant websites, and manually inputting data into the database.
* Data cleaning and preprocessing: Once the data was collected, it would likely have needed to be cleaned and preprocessed in order to ensure that it was in a usable form for analysis. This involved removing duplicates, handling missing values, or standardizing the format of the data.
* Database design: The next step in the process was to design the database schema using SQL commands. This involved creating tables to store the data, and specifying the attributes (i.e., columns) that each table would contain. The SQL script provided below includes commands to create eight main tables: user\_data, actors, directors, genre, ratings, languages, movie\_type, and tvshow. It also includes commands to create four tables for each type of content (movies, TV shows, documentaries, and web series) that a user has watched or downloaded.
* Data modeling: Once the tables were created, the data would have needed to be modeled in order to establish relationships between the tables. This was done using foreign keys, which are used to link records in one table to records in another table. For example, the watch\_id attribute in the watched\_mv table is a foreign key that references the id attribute in the user\_data table. This establishes a relationship between a user and the movies they have watched.
* Data analysis: After the data was cleaned, preprocessed, and modeled, it could then be analyzed using SQL queries. These queries would allow you to extract relevant information from the database, and perform statistical analysis or generate reports.
* Data visualization: If desired, the results of the data analysis could be visualized using a tool such as a chart or graph. This could help to make the results more understandable and easier to interpret.
* Overall, the SQL script provided above outlines a clear and organized method for collecting, storing, and analyzing data in a database. By following this process, you can efficiently and effectively work with the data to gain insights and draw conclusions.

**Tables, Attributes & their relations**

* The **user\_data** table stores information about users of the media platform, including their unique ID, username, email address, date of birth, and payment type.
* The **actors table** stores information about actors, including their unique ID, name, and gender.
* The **directors table** stores information about directors, including their unique ID, name, and gender.
* The **genre table** stores information about movie genres, including their unique ID and name.
* The **ratings table** stores ratings given by users, including the user's ID (which is a foreign key to the **user\_data table**), the rating score, and the date the rating was given.
* The **languages table** stores information about languages, including their unique ID and name.
* The **movie\_type** table stores information about movies, including their unique ID, name, language (which is a foreign key to the **languages table**), rating (which is a foreign key to the **ratings table**), and genre (which is a foreign key to the **genre table**).
* The **tvshow** table stores information about TV shows, including their unique ID, name, language (which is a foreign key to the **languages table**), rating (which is a foreign key to the **ratings table**), genre (which is a foreign key to the **genre table**), and number of parts.
* The **documentaries** table stores information about documentaries, including their unique ID, name, language (which is a foreign key to the **languages table**), rating (which is a foreign key to the **ratings table**), genre (which is a foreign key to the **genre table**), and length.
* The **webseries** table stores information about web series, including their unique ID, name, language (which is a foreign key to the **languages table**), rating (which is a foreign key to the **ratings table**), genre (which is a foreign key to **the genre table**), and number of seasons.
* The **watched\_mv, watched\_tv, watched\_doc, and watched\_wb** tables store information about which movies, TV shows, documentaries, and web series a user has watched, respectively. Each of these tables includes a foreign key to the **user\_data** table (to identify the user) and the name of the media (which is a foreign key to the **movie\_type, tvshow, documentries, or webseries** table, respectively)
* The **downloads\_mv, downloads\_tv, downloads\_doc, and downloads\_wb** tables store information about which movies, TV shows, documentaries, and web series a user has downloaded, respectively. Each of these tables includes a foreign key to the user\_data table (to identify the user) and the name of the media (which is a foreign key to the **movie\_type, tvshow, documentries, or webseries table**, respectively).

**SCRIPT:**

create table user\_data

(

id int primary key ,

username varchar(30),

email varchar(50),

dateofbirth date,

payment\_type varchar(30),

);

create table actors

(

actor\_id int primary key not null,

actor\_name varchar(30) unique not null,

actor\_gender varchar(10),

);

create table directors

(

director\_id int primary key not null,

director\_name varchar(30) unique not null,

director\_gender varchar(10),

);

create table genre

(

genre\_id int primary key not null,

genre\_name varchar(20) unique not null,

);

create table ratings

(

rating\_id int primary key references user\_data(id),

rating\_score decimal unique not null,

rating\_date date,

);

create table languages

(

lannguage\_id int primary key not null,

language\_name varchar(30) unique not null,

);

create table movie\_type

(

mv\_id int primary key not null,

mv\_name varchar(30) unique not null,

mv\_language varchar(30) foreign key references languages(language\_name),

mv\_rating decimal foreign key references ratings(rating\_score),

mv\_genre varchar(20) foreign key references genre(genre\_name),

);

create table tvshow

(

tvs\_tvsid int primary key not null,

tvs\_name varchar(30) unique not null,

tvs\_language varchar(30) foreign key references languages(language\_name),

tvs\_rating decimal foreign key references ratings(rating\_score),

tvs\_genre varchar(20) foreign key references genre(genre\_name),

tvs\_parts int,

);

create table documentries

(

doc\_id int primary key not null,

doc\_name varchar(30) unique not null,

doc\_language varchar(30) foreign key references languages(language\_name),

doc\_rating decimal foreign key references ratings(rating\_score),

doc\_genre varchar(20) foreign key references genre(genre\_name),

doc\_length time,

);

create table webseries

(

wb\_id int primary key not null,

wb\_name varchar(30) unique not null,

wb\_language varchar(30) foreign key references languages(language\_name),

wb\_rating decimal foreign key references ratings(rating\_score),

wb\_genre varchar(20) foreign key references genre(genre\_name),

wb\_seasons int,

);

create table watched\_mv

(

watch\_id int foreign key references user\_data(id),

mv\_name varchar(30) foreign key references movie\_type(mv\_name),

);

create table watched\_tv

(

watch\_id int foreign key references user\_data(id),

tv\_name varchar(30) foreign key references tvshow(tvs\_name),

);

create table watched\_doc

(

watch\_id int foreign key references user\_data(id),

doc\_name varchar(30) foreign key references documentries(doc\_name),

);

create table watched\_wb

(

watch\_id int foreign key references user\_data(id),

wb\_name varchar(30) foreign key references webseries(wb\_name),

);

create table downloads\_mv

(

download\_id int foreign key references user\_data(id),

mv\_name\_d varchar(30) foreign key references movie\_type(mv\_name),

);

create table downloads\_tv

(

download\_id int foreign key references user\_data(id),

tv\_name\_d varchar(30) foreign key references tvshow(tvs\_name),

);

create table downloads\_doc

(

download\_id int foreign key references user\_data(id),

doc\_name\_d varchar(30) foreign key references documentries(doc\_name),

);

create table downloads\_wb

(

download\_id int foreign key references user\_data(id),

wb\_name\_d varchar(30) foreign key references webseries(wb\_name),

);

create table mylist\_mv

(

mylist\_id int foreign key references user\_data(id),

mv\_name\_l varchar(30) foreign key references movie\_type(mv\_name),

);

create table mylist\_tv

(

mylist\_id int foreign key references user\_data(id),

tv\_name\_l varchar(30) foreign key references tvshow(tvs\_name),

);

create table mylist\_doc

(

mylist\_id int foreign key references user\_data(id),

doc\_name\_l varchar(30) foreign key references documentries(doc\_name),

);

create table mylist\_wb

(

mylist\_id int foreign key references user\_data(id),

wb\_name\_l varchar(30) foreign key references webseries(wb\_name),

);

create table actor\_cast\_movies

(

mv\_id int foreign key references movie\_type(mv\_id),

actor\_id int foreign key references actors(actor\_id),

);

create table director\_cast\_movies

(

mv\_id int foreign key references movie\_type(mv\_id),

director\_id int foreign key references directors(director\_id),

);

create table actor\_cast\_webseries

(

wb\_id int foreign key references webseries(wb\_id),

actor\_id int foreign key references actors(actor\_id),

);

create table director\_cast\_webseries

(

wb\_id int foreign key references webseries(wb\_id),

director\_id int foreign key references directors(director\_id),

);

create table actor\_cast\_documentries

(

doc\_id int foreign key references documentries(doc\_id),

actor\_id int foreign key references actors(actor\_id),

);

create table director\_cast\_documentries

(

doc\_id int foreign key references documentries(doc\_id),

director\_id int foreign key references directors(director\_id),

);

create table actor\_cast\_tvshow

(

tvs\_id int foreign key references tvshow(tvs\_tvsid),

actor\_id int foreign key references actors(actor\_id),

);

create table director\_cast\_tvshow

(

tvs\_id int foreign key references tvshow(tvs\_tvsid),

director\_id int foreign key references directors(director\_id),

);

insert into user\_data values

(1,'ali ehtisham','aliehtisham@gmail.com','2002/3/22','visa'),

(2,'maysam hussain ','maysamhussain.com','2000/7/11','mastercard'),

(3,'huaifa asif','huzaifaasif@gmail.com','2002/9/12','visa'),

(4,'abdul haseeb','abdulhaseeb@gmail.com','2003/1/7','unionpay');

insert into actors values

(1,'tom cruise','male'),

(2,'jim carry','male'),

(3,'will smith','male'),

(4,'henry cavil','male'),

(5,'vindiesel','male'),

(6,'paulwalker','male'),

(7,'alexandra daddario','female'),

(8,'anna de armas','female'),

(9,'megan fox','female'),

(10,'sonam bajwa','female');

insert into directors values

(1,'christopher mcquarrie','male'),

(2,'jj abrams','male'),

(3,'chuck russel','male'),

(4,'francis lawrence','male'),

(5,'louis leterrier','male'),

(6,'steven caplejr','male');

insert into genre values

(1,'action'),

(2,'adventure'),

(3,'sci fi'),

(4,'horror'),

(5,'comedy');

insert into ratings values

(1,6,'2021/3/1'),

(2,7,'2021/3/11'),

(3,8,'2022/7/8'),

(4,9,'2021/5/2');

insert into languages values

(1,'english'),

(2,'japanese'),

(3,'chinese'),

(4,'urdu');

insert into movie\_type values

(1,'mission impossible','english',8,'action'),

(2,'the mask','english',8,'comedy'),

(3,'i am legend','english',7,'adventure'),

(4,'man of steel','english',8,'sci fi'),

(5,'fast and furious','english',7,'action'),

(6,'baywatch','english',6,'comedy'),

(7,'hitman','english',6,'action'),

(8,'transformers','english',8,'sci fi'),

(9,'the conjuring','japanese',6,'horror');

insert into tvshow values

(1,'peaky blinders','english',9,'adventure',6),

(2,'breaking bad','english',9,'action',5),

(3,'game of throne','english',8,'adventure',8),

(4,'witcher','english',8,'sci fi',2),

(5,'squid game','chinese',6,'adventure',1);

insert into documentries values

(1,'downfall','english',7,'adventure','2:13:1'),

(2,'found','urdu',6,'horror','2:00:00'),

(3,'free to play','english',7,'action','2:17:00'),

(4,'dont look back','english',8,'horror','1:29:00');

insert into webseries values

(1,'stranger things','english',7,'adventure',3),

(2,'dark','japanese',8,'sci fi',4),

(3,'good doctor','english',8,'adventure',7),

(4,'shameless','english',9,'adventure',11);

insert into actor\_cast\_movies values

(1,1),

(1,3),

(2,2),

(3,3),

(3,9),

(4,4),

(5,5),

(5,6),

(6,8),

(6,9),

(7,1),

(7,2),

(8,8),

(8,9),

(9,6),

(9,7);

insert into actor\_cast\_tvshow values

(1,8),

(1,1),

(2,2),

(2,3),

(2,9),

(3,2),

(3,7),

(4,8),

(5,1),

(5,3);

insert into actor\_cast\_documentries values

(1,2),

(1,4),

(1,1),

(2,5),

(2,7),

(3,8),

(4,8),

(4,9);

insert into actor\_cast\_webseries values

(1,4),

(2,2),

(2,9),

(3,1),

(3,9),

(4,1),

(4,3);

insert into director\_cast\_movies values

(1,6),

(2,5),

(2,3),

(3,1),

(3,5),

(4,2),

(5,4),

(6,3),

(7,2),

(8,4),

(9,3),

(9,1);

insert into director\_cast\_tvshow values

(1,3),

(2,2),

(3,1),

(4,6),

(4,6),

(5,1),

(5,1);

insert into director\_cast\_documentries values

(1,1),

(2,3),

(2,1),

(3,1),

(3,4),

(4,6);

insert into director\_cast\_webseries values

(1,6),

(2,2),

(3,6),

(3,1),

(3,4),

(4,3);

insert into watched\_doc values

(1,'downfall'),

(2,'found'),

(1,'free to play');

insert into watched\_mv values

(1,'the mask'),

(1,'hitman'),

(2,'transformers'),

(3,'mission impossible'),

(3,'baywatch');

insert into watched\_tv values

(3,'peaky blinders'),

(3,'witcher'),

(4,'squid game');

insert into watched\_wb values

(4,'stranger things'),

(4,'dark'),

(3,'shameless');

insert into mylist\_doc values

(3,'found'),

(3,'downfall'),

(2,'free to play');

insert into mylist\_mv values

(1,'baywatch'),

(3,'the mask'),

(4,'the mask'),

(4,'hitman');

insert into mylist\_tv values

(1,'peaky blinders'),

(1,'witcher');

insert into mylist\_wb values

(2,'stranger things'),

(1,'dark');

insert into downloads\_doc values

(1,'found'),

(3,'free to play'),

(4,'found');

insert into downloads\_mv values

(4,'hitman'),

(4,'the mask'),

(3,'i am legend');

insert into downloads\_tv values

(1,'peaky blinders');

insert into downloads\_wb values

(2,'shameless');

select top(3) \*from movie\_type order by mv\_rating desc

select top(3) \*from documentries order by doc\_rating desc

select top(3) \* from tvshow order by tvs\_rating desc

select top(3) \*from webseries order by wb\_rating desc

---JOIN QUERIES

SELECT movie\_type.mv\_name, actors.actor\_name

FROM movie\_type

INNER JOIN actor\_cast\_movies

ON movie\_type.mv\_id = actor\_cast\_movies.mv\_id

INNER JOIN actors

ON actor\_cast\_movies.actor\_id = actors.actor\_id;

SELECT movie\_type.mv\_name, actors.actor\_name

FROM movie\_type

LEFT JOIN actor\_cast\_movies

ON movie\_type.mv\_id = actor\_cast\_movies.mv\_id

LEFT JOIN actors

ON actor\_cast\_movies.actor\_id = actors.actor\_id;

SELECT movie\_type.mv\_name, actors.actor\_name

FROM movie\_type

RIGHT JOIN actor\_cast\_movies

ON movie\_type.mv\_id = actor\_cast\_movies.mv\_id

RIGHT JOIN actors

ON actor\_cast\_movies.actor\_id = actors.actor\_id;

SELECT movie\_type.mv\_name, actors.actor\_name

FROM movie\_type

FULL OUTER JOIN actor\_cast\_movies

ON movie\_type.mv\_id = actor\_cast\_movies.mv\_id

FULL OUTER JOIN actors

ON actor\_cast\_movies.actor\_id = actors.actor\_id;

SELECT tvshow.tvs\_name, actors.actor\_name

FROM tvshow

INNER JOIN actor\_cast\_tvshow

ON tvshow.tvs\_tvsid = actor\_cast\_tvshow.tvs\_id

INNER JOIN actors

ON actor\_cast\_tvshow.actor\_id = actors.actor\_id;

SELECT documentries.doc\_name, directors.director\_name

FROM documentries

INNER JOIN director\_cast\_documentries

ON documentries.doc\_id = director\_cast\_documentries.doc\_id

INNER JOIN directors

ON director\_cast\_documentries.director\_id = directors.director\_id;

SELECT webseries.wb\_name, actors.actor\_name

FROM webseries

INNER JOIN actor\_cast\_webseries

ON webseries.wb\_id = actor\_cast\_webseries.wb\_id

INNER JOIN actors

ON actor\_cast\_webseries.actor\_id = actors.actor\_id;

SELECT actors.actor\_name

FROM movie\_type

INNER JOIN actor\_cast\_movies

ON movie\_type.mv\_id = actor\_cast\_movies.mv\_id

INNER JOIN actors

ON actor\_cast\_movies.actor\_id = actors.actor\_id

WHERE movie\_type.mv\_name = 'i am legend';

SELECT a.actor\_name

FROM tvshow t

INNER JOIN actor\_cast\_tvshow ac ON t.tvs\_tvsid = ac.tvs\_id

INNER JOIN actors a ON ac.actor\_id = a.actor\_id

WHERE t.tvs\_name = 'peaky blinders';

SELECT a.actor\_name

FROM documentries d

INNER JOIN actor\_cast\_documentries ac ON d.doc\_id = ac.doc\_id

INNER JOIN actors a ON ac.actor\_id = a.actor\_id

WHERE d.doc\_name = 'found';

SELECT a.actor\_name

FROM webseries w

INNER JOIN actor\_cast\_webseries ac ON w.wb\_id = ac.wb\_id

INNER JOIN actors a ON ac.actor\_id = a.actor\_id

WHERE w.wb\_name = 'dark';

--AGGREGATE FUNCTIONS

SELECT AVG(rating\_score) FROM ratings INNER JOIN movie\_type ON ratings.rating\_id = movie\_type.mv\_id;

SELECT COUNT(\*) FROM ratings INNER JOIN movie\_type ON ratings.rating\_id = movie\_type.mv\_id WHERE rating\_score >= 4.0;

SELECT MAX(wb\_seasons) FROM webseries;

SELECT COUNT(DISTINCT rating\_id) FROM ratings INNER JOIN movie\_type ON ratings.rating\_id = movie\_type.mv\_id;

SELECT SUM(doc\_length) FROM documentries;

---TRANSACTIONS

BEGIN TRANSACTION;

UPDATE user\_data SET payment\_type = 'credit card' WHERE id = 12345;

COMMIT;

--this transaction deletes watch and download history using movie name and deletes a movie by name

BEGIN TRANSACTION;

DELETE FROM watched\_mv WHERE mv\_name = 'Movie Name';

DELETE FROM downloads\_mv WHERE mv\_name\_d = 'Movie Name';

DELETE FROM movie\_type WHERE mv\_name = 'Movie Name';

COMMIT;

--this transaction is used to change movie genre of a specific movie

BEGIN TRANSACTION;

UPDATE movie\_type SET mv\_genre = 'horror' WHERE mv\_name = 'i am legend';

COMMIT;

---this transaction insert tv show in respective table

BEGIN TRANSACTION;

INSERT INTO tvshow (tvs\_tvsid, tvs\_name, tvs\_language, tvs\_rating, tvs\_genre, tvs\_parts) VALUES (12345, 'TV Show Name', 'English', 4.5, 'Drama', 24);

INSERT INTO languages (lannguage\_id, language\_name) VALUES (5, 'latin');

INSERT INTO ratings (rating\_id, rating\_score, rating\_date) VALUES (2, 4.5, '2022-06-01');

INSERT INTO genre (genre\_id, genre\_name) VALUES (5, 'Drama');

COMMIT;

---PROCEDURES

----this proceduer is used to add new user in user\_data table

go

CREATE PROCEDURE add\_user

@id INT,

@username VARCHAR(30),

@email VARCHAR(50),

@dateofbirth DATE,

@payment\_type VARCHAR(30)

AS

BEGIN

INSERT INTO user\_data (id, username, email, dateofbirth, payment\_type) VALUES (@id, @username, @email, @dateofbirth, @payment\_type);

END

---------------this procedure is used to delete a existing user

go

CREATE PROCEDURE delete\_user

@id INT

AS

BEGIN

DELETE FROM ratings WHERE rating\_id = @id;

DELETE FROM user\_data WHERE id = @id;

END

----------------this procedure tells the name of show a user has watched

go

CREATE PROCEDURE get\_watched\_tv

@user\_id INT

AS

BEGIN

SELECT tvs\_name FROM tvshow INNER JOIN watched\_tv ON tvshow.tvs\_tvsid = watched\_tv.tv\_name WHERE watch\_id = @user\_id;

END

------FUNCTIONS

--this function gets the age of the user by using the date of birth given in the table

GO

CREATE FUNCTION get\_age(@dob DATE)

RETURNS INT

AS

BEGIN

DECLARE @age INT = DATEDIFF(YEAR, @dob, GETDATE());

RETURN @age;

END

----------------thiz function gets the most populer genre among all the movies by averaging the number of movie in each genre

GO

CREATE FUNCTION get\_popular\_genre()

RETURNS VARCHAR(20)

AS

BEGIN

DECLARE @genre VARCHAR(20);

SELECT @genre = genre\_name

FROM (SELECT top(1) genre\_name, COUNT(\*) as num\_movies FROM movie\_type INNER JOIN genre ON movie\_type.mv\_genre = genre.genre\_name GROUP BY genre\_name) as t

ORDER BY num\_movies DESC

RETURN @genre;

END

---------------this fucntion gets the highesr rated movie from the movie\_type table

go

CREATE FUNCTION highest\_rated\_movie()

RETURNS VARCHAR(30)

AS

BEGIN

DECLARE @highest\_rated\_movie VARCHAR(30)

SELECT @highest\_rated\_movie = mv\_name

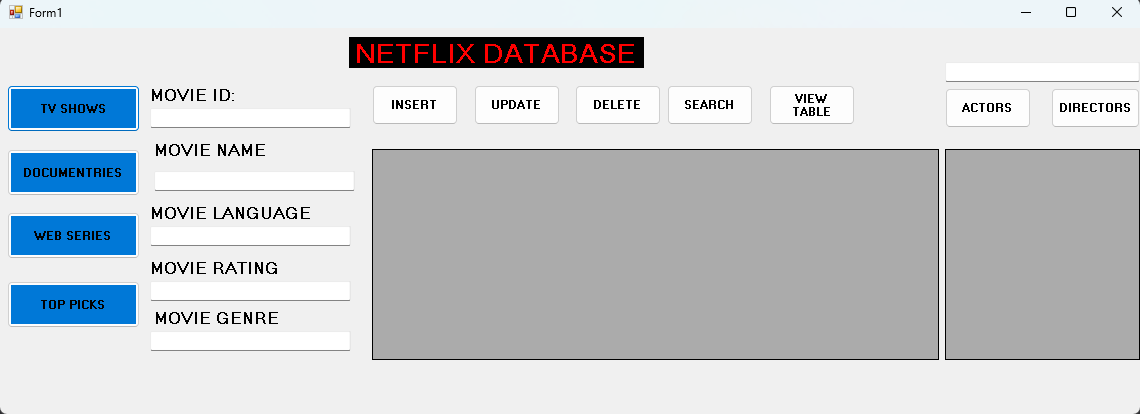
FROM movie\_type INNER JOIN ratings ON movie\_type.mv\_id = ratings.rating\_id

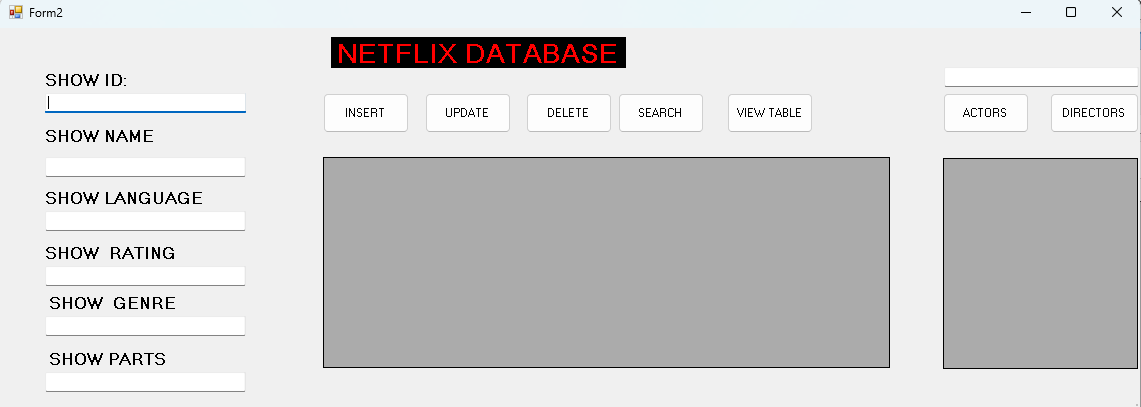
WHERE rating\_score = (SELECT MAX(rating\_score) FROM ratings)

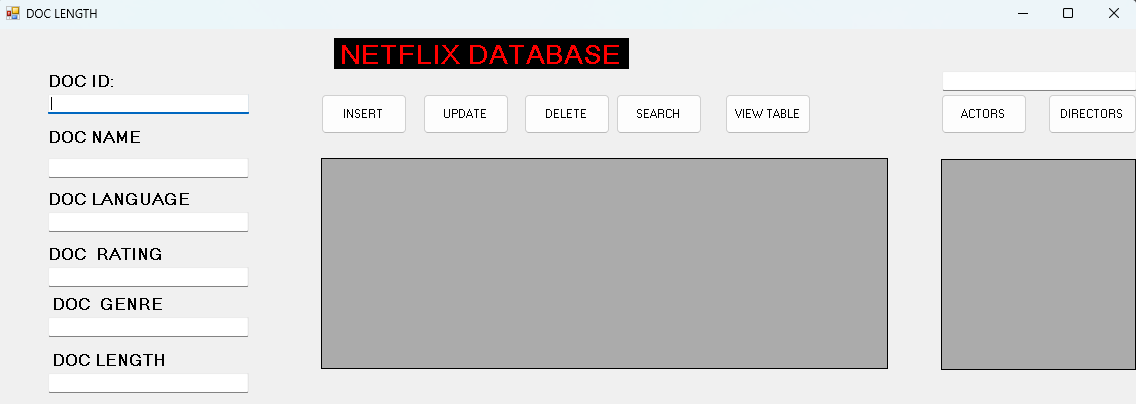
RETURN @highest\_rated\_movie

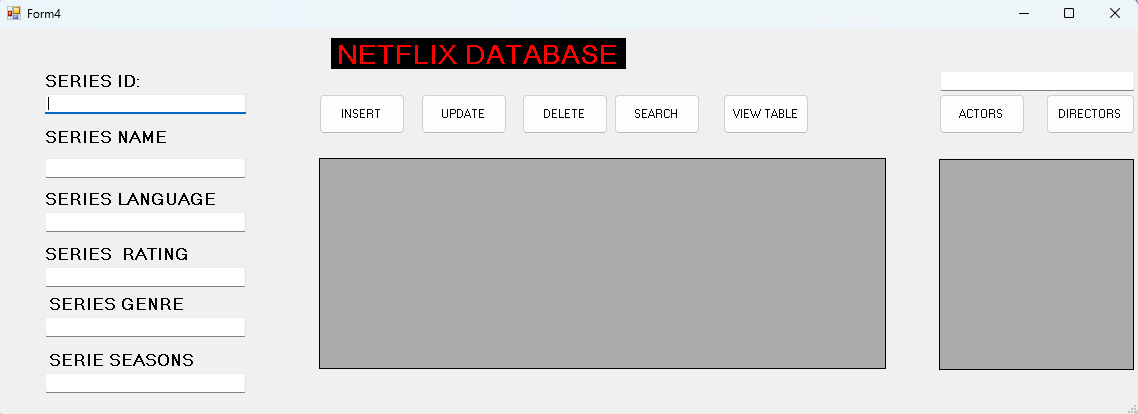
END

**Front-end:**

****

****

****

****

****

**Form 1(movie page):**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace dbms\_final\_proj

{

public partial class Form1 : Form

{

Form2 form2;

Form3 form3;

Form4 form4;

Form5 from5;

public Form1()

{

InitializeComponent();

form2 = new Form2();

form3 = new Form3();

form4 = new Form4();

from5 = new Form5();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'final\_project\_dbmsDataSet.movie\_type' table. You can move, or remove it, as needed.

}

private void button1\_Click(object sender, EventArgs e)

{

form2.ShowDialog();

}

private void button3\_Click(object sender, EventArgs e)

{

form3.ShowDialog();

}

private void button2\_Click(object sender, EventArgs e)

{

form4.ShowDialog();

}

private void button4\_Click(object sender, EventArgs e)

{

string connectionstring= "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("insert into movie\_type values (@mv\_id,@mv\_name,@mv\_language,@mv\_rating,@mv\_genre)", con);

cmd.Parameters.AddWithValue("@mv\_id", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@mv\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@mv\_language", lanBox3.Text);

cmd.Parameters.AddWithValue("@mv\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@mv\_genre", genreBox5.Text);

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("inserted successfully!");

}

private void button5\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("update movie\_type set mv\_name=@mv\_name ,mv\_language=@mv\_language, mv\_rating=@mv\_rating, mv\_genre= @mv\_genre where mv\_id=@mv\_id", con);

cmd.Parameters.AddWithValue("@mv\_id", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@mv\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@mv\_language", lanBox3.Text);

cmd.Parameters.AddWithValue("@mv\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@mv\_genre", genreBox5.Text);

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("updated successfully!");

}

private void button6\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Delete movie\_type where mv\_id=@mv\_id", con);

cmd.Parameters.AddWithValue("@mv\_id", int.Parse(idBox1.Text));

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("deleted successfully!");

}

private void button7\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from movie\_type where mv\_id=@mv\_id", con);

cmd.Parameters.AddWithValue("@mv\_id", int.Parse(idBox1.Text));

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button10\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from movie\_type", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button8\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select actors.actor\_name from actor\_cast\_movies INNER JOIN actors ON actor\_cast\_movies.actor\_id=actors.actor\_id where mv\_id=(Select mv\_id from movie\_type where mv\_name=@mv\_name)", con);

cmd.Parameters.AddWithValue("@mv\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button9\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select directors.director\_name from director\_cast\_movies INNER JOIN directors ON director\_cast\_movies.director\_id=directors.director\_id where mv\_id=(Select mv\_id from movie\_type where mv\_name=@mv\_name)", con);

cmd.Parameters.AddWithValue("@mv\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button11\_Click(object sender, EventArgs e)

{

from5.ShowDialog();

}

}

}

**Form 2(TV shows page)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace dbms\_final\_proj

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("insert into tvshow values (@tvs\_tvsid,@tvs\_name,@tvs\_language,@tvs\_rating,@tvs\_genre,@tvs\_parts)", con);

cmd.Parameters.AddWithValue("@tvs\_tvsid", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@tvs\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@tvs\_language", languageBox3.Text);

cmd.Parameters.AddWithValue("@tvs\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@tvs\_genre", genreBox5.Text);

cmd.Parameters.AddWithValue("@tvs\_parts", int.Parse(partsBox6.Text));

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("inserted successfully!");

}

private void button5\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("update tvshow set tvs\_name=@tvs\_name ,tvs\_language=@tvs\_language, tvs\_rating=@tvs\_rating, tvs\_genre= @tvs\_genre,tvs\_parts=@tvs\_parts where tvs\_tvsid=@tvs\_tvsid", con);

cmd.Parameters.AddWithValue("@tvs\_tvsid", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@tvs\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@tvs\_language", languageBox3.Text);

cmd.Parameters.AddWithValue("@tvs\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@tvs\_genre", genreBox5.Text);

cmd.Parameters.AddWithValue("@tvs\_[arts", int.Parse(partsBox6.Text));

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("updated successfully!");

}

private void button6\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Delete tvshow where tvs\_tvsid=@tvs\_tvsid", con);

cmd.Parameters.AddWithValue("@tvs\_tvsid", int.Parse(idBox1.Text));

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("deleted successfully!");

}

private void button7\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from tvshow where tvs\_tvsid=@tvs\_tvsid", con);

cmd.Parameters.AddWithValue("@tvs\_tvsid", int.Parse(idBox1.Text));

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from tvshow", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button8\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select actors.actor\_name from actor\_cast\_tvshow INNER JOIN actors ON actor\_cast\_tvshow.actor\_id=actors.actor\_id where tvs\_id=(Select tvs\_tvsid from tvshow where tvs\_name=@tvs\_name)", con);

cmd.Parameters.AddWithValue("@tvs\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button9\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select directors.director\_name from director\_cast\_tvshow INNER JOIN directors ON director\_cast\_tvshow.director\_id=directors.director\_id where tvs\_id=(Select tvs\_tvsid from tvshow where tvs\_name=@tvs\_name)", con);

cmd.Parameters.AddWithValue("@tvs\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

}

}

**Form 3 (documentaries page)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace dbms\_final\_proj

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void button4\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("insert into documentries values (@doc\_id,@doc\_name,@doc\_language,@doc\_rating,@doc\_genre,@doc\_length)", con);

cmd.Parameters.AddWithValue("@doc\_id", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@doc\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@doc\_language", languageBox3.Text);

cmd.Parameters.AddWithValue("@doc\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@doc\_genre", genreBox5.Text);

cmd.Parameters.AddWithValue("@doc\_length",lengthBox6.Text);

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("inserted successfully!");

}

private void button5\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("update documentries set doc\_name=@doc\_name ,doc\_language=@doc\_language, doc\_rating=@doc\_rating, doc\_genre= @doc\_genre,doc\_length=@doc\_length where doc\_id=@doc\_id", con);

cmd.Parameters.AddWithValue("@doc\_id", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@doc\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@doc\_language", languageBox3.Text);

cmd.Parameters.AddWithValue("@doc\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@doc\_genre", genreBox5.Text);

cmd.Parameters.AddWithValue("@doc\_length",lengthBox6.Text);

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("updated successfully!");

}

private void button6\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Delete documentries where doc\_id=@doc\_id", con);

cmd.Parameters.AddWithValue("@doc\_id", int.Parse(idBox1.Text));

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("deleted successfully!");

}

private void button7\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from documentries where doc\_id=@doc\_id", con);

cmd.Parameters.AddWithValue("@doc\_id", int.Parse(idBox1.Text));

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from documentries", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button8\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select actors.actor\_name from actor\_cast\_documentries INNER JOIN actors ON actor\_cast\_documentries.actor\_id=actors.actor\_id where doc\_id=(Select doc\_id from documentries where doc\_name=@doc\_name)", con);

cmd.Parameters.AddWithValue("@doc\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button9\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select directors.director\_name from director\_cast\_documentries INNER JOIN directors ON director\_cast\_documentries.director\_id=directors.director\_id where doc\_id=(Select doc\_id from documentries where doc\_name=@doc\_name)", con);

cmd.Parameters.AddWithValue("@doc\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

}

}

**Form 4(web series page)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace dbms\_final\_proj

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void button4\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("insert into webseries values (@wb\_id,@wb\_name,@wb\_language,@wb\_rating,@wb\_genre,@wb\_seasons)", con);

cmd.Parameters.AddWithValue("@wb\_id", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@wb\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@wb\_language", languageBox3.Text);

cmd.Parameters.AddWithValue("@wb\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@wb\_genre", genreBox5.Text);

cmd.Parameters.AddWithValue("@wb\_seasons", seasonsBox6.Text);

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("inserted successfully!");

}

private void button5\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("update webseries set wb\_name=@wb\_name ,wb\_language=@wb\_language, wb\_rating=@wb\_rating, wb\_genre= @wb\_genre,wb\_seasons=@wb\_seasons where wb\_id=@wb\_id", con);

cmd.Parameters.AddWithValue("@wb\_id", int.Parse(idBox1.Text));

cmd.Parameters.AddWithValue("@wb\_name", nameBox4.Text);

cmd.Parameters.AddWithValue("@wb\_language", languageBox3.Text);

cmd.Parameters.AddWithValue("@wb\_rating", decimal.Parse(ratingBox2.Text));

cmd.Parameters.AddWithValue("@wb\_genre", genreBox5.Text);

cmd.Parameters.AddWithValue("@wb\_seasons", seasonsBox6.Text);

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("updated successfully!");

}

private void button6\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Delete webseries where wb\_id=@wb\_id", con);

cmd.Parameters.AddWithValue("@wb\_id", int.Parse(idBox1.Text));

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("deleted successfully!");

}

private void button7\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from webseries where wb\_id=@wb\_id", con);

cmd.Parameters.AddWithValue("@wb\_id", int.Parse(idBox1.Text));

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select \*from webseries", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button8\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select actors.actor\_name from actor\_cast\_webseries INNER JOIN actors ON actor\_cast\_webseries.actor\_id=actors.actor\_id where wb\_id=(Select wb\_id from webseries where wb\_name=@wb\_name)", con);

cmd.Parameters.AddWithValue("@wb\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button9\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("Select directors.director\_name from director\_cast\_webseries INNER JOIN directors ON director\_cast\_webseries.director\_id=directors.director\_id where wb\_id=(Select wb\_id from webseries where wb\_name=@wb\_name)", con);

cmd.Parameters.AddWithValue("wb\_name", castBox1.Text);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

}

}

**Form 4(Top page)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace dbms\_final\_proj

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("select top(3) \*from movie\_type order by mv\_rating desc", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("select top(3) \* from tvshow order by tvs\_rating desc", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button3\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("select top(3) \*from webseries order by wb\_rating desc", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

string connectionstring = "Data Source=DESKTOP-M4MPGOD\\SQLEXPRESS;Initial Catalog=final\_project\_dbms;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionstring);

con.Open();

SqlCommand cmd = new SqlCommand("select top(3) \*from documentries order by doc\_rating desc", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

cmd.ExecuteNonQuery();

con.Close();

}

}

}