Objective - This project to analyse the movies dataset available with various tables. Using the SQL queries for Data Manipulation. The main objective of querying this database is to practice and efficiently use SQL syntax in asking questions and producing desired results.

I have just completed my first SQL project. This project showcases SQL syntaxes that I have learned in the last few weeks since beginning my journey to becoming a Data Analyst. The project was executed with MySQL. Here I analysed SQL dataset of movies. This dataset was obtained from GitHub. This dataset was chosen because it contains all of the necessary tables and will allow me to run basic to intermediate queries. In the database, there are 09 tables. I included the questions for which I planned to query the database.

Key Takeaways:-

1. Jack Nicholson is the actor who played role in ChinaTown movie.
2. The Movie name American Beauty has the Highest Number of Ratings.
3. Used Update and Set Function to change the language of Movie.
4. The Genre Mystery has maximum movie durations.
5. Created the view for first name, last name, title of the movie & role played by particular actor
6. In Year 1958 was the year when the most Genre Mystery movie were produced.
7. Danny Boyle was the only director with most movies directed in single genre.

**SQL Project Sheet :-**

**Question 1 –**

**Write a query where it should contain all the data of the movies which were released after 1995 having their movie duration greater than 120 and should be including A in their movie title.**

**select \* from movie**

**where mov\_year > 1995**

**and mov\_time > 120**

**and mov\_title like '%a%';**

**-- Q2. Write an SQL query to find the actors who played a role in the**

**-- movie 'Chinatown'. Fetch all the fields of actor table. (Hint: Use**

**-- the IN operator)**

**select a.act\_id , a.act\_fname ,a.act\_lname, a.act\_gender , m.mov\_title**

**from actor as a**

**inner join cast as c on a.act\_id = c.act\_id**

**inner join movie as m on c.mov\_id = m.mov\_id**

**where mov\_title in ('Chinatown')**

**group by 1;**

**-- Q3. Write an SQL query for extracting the data from the ratings**

**-- table for the movie who got the maximum number of ratings.**

**select m.mov\_title, max(r.num\_o\_ratings) as max\_ratings**

**from movie as m**

**inner join ratings as r on m.mov\_id = r.mov\_id**

**group by 1**

**order by 2 desc;**

**-- Q4. Write an SQL query to determine the Top 7 movies which were**

**-- released in United Kingdom. Sort the data in ascending order**

**-- of the movie year.**

**select mov\_id, mov\_title, mov\_year, mov\_time, mov\_lang, mov\_dt\_rel, mov\_rel\_country**

**from movie where mov\_rel\_country ='UK'**

**order by 3 limit 7;**

**-- Q5. Set the language of movie language as 'Chinese' for the movie**

**-- which has its existing language as Japanese and their movie**

**-- year was 2001.**

**update movie set mov\_lang='Chinese' where mov\_lang='Japanese' and mov\_year=2001;**

**-- Q6. Print genre title, maximum movie duration and the count the**

**-- number of movies in each genre**

**select g.gen\_title , max(m.mov\_time) as max\_duration, count(m.mov\_id) as movie\_count**

**from genres as g**

**left join movie\_genres as mg on(g.gen\_id = mg.gen\_id)**

**left join movie as m on(mg.mov\_id = m.mov\_id)**

**group by 1**

**order by 3 desc;**

**-- Q7. Create a view which should contain the first name, last name,**

**-- title of the movie & role played by particular actor.**

**create view movie\_details as (**

**select a.act\_fname, a.act\_lname, m.mov\_title, c.role as role\_played**

**from actor as a**

**inner join cast as c on(a.act\_id = c.act\_id)**

**inner join movie as m on(m.mov\_id = c.mov\_id));**

**-- Q8. Display the movies that were released before 31st March 1995.**

**select \* from movie where mov\_dt\_rel <'31/03/1995';**

**-- Q9. Write a query which fetch the first name, last name & role**

**-- played by the actor where output should all exclude Male actors**

**select a.act\_id, a.act\_fname, a.act\_lname, a.act\_gender, c.role**

**from actor as a**

**inner join cast as c on a.act\_id = c.act\_id**

**where act\_gender != 'M';**

**-- Q10. Insert five rows into the cast table where the ids for movie**

**-- should be 936,939,942,930,941 and their respective roles**

**-- should be Darth Vader, Sarah Connor, Ethan Hunt, Travis**

**-- Bickle, Antoine Doinel & their actor ids should be set up as**

**-- 126,140,135,131,144**

**insert into cast (mov\_id, role, act\_id) values**

**(936,'Darth Vader',126),**

**(939,'Sarah Connor',140),**

**(942,'ethan hunt',135),**

**(930,'Travis bickle',131),**

**(941,'Antoine Doinel',144);**

**-- Q11. From the following table,**

**-- write a SQL query to find out who was cast in the movie 'Annie Hall'.**

**-- Return actor first name, last name and role.**

**select act\_fname, act\_lname, role**

**FROM actor**

**JOIN cast ON actor.act\_id = cast.act\_id**

**JOIN movie ON cast.mov\_id = movie.mov\_id AND movie.mov\_title = 'Annie Hall';**

**-- Q12. From the following table,**

**-- write a SQL query to calculate the average movie length and count the number of --- movies in each genre.**

**-- Return genre title, average time and number of movies for each genre.**

**select gen\_title as Genre\_title, AVG(mov\_time) as avg\_movie\_time, count(gen\_title) as Genre\_count**

**FROM movie**

**join movie\_genres**

**join genres**

**group by gen\_title**

**order by 2 desc;**

**-- Q13. From the following tables, write a SQL query to find the years when most of the ‘Mystery Movies’ produced.**

**-- Count the number of generic title and compute their average rating.**

**-- Group the result set on movie release year, generic title.**

**-- Return movie year, generic title, number of generic title and average rating.**

**SELECT m1.mov\_year, g1.gen\_title as Genre\_title, COUNT(g1.gen\_title) as count, AVG(r1.rev\_stars) as Avg\_ratings**

**FROM movie m1**

**JOIN movie\_genres mg1 ON m1.mov\_id = mg1.mov\_id**

**JOIN genres g1 ON mg1.gen\_id = g1.gen\_id**

**JOIN ratings r1 ON m1.mov\_id = r1.mov\_id**

**JOIN movie m2 ON m1.mov\_year = m2.mov\_year**

**JOIN movie\_genres mg2 ON m2.mov\_id = mg2.mov\_id**

**JOIN genres g2 ON mg2.gen\_id = g2.gen\_id**

**WHERE g1.gen\_title = 'Mystery'**

**AND g2.gen\_title = 'Mystery'**

**GROUP BY m1.mov\_year, g1.gen\_title;**

**-- Q14. From the following table, write a SQL query to find the directors who have directed films in a variety of genres.**

**-- Group the result set on director first name, last name and generic title.**

**-- Sort the result-set in ascending order by director first name and last name.**

**-- Return director first name, last name and number of genres movies.**

**SELECT d.dir\_fname, d.dir\_lname, g.gen\_title, COUNT(g.gen\_title)**

**FROM director d**

**JOIN movie\_direction md ON d.dir\_id = md.dir\_id**

**JOIN movie\_genres mg ON md.mov\_id = mg.mov\_id**

**JOIN genres g ON mg.gen\_id = g.gen\_id**

**GROUP BY d.dir\_fname, d.dir\_lname, g.gen\_title**

**ORDER BY d.dir\_fname, d.dir\_lname;**