Connect, Message, Like,Follow&Share, 100% Free Counselling➔ThankYou

Follow me on Linkedin

SQL for Data Science

SQL stands for Structured Query Language which is used to deal with Relational Databases to query from and manipulate databases. In the field of Data Science most of the time you are supposed to fetch the data from any RDBMS and run some simple and complex queries to retrieve and extract data in different ways to understand relationships or irregularities that exist in the dataset. It is a complete beginner-friendly guide that while reading and following will give an experience like a cruise starting with very basic SQL to playing with a dataset with complex queries.

What is a database and Database Management System?

Before defining a database let us define what is data? Data is a raw piece of information about any single or multiple people or any object. And while working in Data science we aim to work with raw data and extract meaningful information that can help to drive business decisions. while working with any data what is a primary requirement for data.

Requirements for Data

1. Integrity – Data should be accurate and consistent. In simple words, it means after and before any changes data should be in a consistent state.
2. Availability – It should be available each time.

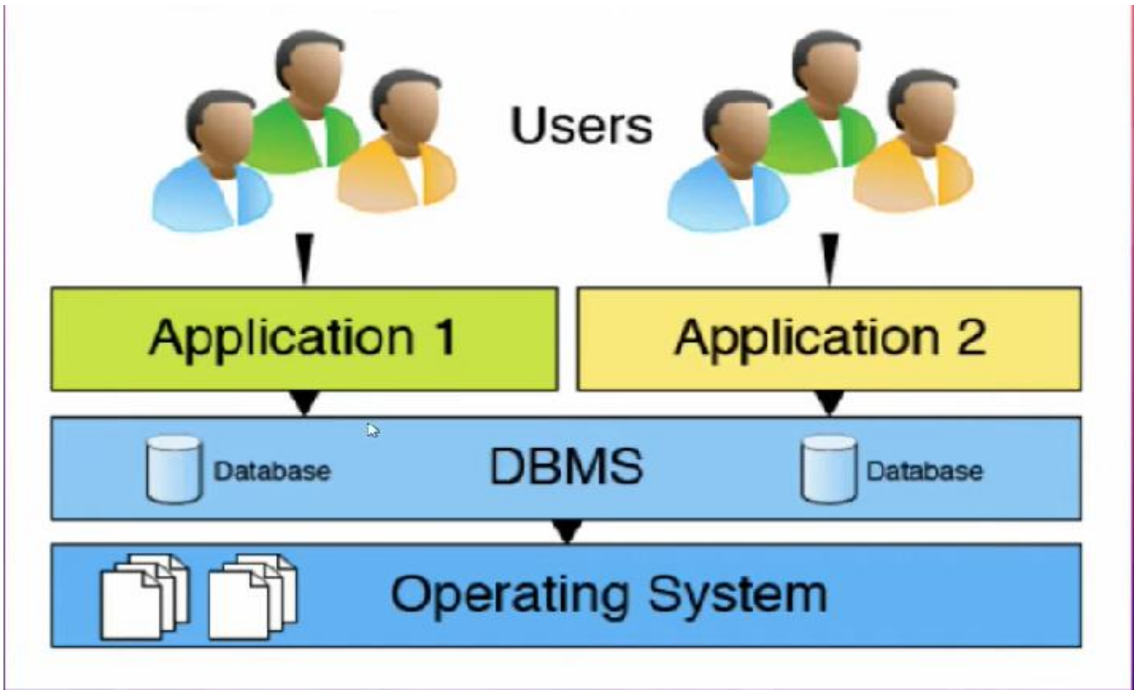
3. security – The information of individuals should be secure from third-party applications without permission.
4. Independent of Application – the same data should be available on different applications.
5. concurrency – If multiple users are accessing the same data then they should get the same data in revert.

By keeping these basic 5 properties in mind any database is designed. Now let us define a database.

Database – It is a shared collection of interrelated data. In simple words, it is a system that is used to store, organize, and efficiently retrieve data.

- **Database Management System(DBMS) –** DBMS is a system that enables users to define, create, manage, and control access to the database. DBMS typically have high costs and require high-end hardware configurations.

We have seen the requirement of data and this is only the core functionality of DBMS and that's why It is so popular that each company uses at least one database. DBMS runs on the operating system, DBMS manages databases and DBMS interacts with Applications, and applications are used by users.



Now let us look at the most popular types of databases that are available in the market.

Types of Database

1. Relational Databases – The database is based on a relational model which means data is stored in form of a table(collection of rows and columns). It is the most popular type of DBMS that is being used by organizations for many past years.
2. NoSQL Databases – Not only SQL databases are used to store a wide range of datasets. It is new in the market and many organizations use it to store unstructured data because it is a document-type database that is used when you have to work with semi-structured or unstructured data.
3. Cloud Databases – A type of database where data is stored online or on a virtual machine and executes over cloud computing platforms. In recent times there are many cloud service providers like AWS, Microsoft Azure, IBM, etc.

These are the most popular databases of recent time and apart from this, there are many different types of databases like distributed database, network database, Hierarchical database, etc. Our article discussion topic lies under Relational database where SQL is used as a medium to communicate with a database. So before diving into SQL let us understand how databases are designed where ER model will help us to understand the procedure of database designing.

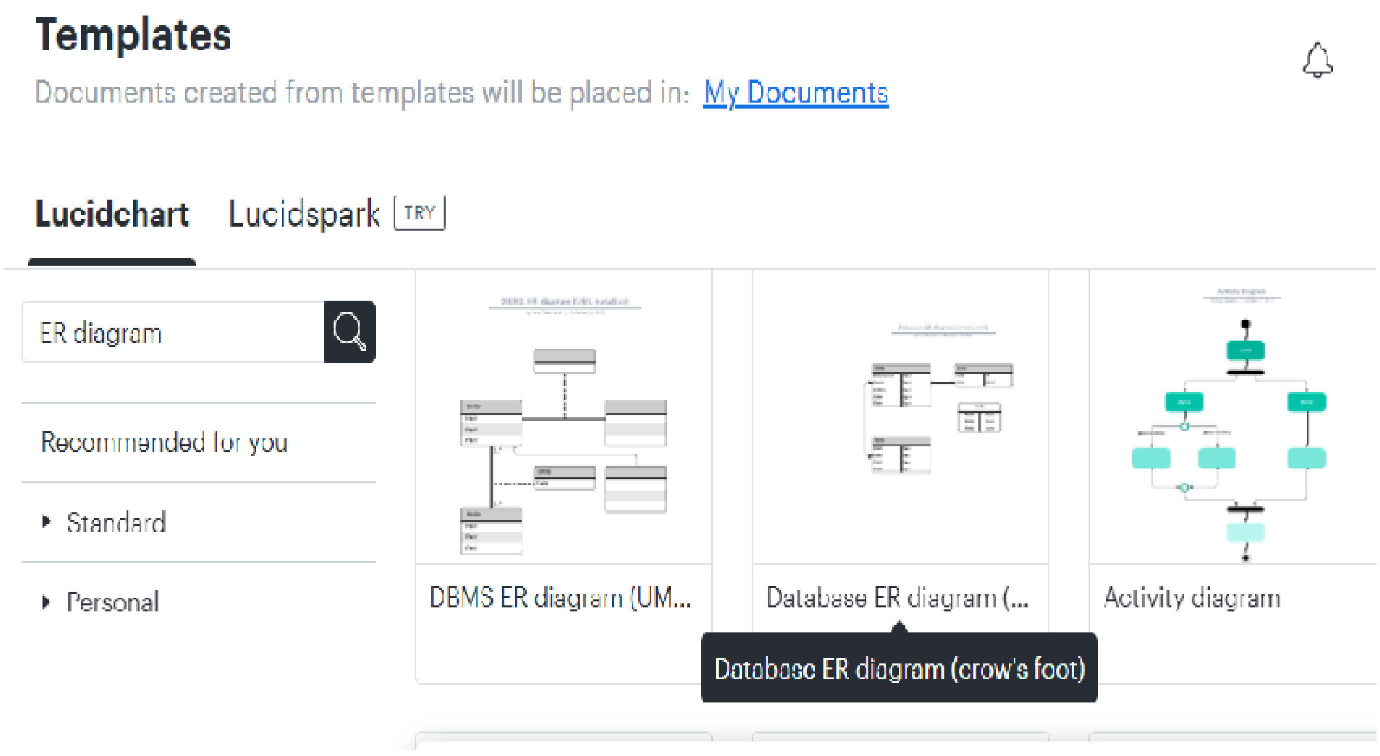
ER Model (Entity-Relationship Model)

ER model is a high-level data model that is used to define data elements and their relationship for a specific use case. In simple words, ER model helps you to express the complete conceptual(logical) structure of your database design for any use case. while ER modelling the structure is portrayed which is known as an ER diagram.

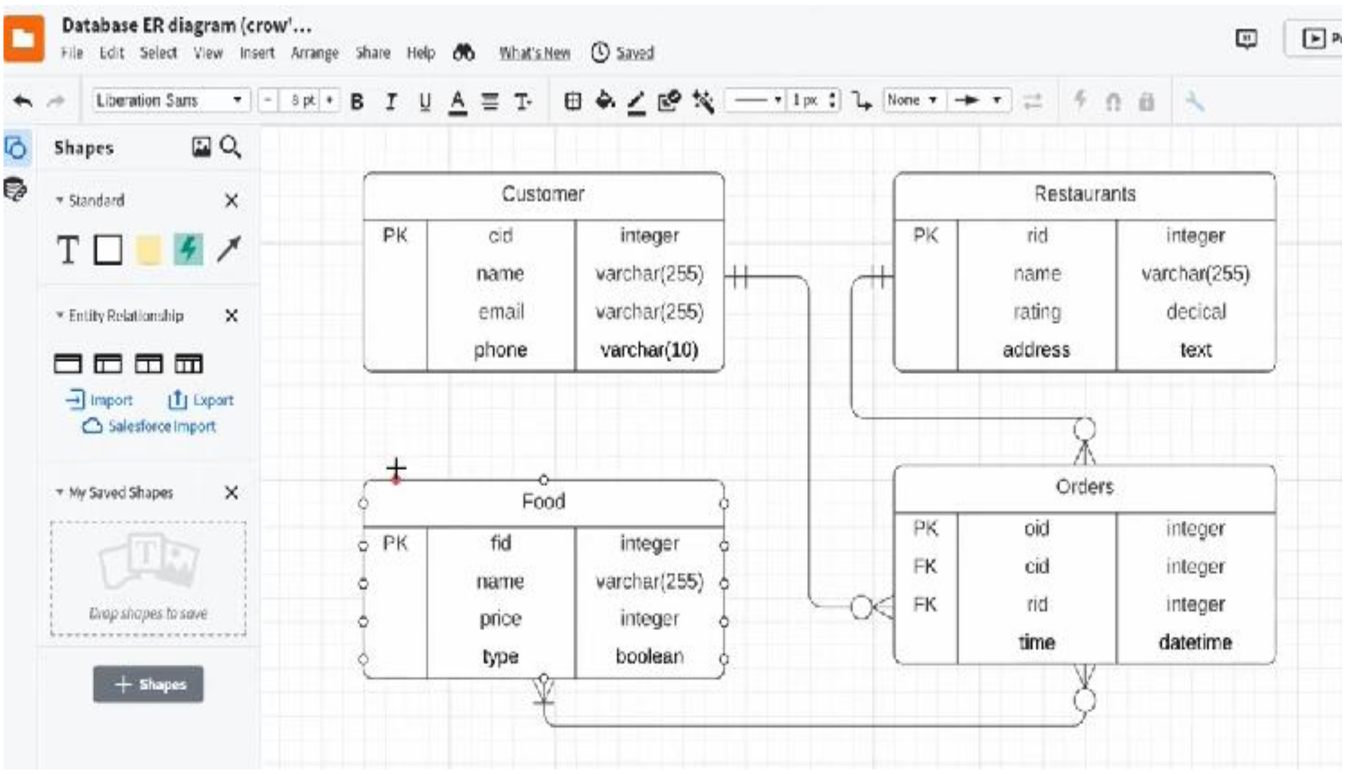
What is a requirement of ER Diagram?

Let’s take a scenario where you are working closely with databases and your senior manager asks to present a database design you think to implement for the problem statement. Now you cannot show the database server to a manager or the client because he is not a technical person so you have to express it in paper or presentation design which is known as an ER diagram. The condition is the same as when you are constructing your new home and consult with a civil engineer, then he does not give or explain the design with a complete building picture and setup, rather he is given a pen-paper design to understand the logical structure.

Now you might be wondering about How to implement an ER diagram? There are many offline and online tools available to draw ER diagrams. For demonstration, I am using a lucid app which is an online tool. you need to sign up and log in with your email address. click on create and choose an option from a template and search for ER diagram on the left side search panel and select template with crowfoot notation. Now you can add the entities and relationship here as shown below figure. Entities mean the tables in your database and the relationship expresses how one table attribute is related to other table attributes. The types of relationship with crowfoot notation are explained below this section.



After choosing a template, start drawing ER diagram. On the left-hand side, you will get a table structure to add in a diagram and define each table with column names and their data types. SQL support almost all the data types as other language supports. To read more about SQL data types please visit this page. The below ER diagram is just a sample of online food management services to demonstrate How an ER diagram looks.



Customer

PK	cid	integer
	name	varchar(255)
	email	varchar(255)
	phone	varchar(10)

Restaurants

PK	rid	integer
	name	varchar(255)
	rating	decical
	address	text

Food

PK	fid	integer
	name	varchar(255)
	price	integer
	type	boolean

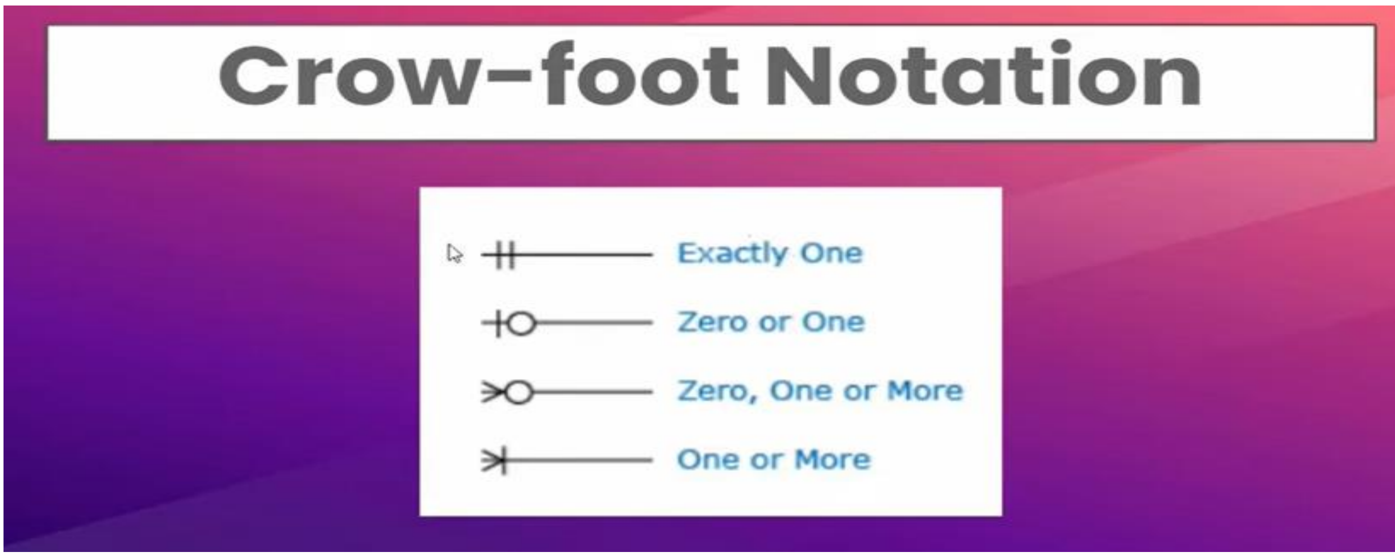
Orders

PK	oid	integer
FK	cid	integer
FK	rid	integer
	time	datetime

After creating the ER diagram you can export it in SQL query by clicking on the left side export button and copying the SQL query. using this SQL query create a database in the localhost server.

Crow-Foot Notation

Crow-foot Notation is an expression to express the relationship between 2 Entities. there ate four types of relationships as One-one relationship, many-to relationship, many-to-one relationship, and vice-versa.

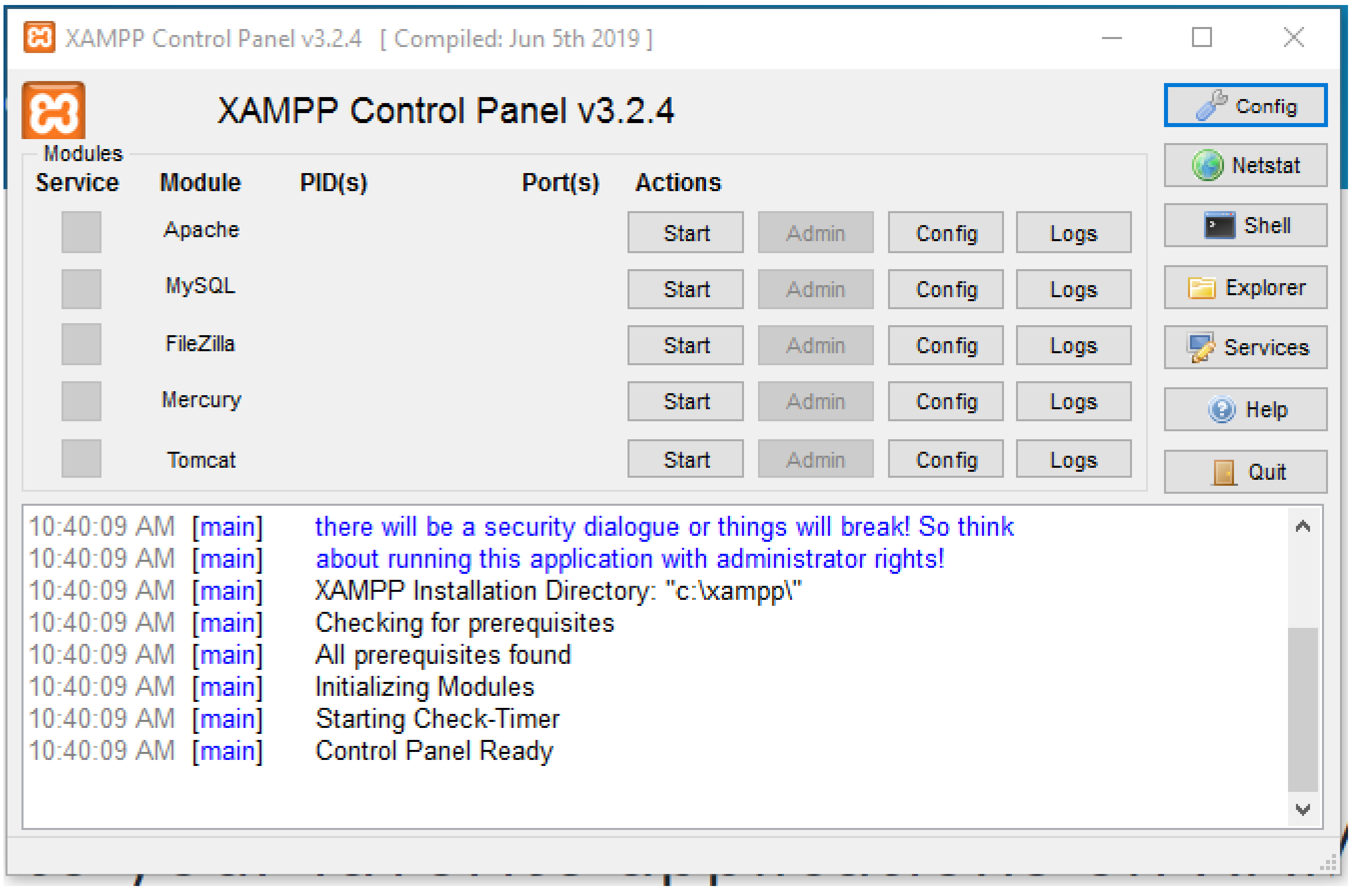


Getting Started with SQL

Database Server

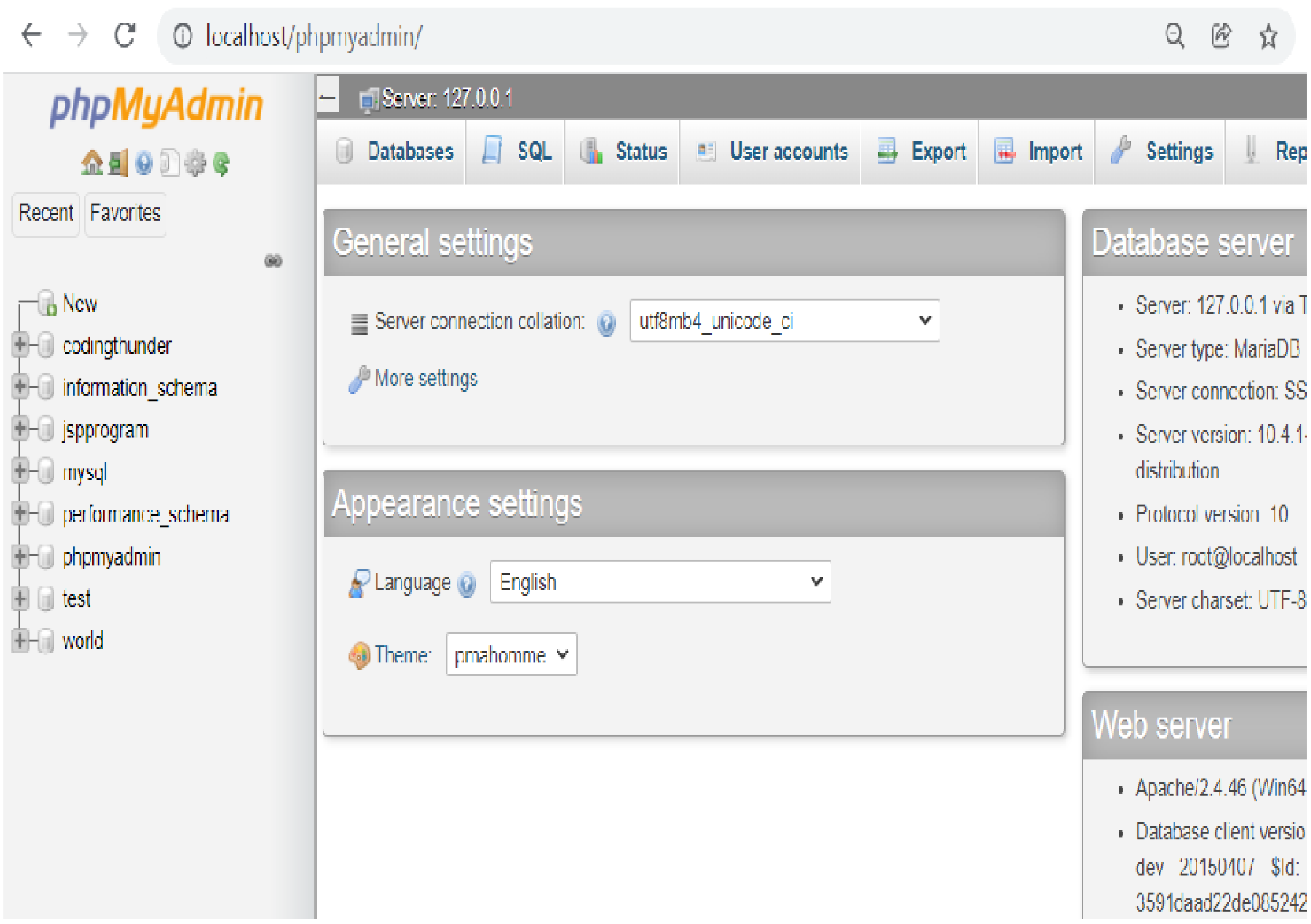
To get started working and playing with SQL queries on datasets we require any database. In this article, we will work with MySQL on the XAMPP database server. XAMPP provides you with a localhost environment with APACHE support where you can create as many databases as you want and work closely with the database on your projects and connect it with python easily. To install the XAMPP, please follow this link. We can work closely with MySQL, PHP, and Perl in the XAMPP database server.

After installation search XAMPP in the windows search box and open the Xampp control panel. to start the server start Apache and MySQL.



```
10:40:09 AM  [main]    there will be a security dialogue or things will break! So think
10:40:09 AM  [main]    about running this application with administrator rights!
10:40:09 AM  [main]    XAMPP Installation Directory: "c:\xampp\"
10:40:09 AM  [main]    Checking for prerequisites
10:40:09 AM  [main]    All prerequisites found
10:40:09 AM  [main]    Initializing Modules
10:40:09 AM  [main]    Starting Check-Timer
10:40:09 AM  [main]    Control Panel Ready
```

After starting just follow this localhost link in your browser and your PHPMyAdmin server will open.



Server connection collation: utf8mb4_unicode_ci

Language: English

Theme: pmahomme

- Server: 127.0.0.1 via T
- Server type: MariaDB
- Server connection: SS
- Server version: 10.4.1 distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8

- Apache/2.4.46 (Win64
- Database client versio dev - 20180407 - $Id: 3591daad22de085242

Constraints in SQL

After knowing about SQL and its data types it's important to know SQL constraints before writing queries on data. Constraints are used to specify some rules over the table columns which limit the user by inserting any other values in the table. a constraint can be used to control type, format, length, and collection of values in a column. let us study the different constraints supported by SQL.

1. Unique – Unique defines that the values column will hold will be unique and no duplicates are allowed. It can accept a NULL value.
2. Primary Key – It is a column that uniquely identifies any record in a table. so the column you make a primary key holds all values unique as well None of the values should be NULL.
3. Not Null – It ensures that the column cannot have Null Values.

4. Default – Default constraint is used to pass a default value in a certain column for a record if while inserting a data no value is provided for the user for a particular column.
5. Check – It Ensures that the value which is inserted by the user should pass a certain condition.
6. Foreign Key – It helps to prevent the data between two tables. Also helps to explain the relationship between the two tables
7. Auto-Increment – If any column is Integer type then it can be defined as Autoincrement which says that each time when the record is inserted increases the value by one.

In the next section of the article, we will understand the practical use of constraints while studying different SQL commands.

TYPES OF SQL COMMANDS

Now we are at SQL to deal with databases. there are different SQL commands which help to deal with databases. let us understand each type of command and what function each includes. SQL is a core language that every RDBMS software uses. Only there can be a little bit of difference between syntax of different DBMS software. we will work with MySQL.

There are different data types in SQL and if you want to read and learn about different data types then please refer to this page.

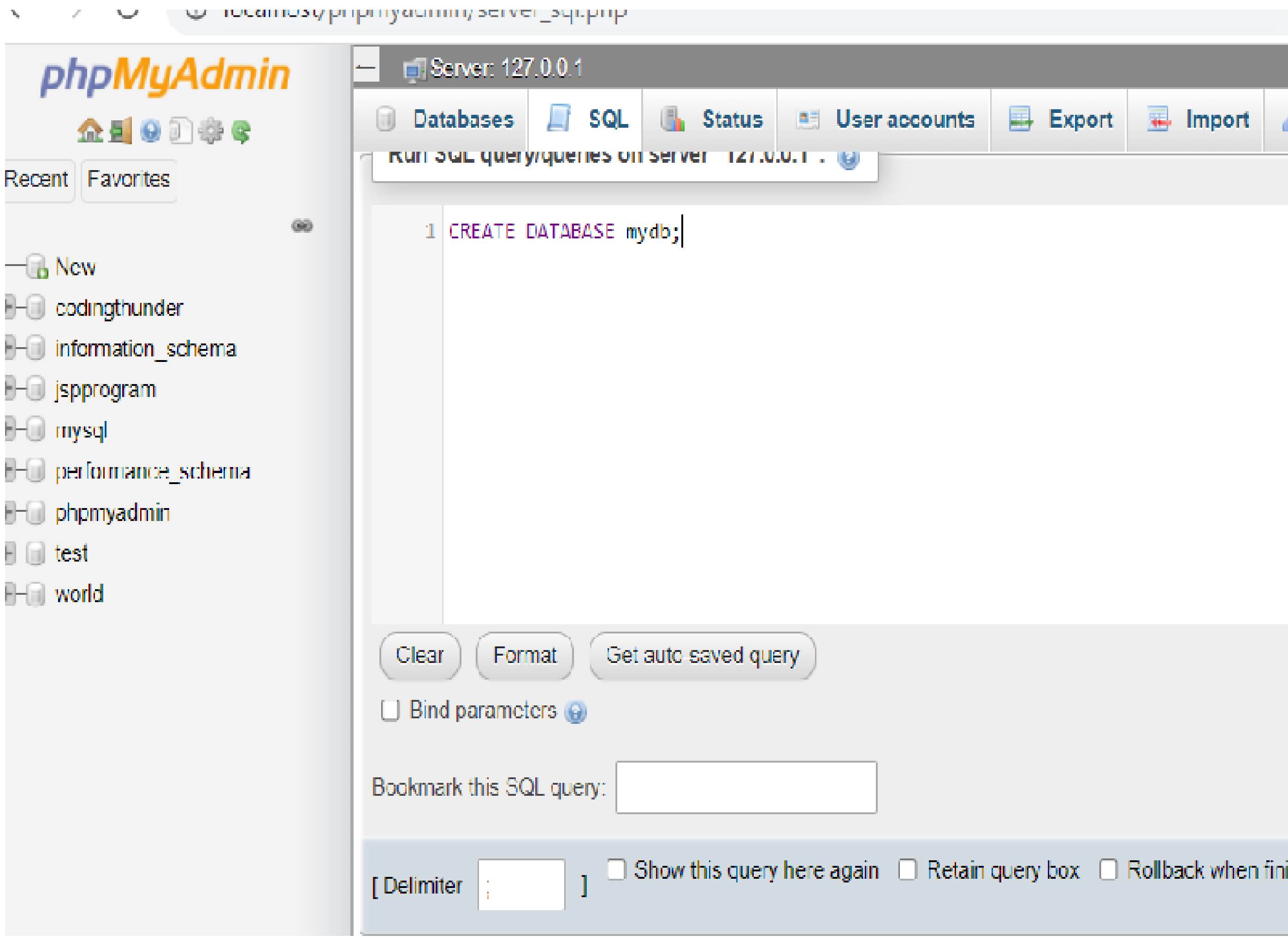
1) DDL (Data Definition Language)

It is used to define database structure, add constraints, indexes. using DDL commands we create a structure of the database. The commands under DDL are as follows.

A) create – It is used in a database or for creating a table. let us see the syntax to create a database and to create a table.

```
create database database_name
create table table_name(columns datatype)
```

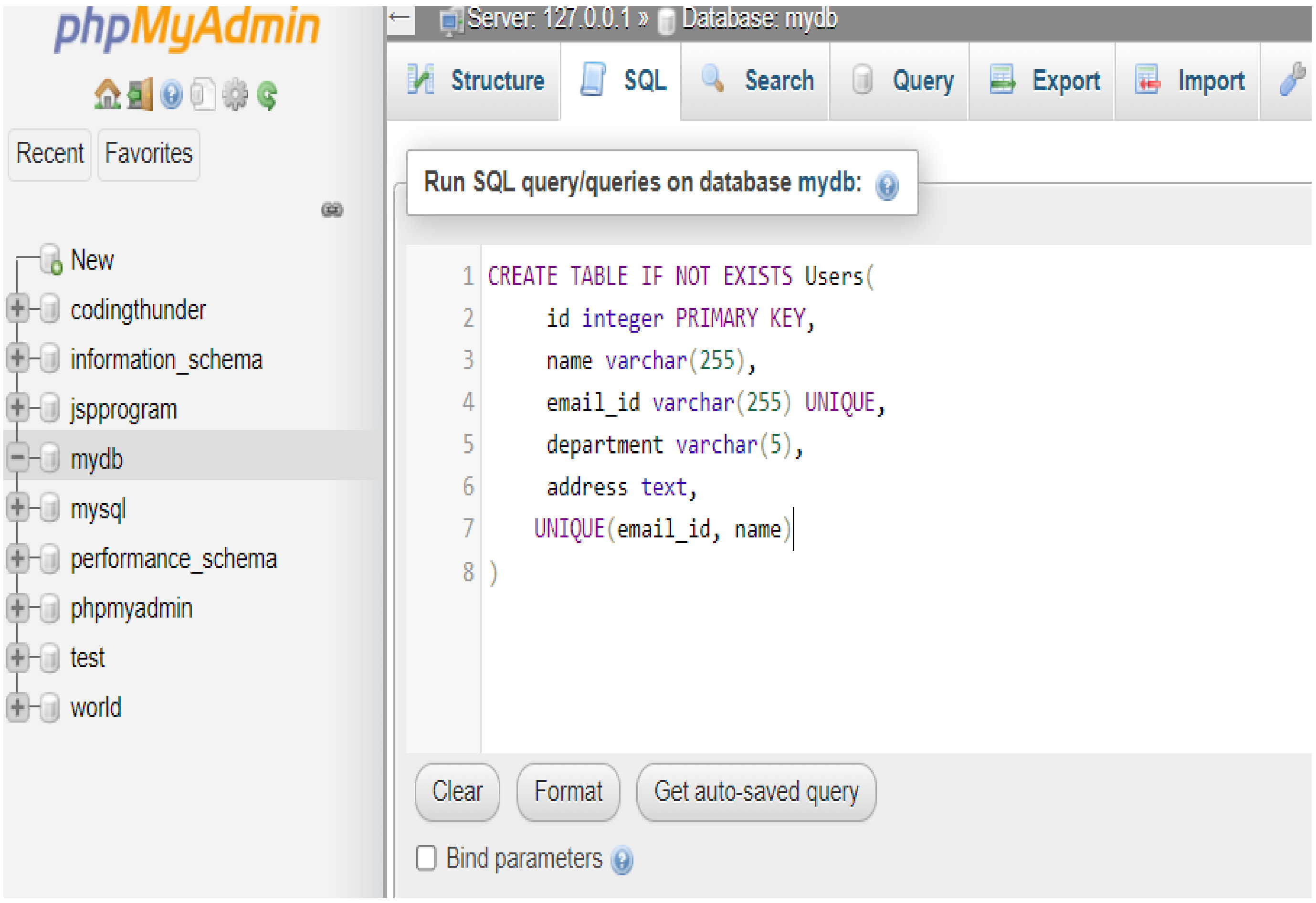
To create a database in the database server using SQL visit PhpMyAdmin and go to the SQL tab and write a query and run using the go button. we can also use Exist, a logical operator while creating a database to check that the table we are creating already exists then it will not create another.


```
CREATE DATABASE mydb;
```

After creating the database go to that database and in the SQL tab create a table. To create a table you can also use the same command as follow.

```
CREATE TABLE IF NOT EXISTS Students(
roll_no integer PRIMARY KEY,
name varchar(255),
email_id varchar(255) UNIQUE,
department varchar(5)
)
```

we have used two constraints in the above query, and it represents roll number as primary key and email should be unique. In the same way, you can use different constrain and one constraint can also be applied on two columns at a time after the end of the query as shown below.



```
CREATE TABLE IF NOT EXISTS Users(
     id integer PRIMARY KEY,
     name varchar(255),
     email_id varchar(255) UNIQUE,
     department varchar(5),
     address text,
    UNIQUE(email_id, name)
)
```

B) Drop database – It is used to delete objects from a database.

drop database database_name

drop table table_name

C) Truncate – It is used to remove all records from a table.

D) Alter – It is used to alter(modify) the structure of a database. Using Alter command you can add the columns, modify the columns, delete the columns, and add or remove the constraint on table columns.

Query to add a new column to a table

```
ALTER TABLE Students ADD COLUMN college varchar(255) NOT NULL;
```

Query to remove any column from a table

```
ALTER TABLE Students DROP COLUMN age;
```

Query to modify the data type of column

```
ALTER TABLE Students MODIFY COLUMN roll_no integer;
```

Query to Add any constraint to a column

```
ALTER TABLE Students ADD CONSTRAINT stud_email
UNIQUE(email_id);
```

Query to drop constraint

```
ALTER TABLE Students DROP CONSTRAINT stud_email;
```

2) DML (Data Manipulation Language)

It is used for accessing and manipulating the data in a database table. the commands under DML are.

A) Insert Command

Using the insert command you can add the records to a table. There are two ways to use the Insert command. one is you want to insert values in every column as database design, and the second is you want to provide values for specific columns. The syntax of both ways is stated below.

```
INSERT INTO Students (column names) (Values); -- 1st sntax
INSERT INTO Students (Values)  -- 2nd syntax
```

You can also add multiple rows at a time.

```
INSERT INTO Students
(101, "ram", "ram@gm.com", "CS"),
```

(102, "shivam", "sh@gm.com", "IT");

B) Update Command

An update is used to update any value in database tables. Suppose I want to change the email id of a student whose roll number is 168.

```
Update Students SET email = "xyz@gmail.com" where roll_no = 168;
```

You can update multiple columns, rows at a time with any condition in the where clause.

C) Delete Command

Delete command is used to delete records from a table. we can delete a single record from the table using a conditional clause.

```
delete from students where roll_no = 28;
```

3) DCL (Data Control Language)

The commands under data control language help you to control the operations on the database. the commands under this are Grant and Revoke.

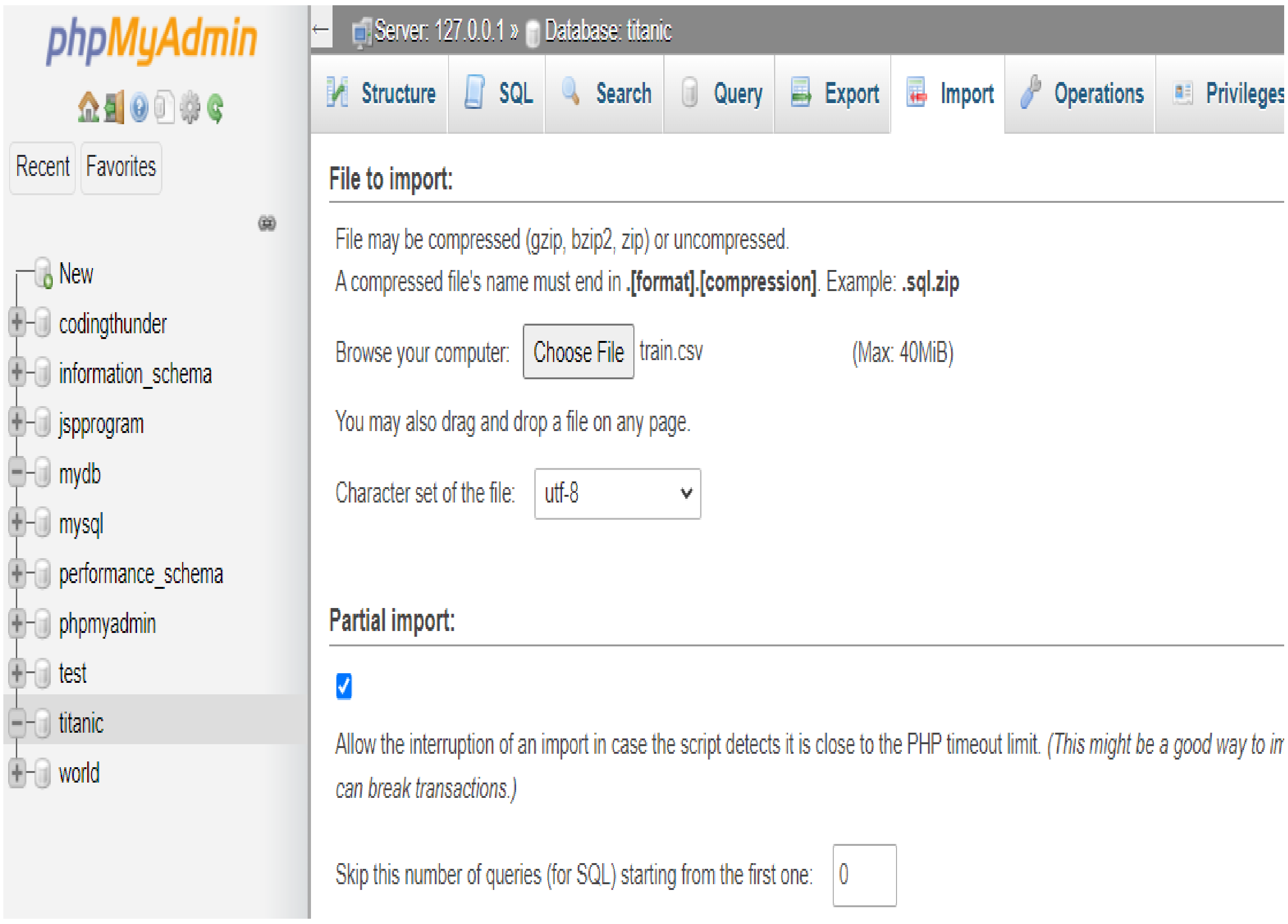
Grant – It is used to permit users to access the database

Revoke – It is used to take the given permission back.

Basic data analysis of Titanic dataset using SQL

Now we are excited to try some queries and datasets and understand how we can carry out data analysis in an easy way that somewhere challenges the Pandas data manipulation functions. We are using a very popular and familiar dataset known as the titanic survival dataset that can be easily found here and if you are not familiar with the dataset then you can read about it on that link as well. In short, The dataset is about the details of the passengers travelling in titanic when its accident happens and it includes details like personage, family details, survived or not. let us get started with running some SQL queries to understand how data analysis is done through SQL.

To load the dataset in XAMPP create a new database and go to the Import tab. And upload the downloaded train file in CSV format, and click Go. the query will execute and a new table named train will be created. the dashboard will look something like the below figures.



File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File train.csv (Max: 40MiB)

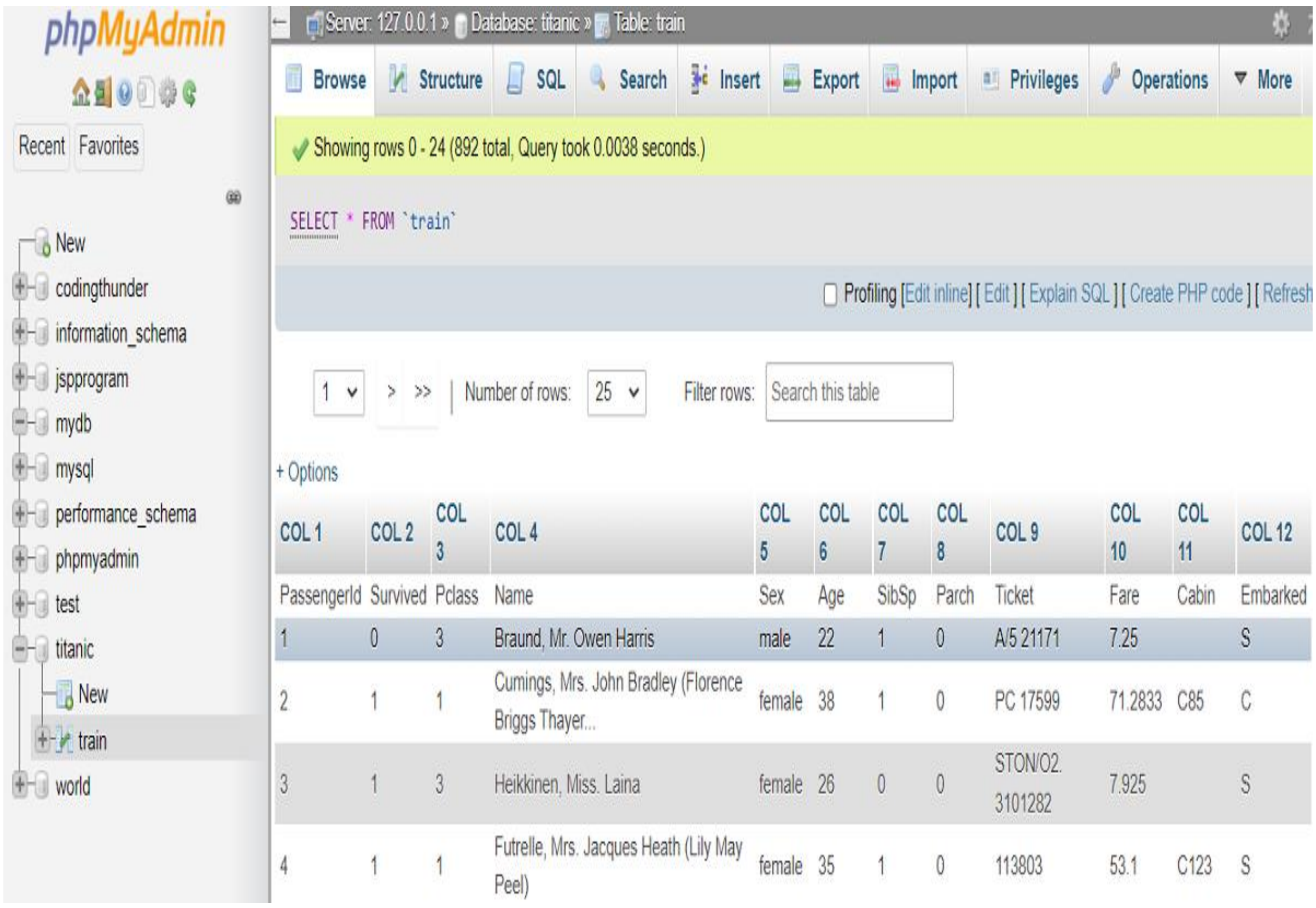
You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to im can break transactions.)*

Skip this number of queries (for SQL) starting from the first one: 0



Showing rows 0 - 24 (892 total, Query took 0.0038 seconds.)

```
SELECT * FROM `train`
```

COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7	COL 8	COL 9	COL 10	COL 11	COL 12
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer...	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S

1) How to retrieve all the records from DataFrame?

Select is a very powerful SQL statement used to retrieve any type of record. Asterisk sign means to get all the rows from the dataset.

SELECT * FROM train;

2) How to Filter Columns?

If we want to retrieve some specific columns rather than all the columns in data so use a select statement followed by name of columns to retrieve.

SELECT Name, Sex, Survived FROM train;

In the above query, it will result in only 3 columns but there will be all rows in the result.

3) How to give a temporary name to the column?

While retrieving the data sometimes we need to assign a column some specific name which is known as Alias. In SQL **AS** command is used to achieve this.

4) Expression

The expression means to retrieve the data by making some modifications in the values of a column. suppose I want to find the present age of people travelling on the titanic. so titanic accident happened in 1912 so if we ad 109 in age then it will be present age. here we have also used the AS keyword.

```
SELECT Name, (Age+109) AS curretntAge FROM train;
```

5) Constant

We can also create constant using SQL. suppose after the titanic accident Government declared a 1 Lakh compensation to all so we are required to add a new column and add a value in front of each name.

```
SELECT Name, 100000 AS Compensation FROM train;
```

6) Find unique details from data

we use a distinct keyword to find the unique categories from columns. suppose we have to find distinct classes in a ship.

```
SELECT Distinct Pclass FROM train;
```

And now we have to find all distinct class combinations with Embarked.

7) Filter the data based on condition

To filter the data based on condition SQL where clause is used in which except aggregate condition all other conditions are included like to extract data in a certain range, less than, greater than or equal to any value. we have to extract the data of passenger class three.

SELECT * FROM train where Pclass = 3;

Now If we want to know how many passengers in class three died. So to apply the two conditions together we use **AND** operator in the where clause.

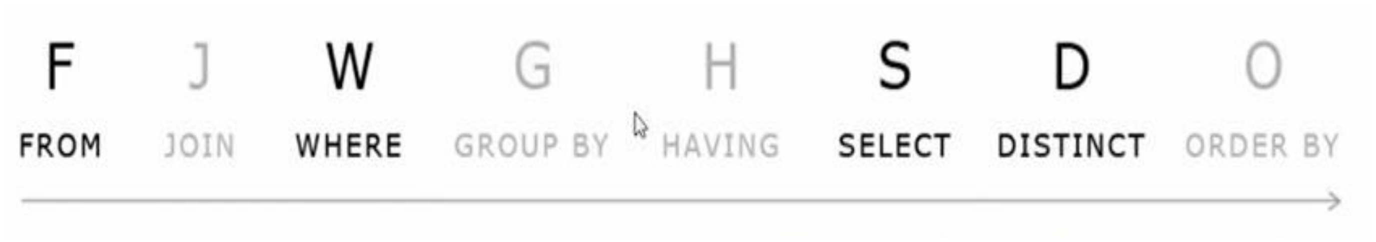
SELECT * FROM train where Pclass = 3 AND Survived = 0;

Similarly, if you have to check either of two conditions then we use **OR** operator in the where clause. Now we have to find how many passengers between the age of 10 to 18 were travelling so in range we use **BETWEEN** Keyword.

SELECT * FROM train where Age between 10 and 18;

Order of Query Execution

It is very important to understand how does query execution means which clause is executed first.



First From clause is executed, then if there is Join included in it is executed then query is sorted as per Where clause than group by clause is run which is always followed by having and then the columns which you want in a select statement is selected and then If you have to use Distinct or order by than result will be arranged.

Exploring More SQL operators to Play with Real-time Data

Now we will use a Bollywood movies dataset that you can easily find here. here many columns will help us understand how we can efficiently analyze the data using SQL queries.

1) Find all the movies whose Genre is Horror or Comedy?

We have seen that we use or operator to match multiple conditions to get data if any one condition is true.

```
SELECT title, genre, FROM movies WHERE genre = "horror" OR genre =
"comedy";
```

But this query becomes complex when values are more so when we have to retrieve the data by matching values to a single column, we use the **in** operator in this case.

```
SELECT title, genre, FROM movies WHERE genre in ("horror", "comedy");
```

2) Retrieving Data using Wildcards

Wildcards are used to substitute one or more characters in a string. Wildcard characters are used with the Like operator. It is used in where clause to match a particular string in a column value

- Like word% – It means retrieving the data where the value starts with a word.
- Like %word – retrieve the data where the value ends with word.
- Like %word% – retrieve data where a word is present anywhere in value.
- Like _word% – Find values that have a word at second position
- Like word__% – Find values that start with a word and are at least 3 characters in length.

Now using this you can implement various queries and different combinations using wildcards. And you can use underscore to match the length of value. Now, let us practice this by some experimentation. I want to retrieve the data where the movie title starts with character A.

```
SELECT title FROM movies WHERE title LIKE A%;
```

Now we want to find the name of Khan or Kapoor actors

3) Playing with SQL Functions

There are different in-built functions in SQL like python to retrieve the data. we will understand and practice each fraction on our dataset.

i) Abs – Abs stands for absolute which is used to get an absolute integer value.

ii) round – It rounds the decimal or float to integer or in defined decimal places. like we want to extract the movie runtime in hours and round to two decimal places.

```
SELECT title, round(runtime/60, 2) as runtime_hrs from movies;
```

iii) Ceil – It rounds the decimal value to the upper integer. For example, for 2.4 it gives the result as 3.

iv) floor – opposite of ceil which rounds the decimal value to lowest near integer. for value 2.7 it will result in 2.

v) Upper and lower – Both function is used on strings to convert in uppercase or lowercase.

vi) concat – It helps in concatenating two strings together in a single column. like we want to express actor and director in a single column.

```
SELECT concat(actor, " ", director) AS crew FROM movies;
```

vii) length – It is used to find the length of a string. like we have to find the length of each title.

```
SELECT title, length(title) AS length FROM movies;
```

viii) substring – It is used to find the specific part of a specific length in any string. In this function, you have to specify the string, starting index, and length of the substring you want.

```
Syntax - substr(string, start(default-1), length)
SELECT title, substr(title, 3, 5) AS short_chr FROM movies;
```

Aggregated Functions

i) minimum – It is used to find the minimum value.

```
SELECT min(india_gross) FROM movies;
```

ii) maximum – It is used to find the maximum value in a column.

```
SELECT max(india_gross) FROM movies;
```

iii) sum – Find the sum of all the values in a column.

```
SELECT sum(india_gross) FROM movies;
```

iv) Average – Find the mean value of the column. we have to find the average income of Indian movies.

```
SELECT avg(india_gross) FROM movies;
```

V) count – It counts the number of values in a column.

```
SELECT count(DISTINCT(actor)) FROM movies;
```

4) Sorting Data

Most of the time we want data to be sorted according to certain columns in the dataset. Order By clause is used to sort the data in ascending or descending order. Ascending is a default order and to sort in descending you need to specify using the DESC keyword.

We have to find the movies ordered by profit in ascending and in descending order.

```
SELECT title, (worldwide_gross - budget) AS profit FROM movies ORDER
BY profit;  -- ascending order
```

```
SELECT title, (worldwide_gross - budget) AS profit FROM movies ORDER
BY profit DESC;  -- descending order
```

Now if someone asks to only extract Top five movies. so In this case we use a LIMIT keyword to limit the output records.

```
SELECT title, (worldwide_gross - budget) AS profit FROM movies ORDER
BY profit ASC LIMIT 5;
```

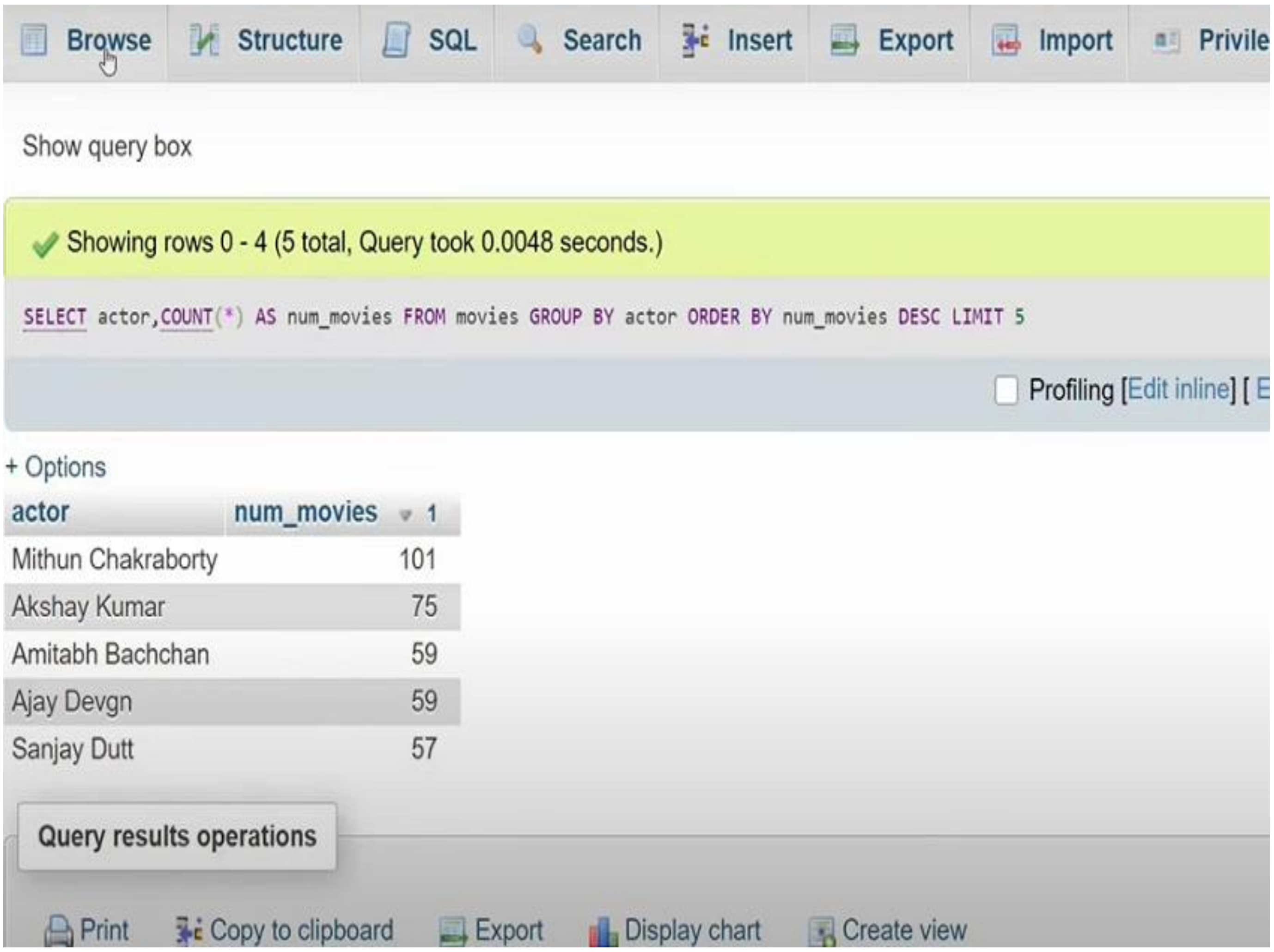
We can also use order by clause on multiple columns. suppose we want to get the data sorted according to a genre in each genre It should be sorted according to the movie title.

```
SELECT * FROM movies ORDER BY genre, title;
```

5) Grouping Data

Grouping Data is a very strong concept and is used in many different cases. On basis of any column, we create groups of distinct values in it, and based on each group we perform some operation. We have to find the top 5 actors who have done the maximum number of movies.

```
SELECT actor, COUNT(*) AS num_movies FROM movies GROUP BY actor
ORDER BY num_movies DESC LIMIT 5;
```



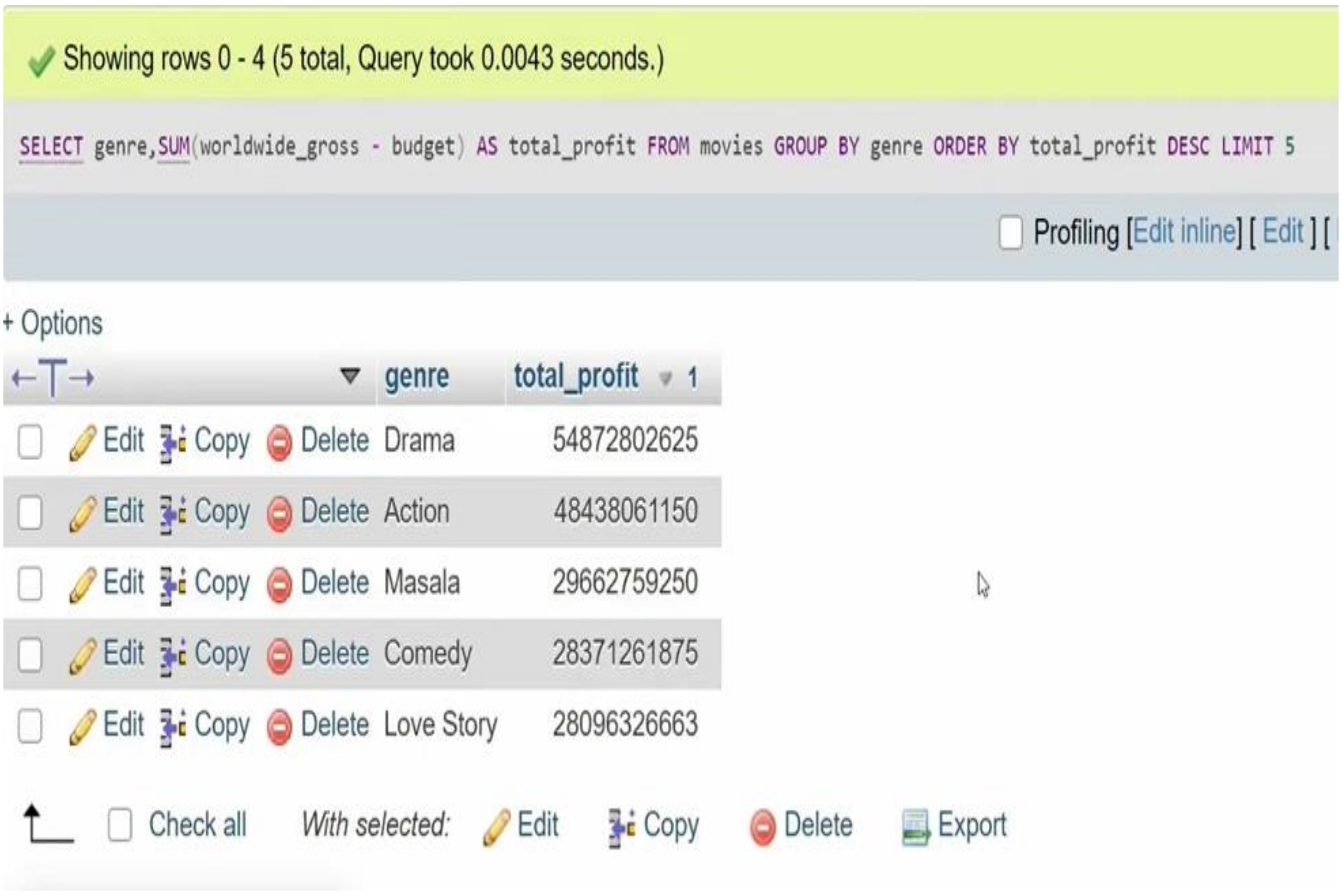
Showing rows 0 - 4 (5 total, Query took 0.0048 seconds.)

```
SELECT actor,COUNT(*) AS num_movies FROM movies GROUP BY actor ORDER BY num_movies DESC LIMIT 5
```

actor	num_movies
Mithun Chakraborty	101
Akshay Kumar	75
Amitabh Bachchan	59
Ajay Devgn	59
Sanjay Dutt	57

Write a SQL query to find the Genre whose movies are most profitable? The question wants you to find the sum of profit of each movie in each Genre and output the genre with maximum profit.

SELECT genre, SUM(worldwide_gross - budget) AS total_profit FROM movies GROUP BY genre ORDER BY total_profit DESC LIMIT 5;



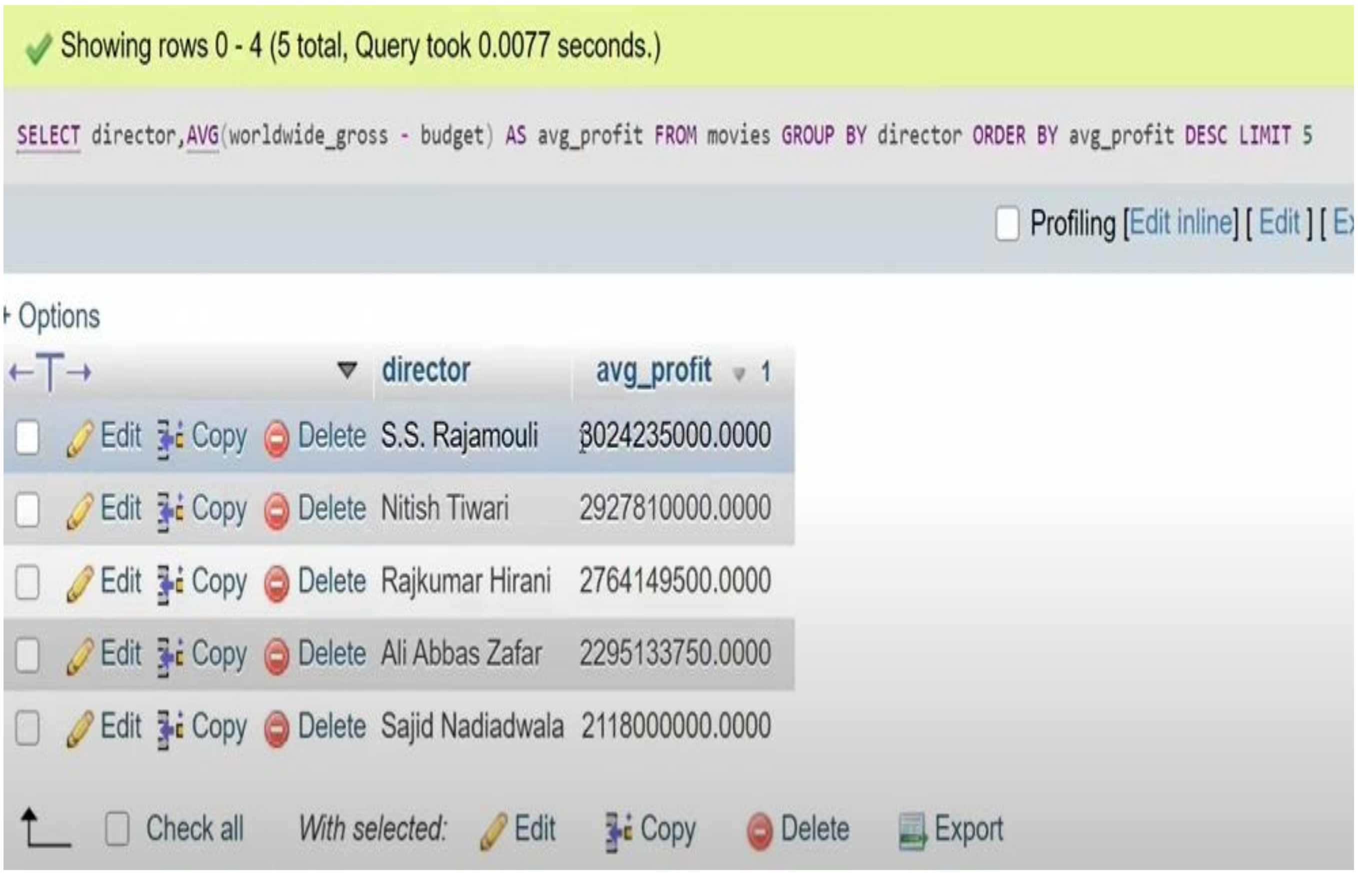
✓ Showing rows 0 - 4 (5 total, Query took 0.0043 seconds.)

```
SELECT genre,SUM(worldwide_gross - budget) AS total_profit FROM movies GROUP BY genre ORDER BY total_profit DESC LIMIT 5
```

genre	total_profit
Drama	54872802625
Action	48438061150
Masala	29662759250
Comedy	28371261875
Love Story	28096326663

Another question may be like find on an average which director movies make a maximum profit?

SELECT director, AVG(worldwide_gross - budget) AS avg_profit FROM movies GROUP BY director ORDER BY avg_profit DESC LIMIT 5;



Showing rows 0 - 4 (5 total, Query took 0.0077 seconds.)

```
SELECT director,AVG(worldwide_gross - budget) AS avg_profit FROM movies GROUP BY director ORDER BY avg_profit DESC LIMIT 5
```

director	avg_profit
S.S. Rajamouli	3024235000.0000
Nitish Tiwari	2927810000.0000
Rajkumar Hirani	2764149500.0000
Ali Abbas Zafar	2295133750.0000
Sajid Nadiadwala	2118000000.0000

Let’s take a question on two columns Group by. Write a SQL query to find the combination of actor and director that has earned maximum profit?

SELECT actor, director, SUM(worldwide_gross - budget) AS profit from movies GROUP BY actor, director ORDER BY profit DESC LIMIT 5;

Here are some questions for you to practice and answer in the comment section below.

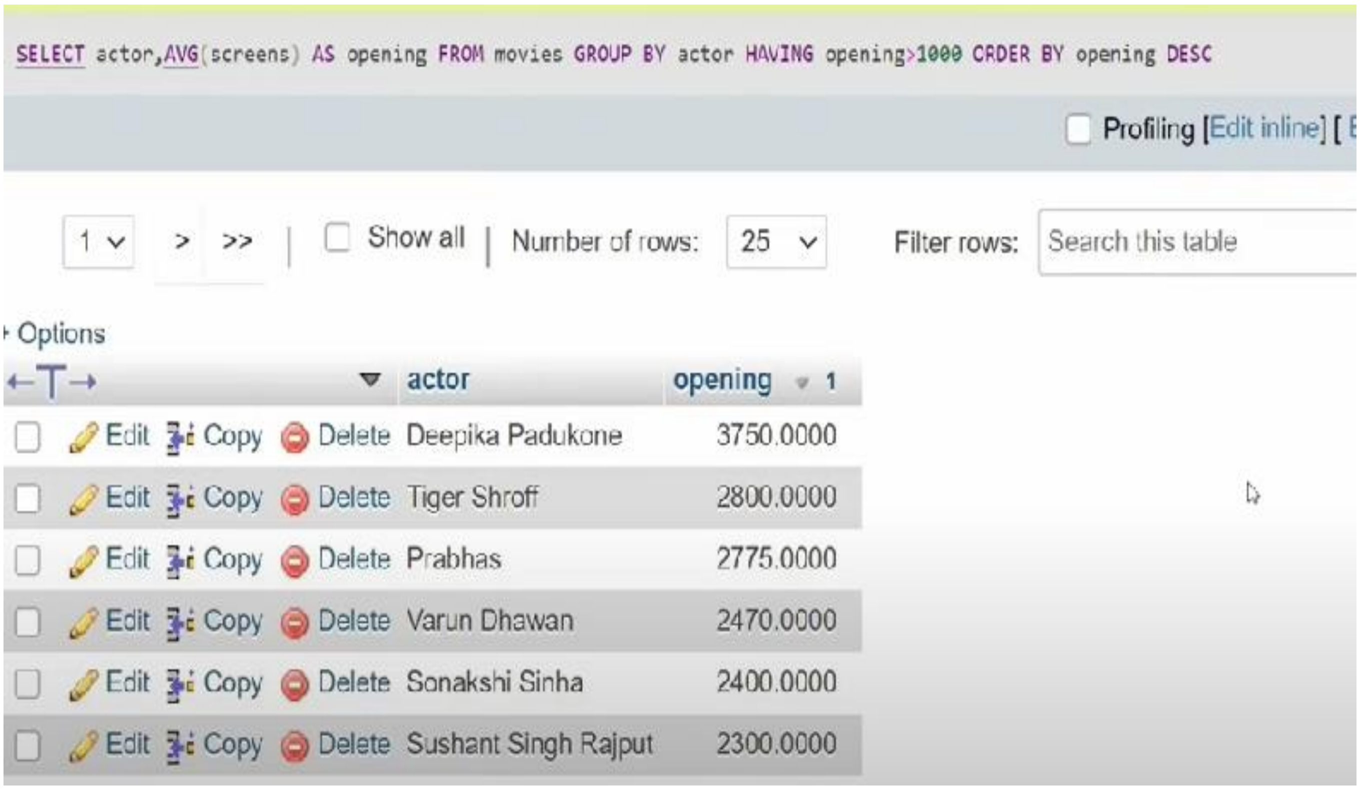
1. Write a SQL query to find the top five movies with a maximum budget using group by clause.
2. write a SQL query to find the top ten actors whose movies have made a maximum profit
3. Write a query to find which actor earns maximum in which Genre

This is the power of the Group by clause. You can carry lots of analysis using Group by and aggregated functions. It gets more power when the Having clause is used.

6) Filter Grouped Data using Having Clause

Having Clause is used to filter the Group By output which is the same as where clause which is used to filter Select statement output. For example, the question is to write a query to find the names of actors whose movies on average get more than 1000 screens.

SELECT actor, AVG(screen) AS opening FROM movies
GROUP BY actor HAVING opening > 1000
ORDER BY opening DESC;



```
SELECT actor,AVG(screens) AS opening FROM movies GROUP BY actor HAVING opening>1000 ORDER BY opening DESC
```

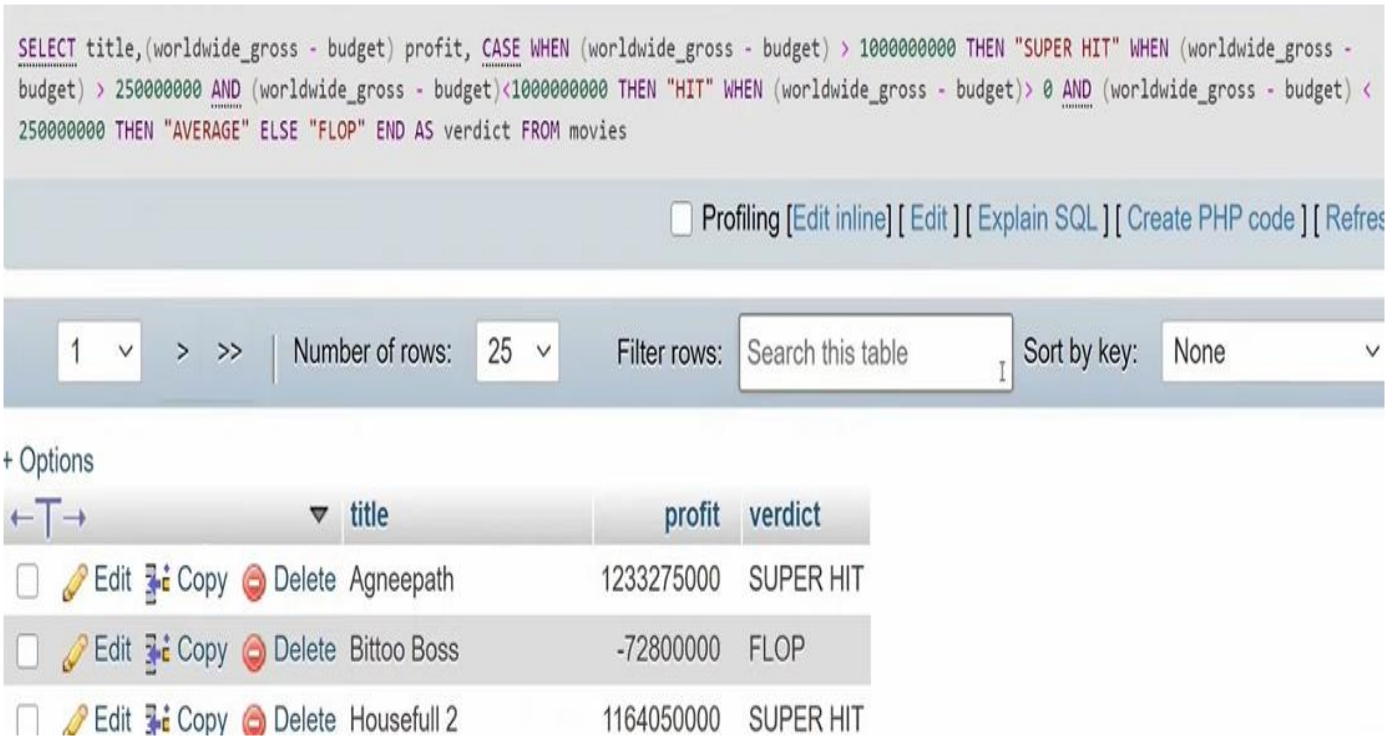
actor	opening
Deepika Padukone	3750.0000
Tiger Shroff	2800.0000
Prabhas	2775.0000
Varun Dhawan	2470.0000
Sonakshi Sinha	2400.0000
Sushant Singh Rajput	2300.0000

Having is always used after Group by and where cannot be used with any aggregated functions.

7) CASE Statements

Case statements are like If-Else statements for databases. It is the same as the control statement in any programming language. suppose we have to classify each movie in four classes according to profit in such a way that If profit is negative then it is a Flop movie, profit between 0 and 25 crores is an average movie, profit between 25 and 100 crores are Hit movies and movies with profit above 100 crores is superhit.

```
SELECT title, (worldwide_gross - budget) AS profit,
CASE
    WHEN profit > 100000000 THEN "Super-HIT"
    WHEN profit > 250000000 AND profit < 100000000 THEN "HIT"
    WHEN profit > 0 AND profit < 250000000 THEN "Average"
    ELSE "FLOP"
END AS verdict
FROM movies;
```



```
SELECT title,(worldwide_gross - budget) profit, CASE WHEN (worldwide_gross - budget) > 1000000000 THEN "SUPER HIT" WHEN (worldwide_gross - budget) > 250000000 AND (worldwide_gross - budget)<1000000000 THEN "HIT" WHEN (worldwide_gross - budget)> 0 AND (worldwide_gross - budget) < 250000000 THEN "AVERAGE" ELSE "FLOP" END AS verdict FROM movies
```

title	profit	verdict
Agneepath	1233275000	SUPER HIT
Bittoo Boss	-72800000	FLOP
Housefull 2	1164050000	SUPER HIT

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refres
1
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Edit Copy Delete

8) SQL Joins

I) inner join – When we perform an inner join between two tables then the resultant table will include the results which are common in both tables no which joining is done.

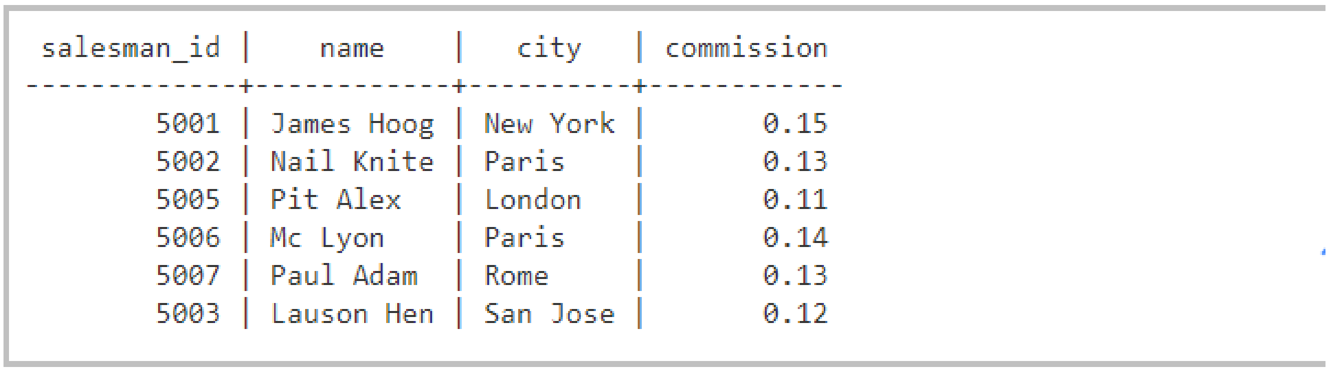
ii) left outer join – The results of the left outer join include all the rows of the left side table and only matching rows of a right side table.

iii) right outer join – The results of the right outer join include all the rows of the right side table and only matching rows of a left-side table.

iv) cross join – cross join means again each record combine it will every other record from another table.

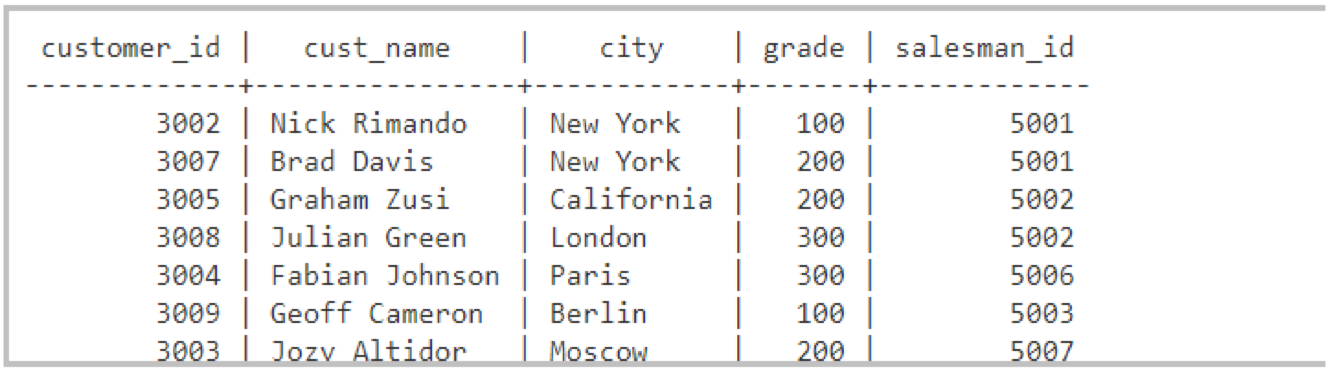
Let us practice and see how joins help to conduct data analysis between two data frames(two tables). suppose we have two tables as customers and salesman whose dummy view is shown below.

Sample table: salesman



```
 salesman_id |    name    |   city   | commission
-------------+------------+----------+------------
        5001 | James Hoog | New York |       0.15
        5002 | Nail Knite | Paris    |       0.13
        5005 | Pit Alex   | London   |       0.11
        5006 | Mc Lyon    | Paris    |       0.14
        5007 | Paul Adam  | Rome     |       0.13
        5003 | Lauson Hen | San Jose |       0.12
```

Sample table: customer



```
 customer_id |   cust_name    |    city    | grade | salesman_id
-------------+----------------+------------+-------+-------------
        3002 | Nick Rimando   | New York   |   100 |        5001
        3007 | Brad Davis     | New York   |   200 |        5001
        3005 | Graham Zusi    | California |   200 |        5002
        3008 | Julian Green   | London     |   300 |        5002
        3004 | Fabian Johnson | Paris      |   300 |        5006
        3009 | Geoff Cameron  | Berlin     |   100 |        5003
        3003 | Jozy Altidor   | Moscow     |   200 |        5007
```

let's take questions to analyze so first we have to find the salesperson and customer who belongs to the same city. so this can be solved with join and without join as well so let us see how we can do it without using joins.

SELECT salesman.name AS "Salesman",

```
customer.cust_name, customer.city
FROM salesman,customer
WHERE salesman.city=customer.city;
```

This type of query is sometimes also known as advance where clause in some context. now let us see how we can do it using joins.

```
SELECT s.name AS "Salesman",  c.cust_name, c.city from salesman s INNER
JOIN customer c on c.city = s.city;
```

here we have given an Alias to table name with the character name.

9) Unions

It is used to concatenate the results of two queries. I have to find IDs from users and group IDs from groups. Now some may have questioned what is the difference between joining and union so let us study some differences between Union and join.

- When we perform join between two tables then both tables can have different columns of different data types but in union number of columns and order of columns should be the same.
- Union joins the two tables vertically while join combines two tables horizontally.

have a look at the below union query example how it combines and retrieves the results of two tables.

Sample table: product

```
PROD_CODE  PROD_NAME       COM_NAME         LIFE
---------- --------------- ---------- ----------
PR001      T.V.            SONY                7
PR002      DVD PLAYER      LG                  9
PR003      IPOD            PHILIPS             9
PR004      SOUND SYSTEM    SONY                8
PR005      MOBILE          NOKIA               6
```

Sample table: purchase

```
    PUR_NO PROD_CODE  PROD_NAME       COM_NAME      PUR_QTY PUR_AMOUNT
---------- ---------- --------------- ---------- ---------- ----------
         2 PR001      T.V.            SONY               15     450000
         1 PR003      IPOD            PHILIPS            20      60000
         3 PR007      LAPTOP          H.P.                6     240000
         4 PR005      MOBILE          NOKIA             100     300000
         5 PR002      DVD PLAYER      LG                 10      30000
         6 PR006      SOUND SYSTEM    CREATIVE            8      40000
```

```
SELECT prod_code,prod_name FROM product
UNION
SELECT prod_code,prod_name FROM purchase;
```

The output of the above query will be like the below figure.

Output:

```
PROD_CODE  PROD_NAME
---------- ---------------
PR001      T.V.
PR002      DVD PLAYER
PR003      IPOD
PR004      SOUND SYSTEM
PR005      MOBILE
PR006      SOUND SYSTEM
PR007      LAPTOP
```

10) Nested Queries Or Subqueries

subquery means query inside a query is known as nested queries likewise we use nested if-else statement. The first inner query will be run and after that outer query which is also known as the main query will execute. The subquery can be written in any SQL clauses like From, where, Having.

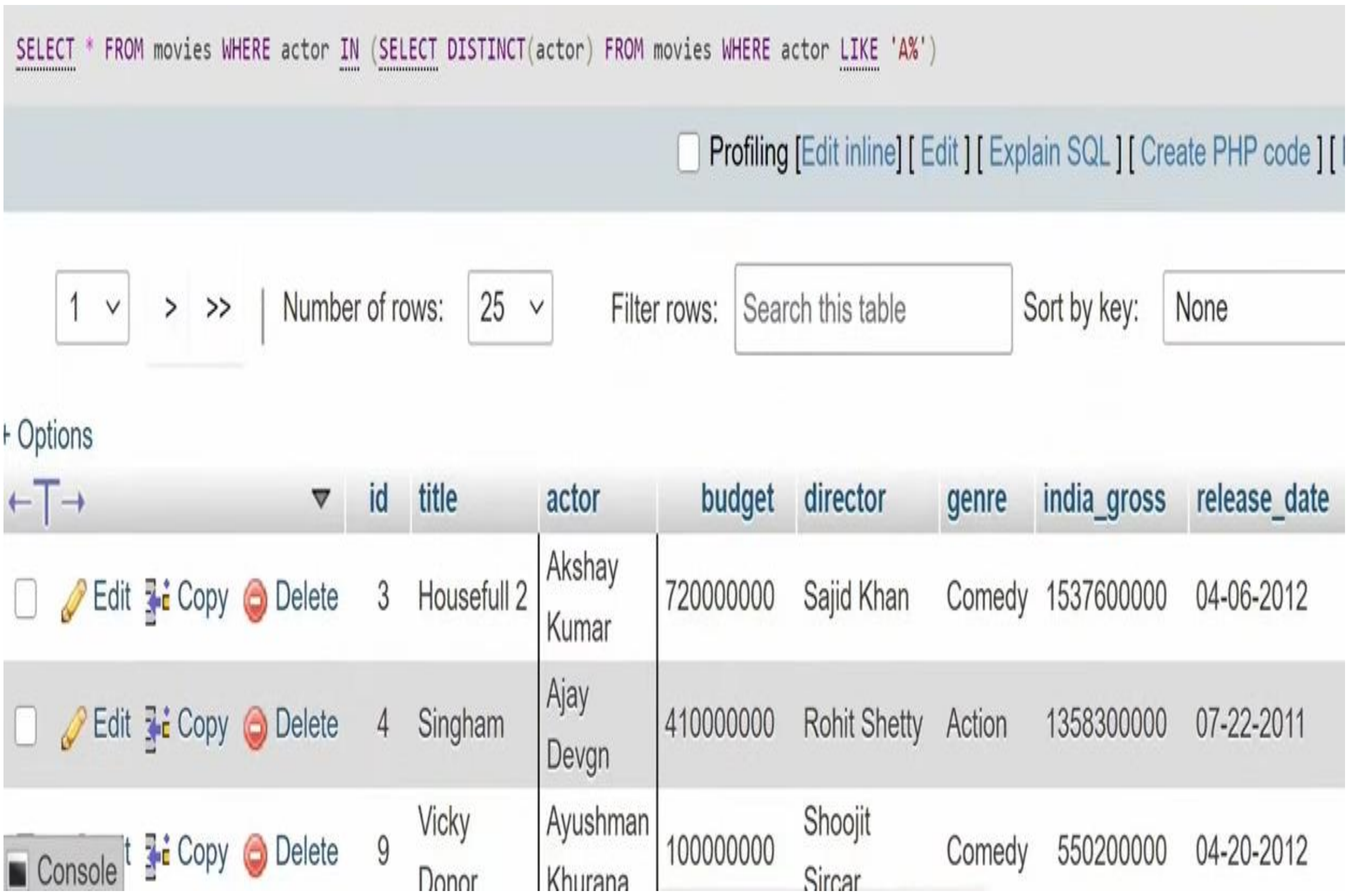
i) We have to find the movie whose budget is maximum.

```
SELECT title FROM movies WHERE budget = (SELECT MAX(budget)
FROM movies);
```

The inner query may result in multiple rows so in that case, we use the **in** operator instead of the assignment operator.

ii) Write a query to find all the movies of actors whose name starts with character A?

```
SELECT * FROM movies WHERE actor IN (SELECT Distinct(actor) from
movies WHERE actor like "A%");
```



```
SELECT * FROM movies WHERE actor IN (SELECT DISTINCT(actor) FROM movies WHERE actor LIKE 'A%')
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [
1
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Console

	id	title	actor	budget	director	genre	india_gross	release_date
Edit Copy Delete	3	Housefull 2	Akshay Kumar	720000000	Sajid Khan	Comedy	1537600000	04-06-2012
Edit Copy Delete	4	Singham	Ajay Devgn	410000000	Rohit Shetty	Action	1358300000	07-22-2011
Copy Delete	9	Vicky Donor	Ayushman Khurana	100000000	Shoojit Sircar	Comedy	550200000	04-20-2012

iii) write a query to find all the movies of the top 5 actors who have earned maximum profit using subquery?

This is a little bit tricky question where we have to use a subquery as well as a group by clause.

SELECT * FROM movies WHERE actor IN (

SELECT actor FROM (

SELECT actor, (worldwise_gross-budget) AS profit FROM movies GROUP BY actor ORDER BY profit DESC LIMIT 5

) A

);

This is an independent subquery that means the inner query is independent to run without interfering with any value with the main query.

iv) write a query to find the most profitable movie for each Genre?

```
SELECT title, genre, (worldwise_gross - budget) AS profit FROM movies m1
WHERE (worldwise_gross - budget) =
(SELECT MAX(worldwise_gross - budget) FROM movies m2
where m2.genre = m1.genre);
```

This is a correlated query that means the inner query is dependent on the main query.