Google console for run SQL Commands : <https://console.cloud.google.com/bigquery?project=maximal-centaur-340003>

My SQL : <https://youtu.be/p3qvj9hO_Bo>

SELECT \* FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality` LIMIT 1000

**DISTINCT will select all different city names.**

SELECT DISTINCT City FROM customers;

SELECT \* FROM customers ORDER BY City;

**Set the value of the City columns to 'Oslo',to only ones where the Country column has the value "Norway".**

UPDATE Customers SET city = `Oslo` WHERE Country = `Norway`;

**This will delete all the rows where country is Norway.**

DELETE FROM Customers WHERE Country = `Norway`;

**Order by will short your column in ascending order from a to Z**

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

ORDER BY Births

LIMIT 10

**Asc is a function to set order from lowest to highest**

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

ORDER BY Births

ASC

LIMIT 10

**Desc is a function to set order from highest to lowest**

SELECT \*

FROM `bigquery-public-data.sdoh\_cdc\_wonder\_natality.county\_natality`

WHERE Year = '2018-01-01'

ORDER BY Births

DESC

LIMIT 10

SELECT \*

FROM `Mobie\_data.Movies`

WHERE Genre = "Comedy" AND Revenue > 300000000

ORDER BY Release\_Date

DESC

SELECT \* FROM Customers

WHERE Country='Germany' AND City='Berlin';

SELECT \* FROM Customers

WHERE City='Berlin' OR City='Brampton';

**INSERT will insert the data into database columns.**

INSERT INTO `customers`(`Name`,`City`,`PostalCode`)

VALUES('Riddhi','Navasari','346789');

DELETE FROM Customers WHERE Name='chirag';

**SELECT TOP will select the first 3 rows from database.**

SELECT TOP 3 \* FROM Customers;

**Limit will select the first 4 rows from database.**

SELECT \* FROM Customers LIMIT 4;

**MIN will finds the price of the cheapest product.**

SELECT MIN(Price) AS SmallestPrice

FROM Products;

**MAX finds the most value from column.**

SELECT MAX(Price) AS LargestPrice

FROM Products;

**COUNT will count the number of items in column**

SELECT COUNT(ProductID) FROM Products;

**SUM will do the summation of whole price column**

SELECT SUM(price) FROM Products

D**ROP will remove the column from Database**

ALTER TABLE customers DROP COLUMN Fax;

**Like %p will select all products with a ProductName ending with "p":**

SELECT \* FROM Products

WHERE ProductName like "%p"

**Like p% will select all products with a ProductName starting with "p":**

SELECT \* FROM Products

WHERE ProductName like "p%"

**Like %or% will selects all customers with a CustomerName that have "or" in any position**

SELECT \* FROM Products

WHERE ProductName LIKE '%or%'



**Like a%o will selects all customers with a ContactName that starts with "a" and ends with "o":**

SELECT \* FROM Customers

WHERE ContactName LIKE 'a%o';

**NOT Like a% will selects all customers with a CustomerName that does NOT start with "a"**

SELECT \* FROM Customers

WHERE CustomerName NOT LIKE 'a%';

**Like ber% will select all Customers with a CityName starting with Ber.**

SELECT \* FROM Customers

WHERE City LIKE 'ber%';

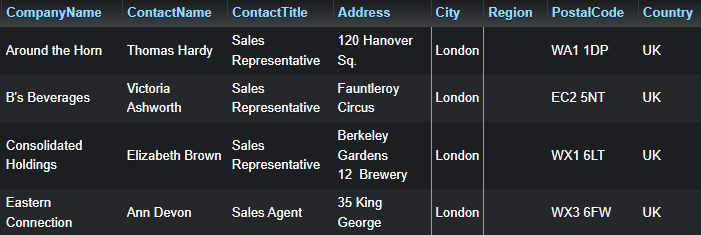


**Like %es% will select all customers that have es in their CityName.**

SELECT \* FROM customers where city LIKE '%es%';

**The following SQL statement selects all customers with a City starting with "L", followed by any character, followed by "n", followed by any character, followed by "on"**

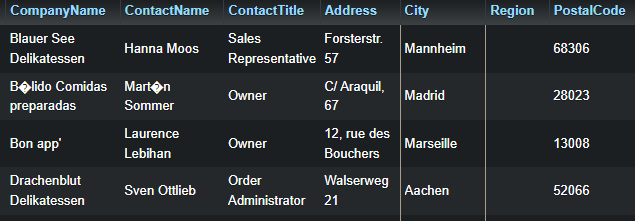
SELECT \* FROM Customers WHERE City LIKE 'L\_n\_on';



**This will select all the customers with a CityName where the second letter of a city is a.**

SELECT \* FROM Customers

WHERE City LIKE '\_a%';



**The following SQL statement selects all customers that are located in "Germany", "France" or "UK":**

SELECT \* FROM Customers

WHERE Country IN ('London','Paris');

**The following SQL statement selects all City with a CityName between London and Paris.**

SELECT \* FROM `customers`

WHERE City BETWEEN 'London' and 'Paris';

**The following SQL statement selects all City with a CityName Not between London and Paris.**

SELECT \* FROM customers

WHERE City NOT BETWEEN 'London' AND 'Paris';

**BETWEEN and NOT BETWEEN more examples :** <https://www.w3schools.com/sql/sql_between.asp#:~:text=The%20following%20SQL%20statement%20selects%20all%20orders%20with%20an%20OrderDate%20between%20%2701%2DJuly%2D1996%27%20and%20%2731%2DJuly%2D1996%27%3A>

**The following SQL statement creates two aliases, one for the CustomerID column and one for the ContactName. This will create two separate column with name ID and CN respectively.**

SELECT CustomerID AS ID, ContactName AS CN FROM customers;

**CONCAT command will use for merge rows.**

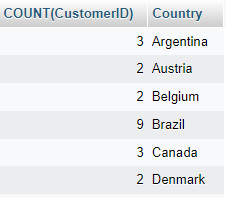
SELECT ContactName, CONCAT (Address , ' ' , City , ',' , Region , ',' , Country, ' ', PostalCode) AS FullAddress FROM customers;

**IN Operator will select all the records where Country is either "Norway" or "France".**

SELECT \* FROM Customers WHERE Country IN ('Norway','France');

**GROUP BY will count the customers in each country.**

SELECT COUNT(CustomerID),Country FROM Customers GROUP BY Country;



**DROP Command will delete the database.**

DROP DATABASE Customers;

DROP TABLE Persons;

**TRUNCATE Command will delete all the records from database.**

TRUNCATE TABLE persons;

**This Command will Add a column of type DATE called Birthday.**

ALTER TABLE Persons

ADD Birthday DATE;

**This Command will Delete a column from Database.**

ALTER TABLE Persons

DROP COLUMN Birthday;

**This command will change company name to agile where country is Germany.**

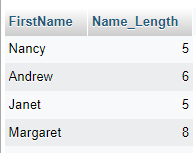
UPDATE customers SET CompanyName = "Agile" WHERE Country = "Germany";

**This command will remove the rows from database where Region value is NULL.**

DELETE FROM employees WHERE Region IS NULL;

**This command will return name length of all name in Name\_Length column.**

SELECT FirstName, character\_length(FirstName) as Name\_Length FROM employees;



**This will return full name in new column.**

SELECT FirstName,LastName, CONCAT(FirstName, " ", LastName) AS Full\_Name FROM employees;

**This command will add a new column named FullName and then it will return full name in new column.**

ALTER TABLE employees ADD FullName varchar(20);

UPDATE employees SET FullName = CONCAT(FirstName, " ", LastName);

**This query will select city where salary is greater than 2000.**

SELECT City, salary FROM Employees GROUP BY city HAVING salary > 2000;

**This query will perform Inner Join function on two table cricket and football.**

SELECT \* FROM

cricket as c inner join football as f

ON c.name = f.name

**SQL SubQuery : This Query will select all the employees whose salary is greater than average salary of all employees.**

SELECT name, salary

FROM employees WHERE salary > (select avg(salary) from employees)

SELECT \* FROM employees WHERE salary > (SELECT salary FROM employees WHERE name = “John”);

**Trigger in SQL: This SQL query will set the student marks to 50 in each row if student marks less than 0 while insert data**

DELIMITER //

create trigger marks\_verify

before INSERT ON students

for each ROW

if new.marks < 0 then set new.marks = 50;

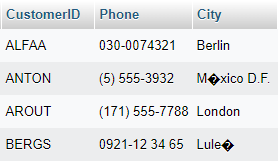
end if; //

**Views in SQL: This SQL query will create new table as cust\_details from customers table.**

CREATE VIEW cust\_detils AS

SELECT `CustomerID`,`Phone`,`City`

FROM customers;



**This query will change the name of View to different name.**

RENAME table cust\_details TO customer\_details;

**This query will delete a view.**

DROP view customer\_details

**This query will perform Inner Join on two different tables.**

SELECT \* FROM employees as e INNER JOIN salaries as s

on e.emp\_no = s.emp\_no;

**SQL Inner Join query on employees table;**

SELECT employees.emp\_no, employees.first\_name, employees.last\_name, salaries.salary

FROM employees INNER JOIN salaries ON employees.emp\_no = salaries.emp\_no GROUP BY employees.emp\_no;

**This query will display the all employees information except marketing.**

SELECT \* FROM rmployees WHERE departmrnt != ‘marketing’;

**This Query will display all the employees who hired between these dates.**

SELECT \* FROM employees WHERE hire\_date > '1985-02-25' AND hire\_date < '1989-02-25' ORDER BY hire\_date ASC;

This Query will display all the employees whose emp\_no is even.

SELECT \* FROM `employees` WHERE emp\_no % 2 = 0;