(SELECT COUNT(CITY) – COUNT(DISTINCT CITY) FROM TABLE WHERE ID%2 = 0;

SELECT CITY, LENGTH(CITY) FROM STATION ORDER BY LENGTH(CITY) ASC, CITY

LIMIT 1

)UNION(…);

SELECT DISTINCT CITY FROM STATION WHERE LEFT(CITY,1) IN (‘A’, ‘E’); //START WITH

//IF ENDS WITH “LEFT” -> “RIGHT”

//IF BOTH = LEFT() IN () AND RIGHT() IN ()

//IF DOES NOT START WITH, IN -> NOT IN

SELECT Name FROM Students WHERE Marks >75 ORDER BY RIGHT(NAME,3), Id; //Order by last three characters, if same, secondary sort them by asc id

SELECT

CASE

WHEN A = B AND B = C then “Eq”

ELSE “NOT EQ”

END

FROM Tri

SELECT CONCAT(Name, ‘(‘, LEFT(Occupation, 1) , ’)’ ) FROM Occupations ORDER BY Name;

UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
WHERE CustomerID = 1; //Be careful when updating records. If you omit the WHERE clause, ALL records will be updated!

DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

SELECT TOP 3 \* FROM Customers;

The following SQL statement shows the equivalent example for MySQL:

SELECT \* FROM Customers  
LIMIT 3;

SELECT MIN(Price) AS SmallestPrice  
FROM Products;

The SQL COUNT(), AVG() and SUM() Functions

SELECT \* FROM Customers   
WHERE CustomerName LIKE 'a%';

//https://www.w3schools.com/sql/sql\_like.asp

//https://www.w3schools.com/sql/sql\_wildcards.asp (LIKE)

SELECT \* FROM Customers

WHERE Country IN ('Germany', 'France', 'UK');

SELECT \* FROM Products  
WHERE Price BETWEEN 10 AND 20;

SELECT CustomerID AS ID, CustomerName AS Customer

FROM Customers;

// JOIN : <https://www.diffen.com/difference/Inner_Join_vs_Outer_Join>

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns. // 그룹 된 값이 갖고 있는 갯수에 따라

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

SELECT OrderID, Quantity,

CASE WHEN Quantity > 30 THEN 'The quantity is greater than 30'

WHEN Quantity = 30 THEN 'The quantity is 30'

ELSE 'The quantity is under 30'

END AS QuantityText

FROM OrderDetails;

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5  
ORDER BY COUNT(CustomerID) DESC;

The EXISTS operator is used to test for the existence of any record in a subquery.

The EXISTS operator returns TRUE if the subquery returns one or more records.

SELECT SupplierName  
FROM Suppliers  
WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price = 22);

SELECT Customers.CustomerName, Orders.OrderID  
INTO CustomersOrderBackup2017 //새로 만듬  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

--Select all: // 주석

//Operators: https://www.w3schools.com/sql/sql\_operators.asp