SQL

**Select query:**

Select \* from customers;

**Distinct:**

Select distinct country from customers;

**List number of different (distinct) customer countries:**

Select count(distinct country) from customers;

**Customers from the country "Mexico":**

Select \* from Customers where country = ‘Mexico’;

**Numeric selection:**

Select \* from customers where customerid=1;

**selects all fields from "Customers" where country is "Germany" AND city is "Berlin":**

Select \* from customers where country = ‘Germany’ and city = ‘Berlin’;

**selects all fields from "Customers" where country is "Germany" OR "Spain":**

Select \* from customers where country = ‘Germany’ or country = ‘Spain’;

**selects all customers from the "Customers" table, sorted by the "Country" column:**

Select \* from customers order by country;

**INSERT:**

Insert into customers (customername,contactname) values(‘John’,’Doe’);

**IS NULL:**

Select \* from customers where address is null;

**UPDATE:**

Update customers set contactname = ‘Double’ where customername=’John’;

**DELETE:**

Delete from customers where country = ‘Germany’;

**MIN and MAX:**

Select min(price) from products;

Select max(price) from products;

Count(), Avg() and sum()

Select sum(price) from products;

**LIKE**

Select \* from customers where customername like ‘A%’;

**IN**

Select \* from customers where customername in (‘John’,’Jay’,’Ashley’,’Sara’);

**BETWEEN**

Select \* from Products where price between 10 and 20;

**AS**

SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' +Country AS Address  
FROM Customers;

**JOIN**

SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
ORDER BY Customers.CustomerName;

SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders  
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID  
ORDER BY Orders.OrderID;

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID  
ORDER BY Customers.CustomerName;

SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City  
FROM Customers A, Customers B  
WHERE A.CustomerID <> B.CustomerID  
AND A.City = B.City   
ORDER BY A.City;

**UNION**

SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;

**UNION ALL ( Allows duplicates )**

SELECT City FROM Customers  
UNION ALL  
SELECT City FROM Suppliers  
ORDER BY City;

**Group By**

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

**Having**

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

**EXISTS**

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

**ANY**

SELECT ProductName  
FROM Products  
WHERE ProductID = ANY  
  (SELECT ProductID  
  FROM OrderDetails  
  WHERE Quantity = 10);

**ALL**

SELECT ProductName  
FROM Products  
WHERE ProductID = ALL  
  (SELECT ProductID  
  FROM OrderDetails  
  WHERE Quantity = 10);

**SELECT INTO ( Backup table )**

SELECT \* INTO CustomersGermany  
FROM Customers  
WHERE Country = 'Germany';

**INSERT INTO**

INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers;

**CASE**

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;

[**IFNULL()**](https://www.w3schools.com/sql/func_mysql_ifnull.asp)**, ISNULL(), NVL(), COALESCE() –** check for null and return the other value

Select ifnull(price,0) from products;

**Stored procedure**

CREATE PROCEDURE SelectAllCustomers @City nvarchar(30)  
AS  
SELECT \* FROM Customers WHERE City = @City  
GO;

EXEC SelectAllCustomers @city = ‘London’;

Comments

--Select all:

/\*Select

\*/

**DDL**

CREATE DATABASE testDB;

DROP DATABASE testDB;

BACKUP DATABASE testDB   
TO DISK = 'filepath';

CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)   
);

Truncate table Persons;

Drop table Persons;

ALTER TABLE Customers  
ADD Email varchar(255);

ALTER TABLE Customers  
Drop column Email;

ALTER TABLE Customers  
modify column email varchar(30);

**Constraints**

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);

ALTER TABLE Persons  
ADD UNIQUE (ID);

ALTER TABLE Persons  
DROP CONSTRAINT UC\_Person;

ALTER TABLE Persons  
ADD PRIMARY KEY (ID);

ALTER TABLE Orders  
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);

ALTER TABLE Persons  
ADD CONSTRAINT CHK\_PersonAge CHECK (Age>=18 AND City='Sandnes');

ALTER TABLE Persons  
ALTER City SET DEFAULT 'Sandnes';

CREATE INDEX idx\_lastname  
ON Persons (LastName);

CREATE TABLE Persons (  
    Personid int NOT NULL AUTO\_INCREMENT,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (Personid)  
);

CREATE VIEW [Products Above Average Price] AS  
SELECT ProductName, Price  
FROM Products  
WHERE Price > (SELECT AVG(Price) FROM Products);