SQL COMMANDS

# **SQL TUTORIALS :**

* SELECT \* FROM Customers;
* SELECT DISTINCT Country FROM Customers;
* SELECT \* FROM Customers WHERE CustomerID=1;
* SELECT \* FROM Customers WHERE Country='Germany' AND (City='Berlin' OR NOT City='München');
* SELECT \* FROM Customers ORDER BY Country ASC, CustomerName DESC;
* INSERT INTO Customers (CustomerName, City, Country) VALUES ('Cardinal', 'Stavanger', 'Norway');
* SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NOT NULL;
* UPDATE Customers SET ContactName='Juan' WHERE Country='Mexico';
* DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
* SELECT \* FROM Customers LIMIT 3;
* SELECT \* FROM Customers WHERE Country='Germany' LIMIT 3;
* SELECT MAX(Price) AS LargestPrice –- also can use MIN(Price) FROM Products;
* SELECT COUNT(ProductID) FROM Products WHERE date = ‘2019-01-01’;
* SELECT AVG(Price) FROM Products WHERE date = ‘2019-01-01’;
* SELECT SUM(Quantity) FROM OrderDetails WHERE date = ‘2019-01-01’;
* SELECT \* FROM Customers WHERE CustomerName LIKE '%or%';
* % Represents zero or more characters
* \_ Represents a single character
* [] Represents any single character within the brackets
* ^ Represents any character not in the brackets
* - Represents a range of characters
* SELECT \* FROM Customers WHERE Country IN (SELECT Country FROM Suppliers);
* SELECT \* FROM Customers WHERE Country NOT IN ('Germany', 'France', 'UK');
* SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20 AND CategoryID NOT IN (1,2,3)

ORDER BY Price;

* SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address  
  FROM Customers;
* SELECT o.OrderID, o.OrderDate, c.CustomerName  
  FROM Customers AS c, Orders AS o  
  WHERE c.CustomerName='Around the Horn' AND c.CustomerID=o.CustomerID;
* **(INNER) JOIN**: Returns records that have matching values in both tables
* **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
* **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
* **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table
* 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;
* ELECT 'Customer' AS Type, ContactName, City, Country FROM Customers  
  UNION ALL  
  SELECT 'Supplier', ContactName, City, Country FROM Suppliers;
* SELECT COUNT(CustomerID), Country FROM Customers ORDER BY COUNT(CustomerID) DESC;
* SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country  
  HAVING COUNT(CustomerID) > 5;
* SELECT SupplierName FROM Suppliers  
  WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);
* SELECT CustomerName, ContactName INTO CustomersBackup2017 FROM Customers;
* INSERT INTO Customers (CustomerName, City, Country)  
  SELECT SupplierName, City, Country FROM Suppliers WHERE Country='Germany';
* 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 ProductName, UnitPrice \* (UnitsInStock + IFNULL(UnitsOnOrder, 0))  
  FROM Products;
* COMMENTS - - single line /\* multi line \*/

# **SQL DATABASE:**

* CREATE DATABASE testDB;
* DROP DATABASE testDB;
* BACKUP DATABASE testDB TO DISK = 'D:\backups\testDB.bak';
* CREATE TABLE Persons (  
      PersonID int,  
      LastName varchar(255),  
      FirstName varchar(255),  
      Address varchar(255),  
      City varchar(255)  
  );
* CREATE TABLE TestTable AS SELECT customername, contactname FROM customers;
* DROP TABLE Shippers; -- remove TABLE completely
* TRUNCATE TABLE Shippers; -- remove only Contents of TABLE
* ALTER TABLE Customers ADD Email varchar(255); -- this will new column
* ALTER TABLE Customers DROP COLUMN Email; -- this will delete column
* ALTER TABLE Persons ALTER COLUMN DateOfBirth year; -- this will change to year
* CREATE TABLE Persons (  
      ID int NOT NULL,  
      LastName varchar(255) NOT NULL,  
      FirstName varchar(255) NOT NULL,  
      Age int  
  );
* ALTER TABLE Persons MODIFY Age int NOT NULL;
* CREATE TABLE Persons (  
      ID int NOT NULL,  
      LastName varchar(255) NOT NULL,  
      FirstName varchar(255),  
      Age int,  
      CONSTRAINT UC\_Person UNIQUE (ID,LastName)  
  );
* ALTER TABLE Persons ADD CONSTRAINT UC\_Person UNIQUE (ID,LastName);
* ALTER TABLE Persons DROP INDEX UC\_Person;
* CREATE TABLE Persons (  
      ID int NOT NULL,  
      LastName varchar(255) NOT NULL,  
      FirstName varchar(255),  
      Age int,  
      PRIMARY KEY (ID)  
  );
* ALTER TABLE Persons ADD PRIMARY KEY (ID);
* ALTER TABLE Persons DROP PRIMARY KEY;
* CREATE TABLE Orders (  
      OrderID int NOT NULL,  
      OrderNumber int NOT NULL,  
      PersonID int,  
      PRIMARY KEY (OrderID),  
      FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
  );
* ALTER TABLE Orders ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
* ALTER TABLE Orders ADD CONSTRAINT FK\_PersonOrder  
  FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
* ALTER TABLE Orders DROP FOREIGN KEY FK\_PersonOrder;
* CREATE TABLE Persons (  
      ID int NOT NULL,  
      LastName varchar(255) NOT NULL,  
      FirstName varchar(255),  
      Age int,  
      CHECK (Age>=18)  
  );
* CREATE TABLE Persons (  
      ID int NOT NULL,  
      LastName varchar(255) NOT NULL,  
      FirstName varchar(255),  
      Age int,  
      City varchar(255),  
      CONSTRAINT CHK\_Person CHECK (Age>=18 AND City='Sandnes')  
  );
* ALTER TABLE Persons ADD CONSTRAINT CHK\_PersonAge CHECK (Age>=18 AND City='Sandnes');
* ALTER TABLE Persons ADD CHECK (Age>=18);
* ALTER TABLE Persons DROP CHECK CHK\_PersonAge;
* CREATE TABLE Persons (  
      ID int NOT NULL,  
      LastName varchar(255) NOT NULL,  
      FirstName varchar(255),  
      Age int,  
      City varchar(255) DEFAULT 'Sandnes'  
  );
* ALTER TABLE Persons ALTER City SET DEFAULT 'Sandnes';
* ALTER TABLE Persons ALTER City DROP DEFAULT;
* CREATE INDEX idx\_lastname ON Persons (LastName);
* ALTER TABLE table\_name DROP INDEX index\_name;
* CREATE TABLE Persons (  
      Personid int NOT NULL AUTO\_INCREMENT,  
      LastName varchar(255) NOT NULL,  
      FirstName varchar(255),  
      Age int,  
      PRIMARY KEY (Personid)  
  );