Some of The Most Important SQL Commands:

* SELECT - extracts data from a database
* SELECT DISTINCT - statement is used to return only distinct (different) values
  + SELECT COUNT(DISTINCT Country) FROM Customers;
* WHERE Clause –
  + <> Not equal. Note: In some versions of SQL this operator may be written as !=
  + SELECT \* FROM Customers WHERE NOT Country='Germany' AND NOT Country='USA';
* LIKE operator –
  + SELECT \* FROM Customers WHERE CustomerName LIKE 'a%'; - selects all customers with a CustomerName starting with "a"
  + SELECT \* FROM Customers WHERE CustomerName LIKE '%a'; - selects all customers with a CustomerName ending with "a"
  + SELECT \* FROM Customers WHERE CustomerName LIKE '%or%'; - selects all customers with a CustomerName that have “or” in any position
  + SELECT \* FROM Customers WHERE CustomerName LIKE '\_r%'; - selects all customers with a CustomerName that have “r” in second position
  + SELECT \* FROM Customers WHERE CustomerName LIKE 'a\_\_%'; - selects all customers with a CustomerName that starts with “a” and are atleast 3 chars in length
  + SELECT \* FROM Customers WHERE ContactName LIKE 'a%o'; - selects all customers with a ContactName that starts with "a" and ends with "o"
  + SELECT \* FROM employees WHERE length(emp\_name)=6 AND emp\_name LIKE '\_\_R%';
  + SELECT \* FROM Customers WHERE City LIKE '[bsp]%'; - selects all customers with a City starting with "b", "s", or "p"
  + SELECT \* FROM Customers WHERE City LIKE '[a-c]%'; - selects all customers with a City starting with "a", "b", or "c"
  + SELECT \* FROM Customers WHERE City LIKE '[!bsp]%'; - select all customers with a City NOT starting with "b", "s", or "p". NOT LIKE ‘[bsp]%’ is also ok.
* IS NULL / IS NOT NULL – identify null and not null values
* IN / NOT IN Keyword - specify multiple values in a WHERE clause
* BETWEEN operator –
  + SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20 AND CategoryID NOT IN (1,2,3);
  + SELECT \* FROM Products WHERE ProductName BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni' ORDER BY ProductName;
  + SELECT \* FROM Orders WHERE OrderDate BETWEEN #07/01/1996# AND #07/31/1996#; or SELECT \* FROM Orders WHERE OrderDate BETWEEN '1996-07-01' AND '1996-07-31';
* SQL Aliases –
  + SELECT CustomerID AS ID, CustomerName AS [Customer Name] FROM Customers; or SELECT CustomerID AS ID, CustomerName AS “Customer Name” FROM Customers;
  + 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;
* ORDER BY Keyword –
  + SELECT \* FROM Customers ORDER BY Country ASC, CustomerName DESC;
* UPDATE - updates data in a database
  + UPDATE Customers SET ContactName = 'Alfred Schmidt', City= 'Frankfurt' WHERE CustomerID = 1;
* DELETE - deletes data from a database
  + DELETE FROM Customers; - delete all records
  + DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
* INSERT INTO - inserts new data into a database
  + INSERT INTO Customers (CustomerName, City, Country) VALUES ('Cardinal', 'Stavanger', 'Norway');
* LIMIT –
  + SELECT \* FROM Customers LIMIT 3;
* MIN() and MAX() –
  + SELECT MIN(Price) AS SmallestPrice FROM Products;
  + SELECT MAX(Price) AS LargestPrice FROM Products;
* COUNT(), AVG() and SUM() –
  + SELECT COUNT(ProductID) FROM Products;
  + SELECT AVG(Price) FROM Products;
  + SELECT SUM(Quantity) FROM OrderDetails;
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index
* (INNER) JOIN: Returns records that have matching values in both tables
  + SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate FROM Orders INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
* LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
  + SELECT Customers.CustomerName, Orders.OrderID FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID ORDER BY Customers.CustomerName;
* 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