**Database**: A structured collection of data that is organized, stored, and managed to provide efficient retrieval and manipulation of data.

**RDBMS**: Relational Database Management System - a software system that manages relational databases.

**SQL** : Structured Query Language - a standard language for communicating with and manipulating databases.

**Primary Key**: A unique identifier for a row in a database table.Always unique and not null

| What | Explanation | Example |
| --- | --- | --- |
| WHERE clause | Used to filter rows in a SQL query based on specified conditions. | **SELECT \* FROM employees WHERE age > 30** |
| String Functions | Functions that manipulate and operate on string values. | **SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM employees** |
| Integer Operations | Mathematical operations performed on integer values in SQL. | **SELECT (quantity \* price) AS total FROM orders** |
| AND | Operator used to combine multiple conditions in a WHERE clause, where all conditions must be true. | **SELECT \* FROM employees WHERE age > 30 AND department = 'IT'** |
| OR | Operator used to combine multiple conditions in a WHERE clause, where at least one condition must be true. | **SELECT \* FROM employees WHERE age > 30 OR department = 'IT'** |
| IN | Operator used to specify multiple possible values in a WHERE clause. | **SELECT \* FROM employees WHERE department IN ('IT', 'HR')** |
| NOT | Operator used to negate a condition in a WHERE clause. | **SELECT \* FROM employees WHERE NOT department = 'IT'** |
| DISTINCT | Keyword used to retrieve unique values in a SELECT statement. | **SELECT DISTINCT department FROM employees** |
| LIKE | Operator used for pattern matching in a WHERE clause. | **SELECT \* FROM employees WHERE last\_name LIKE 'S%'** |
| Subquery | A query nested within another query. | **SELECT \* FROM employees WHERE department IN (SELECT department FROM managers)** |
| MIN | Aggregate function used to retrieve the minimum value in a column. | **SELECT MIN(salary) FROM employees** |
| MAX | Aggregate function used to retrieve the maximum value in a column. | **SELECT MAX(salary) FROM employees** |
| AVG | Aggregate function used to retrieve the average value in a column. | **SELECT AVG(salary) FROM employees** |
| BETWEEN | Operator used to retrieve values within a specified range. | **SELECT \* FROM employees WHERE age BETWEEN 25 AND 35** |
| UNION | Operator used to combine the result sets of two or more SELECT statements. | **SELECT \* FROM employees UNION SELECT \* FROM managers** |
| UNION ALL | Operator used to combine the result sets of two or more SELECT statements, including duplicate rows. | **SELECT \* FROM employees UNION ALL SELECT \* FROM managers** |
| ANY | Operator used to compare a value with a set of values and returns true if any of the comparisons is true. | **SELECT \* FROM products WHERE price > ANY (SELECT price FROM competitors)** |
| ALL | Operator used to compare a value with a set of values and returns true if all of the comparisons are true. | **SELECT \* FROM products WHERE price > ALL (SELECT price FROM competitors)** |
| EXISTS | Operator used to check the existence of rows in a subquery. | **SELECT \* FROM employees WHERE EXISTS (SELECT \* FROM orders WHERE orders.employee\_id = employees.employee\_id)** |

Some of sample queries

—-------------------------------------------------------------------------------

SELECT \* FROM Customers

--where customerName = 'Alfreds Futterkiste'

where customerId = 2

SELECT \* FROM [Customers]

--where customerName = 'Alfreds Futterkiste'

--where lower(customerName) = 'alfreds futterkiste'

where upper(customerName) = 'ALFREDS FUTTERKISTE'

SELECT \* FROM [Customers]

--where customerName = 'Alfreds Futterkiste'

--where lower(customerName) = 'alfreds futterkiste'

--where upper(customerName) = 'ALFREDS FUTTERKISTE'

--where city in ('Berlin','London')

--where city = 'Berlin' or city = 'London'

where country = 'UK'

and city = 'Cowes'

SELECT \* FROM Customers

--WHERE NOT Country='Germany';

WHERE Country NOT in ('Germany','Mexico');

SELECT \* FROM Customers WHERE NOT Country='Germany' AND NOT Country='USA'

SELECT \* FROM Customers WHERE Country='Germany' AND (City='Berlin' OR City='München');

SELECT \* FROM Customers WHERE NOT Country='Germany' AND NOT Country='USA';

SELECT \* FROM Customers

--WHERE NOT Country='Germany';

WHERE Country NOT in ('Germany','Mexico');

SELECT \* FROM Customers WHERE NOT Country='Germany' AND NOT Country='USA';

SELECT \* FROM Customers WHERE Country='Germany' AND (City='Berlin' OR City='München');

SELECT distinct Country FROM Customers;

SELECT \* FROM Customers WHERE City LIKE '\_ondon';

SELECT \* FROM Customers WHERE City LIKE '[bsp]%';

SELECT \* FROM Customers WHERE Country IN (SELECT Country FROM Suppliers);

SELECT \* FROM Customers

ORDER BY Country

--ORDER BY Country DESC

--ORDER BY Country, CustomerName

--ORDER BY Country ASC, CustomerName DESC;

SELECT CustomerName, ContactName, Address

FROM Customers

WHERE Address IS NULL

--WHERE Address IS NOT NULL

SELECT MAX(Price),MIN(SupplierId),MIN(Price),productName as MinimumPrices1111 FROM Products

SELECT COUNT(ProductID) FROM Products;

SELECT SUM(Quantity) FROM OrderDetails

SELECT AVG(Price) FROM Products;

SELECT distinct Country,city FROM Customers where country = 'Mexico';

SELECT \* FROM Orders WHERE OrderDate BETWEEN '1996-07-01' AND '1996-07-31';

SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20 AND CategoryID NOT IN (1,2,3);