**Various Syntax in SQL:**

**SQL SELECT Statement:**

SELECT column1, column2....columnN FROM table\_name;

**SQL DISTINCT Clause :**

SELECT DISTINCT column1, column2....columnN FROM table\_name;

**SQL WHERE Clause:**

SELECT column1, column2....columnN FROM table\_name WHERE CONDITION;

**SQL AND/OR Clause:**

SELECT column1, column2....columnN FROM table\_name WHERE CONDITION-1 {AND|OR} CONDITION-2;

**SQL IN Clause:** The IN operator allows you to specify multiple values in a WHERE clause. The IN operator is a shorthand for multiple OR conditions.

SELECT column1, column2....columnN FROM table\_name WHERE column\_name IN (val-1, val-2,...val-N);

Ex: SELECT \* FROM student WHERE branch IN ('CSE', 'IT', 'ECE');

--select all student in cse,it and ece branch

**SQL BETWEEN Clause:** The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

SELECT column1, column2....columnN FROM table\_name WHERE column\_name BETWEEN val-1 AND val-2;

**SQL LIKE Clause:** The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

* % - The percent sign represents zero, one, or multiple characters
* \_ - The underscore represents a single character

SELECT column1, column2....columnN FROM table\_name WHERE column\_name LIKE { PATTERN };

Ex:

|  |  |
| --- | --- |
| **LIKE Operator** | **Description** |
| WHERE CustomerName LIKE 'a%' | Finds any values that start with "a" |
| WHERE CustomerName LIKE '%a' | Finds any values that end with "a" |
| WHERE CustomerName LIKE '%or%' | Finds any values that have "or" in any position |
| WHERE CustomerName LIKE '\_r%' | Finds any values that have "r" in the second position |
| WHERE CustomerName LIKE 'a\_\_%' | Finds any values that start with "a" and are at least 3 characters in length |
| WHERE ContactName LIKE 'a%o' | Finds any values that start with "a" and ends with "o" |

**SQL ORDER BY Clause:** The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

SELECT column1, column2....columnN FROM table\_name WHERE CONDITION ORDER BY column\_name {ASC|DESC};

**SQL GROUP BY Clause:** The GROUP BY statement group rows that have the same values into summary rows, like "find the number of customers in each country".

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.

SELECT SUM(column\_name) FROM table\_name WHERE CONDITION GROUP BY column\_name;

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

--statement lists the number of customers in each country.

**SQL COUNT Clause:**

SELECT COUNT(column\_name) FROM table\_name WHERE CONDITION;

**SQL HAVING Clause:** he HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

SELECT column\_name(s) FROM table\_name WHERE condition GROUP BY column\_name(s)HAVING condition ORDER BY column\_name(s);

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

--statement lists the number of customers in each country. Only include countries with more than 5 customers.

**SQL CREATE TABLE Statement :**

CREATE TABLE table\_name( column1 datatype, column2 datatype, column3 datatype, ..... columnN datatype, PRIMARY KEY( one or more columns ) );

**SQL DROP TABLE Statement:**

DROP TABLE table\_name;

**SQL CREATE INDEX Statement:**

CREATE UNIQUE INDEX index\_name ON table\_name ( column1, column2,...columnN);

**SQL DROP INDEX Statement:**

ALTER TABLE table\_name DROP INDEX index\_name;

**SQL DESC Statement:**

DESC table\_name;

**SQL TRUNCATE TABLE Statement:**

TRUNCATE TABLE table\_name;

**SQL ALTER TABLE Statement:**

ALTER TABLE table\_name {ADD|DROP|MODIFY} column\_name {data\_ype};

**SQL ALTER TABLE Statement (Rename) :**

ALTER TABLE table\_name RENAME TO new\_table\_name;

**SQL INSERT INTO Statement:**

INSERT INTO table\_name( column1, column2....columnN) VALUES (value1,value2....valueN);

**SQL UPDATE Statement:**

UPDATE table\_name SET column1 = value1, column2 = value2....columnN=valueN [ WHERE CONDITION ];

**SQL DELETE Statement :**

DELETE FROM table\_name WHERE {CONDITION};

**SQL CREATE DATABASE Statement:**

CREATE DATABASE database\_name;

**SQL DROP DATABASE Statement:**

DROP DATABASE database\_name;

**SQL USE Statement:**

USE database\_name;

**SQL COMMIT Statement:**

COMMIT;

**SQL ROLLBACK Statement:**

ROLLBACK;