SQL CLAUSES

- SQL clauses are the components of SQL statements that help in specifying various conditions and operations to manipulate and retrieve data from a database.

1. WHERE Clause

- A WHERE clause in SQL is used with the SELECT query, which is one of the data manipulation language commands.
- WHERE clauses can be used to limit the number of rows to be displayed in the result set, it generally helps in filtering the records.
- It returns only those queries which fulfill the specific conditions of the WHERE clause. WHERE clause is used in SELECT, UPDATE, DELETE statement, etc.

- **Where** Clause is also used together with **BETWEEN** Operator .
- It is used to fetch filtered data in a given range inclusive of two values.
- For example: To fetch records of Employees where Age is between 22 and 24 (inclusive).
- SELECT * FROM Emp1 WHERE Age BETWEEN 22 AND 24;

- **Where** Clause is also used together with the **LIKE** Operator
- It is used to fetch filtered data by searching for a particular pattern in the where clause.
- Eg: To fetch records of Employees where Name starts with the letter S.
- SELECT * FROM Emp1 WHERE Name LIKE 'S%';

- Where Clause is also used with the **IN** Operator
- It is used to fetch the filtered data same as fetched by ‘=’ operator just the difference is that here we can specify multiple values for which we can get the result set.
- Eg : To fetch the Names of Employees where Age is 21 or 23.
- SELECT Name FROM Emp1 WHERE Age IN (21,23);

SQL ORDER BY

- SQL ORDER BY clause sorts the result of the SELECT statement either in ascending or descending order.
- It is used to sort the fetched data in either ascending or descending according to one or more columns.
- ORDER BY default mode is sorting data into ascending order.
- To sort data in descending order use the DESC keyword with ORDER BY clause.
- For example, to fetch all data from the table Student and sort the result in descending order according to the column ROLL_NO.
- SELECT * FROM students ORDER BY ROLL_NO DESC;

SQL HAVING Clause

- Having is a very generally used clause in SQL. Similar to WHERE it helps to apply conditions, but HAVING works with groups. If you wish to filter a group, the HAVING clause comes into action.
- Having clause is used to filter data according to the conditions provided.
- Having clause is only used with the SELECT clause.
- In the query, ORDER BY is to be placed after the HAVING clause, if any.
- HAVING Clause is implemented in column operation.
- Having clause is generally used after GROUP BY.

- Syntax:

```
SELECT col_1, function_name(col_2)
FROM tablename
WHERE condition
GROUP BY column1, column2
HAVING Condition
ORDER BY column1, column2;
```

- To sort according to multiple columns, separate the names of columns by the , operator.
- For example, to fetch all data from the table Student and then sort the result in descending order first according to the column age and then in ascending order according to the column name.
- SELECT * FROM students ORDER BY age DESC , name ASC;

GROUP BY

- The GROUP BY Statement in SQL is used to arrange identical data into groups with the help of some functions. i.e. if a particular column has the same values in different rows then it will arrange these rows in a group.
- Group by multiple columns is say, for example, GROUP BY column1, column2. This means placing all the rows with the same values of columns column 1 and column 2 in one group.

Group By single column

- Group By single column means, placing all the rows with the same value of only that particular column in one group.

Group By Multiple Columns

- Group by multiple columns is say, for example, GROUP BY column1, column2. This means placing all the rows with the same values of columns column 1 and column 2 in one group.
- We know that the WHERE clause is used to place conditions on columns but what if we want to place conditions on groups? This is where the HAVING clause comes into use.

```
SELECT column1, function_name(column2)
FROM table_name
WHERE condition
GROUP BY column1, column2
HAVING condition
ORDER BY column1, column2;
```

SQL LIMIT Clause

- The LIMIT clause in SQL allows users to control the amount of data retrieved and displayed in the result set.
- It is useful when only a subset of records is needed for analysis or display purposes in large databases with thousands of records.
- For example, if we only need to retrieve 3 rows from the student table using LIMIT.

```
SELECT * FROM student
LIMIT 3;


SELECT * FROM Student
ORDER BY Grade DESC
LIMIT 3;
```

SQL LIMIT OFFSET

- LIMIT OFFSET parameter skips a specified number of rows before returning the result set.
- OFFSET can only be used with the ORDER BY clause. It cannot be used on its own.
- OFFSET value must be greater than or equal to zero. It cannot be negative, else returns an error.
- Syntax:

```
SELECT * FROM table_name ORDER BY column_name LIMIT X OFFSET Y;
```

- OR

```
SELECT * FROM table_name ORDER BY column_name LIMIT X,Y;
```

- For example we will skip first 2 values using offset and print only 3 rows.

```
SELECT * FROM Student
ORDER BY ROLLNO LIMIT 2,5;
```

  - Example using the order by clauses :

```
SELECT age
FROM Student
WHERE id<4
ORDER BY age
LIMIT 2, 1;
```