POSTGRESQL CLASS 4

SELECT STATEMENTS I AND II

WHERE CLAUSE

- The PostgreSQL WHERE clause is used to control a query. The WHERE clause eliminates all rows from the output that do not meet the condition. It is generally used with SELECT, UPDATE and DELETE statements to filter the results. It returns the specific result only when the condition is satisfied. The WHERE clause specifies a condition while fetching data from a table or joining multiple tables. WHERE condition can be used with logical operators such as >, <, =, LIKE, NOT, OR, etc.

WHERE CLAUSE

- Equal (=) operator
- AND operator
- OR operator
- IN operator
- LIKE operator
- BETWEEN operator
- Not equal operator (<>) or (! =)
- Combining AND & OR conditions

IN OPERATOR

- This is an alternative to multiple OR conditions.

ORDER BY

- This clause is used to sort data in ascending and descending order in PostgreSQL. We can use one or more column in order by clause to sort data in ascending or descending order, but we make sure that this column is present in that table.
- Normally without using order by function in PostgreSQL, data retrieve in an unspecified order.
- Using order by function in PostgreSQL, our data comes in the specified order.
- The default sequence is ascending order that will return the smallest value first.
- Also, the DESC function is available in PostgreSQL to sort data in descending order.

ORDER BY

- **ASC (Ascending)**: This is an optional clause in PostgreSQL order by clause. It will sort the result by ascending order.
- **DESC (Descending):** This is an optional clause in PostgreSQL order by clause. It will sort the result by descending order.
- **Nulls first:** It will sort all null values before non-nulls in PostgreSQL order by clause result set. This is an optional clause in PostgreSQL order by clause.
- **Nulls last:** It will sort all null values after non-nulls in PostgreSQL order by clause result set. This is an optional clause.

LIKE

- When you use LIKE expression to match a certain string with the pattern, if the pattern matches, then true is returned; else, false is returned by the like expression. NOT LIKE expression behaves oppositely

DISTINCT

- We can remove the duplicate rows from a statement’s result set by using a PostgreSQL DISTINCT clause in the SELECT statement

LIMIT

- PostgreSQL limit clause returns the number of rows from the table, which was mentioned in the limit value at the time of fetching the record from the table.
- SELECT * FROM people LIMIT 100; // This will return only the first 100 rows.

GROUP BY

- PostgreSQL GROUP BY clause is used to collect data from multiple records and then to group the identical or similar results. It is used with the PostgreSQL SELECT statement.
- It, however, does not have a mandatory existence with the PostgreSQL SELECT statement.

HAVING CLAUSE

- PostgreSQL HAVING clause is used to return the groups of rows only when the condition is TRUE. It is used with the PostgreSQL GROUP BY Clause. It, however, does not have a mandatory existence with the PostgreSQL GROUP BY Clause.
- It should always be written before the ORDER BY clause