PART -- II

**1.Distinct class:**

It displays one occurance from the duplicate records.

**2.In-Operator:**

In operator is used to filter the list of values, each value available in a list it applys OR operator.

**3.Between Operator:** It is used to filter the range of values, we should used between operator.

**4.Pattern Matching:**

By using like operator we can filter the records based the partial matching.it is also called as pattern matching concept in SQL. It has achieved two inbuilt special characters.

A . % --> It matches more than one character at a time.

B. - (underscore) -->It matches only one character at a time.

**5.Order-by-Clause :** In order by class is used to sort the records either ascending or descending order. By default it sorts the records in ascending order.

**6.Set Operations:**

Set operations are used to combine records from single or multiple tables.

**Set operations types:**

1.Union

2.Union All

3.Intersect:

4.Minus:

**1.Union:** The union operation is used to combine the records from single or multiple tables. union operation always provides in distinct records, and the output union operation always sorted in ascending order.

**2.Union All:** The union all operation is used to combine the records from single or multiple tables. Union all operation always provides duplicate records, and the output of the union all in any sorted order.

**3.Intersect:** The intersect operation provides common records from both the tables(queries), and the output of intersect operation always sorted in ascending order.

**4.Minus:** The minus operation provides the records which are available in the first query, and the same records are not available in the second query.

**7.Aggeragate function:**

**1.Max:** This function provides the maximum value from the specified column.

**2.Min:** This function provides the minimum value from the specified column.

**3.Sum:** This function provides the sum of all the records available in a column.

**4.Count:** This function provides the number of records available in a column.

**5.Count(\*):** It provides the number of records available in a table.

**6.Average:** This function provides the average value of the specified column.

**8. Group by clause:**

The group by class is used to group the records based on the aggregate function.

**Two rules:**

1.The column used in the specified select statement, the same column name must be specified in the group by clause.

2.After satisfying the first rule of the column name which has specified the group by clause, the same column names need not to be specified in the select statement.

**9. Having clause:**

In having class is used to filter the grouped records using aggregate function and condition.

**10.Having clause Where clause**

1.Having clause must be used after 1.Where clause can be used before the group by clause. the group by class.

2.In having class we must use aggregate function 2.In where we should not use to grouped records. aggregate function to filter the grouped records

3.Having class is used to filter the grouped 3.Where clause is used to filter the

records. Individual records.

**11.Sub-query or nested query:**

Sub query is also called as nested query. In sub query the inner query executes first based on the result of inner query the outer query executes.

**12.Core related sub query:**

Core related sub query is also called as nested query. Here the inner query and outer query executes first based on the matching condition provided.

**13.Table alias:**

In order to differentiate columns in each respective available table we should use alias name.

**14.Joins:**

By using join operation we can join different columns in different tables.

**Types of joins:**

1.Cartesian join or cross join

2.Inner join or Equi join

3.Left join or Left outer join

4.Right join or Right outer join

5.Full join or Full outer join

6.Self join

**1.Cartesian join or cross join:** The cartesian works based on the principle of cartesian products.

**2.Inner join or Equi join:** Inner joins returns matching records from both the tables based on the matching condition provided.

**3.Left join or Left outer join:** Left join provides all the records from the left table and it is only matching records from the right table.

**4.Right join or Right outer join:** Right join provides all the records from the right table and it is only matching records from the left table.

**5.Full join or Full outer join:** Full join provides the matching as well as non matching records from both the tables.

**6.Self join:** Joining the same table by using inner join, left join or right join that indicates self join.

**15.Implicit join:**

While joining two or more tables if we are using where clause to perform join operation that indicates implicit join.

**16.Explicit join:**

While joining two or more tables if we are using on clause or using clause along with key words inner join, left join and right join that indicates explicit join.

**17.Ansi join:**

It is also called as Explicit join.

While joining two or more tables if we are using on clause or using clause along with key words inner join, left join and right join that indicates ansi join.

**18.Natural join:**

Natural join provides matching records from both the tables it works as similar to inner join.

**19.Using clause:**

While joining two or more tables if we are using keywords inner join, left join and right join etc.,, along with using clause that indicates using clause concept.

**20.Normalization:**

Normalization is used to organise the data in such a manner that data redundancy will never occur in the database and data should not be inconsistent manner.

**1.First normalization form:**

If the table has to satisfy 1NF, each column in a table should have automic value, column should not contain repeating group of values.

**2.Second normalization form:**

The table must be in 1NF, every non key column value must entirely depending on primary key.

**3.Third normalization form:**

The table must be in 2NF, there should not be any transitive dependency.

**Transitive dependency:**

A non key column depends on another non key column that indicates the transitive dependency.

**21.Row Num:**

Row num is a sequentialy generated integer number assigned to each record available in a table it always starts with one.

\* If a table contains 10 records row num starts with 1 and ends with 10.

\* The insertion or deletion operation on table it changes the row num.

**22.Row Id:**

Row id contains 16 to 18 characters of length of alpha numeric value, it is a fixed value assigned to each record available in a table.

**23.Inline Query:**

While writing the sql query inplace of table name if we can write simple or complex query that indicates inline query concept.