1.SELECT country,order\_id,substring(order\_id,0,charindex('-',order\_id))Country\_code FROM projectsql

2. select \* from(SELECT Order\_ID,city,product\_name,sales,profit,row\_number() over (partition by city order by profit desc) as city\_sale\_rank FROM projectsql) where city\_sale\_rank <=5

3. select row\_no from(SELECT Order\_ID,city,product\_name,sales,profit,row\_number() over (partition by null) as row\_no

FROM projectsql) order by row\_no desc limit 1

4. SELECT distinct city, state FROM projectsql

5. SELECT segment,category,sales FROM projectsql where sales>500 order by segment

6.SELECT segment,category,sales,

case when sales <=100 then "Low"

when sales>100 and sales<=500 then "Medium"

when sales>500 and sales<=2000 then "Medium"

when sales>2000 then "Best"

end as category\_sales

FROM projectsql

7. select \* from (SELECT city,region,sum(profit) over (partition by region,city) as total\_profit,

count(profit) over (partition by region,city) as count\_profit

,sum(sales) over (partition by region,city) as total\_sales,

count(sales) over (partition by region,city) as count\_of\_sales

FROM projectsql) group by city

8. SELECT \* FROM projectsql where region= "East" or region='West' and sales>500

9. SELECT Order\_ID,city,product\_name,sales,

rank() over (partition by city order by sales desc) as city\_sale\_rank,1

dense\_rank() over (partition by city order by sales desc) as city\_sale\_denserank FROM projectsql

10.

* The **LEFT JOIN** keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

LEFT JOIN Syntax :

SELECT column\_name(s) FROM table1  
LEFT JOIN table2  
ON table1.column\_name = table2.column\_name

* The **INNER JOIN** keyword selects records that have matching values in both tables.
* **The RIGHT JOIN** keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

RIGHT JOIN Syntax

SELECT column\_name(s)  
FROM table1  
RIGHT JOIN table2  
ON table1.column\_name = table2.column\_name;

* A **self join** is a regular join, but the table is joined with itself.

Self Join Syntax

SELECT column\_name(s)  
FROM table1 T1, table1 T2  
WHERE condition;

* The **CROSS JOIN** keyword returns all records from both tables (table1 and table2).

CROSS JOIN Syntax

SELECT column\_name(s)  
FROM table1  
CROSS JOIN table2;

**The SQL UNION Operator**

The UNION operator is used to combine the result-set of two or more SELECT statements.

Every SELECT statement within UNION must have the same number of columns

The columns must also have similar data types

The columns in every SELECT statement must also be in the same order

UNION Syntax

SELECT *column\_name(s)* FROM *table1*  
UNION  
SELECT *column\_name(s)* FROM *table2*;

**Sub Query:**

In SQL a Subquery can be simply defined as a query within another query. In other words we can say that a Subquery is a query that is embedded in WHERE clause of another SQL query. Important rules for Subqueries: You can place the Subquery in a number of SQL clauses: WHERE clause, HAVING clause, FROM clause.