fingerTips

Module-8

Common Table Expressions (CTE)

1) Common Table Expressions

Common Table Expressions (CTEs) in SQL are temporary named result sets that are defined within the scope of a single SQL statement. They are used to simplify complex SQL queries by breaking them down into smaller, more manageable parts. CTEs are a powerful SQL feature that can help improve query performance and readability, and make complex queries easier to understand.

Here are some key features of CTEs:

- Temporary: CTEs are created and used only within the scope of a single SQL statement. They are not stored in the database like regular tables, and are not accessible to other SQL statements.
- Named: CTEs have a name, which is used to refer to the result set within the SQL statement. This allows you to refer to the same result set multiple times within the same query.
- Recursive: CTEs can be used to define recursive queries, where the result of one query is used as input to the next query in the recursion.

Syntax:

```
WITH my_cte AS (
  SELECT column1, column2
  FROM my_table
  WHERE column3 = 'value'
)
SELECT *
FROM my_cte;
```

2) What are the least delivery ratings given by each user?

```
with least_del as
(select u.name, o.delivery_rating,
row_number() over(partition by u.name order by o.delivery_rating) as row_
from users1 u join orders1 o
```

```
on u.user_id=o.user_id)
select * from least_del
where row_=1;
```

3) What is the maximum amount food ordered by each user?

```
with max_amount as
(select u.name, o.amount,
row_number() over(partition by u.name order by o.amount desc) as row_
from users1 u join orders1 o
on u.user_id=o.user_id)
select * from max_amount
where row_=1;
```

4) What are the highest restaurant ratings given by each user?

```
with highest_res as
(select u.name, o.delivery_rating,
row_number() over(partition by u.name order by o.restaurant_rating desc) as row_
from users1 u join orders1 o
on u.user_id=o.user_id)
select * from highest_res
where row_=1;
```

5) What is the minimum amount food ordered by each user?

```
with min_amount as
(select u.name, o.amount,
row_number() over(partition by u.name order by o.amount) as row_
from users1 u join orders1 o
on u.user_id=o.user_id)
select * from min_amount
where row_=1;
```