Problem statement:

You are given two tables: products and orders. The products table contains information about each product, including the product ID and available quantity in the warehouse. The orders table contains details about customer orders, including the order ID, product ID, order date, and quantity requested by the customer.

Write an SQL query to generate a report listing the orders that can be fulfilled based on the available inventory in the warehouse, following a first-come-first-serve approach based on the order date. Each row in the report should include the order ID, product name, quantity requested by the customer, quantity actually fulfilled, and a comments column as below:

If the order can be completely fulfilled then 'Full Order',If the order can be partially fullfilled then 'Partial Order',If order can not be fulfilled at all then 'No Order'.

Table: products(primary key : product\_id)

| **column name** | **datatype** |
| --- | --- |
| product\_id | int |
| product\_name | varchar(10) |
| available\_quantity | int |

Table: orders (primary key : order\_id)

| **column name** | **datatype** |
| --- | --- |
| order\_id | int |
| product\_id | int |
| order\_date | date |
| quantity\_requested | int |

Expected Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **order\_id** | **product\_name** | **quantity\_requested** | **qty\_fulfilled** | **comments** |
| 1 | Product A | 5 | 5 | Full Order |
| 2 | Product A | 7 | 5 | Partial Order |
| 3 | Product B | 10 | 10 | Full Order |
| 4 | Product B | 10 | 10 | Full Order |
| 5 | Product B | 5 | 0 | No Order |
| 6 | Product C | 4 | 4 | Full Order |
| 7 | Product C | 5 | 5 | Full Order |
| 8 | Product D | 4 | 4 | Full Order |
| 9 | Product D | 5 | 5 | Full Order |
| 10 | Product D | 8 | 1 | Partial Order |
| 11 | Product D | 5 | 0 | No Order |

Solution:

with cte as (

select a.\*,

sum(a.quantity\_requested) over (partition by a.product\_id order by a.order\_date)

as rolling\_qty\_req, b.product\_name,b.available\_quantity

from orders a join products b on a.product\_id=b.product\_id

),

cte1 as (

select order\_id,product\_name,quantity\_requested,

case when rolling\_qty\_req<=available\_quantity then quantity\_requested

when rolling\_qty\_req>available\_quantity then quantity\_requested - (rolling\_qty\_req-available\_quantity)

end as qty\_fulfilled, rolling\_qty\_req,available\_quantity from cte

)

select order\_id,product\_name,quantity\_requested,

case when qty\_fulfilled>=0 then qty\_fulfilled else 0 end as qty\_fulfilled,

case when qty\_fulfilled<=0 then 'No Order'

when qty\_fulfilled=quantity\_requested then 'Full Order'

when qty\_fulfilled<quantity\_requested and qty\_fulfilled>0 then 'Partial Order' end as comments

from cte1;