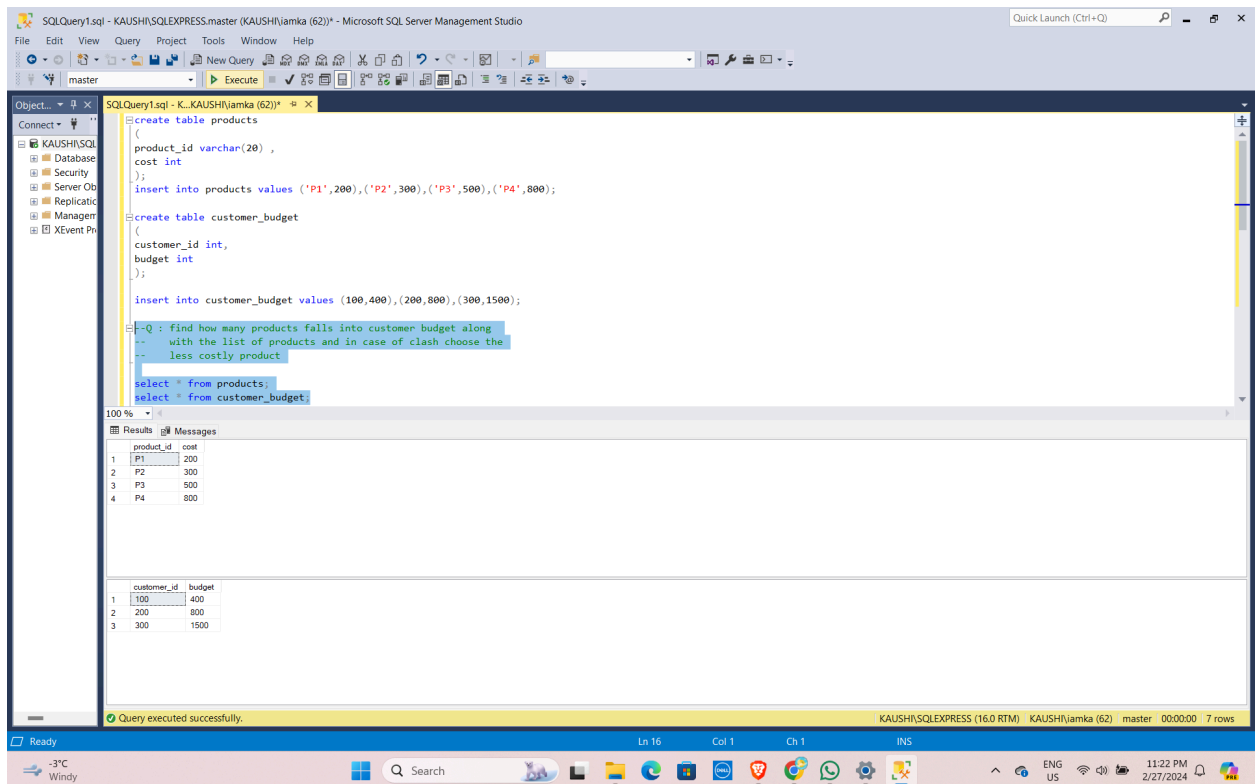


Day22 - Feb27th 2024

1. Woke up 5:30 and started all my activities
2. Started cooking food and packed food to consume at library
3. Headed to library at 8:15am
4. Opened leetcode and solved one Medium question..learned a new way of applying binary search
5. Started marketing my profile for data engineering ..but all I receive is rejection mails
6. To get good hands-on started doing pyspark project and will complete it in 1 week

7. Ended my day by solving a Meesho's SQL interview Question



The screenshot shows the SQL Server Enterprise Manager interface. The query editor contains the following SQL code:

```
create table products
(
    product_id varchar(20) ,
    cost int
);
insert into products values ('P1',200),('P2',300),('P3',500),('P4',800);

create table customer_budget
(
    customer_id int,
    budget int
);
insert into customer_budget values (100,400),(200,800),(300,1500);

--Q : find how many products falls into customer budget along
-- with the list of products and in case of clash choose the
-- less costly product

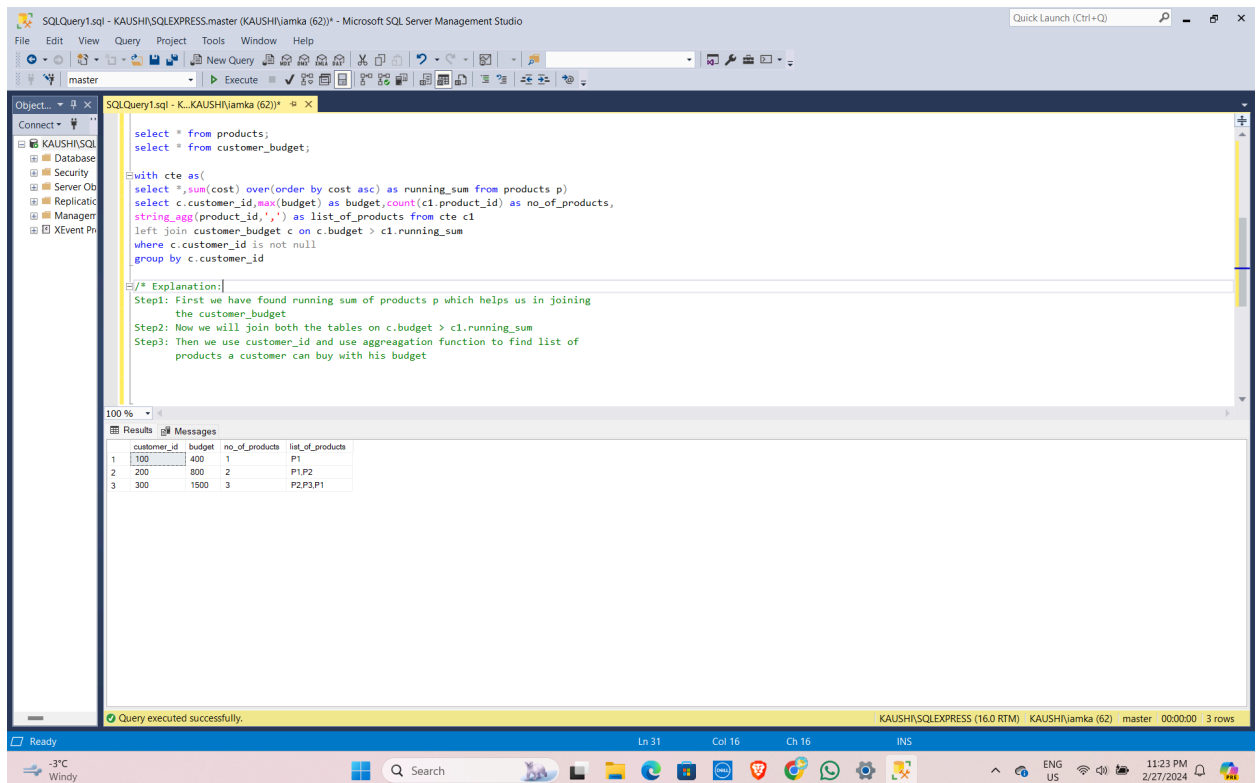
select * from products;
select * from customer_budget;
```

The results pane shows two tables:

| product_id | cost |
|------------|------|
| P1 | 200 |
| P2 | 300 |
| P3 | 500 |
| P4 | 800 |

| customer_id | budget |
|-------------|--------|
| 100 | 400 |
| 200 | 800 |
| 300 | 1500 |

Query executed successfully.



The screenshot shows the SQL Server Enterprise Manager interface. The query editor contains the following SQL code:

```
select * from products;
select * from customer_budget;

with cte as(
    select *,sum(cost) over(order by cost asc) as running_sum from products p)
select c.customer_id,max(budget) as budget,count(c1.product_id) as no_of_products,
string_agg(product_id,',') as list_of_products from cte c1
left join customer_budget c on c.budget > c1.running_sum
where c.customer_id is not null
group by c.customer_id

/* Explanation:
Step1: First we have found running sum of products p which helps us in joining
the customer_budget
Step2: Now we will join both the tables on c.budget > c1.running_sum
Step3: Then we use customer_id and use aggregation function to find list of
products a customer can buy with his budget
```

The results pane shows a table with the following data:

| customer_id | budget | no_of_products | list_of_products |
|-------------|--------|----------------|------------------|
| 100 | 400 | 1 | P1 |
| 200 | 800 | 2 | P1,P2 |
| 300 | 1500 | 3 | P2,P3,P1 |

Query executed successfully.