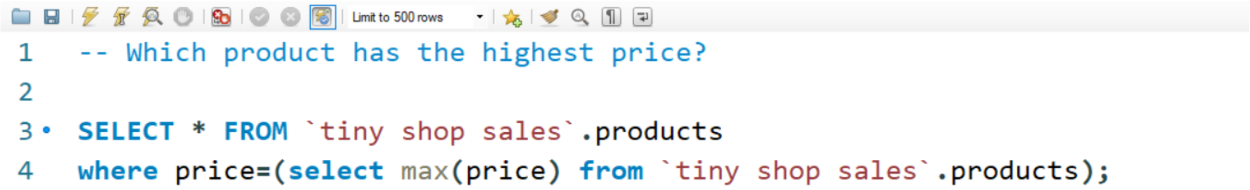
SQL CASE STUDY

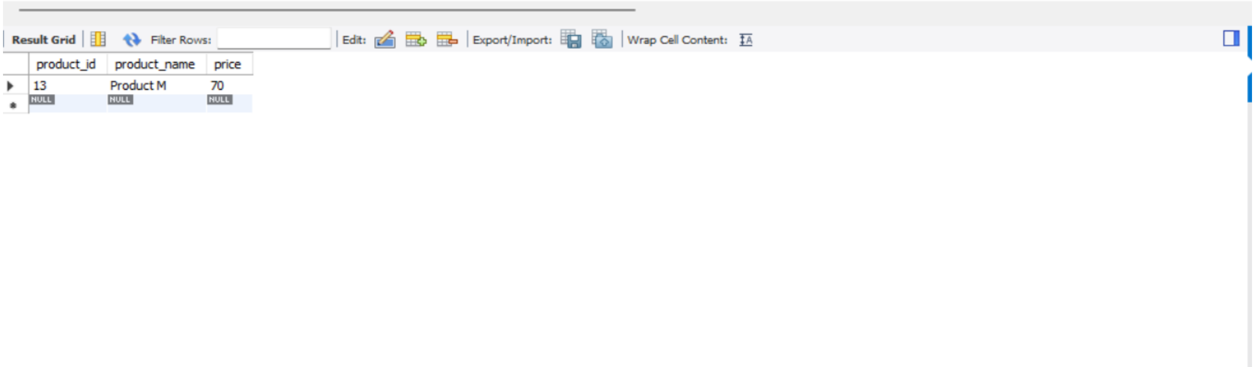
DATA IN MOTION
TINY SHOP SALES

DATA IN MOTION

1. Which product has the highest price? Only return a single row.

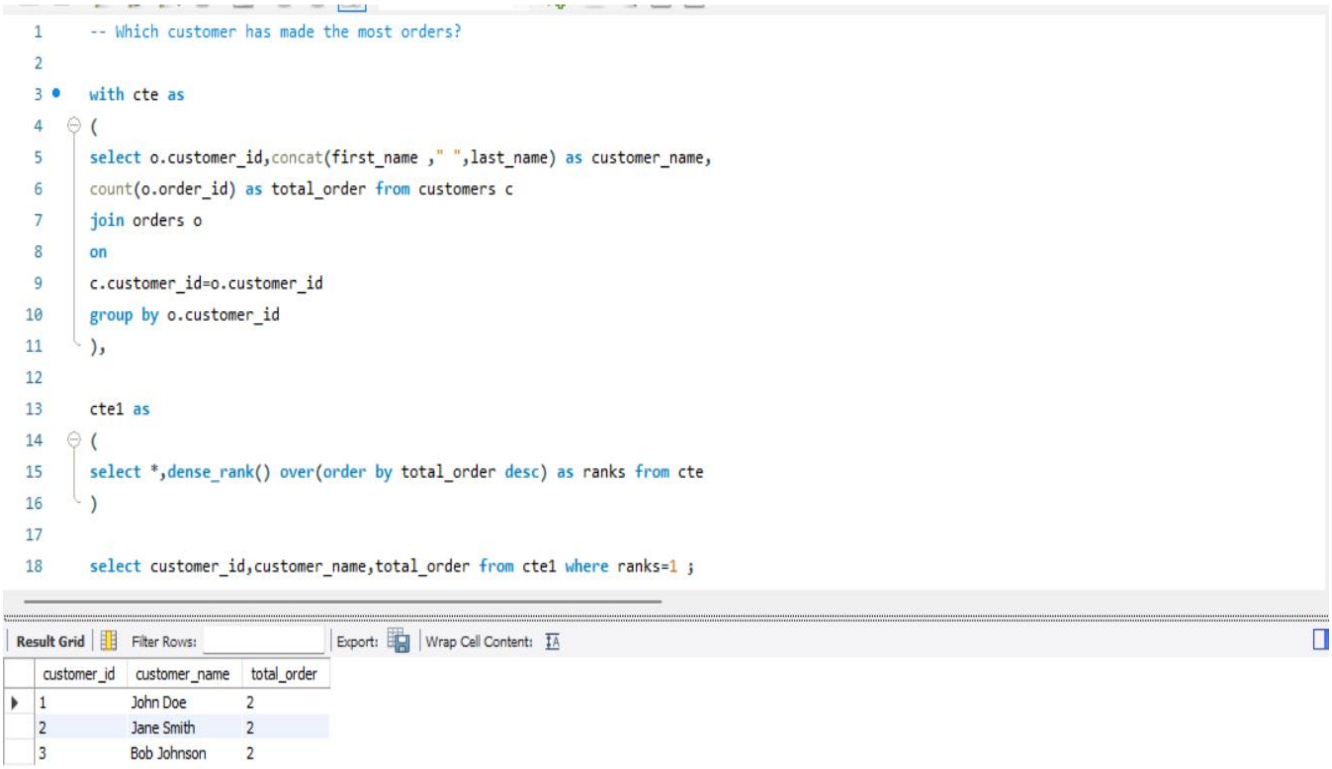


```
-- Which product has the highest price?

SELECT * FROM `tiny shop sales`.products
where price=(select max(price) from `tiny shop sales`.products);
```



product_id	product_name	price
13	Product M	70
NULL	NULL	NULL

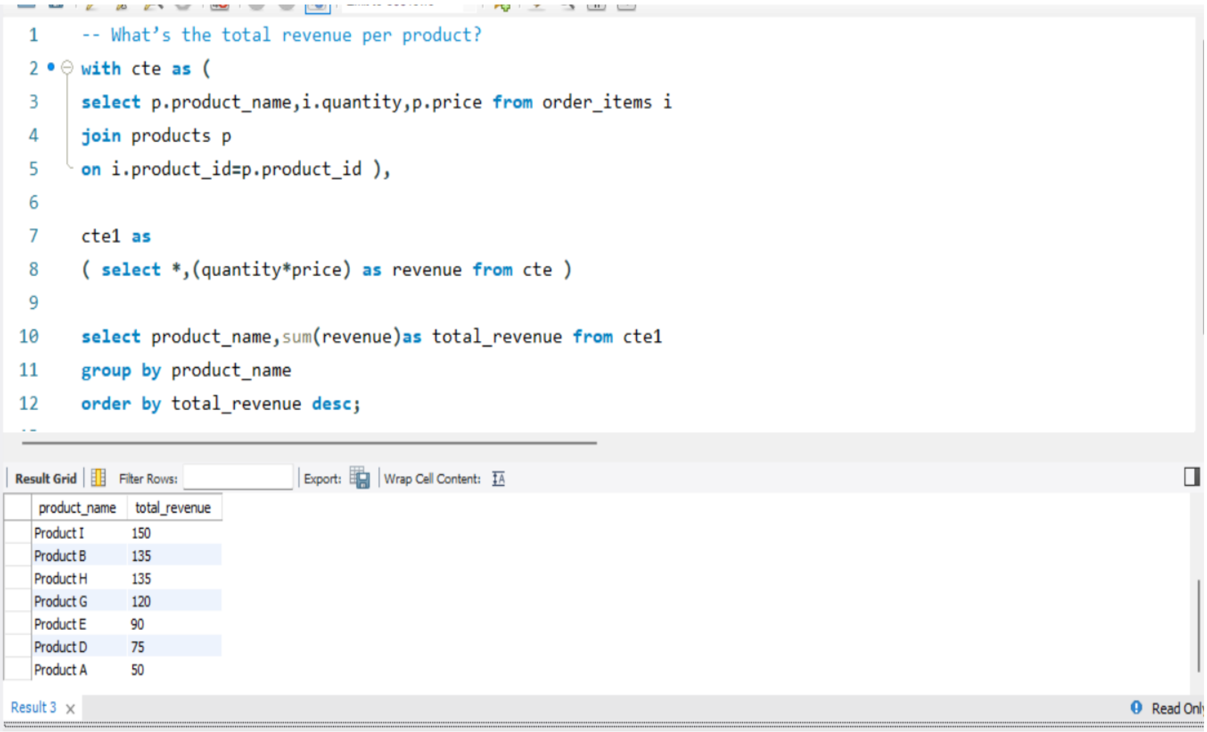
2. Which customer has made the most orders?



```
-- Which customer has made the most orders?

with cte as
(
select o.customer_id,concat(first_name ," ",last_name) as customer_name,
count(o.order_id) as total_order from customers c
join orders o
on
c.customer_id=o.customer_id
group by o.customer_id
),

cte1 as
(
select *,dense_rank() over(order by total_order desc) as ranks from cte
)

select customer_id,customer_name,total_order from cte1 where ranks=1 ;
```

customer_id	customer_name	total_order
1	John Doe	2
2	Jane Smith	2
3	Bob Johnson	2

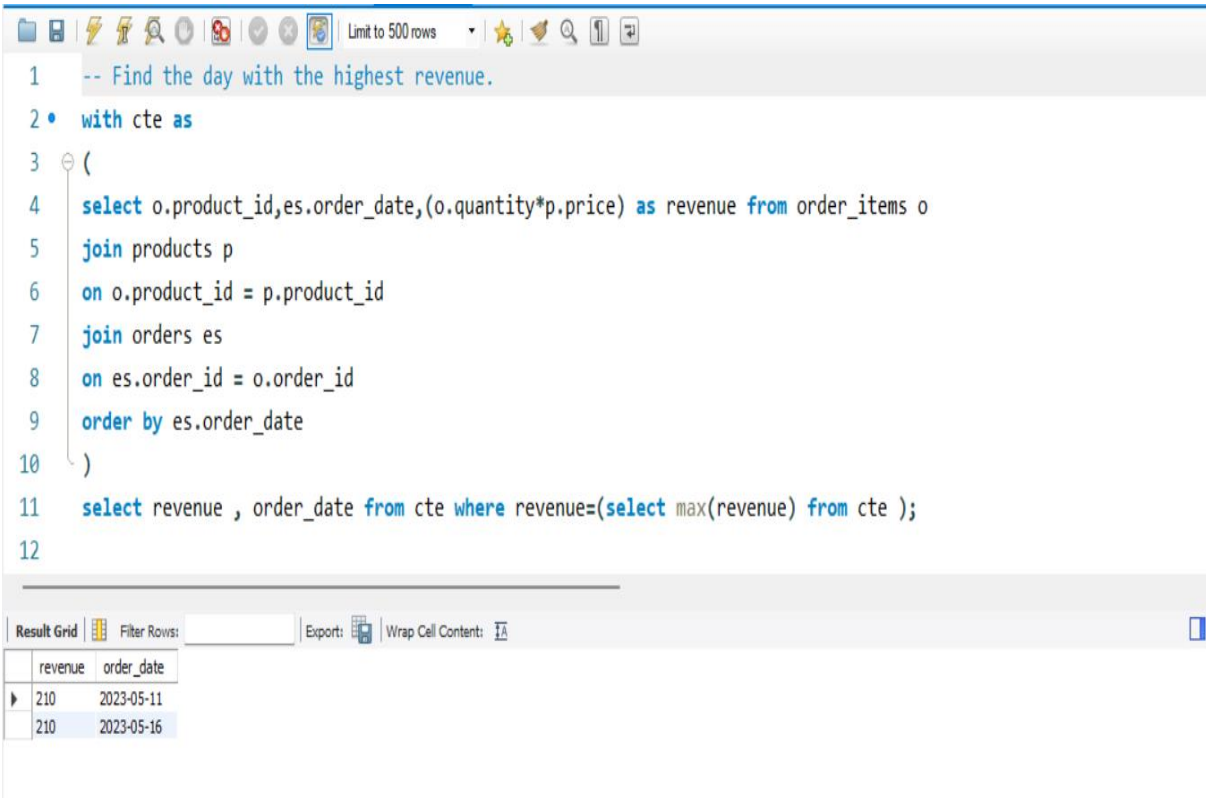
3. What’s the total revenue per product?



```
-- What’s the total revenue per product?
with cte as (
select p.product_name,i.quantity,p.price from order_items i
join products p
on i.product_id=p.product_id ),

cte1 as
( select *,(quantity*price) as revenue from cte )

select product_name,sum(revenue)as total_revenue from cte1
group by product_name
order by total_revenue desc;
```

product_name	total_revenue
Product I	150
Product B	135
Product H	135
Product G	120
Product E	90
Product D	75
Product A	50

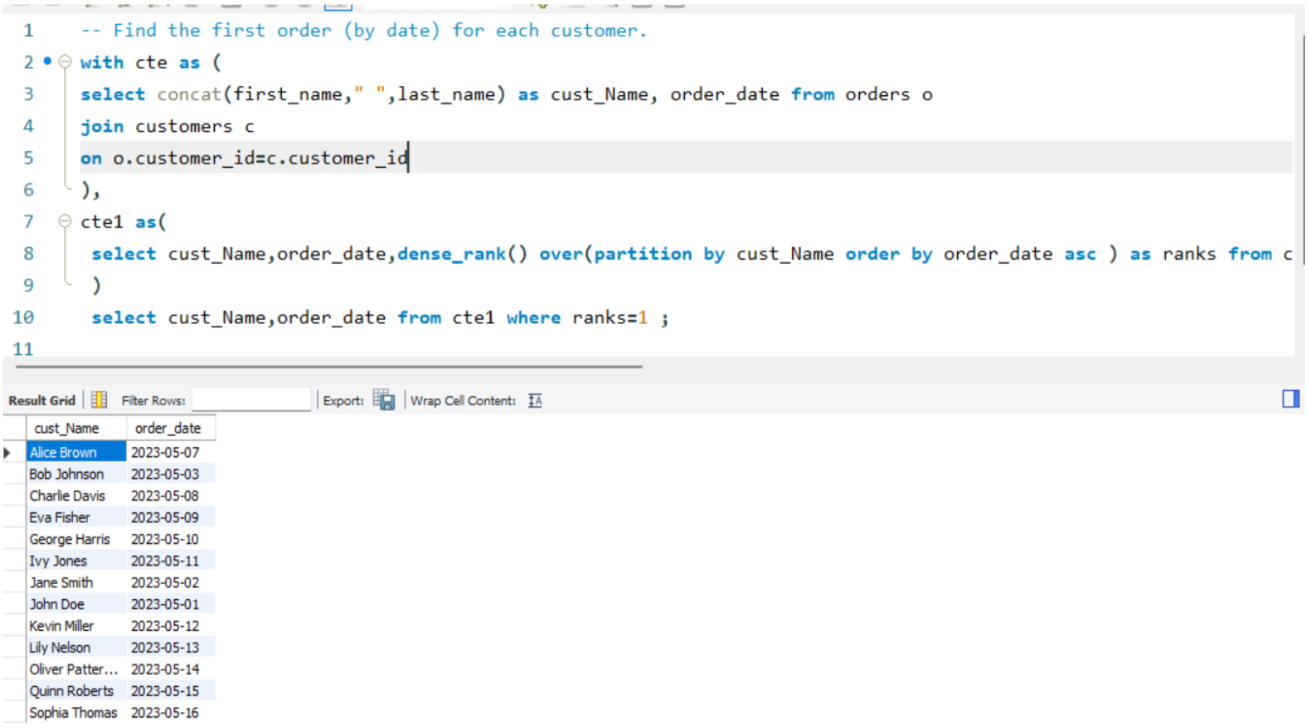
4. Find the day with the highest revenue.



```
-- Find the day with the highest revenue.
with cte as
(
select o.product_id,es.order_date,(o.quantity*p.price) as revenue from order_items o
join products p
on o.product_id = p.product_id
join orders es
on es.order_id = o.order_id
order by es.order_date
)
select revenue , order_date from cte where revenue=(select max(revenue) from cte );
```

revenue	order_date
210	2023-05-11
210	2023-05-16

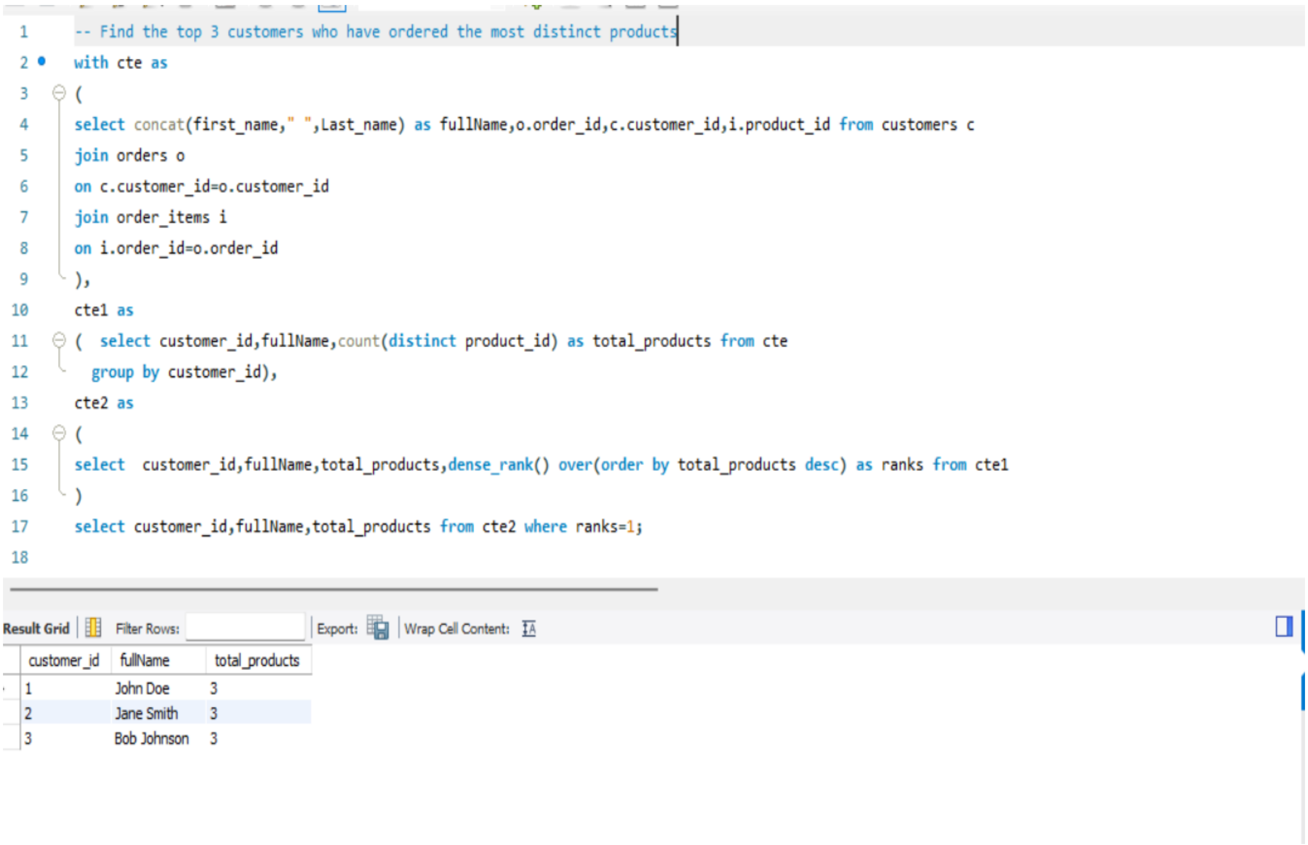
5. Find the first order (by date) for each customer.



```sql
-- Find the first order (by date) for each customer.
with cte as (
select concat(first_name," ",last_name) as cust_Name, order_date from orders o
join customers c
on o.customer_id=c.customer_id
),
cte1 as(
 select cust_Name,order_date,dense_rank() over(partition by cust_Name order by order_date asc ) as ranks from c
 )
 select cust_Name,order_date from cte1 where ranks=1 ;
```

cust_Name	order_date
Alice Brown	2023-05-07
Bob Johnson	2023-05-03
Charlie Davis	2023-05-08
Eva Fisher	2023-05-09
George Harris	2023-05-10
Ivy Jones	2023-05-11
Jane Smith	2023-05-02
John Doe	2023-05-01
Kevin Miller	2023-05-12
Lily Nelson	2023-05-13
Oliver Patter...	2023-05-14
Quinn Roberts	2023-05-15
Sophia Thomas	2023-05-16

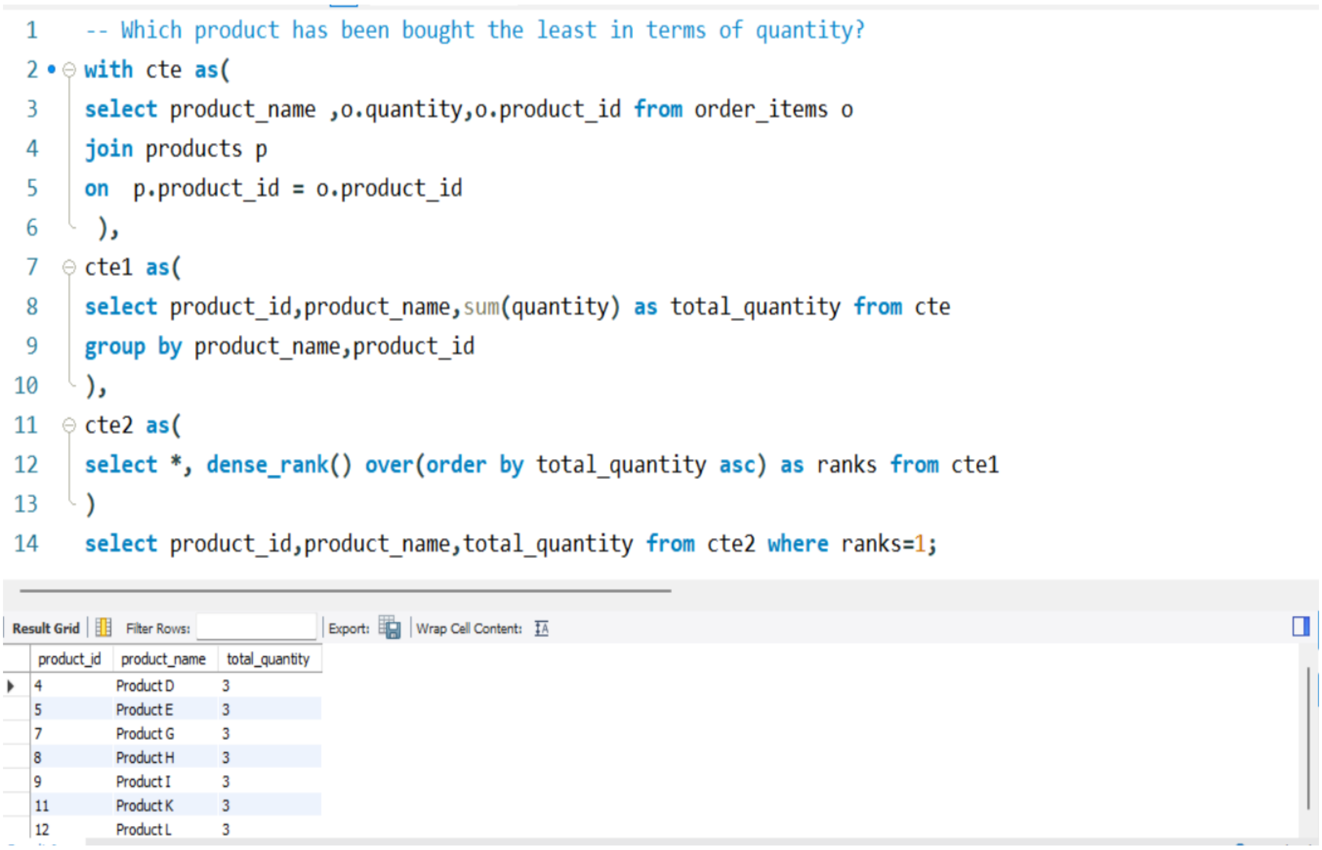
6. Find the top 3 customers who have ordered the most distinct products



```sql
-- Find the top 3 customers who have ordered the most distinct products
with cte as
(
select concat(first_name," ",Last_name) as fullName,o.order_id,c.customer_id,i.product_id from customers c
join orders o
on c.customer_id=o.customer_id
join order_items i
on i.order_id=o.order_id
),
cte1 as
(  select customer_id,fullName,count(distinct product_id) as total_products from cte
  group by customer_id),
cte2 as
(
select  customer_id,fullName,total_products,dense_rank() over(order by total_products desc) as ranks from cte1
)
select customer_id,fullName,total_products from cte2 where ranks=1;
```

customer_id	fullName	total_products
1	John Doe	3
2	Jane Smith	3
3	Bob Johnson	3

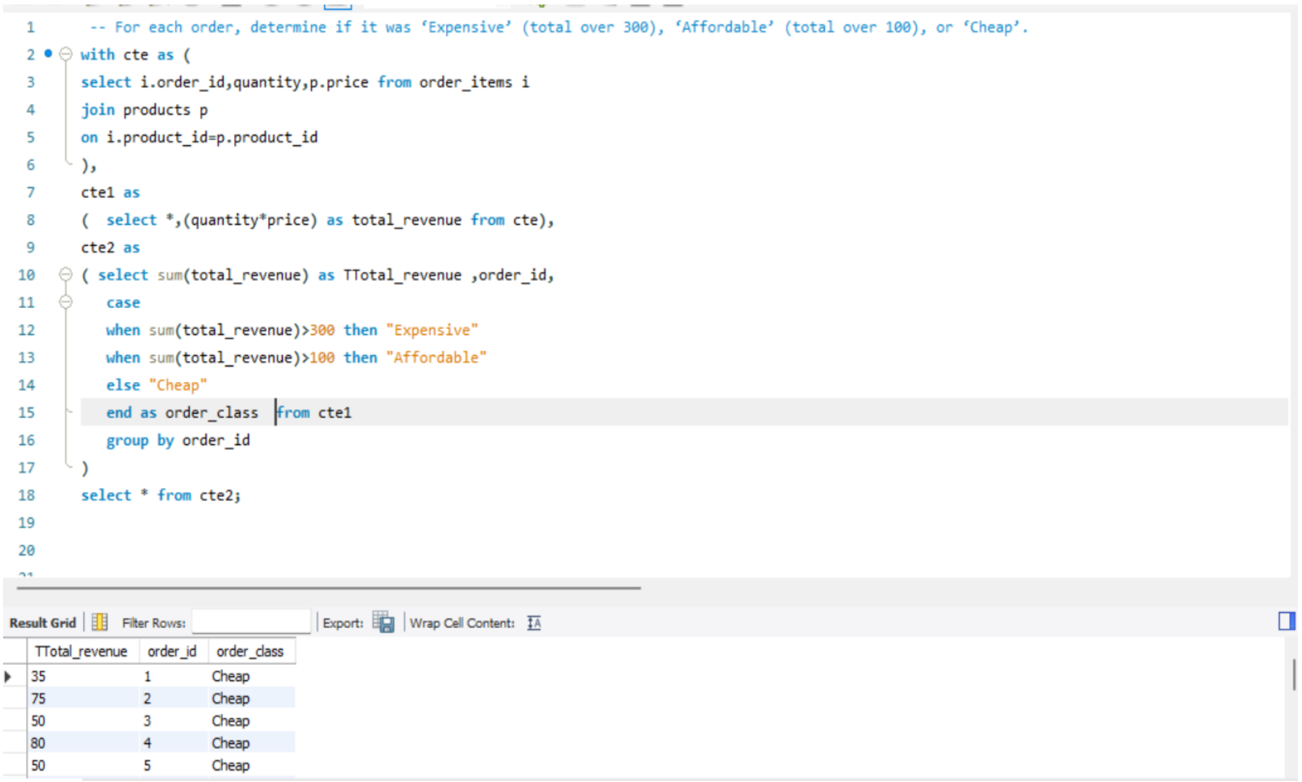
7. Which product has been bought the least in terms of quantity?



```sql
 -- Which product has been bought the least in terms of quantity?
with cte as(
select product_name ,o.quantity,o.product_id from order_items o
join products p
on  p.product_id = o.product_id
 ),
cte1 as(
select product_id,product_name,sum(quantity) as total_quantity from cte
group by product_name,product_id
),
cte2 as(
select *, dense_rank() over(order by total_quantity asc) as ranks from cte1
)
select product_id,product_name,total_quantity from cte2 where ranks=1;
```

product_id	product_name	total_quantity
4	Product D	3
5	Product E	3
7	Product G	3
8	Product H	3
9	Product I	3
11	Product K	3
12	Product L	3

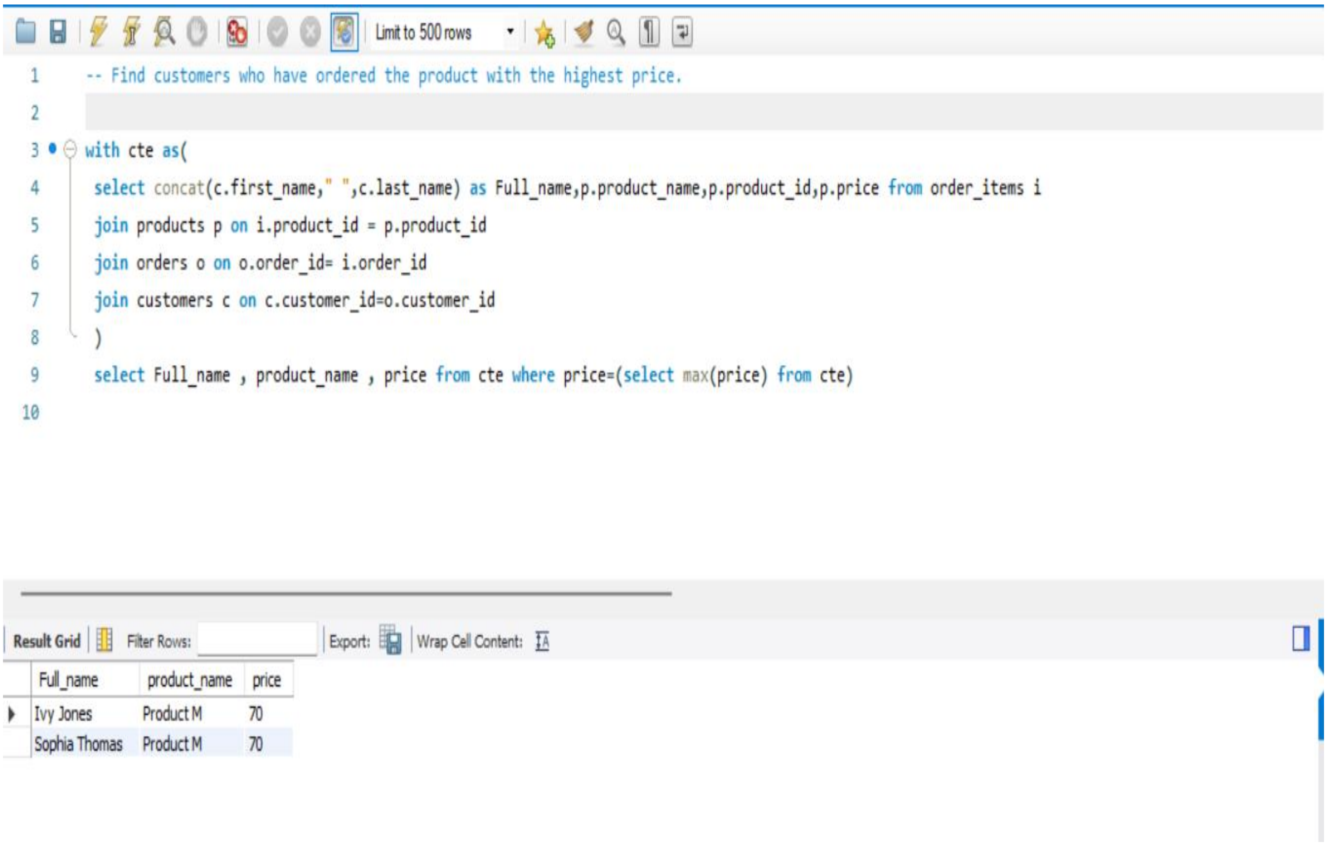
8. For each order, determine if it was ‘Expensive’ (total over 300), ‘Affordable’ (total over 100), or ‘Cheap’.



```sql
 -- For each order, determine if it was ‘Expensive’ (total over 300), ‘Affordable’ (total over 100), or ‘Cheap’.
with cte as (
select i.order_id,quantity,p.price from order_items i
join products p
on i.product_id=p.product_id
),
cte1 as
(  select *,(quantity*price) as total_revenue from cte),
cte2 as
( select sum(total_revenue) as TTotal_revenue ,order_id,
   case
   when sum(total_revenue)>300 then "Expensive"
   when sum(total_revenue)>100 then "Affordable"
   else "Cheap"
   end as order_class  from cte1
   group by order_id
)
select * from cte2;
```

TTotal_revenue	order_id	order_class
35	1	Cheap
75	2	Cheap
50	3	Cheap
80	4	Cheap
50	5	Cheap

9. Find customers who have ordered the product with the highest price.



```
-- Find customers who have ordered the product with the highest price.

with cte as(
 select concat(c.first_name," ",c.last_name) as Full_name,p.product_name,p.product_id,p.price from order_items i
 join products p on i.product_id = p.product_id
 join orders o on o.order_id= i.order_id
 join customers c on c.customer_id=o.customer_id
 )
 select Full_name , product_name , price from cte where price=(select max(price) from cte)
```

Full_name	product_name	price
Ivy Jones	Product M	70
Sophia Thomas	Product M	70