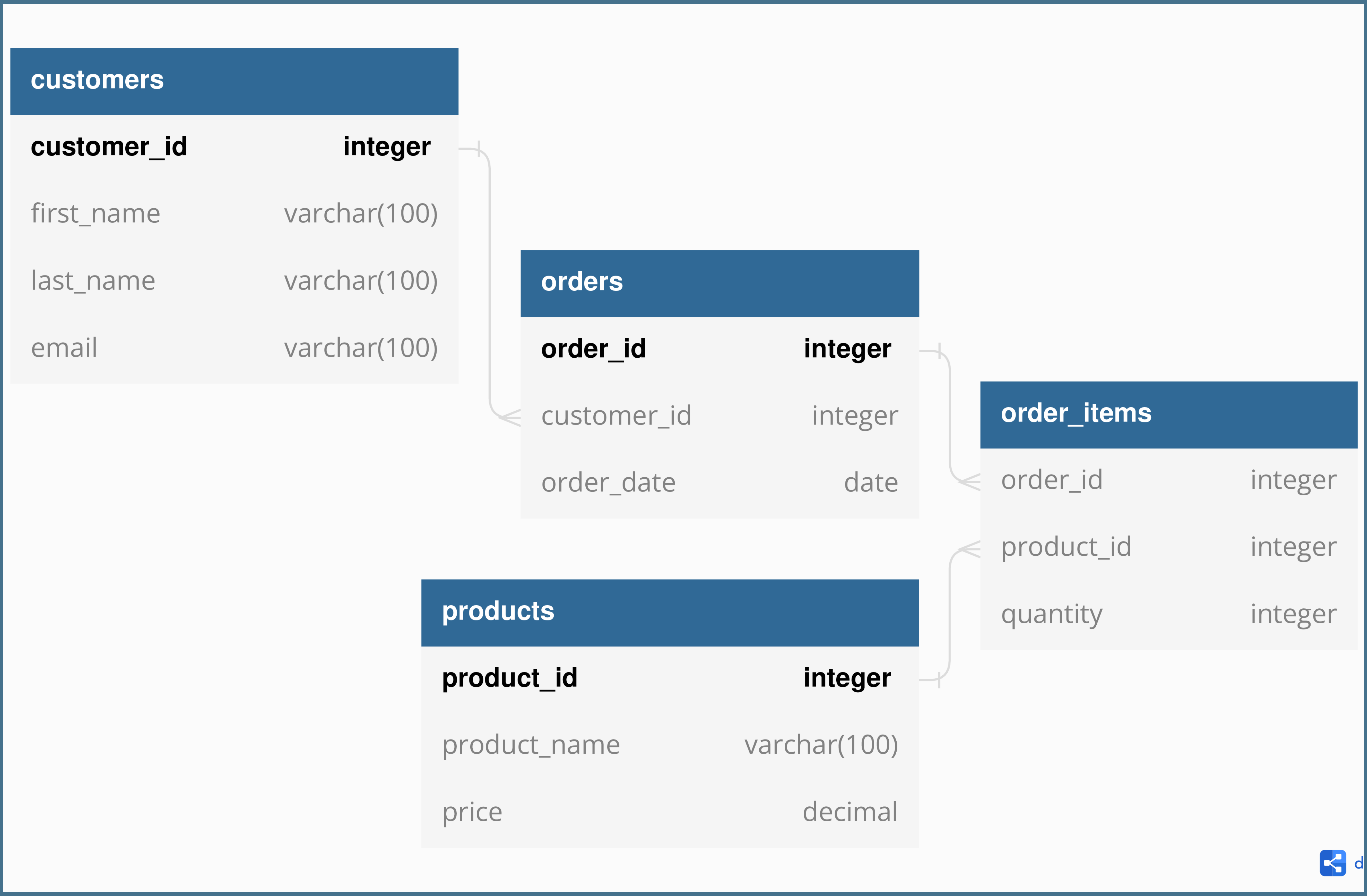
SQL CASE STUDY

DATA IN MOTION
TINY SHOP SALES

DATA IN MOTION

BY GLADYS CORTES

ER Diagram

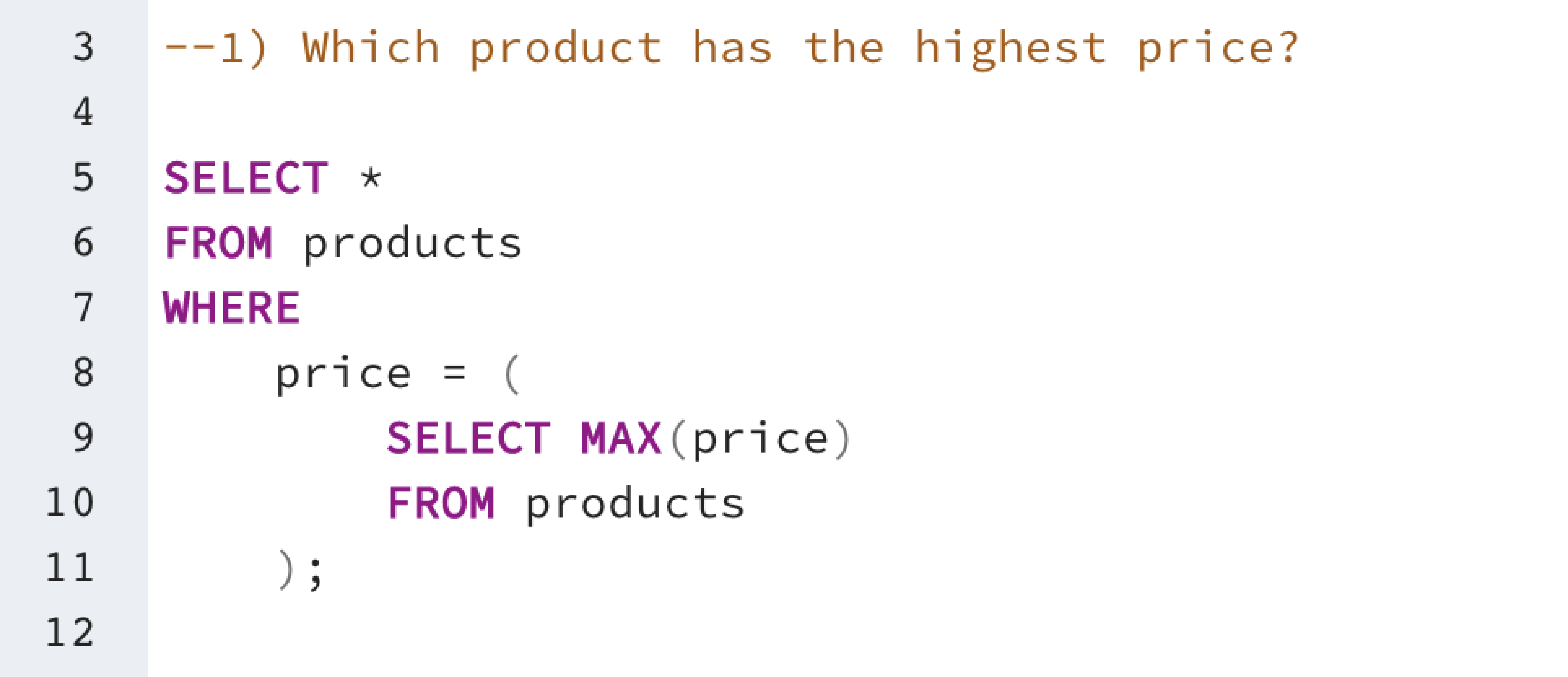


customers	
customer_id	**integer**
first_name	varchar(100)
last_name	varchar(100)
email	varchar(100)

orders	
order_id	**integer**
customer_id	integer
order_date	date

order_items	
order_id	integer
product_id	integer
quantity	integer

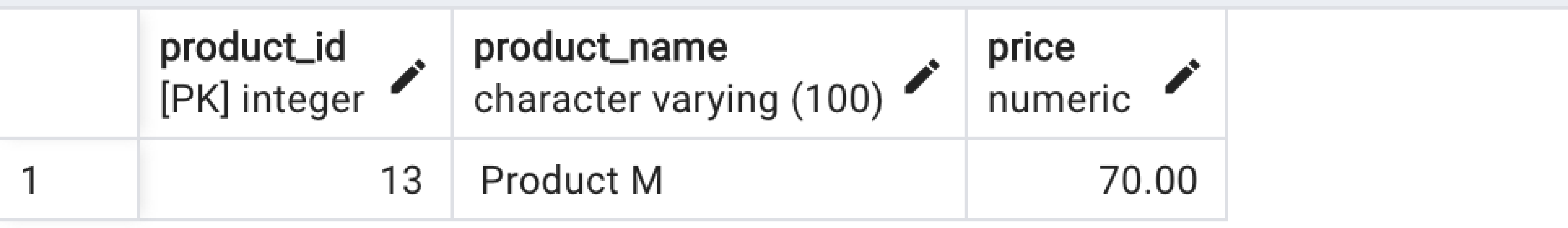
products	
product_id	**integer**
product_name	varchar(100)
price	decimal



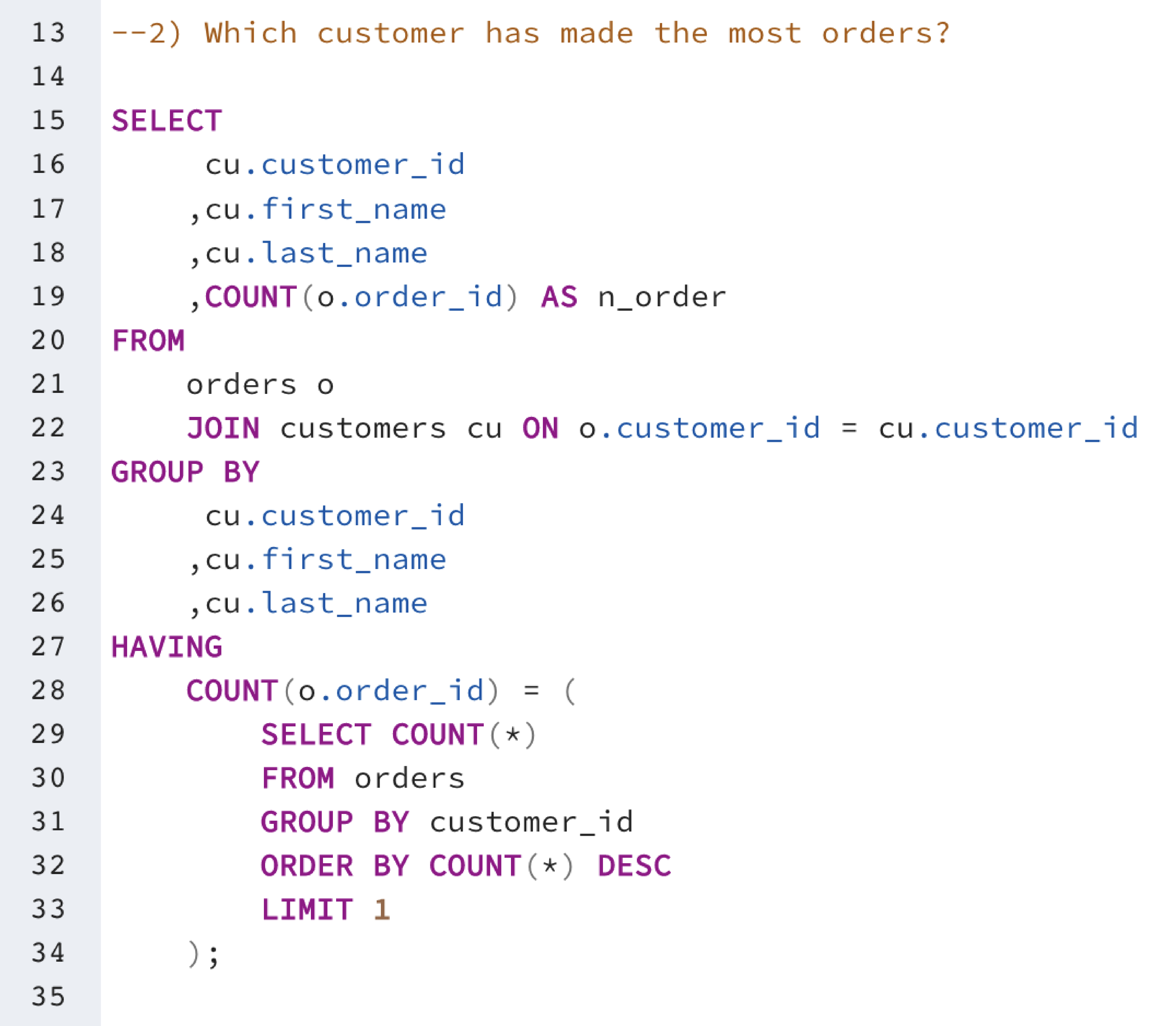
```sql
--1) Which product has the highest price?

SELECT *
FROM products
WHERE
    price = (
        SELECT MAX(price)
        FROM products
    );

```



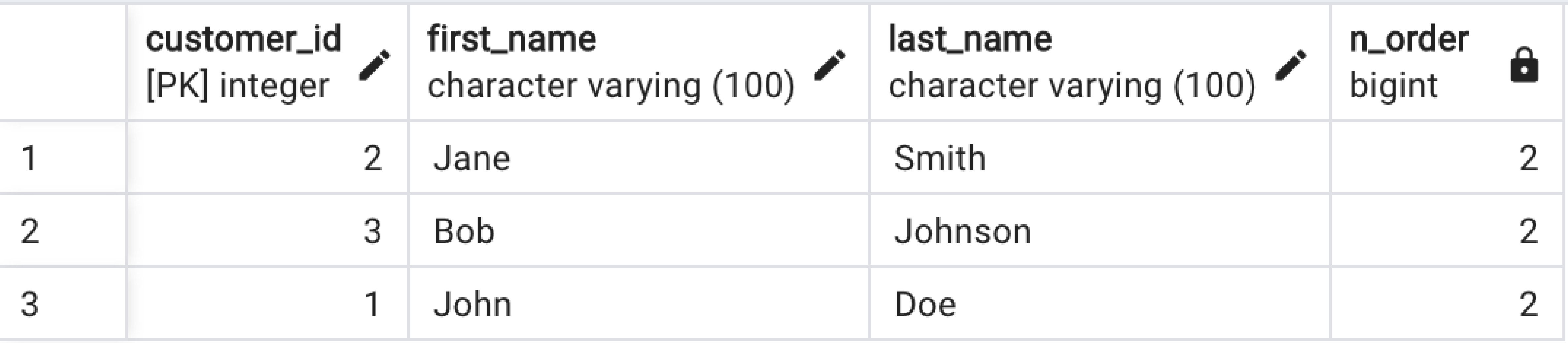
	product_id [PK] integer	product_name character varying (100)	price numeric
1	13	Product M	70.00



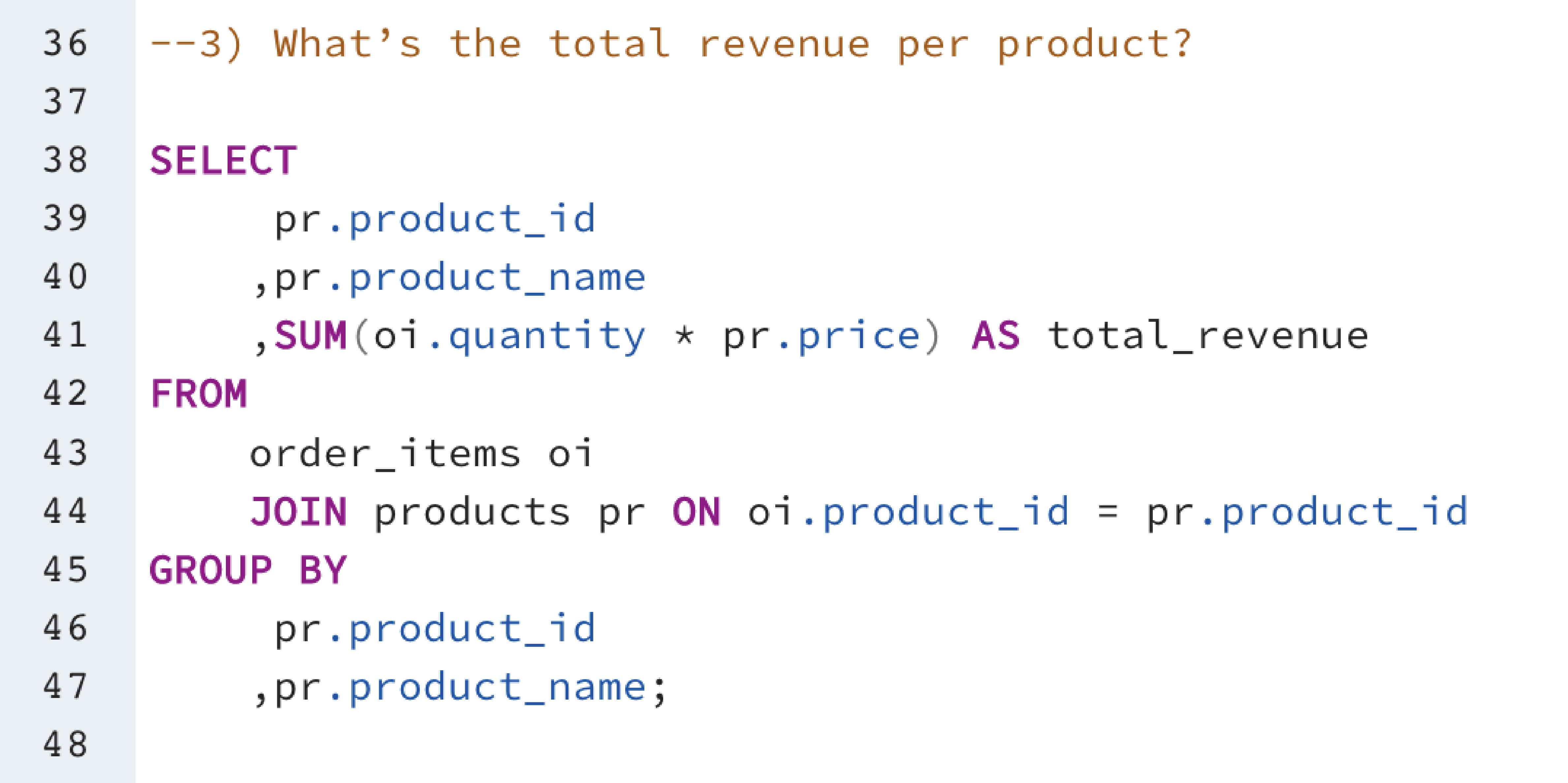
```sql
--2) Which customer has made the most orders?

SELECT
     cu.customer_id
    ,cu.first_name
    ,cu.last_name
    ,COUNT(o.order_id) AS n_order
FROM
    orders o
    JOIN customers cu ON o.customer_id = cu.customer_id
GROUP BY
     cu.customer_id
    ,cu.first_name
    ,cu.last_name
HAVING
    COUNT(o.order_id) = (
        SELECT COUNT(*)
        FROM orders
        GROUP BY customer_id
        ORDER BY COUNT(*) DESC
        LIMIT 1
    );

```



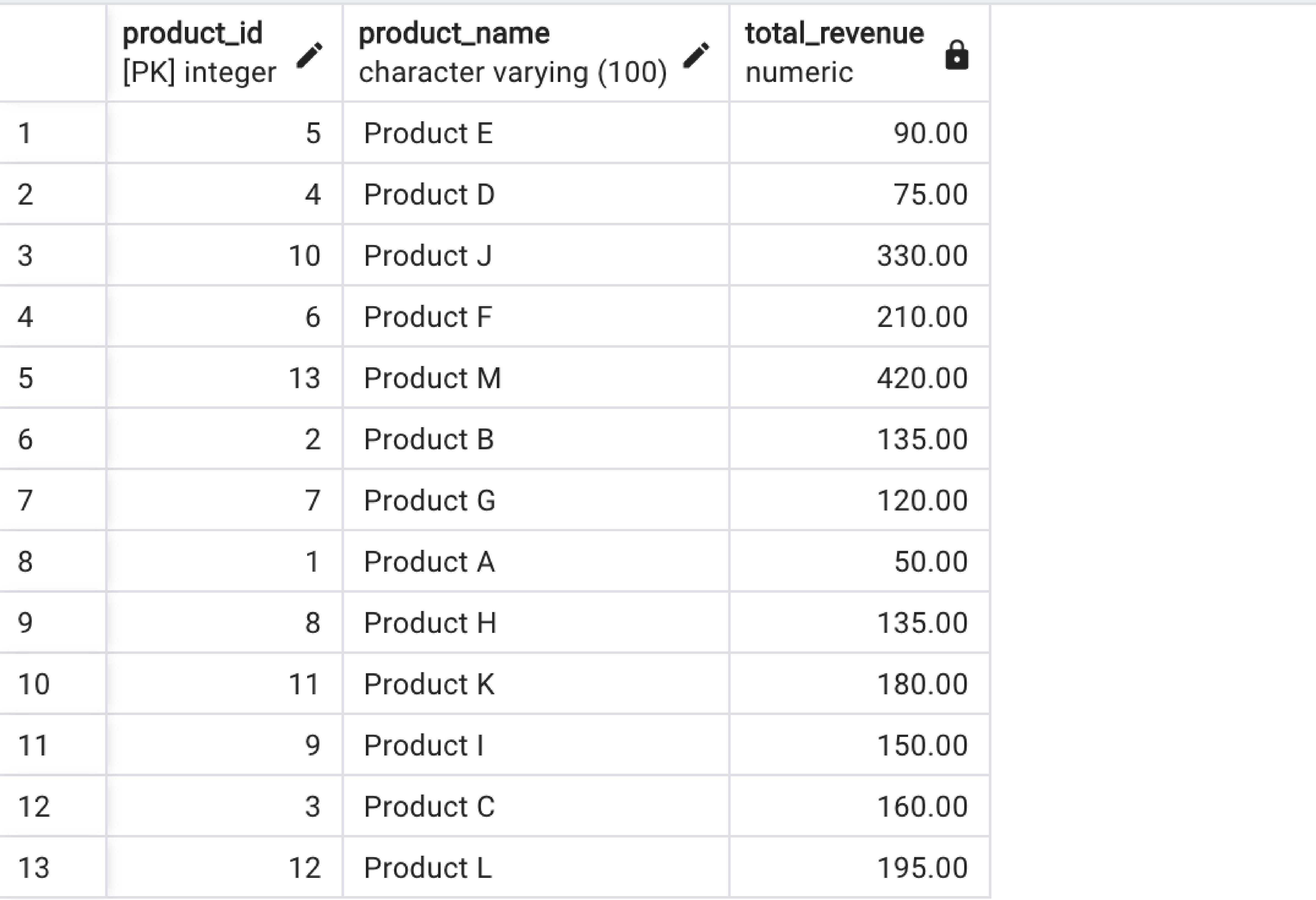
	customer_id [PK] integer	first_name character varying (100)	last_name character varying (100)	n_order bigint
1	2	Jane	Smith	2
2	3	Bob	Johnson	2
3	1	John	Doe	2



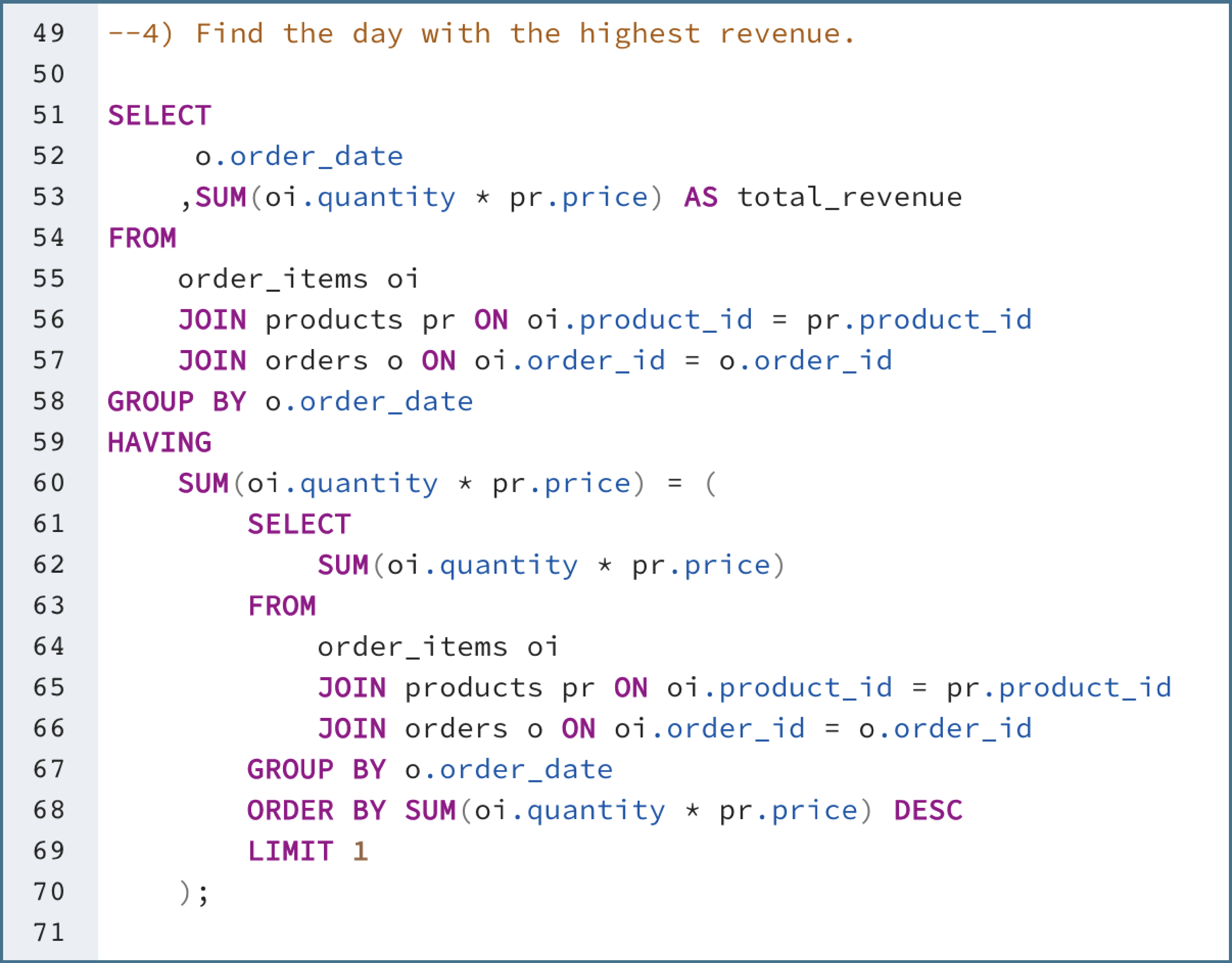
```sql
--3) What's the total revenue per product?

SELECT
     pr.product_id
    ,pr.product_name
    ,SUM(oi.quantity * pr.price) AS total_revenue
FROM
    order_items oi
    JOIN products pr ON oi.product_id = pr.product_id
GROUP BY
     pr.product_id
    ,pr.product_name;

```

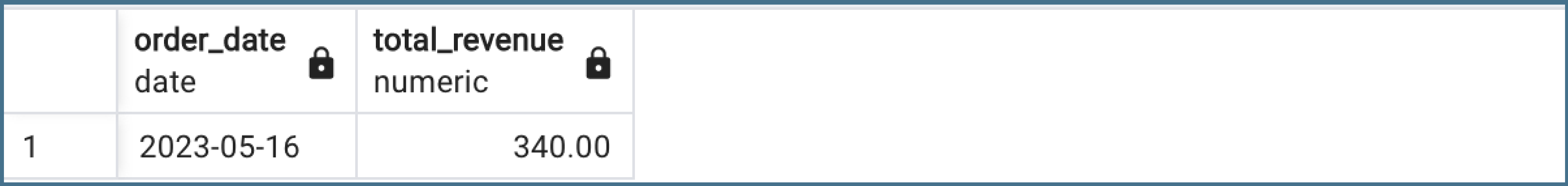


	product_id [PK] integer	product_name character varying (100)	total_revenue numeric
1	5	Product E	90.00
2	4	Product D	75.00
3	10	Product J	330.00
4	6	Product F	210.00
5	13	Product M	420.00
6	2	Product B	135.00
7	7	Product G	120.00
8	1	Product A	50.00
9	8	Product H	135.00
10	11	Product K	180.00
11	9	Product I	150.00
12	3	Product C	160.00
13	12	Product L	195.00

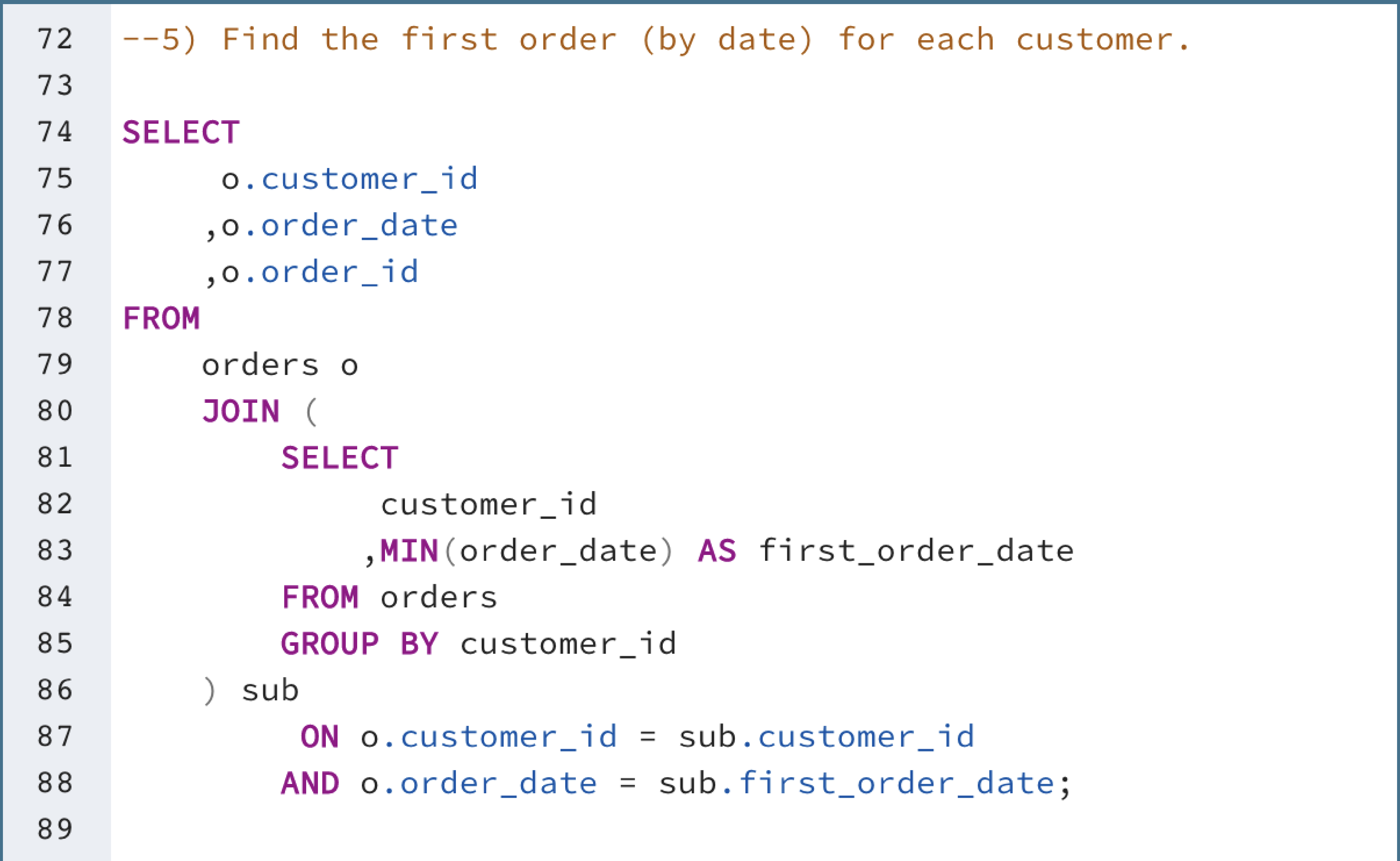


```sql
--4) Find the day with the highest revenue.

SELECT
     o.order_date
    ,SUM(oi.quantity * pr.price) AS total_revenue
FROM
    order_items oi
    JOIN products pr ON oi.product_id = pr.product_id
    JOIN orders o ON oi.order_id = o.order_id
GROUP BY o.order_date
HAVING
    SUM(oi.quantity * pr.price) = (
        SELECT
            SUM(oi.quantity * pr.price)
        FROM
            order_items oi
            JOIN products pr ON oi.product_id = pr.product_id
            JOIN orders o ON oi.order_id = o.order_id
        GROUP BY o.order_date
        ORDER BY SUM(oi.quantity * pr.price) DESC
        LIMIT 1
    );
```



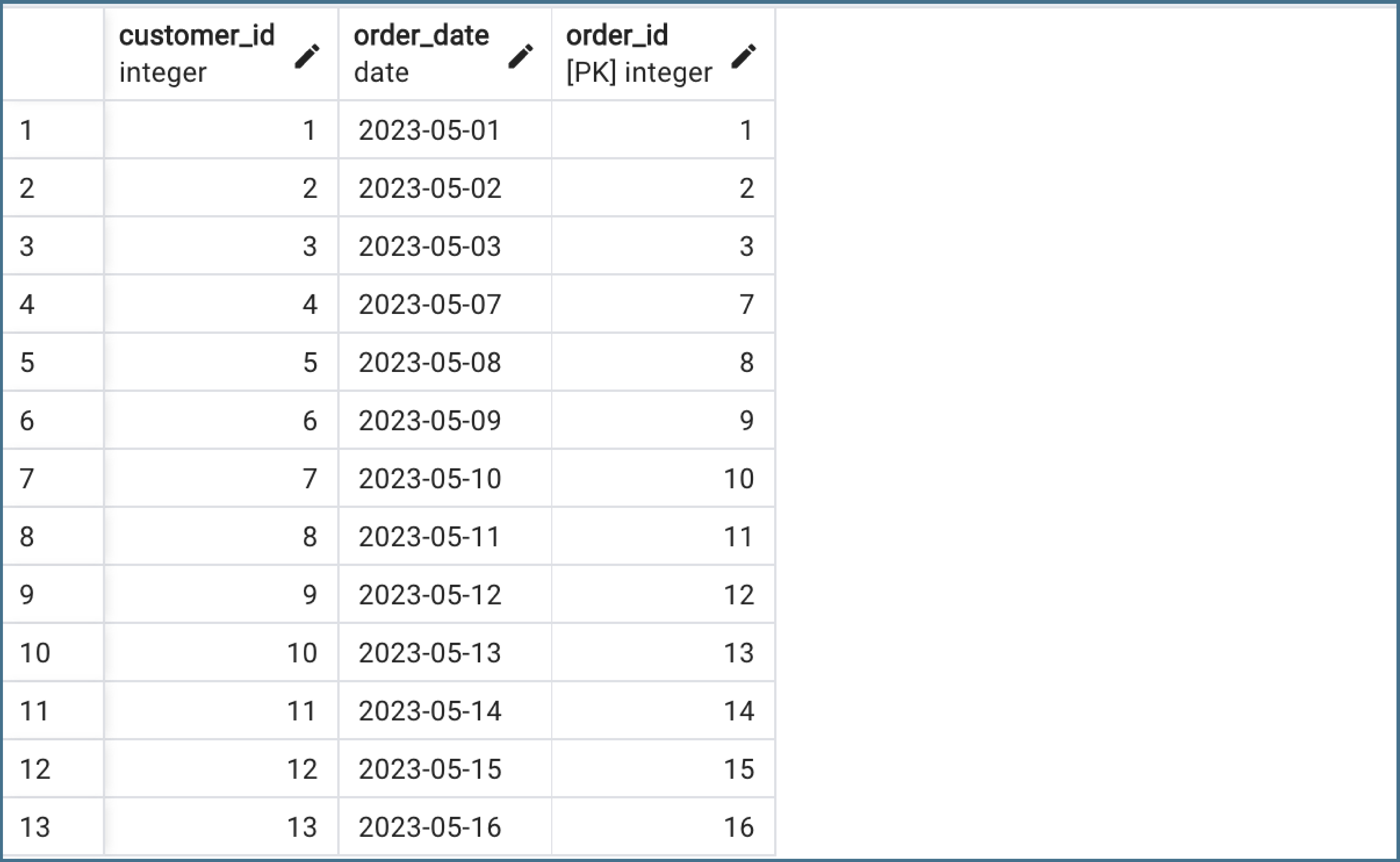
	order_date date	total_revenue numeric
1	2023-05-16	340.00



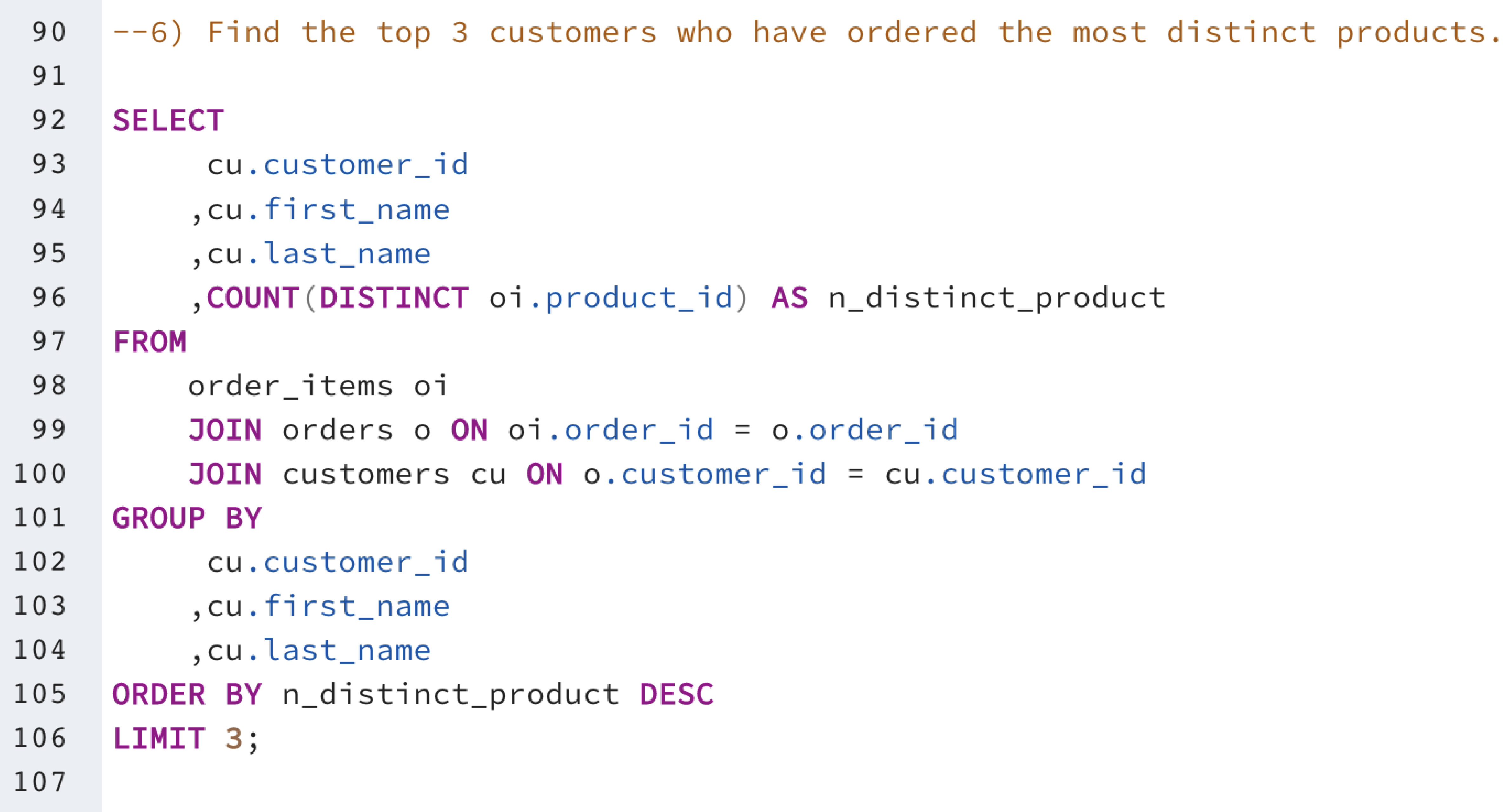
```sql
--5) Find the first order (by date) for each customer.

SELECT
     o.customer_id
    ,o.order_date
    ,o.order_id
FROM
    orders o
    JOIN (
        SELECT
             customer_id
            ,MIN(order_date) AS first_order_date
        FROM orders
        GROUP BY customer_id
    ) sub
         ON o.customer_id = sub.customer_id
        AND o.order_date = sub.first_order_date;

```



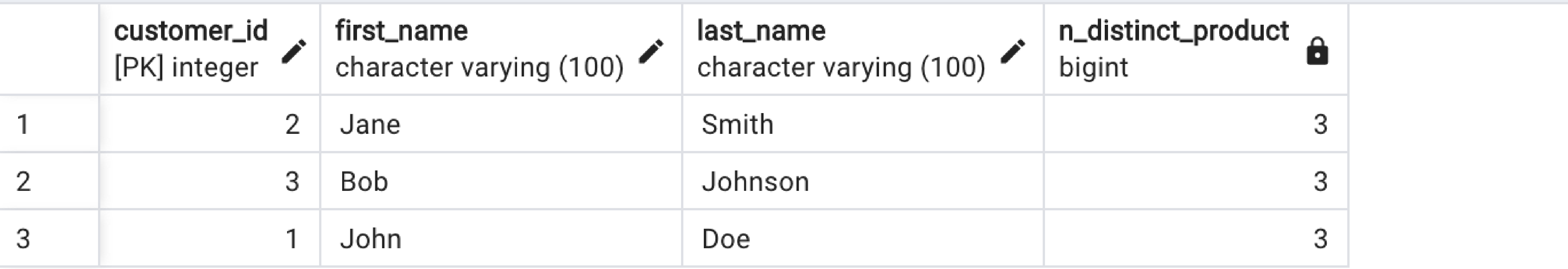
	customer_id integer	order_date date	order_id [PK] integer
1	1	2023-05-01	1
2	2	2023-05-02	2
3	3	2023-05-03	3
4	4	2023-05-07	7
5	5	2023-05-08	8
6	6	2023-05-09	9
7	7	2023-05-10	10
8	8	2023-05-11	11
9	9	2023-05-12	12
10	10	2023-05-13	13
11	11	2023-05-14	14
12	12	2023-05-15	15
13	13	2023-05-16	16



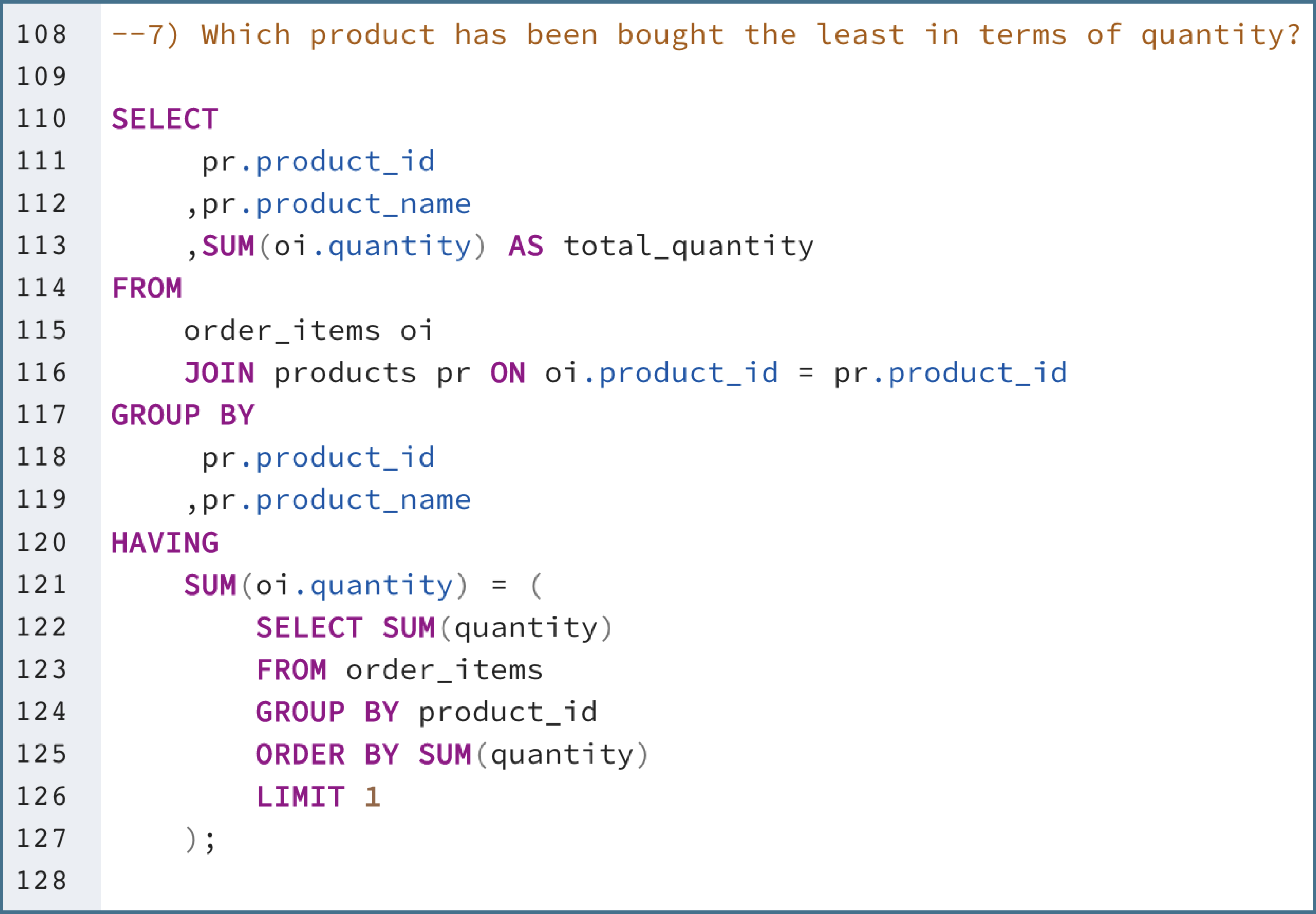
```sql
--6) Find the top 3 customers who have ordered the most distinct products.

SELECT
     cu.customer_id
    ,cu.first_name
    ,cu.last_name
    ,COUNT(DISTINCT oi.product_id) AS n_distinct_product
FROM
    order_items oi
    JOIN orders o ON oi.order_id = o.order_id
    JOIN customers cu ON o.customer_id = cu.customer_id
GROUP BY
     cu.customer_id
    ,cu.first_name
    ,cu.last_name
ORDER BY n_distinct_product DESC
LIMIT 3;

```



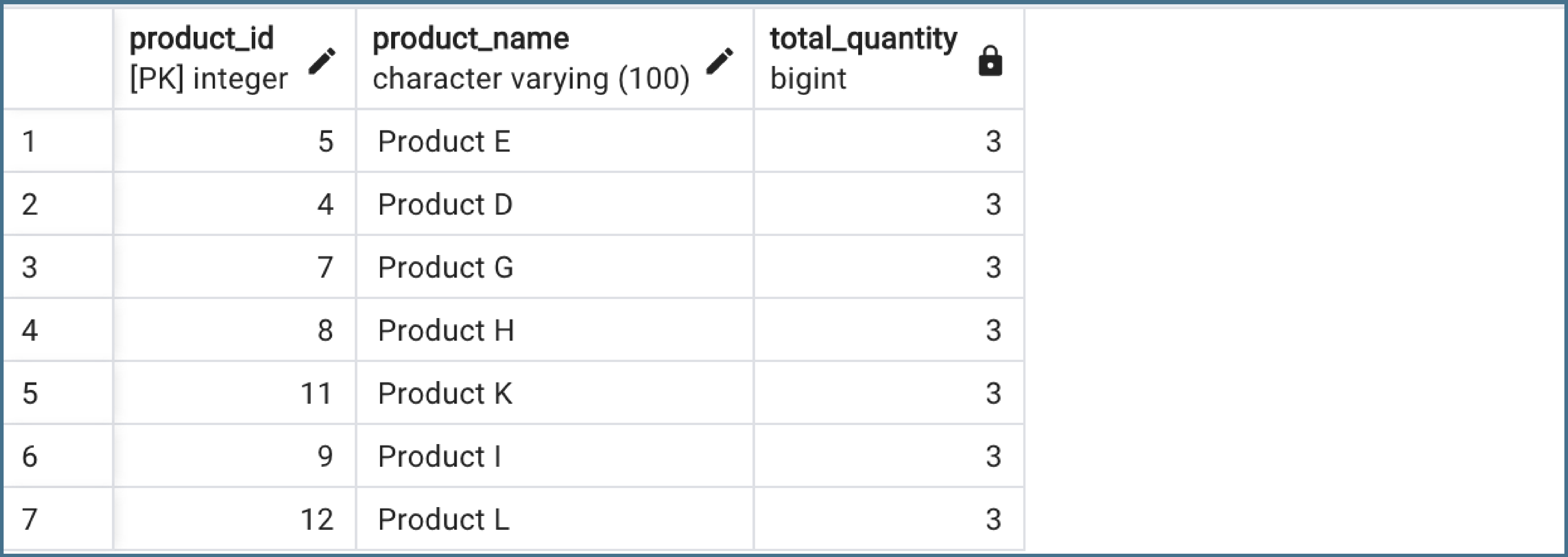
	customer_id [PK] integer	first_name character varying (100)	last_name character varying (100)	n_distinct_product bigint
1	2	Jane	Smith	3
2	3	Bob	Johnson	3
3	1	John	Doe	3



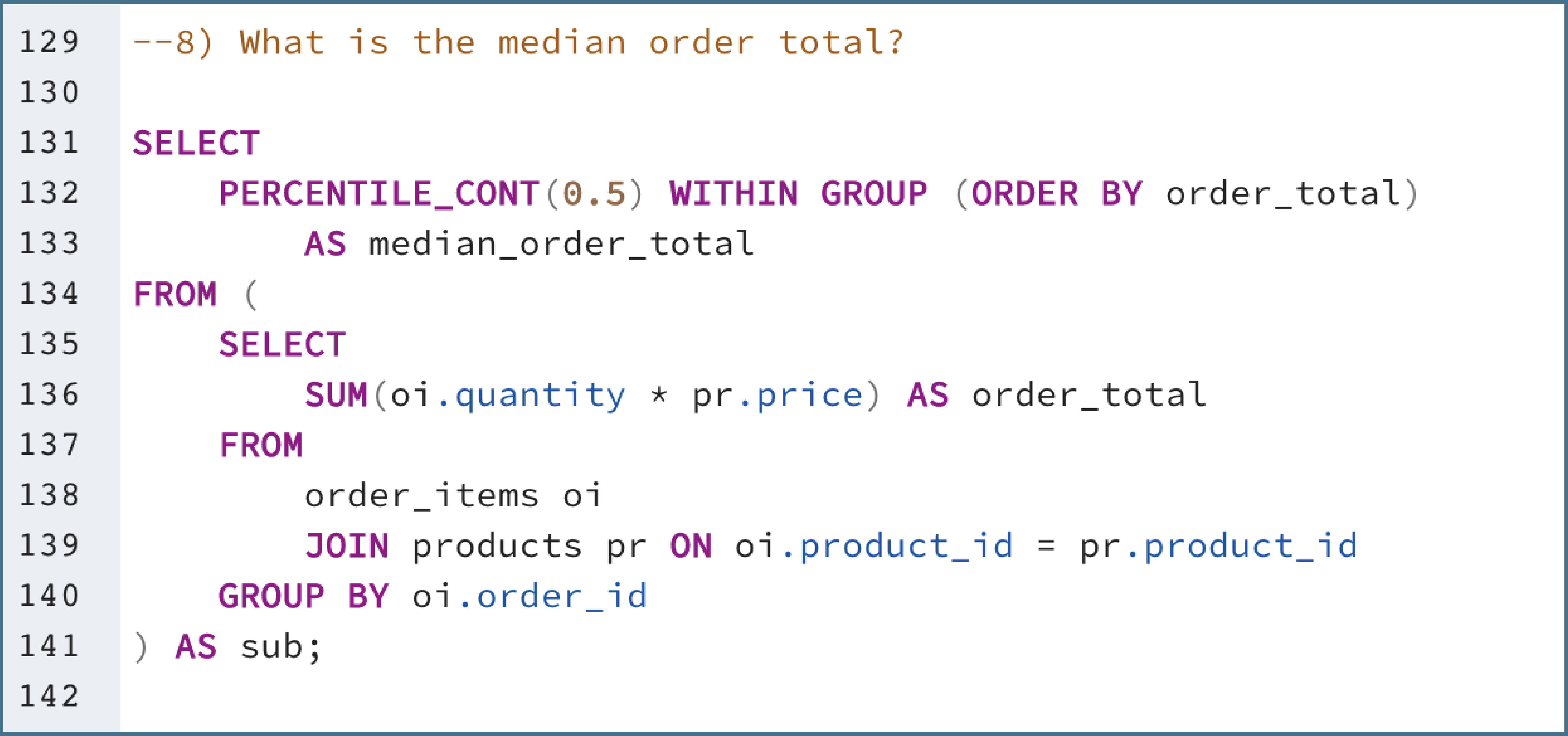
```sql
--7) Which product has been bought the least in terms of quantity?

SELECT
     pr.product_id
    ,pr.product_name
    ,SUM(oi.quantity) AS total_quantity
FROM
    order_items oi
    JOIN products pr ON oi.product_id = pr.product_id
GROUP BY
     pr.product_id
    ,pr.product_name
HAVING
    SUM(oi.quantity) = (
        SELECT SUM(quantity)
        FROM order_items
        GROUP BY product_id
        ORDER BY SUM(quantity)
        LIMIT 1
    );

```

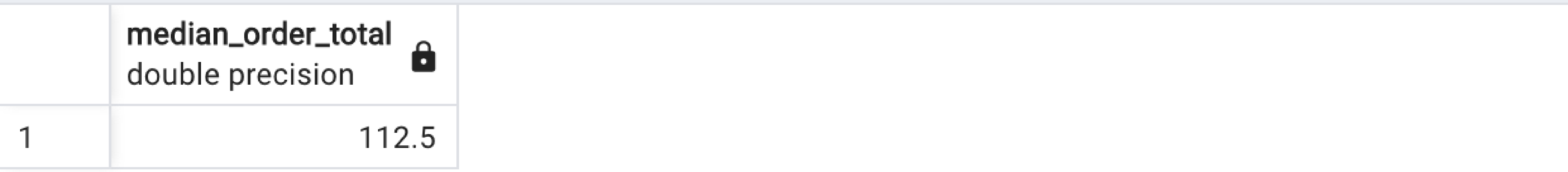


	product_id [PK] integer	product_name character varying (100)	total_quantity bigint
1	5	Product E	3
2	4	Product D	3
3	7	Product G	3
4	8	Product H	3
5	11	Product K	3
6	9	Product I	3
7	12	Product L	3

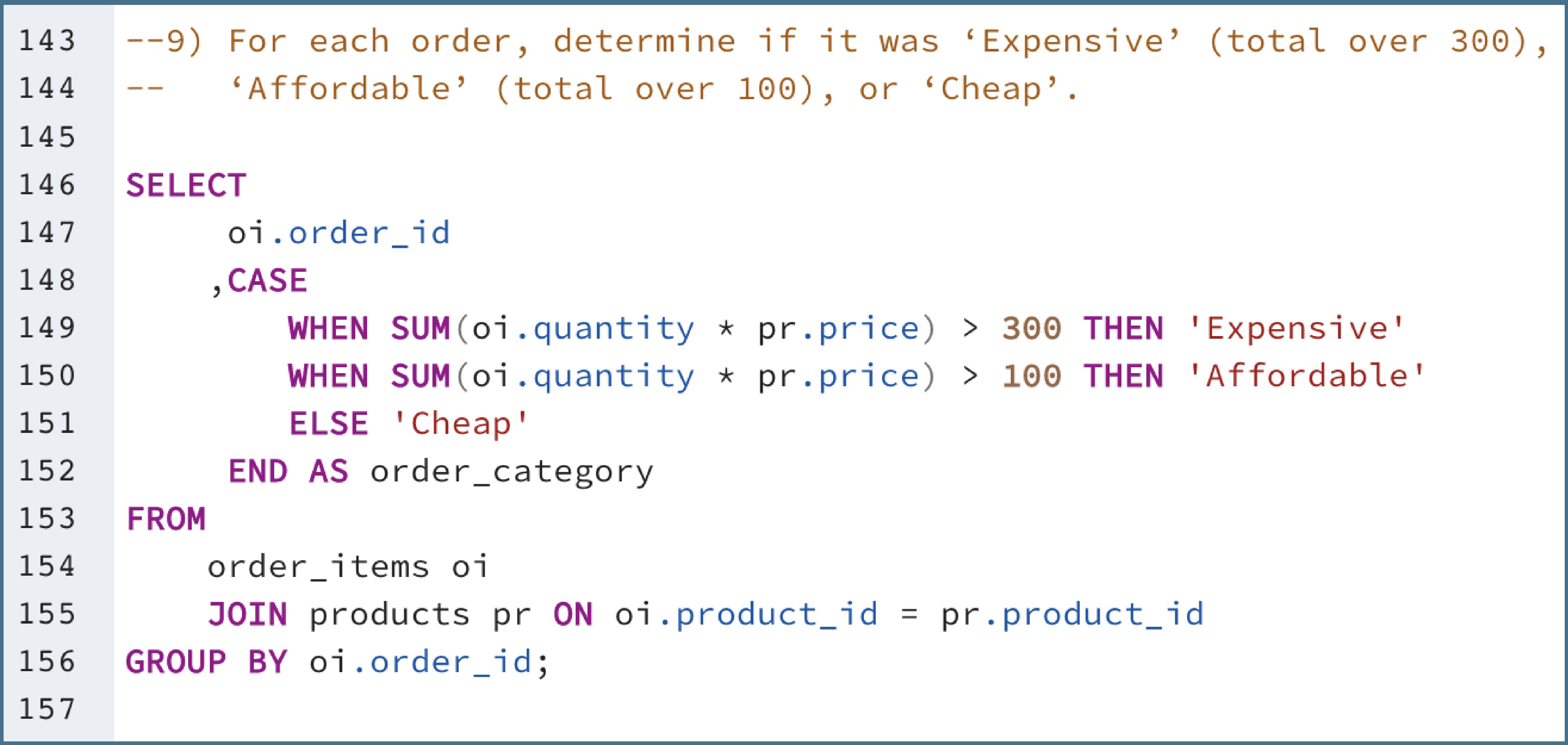


```sql
--8) What is the median order total?

SELECT
    PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY order_total)
        AS median_order_total
FROM (
    SELECT
        SUM(oi.quantity * pr.price) AS order_total
    FROM
        order_items oi
        JOIN products pr ON oi.product_id = pr.product_id
    GROUP BY oi.order_id
) AS sub;
```

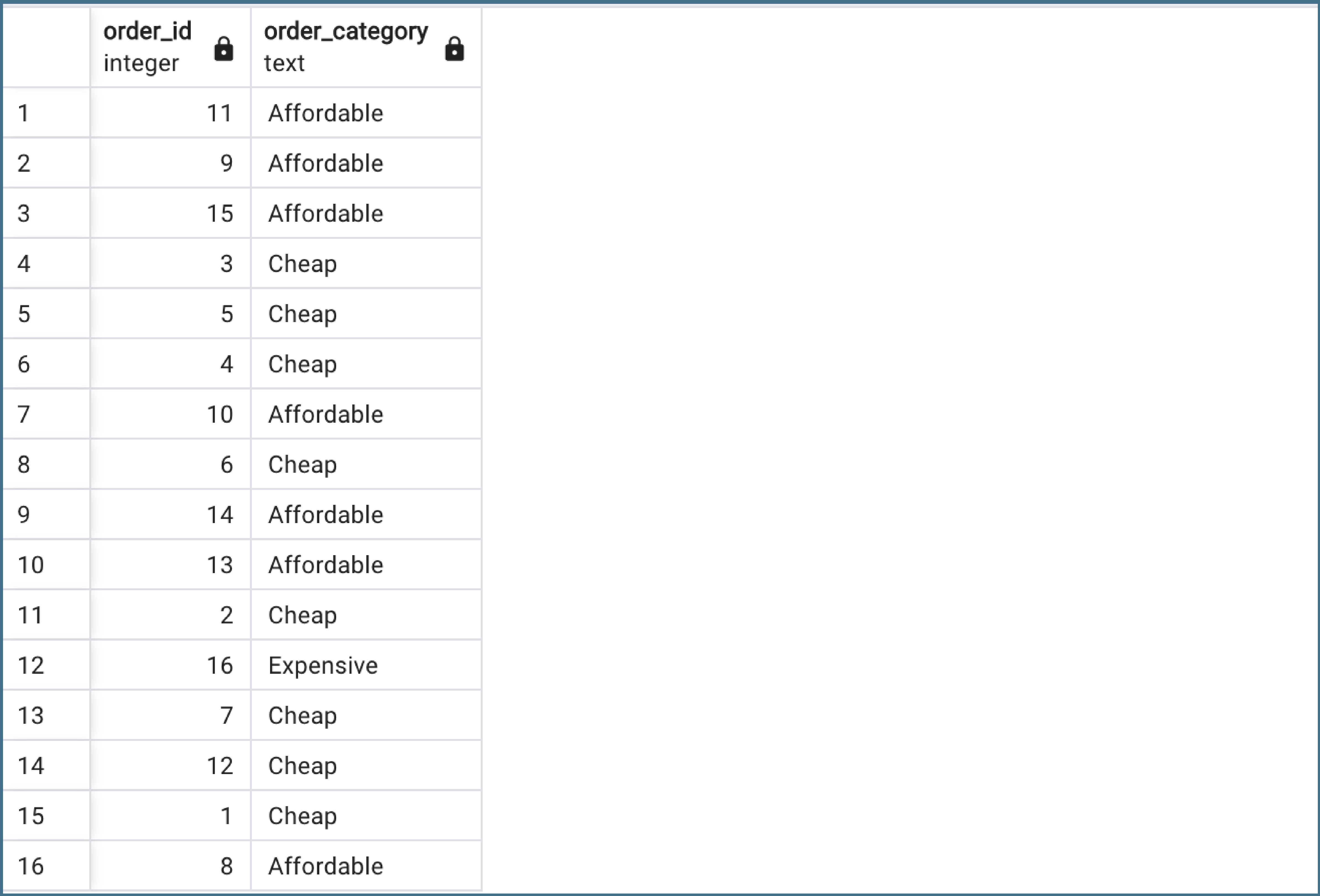


	median_order_total double precision
1	112.5

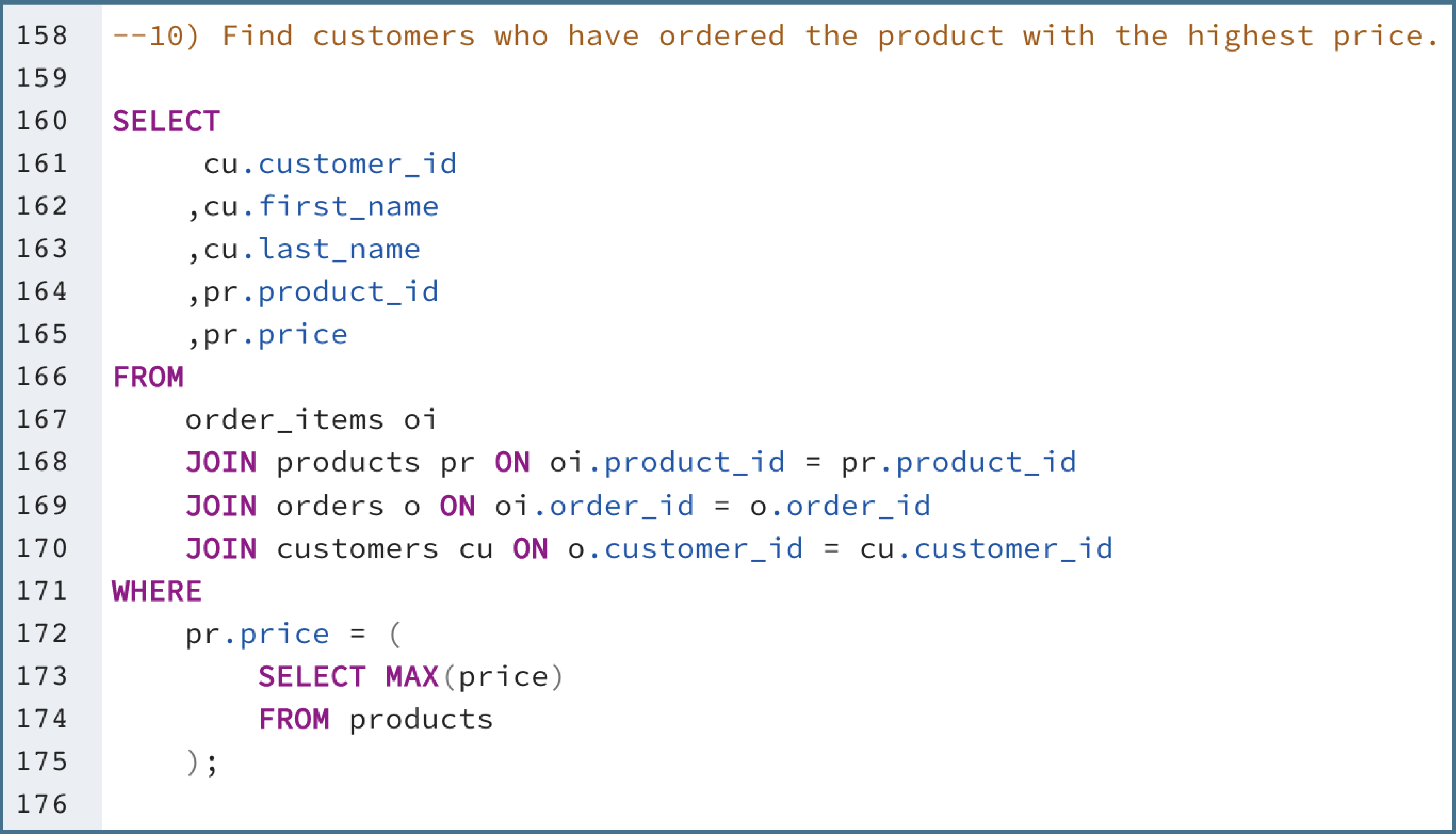


```sql
--9) For each order, determine if it was ‘Expensive’ (total over 300),
--   ‘Affordable’ (total over 100), or ‘Cheap’.

SELECT
     oi.order_id
    ,CASE
        WHEN SUM(oi.quantity * pr.price) > 300 THEN 'Expensive'
        WHEN SUM(oi.quantity * pr.price) > 100 THEN 'Affordable'
        ELSE 'Cheap'
     END AS order_category
FROM
    order_items oi
    JOIN products pr ON oi.product_id = pr.product_id
GROUP BY oi.order_id;
```



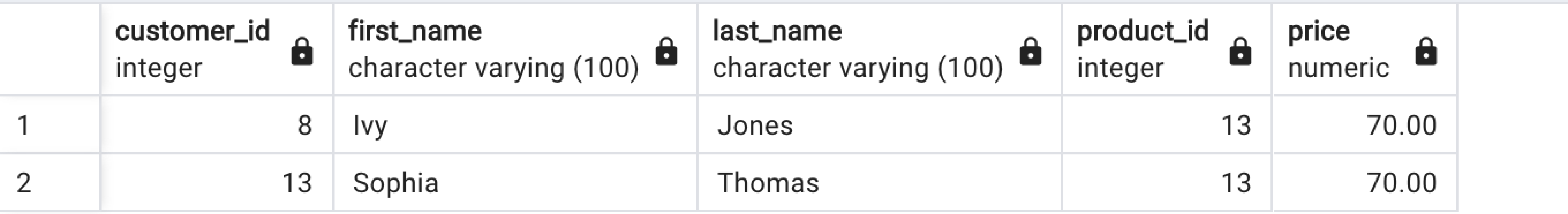
	order_id integer	order_category text
1	11	Affordable
2	9	Affordable
3	15	Affordable
4	3	Cheap
5	5	Cheap
6	4	Cheap
7	10	Affordable
8	6	Cheap
9	14	Affordable
10	13	Affordable
11	2	Cheap
12	16	Expensive
13	7	Cheap
14	12	Cheap
15	1	Cheap
16	8	Affordable



```sql
--10) Find customers who have ordered the product with the highest price.

SELECT
     cu.customer_id
    ,cu.first_name
    ,cu.last_name
    ,pr.product_id
    ,pr.price
FROM
    order_items oi
    JOIN products pr ON oi.product_id = pr.product_id
    JOIN orders o ON oi.order_id = o.order_id
    JOIN customers cu ON o.customer_id = cu.customer_id
WHERE
    pr.price = (
        SELECT MAX(price)
        FROM products
    );

```



	customer_id integer	first_name character varying (100)	last_name character varying (100)	product_id integer	price numeric
1	8	Ivy	Jones	13	70.00
2	13	Sophia	Thomas	13	70.00

THANK YOU!

GLADYS CORTES