ABRILIAN MAULIDHIA

STOCKBIT

TECHNICAL TEST

BUSINESS
INTELLIGENCE
INTERN

Hello,
I’m Brili!

I’m a learner and enthusiast of data with a professional certification as a TensorFlow Developer. Interested in data and new things in technology and try to grow with any experiences.

https://www.linkedin.com/in/abrilianmaulidhia/

Bachelor of Computer Science

International Student
Exchange 2023 Awardee of Universiti
Tun Hussein Onn Malaysia (UTHM)

CONTACT

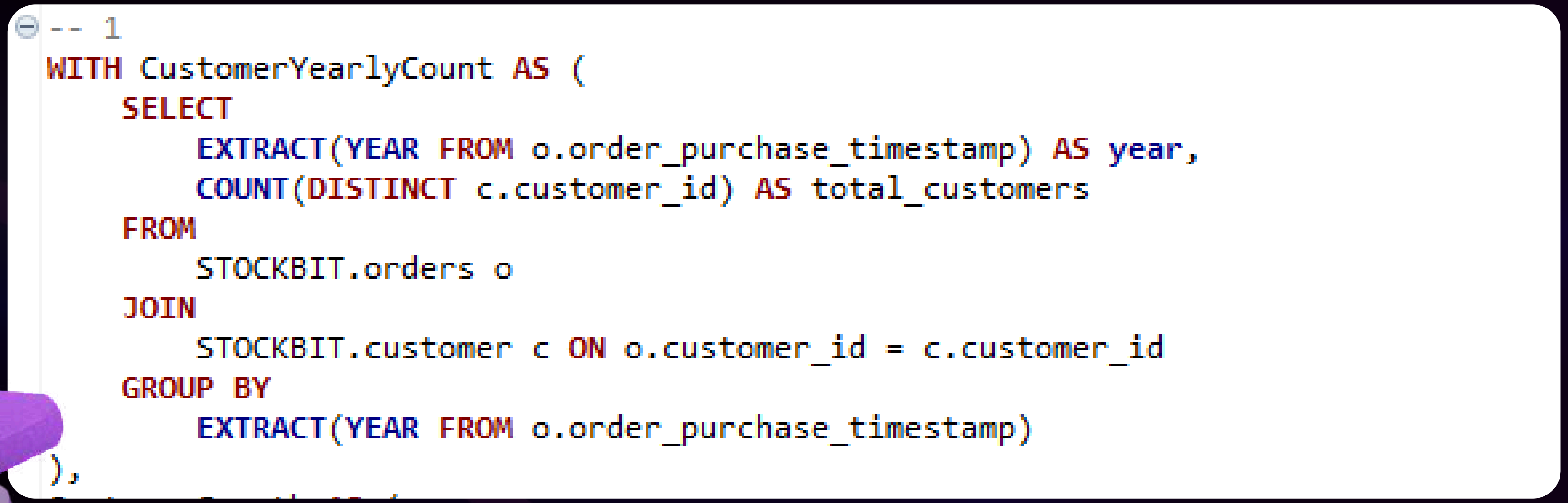
📍 Malang, Indonesia

✉ brilianmaulidhia@gmail.com

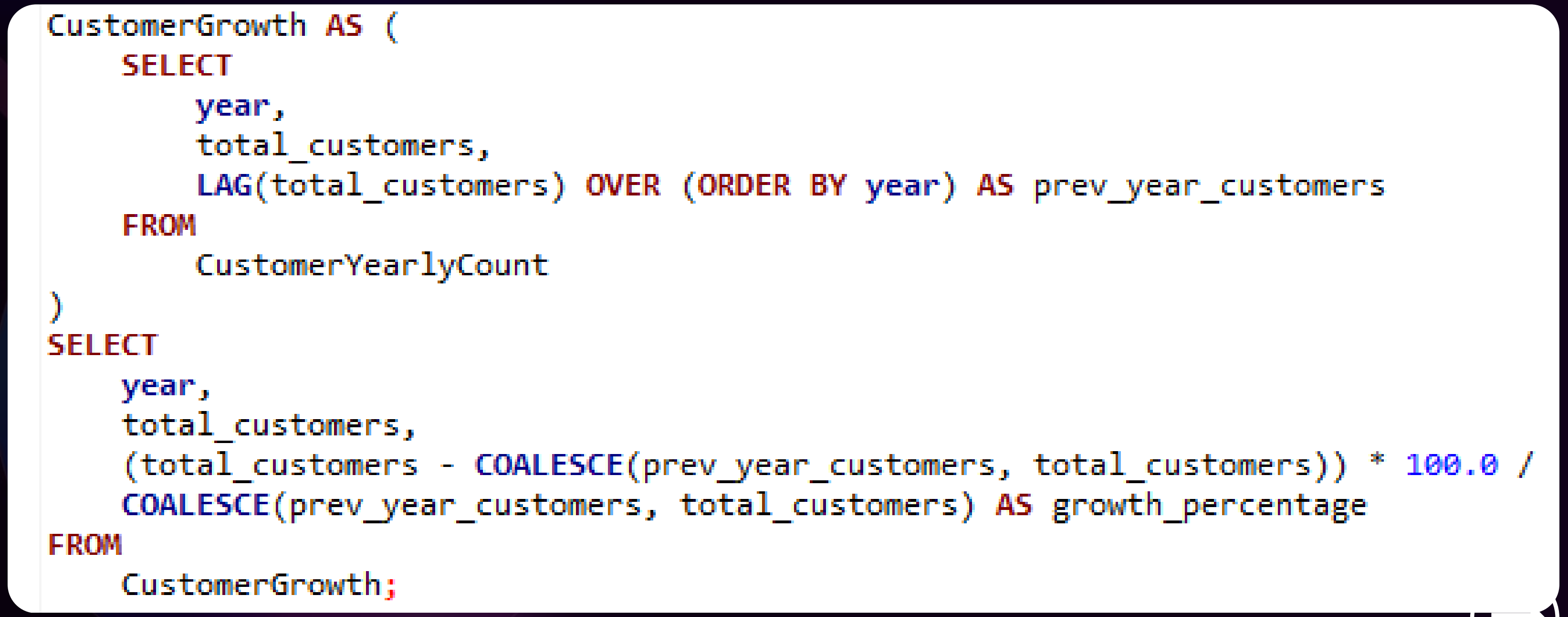
☎ (+62)82233980235

#1

Calculate the total users/customers that are listed each year and show the growth percentage from the beginning to the current year.

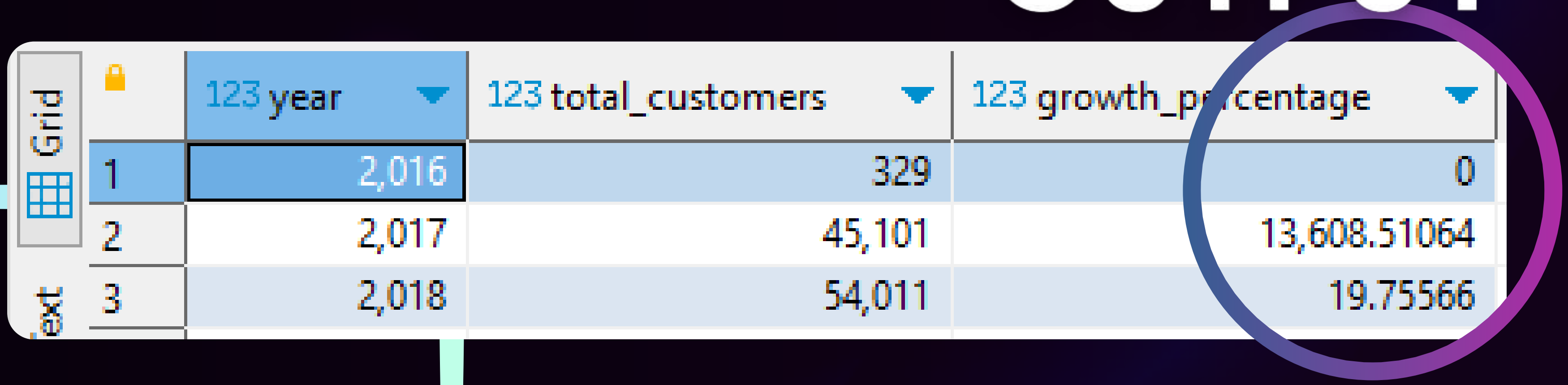


```
-- 1
WITH CustomerYearlyCount AS (
    SELECT
        EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
        COUNT(DISTINCT c.customer_id) AS total_customers
    FROM
        STOCKBIT.orders o
    JOIN
        STOCKBIT.customer c ON o.customer_id = c.customer_id
    GROUP BY
        EXTRACT(YEAR FROM o.order_purchase_timestamp)
),
```



```
CustomerGrowth AS (
    SELECT
        year,
        total_customers,
        LAG(total_customers) OVER (ORDER BY year) AS prev_year_customers
    FROM
        CustomerYearlyCount
)
SELECT
    year,
    total_customers,
    (total_customers - COALESCE(prev_year_customers, total_customers)) * 100.0 /
    COALESCE(prev_year_customers, total_customers) AS growth_percentage
FROM
    CustomerGrowth;
```

OUTPUT



	123 year	123 total_customers	123 growth_percentage
1	2,016	329	0
2	2,017	45,101	13,608.51064
3	2,018	54,011	19.75566

CONCLUSION

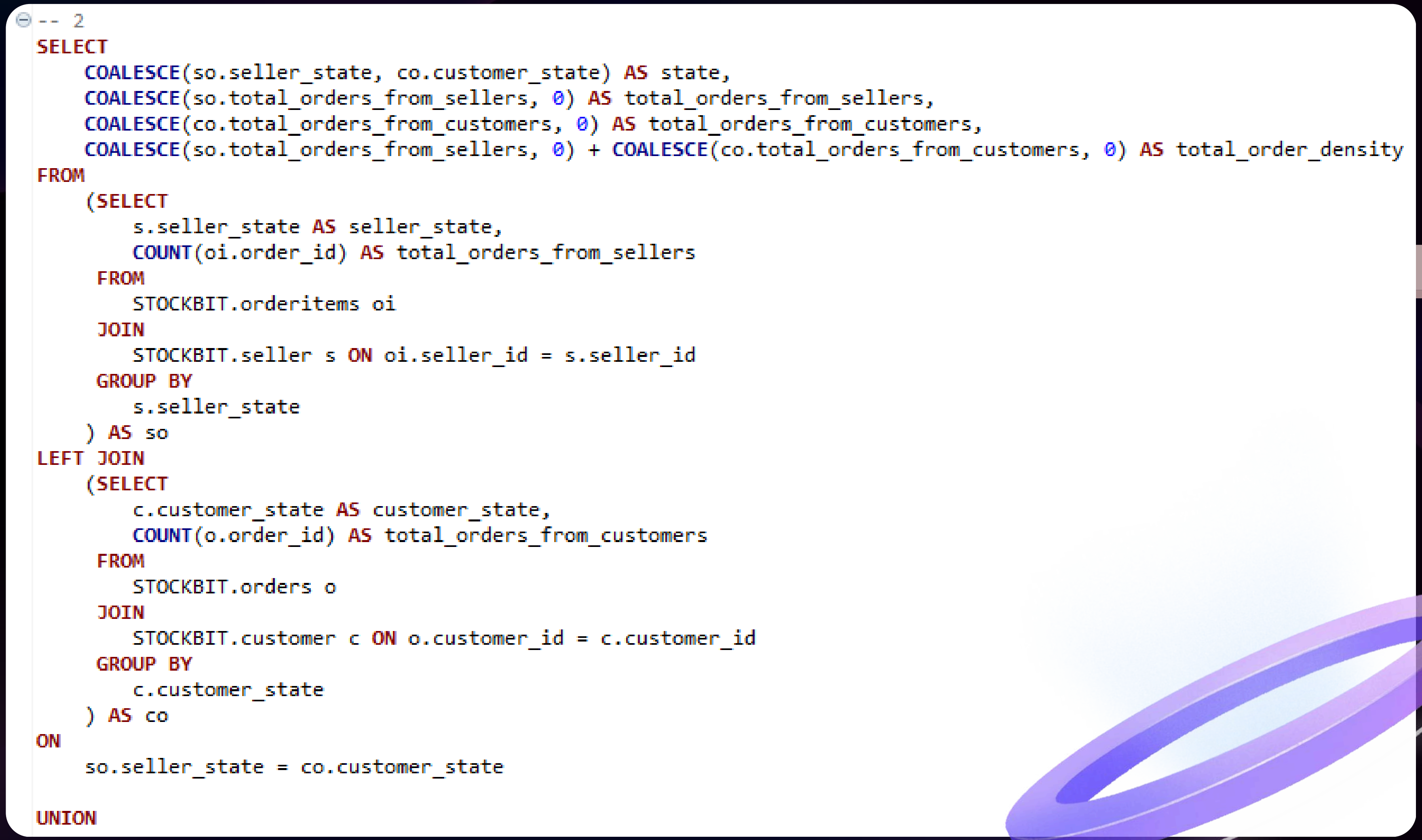
Key Observations

1. 2016
   - There were 329 distinct customers in 2016.
   - Since this is the first year in the dataset, there is no growth percentage to compare against, which is why the growth percentage is 0.00%.
2. 2017
   - The number of distinct customers dramatically increased to 45,101 in 2017.
   - This represents an extraordinary growth of 13,608.51% compared to 2016, indicating a significant expansion in the customer base, possibly due to the company's growth, marketing success, or entry into new markets.
3. 2018
   - In 2018, the number of distinct customers further increased to 54,011.
   - However, the growth rate slowed down considerably compared to the previous year, with a more moderate growth of 19.76%. This suggests that after the rapid expansion in 2017, the customer growth rate stabilized in 2018.

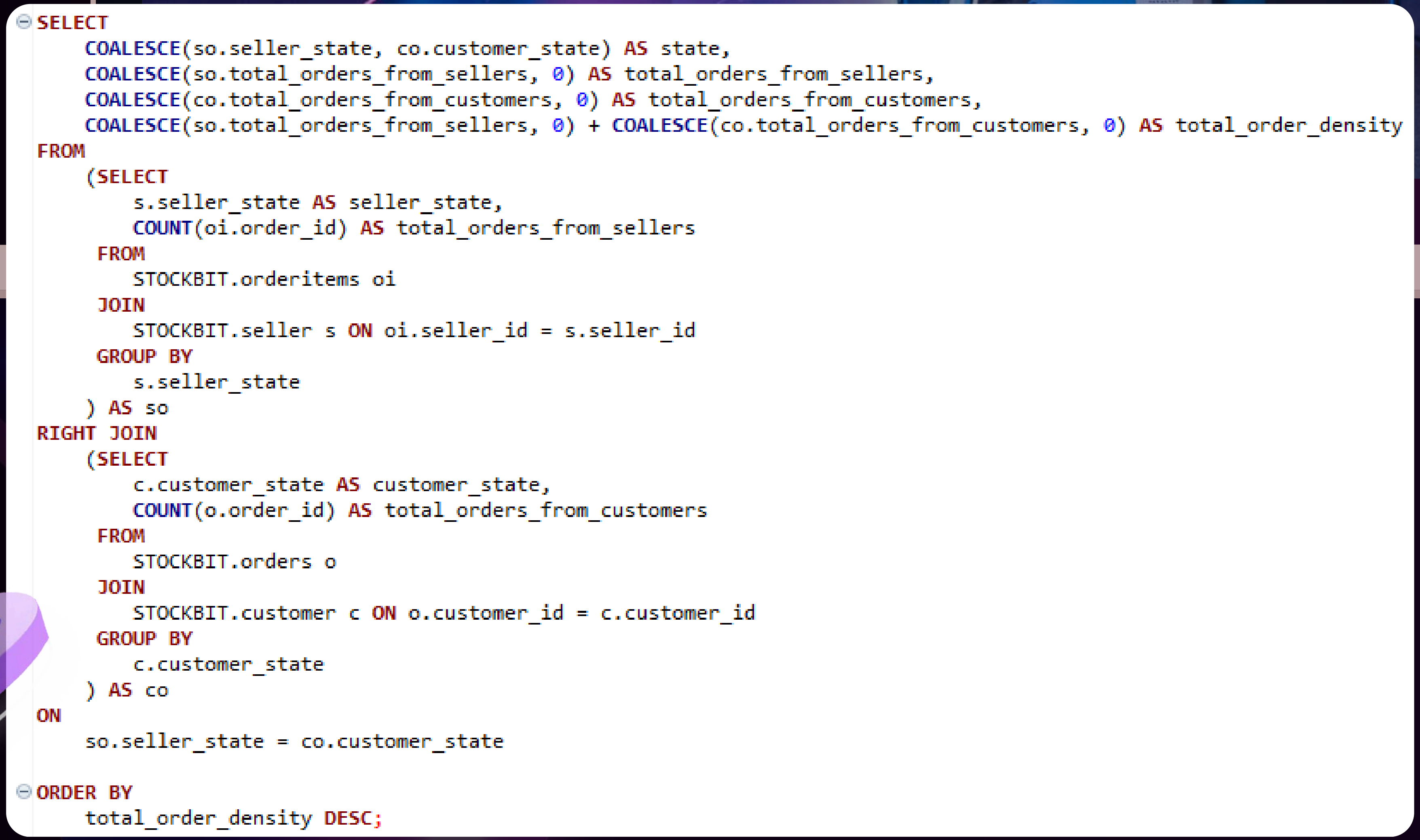
The data indicates explosive growth in the customer base from 2016 to 2017, with a more typical growth rate in 2018. This could be a sign of early-stage scaling (2017), followed by a phase of consolidation or maturity in the business (2018).

#2

Determine in which state should the company locate their warehouse; it should be in an area where the order density is the highest. (density ≠ amount)

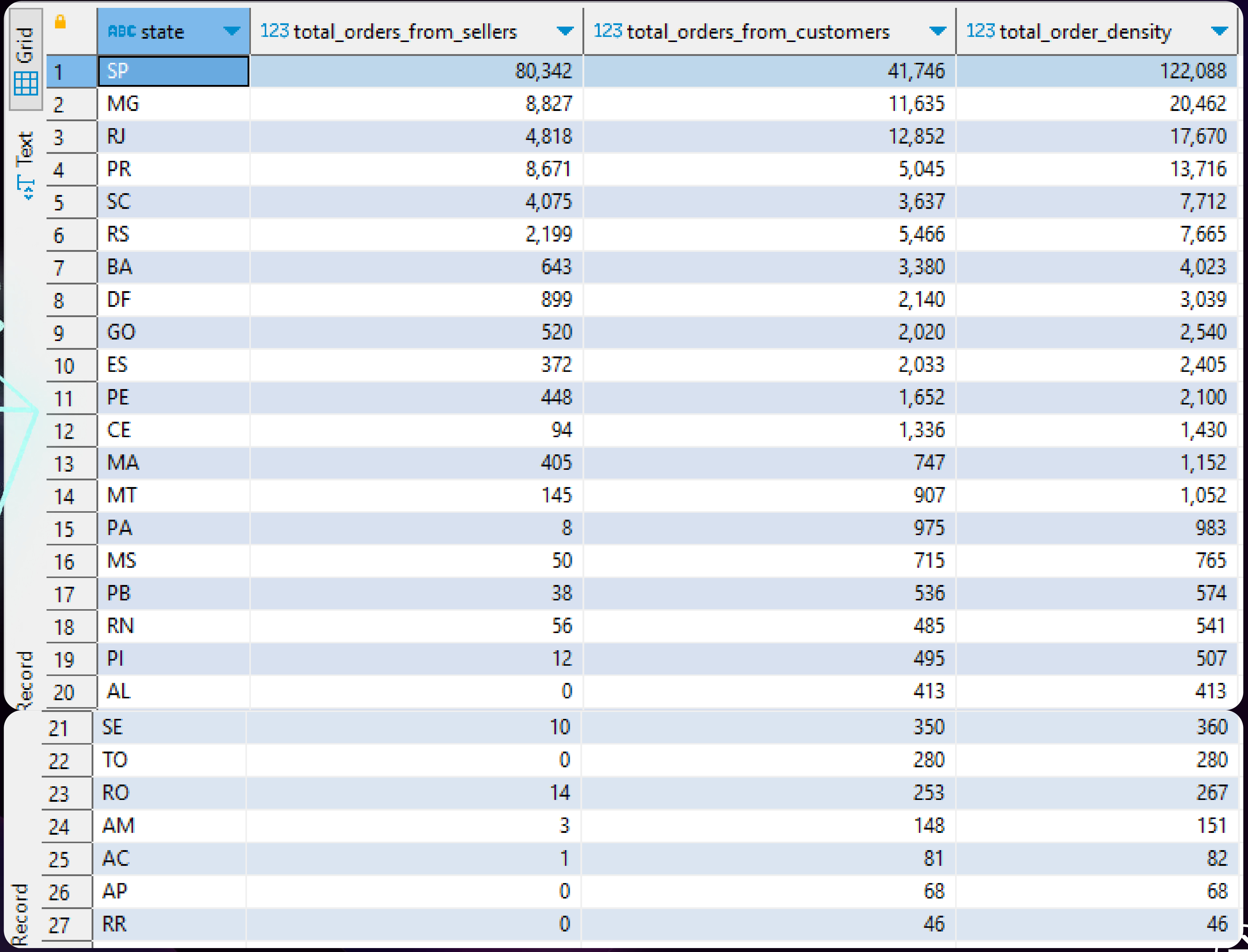


```
-- 2
SELECT
    COALESCE(so.seller_state, co.customer_state) AS state,
    COALESCE(so.total_orders_from_sellers, 0) AS total_orders_from_sellers,
    COALESCE(co.total_orders_from_customers, 0) AS total_orders_from_customers,
    COALESCE(so.total_orders_from_sellers, 0) + COALESCE(co.total_orders_from_customers, 0) AS total_order_density
FROM
    (SELECT
        s.seller_state AS seller_state,
        COUNT(oi.order_id) AS total_orders_from_sellers
     FROM
        STOCKBIT.orderitems oi
     JOIN
        STOCKBIT.seller s ON oi.seller_id = s.seller_id
     GROUP BY
        s.seller_state
    ) AS so
LEFT JOIN
    (SELECT
        c.customer_state AS customer_state,
        COUNT(o.order_id) AS total_orders_from_customers
     FROM
        STOCKBIT.orders o
     JOIN
        STOCKBIT.customer c ON o.customer_id = c.customer_id
     GROUP BY
        c.customer_state
    ) AS co
ON
    so.seller_state = co.customer_state

UNION
```

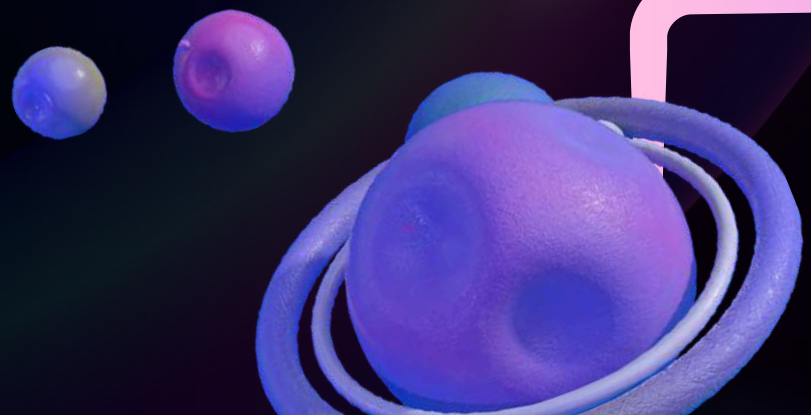


```
SELECT
    COALESCE(so.seller_state, co.customer_state) AS state,
    COALESCE(so.total_orders_from_sellers, 0) AS total_orders_from_sellers,
    COALESCE(co.total_orders_from_customers, 0) AS total_orders_from_customers,
    COALESCE(so.total_orders_from_sellers, 0) + COALESCE(co.total_orders_from_customers, 0) AS total_order_density
FROM
    (SELECT
        s.seller_state AS seller_state,
        COUNT(oi.order_id) AS total_orders_from_sellers
     FROM
        STOCKBIT.orderitems oi
     JOIN
        STOCKBIT.seller s ON oi.seller_id = s.seller_id
     GROUP BY
        s.seller_state
    ) AS so
RIGHT JOIN
    (SELECT
        c.customer_state AS customer_state,
        COUNT(o.order_id) AS total_orders_from_customers
     FROM
        STOCKBIT.orders o
     JOIN
        STOCKBIT.customer c ON o.customer_id = c.customer_id
     GROUP BY
        c.customer_state
    ) AS co
ON
    so.seller_state = co.customer_state

ORDER BY
    total_order_density DESC;
```

OUTPUT



	state	total_orders_from_sellers	total_orders_from_customers	total_order_density
1	SP	80,342	41,746	122,088
2	MG	8,827	11,635	20,462
3	RJ	4,818	12,852	17,670
4	PR	8,671	5,045	13,716
5	SC	4,075	3,637	7,712
6	RS	2,199	5,466	7,665
7	BA	643	3,380	4,023
8	DF	899	2,140	3,039
9	GO	520	2,020	2,540
10	ES	372	2,033	2,405
11	PE	448	1,652	2,100
12	CE	94	1,336	1,430
13	MA	405	747	1,152
14	MT	145	907	1,052
15	PA	8	975	983
16	MS	50	715	765
17	PB	38	536	574
18	RN	56	485	541
19	PI	12	495	507
20	AL	0	413	413
21	SE	10	350	360
22	TO	0	280	280
23	RO	14	253	267
24	AM	3	148	151
25	AC	1	81	82
26	AP	0	68	68
27	RR	0	46	46



CONCLUSION

Key Observations

1. Based on the SQL output for states with sales greater than 5,000, the following states have significant total orders

- SP (São Paulo): 80,342 total orders, 41,746 from sellers, and 122,088 combined orders.
- MG (Minas Gerais): 8,827 total orders, 11,635 from sellers, and 20,462 combined orders.
- RJ (Rio de Janeiro): 4,818 total orders, 12,852 from sellers, and 17,670 combined orders.
- PR (Paraná): 8,671 total orders, 5,045 from sellers, and 13,716 combined orders.

Given the states with the highest order densities, São Paulo (SP) emerges as the optimal location for the warehouse due to the highest combined order density and central positioning based on geolocation.

2. Using the geolocation data

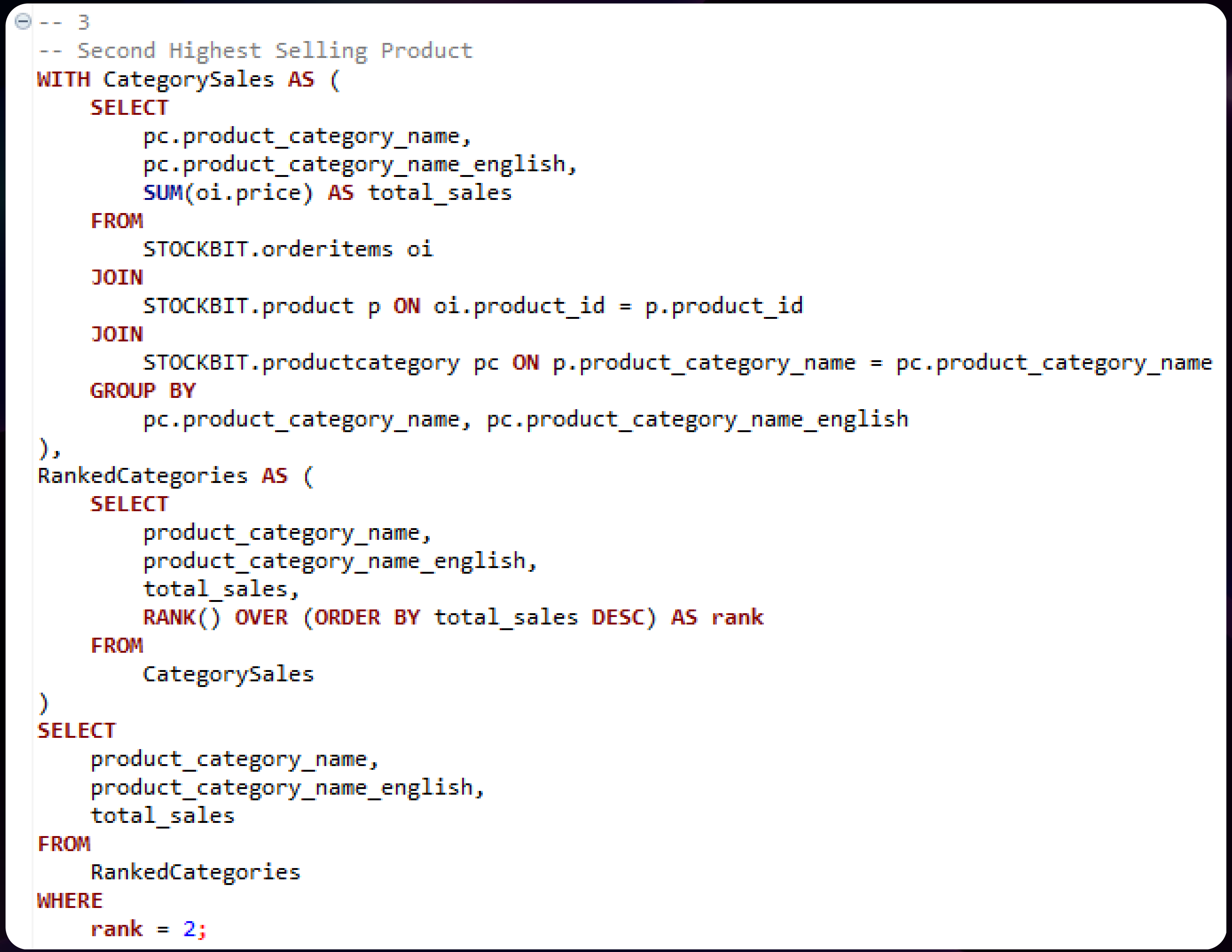
- SP (São Paulo) has the highest geolocation longitude at -41.
- PR (Paraná) has the lowest geolocation longitude at -53.

The midpoint between these two extremes is -47, which falls within the São Paulo (SP) region. This central location, combined with high order density and geographical advantage, makes SP the most strategic warehouse location.

#3

What is the second highest selling product and the second lowest selling products category in English and Portuguese?

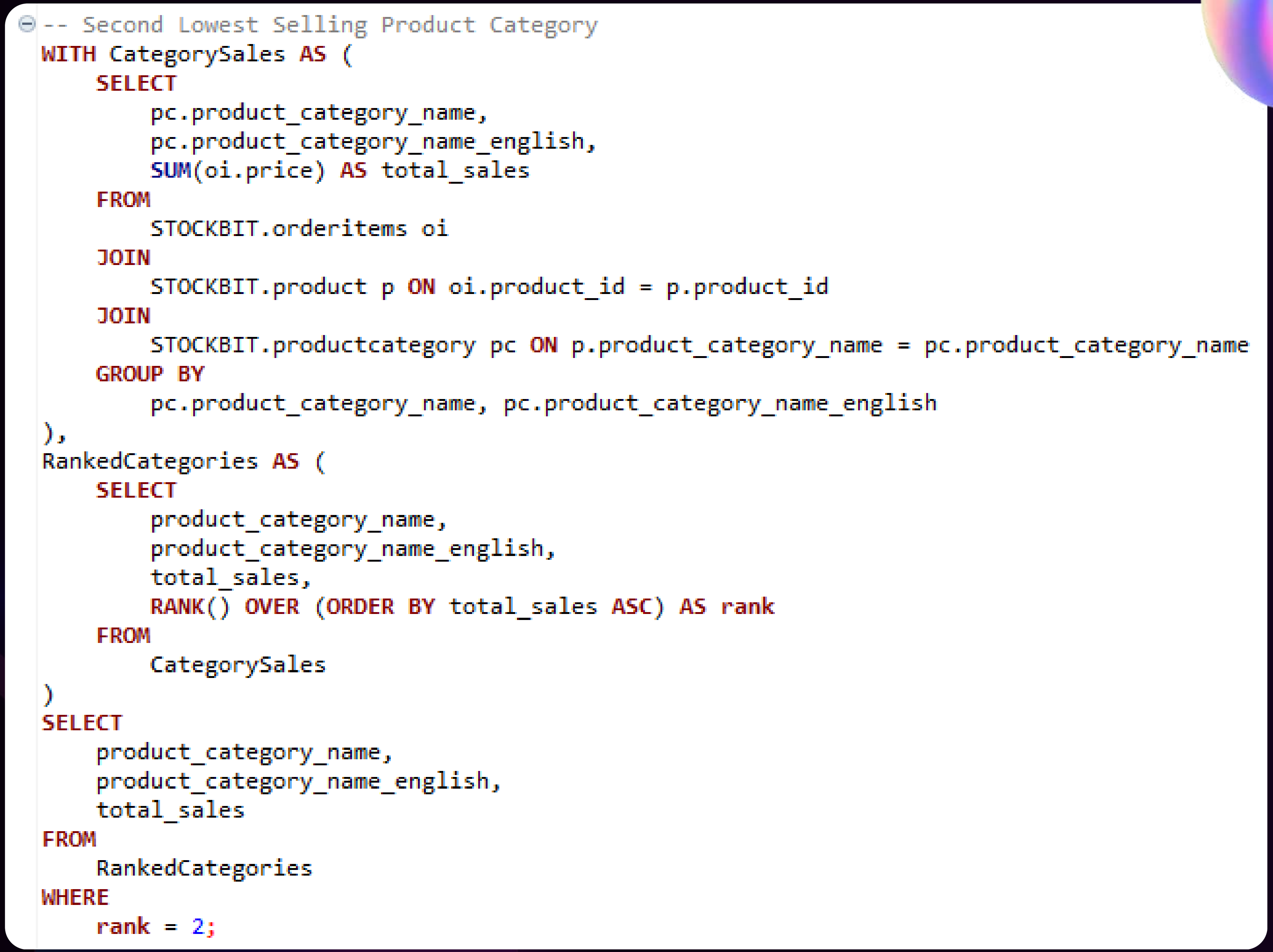
ABRILIAN
MAULIDHIA



```sql
-- 3
-- Second Highest Selling Product
WITH CategorySales AS (
    SELECT
        pc.product_category_name,
        pc.product_category_name_english,
        SUM(oi.price) AS total_sales
    FROM
        STOCKBIT.orderitems oi
    JOIN
        STOCKBIT.product p ON oi.product_id = p.product_id
    JOIN
        STOCKBIT.productcategory pc ON p.product_category_name = pc.product_category_name
    GROUP BY
        pc.product_category_name, pc.product_category_name_english
),
RankedCategories AS (
    SELECT
        product_category_name,
        product_category_name_english,
        total_sales,
        RANK() OVER (ORDER BY total_sales DESC) AS rank
    FROM
        CategorySales
)
SELECT
    product_category_name,
    product_category_name_english,
    total_sales
FROM
    RankedCategories
WHERE
    rank = 2;
```

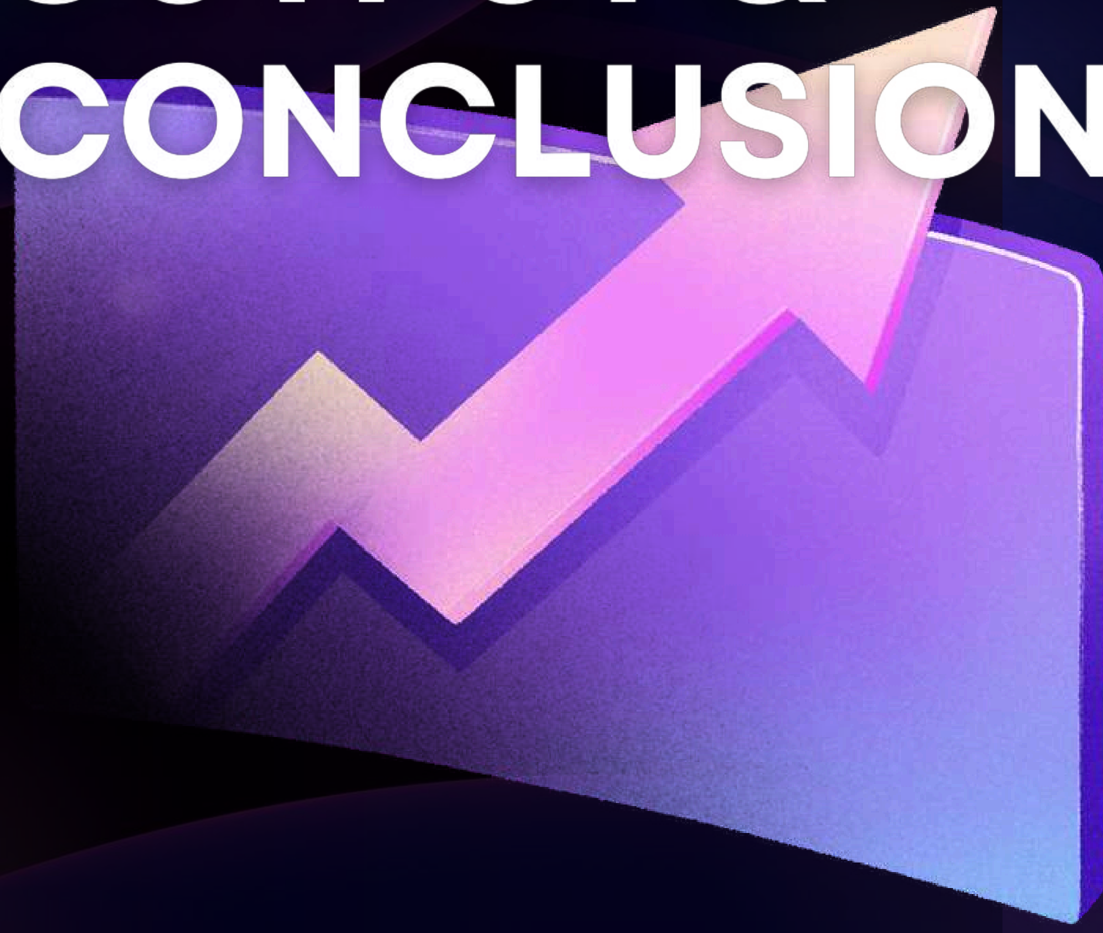
SECOND HIGHEST

SECOND LOWEST

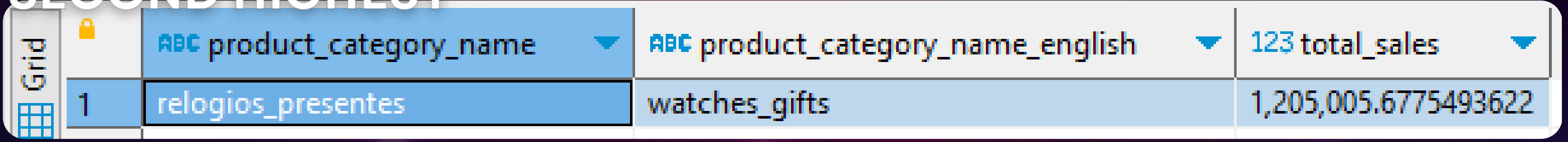


```sql
-- Second Lowest Selling Product Category
WITH CategorySales AS (
    SELECT
        pc.product_category_name,
        pc.product_category_name_english,
        SUM(oi.price) AS total_sales
    FROM
        STOCKBIT.orderitems oi
    JOIN
        STOCKBIT.product p ON oi.product_id = p.product_id
    JOIN
        STOCKBIT.productcategory pc ON p.product_category_name = pc.product_category_name
    GROUP BY
        pc.product_category_name, pc.product_category_name_english
),
RankedCategories AS (
    SELECT
        product_category_name,
        product_category_name_english,
        total_sales,
        RANK() OVER (ORDER BY total_sales ASC) AS rank
    FROM
        CategorySales
)
SELECT
    product_category_name,
    product_category_name_english,
    total_sales
FROM
    RankedCategories
WHERE
    rank = 2;
```

OUTPUT & CONCLUSION



SECOND HIGHEST



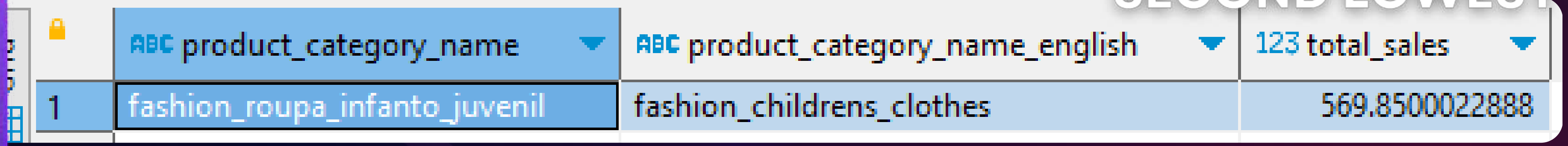
	product_category_name	product_category_name_english	total_sales
1	relogios_presentes	watches_gifts	1,205,005.6775493622

Second Highest Selling Product Category

1. Category Name (Portuguese) : Relógios e Presentes
2. Category Name (English) : Watches & Gifts
3. Total Sales : 1,205,005.68

Watches & Gifts (Relógios e Presentes) had significantly higher sales compared to other categories.

SECOND LOWEST



	product_category_name	product_category_name_english	total_sales
1	fashion_roupa_infanto_juvenil	fashion_childrens_clothes	569.8500022888

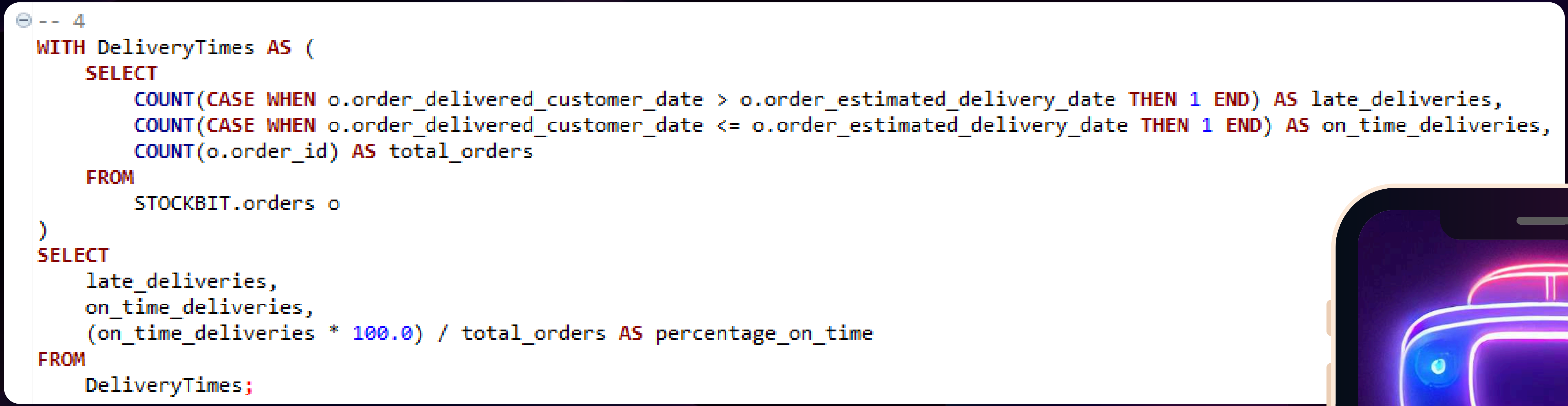
Lowest Selling Product Category

1. Category Name (Portuguese): Fashion Roupa Infanto Juvenil
2. Category Name (English): Fashion Children's Clothes
3. Total Sales: 569.85

Fashion Children's Clothes (Fashion Roupa Infanto Juvenil) had the lowest sales among all categories.

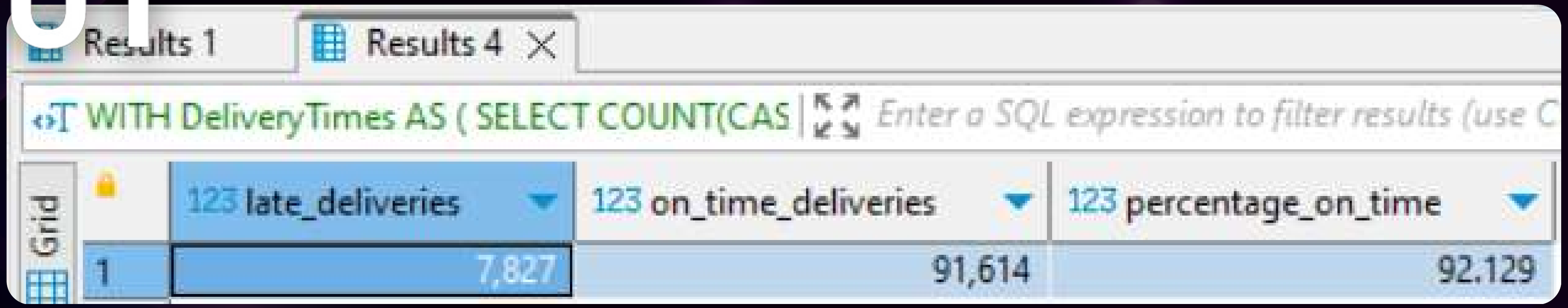
#4

Count how many products arrived late (Exceeding the promised delivery date) and the percentage of orders that are delivered on schedule or faster.



```
-- 4
WITH DeliveryTimes AS (
    SELECT
        COUNT(CASE WHEN o.order_delivered_customer_date > o.order_estimated_delivery_date THEN 1 END) AS late_deliveries,
        COUNT(CASE WHEN o.order_delivered_customer_date <= o.order_estimated_delivery_date THEN 1 END) AS on_time_deliveries,
        COUNT(o.order_id) AS total_orders
    FROM
        STOCKBIT.orders o
)
SELECT
    late_deliveries,
    on_time_deliveries,
    (on_time_deliveries * 100.0) / total_orders AS percentage_on_time
FROM
    DeliveryTimes;
```

OUTPUT



```
WITH DeliveryTimes AS ( SELECT COUNT(CAS
```

	late_deliveries	on_time_deliveries	percentage_on_time
1	7,827	91,614	92.129

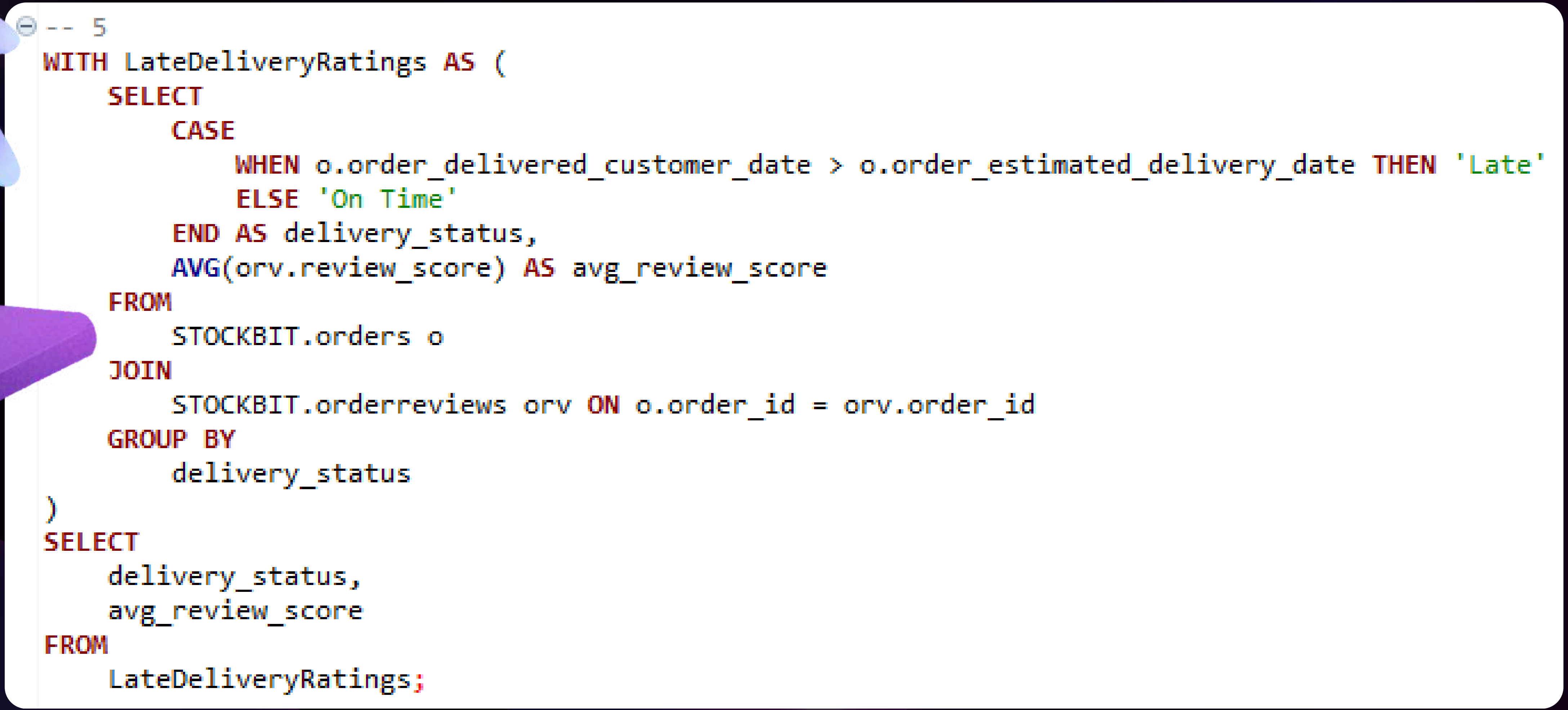
1. Late Deliveries: 7,827
2. On-Time Deliveries: 91,614
3. Percentage of On-Time Deliveries: 92.13%

CONCLUSION

Out of the total deliveries, 7,827 were late, meaning they exceeded the promised delivery date. The percentage of orders that were delivered on or before the estimated delivery date is approximately **92.13%**. This indicates that a high majority of orders were delivered as scheduled or faster.

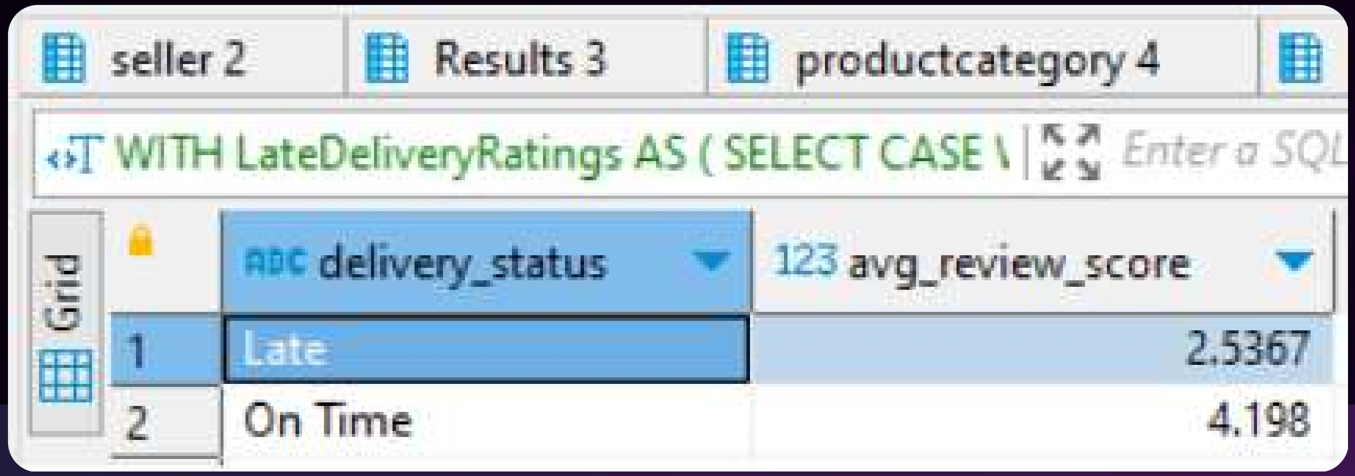
#5

Determine whether late delivery affects the ratings (Star) given on those products. You can also explain your result in short sentences if necessary.



```
-- 5
WITH LateDeliveryRatings AS (
    SELECT
        CASE
            WHEN o.order_delivered_customer_date > o.order_estimated_delivery_date THEN 'Late'
            ELSE 'On Time'
        END AS delivery_status,
        AVG(orv.review_score) AS avg_review_score
    FROM
        STOCKBIT.orders o
    JOIN
        STOCKBIT.orderreviews orv ON o.order_id = orv.order_id
    GROUP BY
        delivery_status
)
SELECT
    delivery_status,
    avg_review_score
FROM
    LateDeliveryRatings;
```

OUTPUT CONCLUSION



```
WITH LateDeliveryRatings AS ( SELECT CASE \
```

	delivery_status	avg_review_score
1	Late	2.5367
2	On Time	4.198

Yes, late delivery does affect the ratings given on those products.

- Late Deliveries: Orders that were delivered late have a significantly lower average review score of 2.54.
- On-Time Deliveries: Orders that were delivered on time have a higher average review score of 4.20.

This suggests that late deliveries are associated with lower customer satisfaction, as reflected in the lower review scores. On-time deliveries tend to result in higher satisfaction and better ratings from customers.

THANK YOU!

For More Details, Please Click Here!