RETAIL SALES ANALYSIS

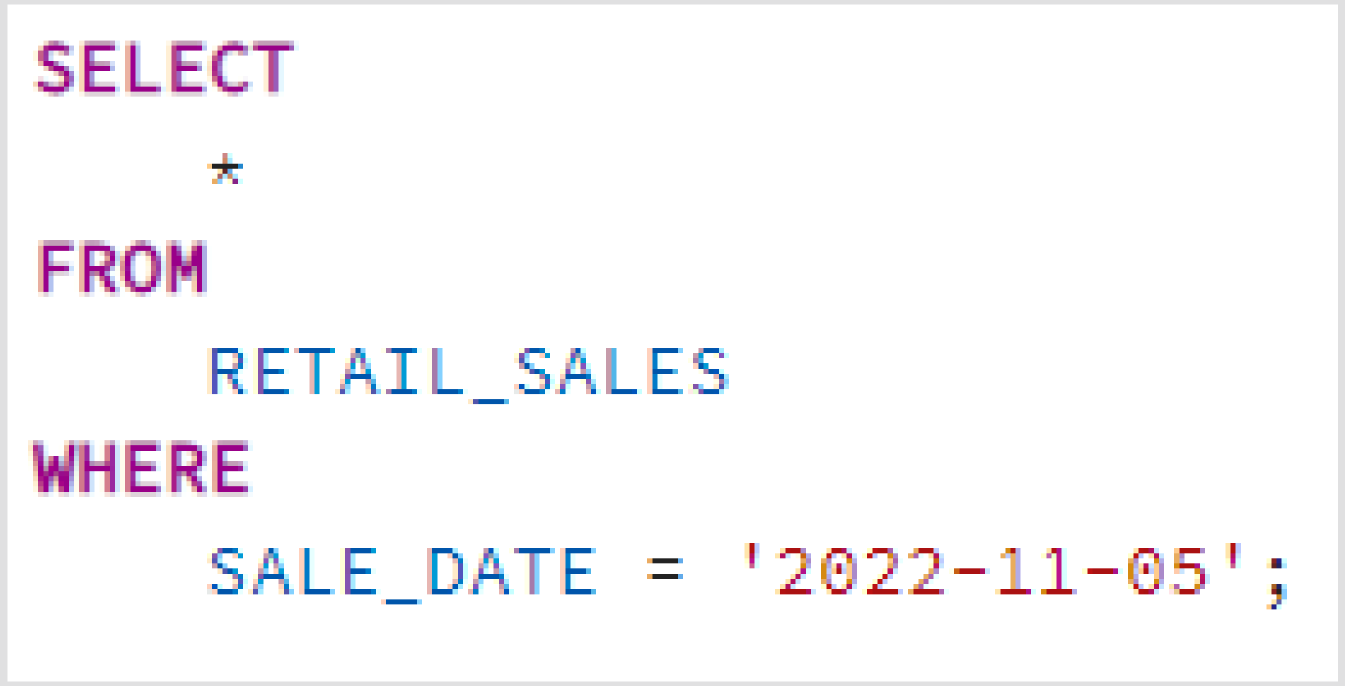
A SQL-BASED DATA ANALYSIS PROJECT

Hi, I'm Sabah Khan.
In this project, I've used SQL queries to explore and answer key questions about retail sales, demonstrating how data can provide valuable insights.

Database Structure:

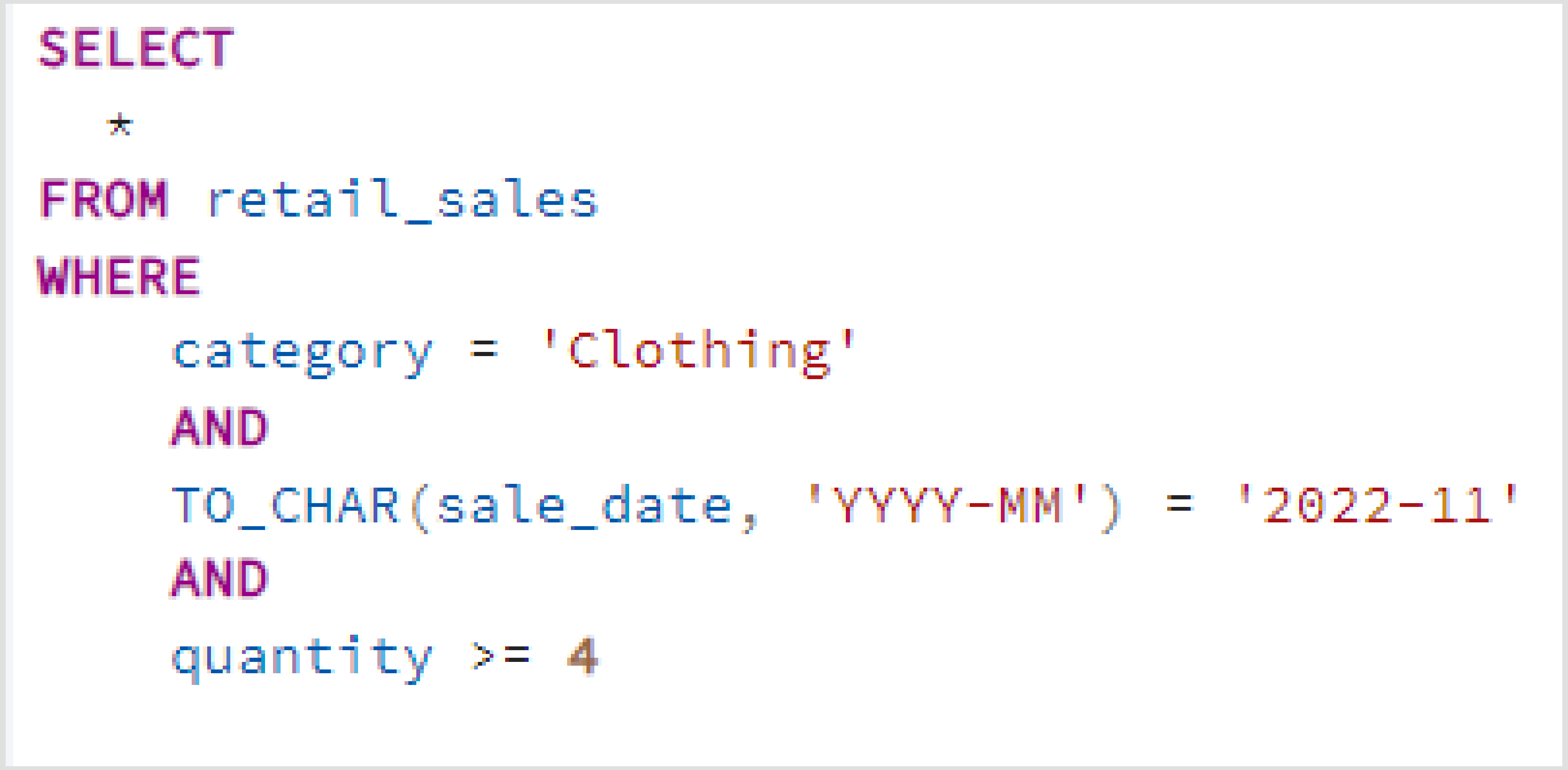
The retail_sales table stores transactional data, with each row representing a single sale. Key information includes transaction identifiers, timestamps, customer demographics, product details (category, quantity, price), COGS, and total sale value. This structure facilitates sales analysis and reporting.

1. **Write a SQL query to retrieve all columns for sales made on 2022-11-05:**



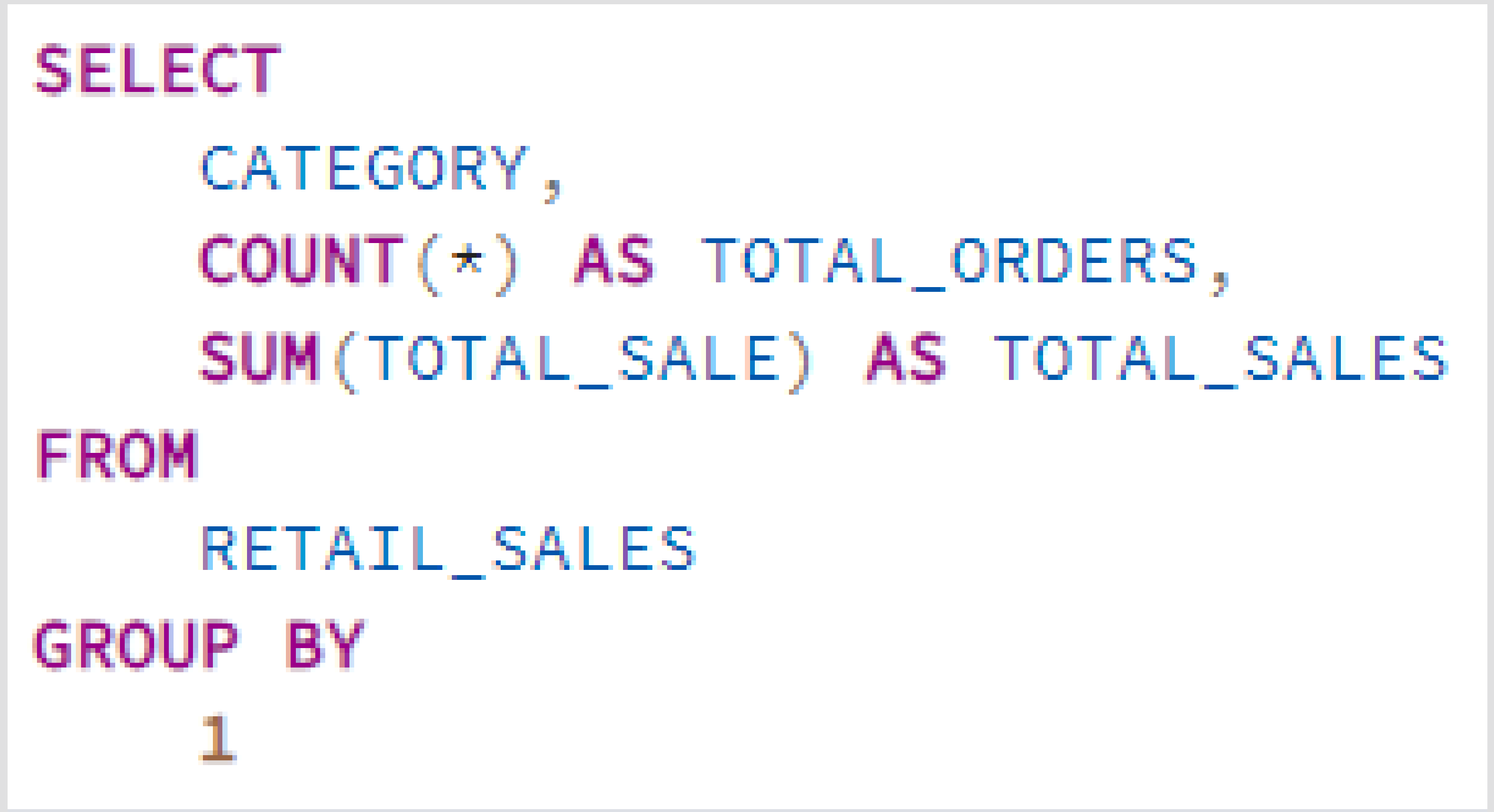
```
SELECT
    *
FROM
    RETAIL_SALES
WHERE
    SALE_DATE = '2022-11-05';
```

2. Write a SQL query to retrieve all transactions where the category is 'Clothing' and the quantity sold is more than 4 in the month of Nov-2022:

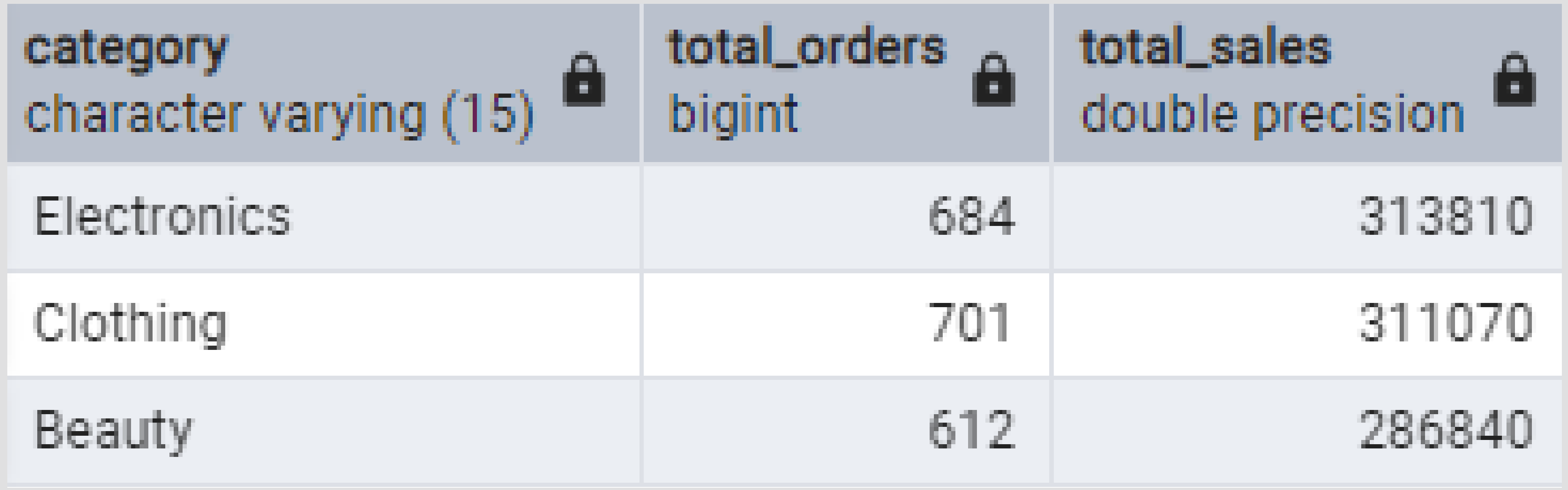


```sql
SELECT
  *
FROM retail_sales
WHERE
    category = 'Clothing'
    AND
    TO_CHAR(sale_date, 'YYYY-MM') = '2022-11'
    AND
    quantity >= 4
```

3.Write a SQL query to calculate the total sales and total orders for each category.

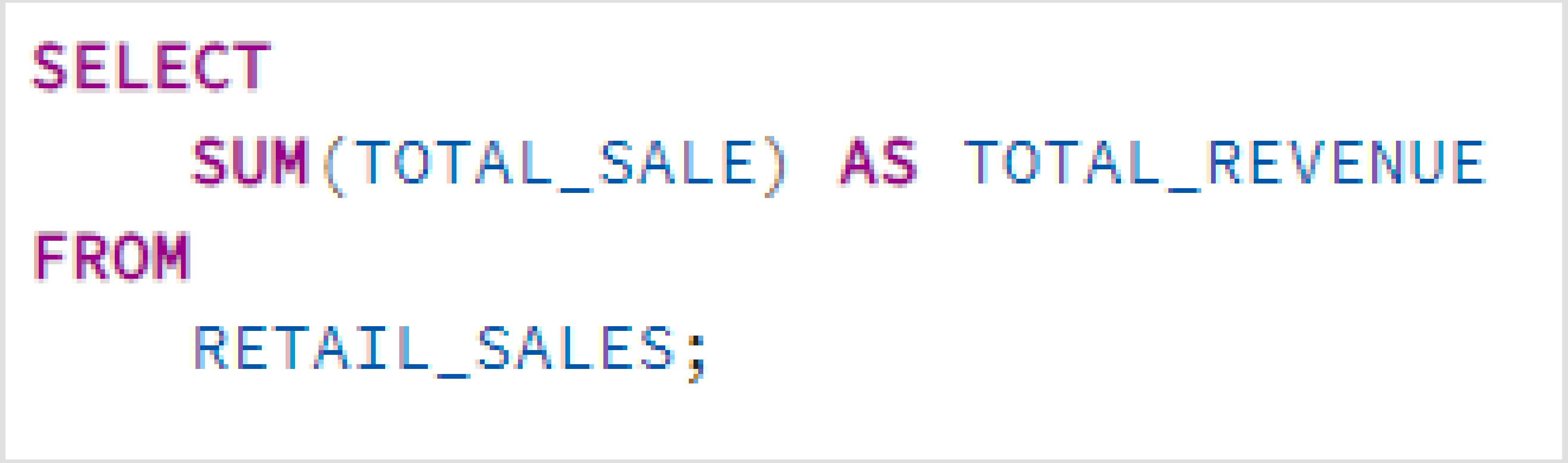


```
SELECT
    CATEGORY,
    COUNT(*) AS TOTAL_ORDERS,
    SUM(TOTAL_SALE) AS TOTAL_SALES
FROM
    RETAIL_SALES
GROUP BY
    1
```

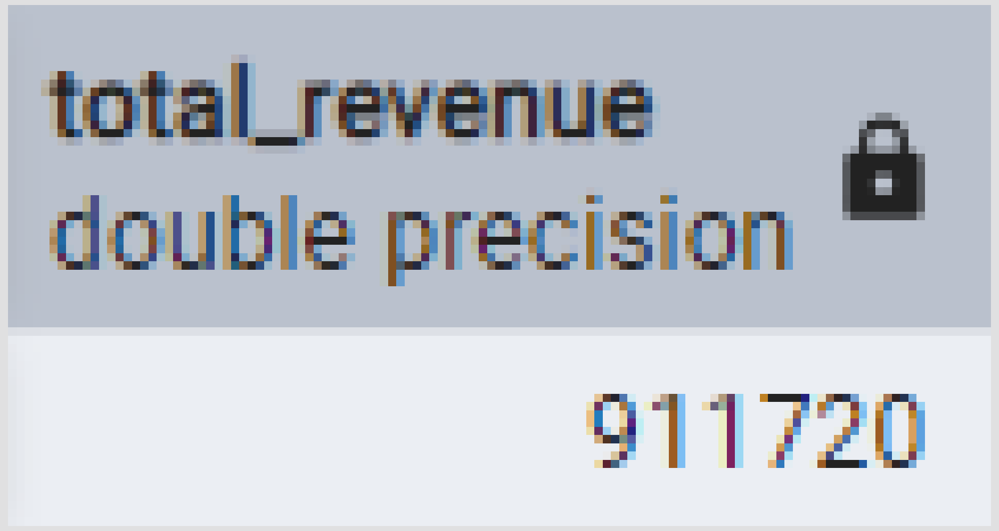


category character varying (15)	total_orders bigint	total_sales double precision
Electronics	684	313810
Clothing	701	311070
Beauty	612	286840

4. Write a query to calculate the total revenue from the dataset.

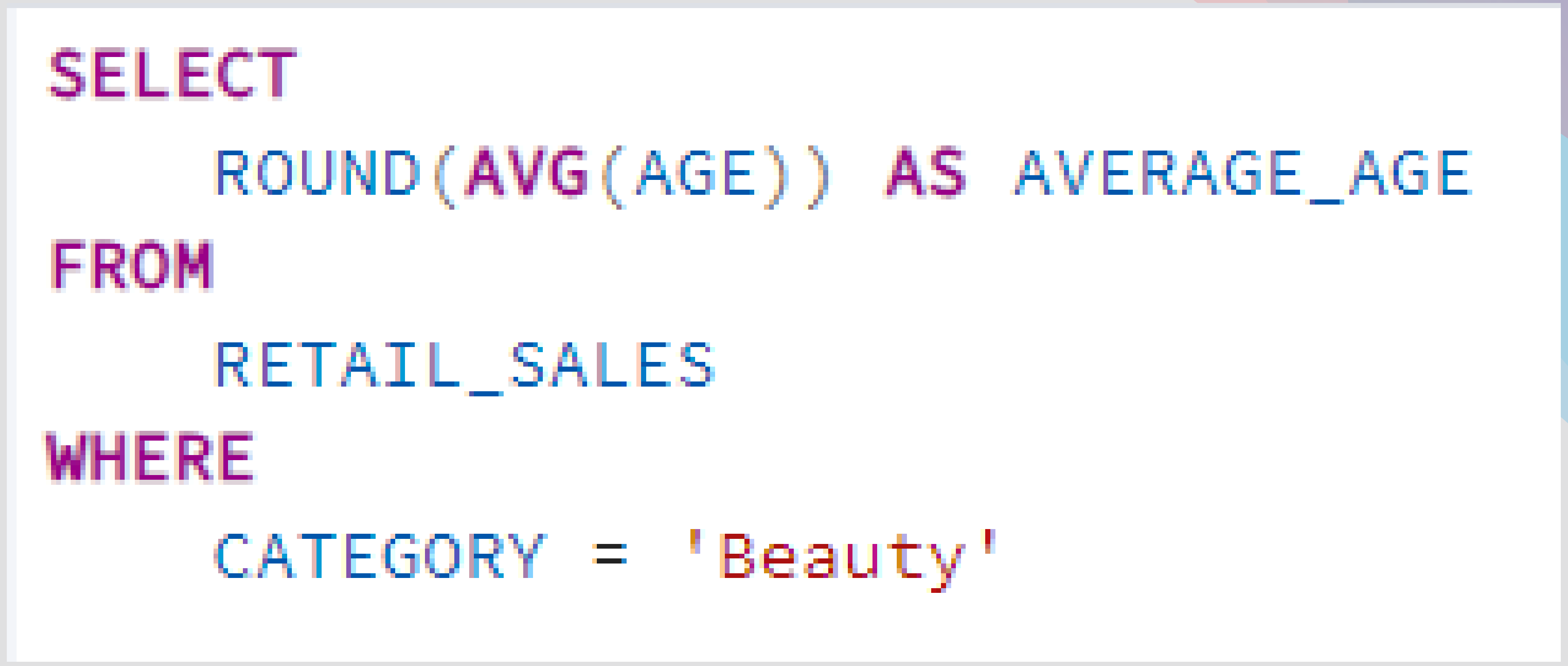


```sql
SELECT
    SUM(TOTAL_SALE) AS TOTAL_REVENUE
FROM
    RETAIL_SALES;
```

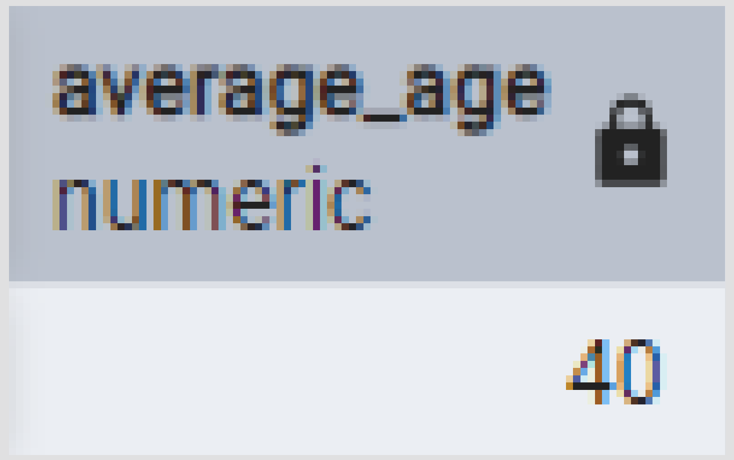


```
total_revenue
double precision
          911720
```

5. Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.

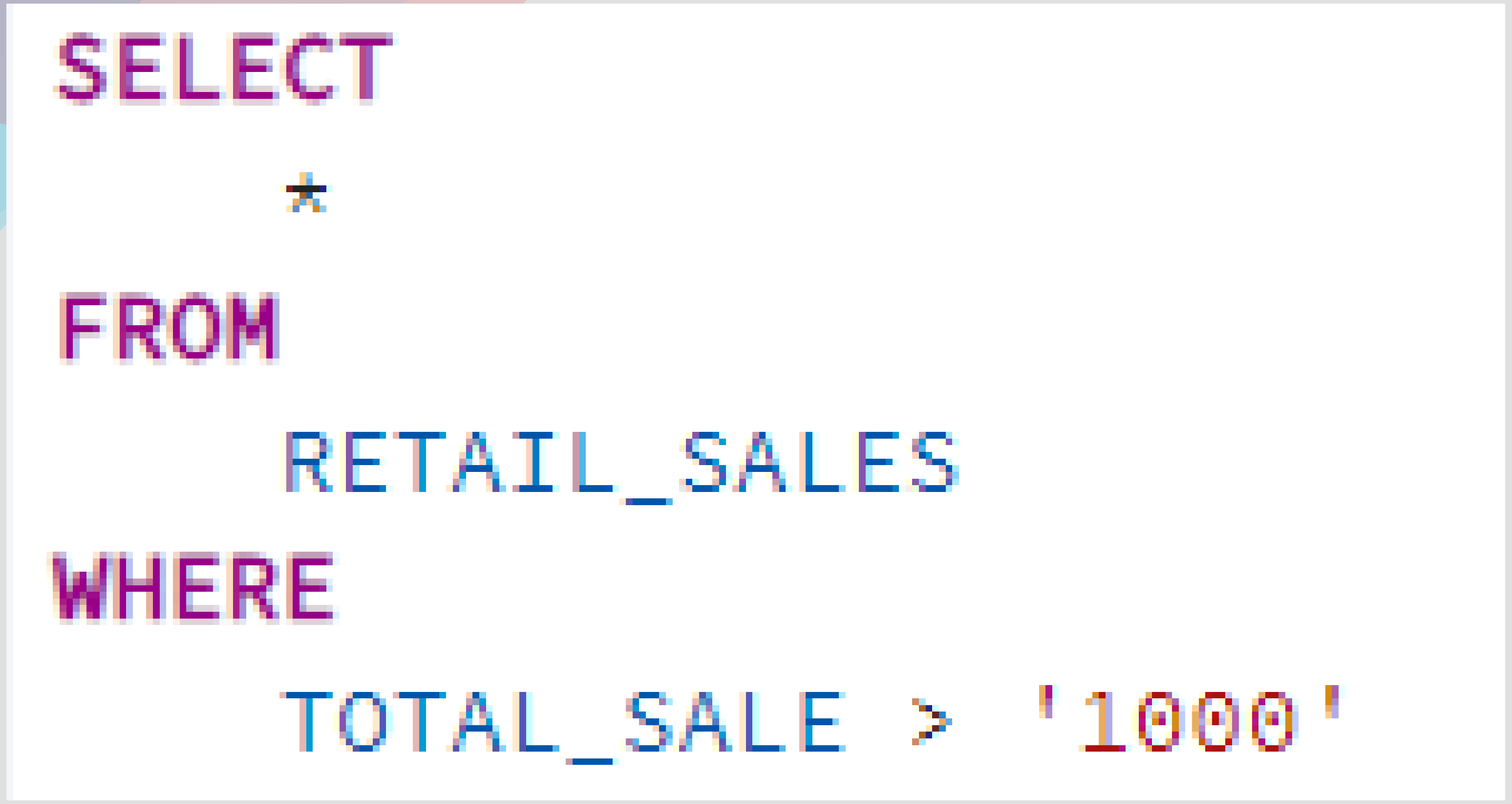


```
SELECT
    ROUND(AVG(AGE)) AS AVERAGE_AGE
FROM
    RETAIL_SALES
WHERE
    CATEGORY = 'Beauty'
```



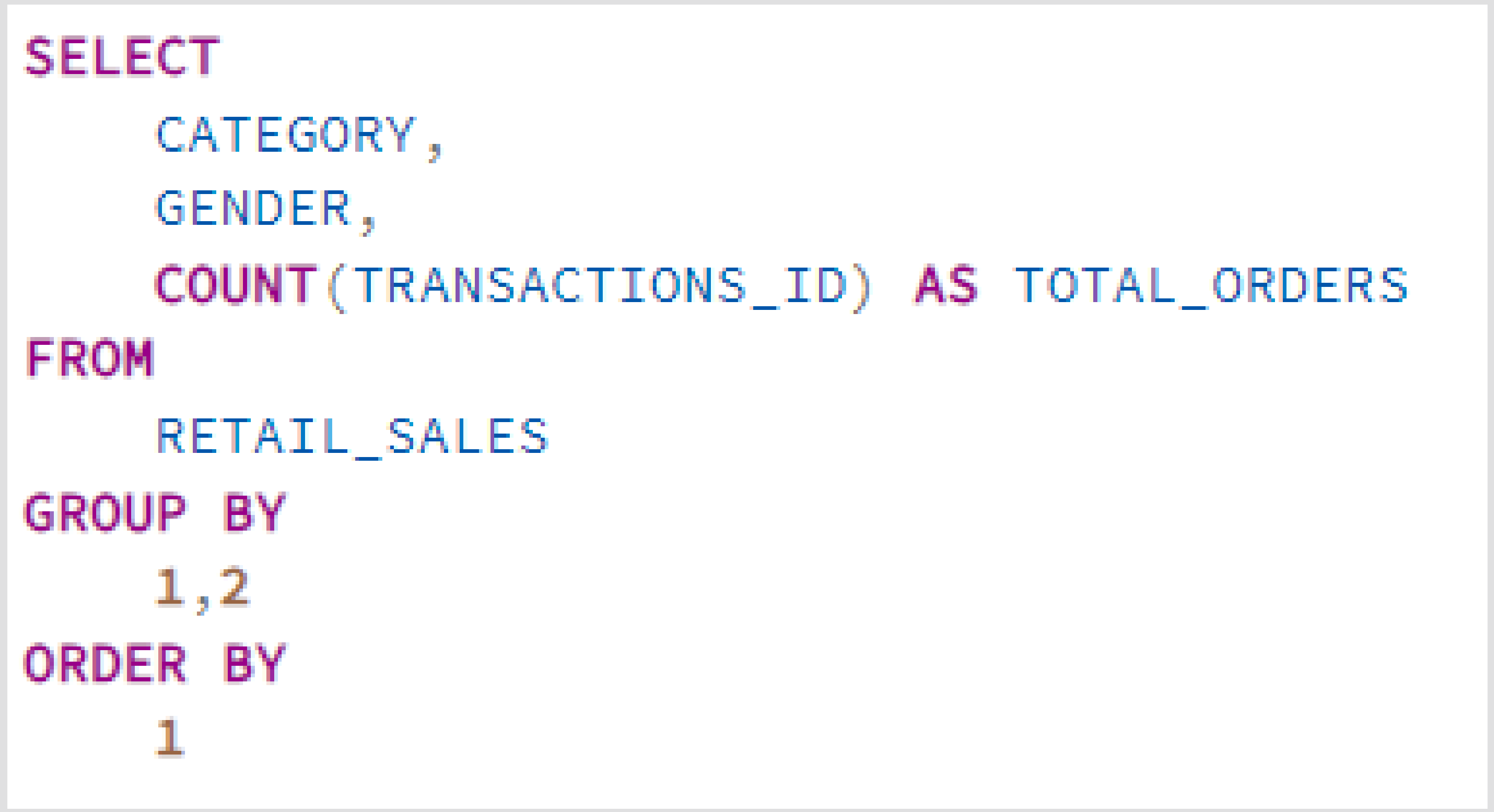
```
average_age
numeric
          40
```

6. Write a SQL query to find all transactions where the total_sale is greater than 1000.

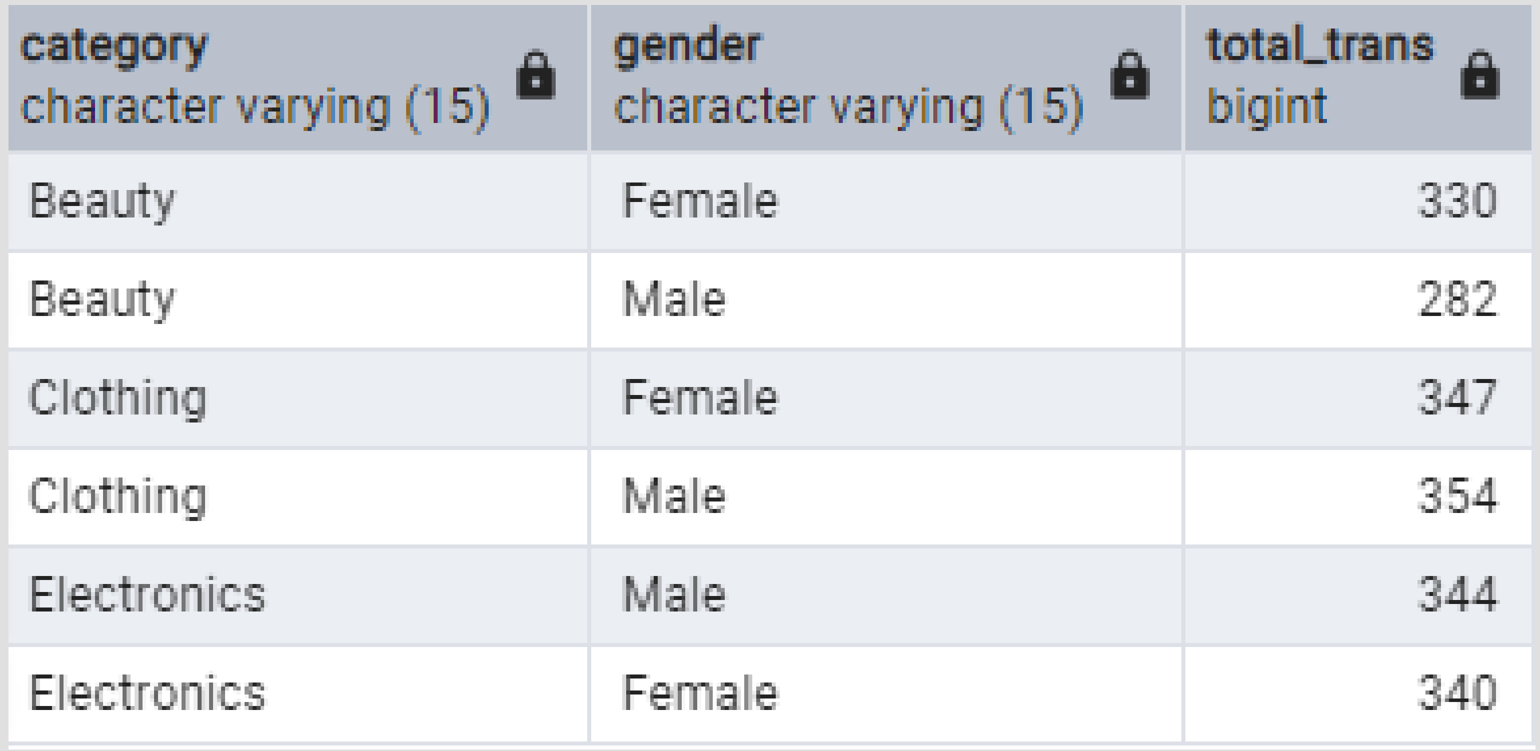


```
SELECT
    *
FROM
    RETAIL_SALES
WHERE
    TOTAL_SALE > '1000'
```

7. Write a SQL query to find the total number of transactions made by each gender in each category.

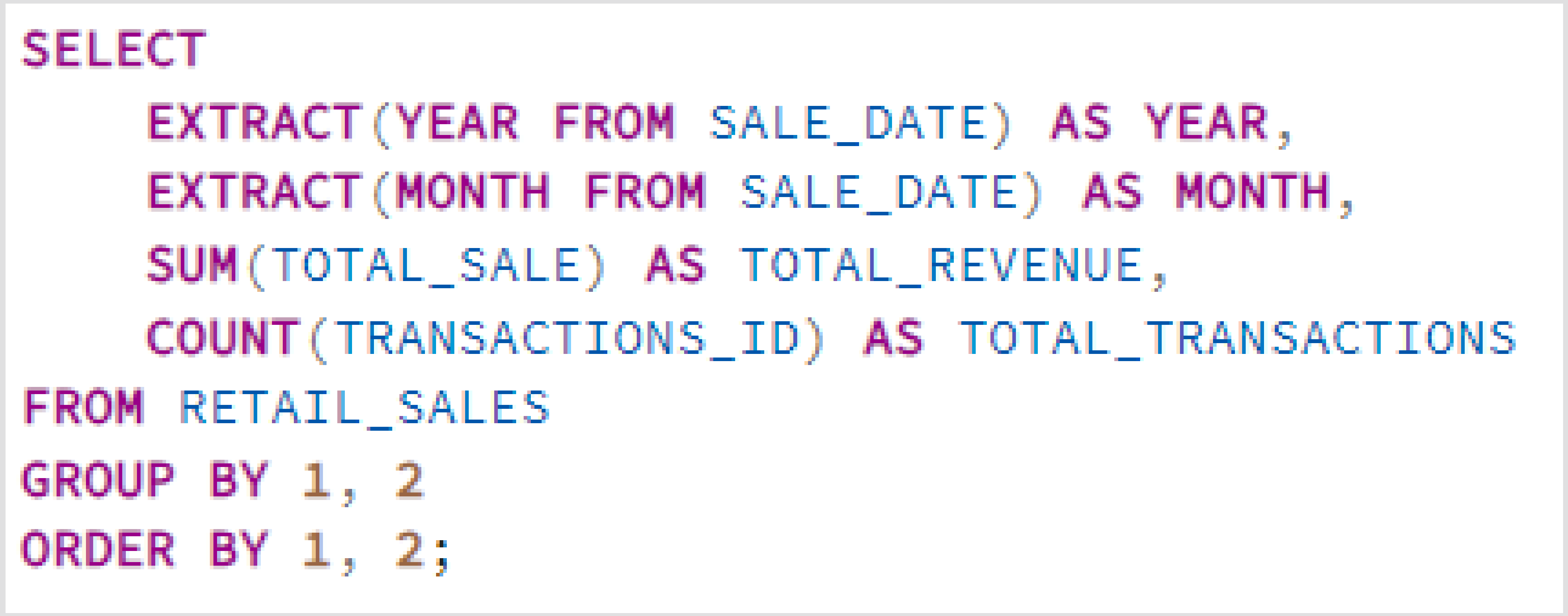


```
SELECT
    CATEGORY,
    GENDER,
    COUNT(TRANSACTIONS_ID) AS TOTAL_ORDERS
FROM
    RETAIL_SALES
GROUP BY
    1,2
ORDER BY
    1
```



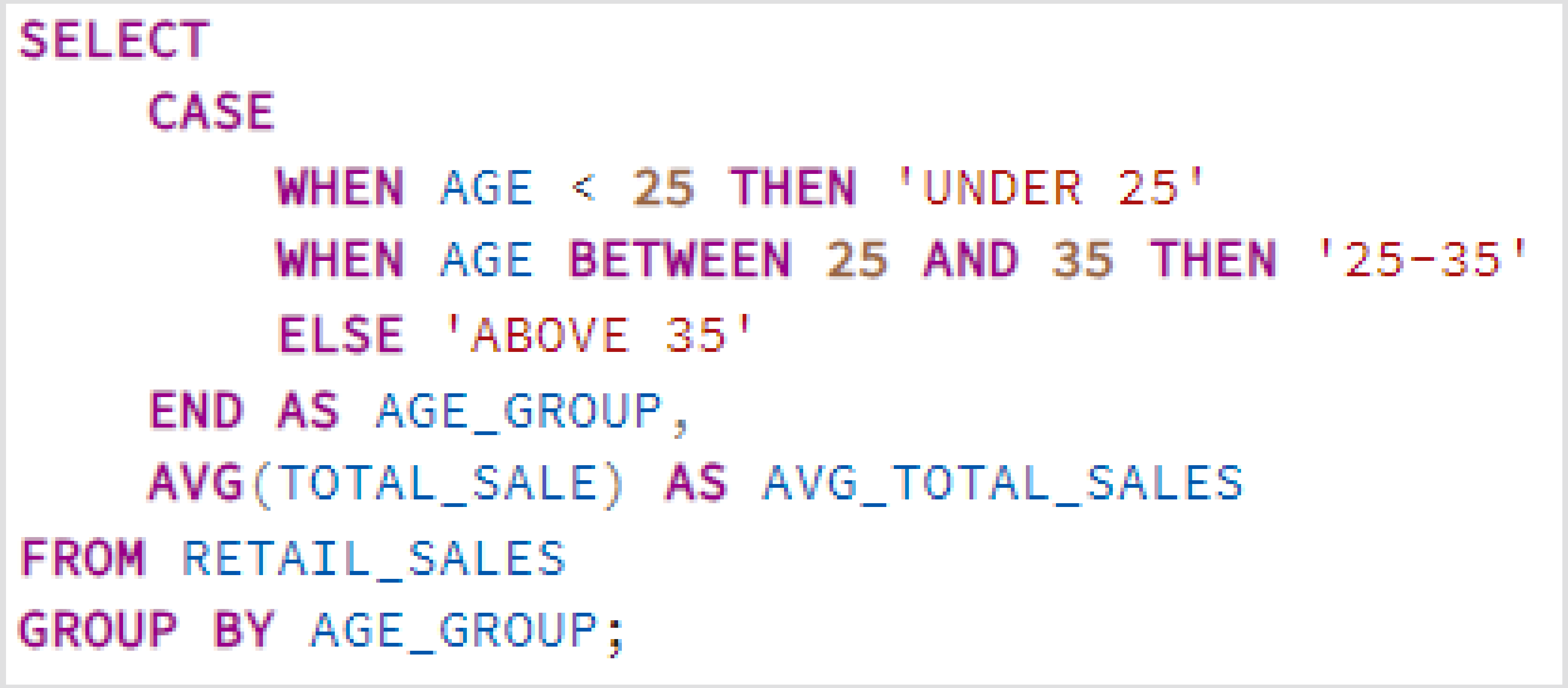
category character varying (15)	gender character varying (15)	total_trans bigint
Beauty	Female	330
Beauty	Male	282
Clothing	Female	347
Clothing	Male	354
Electronics	Male	344
Electronics	Female	340

8. Write a query to calculate the total revenue and the number of transactions for each month.

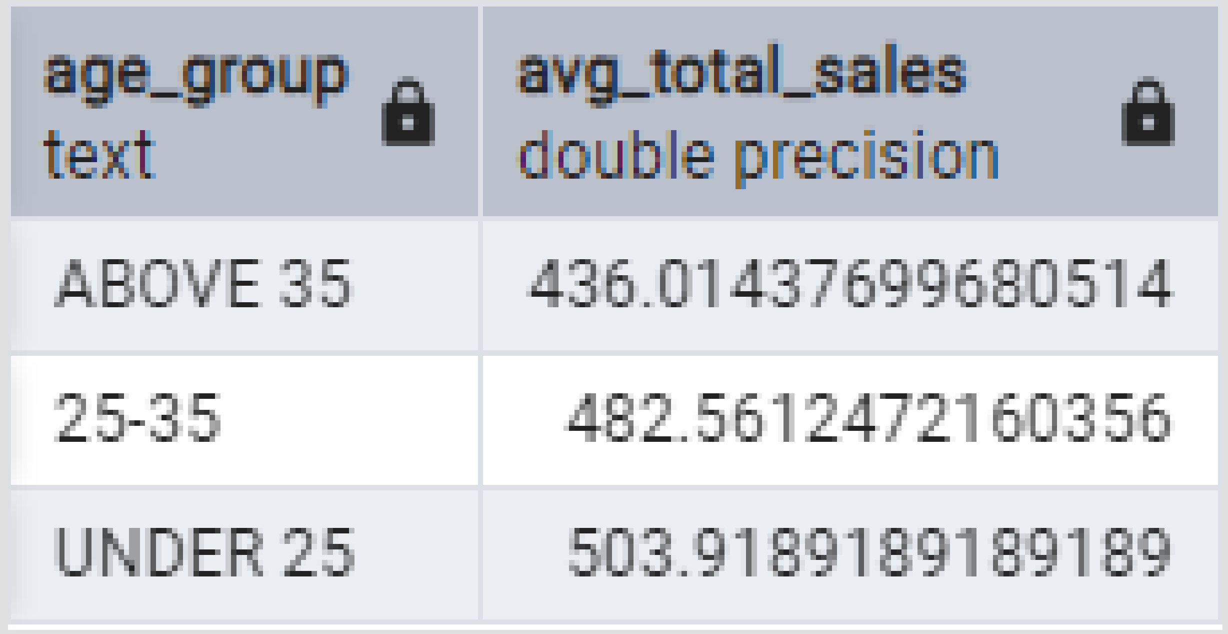


```sql
SELECT
    EXTRACT(YEAR FROM SALE_DATE) AS YEAR,
    EXTRACT(MONTH FROM SALE_DATE) AS MONTH,
    SUM(TOTAL_SALE) AS TOTAL_REVENUE,
    COUNT(TRANSACTIONS_ID) AS TOTAL_TRANSACTIONS
FROM RETAIL_SALES
GROUP BY 1, 2
ORDER BY 1, 2;
```

9. Write a query to calculate the average total sales for customers grouped by their age group (e.g., under 25, 25–35, above 35).

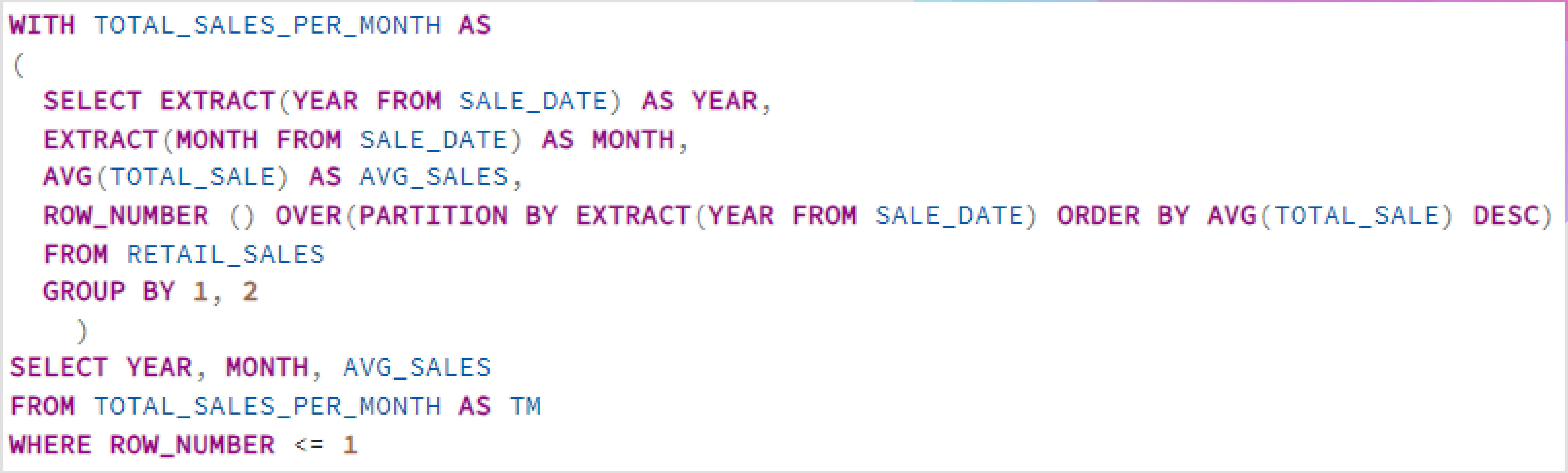


```
SELECT
    CASE
        WHEN AGE < 25 THEN 'UNDER 25'
        WHEN AGE BETWEEN 25 AND 35 THEN '25-35'
        ELSE 'ABOVE 35'
    END AS AGE_GROUP,
    AVG(TOTAL_SALE) AS AVG_TOTAL_SALES
FROM RETAIL_SALES
GROUP BY AGE_GROUP;
```

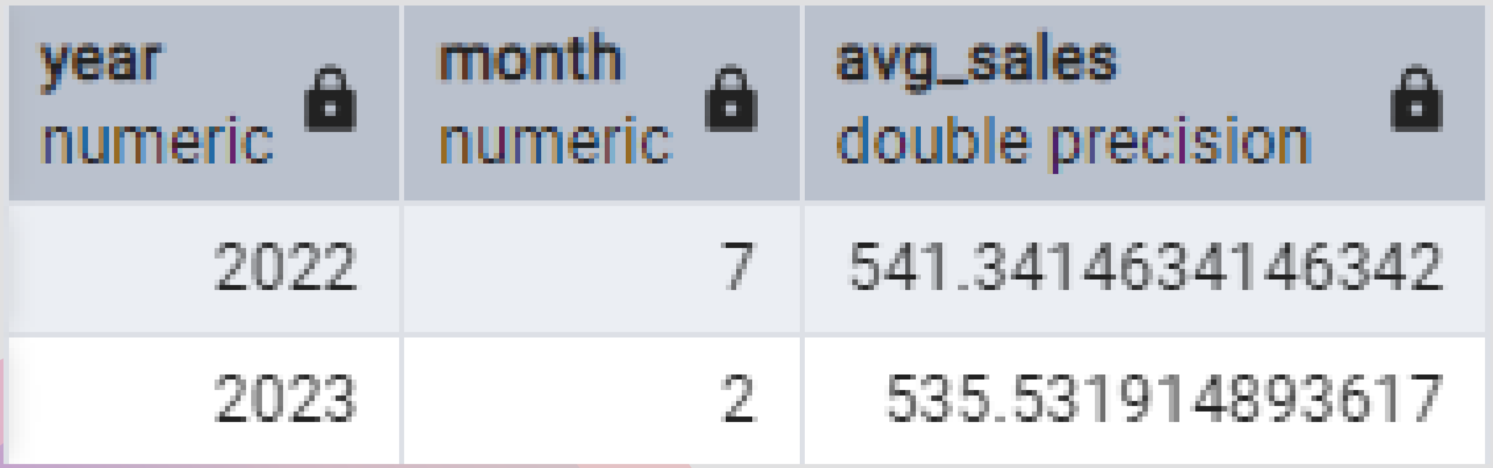


age_group text	avg_total_sales double precision
ABOVE 35	436.01437699680514
25-35	482.5612472160356
UNDER 25	503.9189189189189

10. Write a SQL query to calculate the average sale for each month. Find out best selling month in each year

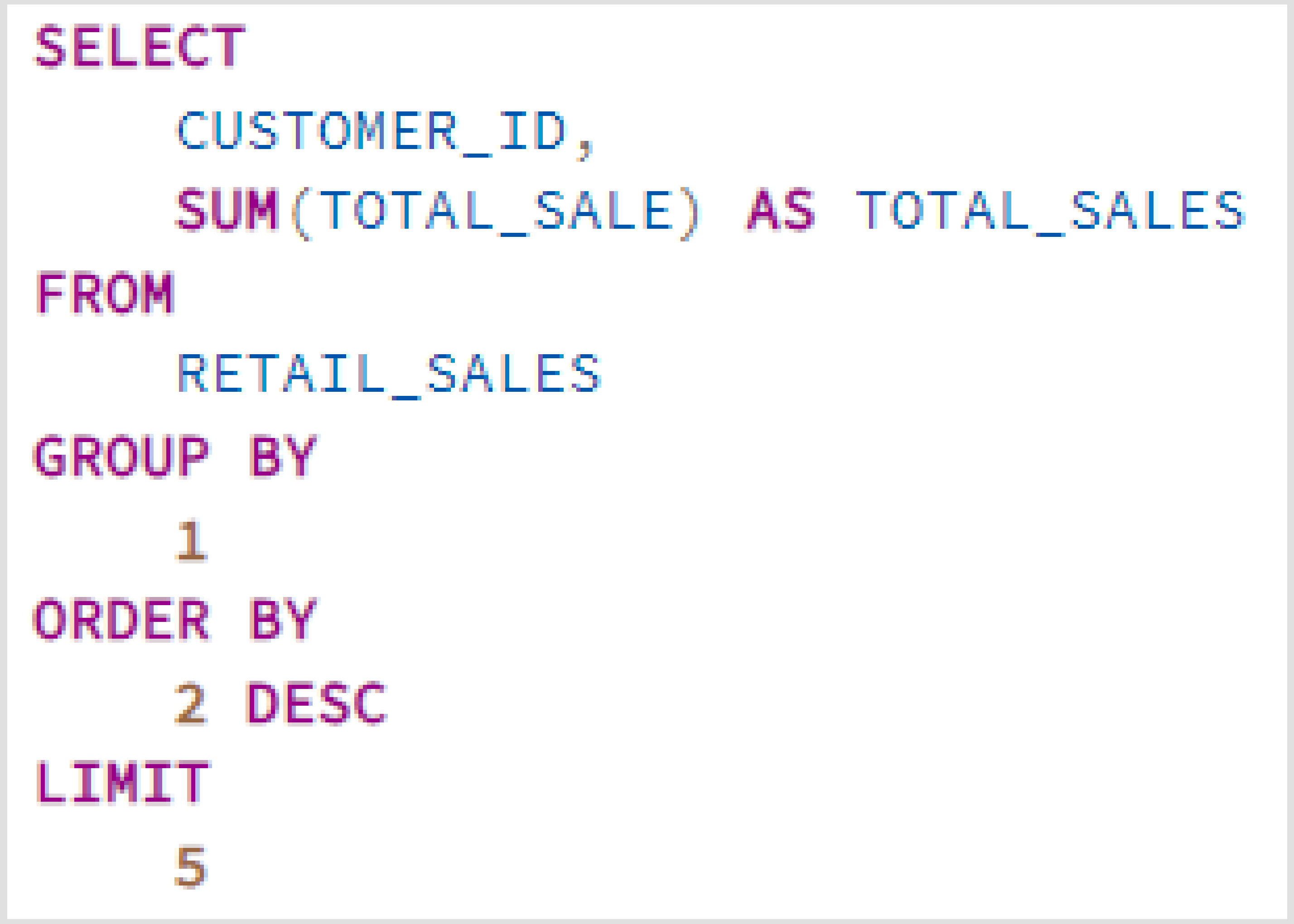


```
WITH TOTAL_SALES_PER_MONTH AS
(
  SELECT EXTRACT(YEAR FROM SALE_DATE) AS YEAR,
  EXTRACT(MONTH FROM SALE_DATE) AS MONTH,
  AVG(TOTAL_SALE) AS AVG_SALES,
  ROW_NUMBER () OVER(PARTITION BY EXTRACT(YEAR FROM SALE_DATE) ORDER BY AVG(TOTAL_SALE) DESC)
  FROM RETAIL_SALES
  GROUP BY 1, 2
    )
SELECT YEAR, MONTH, AVG_SALES
FROM TOTAL_SALES_PER_MONTH AS TM
WHERE ROW_NUMBER <= 1
```

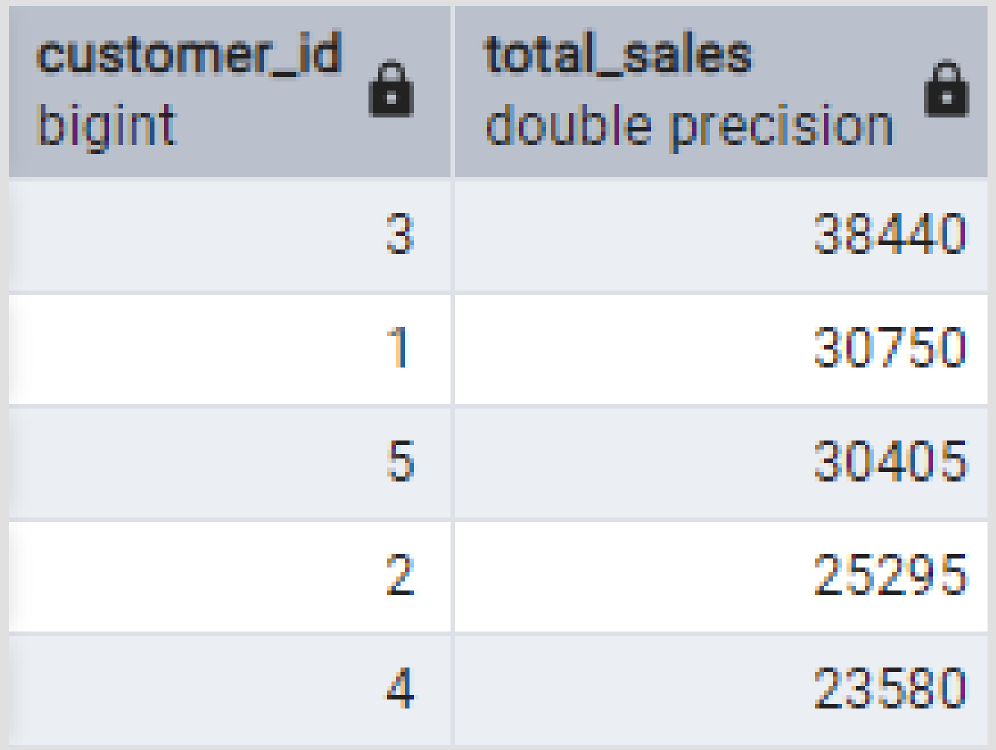


year numeric	month numeric	avg_sales double precision
2022	7	541.3414634146342
2023	2	535.531914893617

11. Write a SQL query to find the top 5 customers based on highest total sales.

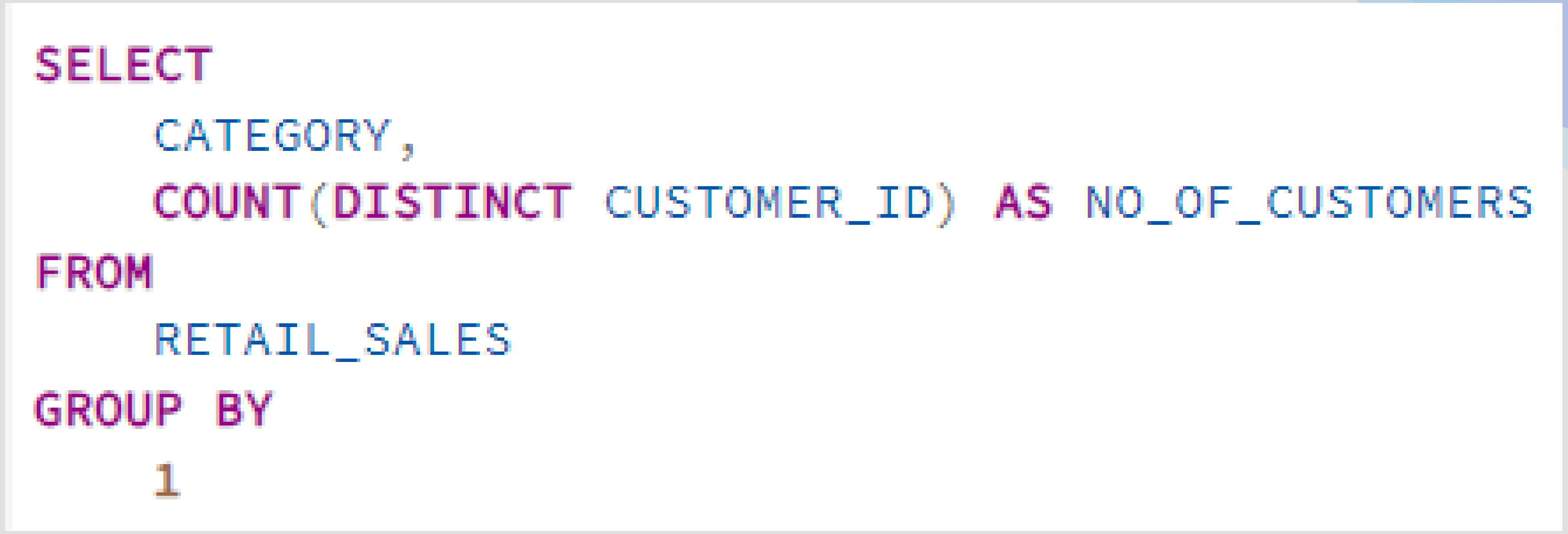


```
SELECT
    CUSTOMER_ID,
    SUM(TOTAL_SALE) AS TOTAL_SALES
FROM
    RETAIL_SALES
GROUP BY
    1
ORDER BY
    2 DESC
LIMIT
    5
```

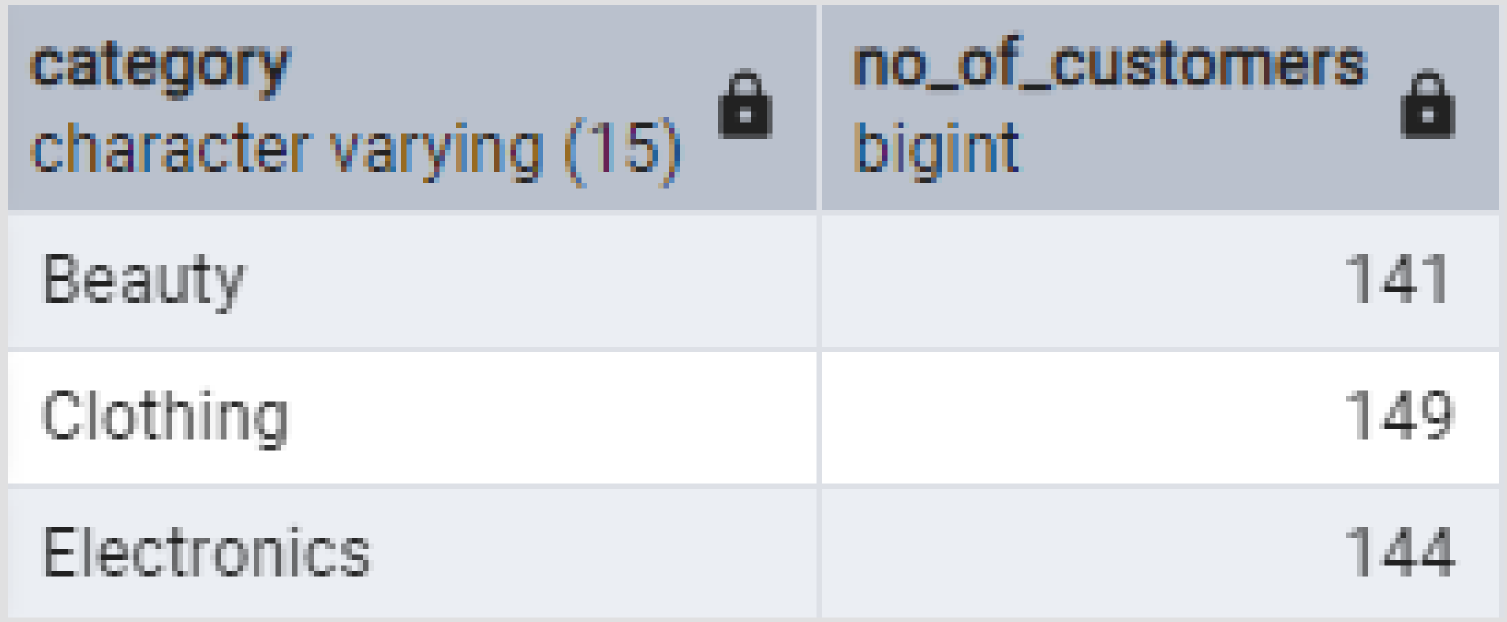


customer_id bigint	total_sales double precision
3	38440
1	30750
5	30405
2	25295
4	23580

12. Write a SQL query to find the number of unique customers who purchased items from each category.

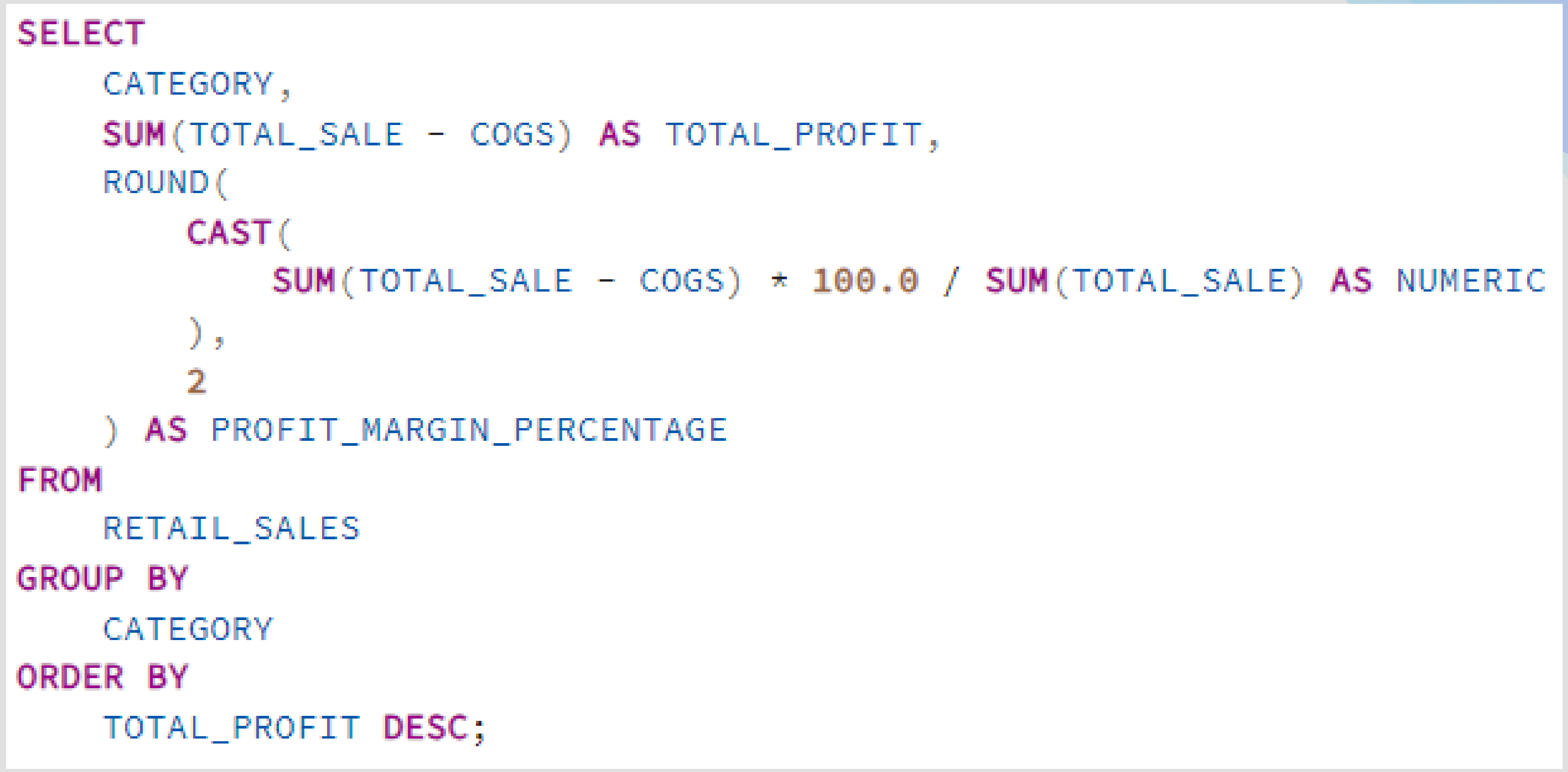


```
SELECT
    CATEGORY,
    COUNT(DISTINCT CUSTOMER_ID) AS NO_OF_CUSTOMERS
FROM
    RETAIL_SALES
GROUP BY
    1
```

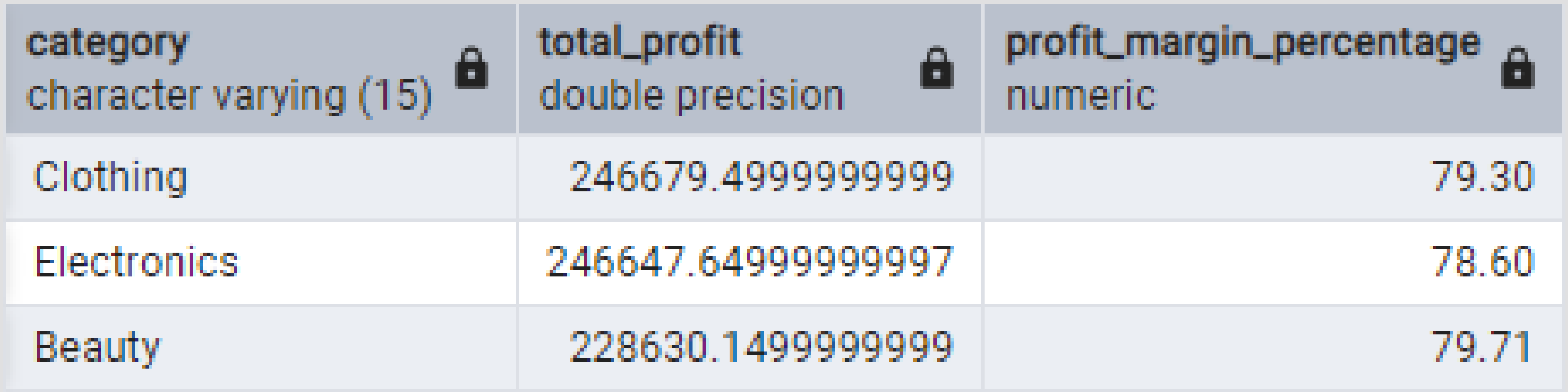


category character varying (15)	no_of_customers bigint
Beauty	141
Clothing	149
Electronics	144

13. Write a query to calculate the total profit and profit margin for each product category.

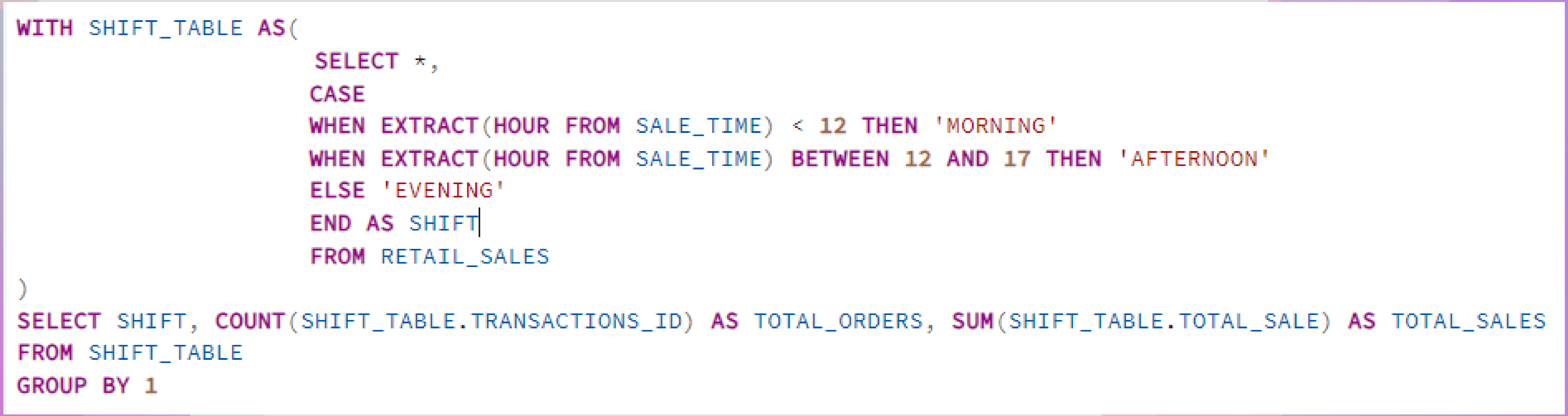


```
SELECT
    CATEGORY,
    SUM(TOTAL_SALE - COGS) AS TOTAL_PROFIT,
    ROUND(
        CAST(
            SUM(TOTAL_SALE - COGS) * 100.0 / SUM(TOTAL_SALE) AS NUMERIC
        ),
        2
    ) AS PROFIT_MARGIN_PERCENTAGE
FROM
    RETAIL_SALES
GROUP BY
    CATEGORY
ORDER BY
    TOTAL_PROFIT DESC;
```

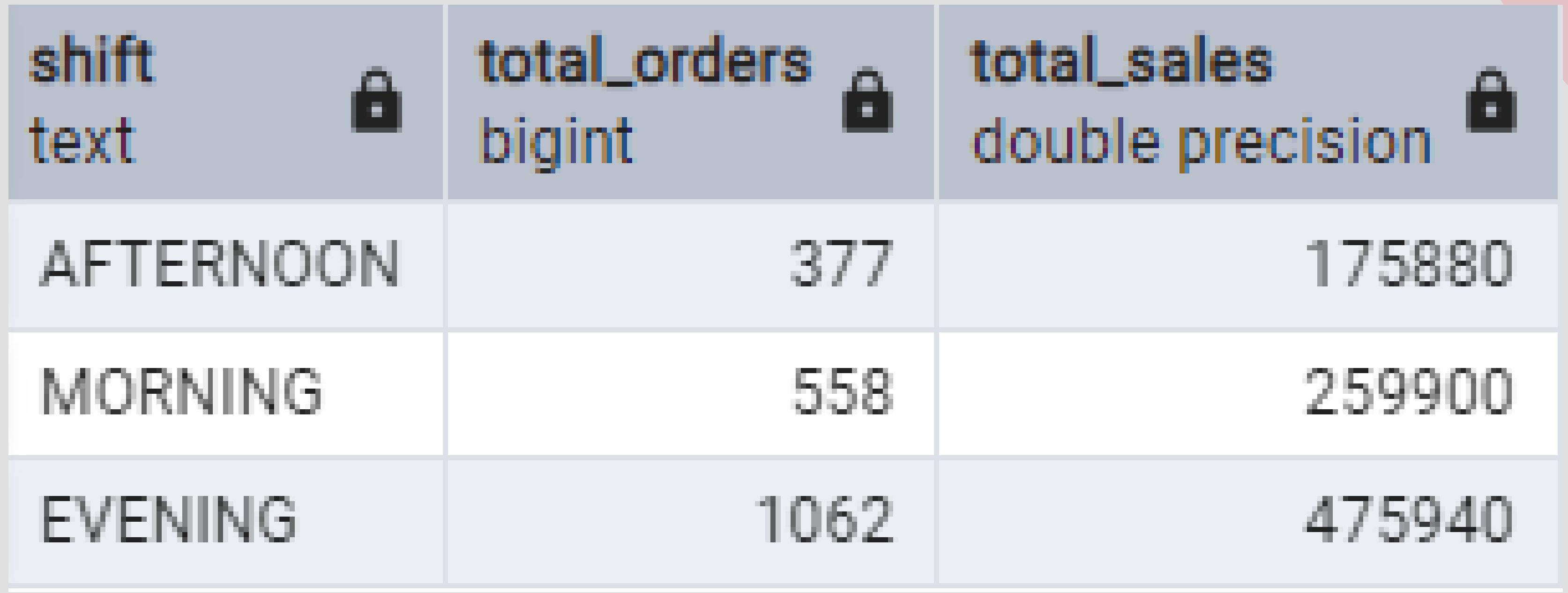


category character varying (15)	total_profit double precision	profit_margin_percentage numeric
Clothing	246679.4999999999	79.30
Electronics	246647.64999999997	78.60
Beauty	228630.1499999999	79.71

14. Write a SQL query to create each shift and number of orders (Example Morning <12, Afternoon Between 12 & 17, Evening >17).

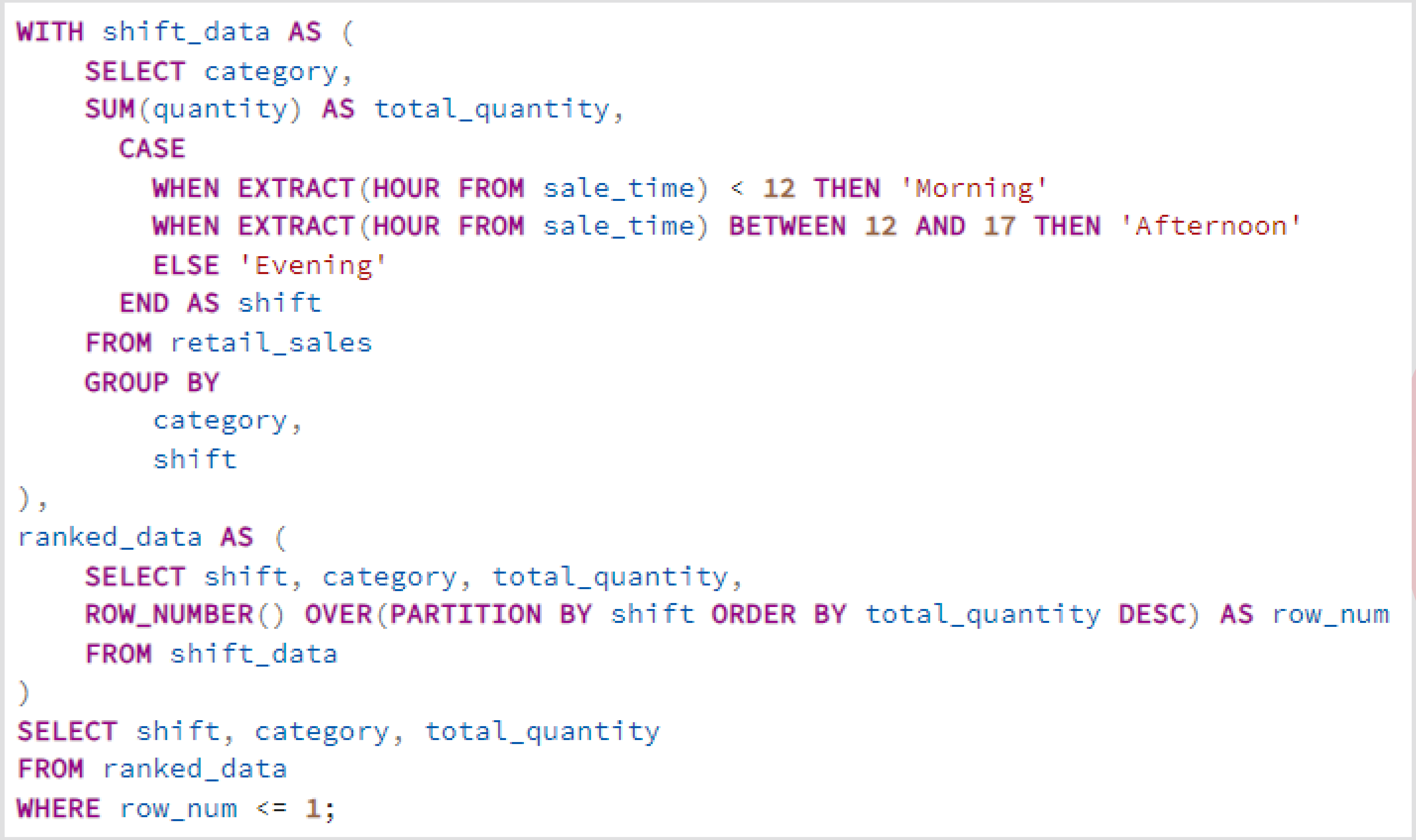


```
WITH SHIFT_TABLE AS(
                     SELECT *,
                     CASE
                     WHEN EXTRACT(HOUR FROM SALE_TIME) < 12 THEN 'MORNING'
                     WHEN EXTRACT(HOUR FROM SALE_TIME) BETWEEN 12 AND 17 THEN 'AFTERNOON'
                     ELSE 'EVENING'
                     END AS SHIFT
                     FROM RETAIL_SALES
)
SELECT SHIFT, COUNT(SHIFT_TABLE.TRANSACTIONS_ID) AS TOTAL_ORDERS, SUM(SHIFT_TABLE.TOTAL_SALE) AS TOTAL_SALES
FROM SHIFT_TABLE
GROUP BY 1
```

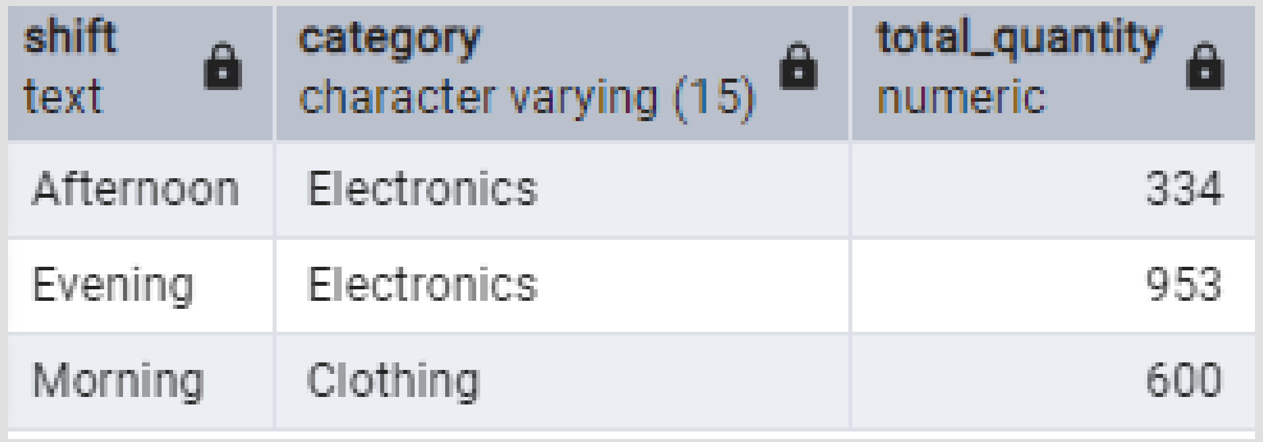


shift text	total_orders bigint	total_sales double precision
AFTERNOON	377	175880
MORNING	558	259900
EVENING	1062	475940

15. Write a query to determine the top selling product category during each shift (Morning, Afternoon, Evening).



```sql
WITH shift_data AS (
    SELECT category,
    SUM(quantity) AS total_quantity,
      CASE
        WHEN EXTRACT(HOUR FROM sale_time) < 12 THEN 'Morning'
        WHEN EXTRACT(HOUR FROM sale_time) BETWEEN 12 AND 17 THEN 'Afternoon'
        ELSE 'Evening'
      END AS shift
    FROM retail_sales
    GROUP BY
        category,
        shift
),
ranked_data AS (
    SELECT shift, category, total_quantity,
    ROW_NUMBER() OVER(PARTITION BY shift ORDER BY total_quantity DESC) AS row_num
    FROM shift_data
)
SELECT shift, category, total_quantity
FROM ranked_data
WHERE row_num <= 1;
```



shift text	category character varying (15)	total_quantity numeric
Afternoon	Electronics	334
Evening	Electronics	953
Morning	Clothing	600

THANK YOU