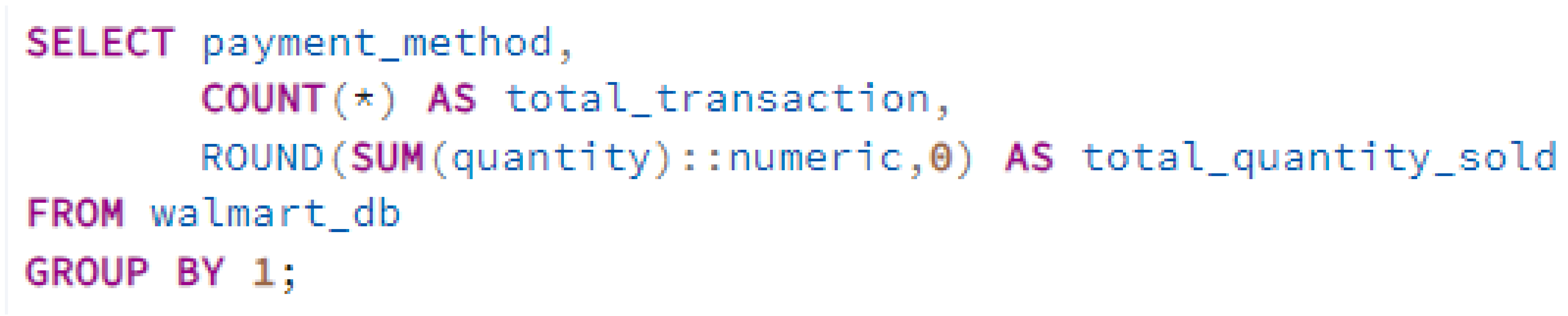
WALMART

SALES REPORT

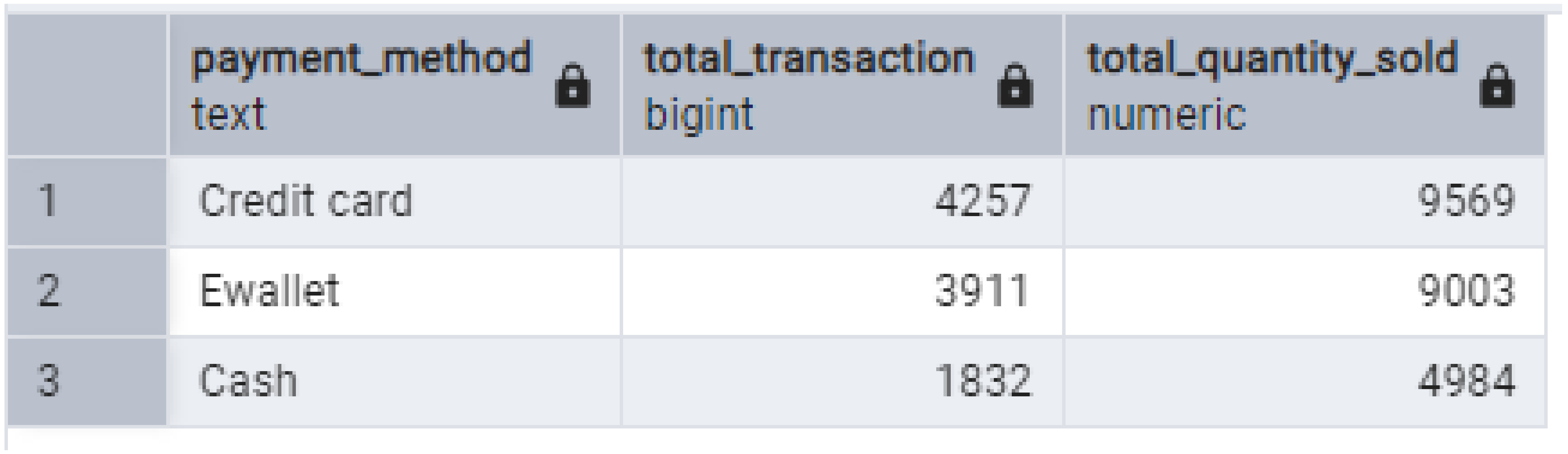
A detailed and complete sales report will be very helpful to see which parts need to be changed or improved.

Walmart

Q.1 FIND THE DIFFERENT PAYMENT METHOD AND NUMBER OF TRANSACTIONS, NUMBER OF QUANTITY SOLD.



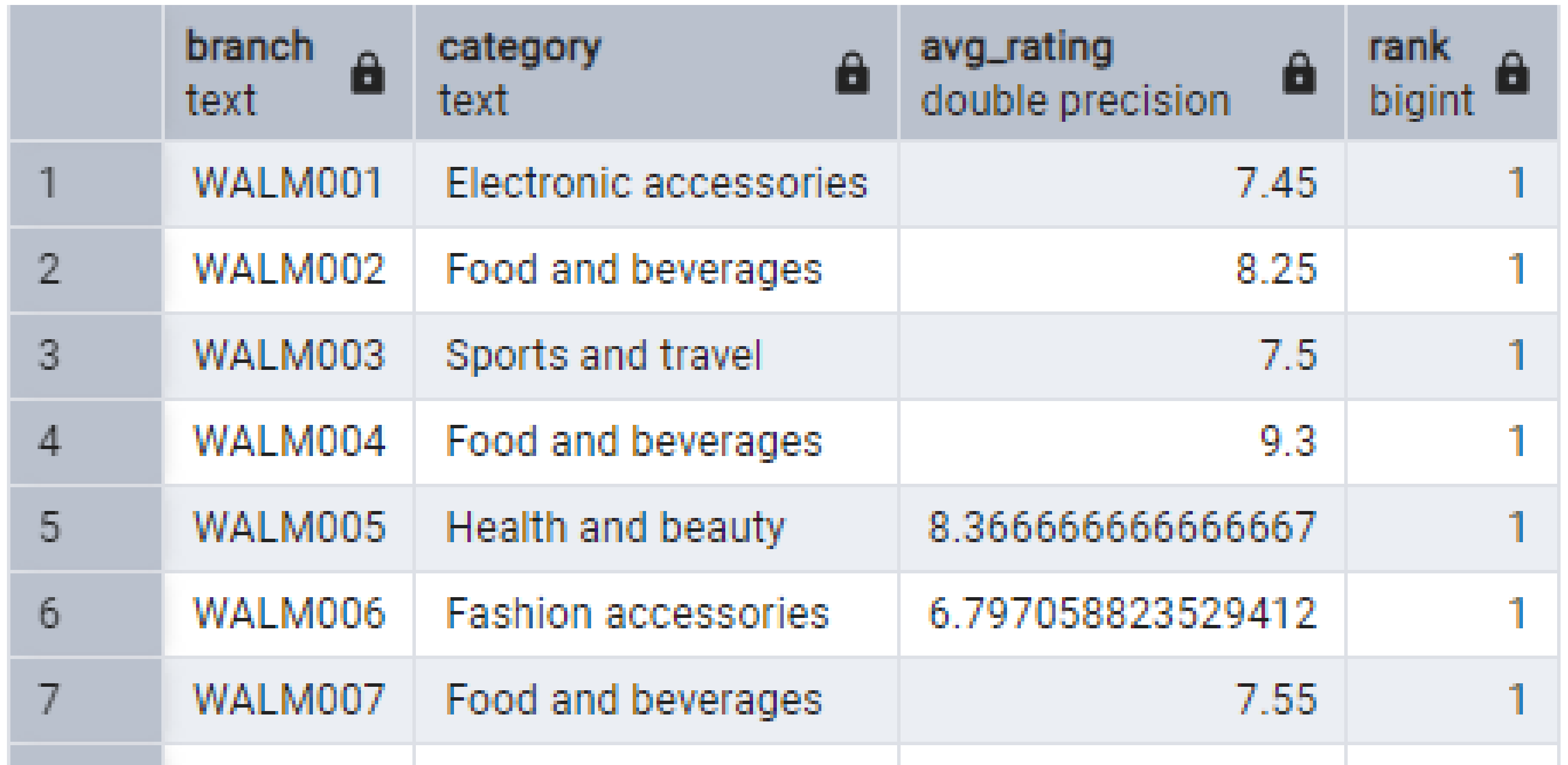
```sql
SELECT payment_method,
       COUNT(*) AS total_transaction,
       ROUND(SUM(quantity)::numeric,0) AS total_quantity_sold
FROM walmart_db
GROUP BY 1;
```



	payment_method text	total_transaction bigint	total_quantity_sold numeric
1	Credit card	4257	9569
2	Ewallet	3911	9003
3	Cash	1832	4984

Q 2. IDENTIFY THE HIGHEST RATED CATEGORY IN EACH BRANCH, DISPLAYING THE BRANCH, CATEGORY, AVERAGE RATING.

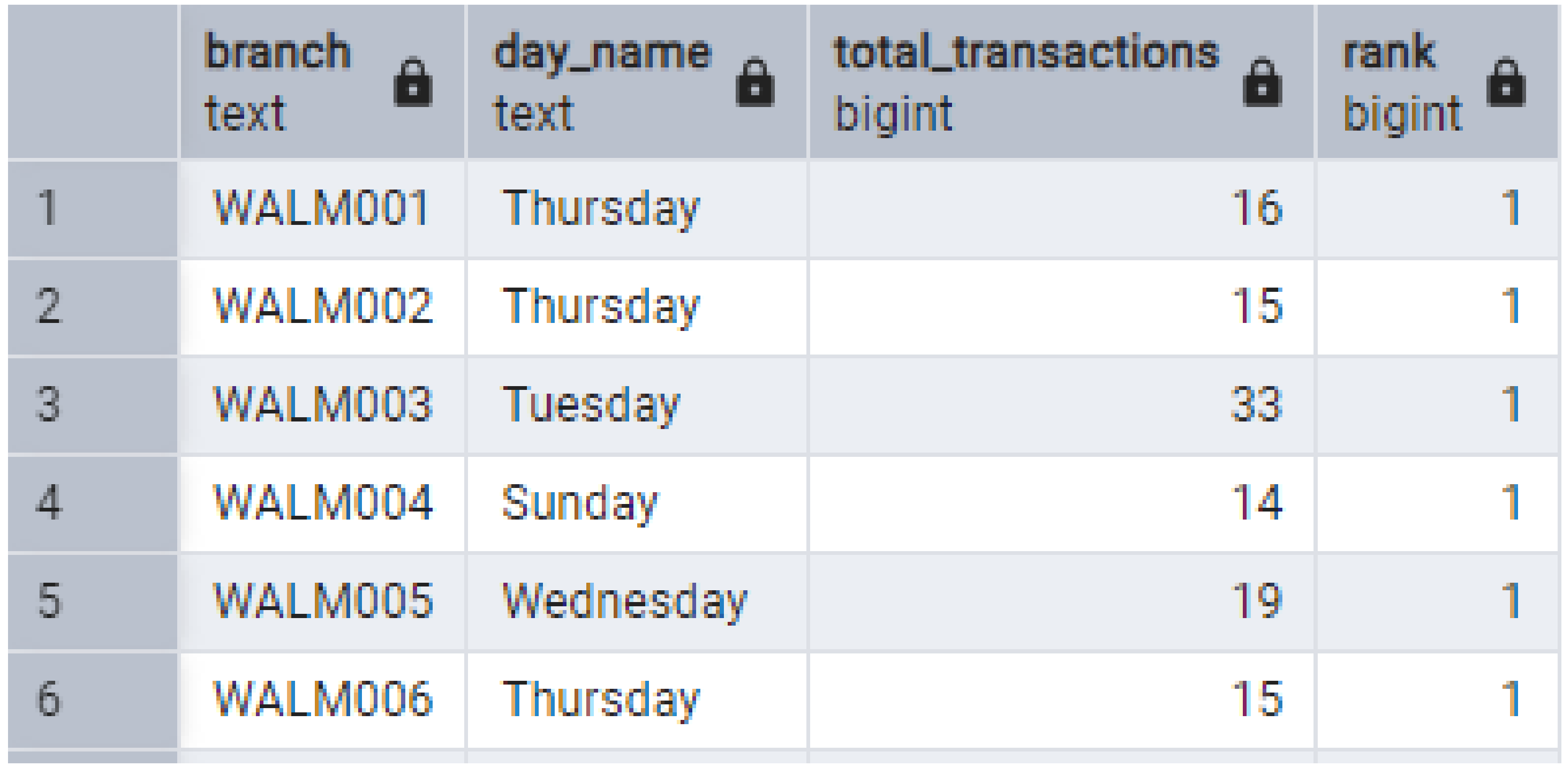
```sql
SELECT * FROM walmart_db;

SELECT *
FROM
(
SELECT branch,
       category,
       AVG(rating) AS avg_rating,
       RANK() OVER(PARTITION BY branch ORDER BY AVG(rating) DESC) AS rank
FROM walmart_db
GROUP BY 1,2
)
WHERE rank = 1;
```



	branch text	category text	avg_rating double precision	rank bigint
1	WALM001	Electronic accessories	7.45	1
2	WALM002	Food and beverages	8.25	1
3	WALM003	Sports and travel	7.5	1
4	WALM004	Food and beverages	9.3	1
5	WALM005	Health and beauty	8.366666666666667	1
6	WALM006	Fashion accessories	6.797058823529412	1
7	WALM007	Food and beverages	7.55	1

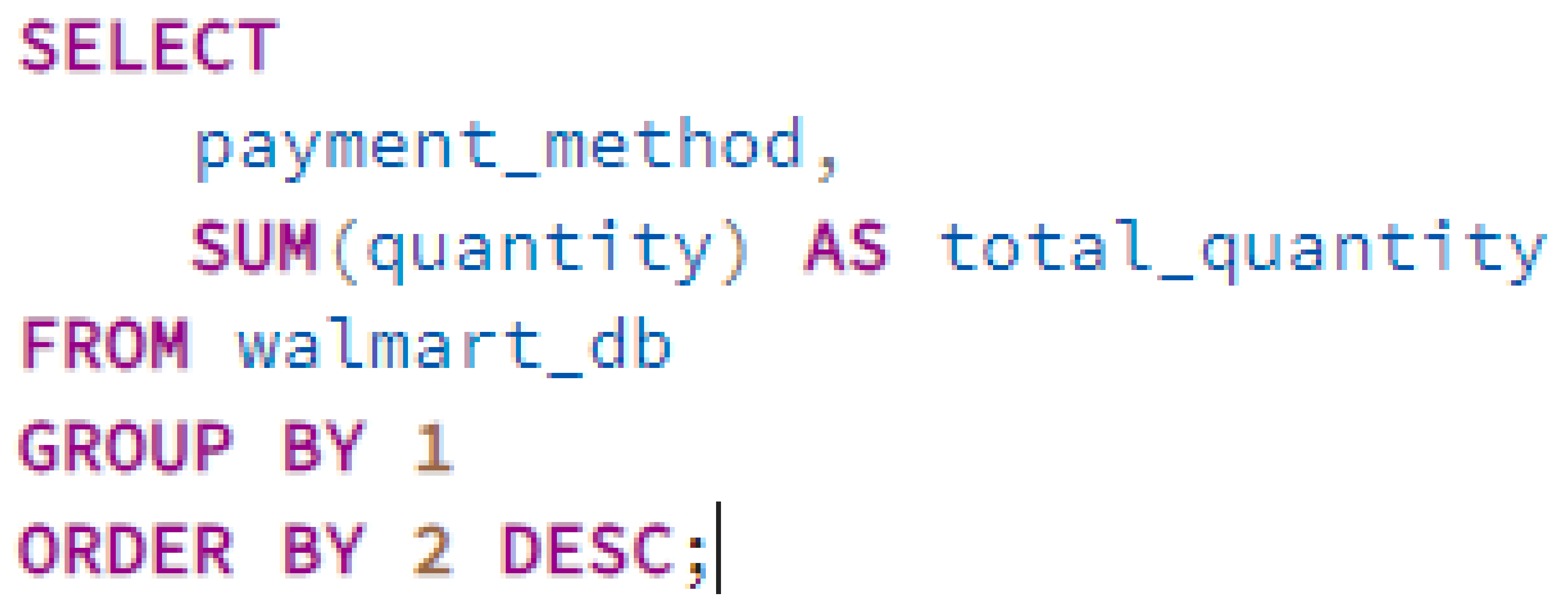
Q 3 IDENTIFY THE BUSIEST DAY FOR EACH BRANCH BASED ON THE NUMBER OF TRANSACTIONS.

```sql
SELECT *
FROM
(SELECT
    branch,
    TO_CHAR(TO_DATE(date,'DD/MM/YY'),'Day') AS day_name,
    COUNT(*) AS total_transactions,
    RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS rank
FROM walmart_db
GROUP BY 1,2)
WHERE rank=1;
```

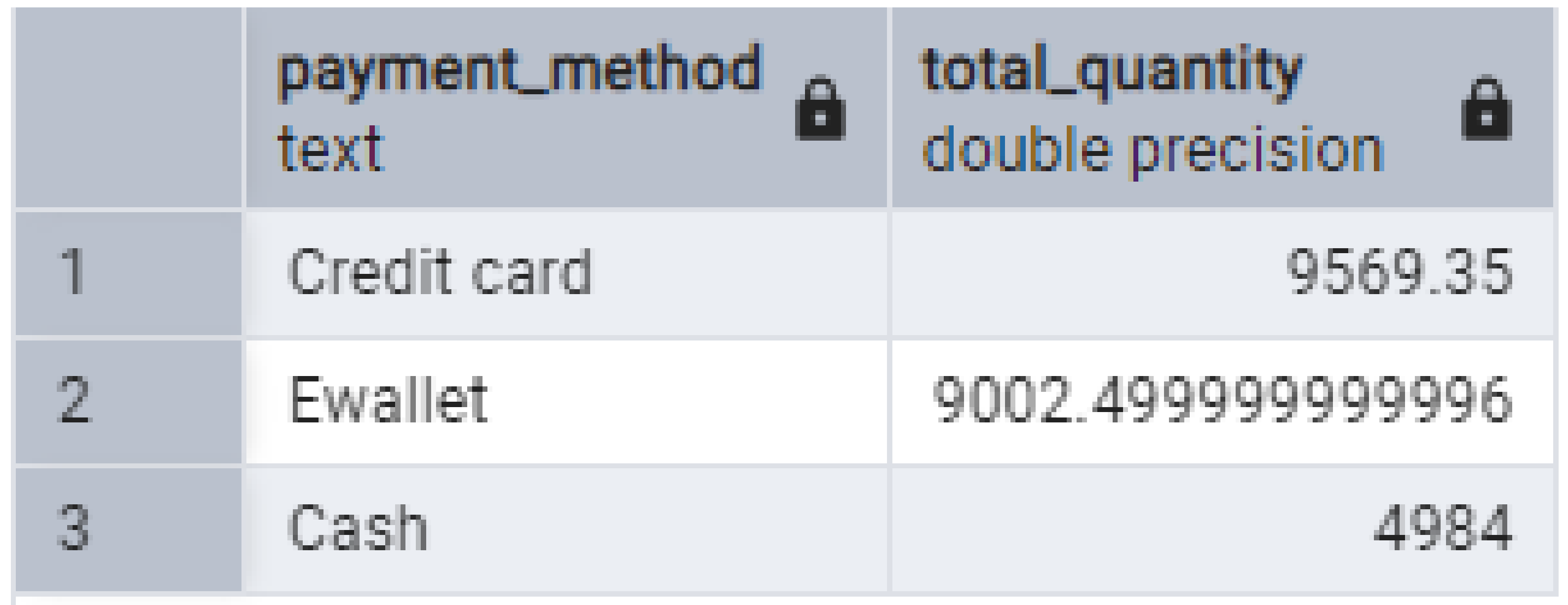


	branch text	day_name text	total_transactions bigint	rank bigint
1	WALM001	Thursday	16	1
2	WALM002	Thursday	15	1
3	WALM003	Tuesday	33	1
4	WALM004	Sunday	14	1
5	WALM005	Wednesday	19	1
6	WALM006	Thursday	15	1

Q4. CALCULATE THE TOTAL QUANTITY OF ITEMS SOLD PER PAYMENT METHOD. LIST PAYMENT_METHOD AND TOTAL QUANTITY.

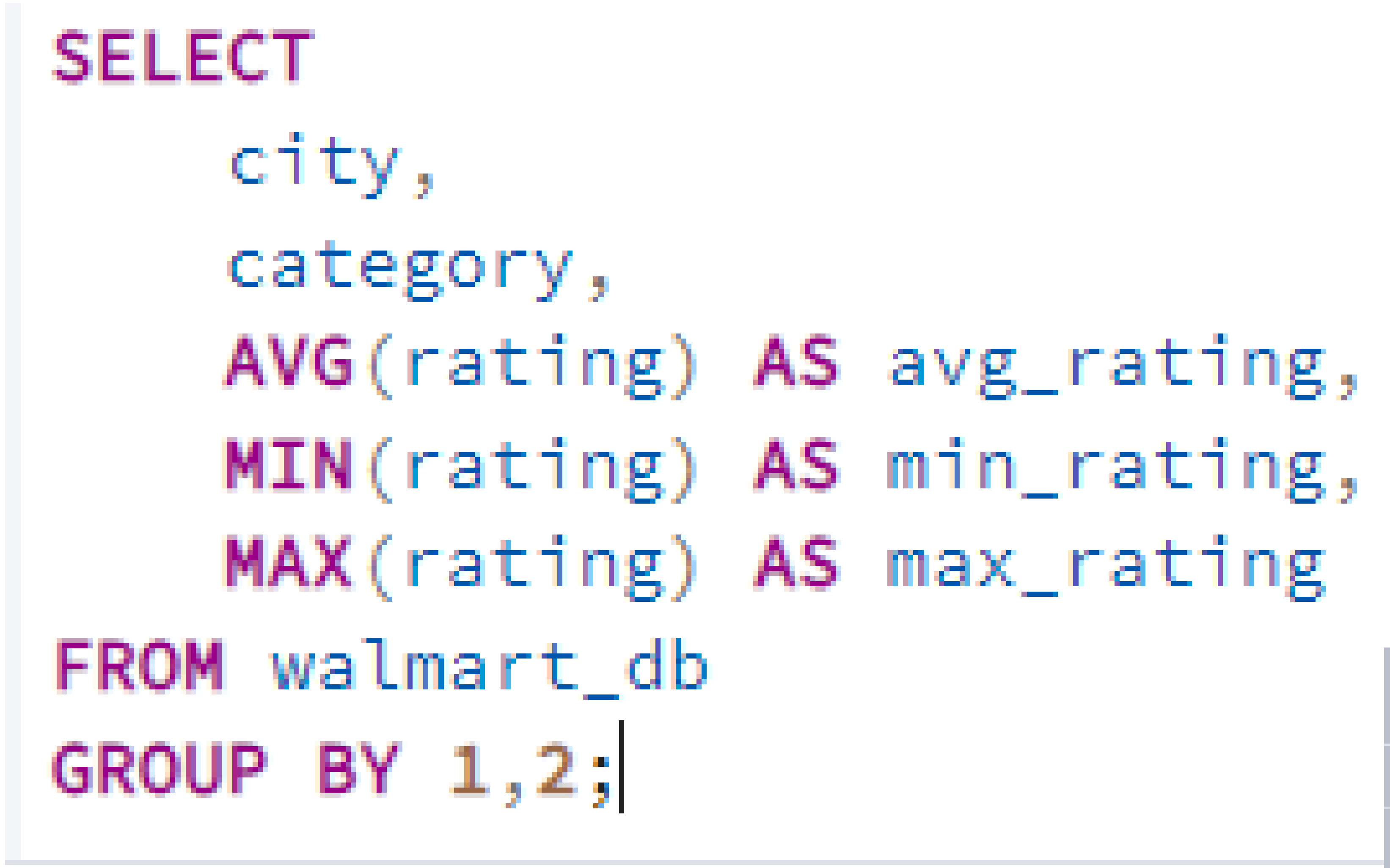


```sql
SELECT
    payment_method,
    SUM(quantity) AS total_quantity
FROM walmart_db
GROUP BY 1
ORDER BY 2 DESC;
```

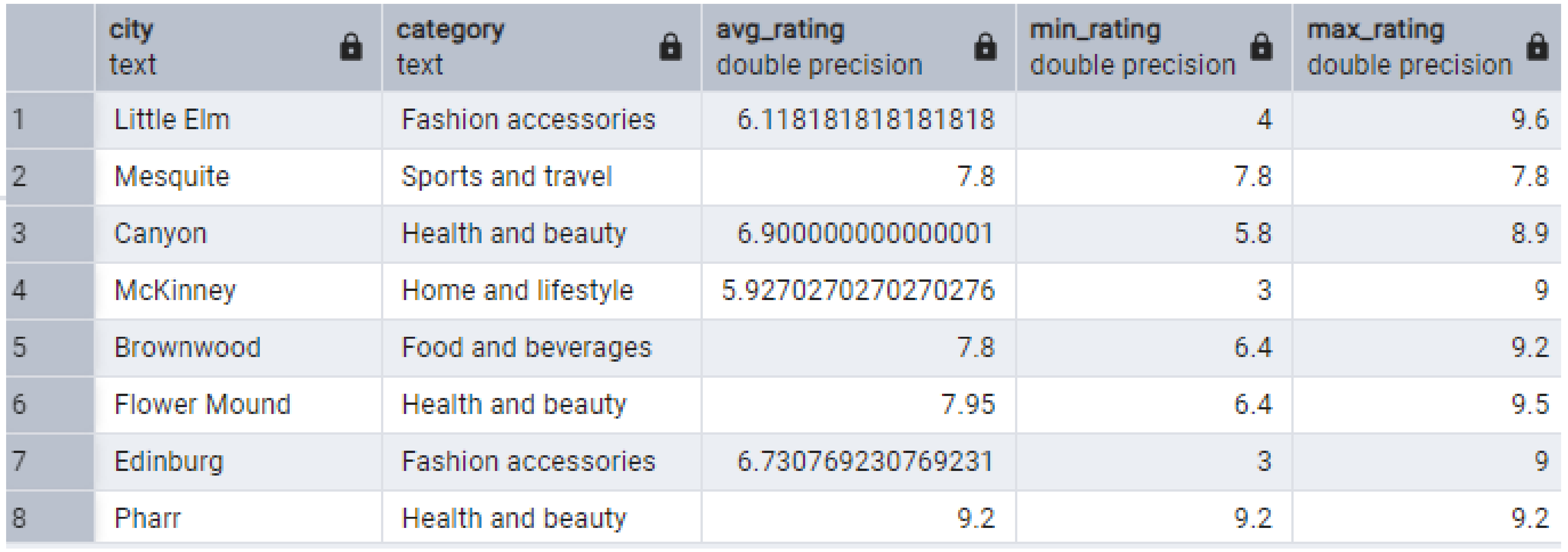


	payment_method text	total_quantity double precision
1	Credit card	9569.35
2	Ewallet	9002.499999999996
3	Cash	4984

Q 5. DETERMINE THE AVERAGE, MINIMUM, AND MAXIMUM RATING OF CATEGORY FOR EACH CITY. LIST THE CITY, AVERAGE RATING, MIN RATING, AND MAX RATING.



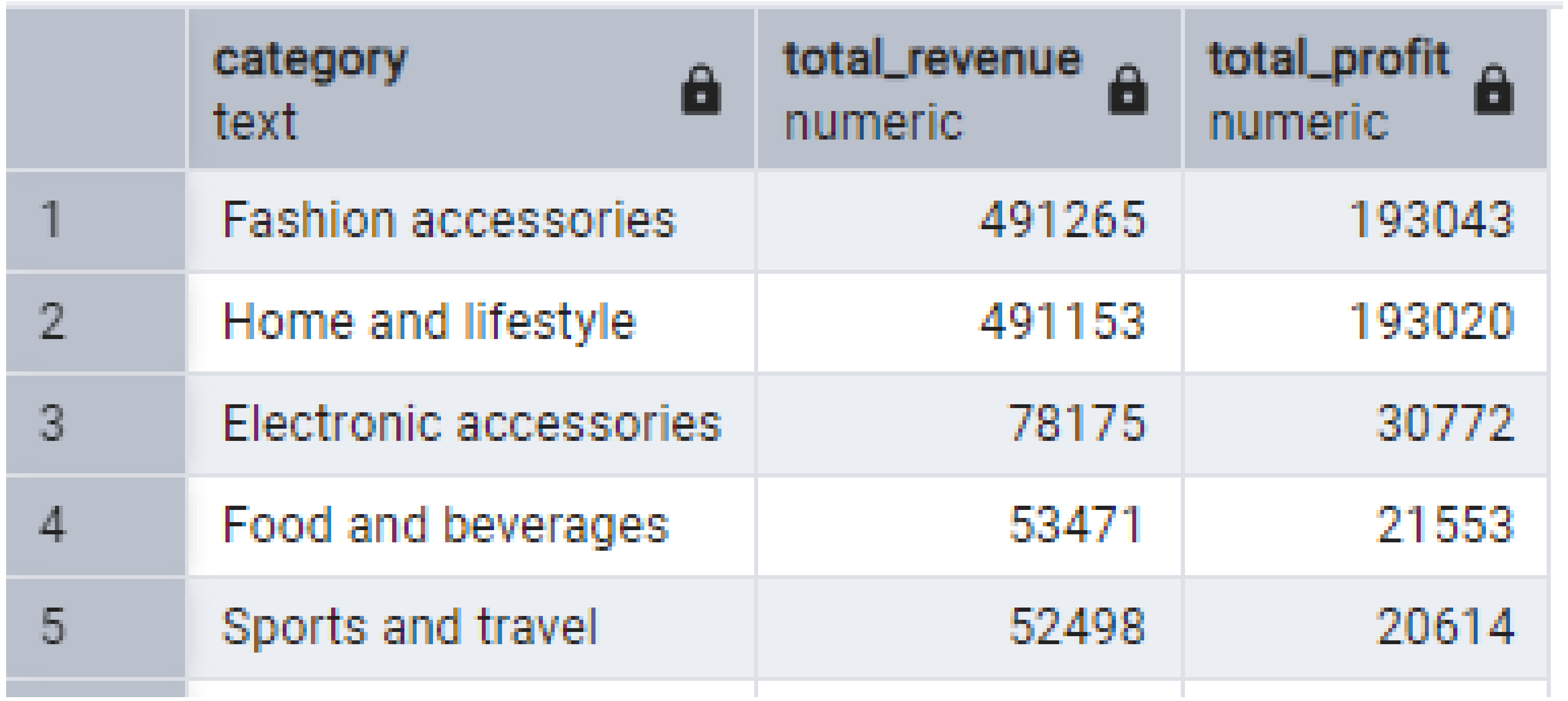
```
SELECT
    city,
    category,
    AVG(rating) AS avg_rating,
    MIN(rating) AS min_rating,
    MAX(rating) AS max_rating
FROM walmart_db
GROUP BY 1,2;
```



	city text	category text	avg_rating double precision	min_rating double precision	max_rating double precision
1	Little Elm	Fashion accessories	6.118181818181818	4	9.6
2	Mesquite	Sports and travel	7.8	7.8	7.8
3	Canyon	Health and beauty	6.900000000000001	5.8	8.9
4	McKinney	Home and lifestyle	5.9270270270270276	3	9
5	Brownwood	Food and beverages	7.8	6.4	9.2
6	Flower Mound	Health and beauty	7.95	6.4	9.5
7	Edinburg	Fashion accessories	6.730769230769231	3	9
8	Pharr	Health and beauty	9.2	9.2	9.2

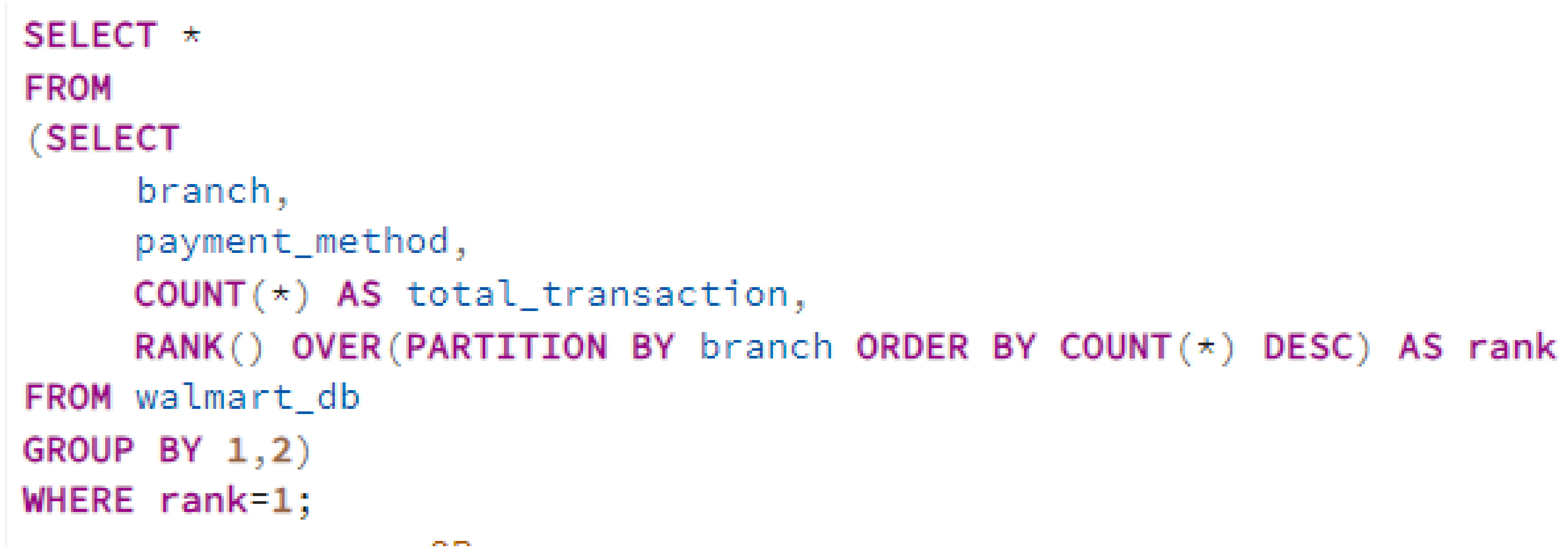
Q 6. CALCULATE THE TOTAL PROFIT FOR EACH CATEGORY BY CONSIDERING TOTAL_PROFIT AS (UNIT_PRICE*QUANTITY*PROFIT_MARGIN). LIST CATEGORY AND TOTAL_PROFIT, ORDERED FROM HIGHEST TO LOWEST PROFIT.

```
SELECT * FROM walmart_db;

SELECT
    category,
    ROUND(SUM(total_revenue)::numeric,0) AS total_revenue,
    ROUND(SUM(total_revenue*profit_margin)::numeric,0) AS total_profit
FROM walmart_db
GROUP BY 1
ORDER BY 3 DESC;
```

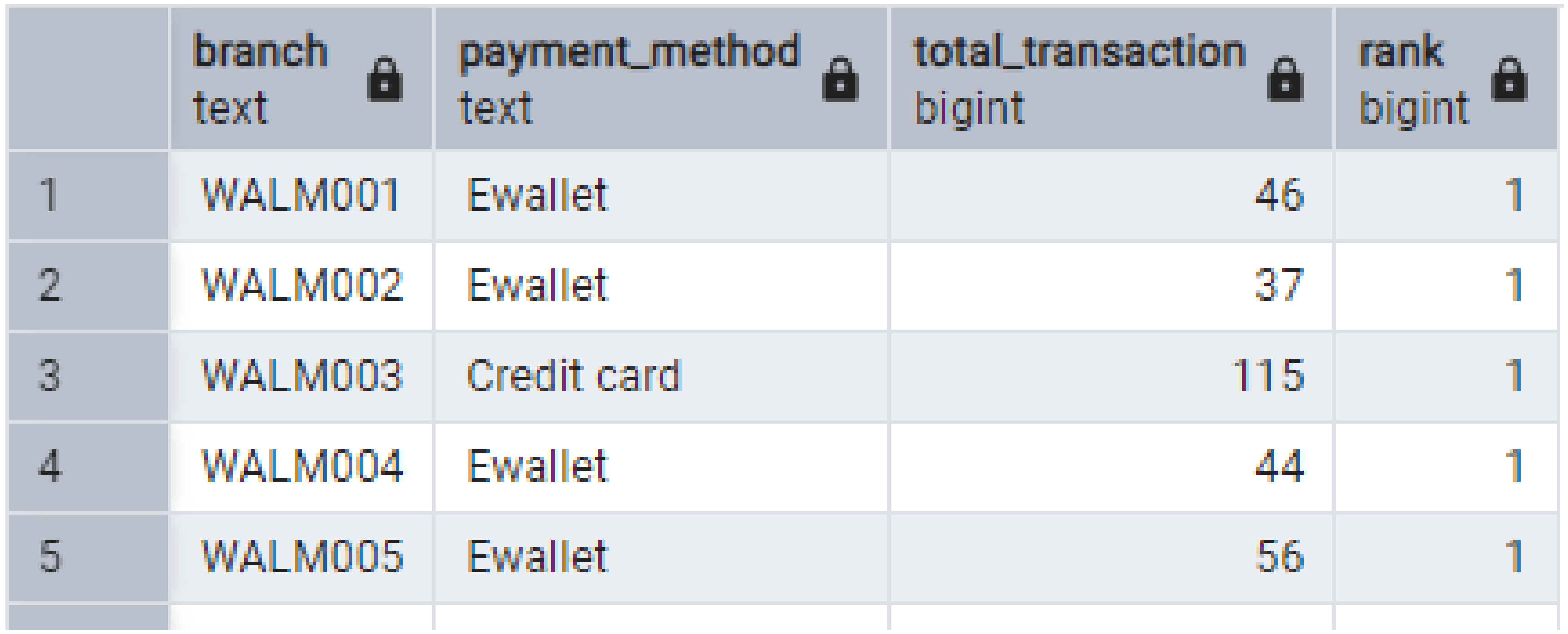


	category text	total_revenue numeric	total_profit numeric
1	Fashion accessories	491265	193043
2	Home and lifestyle	491153	193020
3	Electronic accessories	78175	30772
4	Food and beverages	53471	21553
5	Sports and travel	52498	20614

Q 7. DETERMINE THE MOST COMMON PAYMENT METHOD FOR EACH BRANCH. DISPLAY BRANCH AND THE PREFFERED PAYMENT METHOD.

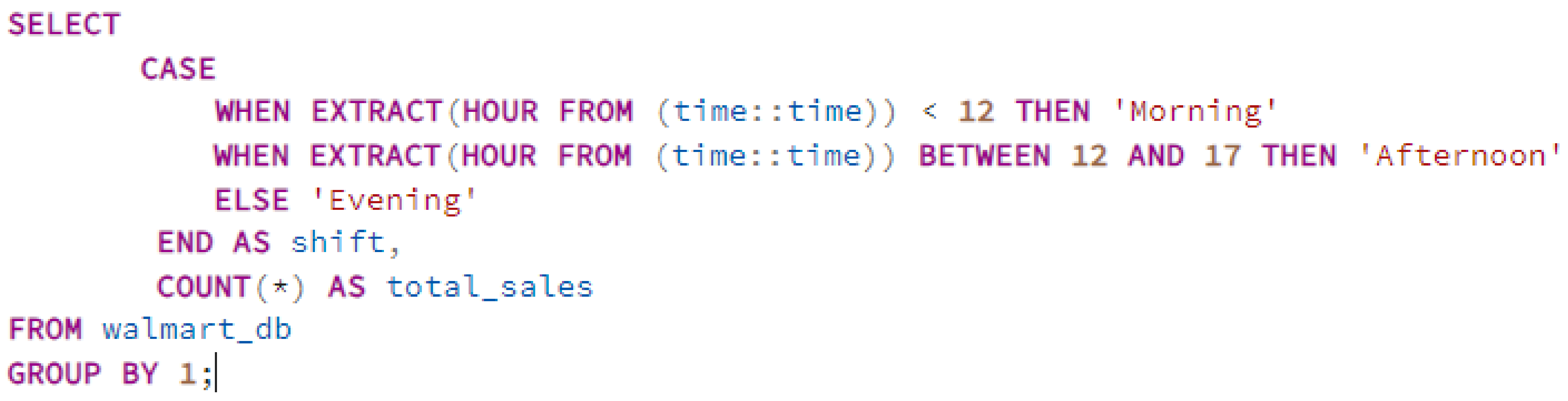


```
SELECT *
FROM
(SELECT
     branch,
     payment_method,
     COUNT(*) AS total_transaction,
     RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS rank
FROM walmart_db
GROUP BY 1,2)
WHERE rank=1;
```

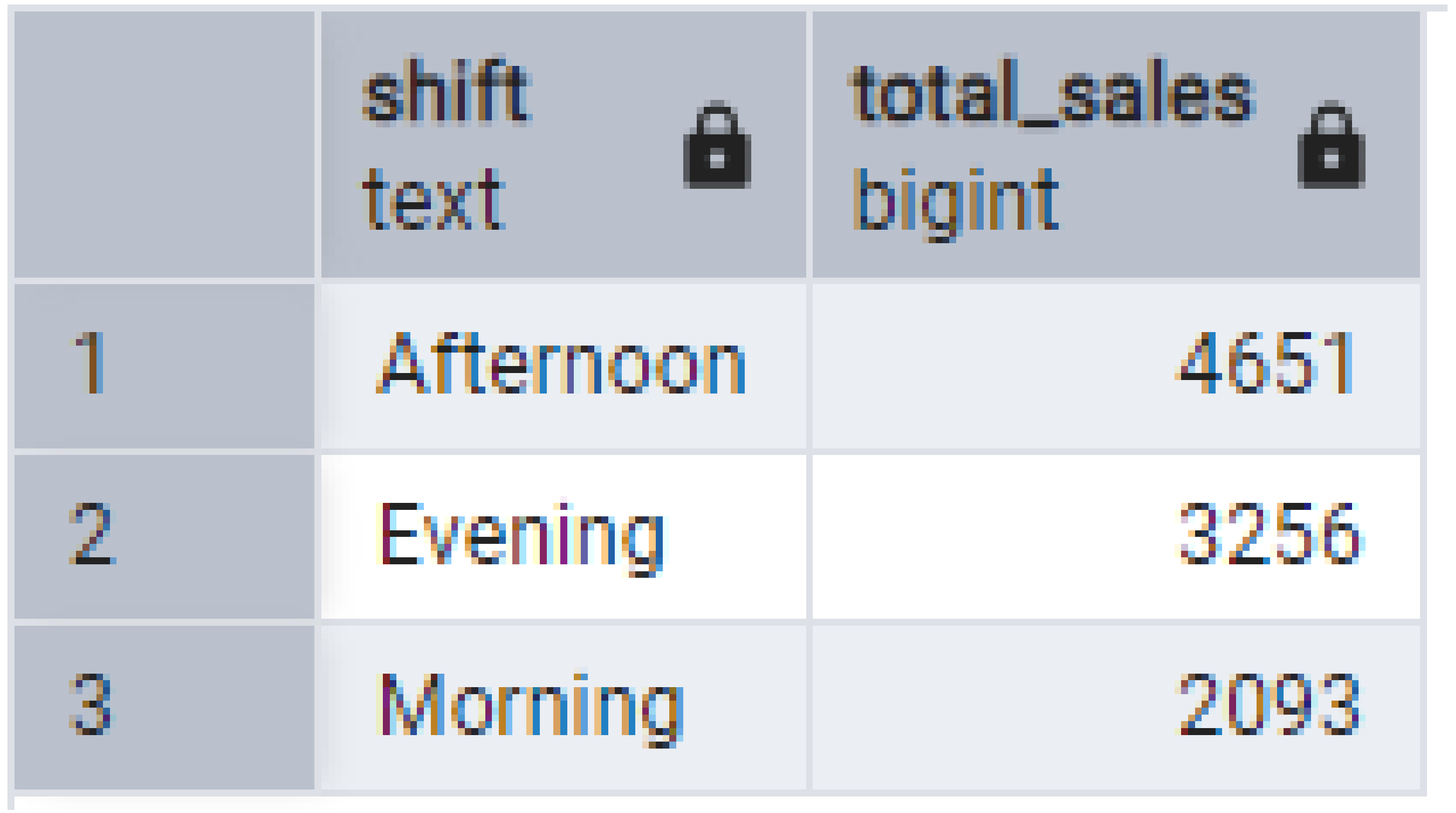


	branch text	payment_method text	total_transaction bigint	rank bigint
1	WALM001	Ewallet	46	1
2	WALM002	Ewallet	37	1
3	WALM003	Credit card	115	1
4	WALM004	Ewallet	44	1
5	WALM005	Ewallet	56	1

Q 8. CATEGORIZE SALES INTO 3 GROUP MORNING, AFTERNOON, EVENING. FIND OUT WHICH OF THE SHIFT AND THE NUMBER OF INVOICES.

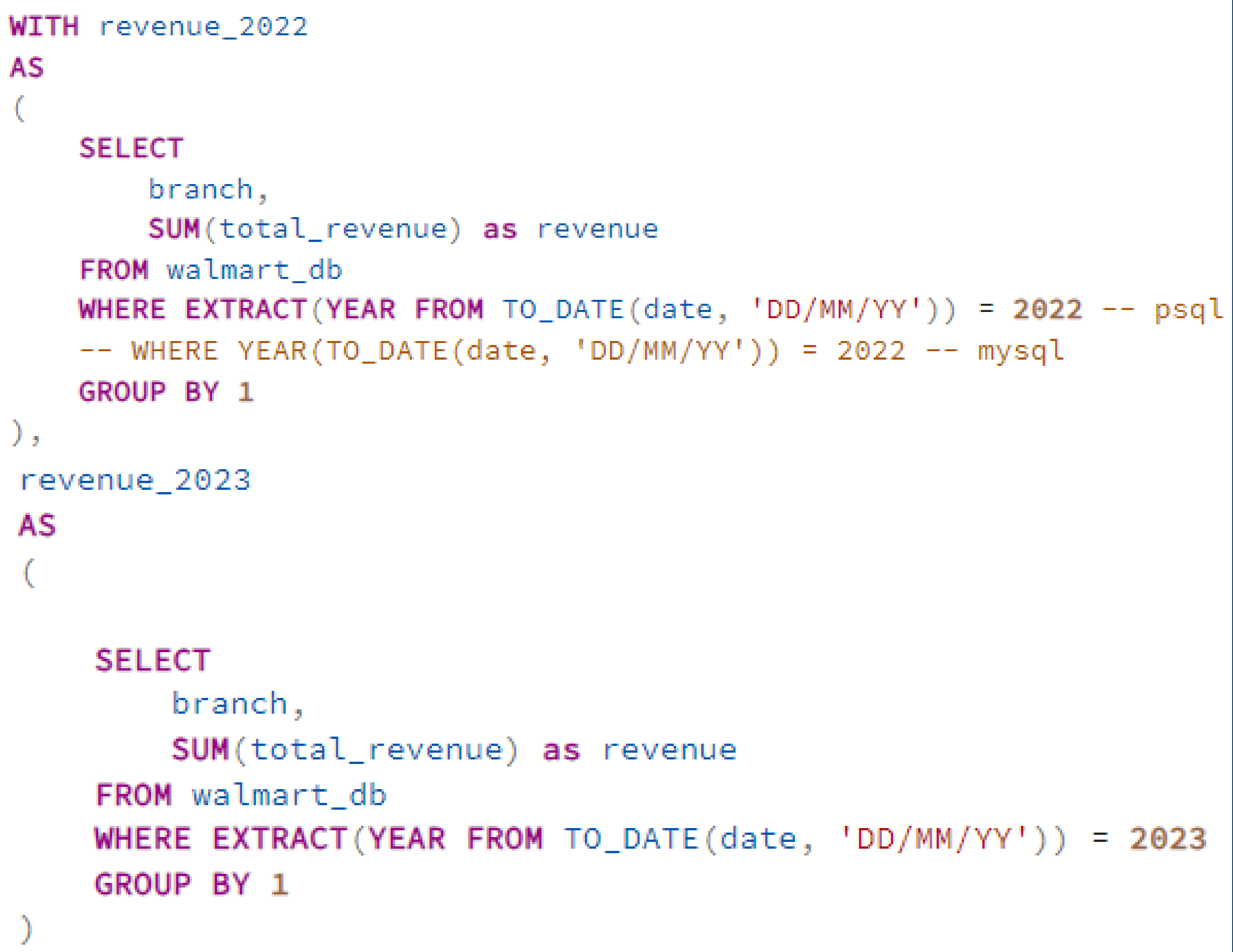


```
SELECT
       CASE
           WHEN EXTRACT(HOUR FROM (time::time)) < 12 THEN 'Morning'
           WHEN EXTRACT(HOUR FROM (time::time)) BETWEEN 12 AND 17 THEN 'Afternoon'
           ELSE 'Evening'
        END AS shift,
        COUNT(*) AS total_sales
FROM walmart_db
GROUP BY 1;
```

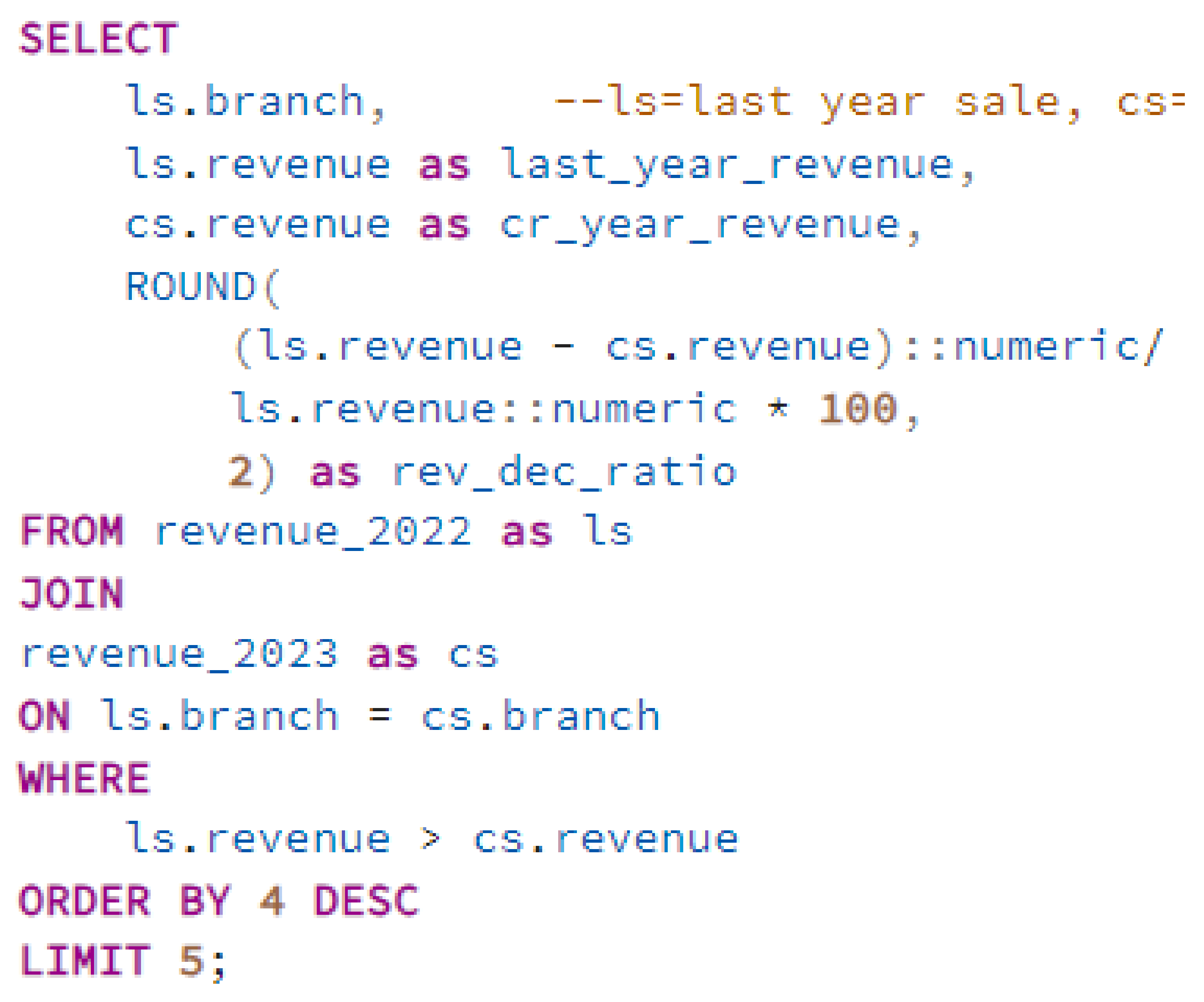


	shift text	total_sales bigint
1	Afternoon	4651
2	Evening	3256
3	Morning	2093

Q 9. IDENTIFY 5 BRANCH WITH HIGHEST DECREASE RATIO IN REVENUE COMPARE TO LAST YEAR(CURRENT YEAR 2023 AND LAST YEAR 2022).



```
WITH revenue_2022
AS
(
    SELECT
        branch,
        SUM(total_revenue) as revenue
    FROM walmart_db
    WHERE EXTRACT(YEAR FROM TO_DATE(date, 'DD/MM/YY')) = 2022 -- psql
    -- WHERE YEAR(TO_DATE(date, 'DD/MM/YY')) = 2022 -- mysql
    GROUP BY 1
),
 revenue_2023
 AS
 (

     SELECT
         branch,
         SUM(total_revenue) as revenue
     FROM walmart_db
     WHERE EXTRACT(YEAR FROM TO_DATE(date, 'DD/MM/YY')) = 2023
     GROUP BY 1
 )
```



```
SELECT
    ls.branch,      --ls=last year sale, cs=
    ls.revenue as last_year_revenue,
    cs.revenue as cr_year_revenue,
    ROUND(
        (ls.revenue - cs.revenue)::numeric/
        ls.revenue::numeric * 100,
        2) as rev_dec_ratio
FROM revenue_2022 as ls
JOIN
revenue_2023 as cs
ON ls.branch = cs.branch
WHERE
    ls.revenue > cs.revenue
ORDER BY 4 DESC
LIMIT 5;
```

THANK YOU

9795265562

github.com/kumar-nikhilnishad

kmrnikhil54@gmail.com

Gorakhpur, Uttar Pradesh