New Wheels Project

Introduction to SQL

Problem Statement

Business Context

A lot of people in the world share a common desire: to own a vehicle. A car or an automobile is seen as an object that gives the freedom of mobility. Many now prefer pre-owned vehicles because they come at an affordable cost, but at the same time, they are also concerned about whether the after-sales service provided by the resale vendors is as good as the care you may get from the actual manufacturers.

New-Wheels, a vehicle resale company, has launched an app with an end-to-end service from listing the vehicle on the platform to shipping it to the customer's location. This app also captures the overall after-sales feedback given by the customer.

Objective

New-Wheels sales have been dipping steadily in the past year, and due to the critical customer feedback and ratings online, there has been a drop in new customers every quarter, which is concerning to the business. The CEO of the company now wants a quarterly report with all the key metrics sent to him so he can assess the health of the business and make the necessary decisions.

As a data analyst, you see that there is an array of questions that are being asked at the leadership level that need to be answered using data. Import the dump file that contains various tables that are present in the database. Use the data to answer the questions posed and create a quarterly business report for the CEO.

Business Questions

Question 1: Find the total number of customers who have placed orders. What is the distribution of the customers across states?

Solution Query:

```
SELECT
        c.state,
         COUNT(DISTINCT c.customer_id) AS TOTAL_NUMBER_OF_CUSTOMERS_PER_STATE,
         t.TOTAL_NUMBER_OF_CUSTOMERS
FROM
        customer_t c
JOIN
         order_t o
         ON c.customer_id = o.customer_id
CROSS JOIN (
        SELECT COUNT(DISTINCT customer_id) AS TOTAL_NUMBER_OF_CUSTOMERS
        FROM order_t
) t
GROUP BY
        c.state, t.TOTAL_NUMBER_OF_CUSTOMERS
ORDER BY
        TOTAL_NUMBER_OF_CUSTOMERS_PER_STATE DESC;
```

Output:

Result: Passed

Query 1

Query:

```
SELECT
    c.state,
    COUNT(DISTINCT c.customer_id) AS TOTAL_NUMBER_OF_CUSTOMERS_PER_STATE,
    t.TOTAL_NUMBER_OF_CUSTOMERS
FROM
    customer_t c
JOIN
    order_t o
    ON c.customer_id = o.customer_id
CROSS JOIN (
    SELECT COUNT(DISTINCT customer_id) AS TOTAL_NUMBER_OF_CUSTOMERS
    FROM order_t
) t
GROUP BY
    c.state, t.TOTAL_NUMBER_OF_CUSTOMERS
ORDER BY
    TOTAL_NUMBER_OF_CUSTOMERS_PER_STATE DESC
```

Output:

Showing first 10 rows out of 49 rows

state	TOTAL_NUMBER_OF_C...	TOTAL_NUMBER_OF_C...
California	97	994
Texas	97	994
Florida	86	994
New York	69	994
District of Columbia	35	994
Colorado	33	994
Ohio	33	994
Alabama	29	994

Observations and Insights:

- States with bigger populations have a higher number of customers. California, Texas and Florida are the top three
- Maine, Vermont and Wyoming are the states with the least number of customers.
- There was a total of 994 unique customers in the last 4 quarters

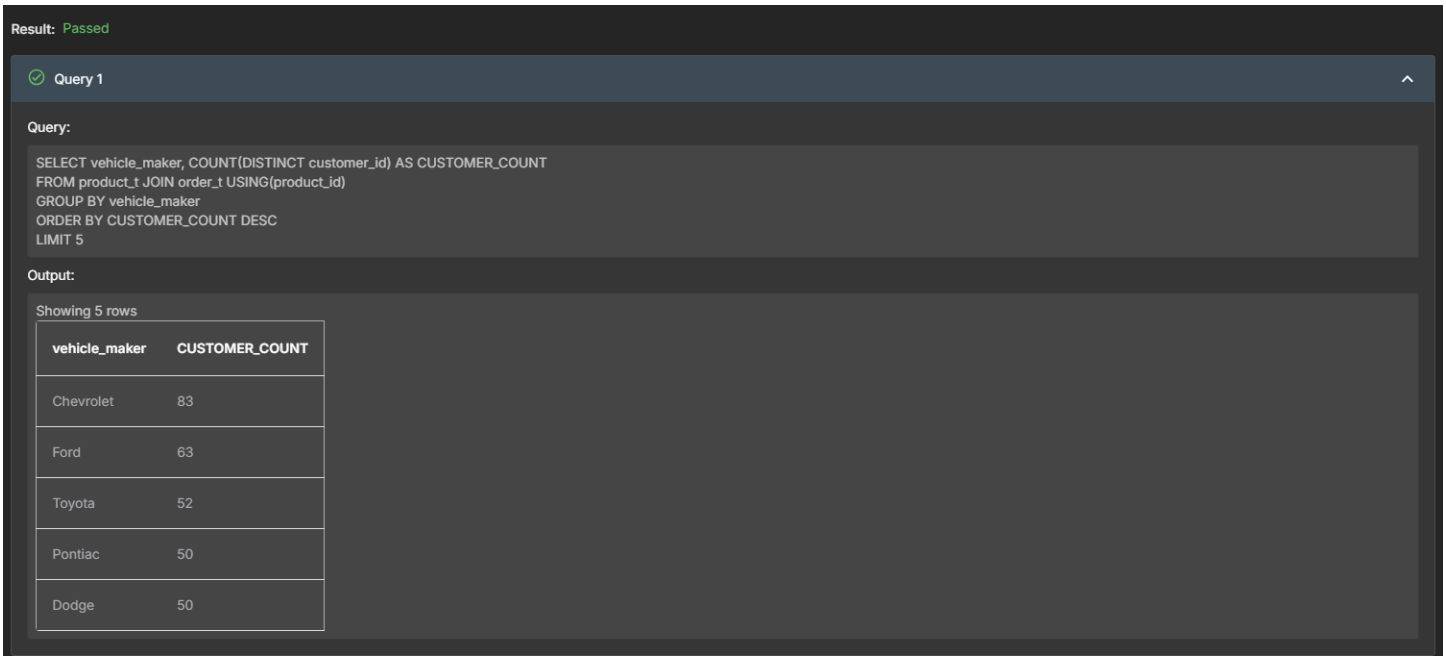
Question 2: Which are the top 5 vehicle makers preferred by the customers?

Solution Query:

```
SELECT
    vehicle_maker,
    COUNT(DISTINCT customer_id) AS CUSTOMER_COUNT
FROM
```

```
        product_t
JOIN
        order_t
        USING(product_id)
GROUP BY
        vehicle_maker
ORDER BY
        CUSTOMER_COUNT DESC
LIMIT 5;
```

Output:



Observations and Insights:

- Chevrolet, Ford, Toyota, Pontiac, and Dodge are the top 5
- The bottom 5 are MG, Daewoo, Austin, Ram, and Citron
- Chevrolet has a significant number of customers over the second highest Ford by 20 customers.

Question 3: Which is the most preferred vehicle maker in each state?

Solution Query:

```
SELECT
```

```
        state,
        vehicle_maker,
        RNK
FROM
        (SELECT
                state,
                vehicle_maker,
                COUNT(DISTINCT customer_id) AS CUSTOMER_COUNT,
                RANK() OVER(PARTITION BY state ORDER BY COUNT(DISTINCT customer_id) DESC) AS
        RNK
        FROM
                product_t
        JOIN order_t USING(product_id)
        JOIN customer_t USING(customer_id)
        GROUP BY
                state, vehicle_maker
        ) AS  RANKED
WHERE
        RNK = 1
ORDER BY
        state ASC;
```

Output:

Result: Passed

Query 1

Query:

```
SELECT state, vehicle_maker, RNK
FROM (SELECT state, vehicle_maker, COUNT(DISTINCT customer_id) AS CUSTOMER_COUNT, RANK() OVER(PARTITION BY state ORDER BY COUNT(DISTINCT customer_id) DESC) AS RNK
      FROM product_t JOIN order_t USING(product_id) JOIN customer_t USING(customer_id)
   GROUP BY state, vehicle_maker) AS  RANKED
WHERE RNK = 1
ORDER BY state ASC
```

Output:

Showing first 10 rows out of 143 rows

state	vehicle_maker	RNK
Alabama	Dodge	1
Alaska	Chevrolet	1
Arizona	Pontiac	1
Arizona	Cadillac	1
Arkansas	Volkswagen	1
Arkansas	Suzuki	1
Arkansas	Pontiac	1
Arkansas	Mitsubishi	1

Observations and Insights:

- Texas favorite vehicle maker is chevrolet, California is Audi then chevrolet, Florida is Toyota
- Wyoming is Buick, Vermont is Mazda, Maine is Mercedez-Benz
- Chevrolet appeared 17 times as a state's favorite vehicle maker, followed by Dodge at 12 and then Pontiac at 11.

Question 4: Find the overall average rating given by the customers. What is the average rating in each quarter?

Consider the following mapping for ratings: "Very Bad": 1, "Bad": 2, "Okay": 3, "Good": 4, "Very Good": 5

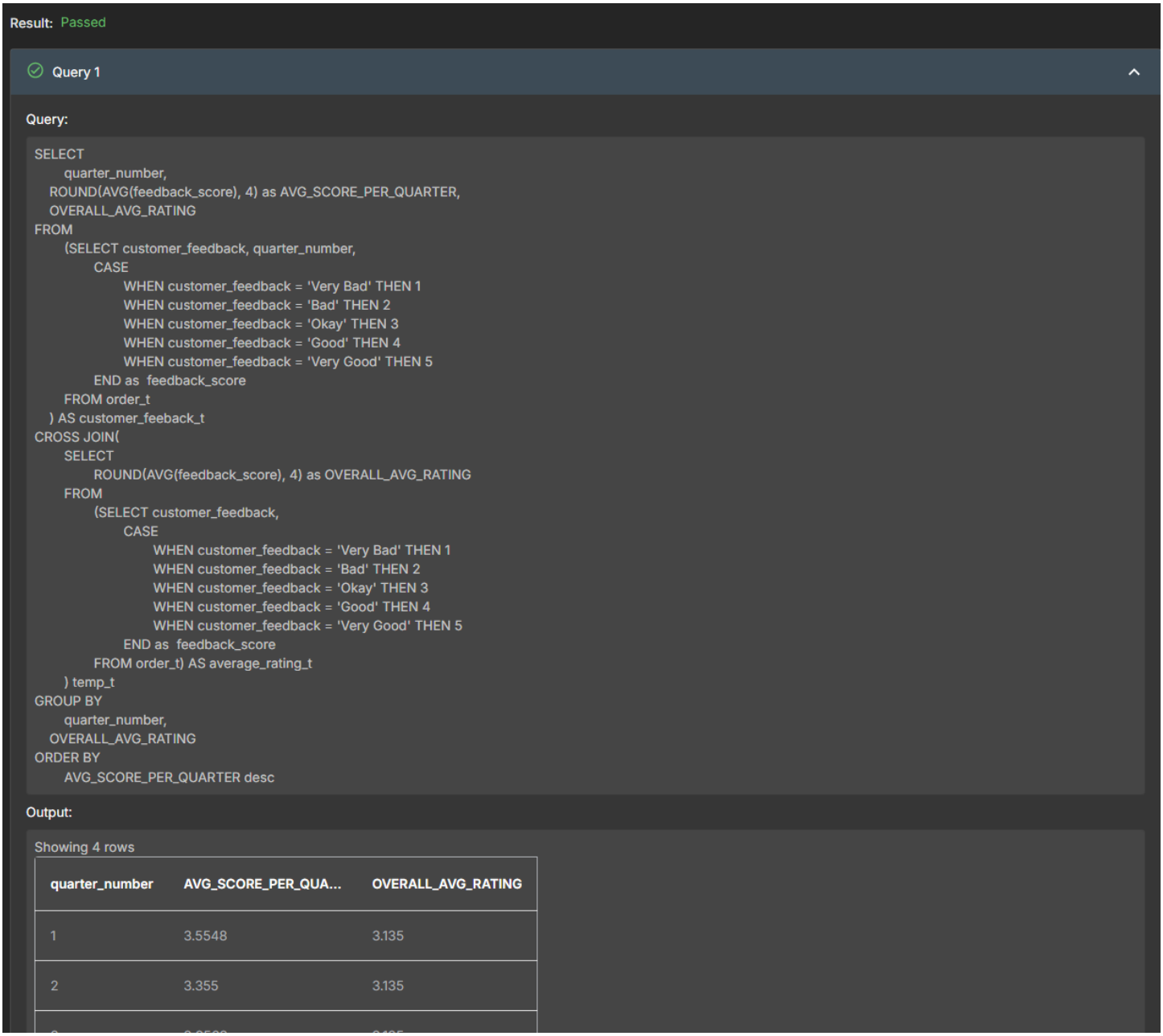
Solution Query:

```
SELECT
        quarter_number,
   ROUND(AVG(feedback_score), 4) as AVG_SCORE_PER_QUARTER,
```

```
    OVERALL_AVG_RATING
FROM
        (SELECT customer_feedback, quarter_number,
                CASE
                        WHEN customer_feedback = 'Very Bad' THEN 1
                        WHEN customer_feedback = 'Bad' THEN 2
                        WHEN customer_feedback = 'Okay' THEN 3
                        WHEN customer_feedback = 'Good' THEN 4
                        WHEN customer_feedback = 'Very Good' THEN 5
                END as  feedback_score
        FROM order_t
        ) AS customer_feeback_t
CROSS JOIN(
        SELECT
                ROUND(AVG(feedback_score), 4) as OVERALL_AVG_RATING
        FROM
                (SELECT customer_feedback,
                        CASE
                                WHEN customer_feedback = 'Very Bad' THEN 1
                                WHEN customer_feedback = 'Bad' THEN 2
                                WHEN customer_feedback = 'Okay' THEN 3
                                WHEN customer_feedback = 'Good' THEN 4
                                WHEN customer_feedback = 'Very Good' THEN 5
                        END as  feedback_score
                FROM order_t) AS average_rating_t
        ) temp_t
```

```
GROUP BY
        quarter_number,
    OVERALL_AVG_RATING
ORDER BY
        AVG_SCORE_PER_QUARTER desc;
```

Output:



Result: Passed

Query 1

Query:

```
SELECT
      quarter_number,
   ROUND(AVG(feedback_score), 4) as AVG_SCORE_PER_QUARTER,
   OVERALL_AVG_RATING
FROM
      (SELECT customer_feedback, quarter_number,
            CASE
                  WHEN customer_feedback = 'Very Bad' THEN 1
                  WHEN customer_feedback = 'Bad' THEN 2
                  WHEN customer_feedback = 'Okay' THEN 3
                  WHEN customer_feedback = 'Good' THEN 4
                  WHEN customer_feedback = 'Very Good' THEN 5
            END as  feedback_score
      FROM order_t
   ) AS customer_feeback_t
CROSS JOIN(
      SELECT
            ROUND(AVG(feedback_score), 4) as OVERALL_AVG_RATING
      FROM
            (SELECT customer_feedback,
                  CASE
                        WHEN customer_feedback = 'Very Bad' THEN 1
                        WHEN customer_feedback = 'Bad' THEN 2
                        WHEN customer_feedback = 'Okay' THEN 3
                        WHEN customer_feedback = 'Good' THEN 4
                        WHEN customer_feedback = 'Very Good' THEN 5
                  END as  feedback_score
            FROM order_t) AS average_rating_t
      ) temp_t
GROUP BY
      quarter_number,
   OVERALL_AVG_RATING
ORDER BY
      AVG_SCORE_PER_QUARTER desc
```

Output:

Showing 4 rows

quarter_number	AVG_SCORE_PER_QUA...	OVERALL_AVG_RATING
1	3.5548	3.135
2	3.355	3.135

Observations and Insights:

- The overall average rating for the year is 3.135 out of 4.

- The first two quarters start with a rating higher than 3 but then fall below 3 for the last 2 quarters
- Quarter four falls down all the way to 2.3970

Question 5: Find the percentage distribution of feedback from the customers. Are customers getting more dissatisfied over time?

Solution Query:

```
SELECT

        quarter_number,

    CONCAT(ROUND(100 * SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) /
COUNT(*), 2) , '%') AS VERY_GOOD_PCT,

    CONCAT(ROUND(100 * SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) /
COUNT(*), 2) , '%') AS GOOD_PCT,

    CONCAT(ROUND(100 * SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) /
COUNT(*), 2) , '%') AS OKAY_PCT,

     CONCAT(ROUND(100 * SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) /
COUNT(*), 2) , '%') AS BAD_PCT,

    CONCAT(ROUND(100 * SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) /
COUNT(*), 2) , '%') AS VERY_BAD_PCT

FROM

        order_t

GROUP BY

        quarter_number

ORDER BY

        quarter_number ASC;
```

Output:

Result: Passed

Query 1

Query:

```
SELECT
    quarter_number,
    ROUND(100.0 * SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) / COUNT(*), 2) || '%' AS VERY_GOOD_PCT,
    ROUND(100.0 * SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) / COUNT(*), 2) || '%' AS GOOD_PCT,
    ROUND(100.0 * SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) / COUNT(*), 2) || '%' AS OKAY_PCT,
    ROUND(100.0 * SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) / COUNT(*), 2) || '%' AS BAD_PCT,
    ROUND(100.0 * SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) / COUNT(*), 2) || '%' AS VERY_BAD_PCT
FROM order_t
GROUP BY quarter_number
ORDER BY quarter_number ASC
```

Output:

Showing 4 rows

quarter_number	VERY_GOOD_PCT	GOOD_PCT	OKAY_PCT	BAD_PCT	VERY_BAD_PCT
1	30.0%	28.71%	19.03%	11.29%	10.97%
2	28.63%	22.14%	20.23%	14.12%	14.89%
3	16.59%	20.96%	21.83%	22.71%	17.9%
4	10.05%	10.05%	20.1%	29.15%	30.65%

Observations and Insights:

- Both quarters one and quarter two have similar Very Good ratings, but starting in quarter 2, the bad and very bad pct starts to go up, and the good pct in quarter 2 is 6% lower than quarter 1.
- The okay pct stays consistent in all 4 quarters.
- By quarter 4, both the bad and very bad pct are around 30% each. The very good and good pct drop off significantly and are at 10% each.

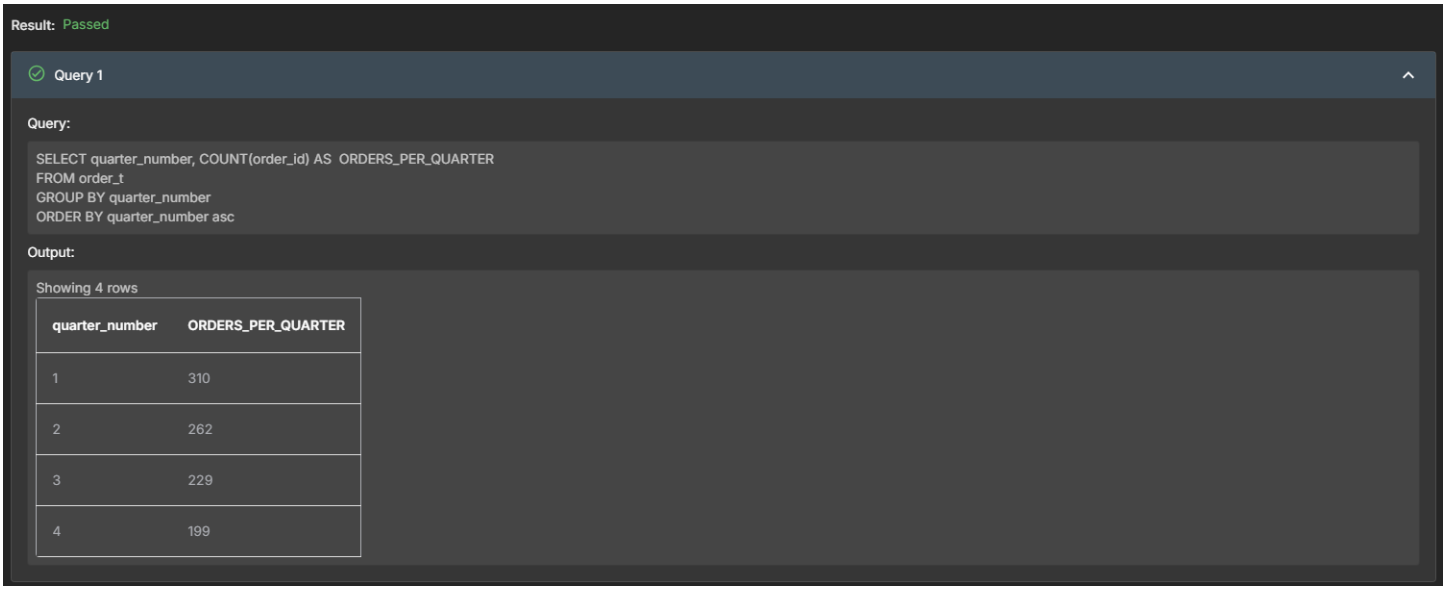
Question 6: What is the trend of the number of orders by quarter?

Solution Query:

```
SELECT
        quarter_number,
        COUNT(order_id) AS  ORDERS_PER_QUARTER
FROM
        order_t
GROUP BY
        quarter_number
ORDER BY
```

quarter_number asc;

Output:



Observations and Insights:

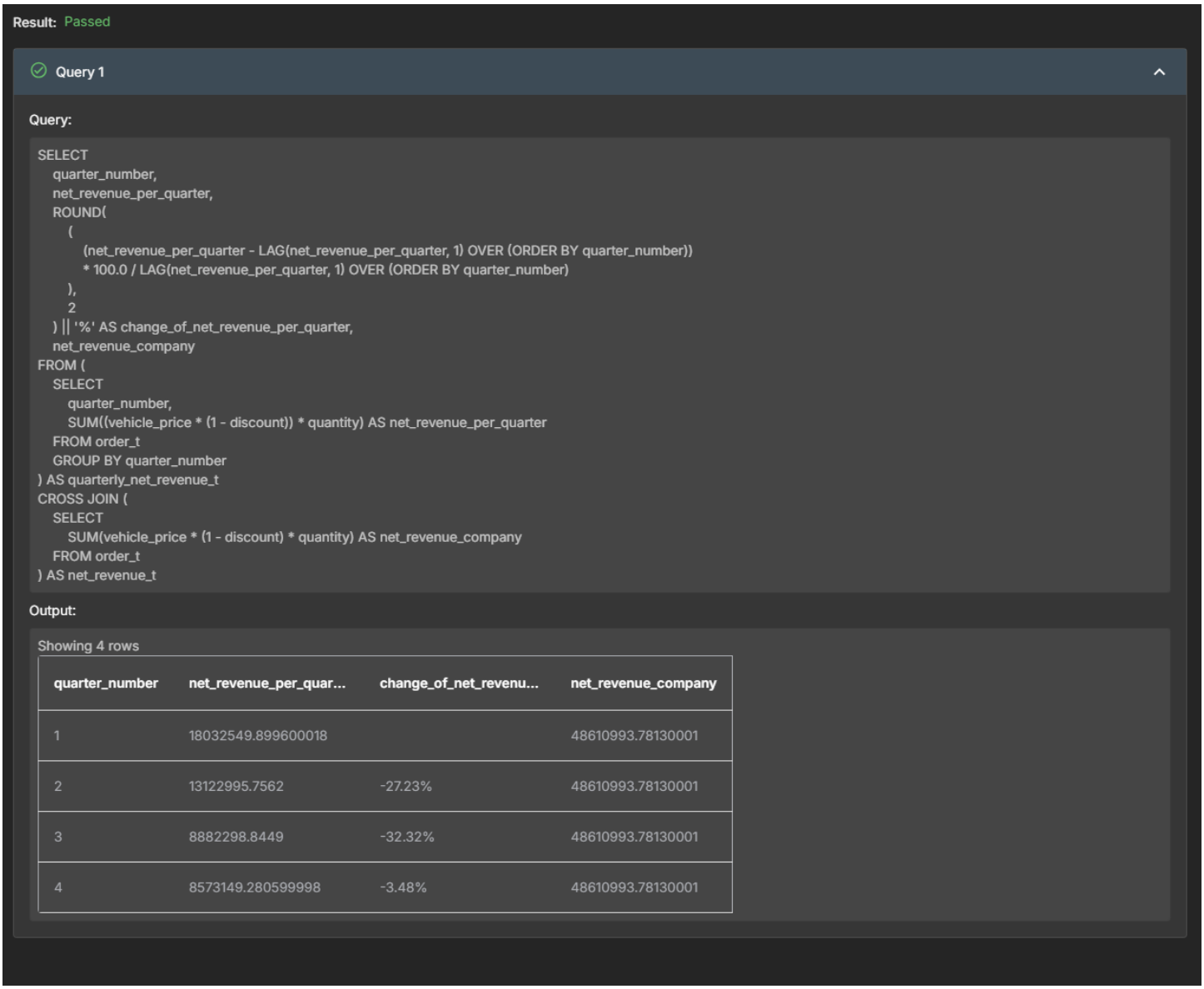
- The number of orders goes down each quarter.
- Quarter 3 had 111 less orders than quarter 1
- Each quarter had more dissatisfied customers, as well as less orders

Question 7: Calculate the net revenue generated by the company. What is the quarter-over-quarter % change in net revenue?

Solution Query:

```
SELECT
	quarter_number,
	net_revenue_per_quarter,
	CONCAT(ROUND((((net_revenue_per_quarter - LAG(net_revenue_per_quarter,1) OVER(ORDER BY quarter_number)) / LAG(net_revenue_per_quarter,1) OVER(ORDER BY quarter_number)) * 100),2), '%') AS change_of_net_revenue_per_quarter,
	net_revenue_company
FROM
	(
	SELECT
		quarter_number,
		SUM((o.vehicle_price * (1 - discount)) * quantity) AS net_revenue_per_quarter
	FROM
		order_t o
	GROUP BY
		quarter_number
	) AS quarterly_net_revenue_t
CROSS JOIN
	(
	SELECT
		SUM(vehicle_price * (1 - discount) * quantity) AS net_revenue_company
	FROM
		order_t
	) AS net_revenue_t;
```

Output:



Observations and Insights:

- From quarter 1 to quarter 2 there is a 27% drop.
- Quarters 2 and 3 have significant drops in revenue from the previous quarters.
- Quarter 3 and 4 had a similar revenue

Question 8: What is the trend of net revenue and orders by quarters?

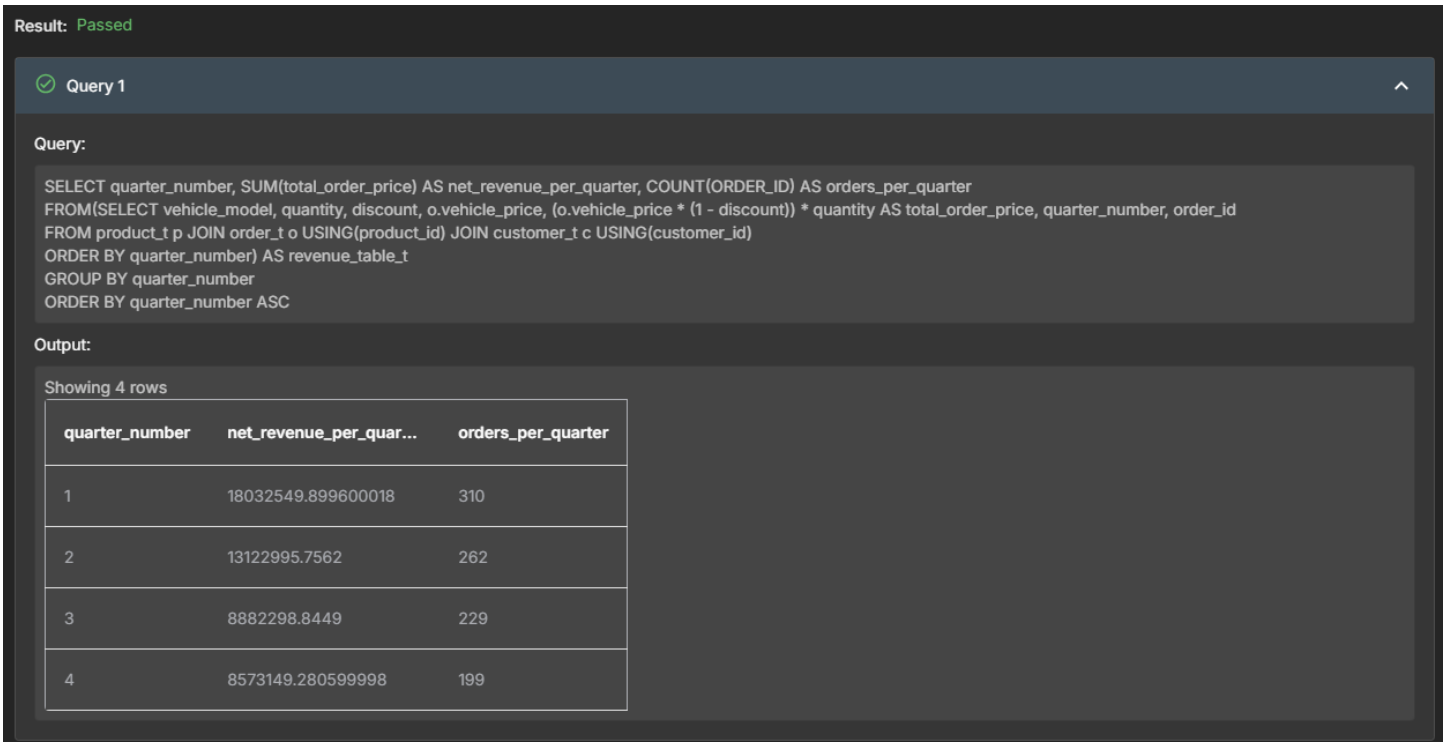
Solution Query:

```
SELECT

    quarter_number,
```

```
    SUM((o.vehicle_price * (1 - discount)) * quantity) AS net_revenue_per_quarter,
    COUNT(ORDER_ID) AS orders_per_quarter
FROM
    order_t o
GROUP BY
    quarter_number
ORDER BY
    quarter_number ASC;
```

Output:



Observations and Insights:

- Quarter 1,2, 3 each had around 40 orders less than the previous. They also had a similar drop in revenue.
- Quarter 3 to 4 had a 30-order drop, but the revenue did not fall that much.
- Possible reasons that the revenue is falling are due to the type of vehicles being sold, as well as the falling approval ratings

Question 9: What is the average discount offered for different types of credit cards?

Solution Query:

```
SELECT
        credit_card_type,
   CONCAT(ROUND(AVG(discount), 2), '%') as average_discount_per_credit_card_type
FROM
        order_t
JOIN
        customer_t USING(customer_id)
GROUP BY
        credit_card_type
ORDER BY
        average_discount_per_credit_card_type DESC;
```

Note: The '`discount`' field in the '`order_t`' table is stored as a percentage, i.e. `0.6` represents a discount of `0.6%`, not `60%`.

Output:

Result: Passed

Query 1

Query:

```
SELECT
    credit_card_type,
    ROUND(AVG(discount), 2) || '%' AS average_discount_per_credit_card_type
FROM order_t
JOIN customer_t USING(customer_id)
GROUP BY credit_card_type
ORDER BY AVG(discount) DESC
```

Output:

Showing first 10 rows out of 16 rows

credit_card_type	average_discount_per...
laser	0.64%
mastercard	0.63%
maestro	0.62%
visa-electron	0.62%
china-unionpay	0.62%
instapayment	0.62%
americanexpress	0.62%
diners-club-us-ca	0.61%

Observations and Insights:

- The highest discount on average is 0.64 for laser
- The lowest discount on average is 0.58 for diners-club-international

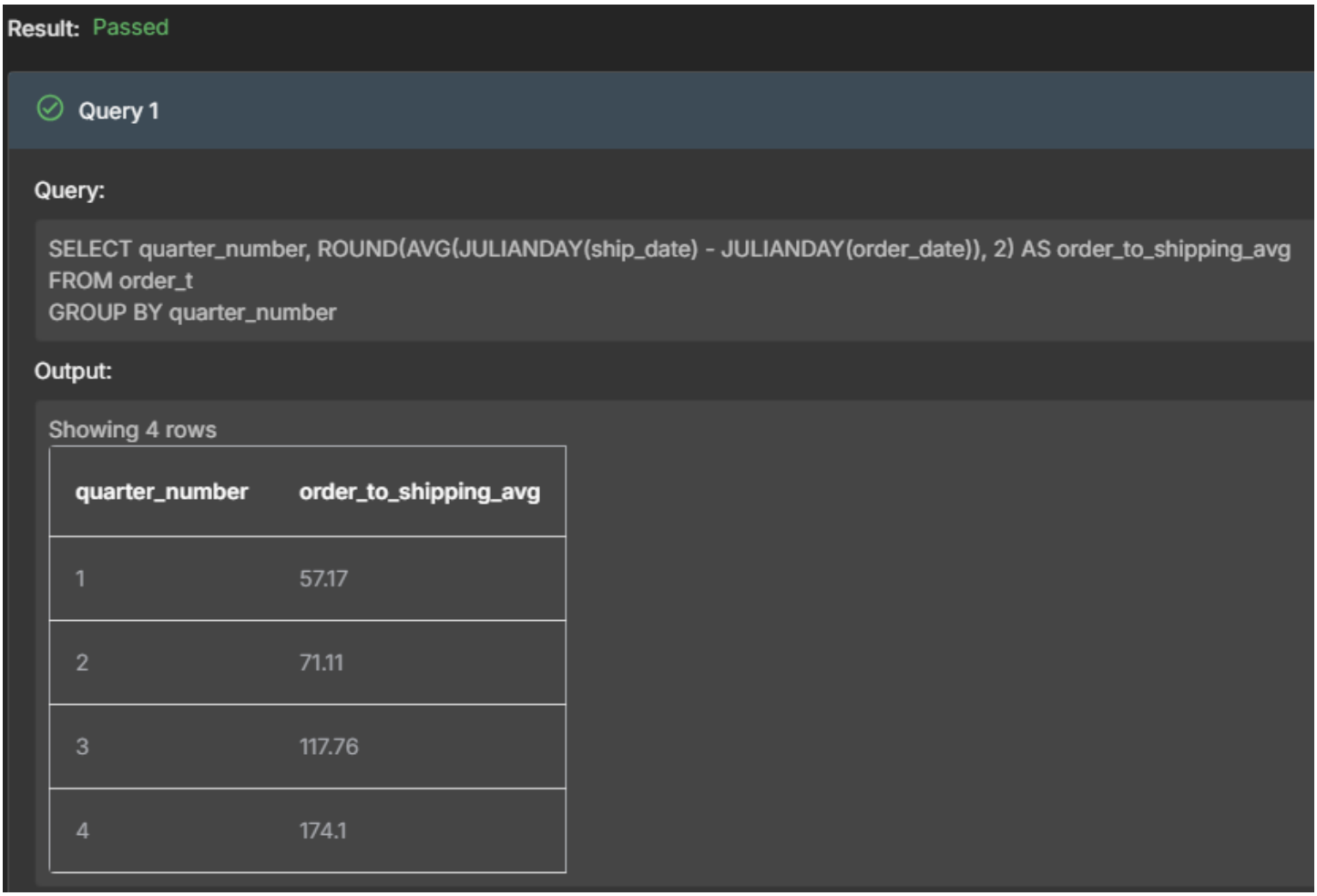
- Discount probably does not affect the sales of the company that much.

Question 10: What is the average time taken to ship the placed orders for each quarter?

Solution Query:

```
SELECT
    quarter_number,
    ROUND(AVG(JULIANDAY(ship_date) - JULIANDAY(order_date)), 2) AS order_to_shipping_avg
FROM
    order_t
GROUP BY
    quarter_number;
```

Output:



Observations and Insights:

- The first quarter had relatively low shipping times with 57.17 days on average
- Each quarter the time to ship increased
- By quarter 4, the average days to ship was 174.1. Shipping time increasing coincides with lower reviews. Could have a possible correlation.

Business Metrics Overview

Total Revenue	Total Orders	Total Customers	Average Rating
$48610993.78	1,000	994	3.135
Last Quarter Revenue	**Last quarter Orders**	**Average Days to Ship**	**% Good Feedback**
$8573149.28	199	97.96	21.50%

Note: These values must be derived using SQL queries. Some of them may have already been obtained while answering previous questions.

Business Recommendations

- New Wheels needs to focus on getting the average rating up. The ratings went down as the year progressed, meaning that there is a problem with the services the company is offering. By the last quarter, only 199 orders were made, and only 20% of ratings were positive. 60% were negative with 20% saying the order was ok.
- Revenue falling coincided with the falling of ratings. Each quarter, the company's revenue seemed to have a correlation with orders made, reviews, and average days to ship. These are all key metrics that the company should focus on when making their assessment.
- The average number of days to ship increased each quarter. The increase in average days to ship seem to correlate with the lowering of ratings

- My last suggestion is to increase the number of vehicles from popular brands such as Chevrolet, Ford, Toyota, Dodge and Pontiac. While I don't have data on the supply of these vehicles, it is possible that a lack of supply could also be a reason that orders went down. Further analysis would be needed to be conducted for this conclusion.