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

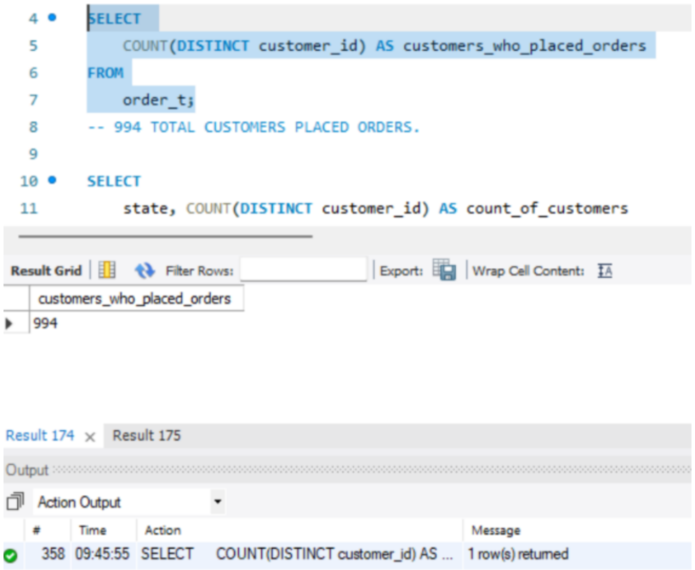
Question 1: Find the total number of customers who have placed orders. What is the distribution of the customers across states?

Solution Query:

```
SELECT
    COUNT(DISTINCT customer_id) AS customers_who_placed_orders
FROM
    order_t;
-- 994 TOTAL CUSTOMERS PLACED ORDERS.

SELECT
    state, COUNT(DISTINCT customer_id) AS count_of_customers
FROM
    customer_t
GROUP BY state
ORDER BY count_of_customers DESC;
```

Output:



```
SELECT
    COUNT(DISTINCT customer_id) AS customers_who_placed_orders
FROM
    order_t;
-- 994 TOTAL CUSTOMERS PLACED ORDERS.

SELECT
    state, COUNT(DISTINCT customer_id) AS count_of_customers
```

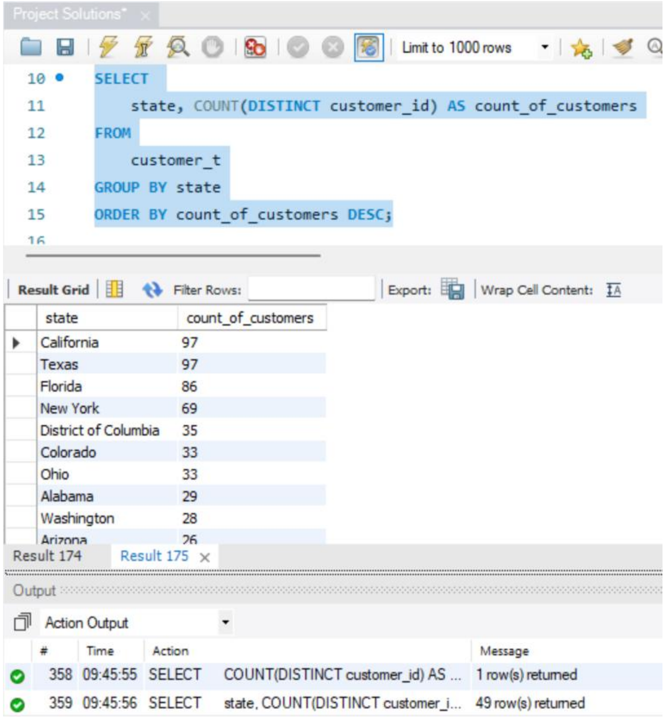
customers_who_placed_orders
994

Result 174 Result 175

Output

Action Output

#	Time	Action		Message
358	09:45:55	SELECT	COUNT(DISTINCT customer_id) AS ...	1 row(s) returned



Project Solutions*

Limit to 1000 rows

```
SELECT
    state, COUNT(DISTINCT customer_id) AS count_of_customers
FROM
    customer_t
GROUP BY state
ORDER BY count_of_customers DESC;
```

state	count_of_customers
California	97
Texas	97
Florida	86
New York	69
District of Columbia	35
Colorado	33
Ohio	33
Alabama	29
Washington	28
Arizona	26

Result 174 Result 175

Output

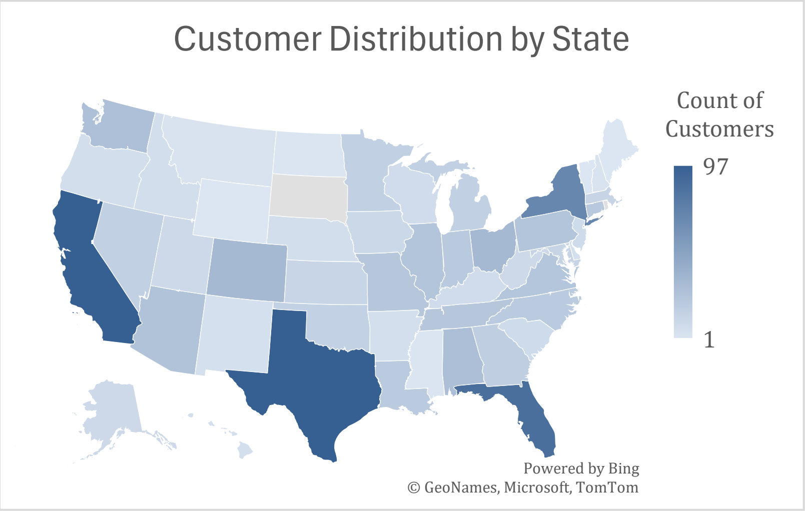
Action Output

#	Time	Action		Message
358	09:45:55	SELECT	COUNT(DISTINCT customer_id) AS ...	1 row(s) returned
359	09:45:56	SELECT	state, COUNT(DISTINCT customer_i...	49 row(s) returned

Observations and Insights:

- 994 total customers placed orders.
- Most of the customers come from California, Texas, Florida, and New York with a total of 69-97 customers per state.
- Followed by DC, CO, OH, AL, WA, AZ, IL, PN, VA, MI, TN, CN, IN, NC, and LA all having 20+ customers.

Customer Distribution by State

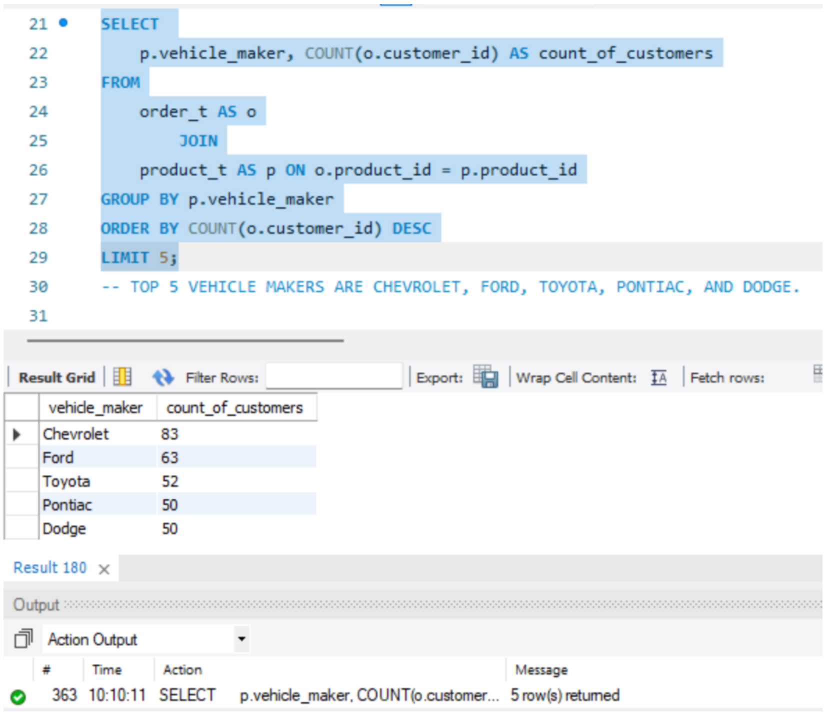


Question 2: Which are the top 5 vehicle makers preferred by the customers?

Solution Query:

```
SELECT
    p.vehicle_maker, COUNT(o.customer_id) AS count_of_customers
FROM
    order_t AS o
        JOIN
    product_t AS p ON o.product_id = p.product_id
GROUP BY p.vehicle_maker
ORDER BY COUNT(o.customer_id) DESC
LIMIT 5;
```

Output:



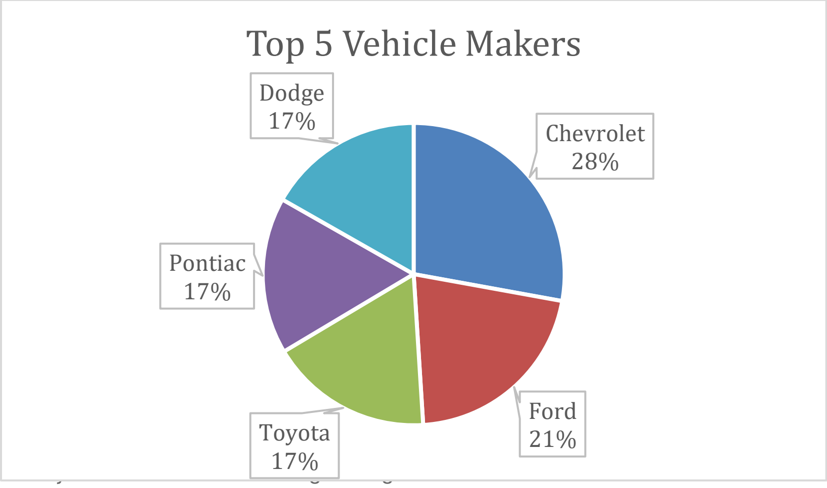
```
SELECT
    p.vehicle_maker, COUNT(o.customer_id) AS count_of_customers
FROM
    order_t AS o
        JOIN
    product_t AS p ON o.product_id = p.product_id
GROUP BY p.vehicle_maker
ORDER BY COUNT(o.customer_id) DESC
LIMIT 5;
-- TOP 5 VEHICLE MAKERS ARE CHEVROLET, FORD, TOYOTA, PONTIAC, AND DODGE.
```

vehicle_maker	count_of_customers
Chevrolet	83
Ford	63
Toyota	52
Pontiac	50
Dodge	50

```
#    Time      Action                                          Message
363  10:10:11  SELECT  p.vehicle_maker, COUNT(o.customer...   5 row(s) returned
```

Observations and Insights:

- The top 5 vehicle makers preferred by the customers are Chevrolet, Ford, Toyota, Pontiac, and Dodge.

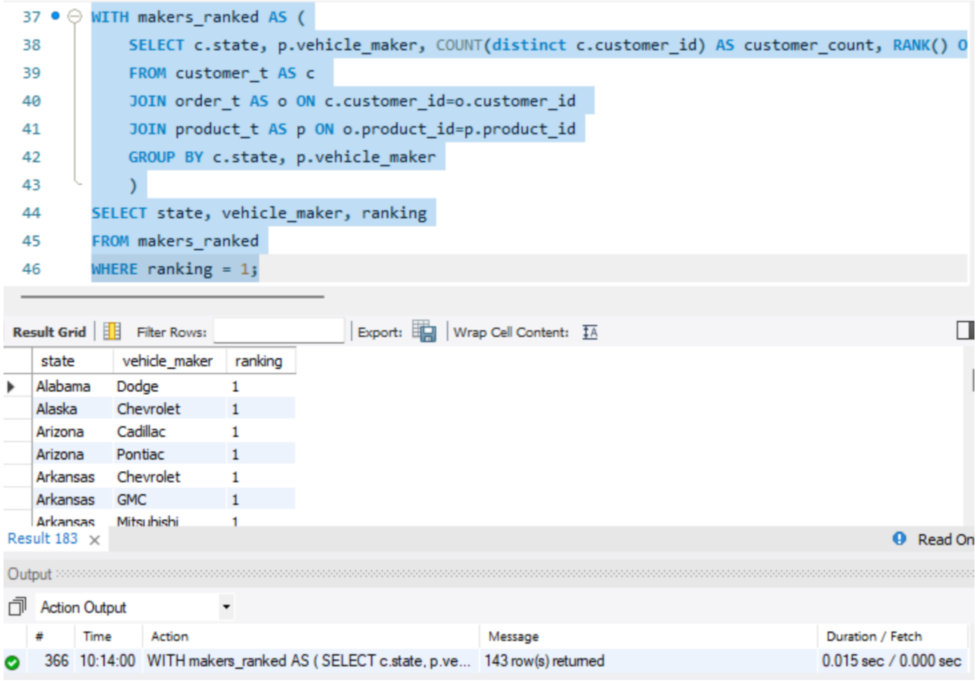


Question 3: Which is the most preferred vehicle maker in each state?

Solution Query:

```
WITH makers_ranked AS (
          SELECT c.state, p.vehicle_maker, COUNT(distinct c.customer_id) AS customer_count, RANK() OVER
(PARTITION BY c.state ORDER BY COUNT(distinct c.customer_id) DESC) AS ranking
          FROM customer_t AS c
          JOIN order_t AS o ON c.customer_id=o.customer_id
          JOIN product_t AS p ON o.product_id=p.product_id
          GROUP BY c.state, p.vehicle_maker
   )
SELECT state, vehicle_maker, ranking
FROM makers_ranked
WHERE ranking = 1;
```

Output:



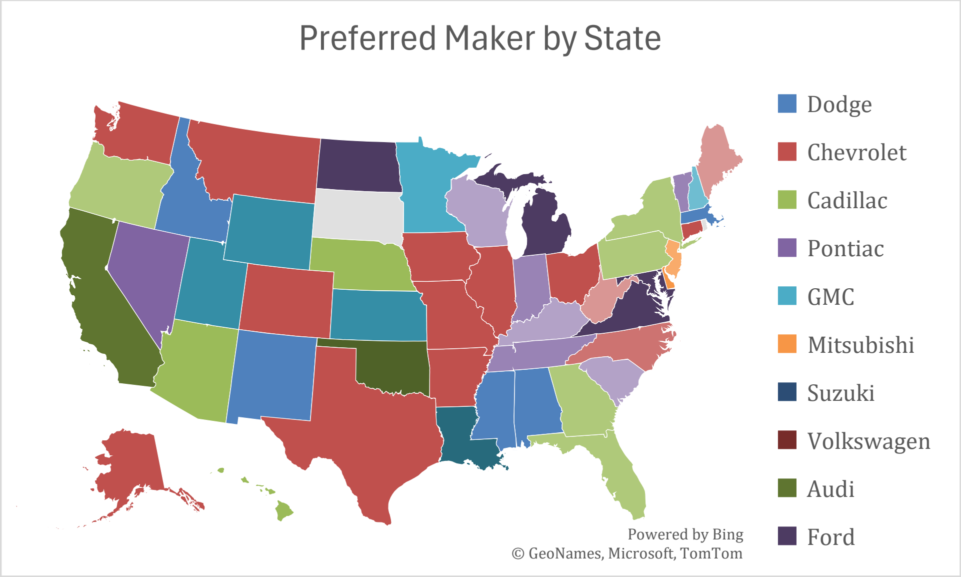
```
WITH makers_ranked AS (
    SELECT c.state, p.vehicle_maker, COUNT(distinct c.customer_id) AS customer_count, RANK() O
    FROM customer_t AS c
    JOIN order_t AS o ON c.customer_id=o.customer_id
    JOIN product_t AS p ON o.product_id=p.product_id
    GROUP BY c.state, p.vehicle_maker
    )
SELECT state, vehicle_maker, ranking
FROM makers_ranked
WHERE ranking = 1;
```

state	vehicle_maker	ranking
Alabama	Dodge	1
Alaska	Chevrolet	1
Arizona	Cadillac	1
Arizona	Pontiac	1
Arkansas	Chevrolet	1
Arkansas	GMC	1
Arkansas	Mitsubishi	1

#	Time	Action	Message	Duration / Fetch
366	10:14:00	WITH makers_ranked AS (SELECT c.state, p.ve...	143 row(s) returned	0.015 sec / 0.000 sec

Observations and Insights:

The chart below demonstrates preferred vehicle maker by state:



There are states, however, where the preferred maker ranked #1 with another. These are listed below:

Arizona- Cadillac, Pontiac

Arkansas- Chevrolet, GMC, Mitsubishi, Pontiac, Suzuki, Volkswagen

California- Audi, Chevrolet, Dodge, Ford, Nissan

Connecticut- Chevrolet, Maserati, Mercury, Volvo

Hawaii- Cadillac, Ford, GMC, Nissan, Pontiac, Toyota

Illinois- Chevrolet, GMC, Ford; **Iowa**- Chevrolet, Chrysler, Dodge, Ford, Hyundai, Isuzu, Jeep, Mazda, Pontiac, Porsche, Subaru

Kansas- Buick, Dodge, Ford, GMC, Honda, Lexus, Maserati, Mazda, Mercedes-Benz, Nissan, Saab, Suzuki, Volkswagen

Kentucky- Acura, Audi, Mercedes-Benz, Mercury, Nissan, Pontiac, Ram, Volvo

Louisiana- Nissan, BMW, Ford, Pontiac, Kia

Massachusetts- Dodge, Chevrolet, Dodge, Toyota

Montana- Chevrolet, Dodge, Mitsubishi

Nebraska- Cadillac, Chevrolet, Mercedes-Benz, Nissan, Pontiac, Toyota, Volkswagen

New Hampshire- Chrysler, Lexus, Lincoln

New Jersey- Hyundai, Mercedes-Benz

New York- Toyota, Pontiac

North Dakota- Ford, Hyundai

Oklahoma- Ferrari, Mazda, Toyota

South Carolina- Acura, BMW, Buick, Dodge, Isuzu, Jaguar, Kia, Mazda, Mitsubishi

Utah- Buick, Chevrolet, Dodge, Isuzu, Lincoln, Maybach, Oldsmobile, Pontiac, Subaru, Volkswagen

Wisconsin- Acura, Cadillac, Chevrolet, Dodge, Honda, Mazda, Nissan, Pontiac

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

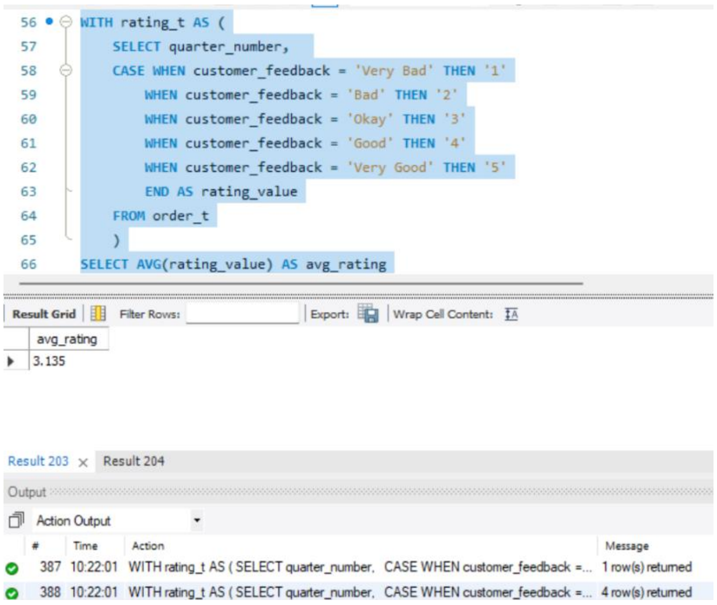
Consider the following mapping for ratings: “Very Bad”: 1, “Bad”: 2, “Okay”: 3, “Good”: 4, “Very Good”: 5

Solution Query:

```
WITH rating_t AS (
        SELECT 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 rating_value
        FROM order_t
   )
SELECT AVG(rating_value) AS avg_rating
FROM rating_t
;
-- Average rating overall: 3.135

WITH rating_t AS (
        SELECT 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 rating_value
        FROM order_t
   )
SELECT quarter_number, AVG(rating_value) AS avg_rating
FROM rating_t
GROUP BY quarter_number
ORDER BY quarter_number ASC;
```

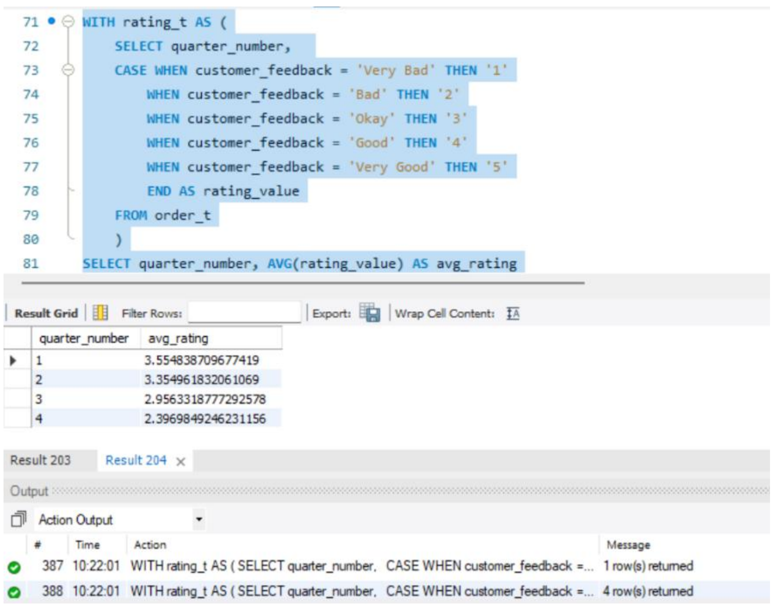
Output:



```sql
WITH rating_t AS (
    SELECT 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 rating_value
    FROM order_t
    )
SELECT AVG(rating_value) AS avg_rating
```

avg_rating
3.135

```
#    Time      Action                                                                 Message
387  10:22:01  WITH rating_t AS ( SELECT quarter_number,  CASE WHEN customer_feedback =...  1 row(s) returned
388  10:22:01  WITH rating_t AS ( SELECT quarter_number,  CASE WHEN customer_feedback =...  4 row(s) returned
```



```sql
WITH rating_t AS (
    SELECT 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 rating_value
    FROM order_t
    )
SELECT quarter_number, AVG(rating_value) AS avg_rating
```

quarter_number	avg_rating
1	3.554838709677419
2	3.354961832061069
3	2.9563318777292578
4	2.3969849246231156

```
#    Time      Action                                                                 Message
387  10:22:01  WITH rating_t AS ( SELECT quarter_number,  CASE WHEN customer_feedback =...  1 row(s) returned
388  10:22:01  WITH rating_t AS ( SELECT quarter_number,  CASE WHEN customer_feedback =...  4 row(s) returned
```

Observations and Insights:

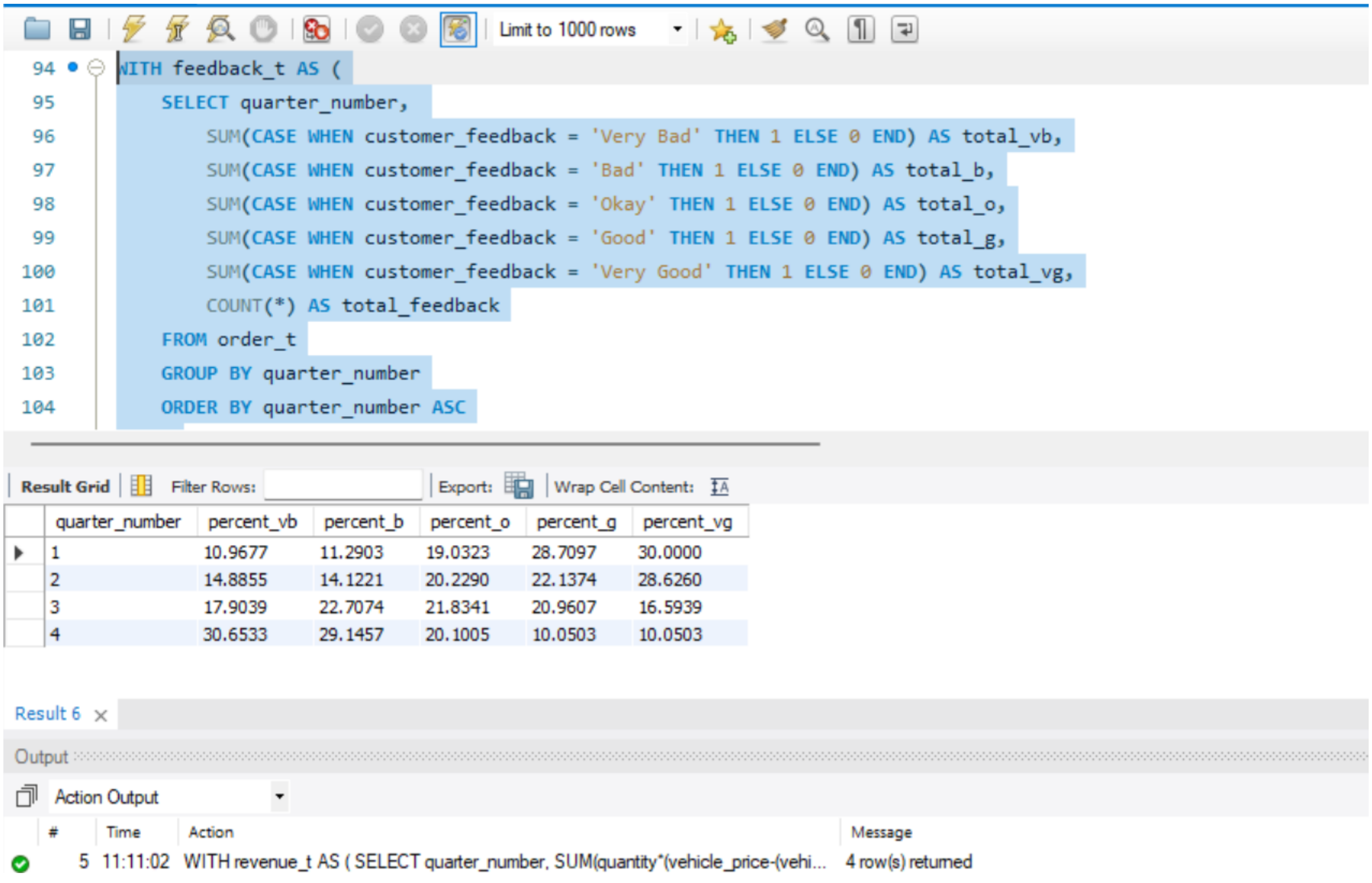
- The overall rating given by the customers for the year is 3.135 out of 5.
- The average rating per quarter is as follows: Quarter 1: 3.555, Quarter 2: 3.355, Quarter 3: 2.956 Quarter 4: 2.397
- The average rating fell from Q1 to Q2 by 0.2 out of 5, Q2 to Q3 fell 0.399 out of 5, and Q3 to Q4 fell by 0.559 out of 5, meaning the rate of declining customer ratings is falling faster over time.

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

Solution Query:

```
WITH feedback_t AS (
        SELECT quarter_number,
                SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) AS total_vb,
                SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) AS total_b,
                SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) AS total_o,
                SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) AS total_g,
                SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) AS total_vg,
                COUNT(*) AS total_feedback
        FROM order_t
        GROUP BY quarter_number
        ORDER BY quarter_number ASC
        )
SELECT quarter_number, (total_vb/total_feedback)*100 AS percent_vb, (total_b/total_feedback)*100 AS
percent_b, (total_o/total_feedback)*100 AS percent_o, (total_g/total_feedback)*100 AS percent_g,
(total_vg/total_feedback)*100 AS percent_vg
FROM feedback_t;
```

Output:



```
WITH feedback_t AS (
    SELECT quarter_number,
        SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) AS total_vb,
        SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) AS total_b,
        SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) AS total_o,
        SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) AS total_g,
        SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) AS total_vg,
        COUNT(*) AS total_feedback
    FROM order_t
    GROUP BY quarter_number
    ORDER BY quarter_number ASC
```

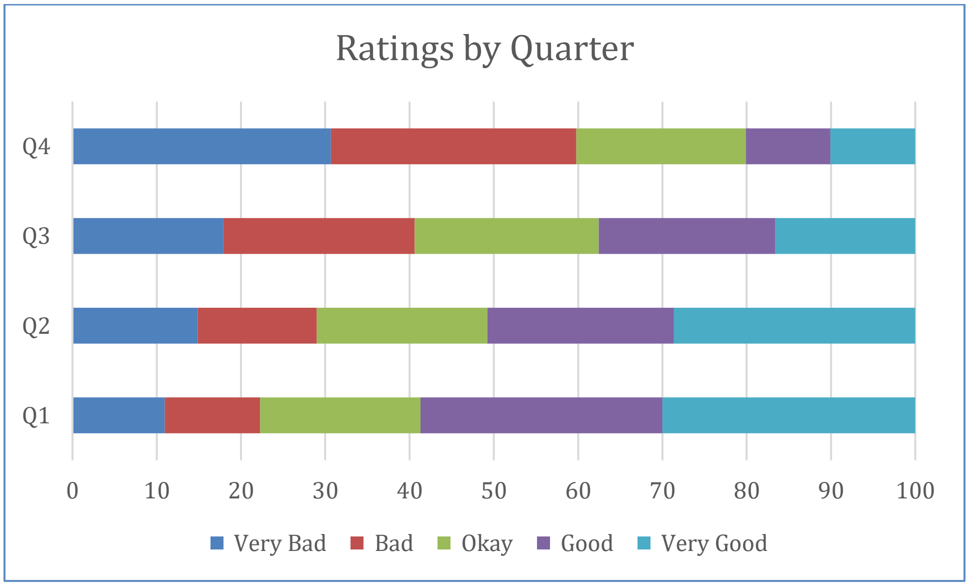
quarter_number	percent_vb	percent_b	percent_o	percent_g	percent_vg
1	10.9677	11.2903	19.0323	28.7097	30.0000
2	14.8855	14.1221	20.2290	22.1374	28.6260
3	17.9039	22.7074	21.8341	20.9607	16.5939
4	30.6533	29.1457	20.1005	10.0503	10.0503

#	Time	Action	Message
5	11:11:02	WITH revenue_t AS (SELECT quarter_number, SUM(quantity*(vehicle_price-(vehi...	4 row(s) returned

Observations and Insights:

- Yes, customers are getting more dissatisfied over time.

- In the first quarter, 30% of customers rated Very Good and only 10% rated Very Bad. By the fourth quarter it flipped, and only 10% of customers rated Very Good while 30% rated Very Bad.

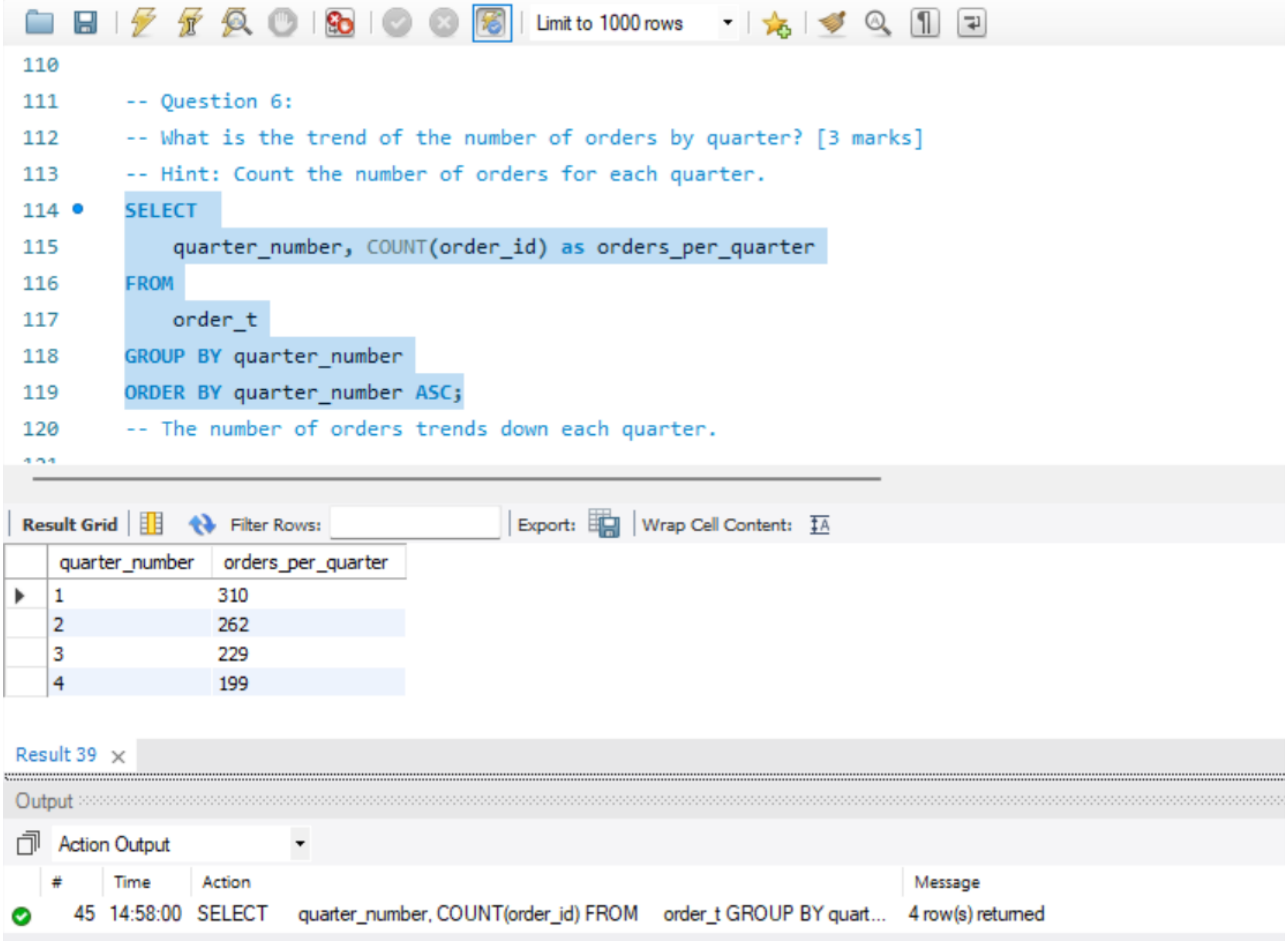


Question 6: What is the trend of the number of orders by quarter?

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:



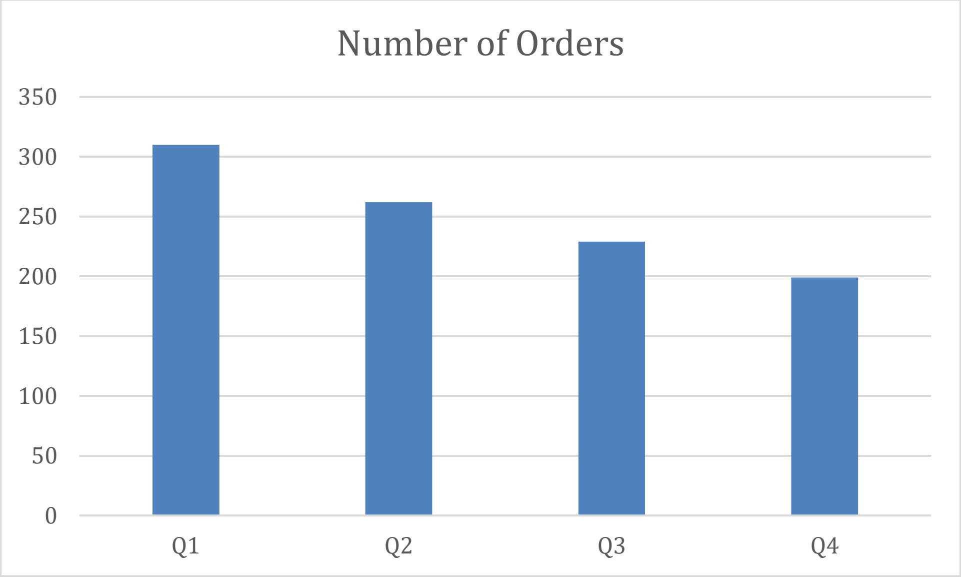
```
-- Question 6:
-- What is the trend of the number of orders by quarter? [3 marks]
-- Hint: Count the number of orders for each quarter.
SELECT
    quarter_number, COUNT(order_id) as orders_per_quarter
FROM
    order_t
GROUP BY quarter_number
ORDER BY quarter_number ASC;
-- The number of orders trends down each quarter.
```

quarter_number	orders_per_quarter
1	310
2	262
3	229
4	199

```
#   Time      Action                                                           Message
45  14:58:00  SELECT  quarter_number, COUNT(order_id) FROM  order_t GROUP BY quart...  4 row(s) returned
```

Observations and Insights:

- The number of orders trends down each quarter.
- The number of orders from Q1 to Q2 dropped by 48, Q2 to Q3 dropped by 33, and Q3 to Q4 dropped by 30.
- This means that though the orders are trending downwards, they are not accelerating in the drop.



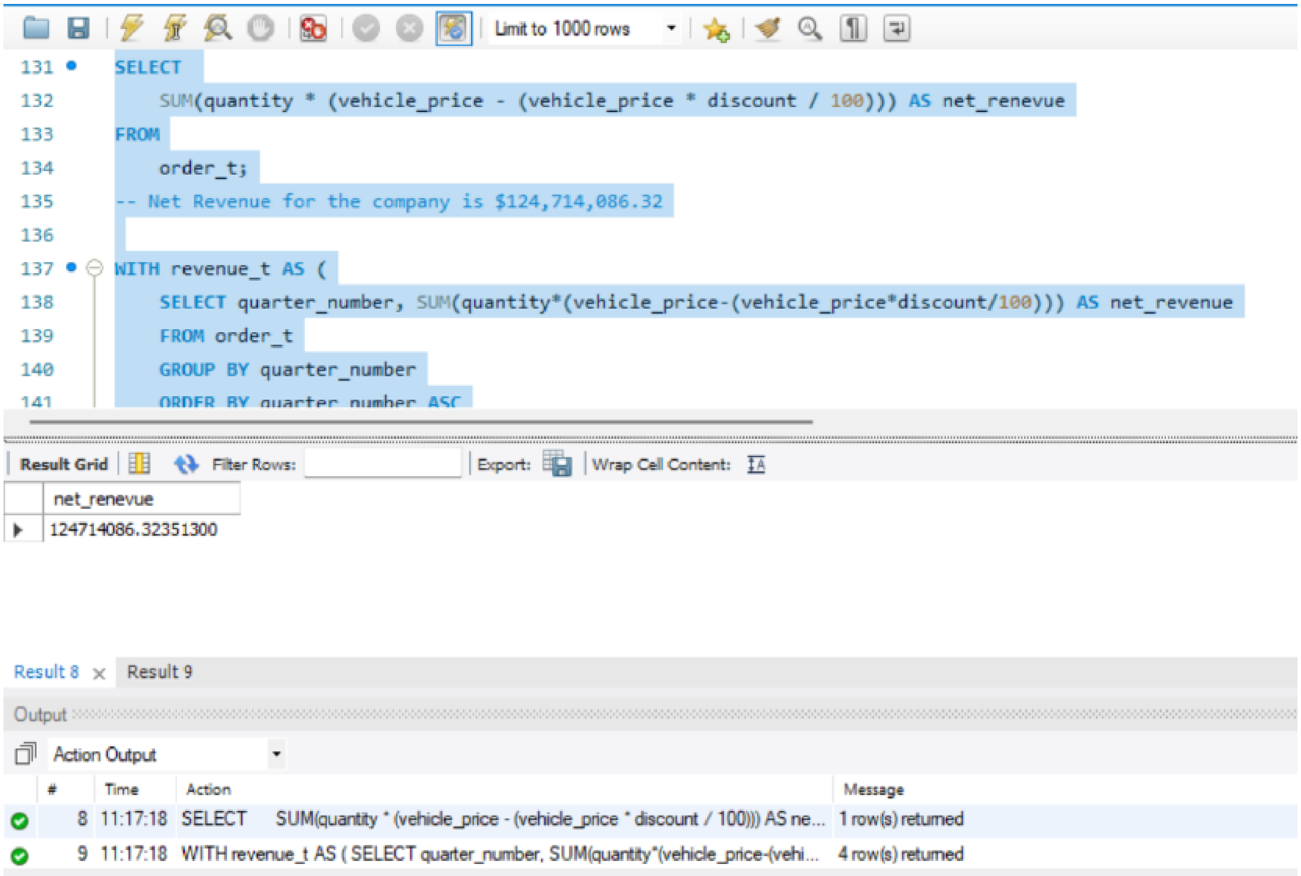
Question 7: Calculate the net revenue generated by the company. What is the quarter-over-quarter % change in net revenue?

Solution Query:

```
SELECT
   SUM(quantity * (vehicle_price - (vehicle_price * discount / 100))) AS net_renevue
FROM
   order_t;
-- Net Revenue for the company is $124,714,086.32

WITH revenue_t AS (
         SELECT quarter_number, SUM(quantity*(vehicle_price-(vehicle_price*discount/100))) AS net_revenue
   FROM order_t
   GROUP BY quarter_number
   ORDER BY quarter_number ASC
   )
SELECT quarter_number, net_revenue,
-- Can add in "LAG(net_revenue) OVER (ORDER BY quarter_number) AS previous_quarter," in order to show previous quarter, too.
         (net_revenue - LAG(net_revenue) OVER (ORDER BY quarter_number ASC))/ LAG(net_revenue) OVER (ORDER BY quarter_number ASC)*100 AS percent_change_revenue
FROM revenue_t
GROUP BY quarter_number
ORDER BY quarter_number ASC;
```

Output:

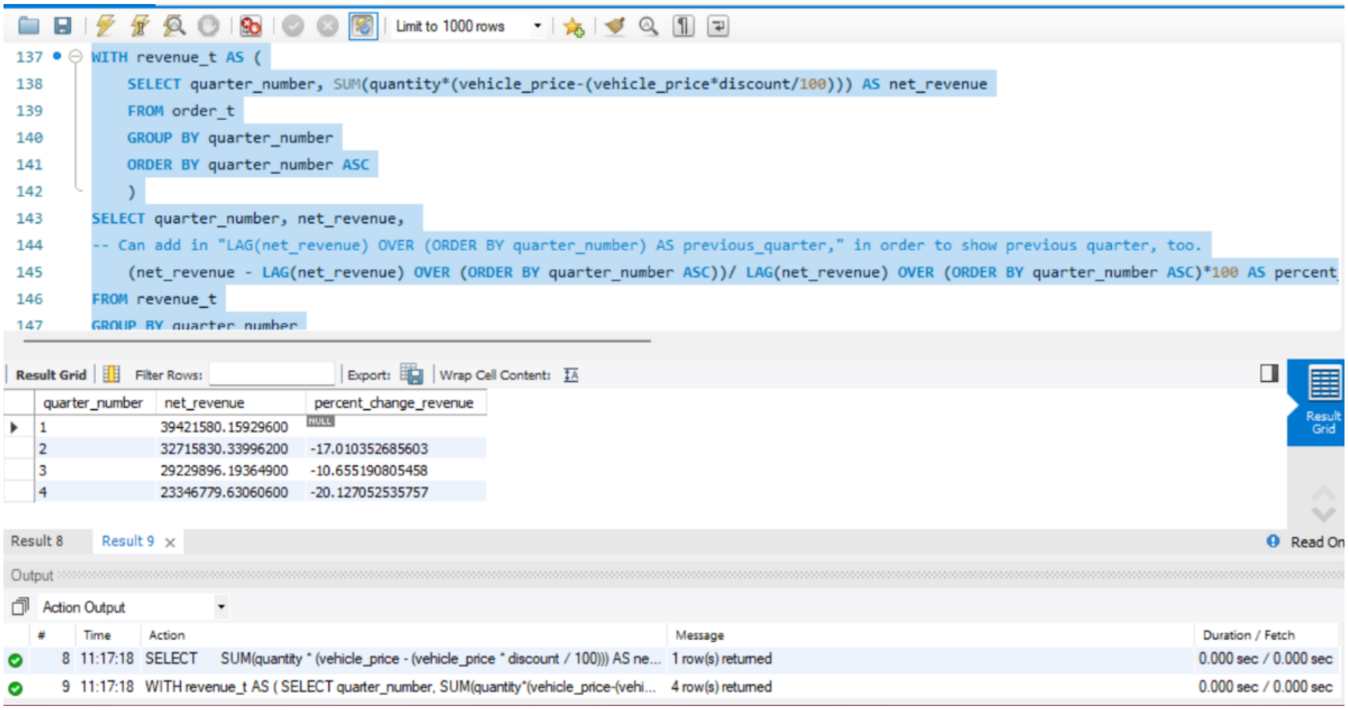


```
SELECT
    SUM(quantity * (vehicle_price - (vehicle_price * discount / 100))) AS net_renevue
FROM
    order_t;
-- Net Revenue for the company is $124,714,086.32

WITH revenue_t AS (
    SELECT quarter_number, SUM(quantity*(vehicle_price-(vehicle_price*discount/100))) AS net_revenue
    FROM order_t
    GROUP BY quarter_number
    ORDER BY quarter_number ASC
```

net_renevue
124714086.32351300

	#	Time	Action	Message
✓	8	11:17:18	SELECT SUM(quantity * (vehicle_price - (vehicle_price * discount / 100))) AS ne...	1 row(s) returned
✓	9	11:17:18	WITH revenue_t AS (SELECT quarter_number, SUM(quantity*(vehicle_price-(vehi...	4 row(s) returned



```sql
WITH revenue_t AS (
    SELECT quarter_number, SUM(quantity*(vehicle_price-(vehicle_price*discount/100))) AS net_revenue
    FROM order_t
    GROUP BY quarter_number
    ORDER BY quarter_number ASC
    )
SELECT quarter_number, net_revenue,
-- Can add in "LAG(net_revenue) OVER (ORDER BY quarter_number) AS previous_quarter," in order to show previous quarter, too.
    (net_revenue - LAG(net_revenue) OVER (ORDER BY quarter_number ASC))/ LAG(net_revenue) OVER (ORDER BY quarter_number ASC)*100 AS percent
FROM revenue_t
GROUP BY quarter_number
```

quarter_number	net_revenue	percent_change_revenue
1	39421580.15929600	NULL
2	32715830.33996200	-17.010352685603
3	29229896.19364900	-10.655190805458
4	23346779.63060600	-20.127052535757

#	Time	Action	Message	Duration / Fetch
8	11:17:18	SELECT SUM(quantity * (vehicle_price - (vehicle_price * discount / 100))) AS ne...	1 row(s) returned	0.000 sec / 0.000 sec
9	11:17:18	WITH revenue_t AS (SELECT quarter_number, SUM(quantity*(vehicle_price-(vehi...	4 row(s) returned	0.000 sec / 0.000 sec

Observations and Insights:

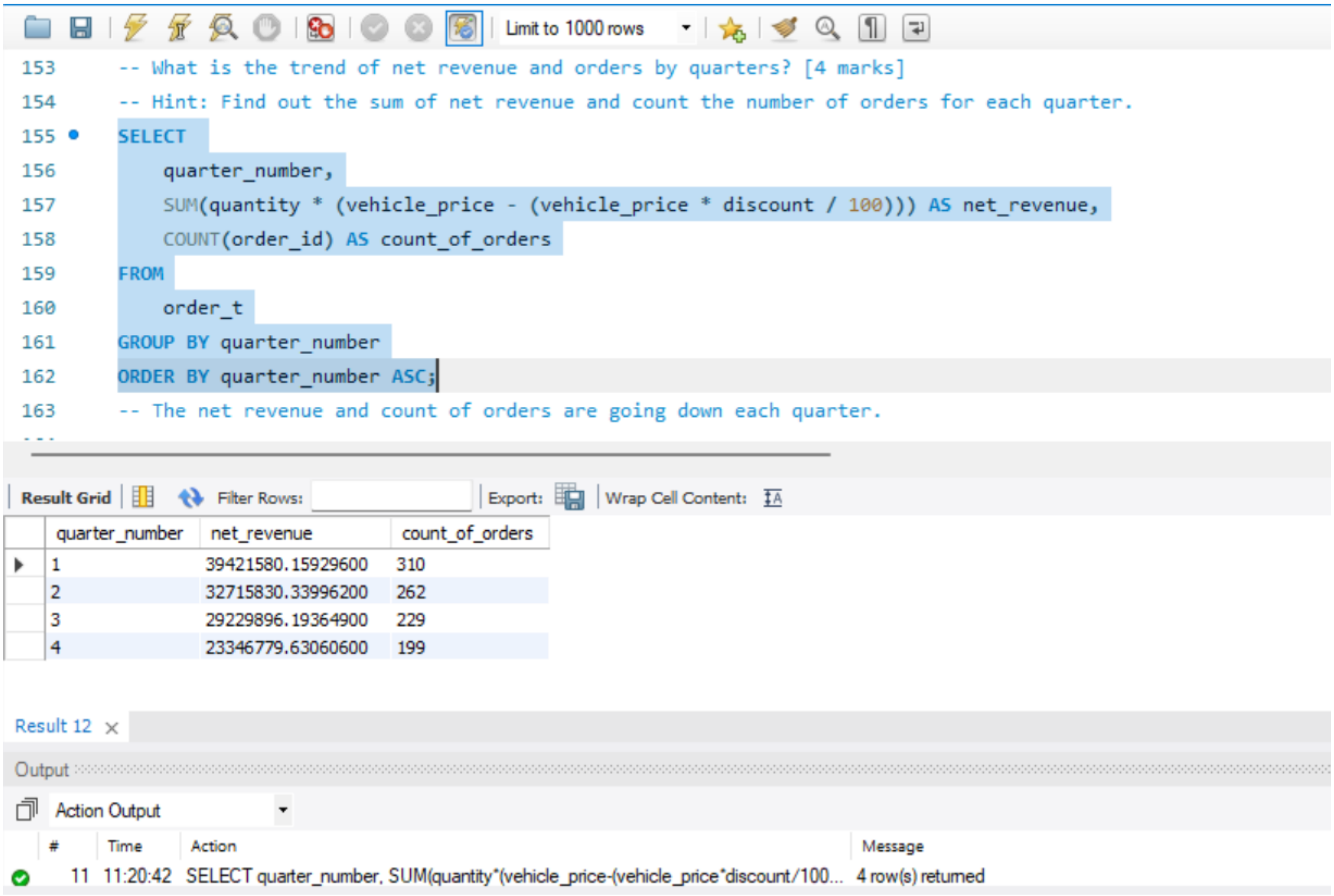
- The net revenue for the company for the year is $124,714,086.32
- The quarter-over-quarter change in net revenue from Q1 to Q2 went down 17.01%, Q2 to Q3 went down 10.66%, and Q3 to Q4 went down 20.13%.

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

Solution Query:

```
SELECT
    quarter_number,
    SUM(quantity * (vehicle_price - (vehicle_price * discount / 100))) AS net_revenue,
    COUNT(order_id) AS count_of_orders
FROM
    order_t
GROUP BY quarter_number
ORDER BY quarter_number ASC;
```

Output:



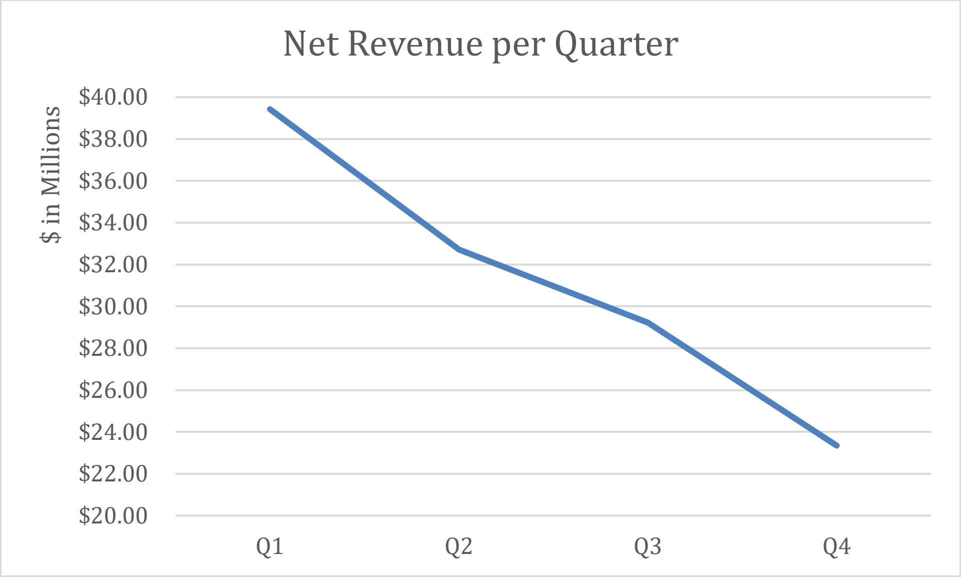
```
-- What is the trend of net revenue and orders by quarters? [4 marks]
-- Hint: Find out the sum of net revenue and count the number of orders for each quarter.
SELECT
    quarter_number,
    SUM(quantity * (vehicle_price - (vehicle_price * discount / 100))) AS net_revenue,
    COUNT(order_id) AS count_of_orders
FROM
    order_t
GROUP BY quarter_number
ORDER BY quarter_number ASC;
-- The net revenue and count of orders are going down each quarter.
```

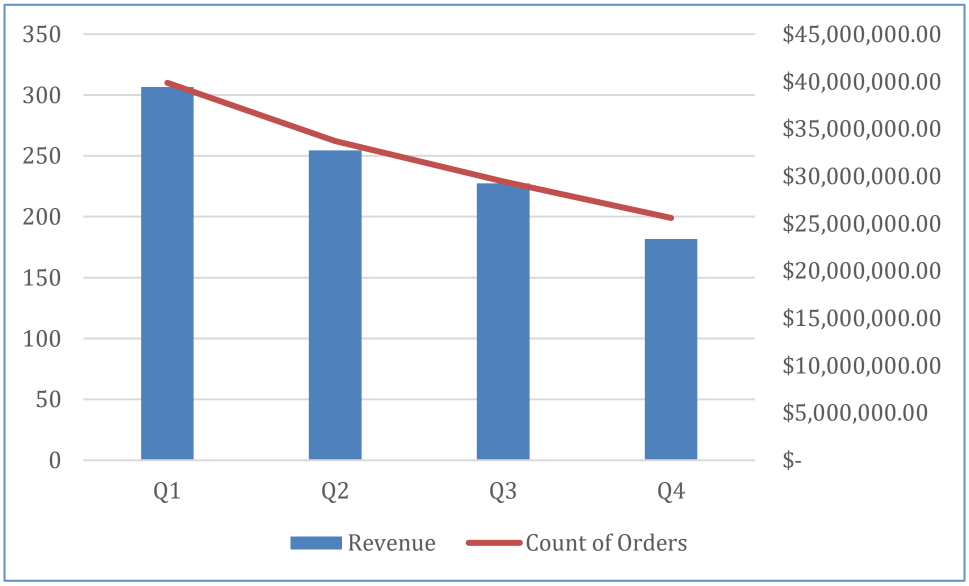
quarter_number	net_revenue	count_of_orders
1	39421580.15929600	310
2	32715830.33996200	262
3	29229896.19364900	229
4	23346779.63060600	199

#	Time	Action	Message
11	11:20:42	SELECT quarter_number, SUM(quantity*(vehicle_price-(vehicle_price*discount/100...	4 row(s) returned

Observations and Insights:

- The net revenue and count of orders per quarter are trending downwards.
- Due to the lower number of orders per quarter, the net revenue also is lowered as reflected in the graphs below.



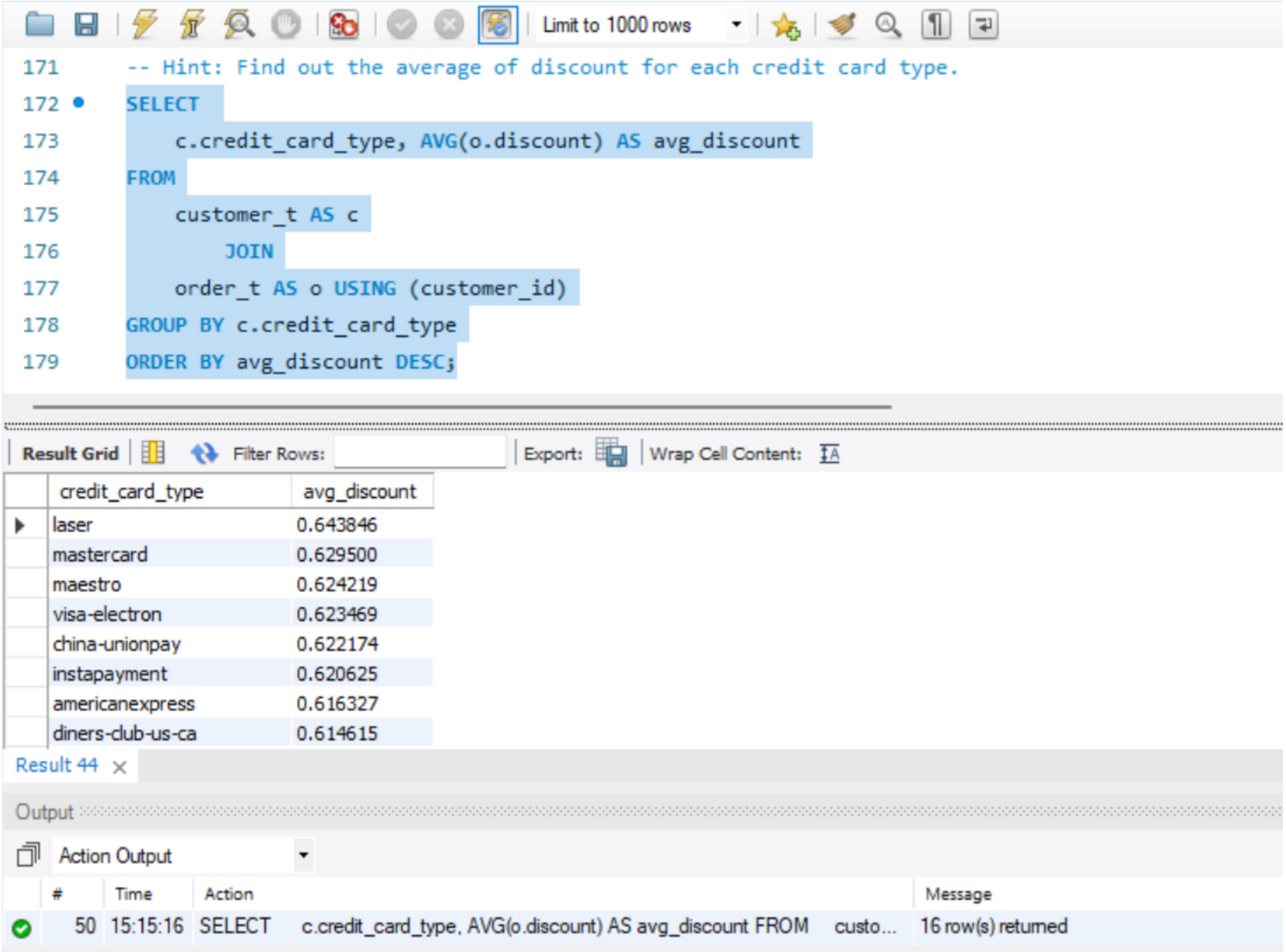


Question 9: What is the average discount offered for different types of credit cards?

Solution Query:

```
SELECT
    c.credit_card_type, AVG(o.discount) AS avg_discount
FROM
    customer_t AS c
        JOIN
    order_t AS o USING (customer_id)
GROUP BY c.credit_card_type
ORDER BY avg_discount DESC;
```

Output:



```
-- Hint: Find out the average of discount for each credit card type.
SELECT
    c.credit_card_type, AVG(o.discount) AS avg_discount
FROM
    customer_t AS c
        JOIN
    order_t AS o USING (customer_id)
GROUP BY c.credit_card_type
ORDER BY avg_discount DESC;
```

credit_card_type	avg_discount
laser	0.643846
mastercard	0.629500
maestro	0.624219
visa-electron	0.623469
china-unionpay	0.622174
instapayment	0.620625
americanexpress	0.616327
diners-club-us-ca	0.614615

#	Time	Action	Message
50	15:15:16	SELECT c.credit_card_type, AVG(o.discount) AS avg_discount FROM custo...	16 row(s) returned

Observations and Insights:

- The average discounts per credit card are as follows:

Laser	0.643846
Mastercard	0.629500
Maestro	0.624219
Visa-electron	0.623469
China-unionpay	0.622174
InstaPayment	0.620625
AmericanExpress	0.616327
Diners Club -us-ca	0.614615
Diners Club-carte blanche	0.614490
Switch	0.610233
Bankcard	0.609545
JCB	0.607382
Visa	0.600833
Diners Club -enroute	0.599792
Solo	0.585000
Diners Club -international	0.584000

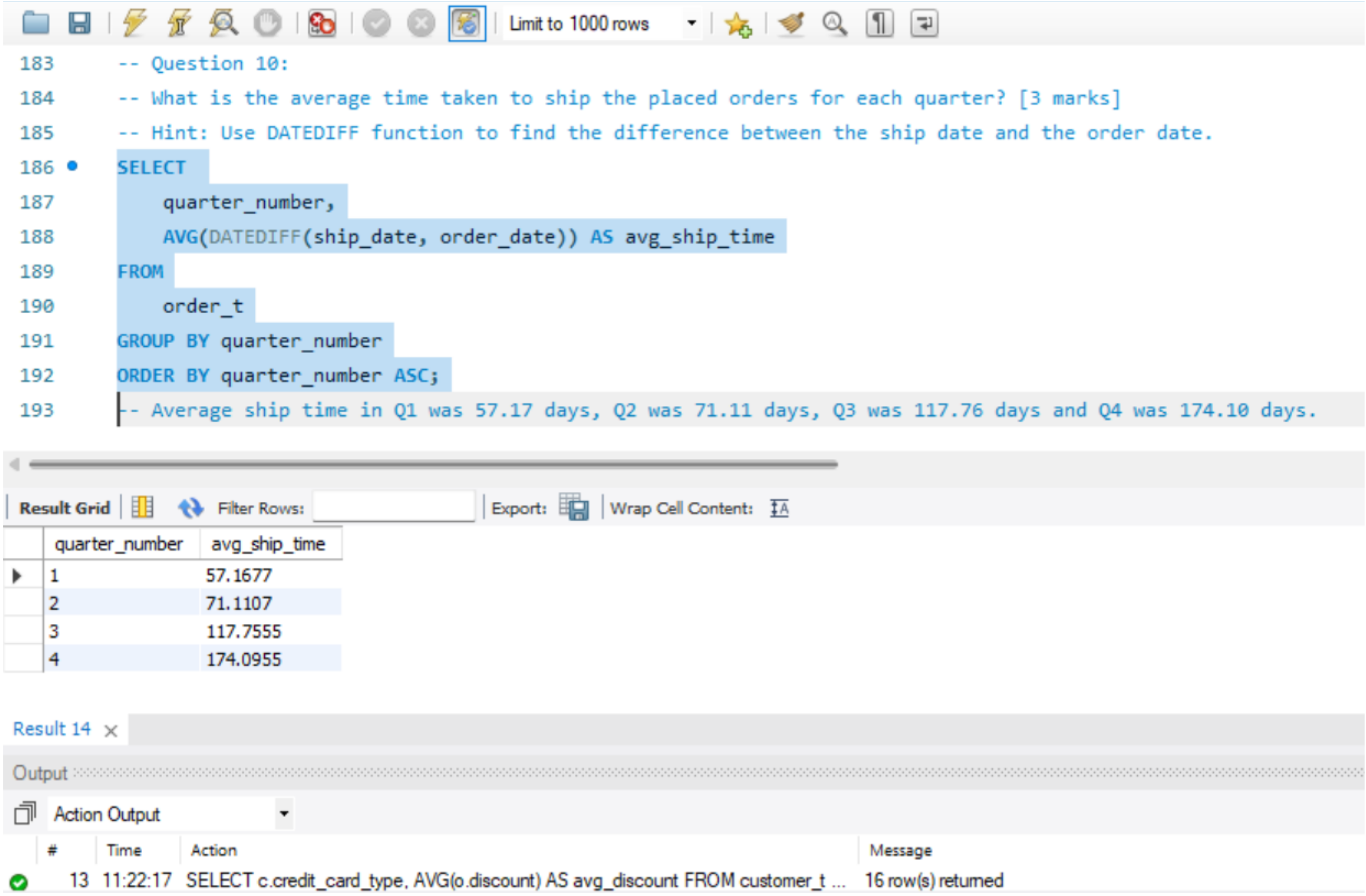
- The difference between the highest and lowest credit card type discount (between Laser and Diners Club-International) is only 0.059846%, resulting in an average vehicle discount price difference of $49.74.
- This suggests that the difference in credit card discounts is not impacting the downward trend of sales for New Wheels.

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

Solution Query:

```
SELECT
    quarter_number,
    AVG(DATEDIFF(ship_date, order_date)) AS avg_ship_time
FROM
    order_t
GROUP BY quarter_number
ORDER BY quarter_number ASC;
```

Output:



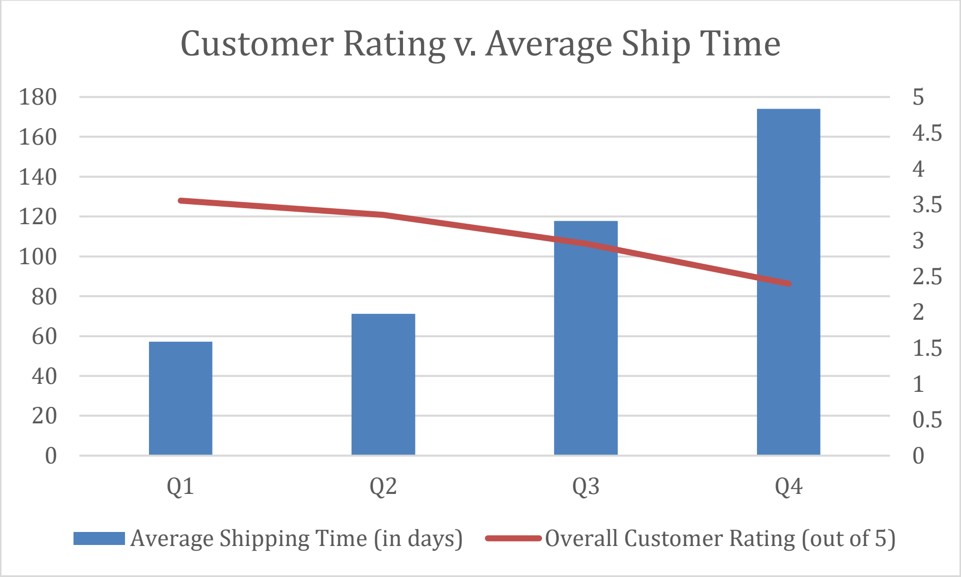
```
-- Question 10:
-- What is the average time taken to ship the placed orders for each quarter? [3 marks]
-- Hint: Use DATEDIFF function to find the difference between the ship date and the order date.
SELECT
    quarter_number,
    AVG(DATEDIFF(ship_date, order_date)) AS avg_ship_time
FROM
    order_t
GROUP BY quarter_number
ORDER BY quarter_number ASC;
-- Average ship time in Q1 was 57.17 days, Q2 was 71.11 days, Q3 was 117.76 days and Q4 was 174.10 days.
```

quarter_number	avg_ship_time
1	57.1677
2	71.1107
3	117.7555
4	174.0955

	#	Time	Action	Message
✓	13	11:22:17	SELECT c.credit_card_type, AVG(o.discount) AS avg_discount FROM customer_t ...	16 row(s) returned

Observations and Insights:

- The average time it took to ship orders placed in Q1 was 57.17 days, Q2 was 71.11 days, Q3 was 117.76 days and Q4 was 174.10 days.
- This is a change of nearly 14 days from Q1 to Q2, 46.65 days from Q2 to Q3, and 56.34 days (nearly 2 months) from Q3 to Q4.
- This data suggests that the increased delay in shipping time plays a large factor in the decrease in customer satisfaction quarter to quarter.



Business Metrics Overview

Total Revenue	Total Orders	Total Customers	Average Rating
$124,714,086.32	1,000	994	3.066
Last Quarter Revenue	**Last quarter Orders**	**Average Days to Ship**	**% Good Feedback**
$23,346,779.63	199	105	20.46%

Business Recommendations

- The biggest impact on customer satisfaction is the time it takes to ship the vehicle after it's been ordered. Customer satisfaction was at its highest in quarter 1 when the average shipping time was 57 days. I recommend going back to lowering your average shipping time, as from quarter 1 to quarter 4 the shipping time increased by an average of 117 days.

- 349 out of 994 of your customer base (35%) comes from California, Texas, Florida, and New York. I recommend offering a sale to draw in more customers from those four states.
  - The most popular vehicle makers in each of those states were Audi in California, Chevrolet in Texas, and Toyota in New York and Florida, therefore a sale on those makers would likely bring in a larger client base during the promotion.

- New Wheels received only "Very Bad" or "Bad" feedback when shipping orders through shippers Realcube, Meembee, Livefish, Shuffletag, Myworks, Cogibox, Buzzdog, Jatri, Aibox, Gigaclub, Mita, Fliptune, Dynava, Dynazzy, Jaxbean, Shufflebeat, Vitz, Babblestorm, Fivechat, Flipstorm, Blogpad, Skinder, Twitterbridge, Flashset, Bluezoom, Tanoodle, Browsezoom, Fatz, Topiclounge, Aimbo, Ntags, Skyndu, Meedoo, Tavu, Rhybox, Npath, Tambee, Meemm, Dabjam, Jabbertype, Tekfly, Mudo, Demivee, Voolith, Yoveo, and Demimbu. Therefore, when possible, try to avoid shipping with these shipper names.