

# 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.

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

### Solution Query:

*(I initially put this, but there seems to be some discrepancy between the data in customer\_t & order\_t. Where I can throughout the project, I attempted to base information off of order\_t as it provided more customers (994) compared to customer\_t (133). So I figured it would be more realistic to work with that.)*

```
SELECT
COUNT(DISTINCT c.customer_id) AS total_active_customers
FROM
order_t AS o
INNER JOIN
customer_t AS c
ON o.customer_id = c.customer_id
WHERE o.quantity >= 1;
```

*What I ended up using for customer count.*

```
SELECT
COUNT(DISTINCT customer_id) AS total_active_customers
FROM order_t ;
```

*Find distribution count*

```
SELECT
state,
customer_per_state,
ROUND(((customer_per_state / 133.0)*100),2) AS percentage_distribution
FROM
(
SELECT
c.state,
COUNT(DISTINCT c.customer_id) AS customer_per_state
FROM
order_t AS o
INNER JOIN
customer_t AS c
ON o.customer_id = c.customer_id
WHERE o.quantity >= 1
GROUP BY c.state
ORDER BY customer_per_state DESC
)
GROUP BY state
ORDER BY customer_per_state DESC;
```

Query 1 **Success**: SELECT COUNT(DISTINCT customer\_id) AS total\_active\_customers FROM order\_t  
Output:  
Showing 1 rows

total_active_customers
994

Query 2 **Success**: SELECT state, customer\_per\_state, ROUND(((customer\_per\_state / 133.0)\*100),2) AS percentage\_distribution FROM order\_t AS o INNER JOIN customer\_t AS c WHERE o.quantity >= 1 GROUP BY c.state ORDER BY customer\_per\_state DESC ) GROUP BY c.state ORDER BY customer\_per\_state DESC  
Output:  
Showing first 10 rows out of 37 rows

state	customer_per_state	percentage_distribution
California	17	12.78
Texas	10	7.52
Florida	9	6.77
New York	7	5.26
Virginia	5	3.76
Michigan	5	3.76

### Output:

### Observations and Insights:

- There are a total of 994 customers who have placed an order.
- From the limited data provided on the customers (only 133 of the 994), almost 1/3 of the 133 (~32.33%) of them reside within CA, TEX, FL, and NY.
- It appears that all of the data collected on New Wheels customers show that they all reside within the USA and are active within 37 states.

- The top 10 states, out of the active 37, account for over 50% (~60.15%) of total orders placed (within the customer table).

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

### Solution Query:

```
SELECT
  p.vehicle_maker,
  COUNT(o.order_id) AS order_count,
  SUM(o.quantity) AS total_quantity_ordered
FROM
  order_t AS o
INNER JOIN
  product_t AS p
ON o.product_id = p.product_id
GROUP BY
  p.vehicle_maker
ORDER BY
  total_quantity_ordered DESC
LIMIT 5;
```

### Output:

SQL query passed

Query 1 **Passed**: SELECT p.vehicle\_maker, COUNT(o.order\_id) AS order\_count, SUM(o.quantity) AS total\_quantity\_ordered FROM order\_t AS o INNER JOIN product\_t AS p ON o.product\_id = p.product\_id GROUP BY p.vehicle\_maker ORDER BY total\_quantity\_ordered DESC LIMIT 5;

Output:

Showing 5 rows

vehicle_maker	order_count	total_quantity_ordered
Chevrolet	83	125
Ford	63	97
Toyota	52	78
Pontiac	50	75
Dodge	50	72

### Observations and Insights:

- The top 5 vehicle makers preferred by customers when looking at total quantity ordered are:**  
1. Chevrolet / 2. Ford / 3. Toyota / 4. Pontiac / 5. Dodge
- In the respective 3, 4, and 5 spot on the list, Toyota, Pontiac, and Dodge are clustered together by a small amount of 6 total items ordered. The median quantity ordered of these vehicle makers, 75, trail Ford by 22, and Chevrolet by 50.
- If marketing efforts toward a specific vehicle maker is to be considered, then I would highly recommend the majority of efforts to be toward Chevrolet as they command ~8.26% of the total 1512 quantity ordered across all vehicle makers.
- On the other hand, the top 5 vehicle makers as a whole command ~29.56% of the total 1512 quantity ordered across all vehicle makers.

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

### Solution Query:

```

SELECT
    state,
    vehicle_maker,
    customer_count,
    total_quantity_ordered,
    state_rank
FROM
(
    SELECT
        state,
        vehicle_maker,
        customer_count,
        total_quantity_ordered,
        state_rank
    FROM
    (
        SELECT
            state,
            vehicle_maker,
            customer_count,
            total_quantity_ordered,
            RANK() OVER (PARTITION BY state ORDER BY
total_quantity_ordered DESC) AS state_rank
        FROM
        (
            SELECT
                c.state,
                p.vehicle_maker,
                COUNT(DISTINCT o.customer_id) AS customer_count,
                SUM(o.quantity) AS total_quantity_ordered
            FROM
                order_t AS o
            FULL JOIN
                customer_t AS c
            ON o.customer_id = c.customer_id
            INNER JOIN
                product_t AS p
            ON o.product_id = p.product_id
            GROUP BY c.state
        )
    )
    WHERE state_rank = 1
    ORDER BY total_quantity_ordered DESC
)
WHERE state IS NOT NULL;

```

```
Query 1 : SELECT state, vehicle_maker, customer_count, total_quantity_ordered, state_rank FROM ( SELECT  
customer_count, total_quantity_ordered, state_rank FROM ( SELECT state, vehicle_maker, customer_count, total_qual  
(PARTITION BY state ORDER BY total_quantity_ordered DESC) AS state_rank FROM ( SELECT c.state, p.vehicle_mak  
o.customer_id) AS customer_count, SUM(o.quantity) AS total_quantity_ordered FROM order_t AS o FULL JOIN custor  
o.customer_id = c.customer_id INNER JOIN product_t AS p ON o.product_id = p.product_id GROUP BY c.state ) ) WH  
BY total_quantity_ordered DESC ) WHERE state IS NOT NULL
```

Output:

state	vehicle_maker	customer_count	total_quantity_ordered	state_rank
California	Buick	17	27	1
Florida	Volvo	9	17	1
Texas	Lotus	10	14	1
New York	Daewoo	7	10	1
Virginia	Volkswagen	5	9	1
District of Columbia	Volvo	5	8	1

### Output:

### Observations and Insights:

- **The state ranking doesn't seem to provide much valuable insight other than the top 5 vehicle makers seem to hold the #1 ranking in a few of the states. Most notably, Chevrolet.**
- Otherwise, the top rank per state doesn't seem to hold much credibility or there is an issue with the data collected as GMC held a majority of orders but wasn't assigned to any state. The issue may be that more of the customer information needs to be collected as you only have 133 customers out of 994 active customers.

**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,
    avg_rating_per_quarter,
    AVG(avg_rating_per_quarter) OVER () AS overall_feedback_average
FROM (
    SELECT
        quarter_number,
        ROUND (AVG(feedback_score), 2) AS avg_rating_per_quarter
    FROM (
        SELECT
            quarter_number,
            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
                ELSE NULL
            END AS feedback_score
        FROM order_t
        WHERE customer_feedback IS NOT NULL
    ) AS feedback_scores
    GROUP BY quarter_number
)
GROUP BY quarter_number
;
```

**Output:**

SQL queries passed

Query 1 **Passed**: SELECT quarter\_number, avg\_rating\_per\_quarter, AVG(avg\_rating\_per\_quarter) OVER () AS overall\_feedback\_average FROM (SELECT quarter\_number, ROUND (AVG(feedback\_score), 2) AS avg\_rating\_per\_quarter FROM (SELECT quarter\_number, 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 ELSE NULL END AS feedback\_score FROM order\_t WHERE customer\_feedback IS NOT NULL ) AS feedback\_scores GROUP BY quarter\_number) GROUP BY quarter\_number

Output:

Showing 4 rows

quarter_number	avg_rating_per_quarter	overall_feedback_aver...
1	3.55	3.065
2	3.35	3.065
3	2.96	3.065
4	2.4	3.065

**Observations and Insights:**

- By creating a numerical score based off the customer feedback responses, we were able to find the overall average rating of New Wheels service to be ‘Okay’ (3.065)
- The quarterly ratings rounded out to ‘Good’ (Q1), ‘Okay’ (Q2 & Q3), and ‘Bad’ (Q4). There is also a distinct decline in feedback. New Wheels gets the best reviews in Q1 and gets consistently worse through Q4.
- Focusing efforts toward maintaining consistency throughout the year could prove to be very beneficial with regards to customer satisfaction.

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

Solution Query:

```
SELECT
COUNT(customer_feedback)
FROM
order_t
WHERE customer_feedback IN ('Bad', 'Very Bad', 'Okay', 'Good', 'Very Good');
```

```
SELECT
ROUND ((SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS very_bad_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS bad_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS okay_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS good_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS very_good_percentage
FROM
order_t;
```

```
SELECT
quarter_number,
ROUND ((SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS very_bad_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS bad_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS okay_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS good_percentage,
ROUND ((SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) / 1000.0)*100, 2) AS very_good_percentage
FROM
order_t
GROUP BY
quarter_number
ORDER BY
quarter_number ASC;
```

Output:

Query 1: SELECT COUNT(customer\_feedback) FROM order\_t WHERE customer\_feedback IN ('Bad', 'Very Bad', 'Okay', 'Good', 'Very Good');  
Output:  
Showing 1 rows

COUNT(customer_feedback)
1000

Query 2: SELECT ROUND ((SUM(CASE WHEN customer\_feedback = 'Very Bad' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS very\_bad\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Bad' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS bad\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Okay' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS okay\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Good' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS good\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Very Good' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS very\_good\_percentage FROM order\_t;  
Output:  
Showing 1 rows

very_bad_percentage	bad_percentage	okay_percentage	good_percentage	very_good_percentage
17.5	18.2	20.2	21.5	22.6

Query 3: SELECT quarter\_number, ROUND ((SUM(CASE WHEN customer\_feedback = 'Very Bad' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS very\_bad\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Bad' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS bad\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Okay' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS okay\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Good' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS good\_percentage, ROUND ((SUM(CASE WHEN customer\_feedback = 'Very Good' THEN 1 ELSE 0 END) / 1000.0)\*100, 2) AS very\_good\_percentage FROM order\_t GROUP BY quarter\_number ORDER BY quarter\_number ASC  
Output:  
Showing 4 rows

quarter_number	very_bad_percentage	bad_percentage	okay_percentage	good_percentage	very_good_percentage
1	3.4	3.5	5.9	8.9	9.3
2	3.9	3.7	5.3	5.8	7.5
3	4.1	5.2	5	4.8	3.8
4	6.1	5.8	4	2	2

### Observations and Insights:

- Overall, the feedback tends to be better than worse as 'Very good' is the highest percentage response to service at 22.6% and 'Very bad' is the lowest percentage response at 17.5%.
- Unfortunately, customer satisfaction also shows a negative trend over time as the 'Very bad' & 'Bad' percentages consistently grow from Q1 to Q4 with Q4 being the highest percentage response for both. Meanwhile 'Okay', 'Good', and 'Very good' consistently decrease in percentage responses over time with Q4 being the lowest percentage responses for all three of those categories.
- Once again, a focus on increasing customer satisfaction in the second half of the calendar year could prove to be very beneficial for New Wheels.

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

### Solution Query:

Initially I went with this

```
SELECT
    quarter_number,
    COUNT(order_id) AS total_orders
FROM
    (SELECT
        c.customer_id,
        o.order_id,
        o.quarter_number
    FROM
        customer_t AS c
    INNER JOIN
        order_t AS o
        ON c.customer_id = o.customer_id
    )
GROUP BY quarter_number
;
```

Eliminated customer\_t for final output.

```
SELECT
    quarter_number,
    COUNT(order_id) AS total_orders
FROM
    (SELECT
        order_id,
        quarter_number
    FROM
        order_t
    )
GROUP BY quarter_number
;
```

### Output:

#### SQL queries passed

Query 1 (Passed): SELECT quarter\_number, COU  
BY quarter\_number

Output:

Showing 4 rows

quarter_number	total_orders
1	310
2	262
3	229
4	199

### Observations and Insights:

- Again we are seeing a negative trend over the calendar year as the quantity of orders placed is consistently decreasing from Q1 to Q4.
- Volume of orders rank highest to lowest in order from Q1, Q2, Q3, Q4.
- A focus on increasing order volume in Q3 and Q4 would be beneficial.

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

**Solution Query:**

— Net Revenue \$124,714,086.32

```
SELECT
    SUM(sales_discounted) AS net_revenue
FROM
    (
        SELECT
            order_id,
            (vehicle_price * (1 - (discount/100)) * quantity) AS sales_discounted
        FROM
            order_t
    );
```

SQL queries passed

Query 1 **Passed**: SELECT SUM(sales\_discounted) FROM order\_t )  
Output:  
Showing 1 rows

net\_revenue

124714086.32351312

— Percent change in revenue over each quarter

```
SELECT
    quarter_number,
    net_revenue,
    previous_quarter_revenue,
    CASE WHEN
        previous_quarter_revenue IS NULL THEN NULL
    ELSE
        ROUND((net_revenue - previous_quarter_revenue) / (previous_quarter_revenue)*100, 2)
    END AS
    quarterly_revenue_percent_change
FROM
    (
        SELECT
            quarter_number,
            net_revenue,
            ROUND (LAG(net_revenue) OVER (ORDER BY quarter_number), 2)
            AS previous_quarter_revenue
        FROM
            (
                SELECT
                    quarter_number,
                    order_id,
                    (vehicle_price * (1 - (discount/100)) * quantity) AS sales_discounted
                FROM
                    order_t
            )
        GROUP BY quarter_number
    )
```

SQL queries passed

Query 1 **Passed**: SELECT quarter\_number, net\_revenue, previous\_quarter\_revenue, CASE WHEN previous\_quarter\_revenue IS NULL THEN NULL ELSE ROUND((net\_revenue - previous\_quarter\_revenue) / (previous\_quarter\_revenue)\*100, 2) END AS quarterly\_revenue\_percent\_change FROM ( SELECT quarter\_number, net\_revenue, ROUND (LAG(net\_revenue) OVER (ORDER BY quarter\_number), 2) AS previous\_quarter\_revenue FROM ( SELECT quarter\_number, order\_id, (discount/100) \* quantity) AS sales\_discounted FROM order\_t ) GROUP BY quarter\_number )

Output:

Showing 4 rows

quarterly\_revenue\_percent\_change

quarter_number	net_revenue	previous_quarter_reve...	quarterly_revenue_per...
1	39421580.16		
2	32715830.34	39421580.16	-17.01
3	29229896.19	32715830.34	-10.66
4	23346779.63	29229896.19	-20.13

**Output:**

### Observations and Insights:

- Net revenue is at \$124,714,086.32 for the entire year.
- When looking at the percent change in revenue over a quarterly analysis, it is evident that there is a consistent drop off in revenue over the course of the year and a majority of that drop is coming from Q1 to Q2 and Q3 to Q4.
- With Q1 revenue as the baseline, Q3 to Q4 is the largest drop in revenue (-20.13%), Q1 to Q2 is the second largest drop in revenue (-17.01%), and Q2 to Q3 (-10.66%) is the lowest drop in revenue.
- Q1 is clearly the strongest quarter and Q4 is clearly the weakest quarter with regards to net revenue.



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

### Solution Query:

```
SELECT
    quarter_number,
    COUNT(order_id) AS total_orders,
    SUM(quantity) AS total_order_quantity,
    ROUND(SUM(sales_discounted), 2) AS net_revenue
FROM
    (
        SELECT
            quarter_number,
            order_id,
            quantity,
            (vehicle_price * (1 - (discount/100)) + quantity) AS sales_discounted
        FROM
            order_t
    )
GROUP BY quarter_number;

SELECT
    quarter_number,
    total_order_quantity,
    net_revenue,
    previous_quarter_revenue,
    previous_quarter_quantity,
    CASE WHEN
        previous_quarter_revenue IS NULL THEN NULL
    ELSE
        ROUND((net_revenue - previous_quarter_revenue) / (previous_quarter_revenue)*100, 2)
    END AS
    quarterly_revenue_percent_change,
    CASE WHEN
        previous_quarter_quantity IS NULL THEN NULL
    ELSE
        ROUND((total_order_quantity - previous_quarter_quantity) /
        (previous_quarter_quantity)*100, 2)
    END AS
    quarterly_orders_percent_change
FROM
    (
        SELECT
            quarter_number,
            total_order_quantity,
            net_revenue,
            LAG(net_revenue) OVER (ORDER BY quarter_number)
            AS previous_quarter_revenue,
            LAG(total_order_quantity) OVER (ORDER BY quarter_number)
            AS previous_quarter_quantity
        FROM
            (
                SELECT
                    quarter_number,
                    ROUND(SUM(quantity), 2) AS total_order_quantity,
                    ROUND(SUM(sales_discounted), 2) AS net_revenue
                FROM
                    (
                        SELECT
                            quarter_number,
                            quantity,
                            (vehicle_price * (1 - (discount/100)) + quantity) AS
                            sales_discounted
                        FROM
                            order_t
                    )
                GROUP BY quarter_number
            )
    )
```

#### SQL queries passed

Query 1 **Passed**: SELECT quarter\_number, COUNT(order\_id) AS total\_orders, SUM(quantity) AS total\_order\_quantity, ROUND(SUM(sales\_discounted), 2) AS net\_revenue FROM ( SELECT quarter\_number, order\_id, quantity, (vehicle\_price \* (1 - (discount/100)) + quantity) AS sales\_discounted FROM order\_t ) GROUP BY quarter\_number

Output:

Showing 4 rows

quarter_number	total_orders	total_order_quantity	net_revenue
1	310	464	39421580.16
2	262	400	32715830.34
3	229	342	29229896.19
4	199	306	23346779.63

Query 2 **Passed**: SELECT quarter\_number, total\_order\_quantity, net\_revenue, previous\_quarter\_revenue, previous\_quarter\_quantity, CASE WHEN previous\_quarter\_revenue IS NULL THEN NULL ELSE ROUND((net\_revenue - previous\_quarter\_revenue) / (previous\_quarter\_revenue)\*100, 2) END AS quarterly\_revenue\_percent\_change, CASE WHEN previous\_quarter\_quantity IS NULL THEN NULL ELSE ROUND((total\_order\_quantity - previous\_quarter\_quantity) / (previous\_quarter\_quantity)\*100, 2) END AS quarterly\_orders\_percent\_change FROM ( SELECT quarter\_number, total\_order\_quantity, net\_revenue, LAG(net\_revenue) OVER (ORDER BY quarter\_number) AS previous\_quarter\_revenue, LAG(total\_order\_quantity) OVER (ORDER BY quarter\_number) AS previous\_quarter\_quantity FROM ( SELECT quarter\_number, ROUND(SUM(quantity), 2) AS total\_order\_quantity, ROUND(SUM(sales\_discounted), 2) AS net\_revenue FROM ( SELECT quarter\_number, quantity, (vehicle\_price \* (1 - (discount/100)) + quantity) AS sales\_discounted FROM order\_t ) GROUP BY quarter\_number ) )

Output:

Showing 4 rows

quarter_number	total_order_quantity	net_revenue	previous_quarter_reve...	previous_quarter_qua...	quarterly_revenue_per...	quarterly_orders_perc...
1	464	39421580.16				
2	400	32715830.34	39421580.16	464	-17.01	-13.79
3	342	29229896.19	32715830.34	400	-10.66	-14.5
4	306	23346779.63	29229896.19	342	-20.13	-10.53

### Output:

### Observations and Insights:

- An important metric to note here is that I took used total order quantity as a count of orders per quarter here instead of the customer count per quarter as quantity is the driving force for revenue.
- Much like the trend in revenue (highest to lowest Q1 -> Q4), we can see that New Wheels quantity ordered goes from highest to lowest in order of Q1 to Q2 to Q3 to Q4.
- The biggest red flag apparent from this query is that the total quantity ordered from Q3 to Q4 drops by 10.53% (the lowest drop in quantity ordered when viewing quarter to quarter), yet the net revenue made by New Wheels from Q3 to Q4 drops by 20.13% (the largest drop in net revenue when viewing quarter to quarter).
- This finding leads me to believe that there may be too many discounts offered in Q4.

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

### Solution Query:

```
SELECT
  c.credit_card_type,
  AVG(discount) OVER (PARTITION BY credit_card_type)
  AS average_discount
FROM
  customer_t AS c
INNER JOIN
  order_t AS o
  ON c.customer_id = o.customer_id
GROUP BY
  c.credit_card_type
ORDER BY
  average_discount ASC;
```

```
SELECT
  c.credit_card_type,
  AVG(discount) OVER (PARTITION BY
  credit_card_type)
  AS average_discount
FROM
  customer_t AS c
INNER JOIN
  order_t AS o
  ON c.customer_id = o.customer_id
GROUP BY
  c.credit_card_type
ORDER BY
  average_discount DESC;
```

### Output:

Query 1 **Result**: SELECT c.credit\_card\_type, AVG(discount) OVER (PARTITION BY credit\_card\_type) AS average\_discount FROM customer\_t AS c INNER JOIN order\_t AS o ON c.customer\_id = o.customer\_id GROUP BY credit\_card\_type  
Output:  
Showing first 10 rows out of 16 rows

credit_card_type	average_discount
diners-club-us-ca	0.45
maestro	0.5
mastercard	0.54
diners-club-carte-blanc	0.55
diners-club-international	0.57
visa-electron	0.59
laser	0.6

SQL queries practice

Query 1 **Result**: SELECT c.credit\_card\_type, AVG(discount) OVER (PARTITION BY credit\_card\_type) AS average\_discount FROM customer\_t AS c INNER JOIN order\_t AS o ON c.customer\_id = o.customer\_id GROUP BY credit\_card\_type  
Output:  
Showing first 10 rows out of 16 rows

credit_card_type	average_discount
instapayment	0.76
switch	0.76
americanexpress	0.73
solo	0.7
jcb	0.67
china-unionpay	0.64
visa	0.64

### Observations and Insights:

- The top 5 highest average in discounts offered by cards are 1. Instapayment 2. Switch 3. American Express 4. Solo 5. Jcb.
- The average of lowest 5 discounts offered by cards are 16. Diners-club-us-ca 15. Maestro 14. Mastercard 13. Diners-club-carte-blanc 12. Diners-club-international
- MIN = .45% Q1 = .56.0% (Median = .625% / Mean = .62.19%) Q3 = .68.5% MAX = .76%
- When looking at which cards correlate to positive revenue for New Wheels, any discount lower than .56% could be considered good, anything between .56% and .68.5% could be considered okay, and anything above .68.5% could be considered bad for New Wheels revenue.

## 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
    avg_shipping_time
FROM
    order_t
GROUP BY
    quarter_number
ORDER BY
    quarter_number;
```

### Output:

SQL queries passed

Query 1 (Passed): SELECT quarter\_number, ROUND(AVG(julianday(ship\_date) - julianday(order\_date)), 2) AS avg\_shipping\_time FROM order\_t GROUP BY quarter\_number ORDER BY quarter\_number;

Output:  
Showing 4 rows

quarter_number	avg_shipping_time
1	57.17
2	71.11
3	117.76
4	174.1

### Observations and Insights:

- The average shipping time seems to have a negative correlation alongside net revenue & orders over the course of the year as shipping time increases while net revenue & orders decrease consistently from Q1 to Q4.
- The second half of the calendar year is still New Wheels major concern when looking at average shipping time. Q3 and Q4 are more than 2x the average time to ship than in Q1 and Q2 (respectively), and Q4 averages more than 3x the time it takes to ship in Q1

Total Revenue	Total Orders	Total Customers	Average Rating
\$124,714,086.32	1000	994	Okay (3.065)
Last Quarter Revenue	Last quarter Orders	Average Days to Ship	% Good Feedback
\$23,346,779.63	199(orders)/306(quantity)	105 days	21.5%

## Business Recommendations

- To enhance New Wheels' performance, especially in the final quarter, consider focusing efforts on improving Q4 metrics across several areas. In Q4, average shipping times are at their peak, while both order volume and total quantity ordered reach their lowest levels. Additionally, revenue declines at a higher rate than the drop in order quantities between Q3 and Q4, suggesting an excessive use of discounts that could be impacting profitability.
- Chevrolet is the clear best-seller among vehicle brands, presenting a prime opportunity for targeted marketing initiatives. Ford follows at a solid second, with Toyota, Pontiac, and Dodge rounding out the top five. Promoting these top-performing brands could drive increased sales and enhance brand alignment with customer preferences.
- Customer reviews reflect a generally positive distribution, with 'Very Good' and 'Good' ratings holding the highest percentage shares. However, there's a concerning trend: the proportion of 'Very Bad', 'Bad', and 'Okay' reviews has steadily increased each quarter. By Q4, negative feedback has reached its highest levels, coinciding with a decrease in positive feedback. This trend is likely tied to shipping delays and the Q4 performance slump. Addressing shipping inefficiencies in the latter half of the year could help mitigate dissatisfaction, as shipping times in Q3 and Q4 are over twice as long as in Q1 and Q2, with Q4 averaging more than triple Q1's shipping duration. Reducing these delays could have a substantial impact on customer satisfaction, particularly during the critical end-of-year period.