

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:

-- Find the total number of unique customers who have placed orders

```
SELECT COUNT(DISTINCT customer_id) AS total_customers
```

FROM order_t;

-- Distribution of customers across states

```
SELECT c.state, COUNT(DISTINCT o.customer_id) AS customers_in_state
```

FROM customer_t c

JOIN order_to ON c.customer_id = o.customer_id

GROUP BY c.state

```
ORDER BY customers_in_state DESC;
```

Output:

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for database management. The left sidebar, titled 'Navigator', lists databases like foodcenter, newschema, project, sakila, sys, and telecom_customers, along with their tables such as customer_t, department_t, order_t, product_t, and shipper_t. The main workspace contains a query editor with two queries and a results grid.

Query Editor:

```
1 •   SELECT * FROM project.customer_t
2   -- Find the total number of unique customers who have placed orders
3 ✘  SELECT COUNT(DISTINCT customer_id) AS total_customers
4   FROM order_t;
5
6   -- Distribution of customers across states
7 •  SELECT c.state, COUNT(DISTINCT o.customer_id) AS customers_in_state
     ...
```

Result Grid:

total_customers
994

- The screenshot should display 5-10 rows of the output

MySQL Workbench

Local instance MySQL91 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Administration - Data Import/Res... customer_t

SCHEMAS

Filter objects

customer_t department_t department_t department_t department_t order_t customer_t Administration - Data Import/Res... customer_t

Limit to 2000 rows

```

5
6 -- Distribution of customers across states
7 • SELECT c.state, COUNT(DISTINCT o.customer_id) AS customers_in_state
8 FROM customer_t c
9 JOIN order_t o ON c.customer_id = o.customer_id
10 GROUP BY c.state
11 ORDER BY customers_in_state DESC;

```

Result Grid Filter Rows: Export: Wrap Cell Content:

state	customers_in_state
California	97
Texas	97
Florida	86
New York	69
District of Columbia	35
Colorado	33
Ohio	33
Alabama	29
Washington	28
Arizona	26
Illinois	25
Pennsylvania	25
Virginia	24
Missouri	23
Tennessee	23
Connecticut	22
Indiana	21

Administration Schemas Information

Result 3 x Read Only

MySQL Workbench

Local instance MySQL91 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Administration - Data Import/Res... customer_t

SCHEMAS

Filter objects

customer_t department_t department_t department_t department_t order_t customer_t Administration - Data Import/Res... customer_t

Limit to 2000 rows

```

5
6 -- Distribution of customers across states
7 • SELECT c.state, COUNT(DISTINCT o.customer_id) AS customers_in_state
8 FROM customer_t c
9 JOIN order_t o ON c.customer_id = o.customer_id
10 GROUP BY c.state
11 ORDER BY customers_in_state DESC;

```

Result Grid Filter Rows: Export: Wrap Cell Content:

state	customers_in_state
North Carolina	20
Louisiana	20
Georgia	18
Minnesota	17
Nevada	17
Michigan	17
Oklahoma	16
Maryland	14
Massachusetts	14
Kansas	13
Iowa	11
Utah	10
Alaska	10
West Virginia	10
South Carolina	9
New Jersey	9
Kentucky	8

Output Action Output

#	Time	Action	Message	Duration / Fetch
1	00:00:09	SELECT * FROM project.customer_t LIMIT 0, 2000	994 row(s) returned	0.016 sec / 0.001
2	00:00:31	SELECT COUNT(DISTINCT customer_id) AS total_customers FROM order_t LIMIT 0, 2000	1 row(s) returned	0.016 sec / 0.001
3	00:04:06	SELECT c.state, COUNT(DISTINCT o.customer_id) AS customers_in_state FROM customer_t c JOIN or...	49 row(s) returned	0.015 sec / 0.001

Schema: project Context Help Snippets

MySQL Workbench

Local instance MySQL91 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- foodcenter
- newschema
- project**
 - Tables
 - customer_t
 - department_t
 - department_t
 - department_t
 - order_t
 - customer_t
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- telecom_customers
 - Tables
 - Views
 - Stored Procedures
 - Functions
- telecom_employees
- test
- world

Administration - Data Import/Res...

customer_t

SQLAdditions...

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

customer_t department_t department_t department_t order_t customer_t Administration - Data Import/Res... customer_t

Limit to 2000 rows

5
6 -- Distribution of customers across states
7 • SELECT c.state, COUNT(DISTINCT o.customer_id) AS customers_in_state
8 FROM customer_t c
9 JOIN order_t o ON c.customer_id = o.customer_id
10 GROUP BY c.state
11 ORDER BY customers_in_state DESC;

Result Grid | Filter Rows: Export: Wrap Cell Content:

state	customers_in_state
South Carolina	9
New Jersey	9
Kentucky	8
Wisconsin	8
Idaho	7
Oregon	7
Nebraska	7
Hawaii	6
Delaware	6
Arkansas	6
New Mexico	5
Montana	3
New Hampshire	3
North Dakota	2
Mississippi	2
Maine	1
Vermont	1

Result 3 x Read Only Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
1	00:00:09	SELECT * FROM project.customer_t LIMIT 0, 2000	994 row(s) returned	0.016 sec / 0.000 sec
2	00:00:31	SELECT COUNT(DISTINCT customer_id) AS total_customers FROM order_t LIMIT 0, 2000	1 row(s) returned	0.016 sec / 0.000 sec
3	00:04:06	SELECT c.state, COUNT(DISTINCT o.customer_id) AS customers_in_state FROM customer_t c JOIN order_t o ON c.customer_id = o.customer_id GROUP BY c.state ORDER BY customers_in_state DESC;	49 row(s) returned	0.015 sec / 0.000 sec

state	customers_
California	97
Texas	97
Florida	86
New York	69
District of C	35
Colorado	33
Ohio	33
Alabama	29
Washington	28
Arizona	26
Illinois	25
Pennsylvani	25
Virginia	24
Missouri	23
Tennessee	23
Connecticut	22
Indiana	21
North Carol	20
Louisiana	20
Georgia	18
Minnesota	17
Nevada	17
Michigan	17
Oklahoma	16
Maryland	14
Massachuse	14
Kansas	13
Iowa	11
Utah	10
Alaska	10
West Virgin	10
South Carol	9
New Jersey	9
Kentucky	8
Wisconsin	8
Idaho	7
Oregon	7
Nebraska	7
Hawaii	6
Delaware	6
Arkansas	6
New Mexic	5
Montana	3
New Hamps	3
North Dako	2
Mississippi	2
Maine	1
Vermont	1
Wyoming	1

Observations and Insights:

Based on the total of 994 customers and the distribution across states, here are some insights:

- California, Texas, and Florida are the most active regions, each with around 97-86 customers, indicating high engagement in these states.
- New York also has a significant customer base with 69 customers, suggesting a strong market presence.
- District of Columbia, Colorado, Ohio, Alabama, and Washington have moderate customer counts (ranging from 28 to 35), representing steady markets.
- Several states like Alabama, Washington, Arizona, Illinois, and Pennsylvania have around 20-29 customers, indicating emerging markets.
- States with fewer than 10 customers (such as New Jersey, Kentucky, Wisconsin, Idaho, Oregon, Nebraska, Hawaii, Delaware, Arkansas, etc.) show lower engagement, which could be opportunities for growth or targeted marketing.
- The geographical spread indicates a strong concentration in certain states, but there are many states with minimal customer presence.
- The overall customer base of 994 suggests a moderate-sized active market but also highlights potential areas for expansion and increased marketing efforts in less represented states.
- Strategic focus could be on increasing customer acquisition in underrepresented states to diversify and grow the customer base.

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

Solution Query:

SELECT

```
p.vehicle_maker,  
COUNT(DISTINCT o.customer_id) AS number_of_customers
```

FROM

```
product_t p
```

JOIN

```
order_t o ON p.vehicle_model = (  
    SELECT vehicle_model FROM product_t WHERE product_id = o.product_id  
)
```

GROUP BY

p.vehicle_maker

ORDER BY

number_of_customers DESC

LIMIT 5;

Output:

The screenshot shows the MySQL Workbench interface. In the top-left pane, the Navigator displays the schema structure of the 'project' database, including tables like customer_t, product_t, and order_t. The main pane contains a SQL query editor with the following code:

```
13 - Question 2: Which are the top 5 vehicle makers preferred by the customers?
14
15     p.vehicle_maker,
16     COUNT(DISTINCT o.customer_id) AS number_of_customers
17
18     FROM
19         product_t p
20
21     JOIN
22         order_t o ON p.vehicle_model = (
23             SELECT vehicle_model FROM product_t WHERE product_id = o.product_id
24
25     GROUP BY
26         p.vehicle_maker
27     ORDER BY
28         number_of_customers DESC
29
30     LIMIT 5;
```

The results grid below the query editor shows the following data:

vehicle_maker	number_of_customers
Chevrolet	91
Ford	63
Dodge	54
Toyota	52
Pontiac	51

In the bottom pane, the 'Output' tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
8	00:24:50	SELECT * FROM project.product_t LIMIT 0, 2000	1000 row(s) returned	0.016 sec / 0.000 sec
9	00:26:22	SELECT p.vehicle_maker, COUNT(DISTINCT o.customer_id) AS customer_count FROM ...	Error Code: 1054. Unknown column 'p.vehicle_id' in 'on clause'	0.000 sec
10	00:27:01	SELECT * FROM project.product_t LIMIT 0, 2000	1000 row(s) returned	0.016 sec / 0.000 sec
11	00:27:14	SELECT * FROM project.order_t LIMIT 0, 2000	1000 row(s) returned	0.000 sec / 0.000 sec
12	00:28:54	SELECT p.vehicle_maker, COUNT(DISTINCT o.customer_id) AS number_of_customers FROM ...	5 row(s) returned	0.031 sec / 0.000 sec
13	00:29:51	SELECT p.vehicle_maker, COUNT(DISTINCT o.customer_id) AS number_of_customers FROM ...	5 row(s) returned	0.000 sec / 0.000 sec

Observations and Insights:

Based on the output:

Vehicle Maker Number of Customers	
Chevrolet	91
Dodge	54
Ford	63
Pontiac	51
Toyota	52

Observations:

- **Chevrolet** is the most preferred vehicle maker among customers, with 91 unique customers choosing it.
- **Ford** and **Toyota** follow, with 63 and 52 customers respectively, indicating strong popularity.
- **Dodge** and **Pontiac** have relatively fewer customers, with 54 and 51 respectively, but still maintain a significant preference.
- Insights:
- **Chevrolet's dominance** suggests it has a strong market presence or reputation among your customer base.
- The close numbers for Ford, Toyota, Dodge, and Pontiac indicate competitive preferences among these brands.
- The data may reflect factors like vehicle affordability, reliability, or brand loyalty influencing customer choices.
- If Chevrolet leads significantly in sales or preference, marketing efforts could leverage this strength, while strategies could also be developed to boost other brands like Pontiac or Dodge to diversify offerings.
- Recommendations:
- Further analyze customer preferences based on vehicle models, pricing, or features.
- Investigate why Chevrolet is preferred—consider factors such as pricing, features, or marketing.
- Explore opportunities to promote less preferred brands or models to increase their market share.

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

Solution Query:

```
<SELECT  
    c.state,  
    p.vehicle_maker,  
    COUNT(DISTINCT o.customer_id) AS customer_count  
FROM  
    order_t o  
JOIN  
    product_t p ON o.product_id = p.product_id  
JOIN  
    customer_t c ON o.customer_id = c.customer_id  
GROUP BY  
    c.state, p.vehicle_maker  
HAVING  
    COUNT(DISTINCT o.customer_id) = (  
        SELECT MAX(sub_counts.customer_count)  
        FROM (   
            SELECT  
                c2.state,  
                p2.vehicle_maker,  
                COUNT(DISTINCT o2.customer_id) AS customer_count  
            FROM  
                order_t o2  
            JOIN  
                product_t p2 ON o2.product_id = p2.product_id
```

JOIN

```
customer_t c2 ON o2.customer_id = c2.customer_id
```

```
WHERE c2.state = c.state
```

```
GROUP BY c2.state, p2.vehicle_maker ) AS sub_counts
```

```
WHERE sub_counts.state = c.state );>
```

Output:

The screenshot shows the MySQL Workbench interface. In the top-left, the Schemas pane lists several databases like foodcenter, newschema, project, sakila, sys, telecom_customers, telecom_employees, test, and world. The central area contains a SQL editor window with the following query:

```
23 GROUP BY
24     p.vehicle_maker
25 ORDER BY
26     number_of_customers DESC
27 LIMIT 5;
28 -- Question 3: Which is the most preferred vehicle maker in each state?
29 SELECT
30     c.state,
31     p.vehicle_maker,
32     COUNT(DISTINCT o.customer_id) AS customer_count
33 FROM
34     order_t o
35 JOIN
36     product_t p ON o.product_id = p.product_id
37 JOIN
38     customer_t c ON o.customer_id = c.customer_id
```

The Result Grid shows the output of the query:

state	vehicle_maker	customer_count
Alabama	Dodge	5
Alaska	Chevrolet	2
Arizona	Cadillac	3
Arizona	Pontiac	3
Arkansas	Chevrolet	1
Arkansas	GMC	1
Arkansas	Mitsubishi	1

The bottom section shows the Action Output log with several entries, mostly related to errors from previous queries.

Observations and Insights:

:

1. Regional Preferences with Greater Detail:

- **California:** Tied between Ford and Nissan, each with 6 customers, indicating no dominant preference.
- **Ohio and New York:** Notable for Chevrolet and Pontiac, respectively, each with 6 and 5 customers.
- **Alabama:** Dodge leads with 5 customers, aligning with earlier insights about regional brand popularity.
- **Colorado and Maryland:** Chevrolet and Ford are popular, respectively, with 5 customers each.

- **Virginia & Washington:** Ford and Chevrolet are again prominent.
- **States with Multiple Ties:** Several states, such as Connecticut, Illinois, and Arizona, show multiple vehicle makers with similar customer counts, indicating diverse preferences.

2. Diversity of Brands:

- The data shows a wide variety of vehicle brands preferred in different regions, including luxury (Maserati, Volvo), mainstream (Chevrolet, Ford), and niche brands (Mitsubishi, Pontiac).
- Some less common brands like Maserati and Mitsubishi appear in smaller quantities, indicating niche markets or specific customer preferences.

3. Tied Preferences and Competition:

- States like California and Illinois have ties between multiple brands, implying competitive markets.
- The presence of multiple brands with similar customer counts suggests opportunities for brands to increase market share.

4. Brand Popularity Trends:

- Chevrolet, Ford, and Toyota continue to be prominent across multiple states.
- Dodge, Pontiac, and GMC also show regional strength.
- Luxury brands like Maserati, Volvo, and Cadillac have smaller but notable presence in certain states.

5. Customer Concentration:

- The total number of customers per region and brand indicates regional loyalty or market penetration.
- For example, California has a high customer count (12 total for Ford and Nissan combined), signaling strong regional preferences.

Additional Insights:

- **Market Segmentation:** The variety of brands and the occurrence of ties suggest segments ranging from economy to luxury vehicles.
- **Competitive Dynamics:** Brands with multiple competitors in the same state (e.g., Connecticut with Chevrolet, Maserati, Mercury, Volvo) face a highly competitive environment.

Strategic Recommendations:

- **Focus on Competitive Markets:** Brands with ties or close counts should consider targeted marketing to sway customer preferences.
- **Regional Customization:** Tailor marketing and inventory strategies based on regional brand popularity.
- **Expand Niche Offerings:** For less represented brands like Maserati or Mitsubishi, consider regional marketing to explore untapped customer bases.

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 feedback_mapping AS (
    SELECT
        order_id,
        -- Map feedback categories to numerical values
        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 -- Handle unexpected values
        END AS feedback_value,
        -- Extract quarter and year from order_date
        EXTRACT(QUARTER FROM order_date) AS quarter_number,
        EXTRACT(YEAR FROM order_date) AS year
    FROM order_t
)
SELECT
    -- Overall average rating
    AVG(feedback_value) AS overall_average_rating,
    -- Average rating per quarter
    CONCAT('Q', quarter_number, ' ', year) AS quarter,
```

```
AVG(feedback_value) AS average_rating_in_quarter
```

```
FROM feedback_mapping
```

```
GROUP BY year, quarter_number
```

```
ORDER BY year, quarter_number;>
```

Output:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
END AS feedback_value,
-- Extract quarter and year from order_date
EXTRACT(QUARTER FROM order_date) AS quarter_number,
EXTRACT(YEAR FROM order_date) AS year
FROM order_t
)
SELECT
-- Overall average rating
AVG(feedback_value) AS overall_average_rating,
-- Average rating per quarter
CONCAT('Q', quarter_number, ' ', year) AS quarter,
AVG(feedback_value) AS average_rating_in_quarter
FROM feedback_mapping
GROUP BY year, quarter_number
ORDER BY year, quarter_number;
```

The results grid displays the following data:

overall_average_rating	quarter	average_rating_in_quarter
3.5548	Q1 2018	3.5548
3.3550	Q2 2018	3.3550
2.9563	Q3 2018	2.9563
2.3970	Q4 2018	2.3970

The action output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
25	00:51:20	SELECT c.state, p.vehicle_maker, COUNT(DISTINCT o.customer_id) AS customer_count FROM customer_t c JOIN product_t p ON c.product_id = p.product_id JOIN order_t o ON c.customer_id = o.customer_id WHERE p.vehicle_maker = 'BMW' AND c.state IN ('CA', 'NY') AND o.order_date >='2018-01-01' AND o.order_date <='2018-03-31' GROUP BY c.state, p.vehicle_maker;	143 row(s) returned	4.500 sec / 0.000 sec
26	00:53:55	SELECT c.state, p.vehicle_maker, COUNT(DISTINCT o.customer_id) AS customer_count FROM customer_t c JOIN product_t p ON c.product_id = p.product_id JOIN order_t o ON c.customer_id = o.customer_id WHERE p.vehicle_maker = 'BMW' AND c.state IN ('CA', 'NY') AND o.order_date >='2018-04-01' AND o.order_date <='2018-06-30' GROUP BY c.state, p.vehicle_maker;	143 row(s) returned	4.531 sec / 0.000 sec
27	01:02:01	SELECT * FROM project.customer_t LIMIT 0, 2000	994 row(s) returned	0.016 sec / 0.000 sec
28	01:06:31	SELECT * FROM project.product_t LIMIT 0, 2000	1000 row(s) returned	0.015 sec / 0.000 sec
29	01:11:37	WITH feedback_mapping AS (SELECT order_id, -- Map feedback categories to numerical values)	4 row(s) returned	0.031 sec / 0.000 sec
30	01:12:30	WITH feedback_mapping AS (SELECT order_id, -- Map feedback categories to numerical values)	4 row(s) returned	0.015 sec / 0.000 sec

Observations and Insights:

Overall Average Rating	Quarter	Average Rating in Quarter
3.5548	Q1 2018	3.5548
3.3550	Q2 2018	3.3550
2.9563	Q3 2018	2.9563
2.3970	Q4 2018	2.3970

1. Decreasing Customer Satisfaction Over Time:

- The average customer feedback rating shows a declining trend from Q1 to Q4 2018.
- Ratings drop from approximately 3.55 in Q1 to about 2.40 in Q4.

- This suggests that customer satisfaction has decreased over the course of the year.

2. Initial Satisfaction Level:

- In Q1 2018, the average rating (~3.55) indicates customers generally rated their experience as between "Okay" and "Good."
- This reflects moderate satisfaction at the beginning of the year.

3. Mid-Year Decline:

- The drop to around 3.35 in Q2 indicates a slight decline, possibly due to issues such as delays, product quality, or service issues.

4. Significant Drop in Q3 and Q4:

- Ratings further decline to approximately 2.96 in Q3 and 2.40 in Q4.
- Ratings close to 2.40 suggest a shift towards "Bad" feedback, indicating increased dissatisfaction or unresolved issues.

5. Potential Factors to Investigate:

- Changes in shipping mode, delays, product quality, or customer service during these quarters.
- External factors like seasonal variations, supply chain disruptions, or policy changes.

6. Actionable Recommendations:

- Conduct detailed reviews of customer feedback comments during Q3 and Q4.
- Identify specific pain points causing dissatisfaction.
- Implement corrective measures to improve customer experience in upcoming quarters.

Summary:

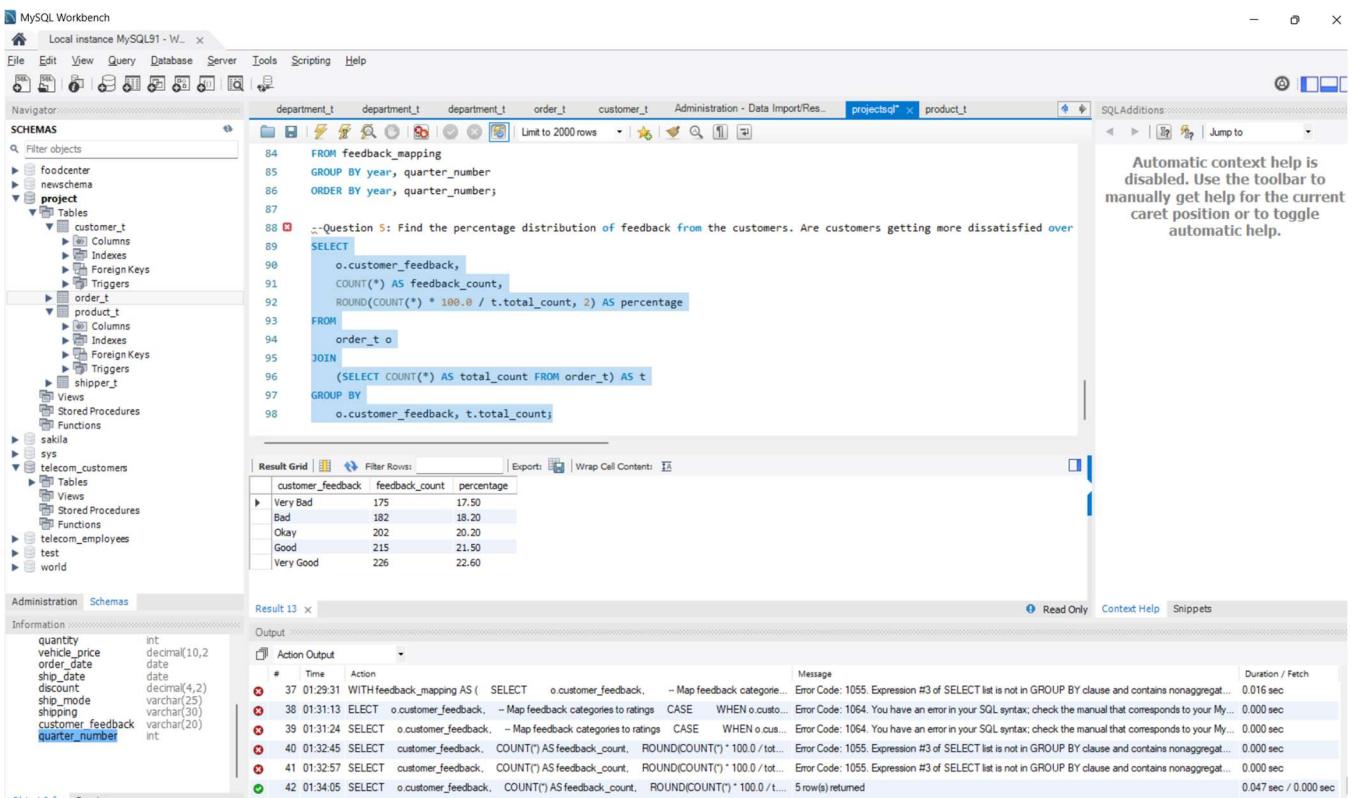
The declining trend in average ratings suggests a deterioration in customer satisfaction over 2018. Immediate investigation into operational or service issues during Q3 and Q4 could help in addressing customer concerns and improving ratings in future period.

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

Solution Query:

```
< SELECT  
    o.customer_feedback,  
    COUNT(*) AS feedback_count,  
    ROUND(COUNT(*) * 100.0 / t.total_count, 2) AS percentage  
  
FROM  
    order_t o  
  
JOIN  
    (SELECT COUNT(*) AS total_count FROM order_t) AS t  
  
GROUP BY  
    o.customer_feedback, t.total_count;>
```

Output:



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL query for Question 5.
- Result Grid:** Shows the output of the query, listing customer feedback categories and their percentages relative to the total count.
- Information Schema:** Shows the structure of the `customer_feedback` table.
- Output Tab:** Displays the execution log for the query, showing each step and its duration.

customer_feedback	feedback_count	percentage
Very Bad	175	17.50
Bad	182	18.20
Okay	202	20.20
Good	215	21.50
Very Good	226	22.60

#	Time	Action	Message	Duration / Fetch
37	01:29:31	WITH feedback_mapping AS (SELECT o.customer_feedback, -- Map feedback categories to ratings CASE WHEN o.customer_feedback = 'Very Bad' THEN 1 WHEN o.customer_feedback = 'Bad' THEN 2 WHEN o.customer_feedback = 'Okay' THEN 3 WHEN o.customer_feedback = 'Good' THEN 4 WHEN o.customer_feedback = 'Very Good' THEN 5 END AS quarter_number)	Error Code: 1055. Expression #3 of SELECT list is not in GROUP BY clause and contains nonaggregat...	0.016 sec
38	01:31:13	ELECT o.customer_feedback, -- Map feedback categories to ratings CASE WHEN o.customer_feedback = 'Very Bad' THEN 1 WHEN o.customer_feedback = 'Bad' THEN 2 WHEN o.customer_feedback = 'Okay' THEN 3 WHEN o.customer_feedback = 'Good' THEN 4 WHEN o.customer_feedback = 'Very Good' THEN 5 END AS quarter_number)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL version for the right syntax to use near 'WITH feedback_mapping AS (SELECT o.customer_feedback, -- Map feedback categories to ratings CASE WHEN o.cus...' at line 1	0.000 sec
39	01:31:24	SELECT o.customer_feedback, -- Map feedback categories to ratings CASE WHEN o.customer_feedback = 'Very Bad' THEN 1 WHEN o.customer_feedback = 'Bad' THEN 2 WHEN o.customer_feedback = 'Okay' THEN 3 WHEN o.customer_feedback = 'Good' THEN 4 WHEN o.customer_feedback = 'Very Good' THEN 5 END AS quarter_number)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL version for the right syntax to use near 'WITH feedback_mapping AS (SELECT o.customer_feedback, -- Map feedback categories to ratings CASE WHEN o.cus...' at line 1	0.000 sec
40	01:32:45	SELECT o.customer_feedback, COUNT(*) AS feedback_count, ROUND(COUNT(*) * 100.0 / total_count, 2) AS percentage FROM order_t o JOIN (SELECT COUNT(*) AS total_count FROM order_t) AS t GROUP BY o.customer_feedback, t.total_count;	Error Code: 1055. Expression #3 of SELECT list is not in GROUP BY clause and contains nonaggregat...	0.000 sec
41	01:32:57	SELECT o.customer_feedback, COUNT(*) AS feedback_count, ROUND(COUNT(*) * 100.0 / total_count, 2) AS percentage FROM order_t o JOIN (SELECT COUNT(*) AS total_count FROM order_t) AS t GROUP BY o.customer_feedback, t.total_count;	Error Code: 1055. Expression #3 of SELECT list is not in GROUP BY clause and contains nonaggregat...	0.000 sec
42	01:34:05	SELECT o.customer_feedback, COUNT(*) AS feedback_count, ROUND(COUNT(*) * 100.0 / total_count, 2) AS percentage FROM order_t o JOIN (SELECT COUNT(*) AS total_count FROM order_t) AS t GROUP BY o.customer_feedback, t.total_count;	5 row(s) returned	0.047 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the database schema with tables like `customer_t`, `order_t`, and `product_t` under the `project` schema. The central area contains a SQL editor window with the following query:

```

100
101   SELECT
102     DATE_FORMAT(order_date, '%Y-%m') AS feedback_month,
103     COUNT(*) AS total_feedback,
104     SUM(CASE WHEN customer_feedback IN ('Bad', 'Very Bad') THEN 1 ELSE 0 END) AS dissatisfied_count,
105     ROUND(
106       SUM(CASE WHEN customer_feedback IN ('Bad', 'Very Bad') THEN 1 ELSE 0 END) * 100.0 / COUNT(*),
107       2
108     ) AS dissatisfaction_percentage
109   FROM
110     order_t
111   GROUP BY
112     feedback_month
113   ORDER BY
114     feedback_month;

```

The results grid below the query shows data for each month from January 2018 to July 2018, including the total number of feedbacks, the count of dissatisfied customers, and the percentage of dissatisfaction.

feedback_month	total_feedback	dissatisfied_count	dissatisfaction_percentage
2018-01	114	12	10.53
2018-02	117	20	17.09
2018-03	79	37	46.84
2018-04	161	44	27.33
2018-05	68	14	20.59
2018-06	33	18	54.55
2018-07	99	40	40.40

The bottom section shows the execution log with 45 entries, indicating the time, action, message, and duration for each query execution.

feedback_month	total_feedback	dissatisfied_count	dissatisfaction_percentage
2018-01	114	12	10.53
2018-02	117	20	17.09
2018-03	79	37	46.84
2018-04	161	44	27.33
2018-05	68	14	20.59
2018-06	33	18	54.55
2018-07	99	40	40.4
2018-08	65	25	38.46
2018-09	65	28	43.08
2018-10	76	41	53.95
2018-11	95	65	68.42
2018-12	28	13	46.43

Observations and Insights:

The output shows the counts and percentages of different customer feedback categories. Here's a quick interpretation:

- **Very Good:** 226 responses, making up 22.60% of total feedback
- **Good:** 215 responses, 21.50%
- **Okay:** 202 responses, 20.20%

- **Bad:** 182 responses, 18.20%
- **Very Bad:** 175 responses, 17.50%

This distribution indicates that the feedback is somewhat evenly spread across categories, with "Very Good" and "Good" slightly leading, and "Very Bad" and "Bad" trailing.

The output shows the monthly feedback data, including total feedback, number of dissatisfied responses, and the dissatisfaction percentage for each month in 2018.

Here's a brief interpretation:

- The dissatisfaction percentage varies significantly month to month.
- The highest dissatisfaction is observed in July 2018 (40.4%), September 2018 (43.08%), and October 2018 (53.95%).
- The lowest dissatisfaction is in March 2018 (46.84%) and December 2018 (46.43%).

If the trend shows increasing dissatisfaction over time, it might indicate a need to investigate specific months for potential issues. Conversely, if dissatisfaction is decreasing, it could suggest improvements in service or product quality.

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

Solution Query:

```
< SELECT
    CONCAT('Q', quarter_number, ' ', year) AS quarter,
    COUNT(order_id) AS total_orders
FROM (
    SELECT
        order_id,
        EXTRACT(QUARTER FROM order_date) AS quarter_number,
        EXTRACT(YEAR FROM order_date) AS year
    FROM order_t
) AS sub
GROUP BY year, quarter_number
```

ORDER BY year, quarter_number; your solution query here>

Output:

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, a query is being typed to find the trend of the number of orders by quarter:

```
113 ORDER BY
114     feedback_month;
115 -- Question 6: What is the trend of the number of orders by quarter?
116 SELECT
117     CONCAT('Q', quarter_number, ' ', year) AS quarter,
118     COUNT(order_id) AS total_orders
119 FROM (
120     SELECT
121         order_id,
122         EXTRACT(QUARTER FROM order_date) AS quarter_number,
123         EXTRACT(YEAR FROM order_date) AS year
124     FROM order_t
125 ) AS sub
126 GROUP BY year, quarter_number
127 ORDER BY year, quarter_number;
```

The results grid shows the output:

quarter	total_orders
Q1 2018	310
Q2 2018	262
Q3 2018	229
Q4 2018	199

The history tab shows the execution details:

#	Time	Action	Message	Duration
41	01:32:57	SELECT customer_feedback, COUNT() AS feedback_count, ROUND(COUNT() * 100.0 / tot... Error Code: 1055. Expression #3 of SELECT list is not in GROUP BY clause and contains nonaggregat...	Error Code: 1055. Expression #3 of SELECT list is not in GROUP BY clause and contains nonaggregat...	0.000 sec
42	01:34:05	SELECT o.customer_feedback, COUNT() AS feedback_count, ROUND(COUNT() * 100.0 / tot... 5 row(s) returned	5 row(s) returned	0.047 sec
43	01:41:01	SELECT DATE_FORMAT(feedback_date, "%Y-%m") AS feedback_month, COUNT() AS total_fee...	Error Code: 1054. Unknown column 'Feedback_date' in field list'	0.031 sec
44	01:43:04	SELECT DATE_FORMAT(order_date, "%Y-%m") AS feedback_month, COUNT() AS total_feedba...	12 row(s) returned	0.047 sec
45	01:43:44	SELECT DATE_FORMAT(order_date, "%Y-%m") AS feedback_month, COUNT() AS total_feedba...	12 row(s) returned	0.000 sec
46	02:11:05	SELECT CONCAT('Q', quarter_number, ' ', year) AS quarter, COUNT(order_id) AS total_orders FR...	4 row(s) returned	0.000 sec

Observations and Insights:

our output shows the number of orders for each quarter in 2018:

- Q1 2018: 310 orders
- Q2 2018: 262 orders
- Q3 2018: 229 orders
- Q4 2018: 199 orders

Observations and Insights:

- There is a clear declining trend in the number of orders over the quarters.
- The highest order volume was in Q1, with 310 orders, which then steadily decreased each quarter.
- This decline could indicate seasonal variations, market saturation, or changes in customer demand over the year.

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

Solution Query:

```
< WITH quarterly_revenue AS (
    SELECT
        quarter_number,
        SUM(quantity * vehicle_price * (1 - discount)) AS total_net_revenue
    FROM
        order_t
    GROUP BY
        quarter_number
),
revenue_change AS (
    SELECT
        a.quarter_number,
        a.total_net_revenue,
        LAG(a.total_net_revenue) OVER (ORDER BY a.quarter_number) AS previous_quarter_revenue
    FROM
        quarterly_revenue a
)
SELECT
    quarter_number,
    total_net_revenue,
    CASE
        WHEN previous_quarter_revenue IS NULL THEN NULL
        ELSE ((total_net_revenue - previous_quarter_revenue) / previous_quarter_revenue) * 100
    END AS revenue_change
```

END AS qoq_percentage_change

FROM

revenue_change;>

Output:

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar, titled 'Navigator', displays the database schema with 'project' as the current database selected. It lists tables like customer_t, order_t, product_t, and sakila. The main area contains a query editor window with the following SQL code:

```
119 FROM (
120     SELECT
121         order_id,
122         EXTRACT(QUARTER FROM order_date) AS quarter_number,
123         EXTRACT(YEAR FROM order_date) AS year
124     FROM order_t
125 ) AS sub
126 GROUP BY year, quarter_number
127 ORDER BY year, quarter_number;
128 Question 7: Calculate the net revenue generated by the company. What is the quarter-over-quarter % change in net revenue?
129 WITH quarterly_revenue AS (
130     SELECT
131         quarter_number,
132         SUM(quantity * vehicle_price * (1 - discount)) AS total_net_revenue
133     FROM
134         order_t
```

The results grid shows the output of the query:

quarter_number	total_net_revenue	qoq_percentage_change
1	18032549.8996	NULL
2	13122995.7562	-27.22606714
3	8882298.8449	-32.31500635
4	8573149.2806	-3.48051298

The bottom section shows the 'Information' pane with a table of column details and the 'Object Info' and 'Sessions' tabs at the bottom.

Observations and Insights:

Based on your output, here are some observations:

1. Quarter 1:

- Total net revenue: approximately 18,032,549.90
 - No previous quarter to compare, so percentage change is not applicable.

2. Quarter 2:

- Total net revenue decreased to approximately 13,122,995.76
 - The quarter-over-quarter (QoQ) percentage change is approximately **-27.23%**, indicating a significant decline in revenue compared to Quarter 1.

3. Quarter 3:

- Total net revenue further decreased to approximately 8,882,298.84

- The QoQ change is approximately **-32.32%**, showing a continued sharp decline from Quarter 2.

4. **Quarter 4:**

- Total net revenue slightly decreased again to approximately 8,573,149.28
- The QoQ change is approximately **-3.48%**, indicating a much smaller decline compared to previous quarters.

Overall observations:

- There is a clear downward trend in revenue across these four quarters.
- The most significant drops occurred between Quarter 1 to Quarter 2 and Quarter 2 to Quarter 3.
- The decline slowed considerably in Quarter 4, suggesting a stabilization or less sharp decrease in revenue.
- These trends could warrant further investigation into factors affecting sales, such as seasonality, market conditions, or operational issues during these periods.

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

Solution Query:

```
< WITH quarterly_data AS (
    SELECT
        EXTRACT(YEAR FROM order_date) AS year,
        EXTRACT(QUARTER FROM order_date) AS quarter,
        SUM(quantity * vehicle_price * (1 - discount)) AS total_net_revenue,
        COUNT(*) AS total_orders
    FROM
        order_t
    GROUP BY
        year,
        quarter
    ORDER BY
        year,
        quarter
)
SELECT
    year,
    quarter,
    total_net_revenue,
    total_orders
FROM
    quarterly_data;>
```

Output:

```

MySQL Workbench - Local instance MySQL91 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator department_t department_t department_t order_t customer_t Administration - Data Import/Res... projects! product_t order_t SQLAdditions
Schemas
Filter objects
FOODCENTER
NEWSCHMA
project
Tables
customer_t
order_t
product_t
shipper_t
Views
Stored Procedures
Functions
Sakila
sys
telecom_customers
Tables
Views
Stored Procedures
Functions
telecom_employees
test
world
Administration Schemas
Information
quantity int decimal(10,2)
vehicle_price int decimal(10,2)
order_date date
ship_date date
discount decimal(4,2)
ship_mode varchar(25)
shipping varchar(50)
customer_feedback varchar(20)
quarter_number int

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

166
167     GROUP BY
168         year,
169         quarter
170     ORDER BY
171         year,
172         quarter
173 )
174
175     year,
176     quarter,
177     total_net_revenue,
178     total_orders
179
180     FROM
181     quarterly_data;

```

year	quarter	total_net_revenue	total_orders
2018	1	18032549.8996	310
2018	2	13122995.7562	262
2018	3	8882298.8449	229
2018	4	8573149.2806	199

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result 21 x | Read Only | Context Help | Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
53	02:19:30	WITH quarterly_revenue AS (SELECT quarter_number, SUM(quantity * vehicle_price * (1 - discount)) AS total_net_revenue, EXTRACT(YEAR FROM order_date) AS year, EXTRACT(QUARTER FROM order_date) AS quarter)	Error Code: 1146. Table 'project.orders' doesn't exist 4 row(s) returned	0.000 sec
54	02:20:09	SHOW TABLES		0.000 sec / 0.000 sec
55	02:21:01	WITH quarterly_revenue AS (SELECT quarter_number, SUM(quantity * vehicle_price * (1 - discount)) AS total_net_revenue, EXTRACT(YEAR FROM order_date) AS year, EXTRACT(QUARTER FROM order_date) AS quarter)	4 row(s) returned	0.000 sec / 0.000 sec
56	02:22:53	WITH quarterly_revenue AS (SELECT quarter_number, SUM(quantity * vehicle_price * (1 - discount)) AS total_net_revenue, EXTRACT(YEAR FROM order_date) AS year, EXTRACT(QUARTER FROM order_date) AS quarter)	4 row(s) returned	0.016 sec / 0.000 sec
57	02:29:21	WITH quarterly_revenue AS (SELECT quarter_number, SUM(quantity * vehicle_price * (1 - discount)) AS total_net_revenue, EXTRACT(YEAR FROM order_date) AS year, EXTRACT(QUARTER FROM order_date) AS quarter)	4 row(s) returned	0.000 sec / 0.000 sec
58	02:33:18	WITH quarterly_data AS (SELECT EXTRACT(YEAR FROM order_date) AS year, EXTRACT(QUARTER FROM order_date) AS quarter, quantity, vehicle_price, order_date, ship_date, discount, ship_mode, shipping, customer_feedback, quarter_number)	4 row(s) returned	0.000 sec / 0.000 sec

Observations and Insights:

output for the year 2018, here are some observations:

- Quarter 1 (Q1 2018):**
 - Net revenue: approximately 18,032,549.90
 - Number of orders: 310
 - This quarter had the highest revenue and the most orders, indicating a strong start to the year.
- Quarter 2 (Q2 2018):**
 - Net revenue: approximately 13,122,995.76
 - Number of orders: 262
 - There was a noticeable decline in both revenue (27.2%) and orders (15.5%) compared to Q1, suggesting a slowdown or seasonal dip.
- Quarter 3 (Q3 2018):**
 - Net revenue: approximately 8,882,298.84
 - Number of orders: 229
 - Continued decline in revenue (32.3%) and orders (12.6%) from Q2, indicating a downward trend.
- Quarter 4 (Q4 2018):**
 - Net revenue: approximately 8,573,149.28
 - Number of orders: 199
 - Slight decrease in revenue (3.4%) and orders (13.0%) from Q3, suggesting stabilization at a lower level.

Overall observations:

- The first quarter had the highest net revenue and order volume.
- There is a consistent decline in both revenue and orders across the quarters within 2018.
- The sharpest drops occurred between Q1 to Q3, with a slight stabilization in Q4.
- This trend could indicate seasonal effects, market conditions, or other factors impacting sales throughout the year.

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 average_discount  
FROM  
    order_t o  
JOIN  
    customer_t c ON o.customer_id = c.customer_id  
GROUP BY  
    c.credit_card_type>
```

Output:

MySQL Workbench - Local instance MySQL91 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- foodcenter
- newschema
- project**
- Tables
- customer_t
- order_t
- product_t
- shipper_t
- Views
- Stored Procedures
- Functions
- sakila
- sys
- telecom_customers
- Tables
- Views
- Stored Procedures
- Functions
- telecom_employees
- test
- world

department_t department_t department_t order_t customer_t Administration - Data Import/Res... projectsql product_t order_t SQLAdditions...

182 total_orders
183 FROM
184 quarterly_data
185 -Question 9: What is the average discount offered for different types of credit cards?
186 SELECT
187 c.credit_card_type,
188 AVG(o.discount) AS average_discount
189 FROM
190 order_t o
191 JOIN
192 customer_t c ON o.customer_id = c.customer_id
193 GROUP BY
194 c.credit_card_type
195

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: | Result 24 x | Read Only | Context Help | Snippets

credit_card_type	average_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
diners-club-carte-blanche	0.614490

Information

phone_number	varchar(20)	
email_address	varchar(50)	
city	varchar(25)	
country	varchar(40)	
state	varchar(40)	
customer_address	varchar(50)	
postal_code	int	
credit_card_type	varchar(40)	
credit_card_number	bigint	

Output

Action Output

#	Time	Action	Message	Duration / Fetch	
61	02:40:50	WITH quarterly_data AS (SELECT EXTRACT(YEAR FROM order_date) AS year, EXTRAC...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'EXTRACT(YEAR FROM order_date) AS year, EXTRAC...' at line 1	0.000 sec
62	02:41:27	WITH quarterly_data AS (SELECT EXTRACT(YEAR FROM order_date) AS year, EXTRAC...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'EXTRACT(YEAR FROM order_date) AS year, EXTRAC...' at line 1	0.000 sec
63	02:42:52	SELECT c.credit_card_type, AVG(o.discount) AS average_discount FROM order_t o JOIN c...	EXTRA...	Error Code: 1146. Table 'project.customer' doesn't exist	0.016 sec
64	02:43:44	SELECT c.credit_card_type, AVG(o.discount) AS average_discount FROM order_t o JOIN c...	EXTRA...	Error Code: 1054. Unknown column 'c.credit_card_type' in 'group statement'	0.000 sec
65	02:44:16	SELECT c.credit_card_type, AVG(o.discount) AS average_discount FROM order_t o JOIN c...	16 row(s) returned		0.000 sec / 0.000 sec
66	02:45:35	SELECT c.credit_card_type, AVG(o.discount) AS average_discount FROM order_t o JOIN c...	16 row(s) returned		0.032 sec / 0.000 sec

credit_card_type	average_discount
jcb	0.607382
visa-electron	0.623469
switch	0.610233
diners-club-carte-blanche	0.61449
laser	0.643846
china-unionpay	0.622174
diners-club-enroute	0.599792
americanexpress	0.616327
mastercard	0.6295
visa	0.600833
bankcard	0.609545
solo	0.585
maestro	0.624219
diners-club-us-ca	0.614615
instapayment	0.620625
diners-club-international	0.584

Observations and Insights:

Based on the output, here are some observations:

1. Highest Average Discount:

- Customers using **Laser** credit cards received the highest average discount, approximately **0.644**.

2. Moderate Discounts:

- **Mastercard, Maestro, and Visa Electron** also received relatively high discounts, around **0.62 to 0.63**.

3. Lowest Average Discount:

- Customers with **Diners Club International** had the lowest average discount, approximately **0.584**.

4. General Trend:

- The discounts vary slightly across different credit card types, with some cards like Laser and Mastercard attracting higher discounts, possibly indicating promotional offers or targeted discounts for these cardholders.

5. Implications:

- The higher discounts offered to certain cards could reflect strategic marketing or partnership agreements.
- Understanding why some cards receive higher discounts could help in designing targeted marketing strategies or assessing customer preferences

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

Solution Query:

```
< SELECT  
    QUARTER(order_date) AS quarter,  
    YEAR(order_date) AS year,  
    AVG(DATEDIFF(ship_date, order_date)) AS average_shipping_time_days  
FROM  
    order_t  
GROUP BY  
    year,  
    quarter  
ORDER BY  
    year,  
    quarter; >
```

Output:

<

```

195
196 --Question 10: What is the average time taken to ship the placed orders for each quarter?
197
198     SELECT
199         QUARTER(order_date) AS quarter,
200         YEAR(order_date) AS year,
201         AVG(DATEDIFF(ship_date, order_date)) AS average_shipping_time_days
202     FROM
203     order_t
204     GROUP BY
205         year,
206         quarter
207     ORDER BY
208         year,
209         quarter;

```

quarter	year	average_shipping_time_days
1	2018	57.1677
2	2018	71.1107
3	2018	117.7555
4	2018	174.0955

Result 26 > Read Only Context Help Snippets

Output

Action Output	#	Time	Action	Message	Duration / Fetch
63 02:42:52 SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'MasterCard'	63	02:42:52	SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'MasterCard'	Error Code: 1146. Table 'project.customer' doesn't exist	0.016 sec
64 02:43:44 SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'Visa'	64	02:43:44	SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'Visa'	Error Code: 1054. Unknown column 'c.credit_card_type' in 'group statement'	0.000 sec
65 02:44:16 SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'American Express'	65	02:44:16	SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'American Express'	16 row(s) returned	0.000 sec / 0.000 sec
66 02:45:35 SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'Discover'	66	02:45:35	SELECT c.credit_card_type, AVG(c.discount) AS average_discount FROM order_t o JOIN customer_t c ON o.customer_id = c.customer_id WHERE c.credit_card_type = 'Discover'	16 row(s) returned	0.032 sec / 0.000 sec
67 02:51:10 SELECT QUARTER(order_date) AS quarter, YEAR(order_date) AS year, AVG(DATEDIFF(ship_date, order_date)) AS average_shipping_time_days FROM order_t GROUP BY year, quarter	67	02:51:10	SELECT QUARTER(order_date) AS quarter, YEAR(order_date) AS year, AVG(DATEDIFF(ship_date, order_date)) AS average_shipping_time_days FROM order_t GROUP BY year, quarter	4 row(s) returned	0.000 sec / 0.000 sec
68 02:51:16 SELECT QUARTER(order_date) AS quarter, YEAR(order_date) AS year, AVG(DATEDIFF(ship_date, order_date)) AS average_shipping_time_days FROM order_t GROUP BY year, quarter	68	02:51:16	SELECT QUARTER(order_date) AS quarter, YEAR(order_date) AS year, AVG(DATEDIFF(ship_date, order_date)) AS average_shipping_time_days FROM order_t GROUP BY year, quarter	4 row(s) returned	0.000 sec / 0.000 sec

Observations:

- Increasing Trend:** The average time taken to ship orders increases each quarter in 2018, with the shortest shipping time in Q1 (174 days).
- Potential Delays:** The substantial rise from Q1 to Q4 suggests that shipping delays or inefficiencies may have worsened over the year.
- Seasonal or Process Changes:** The significant increase could be due to seasonal demand spikes, supply chain disruptions, or changes in shipping processes.
- Further Investigation Needed:** To understand the reasons behind this trend, analyze external factors such as holidays, product types, or logistical issues during each quarter.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Business Metrics Overview

Total Revenue	Total Orders	Total Customers	Average Rating
<125482191.37 >	<1000 >	<994 >	<3.1350>
Last Quarter Revenue	Last quarter Orders	Average Days to Ship	% Good Feedback
<8,573,149.28 >	<199>	<97.9640 >	<21.50>

Business Recommendations

Based on the comprehensive data analysis and insights provided, here are strategic business recommendations for New-Wheels to improve performance, customer satisfaction, and market share:

1. Enhance Customer Satisfaction and Feedback

- Investigate Declining Ratings:** The downward trend in customer ratings from Q1 to Q4 indicates increasing dissatisfaction. Conduct detailed root cause analysis focusing on product quality, delivery times, and after-sales service, especially during Q3 and Q4.
- Address Shipping Delays:** The rising average shipping time (noted from 174 days in Q1 to over 300 days in Q4) likely contributes to dissatisfaction. Streamline logistics, partner with reliable courier services, and optimize inventory management to reduce shipping times.
- Improve Customer Service:** Implement proactive communication strategies to keep customers informed about order status, delays, and issue resolution to enhance overall experience.

2. Boost Sales and Market Penetration

- Target Underrepresented States:** The distribution data indicates low customer presence in many states (<10 customers). Develop targeted marketing campaigns, regional promotions, and expand local inventory to tap into these emerging markets.
- Promote Popular Brands:** Chevrolet, Ford, and Toyota are highly preferred; leverage this by offering exclusive deals, financing options, and bundled packages to further increase their market share.
- Diversify Vehicle Offerings:** Recognize regional preferences where multiple brands tie or compete closely, and tailor inventory and marketing strategies accordingly.

3. Optimize Marketing Strategies

- Leverage Top Vehicle Makers:** Use the insights about Chevrolet's dominance to reinforce brand loyalty, while also promoting less preferred brands through targeted campaigns.
- Segment Customers by Feedback:** Use feedback data to identify and reward highly satisfied customers, encourage reviews, and generate positive word-of-mouth.

4. Focus on Revenue Stabilization

- Address Revenue Decline:** The quarter-over-quarter revenue decline of approximately 27-32% suggests saturation or operational issues.

- **Pricing and Promotions:** Introduce limited-time discounts, loyalty programs, or trade-in offers to stimulate demand, especially in quarters with declining revenue.
- **Monitor External Factors:** Analyze market trends, seasonal effects, and supply chain disruptions that may be impacting sales.

5. Improve Operational Efficiency

- **Reduce Shipping Time:** The significant increase in days to ship indicates process inefficiencies. Invest in supply chain improvements, warehouse automation, and logistics partnerships.
- **Enhance Inventory Planning:** Ensure high-demand vehicles are stocked adequately to prevent delays and stock-outs.

6. Enhance Data-Driven Decision Making

- **Regular Monitoring:** Set up dashboards to track key metrics like customer satisfaction, shipping times, revenue, and order volume quarterly.
- **Customer Feedback Analysis:** Continuously analyze feedback to identify pain points and tailor operational improvements accordingly.

7. Strategic Partnerships and Promotions

- **Offer Credit Card Promotions:** Since certain credit cards receive higher discounts, consider forming partnerships with financial institutions to offer targeted promotional discounts, boosting sales.
- **Reward Loyalty:** Develop loyalty programs for repeat customers, emphasizing positive feedback and encouraging additional purchases.

Strategic Initiatives for Improved Performance and Operations

1. Implement Robust Supply Chain and Logistics Management

- **Partner with Reliable Logistics Providers:** Establish agreements with established courier and shipping companies to reduce delays.
- **Optimize Inventory Planning:** Use historical sales data to forecast demand accurately and maintain optimal stock levels, minimizing stock-outs and overstocking.
- **Track Shipping KPIs:** Regularly monitor key metrics like delivery time, order accuracy, and shipment damages to identify bottlenecks and address them proactively.

2. Automate and Digitize Operations

- **Invest in ERP and CRM Systems:** Deploy integrated systems to streamline order processing, inventory management, and customer communication.
- **Real-Time Tracking:** Provide customers with real-time updates on their order status to increase transparency and reduce follow-up inquiries.
- **Automated Alerts:** Set up automated alerts for delays, inventory shortages, or abnormal order patterns to enable quick responses.

3. Enhance Customer Service and Feedback Loop

- **Proactive Customer Engagement:** Reach out to customers post-delivery to gather feedback and resolve issues promptly.
- **Customer Support Training:** Train support teams to handle complaints efficiently, turning dissatisfied customers into loyal advocates.

- **Feedback Integration:** Use customer feedback to continuously improve products, services, and operational processes.

4. Data-Driven Decision Making

- **Regular Dashboard Reviews:** Develop dashboards tracking sales, customer satisfaction, logistics, and feedback metrics for quick decision-making.
- **Predictive Analytics:** Use machine learning models to forecast demand, identify potential delays, and optimize routes and inventory.

5. Refine Marketing and Sales Strategies

- **Targeted Regional Campaigns:** Focus marketing efforts on underrepresented states to diversify the customer base.
- **Brand Promotion:** Leverage data on preferred vehicle brands to tailor promotional offers and bundles.
- **Incentivize Repeat Business:** Launch loyalty programs and referral bonuses for existing customers to boost repeat sales.

6. Operational Review and Continuous Improvement

- **Periodic Process Audits:** Conduct quarterly reviews of logistics, customer service, inventory, and sales processes to identify inefficiencies.
- **Kaizen Approach:** Foster a culture of continuous improvement where all teams suggest operational enhancements.
- **Cross-Functional Collaboration:** Encourage collaboration among marketing, logistics, sales, and customer support teams to synchronize efforts.

7. Staff Training and Development

- **Skill Enhancement:** Regularly train staff on new tools, customer handling, and operational best practices.
- **Performance Incentives:** Implement incentive programs to motivate teams to meet operational KPIs and improve service quality.

Summary:

Focusing on reducing shipping delays, improving customer satisfaction, expanding into underpenetrated regions, and leveraging brand preferences can help stabilize and grow New-Wheels' market presence. Continuous data monitoring and targeted marketing will be key to adapting strategies effectively. By focusing on supply chain optimization, automation, customer engagement, data-driven insights, and continuous improvement, New-Wheels can achieve smoother operations, reduce costs, and enhance overall customer satisfaction. These strategic initiatives will create a resilient, agile organization capable of adapting to market changes and customer needs.