

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 c.customer_id) AS Total_customers_with_orders  
FROM customer_t c  
JOIN order_t o  
ON c.customer_id = o.customer_id;
```

```
SELECT c.state,
```

```
    COUNT(DISTINCT c.customer_id) AS Total_customers_with_orders  
FROM customer_t c  
JOIN order_t o  
ON c.customer_id = o.customer_id  
GROUP BY c.state  
ORDER BY Total_customers_with_orders DESC;
```

Output:

Result: Passed

 Query 1

Query:

```
SELECT
    COUNT(DISTINCT c.customer_id) AS Total_customers_with_orders
FROM customer_t c
JOIN order_t o
    ON c.customer_id = o.customer_id
```

Output:

Showing 1 rows

Total_customers_with...
994

 Query 2

Query:

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

Output:

Showing first 10 rows out of 49 rows

state	Total_customers_with_...
Texas	97
California	97
Florida	86
New York	69
District of Columbia	35
Ohio	33

Observations and Insights:

- Total customers with orders are 994.
- Texas state has most of the customers with orders.

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 customer_count  
  FROM order_t o  
 JOIN product_t p  
    ON p.product_id = o.product_id  
 GROUP BY p.vehicle_maker  
 ORDER BY customer_count DESC  
 LIMIT 5;
```

Result: Passed

Query 1

Query:

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

Output:

Showing 5 rows

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

Output:

Observations and Insights:

- Chevrolet is most favorite vehicle brand.
-
-

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

Solution Query:

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

Output:

Result: Passed

 Query 1

Query:

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

Output:

Showing first 10 rows out of 143 rows

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

Observations and Insights:

- Dodge is the top-selling brand overall, and in Alabama, the best-selling brand is also Dodge.

Question 4: Find the overall average rating given by the customers.

What is the average rating in each quarter?

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

Solution Query:

```
SELECT
```

```
    quarter_number,
```

```
    ROUND(AVG (customer_feedback_rating), 2) as quarter_avg_rating
```

```
FROM (
```

```
    SELECT
```

```
        customer_id,
```

```
        customer_feedback,
```

```
        quarter_number,
```

```
    CASE
```

```
        WHEN customer_feedback = 'Very Bad' THEN 1
```

```
        WHEN customer_feedback = 'Bad' THEN 2
```

```
        WHEN customer_feedback = 'Okay' THEN 3
```

```
        WHEN customer_feedback = 'Good' THEN 4
```

```
        WHEN customer_feedback = 'Very Good' THEN 5
```

```
    END AS customer_feedback_rating
```

```
    FROM order_t
```

```
    ) AS feedback_rated
```

```
    GROUP BY quarter_number;
```

Output:

Result: Passed

 Query 1

Query:

```

SELECT
    quarter_number,
    ROUND(AVG(customer_feedback_rating), 2) as quarter_avg_rating
FROM (
    SELECT
        customer_id,
        customer_feedback,
        quarter_number,
        CASE
            WHEN customer_feedback = 'Very Bad' THEN 1
            WHEN customer_feedback = 'Bad' THEN 2
            WHEN customer_feedback = 'Okay' THEN 3
            WHEN customer_feedback = 'Good' THEN 4
            WHEN customer_feedback = 'Very Good' THEN 5
        END AS customer_feedback_rating
    FROM order_t
    ) AS feedback_rated
GROUP BY quarter_number
  
```

Output:

Showing 4 rows

quarter_number	quarter_avg_rating
1	3.55
2	3.35
3	2.96
4	2.4

Observations and Insights:

- Quarter 1 has the highest average rating (3.55) — customers were most satisfied in this period.
- Quarter 4 has the lowest average rating (2.40) — customer satisfaction dropped by the end of the year
-
-
-

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

Solution Query:

```

SELECT
    quarter_number,
    ROUND(very_bad_count * 100.0 / total_feedback, 2) AS very_bad_pct,
    ROUND(bad_count * 100.0 / total_feedback, 2) AS bad_pct,
    ROUND(okay_count * 100.0 / total_feedback, 2) AS okay_pct,
    ROUND(good_count * 100.0 / total_feedback, 2) AS good_pct,
    ROUND(very_good_count * 100.0 / total_feedback, 2) AS very_good_pct
FROM (
    SELECT
        quarter_number,
        SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) AS very_bad_count,
        SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) AS bad_count,
        SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) AS okay_count,
        SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) AS good_count,
        SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) AS very_good_count,
        COUNT(*) AS total_feedback
    FROM order_t
    GROUP BY quarter_number
) AS feedback_summary
ORDER BY quarter_number;
  
```

Output:

Result: Passed

(✓) Query 1

Query:

```

SELECT
    quarter_number,
    ROUND(very_bad_count * 100.0 / total_feedback, 2) AS very_bad_pct,
    ROUND(bad_count * 100.0 / total_feedback, 2) AS bad_pct,
    ROUND(okay_count * 100.0 / total_feedback, 2) AS okay_pct,
    ROUND(good_count * 100.0 / total_feedback, 2) AS good_pct,
    ROUND(very_good_count * 100.0 / total_feedback, 2) AS very_good_pct
FROM (
    SELECT
        quarter_number,
        SUM(CASE WHEN customer_feedback = 'Very Bad' THEN 1 ELSE 0 END) AS very_bad_count,
        SUM(CASE WHEN customer_feedback = 'Bad' THEN 1 ELSE 0 END) AS bad_count,
        SUM(CASE WHEN customer_feedback = 'Okay' THEN 1 ELSE 0 END) AS okay_count,
        SUM(CASE WHEN customer_feedback = 'Good' THEN 1 ELSE 0 END) AS good_count,
        SUM(CASE WHEN customer_feedback = 'Very Good' THEN 1 ELSE 0 END) AS very_good_count,
        COUNT(*) AS total_feedback
    FROM order_t
    GROUP BY quarter_number
) AS feedback_summary
ORDER BY quarter_number
  
```

Output:

Showing 4 rows

quarter_number	very_bad_pct	bad_pct	okay_pct	good_pct	very_good_pct
1	10.97	11.29	19.03	28.71	30
2	14.89	14.12	20.23	22.14	28.63
3	17.9	22.71	21.83	20.96	16.59
4	30.65	29.15	20.1	10.05	10.05

Observations and Insights:

- Customers were most satisfied in Q1.
-
-

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

Solution Query:

```
SELECT  
    quarter_number,  
    COUNT(order_id) as quarter_count  
FROM order_t  
GROUP BY quarter_number  
ORDER BY quarter_number;
```

Output:

Result: Passed											
Query 1											
Query:											
<pre>SELECT quarter_number, COUNT(order_id) as quarter_count FROM order_t GROUP BY quarter_number ORDER BY quarter_number</pre>											
Output:											
<p>Showing 4 rows</p> <table border="1"><thead><tr><th>quarter_number</th><th>quarter_count</th></tr></thead><tbody><tr><td>1</td><td>310</td></tr><tr><td>2</td><td>262</td></tr><tr><td>3</td><td>229</td></tr><tr><td>4</td><td>199</td></tr></tbody></table>		quarter_number	quarter_count	1	310	2	262	3	229	4	199
quarter_number	quarter_count										
1	310										
2	262										
3	229										
4	199										

Observations and Insights:

- Quarter 1 had the most orders (310), and Quarter 4 had the fewest (199).
- The number of orders decreased
-

Question 7: Calculate the net revenue generated by the company.

What is the quarter-over-quarter % change in net revenue?

Solution Query:

SELECT

```
quarter_number,  
SUM(quantity * (vehicle_price - (vehicle_price * discount / 100))) AS net_revenue
```

FROM order_t

GROUP BY quarter_number

ORDER BY quarter_number;

SELECT

```
quarter_number,  
net_revenue,  
100 * (net_revenue - LAG(net_revenue) OVER (ORDER BY quarter_number))  
/ LAG(net_revenue) OVER (ORDER BY quarter_number) AS prec_qoq
```

FROM (

SELECT

```
quarter_number,  
SUM((vehicle_price - ((discount/100) * vehicle_price)) * quantity) AS net_revenue
```

FROM order_t

GROUP BY quarter_number

) AS a;

Output:

Result: Passed

(✓) Query 1

Query:

```
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
```

Output:

Showing 4 rows

quarter_number	net_revenue
1	39421580.15929598
2	32715830.33996199
3	29229896.19364898
4	23346779.63060599

(✓) Query 2

Query:

```
SELECT
    quarter_number,
    net_revenue,
    100 * (net_revenue - LAG(net_revenue) OVER (ORDER BY quarter_number))
    / LAG(net_revenue) OVER (ORDER BY quarter_number) AS prec_qoq
FROM (
    SELECT
        quarter_number,
        SUM((vehicle_price - ((discount/100) * vehicle_price)) * quantity) AS net_revenue
    FROM order_t
    GROUP BY quarter_number
) AS a
```

Output:

Showing 4 rows

quarter_number	net_revenue	prec_qoq
1	39421580.15929598	
2	32715830.33996199	-17.010352685603124
3	29229896.19364898	-10.655190805458437
4	23346779.63060599	-20.127052535757088

Observations and Insights:

- Quarter 1 generated the highest revenue
-
-

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

Solution Query:

```
SELECT  
    quarter_number,  
    SUM((vehicle_price * quantity) - (vehicle_price * quantity * (discount/100))) AS final_sales_after_discount,  
    count(order_id) as total_orders  
  
FROM order_t  
  
GROUP BY quarter_number  
  
ORDER BY quarter_number;
```

Output:

Result: Passed																	
Query 1																	
Query:																	
<pre>SELECT quarter_number, SUM((vehicle_price * quantity) - (vehicle_price * quantity * (discount/100))) AS final_sales_after_discount, count(order_id) as total_orders FROM order_t GROUP BY quarter_number ORDER BY quarter_number</pre>																	
Output:																	
<p>Showing 4 rows</p> <table border="1"><thead><tr><th>quarter_number</th><th>final_sales_after_discount</th><th>total_orders</th></tr></thead><tbody><tr><td>1</td><td>39421580.15929598</td><td>310</td></tr><tr><td>2</td><td>32715830.33996199</td><td>262</td></tr><tr><td>3</td><td>29229896.19364898</td><td>229</td></tr><tr><td>4</td><td>23346779.63060599</td><td>199</td></tr></tbody></table>			quarter_number	final_sales_after_discount	total_orders	1	39421580.15929598	310	2	32715830.33996199	262	3	29229896.19364898	229	4	23346779.63060599	199
quarter_number	final_sales_after_discount	total_orders															
1	39421580.15929598	310															
2	32715830.33996199	262															
3	29229896.19364898	229															
4	23346779.63060599	199															

Observations and Insights:

- Both sales and orders decreased steadily from Q1 to Q4.
- The decline in sales is mainly due to fewer orders each quarter.
-
-
-

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

Solution Query:

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

Output:

Result: Passed		
Query 1		
Query:		
<pre>SELECT c.customer_id, c.credit_card_type, AVG(o.discount) AS Average_discount FROM customer_t c JOIN order_t o ON c.customer_id = o.customer_id GROUP BY credit_card_type ORDER BY Average_discount DESC</pre>		
Output:		
Showing first 10 rows out of 16 rows		
customer_id	credit_card_type	Average_discount
68151-1471	laser	0.643846153846154
41190-915	mastercard	0.6294999999999998
67510-0026	maestro	0.6242187499999999
11673-067	visa-electron	0.623469387755102
64980-183	china-unionpay	0.6221739130434784
0172-3762	instapayment	0.620625

Observations and Insights:

- Laser card users receive the highest average discount

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)) ,0)AS days_between  
FROM order_t  
GROUP BY quarter_number  
ORDER BY quarter_number;
```

Output:

Result: Passed

Query 1

Query:

```
SELECT
quarter_number,
ROUND(AVG(JULIANDAY(ship_date) - JULIANDAY(order_date)) ,0)AS days_between
FROM order_t
GROUP BY quarter_number
ORDER BY quarter_number
```

Output:

Showing 4 rows

quarter_number	days_between
1	57
2	71
3	118
4	174

Observations and Insights:

- The average shipping time has increased every quarter.

Business Metrics Overview

Total Revenue	Total Orders	Total Customers	Average Rating
124714086	1000	994	3.14
Last Quarter Revenue	Last quarter Orders	Average Days to Ship	% Good Feedback
23346780	199	98	21.5

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

Business Recommendations

- Repeat purchases are low.
- Reduce Shipping Time. Average 98 days to ship is very high.
- Track order trends, shipping delays, and feedback scores every quarter.