Supply Chain

□ **Generate some insights to solve a supply chain issue in the FMCG domain**

Problem Statement

AtliQ Mart is a growing FMCG manufacturer headquartered in Gujarat, India. It is currently operational in three cities Surat, Ahmedabad and Vadodara. They want to expand to other metros/Tier 1 cities in the next 2 years.

AtliQ Mart is currently facing a problem where a few key customers did not extend their annual contracts due to service issues. It is speculated that some of the essential products were either not delivered on time or not delivered in full over a continued period, which could have resulted in bad customer service. Management wants to fix this issue before expanding to other cities and requested their supply chain analytics team to track the 'On time' and 'In Full' delivery service level for all the customers daily basis so that they can respond swiftly to these issues.

The Supply Chain team decided to use a standard approach to measure the service level in which they will measure 'On-time delivery (OT) %', 'In-full delivery (IF) %', and OnTime in full (OTIF) %' of the customer orders daily basis against the target service level set for each customer.

Task:

Peter Pandey is the data analyst in the supply chain team who joined AtliQ Mart recently. He has been briefed about the the task in the stakeholder business review meeting. Now imagine yourself as Peter Pandey and play the role of the new data analyst who is excited to build this dashboard and perform the following task:

1. Create the metrics according to the metrics list.
2. Create a dashboard according to the requirements provided by stakeholders in the business review meeting. You will be provided with the transcript of this business review meeting in comic form.
3. Create relevant insights not provided in the metric list/stakeholder meeting.

Following are the tables used in this project:
1. dim_customers.csv
2. dim_products.csv
3. dim_date
4. dim_targets_orders
5. fact_order_lines.csv
6. fact_orders_aggregate.csv

....

Column Description for dim_customers:

This table contains all the information about customers

1. customer_id: Unique ID is given to each customer
2. customer_name: Name of the customer
3. city: It is the city where the customer is present

....

Column Description for dim_products:
This table contains all the information about the products

1. product_name: It is the name of the product
2. product_id: Unique ID is given to each of the products

3. category: It is the class to which the product belongs

Column Description for dim_date:
This table contains the dates at daily, monthly level and week numbers of the year

1. date: date at the daily level
2. mmm_yy: date at the monthly level
3. week_no: week number of the year as per the date column

Column Description for dim_targets_orders:
This table contains all target data at the customer level

customer_id: Unique ID that is given to each of the customers
ontime_target %: Target assigned for Ontime % for a given customer
infull_target %: Target assigned for infull % for a given customer
otif_target %: Target assigned for otif % for a given customer

---

Column Description for fact_order_lines:
This table contains all information about orders and each item inside the orders.

1. order_id: Unique ID for each order the customer placed
2. order_placement_date: It is the date when the customer placed the order
3. customer_id: Unique ID that is given to each of the customers
4. product_id: Unique ID that is given to each of the products
5. order_qty: It is the number of products requested by the customer to be delivered
6. agreed_delivery_date: It is the date agreed between the customer and Atliq Mart to deliver the products
7. actual_delivery_date: It is the actual date Atliq Mart delivered the product to the customer
8. delivered_qty: It is the number of products that are actually delivered to the customer

---

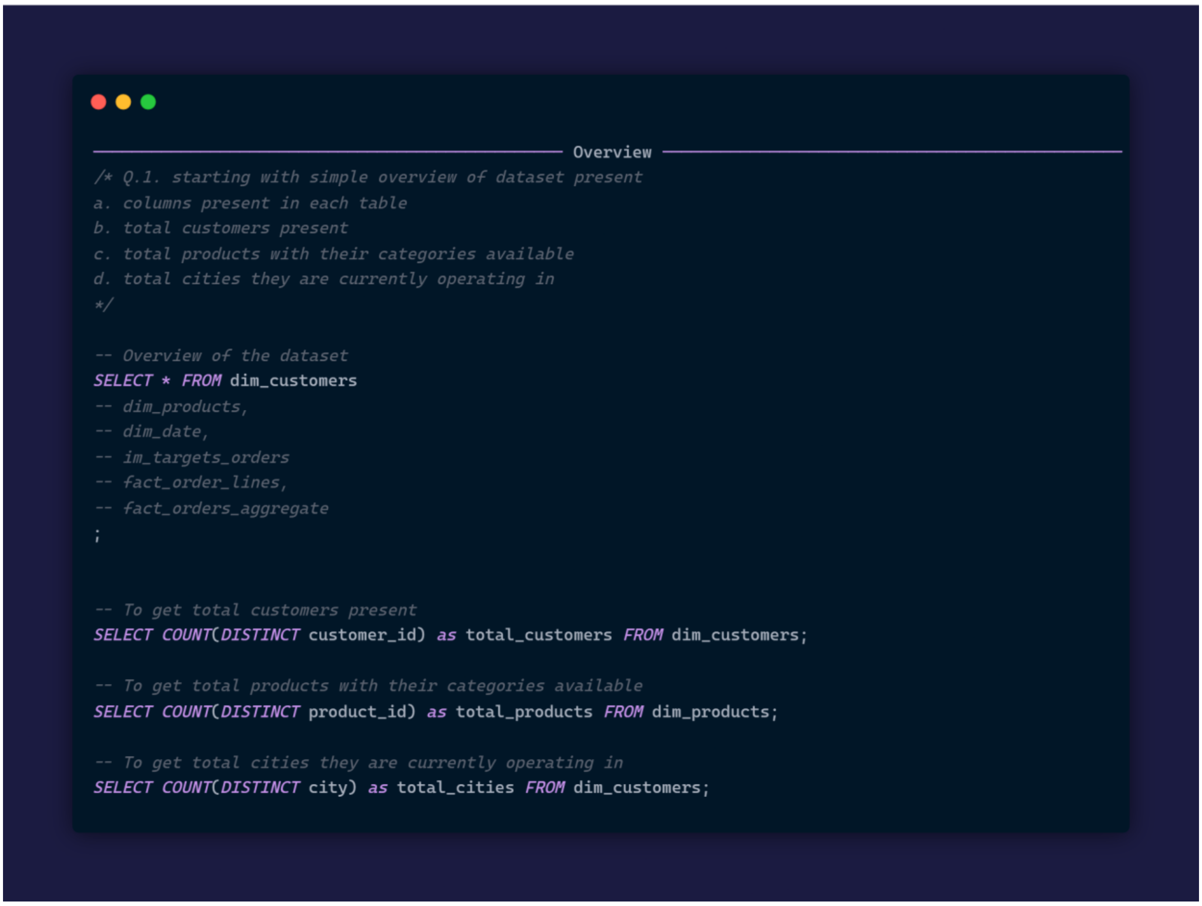
Column Description for fact_orders_aggregate:
This table contains information about OnTime, InFull and OnTime Infull information aggregated at the order level per customer

1. order_id: Unique ID for each order the customer placed
2. customer_id: Unique ID that is given to each of the customers
3. order_placement_date: It is the date when the customer placed the order
4. on_time: '1' denotes the order is delviered on time. '0' denotes the order is not delivered on time.
5. in_full: '1' denotes the order is delviered in full quantity. '0' denotes the order is not delivered in full quantity.
6: otif: '1' denotes the order is delviered both on time and in full quantity. '0' denotes the order is either not delivered on time or not in full quantity.

Overview of data

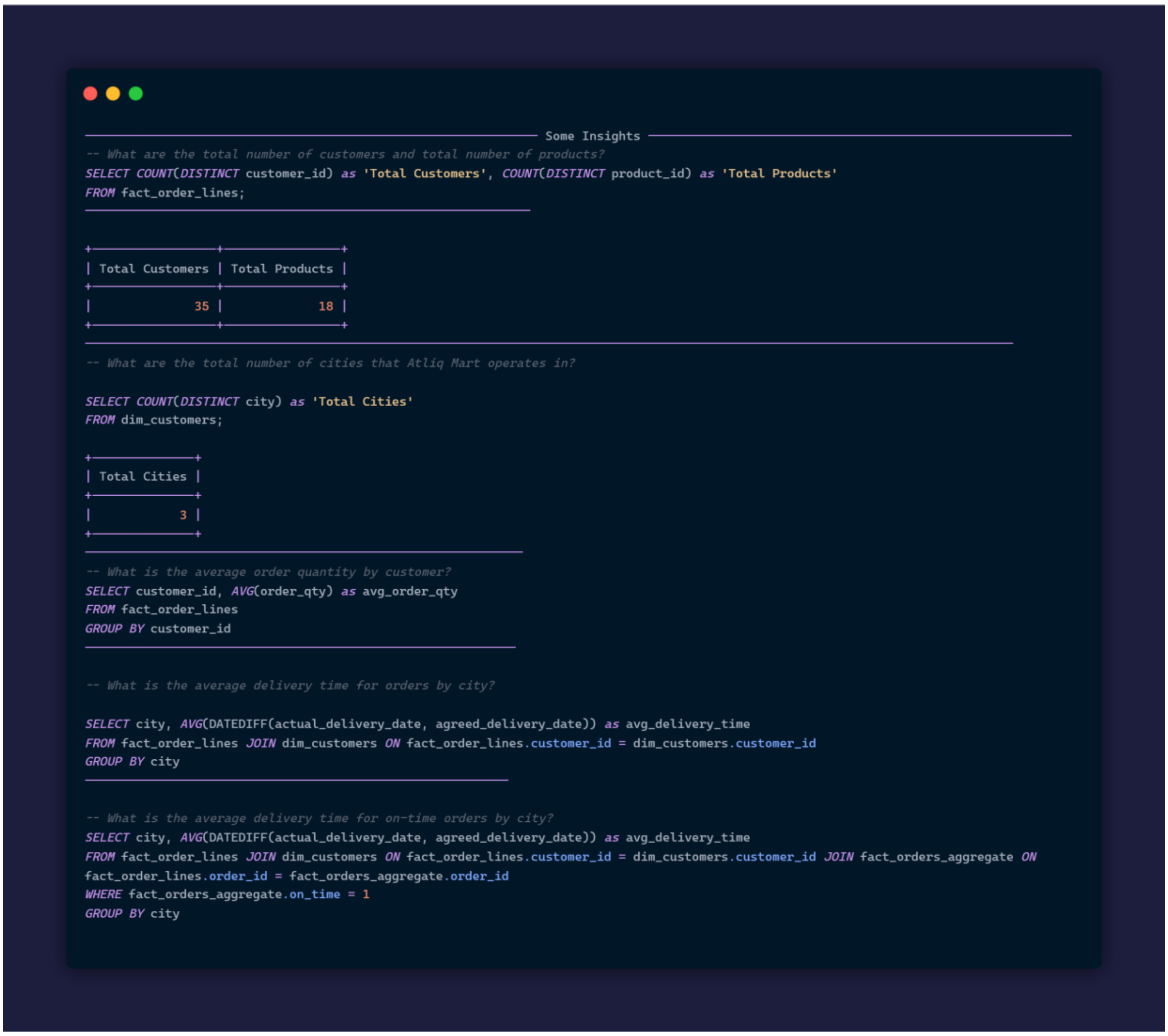
```
-- Overview of the dataset
select * from dim_customers;

-- To get total customers present
```

```
SELECT COUNT(DISTINCT customer_id) as total_customers FROM dim_customers;

-- To get total products with their categories available
SELECT COUNT(DISTINCT product_id) as total_products FROM dim_products;

-- To get total cities they are currently operating in
SELECT COUNT(DISTINCT city) as total_cities FROM dim_customers;
```



Overview

```
/* Q.1. starting with simple overview of dataset present
a. columns present in each table
b. total customers present
c. total products with their categories available
d. total cities they are currently operating in
*/

-- Overview of the dataset
SELECT * FROM dim_customers
-- dim_products,
-- dim_date,
-- im_targets_orders
-- fact_order_lines,
-- fact_orders_aggregate
;


-- To get total customers present
SELECT COUNT(DISTINCT customer_id) as total_customers FROM dim_customers;

-- To get total products with their categories available
SELECT COUNT(DISTINCT product_id) as total_products FROM dim_products;

-- To get total cities they are currently operating in
SELECT COUNT(DISTINCT city) as total_cities FROM dim_customers;
```



```
Some Insights
-- What are the total number of customers and total number of products?
SELECT COUNT(DISTINCT customer_id) as 'Total Customers', COUNT(DISTINCT product_id) as 'Total Products'
FROM fact_order_lines;


+-----------------+----------------+
| Total Customers | Total Products |
+-----------------+----------------+
|              35 |             18 |
+-----------------+----------------+

-- What are the total number of cities that Atliq Mart operates in?

SELECT COUNT(DISTINCT city) as 'Total Cities'
FROM dim_customers;

+--------------+
| Total Cities |
+--------------+
|            3 |
+--------------+

-- What is the average order quantity by customer?
SELECT customer_id, AVG(order_qty) as avg_order_qty
FROM fact_order_lines
GROUP BY customer_id


-- What is the average delivery time for orders by city?

SELECT city, AVG(DATEDIFF(actual_delivery_date, agreed_delivery_date)) as avg_delivery_time
FROM fact_order_lines JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
GROUP BY city


-- What is the average delivery time for on-time orders by city?
SELECT city, AVG(DATEDIFF(actual_delivery_date, agreed_delivery_date)) as avg_delivery_time
FROM fact_order_lines JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id JOIN fact_orders_aggregate ON
fact_order_lines.order_id = fact_orders_aggregate.order_id
WHERE fact_orders_aggregate.on_time = 1
GROUP BY city
```

Ordered Based Analysis

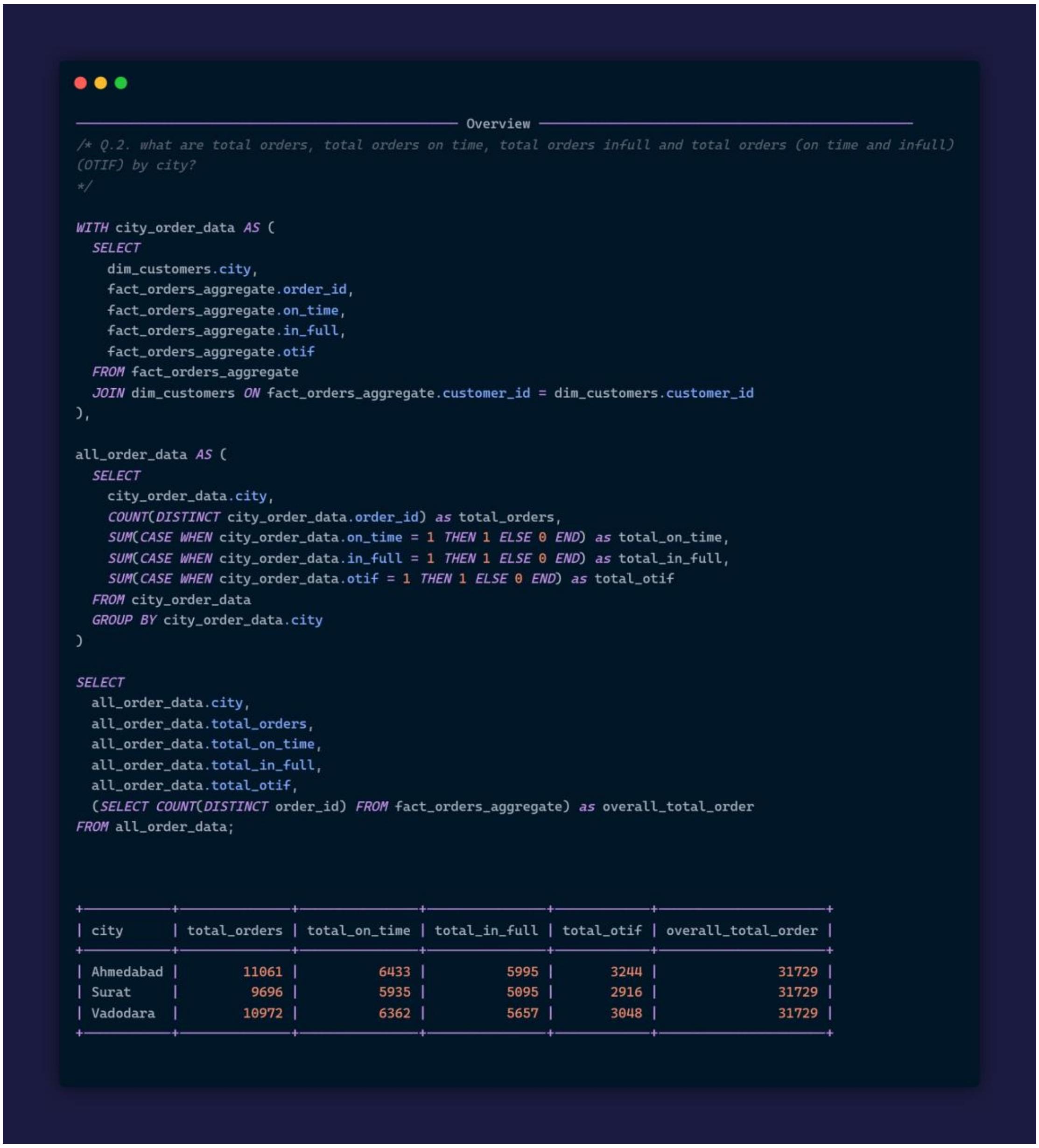
□ What are total orders, total orders on time, total orders infull and total orders (on time and infull)(OTIF) by city.

```
WITH city_order_data AS (
  SELECT
    dim_customers.city,
    fact_orders_aggregate.order_id,
    fact_orders_aggregate.on_time,
    fact_orders_aggregate.in_full,
    fact_orders_aggregate.otif
  FROM fact_orders_aggregate
  JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
),

all_order_data AS (
  SELECT
    city_order_data.city,
    COUNT(DISTINCT city_order_data.order_id) as total_orders,
    SUM(CASE WHEN city_order_data.on_time = 1 THEN 1 ELSE 0 END) as total_on_time,
    SUM(CASE WHEN city_order_data.in_full = 1 THEN 1 ELSE 0 END) as total_in_full,
    SUM(CASE WHEN city_order_data.otif = 1 THEN 1 ELSE 0 END) as total_otif
  FROM city_order_data
  GROUP BY city_order_data.city
)

SELECT
  all_order_data.city,
  all_order_data.total_orders,
  all_order_data.total_on_time,
  all_order_data.total_in_full,
  all_order_data.total_otif,
  (SELECT COUNT(DISTINCT order_id) FROM fact_orders_aggregate) as overall_total_order
FROM all_order_data;
```

Explanation:

- The first subquery, `city_order_data` , joins the fact_orders_aggregate table with the dim_customers table on the customer_id column to get the city information for each order. It then selects the city, order_id, on_time, in_full, and otif columns from the resulting joined table.
- The second subquery, `all_order_data`, groups the `city_order_data` by city and calculates the total number of orders, total number of on-time orders, total number of in-full orders, and total number of OTIF orders for each city.
- The main query then selects the city, total orders, total on-time orders, total in-full orders, and total OTIF orders from the all_order_data subquery, it also selects overall_total_order by counting all the orders from fact_orders_aggregate

This query will give you the total number of orders, total number of on-time orders, total number of in-full orders, and total number of OTIF orders for each city, along with the overall total count of all orders.



```
/* Q.2. what are total orders, total orders on time, total orders infull and total orders (on time and infull)
(OTIF) by city?
*/

WITH city_order_data AS (
  SELECT
    dim_customers.city,
    fact_orders_aggregate.order_id,
    fact_orders_aggregate.on_time,
    fact_orders_aggregate.in_full,
    fact_orders_aggregate.otif
  FROM fact_orders_aggregate
  JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
),

all_order_data AS (
  SELECT
    city_order_data.city,
    COUNT(DISTINCT city_order_data.order_id) as total_orders,
    SUM(CASE WHEN city_order_data.on_time = 1 THEN 1 ELSE 0 END) as total_on_time,
    SUM(CASE WHEN city_order_data.in_full = 1 THEN 1 ELSE 0 END) as total_in_full,
    SUM(CASE WHEN city_order_data.otif = 1 THEN 1 ELSE 0 END) as total_otif
  FROM city_order_data
  GROUP BY city_order_data.city
)

SELECT
  all_order_data.city,
  all_order_data.total_orders,
  all_order_data.total_on_time,
  all_order_data.total_in_full,
  all_order_data.total_otif,
  (SELECT COUNT(DISTINCT order_id) FROM fact_orders_aggregate) as overall_total_order
FROM all_order_data;
```

```
+-----------+--------------+---------------+---------------+------------+---------------------+
| city      | total_orders | total_on_time | total_in_full | total_otif | overall_total_order |
+-----------+--------------+---------------+---------------+------------+---------------------+
| Ahmedabad |        11061 |          6433 |          5995 |       3244 |               31729 |
| Surat     |         9696 |          5935 |          5095 |       2916 |               31729 |
| Vadodara  |        10972 |          6362 |          5657 |       3048 |               31729 |
+-----------+--------------+---------------+---------------+------------+---------------------+
```

Analyzing Delivery Performance

□ Provide insight regarding the share distribution of previous question metrics by customers.

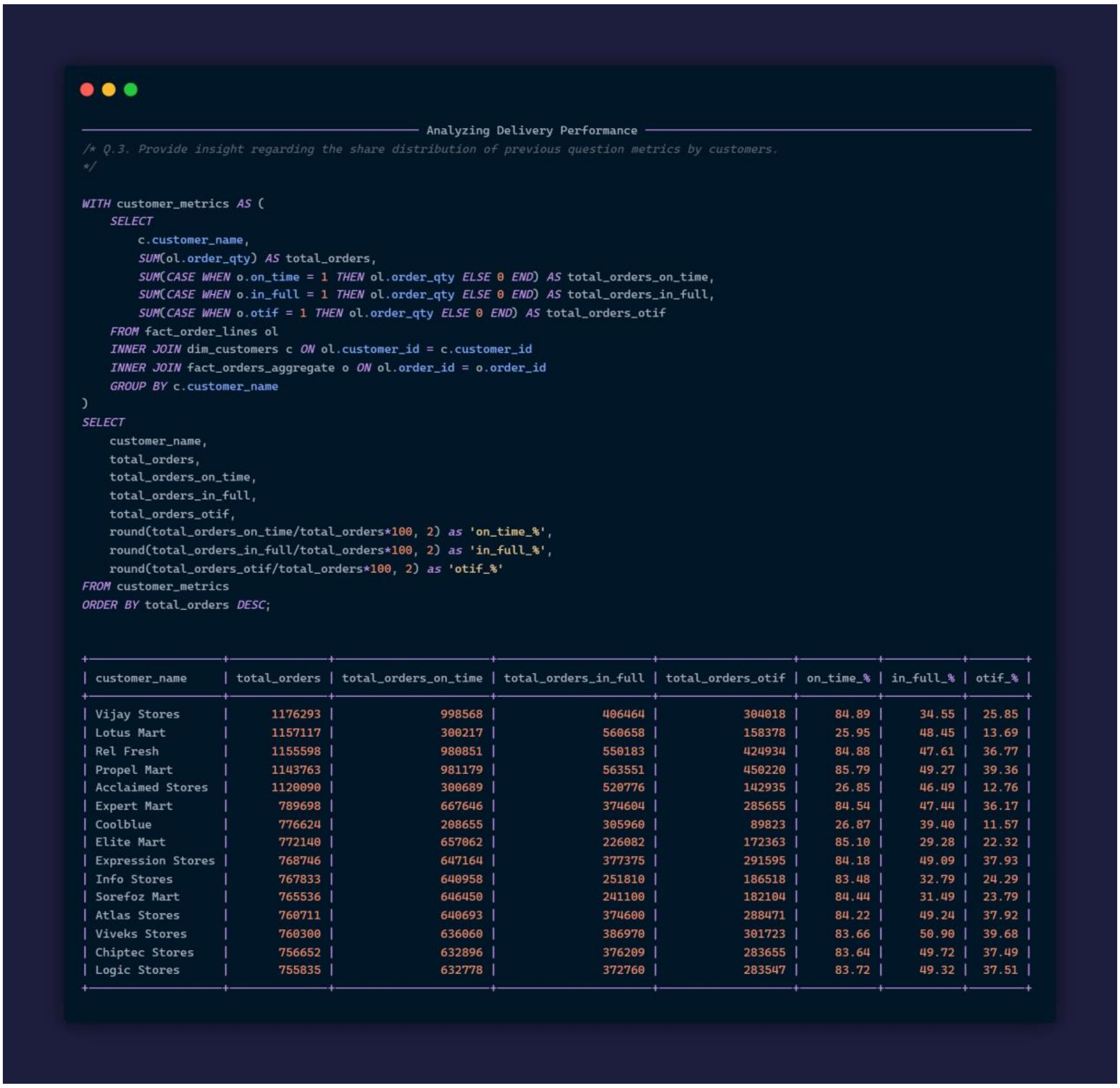
```
WITH customer_metrics AS (
    SELECT
        c.customer_name,
        SUM(ol.order_qty) AS total_orders,
        SUM(CASE WHEN o.on_time = 1 THEN ol.order_qty ELSE 0 END) AS total_orders_on_time,
        SUM(CASE WHEN o.in_full = 1 THEN ol.order_qty ELSE 0 END) AS total_orders_in_full,
        SUM(CASE WHEN o.otif = 1 THEN ol.order_qty ELSE 0 END) AS total_orders_otif
    FROM fact_order_lines ol
    INNER JOIN dim_customers c ON ol.customer_id = c.customer_id
    INNER JOIN fact_orders_aggregate o ON ol.order_id = o.order_id
    GROUP BY c.customer_name
)
```

```
SELECT
    customer_name,
    total_orders,
    total_orders_on_time,
    total_orders_in_full,
    total_orders_otif,
    round(total_orders_on_time/total_orders*100, 2) as 'on_time_%',
    round(total_orders_in_full/total_orders*100, 2) as 'in_full_%',
    round(total_orders_otif/total_orders*100, 2) as 'otif_%'
FROM      customer_metrics
ORDER BY total_orders DESC
```

Explanation:

- This query uses a common table expression (CTE) called "customer_metrics" to first calculate the total number of orders, total number of orders on time, total number of orders in full, and total number of orders on time and in full (OTIF) for each customer.
- The CTE joins the fact_order_lines table with the dim_customers table on the customer_id column, and the fact_orders_aggregate table on the order_id column.
- The CTE then groups the results by customer_name and calculates the sum of order_qty for each of the metrics.
- The main query then selects customer_name,total_orders,total_orders_on_time,total_orders_in_full,total_orders_otif,on_time_percentage,in_full_percen from the CTE, and orders the results by total_orders in descending order so that the customers with the highest number of orders appear first.
- In this query we are calculating three different percentage for on_time_percentage, in_full_percentage and otif_percentage by dividing the respective columns with total_orders column.

This query provides a way to see which customers have the highest share of total orders, total orders on time, total orders in full, and total orders on time and in full (OTIF), and also the percentage of these metrics for each customer. This information can be used to identify which customers are performing well in terms of delivery performance, and which customers may need more attention.



```
────────────────────────── Analyzing Delivery Performance ──────────────────────────
/* Q.3. Provide insight regarding the share distribution of previous question metrics by customers.
*/

WITH customer_metrics AS (
    SELECT
        c.customer_name,
        SUM(ol.order_qty) AS total_orders,
        SUM(CASE WHEN o.on_time = 1 THEN ol.order_qty ELSE 0 END) AS total_orders_on_time,
        SUM(CASE WHEN o.in_full = 1 THEN ol.order_qty ELSE 0 END) AS total_orders_in_full,
        SUM(CASE WHEN o.otif = 1 THEN ol.order_qty ELSE 0 END) AS total_orders_otif
    FROM fact_order_lines ol
    INNER JOIN dim_customers c ON ol.customer_id = c.customer_id
    INNER JOIN fact_orders_aggregate o ON ol.order_id = o.order_id
    GROUP BY c.customer_name
)
SELECT
    customer_name,
    total_orders,
    total_orders_on_time,
    total_orders_in_full,
    total_orders_otif,
    round(total_orders_on_time/total_orders*100, 2) as 'on_time_%',
    round(total_orders_in_full/total_orders*100, 2) as 'in_full_%',
    round(total_orders_otif/total_orders*100, 2) as 'otif_%'
FROM customer_metrics
ORDER BY total_orders DESC;
```

```
+-------------------+--------------+----------------------+----------------------+-------------------+-----------+-----------+--------+
| customer_name     | total_orders | total_orders_on_time | total_orders_in_full | total_orders_otif | on_time_% | in_full_% | otif_% |
+-------------------+--------------+----------------------+----------------------+-------------------+-----------+-----------+--------+
| Vijay Stores      |      1176293 |               998568 |               406464 |            304018 |     84.89 |     34.55 |  25.85 |
| Lotus Mart        |      1157117 |               300217 |               560658 |            158378 |     25.95 |     48.45 |  13.69 |
| Rel Fresh         |      1155598 |               980851 |               550183 |            424934 |     84.88 |     47.61 |  36.77 |
| Propel Mart       |      1143763 |               981179 |               563551 |            450220 |     85.79 |     49.27 |  39.36 |
| Acclaimed Stores  |      1120090 |               300689 |               520776 |            142935 |     26.85 |     46.49 |  12.76 |
| Expert Mart       |       789698 |               667646 |               374604 |            285655 |     84.54 |     47.44 |  36.17 |
| Coolblue          |       776624 |               208655 |               305960 |             89823 |     26.87 |     39.40 |  11.57 |
| Elite Mart        |       772140 |               657062 |               226082 |            172363 |     85.10 |     29.28 |  22.32 |
| Expression Stores |       768746 |               647164 |               377375 |            291595 |     84.18 |     49.09 |  37.93 |
| Info Stores       |       767833 |               640958 |               251810 |            186518 |     83.48 |     32.79 |  24.29 |
| Sorefoz Mart      |       765536 |               646450 |               241100 |            182104 |     84.44 |     31.49 |  23.79 |
| Atlas Stores      |       760711 |               640693 |               374600 |            288471 |     84.22 |     49.24 |  37.92 |
| Viveks Stores     |       760300 |               636060 |               386970 |            301723 |     83.66 |     50.90 |  39.68 |
| Chiptec Stores    |       756652 |               632896 |               376209 |            283655 |     83.64 |     49.72 |  37.49 |
| Logic Stores      |       755835 |               632778 |               372760 |            283547 |     83.72 |     49.32 |  37.51 |
+-------------------+--------------+----------------------+----------------------+-------------------+-----------+-----------+--------+
```

From the above results, we can observe the following insights:

1. Vijay Stores has the highest total number of orders, with a total of 1,176,293 orders.
2. Lotus Mart has the lowest percentage of on-time orders, at 25.95%.
3. Rel Fresh has the highest percentage of in-full orders, at 47.61%.
4. Propel Mart has the highest percentage of orders delivered on-time and in-full (OTIF), at 39.36%.
5. Acclaimed Stores has the lowest percentage of in-full orders, at 46.49%.
6. Expert Mart has the highest percentage of on-time orders, at 84.54%.
7. Coolblue has the lowest percentage of in-full orders, at 39.40%.
8. Elite Mart has the lowest percentage of orders delivered on-time and in-full (OTIF), at 22.32%.
9. Expression Stores has the highest percentage of in-full orders, at 49.09%.
10. Info Stores has the lowest percentage of on-time orders, at 32.79%.
11. Sorefoz Mart has the lowest percentage of orders delivered on-time and in-full (OTIF), at 23.79%.
12. Atlas Stores has the highest percentage of in-full orders, at 49.24%.
13. Viveks Stores has the highest percentage of orders delivered on-time and in-full (OTIF), at 39.68%.
14. Chiptec Stores has the highest percentage of orders delivered on-time and in-full (OTIF), at 37.49%.

15. Logic Stores has the highest percentage of orders delivered on-time and in-full (OTIF), at 37.51%.

Overall, we can see that there is significant variation in the performance metrics across customers. Some customers have high percentages of on-time and in-full orders, while others have low percentages. This suggests that there may be opportunities to improve delivery performance for certain customers. Additionally, the variation in performance across customers may indicate that different customers have different needs and expectations when it comes to delivery.

□ **Calculate % variance between actual and target from on time (OT), infull(IF) and 'ontime and infill'(OTIF) metrics by city.**

The variance is calculated as the difference between the actual and target performance, divided by the target performance, and multiplied by 100 to express it as a percentage.

(actual - target) / target * 100

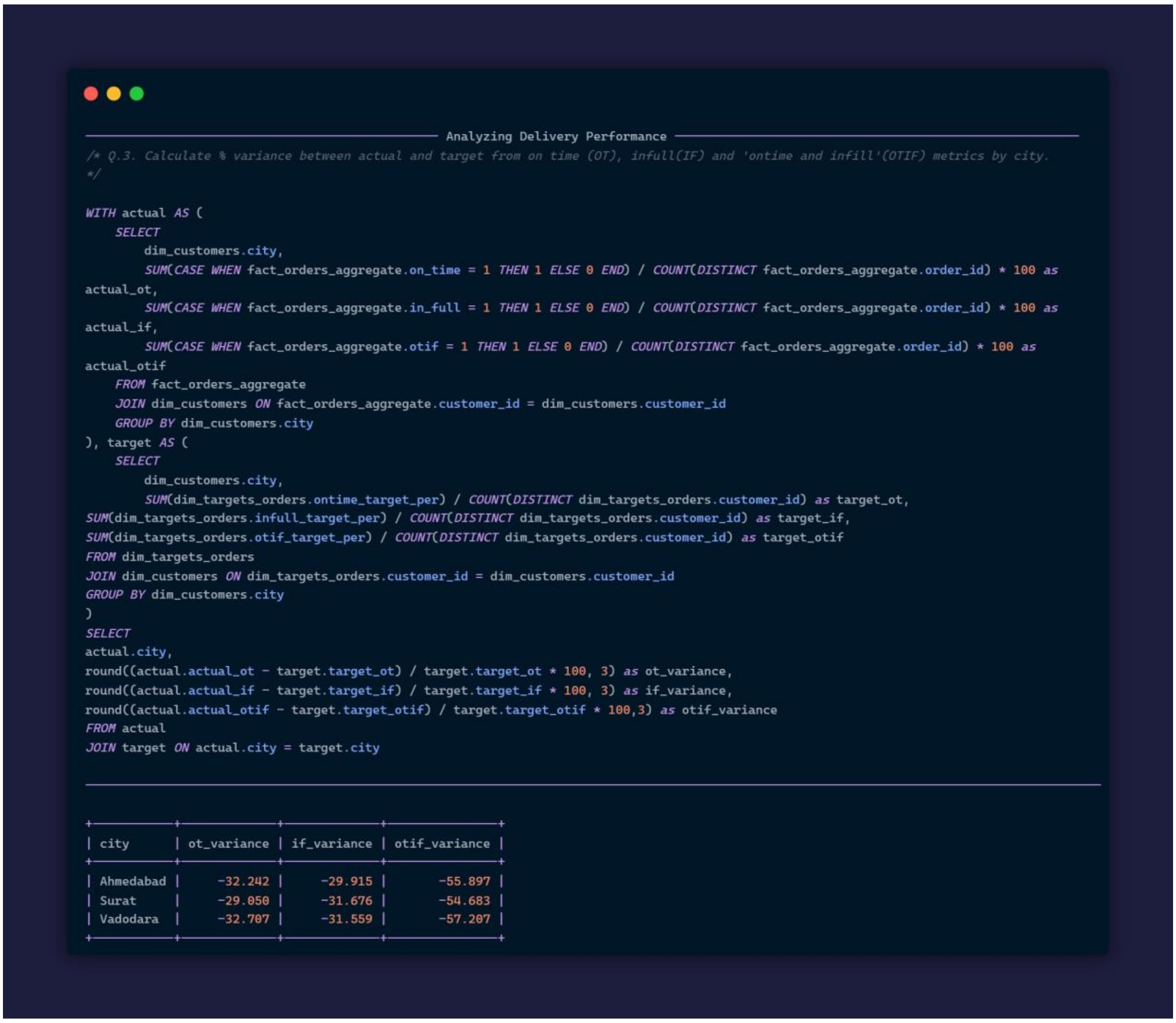
```
WITH actual AS (
    SELECT
        dim_customers.city,
        SUM(CASE WHEN fact_orders_aggregate.on_time = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as a
        SUM(CASE WHEN fact_orders_aggregate.in_full = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as a
        SUM(CASE WHEN fact_orders_aggregate.otif = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as actu
    FROM fact_orders_aggregate
    JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
    GROUP BY dim_customers.city
), target AS (
    SELECT
        dim_customers.city,
        SUM(dim_targets_orders.ontime_target_per) / COUNT(DISTINCT dim_targets_orders.customer_id) as target_ot,
SUM(dim_targets_orders.infull_target_per) / COUNT(DISTINCT dim_targets_orders.customer_id) as target_if,
SUM(dim_targets_orders.otif_target_per) / COUNT(DISTINCT dim_targets_orders.customer_id) as target_otif
FROM dim_targets_orders
JOIN dim_customers ON dim_targets_orders.customer_id = dim_customers.customer_id
GROUP BY dim_customers.city
)
SELECT
actual.city,
round((actual.actual_ot - target.target_ot) / target.target_ot * 100, 3) as ot_variance,
round((actual.actual_if - target.target_if) / target.target_if * 100, 3) as if_variance,
round((actual.actual_otif - target.target_otif) / target.target_otif * 100,3) as otif_variance
FROM actual
JOIN target ON actual.city = target.city
```

This query uses a combination of subqueries, joins, and aggregate functions to calculate the variance between the actual and target performance metrics of on-time (OT), in-full (IF), and on-time and in-full (OTIF) delivery by city.

The first subquery, "actual," calculates the actual performance of OT, IF, and OTIF delivery as a percentage of all orders by city, using a combination of SUM() and COUNT() aggregate functions. The query JOINs the fact_orders_aggregate table with the dim_customers table on the customer_id column, and then GROUPs the results by city. The actual performance is calculated by summing the number of on-time and in-full deliveries, and dividing that by the total number of unique order IDs.

The second subquery, "target," calculates the target performance of OT, IF, and OTIF delivery as a percentage by city, using a similar approach. The query JOINs the dim_targets_orders table with the dim_customers table on the customer_id column, and then GROUPs the results by city. The target performance is calculated by summing the on-time, in-full, and on-time and in-full targets and dividing that by the total number of unique customer IDs.

The final SELECT statement JOINs the "actual" and "target" subqueries on the city column, and uses the ROUND() function to calculate the variance between the actual and target performance for each metric, expressed as a percentage. The query returns the city name, variance for on-time, in-full, and on-time and in-full delivery respectively.



Analyzing Delivery Performance

```
/* Q.3. Calculate % variance between actual and target from on time (OT), infull(IF) and 'ontime and infill'(OTIF) metrics by city.
*/

WITH actual AS (
    SELECT
        dim_customers.city,
        SUM(CASE WHEN fact_orders_aggregate.on_time = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as
actual_ot,
        SUM(CASE WHEN fact_orders_aggregate.in_full = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as
actual_if,
        SUM(CASE WHEN fact_orders_aggregate.otif = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as
actual_otif
    FROM fact_orders_aggregate
    JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
    GROUP BY dim_customers.city
), target AS (
    SELECT
        dim_customers.city,
        SUM(dim_targets_orders.ontime_target_per) / COUNT(DISTINCT dim_targets_orders.customer_id) as target_ot,
SUM(dim_targets_orders.infull_target_per) / COUNT(DISTINCT dim_targets_orders.customer_id) as target_if,
SUM(dim_targets_orders.otif_target_per) / COUNT(DISTINCT dim_targets_orders.customer_id) as target_otif
FROM dim_targets_orders
JOIN dim_customers ON dim_targets_orders.customer_id = dim_customers.customer_id
GROUP BY dim_customers.city
)
SELECT
actual.city,
round((actual.actual_ot - target.target_ot) / target.target_ot * 100, 3) as ot_variance,
round((actual.actual_if - target.target_if) / target.target_if * 100, 3) as if_variance,
round((actual.actual_otif - target.target_otif) / target.target_otif * 100,3) as otif_variance
FROM actual
JOIN target ON actual.city = target.city
```

```
+-----------+-------------+-------------+---------------+
| city      | ot_variance | if_variance | otif_variance |
+-----------+-------------+-------------+---------------+
| Ahmedabad |     -32.242 |     -29.915 |       -55.897 |
| Surat     |     -29.050 |     -31.676 |       -54.683 |
| Vadodara  |     -32.707 |     -31.559 |       -57.207 |
+-----------+-------------+-------------+---------------+
```

□ top/bottom 5 customers by total quantity ordered, in full quantity ordered and 'OnTime and InFull' quantity ordered.

To find the top 5 customers by total quantity ordered:

```
SELECT
    dim_customers.customer_name,
    SUM(fact_order_lines.order_qty) as total_qty_ordered
FROM fact_order_lines
JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
GROUP BY dim_customers.customer_name
ORDER BY total_qty_ordered DESC
LIMIT 5;
```

To find the top 5 customers by in full quantity ordered:

```
SELECT
    dim_customers.customer_name,
    SUM(fact_order_lines.delivery_qty) as in_full_qty_ordered
FROM fact_order_lines
JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
GROUP BY dim_customers.customer_name
```

```
ORDER BY in_full_qty_ordered DESC
LIMIT 5;
```

To find the top 5 customers by 'ontime and infull' quantity ordered:

```
WITH ontime_infull AS (
    SELECT
        fact_order_lines.customer_id,
        SUM(CASE WHEN fact_orders_aggregate.otif = 1 THEN fact_order_lines.delivery_qty
                      ELSE 0 END) as ontime_infull_qty
    FROM fact_order_lines
    JOIN fact_orders_aggregate ON fact_order_lines.order_id = fact_orders_aggregate.order_id
    GROUP BY fact_order_lines.customer_id
)
SELECT
    dim_customers.customer_name,
    ontime_infull.ontime_infull_qty
FROM ontime_infull
JOIN dim_customers ON ontime_infull.customer_id = dim_customers.customer_id
ORDER BY ontime_infull_qty DESC
LIMIT 5;
```

The first query is finding the top 5 customers by total quantity ordered. It starts by joining the fact_order_lines table with the dim_customers table on the customer_id field. It then groups the results by customer_name and sums the order_qty field to calculate the total quantity ordered for each customer. The results are then ordered by the total_qty_ordered field in descending order and limited to the top 5 customers.

The second query is similar to the first query but it is finding the top 5 customers by in full quantity ordered. Instead of summing the order_qty field, it sums the delivery_qty field to calculate the in full quantity ordered for each customer. The results are then ordered by the in_full_qty_ordered field in descending order and limited to the top 5 customers.

The third query is finding the top 5 customers by ontime and infull quantity ordered. It starts by creating a subquery named ontime_infull that joins the fact_order_lines table with the fact_orders_aggregate table on the order_id field. It then groups the results by customer_id and sums the delivery_qty for only those records where the otif field equals 1. The subquery is then joined with dim_customers table on the customer_id field to get the customer name. The results are then ordered by the ontime_infull_qty field in descending order and limited to the top 5 customers.

□ provide actual OT%, IF%, and OTIF% by customers

```
WITH actual AS (
SELECT
dim_customers.customer_name,
SUM(CASE WHEN fact_orders_aggregate.on_time = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as actual_ot
SUM(CASE WHEN fact_orders_aggregate.in_full = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as actual_if
SUM(CASE WHEN fact_orders_aggregate.otif = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as actual_otif
FROM fact_orders_aggregate
JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
GROUP BY dim_customers.customer_name
)
SELECT
actual.customer_name,
round(actual.actual_ot,2) as ot_per,
round(actual.actual_if,2) as if_per,
round(actual.actual_otif,2) as otif_per
FROM actual
ORDER BY actual.customer_name;
```

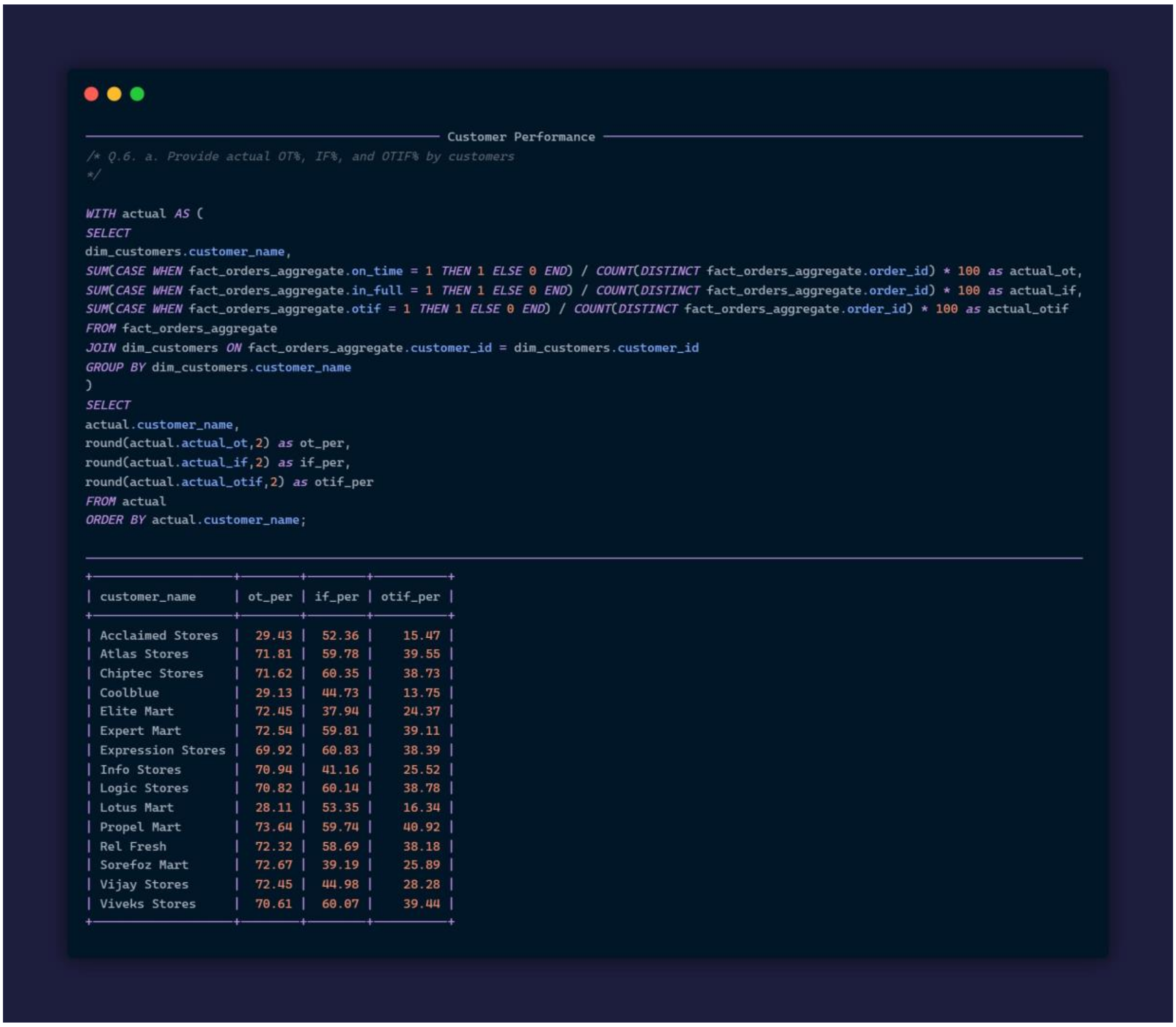
explanation:

This SQL query is used to provide the actual on-time (OT%), in-full (IF%), and on-time and in-full (OTIF%) percentages by customers. It does this by first selecting from the fact_orders_aggregate table and joining it with the dim_customers table on the customer_id column.

The query then uses a SUM and COUNT function in combination with a CASE statement to calculate the actual OT%, IF%, and OTIF% for each customer. The SUM function is used to count the number of orders that are on-time, in-full, or on-time and in-full (based on the value of the corresponding column in the fact_orders_aggregate table). The COUNT function is used to count the total number of orders for each customer.

The actual OT%, IF%, and OTIF% are calculated by dividing the sum of the corresponding orders by the total number of orders for each customer and then multiplying the result by 100 to express it as a percentage. The query then groups the results by the city column in the dim_customers table, so that the percentage values are calculated separately for each customer.

The final result will show the actual on time percentage, in full percentage and on time and in full percentage by customer.



Customer Performance

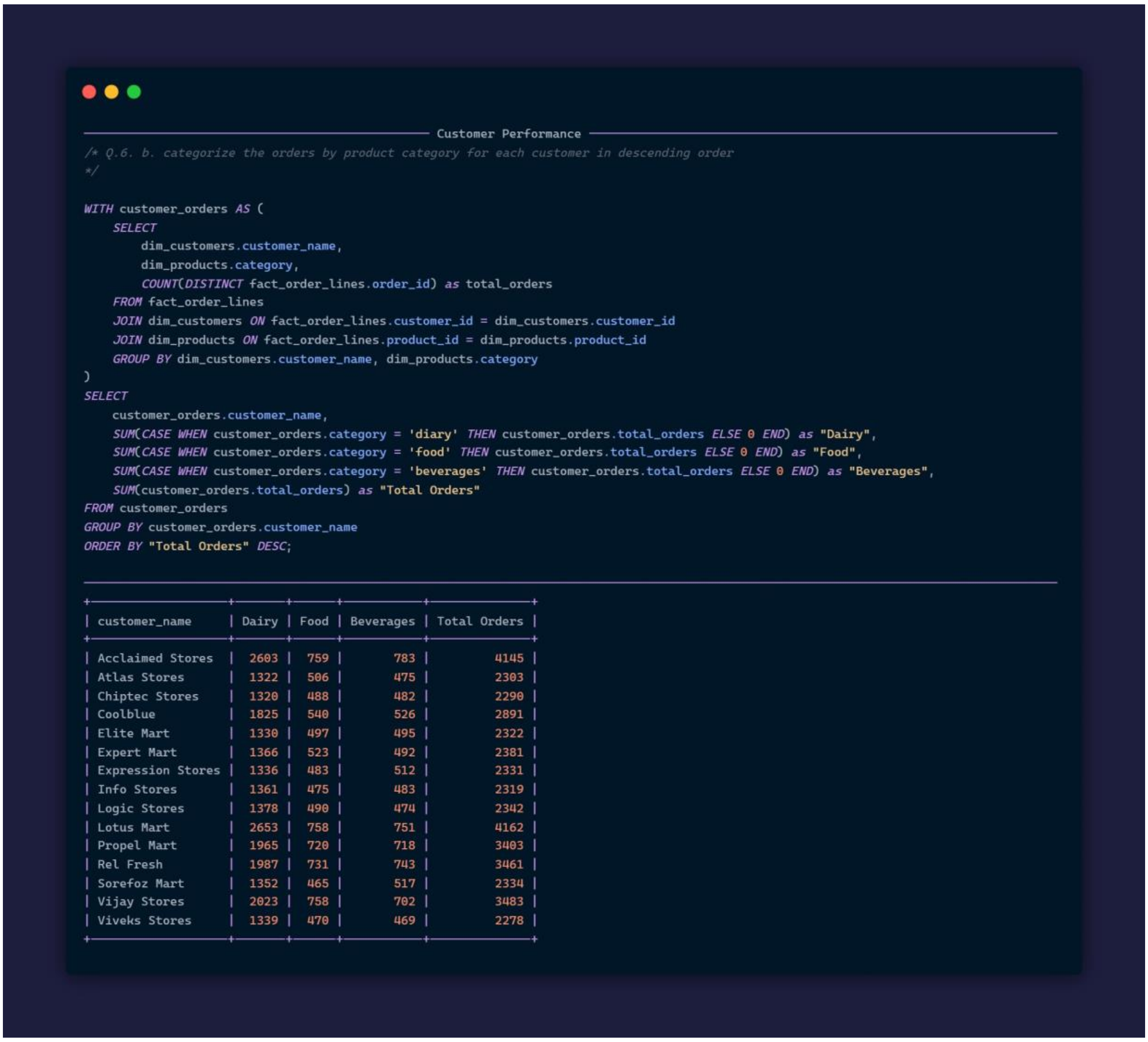
```
/* Q.6. a. Provide actual OT%, IF%, and OTIF% by customers
*/

WITH actual AS (
SELECT
dim_customers.customer_name,
SUM(CASE WHEN fact_orders_aggregate.on_time = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as actual_ot,
SUM(CASE WHEN fact_orders_aggregate.in_full = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as actual_if,
SUM(CASE WHEN fact_orders_aggregate.otif = 1 THEN 1 ELSE 0 END) / COUNT(DISTINCT fact_orders_aggregate.order_id) * 100 as actual_otif
FROM fact_orders_aggregate
JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
GROUP BY dim_customers.customer_name
)
SELECT
actual.customer_name,
round(actual.actual_ot,2) as ot_per,
round(actual.actual_if,2) as if_per,
round(actual.actual_otif,2) as otif_per
FROM actual
ORDER BY actual.customer_name;
```

```
+--------------------+--------+--------+----------+
| customer_name      | ot_per | if_per | otif_per |
+--------------------+--------+--------+----------+
| Acclaimed Stores   |  29.43 |  52.36 |    15.47 |
| Atlas Stores       |  71.81 |  59.78 |    39.55 |
| Chiptec Stores     |  71.62 |  60.35 |    38.73 |
| Coolblue           |  29.13 |  44.73 |    13.75 |
| Elite Mart         |  72.45 |  37.94 |    24.37 |
| Expert Mart        |  72.54 |  59.81 |    39.11 |
| Expression Stores  |  69.92 |  60.83 |    38.39 |
| Info Stores        |  70.94 |  41.16 |    25.52 |
| Logic Stores       |  70.82 |  60.14 |    38.78 |
| Lotus Mart         |  28.11 |  53.35 |    16.34 |
| Propel Mart        |  73.64 |  59.74 |    40.92 |
| Rel Fresh          |  72.32 |  58.69 |    38.18 |
| Sorefoz Mart       |  72.67 |  39.19 |    25.89 |
| Vijay Stores       |  72.45 |  44.98 |    28.28 |
| Viveks Stores      |  70.61 |  60.07 |    39.44 |
+--------------------+--------+--------+----------+
```

□ categorize the orders by product category for each customer in descending order

```
WITH customer_orders AS (
    SELECT
        dim_customers.customer_name,
        dim_products.category,
        COUNT(DISTINCT fact_order_lines.order_id) as total_orders
    FROM fact_order_lines
    JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
    JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
    GROUP BY dim_customers.customer_name, dim_products.category
)
```

```
SELECT
    customer_orders.customer_name,
    SUM(CASE WHEN customer_orders.category = 'diary' THEN customer_orders.total_orders ELSE 0 END) as "Dairy",
    SUM(CASE WHEN customer_orders.category = 'food' THEN customer_orders.total_orders ELSE 0 END) as "Food",
    SUM(CASE WHEN customer_orders.category = 'beverages' THEN customer_orders.total_orders ELSE 0 END) as "Beverages",
    SUM(customer_orders.total_orders) as "Total Orders"
FROM customer_orders
GROUP BY customer_orders.customer_name
ORDER BY "Total Orders" DESC;
```



Customer Performance

```
/* Q.6. b. categorize the orders by product category for each customer in descending order
*/

WITH customer_orders AS (
    SELECT
        dim_customers.customer_name,
        dim_products.category,
        COUNT(DISTINCT fact_order_lines.order_id) as total_orders
    FROM fact_order_lines
    JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
    JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
    GROUP BY dim_customers.customer_name, dim_products.category
)
SELECT
    customer_orders.customer_name,
    SUM(CASE WHEN customer_orders.category = 'diary' THEN customer_orders.total_orders ELSE 0 END) as "Dairy",
    SUM(CASE WHEN customer_orders.category = 'food' THEN customer_orders.total_orders ELSE 0 END) as "Food",
    SUM(CASE WHEN customer_orders.category = 'beverages' THEN customer_orders.total_orders ELSE 0 END) as "Beverages",
    SUM(customer_orders.total_orders) as "Total Orders"
FROM customer_orders
GROUP BY customer_orders.customer_name
ORDER BY "Total Orders" DESC;
```

```
+-------------------+-------+------+-----------+--------------+
| customer_name     | Dairy | Food | Beverages | Total Orders |
+-------------------+-------+------+-----------+--------------+
| Acclaimed Stores  |  2603 |  759 |       783 |         4145 |
| Atlas Stores      |  1322 |  506 |       475 |         2303 |
| Chiptec Stores    |  1320 |  488 |       482 |         2290 |
| Coolblue          |  1825 |  540 |       526 |         2891 |
| Elite Mart        |  1330 |  497 |       495 |         2322 |
| Expert Mart       |  1366 |  523 |       492 |         2381 |
| Expression Stores |  1336 |  483 |       512 |         2331 |
| Info Stores       |  1361 |  475 |       483 |         2319 |
| Logic Stores      |  1378 |  490 |       474 |         2342 |
| Lotus Mart        |  2653 |  758 |       751 |         4162 |
| Propel Mart       |  1965 |  720 |       718 |         3403 |
| Rel Fresh         |  1987 |  731 |       743 |         3461 |
| Sorefoz Mart      |  1352 |  465 |       517 |         2334 |
| Vijay Stores      |  2023 |  758 |       702 |         3483 |
| Viveks Stores     |  1339 |  470 |       469 |         2278 |
+-------------------+-------+------+-----------+--------------+
```

The above query is used to summarize the total number of orders for each customer across different product categories (dairy, food, and beverages) and also the total number of orders for each customer.

The query starts with a subquery named "customer_orders" which retrieves the customer name, product category, and the count of distinct order IDs for each combination of customer name and product category from the "fact_order_lines" table. This subquery is joined with the "dim_customers" and "dim_products" tables to retrieve the customer name and product category. The result is then grouped by customer name and category to get the total number of orders for each combination.

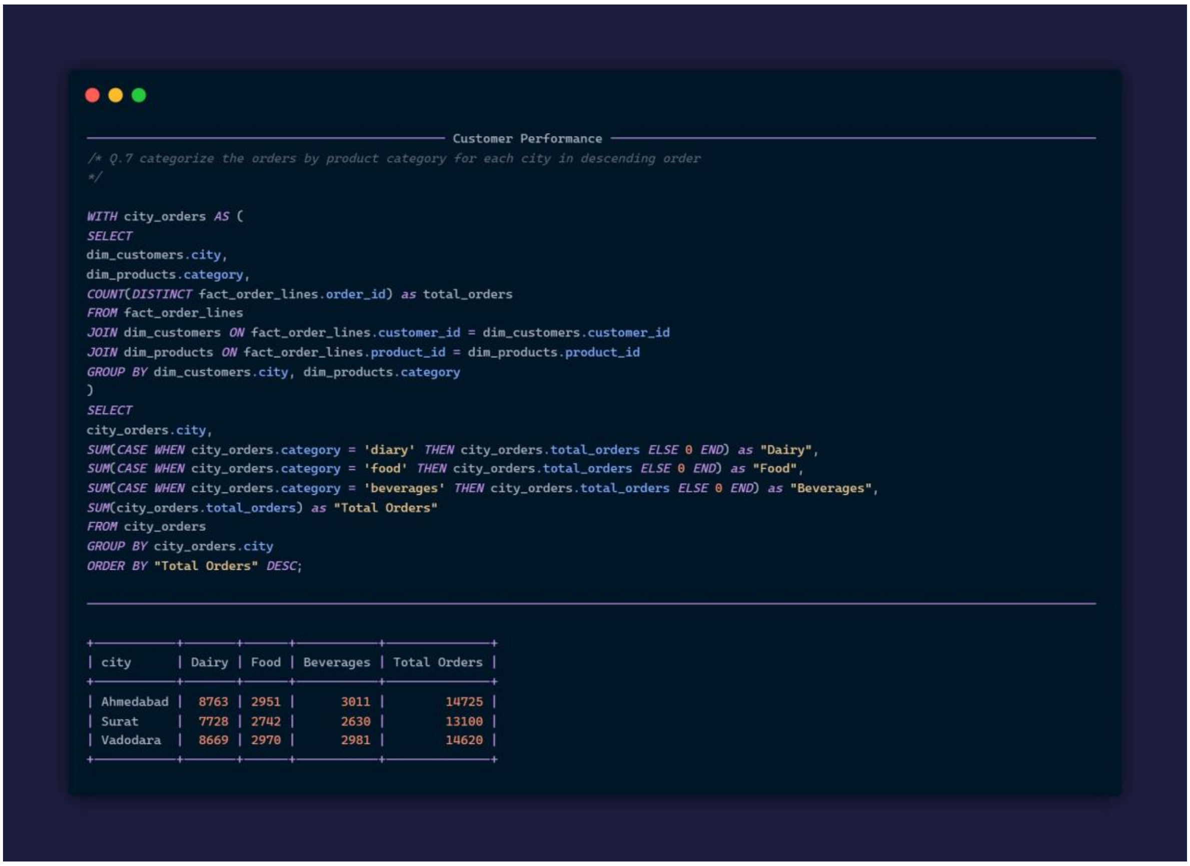
The main query then selects the customer name, and uses a SUM function with a CASE statement to calculate the total number of orders for each category (diary, food, and beverages) and also the total number of orders for each customer. Finally, the result is grouped by customer name and ordered by the total number of orders in descending order.

□ **categorize the orders by product category for each city in descending order**

```
WITH city_orders AS (
SELECT
dim_customers.city,
dim_products.category,
COUNT(DISTINCT fact_order_lines.order_id) as total_orders
FROM fact_order_lines
JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
GROUP BY dim_customers.city, dim_products.category
)
SELECT
city_orders.city,
SUM(CASE WHEN city_orders.category = 'diary' THEN city_orders.total_orders ELSE 0 END) as "Dairy",
SUM(CASE WHEN city_orders.category = 'food' THEN city_orders.total_orders ELSE 0 END) as "Food",
SUM(CASE WHEN city_orders.category = 'beverages' THEN city_orders.total_orders ELSE 0 END) as "Beverages",
SUM(city_orders.total_orders) as "Total Orders"
FROM city_orders
GROUP BY city_orders.city
ORDER BY "Total Orders" DESC;
```

This query uses a common table expression (CTE) named "city_orders" to first calculate the total number of orders for each city and product category combination by joining the fact_order_lines table with the dim_customers and dim_products tables, then grouping by city and category.

Then, it selects from the CTE, using a SUM() function with a CASE statement to calculate the total number of orders for each category (dairy, food, beverages) and a SUM() function to calculate the total number of orders for each city.
Finally, it groups the results by city and orders the output by the total number of orders in descending order. This query will give the total number of orders for each category and total orders for each city.

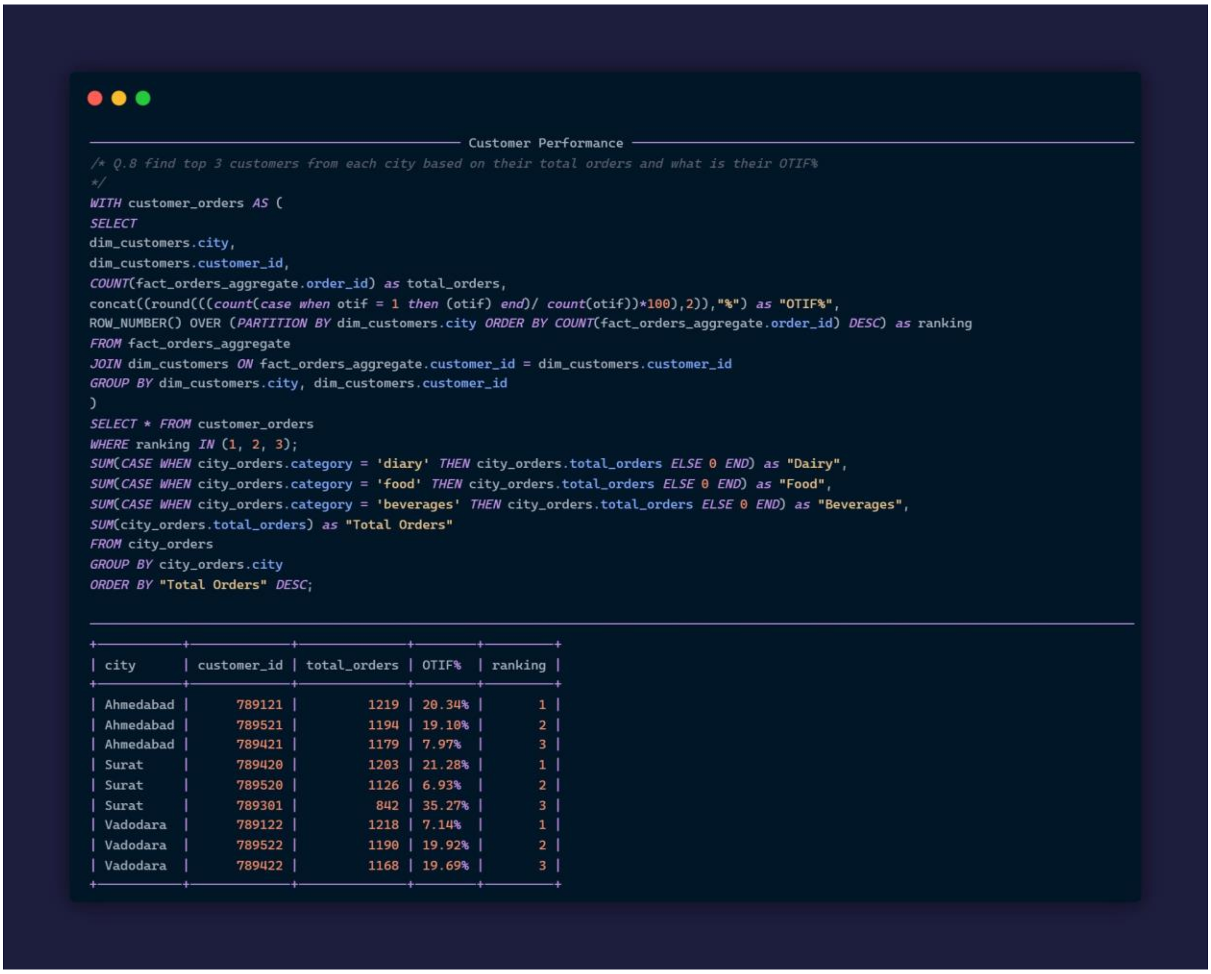


```
Customer Performance
/* Q.7 categorize the orders by product category for each city in descending order
*/

WITH city_orders AS (
SELECT
dim_customers.city,
dim_products.category,
COUNT(DISTINCT fact_order_lines.order_id) as total_orders
FROM fact_order_lines
JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
GROUP BY dim_customers.city, dim_products.category
)
SELECT
city_orders.city,
SUM(CASE WHEN city_orders.category = 'diary' THEN city_orders.total_orders ELSE 0 END) as "Dairy",
SUM(CASE WHEN city_orders.category = 'food' THEN city_orders.total_orders ELSE 0 END) as "Food",
SUM(CASE WHEN city_orders.category = 'beverages' THEN city_orders.total_orders ELSE 0 END) as "Beverages",
SUM(city_orders.total_orders) as "Total Orders"
FROM city_orders
GROUP BY city_orders.city
ORDER BY "Total Orders" DESC;
```

```
+-----------+-------+------+-----------+--------------+
| city      | Dairy | Food | Beverages | Total Orders |
+-----------+-------+------+-----------+--------------+
| Ahmedabad |  8763 | 2951 |      3011 |        14725 |
| Surat     |  7728 | 2742 |      2630 |        13100 |
| Vadodara  |  8669 | 2970 |      2981 |        14620 |
+-----------+-------+------+-----------+--------------+
```

□ **find top 3 customers from each city based on their total orders and what is their OTIF%**

```
WITH customer_orders AS (
SELECT
dim_customers.city,
dim_customers.customer_id,
COUNT(fact_orders_aggregate.order_id) as total_orders,
concat((round(((count(case when otif = 1 then (otif) end)/ count(otif))*100),2)),"%") as "OTIF%",
ROW_NUMBER() OVER (PARTITION BY dim_customers.city ORDER BY COUNT(fact_orders_aggregate.order_id) DESC) as ranking
FROM fact_orders_aggregate
JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
GROUP BY dim_customers.city, dim_customers.customer_id
)
SELECT * FROM customer_orders
WHERE ranking IN (1, 2, 3);
```



Customer Performance

```
/* Q.8 find top 3 customers from each city based on their total orders and what is their OTIF%
*/
WITH customer_orders AS (
SELECT
dim_customers.city,
dim_customers.customer_id,
COUNT(fact_orders_aggregate.order_id) as total_orders,
concat((round(((count(case when otif = 1 then (otif) end)/ count(otif))*100),2)),"%") as "OTIF%",
ROW_NUMBER() OVER (PARTITION BY dim_customers.city ORDER BY COUNT(fact_orders_aggregate.order_id) DESC) as ranking
FROM fact_orders_aggregate
JOIN dim_customers ON fact_orders_aggregate.customer_id = dim_customers.customer_id
GROUP BY dim_customers.city, dim_customers.customer_id
)
SELECT * FROM customer_orders
WHERE ranking IN (1, 2, 3);
SUM(CASE WHEN city_orders.category = 'diary' THEN city_orders.total_orders ELSE 0 END) as "Dairy",
SUM(CASE WHEN city_orders.category = 'food' THEN city_orders.total_orders ELSE 0 END) as "Food",
SUM(CASE WHEN city_orders.category = 'beverages' THEN city_orders.total_orders ELSE 0 END) as "Beverages",
SUM(city_orders.total_orders) as "Total Orders"
FROM city_orders
GROUP BY city_orders.city
ORDER BY "Total Orders" DESC;
```

```
+-----------+-------------+--------------+--------+---------+
| city      | customer_id | total_orders | OTIF%  | ranking |
+-----------+-------------+--------------+--------+---------+
| Ahmedabad |      789121 |         1219 | 20.34% |       1 |
| Ahmedabad |      789521 |         1194 | 19.10% |       2 |
| Ahmedabad |      789421 |         1179 | 7.97%  |       3 |
| Surat     |      789420 |         1203 | 21.28% |       1 |
| Surat     |      789520 |         1126 | 6.93%  |       2 |
| Surat     |      789301 |          842 | 35.27% |       3 |
| Vadodara  |      789122 |         1218 | 7.14%  |       1 |
| Vadodara  |      789522 |         1190 | 19.92% |       2 |
| Vadodara  |      789422 |         1168 | 19.69% |       3 |
+-----------+-------------+--------------+--------+---------+
```

This query is using a common table expression (CTE) called "customer_orders" to first select the city, customer_id, total orders, the OTIF percentage and a ranking based on the total orders for each customer, grouped by the customer's city.
The CTE is selecting data from the fact_orders_aggregate table, joining with the dim_customers table on the customer_id, and grouping the data by the city and customer_id.
The query calculates the total orders for each customer and then calculates the "OTIF%" by taking the count of all orders where otif=1 and dividing it by the total count of orders.
The ranking is assigned using the ROW_NUMBER() function, which assigns a unique number to each row within a result set, based on the order specified in the OVER clause. In this case, it assigns a unique ranking to each customer within each city, based on the total number of orders.

Then the final select statement selects all columns from the CTE where the ranking is in the top 3. So the final result will show top 3 customers from each city in terms of total orders and their OTIF%

Product Performance

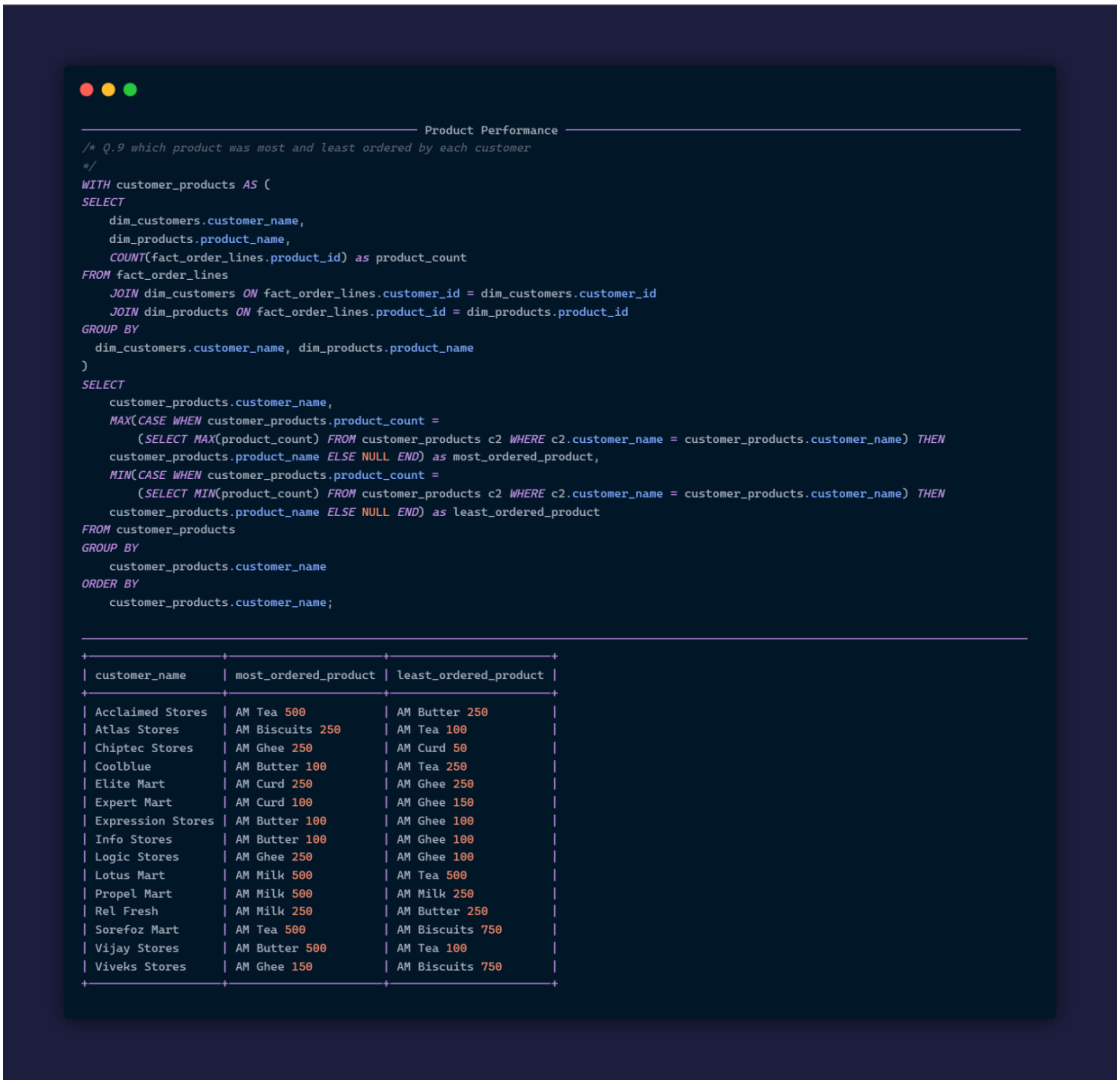
□ which product was most and least ordered by each customer

```
WITH customer_products AS (
SELECT
  dim_customers.customer_name,
  dim_products.product_name,
  COUNT(fact_order_lines.product_id) as product_count
FROM fact_order_lines
  JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
  JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
GROUP BY
  dim_customers.customer_name, dim_products.product_name
)
SELECT
  customer_products.customer_name,
  MAX(CASE WHEN customer_products.product_count =
    (SELECT MAX(product_count) FROM customer_products c2 WHERE c2.customer_name = customer_products.customer_name) THEN        customer
  MIN(CASE WHEN customer_products.product_count =
    (SELECT MIN(product_count) FROM customer_products c2 WHERE c2.customer_name = customer_products.customer_name) THEN        customer
FROM customer_products
GROUP BY
  customer_products.customer_name
ORDER BY
  customer_products.customer_name;
```

This query uses a common table expression (CTE) called "customer_products" to first count the number of orders of each product for each customer. Then, it selects the customer name, and uses the MAX() and MIN() aggregate functions with a subquery to find the most and least ordered product for each customer.

The subquery is used to find the maximum and minimum product count for each customer, and then these values are compared to the product count for each product for that customer in the outer query. The case statement is used to return the name of the product when the count matches the maximum or minimum count, and returns null for other products.

Finally, the query groups the results by customer name, and orders them alphabetically.



Product Performance

```sql
/* Q.9 which product was most and least ordered by each customer
*/
WITH customer_products AS (
SELECT
    dim_customers.customer_name,
    dim_products.product_name,
    COUNT(fact_order_lines.product_id) as product_count
FROM fact_order_lines
    JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
    JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
GROUP BY
  dim_customers.customer_name, dim_products.product_name
)
SELECT
    customer_products.customer_name,
    MAX(CASE WHEN customer_products.product_count =
        (SELECT MAX(product_count) FROM customer_products c2 WHERE c2.customer_name = customer_products.customer_name) THEN
    customer_products.product_name ELSE NULL END) as most_ordered_product,
    MIN(CASE WHEN customer_products.product_count =
        (SELECT MIN(product_count) FROM customer_products c2 WHERE c2.customer_name = customer_products.customer_name) THEN
    customer_products.product_name ELSE NULL END) as least_ordered_product
FROM customer_products
GROUP BY
    customer_products.customer_name
ORDER BY
    customer_products.customer_name;
```

```
+-------------------+----------------------+-----------------------+
| customer_name     | most_ordered_product | least_ordered_product |
+-------------------+----------------------+-----------------------+
| Acclaimed Stores  | AM Tea 500           | AM Butter 250         |
| Atlas Stores      | AM Biscuits 250      | AM Tea 100            |
| Chiptec Stores    | AM Ghee 250          | AM Curd 50            |
| Coolblue          | AM Butter 100        | AM Tea 250            |
| Elite Mart        | AM Curd 250          | AM Ghee 250           |
| Expert Mart       | AM Curd 100          | AM Ghee 150           |
| Expression Stores | AM Butter 100        | AM Ghee 100           |
| Info Stores       | AM Butter 100        | AM Ghee 100           |
| Logic Stores      | AM Ghee 250          | AM Ghee 100           |
| Lotus Mart        | AM Milk 500          | AM Tea 500            |
| Propel Mart       | AM Milk 500          | AM Milk 250           |
| Rel Fresh         | AM Milk 250          | AM Butter 250         |
| Sorefoz Mart      | AM Tea 500           | AM Biscuits 750       |
| Vijay Stores      | AM Butter 500        | AM Tea 100            |
| Viveks Stores     | AM Ghee 150          | AM Biscuits 750       |
+-------------------+----------------------+-----------------------+
```

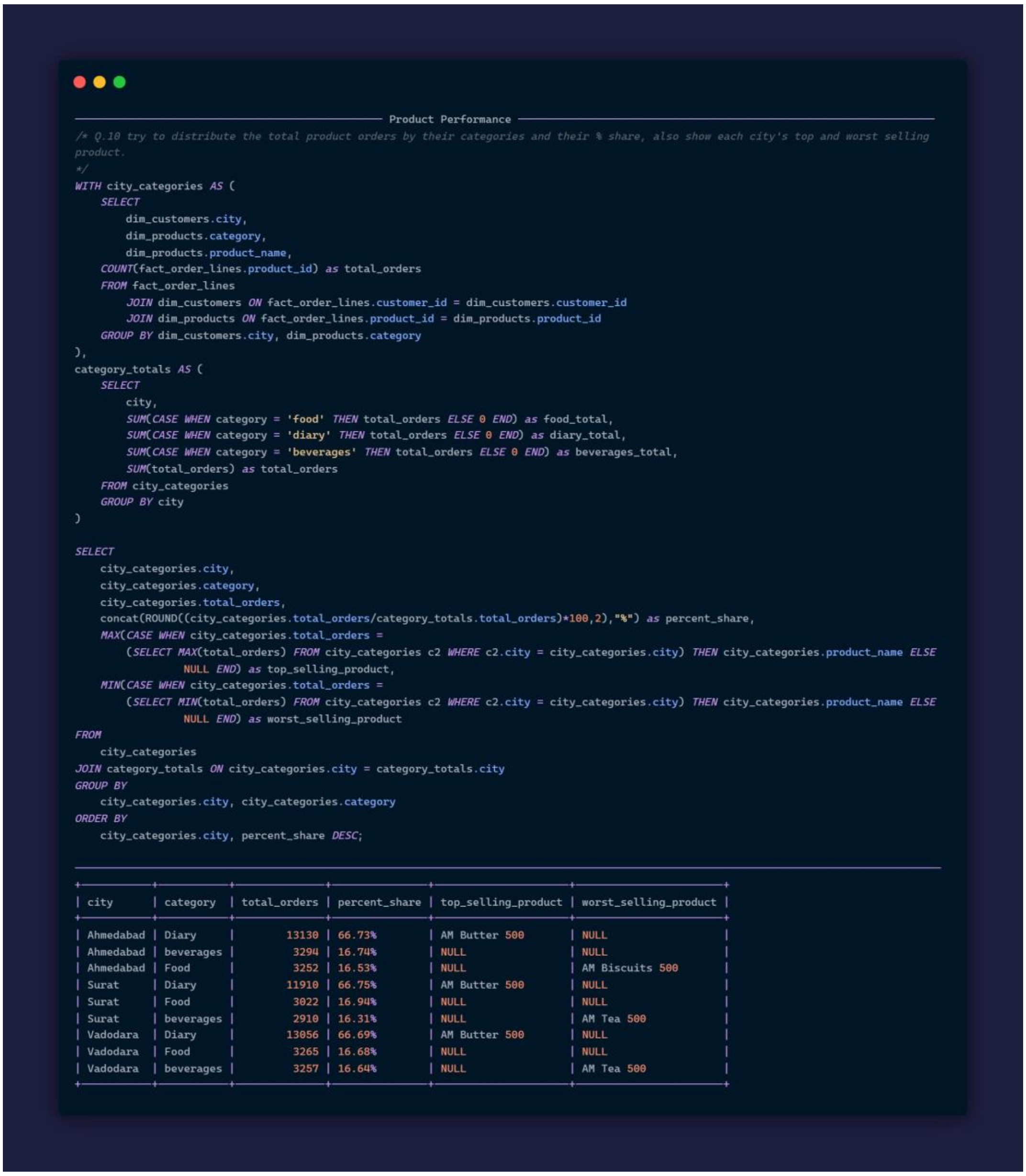
□ **try to distribute the total product orders by their categories and their % share, also show each city's top and worst selling product.**

```
WITH city_categories AS (
SELECT
dim_customers.city,
dim_products.category,
dim_products.product_name,
COUNT(fact_order_lines.product_id) as total_orders
FROM fact_order_lines
JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
GROUP BY dim_customers.city, dim_products.category
),
category_totals AS (
SELECT
city,
SUM(CASE WHEN category = 'food' THEN total_orders ELSE 0 END) as food_total,
SUM(CASE WHEN category = 'diary' THEN total_orders ELSE 0 END) as diary_total,
SUM(CASE WHEN category = 'beverages' THEN total_orders ELSE 0 END) as beverages_total,
SUM(total_orders) as total_orders
```

```
FROM city_categories
GROUP BY city
)

SELECT
city_categories.city,
city_categories.category,
city_categories.total_orders,
concat(ROUND((city_categories.total_orders/category_totals.total_orders)*100,2),"%") as percent_share,
MAX(CASE WHEN city_categories.total_orders =
(SELECT MAX(total_orders) FROM city_categories c2 WHERE c2.city = city_categories.city) THEN city_categories.product_name ELSE NULL EN
MIN(CASE WHEN city_categories.total_orders =
(SELECT MIN(total_orders) FROM city_categories c2 WHERE c2.city = city_categories.city) THEN city_categories.product_name ELSE NULL EN
FROM city_categories
JOIN category_totals ON city_categories.city = category_totals.city
GROUP BY city_categories.city, city_categories.category
ORDER BY city_categories.city, percent_share DESC;
```

This query is using a combination of subqueries, joins, and aggregate functions to analyze sales data from multiple tables.

1. The first subquery, "city_categories", is joining the fact_order_lines table with the dim_customers and dim_products tables on the customer_id and product_id fields respectively. It is then grouping the results by the city and category fields and counting the total number of orders for each group.
2. The second subquery, "category_totals", is taking the output of the first subquery and grouping it again by city. It is then using SUM() with a CASE statement to calculate the total number of orders for each category (food, diary, and beverages) and the total number of orders for each city.
3. The final SELECT statement is joining the "city_categories" subquery with the "category_totals" subquery on the city field. It is then using the MAX() and MIN() aggregate functions along with a subquery to find the top and worst selling products for each city and category. It is also using the ROUND() function to calculate the percent share of each category for each city and concatenating it with the "%" sign. The query is then ordering the results by city and percent_share in descending order.



Product Performance

```sql
/* Q.10 try to distribute the total product orders by their categories and their % share, also show each city's top and worst selling
product.
*/
WITH city_categories AS (
    SELECT
        dim_customers.city,
        dim_products.category,
        dim_products.product_name,
    COUNT(fact_order_lines.product_id) as total_orders
    FROM fact_order_lines
        JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
        JOIN dim_products ON fact_order_lines.product_id = dim_products.product_id
    GROUP BY dim_customers.city, dim_products.category
),
category_totals AS (
    SELECT
        city,
        SUM(CASE WHEN category = 'food' THEN total_orders ELSE 0 END) as food_total,
        SUM(CASE WHEN category = 'diary' THEN total_orders ELSE 0 END) as diary_total,
        SUM(CASE WHEN category = 'beverages' THEN total_orders ELSE 0 END) as beverages_total,
        SUM(total_orders) as total_orders
    FROM city_categories
    GROUP BY city
)

SELECT
    city_categories.city,
    city_categories.category,
    city_categories.total_orders,
    concat(ROUND((city_categories.total_orders/category_totals.total_orders)*100,2),"%") as percent_share,
    MAX(CASE WHEN city_categories.total_orders =
        (SELECT MAX(total_orders) FROM city_categories c2 WHERE c2.city = city_categories.city) THEN city_categories.product_name ELSE
                 NULL END) as top_selling_product,
    MIN(CASE WHEN city_categories.total_orders =
        (SELECT MIN(total_orders) FROM city_categories c2 WHERE c2.city = city_categories.city) THEN city_categories.product_name ELSE
                 NULL END) as worst_selling_product
FROM
    city_categories
JOIN category_totals ON city_categories.city = category_totals.city
GROUP BY
    city_categories.city, city_categories.category
ORDER BY
    city_categories.city, percent_share DESC;
```

```
+-----------+-----------+--------------+---------------+---------------------+-----------------------+
| city      | category  | total_orders | percent_share | top_selling_product | worst_selling_product |
+-----------+-----------+--------------+---------------+---------------------+-----------------------+
| Ahmedabad | Diary     |        13130 | 66.73%        | AM Butter 500       | NULL                  |
| Ahmedabad | beverages |         3294 | 16.74%        | NULL                | NULL                  |
| Ahmedabad | Food      |         3252 | 16.53%        | NULL                | AM Biscuits 500       |
| Surat     | Diary     |        11910 | 66.75%        | AM Butter 500       | NULL                  |
| Surat     | Food      |         3022 | 16.94%        | NULL                | NULL                  |
| Surat     | beverages |         2910 | 16.31%        | NULL                | AM Tea 500            |
| Vadodara  | Diary     |        13056 | 66.69%        | AM Butter 500       | NULL                  |
| Vadodara  | Food      |         3265 | 16.68%        | NULL                | NULL                  |
| Vadodara  | beverages |         3257 | 16.64%        | NULL                | AM Tea 500            |
+-----------+-----------+--------------+---------------+---------------------+-----------------------+
```

The result table shows the sales statistics for different categories of products in three different cities, Ahmedabad, Surat, and Vadodara. The categories are Diary, Beverages, and Food. The table shows the total number of orders for each category in each city, the percentage share of orders for each category, the top-selling product in each category, and the worst-selling product in each category.

In Ahmedabad, the highest number of orders were placed for Diary products, with 66.73% of total orders. The top-selling product in this category is AM Butter 500. Beverages and Food categories both had 16.74% and 16.53% of total orders respectively. No top-selling product is mentioned for these categories and the worst-selling product for Food category is AM Biscuits 500.

Similarly, In Surat, the highest number of orders were placed for Diary products, with 66.75% of total orders. The top-selling product in this category is AM Butter 500. Food category had 16.94% of total orders and Beverages category had 16.31% of total orders. The worst-selling product for Beverages category is AM Tea 500.

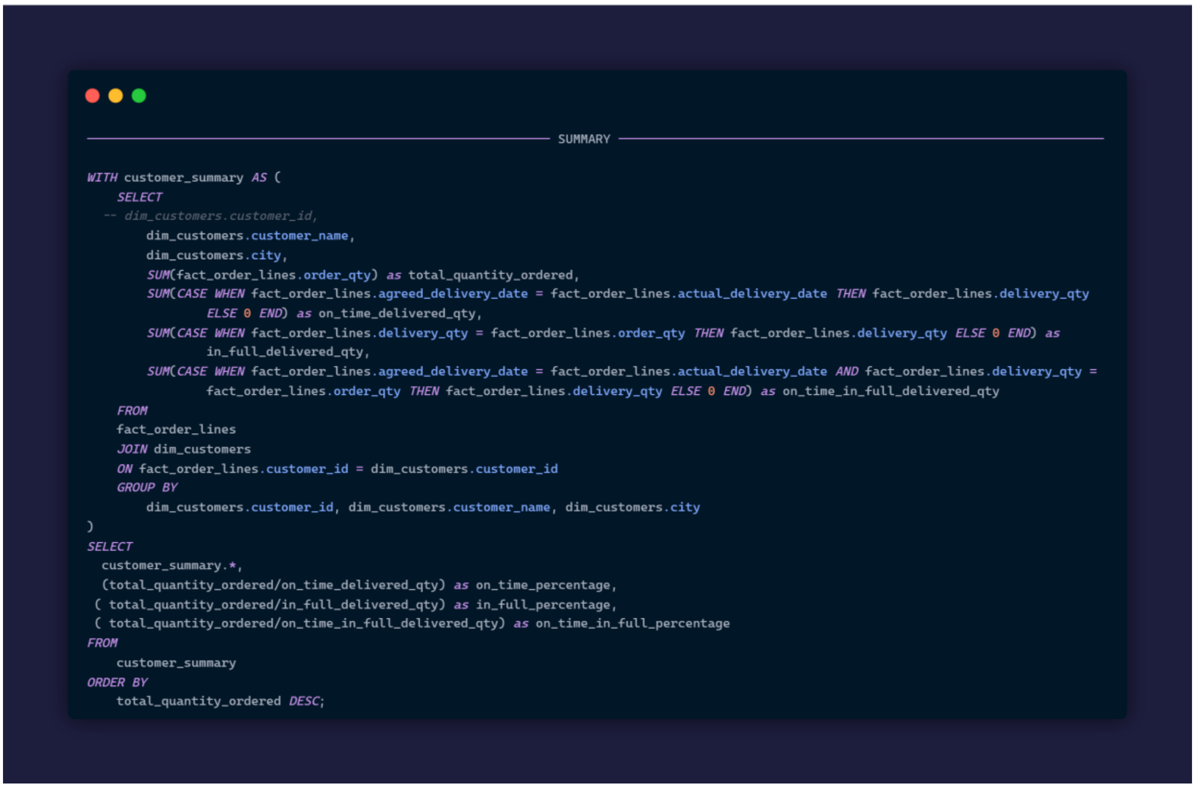
In Vadodara, the highest number of orders were placed for Diary products, with 66.69% of total orders. The top-selling product in this category is AM Butter 500. Food category had 16.68% of total orders and Beverages category had 16.64% of total orders. The worst-selling product for Beverages category is AM Tea 500.

It is evident that Diary products are the most popular across all three cities, with AM Butter 500 being the top-selling product in that category. Beverages and Food categories follow with similar percentages of total orders, and no clear top-selling products or worst-selling products in these categories are mentioned.

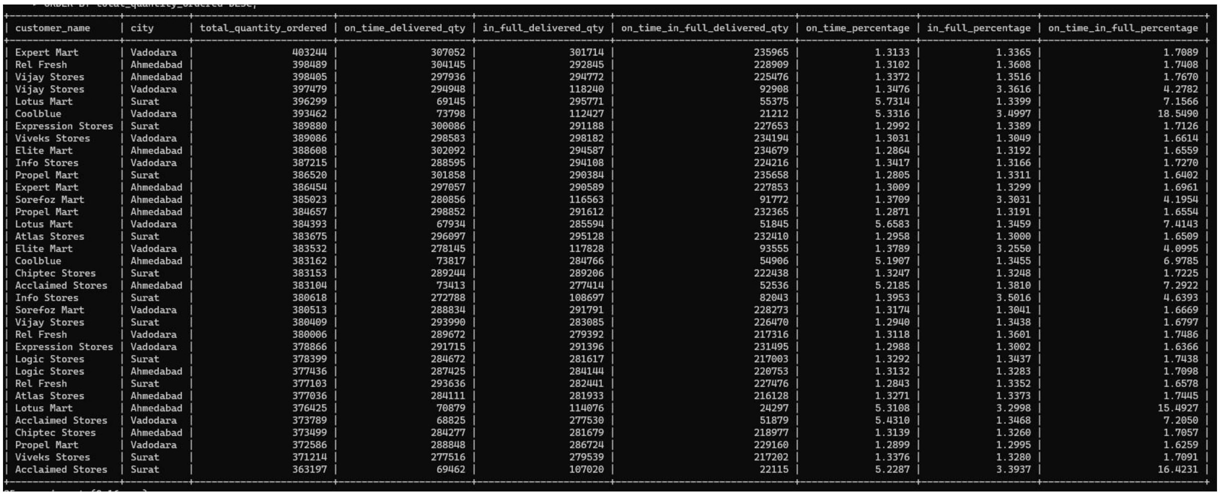
Summary

```
WITH customer_summary AS (
SELECT
  -- dim_customers.customer_id,
  dim_customers.customer_name,
  dim_customers.city,
  SUM(fact_order_lines.order_qty) as total_quantity_ordered,
  SUM(CASE WHEN fact_order_lines.agreed_delivery_date = fact_order_lines.actual_delivery_date THEN fact_order_lines.delivery_qty ELSE
  SUM(CASE WHEN fact_order_lines.delivery_qty = fact_order_lines.order_qty THEN fact_order_lines.delivery_qty ELSE 0 END) as in_full_d
  SUM(CASE WHEN fact_order_lines.agreed_delivery_date = fact_order_lines.actual_delivery_date AND fact_order_lines.delivery_qty = fact
FROM fact_order_lines
JOIN dim_customers ON fact_order_lines.customer_id = dim_customers.customer_id
GROUP BY dim_customers.customer_id, dim_customers.customer_name, dim_customers.city
)
SELECT
  customer_summary.*,
  (total_quantity_ordered/on_time_delivered_qty) as on_time_percentage,
 ( total_quantity_ordered/in_full_delivered_qty) as in_full_percentage,
 ( total_quantity_ordered/on_time_in_full_delivered_qty) as on_time_in_full_percentage
FROM customer_summary
ORDER BY total_quantity_ordered DESC;
```

This query will give you the top and bottom 5 customers by total quantity ordered, in full quantity ordered and 'on-time and in-full' quantity ordered. It will also give you the percentage of on-time, in-full and on-time and in-full deliveries for each customer.



```
WITH customer_summary AS (
    SELECT
  -- dim_customers.customer_id,
        dim_customers.customer_name,
        dim_customers.city,
        SUM(fact_order_lines.order_qty) as total_quantity_ordered,
        SUM(CASE WHEN fact_order_lines.agreed_delivery_date = fact_order_lines.actual_delivery_date THEN fact_order_lines.delivery_qty
                ELSE 0 END) as on_time_delivered_qty,
        SUM(CASE WHEN fact_order_lines.delivery_qty = fact_order_lines.order_qty THEN fact_order_lines.delivery_qty ELSE 0 END) as
                in_full_delivered_qty,
        SUM(CASE WHEN fact_order_lines.agreed_delivery_date = fact_order_lines.actual_delivery_date AND fact_order_lines.delivery_qty =
                fact_order_lines.order_qty THEN fact_order_lines.delivery_qty ELSE 0 END) as on_time_in_full_delivered_qty
    FROM
    fact_order_lines
    JOIN dim_customers
    ON fact_order_lines.customer_id = dim_customers.customer_id
    GROUP BY
        dim_customers.customer_id, dim_customers.customer_name, dim_customers.city
)
SELECT
  customer_summary.*,
  (total_quantity_ordered/on_time_delivered_qty) as on_time_percentage,
 ( total_quantity_ordered/in_full_delivered_qty) as in_full_percentage,
 ( total_quantity_ordered/on_time_in_full_delivered_qty) as on_time_in_full_percentage
FROM
    customer_summary
ORDER BY
    total_quantity_ordered DESC;
```

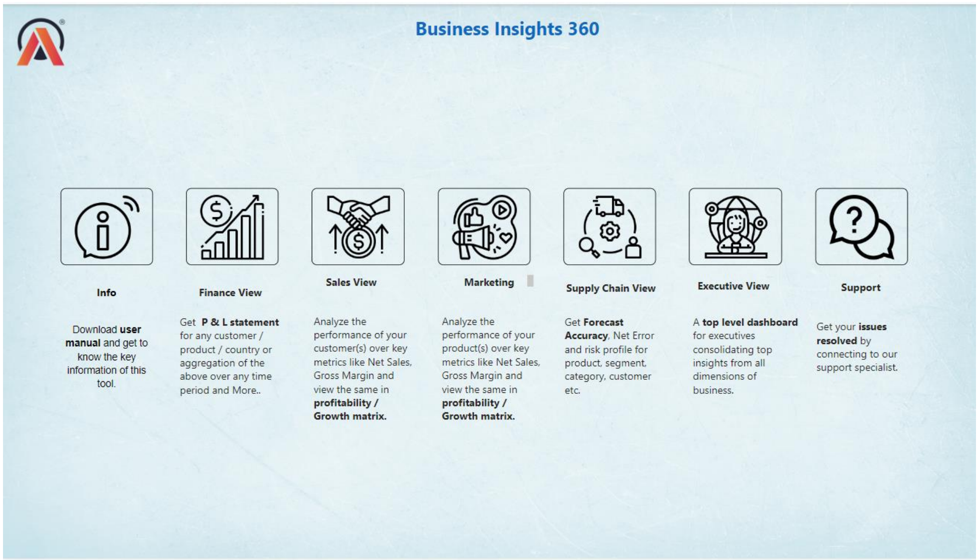


```
    -> ORDER BY total_quantity_ordered DESC;
+-------------------+-----------+------------------------+-----------------------+-----------------------+-------------------------------+--------------------+--------------------+----------------------------+
| customer_name     | city      | total_quantity_ordered | on_time_delivered_qty | in_full_delivered_qty | on_time_in_full_delivered_qty | on_time_percentage | in_full_percentage | on_time_in_full_percentage |
+-------------------+-----------+------------------------+-----------------------+-----------------------+-------------------------------+--------------------+--------------------+----------------------------+
| Expert Mart       | Vadodara  |                 403244 |                307052 |                301714 |                        235965 |             1.3133 |             1.3365 |                     1.7089 |
| Rel Fresh         | Ahmedabad |                 398489 |                304145 |                292845 |                        228909 |             1.3102 |             1.3608 |                     1.7408 |
| Vijay Stores      | Ahmedabad |                 398405 |                297936 |                294772 |                        225476 |             1.3372 |             1.3516 |                     1.7670 |
| Vijay Stores      | Vadodara  |                 397479 |                294948 |                118240 |                         92908 |             1.3476 |             3.3616 |                     4.2782 |
| Lotus Mart        | Surat     |                 396299 |                 69145 |                295771 |                         55375 |             5.7314 |             1.3399 |                     7.1566 |
| Coolblue          | Vadodara  |                 393462 |                 73798 |                112427 |                         21212 |             5.3316 |             3.4997 |                    18.5490 |
| Expression Stores | Surat     |                 389880 |                300086 |                291188 |                        227653 |             1.2992 |             1.3389 |                     1.7126 |
| Viveks Stores     | Vadodara  |                 389086 |                298583 |                298182 |                        234194 |             1.3031 |             1.3049 |                     1.6614 |
| Elite Mart        | Ahmedabad |                 388608 |                302092 |                294587 |                        234679 |             1.2864 |             1.3192 |                     1.6559 |
| Info Stores       | Vadodara  |                 387215 |                288595 |                294108 |                        224216 |             1.3417 |             1.3166 |                     1.7270 |
| Propel Mart       | Surat     |                 386520 |                301858 |                290384 |                        235658 |             1.2805 |             1.3311 |                     1.6402 |
| Expert Mart       | Ahmedabad |                 386454 |                297057 |                290589 |                        227853 |             1.3009 |             1.3299 |                     1.6961 |
| Sorefoz Mart      | Ahmedabad |                 385023 |                280856 |                116563 |                         91772 |             1.3709 |             3.3031 |                     4.1954 |
| Propel Mart       | Ahmedabad |                 384657 |                298852 |                291612 |                        232365 |             1.2871 |             1.3191 |                     1.6554 |
| Lotus Mart        | Vadodara  |                 384393 |                 67934 |                285594 |                         51845 |             5.6583 |             1.3459 |                     7.4143 |
| Atlas Stores      | Surat     |                 383675 |                296097 |                295128 |                        232410 |             1.2958 |             1.3000 |                     1.6509 |
| Elite Mart        | Vadodara  |                 383532 |                278145 |                117828 |                         93555 |             1.3789 |             3.2550 |                     4.0995 |
| Coolblue          | Ahmedabad |                 383162 |                 73817 |                284766 |                         54906 |             5.1907 |             1.3455 |                     6.9785 |
| Chiptec Stores    | Surat     |                 383153 |                289244 |                289206 |                        222438 |             1.3247 |             1.3248 |                     1.7225 |
| Acclaimed Stores  | Ahmedabad |                 383104 |                 73413 |                277414 |                         52536 |             5.2185 |             1.3810 |                     7.2922 |
| Info Stores       | Surat     |                 380618 |                272788 |                108697 |                         82043 |             1.3953 |             3.5016 |                     4.6393 |
| Sorefoz Mart      | Vadodara  |                 380513 |                288834 |                291791 |                        228273 |             1.3174 |             1.3041 |                     1.6669 |
| Vijay Stores      | Surat     |                 380409 |                293990 |                283085 |                        226470 |             1.2940 |             1.3438 |                     1.6797 |
| Rel Fresh         | Vadodara  |                 380006 |                289672 |                279392 |                        217316 |             1.3118 |             1.3601 |                     1.7486 |
| Expression Stores | Vadodara  |                 378866 |                291715 |                291396 |                        231495 |             1.2988 |             1.3002 |                     1.6366 |
| Logic Stores      | Surat     |                 378399 |                284672 |                281617 |                        217003 |             1.3292 |             1.3437 |                     1.7438 |
| Logic Stores      | Ahmedabad |                 377436 |                287425 |                284144 |                        220753 |             1.3132 |             1.3283 |                     1.7098 |
| Rel Fresh         | Surat     |                 377103 |                293636 |                282441 |                        227476 |             1.2843 |             1.3352 |                     1.6578 |
| Atlas Stores      | Ahmedabad |                 377036 |                284111 |                281933 |                        216128 |             1.3271 |             1.3373 |                     1.7445 |
| Lotus Mart        | Ahmedabad |                 376425 |                 70879 |                114076 |                         24297 |             5.3108 |             3.2998 |                    15.4927 |
| Acclaimed Stores  | Vadodara  |                 373789 |                 68825 |                277530 |                         51879 |             5.4310 |             1.3468 |                     7.2050 |
| Chiptec Stores    | Ahmedabad |                 373499 |                284277 |                281679 |                        218977 |             1.3139 |             1.3260 |                     1.7057 |
| Propel Mart       | Vadodara  |                 372586 |                288848 |                286724 |                        229160 |             1.2899 |             1.2995 |                     1.6259 |
| Viveks Stores     | Surat     |                 371214 |                277516 |                279539 |                        217202 |             1.3376 |             1.3280 |                     1.7091 |
| Acclaimed Stores  | Surat     |                 363197 |                 69462 |                107020 |                         22115 |             5.2287 |             3.3937 |                    16.4231 |
+-------------------+-----------+------------------------+-----------------------+-----------------------+-------------------------------+--------------------+--------------------+----------------------------+
```

- The total quantity ordered by all customers is quite large, with the lowest being 38,050 and the highest being 403,244.
- In terms of on-time delivery, the percentage ranges from 1.28 to 5.73, with Expert Mart in Vadodara having the highest on-time percentage of 1.71.
- In terms of in-full delivery, the percentage ranges from 1.3166 to 3.5016, with Expert Mart in Ahmedabad having the highest in-full percentage of 1.3399.
- For on-time and in-full delivery combined, the percentage ranges from 1.6559 to 4.6393 with Expert Mart in Ahmedabad having the highest percentage of 1.6961.
- There are a few outliers, such as Lotus Mart in Surat and Coolblue in Vadodara, which have significantly lower on-time and in-full percentages compared to the other customers.
- There are also a few customers, such as Sorefoz Mart in Vadodara and Info Stores in Surat, that have much higher in-full percentages than on-time percentages.
- The city with the most customers is Ahmedabad and Vadodara.

Dashboard:

1. Overview:



Business Insights 360

Info

Download **user manual** and get to know the key information of this tool.

Finance View

Get **P & L statement** for any customer / product / country or aggregation of the above over any time period and More..

Sales View

Analyze the performance of your customer(s) over key metrics like Net Sales, Gross Margin and view the same in **profitability / Growth matrix.**

Marketing

Analyze the performance of your product(s) over key metrics like Net Sales, Gross Margin and view the same in **profitability / Growth matrix.**

Supply Chain View

Get **Forecast Accuracy**, Net Error and risk profile for product, segment, category, customer etc.

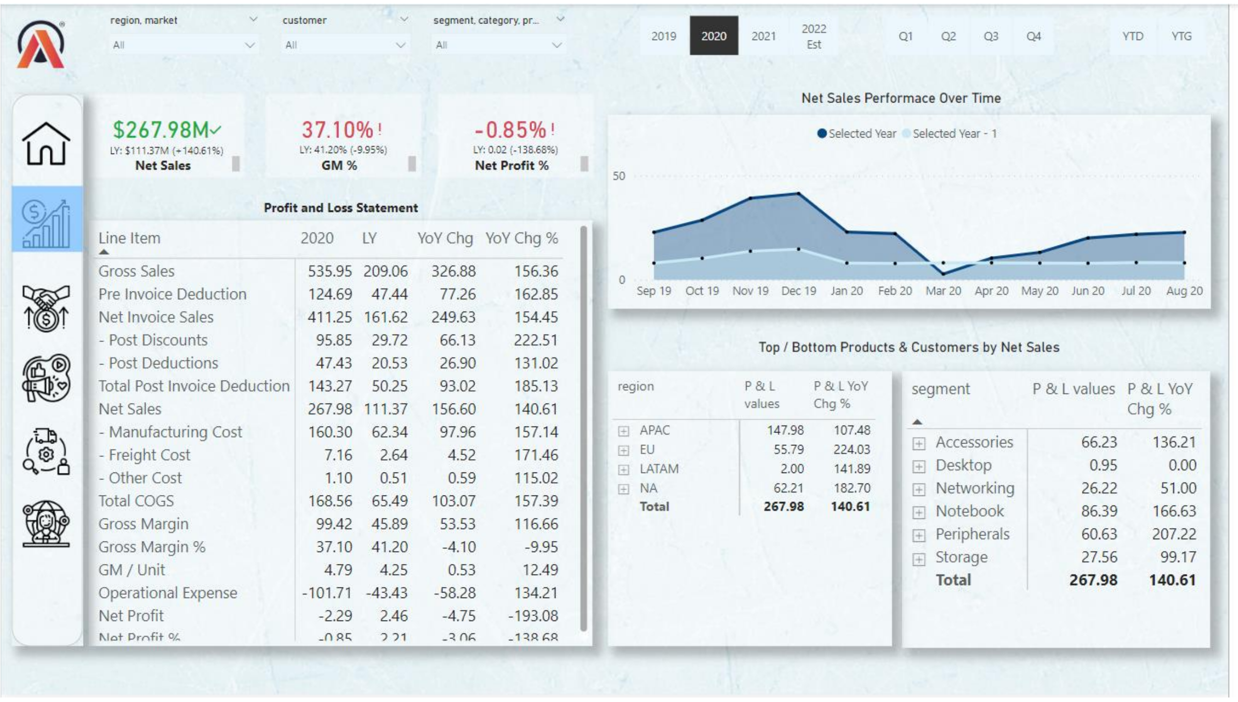
Executive View

A **top level dashboard** for executives consolidating top insights from all dimensions of business.

Support

Get your **issues resolved** by connecting to our support specialist.

2. Finance View:



$267.98M✓
LY: $111.37M (+140.61%)
Net Sales

37.10%!
LY: 41.20% (-9.95%)
GM %

-0.85%!
LY: 0.02 (-138.68%)
Net Profit %

Profit and Loss Statement

Line Item	2020	LY	YoY Chg	YoY Chg %
Gross Sales	535.95	209.06	326.88	156.36
Pre Invoice Deduction	124.69	47.44	77.26	162.85
Net Invoice Sales	411.25	161.62	249.63	154.45
- Post Discounts	95.85	29.72	66.13	222.51
- Post Deductions	47.43	20.53	26.90	131.02
Total Post Invoice Deduction	143.27	50.25	93.02	185.13
Net Sales	267.98	111.37	156.60	140.61
- Manufacturing Cost	160.30	62.34	97.96	157.14
- Freight Cost	7.16	2.64	4.52	171.46
- Other Cost	1.10	0.51	0.59	115.02
Total COGS	168.56	65.49	103.07	157.39
Gross Margin	99.42	45.89	53.53	116.66
Gross Margin %	37.10	41.20	-4.10	-9.95
GM / Unit	4.79	4.25	0.53	12.49
Operational Expense	-101.71	-43.43	-58.28	134.21
Net Profit	-2.29	2.46	-4.75	-193.08
Net Profit %	-0.85	2.21	-3.06	-138.68

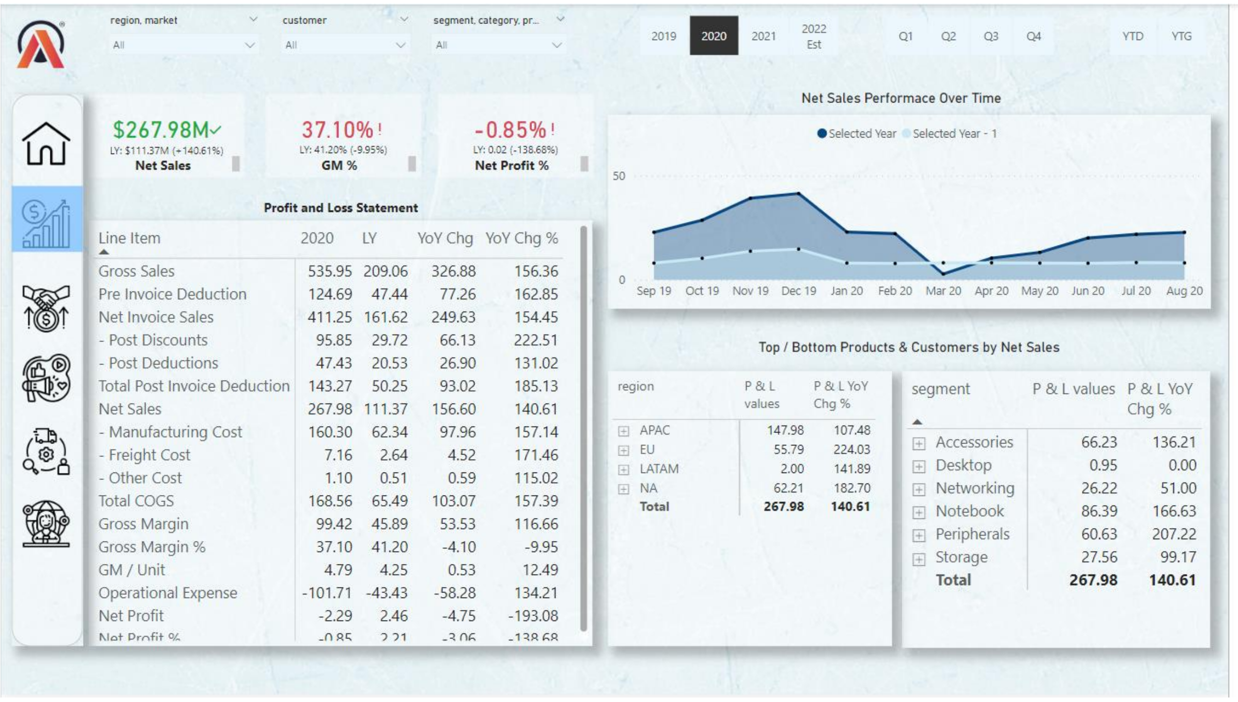
Net Sales Performace Over Time

Top / Bottom Products & Customers by Net Sales

region	P & L values	P & L YoY Chg %
⊞ APAC	147.98	107.48
⊞ EU	55.79	224.03
⊞ LATAM	2.00	141.89
⊞ NA	62.21	182.70
Total	**267.98**	**140.61**

segment	P & L values	P & L YoY Chg %
⊞ Accessories	66.23	136.21
⊞ Desktop	0.95	0.00
⊞ Networking	26.22	51.00
⊞ Notebook	86.39	166.63
⊞ Peripherals	60.63	207.22
⊞ Storage	27.56	99.17
Total	**267.98**	**140.61**

3. Sales View:



$267.98M✓
LY: $111.37M (+140.61%)
Net Sales

37.10%!
LY: 41.20% (-9.95%)
GM %

-0.85%!
LY: 0.02 (-138.68%)
Net Profit %

Profit and Loss Statement

Line Item	2020	LY	YoY Chg	YoY Chg %
Gross Sales	535.95	209.06	326.88	156.36
Pre Invoice Deduction	124.69	47.44	77.26	162.85
Net Invoice Sales	411.25	161.62	249.63	154.45
- Post Discounts	95.85	29.72	66.13	222.51
- Post Deductions	47.43	20.53	26.90	131.02
Total Post Invoice Deduction	143.27	50.25	93.02	185.13
Net Sales	267.98	111.37	156.60	140.61
- Manufacturing Cost	160.30	62.34	97.96	157.14
- Freight Cost	7.16	2.64	4.52	171.46
- Other Cost	1.10	0.51	0.59	115.02
Total COGS	168.56	65.49	103.07	157.39
Gross Margin	99.42	45.89	53.53	116.66
Gross Margin %	37.10	41.20	-4.10	-9.95
GM / Unit	4.79	4.25	0.53	12.49
Operational Expense	-101.71	-43.43	-58.28	134.21
Net Profit	-2.29	2.46	-4.75	-193.08
Net Profit %	-0.85	2.21	-3.06	-138.68

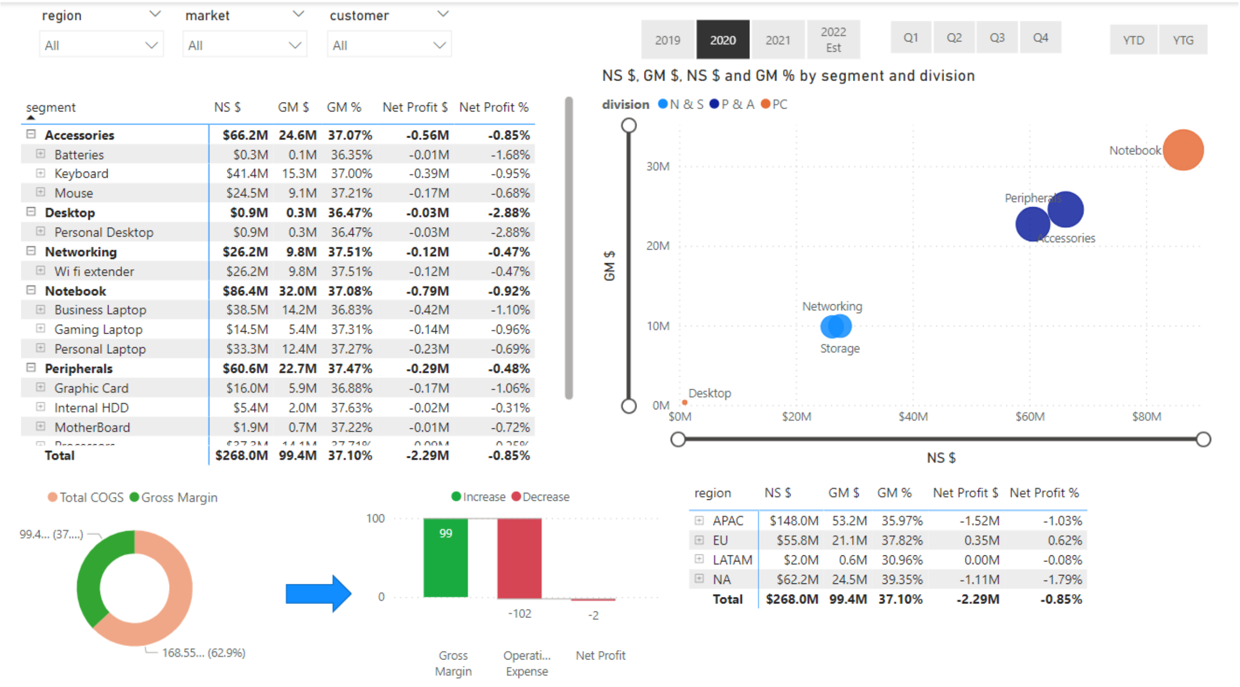
Net Sales Performace Over Time

Top / Bottom Products & Customers by Net Sales

region	P & L values	P & L YoY Chg %
⊞ APAC	147.98	107.48
⊞ EU	55.79	224.03
⊞ LATAM	2.00	141.89
⊞ NA	62.21	182.70
Total	**267.98**	**140.61**

segment	P & L values	P & L YoY Chg %
⊞ Accessories	66.23	136.21
⊞ Desktop	0.95	0.00
⊞ Networking	26.22	51.00
⊞ Notebook	86.39	166.63
⊞ Peripherals	60.63	207.22
⊞ Storage	27.56	99.17
Total	**267.98**	**140.61**

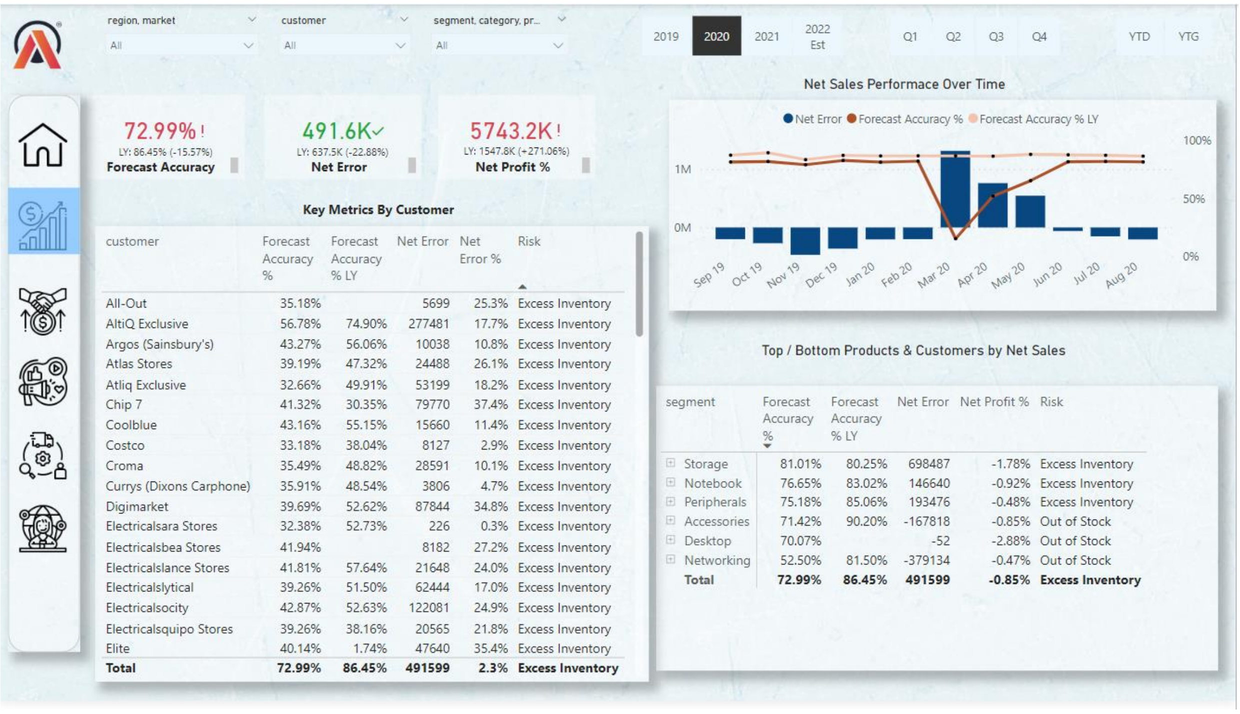
4. Marketing View:



segment	NS $	GM $	GM %	Net Profit $	Net Profit %
⊟ **Accessories**	**$66.2M**	**24.6M**	**37.07%**	**-0.56M**	**-0.85%**
⊞ Batteries	$0.3M	0.1M	36.35%	-0.01M	-1.68%
⊞ Keyboard	$41.4M	15.3M	37.00%	-0.39M	-0.95%
⊞ Mouse	$24.5M	9.1M	37.21%	-0.17M	-0.68%
⊟ **Desktop**	**$0.9M**	**0.3M**	**36.47%**	**-0.03M**	**-2.88%**
⊞ Personal Desktop	$0.9M	0.3M	36.47%	-0.03M	-2.88%
⊟ **Networking**	**$26.2M**	**9.8M**	**37.51%**	**-0.12M**	**-0.47%**
⊞ Wi fi extender	$26.2M	9.8M	37.51%	-0.12M	-0.47%
⊟ **Notebook**	**$86.4M**	**32.0M**	**37.08%**	**-0.79M**	**-0.92%**
⊞ Business Laptop	$38.5M	14.2M	36.83%	-0.42M	-1.10%
⊞ Gaming Laptop	$14.5M	5.4M	37.31%	-0.14M	-0.96%
⊞ Personal Laptop	$33.3M	12.4M	37.27%	-0.23M	-0.69%
⊟ **Peripherals**	**$60.6M**	**22.7M**	**37.47%**	**-0.29M**	**-0.48%**
⊞ Graphic Card	$16.0M	5.9M	36.88%	-0.17M	-1.06%
⊞ Internal HDD	$5.4M	2.0M	37.63%	-0.02M	-0.31%
⊞ MotherBoard	$1.9M	0.7M	37.22%	-0.01M	-0.72%
⊞ Processors	$37.3M	14.1M	37.71%	[illegible]	[illegible]
Total	**$268.0M**	**99.4M**	**37.10%**	**-2.29M**	**-0.85%**

region	NS $	GM $	GM %	Net Profit $	Net Profit %
⊞ APAC	$148.0M	53.2M	35.97%	-1.52M	-1.03%
⊞ EU	$55.8M	21.1M	37.82%	0.35M	0.62%
⊞ LATAM	$2.0M	0.6M	30.96%	0.00M	-0.08%
⊞ NA	$62.2M	24.5M	39.35%	-1.11M	-1.79%
Total	**$268.0M**	**99.4M**	**37.10%**	**-2.29M**	**-0.85%**

5. Supply Chain View:



72.99% !
LY: 86.45% (-15.57%)
Forecast Accuracy

491.6K✓
LY: 637.5K (-22.88%)
Net Error

5743.2K !
LY: 1547.8K (+271.06%)
Net Profit %

Key Metrics By Customer

customer	Forecast Accuracy %	Forecast Accuracy % LY	Net Error	Net Error %	Risk
All-Out	35.18%		5699	25.3%	Excess Inventory
AltiQ Exclusive	56.78%	74.90%	277481	17.7%	Excess Inventory
Argos (Sainsbury's)	43.27%	56.06%	10038	10.8%	Excess Inventory
Atlas Stores	39.19%	47.32%	24488	26.1%	Excess Inventory
Atliq Exclusive	32.66%	49.91%	53199	18.2%	Excess Inventory
Chip 7	41.32%	30.35%	79770	37.4%	Excess Inventory
Coolblue	43.16%	55.15%	15660	11.4%	Excess Inventory
Costco	33.18%	38.04%	8127	2.9%	Excess Inventory
Croma	35.49%	48.82%	28591	10.1%	Excess Inventory
Currys (Dixons Carphone)	35.91%	48.54%	3806	4.7%	Excess Inventory
Digimarket	39.69%	52.62%	87844	34.8%	Excess Inventory
Electricalsara Stores	32.38%	52.73%	226	0.3%	Excess Inventory
Electricalsbea Stores	41.94%		8182	27.2%	Excess Inventory
Electricalslance Stores	41.81%	57.64%	21648	24.0%	Excess Inventory
Electricalslytical	39.26%	51.50%	62444	17.0%	Excess Inventory
Electricalsocity	42.87%	52.63%	122081	24.9%	Excess Inventory
Electricalsquipo Stores	39.26%	38.16%	20565	21.8%	Excess Inventory
Elite	40.14%	1.74%	47640	35.4%	Excess Inventory
Total	**72.99%**	**86.45%**	**491599**	**2.3%**	**Excess Inventory**

Top / Bottom Products & Customers by Net Sales

segment	Forecast Accuracy %	Forecast Accuracy % LY	Net Error	Net Profit %	Risk
⊞ Storage	81.01%	80.25%	698487	-1.78%	Excess Inventory
⊞ Notebook	76.65%	83.02%	146640	-0.92%	Excess Inventory
⊞ Peripherals	75.18%	85.06%	193476	-0.48%	Excess Inventory
⊞ Accessories	71.42%	90.20%	-167818	-0.85%	Out of Stock
⊞ Desktop	70.07%		-52	-2.88%	Out of Stock
⊞ Networking	52.50%	81.50%	-379134	-0.47%	Out of Stock
Total	**72.99%**	**86.45%**	**491599**	**-0.85%**	**Excess Inventory**