Student Name: Rudasingwa Nshuti Bonheur 27367
Date: 2025-09-24

Step 1 Problem Definition

Coffee distribution company

IKAZE coffee company that sells high-quality coffee has both physical shops and an online store, and it serves customers in different places (like Kigali, Butare, and other regions). Sales & Marketing team wants to understand which products and which customers drive revenue, so they can better manage stocks and run targeted promotions.

What is needed
We have transactional sales data across **regions, customers, and products**. The challenge is to calculate:

- Top products per region and quarter
- Running monthly totals and month-over-month growth
- Customer segmentation into quartiles

Expected Outcome
Produce SQL-driven insights to:

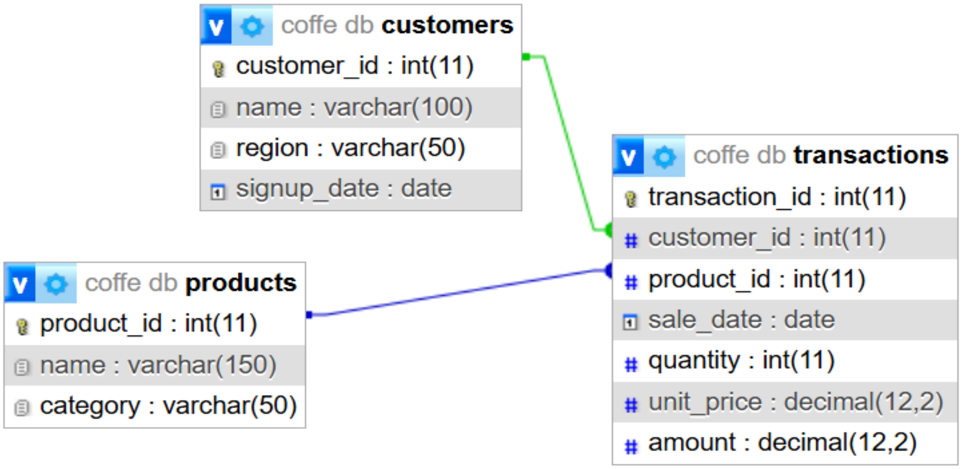
1. Identify **top products per region and quarter**
2. Compute **running monthly sales totals** and **MoM growth**
3. Segment customers into **revenue quartiles**

Step 2 Success Criteria (Measurable Goals)

1. Top 5 products per region & quarter → RANK()
2. Running monthly sales totals → SUM() OVER()
3. Month-over-Month (MoM) growth → LAG()
4. Customer revenue quartiles → NTILE(4)
5. 3-month moving average → AVG() OVER()

Step 3 Database Schema

ER Diagram

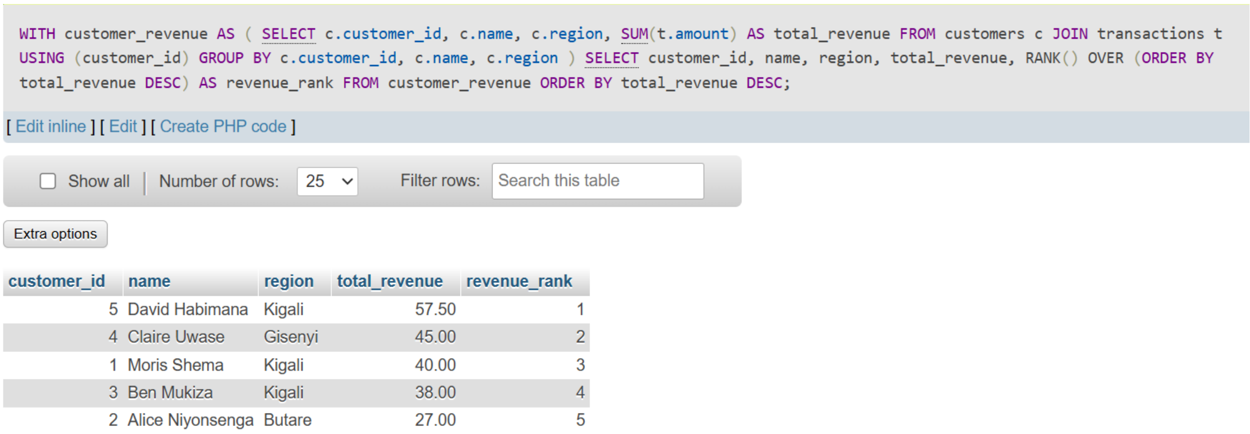


Step 4 Window Functions Implementation

4.1 Ranking —Top Customers and Products

4.1a Top Customers by Revenue

```
WITH customer_revenue AS (
 SELECT c.customer_id, c.name, c.region, SUM(t.amount) AS total_revenue
 FROM customers c
 JOIN transactions t USING (customer_id)
 GROUP BY c.customer_id, c.name, c.region
)
SELECT customer_id, name, region, total_revenue,
       RANK() OVER (ORDER BY total_revenue DESC) AS revenue_rank
FROM customer_revenue
ORDER BY total_revenue DESC;
```



```
WITH customer_revenue AS ( SELECT c.customer_id, c.name, c.region, SUM(t.amount) AS total_revenue FROM customers c JOIN transactions t
USING (customer_id) GROUP BY c.customer_id, c.name, c.region ) SELECT customer_id, name, region, total_revenue, RANK() OVER (ORDER BY
total_revenue DESC) AS revenue_rank FROM customer_revenue ORDER BY total_revenue DESC;
```

[Edit inline] [Edit] [Create PHP code]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options

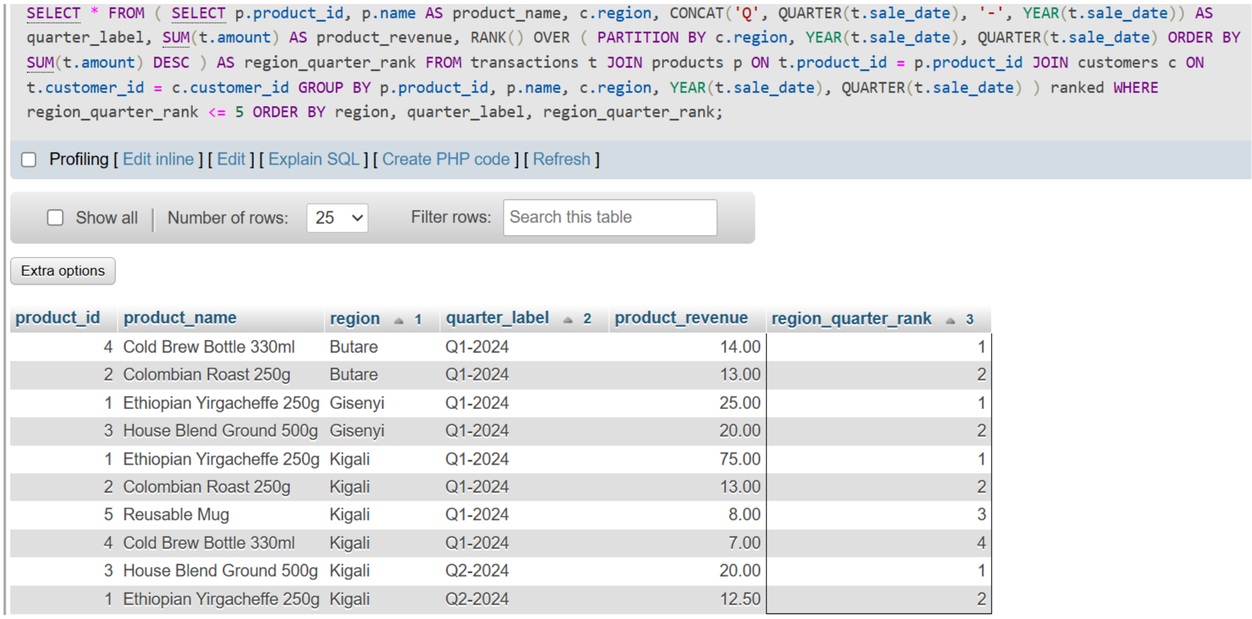
customer_id	name	region	total_revenue	revenue_rank
5	David Habimana	Kigali	57.50	1
4	Claire Uwase	Gisenyi	45.00	2
1	Moris Shema	Kigali	40.00	3
3	Ben Mukiza	Kigali	38.00	4
2	Alice Niyonsenga	Butare	27.00	5

Interpretation:

Ranks customers by total spending. Customers with equal revenue share the same rank. Useful for **targeted loyalty campaigns**.

Top 5 Products per Region & Quarter

```
-- Top 5 products per region and quarter (MySQL 8)

SELECT *
FROM (
  SELECT
    p.product_id,
    p.name AS product_name,
    c.region,
    CONCAT('Q', QUARTER(t.sale_date), '-', YEAR(t.sale_date)) AS quarter_label,
    SUM(t.amount) AS product_revenue,
    RANK() OVER (
      PARTITION BY c.region, YEAR(t.sale_date), QUARTER(t.sale_date)
      ORDER BY SUM(t.amount) DESC
    ) AS region_quarter_rank
  FROM transactions t
  JOIN products p ON t.product_id = p.product_id
  JOIN customers c ON t.customer_id = c.customer_id
  GROUP BY p.product_id, p.name, c.region, YEAR(t.sale_date), QUARTER(t.sale_date)
) ranked
WHERE region_quarter_rank <= 5
ORDER BY region, quarter_label, region_quarter_rank;
```



```
SELECT * FROM ( SELECT p.product_id, p.name AS product_name, c.region, CONCAT('Q', QUARTER(t.sale_date), '-', YEAR(t.sale_date)) AS
quarter_label, SUM(t.amount) AS product_revenue, RANK() OVER ( PARTITION BY c.region, YEAR(t.sale_date), QUARTER(t.sale_date) ORDER BY
SUM(t.amount) DESC ) AS region_quarter_rank FROM transactions t JOIN products p ON t.product_id = p.product_id JOIN customers c ON
t.customer_id = c.customer_id GROUP BY p.product_id, p.name, c.region, YEAR(t.sale_date), QUARTER(t.sale_date) ) ranked WHERE
region_quarter_rank <= 5 ORDER BY region, quarter_label, region_quarter_rank;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

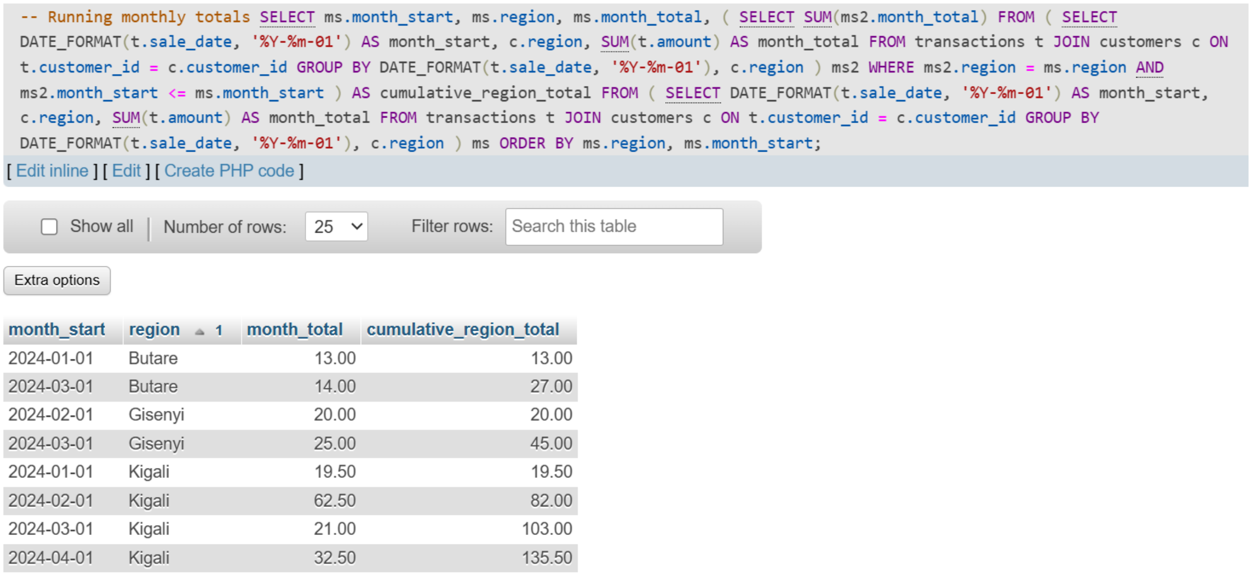
product_id	product_name	region ▲ 1	quarter_label ▲ 2	product_revenue	region_quarter_rank ▲ 3
4	Cold Brew Bottle 330ml	Butare	Q1-2024	14.00	1
2	Colombian Roast 250g	Butare	Q1-2024	13.00	2
1	Ethiopian Yirgacheffe 250g	Gisenyi	Q1-2024	25.00	1
3	House Blend Ground 500g	Gisenyi	Q1-2024	20.00	2
1	Ethiopian Yirgacheffe 250g	Kigali	Q1-2024	75.00	1
2	Colombian Roast 250g	Kigali	Q1-2024	13.00	2
5	Reusable Mug	Kigali	Q1-2024	8.00	3
4	Cold Brew Bottle 330ml	Kigali	Q1-2024	7.00	4
3	House Blend Ground 500g	Kigali	Q2-2024	20.00	1
1	Ethiopian Yirgacheffe 250g	Kigali	Q2-2024	12.50	2

Interpretation:

Shows top 5 products per region and quarter, ties handled correctly. Helps **promotions & inventory planning**.

4.2 Aggregate — Running Monthly Totals & Moving Average

```
SELECT
ms.month_start,
ms.region,
ms.month_total,
(
SELECT SUM(ms2.month_total)
FROM (
SELECT DATE_FORMAT(t.sale_date, '%Y-%m-01') AS month_start,
c.region,
SUM(t.amount) AS month_total
FROM transactions t
JOIN customers c ON t.customer_id = c.customer_id
GROUP BY DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region
) ms2
WHERE ms2.region = ms.region
AND ms2.month_start <= ms.month_start
) AS cumulative_region_total
FROM (
SELECT DATE_FORMAT(t.sale_date, '%Y-%m-01') AS month_start,
c.region,
SUM(t.amount) AS month_total
FROM transactions t
JOIN customers c ON t.customer_id = c.customer_id
GROUP BY DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region
) ms
ORDER BY ms.region, ms.month_start;
```



```
-- Running monthly totals SELECT ms.month_start, ms.region, ms.month_total, ( SELECT SUM(ms2.month_total) FROM ( SELECT
DATE_FORMAT(t.sale_date, '%Y-%m-01') AS month_start, c.region, SUM(t.amount) AS month_total FROM transactions t JOIN customers c ON
t.customer_id = c.customer_id GROUP BY DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region ) ms2 WHERE ms2.region = ms.region AND
ms2.month_start <= ms.month_start ) AS cumulative_region_total FROM ( SELECT DATE_FORMAT(t.sale_date, '%Y-%m-01') AS month_start,
c.region, SUM(t.amount) AS month_total FROM transactions t JOIN customers c ON t.customer_id = c.customer_id GROUP BY
DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region ) ms ORDER BY ms.region, ms.month_start;
```

[Edit inline] [Edit] [Create PHP code]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options

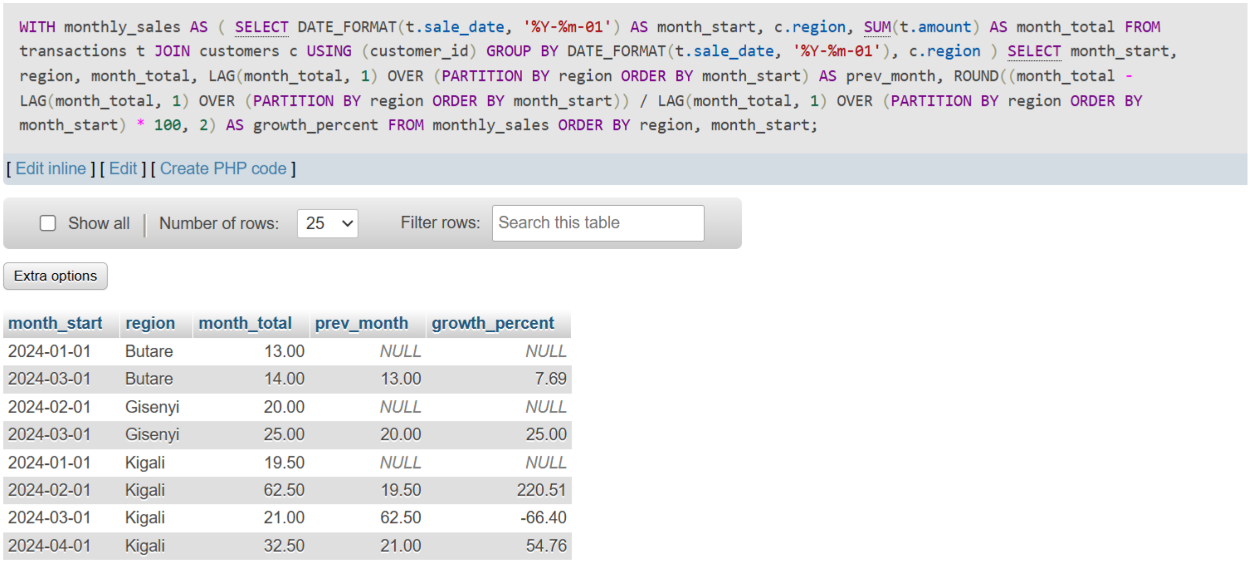
month_start	region ▲ 1	month_total	cumulative_region_total
2024-01-01	Butare	13.00	13.00
2024-03-01	Butare	14.00	27.00
2024-02-01	Gisenyi	20.00	20.00
2024-03-01	Gisenyi	25.00	45.00
2024-01-01	Kigali	19.50	19.50
2024-02-01	Kigali	62.50	82.00
2024-03-01	Kigali	21.00	103.00
2024-04-01	Kigali	32.50	135.50

Interpretation:

Cumulative totals show sales growth; 3-month moving averages smooth seasonal fluctuations.

4.3 Navigation — Month-over-Month Growth

```
WITH monthly_sales AS (
  SELECT DATE_FORMAT(t.sale_date, '%Y-%m-01') AS month_start,
         c.region, SUM(t.amount) AS month_total
  FROM transactions t
  JOIN customers c USING (customer_id)
  GROUP BY DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region
)
SELECT month_start, region, month_total,
       LAG(month_total, 1) OVER (PARTITION BY region ORDER BY month_start) AS prev_month,
       ROUND((month_total - LAG(month_total, 1) OVER (PARTITION BY region ORDER BY month_start))
             / LAG(month_total, 1) OVER (PARTITION BY region ORDER BY month_start) * 100, 2) AS growth_percent
FROM monthly_sales
ORDER BY region, month_start;
```


[Edit inline] [Edit] [Create PHP code]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options

```
WITH monthly_sales AS ( SELECT DATE_FORMAT(t.sale_date, '%Y-%m-01') AS month_start, c.region, SUM(t.amount) AS month_total FROM
transactions t JOIN customers c USING (customer_id) GROUP BY DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region ) SELECT month_start,
region, month_total, LAG(month_total, 1) OVER (PARTITION BY region ORDER BY month_start) AS prev_month, ROUND((month_total -
LAG(month_total, 1) OVER (PARTITION BY region ORDER BY month_start)) / LAG(month_total, 1) OVER (PARTITION BY region ORDER BY
month_start) * 100, 2) AS growth_percent FROM monthly_sales ORDER BY region, month_start;
```

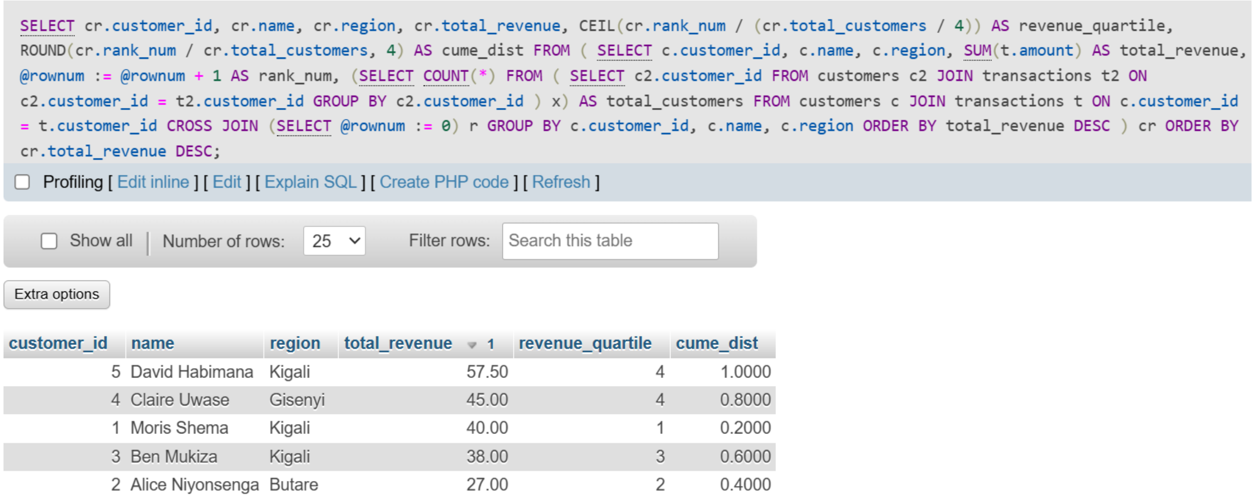
month_start	region	month_total	prev_month	growth_percent
2024-01-01	Butare	13.00	NULL	NULL
2024-03-01	Butare	14.00	13.00	7.69
2024-02-01	Gisenyi	20.00	NULL	NULL
2024-03-01	Gisenyi	25.00	20.00	25.00
2024-01-01	Kigali	19.50	NULL	NULL
2024-02-01	Kigali	62.50	19.50	220.51
2024-03-01	Kigali	21.00	62.50	-66.40
2024-04-01	Kigali	32.50	21.00	54.76

Interpretation:

LAG() calculates previous month sales; growth % identifies **increasing or decreasing trends**.

4.4 Distribution — Customer Revenue Quartiles

```
SELECT
  cr.customer_id,
  cr.name,
  cr.region,
  cr.total_revenue,

  CEIL(cr.rank_num / (cr.total_customers / 4)) AS revenue_quartile,

  ROUND(cr.rank_num / cr.total_customers, 4) AS cume_dist

FROM (
  SELECT
    c.customer_id,
    c.name,
    c.region,
    SUM(t.amount) AS total_revenue,

    @rownum := @rownum + 1 AS rank_num,

    (SELECT COUNT(*)
     FROM (
       SELECT c2.customer_id
       FROM customers c2
       JOIN transactions t2 ON c2.customer_id = t2.customer_id
       GROUP BY c2.customer_id
     ) x) AS total_customers

  FROM customers c
  JOIN transactions t ON c.customer_id = t.customer_id
  CROSS JOIN (SELECT @rownum := 0) r
  GROUP BY c.customer_id, c.name, c.region
  ORDER BY total_revenue DESC
) cr
ORDER BY cr.total_revenue DESC;
```



```
SELECT cr.customer_id, cr.name, cr.region, cr.total_revenue, CEIL(cr.rank_num / (cr.total_customers / 4)) AS revenue_quartile,
ROUND(cr.rank_num / cr.total_customers, 4) AS cume_dist FROM ( SELECT c.customer_id, c.name, c.region, SUM(t.amount) AS total_revenue,
@rownum := @rownum + 1 AS rank_num, (SELECT COUNT(*) FROM ( SELECT c2.customer_id FROM customers c2 JOIN transactions t2 ON
c2.customer_id = t2.customer_id GROUP BY c2.customer_id ) x) AS total_customers FROM customers c JOIN transactions t ON c.customer_id
= t.customer_id CROSS JOIN (SELECT @rownum := 0) r GROUP BY c.customer_id, c.name, c.region ORDER BY total_revenue DESC ) cr ORDER BY
cr.total_revenue DESC;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
☐ Show all | Number of rows: 25
Filter rows: Search this table
Extra options

customer_id	name	region	total_revenue ▼ 1	revenue_quartile	cume_dist
5	David Habimana	Kigali	57.50	4	1.0000
4	Claire Uwase	Gisenyi	45.00	4	0.8000
1	Moris Shema	Kigali	40.00	1	0.2000
3	Ben Mukiza	Kigali	38.00	3	0.6000
2	Alice Niyonsenga	Butare	27.00	2	0.4000

Interpretation:

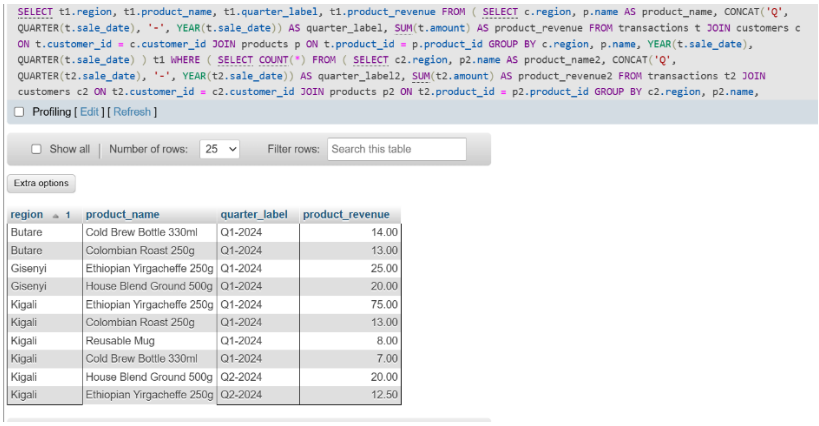
Quartiles show top/bottom revenue customers; helps **segment marketing strategies**.

Step 6 — Results Analysis

6.1 Descriptive Analysis

- Top customers consistently generate high revenue.
- Certain products dominate the top 5 per region/quarter.
- Cumulative monthly sales increase steadily.
- MoM growth varies by month and region.
- Customer quartiles highlight revenue concentration among top 25%.

Top 5 products per region per quarter

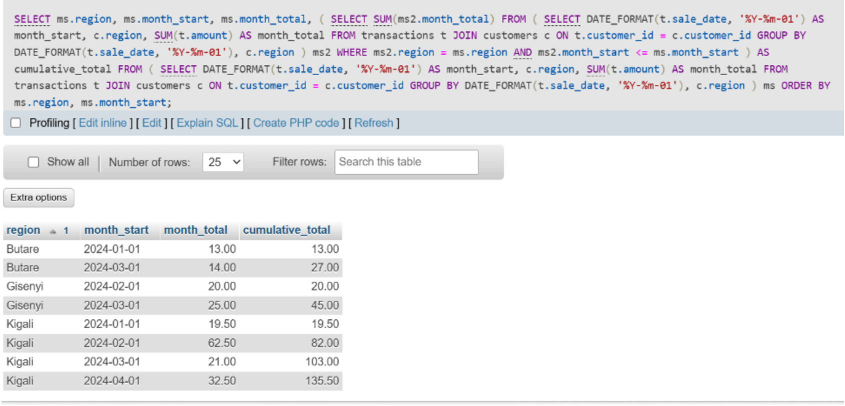


```
SELECT t1.region, t1.product_name, t1.quarter_label, t1.product_revenue FROM ( SELECT c.region, p.name AS product_name, CONCAT('Q',
QUARTER(t.sale_date), '-', YEAR(t.sale_date)) AS quarter_label, SUM(t.amount) AS product_revenue FROM transactions t JOIN customers c
ON t.customer_id = c.customer_id JOIN products p ON t.product_id = p.product_id GROUP BY c.region, p.name, YEAR(t.sale_date),
QUARTER(t.sale_date) ) t1 WHERE ( SELECT COUNT(*) FROM ( SELECT c2.region, p2.name AS product_name2, CONCAT('Q',
QUARTER(t2.sale_date), '-', YEAR(t2.sale_date)) AS quarter_label2, SUM(t2.amount) AS product_revenue2 FROM transactions t2 JOIN
customers c2 ON t2.customer_id = c2.customer_id JOIN products p2 ON t2.product_id = p2.product_id GROUP BY c2.region, p2.name,
```

☐ Profiling [Edit] [Refresh]
☐ Show all | Number of rows: 25
Filter rows: Search this table
Extra options

region ▲ 1	product_name	quarter_label	product_revenue
Butare	Cold Brew Bottle 330ml	Q1-2024	14.00
Butare	Colombian Roast 250g	Q1-2024	13.00
Gisenyi	Ethiopian Yirgacheffe 250g	Q1-2024	25.00
Gisenyi	House Blend Ground 500g	Q1-2024	20.00
Kigali	Ethiopian Yirgacheffe 250g	Q1-2024	75.00
Kigali	Colombian Roast 250g	Q1-2024	13.00
Kigali	Reusable Mug	Q1-2024	8.00
Kigali	Cold Brew Bottle 330ml	Q1-2024	7.00
Kigali	House Blend Ground 500g	Q2-2024	20.00
Kigali	Ethiopian Yirgacheffe 250g	Q2-2024	12.50

Cumulative Monthly Sales per region

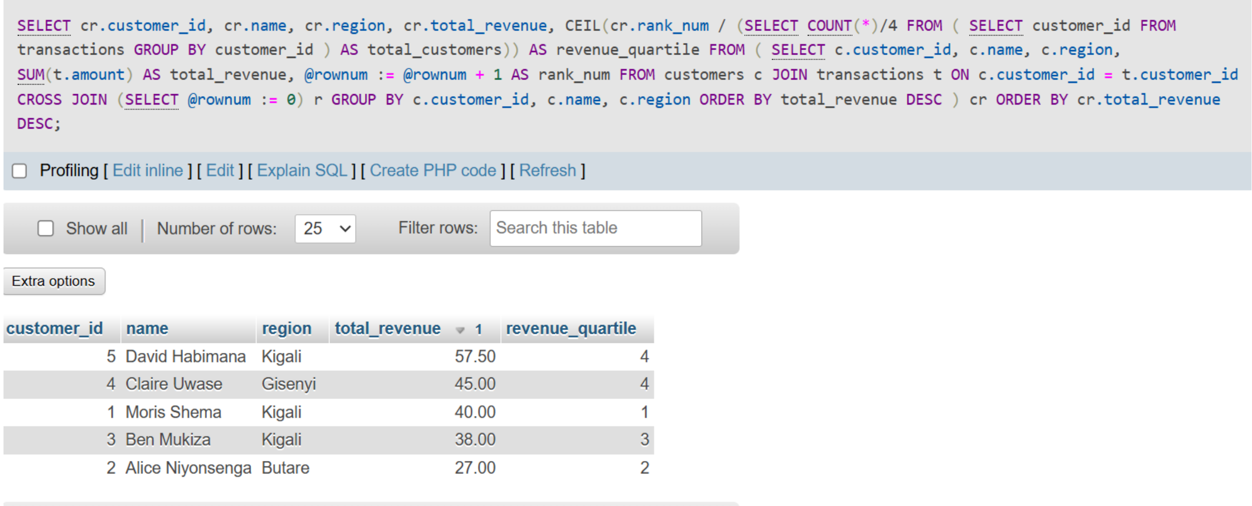


```
SELECT ms.region, ms.month_start, ms.month_total, ( SELECT SUM(ms2.month_total) FROM ( SELECT DATE_FORMAT(t.sale_date, '%Y-%m-01') AS
month_start, c.region, SUM(t.amount) AS month_total FROM transactions t JOIN customers c ON t.customer_id = c.customer_id GROUP BY
DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region ) ms2 WHERE ms2.region = ms.region AND ms2.month_start <= ms.month_start ) AS
cumulative_total FROM ( SELECT DATE_FORMAT(t.sale_date, '%Y-%m-01') AS month_start, c.region, SUM(t.amount) AS month_total FROM
transactions t JOIN customers c ON t.customer_id = c.customer_id GROUP BY DATE_FORMAT(t.sale_date, '%Y-%m-01'), c.region ) ms ORDER BY
ms.region, ms.month_start;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

region ▲ 1	month_start	month_total	cumulative_total
Butare	2024-01-01	13.00	13.00
Butare	2024-03-01	14.00	27.00
Gisenyi	2024-02-01	20.00	20.00
Gisenyi	2024-03-01	25.00	45.00
Kigali	2024-01-01	19.50	19.50
Kigali	2024-02-01	62.50	82.00
Kigali	2024-03-01	21.00	103.00
Kigali	2024-04-01	32.50	135.50

Cumulative Monthly Sales



```
SELECT cr.customer_id, cr.name, cr.region, cr.total_revenue, CEIL(cr.rank_num / (SELECT COUNT(*)/4 FROM ( SELECT customer_id FROM
transactions GROUP BY customer_id ) AS total_customers)) AS revenue_quartile FROM ( SELECT c.customer_id, c.name, c.region,
SUM(t.amount) AS total_revenue, @rownum := @rownum + 1 AS rank_num FROM customers c JOIN transactions t ON c.customer_id = t.customer_id
CROSS JOIN (SELECT @rownum := 0) r GROUP BY c.customer_id, c.name, c.region ORDER BY total_revenue DESC ) cr ORDER BY cr.total_revenue
DESC;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

customer_id	name	region	total_revenue ▼ 1	revenue_quartile
5	David Habimana	Kigali	57.50	4
4	Claire Uwase	Gisenyi	45.00	4
1	Moris Shema	Kigali	40.00	1
3	Ben Mukiza	Kigali	38.00	3
2	Alice Niyonsenga	Butare	27.00	2

6.2 Why it happened

- High revenue from top customers and products drives trends.
- Seasonal promotions, product launches, and inventory availability affect MoM growth.
- Revenue concentration in top quartile shows **loyal high-value customers**.

6.3 Prescriptive Analysis

1. Promote top-performing products regionally.
2. Reward top quartile customers with loyalty programs.
3. Adjust inventory based on cumulative sales trends.
4. Investigate regions/months with declining MoM growth.

5. Upsell or engage lower quartile customers to boost revenue.

Step 7: References

Tutorials, docs and websites:

1. Mode Analytics(2024). Guide to SQL Window Functions.
   Available at: https://mode.com/sql-tutorial/sql-window-functions
2. Alex The Analyst. (2021). SQL Tutorial for Beginners - Full Course.
   YouTube. Available at: https://youtu.be/7S_tz1z_5bA?si=8x8O32gcNFQ0A6bk
3. SQL Window Functions partition by, order by, row_number, rank(), dense_rank
   Available at: https://youtu.be/rIcB4zMYMas?si=FzXtcfFUgM3iEYly
4. MySQL Official Documentation -Window Functions.
   Available at: https://dev.mysql.com/doc/refman/8.0/en/window-functions.html
5. GeeksforGeeks – Window Functions in SQL.
   Available at: https://www.geeksforgeeks.org/sql/window-functions-in-sql/
6. FreeCodeCamp – SQL Window Functions Guide
   Available at: https://www.freecodecamp.org/news/window-functions-in-sql/
   Business resources:
7. Mahembe Coffee: A Cherry Producer, Green Bean Processor, & Trader in Rwanda
   Available at: https://www.iisd.org/system/files/2024-08/responsible-agricultural-investment-rwanda-mahembe-coffee.pdf
8. Coffee Value Chain in Ethiopia Available at:
   https://www.sciencedirect.com/science/article/pii/S2666719325002043