PL/SOL ASSIGNMENT

Step 1: Problem Definition

Business Context

A large retail company in the retail industry aims to enhance customer experience by expanding its e-commerce platform while updating physical store layouts. The Sales and Operations departments are responsible for monitoring sales performance, customer engagement, and resource allocation.

Data Challenge

The company has sales, customer, and operational data across multiple regions and stores. Management lacks insights into which products drive online and in-store revenue, how customers' buying behavior varies between channels, and how resource allocation impacts overall sales and customer satisfaction.

Expected Outcome

Identify the top-performing products by sales channel and region, analyze customer purchasing patterns, and recommend resource allocation strategies to optimize both online growth and physical store improvements.

Step 2: Success Criteria

The following are five specific, measurable goals for analyzing the retailer's sales and customer behavior:

1. Identify Top 5 Products per Region

- Objective: Determine the highest-selling products in each region to guide marketing and inventory decisions.

- Window Function: RANK()
- Measurable Outcome: List of top 5 products by revenue per region.

2. Calculate Running Monthly Sales Totals

- Objective: Track how total revenue accumulates month by month to monitor sales performance.
- Window Function: SUM() OVER(PARTITION BY region ORDER BY month)
- Measurable Outcome: Cumulative monthly sales per region, showing growth trends.

3.Measure Month-over-Month Growth

- Objective: Compare sales performance across consecutive months to evaluate growth or decline.
- Window Function: LAG() or LEAD()
- Measurable Outcome: Monthly revenue differences and percentage growth for each region or product.

4. Segment Customers into Quartiles by Spending

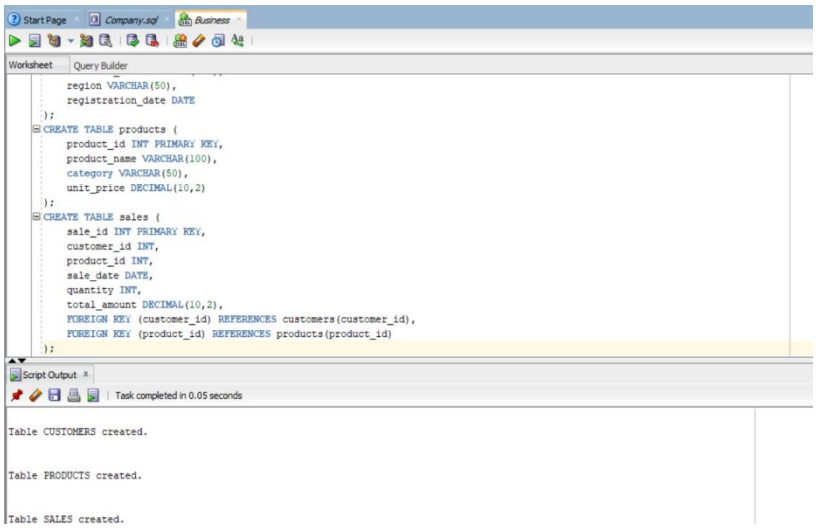
- Objective: Group customers into four segments (high, medium, low spenders) to target promotions effectively.
- Window Function: NTILE(4)

- Measurable Outcome: Each customer assigned to a quartile based on total spending.

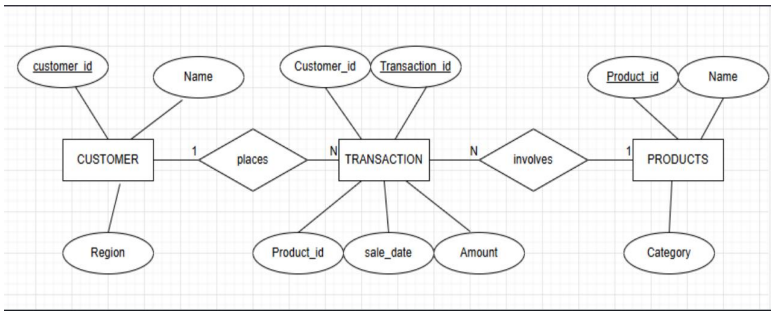
5. Compute Three-Month Moving Average of Sales

- Objective: Smooth out short-term fluctuations to better understand overall sales trends.
- Window Function: AVG() OVER(PARTITION BY region ORDER BY month ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)
- Measurable Outcome: Rolling average revenue per region for every three-month period

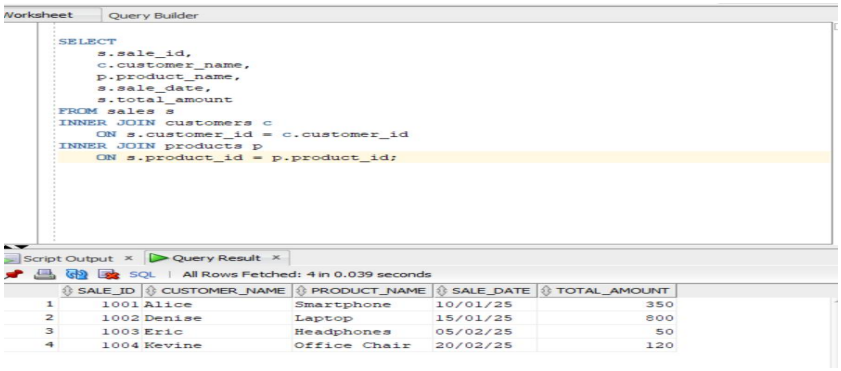
Creation of table



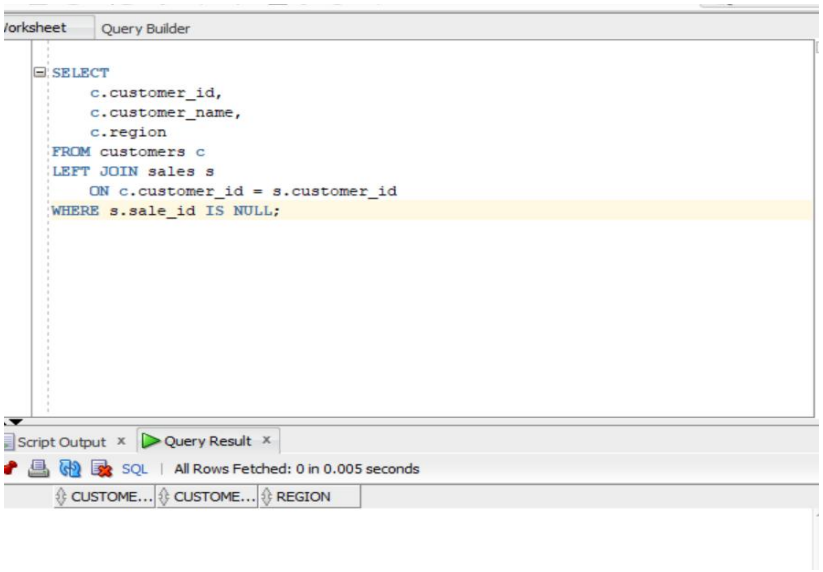
ER DIAGRAM



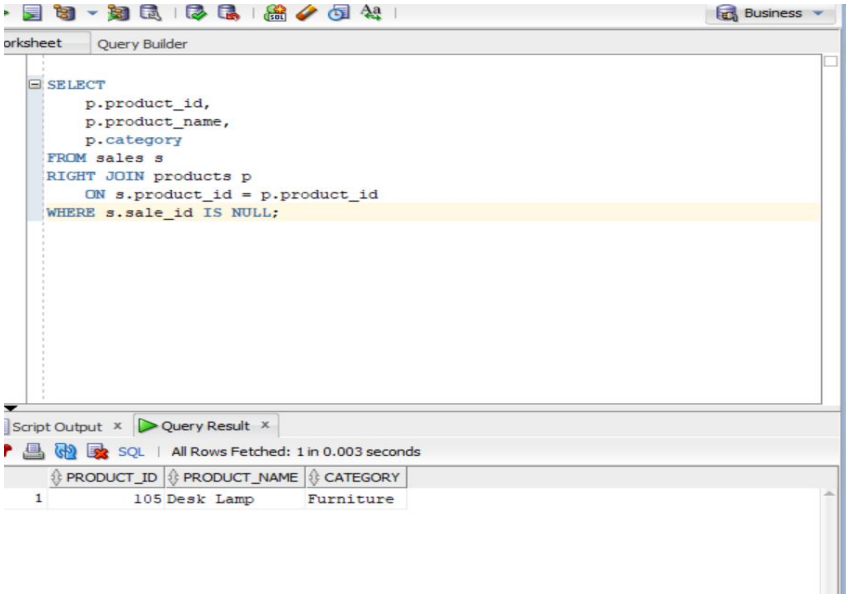
SQL joins implementation



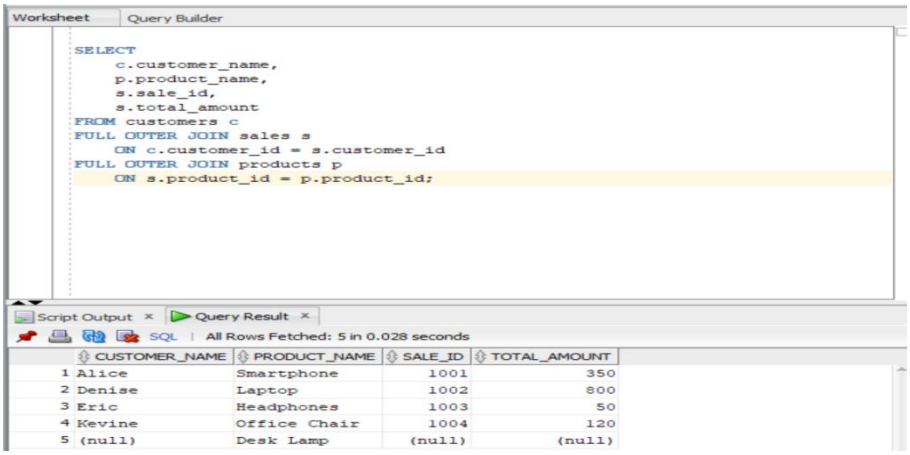
Left join



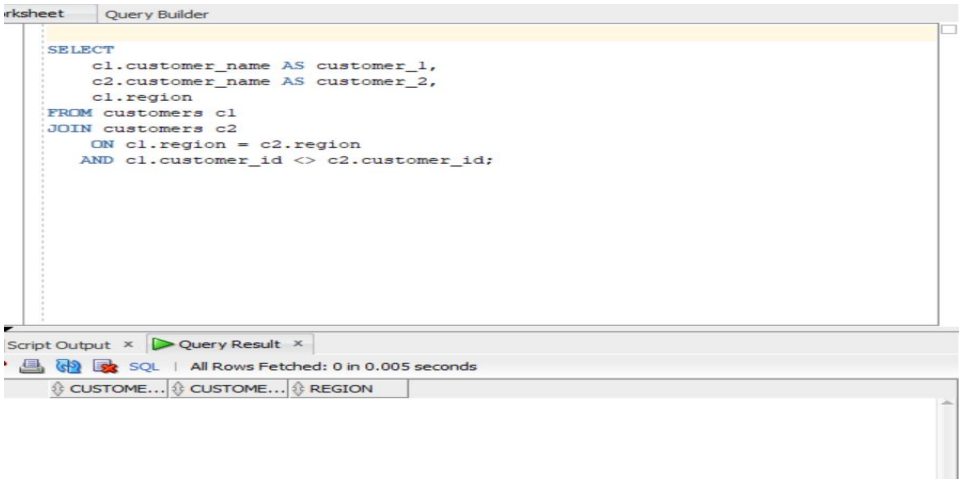
Right join



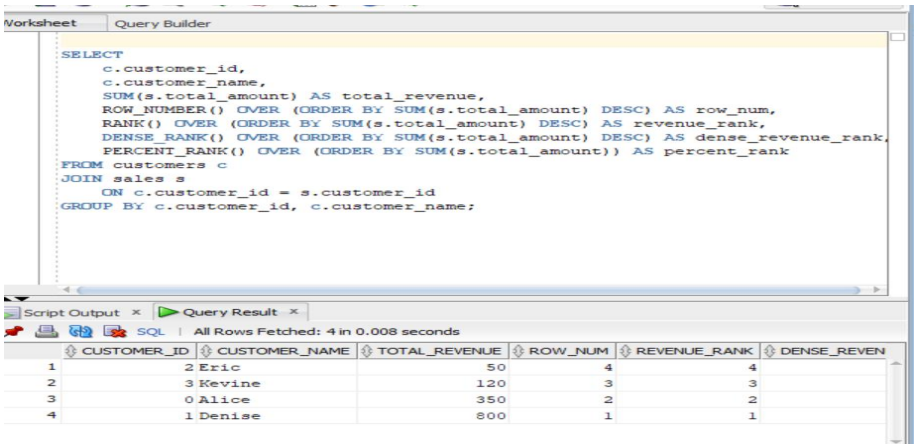
Full outer join



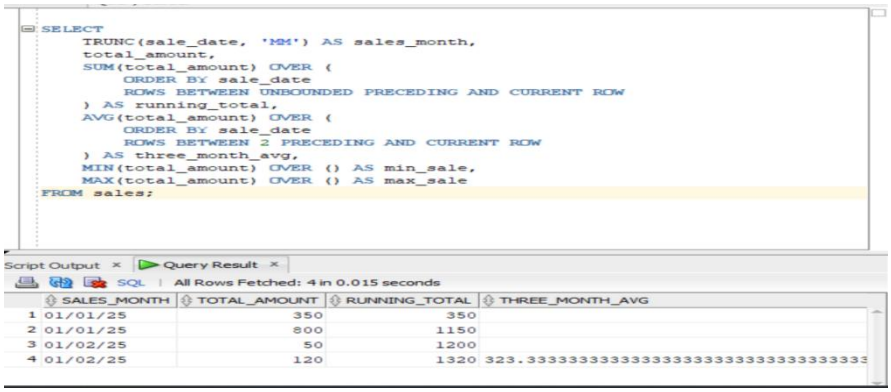
Self join



Ranking function



Aggregate window function

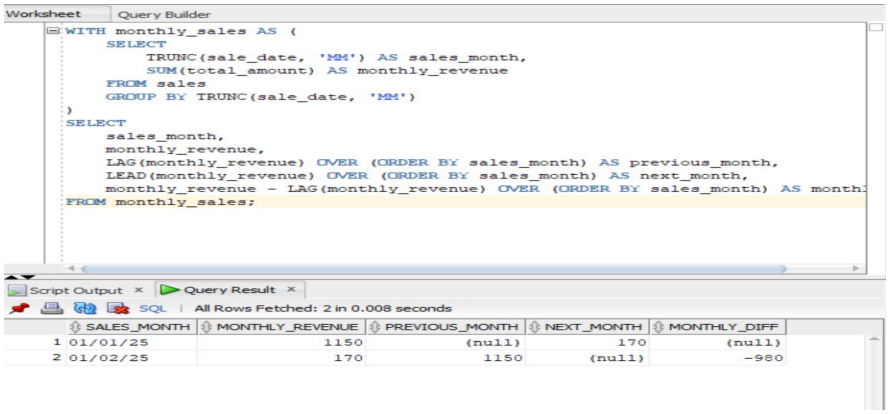


```
SELECT
    TRUNC(sale_date, 'MM') AS sales_month,
    total_amount,
    SUM(total_amount) OVER (
        ORDER BY sale_date
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS running_total,
    AVG(total_amount) OVER (
        ORDER BY sale_date
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
    ) AS three_month_avg,
    MIN(total_amount) OVER () AS min_sale,
    MAX(total_amount) OVER () AS max_sale
FROM sales;
```

Script Output | Query Result | SQL | All Rows Fetched: 4 in 0.015 seconds

	SALES_MONTH	TOTAL_AMOUNT	RUNNING_TOTAL	THREE_MONTH_AVG
1	01/01/25	350	350	
2	01/01/25	800	1150	
3	01/02/25	50	1200	
4	01/02/25	120	1320	323.33333333333333333333333333333333

Navigation function

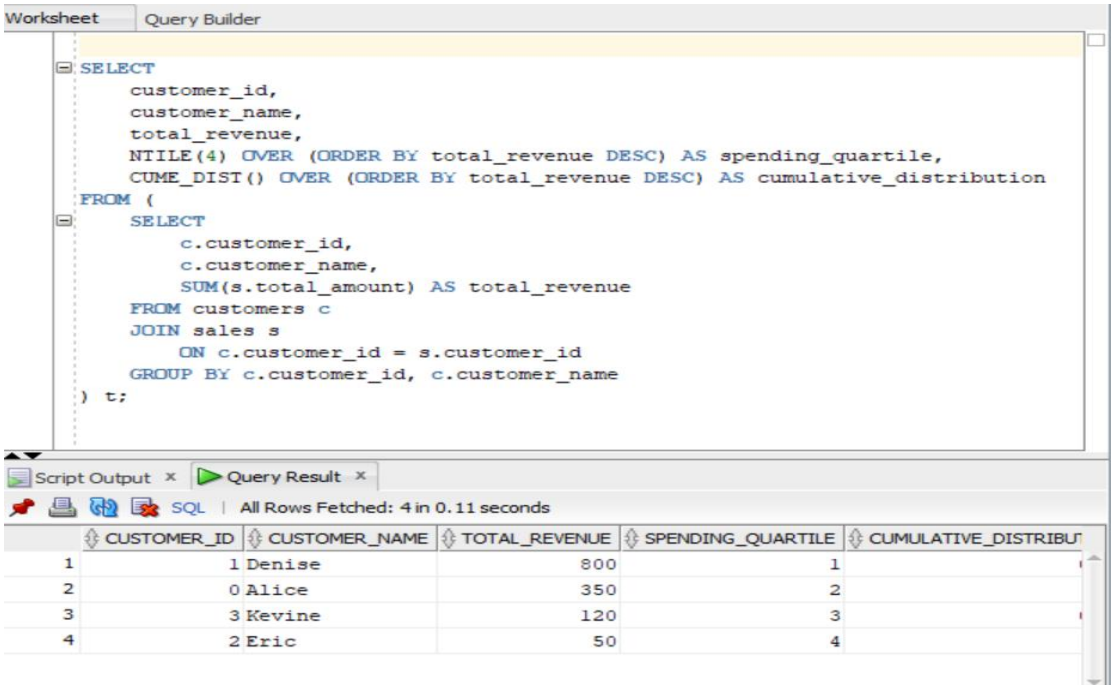


```
WITH monthly_sales AS (
    SELECT
        TRUNC(sale_date, 'MM') AS sales_month,
        SUM(total_amount) AS monthly_revenue
    FROM sales
    GROUP BY TRUNC(sale_date, 'MM')
)
SELECT
    sales_month,
    monthly_revenue,
    LAG(monthly_revenue) OVER (ORDER BY sales_month) AS previous_month,
    LEAD(monthly_revenue) OVER (ORDER BY sales_month) AS next_month,
    monthly_revenue - LAG(monthly_revenue) OVER (ORDER BY sales_month) AS month
FROM monthly_sales;
```

Script Output | Query Result | SQL | All Rows Fetched: 2 in 0.008 seconds

	SALES_MONTH	MONTHLY_REVENUE	PREVIOUS_MONTH	NEXT_MONTH	MONTHLY_DIFF
1	01/01/25	1150	(null)	170	(null)
2	01/02/25	170	1150	(null)	-980

Distribution function



```
SELECT
    customer_id,
    customer_name,
    total_revenue,
    NTILE(4) OVER (ORDER BY total_revenue DESC) AS spending_quartile,
    CUME_DIST() OVER (ORDER BY total_revenue DESC) AS cumulative_distribution
FROM (
    SELECT
        c.customer_id,
        c.customer_name,
        SUM(s.total_amount) AS total_revenue
    FROM customers c
    JOIN sales s
        ON c.customer_id = s.customer_id
    GROUP BY c.customer_id, c.customer_name
) t;
```

Script Output | Query Result | SQL | All Rows Fetched: 4 in 0.11 seconds

	CUSTOMER_ID	CUSTOMER_NAME	TOTAL_REVENUE	SPENDING_QUARTILE	CUMULATIVE_DISTRIBU
1	1	Denise	800	1	
2	0	Alice	350	2	
3	3	Kevine	120	3	
4	2	Eric	50	4	

Results analysis

1. Description analysis

The analysis shows that a small group of products and customers generate a significant portion of total revenue across regions. Sales increased steadily over time, with noticeable month-to-month variations and seasonal spikes. Customer segmentation reveals clear differences in spending behavior, with top quartile customers contributing the highest share of sales.

2. Diagnostic analysis

High-performing products benefited from strong customer demand and effective placement in both online and physical channels. Revenue growth aligns with increased digital adoption and targeted marketing efforts, while slower periods correspond to delayed store upgrades or lower promotional activity. Differences in customer spending are influenced by purchasing frequency, customer loyalty, and regional preferences.

3. Prescriptive analysis

The company should prioritize high-revenue products and top-spending customer segments through personalized promotions and loyalty programs. Underperforming products and inactive customers should be targeted with discounts or reconsidered for inventory optimization. A balanced investment strategy across digital platforms, supply chain efficiency, and physical store improvements is recommended to sustain long-term growth.

REFERENCES

1. Oracle Corporation. *Oracle Database SQL Language Reference.* https://docs.oracle.com/en/database/

2. PostgreSQL Global Development Group. *PostgreSQL Window Functions Documentation.* https://www.postgresql.org/docs/current/tutorial-window.html

3. MySQL. *MySQL 8.0 Reference Manual – Window Functions*. https://dev.mysql.com/doc/refman/8.0/en/window-functions.html

4. Microsoft. *SQL Server Documentation – Analytical Functions*. https://learn.microsoft.com/en-us/sql/

5. W3Schools. *SQL JOIN and Window Functions Tutorial*. https://www.w3schools.com/sql/

6. Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit* (3rd ed.). Wiley.