**Instructions:**

**Load All Data in S3 Bucket by creating respective folder and try loading using copy command. Do the same in Matillion via S3 load and file/fixed iterator so that both you have the hold.**

**1. Customer and Sales Performance Analysis**

* **Business Scenario**: You are a business analyst for a retail company. You have multiple datasets: one for customer details, one for product sales, and one for customer transactions. You need to analyze customer behavior and sales performance over time.
* **Window Functions**: **ROW\_NUMBER(), RANK(), DENSE\_RANK(), LAG(), LEAD()**

**Assignment Problem:**

* **Task 1**: Rank customers based on their total purchase amount per year. Identify the top 5 customers each year and track their ranking over time. (Use RANK())
* **Task 2**: For each customer, get their previous transaction amount and compare it with their current one. (Use LAG())
* **Task 3**: Calculate the running total of sales for each customer in the last 3 years. (Use SUM() OVER)
* **Task 4**: Divide customers into quartiles based on their total sales per year and analyze their spending patterns. (Use NTILE())

**Tables:**

* customers (Customer Information)
* sales (Product Sales Information)
* transactions (Customer Transactions Data)

**DDL for Tables:**

**-- Customers Table**

**CREATE TABLE customers (**

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255),

region VARCHAR(50),

join\_date DATE

**);**

**-- Sales Table**

**CREATE TABLE sales (**

sale\_id INT PRIMARY KEY,

product\_id VARCHAR(50),

sale\_amount DECIMAL(10, 2),

sale\_date DATE,

customer\_id INT REFERENCES customers(customer\_id)

**);**

**-- Transactions Table**

**CREATE TABLE transactions (**

transaction\_id INT PRIMARY KEY,

customer\_id INT REFERENCES customers(customer\_id),

transaction\_date DATE,

transaction\_amount DECIMAL(10, 2)

**);**

**2. Employee Performance Evaluation**

* Business Scenario: You are responsible for evaluating the performance of employees over time for a multinational company. You have datasets for employee details, their monthly evaluations, and their department transfers. You need to track their performance and career progression.
* Window Functions: **ROW\_NUMBER(), DENSE\_RANK(), FIRST\_VALUE(), LAST\_VALUE()**

**Assignment Problem:**

* **Task 1:** Track the rank of employees in terms of monthly evaluation scores within each department for the last 3 years. (Use DENSE\_RANK())
* **Task 2:** Identify employees whose performance improved consistently over the last 6 months by comparing their current evaluation with their previous ones. (Use LAG())
* **Task 3:** Find the first department each employee joined and their current department. (Use FIRST\_VALUE(), LAST\_VALUE())
* **Task 4:** Calculate the cumulative average evaluation score of each employee over the years. (Use AVG() OVER)

**Tables:**

* employees (Employee Information)
* evaluations (Monthly Employee Evaluation Scores)
* departments (Department Transfer History)

**DDL For Tables:**

**-- Employees Table**

CREATE TABLE **employees** (

employee\_id INT PRIMARY KEY,

employee\_name VARCHAR(255),

hire\_date DATE,

current\_department VARCHAR(100)

);

**-- Evaluations Table**

CREATE TABLE **evaluations** (

evaluation\_id INT PRIMARY KEY,

employee\_id INT REFERENCES employees(employee\_id),

evaluation\_month DATE,

evaluation\_score DECIMAL(5, 2)

);

**-- Departments Table**

CREATE TABLE **departments** (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100),

employee\_id INT REFERENCES employees(employee\_id),

transfer\_date DATE

);

**3. Product Pricing and Sales Trends**

* Business Scenario: You are tasked with analyzing product pricing trends and their impact on sales. You have datasets for product prices, sales transactions, and promotions. The company wants to understand how price changes and promotions affect product sales.
* Window Functions: LEAD(), LAG(), SUM() OVER, MAX() OVER

**Assignment Problem:**

* **Task 1:** For each product, compare the current price with the previous price and the next price, and analyze how price changes affect sales. (Use LAG(), LEAD())
* **Task 2:** Calculate the cumulative sales for each product per month over the last 2 years. (Use SUM() OVER)
* **Task 3:** Find the maximum price each product has ever been sold at and compare it to the current price. (Use MAX() OVER)
* **Task 4:** Identify products that had promotional discounts and track their sales before, during, and after the promotion. (Use LAG(), LEAD())

**Tables:**

* products (Product Information)
* prices (Product Pricing History)
* promotions (Promotional Events)
* sales (Sales Transactions Data)

**DDL For Tables:**

**-- Products Table**

CREATE TABLE products (

product\_id VARCHAR(50) PRIMARY KEY,

product\_name VARCHAR(255),

category VARCHAR(100)

);

**-- Prices Table**

CREATE TABLE prices (

price\_id INT PRIMARY KEY,

product\_id VARCHAR(50) REFERENCES products(product\_id),

price DECIMAL(10, 2),

price\_date DATE

);

**-- Promotions Table**

CREATE TABLE promotions (

promotion\_id INT PRIMARY KEY,

product\_id VARCHAR(50) REFERENCES products(product\_id),

discount DECIMAL(5, 2),

start\_date DATE,

end\_date DATE

);

**-- Sales Table**

CREATE TABLE sales (

sale\_id INT PRIMARY KEY,

product\_id VARCHAR(50) REFERENCES products(product\_id),

sale\_date DATE,

quantity\_sold INT,

total\_amount DECIMAL(10, 2)

);

**4. Financial Transaction Monitoring**

* **Business Scenario**: You are working for a bank, and you need to monitor large-scale transactions for suspicious activities. You have datasets for customer details, account transactions, and suspicious activities flagged by the system. The goal is to track financial behavior over time and detect irregularities.
* **Window Functions**: SUM() OVER, NTILE(), FIRST\_VALUE(), LAST\_VALUE()

**Assignment Problem:**

* **Task 1**: Calculate the cumulative sum of transactions for each customer in the last 2 years and flag customers whose transactions exceed a threshold. (Use SUM() OVER)
* **Task 2**: Divide customers into deciles (10 groups) based on their total transactions and analyze which decile is most prone to suspicious activity. (Use NTILE())
* **Task 3**: Track the first and last large transaction for each customer in the past 4 years. (Use FIRST\_VALUE(), LAST\_VALUE())
* **Task 4**: Compare each customer’s current transaction with their previous one to detect sudden increases in transaction amounts. (Use LAG())

**Tables:**

* customers (Bank Customer Information)
* accounts (Bank Accounts)
* transactions (Transaction Data)
* suspicious\_activities (Flagged Suspicious Activities)

**DDL for Tables:**

**-- Customers Table**

CREATE TABLE customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255),

region VARCHAR(100),

join\_date DATE

);

**-- Accounts Table**

CREATE TABLE accounts (

account\_id INT PRIMARY KEY,

customer\_id INT REFERENCES customers(customer\_id),

account\_type VARCHAR(100),

balance DECIMAL(15, 2),

opened\_date DATE

);

**-- Transactions Table**

CREATE TABLE transactions (

transaction\_id INT PRIMARY KEY,

account\_id INT REFERENCES accounts(account\_id),

transaction\_date DATE,

transaction\_amount DECIMAL(15, 2),

transaction\_type VARCHAR(50)

);

**-- Suspicious Activities Table**

CREATE TABLE suspicious\_activities (

activity\_id INT PRIMARY KEY,

transaction\_id INT REFERENCES transactions(transaction\_id),

flagged\_reason VARCHAR(255),

flagged\_date DATE

);

**Datasets:**

**     **

**Additional Complex Problems :**

**These problems require deeper analytical thinking and the creative use of window functions to solve business challenges.**

**1. Customer Purchase Patterns and Anomalies**

* **Dataset**: customers, transactions, sales, products
* **Window Functions**: LAG(), LEAD(), SUM() OVER, RANK(), ROW\_NUMBER(), NTILE(), COUNT() OVER, PERCENT\_RANK()

**Problem Statements:**

* **Task 1**: Detect anomalies in customer purchasing behavior by identifying customers whose current transaction amount is 3 times larger than their previous one. (Use LAG() to compare current and previous transaction amounts, and flag outliers.)
* **Task 2**: For each customer, calculate their rolling average transaction amount for the last 3 transactions and identify customers whose latest transaction is significantly above their rolling average. (Use LAG() and SUM() OVER for a rolling calculation.)
* **Task 3**: Rank products based on total sales in the last year and divide them into quartiles. Analyze the spending behavior of customers who bought products in the top quartile versus the bottom quartile. (Use RANK() and NTILE() to divide products into groups.)
* **Task 4**: For each product, calculate the cumulative sales per month over the past year and identify months where sales dropped by more than 50% compared to the previous month. (Use SUM() OVER and LAG() to analyze sales patterns month over month.)
* **Task 5**: Identify the top 3 customers based on total sales per month and check if the same customers stay in the top 3 across consecutive months. Track any new customers entering the top 3 and compare their monthly growth rates. (Use RANK() and LAG().)

**2. Employee Promotion and Performance Trends**

* **Dataset**: employees, departments, evaluations
* **Window Functions**: FIRST\_VALUE(), LAST\_VALUE(), RANK(), DENSE\_RANK(), NTILE(), LEAD(), LAG(), PERCENT\_RANK()

**Problem Statements:**

* **Task 1**: For each employee, track their career progression by identifying the first department they joined and the current department they belong to. Analyze the average time spent by employees in each department before being transferred. (Use FIRST\_VALUE(), LAST\_VALUE().)
* **Task 2**: Identify employees whose evaluation scores consistently increased or decreased over the last 6 months. Calculate their performance growth rate over this period. (Use LAG() to track changes in evaluation scores.)
* **Task 3**: Divide employees into deciles based on their evaluation scores for the past year. Identify if any department has a disproportionate number of employees in the top decile and analyze department-level performance. (Use NTILE() to group employees into deciles.)
* **Task 4**: For employees who were promoted within the past 2 years, compare their evaluation scores 6 months before and after their promotion to check if the promotion had a positive impact on performance. (Use LEAD(), LAG() to compare pre- and post-promotion performance.)
* **Task 5**: Identify employees who entered the top 10% of performers (based on evaluation scores) for the first time within the past year. Track how long they stayed in the top 10% and analyze what factors contributed to their rise. (Use PERCENT\_RANK().)

**3. Product Pricing and Promotion Optimization**

* **Dataset**: products, prices, promotions, sales
* **Window Functions**: LAG(), LEAD(), NTILE(), ROW\_NUMBER(), AVG() OVER, MAX() OVER, MIN() OVER

**Problem Statements:**

* **Task 1**: Identify products whose prices increased by more than 20% within the last 6 months and check if the price increase affected the sales volume. (Use LAG() to compare current and previous prices, and SUM() OVER to track sales volume.)
* **Task 2**: For each product, calculate the rolling average price over the last 3 price updates and determine whether promotional discounts during this time led to a significant increase in sales. (Use LAG() and AVG() OVER to compute the rolling average.)
* **Task 3**: Rank promotions based on their effectiveness (total sales generated during the promotion) and identify which products consistently benefited the most from promotions. (Use RANK() and ROW\_NUMBER() to analyze promotions.)
* **Task 4**: Identify products that experienced the highest price volatility (measured by the difference between the maximum and minimum price over the last year) and analyze how this affected their overall sales. (Use MAX() OVER, MIN() OVER.)
* **Task 5**: For each product, divide price updates into quartiles based on the price level and analyze whether products in the top quartile of prices experience more frequent promotions. (Use NTILE() to divide price levels and LEAD() to analyze promotions.)

**4. Bank Account Monitoring and Fraud Detection**

* **Dataset**: customers, accounts, transactions, suspicious\_activities
* **Window Functions**: SUM() OVER, NTILE(), LAG(), LEAD(), FIRST\_VALUE(), ROW\_NUMBER()

**Problem Statements:**

* **Task 1**: Calculate the running total of transaction amounts for each customer over the past year and identify customers whose transaction volume suddenly spikes by more than 50% compared to their historical average. (Use SUM() OVER and LAG() to track changes.)
* **Task 2**: Divide customers into quartiles based on their total transaction amounts over the past 3 months. Analyze which quartile has the most flagged suspicious activities and investigate patterns. (Use NTILE() to divide customers.)
* **Task 3**: Identify the first flagged suspicious activity for each customer and track whether they had any similar flagged activities in the following 6 months. (Use FIRST\_VALUE() and LEAD().)
* **Task 4**: For each account, track the last 3 transactions and check if they were made in locations different from the customer’s usual location (based on flagged suspicious activities). (Use LAG() and LEAD() to compare transaction locations.)
* **Task 5**: Identify customers who open multiple accounts and transfer large amounts of money between their own accounts. Track these internal transfers over time and identify any unusual patterns. (Use ROW\_NUMBER() to identify and track internal transfers.)

**5. Cross-Department Analysis of Employee Transfers and Performance**

* **Dataset**: employees, departments, evaluations
* **Window Functions**: RANK(), PERCENT\_RANK(), SUM() OVER, NTILE(), LAG()

**Problem Statements:**

* **Task 1**: Rank employees by their cumulative performance scores within each department and identify employees who are consistently ranked in the top 5% of performers in their department. (Use RANK() and PERCENT\_RANK().)
* **Task 2**: For employees who have transferred departments within the past 2 years, calculate the cumulative average evaluation score for both departments and compare whether their performance improved after the transfer. (Use SUM() OVER and LAG().)
* **Task 3**: Divide employees into quartiles based on their evaluation scores over the past year and identify departments where employees in the bottom quartile are frequently transferred out. (Use NTILE().)
* **Task 4**: Identify employees who received the highest performance score increase after a department transfer and analyze whether there is a correlation between department transfers and performance improvements. (Use LAG() and RANK().)
* **Task 5**: Track the department transfers of employees over time and rank departments based on the number of high-performing employees they gain or lose after transfers. (Use RANK() and PERCENT\_RANK() to track department movement.)

**6. Fraud Detection Based on Unusual Account Behavior**

* **Dataset**: accounts, transactions, suspicious\_activities
* **Window Functions**: LAG(), LEAD(), NTILE(), FIRST\_VALUE(), LAST\_VALUE(), SUM() OVER

**Problem Statements:**

* **Task 1**: For each account, track the last 5 transactions and check if the current transaction is significantly larger than the previous ones. Flag accounts where this pattern occurs. (Use LAG() to compare transaction amounts.)
* **Task 2**: Identify accounts with unusually high transaction amounts by dividing accounts into deciles based on total transaction volume in the past year. Focus on accounts in the top decile for further investigation. (Use NTILE().)
* **Task 3**: Track accounts with multiple flagged suspicious activities and analyze whether the flagged activities follow a similar pattern in terms of transaction amount or location. (Use LEAD() and LAG() to compare flagged activities.)
* **Task 4**: Identify the first and last suspicious activity for each account and analyze whether the account's balance or transaction volume changed significantly during this period. (Use FIRST\_VALUE(), LAST\_VALUE().)
* **Task 5**: For each account, calculate the cumulative transaction volume over the past year and flag accounts that suddenly exhibit a large spike in cumulative volume, possibly indicating fraudulent behavior. (Use SUM() OVER and LAG().)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LETS RIGG YOU MORE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**RESTAURANT SALES DATA ANALYSIS**

**Datasets:  **

Here is the **DDL (Data Definition Language)** to create the tables for the restaurant datasets:

**1. restaurant\_customers Table**

CREATE TABLE restaurant\_customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(100) NOT NULL

);

**2. restaurant\_transactions Table**

This table stores details of each transaction made by customers over the last 3 years.

CREATE TABLE restaurant\_transactions (

transaction\_id INT PRIMARY KEY,

customer\_id INT,

transaction\_date DATE,

transaction\_amount DECIMAL(10, 2),

items\_ordered INT,

payment\_method VARCHAR(50),

branch VARCHAR(50),

customer\_feedback INT CHECK (customer\_feedback BETWEEN 1 AND 5),

FOREIGN KEY (customer\_id) REFERENCES restaurant\_customers(customer\_id)

);

**Key Columns:**

* **customer\_id**: Links transactions to specific customers.
* **transaction\_amount**: Tracks how much was spent in each transaction.
* **items\_ordered**: Number of items ordered in a particular transaction.
* **payment\_method**: Records the payment type (Cash, Credit Card, Digital).
* **branch**: Indicates which branch the customer visited.
* **customer\_feedback**: Stores customer feedback rating (1 to 5 scale).

These tables are designed to maintain relational integrity between customers and their transactions. Let me know if you need any additional queries or data modifications!

Here are some **complex business scenario questions** using **window functions** based on the restaurant dataset. These exercises involve various window functions like RANK(), ROW\_NUMBER(), LAG(), LEAD(), NTILE(), and others to solve real-world analytical problems.

These questions will encourage creative problem-solving using window functions, allowing you to gain insights into customer behavior, branch performance, customer loyalty, seasonal trends, and the effectiveness of promotions. Let me know if you need further elaboration on any of the tasks!

**1. Identifying High-Spending Customers**

* **Goal**: Find out which customers are consistently the highest spenders.
* **Window Functions**: SUM() OVER, RANK(), ROW\_NUMBER()

**Problem Statements:**

* **Task 1**: For each branch, rank customers based on their total spending over the last 2 years. Identify the top 5 customers from each branch.
* **Task 2**: Calculate the running total of transaction amounts for each customer and branch. Track how much each customer spent cumulatively and identify customers whose spending reached $1,000 first.
* **Task 3**: For each branch, assign a rank to customers based on their spending in the last 6 months. If two customers spent the same amount, give them the same rank (use RANK()), but ensure no gaps in ranking (use DENSE\_RANK()).

**2. Customer Loyalty Analysis**

* **Goal**: Identify customer loyalty trends and track how their spending behavior changes over time.
* **Window Functions**: LAG(), LEAD(), NTILE()

**Problem Statements:**

* **Task 1**: For each customer, calculate the change in their transaction amount compared to their previous transaction. Identify customers whose spending consistently increased for their last 3 transactions.
* **Task 2**: Divide all customers into quartiles based on their total spending in the last year. Analyze whether customers in the top quartile have more positive feedback compared to the bottom quartile.
* **Task 3**: Use LAG() to track whether customers’ spending drops sharply after receiving poor feedback. Identify customers whose spending fell by more than 50% after giving a rating of 2 or lower.

**3. Branch Performance Analysis**

* **Goal**: Analyze how each restaurant branch is performing in terms of customer satisfaction and revenue.
* **Window Functions**: AVG() OVER, MAX() OVER, MIN() OVER

**Problem Statements:**

* **Task 1**: For each branch, calculate the average transaction amount and compare the highest and lowest transaction amounts within each branch over the last 3 years.
* **Task 2**: Track monthly performance for each branch. Use window functions to calculate the rolling average transaction amount for each branch and identify months where the rolling average significantly drops.
* **Task 3**: Rank branches based on their total revenue and average customer feedback. Identify the branch with the highest revenue but lowest feedback score.

**4. Frequent Visitors and Rewards Program**

* **Goal**: Identify the most frequent customers and track their visits for loyalty program purposes.
* **Window Functions**: COUNT() OVER, ROW\_NUMBER(), NTILE()

**Problem Statements:**

* **Task 1**: Count how many times each customer visited the restaurant in the last 3 years. Identify the top 10% of customers who visit most frequently.
* **Task 2**: For each branch, assign a row number to each customer's visit and calculate how many customers visited more than once per month in the last year.
* **Task 3**: Divide customers into deciles based on their number of visits. Compare spending patterns of customers in the top decile vs. the bottom decile.

**5. Seasonal Trends and Peak Hours**

* **Goal**: Analyze customer behavior to identify seasonal trends and peak hours for business optimization.
* **Window Functions**: SUM() OVER, PERCENT\_RANK(), NTILE()

**Problem Statements:**

* **Task 1**: Identify peak months for customer spending by calculating the monthly total revenue for each branch. Compare the cumulative spending per month to the total yearly revenue and track which months contribute the most.
* **Task 2**: For each day of the week, calculate the average transaction amount during each hour of operation. Rank the hours based on transaction volume and identify the peak hours of business.
* **Task 3**: Divide days into quartiles based on transaction volume and track whether days in the top quartile also have higher average customer feedback scores.

**6. Customer Retention and Churn**

* **Goal**: Analyze customer retention patterns and identify signs of churn.
* **Window Functions**: LAG(), LEAD(), FIRST\_VALUE()

**Problem Statements:**

* **Task 1**: Track the time gap between consecutive visits for each customer. Identify customers whose last visit was more than 6 months ago as potential churn candidates.
* **Task 2**: For each customer, calculate the difference between the first transaction amount and the last transaction amount. Identify customers whose spending significantly decreased over time.
* **Task 3**: Use LAG() and LEAD() to track whether customers with low spending in their first visit increased or decreased their spending in subsequent visits.

**7. Discount Effectiveness and Feedback Analysis**

* **Goal**: Measure the effectiveness of discounts and analyze how discounts impact customer feedback.
* **Window Functions**: AVG() OVER, LAG(), NTILE()

**Problem Statements:**

* **Task 1**: For each branch, track transactions that received a discount and compare the average feedback score for transactions with and without discounts.
* **Task 2**: Calculate the average transaction amount before and after a discount was introduced for each branch. Analyze whether introducing discounts increased average spending.
* **Task 3**: Rank transactions by the amount of discount received and analyze whether customers who received larger discounts gave better feedback scores.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*hogaya aur chahye tha kya ? \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*