**Data Analyst Capstone Project**

Welcome to the SQL Capstone Project. This evaluation is designed to measure your ability to work with relational databases by writing SQL queries to solve business problems. The project will test your knowledge of data retrieval, aggregation, joins, subqueries, and data manipulation across multiple tables.

**Project Overview**

You will be provided with a database containing the following tables:

* **users\_customuser:** customer demographic and contact information
* **savings\_savingsaccount:** records of deposit transactions
* **plans\_plan:** records of plans created by customers
* **withdrawals\_withdrawal:** records of withdrawal transactions

**Evaluation Criteria**

Your solutions will be assessed on:

* **Accuracy:** Correctness of query results
* **Efficiency:** Optimal query structure and performance
* **Completeness:** Addressing all requirements of each question
* **Readability:** Proper formatting, comments and logical organization

This assessment is designed to evaluate both your technical SQL skills and your problem-solving methodology. Focus on delivering clear, accurate, and efficient solutions.

**Submission Requirements**

**Please follow these instructions carefully to submit your SQL Project:**

1. Create a new public repository named **DataAnalytics-Project**
2. Commit all your SQL files and [README.md](http://readme.md) to the repository based on the repository structure below. The SQL files must be named according to the format in the repository structure.
3. Submit the repository link.

**SQL File Requirements**

Each SQL file must:

* Contain a single SQL query that answers the question
* Be properly formatted with clear indentation
* Include comments for complex sections

**README.md Specifications**

Your README file should include:

* Per-Question Explanations: Explain your approach to each question.
* Challenges: Document any particular difficulties you encountered and how you resolved them.

**Repository Structure**

**DataAnalytics-Project/**

**│**

**├── Project\_Q1.sql**

**├── Project\_Q2.sql**

**├── Project\_Q3.sql**

**├── Project\_Q4.sql**

**│**

**└── README.md**

**Important Notes**

* Include only **SQL files** - **no database dumps** or other files

**QUESTIONS**

**1. High-Value Customers with Multiple Products**

**Scenario:** The business wants to identify customers who have both a savings and an investment plan (cross-selling opportunity).

**Task:** Write a query to find customers with at least one funded savings plan AND one funded investment plan, sorted by total deposits.

**Tables:**

* users\_customuser
* savings\_savingsaccount
* plans\_plan

**Expected Output:**

| owner\_id | name | savings\_count | investment\_count | total\_deposits |
| --- | --- | --- | --- | --- |
| 1001 | John Doe | 2 | 1 | 15000.00 |

**2. Transaction Frequency Analysis**

**Scenario:** The finance team wants to analyze how often customers transact to segment them (e.g., frequent vs. occasional users).

**Task:** Calculate the average number of transactions per customer per month and categorize them:

* "High Frequency" (≥10 transactions/month)
* "Medium Frequency" (3-9 transactions/month)
* "Low Frequency" (≤2 transactions/month)

**Tables:**

* users\_customuser
* savings\_savingsaccount

**Expected Output:**

| frequency\_category | customer\_count | avg\_transactions\_per\_month |
| --- | --- | --- |
| High Frequency | 250 | 15.2 |
| Medium Frequency | 1200 | 5.5 |

**3. Account Inactivity Alert**

**Scenario:** The ops team wants to flag accounts with no inflow transactions for over one year.

**Task:** Find all active accounts (savings or investments) with no transactions in the last 1 year (365 days) .

**Tables:**

* plans\_plan
* savings\_savingsaccount

**Expected Output:**

| plan\_id | owner\_id | type | last\_transaction\_date | inactivity\_days |
| --- | --- | --- | --- | --- |
| 1001 | 305 | Savings | 2023-08-10 | 92 |

**4. Customer Lifetime Value (CLV) Estimation**

**Scenario:** Marketing wants to estimate CLV based on account tenure and transaction volume (simplified model).

**Task:** For each customer, assuming the profit\_per\_transaction is 0.1% of the transaction value, calculate:

* Account tenure (months since signup)
* Total transactions
* Estimated CLV (Assume: CLV = (total\_transactions / tenure) \* 12 \* avg\_profit\_per\_transaction)
* Order by estimated CLV from highest to lowest

**Tables:**

* users\_customuser
* savings\_savingsaccount

**Expected Output:**

| customer\_id | name | tenure\_months | total\_transactions | estimated\_clv |
| --- | --- | --- | --- | --- |
| 1001 | John Doe | 24 | 120 | 600.00 |

**Hints:**

* owner\_id is a foreign key to the ID primary key in the users table
* plan\_id is a foreign key to the ID primary key in the plans table
* savings\_plan : is\_regular\_savings = 1
* investment\_plan: is\_a\_fund = 1
* confirmed\_amount is the field for value of inflow
* amount\_withdrawn is the field for value of withdrawal
* all amount fields are in kobo