**Data Engineer - Candidate Questions**

The following questions are designed to test the candidate’s knowledge and skills in relation to the position being filled.

**Q1.**

Problem, given the table definitions below write a single SQL query that fulfils the following business requirement.

**Source Tables**

You are given the following tables:

REVENUE\_ANALYSIS

ACTIVITY\_DATE DATE NOT NULL, Date wager was madeMEMBER\_ID INTEGER NOT NULL, Unique Player identifier  
GAME\_ID SMALLINT NOT NULL, Unique Game identifier  
WAGER\_AMOUNT REAL NOT NULL, Total amount wagered on the gameNUMBER\_OF\_WAGERS INTEGER NOT NULL, Number of wagers on the gameWIN\_AMOUNT REAL NOT NULL, Total amount won on the gameACTIVITY\_YEAR\_MONTH INTEGER NOT NULL, YYYYMMBANK\_TYPE\_ID SMALLINT DEFAULT 0 NOT NULL 0=Real money, 1=Bonus money

The “REVENUE\_ANALYSIS” table provides a summary of each member’s activity on a given date, listing the total amounts wagered and won for each game played that day. If a member does not play on a given day there will be no entries in the table. You can assume this is a very large table ~100M rows.

CALENDAR

CALENDAR\_DATE DATE NOT NULL, Base date (YYYY-MM-DD)  
CALENDAR\_YEAR INTEGER NOT NULL, 2010, 2011 etc  
CALENDAR\_MONTH\_NUMBER INTEGER NOT NULL, 1-12  
CALENDAR\_MONTH\_NAME VARCHAR(100), January, February etc  
CALENDAR\_DAY\_OF\_MONTH INTEGER NOT NULL, 1-31  
CALENDAR\_DAY\_OF\_WEEK INTEGER NOT NULL, 1-7  
CALENDAR\_DAY\_NAME VARCHAR(100), Monday, Tuesday etc  
CALENDAR\_YEAR\_MONTH INTEGER NOT NULL, 201011, 201012, 201101 etc

This CALENDAR table provides a base “date” dimension, one row per day from 2015 to 2020

The required solution is a view with the following columns:

MEMBER\_ID,  
CALENDAR\_YEAR\_MONTH,  
MEMBER\_LIFECYCLE\_STATUS,  
LAPSED\_MONTHS

Active means that the member has made a minimum of one **real money** wager in the month.

Each month a member has a certain lifecycle type. The member’s status will change on a monthly basis based on their previous and current month's activity. The statuses are:

**New** = First time they place a real money wager

**Retained** = Active in the prior calendar month and the current calendar month  
**Unretained** = Active in the prior calendar month, but not active in the current calendar month  
**Reactivated** = Not active in the prior calendar month, but active in the current calendar month  
**Lapsed** = Not active in the prior calendar month or the current calendar month

The view should display one row per member per month, starting from the month in which they first placed a real money wager. This view should give their lifecycle status for that month, and if the member has lapsed, it should show a rolling count of the number of months since they were last active.

We have provided the following files (DB\_Setup.zip) to help set up the environment:

* table\_setup.sql is generated from DB2 and contains the DDL of CALENDAR and REVENUE\_ANALYSIS tables
* Calendar\_Test\_Data.csv holds the CALENDAR data between 2015-2020
* Revenue\_Analysis\_Test\_Data.csv holds a minimal amount of example data for REVENUE\_ANALYSIS, please feel free to add more test data to REVENUE\_ANALYSIS to cover different scenarios.

Carmen: Please see the attached view.csv for the final view, logic.ipynb for the logic for one member and retention.ipynb the codes to produce view.csv for all members. I don’t have any SQL databases on my laptop, so I used Python for this exercise, but the logic should be the same.

1. For each member, aggregate the numbers of wagers by each month
2. Join the revenue and calendar tables, and expand the rows between the minimum and maximum month.
3. Calculate the lifecycle status with the given definitions.
4. If the costumer is lapsed in the given month, calculate the total months that the customer hasn’t played since the last active month.

**Q2.**

Given Q1 above list any questions you would want to ask about the requirements or source tables.

Carmen: The problem is well defined, and the information is well documented. However, in a real situation, I would ask for the names of the server, database and schema for the source tables sit under. As I would need to establish a SQL-Python connection to retrieve the data. I would also ask if we have any data that are outside the range between 2015 to 2020(as this is the range in the Calendar table), would those data be included in the exercise as well?

**Q3.**

Given Q1 above list the factors that may have an impact on the performance of the view along with suggestions as to how these factors may be mitigated.

Carmen: in the logic, I used loops frequently in the Python Jupyter notebook, this will be translated to Cursor in SQL when writing the view. Especially, I used nested loops, that will be very bad for the performance. Replace cursor with temporary tables, then join them back together.

Copyright

This document is produced by Gamesys Limited, a subsidiary of Bally’s Corporation (“Group”).

It contains Group confidential and proprietary information; under no circumstances should it be delivered or disclosed to any person not employed by a Group company without express written permission.