Assignment Background

As context for this assignment, you own an online forum for retro computing enthusiasts.

As the owner, you want to analyze forum data to explore the following topics and tasks:

- ✓ *User Growth Analysis* – To calculate user growth year-over-year.
- ✓ *Rewards & Recognition* – To identify and incentivize top contributors.
- ✓ *Community Activity* – To evaluate community engagement via the volume of posts and comments.
- ✓ *Question Resolution* – To assess content quality by counting posts with accepted answers.
- ✓ *Content Moderation* – To identify posts marked as spam or offensive to maintain standards.

Prepare the Environment

Create a new worksheet in Snowflake ("Final Project"):

- Write all your SQL statements in this file.
- During the project, include supporting comments.

Create a new Excel file:

- Store query results in this file, WHEN it's requested below.
- You do NOT need to store the results of EVERY query.
- When storing results, put them in separate worksheet tabs.

Set your role as "sysadmin". Then, confirm context in the upper part of the worksheet.

Create the Database

Create a new database called "forum".

Create the Warehouses

Create separate warehouses for loading and querying data:

- forum_loading_wh
- forum_query_wh

Use the following settings:

- Size – small
- Auto-Resume – enabled
- Auto-Suspend – 5 minutes (not 5 seconds)

Create a Resource Monitor

Switch to the "accountadmin" role. We will use this account to implement monitoring.

Create a new resource monitor ("forum_rm"):

- Include a monthly quota of 100 credits.
- Ensure the start timestamp is immediate.
- Notify admins at 80% usage.
- Suspend but continue existing queries at 95% usage.
- Suspend and cancel existing queries at 100% usage.

Apply the monitor to the "forum_query_wh" warehouse.

Set these query timeouts in "forum_query_wh":

- statement timeout: 20 minutes
- statement queued timeout: 10 minutes

You will need to convert these values to seconds.

Run a query to "show" all warehouses. (Store results in Excel.)

Create the Tables

Set the following context. Then, confirm context in the upper part of the worksheet.

- role – sysadmin
- database – forum.public
- warehouse – forum_loading_wh

Create the tables using these statements.

Prepare for Data Loading

Create a stage to access the data: *s3://retrocomputing-forum/*

Then, list the files and properties of the files in staging.

Create three separate file formats:

- forum_csv (comma-delimited)
- forum_pipe (pipe-delimited)
- forum_tab (tab-delimited)

Each file format should do the following:

- Skip the header row.
- Convert any blanks to SQL nulls.
- Note strings could be enclosed in double quotes.

Load the Data

Load these tables using their respective files:

- posts (comma-delimited)
- post_history (comma-delimited)
- post_links (comma-delimited)
- tags (tab-delimited)
- users (comma-delimited)
- votes (pipe-delimited)

Load the badges table. While doing so...

- Use the "badges.csv" file, which is comma-delimited.
- Reorder the columns, if necessary.
- Transform the badge classes using this list:
    - 1 = 'Gold'
    - 2 = 'Silver'
    - 3 = 'Bronze'
- Then, query the entire table, but filter for gold badges.
- (Store results in Excel.)

Load the comments file (JSON) into the "comments_json" table.

Then, create a view that structures this comments data, so it can be queried.

Transform Data (After Loading)

Create a field called "years_of_activity" in the users table.

Use the following instructions to create the field:

- Create a clone of the users table. Call it "users_dev".
- Add the column (an integer) to the development table.
- Populate the column using the calculation below.
- Move the development table to production.
- Then, delete the development table.

    TIMESTAMPDIFF(YEAR, creation_date, last_access_date)

Then, query the updated users table.

- Only select users with ≥ 7 years of activity.
- (Store the results in Excel.)

Create a New Role

Switch to the "useradmin" role. We will use this to create the new role.

Create a new role called "forum_query_role". Then, add this role to your account.

Switch to the "securityadmin" role. We will use this to grant privileges to the new role.

Grant the following privileges to this new role:

- Use and operate this warehouse: *forum_query_wh*
- Use this database and its schemas: *forum*
- Read from all tables and views in that database *

List all roles using the "show" function. (Store results in Excel.)

Switch to the new role and the "forum_query_wh" warehouse.

Use this context to complete the queries below.

Answer Business Questions

Write SQL statements to answer the questions below.

"What is the date range of forum posts?"

"How many users do we have?"

"Who are the top 5 users in terms of reputation?"

"Are we seeing a growth in the number of users year-over-year?" (Store results in Excel.)

"What % of users accessed the site recently?" (i.e. On or after 1/1/2023.)

"What gold badge was earned the most?" (Store results in Excel.)

"Which 10 users earned the most badges?" (Store results in Excel.)

"How many posts were created per year?" (Store results in Excel.)

"What % of posts have an accepted answer?"

- Posts with accepted answers have an "accepted_answer_id".
- Only consider posts with a "post_type_id" of 1 (i.e. questions).

"What % of posts received no answers?"

- Posts without answers have an "answer_count" of 0.
- Only consider posts with a "post_type_id" of 1 (i.e. questions).

"Which posts received the most updates?"

- You will need the "posts" and "post_history" tables.
- Only include posts with a non-null title.
- Only include the top 50 posts.
- (Store results in Excel.)

"Which users contributed the most comments?"

- Only include the top 10 users with the most comments.
- (Store results in Excel.)

"How many distinct posts received a vote of 'spam' or 'offensive'?"

- "spam" is a *vote_type_id* of 12
- "offensive" is a *vote_type_id* of 4

Submit Your Work

On the next page, you will submit your SQL statements and query results.