Scenario Description

You are assigned to work on a project from a new client. The client has a web application that they want to do reporting on. This requires us to import data from a single MySQL Database and also support ingesting one type of CSV file into Hadoop on a regular basis. They would like all the data to be available in Hive for easy querying from Hue and for preprocessing to be done to generate new tables which is used to generate reports for the client.

Data Sources

There is a single MySQL Database with 2 Tables:

* user
  + Description
    - Table with all the users in the system
    - Contains Inserts, Updates and Deletes of existing data
  + Schema
    - id - integer - primary key
      * Unique Identifier for the User
    - name - text
      * Users name
* activitylog
  + Description
    - A table with all the activity that users have been doing on the website
    - Data is only appended onto the table
      * The user\_id is periodically updated (set to null) in accordance with the deletes that are occurring on the user table, but these changes can be disregarded.
  + Schema
    - id - integer - primary key
      * Unique Identifier for the Activity event
    - user\_id - integer - foreign key to user.id
      * ID of the user that performed the activity
      * Note: may be null since the users will be deleted.
    - type - text
      * Type of activity that was done
      * Possible values:
        + UPDATE
        + DELETE
        + INSERT
    - timestamp - timestamp
      * Epoch time of when the operation was done

In addition to the above SQL database, there are periodic CSV dumps that are taken from another system:

* user\_upload\_dump
  + Description
    - When users upload files, information about who and what file is captured in another system. This data we can obtain through CSV Dumps.
    - Assume these CSV files are generated by the client at any time and placed at a location you specify.
      * CSV files can also be uploaded when you’re processing them
    - Also allow the client to upload multiple CSV files in case they want to ingest historical data
  + Schema
    - user\_id
      * ID of the user that performed the upload
    - file\_name
      * Name of the file that was uploaded
    - timestamp
      * Epoch time of when the file was uploaded

Data Sources Script

To replicate adding data to the MySQL database and creating new CSV files, a script has been provided to perform these actions: **practical\_exercise\_data\_generator.py.** This script can be ran from your local machine. Bring the script down to your local machine, and follow the instructions on the script to get it running.

Deliverables

1. A shell script which executes a job to import data from the MySQL Tables into Hive
2. A shell script to ingest CSV files from the local file system into HDFS and into Hive. usf
   1. Hint: Can be done by either running the load command or creating an external table
3. A shell script to execute a Hive query to generate the reporting tables
   1. See the below “Reporting Tables” section
4. All the scripts and some basic documentation should be pushed into a [GitHub](https://github.com/) Repository.

Reporting Tables

Here are the reporting tables that need to be generated after ingesting data

* user\_report
  + Description:
    - Aggregated usage and statistics on the site for each user
  + How to Generate:
    - Regenerate the table on every run
    - Join the user, activitylog and user\_upload\_dump tables in Hive to get the necessary data.
  + Fields:
    - user\_id
      * Unique Identifier for the user
    - total\_updates
      * Total number of updates that user has done
      * In the event there aren’t any updates, set it to 0
    - total\_inserts
      * Total number of inserts that user has done
      * In the event there aren’t any inserts, set it to 0
    - total\_deletes
      * Total number of deletes that user has done
      * In the event there aren’t any deletes, set it to 0
    - last\_activity\_type
      * The last type of activity the user has performed
      * In the event there isn’t any activities for the user, set it to NULL
    - is\_active
      * Has the user performed any activity within the last 2 days
      * In the event there is no activity, set to false
      * Output type should be boolean
    - upload\_count
      * Number of files the user has uploaded
      * In the event there aren’t any uploads, set it to 0
* user\_total
  + Description:
    - Total number of users using the website at certain points in time
  + How to Generate:
    - Append a new row into the table on each run
  + Fields:
    - time\_ran
      * Current time when the entry was added
    - total\_users
      * Total number of users in the system
    - users\_added
      * Number of users added since the last time the job was ran

FAQ

* NONE