Assignment 4

Due November 22, 2017 @ 11am

For this assignment we’re going to focus more on temporal data. Namely, we’ll be adding audit and history tables to our company database. In reality, you would want to have audit tables wherever you want to be able to keep track of important data, and these audit tables would probably be in a separate schema. For the purposes of this assignment we’ll only be adding a couple of these and you can include them in the same schema.

**Part 1 – Structural Updates: /10**

We want to **create two audit tables** based on our tables - employees and employee\_jobs (or whatever your equivalent is). For these audit tables, we want to keep track of all the columns in those tables, along with the audit date, and audit action (insert/update/delete), and the user who caused the action.

We also want to create an **employee history table**. This table will be used mainly for reporting, so will have a flatter format than most tables. We want to keep track of all pertinent information to an employee. This includes a variety of data:

* the employee’s personal information
  + all name fields
  + gender
  + ssn
  + birthdate
  + marital status
* other employee information
  + employee status
  + hire date,
  + rehire date
  + termination\_date
  + termination\_reason
  + termination\_type
* the employee’s job information
  + job code
  + job title
  + employee’s job start date
  + employee’s job end date
  + pay amount
  + standard hours
  + employee type
  + employment status
  + department code
  + department name
  + location code
  + location name
  + pay frequency
  + pay type
  + supervisor job

If any of this information regarding the employee changes, you will want to log a new employee history record with the updated data.

**Part 2 – Adding Triggers: /25**

In this section you will need to create triggers to populate your new audit/history tables. Assume that we have an application on top of our database that allows users to manipulate the data we store. When the user performs an action that affects one or more of the attributes that is contained in your history or audit tables, your trigger(s) should fire accordingly. Make sure that you consider all possible actions that can happen (insert, update, and delete). Since we don’t have an application to work with, for this assignment you can just use a few insert/update/delete statements to test how your triggers works**. Make sure you keep a script of any** **insert/update/delete statements so that the markers can verify the changes.**

**Part 3 – Updating Your Load Procedures: /25**

You need to update your load procedure take into account the history/audit tables as well. It’s possible that the client has made a change to the data on their end and you want to note the changes. You want to record the state of the information that the client passed to you. This will essentially create an ‘initial’ record of the employee information. This way, if they complain that your system messed up their data, you’ll have a record that that was the data they provided to you. Due to performance concerns, you don’t want your load procedure to cause your triggers to fire – see the notes for an example of this.

In order to make sure your load is working properly, there is a new employee load file supplied. There have been several data changes from the last employee file we loaded that need to be added/updated in our system. Running your load procedure should update your database and populate the new audit/history tables.

**Optional/Bonus marks: /5**

Populate your employee history table with all the current data before updating your database with the new employee file.

**Part 4 – Reports: /35**

**Question1: /25**

You need to send a report of the updated employee data back to the client every day, in the same format they provided. Write a report that provides the most up-to-date data in your database to create a file in the format of the ‘employeeFile’ file.

**Question 2: /10**

Write a report discussing any issues you encountered in this assignment.

* Discuss the changes you made to your database structure. Was there anything outside of the attributes specified in the assignment that you needed to add?
* Discuss what type of trigger you used (before, after, instead of, insert, update, delete) for your triggers and why.
* Did you run into any issues changing your data load code? What about when you loaded the new data file?
* Include an updated ER diagram from dbvisualizer for the marker.

**Part 5 – Style /5**

**What to hand in**

Part 1:

* an .sql file of the DDL changes that you made

Part 2:

* an .sql file (or files) with the audit triggers you wrote.
* An .sql file with any insert/updates/deletes you used for testing

Part 3:

* an .sql file (or files) with the updated load procedure

Part 4: Question 1

* an .sql file with the view definition used to create your report.
* A .csv file of the report queried from your database

Part 4: Question 2

* A .txt file with your report
* An updated ER diagram from dbvisualizer

**REMEMBER:** Your database needs to match the result of running your scripts in order for the markers to mark it. Significant marks will be deducted if your database doesn’t contain proper data.