CS 1555/2055 – Database Management Systems (Spring 2020)
Dept. of Computer Science, University of Pittsburgh

Assignment #6: Views and More SQL

Release: Feb. 21, 2020 Due: 8:00 PM, Feb. 27, 2020

Goal

In this homework, you will practice SQL, create views, and use them to answer your queries. You will also explore the difference between a (virtual) view and materialized view.

Description

- Assume the following relational database schema that supports a cell phone company, *P_Mobile*, that specializes in inexpensive wireless services to the University of Pittsburgh community of students, faculty, staff, and friends (*pn*, is short for phone number):
  - CUSTOMERS = (SSN, fname, lname, cell_pn, home_pn, street, city, zip, state, free_min, DOB, free_SMS)
  - RECORDS (from_pn, to_pn, start_timestamp, duration, type)
  - STATEMENTS (cell_pn, start_date, end_date, total_minutes, total_SMS, amount_due)
  - PAYMENTS (cell_pn, paid_on, amount_paid)
  - DIRECTORY (pn, fname, lname, street, city, zip, state)
- For this assignment, create tables from the sample solutions of Q2 and Q3 in assignment#2 (HW2), and populate the database with the data from the INSERT statements in the sample solutions of Q4 from HW2 (include all that in you 'db' script). Then, answer the following questions [for a total of 100 points]:

Questions

1. [40 points: 10 points each] Create the following views:
   (a) A views named 'PA_NOT_CUSTOMERS' that lists the full name, phone number, and the city of people who are not customers of *P_Mobile* and live in Pennsylvania state.
   (b) A materialized view named 'MV_PA_CUSTOMERS' that lists the full name, phone number, and the city of people who are customers of *P_Mobile* and live in Pennsylvania state.
   (c) A view named 'NUMBER_OF_RECEIVED_CALLS' that lists the phone number of customers along with the number of calls they have received.
   (d) A materialized view named 'MV_OUTSTANDING_BAL' that lists full name, cell phone, and balance (i.e., the difference between the amount due and amount paid in total) of all customers in a descending order based on the balance. (NULLs are not accepted in the result, if a customer has no balance (NULL balance), then the value should be 0).

2. [45 points] Write SQL statements that answer the following queries. Use ONLY the views created above or create your appropriate view. Note that a credit balance means that the customer paid more than what they owe, that is, their balance is less than 0.

   (a) [10 points]List the full name and cell phone number of the customer(s) who has the maximum credit (i.e., minimum balance) in their account(s).

   (b) [10 points]List the full name, outstanding balance, and number of received calls of the customer who received the most calls from anyone (i.e., not limited to P_Mobile customers).

   (c) [10 points]List the full name and cell phone number of the customer(s) who has the lowest outstanding balance excluding the ones that have credit in their accounts.

   (d) [15 points]Find the ratio of number of people who are not *P_Mobile* customers and live in Pittsburgh to the total number of people living in Pittsburgh (i.e., percentage of potential market for *P_Mobile* in Pittsburgh).

3. [5 points] INSERT the following tuples into your database:

   (a) A payment of $90 to the account 4124564564 on February 2nd, 2020.

   (b) A person (non customer) with the following information:

      i. Name: John Do
      ii. Phone: 1234565089
      iii. Street: 123 Cool St
      iv. City: Pittsburgh
      v. State: PA
      vi. Zip: 15213

4. [10 points] Run the following queries:

   (a) SELECT * FROM MV_OUTSTANDING_BAL WHERE cell_pn = 4124564564;

   (b) SELECT * FROM PA_NOT_CUSTOMERS WHERE pn = 1234565089;

   Now, 1) refresh the materialized view 'MV_OUTSTANDING_BAL', 2) re-run the above SELECT statements, and 3) record your observations from the run before the refresh, and the run after the refresh. (include your answers as a comment in your 'db' script)

Comments and Suggestion

- In order to refresh materialized views, run the following (note the single quotes are not part of the name):

```
BEGIN
 DBMS_MVIEW.REFRESH('<name_of_materialized_view>');
END;
/ -- use this only in SQLplus
```

- If the data type of an attribute is string, you should define appropriately (long enough) to avoid unsuccessful insertion using provided sample data.

- The list of Oracle errors ORA-XXXX are available under doc. Also you can get them by searching them in the Internet.
- Providing additional information in submission files (see **What to submit**) should be done by using the comment feature of SQL: any text after a double dash is considered a comment until the end of the line. For example, add the lines below, before the SQL query in your answer to Question #2:

  ```
  --------------------------------------------
  -- Question #2:
  ```

  You are also encouraged to leave a couple of empty lines between answers, for better clarity.

What to submit

You are required to submit **two** text files under your **pitt_user_name** (e.g, pitt01). **include your name and pitt user name at the top of the text file**, and identify the question number before each answer

- **<pitt_user_name>-db.sql**
  In this file, please submit the answers to all questions (i.e., DROP, CREATE, ALTER and INSERT statements, etc) including the answers to Q4 as a comment. In addition to providing the answers, you are expected to:
  - **include your name and pitt user name at the top of the text file**, and
  - use SQL **DROP TABLE/VIEW** statements at the beginning of this file so that you can make sure your database does not have pre-existed tables.

  The entire text file should be composed of **valid SQL statements**.
- **<pitt_user_name>-output.txt**
  In this file, please submit the query results of your 'db' script. You could use the command "SPOOL log_file_name" in SQLPLUS to record your query results.

How to submit your assignment

1. Submit your assignment(the 2 files described above) through the Web-base submission interface you have used to submit the previous assignments. **It is your responsibility to make sure the assignment was properly submitted.**.
2. Submit your assignment by the due date (8:00pm, Wednesday Feb. 27, 2020). There is no late submission.

Academic Honesty

The work in this assignment is to be done *independently.* Discussions with other students on the assignment should be limited to understanding the statement of the problem. Cheating in any way, including giving your work to someone else will result in an F for the course and a report to the appropriate University authority.