

CS 2550 – PRINCIPLES OF DATABASE SYSTEMS (SPRING 2017)
DEPT. OF COMPUTER SCIENCE, UNIVERSITY OF PITTSBURGH

Assignment #2: SQL

Release: January 17, 2017

Due: 11:59 PM, January 31, 2017.

Description Assume the following relational database schema that supports technical support ticketing system, *CS-Tech*, that keeps track of the tickets, their status, who is working on it, and who submitted it:

- TECH_PERSONNEL (pplSoft, fname, lname, pittID, expertise, office_phone)
where fname is first name, and lname is last name.
- USERS (pplSoft, fname, lname, pittID, office_phone)
- USER_OFFICE(office_no, building)
- CATEGORIES (category_id, category, description).
Where this table lists all possible categories of submitted tickets.
- INVENTORY(machine_name, IP, network_port, MACADDR, location_id)
- LOCATIONS(location_id, location, building, notes)
- TICKETS (ticket_number, owner_pplSoft, date_submitted, date_closed, days_worked_on, category_id, machine_name, description)
- ASSIGNMENT (ticket_number, tech_pplSoft, date_assigned, status, outcome)
Where status field is an enumeration, could be: assigned, in_progress, delegated, closed_successful, or closed_unsuccessful.

Assumptions We assume the following:

- The pplSoft is unique for students, faculty, and staff. Tickets could be assigned to only one personnel from tech at a time, but could be reassigned anytime to another personnel.
- When a user submits a ticket, the ticket info are stored in the TICKET table. Later, when a tech_personnel is in charge of it, an entry is added to the ASSIGNMENT table, with ticket status assigned. The in_progress status means that some effort was put into solving the ticket, but not finished yet. Delegated status means that this personnel delegated the ticket to another personnel, may be because of the field of expertise. Once the ticket is resolved, its status is set to closed_successful in the last assignment records. If, on the other hand, the tech failed to resolve it, they may close the ticket unsuccessful, setting the status to closed_unsuccessful.
- State your own assumption clearly if your solution is not based on the provided description.

Questions

Answer the following questions [for a total of 100 points]:

1. [16 points] Use CREATE TABLE statement to create tables for these relations. You need to define the primary keys, foreign keys (if any) and alternate keys (if any) in your statement. You need to make all primary keys deferrable and all foreign key constraints to be initially deferred deferrable. You also need to specify the integrity constraints (e.g., NOT NULL and default values) based on your assumed semantics and explained your assumptions. Make these semantic integrity constraints initially immediate deferrable.
2. [10 points] After creating the above tables, use ALTER TABLE statement to incorporate the following in the *CS_Tech* database schema.
 - (a) In the TICKETS table, set the default value of days_worked_on attribute to be zero.
 - (b) In the TICKETS table, enforce that the days_worked_on has a non-negative value.
 - (c) Add a new attribute supervisor to table TECH_PERSONNEL, which is the pplSoft of the personnel's supervisor, and set its default value to be Bob Hoffman's pplSoft from the sample data `sample-data.txt` (1110001).
 - (d) Modify the USER_OFFICE table by adding the attribute pplSoft which is a foreign key constraint FK_USER_OFFICE that references USERS(pplSoft). Make FK_USER_OFFICE not deferrable. Note that offices can be shared among multiple persons.
 - (e) In USER_OFFICE table, drop the constraint FK_USER_OFFICE and add it again only this time make it deferrable.
3. [0 points] After creating the database using your SQL statements, populate the database according to the data in `sample-data.txt` using the SQL INSERT command.
4. [25 points] Express the following queries in single SQL expressions and answer them using the database you have created above.
 - (a) [1 point] List the first name, last name and phone number of all users who submitted more than 3 tickets. List them in a descending order of number of tickets submitted.
 - (b) [2 points] List the first and last names of all Tech personnel who worked on resolved tickets more than any other tech personnel, and the number of tickets.
 - (c) [4 points] List as Device Name all names of the machines that had the maximum number of problems in the two months of December 2015 and January 2016.
 - (d) [5 points] Calculate the average number of days each ticket is being worked on, for tickets submitted during the month of January 2016.
 - (e) [6 points] Calculate the total number of days spent on resolving tickets for the month of January 2016 for each tech personnel. List them in an ascending order.

- (f) [7 points] Using a single SQL query, display the week (e.g., 1-52), start and end date and the number of successfully closed tickets that were submitted weekly (i.e., every 7 days) between January 1, 2015 and December 31, 2015, sorted on the week in descending order.
5. [10 points] Create the following views:
- (a) [2 points] A view named UnresolvedTKTS that lists fname, lname of users, along with the unresolved tickets' numbers, date submitted, the description of the ticket. Unresolved tickets means those with no value entered in the closed_date attribute of the TICKETS table.
 - (b) [4 points] A view named ProblematicMachine that lists all information about the machines, for which more than 5 tickets were opened during the previous month (i.e., if we run the view in Oct, then it lists tickets for which more than 5 tickets were submitted in Sep, and if we run it in Nov, it lists tickets for which more than 5 tickets were submitted in Oct, and so on).
 - (c) [4 points] A view named TechPerformance which lists the number of closed tickets (i.e., data_closed attribute has a value in TICKETS table) that each tech personnel resolved (i.e., the status attribute has the value 'closed_successful'). The view should be sorted in a descending order.
6. [10 points: 5 points each] Write the SQL statements that answer the following queries.
- (a) Find the machine(s) located in fifth floor, that has(have) the most number of submitted tickets. You can make use of the created views if needed.
 - (b) Find the day of the week that has the most number of submitted tickets in December 2015. For example, if you count all tickets submitted on all Mondays and they are 100, and there is no other day that has more tickets, then your answer should be Monday, and how many tickets were submitted. You can make use of the created views if needed.
7. [10 points: 5 points each] Write the following triggers.
- (a) Write a trigger called ClosedTicket that whenever the status attribute of the ASSIGNMENT table is changed to 'closed_successful', or to 'closed_unsuccessful', it sets the value of the date_closed attribute of the TICKETS table to the current date.
 - (b) Write a trigger called WorkedDays that whenever the status of a ticket is about to change from 'in_progress' to: 'delegated', 'closed_successful' or 'closed_unsuccessful', it calculates number of days between the current date, and the date_assigned and updates the days_worked_on attribute of TICKETS table accordingly.
8. [19 points]
- (a) [5 points] Create a view named WORKING_TICKETS that lists all the information about tickets along with the first name and last name of the tech personnel who last worked on each ticket and its current status.
 - (b) [5 points] Create a materialized view MV_WORKING_TICKETS for the view WORKING_TICKETS. MV_WORKING_TICKETS should be built immediately.

- (c) [5 points] Write 2 SQL statements Q1 and Q2 that use the views MV_WORKING_TICKETS and WORKING_TICKETS, respectively, to find the number of tickets closed successfully.
- (d) [4 points] Compare the execution time of Q1 and Q2. For this you need to invoke `set timing on;` in *sqlplus*. Also in order to get meaningful results you need to experiment with a large number of tickets. You can use the provided script to generate a large datasets by invoking it as: `ticketGen.pl -t XXX`, where XXX is the number of tickets. Experiment with datasets as large as 500, 1000 and 1500 and report the results.

Comments and Suggestions

- Note that CREATE DOMAIN statement is not supported in Oracle.
- 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 googling.
- 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

```
-----  
-- Question #4:
```

before the SQL query that is your answer to Question #4. You are also encouraged to leave a couple of empty lines between answers, for better clarity.

What to submit

You are required to electronically submit **exactly three** text files, all under your `pitt_user_name` (e.g., pitt01).

- **pitt_user_name-db.sql**
In this file, please submit the answers to question 1 and 2. (i.e., CREATE TABLE, ALTER TABLE and INSERT statements.) 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
 - identify the question number before each answer using SQL comments.
 - you must use SQL **DROP TABLE** statements at the beginning of this file so that you can make sure your database does not have pre-existed tables which have the same name as those 6 tables in this assignment.
 - you must use PURGE RECYCLEBIN statement to empty your recycle bin and prevent it from overflowing.

The entire text file should be composed of **valid SQL statements**.

- **pitt_user_name-query.sql**

In this file, please submit the answers to questions 3 to 8. 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
- identify the question number before each answer
- At the beginning of this file, you must write simple queries to list the content of the seven 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 pitt01-query.sql. You could use the command “SPOOL log_file_name” in SQLPLUS to record your query results. 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
- identify the question number before each answer

To submit your assignment:

1. Submit your assignments through the Web-based submission interface (go to the class web page <http://db.cs.pitt.edu/courses/cs2550/current.term> and click the Submit button). **It is your responsibility to make sure the assignment was properly submitted.**
2. You must submit your assignment before the due date (**11:59 PM, January 31, 2017**). The timestamp of the electronic submission will determine if you have met the deadline.
3. It is your responsibility to submit a file that is working properly.

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.