

Assignment#2: Relational Model, SQL DDL & Basic DML

Release: Sep. 3, 2020

Due: 8:00 PM, Sep. 15, 2020.

Goal:

Gain familiarity with SQL DDL and basic DML (i.e., SELECT... FROM.. WHERE). You will create and manipulate the database using SQL in the PostgreSQL server.

Description:

Consider 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 them, and who submitted them:

- 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)
- CATEGORIES (category_id, category, description).
Where this table lists all possible categories of submitted tickets. Examples of categories are *Display_problem*, *HD_problem*, *OS_update*, *Virus_infected*, etc.
- INVENTORY (machine_name, ip_address, network_port, mac_address, location_id)
- LOCATIONS (location_id, location, building, notes)
- TICKETS (ticket_number, owner_pplSoft, date_submitted, date_closed, days_worked_on, category_id, machine_name, location, description)
- ASSIGNMENTS (ticket_number, tech_pplSoft, date_assigned, status)
Where status field is an enumeration, could be: *assigned*, *in_progress*, *delegated*, *closed_successful*, or *closed_unsuccessful*.

Assumptions:

We assume the following:

- **pplSoft** is unique for students, faculty, and staff. Tickets could be assigned to only one staff from tech at a time, but could be reassigned anytime to another tech staff.
- When a user submits a ticket, the ticket info is stored in the TICKET table. Later, when a tech staff is in charge of it, an entry is added to the ASSIGNMENTS table, marking the ticket's **status** to be *assigned*. The *in_progress* status means that some effort was put into solving the ticket, but not finished yet. The *delegated* status means that this tech staff delegated the ticket to another tech staff, could be because of the field of **expertise**. Once

the ticket is resolved, its status is set to *closed_successful*. If, on the other hand, the tech failed to resolve it, they may close the ticket unsuccessfully, setting the ***status*** attribute value to be *closed_unsuccessful*.

- The attribute ***description*** in the tables CATEGORIES and TICKETS is a free text type that has a string length limit similar to Twitter. That is 280 characters, but the most common length of a description is 33 characters.
- Choosing an appropriate data types for attributes is crucial, whether it is a character, varchar, integer, date, etc. Also, state your assumption on why you chose a certain data type.
- State your own assumption(s) clearly if your solution is not based on the provided description.

Questions:

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

1. [10 points] For each table given in the **Description** above, list its *arity* and *cardinality* before creating the tables in PostgreSQL.
2. [42 points] Use SQL CREATE TABLE statement to create tables for the relations given in the **Description** above. You need to define the primary keys, foreign keys (if any), alternate keys (if any), as well as any additional constraints, such default values and whether an attribute can take a NULL as a value or not.
3. [20 points: 4 points each] Use ALTER TABLE statement to incorporate the following in the *CS_Tech* database schema.
 - (a) Drop the attribute ***location*** from table TICKETS.
 - (b) Add a new attribute ***outcome*** to table ASSIGNMENTS, which is a free text (similar to ***description*** attribute) explaining the results of working on that ticket.
 - (c) Add a new attribute ***years_of_experience*** to table TECH_PERSONNEL with default value zero.
 - (d) Add a new attribute ***supervisor*** to table TECH_PERSONNEL, which is the pplSoft of the personnel's supervisor, and set it's default value to be Bob Hoffman's pplSoft from the sample data `sample-data.txt` (1110001).
 - (e) In the TICKETS table, set the default value of the attribute ***days_worked_on*** to be zero.
4. [10 points] After creating the database using your SQL statements, populate the database according to the data given in the file `sample-data.txt` using the SQL INSERT command. Note that the data type "date" has many formats, make sure to use the function *TO_DATE*('01-01-1990', 'MM-DD-YYYY') in order to store a string as a date correctly.
5. [10 points] Create a test set of SQL Insert statements to check the validity of your specified constraints in the create statements. These could be variations of the provided sample data above (Question 4). As in any system development, you need to have an insert statement that checks each constraint, e.g., Foreign key violation, or inserting NULL to an attribute that cannot take a NULL.

6. [8 points: 4 points each] Express the following queries in SQL and answer them using the populated database from Question 4.
- (a) Retrieve the ticket numbers and status of successfully closed tickets.
 - (b) Retrieve the ticket numbers and descriptions for the machine `kas0.cs.pitt.edu`.

What to submit:

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

1. **<pitt_user_name>-db.sql (e.g., pitt01-db.sql)**

In this SQL script file, please submit the answers to Questions 1, 2, 3, and 4. For the first question, please provide your answer as an SQL comment. Then answer Questions 2, 3, and 4 in valid SQL statements (i.e., CREATE TABLE and ALTER TABLE). In addition to providing the answers, you are expected to:

- **include your name and pitt user name at the top of the SQL script as an SQL comment**, and
- identify the question number before each answer using SQL comments.
- you must use SQL **DROP TABLE IF EXISTS <table_name>;** 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 the tables described above in this assignment.

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

2. **<pitt_user_name>-my-test-data.sql (e.g., pitt01-my-test-data.sql)**

In this file, please submit the SQL insert statements that you have generated to test the correctness of your integrity constraints specified in your create table statements in Question 5.

- **include your name and pitt user name at the top of the SQL script as an SQL comment**, and
- identify the purpose of each insert before each insert as SQL comments. E.g.,
 - This Insert test the validity of the FK ticket number in table ASSIGNMENTS

3. **<pitt_user_name>-query.sql (e.g., pitt01-query.sql)**

In this SQL script file, please submit the answers to Question 6. In addition to providing the answers, you are expected to:

- **include your name and pitt user name at the top of the SQL script as an SQL comment**, and
- identify the question number before each answer using SQL comments.
- at the beginning of this file, you must write simple queries to list the content of the tables described above.

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

4. **<pitt_user_name>-output.txt (e.g., pitt01-output.txt)**

In this file, please submit the output of your query SQL script (i.e., the one that is named <pitt_user_name>-query.sql). To do that in PSQL you can use the command “\o” as seen in the recitation video.

In addition to providing the answers, you are expected to:

- **include your name and pitt user name at the top of the text file.**

How to submit it:

1. After preparing your solution, submit your files (i.e., three SQL script files and a text output file) that contain your solution through the Web-base submission interface you have used for Assignment #1. **It is your responsibility to make sure the assignment was properly submitted.**
2. Submit your files by the due date **(8:00pm Sep. 15, 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.