

Assignment 4

CSC 370, Fall 2013

Due: December 4, 2013 (code submitted via `conneX` before 10:30 am.)

Important

All work for this assignment is to be done using PostgreSQL 9.1 and the `psycopg2` Python module. Each student was assigned their own account/database for Assignment 3 on `studentdb.csc`, and you must use the same account and password for your account in this assignment.

Learning Outcomes

By the end of this assignment, you should be able to:

- Write a Python program that connects to your Postgres DBMS.
- Execute `select` queries from within the program and integrate the results from the queries with the rest of your program.
- Execute modification statements (`update`, `delete` and `insert` operations) from within the Python script.
- Utilize exception handling to provide appropriate program behavior in the event of unexpected events when interacting with the DBMS.
- Interact – in a simple way – with a user (who is a hypothetical employee of the “Green-not-Greed” organization) by accepting the user’s inputs and responding with appropriate results.

A continuation...

In the previous assignment you modelled the database needed for a hypothetical environmental-activist organization named “Green-not-Greed” (GnG). In this assignment you will use the tables, queries, views, etc. and use them from within a Python program that you write. This program is intended to be used by employees of GnG who are going about the tasks of the organization. In order to keep our focus on interacting with the database (i.e., the classical “application-server” tier level of coding) we will keep the user interface very simple (i.e., text based). However, the information provided by the program to the user should be clear – that is, cryptic prompts and confusing output must be avoided.

As you write the program you may decide that some of your tables and queries need to be refactored. You are allowed to do this, but please ensure that these changes are carefully documented in a `CHANGES.txt` document.

In each of the phases below you must use exception handling where appropriate. Working with databases is fraught with peril – much can go wrong, and we want the user isolated (as best possible) from DBMS errors. If an exception is likely to occur, you must handle it.

Phase 1: Queries from Assignment 3

Your program must allow the GnG user to select from the queries you prepared in Assignment 3, and to produce the output in a way the user can read and interpret. Note that your queries should be briefly described; simply allowing users to select from “query 1”, “query 2”, ..., “query n” is not enough.

Phase 2: Setting up a Campaign with volunteers, activities, etc.

Your program will enable a GnG user to enter the information needed to set up a campaign, including adding new volunteers to organization who have joined specifically to help with the campaign, scheduling events, etc. Existing volunteers can help with the campaign. The GnG user must be able to see the state of a campaign at suitable points during campaign setup.

Phase 3: Some accounting information

GnG does have donors and campaigns have costs. Reporting on these fund inflows and outflows is often needed as the group plans for future events. The GnG user must be able to obtain this kind of data in both a textual and a quasi-graphical format (e.g, ASCII bar charts) and at a suitable level of detail.

Phase 4: Membership history

Another useful tool when planning is to invite specific members/volunteers to take on tasks for campaigns, yet organizers are very mindful that time is finite. Member burnout is commonplace in many volunteer organizations, and we can help organizers here by showing them the way members have been involved. Some of this browsing through membership history may also suggest ideas or annotations to be added to campaigns or member records. The GnG user must be able to browse both membership history and add annotations to campaigns, member records, or any other piece of data you believe appropriate.

Phase 5: Your own idea(s)

Create one more way in which the GnG user can interact with the database – via the Python program – in a non-trivial way.

What to submit via connex

- (a) A single Python script named `gng.py` that implements your functionality for the assignment. If you wish to submit more than one Python file, however, you must first obtain written permission from the instructor.
- (b) A text file named `PHASE5.txt` which lists the functionality you have invented for the GnG program in Phase 5 (above).
- (c) The SQL file produced by the `pg_dump` command for your database. This file contains all of the SQL commands needed to reconstruct your database with all tables, table data, and views (i.e., your SQL queries).
- (d) A text file named `CHANGES.txt` which lists the changes you have made to the queries or tables (or both) from what you submitted for Assignment 3.

Evaluation

As there are many possible correct solutions to the problem, evaluation will be done via a demonstration of work in front of a member of the CSC 370 teaching team. Information on demos (i.e., where, when, how to sign up) will be distributed shortly before the assignment due date.

The marking scheme below will be used:

- “A” grade: An exceptional submission demonstrating creativity and initiative. The program provided is thorough and shows insight, enables the user to interact with the data in a clear way, and is free of unexplained behavior. Phase 5 shows exceptional creativity.

- “B” grade: A submission completing the requirements of the assignment. The program provided is thorough, enables the user to interact with the data, and is free of unexplained behavior. Phase 5 has been completed and is non-trivial.
- “C” grade: A submission completing most of the requirements of the assignment. The program has some rough spots, but the user is able to interact with the data. There is some unexplained behavior. Phase 5 might not have been completed.
- “D” grade: A serious attempt at completing the requirements of the assignment. There are many problems with the submitted work.
- “F” grade: Either no submission is given, or submission represents very little work.