

**CMPSC 380**  
**Principles of Database Systems**  
**Fall 2016**

**Laboratory Assignment Three: Using the SQL Data Manipulation Language**

## Introduction

In this laboratory assignment, you will create a wide variety of DML commands like the SQL select, insert, update, and delete and execute them against a SQLite database. You will continue gaining experience with the SQL programming language by developing a database with real-world data. You will then learn how to run SQL queries using Python programming language. **This is a team-based lab, you must work in teams of two (preferably) or three. You may select your own team member.**

## Downloading and Preparing the Data Sets

During the completion of this laboratory assignment, you will develop a database system with real-world data set(s) of your choice. Your first step should be to choose an interesting data set(s) that is publicly available, for example from <http://catalog.data.gov/dataset> or <https://archive.ics.uci.edu/ml/datasets.html>. In selecting appropriate data set(s), you should keep in mind that you must create more than one table which will be involved in your queries. This can be done by dividing up your data set by some logical means (i.e., select data with enough columns for this to be possible). Download your data set and ensure it is in the format you can work with in creation of the tables. Discuss all steps with your team member(s).

## Creating Database Tables

Next, you should decide on how to organize your data. First, you need to draw an Entity-Relational Diagram (schema) to visually represent the relationship between your tables. Then, using the process utilized in the previous labs, create the tables while correctly specifying primary keys. Finally, you need to populate your tables with your selected data. As you are populating your database with data, please make sure that you take into account the integrity constraints defined for the schema which were discussed in class. Discuss all steps with your team member(s).

## Querying the Database through Python

In this part of the tutorial, you will learn how to run queries through a Python programming language. To learn more about Python, please consult the following free book resource:

- <http://greenteapress.com/wp/think-python/>

Please be sure to ask the instructor questions as needed. Follow the SQL with Python code tutorial found at the following link:

- [https://www.tutorialspoint.com/sqlite/sqlite\\_python.htm](https://www.tutorialspoint.com/sqlite/sqlite_python.htm)

Once you have completed the tutorial, you should have an understanding of how to query SQLite databases using only the Python programming language. Now, write a program in Python (following the examples in the tutorial) that will demonstrate your abilities and run queries on your own database using the following statements:

- INSERT
  - In addition to running the four `insert` statements that will conform to the integrity constraints, you should try to run at least one `insert` that will be rejected by SQLite because of the fact that it violates the database’s constraints.
- DELETE
  - Now you should run at least one `delete` statement and observe how it changes the state of the database.
- UPDATE
  - Section 3.9.3 explains the operation of the SQL `update` statement. For each of the tables in your chosen schema, you should run at least one `update` statement that will yield data that adheres to the integrity constraints. Again, as you run the `updates`, please take care to record the “before” and “after” state of the database, in addition to commenting on why the `update` changed the database in the way that it did. If possible, try to execute a wide variety of different `update` operations.
- SELECT
  - The SQL `select` statement allows you to query the database, thereby discovering what data values are stored inside of it. To learn more about the basic structure of SQL queries, please review Sections 3.3 and 4.1 of your textbook. Then, you should write at least ten `select` statements that interact with a wide variety of the tables in the relational database. Whenever possible, these queries should also use various features of the SQL language, **such as queries on multiple tables, aggregation, and joins**. Please see the course instructor if you encounter trouble with the `select` statements.

## Summary of the Required Deliverables

This assignment invites you to submit an electronic version of the following deliverables through Bitbucket. Please create a subdirectory in your “labs” directory called “lab3”. Please also submit one lab response from team. Here, you will have to decide which member of the group is going to make this submission for your group. Submit the following materials. Finally, do not forget to add your names to this typed-up submission which is preferably, written in L<sup>A</sup>T<sub>E</sub>X.

1. A commentary on the general structure of `select`, `update`, `insert`, and `delete` statements. These make up four queries that must be included with this lab.
2. Convincing evidence to demonstrate that your create table statements ran correctly. SQLite3 may be used, for the table creation. Queries are all to be performed via Python coding.

3. The final listing of the create table statements for your database schema.
4. The completed and properly commented Python source code which successfully performs all your queries. This code could have all query statements found in series in the code. In your Python code, please use print statements to describe what the query is doing.
5. For each query command where the table is altered, include in your response, a descriptive selection of some of the rows *before* and *after* they are modified by your Python program.
6. A brief reflection on the challenges that you faced when completing this laboratory assignment.

You must place all of the required deliverable into a Bitbucket repository directory named `lab3`. Please see the instructor if you have any questions about this assignment.