

Homework 02

CSC/DSCI 1301 – Principles of CS/DS I

Due Date: November 12th at midnight

Please thoroughly read the instructions for the homework and its deliverables! Your program should be typed up. We will not accept handwritten code or images of code as submissions.

Program: Python Database

In the last week, we discussed dictionaries and their built-in methods in detail. Python dictionaries can be used as a rudimentary database to store information. Databases are used to store and manage all sorts of information, including customer records, product catalogs, financial data, and scientific data. This is what you will be implementing in this homework assignment.

Dictionaries as a Database

Databases are made up of tables, which are like spreadsheets. Each table has rows and columns, and each row represents a single record. The columns represent different pieces of information about the record. For example, a customer database might have a table with columns for the customer's name, address, phone number, and email address.

A Python dictionary can be used to represent a record of a database table. The keys of the dictionary are the names of the columns, and the values are the data stored in those columns. These record dictionaries can be collected and stored in another container like a list or another dictionary to build a database table.

Typically, most tables have a special column that serves as a unique identifier for each record in the table. This is commonly an integer ID column. For example, our Panther ID numbers serve as unique identifiers of members at GSU.

Your database tables must have a unique identifier for each record. This will allow the user to specify specific records to update, remove, or print from the database. Because a unique identifier is required, the database tables should also be implemented as nested dictionaries. The keys of the table dictionaries should be the unique identifier for the records stored within them and the values of the table dictionaries should be the record dictionaries containing the columns.

Your Database

It is up to you to decide what information your database will store. The requirements for your Python database are as follows:

1. The database must contain at least 2 tables.
2. Each table must also contain at least 5 columns.
3. Each table must contain columns of at least 2 different data types.
4. The records of each table must have their own unique identifier.

Menu Commands

Beyond just storing data, a database must also give users a way to access the data stored within them. Your database program should allow the user to enter multiple menu commands as input until a stop word is detected to end the program. There should be at least two valid stop words, one of which should be a single letter shortcut. Ex. 'quit' and 'q'.

Add

There should be at least two valid add commands for each of the tables in your database. One of which should be a shortcut. Ex. 'add student' and 'as'. After reading this command, the program should then prompt the user to add all the required information for each column in the table.

Remove

There should be at least two valid remove commands for each of the tables in your database. One of which should be a shortcut. Ex. 'remove student' and 'rs'. After reading this command, the program should then prompt the user to specify the ID of the record they wish to remove from the table.

Print

There should be at least two types of print commands for each of the tables in your database. The first print command should print all the records in the table, while the second print command should allow the user to print a specific record from a table.

There should also be two valid print commands for each type of print. One of which should be a shortcut. Ex. 'print student' and 'ps'. After reading this command, the program should either print all the records of the table into the terminal or it should prompt the user for the record identifier and print only the specified record.

Update

There should be at least two valid modify commands for each of the tables in your database. One of which should be a shortcut. Ex. 'update student' and 'us'. After reading this command, the program should prompt the user to specify the ID of the record they wish to modify, then prompt the user to specify the name of the column they want to update, and finally prompt the user to specify what the new value is.

Error Handling

You should implement some simple error handling logic into your code. Your program should handle invalid inputs gracefully and not crash. Your program should inform the user if the command they entered is invalid. The message should be informative and context specific! Inform the user if the command keyword is wrong, if the table name is wrong, or if the record ID is wrong.

Design Document

In addition to your program, you will need to write a short design document describing the organization and functionality of your code. This document does not need to be more than a page long. However, it must cover all the following points.

1. A diagram or description of the database tables implemented in your database.
 - a. The **name** of every **table** in your database.
 - b. The **name** and expected **data type** of every **column** in your database.
 - c. An **explanation** of how you have implemented them in Python.
2. A description of all the valid menu commands for your program.
 - a. The command **keywords** and their correct **syntax**.
 - b. **Examples** of their usage.

Deliverables

For this program you will need to provide the python files containing your code as well as a recording or screenshots of the output of your program. Please name your files as follows:

- Design Document/Instruction Manual
 - lastname_firstname_filename.pdf
 - For example: **hawamdeh_faris_database.pdf**
- Python Files
 - lastname_firstname_filename.py
 - For example: **hawamdeh_faris_database.py**
- Recording
 - lastname_firstname_filename.mp4
 - For example: **hawamdeh_faris_database.mp4**
- Screenshots
 - lastname_firstname_filename.png
 - For example: **hawamdeh_faris_database.png**

Important

Homework should be your own **individual effort**! You may not work on this as a group. If you have questions about this assignment, please ask your instructor, your Lab TAs, or a Tutor. Do not ask your peers!

Your style of coding is just as important as the correctness of your program. Unlike your programs in the lab, you will be evaluated on your variable names and use of comments in addition to the program output. Your code must include a block comment at the top of your Python file with your name and a description of your program. Your variable names should be meaningful and conform to snake_case.