Chapter 16 – Extra Practice

Extra Practice

Extra practice is for those who would like to do some extra practice projects to further hone their skills learned in each assignment. There are no additional points to be gained by completing these projects.

Player Manager

Create a program that allows you to store data for players of a game.

```
Player Manager

COMMAND MENU
view - View players
add  - Add a player
del  - Delete a player
exit - Exit program

Command: view
Name             Wins    Losses    Ties    Games
-------------------------------------------------
Mike               4        3         7       14
Joel               3        7        10       20

Command: add
Name: anne
Wins: 9
Losses: 5
Ties: 3
Anne was added to database.

Command: view
Name             Wins    Losses    Ties    Games
-------------------------------------------------
Anne               9        5         3       17
Mike               4        3         7       14
Joel               3        7        10       20

Command: del
Name: anne
Anne was deleted from database.

Command: exit
Bye!
```

Specifications

- In the Assignment 16 section of Canvas, you will find a database file (players_db.sqlite) that contains a Player table that stores the data for each player.
- Use the three-tier architecture (presentation, business, database) for this program, and store the code for each tier in a separate file.
- Display the players in order by wins, starting with the player with the most wins.
- Assume that the name for each player is unique.

Possible Enhancements

- Add an "update" command. This command should prompt the user to enter the name of a player. Then, it should let the user update the wins, losses, and ties for the player.

Task List

Create a program that allows you to manage a task that's stored in a database.

```
Task List

COMMAND MENU
view      - View pending tasks
history   - View completed tasks
add       - Add a task
complete - Complete a task
delete    - Delete a task
exit      - Exit program

Command: view
1. Buy toothbrush
2. Do homework

Command: complete
Number: 2

Command: add
Description: Pay bills

Command: view
1. Buy toothbrush
2. Pay bills

Command: history
1. Get bike fixed (DONE!)
2. Call your mom (DONE!)
3. Do homework (DONE!)

Command: exit
Bye!
```

Specifications

- In the Assignment 16 section of Canvas, you will find a database file (task_list_db.sqlite) that contains a Task table that stores the tasks.
- Use the three-tier architecture (presentation, business, database) for this program, and store the code for each tier in a separate file.
- The view command should only display tasks that have not been completed.
- The complete command should only mark a task completed, not delete it from the database.
- The history command should allow you to view tasks that have been completed, but not deleted.

Product Manager

Create a program that manages the products that are available from a guitar shop.

```
Product Manager

CATEGORIES
Guitars | Basses | Drums

COMMAND MENU
view   - View products by category
update - Update product price
exit   - Exit program

Command: view
Category name: basses
Code       Name                                             Price
------------------------------------------------------------------
precision  Fender Precision                                 799.99
hofner     Hofner Icon                                      499.99

Command: update
Product code: hofner
New product price: 399.50
Product updated.

Command: view
Category name: basses
Code       Name                                             Price
------------------------------------------------------------------
precision  Fender Precision                                 799.99
hofner     Hofner Icon                                      399.50

Command: exit
Bye!
```

Specifications

- In the Assignment 16 section of Canvas, you will find a database file (guitar_shop.sqlite) that contains Category and Product tables. These tables store the data for the categories and the products within each category.
- Use the three-tier architecture (presentation, business, database) for this program, and store the code for each tier in a separate file.
- Display the products alphabetically by product name.
- Assume that the name for each category is unique.
- Assume that the code for each product is unique.