Assignment #4 – Storage Concepts

To begin, open Jupyter Notebook through your preferred environment such as Google Colab or Anaconda. Once opened, create a new notebook. In Google Colab this is done by selecting **File** then **New Notebook**. Next, carefully write the code provided into each cell of the notebook as shown, including any comments.

Execute each cell by clicking the **Run or Play** button located on the left side of each cell. Should you have an error during execution, review the code within the cell. Pay close attention to potential typos, incorrect spacing, or misspellings, verifying each line against the provided code. Correct and rerun the cell.

Continue this process for all cells. Then save as a pdf by selecting **File** then **Print**.

Submit PDF to Canvas. Code and output must be clearly identified for full credit.

50 points

```
[ ] # database
    import sqlite3
```

```
[ ] # store data

    # create a todo list table
    def create_todo_table(conn):
        cursor = conn.cursor()
        cursor.execute('''CREATE TABLE IF NOT EXISTS todos (
                        id INTEGER PRIMARY KEY,
                        task TEXT NOT NULL,
                        completed INTEGER DEFAULT 0
                        )''')
        conn.commit()
```

```
[ ] # interact/manipulate

    # add a new task to the todo list
    def add_task(conn, task):
        cursor = conn.cursor()
        cursor.execute('INSERT INTO todos (task) VALUES (?)', (task,))
        conn.commit()

    # update the completion status of a task, 1 complete, 0 not complete
    def update_task_status(conn, task_id, completed):
        cursor = conn.cursor()
        cursor.execute('UPDATE todos SET completed = ? WHERE id = ?', (completed, task_id))
        conn.commit()

    # delete a task from the todo list
    def delete_task(conn, task_id):
        cursor = conn.cursor()
        cursor.execute('DELETE FROM todos WHERE id = ?', (task_id,))
        conn.commit()
```

```
[ ] # retrieve data

    # retrieve/view all of the tasks in the todo list
    def get_tasks(conn):
        cursor = conn.cursor()
        cursor.execute('SELECT * FROM todos')
        tasks = cursor.fetchall()
        return tasks
```

```
[ ] # Main function
    def main():

        # connect to the SQLite database
        conn = sqlite3.connect('todo.db')

        # create the todo list table
        create_todo_table(conn)

        # loop until you exit
        while True:
            # menu interface for tasks
            print("\nTODO LIST")
            print("1. Add Task")
            print("2. View Tasks")
            print("3. Update Task Status")
            print("4. Delete Task")
            print("5. Exit")

            choice = input("Enter your choice: ")

            if choice == '1':
                task = input("Enter task: ")
                add_task(conn, task)
                print("Task added successfully!")

            elif choice == '2':
                tasks = get_tasks(conn)
                if not tasks:
                    print("No tasks found.")
                else:
                    for task in tasks:
                        print(f"{task[0]}. {task[1]} - {'Completed' if task[2] else 'Incomplete'}")
```

```
            elif choice == '3':
                task_id = int(input("Enter task ID: "))
                completed = int(input("Enter completion status (1 for completed, 0 for incomplete): "))
                update_task_status(conn, task_id, completed)
                print("Task status updated successfully!")

            elif choice == '4':
                task_id = int(input("Enter task ID: "))
                delete_task(conn, task_id)
                print("Task deleted successfully!")

            elif choice == '5':
                print("Exiting...")
                break

            else:
                print("Invalid choice. Please try again.")

        # close the database connection
        conn.close()
```

```
if __name__ == "__main__":
    main()
```