sqlite3

September 30, 2023

[3]:
```
import sqlite3

# Create or connect to the database
conn = sqlite3.connect('tasks.db')
cursor = conn.cursor()

# Create the 'tasks' table if it doesn't exist
cursor.execute('''
    CREATE TABLE IF NOT EXISTS tasks (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        task_name TEXT,
        task_description TEXT,
        task_status TEXT
    )
''')
conn.commit()

def add_task(task_name, task_description):
    cursor.execute('INSERT INTO tasks (task_name, task_description,␣
 ↪task_status) VALUES (?, ?, ?)',
                   (task_name, task_description, 'Pending'))
    conn.commit()

def get_tasks():
    cursor.execute('SELECT * FROM tasks')
    return cursor.fetchall()

def update_task(task_id, new_task_name, new_task_description, new_task_status):
    cursor.execute('UPDATE tasks SET task_name=?, task_description=?,␣
 ↪task_status=? WHERE id=?',
                   (new_task_name, new_task_description, new_task_status,␣
 ↪task_id))
    conn.commit()

def delete_task(task_id):
    cursor.execute('DELETE FROM tasks WHERE id=?', (task_id,))
    conn.commit()
```

```
def main():
    while True:
        print("\nTask Management System")
        print("1. Add Task")
        print("2. List Tasks")
        print("3. Update Task")
        print("4. Delete Task")
        print("5. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            task_name = input("Enter the task name: ")
            task_description = input("Enter the task description: ")
            add_task(task_name, task_description)
            print("Task added successfully.")
        elif choice == '2':
            tasks = get_tasks()
            if tasks:
                for task in tasks:
                    print(f"ID: {task[0]}, Name: {task[1]}, Description:␣
 ↪{task[2]}, Status: {task[3]}")
            else:
                print("No tasks found in the database.")
        elif choice == '3':
            task_id = input("Enter the ID of the task to update: ")
            new_task_name = input("Enter the new task name: ")
            new_task_description = input("Enter the new task description: ")
            new_task_status = input("Enter the new task status: ")
            update_task(task_id, new_task_name, new_task_description,␣
 ↪new_task_status)
            print("Task updated successfully.")
        elif choice == '4':
            task_id = input("Enter the ID of the task to delete: ")
            delete_task(task_id)
            print("Task deleted successfully.")
        elif choice == '5':
            break
        else:
            print("Invalid choice. Please enter a valid option.")

if __name__ == "__main__":
    main()

# Close the connection when done
conn.close()
```

```
Task Management System
1. Add Task
2. List Tasks
3. Update Task
4. Delete Task
5. Exit
Enter your choice: 1
Enter the task name: ML Coding
Enter the task description: semi-supervised
Task added successfully.

Task Management System
1. Add Task
2. List Tasks
3. Update Task
4. Delete Task
5. Exit
Enter your choice: 1
Enter the task name: DL Coding
Enter the task description: lower layers may identify edges
Task added successfully.

Task Management System
1. Add Task
2. List Tasks
3. Update Task
4. Delete Task
5. Exit
Enter your choice: 1
Enter the task name: machine learning algorithms
Enter the task description: Linear regression; Logistic regression; Decision
tree; SVM algorithm; Naive Bayes algorithm; KNN algorithm; K-means; Random
forest algorithm
Task added successfully.

Task Management System
1. Add Task
2. List Tasks
3. Update Task
4. Delete Task
5. Exit
Enter your choice: 1
Enter the task name: X-AI
Enter the task description: Explainable Machine Learning (XML)
Task added successfully.

Task Management System
```

```
1. Add Task
2. List Tasks
3. Update Task
4. Delete Task
5. Exit
Enter your choice: 3
Enter the ID of the task to update: 3
Enter the new task name: machine learning algorithms
Enter the new task description: Linear regression; Logistic regression; Decision
tree; SVM algorithm; Naive Bayes algorithm; KNN algorithm; K-means
Enter the new task status: Emergency
Task updated successfully.

Task Management System
1. Add Task
2. List Tasks
3. Update Task
4. Delete Task
5. Exit
Enter your choice: 2
ID: 1, Name: In, Description: when luck suck , Status: Pending
ID: 2, Name: ML Coding, Description: semi-supervised, Status: Pending
ID: 3, Name: machine learning algorithms, Description: Linear regression;
Logistic regression; Decision tree; SVM algorithm; Naive Bayes algorithm; KNN
algorithm; K-means, Status: Emergency
ID: 4, Name: machine learning algorithms, Description: Linear regression;
Logistic regression; Decision tree; SVM algorithm; Naive Bayes algorithm; KNN
algorithm; K-means; Random forest algorithm, Status: Pending
ID: 5, Name: X-AI, Description: Explainable Machine Learning (XML) , Status:
Pending

Task Management System
1. Add Task
2. List Tasks
3. Update Task
4. Delete Task
5. Exit
Enter your choice: 5
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: