



PYTHON BOOTCAMP

www.jomhack.com

DATABASE (SQLITE)



sqlite:

- SQLite3 comes pre-installed with Python
- Serverless
- Single file database (self-contained)

Basic Commands:

- CREATE: create a table for database
- INSERT: create data
- SELECT: read data
- UPDATE: update data
- DELETE: delete data

DATABASE (SQLITE)



Initialize database (CREATE):

```
2 import sqlite3
3
4 class DatabaseManager:
5     def __init__(self, db_name='example.db'):
6         self.db_name = db_name
7         self.init_database()
8
9     def init_database(self):
10        """Initialize database with tables"""
11        with sqlite3.connect(self.db_name) as conn:
12            cursor = conn.cursor()
13
14            cursor.execute('''
15                CREATE TABLE IF NOT EXISTS users (
16                    id INTEGER PRIMARY KEY AUTOINCREMENT,
17                    name TEXT NOT NULL,
18                    email TEXT UNIQUE NOT NULL,
19                    age INTEGER,
20                    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
21                )
22            ''')
23
24            cursor.execute('''
25                CREATE TABLE IF NOT EXISTS posts (
26                    id INTEGER PRIMARY KEY AUTOINCREMENT,
27                    user_id INTEGER,
28                    title TEXT NOT NULL,
29                    content TEXT,
30                    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
31                    FOREIGN KEY (user_id) REFERENCES users (id)
32                )
33            ''')
```

DATABASE (SQLITE)



Create data (INSERT):

```
35     def create_user(self, name, email, age):
36         """Create a new user"""
37         try:
38             with sqlite3.connect(self.db_name) as conn:
39                 cursor = conn.cursor()
40                 cursor.execute('''
41                     INSERT INTO users (name, email, age)
42                     VALUES (?, ?, ?)
43                     ''', (name, email, age))
44                 return cursor.lastrowid
45         except sqlite3.IntegrityError as e:
46             print(f"Error: {e}")
47             return None
```

Create data (INSERT):

```
49     def create_post(self, user_id, title, content):
50         """Create a new post"""
51         with sqlite3.connect(self.db_name) as conn:
52             cursor = conn.cursor()
53             cursor.execute('''
54                 INSERT INTO posts (user_id, title, content)
55                 VALUES (?, ?, ?)
56                 ''', (user_id, title, content))
57             return cursor.lastrowid
```

DATABASE (SQLITE)



Read data (SELECT):

```
59 def get_all_users(self):
60     """Get all users"""
61     with sqlite3.connect(self.db_name) as conn:
62         cursor = conn.cursor()
63         cursor.execute('SELECT * FROM users')
64         return cursor.fetchall()
```

Read data (SELECT):

```
66 def get_user_posts(self, user_id):
67     """Get posts by user"""
68     with sqlite3.connect(self.db_name) as conn:
69         cursor = conn.cursor()
70         cursor.execute('''
71             SELECT p.id, p.title, p.content, p.created_at
72             FROM posts p
73             WHERE p.user_id = ?
74             ORDER BY p.created_at DESC
75             ''', (user_id,))
76         return cursor.fetchall()
```

DATABASE (SQLITE)



Delete data (DELETE):

```
78     def delete_user(self, user_id):
79         """Delete user and their posts"""
80         with sqlite3.connect(self.db_name) as conn:
81             cursor = conn.cursor()
82             cursor.execute('DELETE FROM posts WHERE user_id = ?', (user_id,))
83             cursor.execute('DELETE FROM users WHERE id = ?', (user_id,))
84             return cursor.rowcount > 0
```

DATABASE (SQLITE)



Run on Terminal Function:

```
86 def display_menu():
87     """Display the main menu"""
88     print("\n" + "="*40)
89     print("          DATABASE MANAGER")
90     print("="*40)
91     print("1. Create User")
92     print("2. View All Users")
93     print("3. Create Post")
94     print("4. View User Posts")
95     print("5. Delete User")
96     print("6. Exit")
97     print("-"*40)
```

```
99 def main():
100     """Main interactive CLI function"""
101     db = DatabaseManager()
102
103     while True:
104         display_menu()
105         choice = input("Enter your choice (1-6): ").strip()
106
107         if choice == '1':
108             print("\n--- Create New User ---")
109             name = input("Enter name: ").strip()
110             email = input("Enter email: ").strip()
111             try:
112                 age = int(input("Enter age: ").strip())
113                 user_id = db.create_user(name, email, age)
114                 if user_id:
115                     print(f"✓ User created successfully! ID: {user_id}")
116                 else:
117                     print("X Failed to create user")
118             except ValueError:
119                 print("X Invalid age. Please enter a number.")
120
121         elif choice == '2':
122             print("\n--- All Users ---")
123             users = db.get_all_users()
124             if users:
125                 for user in users:
126                     print(f"ID: {user[0]} | Name: {user[1]} | Email: {user[2]} | Age: {user[3]}")
127             else:
128                 print("No users found.")
```

DATABASE (SQLITE)



Run on Terminal Function:

```
130 elif choice == '3':
131     print("\n--- Create New Post ---")
132     try:
133         user_id = int(input("Enter user ID: ").strip())
134         title = input("Enter post title: ").strip()
135         content = input("Enter post content: ").strip()
136         post_id = db.create_post(user_id, title, content)
137         if post_id:
138             print(f"✓ Post created successfully! ID: {post_id}")
139         else:
140             print("X Failed to create post")
141     except ValueError:
142         print("X Invalid user ID. Please enter a number.")
143
144 elif choice == '4':
145     print("\n--- View User Posts ---")
146     try:
147         user_id = int(input("Enter user ID: ").strip())
148         posts = db.get_user_posts(user_id)
149         if posts:
150             for post in posts:
151                 print(f"\nPost ID: {post[0]}")
152                 print(f>Title: {post[1]}")
153                 print(f>Content: {post[2]}")
154                 print(f>Created: {post[3]}")
155                 print("-" * 30)
156             else:
157                 print("No posts found for this user.")
158     except ValueError:
159         print("X Invalid user ID. Please enter a number.")
```

```
161 elif choice == '5':
162     print("\n--- Delete User ---")
163     try:
164         user_id = int(input("Enter user ID to delete: ").strip())
165         confirm = input(f"Are you sure you want to delete user {user_id}? (y/N): ").strip().lower()
166         if confirm == 'y':
167             if db.delete_user(user_id):
168                 print("✓ User deleted successfully!")
169             else:
170                 print("X User not found or deletion failed.")
171         else:
172             print("Deletion cancelled.")
173     except ValueError:
174         print("X Invalid user ID. Please enter a number.")
175
176 elif choice == '6':
177     print("\nGoodbye! 🙋")
178     break
179
180 else:
181     print("X Invalid choice. Please enter 1-6.")
182
183     input("\nPress Enter to continue...")
184
185 if __name__ == "__main__":
186     main()
```