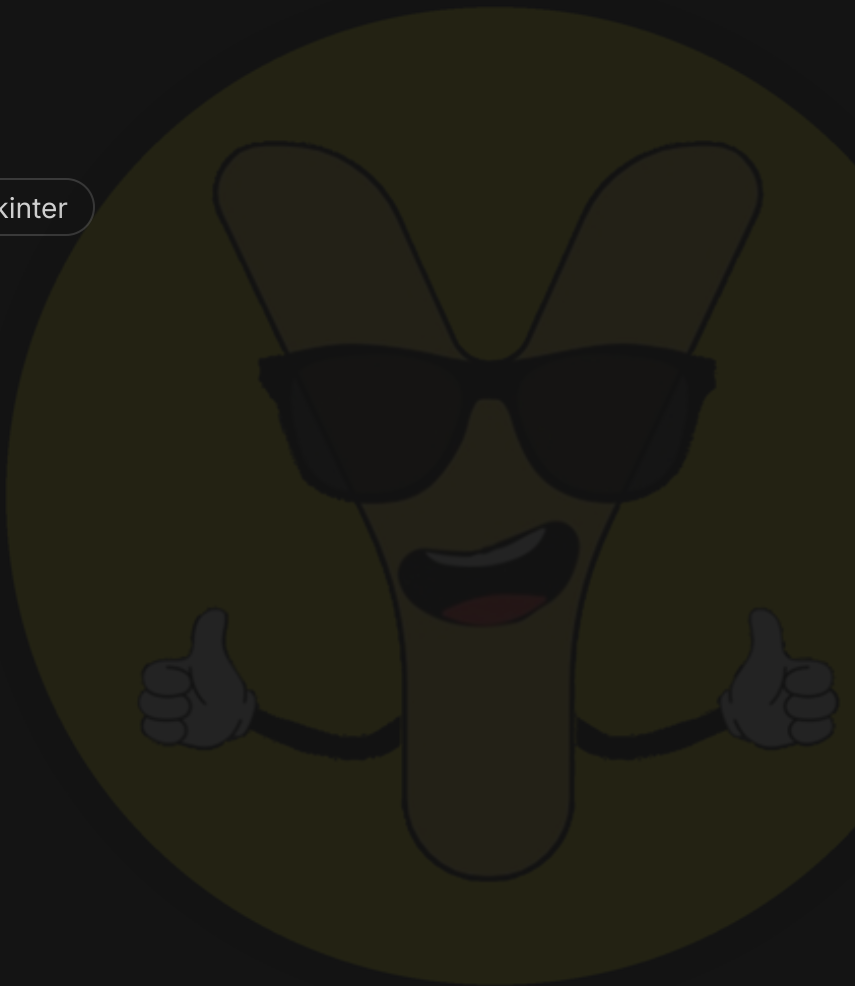
AKTU · 2ND YEAR · BCC302 / BCC402

Python Programming

Unit 4 — GUI Programming and Databases

- Introduction to GUI programming
- Tkinter programming: widgets, geometry managers
- Event handling in Tkinter
- Database connectivity with Python
- SQLite database operations: connect, execute, commit, close
- CRUD operations using Python and SQLite



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Introduction to GUI Programming

Graphical User Interface (GUI) allows users to interact with software through visual elements rather than text commands.

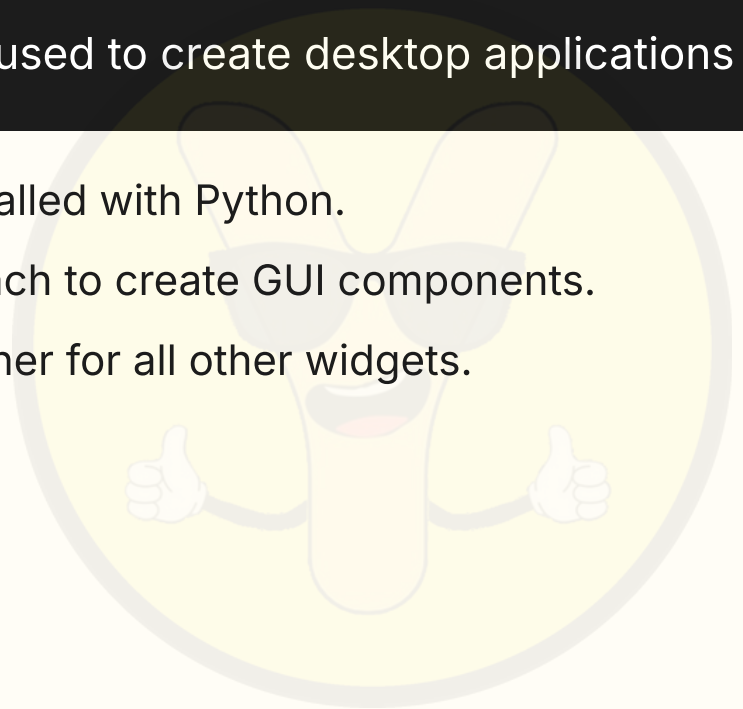
- GUI provides a user-friendly environment using icons, menus, and windows.
- It relies on an event-driven programming model.
- Python supports multiple GUI toolkits like Tkinter, PyQt, and Kivy.
- Tkinter is the standard Python interface to the Tk GUI toolkit.
- Advantages: Intuitive, reduces user training, improves productivity.
- Disadvantages: Higher memory consumption, complex development compared to CLI.

Tkinter Programming

TKINTER

Tkinter is a built-in Python library used to create desktop applications with a cross-platform GUI.

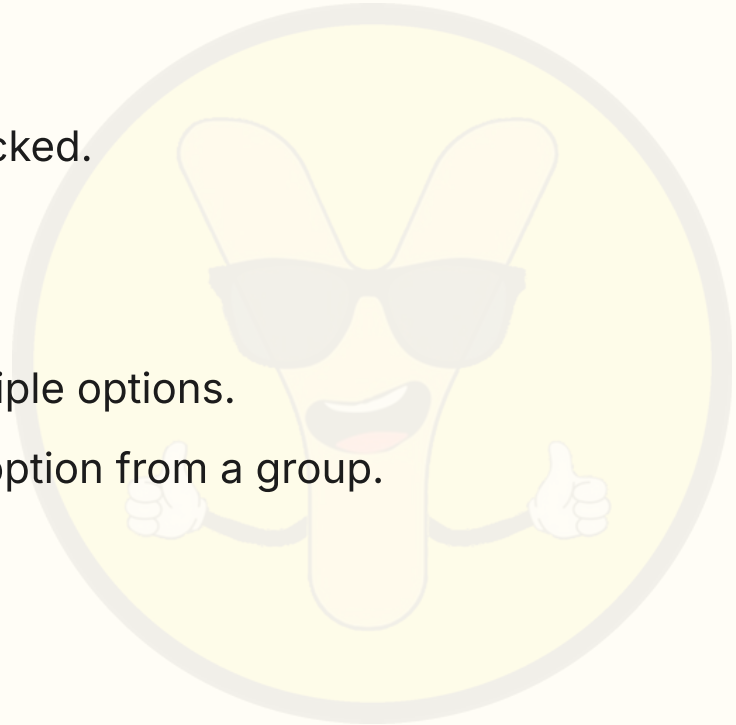
- It is lightweight and comes pre-installed with Python.
- It follows an object-oriented approach to create GUI components.
- The root window is the main container for all other widgets.



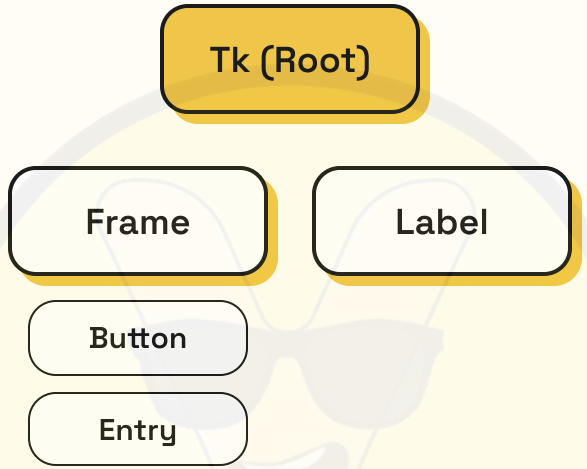
Tkinter Widgets

Widgets are the building blocks of a Tkinter application.

- Label: Displays text or images.
- Button: Triggers an action when clicked.
- Entry: Single-line text input field.
- Text: Multi-line text input field.
- Checkbutton: Allows selecting multiple options.
- Radiobutton: Allows selecting one option from a group.



Tkinter Widget Hierarchy

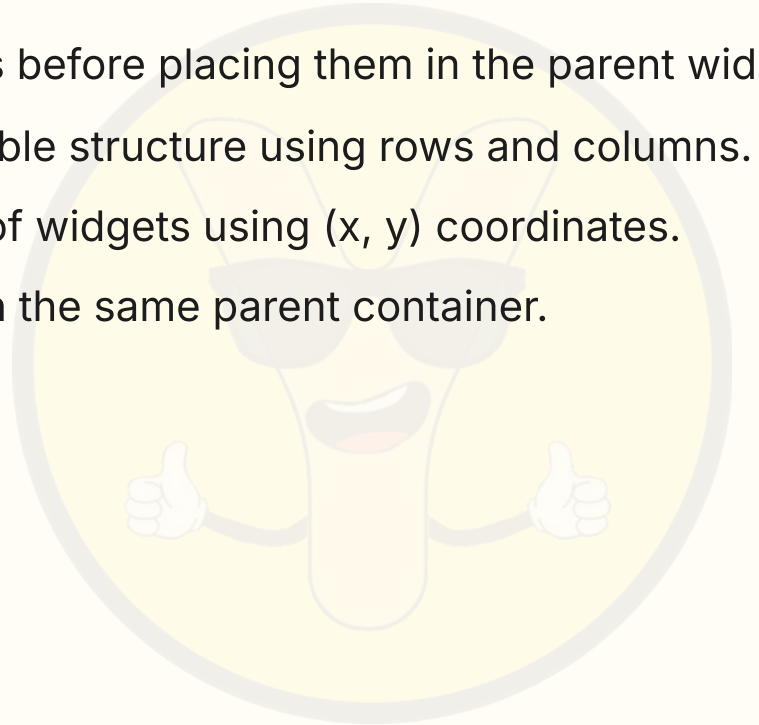


The hierarchical structure of a basic Tkinter GUI application.

Geometry Managers

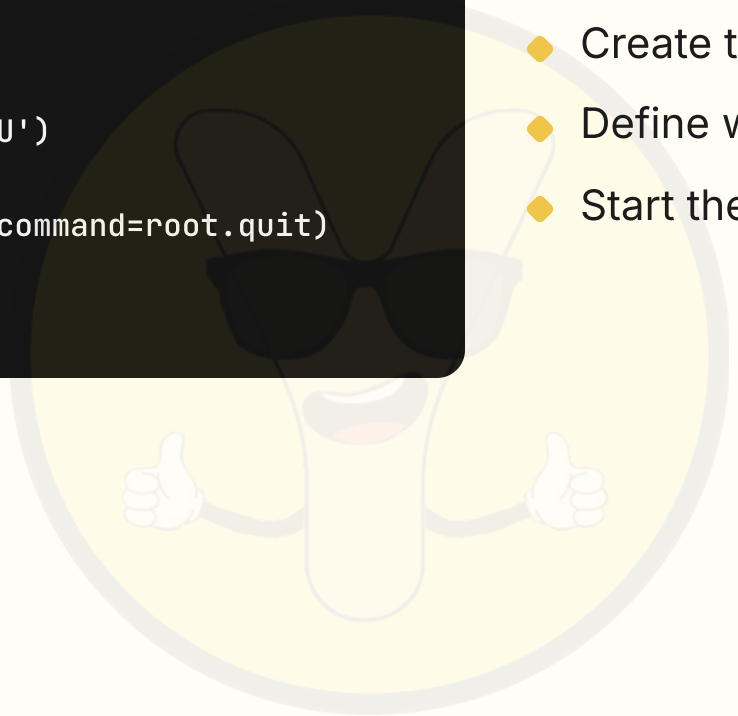
Geometry managers control the placement and size of widgets within the parent container.

- pack(): Organizes widgets in blocks before placing them in the parent widget.
- grid(): Organizes widgets in a 2D table structure using rows and columns.
- place(): Allows explicit positioning of widgets using (x, y) coordinates.
- Note: Never mix pack() and grid() in the same parent container.



Tkinter Basic Example

```
import tkinter as tk
root = tk.Tk()
root.title('My App')
label = tk.Label(root, text='Hello AKTU')
label.pack()
button = tk.Button(root, text='Exit', command=root.quit)
button.pack()
root.mainloop()
```

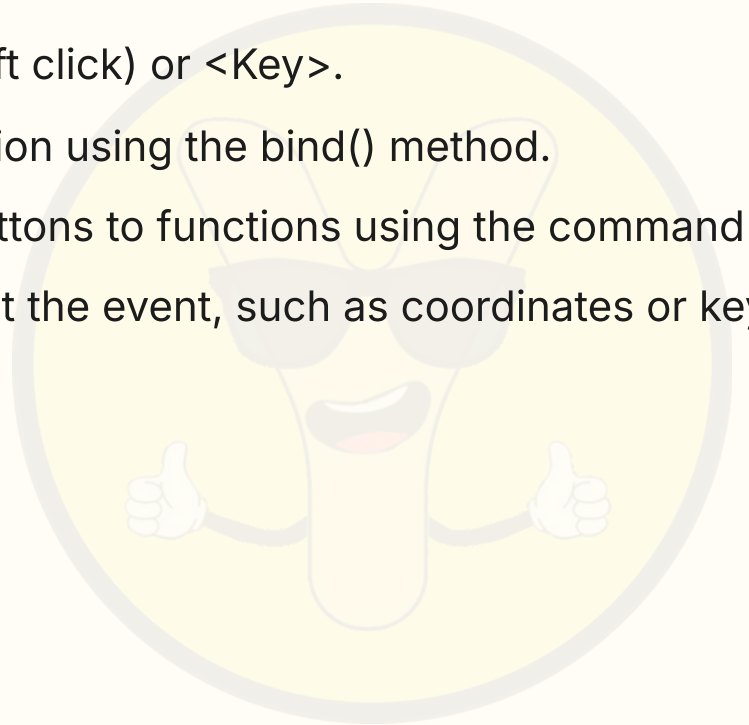


- Import tkinter module.
- Create the main root window.
- Define widgets and use a geometry manager.
- Start the event loop using mainloop().

Event Handling in Tkinter

Event-driven programming means the application waits for user actions like clicks or key presses.

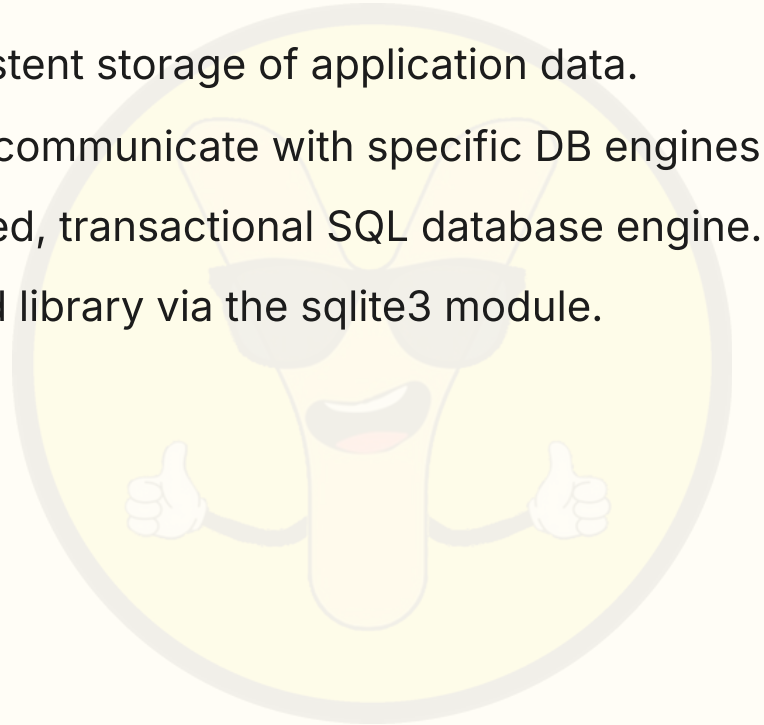
- Event: An action like <Button-1> (left click) or <Key>.
- Binding: Linking an event to a function using the bind() method.
- Command: A simpler way to link buttons to functions using the command parameter.
- Event Object: Contains details about the event, such as coordinates or keys pressed.



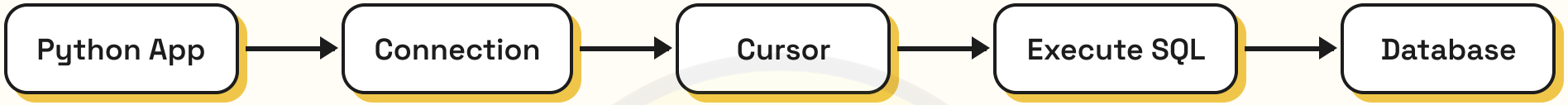
Database Connectivity with Python

Python interacts with databases using the Python Database API Specification v2.0.

- Database connectivity allows persistent storage of application data.
- Python uses drivers/connectors to communicate with specific DB engines.
- SQLite is a serverless, self-contained, transactional SQL database engine.
- It is included in the Python standard library via the sqlite3 module.



Database Connectivity Flow

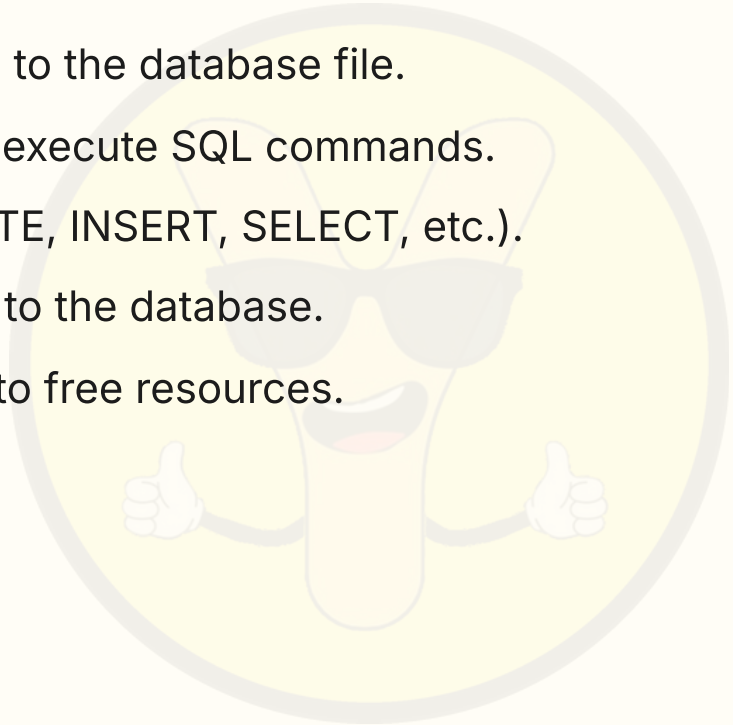


Standard workflow for interacting with a database in Python.

SQLite Database Operations

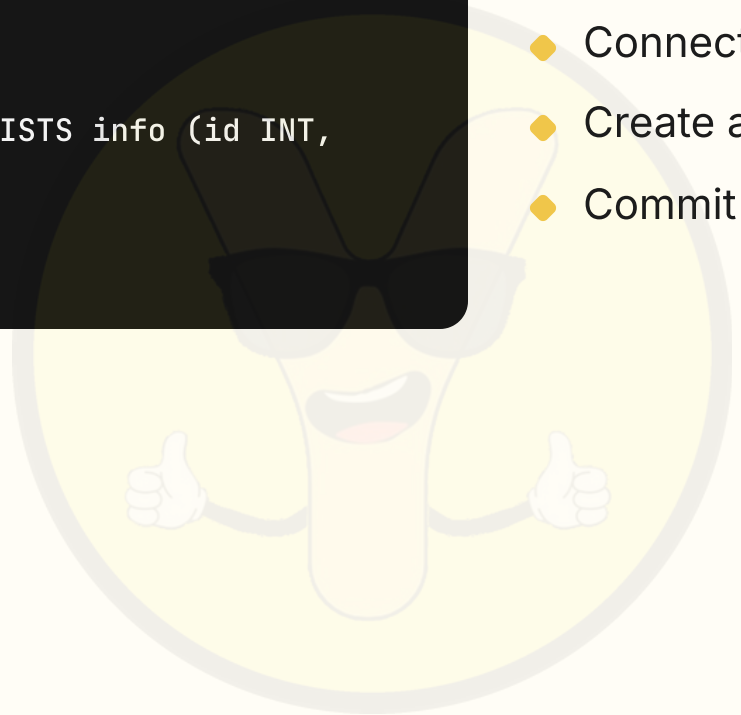
Basic steps to perform operations in SQLite.

- connect(): Establishes a connection to the database file.
- cursor(): Creates a cursor object to execute SQL commands.
- execute(): Runs SQL queries (CREATE, INSERT, SELECT, etc.).
- commit(): Saves the changes made to the database.
- close(): Terminates the connection to free resources.



SQLite Connection Example

```
import sqlite3
conn = sqlite3.connect('student.db')
cursor = conn.cursor()
cursor.execute('CREATE TABLE IF NOT EXISTS info (id INT,
name TEXT)')
conn.commit()
conn.close()
```

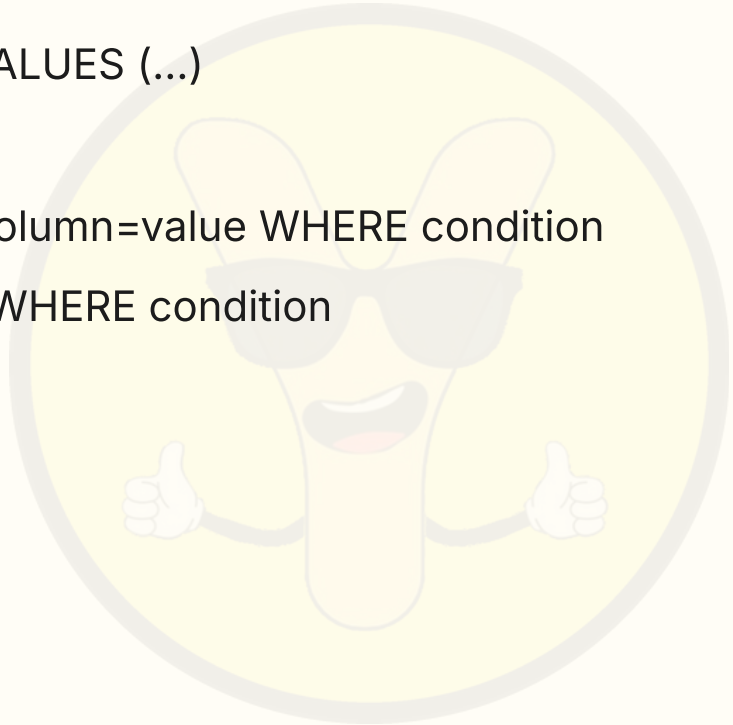


- Import sqlite3 module.
- Connect to the database file.
- Create a table if it does not exist.
- Commit changes and close the connection.

CRUD Operations

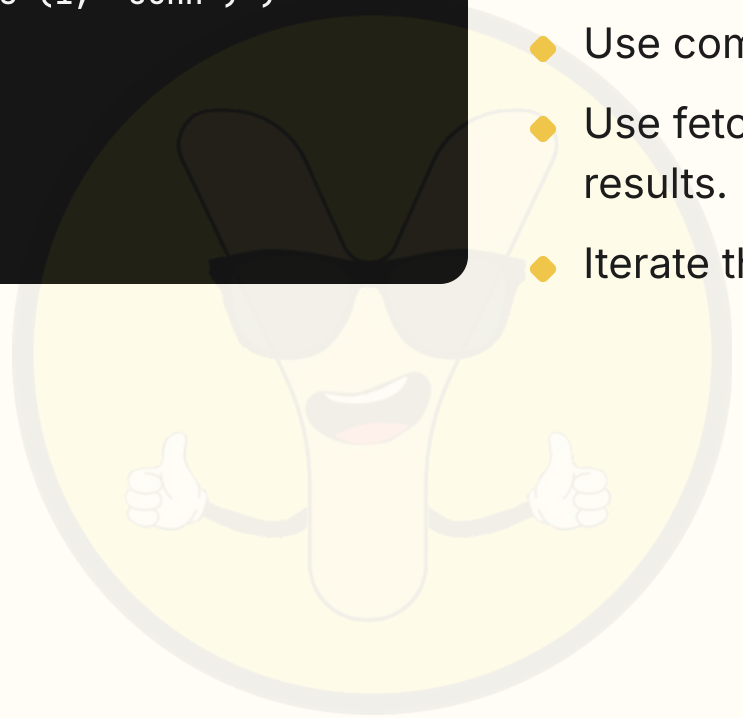
CRUD stands for Create, Read, Update, and Delete, the four basic functions of persistent storage.

- Create: INSERT INTO table_name VALUES (...)
- Read: SELECT * FROM table_name
- Update: UPDATE table_name SET column=value WHERE condition
- Delete: DELETE FROM table_name WHERE condition



CRUD: Insert and Select

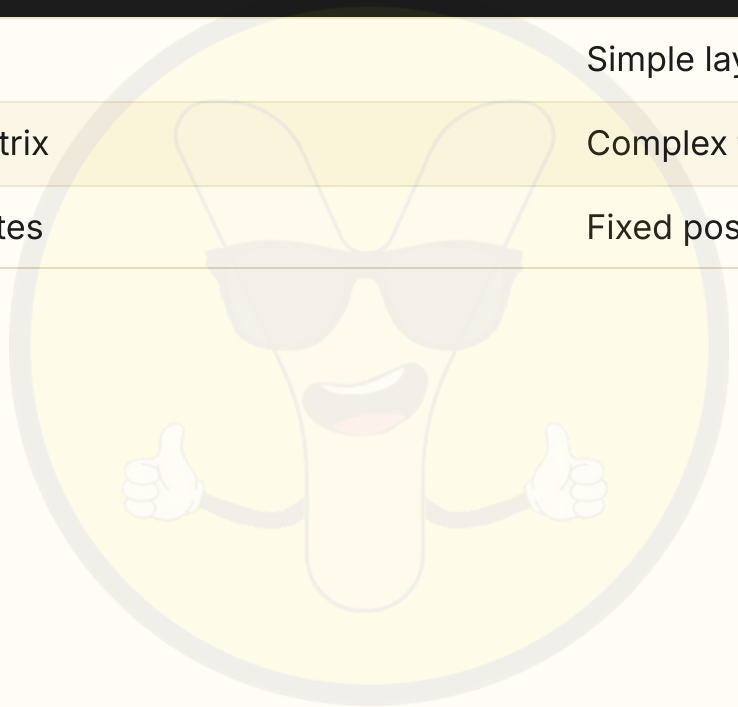
```
cursor.execute('INSERT INTO info VALUES (1, "John")')
conn.commit()
cursor.execute('SELECT * FROM info')
rows = cursor.fetchall()
for row in rows:
    print(row)
```



- Use execute() for INSERT operations.
- Use commit() to persist data.
- Use fetchall() or fetchone() to retrieve query results.
- Iterate through results to process data.

Comparison of Geometry Managers

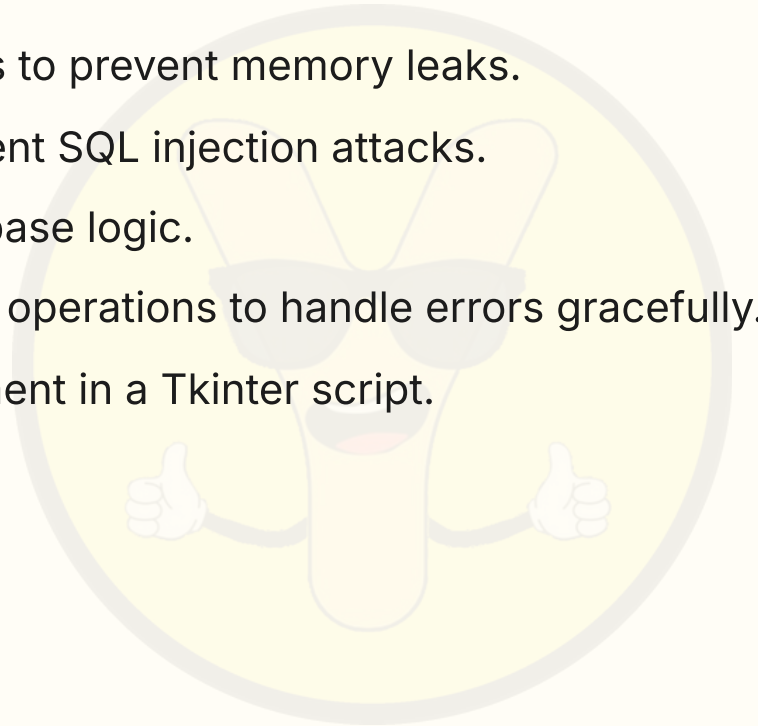
Manager	Mechanism	Best Use Case
pack	Stacking	Simple layouts
grid	Table/Matrix	Complex forms
place	Coordinates	Fixed positioning



Best Practices

Guidelines for efficient GUI and Database programming.

- Always close database connections to prevent memory leaks.
- Use parameterized queries to prevent SQL injection attacks.
- Keep GUI logic separate from database logic.
- Use try-except blocks for database operations to handle errors gracefully.
- Ensure mainloop() is the last statement in a Tkinter script.



Exam corner

- **Q1** Explain the role of geometry managers in Tkinter with examples.
- **Q2** What is event-driven programming? How does Tkinter handle events?
- **Q3** Differentiate between pack(), grid(), and place() geometry managers.
- **Q4** Write a Python program to create a GUI with a button that saves user input to an SQLite database.
- **Q5** Explain the steps involved in connecting Python to an SQLite database.
- **Q6** Describe the CRUD operations in SQL and how they are implemented in Python.
- **Q7** What is the purpose of commit() and close() methods in database connectivity?

Unit 4 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 5 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

