

# Python Programming

## Unit 04 – Lecture 04: Tkinter + Databases (Concepts and Integration)

Tofik Ali

School of Computer Science, UPES Dehradun

February 14, 2026

Repository: <https://github.com/tali7c/Python-Programming>

Core Concepts  
ooooo

Demo  
o

Interactive  
ooo

Summary  
oo

# Quick Links

Core Concepts

Demo

Interactive

Summary

# Agenda

1 Core Concepts

2 Demo

3 Interactive

4 Summary

# Learning Outcomes

- Explain why apps use databases instead of plain files

# Learning Outcomes

- Explain why apps use databases instead of plain files
- Compare relational (SQL) and NoSQL databases

# Learning Outcomes

- Explain why apps use databases instead of plain files
- Compare relational (SQL) and NoSQL databases
- Describe a basic GUI-to-database workflow

# Learning Outcomes

- Explain why apps use databases instead of plain files
- Compare relational (SQL) and NoSQL databases
- Describe a basic GUI-to-database workflow
- Build a small Tkinter app that stores records in SQLite

# Why Databases?

- Data persistence across runs

# Why Databases?

- Data persistence across runs
- Fast search and structured storage

# Why Databases?

- Data persistence across runs
- Fast search and structured storage
- Constraints (unique IDs), integrity, transactions

# Why Databases?

- Data persistence across runs
- Fast search and structured storage
- Constraints (unique IDs), integrity, transactions
- Multi-user / multi-program access (in bigger systems)

# Relational vs NoSQL (Quick Comparison)

---

Relational (SQL)	NoSQL
Tables (rows/columns)	Documents / key-value / graphs
Fixed schema	Flexible schema
SQL queries	Query APIs (varies)
Example: SQLite, MySQL	Example: MongoDB

---

# SQLite for Learning

- SQLite is embedded (no server)

# SQLite for Learning

- SQLite is embedded (no server)
- Uses one file as the database

# SQLite for Learning

- SQLite is embedded (no server)
- Uses one file as the database
- Great for demos and small apps

# SQLite for Learning

- SQLite is embedded (no server)
- Uses one file as the database
- Great for demos and small apps
- Python has built-in `sqlite3` module

# GUI → Database Flow

- User enters values in GUI

# GUI → Database Flow

- User enters values in GUI
- App validates input

# GUI → Database Flow

- User enters values in GUI
- App validates input
- App runs SQL query (INSERT/SELECT/UPDATE/DELETE)

# GUI → Database Flow

- User enters values in GUI
- App validates input
- App runs SQL query (INSERT/SELECT/UPDATE/DELETE)
- App shows updated view (Listbox/Table)

# Demo: Tkinter + SQLite Student Records

- File: demo/tkinter\_sqlite\_student\_app.py

# Demo: Tkinter + SQLite Student Records

- File: `demo/tkinter_sqlite_student_app.py`
- Features:

# Demo: Tkinter + SQLite Student Records

- File: `demo/tkinter_sqlite_student_app.py`
- Features:
  - create table if not exists

# Demo: Tkinter + SQLite Student Records

- File: `demo/tkinter_sqlite_student_app.py`
- Features:
  - create table if not exists
  - add a student record

# Demo: Tkinter + SQLite Student Records

- File: `demo/tkinter_sqlite_student_app.py`
- Features:
  - create table if not exists
  - add a student record
  - display all records

# Checkpoint 1

**Question:** Give one advantage of databases over plain text files.

## Checkpoint 2

**Question:** In a relational database, what is a **primary key** and why is it useful?

# Think-Pair-Share

Discuss:

- For a “Todo App”, would you use a file or a database? Why?

# Key Takeaways

- Databases provide structured storage, querying, and integrity

# Key Takeaways

- Databases provide structured storage, querying, and integrity
- Relational DBs use tables and SQL; NoSQL offers flexible models

# Key Takeaways

- Databases provide structured storage, querying, and integrity
- Relational DBs use tables and SQL; NoSQL offers flexible models
- SQLite is a practical starting point using Python's `sqlite3`

# Key Takeaways

- Databases provide structured storage, querying, and integrity
- Relational DBs use tables and SQL; NoSQL offers flexible models
- SQLite is a practical starting point using Python's `sqlite3`
- GUI apps typically follow input → validate → query → refresh view

# Exit Question

Write one SQL command to create a table `students(id, name)` where `id` is a primary key.