Python Programming
Unit 04 – Lecture 06 Notes
Transactions and Database Error Handling

Tofik Ali

February 14, 2026

Contents

1 Lecture Overview

Databases must remain consistent even when errors happen. Transactions solve this by providing **all-or-nothing** behavior: either every operation succeeds or none is applied.

2 Core Concepts

2.1 What is a Transaction?

A transaction groups multiple SQL operations into one unit of work.

Example: transferring money requires two updates:

- subtract from account A,
- add to account B.

If the second update fails, the first update must be undone. That is rollback.

2.2 Commit and Rollback

- `commit()` makes changes permanent.
- `rollback()` undoes changes since the last commit.

2.3 Safe Pattern in Python

```python
import sqlite3

con = sqlite3.connect("app.db")
try:
    con.execute("INSERT ...")
    con.execute("UPDATE ...")
    con.commit()
except sqlite3.Error as e:
    con.rollback()
    print("DB Error:", e)
finally:
    con.close()
```

2.4 Common Database Exceptions

- `sqlite3.IntegrityError`: constraint failed (UNIQUE, NOT NULL, etc.)
- `sqlite3.OperationalError`: bad SQL, missing table, locked database
- `sqlite3.ProgrammingError`: wrong API usage (closed cursor, wrong parameters)
- `sqlite3.Error`: parent class for SQLite exceptions

2.5 Context Manager Shortcut

SQLite connections support context managers:

```python
import sqlite3

with sqlite3.connect("app.db") as con:
    con.execute("INSERT ...")
    # commit happens automatically if no exception
    # rollback happens automatically on exception
```

This is a clean and safe pattern for many scripts.

3 Demo Walkthrough

File: `demo/transactions_rollback_demo.py`

The demo:

- creates a table with a UNIQUE constraint,
- attempts two inserts in one transaction,

- the second insert violates the constraint,
- rollback ensures no partial insertion remains.

4 Interactive Checkpoints (with Solutions)

Checkpoint 1 Solution

Question: What problem happens without transactions?
Answer: partial updates can remain in the database, causing inconsistent data.

Checkpoint 2 Solution

Question: Which exception for duplicate unique value?
Answer: `IntegrityError`.

5 Practice Exercises (with Solutions)

Exercise 1: Add UNIQUE Constraint

Task: Create a table where `sapid` must be unique.
Solution:

```
CREATE TABLE students (
  id INTEGER PRIMARY KEY,
  sapid TEXT UNIQUE,
  name TEXT
);
```

Exercise 2: Rollback on Failure

Task: Insert two rows; if any insert fails, rollback.
Solution (pattern):

```
try:
    con.execute("INSERT ...", (...,))
    con.execute("INSERT ...", (...,))
    con.commit()
except sqlite3.Error:
    con.rollback()
```

6 Exit Question (with Solution)

Question: What does `rollback()` do?
Answer: It cancels all uncommitted changes and restores the database to the last committed state.