ASSIGNMENT-16.1

NAME: ELAKANTI SRINADH

HALLTICKET NUMBER:2403A52424

BATCH:15

SUBJECT:AI CODING

# TASK-1

**QUESTION:**

1 – Student Information System Schema Task:

Design a database schema for a Student Information System and generate queries using AI.

Instructions:

* Tables: Students, Courses, Enrollments.
* Define primary keys, foreign keys, and relationships.
* Generate queries:

o Insert a new student record. o Fetch all courses enrolled by a student. o Count number of students in each course.

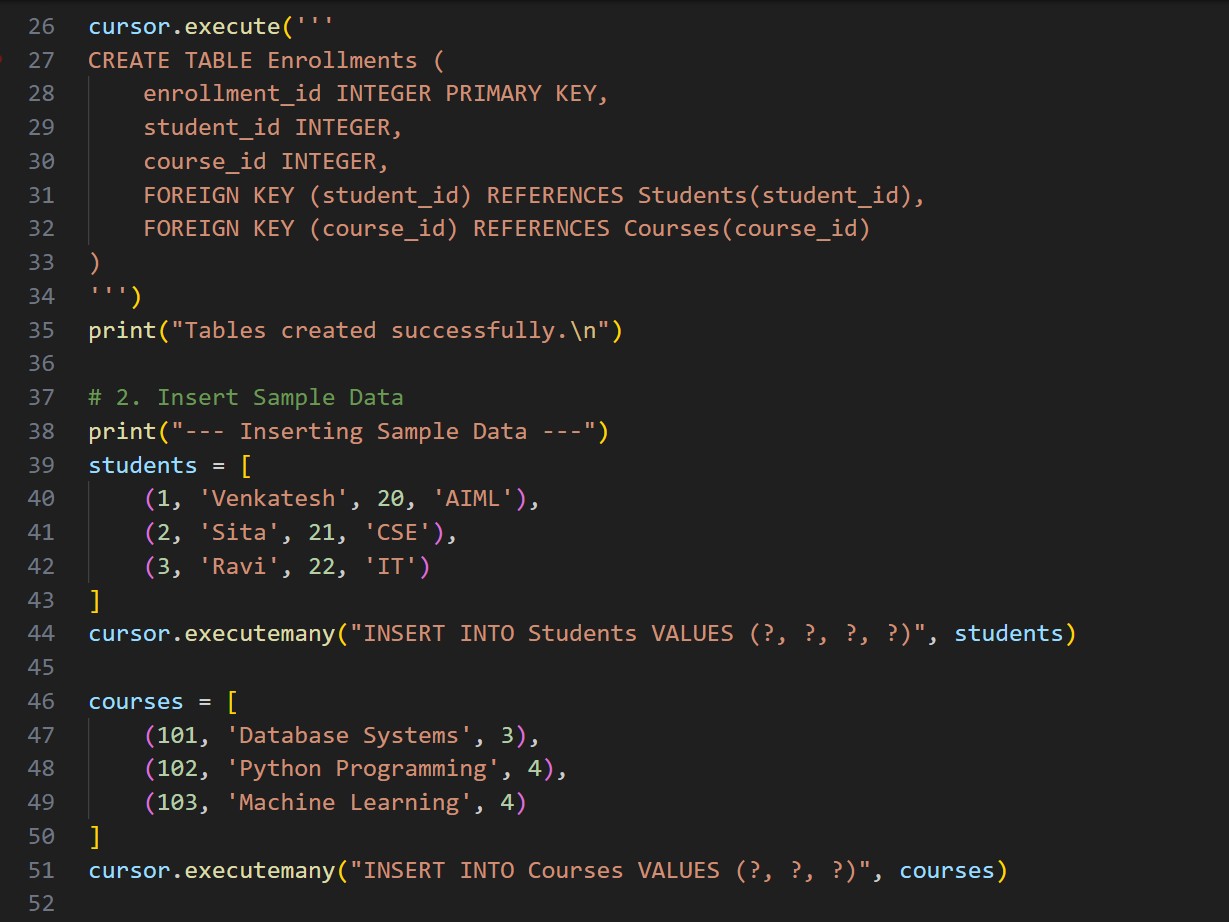
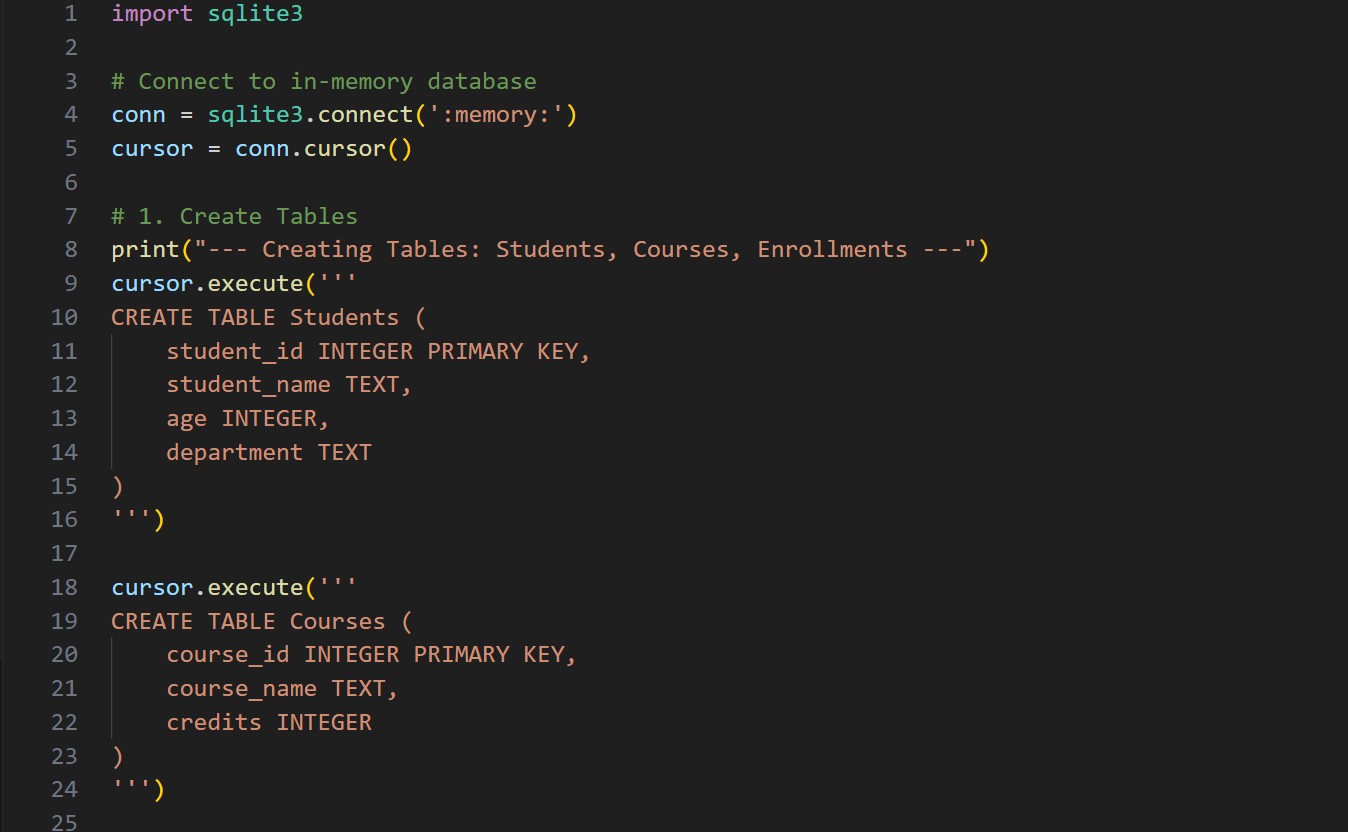
Expected Output:

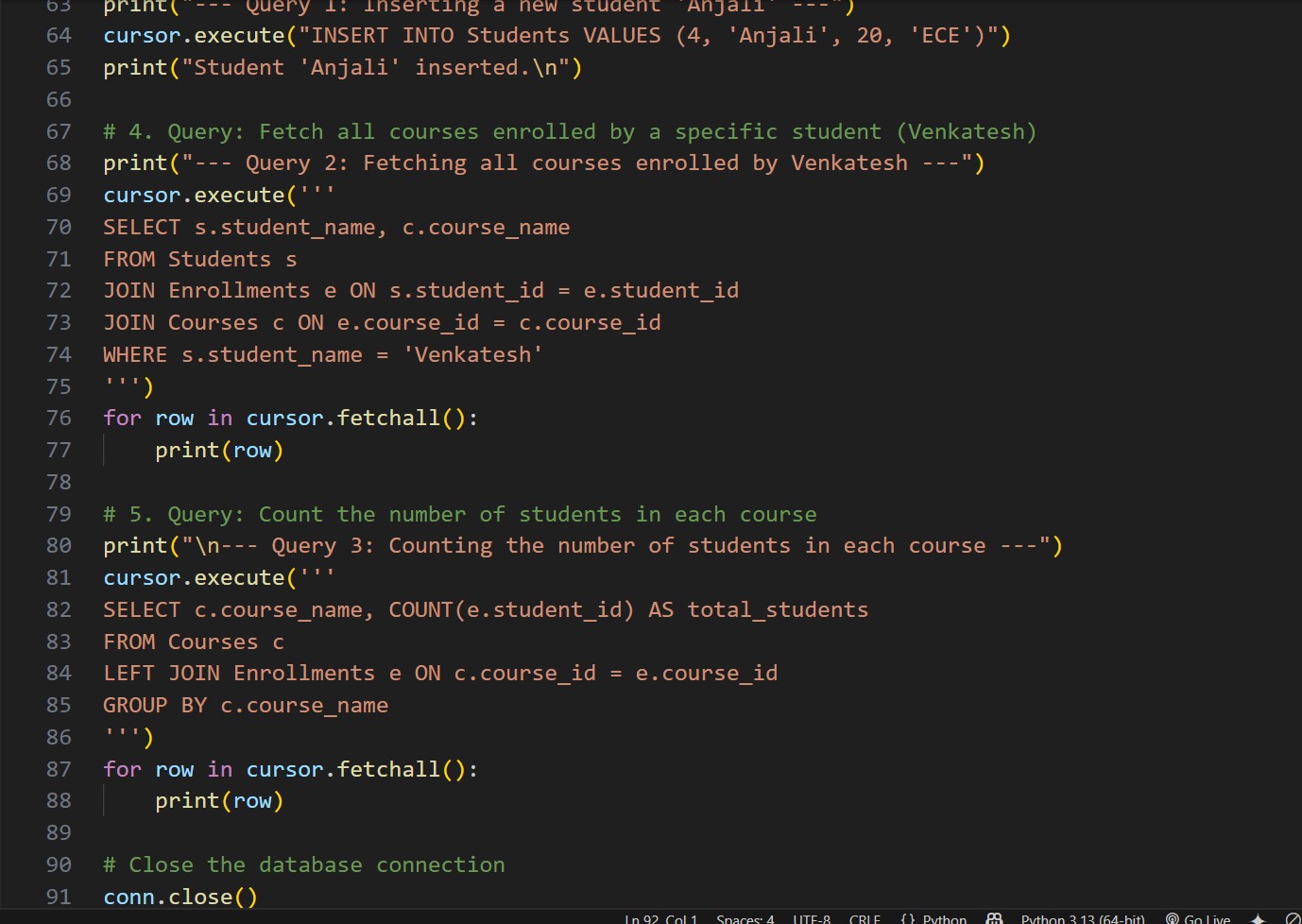
• SQL CREATE TABLE statements and queries for student–course relationships

**PROMPT:**

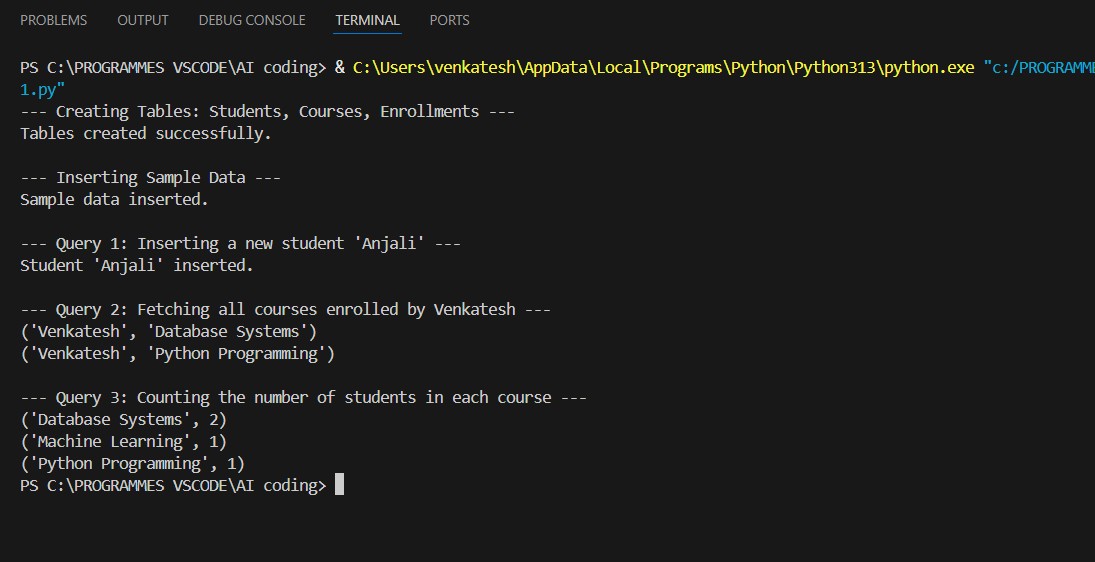
Use AI to design a database schema for a Student Information System with three tables — Students, Courses, and Enrollment . Define the primary and foreign keys. Generate SQL queries to insert a student, fetch all courses enrolled by a student, and count the number of students in each course

**CODE:**





**OUTPUT:**



**OBSERVATION:**

AI successfully generated the database schema with relationships between the tables using primary and foreign keys. It also provided correct SQL queries for inserting a student record, retrieving courses enrolled by a student, and counting students in each course. The output was accurate and easy to understand.

# TASK-2

**QUESTION:**

2 – Hospital Management Database Task:

Create schema and queries for a Hospital Management System.

Instructions:

* Tables: Doctors, Patients, Appointments.
* Use AI to define constraints (unique IDs, valid dates).
* Generate queries:
* List all appointments for a specific doctor.
* Retrieve patient history by patient ID. o Count total patients treated by each doctor.

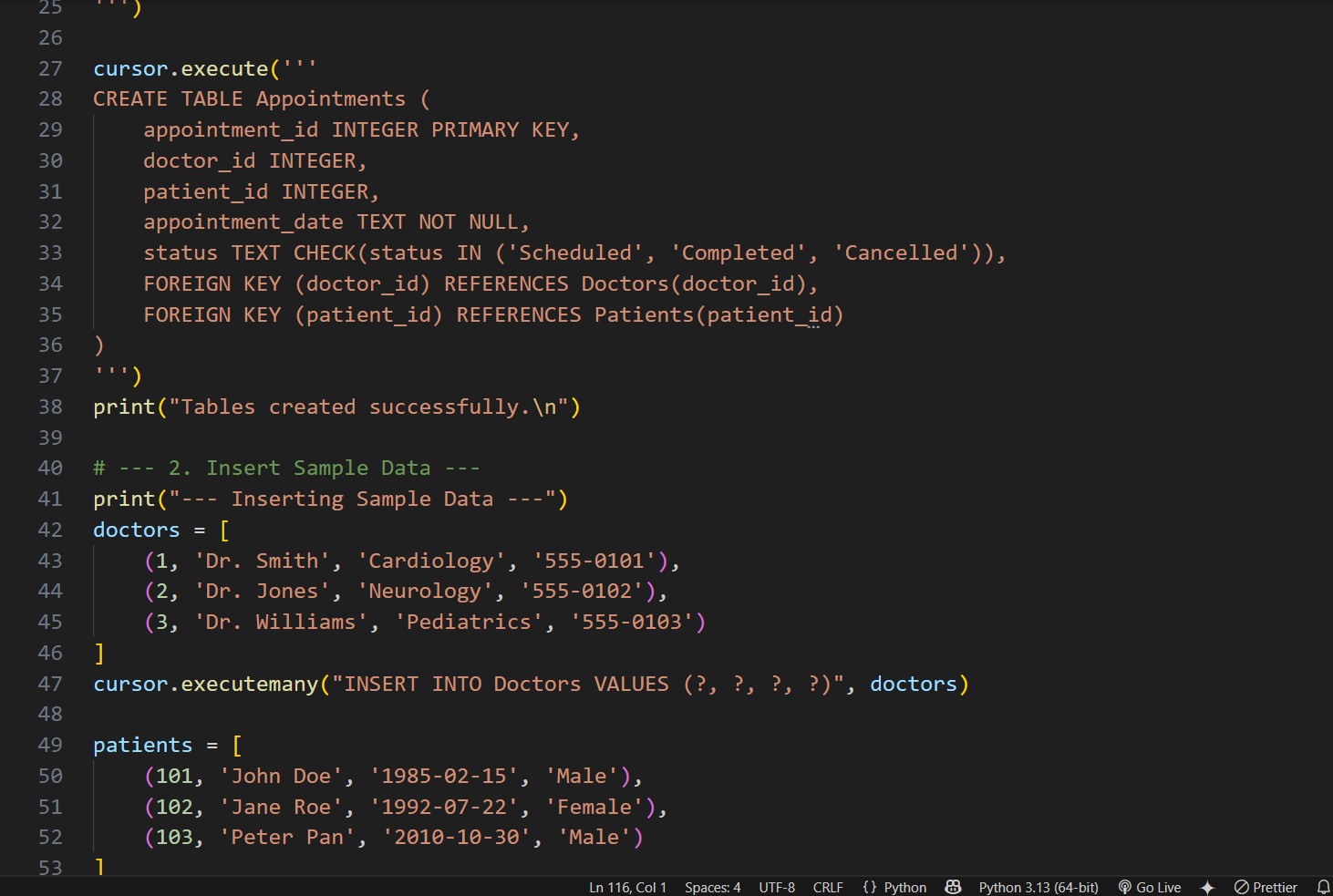
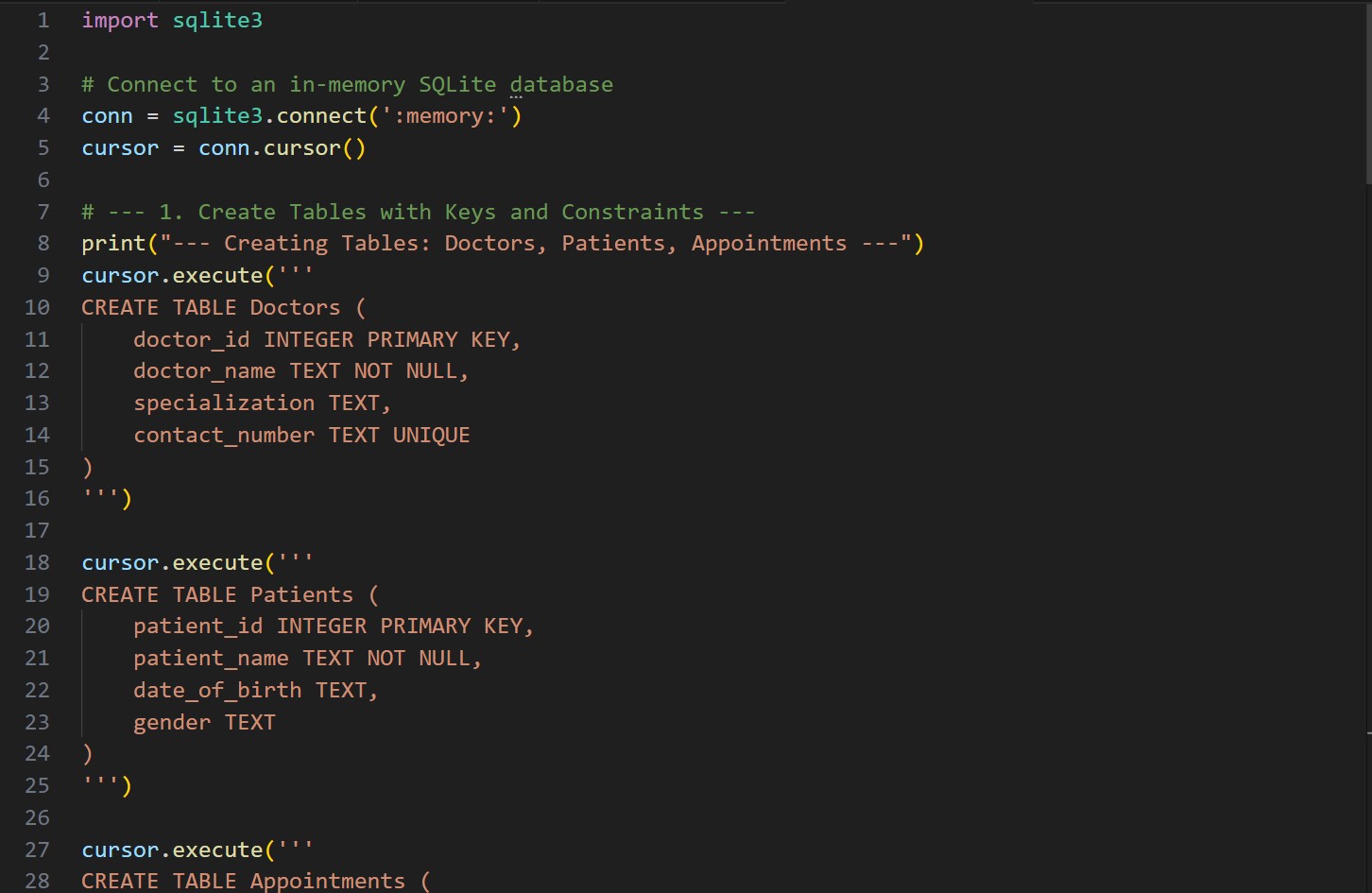
Expected Output:

• Normalized schema and SQL queries with joins

**PROMPT:**

Use AI to create a Hospital Management System database with three tables — Doctors, Patients, and Appointments. Define primary keys, foreign keys, and constraints for valid data. Generate SQL queries to list all appointments for a doctor, get patient history by ID, and count total patients treated by each doctor.

**CODE:**



**OUTPUT:**



**OBSERVATION:**

AI generated a well-structured database schema with relationships between doctors, patients, and appointments. It used primary and foreign keys correctly and included constraints for data validity. The queries for listing appointments, viewing patient history, and counting patients per doctor were accurate and easy to understand.

# TASK-3

**QUESTION:**

3 – Library Database Task:

Design schema for a Library Management System.

Instructions:

* Tables: Books, Members, Loans.
* Use AI to suggest indexing strategy for faster queries.
* Generate queries:

o Retrieve all books currently issued. o Find overdue books (loan date > 30 days). o Count number of books loaned by each member.

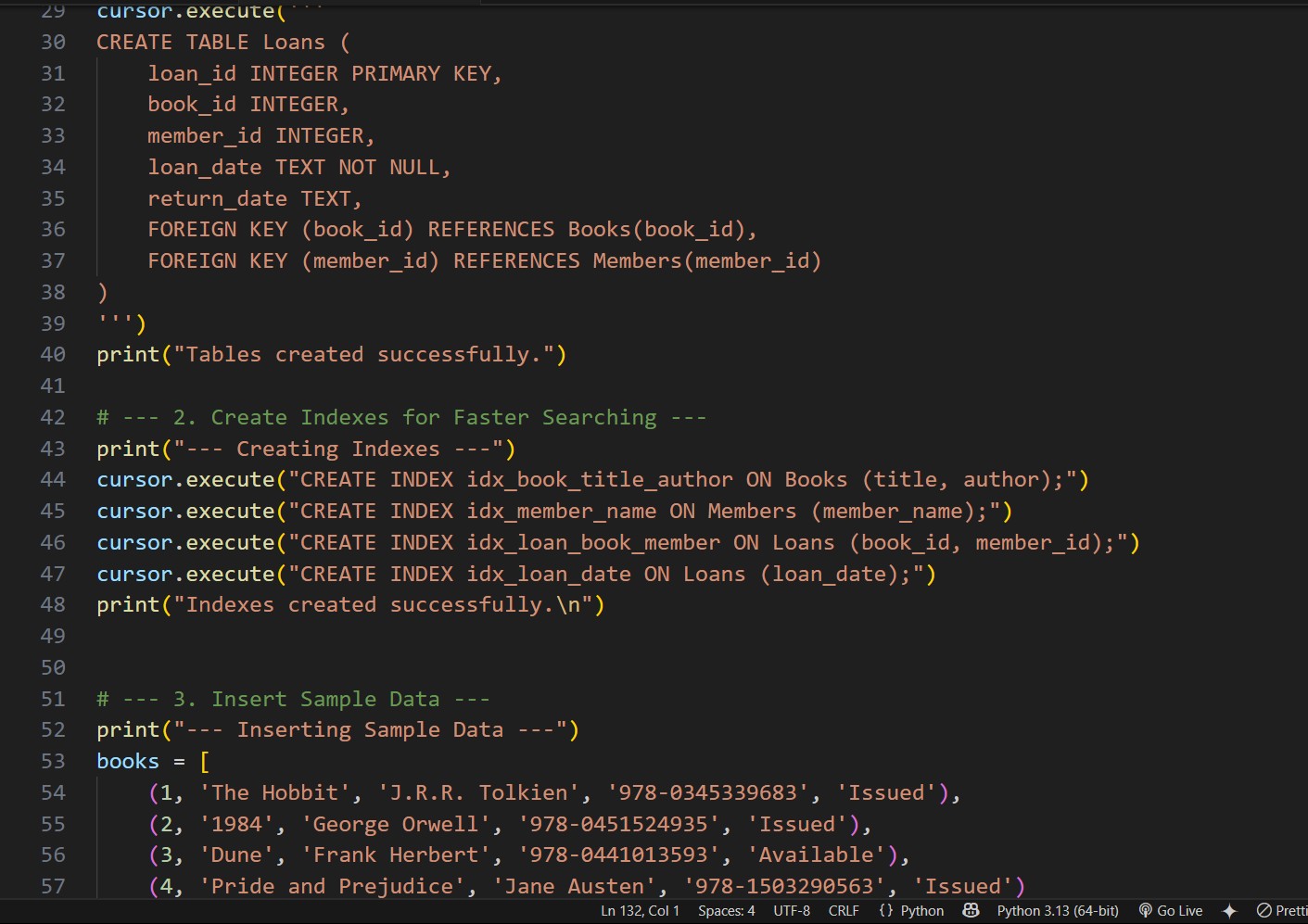
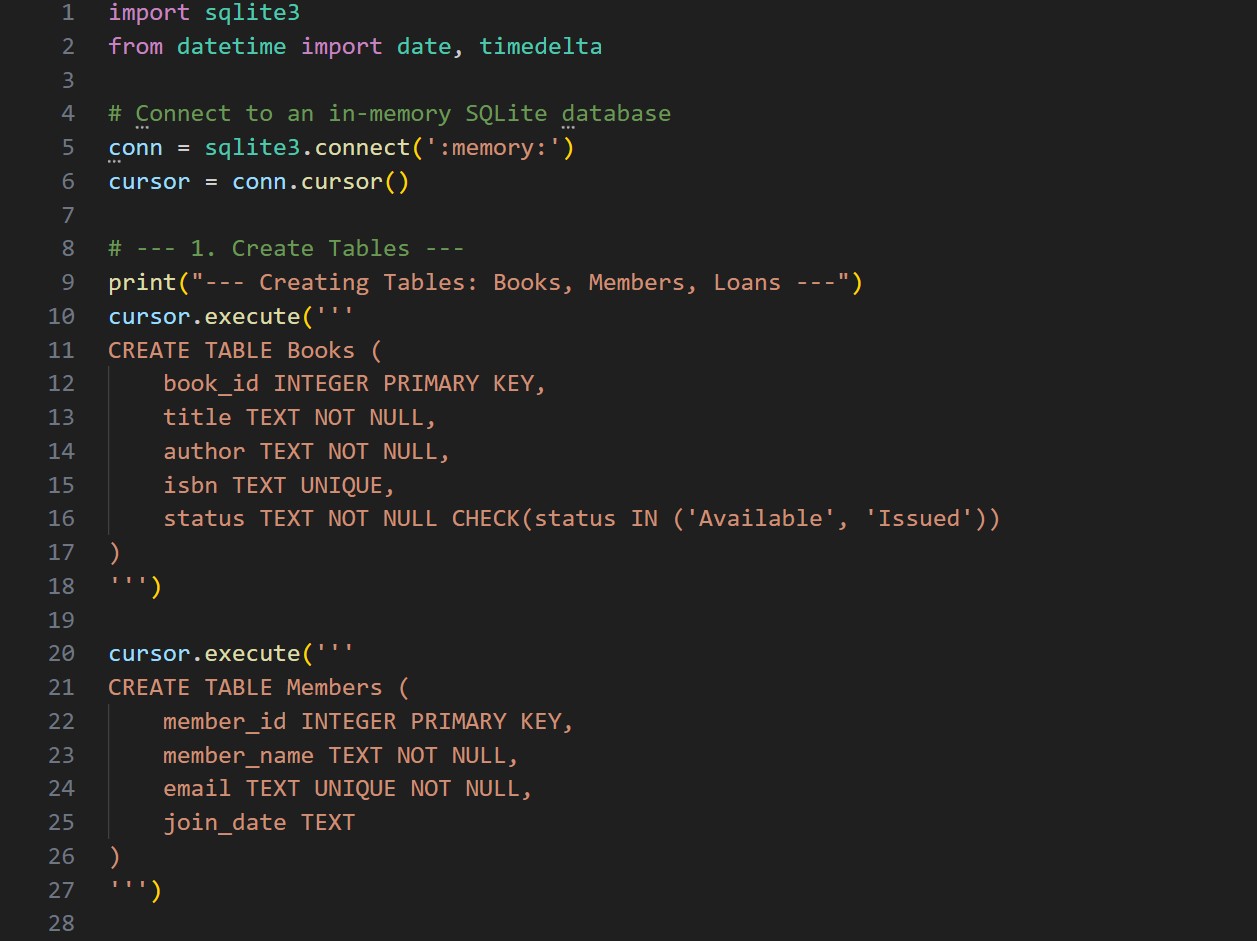
Expected Output:

• Schema + SQL queries demonstrating joins and conditions

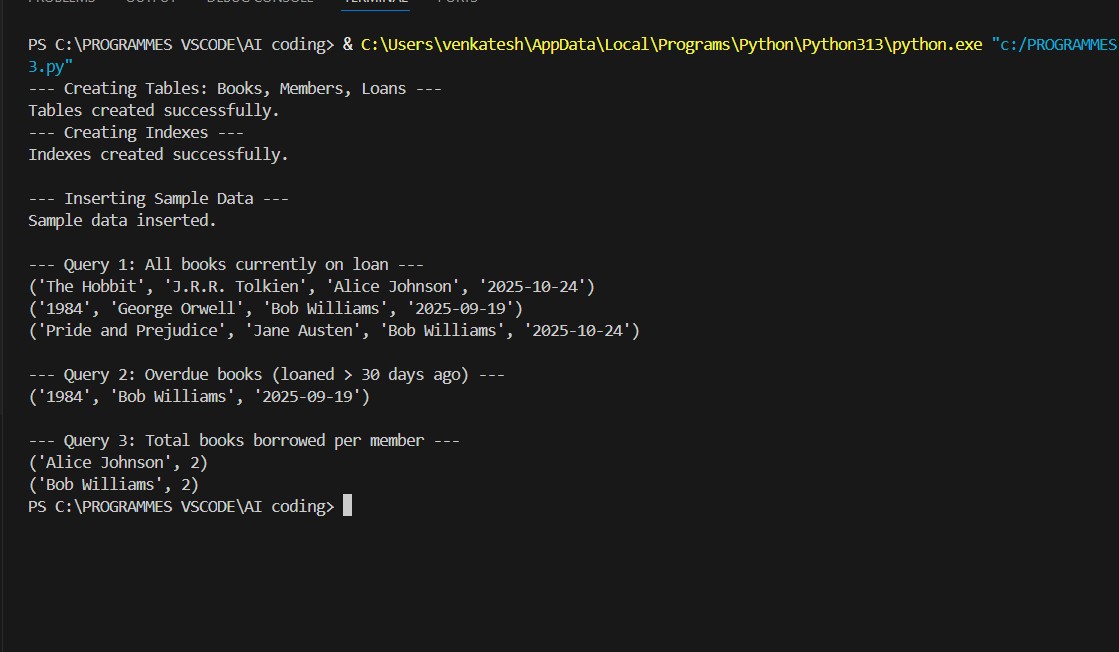
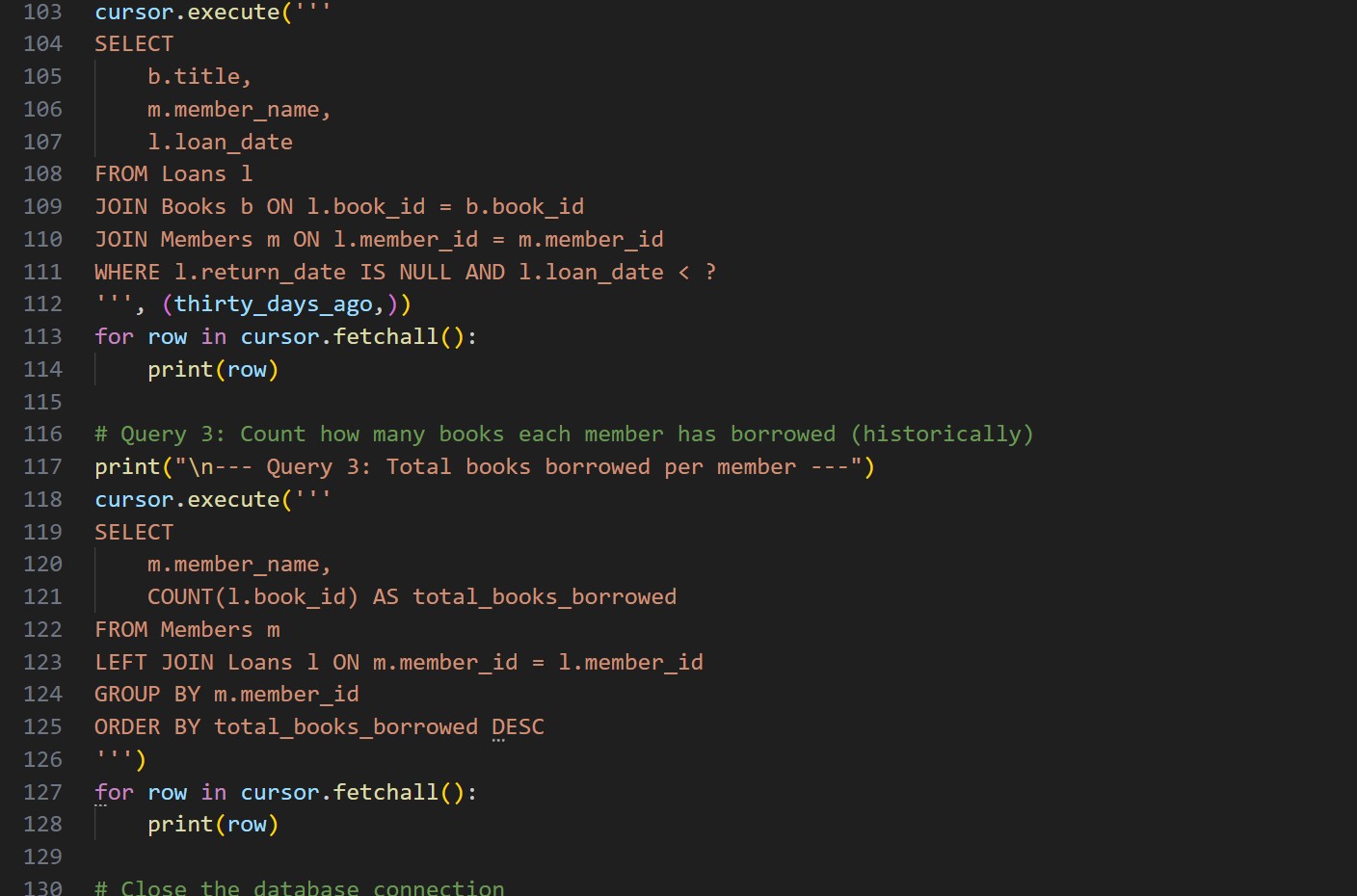
**PROMPT:**

Use AI to design a Library Management System database with three tables Books, Members, and Loans. Define primary keys, foreign keys, and suggest indexes for faster searching. Generate SQL queries to get all books currently issued, find overdue books loaned more than 30 days ago), and count how many books each member has borrowed.

**CODE:**



**OUTPUT:**



**OBSERVATION:**

AI created a normalized database schema with proper relationships between books, members, and loans. It suggested using indexes on frequently searched columns like member\_id and book\_id to improve performance. The generated SQL queries for issued books, overdue books, and total loans per member were correct, efficient, and easy to execute.

# TASK-4

**QUESTION:**

4 – Real-Time Application: Online Shopping Database Scenario:

Design a database for an E-commerce platform.

Requirements:

* Tables: Users, Products, Orders, OrderDetails.
* Generate queries to:

o Retrieve all orders by a user. o Find the most purchased product. o Calculate total revenue in a given month.

* AI should also suggest normalization improvements and query optimization.

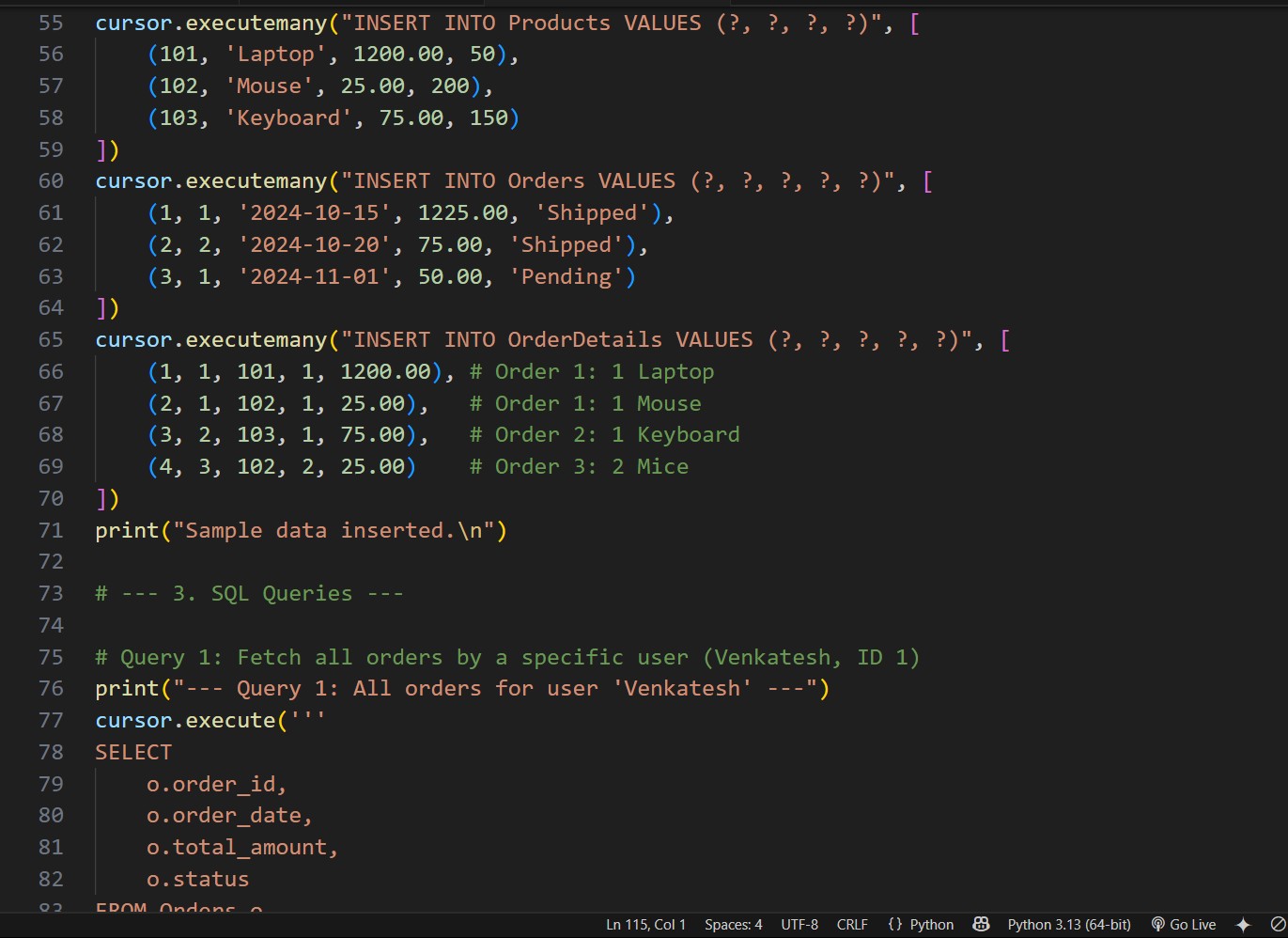
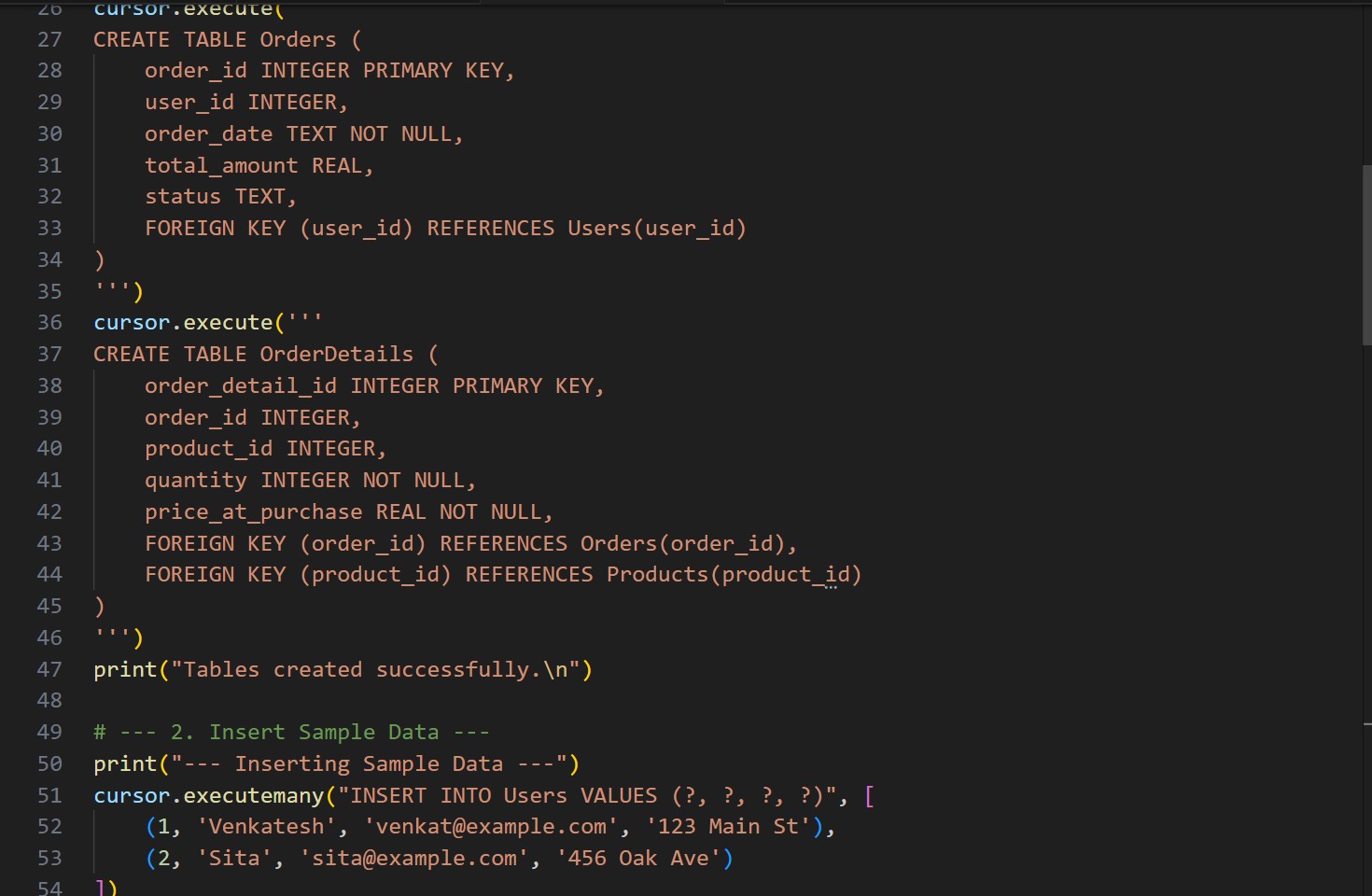
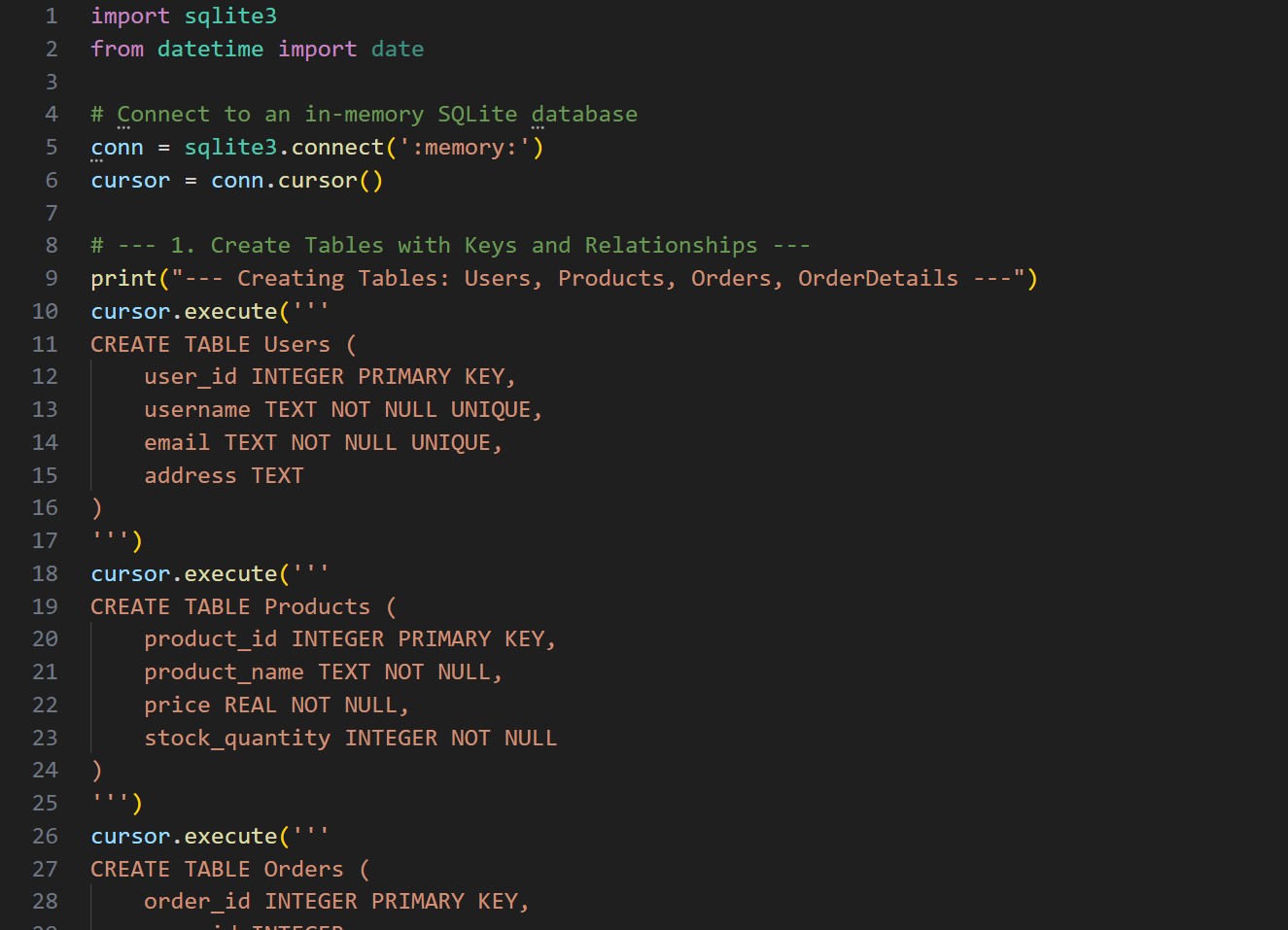
Expected Output:

* Complete schema with relationships + SQL queries for analytics

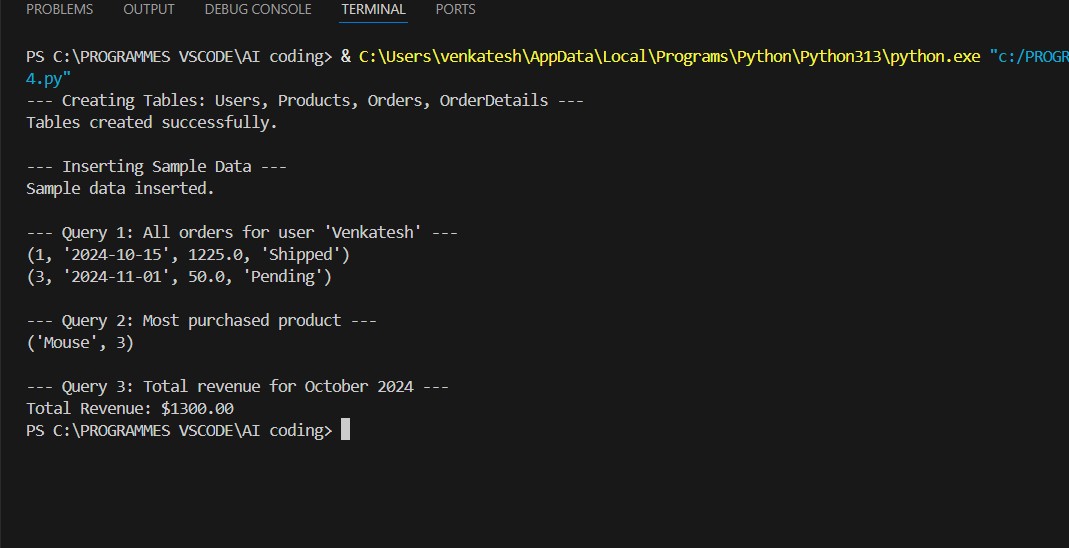
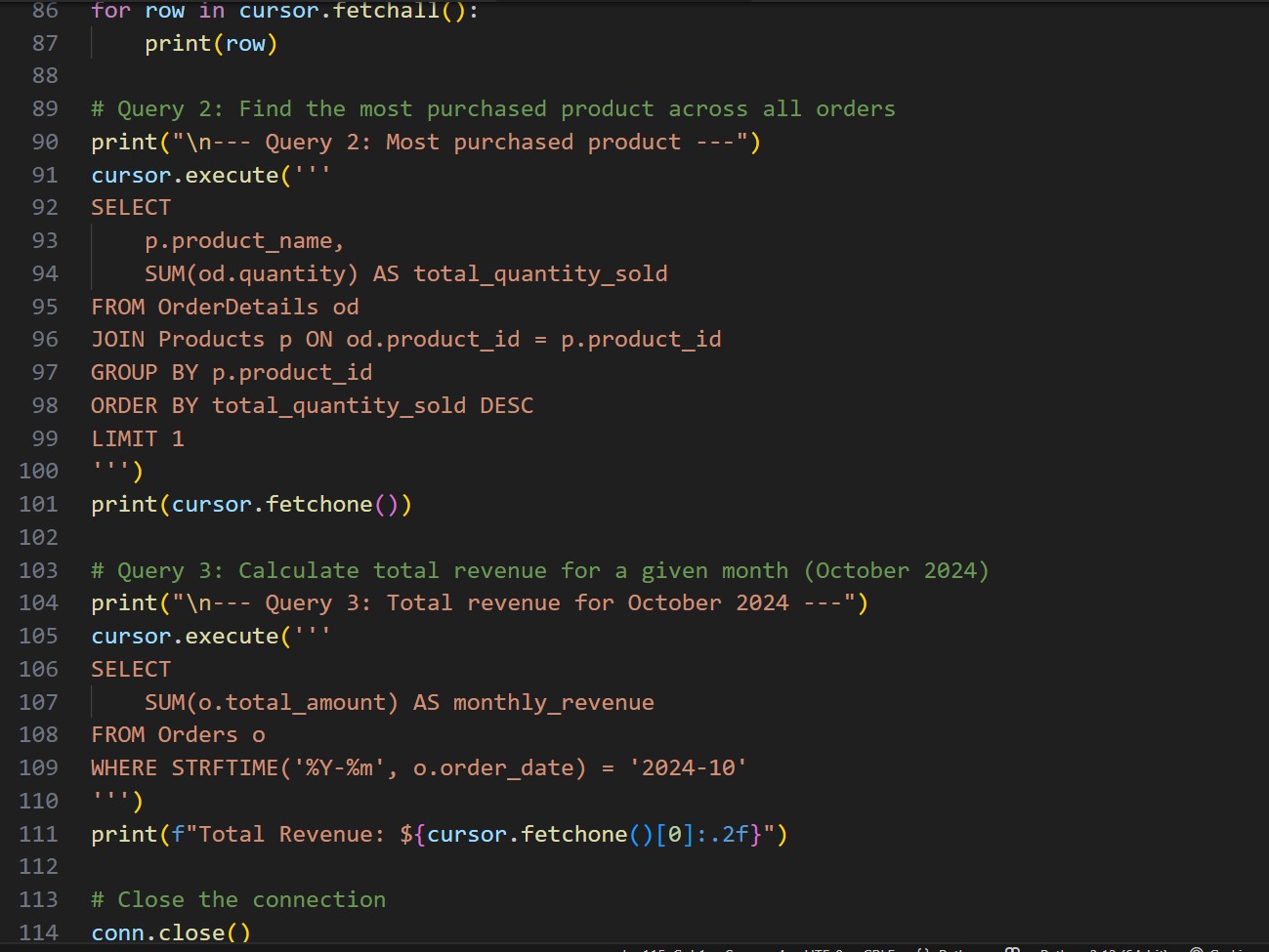
**PROMPT:**

Use AI to design an Online Shopping Database with four tables Users, Products, Orders, and OrderDetails. Define primary and foreign keys with proper relationships. Generate SQL queries to fetch all orders by a user, find the most purchased product, and calculate total revenue for a given month. Also, suggest normalization and query optimization techniques

**CODE:**



**OUTPUT:**



**OBSERVATION:**

AI successfully generated a normalized database schema with clear relationships between users, products, and orders. It included foreign keys for data consistency and suggested indexing for faster query execution. The SQL queries for user orders, top-selling products, and monthly revenue were accurate, efficient, and easy to understand.