**1. Multiple Threads**

# Start 3 threads that print different messages with delays.2

**2 Thread with Arguments**

# Create a thread that calculates square of a given number.

**4. Thread Class Subclassing**

# Create a custom thread class that prints thread ID and timestamp.

**4. Join Threads**

# Start two threads and wait for them to complete using join().

**5 Shared Resource (Without Lock)**

# Create two threads that increment a shared counter 10000 times. No lock used.

**6. Shared Resource (With Lock)**

# Fix the race condition from Challenge 6 using threading.Lock.

**7. Daemon Thread**

# Create a daemon thread that prints a message every second in the background.

### ****8. Use of Timer****

# Create a thread that prints a message after 5 seconds using threading.Timer.

## ****9. Create a Table for Products****

**Challenge:**  
Write Python code to create a table named products with columns: id (serial, primary key), name (varchar), price (numeric), and in\_stock (boolean).

Hint: Use *CREATE TABLE* SQL statement.

## ****10. Insert Multiple Records****

**Challenge:**  
Insert at least 3 different products into the products table using a single Python script.

Hint: Use *executemany* or multiple *execute* calls.

## ****11. Query Products in Stock****

**Challenge:**  
Write a Python script to fetch and print all products where in\_stock is True.

Hint: Use a *SELECT* statement with a *WHERE* clause.

## ****12. Update Product Price****

**Challenge:**  
Update the price of a product (by name) to a new value. The product name and new price should be variables in your script.

Hint: Use a parameterized *UPDATE* statement.

## ****13. Delete Out-of-Stock Products****

**Challenge:**  
Delete all products from the table where in\_stock is False.

Hint: Use a *DELETE* statement with a *WHERE* clause.

## ****14. Count Products****

**Challenge:**  
Write a Python script to count and print the total number of products in the table.

Hint: Use *SELECT COUNT(\*)*.

## ****16. List Products with Price Above a Threshold****

**Challenge:**  
Fetch and print all products with a price greater than a given threshold (e.g., 100).

Hint: Use a parameterized *SELECT* statement.

## ****17. Add a Timestamp Column****

**Challenge:**  
Alter the products table to add a new column created\_at (timestamp) with a default value of the current time.

Hint: Use *ALTER TABLE* and *DEFAULT NOW()*.

## ****18. Fetch Products Ordered by Price****

**Challenge:**  
Write a script to fetch and print all products, ordered by price in descending order.

Hint: Use *ORDER BY* in your *SELECT* statement.