1. Set the variable test1 to the string 'This is a test of the emergency text system,' and save test1 to a file named test.txt.

test1 = 'This is a test of the emergency text system'

with open('test.txt', 'w') as file:

file.write(test1)

2. Read the contents of the file test.txt into the variable test2. Is there a difference between test 1 and test 2?

with open('test.txt', 'r') as file:

test2 = file.read()

if test1 == test2:

print("There is no difference between test1 and test2.")

else:

print("There is a difference between test1 and test2.")

3. Create a CSV file called books.csv by using these lines:

title,author,year

The Weirdstone of Brisingamen,Alan Garner,1960

Perdido Street Station,China Miéville,2000

Thud!,Terry Pratchett,2005

The Spellman Files,Lisa Lutz,2007

Small Gods,Terry Pratchett,1992

import csv

lines = [

['title', 'author', 'year'],

['The Weirdstone of Brisingamen', 'Alan Garner', '1960'],

['Perdido Street Station', 'China Miéville', '2000'],

['Thud!', 'Terry Pratchett', '2005'],

['The Spellman Files', 'Lisa Lutz', '2007'],

['Small Gods', 'Terry Pratchett', '1992']

]

with open('books.csv', 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(lines)

4. Use the sqlite3 module to create a SQLite database called books.db, and a table called books with these fields: title (text), author (text), and year (integer).

import sqlite3

# Connect to the database (or create it if it doesn't exist)

conn = sqlite3.connect('books.db')

# Create a cursor object to execute SQL commands

cursor = conn.cursor()

# Create the 'books' table with the specified fields

cursor.execute('''CREATE TABLE books

(title TEXT, author TEXT, year INTEGER)''')

# Commit the changes and close the connection

conn.commit()

conn.close()

5. Read books.csv and insert its data into the book table.

import sqlite3

import csv

# Connect to the database

conn = sqlite3.connect('books.db')

cursor = conn.cursor()

# Read the CSV file

with open('books.csv', 'r') as file:

reader = csv.reader(file)

next(reader) # Skip the header row

# Iterate over the rows and insert data into the table

for row in reader:

title, author, year = row

cursor.execute("INSERT INTO books (title, author, year) VALUES (?, ?, ?)",

(title, author, year))

# Commit the changes and close the connection

conn.commit()

conn.close()

6. Select and print the title column from the book table in alphabetical order.

import sqlite3

# Connect to the database

conn = sqlite3.connect('books.db')

cursor = conn.cursor()

# Select the title column from the books table in alphabetical order

cursor.execute("SELECT title FROM books ORDER BY title")

# Fetch all the rows

rows = cursor.fetchall()

# Print the titles

for row in rows:

print(row[0])

# Close the connection

conn.close()

7. From the book table, select and print all columns in the order of publication.

import sqlite3

# Connect to the database

conn = sqlite3.connect('books.db')

cursor = conn.cursor()

# Select all columns from the books table ordered by year

cursor.execute("SELECT \* FROM books ORDER BY year")

# Fetch all the rows

rows = cursor.fetchall()

# Print the rows

for row in rows:

print(row)

# Close the connection

conn.close()

8. Use the sqlalchemy module to connect to the sqlite3 database books.db that you just made in exercise 6.

from sqlalchemy import create\_engine

# Create the database engine

engine = create\_engine('sqlite:///books.db')

# Connect to the database

conn = engine.connect()

# Perform database operations...

# Close the connection

conn.close()

9. Install the Redis server and the Python redis library (pip install redis) on your computer. Create a Redis hash called test with the fields count (1) and name ('Fester Bestertester'). Print all the fields for test.

import redis

# Connect to the Redis server

r = redis.Redis(host='localhost', port=6379)

# Create the hash

r.hset('test', 'count', 1)

r.hset('test', 'name', 'Fester Bestertester')

# Print all the fields in the hash

fields = r.hgetall('test')

for field, value in fields.items():

print(f'{field.decode()}: {value.decode()}')

10. Increment the count field of test and print it.

import redis

# Connect to the Redis server

r = redis.Redis(host='localhost', port=6379)

# Increment the count field by 1

r.hincrby('test', 'count', 1)

# Print the updated count value

count = r.hget('test', 'count')

print(f'Updated count: {count.decode()}')