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.

ANS :- test1 = 'This is a test of the emergency text system.'

# Save test1 to test.txt

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?

ANS :- # Read the contents of test.txt into test2

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

test2 = file.read()

# Compare test1 and test2

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

ANS :- 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']

]

filename = 'books.csv'

with open(filename, 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(lines)

print(f"File '{filename}' has been created.")

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).

ANS :- import sqlite3

# Connect to the database (create 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

cursor.execute('''

CREATE TABLE IF NOT EXISTS books (

title TEXT,

author TEXT,

year INTEGER

)

''')

# Commit the changes and close the connection

conn.commit()

conn.close()

print("Database 'books.db' and table 'books' have been created.")

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

ANS :- import csv

import sqlite3

# Connect to the database

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

cursor = conn.cursor()

# Read the CSV file and insert data into the table

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

csv\_reader = csv.reader(file)

next(csv\_reader) # Skip the header row

for row in csv\_reader:

title, author, year = row

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

# Commit the changes and close the connection

conn.commit()

conn.close()

print("Data from 'books.csv' has been inserted into the 'books' table.")

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

ANS :- import sqlite3

# Connect to the database

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

cursor = conn.cursor()

# Select and print the title column in alphabetical order

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

# Fetch all rows and print the title column

rows = cursor.fetchall()

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.

ANS :- import sqlite3

# Connect to the database

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

cursor = conn.cursor()

# Select and print all columns in the order of publication

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

# Fetch all rows and print all columns

rows = cursor.fetchall()

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.

ANS :- from sqlalchemy import create\_engine

# Connect to the SQLite database

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

# Test the connection by executing a query

result = engine.execute("SELECT 'Connection successful!' AS message")

print(result.scalar())

# Close the connection

engine.dispose()

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.

ANS :- import redis

# Connect to Redis

r = redis.Redis(host='localhost', port=6379, db=0)

# Create a Redis hash called 'test'

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

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

# Print all fields for 'test'

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.

ANS :- import redis

# Connect to Redis

r = redis.Redis(host='localhost', port=6379, db=0)

# Increment the count field of test

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

# Get the updated count value and print it

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

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