-- Create the database

CREATE DATABASE IF NOT EXISTS treasure\_hunt\_challenge;

-- Use the database

USE treasure\_hunt\_challenge;

-- Create students table

CREATE TABLE students (

student\_id INT PRIMARY KEY,

student\_name VARCHAR(255),

enrollment\_date DATE

);

-- Insert sample data into students table

INSERT INTO students VALUES

(1, 'John Doe', '2023-01-01'),

(2, 'Jane Smith', '2023-02-15'),

(3, 'Carlos Rodríguez', '2023-03-10');

-- Create courses table

CREATE TABLE courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(255),

instructor VARCHAR(255)

);

-- Insert sample data into courses table

INSERT INTO courses VALUES

(101, 'Introduction to Databases', 'Prof. Anderson'),

(102, 'Web Development Basics', 'Prof. Garcia'),

(103, 'Data Structures and Algorithms', 'Prof. Johnson');

-- Create enrollment table to represent the relationship between students and courses

CREATE TABLE enrollment (

enrollment\_id INT PRIMARY KEY,

student\_id INT,

course\_id INT,

FOREIGN KEY (student\_id) REFERENCES students(student\_id),

FOREIGN KEY (course\_id) REFERENCES courses(course\_id)

);

-- Insert sample data into enrollment table

INSERT INTO enrollment VALUES

(1, 1, 101),

(2, 1, 102),

(3, 2, 103),

(4, 3, 101),

(5, 3, 102);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SQL Queries for Treasure Hunt:

Query 1: Get all student names and their enrollment dates:

SELECT student\_name, enrollment\_date FROM students;

Query 2: Get the names of courses and their respective instructors:

SELECT course\_name, instructor FROM courses;

Query 3: Get a list of students who are enrolled in at least one course:

SELECT DISTINCT students.student\_name

FROM students

JOIN enrollment ON students.student\_id = enrollment.student\_id;

Query 4: Get the count of courses each student is enrolled in:

SELECT students.student\_name, COUNT(enrollment.course\_id) AS enrolled\_courses

FROM students

LEFT JOIN enrollment ON students.student\_id = enrollment.student\_id

GROUP BY students.student\_name;

Query 5: Find the courses with more than two enrolled students:

SELECT courses.course\_name

FROM courses

JOIN enrollment ON courses.course\_id = enrollment.course\_id

GROUP BY courses.course\_name

HAVING COUNT(DISTINCT enrollment.student\_id) > 2;

Query 6: Get the student names who are enrolled in the "Introduction to Databases" course:

SELECT students.student\_name

FROM students

JOIN enrollment ON students.student\_id = enrollment.student\_id

JOIN courses ON enrollment.course\_id = courses.course\_id

WHERE courses.course\_name = 'Introduction to Databases';

Query 7: Find the courses with no enrolled students:

SELECT courses.course\_name

FROM courses

LEFT JOIN enrollment ON courses.course\_id = enrollment.course\_id

WHERE enrollment.enrollment\_id IS NULL;

Query 8: Get the student names and the courses they are enrolled in, sorted by student name:

SELECT students.student\_name, courses.course\_name

FROM students

LEFT JOIN enrollment ON students.student\_id = enrollment.student\_id

LEFT JOIN courses ON enrollment.course\_id = courses.course\_id

ORDER BY students.student\_name;

Query 9: Count the total number of students enrolled in all courses:

SELECT COUNT(DISTINCT enrollment.student\_id) AS total\_students

FROM enrollment;

Query 10: Find the student(s) with the earliest enrollment date:

SELECT student\_name, enrollment\_date

FROM students

WHERE enrollment\_date = (SELECT MIN(enrollment\_date) FROM students);