CREATE DATABASE studentsDB;

USE studentsDB;

-- Create the students table

CREATE TABLE students (

student\_id INT PRIMARY KEY,

student\_name VARCHAR(50),

age INT,

course\_id INT

);

-- Insert updated records into the students table

INSERT INTO students (student\_id, student\_name, age, course\_id)

VALUES

(1, 'Aarav Sharma', 24, 1),

(2, 'Isha Gupta', 19, 2),

(3, 'Rohan Mehta', 23, 3),

(4, 'Priya Patel', 27, 2),

(5, 'Ananya Singh', 20, 3);

-- Create the courses table

CREATE TABLE courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(50)

);

-- Insert updated records into the courses table

INSERT INTO courses (course\_id, course\_name)

VALUES

(1, 'Mathematics'),

(2, 'Physics'),

(3, 'Computer Science');

-- Query 1: Join students and courses to display student names with course names

SELECT

students.student\_name AS Student\_Name,

courses.course\_name AS Course\_Name

FROM

students

JOIN

courses

ON

students.course\_id = courses.course\_id;

-- Query 2: Count the total students in each course

SELECT

courses.course\_name AS Course\_Name,

COUNT(students.student\_id) AS Total\_Students

FROM

students

JOIN

courses

ON

students.course\_id = courses.course\_id

GROUP BY

courses.course\_name;

-- Query 3: Select student names for students older than 21 years

SELECT

student\_name

FROM

students

WHERE

age > 21;

-- Query 4: Calculate the average age of students grouped by course names

SELECT

courses.course\_name AS Course\_Name,

AVG(students.age) AS Average\_Age

FROM

students

JOIN

courses

ON

students.course\_id = courses.course\_id

GROUP BY

courses.course\_name;

-- Query 5: Find the course with the highest number of students

SELECT

courses.course\_name AS Course\_Name,

COUNT(students.student\_id) AS Total\_Students

FROM

students

JOIN

courses

ON

students.course\_id = courses.course\_id

GROUP BY

courses.course\_name

ORDER BY

Total\_Students DESC

LIMIT 1;