**Database Management System**

**(Assignment No 01)**



Session (2022-2026)

Program/Class

**BS-Computer Science / 5rd Section-A**

Submitted By:

Student Name: Shaheer Ali Roll Number: 301-221044

Supervised By:

Ms. Muneeba Darwaish

Lecturer

CS& IT Department

**Hazara University, Mansehra**

**Type Of Database To Create**

You are managing a **student management system** database that contains the following tables:

**1. Students**

| **student\_id** | **first\_name** | **last\_name** | **enrollment\_year** | **major** | **gpa** |
| --- | --- | --- | --- | --- | --- |
| 1 | John | Doe | 2022 | CS | 3.5 |
| 2 | Jane | Smith | 2021 | IT | 3.7 |
| 3 | Alice | Johnson | 2023 | CS | 3.2 |
| 4 | Bob | Lee | 2021 | IT | 2.8 |
| 5 | Charlie | Brown | 2022 | CS | 3.9 |

**2. Courses**

| **course\_id** | **course\_name** | **credit\_hours** |
| --- | --- | --- |
| 101 | Database Systems | 3 |
| 102 | Computer Networks | 4 |
| 103 | Software Engineering | 3 |
| 104 | Cybersecurity Basics | 2 |

**3. Enrollments**

| **enrollment\_id** | **student\_id** | **course\_id** | **semester** | **grade** |
| --- | --- | --- | --- | --- |
| 1 | 1 | 101 | Spring | A |
| 2 | 2 | 102 | Fall | B |
| 3 | 3 | 101 | Fall | C |
| 4 | 4 | 104 | Spring | B |
| 5 | 1 | 103 | Fall | A |

**Creating Database StudentManagement**

CREATE DATABASE StudentManagement;

USE StudentManagement;

CREATE TABLE Students (

student\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

enrollment\_year INT,

major VARCHAR(50),

gpa DECIMAL(3, 2)

);

INSERT INTO Students (student\_id, first\_name, last\_name, enrollment\_year, major, gpa) VALUES

(1, 'John', 'Doe', 2022, 'CS', 3.5),

(2, 'Jane', 'Smith', 2021, 'IT', 3.7),

(3, 'Alice', 'Johnson', 2023, 'CS', 3.2),

(4, 'Bob', 'Lee', 2021, 'IT', 2.8),

(5, 'Charlie', 'Brown', 2022, 'CS', 3.9);

CREATE TABLE Courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100),

credit\_hours INT

);

INSERT INTO Courses (course\_id, course\_name, credit\_hours) VALUES

(101, 'Database Systems', 3),

(102, 'Computer Networks', 4),

(103, 'Software Engineering', 3),

(104, 'Cybersecurity Basics', 2);

CREATE TABLE Enrollments (

enrollment\_id INT PRIMARY KEY,

student\_id INT,

course\_id INT,

semester VARCHAR(20),

grade CHAR(1),

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

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

);

INSERT INTO Enrollments (enrollment\_id, student\_id, course\_id, semester, grade) VALUES

(1, 1, 101, 'Spring', 'A'),

(2, 2, 102, 'Fall', 'B'),

(3, 3, 101, 'Fall', 'C'),

(4, 4, 104, 'Spring', 'B'),

(5, 1, 103, 'Fall', 'A');

**Write SQL queries to answer the following questions:**

1. **Retrieve all students** who are enrolled in a **CS major** and have a **GPA greater than 3.0**.

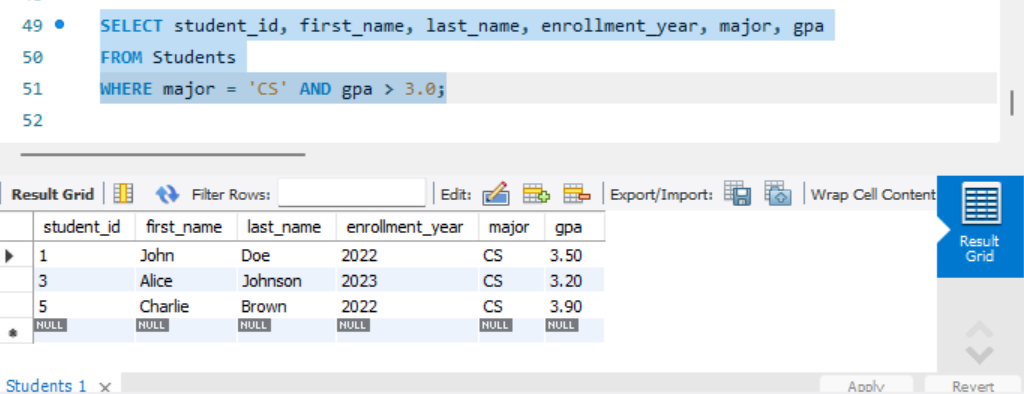
**Query:**

SELECT student\_id, first\_name, last\_name, enrollment\_year, major, gpa

FROM Students

WHERE major = 'CS' AND gpa > 3.0;

**Screenshot:**



1. Find the **average GPA** of students enrolled in the **IT major**.

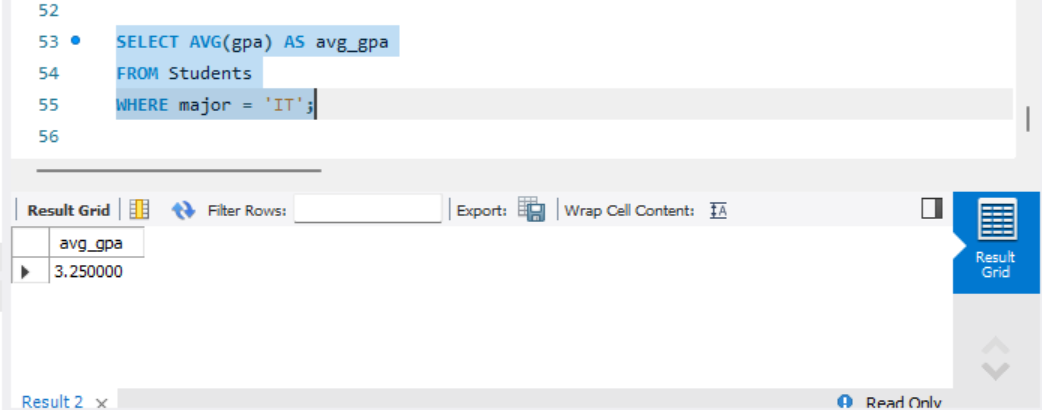
**Query:**

SELECT AVG(gpa) AS avg\_gpa

FROM Students

WHERE major = 'IT';

**Screenshot:**



1. List all courses along with the **number of students enrolled** in each course.

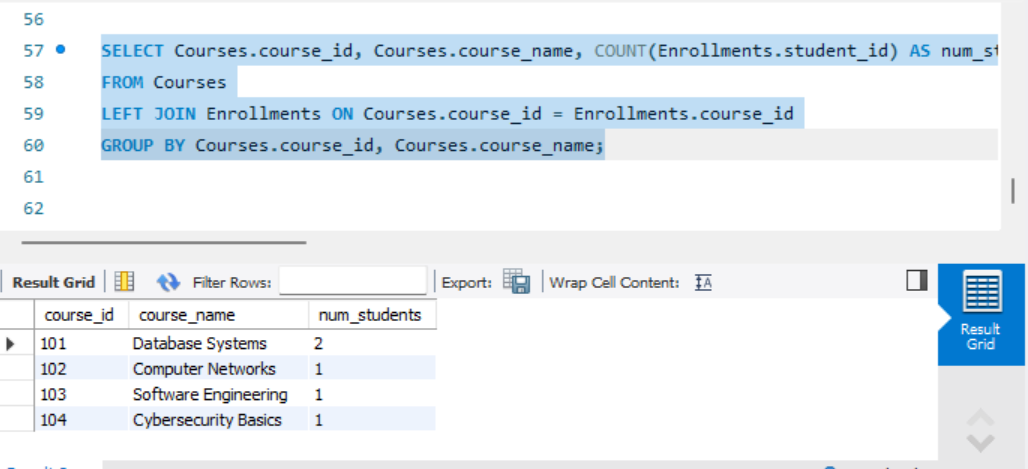
**Query:**

SELECT Courses.course\_id, Courses.course\_name, COUNT(Enrollments.student\_id) AS num\_students

FROM Courses

LEFT JOIN Enrollments ON Courses.course\_id = Enrollments.course\_id

GROUP BY Courses.course\_id, Courses.course\_name;

**Screenshot:**

1. Identify students who are taking the **Database Systems course** in any semester.

**Query:**

SELECT Students.student\_id, Students.first\_name, Students.last\_name

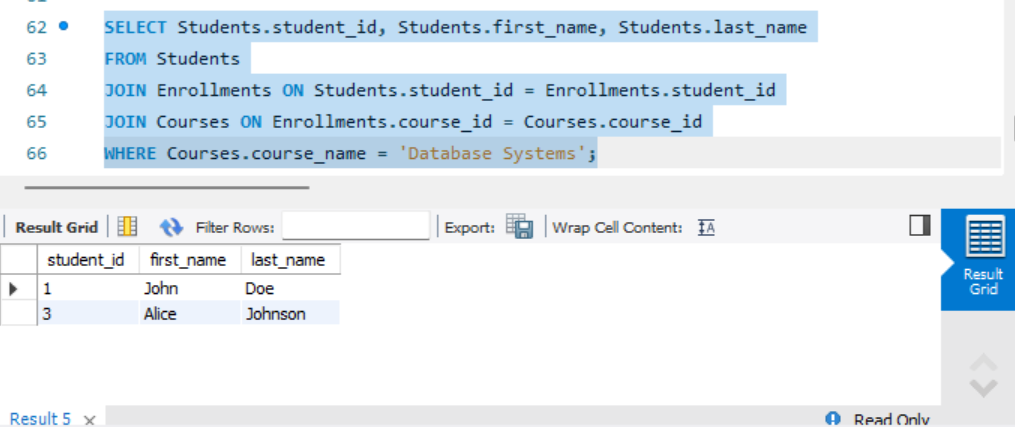
FROM Students

JOIN Enrollments ON Students.student\_id = Enrollments.student\_id

JOIN Courses ON Enrollments.course\_id = Courses.course\_id

WHERE Courses.course\_name = 'Database Systems';

**Screenshot:**



1. Display students who received a **grade of A in at least two courses**.

**Query:**

SELECT Students.student\_id, Students.first\_name, Students.last\_name

FROM Students

JOIN Enrollments ON Students.student\_id = Enrollments.student\_id

WHERE Enrollments.grade = 'A'

GROUP BY Students.student\_id, Students.first\_name, Students.last\_name

HAVING COUNT(Enrollments.course\_id) >= 2;

**Screenshot:**

