```
-- Create the database
CREATE DATABASE StudentManagement;
drop database StudentManagement;

-- Use the database
USE StudentManagement;

-- Create the Students table
CREATE TABLE Students (
    StudentID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(50),
    Gender VARCHAR(1) CHECK (Gender IN ('M', 'F')),
    Age INT,
    Grade VARCHAR(10),
    MathScore INT,
    ScienceScore INT,
    EnglishScore INT,
    Email VARCHAR(100)
);
-- Insert sample data into the Students table
-- Insert sample data into the Students table with email values
INSERT INTO Students (Name, Gender, Age, Grade, MathScore, ScienceScore, EnglishScore, Email)
VALUES
('Alice', 'F', 14, 'A', 90, 85, 88, 'alice14@example.com'),
('Bob', 'M', 15, 'B', 70, 75, 72, 'bob15@example.com'),
('Charlie', 'M', 14, 'A', 95, 92, 91, 'charlie14@example.com'),
('Diana', 'F', 15, 'B', 68, 72, 74, 'diana15@example.com'),
('Ethan', 'M', 14, 'C', 55, 60, 58, 'ethan14@example.com'),
('Fiona', 'F', 15, 'A', 85, 90, 88, 'fiona15@example.com'),
('George', 'M', 14, 'B', 75, 78, 80, 'george14@example.com'),
('Hannah', 'F', 15, 'C', 60, 65, 62, 'hannah15@example.com'),
```

```
('Ian', 'M', 14, 'A', 88, 92, 89, 'ian14@example.com'),
('Jill', 'F', 15, 'B', 72, 74, 76, 'jill15@example.com');

SELECT * FROM Students;
DROP TABLE Students;

CREATE TABLE courses (
    course_id INT PRIMARY KEY AUTO_INCREMENT,
    course_name VARCHAR(255) NOT NULL,
    course_description TEXT
);

INSERT INTO courses (course_name, course_description)
VALUES
('Computer Science 101', 'Introduction to computer science concepts and programming fundamentals.'),
('Mathematics for Engineers', 'Covers calculus, linear algebra, and differential equations for engineering students.'),
('Business Management', 'Principles of business management and leadership strategies.'),
('Data Science Essentials', 'Data analysis, machine learning basics, and statistical methods.'),
('Digital Marketing Basics', 'Overview of SEO, content marketing, and social media strategies.');

select * from courses;
drop table courese;

CREATE TABLE enrolments (
    enrolment_id INT PRIMARY KEY AUTO_INCREMENT,
    student_id INT NOT NULL,
    course_id INT NOT NULL,
    enrolment_date DATE NOT NULL,
    FOREIGN KEY (student_id) REFERENCES students(StudentID),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
```

);

INSERT INTO enrolments (student_id, course_id, enrolment_date)

VALUES

(1, 2, '2023-09-01'),

(2, 3, '2023-09-05'),

(3, 1, '2023-09-10'),

(4, 2, '2023-09-15'),

(5, 4, '2023-10-01');

select * from enrolments;

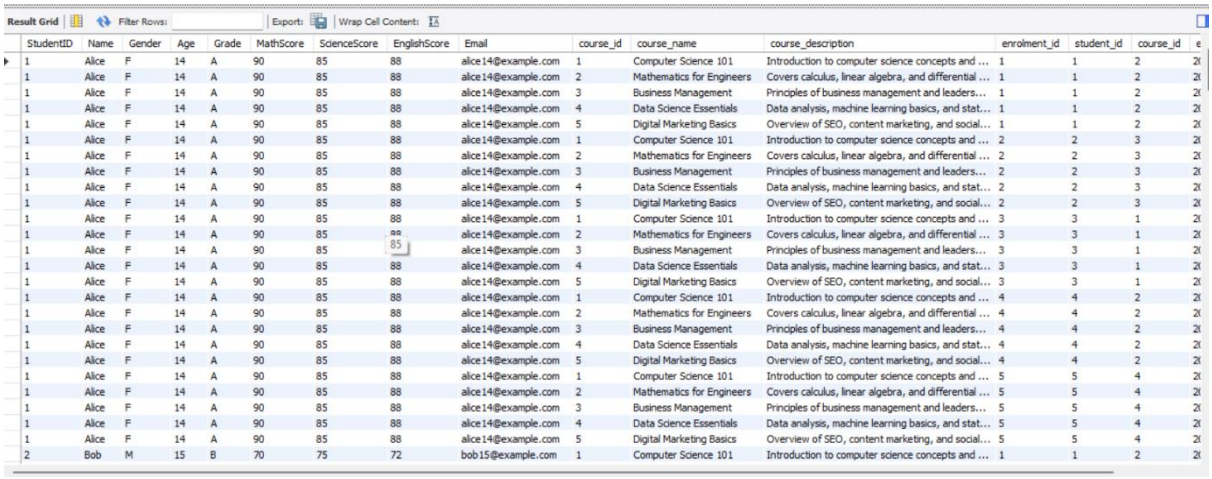
drop table enrolments;

-- Task 1: List all students and the courses they are enrolled in.

-- ● Use an INNER JOIN to combine Students, Courses, and Enrolments tables.

-- ● Select the student name and course name for all enrolled students.

select * from students inner join courses inner join enrolments;



StudentID	Name	Gender	Age	Grade	MathScore	ScienceScore	EnglishScore	Email	course_id	course_name	course_description	enrolment_id	student_id	course_id	e
1	Alice	F	14	A	90	85	88	alice14@example.com	1	Computer Science 101	Introduction to computer science concepts and ...	1	1	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	2	Mathematics for Engineers	Covers calculus, linear algebra, and differential ...	1	1	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	3	Business Management	Principles of business management and leaders...	1	1	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	4	Data Science Essentials	Data analysis, machine learning basics, and stat...	1	1	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	5	Digital Marketing Basics	Overview of SEO, content marketing, and social...	1	1	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	1	Computer Science 101	Introduction to computer science concepts and ...	2	2	3	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	2	Mathematics for Engineers	Covers calculus, linear algebra, and differential ...	2	2	3	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	3	Business Management	Principles of business management and leaders...	2	2	3	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	4	Data Science Essentials	Data analysis, machine learning basics, and stat...	2	2	3	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	5	Digital Marketing Basics	Overview of SEO, content marketing, and social...	2	2	3	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	1	Computer Science 101	Introduction to computer science concepts and ...	3	3	1	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	2	Mathematics for Engineers	Covers calculus, linear algebra, and differential ...	3	3	1	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	3	Business Management	Principles of business management and leaders...	3	3	1	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	4	Data Science Essentials	Data analysis, machine learning basics, and stat...	3	3	1	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	5	Digital Marketing Basics	Overview of SEO, content marketing, and social...	3	3	1	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	1	Computer Science 101	Introduction to computer science concepts and ...	4	4	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	2	Mathematics for Engineers	Covers calculus, linear algebra, and differential ...	4	4	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	3	Business Management	Principles of business management and leaders...	4	4	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	4	Data Science Essentials	Data analysis, machine learning basics, and stat...	4	4	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	5	Digital Marketing Basics	Overview of SEO, content marketing, and social...	4	4	2	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	1	Computer Science 101	Introduction to computer science concepts and ...	5	5	4	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	2	Mathematics for Engineers	Covers calculus, linear algebra, and differential ...	5	5	4	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	3	Business Management	Principles of business management and leaders...	5	5	4	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	4	Data Science Essentials	Data analysis, machine learning basics, and stat...	5	5	4	2(
1	Alice	F	14	A	90	85	88	alice14@example.com	5	Digital Marketing Basics	Overview of SEO, content marketing, and social...	5	5	4	2(
2	Bob	M	15	B	70	75	72	bob15@example.com	1	Computer Science 101	Introduction to computer science concepts and ...	1	1	2	2(

SELECT

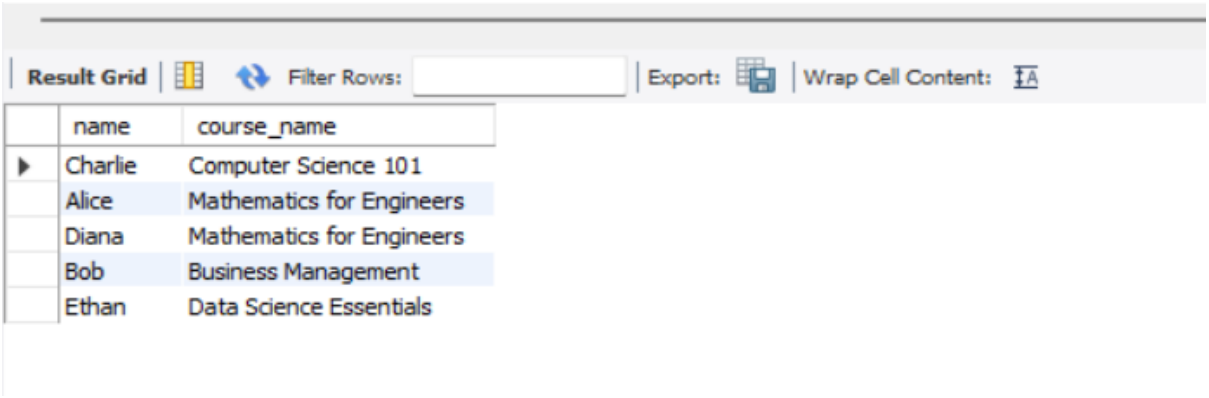
students.name, courses.course_name

FROM

enrolments

INNER JOIN

```
    students ON students.StudentID = enrolments.student_id
        INNER JOIN
    courses ON courses.course_id = enrolments.course_id;
```

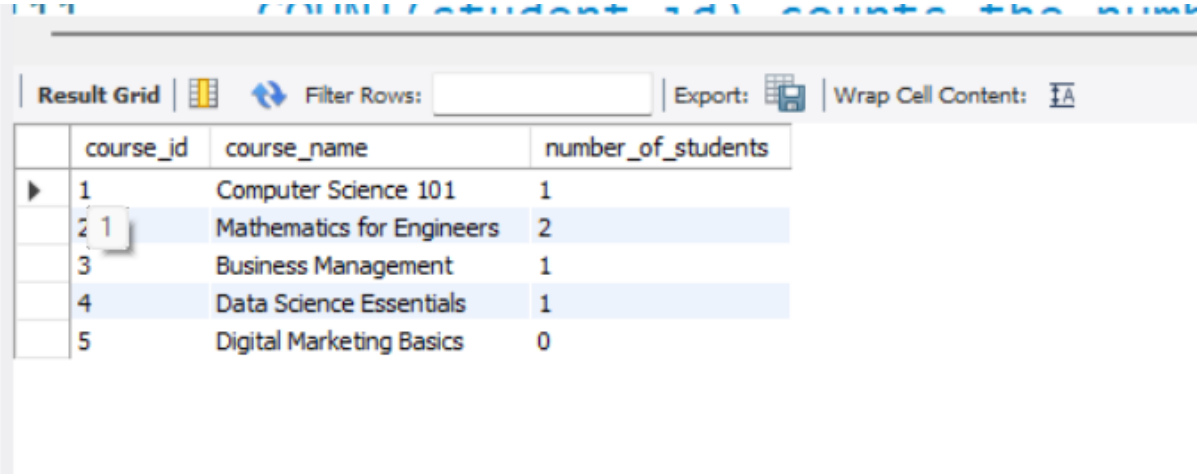


name	course_name
Charlie	Computer Science 101
Alice	Mathematics for Engineers
Diana	Mathematics for Engineers
Bob	Business Management
Ethan	Data Science Essentials

```
-- Explanation:
-- This query retrieves student names and the courses they are enrolled in using INNER JOIN between the Students, Courses, and Enrolments tables.

-- Task 2: Find the number of students enrolled in each course.
-- ● Use a LEFT JOIN between Courses and Enrolments.
-- ● Use GROUP BY to group results by course_id and course_name.
-- ● Use COUNT(student_id) to calculate the number of enrolled students.
-- ● Ensure courses with no enrolments are included in the results.

SELECT
    c.course_id,
    c.course_name,
    COUNT(e.student_id) AS number_of_students
FROM
    courses c
LEFT JOIN
    enrolments e ON c.course_id = e.course_id
GROUP BY
    c.course_id, c.course_name;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_id	course_name	number_of_students
1	Computer Science 101	1
2	Mathematics for Engineers	2
3	Business Management	1
4	Data Science Essentials	1
5	Digital Marketing Basics	0

-- Explanation:

-- LEFT JOIN ensures that courses with no enrolments are included.

-- The GROUP BY clause groups data by course_id and course_name.

-- COUNT(student_id) counts the number of enrolled students per course.

-- Task 3: List students who have enrolled in more than one course.

-- ● Use the Enrolments table.

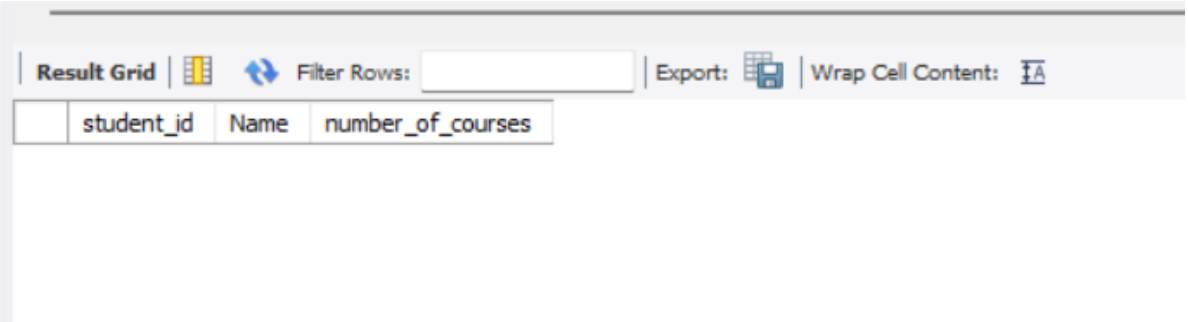
-- ● Group data by student_id.

-- ● Use COUNT(course_id) to calculate the number of courses per student.

-- ● Use the HAVING clause to filter students with enrolments greater than 1.

```
SELECT
   e.student_id,
   s.Name,
   COUNT(e.course_id) AS number_of_courses
FROM
   enrolments e
JOIN
   students s ON e.student_id = s.StudentID
GROUP BY
   e.student_id, s.Name
HAVING
```

COUNT(e.course_id) > 1;



-- Explanation:

-- The query groups enrolments by student and counts the number of courses.

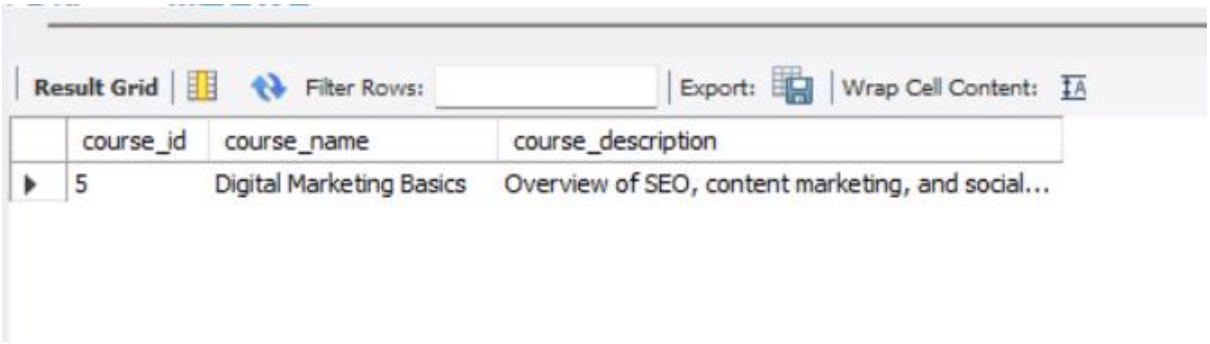
-- The HAVING COUNT(e.course_id) > 1 filter ensures only students with multiple enrolments are included.

-- Task 4: Find courses with no enrolled students.

-- ● Use a LEFT JOIN between Courses and Enrolments.

-- ● Use WHERE enrolment_id IS NULL to filter courses with no enrolments.

```
SELECT
    c.course_id,
    c.course_name,
    c.course_description
FROM
    courses c
LEFT JOIN
    enrolments e
ON
    c.course_id = e.course_id
WHERE
    e.enrolment_id IS NULL;
```



-- Explanation:

-- LEFT JOIN allows the inclusion of courses even if there are no corresponding rows in Enrolments.

-- The WHERE e.enrolment_id IS NULL condition filters courses without any enrolments.