Tamara Gray

Project Overview: Student Management System (SMS) Database Objective

The primary objective of this project is to design and create a relational database for a Student Management System (SMS) using SQL. The database should efficiently store and manage data on students, courses, instructors, and enrollments, enabling users to perform key administrative functions such as adding students, assigning courses, and generating dynamic reports through SQL queries.

Tables to be Created

1. **Students** (StudentID, Name, Gender, DOB, DepartmentID)
2. **Departments** (DepartmentID, DepartmentName)
3. **Courses** (CourseID, CourseName, DepartmentID)
4. **Enrolments** (EnrollmentID, StudentID, CourseID, EnrollmentDate)
5. I**nstructors** (provide relevant columns in relation to your tables) (InstructorID, Name, DepartmentID, Gender)

Insightful Reporting Questions (SQL-Based)

Student & Enrollment Reports

- How many students are currently enrolled in each course?
- Which students are enrolled in multiple courses, and which courses are they taking?
- What is the total number of students per department across all courses?

Course & Instructor Analysis

- Which courses have the highest number of enrollments?
- Which department has the least number of students?

Data Integrity & Operational Insights

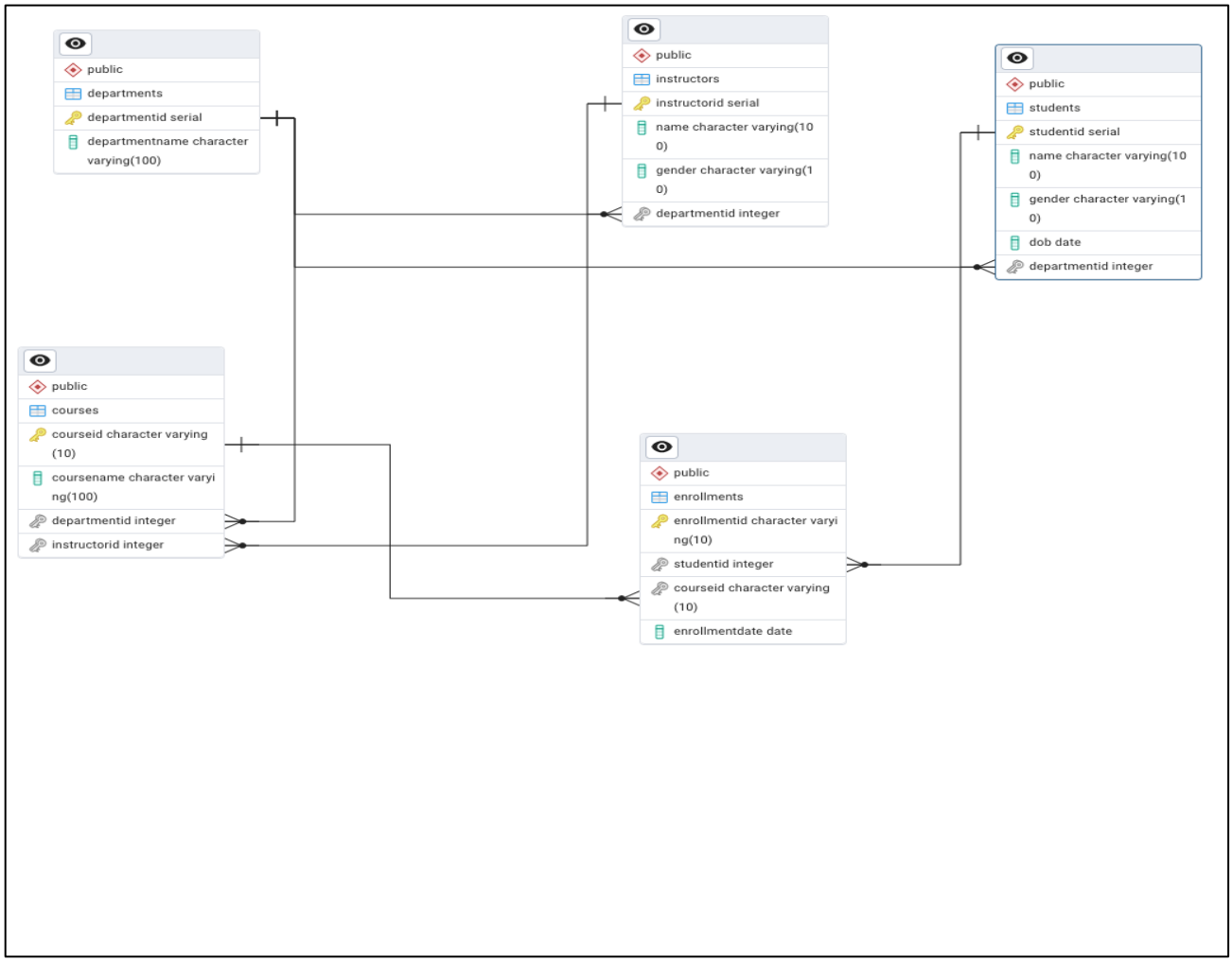
- Are there any students not enrolled in any course?
- How many courses does each student take on average?
- What is the gender distribution of students across courses and instructors?
- Which course has the highest number of male or female students enrolled?

Tamara Gray

Relationships:

- One Department has many Students
- One Department offers many Courses
- One Student can enroll in many Courses (many-to-many via Enrollments)

Entity Relationship Diagram



Tamara Gray

Entities (Tables)

1. **Students**
   Primary Key: StudentID
   Fields: Name, Gender, Date of Birth (DOB), DepartmentID
   Purpose: Contains details about each student and links them to a specific department.

2. **Departments**
   Primary Key: DepartmentID
   Fields: DepartmentName
   Purpose: Holds information about various departments within the institution.

3. **Courses**
   Primary Key: CourseID
   Fields: CourseName, DepartmentID
   Purpose: Stores details of courses, with each course associated with one department.

4. **Enrollments**
   Primary Key: EnrollmentID
   Fields: StudentID, CourseID, EnrollmentDate
   Purpose: Records which students are enrolled in which courses and when, representing the many-to-many relationship between students and courses.

5. **Instructors**
   Primary Key: InstructorID
   Fields: Name, Gender, DOB, DepartmentID, Phone
   Purpose: Maintains instructor information, with each instructor linked to a department.

Tamara Gray

Relationships

- **Students to Departments**

Each student belongs to one department (many students to one department).
The DepartmentID in the Students table is a foreign key referencing the Departments table.

- **Courses to Departments**

Each course is assigned to one department (many courses to one department).

The DepartmentID in the Courses table references the Departments table.

- **Enrollments link Students and Courses**

There is a many-to-many relationship between students and courses, managed through the Enrollments table.
StudentID in Enrollments references Students, and CourseID references Courses.

- **Instructors to Departments**

Each instructor belongs to a single department (many instructors to one department).
The DepartmentID in the Instructors table refers to Departments.

Overall Structure

The **Departments** table acts as a central entity connecting to **Students**, **Courses**, and **Instructors** through one-to-many relationships.

The **Enrollments** table bridges **Students** and **Courses**, effectively modeling their many-to-many relationship.

Foreign key constraints ensure consistency and integrity across these relationships.

Tamara Gray

SQL SCRIPTS

Creating the database

The SQL environment used was PostgreSQL, to create the database **StudentManagement,** the create database option was done in the environment.

Creating the tables

```
CREATE TABLE Departments (
    DepartmentID SERIAL PRIMARY KEY,
    DepartmentName VARCHAR(100) NOT NULL
);


CREATE TABLE Students (
    StudentID SERIAL PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Gender VARCHAR(10) CHECK (Gender IN ('MALE', 'FEMALE')),
    DOB DATE,
    DepartmentID INT REFERENCES Departments(DepartmentID)
);

CREATE TABLE Courses (
    CourseID VARCHAR(10) PRIMARY KEY,
    CourseName VARCHAR(100) NOT NULL,
    DepartmentID INT REFERENCES Departments(DepartmentID)
);

CREATE TABLE Instructors (
    InstructorID SERIAL PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Gender CHAR(1) CHECK (Gender IN ('MALE', 'FEMALE')),
    DepartmentID INT REFERENCES Departments(DepartmentID)
);

CREATE TABLE Enrollments (
    EnrollmentID VARCHAR(10) PRIMARY KEY,
    StudentID INT REFERENCES Students(StudentID),
    CourseID VARCHAR(10) REFERENCES Courses(CourseID),
   EnrollmentDate DATE
);
```

UPDATING THE COURSES TABLE – LATER ON IN THE ANALYSIS

```
ALTER TABLE courses
ADD COLUMN instructorid INT,
ADD CONSTRAINT fk_instructor
   FOREIGN KEY (instructorid)
   REFERENCES instructors(instructorid);
```

THEN… INSERTING values for instructorid into the courses tables.
For example: - This was repeated for all instructorid and each course.

```
UPDATE courses
SET instructorid = 5
WHERE courseid = 'HIS101';
```

Inserting values into the tables

Departments Table

```
INSERT INTO Departments (DepartmentName) VALUES
('Finance'),
('IT'),
('History'),
('Law'),
('Medicine'),
('Linguistics');
```

Students Table

```
INSERT INTO Students (StudentID, Name, Gender, DOB, DepartmentID)
VALUES
(1, 'Samantha Brown', 'FEMALE', '2000-03-15', 1),
(2, 'Gary Strap', 'MALE', '2001-07-22', 2),
(3, 'Corey Stevens', 'MALE', '1999-11-30', 3),
(4, 'Michelle Roe', 'FEMALE', '2002-01-05', 1),
(5, 'Shanice Pilgrim', 'FEMALE', '2000-09-18', 5),
(6, 'Davia Rose', 'FEMALE', '2001-05-27', 2),
(7, 'Matthew Dunn', 'MALE', '1998-12-12', 1),
(8, 'Jack Fern', 'MALE', '2002-04-03', 2),
(9, 'Robert Helms', 'MALE', '2000-06-14', 5),
(10, 'Paul Bogle', 'MALE', '1999-10-25', 4),
(11, 'Winston Bright', 'MALE', '2001-02-19', 5),
(12, 'Drew Barrymore', 'FEMALE', '2002-08-10', 6),
```

(13, 'Nicki Minaj', 'FEMALE', '2000-11-05', 1),
(14, 'Alissa Barrett', 'FEMALE', '2001-01-21', 2),
(15, 'Faval Green', 'FEMALE', '1999-04-08', 5),
(16, 'Portia Miller', 'FEMALE', '2000-07-31', 1),
(17, 'Stewie Davenport', 'MALE', '2002-03-17', 5),
(18, 'Ackelia Beet', 'FEMALE', '2001-09-09', 6),
(19, 'Fernando Nunez', 'MALE', '2000-02-13', 1),
(20, 'Krissan White', 'FEMALE', '1998-08-29', 2),
(21, 'Rue Robinson', 'FEMALE', '2001-06-06', 3),
(22, 'Daedra Phillips', 'FEMALE', '2002-10-01', 4),
(23, 'Shomarie Steele', 'MALE', '1999-12-24', 1),
(24, 'Peter Pan', 'MALE', '2000-05-20', 6),
(25, 'Cinderella Jones', 'FEMALE', '2001-03-11', 1),
(26, 'Stacy-Ann Dean', 'FEMALE', '2002-07-04', 2),
(27, 'Robyn Fenty', 'FEMALE', '2000-01-29', 3),
(28, 'Elizabeth Wright', 'FEMALE', '1999-06-15', 1),
(29, 'Roxxane Gordon', 'FEMALE', '2001-12-01', 1),
(30, 'King Anderson', 'MALE', '2002-02-26', 2),
(31, 'Olivia Carter', 'FEMALE', '2000-04-15', 1),
(32, 'Liam Thompson', 'MALE', '1999-11-22', 4),
(33, 'Emma Rodriguez', 'FEMALE', '2002-06-08', 4),
(34, 'Noah Wilson', 'MALE', '2001-01-30', 1),
(35, 'Ava Patel', 'FEMALE', '2000-09-17', 4),
(36, 'Elijah Kim', 'MALE', '1998-12-02', 5),
(37, 'Sophia Nguyen', 'FEMALE', '2001-05-24', 1),
(38, 'James Murphy', 'MALE', '2002-03-12', 3),
(39, 'Isabella Wright', 'FEMALE', '1999-07-29', 5),
(40, 'Benjamin Lee', 'MALE', '2000-08-18', 1),
(41, 'Mia Johnson', 'FEMALE', '2001-10-04', 5),
(42, 'Alleah Martinez', 'FEMALE', '1998-05-11', 6),
(43, 'William Scott', 'MALE', '1999-02-26', 1),
(44, 'Amelia Green', 'FEMALE', '2000-12-07', 2),
(45, 'Miah Adams', 'FEMALE', '2001-04-01', 2),
(46, 'Daniel Parker', 'MALE', '2002-09-06', 2),
(47, 'Matthew Bailey', 'MALE', '1998-10-09', 2),
(48, 'Ethan Collins', 'MALE', '1999-03-15', 6),
(49, 'Zoe Bennett', 'FEMALE', '2000-06-23', 5),
(50, 'Leandra Russell', 'FEMALE', '2001-08-30', 3);

Courses Table

INSERT INTO Courses (CourseID, CourseName, DepartmentID) VALUES
('FIN101', 'Corporate Finance', 1),
('FIN102', 'Investment Analysis', 1),

('FIN103', 'Introduction to Managerial Accounting', 1),
('FIN104', 'Financial Management', 1),
('FIN105', 'Advance Financial Management', 1),
('IT101', 'Introduction to Programming', 2),
('IT102', 'Database Systems', 2),
('HIS101', 'World History', 3),
('LAW101', 'Constitutional Law', 4),
('LAW102', 'Criminal Law', 4),
('LAW103', 'International Law', 4),
('MED101', 'Anatomy and Physiology Medicine', 5),
('LIN101', 'Phonetics and Phonology Linguistics', 6),
('LIN102', 'Syntax and Semantics Linguistics', 6);

Instructors Table

INSERT INTO Instructors (InstructorID, Name, DepartmentID, Gender)
VALUES
(1, 'Joe Brown', 1, 'Male'),
(2, 'Kelly Prance', 1, 'Female'),
(3, 'Steve Jobs', 2, 'Male'),
(4, 'John Snow', 2, 'Male'),
(5, 'Steve Whitaker', 3, 'Male'),
(6, 'Percy Jones', 4, 'Male'),
(7, 'Sheena Bailey', 5, 'Female'),
(8, 'Ackera Sommons', 6, 'Female'),
(9, 'Kandace Rowe', 6, 'Female');

Enrollments Table

INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID,
EnrollmentDate) VALUES
('E001', 1, 'FIN101', '2024-08-28'),
('E002', 2, 'IT102', '2024-09-03'),
('E003', 3, 'HIS101', '2024-09-07'),
('E004', 4, 'FIN102', '2024-09-01'),
('E005', 5, 'MED101', '2024-09-05'),
('E006', 6, 'IT102', '2024-08-30'),
('E007', 7, 'FIN101', '2024-09-06'),
('E008', 8, 'IT101', '2024-09-02'),
('E009', 9, 'MED101', '2024-09-08'),
('E010', 16, 'LAW102', '2024-09-04'),
('E011', 11, 'MED101', '2024-09-10'),
('E012', 12, 'LIN102', '2024-09-01'),
('E013', 13, 'FIN102', '2024-09-09'),

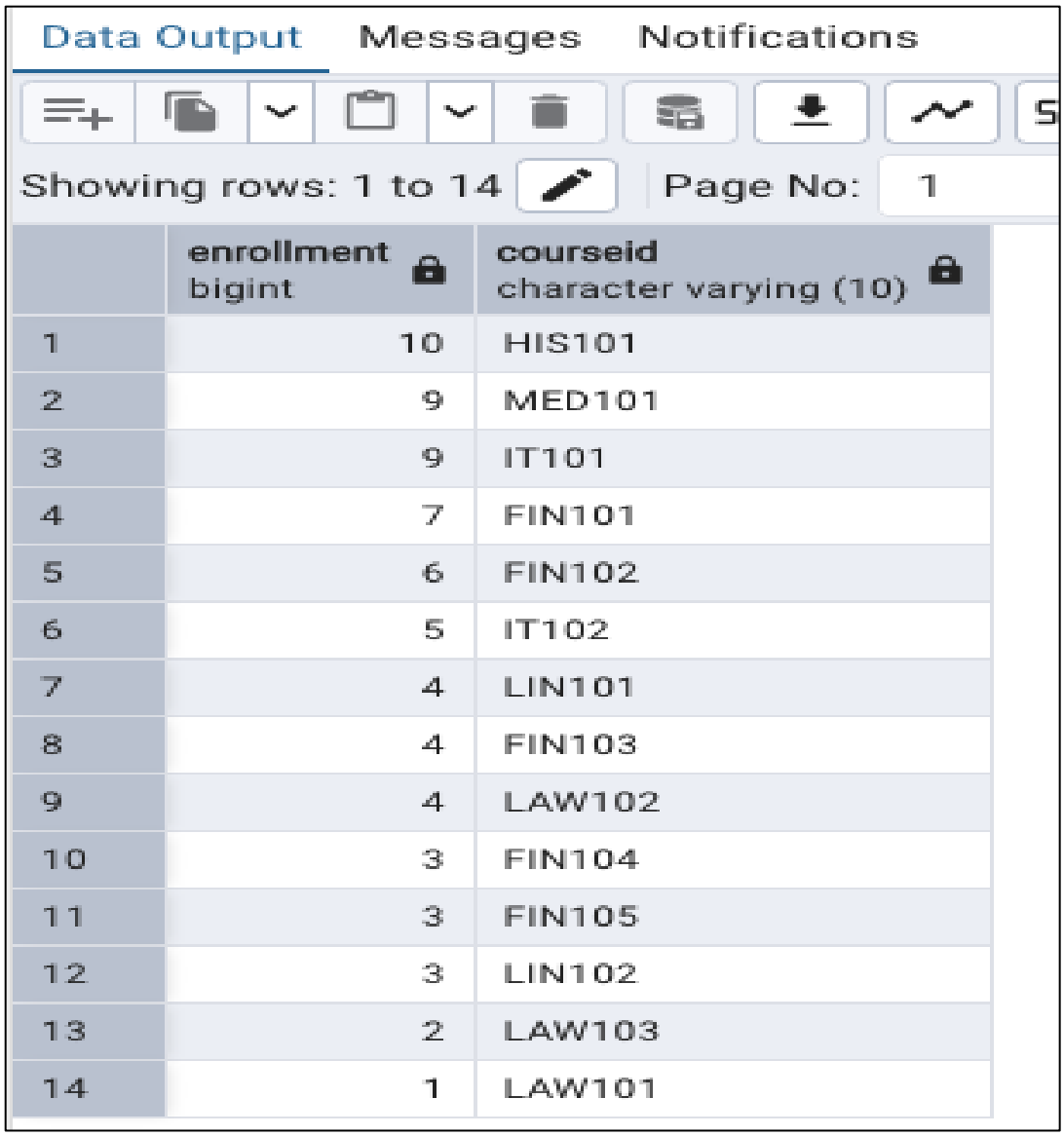
('E014', 14, 'IT101', '2024-08-31'),
('E015', 15, 'MED101', '2024-09-11'),
('E016', 16, 'FIN101', '2024-09-03'),
('E017', 17, 'MED101', '2024-09-06'),
('E018', 18, 'LIN101', '2024-09-02'),
('E019', 19, 'FIN101', '2024-09-12'),
('E020', 20, 'IT101', '2024-09-05'),
('E021', 21, 'HIS101', '2024-08-29'),
('E022', 7, 'LAW102', '2024-09-10'),
('E023', 23, 'FIN101', '2024-09-04'),
('E024', 24, 'LIN101', '2024-09-13'),
('E025', 25, 'FIN102', '2024-09-03'),
('E026', 26, 'IT102', '2024-09-06'),
('E027', 27, 'HIS101', '2024-09-07'),
('E028', 8, 'FIN102', '2024-09-01'),
('E029', 29, 'FIN101', '2024-08-30'),
('E030', 30, 'IT101', '2024-09-08'),
('E031', 2, 'FIN103', '2024-09-04'),
('E032', 14, 'FIN104', '2024-09-04'),
('E033', 21, 'FIN103', '2024-09-01'),
('E034', 8, 'LAW103', '2024-09-01'),
('E035', 8, 'HIS101', '2024-09-02'),
('E036', 31, 'FIN102', '2024-09-07'),
('E037', 32, 'LAW101', '2024-09-01'),
('E038', 33, 'LAW103', '2024-08-30'),
('E039', 34, 'FIN103', '2024-09-08'),
('E040', 35, 'LAW102', '2024-09-05'),
('E041', 36, 'MED101', '2024-08-30'),
('E042', 38, 'HIS101', '2024-09-08'),
('E043', 39, 'MED101', '2024-09-04'),
('E044', 40, 'FIN101', '2024-09-10'),
('E045', 41, 'MED101', '2024-08-28'),
('E046', 42, 'LIN101', '2024-09-03'),
('E047', 43, 'FIN104', '2024-09-13'),
('E048', 44, 'IT101', '2024-09-03'),
('E049', 45, 'IT101', '2024-09-03'),
('E050', 46, 'IT102', '2024-09-10'),
('E051', 47, 'IT101', '2024-09-01'),
('E052', 49, 'MED101', '2024-09-02'),
('E053', 50, 'HIS101', '2024-09-12'),
('E054', 1, 'IT101', '2024-09-05'),
('E055', 20, 'LIN102', '2024-08-29'),
('E056', 42, 'HIS101', '2024-09-10'),

('E057', 13, 'FIN103', '2024-09-04'),
('E058', 31, 'FIN105', '2024-09-13'),
('E059', 15, 'IT101', '2024-09-03'),
('E060', 42, 'LIN102', '2024-09-06'),
('E061', 1, 'FIN102', '2024-09-07'),
('E062', 34, 'IT102', '2024-09-01'),
('E063', 38, 'LIN101', '2024-08-30'),
('E064', 13, 'FIN104', '2024-09-01'),
('E065', 18, 'HIS101', '2024-09-05'),
('E066', 43, 'FIN105', '2024-08-30'),
('E067', 43, 'HIS101', '2024-09-06'),
('E068', 50, 'LAW102', '2024-09-02'),
('E069', 19, 'FIN105', '2024-09-08'),
('E070', 1, 'HIS101', '2024-09-04');

Tamara Gray

1. How many students are currently enrolled in each course?

```
SELECT COUNT(enrollmentid) AS enrollment, courseid
FROM enrollments
GROUP BY courseid
ORDER BY enrollment DESC;
```



Data Output | Messages | Notifications

Showing rows: 1 to 14 | Page No: 1

	enrollment bigint	courseid character varying (10)
1	10	HIS101
2	9	MED101
3	9	IT101
4	7	FIN101
5	6	FIN102
6	5	IT102
7	4	LIN101
8	4	FIN103
9	4	LAW102
10	3	FIN104
11	3	FIN105
12	3	LIN102
13	2	LAW103
14	1	LAW101

Tamara Gray

2. Which students are enrolled in multiple courses, and which courses are they taking?

```
WITH StudentsCTE AS (
                    SELECT studentid
                    FROM enrollments
                    GROUP BY studentid
                    HAVING COUNT(*) > 1
)
SELECT
        s.name ,
        e.courseid,
        c.coursename
FROM StudentsCTE se
JOIN students s
        ON se.studentid = s.studentid
JOIN enrollments e
        ON e.studentid = s.studentid
JOIN courses c
        ON c.courseid = e.courseid
ORDER BY s.name;
```

	name character varying (100)	courseid character varying (10)	coursename character varying (100)
1	Ackelia Beet	HIS101	World History
2	Ackelia Beet	LIN101	Phonetics and Phonology Linguistics
3	Alissa Barrett	FIN104	Financial Management
4	Alissa Barrett	IT101	Introduction to Programming
5	Alleah Martinez	LIN101	Phonetics and Phonology Linguistics
6	Alleah Martinez	LIN102	Syntax and Semantics Linguistics
7	Alleah Martinez	HIS101	World History
8	Faval Green	MED101	Anatomy and Physiology Medicine
9	Faval Green	IT101	Introduction to Programming
10	Fernando Nunez	FIN105	Advance Financial Management
11	Fernando Nunez	FIN101	Corporate Finance
12	Gary Strap	IT102	Database Systems
13	Gary Strap	FIN103	Introduction to Managerial Accounting
14	Jack Fern	LAW103	International Law
15	Jack Fern	IT101	Introduction to Programming
16	Jack Fern	FIN102	Investment Analysis
17	Jack Fern	HIS101	World History
18	James Murphy	HIS101	World History
19	James Murphy	LIN101	Phonetics and Phonology Linguistics
20	Krissan White	LIN102	Syntax and Semantics Linguistics
21	Krissan White	IT101	Introduction to Programming

Tamara Gray

22	Leandra Russell	LAW102	Criminal Law
23	Leandra Russell	HIS101	World History
24	Matthew Dunn	LAW102	Criminal Law
25	Matthew Dunn	FIN101	Corporate Finance
26	Nicki Minaj	FIN102	Investment Analysis
27	Nicki Minaj	FIN103	Introduction to Managerial Accounting
28	Nicki Minaj	FIN104	Financial Management
29	Noah Wilson	IT102	Database Systems
30	Noah Wilson	FIN103	Introduction to Managerial Accounting
31	Olivia Carter	FIN105	Advance Financial Management
32	Olivia Carter	FIN102	Investment Analysis
33	Portia Miller	LAW102	Criminal Law
34	Portia Miller	FIN101	Corporate Finance
35	Rue Robinson	HIS101	World History
36	Rue Robinson	FIN103	Introduction to Managerial Accounting
37	Samantha Brown	FIN101	Corporate Finance
38	Samantha Brown	IT101	Introduction to Programming
39	Samantha Brown	FIN102	Investment Analysis
40	Samantha Brown	HIS101	World History
41	William Scott	HIS101	World History
42	William Scott	FIN105	Advance Financial Management
43	William Scott	FIN104	Financial Management

Tamara Gray

3. What is the total number of students per department across all courses?

```
SELECT COUNT(DISTINCT s.studentid) AS Total_Students_Enrolled,
        s.departmentid,
        d.departmentname
FROM students s
JOIN departments d
        ON s.departmentid = d.departmentid
JOIN enrollments e
        ON s.studentid = e.studentid
GROUP BY s.departmentid , d.departmentname
ORDER BY Total_Students_Enrolled DESC;
```

	total_students_enrolled bigint	departmentid integer	departmentname character varying (100)
1	13	1	Finance
2	11	2	IT
3	9	5	Medicine
4	5	3	History
5	4	6	Linguistics
6	3	4	Law

4. Which courses have the highest number of enrollments?

```
SELECT courseid, COUNT(*) AS Total_Enrollments
FROM enrollments
GROUP BY courseid
ORDER BY Total_Enrollments DESC
LIMIT 1;
```

	courseid character varying (10)	total_enrollments bigint
1	HIS101	10

Tamara Gray

5. Which department has the least number of students?

```
SELECT d.departmentid, d.departmentname, COUNT(*) AS Total_Students
FROM students s
JOIN departments d
        ON s.departmentid = d.departmentid
GROUP BY d.departmentid, d.departmentname
ORDER BY Total_Students ASC
LIMIT 1;
```

	departmentid [PK] integer	departmentname character varying (100)	total_students bigint
1	4	Law	5

6. Are there any students not enrolled in any course?

```
SELECT s.studentid, name
FROM students s
LEFT JOIN enrollments e
        ON s.studentid = e.studentid
WHERE e.studentid IS NULL
ORDER BY name ASC;
```

	studentid [PK] integer	name character varying (100)
1	22	Daedra Phillips
2	28	Elizabeth Wright
3	48	Ethan Collins
4	10	Paul Bogle
5	37	Sophia Nguyen

Tamara Gray

7. How many courses does each student take on average?

```
WITH StudentCourseCount AS (
        SELECT studentid, COUNT(courseid) AS course_count
        FROM enrollments
        GROUP BY studentid
)

SELECT ROUND(AVG(course_count),2) AS Avg_course_taken_per_student
FROM StudentCourseCount;
```

	avg_course_taken_per_student numeric
1	1.56

8. What is the gender distribution of students across courses and instructors?

```
SELECT      COUNT(*) AS gender_count,
            s.gender,
            c.courseid,
            c.coursename,
            i.instructorid,
            i.name
FROM students s
JOIN enrollments e
        ON s.studentid = e.studentid
JOIN courses c
        ON e.courseid = c.courseid
JOIN instructors i
        ON c.instructorid = i.instructorid
GROUP BY
            s.gender,
            c.courseid,
            i.instructorid,
            i.name
ORDER BY s.gender, c.coursename
```

Tamara Gray

gender_count bigint	gender character varying (10)	courseid character varying (10)	coursename character varying (100)	instructorid integer	name character varying (100)
1	FEMALE	FIN105	Advance Financial Management	2	Kelly Prance
5	FEMALE	MED101	Anatomy and Physiology Medicine	7	Sheena Bailey
3	FEMALE	FIN101	Corporate Finance	1	Joe Brown
3	FEMALE	LAW102	Criminal Law	6	Percy Jones
2	FEMALE	IT102	Database Systems	4	John Snow
2	FEMALE	FIN104	Financial Management	2	Kelly Prance
1	FEMALE	LAW103	International Law	6	Percy Jones
2	FEMALE	FIN103	Introduction to Managerial Accounting	1	Joe Brown
6	FEMALE	IT101	Introduction to Programming	3	Steve Jobs
5	FEMALE	FIN102	Investment Analysis	1	Joe Brown
2	FEMALE	LIN101	Phonetics and Phonology Linguistics	8	Ackera Sommons
3	FEMALE	LIN102	Syntax and Semantics Linguistics	9	Kandace Rowe
6	FEMALE	HIS101	World History	5	Steve Whitaker
2	MALE	FIN105	Advance Financial Management	2	Kelly Prance
4	MALE	MED101	Anatomy and Physiology Medicine	7	Sheena Bailey
1	MALE	LAW101	Constitutional Law	6	Percy Jones
4	MALE	FIN101	Corporate Finance	1	Joe Brown
1	MALE	LAW102	Criminal Law	6	Percy Jones
3	MALE	IT102	Database Systems	4	John Snow
1	MALE	FIN104	Financial Management	2	Kelly Prance
1	MALE	LAW103	International Law	6	Percy Jones
2	MALE	FIN103	Introduction to Managerial Accounting	1	Joe Brown
3	MALE	IT101	Introduction to Programming	3	Steve Jobs

24	1	MALE	FIN102	Investment Analysis	1	Joe Brown
25	2	MALE	LIN101	Phonetics and Phonology Linguistics	8	Ackera Sommons
26	4	MALE	HIS101	World History	5	Steve Whitaker

Tamara Gray

9. Which course has the highest number of male or female students enrolled?

```
SELECT
        c.courseid,
        c.coursename,
        s.gender,
        COUNT(*) AS num_enrolled
FROM courses c
JOIN enrollments e
        ON c.courseid = e.courseid
JOIN students s
        ON s.studentid = e.studentid
GROUP BY c.courseid, c.coursename, s.gender
ORDER BY num_enrolled DESC
LIMIT 1;
```

	courseid character varying (10)	coursename character varying (100)	gender character varying (10)	num_enrolled bigint
1	HIS101	World History	FEMALE	6