Case Study - college students

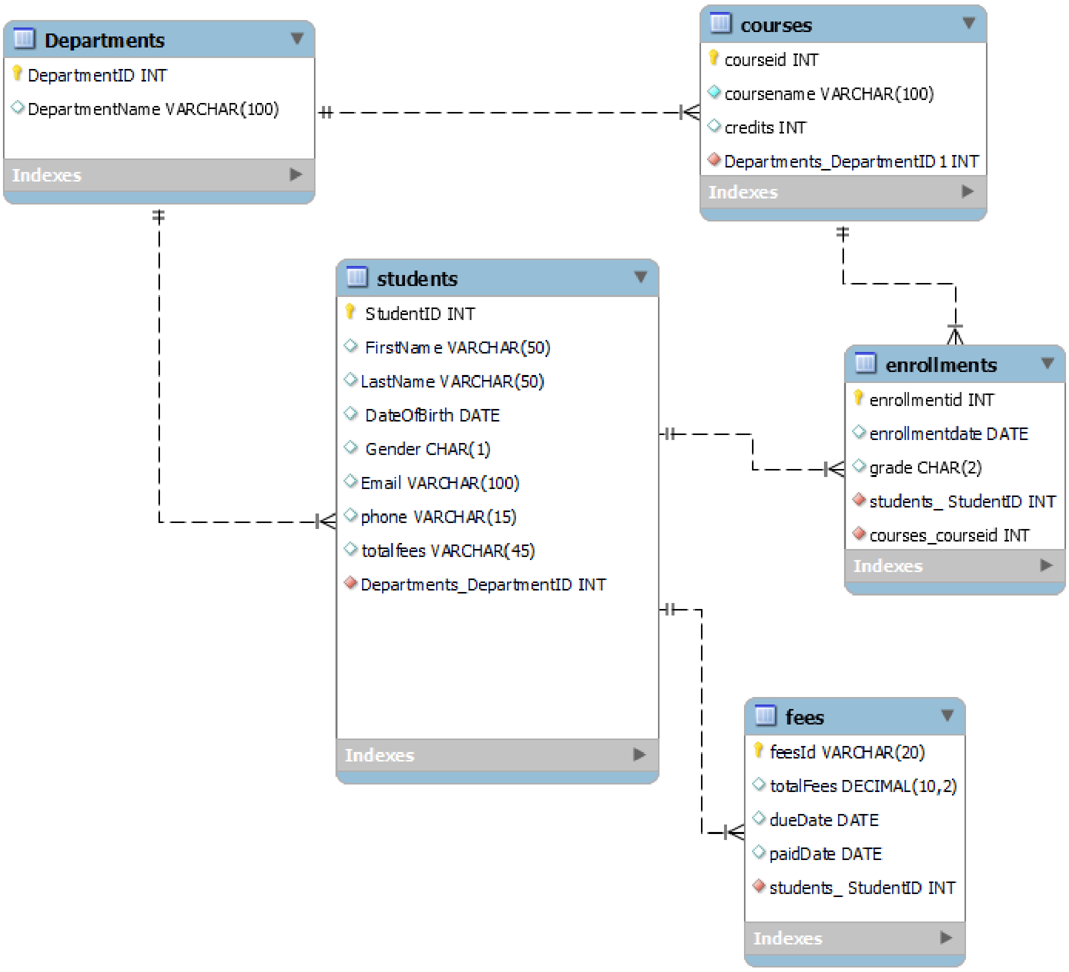
Introduction:

This case study examines how a colleges manages its departments, students, courses, enrollments, and fees using a database. It looks at how these different parts are connected and organized to keep track of student information, course enrollments, and payments. The goal is to show how databases help colleges stay organized and run smoothly. The study also aims to suggest ways to improve the database design for better data management.

Problem Statement:

Develop a database system for a college to efficiently manage departments, students, courses, enrollments, and fees. Ensure the system accurately tracks student enrollments, course details, and financial transactions to support administrative tasks and improve organisational efficiency.

Er diagram



Dataset

Database creation:

```
create database college_students;
```

Use Database:

```
use college_students;
```

Creating the Departments table

```
CREATE TABLE Departments ( DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100) );
```

Insert data in Departments table

```
INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1001, 'Computer Science'),(1002, 'Mathematics'),(1003, 'Physics'),

(1004, 'Chemistry'),(1005, 'Biology');
```

Creating the Students table

```
CREATE TABLE Students (

StudentID INT PRIMARY KEY,FirstName VARCHAR(50),

LastName VARCHAR(50),   DateOfBirth DATE,Gender CHAR(1),

DepartmentID INT, Email VARCHAR(100),  Phone VARCHAR(15),

TotalFees DECIMAL(10, 2),

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID));
```

Insert data in Students table

```
INSERT INTO Students (StudentID, FirstName, LastName, DateOfBirth, Gender, DepartmentID, Email, Phone, TotalFees) VALUES

(332500, 'Alice', 'Smith', '2000-01-01', 'F', 1001, 'alice.smith@example.com', '123-456-7890', 5000.00),

(332501, 'Bob', 'Johnson', '1999-02-02', 'M', 1001, 'bob.johnson@example.com', '123-456-7891', 5100.00),
(332502, 'Carol', 'Williams', '2001-03-03', 'F', 1002,
'carol.williams@example.com', '123-456-7892', 5200.00),

(332503, 'Dave', 'Brown', '2000-04-04', 'M', 1002, 'dave.brown@example.com',
'123-456-7893', 5300.00),

(332504, 'Eve', 'Jones', '1998-05-05', 'F', 1003, 'eve.jones@example.com',
'123-456-7894', 5400.00),

(332505, 'Frank', 'Garcia', '1997-06-06', 'M', 1003, 'frank.garcia@example.com',
'123-456-7895', 5500.00),

(332506, 'Grace', 'Martinez', '2001-07-07', 'F', 1004,
'grace.martinez@example.com', '123-456-7896', 5600.00),

(332507, 'Hank', 'Rodriguez', '2000-08-08', 'M', 1004,
'hank.rodriguez@example.com', '123-456-7897', 5700.00),

(332508, 'Ivy', 'Wilson', '1999-09-09', 'F', 1005, 'ivy.wilson@example.com',
'123-456-7898', 5800.00),

(332509, 'Jack', 'Lee', '1998-10-10', 'M', 1005, 'jack.lee@example.com',
'123-456-7899', 5900.00),

(332510, 'Ken', 'White', '2000-11-11', 'M', 1001, 'ken.white@example.com',
'123-456-7900', 6000.00),

(332511, 'Luna', 'Harris', '2001-12-12', 'F', 1001, 'luna.harris@example.com',
'123-456-7901', 6100.00),

(332512, 'Mike', 'Clark', '1997-01-01', 'M', 1002, 'mike.clark@example.com',
'123-456-7902', 6200.00),

(332513, 'Nina', 'Lewis', '1998-02-02', 'F', 1002, 'nina.lewis@example.com',
'123-456-7903', 6300.00),

(332514, 'Oscar', 'Robinson', '1999-03-03', 'M', 1003,
'oscar.robinson@example.com', '123-456-7904', 6400.00),

(332515, 'Pam', 'Walker', '2000-04-04', 'F', 1003, 'pam.walker@example.com',
'123-456-7905', 6500.00),

(332516, 'Quinn', 'Hall', '2001-05-05', 'M', 1004, 'quinn.hall@example.com',
'123-456-7906', 6600.00),

(332517, 'Rose', 'Young', '2002-06-06', 'F', 1004, 'rose.young@example.com',
'123-456-7907', 6700.00),

(332518, 'Sam', 'Allen', '1997-07-07', 'M', 1005, 'sam.allen@example.com',
'123-456-7908', 6800.00),
(332519, 'Tina', 'King', '1998-08-08', 'F', 1005, 'tina.king@example.com', '123-456-7909', 6900.00),

(332520, 'Uma', 'Lopez', '1999-09-09', 'F', 1001, 'uma.lopez@example.com', '123-456-7910', 5000.00),

(332521, 'Victor', 'Gonzalez', '2000-10-10', 'M', 1001, 'victor.gonzalez@example.com', '123-456-7911', 5100.00),

(332522, 'Wendy', 'Carter', '2001-11-11', 'F', 1002, 'wendy.carter@example.com', '123-456-7912', 5200.00),

(332523, 'Xavier', 'Mitchell', '1997-12-12', 'M', 1002, 'xavier.mitchell@example.com', '123-456-7913', 5300.00),

(332524, 'Yara', 'Perez', '1998-01-01', 'F', 1003, 'yara.perez@example.com', '123-456-7914', 5400.00),

(332525, 'Zane', 'Evans', '1999-02-02', 'M', 1003, 'zane.evans@example.com', '123-456-7915', 5500.00),

(332526, 'Aiden', 'Turner', '2000-03-03', 'M', 1004, 'aiden.turner@example.com', '123-456-7916', 5600.00),

(332527, 'Bella', 'Diaz', '2001-04-04', 'F', 1004, 'bella.diaz@example.com', '123-456-7917', 5700.00),

(332528, 'Cameron', 'Parker', '1997-05-05', 'M', 1005, 'cameron.parker@example.com', '123-456-7918', 5800.00),

(332529, 'Diana', 'Sanchez', '1998-06-06', 'F', 1005, 'diana.sanchez@example.com', '123-456-7919', 5900.00),

(332530, 'Ethan', 'Reed', '1996-07-07', 'M', 1001, 'ethan.reed@example.com', '123-456-7920', 6000.00);
```

Creating the Courses table

```
CREATE TABLE Courses (

  CourseID INT PRIMARY KEY,

  CourseName VARCHAR(100),

   Credits INT, DepartmentID INT,

  FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID));
```

Insert data in Courses table

```
INSERT INTO Courses (CourseID, CourseName, Credits, DepartmentID) VALUES

(101, 'Introduction to Computer Science', 4, 1001),

(102, 'Data Structures and Algorithms', 3, 1001),

(103, 'Database Systems', 3, 1001),(104, 'Calculus I', 4, 1002),

(105, 'Linear Algebra', 3, 1002),(106, 'Differential Equations', 3, 1002),

(107, 'General Physics I', 4, 1003),(108, 'Quantum Mechanics', 3, 1003),

(109, 'Organic Chemistry', 4, 1004),(110, 'Molecular Biology', 4, 1005);
```

Creating the Enrollments table

```
CREATE TABLE Enrollments (

   EnrollmentID INT PRIMARY KEY,   StudentID INT,

   CourseID INT,EnrollmentDate DATE, Grade CHAR(2),

   FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

   FOREIGN KEY (CourseID) REFERENCES Courses(CourseID));
```

Insert data in Enrollments table

```
INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, EnrollmentDate, Grade) VALUES

(1, 332500, 103, '2022-09-01', 'A'),(2, 332501, 109, '2022-09-02', 'B'),

(3, 332502, 107, '2022-09-03', 'A'),(4, 332503, 104, '2022-09-04', 'C'),

(5, 332504, 101, '2022-09-05', 'B'),(6, 332505, 110, '2022-09-06', 'A'),

(7, 332506, 105, '2022-09-07', 'B'),(8, 332507, 108, '2022-09-08', 'C'),

(9, 332508, 102, '2022-09-09', 'A'),(10, 332509, 106, '2022-09-10', 'B'),

(11, 332510, 104, '2022-09-11', 'A'),(12, 332511, 103, '2022-09-12', 'B'),

(13, 332512, 108, '2022-09-13', 'C'),(14, 332513, 105, '2022-09-14', 'A'),

(15, 332514, 101, '2022-09-15', 'B'),(16, 332515, 110, '2022-09-16', 'A'),

(17, 332516, 107, '2022-09-17', 'C'),(18, 332517, 109, '2022-09-18', 'B'),
(19, 332518, 106, '2022-09-19', 'A'),(20, 332519, 102, '2022-09-20', 'B'),
(21, 332520, 104, '2022-09-21', 'C'),(22, 332521, 103, '2022-09-22', 'A'),
(23, 332522, 108, '2022-09-23', 'B'),(24, 332523, 105, '2022-09-24', 'A'),
(25, 332524, 101, '2022-09-25', 'B'),(26, 332525, 110, '2022-09-26', 'C'),
(27, 332526, 107, '2022-09-27', 'A'),(28, 332527, 109, '2022-09-28', 'B'),
(29, 332528, 106, '2022-09-29', 'A'),(30, 332529, 102, '2022-09-30', 'B'),
(31, 332530, 103, '2022-10-01', 'C');
```

Creating the Fees table

```
CREATE TABLE Fees (
   FeeID VARCHAR(20) PRIMARY KEY, StudentID INT,
   TotalFees DECIMAL(10, 2), PaidAmount DECIMAL(10, 2),
   DueDate DATE, PaidDate DATE,
   FOREIGN KEY (StudentID) REFERENCES Students(StudentID));
```

Insert values in Fees table

```
INSERT INTO Fees (FeeID, StudentID, TotalFees, PaidAmount, DueDate,
PaidDate) VALUES
('cb19s01', 332500, 40000.00, 35000.00, '2022-09-20', '2022-09-10'),
('cb19s02', 332501, 40000.00, 37500.00, '2022-09-20', '2022-09-11'),
('cb19s03', 332502, 40000.00, 30000.00, '2022-09-20', NULL),
('cb19s04', 332503, 40000.00, 36000.00, '2022-09-20', '2022-09-12'),
('cb19s05', 332504, 40000.00, 33000.00, '2022-09-20', '2022-09-13'),
('cb19s06', 332505, 40000.00, 39000.00, '2022-09-20', NULL),
('cb19s07', 332506, 40000.00, 34000.00, '2022-09-20', '2022-09-15'),
('cb19s08', 332507, 40000.00, 38000.00, '2022-09-20', '2022-09-14'),
('cb19s09', 332508, 40000.00, 29500.00, '2022-09-20', '2022-09-16'),
('cb19s10', 332509, 40000.00, 31000.00, '2022-09-20', NULL),
('cb19s11', 332510, 40000.00, 32500.00, '2022-09-20', '2022-09-17'),
('cb19s12', 332511, 40000.00, 40000.00, '2022-09-20', '2022-09-18'),

('cb19s13', 332512, 40000.00, 33500.00, '2022-09-20', NULL),

('cb19s14', 332513, 40000.00, 37000.00, '2022-09-20', '2022-09-19'),

('cb19s15', 332514, 40000.00, 36000.00, '2022-09-20', '2022-09-20'),

('cb19s16', 332515, 40000.00, 34000.00, '2022-09-20', NULL),

('cb19s17', 332516, 40000.00, 37500.00, '2022-09-20', '2022-09-10'),

('cb19s18', 332517, 40000.00, 30000.00, '2022-09-20', '2022-09-11'),

('cb19s19', 332518, 40000.00, 38000.00, '2022-09-20', NULL),

('cb19s20', 332519, 40000.00, 39000.00, '2022-09-20', '2022-09-12');
```

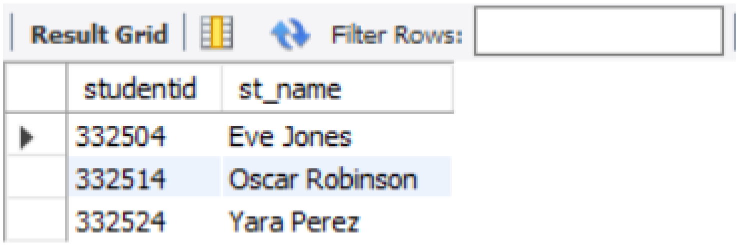
To view tabel and data

```
select*from students ;

select*from departments ;

select*from courses ;

select*from enrollments ;

select*from fees;
```

Case Study Questions & Answers:

1. To obtain the names of students enrolled in a particular course.

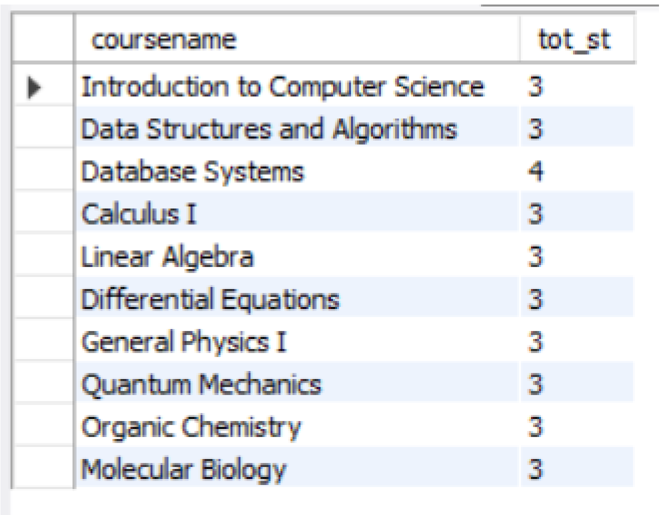
```
select

s.studentid,CONCAT(s.FirstName, ' ', s.LastName) AS st_name

from students s join enrollments e on s.studentid=e.studentid

where courseid=101;
```



studentid	st_name
332504	Eve Jones
332514	Oscar Robinson
332524	Yara Perez

2. To obtain the count of students enrolled in each course

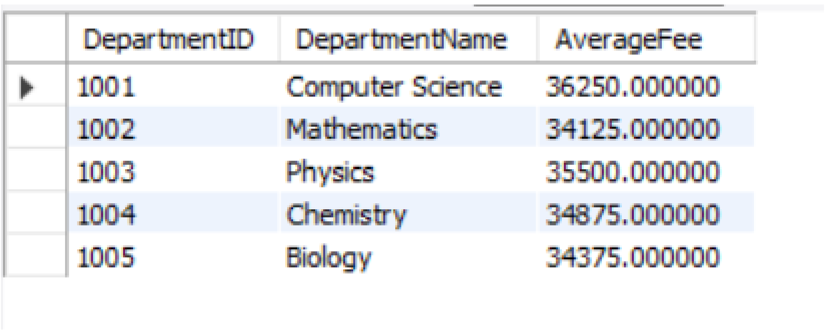
```
Select

 c.coursename,count(e.courseid) as tot_st

from courses c

JOIN Enrollments e

ON c.CourseID = e.CourseID

 group by coursename;
```



coursename	tot_st
Introduction to Computer Science	3
Data Structures and Algorithms	3
Database Systems	4
Calculus I	3
Linear Algebra	3
Differential Equations	3
General Physics I	3
Quantum Mechanics	3
Organic Chemistry	3
Molecular Biology	3

3. What is the average fee paid by students in each department

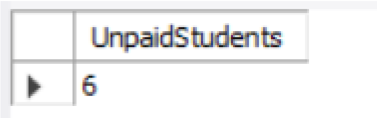
```
SELECT

 d.DepartmentID,  d.DepartmentName,

 AVG(f.PaidAmount) AS AverageFee

FROM Departments d

JOIN Students s ON d.DepartmentID = s.DepartmentID

JOIN Fees f ON s.StudentID = f.StudentID

GROUP BY d.DepartmentID, d.DepartmentName;
```



	DepartmentID	DepartmentName	AverageFee
▶	1001	Computer Science	36250.000000
	1002	Mathematics	34125.000000
	1003	Physics	35500.000000
	1004	Chemistry	34875.000000
	1005	Biology	34375.000000

4. **How many students have unpaid fees**

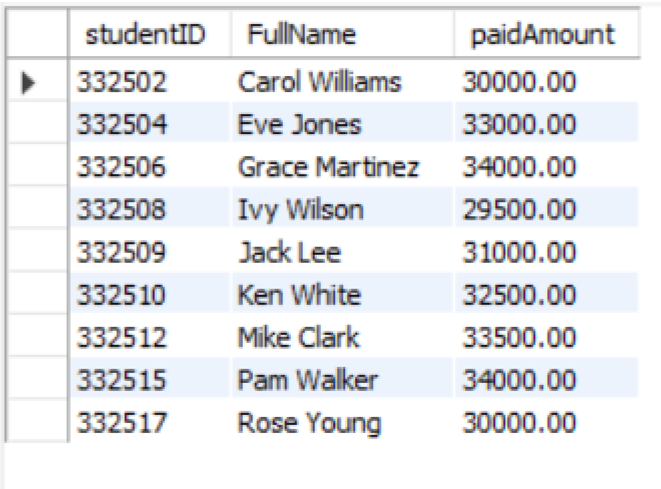
```
SELECT COUNT(DISTINCT StudentID) AS UnpaidStudents

FROM Fees

WHERE paidDate IS NULL;
```



	UnpaidStudents
▶	6

5. list the Student IDs and full names of students that paid fees under 3500

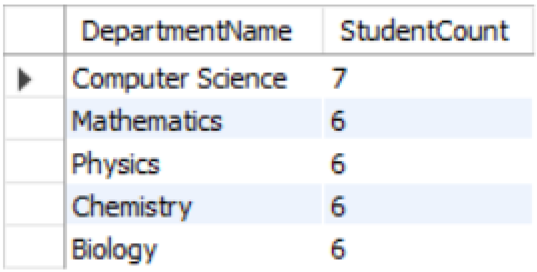
```
SELECT

s.studentID ,CONCAT(s.FirstName, ' ', s.LastName) AS FullName, f.paidAmount

FROM Students s

JOIN Fees f ON s.StudentID = f.StudentID

WHERE f.paidAmount < 35000

order by studentid;
```



	studentID	FullName	paidAmount
▶	332502	Carol Williams	30000.00
	332504	Eve Jones	33000.00
	332506	Grace Martinez	34000.00
	332508	Ivy Wilson	29500.00
	332509	Jack Lee	31000.00
	332510	Ken White	32500.00
	332512	Mike Clark	33500.00
	332515	Pam Walker	34000.00
	332517	Rose Young	30000.00

6. To find students count of each department

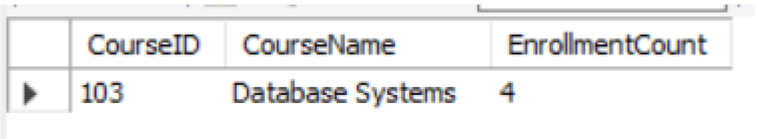
```
SELECT d.DepartmentName, COUNT(s.StudentID) AS StudentCount

FROM Students s

JOIN Departments d ON s.DepartmentID = d.DepartmentID

GROUP BY d.DepartmentName;
```



	DepartmentName	StudentCount
▶	Computer Science	7
	Mathematics	6
	Physics	6
	Chemistry	6
	Biology	6

7. Which course has the highest number of enrollments

```
SELECT c.CourseID, c.CourseName, COUNT(e.StudentID) AS EnrollmentCount

FROM Courses c

JOIN Enrollments e

ON c.CourseID = e.CourseID

GROUP BY c.CourseID, c.CourseName

ORDER BY EnrollmentCount DESC LIMIT 1;
```

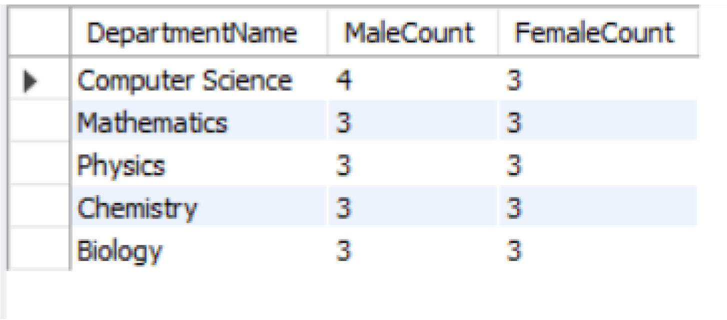


	CourseID	CourseName	EnrollmentCount
▶	103	Database Systems	4

8. To count the number of male and female students in each department

```
SELECT

d.DepartmentName,

SUM(CASE WHEN s.Gender = 'M' THEN 1 ELSE 0 END) AS MaleCount,

 SUM(CASE WHEN s.Gender = 'F' THEN 1 ELSE 0 END) AS FemaleCount

FROM Students s
```

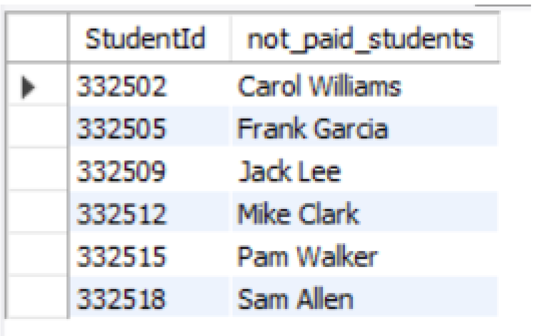
```
JOIN Departments d

ON s.DepartmentID = d.DepartmentID

GROUP BY d.DepartmentName;
```



DepartmentName	MaleCount	FemaleCount
Computer Science	4	3
Mathematics	3	3
Physics	3	3
Chemistry	3	3
Biology	3	3

9.To find the students fees not paid

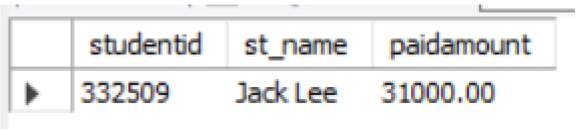
```
SELECT

s.StudentId,

CONCAT(s.FirstName, ' ', s.LastName) AS not_paid_students

FROM Students s

JOIN Fees f

ON s.StudentID = f.StudentID

WHERE f.PaidDate IS  NULL;
```



StudentId	not_paid_students
332502	Carol Williams
332505	Frank Garcia
332509	Jack Lee
332512	Mike Clark
332515	Pam Walker
332518	Sam Allen

10 .To get the fees paid detail of specific student id

```
select

s.studentid,

CONCAT(s.FirstName, ' ', s.LastName) AS st_name,
f.paidamount

from students s

join fees f

on s.studentid=f.studentid

where s.studentid=332509;
```



	studentid	st_name	paidamount
▶	332509	Jack Lee	31000.00