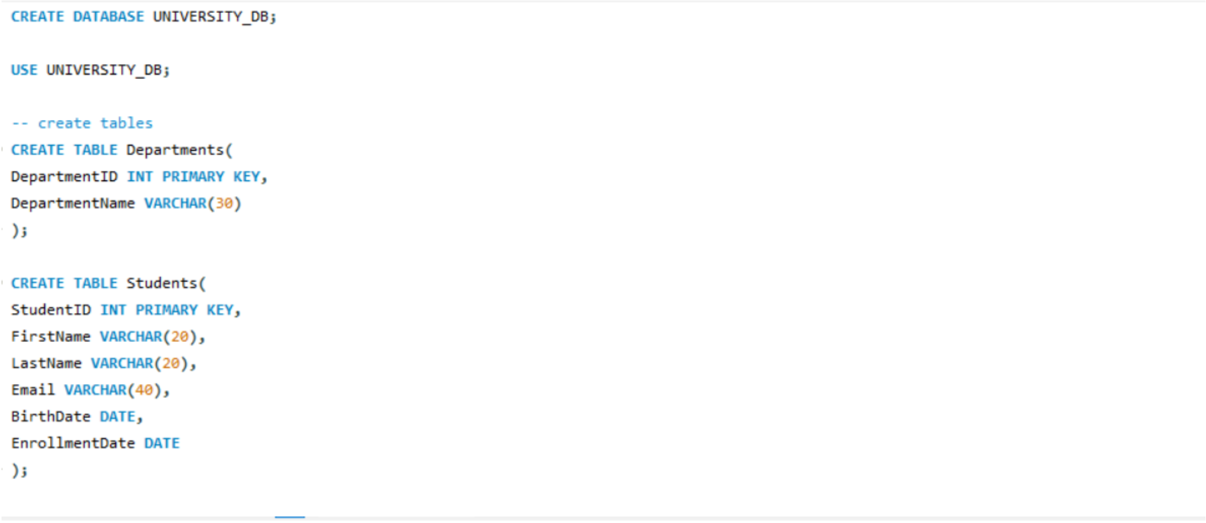
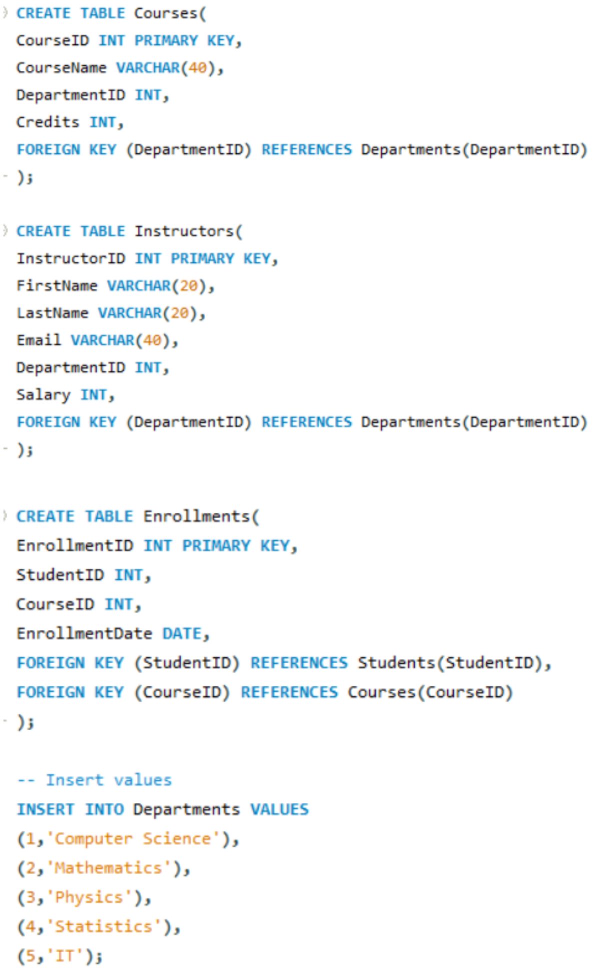
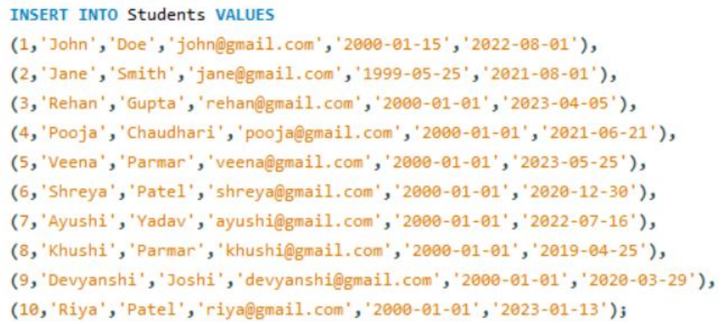


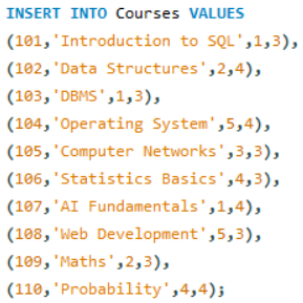
```
CREATE DATABASE UNIVERSITY_DB;

USE UNIVERSITY_DB;

-- create tables
CREATE TABLE Departments(
DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(30)
);

CREATE TABLE Students(
StudentID INT PRIMARY KEY,
FirstName VARCHAR(20),
LastName VARCHAR(20),
Email VARCHAR(40),
BirthDate DATE,
EnrollmentDate DATE
);
```



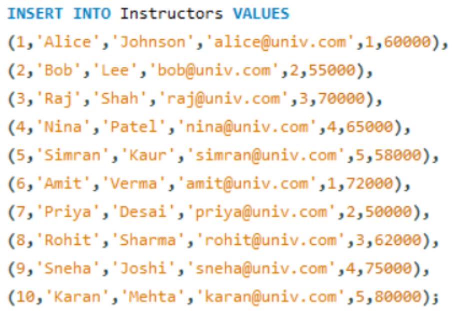
```
CREATE TABLE Courses(
CourseID INT PRIMARY KEY,
CourseName VARCHAR(40),
DepartmentID INT,
Credits INT,
FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);

CREATE TABLE Instructors(
InstructorID INT PRIMARY KEY,
FirstName VARCHAR(20),
LastName VARCHAR(20),
Email VARCHAR(40),
DepartmentID INT,
Salary INT,
FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);

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

-- Insert values
INSERT INTO Departments VALUES
(1,'Computer Science'),
(2,'Mathematics'),
(3,'Physics'),
(4,'Statistics'),
(5,'IT');
```



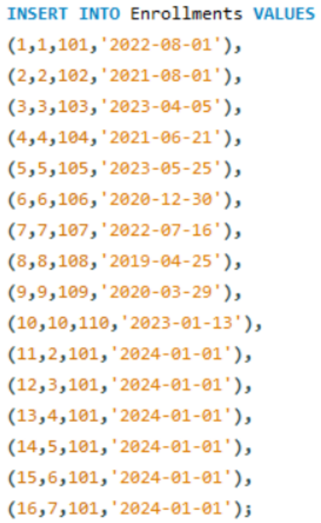
```sql
INSERT INTO Students VALUES
(1,'John','Doe','john@gmail.com','2000-01-15','2022-08-01'),
(2,'Jane','Smith','jane@gmail.com','1999-05-25','2021-08-01'),
(3,'Rehan','Gupta','rehan@gmail.com','2000-01-01','2023-04-05'),
(4,'Pooja','Chaudhari','pooja@gmail.com','2000-01-01','2021-06-21'),
(5,'Veena','Parmar','veena@gmail.com','2000-01-01','2023-05-25'),
(6,'Shreya','Patel','shreya@gmail.com','2000-01-01','2020-12-30'),
(7,'Ayushi','Yadav','ayushi@gmail.com','2000-01-01','2022-07-16'),
(8,'Khushi','Parmar','khushi@gmail.com','2000-01-01','2019-04-25'),
(9,'Devyanshi','Joshi','devyanshi@gmail.com','2000-01-01','2020-03-29'),
(10,'Riya','Patel','riya@gmail.com','2000-01-01','2023-01-13');
```



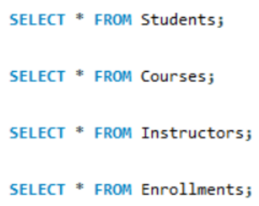
```sql
INSERT INTO Courses VALUES
(101,'Introduction to SQL',1,3),
(102,'Data Structures',2,4),
(103,'DBMS',1,3),
(104,'Operating System',5,4),
(105,'Computer Networks',3,3),
(106,'Statistics Basics',4,3),
(107,'AI Fundamentals',1,4),
(108,'Web Development',5,3),
(109,'Maths',2,3),
(110,'Probability',4,4);
```



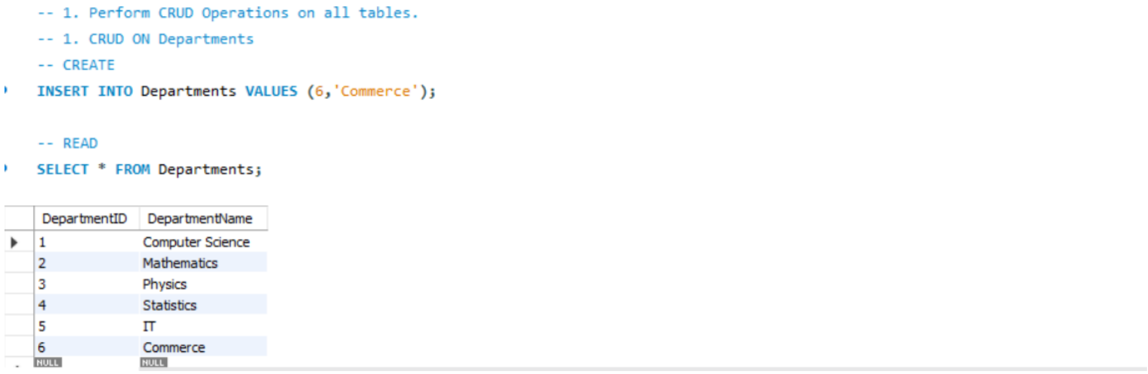
```sql
INSERT INTO Instructors VALUES
(1,'Alice','Johnson','alice@univ.com',1,60000),
(2,'Bob','Lee','bob@univ.com',2,55000),
(3,'Raj','Shah','raj@univ.com',3,70000),
(4,'Nina','Patel','nina@univ.com',4,65000),
(5,'Simran','Kaur','simran@univ.com',5,58000),
(6,'Amit','Verma','amit@univ.com',1,72000),
(7,'Priya','Desai','priya@univ.com',2,50000),
(8,'Rohit','Sharma','rohit@univ.com',3,62000),
(9,'Sneha','Joshi','sneha@univ.com',4,75000),
(10,'Karan','Mehta','karan@univ.com',5,80000);
```



```sql
INSERT INTO Enrollments VALUES
(1,1,101,'2022-08-01'),
(2,2,102,'2021-08-01'),
(3,3,103,'2023-04-05'),
(4,4,104,'2021-06-21'),
(5,5,105,'2023-05-25'),
(6,6,106,'2020-12-30'),
(7,7,107,'2022-07-16'),
(8,8,108,'2019-04-25'),
(9,9,109,'2020-03-29'),
(10,10,110,'2023-01-13'),
(11,2,101,'2024-01-01'),
(12,3,101,'2024-01-01'),
(13,4,101,'2024-01-01'),
(14,5,101,'2024-01-01'),
(15,6,101,'2024-01-01'),
(16,7,101,'2024-01-01');
```

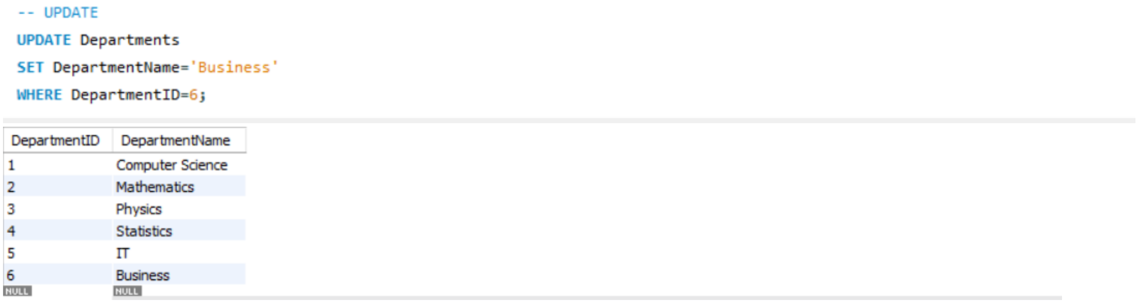


```sql
SELECT * FROM Students;

SELECT * FROM Courses;

SELECT * FROM Instructors;

SELECT * FROM Enrollments;
```



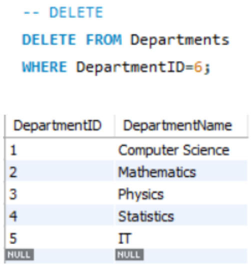
```sql
-- 1. Perform CRUD Operations on all tables.
-- 1. CRUD ON Departments
-- CREATE
INSERT INTO Departments VALUES (6,'Commerce');

-- READ
SELECT * FROM Departments;
```

DepartmentID	DepartmentName
1	Computer Science
2	Mathematics
3	Physics
4	Statistics
5	IT
6	Commerce
NULL	NULL



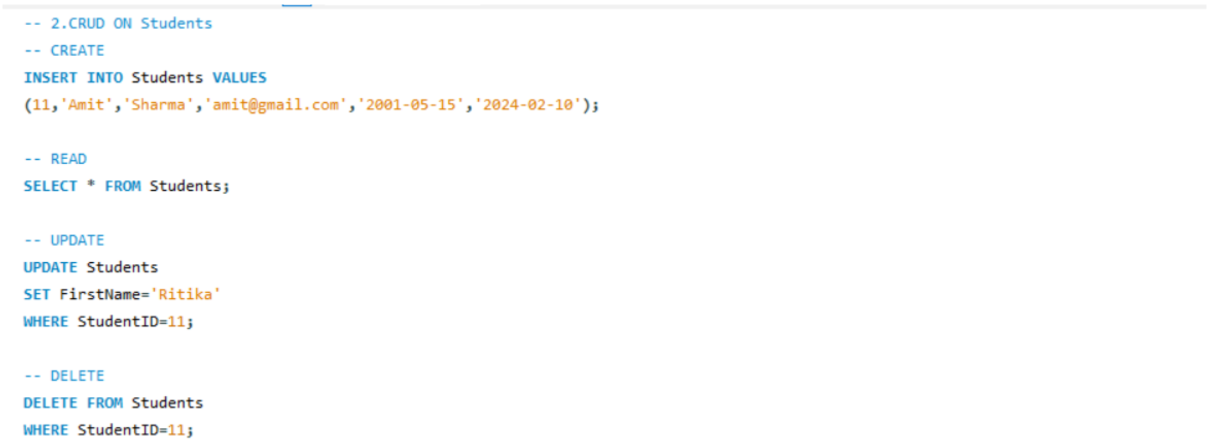
```sql
-- UPDATE
UPDATE Departments
SET DepartmentName='Business'
WHERE DepartmentID=6;
```

DepartmentID	DepartmentName
1	Computer Science
2	Mathematics
3	Physics
4	Statistics
5	IT
6	Business
NULL	NULL

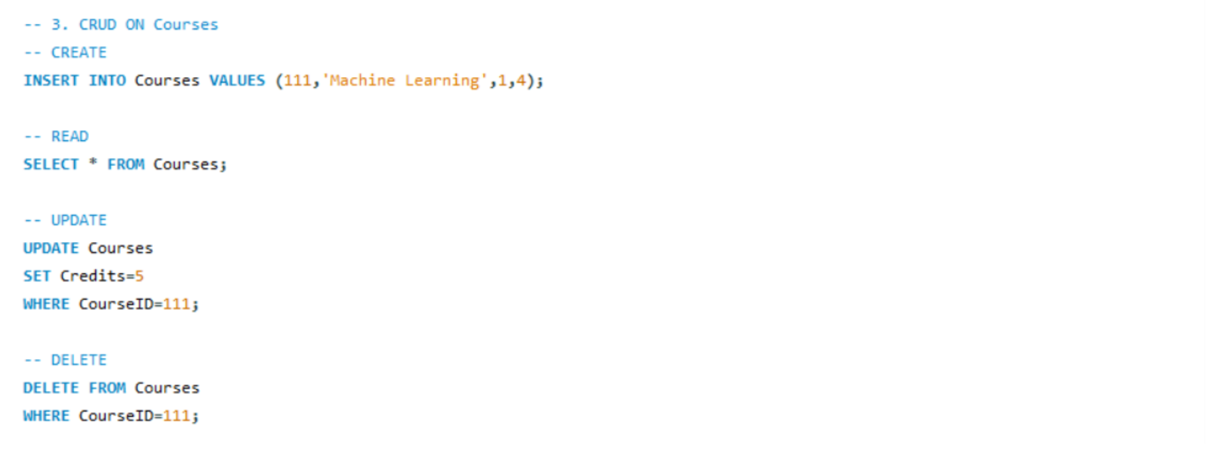


```sql
-- DELETE
DELETE FROM Departments
WHERE DepartmentID=6;
```

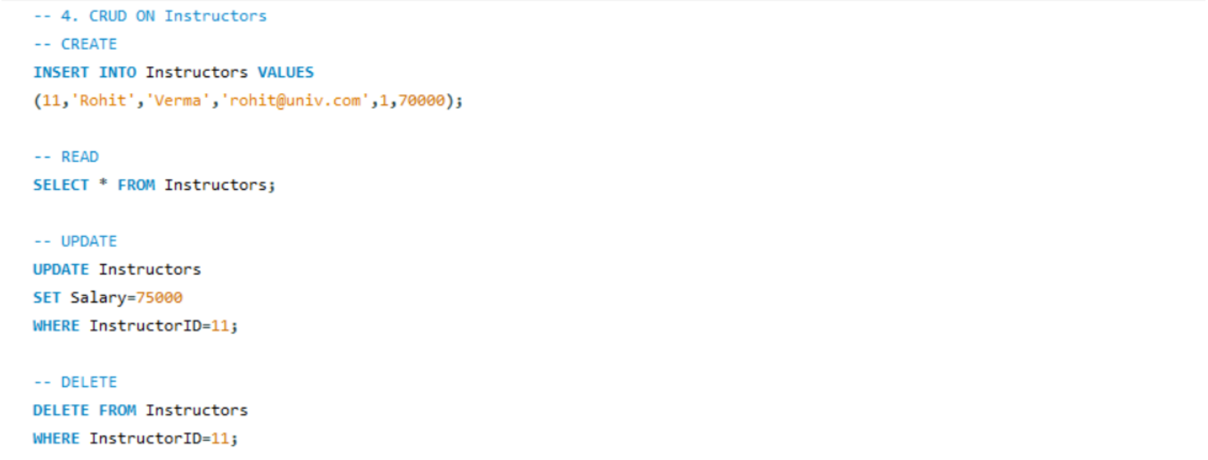
DepartmentID	DepartmentName
1	Computer Science
2	Mathematics
3	Physics
4	Statistics
5	IT
NULL	NULL



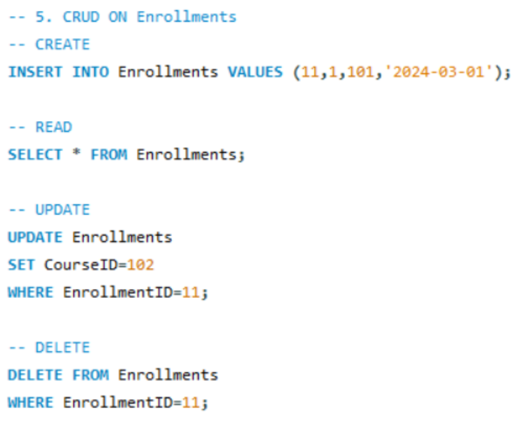
```sql
-- 2.CRUD ON Students
-- CREATE
INSERT INTO Students VALUES
(11,'Amit','Sharma','amit@gmail.com','2001-05-15','2024-02-10');

-- READ
SELECT * FROM Students;

-- UPDATE
UPDATE Students
SET FirstName='Ritika'
WHERE StudentID=11;

-- DELETE
DELETE FROM Students
WHERE StudentID=11;
```



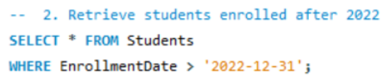
```
-- 3. CRUD ON Courses
-- CREATE
INSERT INTO Courses VALUES (111,'Machine Learning',1,4);

-- READ
SELECT * FROM Courses;

-- UPDATE
UPDATE Courses
SET Credits=5
WHERE CourseID=111;

-- DELETE
DELETE FROM Courses
WHERE CourseID=111;
```



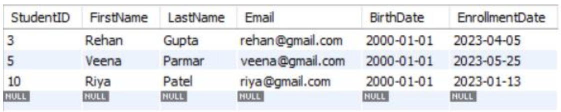
```
-- 4. CRUD ON Instructors
-- CREATE
INSERT INTO Instructors VALUES
(11,'Rohit','Verma','rohit@univ.com',1,70000);

-- READ
SELECT * FROM Instructors;

-- UPDATE
UPDATE Instructors
SET Salary=75000
WHERE InstructorID=11;

-- DELETE
DELETE FROM Instructors
WHERE InstructorID=11;
```



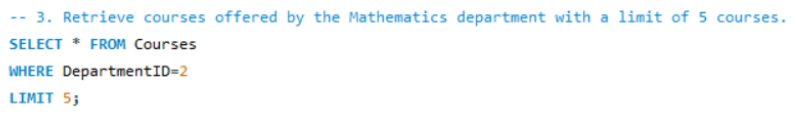
```
-- 5. CRUD ON Enrollments
-- CREATE
INSERT INTO Enrollments VALUES (11,1,101,'2024-03-01');

-- READ
SELECT * FROM Enrollments;

-- UPDATE
UPDATE Enrollments
SET CourseID=102
WHERE EnrollmentID=11;

-- DELETE
DELETE FROM Enrollments
WHERE EnrollmentID=11;
```



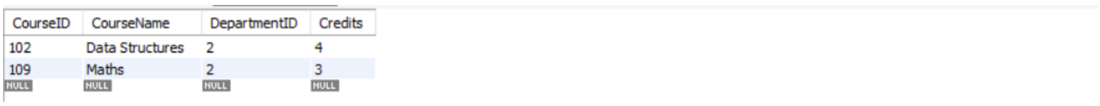
```
--  2. Retrieve students enrolled after 2022
SELECT * FROM Students
WHERE EnrollmentDate > '2022-12-31';
```



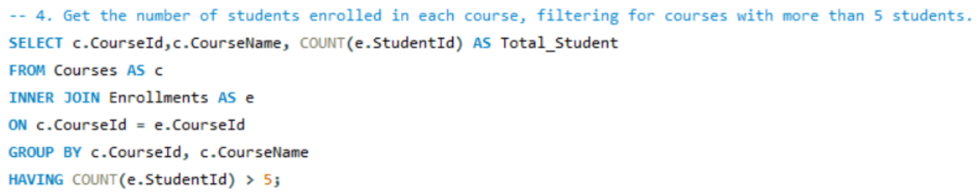
StudentID	FirstName	LastName	Email	BirthDate	EnrollmentDate
3	Rehan	Gupta	rehan@gmail.com	2000-01-01	2023-04-05
5	Veena	Parmar	veena@gmail.com	2000-01-01	2023-05-25
10	Riya	Patel	riya@gmail.com	2000-01-01	2023-01-13
NULL	NULL	NULL	NULL	NULL	NULL



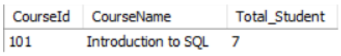
```sql
-- 3. Retrieve courses offered by the Mathematics department with a limit of 5 courses.
SELECT * FROM Courses
WHERE DepartmentID=2
LIMIT 5;
```



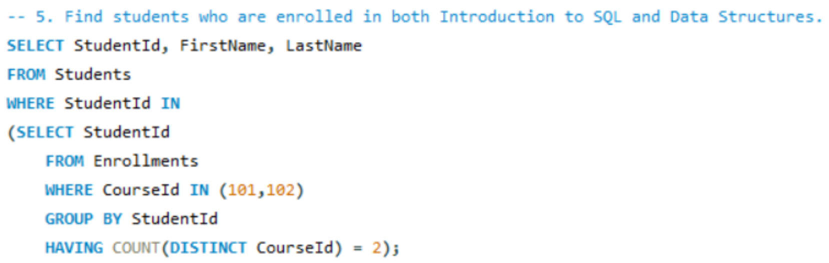
CourseID	CourseName	DepartmentID	Credits
102	Data Structures	2	4
109	Maths	2	3
NULL	NULL	NULL	NULL



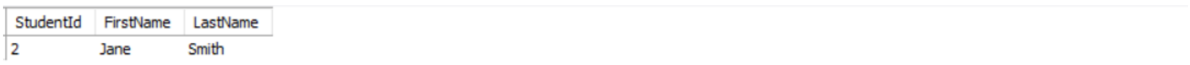
```sql
-- 4. Get the number of students enrolled in each course, filtering for courses with more than 5 students.
SELECT c.CourseId,c.CourseName, COUNT(e.StudentId) AS Total_Student
FROM Courses AS c
INNER JOIN Enrollments AS e
ON c.CourseId = e.CourseId
GROUP BY c.CourseId, c.CourseName
HAVING COUNT(e.StudentId) > 5;
```



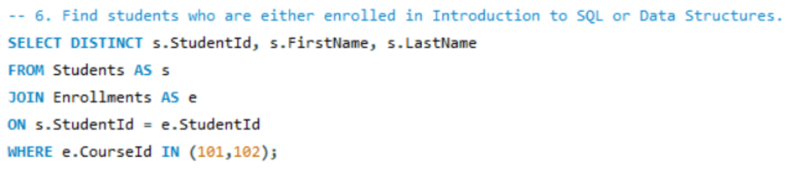
CourseId	CourseName	Total_Student
101	Introduction to SQL	7



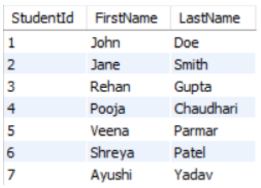
```sql
-- 5. Find students who are enrolled in both Introduction to SQL and Data Structures.
SELECT StudentId, FirstName, LastName
FROM Students
WHERE StudentId IN
(SELECT StudentId
    FROM Enrollments
    WHERE CourseId IN (101,102)
    GROUP BY StudentId
    HAVING COUNT(DISTINCT CourseId) = 2);
```



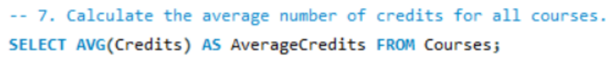
StudentId	FirstName	LastName
2	Jane	Smith



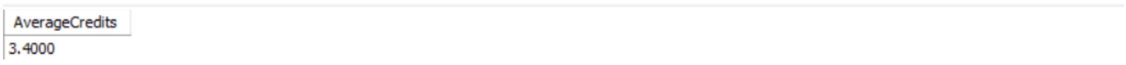
```sql
-- 6. Find students who are either enrolled in Introduction to SQL or Data Structures.
SELECT DISTINCT s.StudentId, s.FirstName, s.LastName
FROM Students AS s
JOIN Enrollments AS e
ON s.StudentId = e.StudentId
WHERE e.CourseId IN (101,102);
```



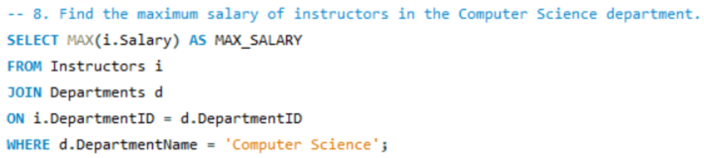
StudentId	FirstName	LastName
1	John	Doe
2	Jane	Smith
3	Rehan	Gupta
4	Pooja	Chaudhari
5	Veena	Parmar
6	Shreya	Patel
7	Ayushi	Yadav



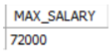
```sql
-- 7. Calculate the average number of credits for all courses.
SELECT AVG(Credits) AS AverageCredits FROM Courses;
```



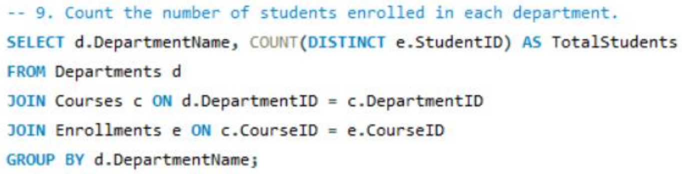
AverageCredits
3.4000



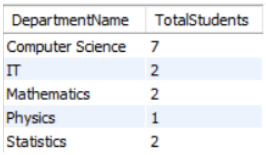
```sql
-- 8. Find the maximum salary of instructors in the Computer Science department.
SELECT MAX(i.Salary) AS MAX_SALARY
FROM Instructors i
JOIN Departments d
ON i.DepartmentID = d.DepartmentID
WHERE d.DepartmentName = 'Computer Science';
```



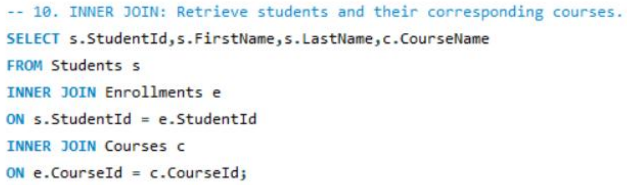
MAX_SALARY
72000



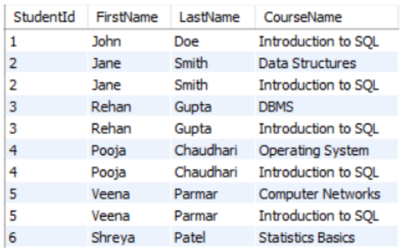
```sql
-- 9. Count the number of students enrolled in each department.
SELECT d.DepartmentName, COUNT(DISTINCT e.StudentID) AS TotalStudents
FROM Departments d
JOIN Courses c ON d.DepartmentID = c.DepartmentID
JOIN Enrollments e ON c.CourseID = e.CourseID
GROUP BY d.DepartmentName;
```



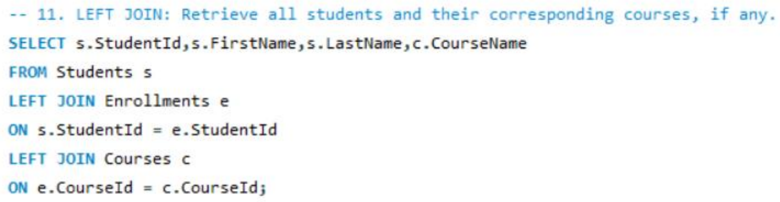
DepartmentName	TotalStudents
Computer Science	7
IT	2
Mathematics	2
Physics	1
Statistics	2



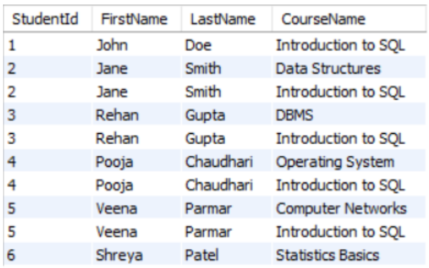
```sql
-- 10. INNER JOIN: Retrieve students and their corresponding courses.
SELECT s.StudentId,s.FirstName,s.LastName,c.CourseName
FROM Students s
INNER JOIN Enrollments e
ON s.StudentId = e.StudentId
INNER JOIN Courses c
ON e.CourseId = c.CourseId;
```



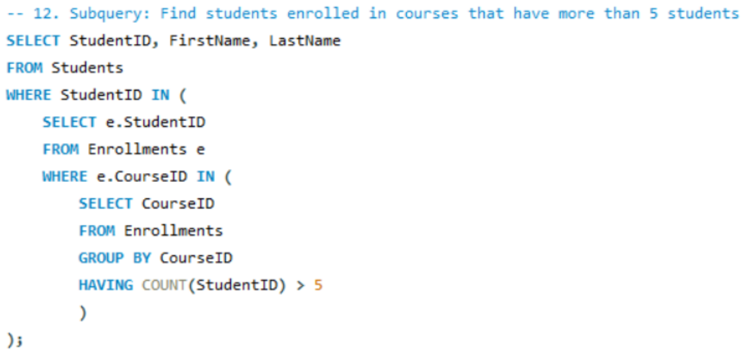
StudentId	FirstName	LastName	CourseName
1	John	Doe	Introduction to SQL
2	Jane	Smith	Data Structures
2	Jane	Smith	Introduction to SQL
3	Rehan	Gupta	DBMS
3	Rehan	Gupta	Introduction to SQL
4	Pooja	Chaudhari	Operating System
4	Pooja	Chaudhari	Introduction to SQL
5	Veena	Parmar	Computer Networks
5	Veena	Parmar	Introduction to SQL
6	Shreya	Patel	Statistics Basics



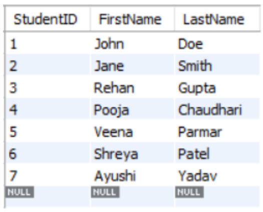
```sql
-- 11. LEFT JOIN: Retrieve all students and their corresponding courses, if any.
SELECT s.StudentId,s.FirstName,s.LastName,c.CourseName
FROM Students s
LEFT JOIN Enrollments e
ON s.StudentId = e.StudentId
LEFT JOIN Courses c
ON e.CourseId = c.CourseId;
```



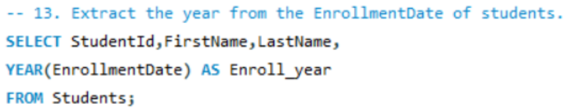
StudentId	FirstName	LastName	CourseName
1	John	Doe	Introduction to SQL
2	Jane	Smith	Data Structures
2	Jane	Smith	Introduction to SQL
3	Rehan	Gupta	DBMS
3	Rehan	Gupta	Introduction to SQL
4	Pooja	Chaudhari	Operating System
4	Pooja	Chaudhari	Introduction to SQL
5	Veena	Parmar	Computer Networks
5	Veena	Parmar	Introduction to SQL
6	Shreya	Patel	Statistics Basics



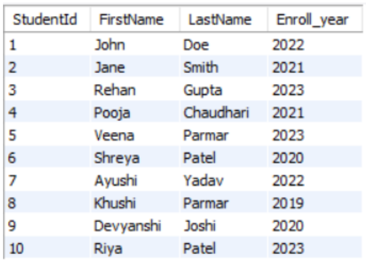
```sql
-- 12. Subquery: Find students enrolled in courses that have more than 5 students
SELECT StudentID, FirstName, LastName
FROM Students
WHERE StudentID IN (
    SELECT e.StudentID
    FROM Enrollments e
    WHERE e.CourseID IN (
        SELECT CourseID
        FROM Enrollments
        GROUP BY CourseID
        HAVING COUNT(StudentID) > 5
        )
);
```



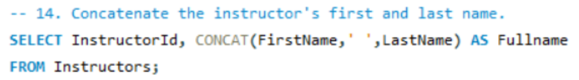
StudentID	FirstName	LastName
1	John	Doe
2	Jane	Smith
3	Rehan	Gupta
4	Pooja	Chaudhari
5	Veena	Parmar
6	Shreya	Patel
7	Ayushi	Yadav
NULL	NULL	NULL



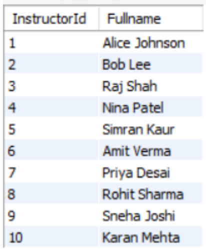
```sql
-- 13. Extract the year from the EnrollmentDate of students.
SELECT StudentId,FirstName,LastName,
YEAR(EnrollmentDate) AS Enroll_year
FROM Students;
```



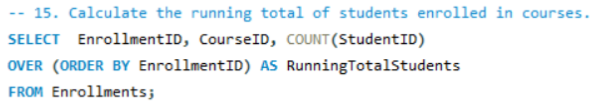
StudentId	FirstName	LastName	Enroll_year
1	John	Doe	2022
2	Jane	Smith	2021
3	Rehan	Gupta	2023
4	Pooja	Chaudhari	2021
5	Veena	Parmar	2023
6	Shreya	Patel	2020
7	Ayushi	Yadav	2022
8	Khushi	Parmar	2019
9	Devyanshi	Joshi	2020
10	Riya	Patel	2023



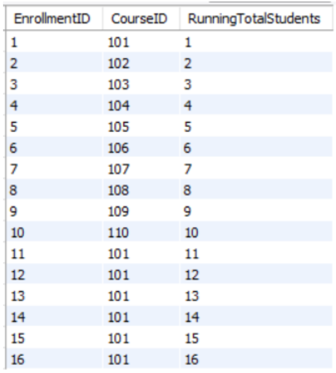
```
-- 14. Concatenate the instructor's first and last name.
SELECT InstructorId, CONCAT(FirstName,' ',LastName) AS Fullname
FROM Instructors;
```



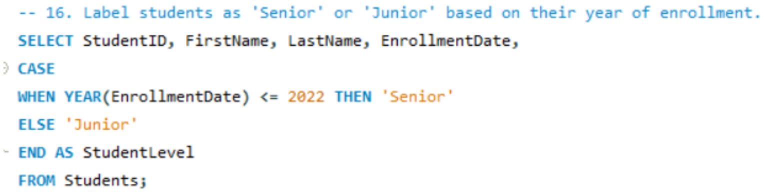
InstructorId	Fullname
1	Alice Johnson
2	Bob Lee
3	Raj Shah
4	Nina Patel
5	Simran Kaur
6	Amit Verma
7	Priya Desai
8	Rohit Sharma
9	Sneha Joshi
10	Karan Mehta



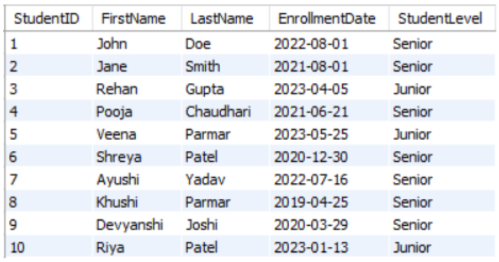
```
-- 15. Calculate the running total of students enrolled in courses.
SELECT  EnrollmentID, CourseID, COUNT(StudentID)
OVER (ORDER BY EnrollmentID) AS RunningTotalStudents
FROM Enrollments;
```



EnrollmentID	CourseID	RunningTotalStudents
1	101	1
2	102	2
3	103	3
4	104	4
5	105	5
6	106	6
7	107	7
8	108	8
9	109	9
10	110	10
11	101	11
12	101	12
13	101	13
14	101	14
15	101	15
16	101	16



```
-- 16. Label students as 'Senior' or 'Junior' based on their year of enrollment.
SELECT StudentID, FirstName, LastName, EnrollmentDate,
CASE
WHEN YEAR(EnrollmentDate) <= 2022 THEN 'Senior'
ELSE 'Junior'
END AS StudentLevel
FROM Students;
```



StudentID	FirstName	LastName	EnrollmentDate	StudentLevel
1	John	Doe	2022-08-01	Senior
2	Jane	Smith	2021-08-01	Senior
3	Rehan	Gupta	2023-04-05	Junior
4	Pooja	Chaudhari	2021-06-21	Senior
5	Veena	Parmar	2023-05-25	Junior
6	Shreya	Patel	2020-12-30	Senior
7	Ayushi	Yadav	2022-07-16	Senior
8	Khushi	Parmar	2019-04-25	Senior
9	Devyanshi	Joshi	2020-03-29	Senior
10	Riya	Patel	2023-01-13	Junior