Project Title: Academic Management System (using SQL)

Design and develop an Academic Management System using SQL. The projects should involve three tables 1.StudentInfo 2. CoursesInfo 3.EnrollmentInfo. The Aim is to create a system that allows for managing student information and course enrollment. The project will include the following tasks:

Query:

-- Create the database

```
CREATE DATABASE AcademicManagementSystem;
```

-- Use the database

```
USE AcademicManagementSystem;
```

-- Create StudentInfo table

```
CREATE TABLE StudentInfo (STU_ID VARCHAR(10) PRIMARY KEY,STU_NAME VARCHAR(50)
NOT NULL,DOB DATE,PHONE_NO VARCHAR(10),EMAIL_ID VARCHAR(50),ADDRESS TEXT);
```

-- Create CoursesInfo table

```
CREATE TABLE CoursesInfo (COURSE_ID VARCHAR(10) PRIMARY KEY,COURSE_NAME
VARCHAR(50) NOT NULL,COURSE_INSTRUCTOR_NAME VARCHAR(50));
```

-- Create EnrollmentInfo table with foreign key constraints

```
CREATE TABLE EnrollmentInfo (ENROLLMENT_ID VARCHAR(10) PRIMARY KEY,    STU_ID
VARCHAR(10),COURSE_ID VARCHAR(10),ENROLL_STATUS VARCHAR(20) CHECK
(ENROLL_STATUS IN ('Enrolled', 'Not Enrolled')),FOREIGN KEY (STU_ID) REFERENCES
StudentInfo(STU_ID),FOREIGN KEY (COURSE_ID) REFERENCES
CoursesInfo(COURSE_ID));
```

-- Insert sample data into StudentInfo table

```
INSERT INTO StudentInfo (STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID, ADDRESS)
VALUES('S001', 'John Smith', '2000-05-15', 9878675432, 'john.smith@email.com', '123 Main
St, City'),('S002', 'Emma Wilson', '2001-03-22', 9878675412, 'emma.w@email.com', '456
Oak Ave, Town'),('S003', 'Michael Brown', '2000-11-30', 9878675430,
'michael.b@email.com', '789 Pine Rd, Village'),('S004', 'Sarah Davis', '2001-07-18',
9878675422, 'sarah.d@email.com', '321 Elm St, City'),('S005', 'James Johnson', '2000-09-
25', 9878675932, 'james.j@email.com', '654 Maple Dr, Town');
```

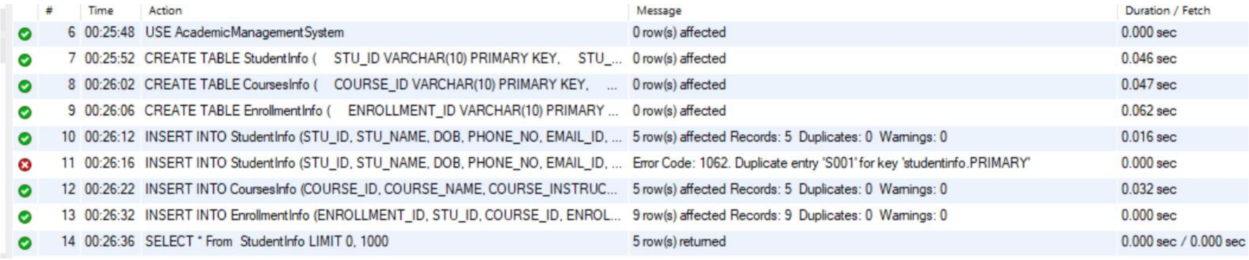
-- Insert sample data into CoursesInfo table

INSERT INTO CoursesInfo (COURSE_ID, COURSE_NAME, COURSE_INSTRUCTOR_NAME) VALUES('C101', 'Introduction to Computer Science', 'Dr. Robert Anderson'),('C102', 'Database Management Systems', 'Prof. Lisa Chen'),('C103', 'Web Development', 'Mr. David Wilson'),('C104', 'Data Structures', 'Dr. Emily Brown'),('C105', 'Artificial Intelligence', 'Prof. Mark Thompson');

-- Insert sample data into EnrollmentInfo table

INSERT INTO EnrollmentInfo (ENROLLMENT_ID, STU_ID, COURSE_ID, ENROLL_STATUS) VALUES('E0001', 'S001', 'C101', 'Enrolled'),('E0002', 'S001', 'C102', 'Enrolled'),('E0003', 'S002', 'C101', 'Enrolled'),('E0004', 'S003', 'C103', 'Enrolled'),('E0005', 'S004', 'C102', 'Not Enrolled'),('E0006', 'S002', 'C104', 'Enrolled'),('E0008', 'S003', 'C105', 'Not Enrolled'),('E0009', 'S004', 'C101', 'Enrolled'),('E0010', 'S005', 'C103', 'Enrolled');

Output:



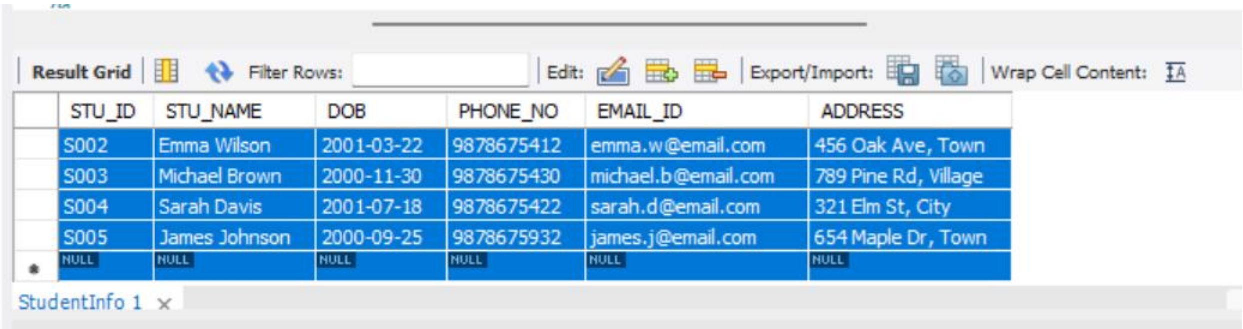
	#	Time	Action	Message	Duration / Fetch
✓	6	00:25:48	USE AcademicManagementSystem	0 row(s) affected	0.000 sec
✓	7	00:25:52	CREATE TABLE StudentInfo (STU_ID VARCHAR(10) PRIMARY KEY, STU_...	0 row(s) affected	0.046 sec
✓	8	00:26:02	CREATE TABLE CoursesInfo (COURSE_ID VARCHAR(10) PRIMARY KEY, ...	0 row(s) affected	0.047 sec
✓	9	00:26:06	CREATE TABLE EnrollmentInfo (ENROLLMENT_ID VARCHAR(10) PRIMARY ...	0 row(s) affected	0.062 sec
✓	10	00:26:12	INSERT INTO StudentInfo (STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID, ...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
✗	11	00:26:16	INSERT INTO StudentInfo (STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID, ...	Error Code: 1062. Duplicate entry 'S001' for key 'studentinfo.PRIMARY'	0.000 sec
✓	12	00:26:22	INSERT INTO CoursesInfo (COURSE_ID, COURSE_NAME, COURSE_INSTRUC...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.032 sec
✓	13	00:26:32	INSERT INTO EnrollmentInfo (ENROLLMENT_ID, STU_ID, COURSE_ID, ENROL...	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.000 sec
✓	14	00:26:36	SELECT * From StudentInfo LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

-- Retrieve the Student Information

Query:

SELECT * From StudentInfo;

Output:

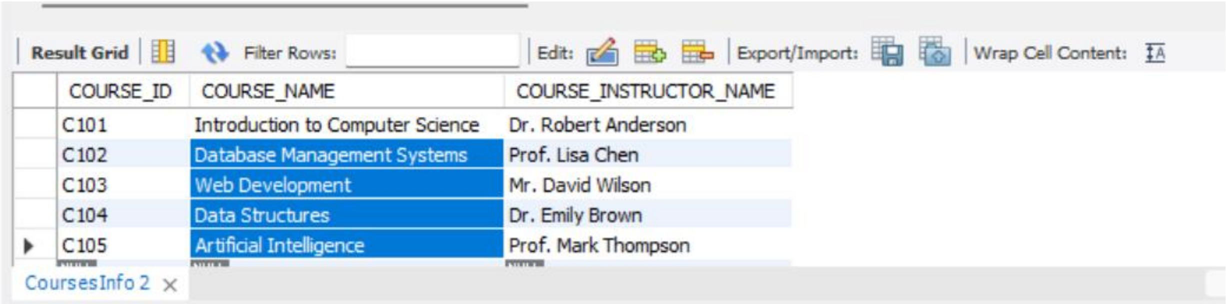


	STU_ID	STU_NAME	DOB	PHONE_NO	EMAIL_ID	ADDRESS
	S002	Emma Wilson	2001-03-22	9878675412	emma.w@email.com	456 Oak Ave, Town
	S003	Michael Brown	2000-11-30	9878675430	michael.b@email.com	789 Pine Rd, Village
	S004	Sarah Davis	2001-07-18	9878675422	sarah.d@email.com	321 Elm St, City
	S005	James Johnson	2000-09-25	9878675932	james.j@email.com	654 Maple Dr, Town
*	NULL	NULL	NULL	NULL	NULL	NULL

Query:

SELECT * FROM CoursesInfo;

Output:

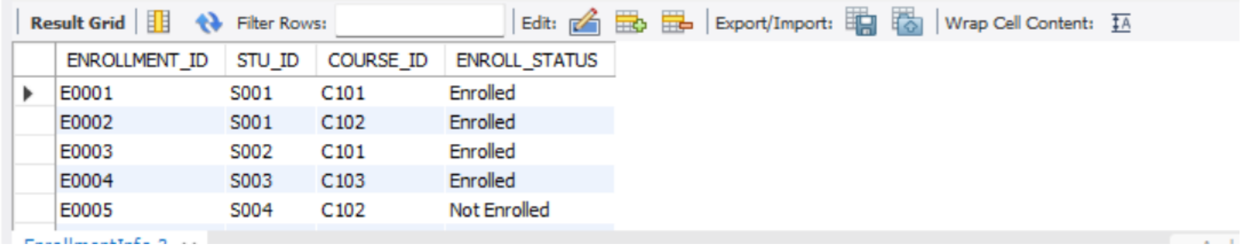


COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME
C101	Introduction to Computer Science	Dr. Robert Anderson
C102	Database Management Systems	Prof. Lisa Chen
C103	Web Development	Mr. David Wilson
C104	Data Structures	Dr. Emily Brown
C105	Artificial Intelligence	Prof. Mark Thompson

Query:

SELECT * FROM EnrollmentInfo;

Output:



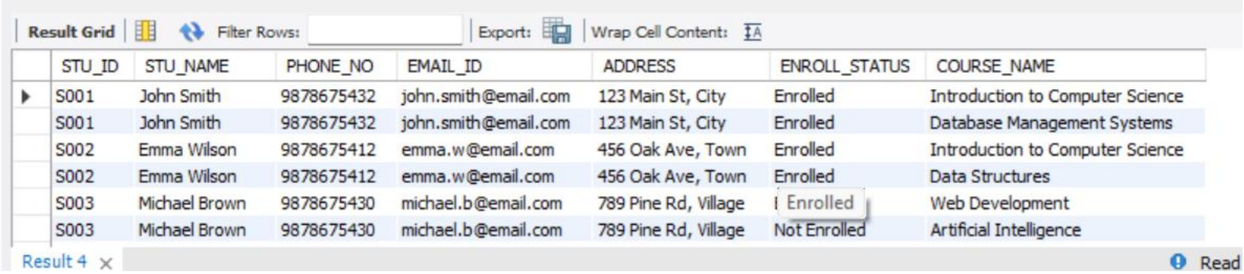
ENROLLMENT_ID	STU_ID	COURSE_ID	ENROLL_STATUS
E0001	S001	C101	Enrolled
E0002	S001	C102	Enrolled
E0003	S002	C101	Enrolled
E0004	S003	C103	Enrolled
E0005	S004	C102	Not Enrolled

-- a) Write a query to retrieve student details, such as student name, contact informations, and Enrollment status

Query:

SELECT   s.STU_ID, s.STU_NAME, s.PHONE_NO,s.EMAIL_ID,s.ADDRESS, e.ENROLL_STATUS, c.COURSE_NAME FROM StudentInfo s LEFT JOIN EnrollmentInfo e ON s.STU_ID = e.STU_ID LEFT JOIN CoursesInfo c ON e.COURSE_ID = c.COURSE_ID;

Output:



STU_ID	STU_NAME	PHONE_NO	EMAIL_ID	ADDRESS	ENROLL_STATUS	COURSE_NAME
S001	John Smith	9878675432	john.smith@email.com	123 Main St, City	Enrolled	Introduction to Computer Science
S001	John Smith	9878675432	john.smith@email.com	123 Main St, City	Enrolled	Database Management Systems
S002	Emma Wilson	9878675412	emma.w@email.com	456 Oak Ave, Town	Enrolled	Introduction to Computer Science
S002	Emma Wilson	9878675412	emma.w@email.com	456 Oak Ave, Town	Enrolled	Data Structures
S003	Michael Brown	9878675430	michael.b@email.com	789 Pine Rd, Village	Enrolled	Web Development
S003	Michael Brown	9878675430	michael.b@email.com	789 Pine Rd, Village	Not Enrolled	Artificial Intelligence

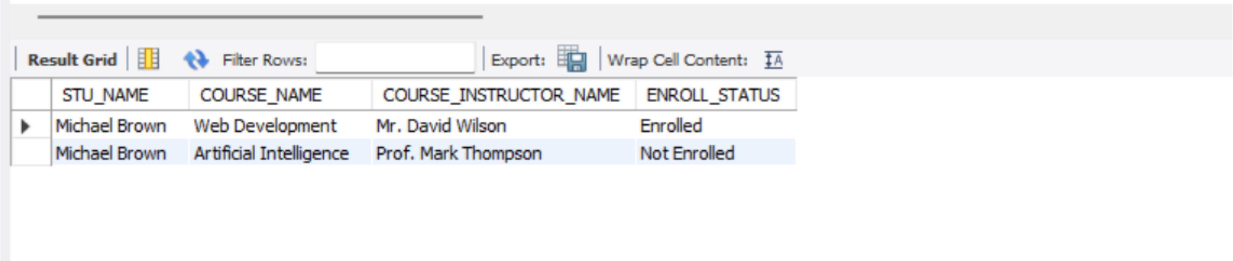
Query:

-- b) Write a query to retrieve a list of courses in which a specific student is enrolled

Query:

```
SELECT s.STU_NAME,    c.COURSE_NAME,    c.COURSE_INSTRUCTOR_NAME,
 e.ENROLL_STATUS FROM StudentInfo s JOIN EnrollmentInfo e ON s.STU_ID = e.STU_ID
JOIN CoursesInfo c ON e.COURSE_ID = c.COURSE_ID WHERE s.STU_ID = 'S003';
```

Output:



STU_NAME	COURSE_NAME	COURSE_INSTRUCTOR_NAME	ENROLL_STATUS
Michael Brown	Web Development	Mr. David Wilson	Enrolled
Michael Brown	Artificial Intelligence	Prof. Mark Thompson	Not Enrolled

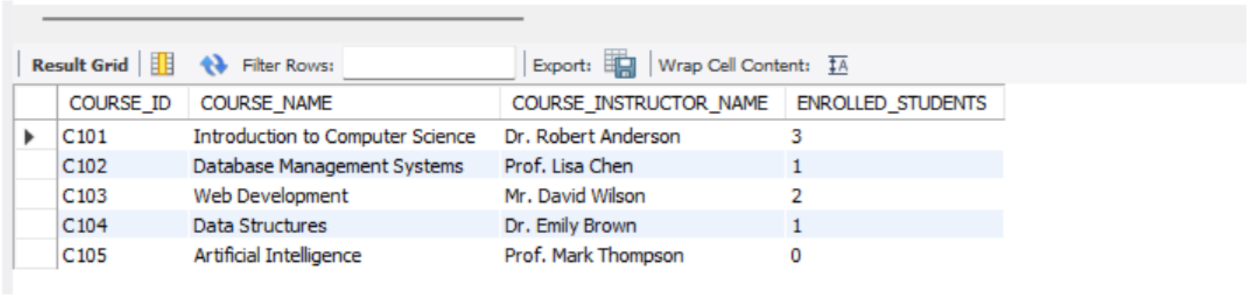
-- c) Write a query to retrieve course information, including course name, instructor information.

Query:

```
SELECT

    COURSE_ID,
    COURSE_NAME,
    COURSE_INSTRUCTOR_NAME,
    (SELECT COUNT(*)
     FROM EnrollmentInfo e
     WHERE e.COURSE_ID = c.COURSE_ID AND e.ENROLL_STATUS = 'Enrolled') as
ENROLLED_STUDENTS
FROM CoursesInfo c;
```

Output:



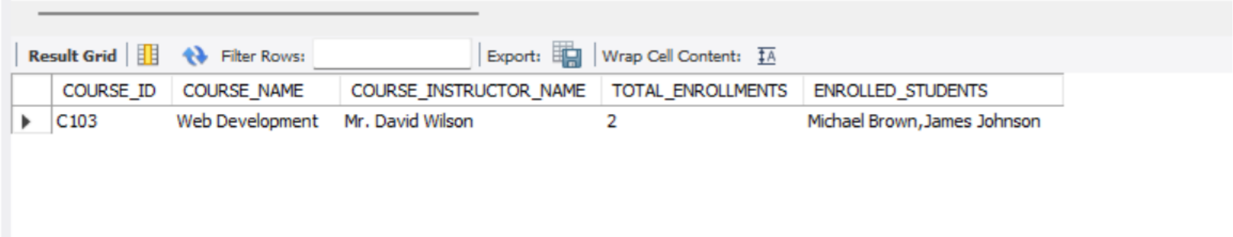
COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME	ENROLLED_STUDENTS
C101	Introduction to Computer Science	Dr. Robert Anderson	3
C102	Database Management Systems	Prof. Lisa Chen	1
C103	Web Development	Mr. David Wilson	2
C104	Data Structures	Dr. Emily Brown	1
C105	Artificial Intelligence	Prof. Mark Thompson	0

-- d) Write a query to retrieve course information for a specific course .

Query:

```
SELECT
   c.COURSE_ID,
   c.COURSE_NAME,
   c.COURSE_INSTRUCTOR_NAME,
   COUNT(e.STU_ID) as TOTAL_ENROLLMENTS,
   GROUP_CONCAT(s.STU_NAME) as ENROLLED_STUDENTS
FROM CoursesInfo c
LEFT JOIN EnrollmentInfo e ON c.COURSE_ID = e.COURSE_ID
LEFT JOIN StudentInfo s ON e.STU_ID = s.STU_ID
WHERE c.COURSE_ID = 'C103'
GROUP BY c.COURSE_ID, c.COURSE_NAME, c.COURSE_INSTRUCTOR_NAME;
```

Output:



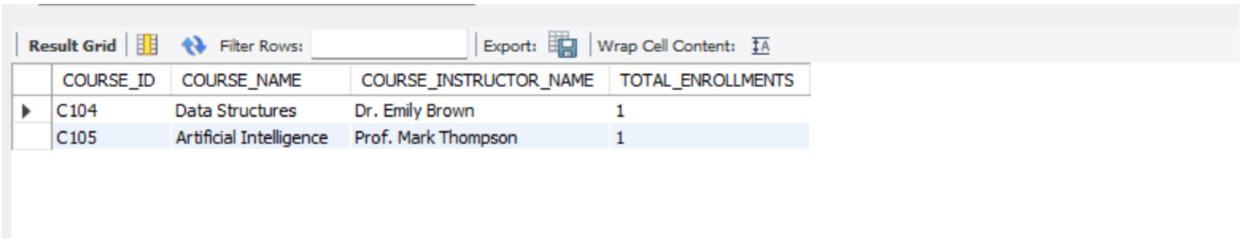
COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME	TOTAL_ENROLLMENTS	ENROLLED_STUDENTS
C103	Web Development	Mr. David Wilson	2	Michael Brown,James Johnson

-- e) Write a query to retrieve course information for multiple courses.

Query:

```
SELECT
   c.COURSE_ID,
   c.COURSE_NAME,
   c.COURSE_INSTRUCTOR_NAME,
   COUNT(e.STU_ID) as TOTAL_ENROLLMENTS
FROM CoursesInfo c
LEFT JOIN EnrollmentInfo e ON c.COURSE_ID = e.COURSE_ID
WHERE c.COURSE_ID IN ('C104', 'C105')
GROUP BY c.COURSE_ID, c.COURSE_NAME, c.COURSE_INSTRUCTOR_NAME;
```

Output:



COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME	TOTAL_ENROLLMENTS
C104	Data Structures	Dr. Emily Brown	1
C105	Artificial Intelligence	Prof. Mark Thompson	1

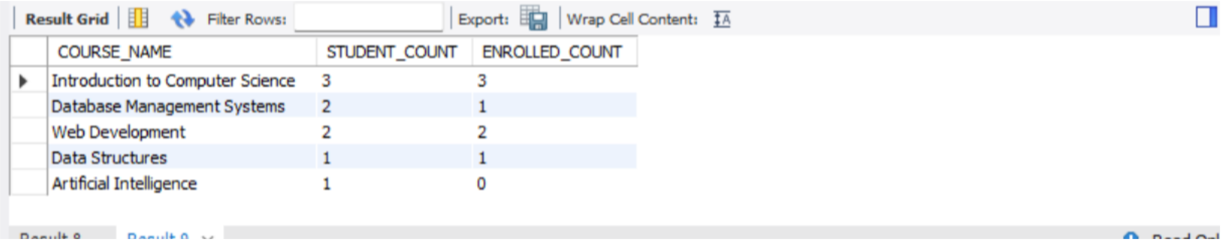
-- f) Test the queries to ensure accurate retrieval of student information.(execute the queries and verify the results against the expected output.)

-- Test 1: Count of students per course

Query:

```
SELECT
   c.COURSE_NAME,
   COUNT(e.STU_ID) as STUDENT_COUNT,
   COUNT(CASE WHEN e.ENROLL_STATUS = 'Enrolled' THEN 1 END) as ENROLLED_COUNT
FROM CoursesInfo c
LEFT JOIN EnrollmentInfo e ON c.COURSE_ID = e.COURSE_ID
GROUP BY c.COURSE_NAME;
```

Output:



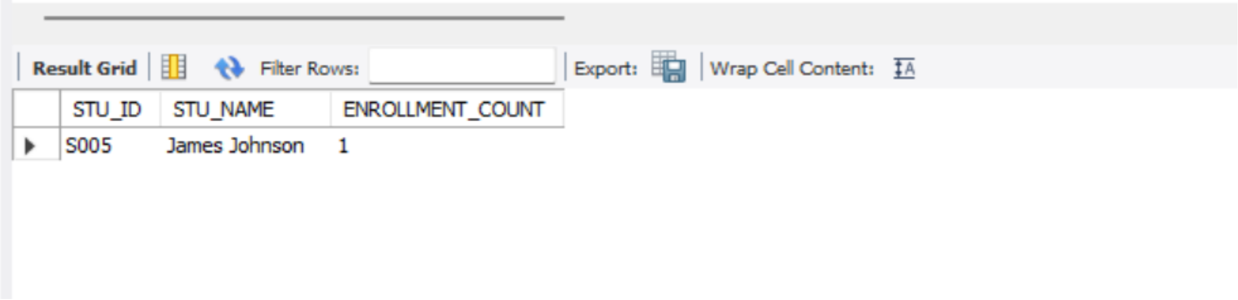
COURSE_NAME	STUDENT_COUNT	ENROLLED_COUNT
Introduction to Computer Science	3	3
Database Management Systems	2	1
Web Development	2	2
Data Structures	1	1
Artificial Intelligence	1	0

-- Test 2: Check for any students without enrollments

Query:

```
SELECT
  s.STU_ID,
  s.STU_NAME,
  COUNT(e.ENROLLMENT_ID) as ENROLLMENT_COUNT
FROM StudentInfo s
LEFT JOIN EnrollmentInfo e ON s.STU_ID = e.STU_ID
GROUP BY s.STU_ID, s.STU_NAME
HAVING ENROLLMENT_COUNT = 1;
```

Output:



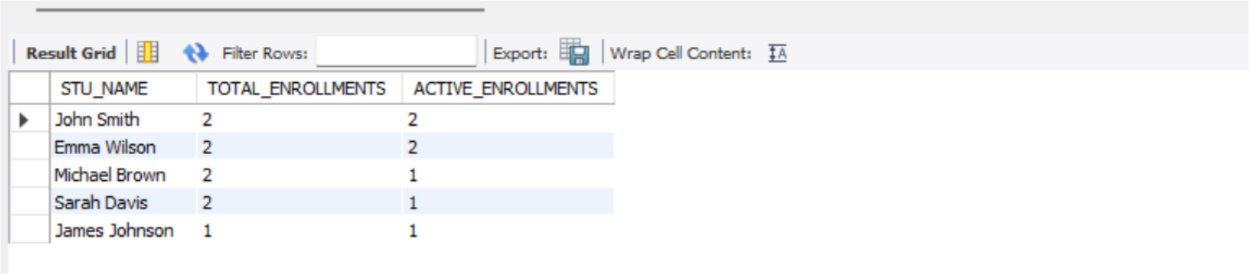
STU_ID	STU_NAME	ENROLLMENT_COUNT
S005	James Johnson	1

-- Test 3: Verify course enrollment details

Query:

```
SELECT
  s.STU_NAME,
  COUNT(e.ENROLLMENT_ID) as TOTAL_ENROLLMENTS,
  SUM(CASE WHEN e.ENROLL_STATUS = 'Enrolled' THEN 1 ELSE 0 END) as
ACTIVE_ENROLLMENTS
FROM StudentInfo s
LEFT JOIN EnrollmentInfo e ON s.STU_ID = e.STU_ID
GROUP BY s.STU_NAME;
```

Output:



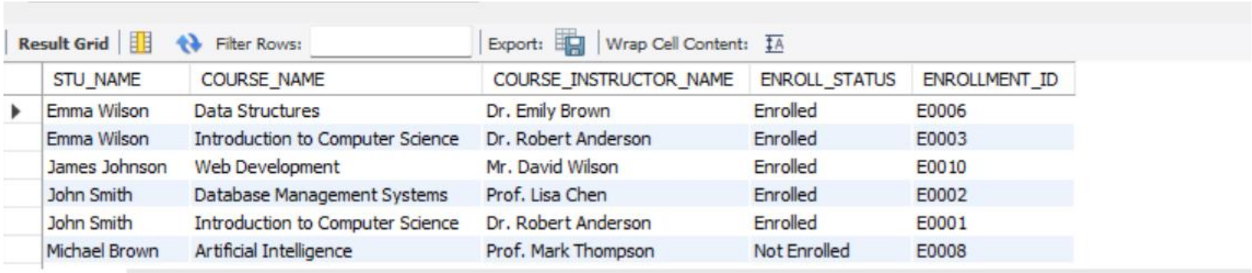
STU_NAME	TOTAL_ENROLLMENTS	ACTIVE_ENROLLMENTS
John Smith	2	2
Emma Wilson	2	2
Michael Brown	2	1
Sarah Davis	2	1
James Johnson	1	1

-- Test 4: Complete enrollment report

Query:

```
SELECT
   s.STU_NAME,
   c.COURSE_NAME,
   c.COURSE_INSTRUCTOR_NAME,
   e.ENROLL_STATUS,
   e.ENROLLMENT_ID
FROM StudentInfo s
JOIN EnrollmentInfo e ON s.STU_ID = e.STU_ID
JOIN CoursesInfo c ON e.COURSE_ID = c.COURSE_ID
ORDER BY s.STU_NAME, c.COURSE_NAME;
```

Output:



STU_NAME	COURSE_NAME	COURSE_INSTRUCTOR_NAME	ENROLL_STATUS	ENROLLMENT_ID
Emma Wilson	Data Structures	Dr. Emily Brown	Enrolled	E0006
Emma Wilson	Introduction to Computer Science	Dr. Robert Anderson	Enrolled	E0003
James Johnson	Web Development	Mr. David Wilson	Enrolled	E0010
John Smith	Database Management Systems	Prof. Lisa Chen	Enrolled	E0002
John Smith	Introduction to Computer Science	Dr. Robert Anderson	Enrolled	E0001
Michael Brown	Artificial Intelligence	Prof. Mark Thompson	Not Enrolled	E0008

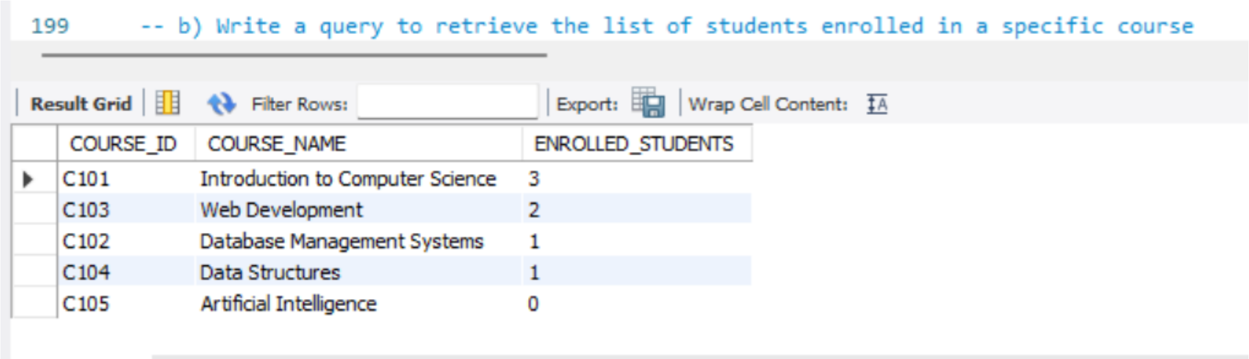
-- Reporting and Analytics

-- a) Write a query to retrieve the number of students enrolled in each course

Query:

```
SELECT
   c.COURSE_ID,
   c.COURSE_NAME,
   COUNT(CASE WHEN e.ENROLL_STATUS = 'Enrolled' THEN 1 END) as
ENROLLED_STUDENTS
FROM CoursesInfo c
LEFT JOIN EnrollmentInfo e ON c.COURSE_ID = e.COURSE_ID
GROUP BY c.COURSE_ID, c.COURSE_NAME
ORDER BY ENROLLED_STUDENTS DESC;
```

Output:



```
-- b) Write a query to retrieve the list of students enrolled in a specific course
```

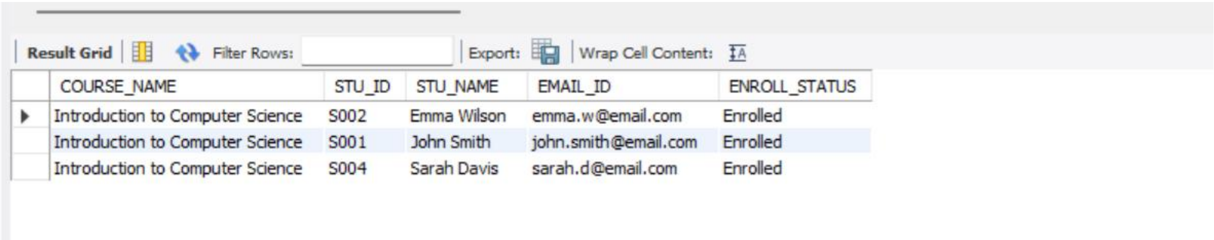
	COURSE_ID	COURSE_NAME	ENROLLED_STUDENTS
▶	C101	Introduction to Computer Science	3
	C103	Web Development	2
	C102	Database Management Systems	1
	C104	Data Structures	1
	C105	Artificial Intelligence	0

-- b) Write a query to retrieve the list of students enrolled in a specific course

Query:

```
SELECT
        c.COURSE_NAME,
  s.STU_ID,
  s.STU_NAME,
  s.EMAIL_ID,
  e.ENROLL_STATUS
FROM CoursesInfo c
JOIN EnrollmentInfo e ON c.COURSE_ID = e.COURSE_ID
JOIN StudentInfo s ON e.STU_ID = s.STU_ID
WHERE c.COURSE_ID = 'C101' AND e.ENROLL_STATUS = 'Enrolled'
ORDER BY s.STU_NAME;
```

Output:



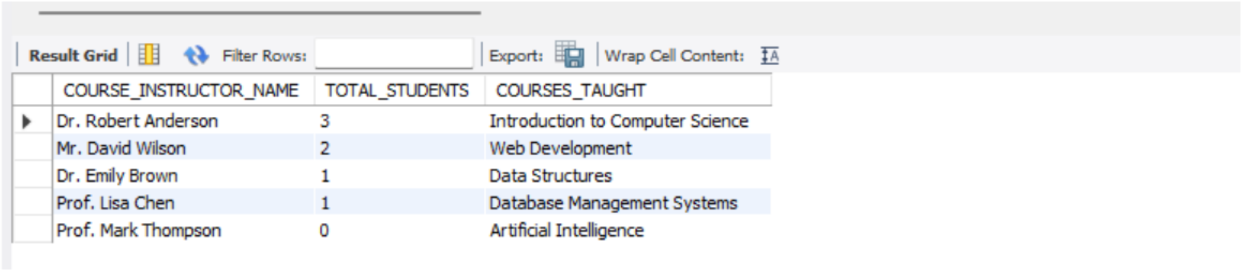
	COURSE_NAME	STU_ID	STU_NAME	EMAIL_ID	ENROLL_STATUS
▶	Introduction to Computer Science	S002	Emma Wilson	emma.w@email.com	Enrolled
	Introduction to Computer Science	S001	John Smith	john.smith@email.com	Enrolled
	Introduction to Computer Science	S004	Sarah Davis	sarah.d@email.com	Enrolled

-- c) Query to retrieve the count of enrolled students for each instructor

Query:

```
SELECT
   c.COURSE_INSTRUCTOR_NAME,
   COUNT(DISTINCT CASE WHEN e.ENROLL_STATUS = 'Enrolled' THEN s.STU_ID END) as
TOTAL_STUDENTS,
   GROUP_CONCAT(DISTINCT c.COURSE_NAME) as COURSES_TAUGHT
FROM CoursesInfo c
LEFT JOIN EnrollmentInfo e ON c.COURSE_ID = e.COURSE_ID
LEFT JOIN StudentInfo s ON e.STU_ID = s.STU_ID
GROUP BY c.COURSE_INSTRUCTOR_NAME
ORDER BY TOTAL_STUDENTS DESC;
```

Output:



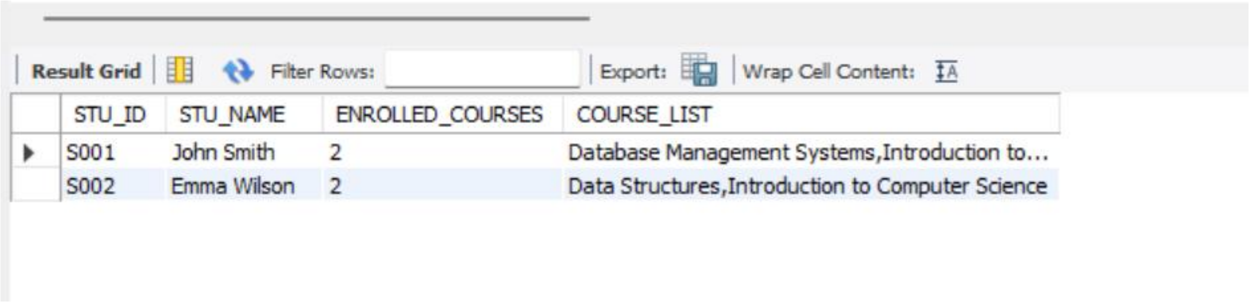
COURSE_INSTRUCTOR_NAME	TOTAL_STUDENTS	COURSES_TAUGHT
Dr. Robert Anderson	3	Introduction to Computer Science
Mr. David Wilson	2	Web Development
Dr. Emily Brown	1	Data Structures
Prof. Lisa Chen	1	Database Management Systems
Prof. Mark Thompson	0	Artificial Intelligence

-- d) Query to retrieve students enrolled in multiple courses

Query:

```
SELECT
   s.STU_ID,
   s.STU_NAME,
   COUNT(CASE WHEN e.ENROLL_STATUS = 'Enrolled' THEN 1 END) as
ENROLLED_COURSES,
   GROUP_CONCAT(DISTINCT c.COURSE_NAME) as COURSE_LIST
FROM StudentInfo s
JOIN EnrollmentInfo e ON s.STU_ID = e.STU_ID
JOIN CoursesInfo c ON e.COURSE_ID = c.COURSE_ID
WHERE e.ENROLL_STATUS = 'Enrolled'
GROUP BY s.STU_ID, s.STU_NAME
HAVING ENROLLED_COURSES > 1
ORDER BY ENROLLED_COURSES DESC;
```

Output:



STU_ID	STU_NAME	ENROLLED_COURSES	COURSE_LIST
S001	John Smith	2	Database Management Systems,Introduction to...
S002	Emma Wilson	2	Data Structures,Introduction to Computer Science

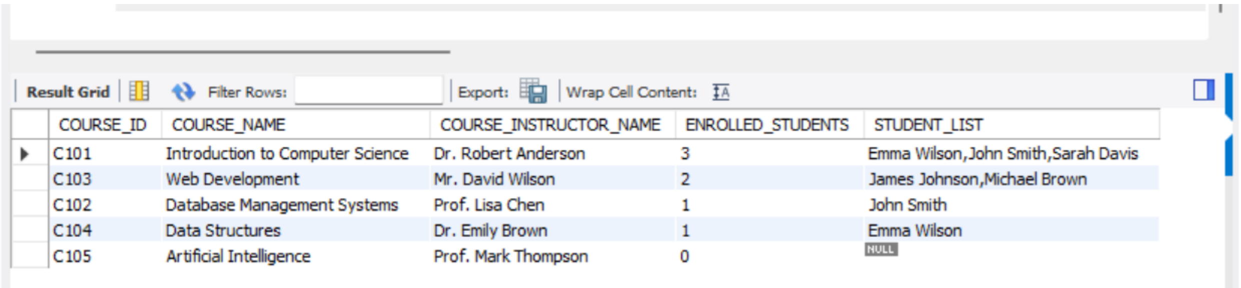
-- e) Query to retrieve courses with the highest number of enrolled students

Query:

```
SELECT

   c.COURSE_ID,
   c.COURSE_NAME,
   c.COURSE_INSTRUCTOR_NAME,
   COUNT(CASE WHEN e.ENROLL_STATUS = 'Enrolled' THEN 1 END) as
ENROLLED_STUDENTS,
   GROUP_CONCAT(DISTINCT s.STU_NAME) as STUDENT_LIST
FROM CoursesInfo c
LEFT JOIN EnrollmentInfo e ON c.COURSE_ID = e.COURSE_ID
LEFT JOIN StudentInfo s ON e.STU_ID = s.STU_ID AND e.ENROLL_STATUS = 'Enrolled'
GROUP BY c.COURSE_ID, c.COURSE_NAME, c.COURSE_INSTRUCTOR_NAME
ORDER BY ENROLLED_STUDENTS DESC;
```

Output:



COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME	ENROLLED_STUDENTS	STUDENT_LIST
C101	Introduction to Computer Science	Dr. Robert Anderson	3	Emma Wilson,John Smith,Sarah Davis
C103	Web Development	Mr. David Wilson	2	James Johnson,Michael Brown
C102	Database Management Systems	Prof. Lisa Chen	1	John Smith
C104	Data Structures	Dr. Emily Brown	1	Emma Wilson
C105	Artificial Intelligence	Prof. Mark Thompson	0	NULL