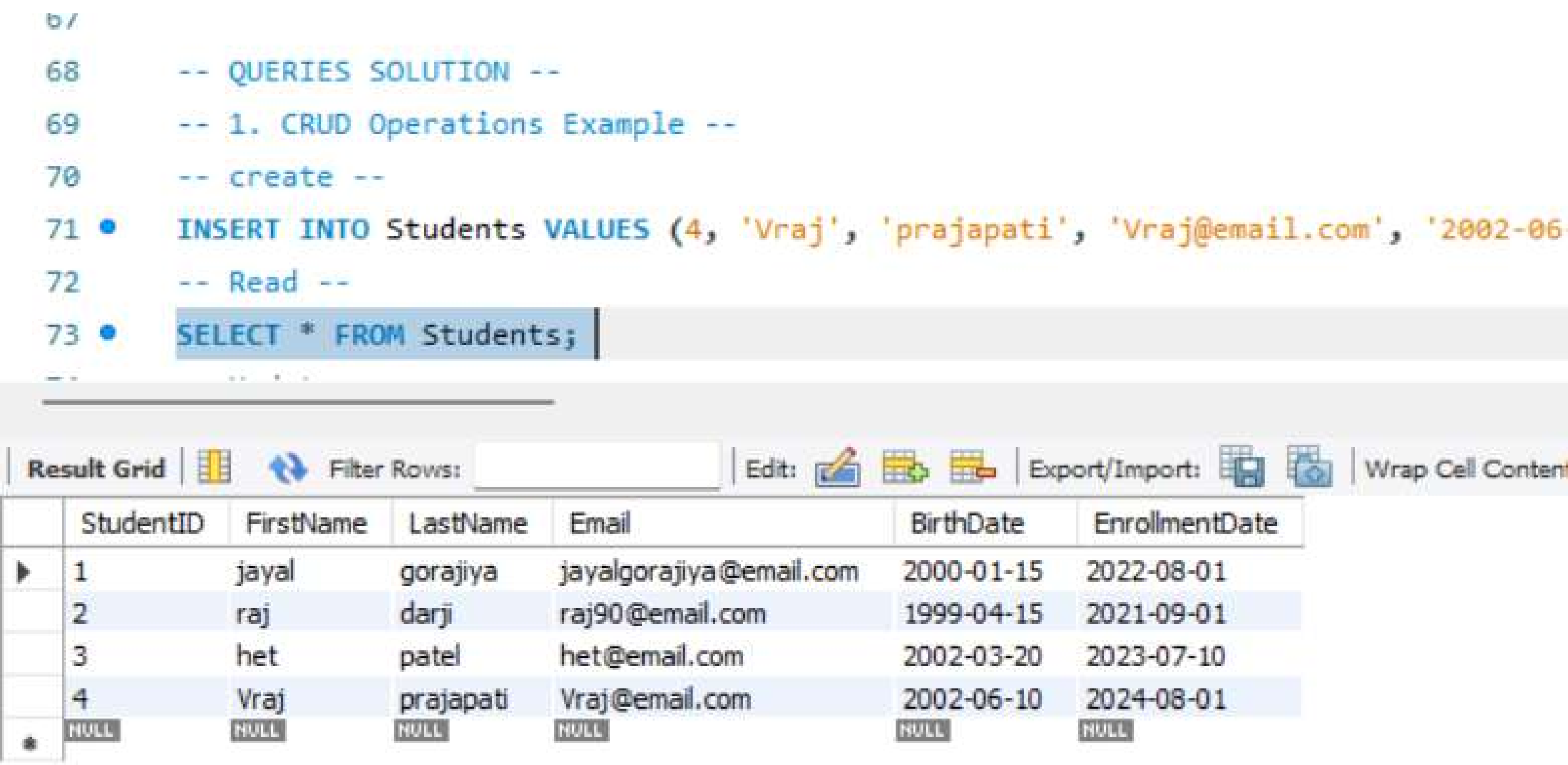
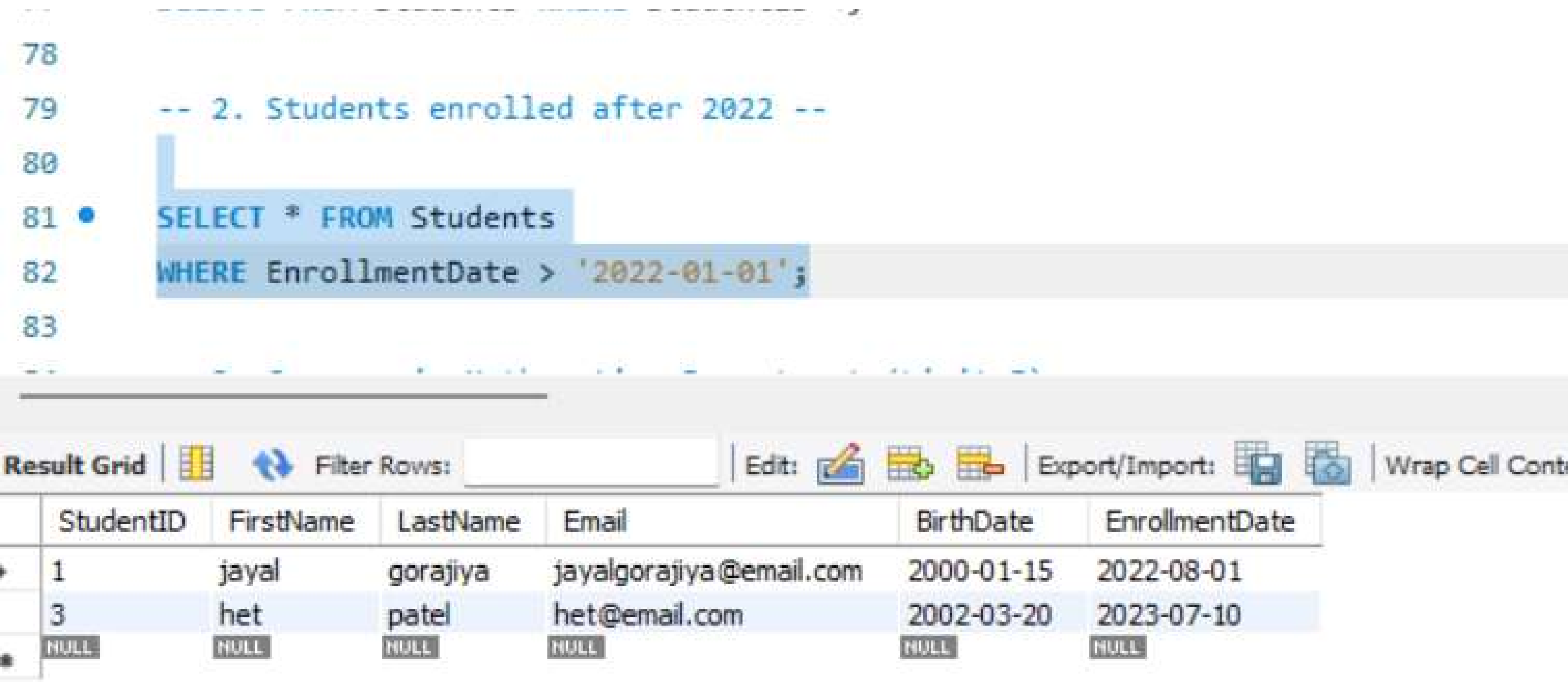


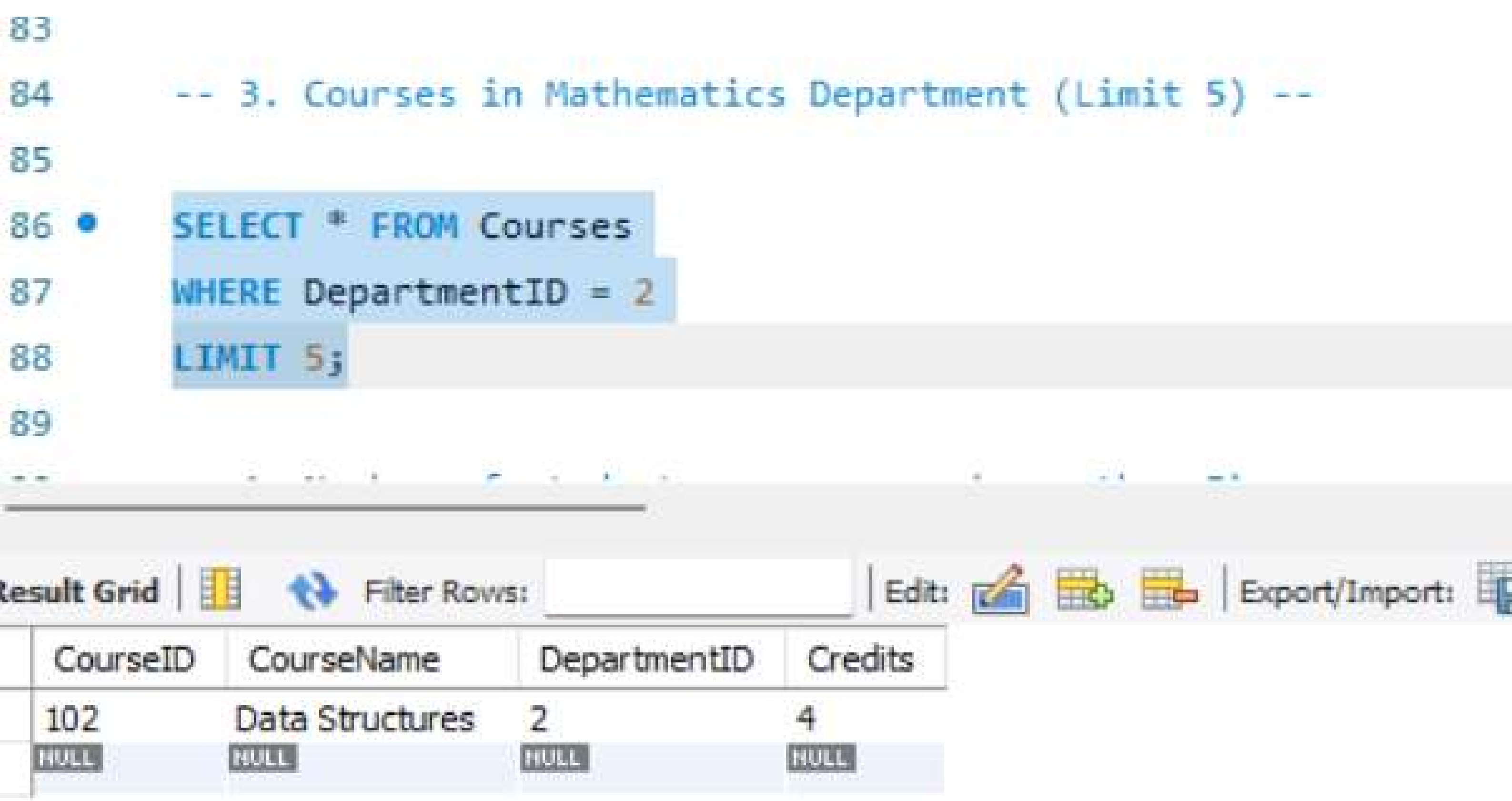
```sql
-- QUERIES SOLUTION --
-- 1. CRUD Operations Example --
-- create --
INSERT INTO Students VALUES (4, 'Vraj', 'prajapati', 'Vraj@email.com', '2002-06-
-- Read --
SELECT * FROM Students;
```

StudentID	FirstName	LastName	Email	BirthDate	EnrollmentDate
1	jayal	gorajiya	jayalgorajiya@email.com	2000-01-15	2022-08-01
2	raj	darji	raj90@email.com	1999-04-15	2021-09-01
3	het	patel	het@email.com	2002-03-20	2023-07-10
4	Vraj	prajapati	Vraj@email.com	2002-06-10	2024-08-01
NULL	NULL	NULL	NULL	NULL	NULL



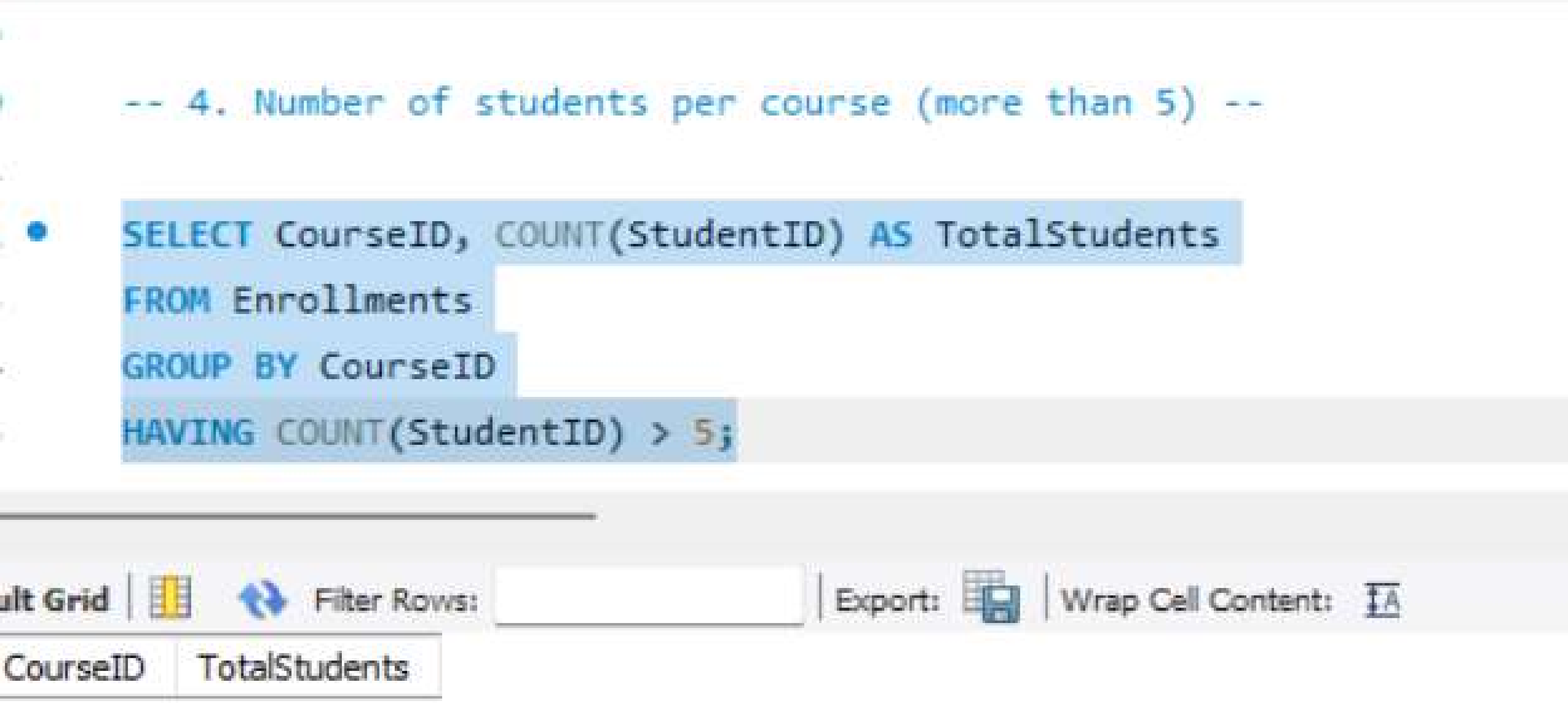
```
-- 2. Students enrolled after 2022 --

SELECT * FROM Students
WHERE EnrollmentDate > '2022-01-01';
```

StudentID	FirstName	LastName	Email	BirthDate	EnrollmentDate
1	jayal	gorajiya	jayalgorajiya@email.com	2000-01-15	2022-08-01
3	het	patel	het@email.com	2002-03-20	2023-07-10
NULL	NULL	NULL	NULL	NULL	NULL



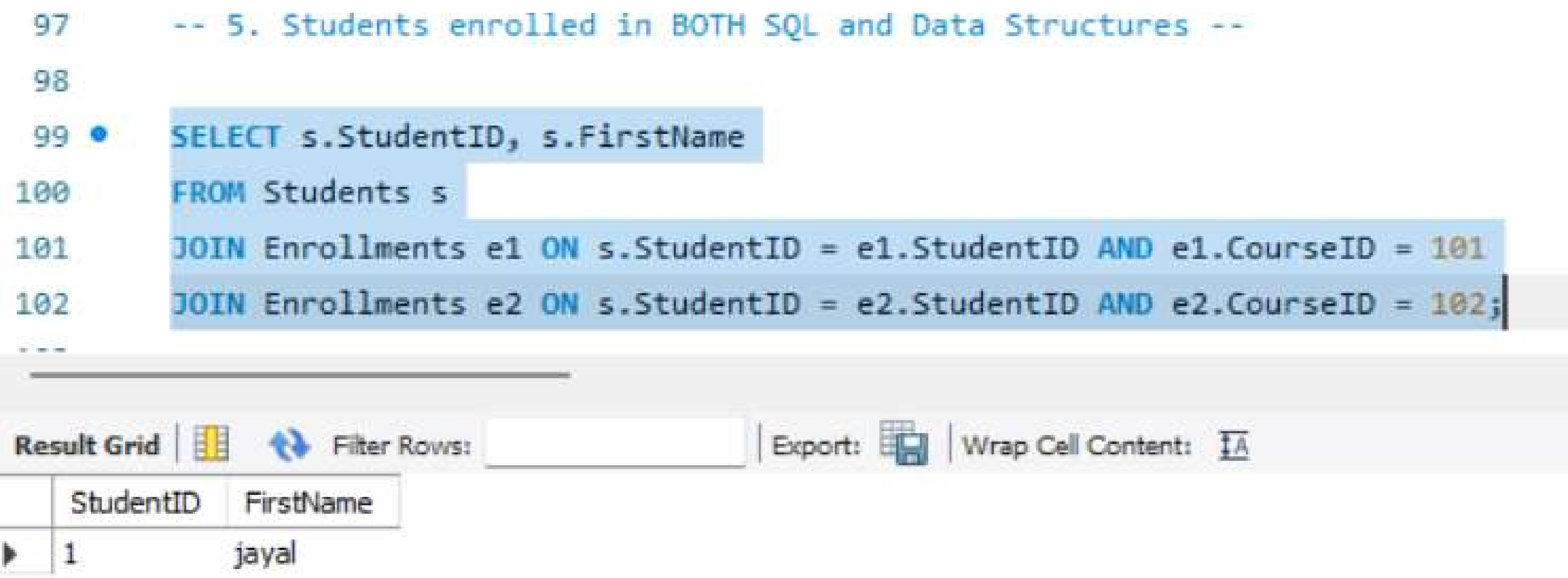
```
-- 3. Courses in Mathematics Department (Limit 5) --

SELECT * FROM Courses
WHERE DepartmentID = 2
LIMIT 5;
```

CourseID	CourseName	DepartmentID	Credits
102	Data Structures	2	4
NULL	NULL	NULL	NULL



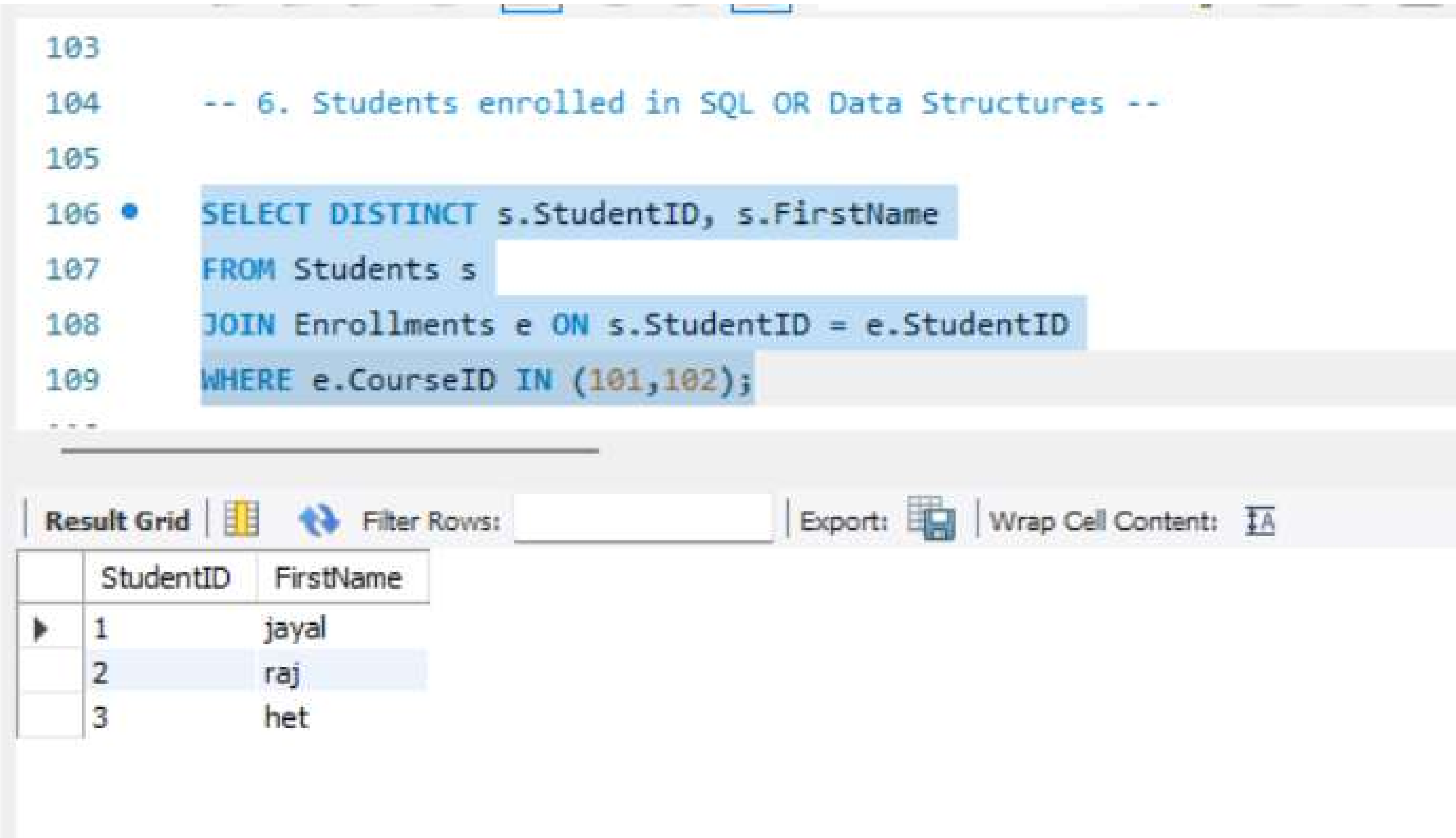
```
-- 4. Number of students per course (more than 5) --

SELECT CourseID, COUNT(StudentID) AS TotalStudents
FROM Enrollments
GROUP BY CourseID
HAVING COUNT(StudentID) > 5;
```

CourseID	TotalStudents



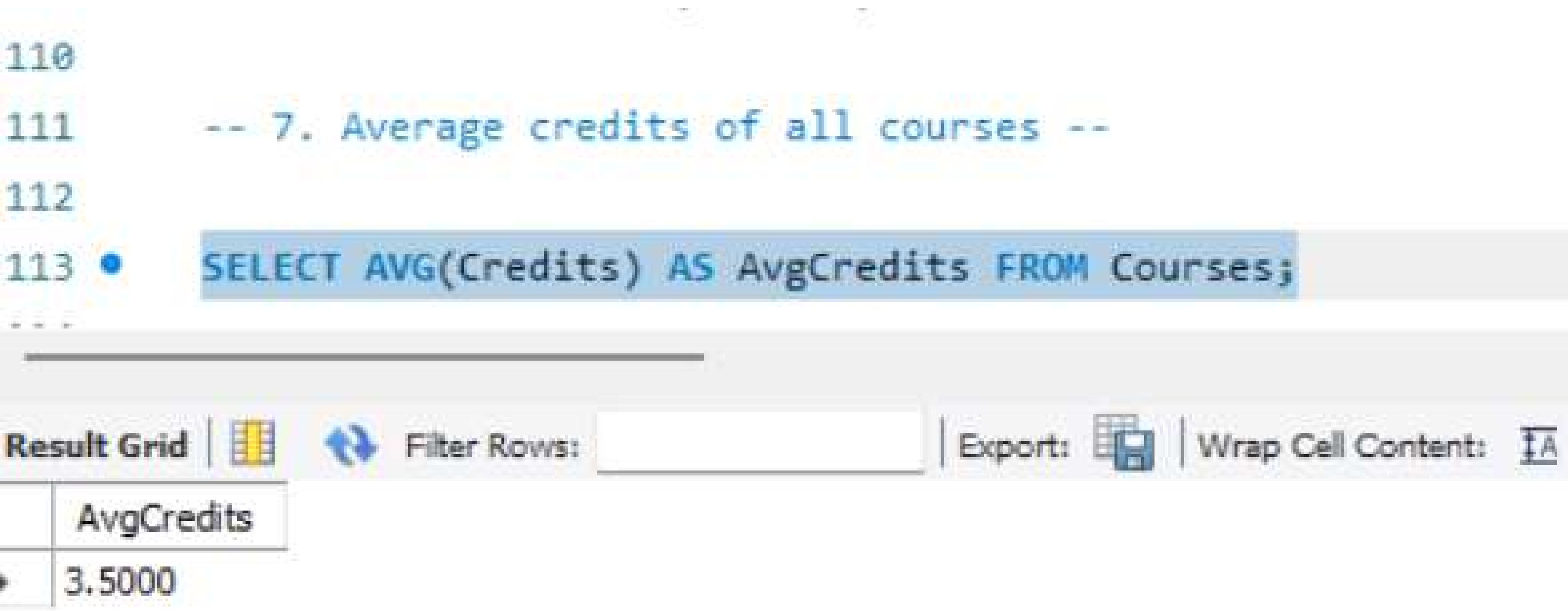
```sql
-- 5. Students enrolled in BOTH SQL and Data Structures --

SELECT s.StudentID, s.FirstName
FROM Students s
JOIN Enrollments e1 ON s.StudentID = e1.StudentID AND e1.CourseID = 101
JOIN Enrollments e2 ON s.StudentID = e2.StudentID AND e2.CourseID = 102;
```

StudentID	FirstName
1	jayal



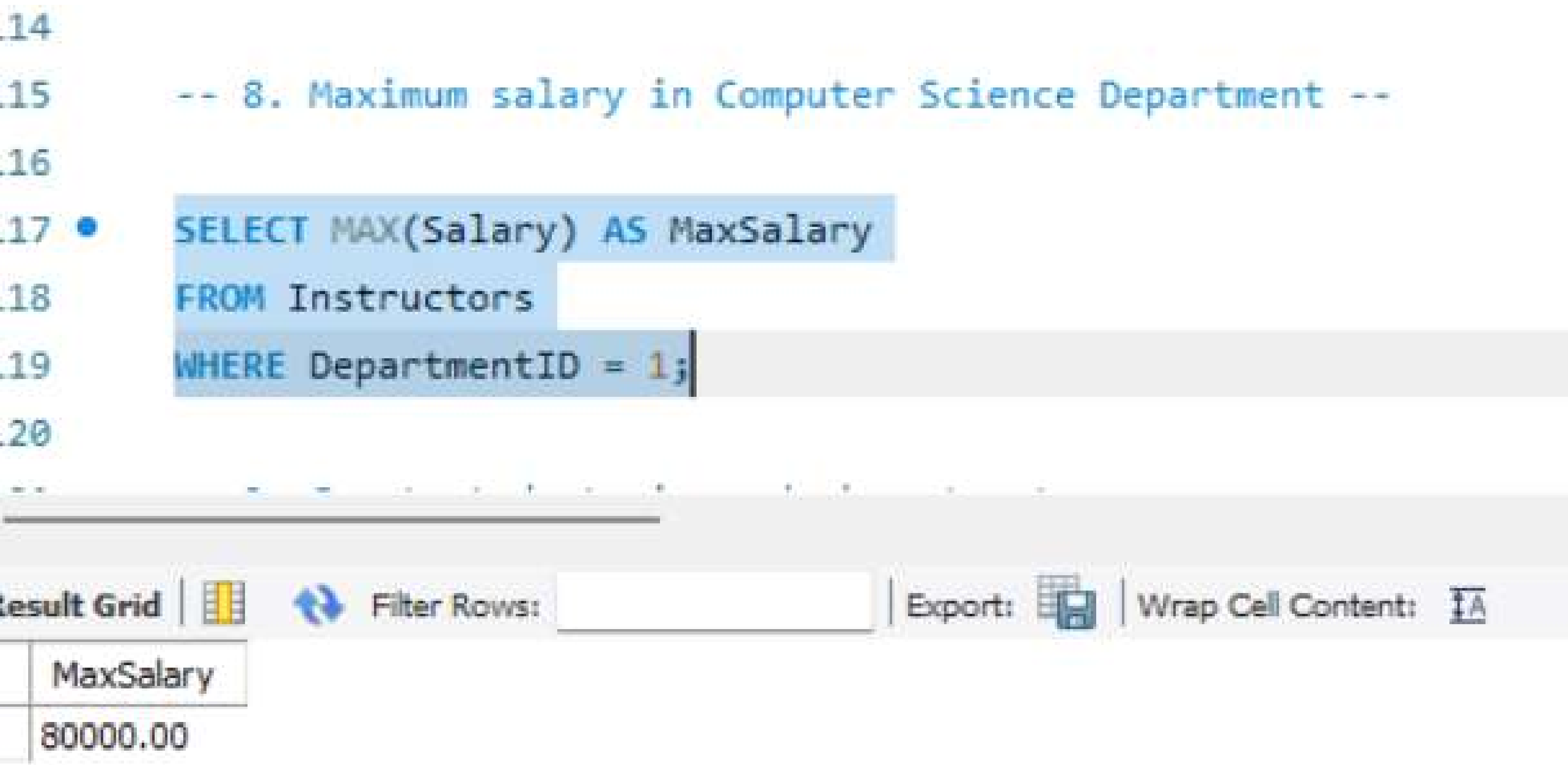
```sql
-- 6. Students enrolled in SQL OR Data Structures --

SELECT DISTINCT s.StudentID, s.FirstName
FROM Students s
JOIN Enrollments e ON s.StudentID = e.StudentID
WHERE e.CourseID IN (101,102);
```

StudentID	FirstName
1	jayal
2	raj
3	het



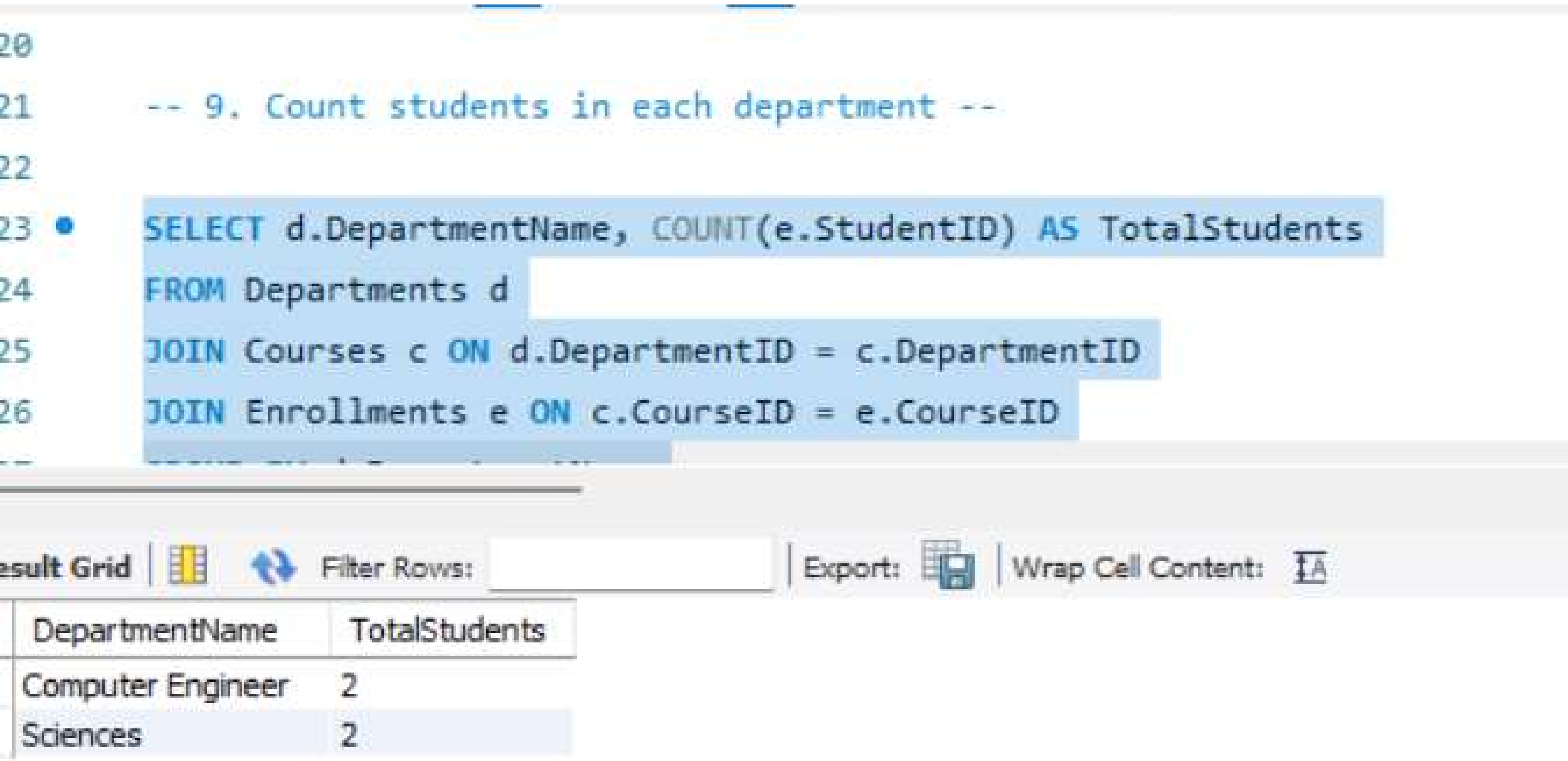
```
-- 7. Average credits of all courses --

SELECT AVG(Credits) AS AvgCredits FROM Courses;
```

```
AvgCredits
3.5000
```



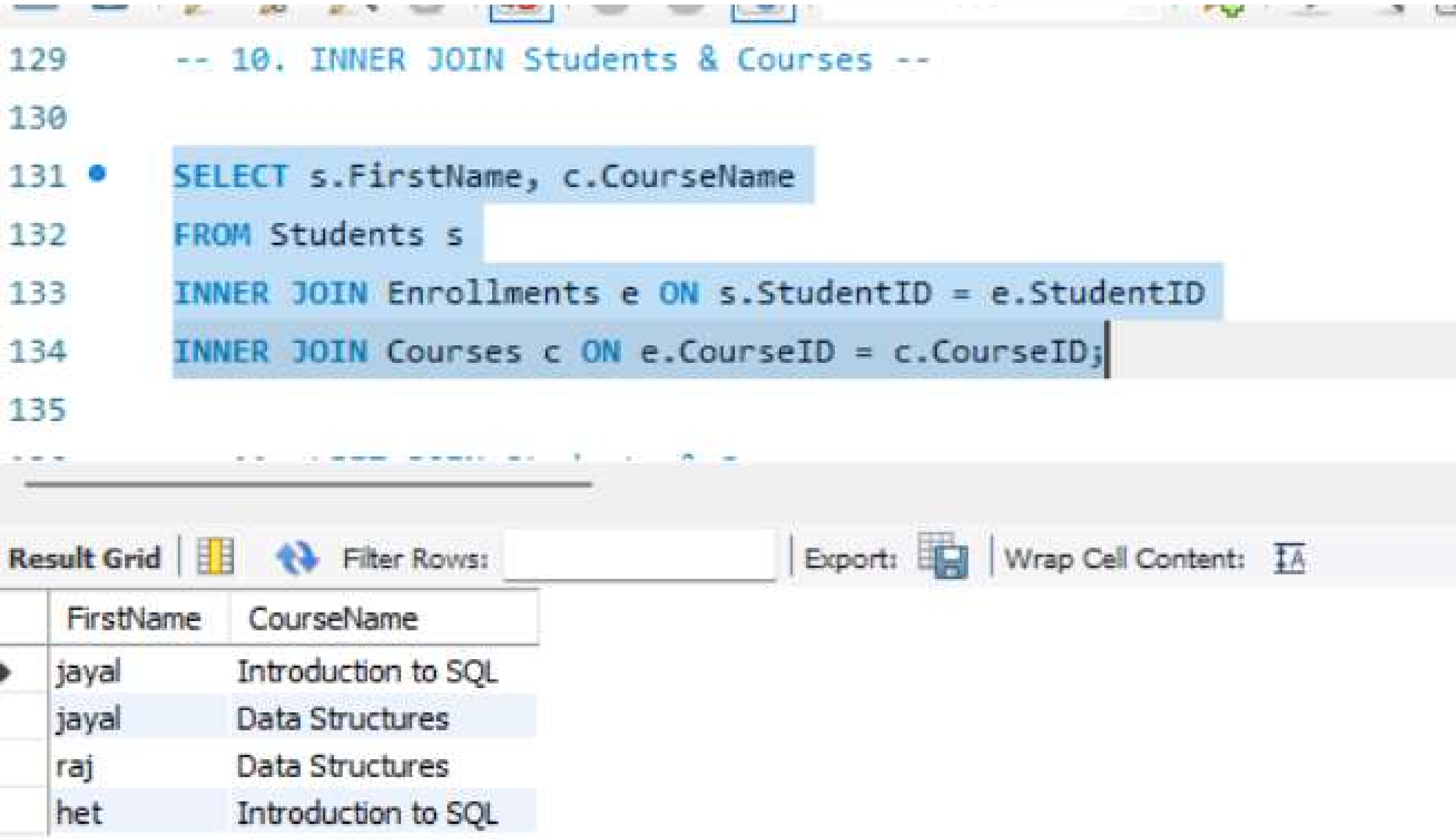
```sql
-- 8. Maximum salary in Computer Science Department --

SELECT MAX(Salary) AS MaxSalary
FROM Instructors
WHERE DepartmentID = 1;
```

```
MaxSalary
80000.00
```



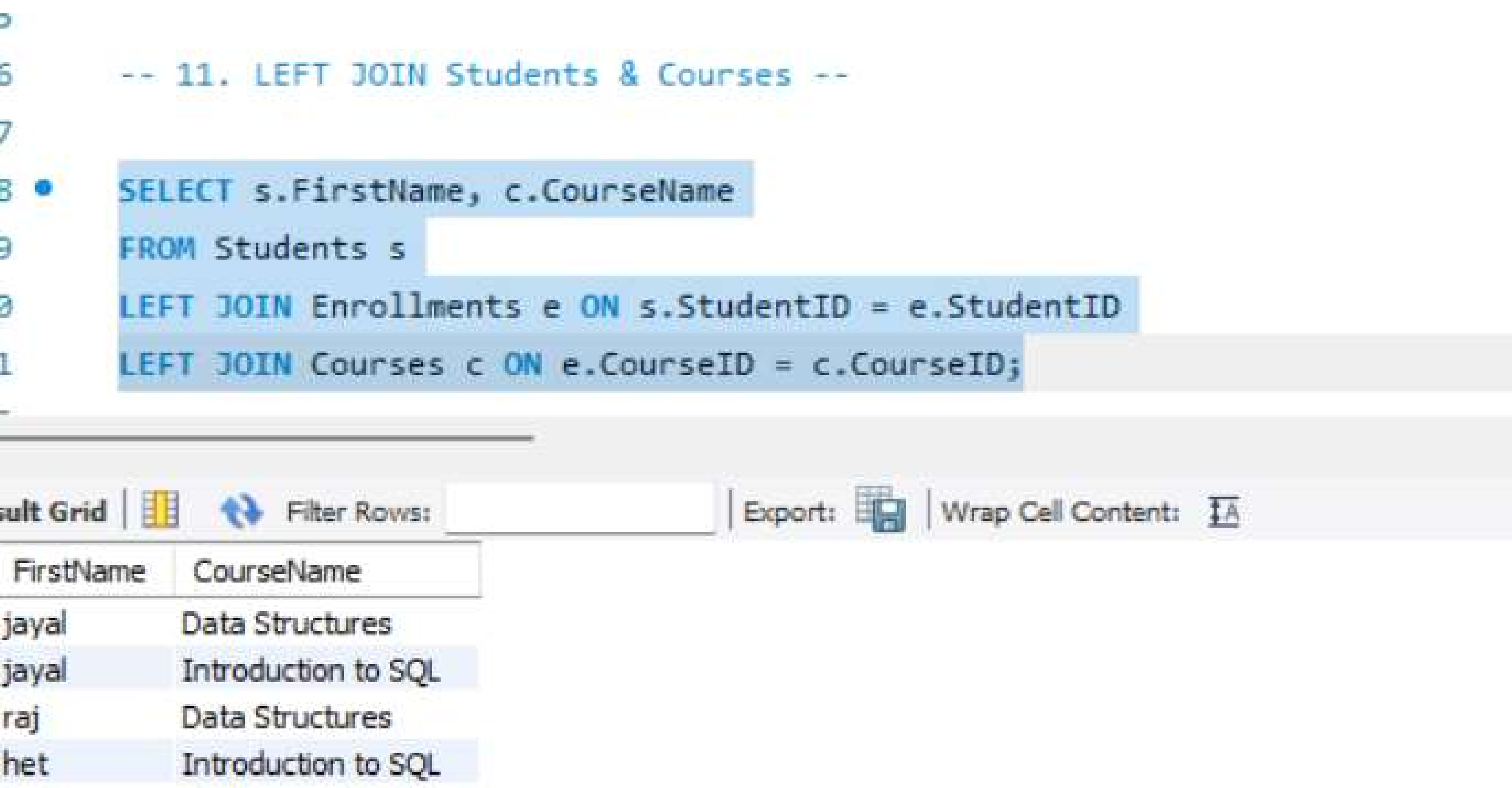
```sql
-- 9. Count students in each department --

SELECT d.DepartmentName, COUNT(e.StudentID) AS TotalStudents
FROM Departments d
JOIN Courses c ON d.DepartmentID = c.DepartmentID
JOIN Enrollments e ON c.CourseID = e.CourseID
```

DepartmentName	TotalStudents
Computer Engineer	2
Sciences	2



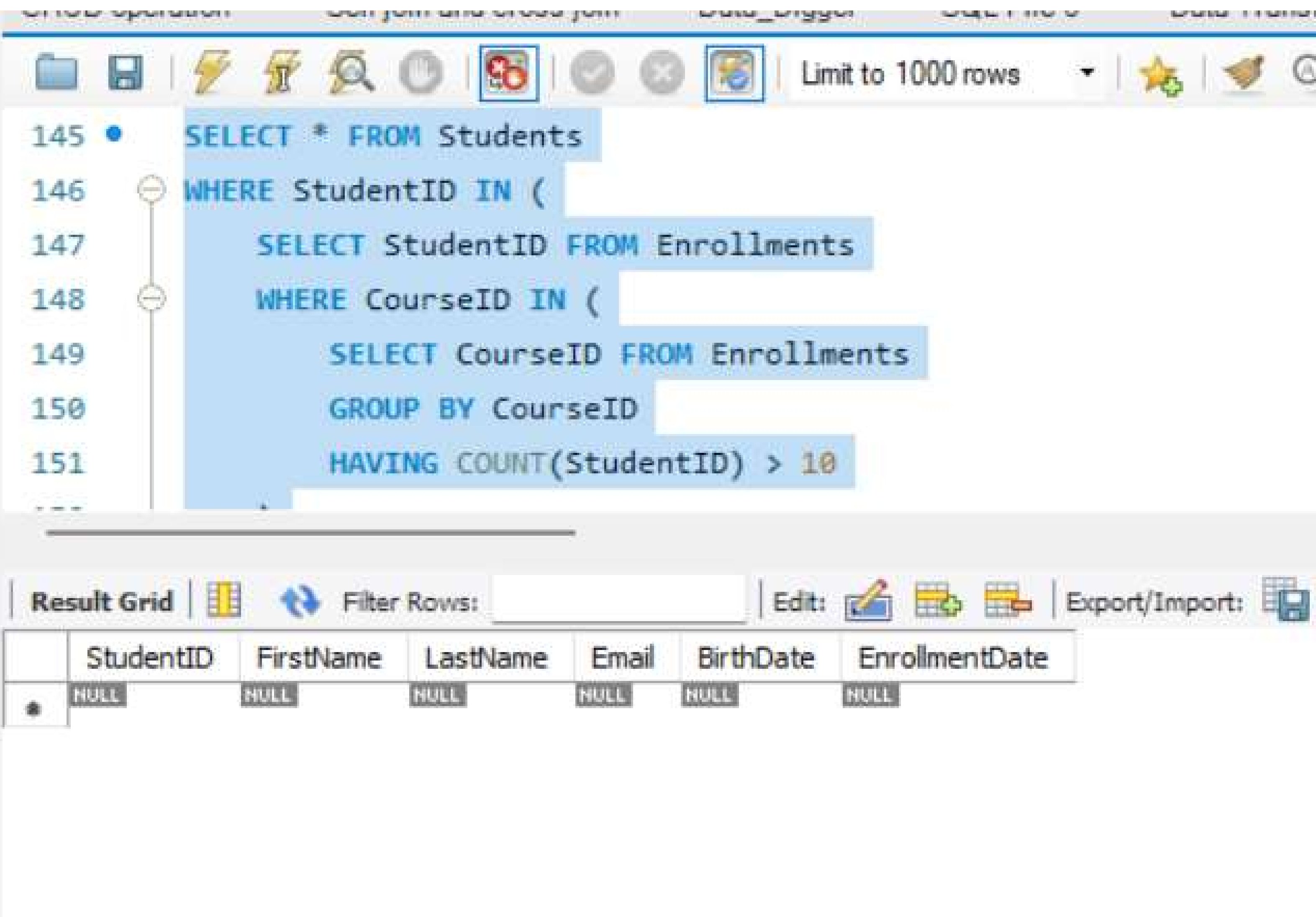
```
-- 10. INNER JOIN Students & Courses --

SELECT s.FirstName, c.CourseName
FROM Students s
INNER JOIN Enrollments e ON s.StudentID = e.StudentID
INNER JOIN Courses c ON e.CourseID = c.CourseID;
```

FirstName	CourseName
jayal	Introduction to SQL
jayal	Data Structures
raj	Data Structures
het	Introduction to SQL



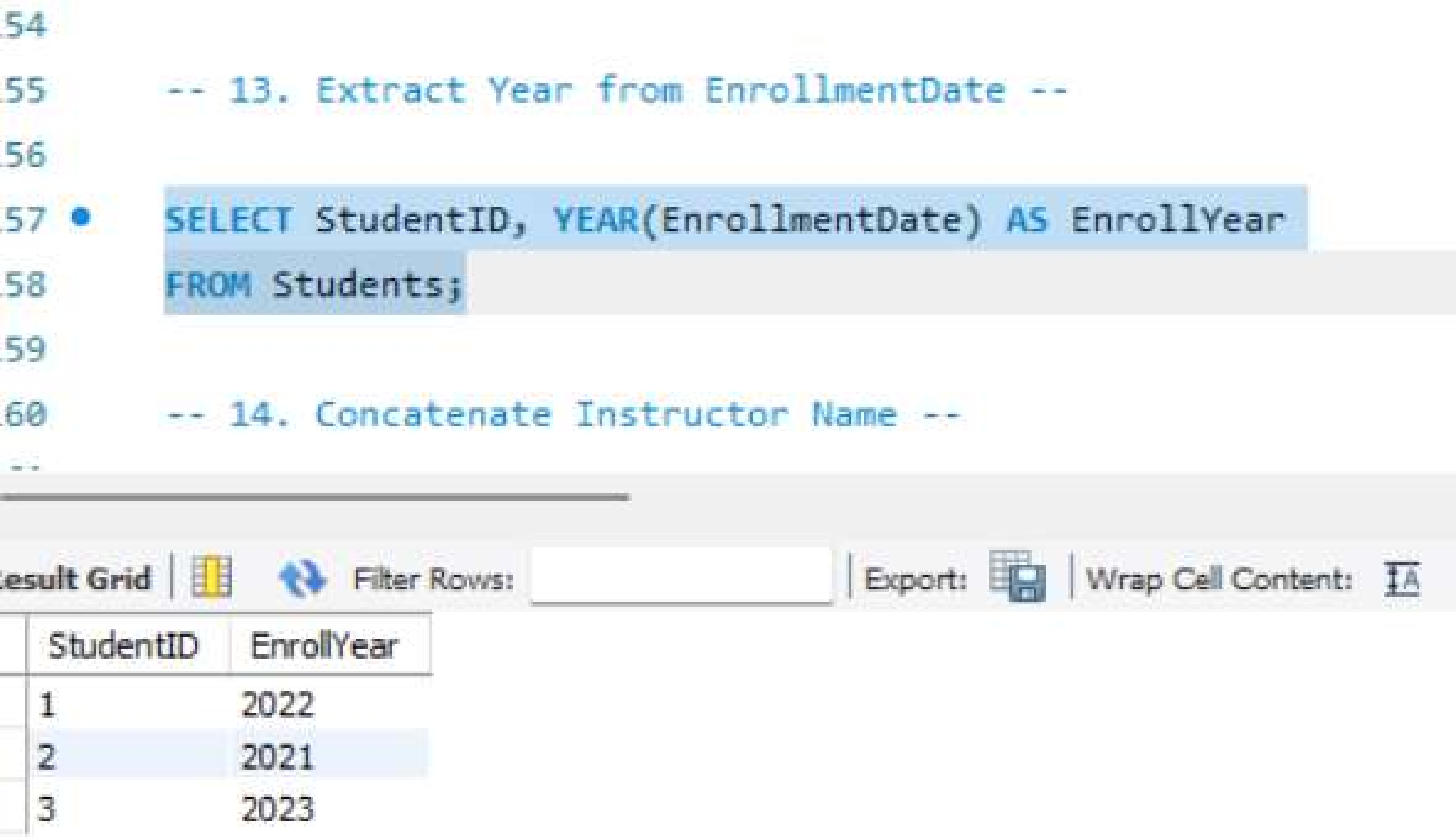
```
-- 11. LEFT JOIN Students & Courses --

SELECT s.FirstName, c.CourseName
FROM Students s
LEFT JOIN Enrollments e ON s.StudentID = e.StudentID
LEFT JOIN Courses c ON e.CourseID = c.CourseID;
```

FirstName	CourseName
jayal	Data Structures
jayal	Introduction to SQL
raj	Data Structures
het	Introduction to SQL



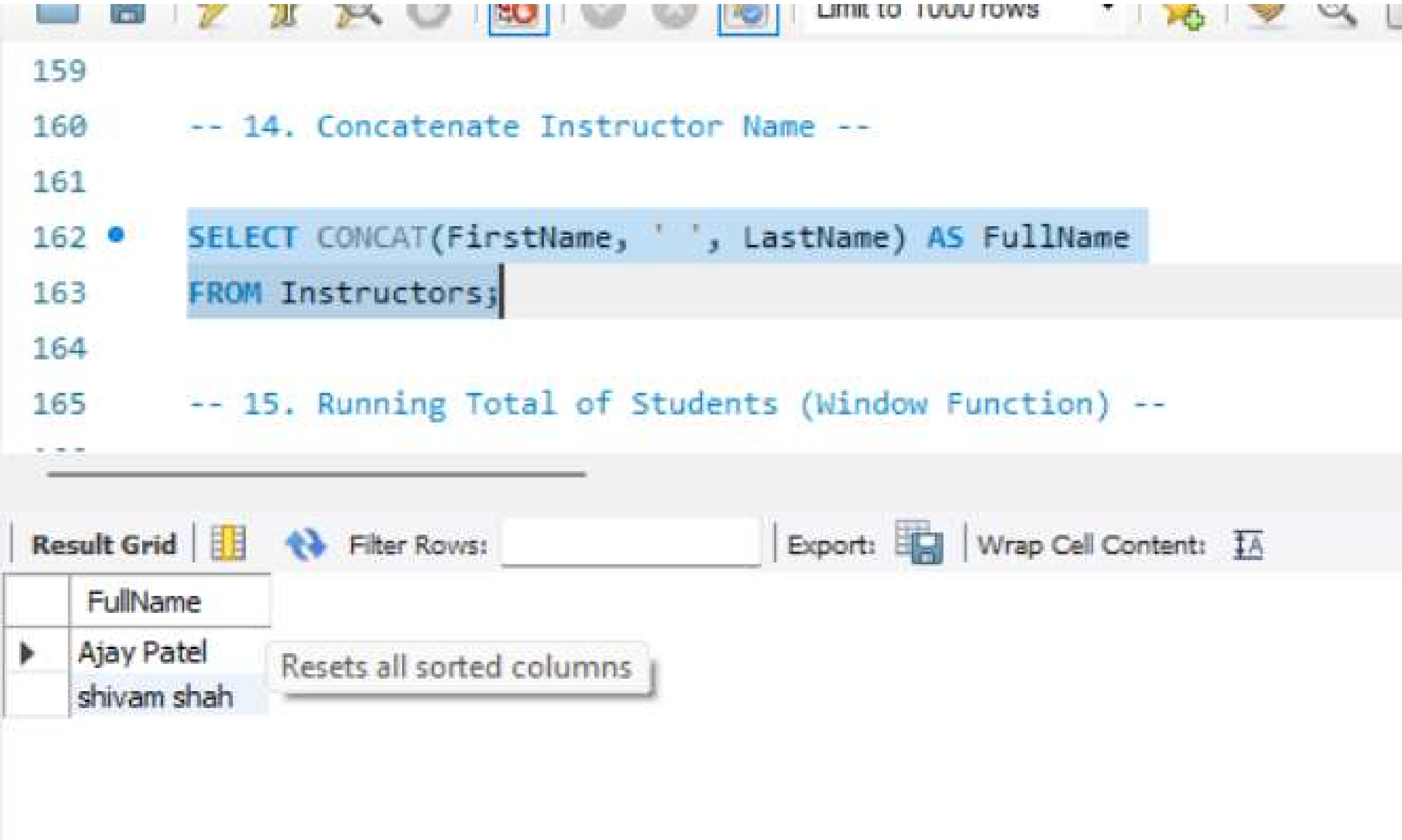
```sql
SELECT * FROM Students
WHERE StudentID IN (
    SELECT StudentID FROM Enrollments
    WHERE CourseID IN (
        SELECT CourseID FROM Enrollments
        GROUP BY CourseID
        HAVING COUNT(StudentID) > 10
```

StudentID	FirstName	LastName	Email	BirthDate	EnrollmentDate
NULL	NULL	NULL	NULL	NULL	NULL



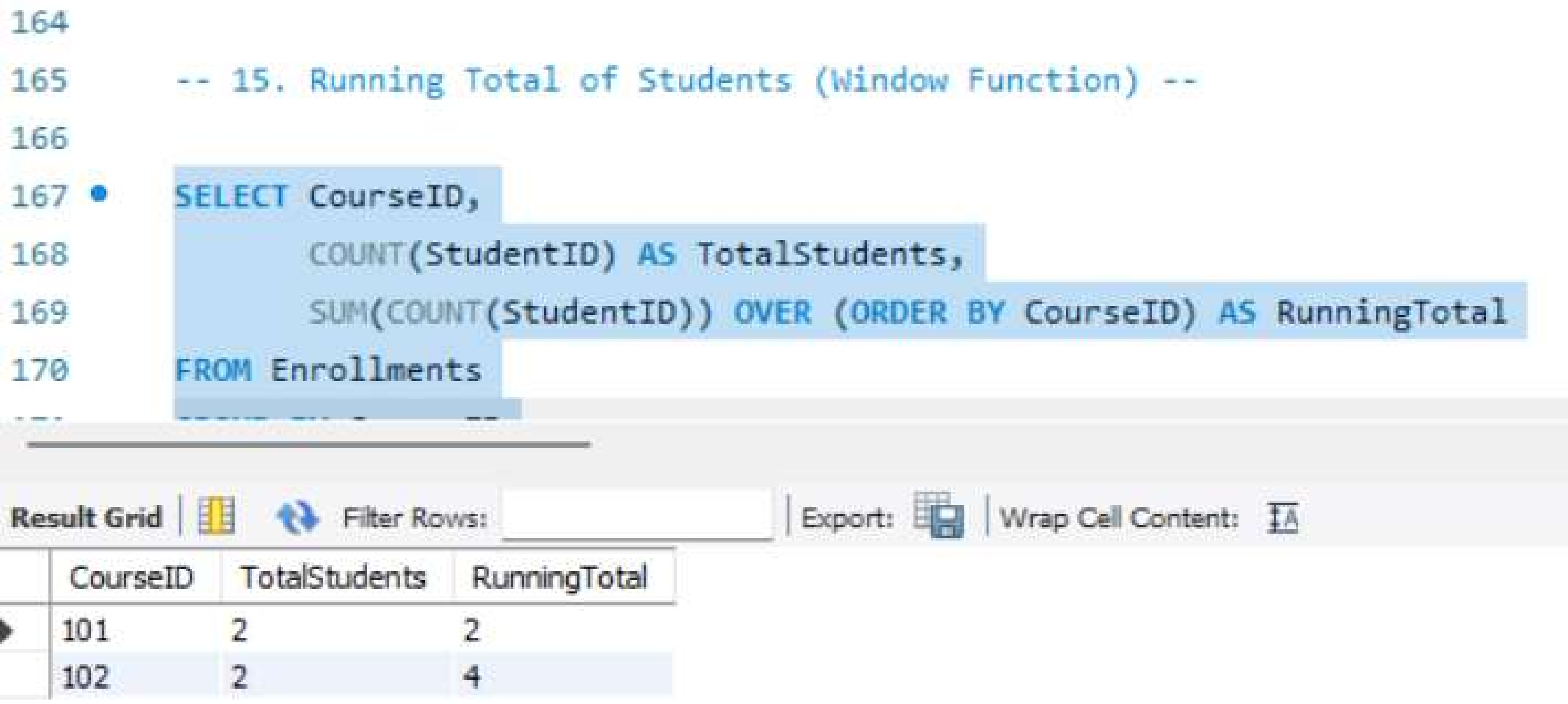
```
-- 13. Extract Year from EnrollmentDate --

SELECT StudentID, YEAR(EnrollmentDate) AS EnrollYear
FROM Students;

-- 14. Concatenate Instructor Name --
```

StudentID	EnrollYear
1	2022
2	2021
3	2023



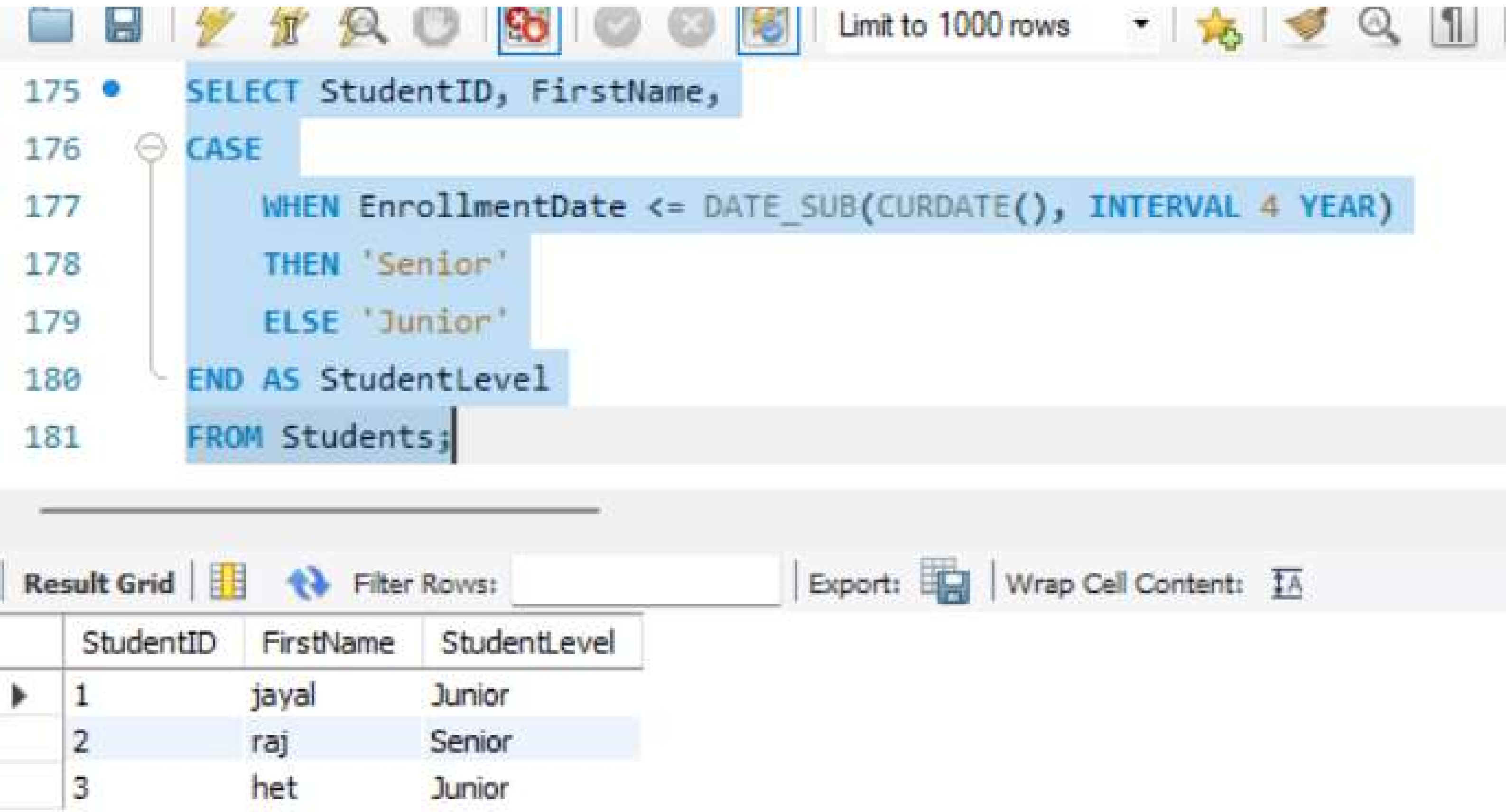
```sql
-- 14. Concatenate Instructor Name --

SELECT CONCAT(FirstName, ' ', LastName) AS FullName
FROM Instructors;

-- 15. Running Total of Students (Window Function) --
```

```
FullName
Ajay Patel
shivam shah
```



```sql

-- 15. Running Total of Students (Window Function) --

SELECT CourseID,
       COUNT(StudentID) AS TotalStudents,
       SUM(COUNT(StudentID)) OVER (ORDER BY CourseID) AS RunningTotal
FROM Enrollments
```

CourseID	TotalStudents	RunningTotal
101	2	2
102	2	4



```sql
SELECT StudentID, FirstName,
CASE
    WHEN EnrollmentDate <= DATE_SUB(CURDATE(), INTERVAL 4 YEAR)
    THEN 'Senior'
    ELSE 'Junior'
END AS StudentLevel
FROM Students;
```

StudentID	FirstName	StudentLevel
1	jayal	Junior
2	raj	Senior
3	het	Junior