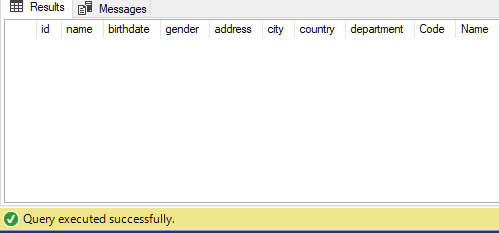
--1. Retrieves all students who do not have a specified department.

SELECT \* FROM Students AS S

LEFT JOIN Departments AS D ON S.department = D.Code

WHERE D.Code IS NULL;



--2 Creates a view named "StudentDetails" that combines data from the Students, enroll, and Courses tables.

CREATE VIEW StudentDetails AS

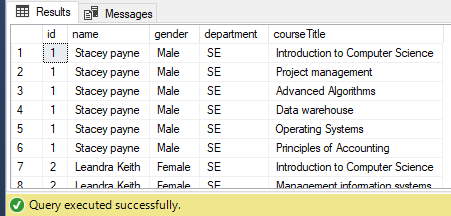
SELECT S.id, S.name, S.gender, S.department, C.title AS courseTitle

FROM Students S,enroll E, Courses C

where S.id = E.studentId and E.courseId = C.id;

-- test

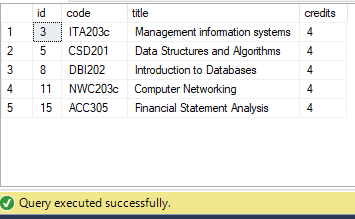
select \* from dbo.StudentDetails



--3. Retrieves the top 5 rows from the Courses table

SELECT TOP 5 \* FROM Courses

ORDER BY credits DESC;

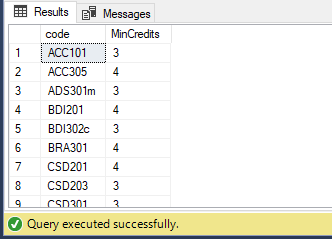


--4. Calculates the minimum number of credits for each department

SELECT code, MIN(credits) AS MinCredits

FROM Courses

GROUP BY code;



--5. Retrieves the maximum mark obtained by each student

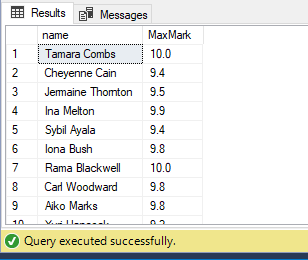
SELECT s.name, MAX(marks.mark) AS MaxMark

FROM Students s

JOIN enroll e ON s.id = e.studentId

JOIN marks ON e.enrollId = marks.enrollId

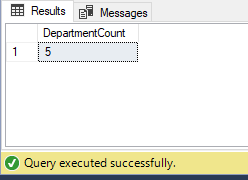
GROUP BY s.name;



--6. Counts the number of distinct departments that are present in the Students table.

SELECT COUNT(DISTINCT department) AS DepartmentCount

FROM Students;



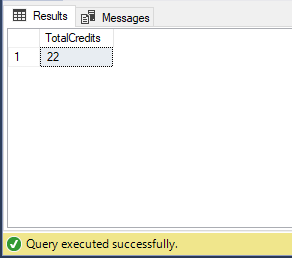
--7. Calculates the total credits of courses that a specific student (with id 5) has enrolled in.

SELECT SUM(C.credits) AS TotalCredits

FROM Courses C

INNER JOIN enroll E ON C.id = E.courseId

WHERE E.studentId = 5;



--8. Find the student who have the highest avgerage score

SELECT TOP 1 S.id, S.name, AVG(M.mark) AS AverageScore

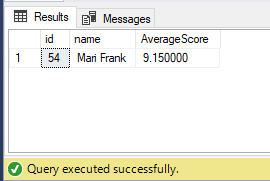
FROM Students S

JOIN enroll E ON S.id = E.studentId

JOIN marks M ON E.enrollId = M.enrollId

GROUP BY S.id, S.name

ORDER BY AverageScore DESC;



--9 This stored procedure takes a department code as input and retrieves the count of students in that department.

CREATE PROCEDURE GetStudentCountByDepartment

@departmentCode VARCHAR(10)

AS

BEGIN

declare @n int

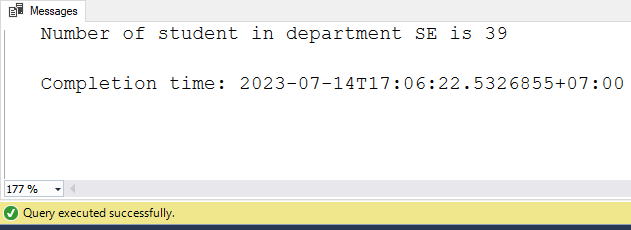
SELECT @n = COUNT(\*) FROM Students WHERE department = @departmentCode;

print concat (N'Number of student in department ',@departmentCode, ' is ', @n)

END

-- Test

exec GetStudentCountByDepartment @departmentCode = 'SE'



--10 Trigger that prevent changing name of any male student

create trigger StopChangeMaleName

on dbo.Students

for update

as

begin

declare @count\_male int

select @count\_male = count(\*) from inserted where gender = 'Male'

if (@count\_male > 0)

begin

print N'Không được sửa tên sinh viên nam'

rollback tran

end

end

--test case

update dbo.Students set name = 'pikachu' where name ='Stacey payne'

