Lab#2 CS4350 Database Systems

**Name: Loc Nguyen**

Given the following relational database schema:

Student = (SSN, Name, Major)

Course = ( CourseNumber, PrerequisiteCourseNumber, Course Title, NumberUnits)

Section = ( CourseNumber, Quarter, RoomNumber, DayTime), where DayTime is of the form MW 1:00-2:00PM.

Enrollment = (SSN,CourseNumber, Quarter, Grade)// Grade is either Null or a letter grade.

Express the following queries by SQL statements and test them using any appropriate database product. Submit screenshots of your SQL statements and their outputs. Create your own table to test your SQL statements. Submit screenshots of the SQL statements and its output.

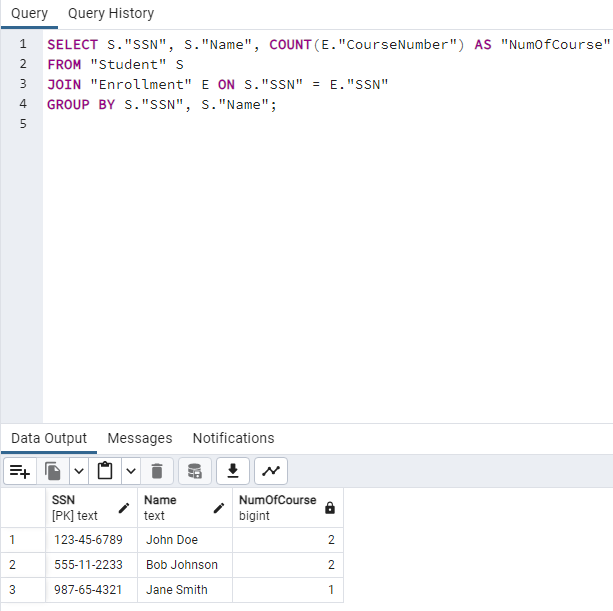
1. List the name, SSN and the number of courses the student has taken (courses with the same CourseNumber taken in different quarters are counted as different courses)

SELECT S.SSN, S.Name, COUNT(E.CourseNumber) AS NumOfCourse

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

GROUP BY S.SSN, S.Name;



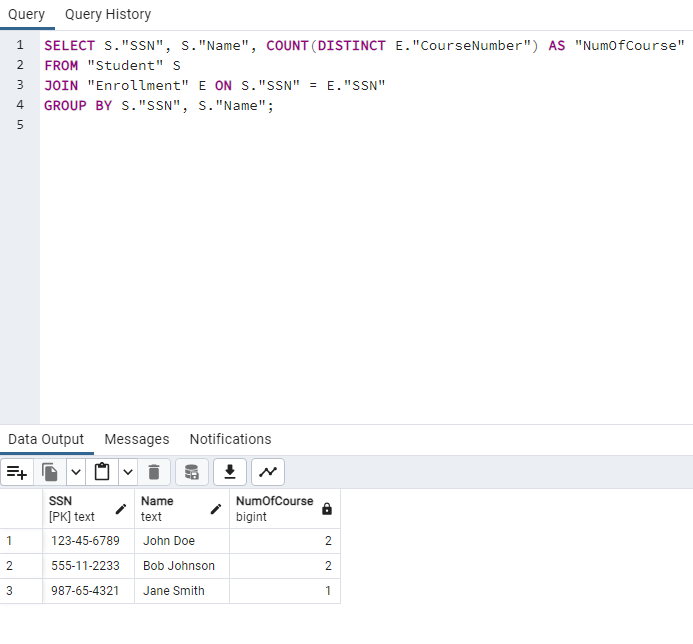
1. Answer #1 assuming courses with the same CourseNumber taken in different quarters are considered as one course

SELECT S.SSN, S.Name, COUNT(DISTINCT E.CourseNumber) AS NumOfCourse

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

GROUP BY S.SSN, S.Name;



1. List the name, SSN and number of courses the student has taken and completed

i.e. received a letter grade (courses with the same CourseNumber taken in

different quarters are counted as different courses).

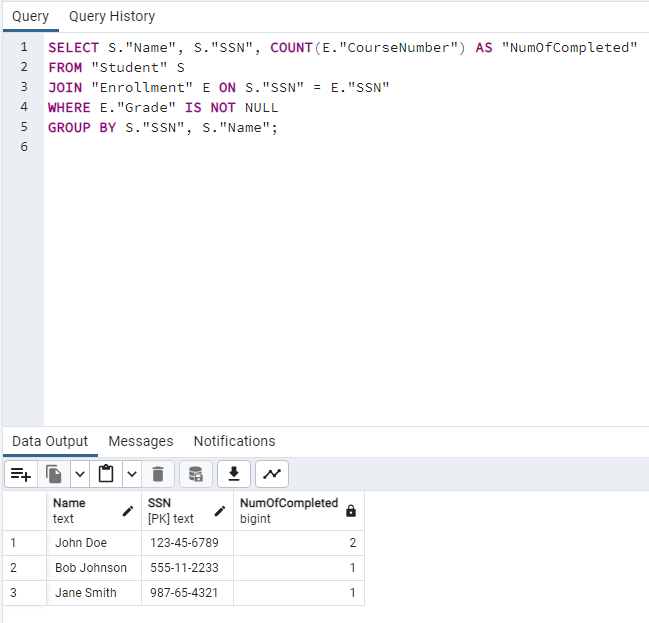
SELECT S.Name, S.SSN, COUNT(E.CourseNumber) AS NumOfCompleted

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

WHERE E.Grade IS NOT NULL

GROUP BY S.SSN, S.Name;



1. List the name, SSN and number of courses the student has taken and completed

with a letter grade C or better (courses with the same CourseNumber taken in

different quarters are counted as different courses).

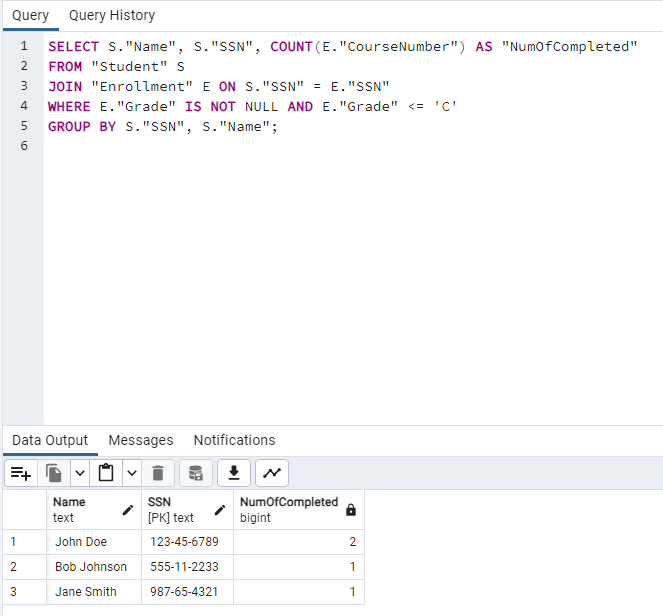
SELECT S.Name, S.SSN, COUNT(E.CourseNumber) AS NumOfCompleted

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

WHERE E.Grade <= ‘C’ AND E.Grade IS NOT NULL

GROUP BY S.SSN, S.Name;

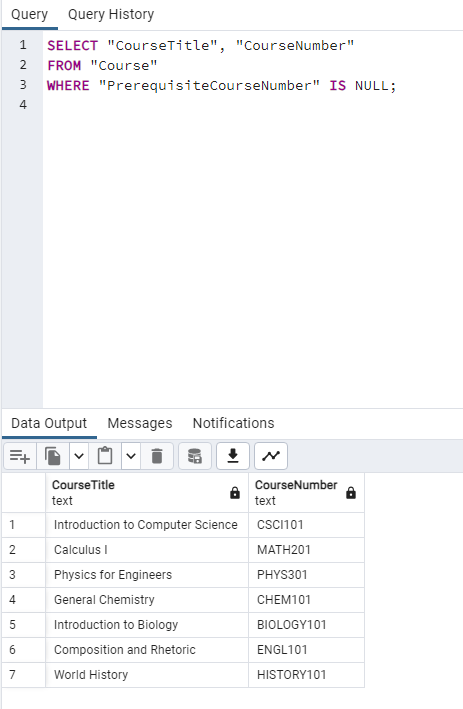


1. List the Course Title and CourseNumber which does not have any prerequisite.

SELECT C.CourseTitle, C.CourseNumber

FROM Course C

WHERE C.PrerequisiteCourseNumber IS NULL;



1. List the name of every student and SSN who earned an A in all courses he or she has completed, i.e. every grade is either Null or an A)

SELECT S.Name, S.SSN

FROM Student S

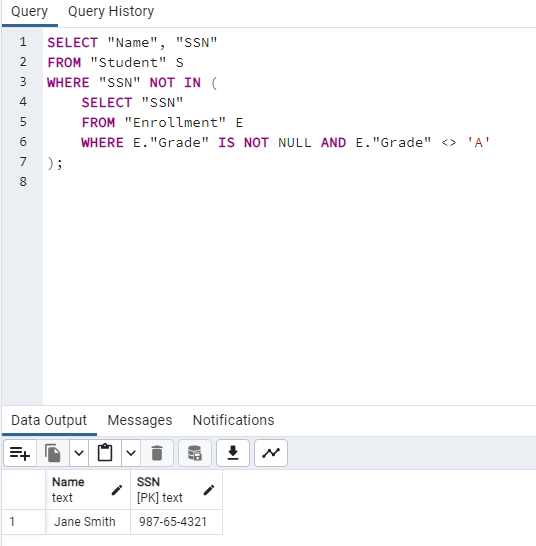
WHERE S.SSN NOT IN (

SELECT E.SSN

FROM Enrollment E

WHERE E.Grade IS NOT NULL AND E.Grade <> ‘A’

);



1. List the name of every student, SSN and the CourseNumber who has taken the course at least three times.

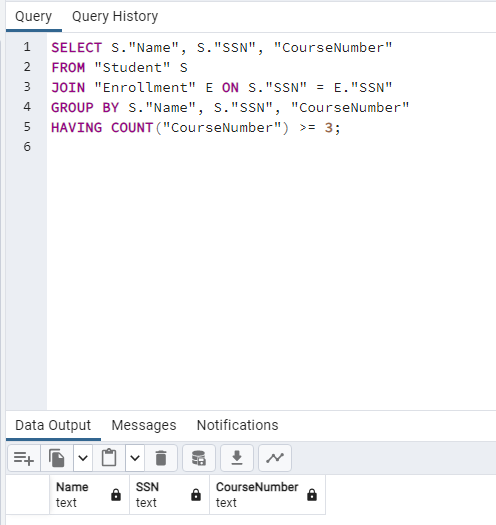
SELECT S.Name, S.SSN, E.CourseNumber

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

GROUP BY S.Name, S.SSN, E.CourseNumber

HAVING COUNT(E.CourseNumber) >= 3;



1. List the name of every student, SSN, every quarter, and the number of courses he or she has taken in the listed quarter.

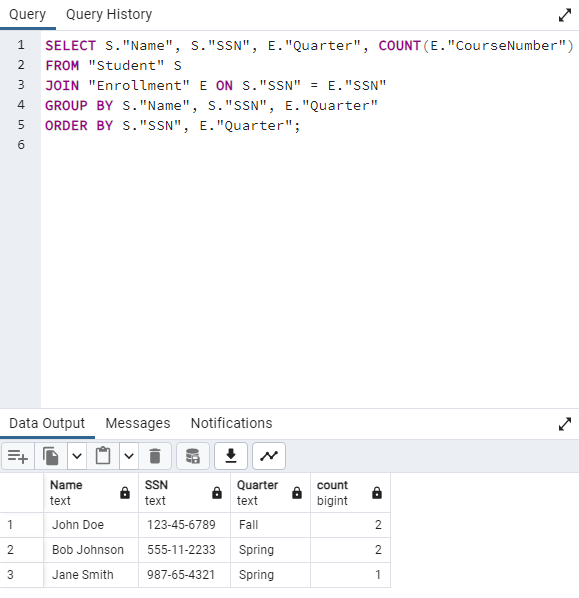
SELECT S.Name, S.SSN, E.Quarter, COUNT(E.CourseNumber)

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

GROUP BY S.Name, S.SSN, E.Quarter

ORDER BY S.SSN, E.Quarter;



1. List the name of every student and SSN who has not taken any course more than once

SELECT S.Name, S.SSN

FROM Student S

WHERE S.SSN NOT IN (

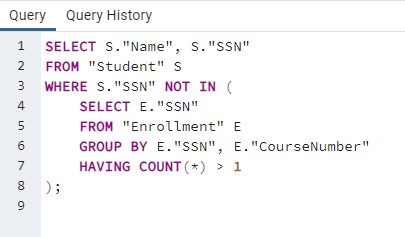
SELECT E.SSN

FROM Enrollment E

GROUP BY E.SSN, E.CourseNumber

HAVING COUNT(\*) > 1

);



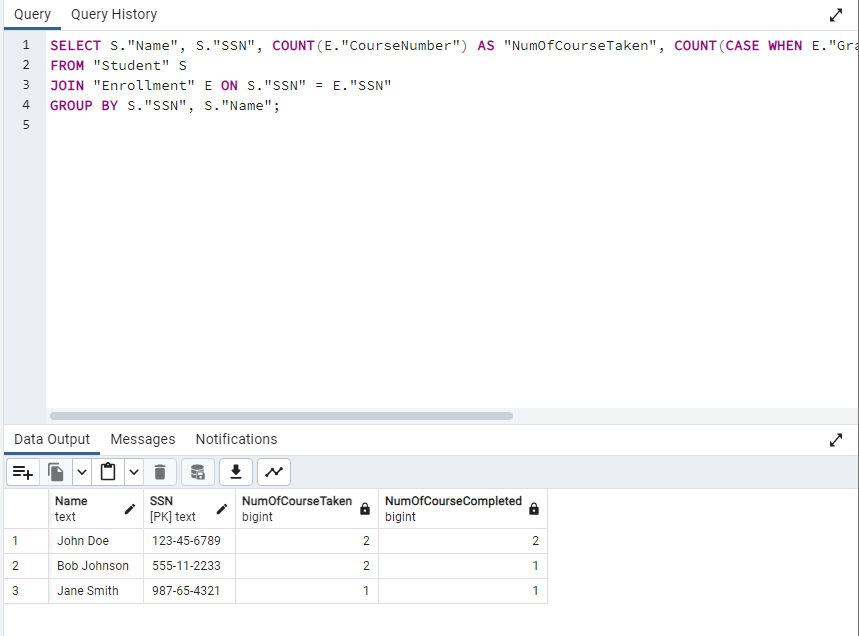
1. List the name, SSN , the number of courses the student has taken, and the number of courses completed(courses with the same CourseNumber taken in different quarters are counted as different courses).

SELECT S.Name, S.SSN, COUNT(E.CourseNumber) AS NumOfCourseTaken, COUNT (CASE WHEN E.Grade IS NOT NULL THEN E.CourseNumber END) AS NumOfCourseCompleted

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

GROUP BY S.SSN, S.Name;



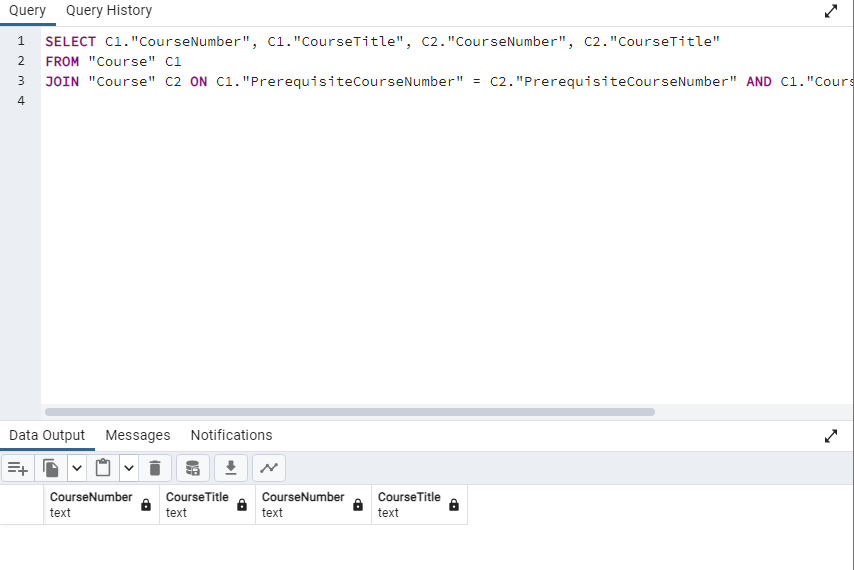
1. List every two CourseNumber and their titles which have the same prerequisites

SELECT C1.CourseNumber, C1.CourseTitle, c2.CourseNumber, C2.CourseTitle

FROM Course C1

JOIN Course C2 ON C1.PrerequisiteCourseNumber = C2.PrerequisiteCourseNumber AND C1.CourseNumber < C2.CourseNumber;

\*\*\*Prevents duplicates when C2.CourseNumber is > C1.CourseNumber



1. List the name of every student and SSN who has completed all the courses he/she has taken and earned an A in each course, i.e. every grade is an A and no NULL value

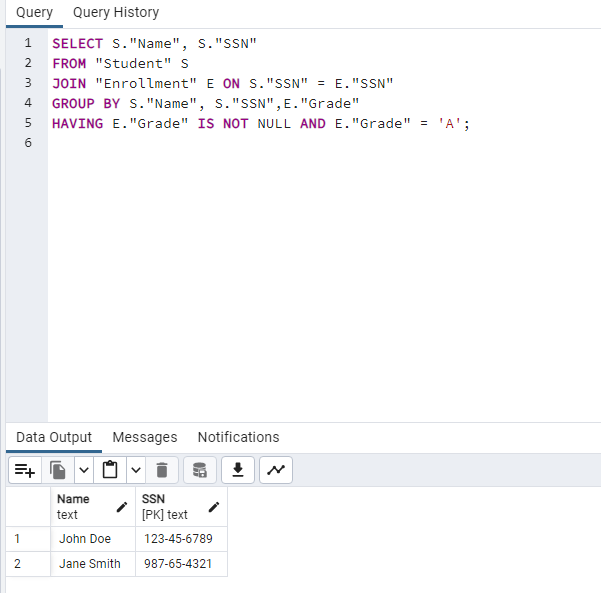
SELECT S.Name, S.SSN

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

GROUP BY S.Name, S.SSN, E.Grade

HAVING E.Grade IS NOT NULL AND E.Grade = ‘A’;



1. List the name of every student and SSN who earned no A in any course.

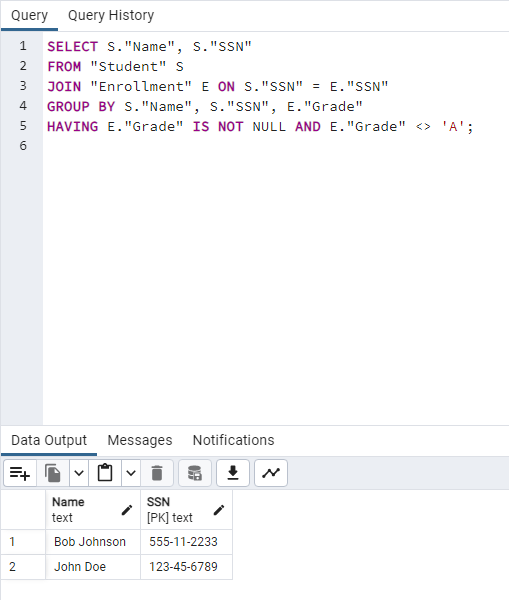
SELECT S.Name, S.SSN

FROM Student S

JOIN Enrollment E ON S.SSN = E.SSN

GROUP BY S.Name, S.SSN, E.Grade

HAVING E.Grade IS NOT NULL AND E.Grade <> ‘A’;



1. List the name and major of every student who has only taken courses that meet MW afternoon (12 or after).

SELECT St.Name, St.Major

FROM Student St

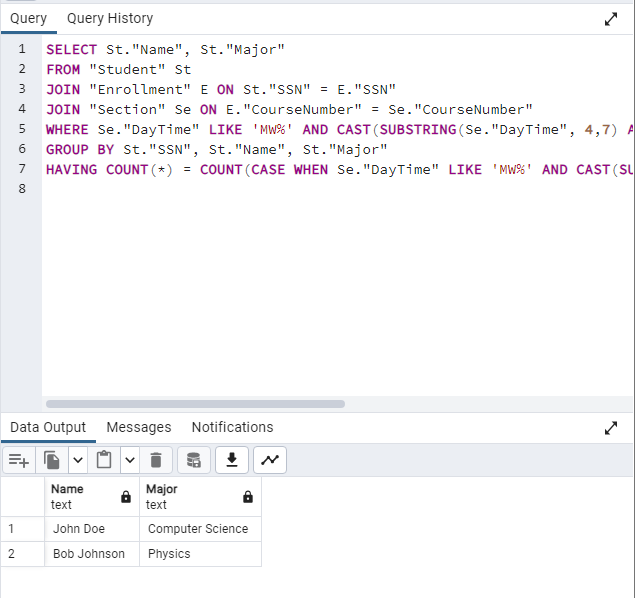
JOIN Enrollment E ON St.SSN = E.SSN

JOIN Section Se ON E.CourseNumber = Se.CourseNumber

WHERE Se.DayTime LIKE ‘MW%’ AND CAST(SUBSTRING(Se.Daytime, 4, 7) AS TIME) >= ’12:00:00’

GROUP BY St.SSN, St.Name, St.Major

HAVING COUNT(\*) = COUNT(CASE WHEN Se.DayTime LIKE ‘MW%’ AND CAST(SUBSTRING(Se.Daytime, 4, 7) AS TIME) >= ’12:00:00’ THEN 1 END)



1. List the name and major of every student who has taken all the courses that meet MW afternoon.

SELECT St.Name, St.Major

FROM Student St

JOIN Enrollment E ON St.SSN = E.SSN

JOIN Section Se ON E.CourseNumber = Se.CourseNumber

WHERE Se.DayTime LIKE ‘MW%’ AND CAST(SUBSTRING(Se.Daytime, 4, 7) AS TIME) >= ’12:00:00’

GROUP BY St.SSN, St.Name, St.Major

HAVING COUNT(DISTINCT E.CourseNumber) = (

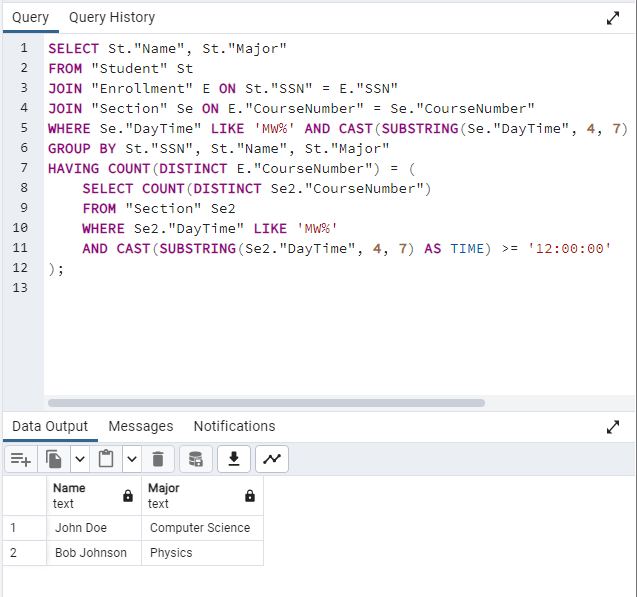
SELECT COUNT(DISTINCT Se2.CourseNumber)

FROM Section Se2

WHERE Se2.DayTime LIKE ‘MW%’

AND CAST(SUBSTRING(Se2.DayTime, 4, 7) AS TIME) >= ’12:00:00’

);



1. List the name and major of every student who has not taken any course that meets MW afternoon.

SELECT S.Name, S.Major

FROM Student S

WHERE S.SSN NOT IN (

SELECT DISTINCT S2.SSN

FROM Student S2

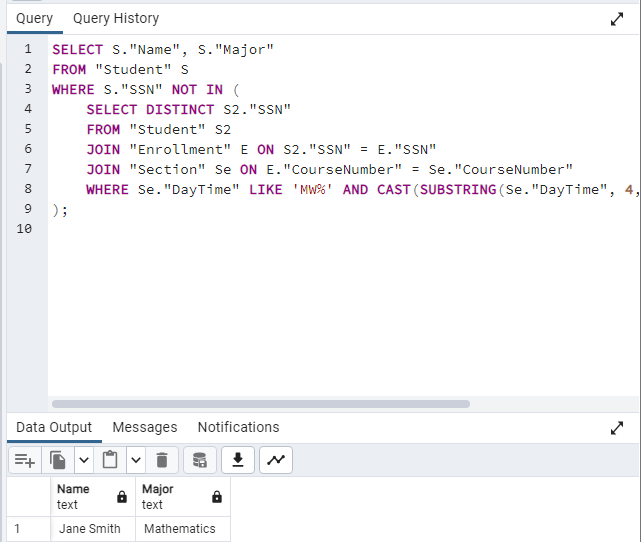
JOIN Enrollment E ON S2.SSN = E.SSN

JOIN Section Se ON E.CourseNumber = Se.CourseNumber

WHERE Se2.DayTime LIKE ‘MW%’

AND CAST(SUBSTRING(Se2.DayTime, 4, 7) AS TIME) >= ’12:00:00’

);



1. List every CourseNumber and Quarter which has the highest enrollment.

SELECT E.CourseNumber, E.Quarter

FROM Enrollment E

GROUP BY CourseNumber, Quarter

HAVING COUNT(\*) = (

SELECT COUNT(\*)

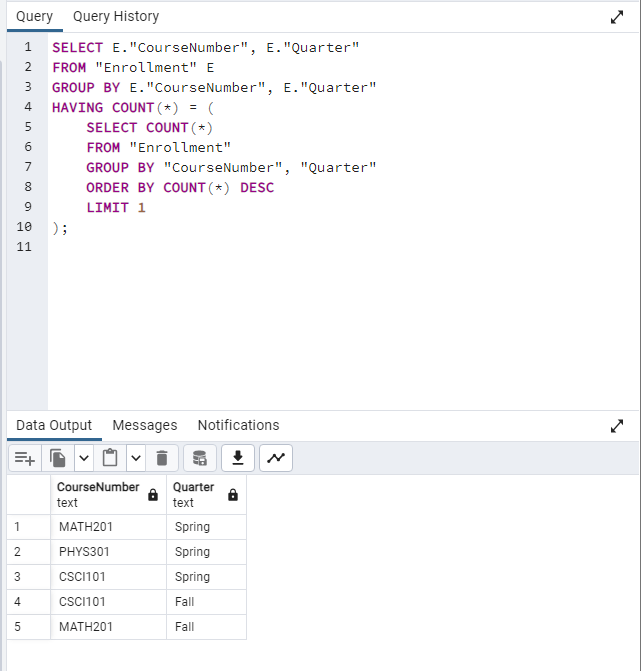
FROM Enrollment

GROUP BY CourseNumber, Quarter

ORDER BY COUNT(\*) DESC

LIMIT 1

);



1. List every CourseNumber and CourseTitle which has the highest enrollment based on all quarters.

SELECT E.CourseNumber, C.CourseTitle

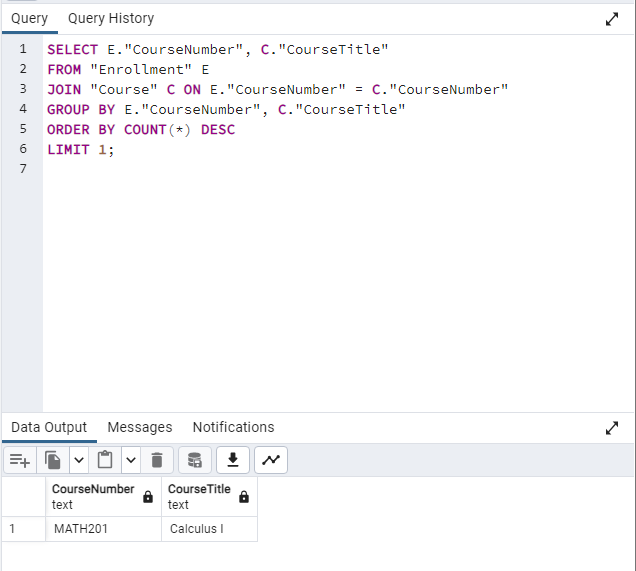
FROM Enrollment E

JOIN Course C ON E.CourseNumber = C.CourseNumber

GROUP BY E.CourseNumber, C.CourseTitle

ORDER BY COUNT(\*) DESC

LIMIT 1;



1. List the name and major of every student who has completed the highest number of units

SELECT S.Name, S.Major

FROM Student S

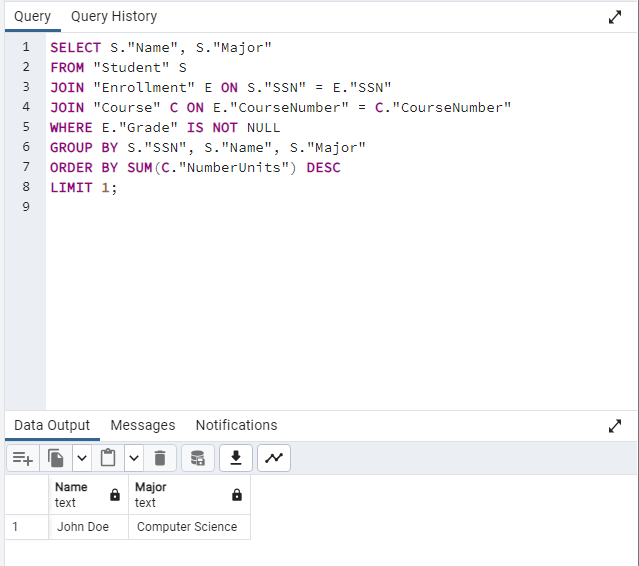
JOIN Enrollment E ON S.SSN = E.SSN

JOIN Course C ON E.CourseNumber = C.CourseNumber

WHERE E.Grade IS NOT NULL  
GROUP BY S.SSN, S.Name, S.Major

ORDER BY SUM(C.NumberUnits) DESC

LIMIT 1;



1. List every Course title which is a prerequisite for the largest number of courses

SELECT C.CourseTitle

FROM Course C

WHERE C.CourseNumber IN (

SELECT PrerequisiteCourseNumber

FROM Course

WHERE PrerequisiteCourseNumber IS NOT NULL

GROUP BY PrerequisiteCourseNumber

ORDER BY COUNT(DISTINCT CourseNumber) DESC

LIMIT 1

);

