Group Number -

Group Members -

Questions

1- Create all the tables as you proposed and designed according to your group ERD.

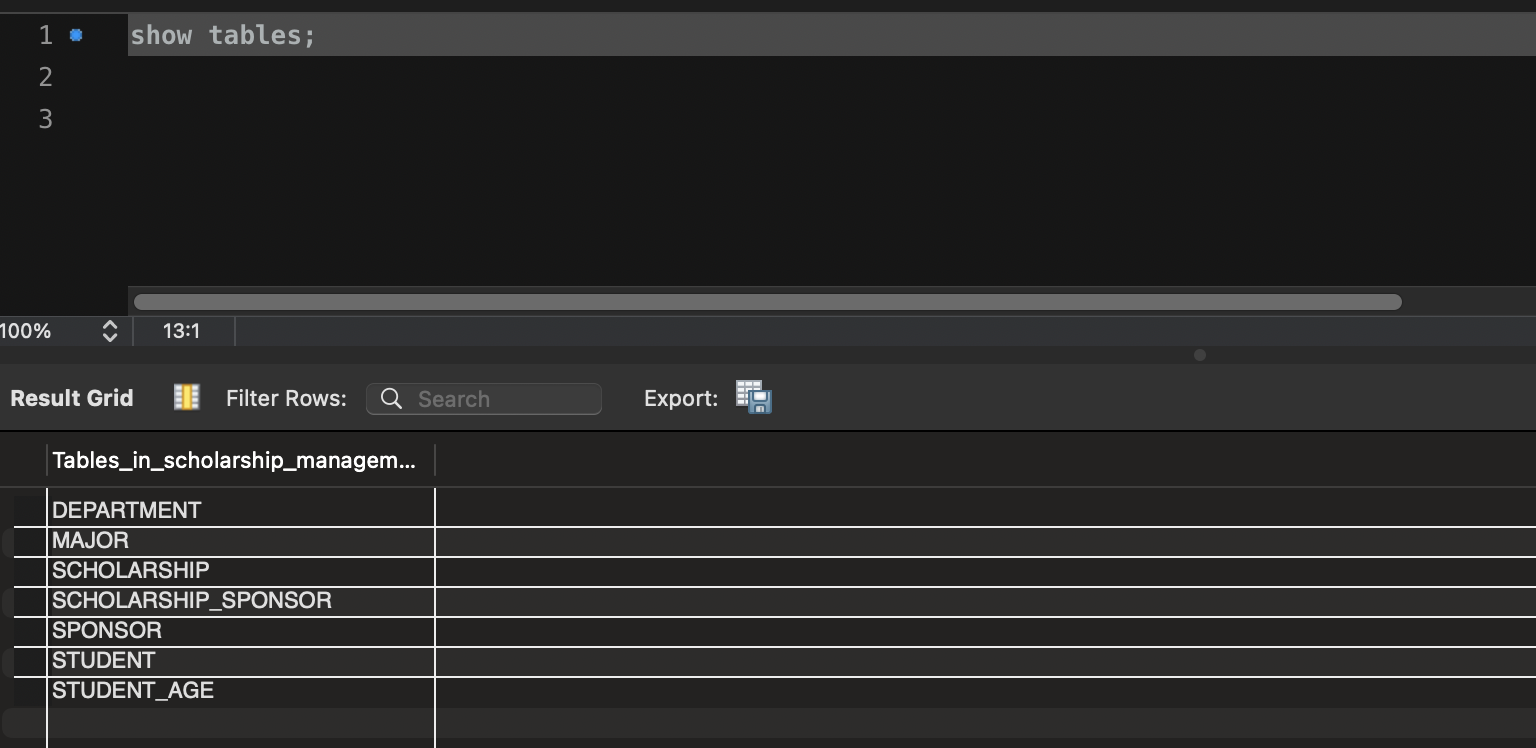
2- Make sure you specify the primary and foreign keys and other constraints in the tables

3- Insert / populate data into table. each table must have more than 7 rows/records of data.

4- Submit some sample queries of your database (create, insert, select). Keep it simple.

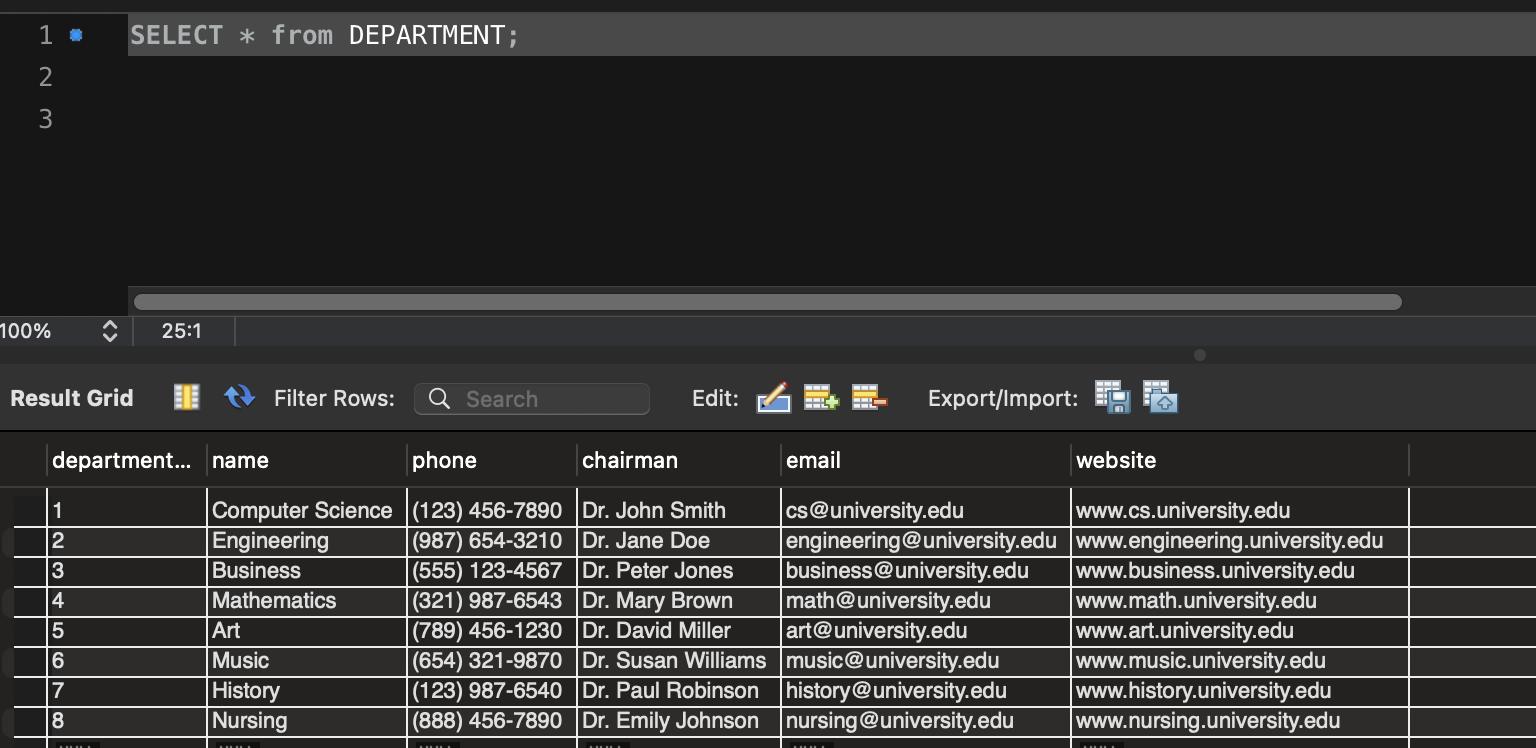
**Tables Proposed**

1. Department
2. Major
3. Scholarship
4. Scholarship\_Sponsor
5. Sponsor
6. Student
7. Student\_Age



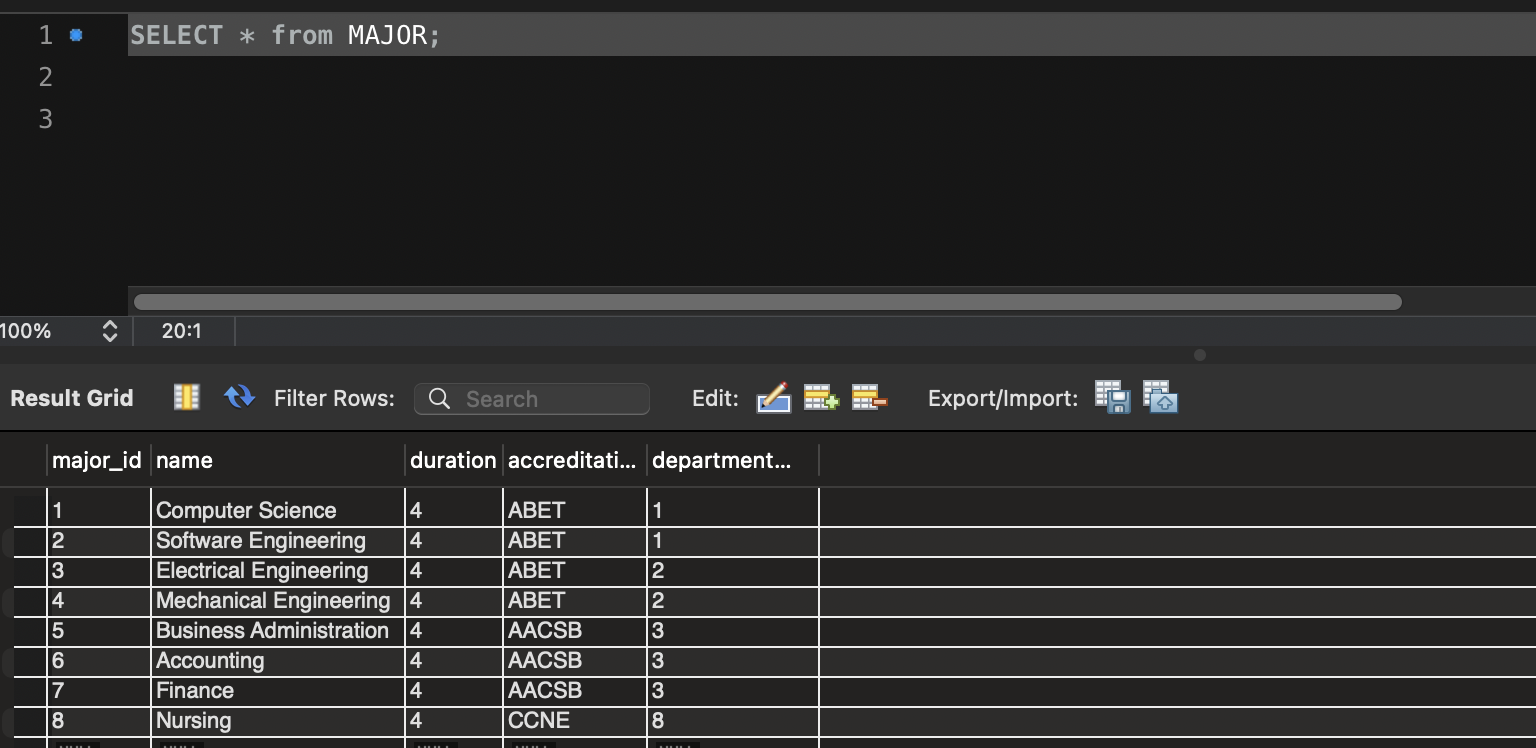
**Primary and Foreign Keys**

Department Table - department\_id INT PRIMARY KEY,



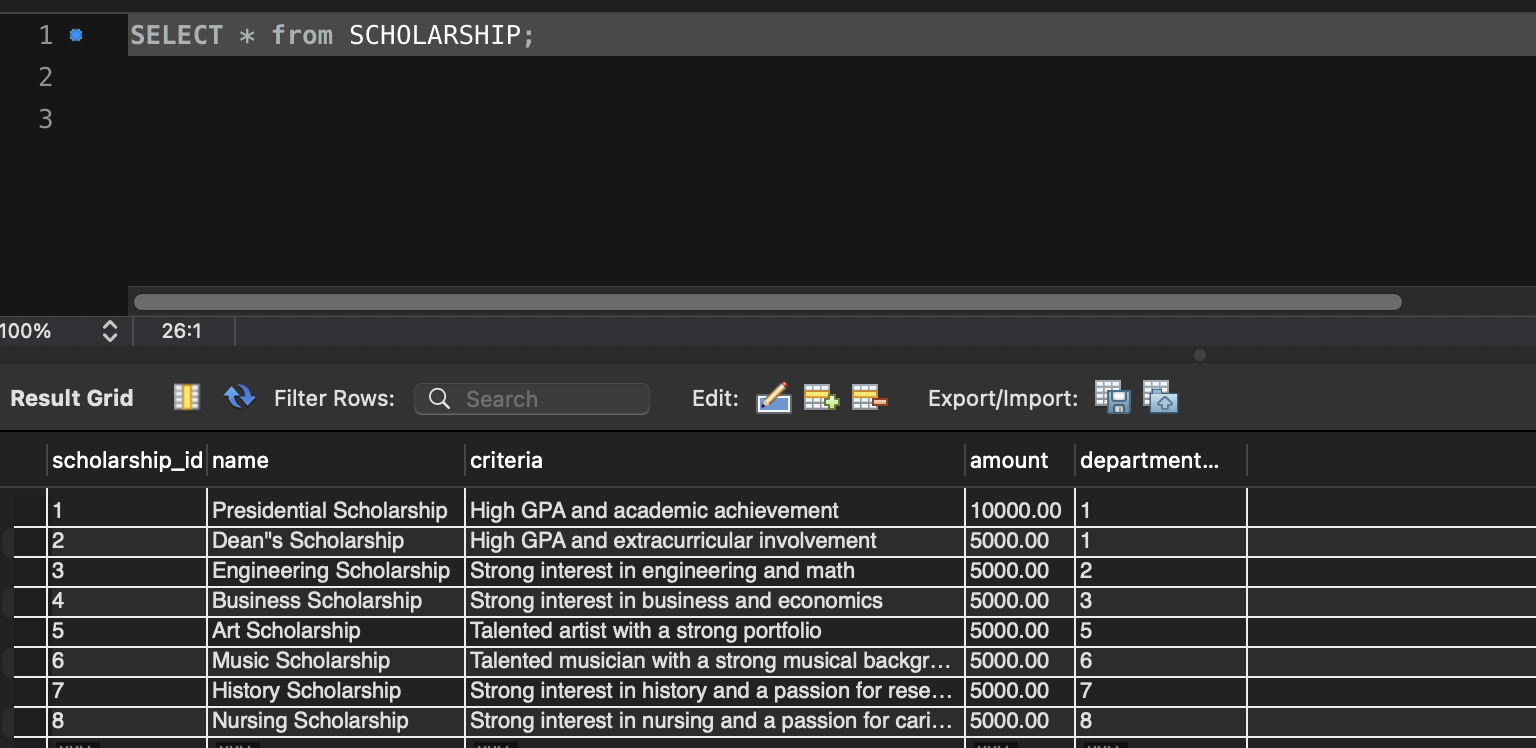
Major Table - major\_id INT PRIMARY KEY,

FOREIGN KEY (department\_id) REFERENCES DEPARTMENT(department\_id)



Scholarship - scholarship\_id INT PRIMARY KEY,

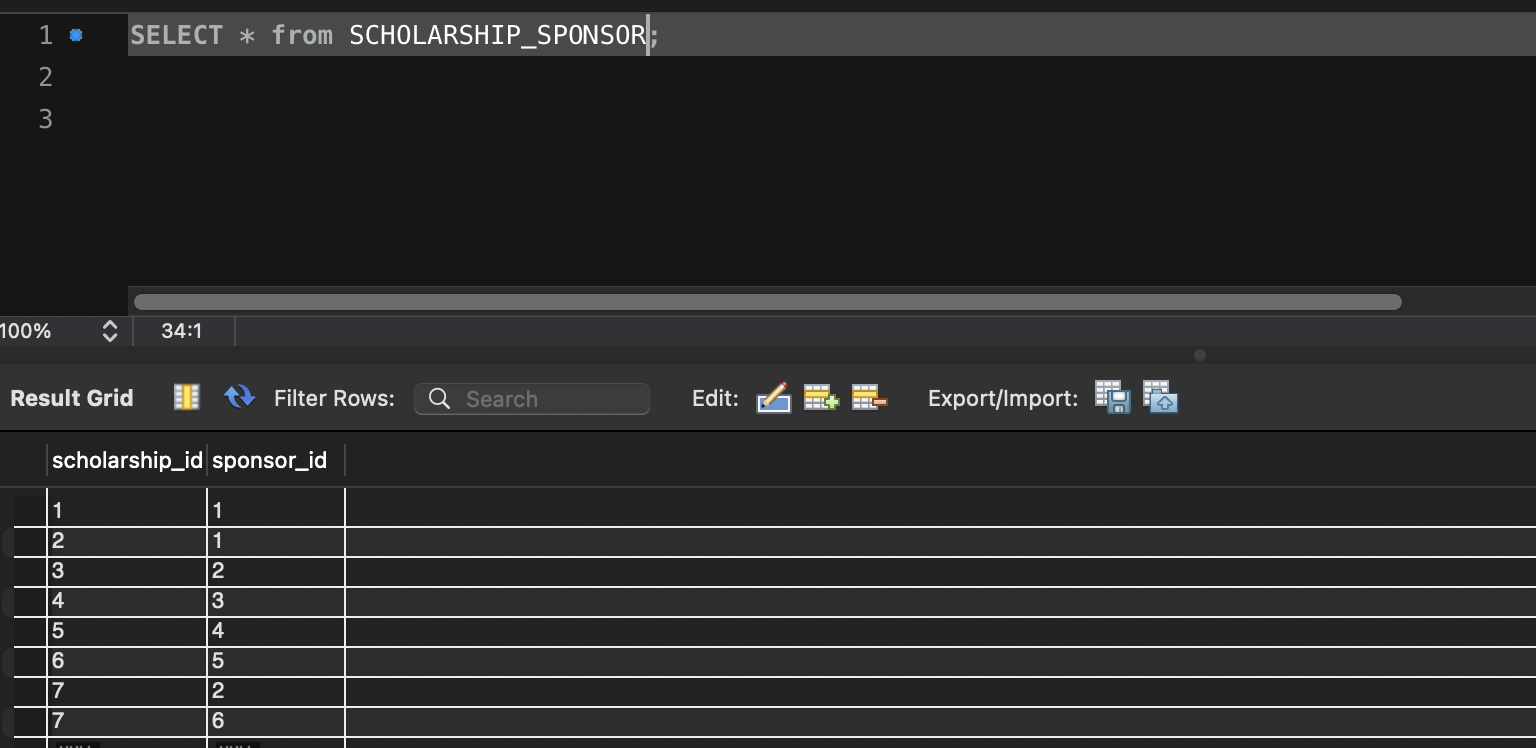
FOREIGN KEY (department\_id) REFERENCES DEPARTMENT(department\_id)



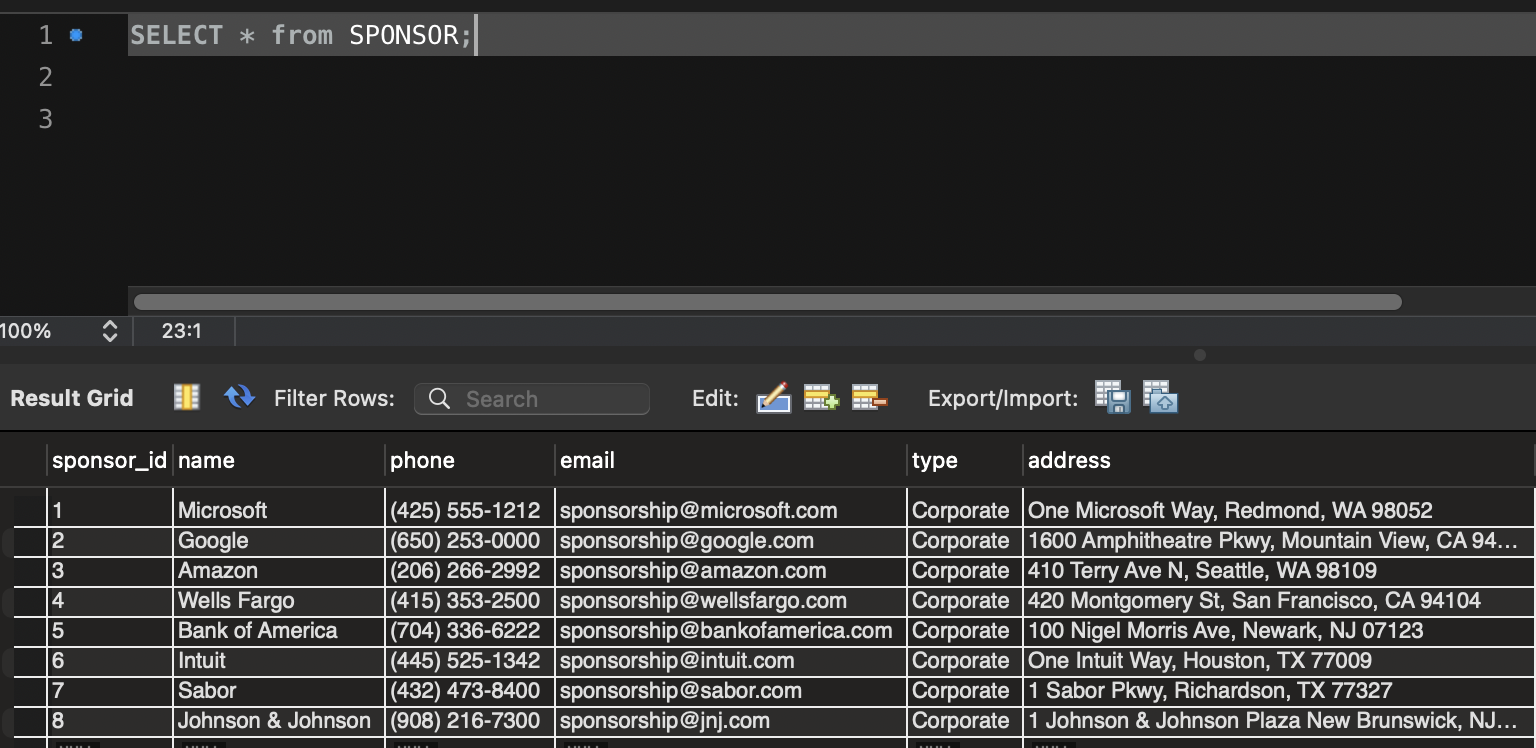
Scholarship\_Sponsor - PRIMARY KEY (scholarship\_id, sponsor\_id),

FOREIGN KEY (scholarship\_id) REFERENCES SCHOLARSHIP(scholarship\_id),

FOREIGN KEY (sponsor\_id) REFERENCES SPONSOR(sponsor\_id)



Sponsor - sponsor\_id INT PRIMARY KEY

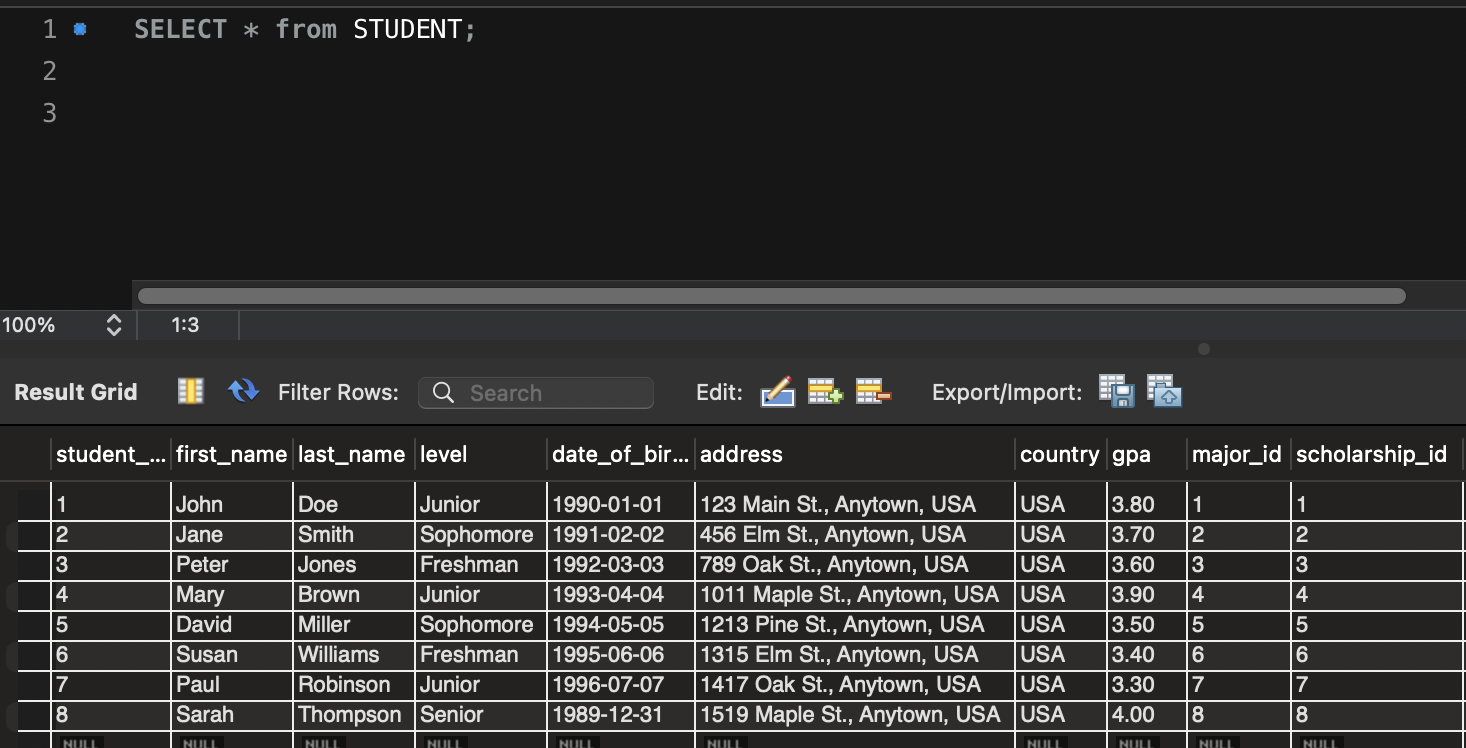


Student - student\_id INT PRIMARY KEY,

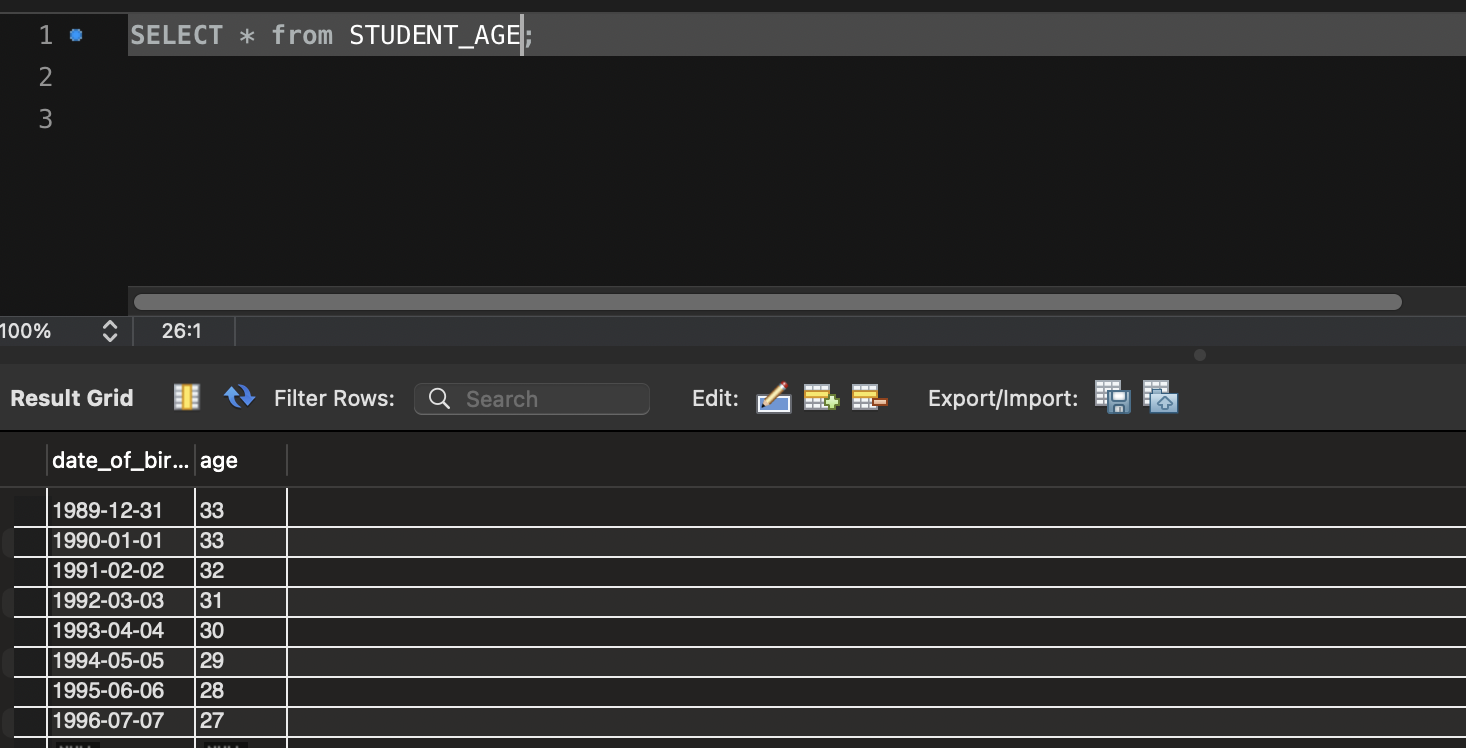
FOREIGN KEY (date\_of\_birth) REFERENCES STUDENT\_AGE(date\_of\_birth),

FOREIGN KEY (major\_id) REFERENCES MAJOR(major\_id),

FOREIGN KEY (scholarship\_id) REFERENCES SCHOLARSHIP(scholarship\_id)



Student\_Age - PRIMARY KEY (date\_of\_birth)



Sample Queries

1) Find the name, department, and scholarship for all students with a GPA of 3.8 or higher.

Query - SELECT s.first\_name, s.last\_name, d.name AS department\_name, sc.name AS scholarship\_name

FROM STUDENT s

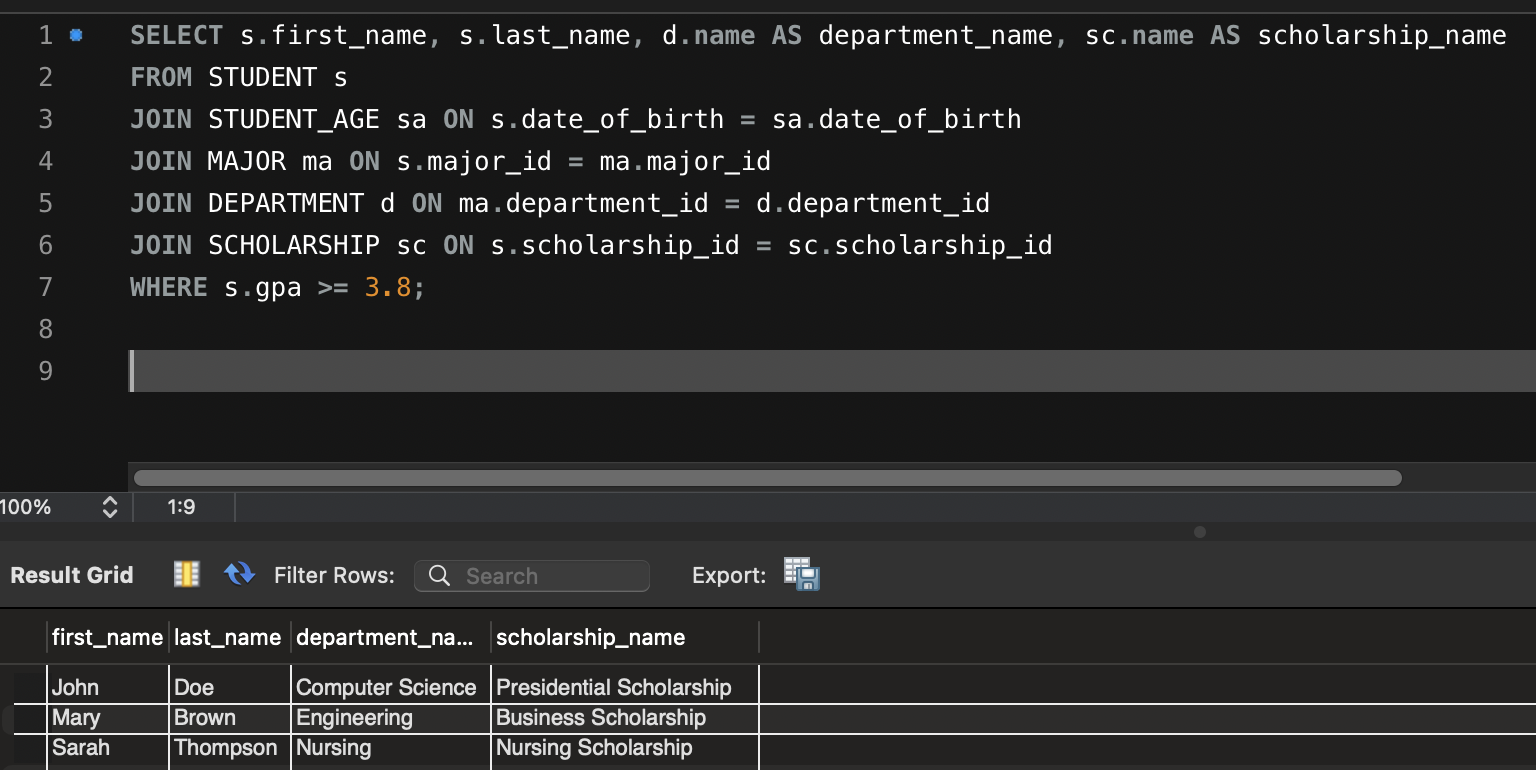
JOIN STUDENT\_AGE sa ON s.date\_of\_birth = sa.date\_of\_birth

JOIN MAJOR ma ON s.major\_id = ma.major\_id

JOIN DEPARTMENT d ON ma.department\_id = d.department\_id

JOIN SCHOLARSHIP sc ON s.scholarship\_id = sc.scholarship\_id

WHERE s.gpa >= 3.8;



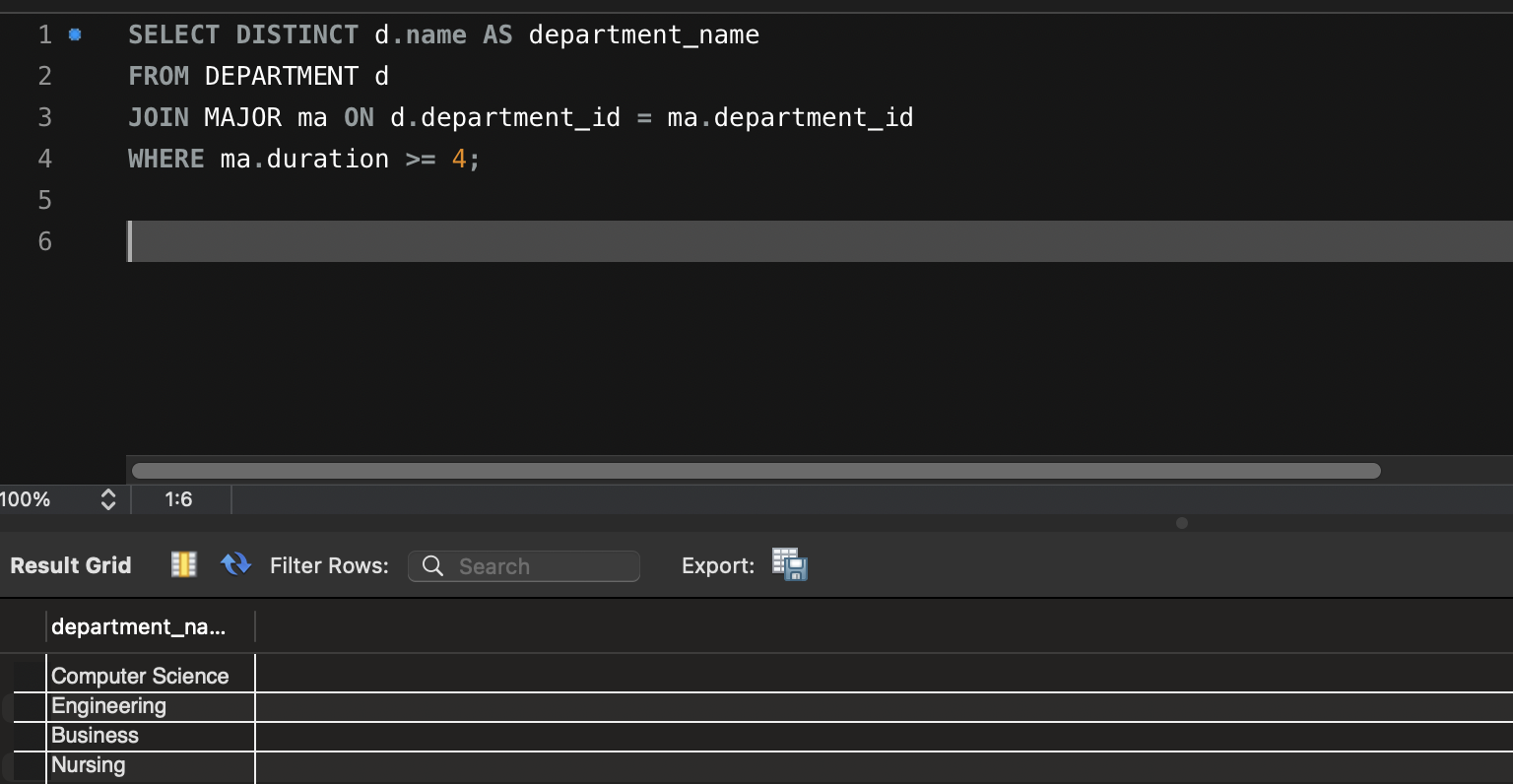
2) List the names of all departments that offer majors with a duration of 4 or more years.

Query - SELECT DISTINCT d.name AS department\_name

FROM DEPARTMENT d

JOIN MAJOR ma ON d.department\_id = ma.department\_id

WHERE ma.duration >= 4;



3) Find the total amount of scholarship money awarded to students in the Computer Science department.

Query - SELECT SUM(sc.amount) AS total\_amount\_awarded

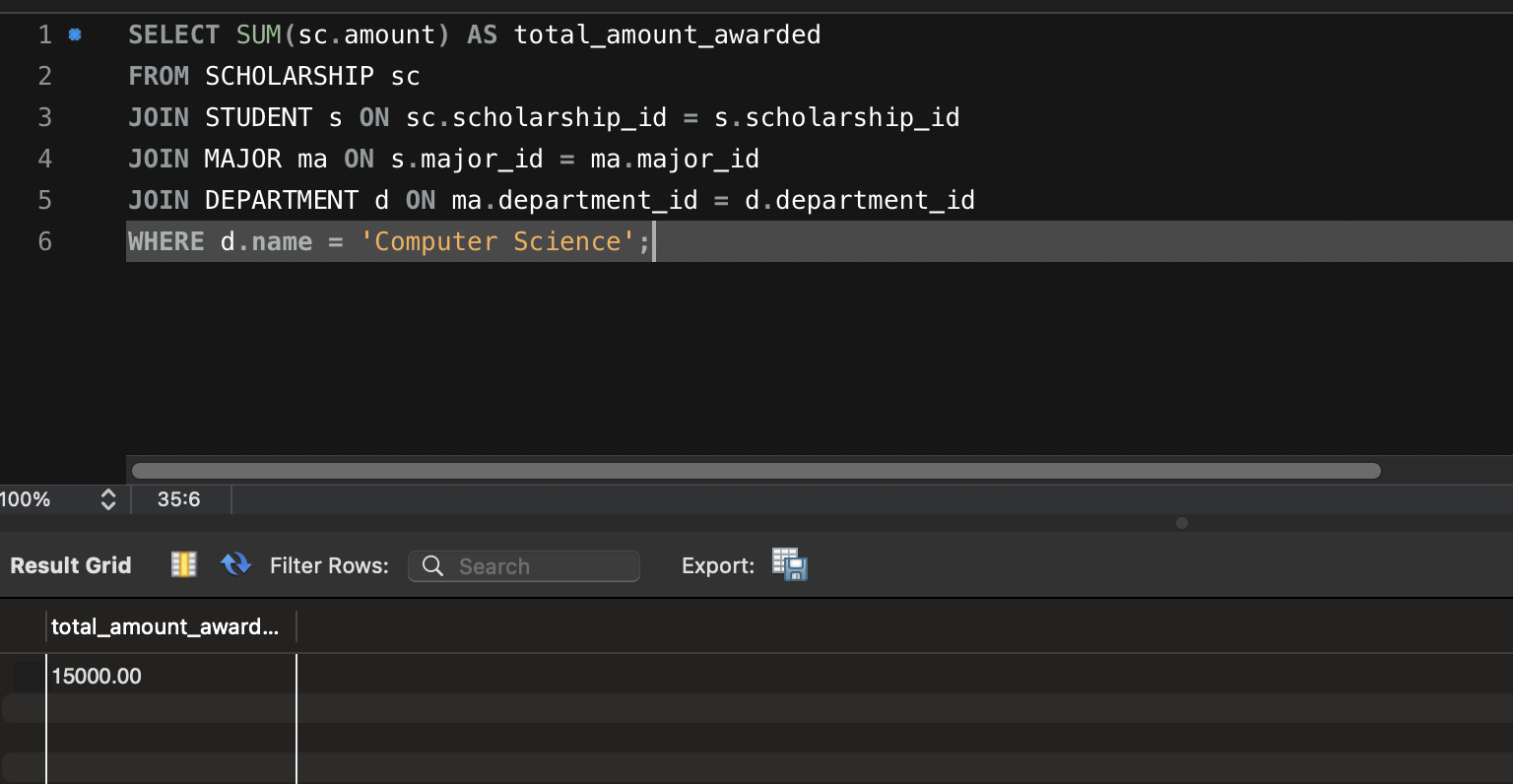
FROM SCHOLARSHIP sc

JOIN STUDENT s ON sc.scholarship\_id = s.scholarship\_id

JOIN MAJOR ma ON s.major\_id = ma.major\_id

JOIN DEPARTMENT d ON ma.department\_id = d.department\_id

WHERE d.name = 'Computer Science';



4) Identify the most common scholarship among students in the Engineering department.

Query - SELECT sc.name AS most\_common\_scholarship, COUNT(\*) AS number\_of\_students

FROM SCHOLARSHIP sc

JOIN STUDENT s ON sc.scholarship\_id = s.scholarship\_id

JOIN MAJOR ma ON s.major\_id = ma.major\_id

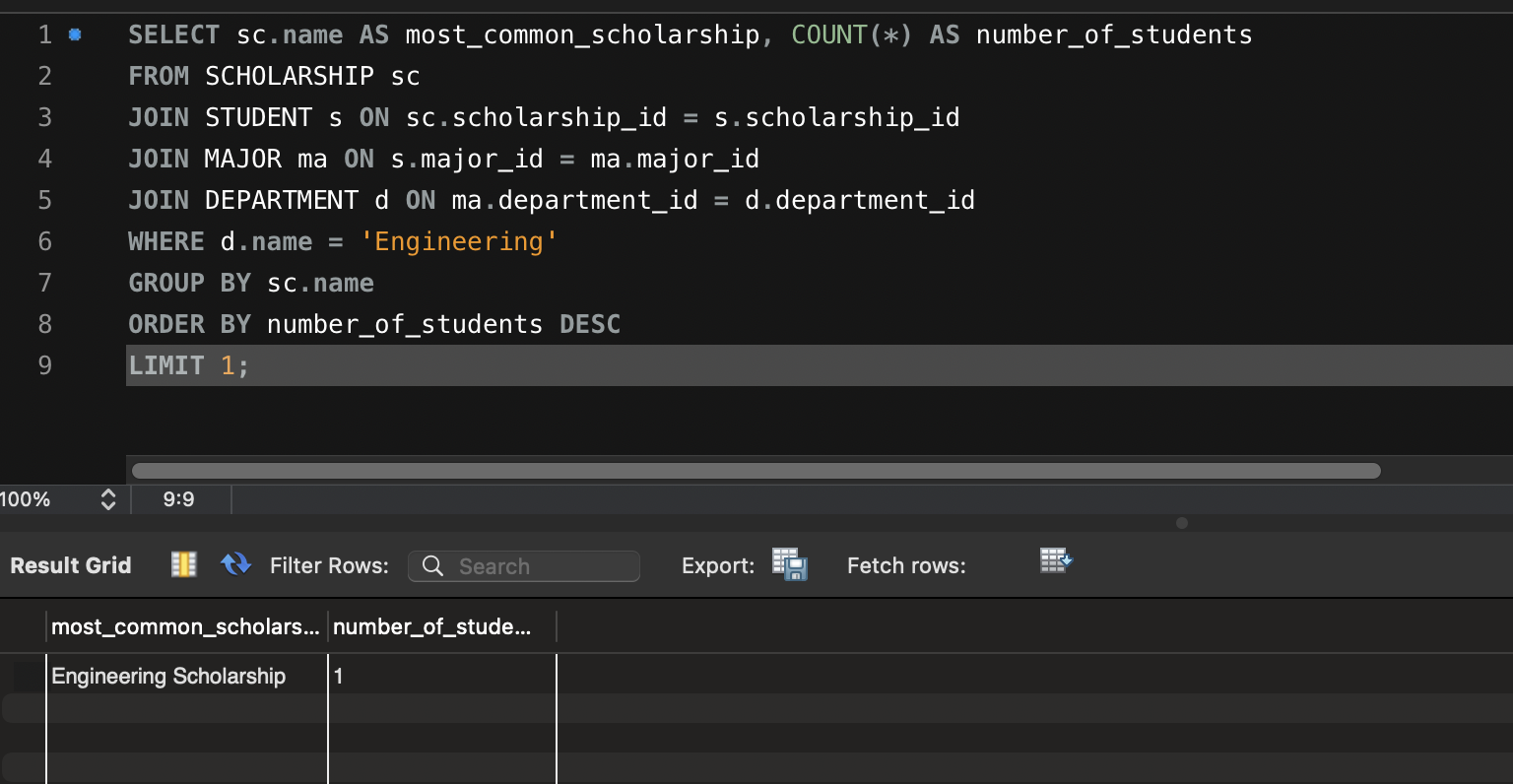
JOIN DEPARTMENT d ON ma.department\_id = d.department\_id

WHERE d.name = 'Engineering'

GROUP BY sc.name

ORDER BY number\_of\_students DESC

LIMIT 1;



5) Display the names of all sponsors that support scholarships for students in the Business department.

Query - SELECT DISTINCT sp.name AS sponsor\_name

FROM SPONSOR sp

JOIN SCHOLARSHIP\_SPONSOR scsp ON sp.sponsor\_id = scsp.sponsor\_id

JOIN SCHOLARSHIP sc ON scsp.scholarship\_id = sc.scholarship\_id

JOIN STUDENT s ON sc.scholarship\_id = s.scholarship\_id

JOIN MAJOR ma ON s.major\_id = ma.major\_id

JOIN DEPARTMENT d ON ma.department\_id = d.department\_id

WHERE d.name = 'Business';