

SQL ASSIGNMENT

```
1. SELECT pname
FROM prof
WHERE dname IN (
    SELECT dname
    FROM dept
    WHERE numphds < 50
);
```

```
2. SELECT sname, gpa
FROM student
WHERE gpa = (
    SELECT MIN(gpa)
    FROM student
);
```

```
3. SELECT sc.cno, sc.sectno, AVG(s.gpa) AS average_gpa
FROM section sc
JOIN "Enroll" e ON sc.cno = e.cno AND sc.sectno = e.sectno
JOIN dept d ON e.dname = d.dname
JOIN student s ON e.sid = s.sid
WHERE d.dname = 'Computer Sciences'
```

GROUP BY sc.cno, sc.sectno, sc.dname;

4. SELECT s.sectno, s.dname

FROM section s

JOIN "Enroll" e ON s.sectno = e.sectno AND s.dname = e.dname

GROUP BY s.sectno, s.dname

HAVING COUNT (e.sid) > 6

5. SELECT sname, s.sid

FROM student s

JOIN "Enroll" e ON s.sid = e.sid

GROUP BY s.sid, s.sname

HAVING COUNT (e.sid) = (

SELECT COUNT(sid)

FROM "Enroll"

GROUP BY sid

ORDER BY COUNT (sid) DESC

LIMIT 1

);

6. SELECT DISTINCT d.dname

FROM dept d

JOIN major m ON d.dname = m.dname

JOIN student s ON s.sid = m.sid

WHERE s.age > 18;

7. SELECT sname, m.dname

FROM student s

JOIN "Enroll" e ON s.sid = e.sid

JOIN course c ON e.dname = c.dname

JOIN major m ON s.sid = m.sid

WHERE c.cname LIKE '%College Geometry%';

8.

9. SELECT s.sname, s.sid

FROM student s

JOIN enroll e ON s.sid = e.sid

JOIN course c ON e.dname = c.dname AND e.cno = c.cno

JOIN dept d ON c.dname = d.dname

WHERE d.dname IN ('Computer Sciences', 'Mathematics')

```
GROUP BY s.sid, s.sname  
  
HAVING COUNT(DISTINCT d.dname) = 2;
```

```
10. SELECT MAX(s.age) - MIN(s.age) AS age_difference  
  
FROM student s  
  
JOIN major m ON s.sid = m.sid  
  
JOIN dept d ON m.dname = d.dname  
  
WHERE d.dname = 'Computer Sciences';
```

```
11. SELECT d.dname AS department_name, AVG(s.gpa) AS average_gpa  
  
FROM dept d  
  
JOIN major m ON d.dname = m.dname  
  
JOIN student s ON m.sid = s.sid  
  
GROUP BY d.dname  
  
HAVING MIN(s.gpa) < 1.0;
```

```
/*1.Print the names of professors who work in departments that have fewer than 50  
PhD students.*/
```

```
SELECT p.pname  
FROM prof p, dept d  
WHERE p.dname = d.dname  
      AND d.numphds < 50;
```

```
/*2. Print the names of the students with the lowest GPA.*/
```

```
SELECT s.sname, gpa  
FROM student s
```

```
JOIN (  
    SELECT MIN(gpa) AS min_gpa  
    FROM student  
) m ON s.gpa = m.min_gpa;
```

/*3.For each Computer Sciences class, print the class number, section number, and the average gpa of the students enrolled in the class section.*/

```
SELECT e.cno, e.sectno, AVG(s.gpa) AS avg_gpa  
FROM enroll e, student s  
WHERE e.sid = s.sid  
    AND e.dname = 'Computer Sciences'  
GROUP BY e.cno, e.sectno;
```

/*4.Print the names and section numbers of all sections with more than six students enrolled in them.*/

```
SELECT s.sectno, s.dname  
FROM section s  
JOIN enroll e ON s.sectno = e.sectno AND s.dname = e.dname  
GROUP BY s.sectno, s.dname  
HAVING COUNT (e.sid) > 6;
```

/*5.Print the name(s) and sid(s) of the student(s) enrolled in the most sections.*/

```
SELECT s.sname, s.sid  
FROM student s  
WHERE s.sid IN (  
    SELECT e.sid  
    FROM enroll e  
    GROUP BY e.sid  
    HAVING COUNT(*) = (  
        SELECT MAX(section_count)  
        FROM (  

```

```

        SELECT COUNT(*) AS section_count
        FROM enroll
        GROUP BY sid
    ) AS counts
)
);

```

/*6.Print the names of departments that have one or more majors who are under 18 years old.*/

```

SELECT dname
FROM dept
WHERE EXISTS (
    SELECT 1
    FROM major, student
    WHERE major.dname = dept.dname
        AND major.sid = student.sid
        AND student.age < 18
);

```

/*7.Print the names and majors of students who are taking one of the College Geometry courses.*/

```

SELECT sname, m.dname
FROM student s
JOIN enroll e ON s.sid = e.sid
JOIN course c ON e.dname = c.dname
JOIN major m ON s.sid = m.sid
WHERE c.cname LIKE '%College Geometry%';

```

/*8.For those departments that have no major taking a College Geometry course print the department name and the number of PhD students in the department.*/

```

SELECT d.dname, d.numphds
FROM dept d
LEFT JOIN major m ON d.dname = m.dname
LEFT JOIN enroll e ON m.sid = e.sid

```

```
LEFT JOIN course c ON e.dname = c.dname AND e.cno = c.cno
WHERE c.cname NOT LIKE '%College Geometry%'
GROUP BY d.dname, d.numphds;
```

/*9. Print the names of students who are taking both a Computer Sciences course and a Mathematics course.*/

```
SELECT s.sname
FROM student s
WHERE s.sid IN (
    SELECT e.sid
    FROM enroll e
    WHERE e.cno IN (
        SELECT c.cno
        FROM course c
        WHERE c.dname = 'Computer Sciences'
    )
)
AND s.sid IN (
    SELECT e.sid
    FROM enroll e
    WHERE e.cno IN (
        SELECT c.cno
        FROM course c
        WHERE c.dname = 'Mathematics'
    )
);
```

/*10. Print the age difference between the oldest and the youngest Computer Sciences major.*/

```
SELECT MAX(s.age) - MIN(s.age) AS age_DIFFERENCE
FROM student s
WHERE s.sid IN (
    SELECT m.sid
    FROM major m, dept d
    WHERE m.dname = d.dname
```

```

    AND d.dname = 'Computer Sciences'
);

/* 11.For each department that has one or more majors with a GPA under 1.0, print
the name of the department and the average GPA of its majors.*/

SELECT d.dname AS department_name, AVG(s.gpa) AS average_gpa
FROM dept d
JOIN major m ON d.dname = m.dname
JOIN student s ON m.sid = s.sid
GROUP BY d.dname
HAVING MIN(s.gpa) < 1.0;

/* 12.Print the ids, names and GPAs of the students who are currently taking all
the Civil Engineering courses.*/

SELECT s.sid, s.sname, s.gpa
FROM student s
WHERE NOT EXISTS (
    SELECT c.cno
    FROM course c
    WHERE c.dname = 'Civil Engineering'
    AND NOT EXISTS (
        SELECT 1
        FROM enroll e
        WHERE e.sid = s.sid
        AND e.cno = c.cno
    )
);

```