

Basic Joins

1. List All Students and Their Departments

- Query the student table to list all students with their corresponding department names.

2. Find Courses and Their Instructors

- Retrieve all courses and the names of the instructors teaching them.

3. Get All Sections for a Given Course

- Find all sections for the course with course code 'CS-101' and show the term, year, building, and room number.

4. List All Instructors and Their Teaching Sections

- Display each instructor's ID, name, and the course section they are teaching.

5. Find Students Enrolled in a Specific Course

- List all students enrolled in the course 'BIO-101', including their IDs and names.

Intermediate Joins

6. Find Students and Their Advisors

- Retrieve all students with their advisor's ID and name.

```
SELECT student.ID AS student_id, student.name AS student_name, advisor.i_ID AS advisor_id,  
instructor.name AS advisor_name
```

```
FROM student
```

```
JOIN advisor ON student.ID = advisor.s_ID
```

```
JOIN instructor ON advisor.i_ID = instructor.ID;
```

7. Get Courses with Prerequisites

- List all courses that have prerequisites, including the course code, course name, and the prerequisite course code.

```
SELECT course.course_id, course.title, prereq.prereq_id
```

```
FROM course
```

```
JOIN prereq ON course.course_id = prereq.course_id;
```

8. List Departments and Their Courses

- Show each department with the courses it offers, including department name and course title.

```
SELECT department.dept_name, course.title
```

```
FROM department
```

```
JOIN course ON department.dept_name = course.dept_name;
```

9. Find Courses Taught by a Specific Instructor

- Find all courses taught by the instructor with ID '83821', including course codes and titles.

```
SELECT course.course_id, course.title
FROM teaches
JOIN course ON teaches.course_id = course.course_id
WHERE teaches.ID = '83821';
```

10. Show Students with Their Grades in a Specific Course

- List students and their grades for the course 'CS-319'.

```
SELECT student.ID, student.name, takes.grade
FROM takes
JOIN student ON takes.ID = student.ID
WHERE takes.course_id = 'CS-319';
```

Advanced Joins

11. List Students Who Took All Courses in a Department

- Find students who have taken all courses offered by the Comp. Sci. department.

```
SELECT student.ID, student.name
FROM student
JOIN takes USING(ID)
JOIN course ON takes.course_id = course.course_id
WHERE course.dept_name = 'Comp. Sci.'
GROUP BY student.ID, student.name
HAVING COUNT(DISTINCT course.course_id) = (
    SELECT COUNT(DISTINCT course_id)
    FROM course
    WHERE dept_name = 'Comp. Sci.'
);
```

12. Find Instructors Teaching Multiple Courses

List instructors who teach more than one course, including their IDs and the count of courses they teach.

```
SELECT teaches.ID, COUNT(DISTINCT teaches.course_id) AS course_count
FROM teaches
GROUP BY teaches.ID
HAVING COUNT(DISTINCT teaches.course_id) > 1;
```

13. Find Students with Advisors in a Specific Department

- List students whose advisors are from the Comp. Sci. department, including student IDs and advisor names.

-

```
SELECT student.ID, student.name, instructor.name AS advisor_name
FROM student
JOIN advisor ON student.ID = advisor.s_ID
JOIN instructor ON advisor.i_ID = instructor.ID
WHERE instructor.dept_name = 'Comp. Sci.';
```

14. List Departments with No Courses Offered

- Find departments that do not offer any courses, including department names.

```
SELECT department.dept_name
FROM department
LEFT JOIN course ON department.dept_name = course.dept_name
WHERE course.course_id IS NULL;
```

15. Find the Average Grade of Students in Each Course

- Calculate the average grade for each course, including course codes, titles, and average grades.

```
SELECT takes.course_id, course.title, AVG(CAST(takes.grade AS DECIMAL)) AS  
average_grade
```

```
FROM takes
```

```
JOIN course ON takes.course_id = course.course_id
```

```
GROUP BY takes.course_id, course.title;
```

```
SELECT
```

```
    takes.course_id, course.title , AVG(
```

```
        CASE takes.grade
```

```
        WHEN 'A' THEN 4.0
```

```
        WHEN 'A-' THEN 3.7
```

```
        WHEN 'B+' THEN 3.3
```

```
        WHEN 'B' THEN 3.0
```

```
        WHEN 'B-' THEN 2.7
```

```
        WHEN 'C+' THEN 2.3
```

```
        WHEN 'C' THEN 2.0
```

```
        WHEN 'C-' THEN 1.7
```

```
        WHEN 'D+' THEN 1.3
```

```
        WHEN 'D' THEN 1.0
```

```
        WHEN 'F' THEN 0.0
```

```
        ELSE NULL
```

```
    END ) AS average_grade
```

```
FROM takes
```

```
JOIN course ON takes.course_id = course.course_id
```

```
GROUP BY takes.course_id, course.title;
```

Expert Joins

16. Find the Most Popular Time Slot for Each Course

- Determine which time slot (e.g., 'A', 'B', 'C', etc.) is most frequently assigned to sections of each course.

```
SELECT section.course_id, section.time_slot_id, COUNT(*) AS frequency
FROM section
GROUP BY section.course_id, section.time_slot_id
ORDER BY section.course_id, frequency DESC;
```

17. List Students with the Highest Number of Credits Taken in Each Department

- Find the student who has taken the highest number of credits in each department.

```
SELECT student.dept_name, student.ID, student.name, student.tot_cred AS max_credits
FROM student
WHERE student.tot_cred = (
    SELECT MAX(s.tot_cred)
    FROM student AS s
    WHERE s.dept_name = student.dept_name
);
```

```
SELECT student.dept_name, student.ID, student.name, student.tot_cred AS max_credits
FROM student
WHERE student.tot_cred = (
    SELECT dept_name, MAX(s.tot_cred)
    FROM student AS s
    GROUP BY dept_name
    HAVING student.dept_name=s.dept_name;
);
```

18. Find Courses That Were Not Taught in a Specific Semester

- List courses that were not taught in the Spring 2018 semester.

19. Find the Departments with the Most Number of Courses Taught

- Count the number of courses taught by each department and list them in descending order.