1. Retrieve all students who enrolled in a specific course.

SQL

SELECT s.first\_name, s.last\_name

FROM Students s -- Select the first and last names from the Students table.

JOIN Enrollments e ON s.student\_id = e.student\_id -- Join the Students table with the Enrollments table using the student\_id column. This connects students to their enrollment records.

WHERE e.course\_code = 'CS101'; -- Filter the results to include only enrollments for the course with the code 'CS101'. Replace 'CS101' with the desired course code.

**Explanation:**

SELECT s.first\_name, s.last\_name: This specifies that we want to retrieve the first name and last name of the students.

FROM Students s: This indicates that we are selecting from the "Students" table, and we are using the alias "s" for brevity.

JOIN Enrollments e ON s.student\_id = e.student\_id: This is the core of the query. It joins the "Students" table with the "Enrollments" table based on the student\_id. The student\_id is the primary key in the "Students" table and a foreign key in the "Enrollments" table. This join allows us to connect students with their enrollment records. We use the alias "e" for "Enrollments".

WHERE e.course\_code = 'CS101': This clause filters the joined results. It only includes rows where the course\_code in the "Enrollments" table is equal to 'CS101'. This ensures we only get students enrolled in that specific course.

In essence, this query combines information from the "Students" and "Enrollments" tables to find the names of students associated with a particular course.

2. Find all faculty members in a particular department.

SQL

SELECT f.first\_name, f.last\_name

FROM Faculty f -- Select the first and last names from the Faculty table.

JOIN Departments d ON f.department\_id = d.department\_id -- Join the Faculty table with the Departments table using the department\_id column.

WHERE d.department\_name = 'Computer Science'; -- Filter the results to include only faculty members in the 'Computer Science' department. Replace 'Computer Science' with the desired department name.

**Explanation:**

SELECT f.first\_name, f.last\_name: This selects the first name and the last name of faculty members.

FROM Faculty f: This specifies that we are selecting from the "Faculty" table, aliased as "f".

JOIN Departments d ON f.department\_id = d.department\_id: This joins the "Faculty" table with the "Departments" table using the department\_id column. This is how we link faculty to their respective departments. We use the alias "d" for "Departments".

WHERE d.department\_name = 'Computer Science': This filters the results to include only faculty members where the department\_name in the "Departments" table is 'Computer Science'. You would replace 'Computer Science' with the name of whatever department you're interested in.

This query retrieves faculty names by linking the "Faculty" and "Departments" tables and then filtering by the department name.

3. List all courses a particular student is enrolled in.

SQL

SELECT c.course\_name

FROM Courses c -- Select the course name from the Courses table.

JOIN Enrollments e ON c.course\_code = e.course\_code -- Join the Courses table with the Enrollments table using the course\_code.

JOIN Students s ON e.student\_id = s.student\_id -- Join the Enrollments table with the Students table using the student\_id.

WHERE s.first\_name = 'Alice'; -- Filter the results to include only enrollments for the student named 'Alice'. Replace 'Alice' with the desired student's first name.

**Explanation:**

SELECT c.course\_name: This selects the course\_name from the "Courses" table.

FROM Courses c: This specifies that we are selecting from the "Courses" table, aliased as "c".

JOIN Enrollments e ON c.course\_code = e.course\_code: This joins "Courses" with "Enrollments" using course\_code to link courses to enrollment records. We use the alias "e" for "Enrollments".

JOIN Students s ON e.student\_id = s.student\_id: This joins "Enrollments" with "Students" using student\_id to link enrollments to students. We use the alias "s" for "Students".

WHERE s.first\_name = 'Alice': This filters the results to show only the courses for the student whose first name is 'Alice'. Replace 'Alice' with the name of the student you want.

This query retrieves the names of courses by connecting the "Courses", "Enrollments", and "Students" tables and filtering based on the student's name.

4. Retrieve students who have not enrolled in any course.

SQL

SELECT s.first\_name, s.last\_name

FROM Students s -- Select the first and last names from the Students table.

WHERE s.student\_id NOT IN (SELECT DISTINCT student\_id FROM Enrollments); -- Use a subquery to find student\_ids in Enrollments and exclude them from the results.

**Explanation:**

SELECT s.first\_name, s.last\_name: This selects the first name and last name of students.

FROM Students s: This specifies that we are selecting from the "Students" table, aliased as "s".

WHERE s.student\_id NOT IN (SELECT DISTINCT student\_id FROM Enrollments): This is the filtering condition.

SELECT DISTINCT student\_id FROM Enrollments: This subquery selects all the unique student\_id values from the "Enrollments" table. This gives us a list of all students who have enrolled in at least one course.

NOT IN: The NOT IN operator then filters the students from the main query. It only includes students whose student\_id is not found in the list generated by the subquery.

This query finds students who are not present in the "Enrollments" table, indicating they haven't enrolled in any courses.

5. Find the average grade of students in a specific course.

SQL

SELECT AVG(CASE -- Calculate the average grade using a CASE statement to convert letter grades to numeric values.

WHEN e.grade = 'A' THEN 4

WHEN e.grade = 'B' THEN 3

WHEN e.grade = 'C' THEN 2

WHEN e.grade = 'D' THEN 1

WHEN e.grade = 'F' THEN 0

ELSE 0 -- Handle other cases or NULL

END) AS average\_grade

FROM Enrollments e -- Select from the Enrollments table.

WHERE e.course\_code = 'CS101'; -- Filter for the specific course ('CS101'). Replace 'CS101' with the desired course code.

**Explanation:**

SELECT AVG(...) AS average\_grade: This calculates the average of the grade values and assigns the alias "average\_grade" to the result.

CASE WHEN e.grade = 'A' THEN 4 WHEN e.grade = 'B' THEN 3 ... ELSE 0 END: This CASE statement converts letter grades to numerical values: A=4, B=3, C=2, D=1, F=0. The ELSE 0 handles cases where the grade might be something other than A-F (or NULL), assigning a 0 in those situations to avoid errors.

FROM Enrollments e: This specifies that we are selecting from the "Enrollments" table, aliased as "e".

WHERE e.course\_code = 'CS101': This filters the enrollments to include only those for the course with the code 'CS101'. Replace 'CS101' with the course code you want to analyze.

This query calculates the average grade for a given course by converting letter grades to numbers and then using the AVG function.