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

|  |
| --- |
| sql |
| CopyEdit |
| SELECT s.student\_id, s.first\_name, s.last\_name, c.course\_name |
| FROM Students s |
| JOIN Enrollments e ON s.student\_id = e.student\_id |
| JOIN Courses c ON e.course\_id = c.course\_id |
| WHERE c.course\_code = 'CS101'; |

**Explanation:**  
This query finds all students who are enrolled in the course **"CS101"**.  
It joins three tables:

* Students (student details),
* Enrollments (which links students to courses),
* Courses (course details).

**2. Find all faculty members in a particular department**

|  |
| --- |
| sql |
| CopyEdit |
| SELECT faculty\_id, faculty\_name |
| FROM Faculty |
| WHERE dept\_id = (SELECT dept\_id FROM Departments WHERE dept\_name = 'Computer Science'); |

**Explanation:**  
It finds all faculty members who belong to the **"Computer Science"** department.  
First, it looks up the dept\_id for **"Computer Science"** from the Departments table.  
Then, it finds faculty members in that department.

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

|  |
| --- |
| sql |
| CopyEdit |
| SELECT c.course\_code, c.course\_name |
| FROM Courses c |
| JOIN Enrollments e ON c.course\_id = e.course\_id |
| JOIN Students s ON e.student\_id = s.student\_id |
| WHERE s.email = 'sokso@example.com'; |

**Explanation:**  
This query finds all courses that the student with the email **"sokso@example.com"** is enrolled in.  
It joins:

* Courses (course details),
* Enrollments (link between students and courses),
* Students (to match by email).

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

|  |
| --- |
| sql |
| CopyEdit |
| SELECT s.student\_id, s.first\_name, s.last\_name |
| FROM Students s |
| LEFT JOIN Enrollments e ON s.student\_id = e.student\_id |
| WHERE e.student\_id IS NULL; |

**Explanation:**  
It finds students who **haven't enrolled in any course**.  
It uses a **LEFT JOIN** (to include all students), then checks where there is **no match** in the Enrollments table.

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

|  |
| --- |
| sql |
| CopyEdit |
| SELECT c.course\_code, c.course\_name, AVG( |
| CASE |
| WHEN grade = 'A' THEN 4.0 |
| WHEN grade = 'B' THEN 3.0 |
| WHEN grade = 'C' THEN 2.0 |
| WHEN grade = 'D' THEN 1.0 |
| ELSE 0 |
| END |
| ) AS average\_gpa |
| FROM Enrollments e |
| JOIN Courses c ON e.course\_id = c.course\_id |
| WHERE c.course\_code = 'CS101' |
| GROUP BY c.course\_code, c.course\_name; |

**Explanation:**  
It calculates the **average GPA** for the course **"CS101"**.

* Grades are converted into GPA values (A = 4.0, B = 3.0, etc.).
* The AVG() function finds the **average GPA** of all students in that course.