Name: Mang Chhunleang

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

```
121     -- 1. Retrieve all students who enrolled in a specific course
122     -- Example: Find all students taking 'CS101' (Intro to Programming)
123 •   SELECT s.student_id, s.name, s.email, e.grade
124     FROM Students s
125     JOIN Enrollments e ON s.student_id = e.student_id
126     JOIN Courses c ON e.course_id = c.course_id
127     WHERE c.course_code = 'CS101';
128
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	email	grade
1	Emily Johnson	emily.johnson@university.edu	A
2	Michael Chen	michael.chen@university.edu	B
10	Robert Johnson	robert.johnson@grad.university.edu	B

Explanation:

- Purpose: Finds all students enrolled in "CS101" and their grades.
- JOINs:
    - Links Students → Enrollments (to get enrollment records).
    - Links Enrollments → Courses (to filter by course code).
- Filter: WHERE c.course_code = 'CS101' narrows results to one course.

2. Find all faculty members in a particular department.

```
129     -- 2. Find all faculty members in a particular department.
130     -- Example: Find all Computer Science faculty
131 •   SELECT f.faculty_id, f.name, f.email, d.name AS department
132     FROM Faculty f
133     JOIN Departments d ON f.department_id = d.department_id
134     WHERE d.name = 'Computer Science';
135
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

faculty_id	name	email	department
1	Dr. Alice Smith	alice.smith@university.edu	Computer Science
2	Prof. John Doe	john.doe@university.edu	Computer Science

Explanation:

- Purpose: Lists all faculty members in the Computer Science department.
- JOIN: Links Faculty → Departments to map faculty to their departments.
- Filter: WHERE d.name = 'Computer Science' isolates one department.

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

```
137     --  3. List all courses a particular student is enrolled in.
138     -- Example: Find all courses for student_id 3 (Sarah Williams)
139     SELECT c.course_code, c.name AS course_name,
140            d.name AS department, e.grade
141     FROM Courses c
142     JOIN Enrollments e ON c.course_id = e.course_id
143     JOIN Departments d ON c.department_id = d.department_id
144     WHERE e.student_id = 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_code	course_name	department	grade
CS202	Database Systems	Computer Science	A

Explanation:

- Purpose: Shows all courses taken by student ID 3 (Sarah Williams).
- JOINs:
  - Courses → Enrollments (to find courses the student enrolled in).
  - Courses → Departments (to show department names).
- Filter: WHERE e.student_id = 3 targets one student.

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

```
147     --  4. Retrieve students who have not enrolled in any course.
148     SELECT s.student_id, s.name, s.email
149     FROM Students s
150     LEFT JOIN Enrollments e ON s.student_id = e.student_id
151     WHERE e.enrollment_id IS NULL;
152
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	email

Explanation:

- Purpose: Identifies students with zero enrollments.
- LEFT JOIN: Includes all students, even those without enrollments.
- Filter: WHERE e.enrollment_id IS NULL finds students with no matching enrollment records.

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

```
154     --  5. Find the average grade of students in a specific course.
155     -- Example: Average grade for 'CS101' (using GPA values)
156     SELECT
157         c.course_code,
158         c.name AS course_name,
159         AVG(CASE
160             WHEN e.grade = 'A' THEN 4.0
161             WHEN e.grade = 'B+' THEN 3.5
162             WHEN e.grade = 'B' THEN 3.0
163             WHEN e.grade = 'C+' THEN 2.5
164             WHEN e.grade = 'C' THEN 2.0
165             WHEN e.grade = 'D+' THEN 1.5
166             WHEN e.grade = 'D' THEN 1.0
167             ELSE 0.0
168         END) AS average_gpa
169     FROM Courses c
170     JOIN Enrollments e ON c.course_id = e.course_id
171     WHERE c.course_code = 'CS101'
172     GROUP BY c.course_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_code	course_name	average_gpa
CS101	Introduction to Programming	3.33333

Explanation:

- Purpose: Computes the average GPA for "CS101".
- CASE: Converts letter grades (A, B+, etc.) to numeric GPA values.
- AVG(): Calculates the average of those numeric values.
- JOIN + Filter: Links courses to enrollments and filters for CS101.
- GROUP BY: Ensures results are per-course.