Query

sql
CopyEdit

```
SELECT p.project_id,
       ROUND(AVG(e.experience_years), 2) AS average_years
FROM Project p
LEFT JOIN Employee e
  ON p.employee_id = e.employee_id
GROUP BY p.project_id;
```

Objective

Return the **average years of experience** of employees assigned to each project.

Explanation Breakdown

1. FROM Project p

- We start with the Project table. It likely contains multiple rows, each mapping a project_id to an employee_id.

2. LEFT JOIN Employee e ON p.employee_id = e.employee_id

- This connects the project to the Employee table to access experience_years for each assigned employee.
- Using a **LEFT JOIN** ensures that even if an employee's data is missing (i.e., employee_id in Project but not in Employee), the project is still included in the result.

3. AVG(e.experience_years)

- Calculates the **average years of experience** for all employees working on the same project.
- If a project has more than one employee assigned (multiple rows in Project table), their experience values are averaged.

4. ROUND(..., 2)

- Rounds the average to **2 decimal places**, as required in reporting or analytics.

5. GROUP BY p.project_id

- Groups results by project_id, so we get **one row per project**, with its corresponding average experience.

Key SQL Concepts & Their Roles

Concept	Description
LEFT JOIN	Ensures all projects are shown, even if employee info is missing.
JOIN Condition	Matches employee_id from Project to Employee to fetch experience.
AVG(column)	Computes the average value for a group (e.g., all employees in a project).
ROUND(value, 2)	Limits result to two decimal places.
GROUP BY	Necessary to calculate the average per project.

New Concepts You Should Remember

◆ AVG() – Average Function

- Works only on numeric columns.
- Ignores NULL values in the aggregation.

◆ Handling Missing Data with LEFT JOIN

- LEFT JOIN is useful when you want all records from the left table (here, Project) even if the right-side table (here, Employee) has no matching row.

Sample Scenario

Let’s say:

- Project 1 has 3 employees with 5, 6, and 7 years.
- Project 2 has 1 employee with 2 years.
- Project 3 has no employee details.

Output:

project_id	average_years
1	6.00
2	2.00
3	NULL or 0.00 (if handled with IFNULL)

Summary (Revision Notes)

- **What was asked**: Find average employee experience per project.
- **Tables involved**: Project (contains assignments), Employee (contains experience).
- **Key operations**: JOIN, AVG(), ROUND(), GROUP BY.
- **Why LEFT JOIN?**: To include projects even if employee experience is missing.