3. Multi-Valued Dependencies (4NF Case - Employee & Skills Table)

Table Data:

EmployeeID | EmployeeName | Skill | Project 201 | John | Java | Alpha
201 | John | Python | Alpha
201 | John | Java | Beta

202 | Sarah | SQL | Gamma

Tasks:

- Identify multi-valued dependencies (MVDs). - Check if this table violates 4NF.
- Normalize it to 4NF.

Hint:

An employee can have multiple skills. An employee can be assigned to multiple projects. Skills and projects are independent of each other.

Step 1: Understanding Multi-Valued Dependencies (MVDs)
A multi-valued dependency (MVD) occurs when one attribute in a table is independent of another but both are dependent on a primary key.

Identifying Functional Dependencies and MVDs

1 **Functional Dependencies (FDs):**

   ○ EmployeeID → EmployeeName (Each EmployeeID determines the EmployeeName)

2 **Multi-Valued Dependencies (MVDs):**

- EmployeeID →→ Skill (An employee can have multiple skills, independent of projects)
- EmployeeID →→ Project (An employee can work on multiple projects, independent of skills)

Since **Skill and Project are independent**, this table has an **MVD**, which can lead to redundancy.

Step 2: Checking for 4NF Violations
The given table is in 4NF because:

It has no non-trivial multi-valued dependencies (i.e., if an MVD exists, one of the attributes must be a primary key).

EmployeeID →→ Skill exists, meaning multiple skills are stored redundantly for each project.

EmployeeID →→ Project exists, meaning multiple projects are stored redundantly for each skill.

This causes **data duplication** and **insertion/update anomalies**.

Step 3: Normalization to 4NF
To remove **multi-valued dependencies**, we must **split the table into separate tables**, ensuring skills and projects are stored independently.

click this link for see -> https://dbdiagram.io/d/practice-question-3-67e6380d4f7afba18487892c