

why when we create employee table we didn't make the manager id and employee id as composite primary key as we did here Student Subjects Table

Stud\_ID, Subject, Subj\_Desc, G

Each row = one subject record for one student.

Composite primary key = (Stud\_ID, Subject). when we find repeating group? what is the rule?

That's a really sharp observation — and the difference comes down to **the nature of the relationship** and **what uniquely identifies a row**.

## 1. General Rule

- You create a **composite primary key** when **no single attribute alone can uniquely identify a row**, but the *combination* of attributes does.
- This usually happens in **many-to-many relationships** or when you're resolving a **repeating group**.

## 2. Student–Subject Example

- A student can take many subjects.
- A subject can be taken by many students.
- So the table `StudentSubjects` needs both `Stud_ID` and `Subject` together to uniquely identify a row.
- That's why the composite key `(Stud_ID, Subject)` is required.

If you only used `Stud_ID`, you'd have duplicates (same student across multiple subjects).

If you only used `Subject`, you'd have duplicates (same subject across multiple students).

The **pair** is what makes each row unique.

## 3. Employee–Manager Example

- Each employee has **one unique identifier**: `employeeID`.
- That alone is enough to uniquely identify a row in the Employee table.
- The managerID is just a **foreign key** pointing to another employee (the supervisor).
- You don't need `(employeeID, managerID)` as a composite key, because `employeeID` by itself already guarantees uniqueness.

If you made `(employeeID, managerID)` the primary key, you'd be saying:

- An employee could appear multiple times with different managers (i.e., many-to-many).
- But in the usual business rule, an employee has only one manager → so no repeating group exists here.

## 4. The Rule of Thumb

- If the entity itself has a natural unique identifier (like `employeeID`, `studentID`, `sectorID`, etc.), use that as the primary key.
- If the table exists only to resolve a many-to-many relationship (like Student–Subject, Employee–Project, etc.), then use a composite primary key.

✓ So:

- **Employee table** → `employeeID` is enough (managerID is just a foreign key).
- **StudentSubjects table** → needs `(Stud_ID, Subject)` as composite PK because it's a many-to-many relationship.