Here's an explanation of **Functional Dependency**, **Partial Dependency**, and **Transitive Dependency** with examples in tabular form:

1. **Functional Dependency (FD)**

- **Definition**: A functional dependency occurs when one attribute (or set of attributes) determines another attribute.
- **Notation**: If attribute $A \rightarrow B$, then A uniquely determines B.

Example: Consider the following table:

StudentID	StudentName	CourseID	CourseName
S001	John	C101	Physics
S002	Alice	C102	Math

In this table:

- `StudentID → StudentName`: The `StudentID` determines a unique `StudentName`. So, for every `StudentID`, there is exactly one `StudentName`.
- `CourseID → CourseName`: Similarly, `CourseID` determines a unique `CourseName`.

2. **Partial Dependency**

- **Definition**: A partial dependency occurs when a non-prime attribute depends on only a part of a composite primary key.
- **Notation**: If we have a composite primary key `(A, B)` and an attribute `C` depends on `A` but not on `B`, it is a partial dependency.

Example: Consider the following table with a composite primary key `(StudentID, CourseID)`:

StudentID	CourseID	StudentName	CourseName
S001	C101	John	Physics
S002	C102	Alice	Math

Here, the composite primary key is `(StudentID, CourseID)`.

- `StudentID → StudentName`: The `StudentID` determines the `StudentName`, but this dependency is based only on `StudentID`, not the whole composite key.
- `CourseID → CourseName`: Similarly, `CourseID` determines `CourseName`.

These are partial dependencies because they depend only on a part of the composite key (`StudentID` or `CourseID`), not the entire key `(StudentID, CourseID)`.

3. **Transitive Dependency**

- **Definition**: A transitive dependency occurs when an attribute depends on another attribute through a third attribute. In other words, if $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$ is a transitive dependency.

Example: Consider the following table:

EmployeeID	EmployeeName	Department	DepartmentHead
E001	Bob	Sales	Tom
E002	Alice	IT	Jerry

Here:

- `EmployeeID → EmployeeName`: `EmployeeID` determines `EmployeeName`.
- `Department → DepartmentHead`: `Department` determines `DepartmentHead`.

Since `EmployeeID → Department` (via some relation between employees and departments), and `Department → DepartmentHead`, we can say `EmployeeID → DepartmentHead` is a **transitive dependency**.

In this case, `EmployeeID` indirectly determines the `DepartmentHead` through the `Department`, which violates the rule of normalization (i.e., 3NF), where there should be no transitive dependencies.

Summary:

Type of Dependency	Example	Explanation
Functional Dependency (FD)	`StudentID → StudentName`	One attribute determines another uniquely.
Partial Dependency	`StudentID → StudentName` (when `StudentID, CourseID` is the composite key)	Non-prime attribute depends on part of the composite key.
Transitive Dependency	`EmployeeID → DepartmentHead` (via `Department`)	One attribute depends on another through a third attribute.

These dependencies help in ensuring that the database is normalized, reducing redundancy and maintaining data integrity.