**DIFFERENTIATE BETWEEN PARTIAL DEPENDENCY AND TRANSITIVE DEPENDENCY**

## **Partial Dependency**

A partial dependency occurs when a non-prime attribute (an attribute that is not part of any candidate key) is functionally dependent on part of a composite primary key rather than on the whole key. The 2NF eliminates partial dependency.

Example:

Employee\_task Table

| **Employee\_ID** | **Task\_No** | **Employee\_name** | **Task\_Name** |
| --- | --- | --- | --- |
| C01 | 58 | Mona | App development |
| C02 | 34 | Geninne | UI/UX Designing |

Here, the prime key attributes are Employee\_ID and Task\_No, and also:

Employee\_ID = A unique ID of the employee

Employee\_Name = Name of the employee

Task\_No = A unique ID of the task

Task\_Name = The name of the task

As stated above, the non-prime attributes are Employee\_Name and Task\_Name. These must be dependent functionally on the part of the candidate key so as to be Partial Dependent.

The Employee\_Name can be determined using the Employee\_ID. It actually makes the relation Dependent Partially.

The Task\_Name can be determined using the Task\_No. It makes the relation Dependent Partially.

Thus, the Employee\_Task relation would violate the 2NF(Second Normal Form) in Normalization and is considered to be a bad database design.

We have to decompose the table to remove dependencies.

(Employee Info)

| **Employee\_ID** | **Task\_No** | **Employee\_Name** |
| --- | --- | --- |
| C01 | 58 | Mona |
| C02 | 34 | Geninne |

(Task Info)

| **Task\_No** | **Task\_Name** |
| --- | --- |
| 58 | App Development |
| 34 | UI/UX Design |

In this case we have successfully taken it to a second Normal form.

**Transitive Dependency**

A transitive dependency occurs when a non-prime attribute depends on another non-prime attribute rather than directly on the primary key. In other words, if attribute A determines attribute B, and attribute B determines attribute C, then C is transitively dependent on A.

Examples:

In a table Employees with a primary key *EmployeeID*, if *EmployeeID* determines *DepartmentID*, and *DepartmentID* determines *DepartmentName*, then *DepartmentName* is transitively dependent on *EmployeeID*..

Transitive dependencies can also cause redundancy and update anomalies. To eliminate transitive dependencies, the table should be further decomposed, typically leading to the design that adheres to the Third Normal Form (3NF).