Normalization work Document

A close-up of a document

AI-generated content may be incorrect.

A source of data for the database

0NF : is the collection of data from your data source

It is best to look at the labels on your resource document

Employee (**EmployeeID**, EmployeeName, DepartmentNumber, DepartmentName, (ProjectNumber, ProjectName, DepartmentName,WeeklyHours))

Optionally at this point you could name your table. You MAY discover that you might need to rename tables already given a name. This is why this step is optionally and naming of table can wait until the 3NF form is done.

1NF: for an entity to be in this normal form:

1. A table MUST contain atomic attributes. (optionally you **could** atomize in 0NF)
2. A table cannot contain any repeating groups of attributes.
   1. Cut the repeating group from the original entity table AND  
      make it its own table
   2. Copy down the primary key in the original entity table to the new entity table **AND** indicate the attribute as a foreign key
   3. Decide on a primary attribute in your new table that is the unique identifier for the table (primary key for the new table)
3. Optionally at this point you could name your table. You MAY discover that you might need to rename tables already given a name. This is why this step is optionally and naming of table can wait until the 3NF form is done.

Employee (**EmployeeID**, EmployeeFirstName, EmployeeLastName DepartmentNumber, DepartmentName)

Project (**ProjectNumber**, ***EmployeeID***, ProjectName, DepartmentName, WeeklyHours)

With BOTH the projectnumber and employeeid, can I uniquely identify a row in the project table?   
 YES

The result of using two OR more attributes to identify the unique row of the table is called a **composite (compound) key**

**2NF Normal Form : Removing Partial Dependencies**

What is a partial dependency?

A partial dependency exists **ONLY** in tables with compound primary keys.

An attribute is consider to be partially dependent IF it is related on “part of” the compound primary key. Separating out the attribute with the portion of the compound primary key can stand as a table by themselves.

Steps:

1. Examine each non-key attribute against the primary key. IF it can stand by itself along with a portion of the primary key, then form a new table. IF the attribute needs ALL parts of the primary key, then it REMAINS in the table and is NOT a partial dependency.
2. For an attribute that is considered a partial dependency:
   1. Create a new table
   2. Copy the portion of the primary key to the new table and it will become the primary key of the new table.
   3. Move the primary dependency attribute from the original table into the new table.
3. Optionally at this point you could name your table. You MAY discover that you might need to rename tables already given a name. This is why this step is optionally and naming of table can wait until the 3NF form is done.

Note: as you proceed through these normalization levels you MAY discover that you need to refine the attribute name to BETTER identify the attribute. Here, it was discovered that DepartmentName in the Project table was confused with the DepartmentName to which the Employee belonged. One solution is to alter the attribute is it better identifies itself. DepartmentName became ProjectDepartmentName.

Employee (**EmployeeID**, EmployeeFirstName, EmployeeLastName DepartmentNumber, DepartmentName)

ProjectHours (**ProjectNumber**, ***EmployeeID***, WeeklyHours)

Project (**ProjectNumber,** ProjectName, ProjectDepartmentName )