Normalization is a technique of organizing the data in the database

Systematic approach of decomposing tables to eliminate data redundancy

Mutli step process that puts data into tabula form removing the duplicated data from it relational tables

Normalization used for 2 purpose

* Eliminate repeated data (not only make the process slow but may cause trouble in the later stages of transection)
* Ensure data dependencies make logical sense

Reason for normalization is because of “problems” that occurred In the data provided these are known as Data anomalies

If table not properly normalized and has data redundancy will waste memory and will cause the data base to be difficult to update

Type of anomalies

1. Insertion -> is the inability to add data to the database due to the absence of other data.

Example : Company hired in a new position in a company Mr Gerald is selected but the department ha not been allocated to him in that case if his data (department) needs to be updated in the system will put him a null

If got a lot more of the same scenario will all be null

1. Update -> data updated which is not completed leading to inconsistency and data redundancy

Example : Gerald leaves the company or change position from engineer to Production support, employee record needs to update is mistake any records are missed will lead to data inconsistency

1. Deletion -> when data that shouldn’t be deleted is deleted

Example: in a table filled with employee data. 2 information are kept together (lets say department and personal information) so if employee personal information is deleted will cause department record to be deleted

To prevent/eliminate these problems -> normalization.

1st normal form

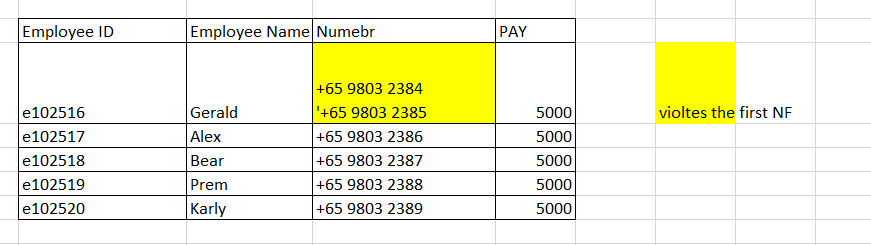
Use atomicity to solve the problem -> values in the table should not be further divided i.e single cell cannot hold multiple values

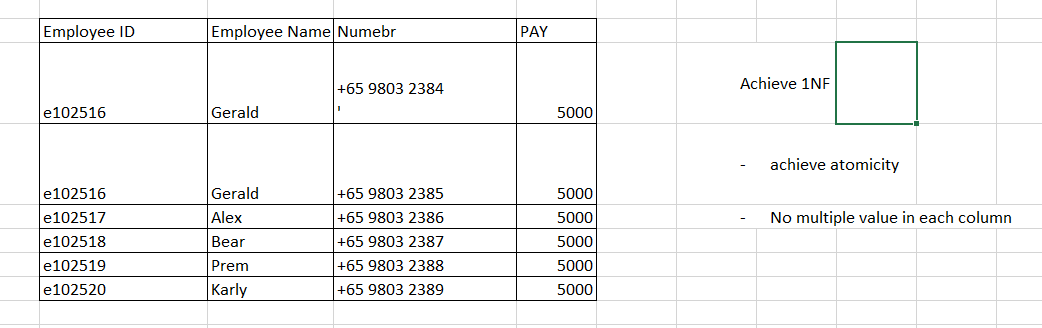
Following functions of 1st formal form

1)Remove repeating groups from the table

2)Create a separate table for each set of related data

3)identify each set of related data with a primary key



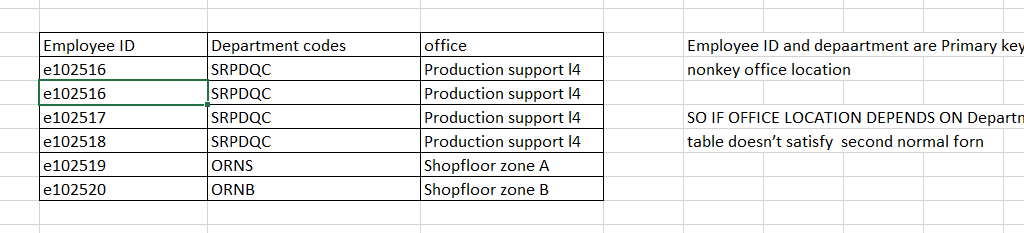


2nd NF

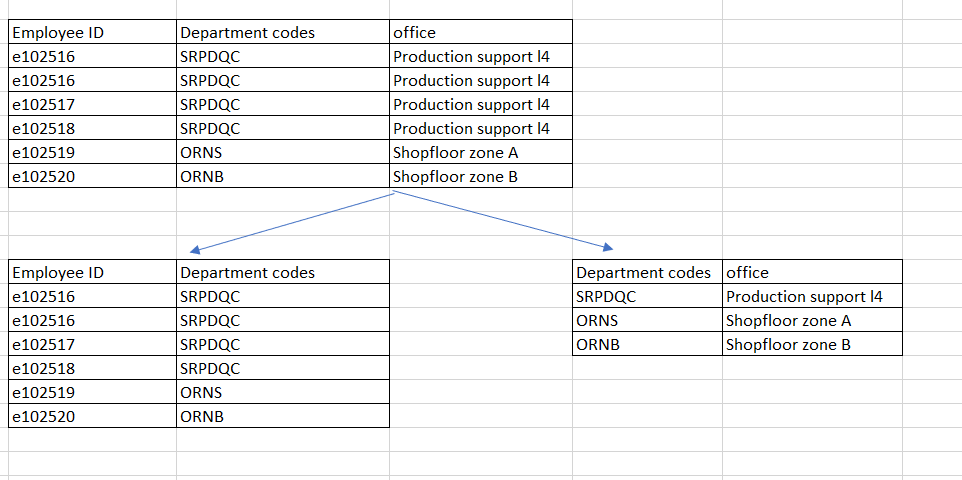
Has to fulfil the following condition

* In 1st NF
* Table should not contain partial dependency (proper subset of a candidate key determines a non-prime attribute)

Attributes that form a candidate key in a table are called prime attributes and the rest of the attributes of the relation are non-prime



To satisfy 2NF break table to 2 parts (removed partial functional dependency)



3NF – reduce the need for restructure , and remove the undesirable data anomalies, make data model more informative

Normalizing the table to reduce the duplication of data and ensure referential integrity

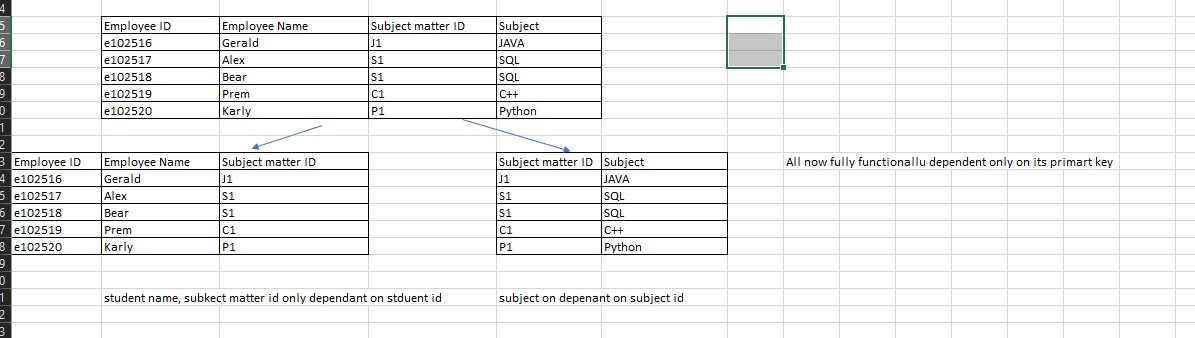
Has to fulfil the following condition

1. To be in 2NF
2. There should be no transitive dependency for non-prime attributes

Transitive decency

C deepened on B and B depend on A

So in 3rd NF all non prime attributes must depend only on prime attribute

Denormalization is the process of adding redundant column to the databse in order to improve performance.

This is to avoid additional joins. Contrains takes time to process and add complexity. So when optimizing for performance some items require less data protection, we can use 1NF instead