**5. Normalization**

**Normalization** is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. To overcome these anomalies we need to normalize the data by converting into normal forms as given below:

* First normal form(1NF)
* Second normal form(2NF)
* Third normal form(3NF)
* Boyce & Codd normal form (BCNF)

We have to perform these normalizations on DOCTOR, PATIENT, APPOINTMENT, DIAGNOSIS AND TREATMENT table.

**5.1 1NF**

As per the rule of first normal form, an attribute of a table cannot hold multiple values. It should hold only atomic values.

This doesn’t hold good in DOCTOR table as the attribute “phone” can be multi-valued, therefore is it broken into another table to bring the table into 1NF form.

Rest of the tables are in 1NF form, therefore no change.

After 1NF form the FDs obtained are as follows:

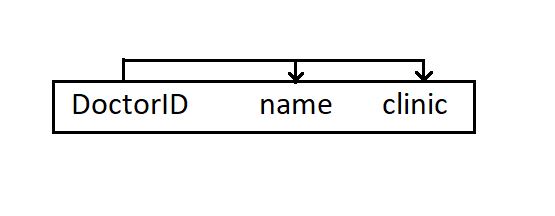


Table DOCTOR

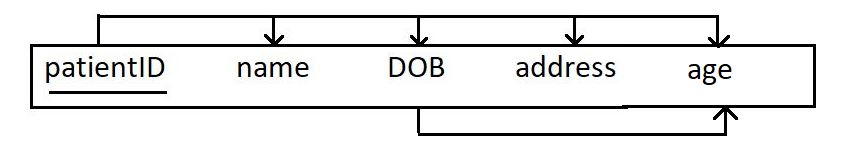


Table Patient

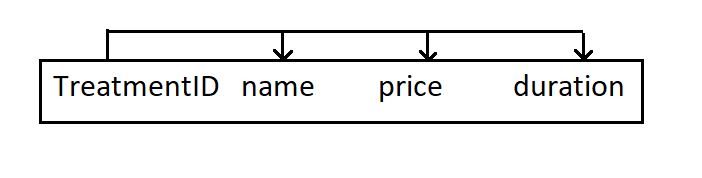


Table TREATMENT

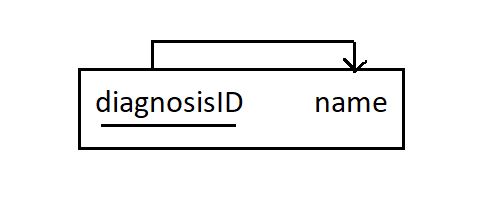


Table DIAGNOSIS

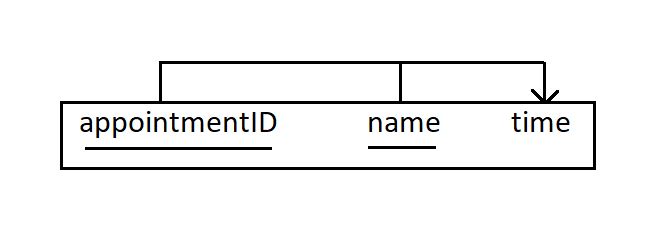


Table APPOINMENT

**5.2 2NF**

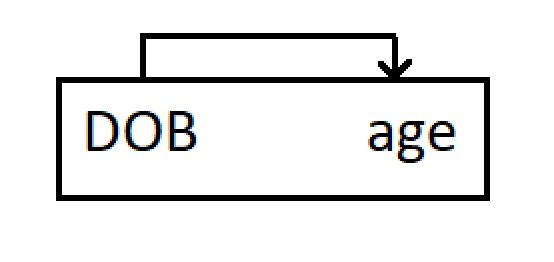
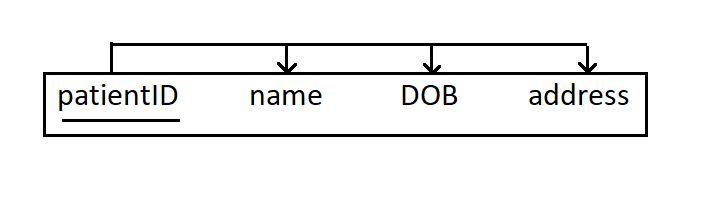
A table is said to be in 2NF if no non-prime attribute is dependent on the proper subset of any candidate key of table.

From the above figures we can see that no table violates 2NF form

**5.2 3NF**

A table design is said to be in 3NF if there is no transitive functional dependency of non-prime attribute to any prime-attribute.

In PATIENT table age is dependent on DOB hence giving a transitive relation which has to broken down into separate FDs to convert PATIENT into 3NF form.



Rest of the tables are in 3NF form.

**5.2 BCNF**

Table complies with BCNF if it is in 3NF and for every functional dependency X->Y, X should be the super key of the table.

From the above figures we can see that no table violates BCNF form