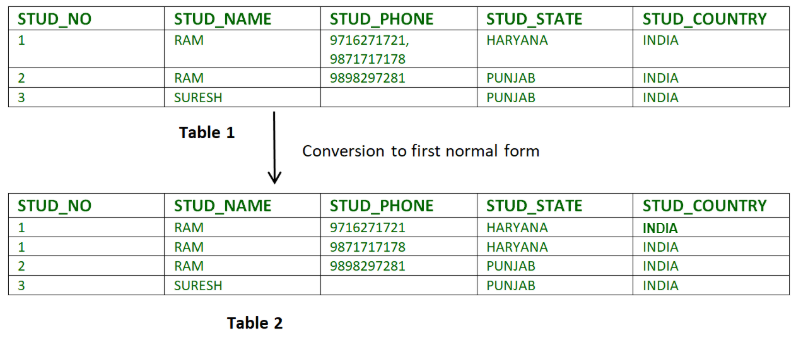
**Normalization** is the process of minimizing **redundancy** from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and update anomalies. So, it helps to minimize the redundancy in relations. **Normal forms** are used to eliminate or reduce redundancy in database tables.

**1. First Normal Form –**

If a relation contain composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is **singled valued attribute**.

**Example 1 –** Relation STUDENT in table 1 is not in 1NF because of multi-valued attribute STUD\_PHONE. Its decomposition into 1NF has been shown in table 2.



**Example 2 –**

ID Name Courses

------------------

1 A c1, c2

2 E c3

3 M C2, c3

In the above table Course is a multi-valued attribute so it is not in 1NF.

Below Table is in 1NF as there is no multi-valued attribute

ID Name Course

------------------

1 A c1

1 A c2

2 E c3

3 M c2

3 M c3

### 2. Second Normal Form –

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has **No Partial Dependency,**i.e.**,**no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

**Partial Dependency –** If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

* **Example 1 –** Consider table-3 as following below.
* STUD\_NO COURSE\_NO COURSE\_FEE
* 1 C1 1000
* 2 C2 1500
* 1 C4 2000
* 4 C3 1000
* 4 C1 1000
* 2 C5 2000

{Note that, there are many courses having the same course fee. }

Here,  
COURSE\_FEE cannot alone decide the value of COURSE\_NO or STUD\_NO;  
COURSE\_FEE together with STUD\_NO cannot decide the value of COURSE\_NO;  
COURSE\_FEE together with COURSE\_NO cannot decide the value of STUD\_NO;  
Hence,  
COURSE\_FEE would be a non-prime attribute, as it does not belong to the one only candidate key {STUD\_NO, COURSE\_NO} ;  
But, COURSE\_NO -> COURSE\_FEE, i.e., COURSE\_FEE is dependent on COURSE\_NO, which is a proper subset of the candidate key. Non-prime attribute COURSE\_FEE is dependent on a proper subset of the candidate key, which is a partial dependency and so this relation is not in 2NF.

To convert the above relation to 2NF,  
we need to split the table into two tables such as :  
Table 1: STUD\_NO, COURSE\_NO  
Table 2: COURSE\_NO, COURSE\_FEE

**Table 1** **Table 2**

STUD\_NO COURSE\_NO COURSE\_NO COURSE\_FEE

1 C1 C1 1000

2 C2 C2 1500

1 C4 C3 1000

4 C3 C4 2000

4 C1 C5 2000

2 C5

**NOTE:** 2NF tries to reduce the redundant data getting stored in memory. For instance, if there are 100 students taking C1 course, we don’t need to store its Fee as 1000 for all the 100 records, instead, once we can store it in the second table as the course fee for C1 is 1000.

* **Example 2 –** Consider following functional dependencies in relation  R (A,  B , C,  D )
* AB -> C [A and B together determine C]

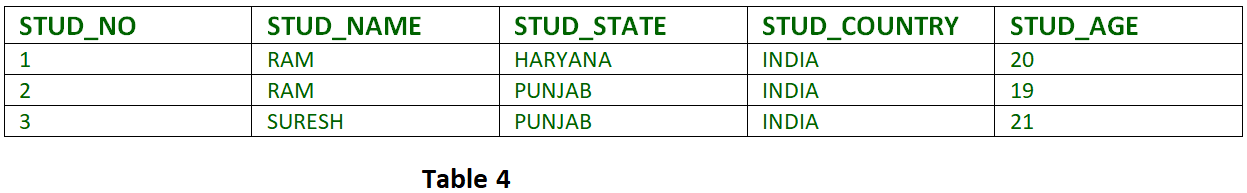
BC -> D [B and C together determine D]

In the above relation, AB is the only candidate key and there is no partial dependency, i.e., any proper subset of AB doesn’t determine any non-prime attribute.

### 3. Third Normal Form –

A relation is in third normal form, if there is **no transitive dependency** for non-prime attributes as well as it is in second normal form.  
A relation is in 3NF if **at least one of the following condition holds** in every non-trivial function dependency X –> Y

* 1. X is a super key.
  2. Y is a prime attribute (each element of Y is part of some candidate key).

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/Normalisation_normalforms_3.png)

**Transitive dependency –** If A->B and B->C are two FDs then A->C is called transitive dependency.

* 1. **Example 1 –** In relation STUDENT given in Table 4,

FD set: {STUD\_NO -> STUD\_NAME, STUD\_NO -> STUD\_STATE, STUD\_STATE -> STUD\_COUNTRY, STUD\_NO -> STUD\_AGE}  
Candidate Key: {STUD\_NO}

For this relation in table 4, STUD\_NO -> STUD\_STATE and STUD\_STATE -> STUD\_COUNTRY are true. So STUD\_COUNTRY is transitively dependent on STUD\_NO. It violates the third normal form. To convert it in third normal form, we will decompose the relation STUDENT (STUD\_NO, STUD\_NAME, STUD\_PHONE, STUD\_STATE, STUD\_COUNTRY\_STUD\_AGE) as:  
STUDENT (STUD\_NO, STUD\_NAME, STUD\_PHONE, STUD\_STATE, STUD\_AGE)  
STATE\_COUNTRY (STATE, COUNTRY)

* 1. **Example 2 –** Consider relation R(A, B, C, D, E)  
     A -> BC,  
     CD -> E,  
     B -> D,  
     E -> A  
     All possible candidate keys in above relation are {A, E, CD, BC} All attributes are on right sides of all functional dependencies are prime.

### 4. Boyce-Codd Normal Form (BCNF) –

A relation R is in BCNF if R is in Third Normal Form and for every FD, LHS is super key. A relation is in BCNF iff in every non-trivial functional dependency X –> Y, X is a super key.