

NORMALIZATION: INTRODUCTION

1. Normalization is a sequential process applied to organizing the data in the database.
2. Normalization is used to minimize the redundancy and to remove Insertion, Update, and Deletion Anomalies.
3. In Normalization the larger table is divided into smaller tables.
4. These tables are linked with each other with the help of keys.

TYPES OF NORMAL FORMS

Normalization is a step-by-step process in which total 7 stages are used such as:

1. 1NF also known as 1st Normalization Form
2. 2NF also known as 2nd Normalization Form
3. 3NF also known as 3rd Normalization Form
4. BCNF also known as Boyce Codd's Normalization Form
5. 4NF also known as 4th Normalization Form
6. 5NF also known as 5th Normalization Form
7. 6NF also known as 6th Normalization Form

Each step has its own functioning and importance.

1NF A relation is in 1NF if

All the contained values should be atomic values.

2NF A relation will be in 2NF if

It is in 1NF and

All non-key attributes are fully functional dependent on the key attribute or primary key

3NF A relation will be in 3NF if

It is in 2NF and

No transition dependency exists between relations

BCNF It is a strict version of 3NF

4NF A relation will be in 4NF if

It is in Boyce Codd's normal form and

Hold no multi-valued attribute dependency

5NF A relation is in 5NF If

It is in 4NF and

Should not hold any join dependency between relations

6NF Only proposed, no final or approved definition yet.

1ST NORMALIZATION FORM OR 1NF

1NF A relation is in 1NF if

All the contained values should be atomic values.

Means all multivalued and composite attributes are decomposed into atomic attributes.

SID	Name	Class	CID	Course	Mobile	Salutation
01	Rajesh	1 year	027, 049	DBMS, DSA	68759487	Mr.
02	Vikesh	2 year	032, 091	CN, OS	34980988	Mr.
03	Harsh	2 year	027	DBMS	29879878	Mr.
04	Jyoti	3 year	032	OS	48799807, 29875099	Miss

This is a simple database table and not normalized yet.

Let's convert this table in 1NF table.

SID	Name	Class	CID	Course	Mobile	Salutation
01	Rajesh	1 year	027	DBMS	68759487	Mr.
01	Rajesh	1 year	049	DSA	68759487	Mr.
02	Vikesh	2 year	032	CN	34980988	Mr.
02	Vikesh	2 year	091	OS	34980988	Mr.
03	Harsh	2 year	027	DBMS	29879878	Mr.
04	Jyoti	3 year	091	OS	48799807	Miss
04	Jyoti	3 year	091	OS	29875099	Miss

As we can see in the normalized table all the attributes which have multiple vales such as Course Id, Course Name and Contact number for each respective single entity, are now separated in individual lines. So, all the attributes have atomic values.

2nd NORMALIZATION FORM OR 2NF

2NF A relation will be in 2NF if

It is in 1NF and

All non-key attributes are fully functional dependent on the key attribute or primary key

Find out functional dependencies

Let's take a table which already have 1NF properties means all the attributes have atomic values.

SID	Name	Class	CID	Course	Mobile	Salutation
01	Rajesh	1 year	027	DBMS	68759487	Mr.
01	Rajesh	1 year	049	DSA	68759487	Mr.
02	Vikesh	2 year	032	CN	34980988	Mr.
02	Vikesh	2 year	091	OS	34980988	Mr.
03	Harsh	2 year	027	DBMS	29879878	Mr.
04	Jyoti	3 year	091	OS	48799807	Miss
04	Jyoti	3 year	091	OS	29875099	Miss

Find out functional dependencies

Sid → Name

Sid → Class

Sid → CID

Sid → Course

CID → Course

Sid → Mobile

Name → Salutation

Here in this table Course attributes have 2 functional dependencies, such as SID and CID

SID → Course

CID → Course

So now we should remove this partial dependency, so the resultant table is become as given below:

SID	Name	Class	CID	Mobile	Salutation
01	Rajesh	1 year	027	68759487	Mr.
01	Rajesh	1 year	049	68759487	Mr.
02	Vikesh	2 year	032	34980988	Mr.
02	Vikesh	2 year	091	34980988	Mr.
03	Harsh	2 year	027	29879878	Mr.
04	Jyoti	3 year	091	48799807	Miss
04	Jyoti	3 year	091	29875099	Miss

CID	Course
027	DBMS
049	DSA
032	CN
091	OS

So now we have two tables instead of one but these tables are interconnected with the help of key attributes such as CID is present both the tables.

3rd NORMALIZATION FORM OR 3NF

3NF A relation will be in 3NF if

It is in 2NF and

No transition dependency exists between relations

For any Non-Trivial functional dependency such as

$X \longrightarrow Y$

X is a super key

or

Y is a prime attribute, means each element of Y is part of some candidate key

What is Trivial dependency?

If $X \longrightarrow Y$ and $Y \longrightarrow Z$, means

$X \longrightarrow Y \longrightarrow Z$

This is called **Trivial dependency**.

Let's take a table which already have 2NF properties.

SID	Name	Class	CID	Mobile	Salutation
01	Rajesh	1 year	027	68759487	Mr.
01	Rajesh	1 year	049	68759487	Mr.
02	Vikesh	2 year	032	34980988	Mr.
02	Vikesh	2 year	091	34980988	Mr.
03	Harsh	2 year	027	29879878	Mr.
04	Jyoti	3 year	091	48799807	Miss
04	Jyoti	3 year	091	29875099	Miss

Find out functional dependencies

Sid \longrightarrow Name

Sid \longrightarrow Class

Sid \longrightarrow CID

Sid \longrightarrow Mobile

Name \rightarrow Salutation

Here we have found one trivial dependency such as $SID \longrightarrow NAME \longrightarrow Salutation$

So, we have to remove this

So now the resultant tables are as follows:

SID	Name	Class	CID	Mobile	Salutation_ID
01	Rajesh	1 year	027	68759487	1
01	Rajesh	1 year	049	68759487	1
02	Vikesh	2 year	032	34980988	1
02	Vikesh	2 year	091	34980988	1
03	Harsh	2 year	027	29879878	1
04	Jyoti	3 year	091	48799807	2
04	Jyoti	3 year	091	29875099	2

Salutation_ID	Salutation
1	Mr.
2	Miss

BOYCE CODD'S NORMALIZATION FORM OR BCNF

BCNF is a strict version of 3NF

Because in 3NF we have

$X \longrightarrow Y$

X is a super key

or

Y is a prime attribute, means each element of Y is part of some candidate key

But in BCNF

$X \longrightarrow Y$

X is a super key.

Means all the attributes must dependent on the super key only.

Let's take a table which already have 3NF properties.

SID	Name	Class	CID	Mobile	Salutation_ID
01	Rajesh	1 year	027	68759487	1
01	Rajesh	1 year	049	68759487	1
02	Vikesh	2 year	032	34980988	1
02	Vikesh	2 year	091	34980988	1
03	Harsh	2 year	027	29879878	1
04	Jyoti	3 year	091	48799807	2
04	Jyoti	3 year	091	29875099	2

Find out functional dependencies

Sid \longrightarrow Name

Sid \longrightarrow Class

Sid \longrightarrow CID

Sid \longrightarrow Mobile

Name \rightarrow Salutation ID

Here in this table Salutation_Id is dependent on Name. Here Name may be considered as a candidate key, but name may be repeated, so its not unique, so the Salutation_Id should be dependent on SID only, so we have to remove this dependency such as

Name \longrightarrow Salutation_Id

And make it as

SID → Salutation_Id

So now the resultant tables are as follows:

SID	Class	CID	Mobile	Salutation_ID
01	1 year	027	68759487	1
01	1 year	049	68759487	1
02	2 year	032	34980988	1
02	2 year	091	34980988	1
03	2 year	027	29879878	1
04	3 year	091	48799807	2
04	3 year	091	29875099	2

SID	Name
01	Rajesh
02	Vikesh
03	Harsh
04	Jyoti

4th NORMALIZATION FORM OR 4NF

4NF A relation will be in 4NF if

It is in Boyce Codd's normal form and

Hold no multi-valued attribute dependency

Let's take a table which already have BCNF properties.

SID	Class	CID	Mobile	Salutation_ID
01	1 year	027	68759487	1
01	1 year	049	68759487	1
02	2 year	032	34980988	1
02	2 year	091	34980988	1
03	2 year	027	29879878	1
04	3 year	091	48799807	2
04	3 year	091	29875099	2

Here CID, Mobile and Class are Multivalued Attributes

So, we should separate these attributes to make table simpler.

SID	CID	Salutation_ID
01	027	1
01	049	1
02	032	1
02	091	1
03	027	1
04	091	2

Table Main

SID	Class
01	1
02	2
03	2
04	3

Table Class

SID	Mobile
01	68759487
02	34980988
03	29879878
04	48799807
04	29875099

Table Mobile

Here we separate the Mobile attribute but we can't remove the CID attribute because it connects the Course table to this main table such as:

SID	CID	Salutation_ID
01	027	1
01	049	1
02	032	1
02	091	1
03	027	1
04	091	2

Table Main

SID	Class
01	1
02	2
03	2
04	3

Table Class

SID	Mobile
01	68759487
02	34980988
03	29879878
04	48799807
04	29875099

Table Mobile

CID	Course
027	DBMS
049	DSA
032	CN
091	OS

Table Course

5th NORMALIZATION FORM OR 5NF

5NF A relation is in 5NF If

It is in 4NF and

Should not hold any join dependency between relations

Or the Join should be lossless.

In simple words in the whole normalization process the complex table is divided into smaller and simpler table, but it does not mean that these all tables are separated from each other, instead these all are interconnected with each other with the help of keys. So, when we want to access the data, the whole data (separated and scattered in separate tables) should be shown as a single unit. For this these all tables perform Join Operation(pre-defined). So, in 5NF we confirmed that the “no data loss” is occurred during the normalization process and when the Join Operation is performed, the whole data is shown as a single unit. Means the Join should be lossless.

For example, we take our all tables which we previously used and check that are they fulfill 5NF regulations or not.

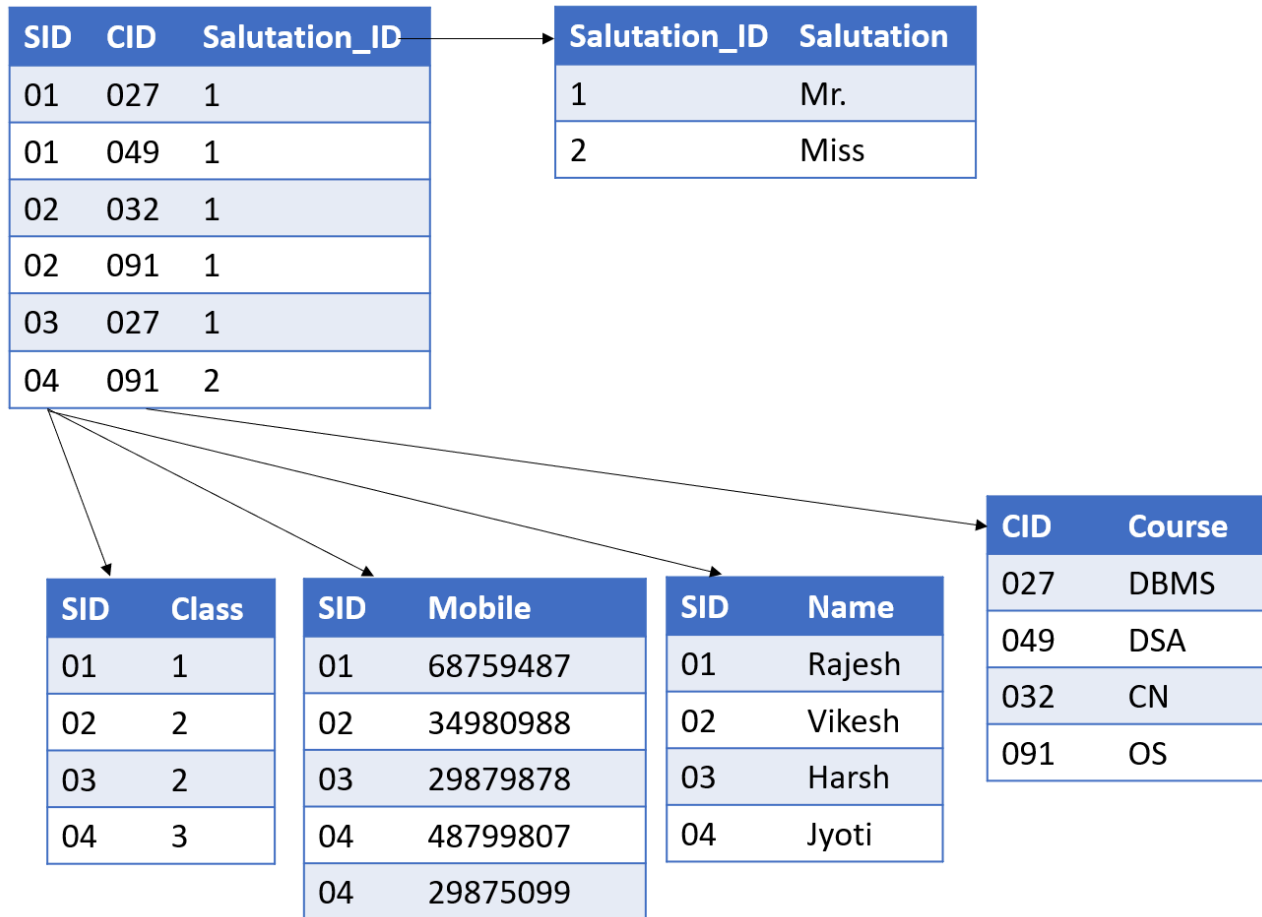
This is our original table

SID	Name	Class	CID	Course	Mobile	Salutation
01	Rajesh	1 year	027, 049	DBMS, DSA	68759487	Mr.
02	Vikesh	2 year	032, 091	CN, OS	34980988	Mr.
03	Harsh	2 year	027	DBMS	29879878	Mr.
04	Jyoti	3 year	032	OS	48799807, 29875099	Miss

This is Our 1NF table.

SID	Name	Class	CID	Course	Mobile	Salutation
01	Rajesh	1 year	027	DBMS	68759487	Mr.
01	Rajesh	1 year	049	DSA	68759487	Mr.
02	Vikesh	2 year	032	CN	34980988	Mr.
02	Vikesh	2 year	091	OS	34980988	Mr.
03	Harsh	2 year	027	DBMS	29879878	Mr.
04	Jyoti	3 year	091	OS	48799807	Miss
04	Jyoti	3 year	091	OS	29875099	Miss

These are the all resultant tables which we created during normalization process.



As we can see that all the tables are interconnected with the help of keys. So, when we want any data of any entity we can retrieve it without any loss.

