Types of Normalization

Types of Normal Form

- 1NF
- 2NF
- 3NF
- BCNF
- 4NF

1st Normal Form (1NF)

The values in each column of a table are atomic (No multi-value attributes allowed).
Each table has a primary key: minimal set of attributes which can uniquely identify a record
There are no repeating groups: two columns do not store similar information in the same table.

Example of a table not in 1NF :

Genres	**Name**	**Movie**
Marvel	Kevin Feige	The Avengers
		Captain America
DCEU	Zack Snyder	Batman Vs SuperMan
		Suicide Squad

This table contains Attribute values which are not single. This is not in Normalised form.
To make it into 1NF we have to decompose the table into atomic elements.

Table in 1NF after eliminating:

Genres	Name	Movie
Marvel	Kevin Feige	The Avengers
Marvel	Kevin Feige	Captain America
DCEU	Zack Snyder	Batman Vs Superman
DCEU	Zack Snyder	Suicide Squad

Second Normal Form (2NF)

A table is said to be in 2NF if both the following conditions hold:

- Table is in 1NF (First normal form)
- No non-prime attribute is dependent on the proper subset of any candidate key of the table.

Prime attribute :an attribute, which is a part of the prime-key, is known as a prime attribute.
Non-prime attribute : an attribute, which is not a part of the prime-key, is said to be a non-prime attribute.

Example of a table not in 2NF:

Studio	Movie	Budget	city
Marvel	Avengers	100	New York
Marvel	Captain America	120	New York

DCEU	Batman Vs Superman	150	Gotham
DCEU	Suicide Squad	75	Gotham

Here the Primary key is (studio, movie) and the city depends only on the studio and not on the whole key.
So, this is not in 2NF form.

Solution of 2 NF

Old Scheme ->{Studio, Movie, Budget, City}
New Scheme -> {Movie, Studio, Budget}
New Scheme ->{Studio, City}

Movie	**Studio**	**Budget**
The Avengers	Marvel	100
Captain America	Marvel	120
Batman Vs Superman	DCEU	150
Suicide Squad	DCEU	75

Studio	**City**
Marvel	New York
DCEU	Gotham

Now the 2 tables are in 2NF form.

Third normal form 3 NF

This form dictates that all non-key attributes of a table must be functionally dependent on a candidate key i.e. there can be no interdependencies among non-key attributes. For a table to be in 3NF, there are two requirements

- The table should be second normal form
- No attribute is transitively dependent on the primary key

Example of a table not in 3nf

Studio	CityTemp	Studio City
Marvel	96	New York
DCEU	99	Gotham
Fox	96	New York
Paramount	95	Hollywood

Here Studio is the primary key and both studio temp and city depends entirely on the Studio.

1. **Primary Key** ->**{Studio}**
2. **{Studio}** -> **{Studio City}**
3. **{StudioCity}** ->**{CityTemp}**
4. **{Studio}** -> **{CityTemp}**
5. **CityTemp transitively depends on Studio hence violates 3NF**

It is called **transitive dependency**.

Solution of 3NF
Old Scheme -> {Studio, StudioCity, CityTemp}
New Scheme-> {Studio, StudioCity}
New Scheme ->{StudioCity, CityTemp}

Studio	Studio City
Marvel	New York
DCEU	Gotham
FOx	New York
Paramount	Hollywood

Studio City	CityTemp
New York	96
Gotham	95
Hollywood	99

Boyce Codd Normal Form (BCNF) – 3.5NF

BCNF does not allow dependencies between attributes that belong to candidate keys. BCNF is a refinement of the third normal form in which it drops the restriction of a non-key attribute from the 3rd normal form.
Third normal form and BCNF are not same if the following conditions are true:

- The table has two or more candidate keys
- At least two of the candidate keys are composed of more than one attribute
- The keys are not disjoint i.e. The composite candidate keys share some attributes.

Example of BCNF

Scheme -> {MovieTitle, MovieID, PersonName, Role, Payment }

Key1 -> {MovieTitle, PersonName}

Key2 ->{MovieID, PersonName}

MovieTitle	MovieID	PersonName	Role	Payment
The Avengers	M101	Robert Downet Jr.	Tony Stark	200m
The Avengers	M101	Chris Evans	Chris Rogers	120m
Batman Vs Superman	D101	Ben Afflek	Bruce Wayne	180m
Batman Vs Superman	D101	Henry Cavill	Clarke Cent	125m
A walk to remember	P101	Mandy Moore	Jamie Sullivan	50m

Dependency between MovieID & MovieTitle Violates BCNF

Solution of BCNF

Place the two candidate primary keys in separate entities

Place each of the remaining data items in one of the resulting entities according to its dependency on the primary key.

New Scheme ->{MovieID, PersonName, Role, Payment}

New Scheme ->{MovieID, MovieTitle}

MovieID	PersonName	Role	Payment
M101	Robert Downey Jr.	Tony Stark	200m
M101	Chris Evans	Chris Rogers	125m
D101	Ben Afflek	Bruce Wayne	175m
D101	Henry Cavill	Clarke Cent	120m
P101	Mandy Moore	Jamie Sullivan	50m

MovieID	MovieTitle
M101	The Avengers
D101	Batman VS Superman
P101	A walk to remember

4 NF

Fourth normal form (4NF) is a level of database normalization where there are no non-trivial multivalued dependencies other than a candidate key.
It builds on the first three normal forms (1NF, 2NF and 3NF) and the Boyce- Codd Normal Form (BCNF).
It states that, in addition to a database meeting the requirements of BCNF, it must not contain more than one multivalued dependency.

Example of 4NF
Scheme -> {MovieName, ScreeningCity, Genre)

MovieName	**ScreeningCity**	**Genre**
The Avengers	Los Angeles	Sci-Fi
The Avengers	New York	Sci-Fi
Batman vs Superman	Santa Cruz	Drama
Batman vs Superman	Durham	Drama
A Walk to remember	New York	Romance

Many Movies can have the same Genre and Many Cities can have the same movie.
So this table violates 4NF .

Solution of 4NF
Move the two multi-valued relations to separate tables Identify a primary key for each of the new entities.
New Scheme -> {MovieName, ScreeningCity}
New Scheme -> {MovieName, Genre}

MovieName	**ScreeningCity**
Batman vs Superman	Santa Cruz
The Avengers	Los Angeles
A Walk to remember	New york
Batman vs Superman	Durham
The Avengers	New york

MovieName	**Genre**
Batman vs Superman	Drama
The Avengers	Sci-Fi
A Walk to remember	Romance

We split the table into two tables with one multivalued value in each.