Normalization

In DBMS

What is Normalization?

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.
- The normal form is used to reduce redundancy from the database table.

Why do we need Normalization?

The main reason for normalizing the relations is removing these anomalies.
Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows.
Normalization consists of a series of guidelines that helps to guide you in creating a good database structure.

Data modification anomalies can be categorized into three types:

- **Insertion Anomaly:** Insertion Anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.
- **Deletion Anomaly:** The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.
- **Updatation Anomaly:** The update anomaly is when an update of a single data value requires multiple rows of data to be updated.

Types of normalization

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Fifth Normal Form (5NF)

First Normal Form (1NF)

First normal form enforces these criteria:

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key

First Normal Form

Table_Product		
Product Id	Colour	Price
1	Black, red	Rs.210
2	Green	Rs.150
3	Red	Rs. 110
4	Green, blue	Rs.260
5	Black	Rs.100

This table is not in first normal form because the “Colour” column contains multiple Values.

After decomposing it into first normal form it looks like:

Product_id	Price
1	Rs.210
2	Rs.150
3	Rs. 110
4	Rs.260
5	Rs.100

Product_id	Colour
1	Black
1	Red
2	Green
3	Red
4	Green
4	Blue
5	Black

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 table.

An attribute that is not part of any candidate key is known as non-prime attribute.

SECOND NORMAL FORM

Table purchase detail		
Customer_id	Store_id	Location
1	1	Patna
1	3	Noida
2	1	Patna
3	2	Delhi
4	3	Noida

- This table has a composite primary key i.e. customer id, store id. The non key attribute is location. In this case location depends on store id, which is part of the primary key.

After decomposing it into second normal form it looks like:

Table Purchase	
Customer_id	Store_id
1	1
1	3
2	1
3	2
4	3

Table Store	
Store_id	Location
1	Patna
2	Delhi
3	Noida

Third Normal Form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

- Table must be in 2NF
- Transitive functional dependency of non-prime attribute on any super key should be removed.

An attribute that is not part of any candidate key is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:

- X is a super key of table
- Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

THIRD NORMAL FORM

Table Book Details			
Bood_id	Genre_id	Genre type	Price
1	1	Fiction	100
2	2	Sports	110
3	1	Fiction	120
4	3	Travel	130
5	2	sports	140

- In the table, book_id determines genre_id and genre_id determines genre type. Therefore book_idd determines genre type via genre_id and we have transitive functional dependency.

After decomposing it into third normal form it looks like:

TABLE BOOK		
Book_id	Genre_id	Price
1	1	100
2	2	110
3	1	120
4	3	130
5	2	140

TABLE GENRE	
Genre_id	Genre type
1	Fiction
2	Sports
3	Travel

Boyce-Codd Normal Form (BCNF)

- It is an advance version of 3NF that's why it is also referred as 3.5NF. BCNF is stricter than 3NF. A 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.

Boyce-Codd Normal Form

Student	Course	Teacher
Aman	DBMS	AYUSH
Aditya	DBMS	RAJ
Abhinav	E-COMM	RAHUL
Aman	E-COMM	RAHUL
abhinav	DBMS	RAJ

- KEY: {Student, Course}
- Functional dependency

  {student, course} -> Teacher

  Teacher-> Course
- Problem: teacher is not superkey but determines course.

After decomposing it into Boyce-Codd normal form it looks like:

Student	Course
Aman	DBMS
Aditya	DBMS
Abhinav	E-COMM
Aman	E-COMM
Abhinav	DBMS

Course	Teacher
DBMS	AYUSH
DBMS	RAJ
E-COMM	RAHUL

Fourth Normal Form (4NF)

- 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.

FOURTH NORMAL FORM

Student	Major	Hobby
Aman	Management	Football
Aman	Management	Cricket
Raj	Management	Football
Raj	Medical	Football
Ram	Management	Cricket
Aditya	Btech	Football
Abhinav	Btech	Cricket

- Key: {students, major, hobby}
- MVD: ->-> Major, hobby

After decomposing it into fourth normal form it looks like:

Student	Major
Aman	Management
Raj	Management
Raj	Medical
Ram	Management
Aditya	Btech
Abhinav	Btech

Student	Hobby
Aman	Football
Aman	Cricket
Raj	Football
Ram	Cricket
Aditya	Football
Abhinav	Cricket

Fifth Normal Form (5NF)

A database is said to be in 5NF, if and only if,

- It's in 4NF.
- If we can decompose table further to eliminate redundancy and anomaly, and when we re-join the decomposed tables by means of candidate keys, we should not be losing the original data or any new record set should not arise. In simple words, joining two or more decomposed table should not lose records nor create new records.

FIFTH NORMAL FORM

Seller	Company	Product
Aman	Coca cola company	Thumps Up
Aditya	Unilever	Ponds
Aditya	Unilever	Axe
Aditya	Uniliver	Lakme
Abhinav	P&G	Vicks
Abhinav	Pepsico	Pepsi

- Key: {seller, company, product}
- MVD: Seller ->-> Company, product

  Product is related to company.

After decomposing it into fifth normal form it looks like:

Seller	Product
Aman	Thumps Up
Aditya	Ponds
Aditya	Axe
Aditya	Lakme
Abhinav	Vicks
Abhinav	Pepsi

Seller	Company
Aman	Coca cola company
Aditya	Unilever
Abhinav	P&G
Abhinav	Pepsico

Company	Product
Coca cola company	Thumps Up
Unilever	Ponds
Unilever	Axe
Unilever	Lakme
Pepsico	Pepsi
P&G	Vicks

Every relation in BCNF is restricted to be in 3 NF.

Every relation in 3 NF is restricted to be in 2 NF.

Every relation in 2 NF is restricted to be in 1 NF

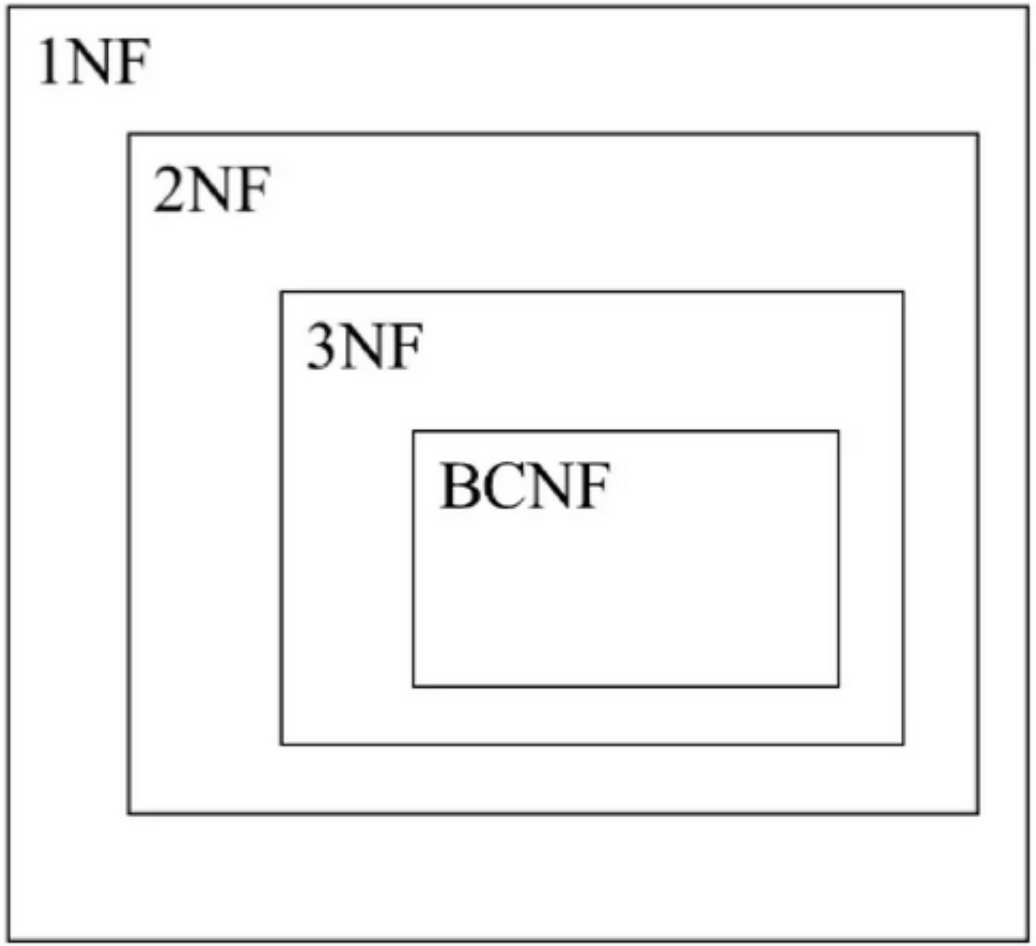
Dependency based

4NF & 5NF are also used for better schema design.

4NF is known as PROJECT JOIN NORMAL FORM .

5 NF is known as DOMAIN KEY NORMAL FORM .

Generally while designing a relational schema we always try to achieve highest normal form , which is possible by normalization.



a relation in BCNF, is also in 3NF

a relation in 3NF is also in 2NF

a relation in 2NF is also in 1NF