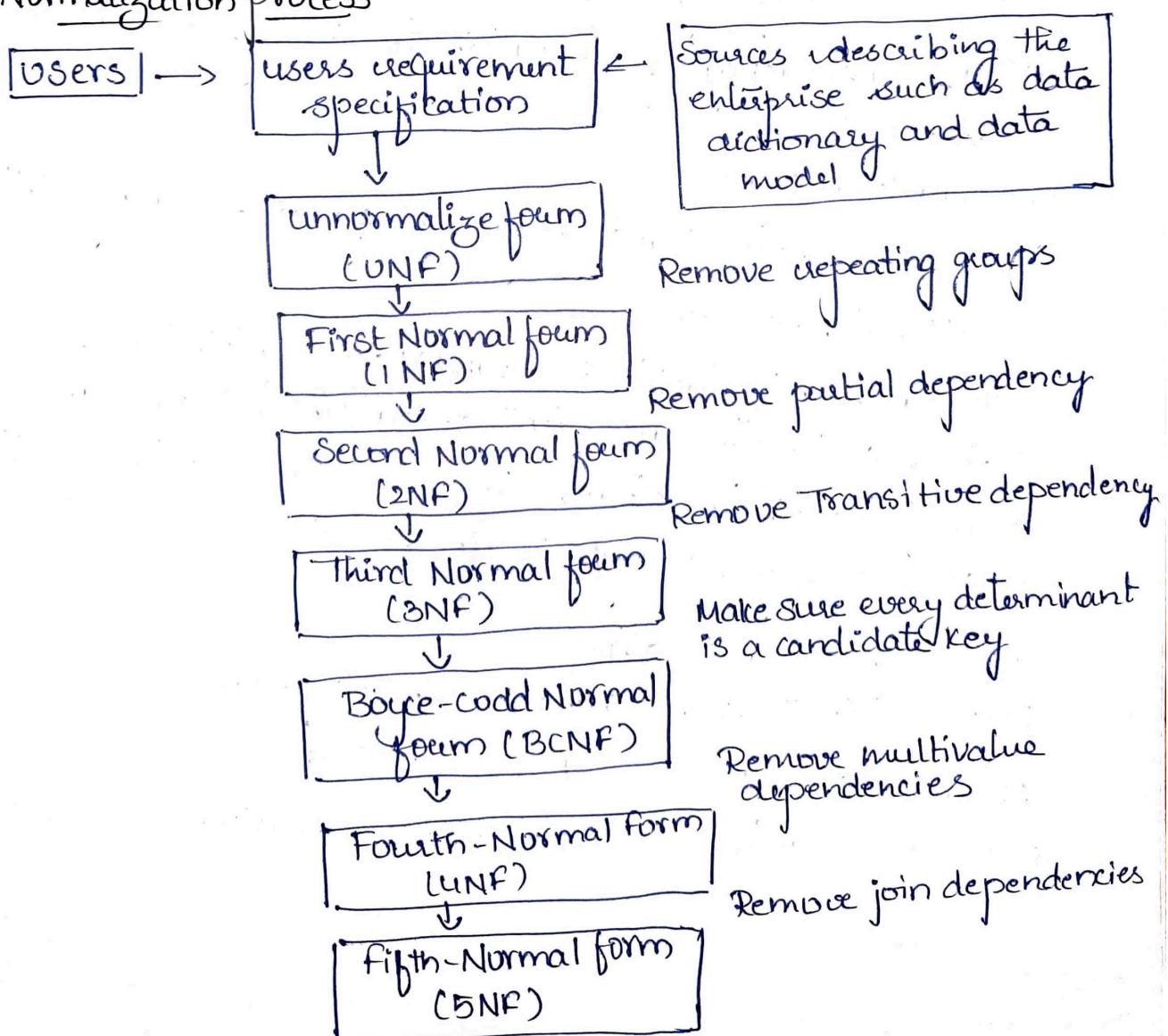


Normalization :- Database design is based on ER model but there exists some amount of inconsistency, redundancy, ambiguity. In order to remove the problems from the database design we need to do some refinement process, that process is called as normalization.

Normalization process involves bottom-up approach

Normalization is mainly used to eliminate data duplication and it allows insert, update and delete operations works more efficiently and improve performance of database.

Normalization process



Normalization is considered as a strong mathematical foundation. Before we discuss normalization let's discuss a few building blocks of normalization.

They are i] Determinant

- ii] Functional dependency
- iii] Full functional dependency
- iv] Partial dependency

v] Transitive dependency (or) indirect dependency.

Determinant:- In a given relation $R(x, y)$ x and y are attributes, where attribute x is said to be determinant if it uniquely defines the value of attribute y .

ii] The dependency of one attribute is represented as

$$x \rightarrow y$$

e.g:- marks \rightarrow grade

Here, the determinant may or may not be a key attribute.

Functional dependency

⇒ In a given relation R x and y are attributes, where y is functionally dependent on attribute x if and only if attribute x determines exactly one value of attribute y . Here x may be composite in nature.

e.g:- consider a relation

Report (cid, sid, sname, cname, DOB, marks, grade)

FD₁, marks \rightarrow grade FD₃ sid \rightarrow sname

FD₂, cid \rightarrow cname FD₄ sid, cid \rightarrow marks.

Partial Function dependency:-

In a given relation R, x and y are attribute, where y is partially function dependent on x if and only if attribute y depends on attribute x, where x may be composite in nature.

Full functional dependency:-

In a given relation R, x and y are attribute where y is fully functional dependent on attribute x if and only if attribute y is not functionally dependent on a subset of attribute x, where x may be composite in nature.

Transitive Functional dependency:-

$\text{sid, cid} \rightarrow \text{Marks}$
 $\text{Marks} \rightarrow \text{grade}$
 $\text{sid, cid} \rightarrow \text{grade}$

Indirect dependency

In a given relation R, x, y and z are attributes, where attribute z is fully functionally dependent on attribute y, attribute y is fully functionally dependent on x, so that attribute z is transitive dependent on attribute x, where x may be composite in nature.

i] Normal form (1NF):- A relation R is said to be in first Normal form if and only if it contains atomic values i.e. all the attributes of relation must contain atomic values. Atomic means which cannot be divisible further (or) the smallest level to which the data can be broken down and remain meaningful is called atomic.

Rules to maintain 1st normal form

- i] All columns or attribute must contain a single value
- ii] Attributes or columns in a relation must have a unique name
- iii] Domains of attribute must not be changed.
- iv] Order of data must be in any manner.

Items	Hues	cost	Items	Hues	cost
1.	Pink, black	15.99	1	Pink	15.99
2.	red	23.99	1	black	15.99
3.	black	17.50	2	red	23.99
4.	red, gray	9.99	3	black	17.50
5.	brown	29.99	4	red	9.99
			4	gray	9.99
			5	brown	29.99

- ii] Second Normal form (2NF) :- Relation R is said to be in 2NF if and only if
- i] it must be in 1NF
 - ii] no partial dependency exists b/w key and nonkey attributes

①	Candidate - Id	Course - Id	Candidate name	Course_name
	C829	A09	Beverly	css
	C736	A07	Sheldon	PHP
	C546	A03	Lennard	HTML
	C952	A05	Zach	Ruby

(2)

Teacher-ID	Subject	Teacher-age
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

1st table in Second Normal form

(1)

Candidate-ID	Course-id	Candidate-name	Course-ID	Course-name
C829	A09	Beverly	C829	CSS
C736	A07	Sheldon	C736	PHP
C546	A03	Leonard	C546	HTML
C952	A05	Zach	C952	Ruby

(2)

2nd table in Second Normal form

Teacher-id	Subject	Teacher-id	Teacher-age
25	Chemistry	25	30
25	Biology	25	30
47	English	47	35
83	Math	83	38
83	Computer	83	38

Third Normal form (3NF) :- A relation R is said to be in 3NF if and only if

- i] it must be in the 2NF
- ii] No transitive dependency exists b/w key and non-key attributes.

Emp-id	Emp-name	Emp-zip	Emp-state	Emp-city
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Ian	60007	US	Chicago
555	katharine	06389	UK	Norwich
666	john	462007	MP	Bhopal

Emp-zip	Emp-state	Emp-city	Emp-id	Empname	Empzip
201010	UP	Noida	222	Harry	201010
02228	US	Boston	333	Stephan	02228
60007	US	Chicago	444	Ian	60007
06389	UK	Norwich	555	katharine	06389
462007	MP	Bhopal	666	john	462007

Boyce-Codd Normal form (BCNF) :- Which will be defined as 3.5 NF. In a given relation R is said to be BCNF if and only if

- i] it must be in the 3NF.
- ii] Every determinant must be a candidate key
- iii] BCNF is also called as stronger 3NF but every 3NF is not a BCNF. The determinant which is

If a candidate key is in a relation then only the relation is said to be BCNF whereas in 3NF a determinant may or may not be a key attribute.

Sid-ID	email-ID	cid	Marks
S101	AAA@gmail.com	m ₃	60
S101	AAA@gmail.com	P ₃	70
S102	BBB@gmail.com	C ₃	80
S102	BBB@gmail.com	M ₃	90
S103	ccc@gmail.com	P ₃	70
S103	ccc@gmail.com	C ₃	80

In the above table we have two candidate keys.

- 1) Sid, cid
- 2) email-id, cid

and marks is a non-key attribute.

Hence the above table is in 3NF because the nonkey attribute i.e. marks is non-transitively and fully functionally depend upon the key attribute of the given relation.

- 1) Sid-id which will decides email-id.
- 2) email-id decides student-id
- 3) student-id and course-id which will decides the rest of the attribute of the relation
- 4) course-id and email-id which will decide the rest of the attributes in the given relation.

In the above 4 determinants two determinants combination of {sid-ID, cid} and {cid, email-ID} are candidate keys but

- ii] sid-id decides email-id, but sid-id alone is not a candidate key. Similarly email-id decides sid-id but email-id alone is not a candidate key
- iii] To make the above relation in BCNF we need to decompose the relation into two relations
- relation contains attribute of sid-id, email-id.
 - another relation contains attribute of sid-id, cid, marks

Result - R₁

sid-id	email-id
S101	AAA@gmail.com
S102	BBB@gmail.com
S103	CCC@gmail.com

sid-id	cid	marks
S101	M ₃	60
S101	P ₃	70
S102	C ₃	80
S102	M ₃	90
S103	P ₃	70
S103	C ₃	80

- The above relations are not only in 3NF but they are also in BCNF
- In table 1, sid-id decides email-id and email-id decides sid-id where both the determinants are candidate keys as well as the second table sid, cid decides marks where the determinant is the candidate key so that both the tables are in 3NF as well as BCNF

Properties of relational decomposition

Decomposition is a tool which is used to eliminate redundancy and it also allows not to occur any new problems.

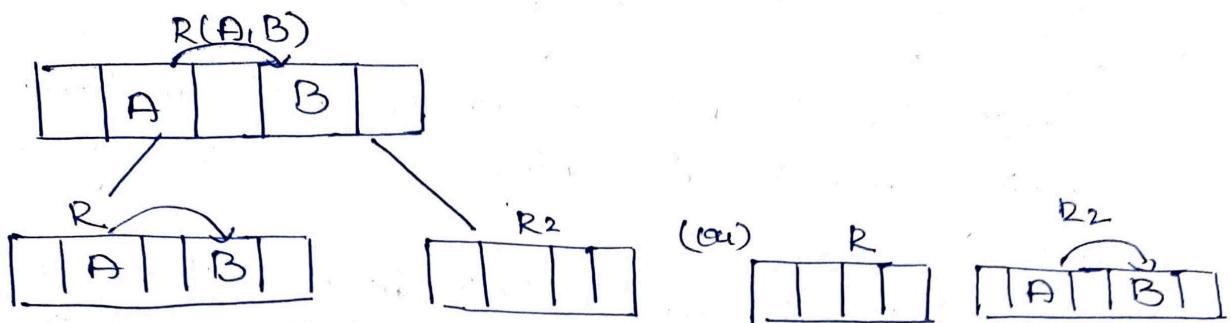
ii] Decomposition is used to check the recover the original relation from decomposed smaller relation without having loss of information which is called as loss less join decomposition property (or) non-additive decomposition property and it is also used to check whether the integrity constraints working efficiently or not. By using a property called as dependency preserving property.

The two properties of relational decomposition

- i) Lossless join decomposition property (or) non-additive decomposition property
- ii) Dependency preserving property.

Dependency preserving decomposition property
When there is a functional dependency of A determines B then whenever we see that attribute value of A, we must see the same attribute value for B which is called as preserving functional dependency.

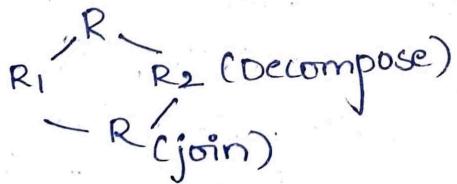
ii] In a given relation R where A and B are attributes where B is functionally dependent on A ($A \rightarrow B$). So when we decompose the relation R into two relations R_1 and R_2 , it must also hold functional dependency.



If a relation R is recomposed into multiple relations or smaller relation then the attributes which are holding functional dependencies must be present in any of the smaller relation.

Lossless decomposition property

Whenever a given relation R is decomposed into smaller relations and if you join the smaller relations in order to get the original relation by using join i.e. natural join without having any loss of information then that type of property is called as lossless join decomposition property (or) non-additive join decomposition property.

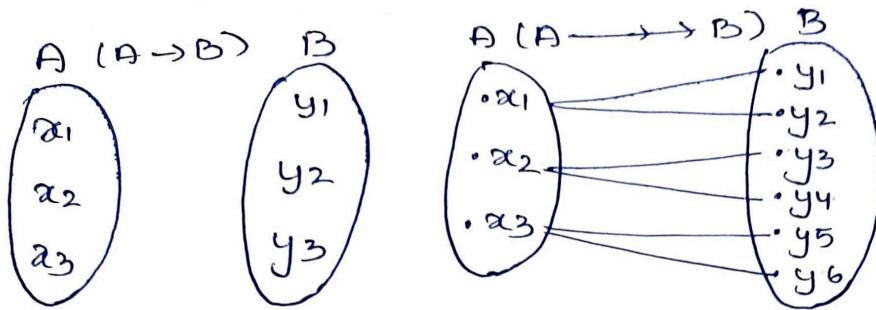


Multivalued dependencies

Multivalued dependencies is a dependency where one attribute value is potentially multi-valued fact about another attribute.

$$A \rightarrow B \text{ (FD)}$$

$$A \rightarrow\rightarrow B \text{ (MVD)}$$



Multivalue dependencies:- Multivalue dependencies are the dependency where two or more independent attribute or multivalued fact about the same attribute within the same relation.

Fourth Normal form (4NF) :- A relation R is said to be in the 4NF if and only if it satisfies the following condition

- i] Relation must be in 3NF (or) BCNF
- ii] No multivalue dependencies exist in the relation

Rules to transform a relation into 4NF

Consider a relation R with the following attributes $A, B, C \Rightarrow R(A, B, C)$ where B and C are multivalued facts of attribute A which exists a multivalued dependency in relation so that we need to decompose the relation into two relations, one relation contains (A, B) as attribute another relation contains (A, C) as attribute which comes under 4NF

$$\begin{array}{c} R(A, B, C) \\ / \quad \backslash \\ R_1(A, B) \quad R_2(A, C) \end{array}$$

Course-name	Sname	textbook
C	Ravi	x ₁
C	Ravi	x ₂
C	Arjun	x ₃
Java	Shiva	core-java
Java	Shiva	adv-java
SQL	Raj	B ₁
SQL	Raj	B ₂

cname	sname	cname	sname
C	Ravi	C	x ₁
C	Arjun	C	x ₂
java	Shiva	C	x ₃
sql	Raj	java	core-java
		java	core-java
		sql	B ₁
		sql	B ₂

Fifth- Normal form (5NF):- Join dependency refers lossless join decomposition property. This normal form is rarely used for practicals, but is mainly useful for theoretic purpose.

ii] join dependency refers decomposition of relation into multiple relation without any loss of information and it should maintain the dependencies of original relation.

iii] A relation R is said to be in 3NF if and only if the following conditions are satisfied.

a] It must be in 4NF

b] If the relation exist join dependency the decomposition the relations and the decomposition should be done in a way that further it should not be non-loss decomposed.

<u>Factory</u>	<u>Component</u>	<u>Project</u>
GM	Engine	MPC
GMI	Gearbox	125A
Honda	Engine	125A
GM	Engine	125A

A factory produces a component and the component is delivered to a project.

$R_1 : \text{Factory, component} \quad R_1 \bowtie R_2 \bowtie R_3 = R.$

$R_2 : \text{Component, project}$

$R_3 : \text{Factory, Project}$