

Normalization

'Normalization'

→ it is a technique to Remove or Reduce Redundancy from a table.

SID	Sname	Age
1	RAM	20
2	Vaishu	25
1	RAM	20

Row level

Student ID	Course ID	Course Name	Faculty ID	Faculty Name	Salary
1	C ₁	DBMS	F ₁	John	30000
2	C ₂	JAVA	F ₂	Bob	40000
3	C ₁	DBMS	F ₁	John	30000
4	C ₁	DBMS	F ₁	John	30000
:	:				

'Normalization'

→

Insertion

Anomaly

Delete from Student
where SID = 2

Deletion

Anomaly

Row level.

Update

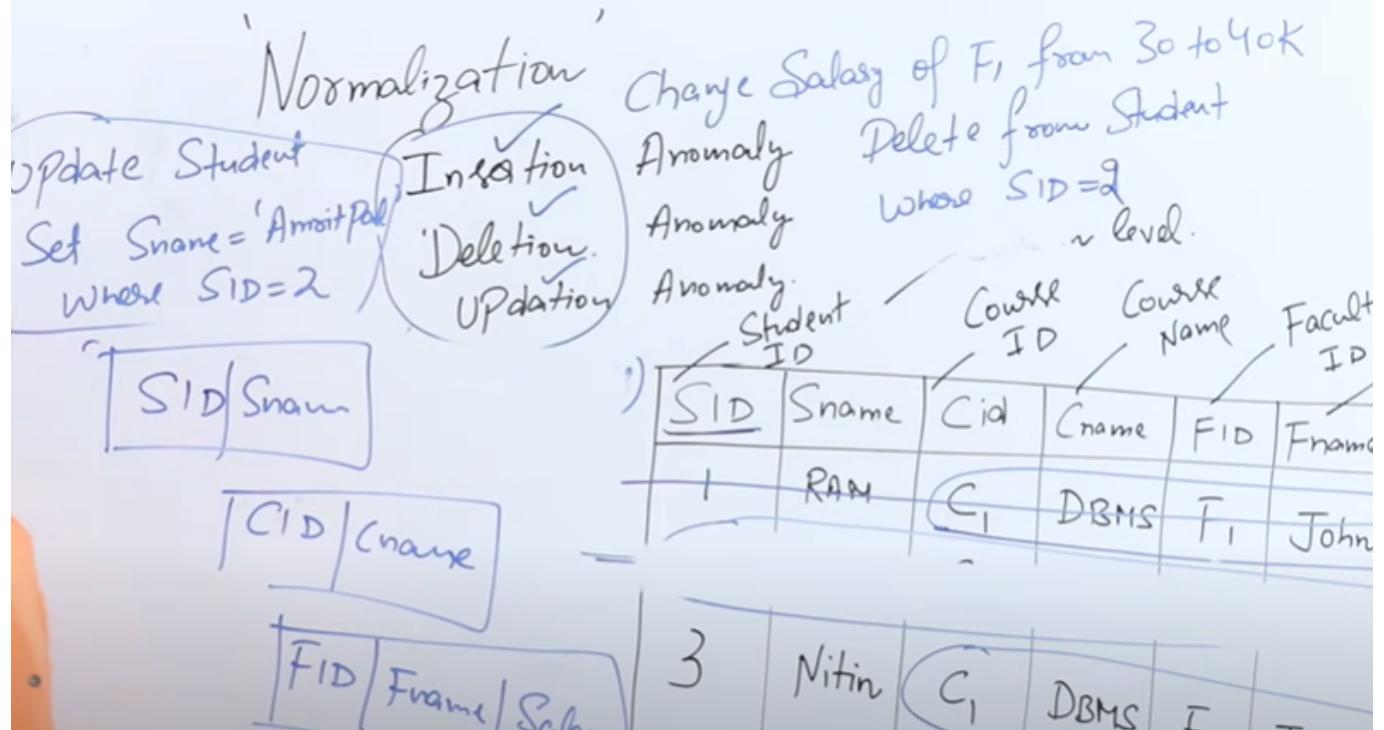
Anomaly

Primary Key (Unique + Not Null)

SID	Sname	Age
1	RAM	20
2	Vaishu	25
1	RAM	20

Row level

Student ID	Course ID	Course Name	Faculty ID	Faculty Name	Salary
1	C ₁	DBMS	F ₁	John	30000
2	C ₂	JAVA	F ₂	Bob	40000
3	C ₁	DBMS	F ₁	John	30000
4	C ₁	DBMS	F ₁	John	30000
:	:				



- **Atomic Values**: Each cell in a table contains indivisible, atomic values. Means a Relation should not contain any multivalued or composite attributes.
- **Unique Columns**: Each column must have a distinct name to identify the data it contains.
- **Primary Key**: A table in 1NF should have a primary key that uniquely identifies each record.
- **Eliminating Duplicates**: Duplicate rows are removed to prevent data redundancy.
- **Prime attribute**: A attribute is said to be prime if it is part of any of the candidate key
- **Non-Prime attribute**: A attribute is said to be non-prime if it is not part of any of the candidate key
- **Partial Dependency**: When a non – prime attribute is dependent only on a part (Proper subset) of candidate key then it is called partial dependency. (PRIME > NON-PRIME)
- **Full Dependency**: When a non – prime attribute is dependent on the entire candidate key then it is called Full dependency
- **Transitive Dependency**: A functional dependency from non-Prime attribute to non-Prime attribute is called transitive E.g. - R(A, B, C, D) with A as a candidate key $A \rightarrow B$ $B \rightarrow C$ [transitive dependency] $C \rightarrow D$ [transitive dependency]
- **Multivalued Dependency**
- Denoted by, $A \rightarrow\rightarrow B$, Means, for every value of A, there may exist more than one value of B.
- E.g. $S_name \rightarrow\rightarrow Club_name$ S_Name Club_name Kamesh Dance Kamesh Guitar

1NF

- Should not have multivalued attributes

First Normal Form

→ Table should not contain
any multivalued Attribute.

Student

Not in 1 st NF		
Rollno	Name	Course
1	Sai	C/C++
2	Harsh	Java
3	Onkar	C/DBMS

Table should not contain multivalued Attribute.

Not in 1st NF	
Rollno	Name
1	Sai
1	Harsh
2	Onkar

Course

~~C/C++~~

Java

~~C/DBMS~~

1

Primary Key:

Rollno	Name	Course
1	Sai	C
1	Sai	C++
2	Harsh	Java
3	Onkar	C
3	Onkar	DBMS

Rollno Courses

2

Primary Key: Rollno

Rollno	Name	Course1	Course2	Course3
1	Sai	C	C++
2	Harsh	Java	Null	Null Null
3	Onkar	C	DBMS	

3

Any

foreign key

Rollno	Name	Rollno	Course
1	Sai	1	C
2	Harsh	1	C++
3	Aniket	2	Java
		3	C
		3	DBMS

Base table

Base table.

Primary Key: Rollno

Primary Key: Rollno Course

foreign key: Rollno

2NF

- Relation R is in 2NF if :-
- R should be in 1 NF.
- R should not contain any Partial dependency. (that is every non-prime attribute should be fully dependent upon candidate key)

Q R(A, B, C) $B \rightarrow C$

A	B	C
a	1	X
b	2	Y
a	3	Z
C	3	Z
D	3	Z
E	3	Z

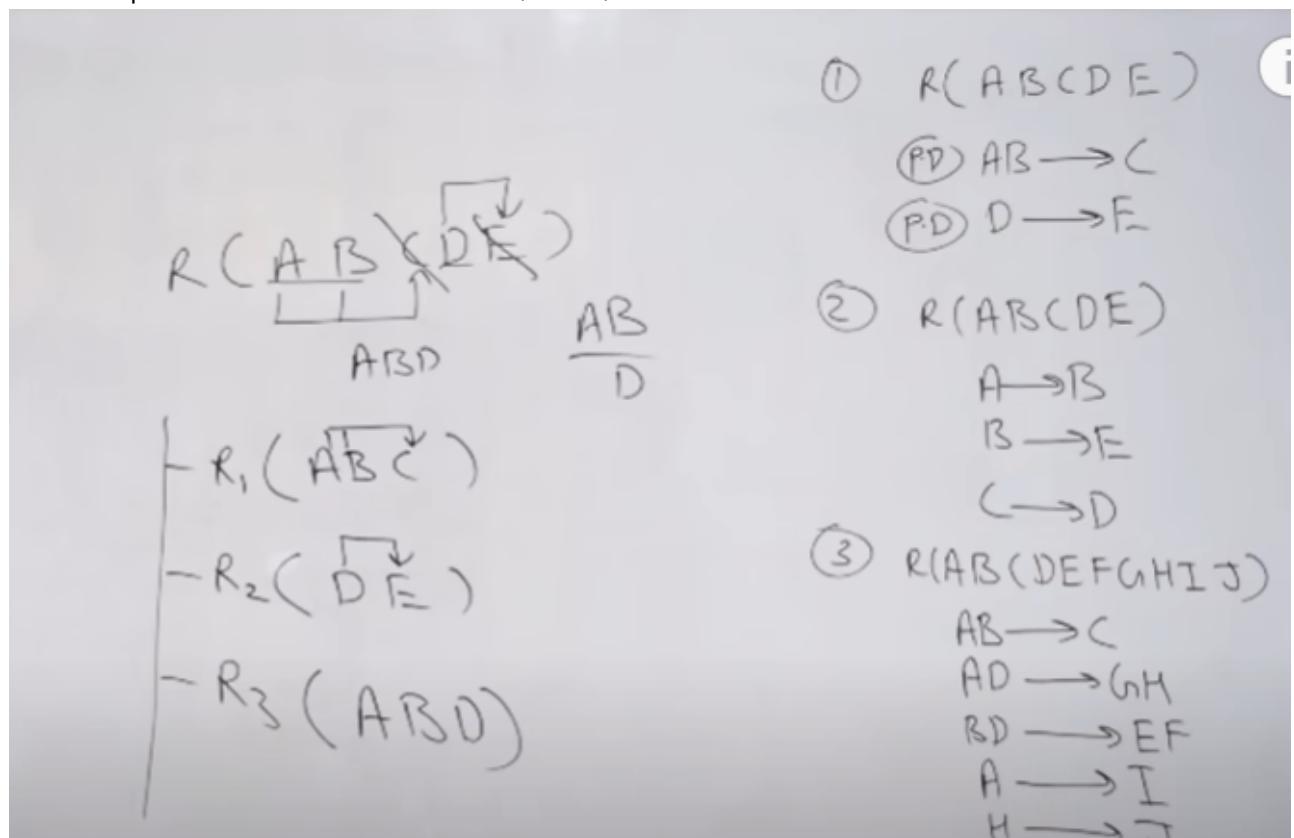
A	B
A	1
B	2
A	3
C	3
D	3
E	3

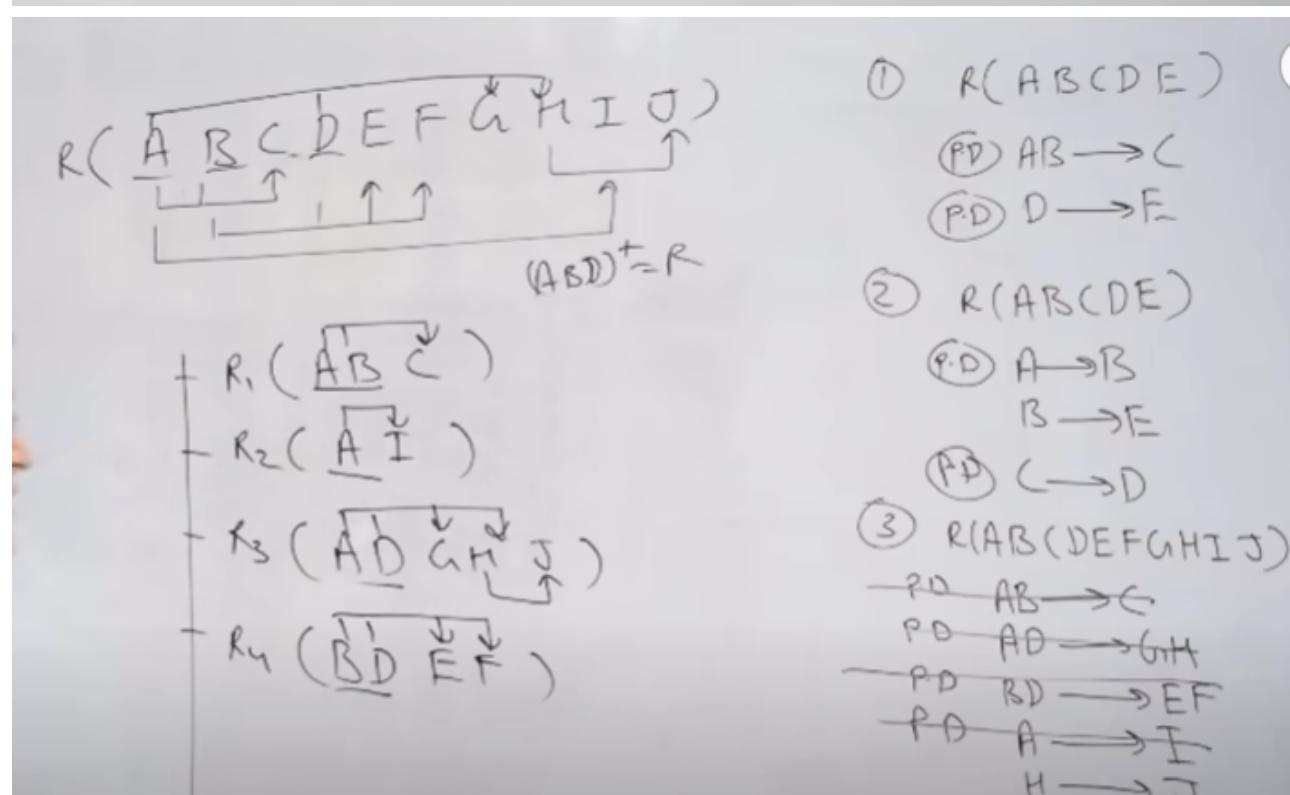
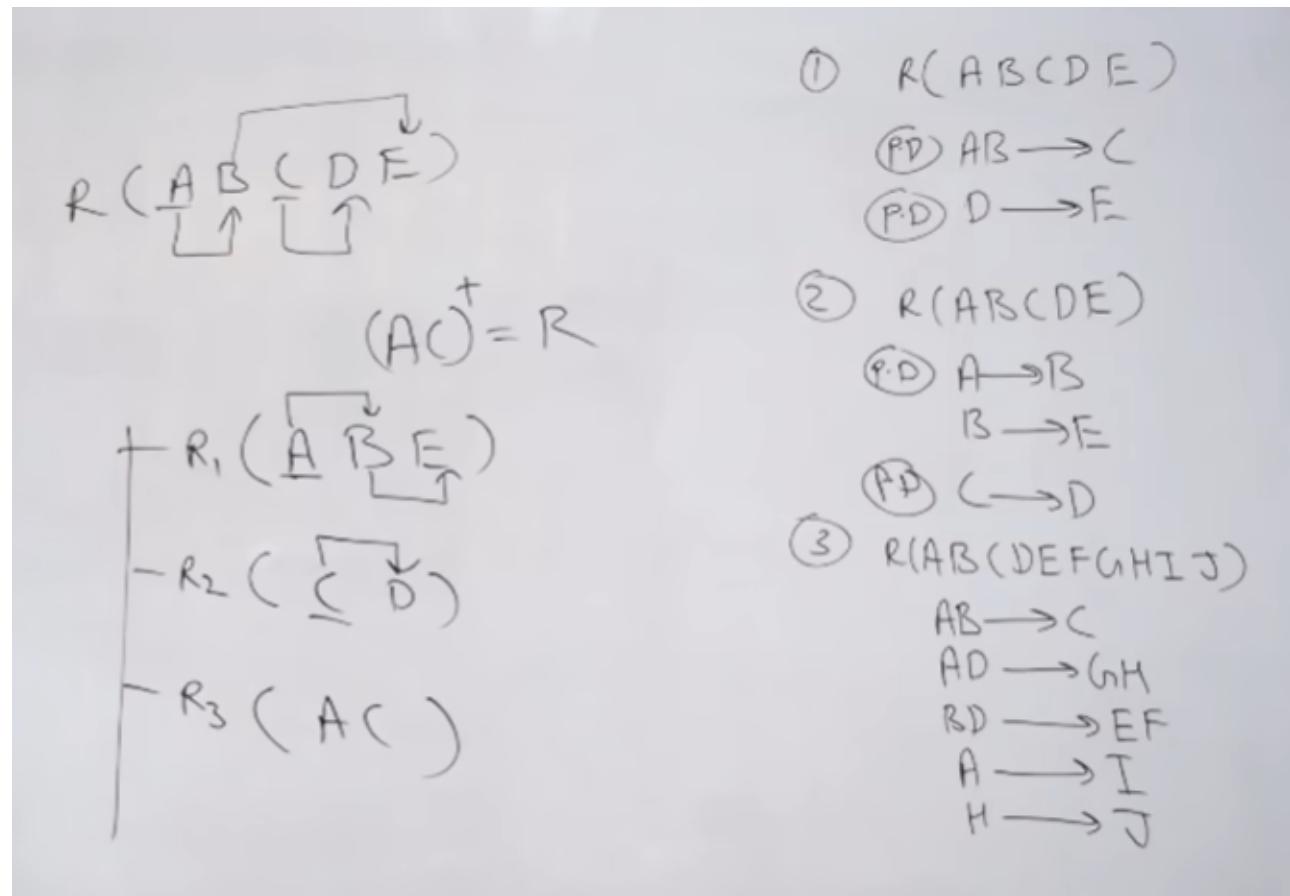
B	C
1	X
2	Y
3	z

Decompose to 2NF

1. Consider the universal relational schema R (A, B, C, D, E, F, G, H, I, J) and a set of following functional dependencies. F = {AB → C, A → DE, B → F, F → GH, D → IJ} determine the keys for R ? Decompose R into 2nd normal form.

- Answer : Its in 1NF
- To decompose in 2NF : Form 3 tables : AB, ADEIJ, BFGH

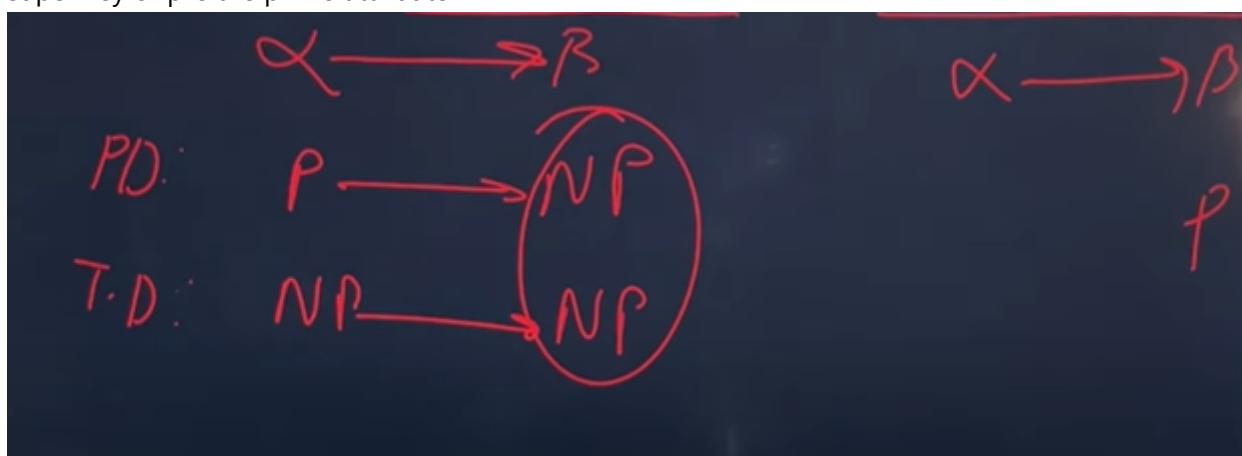




3NF

- Let R be the relational schema, it is said to be in 3 NF
- R should be in 2NF
- It must not contain any transitive dependency

- A relational schema R is said to be 3 NF if every functional dependency in R from $\alpha \rightarrow \beta$, either α is super key or β is the prime attribute



- NPA \rightarrow NPA NOT ALLOWED

$R(A, B, C)$

$A \rightarrow B$

$B \rightarrow C$

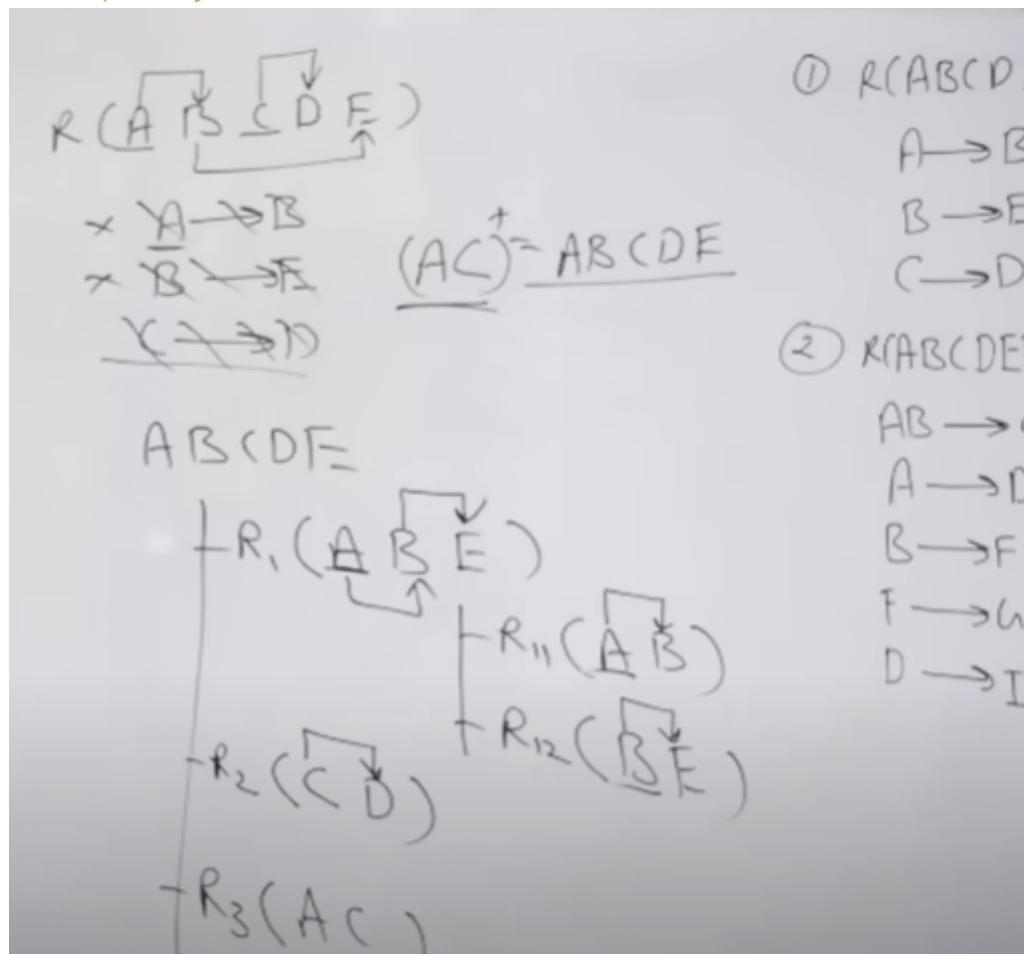
A	B	C
A	1	P
B	2	Q
C	2	Q
D	2	Q
E	3	R
F	3	R
G	4	S

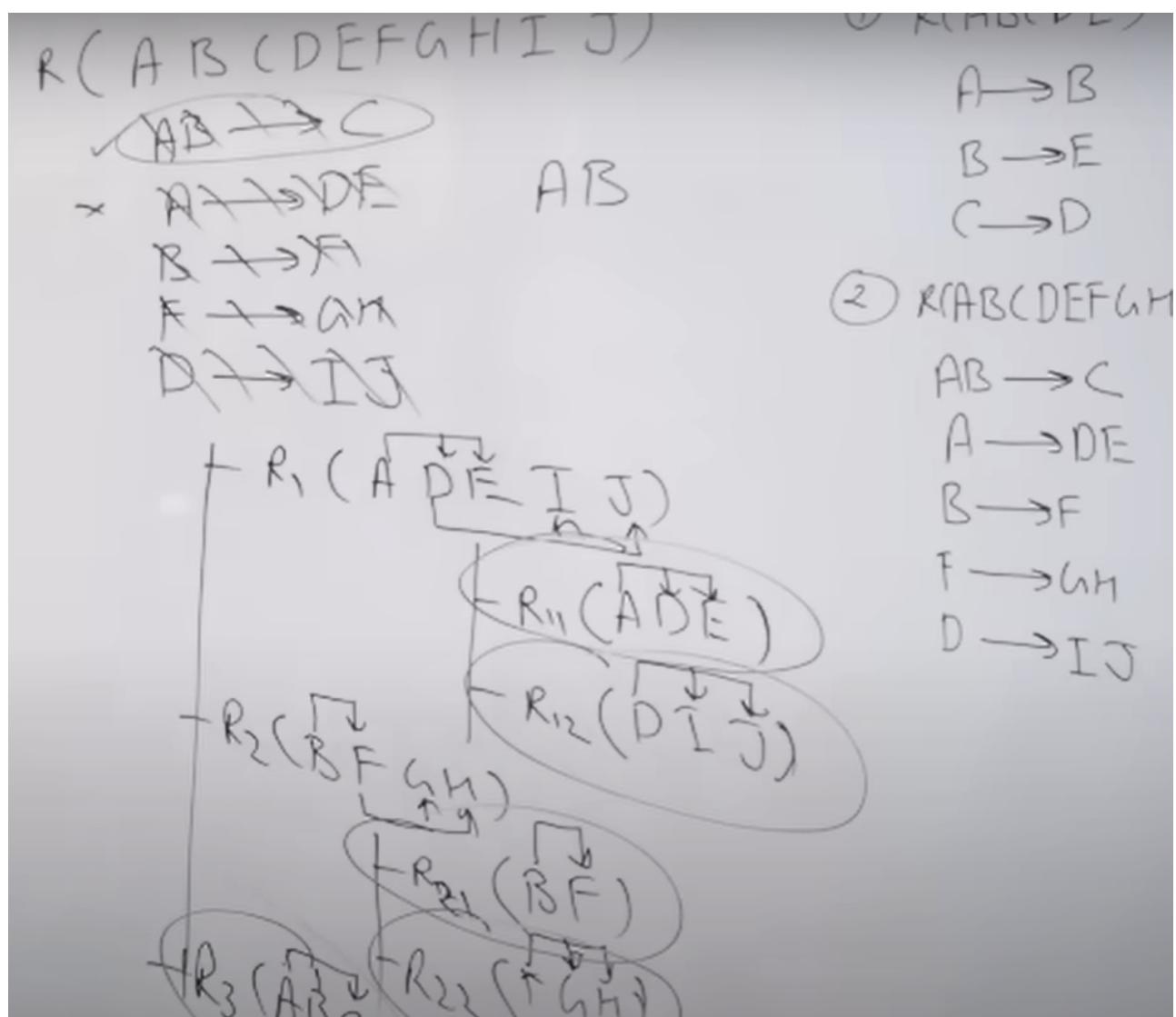
A	B
A	1
B	2
C	2
D	2
E	3
F	3
G	4

B	C
1	P
2	Q
3	R
4	S

Decompose to 3NF

- a - super key or b - PA





③ $R(ABCDE)$

$AB \rightarrow C$

AB

$\nwarrow BC \rightarrow D$

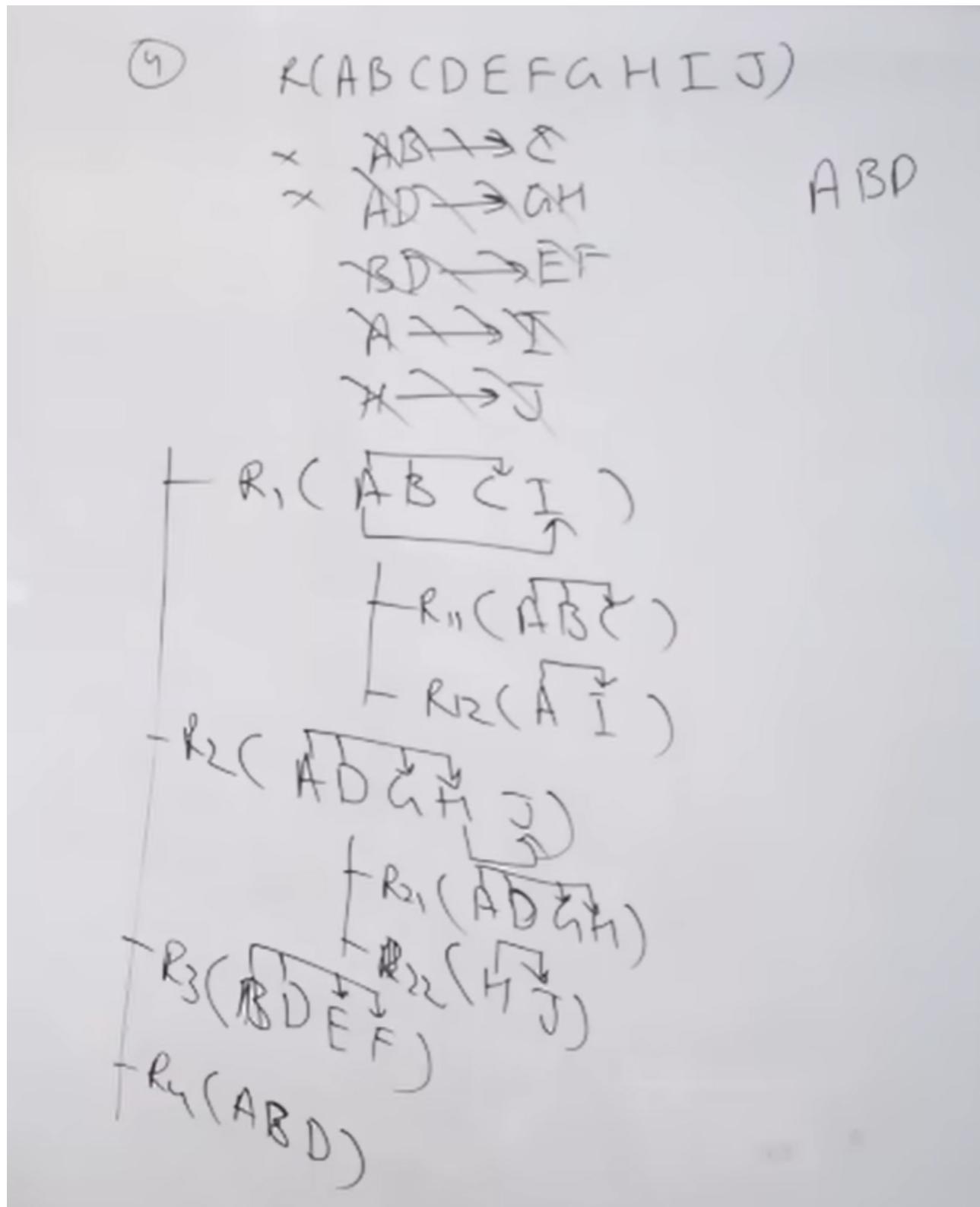
$D \rightarrow E$

+ $R_1(\underline{B} \underline{C} \underline{E})$

+ $R_{11}(B D)$

+ $R_{12}(D E)$

+ $R_2(\underline{A} \underline{B} \underline{C})$



BCNF

- A relational schema R is said to be BCNF if every functional dependency in R from α to β
- $\alpha \rightarrow \beta$
- α must be a super key or CK

Important Points

- A Relation with two attributes is always in BCNF.

- A Relation schema R consist of **only prime attributes** then R is always in **3NF**, but may or may not be in **BCNF**.
- Boyce-Codd Normal Form guarantees a good decomposition, without anomalies

$R(A, B, C)$	A	B	C	A	B	C	B
$AB \rightarrow C$	A	C	B	A	B	B	C
	B	B	C	B	B	C	B
$C \rightarrow B$	B	A	D	B	A	D	A
	A	A	E	A	A	E	A
	C	C	B	C	C	C	B
	D	C	B	D	C	D	A
	E	C	B	e	C	E	A
	F	C	B	f	c		

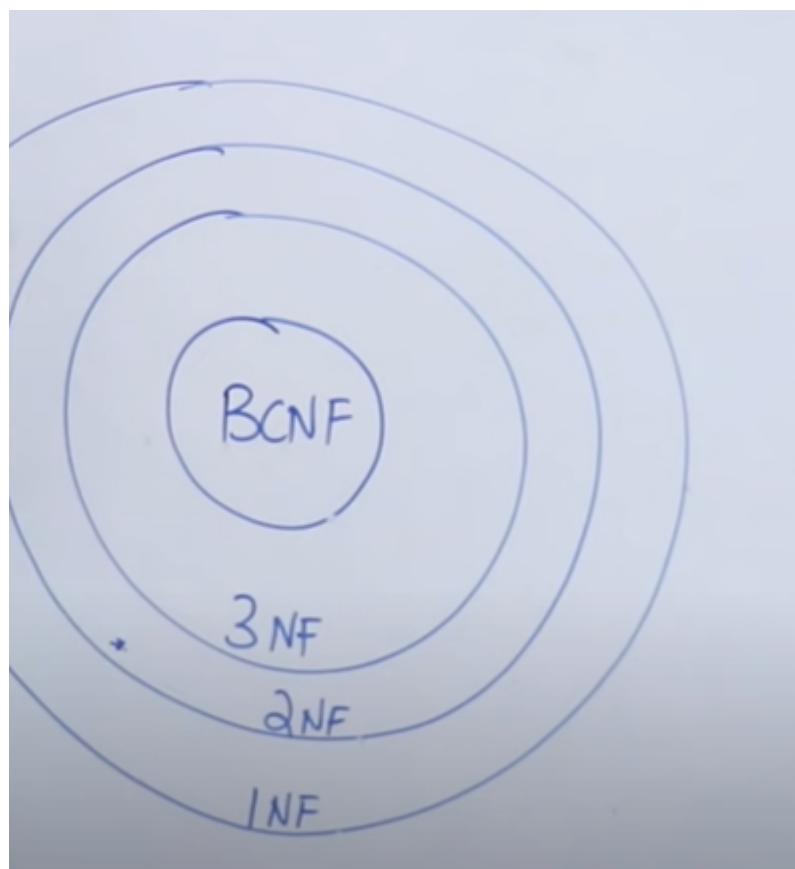
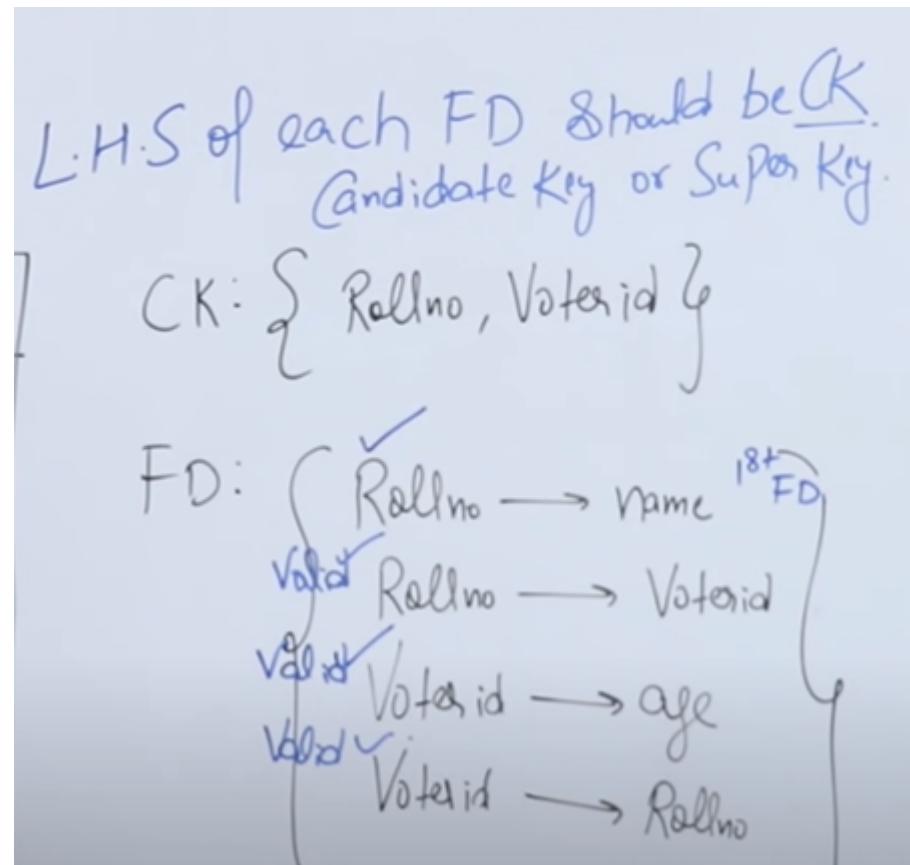
$BCNF$ (Boyce Codd Normal Form)

Student:

Rollno	Name	Voterid	age
1	Ravi	K0123	20
2	Varun	M034	21
3	Ravi	K786	23
4	Rahul	D286	21

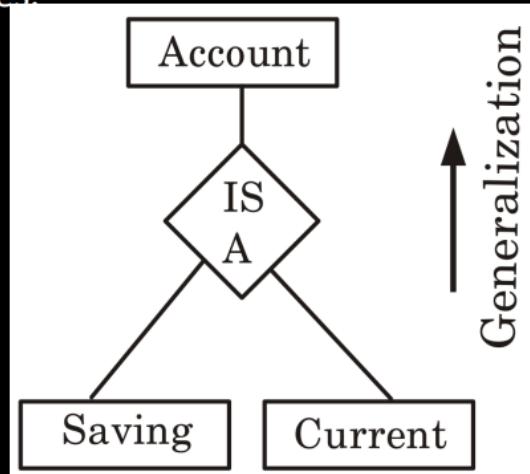
CK: { Rollno, Voterid }

FD: {
 $\text{Rollno} \rightarrow \text{name}$
 $\text{Rollno} \rightarrow \text{Voterid}$
 $\text{Voterid} \rightarrow \text{age}$
 $\text{Voterid} \rightarrow \text{Rollno}$ }



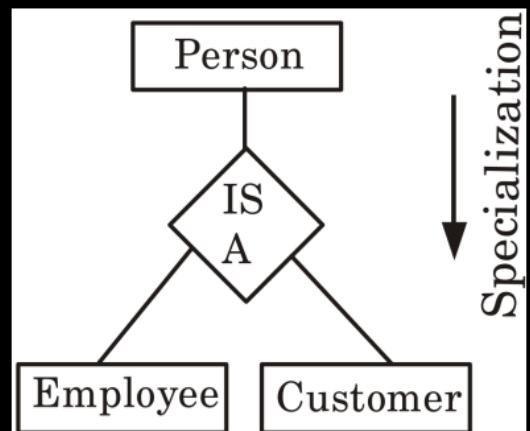
Generalization

- Involves merging two lower-level entities to create a higher-level entity.
- A bottom-up approach that builds complexity from simpler components.
- Highlights similarities among lower-level entity sets while hiding differences.
- Leads to a simplified, structured data representation, aiding in database design and querying processes.



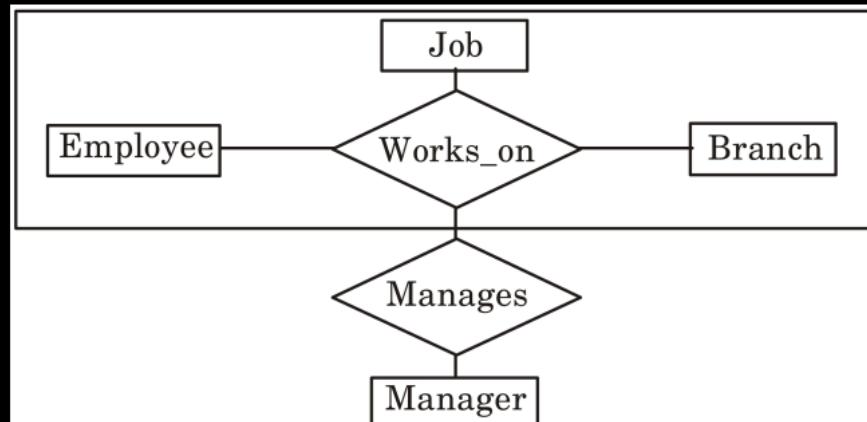
Specialization

- A process where a higher-level entity is broken down into more specific, lower-level entities.
- This top-down approach delineates complexity into simpler components.
- Acts as the converse of the generalization process, focusing on differentiating properties rather than similarities.



Aggregation

- A concept wherein relationships are abstracted to form higher-level entities, enabling a more organized representation of complex relationships.



Advantages of BCNF

1. No anomalies : ✓ (Due to no redundancy)
2. Lossless Join : ✓
3. Dependency Preservation : X

Summary

1. 1NF : DEFAULT(Unless told otherwise)
2. 2NF : Partial Dependency (0, 100)
3. 3NF : Transitive Dependency (NPA \rightarrow NPA not be there)
4. BCNF : left should be candidate/super key

1 st Normal form	2 nd Normal form	3 rd Normal form	BCNF
<ul style="list-style-type: none"> * No Multi-valued attribute - Only Single valued 	<ul style="list-style-type: none"> * In 1st NF + No Partial Dependency + Only Full Dependency <p>$(AB) \rightarrow C$</p>	<ul style="list-style-type: none"> * In 2nd NF + No Transitive Dependency + No Non-Prime should determine non-prime <p>$X \rightarrow Y \rightarrow Z$</p>	<ul style="list-style-type: none"> * In 3rd NF + L.H.S must be CK or SK

$R(ABCDEF)$, Check the Highest Normal form?

FD: $\{AB \rightarrow C, C \rightarrow DE, E \rightarrow F, F \rightarrow A\}$

Step1: Find all CKs in Relation

CK: $\{AB, FB, EB, CB\}$

Step2: Write all Prime Attributes.

$\{A, B, C, E, F\}$

Step3: Write all Non-Prime Attrib.

$\{D\}$

$$\begin{aligned} A^+ &= A \\ B^+ &= B \\ AB^+ &= ABCDEF \\ FB^+ &= FBACDE \\ EB^+ &= EBFACD \\ CB^+ &= CBDEFA \end{aligned}$$

L.H.S is proper subset | R.H.S Non Pr.

	$AB \rightarrow C$	$C \rightarrow DE$	$E \rightarrow F$	$F \rightarrow A$
BCNF	✓	X	X	X
3NF	✓	X	✓	✓
2NF	✓	X	✓	✓
1NF	✓	✓	✓	✓

Ques. Which of the following statement is true?

- A) A relation is in 3rd NF then it is always in BCNF
- B) A relation is in 2nd NF then it is not in 1st NF
- C) A relation is in BCNF then it is in 2 NF
- D) A relation is in 2NF then it contains partial dependency.

Ques. Relation R has eight attributes ABCDEFGH.

$$F = \{CH \rightarrow G, A \rightarrow BC, B \rightarrow CFH, E \rightarrow A, F \rightarrow EG\}$$

How many candidate keys in R

- A) 3 b) 4 c) 5 d) 6

C, B

Decomposition

1. Dependency Preservation Decomposition - 3NF, Not BCNF

2. Lossless Decomposition - All NF

- Example
 - $R(ABCD) = \{AB \rightarrow CD, D \rightarrow A\}$
 - CK = {AB, BD}
 - PA = A, B, D
 - \Rightarrow 3NF, Not in BCNF ($D \rightarrow A$)
 - 2 Tables - DA{ $D \rightarrow A$ }, BCD{ $BD \rightarrow C$ } : AB \rightarrow CD missed : Not Preserved
-

Minimal Cover

For the following Functional dependencies, find the correct Minimal Cover

$$\{ A \rightarrow B, C \rightarrow B, D \rightarrow ABC, AC \rightarrow D \}$$

- a) $A \rightarrow B, C \rightarrow B, D \rightarrow A, AC \rightarrow D$
- b) $A \rightarrow B, C \rightarrow B, D \rightarrow C, AC \rightarrow D$
- c) $A \rightarrow BC, D \rightarrow CA, AC \rightarrow D$
- d) $A \rightarrow B, C \rightarrow B, D \rightarrow AC, AC \rightarrow D$

Step 1 : Break down : $D \rightarrow ABC$ and not $AC \rightarrow D$
 Step 2 : Remove redundant dependencies : Take closures of LHS and check
 Step 3 : LHS should have single attribute : $AC \rightarrow D$, If $C^* \rightarrow A$ or $A^* \rightarrow C$
 Answer - D

Equivalence of Functional Dependencies

$$X = \{A \rightarrow B, B \rightarrow C\} \quad Y = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$$

1. If X covers Y
2. If Y covers X

Steps :

1. Take closures of LHS of Y from X and check if RHS comes or not
 2. Same do for X
-
-

$$R(A,B,C,D,E) \quad AB \rightarrow C : PD \quad D \rightarrow E : PD$$

CK : ABD

Not in 2F : bcoz of PD

Decomposition

R1(ABC) : AB->C

R2(DE) : D->E

R3(ABD)

2nf

CK - AB

AB -> C,

A -> DE, : Partial Dependency B -> F, : Partial Dependency F -> GH, D -> IJ

R1(ADEIJ) : A A->DE D->IJ

R2(BFGH) : B B->F F->GH

R3(ABC) : AB AB->C

3NF

R1(ADEIJ) : A R11(DIJ) D->IJ R12(ADE) A->DE

R2(BFGH) : B R21(BF) B->F R22(FGH) F->GH

R3(ABC) : AB AB->C

Final Decomposition :

R11(DIJ) : D D->IJ

R12(ADE) : A A->DE

R21(BF) : B B->F

R22(FGH) : F F->GH

R3(ABC) : AB AB->C

BCNF

Same as 3NF