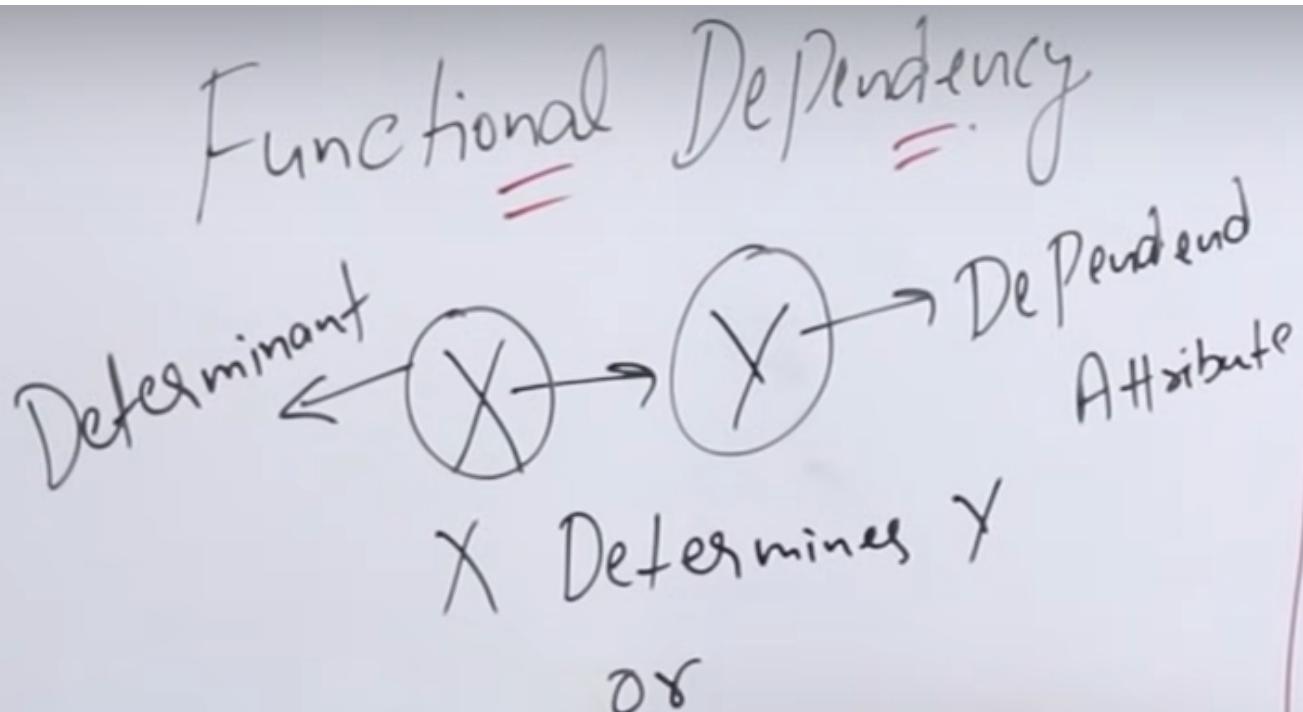


Functional Dependency

FUNCTIONAL DEPENDENCY

- A formal tool for analysis of relational schemas.
- In a Relation R, if ' α ' \sqsubseteq R AND ' β ' \sqsubseteq R, then attribute or a Set of attribute ' α ' Functionally derives an attribute or set of attributes ' β ', iff each ' α ' value is associated with precisely one ' β ' value.
- For all pairs of tuples t_1 and t_2 in R such that
 - If $T_1[\alpha] = T_2[\alpha]$
 - Then, $T_1[\beta] = T_2[\beta]$

X	Y	Z
1	4	2
1	4	3
2	6	3
3	2	2



Y is Determined by X

$Sid \rightarrow Sname$

$1 \rightarrow Ran + tit$

\hookrightarrow RanJit

or

Valid: Y is Determined by X.

Sid	\rightarrow Sname
1	Ranjit
2	Varun

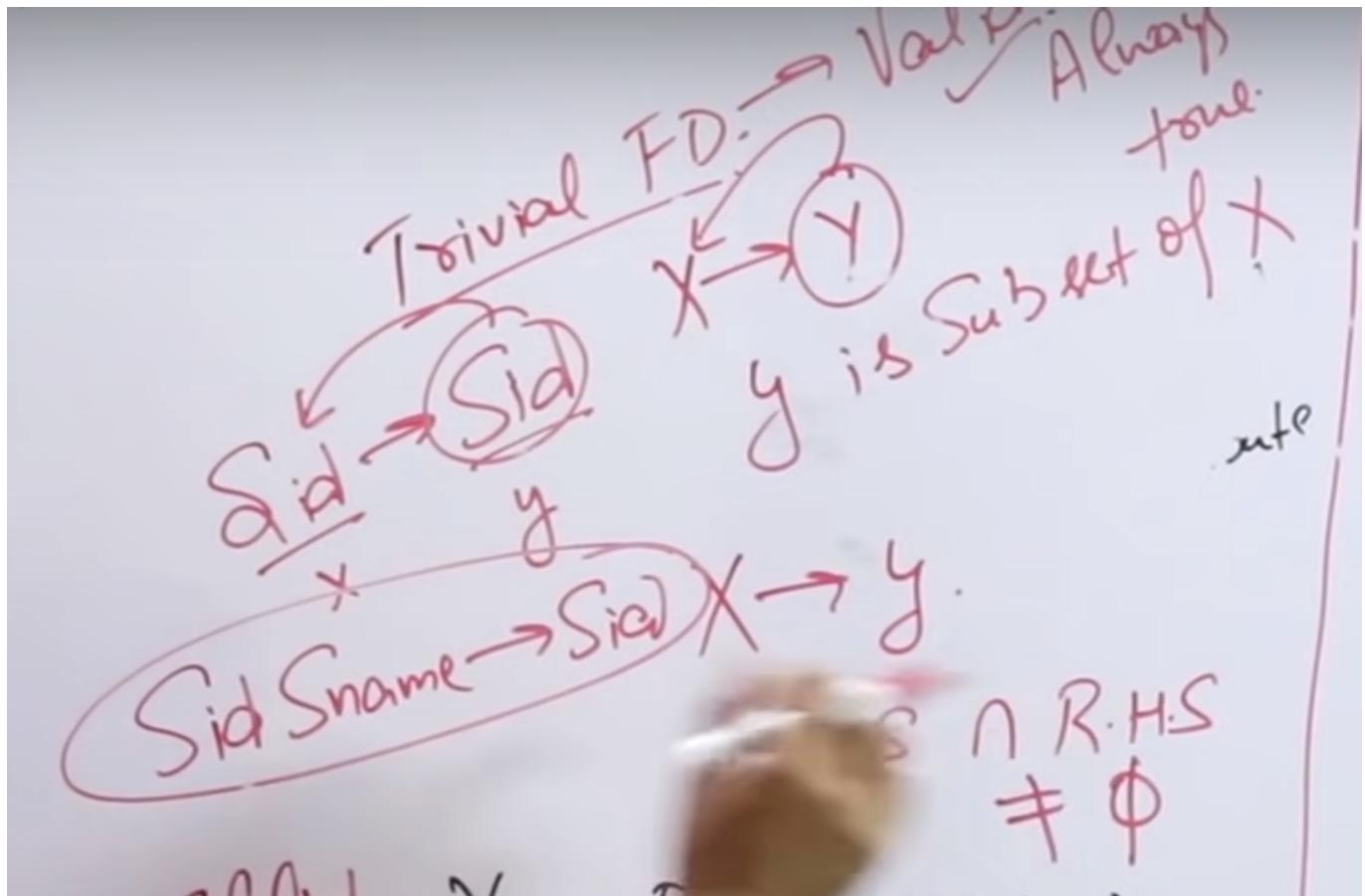
Valid:

Sid	\rightarrow Sname
1	Ranjit
1	Ranjit

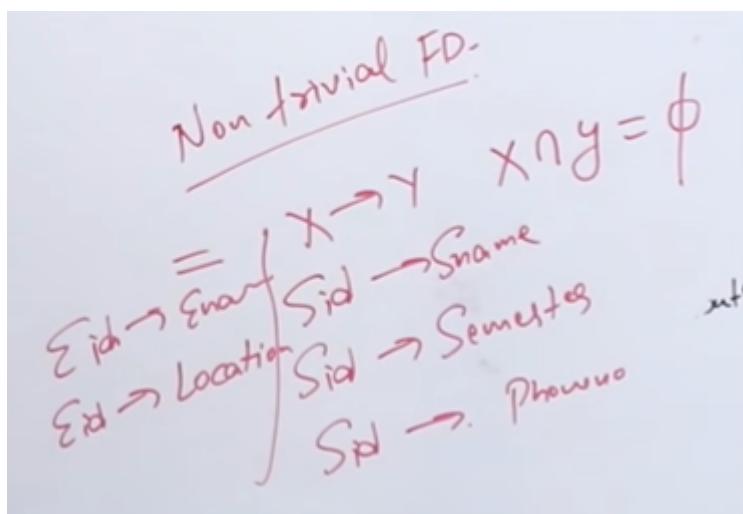
Sid	\rightarrow Sname
1	Ranjit
1	Varun

Sid	\rightarrow Sname
1	Ranjit
2	Ranjit

Trivial



Non-Trivial



Properties of FD / Armstrong axioms

- Reflexivity: If Y is a subset of X , then $X \rightarrow Y$
- Augmentation: If $X \rightarrow Y$, then $XZ \rightarrow YZ$
- Transitivity: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$ From these rules, we can derive these secondary rules-
- Union: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
- Decomposition: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
- Pseudo transitivity: If $X \rightarrow Y$ and $WY \rightarrow Z$, then $WX \rightarrow Z$

- Composition: If $X \rightarrow Y$ and $Z \rightarrow W$, then $XZ \rightarrow YW$

Reflexivity: if y is subset of X then $X \rightarrow Y$

Augmentation: if $X \rightarrow Y$, then $XZ \rightarrow YZ$

Transitive: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

Union: if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$

Decomposition: if $X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$

Pseudotransitivity: if $X \rightarrow Y$ and $WY \rightarrow Z$ then $WX \rightarrow Z$

Composition: if $X \rightarrow Y$ and $Z \rightarrow W$ then $XZ \rightarrow YW$

 SUBSCRIBE

Properties of FD:

Reflexivity: if y is subset of X then $X \rightarrow Y$ $\text{Sid} \rightarrow \text{Sid}$

Augmentation: if $X \rightarrow Y$, then $XZ \rightarrow YZ$
 $\text{Sid} \rightarrow \text{Sname}$ $\text{Sid} \rightarrow \text{Phone}$ $\text{Sname} \rightarrow \text{Sname Phone}$

Transitive: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

Union: if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$

Decomposition: if $X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$

* Transitive: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$
 $\text{Sid} \rightarrow \text{Sname}$ and $\text{Sname} \rightarrow \text{City}$
 $\text{Sid} \rightarrow \text{City}$ ✓

Union: if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$
 $X \rightarrow YZ$

Decomposition: if $X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$

~~$X \rightarrow Y \rightarrow Z$ $X \rightarrow Z, Y \rightarrow Z \rightarrow X$~~

Pseudotransitivity: if $X \rightarrow Y$ and $W \rightarrow Z$ then $W \rightarrow Z$

Composition: if $X \rightarrow Y$ and $Z \rightarrow W$ then $XZ \rightarrow YW$

Finding Closures

Q - Find CK(unique) So to get uniqueness/ If a attribute/set of attributes uniquely identifies all attributes : Closure

Closure Method

$R(A B C D E)$
 $FD = \{ A \rightarrow B, B C \rightarrow D, E \rightarrow C, D \rightarrow A \}$

→ Candidate Key

$R(A B C D)$

$FD = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D \}$

$R(A B C D)$

$FD = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D \}$

Ex 1

→ Candidate Key

$R(A B C D)$

$CK = \{ A \}$ $FD = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D \}$

$A^+ = B C D A$ $(A B)^+ = A B C D$
 $B^+ = B C D$ $C^+ = C D$
 $D^+ = D$

$AB = CKX$
 $X = \text{porky}$

Ex 2

$R(A B C D)$

$FD = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A \}$

$A^+ = ABCD$ $CR = \{ A, B, C, D \}$

$B^+ = BCDA$

$C^+ = CDAB$

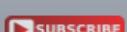
$D^+ = DABC$

Prime Attribute

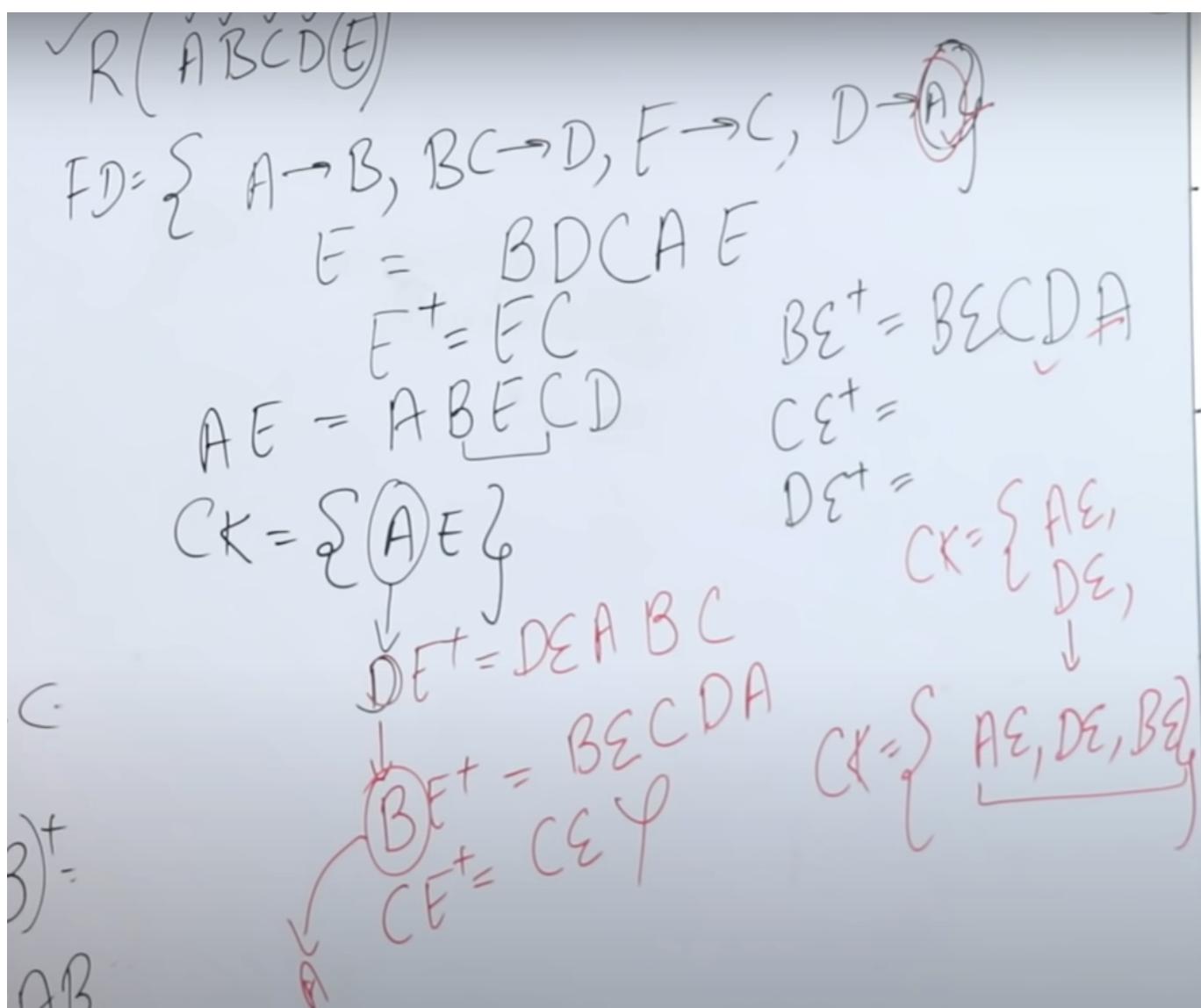
= $\{ A, B, C, D \}$

Non Prime Attribute

= $\{ \}$



Ex 3



GATE Q

GATE Question: Consider the relation scheme $R = \{E, F, G, H, I, J, K, L, M, N\}$ and the set of functional dependencies $\{E, F\} \rightarrow \{G\}, \{F\} \rightarrow \{I, J\}, \{E, H\} \rightarrow \{K, L\}, K \rightarrow \{M\}, L \rightarrow \{N\}$ on R . What is the key for R ? (GATE-CS-2014)

- A. $\{E, F\}$
- B. $\{E, F, H\}$
- C. $\{E, F, H, K, L\}$
- D. $\{E\}$

 $EFH \rightarrow EFGHIJKLMNOP$ $EF \rightarrow G$ $F \rightarrow IJ$ $EH \rightarrow KL$ $K \rightarrow M$ $L \rightarrow N$

$\{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

Which of the following functional dependencies is NOT implied by the above set? (GATE IT 2005)

- | | |
|------------------------|--------------------------|
| A. $CD \rightarrow AC$ | $CD^+ \rightarrow CDEAB$ |
| B. $BD \rightarrow CD$ | $BD^+ \rightarrow BD$ |
| C. $BC \rightarrow CD$ | $BC^+ \rightarrow BCDEA$ |
| D. $AC \rightarrow BC$ | $AC^+ \rightarrow ACBDE$ |

GATE Question: Consider a relation scheme $R = (A, B, C, D, E, H)$ on which the following functional dependencies hold: $\{A \rightarrow B, BC \rightarrow D, E \rightarrow C, D \rightarrow A\}$.

What are the candidate keys of R? [GATE 2005]

- | | |
|-------------------|------------------|
| (a) AE, BE | |
| (b) AE, BE, DE | |
| (c) AEH, BEH, BCH | $EH \rightarrow$ |
| (d) AEH, BEH, DEH | ✓ |

Equivalence of Two FD sets

- Two FD sets F_1 and F_2 are equivalent if – $F_1^+ = F_2^+$ Or $F_1 \sqsubseteq F_2^+$ and $F_2 \sqsubseteq F_1^+$

Steps :

- Find F's left hand side attribute's closure using G Set
- Check F's FD with the closure you got
- Find G's left hand side attribute's closure using F Set
- Check G's FD with the closure you got

5. If you can get all the FD with the closures, they are said to be equivalent.

F	G
$A \rightarrow C$	$A \rightarrow CD$
$AC \rightarrow D$	
$E \rightarrow AD$	$E \rightarrow AH$
$E \rightarrow H$	

F	G
$A \rightarrow C$	$A \rightarrow CD$
$AC \rightarrow D$	
$E \rightarrow AD$	$E \rightarrow AH$
$E \rightarrow H$	

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

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

Minimal Cover / Canonical Cover

For the following Functional dependencies, find the correct Minimal Cover

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

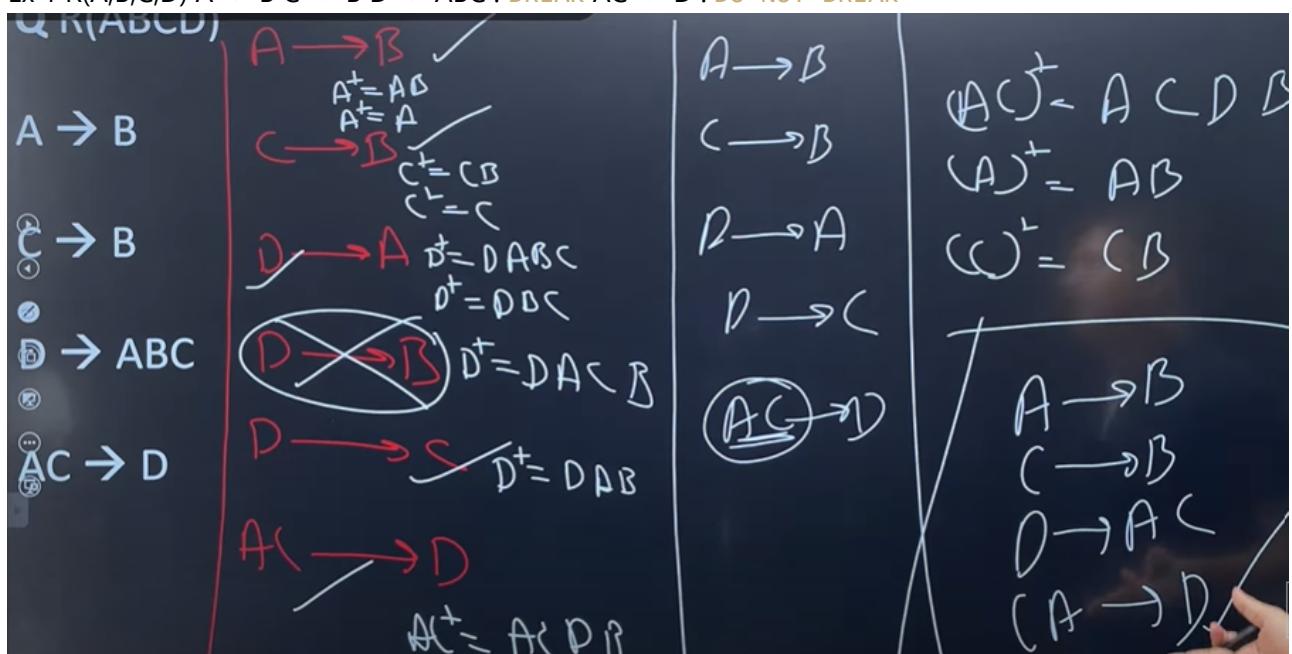
- a) $A \rightarrow B, C \rightarrow B, D \rightarrow A, AC \rightarrow D$
- b) $A \rightarrow B, C \rightarrow B, D \rightarrow C, AC \rightarrow D$
- c) $A \rightarrow BC, D \rightarrow CA, AC \rightarrow D$
- d) $A \rightarrow B, C \rightarrow B, D \rightarrow AC, AC \rightarrow D$

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

To find the MINIMAL COVER / CANONICAL COVER / IRREDUCIBLE SET

- A **canonical cover** (also known as a minimal cover) for a set of functional dependencies in a database is a minimal set of functional dependencies that is equivalent to the original set, but with redundant dependencies and extraneous attributes removed. It is used in the normalization process of database design to simplify the set of functional dependencies and to find a good set of relations.
- There may be any following type of redundancy in the set of functional dependencies: -
 - Complete production may be Redundant.
 - One or more than one attributes may be redundant on right hand side of a production.
 - One or more than one attributes may be redundant on Left hand side of a production.

- Ex 1 R(A,B,C,D) $A \rightarrow B$ $C \rightarrow B$ $D \rightarrow ABC$: BREAK $AC \rightarrow D$: DO NOT BREAK



Normalization

'Normalization'

→ it is a technique to Remove or Reduce

Redundancy from a table.

SID	Sname	Age
1	RAM	20
2	Vasun	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

Deletion

Anomaly

where SID=2

Update

Anomaly

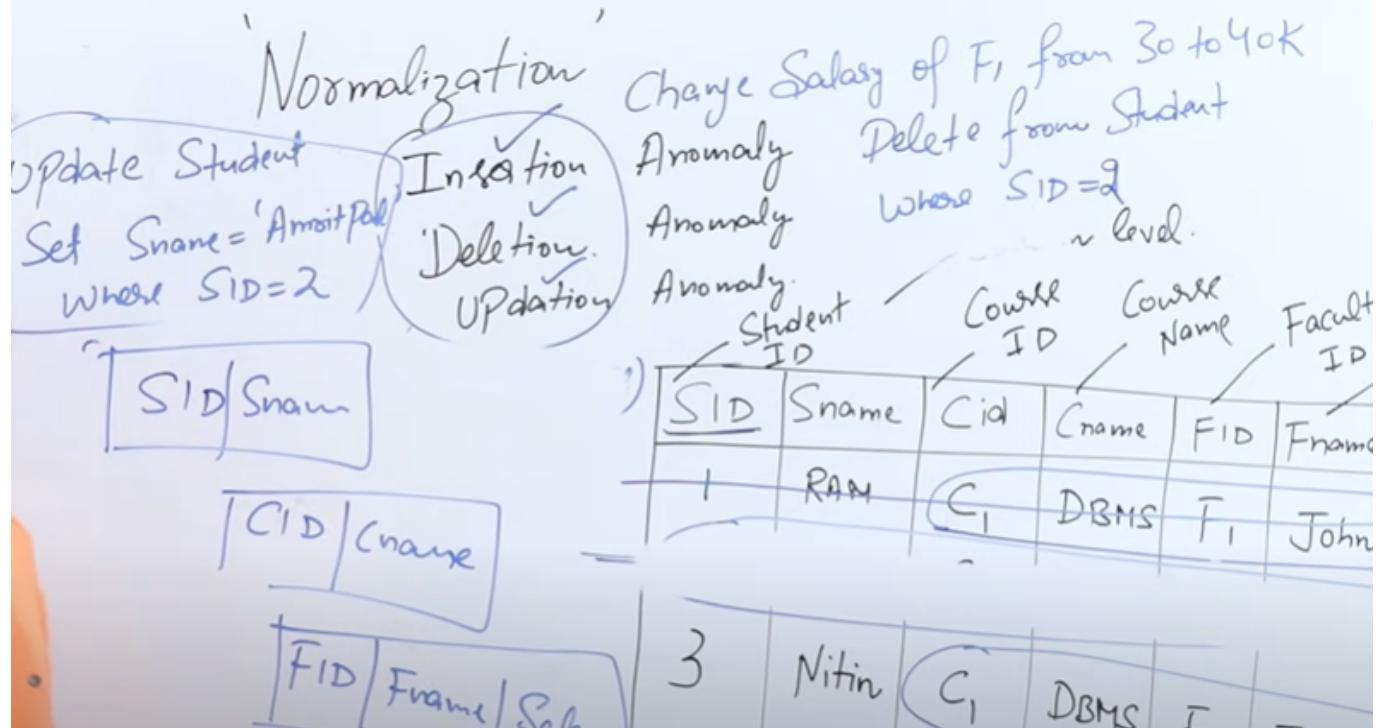
at Row level.

Primary Key (Unique + Not Null)

SID	Sname	Age
1	RAM	20
2	Vasun	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	Course
1	Sai	C/C++
2	Harsh	Java
3	Onkar	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
1	Sai	C	C++
2	Harsh	Java	Null
3	Onkar	C	DBMS

Course1

Null Null

3

Any

foreign Key

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

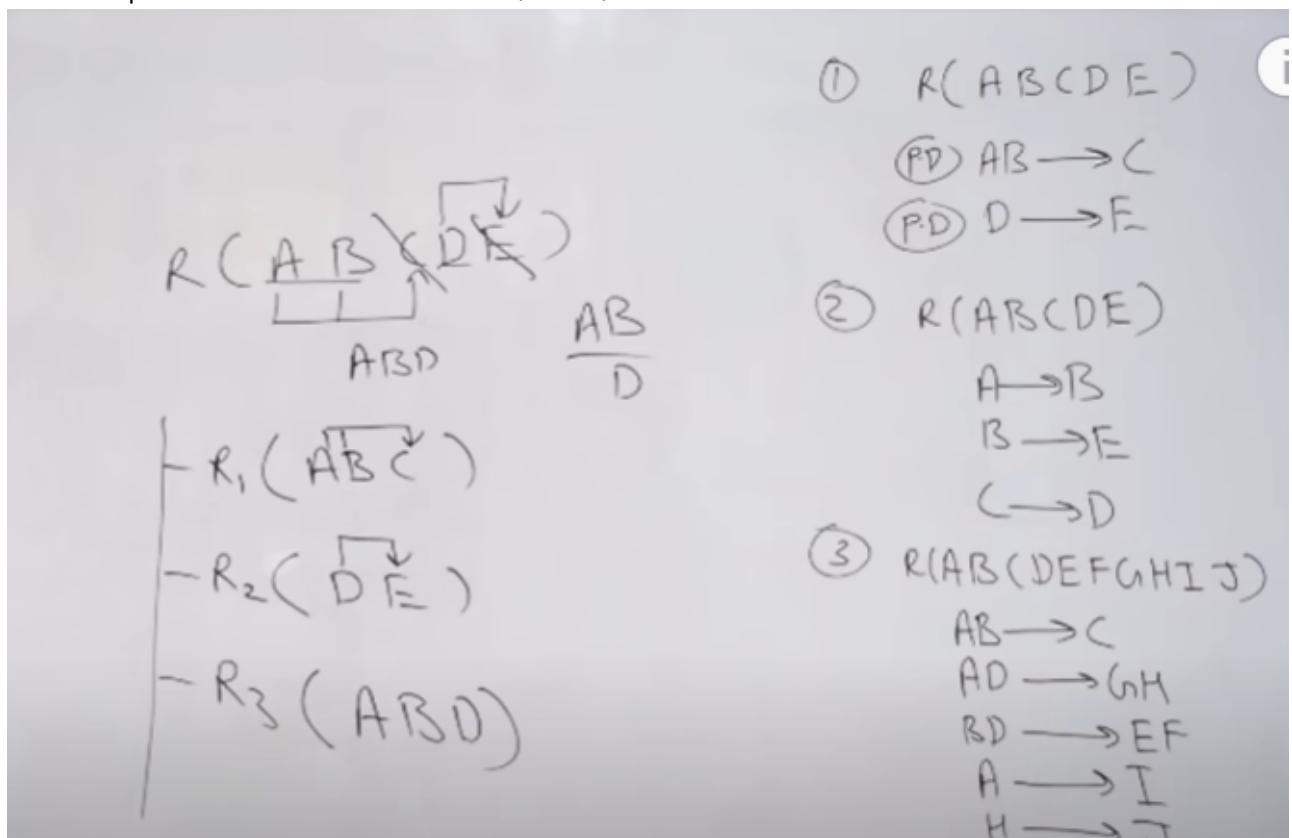
A	B
A	1
B	2
A	3
C	3
D	3
E	3

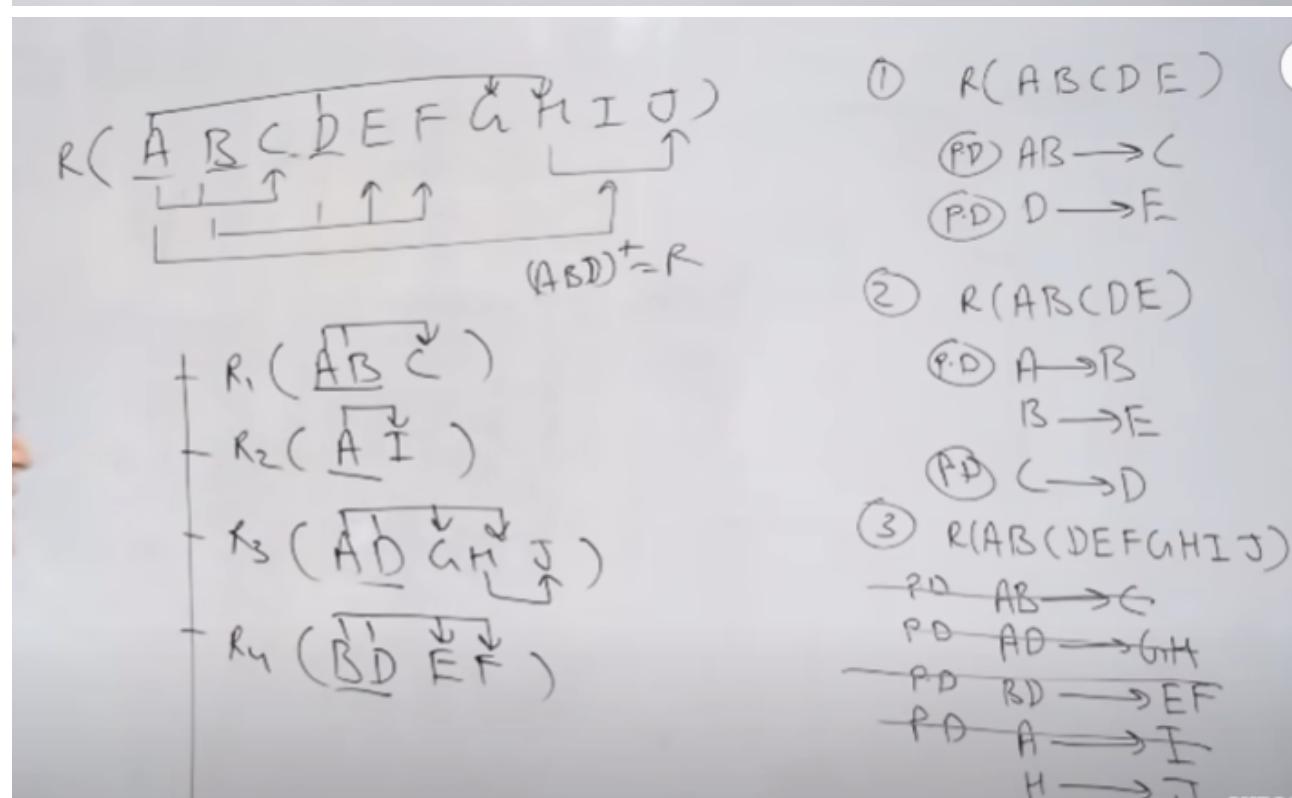
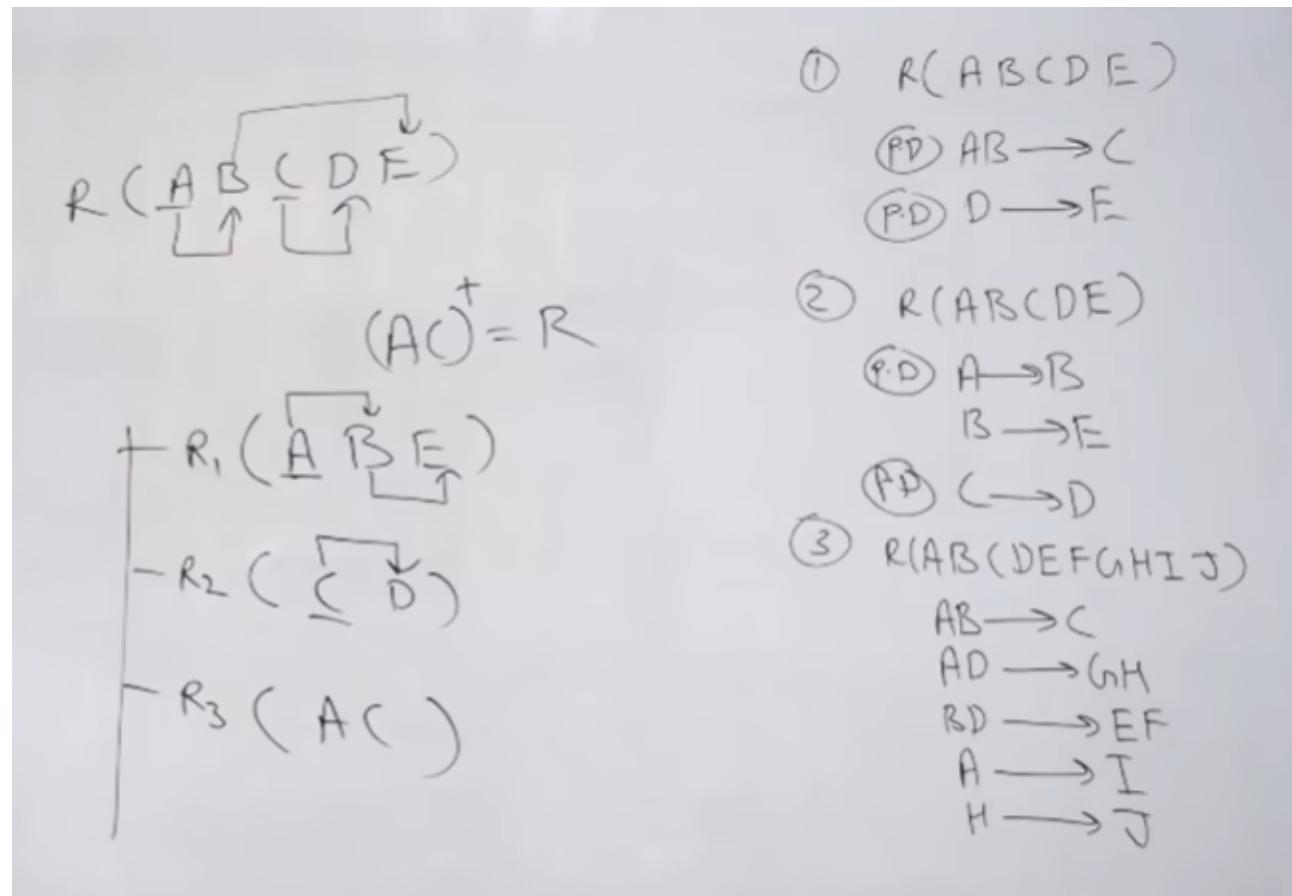
B	C
1	X
2	Y
3	z

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

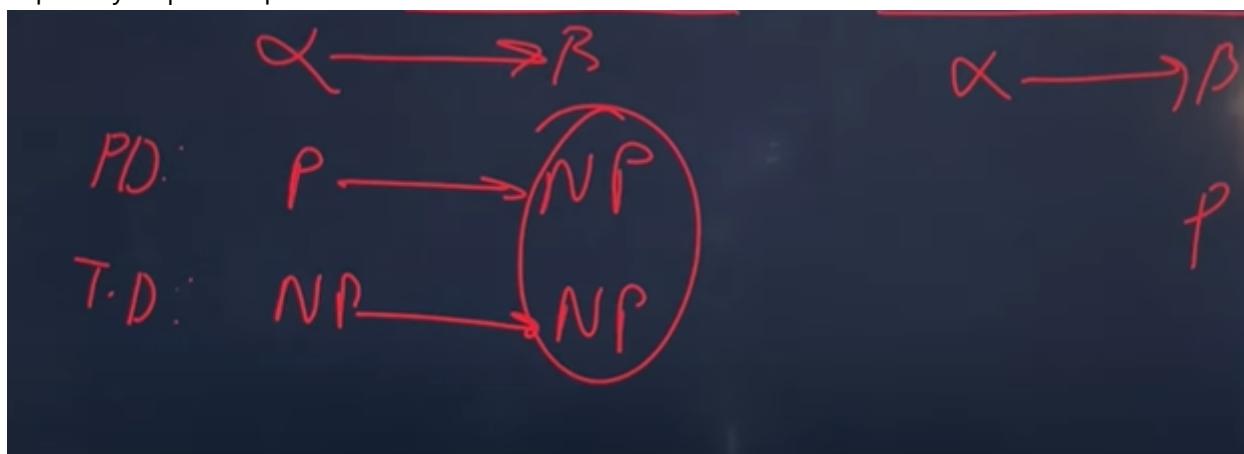




3NF

- 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

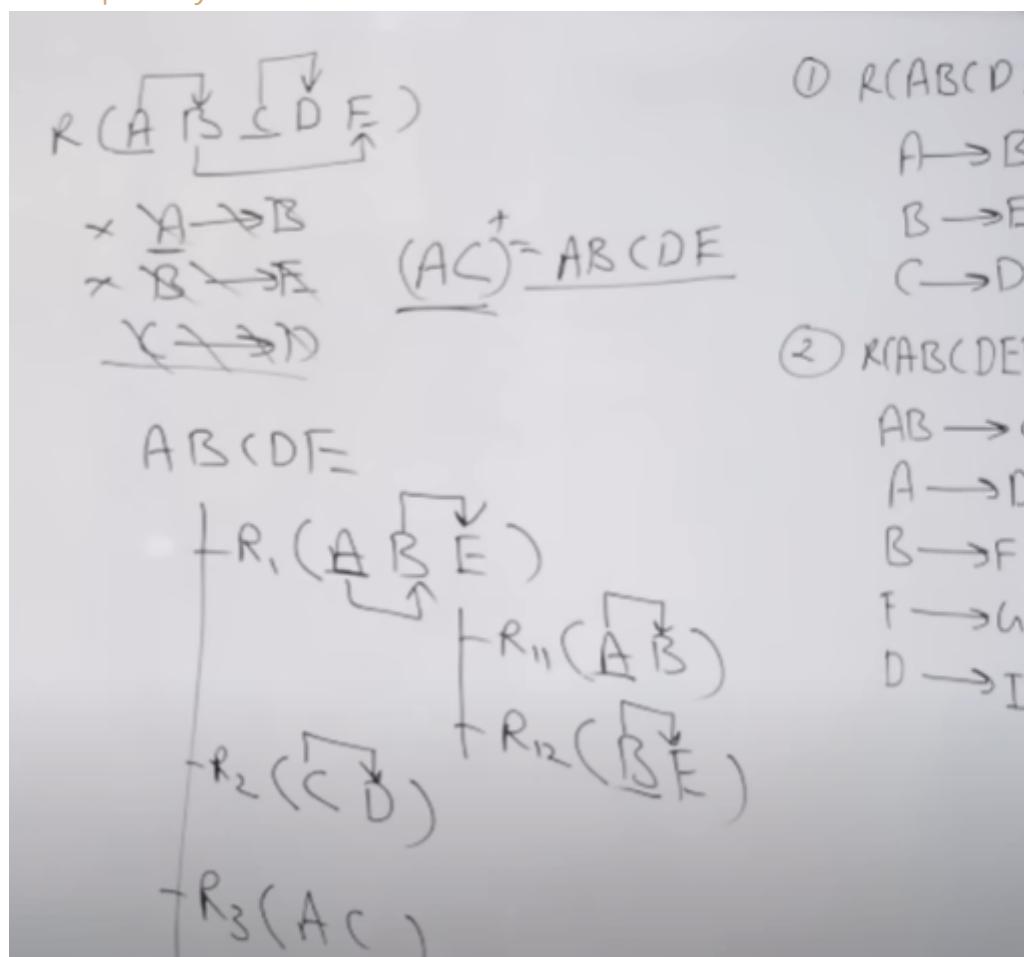


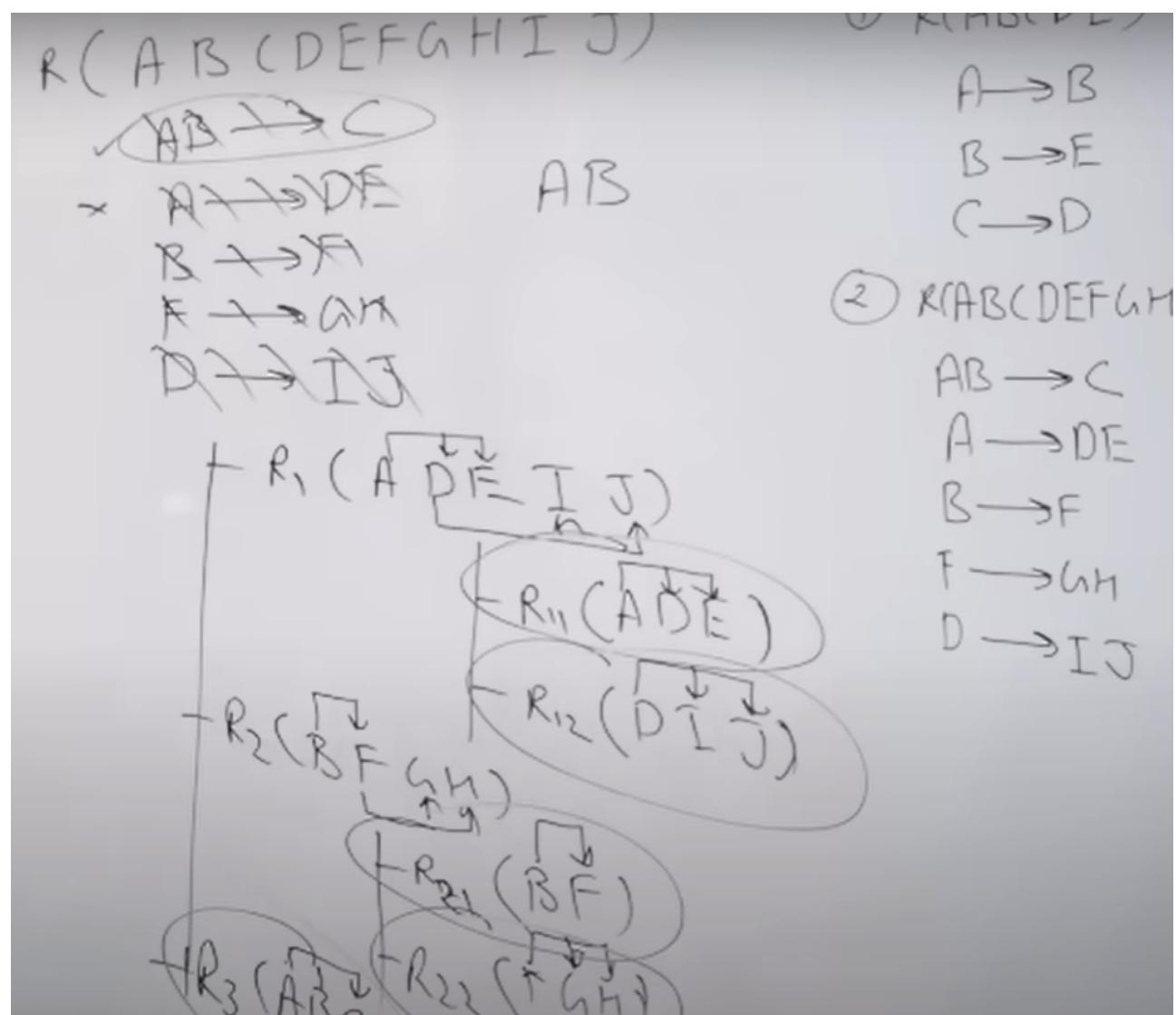
- NPA \rightarrow NPA NOT ALLOWED

$R(A, B, C)$																										
$A \rightarrow B$	$B \rightarrow C$																									
		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>1</td> <td>P</td> </tr> <tr> <td>B</td> <td>2</td> <td>Q</td> </tr> <tr> <td>C</td> <td>2</td> <td>Q</td> </tr> <tr> <td>D</td> <td>2</td> <td>Q</td> </tr> <tr> <td>E</td> <td>3</td> <td>R</td> </tr> <tr> <td>F</td> <td>3</td> <td>R</td> </tr> <tr> <td>G</td> <td>4</td> <td>S</td> </tr> </tbody> </table>	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	C																								
A	1	P																								
B	2	Q																								
C	2	Q																								
D	2	Q																								
E	3	R																								
F	3	R																								
G	4	S																								
		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>1</td> </tr> <tr> <td>B</td> <td>2</td> </tr> <tr> <td>C</td> <td>2</td> </tr> <tr> <td>D</td> <td>2</td> </tr> <tr> <td>E</td> <td>3</td> </tr> <tr> <td>F</td> <td>3</td> </tr> <tr> <td>G</td> <td>4</td> </tr> </tbody> </table>	A	B	A	1	B	2	C	2	D	2	E	3	F	3	G	4								
A	B																									
A	1																									
B	2																									
C	2																									
D	2																									
E	3																									
F	3																									
G	4																									
		<table border="1"> <thead> <tr> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>P</td> </tr> <tr> <td>2</td> <td>Q</td> </tr> <tr> <td>3</td> <td>R</td> </tr> <tr> <td>4</td> <td>S</td> </tr> </tbody> </table>	B	C	1	P	2	Q	3	R	4	S														
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

$\wedge B \hookrightarrow D$

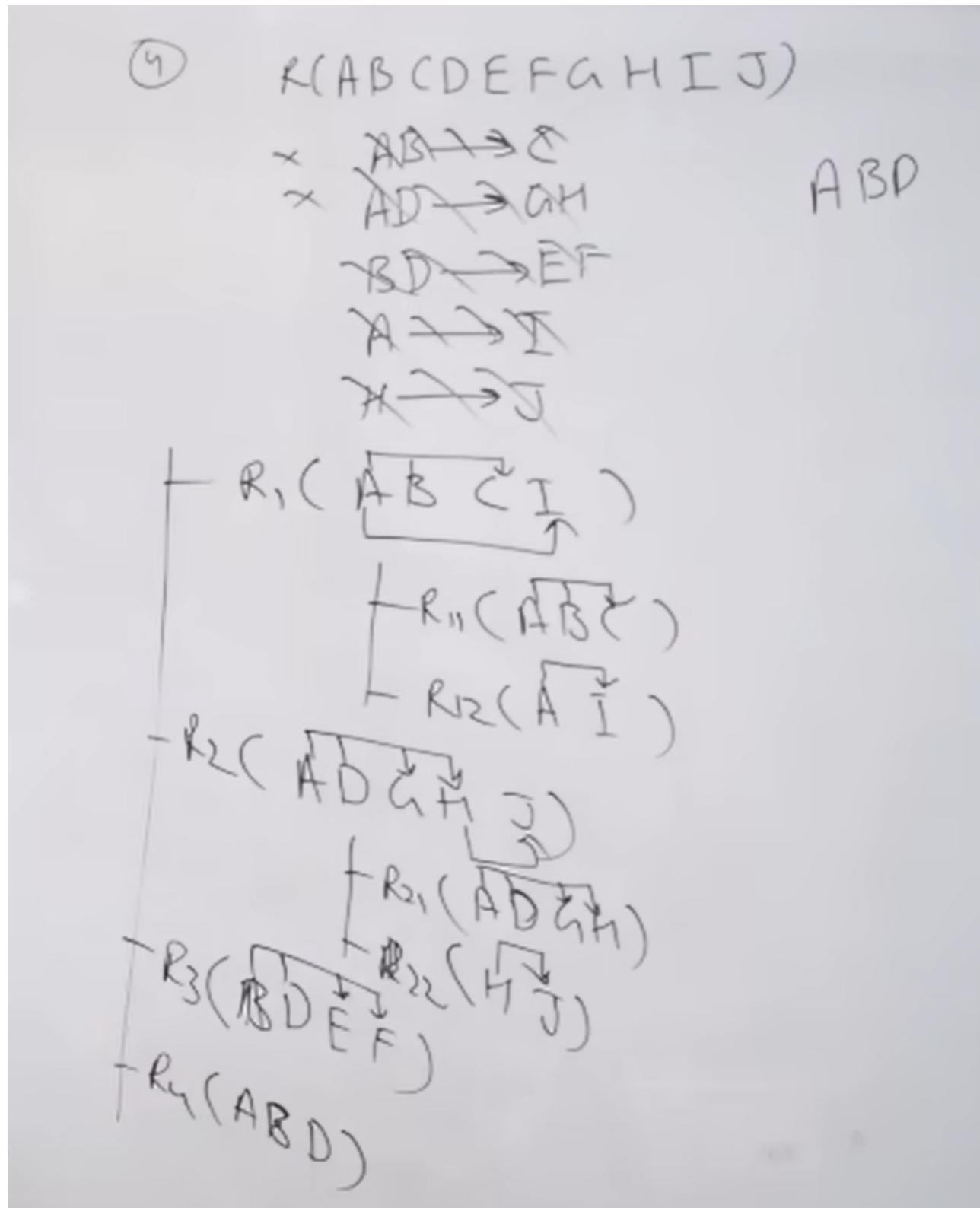
$D \hookrightarrow E$

$+ R_1(B \hookrightarrow E)$

$+ R_{11}(B \hookrightarrow D)$

$+ R_{12}(D \hookrightarrow E)$

$+ R_2(\underline{AB}^C)$



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	Vasun	M034	21
3	Ravi	K786	23
4	Rahul	D286	21

CK: { Rollno, Voterid }

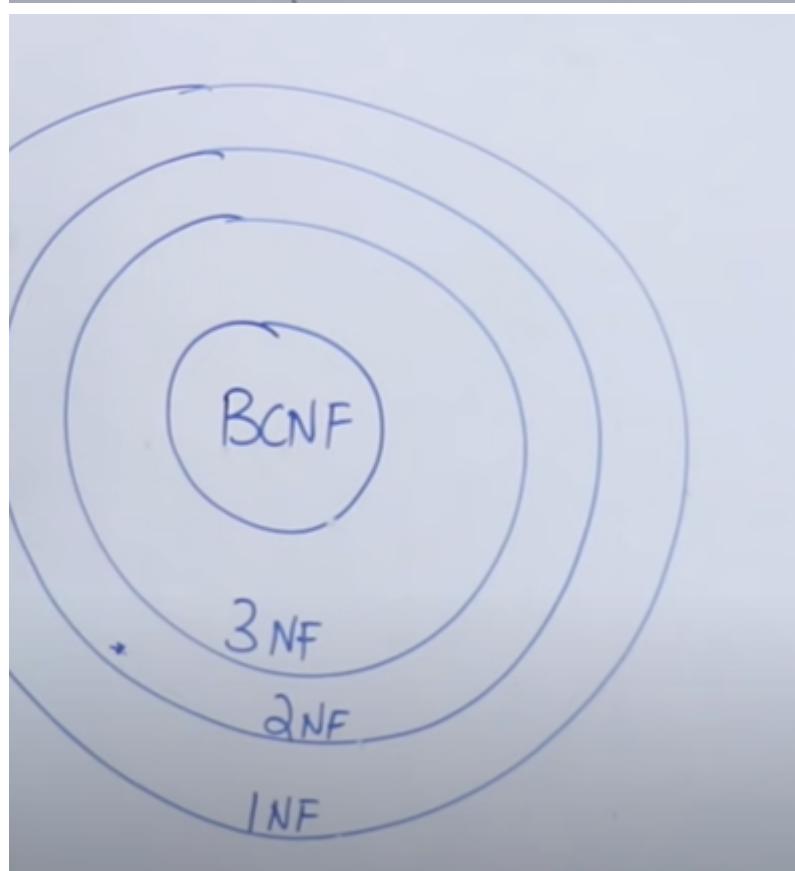
FD: {
 $\text{Rollno} \rightarrow \text{name}$
 $\text{Rollno} \rightarrow \text{Voterid}$
 $\text{Voterid} \rightarrow \text{age}$
 $\text{Voterid} \rightarrow \text{Rollno}$ }

L.H.S of each FD should be K.
 Candidate Key or Super Key.

CK: { Rollno, Voterid }

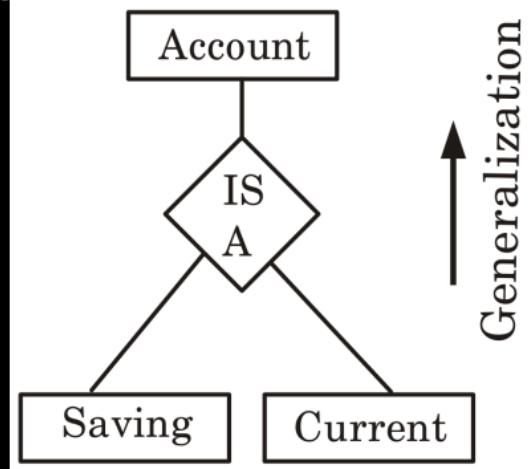
FD:

\checkmark $\text{Rollno} \rightarrow \text{Name}$ $\text{Voterid} \rightarrow \text{Voterid}$ $\text{Voterid} \rightarrow \text{age}$ $\text{Voterid} \rightarrow \text{Rollno}$	$\xrightarrow{\text{1NF}}$ FD
---	---



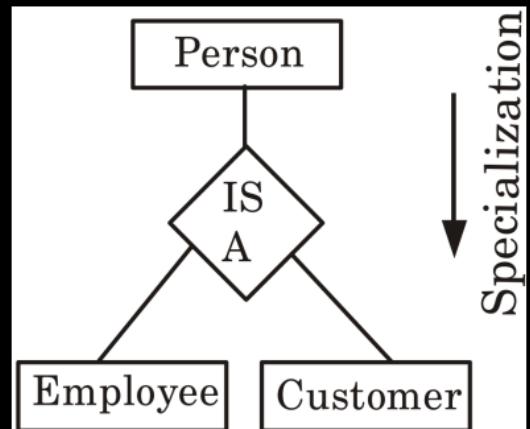
Generalization

- 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.



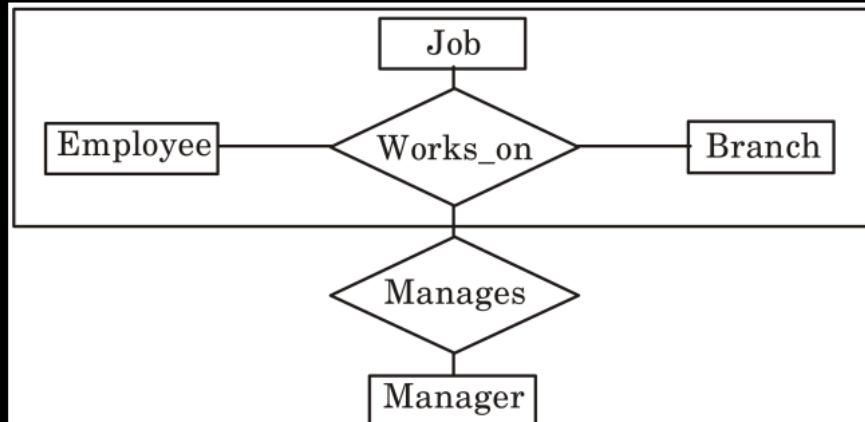
Specialization

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

Step 1: Find all CKs in Relation

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

Step 2: Write all Prime Attributes.

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

Step 3: Write all Non-Prime Attrib.

$\{D\}$

$$AB^+ = ABCDEF$$

$$FB^+ = FBACDE$$

$$EB^+ = EBFACD$$

$$CB^+ = CBDEFA$$

$$A^+ = A$$

$$B^+ = B$$

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. **Loseless 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
-

Q : Convert to BCNF

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

CK : ABD

Not in 2F : bcoz of PD

Decomposition

R1(ABC) : AB- \rightarrow C

R2(DE) : D- \rightarrow E

R3(ABD)

2nf

CK - AB

AB \rightarrow C,

A \rightarrow DE, : Partial Dependency B \rightarrow F, : Partial Dependency F \rightarrow GH, D \rightarrow IJ

R1(ADEIJ) : A A- \rightarrow DE D- \rightarrow IJ

R2(BFGH) : B B- \rightarrow F F- \rightarrow GH

R3(ABC) : AB AB- \rightarrow C

3NF

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

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

R3(ABC) : AB AB- \rightarrow C

Final Decomposition :

R11(DIJ) : D D- \rightarrow 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

Projecting FD

1. FD = {A->A, A->B, A->C, A->D} Project on {ABD}

$A^* = ABCD$: A-> B, A->D $B^* = B$ $D^* = DC$ $BD^* = BD$

$T(ABD) = \{A \rightarrow A, A \rightarrow B, A \rightarrow D\}$

2. FD = {A->B, BC->D, D->C} Project on {BCD}

$B^* = B$ $C^* = C$ $D^* = DC$: D->C $BC^* = BCD$: BC->D $BD^* = BCD$: BD->C $BCD^* = BCD$

$T(BCD) = \{D \rightarrow C, BC \rightarrow D, BD \rightarrow C\}$

HW

5. Projecting a set of FDs onto a subset of the attributes.

Suppose we have a relation on attributes $ABCDE$ with these FDs:

$$A \rightarrow C, C \rightarrow E, E \rightarrow BD$$

(a) Project the FDs onto attributes ABC .

Solution:

To project onto a set of attributes, we systematically consider every possible LHS of an FD that might hold on those attributes.

- $A^+ = ACEBD$, therefore $A \rightarrow BC$. (It also functionally determines DE , but these are not in our set of attributes. And it functionally determines itself, but we don't need to write down dependencies that are tautologies.)
- $B^+ = B$. This yields no FDs for our set of attributes.
- $C^+ = CEBD$, therefore $C \rightarrow B$.
- We don't need to consider any supersets of A . A already determines all of our attributes ABC , so supersets of A will only yield FDs that already follow from $A \rightarrow BC$.
- The only remaining subset of the attributes ABC to consider is BC . $BC^+ = BCED$. This yields no FDs for our set of attributes.
- So the projection of the FDs onto ABC is: $\{A \rightarrow BC, C \rightarrow B\}$.

(b) Project the FDs onto attributes ADE .

- $A^+ = ACEBD$, therefore $A \rightarrow DE$.
 - $D^+ = D$. This yields no non-trivial FDs.
 - $E^+ = EBD$, therefore $E \rightarrow D$.
 - Again, we don't need to consider any supersets of A, since A determines all the attributes ADE also.
 - The only remaining subset of the attributes ABC to consider is DE . $DE^+ = DEB$. This yields no FDs for our set of attributes.
 - So the projection of the FDs onto ADE is: $\{A \rightarrow DE, E \rightarrow D\}$.
-