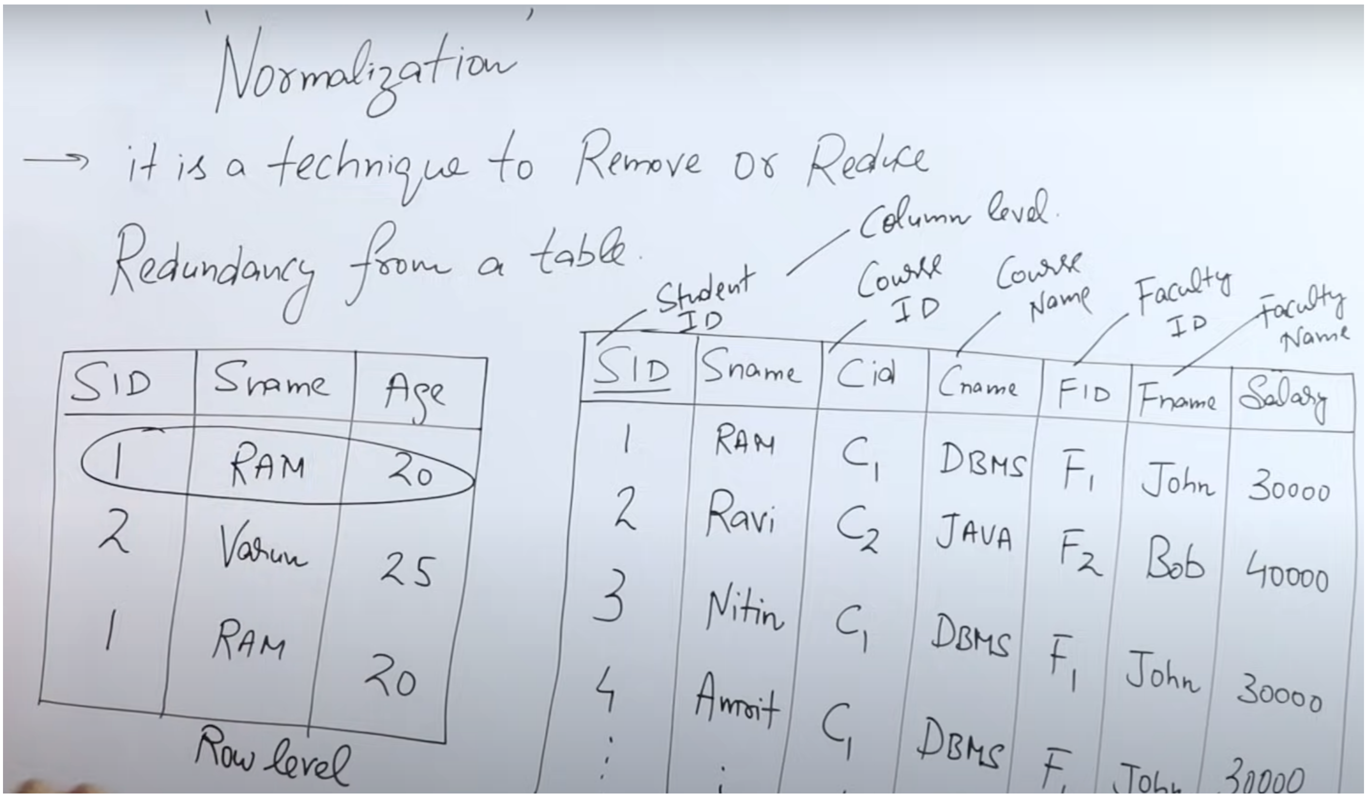
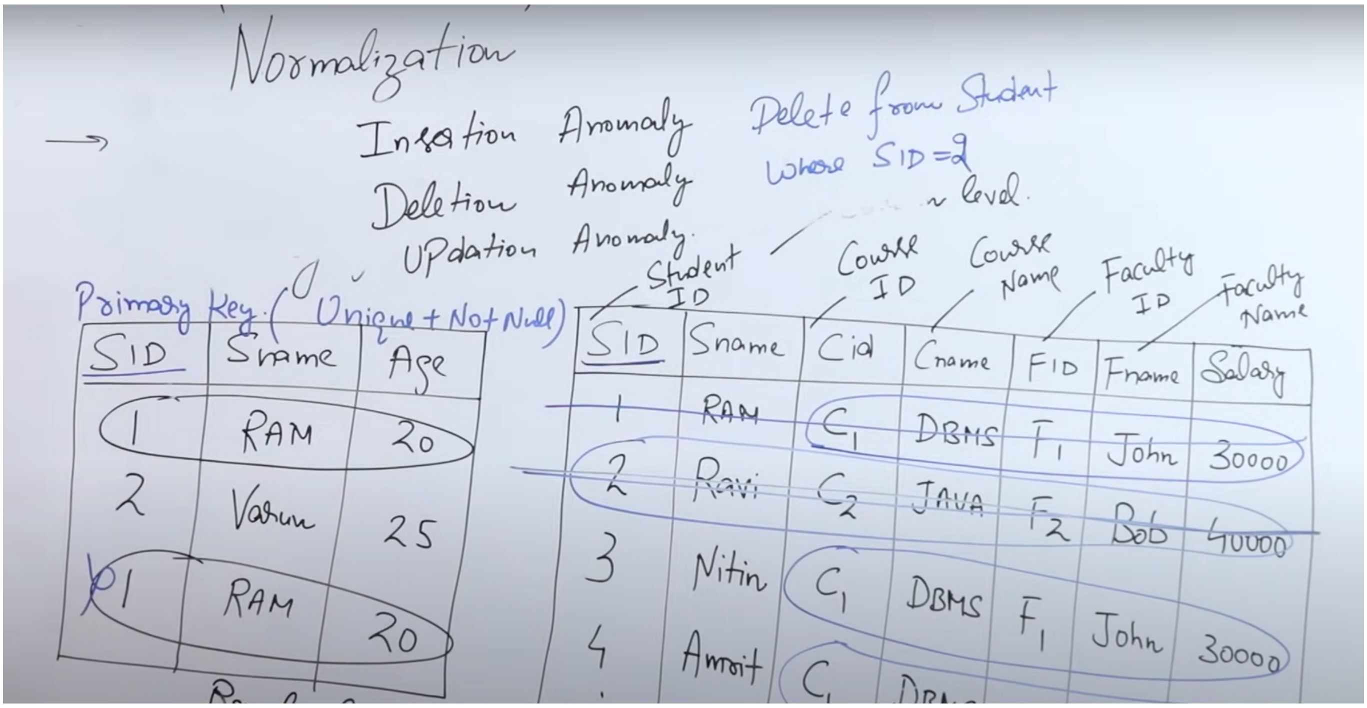
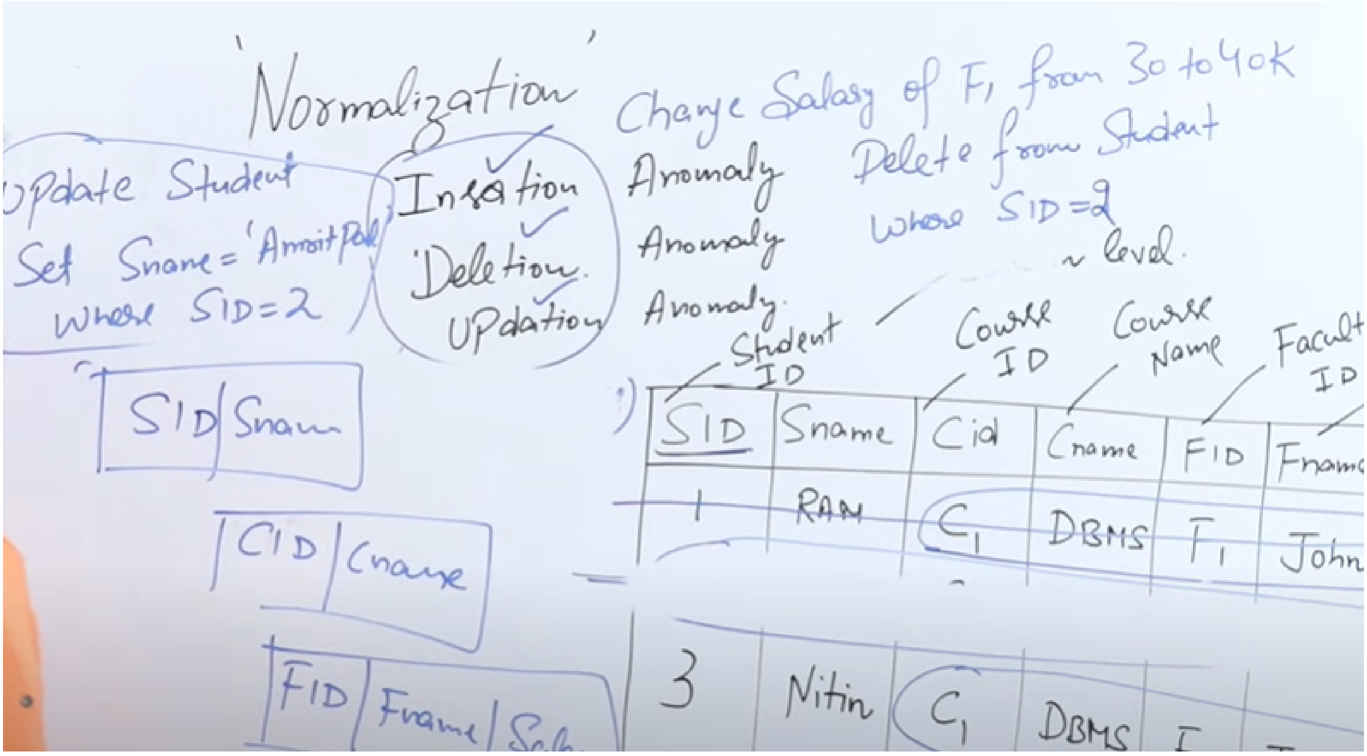
Normalization



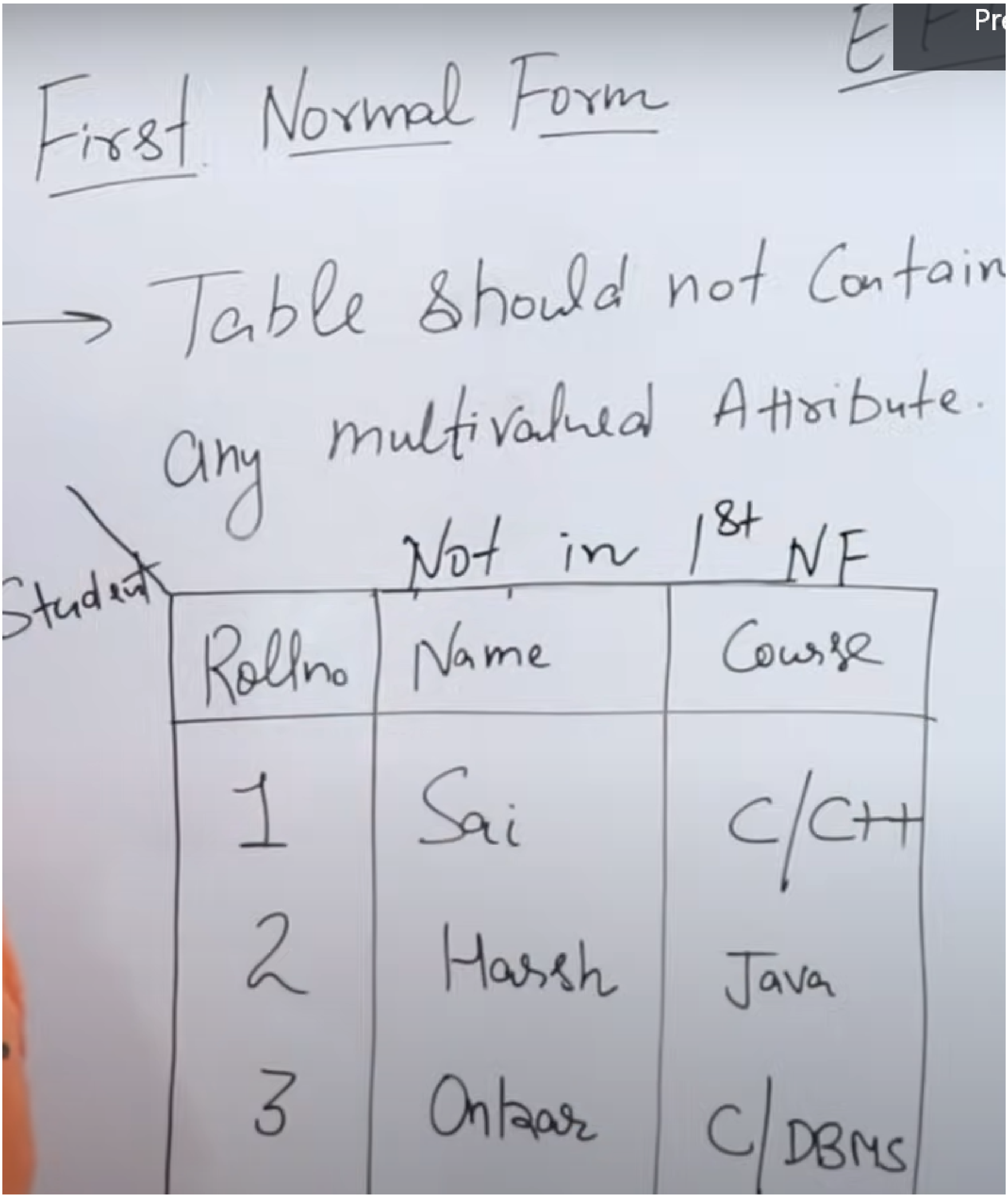


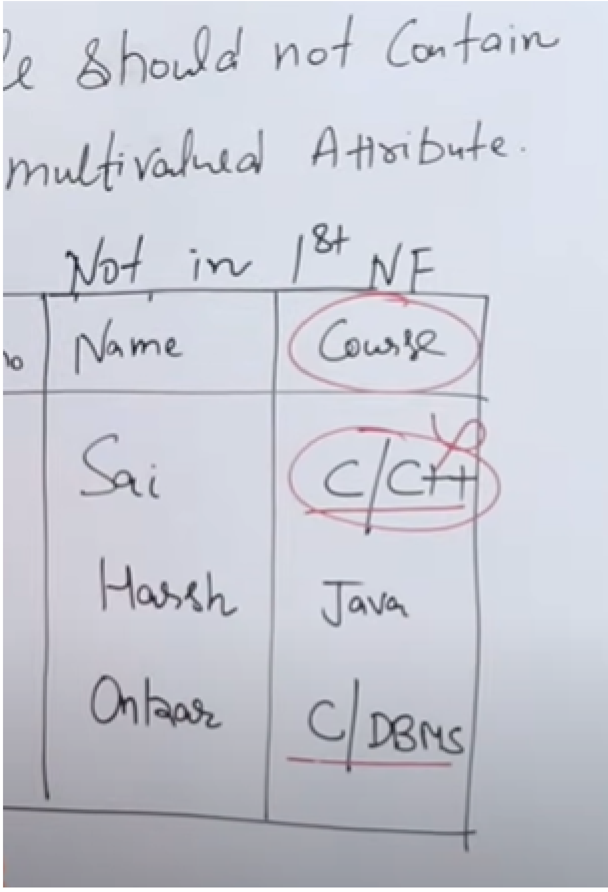


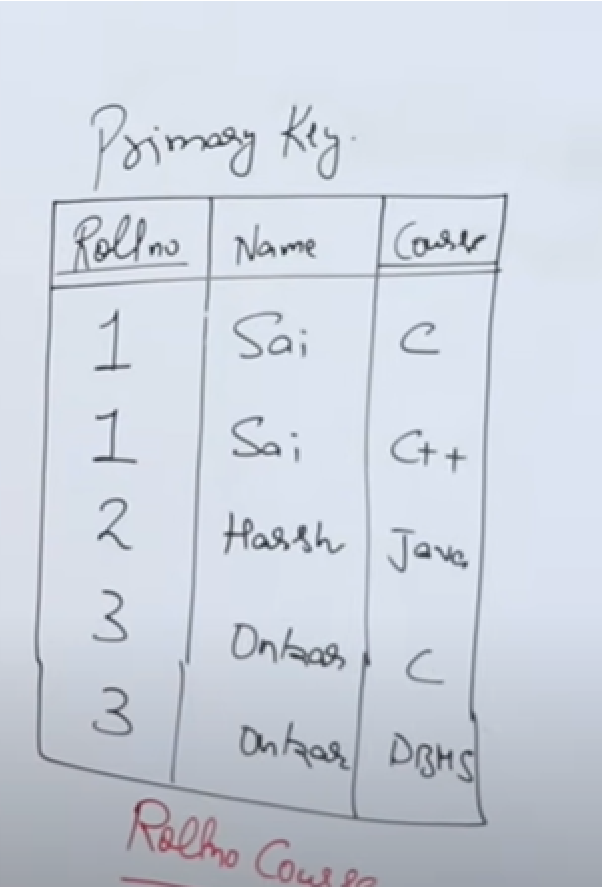
- `Atomic Values`: Each cell in a table contains indivisible, atomic values. Means a Relation should not contain any multivalued or composite attributes.
- `Unique Columns`: Each column must have a distinct name to identify the data it contains.
- `Primary Key`: A table in 1NF should have a primary key 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->B B->C [transitive dependency] C->D [transitive dependency
- `Multivalued Dependency`
- Denoted by, A ->-> B, Means, for every value of A, there may exist more than one value of B.
- E.g. S_name -> -> Club_name S_Name Club_name Kamesh Dance Kamesh Guitar

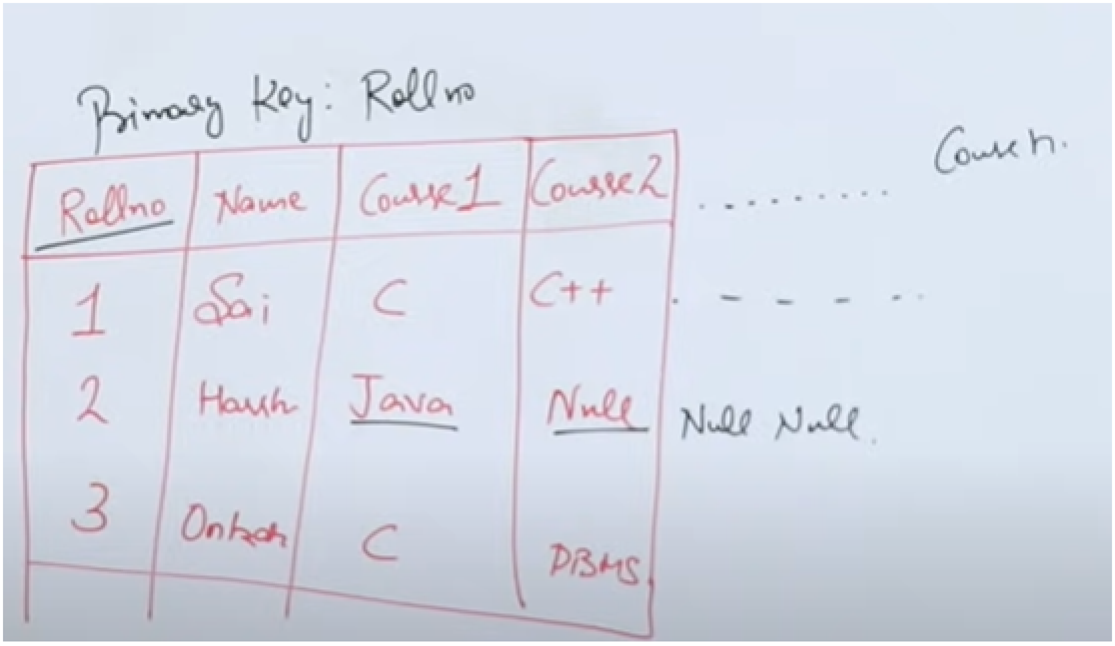
1NF

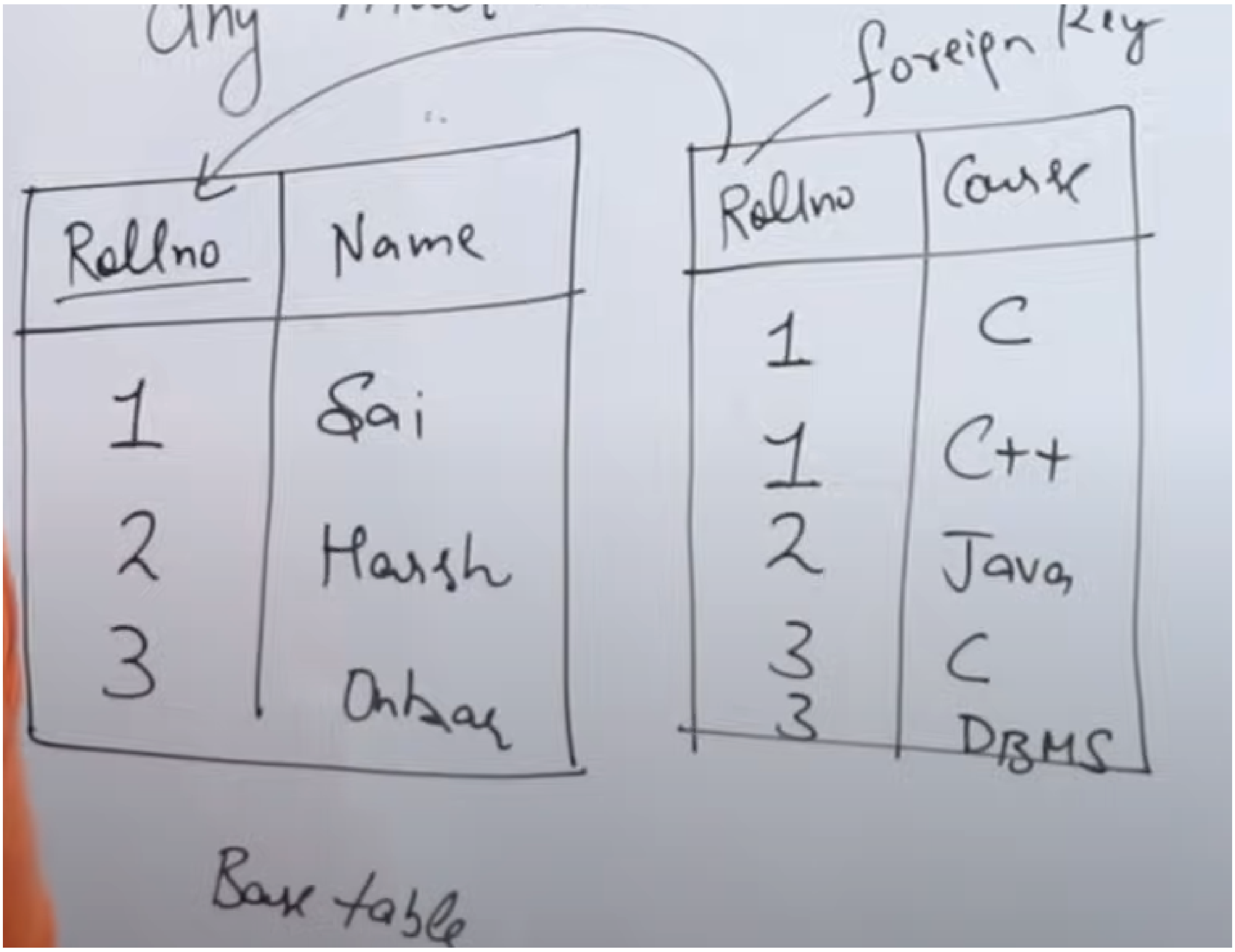
- Should not have multivalued attributes

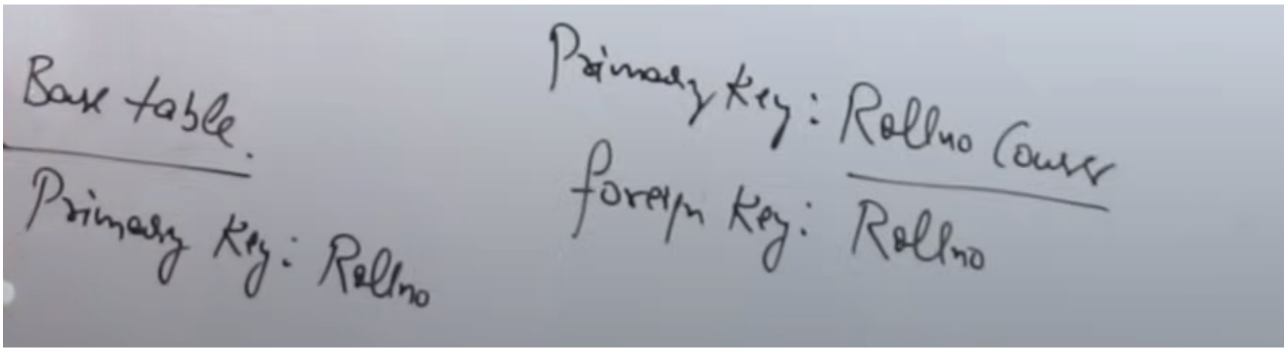



Should not Contain
multivalued Attribute.
Not in 1st NF
Name
Course
Sai
C/C++
Harsh
Java
Onkar
C/DBMS
1


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

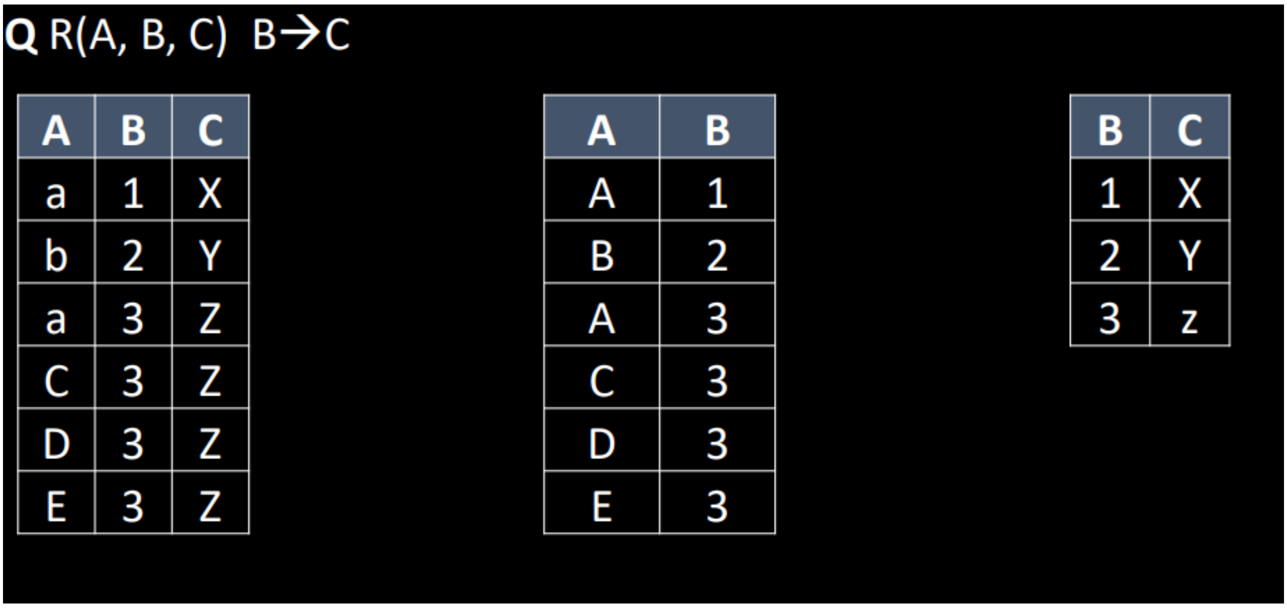

Primary Key: Rollno
Rollno
Name
Course1
Course2
Course n.
1
Sai
C
C++
2
Harsh
Java
Null
Null Null.
3
Onkar
C
DBMS
3





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→C

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

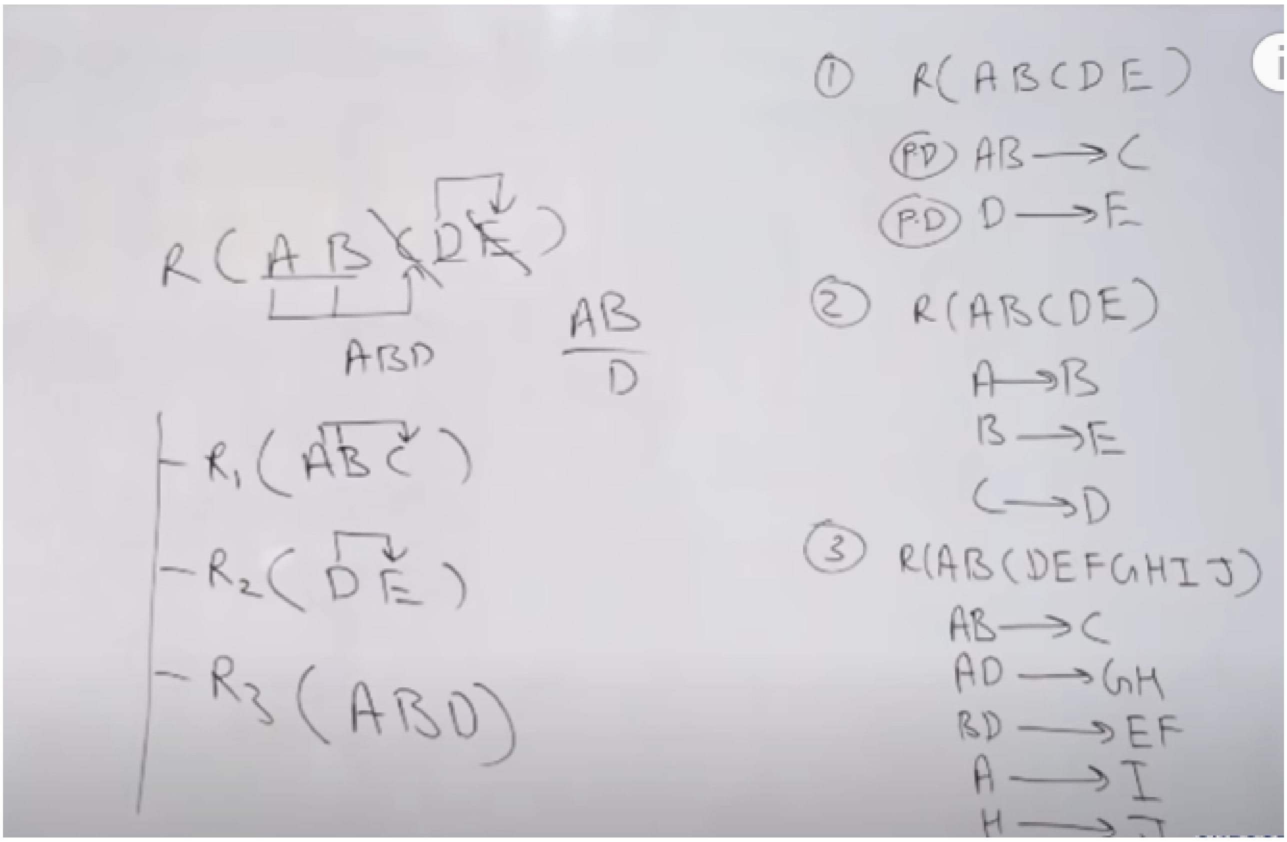
A	B
A	1
B	2
A	3
C	3
D	3
E	3

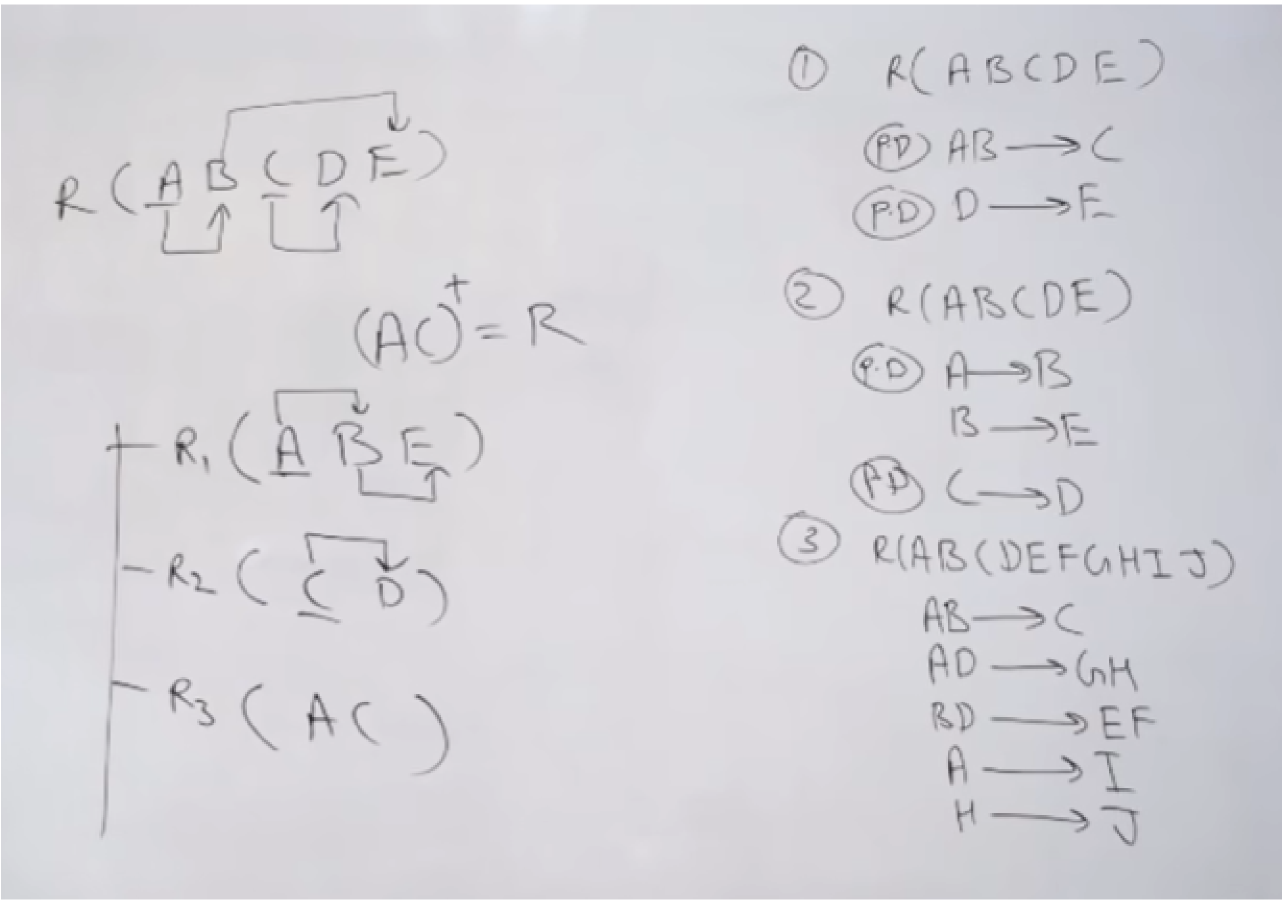
B	C
1	X
2	Y
3	z

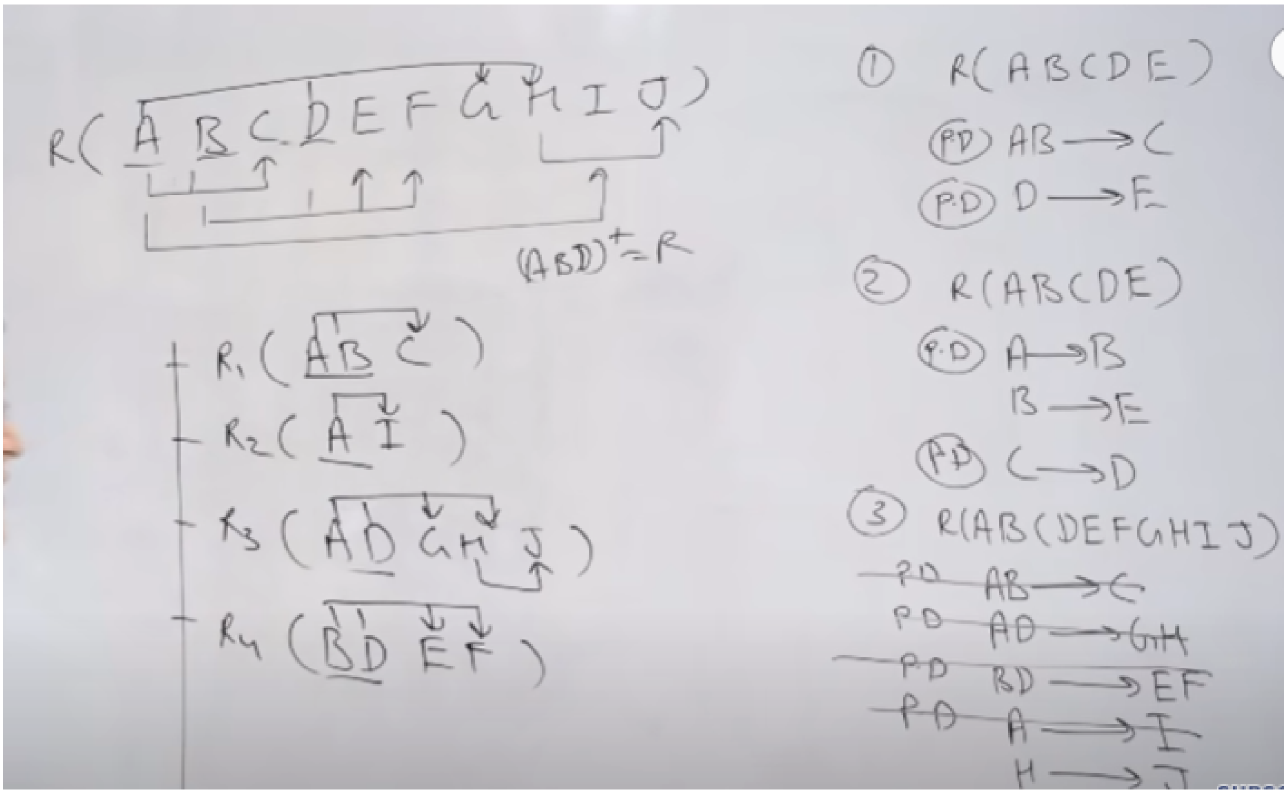
Decompose to 2NF

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



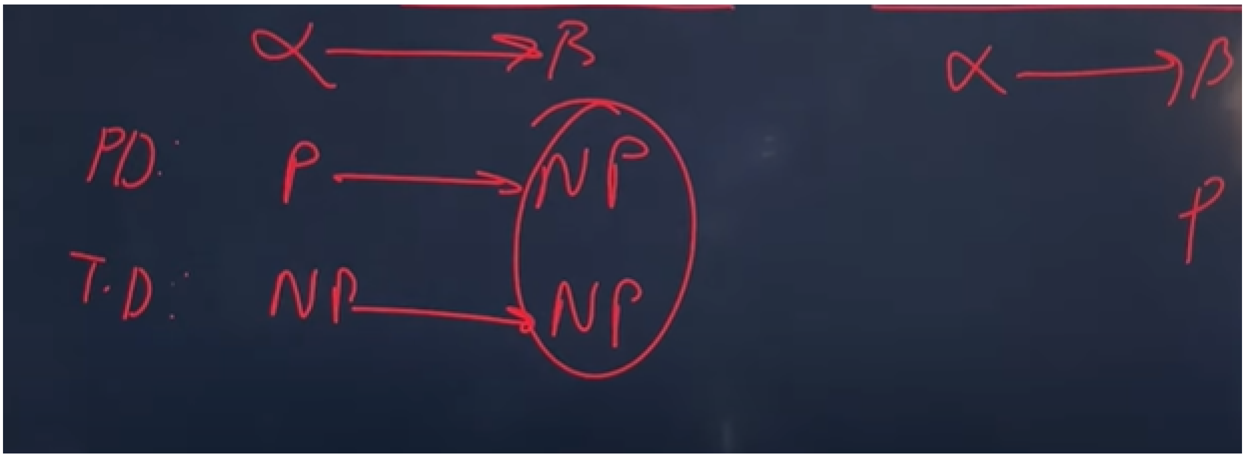




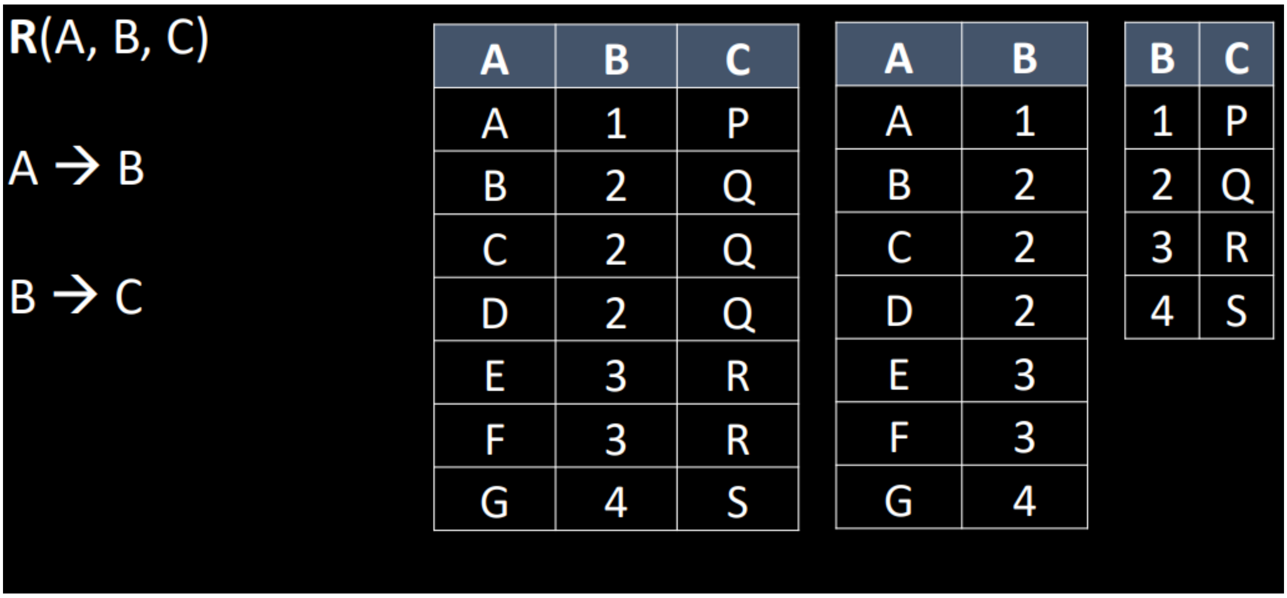
3NF

- 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 α-->β, either α is super key or β is the prime attribute



- `NPA->NPA NOT ALLOWED`



R(A, B, C)

A → B

B → C

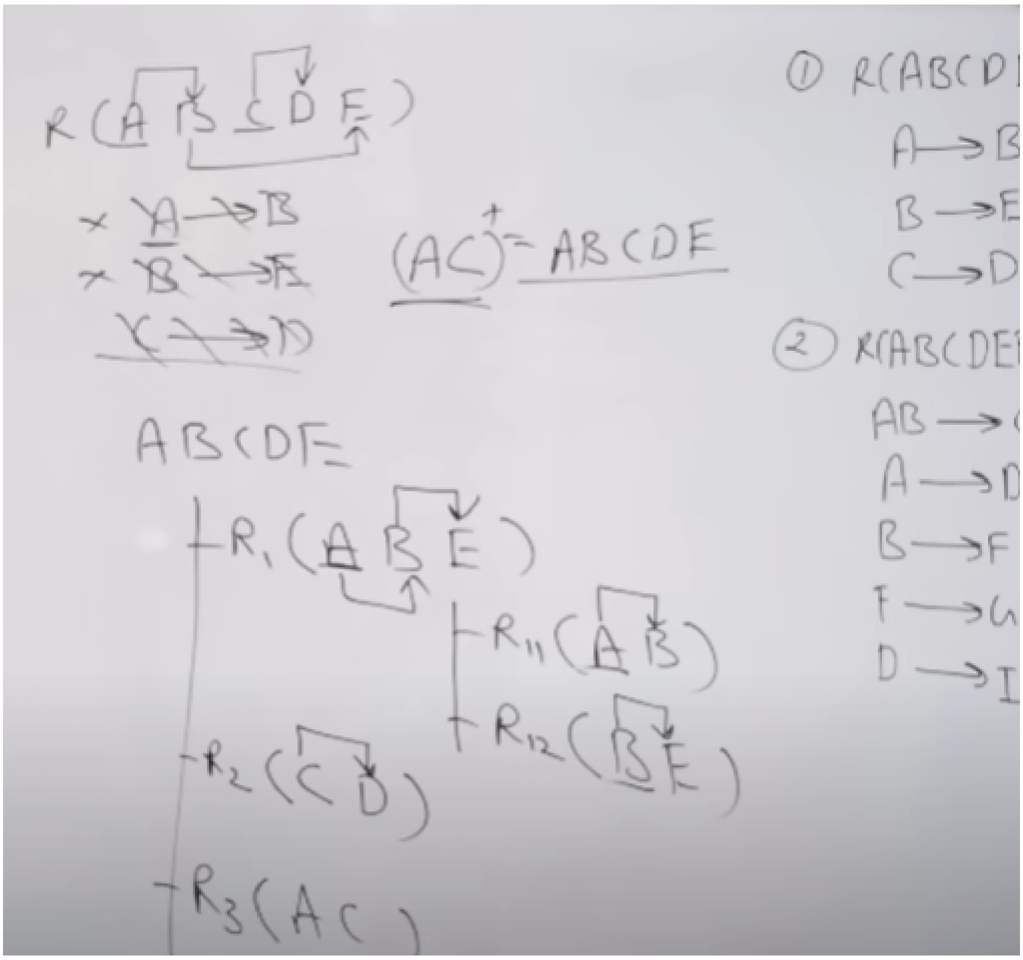
A	B	C
A	1	P
B	2	Q
C	2	Q
D	2	Q
E	3	R
F	3	R
G	4	S

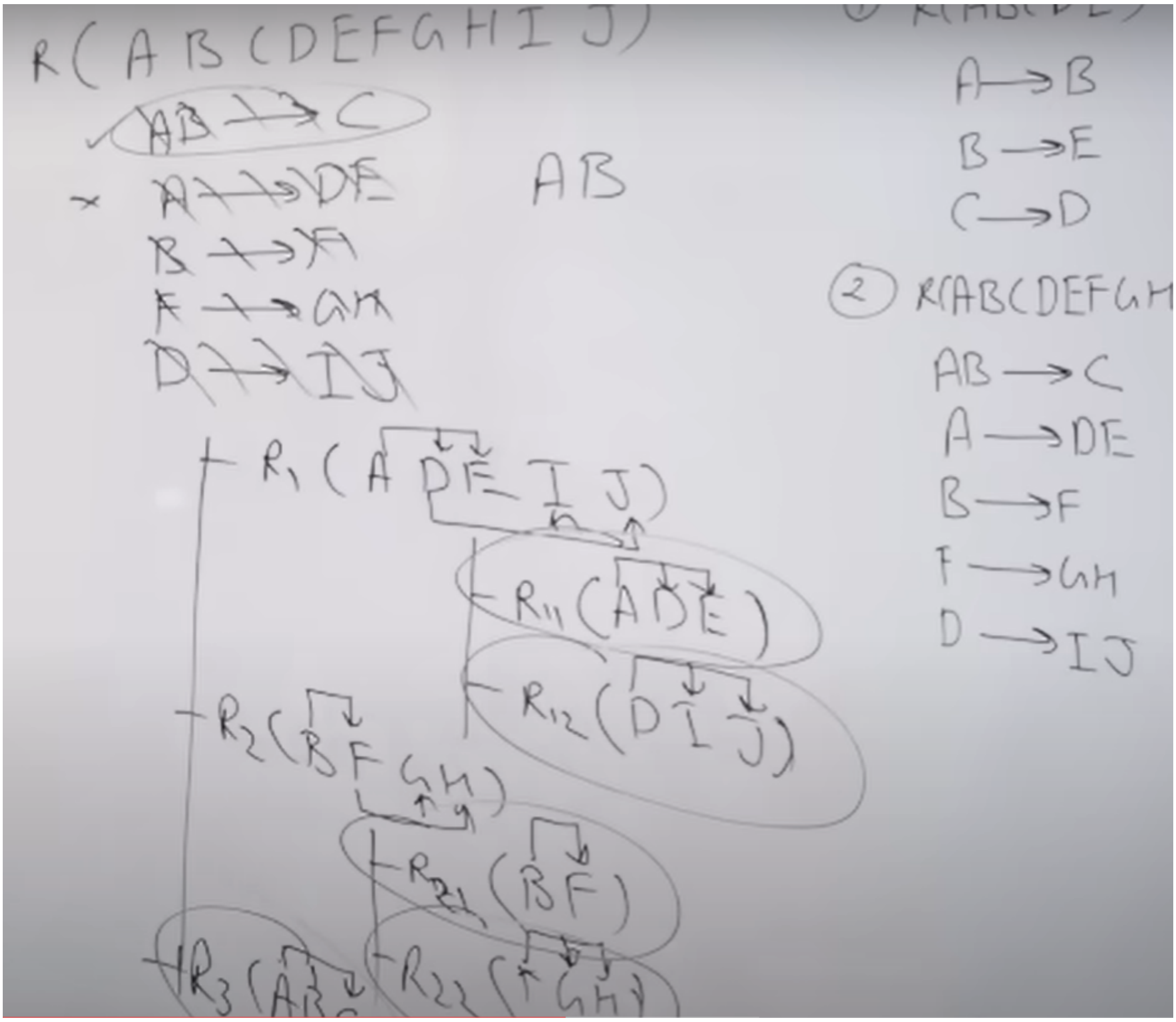
A	B
A	1
B	2
C	2
D	2
E	3
F	3
G	4

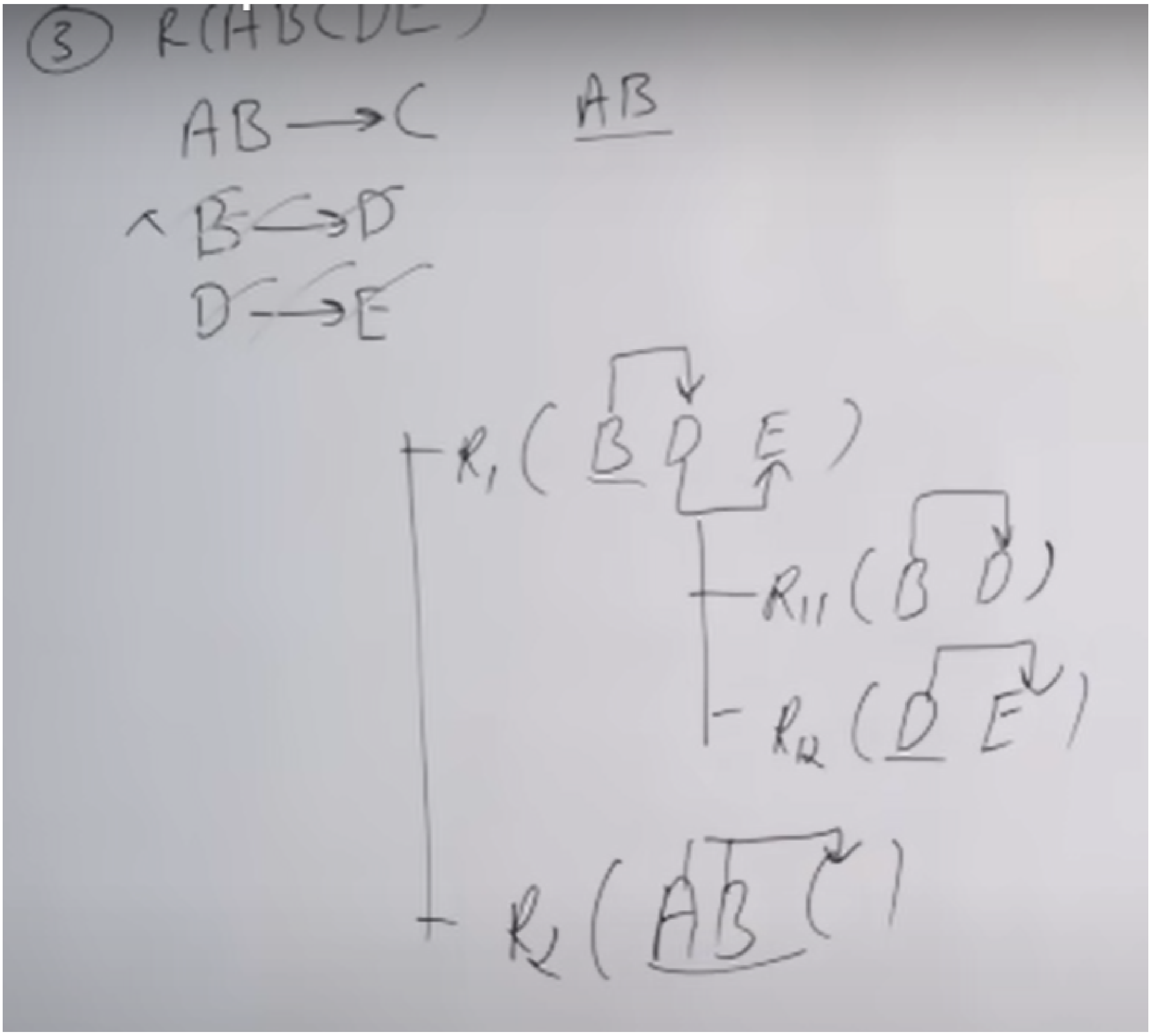
B	C
1	P
2	Q
3	R
4	S

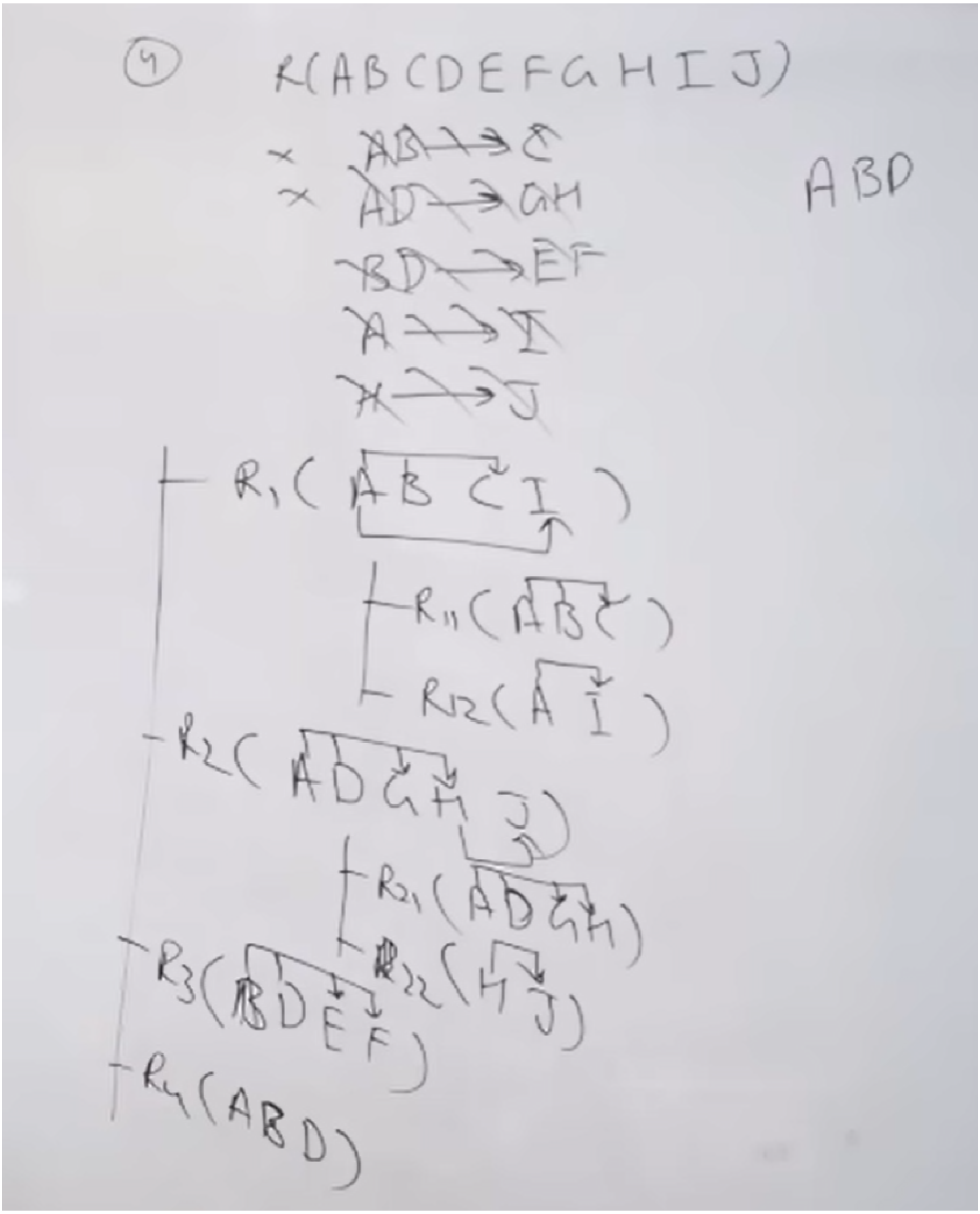
Decompose to 3NF

- `a - super key or b - PA`




R(ABCDEFGHIJ)
AB → C
A → DE
B → F
F → GH
D → IJ
AB
R1(ADEIJ)
R11(ADE)
R12(DIJ)
R2(BFGH)
R21(BF)
R22(FGH)
A → B
B → E
C → D
2 R(ABCDEFGH
AB → C
A → DE
B → F
F → GH
D → IJ


③ R(ABCDE)
AB → C
AB
B → D
D → E
R1 (B D E)
R11 (B D)
R12 (D E)
R2 (A B C)



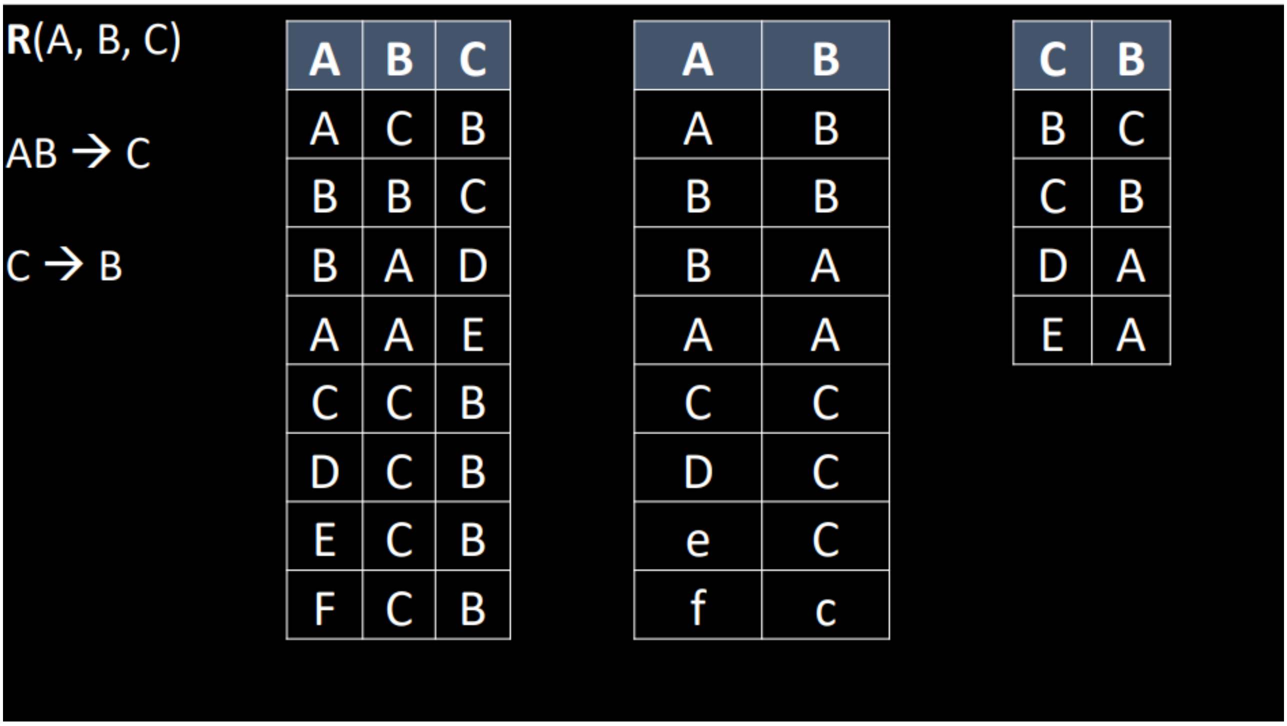
BCNF

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

Important Points

- A Relation with `two attributes is always in BCNF`.

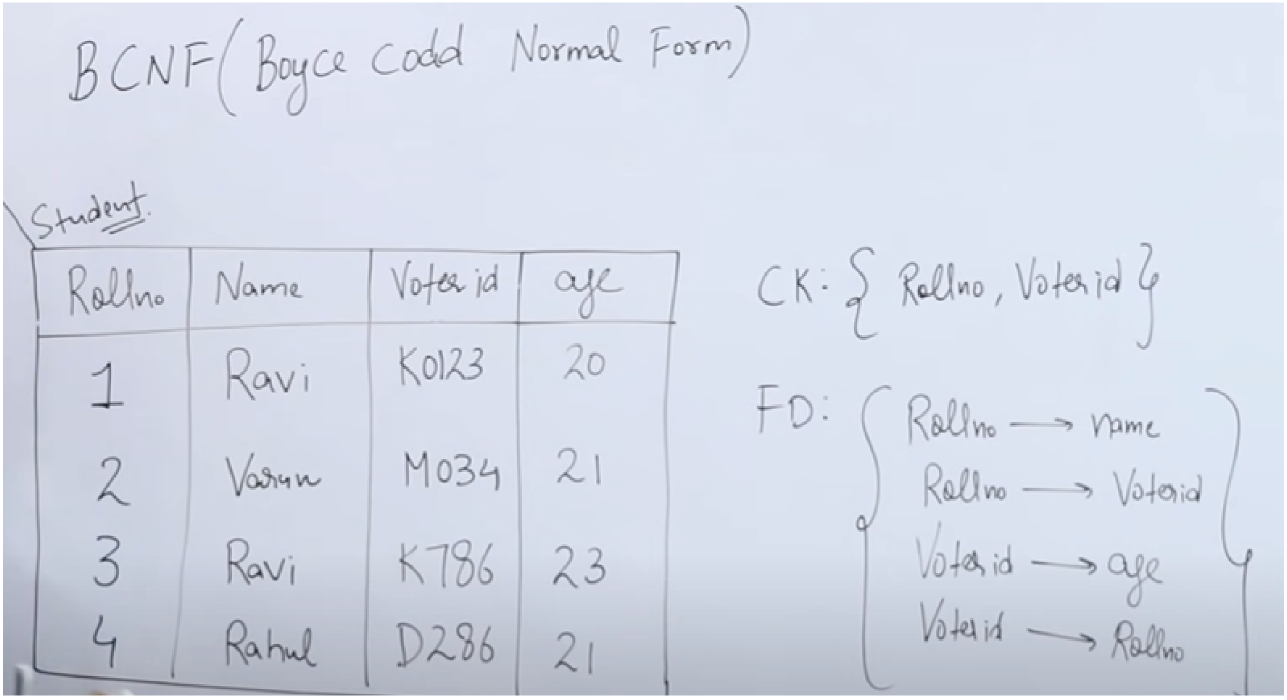
- A Relation schema R consist of `only prime attributes then R is always in 3NF`, but `may or may not be in BCNF`.
- Boyce-Codd Normal Form guarantees a good decomposition, without anomalies

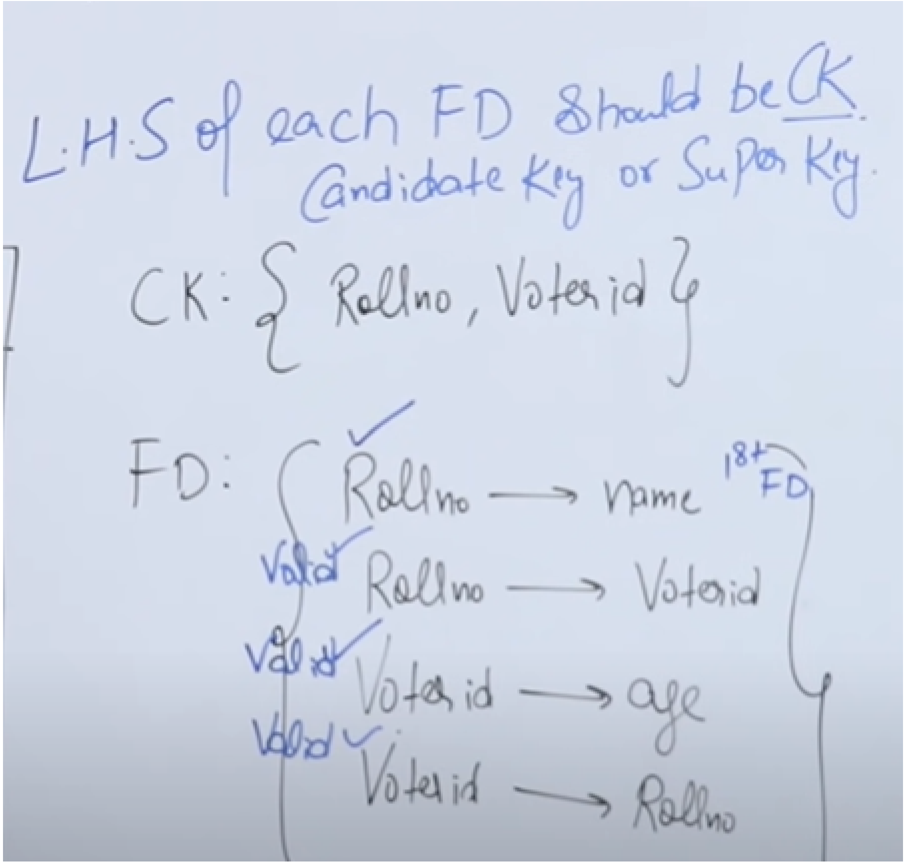


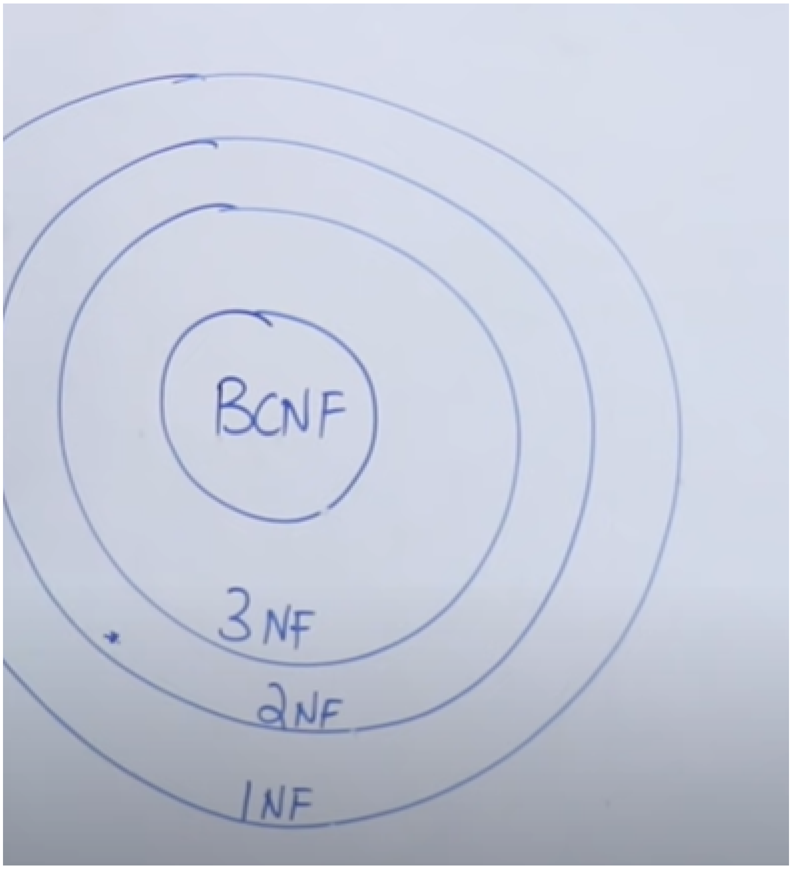
A	B	C
A	C	B
B	B	C
B	A	D
A	A	E
C	C	B
D	C	B
E	C	B
F	C	B

A	B
A	B
B	B
B	A
A	A
C	C
D	C
e	C
f	c

C	B
B	C
C	B
D	A
E	A

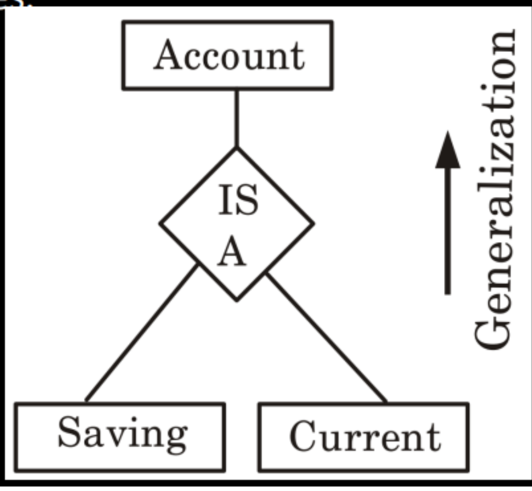



L.H.S of each FD should be CK
Candidate Key or Super Key.
CK: { Rollno, Voterid }
FD: Rollno → name
18+ FD
Valid Rollno → Voterid
Valid Voterid → age
Valid Voterid → Rollno


BCNF
3NF
2NF
1NF

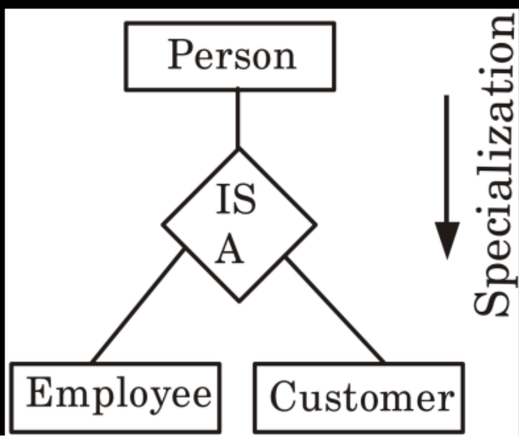
Generalization

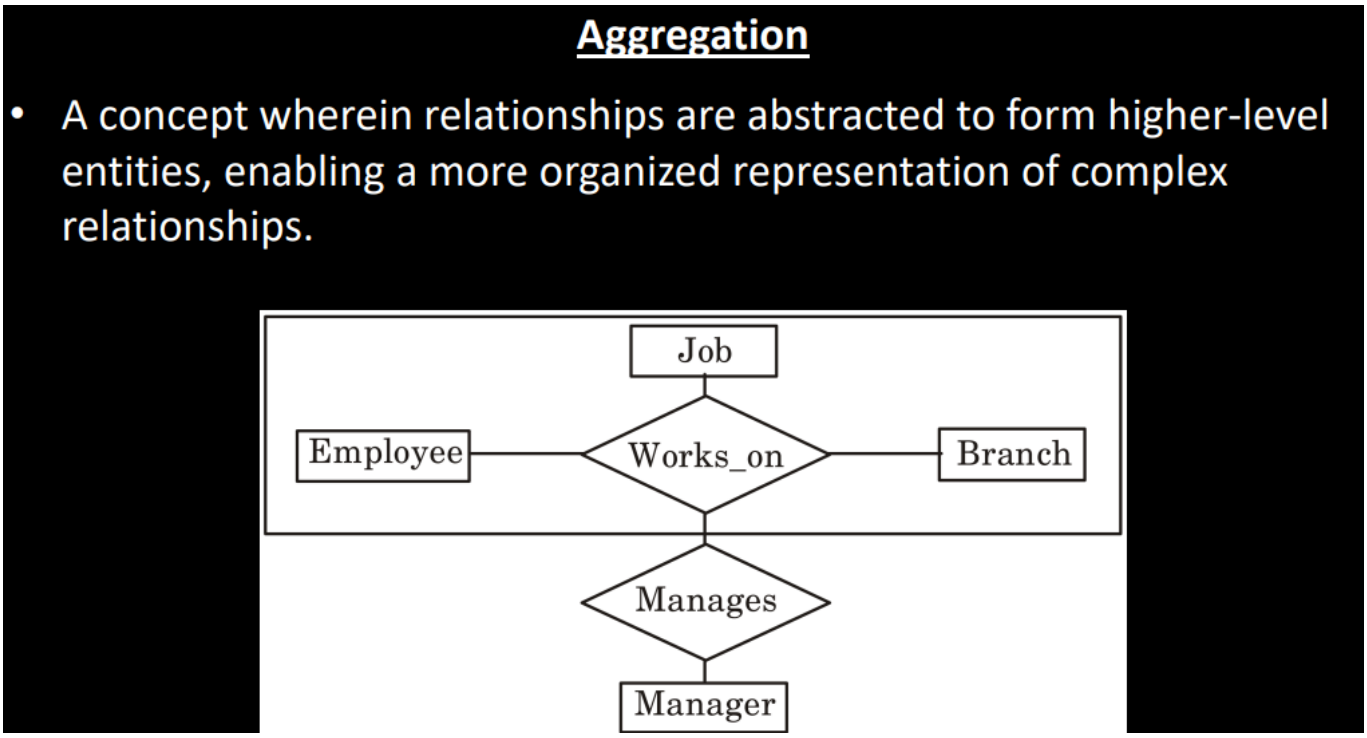
- Involves merging two lower-level entities to create a higher-level entity.
- A bottom-up approach that builds complexity from simpler components.
- Highlights similarities among lower-level entity sets while hiding differences.
- Leads to a simplified, structured data representation, aiding in database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



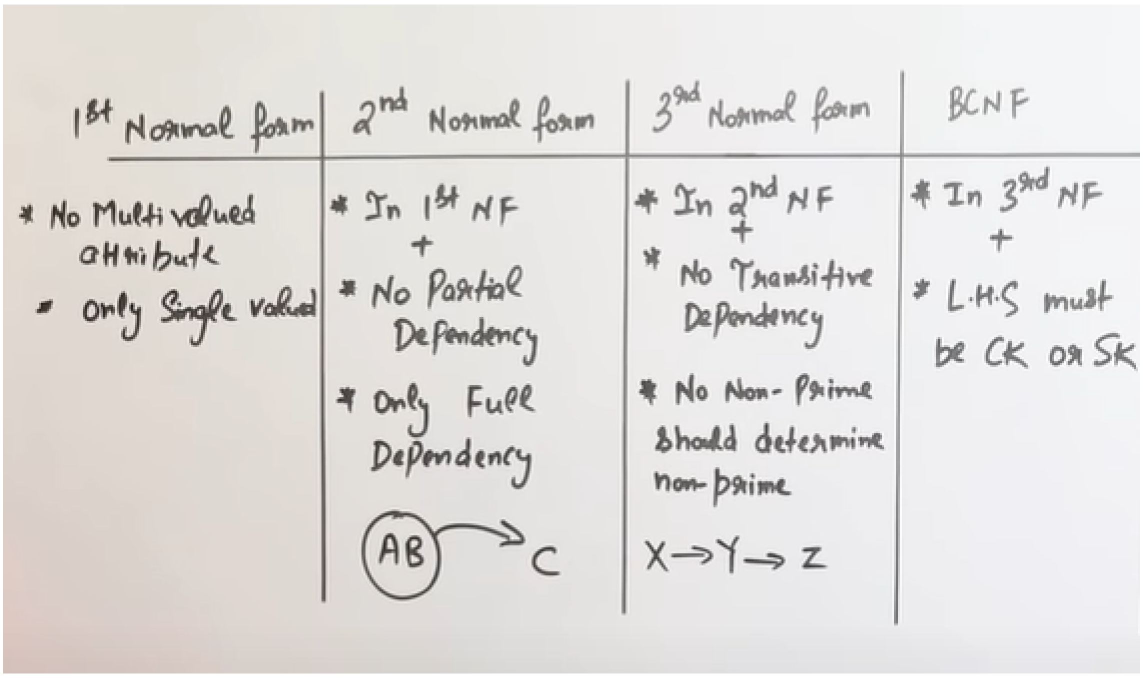


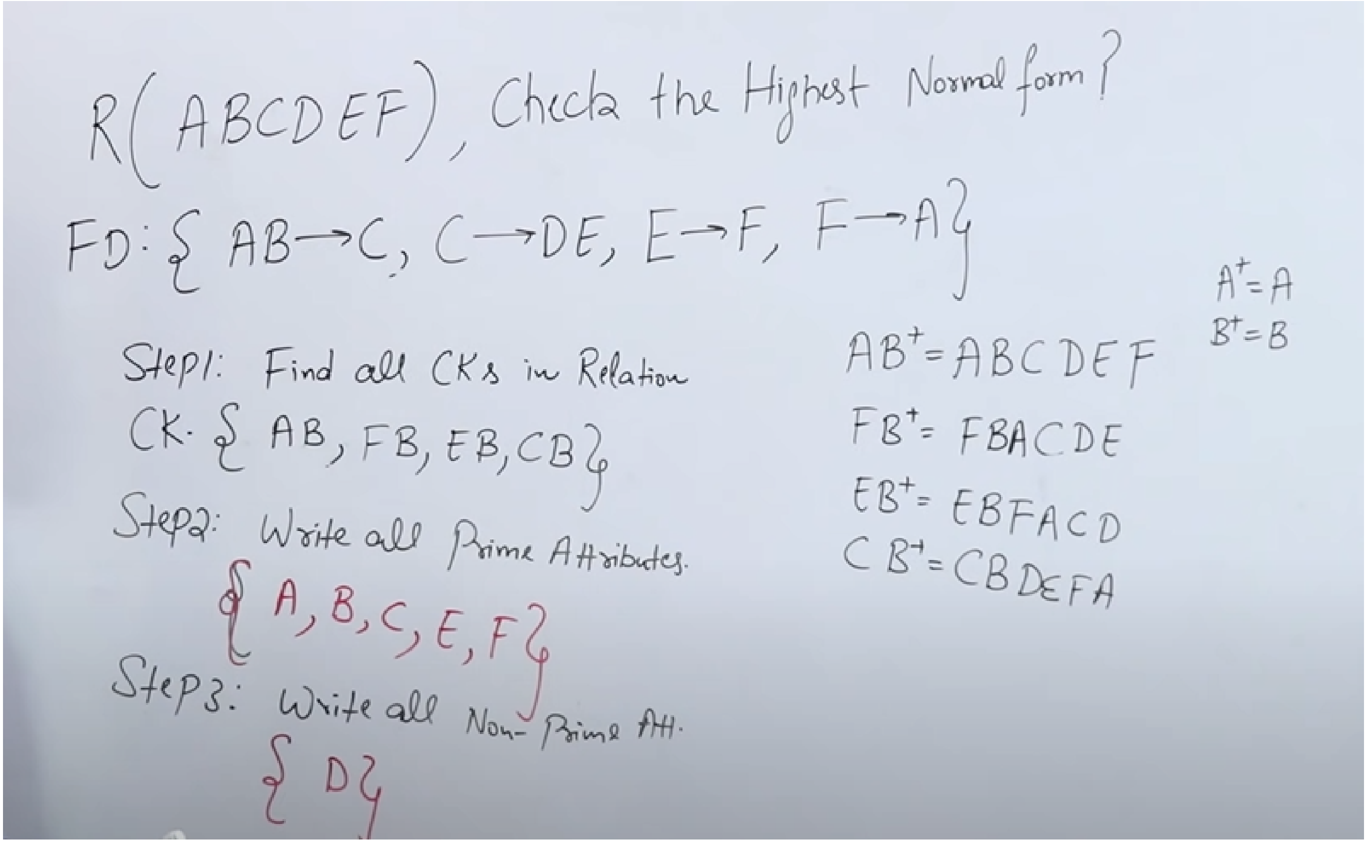
Advantages of BCNF

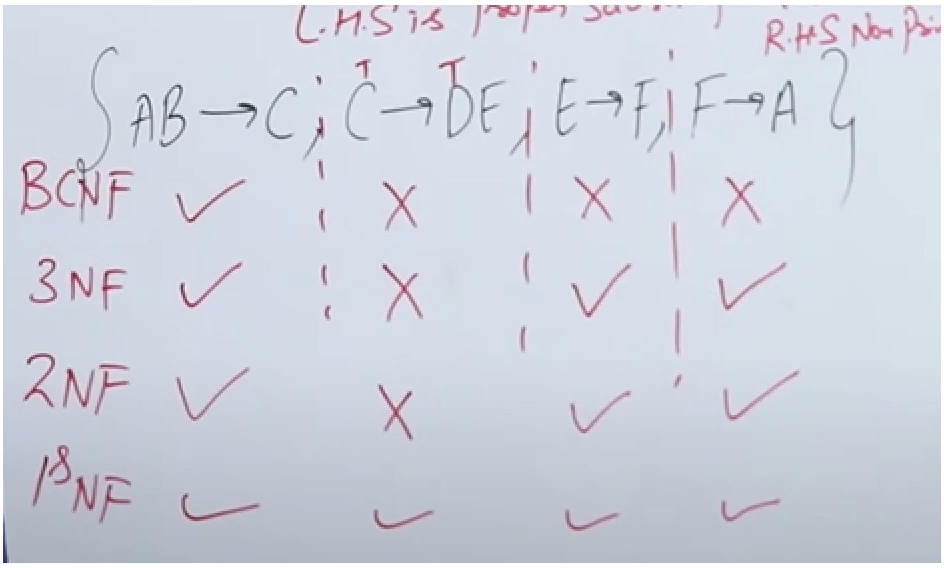
1. No anomalies : ✓ (Due to no redundancy)
2. Lossless Join : ✓
3. Dependency Preservation : ✗

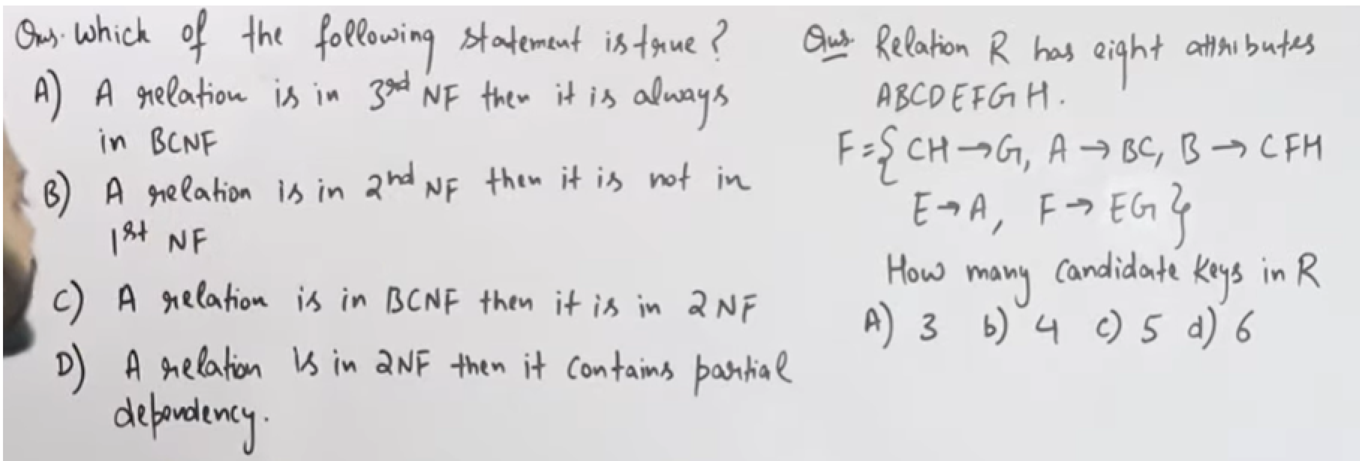
Summary

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









C, B

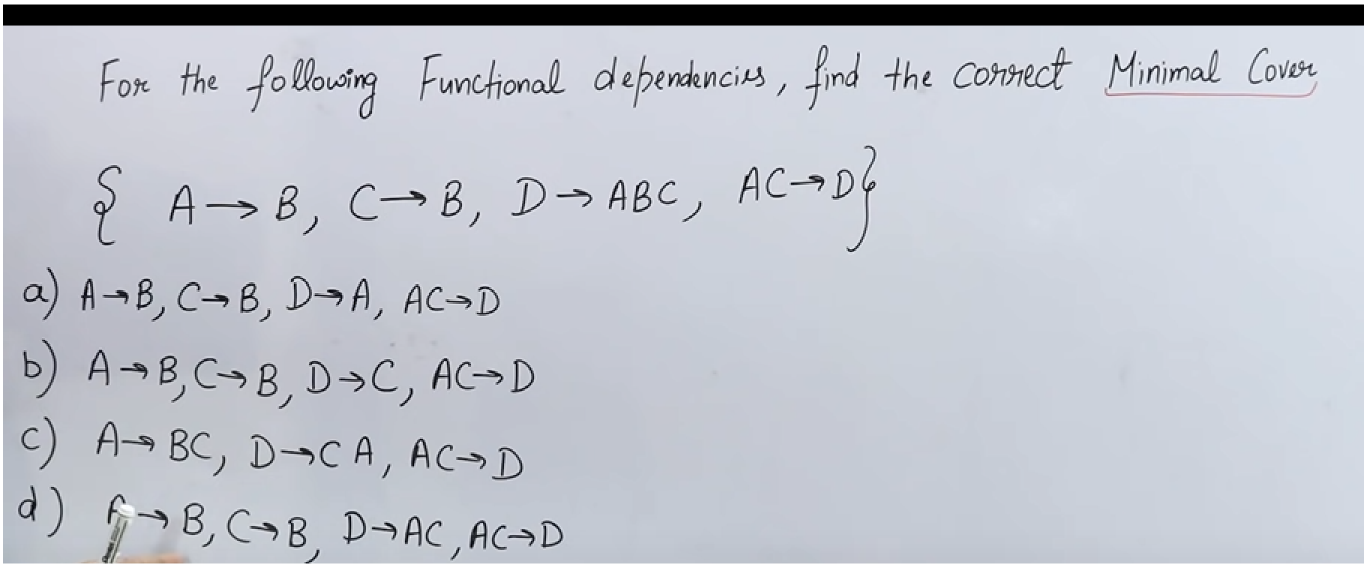
Decomposition

1. `Dependency Preservation Decomposition` - 3NF, Not BCNF
2. `Loseless Decomposition` - All NF

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

---

Minimal Cover



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

Equivalence of Functional Dependenies

x = {A->B, B->C} Y = {A->B, B->C, A->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) AB -> C : PD D->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