COMPUTER SCIENCE

Database Management System

FD's & Normalization

Lecture_11

Vijay Agarwal sir

01 Normal Forms

02 Normal Forms Decomposition

RDBMS Concept

FD Concept & its type

Attribute closure

Keys Concept

Super Keys

Candidate key

Finding Multiple C.K

Membership set

Equality b/w 2 FD set

Minimal Cover

Closure of FD set

Finding # Super keys & Max. C.k

Properties of Decomposition.

- Lossless Join Decomposition.
- Dependency Preserving Decomposition.


RDBMS

Normal Forms

Normal Form is a set of Rules to Reduce/eliminate the Redundancy

Redundandeny is unnecessary Repeattion of Data.

Normal Forms

There are Various Normal Forms

1. 1NF (First Normal Form)
2. 2NF (Second Normal Form)
3. 3NF (Third Normal Form)
4. BCNF (Boyce Codd Normal Form)

4NF, 5NF ... X

Normal Forms

Note Every Higher Normal Form Contain the Lower Normal Form.

Note If a Relation R is in 2NF. then its already is in 1NF.

Note If a Relation R is in 3NF then R already is in 2NF & 1NF

Note If a Relation R is in BCNF then R already is in 3NF, 2NF & 1NF.

Normal Forms

First Normal Form [1NF] :

A Relation Schema R is in 1NF if All Attribute of R are atomic.

(OR)

A Relation Schema R is in 1NF if R does not contain Any Multivalued Attribute.

STUDENT

RollNo	Name	Subject
1	Ajay	C/JAVA

Multivalued Attribute

Not in 1NF

∴

A	B	C
Roll	Name	Subject
1	Ajay	C
1	Ajay	JAVA

Now In 1NF

R (ABC) [A → B]

$(A)^+ = [AB]$

C is Not Present in FD, Whenever Any Attribute Not Present in FD then Add in (Make a part of) Candidate key.

Candidate key = [AC].

Normal Forms

- 1NF Ensured by using Candidate key.

Note: Default RDBMS is in 1NF.

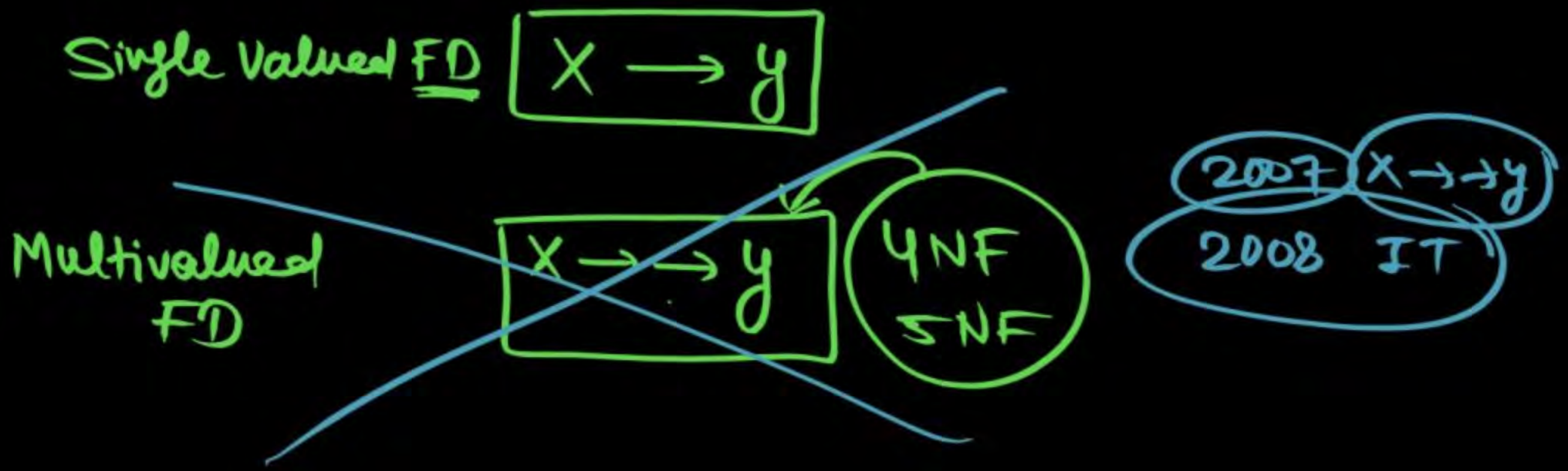
Note: In 1NF Redundancy Level is too High.

Redundancy Level:

1NF > 2NF > 3NF > BCNF

0% Redundancy ($X \to Y$)

Normal Forms



Key / Prime Attribute : is a Set of Attribute that belongs / Present in Some (Any) Candidate key

Non key / Non Prime Attribut : Set of Attributes that Not belongs / Not Present in Any Candidat key.

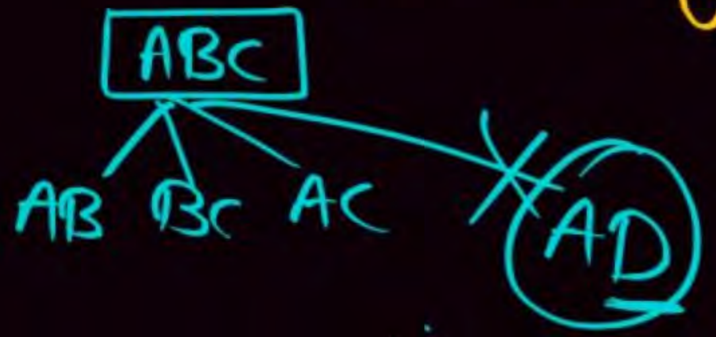
AB

Subset

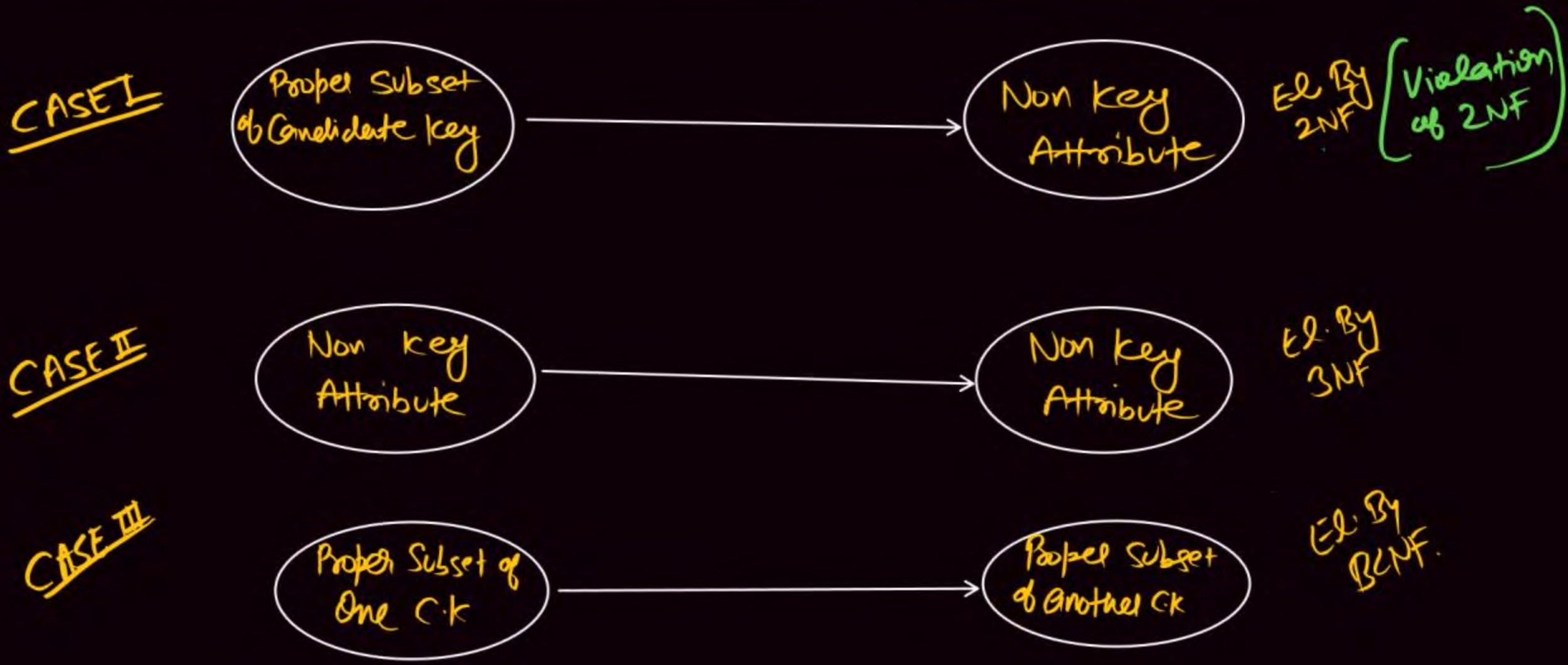
ϕ.
A
B
AB.

AB
A
B

Proper Subset



Possible Non Trivial FD which *cause* Redundancy.



CASE I

Proper Subset of Candidate key → Non key Attribute

R(ABCDEF) {AB→C , C→D , D→F , B→E}

Candidate key = {AB}

Non key Attribute = {C, D, E, F}

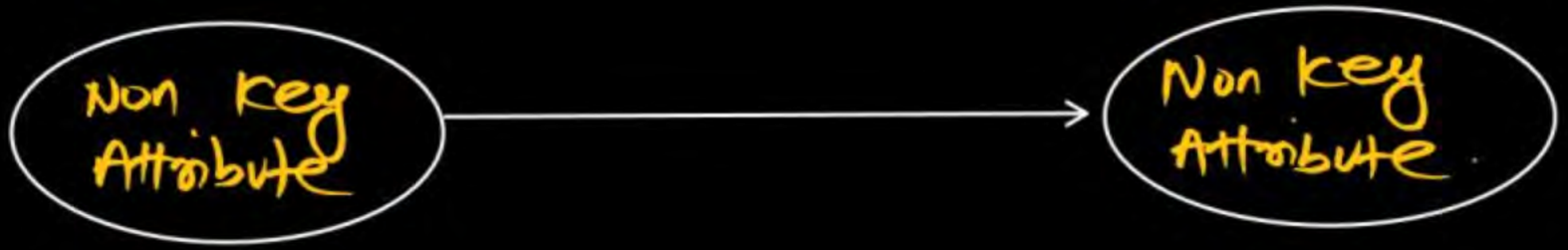
B → E

B: Proper Subset of C.k

E: Non key Attribute

Normal Forms

CASE II



Eg $R(ABC)$ $\{A \to B, B \to C\}$

Candidate key = [A]

Non key Attribute = [B, C]

CASE II

Proper Subset of One Candidate key → Proper Subset of another Candidate key

CK is Minimal.

eg R(ABCD) [AB → CD, D → A]

Candidate key = [AB, DB]

D → A

D: Proper Subset of One Candidate key

A: Proper Subset of another CK

B → A

A → B

CK: A (or) B | not AB

D → B

B → D

CK: B (or) D

Not DB

FD① FD②

R(ABC) [AB → C , A → C]

Which is the Partial FD.

Ans(A)

(a) Only I (35.71.)

(b) Only II (54.76)

(c) Both (9.521.)

(d) None

R(ABC) [AB → C, B → C]

① ②

Identify Partial FD

Ans [C]

(a) None (8.16%)

(b) Both (4.08%)

(c) AB → C (14.29%)

(d) B → C (77.5%)

Partial FD

AB → C is Partial FD

~~AB → C~~

~~AB → C~~

A → C

OR

B → C

Normal Forms

Navathe

15.3.5 Second Normal Form

Second normal form (2NF) is based on the concept of *full functional dependency*. A functional dependency $X \rightarrow Y$ is a **full functional dependency** if removal of any attribute A from X means that the dependency does not hold any more; that is, for any attribute $A \in X$, $(X - \{A\})$ does *not* functionally determine Y. A functional dependency $X \rightarrow Y$ is a partial dependency if some attribute $A \in X$ can be removed from X and the dependency still holds; that is, for some $A \in X$, $(X - \{A\}) \rightarrow Y$. In Figure 15.3(b), {Ssn, Pnumber} → Hours is a full dependency (neither Ssn → Hours nor Pnumber → Hours holds). However, the dependency {Ssn, Pnumber} → Ename is partial because Ssn → Ename holds.

Definition. A relation schema R is in 2NF if every nonprime attribute A in R is *fully functionally dependent* on the primary key of R.

$X \rightarrow y$ is FULL FD

$A \in X$

$(X - A) \not\rightarrow y$

$X \rightarrow y$ is Partial

$AB \rightarrow C$ is Partial FD

$(AB - A) \rightarrow C \Rightarrow B \rightarrow C$ (or)

$(AB - B) \rightarrow C \Rightarrow A \rightarrow C$ (or)

Ssn Pnumber → Hours Full FD

neither Ssn ↛ Hours

Pnumber ↛ Hours.

Ssn Pnumber → Ename is Partial FD

Ssn → Ename

$AB \longrightarrow C$ is Partial FD

$B \rightarrow C$

$A \rightarrow C$

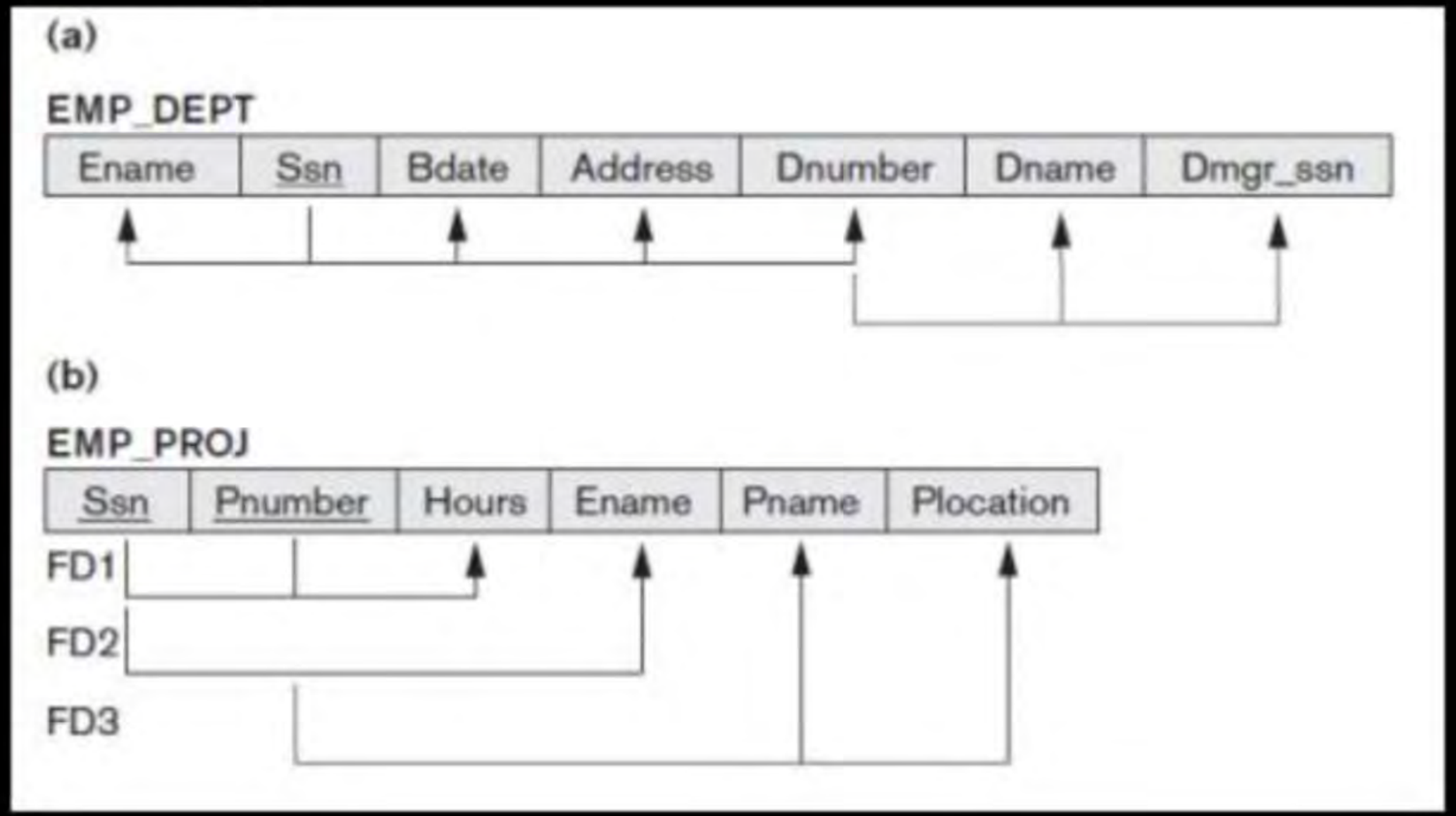
$AB \rightarrow C$ is Full FD

if $B \not\rightarrow C$

(or)

if $A \not\rightarrow C$

Figure 15.3


(a)
EMP_DEPT
Ename
Ssn
Bdate
Address
Dnumber
Dname
Dmgr_ssn
(b)
EMP_PROJ
Ssn
Pnumber
Hours
Ename
Pname
Plocation
FD1
FD2
FD3

Normal Forms

kooth

8.17 A functional dependency $\alpha \rightarrow \beta$ is called a partial dependency if there is a proper subset γ of α such that $\gamma \rightarrow \beta$. We say that β is *partially dependent* on α. A relation schema R is in second normal form (2NF) if each attribute A in R meets one of the following criteria:

- It appears in a candidate key.
- It is not partially dependent on a candidate key.

$\alpha \rightarrow \beta$ is Partial FD

γ is Proper Subset of α

$\gamma \rightarrow \beta$

$AB \rightarrow C$ is Partial FD

Proper Subset of AB < A, B

$A \rightarrow C$ OR $B \rightarrow C$

Identify Partial FD ?

R(ABCDEF) {ABC → DE, DE → ABC, AB → D, DE → F, E → C}

(i) ABC → D [P]

(ii) AE → C [P]

(iii) AF → D

(iv) AB → D

(v) AC → D

(vi) BC → D

(vii) DE → C

(viii) AB → F

(i) ABC → D

AB → D

∴ So ABC → D is Partial FD.

(ii) AE → C

for checking AE → C is Partial FD ?

A → C ? $[A]^+ = [A]$

E → C ? $[E]^+ = [EC]$

So AE → C is Partial FD

Identify partial FD ?

R(ABCDEF) {ABC → DE, DE → ABC, AB → D, DE → F, E → C}

(i) ABC → D

(ii) AE → C

(iii) AF → D

(iv) AB → D

(v) AC → D

(vi) BC → D

(vii) DE → C

(viii) AB → F


THANK
YOU!