COMPUTER SCIENCE
PW
Database Management System
Transaction Concept: ACID Properties, Schedule types, Serializable Schedule(Conflict & View Serializable)
Lecture_03
GATE WALLAH
Vijay Agarwal sir

FD & Normalization → 3

- RDBMS Concept
- FD & its type
- Attribute Closure
- Super key
- Candidate key
- finding Multiple Candidate key
- Membership Set
- Closure of FD Set $[F]^+$
- Equality b/w 2 FD Set
- minimal Cover
- Properties of Decomposition
  - → Lossless Join
  - → Dependency Preserving

Normalization : is Process to Reduce the Redundancy.

Normal Forms

Normal Form : Set of Rules, used to Reduce/eliminate the Redundancy.

1. 1NF
2. 2NF
3. 3NF
4. BCNF

Note: Every Higher Normal Form Satisfy the Lower Normal Form.

- If Relation R is in 2NF, then its already is in 1NF.

- If Relation R is in BCNF that means its already is in 3NF, 2NF and 1NF also.

1NF (First Normal Form)

Relational Schema R is in 1NF iff 'R' does not Contain Any Multi Valued Attribute

OR

R is in 1NF iff all attribute of R are atomic

Sid	name	Subject
S1	A	C/JAVA

Multivalued Attribute

R is Not in 1NF

R(ABC)

AC

A	B	C
Sid	sname	Subject
S1	A	C
S1	A	JAVA

R is in 1NF.

$R(ABCD) \quad [A \to B,\ B \to C]$

$(A)^+ = [ABC]$

$(AD)^+ = [ABCD]$

AD is Candidate key

R is in 1NF.

Note Default RDBMS is in 1NF.

Candidate key ensured R is in 1NF.

In 1NF Redundancy level is very High.

$$\boxed{1NF > 2NF > 3NF > BCNF}$$

Possible Non Trivial FD Which Create Redundancy

Case I

Proper subset of Candidate key → Non key / Non prime Attribute

Elimate by 2NF

Case II

Non key Attribute → Non key Attribute

El by 3NF

Case III

Proper subset of one C.K → Proper subset of another C.K

El By BCNF.

2NF Not allowed Case I but allow Case 2 & Case 3.

3NF does not allowed Case 1, Case 2 but allowed Case 3.

BCNF Not allowed Case I, Case 2, Case 3.

BCNF has 0% Redundancy.

Partial Dep.

Case I :- Proper Subset of C.K → Non key Attribute

(eg) R(ABCDEF) [AB → C , C → DF , B → E]

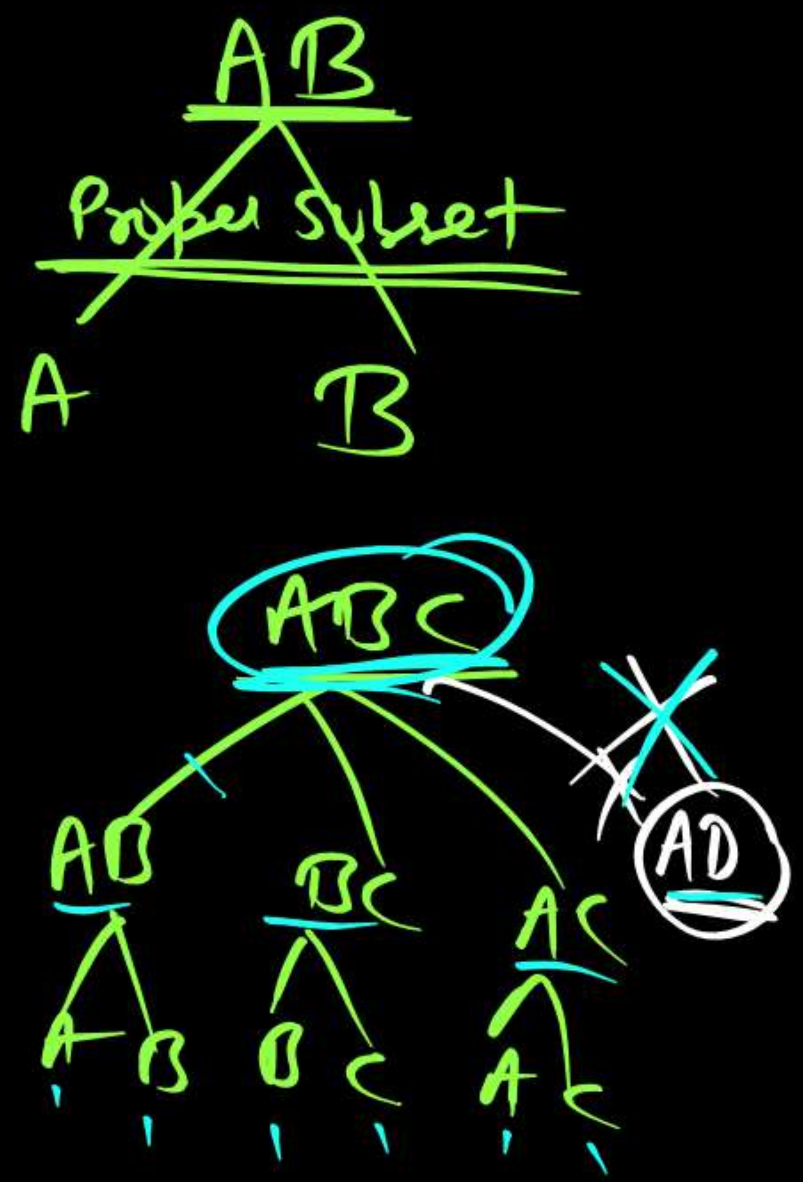
Candidate key = [AB] → A, B

Non key / Non Prime Attribute = [C, D, E, F]

B → E

B: Proper Subset of C.K

E: Non key Attribute



Case II : Non key Attribute $\xrightarrow{\text{Transitive Dep.}}$ Non key Attribute

eg $R(ABC)$ $[A \to B,\ B \to C]$

Candidate key = (A)

Non Prime | Non key = (B, C)

$B \to C$

Non key Attribute   Non key Attribute

Case III

Proper Subset of One C.K $\longrightarrow$ Proper Subset of another C.K

$R(ABCD)\ [AB \to CD,\ D \to A]$

Candidate key = (AB, DB)

$D \longrightarrow A$

Proper Subset of One C.K

Proper Subset of another C.K

FD Which Does not Create Redundancy

$X \rightarrow y$

(1) Trivial FD

(2) $X \longrightarrow y$

Every determinant is Super key

No Redundancy

- Concept
- NF type
- Decomposition
- Question (PYQ)

2NF : [Second Normal Form]

Let R be the Relational Schema. R is in 2NF

iff

1. R is in 1NF.
2. R does not Contain Any Partial Dependency (P.D)

Partial Dependency

Proper Subset of Candidate key $\rightarrow$ Non key Attribute

R(ABCDEFGH) {AB → C, C → D, B → E, E → F, A → G, G → H}

P.D P.D

Candidate key = {AB}

Non Prime | Non key Attribute = {C, D, E, F, G, H}

AB

Proper Subset

A B

Check 2NF:

PD ?

{B → E A → G}

Proper Subset of C.K → Non key Attribute

R Not in 2NF.

Partial Dep.

Proper subset of Candidate key $\longrightarrow$ Non key / Non Prime Attribute

Normal Forms

Second Normal Form

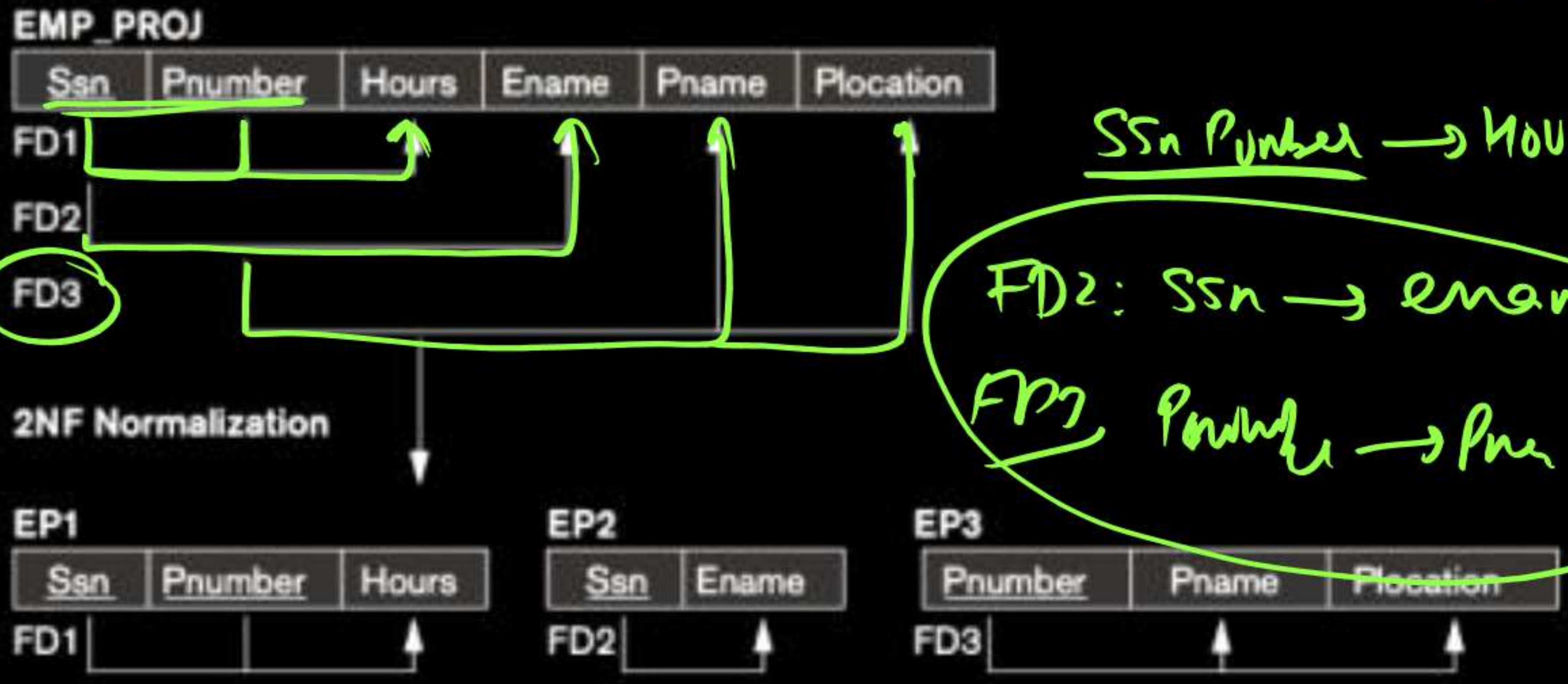
AB → C

B → C

Candidate key: AB

N.K = (C)

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.



Ssn Pnumber → Hours

FD2: Ssn → ename

FD3 Pnumber → Pname Ploc

P.D

FD2 & FD3

Q. Let R(A, B, C, D, E, P, G) be a relational schema in which the following functional dependencies are known to hold:

$AB \rightarrow CD$, $DE \rightarrow P$, $C \rightarrow E$, $P \rightarrow C$ and $B \rightarrow G$.

(GATE : 2 marks)

The relational schema R is

(A) In BCNF

(B) In 3NF, but not in BCNF

(C) In 2NF, but not in 3NF

(D) Not in 2NF

Candidate key = (AB)

Non key Attribute = (C, D, E, P, G)

$B \rightarrow G$

Partial Dep.

3NF : (Third Normal Form)

Relation Schema 'R' is in 3NF

iff

(1) R is in 2NF.

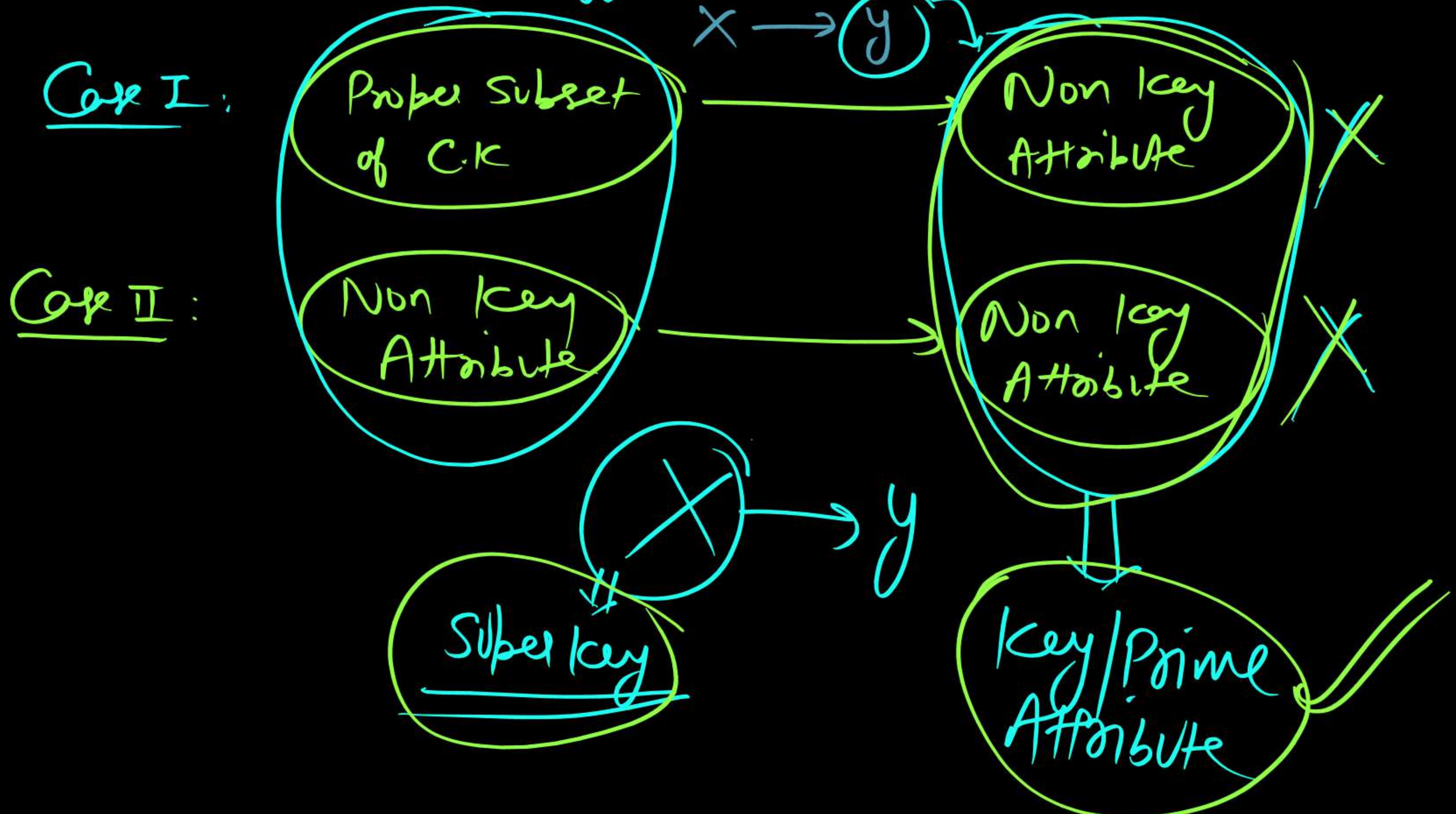
(2) R does not contain Any Transitive Dependency

$$\Downarrow$$

$$B \to C$$

(Q) $R(ABC)$ $[A \to B,\ B \to C]$

Relation R is in 3NF iff it does not allow both Case



Let R be the Relational Schema is in 3NF.

iff every Non Trivial FD $X \rightarrow Y$ must satisfy the following

Condition : $X \longrightarrow Y$ $\qquad X \longrightarrow A$

either X : Super key

or

Y : Prime/key Attribute

$R(ABCDE)$ $\{AB \rightarrow CD,\ D \rightarrow E\}$

Check 2NF ?

Candidate key = (AB)

Check PD ?

Non key Attribute = (C, D, E)

$D \rightarrow E$

Not Proper subset

$\therefore R$ in 2NF.

Check 3NF ?

$X \rightarrow Y$

X: superkey

or

Y: Prime Attribute

$AB \rightarrow CD$ AB Superkey

$D \rightarrow E$

Nonkey Attribute $\rightarrow$ Nonkey Attribute

Not in 3NF

R(ABCDE) {AB → C, D → E, AB → E}

Candidate key = {ABD}

Non key Attribute = {C, E}

AB → C   D → E   AB → E

3 P.D.

R(ABCDE) [AB → CD, D → E, AB → E]

Candidate key = {AB}

Non key Attribute = {C, D, E}

Check 2NF

	$AB \to CD$	$\boxed{D \to E}$	$AB \to E$	
2NF	✓	✓	✓	2NF ✓
3NF	✓	N.K → N.K ✗	✓	3NF ✗

R(ABCD) {AB → CD, D → A}

Candidate key = (AB, DB)   Key Attribute = {A, B, D}
Non key = {C}.

Check 2NF? Check PD?   No P.D ; So R is in 2NF.

X → Y

Check 3NF   AB → CD; AB is Superkey.

D → A ; D is Not Superkey But
A is Prime|key Attribute
So R is in 3NF

Normal Forms

Third Normal Form

C.K : [A]

Non Prime : [B, C]

R(ABC) [A→B, B→C]

Definition: According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.

Definition: A relation schema R is in third normal form (3NF) if, whenever a nontrivial functional dependency $X \rightarrow A$ holds in R either (a) X is a superkey of R, or (b) A is a prime attribute of R.

C.K : Ssn

Ssn → enam, bdate, Add, Dnumber

Dnumber → Dnam, Dmgr Ssn

BCNF : R is in BCNF if Not allowed all 3 Case

$X \longrightarrow Y$

	X		Y	
Case I	Proper Subset of C.K	$\longrightarrow$	Non key Attribute	✗
Case II	Non key Attribute	$\longrightarrow$	Non key Attribute	✗
Case III	Proper Subset of One CK	$\longrightarrow$	Proper Subset of another CK	✗

BCNF :

every Non Trivial $X \rightarrow y$ FD satisfy the following Condition

$$X \longrightarrow y$$

X : Super key

Boyce – Codd Normal Form

Definition: A relation schema R is in BCNF if whenever a nontrivial functional dependency $X \rightarrow A$ holds in R, then X is a superkey of R.

$X \longrightarrow A$

X : super key

$X \rightarrow y$

2NF : ⇒ P.D — Proper Subset of C.K → Non key Attribute ✗

3NF : X Super key (or) y: Prime/key Attribute (OR) Non key Attribute → Non key Attribute ✗

BCNF X : Super key (or) Proper Subset of One CK → Proper Subset of another C.K ✗

Q. In a relational data model, which one of the following statements is TRUE?

(GATE 2022)

(A) A relation with only two attributes is always in BCNF.

(B) If all attributes of a relation are prime attributes, then the relation is in BCNF.

(C) Every relation has at least one non-prime attribute.

(D) BCNF decompositions preserve functional dependencies.

Note If All Attribute of Relation R is Prime/key Attribute then R is in 3NF But may/may Not in BCNF.

R(ABCDE) {A → B, B → C, C → D, D → A}

Candidate key: {AE, BE, CE, DE}

Prime Attribute = (A, B, C, D, E)

Check 3NF? X : super key

or

y : Prime Attribute

R is in 3NF

BCNF

X → y   X : super key

Not in BCNF

R(AB) [A → B] | [A → B] [B → A] | [B → A]

C.K : A | C.K : A, B | CK : B

Check BCNF

X : superkey ✓ ✓ ✓

BCNF

Binary Relation (Relation with 2 Attribute is in BCNF)

$R(ABCDE)$ $\{A \to B,\ B \to C,\ C \to D,\ D \to E,\ C \to A\}$

Candidate key = $\{A, C, B\}$

Check 2NF ? YES its in 2NF

✗ 3NF

Check 3NF ? $A \to B$ ✓ $B \to C$ ✓ $C \to D$ ✓ $D \to E$ $C \to A$ ✓

If All Candidate key is simple key then R is in 2NF But May / may Not in 3NF & BCNF.

If Relation R has One C.K then R is in 1NF

(a) 1NF

(b) 2NF

(c) 3NF

(d) BCNF

R(ABCDE) {A → B, C → D, D → E}

Candidate key = (AC)

Check 2NF ? Not in 2NF

Q. Which of the following statement is/are true? (MSQ)

(A) Second normal form (2NF) have transitive dependency.

(B) ~~No relation can be in both BCNF and 3NF.~~ $\Rightarrow R(ABC)\ [A \to B,\ B \to C,\ C \to A]$

(C) Second normal form(2NF) does not have partial dependency.

(D) ~~In BCNF lossless join & dependency - preserving decomposition is always possible.~~

A & C

Q. Let R(A, B, C, D, E, P, G) be a relational schema in which the following functional dependencies are known to hold:

$AB \rightarrow CD$, $DE \rightarrow P$, $C \rightarrow E$, $P \rightarrow C$ and $B \rightarrow G$.

The relational schema R is

(A) In BCNF

(B) In 3NF, but not in BCNF

(C) In 2NF, but not in 3NF

(D) Not in 2NF

Q

Consider the following statements: **[MSQ]**

In Correct

S_1: If every attribute is prime attribute in R, then Relation R will always be in BCNF.

Binary Rel^n → BCNF

Correct S_2: Any Relation with two Attribute is in 3NF and 2NF. → Correct

Cor S_3: If every key of relation R is a simple candidate key (No composite key) then the relation R not always in 3NF.

In S_4: In BCNF there is always a lossless join and Dependency Preserving Decomposition.

Which of the above statement are incorrect

(A) S_1

(B) S_2

(C) S_3

(D) S_4

R(ABCDE) {A→B, B→C, C→D, D→E, C→A}

C.K: {A, C, B}

$D \rightarrow E$

Non key   Non key

Q

The relation scheme student Performance (name, courseNO, rollNo, grade) has the following functional dependencies:

[2004: 2 Marks]

name, courseNo $\rightarrow$ grade

RollNo, courseNo $\rightarrow$ grade

name $\rightarrow$ rollNo

rollNO $\rightarrow$ name

C.K [name Cno, Rollno Cno.]

The highest normal form of this relation scheme is

(A) 2 NF

(B) 3 NF

(C) BCNF

(D) 4 NF

Normal Form Decomposition

2NF Decomposition

Which FD Create P.D take a Closure & Make a Separate table

& Remaing all Attribut & original C.K

Lossless + D.P

$X \to Y$ $t_1.X = t_2.X$ then $t_1.Y = t_2.Y$ must be same

Multi Valued FD.

$\boxed{X \to\to Y}$

Roll	Name	Course
S_1	A/B	C/JAVA

S_1	A	C
S_1	A	JAVA
S_1	B	C
S_1	B	JAVA

2NF Decomposition

Q R(ABCDEFGH) {$AB \to C$, $C \to D$, $B \to E$, $E \to F$, $A \to GH$}

Candidate key = (AB)

Non key Attribute = (C, D, E, F, G, H)

2NF Decomposition

$(B)^+ = (BEF)$

$(A)^+ = (AGH)$

R_1: A B C D

R R_2: A G H

R_3: B E F

$(B)^+ = (BEF)$ S.K of R_3

$R_{12}(ABCDGH) \cap R_3(BEF)$

$R_{123}(ABCDEFGH)$ lossless

2NF Decomposition

Q

R(ABCDE) F: [$A \rightarrow B$, $B \rightarrow E$, $C \rightarrow D$]

Decompose it into 2NF.

Candidate key = [AC]

2NF Decomposition

$[A]^+ = [ABE]$

$[C]^+ = [CD]$

R_1

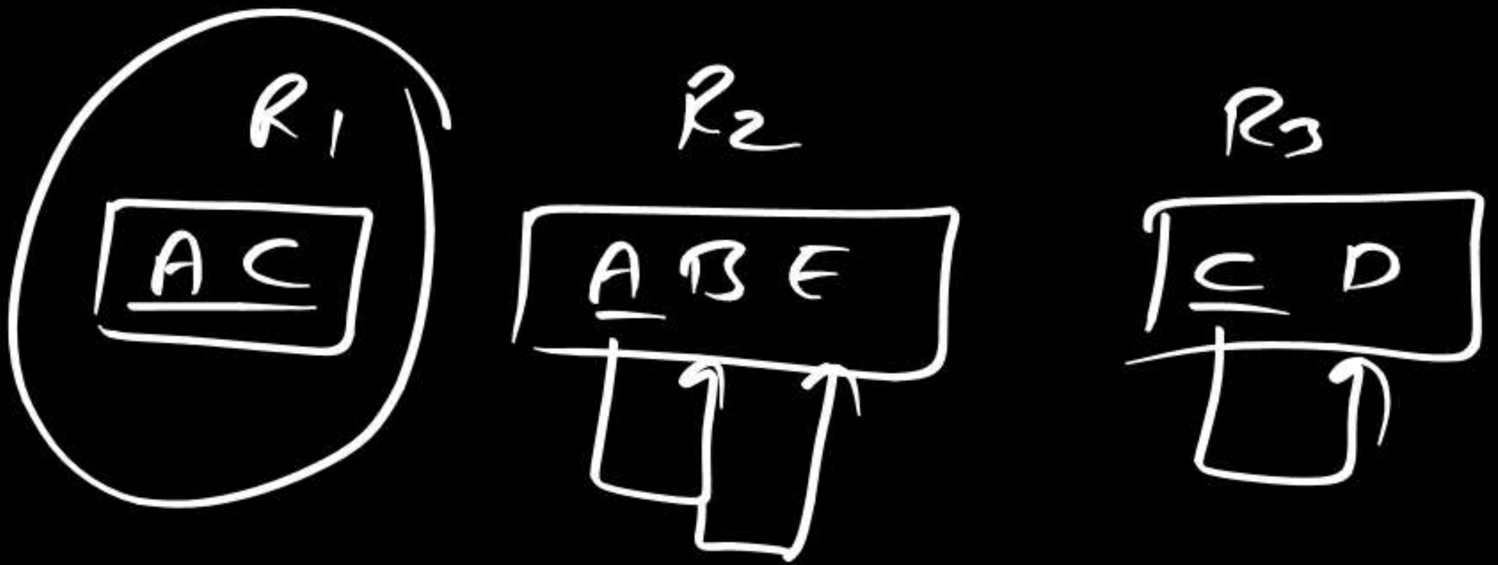
<u>A</u>	B	E

R_2

<u>C</u>	D

Whenever we try to Join R_1 & R_2 then we getting Suburious Tuple (Extra Tuple)

Bcz No Common Attribute, its' Lossy Join



$R_1 \cap R_2 \Rightarrow (A)^+ = (ABE)$ super key R_2

$R_{12}(ABCE) \cap R_3(CD) \Rightarrow [C]$

$[C]^+ = [CD]$ super key of R_3

Lossless + D.P

3NF Decomposition

FD 3NF Violate

Non key $\longrightarrow$ Non key

3NF ✗

3NF Decomposition

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

3NF Decomposition R(AB)

$B \to C$

R_2 (B C)

$R(A \to B,\ B \to C)$

C.K : [A]

Check 2NF ✓

Check 3NF : $B \to C$ Voilate 3NF

3NF Decomposition

$R_1(\underline{A}\ B)$

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

$B \to C$

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

$R_1(AB)$ $R_2(BC)$

$R_1(AB) \cup R_2(BC) \Rightarrow ABC$

Lossless? $R_1(AB) \cap R_2(BC)$

$(B)^+ = \{BC\}$ S.K of R_2

Lossless + DP

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

A	B	C
1	x_1	A
2	x_1	A
3	x_1	A
4	x_1	A
5	y_1	B
6	y_1	B
7	y_1	B
8	y_1	B

10,000

5000

5000

3NF

A	B
1	x_1
2	x_1
3	x_1
4	x_1
5	y_1
6	y_1
7	y_1
8	y_1

B	C
x_1	A
y_1	B

R(ABCDEFGHIJ) {AB→C, A→DE, D→IJ, B→F, F→GH}

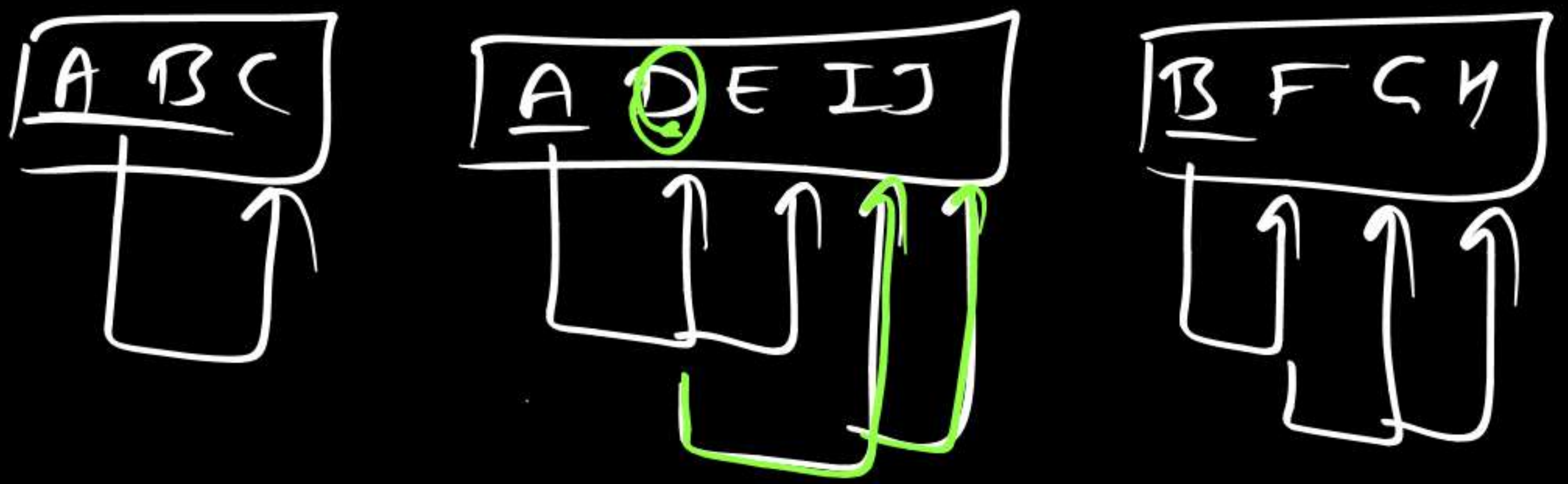
Candidate key = {AB

Non key Attribute = {B, C, D, E, F, G, H, I J}

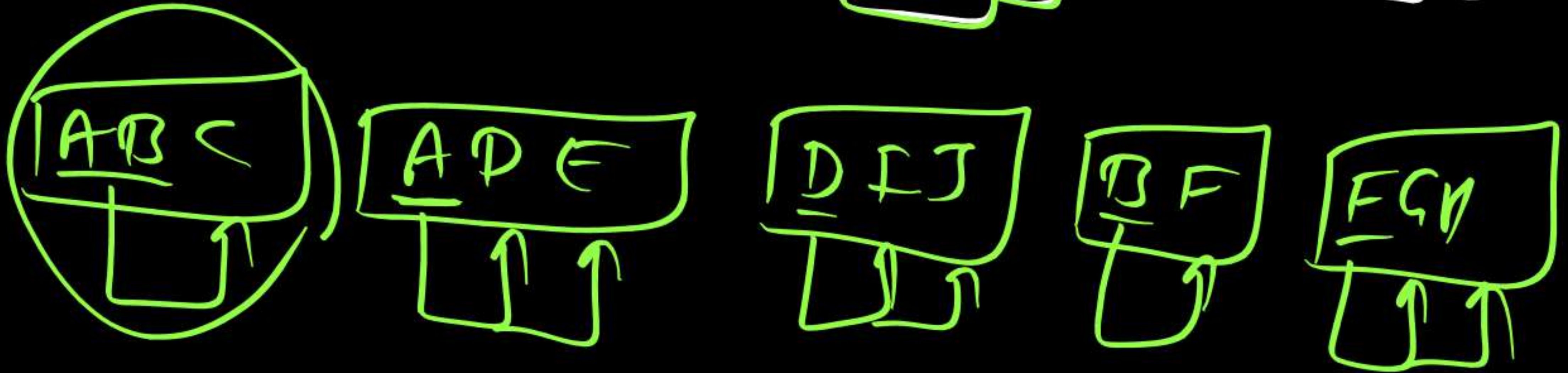
2NF Decomposition

$(A)^+ = \{ADEIJ\}$

$(B)^+$ {BFGH



3NF Decom ⇒



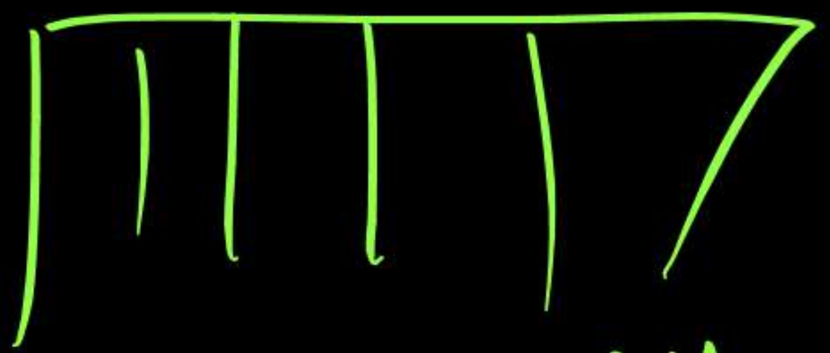
$R(ABCDEFGHIJ)$ $\{AB \to C,\ A \to DE,\ D \to IJ,\ B \to F,\ F \to GH\}$

F.Name M.Name

Name Roll Grade

Roll	Name	F.N	M.N	Address	Sem	Grade	- - - - .
		x	y	—	I		
		"	"	"	II		
		"	"	"	III		
		"	"	"	IV		
		"	"	"	V		
		"	"	"	VI		
		"	"	"	VII		
					VIII		

50 Attribute



10 Atr

Sen Sub CGPA

40 Attribute

Unnessary

7 Time Repeat.

$40 \times 7 = 280$ entries

1 Lakh ⇒ 280 Lakh

2.8 Crore

3NF Decomposition

Q R(ABCDEF) [AB → C, C → D, D → E, E → F]

Candidate key = AB

3NF Violation

$C \rightarrow D$

$D \rightarrow E$

$E \rightarrow F$

$R(ABC\cancel{D}\cancel{E}\cancel{F})$

R_2 $\boxed{C\ D}$

R_3 $\boxed{D\ E}$

R_4 $\boxed{E\ F}$

$R_1(\underline{AB}C)$

$R_2(\underline{C}D)$

$R_3(\underline{D}E)$

$R_4(\underline{E}F)$

Q

$R(ABCDE) \quad \{AB \rightarrow C, C \rightarrow D, B \rightarrow E\}$

Decompose into 2NF, 3NF, BCNF

BCNF

Voilate

$B \rightarrow C$

$C \rightarrow D$

$R(\_\ \_\ \cancel{C}\ \_\ \_\ \_)$

$\boxed{\underline{B}\ \ C}$

2NF

~~3NF~~

BCNF Decomposition

Candidate key = (A)

Q R(ABCDE) {A → B, B → C, C → D, D → E}

Consider New Updated Relation

BCNF Decomposition

R(AB~~C~~D~~E~~)

① B → C

× C → D

② D → E

R_2(BC)

R_3(DE)

R_1(ABD)

R_2(BC)

R_3(DE)

BCNF Decomposition

Candidate key = (A)

Q R(ABCDE) $\{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E\}$

BCNF Decomposition

R(ABCDE)

② $B \rightarrow C$

① $C \rightarrow D$

$D \rightarrow E$

$R(AB\cancel{C}\cancel{D}E)$

$R_2(CD)$

$R_3(BC)$

$R_1(ABE)$
$R_2(CD)$
$R_3(BC)$

BCNF Decomposition

Candidate key = (A)

Q R(ABCDE) $\{A \to B, B \to C, C \to D, D \to E\}$

BCNF Decomposition

R(ABCDE)

(3) $B \to C$

(2) $C \to D$

(1) $D \to E$

$R_2(DE)$

$R_3(CD)$

$R_4(BC)$

$R_1(AB)$
$R_2(DE)$
$R_3(CD)$
$R_4(BC)$

BCNF Decomposition

Q R(ABCD) {AB → CD, D → A}

Candidate key = (AB, DB)

C.K: BD

C.K: D

R_2

R_1 $\boxed{\underline{B}C\underline{D}}$ $\boxed{\underline{D}\ A}$

BCNF Decomposition

$D \rightarrow A$

R($\cancel{A}$BCD)

R_2(D A)

$R_1(BCD) \cap R_2(DA) = D$

$(D)^+ = (DA)$ super key of R_2

lossless

But Dep. Not preserved

Q

R(ABCDEFGHIJ) $\{AB \rightarrow C, A \rightarrow DE, B \rightarrow F, F \rightarrow GH, D \rightarrow IJ\}$

ABC ADE BF FGH DIJ

BCNF

+ lossless

+ Dep Preserved

Q

Relation R is decomposed using a set of functional dependencies, F, and relation S is decomposed using another set of functional dependencies, G. One decomposition is definitely BCNF, the other is definitely 3NF, but it is not know which is which. To make a guaranteed identification, which one of the following tests should be used on the decompositions? (Assume that the closures of F and G are available).

[2002: 2 Marks]

(A) Dependency-preservation

(B) Lossless-join

(C) BCNF definition

(D) 3 NF definition

Q

Which of the following relational schema with given FD's follows is/are in BCNF?

A. R(ABCDE) and FD's are $\{A \to B, B \to C, C \to D, D \to E, C \to A\}$

B. R(ABCDE) and FD's are $\{A \to B, C \to D, D \to E\}$

C. R(ABCD) and FD's are $\{A \to B, B \to C, C \to D, D \to A\}$

D. R(ABCD) and FD's are $\{A \to B, B \to C, C \to A\}$

Q

Consider the following Relation:

R(ABCDEFG) with FD set of Relation R $\{A \rightarrow B, C \rightarrow D, E \rightarrow FG\}$.
What is the minimum number of relations required to decompose into BCNF which satisfy lossless join and Dependency preserving decomposition ________

Transaction :

Read(A)

$A = A - 100$

Write(A)

Read(B)

$B = B + 100$

Write(B)

Transaction Concept: ACID Properties, Schedule types, Serializable Schedule (Conflict & View Serializable)

Transaction Concept

- A transaction is a unit of program execution that accesses and possibly updates various data items.
- E.g. Transaction to transfer Rs 500 from account A to account B:

1. read (A)
2. A : = A – 500
3. write(A)
4. read(B)
5. B: = B + 500
6. write(B)

1 Transaction

ACID Properties

- A transaction is a unit of program execution that accesses and possibly updates various data items. To preserve the integrity of data the database system must ensure:

A. Atomicity

C. Consistency

I. Isolation

D. Durability

A
C
I
D

Atomicity : (Full or None)

Recovery ⇒ ROLLBACK

Transaction log.

UNDO All Modification

Consistency :

Before

A : 4000

B : 2000

$\overline{6000}$

$\curvearrowright 500$

After

A : 3500

B : 2000

$\overline{6000}$

Isolation: Concurrent Execution of Two or More should be equal to Any Serial Schedule.

Durability: Any change in the DB must persist for long period of time.

ACID Properties

- **Atomicity:** Either all operations of the transaction are properly reflected in the database or none are.
- **Consistency:** Execution of a transaction in isolation preserves the consistency of the database.
- **Isolation:** Although multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions. Intermediate transaction results must be hidden from other concurrently executed transactions.

- That is, for every pair of transactions T_i and T_j, it appears to T_i that either T_j, finished exection before T_i started, or T_j started execution after T_i finished.

- **Durability:** After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

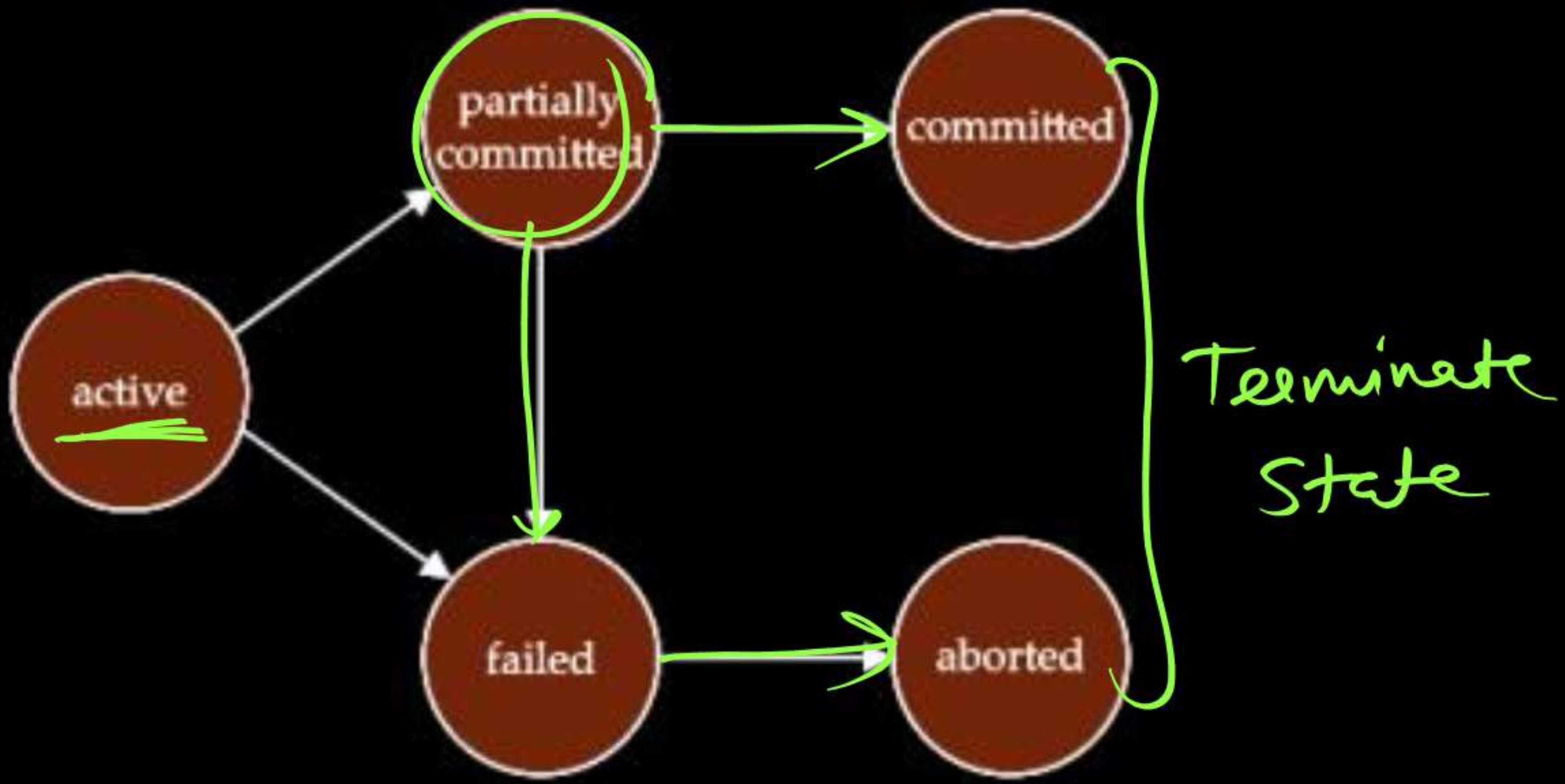
Transaction State

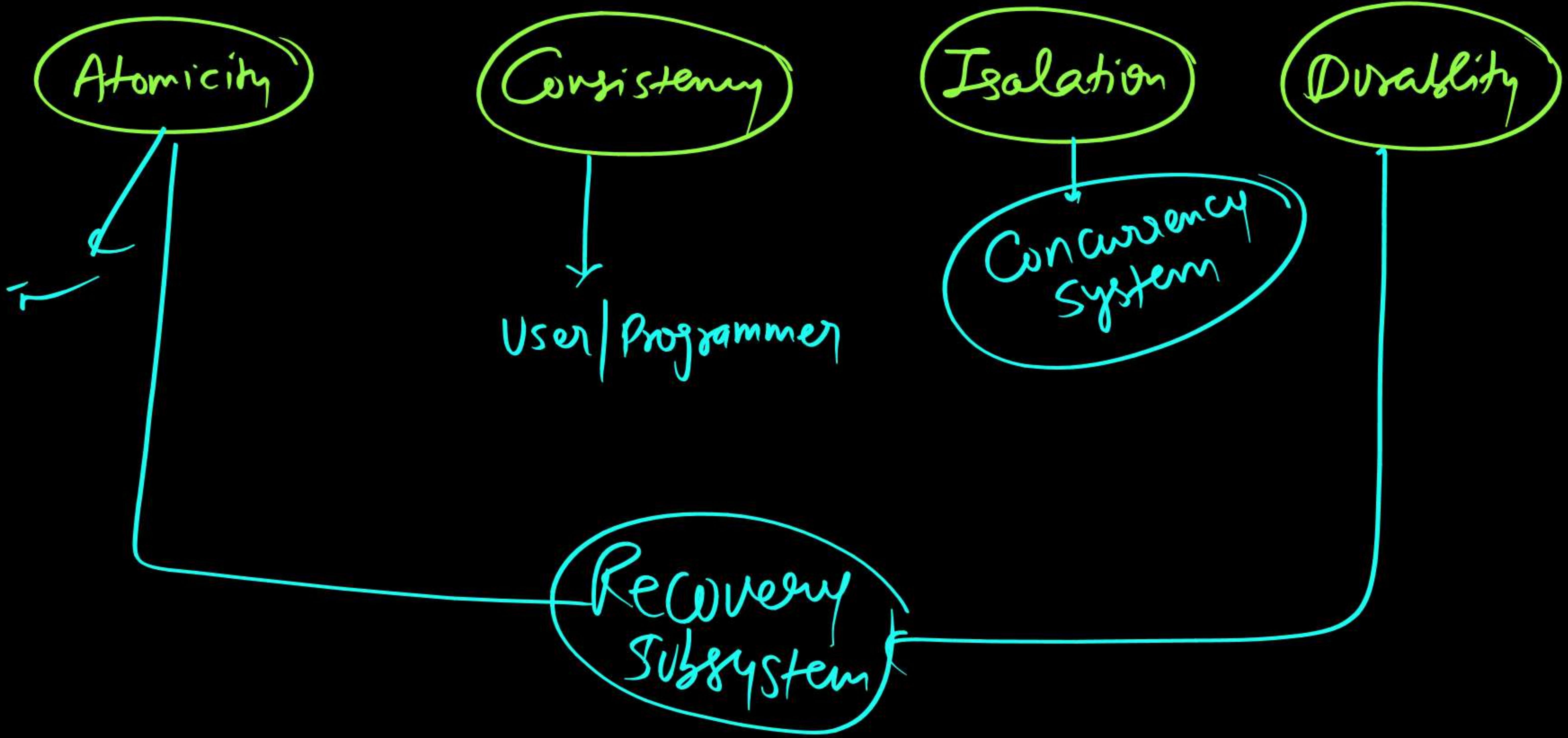
- Active : the initial state; the transaction stays in this state while it is executing.
- Partially committed : after the final statement has been executed.
- Failed: after the discovery that normal execution can no longer proceed.
- Aborted: after the transaction has been rolled back and the database restored to its state prior to the start of the transaction. Two options after it has been aborted:

- Restart the transaction
  - Can be done only if no internal logical error
- Kill the transaction

- Committed: After successful completion.

Transaction State (Cont.)




Atomicity
Consistency
Isolation
Durablity
User/Programmer
Concurrency system
Recovery subsystem

Schedule

Time Order Sequence of Two or More Transaction

Serial Schedule

Execution of One Transaction Completely then another will start

Non-Serial Schedule

Interleaved execution of Two or More Transaction

Serial

T_1	T_2

OR

T_1	T_2

$\langle T_1 \; T_2 \rangle$

T_1 followed by T_2

$\langle T_2 \; T_1 \rangle$

T_2 followed by T_1

Non serial

T_1	T_2

T_1	T_2

T_1	T_2

& Many more

$2! = 2$ serial schedule

2 Transaction ⇒ Serial Schedule
$\langle T_1 \; T_2 \rangle$
$\langle T_2 \; T_1 \rangle$

3 Transaction ⇒ $3! \Rightarrow 6$ Serial Schedule

$\langle T_1 \; T_2 \; T_3 \rangle$
$\langle T_1 \; T_3 \; T_2 \rangle$
$\langle T_2 \; T_1 \; T_3 \rangle$
$\langle T_2 \; T_3 \; T_1 \rangle$
$\langle T_3 \; T_1 \; T_2 \rangle$
$\langle T_3 \; T_2 \; T_1 \rangle$

n Transaction ⇒ $n!$ Serial Schedule

Schedules

- Schedule: a sequences of instructions that specify the chronological order in which instructions of concurrent transactions are executed.
  - A schedule for a set of transactions must consist of all instructions of those transactions
  - Must preserve the order in which the instructions appear in each individual transaction.
- A transaction that successfully completes its execution will have a commit instructions as the last statement
  - By default transaction assumed to execute commit instruction as its last step

- A transaction that fails to successfully complete its execution will have an abort instruction as the last statement.

Let T_1 transfer 100 Rs from A to B, and T_2 transfer 10% of the balance from A to B.

Schedule 1

A = 2000
B = 3000
5000

$T_1 : A \xrightarrow{100Rs} B$

$T_2 : A \xrightarrow{10\%} B$

T_1	T_2
read (A)	
A: = A – 100	
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	
	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
	read (B)
	B := B + temp
	write (B)
	Commit

$S_1 < T_1\ T_2 >$

Serial schedule in which T_1 is followed by T_2:

Schedule 2

A = 2000
B = 3000
5000

T_1	T_2
	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
	read (B)
	B := B + temp
	write (B)
	Commit
read (A)	
A: = A – 100	
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	

$S_2 < T_2\ T_1 >$

serial schedule where T_2 is followed by T_1

A=2000
B=3000
5000

Schedule 3

T_1	T_2
read (A)	
A: = A – 100	
write (A)	
	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
read (B)	
B: = B + 100	
write (B)	
commit	
	read (B)
	B := B + temp
	write (B)
	Commit

C_1

A=2000
B=3000
5000

Schedule 4

T_1	T_2
read (A)	
A: = A – 100	
	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
	read (B)
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	
	B := B + temp
	write (B)
	Commit

~~C_2~~

THANK YOU!