COMPUTER SCIENCE

Database Management System

Transaction & Concurrency Control

Transaction Concept Part-02

Lecture_2

Vijay Agarwal sir

TOPICS TO BE COVERED

01 NF Decomposition

02 Transaction Concept

Partial FD & Full FD.

RDBMS Concept

FD Concept & its type.

Attribute closure

Key Concept

Super key

Candidate key

Finding Multiple C.K

Membership set

Equality b/w 2 FD set

Minimal cover

finding # Super keys

FD closure.

Lossless Join

Dependency Preserving Decomposition

Normal Form

1NF 2NF 3NF BCNF

& its Decomposition.

2NF:

Codd

$R_1(A\cancel{B}CDE)$

$A \to B$

$R_2(\underline{A}B)$

$B \to D$

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

GATE

(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.

$\boxed{\text{Ans (A)}}$

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

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

(B) No relation can be in both BCNF and 3NF.

(C) Second normal form(2NF) does not have partial dependency. from the Non key Attribute.

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

$\boxed{\text{(A) \& (C).}}$

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 $B \rightarrow G$.

[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.

$B \rightarrow G$

Proper subset of CK → Non key Attribute.

Not in 2NF

$\boxed{\text{Ans (D)}}$

Q

Consider the following statements: [MSQ]

Incorrect ← S_1: If every attribute is prime attribute in R, then Relation R will always be in BCNF. [3NF But may Not in BCNF]

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

Binary Relation → BCNF, 3NF, 2NF, 1NF

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

Incorrect ← 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

$\boxed{\text{Ans (A) \& (D).}}$

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

Candidate key = {A, C, B}

D → E

Non key   Non key

D is Not Super key

OR

E : Not Prime Attribute.

Q

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

[2004: 2 Marks]

$$\text{name, courseNo} \rightarrow \text{grade}$$

$$\text{RollNo, courseNo} \rightarrow \text{grade}$$

$$\text{name} \rightarrow \text{rollNo}$$

$$\text{rollNO} \rightarrow \text{name}$$

The highest normal form of this relation scheme is

(A) 2 NF

(B) 3 NF

(C) BCNF

(D) 4 NF

BCNF Decomposition

Q R(ABCDEFG) {A → BF, F → DEG, ~~A → D~~}

Candidate key = AC

Check BCNF ?

A → BF
F → DEG

Violate BCNF

BCNF Decomposition

R(ABCDEFG)

① A → BF
F → DEG

ABF

ACDEG   ABF

BCNF Decomposition.

① F → DEG
② A → BF

R_1(ABCDEFG)

R_2(FDEG)

R_3(ABF)

R_1(AC)
R_2(FDEG)
R_3(ABF)

Q

~~Directly~~

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

Directly BCNF

Candidate key = [AB]

Check BCNF

A → DE

B → F

F → GH

D → IJ

① D → IJ

② F → GH

③ B → F

④ A → DE

R_1(ABC~~DEFGHIJ~~)

R_1 ABC

R_2 DIJ

R_3 FGH

R_4 BF

R_5 ADE

BCNF ✗

Lossless

✗ D.P

Step by step.

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

Candidate key = [AB], Non key Attribute = [C, D, E, F, G, H, I, J]

OR

Check 2NF ?

$A \to DE$, $B \to F$

Violate 2NF

2NF Decomposition

$(A)^+ = [ADEIJ]$

$(B)^+ = [BFGH]$

R(ABC~~DEFGHIJ~~)

ABC	ADEIJ	BFGH

Check 3NF ? $D \to IJ$, $F \to GH$

R_1	R_2	R_3	R_4	R_5
AB C	A D E	D I J	B F	F G H

R is in 3NF

Check BCNF ?

YES R is in BCNF

Bcz Every determinant is a Super key.

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

Ans (C).

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

MSQ → one option or more than one option.

D: Not Super key or E: Not Prime Attribute

Non key → Non key

2NF Not in 3NF

(A) 2NF R(ABCDE) and FD's are {A → B, B → C, C → D, D → E, C → A}

↳ Candidate key = [A, C, B]

(B) 1NF R(ABCDE) and FD's are {A → B, C → D, D → E}

↳ Candidate key = [AC] → Violate 2NF.

(C) BCNF R(ABCD) and FD's are {A → B, B → C, C → D, D → A}

Candidate keys = [A, B, C, D]

Every X → Y, ↳ X: super key } BCNF

(D) 3NF R(ABCD) and FD's are {A → B, B → C, C → A}

Candidate key = [AD, BD, CD]

A → B, B → C, C → A } X → Y, X is Not Super key But Y is Prime Attribute So R is in 3NF But Not BCNF.

$\boxed{\text{Ans (C)}}$

↳ BCNF

Q

Consider the following Relation:

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

Candidate key = ACE.

Check BCNF ?

A→B
C→D
E→FG.

Violate BCNF

BCNF Decomposition.

(1) A→B
(2) C→D
(3) E→FG

$R_1(A\cancel{B}C\cancel{D}E\cancel{F}\cancel{G})$ R_1 $\boxed{ACE}$

R_2 $\boxed{AB}$

R_3 $\boxed{CD}$

R_4 $\boxed{EFG}$

BCNF + Lossless + D.P

Q.

Consider the following four relational schemas. For each schema, all non-trivial functional dependencies are listed. The underlined attributes are the respective primary keys.

Schema I: Registration (<u>rollno</u>, courses)

BCNF

Field 'courses' is a set-valued attribute containing the set of courses a student has registered for.

Non-trivial functional dependency:

rollno → courses

Schema II: Registration (<u>rollno, courseid</u>, email)

3NF. Not in BCNF

Non-trivial functional dependencies:

rollno, courseid → email

email → rollno → P.A

Schema III: Registration (<u>rollno, courseid</u>, marks, grade)

2NF Not in 3NF

Non-trivial functional dependencies:

rollno, courseid → marks, grade

marks → grade → N.K → N.K

Schema IV: Registration (<u>rollno, courseid</u>, marks, credit)

In 1NF Not in 2NF

Non-trivial functional dependencies:

rollno, courseid → credit

courseid → credit → Subset C.K → N.K

Which one of the relational schemas above is in 3NF but not in BCNF?

Ans (B).

(A) Schema I

(B) Schema II

(C) Schema III

(D) Schema IV

[MCQ: 2018: 2M]

Q. A database of research articles in a journal uses the following schema.

(VOLUME, NUMBER, STARTPAGE, ENDPAGE, TITLE, YEAR, PRICE)

The primary key is (VOLUME, NUMBER, STARTPAGE, ENDPAGE) and the following functional dependencies exist in the schema.

Not in 2NF

OLD Design

FD I (VOLUME, NUMBER, STARTPAGE, ENDPAGE) → TITLE

FD II (VOLUME, NUMBER) → YEAR

FD III (VOLUME, NUMBER, STARTPAGE, ENDPAGE) → PRICE.

Proper subset of C.K → Non key Attribute

Violation of 2NF

The database is redesigned to use the following schemas.

New Design

R_1 (VOLUME, NUMBER, STARTPAGE, ENDPAGE, TITLE, PRICE)

R_2 (VOLUME, NUMBER, YEAR)

Which of the weakest normal form that the new database satisfies, but the old one does not? [MCQ: 2016: 1M]

(A) 1NF

(B) 3NF

(C) 2NF

(D) BCNF

Ans (C) 2NF

Q. Consider a relational table R that is in 3NF, but not in BCNF. Which one of the following statements is TRUE?

[MCQ: 2020-2M]

(A) R has a nontrivial functional dependency X→A, where X is not a superkey and A is a non-prime attribute and X is a proper subset of some key.

(B) R has a nontrivial functional dependency X→A, where X is not a superkey and A is a non-prime attribute and X is not a proper subset of any key.

(C) A cell in R holds a set instead of an atomic value.

(D) R has a nontrivial functional dependency X→A, where X is not a superkey and A is a prime attribute.

Q.

Consider a relation R (A, B, C, D, E) with the following three functional dependencies.

$AB \rightarrow C$; $BC \rightarrow D$; $C \rightarrow E$;

The number of superkeys in the relation R is _____.

[NAT:2022-1M]

Candidate key = AB

$\#$ Super key $= 2^{5-2}$

$= 2^3$

$= 8$ Super key

$\underline{AB}\ \underbrace{C\ D\ E}$

$\Downarrow$

$2^3 = 8$ S.K Ans

Multivalued FD : $[X \to\to y]$

When we have 2 or more than two multivalued Attribute then Multivalued FD.

Single Valued FD $(X \to y)$

If $t_1.X = t_2.X$ then $t_1.y = t_2.y$ must be same.

Not in syllabus

Multivalued FD $[X \rightarrow\rightarrow Y]$

RollNo.	Sname	Book
1	X/Y	A/B

X	Y	Z
Rollno	Sname	Book
1	X	A
1	X	B
1	Y	A
1	Y	B

t	X	Y	Z
t_1	X_1	Y_1	Z_1
t_2	X_1	Y_1	Z_2
t_3	X_1	Y_2	Z_1
t_4	X_1	Y_2	Z_2

$X \rightarrow\rightarrow Y$ exist

$t_1.X = t_2.X = t_3.X = t_4.X$

&&

$t_1.Y = t_2.Y$ && $t_3.Y = t_4.Y$

&&

$t_1.Z = t_3.Z$ && $t_2.Z = t_4.Z$

Transaction

Read(A)

A = A − 1000

Write(A)

Read(B)

B = B + 1000

Write(B)

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:

A=2000

500 Rs

A to B

1. read (A)
2. A : = A – 500
3. write(A)
4. read(B) 4000
5. B: = B + 500
6. write(B)

Read (A) : Access the Data Item A from DB.

Write (A) : Updating the Data Item A.

<u>Commit</u> : Successfull Completion of the transaction.

To Maintain Integrity Transaction Must Satisfied ACID Property.

A : Atomicity.

C : Consistency.

I : Isolation.

D : Durablity.

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

ACID

Atomicity

(FULL or NONE) : Either execute All operation of the transaction successfully (FULL) OR None of them.

Reason for Transaction failure :

- Power failure
- S/w crash
- H/w crash
- System crash etc (N/w Issue....)

Due to Any of these Reasons if Atomicity is getting failed then Recovery Management Component are there.

If transaction is Failed before Commit then

Recovery Management Component ROLLBACK (UNDO ALL MODIFICATION) the transaction. (with the Help of log's File)

Log's file : Log's Contain all the Activity of the transaction.

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.

② Consistency

Before & After the transaction DB Must be Consistent.

Before

A : 4000

B : 2000

$\oplus$ 6000

A to B 500

AFTER

A : 3500

+ B : 2500

6000

③ **Isolation** : Concurrent Execution of 2 or more Transaction equal to Any Serial Schedule.

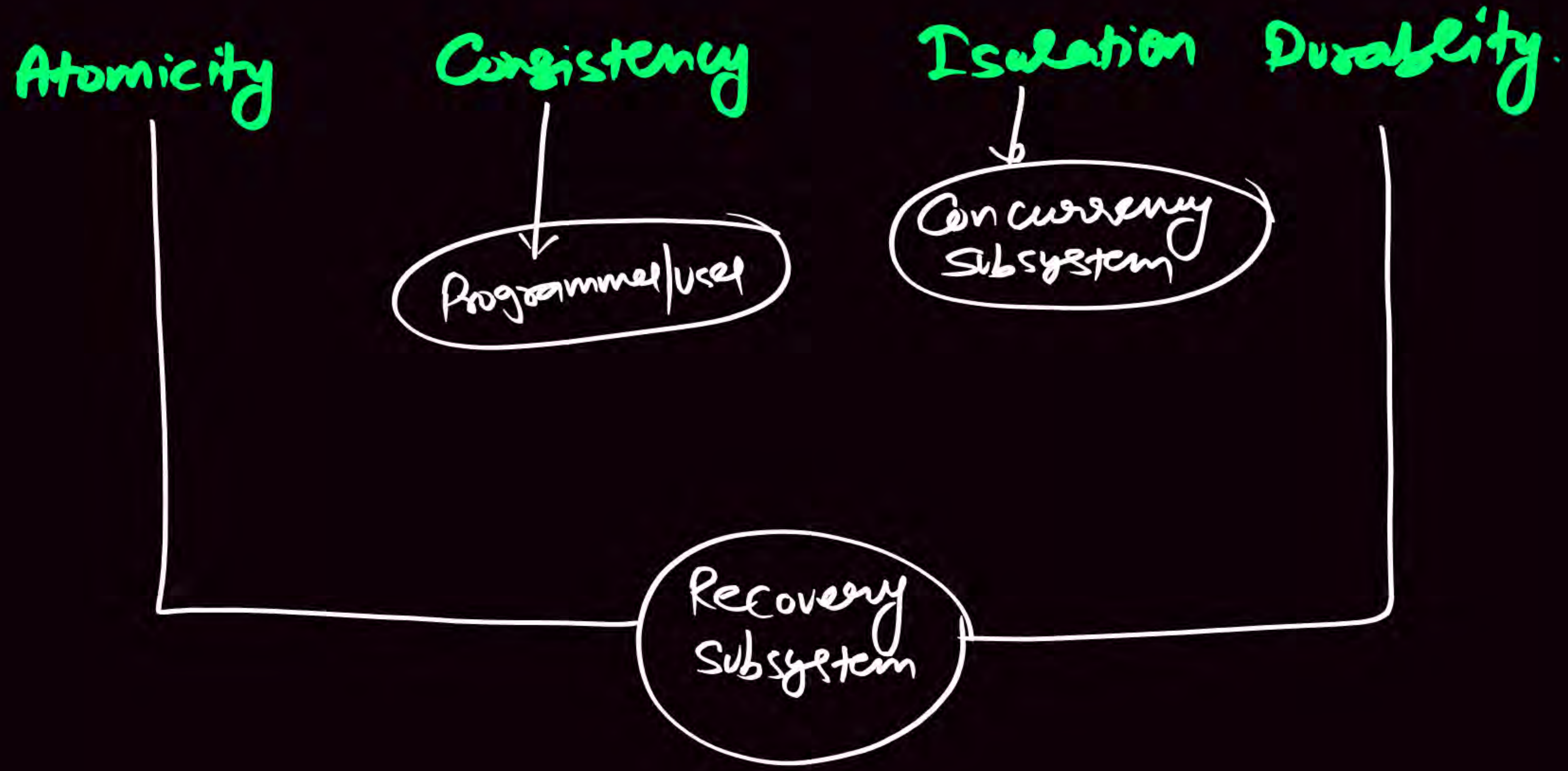
④ **Durability** : Any change in the Database Must Persist for long Period of time.

OR

Database Must be able to Recover Under Any Case of failure.

- 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.


Atomicity
Consistency
Isolation
Durability
Programmer/user
Concurrency subsystem
Recovery subsystem

Consistency is taken Care by User/Programmer.

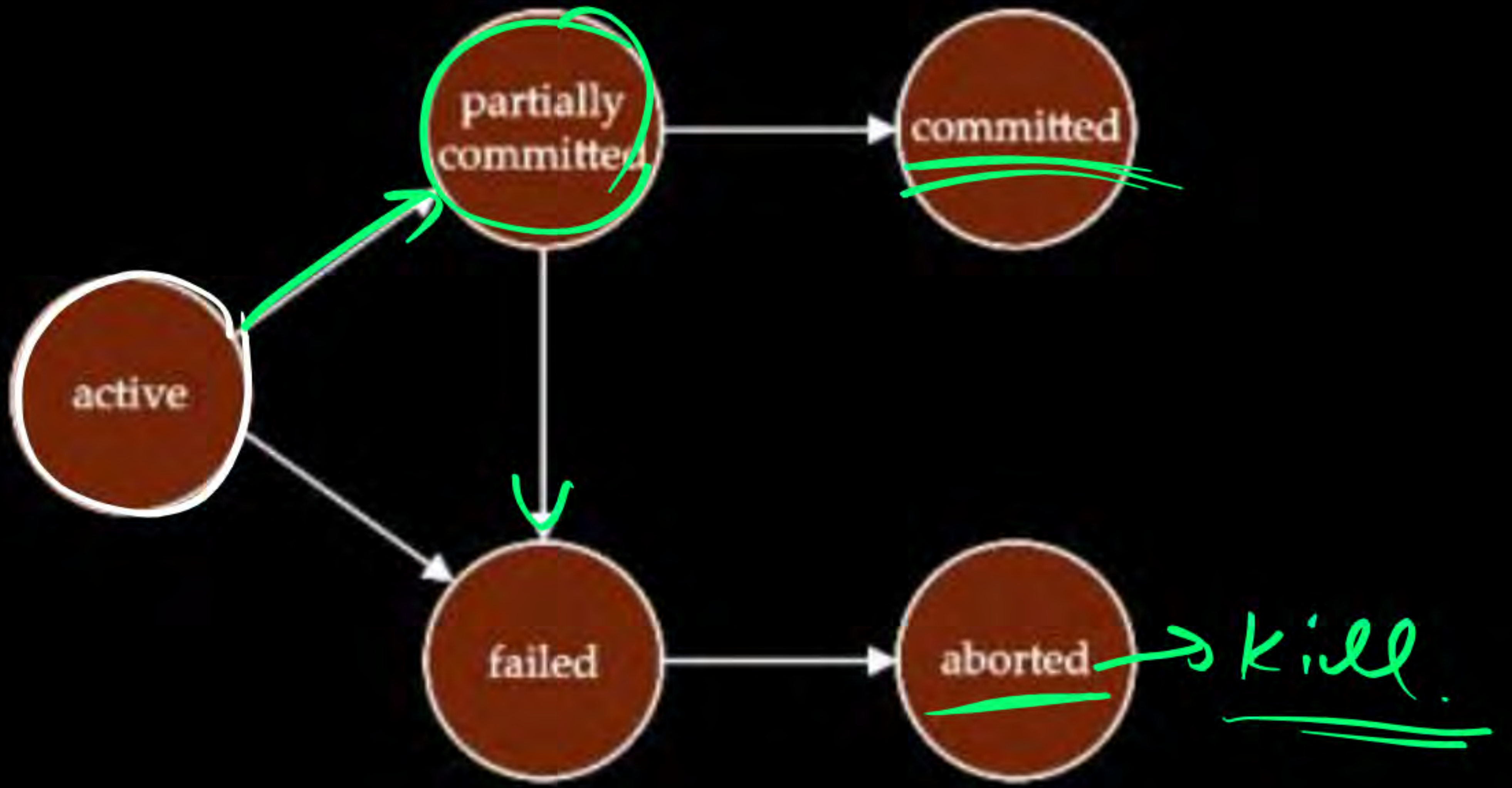
Transaction State

- 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.)



SCHEDULE

Time Order sequence of two or more Transaction.

SCHEDULE

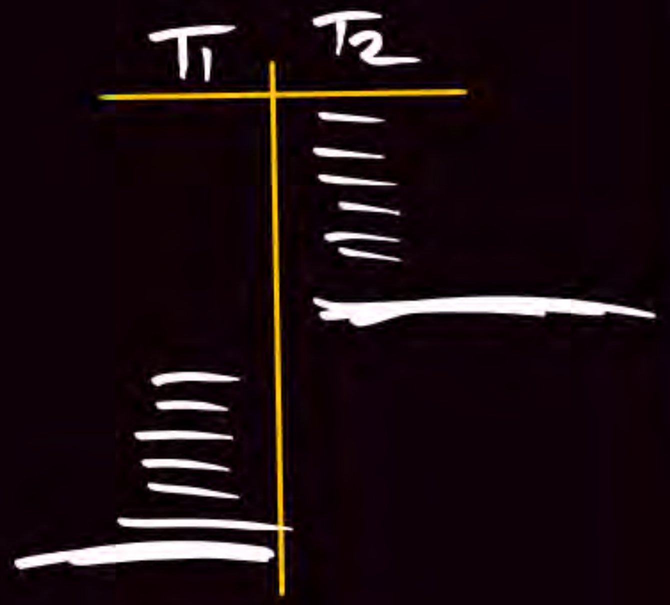
- Serial schedule
- Non serial schedule.

Serial Schedule

Execution of One Transaction Complete then another will Start.

$\langle T_1\ T_2 \rangle$

T_1 followed by T_2



$\langle T_2\ T_1 \rangle$

T_2 followed by T_1

Non Serial Schedule (concurrent execution)

Interleaved execution of Two or More Transaction.

& Many More.

Note

$= 2! = 2$

If 2 Transaction Total # serial schedule = 2

(i) $\langle T_1\ T_2 \rangle$; T_1 followed by T_2

(ii) $\langle T_2\ T_1 \rangle$; T_2 followed by T_1

Note If n transaction then n! serial schedule

Q) If 3 Transaction the $3! \Rightarrow 6$ serial schedule

$$\left.\begin{array}{l} \langle T_1\ T_2\ T_3 \rangle \\ \langle T_1\ T_3\ T_2 \rangle \\ \langle T_2\ T_1\ T_3 \rangle \\ \langle T_2\ T_3\ T_1 \rangle \\ \langle T_3\ T_1\ T_2 \rangle \\ \langle T_3\ T_2\ T_1 \rangle \end{array}\right\} \text{6 serial schedule.}$$

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.

A: 2000
+ B: 3000
5000

Schedule 1

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:

A: 2000
+ B: 3000
5000

Schedule 2

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)
write (A) read (B) B: = B + 100 write (B) commit	
	read (B) B := B + temp write (B) Commit

C_2

Any Doubt ?

THANK YOU!