Database

6-10-2025

chapter 6

ER model (Entity-Relationship)

The ER data model employs three basic concepts:

→ Entity sets / table
→ Relationship sets
→ attributes.

The ER diagram, which can express the overal logical structure of a database graphically.

Lab mid

Lab final

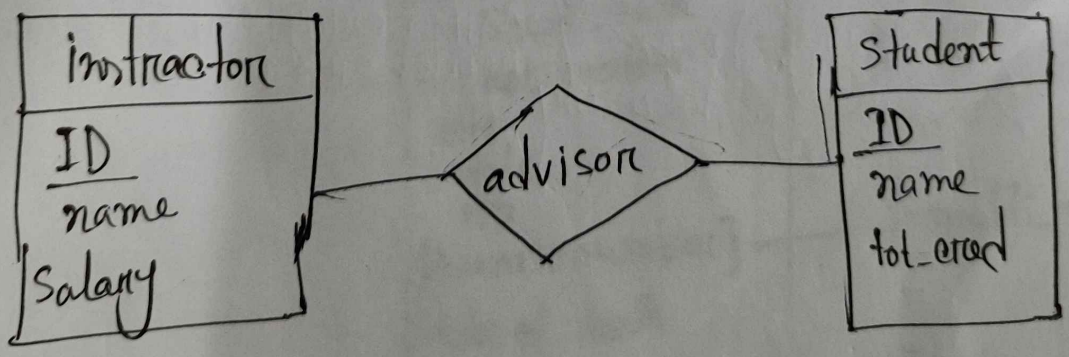
1. Proposal - 5
2. Schema diagram - 5
3. E-R diagram - 6
4. Constraints of the table - 6
5. Implementation - 8

} 7

6. Report - 7
7. Presentation - 10

Representing Relationship sets via E-R Diagrams

→ Diamonds represent relationship sets.

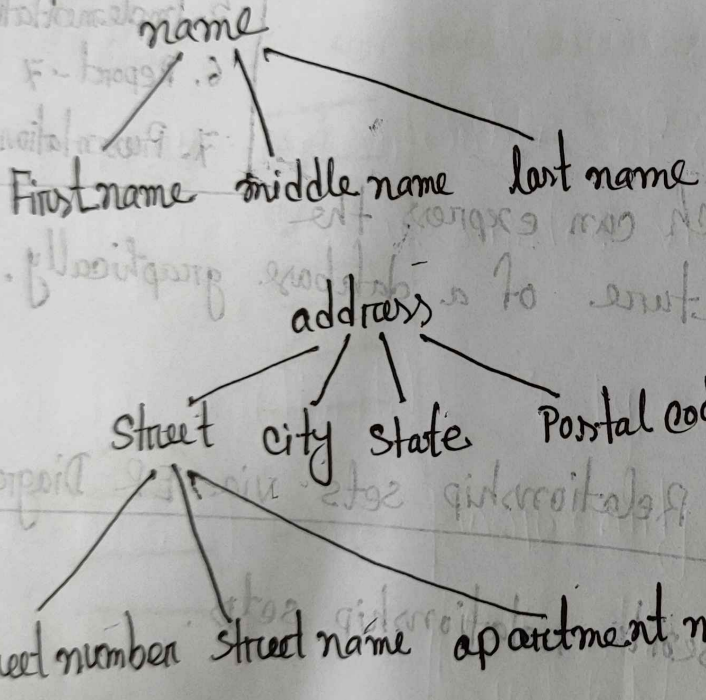


Complex attributes

1. simple : যে attribute এর value কে sub part এ devide করা Possible না। age, ct_marks.

Composite : Sub part এ devide করা Possible.
name, address

Composite attributs allow us to divided attributes into subparts.



Database

7-10-2025

একজনের জন্য ১টাই value accept করে। average CGPA..

multiple value accept করে। reg_courses, contact_number.

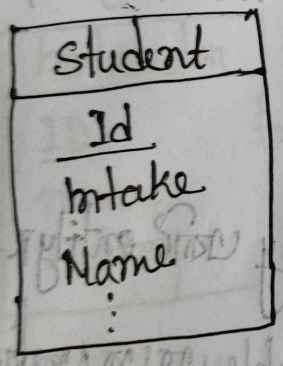
⇒ Single valued and multivalued attributes

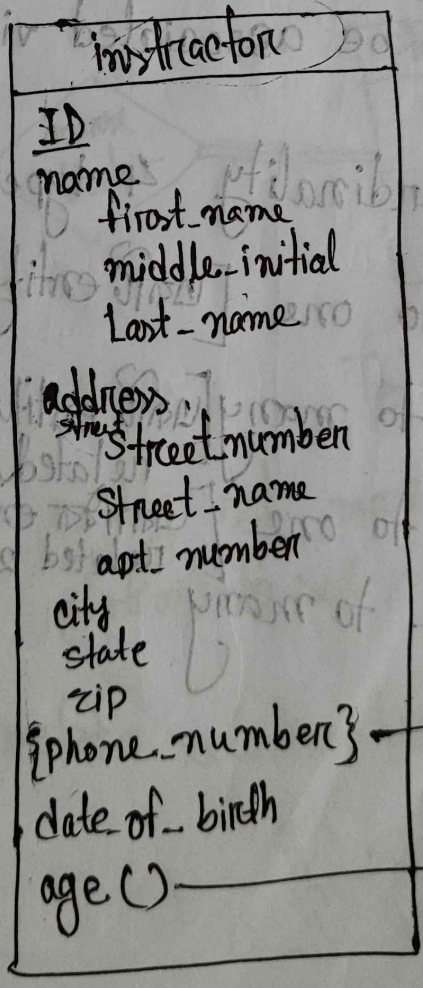
⇒ Derived attributes (table exist করে না তবে অন্য কোনো attribute এর help নিয়ে value পাওয়া possible. এরে attribute গুলো Derived attributes)

Single valued

ID	DOB	Age
1	10.10.1998	
2	5.5.1999	

Derived attribut

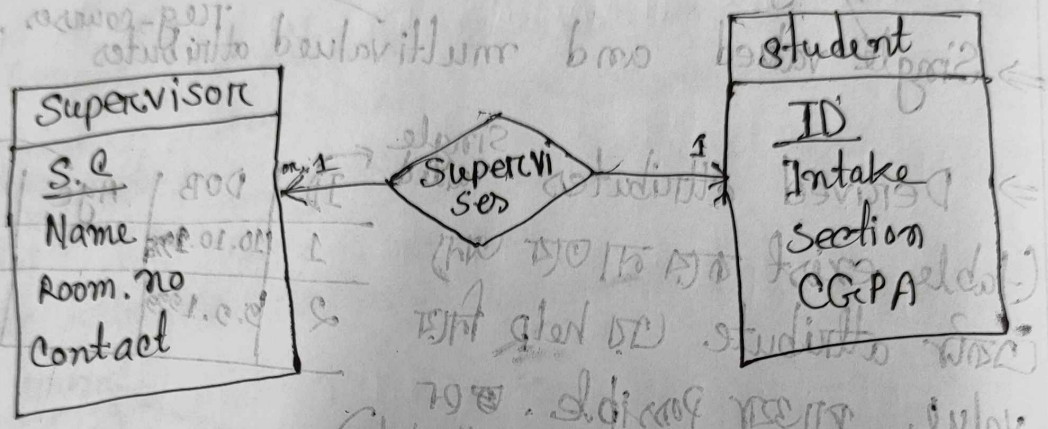




multi_valued

Derived

Mapping Cardinality Constraints



Express the number of entities to which another entity can be associated via a relationship set.

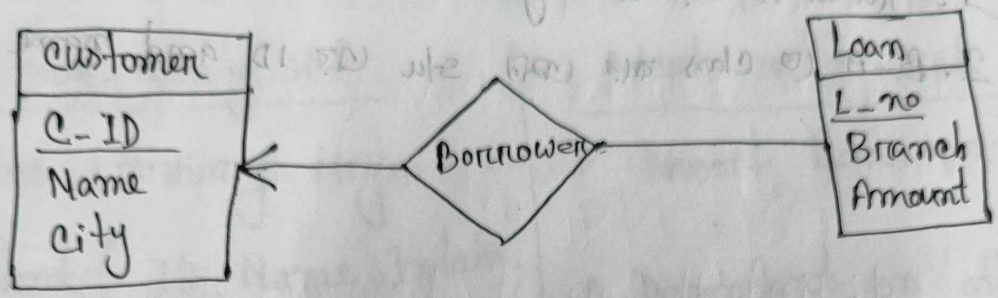
mapping cardinality 4 types :

(↔) → one to one [একটি entity exactly একটি entity র সাথে related]

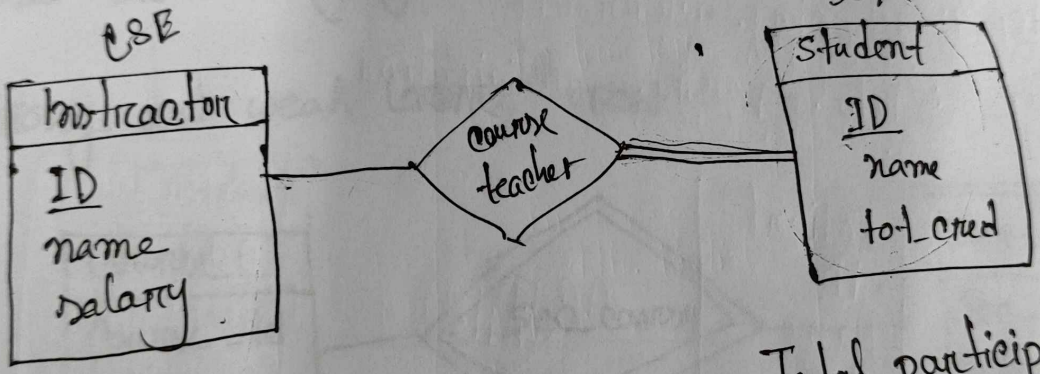
(←)/(→) one to many [একটি entity অন্য table এর এক/একাধিক entity র সাথে related]

(→) → many to one [একাধিক entity অন্য table এর একটি entity র সাথে related হবে।]

(—) → many to many



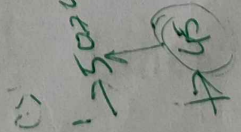
Total and Partial Participation



Total participation → (=) double line

Total participation:
Every entity in the entity set participates in at least one relationship in the relationship set.

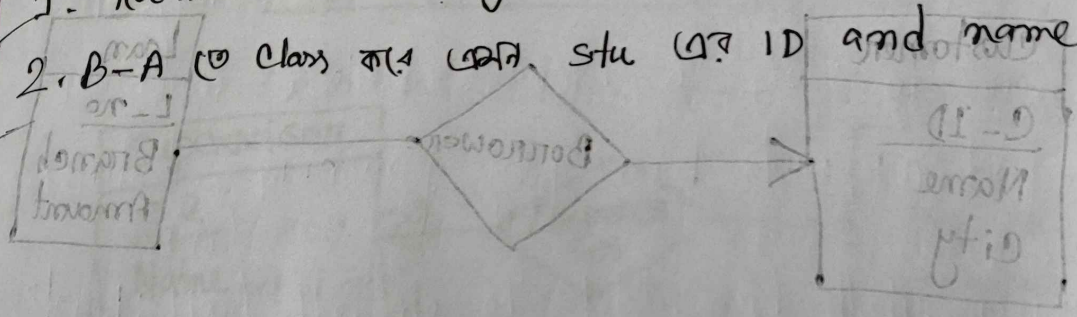
Partial participation: [illegible] relationship ↳ Participate [illegible]

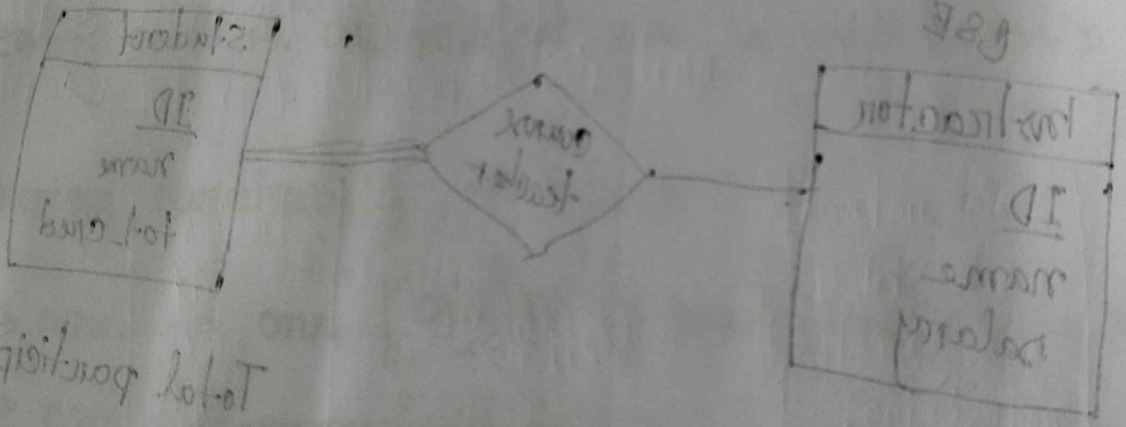


Lab Mid

1. room no এর average ≤ room no → Dep name

2. B-A এর class কয়টা করেছে stu এর ID and name.





Total and Partial participation

Total participation: Every entity in the entity set participates in at least one relationship in the relationship set.

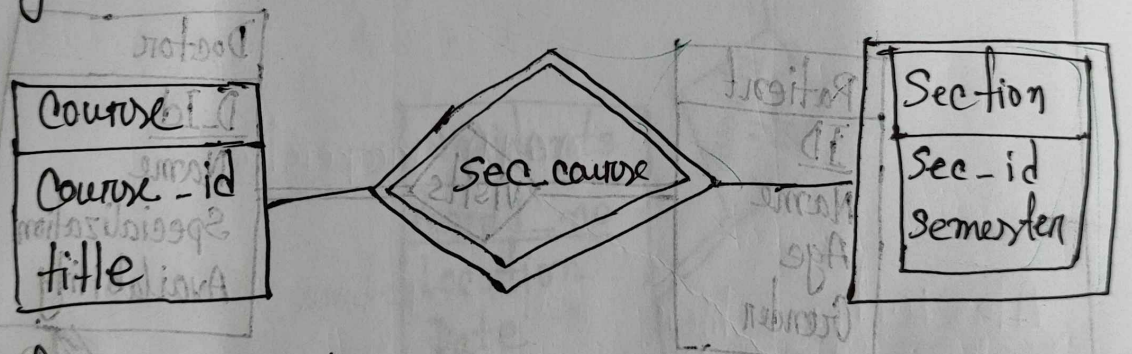
Partial participation: Some relationship is partial participation.

DB

13-10-2025

Strong entity set	Weak entity set
1. Has primary key → student (ID, Name, Intake, section, C.GPA)	1. Doesn't have primary key
2. Isn't dependent on another table for identifying it's row	2. Dependent on other table/ table for identifying it's row.

Strong and weak এর মধ্যে 2 টা কাজ :



Step for constructing an E.R diagram:

Exercise 3: Construct an E.R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examination conducted. Your

design should include an E.R diagram, a set of relation schemas. and a list of constraints. include primary key and foreign key constrains.

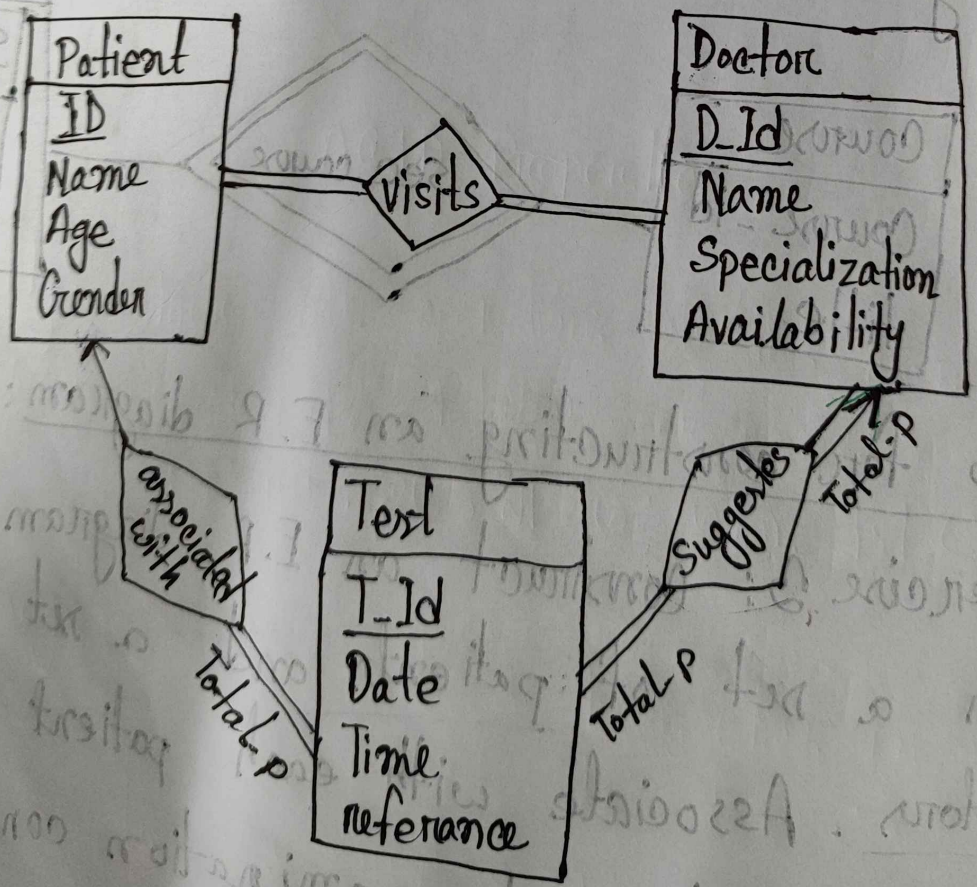
Step 1: Find the no of table and their corresponding attributes.

Step 2: Find P.K and F.K

Step 3: Find the relationship between tables

Step 4: Mapping cardinality

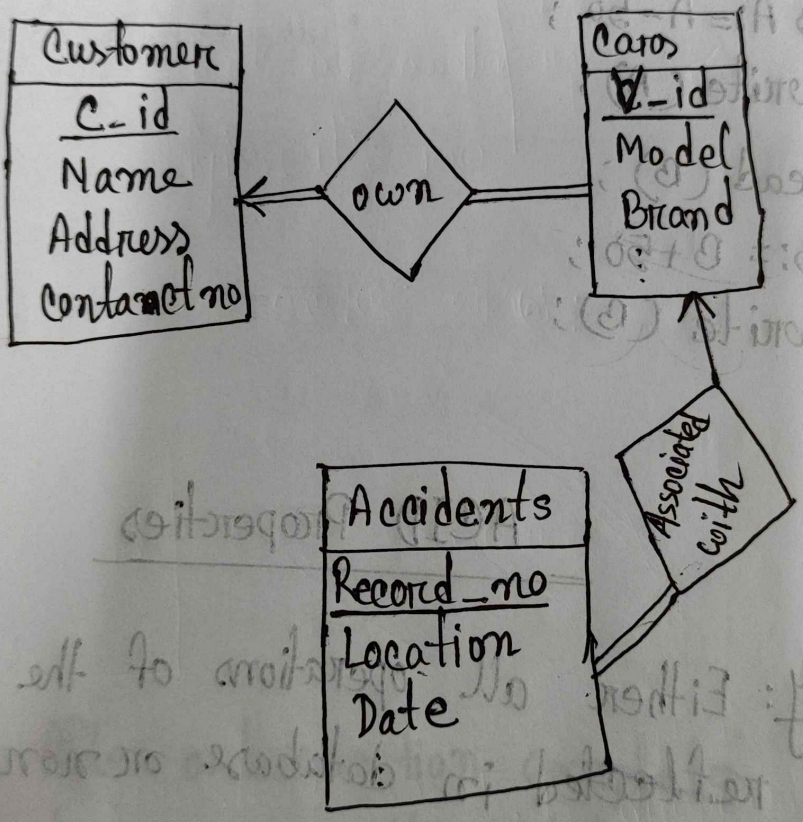
Step 5: Total partial participation



DB

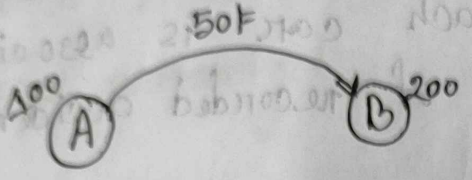
14-10-2025

Example 2 : Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car is associated with it zero to any numbers of recorded accidents.



Exercise 2 Practice

Transaction [150 or 14 no] sure



A transaction is a unit of Program execution that accesses and possibly updates various data items.

1. read (A);
2. A := A - 50;
3. write (A);
4. read (B);
5. B := B + 50;
6. write (B);

ACID Properties

Atomicity: Either all operations of the transaction are Properly reflected in database or none are.

[সব সুদ্ধ মরে num হবে, না হলে আবার 1st থেকে start]

Consistancy: Execution of a transaction in isolation preserves the consist of the data.

[transaction এর আগে ও পরে same থাকবে]

Isolation: Concurrent transaction, conflict serializable হলে তা isolation property maintain করবে।

Durcability: After a transaction completes successfully, the changes it has to the database persist, even if there are system failures. [Transaction শেষ হওয়ার

Project proposal

title

Background :

Motivation : point করে

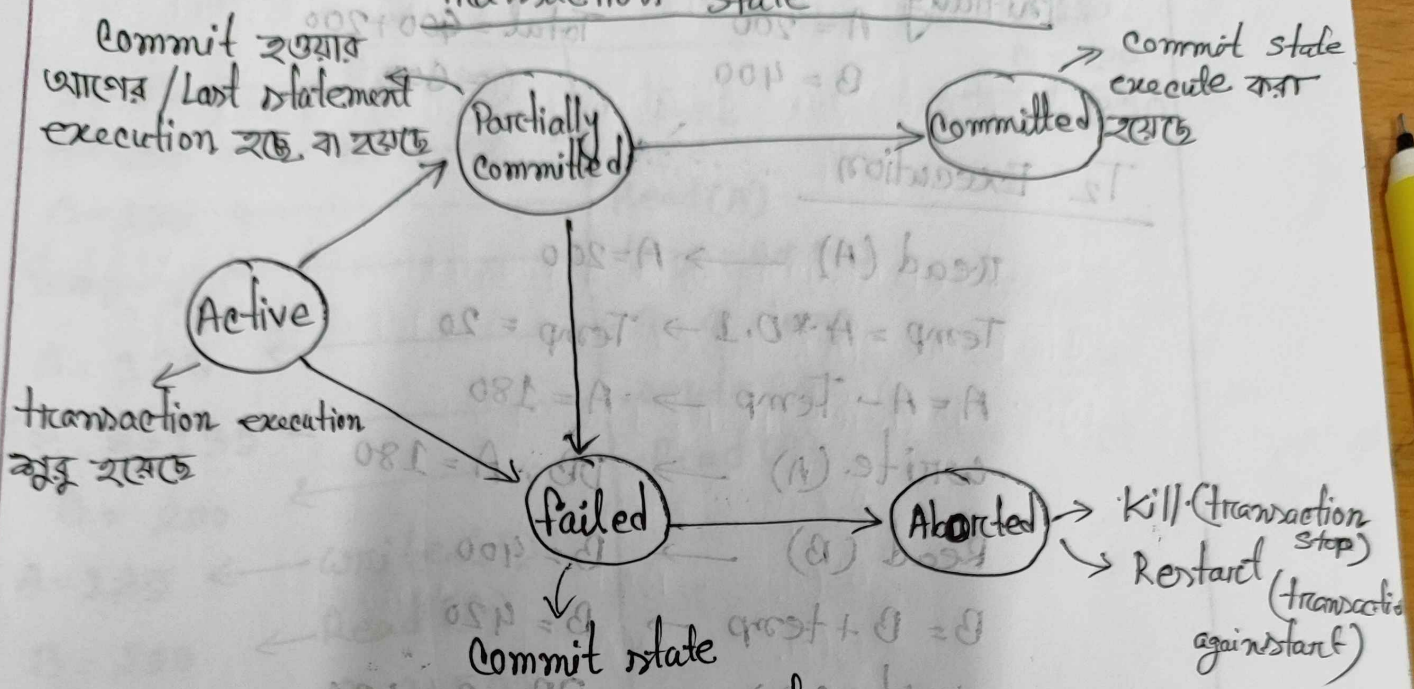
Key features : point করে

used tools and language: ..

DB

20-10-2025

Transaction state



Concurrent Executions

→ Serial execution : (একটা Transaction successfully compleat হওয়ার পর অন্য একটা Transaction execute হবে)

→ Concurrent execution : (একটা Transaction fulfill compleat হওয়ার আগে অন্য একটা Transaction execution start হবে)

Initially A = 200 Total = 400 + 200
B = 400 = 600

T_2 Execution

Read (A) ⟶ A = 200
Temp = A * 0.1 → Temp = 20
A = A − Temp → A = 180
write (A) ⟶ DB, A = 180
Read (B) ⟶ B = 400
B = B + temp → B = 420
write (B) → DB, B = 420

Final, A = 180
B = 420

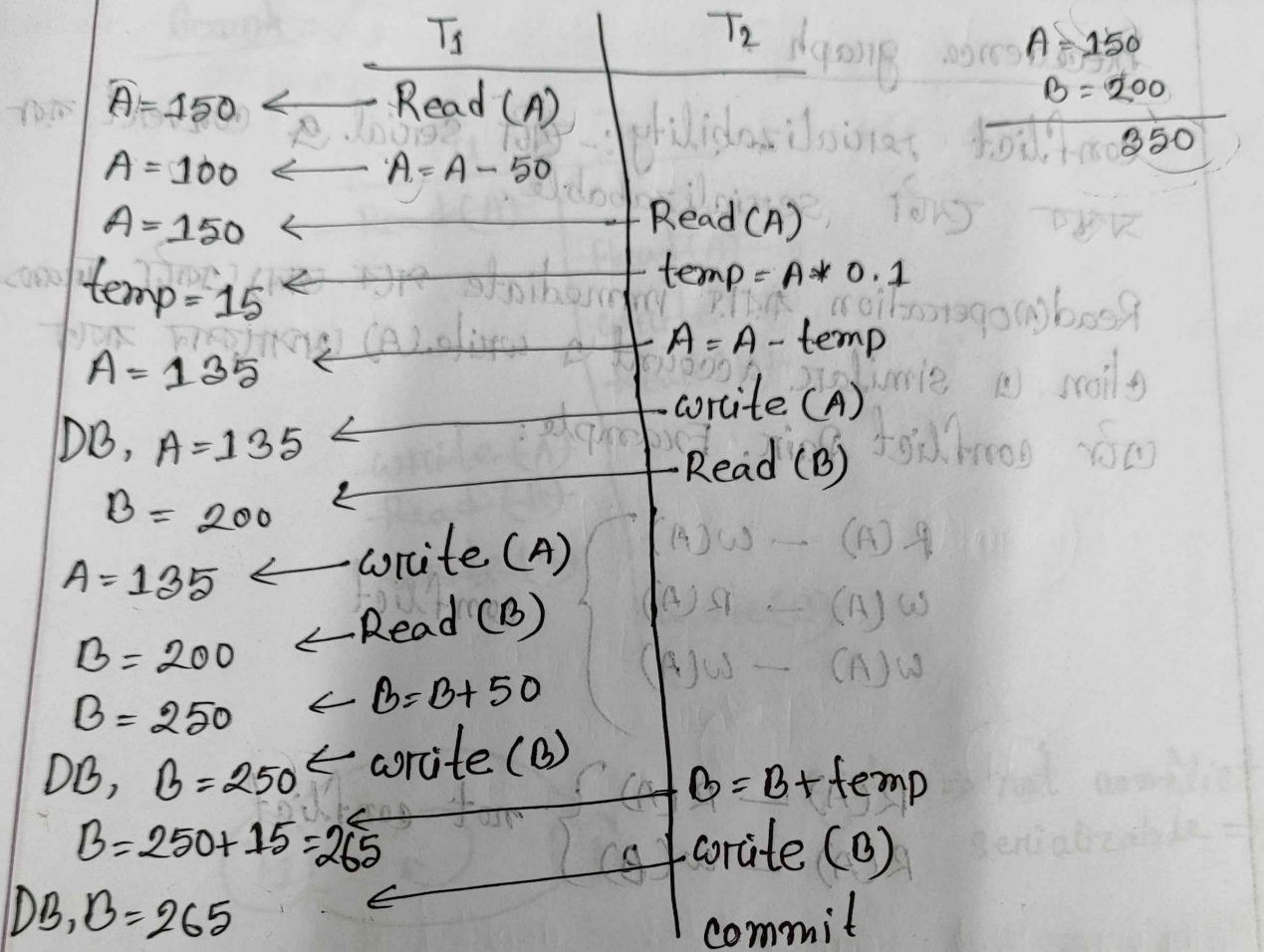
T_1 Execution

Read (A) → A = 180
A = A − 50 → A = 130
~~Read~~
write (A) → DB, A = 130
Read (B) → B = 420
B = B + 50 → B = 470
write (B) → DB, B = 470

Final, A = 130
B = 470
Total = 130 + 470 = 600
So, ~~Committed~~ Consistance

Database

21-10-2025



Final, A = 135
B = 265
400

Not consistence

Precedence graph :

Conflict serializability :- যেটা serial এ convert করা সম্ভব সেটা serializabable.

Read(A) operation করার immediate পরে অন্য একটা Transaction এ similar account এ write(A) অপারেশন করলে সেটা conflict Pair. Example :

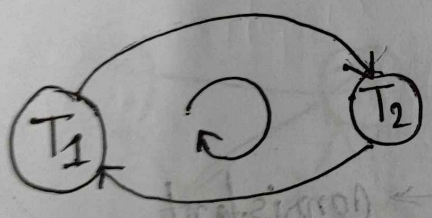
R(A) — W(A)
W(A) — R(A)
W(A) — W(A)
} conflict

R(A) — R(A)
R(A) — W(B)
} not conflict

If, a concurrent schedule (S) can be converted a serial schedule(s) by avoiding ~~non~~ conflict pairs, then we can say that S is conflict serializable.

Graph:

T_1	T_2
Read (A)	
	Read (A)
	write (A)
	Read (B)
write (A)	
Read (B)	
write (B)	
	write (B)



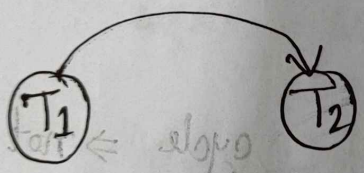
cycle ⇒ not conflict serializable ⇒ Inconsistant

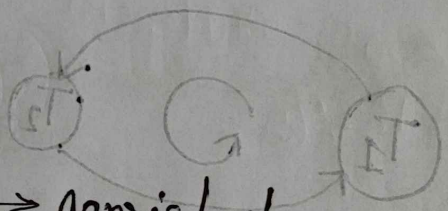
CT

Chapter-6, Transaction

আজানবার

T_1	T_2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	
	R(B)
	W(B)





No cycle ~~conflict~~ serializable → consistant

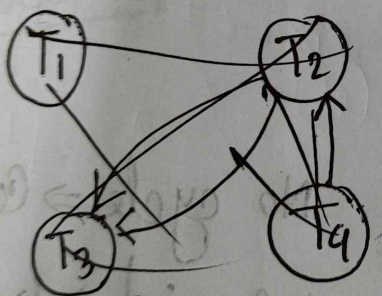
Database

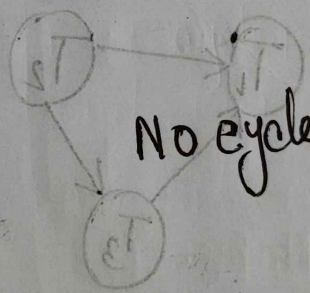
27-10-25

Conflict-Serializability

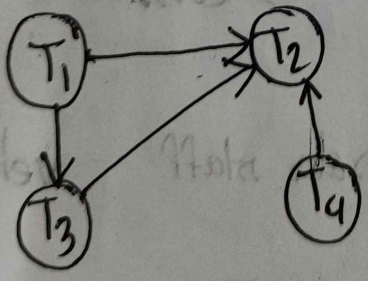
T_1	T_2	T_3	T_4
			R(A)
	R(A)		
		R(A)	
W(B)			
	W(A)		
		R(B)	
	W(B)		

Is the schedule conflict serial ?

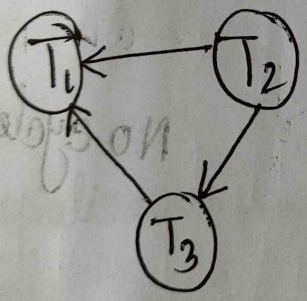


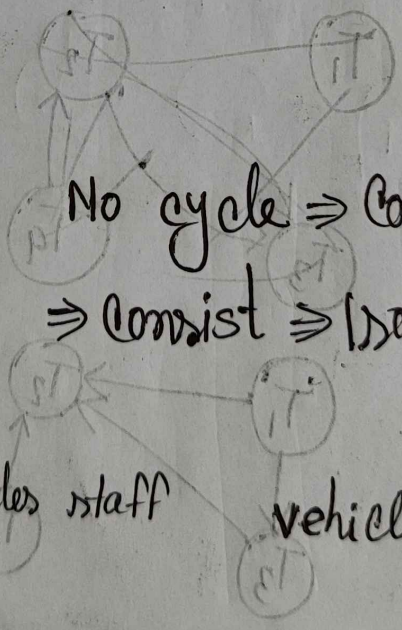


No cycle ⇒ conflict serializable



T_1	T_2	T_3
R(x)		
		R(y)
		R(x)
	R(y)	
	R(z)	
		W(y)
	W(z)	
R(z)		
W(x)		
W(z)		



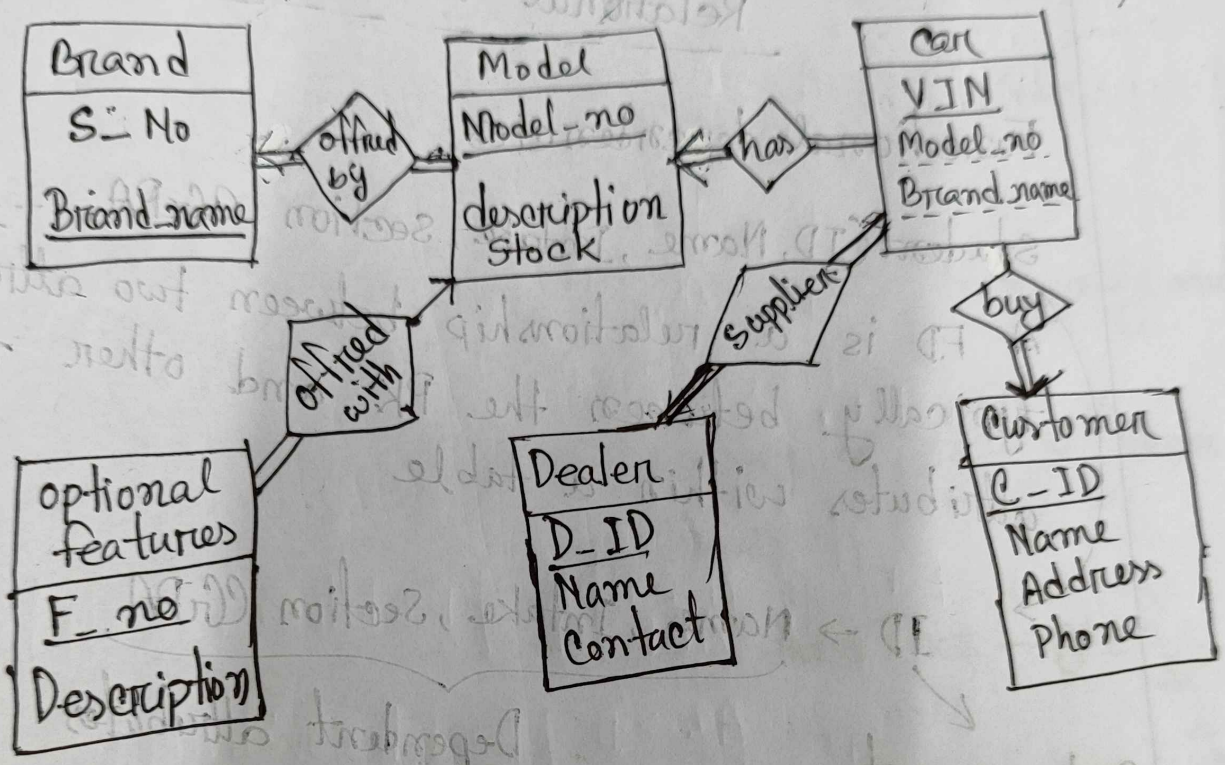


No cycle ⇒ Conflict serializable

⇒ Consist ⇒ Isolation property maintain

Dealers customer sales staff vehicle model brand

Exercise 2


Brand
S_No
Brand_name
offred by
Model
Model_no
description
stock
has
Car
VIN
Model_no
Brand_name
offred with
supplies
buy
optional features
F_no
Description
Dealer
D_ID
Name
Contact
Customer
C_ID
Name
Address
Phone

Database 3-11-2025

Chapter 7

Relational Database Design

Functional dependencies:

Student (ID, Name, Intake, Section, CGPA, ---)

A FD is a relationship between two attributes, typically between the PK and other non-key attributes within a table

ID → Name, Intake, Section, CGPA

ID: Determinant attribute

Name, Intake, Section, CGPA: Dependent attributes

ID, Name →

Composite PK

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_1	c_2	d_2	e_1
a_3	b_2	c_1	d_1	e_1
a_4	b_2	c_2	d_2	e_1

A → BCDE ✓

B → DE ✗

D → BC ✗

Rules of functional dependencies:

1. Transitivity rule:- If, $x \to y$ & $y \to z$ then $x \to z$
2. Union rule:- If, $x \to y$ & $x \to z$ then, $x \to yz$
3. Decomposition rule: If, $x \to yz$, then $x \to y$ and $x \to z$
4. Pseudotransitivity :- If, $x \to y$ & $yz \to w$, then $xz \to w$
5. Augmentation :- If, $x \to y$ then $xz \to yz$

Example:

Q. $R(A, B, C, G, H, I)$, $F = \{A \to B, A \to C, CG \to H, CG \to I, B \to H\}$

⇒ Closure of F / F+

$A \to BC$, $A \to H$, $A \to BCH$

~~$A \to H$, $CG \to H$~~

⇒ $F+ = \begin{cases} A \to BC \text{ [Union]} \\ CG \to HI \text{ [Union]} \\ A \to H \text{ [Transitivity]} \end{cases}$

$AG \to H$ [Pseudotransitivity]

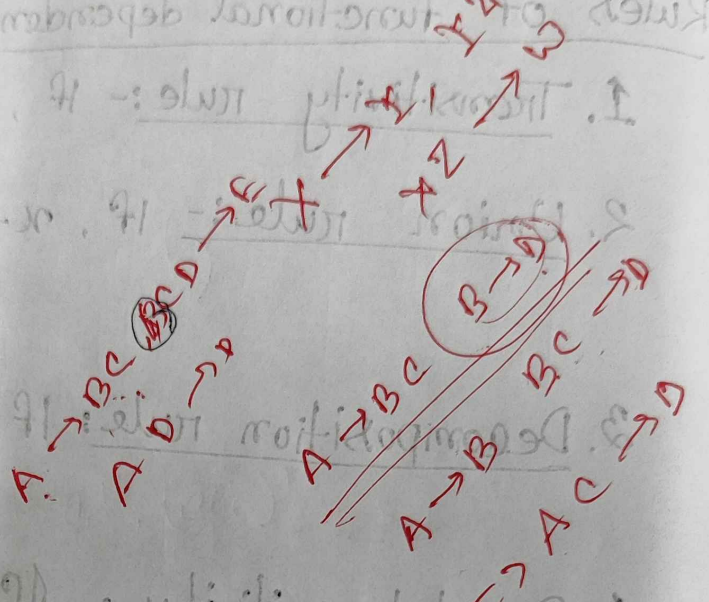
$AG \to I$ [Pseudotransitivity]

DB

24-11-2025

$R(A, B, C, D, E)$

$F = \{ A \to BC$
$CD \to E$
$B \to D$
$E \to A \}$



Find F+:

$\Rightarrow$

$F+ = \{$ ~~$A \to BDE$~~ [Pseudotransitivity]
$ABD \to E$

$AC \to D$ [Pseudotransitivity]

$BC \to E$ [Pseudotransitivity]

$E \to BC$ [transitivity]

$A \to B$ and $A \to C$ [Decomposition]

$CD \to A$ [transitivity] $\}$

R(A, B, C)

F = { A → BC
B → C
A → B
AB → C }

F+ = { A → C [transitivity]
~~A → C and B → A~~ [Decomposition]

closure of attribute sets

R(A, B, C)

F = { A → BC
A → B
B → C
AB → C }

Find A+, B+, (AB)+

⇒ A+ = ABC

B+ = BC

(AB)+ = ABC

A+ = ABC

B+ = BC

Q. $R(A, B, C, G, H, I)$

$F = \{ A \to B$
$A \to C$
$CG \to H$
$CG \to I$
$B \to H \}$

Find $A^+, B^+, C^+, (CG)^+$

⇒

$A^+ = ABCH$ ✓

$B^+ = BH$ ✓

$C^+ = C$ ✓

$(CG)^+ = CGHI$ ✓

Q. $R(A, B, C, D, E)$

$F = \{ A \to BC$
$CD \to E$
$B \to D$
$E \to A \}$

Find $A^+, B^+, C^+, (CD)^+, E^+$

⇒

$A^+ = ABCDE$

$B^+ = BD$

$C^+ = C$

$(CD)^+ = CDEAB$

$E^+ = EABCDE$

$A^+ = ABCDE$

DB

10-11-2025

$R = (A, B, C, D, E)$ and $F = \{A \to BC, CD \to E, B \to D, E \to A\}$.
Compute A^+, B^+, $(CD)^+$, E^+ also, compute F^+.

⇒

$A^+ = ABCDE$ ✓

$B^+ = BD$ ✓

$(CD)^+ = CDEAB$ ✓

$E^+ = EABCD$ ✓

SK = A, CD, E
CK = A, E
PK = A or E

$B \to D$, $CD \to E$
$\Rightarrow BC \to E$

$F^+ = A \to B$ and $A \to C$ [Decomposition] ✓
$= BC \to E$ [Pseudotransitivity] ✓
$= CD \to A$ [Transitivity] ✓

SK = BC, ABC, AC
CK

Finding super key, candidate key & Primary key from functional dependency sets:

$R(A, B, C)$ → Determinant
$F = \{A \to BC,$
$B \to C$
$A \to B$
$AB \to C\}$

Find sk, ck, PK for this table.

1. Calculate the closure of all determinant attribute.

$A^+ = ABC$
$B^+ = BC$
$AB^+ = ABC$

AC, ABC
CK = AC, AB
PK = A

So, SK = A, AB [যে closure এ সব attribute চলে আসে]
CK = A, B, C
PK = A

Q. $R = (A, B, C, D, E, F)$

$F = \{ AB \to C,\ BC \to AD,\ D \to E,\ CF \to B \}$

$AB^+ = ABCDE$

$BC^+ = BCADE$

$D^+ = DE$

$CF^+ = CFBADE$

$SK = CF$

$CK = CF$

$PK = CF$

$A^+ = ABCH(G)$

$(AG)^+ = AGBCHI$

Q. $R(A, B, C, G, H, I)$

$F = \{ A \to B,\ A \to C,\ CG \to H,\ CG \to I,\ B \to H \}$

Find out SK, CK, PK, for this scenario

$A^+ = ABCH$

$(CG)^+ = CGHI$

$B^+ = BH$

$(AG)^+ = AGBCHI$

$AG^+ = AGBCHI$

$(ACG)^+ = ACGBHI$

$SK = AG, ACG$

$CK = AG$

$PK = AG$

Q. R(A, B, C, D, E, F)

F = { AB → C,
BC → AD
D → E }

⇒ $(AB)^+ = ABCDE$

$(BC)^+ = BCADE$

$D^+ = DE$

$(ABF)^+ = ABFCDE$

$(BCF)^+ = BCFADE$

SK = ABF, BCF

CK = ABF, BCF

PK = ABF / BCF

DB

11-12-2025

Q. R(A, B, C, D, E, H)

$F = \{ A \to B,\ BC \to D,\ E \to C,\ D \to A \}$

⇒

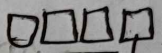
$A^+ = AB$

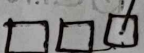
$BC^+ = BCDAB$

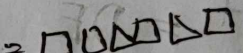
$E^+ = ECH$

$D^+ = DAB$

$(BCEH)^+ = BCEHDA$

$B^+ =$ 

$D^+ =$ 

$BD^+ =$ 

SK = BCEH

CK = BCEH

PK = BCEH

Canonical cover :

$R(A, B, C)$

$F = \{ A \to BC \Rightarrow A \to B, A \to C$
$B \to C$
$A \to B$
$AB \to C$

Step 1 (কোন টাকে Decomposition rule apply হবে, apply করবো।)
Apply decomposition rule if possible.

$F = \{ A \to C$
$B \to C$
$A \to B$
$AB \to C \}$

Step 2: Check whether $A \to C$ is necessary or not.

$A^+ = ABC$

$-(A)^+ = ABC$

So, $A \to C$ is not necessary

$F = \{ B \to C$
$A \to B$
$AB \to C \}$

Step 3 Check whether $B \to C$ is necessary or not.

$B^+ = BC$

$-(B^+) = B$

So, $B \to C$ is necessary.

$F = \{ B \to C$
$A \to B$
$AB \to C \}$

Step 4: Check whether $A \to B$ is ~~Step 3~~ : necessary or not.

$A^+ = ABC$

$-(A)^+ = A$

So, $A \to B$ is necessary.

$F = \{ B \to C$
$A \to B$
$AB \to C \}$

Step 5: Check whether $AB \to C$ is necessary or not.

$AB^+ = ABC$

$-(AB)^+ = ABC$

So, AB^+ is not necessary

$F = \{ B \to C$
$A \to B \}$

So the canonical cover is =

$\{ A \to B$
$\{ B \to C$

DB

17-11-2025

R (A, B, C, D, E)

F = { A → BC
CD → E
B → D
E → A }

using the functional dependencies compute the canonical cover of F.

⇒ step 1: using decomposition rule,

F = { A → B
A → C
CD → E
B → D
E → A }

step 2: checking A → B is necessary or not.

$A^+ = ABCDE$

$-A^+ = AC$

So, A → B is necessary

F = { A → B
A → C
CD → E
B → D
E → A }

Step 3: Checking $A \to C$ is necessary or not.

$A^+ = ABCDE$

$-A^+ = ABD$

So, $A \to C$ in necessary.

$F = \{ A \to B$
$A \to C$
$CD \to E$
$B \to D$
$E \to A \}$

Step 4: Checking $CD \to E$ is necessary or not.

$CD^+ = CDEABC$

$-CD^+ = CD$

is necessary

Step 5: $B \to D$

$B^+ = BD$

$-B^+ = B$

is necessary

Step 6: $E \to A$

$E^+ = EABCDE$

$-E^+ = E$

is necessary

$F = \{ A \to B$
$A \to C$
$CD \to E$
$B \to D$
$E \to A \}$

Normalization

→ Is a Database technique

→ Is a process of reducing / removing redundancy from Database

Emp_ID	Name	Contactno	Dept	Dept_manager
101	A	0171	Eng	FH
102	B	013	HR	MR
103	C	015	MK	MJ
104	D	016	Eng	FH
105	E	015	MK	MJ
101	A	0171	Eng	FH

II. X.

1. Row-wise redundancy
2. Column-wise redundancy
   - (I) Insertion Anomaly
   - (II) Delete "
   - (III) Update "

1NF , 2NF, 3NF

1NF:-

St_ID	Name	course
101	A	Ds. DBMS
102	B	OS
103	C	DBMS
104	D	Algo, DBMS

C, C++, OOP

1st 1NF :-

St_ID	Name	Course 1	Course 2
101	A	OS	DMMS
102	B	OS	NULL
103	C	DBMS	NULL
104	D	Algo	DBMS

2nd 2NF :-

Std_ID	Name	course
101	A	OS
101	A	DBMS
102	B	OS
103	C	DBMS
104	D	Algo
104	D	DBMS

2NF :-

1. Table must be in 1 NF
2. Doesn't allow partial functional dependency

Composite PK

Stud_ID	Name	Address	Course_no	Course_title	Credit	Grade
101	A	DH	207	DB	3	A+
101	A	DH	208	DB_Lab	1.5	A+
102	B	CH	207	DB	3	A+
102	B	CH	208	DB_Lab	1.5	A

Stud_ID, Course_no → Name, Address, Course_title, Credit, Grade

Not allow in 2NF

Partial dependency:

Stu_ID → Name, Address

Course_no → Course_title, Credit

Student

Stu_ID	Name	Address
101	A	DH
102	B	CH

PK

Course

Course_no	Course_title	Credit
207	DB	3
208	DB_Lab	1.5

Composite PK

Stu_ID	Course_no	Grade
101	207	A+
101	208	A+
102	207	A+
102	208	A

3NF :-

1. Table must be in 2NF
2. Doesn't allow transitive fd (functional dependency)

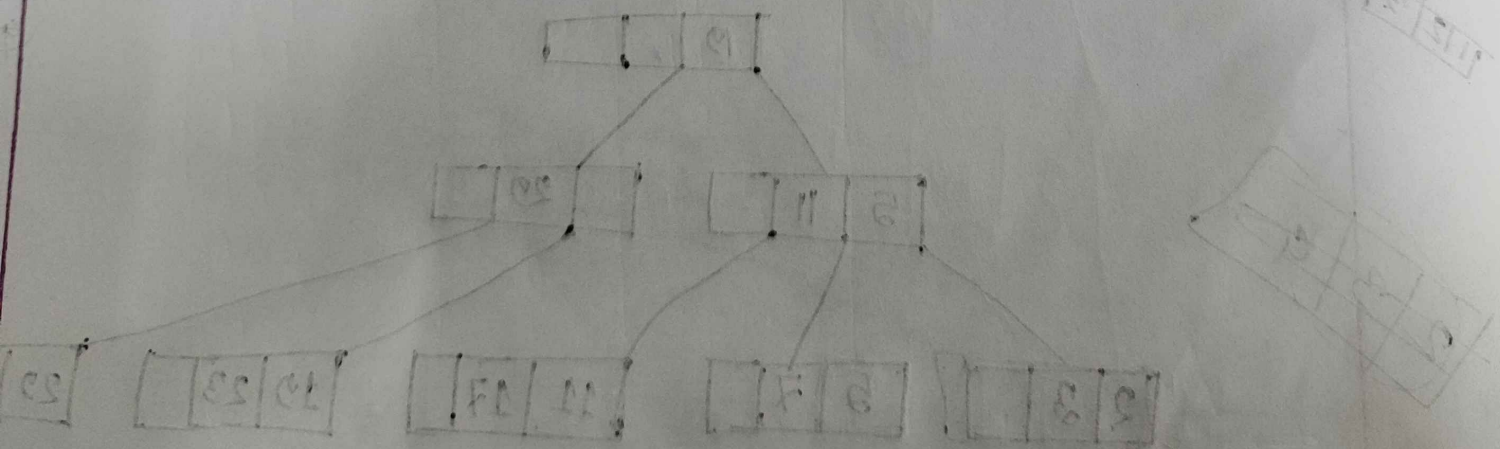
Course_title	Teacher_ID	Teacher_name	credit
C	1	FAM	3
OS	2	IFT	3
DB	1	FAM	3
OS_Lab	2	FAM	2

PK

Teacher_ID	Teacher_name
1	FAM
2	IFT

PK

Course_title	credit	Teachers
C	3	1
OS	3	2
DB	3	1
OS_Lab	2	1



chapter-12

Indexing & Hashing

B+ Tree (n = Order of B+ tree)

if $n = 4$

$$\left\lceil \frac{n-1}{2} \right\rceil = \left\lceil \frac{4-1}{2} \right\rceil = \lceil 1.5 \rceil = 2$$

⇒ Each leaf node can contain maximum $(n-1)$ values.

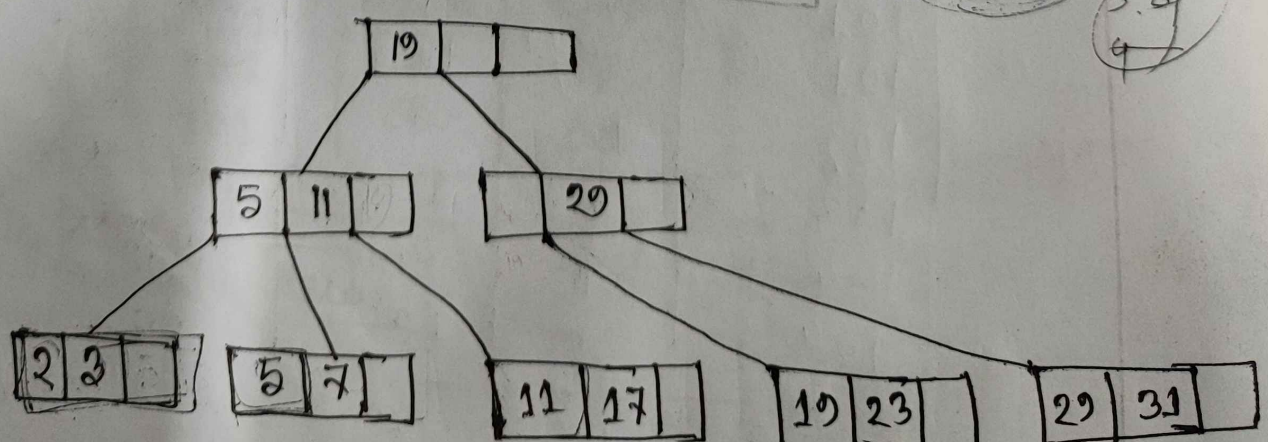
⇒ " " have to contain at least $\left\lceil \frac{n-1}{2} \right\rceil$ values.

Example.

2, 3, 5, 7, 11, 17, 19, 23, 29, 31 $(n = 4)$

leaf = 4 - 1 = 3

$$min = \left\lceil \frac{4-1}{2} \right\rceil = 2$$



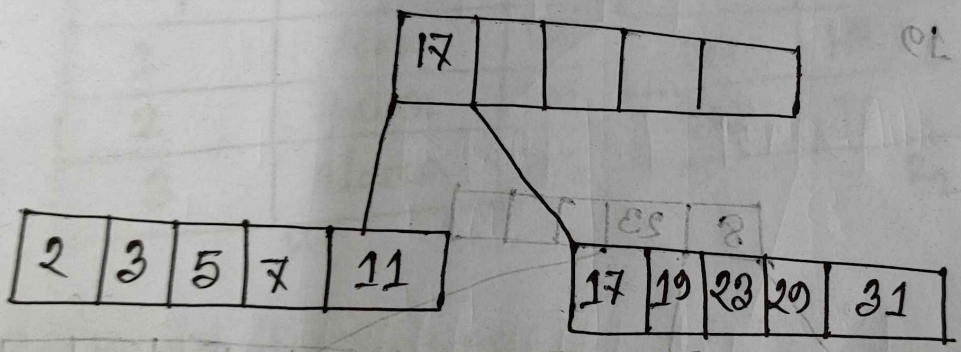
2, 3, 5, 7, 11, 17, 19, 23, 29, 31 (n=6)

(i) insert 8

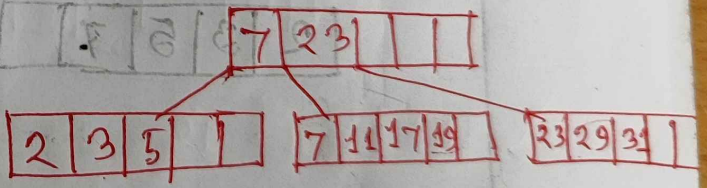
(ii) insert 45

⇒ Leaf node = 6-1
= 5

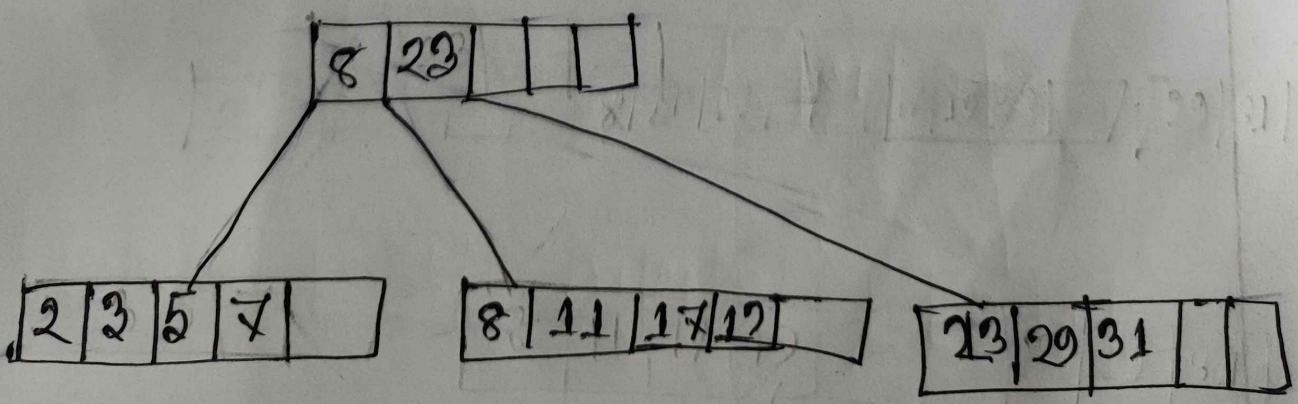
min = $\lceil \frac{6-1}{2} \rceil$
= 3



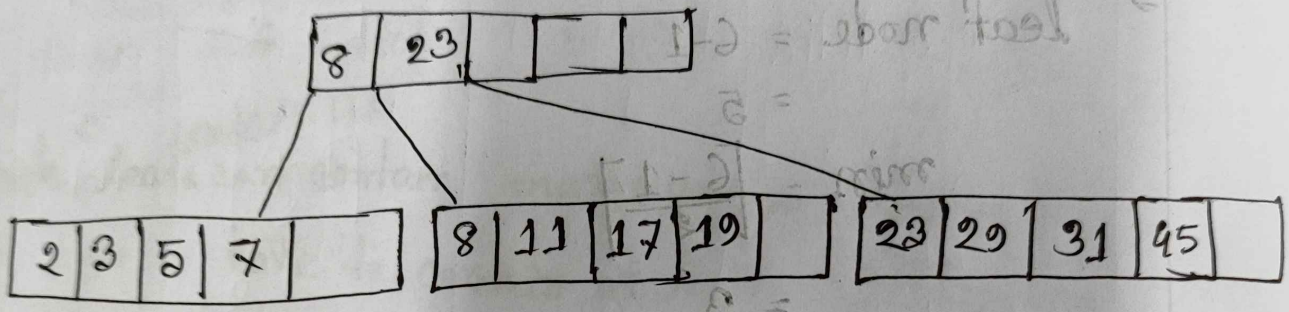
(i) insert 8



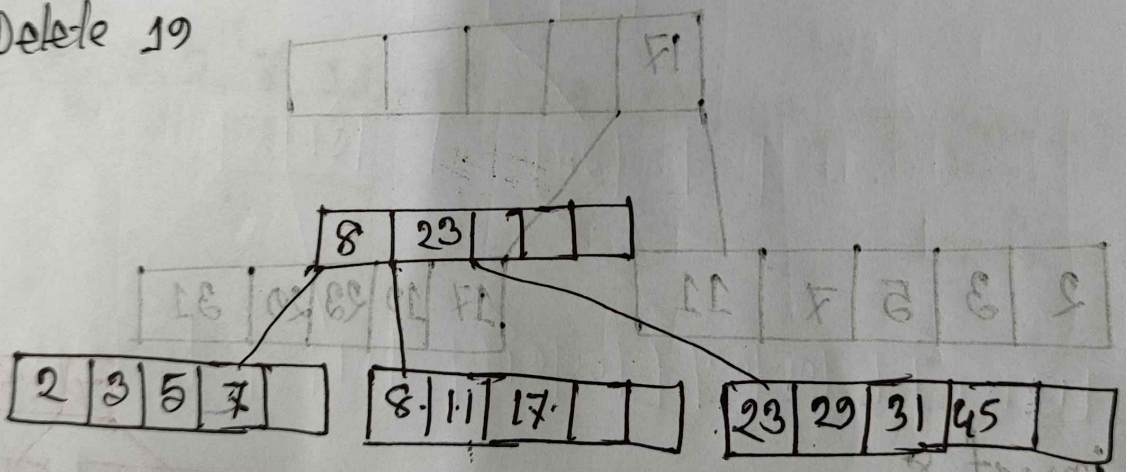
(ii)



(II) Insert 45



(III) Delete 19



Q6.

(i)

Emp_ID (PK)	Emp_name	Dept_ID	Dept_name	Dept_Location
1	Alex	D01	HR	New York
2	Brian	D02	IT	Chicago
3	Clara	D01	HR	New York

(I) 1NF :- Already in 1NF

2NF :-

R_1

Emp_ID (PK)	Emp_name	Dept_ID (FK)
1	Alex	D01
2	Brian	D02
3	Clara	D01

R_2

Dept_ID (PK)	Dept_name	Location
D01	HR	New York
D02	IT	Chicago

3NF : Already in 3NF

(II) R_1 : PK → Emp_ID, FK → Dept_ID

R_2 : PK → Dept_ID

Q7. 4, 6, 8, 17, 19, 20, 30, 40, 45, 51 $n=4$

(i)

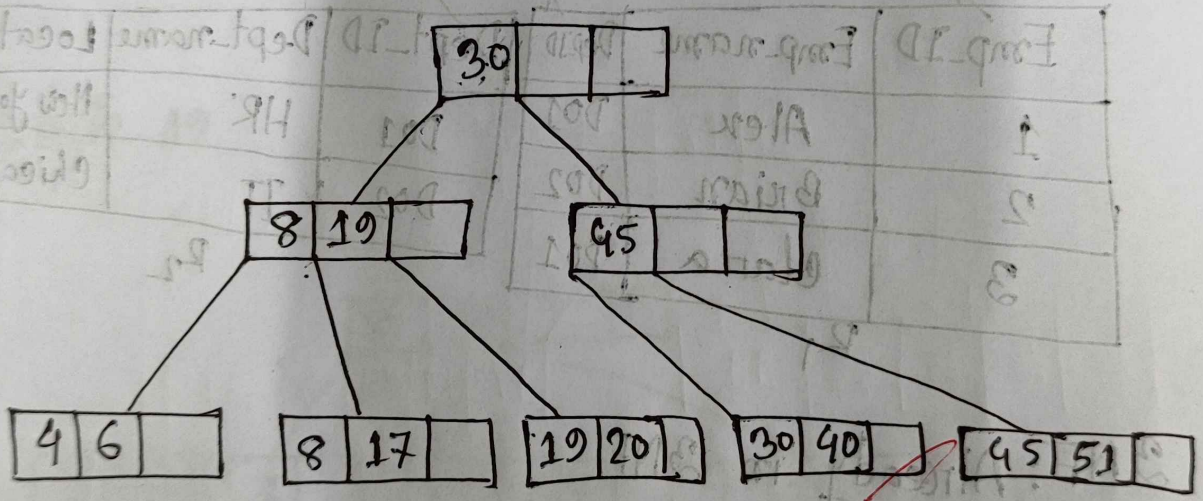
Max $= n-1$

$= 4-1$

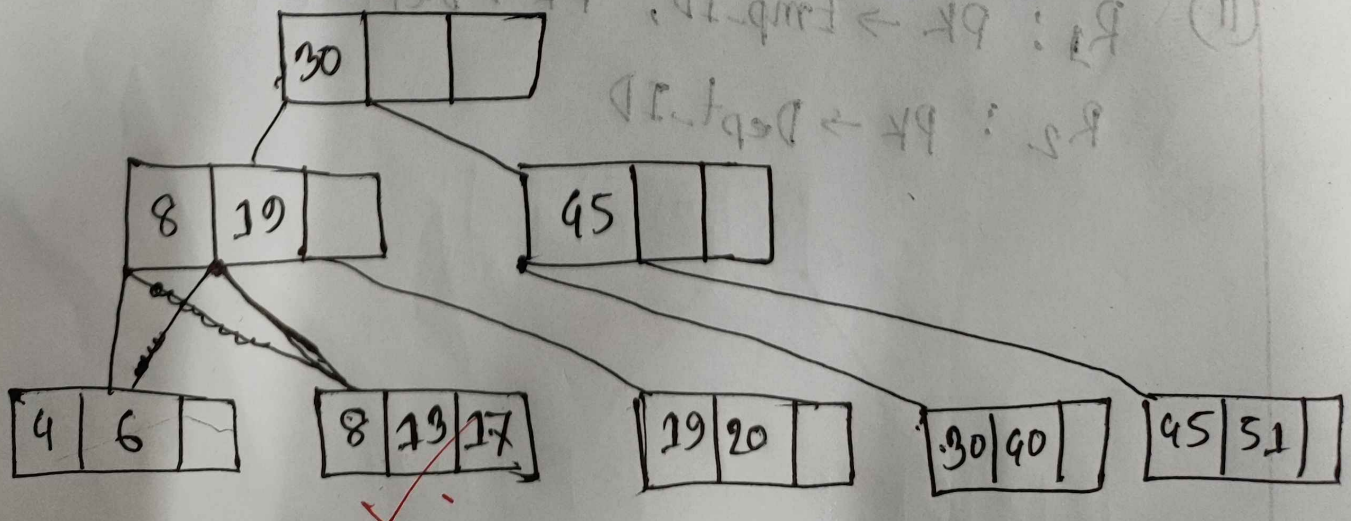
$= 3$

Min $= \left\lceil \frac{4-1}{2} \right\rceil$

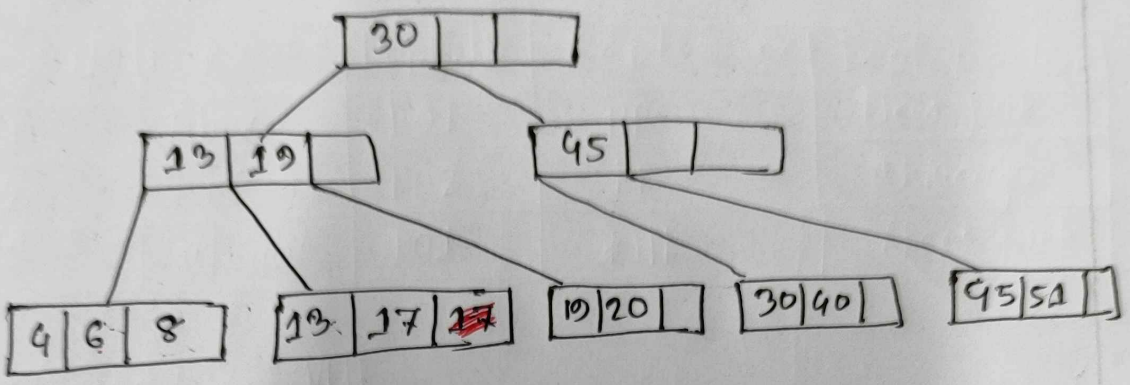
$= 2$



(ii) Insert 13



(III) Insert 17:



(IV) Delete 6.