Structure of DBMS

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

★ Problems of File System (Traditional approch) ⇒

① Security ⇒ Difficult to remember diff passwords for multiple files.

② Data Redundancy ⇒ Duplicate
⇒ Diff tables have same details of a person.
⇒ If update one, other will not change.

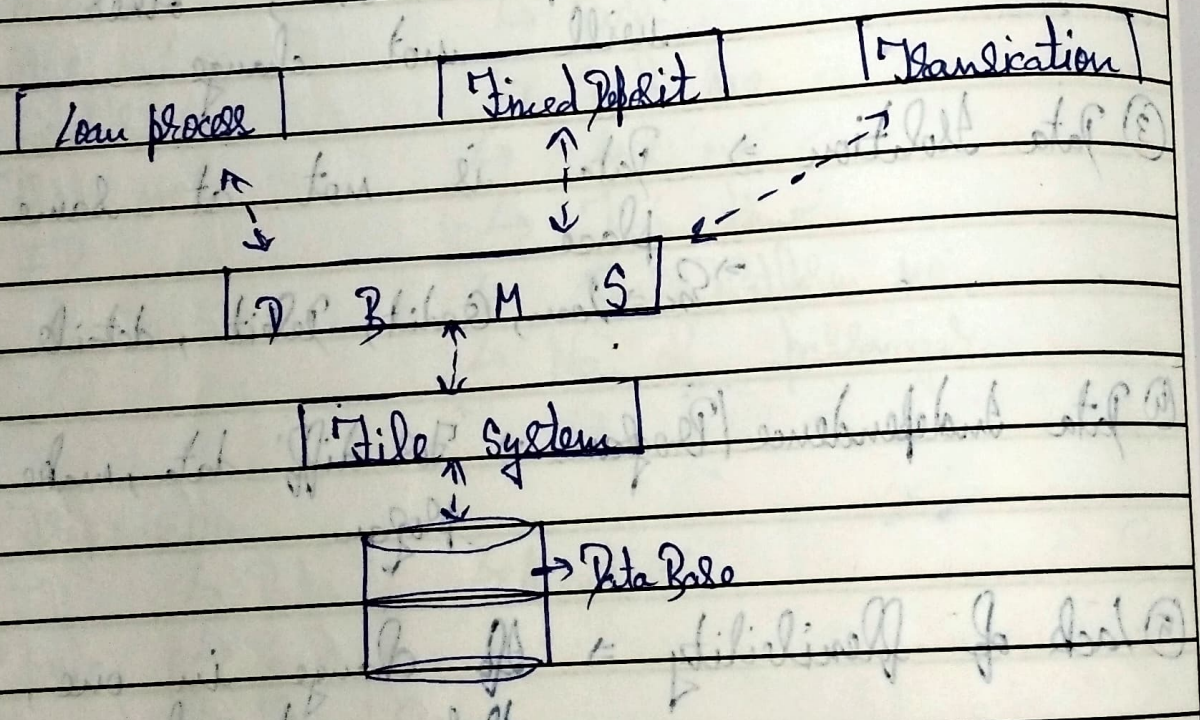
③ Data Isolation ⇒ Data is not at same place
⇒ Ex ⇒ Loan, Credit, Debit, details

④ Data Independence / Program ⇒ For diff data, make diff program

⑤ Lack of flexibility ⇒ If change in one, other not change.
⇒ Change phone no. in hostel but didn't change in college, Exam.

(6) Concurrent access ⇒ Can't access different data together.
(साथ-साथ)
⇒ Can't access single data at different places at same time
⇒ Ex ⇒ RTV result

☆ Where does DBMS fits in ⇒



Main Ideas, Questions & Summary:-

Library Ref:-

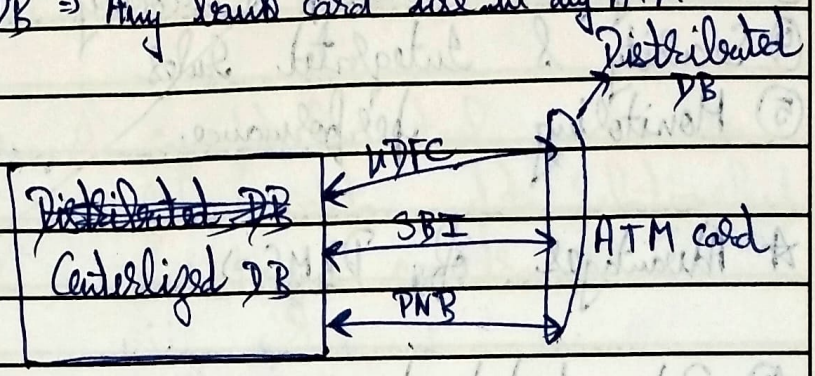
POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

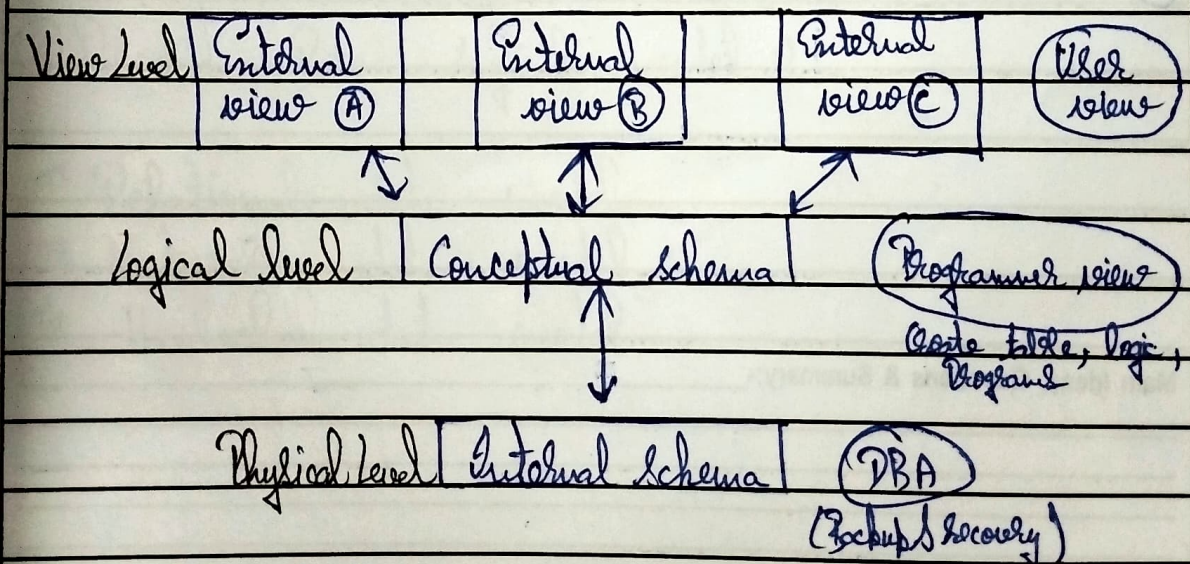
★ Types of DB =>

① Centerlized DB (TCS) => Bank card only use in same ATM

② Distributed DB (Bank) => Any bank card use in any ATM



★ 3 Layer Archieture => / Levels of distraction



Same DB, diff. view

Details, Amount, gallery from same DB, same time

★ Use of DBMS ⇒

(1) Managing information contain
(2) Relationship with user
(3) Stragegied backup & Recovery
(4) Security & Integrated Rules
(5) Monitoring performance

★ Advantages of DBMS ⇒

(1) Data Independence
(2) Reduction in Redundancy
(3) Better Security
(4) Enable backup & Recovery
(5) Better flexibility
(6)

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:
2/2/2022						

★ Integrity constraint ⇒ Change data at one place, change in all.
⇒ Data is at same place.

★ Data Model ⇒ A conceptual tool use to describe data, data relationship, data semantics, consistency constraint.

• Types ⇒

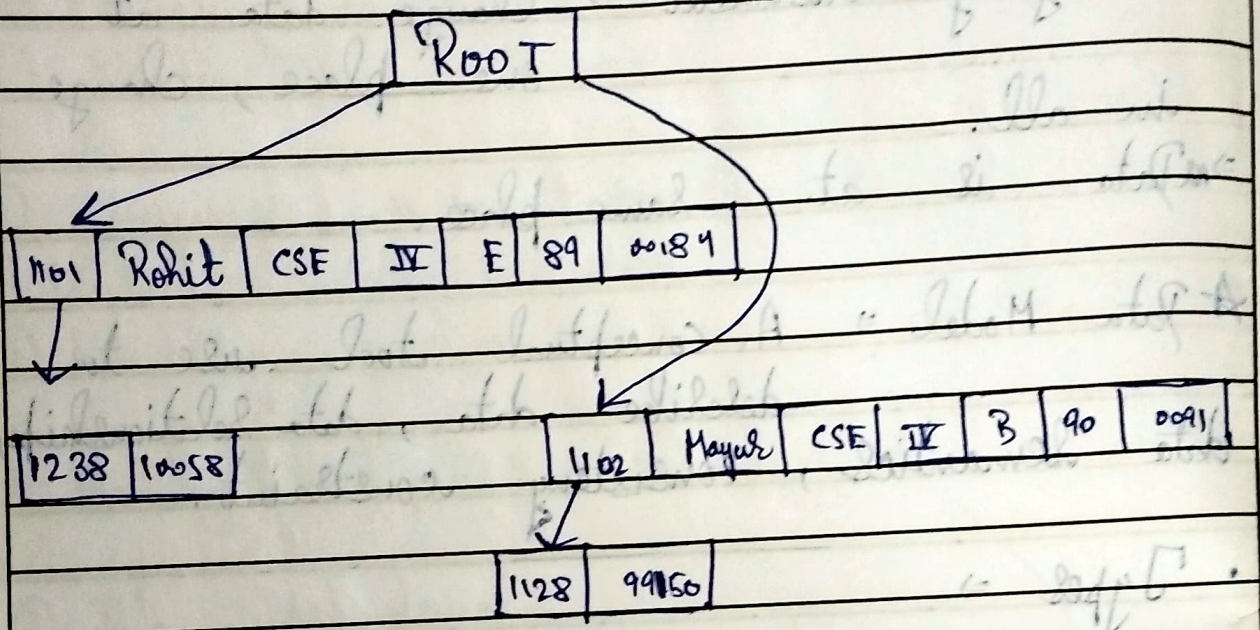
① Object Base logical model ⇒

⇒ Entity Relationship

② Record Base logical model ⇒

⇒ Relational data model
⇒ Network data model
⇒ Heirachal data model

★ Record Base data model ⇒
(Hierarchial)

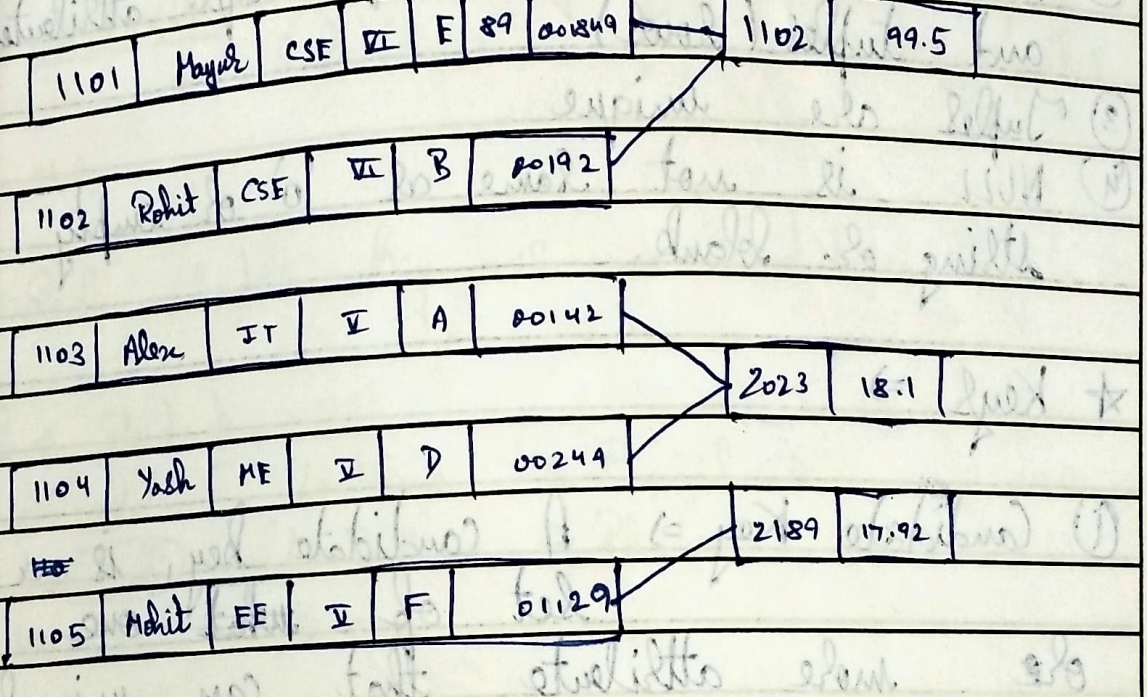


Main Ideas, Questions & Summary:-

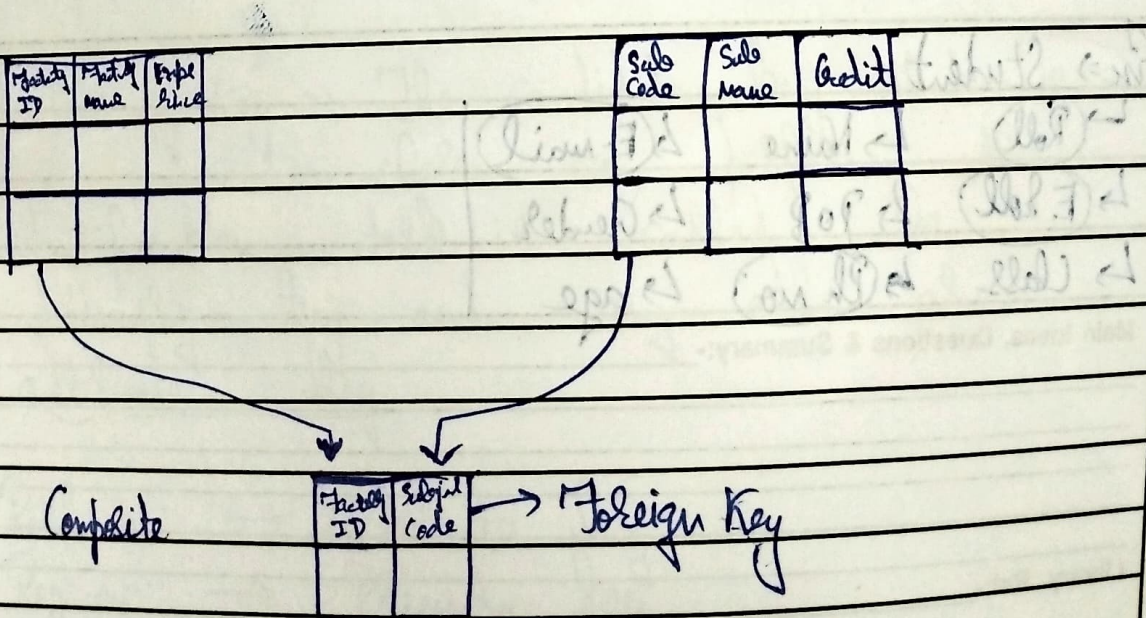
Library Ref:-

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Network data model ⇒



☆ Relational data model ⇒



✱ Basics of RDM ⇒

① Data is seen as 2D table, known as Relation.
② A Relation (table) consist of unique attributes and tuples (rows)
③ Tuples are unique
④ NULL is not same as '0' or empty string or blank.

✱ Keys ⇒

① Candidate Key ⇒ A candidate key, is a set of one or more attribute that can uniquely identify a row in a given table.

Ex ⇒ Student

↳ (Roll)	↳ Name	↳ (E-mail)
↳ (E. Roll)	↳ DOB	↳ Gender
↳ Class	↳ (Ph no)	↳ age

Main Ideas, Questions & Summary:-

Library Ref:-

☆ Tuple/Field ☆

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

(2) Super key ⇒ An attribute or group of attribute that is sufficient to distinguish every tuple in a relation from every other one.

⇒ Each super key is a candidate key.

Ex ⇒ Student

↳ (Rod) ↳ Name ↳ E-mail

↳ (E.Roll) ↳ DoB ↳ Ph no

↳ class ↳ age

☆ ACID ⇒ Atomicity Consistences Isolation Durability

☆ Primary Key ⇒ The candidate key that is chosen to perform the identification task is called the primary key and any other are alternate key.

Roll no → Alternate key

Reg no → Primary key

⇒ Every tuple must have key defn a unique value for its primary key.

⇒ A primary key which is combination of more than one attributes is called composite key.

★ Foreign key ⇒ It is a copy of primary key that has been imported from one relation into another to represent the existence of a relation b/w them.

⇒ Foreign key is the copy of a whole parent primary key, i.e if a primary key is composite then above is foreign key.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

★ Overlapping Candidate keys ⇒ Two candidate keys overlap if they involve any attribute in common.

Ex ⇒ Customer name, Customer ID
Customer name, Email ID
⇓
Overlapping

⇒ Attributes that does not participate in any candidate key is called non-key attributes.

★ Prefrences ⇒

① Numeric no
② Single attribute
③ Minimal composite key

Ex ⇒ PNR no → Primary key
Name → Non-key

★ Conceptual design ⇒

Entity relationship modelling
Database design technique
Top-down approach
Bottom-up approach → Normalisation

⇒ It is a graphical technique for understanding and organising data independent of actual relationship.

⇒ Terms used ⇒

① Entity ⇒ Anything that may have an independent existence and about which we intend to collect data also known entity type.

② Entity Instance ⇒ Particular meber of entity type.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

③ Attributes ⇒ Properties of characteristics that defines entity.

④ Relationship ⇒ Association b/w entities.
⇒ Self association also

⑤ Domain value ⇒ The set of possible values for an attribute is called the domain.

★ Attributes type ⇒
- Simple V/s Composite

① Simple ⇒ That cannot be divided into simpler components.
⇒ Ex ⇒ Enrollment no
⇒ Gender

② Composite ⇒ That can be further divided
⇒ Ex ⇒ Address

(3) Single valued ⇒ Can take one single value on each entity istance.
⇒ Ex ⇒ Semester.

(4) Multi valued ⇒ Can take multivalued on each entity instance.
⇒ Ex ⇒ Hobbies
⇒ Languages

(5) Stored ⇒ Attributes that mean to be stored permanently.
⇒ Ex ⇒ Citizenship
⇒ Name
⇒ Gender*

(6) Derived ⇒ Attributes that can be calculated from other attributes.
⇒ Ex ⇒ Interest
⇒ DOJ → Work experience

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Regular V/S weak entity type ⇒

- Entity that has it's own key attribute

☆ Weak Entity ⇒

- Entity that depend upon different/another entity for its existance & don't have key attribute.

? Employee not take salary, give father.

Product (R.E)

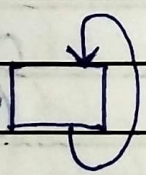
Customer (R.E) — Order (W.E)

★ Relationship ⇒ Connection

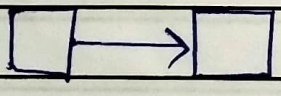
⇒ A relationship tie b/w 2 entity type defines set of all association b/w these entity type.

⇒ Each instance of relationship b/w members of these entity type is called a relationship instance

★ Degree of relationship ⇒ The no of entity type involved



unary



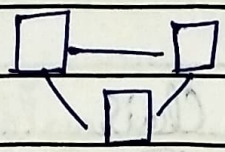
Binary

Main Ideas, Questions & Summary:-

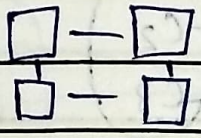
Library Ref:-

POORNIMA UNIVERSITY

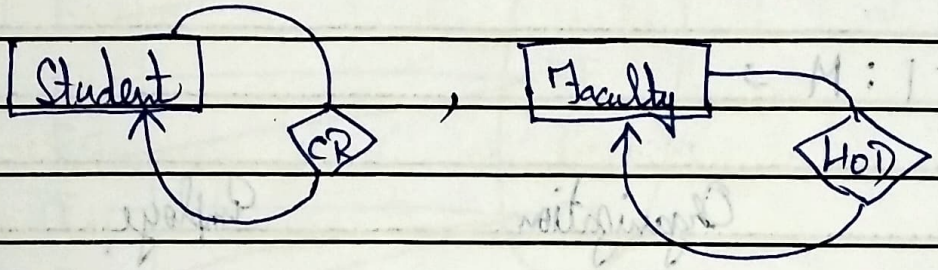
Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

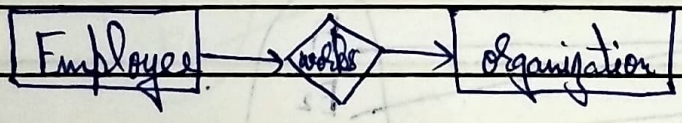


Ternary



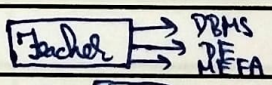
n- ary

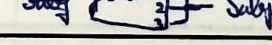
⇒ 

⇒ 

★ Cardinality ⇒ Relationship can have diff Connectivity

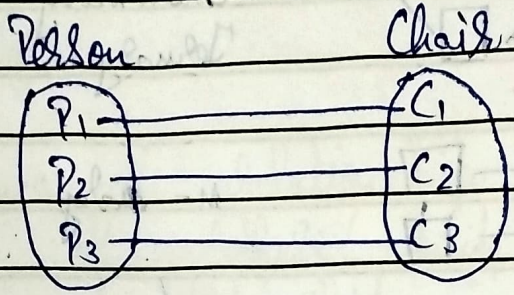
One to one (1:1) → 

One to many (1:M) → 

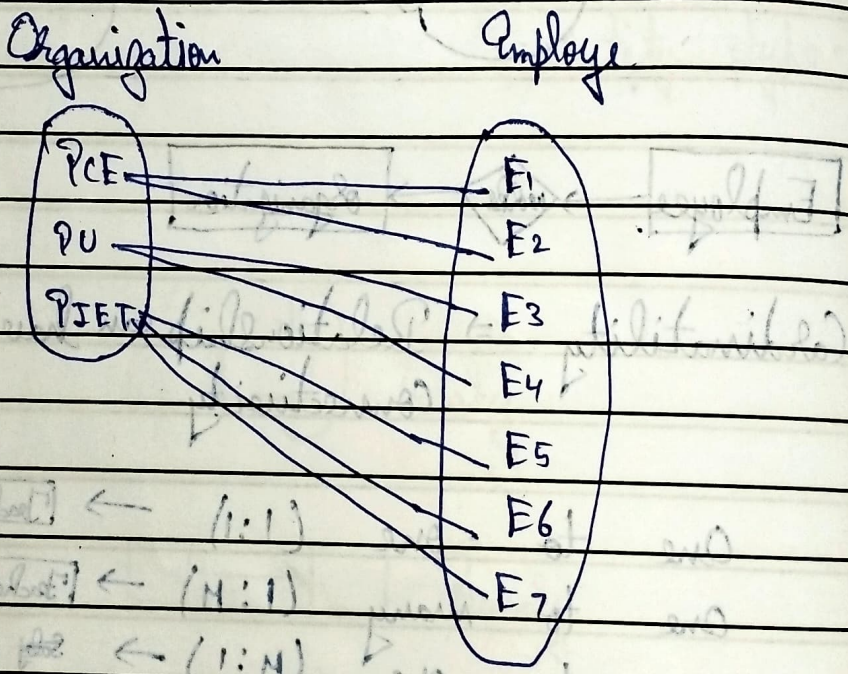
many to one (M:1) → 

many to many (M:N) → Many Student → Many courses

☆ 1:1 ⇒ When 1 instance of entity type have relationship with 1 entity type of another.



☆ 1:M ⇒



← (1:1)
← (M:1)
← (1:N)
← (M:N)

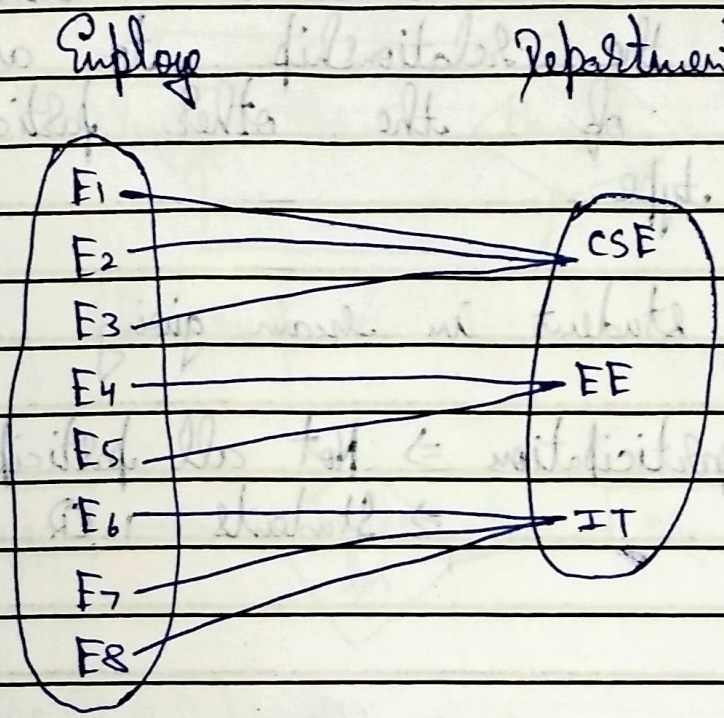
Main Ideas, Questions & Summary:-

Library Ref:-

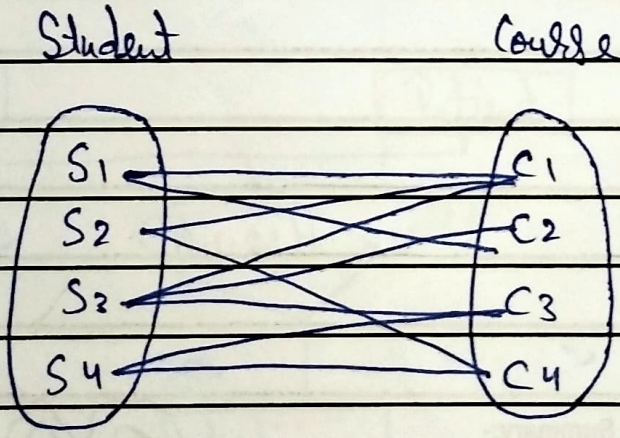
POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

★ M:1 ⇒



★M:M ⇒



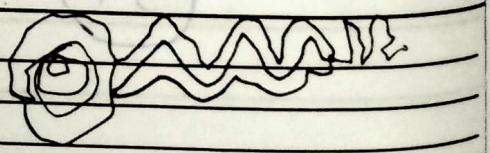
★ Relationship Participation ⇒

① Total participation ⇒ Every entity instances must be connected through the relationship to another instance of the other participating entity type.

Ex ⇒ All student in exam giving

② Partial participation ⇒ Not all participate.
⇒ Students, 1 CR

Main Ideas, Questions & Summary:-

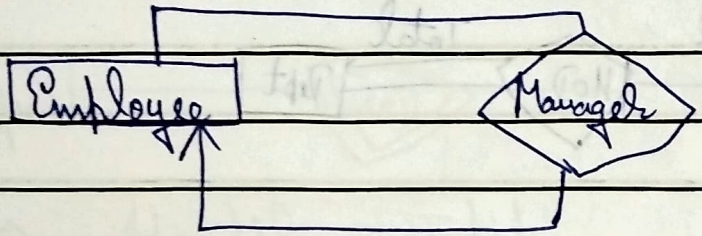


Library Ref:-

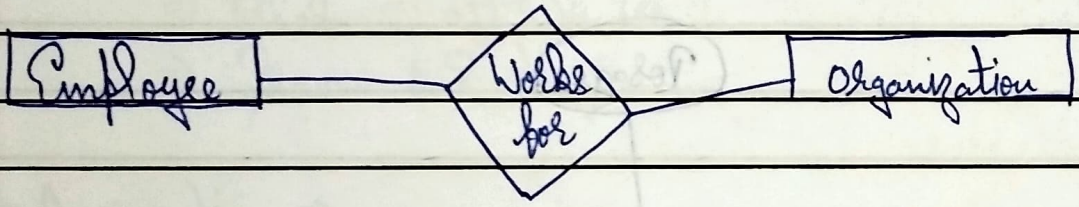
POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

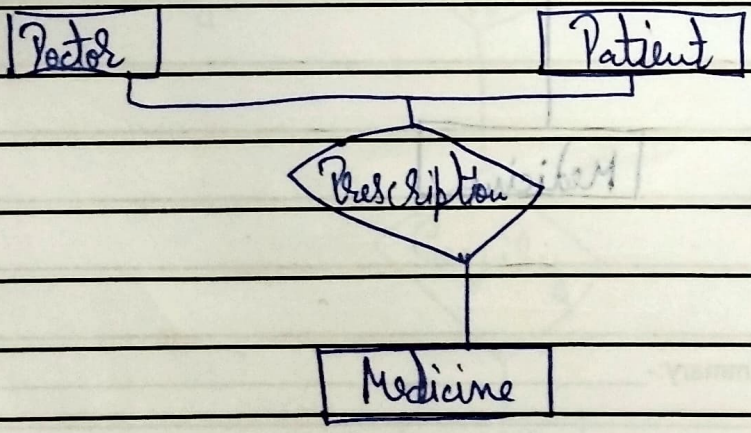
★ Unary relationship ⇒



★ Binary relationship ⇒

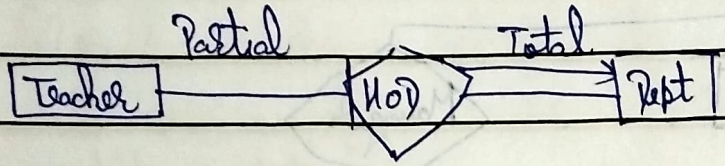


★ Ternary relationship ⇒

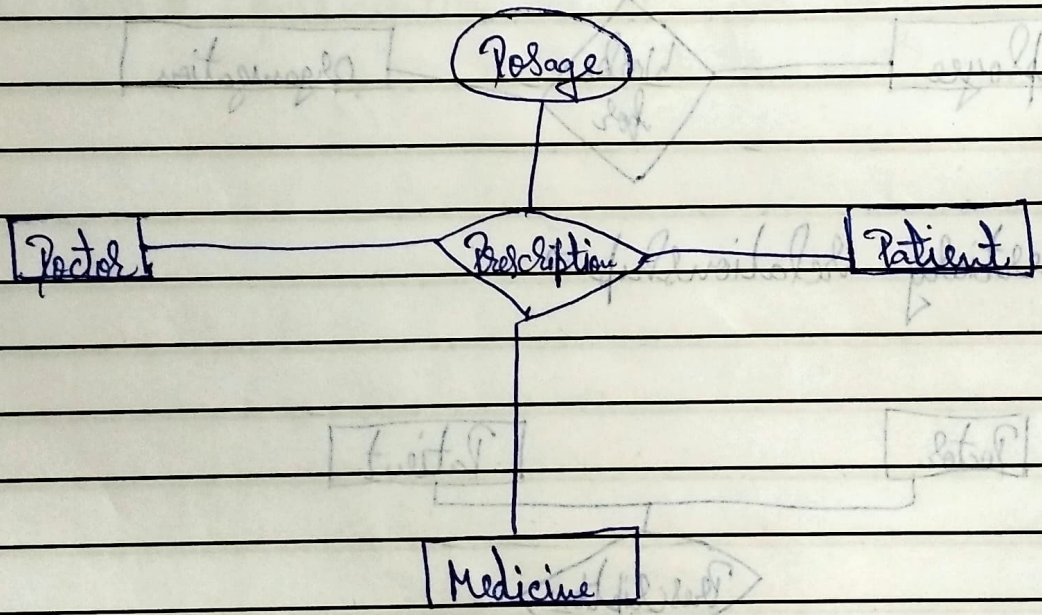


★ Relationship participation ⇒

① Total relation ⇒



★ Attributes of relationship ⇒



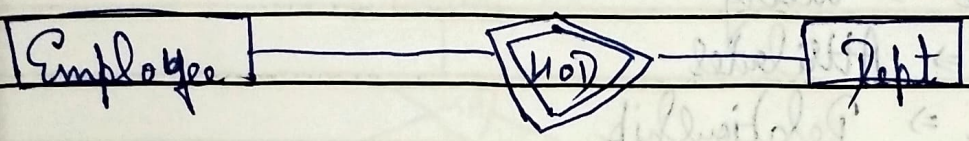
Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

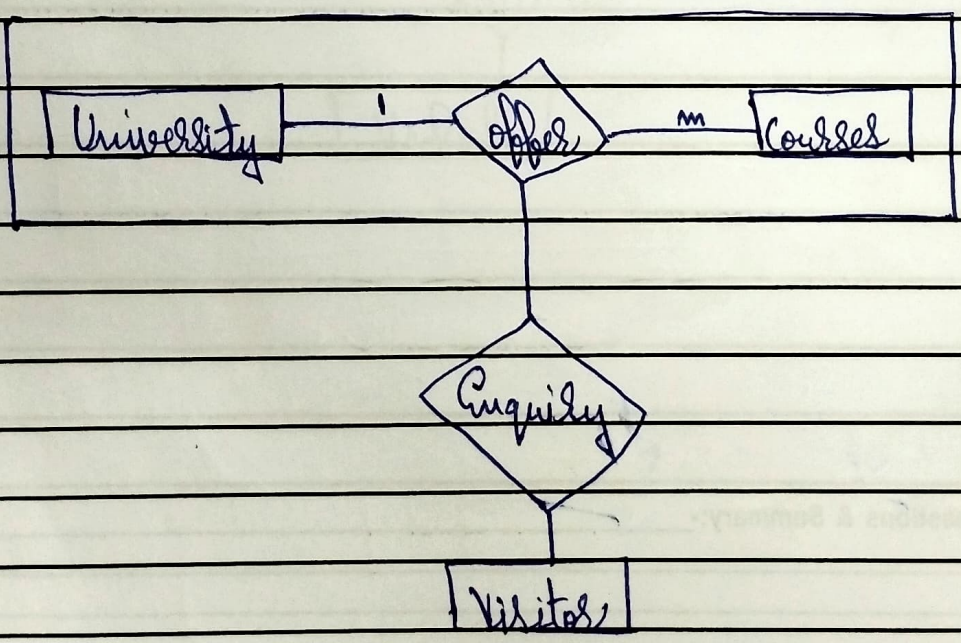
Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Weak Relationship ⇒



- Steps ⇒ Identify entity
  ⇒ Identify key attributes
  ⇒ Identify relationship b/w entity
  ⇒ Other attributes
  ⇒ Draw E-R diagram

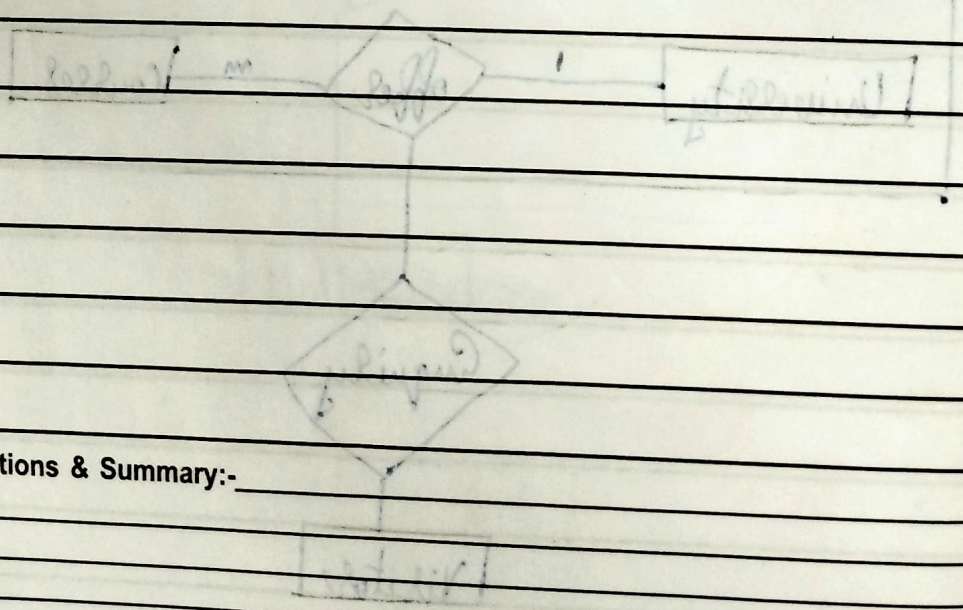
☆ Aggregation ⇒



★ E-R Model ⇒ display database graphically.

• Components ⇒

① Rectangle ⇒ Entity
② Ellipse ⇒ Attributes
③ Diamond ⇒ Relationship
④ Line ⇒ Links b/w entity attributes
⑤ Double ellipse ⇒ Multivalue attributes
⑥ Dashed ellipse ⇒ Derived attribute
⑦ Double line ⇒ Total participation
⑧ Double rectangle ⇒ Weak entity
⑨ Underline ⇒ Primary key



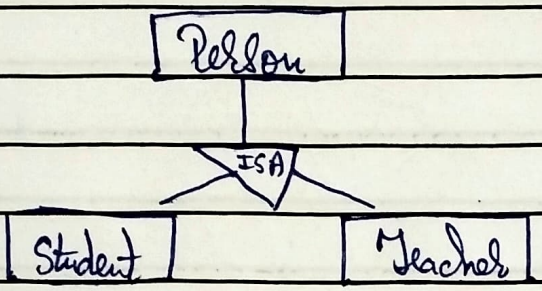
Main Ideas, Questions & Summary:-

Library Ref:-

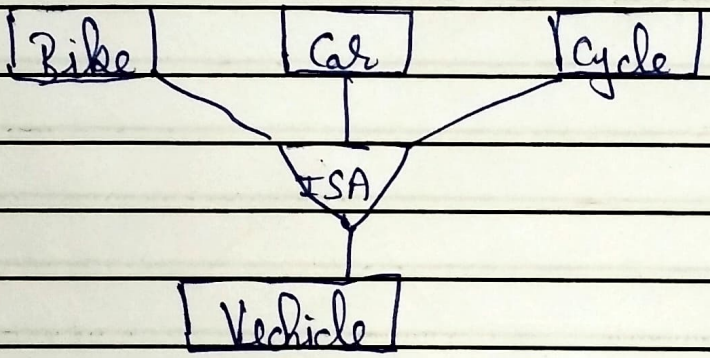
POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Specialisation ⇒ Top to Bottom approach



☆ Generalisation ⇒ Botton to Top



POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Normalization ⇒ Technique of organization data in database.
(Schema Refinement)

(2) Remove or Reduce Redundancy from table.

(3) 2 type duplicacy ⇒ Row level
⇒ Column level

(4) Row level ⇒ Use primary key

	SID	Name	Age
Same → →	1	Ram	18
	2	Mani	25
→ →	1	Ram	18

(5) Column level ⇒ Problems occur like ⇒
(i) Insertion Anomaly
(ii) Deletion Anomaly
(iii) Updation Anomaly

☆ Problem without normalization ⇒ We have data redundency then it will take more space & difficult to handle the database.

☆ 1st Normal form ⇒ Table should not contain any multivalued Attribute.

Roll no	Name	Course
1	Sai	C/C++
2	Harsh	Java
3	Onkar	C/DBMS

Not a 1NF ↑ , convert it

① 1st method ⇒

Roll no	Name	Course
1	Sai	C
1	Sai	C++
2	Harsh	Java
3	Onkar	C
3	Onkar	DBMS

(Roll no + Course) both use as primary key.

Main Ideas, Questions & Summary:- Composite primary key

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

② 2nd method ⇒

	Roll no.	Name	Course 1	Course 1	
	1	Sai	C	C++	
	2	Harsh	Java	NULL	
	3	Onkar	C	DBMS	

Roll no. as primary key

③ 3rd method ⇒

Roll no.	Name
1	Sai
2	Harsh
3	Onkar

Base table

Primary ⇒ Roll no.

Roll no.	Course
1	C
1	C++
2	Java
3	C
3	DBMS

Primary key ⇒ Roll.no + Course

Foreign key ⇒ Roll no.

✱ Functional dependency ⇒

① It is a relationship that exist b/w 2 attributes. Generally exist b/w primary key & non-key attribute.

② Let X & Y 2 attributes of relation. If there is only one attribute value of Y corresponding to it for given value of X, then Y is said to be functionally dependent on X.

③ $X \longrightarrow Y$ (X: determinant, Y: dependent attribute)

④ 2 types ⇒ Trivial functional dependency
⇒ Non-trivial functional dependency

⑤ Trivial ⇒ $A \to B$ is trivial if, B is a subset of A
⇒ Also, $A \to A$, $B \to B$
⇒ $A \cap B \neq \phi$

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

⑥ Non-trivial ⇒ $A \to B$ is non trivial if, B is not a subset of A

⇒ If $A \cap B$ = NULL, $A \to B$ complete non-trivial

★ Types of functional dependency ⇒

① Closure method ⇒

★ Closure of functional dependency ⇒

⇒ Used to find all the candidate key in a table

⇒ denoted by F^+

Qu ⇒ R(ABCD)

FD { $A \to B$, $B \to C$, $C \to D$ }

Ans ⇒ $A^+ = BCDA$

$B^+ = CDB$

$C^+ = DC$

$D^+ = D$

C.K = { A }

$(AB)^+ = ABCD$

S.K = { AB }

not C.K

↳ Member of candidate key

Prime Attribute = A

Non Prime " = B, C, D

Q ⇒ R = (A B C D E)

FD = { A → B , BC → D , E → C , D → A }

Ans As E is not in right side, Combine it all

E^+ = EC

$(AE)^+$ = A B C D E	check right in FD & Replace C.K left
$(BE)^+$ = A B C D E	
$(DE)^+$ = A B C D E	
$(CE)^+$ = ~~A B C D E~~ CE α	

C.K = { AE* , DE , BE }

* Prime Att ⇒ A, B , D, E

Non " " ⇒ C

** If E^+ closure contain all relation then we stop.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

★ Properties of Functional Dependancy ⇒

① Reflexive ⇒ If Y is subset of X then,
$X \to Y$
$\Rightarrow X \supseteq Y$, $X \to Y$

② Augmentation/ Partial dependancy ⇒ If $X \to Y$, then
$XZ \to YZ$ Or,
$XZ \to Y$

③ Transitive ⇒ If $X \to Y$ and $Y \to Z$ then,
$X \to Z$

④ Union ⇒ If $X \to Y$ and $X \to Z$ then,
$X \to YZ$

⑤ Decomposition/ Projective rule ⇒ If $X \to YZ$ then,
$X \to Y$ and $X \to Z$ / Reverse false

⑥ Pseudo Transitive ⇒ If $X \to Y$ and $WY \to Z$ then,
$WX \to Z$

⑦ Composition ⇒ If $X \to Y$ and $Z \to W$ then
$XZ \to YW$

★ 2NF ⇒

① Table must be in 1NF.

② All non-prime attributes should be fully functional dependent on primary key.

③ No partial dependency.

Eg ⇒

Customer ID	Store ID	location
1	1	Delhi
1	3	Mumbai
2	1	Delhi
3	2	Banglore
4	3	Mumbai

Ans Candidate key ⇒ Customer ID Store ID

Prime Att ⇒ Customer ID

⇒ Store ID

Non-Prime Att ⇒ location

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

Customer Id	Store Id
1	1
1	3
2	1
3	2
4	3

Store Id	Location
1	Delhi
2	Banglore
3	Mumbai

A proper subset of candidate key is determining some non-prime attribute.

☆ Partial dependency ⇒ LHS should be proper subset of CK and RHS should be a non-prime Att.

Proper Subset ⇒ A, B.
⇒ Not AB, BA

Q ⇒ R(ABCDEF)
FD { C → F, E → A, EC → D, A → B }

Ans ① Find Candidate key
② Determine Prime & non prime Att
③ Identify partial dependency, If exist not a 2NF [check in FD]

If not in Partial depency than Total dependency

★ 3NF ⇒

① Table must be in 2NF

② There should be no transitive dependency in table

- Transitive dependency ⇒ Non-prime attribute determined by non prime attribute

$A \rightarrow B$, $B \rightarrow C$

A: Prime, B: N.P, B: N.P, C: N.P

⇒ LHS must be a CK or SK OR RHS is a prime attr

Name	City	State

Name → City

City → State

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

Q ⇒ R(ABCD)

FD { AB → CD , D → A }

Ans $B^+ \rightarrow \alpha$

$AB^+ \rightarrow ABCD$ ✓

$DB^+ \rightarrow ABDC$ ✓

CK = { AB , DB }

PA = { A , B , D }

NPA = { C }

** Non partial dependency, So it is 2NF

- AB → CD , D → A

  AB ✓ CK ; CD ; D → A ✗ (Not C.K) ; A ✓ P.A

It is 3NF

★ Boyce Codd Normal form (BCNF) ⇒

① It is a advance version of 3NF.

② It is a stricter than 3NF

③ For every functional dependency X → Y, X is a super or Candidate key.

④ For BCNF, Table should be in 3NF. & every FD, LHS is super key

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Decomposition

- Lossless decomposition
- Dependency Preserving

☆ 3NF always ensures "dependency preserving decomposition" but not in BCNF

☆ Both 3 & BCNF ensures lossless decomposition.

Roll no	Name	Id	Age

Ck = Roll, Id

FD = { Roll → Name
Roll → Id
Id → age
Id → Roll no }

POORNIMA UNIVERSITY UNIT => 4

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Transaction => It is a set of operations used to perform a logical unit of work.

(2) Transaction generally represent change in database

(3) Performed by single user to perform operations for accessing the contents of database.

(4) Each transaction must succed or fail i.e., it can not remain in an intermediate state.

(5) Operations are => Read (access)
=> Write (change)
=> Commit
=> Rollback

(6) Open_Acount(X)
Old_Balance = X.balance
New_Balance = Old_Balance - 800
X.balance = New_Balance
Close_Acount(X)

★ Operations ⇒

① Read (x) ⇒ Used to read value of X from DB & store it in a buffer in main memory.

② Write (x) ⇒ Used to write the value back to database from buffer.

Ex ⇒ R(x);
X = X - 500;
W(x);

③ Commit ⇒ Used to save work done permanently.

④ Rollback ⇒ Used to undo the work done.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

★ Transaction properties ⇒

① ACID ⇒ Atomicity, Consistency, Isolation, Durability

② A database contain these properties in order to ensure accuracy, completeness & data integrity.

③ (i) Atomicity ⇒ This property says that at the end of each transaction, either no changes have been done to database or database has been changed in a consistent manner.

(ii) To make changes permanent, COMMIT is ~~used~~ issued & to abort ROLLBACK is ~~used~~ issued

(iii) If we transfer 500 from A to B, if the transaction is completed, amount of 500 must be deducted from A & same must be added to B. But if transation aborted then neither of balance will be changed.

(4) (i) Consistency ⇒ The integrity constraints are maintained so that the database is consistent before & after the transaction.

(ii) Before transaction start & after completed, sum of money should be same.

(iii) A = 2000 Rs } 5000 Rs
B = 3000 Rs }

T1 ⇒ R(A)
A = A-1000 (1000)
W(A)
R(B)
B = B+1000 (4000)
W(B)
Commit

A = 1000 Rs } 5000 Rs [Sum Same]
B = 4000 Rs }

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

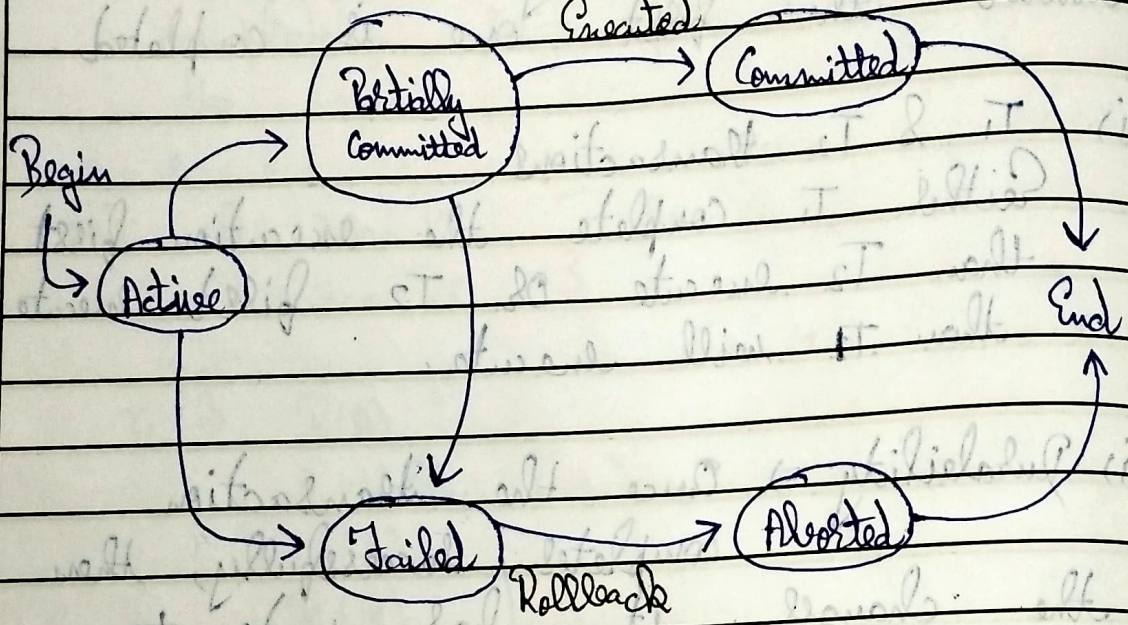
Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

(5) (i) Isolation ⇒ It shows that the data which is used at the time of execution of a transaction cannot be used by second transaction until the first one is completed.

(ii) T_1 & T_2 transactions:
Either T_1 complete the execution first than T_2 execute or T_2 first execute than T_1 will execute.

(6) (i) Durability ⇒ Once the transaction completes successfully, then the changes it has made to database are permanent.

☆ Transaction State / Model ⇒ 5 States



Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

(1) (i) Active state ⇒ First state of every transaction.

(ii) A transaction under execution said to be in active state.

(2) Partially comitted ⇒ In this, transaction executes its final operation, but data is not saved in DB permanently.

(3) Committed state ⇒ In this, all operations are successfully executed and data is permanently saved to DB.

(4) Failed ⇒ In this, the normal execution can't proceed. & this transaction said to be in Failed state.

(5) Aborted ⇒ In this, rollback is occur due to some failure.

☆ Schedule ⇒ A series of operation from one transaction to another transaction is known as schedule.

(2) Used to preserve the order of operation of transaction

(3) It is execution sequence of multiple transaction.

(4) Types ⇒ Serial schedule
⇒ Non Serial Schedule / Concurrent Sch
⇒ Serializable schedule.

(5) It is an order in which the transaction are executed in system.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Serial Schedule ⇒ It is an order in which the transactions are executed serially.

(2) It is schedule where one transaction is executed completely before starting another transaction.

(3) In this first transaction completes then next transaction is executed.

(4) Result in consistent database

(5) Low performance / low throughput
⇒ No of trans executed / time

(6) A with 2000 & B with 1000.
Let 2 transtion T_1 & T_2
T_1 transfers 100 A to B while T_2, 500 A → B
Then $< T_1, T_2 >$ or $< T_2, T_1 >$ are 2 valid schedule

→

T_1	T_2
Read (A);	
A = A - 100;	
Write (A);	
Read (B);	
B = B + 100;	
Write (B);	
	Read (A);
	A = A - 500;
	Write (A);
	Read (B)
	B = B + 500;
	Write (B);

After completion of both transaction, sum of A+B is preserve / same, so it is consistent.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:-

☆ Concurrent schedule ⇒ It is an order in which several transactions are executed concurrently

② High performance / High throughput

③ It can be consistent or inconsistent database.

④

T_1	T_2		
Read (A);		A = 2000	
A = A - 100		A = 1900	
W (A);			
	R (A);	A = 1900	Consistent
	A = A - 500;	A = 1400*	database
	W (A);		
R (B);		B = 1000	
B = B + 100;		B = 1100	
W (B);			
	R (B);	B = 1100	
	B = B + 500;	B = 1600*	
	W (B);		

(5)

T_1	T_2	
R(A);		A = 2000
A = A - 100;		A = 1900
	R(A);	A = 2000
	A = A - 500;	A = 1500 *
	W(A);	
	R(B);	B = 1000
W(A);		
R(B);		
B = B + 100;		B = 1100
W(B);		
	B = B + 500;	B = 1600 *
	W(B);	

Inconsistent database

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

* Serializability ⇒ It is used to check whether the transaction is serial or concurrent. If transaction is concurrent than it will converted to serial schedule.

(2) It is of 2 types ⇒ Conflict serializability
⇒ View serializability

(3) Ex ⇒

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

$T_1 \rightarrow T_2 \rightarrow T_3$

$T_1 \rightarrow T_3 \rightarrow T_2$

$T_2 \rightarrow T_3 \rightarrow T_1$

$T_2 \rightarrow T_1 \rightarrow T_3$

$T_3 \rightarrow T_1 \rightarrow T_2$

$T_3 \rightarrow T_2 \rightarrow T_1$

★ Conflict Equivalent ⇒

R(A)	R(A)	Non conflict pair
R(B)	R(A)	Non conflict pair
W(B)	R(A)	
R(B)	W(A)	
W(A)	W(B)	
R(A)	W(A)	Conflict pair
W(A)	R(A)	
W(A)	W(A)	

Q⇒ Check conflict equivalent

S		S'	
T_1	T_2	T_1	T_2
R(A) ✓		R(A) ✓	
W(A) ✓		W(A) ✓	
	R(A) ✓	R(B)	
	W(A) ✓		R(A) ✓
R(B)	✓ Adjacent check		W(A) ✓

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

Ans

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

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

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

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

$S \equiv S'$

★ If any table have conflict equivalent list, than it will be Serial Schedule.

★ 2 Schedule are said to be conflict equivalent if one Schedule can converted into other Schedule after Swaping non-conflicting operation.

☆ Conflict Serializability ⇒ A schedule is said to be conflict serializability if it is conflict equivalent to serial schedule.

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)		

- Check conflict pairs in other transaction and draw edges.

R → W, W → R, W } Conflicts

☆☆

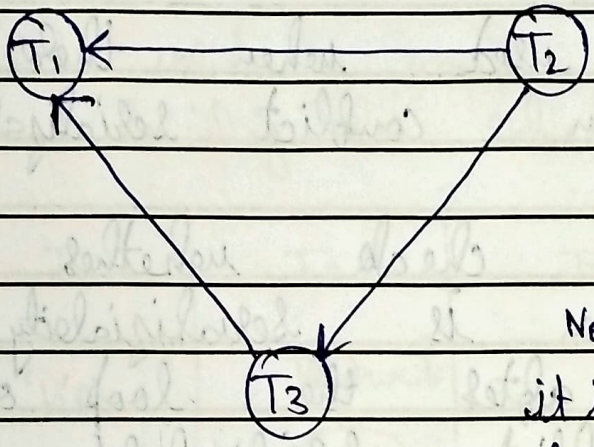
Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Precedence graph =>



If No loop / No cycle exist it is conflict serializable

☆ $T_2 \rightarrow T_3 \rightarrow T_1$

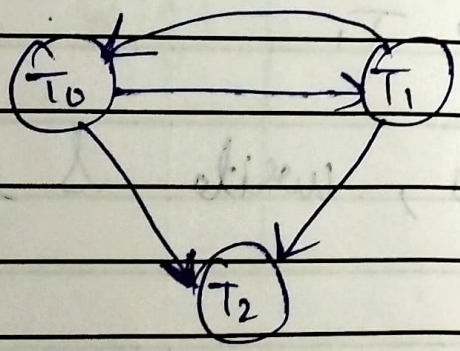
Check, Indegree=0, write & erase it

☆ View Serializability ⇒ If a given schedule is found to be view equivalent to some serial schedule, then called — —.

(2) It is used when there is loop in conflict serializability.

(3) Used to check whether the transaction is serializiality or not after the loop occur in conflict serializability.

(4) Eg ⇒



Main Ideas, Questions & Summary:-

☆ Blind writes
★ View equivalent

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Blind write ⇒ Writing without reading is called as blind write.

- No blind writes means not a view serializable schedule.

Ex ⇒

T_1	T_2	T_3
R(A)	Blind write	
	W(A)	
W(A)		
		W(A)

☆ Recoverability ⇒ Sometime a transaction may not executed completely due to S/W issue, system crash, etc. In that case, failed transactions roll back. But some transaction may also have used value produced by failed transaction. So we also have to rollback those transaction.

T_1	T_2	
R(A)		
A = A - 5		Irrecoverable
W(A)		
	R(A)	
	A = A - 2	
	W(A)	
	Commit	
R(B)		
*fail		

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

★ Types of Recoverable schedule ⇒

(1) Cascading Schedules
(2) Cascadeless "
(3) Strict "

★ Cascading schedules ⇒ If in a schedule, failure of one transaction causes several other dependent transaction to rollback or abort. Then such a schedule called cascading schedule.

Eg ⇒	T_1	T_2
	R(A)	
	W(A)	
		R(A)
		W(A)
	Failure	

T_2 depend on T_1.

☆ Cascadeless Schedules ⇒ If in a schedule, a transaction is not allowed to read a data item until the last transaction that has written it is committed or aborted, then such called — —

Eg ⇒

T_1	T_2	T_3
R(A)		
W(A)		
Commit		
	R(A)	
	R(A)	
	Commit	
		R(A)
		W(A)
		Commit

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

Ex ⇒

T_1	T_2
R(A)	
W(A)	
	W(A) // uncomitted write
Commit	

Cascadless schedule allows only comitted read operation. However it allow uncomitted write operation.

✓
R(A)
W(A)
W(A)

✗✗
R(A)
W(A)
R(A)

✓
R(A)
W(A)
R(B)

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

★ Concurrency Control ⇒ In this, multiple transactions can be executed simultaneously.

(2) It effect the transaction result.

(3) Highly important to maintain the order of execution of those transaction.

★ Problems of concurrency control ⇒

(1) Lost updates
(2) Dirty Read
(3) Unrepeatable Read

☆ Lost update problem ⇒ If 2 transaction T_1 and T_2 read a record and then update it, then the effect of updating of first record will be overwritten by second update.

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

☆ Dirty Read ⇒ Dirty read occur when, one transaction updates an item of database, and then transaction fails for some reason. The updated database item is accessed by another transaction before it is changed back to original value.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

⇒ T_1 updates a record which is read by T_2. If T_1 aborts then T_2 now has values which have never formed part of stable database.

Ex ⇒

T_1	T_2
W(A)	
	R(A)
ROLL BACK	
—	—

☆ Inconsistent Retrievals problem ⇒

① Also known as unrepeatable read.

② T_1 reads a records and then does some other processing during which the transaction T_2 updates the record. Now when the transaction T_1 reads the record, then new value will be inconsistent with previous value.

Ques => A = 200 B = 250 C = 150

T_1	T_2
R(A)	
Sum = A (200)	
R(B)	
Sum = Sum + B(450)	
	R(c)
	C = C - 50 (100)
	W(c)
	R(A)
	A = A + 50 (250)
	W(A)
	COMMIT
R(C)	
Sum = Sum + C (550)	
Initial = 600	
After = 550	

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

☆ Concurrency control protocol ⇒ It ensures atomicity, isolation & serializability of concurrent transactions.

② Types ⇒ lock based protocol
⇒ Time stamp "
⇒ Validation based "

☆ Lock based protocol ⇒ In this, any transaction cannot read or write data until it acquires an appropriate lock on it.

① Shared lock ⇒ Also known as read only lock. In this, the data item can only read by transaction

② Exclusive lock ⇒ In this, data item can be both read as well as written by transaction.

⇒ Multiple transaction do not modify the same data simultaneously.

★ 4 types of lock protocols ⇒

① Simplistic lock protocol ⇒

(i) Simplest way of locking data

(ii) Allow transaction to get the lock on data before insert or delete or update.

(iii) Unlock data after completing transaction.

② Pre-claiming lock protocol ⇒

(i) In this, we make a list of all data items on which we have to apply lock.

Main Ideas, Questions & Summary:-

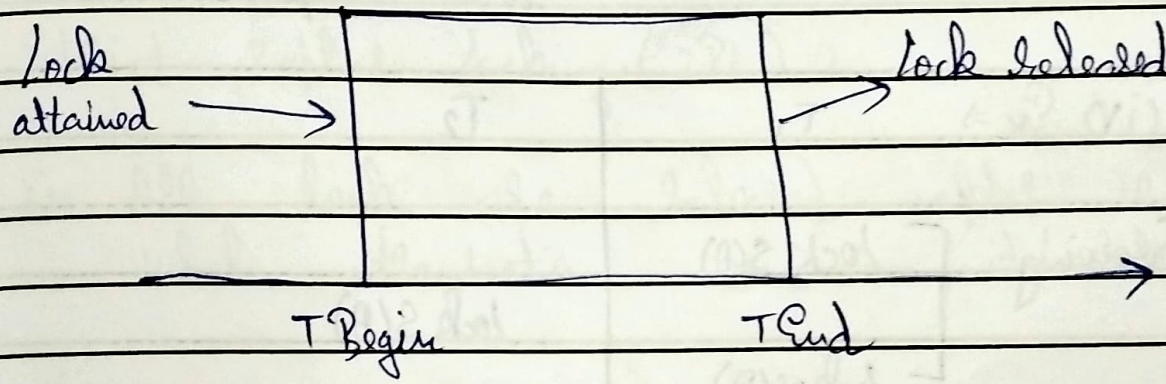
Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

(ii) If all lock are granted then this protocol allows transaction to begin. After transaction complete then it releases lock.

(iii) If lock not granted, roll back state.



(3) 2-phase locking (2PL) ⇒

(i) Divides execution phase into 3 parts.

(ii) Growing phase ⇒ locks are acquired & no locks are released.

(iii) Shrinking phase ⇒ locks are released and no locks are acquired.

(iv) Ex ⇒

	T_1	T_2
Growing	lock S(A)	
		lock S(A)
	lock X(B)	
Shrinking	Unlock (A)	
		lock X(C)
	Unlock (B)	
		Unlock (A)
		Unlock (C)

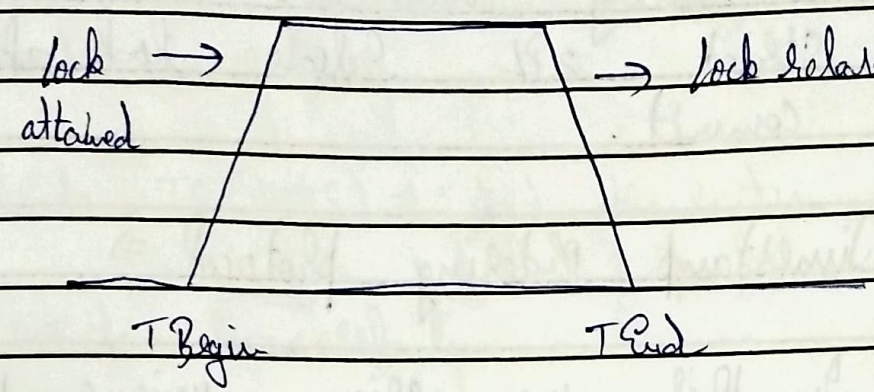
Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

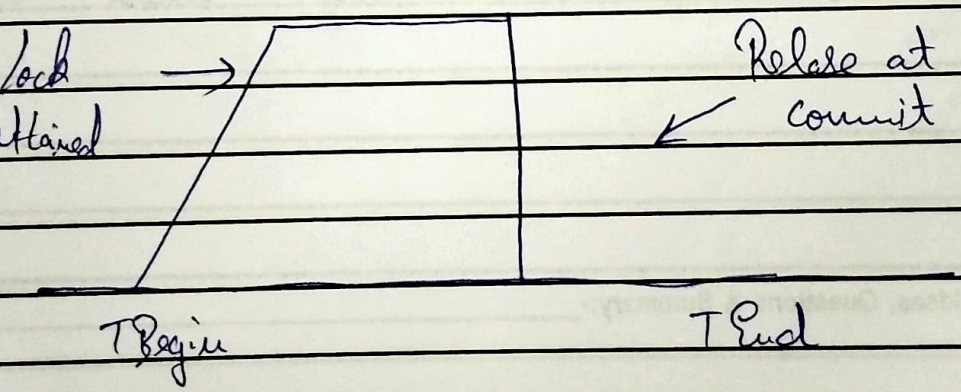
(iv)



(4) Strict 2-phase locking (Strict 2PL) ⇒

(i) All locks are released after the whole transaction is commit/abort.

(ii)



Q ⇒ Difference b/w 2PL & Strict 2PL ?

Ans 2PL release lock while executing transaction where as strict 2PL release lock after commit.

☆ Timestamp ordering protocol ⇒

① In this we assign unique value to every transaction.

② This we tells the order of transaction, when did it enter into system.

③ The older transaction have lower value than latest transaction.

Main Ideas, Questions & Summary:-

Library Ref:-

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

(4) Read - TS (RTS) ⇒ last transaction no which performed read successfully

(5) Write - TS (WTS) ⇒ last transaction no. which performed write successfully.

(6) TS (T_i)
↓
Time Stamp → Transaction no., T_1, T_2.

(7) Rules ⇒

☆ Validation based protocol ⇒

① Also known as optimistic concurrency control technique.

② Read phase ⇒ Transaction T is read and executed. Used to read the value of various data item.

③ Validation phase ⇒ Temporary variable value will be validated against the actual data to see if it violates the serializibility.

④ Write phase ⇒ The temporary results are written to the database or otherwise transaction is rolled back.

Main Ideas, Questions & Summary:-

Library Ref:-

POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

(5) Each phase have diff timestamp

(i) Start (Ti) ⇒ It contains the time when Ti started its execution.

(ii) Validation (Ti) ⇒ Contains time when Ti finished its read phase & start its validation phase

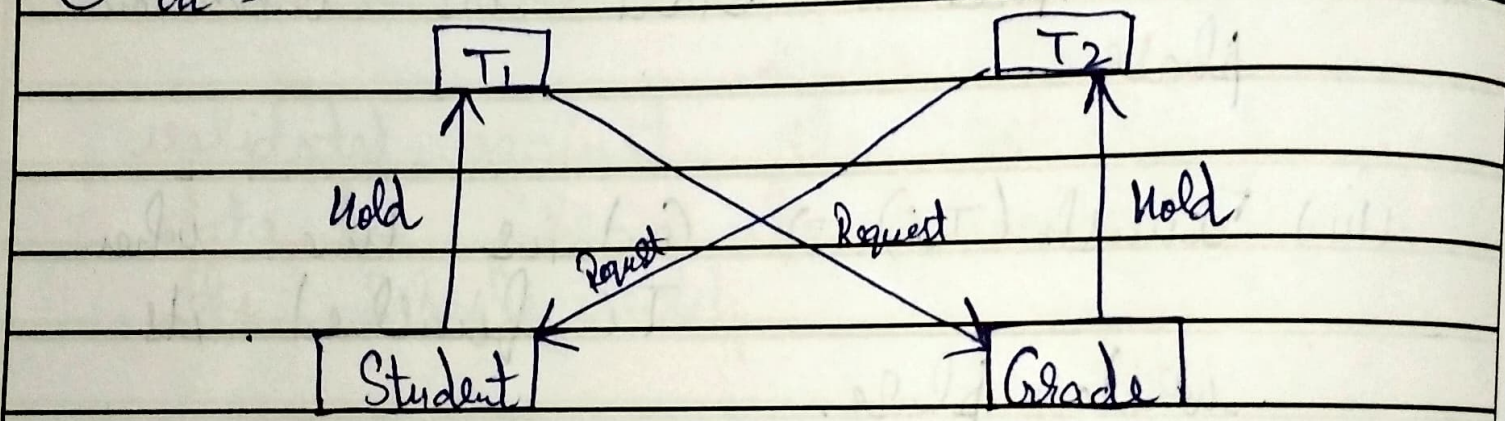
(iii) Finish (Ti) ⇒ Contains time when Ti finished its write phase.

☆ Deadlock handling ⇒

① It is condition where 2 or more transaction are waiting indefinietly for one another to give up locks.

② It is said to be one of the most feared complications

③ Eg ⇒



POORNIMA UNIVERSITY

Date	Unit No.	Lecture No.	Name of Faculty	Subject Name & Code	Main Topics:-	Page no.:

* Deadlock Avoidance ⇒

① When a database is stuck in a deadlock state, then it is better to avoid rather than aborting.

② Method like "Wait for graph" is used for detecting the deadlock situation but only suitable for smaller DB.

* Deadlock Detection ⇒