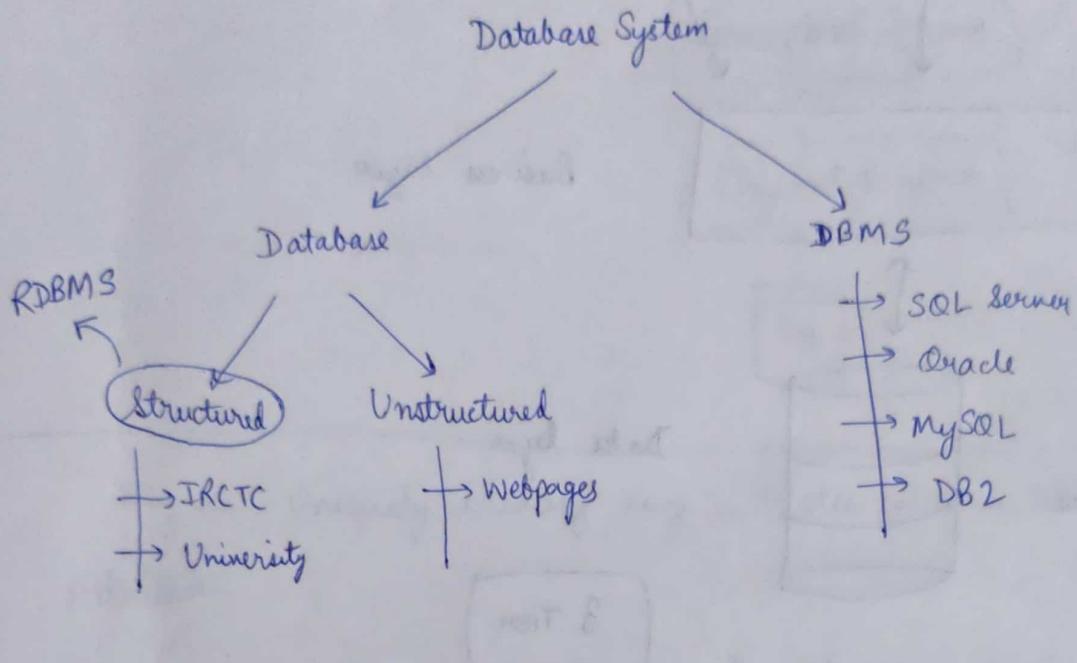


DBMS

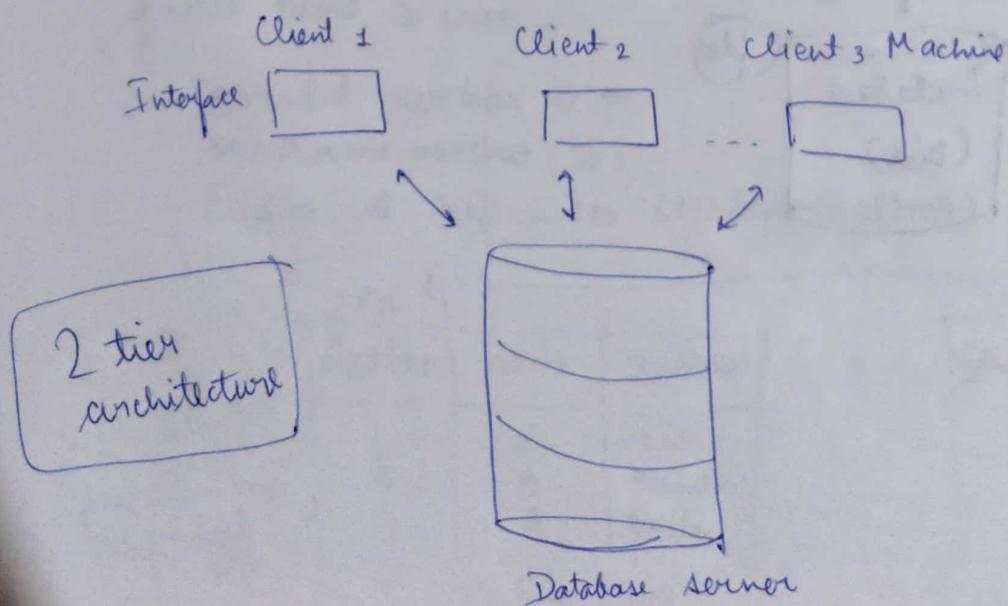


Database → Collection of Related Data

RDBS → Relational Database Management System

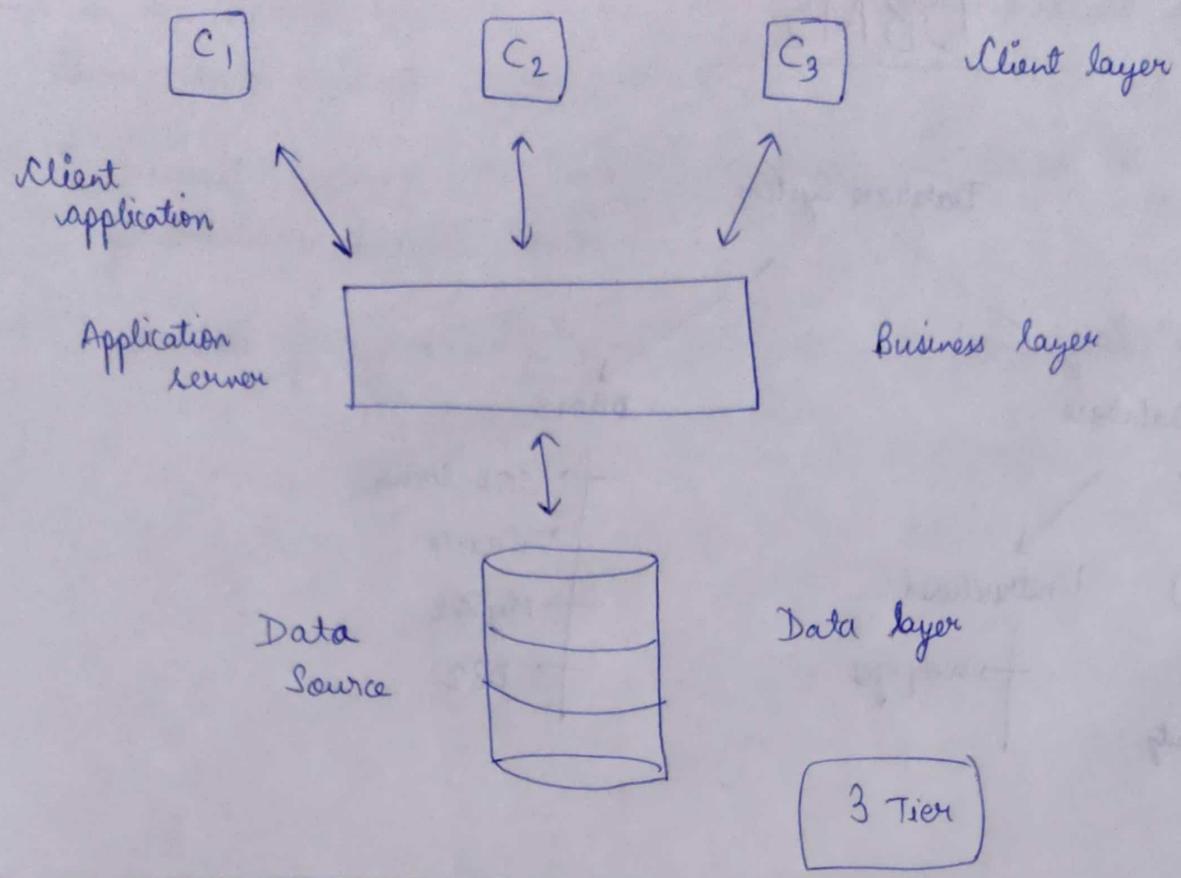
Files v/s DBMS

1. Unnecessary memory fetched via files
2. No attribute required in DBMS for fetching data
3. Concurrency in DBMS
4. Security in DBMS (Role based security)
5. Redundancy

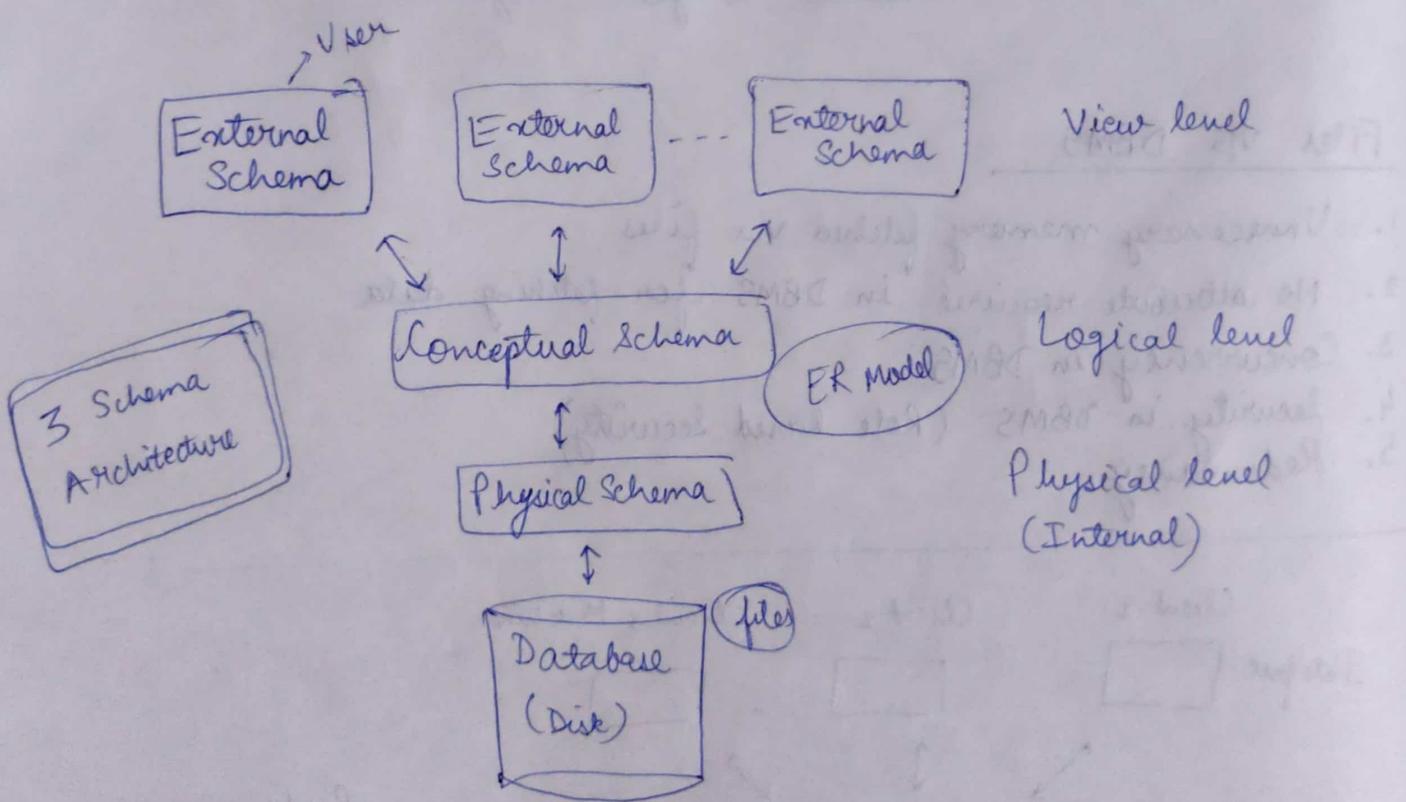


Problems :

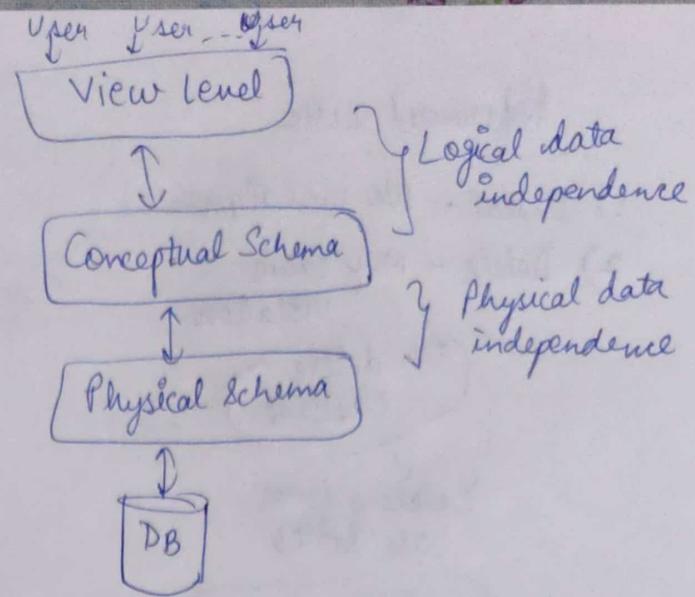
- Scalability
- Security



Schema \rightarrow logical representation of data



Data Independence



Key → Uniquely identify any 2 tuples in the table
 ↓
 Attribute

Set of attributes which uniquely identify → Candidate key

Primary key → one attribute chosen from candidate key

Alternative key → Remaining attributes

PRIMARY KEY = { Unique + Not Null }

Foreign Key → It is an attribute or set of attributes that references to Primary Key of same table or another table (relation)
 ↳ Maintains referential integrity

create table course

(
 Courseid varchar(10),
 Course_name varchar(20),
 Rollno int references Student(Rollno)
);

Student

Rollno	name	address
1	A	Delhi
2	B	Mumbai
3	A	Noida

Referenced Table

Course

f.k.

Course-id	Course-name	Rollno
C_1	DBMS	1
C_2	OS	2
:	:	

Base Table

Referencing Table

Referenced Table

- 1) Insert - No violation
- 2) Delete - May cause violation

(On delete cascade)

Deleting from all tables

(On delete set Null)

→ can be a problem

- 3) Updation → May cause violation

Referencing Table

- 1) Insert → May cause violation
- 2) Delete → no violation
- 3) Updation → May cause violation

SuperKey → Combination of all possible attributes which uniquely identify two tuples in a table.

Superset of any candidate key is Superkey.

Candidate Key = Roll no.

Roll no.	name	age

S.K. = Roll no., name, age
 Roll no., name
 Roll no., age

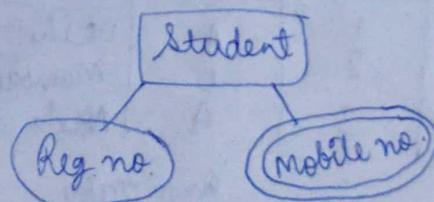
Entity Relationship Model (ER Model)

Entity → Any object which has physical existence

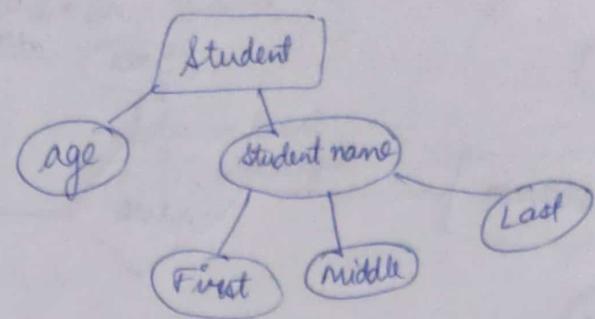
Attribute → types

Types of attributes

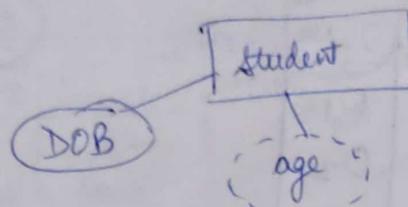
- 1) Single vs Multivalued Attributes



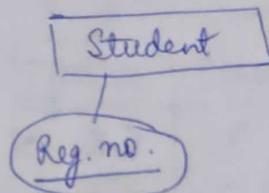
2) Simple vs Composite attributes



3) Stored vs Derived attributes



4) Key vs Non-Key attributes



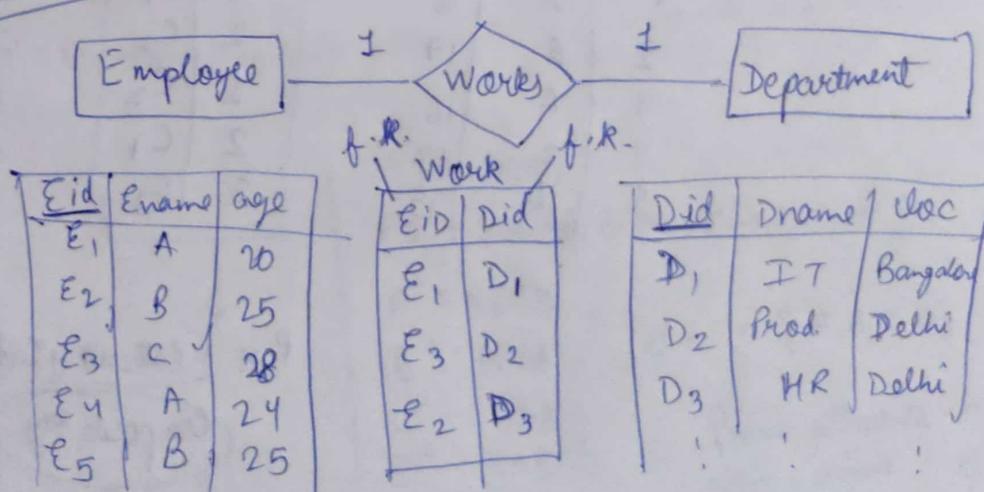
5) Required vs optional attributes

6) Complex attribute → Composite + multivalued

Degree of Relationship (Cardinality)

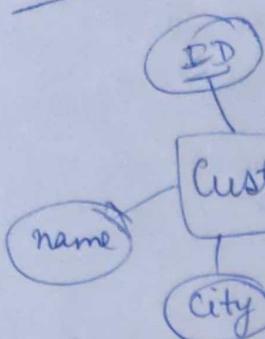
1-1
1-M
M-1
M-N

One-to One



P.K. = Either Eid or Did

One-to-many relationship

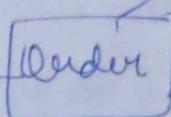


→ Descriptive attribute

Date

Give

M



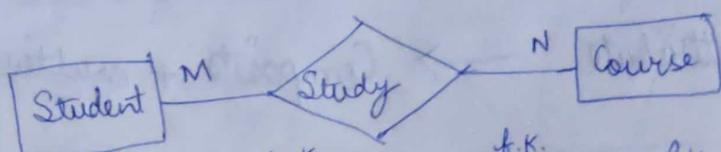
ID	Name	City
C ₁	A	Bangalore
C ₂	B	Delhi
C ₃	C	Mumbai
C ₄	A	Mumbai

ID	O-No	Date
C ₁	O ₁	
C ₁	O ₂	
C ₂	O ₃	
C ₂	O ₄	

O-no.	Item_name	Cost
O ₁	Bucket	100
O ₂	Shoes	200
O ₃	Shirt	150
O ₄	Jeans	200

P.K. → O-No.

Many-to-Many relationship
(M-N)



Roll.no.	name	age
1	A	16
2	B	17
3	A	16
4	C	17
5	D	15

Roll.no	Cid
1	C ₁
2	C ₂
1	C ₂
2	C ₁
3	C ₃

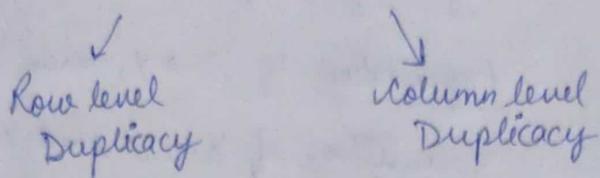
C-id	name	credit
C ₁	Maths	4
C ₂	Phy.	4
C ₃	Chem.	4
C ₄	Hindi	4

P.K. = Roll_no + Cid

Composite Key

Normalization

→ It is a technique to remove or reduce redundancy from a table.



First Normal Form

→ Table should not contain any multivalued attribute.

Roll	Name	Course
1	Sai	C/C++
2	Harsh	Java
3	Onkar	C/DBMS

→ Not in
1st NF

Solution to
convert in
1st N.F.

P.K.

Rollno	Name
1	Sai
2	Harsh
3	Onkar

+

Base Table

F.K.

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

Primary Key → Roll + Course

Closure Method

→ Candidate Key

R(ABCD)

FD {A → B, B → C, C → D}

A⁺ = BCDA

B⁺ = BCD

C⁺ = CD

D⁺ = D

C.K = {A}

Prime attribute = A

Non-prime = B,C,D

R(ABCD)

FD = {A → B, B → C, C → D, D → A}

A⁺ = ABCD

B⁺ = BCDA

C⁺ = CDAB

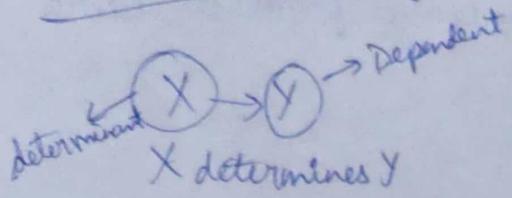
D⁺ = DABC

C.K. = {A,B,C,D}

Prime attribute = A,B,C,D

Non-prime = ∅

Functional Dependency



Reflexivity: if Y is subset of X then $X \rightarrow Y$

Augmentation: if $X \rightarrow Y$, then $XZ \rightarrow YZ$

Transitive: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

Union: if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$

Decomposition: if $X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$

Pseudo transitivity: if $X \rightarrow Y$ and $WY \rightarrow Z$
then $WX \rightarrow Z$

Composition: if $X \rightarrow Y$ and $Z \rightarrow W$ then
 $XZ \rightarrow YW$

Second Normal Form

- table or relation must be in 1st Normal form
- All the non-prime attributes should be fully functional dependent on candidate key.
- There should be no partial dependency in the relation.

Third Normal Form

- table or relation must be in second normal form.
- there should be no transitive dependency in table.

BCNF (Boyce Codd Normal Form)

- table is in 3rd NF

- LHS of each FD should be candidate key or Super Key

1 st Normal Form	2 nd Normal Form	3 rd Normal Form	BCNF	4 th Normal Form	5 th Normal Form
<ul style="list-style-type: none"> * No multivalued attribute * Only single valued <p>eg:</p> <p style="text-align: center;">↓ separate columns</p>	<ul style="list-style-type: none"> * In 1st NF + * NO partial dependency * Only full dependency <p>(AB) → C Non Unique</p> <p>$B \rightarrow C \quad X$ $A \rightarrow C \quad X$ ↓ not full dependency</p>	<ul style="list-style-type: none"> * In 2nd N.F. + * NO transitive Dependency * Non-prime should determine non-prime <p>$X \rightarrow Y \rightarrow Z$</p>	<ul style="list-style-type: none"> * In 3rd NF. + * L.H.S. must be C.K. or S.K. <p>$X \rightarrow Y$</p>	<ul style="list-style-type: none"> * In BCNF + * No multivalued Dependency <p>$X \rightarrow Y$</p>	<ul style="list-style-type: none"> * In 4th N.F. + * Lossless Decomposition

Find minimal cover for following:

$$\{ A \rightarrow B, C \rightarrow B, D \rightarrow ABC, AC \rightarrow D \}$$

Step 1

$$\begin{array}{ccccccc} A \rightarrow B & C \rightarrow B & D \rightarrow A & D \rightarrow B & D \rightarrow C & AC \rightarrow D \\ \times & \times & \times & \times & \times & & \times \end{array}$$

Step 2 Removing redundant

$$\{ A \rightarrow B \\ C \rightarrow B \\ D \rightarrow A \\ D \rightarrow C \\ AC \rightarrow D \}$$

$$\begin{aligned} A^+ &= A \\ C^+ &= C \\ D^+ &= BCD \\ D^+ &= ABCD \\ D^+ &= DAB \\ AC^+ &= ACB \end{aligned}$$

Step 3

$$\begin{array}{l} A \rightarrow B \\ C \rightarrow B \\ D \rightarrow A \\ D \rightarrow C \\ AC \rightarrow D \end{array} \Rightarrow D \rightarrow AC$$

$$\begin{array}{l} C^+ = C \\ A^+ = AB \end{array}$$

JOINS

- + Cross Join
- Natural Join
- Conditional Join
- Equi Join
- Self Join
- + Outer Join
 - Left
 - Right
 - Full

Join = Cross Product + Condition

E-No	E-name	Address
1	Ram	Delhi
2	Varen	Chd
3	Ravi	Chd
4	Amrit	Delhi
5	Nitesh	Noida

'Employee'

Dept No.	Name	E.no
D1	HR	1
D2	IT	2
D3	MRKT	4
D4	Finance	5

'Department'

Q. find E-name of Emp whose working in HR dept.

Select E-name from Emp, Dept where

Emp.Eno = Dept.Eno;

O/P:

Ram

Varen

~~Amrit~~

Nitesh

Self Join

Find Student Id who is enrolled in at least 2 courses.

Student	S_id	C_id	Since	Course	
				F.K	F.K
	S1	C1	2016		
	S2	C2	2017		
	S1	C2	2017		

Study

Select ~~T1.Sid~~ from Study as T1,
Study as T2

where T1.Sid = T2.Sid and

T1.Cid <> T2.Cid;

O/P:

S1

Equi Join

Find the Emp name who worked in a department having location same as their address?

Emp

E-no	E-name	Address
1	Ram	Delhi
2	Varen	Chd
3	Rani	Chd
4	Amrit	Delhi

Dept

Dept No	Location	E-no
D ₁	Delhi	1
D ₂	Pune	2
D ₃	Patna	4

Select E-name from Emp, Dept

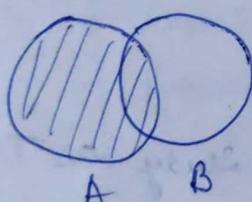
where Emp.E-no = Dept.E-no AND

Emp.Address = Dept.Location;

O/P: RAM

Left Outer Join

It gives the matching rows and the rows which are in left table but not in right table.



Emp

Emp-no	E-name	Dept-no
E ₁	Varen	D ₁
E ₂	Amrit	D ₂
E ₃	Rani	D ₁
E ₄	Nitesh	-

Dept

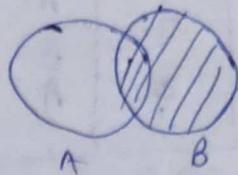
Dept-No	D-name	Loc
D ₁	IT	Delhi
D ₂	HR	Hyd
D ₃	Finance	Pune

Select emp-no, e-name, d-name, loc from Emp left outer Join dept on

emp.dept-no = dept.dept-no;

Right Outer Join

It gives the matching rows and the rows which are in right table but not in left table.



Emp

Emp-no	E-name	Dept-no
E ₁	Varun	D ₁
E ₂	Amrit	D ₂
E ₃	Ravi	D ₃

Dept

Dept-no	D-name	Loc
D ₁	IT	Delhi
D ₂	HR	Hyd
D ₃	Finance	Pune
D ₄	Testing	Noida

Select emp-no, e-name, d-name, loc from emp
right outer joins dept on
emp.dept-no = dept.dept-no;

O/P:

Emp-no	E-name	D-name	Loc
E ₁	Varun	IT	Delhi
E ₂	Amrit	HR	Hyd
E ₃	Ravi	Finance	Pune
-	-	Testing	Noida

Relational Algebra

→ Procedural Query Language

What to do
How to do

Operators

Basic operator

→ Projection (Π)

→ Selection (σ)

→ Cross Product (\times)

→ Union (\cup)

→ Rename (ρ)

→ Set Difference (-)

Derived operator

→ Join (\bowtie)

→ Intersect (\cap)

$$X \cap Y = X - (X - Y)$$

→ Division (\div)

Projection (Π)

Roll No	Name	Age
1	A	20
2	B	21
3	A	19

Student

Query: Retrieve the roll no from table (Student)



Π_{roll} (Student)

$\Pi_{\text{roll, name}}$ (Student)

→ Fetching 2 columns
roll, name

Selection (σ)

Query: Retrieve the name of student whose Rollno = '2'

$\Pi_{\text{name}} (\sigma_{\text{Rollno} = '2'} \text{ (student)})$

Selection → Rows
Projection → Columns

($R_1 \times R_2$)

Cross Product

R_1

	A	B	C
1	2	3	
2	1	4	

R_2

	C	D	E
3	4	5	
2	1	2	

A	B	C	C	D	E
1	2	3	3	4	5
1	2	3	2	1	2
2	1	4	3	4	5
2	1	4	2	1	2

Set difference (-)

$A - B = A$ but not B

$$= A \cap B'$$

Roll no	Name
1	A
2	B
3	C

(Student)

Emp. No	Name
7	E
1	A

(Employee)

→ No. of columns must be same in number.

→ Domain of every column must be same

Rollno	Name
2	B
3	C

$$(\pi_{\text{Name}}(\text{Student}) - \pi_{\text{Name}}(\text{Employee}))$$

→ B, C

Union

→ No. of columns must be same in number

→ Domain of every column must be same

$$(\text{Student}) \cup (\text{Employee})$$

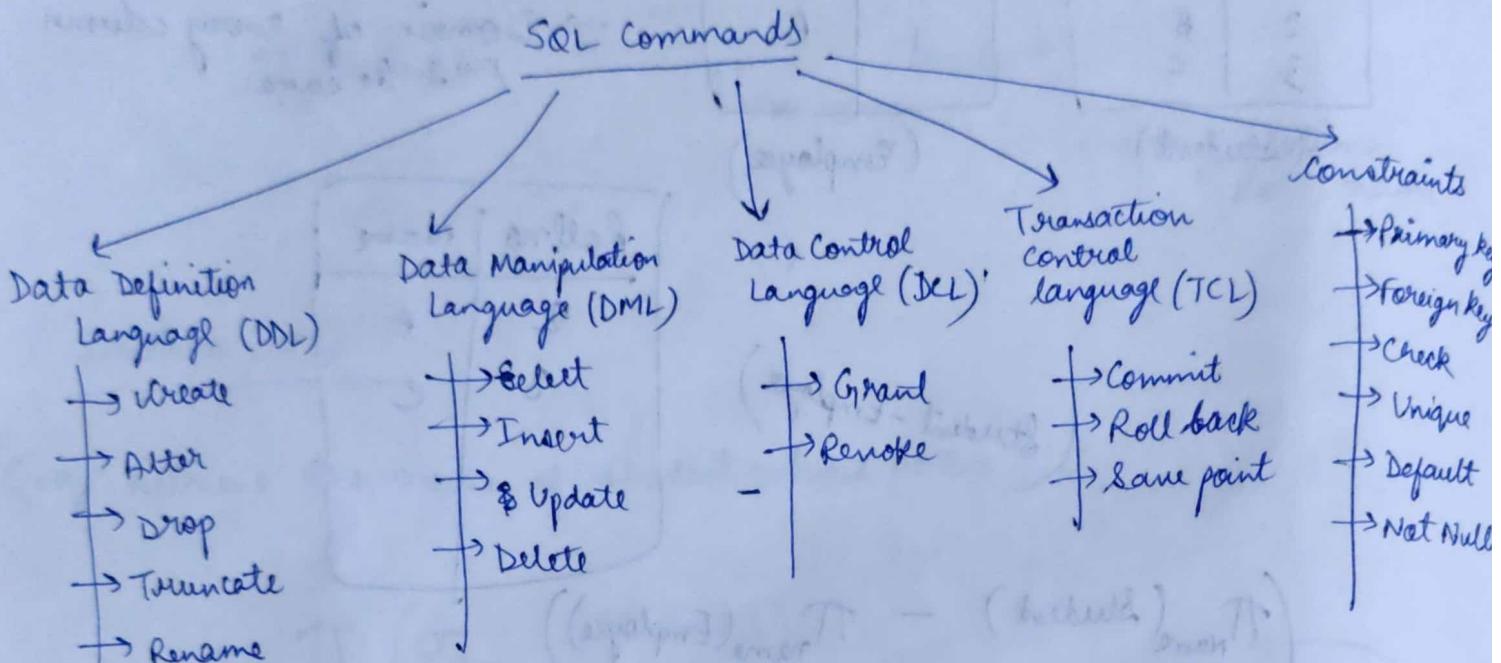
Roll	Name
1	A
2	B
3	C
7	E

↓
↓
↓
↓
↓

Division Method

$/, \div$
(Derived)

$A(X, Y) / B(Y) = H$ results X values for
that there should be a tuple $\langle X, Y \rangle$ for
every \bar{Y} value of relation B .



Create table <table-name>

(
 Col1 name datatype,
 Col2 name datatype,
 Col3 name datatype
);

desc table-name;

Create table emp

(
 id int,
 name varchar(20),
 salary number(10)
);

desc emp;

Alter Command

(Add) Alter table Student
add address Varchar(30);

(Modify) Alter table Student
modify id Varchar(10);

(Drop) Alter table Student
drop column address;

(Rename) Alter table Student
rename column id to roll_no;

(Rename table)

Alter table student
rename to stdt;

Alter table student
add primary key (roll_no);

Alter

DDL

Update

DML

Update Emp

Set salary = salary * 2

where id = 1;

Delete

DML

Drop

DDL

Truncate

DDL

Delete from table_name

Drop table
Student;

Truncate Student;

(Deletes all rows
in 1 go)

Constraints in SQL

1) Unique

2) NOT NULL

3) Primary Key = Unique + Not Null

4) Check

Eg: Check (age > 18)

5) Foreign Key

6) Default

Eg: Salary int default 10000;

Emp

E_id	E-name	Dept	Salary
1	Ram	HR	70000
2	Amrit	MRKT	20000
3	Ravi	HR	30000
4	Zahil	MRKT	40000
5	Varan	IT	50000

Q.1

Write a SQL query to display maximum salary from Emp table.

Select max(Salary) from Emp;

Q.2 Write a SQL query to display employee name who is taking maximum salary.

Select E-name from Emp where Salary = (Select max(Salary) from Emp);

Q.3 Write a SQL query to display second highest salary from Emp table?

~~Select Salary from Emp
Order by DESC~~ Select ^{Max} (Salary) from Emp
where Salary <> (Select max(Salary) from Emp);

Q.4 Write a SQL query to display employee name who is taking second highest salary?

Select E-name from Emp where
Salary = (Select max(Salary) from Emp where
Salary <> (Select max(Salary) from emp));

Q.5 Write a query to display all the dept names along with number of employees working in that dept.

Select Dept, count(*) from Emp
group by Dept;

Q.6 write a query to display all the dept names where number of employees are less than 2.

Select dept from emp
group by dept
having count(*) < 2;

Q.7 Write a query to display highest salary department wise and name of emp who is taking that salary.

Select E-name from emp
where salary IN (select max(salary) from
Emp group by dept));

P.K.		
Eid	Ename	Address
1	Ravi	Chd
2	Varun	Delhi
3	Nitin	Pune
4	Robin	Bangalore
5	Ammy	Chd

Q.1 Detail of emp whose address is either Delhi or Chd or Pune.

Select * from emp
where Address in
('Delhi', 'Chd', 'Pune');

Q.3 Find the detail of emp who is working on at least one project.

Select * from Emp where
Eid Exists (select eid from
Project where
Emp.Eid=Project.Eid);

Q.2 Find the name of Emps who are working on a project.

Select Ename from Emp where
Eid In (select Distinct(Eid) from
Project);

Aggregate functions

Max, Min, Count, Avg, Sum

Select Max(Salary) from Emp;

Select Min(Salary) from Emp;

Select Count(*) from Emp;

Select Sum(Salary) from Emp;

Select Avg (Salary) from Emp;

Correlated & Subquery

→ It is a subquery that uses values from outer query

→ Top down approach

Q. Find all employees detail who work in a department.

Select * from Emp where

exists (Select * from Dept where Dept.Eid = Emp.Eid);

Q. Find Nth highest Salary using SQL.

Select id, salary from Emp e1

where N-1 = (Select count(Distinct(Salary))

from Emp e2

where e2.salary > e1.salary);

PL-SQL (Procedural SQL)

Function

Procedure

Trigger

Cursor

Transaction

It is a set of operations used to perform a logical unit of work.

A transaction generally represent change in database.

Operations → Read, Write

ACID Properties



Atomicity Consistency Isolation Durability

Atomicity → Either all or none

Consistency → Before transaction starts and after the transaction completed, sum of money should be same

Isolation → Multiple transactions running parallel into serial schedule

Durability → All changes should be permanent

Transaction States



Schedule → It is chronological execution sequence of multiple transactions

Serial / Parallel

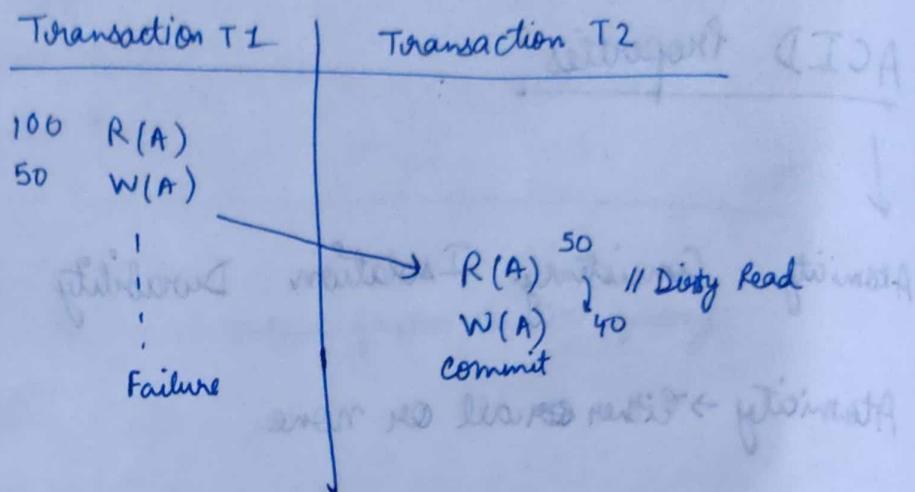
Serial	Parallel
$T_1 T_2 T_3$	$T_1 T_2 T_1 T_3 T_2 T_1 \dots$

Performance

Throughput = No. of transactions executed
per time

Types of problems in concurrency

1) Dirty Read

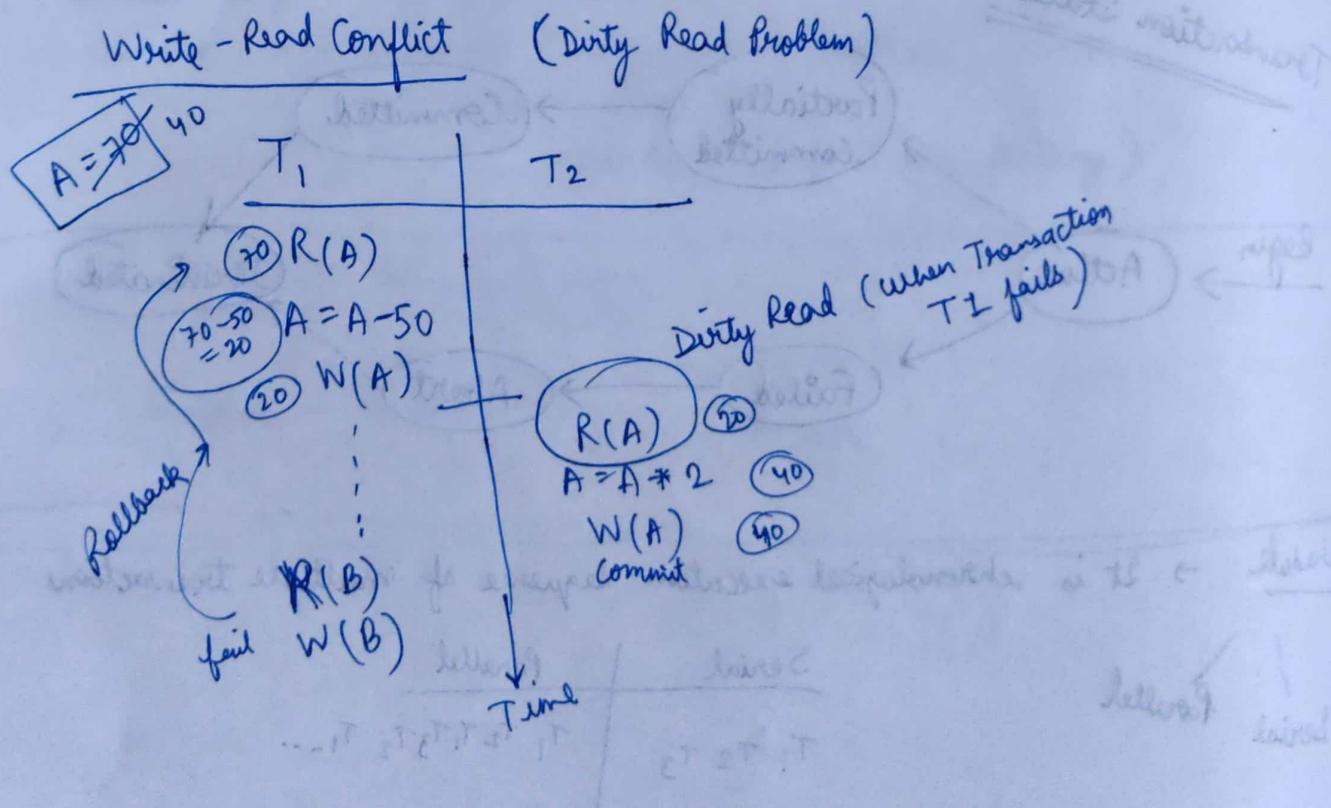


2) Incorrect Summary

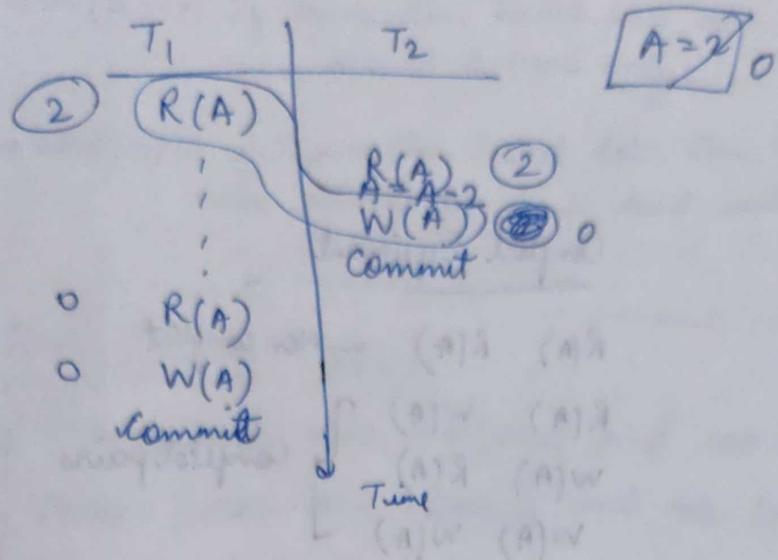
3) Lost update

4) Unrepeatable read

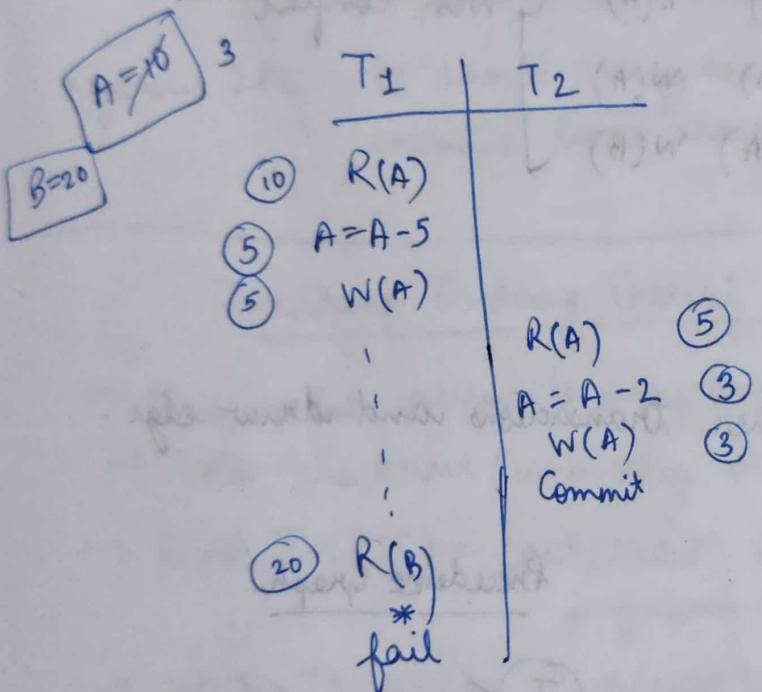
5) Phantom read



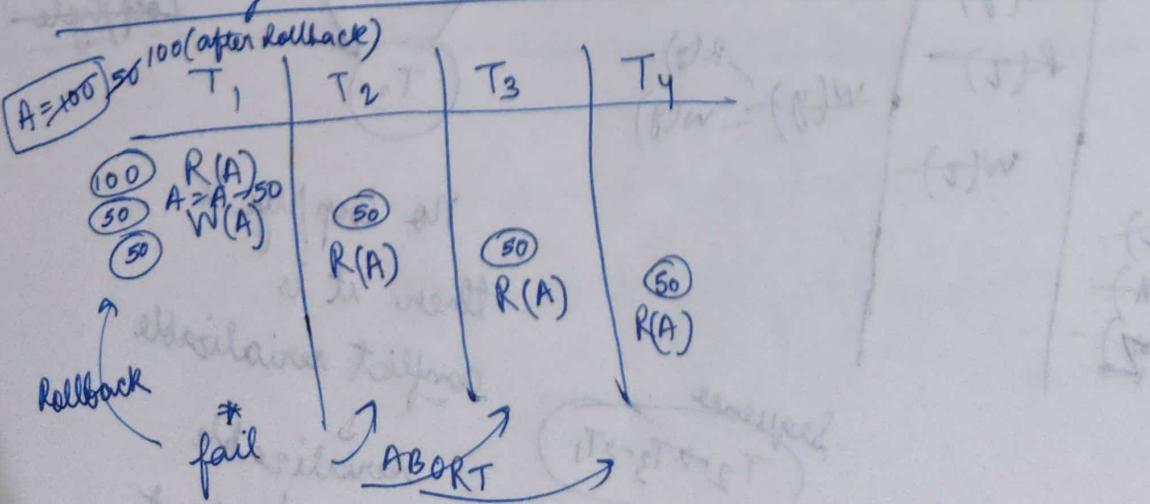
Read - Write Conflict or unrepeatable read



Irrecoverable Schedule



Cascading Schedule vs Cascadeless Schedule



Serializability

- To make serial transactions

T ₁	T ₂
R(A)	
W(A)	R(A)

Conflict
View

Conflict Equivalent

R(A) R(A) - Non conflict

R(A) W(A)
W(A) R(A)
W(A) W(A)

} Conflict pairs

R(B) R(A)
W(B) R(A)
R(B) W(A)
W(A) W(B)

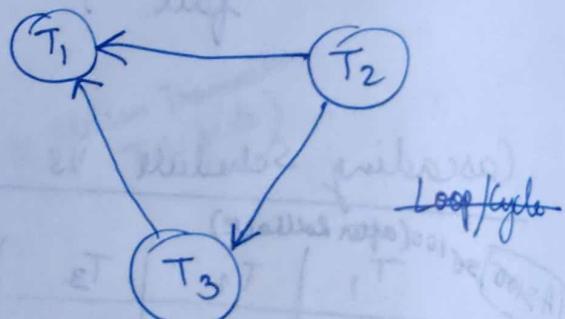
} Non conflict

Conflict serializability

- Check conflict pairs in other transactions and draw edge.

T ₁	T ₂	T ₃
R(x)		
	R(y)	
	R(z)	
		R(y)
		R(z)
		W(y)
		W(z)
		R(y)
		W(y)

Precedence Graph



Sequence
 $T_2 \rightarrow T_3 \rightarrow T_1$

No loop/cycle
then it is
conflict serializable

↓
serializable
↓
consistent

Shared - Exclusive Locking

- Shared Lock (S) \Rightarrow if transaction locked data item in shared mode then allowed to read only.
- Exclusive Lock (X) \Rightarrow if transaction locked data item in exclusive mode then allowed to Read and write both.

2 - Phase Locking (2PL)

- Growing Phase: locks are acquired and no locks are released
- Shrinking Phase: locks are released and no locks are acquired

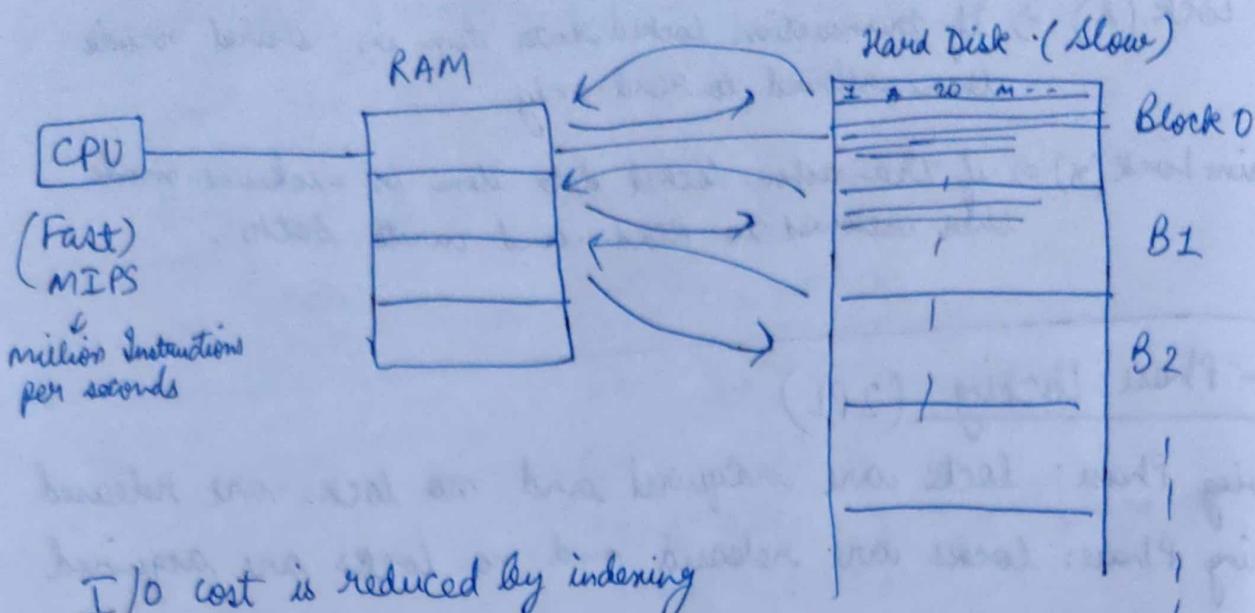
Strict 2PL: It should ~~not~~ satisfy the basic 2PL and all exclusive locks should hold until commit / abort

Rigorous 2PL: It should satisfy the basic 2PL and all shared, exclusive locks should hold until commit / abort.

TimeStamp Ordering Protocol

- Unique value assign to every transaction
- Tells the order (when they enter into system)
- $\text{Read-TS (RTS)} = \text{Last (latest) transaction no. which performed read successfully}$
- $\text{Write-TS (WTS)} = \text{Last (latest) transaction no. which performed write successfully}$

Why Indexing is used?



Types of Indexes

1. Primary
2. Clustered
3. Secondary

Ordered file	Primary Index	Clustered Index
Unordered file	Secondary Index	Secondary Index
	Key	Non-Key

View in Database

- Virtual Table
- View is the result set of a stored query
- Read-only vs. updatable views
- Materialized views

RAID → Redundant Array of Independent Disks

Q. Differentiate ~~the~~ DBMS vs RDBMS

- DBMS is a software that is used to define, create and maintain a database that provides controlled access to the data.
- RDBMS is used to store or manage only the data that are in the form of tables.

Q. What is primary key?

- P.K. = Unique + Not Null

Q. What is foreign key?

- It maintains referential integrity in database.

Q. Differentiate Delete, Drop & Truncate

Delete	Drop	Truncate
Removes rows from table	Removes table from the database	Removes all rows from table
DML	DDL	DDL
Can be rolled back	Cannot be rolled back	Cannot be rolled back

Q. What are triggers?

- A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.