2nd chp

Dependency : যে depend করে তাকে ... exist করে

Column গুলো একটার উপর depend করে
যে independed যে exist না করলে depend
করা যায় না
তাকে functional dependency বলে।

function dependency is a method describe the relation between attributes.

(x) → y
ind   dep

Y is functionally depended on X.

(2) CZ এখানে x independent
so X দ্বারা ...
Y হচ্ছে x functionally depended on X

functional dependency:

- Trivial — composite key component key থেকে depend হলে
- non-trivial — না হলে non trivial.
- Partial — General dependency — ~~independent~~ primary key এর depend.
- Transitive.
- multi-valued

partial → কোনো dependency হলে, partial.

eid did ename dname

composite key

eid, did → dname, ename

eid → ename

did → dnam

(সম্পূর্ণ ভাবে আংশিক ভাবে dependent হলে)

transitive : non key এর উপর non key depend হলে

eid did ename dname

key non key

did দিয়ে eid আইডি dname

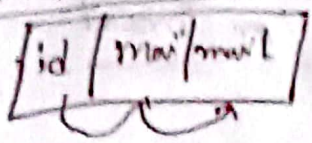
খোঁজা যাবে।

a → b
b → c
A → C

কোনো transitive হলো

A B B C

⑤ multivalued dependency :



যেখানে কোন multivalued attribute এর primary key তার accur হয় সেখানে multivalued dependency থাকে হয়।

এই থাকার clear থাকলে Normalization করা যায়

Trivial f.D. ⇒ $X = Y$ is subset of X (sid → sid)

reflexivity ⇒ gid → sid always valid. $A \to B$

Augmentation ⇒ if $A \to B$ then $AC \to BC$

Transitivity ⇒ If $A \to$ and $B \to C$ then AC

Projectivity ⇒ $A \to BC$, then $A \to B$, $A \to C$

union ⇒ If $A \to B$ and $A \to C$ then $A \to BC$

Pseudo transitive ⇒ If $A \to B$, $DB \to C$ then $DA \to C$

(Inference rules of functional dependency)

Explain the test. Integrity

To preserve referential integrity during an insert operation.

① Check for foreign key constants.

② Check for unique constants.

③ Check for cascading updates and deletes.

④ Test the insert operation.

Delete

① Check for existing foreign key relationships.

② Check for cascading deletes.

③ Check for constraints

④ Check for triggers.

⑤ Test the delete operation.

Update

- Check for existing foreign key relationships
- Check for unique constraints.
- " " cascading updates.
- " " " deletes.
- Test the update operation.

ensure database integrity.

- Entity integrity : It is concerned with the concept of primary key. has to be unique and not null.
- Referential integrity : this is concerned with the concept of foreign keys. The first rule that the foreign keys value would be refer to a primary of another table.

Domain integrity: The concept ensure that all data in a ~~defined~~ database can be traced and connected to other data. This ensure everything searchable

inconsistance এড়াতে থেকে save guard করার জন্য যে constant use করে ঐ integrity তাকে (constraint)

organizing data in data called → Normalization

Decomposed - ভেঙ্গে ফেলা

inconsistency, anomally দূর করার Normalization use হয়

insert update delete

null থাকলে ... update ... delete anomally

এগুলো যাতে না ঘটে তার জন্য যা করা হয়, তাই Normalization.

6 types Normalization

1NF - transitive
2NF - partial
3NF - trivial
BCNF - multivalued
3.5NF

4NF - loop less decomposition
5NF

1NF না হলে 2NF হবে না

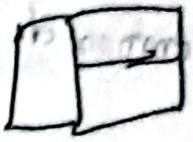
Searially করতে হবে

step wise normalized হবে

1NF

- an attribute of a relation can't hold multiple values for the 1NF.

used to remove grouping



id	phone	quantity
01	pen	5
	pencil	10
		6

decompose করে পেলাম।

01	pen	5
01	pencil	10

If a table has no repeating group then called 1NF

ID	Course
101	DBMS

→

ID	Course
101	OOP
101	DBMS

A → B
B → C

2NF

যদি সবসময় table PK দ্বারা অন্য key-কে access হয়। Composite key

AB → C

A relation schema is said to be 2NF if it is in 1NF and have no partial dependency.

1NF থাকতে হবে

eid
composite key
(pk না but pk মনে হবে)

eid	did	~~Ename~~	dname	zip	city	stm

Jodi eid → ename
did → ename
হলে dependency থাকবে না

partial dependency হবে

Req: eid → ename তাহলে partial হবে

ename কারণ eid নিজেই ename বের করে দিতে পারে।

তাই

eid	did	ename

eid	ename

composite key থাকলে (partial dependency) আলাদা table হবে।

key তখন আলাদা table হবে।

3NF ③

transitive dependency — non key → non key dependency

id → zip
zip → city

id → zip → city

অথবা id থেকে city পাবো

id	zip	city

⇒

id	zip

,

zip	city

এদের টেবিল ভাঙতে হবে

4NF ⑤

multivalued আলাদা করে

ID	mbl	mbl

⇒

id	mbl

id	mbl

এদের আলাদা করতে হবে

⑥

loss less decomposition ensure করতে 5NF

লাগে, Normalization করে তার মধ্যে কোন ...

টেবিলের value গুলো যাতে হারিয়ে না যায় ...

(৭) BCNF is called B.C.N.F

Boyce code normal form.

যদি candidate key হলেও Primary key হয় table

হবে তাকে BCNF বলে

zip	city	mobile

⇒

zip	city

mobile	city

pk-zip / pk-zip

- Primary key ছাড়া column যেন depend.
- insert, update, delete anomaly দূর করে

X → Y

super key

candidate key

primary key

6*50 (6)
2650 1000 620
3700 (22)

Indexing And Hashing

* data Search করার way

- index ← কোনো কোথায় আছে তার
- Data file ← data গুলো থাকে
- data dictionary (meta data) ← Structure সম্পর্কে তথ্য থাকে

• indexing করলে তাড়াতাড়ি খুঁজে পাওয়া যায়। index and hasing same কাজ করে।

• How to define used index or hasing

- access type ← sequential হলে index ← sequally; random → hasing
- access time — location time বের করতে কত সময় লাগে
- insertion time — data insert করতে, access and insert করতে কত সময় প্রয়োজন
- relation time — relation করতে কত time লাগে তা ইউজ করা
- Speach overhead: indexing value কম লাগে, hasing value বেশি লাগে। তাই বেশির ভাগ company hasing, ছোট company used to index

Index <u>Ordered indices</u>

~~Hashing~~ "3 type".

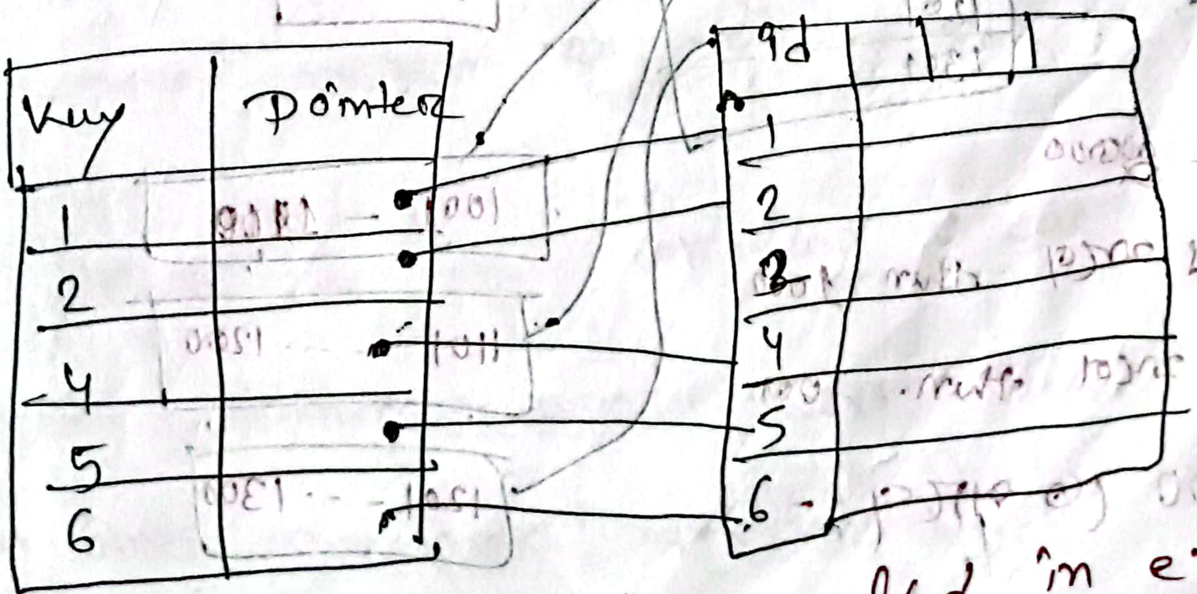
- Primary index (Order) → sorted way তে সবগুলো data sort করা primary key অনুযায়ী (1,2,3,4,5)
- Secondary index (unordered) → candidate key বা non key, ও এর mixed ও অনুযায়ী করলে (7,6,2,5)
- Cluster index (ordered) → শুধু মাত্র non-key তিনটা করলে cluster index. (1,1,1,2,2,2,3,3,3)

Dense | Sparse | multilevel.

Dense

→ সবগুলো key আছে সব instance দিয়ে index file create করে

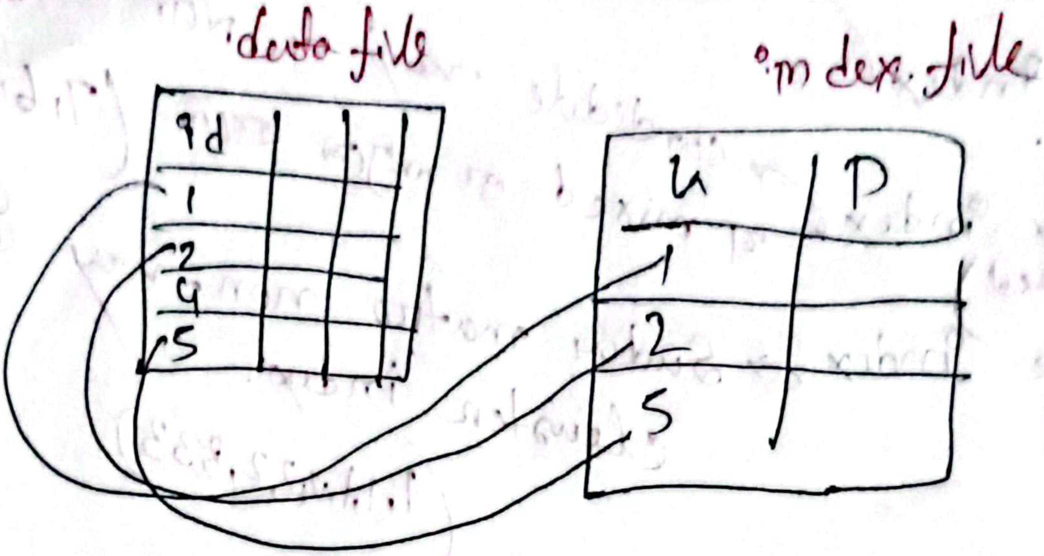
index file | data file.



An index Entry is created in every Search key value

নোট

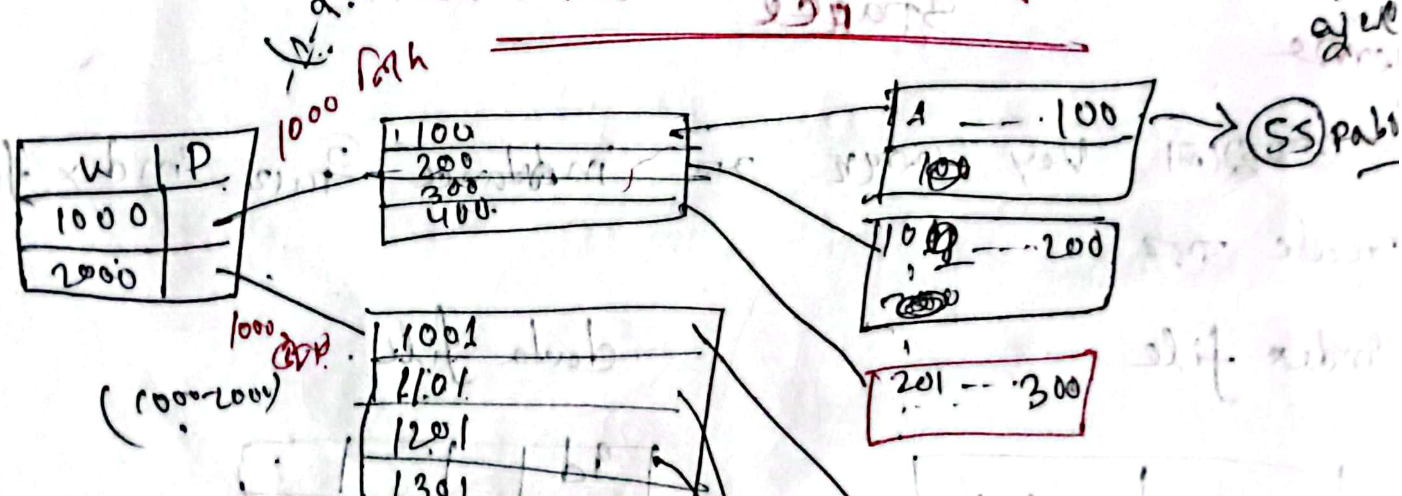
Sparse: যেকোনো কোনো শুধু একবার Line 1, 3 — আবিরেও তা randomly নিতে পারে

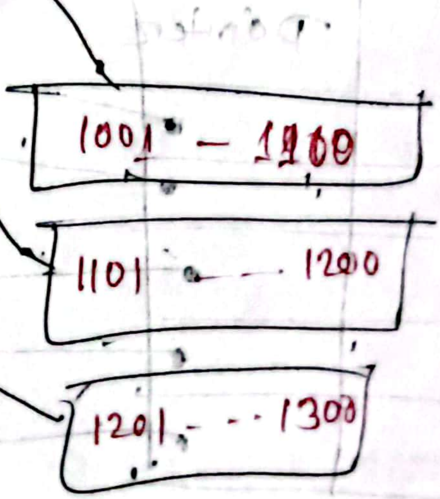


An index entry is created only on some record not all.

Ordering maintain করা হয়

Multilevel indexing





Live 1001 খুঁজবো

2000 মধ্যে আছে then 1000

এর box আছে then 1001

থেকে 1100 তে আছে।

Hash file organazation: Buket a data store. hash অনুযায়ী data search করে location অনুযায়ী যায়।

Index serially search হয়। hash randomly যাওয়া যায় So, data easily search হয়। ← advantage why hash used.

Disadvantage of hash function:

Worst hash function. এটা হচ্ছে যদি আমাদের অনেক ভুল হয় buket overflow হয় কারণ Sababir তাকে address generate না করলে।

· why buket overflow occur?

If the buket doesn't have enough space. a buket overflow occur.

①. Insufficient buket থাকলে

②. Mapping problem ② Skew. Scew

· যে buket এ জায়গা নাই তাও থাকলে। Scwee বলে এটাকে

worst hash function থাকলে buket overflow হয় এটা অনেক সময় হয়।

· what will do when buket overflow occur?

· overflow channing: আমাদের একটা buket থাকে যেখানে এক্সট্রা বুকে রেখে আমরা যাই।

Open hashing. Linear probing : Similarly ইনপুট দেওয়া হবে
এবং আমাদের এখন based হিসেবে linear probing হবে

Ideal hash function: Same numbers ও distributed
hon করবে

Distribution 2 type.

uniform : Sundar ভাবে ছড়ানো থাকে
random - " " " না।

Multilevel indexing: It used in database
to improve the efficiency of accessing and ret-
-rieving data from Large databases.

Multilevel indexing helps in breaking down the
index into several smaller indices in order
to make the outermost level so small that
it can be saved in a single disk block
which can easily access anywhere in the main
memory.

Skew in hashing

- A bucket may overflow even when other buckets still have space. This is called bucket skew.

Causes of skew:

(1) multiple records may have the same search key.

(2) the chosen hash function may result in non uniform distribution of search keys.

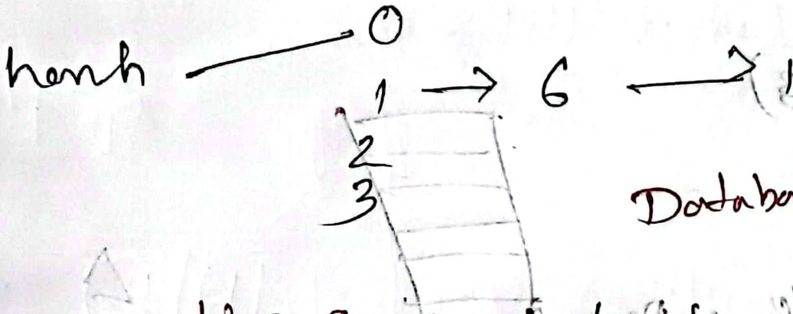
Resolve skew problem:

(1) Implement data partitioning to distribute data across partitions.

(2) Use proper indexing techniques to optimize query performance and balance workload.

(3) Implement dynamic load balancing algorithms to distribute workload across nodes/partitions.

Overflow chaining: when buckets are full, a new bucket are allocated for the same hash result and is linked after the previous one. this mechanism called (Closed hashing)

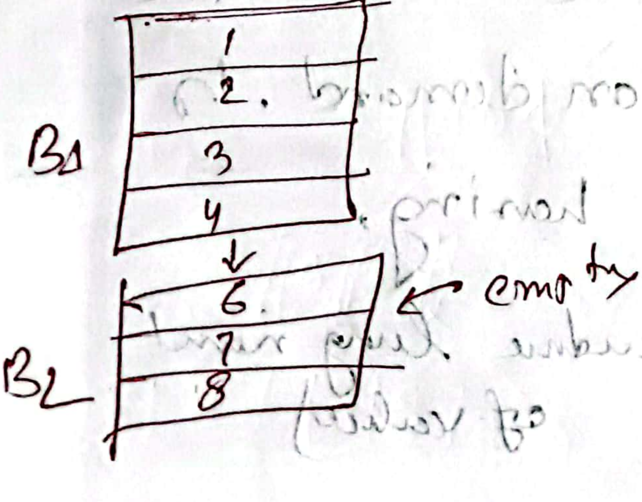


(অনেক bucket fixed থাকে না, add করা হয়)

Database closed hashing use করে

Linear probing: Size fixed থাকে, যদি bucket full হয় (অথবা থাকে or empty item then porer tai add করো. If the bukets are full the system insert it in some other bucket if it is empty.

(Open hashing)



Hash. Index

a hash index organizes the search key with their associated pointers into a hash file structure.

Search key (2, 5, 3, 8)

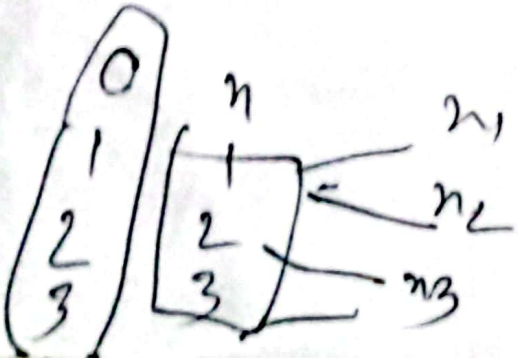
hash table

(Hash function / x mod 10, n mod 10)

Dynamic hasing

Dynamic hasing provides a mechanism in which data buckets are added and removed dynamically and on demand. It also known as extended hasing.

[made to procedure long run of values]



B+ trees

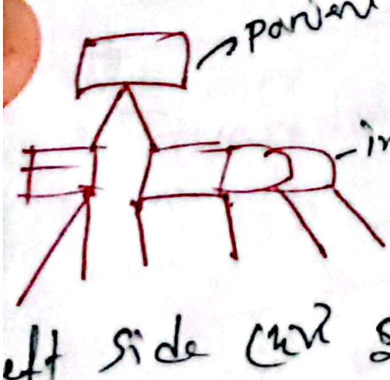
• a symbolic tree to represent the working procedure of database.

- B+tree
- ISAM (indexed seq access method)

fill factor >> extra node হলেও 50% data থাকবে

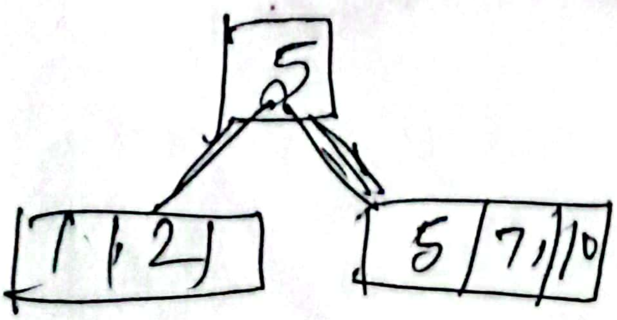
Ex

rules: serially নিলাম n = 3 হলেও (4) টি হলে middle টাকে parent করে দিবো, then, n = 4 হলে middle টি নিয়ে প্রথম node full হবে [1|2|(3)|4|5] ← internal node এর parent আর হবে না।



internal (এর মান parent এর same টি internal থাকে না but node থাকে

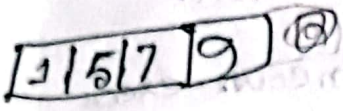
left side ছোট হবে, Right side বড় অথবা সমান হবে

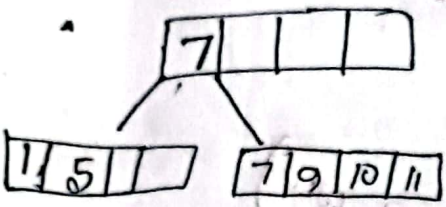


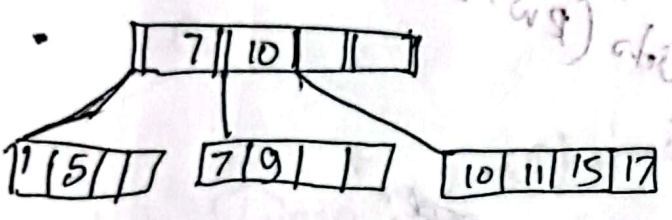
Ex order tree

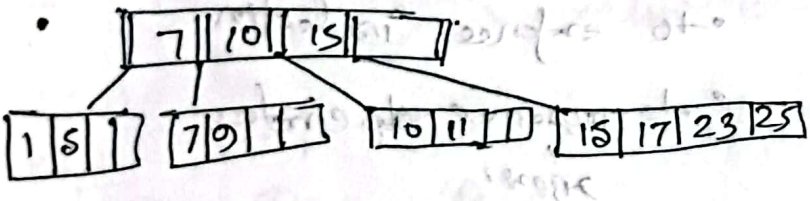
Ex

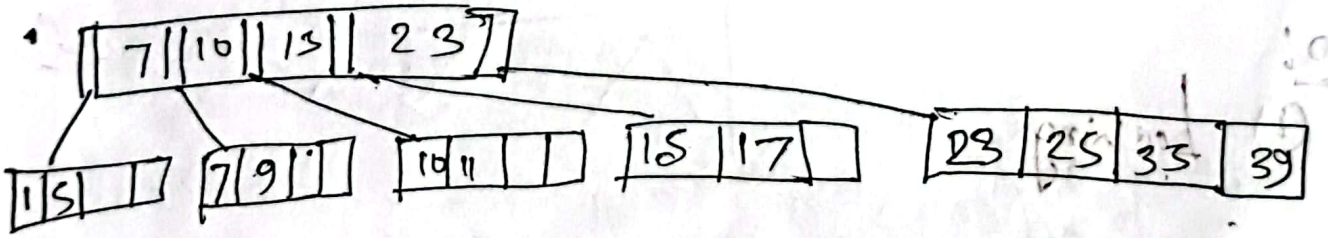
- 1, 5, 7, 9, 10, 11, 15, 17, 23, 25, 35, 39, 40. $n = 4$

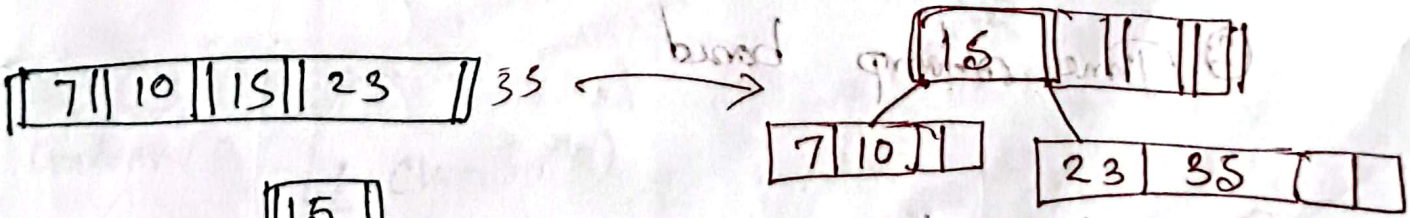
- 

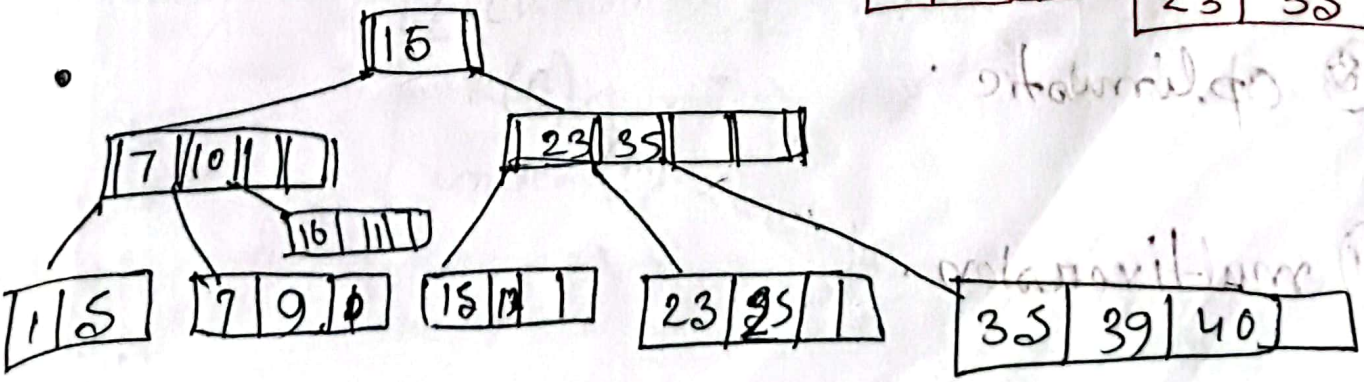
- 

- 

- 

- 

- 

- 

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

Acid properties:

atomicity
consistency.
isolation.
Durability.

Atomicity: either all or none. until commit না process আগে এর rollback হবে, 1st থেকে st: হবে না transection field হবে.

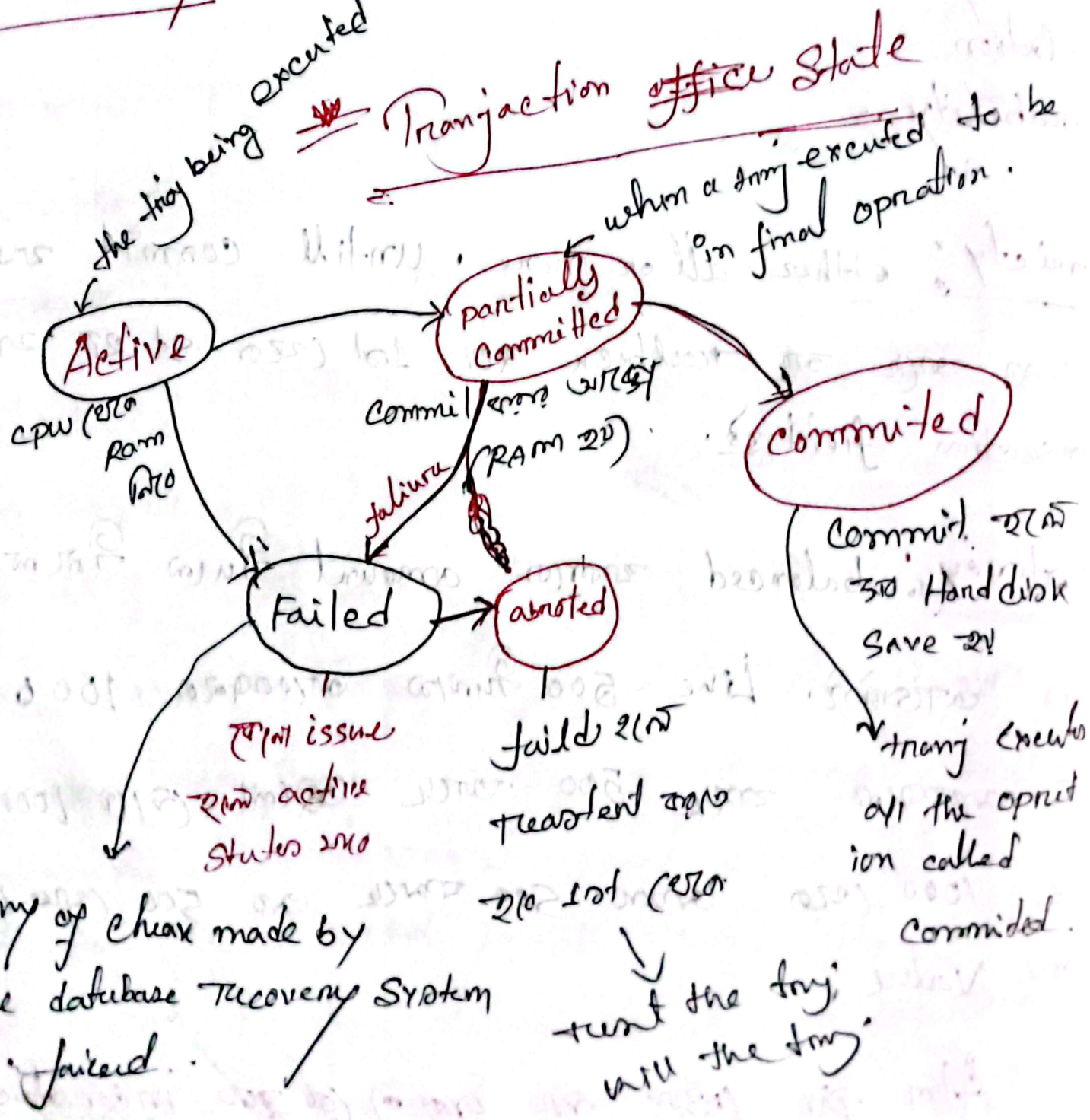
Consistency: balanced থাকবে. amount দিলে দিলাম Line 500 দিলাম অ্যাকাউন্টে 1000 ছিল অ্যাকাউন্টে দিলাম অ্যাকাউন্টের থাকে 500 আছে যোগ করে 1000 হবে 1000 থেকে অন্যের 1500 থাকে, এবং 500 থেকে Same value;

Atm pin দেওয়ার পর টাকা না দিলে হবে inconsistency

Isolation: parallal schidule (A ⇄ B) কে serial convert করে (until 1st is complete 2nd can be used)

Durability: যে change করা হয়েছে permanently saved থাকে

Tranjaction ~~office~~ State



Shadow copy technique

It is a technique which is based on making copies in database.

→ A pointer called db pointer ... database ... Current constant copy ... Point ... db ... update ... shadow ...

—x—

① Let us assume only one trxj is active at a time.

② A pointer called dbpointer always points to the current constant copy of the database.

③ All updates are made on a shadow copy of the database and the db pointer is made to point to the updated shadow copy only after the trxj reaches partial commit and all updated pages have been flushed to disk.

④ In case trxj failed, old constant copy pointed to by db pointer can be used and the shadow copy can be deleted.

These procedures ensure Atomicity, Concurrency, Durability but not Isolation Property.

Concurrent Execution

helps in reducing waiting times, improved throughput and resource utilization.

Problem:

(i) Lost update problem (w-w conflict)

(ii) Tempory update or dirty read. (w-r)

→ Occurs when one transaction updates a database item and the transaction failed but the update is read by some other transaction.

T_1	T_2
R(A)	
A=A+2	
W(A)	R(A)
	R
R(B)	W(A)
	commit.

In a multi-user system, multiple users can access and use the same database at one time which is known as concurrent execution.

Serializability

Shidule কে কখন serializable বলে কিভাবে?

two types : ~~view~~ View
conflict

S

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

Serial shidule,

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

parallel

conflict equivalent

non conflict
Standard conflict

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

non conflict

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

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

non conflict
So swap করা যাবে

⇒

T_1	T_2
R(A)	
W(A)	
RB	RA
	W(A)

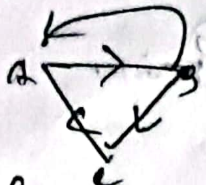
Same করা যাবে

⇒

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

conflict equivalent Shidule.

View serializablity

যদি  loop থাকে তখন এটি Non conflict Serializable. যদি Loop না থাকে the Conflict and Serializable. যেকোন loop থাকলে তখন check করতে। ওটি View Serializability used হয়।

T1	T2	T3
R(A)	A=A-40	
A=A-40	W(A)	A=A-20
W(A)		W(A)

=>

T1	T2	T3
R(A)		
W(A)	W(A)	W(A)

দুই ট্রানজেকশনের মধ্যে আছে Loop থাকলে conflict answer দিবেন View Serializablity

যেমন:

Recoverable Schedule :

Serializable হলেও সব failure হয় তখন তাকে Recoverable schedule. (roll back করে fail হলে Atomicity রক্ষা)

Dirty read

No → Recoverable

Yes → dirty read হয় (কোনো committed হলে Recoverable

uncommitted read করলে

T_1	T_2
R(A)	
A = A + 10	
W(A) →	R(A) ←
	A = A - 5
	W A
Failure	e.

T_2 commit করে

Cascading

dirty read

also recoverable.

< Cascadeless Shidule

each transaction

- ... table ... rollback ...

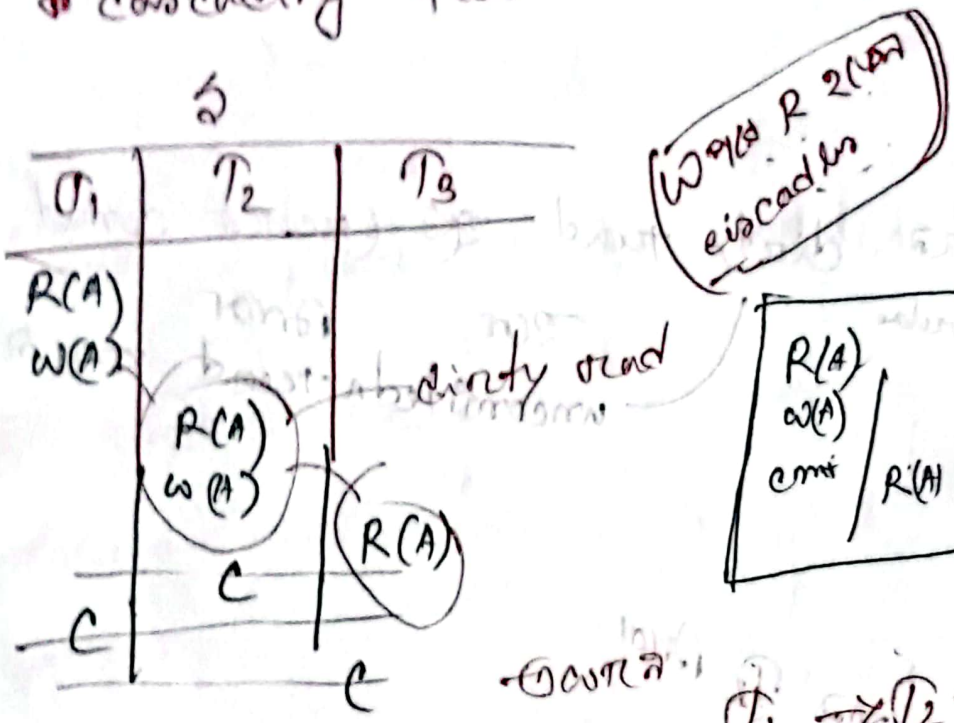
T_1 rollback ... T_2 ... cascadeless Shidule.

- cascading rollback ... cascadeless shidule.

recoverable

cascadeless ...

cascadeless ...



Write R ... cascading

dirty read

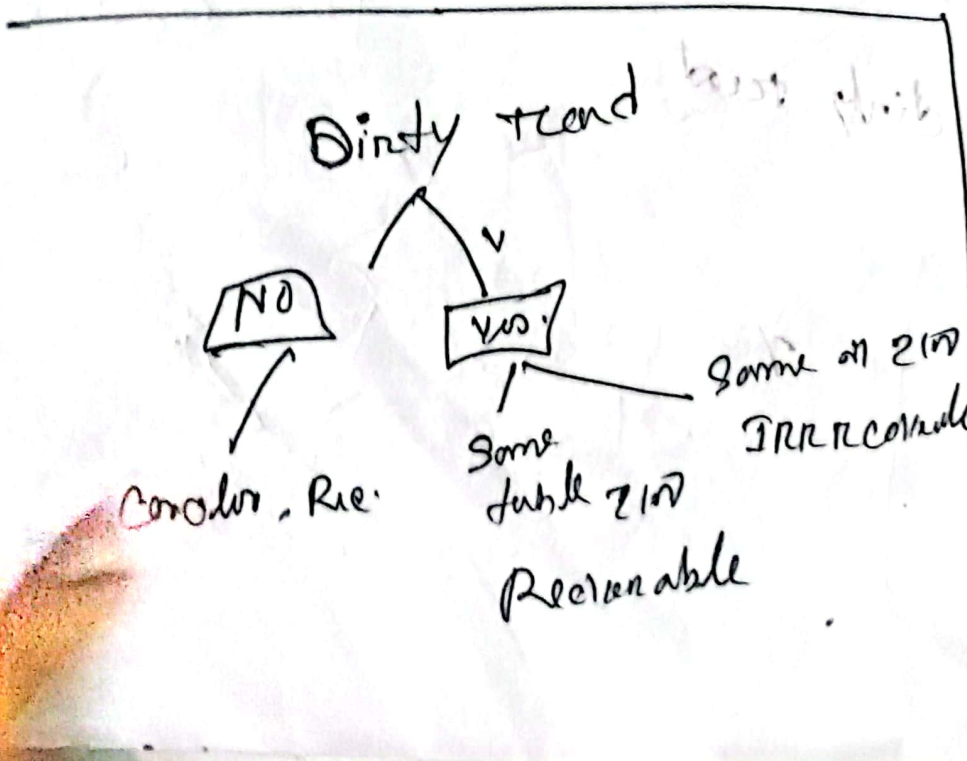
cascadeless ...

Serially

$T_1 \rightarrow T_2 \rightarrow T_3$

commit ...

recoverable



Dirty read or write-read Conflict

	T_1	T_2
A=70	R(A)	
	A=50-A =20	
	W(A) 20	
		R(A) 20
		A=A*2 = 40
		commit
fail	R(B)	
	W(B)	

fail হলে commit হয়ে গেছে rollback হবে না

So this data আগে commit হয়নি তাই So that এটা dirty read

R-R হলে problem হয় না

Read - write conflict

T_1	T_2
R(A)	
	R(A)
	W(A) A=A-2)
W(A)	commit

যদি T_1 R করার পর parallel W করে আবার W করে Tolle, যদি commit T_2 করে T_1 W হয় না এবং aborted হয় Read hoe হবে.

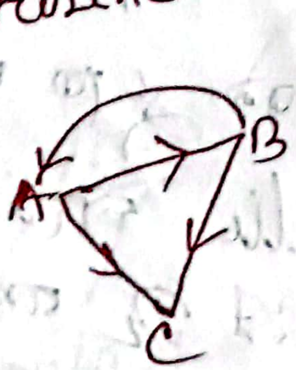
The recovery manager is responsible for ensuring Atomicity and Duribility.

⇒ atomicity is guaranteed by undoing the actions of the transactions that did not commit. (abroted)

⇒ Duribility is guaranted by making sure that all actions of commited transaction survive crashes and failurs.
(Commited gula failure যেন টিকে থাকে)

Testing of serilizability.

• Precendence grap এর সাহায্যে test হয়



cycle হলে not serial | cycle না হলে serial.

Chapter 8

Concurrency control techqus.

controling concerad. execution.

- C-I is a proces of managing simultaneous execution tranjaction of shared db.

purpose of concurency control

- To enforce isolation (ensure)
- To preserve db consistency
- To resolve conflicts (RW, WR, WW)

- to enforce isolation
- to preserve db consistency

Techniques:

1. Locking
2. Time stamp based
3. Optimistic

Lock-Based protocol.

A lock guarantees exclusive use of data items to a current transaction.

who does it's work → acquire data items (lock acquire)
→ after completion of transaction (release lock)

Lock Types:
- Shared = (lock-S) → read only data items values.
- Exclusive = (lock-X) → used for both read and write data items.

Ex

T_1	T_2
Lock-X (B)	
R(B)	
B-50	
W(B)	
unlock(B)	
	Lock-S (B)
	R(B)
	unlock (B)

unlock → shared type

any number of transaction hold

only hold 1 transaction at a

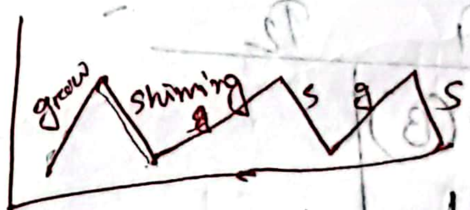
Granting of Locks ← data synchronization

when a transaction request a lock on a data item in a particular mode no other transaction has a lock on the same data item in a conflicting mode, the lock can be granted only once the lock requested is granted.

[trj proceed]

Two phase locking protocols

Shared / exclusive locked update version.



Two types.

Growing Phase: Locks are acquired and no locks are released (প্রথমে শুধুই Lock করবে 1st phase)

Shrinking phase: Locks are released and no locks (unlock করার সুযোগ / Lock করা যাবে না ... এই phase)

Advantage: serializability achevied (ensured by 2pl)

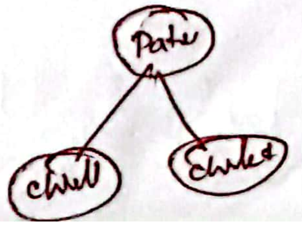
disadvantage: • May not free from irrecovera bility.

• Not free from deadlock. • Not free from cascading rodlback

• Not free from starvation.

graph based protocol Tree protocal in DBMS.

- only exclusive lock are allowed.
- A data item Q can be Locked by transaction only if the parent of Q is currently Locked by transaction (item Lock করতে parent lock করতে)
- Data item can be unlocked at any time.
- A data item cannot subsequently be relocked by transaction.
- The tree protocol ensures conflict serializability as well as freedom from deadlock.
- unlocking may occur earlier in tree locking protocal than in two phase locking protocal.



প্রথম থেকে ধরে child access করব

Time stamp odering protocol.

- Unique value assign to every tranjaction.
- Telles the order (when they enters into system

- Read_TS (RTS) (Last (latest) tranjaction no which performed read sucessfully.
last কোন সময়ে এক্সেস করেছে

- write_TS (WTS) (last (latest traj no which performed write sucessfully)

10	20	30
T_1	T_2	T_3

W A)

W(A) last কোন সময়ে wts (20)

W(A) — same

√ " " "

RTS —

Rules TS time ↑ younger

- je tranjection ~~আগে~~ asbe piority বেশি পাবে

100 T_1	200 T_2
R(A)	
	W(A)

100 T_1	200 T_2
R(A)	
	R(A)

100 T_1	200 T_2
W(A)	
	W(A)

allow

100 T_1	200 T_2
	R(A)
W(A)	

rollback

100 T_1	200 T_2
	W(A)
R(A)	

rollback

100 T_1	200 T_2
	W(A)
W(A)	

rollback

conflict serializability follow করে।

এগুলো allow না করলে হলে roll back করে দিবে, তখন younger trans priority পাবে, rollback না করলে T_2 গুলো failure হবে

Dead lock

- If a two or more process are waiting one happening of some event, which called deadlock.

হবে কোনো then করব ← Dead lock detection & recobery
ফিরাব

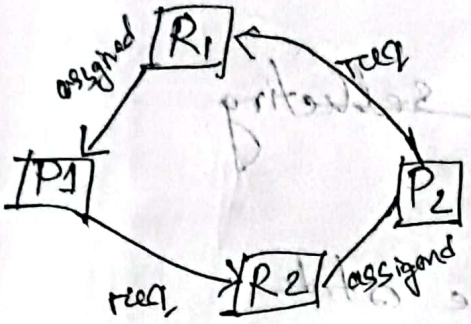
Dead lock detection.

- wait for graph
  single instance
  (cycle হবে না)
- Bankers algorithm
  (multiple instance)

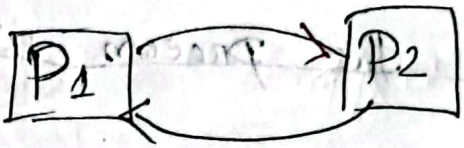
prevention : না হওয়ার আগেই হয় না বাধা

আগে আগেই থেকে

Ex for wait for graph. => cycle হলে deadlock হবে (A → B → A)



=>



cycle থাকলে deadlock

Deadlock recovery

~~Process termination~~

Resource preemption:

Selecting a victim: which resources and process are to be ~~preempted~~ displace. (we should rollback those transaction ~~to rollback~~ that will be incur the minimum cost), which were selecting as a victim which will be incur the minimum cost

[যে transaction এর problem হবে তারা সবাই victim হবে), basic time logic

Roolback : → Partial → complete
Rollback the process which was selecting as a victim

- ~~Rollback the process which was sellecting as victim.~~
- Rollback the process to some safe state
- Rollback the process and restard it.
- Abrot the process

Starvation : It may happen that the same process is always picked as a victim. As a result the process never completes. It is task we must ensure that a process can be picked as a victim only finite small numbers of time.

[যে victim তা বার বার যেন picked না হয়] তাহলে সেটা হবে Starvation হবে.

Recovery System

Types of failure

- 3 types

(1) Transaction failure : Due to some reasons transaction failure.

Logical errors : BAD Inputs, Data not found, overflow, resource limit exceeded.

System errors : DeADLock.

(2) system crash : There was hardware malfunction or a bug in the database software or the operating system.

(3) Disk failure : A disk block Losses Its content. During data transfer operation, so copies of data are Backup or Sorted.

Classification of Storage.

- Volatile Storage.
- nonvolatile storage.
- stable storage. (plays vital roles in recovery algorithms)

Check points

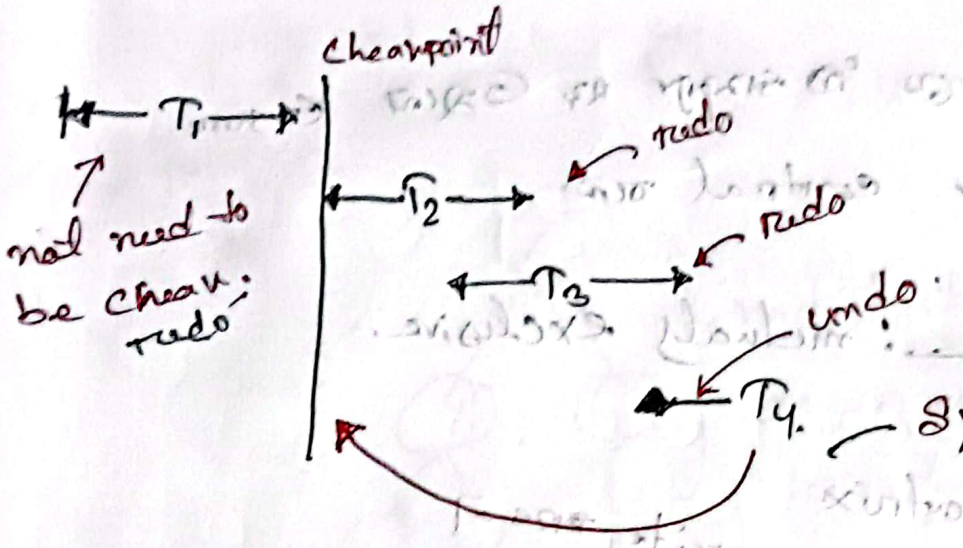
. It is a meachanism where all the previous logs are removed from the system and stored permanently in storage disk.

there the two major difficulties with this approach.

- Extra time in processing entire log file
- redoing of unnecessary transaction.

redo mean completed trn
undo " uncompleted trn

To Solve problem by Checkpoint :



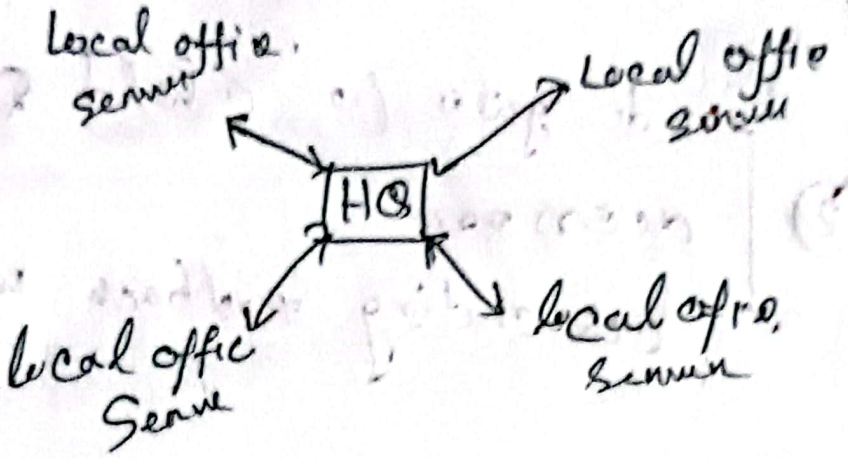
-এখানে T_1 এর checkpoint হয়েছে so এটার check করার দায়িত্ব নাই। rollback করলে checkpoint থেকে-যেগুলা T_2 T_3 redo হবে, যে গুলো completed uncompleted গুলা undo, এক্ষেত্রে time-এর T_4 বন্ধ দায়িত্বে, তবে checkpoints এর আগের গুলো বন্ধ দায়িত্বে। checkpoint দায়িত্ব নাই!

Distributed database

Distributed.

~~Centralized~~ database.

centralized



Server computer.

fragmentation: যে server গুলা local server থেকে data transfer করবে (fragment করে data করবে)

Replication: মূল database কে copy করে রাখা

replicate করা, তাই replication

copy করা

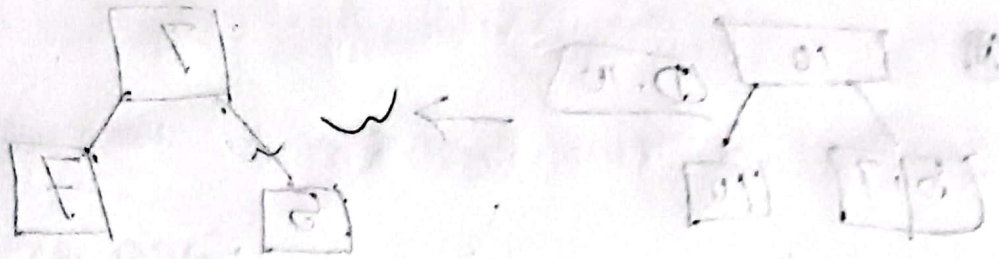
heterogenious

homogenious

• ভিন্ন ভিন্ন হবে environment	• একদম same হবে system

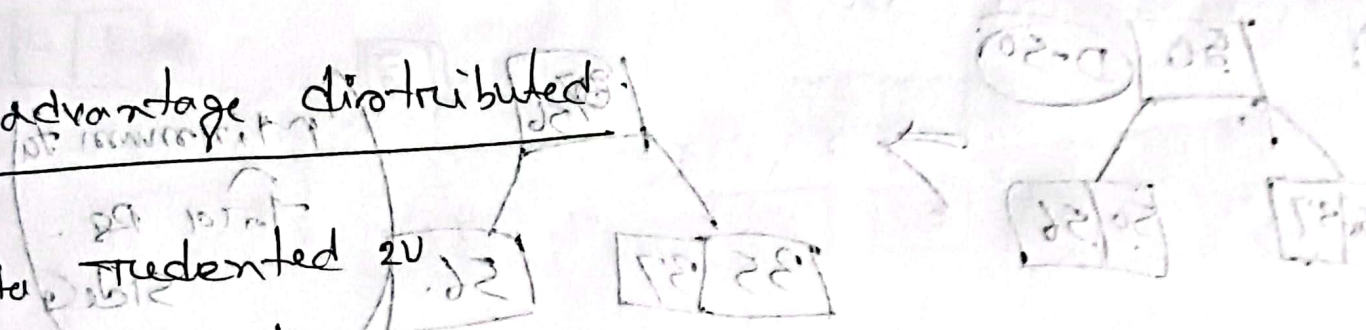
multi database, architecture

- ~~Centralized~~ and distributed database. advantage
- load server work, local autonomy
- Reliability
- Scalibility
- heterogenity.
- time save



Disadvantage distributed:

- Data redented
- Senconization.



features:

(i) used to create delete, retrive.

(ii) universally updated

(iii) hoarge volumn data processed.

(iv) heterogenous data base platform

Lock Compatibility matrix

	S	X
S	T	F
X	F	F

* 2PL ensured Serializability by applying to the trnj data which blocks other trnj to access the same data at the same time.

[... data access ...]

It helps elimate concurrency problem in DBMS but it doesnt stop deadlocks.

Strict 2PL: All exclusive locks held by the trnj be released until the trnj commits.

Rigrous 2PL: All Shared and exclusive held by the trnj be released until the trnj ends