

3S SIR

'B' Day

Cycle - 12

Day

✓

Time : 8:50 Date : 11/9/19

Sub :

Concurrent Schedule: to fulfill protocols
→ Inconsistency

Conflict Equivalent Schedule

Serial Schedule

Conflict Serializable Schedule

Precedence Graph:

Directed Graph



when →

(1) T_1 executes read(D) before T_2 executes

or, (2) T_1 executes write(D) before

or, (3) T_1 executes read(D) before T_2 executes write(D)

or, (3) T_1 executes write(D) before T_2 executes write(D)

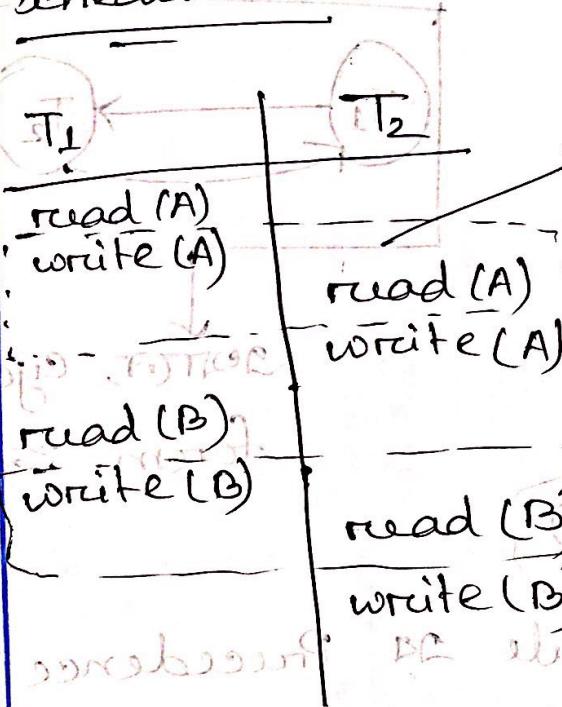
BEGINNING OF PAGE

Sub:

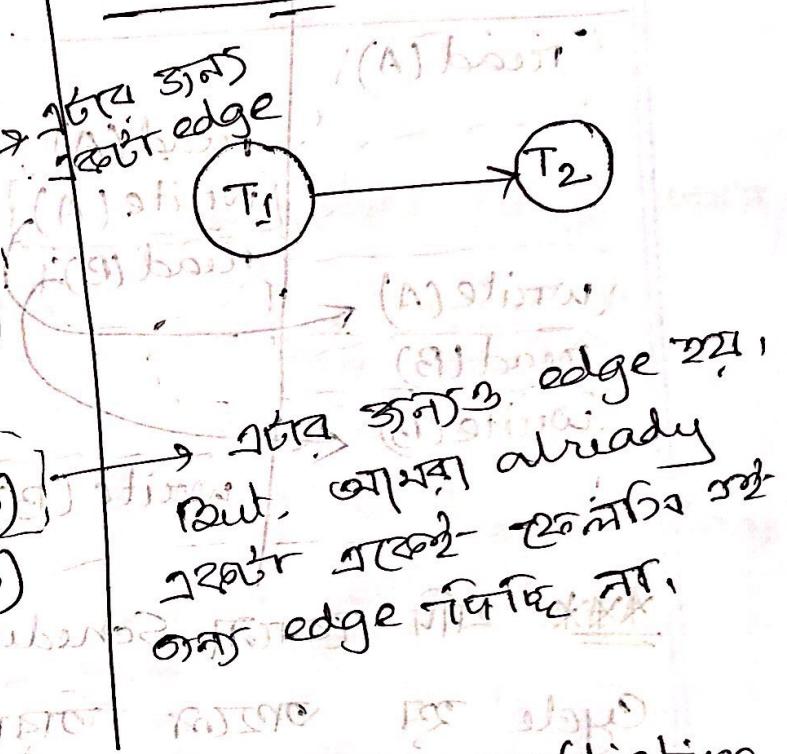
Day: _____
 Time: _____ Date: / /

T₁ & T₂ contain conflicting instructions.

Schedule - 3



Precedence Graph:

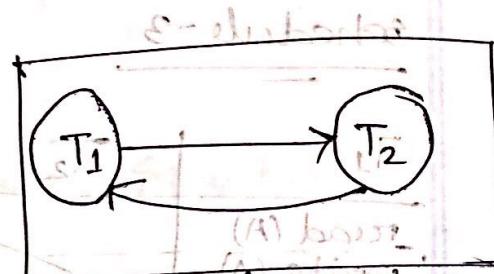
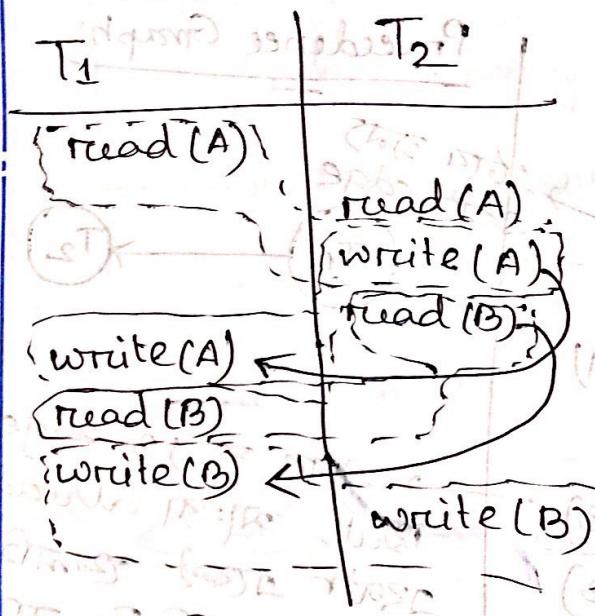


* edge දීව තෙවනු, යුතු වාක්‍ය සඳහා conflicting instructions.

Sub:

Day						
Time 1						

Schedule - 4: Inconsistency analysis ST 2. It.



গুরুত্বপূর্ণ কানোনো অবস্থা।
কানোনো অবস্থা করে যাবে।
কানোনো অবস্থা করে যাবে।

* * * যদি কোনো Schedule এর Precedence Graph

Cycle হ্যু অবস্থা তারু Conflict Serializable
না, তারু মান কৈ Schedule এর inconsistency
আছে। (So, Schedule - 3 → Conflict
Serializable)

Schedule - 4 → Conflict
Serializable না)

New Chapter

"Concurrency Control"

Lock:

(1) S - Lock

কোনো Data item প্রয়োগ করতে ক্ষমতা নেওয়া

ব্যাপ্তি কর্তৃত স্বামী ক্ষমতা প্রদান না।
 অন্য কর্তৃত স্বামী ক্ষমতা প্রদান না।
 (Read-only mode মানে ইত্যাদিমান read)
 modes

(1) Shared Lock → প্রযোজন করা যাবার

ক্ষমতা = S (D)

→ Data item

→ (2) Exclusive Lock

অক্ষমতা করা যাবার

Lock - X (D)

→ Data Item

④ Shared-Lock → Data Item can only be
 read, can't be written. (No Update)

④ Exclusive-Lock → Data Item can both be
 read & written.

Sub:

'D' Day
cycle - 12

Day
Time: 9:40 Date: 15/9/19

Lock:

Read-only mode

Shared mode

Lock-S(D)

Data item

Exclusive mode (Read & Write)

Lock-X(D)

① Shared mode \Rightarrow Transaction Data item

just read করতে পারে,

② Exclusive mode \Rightarrow read, write করতে পারে,

Several Transactions \gg Lock-S(D)
can hold

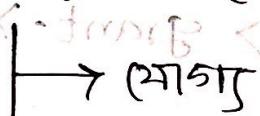
At a time 1 Transaction \gg Lock-X(D)
can hold

Mutual
Exclusion

Real Life Locking: (উদাহরণ একাউন্ট)

যোবেদন

ঘোষণা বিবেচনা



অপেক্ষমান তালিকা

Data Item (D) → T_a

(a) T_a To access D → Lock → D

Step 1: Request → (concurrency control manager)

Lock-X(D) Read & Write

Lock-S(D) : (a) lock

Read : (a+A) pending

Step 2: Lock-Compatibility Matrix

→ Step 2: 1: compatible → (can Lock)

Read > Update > write

>> Unlock

→ Step 2: Not compatible → wait

A → 500
Sub:
M → 1000

A → B

A + B >> Display

Concurrency Control Manager (CCM)

T₅

T₆

read(A);
Lock-X(A);
read(A);
A := A - 500;
write(A);
unlock(A);

Lock-S(A) → grant-S(A, T₅)

read(A); ←

Lock-S(B) → grant-S(B, T₆)

read(B); ←

display(A+B);

Lock-X(B);

→ wait

unlock(A);

unlock(B);

Lock-X(B);

→ grant-X(B, T₅)

read(B);

B := B + 500;

write(B);

unlock(B);

Lock Compatibility Matrix:

		T_j Request	
		Lock - S (D)	Lock - X (D)
T_i hold		Lock - S (D)	True
	Lock - X (D)	False	

→ Locking ↘ Unlocking ↗ Prob অত্যেক্ষণ
 ↙ Inconsistent state ↛ Lock করেও একটি পরিস্থিতি
 ↛ আপনার মতো করে নেওয়া হচ্ছে।

4500/-
 3000/-
 7500/-

কেন্দ্রীয় T_6 এ display() ,

জন্মলেখ লক করেও একটি

Inconsistent state এ চলে যাবে।

④ T_5 এ unlock অনেক আগে

7500/- এবং 2500/- পুর নথি জো দেওয়া।

প্রতিপাদন { বাবু সত্ত্বা } ← স্টেটস

BS
SIRC

Sub:

'E' Day
Cycle-12

Day
Time: 8:50 Date: 16/9/19

T₅

```
Lock-X(A);  
read(A);  
A := A - 500;  
write(A);  
unlock(A);
```

```
Lock-X(B);  
read(B);  
B := B + 500;
```

```
write(B);  
unlock(B);  
commit;
```

কার্য A updated হলেও

কার্য B updated হয়নি,

এটা অসমর্থ কারণ

আগেতোকে Unlock

করিনি।

T₆: ~~multiple parallel transaction~~

```
Lock-X(A);  
read(A);  
unlock(A);  
Lock-S(A);  
read(A);  
unlock(A);  
Lock-S(B);  
read(B);  
unlock(B);  
display(A+B);  
commit;
```

মাত্র
updated

not updated

inconsistency

কার্য A updated হলেও

কার্য B updated হয়নি,

এটা অসমর্থ কারণ

আগেতোকে Unlock

Moral of the day: "কোথা পর্যন্ত ধৈর্য ফেরব"

unlock → at the end of transaction

So, new modification see like -

T₇

```

Lock-X(A);
read(A);
A := A - 500;
write(A);
    
```

T₈

Concurrency
Control Manager

→ grant-X(A, T₇)

007-A := A

then → Lock-S(A);

read(A);

Lock-S(B);

read(B);

display(A+B);

unlock(A);

unlock(B);

Request
wait

→ grant-S(A, T₈)

Req. wait

→ grant-S(B, T₈)

4500/-
3500/-
8000/-

Lock-X(B);

read(B);

B := B + 500;

write(B);

unlock(A);

unlock(B);

then

grant privilege

Sub: DBMS

Day: / / / / /
Time: / / / / /
Date: / / /

T₁₇

Lock-X(A);
read(A);
 $A := A - 500$;
write(A);

Growing
Phase

Lock-X(B);
read(B);
 $B := B + 500$;
write(B);
unlock(A);
unlock(B);

Shrinking phase

T₁₈ nutritive library

Lock-S(B);
read(B);
lock-S(A);
read(A);
display(A+B);

Lock Point

Shrinking Phase

unlock(A);
unlock(B);

before

lock point

Concurrency
Control Log

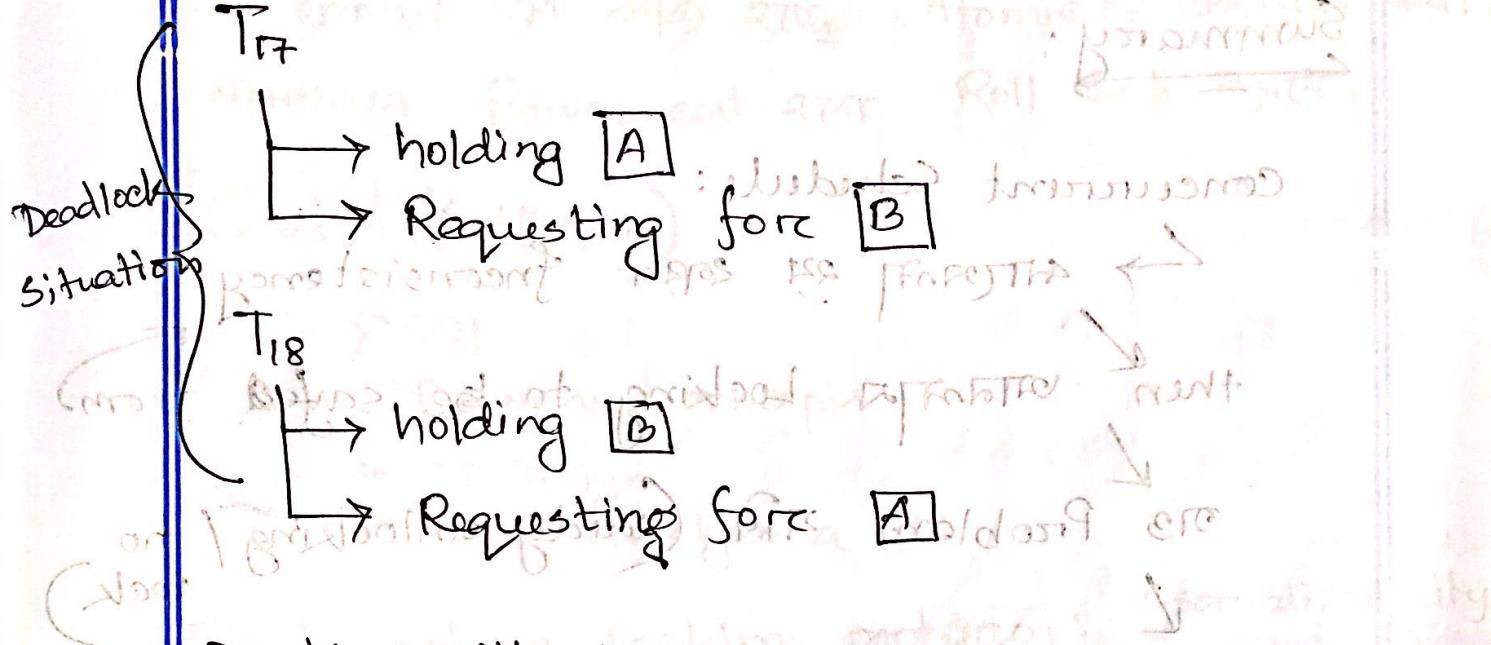
Manager

(A) X - lock

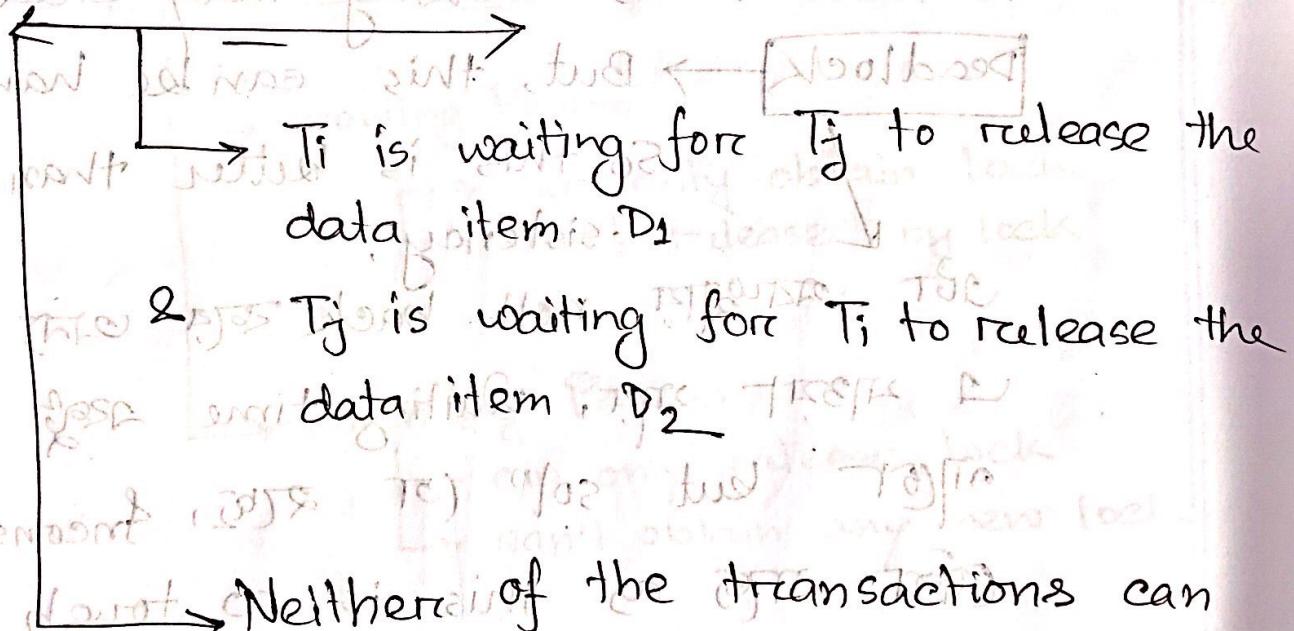
(B) lock

(A) lock

(B) lock



Dead Lock Situation:



Summary:

concurrent Schedule:

↳ **Problem** 2nd inconsistency

then **Analysis** Locking to be safe from

③ Problem **死锁** (Early unlocking / no lock)

③ Problem as locking may create

Deadlock

But, this can be handled.

So, this is better than inconsistency.

④ **死锁** 由于回滚， back out of ③ trans.

→ **死锁** 由于 waiting time 超过 cap

或 but solⁿ (or 2nd inconsistency)

原因 由于 this is so tough to

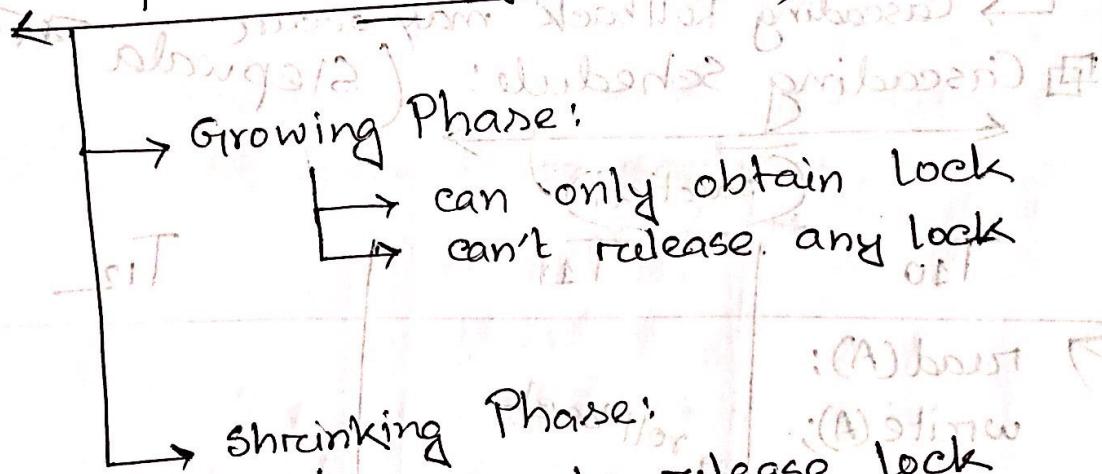
find solⁿ due to reasoning

④ Commit এবং Roll back. Atomicity ensure করে।
 মাধ্যমে Power cut হলে Roll back করবে।

QUIZ (Syllabus)

- SQL
- DB primary concepts
- DB Design
- Relational Algebra

■ Two-phase locking protocol (serializability maintained)



ক্ষেত্রটি T_1, T_2 এর schedule দ্বাৰা পৰিচয় কৰা হৈলো।

T_6, T_7 2-phase locking Protocol মনে কৰা হৈলো।

⇒ यदि कोई schedule serializable है तो उसके लिए consistency.

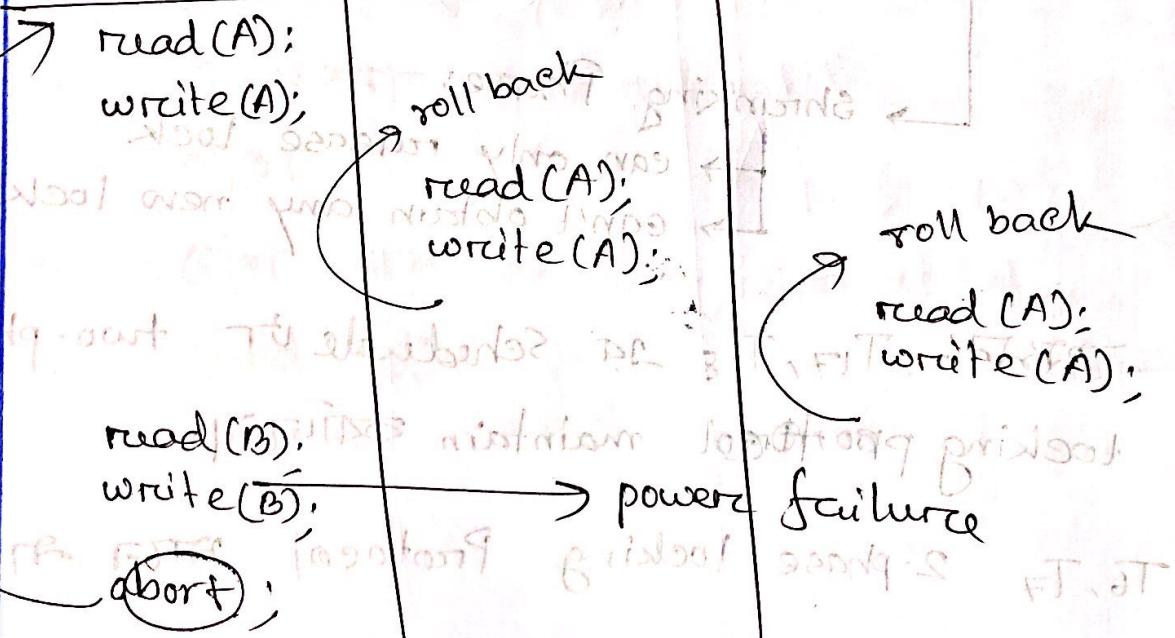
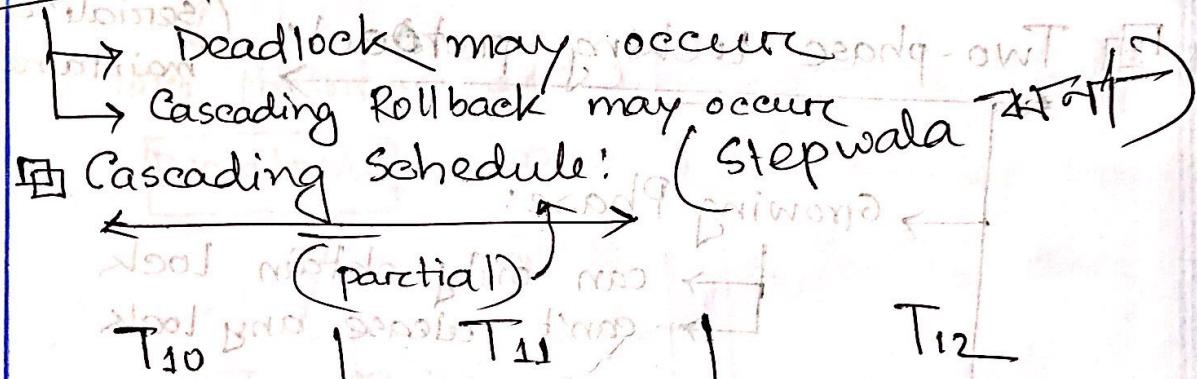
जबकि कोई schedule non-serializable है तो consistency.

Adv. →

⇒ 2-phase locking protocol ensures serializability.

⇒ 2-phase locking protocol \gg 2PL

Disadv.



Sub:

DBMS
Chap 10

Day

Time:

Date:

28

④ T₁₁ will roll back after failure as it's dep. on T₁₀

⑤ T₁₂ will roll back as it's dependent on T₁₁
⇒ If one transaction fails, cascading rollback occurs.

⇒ If there is interdependency between transactions, roll back of one transaction causes cascading roll back.

Cascading Rollback: Failure of a transaction creates

series of roll back

of several transactions.

Consistency

Consistency

roll back, deadlock

(A) also

(A) loss

losing all the

data if you

commit after

data is lost

and cannot be

recovered

: (A)

BS
SIR

Sub:

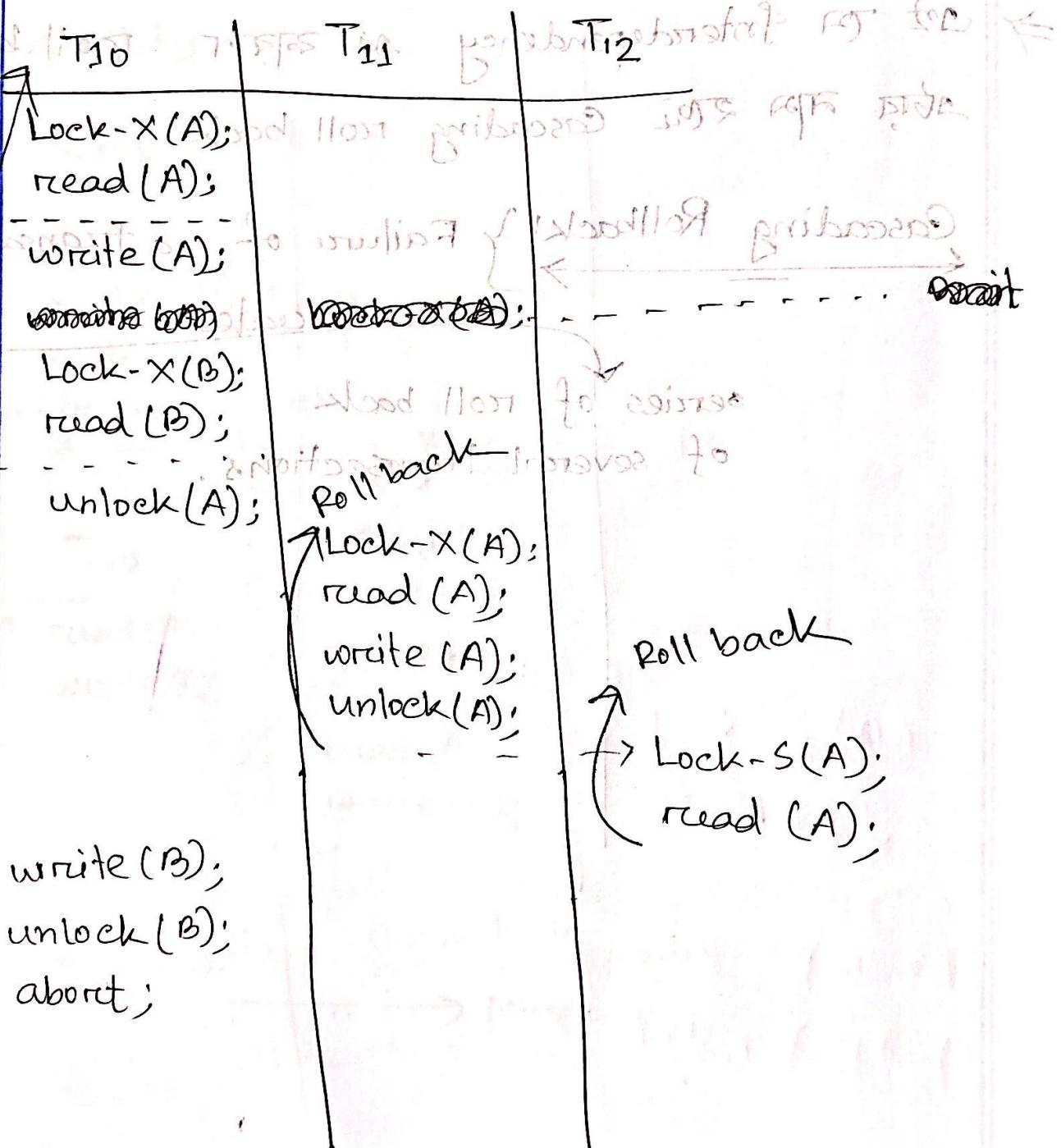
'C' Day
cycle - 12

Day: ✓
Time: 11:40 Date: 21/9/19

Cascade in Schedule:

⇒ Cascade হয় dependency

shape এবং উপর এভিজি করে।



As T_{11}, T_{12} are dependent on T_{10} , এখন

T_{10} roll back করবে তখন যদি হয় T_{11}, T_{12} ও roll back করবে, এটা- cascading rollback.

Two phase locking Protocol:

ensures

Serializability (Consistency)

Problems:

⇒ Cascading Rollback
⇒ Deadlock

} may occur

*** Consistency Problem

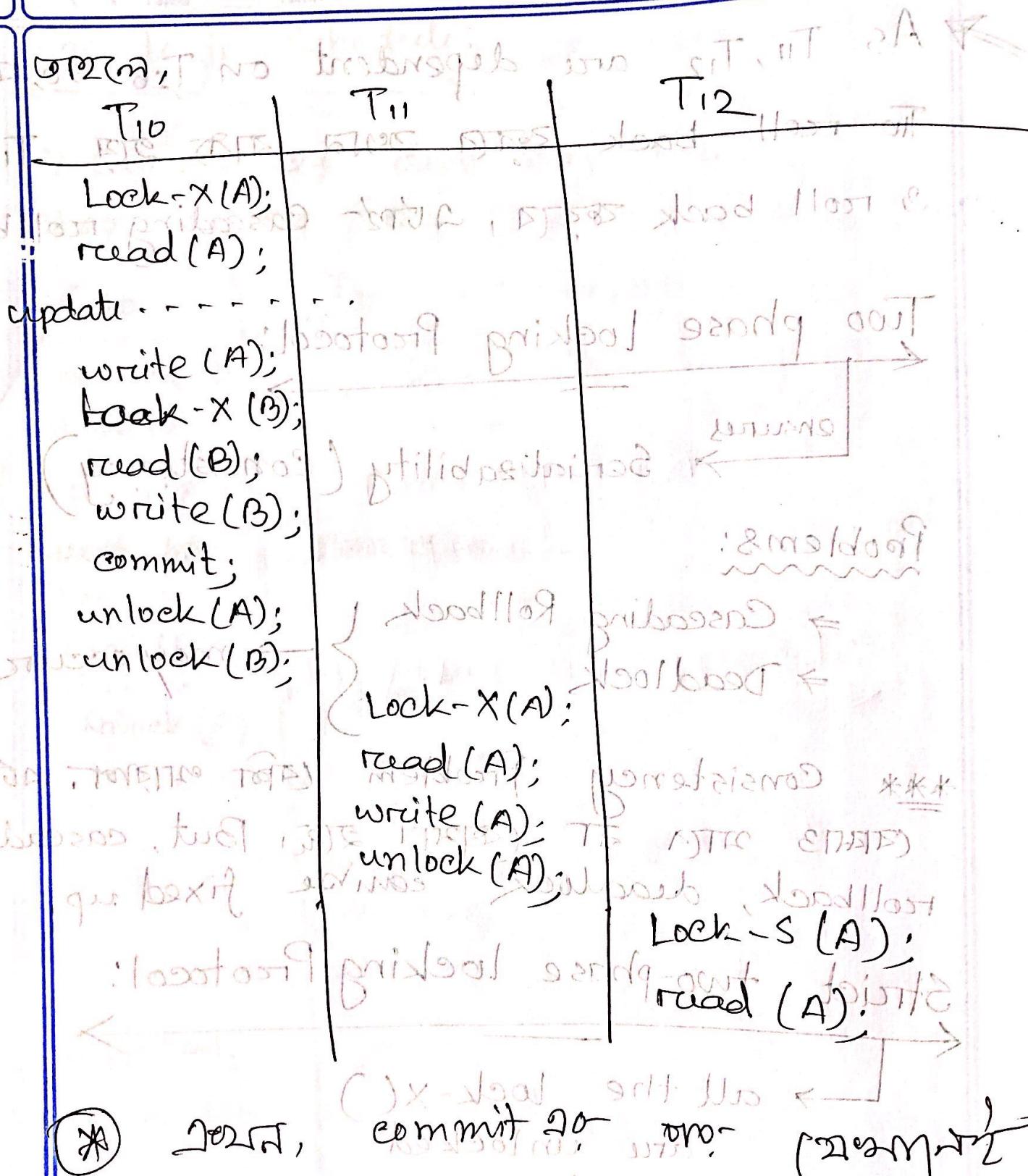
যেকোন মাঝে না কোথায় হবে, But, cascading rollback, deadlock কর্তৃত সমস্যা হবে।

Strict two-phase locking Protocol:

all the lock-x() are unlocked

after commit

*** 'Commit' unlock এর মাধ্যমে cascading problem হবে।



जॉर्डन, commit तो एवं

(प्रारम्भिक)

abort (2126 नाम्यता, (एस (एस,

दोषों के लिए विवरण, दोषों के विवरण)

Sub:

Day: / /
Time: / / Date: / /

Strict Two-phase locking protocol:

Concurrency
factors

all the lock-X(1) & lock-S(2)
are unlocked
after commit

ensures

cascadeless
schedule

Rigorous Two phase locking Protocol:

Isolation
+
Concurrency

all the locks are unlocked
after commit

(কোর্ট দাতা item নিয়ে একাধিক Transaction
কোর্ট নয়ে কোর্ট করতে পারেন, But, এন্ত
data item মাঝে Concurrently execute
করতে পারবে)

Deadlock:

handle

(1) Deadlock Prevention

(2) Deadlock Detection & Recovery

Deadlock Prevention:

(1) Non-preemptive approach

(2) Preemptive approach

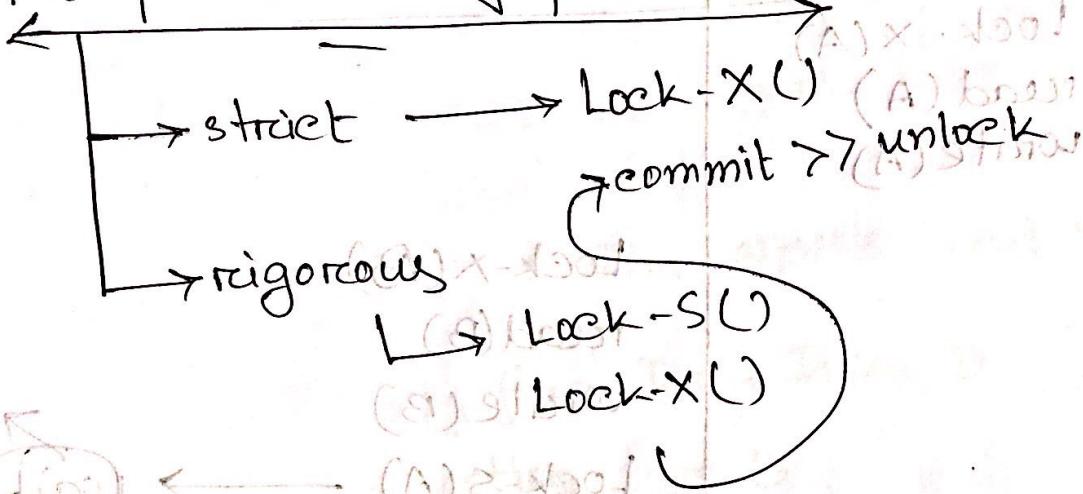
Non-preemptive / Wait-die approach:

senior → এক পঁচান

junior → দুয়ু দিয়া এব

Sub:

Two-phase locking protocol:



deadlock Prevention:

(1) Wait-die approach:

non-preemptive technique.

Example: T_i is holding data item D .
 T_j is requesting for D .

T_j is older than T_i :
 $\text{timestamp}(T_j) < \text{timestamp}(T_i)$

T_j will wait

T_j is younger than T_i :
 $\text{timestamp}(T_i) < \text{timestamp}(T_j)$

T_j will roll back

Sub:

Day

Time:

Date: / /

first Q
ans

Q1-51-18

Q1-51-18

ANS

T₁₃

Lock-X(A)
read(A)
write(A)

Job starts T₁₄ prints out

Lock-X(B)

read(B)

write(B)

Lock-S(A)

Lock-S(B)

in progress sits - fish (1)

ultimately wait

as

T₁₄ rolls

Deadlock
wait senior T₁₃

ultimately wait
as T₁₄ rolls

ultimately wait

Senior T₁₄ waits

IT reads subtopic

(i) quaternary (ii) quaternary

IT writes subtopic

(i) quaternary (ii) quaternary

Non linear IT

Sub: _____

Day: _____
Time: _____ Date: / /

ii) Preemptive Technique / Wound-wait approach

Example: 1) T_i is in Src. আবলে চার্ট; মেধ ছলে

দিয়ে,

2) T_j কর T_i আবলে wait করতে,

T_i is holding for Data Item D

T_j is requesting for D

T_j is older than T_i

T_i will be wounded by T_j if T_j is older than T_i

T_j is younger than T_i

T_j will wait



T_i is waiting

Tr₃
lock - X(A)

read(A)
write(A)

Lock-S(B)

Tr₁₄ T স্বতন্ত্র হলো (ii)

roll back (মেরু পুনৰ্গঠন)

Lock-X(B)

read(B)

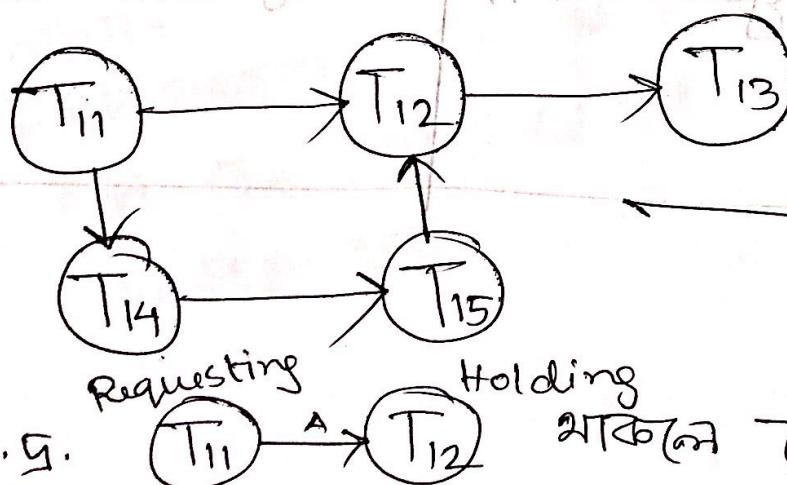
write(B)

Lock-S(A)

wait

Deadlock Detection & Recovery:

Wait for Graph : (Directed Graph)



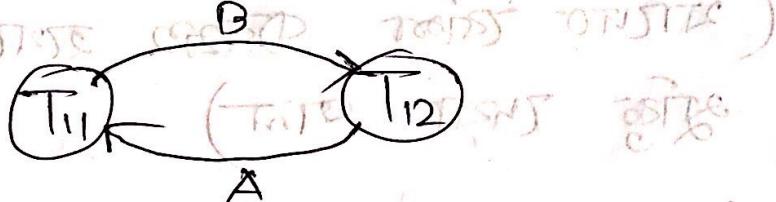
ক্ষয়ালে:
Cycle form
হয়নি।

বিধ. ১. যদি T_{11} অপেক্ষা
wait করছে T_{12} এর ওপর, Resource A এড়ে
পারে না T_{12}

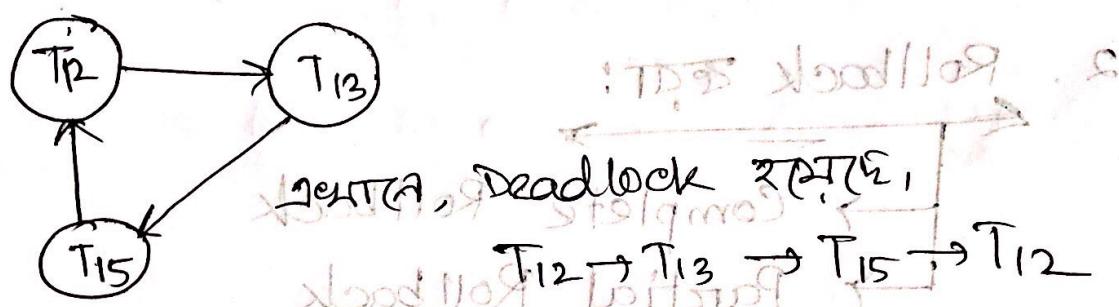
* Wait-for Graph ଦେଖିଯାଇଲୁଛି ପାଇଁ

Deadlock ହେଉଥିବା ଏବଂ ତୈଥେ କୋଣେ Cycle ଥାଏ ?

ଯେତ୍ରମାତ୍ରା



Cycle \Rightarrow Deadlock
denotes



Recovery : \rightarrow Rollback

Action :

1. To select a victim: (କାହାଦେଟିକେ roll back କରାଯାଏ)

minimum cost

depending
on

\rightarrow କଥିତ୍ୟ ଅନ୍ତର୍ଭବ ହେଉଥିବା ସମ୍ଭାବନା ଯାହା

(ଯାଏ କୌଣସି କଥିତ୍ୟ ଅନ୍ତର୍ଭବ ହଲ ତାକୁ roll back
କରାଯାଏ)

\rightarrow ଯାଏ ଅନେକ କଥିତ୍ୟ ଯାହା ତାକୁ 3 roll back
କରାଯାଏ

Sub : _____

Day _____
Time : / / Date : / /

→ ক্ষতিশুলো Data item ক্রিয়াক্রম হয়েছে এবং

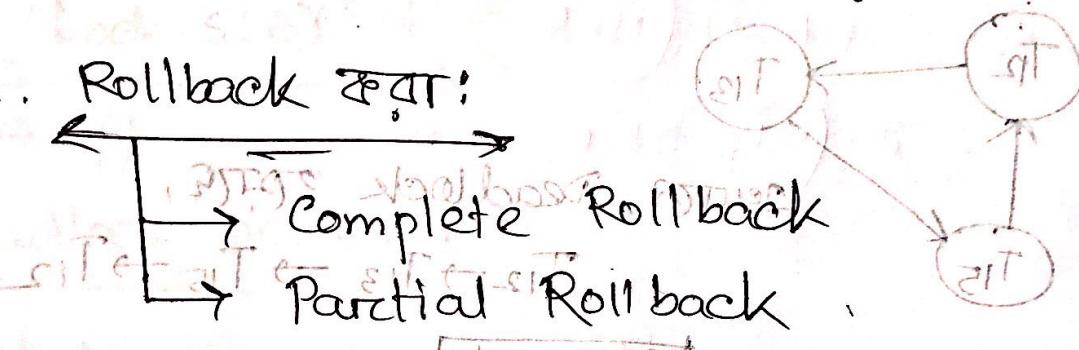
ব্লঙ্গশুলো Data item ক্রিয়াক্রম হয়েছে।

(যাদের সৈকি ক্রিয়াক্রম হয়েছে যা হাত গুড়ে
শুরু হয়ে গেল)

→ Dependency অন্তর্ভুক্ত মাধ্যমে dependencies

এবং এমন তাকে roll-back করা তে এলি।

2. Rollback করা:



(যদি কোথাও পার্ট ক্রান্তি হয়ে থাকে)

ফোস মুনিমিন : initial & possible of T. 1

পুরো ক্রান্তি করা ক্ষমতা

পুরো ক্রান্তি করা ক্ষমতা ক্রান্তি করা ক্ষমতা

পুরো ক্রান্তি করা ক্ষমতা ক্রান্তি করা ক্ষমতা

(ক্রান্তি করা ক্ষমতা)

পুরো ক্রান্তি করা ক্ষমতা ক্রান্তি করা ক্ষমতা

(ক্রান্তি করা ক্ষমতা)

Deadlock Recovery:

→ Rollback

- কাহাকে (Select a victim) → min. cost
- ফুল/পার্টিয়াল Rollback করবে (Full/ Partial)
- মাত্রাবৃত্তি " → depending on min. cost
Particular min. cost এর মাত্রাবৃত্তি rollback
করতে পার্নি, এটাকে বলে (Starvation)

Normalization:

→ Technique to organize data in
multiple tables.

→ aim is to minimize Redundancy.
→ দ্বন্দ্ব করা না.

Redundancy: একই Data'য়ে নির্বাচন ক্ষেত্রগাম্য

উপায়গীতি

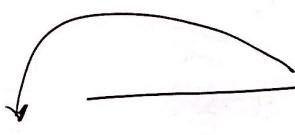
PROBLEMS

Anomalies:

1. Insertion Anomaly
(Abnormality)

Student

Roll No.	Name	Dept	HOD	Officer-Tel
1.	Rancho	ME	Virus	+91 0808
2.	Farhan	ME	Virus	+91 0808
3.	Raju	ME	Virus	+91 0808
4.	ACID	CSE		



এখানে, ME Dept. এই কোন কোর্স নেই
বস্বায় Insert করা লাভ। এটা Insertion

Anomaly.

2. Deletion Anomaly: এখন Rancho স্না

চলে গোলি উদ্যোগ DB ক্ষেত্র মডেলিং,
জাহাজ কুদুর প্রাচী HOD, Officer-Tel
চলে যাব। But আর তা জুনৈতি Virus
(ক) আবে, so DB ক প্রাক্তন দরকার
হিল, এটা Deletion Anomaly.

3. Update Anomaly :

Inconsistency रहे पाएँ, यह आप शायद Update
करने के लिए अलग आप शायद करते होए, यहाँ
आपले DB Concept support करते ना,
तर्फ Inconsistency रहे पाएँ, तो करावी
उत्तर गायब करा या Update.

Normal Form:

1NF:

(1st Normal Form) 1. All values of an attribute

from same domain ↪
type same

*** 3. Each attribute contains

a single value for
each tuple (at most 1)

RollNo	Name	ContactNo
1	Rancho	+910808, +918880, +910707

एकत्र change

संख्या उत्तर एकत्रीकरण

3rd no. same no.

odd संख्या लेखा

योग्य Particular
no. देता करावी

⇒ Normalization ଏ ଏହି ମୋଡ଼୍ଯୁଲୋ ସଂବାଧ କରେ
ଶାଖା ପତ୍ର ଅନ୍ତରୀଳରେ ପାଠୀରେ
ତଥାରେ,

Person

<u>Roll No.</u>	Name	Dept.	HOD
1.	Rancho		

Person contact

<u>Roll No.</u>	<u>SL No.</u>	Contact No.
1	1	+910808
4	2	+910707
1	3	+910808
2	1	
2	2	

2NF (2nd Normal Form)

অথবা প্রথম পর্যায় First Normal form
জোত হয় then 2NF হবে যাব।

→ No partial Dependency can be there
in DB.

RollNo	Name	Dept.	CourseNo	Course Title	Marks	Credit
1			CSE 3107		75	
1			CSE 3109		85	

Partial

Partial

এখনকালে শোবার Redundancy হাতে , কলাধিন
একই Course এই ক্ষেত্রে মোনের Student
এই Reg. হয় , একই Student এই অনেক
course এই Reg. হয় , 2 অক্ষরে Redundancy
হয়।

3010 रेखा Table और ट्रैम्पनी

student

Roll No	Name	Dept

Course	CourseNo	Title	Credit

Marks

RollNo	CourseNo	Marks

Chapter-1

আর অ্যাবস্থা যা পজিশন আছে,

Data ক্ষেত্রে Level এ আজ দ্বা

হয়

Physical Level

(ডিম্বাশীর Data নিয়ে কাজ করে,
Insertion, deletion, Update ইত্যাদি)

Logical Level

প্রয়োজনীয় Data নিয়ে কাজ করে,
যাতে কোন পদ্ধতি নথিক
View level (সুবা DB user কে
বেধানো পর্বতৰ সম্ভ. যার
সাহেন প্রয়োজন হচ্ছে।)

→ Next chapter

→ SQL → Enough কাজের কথী

→ Relational Algebra → SQL এর Query টুলে।
Algebra ক্ষেত্র দেখো। (Small Query টুলো)
(নোট + Symbol)
cte মা (লজিক টাইপ)

Query.



Database Design: (ଏହାର ସମ୍ବନ୍ଧରେ Chap.

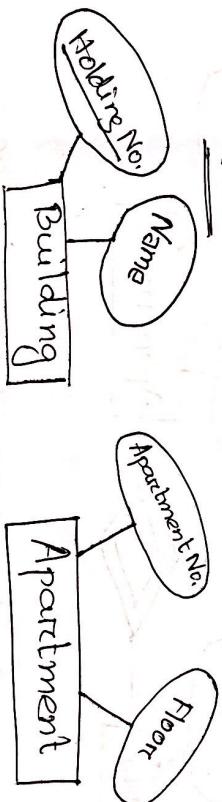
ଏହି ଚାପଟିରେ କୌଣସିବୁ ତଥା ମଧ୍ୟ ଚାପେ।

ବାକି ଯାଇଥାରେ ତଥା

Weak Entity Set

doesn't have sufficient
attr. to form a Prim. Key

ଫେଜାନ:



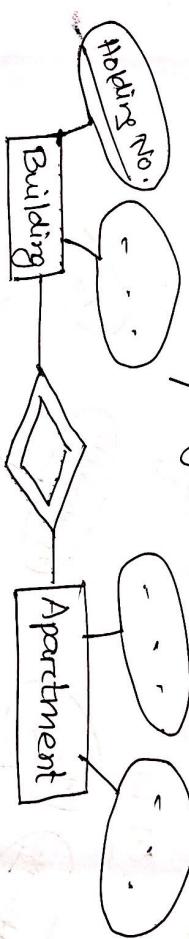
ଏହାରେ, Building No. ମାତ୍ରାକୁ ଅନୁକୂଳେ Unique ଆବଶ୍ୟକ କରିବାକୁ ଯାଏ ନା, So, ଏହା Building Table ମେଳି
ଦୀର୍ଘ ବିଲା ହିଁ ନାହିଁ. Key ହେଉଥିବା କୁବୁରାତେ, ଏହାରେ
Apartment ହେଉଥିବା କୁବୁରାତେ, ଏହାରେ ଏହାରେ ଏହାରେ
ଏହାରେ ଏହାରେ (Building) ହେଉଥିବା କୁବୁରାତେ, ଏହାରେ

Entity Set.

Building Table or Apartment Table
Identify रहे, So Building table का
आकर्षण याते पारि Identifying Relation.

उद्देश्य, एवं इसके लिये निम्न
रखाने का तरीका अंकु - स्थ. एटोल

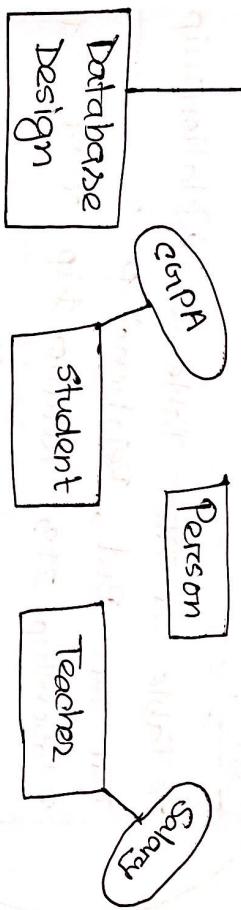
या ही Identifying Relationship.



* * * १५ chapter एवं टेक्स्ट रे Page

मिल E-R Diagram Symbol गोड़,
अपने Must रहेंगे,

Extended -ER- features:



Aggregation:

⇒ Table এই শ্বালে Table এর Relationship আছে
 কিন্তু যদি, But, Relationship এর শ্বালে
 Relationship এর Relationship এর ক্ষেত্রে ক্ষেত্র যদি,
 এবং একই একই Aggregation.

বললেও এটি একই নিয়ে।

✓ Specialization

✓ Generalization

✓ Weak Entity Set & Strong Entity Set

NORMALIZATION:

Roll	Name	Contact No.	Dept	HOD	Office Tel	Course No	Title	Marky
1	John	1234567890	CS	Dr. Smith	033-12345678	101	Software Engg.	85
2	Jane	9876543210	EE	Dr. Johnson	033-12345678	202	Electrical Engg.	78
3	Mike	1234567890	CE	Dr. Williams	033-12345678	303	Civil Engg.	88
4	Sarah	9876543210	ME	Dr. Brown	033-12345678	404	Mechanical Engg.	75

→ একই শ্বালে Redundancy হচ্ছে, হচ্ছে, cause
 একই table এর মধ্যে আছে।

1NF

ନିର.

2NF

ଏକାଳେ ଥିବାଲେଖନ ପରିବାର ନା,

• **Dependency Diagram** *Full*

Roll	Name	Dept	HOD	Tel	Course No.	Title	Marks

Transitive

Partial

Dept. → non key attr. as well as other attributes

Primary key → may or may not related to / share
transitive dependency with other key & attr.

3NF

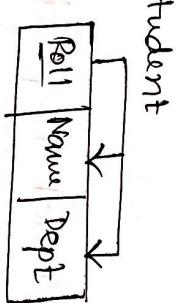
କୋଳେ Transitive Dependency ଥାକି

Transitive dependency on a non-key

Dependency { Attribute

ମାତ୍ର ନା.

ଫ୍ରେଜର ଡିପାର୍ଟ୍ମେଣ୍ଟ:



Student Marks

Roll	Course	Marks

and
marks କାହାର କାହାର HOD, Office, Tel କାହାର

Dept.	HOD	Tel

\Rightarrow BCNF / 3.5NF \rightarrow (Dept, Tel) (at least 2 keys)
(Professor Code NF)

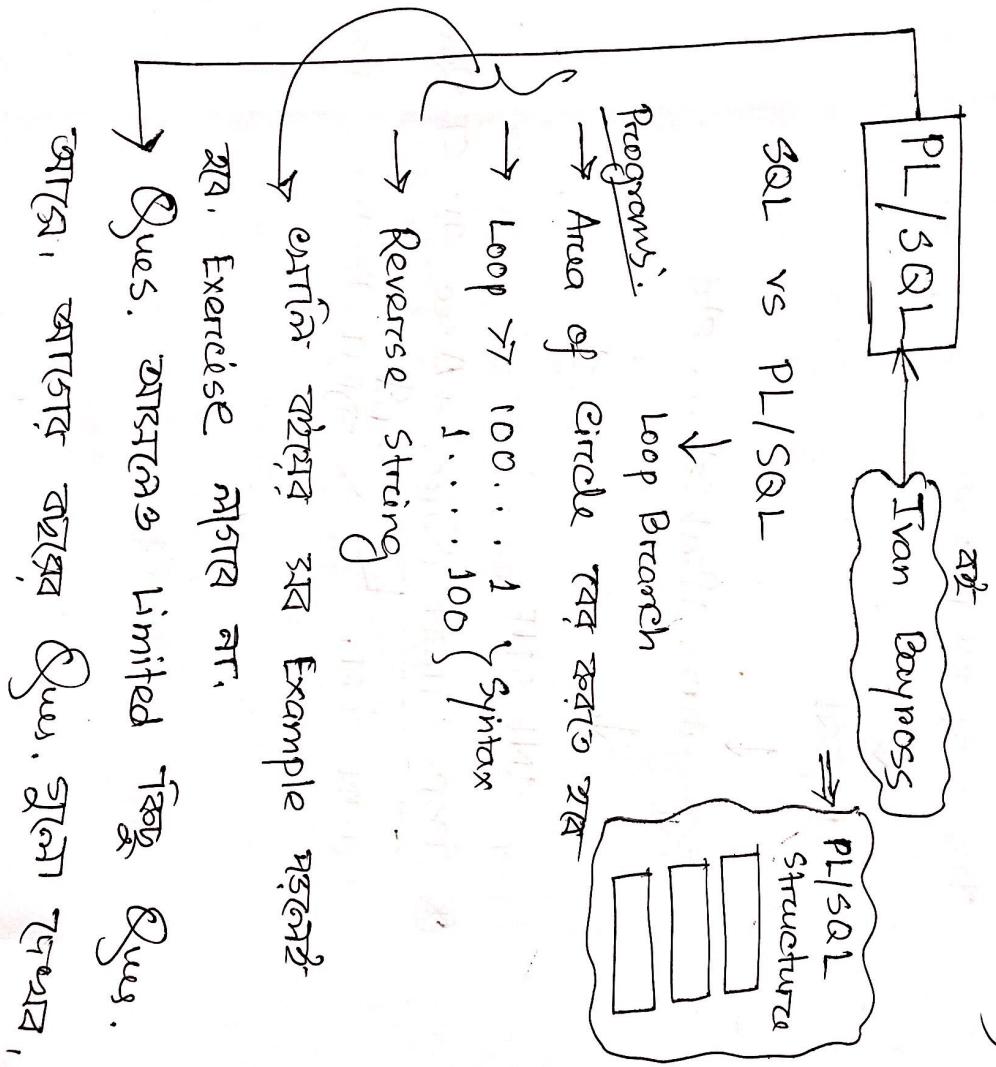
4NF, 5NF \rightarrow ସମ୍ପର୍କକାରୀ କଲ୍ପନା ଯୋଗାର କାମ

But Real life କିମ୍ବା ଲୋକଙ୍କୁ

Transaction:

✓ যা পঞ্জানে হয়েছে কোটি স্থান enough.
অন্ধে রক্ষা রেখা রেখে দাবু।

Timestamp-base protocol (A section
পোতায় আকাল)



Ques.
⇒ Varchar vs Varchar & পার্সেজ কী?

= Ans. StackOverflow তে রেস্যু মার.

পার্সেজ Sequence

ক্লাস এ এ পড়ানো হয়েছে

: MySQL



Aggregation, Weakset ... etc

ANF, SNF

⊗ বিদ্যুৎ স্বাক্ষর Ques. + A sec. নির্ভুল Ques.
3-NF নিঃব, আলোচনা স্বাক্ষর Portion (যাই

RAID:

⇒ Transaction complete হলে system crash
হোলি data stored আছে কি? কীভাবে?
অন্য ফাইলস কপি করে, এখন কৈমন RAID
এর concept আছে,

RAID → Redundant

Array
of
Independent
Disks

⊗ অধীন লক্ষণ মাপার পদ্ধতিই ইত্যোঁ
⊗ এখান প্রক পদ্ধতি Concept আছে

Mirroring, Striping, etc.

Error Checking

→ এবের আগতে দ্রুত প্রক্রান্ত
নাই.

The chapter এর রেখাচিত্র Exercise এর
সুলভ, সহজ, কমপক্ষে ৩ঠান রেখাচিত্র
common আছাব না।

XAMPP → Dev. Env.

SQL & PLSQL → Language

Maria DB / SQL, MySQL → RDBMS

এই কোনোর লাঙুগে
Oracle DB

Concept of
Cursor & Trigger

এটা মুক্ত প্রত্যেক পাঠ্যে।

All Sir প্রত্যেক পাঠ্যে।