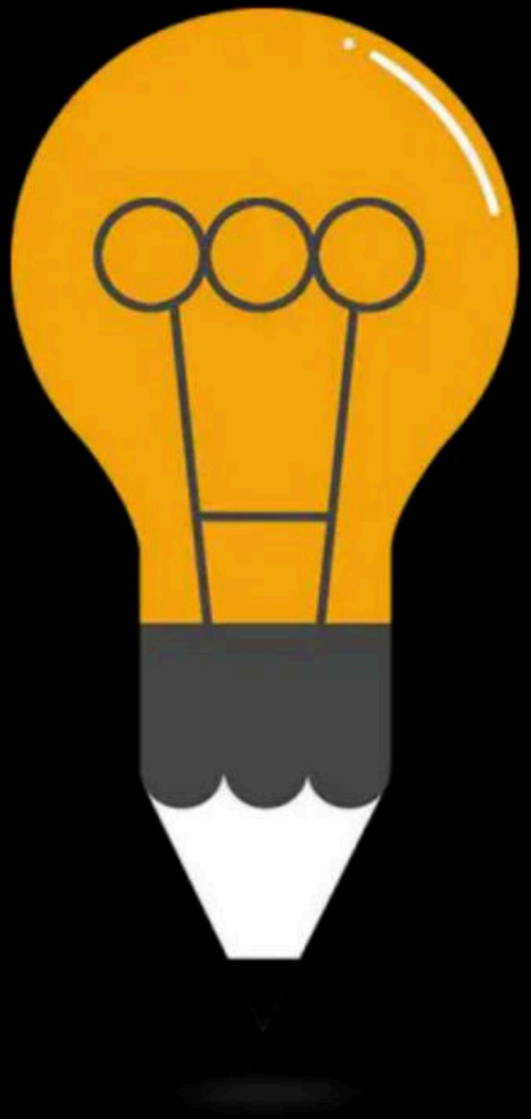
unacademy

File Organization and Indexing: Part I

Complete Course on Database Management System

Vishvadeep Gothi • Lesson 29 • July 26, 2023

unacademy



Today ⇒ 2 session
11:15 am
4:00 pm
only today

DBMS

Timestamp based protocols & Indexing

By: Vishvadeep Gothi

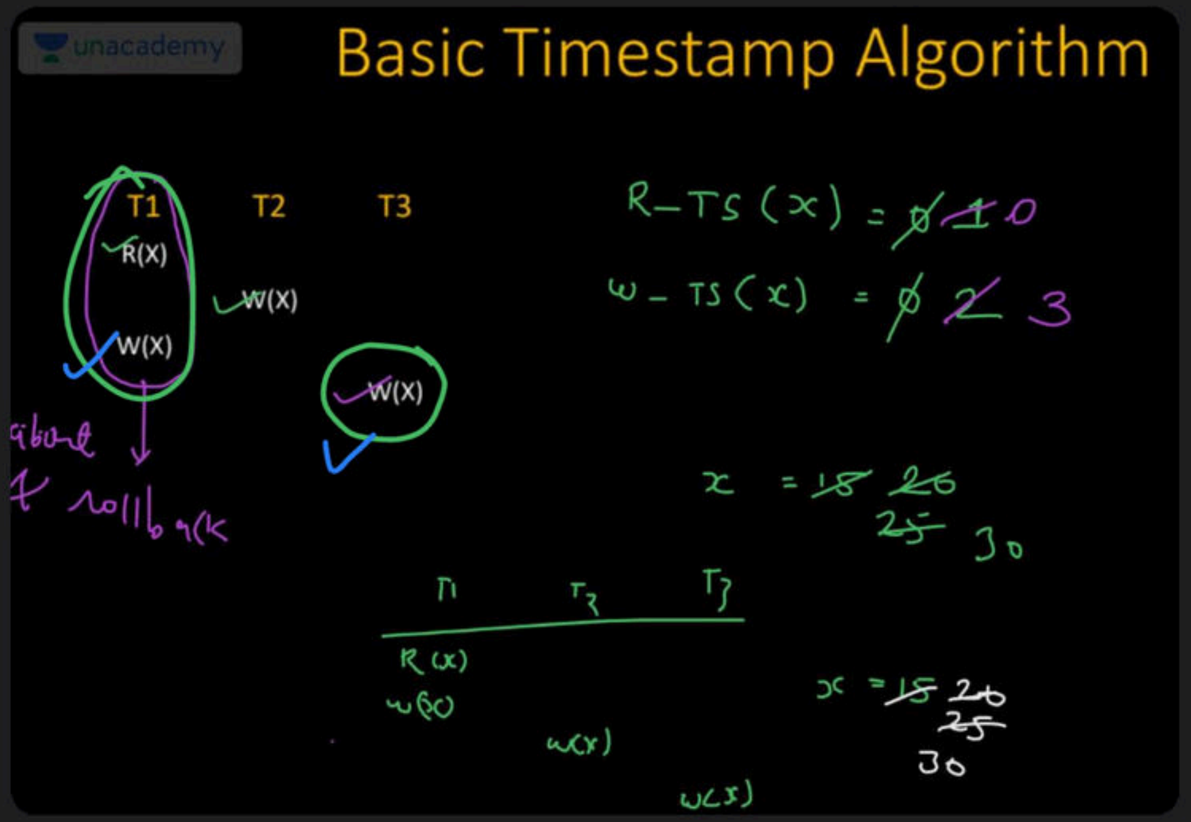
16, 17

Thur
fri
sat
sunday
11:15 am
11:00 am

unacademy

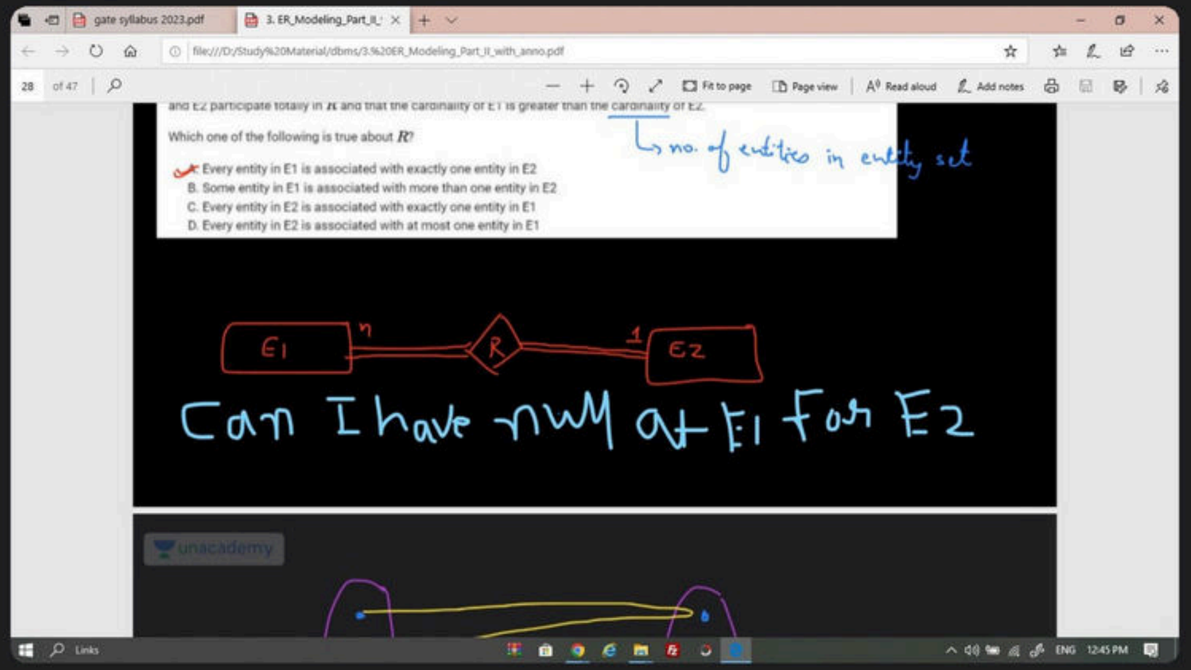
▲ 1 • Asked by Vaishnavij...

sir in this example if we are not considering the effect of W(x) by T1, then the final value should be 25, how 30?



▲ 1 • Asked by Kumar

Please help me with this doubt



unacademy

▲ 2 • Asked by Shreyas

Please help me with this doubt

Consider the following log sequence of two transactions on a bank account, with initial balance 12000, that transfer 2000 to a mortgage payment and then apply a 5% interest.

1. T1 start
2. T1 B old $=$ 12000 new $=$ 10000
3. T1 M old $=$ 0 new $=$ 2000
4. T1 commit
5. T2 start
6. T2 B old $=$ 10000 new $=$ 10500
7. T2 commit

Suppose the database system crashes just before log record 7 is written. When the system is restarted, which one statement is true of the recovery procedure?

A. We must redo log record 6 to set B to 10500
B. We must undo log record 6 to set B to 10000 and then redo log records 2 and 3
C. We need not redo log records 2 and 3 because transaction T1 has committed
D. We can apply redo and undo operations in arbitrary order because they are idempotent

unacademy

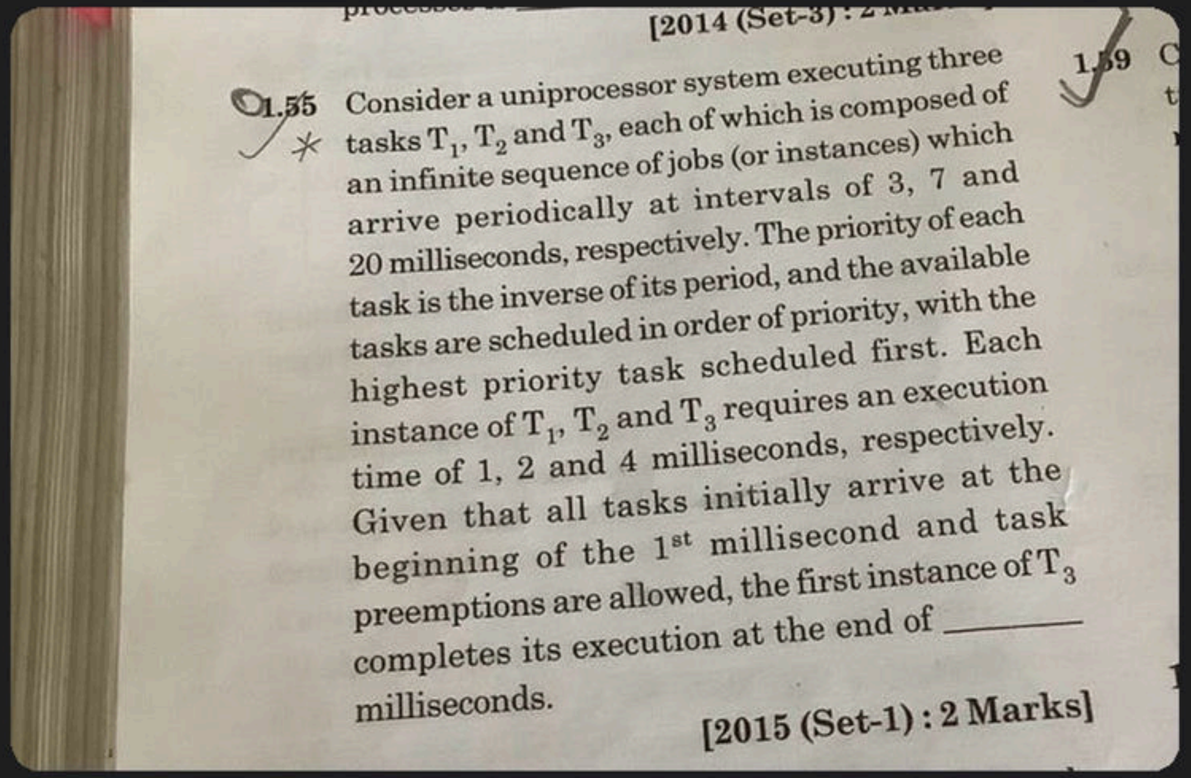
▲ 1 • Asked by Divya

Sir wait_&_die mein younger one(if requests) will get aborted and will restart with "same timestamp"? sir yeh same timestamp mtlb? same arrival time to nhi hoga as it was before?

unacademy

▲ 1 • Asked by Faizan

Please help me with this doubt



[2014 (Set-3) : 2 Ma...

1.55 * Consider a uniprocessor system executing three tasks T_1, T_2 and T_3, each of which is composed of an infinite sequence of jobs (or instances) which arrive periodically at intervals of 3, 7 and 20 milliseconds, respectively. The priority of each task is the inverse of its period, and the available tasks are scheduled in order of priority, with the highest priority task scheduled first. Each instance of T_1, T_2 and T_3 requires an execution time of 1, 2 and 4 milliseconds, respectively. Given that all tasks initially arrive at the beginning of the 1st millisecond and task preemptions are allowed, the first instance of T_3 completes its execution at the end of _____ milliseconds.

[2015 (Set-1) : 2 Marks]

GS PYQ vishvadeep

unacademy

Timestamp

Read Timestamp(A): Youngest transaction who read A

Write Timestamp(A): Youngest transaction who write A

unacademy

Basic Timestamp Algorithm (Timestamp ordering)

Whenever a Transaction T issues a W_item(X) operation, check the following conditions:

- If R_TS(X) > TS(T) or if W_TS(X) > TS(T), then abort and rollback T and reject the operation.
- Else execute W_item(X) operation of T and set W_TS(X) to TS(T).

Whenever a Transaction T issues a R_item(X) operation, check the following conditions:

- If W_TS(X) > TS(T), then abort and rollback T and reject the operation, else
- If W_TS(X) <= TS(T), then execute the R_item(X) operation of T and set R_TS(X) to the larger of TS(T) and current R_TS(X).

unacademy

Basic Timestamp Algorithm

Restarted transaction gets a younger timestamp

↳ eliminates starvation

Basic Timestamp Algorithm

T1	T2
	R(X) or W(X)
W(X)	

T1	T2
	W(X)
R(X)	

unacademy

Basic Timestamp Algorithm

T1	T2	T3
		R(X)
	R(X)	
R(X)		
W(X)		

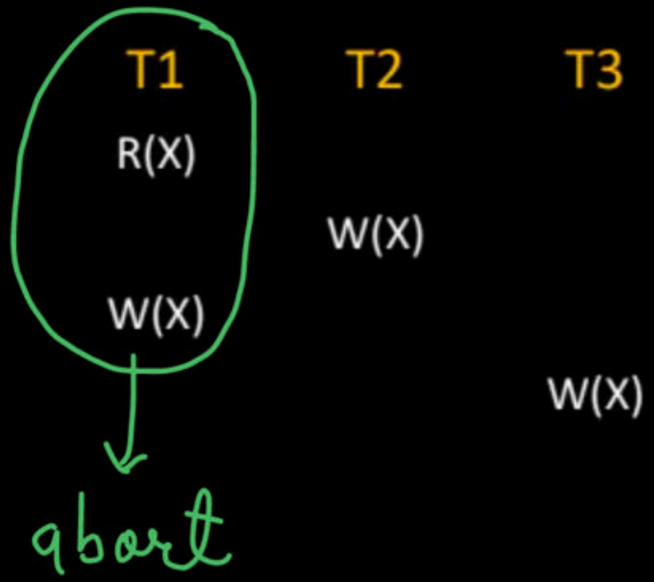
unacademy

Question

T1	T2	T3	T4
			R(X)
		R(X)	
	R(X)		
	W(X)		
		W(X)	
			W(X)
R(X)			

unacademy

Basic Timestamp Algorithm

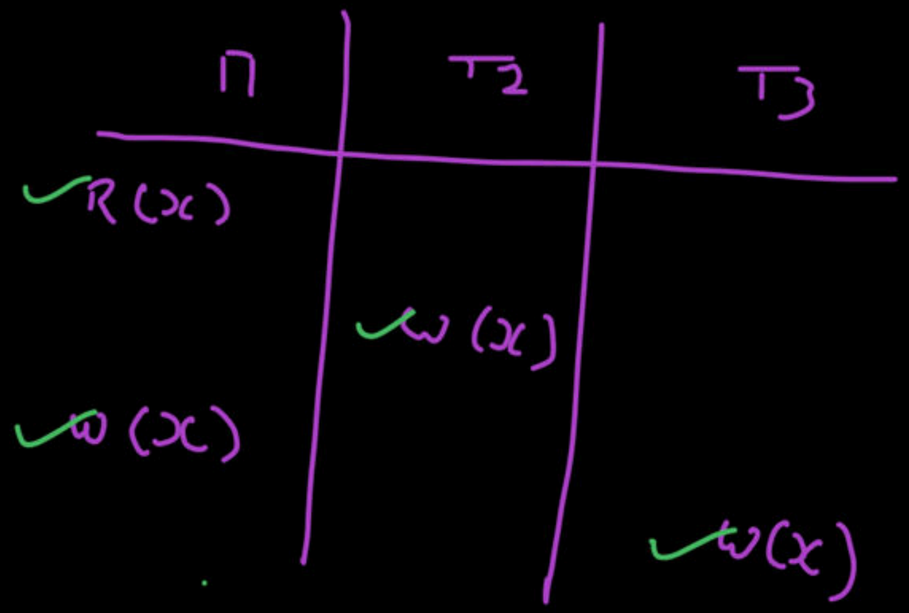


unacademy

Thomas Write Rule

- Read is same as basic timestamp rules
- Write(A) in transaction T:
  - If R_TS(A)>TS(T) then abort T, rollback and restart T
  - Else If W_TS(A)>TS(T) then skip write operation
  - Else perform Write(A) of T and update W_TS(A)=TS(T)

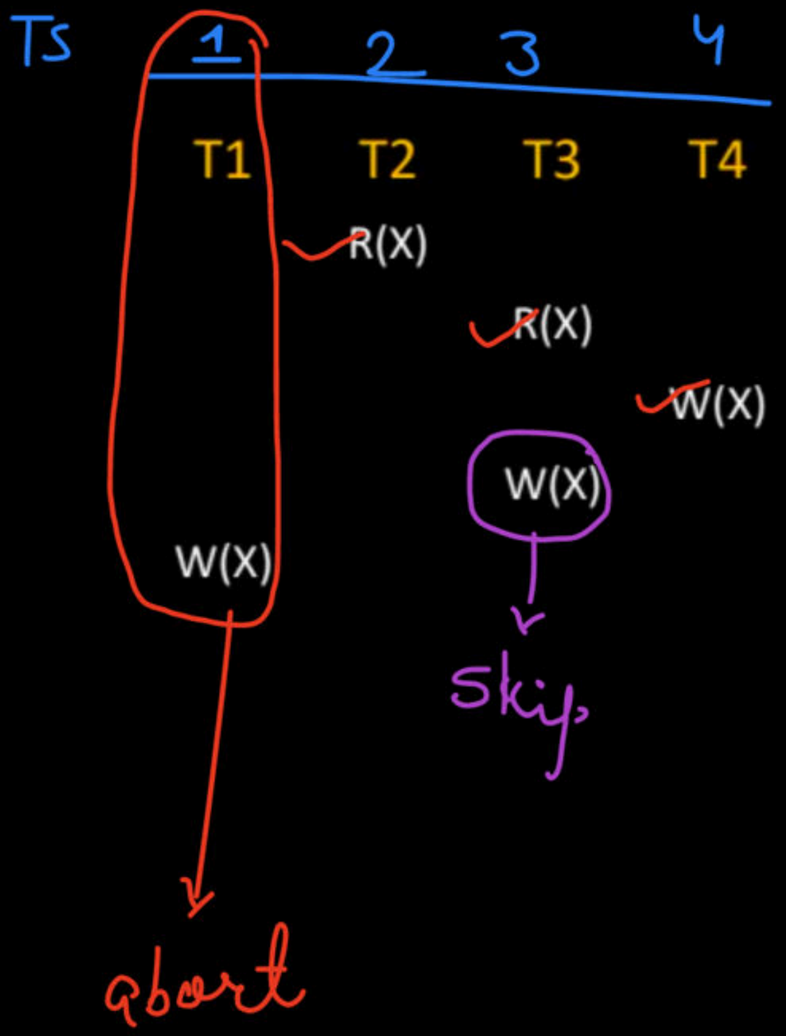
W_TS(A) > TS(T)



R_TS(x) = ~~0~~ 1

W_TS(x) = ~~0~~ ~~2~~ 3

Question



R_Ts(x) = ~~0~~ ~~2~~ 3

W_Ts(x) = ~~0~~ 4

T1 T2 T3 T4

R(x)

W(x)

R(x)

W(x)

W(x) → skip

W(x)

W(x) → skip

R_TS(x) = ∅ x2

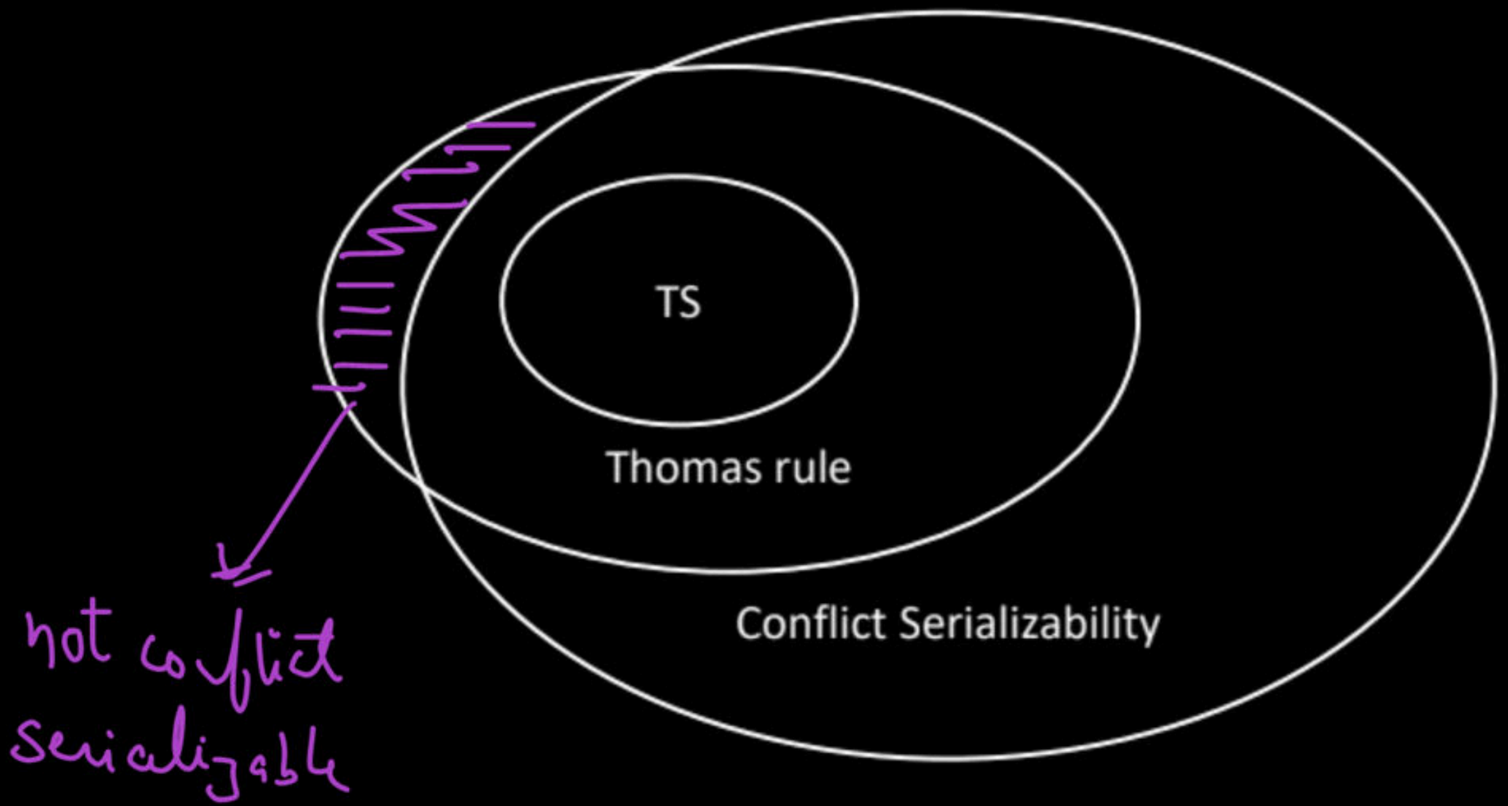
W_TS(x) = ∅ x 3 4

Timestamp ordering algo

- serializability
- No deadlock

Timestamp

- Basic TS allows conflict serializable schedules
- Thomas rule allows more than conflict serializable schedules



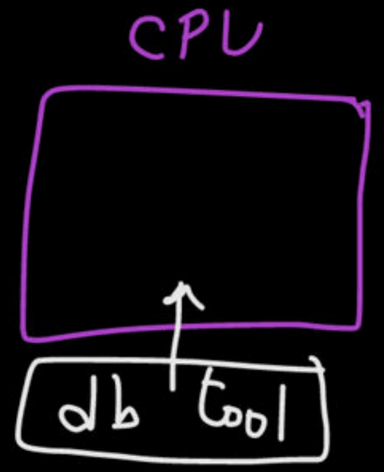
unacademy

No Need to Study

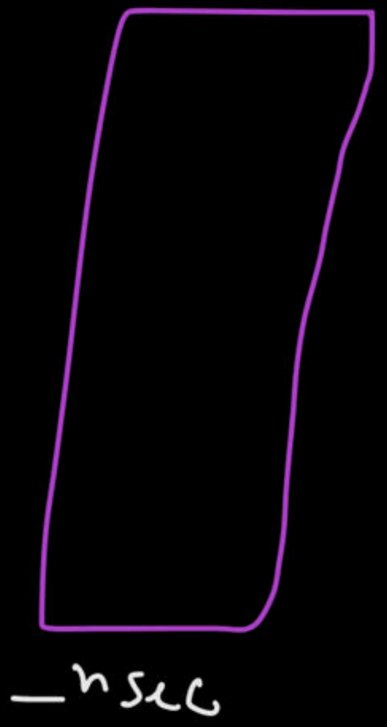
- Multiversion Protocol
- Multigranularity Protocol

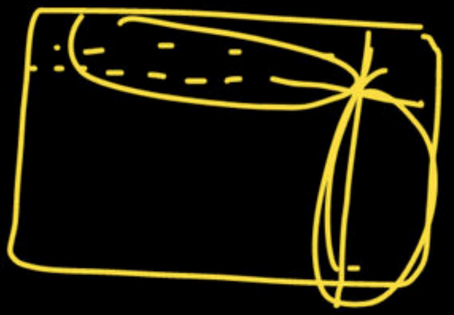
Memory Structure

- Main Memory
- Secondary Memory



select * from table





Query optimizer :-

Runs query in such a way that min. time taken to fetch records from disk.

```
select Ename from Employees
where dateofjoining > 2001 and
salary > 50000
```

ex:- Employees

Eid	Ename	Gender	salary	dateofjoining

assume.

no. of records for d.o.j. > 2001 ⇒ 5000

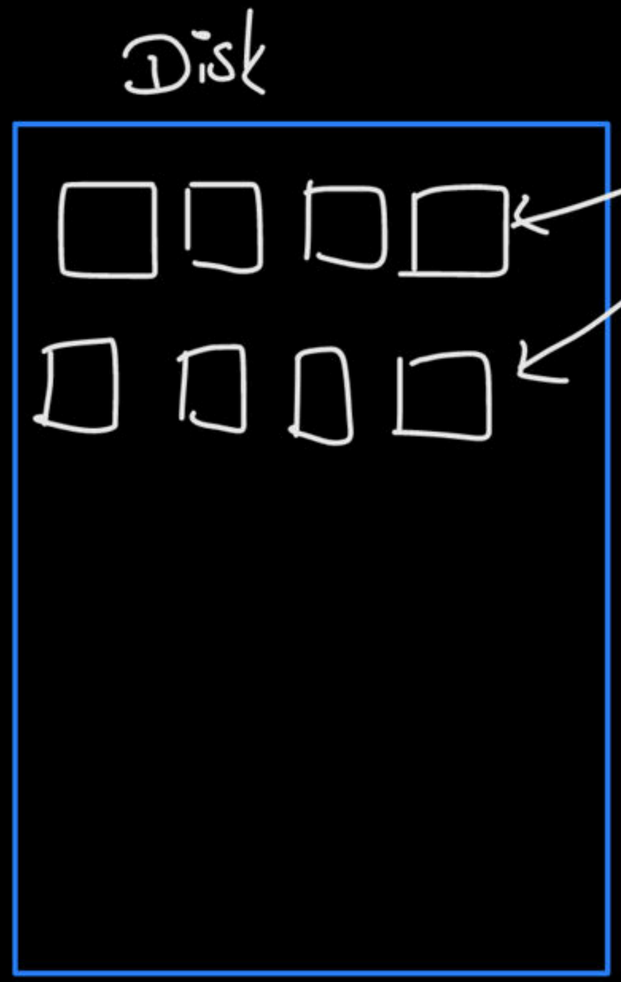
no. of records for Salary > 50000 ⇒ 100

---

and

both condit^n here record => 82

Disk Blocks and Record Storages



ex 1 :-

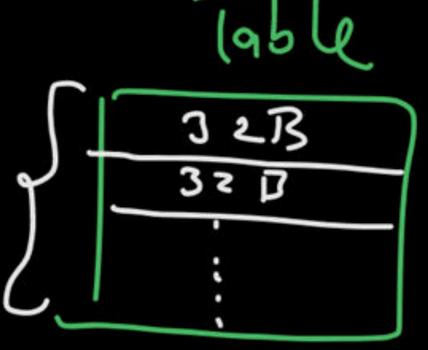
Block size = 1K bytes

Table ⇒ 16 records

record ⇒ 32 bytes

Table size = 16 * 32 B

= 2^9 B = 512 B



entire table stored in one block only

ex. 2

Table ⇒ 512 records

each record ⇒ 32 bytes

block size ⇒ 1 K byte

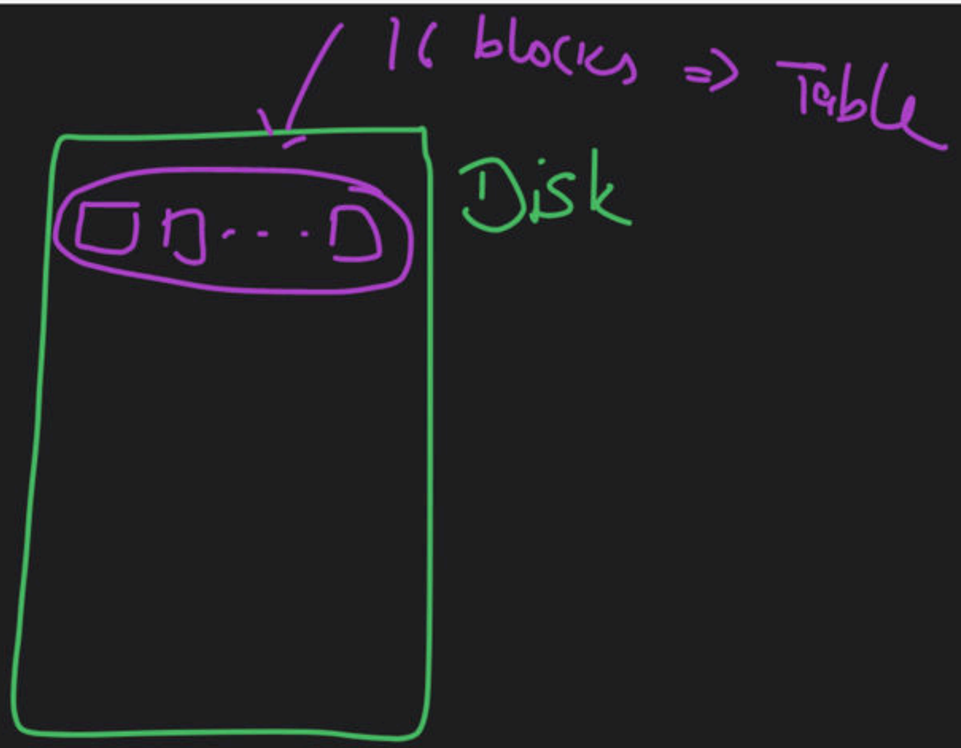
Table size = $512 * 32B = 2^{14}B = 16KB$

no. of block required to store entire table = $\frac{16KB}{1KB} = 16$ blocks

```
select Ename from employee
where eid = 76
```

⇓

Search each record linearly to get the required record.



select * from employee where eid < 90

⇓

In General all blocks are accessed linearly

but if records are stored in sorted order of eid,
then only till that block where eid becomes 90.

unacademy

Eid	Ename	Salary
1	E1	50000
2	E2	60000
3	E2	10000
4	E3	45000
5	E4	42000
6	E3	65000
7	E2	55000

⇒ Physical storage of db table

→ ① Basis on which column the records must be ordered in disk so that max. no. of times Queries can provide Quick results

② can we reach to a specific record directly.

↳ Through **indexing**

unacademy

File Organization

- Sorted File Organization
- Heap File Organization

unacademy

Fixed Length Records

Fixed Length Records: What if deletion done?

Fixed Length Records: What if deletion done?

Add.	Header Last = 8	Pointer to next free address = 2	
1			
2		5	
3			
4			
5		7	
6			
7		8	
8		NULL	Last
9			

unacademy

Variable Length Records

Add.	Header Last = 8	Pointer to next free address = 2	
1			
2		5	
3			
4			
5		7	
6			
7		8	
8		NULL	Last
9			

unacademy

Why Indexing

unacademy

Index File

unacademy

Indexing Techniques

- Clustered Indexing
- Non-Clustered Indexing

unacademy

Clustered Indexing

- Data order and index order are same

unacademy

Non-Clustered Indexing

- Data order and index order are not same

unacademy

Dense Vs Sparse Index

1. Dense: Index record is for each database record
2. Sparse (non-dense): Index record is for a few database records only

Spanned vs Unspanned File Organization

unacademy

Indexing Techniques

- Primary Indexing
- Clustering Indexing
- Secondary key Indexing
- Secondary non-key Indexing

unacademy

Primary Indexing

- Indexing done on primary key or any super key
- Data must be ordered on index
- Its always sparse index

unacademy

Clustering Indexing

- Indexing done on non-key field
- Data must be ordered on index field

unacademy

Secondary key Indexing

- Indexing done on primary key or any super key
- Data must not be ordered on index field
- It can be dense of sparse index

unacademy

Secondary Non-key Indexing

- Indexing done on non-key field
- Data must not be ordered on index field

Primary Index

Rno.	Block Pointer
1	B1
5	B2

Rno.	Block Pointer
9	B3
13	B4

1	A
2	B
3	C
4	D

5	A
6	B
7	E
8	W

9	V
10	D
11	I
12	A

13	C
14	S
15	H
16	A

unacademy

Primary Index

1	A
2	B
3	C
4	D

5	A
6	B
7	E
8	W

9	V
10	D
11	I
12	A

Select * from Students

where rno=1

unacademy

Primary Index

1	A
2	B
3	C
4	D

5	A
6	B
7	E
8	W

9	V
10	D
11	I
12	A

Select * from Students

where rno=12

unacademy

Question

Consider a database file of 65536 records, each record of size 64 bytes. Key field is 10 bytes and block pointer size is 22 bytes. Assume that block size is 256 bytes.

1. The number of blocks required to store file?
2. The number of blocks required to store the index file for primary indexing?

Clustering Index

S_Name	Block Pointer
A	B1
B	B2
B	B3
E	B4

1	A	B1
2	A	
3	A	
4	B	
5	B	B2
6	B	
7	B	
8	B	
9	B	B3
10	B	
11	C	
12	D	
13	E	B4
14	F	
15	F	
16	G	

Clustering Index

S_Name	Block Pointer
A	B1
B	B1
C	B3
D	B3
E	B4
F	B4
G	B4

1	A	B1
2	A	
3	A	
4	B	
5	B	B2
6	B	
7	B	
8	B	
9	B	B3
10	B	
11	C	
12	D	
13	E	B4
14	F	
15	F	
16	G	

unacademy

Question

DB File size 1G records

Record size = 64bytes

Block size = 4096bytes

Index field = 10 bytes

Block pointer size = 22 bytes

Number of distinct values in index file = 16384

Indexing is done on non-key, data is ordered on non-key and indexing is done for each distinct non-key value

unacademy

Happy Learning.!