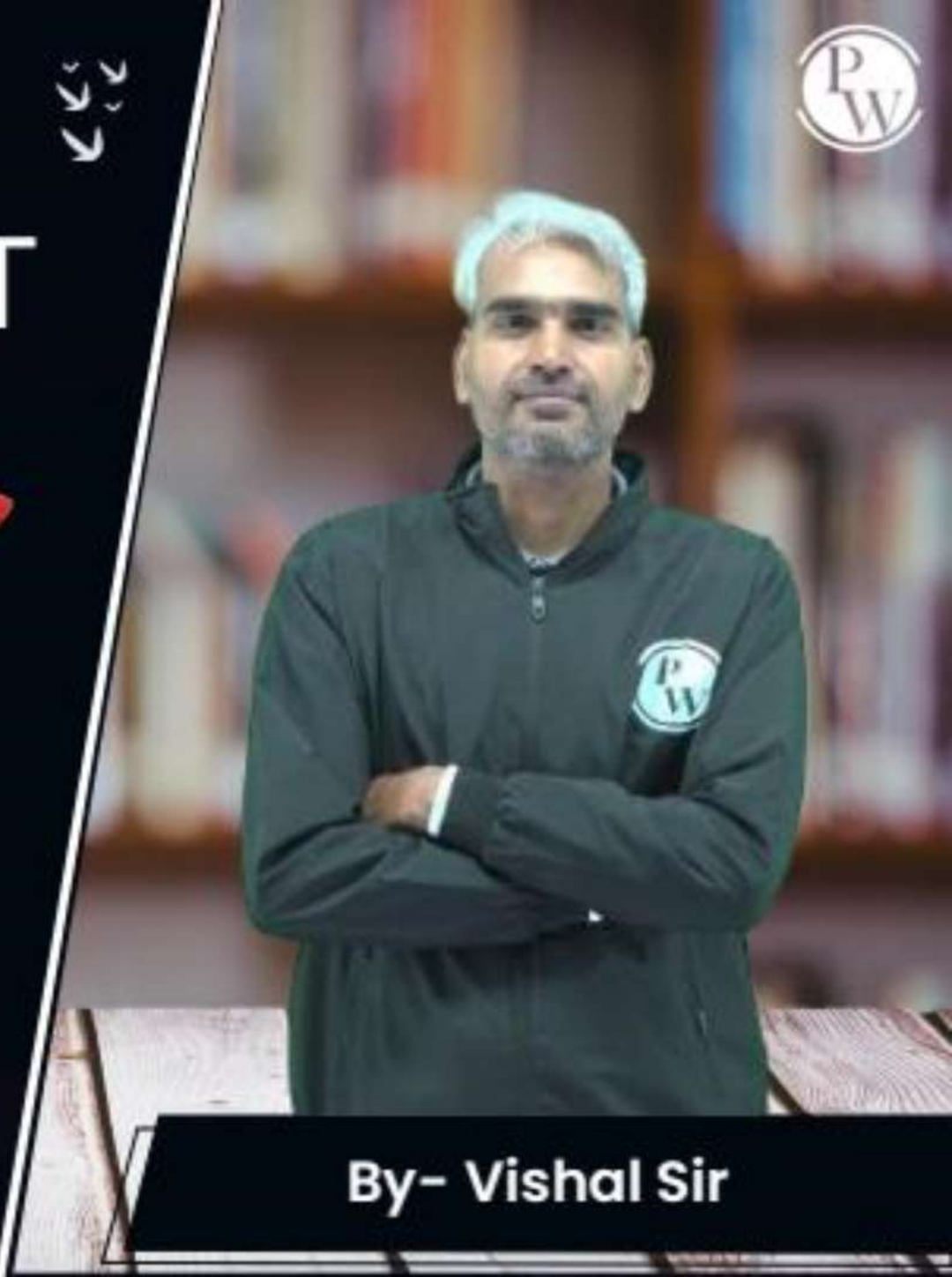
Computer Science & IT

Database Management System

File organization And Indexing

Lecture No. 01



By- Vishal Sir

Recap of Previous Lecture

Topic Basic time stamp ordering protocol

Topic Time stamp ordering protocol with Thomas write rule

Slide

Topics to be Covered

- Topic: Wait-die protocol for deadlock prevention
- Topic: Wound-wait protocol for deadlock prevention
- Topic: Database, File and Records
- Topic: Organization of Records
- Topic: Categories of index

Topic : Deadlock prevention algorithms

Time stamp assigned by DBMS can be used to prevent the deadlock in lock based protocols.

If dependency graph is Cyclic, then deadlock may be possible

$\Rightarrow$

Dependency graph

eg: $T_1 \rightarrow T_2$

it means transaction T_1 requires resource hold by transaction T_2

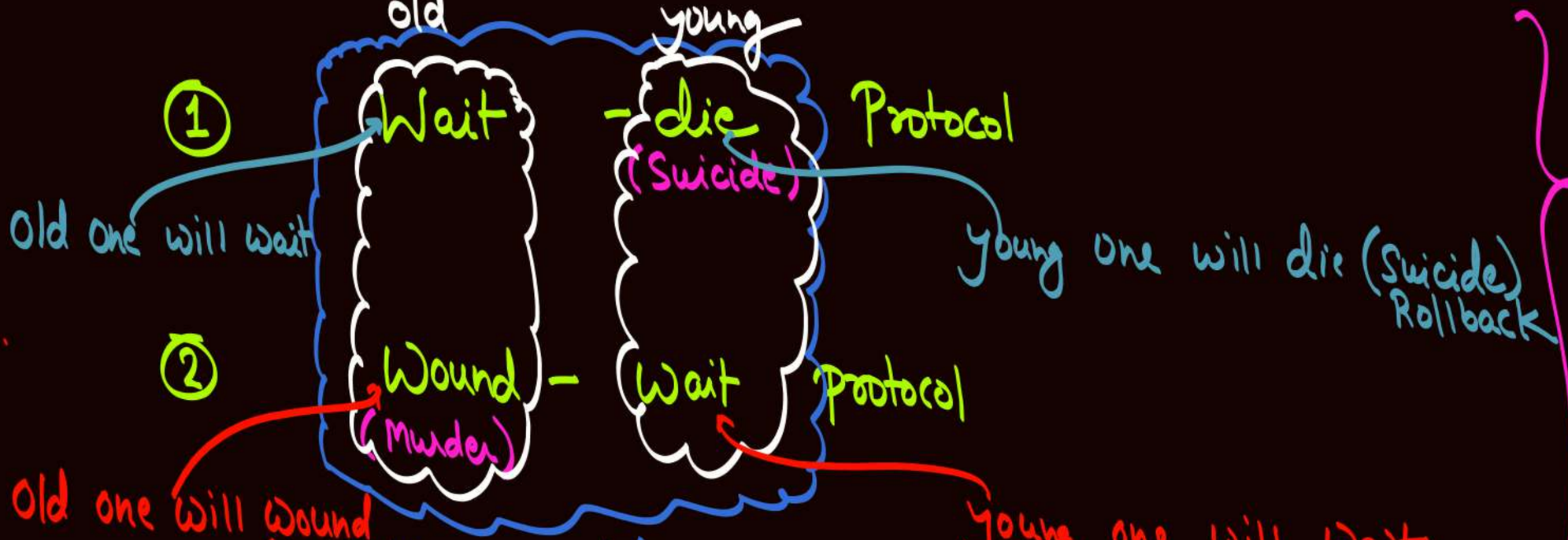
$\Rightarrow$

Cyclic dependency graph

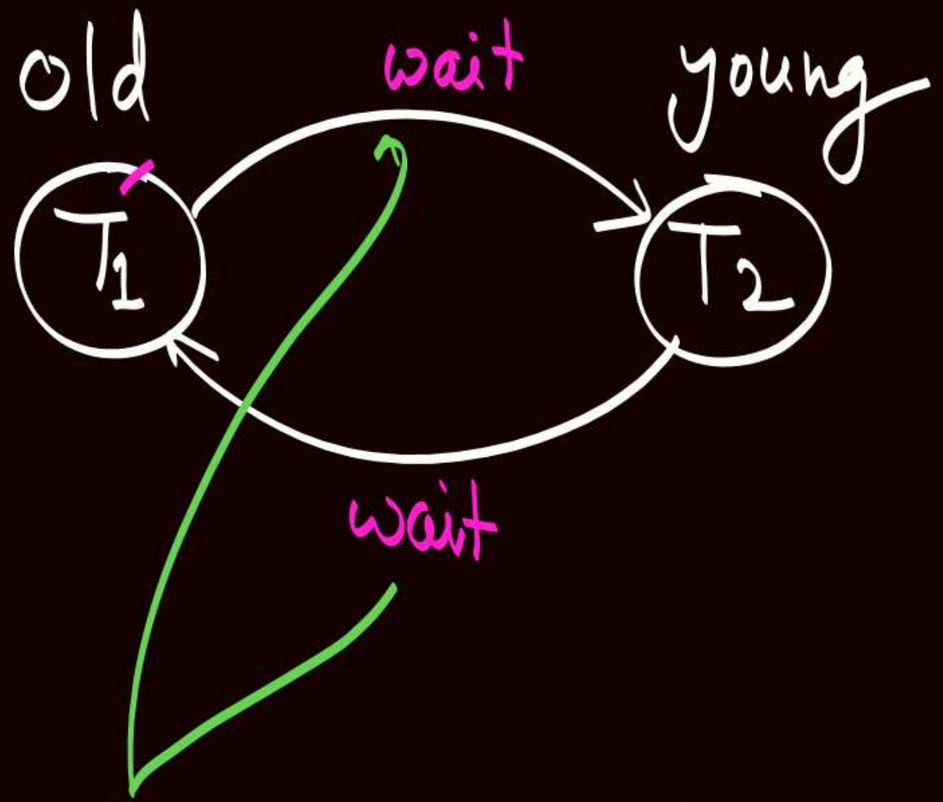
if cycle exists then deadlock may be Possible

If we can Prevent the formation of Cycles in the dependency graph, then there will be no deadlock

There are two algorithms defined to prevent deadlock in lock based protocols.



Rollbacking will always be w.r.t. young transaction



If waiting is allowed in both direction.
then formation of cycle is possible

old - young

Topic : Wait-die algorithms

Let T_i & T_j are two transactions such that $TS(T_i) < TS(T_j)$ { i.e. T_i : old & T_j : new }

① If old transaction { i.e. T_i } requires the resource held by new transaction { i.e. T_j }, then old one is allowed to wait

"Ti" is allowed to wait

② If young transaction { i.e. T_j } requires the resource held by old transaction { i.e. T_i }, then rollback young transaction

New transaction not allowed to wait (die)

old young

Topic : Wound-wait algorithms

Let T_i & T_j are two transactions such that $TS(T_i) < TS(T_j)$
{i.e: T_i = old & T_j : new}

① If old transaction (T_i) requires the resource held by new transaction {T_j}, then old one will Wound young one {i.e: Rollback}

Rollback {old one will kill new one} {Wound}

② If young transaction {i.e; T_j} requires the resource held by old transaction {i.e; T_i}, then young transaction is allowed to wait

Tj is allowed new to wait

New - Chapter

Ch-4 : File organization & Indexing

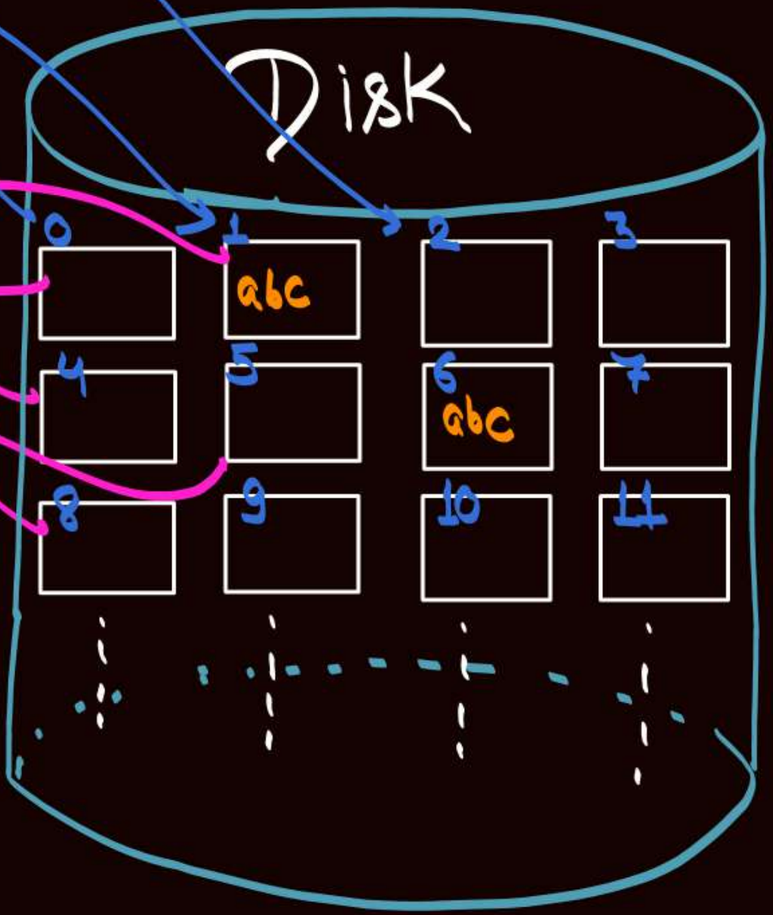
Topic : Database-File-Records

- Database is a collection of file.
- Files are used to store the records
- Disk blocks are used to store the records of the file.

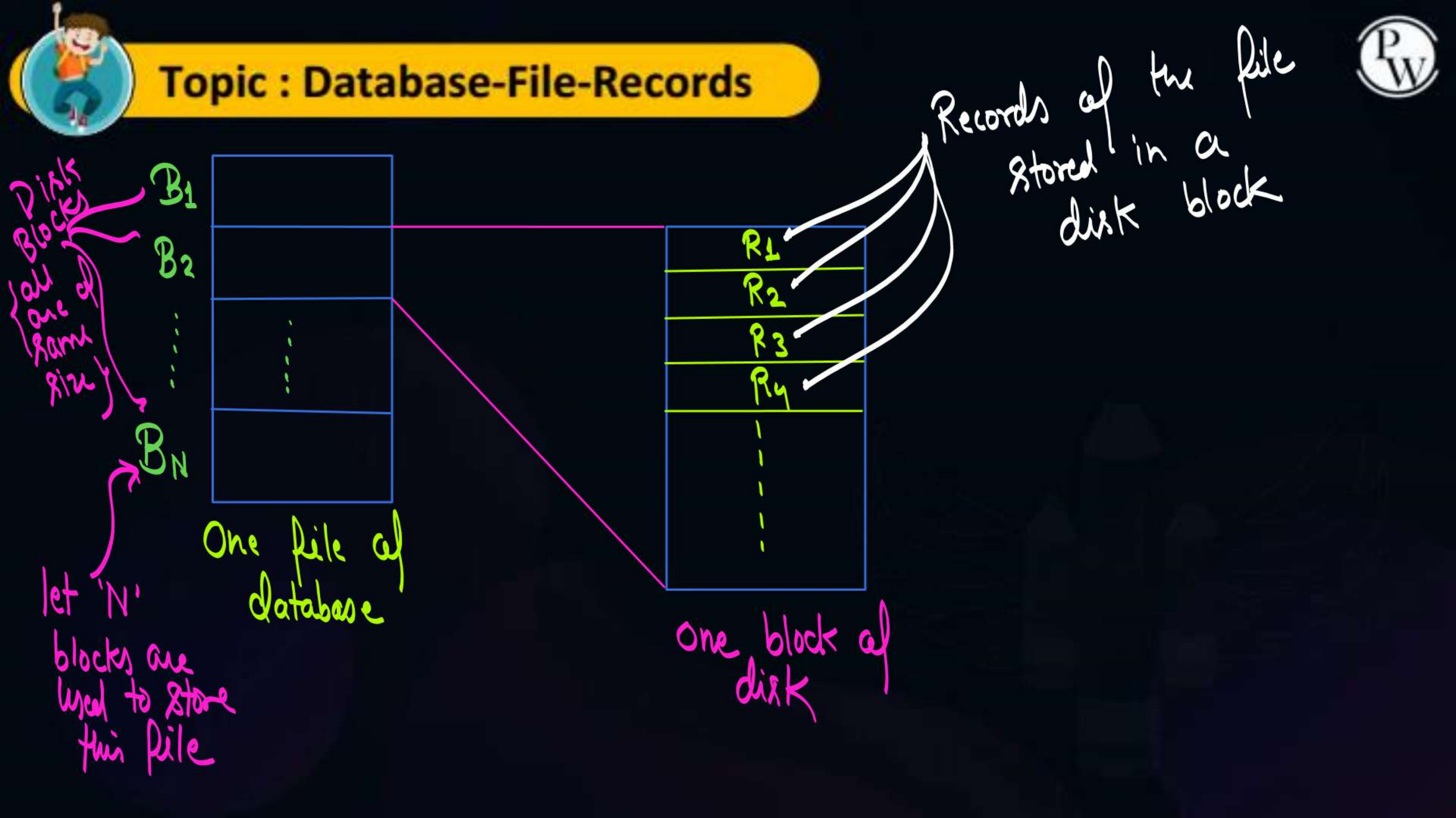
  Records of different files can not be stored in the same block of the disk.

* Each block of the disk will be of Equal size
* Each block of disk is identified by a unique number known as "disk block address"
* One disk block can store multiple records of the same file (Not of different files)

Disk Blocks



Topic : Database-File-Records



Topic : Types of Records

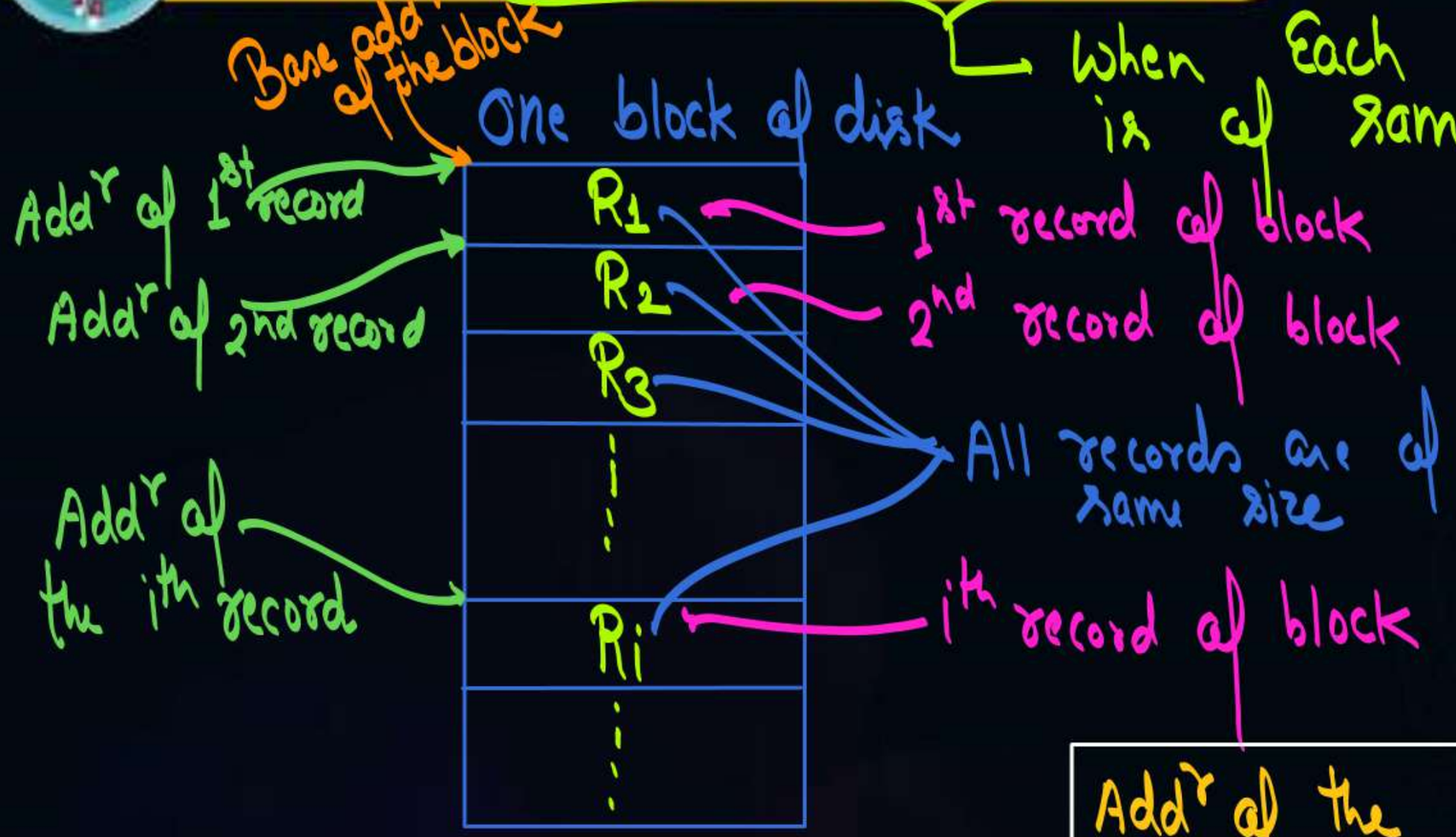
Two types of records are possible

(1) Fixed length records

(2) Variable length records.

Topic : Fixed length records

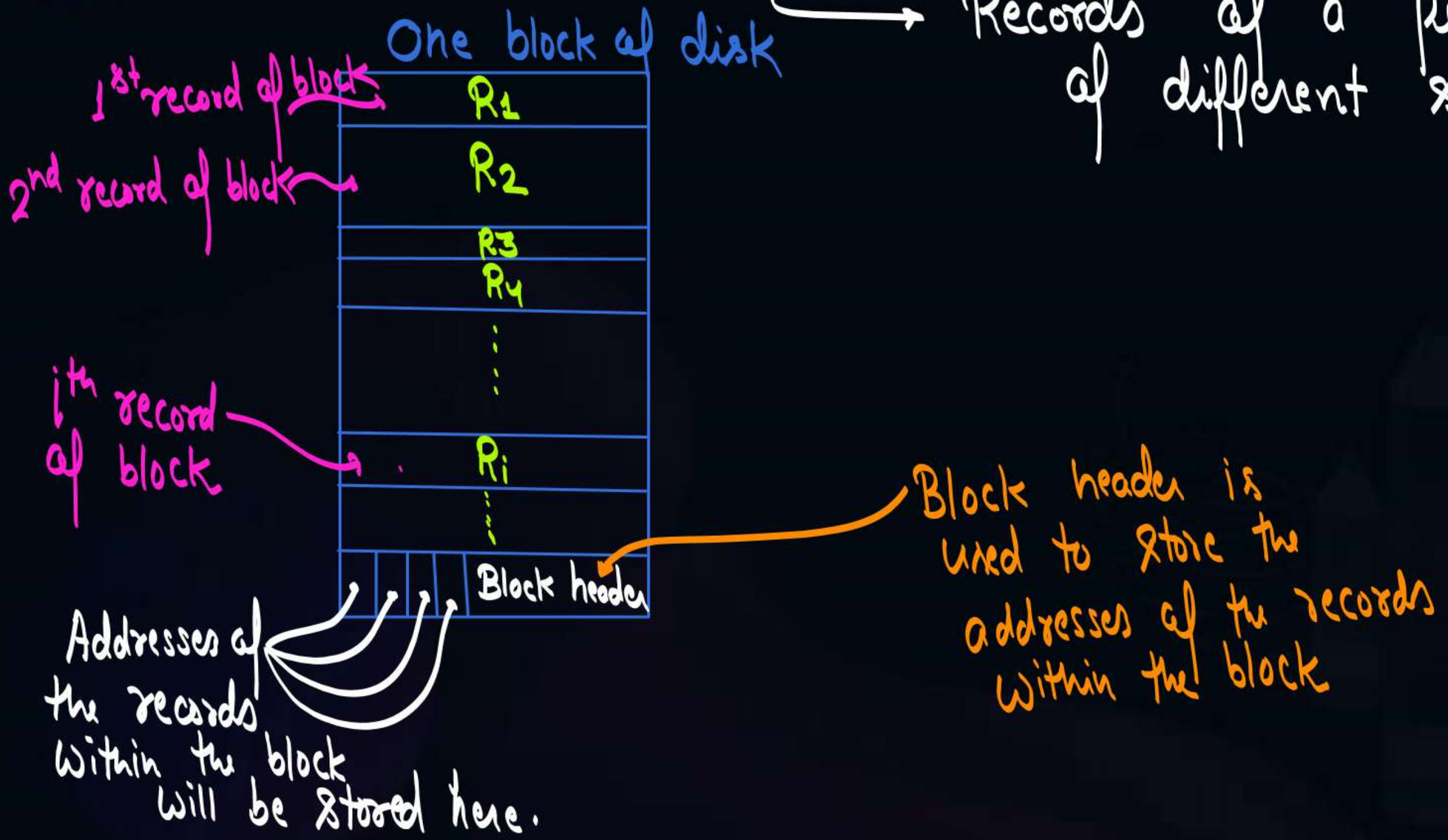
When Each record of the file is of same size



Addr of the i^{th} record of a block = Base addr of the block + $(i-1)$ * Record size

Topic : Variable length records

→ Records of a file may be of different sizes.



Block heads may be required with fixed length records as well, in order to store the address of the next block used to store the records of the file { it will be required if "link allocation" is used to allocate the blocks of the disk }

Pointer used to provide the address of the next block of the file is called "Block anchor"

Note :- ① If nothing is specified in the question about "block header size" then we will consider it Zero (negligible) { i.e. Complete block size can be used to store the records of the file }.

② If "block header size" is specified in the question, then effective space available to store the records of the file = (Disk block size − Block header size)

Topic : Blocking Factor (Bf)

Blocking factor (Bf) of the block of the disk is defined as average number of records stored per block of disk

$$\text{Blocking factor} = \frac{\text{Block size}}{\text{Avg. record size}}$$ { when block header size is not given }

$$\text{Blocking factor} = \frac{\text{Block size} - \text{Block header size}}{\text{Avg record size}}$$ { When Block header size is given }

Topic : IO cost

IO Cost of an access can be defined as number of disk blocks that needs to be transferred from secondary memory to main memory

If no. of blocks transferred are more, then high IO Cost

If no. of blocks transferred are less, then low IO Cost

Topic : Organization of Records

There are two ways to organize records of the file in the blocks of disk

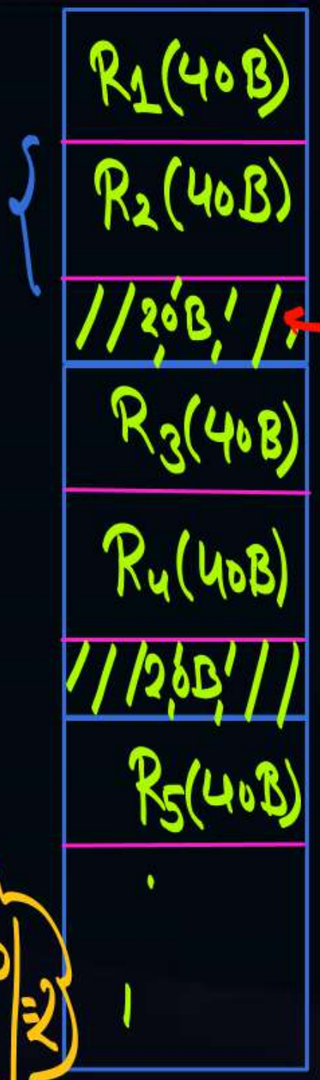
1. Unspanned Organization
2. Spanned organization

Topic : Un-spanned Organization

{ In unspanned organization a record must be stored Completely within a single block }

Let, Block size = 100 Bytes

Record size = 40 Bytes



B_1 = 100 Bytes

B_2 = 100 Bytes

B_3 = 100 Bytes

Available space is not sufficient to store a record Completely

∴ It will be wasted.

↓

it is called internal fragmentation

Blocking factor = 2

(using un-spanned organization)

Blocking factor (using unspanned orgn) = $\left\lfloor \frac{\text{Block size} - \text{Block header size}}{\text{Record size}} \right\rfloor = \left\lfloor \frac{100-0}{40} \right\rfloor = 2$

Topic : spanned Organization

{In spanned organization a record is allowed to span in two blocks}

Let, Block size = 100 Bytes

Record size = 40 Bytes

B_1 = 100 Bytes

B_2 = 100 Bytes

Using spanned organization there will be no internal fragmentation

Blocking factor = 2.5
(using spanned organization)

$$\text{Blocking factor (using spanned orgn)} = \frac{\text{Block size} - \text{Block header size}}{\text{Record size}} = \frac{100 - 0}{40} = 2.5$$

Unspanned org^n vs Spanned org^n

Unspanned org^n	Spanned org^n
① Internal fragmentation	① No internal fragmentation
② Less IO Cost { one address is found, to access any record we need to transfer only one block from disk }	② More IO Cost { once address is found, in order to access some records of the file we may need to transfer two blocks from disk to M.M! }
③ Blocking factor is an integer value	③ Blocking factor may be in fraction
④ More suitable for fixed length records	④ More suitable for variable length records.

2 mins Summary

- **Topic** Wait-die protocol for deadlock avoidance
- **Topic** Wound-wait protocol for deadlock avoidance
- **Topic** Database, File and Records
- **Topic** Organization of Records
- **Topic** Categories of index

THANK - YOU