

Computer Science & IT

Database Management System



File organization and indexing

Lecture No. 01



By- Vishal Sir

Topics to be Covered



- ✓ Topic Database, Files and records
- ✓ Topic Organization of records
- ✓ Topic Blocking factor
- ✓ Topic IO cose without index file
- Topic





Topic : Database-File-Records

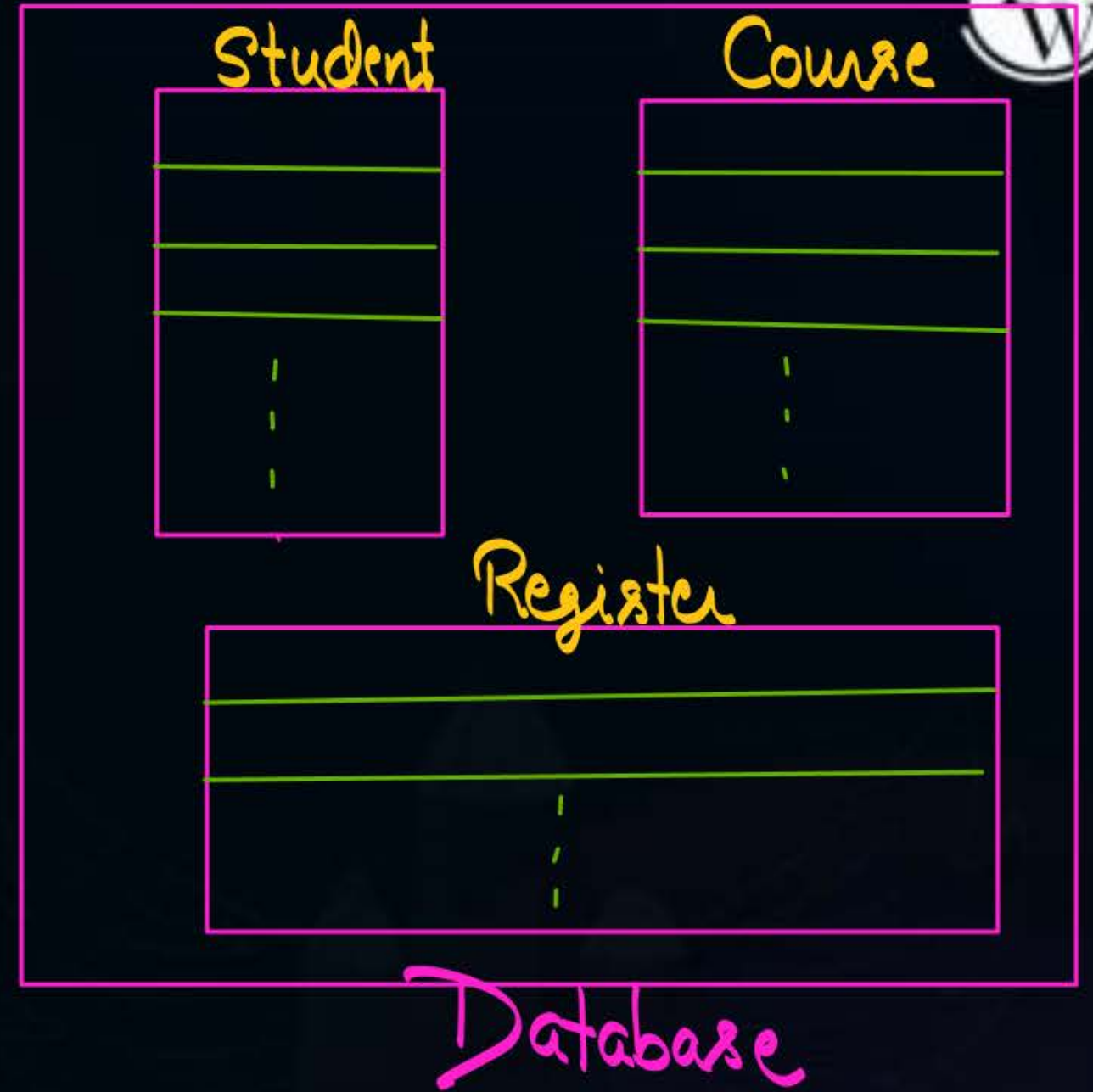


- ✓ * Database is a collection of files
- ✓ * Files are used to store the records.

* Disk blocks are used to store the records of the file

One disk block can be used to store the records of a single file.

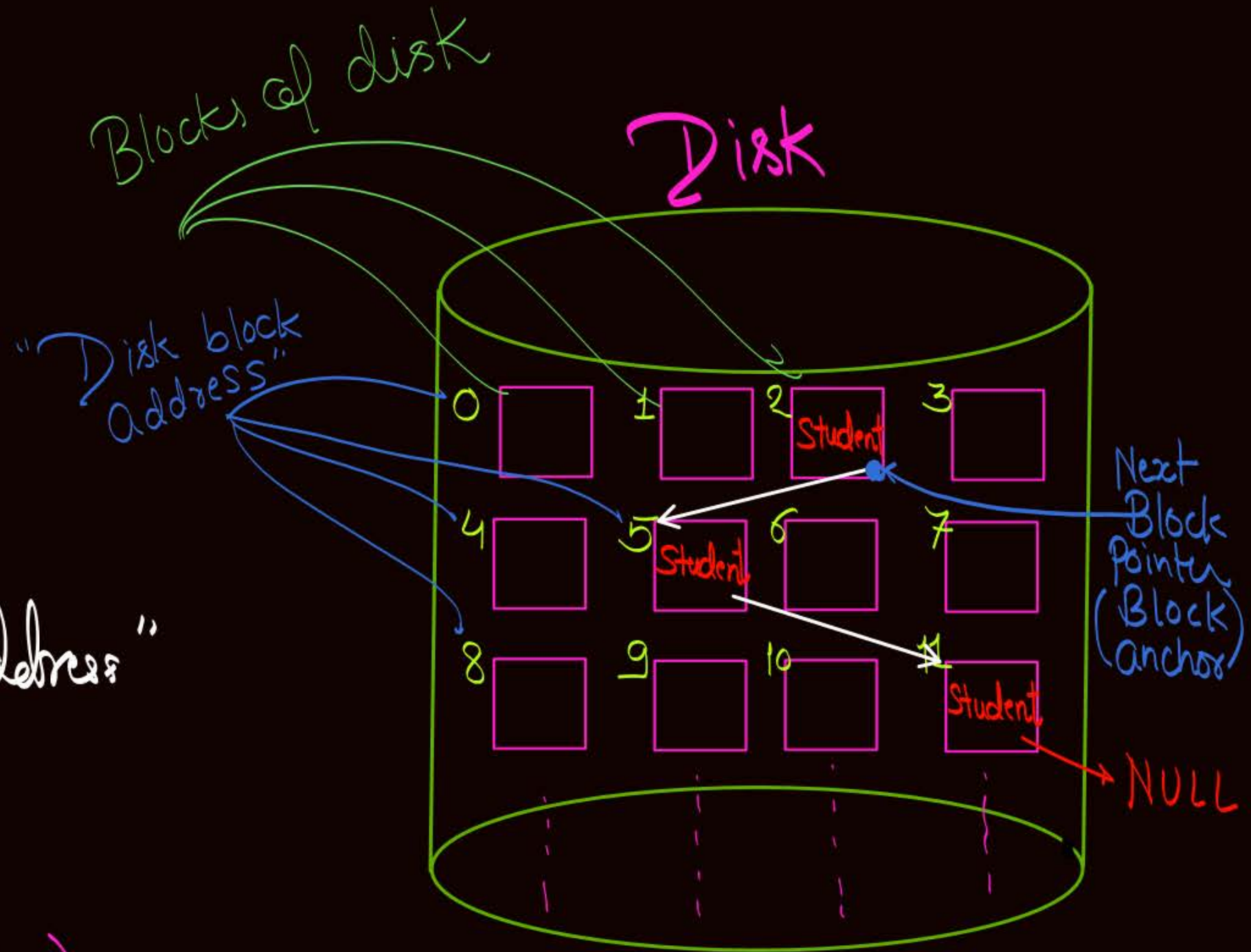
↳ Records of different files can not be stored in the same disk block



⊛ Each disk block within the disk will be of same size

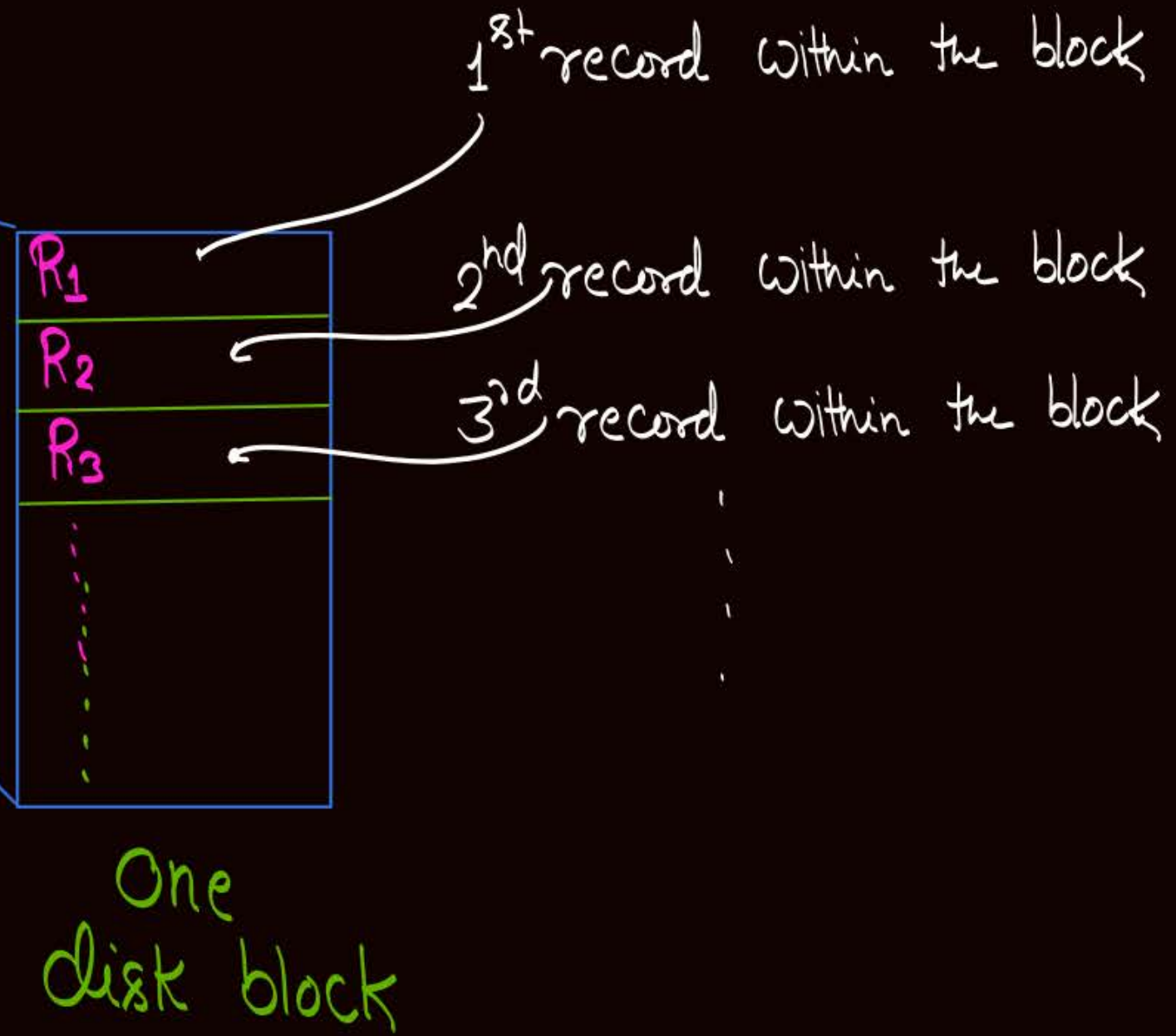
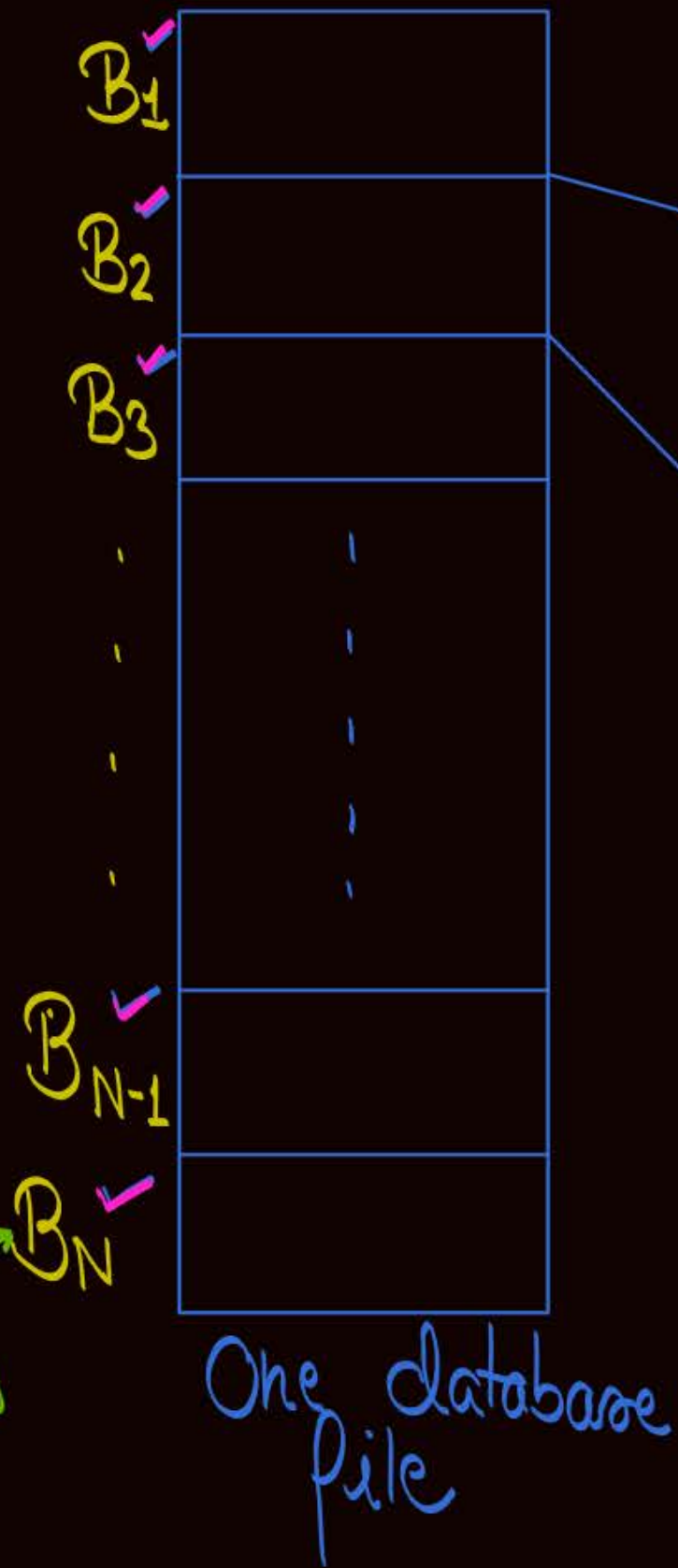
→ Each block of the disk is identified by a unique number known as "disk block address"

→ Each disk block can store multiple records of the same file (not from different files)



Multiple disk blocks may be required to store the records of a file

Let 'N' disk blocks are required to store this file





Topic : Types of Records

→ Two types of records are possible in file system

① Fixed length records

② Variable length records.

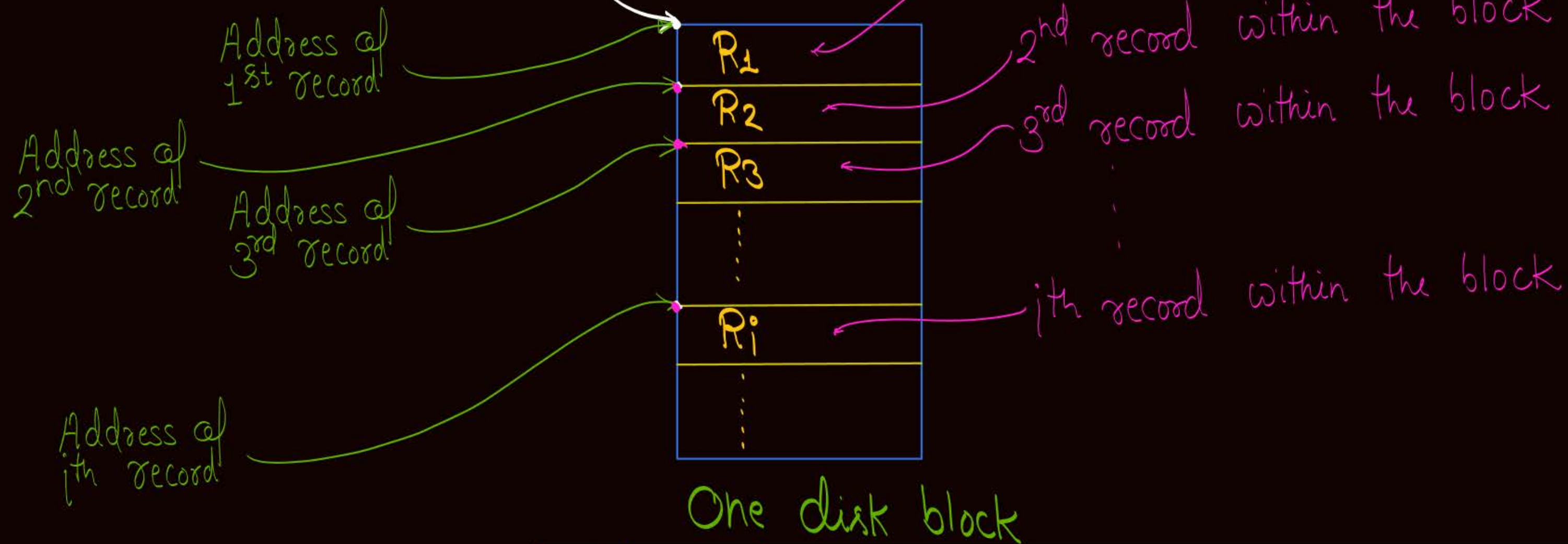


Topic : Fixed length records

When each record of the file is
of same size

✓ Fixed length Records: -

Base add^r of disk block
(starting address)



$$\text{Address of the } i^{\text{th}} \text{ record within the block} = \text{Base add}^r \text{ of block} + (i-1) * \text{Record size}$$

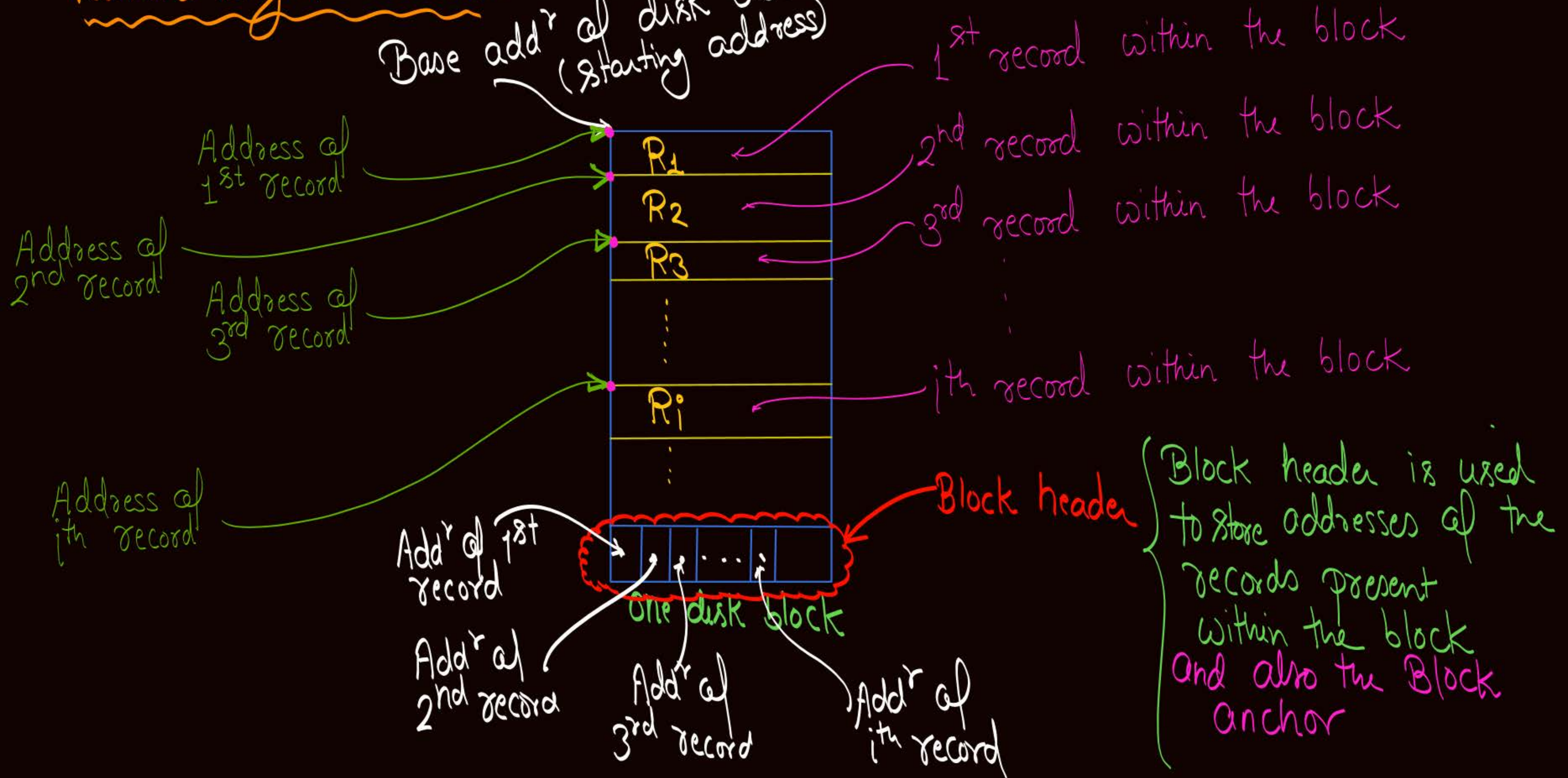


Topic : Variable length records

- * When records of the same file may be of different sizes.

Variable length Records: -

Base add^r of disk block
(starting address)



"Block anchor" :- Block anchor gives the address of the next block of the disk used to store the records of the file.

* Note:- Block header may be required with fixed length records as well, in order to store the address of the next disk block of the file {i.e.; to store block anchor}

Note: →

① In the questions, if nothing is given about block header size and/or block anchor size then consider it as "Zero", i.e., the complete disk block size will be used to store the records of the file.

② In the question, If block header size is given, then Effective space available in the disk block to store the records of the file will be
$$= (\text{Disk block size} - \text{Block header size})$$



Topic : Blocking Factor

'Bf'



- Blocking factor of database block is defined as no. of records that can be stored in one disk block.

$$\text{Blocking factor} = \left(\frac{\text{Disk block size}}{\text{Record size}} \right) \quad \text{When block header size is not given}$$

$$\text{Blocking factor} = \left(\frac{\text{Disk block size} - \text{Block header size}}{\text{Record size}} \right) \quad \text{When Block-header size is given}$$



Topic : IO cost



- * IO cost of a record access can be defined as number of disk blocks that needs to be transferred from secondary memory to main memory in order to access that record



Topic : Organization of Records

There are two ways in which records of the file can be organized within the disk block

- ✓ ① Unspanned organization → { Every record must be stored completely within the same block }
- ✓ ② Spanned organization → { Some of the records may span in two blocks }



Topic : Un-spanned Organization

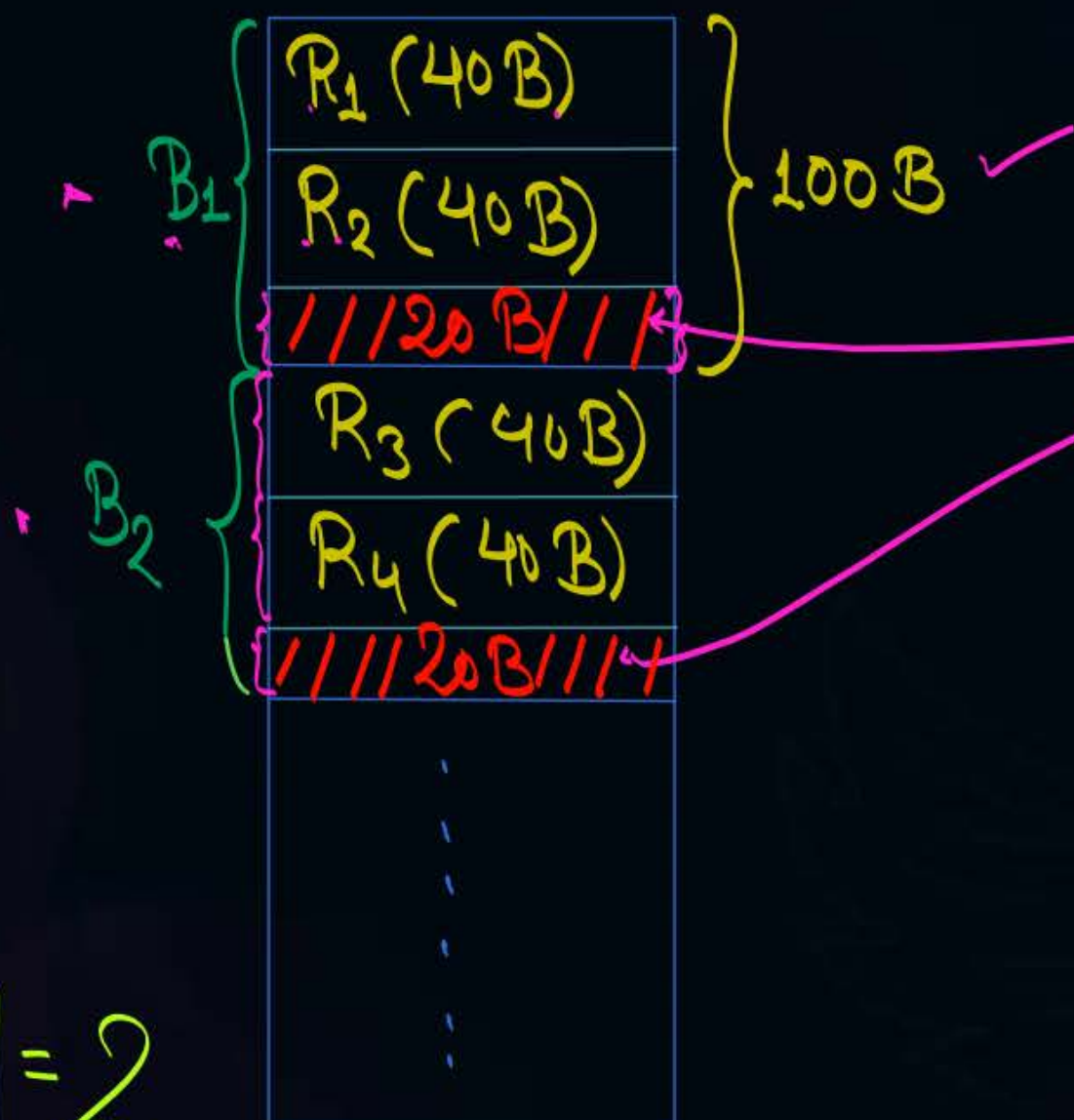
Consider,

✓ Disk block size = 100 Bytes

✓ Record size = 40 Bytes

$$\text{Blocking factor} = \left\lfloor \frac{\text{Block size}}{\text{Record size}} \right\rfloor$$

$$\text{Blocking factor} = \left\lfloor \frac{100}{40} \right\rfloor = \lfloor 2.5 \rfloor = 2$$



Available space is not sufficient to store a record completely, therefore this space is wasted and it is called "internal fragmentation".



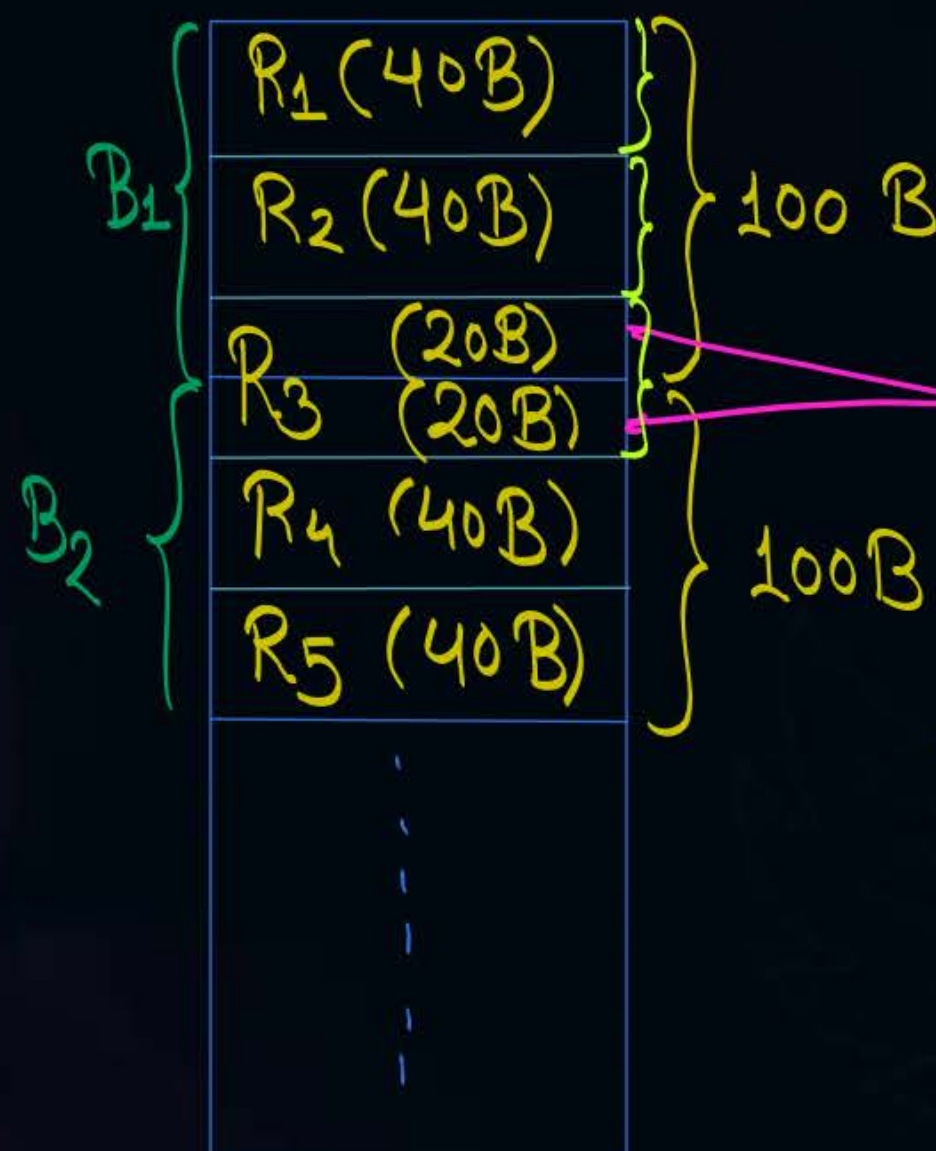
Topic : spanned Organization

Consider,

- ✓ Disk block size = 100 Bytes
- ✓ Record size = 40 Bytes

$$\text{Blocking factor} = \frac{\text{Block size}}{\text{Record size}}$$

$$\text{Blocking factor} = \frac{100}{40} = 2.5$$



R₃ is spanned
in block B₁ & B₂

* Using Spanned organization
there will be no
internal fragmentation { except
may be in the last block of file }



Topic : Search-key & Type of File

* Search key :- Attribute/field used to search for a record in a database file is called search key

* it need not
Contain Unique
Values

$Eid \neq E200$

Retrieve Record of the Employee with " $Eid = E_6$ " ✓

↓
Search opⁿ will be performed based on attribute = " Eid "

↓
∴ Search key is " Eid "

Employee database file

<u>Eid</u>	--other attributes--	<u>$Dept-id$</u>
E_3		2
E_1		1
E_4		2
E_6		3
E_2		4
E_5		1
⋮	⋮	⋮

} B_1 ✓

} B_2 ✓

} B_3 ✓

⋮

Retrieve records of all the Employee with $dept-id = 2$

↓
Search opⁿ will be performed w.r.t. " $dept-id$ "

↓
Search key is " $dept-id$ "



Topic : Search-key & Type of File

Search key :- Attribute/field used to search for a record in a database file is called search key

- ★ Types of file :- There are two types of files
- (1) Unordered file :- If records of the file are not Physically ordered (i.e. not sorted) based on values of search key attribute, then file is called unordered file w.r.t. that attribute
(Heap file)
 - (2) Ordered file :- If records of the file are Physically ordered (sorted) based on the values of search key attribute, then file is called ordered file w.r.t. that attribute

Retrieve Record
of the Employee
with "Eid = E₆"

⇓
Search for records
will be performed
based on attribute "Eid"

⇓
∴ Eid will be the
Search Key

⇓
Records of the file are not
ordered w.r.t. attribute "Eid"
∴ W.r.t. "Eid" as Search Key,
file is unordered

Employee database file

Eid	--other attributes--	Dept-id	
E ₃		2	} B ₁
E ₁		1	
E ₄		2	} B ₂
E ₆		3	
E ₂		4	} B ₃
E ₅		1	
⋮	⋮	⋮	

Retrieve records of
all the Employee
with dept-id = 2

⇓
Search for records
will be performed
based on attribute "dept-id"

⇓
∴ "dept-id" will be
the Search Key

⇓
Records of the file are not
ordered w.r.t. "dept-id" ✓
∴ W.r.t. "dept-id" as Search Key
file is unordered

Retrieve Record
of the Employee
with "Eid = E₆"

⇓
Search for records
will be performed
based on attribute "Eid"

⇓
∴ Eid will be the
Search Key

⇓
Records of the file are
Ordered w.r.t. attribute "Eid"
∴ W.r.t. "Eid" as Search Key,
file is ordered

Employee database file

Eid	--other attributes--	Dept-id	
E ₁		1	} B ₁
E ₂		4	
E ₃		2	} B ₂
E ₄		2	
E ₅		1	} B ₃
E ₆		3	
	⋮	⋮	

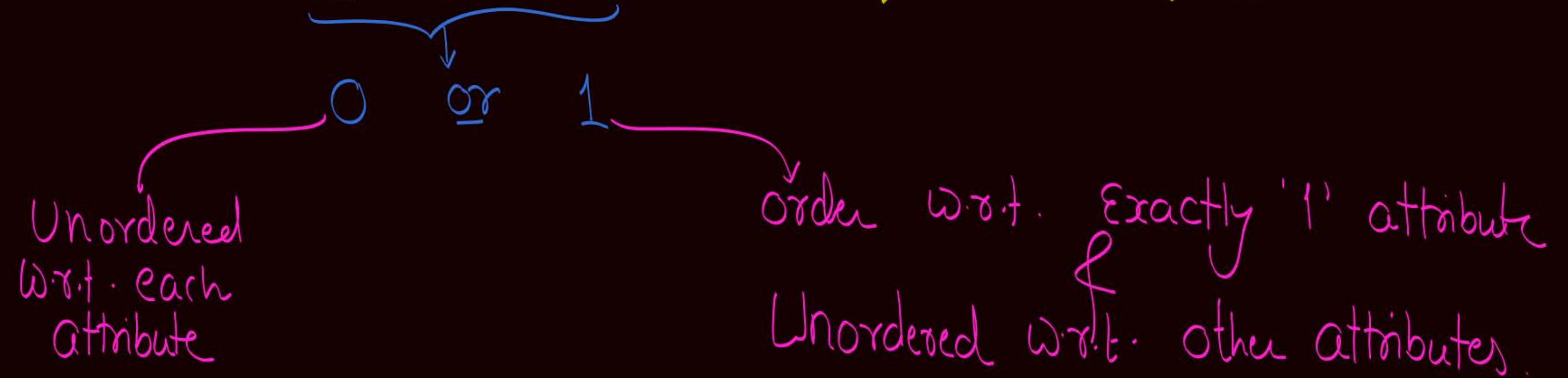
Retrieve records of
all the Employee
with dept-id = 2

⇓
Search for records
will be performed
based on attribute "dept-id"

⇓
∴ "dept-id" will be
the Search Key

⇓
Records of the file are not
ordered w.r.t. "dept-id" ✓
∴ W.r.t. "dept-id" as Search Key
file is unordered

Note: → A database file can be ordered w.r.t.
at most one attribute, at a time.





Topic : IO cost without indexing

→ Let " N " is the total number of disk blocks required to store the given database file,

(i) Worst case IO cost = N { Search will be linear search }
(When file is un-ordered)

(ii) Worst Case IO Cost = $\lceil \log_2 N \rceil$ { When file is ordered }
(When file is ordered) { We can perform binary search }

→ If database file is too large, then ' N ' value will also be a large value and in that case $\lceil \log_2 N \rceil$ will also be a significant value.
∴ In order to reduce this IO cost concept of Indexing is used.

Ordered file

Vs

Unorder file
{ Also known as Heap file }

① Binary search is possible

① Binary search is not possible

② Insertion of a new record in an ordered file will be a costly operation

{ Because record must be stored at its appropriate position in the file }

② Insertion of a new record will be easy
{ we can simply insert the new record in the last block of the file }



2 mins Summary



Topic

Database, Files and records

Topic

Organization of records

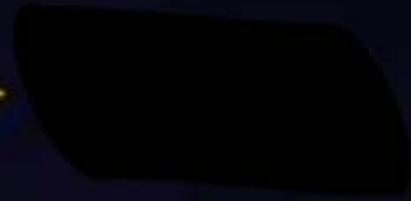
Topic

Blocking factor

Topic

IO cose without index file

Topic



THANK - YOU