

# 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

P  
W

✓ \* 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.

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

Student

:

Course

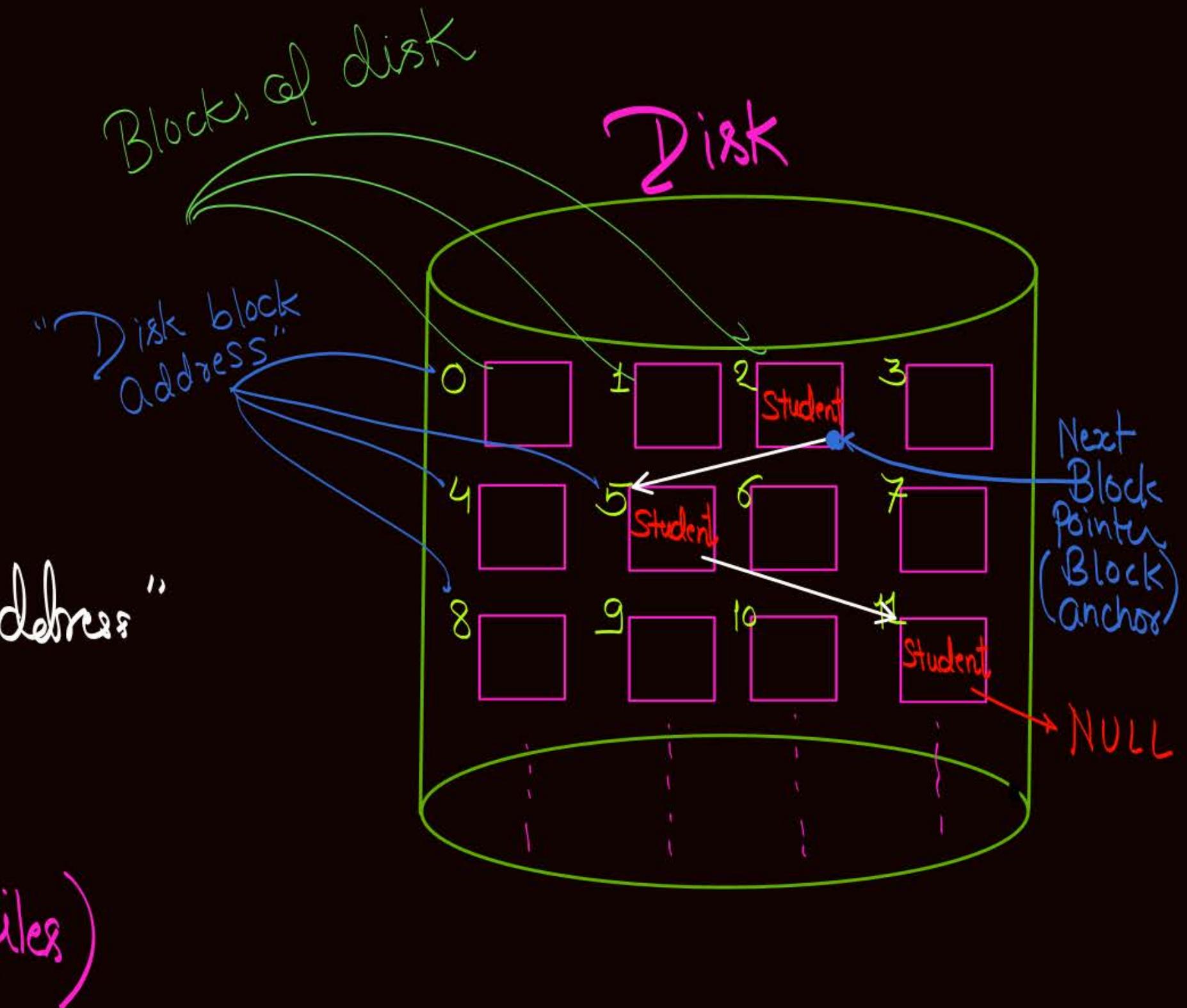
:

Registers

:

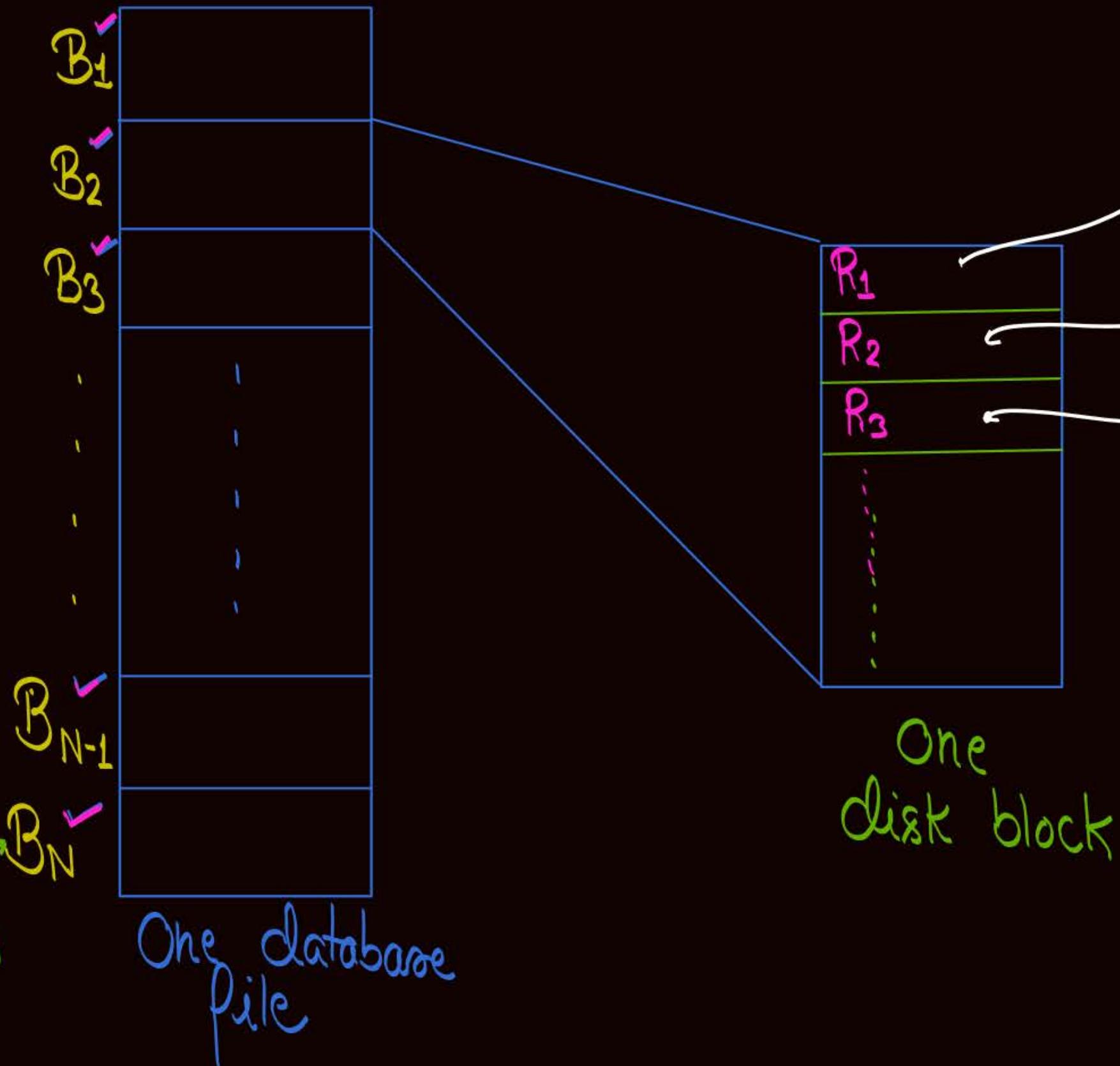
Database

- 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



1<sup>st</sup> record within the block  
2<sup>nd</sup> record within the block  
3<sup>rd</sup> record within the block

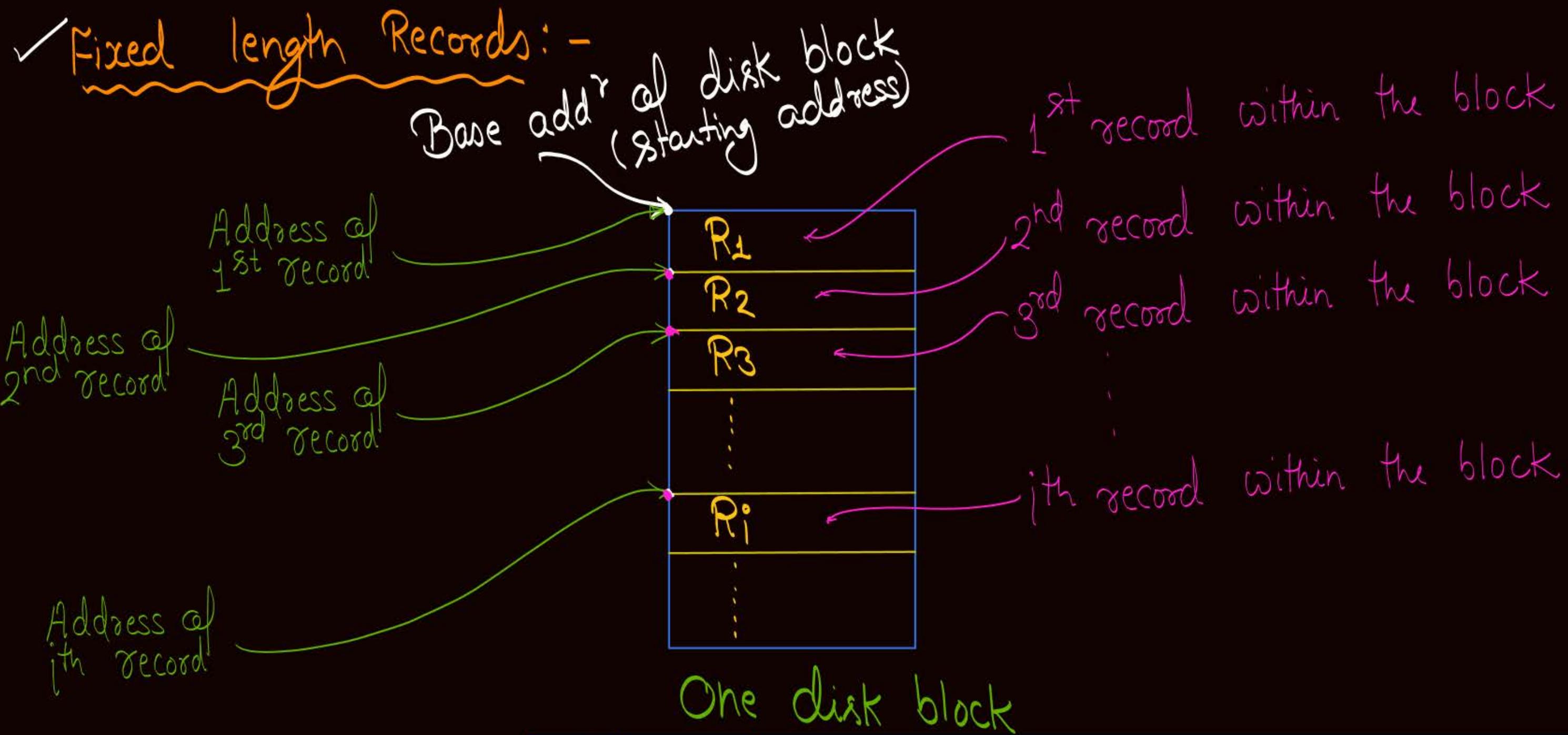


## Topic : Types of Records

- \* 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

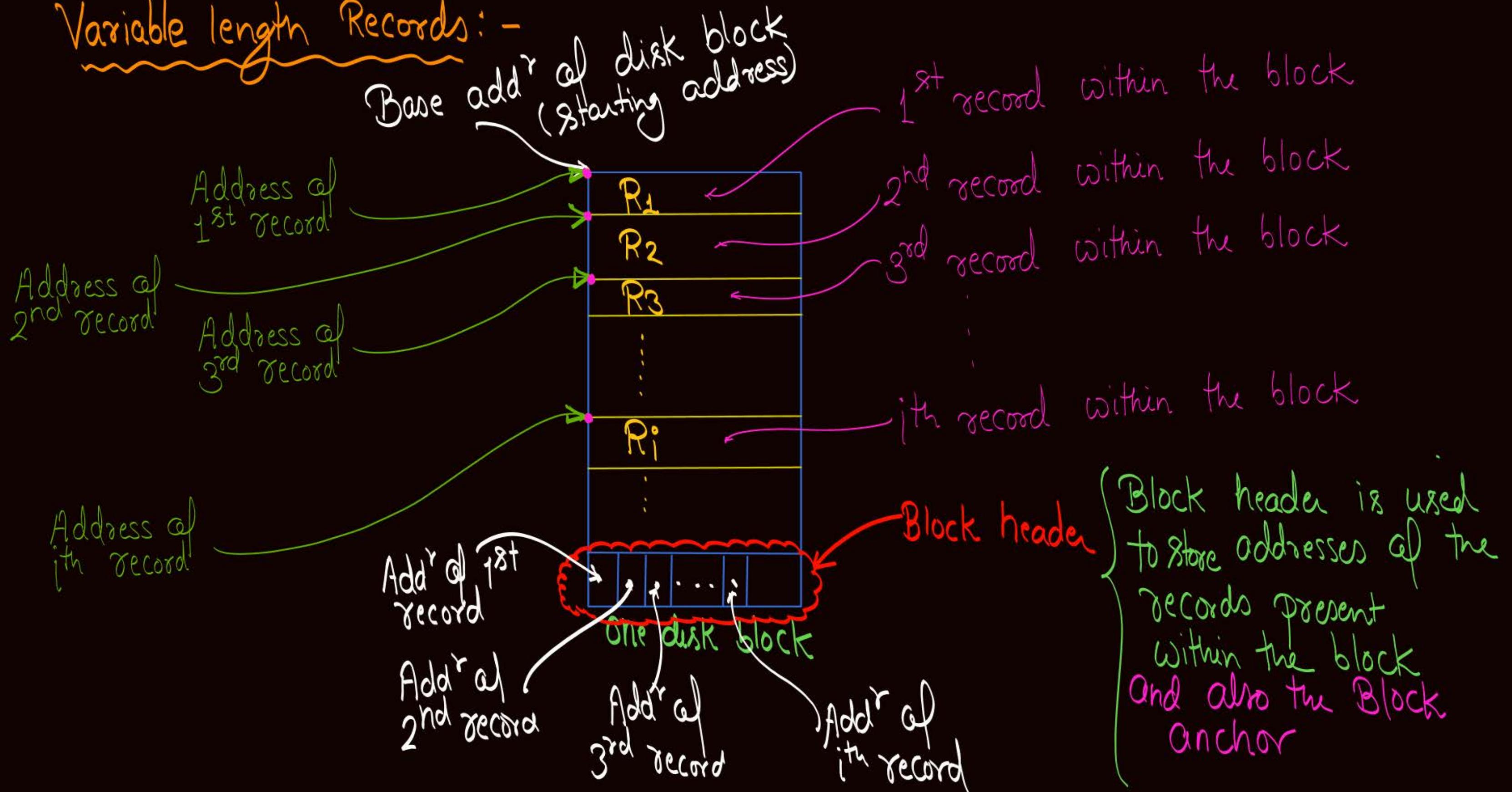


$$\text{Address of the } i\text{th record within the block} = \text{Base add}' \text{ of block} + (i-1) * \text{Record size}$$

## Topic : Variable length records

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

## Variable length Records :-



"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 & to store block anchor { }

Note →

① In the question, 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'

P  
W

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

$$\text{Blocking factor} = \left( \frac{\text{Disk block size}}{\text{Record size}} \right)$$

When block header size is not given

$$\text{Blocking Factor} = \left( \frac{\text{Disk block size} - \text{Block header size}}{\text{Record size}} \right)$$

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

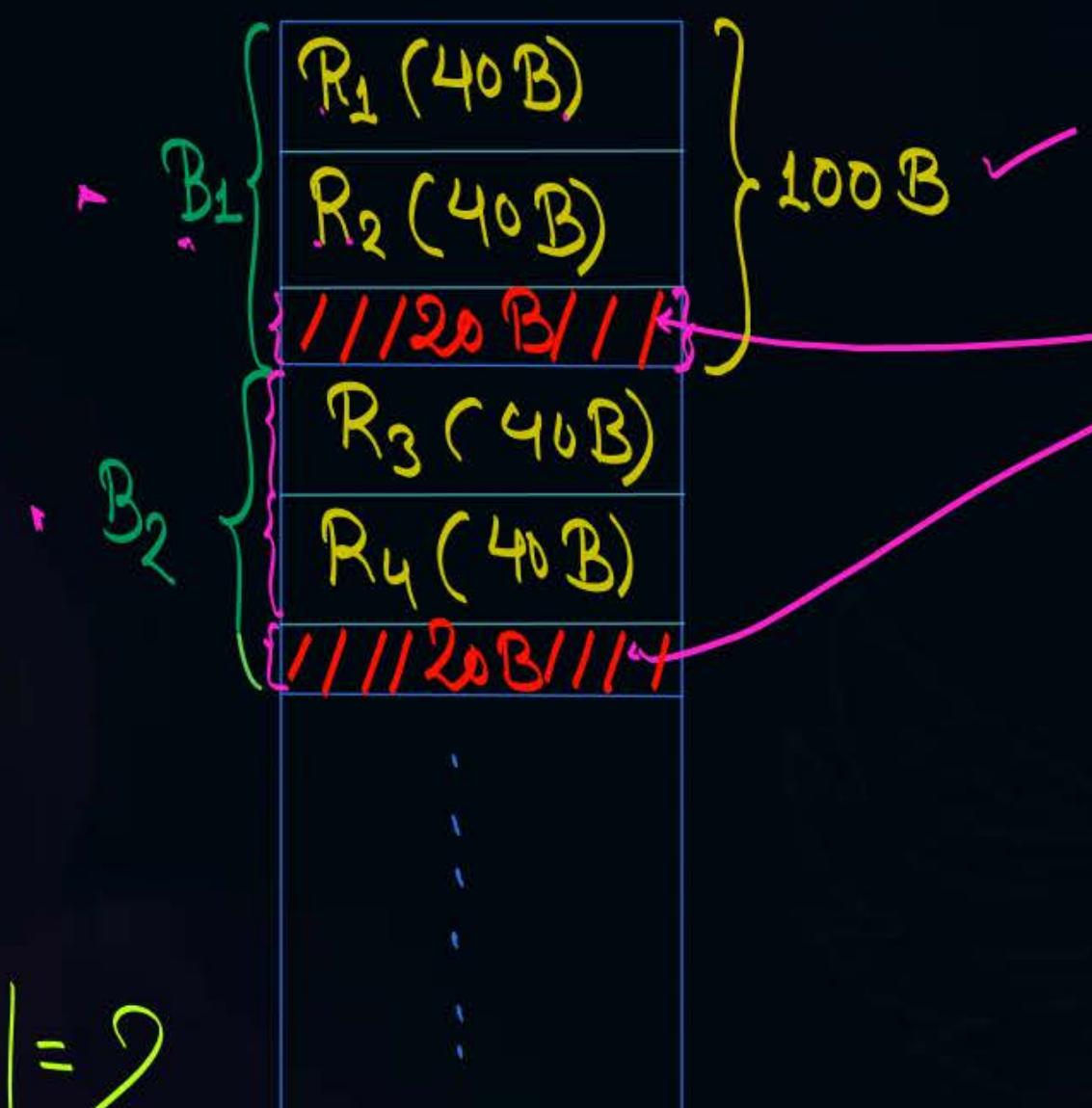
Consider,

Disk block size = 100 Bytes

Record size = 40 Bytes

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

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





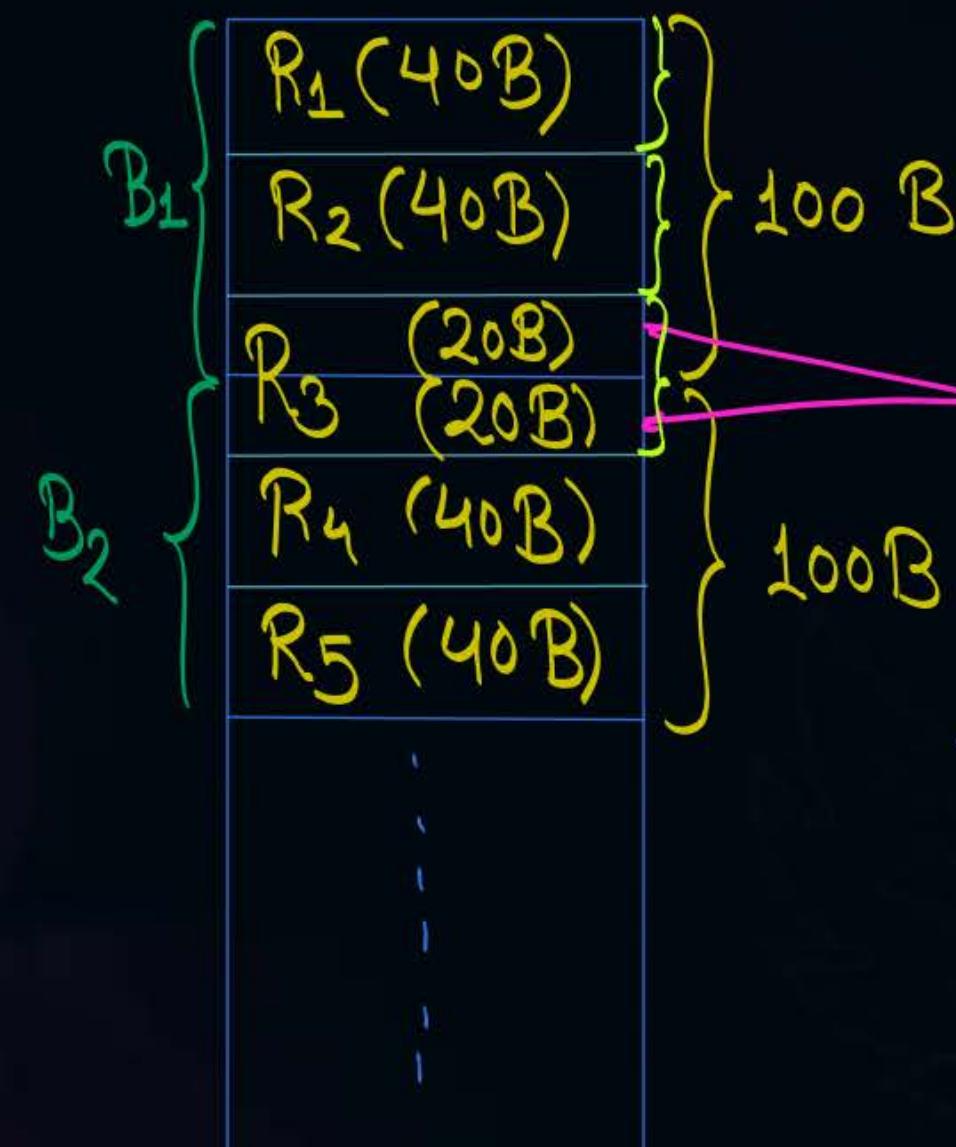
## Topic : spanned Organization

Consider,

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

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

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



R<sub>3</sub> is spanned  
in block B<sub>1</sub> & B<sub>2</sub>

- \* 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 > E200

- Retrieve Record of the Employee with "Eid = E6"

Search opn will be performed based on attribute : "Eid"

i) Search key is "Eid"

Employee database file		
Eid	- - other attributes - -	Dept-id
E3		2
E1		1
E4		2
E6		3
E2		4
E5		1
:	:	:

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
  - (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 = E<sub>6</sub>"

↓  
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 <sub>3</sub>		2
E <sub>1</sub>		1
E <sub>4</sub>		2
E <sub>6</sub>		3
E <sub>2</sub>		4
E <sub>5</sub>		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 = E<sub>5</sub>"

↓  
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 <sub>1</sub>		1
E <sub>2</sub>		4
E <sub>3</sub>		2
E <sub>4</sub>		2
E <sub>5</sub>		1
E <sub>6</sub>		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.

0 or 1

Unordered  
w.r.t. each  
attribute

order w.r.t. exactly '1' attribute  
Unordered w.r.t. other attributes.



## 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.  
We can perform binary search}  
(When file is ordered)

→ 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

## Unordered file

{Also known as Heap file}

① Binary search is 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 }

① Binary search is not possible

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