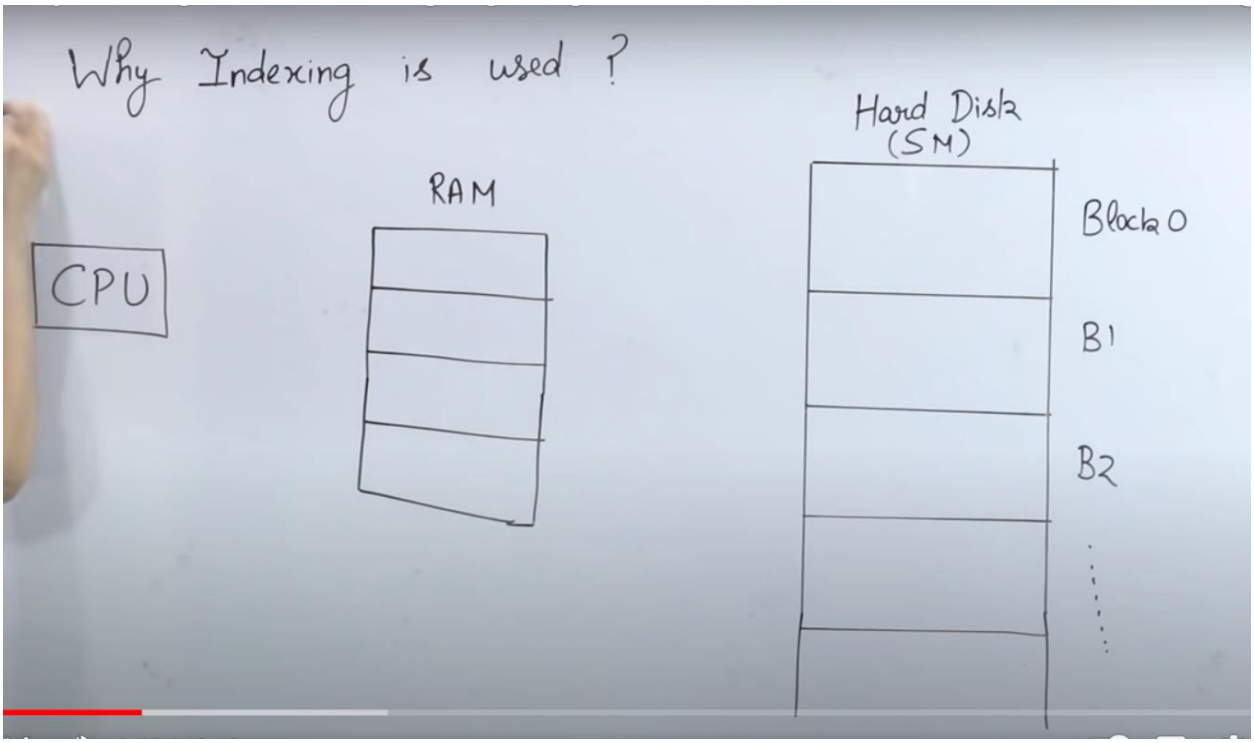
INDEXING

# GATE SMASHERS

### LEC 93: Why Indexing is used | Indexing Beginning | DBMS




CPU speed is MIPS(Million Instruction Per Second)

Hard Disk is slow, RAM is fast. Hard disk is divided into blocks and each block is loaded into RAM and CPU communicates with RAM. If CPU needs to show the result of below query:
select * from somerandomtable where somefield='value';

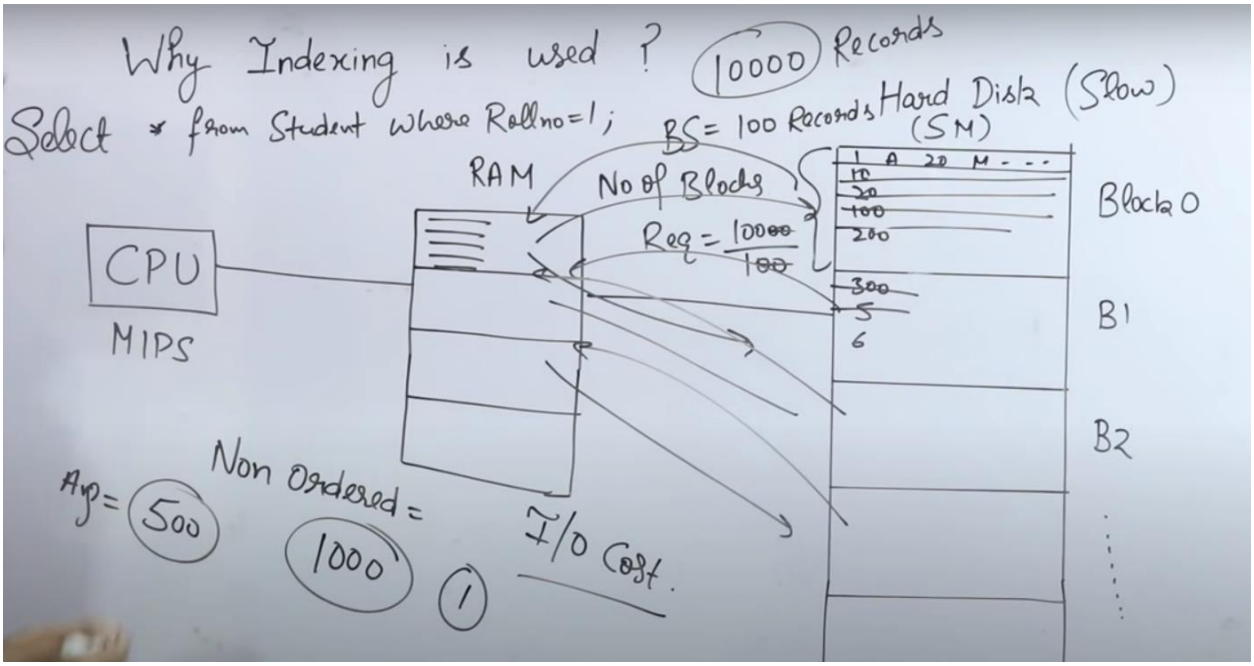
First the instruction would be given to CPU, and it would pass it to RAM and then RAM will take blocks one by one each from Hard Disk and try to find the above query results. The problem lies in the **I/O cost** which is the cost of CPU to bring each block into its own memory.

Lets suppose we have 10000 records in our somerandomtable and saved on the hard disk and block size of each block is 100 means each block can store upto 1000 records. Lets find out the number of blocks required to store 10000 records:

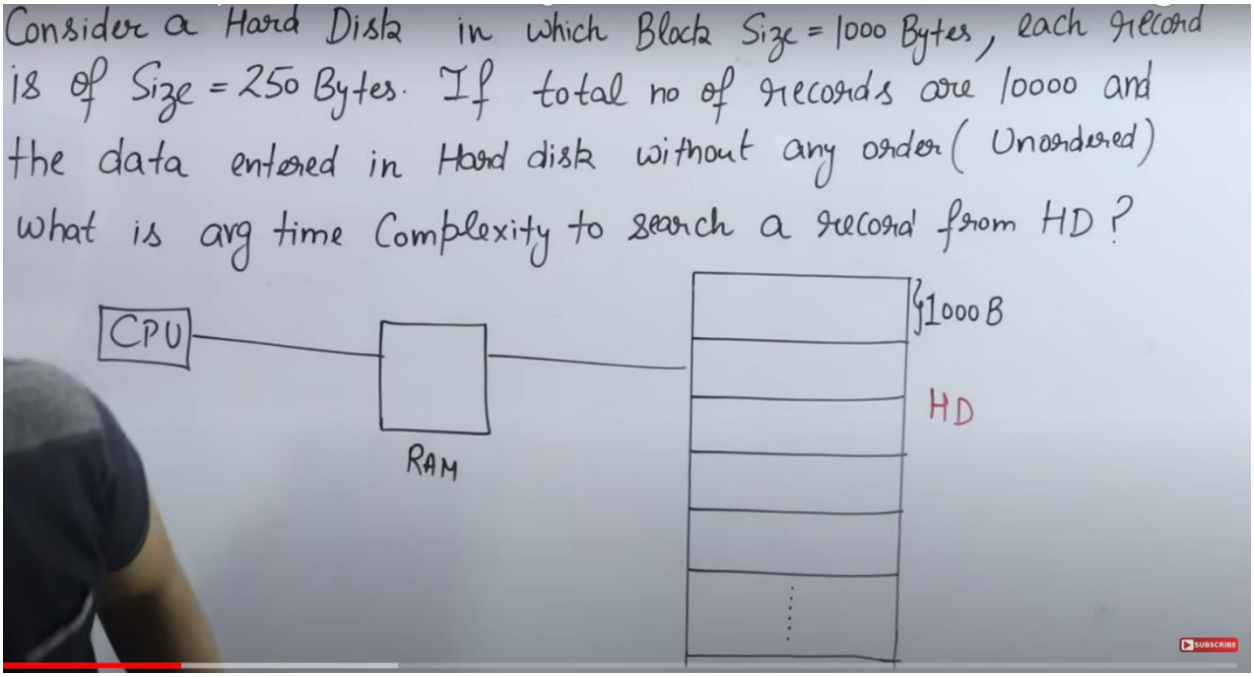
No. of Blocks = 10000/100 = 100

Now lets suppose data is in random order( **unsorted** ) in this case we need to bring 100 (total) blocks in memory one by one in worst case to search a single record.

# INDEXING




**Lec-94: Numerical Example on I/O Cost in Indexing | Part-1 | DBMS**




No. of Records we can put in each block = 1000B/250B = 4

No. of Blocks required to store the 10000 records = 10000/4 = 25000

# INDEXING

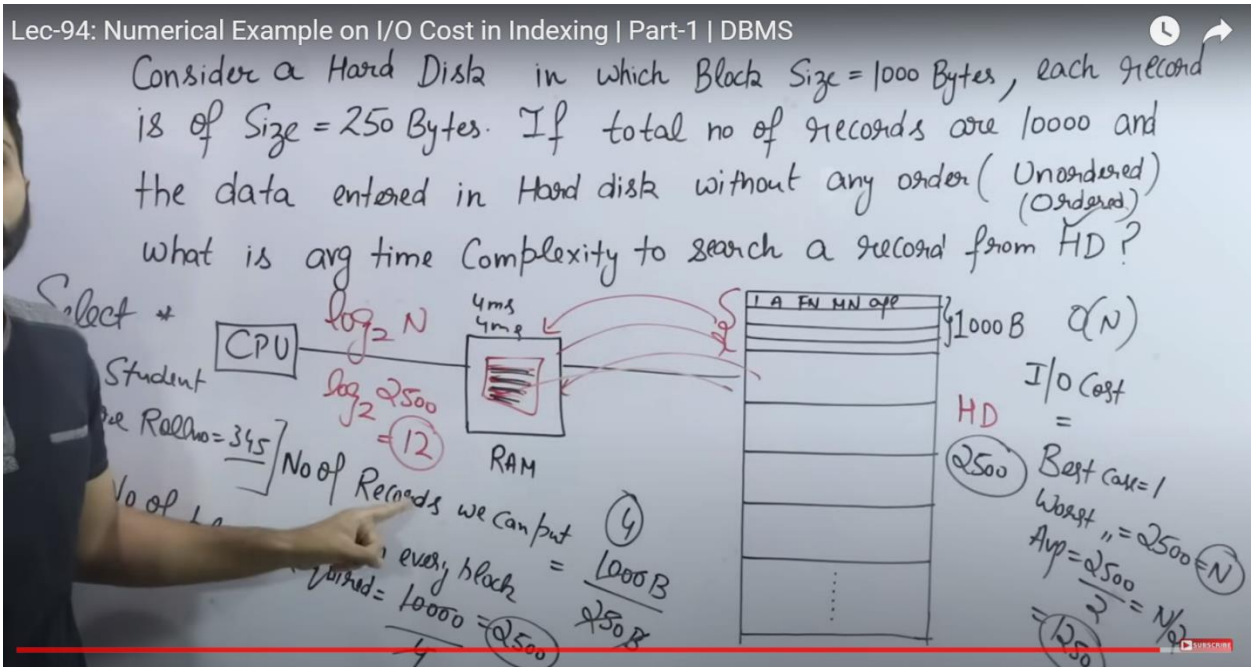
Time to search a record in hard disk blocks:

Best case = 1block means we found record in first block

Worst case = 2500 means we found out record in last block = N

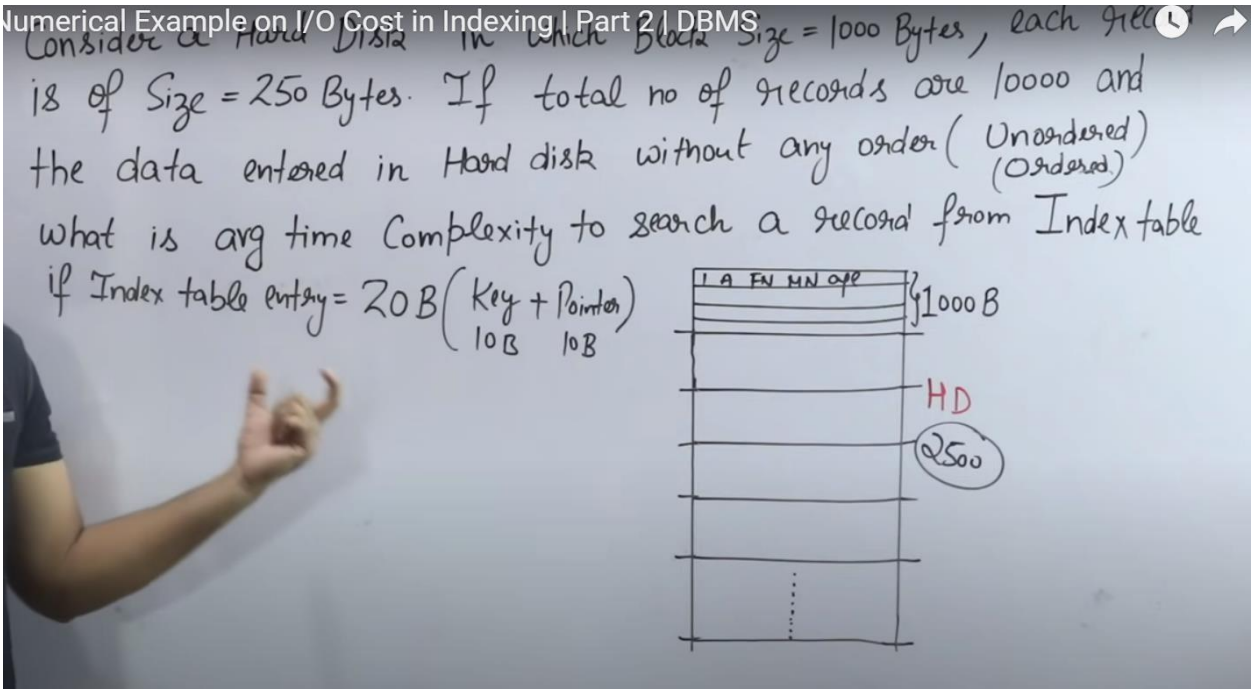
Average case = 2500/2 = 1250 = N/2

If data is sorted then lgn, in our case lg2 2500 = 12




**Lec-95: Numerical Example on I/O Cost in Indexing | Part 2 | DBMS**

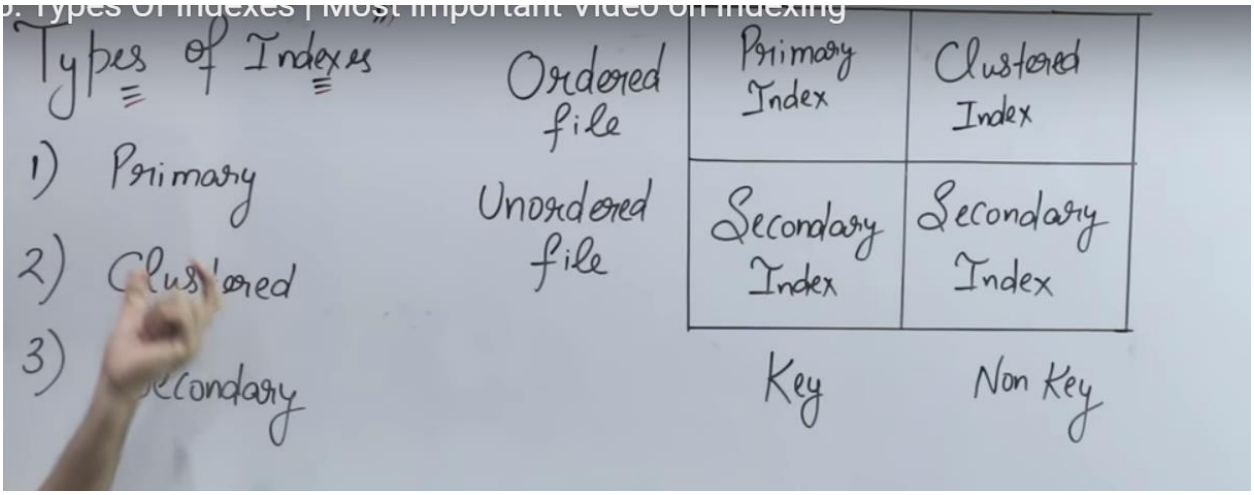
# INDEXING




Index table: a table which contains keys and pointers

**Lec-96: Types Of Indexes | Most Important Video on Indexing**

Ggfff

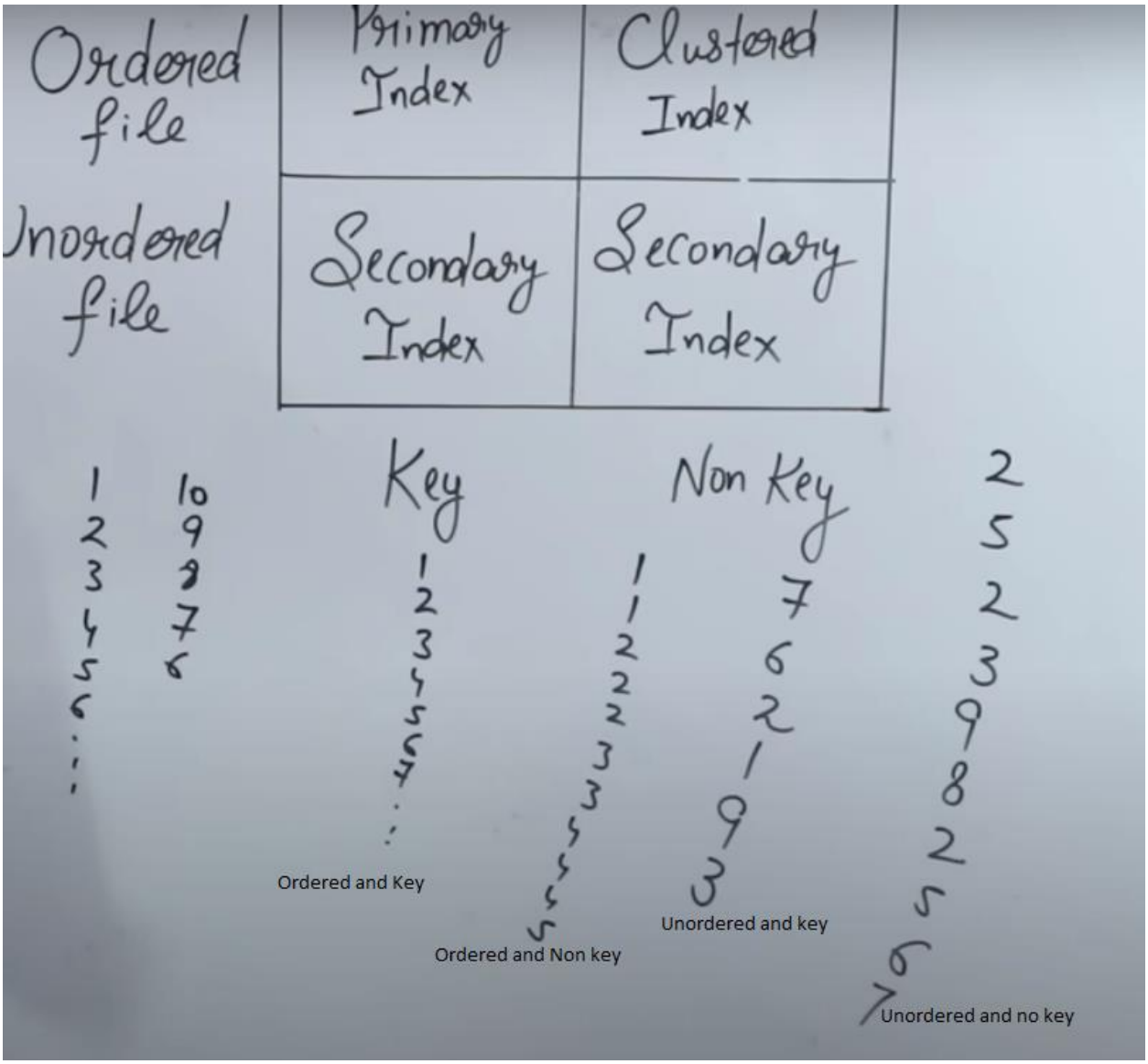



Here Key means some unqiue attribute and non key means no unique attribute.

In Oracle and sql Server if we create a primary key, primary index is automatically created.

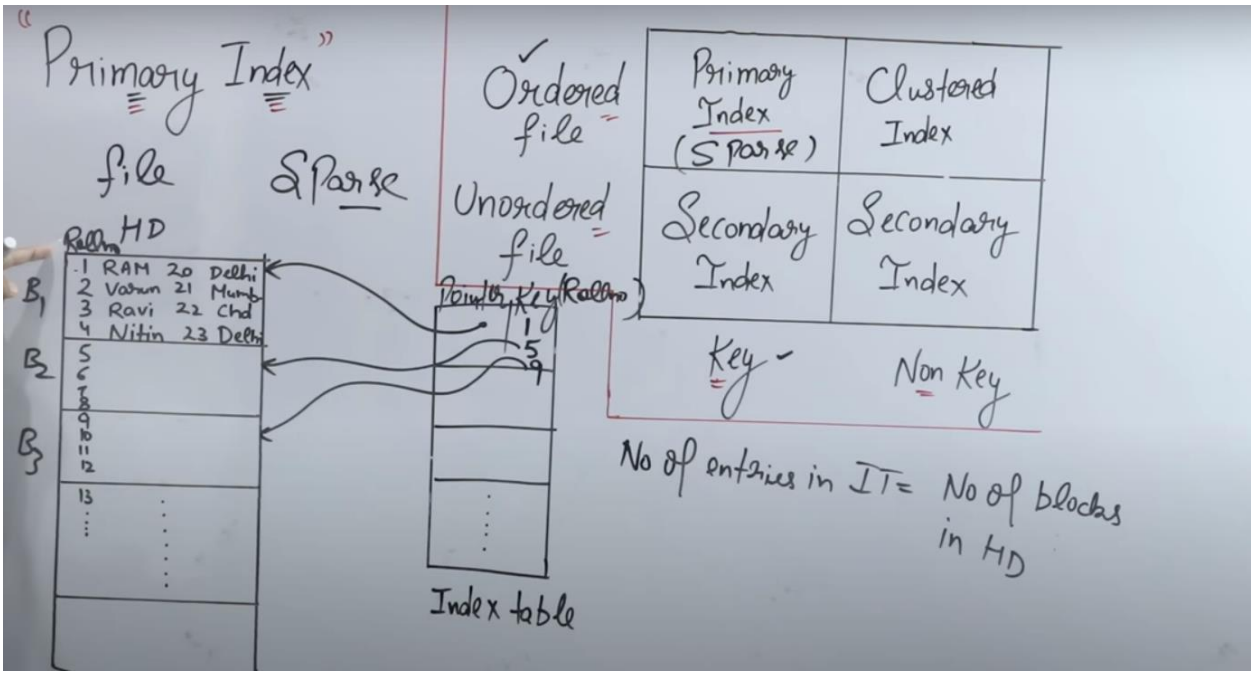
Examples

# INDEXING




**Lec-97: Primary Index With Example | GATE, PSU and UGC NET | DBMS**

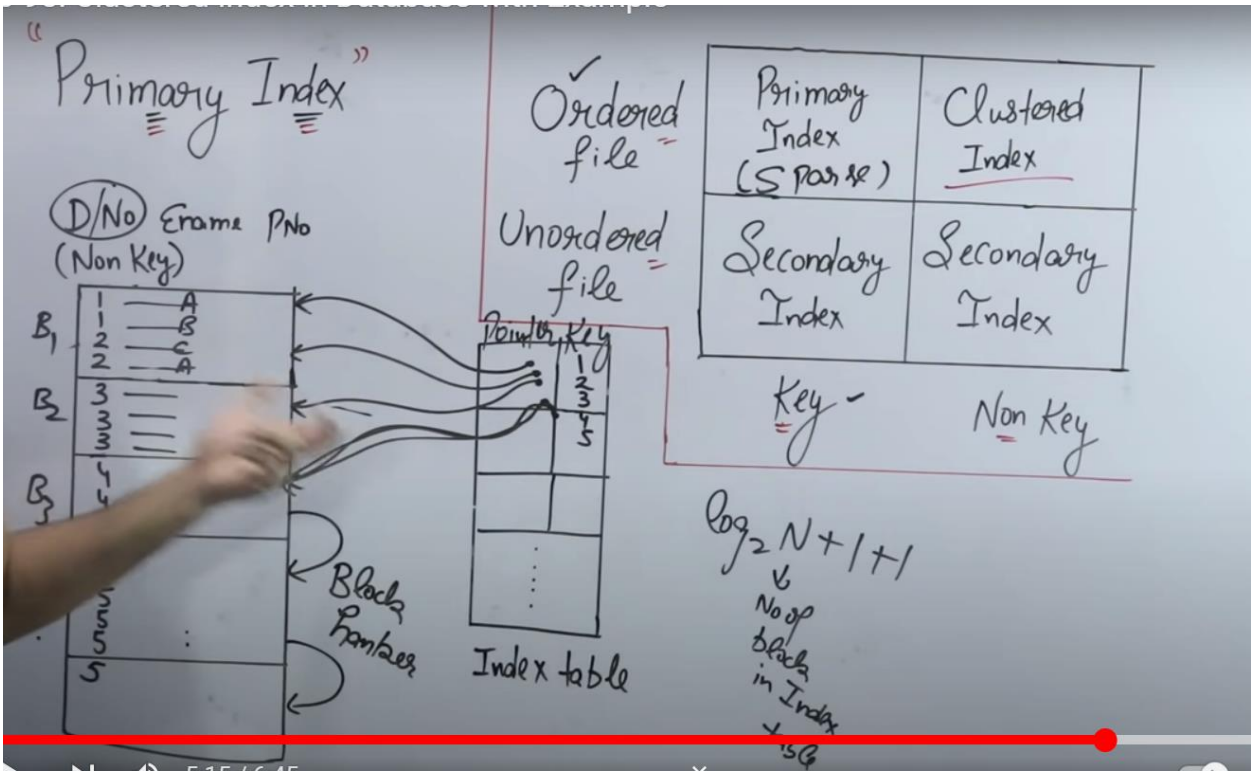
# INDEXING




For each Hard Disk block, one pointer is added in the Index table to point to that specific block.

**Lec-98: Clustered Index in Database with Example**

# INDEXING



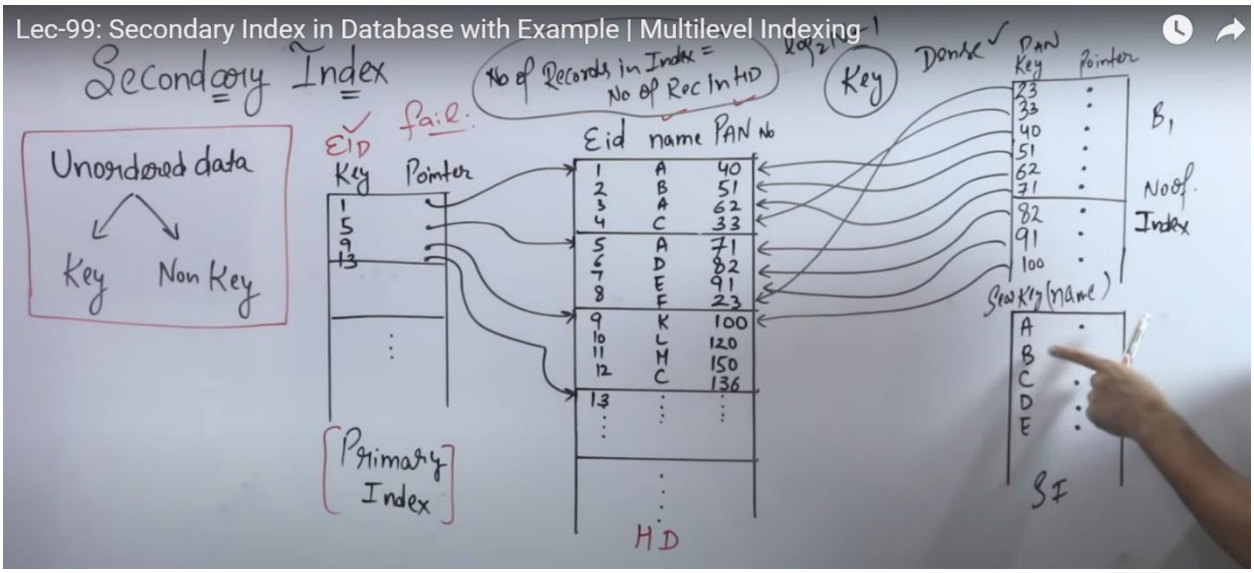

Block Hanger is used to tell that the searched value might be present in the next block.
Time complexity to search in clustered index is lg2 N + 1(int the hard disk block) + c(might be present in the next blocks as indicated by Block hanger)

**Lec-99: Secondary Index in Database with Example | Multilevel Indexing**

Secondary index is the second index we use although we do have a primary index, but when the user want to access non key values like name in any table which is usually not unique then we use the secondary index.

There are two cases, first one is we have a unique key so we created a primary index, but now we need to search the table based on another candidate key now we can create a index table of all the entries in the original table. We basically will store all the pointers of each entry in index table.

# INDEXING




Second case is when we have attribute which is non key. We will use Block of Record pointers in order to create an intermediate table to store the references of each block values in it. For pointing index table to intermediate table there would be a sparse table and for pointing intermediate table to hard disk, we need a dense table.