* Let's assume a table have small size → its index file fits in 1 HDD block

↳ Single level indexing is enough

⇓

Assume this table have 3 Million records → index file also can be stored in 1 HDD block

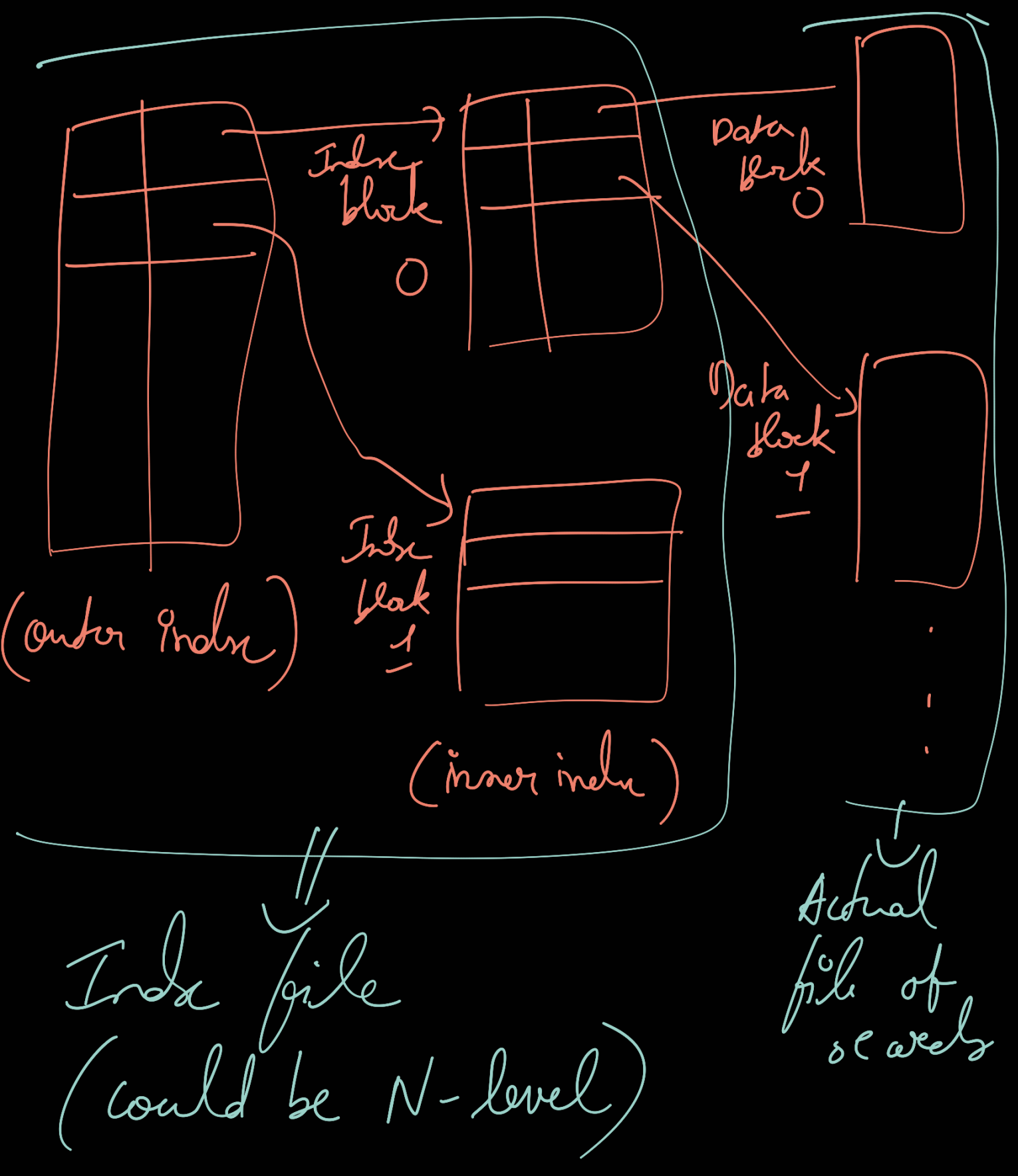
⇓

Multilevel indexing is used

⇓

Similar to Multilevel indexing of files in file system

If an index is small enough to be kept entirely in main memory, the search time to find an entry is low. However, if the index is so large that not all of it can be kept in memory, index blocks must be fetched from disk when required. (Even if an index is smaller than the main memory of a computer, main memory is also required for a number of other tasks, so it may not be possible to keep the entire index in memory.) The search for an entry in the index then requires several disk-block reads.



* If a table/collection has primary key, DBMS automatically makes Index file with that key

↳ _id in MongoDB has default index by MongoDB

* If table has no primary key or unique constraint

⇓

No index file is created

⇓

Each query does full table scan → could be less expensive for small data set but

will largely gets impacted more of large datasets like millions