

Explain B+ tree and B Tree Index files in DBMS.



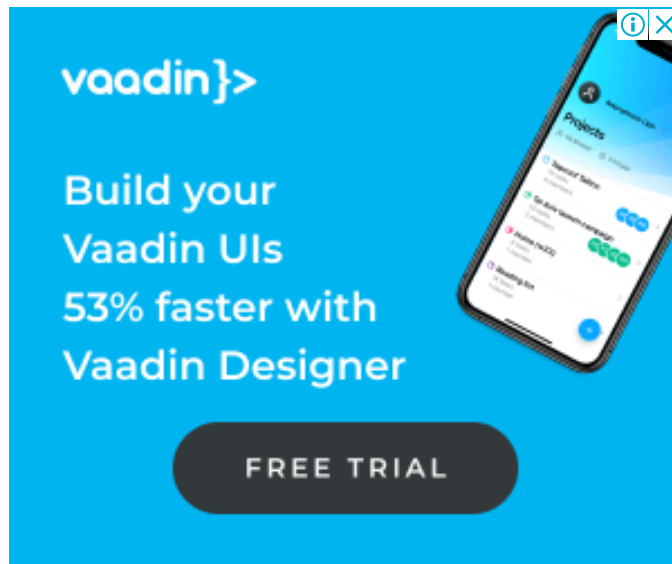
written 18 months ago by modified 7 months ago by



tanya.tanya • 130



Sayali Saw • 30



Subject: Database Management System

Topic: Storage and Indexing

Difficulty: Medium

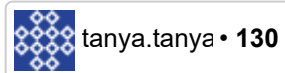
mumbai university (/t/mumbai university/) b tree (/t/b tree/) b+ tree (/t/b+ tree/) dbms(23) (/t/dbms(23)/) • 4.0k views



ADD COMMENT



written 18 months ago by



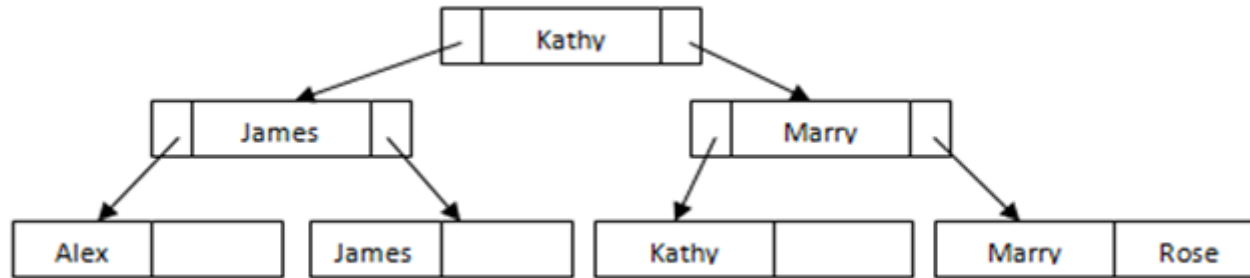
B+ tree

B+ tree is used to store the records in the secondary memory. If the records are stored using this concept, then those files are called as B+ tree index files. Since this tree is balanced and sorted, all the nodes will be at same distance and only leaf node has the actual value, makes searching for any record easy and quick in B+ tree index files.

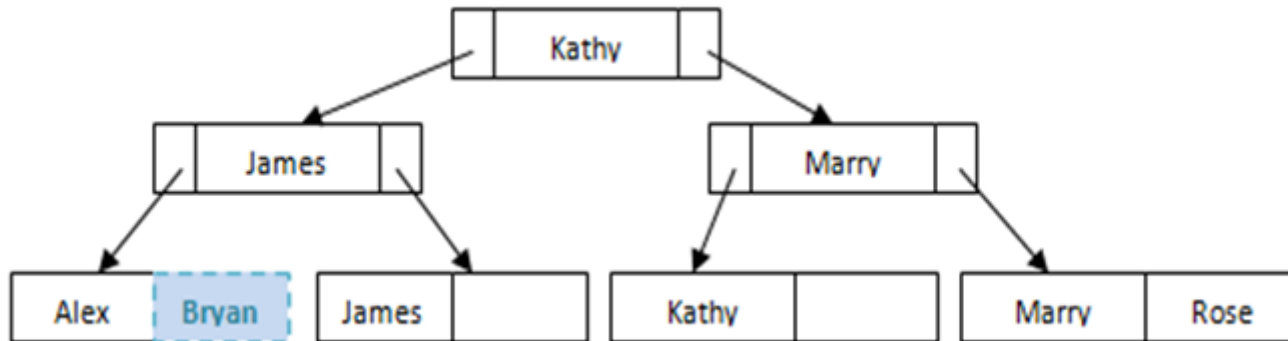
Even insertion/deletion in B+ tree does not take much time. Hence B+ tree forms an efficient method to store the records.

Searching, inserting and deleting a record is done in the same way we have seen above. Since it is a balance tree, it searches for the position of the records in the file, and then it fetches/inserts /deletes the records. In case it finds that tree will be unbalanced because of insert/delete/update, it does the proper re-arrangement of nodes so that definition of B+ tree is not changed.

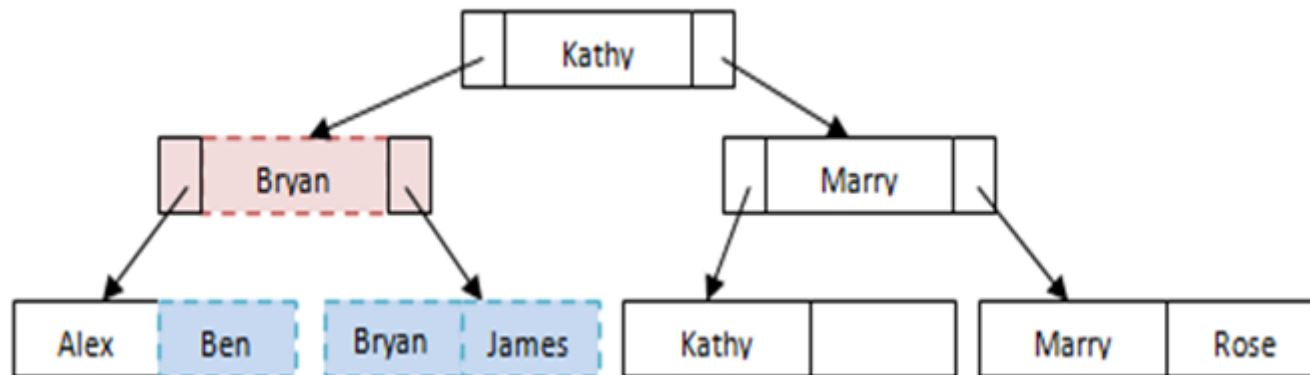
Below is the simple example of how student details are stored in B+ tree index files.



Suppose we have a new student Bryan. Where will he fit in the file? He will fit in the 1st leaf node. Since this leaf node is not full, we can easily add him in the node.



But what happens if we want to insert another student Ben to this file? Some re-arrangement to the nodes is needed to maintain the balance of the file.



Benefits of B+ Tree index files

As the file grows in the database, the performance remains the same. It does not degrade like in ISAM. This is because all the records are maintained at leaf node and all the nodes are at equi-distance from root. In addition, if there is any overflow, it automatically re-organizes the structure.

Even though insertion and deletion are little complicated, it can be done in fraction of seconds.

Leaf node allows only partial/ half filled, since records are larger than pointers.

B Tree index Files

B tree index file is similar to B+ tree index files, but it uses binary search concepts. In this method, each root will branch to only two nodes and each intermediary node will also have the data. And leaf node will have lowest level of data. However, in this method also, records will be sorted. Since all intermediary nodes also have records, it reduces the traversing till leaf node for the data. A simple B tree can be represented as below:

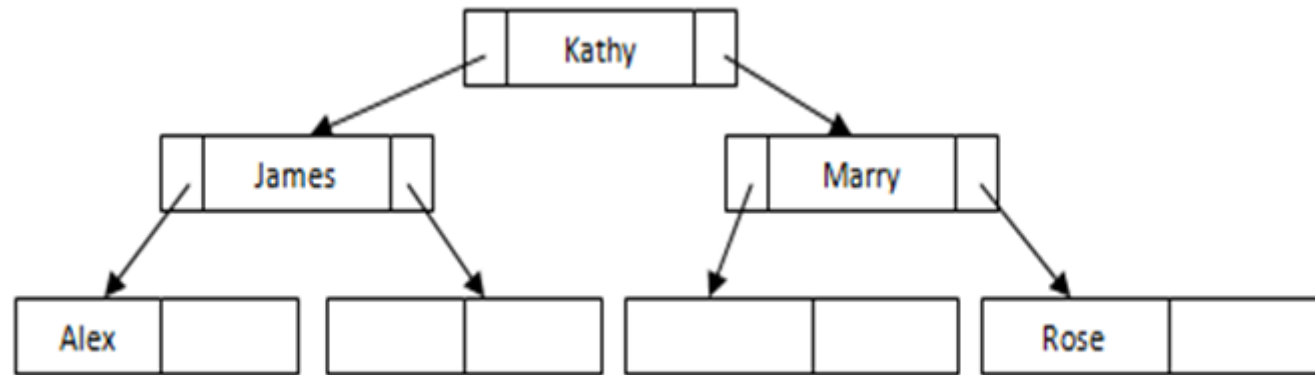


Fig: B tree

See the difference between this tree structure and B+ tree for the same example above. Here there is no repetition or pointers till leaf node. All the records are stored in all the nodes.

If we need to insert any record, it will be done as B+ tree index files, but it will make sure that each node will branch only to two nodes. If there is not enough space in any of the node, it will split the node and store the records.

Example of Simple Insert

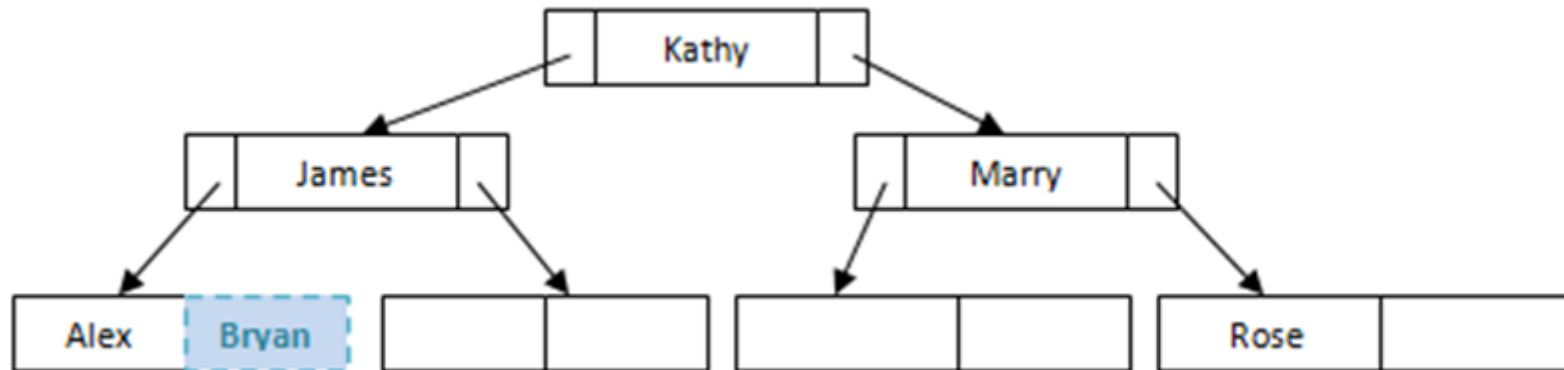


Fig: Example of Simple Insert

Example of splitting the nodes while inserting

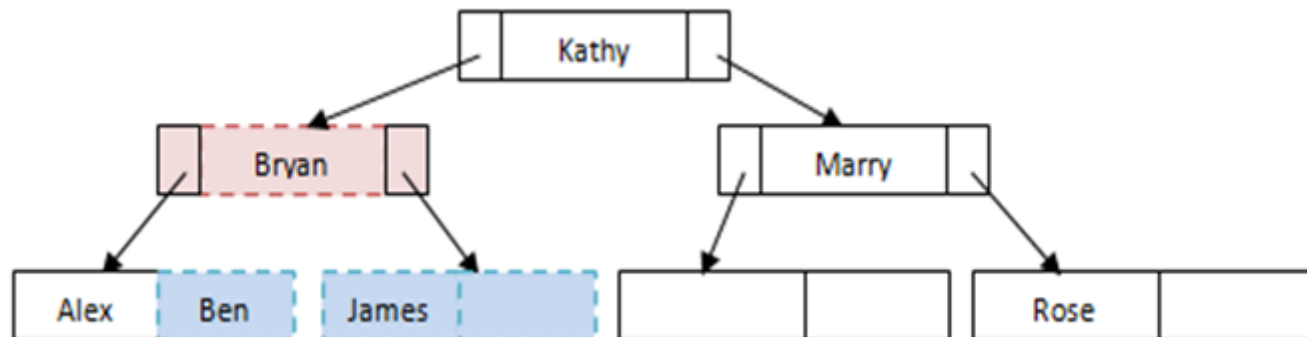


Fig: Example of splitting the nodes while inserting



0

[ADD COMMENT](#)

Please log in (</site/login/?next=/p/34173/explain-b-tree-and-b-tree-index-files-in-dbms/>) to add an answer.

Continue reading

Find answer to specific questions by searching them here. It's the best way to discover useful content.

[Find more \(/local/search/page/\)](/local/search/page/)

[\(market://details?\)](#)

id=com.ques10&utm_source=global_co&utm_medium=prtnr&utm_content=Mar2515&utm_campaign=PartBadge&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1)

Engineering in your pocket

Download our mobile app and study on-the-go. You'll get subjects, question papers, their solution, syllabus - All in one app.

COMPANY

[About Us \(/info/about/\)](/info/about/)[Community \(/user/list/\)](/user/list/)[Blog \(/t/blog/\)](/t/blog/)

CONTENT

[New Post \(/p/new/post/\)](/p/new/post/)[All Posts \(/t/latest/\)](/t/latest/)[Badges \(/b/list/\)](/b/list/)

SUPPORT

[Contact Us \(/info/contact/\)](/info/contact/)[Privacy \(/info/privacy/\)](/info/privacy/)[Terms \(/info/terms/\)](/info/terms/)[https://facebook.com/ques10\)](https://facebook.com/ques10)[https://www.youtube.com/playlist?list=UUH-Ir-oL_apReZzPeFGJopg\)](https://www.youtube.com/playlist?list=UUH-Ir-oL_apReZzPeFGJopg)[\(https://in.linkedin.com/company/ques10\)](https://in.linkedin.com/company/ques10)[tel:+919503733731\)](tel:+919503733731)[\(https://wa.me/919503733731\)](https://wa.me/919503733731)