

CMU 15-445 - Database Systems Lecture 7 - B-Trees and Indexes

B-Trees and Indexes

A table index is a replica of a subset of a table's columns that is stored in such a way that it allows the DBMS to find tuples faster than sequential scans.

Indexes dramatically speed up some types of queries, but they also take up space, and need to be kept in sync with the data. Too many indexes can make reads fast but writes slower.

The query optimizer decided which index to use when it receives a query.

B+ Tree

A B+ Tree is a self-balancing tree data structure that keeps data sorted and gives $O(\log n)$ search, sequential access, insertion, and deletion.

A B+ Tree is used in almost all DBMSs which support order-preserving indexes.

A B+ Tree comes from a class of data structures called B Trees. It has the following properties - 1. It is perfectly balanced (every leaf is at the same depth). 2. Every inner node other than the root is always at least half full. 3. Every inner node with k keys has $k+1$ children.

That is, if we have a B+ Tree of degree M , that means its nodes can contain at most $M - 1$ keys and M children. Property 2 says that every inner node has between $M/2 - 1$ to $M - 1$ keys.

Andy then explains the insertion and deletion algorithms in a B+ Tree, which I won't describe here. But they are pretty much what you would expect.