

**1. What is a B-Tree and how does it work? Explain and give an example.**

B-Trees are the most widely used indexing structures. B-trees keep key-value pairs sorted by key, which allows efficient key-value lookups and range queries. B-trees break the database down into fixed-sized blocks or pages, and read or write one page at a time. Each page can be identified using an address or location, which allows one page to refer to another. One page is designated as the root of the B-tree. The page contains several keys and references to child pages. Each child is responsible for a continuous range of keys, and the keys between the references indicate where the boundaries between those ranges lie.

A B-tree contains the following properties:

1. Every node in a B-tree contains at most  $m$  children.
2. Every node in a B-tree except the root node and the leaf node contain at least  $m/2$  children.
3. The root nodes must have at least 2 nodes.
4. All leaf nodes must be at the same level.

**2. What is Column Oriented Storage and how is it different than an RDBMS? Give an example.**

Column oriented storage stores all the values from each column together instead of all the values from a row. If each column is stored in a separate file, a query only needs to read and parse those columns that are used in that query, which can save a lot of work. The column-oriented storage layout relies on each column file containing the rows in the same order. Compared to a RDBMS, which is a row-oriented database, reading and writing operations are slower. However, column oriented storage is more efficient in performing operations to the entire dataset. They also permit high compression rates due to little distinct or unique values in columns. Overall, column oriented storage is better for online analytical processing, while RDBMS is better for online transaction processing.

**3. What is an SSTable and how does it work? What are some advantages over log segments with hash indexes?**

An SSTable contains key-value pairs, where the sequence of key-value pairs is sorted by key. An SSTable can be mapped into memory, which allows for performing lookups and scans without touching disks.

SSTables have several advantages:

1. Merging segments is simple and efficient, even if the files are bigger than the available memory.

2. In order to find a particular key in the file, you no longer need to keep an index of all the keys in memory.
3. Since read requests need to scan over several key-value pairs in the requested range, it is possible to group those records into a block and compress it before writing to disk.

#### **4. What is tail recursion and how is it different than regular recursion?**

Tail recursion is a regular recursive function except the recursive call is the last thing executed by the function. Tail recursion is advantageous to regular recursion because it can be optimized by a compiler.