Assignment 7, 4/21/2025

Drew Caldwell, CSCI-C442

Complete the following tasks using any illustration necessary:

- What is the primary purpose of an index in a relational database?

The main purpose of indexing is to improve the speed and efficiency of data retrieval. Indexing allows the database engine to find rows faster rather than scanning an entire table. Basically, indexing allows for a specific match when needed instead of scanning entire tables using where statements.

- Describe how a B-tree index works to minimize disk access times. Be sure to include the concepts of leaf nodes and branching.

A B-tree index is a self-balancing tree data structure that will maintain sorted data and allow for efficient insert, delete, and select statements. It minimizes disk I/O by using branching which is the levels of which the tree is structured. This makes it so that each internal node contains keys that point to child nodes, narrowing down the search space at each level. Leaf nodes are used then to contain the actual pointers to the data.

As the tree becomes balanced, the number of levels from root to any leaf node is the same. Using this will give us predictable performance when it comes to our operations. Since B-trees reduce the number of disk reads needed to find data, they are useful for large datasets.

- Explain the difference between a clustered index and a non-clustered index in a relational database. Provide an example of when each would be used.

A **clustered index** is where the data rows are stored in order based on their index key. Each table only has one clustered index since data rows can be sorted in only one way.

EXAMPLE: An example of a clustered index would be on a column like orderNumber, employeeID, or customerID that are frequently queried.

A **non-clustered index** stores pointers to the actual data rows. The data itself is not stored in the order of the index key. A table may have multiple non clustered indexes.

EXAMPLE: An example would be a lname, or customerName columns that are somewhat unique but frequently searched.

- Suppose you are designing an index for a table with frequent range queries, such as retrieving all rows where the value of column A is between 10 and 20. Which type of index would be most suitable? Explain your reasoning.

For frequent range queries a cluster index on column A would work well. Since the data in the rows is stored in the order of the index column, the database could efficiently retrieve all rows in the specified range using minimal disk I/O. The leaf nodes of a B-tree clustered index contain the actual data and since the values here continue, the system can quickly scan from one leaf node to the next.