Database indexing

Database indexing is a technique used to improve the speed and efficiency of data retrieval operations on a database table. An index is a data structure, usually a B-tree or a hash table, that allows the database management system (DBMS) to locate and access data more quickly.

Key Concepts of Database Indexing

1. **Index Structure**:
    - **B-tree Indexes**: These are the most common type of indexes, providing balanced tree structures that allow for efficient range queries and ordered retrieval of data.
    - **Hash Indexes**: These use a hash table to directly map keys to their associated records, allowing for very fast exact-match queries but less efficient range queries.
2. **Primary Index**:
    - Automatically created by the DBMS when a primary key is defined on a table. It uniquely identifies each record in the table.
3. **Secondary Index**:
    - Created on non-primary key columns to improve the performance of queries involving those columns.
4. **Unique Index**:
    - Ensures that the indexed columns do not have duplicate values.
5. **Composite Index**:
    - Created on multiple columns, allowing for more efficient queries that filter based on several fields.
6. **Full-Text Index**:
    - Optimizes searches for text data within large text fields, such as searching for specific words or phrases.

Benefits of Indexing

- **Faster Data Retrieval**: By reducing the amount of data the DBMS needs to scan, indexes significantly speed up the execution of queries.
- **Efficient Sorting and Filtering**: Indexes can improve the performance of ORDER BY, GROUP BY, and WHERE clauses.
- **Enforcement of Uniqueness**: Unique indexes ensure that columns do not contain duplicate values, maintaining data integrity.

Drawbacks of Indexing

- **Increased Storage Requirements**: Indexes take up additional disk space.
- **Slower Write Operations**: Insert, update, and delete operations can be slower because the indexes need to be updated along with the table data.
- **Complexity**: Managing and optimizing indexes can add complexity to database design and maintenance.