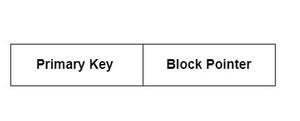
1. **Define the terms Primary Index and Secondary Index. Differentiate between them on basis of the Evaluation Criteria for indices**

A Primary Index is an ordered file whose records are of fixed length with two fields. The first field of the index is the primary key of the data file in an ordered manner, and the second field of the ordered file contains a block pointer that points to the data block where a record containing the key is available.



A primary index is an index structure that is created on the primary key of a database table. The primary key uniquely identifies each record in the table, and the primary index organizes the data in the table based on the primary key values. It typically uses a B-tree or a hash table to provide fast access to data rows based on the primary key.

Types of Primary Indexing

**Dense Indexing:** In Dense Index has an index entry for every search key value in the data file. This approach ensures efficient data retrieval but requires more storage space.

No of Index Entry = No of DB Record

**Sparse Indexing:** Sparse indexing involves having fewer index entries than records. The index entries point to blocks of records rather than individual records. While it reduces storage overhead, it may require additional disk accesses during retrieval.

No of Index Entry = No of Block

**Secondary Index:**

A secondary index, on the other hand, is an index structure that is created on a non-primary key column of a database table. Unlike the primary index, which is based on the primary key, a secondary index is created on other columns to improve the performance of queries that involve those columns. Secondary indices can speed up the retrieval of data based on the values of the indexed column(s).

Secondary indexing is a database management technique used to create additional indexes on data stored in a database. The main purpose of secondary indexing is to improve the performance of queries and to simplify the search for specific records within a database. A secondary index provides an alternate means of accessing data in a database, in addition to the primary index. The primary index is typically created when the database is created and is used as the primary means of accessing data in the database. Secondary indexes, on the other hand, can be created and dropped at any time, allowing for greater flexibility in managing the database.

|  |  |  |
| --- | --- | --- |
| **Points** | **Primary Index** | **Secondary Index** |
| Key Basis | Based on the primary key of the table, which uniquely identifies each record. | Based on columns other than the primary key, which are not necessarily unique. |
| Data Organization | Organizes data based on the primary key values. | Organizes data based on the values of the indexed column(s). |
| Access Speed | Provides extremely fast access to individual records based on the primary key. | Provides fast access to records based on the indexed column(s), but access speed may vary depending on the selectivity of the indexed column. |
| Storage Overhead | Generally, has low storage overhead as it's based on the primary key. | May have higher storage overhead, especially if the indexed column(s) have many duplicate values or if the index structure is large. |
| Insertion/Deletion Performance | Insertion and deletion performance might degrade if the primary key values are not sequentially inserted or deleted. | Insertion and deletion performance might be affected, especially if the indexed column values frequently change or if the index structure is large. |
| Update Anomaly | Less prone to update anomalies since it's based on the primary key, which typically remains stable. | More prone to update anomalies, especially if the indexed column values frequently change. |
| Maintenance Overhead | Typically requires minimal maintenance as it's usually managed automatically by the database system. | May require more frequent maintenance, especially in high-update scenarios. |
| Query Optimization | Optimizes queries that involve primary key values or range queries on the primary key. | Optimizes queries that involve the indexed column(s) or cover columns in the index. |
| Uniqueness Constraint | Enforces the uniqueness constraint on the primary key column(s). | Does not enforce uniqueness unless explicitly specified. |
| Clustering | Often used for clustering the data physically, especially in clustered index structures. | Does not necessarily imply clustering of data, though some database systems might offer clustered secondary indices. |
| Index Creation | Automatically created when defining the primary key constraint on a table. | Must be explicitly created based on the chosen columns. |
| Table Independence | Tightly coupled with the table structure and typically cannot be used independently. | Can be created on any column(s) of a table, providing more flexibility and independence. |
| Usage in Joins | Often used in join operations for fast retrieval of related records. | Can also be used in join operations, especially when joining on the indexed column(s). |
| Null Values Handling | May or may not allow null values depending on the database system and primary key definition. | Can be created on columns with null values, but the behavior might vary depending on the database system and index settings. |