Secondary indexes

CloudFolks HUB

1. Introduction Indexing

1. Indexing in a database is like a shortcut that helps find data quickly
2. Instead of searching the entire table, the index lets the database jump directly to the needed data
3. It works like an index in a book, where you quickly find the page number for a topic
4. There are two main types of indexes
   1. Primary Index
   2. Secondary Index

2. Primary Index

1. The index automatically created on the primary key column(s)
2. The primary key is always unique, ensuring that each row can be identified individually
3. Example

   1.

CustomerID(Primary Key Index)	OrderID	Name	Email
1	101	John Doe	john.doe@example.com
2	102	Jane Smith	jane.smith@example.com
3	103	Alice Johnson	alice.j@example.com
4	104	Bob Brown	bob.brown@example.com

   2. Primary Key (Primary Index) CustomerID

3. Secondary Index

1. An additional index created on columns other than the primary key
2. Used for efficiently searching data based on non-primary key fields
3. Example

   1.

CustomerID(Primary Key Index)	OrderID(Secondary Index)	Name	Email
1	101	John Doe	john.doe@example.com
2	102	Jane Smith	jane.smith@example.com
3	103	Alice Johnson	alice.j@example.com
4	104	Bob Brown	bob.brown@example.com

   Subtopic

   2. Primary Key (Primary Index) CustomerID
   3. Secondary Indexe OrderID Create a secondary index on OrderID to search for customers using their order details

4. Advantages of Secondary Indexing

1. Faster Queries Enables quicker data retrieval for non-primary key queries, reducing execution time
2. Flexible Querying Allows querying on columns other than the primary key for more access options
3. Improved Efficiency Eliminates full table scans, optimizing query performance
4. Complex Use Cases Supports frequent queries on additional attributes like Email
5. Real-Time Updates (DynamoDB) Automatically updates indexes to stay consistent with the main table
6. Partition Optimization (LSI) Allows multiple query patterns within the same partition in DynamoDB