Index

NGUYEN HongPhuong
Email: phuongnh@soict.hust.edu.vn
Site: http://users.soict.hust.edu.vn/phuongnh
Face: https://www.facebook.com/phuongnhbk
Hanoi University of Science and Technology

Contents

Introduction

- Poorly designed indexes and a lack of indexes are primary sources of database application bottlenecks. Designing efficient indexes is paramount to achieving good database and application performance.
- The technical purpose of the database index is to limit as much as possible disk IO while executing a query.

Introduction (2)

- Users can not see the indexes, they are just used to speed up searches/queries
- Updating a table with indexes takes more time than a table without indexes
- So, only create indexes on columns that will be frequently searched against

Introduction (3)

- An index is a separate data structure managed by the database, which can be used while executing a query, in order to avoid reading the entire data for a query that only requires a small part of it.
- Different implementations of an index will improve query performance for different type of operators.

Types of index

- A table/view can contain the following types of indexes:
  - Clustered
  - Non clustered

Clustered

- Clustered indexes sort and store the data rows in the table/view based on their key values
- There should be only one clustered index per a table, because the data rows themselves can be stored in only one order.

Non-clustered

- Non-clustered indexes have a structure separate from the data rows
- The pointer from an index row in a non-clustered index to a data row is called a row locator
- You can add non-key columns to the leaf level of the non-clustered index to by-pass existing index key limits, and execute fully covered, indexed, queries.

Some kinds of indexes in PostGreSQL

- PostgreSQL comes with many implementations by default for the index data structure
  - B-Tree Index - very useful for single value search or to scan a range, but also for pattern matching.
  - Hash Index - very efficient when querying for equality.
  - Generalized Inverted Index (GIN) - useful for indexing array values and testing the presence of an item.

Some kinds of indexes in PostGreSQL

- Generalized Search Tree (GiST) - a more complex index structure useful for more exotic searches such as nearest-neighbor or pattern matching.
- Space Partitioned GiST (SP-GiST) - similar with GiST, this index implementation supports space partitioned trees such as quadtrees, k-d trees, and radix trees.
- Block Range Index (BRIN) - this type of index stores summary information for each table block range

- B-Tree indexes are the default option when creating an index without specifying the type.

B-Tree Data structure

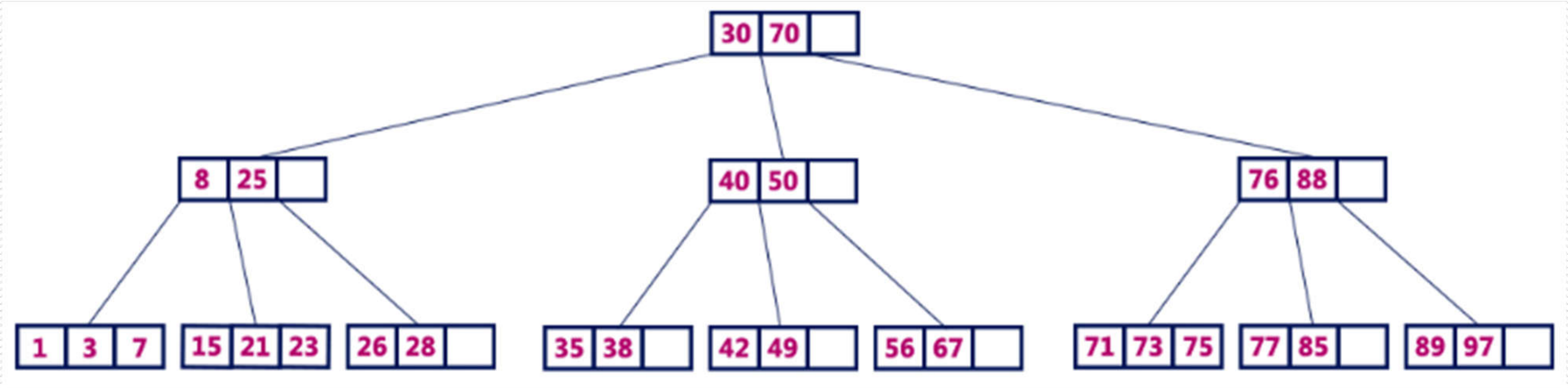
- B-Tree is a self-balanced search tree in which every node contains multiple keys and has more than two children.
- B-Tree of Order m has the following properties...
  - Property #1 - All leaf nodes must be at same level.
  - Property #2 - All nodes except root must have at least [m/2]-1 keys and maximum of m-1 keys.

B-Tree Data structure

- Property #3 - All non leaf nodes except root (i.e. all internal nodes) must have at least m/2 children.
- Property #4 - If the root node is a non leaf node, then it must have atleast 2 children.
- Property #5 - A non leaf node with n-1 keys must have n number of children.
- Property #6 - All the key values in a node must be in Ascending Order.

B-Tree Data structure

- For example, B-Tree of Order 4 contains a maximum of 3 key values in a node and maximum of 4 children for a node.



INDEX in SQL

- Syntax for create index
  - CREATE INDEX index_name ON table_name;
- Single-Column Index
  - CREATE INDEX index_name ON table_name (column_name);
- Unique index
  - CREATE UNIQUE INDEX index_name ON table_name (column_name);

INDEX in SQL

- Composite Index
  - CREATE INDEX index_name ON table_name (column_name1, column_name2);
- Drop index
  - DROP INDEX *table_name*.*index_name*;
  - DROP INDEX *index_name* ON *table_name*;

Which cases should not we use index?

- Small tables
- Tables are often updated and inserted
- Not be applied on columns which have a large number of NULL value.
- Not be applied on columns which are often updated.