
Pivotal
A NEW PLATFORM FOR A NEW ERA

GPDB Indexing Strategies

Pivotal® Greenplum Database

Agenda

- Introduction
- Supported Index Types
- When to Use an Index
- Costs Associated With Index Use
- Test it out in the lab

Pivotal

Indexes

Most data warehouse environments operate on large volumes of data:

- Queries have low selectivity
- Sequential scan is the preferred method to access data in a Greenplum MPP environment

For queries with high selectivity:

- Indexes may improve performance
- Avoid:
  - Indexes on frequently updated columns
  - Overlapping indexes
- Use bitmap indexes for columns with low cardinality
- Drop indexes before data load and recreate indexes after load
- Analyze after recreating indexes

Pivotal

Using B-Tree or Bitmap Indexes

B-Tree:

- Is used for high cardinality columns
- Is used for those columns that are single row queries
- Can be expensive (storage, time to create)

Bitmap:

- Is used for low cardinality columns
- Is typically a fraction of the size of the indexed data
- Is best when data is queried instead of updated often

Pivotal

Create Index Syntax

The following is the syntax to create an index:

```
CREATE [UNIQUE] INDEX [CONCURRENTLY]
    name ON table
   [USING method]          ( {column | (expression)}
   [opclass] [, ...] )
 [ WITH ( FILLFACTOR = value )
[WHERE predicate]
```

The following is an example of how to create a bitmap index:

```
CREATE INDEX city_state_idx ON city USING bitmap
(state_name);
```

Pivotal

B-Tree Index

- Supports single value row lookups
- Can be unique or non-unique; unique is supported only on a column that is, or is part of, the distribution key
- Can be single or multi-column
- For a multi-column index, all columns in the index must be included in the predicate for the index to be used

The following is an example of a B-tree index:

```
CREATE INDEX transid_btridx
    ON facts.transaction
    USING BTREE (transactionid);
```

Pivotal

Bitmap Index

- Index a single column
- Efficient for queries with multiple conditions on the predicate
- Provides very fast retrieval
- Best for low cardinality columns, such as:
  - Product category
  - State, or Zip code

The following are examples of bitmap indexes:

```
CREATE INDEX store_pharm_bmidx ON dimensions.store
    USING BITMAP (pharmacy);
CREATE INDEX store_grocery_bmidx ON dimensions.store
    USING BITMAP (grocery);
CREATE INDEX store_deli_bmidx ON dimensions.store
    USING BITMAP (deli);
```

Pivotal

Index on Expressions

- Should only be used when the expression appears often in query predicates
- Has a very high overhead maintaining the index during insert and update operations

The following shows how it is used:

```
CREATE INDEX lcase_storename_idx
    ON store (LOWER(storename));
```

This syntax supports the following query:

```
SELECT * FROM store WHERE LOWER(storename) = 'top foods';
```

Pivotal

Index with Predicate (Partial Index)

- Pre-selects rows based on predicate
- Is used to select small numbers of rows from large tables

The following is an example of a partial index:

```
CREATE INDEX canada_stores_idx
    ON facts.transaction
    WHERE storeid IN(8,32);
```

Pivotal

Greenplum Indexes – Partitioned Tables

Consider whether only the most recent data should be indexed, as in this example:

- A partitioned transaction table requires a B-tree index on the transaction id to support single row queries.
- Customer Support only needs to access the past 30 days of data.
- The transaction table has weekly partitions.

The solution is to index the 4 most recent partitions on a *rolling* basis.

To Index or Not to Index

Consider the following:

- Will the optimizer use the index?
- Is the column(s) used in query predicates?
  - Does the frequency of use justify the overhead?
  - Is the space available?
- Are you working with *compressed* append-only tables?

Note: Greenplum Database will automatically create PRIMARY KEY indexes for tables with primary keys.

Pivotal

Maintaining Indexes

To maintain overall performance:

- Check the disk space usage for your index
- Update or reindex your indexes if queries are taking too much time
- Clustered indexes can reduce disk seek time

Pivotal

Review

- Supported Index Types
- When to Use an Index
- Costs Associated With Index Use
- Test it out in the lab

Pivotal


Pivotal
A NEW PLATFORM FOR A NEW ERA