SQL indexes

In this we will see how to create, delete and uses of index in database.

An index is a schema object. It is used by the server to speed up the retrieval of rows by using a pointer. It can reduce disk I/O(input/output) by using a rapid path access method to locate data quickly. An index helps to speed up select queries and where clauses, but it slows down data input, with the update and the insert statements. Indexes can be created or dropped with no effect on the data.

For example, if you want to reference all pages in a book that discusses a certain topic, you first refer to the index, which lists all the topics alphabetically and are then referred to one or more specific page numbers.

Creating an Index – It’s syntax is:

CREATE INDEX index

ON TABLE column;

where index is the name given to that index and TABLE is the name of the table on which that index is created and column is the name of that column for which it is applied.

For multiple columns –

CREATE INDEX index

ON TABLE (cloumn1, column2,.....);

Unique Indexes –

CREATE UNIQUE INDEX index

ON TABLE column;

Unique indexes are used for the maintenance of the integrity of the data present in the table as well as for the fast performance, it does not allow multiple values to enter into the table.

When should indexes be created –

A column contains a wide range of values

A column does not contain a large number of null values

One or more columns are frequently used together in a where clause or a join condition

When should indexes be avoided –

The table is small

The columns are not often used as a condition in the query

The column is updated frequently

Removing an Index – To remove an index from the data dictionary by using the DROP INDEX command.

DROP INDEX index;

To drop an index, you must be the owner of the index or have the DROP ANY INDEX privilege.

Confirming Indexes –

You can check the different indexes present on a particular table given by the user or the server itself and their uniqueness.

select \*

from USER\_INDEXES;

It will show you all the indexes present in the server, in which you can locate your own tables too.

Indexes are special lookup tables that the database search engine can use to speed up data retrieval. Simply put, an index is a pointer to data in a table. An index in a database is very similar to an index in the back of a book.

For example, if you want to reference all pages in a book that discusses a certain topic, you first refer to the index, which lists all the topics alphabetically and are then referred to one or more specific page numbers.

An index helps to speed up SELECT queries and WHERE clauses, but it slows down data input, with the UPDATE and the INSERT statements. Indexes can be created or dropped with no effect on the data.

Creating an index involves the CREATE INDEX statement, which allows you to name the index, to specify the table and which column or columns to index, and to indicate whether the index is in an ascending or descending order.

Indexes can also be unique, like the UNIQUE constraint, in that the index prevents duplicate entries in the column or combination of columns on which there is an index.

The CREATE INDEX Command

The basic syntax of a CREATE INDEX is as follows.

CREATE INDEX index\_name ON table\_name;

Single-Column Indexes

A single-column index is created based on only one table column. The basic syntax is as follows.

CREATE INDEX index\_name

ON table\_name (column\_name);

Unique Indexes

Unique indexes are used not only for performance, but also for data integrity. A unique index does not allow any duplicate values to be inserted into the table. The basic syntax is as follows.

CREATE UNIQUE INDEX index\_name

on table\_name (column\_name);

Composite Indexes

A composite index is an index on two or more columns of a table. Its basic syntax is as follows.

CREATE INDEX index\_name

on table\_name (column1, column2);

Whether to create a single-column index or a composite index, take into consideration the column(s) that you may use very frequently in a query's WHERE clause as filter conditions.

Should there be only one column used, a single-column index should be the choice. Should there be two or more columns that are frequently used in the WHERE clause as filters, the composite index would be the best choice.

Implicit Indexes

Implicit indexes are indexes that are automatically created by the database server when an object is created. Indexes are automatically created for primary key constraints and unique constraints.

The DROP INDEX Command

An index can be dropped using SQL DROP command. Care should be taken when dropping an index because the performance may either slow down or improve.

The basic syntax is as follows −

DROP INDEX index\_name;

You can check the INDEX Constraint chapter to see some actual examples on Indexes.

When should indexes be avoided?

Although indexes are intended to enhance a database's performance, there are times when they should be avoided.

The following guidelines indicate when the use of an index should be reconsidered.

Indexes should not be used on small tables.

Tables that have frequent, large batch updates or insert operations.

Indexes should not be used on columns that contain a high number of NULL values.

Columns that are frequently manipulated should not be indexed.