

MySQL Indexes

Last Updated Thursday, 13 March 2008

A database index is a data structure that improves the speed of operations in a table. Indexes can be created using one or more columns, providing the basis for both rapid random lookups and efficient ordering of access to records. While creating index it should be considered that what are the columns which will be used to make SQL queries and create one or more indexes on those columns. Practically, Indexes are also type of tables which keeps primary key or index field and a pointer to each record in to the actual table. The users cannot see the indexes, they are just used to speed up queries and will be used by Database Search Engine to locate records very fast. INSERT and UPDATE statements takes more time on tables having indexes where as SELECT statements become fast on those tables. The reason is that while doing insert or update, database need to inert or update index values as well. Simple and Unique Index: You can create a unique index on a table. A unique index means that two rows cannot have the same index value. Here is the syntax to create an Index on a table

```
ON table_name ( column1, column2,...);
```

You can use one or more columns to create an index. For example we can create an index on tutorials_tbl using tutorial_author

```
CREATE UNIQUE INDEX AUTHOR_INDEX
```

```
ON tutorials_tbl (tutorial_author)
```

You can creates a simple index on a table. Just omit UNIQUE keyword from the query to create simple index. Simple index allows duplicate values in a table. If you want to index the values in a column in descending order, you can add the reserved word DESC after the column name:

```
mysql> CREATE UNIQUE INDEX AUTHOR_INDEX
```

ON tutorials_tbl (tutorial_author DESC)

ALTER command to add and drop INDEX: There are four types of statements for adding indexes to a table:

```
ALTER TABLE tbl_name ADD PRIMARY KEY (column_list) :
```

This statement adds a PRIMARY KEY, which means that indexed values must be unique and cannot be NULL.

```
ALTER TABLE tbl_name ADD UNIQUE index_name (column_list):
```

This statement creates an index for which values must be unique (with the exception of NULL values, which may appear multiple times).

```
ALTER TABLE tbl_name ADD INDEX index_name (column_list):
```

This adds an ordinary index in which any value may appear more than once.

```
ALTER TABLE tbl_name ADD FULLTEXT index_name (column_list):
```

This creates a special FULLTEXT index that is used for text-searching purposes. Here is the example to add index in an existing table.

```
mysql> ALTER TABLE testalter_tbl ADD INDEX (c);
```

You can drop any INDEX by using DROP clause along with ALTER command. Try out following example to drop above created index.

```
mysql> ALTER TABLE testalter_tbl DROP INDEX (c);
```

You can drop any INDEX by using DROP clause along with ALTER command. Try out following example to drop above created index.

```
ALTER Command to add and drop PRIMARY KEY: You can add primary key as well in the same way. But make sure Primary Key works on columns which are NOT NULL. Here is the example to add primary key in an existing table. This will make a column NOT NULL first and then add it as a primary key.
```

```
mysql> ALTER TABLE testalter_tbl MODIFY i INT NOT NULL;
```

```
mysql> ALTER TABLE testalter_tbl ADD PRIMARY KEY (i);
```

You can use ALTER command to drop a primary key as follows:

```
mysql> ALTER TABLE testalter_tbl DROP PRIMARY KEY;
```

To drop an index that is not a PRIMARY KEY, you must specify the index name.

Displaying INDEX Information: You can use SHOW INDEX command to list out all the indexes associated with a table. Vertical-format output (specified by \G) often is useful with this statement, to avoid long line wraparound: Try out following example:

```
mysql> SHOW INDEX FROM table_name\G
```

.....