

MYSQL - INDEXES

<https://www.tutorialspoint.com/mysql/mysql-indexes.htm>

Copyright © tutorialspoint.com

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 taken into consideration which all columns will be used to make SQL queries and create one or more indexes on those columns.

Practically, indexes are also a type of tables, which keep primary key or index field and a pointer to each record into the actual table.

The users cannot see the indexes, they are just used to speed up queries and will be used by the Database Search Engine to locate records very fast.

The INSERT and UPDATE statements take more time on tables having indexes, whereas the SELECT statements become fast on those tables. The reason is that while doing insert or update, a database needs to insert or update the 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.

```
CREATE UNIQUE INDEX index_name
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 create a simple index on a table. Just omit the **UNIQUE** keyword from the query to create a simple index. A Simple index allows duplicate values in a table.

If you want to index the values in a column in a 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_ilist** – This statement adds a **PRIMARY KEY**, which means that the indexed values must be unique and cannot be NULL.
- **ALTER TABLE tbl_name ADD UNIQUE index_name column_ilist** – This statement creates an index for which the values must be unique *except for the 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.

The following code block is an example to add index in an existing table.

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

You can drop any INDEX by using the **DROP** clause along with the ALTER command.

Try out the following example to drop the above-created index.

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

You can drop any INDEX by using the DROP clause along with the ALTER command.

ALTER Command to add and drop the PRIMARY KEY

You can add a primary key as well in the same way. But make sure the Primary Key works on columns, which are NOT NULL.

The following code block is an example to add the 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 the 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 the **SHOW INDEX** command to list out all the indexes associated with a table. The vertical-format output *specified by* **\G** often is useful with this statement, to avoid a long line wraparound –

Try out the following example –

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