MySQL Index Types

Unique INDEX

- Unique is a type of MySQL Index which specifies that all values of the tables columns, when implemented, have to be distinct.
- There can be no duplicate values in the column which is indexed existing in a single column unique index.
- Whereas in a multi-column unique index, columns values can be replicated in a single table column, but the grouping of column values in every row should be unique.
- Hence, this Unique index is useful to avoid the duplicity of values in the columns.
- We define the MySQ
- CREATE UNIQUE INDEX Indexname ON Tablename(indexcolumn1, indexcolumn2,…);
- CREATE TABLE Tablename(Column1 CHAR (30) NOT NULL, UNIQUE INDEX (column2));
- ALTER TABLE Tablename ADD UNIQUE INDEX (column1, column2);L Index after we have created the table.

Primary Key Or CLUSTERED Index

- Primary Key is a type of Unique Index which specifies that the column mentioned as Primary key must contain no NULL value.
- Each table row should have a value for the table column or group of columns.
- Due to this, we need to normally define a primary index on the lowest number of columns as possible, and mostly, the primary key is set on a single column.
- Once the column values are set in, the primary key cannot be modified.
- CREATE TABLE Tablename (Columnname Data_type PRIMARY KEY);
- CREATE TABLE Tablename (Column1 CHAR(30) NOT NULL, Column2 CHAR(30) NOT NULL, PRIMARY KEY (Column1, Column2));
- ALTER TABLE Tablename ADD PRIMARY KEY(Column1, Column2);

Simple , Regular , Normal INDEX

- In this type of Index named as simple, regular or normal, the specified column values do not require to be unique and as well as can be NULL also.
- They are supplemented basically to aid the database searching process for records quicker.
- Find the query code below to create the simple, regular and normal type of index:
- CREATE INDEX Indexname ON Tablename (Indexcolumn1, Indexcolumn2, …);
- CREATE TABLE Tablename(Column1 CHAR(30) NOT NULL, INDEX(Column2));
- ALTER TABLE Tablename ADD INDEX(Column1, Column2);

Full-Test Index

- This type of index is implemented for full-text searches as the name itself implies it.
- Suppose you need to find the blob of text which includes a certain word or combination of words, or it can even be a substring found within the higher block text, then this full-text index plays a vital role and,
- Therefore, broadly implemented in search engines and e-commerce.
- But this indexes are maintained only for MyISAM and InnoDB tables in the database and can consist of only VARCHAR, CHAR, and TEXT table columns.
- CREATE FULLTEXT INDEX Indexname ON Tablename(indexcolumn1, indexcolumn2, …);
- CREATE TABLE Tablename (column1 TEXT, FULLTEXT(column1));
- ALTER TABLE Tablename ADD FULLTEXT(column1, column2);

Descending INDEX

- This Descending index is a consistent index stored in the inverse order and is available only version 8+ of MySQL.
- It is supportive when a user run queries to find out the most newly added data, such as you may want to view the six newest posts or the five most current comments on all the posts.
- Scanning the indexes in inverse order previously have caused a performance issue.
- It will be more resourceful to use the descending index in onward order.
- CREATE INDEX Indexname ON Tablename (Indexcolumn1 DESC);
- CREATE TABLE Tablename (Column1 INT, Column2 INT, INDEX asc_index1(Column1 ASC, Column2 ASC), INDEX desc_index1 (Column1 DESC, Column2 DESC));
- ALTER TABLE Tablename ADD INDEX(column1 DESC, column2 ASC);