MySQL - Unique Index

MySQL Indexes are used to return the data from the database real quick. The users cannot see the indexes performing, instead they are just used to speed up the queries.

However a unique index, in addition to speeding up data retrieval queries, is also used to maintain data integrity in a table. When a unique index is defined on a table column, we cannot add any duplicate values into that column.

MySQL Unique Index

A unique index can be created on one or more columns of a table using the CREATE UNIQUE INDEX statement in MySQL.

- **If we are creating unique index on only a single column, all the rows in that column must be unique.**
- **We cannot create a unique index where NULL values are present in multiple rows in a single column.**
- **If we are creating unique index on multiple columns, the combination of rows in those columns must be unique.**
- **We cannot create a unique index on multiple columns if the combination of columns contains NULL values in more than one row.**

Syntax

Following is the syntax for creating a unique index in MySQL –

```
CREATE UNIQUE INDEX index_name

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

Example

Let us first create a table named CUSTOMERS using the following query –

```
CREATE TABLE CUSTOMERS (
   ID INT NOT NULL,
   NAME VARCHAR(15) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS VARCHAR(25),
   SALARY DECIMAL(10, 2),
   PRIMARY KEY(ID)
);
```

```
INSERT INTO CUSTOMERS VALUES
(1, 'Ramesh', '32', 'Ahmedabad', 2000),
(2, 'Khilan', '25', 'Delhi', 1500),
(3, 'Kaushik', '23', 'Kota', 2500),
(4, 'Chaitali', '26', 'Mumbai', 6500),
(5, 'Hardik','27', 'Bhopal', 8500),
(6, 'Komal', '22', 'MP', 9000),
(7, 'Muffy', '24', 'Indore', 5500);
```

Now, create a unique index for the column named **SALARY** in the CUSTOMERS table using the following query –

```
CREATE UNIQUE INDEX unique_ind ON CUSTOMERS (SALARY);
```

Inserting Duplicate Values

Now, let us try to update the value in the SALARY column with a duplicate (already existing data) value using the following query −

```
UPDATE CUSTOMERS SET SALARY = 2000 WHERE ID = 2;
```

Error

The above query results in an error because a column that has unique index cannot contain duplicate values in it.

```
ERROR 1062 (23000): Duplicate entry '2000.00' for key 'customers.unique_ind'
```

Creating Unique Index on Multiple Columns

In MySQL, we can also create a unique index on multiple columns of a table using the CREATE UNIQUE INDEX statement. To do so, you just need to pass the name of the columns (you need to create the index on) to the query.

Example

Assume the previously created CUSTOMERS table and create a unique index on the columns named **NAME** and **AGE** using the following query −

```
CREATE UNIQUE INDEX mul_unique_index ON CUSTOMERS(NAME, AGE);
```

Verification

Using the following query, we can list all the indexes that are created on the CUSTOMERS table −

```
SHOW INDEX FROM CUSTOMERS\G
```