

Unique Indexes

SQL Unique Indexes

The **SQL Unique Index** ensures that no two rows in the indexed columns of a table have the same values (no duplicate values allowed).

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

Following are the points to be noted before creating a Unique Index on a table:

- **If the unique index is only created on a single column** – The rows in that column will be unique, ensuring that no duplicate values exist in the column.
- **If a single column contains NULL in multiple rows** – We cannot create a unique index on that column, as NULL values are not considered equal and hence cannot enforce uniqueness.
- **If the unique index is created on multiple columns** – The combination of rows in these columns will be unique, meaning that the combination of values across the specified columns must be unique.
- **We cannot create a unique index on multiple columns if the combination of columns contains NULL in more than one row** – The presence of NULL in multiple rows across the indexed columns prevents enforcing uniqueness on those combinations.

Syntax

Following is the syntax for creating a **UNIQUE INDEX** in SQL:

```
CREATE UNIQUE INDEX index_name  
ON table_name  
(column1, column2, ..., columnN);
```

Here,

- **index_name** – The name of the index that you want to create.
- **table_name** – The name of the table on which you want to create the index.
- **(column1, column2, ..., columnN)** – The names of one or more columns on which the unique index is being created.

Example

First of all, let us 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, 4),  
PRIMARY KEY (ID)  
);
```

Insert some values into the above-created table using the following query:

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

Once the table is created, let us create a unique index for the column named **SALARY** in the CUSTOMERS table using the following query:

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

But, when we execute the above query, the output is obtained as follows:

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

Since a unique index could not be created on SALARY column (due to duplicate values), let us create **Unique Index** on the **NAME** column of the same table, using the following query:

```
CREATE UNIQUE INDEX UNIQUE_ID  
ON CUSTOMERS  
(NAME);
```

Output

When we execute the above query, the output is obtained as follows:

```
Query OK, 0 rows affected (0.03 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Verification

Let's verify whether the unique index for the column NAME is created or not using the following query:

```
SHOW INDEX FROM CUSTOMERS;
```

As you observe the output below, you can find the column NAME along with the ID (PRIMARY KEY) in the list of indexes.

Table	Non_unique	Key_name	Seq_in_index	Column_name
customers	0	PRIMARY	1	ID
customers	0	UNIQUE_ID	1	NAME

Updating with Duplicate Values

If we try to update the columns that have unique index with duplicate values, the database engine generates an error.

Example

Assume the previously created CUSTOMERS table and create a unique index on the column named ADDRESS using the following query:

```
CREATE UNIQUE INDEX ADD_UNIQUE_INDEX ON  
CUSTOMERS(ADDRESS);
```

Now, let us update the value in the column named ADDRESS with a duplicate (already existing data) value using the following query:

```
UPDATE CUSTOMERS  
SET ADDRESS = 'Mumbai'  
WHERE ADDRESS = 'Delhi';
```

Output

On executing the above query, the output is displayed as follows:

```
ERROR      1062      (23000):  Duplicate      entry      'Mumbai'      for      key  
'customers.ADD_UNIQUE_INDEX'
```

Creating a unique index on Multiple Fields

We can also create a unique index on multiple fields or 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

Instead of creating a new table, let us consider the previously created CUSTOMERS table. We will create a unique index on the columns NAME and AGE using the following query:

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

Output

When we execute the above query, the output is obtained as follows:

```
Query OK, 0 rows affected (0.04 sec)  
Records: 0 Duplicates: 0 Warnings: 0.
```

Verification

Now, let us list all the indexes that are created on the CUSTOMERS table using the following query:

```
SHOW INDEX FROM CUSTOMERS;
```

As you observe you can find the column names NAME, and AGE along with the ID (PRIMARY KEY) in the list of indexes.

Table	Non_unique	Key_name	Seq_in_index	Column_name
customers	0	PRIMARY	1	ID
customers	0	MUL_UNIQUE_INDEX	1	NAME
customers	0	MUL_UNIQUE_INDEX	2	AGE