⌂ › ▤ › UNIQUE Key in SQL

UNIQUE Key in SQL

Challenge Inside! : Find out where you stand! Try quiz, solve problems & win rewards! — Go to Challenge

Learn via video course

DBMS Course
By **Srikanth Varma**   ₹ Free
⭐ 5   Enrolled: 18119

Start Learning

View all courses →

Overview

A unique key in SQL is the set of fields or columns of a table that helps us uniquely identify records. The unique key guarantees the uniqueness of the columns in the database. It is similar to the primary key but can accept a null value, unlike it. A primary key constraint **automatically defines a unique key constraint**. More than one unique key can exist in a table, unlike the primary key, which can exist only once.

Student

Roll_no	Name	Class	Phone_no	Registration_no
1	Andrew	5	9854672256	895
2	Andrew	6	9955512456	564
3	Augosto	5		567

↑ Primary key   ↑ Unique key   ↑ Unique key

SCALER
Topics

How can I make a column Unique in SQL?

We can make a column of a table unique in the following ways:

Method 1

We can make the column unique by adding the UNIQUE keyword at the end of its definition. **Let's understand it with an example**. We will create a table named employee, having the **name** column unique.

```
CREATE TABLE employee (
id INT NOT NULL PRIMARY KEY,
name VARCHAR(100) UNIQUE,
address VARCHAR(100),
designation VARCHAR(100)
);
```

The above SQL Query creates a table named employee and makes the column name unique by adding the UNIQUE clause at the end of the **column definition (name VARCHAR(100) UNIQUE)**. Thus, the new employee table we created has a column named **name** that contains unique values in its row.

Method 2

If the table already exists, the table gets modified, and the unique constraint is added to the column using the ALTER TABLE clause.

```
ALTER TABLE employee
ADD CONSTRAINT UQ_employee_name UNIQUE(name);
```

The above example adds the unique constraint to a column if the table employee already exists. This example uses the ALTER table clause to modify the table and make the column named name unique by using the UNIQUE constraint. The ALTER TABLE takes the name of the **table (employee)**. It is then followed by the clause ADD CONSTRAINT and the name of the unique constraint we want to use. The unique constraint name we have used in the query is UQ_employee_name. The column name follows it in **parenthesis used with the UNIQUE** keyword.

Method 3

We can also apply a UNIQUE constraint on multiple columns simultaneously. Let's see how:

```
ALTER TABLE employee
ADD CONSTRAINT UQ_employee_name_address UNIQUE(name, address);
```

The above example uses the clause ALTER TABLE to modify the table employee. The constraint name **(UQ_employee_name_address)** is used with the clause ADD CONSTRAINT. It gets followed by the UNIQUE keyword containing the column names as a list enclosed with parenthesis. The name and address columns in the above example make a unique key.

How do I change the Unique Key in SQL?

The **unique key in SQL** can be changed by first deleting the existing unique key. The new unique key gets created with a new definition.

Example

To change the unique key from UNIQUE KEY

user_id (user_id , account_id) to UNIQUE KEY user_id (user_id,account_id,pet_id).

The first way to change the unique key is to alter the existing table by dropping the previous unique key and adding a new one in the same statement.

Let's see it with an example.

```
ALTER TABLE your_table
    DROP INDEX user_id,
    ADD UNIQUE KEY `user_id` (`user_id`,`account_id`,`pet_id`)
```

It is done separately by dropping the existing key and creating the new one.

```
UPDATE your_table
SET uniq_key = CONCAT(user_id, account_id, pet_id)
WHERE uniq_key= ''
```

How to find a Unique Key in SQL?

We can find the UNIQUE key in SQL in the following two ways. Using the DISTINCT keyword or checking the constraint type with the **WHERE** clause.

Examples to find a unique key in SQL using the DISTINCT keyword

- select distinct * from employees; The above SQL statement retrieves any row if the table has at least a single unique column.

- select distinct first_name from employees; The above SQL statement removes all the duplicate values of first names and retrieves the unique first names from the table.

- select distinct count(*) from employees; The above SQL query retrieves the number of unique rows present in the table if it contains at least a single row having unique data.

Examples to find a unique key in SQL by checking the constraint type

Using the following query, we can also find all the UNIQUE constraints in a table.

We can create a table named ExampleTable.

```
CREATE TABLE ExampleTable (
    ID  int          ,
    FirstName       varchar(50)  ,
    LastName        varchar(50)  ,
    Primary Key (ID),
    CONSTRAINT UK_FName_LName UNIQUE(FirstName,LastName)
)
```

The above SQL Query creates the table ExampleTable with fields named ID, FirstName, and LastName. The primary key gets set on the column ID. We use the constraint name **UK_FName_LName** with the CONSTRAINT clause. It gets followed by the UNIQUE keyword with the name of multiple columns as unique keys enclosed in parenthesis.

```
select CCU.CONSTRAINT_NAME, CCU.COLUMN_NAME
from INFORMATION_SCHEMA.TABLE_CONSTRAINTS as TCons
inner join INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE as CCU
    on      TCons.CONSTRAINT_CATALOG = CCU.CONSTRAINT_CATALOG
        and TCons.CONSTRAINT_SCHEMA = CCU.CONSTRAINT_SCHEMA
        and TCons.CONSTRAINT_NAME = CCU.CONSTRAINT_NAME
where       TCons.CONSTRAINT_TYPE = 'UNIQUE'
        and TCons.TABLE_NAME = 'ExampleTable'
```

The above SQL Query matches the columns with the constraint type as UNIQUE and prints them with their constraint name in the output.

Output:

```
|CONSTRAINT_NAME  |      COLUMN_NAME|
|:---------------:    |   :--------------: |
|UK_FName_LName  |  FirstName|
|UK_FName_LName  |  LastName|
```

How to Remove a Unique Key in SQL?

Removing a UNIQUE key in SQL is similar to other constraints. The first step is to find the existing unique key in the table. The next step is to use the **ALTER TABLE** statement in SQL query with the **DROP** constraint that helps

delete the unique key found in the previous step.

```
ALTER TABLE TableB
DROP CONSTRAINT UK_FirstName
```

Conclusion

- A unique key in SQL is a table's set of fields or columns that helps us uniquely identify records.

- The unique key in SQL can be changed by first deleting the existing unique key. The new unique key gets created with a new definition.

- We can find the UNIQUE key in SQL using the **DISTINCT** keyword or checking the constraint type with the **WHERE** clause.

- We can remove a UNIQUE key in SQL in two steps. The first step is to find the existing unique key in the table. The next step is to use the **ALTER TABLE** statement in SQL query with the DROP constraint that helps delete the unique key.

Challenge Time!

Time to test your skills and win rewards!

Start Challenge

Note: Rewards will be credited after the next product update.

This article is written by

N Nisha Bharti

Updated - 25 Jul 2022 • 5 mins read • Published : 25 Jul 2022

Scroll to top!

SQL

How would you rate this article?

☆ ☆ ☆ ☆ ☆

Article Outline
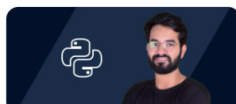
▶ Free Courses by top Scaler instructors

Java Course Online for...
A Course by **Tarun Luthra**
Learn programming fundamentals and core concepts of Java, the most wid...
☆ 5   👤 45468

Free Python Certification...
A Course by **Rahul Janghu**
Learn programming fundamentals and core concepts of Python.
☆ 4.90   👤 29048

Free C++ Course: Learn the...
A Course by **Prateek Narang**
Learn logic building & basics of programming by learning C++, one o...
☆ 5   👤 24436

DBMS Course
A Course by **Srikanth Varma**
Learn the working of Database Management Systems and importan...
☆ 5   👤 18119

Free Javascript Course -...
A Course by **Mrinal Bhattacharya**
Learn programming fundamentals and core concepts of JavaScript, the...
☆ 4.8   👤 17582

Python and SQL...
A Course by **Srikanth Varma**
Learn Python and SQL for Science along with librarie...
☆ 5

View All

💡 Got suggestions? **We would love to hear your feedback.**