

**SQL Not Null Constraint** 

**SQL Unique Constraints** 

**SQL Primary Key** 

SQL Foreign Key

SQL Check

**SQL** Default

**SQL** Additional Topics

**Related Topics** 

**SQL** Constraints

SQL CREATE INDEX

**SQL CHECK Constraint** 

**SQL UNIQUE Constraint** 

**SQL NOT NULL Constraint** 

**SQL DEFAULT Constraint** 

SQL Create Index

## **SQL PRIMARY KEY**

In this tutorial, we'll learn about the PRIMARY KEY in SQL and how to use them with the help of examples.

∷≡

In SQL, the PRIMARY KEY constraint is used to uniquely identify rows.

The PRIMARY KEY constraint is simply a combination of NOT NULL and UNIQUE constraints. Meaning, the column cannot contain duplicate as well as NULL values.

#### **Primary Key Syntax**

```
CREATE TABLE Colleges (
 college_id INT,
 college_code VARCHAR(20) NOT NULL,
 college_name VARCHAR(50),
 CONSTRAINT CollegePK PRIMARY KEY (college_id)
                                                                          Run Code >>
```

Here, the college\_id column is the PRIMARY KEY. This means, the values of this column must be unique as well as it cannot contain NULL values.

Note: The above code works in all major database systems. However, depending on a database, there may be alternative syntaxes to create the primary key.

## **Primary Key Error**

If we try to insert null or duplicate values in the college\_id column-in the above tablewe will get an error. For example,



Here, the SQL command gives us an error because we cannot insert same value for the college\_id field in a table because of the UNIQUE constraint.

**Note:** In a table, there can be only one primary key.

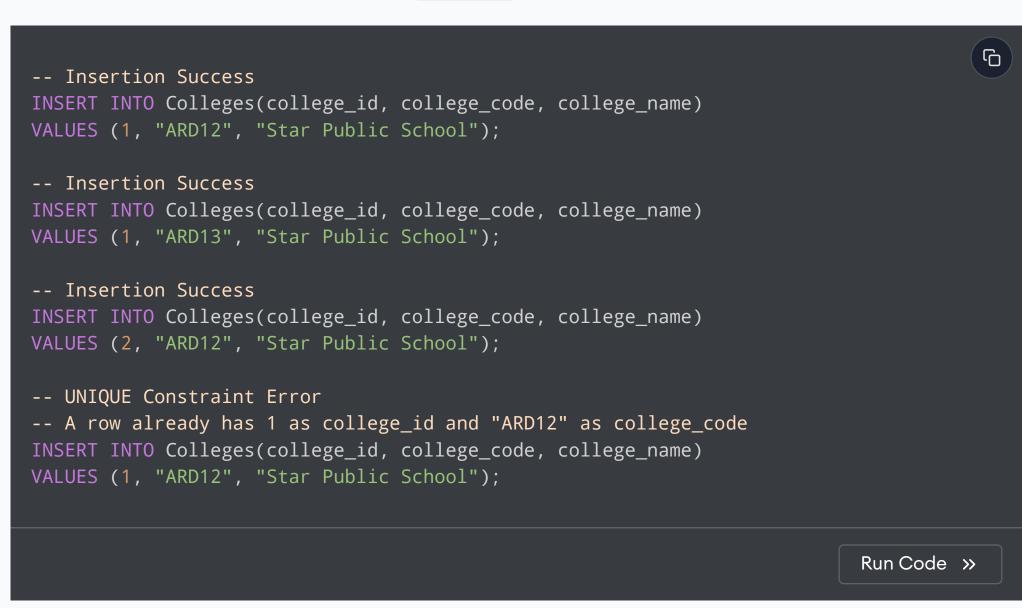
#### **Primary Key With Multiple Columns**

```
A primary key may also be made up of multiple columns. For example,
 CREATE TABLE Colleges (
   college_id INT,
   college_code VARCHAR(20),
   college_name VARCHAR(50),
   CONSTRAINT CollegePK PRIMARY KEY (college_id, college_code)
                                                                          Run Code »
```

Here, the PRIMARY KEY constraint named CollegePK is made up of college\_id and college\_code columns.

This means, the combination of college\_id and college\_code must be unique as well as these two columns cannot contain NULL values.

Now let's try to insert records in the Colleges table,



### Primary Key Constraint With Alter Table

We can also add the PRIMARY KEY constraint to a column in an existing table using the ALTER TABLE command. For example,

### For single column

```
ALTER TABLE Colleges
  ADD PRIMARY KEY (college_id);
For multiple column
 ALTER TABLE Colleges
  ADD CONSTRAINT CollegePK PRIMARY KEY (college_id, college_code);
Here, the SQL command adds the PRIMARY KEY constraint to the specified column(s) in the
```

existing table.

## **Auto Increment Primary Key**

It is a common practice to automatically increase the value of the primary key when a new row is inserted. For example,

## **SQL Server**

```
-- using IDENTITY(x, y) to auto increment the value
-- x -> start value, y -> steps to increase
CREATE TABLE Colleges (
 college_id INT IDENTITY(1,1),
 college_code VARCHAR(20) NOT NULL,
 college_name VARCHAR(50),
 CONSTRAINT CollegePK PRIMARY KEY (college_id)
-- inserting record without college_id
INSERT INTO Colleges(college_code, college_name)
VALUES ("ARD13", "Star Public School");
```

## Oracle

```
-- creating sequence of numbers
CREATE SEQUENCE auto_inc
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10;
CREATE TABLE Colleges (
 college_id INT,
 college_code VARCHAR(20) NOT NULL,
  college_name VARCHAR(50),
 CONSTRAINT CollegePK PRIMARY KEY (college_id)
-- creating trigger before insert to
-- add auto incremented value
CREATE TRIGGER auto_inc_trigger
BEFORE INSERT ON Colleges
FOR EACH ROW
BEGIN
SELECT auto_inc.nextval INTO :new.college_id FROM dual
END;
-- inserting record without college_id
INSERT INTO Colleges(college_code, college_name)
VALUES ("ARD13", "Star Public School");
```

# **MySQL**

```
-- AUTO_INCREMENT keyword auto increments the value
CREATE TABLE Colleges (
 college_id INT AUTO_INCREMENT,
 college_code VARCHAR(20) NOT NULL,
 college_name VARCHAR(50),
 CONSTRAINT CollegePK PRIMARY KEY (college_id)
-- inserting record without college_id
INSERT INTO Colleges(college_code, college_name)
VALUES ("ARD13", "Star Public School");
```

# **PostgreSQL**

```
-- SERIAL keyword auto increments the value
CREATE TABLE Colleges (
 college_id INT SERIAL,
 college_code VARCHAR(20) NOT NULL,
 college_name VARCHAR(50),
 CONSTRAINT CollegePK PRIMARY KEY (college_id)
-- inserting record without college_id
INSERT INTO Colleges(college_code, college_name)
VALUES ("ARD13", "Star Public School");
```

## **Remove Primary Key Constraint** We can remove the PRIMARY KEY constraint in a table using the DROP clause. For example,

**SQL Server, Oracle** ALTER TABLE Colleges

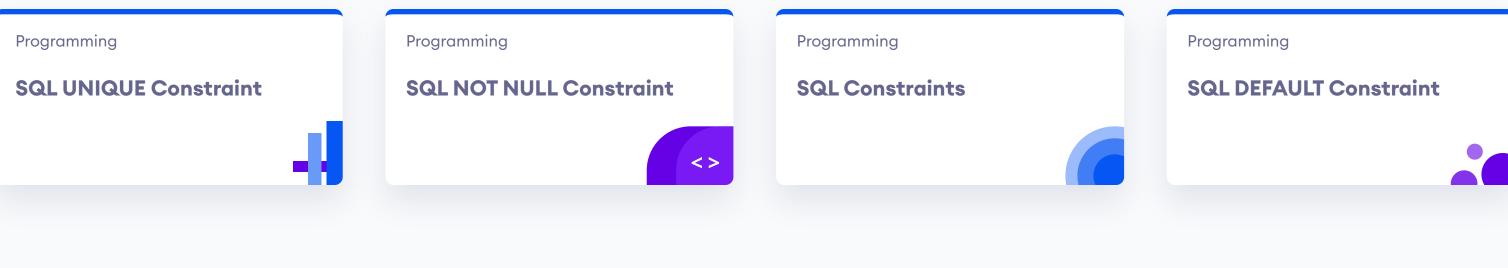
```
DROP CONSTRAINT CollegePK;
MySQL
 ALTER TABLE Colleges
 DROP PRIMARY KEY;
Here, the SQL command removes the PRIMARY KEY constraint from the Colleges table.
```

**Recommended Reading:** 

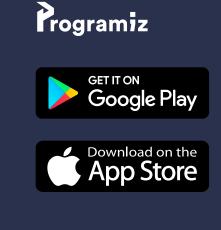
### • SQL FORIEGN KEY • SQL Constraints

- - **Next Tutorial: Previous Tutorial: SQL Foreign Key SQL Unique Constraints** Did you find this article helpful?

## **Related Tutorials** Programming



• •



DSA Tutorial

**SQL** Tutorial

C Examples