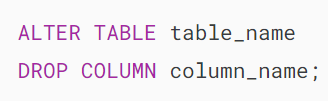
# **MySQL DROP COLUMN**

**Summary**: in this tutorial, you will learn how to drop a column from a table using the MySQL DROP COLUMN statement.

## **Introduction to MySQL DROP COLUMN statement**

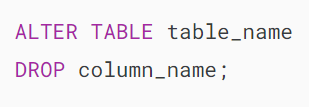
In some situations, you want to remove one or more columns from a table. In such cases, you use the following ALTER TABLE DROP COLUMN statement:



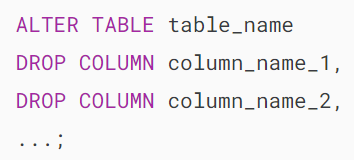
In this syntax:

* First, specify the name of the table that contains the column which you want to drop after the ALTER TABLE keywords.
* Second, specify the name of the column that you want to drop in theDROP COLUMN clause.

Note that the keyword COLUMN keyword in the DROP COLUMN clause is optional so you can use the shorter statement as follows:



To remove multiple columns from a table using a single ALTER TABLE statement, you use the following syntax:

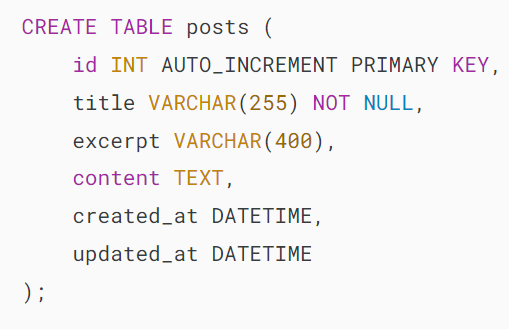


There are some important points you should remember before removing a column from a table:

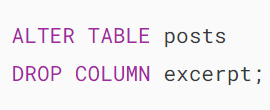
* Removing a column from a table makes all database objects such as [stored procedures](https://www.mysqltutorial.org/mysql-stored-procedure/), [views](https://www.mysqltutorial.org/mysql-views/), and [triggers](https://www.mysqltutorial.org/mysql-triggers/) that referencing the dropped column invalid. For example, you may have a stored procedure that refers to a column. When you remove the column, the stored procedure becomes invalid. To fix it, you have to manually change the stored procedure’s code.
* The code from other applications that depends on the dropped column must be also changed, which takes time and efforts.
* Dropping a column from a large table can impact the performance of the database during the removal time.

## **MySQL DROP COLUMN examples**

First,  [create a table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) named posts for the demonstration.

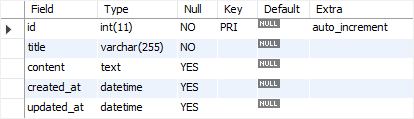


Next, use the ALTER TABLE DROP COLUMN statement to remove the excerpt column:

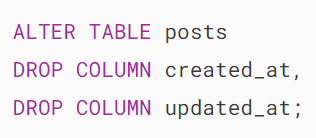


Then, view the table structure using the DESCRIBE statement:



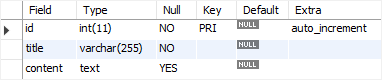


After that, use the ALTER TABLE DROP COLUMN statement to drop the created\_at and updated\_at columns:



Finally, use the DESCRIBE statement to verify the removal:

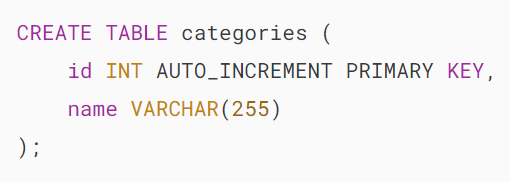




### **MySQL drop a column which is a foreign key example**

If you remove the column that is a [foreign key](https://www.mysqltutorial.org/mysql-basics/mysql-foreign-key/), MySQL will issue an error. Consider the following example.

First, [create a table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) named categories:



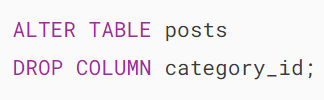
Second, [add a column](https://www.mysqltutorial.org/mysql-basics/mysql-add-column/) named category\_id to the posts table.



Third, make the category\_id column as a foreign key column of that references to the id column of the categories table.



Fourth, drop the category\_id column from the posts table.



MySQL issued an error message:



To avoid this error, you must remove the foreign key constraint before dropping the column.

In this tutorial, we have shown you how to use MySQL DROP COLUMN statement to remove one or more columns from a table.

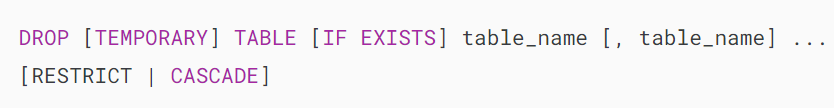
# **MySQL DROP TABLE**

**Summary**: in this tutorial, you will learn how to use the MySQL DROP TABLE statement to drop a table from the database.

## **MySQL DROP TABLE statement syntax**

To remove existing tables, you use the MySQL DROP TABLE statement.

Here is the basic syntax of the DROP TABLE statement:



The DROP TABLE statement removes a table and its data permanently from the database. In MySQL, you can also remove multiple tables using a single DROP TABLE statement, each table is separated by a comma (,).

The TEMPORARY option allows you to remove [temporary tables](https://www.mysqltutorial.org/mysql-basics/mysql-temporary-table/) only. It ensures that you do not accidentally remove non-temporary tables.

The IF EXISTS option conditionally drop a table only if it exists. If you drop a non-existing table with the IF EXISTS option, MySQL generates a NOTE, which can be retrieved using the [SHOW WARNINGS](https://www.mysqltutorial.org/mysql-stored-procedure/mysql-show-warnings/) statement.

Note that the DROP TABLE statement only drops tables. It doesn’t remove specific user privileges associated with the tables. Therefore, if you create a table with the same name as the dropped one, MySQL will apply the existing privileges to the new table, which may pose a security risk.

The RESTRICT and CASCADE  options are reserved for the future versions of MySQL.

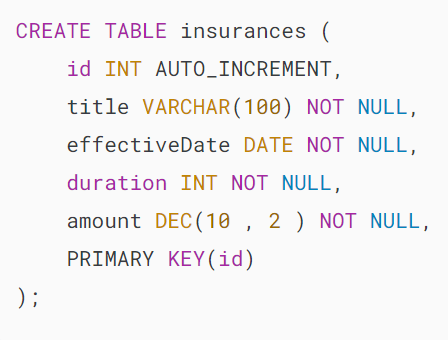
To execute the DROP TABLE statement, you must have DROP privileges for the table that you want to remove.

## **MySQL DROP TABLE examples**

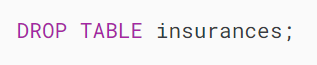
Let’s take some examples of using the DROP TABLE statement.

### **1) Using MySQL DROP TABLE to drop a single table example**

First, create a table named insurances for testing purposes:

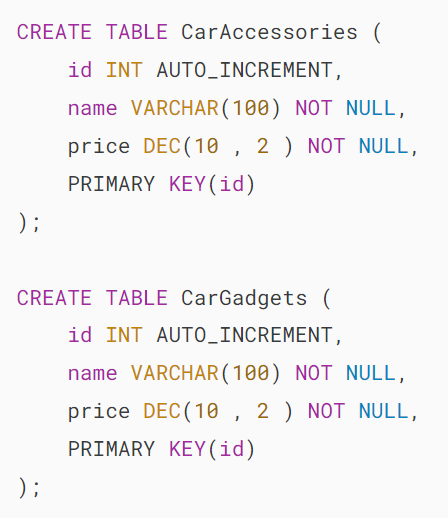


Second, use the DROP TABLE to delete the insurances table:



### **2) Using MySQL DROP TABLE to drop multiple tables**

First, create two tables named CarAccessories and CarGadgets:

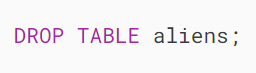


Second, use the DROP TABLE statement to drop the two tables:

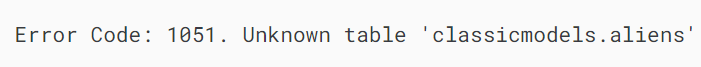
### 

### **3) Using MySQL DROP TABLE to drop a non-existing table**

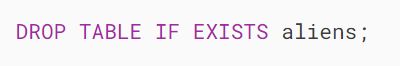
This statement attempts to drop a non-existing table:



MySQL issued the following error:



However, if you use the IF EXISTS option in the DROP TABLE statement:



MySQL issued a warning instead:



To show the warning, you can use the SHOW WARNINGS statement:



MySQL DROP TABLE removes non existing table

## **MySQL DROP TABLE based on a pattern**

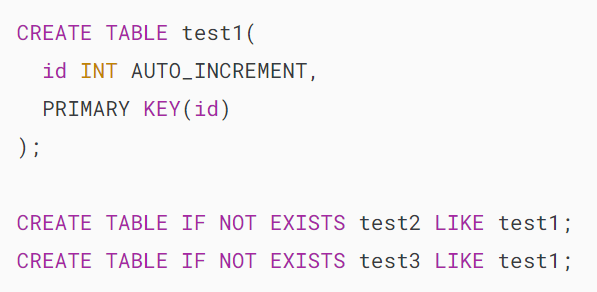
Suppose that you have many tables whose names start with test in your database and you want to remove all of them using a single DROP TABLE statement.

Unfortunately, MySQL does not have the DROP TABLE LIKE statement that can remove tables based on pattern matching:



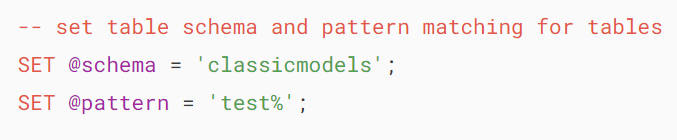
However, there are some workarounds. We will discuss one of them here for your reference.

First, create three tables test1,test2, test4 for demonstration:

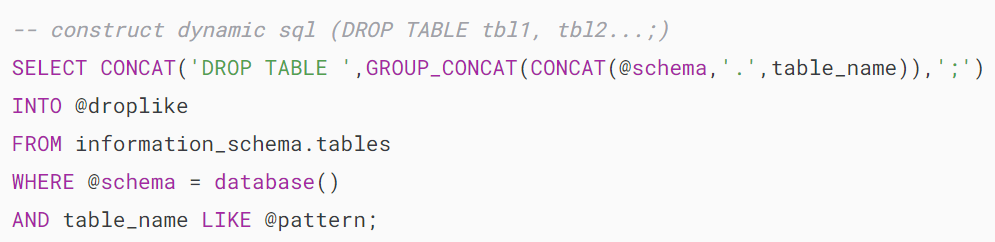


Suppose you want to remove all test\* tables.

Second, declare two variables that accept database schema and a pattern that you want to the tables to match:

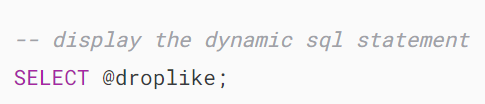


Third, construct a dynamic DROP TABLE statement:



The query instructs MySQL to go to the information\_schema  table, which contains information on all tables in all databases, and concatenates all tables in the database @schema ( classicmodels ) that matches the pattern @pattern ( *test%* ) with the prefix DROP TABLE . The [GROUP\_CONCAT](https://www.mysqltutorial.org/mysql-group_concat/) function creates a comma-separated list of tables.

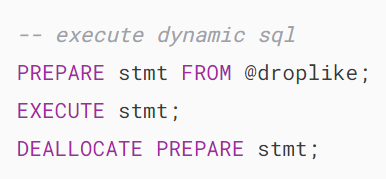
Fourth, display the dynamic SQL to verify if it works correctly:



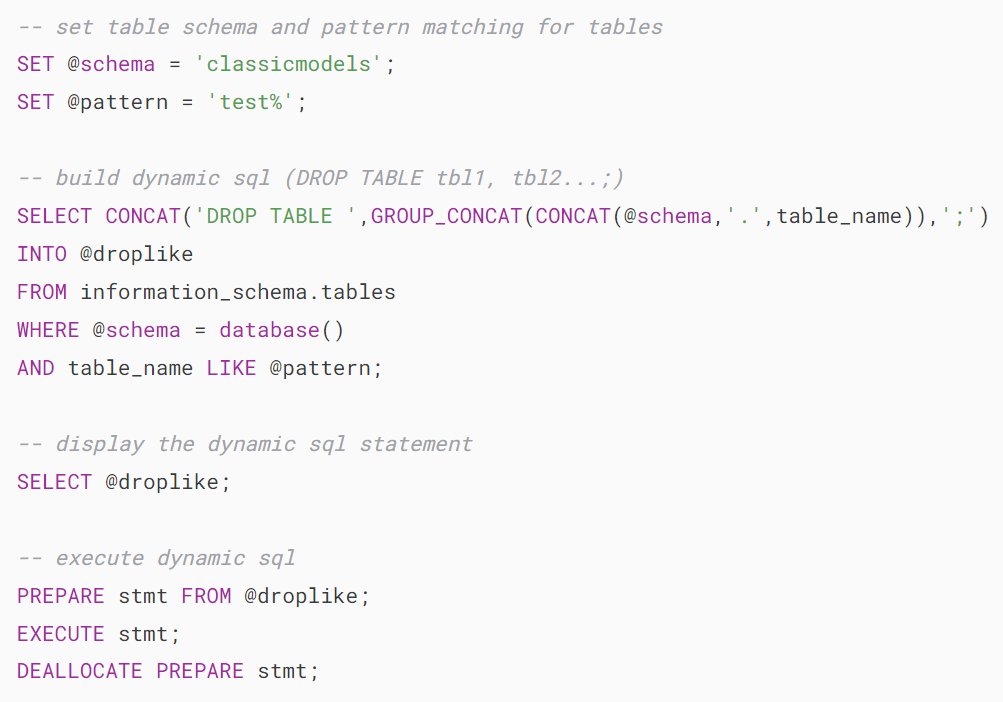
mysql drop table like

As you can see, the output is what we expected.

Fifth, execute the statement using a [prepared statement](https://www.mysqltutorial.org/mysql-stored-procedure/mysql-prepared-statement/) as shown in the following query:



Putting it all together.



So if you want to drop multiple tables that have a specific pattern in a database, you just use the script above to save time.

All you need to do is replace the pattern and the database schema in @pattern and @schema variables.

If you often have to deal with this task, you can develop a [stored procedure](https://www.mysqltutorial.org/mysql-stored-procedure/) based on the script and reuse this stored procedure.

In this tutorial, you have learned how to use the MySQL DROP TABLE statement to remove existing tables from a database.

# **MySQL Temporary Tables**

**Summary**: in this tutorial, we will discuss **MySQL temporary tables** and show you how to create, use, and drop temporary tables.

## **Introduction to MySQL temporary tables**

In MySQL, a temporary table is a special type of table that allows you to store a temporary result set, which you can reuse several times in a single session.

A temporary table is handy when it is impossible or expensive to query data that requires a single [SELECT](https://www.mysqltutorial.org/mysql-basics/mysql-select-from/) statement. In such cases, you can use a temporary table to store the immediate result and use another query to process it.

A MySQL temporary table has the following features:

* A temporary table is created by using CREATE TEMPORARY TABLE statement. Notice that the keyword TEMPORARY is added between the CREATE and TABLE keywords.
* MySQL removes the temporary table automatically when the session ends or the connection is terminated. Also, you can use the  [DROP TABLE](https://www.mysqltutorial.org/mysql-drop-table) statement to remove a temporary table explicitly when you are no longer using it.
* A temporary table is only available and accessible to the client that creates it. Different clients can create temporary tables with the same name without causing errors because only the client that creates the temporary table can see it. However, in the same session, two temporary tables cannot share the same name.
* A temporary table can have the same name as a regular table in a database. For example, if you create a temporary table named employees in the [sample database](https://www.mysqltutorial.org/getting-started-with-mysql/mysql-sample-database/), the existing employees table becomes inaccessible. Every query you issue against the employees table is now referring to the temporary table employees. When you drop the employees temporary table, the regular employees table is available and accessible.

Even though a temporary table can have the same name as a regular table, it is not recommended. Because this may lead to confusion and potentially cause an unexpected data loss.

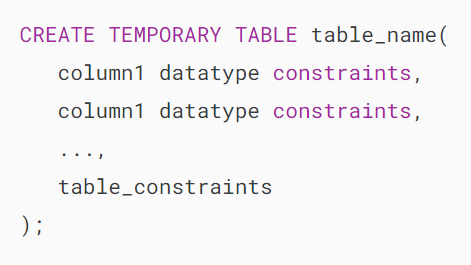
For example, if the connection to the database server is lost and you reconnect to the server automatically, you cannot differentiate between the temporary table and the regular one.

Then, you may issue a DROP TABLE  statement to remove the permanent table instead of the temporary table, which is not expected.

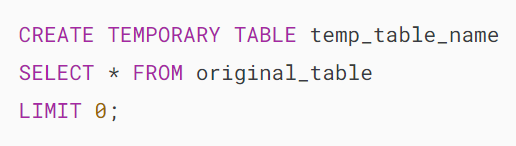
To avoid this issue, you can use the DROP TEMPORARY TABLE statement to drop a temporary table instead of the DROP TABLE statement

## **MySQL CREATE TEMPORARY TABLE statement**

The syntax of the CREATE TEMPORARY TABLE statement is similar to the syntax of the CREATE TABLE statement except for the TEMPORARY keyword:

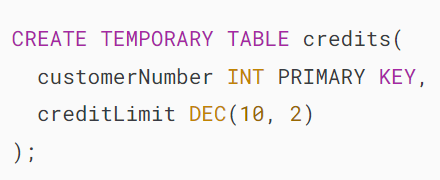


To create a temporary table whose structure is based on an existing table, you cannot use the CREATE TEMPORARY TABLE ... LIKE statement. Instead, you use the following syntax:

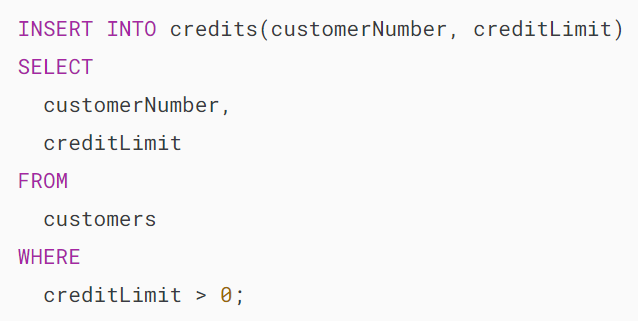


### **1) Creating a temporary table example**

First, create a new temporary table called credits that stores customers’ credits:

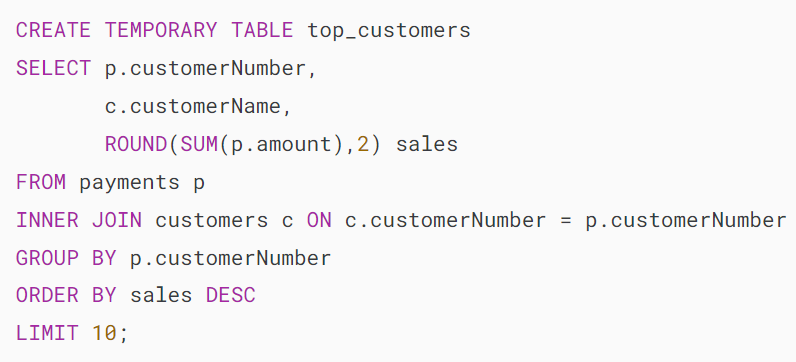


Then, insert rows from the customers table into the temporary table credits:

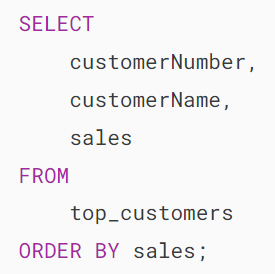


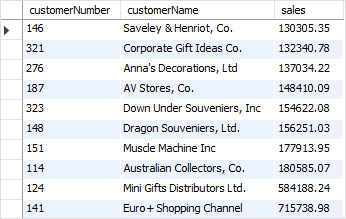
### **2) Creating a temporary table whose structure is based on a query example**

The following example creates a temporary table that stores the top 10 customers by revenue. The structure of the temporary table is derived from a SELECT statement:



Now, you can query data from the top\_customers temporary table like querying from a permanent table:





## **Dropping a temporary table**

You can use the DROP TABLE statement to remove temporary tables however it is good practice to add the TEMPORARY keyword as follows:



The DROP TEMPORARY TABLE  statement removes a temporary table only, not a regular table. It helps you avoid the risk of mistakenly dropping a regular table with the same name as the temporary table.

For example, to remove the top\_customers temporary table, you use the following statement:



Notice that if you try to remove a regular table with the DROP TEMPORARY TABLE statement, you will get an error message saying that the table that you are trying to drop is unknown.

If you develop an application that uses connection pooling or persistent connections, it is not guaranteed that the temporary tables are removed automatically when your application is terminated.

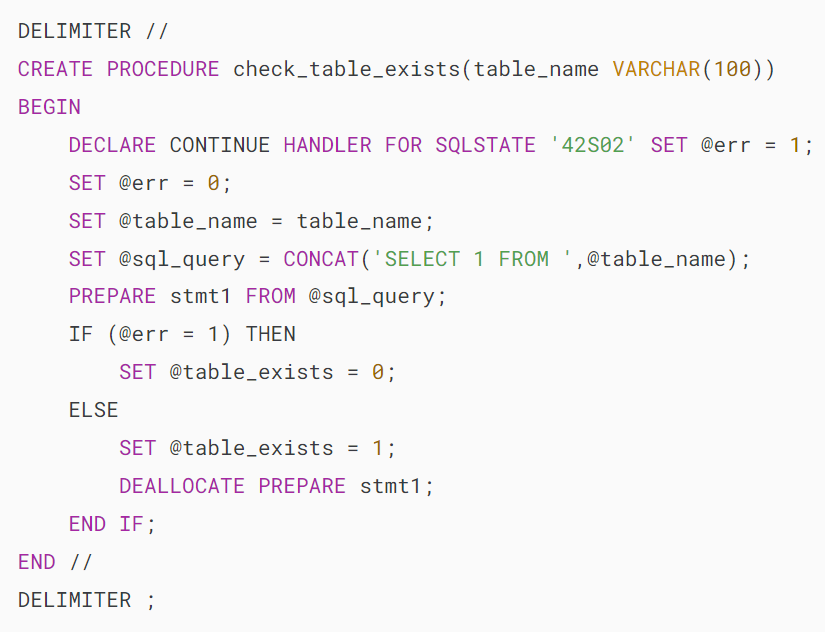
Because the database connection that the application uses may be still open and placed in a connection pool for other clients to reuse later.

Therefore, it is a good practice to always remove the temporary tables whenever you are no longer use them.

## **Checking if a temporary table exists**

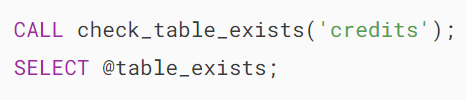
MySQL does not provide a function or statement to directly check if a temporary table exists.

However, you can create a [stored procedure](https://www.mysqltutorial.org/mysql-stored-procedure/) that checks if a temporary table exists or not as follows:



In this procedure, we try to select data from a temporary table. If the temporary table exists, the @table\_exists variable is set to 1, otherwise, it is set to 0.

This statement calls the check\_table\_exists stored procedure to check if the temporary table credits exists:



## **Summary**

* MySQL automatically deletes all temporary tables once the session is ended.
* Use the CREATE TEMPORARY TABLE statement to create a temporary table.
* Use the DROP TEMPORARY TABLE statement to drop a temporary table.

# **How to Use the MySQL Generated Columns**

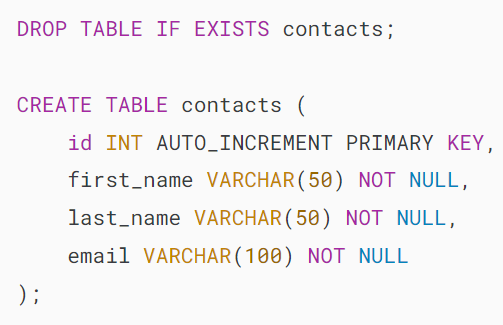
**Summary**: in this tutorial, you will learn how to use the MySQL generated columns to store data computed from an expression or other columns.

## **Introduction to MySQL Generated Column**

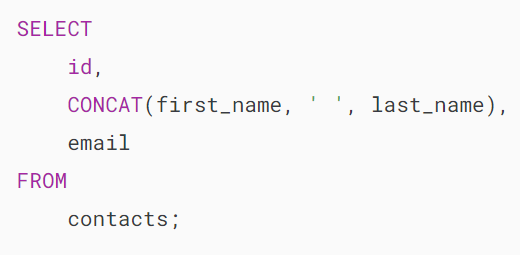
When you create a new table, you specify the table columns in the [CREATE TABLE](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) statement. Then, you use the [INSERT](https://www.mysqltutorial.org/mysql-basics/mysql-insert/), [UPDATE](https://www.mysqltutorial.org/mysql-basics/mysql-update/), and [DELETE](https://www.mysqltutorial.org/mysql-basics/mysql-delete/) statements to modify directly the data in the table columns.

MySQL 5.7 introduced a new feature called the generated column. Columns are generated because the data in these columns are computed based on predefined expressions.

For example, you have the contacts with the following structure:

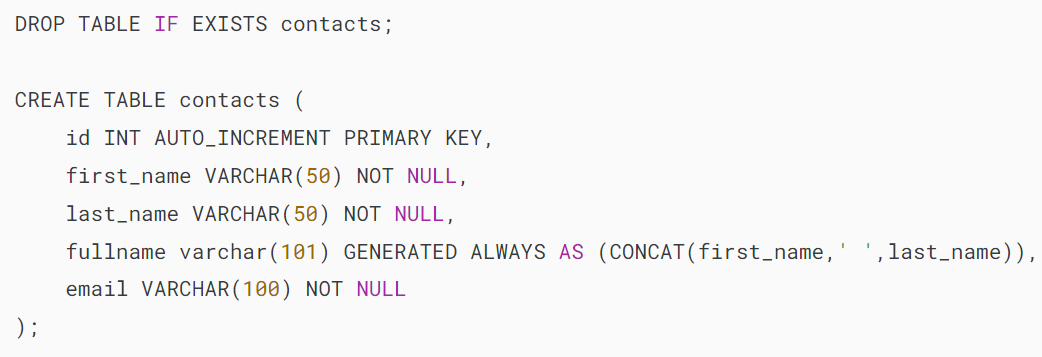


To get the full name of a contact, you use the [CONCAT()](https://www.mysqltutorial.org/mysql-string-functions/mysql-concat/) function as follows:



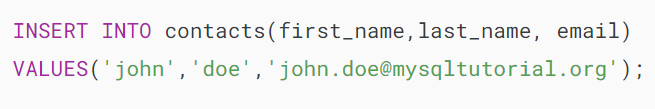
This is not the most beautiful query yet.

By using a Generated Column, you can recreate the contacts table as follows:

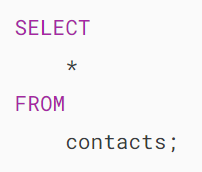


The GENERATED ALWAYS as (expression) is the syntax for creating a generated column.

To test the fullname column, you [insert](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) a row into the contacts table.



Now, you can [query data](https://www.mysqltutorial.org/mysql-basics/mysql-select-from/) from the contacts table.

MySQL generated column - example

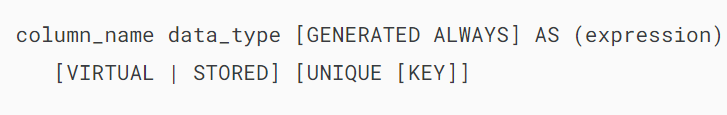
The values in the fullname column are computed on the fly when you query data from the contacts table.

MySQL provides two types of generated columns: stored and virtual. The virtual columns are calculated on the fly each time data is read whereas the stored columns are calculated and stored physically when the data is updated.

Based on this definition, the  fullname column that in the example above is a virtual column.

## **Syntax for MySQL Generated Column**

The syntax for defining a generated column is as follows:



First, specify the column name and its data type.

Next, add the GENERATED ALWAYS clause to indicate that the column is a generated column.

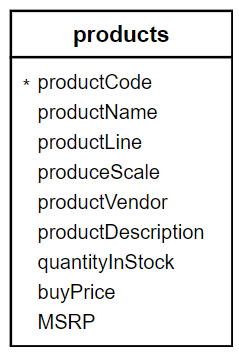
Then, indicate the type of the generated column by using the corresponding option: VIRTUAL or STORED. By default, MySQL uses VIRTUAL if you don’t specify explicitly the type of the generated column.

After that, specify the expression within the braces after the AS keyword. The expression can contain literals, and built-in functions with no parameters, operators, or references to any column within the same table. If you use a function, it must be scalar and deterministic.

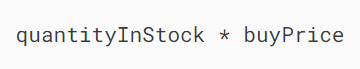
Finally, if the generated column is stored, you can define a [unique constraint](https://www.mysqltutorial.org/mysql-basics/mysql-unique-constraint/) for it.

## **MySQL stored column example**

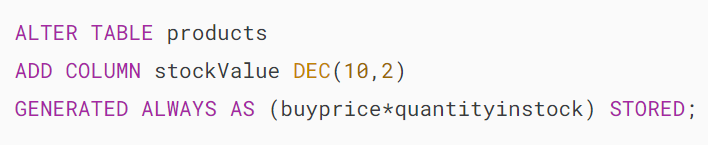
Let’s look at the products table in the [sample database](https://www.mysqltutorial.org/getting-started-with-mysql/mysql-sample-database/):



The data from quantityInStock and buyPrice columns allow us to calculate the stock’s value per SKU using the following expression:

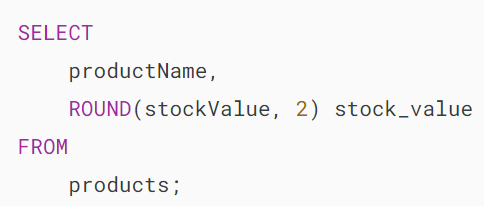


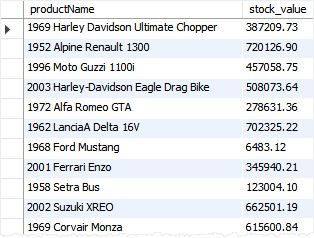
However, we can add a stored generated column named stock\_value to the products table using the following [ALTER TABLE ...ADD COLUMN](https://www.mysqltutorial.org/mysql-basics/mysql-add-column/) statement:



Typically, the ALTER TABLE statement requires a full table rebuild, therefore, it is time-consuming if you change the big tables. However, this is not the case for the virtual column.

Now, we can query the stock value directly from the products table.





## **Summary**

* Use a MySQL Generated column to store data computed from an expression or other columns.