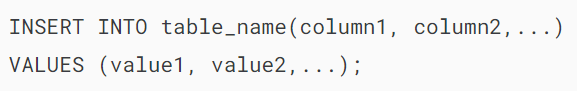
# **MySQL INSERT**

**Summary**: in this tutorial, you will learn how to use the MySQL INSERT statement to insert one or more rows into a table.

## **Introduction to the MySQL INSERT statement**

The INSERT statement allows you to insert one or more rows into a table. The following illustrates the syntax of the INSERT statement:



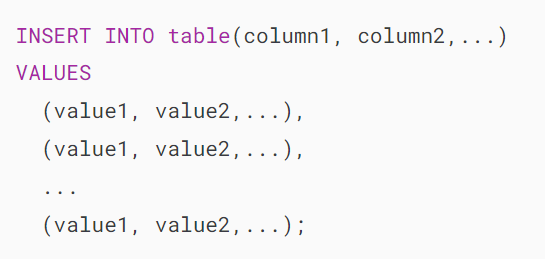
In this syntax,

* First, specify the table name and a list of comma-separated columns inside parentheses after the INSERT INTO clause.
* Then, put a comma-separated list of values of the corresponding columns inside the parentheses following the VALUES keyword.

When using the INSERT statement, you need to ensure that the number of columns matches the number of values.

Additionally, you need to specify that the positions of columns correspond precisely to the positions of their corresponding values.

To [insert multiple rows](https://www.mysqltutorial.org/mysql-basics/mysql-insert-multiple-rows/) into a table using a single INSERT statement, you use the following syntax:



In this syntax, rows are separated by commas in the VALUES clause.

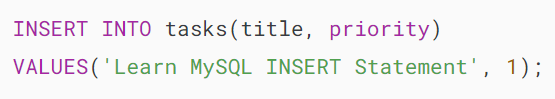
## **MySQL INSERT statement examples**

Let’s [create a new table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) called tasks for practicing the INSERT statement:

### 

### **1) Basic MySQL INSERT statement example**

The following statement inserts a new row into the tasks table:



Output:



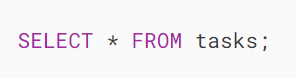
The output indicates that the statement inserted one row into the tasks table successfully.

In this example, we specified values for the title and priority columns only. For other columns, MySQL uses the default values.

The task\_id column is an [AUTO\_INCREMENT](https://www.mysqltutorial.org/mysql-basics/mysql-auto_increment/) column, meaning that MySQL generates a sequential integer whenever a row is inserted into the table.

The start\_date, due\_date, and description columns use NULL as the default value. Therefore, MySQL insert [NULL](https://www.mysqltutorial.org/mysql-basics/mysql-null/) into these columns if you don’t specify their values in the INSERT statement.

The following retrieves data from the tasks table:



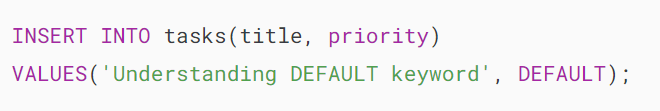
### 

### **2) Inserting rows using default value example**

If you want to insert a default value into a column, you have two ways:

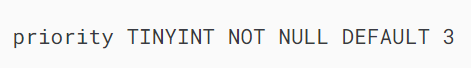
* Ignore both the column name and value in the INSERT statement.
* Specify the column name in the INSERT INTO clause and use the DEFAULT keyword in the VALUES clause.

The following example demonstrates the second way:

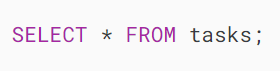


In this example, we specified the value for title and priority columns.

Because the default value for the column priority is 3 as declared in the table definition, the statement inserts number 3 into the priority column.



The following statement returns the contents of the tasks table after the insert:



### 

### **3) Inserting dates into the table example**

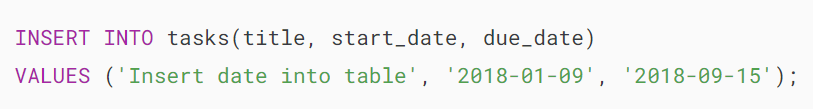
To insert a literal date value into a column, you use the following format:

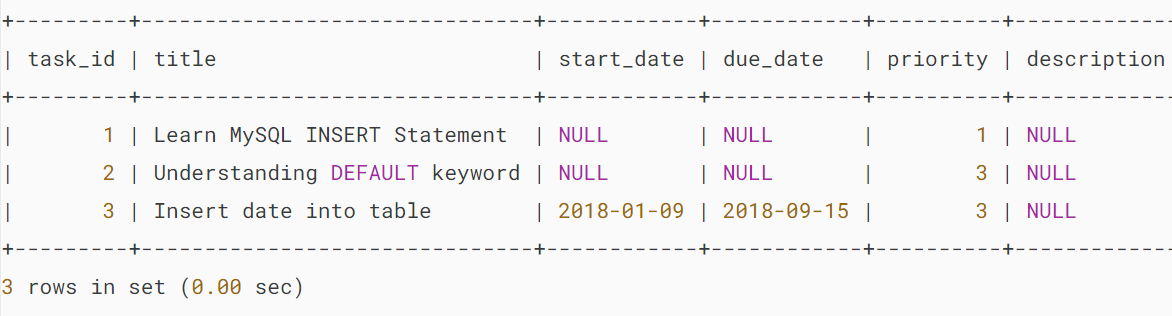


In this format:

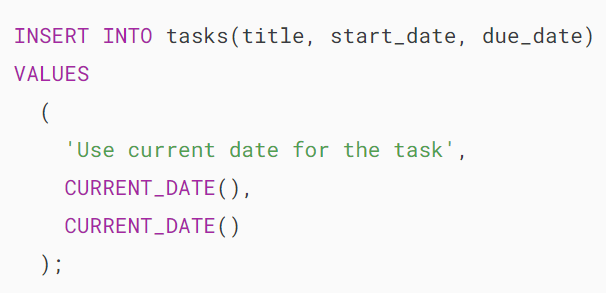
* YYYY represents a four-digit year e.g., 2018.
* MM represents a two-digit month e.g., 01, 02, and 12.
* DD represents a two-digit day e.g., 01, 02, 30.

The following statement inserts a new row to the tasks table with the start and due date values:





It is possible to use expressions in the VALUES clause. For example, the following statement adds a new task using the current date for the start date and due date columns:



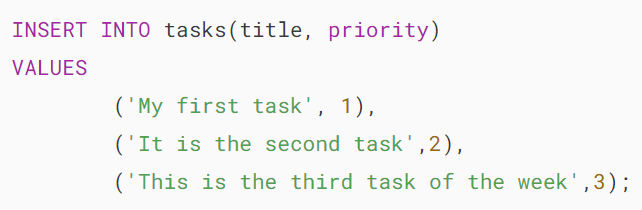
In this example, we used the CURRENT\_DATE() function as the values for the start\_date and due\_date columns. Note that the [CURRENT\_DATE()](https://www.mysqltutorial.org/mysql-date-functions/mysql-curdate/) function is a [date function](https://www.mysqltutorial.org/mysql-date-functions/) that returns the current system date.

Here are the contents of the tasks table after insert:

### 

### **4) Inserting multiple rows example**

The following statement inserts three rows into the tasks table:



In this example, each row of data is specified as a list of values in the VALUES clause.

MySQL returns the following message:



It means that the three rows have been inserted successfully with no duplicates or warnings.

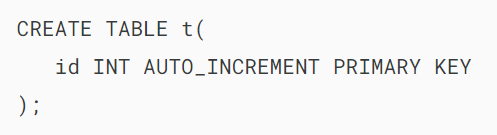


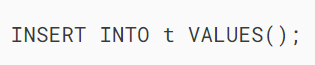
The table tasks has the following data:

### 

### **5) Dealing with auto-increment column**

Suppose you have a table that has an auto-increment column:

The following statement insert a new row into the t table, which uses the generated value:



In this statement, we don’t specify any column after the table name and any values inside the VALUES() clause.

Here’s the contents of the t table:



Output:

## 

## **Summary**

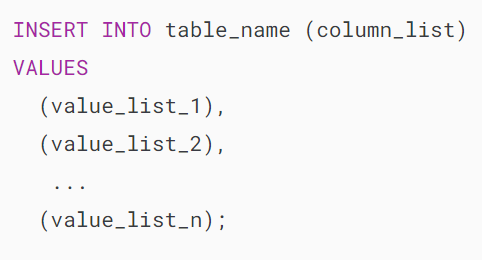
* Use the INSERT statement to insert one or more rows into a table.

# **MySQL Insert Multiple Rows**

**Summary**: in this tutorial, you will learn how to use a single MySQL INSERT statement to insert multiple rows into a table.

## **MySQL INSERT multiple rows statement**

To insert multiple rows into a table, you use the following form of the INSERT statement:



In this syntax:

* First, specify the name of the table where you want to insert multiple rows after the INSERT INTO keywords.
* Second, list the columns in the table into which you want to insert data. This column list is optional, but if provided, you should provide corresponding values for each column in the VALUES value.
* Third, specify a comma-separated list of row data after the VALUES keyword. Each item on the list represents a row. The number of values in each item must be the same as the number of columns in the column\_list.

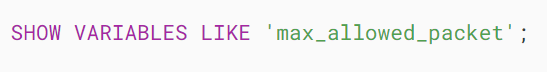
Note that to insert rows from a query into a table, you use the [INSERT INTO … SELECT](https://www.mysqltutorial.org/mysql-basics/mysql-insert-into-select/) statement.

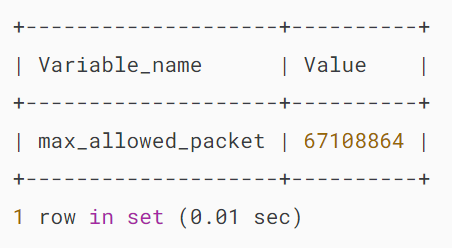
### **MySQL INSERT multiple rows limit**

In theory, you can insert any number of rows using a single INSERT statement.

However, when the MySQL server receives an INSERT statement whose size is bigger than the value specified by the max\_allowed\_packet option, it issues a packet too large error and terminates the connection.

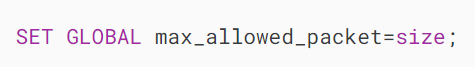
This statement shows the current value of the max\_allowed\_packet variable:





The number in the Value column is the number of bytes. Note that the value in your database server may be different.

To set a new value for the max\_allowed\_packet variable, you use the SET GLOBAL statement:



In this statement, the size is an integer that represents the number of the maximum allowed packet size in bytes.

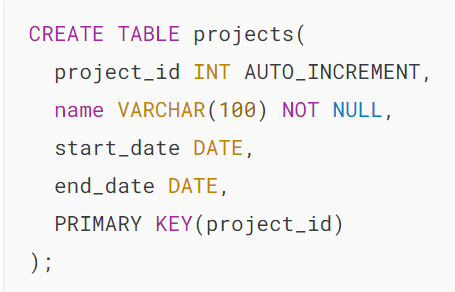
Notice that the max\_allowed\_packet does not impact the [INSERT INTO .. SELECT](https://www.mysqltutorial.org/mysql-basics/mysql-insert-into-select/) statement. The INSERT INTO .. SELECT statement can insert as many rows as you want.

## **MySQL INSERT multiple rows examples**

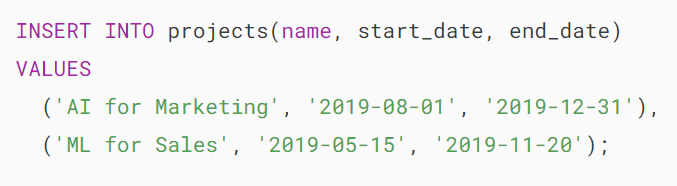
Let’s take some examples of using the INSERT multiple rows statement.

### **1) Insert multiple rows into a table**

First, [create a new table](https://www.mysqltutorial.org/mysql-basics/mysql-create-database/) called projects for the demonstration:



Second, insert two rows into the projects table using the INSERT multiple rows statement:

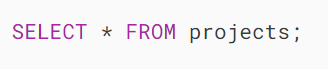


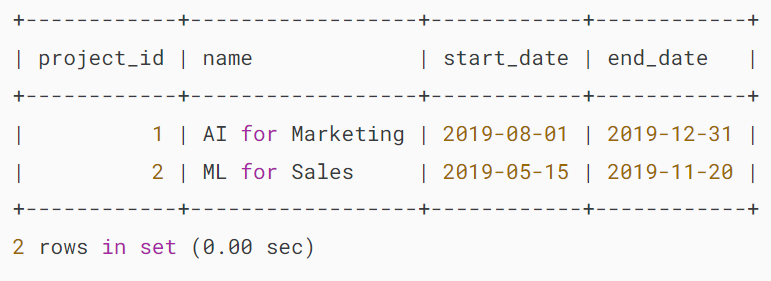
MySQL issued the following message:



The output indicates that the statement has inserted two rows into the projects table successfully.

Third, retrieve data from the projects table to verify the inserts:

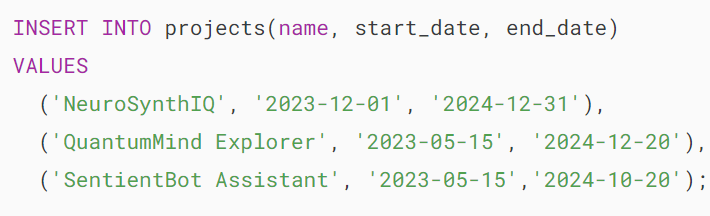


MySQL Insert multiple rows example

### **2) Using the LAST\_INSERT\_ID() function**

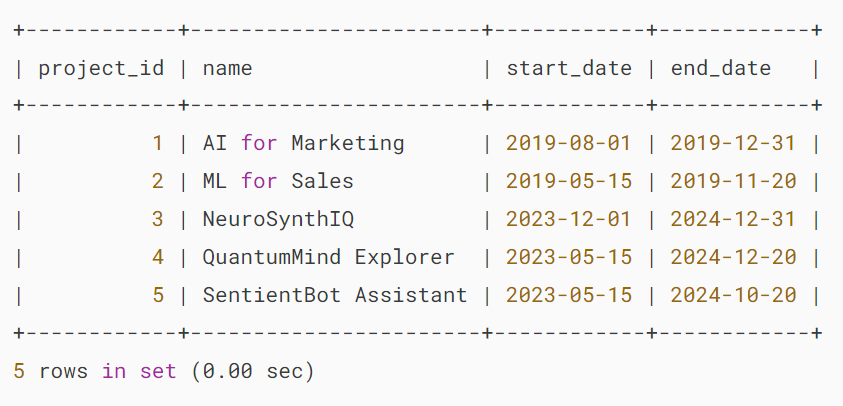
When you insert multiple rows and use the [LAST\_INSERT\_ID()](https://www.mysqltutorial.org/mysql-functions/mysql-last_insert_id/) function to get the last inserted id of an [AUTO\_INCREMENT](https://www.mysqltutorial.org/mysql-basics/mysql-auto_increment/) column, you will get the id of the first inserted row, not the id of the last inserted row. For example:

First, insert three rows into the projects table:



Second, retrieve data from the projects table:

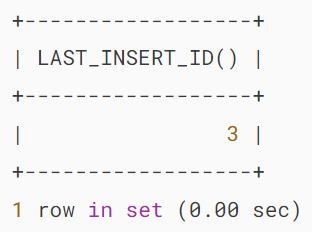




Third, get the last inserted id:



Output:



The output shows that the LAST\_INSERT\_ID() returns the id of the first row in the three rows, not the id of the last row.

## **Summary**

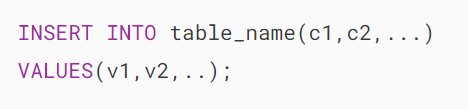
* Use the MySQL INSERT statement to insert multiple rows into a table.

# **MySQL INSERT INTO SELECT**

**Summary**: in this tutorial, you will learn how to use the MySQL INSERT INTO SELECT statement to insert data into a table, where data comes from the result of a SELECT statement.

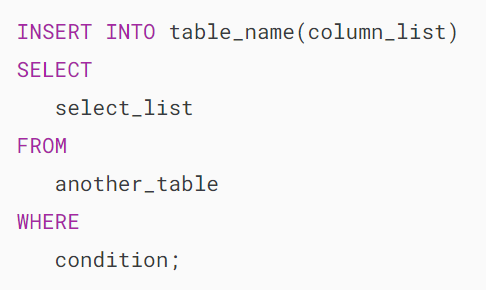
## **Introduction to MySQL INSERT INTO SELECT statement**

The INSERT statement allows you to insert one or more rows into a table with a list of column values specified in the VALUES clause:



Besides using row values in the VALUES clause, you can use the result of a [SELECT](https://www.mysqltutorial.org/mysql-basics/mysql-select-from/) statement as the data source for the INSERT statement.

The following illustrates the syntax of the INSERT INTO SELECT statement:



In this syntax, instead of using the VALUES clause, you use a SELECT statement. The SELECT statement may retrieve data from one or more tables.

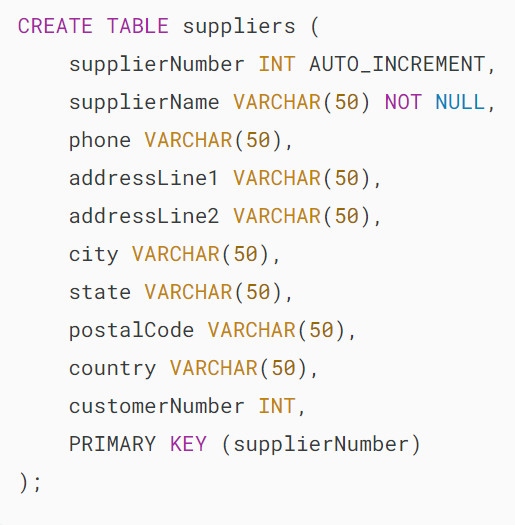
Note that the number of columns in the column\_list and select\_list must be equal.

The INSERT INTO SELECT statement can very useful when you want to copy data from other tables to a table or to summarize data from multiple tables into a table.

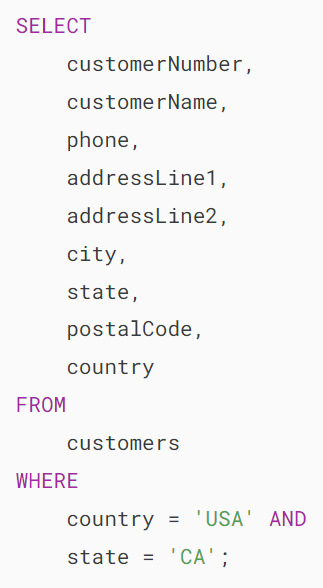
Please note that it’s possible to select rows in a table and insert them into the same table. In other words, the table\_name and another\_table in the INSERT INTO ... SELECT statement can reference the same table.

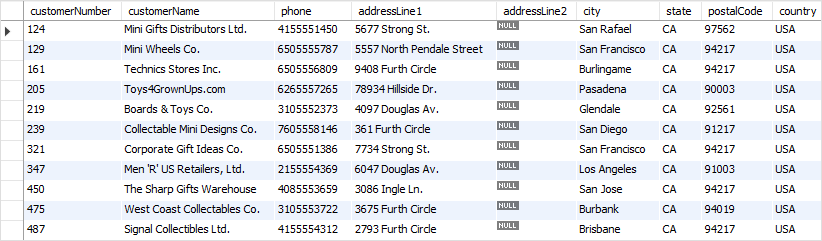
## **MySQL INSERT INTO SELECT example**

First, [create a new table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) called suppliers:



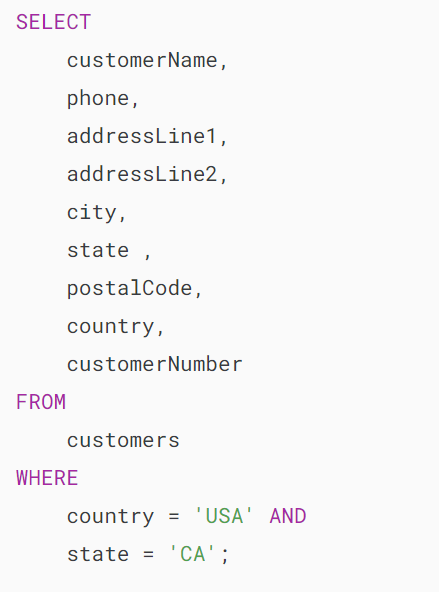
Suppose all customers in California, USA become the company’s suppliers. The following query finds all customers who are located in California, USA:





Second, insert customers who are located in California USA from the customers table into the suppliers table using the INSERT INTO SELECT statement:

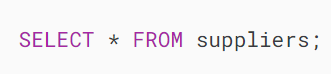




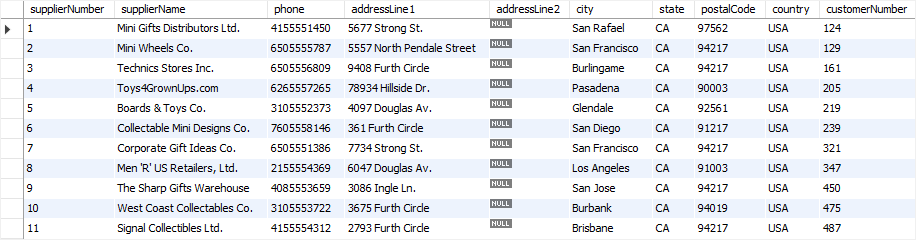
It returned the following message indicating that 11 rows had been inserted successfully.



Third, verify the insert by querying data from the suppliers table:

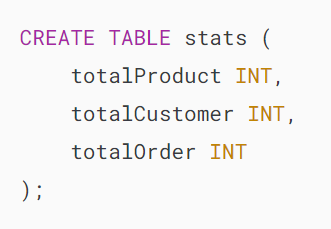


Here is the output:

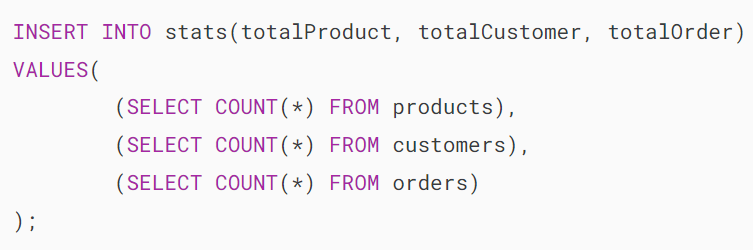


## **Using SELECT statement in the VALUES list**

First, [create a new table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) called stats:



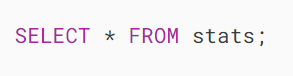
Second, use the [INSERT](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) statement to insert values that come from the SELECT statements:



In this example:

* First, use the SELECT statements with the COUNT() functions to get the total products, employees, and orders.
* Second, use the values returned from the [SELECT](https://www.mysqltutorial.org/mysql-basics/mysql-select-from/) statement in place of values in the VALUES clause of the [INSERT](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) statement.

Third, query data from the table stats:



mysql insert into select in values list

## **Summary**

* Use the MySQL INSERT INTO SELECT statement to insert data into a table from a result set.

# **MySQL INSERT ON DUPLICATE KEY UPDATE Statement**

**Summary**: in this tutorial, you will learn how to use MySQL INSERT ON DUPLICATE KEY UPDATE statement to insert data into a table or update data if a duplicate key violation error occurs.

Note that this tutorial is relevant to MySQL 8.0.19 or later.

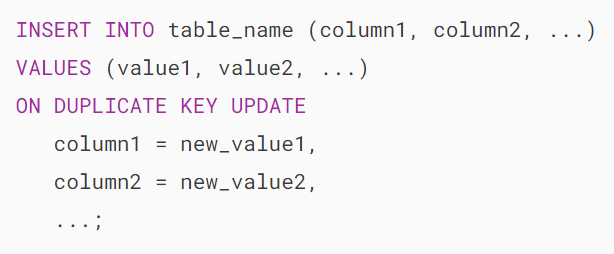
## **Introduction to the MySQL INSERT ON DUPLICATE KEY UPDATE statement**

In MySQL, the INSERT ON DUPLICATE KEY UPDATE statement allows you to [insert new rows into a table](https://www.mysqltutorial.org/mysql-basics/mysql-insert/).

If a duplicate key violation occurs, you can use the INSERT ON DUPLICATE KEY UPDATE statement to update existing rows instead of throwing an error.

This INSERT ON DUPLICATE KEY UPDATE statement is useful when you deal with [unique constraints](https://www.mysqltutorial.org/mysql-basics/mysql-unique-constraint/) or [primary keys](https://www.mysqltutorial.org/mysql-basics/mysql-primary-key/).

Here’s the syntax of INSERT ON DUPLICATE KEY UPDATE statement:



In this syntax:

* INSERT INTO table\_name: Specify the table name after the INSERT INTO keywords.
* (column1, column2, ...): List the columns in the table where you want to insert data
* VALUES(...): Provide values to be inserted into the corresponding columns
* ON DUPLICATE KEY UPDATE: Specify the action to take if a duplicate key violation occurs.
* column1 = new\_value1, column2=new\_value2: Define how existing rows should be updated if a duplicate key is encountered.

The statement returns the number of affected rows based on the action it performs:

* 1 is returned when a new row is inserted.
* 2 is returned when an existing row is updated.
* 0 is returned when no changes are made to an existing row.

### **Row aliases**

MySQL allows you to define a row alias for the inserted row using the AS alias\_name after the VALUES clause.

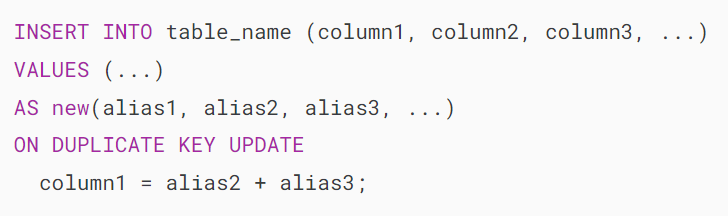
Then, you can use the alias within the ON DUPLICATE KEY UPDATE clause to reference the inserted row’s values.

Here’s the syntax for defining a row alias:

### 

### **Column aliases**

MySQL also allows you to assign aliases to columns to avoid ambiguity, especially with long column names:



## **MySQL INSERT ON DUPLICATE KEY UPDATE examples**

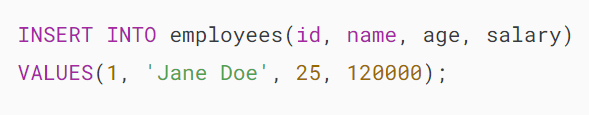
Let’s take some examples of using the INSERT ON DUPLICATE KEY UPDATE to understand how it works.

We’ll create an employees table for the demonstration:

### 

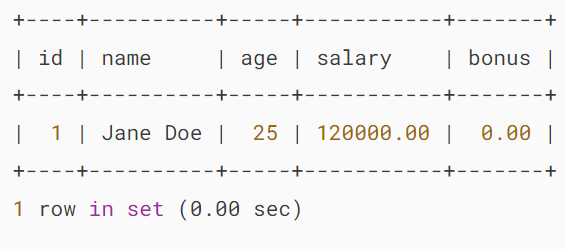
### **1) Using row alias example**

First, [insert a new row](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) into the employees table:

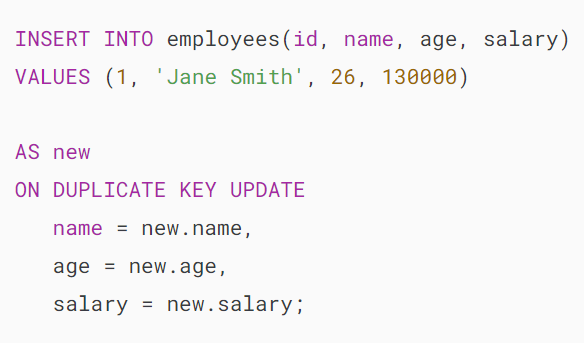


Second, retrieve data from the employees table:





Third, insert a new row into the employees table, or update the existing row if a duplicate key violation occurs:

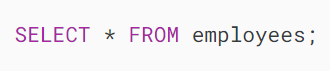




In this example, we use new as the row alias. Because the row with id 1 already exists, the statement updates the row instead of inserting a new one.

In the ON DUPLICATE KEY UPDATE clause, we access the new value specified in the VALUES clause via the row alias and use these new values to update the name, age, and salary column.

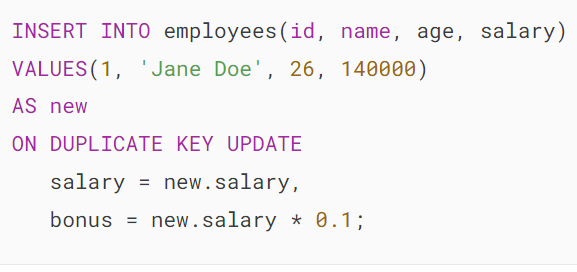
Finally, retrieve the data from the employees table:



### 

### **2) Using MySQL INSERT ON DUPLICATE KEY UPDATE statement to update another column example**

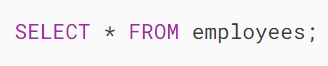
First, insert a new row or update the new salary and bonus column if the row exists:



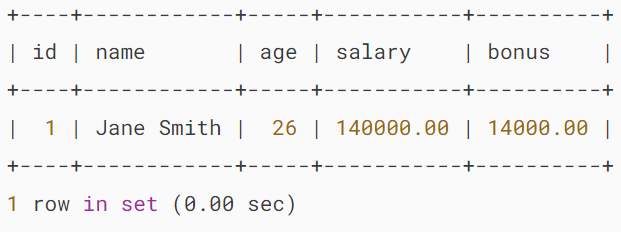
Output:



Second, retrieve data from the employees table:



Output:



## **Summary**

* Use the ON DUPLICATE KEY UPDATE option of the INSERT statement to insert data into a table and update existing data if a duplicate error occurs.

# **MySQL INSERT IGNORE Statement**

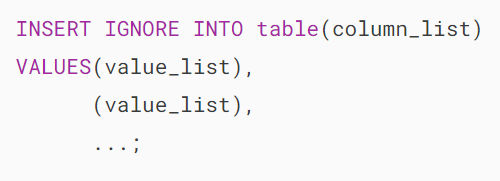
**Summary**: in this tutorial, you will learn how to use the MySQL INSERT IGNORE statement to insert data into a table.

## **Introduction to MySQL INSERT IGNORE statement**

When you use the [INSERT](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) statement to add multiple rows to a table and if an error occurs during the processing, MySQL terminates the statement and returns an error. Consequently, the table remains unchanged with no inserted rows.

The INSERT IGNORE statement allows you to disregard rows containing invalid data that would otherwise trigger an error and insert only rows that contain valid data.

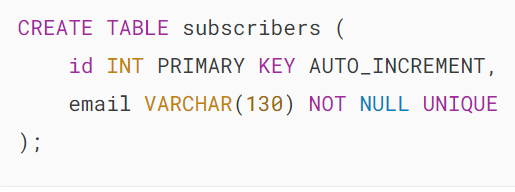
Here’s the basic syntax of the INSERT IGNORE statement:



Note that the IGNORE option is an extension of MySQL to the SQL standard.

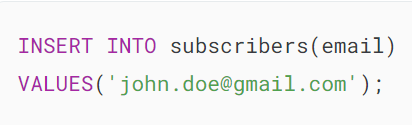
## **MySQL INSERT IGNORE example**

We will [create a new table](https://www.mysqltutorial.org/mysql-create-table/) called subscribers for the demonstration.



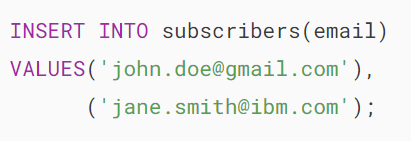
The [UNIQUE](https://www.mysqltutorial.org/mysql-basics/mysql-unique-constraint/) constraint ensures that no duplicate email exists in the email column.

The following statement inserts a new row into the  subscribers table:



It worked as expected.

Let’s execute another statement that inserts two rows into the  subscribers table:



It returns an error.

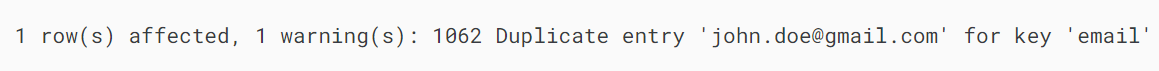


As indicated in the error message, the email john.doe@gmail.com violates the UNIQUE constraint.

However, if you use the INSERT IGNORE statement instead.



MySQL returned a message indicating that one row was inserted and the other row was ignored.



To find the detail of the warning, you can use the SHOW WARNINGS command as shown below:



MySQL INSERT IGNORE - warning

In conclusion, when you use the INSERT IGNORE statement, instead of issuing an error, MySQL issues a warning in case an error occurs.

If you query data from subscribers table, you will find that only one row was inserted and the row that caused the error was not.

MySQL INSERT IGNORE - subscribers table

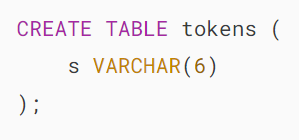
## **MySQL INSERT IGNORE and STRICT mode**

When the [strict mode](https://dev.mysql.com/doc/refman/8.0/en/sql-mode.html#sql-mode-strict) is on, MySQL returns an error and aborts the INSERT statement if you try to insert invalid values into a table.

However, if you use the INSERT IGNORE statement, MySQL will issue a warning instead of an error. In addition, it will try to adjust the values to make them valid before adding the value to the table.

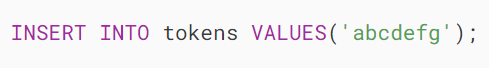
Consider the following example.

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

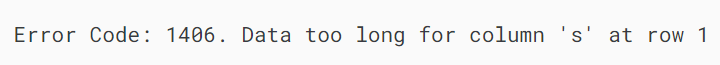


In this table, the column s accepts only strings whose lengths are less than or equal to six.

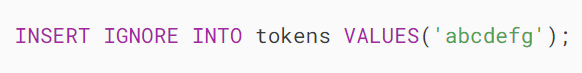
Second, insert a string whose length is seven into the tokens table.



MySQL issued the following error because the strict mode is on.



Third, use the INSERT IGNORE statement to insert the same string.



MySQL truncated data before inserting it into the tokens table. In addition, it issues a warning.

MySQL INSERT IGNORE - strict mode

## **Summary**

* Use the MySQL INSERT IGNORE statement to insert rows into a table and ignore errors for rows that cause errors.

# **MySQL Insert DateTime**

**Summary**: in this tutorial, you will learn how to insert DATETIME values into a table in MySQL database.

## **Defining a DATETIME column**

The [DATETIME](https://www.mysqltutorial.org/mysql-basics/mysql-datetime/) data type is used to store both date and time values. To define a column with the DATETIME data type, you use the following syntax:



In practice, you use the DATETIME columns to store both date and time values such as event times, logging times, and more.

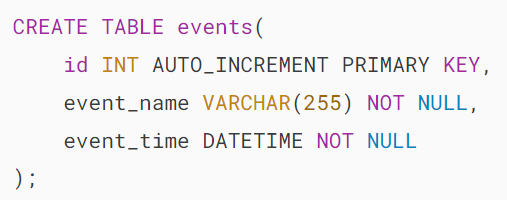
To insert data into the DATETIME columns, you need to ensure that the datetime values are in the 'YYYY-MM-DD HH:MM:SS' format.

If you have datetime values in different formats, you need to format them to match the 'YYYY-MM-DD HH:MM:SS' expected by MySQL.

We’ll show you some examples of inserting data into the DATETIME column.

## **Inserting a datetime value example**

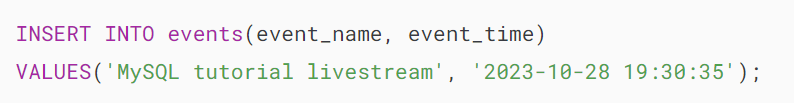
First, [create a table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) called events:



The events table has three columns:

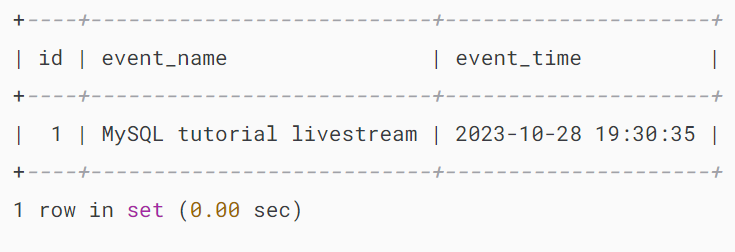
* id: This is an [auto-increment](https://www.mysqltutorial.org/mysql-basics/mysql-auto_increment/) [primary key](https://www.mysqltutorial.org/mysql-basics/mysql-primary-key/) column.
* event\_name: This column stores the name of the event.
* event\_time: This column has the DATETIME data type that stores the event’s date and time.

Second, [insert a new row](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) into the events table and use the datetime format 'YYYY-MM-DD HH:MM:SS':



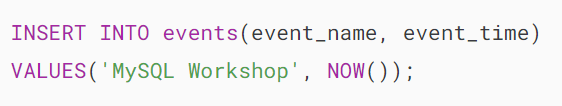
Third, query data from the events table:





## **Inserting the current datetime**

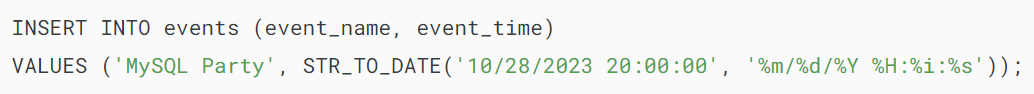
To insert the current date and time into a DATETIME column, you use the NOW() function as the datetime value. For example:



In this example, we use the NOW() function to get the current datetime value and use it to insert it into the event\_time column of the events table.

## **Inserting a datetime string example**

If you want to insert a datetime string into a DATETIME column, you need to use the [STR\_TO\_DATE()](https://www.mysqltutorial.org/mysql-date-functions/mysql-str_to_date/) function to convert it to an expected format. For example:



In this example, we use the STR\_TO\_DATE() function to convert the datetime string '10/28/2023 20:00:00' to the expected format.

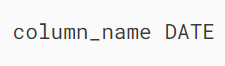
## **Summary**

* Use the datetime value with the format 'YYYY-MM-DD HH:MM:SS' to insert into a DATETIME column.

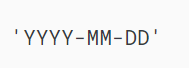
# **MySQL Insert Date**

**Summary**: in this tutorial, you will learn how to insert date values into a table in the MySQL database.

The [DATE](https://www.mysqltutorial.org/mysql-basics/mysql-date/) data type allows you to store the date values. The following shows how to define a column with the DATE data type:



To insert a date value into a column with the DATE data type, you use the following date format:



In this format:

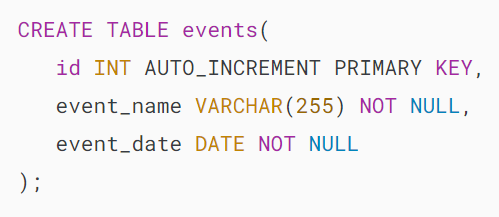
* YYYY is four digits that represent the year e.g., 2023
* MM is two digits that represent the month e.g., 10
* DD is two digits that represent the day e.g., 30.

The valid range for the DATE values is '1000-01-01' to '999-12-31'. If you attempt to insert an invalid date value into a date column, the

We’ll take some examples of inserting date values into the DATE column.

## **Inserting a date value into a date column**

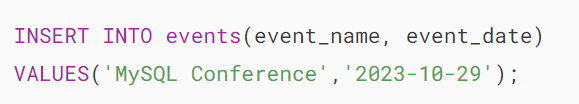
First, [create a table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) called sales\_orders:



The events table has three columns:

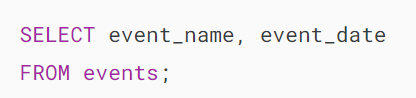
* id is the [auto-increment](https://www.mysqltutorial.org/mysql-basics/mysql-auto_increment/) [primary key](https://www.mysqltutorial.org/mysql-basics/mysql-primary-key/) column that uniquely identifies each event.
* event\_name column stores the name of the event.
* event\_date column stores the date of the event.

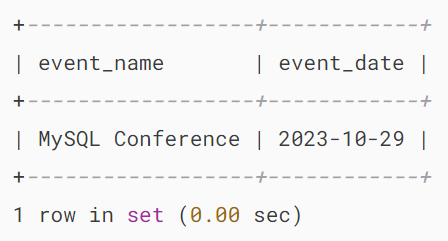
Second, [insert a new row](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) into the events table:



In this statement, we insert a new event into the events table with the event date of '2023-10-29'.

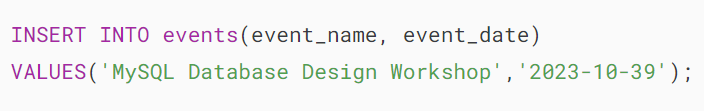
Third, query data from the events table:





The output shows that the mysql client displays the date value as ‘YYYY–MM–DD‘.

If you attempt to insert an invalid date value into a DATE column, MySQL will issue an error. For example:



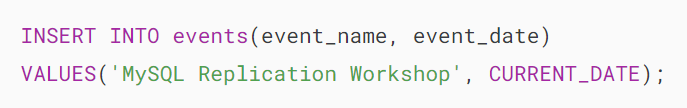
Error:



In this example, we attempt to insert an invalid date value '2023-10-39' into the event\_date column, resulting in an error.

## **Inserting the current date**

To insert the current date of the database server into a date column, you use the [CURRENT\_DATE](https://www.mysqltutorial.org/mysql-date-functions/mysql-curdate/) function. For example:



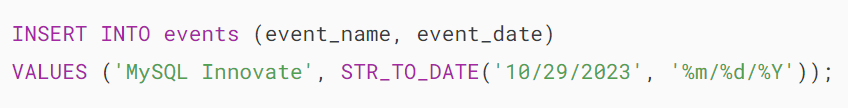
In this example, the CURRENT\_DATE function returns the current date of the MySQL database server, which is then used in the INSERT statement to insert the current date value into the events table.

To insert the current UTC date into a date column, you use the [UTC\_DATE()](https://www.mysqltutorial.org/mysql-date-functions/mysql-utc_date/) function to get the current date in UTC and then insert it. For example:

## 

## **Inserting a date string example**

If you have a date string that is not in the 'YYYY-MM-DD' format and you want to insert it into a date column, you can use the [STR\_TO\_DATE()](https://www.mysqltutorial.org/mysql-date-functions/mysql-str_to_date/) function to convert it into a date first. For example:



In this example, the date value is in the format MM/DD/YYYY, which is not what MySQL expects.

Therefore, we use the STR\_TO\_DATE() function to convert that string into a date value and insert it into the date column.

## **Summary**

* Use the date value in the format 'YYYY-MM-DD' when inserting it into a date column.
* Use the CURRENT\_DATE to insert the current date from the MySQL database server into a date column.
* Use the UTC\_DATE() function to insert the current date in UTC into a date column.
* Use the STR\_TO\_DATE() function to convert a date string into a date before inserting it into a date column.