



[Sign in](#) | [Register](#)
[Logout](#)

[MS SQL](#) [Oracle](#) [DB2](#) [Access](#) [MySQL](#) [PostgreSQL](#) [Sybase](#) [PHP](#) [SQL Etc](#) [SQL Scripts & Samples](#) [Tips](#) [Database Forum](#)
[Database Journal](#) | [DBA Support](#) | [SQLCourse](#) | [SQLCourse2](#)

» [Database Journal Home](#)

» [Database Articles](#)

» [Database Tutorials](#)

[MS SQL](#)

[Oracle](#)

[DB2](#)

[MS Access](#)

[MySQL](#)

» **RESOURCES**

[SQL Scripts & Samples](#)

[Tips](#)

» [Database Forum](#)

» [Slideshows](#)

» [Sitemap](#)

Free Newsletters:

☒ [DatabaseDaily](#)

News Via RSS Feed



Featured Database Articles

[MySQL](#)

Posted May 28, 2010

MySQL Data Manipulation and Query Statements

By [Rob Gravelle](#)

The Data Manipulation Language (DML) includes the major commands, INSERT, UPDATE, DELETE, and SELECT...INTO. Today's article focuses specifically on the uber-useful INSERT, UPDATE, DELETE, and SELECT statements.

Standard SQL is an American National Standards Institute (ANSI) standard, but many database vendors have created their own proprietary extensions and different versions of SQL. The types of SQL statements that we'll be addressing here today are compliant with the ANSI standard, as well as the Data Manipulation Language (DML) SQL standard. The DML includes the following major commands: INSERT, UPDATE, DELETE, and SELECT...INTO. The SELECT without the INTO is considered to be outside of the DML because its results are read-only and does not manipulate data. However, we will include it as part of the larger ANSI standard. Other commands, such as CREATE, ALTER, and DROP conform to another standard called the Data Definition Language (DDL). Today's article will be focusing specifically on the uber-useful INSERT, UPDATE, DELETE, and SELECT statements.

The INSERT Statement

Before we can query the database for information, we must first populate the tables. This is accomplished using the INSERT statement in MySQL, as it does not support the SELECT... INTO DML command. The INSERT statement comes in two flavors: the first doesn't specify the column names where the data will be inserted, only their values; the second form specifies both the column names and the values to be inserted. Here is the syntax for each:

Form 1:

```
INSERT INTO table_name  
VALUES (value1, value2, value3,...)
```

Form 2:

```
INSERT INTO table_name (column1, column2, column3,...)  
VALUES (value1, value2, value3,...)
```

Let's use the INSERT statement to add a row to the following 'employee' table:

id	shop_id	gender	name	salary
1	1	m	Jon Simpson	4500
2	1	f	Barbara Breitenmoser	4700
3	2	f	Kirsten Ruegg	5600
4	3	m	Ralph Teller	5100
5	3	m	Peter Jonson	5200

Here's some code to create and populate the table:

```
#
# Table structure for table 'employees'
#
CREATE TABLE "employees" (
  "id" int(11) NOT NULL AUTO_INCREMENT,
  "shop_id" int(11) DEFAULT NULL,
  "gender" enum('m','f') DEFAULT NULL,
  "name" varchar(32) DEFAULT NULL,
  "salary" int(11) DEFAULT NULL,
  PRIMARY KEY ("id")
) AUTO_INCREMENT=6;
#
# Data for table 'employees'
#
INSERT INTO "employees" ("id", "shop_id", "gender", "name", "salary")
VALUES (1,1,'m','Jon Simpson',4500);
INSERT INTO "employees" ("id", "shop_id", "gender", "name", "salary")
VALUES (2,1,'f','Barbara Breitenmoser',4700);
INSERT INTO "employees" ("id", "shop_id", "gender", "name", "salary")
VALUES (3,2,'f','Kirsten Ruegg',5600);
INSERT INTO "employees" ("id", "shop_id", "gender", "name", "salary")
VALUES (4,3,'m','Ralph Teller',5100);
INSERT INTO "employees" ("id", "shop_id", "gender", "name", "salary")
VALUES (5,3,'m','Peter Jonson',5200);
```

Enter the following to add a sixth row to the table:

```
MySQL>INSERT INTO employees VALUES (6, 3, 'm', 'Al Bundy', 1000);
```

You should see the new row in the table:

```
MySQL>SELECT * FROM employees;
```

id	shop_id	gender	name	salary
1	1	m	Jon Simpson	4500
2	1	f	Barbara Breitenmoser	4700

3	2	f	Kirsten Ruegg	5600
4	3	m	Ralph Teller	5100
5	3	m	Peter Jonson	5200
6	3	m	Al Bundy	1000

If you make a mistake, such as omitting a column, you'll receive an error message:

```
MySQL>INSERT INTO employees VALUES (6, 3, 'm', 'Al Bundy');
```

```
MySQL>SQL Error: Column count doesn't match value count at row 1
```

In this example of Form 2 the 'name' column is mislabeled - a common mistake:

```
MySQL>INSERT INTO employees (id, shop_id, gender, full_name, salary)
-->VALUES (6, 3, 'm', 'Al Bundy', 1000) ;
MySQL>SQL Error: Unknown column 'full_name' in 'field list'
```

The SELECT Statement

Of all the SQL statements, the SELECT is probably the best known. It is also the hardest to master because of all the possible variations. The SELECT statement is used to select data from a database. The query results are presented in a tabular format called the result-set, or resultset. Some people call them recordsets as well. The basic syntax is deceptively simple. In fact, we've already used it to display our new row above. The asterisk symbol (*) is shorthand for "all the columns":

```
MySQL>SELECT * FROM employees;
```

You can also explicitly name the columns that you want by listing them in a comma delimited list. Notice that the column order does not have to match that of the table. The following query lists the salaries for each shop:

```
MySQL>SELECT salary, shop_id FROM employees;
```

salary	shop_id
--------	---------

4500	1
------	---

4700	1
------	---

5600	2
------	---

5100	3
------	---

5200	3
------	---

1000	3
------	---

A Few Comments Regarding Syntax

Several questions come up a lot when people begin to write SQL statements. One is whether the closing semicolon is mandatory. The simple answer is that it is not. The semicolon character is a statement terminator. It is a part of the ANSI SQL-92 standard, but was never used within Transact-SQL, the most common SQL flavor. Indeed, it was possible to code T-SQL for years without ever encountering a semicolon. If you're sending just one statement to the database, technically you can dispense with the statement terminator; however in a script or program, there are times that you need it if you're sending more than one statement.

Another common question is whether SQL statements are case sensitive. Well, wonder no more. They are not. Neither keywords such as SELECT or FROM are case sensitive, nor are table and field names. Having said that, it is a good practice to keep keywords in uppercase and data fields in the case that matches their definition, whether that is upper, lower or mixed case.

Narrowing Down the Resultset Using the WHERE Clause

The WHERE clause can be used to limit the resultset to records that match a specified criterion. The following WHERE clause narrows down our previous resultset to only female employees:

```
MySQL>SELECT salary, shop_id FROM employees  
-->WHERE gender = 'f';
```

salary shop_id

4700 1

5600 2

Here are the basic operators supported by MySQL:

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal

The UPDATE Statement

Unlike the INSERT statement, which is used to add new records, The UPDATE statement is used to update existing records in a table. The WHERE clause specifies which record or records should be updated. Don't forget to use it, or the UPDATE will be applied to ALL RECORDS! Here is the syntax:

```
UPDATE table_name
SET column1=value,
    column2=value2,
    ...
WHERE some_column=some_value
```

We can use the UPDATE statement to change one of our female employees' last name after marriage. To specify the exact record, it's best to use the id field, as it is unique. Don't use Kirsten's maiden name, as it is not guaranteed to be unique. In a large organization, duplicate names are a definite concern!

```
MySQL>UPDATE employees
-->SET name='Kirsten Johnson'
-->WHERE id=3;
```

The name has been changed:

```
MySQL>SELECT * FROM employees;
```

id	shop_id	gender	name	salary
1	1	m	Jon Simpson	4500
2	1	f	Barbara Breitenmoser	4700
3	2	f	Kirsten Johnson	5600

4	3	m	Ralph Teller	5100
5	3	m	Peter Jonson	5200
6	3	m	Al Bundy	1000

The DELETE Statement

The DELETE statement is used to delete rows in a table. That's easy enough to grasp, but keep in mind that delete means delete, as in "never to be seen again"! You may want to take care in using the DELETE statement for this reason. In fact, my place of work has a policy of never using the DELETE statement. We use an active indicator field for that purpose instead. Like UPDATE, the DELETE statement uses a WHERE clause to specify which record(s) to affect:

```
DELETE FROM table_name
WHERE some_column=some_value
```

Note that there is no column list because the DELETE always applies to entire rows and not individual columns. For that, you have to use an UPDATE and set the column to NULL:

```
MySQL>UPDATE employees
-->SET salary=NULL
-->WHERE id=2;
```

id	shop_id	gender	name	salary
1	1	m	Jon Simpson	4500
2	1	f	Barbara Breitenmoser	(NULL)
3	2	f	Kirsten Johnson	5600

4	3	m	Ralph Teller	5100
5	3	m	Peter Jonson	5200
6	3	m	Al Bundy	1000

Bear in mind that this operation can fail if you try to set a non nullable column to NULL. In the following example, the overworked DBA tried to set the id field to NULL, which is not advisable!

```
MySQL>UPDATE employees
-->SET id=NULL
-->WHERE id=2;
MySQL>SQL Error: Column 'id' cannot be null
```

That covers the major Data Manipulation and Query statements supported by MySQL. Next time, we'll be taking a closer look at some MySQL variations and specifics.

Additional Resources

MySQL [Reference Manual](#)

» [See All Articles by Columnist Rob Gravelle](#)

[MySQL Archives](#)

1 Comments ([click to add your comment](#))

By kanchan khurana July 15 2010 12:49 PDT

its a good deal 4 having you. i like the way of your teaching with examples. thank you.

[Reply to this comment](#)



Comment and Contribute

Your name/nickname

Your email

Subject

(Maximum characters: 1200). You have

1200

 characters left.



Type the text

[Privacy & Terms](#)



Submit Your Comment

Latest Forum Threads

MySQL Forum

Topic	By	Replies	Updated
MySQL in high availability and transection secure banking application	klamor	1	August 28th, 10:24 AM

MySQL rollback	UAL225	1	August 28th, 10:15 AM
Browsing a DB file that uses MySql	finleytech	1	July 26th, 10:51 AM
php cookie won't pass variable to next page	ITdevGirl	0	June 22nd, 12:13 PM



Property of QuinStreet Enterprise.

[Terms of Service](#) | [Licensing & Reprints](#) | [About Us](#) | [Privacy Policy](#) | [Advertise](#)

Copyright 2018 QuinStreet Inc. All Rights Reserved.