



MariaDB: Joins

This MariaDB tutorial explains how to use MariaDB **JOINS** (inner and outer) with syntax, visual illustrations, and examples.

Description

MariaDB JOINS are used to retrieve data from multiple tables. A MariaDB JOIN is performed whenever two or more tables are joined in a SQL statement.

There are different types of MariaDB joins:

- MariaDB INNER JOIN (or sometimes called simple join)
- MariaDB LEFT OUTER JOIN (or sometimes called LEFT JOIN)
- MariaDB RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)

So let's discuss MariaDB JOIN syntax, look at visual illustrations of MariaDB JOINS, and explore MariaDB JOIN examples.

INNER JOIN (simple join)

Chances are, you've already written a statement that uses a MariaDB INNER JOIN. It is the most common type of join. MariaDB INNER JOINS return all rows from multiple tables where the join condition is met.

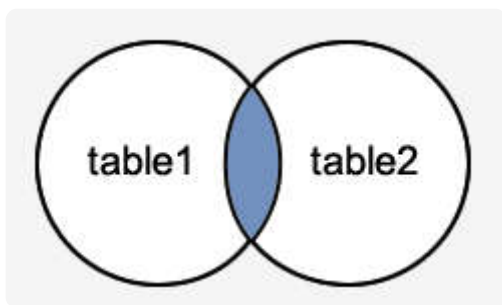
Syntax

The syntax for the INNER JOIN in MariaDB is:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.column = table2.column;
```

Visual Illustration

In this visual diagram, the MariaDB INNER JOIN returns the shaded area:



The MariaDB INNER JOIN would return the records where *table1* and *table2* intersect.

Example

Here is an example of a MariaDB INNER JOIN:

```
SELECT sites.site_id, sites.site_name, pages.page_title
FROM sites
INNER JOIN pages
ON sites.site_id = pages.site_id;
```

This MariaDB INNER JOIN example would return all rows from the *sites* and *pages* tables where there is a matching *site_id* value in both the *sites* and *pages* tables.

Let's look at some data to explain how the INNER JOINS work:

We have a table called *sites* with two fields (*site_id* and *site_name*). It contains the following data:

site_id	site_name
1000	TechOnTheNet.com
2000	CheckYourMath.com
3000	BigActivities.com
4000	Google.com

We have another table called *pages* with three fields (*page_id*, *site_id*, and *file_size*). It contains the following data:

page_id	site_id	page_title
1	1000	MariaDB
2	1000	Oracle
3	2000	Convert cm to inches
4	3000	Coloring pages
5	5000	Great stuff

If we run the MariaDB SELECT statement (that contains an INNER JOIN) below:

```
SELECT sites.site_id, sites.site_name, pages.page_id, pages.page_title
FROM sites
INNER JOIN pages
ON sites.site_id = pages.site_id;
```

Our result set would look like this:

site_id	site_name	page_id	page_title
1000	TechOnTheNet.com	1	MariaDB
1000	TechOnTheNet.com	2	Oracle
2000	CheckYourMath.com	3	Convert cm to inches
3000	BigActivities.com	4	Coloring pages

The row for *Google.com* from the *sites* table would be omitted, since the *site_id* of 5000 does not exist in both tables. The row for 4 (*page_id*) from the *pages* table would be omitted, since the *site_id* of 5000 does not exist in the *sites* table.

Old Syntax

As a final note, it is worth mentioning that the MariaDB INNER JOIN example above could be rewritten using the older implicit syntax as follows (but we still recommend using the INNER JOIN keyword syntax):

```
SELECT sites.site_id, sites.site_name, pages.page_id, pages.page_title
FROM sites, pages
WHERE sites.site_id = pages.site_id;
```

LEFT OUTER JOIN

Another type of join is called a MariaDB LEFT OUTER JOIN. This type of join returns all rows from the LEFT-hand table specified in the ON condition and **only** those rows from the other table where the joined fields are equal (join condition is met).

Syntax

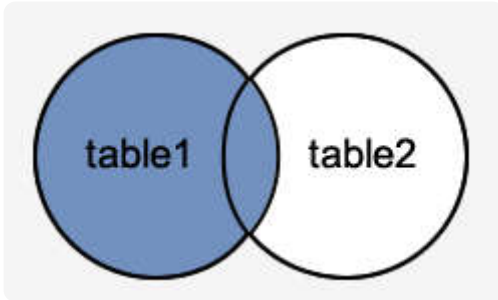
The syntax for the MariaDB LEFT OUTER JOIN is:

```
SELECT columns
FROM table1
LEFT [OUTER] JOIN table2
ON table1.column = table2.column;
```

In some databases, the LEFT OUTER JOIN keywords are replaced with LEFT JOIN.

Visual Illustration

In this visual diagram, the MariaDB LEFT OUTER JOIN returns the shaded area:



The MariaDB LEFT OUTER JOIN would return the all records from *table1* and only those records from *table2* that intersect with *table1*.

Example

Here is an example of a MariaDB LEFT OUTER JOIN:

```
SELECT sites.site_id, sites.site_name, pages.page_id, pages.page_title
FROM sites
LEFT JOIN pages
ON sites.site_id = pages.site_id;
```

This LEFT OUTER JOIN example would return all rows from the *sites* table and only those rows from the *pages* table where the joined fields are equal.

If a *site_id* value in the *sites* table does not exist in the *pages* table, all fields in the *pages* table will display as <null> in the result set.

Let's look at some data to explain how LEFT OUTER JOINS work:

We have a table called *sites* with two fields (*site_id* and *site_name*). It contains the following data:

site_id	site_name
1000	TechOnTheNet.com
2000	CheckYourMath.com
3000	BigActivities.com
4000	Google.com

We have a second table called *pages* with three fields (*page_id*, *site_id*, and *page_title*). It contains the following data:

page_id	site_id	page_title
1	1000	MariaDB
2	1000	Oracle

page_id	site_id	page_title
3	2000	Convert cm to inches
4	3000	Coloring pages
5	5000	Great stuff

If we run the SELECT statement (that contains a LEFT OUTER JOIN) below:

```
SELECT sites.site_id, sites.site_name, pages.page_id, pages.page_title
FROM sites
LEFT JOIN pages
ON sites.site_id = pages.site_id;
```

Our result set would look like this:

site_id	site_name	page_id	page_title
1000	TechOnTheNet.com	1	MariaDB
1000	TechOnTheNet.com	2	Oracle
2000	CheckYourMath.com	3	Convert cm to inches
3000	BigActivities.com	4	Coloring pages
4000	Google.com	<null>	<null>

The rows for *Google.com* would be included because a LEFT OUTER JOIN was used. However, you will notice that the *page_id* and *page_title* fields for those records contains a <null> value.

RIGHT OUTER JOIN

Another type of join is called a MariaDB RIGHT OUTER JOIN. This type of join returns all rows from the RIGHT-hand table specified in the ON condition and **only** those rows from the other table where the joined fields are equal (join condition is met).

Syntax

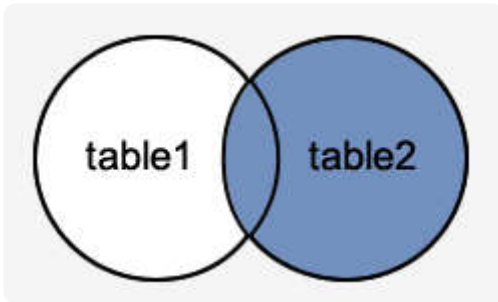
The syntax for the MariaDB RIGHT OUTER JOIN is:

```
SELECT columns
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;
```

In some databases, the RIGHT OUTER JOIN keywords are replaced with RIGHT JOIN.

Visual Illustration

In this visual diagram, the MariaDB RIGHT OUTER JOIN returns the shaded area:



The MariaDB RIGHT OUTER JOIN would return the all records from *table2* and only those records from *table1* that intersect with *table2*.

Example

Here is an example of a MariaDB RIGHT OUTER JOIN:

```
SELECT sites.site_id, sites.site_name, pages.page_id, pages.page_title
FROM sites
RIGHT JOIN pages
ON sites.site_id = pages.site_id;
```

This RIGHT OUTER JOIN example would return all rows from the *pages* table and only those rows from the *sites* table where the joined fields are equal.

If a *site_id* value in the *pages* table does not exist in the *sites* table, all fields in the *sites* table will display as <null> in the result set.

Let's look at some data to explain how RIGHT OUTER JOINS work:

We have a table called *sites* with two fields (*site_id* and *site_name*). It contains the following data:

site_id	site_name
1000	TechOnTheNet.com
2000	CheckYourMath.com
3000	BigActivities.com
4000	Google.com

We have a second table called *pages* with three fields (*page_id*, *site_id*, and *page_title*). It contains the following data:

page_id	site_id	page_title
1	1000	MariaDB
2	1000	Oracle

page_id	site_id	page_title
3	2000	Convert cm to inches
4	3000	Coloring pages
5	5000	Great stuff

If we run the SELECT statement (that contains a RIGHT OUTER JOIN) below:

```
SELECT sites.site_id, sites.site_name, pages.page_id, pages.page_title
FROM sites
RIGHT JOIN pages
ON sites.site_id = pages.site_id;
```

Our result set would look like this:

site_id	site_name	page_id	page_title
1000	TechOnTheNet.com	1	MariaDB
1000	TechOnTheNet.com	2	Oracle
2000	CheckYourMath.com	3	Convert cm to inches
3000	BigActivities.com	4	Coloring pages
<null>	<null>	5	Great stuff

The row for 5 (*page_id*) would be included because a RIGHT OUTER JOIN was used. However, you will notice that the *site_id* and *site_name* fields for that record contains a <null> value.

Copyright TechOnTheNet.com