MariaDB: SELECT Statement

This MariaDB tutorial explains how to use the MariaDB **SELECT statement** with syntax and examples.

Description

The MariaDB SELECT statement is used to retrieve records from one or more tables in MariaDB.

Syntax

In its simplest form, the syntax for the SELECT statement in MariaDB is:

```
SELECT expressions
FROM tables
[WHERE conditions];
```

However, the full syntax for the MariaDB SELECT statement is:

```
SELECT [ ALL | DISTINCT ]
expressions
FROM tables
[WHERE conditions]
[GROUP BY expressions]
[HAVING condition]
[ORDER BY expression [ ASC | DESC ]]
[LIMIT [offset_value] number_rows | LIMIT number_rows OFFSET offset_value]
[PROCEDURE procedure_name]
[INTO [ OUTFILE 'file_name' options
       | DUMPFILE 'file_name'
       | @variable1, @variable2, ... @variable_n ]
[FOR UPDATE | LOCK IN SHARE MODE];
```

Parameters or Arguments

ALL

Optional. *ALL* returns all matching rows.

DISTINCT

Optional. *DISTINCT* removes duplicates from the result set.

expressions

The columns or calculations that you wish to retrieve. Use * if you wish to select all columns.

tables

The tables that you wish to retrieve records from. There must be at least one table listed in the FROM clause.

WHERE conditions

Optional. The conditions that must be met for the records to be selected.

GROUP BY expressions

Optional. It collects data across multiple records and groups the results by one or more columns.

HAVING condition

Optional. It is used in combination with the GROUP BY to restrict the groups of returned rows to only those whose the condition is TRUE.

ORDER BY expression

Optional. It is used to sort the records in your result set.

LIMIT

Optional. If LIMIT is provided, it controls the maximum number of records to retrieve. At most, the number of records specified by *number_rows* will be returned in the result set. The first row returned by LIMIT will be determined by *offset_value*.

PROCEDURE

Optional. If provided, *procedure_name* is the name of the procedure that should process the data in the result set.

INTO

Optional. If provided, it allows you to write the result set to either a file or variable.

Value	Explanation

Value	Explanation
INTO OUTFILE 'filename' options	Writes the result set to a file called *filename* on the server host. For *options*, you can specify: FIELDS ESCAPED BY *'character'* FIELDS TERMINATED BY *'character'* [OPTIONALLY ENCLOSED BY *'character'*] LINES TERMINATED BY *'character'* where *character* is the character to display as the ESCAPE, ENCLOSED, or TERMINATED character. For example: SELECT supplier_id, supplier_name FROM suppliers INTO OUTFILE 'results.txt' FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
INTO DUMPFILE 'filename'	Writes one row of the result set to a file called *filename* on the server host. With this method, there is no column termination, no line termination, or escape processing.
INTO @variable1, @variable2, ... @variable_n	Writes the result set to one or more variables, as specified by @variable1, @variable2, ... @variable_n

FOR UPDATE

Optional. Records affected by the query are write-locked until the transaction has completed

LOCK IN SHARE MODE

Optional. Records affected by the query can be used by other transactions but can not be updated or deleted by those other transactions.

Example - Select all columns from one table

Let's look at how to use a MariaDB SELECT query to select all columns from a table.

For example:

```
SELECT *
FROM sites
WHERE site_name = 'TechOnTheNet.com'
ORDER BY site_id ASC;
```

In this SELECT example, we've used * to signify that we wish to select all fields from the *sites* table where the *site_name* is 'TechOnTheNet.com'. The results are sorted by *site_id* in ascending order.

Example - Select individual columns from one table

When using the SELECT statement in MariaDB, you do not have to select all columns from the table. Instead, you can select the individual columns that you would like to return in your result set.

For example:

```
SELECT site_id, site_name
FROM sites
WHERE site_id < 32
ORDER BY site_id ASC, site_name DESC;
```

This MariaDB SELECT example would return only the *site_id* and *site_name* fields from the *sites* table where the *site_id* is less than 32. The results are sorted by *site_id* in ascending order and then *site_name* in descending order.

Example - Select columns from multiple tables

The SELECT statement in MariaDB can also select columns from more than one table.

For example:

```
SELECT pages.page_id, sites.site_name
FROM sites
INNER JOIN pages
ON sites.site_id = pages.site_id
WHERE sites.site_name = 'TechOnTheNet.com'
ORDER BY pages.page_id;
```

This SELECT statement example joins two tables to return a result set that includes the *page_id* and *site_name* fields. The results of the SELECT statement are filtered where the *site_name* is 'TechOnTheNet.com' and the *site_id* value matches in both the *sites* and *pages* table. The results are sorted by *page_id* in ascending order.

Example - write to a file

Finally, let's look at how to use the MariaDB SELECT statement to write the results of the SELECT statement to a file.

For example:

```
SELECT site_id, site_name
FROM sites
WHERE site_name = 'TechOnTheNet.com'
ORDER BY site_id DESC
INTO OUTFILE 'results.txt'
     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
     LINES TERMINATED BY '\n';
```

This MariaDB SELECT example would return only the *site_id* and *site_name* fields from the *sites* table where the *site_name* is 'TechOnTheNet.com'. The results would be sorted by *site_id* in descending order and written to a file called results.txt.