MariaDB: ALIASES

This MariaDB tutorial explains how to use MariaDB **ALIASES** (temporary names for columns or tables) with syntax and examples.

Description

MariaDB ALIASES can be used to create a temporary name for columns or tables.

- COLUMN ALIASES are used to make column headings in your result set easier to read.
- TABLE ALIASES are used to shorten your SQL to make it easier to read or when you are performing a self join (ie: listing the same table more than once in the FROM clause).

Syntax

The syntax to alias a column in MariaDB is:

```
column_name [ AS ] alias_name
```

OR

The syntax to alias a table in MariaDB is:

```
table_name [ AS ] alias_name
```

Parameters or Arguments

column_name
The original name of the column that you wish to alias.

table_name
The original name of the table that you wish to alias.

AS
Optional. Whether you specify the *AS* keyword or not has no impact on the alias in MariaDB.

alias_name
The temporary name to assign to the column or table.

Note

- If the *alias_name* contains spaces, you must enclose the *alias_name* in quotes.
- It is acceptable to use spaces when you are aliasing a column name. However, it is not generally good practice to use spaces when you are aliasing a table name.
- The *alias_name* is only valid within the scope of the SQL statement.
- Most programmers will specify the *AS* keyword when aliasing a column name, but not when aliasing a table name.

Example - ALIAS a column

Generally, aliases are used to make the column headings in your result set easier to read in MariaDB. For example, when using the COUNT function, you might alias the result of the COUNT function.

For example:

```
SELECT site_name, COUNT(site_id) AS Total
FROM sites
GROUP BY site_name;
```

In this example, we've aliased the COUNT(site_id) field as *Total*. As a result, *Total* will display as the heading for the second column when the result set is returned. Because our *alias_name* did not include any spaces, we are not required to enclose the *alias_name* in quotes.

However, it would have been perfectly acceptable to write this example using quotes as follows:

```
SELECT site_name, COUNT(site_id) AS "Total"
FROM sites
GROUP BY site_name;
```

Next, let's look at an example where we are required to enclose the *alias_name* in quotes because the alias contains spaces.

For example:

```
SELECT site_name, COUNT(site_id) AS "Total Sites"
FROM sites
GROUP BY site_name;
```

In this example, we've aliased the COUNT(site_id) field as "Total Sites". Since there are spaces in this *alias_name*, "Total Sites" must be enclosed in quotes.

Example - ALIAS a Table

When you create an alias on a table in MariaDB, it is either because you plan to list the same table name more than once in the FROM clause (ie: self join), or you want to shorten the table name to make the SQL statement easier to read.

Let's look at an example of how to alias a table name in MariaDB

For example:

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

In this example, we've created an alias for the *sites* table called *s*. Now within this SQL statement, we can refer to the *sites* table as *s*.

When creating table aliases, it is not necessary to create aliases for all of the tables listed in the FROM clause. You can choose to create aliases on any or all of the tables.

For example, we could modify our example above and create an alias for the *pages* table as well.

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

Now we have an alias for the *pages* table called *p* as well as the alias for the *sites* table called *s*.