

Aliases in SQL

The dictionary meaning of the word **Alias** is “an assumed or an additional name that a person sometimes uses”. In the same way, while selecting any column we may want to show the modified column name rather than the one stored in the table, this can be done with the help of aliases. Following are the situations in which aliases can be used:

1. To make the column name more readable while printing the content.
2. Sometimes the stored names are either smaller or larger than the desired name, in this case, aliases can be used to print the desired column names.
3. Aliases are also preferred when more than one table is used and they contain the same column names.

Note: Aliases can be used for column names and also for table names.

The basic syntax for using an Alias

The **as** keyword is used to define an alias.

For column Alias:

```
SELECT column as alias_name FROM table_name;
```

- **column:** fields in the table
- **alias_name:** temporary alias name to be used in replacement of original column name
- **table_name:** name of the table

For table alias:

```
SELECT column FROM table_name as alias_name;
```

- **column:** fields in the table
- **table_name:** name of table
- **alias_name:** temporary alias name to be used in replacement of original table name

Let's see an example to better understand the syntax for column and table aliases.

```
SELECT e.emp_no as "Employee Code",  
       CONCAT(e.first_name, ' ', e.last_name) as "Name"  
FROM employees as e
```

In this example, for the table employee, the alias used is "e", the use of a shorter name for the table is useful while using the columns of that table, like In this example, we used e.emp_no. We also used the column alias for e.emp_no as "Employee Code" which makes this column more readable. The best use of an alias is while using group functions or concatenation functions. In this example, we used "Name" as an alias for CONCAT(e.first_name, ' ', e.last_name).

In the table below, a comparison of the output readability is shown with and without aliases.

Without using Aliases			Using Aliases		
<pre>SELECT employees.emp_no, CONCAT(employees.first_name, ' ', employees.last_name) FROM employees</pre>			<pre>SELECT e.emp_no AS "Employee Code", CONCAT(e.first_name, ' ', e.last_name) AS "Name" FROM employees AS e</pre>		
	emp_no	concat		Employee Code	Name
1	10001	Georgi Facello	1	10001	Georgi Facello
2	10002	Bezalel Simmel	2	10002	Bezalel Simmel
3	10003	Parto Bamford	3	10003	Parto Bamford
4	10004	Chirstian Koblick	4	10004	Chirstian Koblick
5	10005	Kyoichi Maliniak	5	10005	Kyoichi Maliniak
6	10006	Anneke Preusig	6	10006	Anneke Preusig
7	10007	Tzvetan Zielinski	7	10007	Tzvetan Zielinski

As we can see that using aliases, the column names are more readable than the original ones.