SQL Aliases

‹ Previous Next ›

SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name.

Aliases are often used to make column names more readable.

An alias only exists for the duration of that query.

An alias is created with the `AS` keyword.

Example


```
SELECT CustomerID AS ID
FROM Customers;
```

Try it Yourself »

AS is Optional

Actually, in most database languages, you can skip the AS keyword and get the same result:

Example

```
SELECT CustomerID ID
FROM Customers;
```

Try it Yourself »

Syntax

When alias is used on column:

```
SELECT column_name AS alias_name
FROM table_name;
```

When alias is used on table:

```
SELECT column_name(s)
FROM table_name AS alias_name;
```

Demo Database

Below is a selection from the **Customers** and **Orders** tables used in the examples:

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y	Ana Trujillo	Avda. de la Constitución	México D.F.	05021	Mexico

	helados		2222			
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

Orders

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	7/4/1996	3
10249	81	6	7/5/1996	1
10250	34	4	7/8/1996	2

Alias for Columns

The following SQL statement creates two aliases, one for the CustomerID column and one for the CustomerName column:

Example

```
SELECT CustomerID AS ID, CustomerName AS Customer
FROM Customers;
```

Try it Yourself »

Using Aliases With a Space Character

If you want your alias to contain one or more spaces, like "`My Great Products`", surround your alias with square brackets or double quotes.

Example

Using [square brackets] for aliases with space characters:

```
SELECT ProductName AS [My Great Products]
FROM Products;
```

Try it Yourself »

Example

Using "double quotes" for aliases with space characters:

```
SELECT ProductName AS "My Great Products"
FROM Products;
```

Try it Yourself »

Note: Some database systems allows both [] and "", and some only allows one of them.

Concatenate Columns

The following SQL statement creates an alias named "Address" that combine four columns (Address, PostalCode, City and Country):

Example

```
SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address
FROM Customers;
```

Try it Yourself »

Note: To get the SQL statement above to work in MySQL use the following:

MySQL Example

```
SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address
FROM Customers;
```

Try it Yourself »

Note: To get the SQL statement above to work in Oracle use the following:

Oracle Example

```
SELECT CustomerName, (Address || ', ' || PostalCode || ' ' || City || ', ' || Country) AS Address
FROM Customers;
```

Alias for Tables

The same rules applies when you want to use an alias for a table.

Example

Refer to the Customers table as Persons instead:

```
SELECT * FROM Customers AS Persons;
```

Try it Yourself »

It might seem useless to use aliases on tables, but when you are using more than one table in your queries, it can make the SQL statements shorter.

The following SQL statement selects all the orders from the customer with CustomerID=4 (Around the Horn). We use the "Customers" and "Orders" tables, and give them the table aliases of "c" and "o" respectively (Here we use aliases to make the SQL shorter):

Example

```
SELECT o.OrderID, o.OrderDate, c.CustomerName
FROM Customers AS c, Orders AS o
WHERE c.CustomerName='Around the Horn' AND c.CustomerID=o.CustomerID;
```

Try it Yourself »

The following SQL statement is the same as above, but without aliases:

Example

```
SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName
FROM Customers, Orders
WHERE Customers.CustomerName='Around the Horn' AND Customers.CustomerID=Orders.CustomerID;
```

Try it Yourself »

Aliases can be useful when:

- There are more than one table involved in a query
- Functions are used in the query
- Column names are big or not very readable
- Two or more columns are combined together

Test Yourself With Exercises

Exercise:

When displaying the `Customers` table, make an ALIAS of the `PostalCode` column, the column should be called `Pno` instead.

```
SELECT CustomerName,
Address,
PostalCode
FROM Customers;
```

Submit Answer »

Start the Exercise