# **SQL Aliases**

**SQL Aliases:**

1. SQL aliases are used to give a table, or a column in a table, a temporary name.
2. The 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; |

1. Aliases are often used to make column names more readable.
2. An alias only exists for the duration of that query.
3. An alias is created with the AS keyword.
4. Example:

|  |
| --- |
| SELECT CustomerID AS ID  FROM Customers; |

|  |
| --- |
| **ID** |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |

**AS is Optional:**

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

|  |
| --- |
| SELECT CustomerID ID  FROM Customers; |

|  |
| --- |
| **ID** |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

**Alias for Columns:**

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

|  |
| --- |
| SELECT CustomerID AS ID, CustomerName AS Customer  FROM Customers; |

|  |  |
| --- | --- |
| **ID** | **Customer** |
| 1 | Alfreds Futterkiste |
| 2 | Ana Trujillo Emparedados y helados |
| 3 | Antonio Moreno Taquería |
| 4 | Around the Horn |
| 5 | Berglunds snabbköp |
| 6 | Blauer See Delikatessen |
| 7 | Blondel père et fils |
| 8 | Bólido Comidas preparadas |

**Using Aliases With a Space Character:**

1. If you want your alias to contain one or more spaces, like "My Great Products", surround your alias with square brackets or double quotes.
2. Example:  
   Using [square brackets] for aliases with space characters:

|  |
| --- |
| SELECT ProductName AS [My Great Products]  FROM Products; |

|  |
| --- |
| **My Great Products** |
| Chais |
| Chang |
| Aniseed Syrup |
| Chef Anton's Cajun Seasoning |
| Chef Anton's Gumbo Mix |
| Grandma's Boysenberry Spread |
| Uncle Bob's Organic Dried Pears |

Example:

|  |
| --- |
| SELECT ProductName AS "My Great Products"  FROM Products; |

|  |
| --- |
| **"My Great Products"** |
| Chais |
| Chang |
| Aniseed Syrup |
| Chef Anton's Cajun Seasoning |
| Chef Anton's Gumbo Mix |
| Grandma's Boysenberry Spread |
| Uncle Bob's Organic Dried Pears |
| Northwoods Cranberry Sauce |
| Mishi Kobe Niku |

**Concatenate Columns:**

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

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

|  |  |
| --- | --- |
| **CustomerName** | **Address** |
| Alfreds Futterkiste | Obere Str. 57, 12209 Berlin, Germany |
| Ana Trujillo Emparedados y helados | Avda. de la Constitución 2222, 05021 México D.F., Mexico |
| Antonio Moreno Taquería | Mataderos 2312, 05023 México D.F., Mexico |
| Around the Horn | 120 Hanover Sq., WA1 1DP London, UK |
| Berglunds snabbköp | Berguvsvägen 8, S-958 22 Luleå, Sweden |
| Blauer See Delikatessen | Forsterstr. 57, 68306 Mannheim, Germany |

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

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

|  |  |
| --- | --- |
| **CustomerName** | **Address** |
| Alfreds Futterkiste | Obere Str. 57, 12209, Berlin, Germany |
| Ana Trujillo Emparedados y helados | Avda. de la Constitución 2222, 05021, México D.F., Mexico |
| Antonio Moreno Taquería | Mataderos 2312, 05023, México D.F., Mexico |
| Around the Horn | 120 Hanover Sq., WA1 1DP, London, UK |
| Berglunds snabbköp | Berguvsvägen 8, S-958 22, Luleå, Sweden |
| Blauer See Delikatessen | Forsterstr. 57, 68306, Mannheim, Germany |

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

Example:

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

**Alias for Tables:**

1. The same rules applies when you want to use an alias for a table.
2. Example:  
   Refer to the Customers table as Persons instead:

|  |
| --- |
| SELECT \* FROM Customers AS Persons; |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **ContactName** | **Address** | **City** | **PostalCode** | **Country** |
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |

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; |

|  |  |  |
| --- | --- | --- |
| **OrderID** | **OrderDate** | **CustomerName** |
| 10383 | 12/16/1996 | Around the Horn |
| 10355 | 11/15/1996 | Around the Horn |

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; |

|  |  |  |
| --- | --- | --- |
| **OrderID** | **OrderDate** | **CustomerName** |
| 10383 | 12/16/1996 | Around the Horn |
| 10355 | 11/15/1996 | Around the Horn |

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