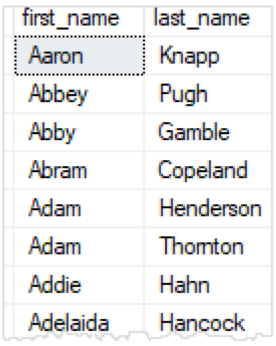
LAB - ALIAS

In this guided lab, you will learn how to use the SQL Server aliases including **column alias** and **table alias**.

Column alias

When you use the `SELECT` statement to query data from a table, SQL Server uses the column names as the column headings for the output. See the following example:

```
SELECT
    first_name,
    last_name
FROM
    sales.customers
ORDER BY
    first_name;
```

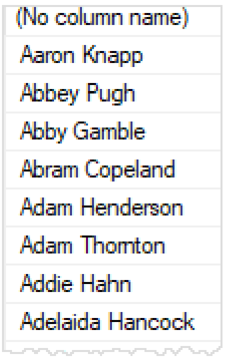


first_name	last_name
Aaron	Knapp
Abbey	Pugh
Abby	Gamble
Abram	Copeland
Adam	Henderson
Adam	Thornton
Addie	Hahn
Adelaida	Hancock

As clearly shown in the output, the `first_name` and `last_name` column names were used for the column headings respectively.

To get full names of customers, you can concatenate the first name, space, and the last name using the concatenation `+` operator as shown in the following query:

```
SELECT
    first_name + ' ' + last_name
FROM
    sales.customers
ORDER BY
    first_name;
```



(No column name)
Aaron Knapp
Abbey Pugh
Abby Gamble
Abram Copeland
Adam Henderson
Adam Thornton
Addie Hahn
Adelaida Hancock

SQL Server returned the full name column as (`No column name`) which is not meaningful in this case.

To assign a column or an expression a temporary name during the query execution, you use a column alias.

The following illustrates the column alias syntax:

```
column_name | expression  AS column_alias
```

In this syntax, you use the `AS` keyword to separate the column name or expression and the alias.

Because the `AS` keyword is optional, you can assign an alias to a column as follows:

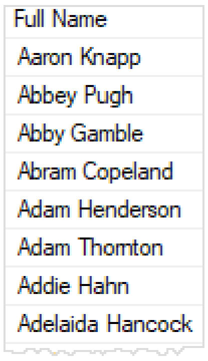
```
column_name | expression column_alias
```

Back to the example above, you can rewrite the query using a column alias:

```
SELECT
    first_name + ' ' + last_name AS full_name
FROM
    sales.customers
ORDER BY
    first_name;
```

Note that if the column alias contains spaces, you need to enclose it in quotation marks as shown in the following example:

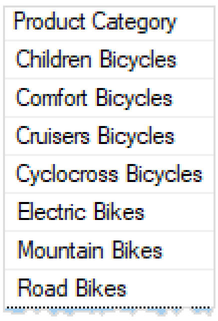
```
SELECT
    first_name + ' ' + last_name AS 'Full Name'
FROM
    sales.customers
ORDER BY
    first_name;
```



Full Name
Aaron Knapp
Abbey Pugh
Abby Gamble
Abram Copeland
Adam Henderson
Adam Thomton
Addie Hahn
Adelaida Hancock

The following example shows how to assign an alias to a column:

```
SELECT
    category_name 'Product Category'
FROM
    production.categories;
```



Product Category
Children Bicycles
Comfort Bicycles
Cruisers Bicycles
Cyclocross Bicycles
Electric Bikes
Mountain Bikes
Road Bikes

In this example, the product category column alias is much more clear than the `category_name` column name.

When you assign a column an alias, you can use either the column name or the column alias in the `ORDER BY` clause as shown in the following example:

```
SELECT
    category_name 'Product Category'
FROM
    production.categories
ORDER BY
    category_name;


SELECT
    category_name 'Product Category'
FROM
    production.categories
ORDER BY
    'Product Category';
```

Note that the `ORDER BY` clause is the very last clause to be processed therefore the column aliases are known at the time of sorting.

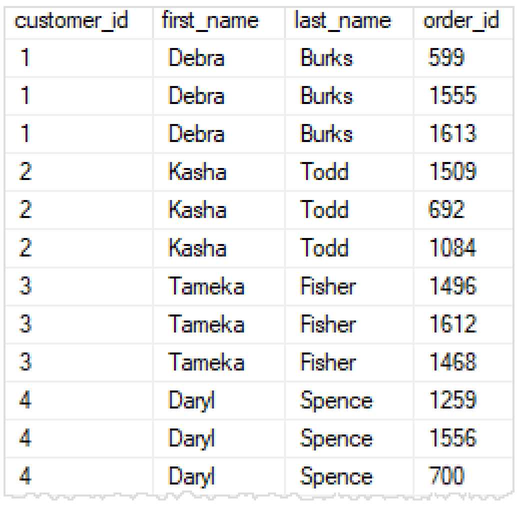
Table alias

Similar to the column alias, a table alias can be assigned either with or without the `AS` keyword:

```
table_name AS table_alias
table_name table_alias
```

See the following example:

```
SELECT
    sales.customers.customer_id,
    first_name,
    last_name,
    order_id
FROM
    sales.customers
INNER JOIN sales.orders
    ON sales.orders.customer_id = sales.customers.customer_id;
```



customer_id	first_name	last_name	order_id
1	Debra	Burks	599
1	Debra	Burks	1555
1	Debra	Burks	1613
2	Kasha	Todd	1509
2	Kasha	Todd	692
2	Kasha	Todd	1084
3	Tameka	Fisher	1496
3	Tameka	Fisher	1612
3	Tameka	Fisher	1468
4	Daryl	Spence	1259
4	Daryl	Spence	1556
4	Daryl	Spence	700

In this example, both the `customers` and the `orders` tables have a column with the same name `customer_id`, therefore, you need to refer to the column using the following syntax:

```
table_name.column_name
```

such as:

```
sales.custoners.customer_id
sales.orders.customer_id
```

If you did not do so, SQL server would issue an error.

The query above is quite difficult to read. Fortunately, you can improve its readability by using the table alias as follows:

```
SELECT
    c.customer_id,
    first_name,
    last_name,
    order_id
FROM
    sales.customers c
INNER JOIN sales.orders o
   ON o.customer_id = c.customer_id;
```

In this query, `c` is the alias for the `sales.customers` table and `o` is the alias for the `sales.orders` table.

When you assign an alias to a table, you must use the alias to refer to the table column. Otherwise, SQL Server will issue an error.

In this lab, you have learned how to use the SQL Server alias including column alias and table alias.