Secret identities: The importance of aliases

In this reading, you will learn about using aliasing to simplify your SQL queries. **Aliases** are used in SQL qu
create temporary names for a column or table. Aliases make referencing tables and columns in your SQL
much simpler when you have table or column names that are too long or complex to make use of in queri
Imagine a table name like `special_projects_customer_negotiation_mileages`. That would be
re-enter every time you use that table. With an alias, you can create a meaningful nickname that you can
your analysis. In this case `special_projects_customer_negotiation_mileages` can be aliased
`mileage`. Instead of having to write out the long table name, you can use a meaningful nickname that yo

Basic syntax for aliasing

Aliasing is the process of using aliases. In SQL queries, aliases are implemented by making use of the `AS`
The basic syntax for the `AS` command can be seen in the following query for aliasing a table:

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

Notice that `AS` is preceded by the table name and followed by the new nickname. It is a similar approach
a column:

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

In both cases, you now have a new name that you can use to refer to the column or table that was aliased

Alternate syntax for aliases

If using `AS` results in an error when running a query because the SQL database you are working with does
it, you can leave it out. In the previous examples, the alternate syntax for aliasing a table or column woul

- `FROM table_name alias_name`