

Formatting Best Practices

In case you want to test any of the ideas below, I have embedded a SQL workspace environment at the bottom of this page.

Formatting Your Queries

Using Upper and Lower Case in SQL

SQL queries can be run successfully whether characters are written in upper- or lower-case. In other words, SQL queries are not case-sensitive. The following query:

```
SELECT account_id  
FROM orders
```

is the same as:

```
select account_id  
from orders
```

which is also the same as:

```
SeLeCt AcCoUnt_id  
FrOm oRdErS
```

However, you may have noticed that we have been capitalizing SELECT and FROM, while we leave table and column names in lower case. This is because even though SQL is case-insensitive, **it is common and best practice to capitalize all SQL commands, like SELECT and FROM, and keep everything else in your query lower case.**

Capitalizing command words makes queries easier to read, which will matter more as you write more complex queries. For now, it is just a good habit to start getting into, to make your SQL queries more readable.

One other note: The text data stored in SQL tables can be either upper or lower case, and SQL *is* case-sensitive in regard to this text data.

Avoid Spaces in Table and Variable Names

It is common to use underscores and avoid spaces in column names. It is a bit annoying to work with spaces in SQL. In Postgres if you have spaces in column or table names, you need to refer to these columns/tables with double quotes around them (Ex: FROM "Table Name" as opposed to FROM table_name). In other environments, you might see this as square brackets instead (Ex: FROM [Table Name]).

Use White Space in Queries

SQL queries ignore spaces, so you can add as many spaces and blank lines between code as you want, and the queries are the same. This query

```
SELECT account_id FROM orders
```

is equivalent to this query:

```
SELECT account_id
```

```
FROM orders
```

and this query (but please don't ever write queries like this):

```
SELECT         account_id
```

```
FROM           orders
```

Semicolons

Depending on your SQL environment, your query may need a semicolon at the end to execute. Other environments are more flexible in terms of this being a "requirement." It is considered best practice to put a semicolon at the end of each statement, which also allows you to run multiple queries at once if your environment allows this.

Best practice:

```
SELECT account_id
```

```
FROM orders;
```

Always remember, **just because a query successfully runs, does not mean you have the correct results.** To see if your query worked like you wanted it to, you need to examine your output to see if it satisfies the problem or the question.