





By specifying which columns we want to returned by our query, we can choose only the column values we need. Just as you use a WHERE clause to limit the number of rows, you can use column selection to limit the number of columns. It’s about letting SQL do the heavy lifting for you.





This is a good programming practice to follow, but it has other benefits. As your tables get larger, it speeds up retrieval of your results. You’ll also see more speed when you eventually use SQL with another programming language, such as PHP.

**Q: What if I need all the columns from my table returned by a query? Should I actually be naming them in the SELECT rather than using the \* ?**

**A:** If you need them all, then by all means use the \*. It’s only when you don’t need all that you should try not to use it.

**Q:** **I tried to copy and paste a query from the Internet, and I kept getting errors when I tired to use it. Am I doing something wrong?**

**A:**  Queries pasted from web browsers sometimes contain invisible character that look like spaces but mean something different to SQL. Pasting them into a text editor is one way to see and remove these “gremlin” characters. You best bet is to paste it into a text editor first and take a close look at it.

**Q:** **So** **I should paste it into something like Microsoft Word?**

**A:** No, Word isn’t a good choice, since it does nothing to show you the invisible formatting that might be in the text. Tyr Notepad (PC) or TextEdit in plain-text mode (Mac).

**Q: About escaping the apostrophe, is there any reason to use one method over the other?**

**A:** Not really. We tend to use the backslash method only because we find that’s it’s easier to spot where that extra apostrophe is when things go wrong in a query. For Example, this is easier to process visually:



Other than that, there’s really no reason to favor one method over the other, both methods allow you to enter apostrophes into your text columns.