6 Common SQL Syntax Mistakes All Learners Make

Adrian Dembek

Tags: HOW TO HOW TO IN SQL

We all make mistakes when learning a new language – especially at the beginning. New words, complicated grammar... Everyone needs time to master a language. But when we get immediate feedback, we can progress more quickly. The same goes for learning SQL. In this article, we'll look at 6 common SQL syntax mistakes most SQL learners make.

Mastering SQL Queries

Interactive SQL courses are one way to get immediate feedback on your queries. These courses, like the ones offered in LEARNSQL.COM, usually introduce new concepts and then ask you to have a go at them right away. If you make a mistake, you'll instantly get a message so you can correct it (and improve your SQL skills).

6 Common SQL Syntax Mistakes

Below is a list of the SQL syntax mistakes that are especially easy for beginners to make. I bet you have made at least one of them a few times in your life! As I explain each SQL syntax mistake, I'll use examples based on the `car` table exercises from

LearnSQL.com's SQL QUERIES course. If you've done any of this course, they will be familiar to you; if not, the course is free to try.

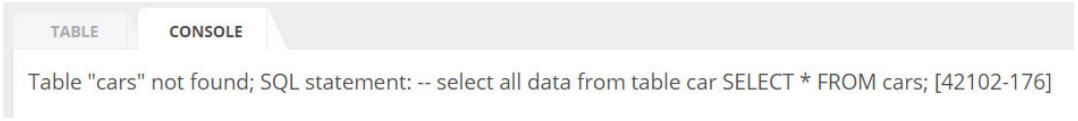
> LearnSQL.com provides a one-stop-shop for all things SQL, covering basic to advanced concepts in one single platform. LearnSQL.com is specifically geared towards SQL. It offers 30 interactive SQL courses that range in difficulty from beginner to advanced and monthly SQL challenges to practice your SQL skills.

SQL Syntax Mistake #1: Misspelling Table Names

When you start working with databases, you should get to know the table names right away. Let's say we want all the rows and columns from the `car` table. Will this statement work?

```
SELECT * FROM cars;
```

Oops! It's not working! Fortunately, the error message gives us a hint about the problem. In this case, there is no “cars” table in the database, which means we probably misspelled the table name.



The correct statement is:

```
SELECT * FROM car;
```

Now everything should work!

Many common syntax errors in SQL are simply misspellings: for example, typing `commodiy` instead of `commodity` or forgetting the underscore in `postal_codes`. In these cases, the “Table X not found...” message would mean there is a typo in the query and we should correct it.

SQL Syntax Mistake #2: Using Incorrect or Non-Existent Column Names

When you start working with a new database (or with any database that has many columns), **always double-check variable names**. Unfortunately, SQL is not that smart. If you misspell a word, it will not correct it. Each letter in the word should be correct (and in the correct place).

Let's find the production years for our cars:

```
SELECT year FROM car;
```

Again, something is wrong. Let's see the console.

TABLE CONSOLE

Column "year" not found; SQL statement: SELECT year FROM car; [42122-176]

The “year” column does not exist. So we have to check the table once again (for example, by using `SELECT * FROM car` to see all the column names) and find the correct name for that column. Instead of `year`, it is `production_year`! So we type:

```
SELECT production_year FROM car;
```

Now it is working!

The same “Column “x” not found…” message will also occur when you refer to a column from a table you intended (but forgot!) to JOIN.

Suppose there is an additional table called `customer` that has columns called `customer_id`, `name`, and `car_id`. If we `JOIN` the `car` and `customer` tables, we could match cars and their prices with customer data (using the “car_id” column).

But if we execute the query below…

```
SELECT price, name FROM car;
```

... we'll have a SQL syntax mistake. There is no “name” column in the table `car`, and unless we tell it otherwise SQL will look for it there. This is why forgetting to join two tables will also generate a “column not found” error. The solution is to do the following:

```
SELECT price, name FROM car JOIN customer
```

```
ON car.id=customer.car_id;
```

SQL Syntax Mistake #3: Forgetting SELECT List Commas

Now suppose we want to select the `brand` and `model` columns from car. We write the following query:

```
SELECT brand model FROM car;
```

Unfortunately, it doesn't work. In SQL, you have to separate column names with commas (as we usually do when making lists in English). All we need to do is add a comma:

```
SELECT brand, model FROM car;
```

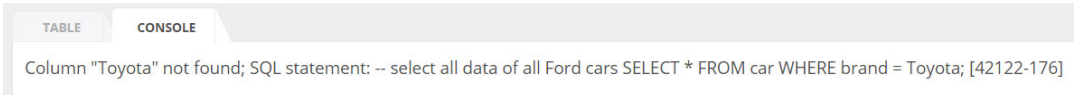
Mistake fixed! The query will work exactly as we want.

SQL Syntax Mistake #4: Leaving Out Quotation Marks

Another thing we need to remember is how to filter text values. In SQL, text values (like a brand of car or a city) are stored in text columns. What if we want to see all Toyotas in our car database? We type:

```
SELECT *
FROM car
WHERE brand = Toyota;
```

We get an error in the console, as shown below:



It says that there is no `Toyota` column in the table. When we do not use quotation marks around a text value, SQL treats that value as a column name.

This query will not work because we did not put the value (`'Toyota'`) in single quotation marks. Let's correct the mistake:

```
SELECT * FROM
car WHERE
brand = 'Toyota';
```

This is the proper way to filter text values. Note that SQL uses single quotation marks (`'text'`) instead of double ones (`"text"`) as some other programming languages do. Anyone with a programming background should be especially cautious about the kind of quotes they use here!

SQL Syntax Mistake #5: Not Specifying Table Names after SELECTs

Some beginners forget to specify which table they want to select columns from. They write:

```
SELECT brand, price;
```

SQL is just a programming language and it has its rules. If you list columns, you need to specify the table from which the columns should be extracted. This is done with the `FROM` clause:

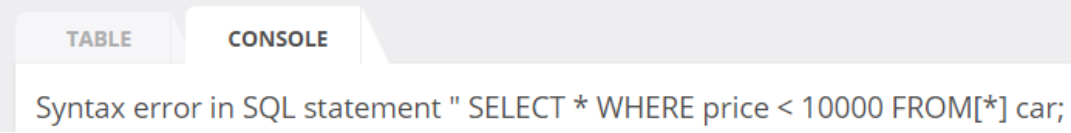
```
SELECT brand, price FROM car;
```

SQL Syntax Mistake #6: Ordering Statements Incorrectly

Learning SQL grammar is not that complicated. The order in which statements appear is one of its components. Let's try to select cars with a price under $10,000:

```
SELECT * WHERE price < 10000 FROM car;
```

Here we have another instance of common syntax errors in SQL:



It simply says there is a SQL syntax mistake in the statement. We should look at the statement and see if everything is in the correct order. The order should always be:

1. `SELECT`
2. `FROM`
3. `WHERE`

Now our code should read:

```
SELECT * FROM car WHERE price < 10000;
```

All other SQL statements should be written in the correct order. For example:

- `SELECT brand, avg(price)`
- `FROM car`
- `WHERE price < 100000`
- `GROUP BY brand`
- `ORDER BY brand`

Keep Learning to Avoid Common Syntax Errors in SQL!

Beginning mistakes can usually be corrected quickly. Don't get discouraged if you keep making the same common syntax errors in SQL – that's part of the learning process. An interactive SQL course like LearnSQL.com will help you find, fix, and outgrow your coding mistakes.

> Want to learn solid SQL Fundamentals? Click here to try out the first course from this track!

What Would You Add to This List?

What mistakes did you make as a beginning SQLer? What errors do you find yourself making when you're not careful? Share your thoughts and observations in the comments section below. Other SQL students will appreciate it!