

## Why Data Science?

### Exploring Data with SQL

4 min

Like most organizations, Codecademy uses **SQL (Structured Query Language)** to access its [database](#).

A **database** is a set of data stored in a computer. This data is usually structured into *tables*. Tables can grow large and have a multitude of columns and records.

Spreadsheets, like Microsoft Excel and Google Sheets, allow you to view and manipulate data directly: with selecting, filtering, sorting, etc. By applying a number of these operations you can obtain the subset of data you are seeking.

SQL (pronounced “S-Q-L” or “sequel”) allows you to write **queries** which define the subset of data you are seeking. Unlike Excel and Sheets, your computer and SQL will handle how to get the data; you can focus on what data you would like. You can save these queries, refine them, share them, and run them on different databases.

It is a great way to access data and a great entry point to programming because its syntax (the specific vocabulary that gives instructions to the computer) is very human-readable. Without knowing any SQL, you might still be able to guess what each command will do.

On her first day at Codecademy, Catherine wants to become familiar with the company’s data, so she connects to the database and uses SQL to explore the database.

#### Instructions

##### 1. Checkpoint 1 Passed

1.

One of the tables in Codecademy’s database is called `browse`. It contains information on each time someone visited the Codecademy’s website. Paste the following code into the code editor (middle panel) and click Run.

```
SELECT *  
FROM browse  
LIMIT 10;
```

This code will select all (\*) columns from `browse` table for the first 10 records.

Once you correctly enter the code and click Run, this instruction will turn green, letting you know that you completed this checkpoint.

Examine the data in the far right panel:

- What columns are there?
- What kinds of questions do you have about the Codecademy’s website?

The columns are:

- `user_id`
- `browse_date`
- `item_id`

test.sqlite

SELECT \* FROM browse LIMIT 10;

Query Results		
user_id	browse_date	item_id
336f9fdc-aaeb-48a1-a773-e3a935442d45	2017-12-20	3
336f9fdc-aaeb-48a1-a773-e3a935442d45	2017-12-20	22
336f9fdc-aaeb-48a1-a773-e3a935442d45	2017-12-20	25
336f9fdc-aaeb-48a1-a773-e3a935442d45	2017-12-20	24
4596bb1a-7aa9-4ac9-9896-022d871cdcde	2017-12-20	0
4596bb1a-7aa9-4ac9-9896-022d871cdcde	2017-12-20	2
2fdb3958-ffc9-4b84-a49d-5f9f40e9469e	2017-12-20	26
2fdb3958-ffc9-4b84-a49d-5f9f40e9469e	2017-12-20	24
fc394c75-36f1-4df1-8665-23c32a43591b	2017-12-20	12
fc394c75-36f1-4df1-8665-23c32a43591b	2017-12-20	24
Database Schema		
browse		
name	type	
user_id	TEXT	
browse_date	TEXT	
item_id	INTEGER	
Rows: 2352		

items	
name	type
id	INTEGER
item	TEXT
price	REAL
Rows: 33	
checkout	
name	type
user_id	TEXT
checkout_date	TEXT
item_id	INTEGER
Rows: 539	
purchase	
name	type
user_id	TEXT
purchase_date	TEXT
item_id	INTEGER
Rows: 472	