



DATA POWER! HOUR OF CODE LESSON PLAN AND TEACHER GUIDE



OVERVIEW

The world thrives on good data! We collect, sort, and analyze data on nearly everything. Knowing how to use data effectively improves our decision-making, whether we are troubleshooting a machine on a production line, looking for a good restaurant, or trying to solve global environmental problems.

In this hands-on NCLab tutorial, students write simple queries, the first step in data analysis. They will be introduced to the power of the PostgreSQL language, used by governments and businesses to manage database systems all over the world. PostgreSQL is written in plain English and is easy to learn. NCLab's system gives your students instant feedback on their queries so that they can immediately correct and resubmit them to the grader.

The only prerequisite is some comfort with typing commands. Although this tutorial introduces students to a serious language, they do not need to know coding or advanced math to be successful. The examples are taken from a database of world geography and are simple to follow and execute.

The exercises are simple enough for middle school students and older. However, we recommend this tutorial for high school students interested in exploring data as a career.

OBJECTIVES

Students will learn to:

Select data from a relational database by writing queries in PostgreSQL. Select specific tables and columns in a database.	Levels 1 and 2
Display data in order and set limits on the number of rows.	Levels 3, 4, and 5
Filter the data using specific criteria. Filter the data using AND, OR conditions.	Levels 6, 7 and 8
Use search criteria to find data.	Level 9
Aggregate data using COUNT, SUM, AVERAGE, MIN and MAX functions.	Level 10

CSTA Standard*: 3B-DA-06: Select data collection tools and techniques to generate data sets that support a claim or communicate information. **Concept:** Data & Analysis **Practice:** Communicating about Computing.

*Computer Science Teachers Association (2017). CSTA K–12 Computer Science Standards, Revised 2017. Retrieved from <http://www.csteachers.org/standards>. (For reference only. This lesson has not yet been submitted to CSTA for a crosswalk.)

MATERIALS AND PREP

Personal computers or tablets, with Internet access: one per student or student pair.

- Both PC and MAC platforms are supported.
- Use Chrome, Firefox, Safari, or Edge browsers. Internet Explorer is NOT supported and may cause problems with displaying graphics.
- Desktop computers, laptops and Chromebooks are preferred. Students find typing easier with a keyboard.
- iPad tablets are supported. Have students use their iPads in landscape mode for best results. Android tablets are not supported at this time.
- Phones are not supported.



Projector or Smartboard (optional) attached to a computer for demonstration or modeling.

Hour of Code Requirements: If you are doing this exercise as part of Hour of Code, please visit <https://hourofcode.com> for instructions on how to set up your class for a coding event. Students can select Data Power! on the Hour of Code Activities page by typing "NCLab" in the "Created By" menu, then selecting the graphic for the tutorial.

Direct Link: You can also use the tutorial as a standalone using this link:

<https://hoc.nclab.com/sql/>

YouTube videos: some schools block YouTube. The demonstration video embedded in the tutorial may need to be unblocked by an administrator to make it available to students.

Wrap up exploration of online databases, jobs in the data field, and PostgreSQL: (optional) There are many excellent resources on the Internet (see suggested resources). Most are free.

Printables: (optional) Students can take notes as they make their way through the tutorial and use the notes to complete an exit ticket. Printable materials are attached at the end of this lesson plan. Hour of Code also prints a certificate of completion.

Saved progress: If students would like to save their progress in the tutorial and keep it for reference, they will need to create a free NCLab account. The tutorial is intended for high school students, but if you plan to use it with students under the age of 13, please review your school's privacy policy. NCLab is COPPA compliant and does not collect or share personal information about students.

TEACHING TIPS

- Spend an hour or two ahead of time trying out the tutorial yourself.
- Understand basic sources of errors:
 - Missing SELECT statement
 - Missing semi-colon at the end of the statement.
 - Spelling
 - Logic and sequence of commands.
 - PostgreSQL is not case sensitive. However, we recommend the habit of writing keywords in all caps. This helps students see the keywords, and all caps is required in some SQL flavors.
 - Have the solution guide handy when you are teaching. You can find it at the end of this lesson plan and separately via this link:



[Solution Manual for Data Power \(SQL Hour of Code\)](#)

- Suggested internet explorations (addresses may change – check links before using):
 - Search terms: “Data vs. Information”; “Big Data”
 - Online databases such as:
 - [Our World in Data \(https://ourworldindata.org/\)](https://ourworldindata.org/)
 - Jobs and Careers
 - <https://www.dol.gov/general/topic/statistics/occupations>
 - Search the internet for “data jobs”. There are several excellent websites with career information.
 - The PostgreSQL website. <https://www.postgresql.org/>
- Decide whether your students will be working alone, in pairs or as teams, or if certain students would benefit with being partnered with a mentor student. One effective pair method is “Navigator/Pilot”. The Pilot is seated at the computer and controls the mouse and the keyboard. The Navigator stands next to the Pilot, reads all the instructions, and gives guidance. Students can switch roles for the next level or for the same level on the other student’s computer.

BASIC COMMANDS AND KEYWORDS

Note: Capitalization is not required but makes the code more readable. End full statements with a semi-colon.



COMMAND	EXAMPLE	PURPOSE
SELECT ... FROM	SELECT * FROM world.city;	Displays data from tables. In this example, all columns and all rows will be displayed from the table world.city
ORDER BY	SELECT name, population FROM world.country ORDER BY population DESC;	Displays data in order. Default is ascending order; add DESC for descending order.
LIMIT	SELECT * FROM world.city LIMIT 10;	Displays only the first <i>n</i> rows of the table.
WHERE	SELECT name, population FROM world.country WHERE population < 100000;	The WHERE clause filters data according to a specific condition.
AND		Filters for data that meet both conditions.
OR		Filters for data that meet at least one of the conditions.
LIKE, %	SELECT name FROM world.country WHERE name LIKE 'Ant%';	Searches for data using a search term. The wildcard % represents 0 or more additional characters and can be used at the beginning, middle, or end of the search term. Use quotation marks around the search term.
COUNT, SUM, AVG, MIN, MAX	SELECT AVG(population) FROM world.country;	Performs aggregate calculation on selected rows and displays the results: COUNT (number of rows); SUM (total value), AVG (average, or the SUM divided by the COUNT), MIN (lowest value), MAX (highest value).

VOCABULARY



Data vs. Information: Data (singular = datum) are items such as text or numbers that are not organized. Even very simple efforts to organize a set of numbers, for example a phone number xxx-xxx-xxxx or social security number xx-xxx-xxxx, gives it recognizable meaning. A phone number is information that we can use to contact someone.

Database: An organized set of data. Most databases are organized in tables of columns which represent meaningful categories (for example, a customer id, name, address, email, and phone number) and rows which represent instances (each customer). There are other approaches to organizing data such as dictionaries (key:value pairs), JSON documents, or graphic storage using edges and vertices. Database design depends on how the data is to be used.

Relational Database: A relational database connects tables using key columns. A table containing contact information about a customer may use a customer ID as a primary key. This customer ID key may also be used in a table containing transactions; in that table, it is a foreign key.

Structured Query Language (SQL): SQL is a programming language used to manage relational databases and extract data for specific purposes. It is pronounced 'sequel'. The language originated with E.F. Codd in 1970 as a solution for data management.

PostgreSQL: PostgreSQL is a SQL "flavor" described as an object-relational database. It uses basic SQL queries, but has evolved to address a range of information needs.

<https://www.postgresql.org/docs/6.3/c0101.htm>

Open Source: PostgreSQL is open source, meaning anyone can use the source code without a specifically granted license or fee. There are several advantages:

- Widespread adoption – anywhere, any time. SQL has been in use for 50 years, and PostgreSQL for about 30 years. <https://www.postgresql.org/docs/current/history.html>
- People contribute debugging, refinements, and applications.
- Sense of community due to this active involvement.
- People and corporations make money not on the code itself, but by being experts and solving problems for others.

PostgreSQL is particularly dedicated to this concept:

<https://www.postgresql.org/about/licence/>

Aggregate Function: A function that performs a computation on a set of values. In a table, this is the number of rows, subject to specified conditions. Examples of SQL aggregate functions include: SUM, COUNT, AVERAGE, MIN, MAX:

- SUM adds up the total value of a set of values.
- COUNT counts the number of items in a set of values.
- AVERAGE is the SUM divided by the COUNT. It is a measure of "central tendency".
- MIN finds the lowest value in a set of values.
- MAX finds the highest value in a set of values.

THE TUTORIAL AS A LESSON

Students can simply go through the tutorial as an activity. A quick run through might take one lesson period (about 45 to 55 minutes); a more thorough run with some side reading about two periods. The tutorial can be embedded in a longer course or used as a starting point for research using the suggested links.



NCLab offers full courses in data analytics in PostgreSQL and Python, taught primarily at two-year colleges as a certification. The tutorial can be used in that context as a taste of the PostgreSQL courses.

LESSON PLAN (60 TO 90 MINUTES)

EXPLORE AND DISCUSS (5 -10 MINUTES)

Several searches are suggested in the tutorial as students progress through it. A good starting point is to simply type a question in the search engine such as:

How much data is there?

What do we use data for?

How do we use data appropriately?

Sample opening: "Databases manage data systematically. In this tutorial, we learn how to write queries, which extract the data we need from a database." Refer to objectives and standards (page 1).

Hand out student guides, which include objectives, vocabulary and a notetaker (appended to this document). Encourage students to view the procedural video in the first level. If they are not able to view the video due to school restrictions on YouTube, you can view it as a class using this link:

<https://www.youtube.com/watch?v=w0LGKaVfEF0>

HOURLY OF CODE TUTORIAL (40 - 60 MINUTES)

Direct students to the Hour of Code activity (see instructions under Materials and Prep).

Students go through the tutorial on their own or in teams.

Decide whether you want stop points during the tutorial for discussion. For example:

- After Level 5, review filtering data. This is a good stop point if you are running the tutorial over two class periods.
- If you feel students need a review on basic aggregating, discuss SUM, COUNT, AVERAGE, MIN and MAX concepts before starting the last level. (refer to Vocabulary section)

WRAP UP DISCUSSION (5 - 10 MINUTES)

Possible discussion points:

1. **How data is organized** (tables, columns, rows). Tables contain related data which can be further related to data in other tables. Columns set criteria and data type. Rows represent instances.

2. **How data is filtered** (order, limits, conditions, search terms) Ordered results are easier to look up (alphabetical or numerical order), limits (tables can have millions of rows – may just want a small sample), conditions (for example, areas that have annual rainfall less than 150 mm. and average temperatures greater than 25 deg. C), and search terms (include or exclude characters, words, or other data from the results; use when we don't know the exact text).
3. **How data is aggregated.** Examples of total counts, sums, averages, minimum/maximum values in real life. (population count, total money in an account, average height, temperature limits for an industrial process)
4. How easy/difficult is the **coding** itself?
5. What **career paths** looked interesting?

Today, we participated in the Hour of Code, along with millions of other coders around the world. You took your first steps in writing SQL queries. When you look at any data, think about how it is organized, collected and filtered, whether you are viewing a graph in a news article, researching statistics, or making a decision based on data you have collected yourself.

ASSESSMENT (5 - 10 MINUTES)

Have students complete the exit ticket.

Note: the third question could be used as a starting point for a small project.

IMAGINE THAT YOU ARE COLLECTING DATA TO SOLVE A PROBLEM.

CREATE ONE POSSIBLE TABLE (COLUMN HEADINGS AND A SAMPLE ROW) THAT YOU WOULD USE. TRY WRITING A QUERY!

DIFFERENTIATION AND EXTENSION ACTIVITIES

Support:

- Model Level 1, then have the students start on the tutorial.
- Read instructions aloud. Also, read the code aloud.
- Use peer support, such as the Navigator/Pilot method.
- Remind students that they can redo a level as many times as needed to pass. Students must overcome the fears and frustrations associated with failed attempts. Failure is common and even desirable in computer programming, design, and engineering. Failure can be used to set limits and improve accessibility, speed, capacity, instructions, applications and more.

For students who finish the games early or would like to explore further after the lesson:

- Use the link when prompted to create an NCLab account. Experiment with the PostgreSQL app in Creative Suite/Programming/PostgreSQL. The world database and several others are available for practicing queries.
- Explore one of the suggested websites (databases, careers, PostgreSQL). Add questions to their notes.
- Be a roving "Expert" to help other students.

RESOURCES



SOLUTIONS FOR THE TUTORIAL LEVELS

Level	Topic	Solution Code
1	Getting started with SQL. Display data in a table.	<code>SELECT * FROM world.city;</code>
2	Select one column from a table.	<code>SELECT name, population FROM world.city;</code>
3	Display data in order.	<code>SELECT name, population FROM world.city ORDER BY population;</code>
4	Practice ordering data.	<code>SELECT name, population FROM world.city ORDER BY population DESC;</code>
5	Set limits on the data to be displayed.	<code>SELECT name, gnp FROM world.country ORDER BY gnp DESC LIMIT 10;</code>
6	Filter the data to be displayed.	<code>SELECT name, region, population FROM world.country WHERE region = 'Western Europe' ORDER BY population DESC;</code>
7	Select data based on conditions (AND).	<code>SELECT name, continent, indepyear FROM world.country WHERE indepyear >= 1800 AND continent = 'South America' ORDER BY indepyear;</code>
8	Select data based on conditions (OR).	<code>SELECT name, continent, population FROM world.country WHERE population < 1000 OR population > 1000000000 ORDER BY population DESC;</code>
9	Select data using search criteria.	<code>SELECT name, continent, population FROM world.country WHERE name LIKE 'B%a'</code>
10	Display aggregated data. Learn about NCLab courses and PostgreSQL.	<code>SELECT MIN(surfacearea) FROM world.country WHERE continent = 'Europe';</code>

The following pages are printable resources that you can share with your students.

1. Student guide to the tutorial
2. Vocabulary
3. Notetaker
4. Exit Ticket (Reflection)
 - a. Sample answers for Questions 2 and 3:
 - i. “Display data for country name and population from the table “Country” in the “World” database. Only show populations less than 100,000 from countries whose names begin with A. Order the data from greatest to least population.”

I learned that no one officially lives in Antarctica/ Why would people create a country that only has 8,000 people? / I don’t see the value of searching for countries that begin with the letter A. etc.

- ii. Sample table:

Online Store:

This table keeps track of T-shirt styles, sizes, colors and quantity ordered. The query is looking for styles that might not be popular and should be discontinued.

Database name: Streetgear

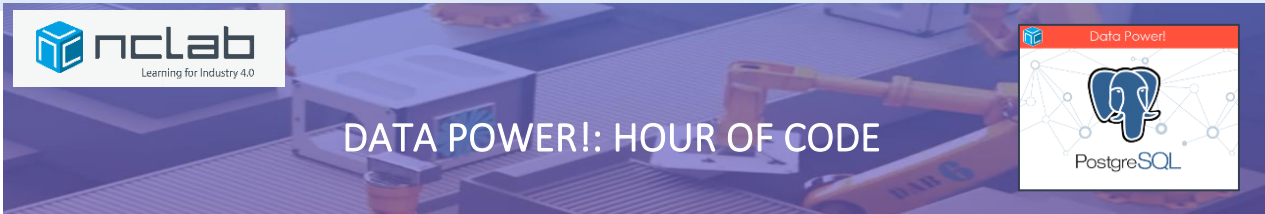
Table name: Tshirts

Style	Size	Color	Quantity_ordered
11530	XS	Black	3
...

Sample query:

```
SELECT style, quantity_ordered from street.gear.tshirts
WHERE quantity_ordered < 5
ORDER BY style;
```


STUDENT GUIDE



WHAT AM I LEARNING?

In this hands-on NCLab tutorial, you will write simple queries, the first step in data analysis. You will be introduced to the power of the PostgreSQL language, used by governments and businesses to manage database systems all over the world. PostgreSQL is written in plain English and is easy to learn.

WHY?

The world thrives on good data! We collect, sort, and analyze data on nearly everything. Knowing how to use data effectively improves our decision-making, whether we are troubleshooting a machine on a production line, looking for a good restaurant, or trying to solve global environmental problems.

HOW?

You will be learning how to write queries by completing these levels in the Tutorial. Along the way, you will explore some of the history of SQL, and about the important concepts behind these simple queries.

Select data from a relational database by writing queries in PostgreSQL. Select specific tables and columns in a database.	Levels 1 and 2
Display data in order and set limits on the number of rows.	Levels 3, 4, and 5
Filter the data using specific criteria. Filter the data using AND, OR conditions.	Levels 6, 7 and 8
Use search criteria to find data.	Level 9
Aggregate data using COUNT, SUM, AVERAGE, MIN and MAX functions.	Levels 10

VOCABULARY, NOTETAKERS, AND EXIT TICKET

The Vocabulary sheet: use as a reference.

The Notetaker: write notes, ideas, and questions as you go through the tutorial.

The Exit Ticket: reflect on the tutorial and try out your skills.

VOCABULARY

Data vs. Information: Data (singular = datum) are items such as text or numbers that are not organized. Even very simple efforts to organize a set of numbers, for example a phone number xxx-xxx-xxxx or social security number xx-xxx-xxxx, give data recognizable meaning and transforms them into information. A phone number is information that we can use to contact someone.

Database: An organized set of data. Most databases are organized in tables of columns which represent meaningful categories (for example, a customer id, name, address, email, and phone number) and rows which represent instances (each customer). There are other approaches to organizing data such as dictionaries (key:value pairs), JSON documents, or graphic storage using edges and vertices. Database design depends on how the data is to be used.

Relational Database: A relational database connects tables using key columns. A table containing contact information about a customer may use a customer ID as a primary key. This customer ID key may also be used in a table containing transactions; in that table, it is a foreign key.

Structured Query Language (SQL): SQL is a programming language used to manage relational databases and extract data for specific purposes. It is pronounced 'sequel'. The language originated with E.F. Codd in 1970 as a solution for data management.

PostgreSQL: PostgreSQL is a SQL "flavor" described as an object-relational database. It uses basic SQL queries, but has evolved to address a range of information needs.

<https://www.postgresql.org/docs/6.3/c0101.htm>

Open Source: PostgreSQL is open source, meaning anyone can use the source code without a specifically granted license or fee. There are several advantages:

- Widespread adoption – anywhere, any time. SQL has been in use for 50 years, and PostgreSQL for about 30 years. <https://www.postgresql.org/docs/current/history.html>
- People contribute debugging, refinements, and applications.
- Sense of community due to this active involvement.
- People and corporations make money not on the code itself, but by being experts and solving problems for others.

PostgreSQL is particularly dedicated to this concept:

<https://www.postgresql.org/about/licence/>

Aggregate Function: A function that performs a computation on a set of values. In a table, this is the number of rows, subject to specified conditions. Examples of SQL aggregate functions include: SUM, COUNT, AVERAGE, MIN, MAX:

- SUM adds up the total value of a set of values.
- COUNT counts the number of items in a set of values
- AVERAGE is the SUM divided by the COUNT. It is a measure of "central tendency".
- MIN finds the lowest value in a set of values.
- MAX finds the highest value in a set of values.

NOTETAKER



DATA POWER!: HOUR OF CODE

NAME: _____

WRITE BRIEF NOTES ON THE TUTORIAL. WHAT DID YOU LEARN?



LEVELS 1 AND 2

LEVELS 3, 4, AND 5

LEVELS 6, 7, AND 8

LEVEL 9

LEVEL 10

REFLECTION



DATA POWER!: HOUR OF CODE



NAME: _____

REFLECT ON WHAT YOU LEARNED AND TRY OUT YOUR SKILLS!

HOW WAS THE CODING ITSELF? WHAT WAS EASY AND WHAT WAS DIFFICULT?

REWRITE THIS QUERY IN PLAIN ENGLISH. WHAT DID YOU LEARN FROM THE RESULTS?

SELECT name, population FROM world.country
WHERE population < 100000 AND name LIKE 'A%'
ORDER BY population DESC;

name	population
Andorra	78000
American Samoa	68000
Antigua and Barbuda	68000
Anguilla	8000
Antarctica	0

IMAGINE THAT YOU ARE COLLECTING DATA TO SOLVE A PROBLEM. CREATE ONE POSSIBLE TABLE (COLUMN HEADINGS AND A SAMPLE ROW) THAT YOU WOULD USE. TRY WRITING A QUERY! HERE ARE SOME TOPIC SUGGESTIONS:

Online Store
Music Collection

Farm or Greenhouse Production
Climate/Ecosystem Issue

Product Test
Personal Budget

Topic and purpose of table:

Database name:			
Table name:			
Write column headings in the first row and a sample data entry in the second row:			

Sample Query: