2. Understanding the Relational Model

Contents

[1. Overview 1](#_Toc13542)

[2. Introduction to PostgreSQL 1](#_Toc9612)

[3. Introduction to Relational Databases 1](#_Toc10825)

[4. Summary 1](#_Toc24503)

# 1. Overview

=>slides: Pg. 1

In this module, we will discuss and understand the relational model. This model is at the core of many databases and is critical to understand when querying data from Postgres.

=>slides: Pg. 2

Data comes in many forms, and SQL is the language that we use to access that data. In this module, we will take a look at SQL and its special purpose in the data environment. We will discuss the environment we will use during these Postgres modules. We will also explore the relational database module.

=>slides: Pg. 3

How do we make the connection between data and our day-to-day work? Data is stored in a database. At its core, a database is a container that helps us to logically organize data. You can think of this quite simply. Card catalogs are a physical example of a database. A card catalog includes records for individual books that are organized by category and contain additional information about these items. Databases are most often relational in nature, which we will discuss later in this module. SQL is the language we use to interact with that data to analyze the data and to get the information that we need.

# Introduction to PostgreSQL

=>slides: Pg. 4

Unlike general programming languages like JavaScript or Python, SQL is a special purpose language. Its sole purpose is to interact with data. SQL, or Structured Query Language, is a platform that complies with the national standard ANSI, which allows for SQL to be used across databases. While database providers can add additional features such as PL/SQL and Oracle SQL or the MDX language in Microsoft SQL Server, the core SQL functionality is in line with the standard and is the same across platforms. This important feature will make it easy for you to learn other query languages or other database platforms once you've mastered SQL. Even most query languages that aren't pure SQL languages have a close relationship to SQL. During this course, however, we'll focus on ANSI-compliant SQL statements.

=>slides: Pg. 5

The database platform we will use for this course is Postgres. PostgreSQL complies with ANSI standards and also provides additional functionality. Postgres is an open-source and freely available database platform that is used by many startups and smaller enterprises. Other companies of all sizes are increasingly adopting Postgres due to its little operating costs and technical functionality. Importantly, Postgres includes a graphical user interface called pgAdmin. PgAdmin is the tool we will use throughout this course to author our SQL and to interact with our data. In addition to pgAdmin, there are many free and low-cost tools and programs available that interact across a variety of SQL-based platforms. As you become more proficient in working with data, you are likely to find a tool that you like best, but for now, we'll stick with the free pgAdmin tool.

# Introduction to Relational Databases

=>slides: Pg. 6

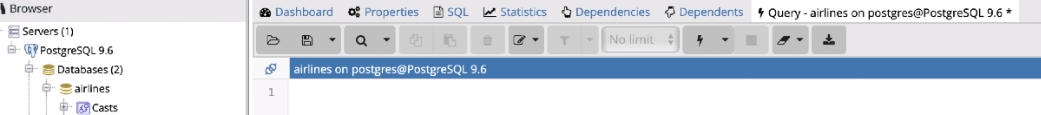
Understanding the relational database model is an important part of learning SQL. There are tables, there are columns, and there are rows. A table contains all records for a particular set of data. For example, you might have a table of employees, a table of telephone numbers, or a table of financial transactions. Within a table you have columns, which are the fields or variables in the dataset. For example, in a people table, you may have column for first name, a column for last name, or a column for preferred name. Rows are records within a table. For example, in a people table, John, Bob, Rose, and Scott may each have a row to store their data.

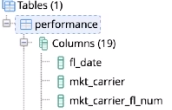
=>slides: Pg. 7

Let's look at an example. On this slide, we have a people table. This table include PersonID, FirstName, and LastName. We also have an address table, which includes AddressID, PersonID, and the Address. What's important about these tables is the relationship. The relationship is what makes a database relational. In this case, there's a relationship between the people table and the address table. Notice the PersonID in the people table. John Smith has an ID of 123 while Jane Doe has an ID of 124. These IDs tell the database what record we are interested in and uniquely identifies these records. Looking at the address table, we can see that each address also has an ID. The record with AddressID 9001 is 1 Main St. while the record with AddressID 9002 is 9 Cherry Dr. Notice that each record also has a PersonID. For example, AddressID 9003 has a PersonID of 125. These IDs are examples of what is known as a key. The PersonID is the primary key on the people table. It uniquely identifies each record. On the address table, the AddressID is a primary key; for example, AddressID 9003. For this record, we see that the secondary key is the PersonID 125. Keys are what allow us to connect information across tables. In this case, we can see that PersonID 125, Bob Clark, has the mailing address of 8 Mesa Ave. The PersonID is the same on both of these rows. This information allows us to join these records together to glean additional information. We will do this in later modules. Although database design is not the focus of this course, it is critical to remember that database design is important, as it will control what questions you and other database users have the ability to ask of the data later.

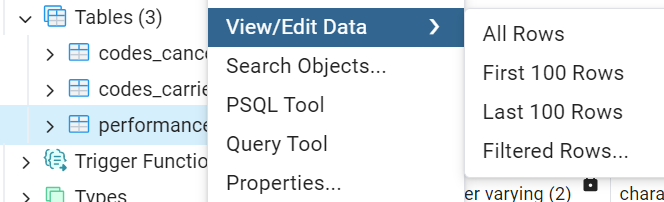
=>slides: Pg. 8

Now, let's take a look and explore some data structures using pgAdmin.

This window is the pgAdmin interface. Don't worry if the items on this screen look overwhelming. You'll learn how to use them in due course. For this example, I've loaded data from the Federal Aviation Administration for on-time performance for major United States air carriers during the month of January. Let's use this data to examine the structure of a database.



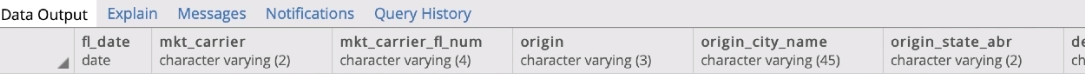
You'll notice in our airlines database that we have one table, performance. This table has 19 columns, the date of the flight, the market carrier, the flight number, the origin and destination, and performance statistics. It's easy to see this graphically in pgAdmin. This is the beginning of understanding the database structure. We have the airlines database. We have the performance table, which contains a great deal of information about the on-time performance of major airlines. Within that performance table, we have 19 columns that provide a variety of information. These columns are commonly referred to as fields or variables. We also have rows in the performance table. While in future modules we will learn the command syntax necessary to select the rows that we are interested in, pgAdmin allows you to view the rows by issuing commands graphically.



Right-click on performance, select View/Edit Data, and choose the First 100 rows option. PgAdmin will automatically generate the code to select all rows, or records, from our table.



To run this statement, we simply click the execute script icon, which is denoted by a play buttont. The query will retrieve the data that we are interested in. As you'll see, there are 15 rows visible if we go nearly full screen. We can see the 19 columns by scrolling from left to right. These columns provide a great deal of information about a number of flights. For example, these 15 rows contain information about United Airlines flights, the marketing carrier UA. We can see a variety of information ranging from their origin to their destination, their departure delays, arrival delays, and any cancellations. Each row refers to a separate flight.



For example, row 1 refers to United Airlines flight 2429, whereas row 6 refers to United Airlines flight 2422. Each row contains a record for a given flight. In this database, the combination of the marketing carrier and flight number columns is likely the primary key. That is the primary key is a unique identifier for each record. Now we have our complete database structure. We have a database on airlines, and within that database we have a performance table, which includes performance-related information. The 19 columns in this table make up details about performance, and each row refers to an individual record, in this case, a flight.

# Summary

=>slides: Pg. 9

In summary, relational databases are tools that allow us to store large quantities of information. Importantly, SQL is the tool that allows us to work with these databases. In future modules, we will learn how to use the SELECT statement to retrieve information and then use additional criteria, such as the WHERE clause, to filter the information that we retrieve.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*