



# Introduction to SQL

FinTech  
Lesson 7.1



# Learning Outcomes

---

By the end of this unit, you will be able to:

01

Create a data model to represent the objects and relationships in a dataset.

02

Create schemas, tables, and databases for relational data.

03

Retrieve data using advanced database queries.

# Class Objectives

---

By the end of today's class, you will:



Install PostgreSQL and pgAdmin on your computer.



Create databases and tables using pgAdmin.



Define SQL data types, primary keys, and unique values.



Load CSV files into a database.



Query data from a database.



Articulate the four basic functions of CRUD and apply them to a database.



Combine data from multiple tables using JOINS.

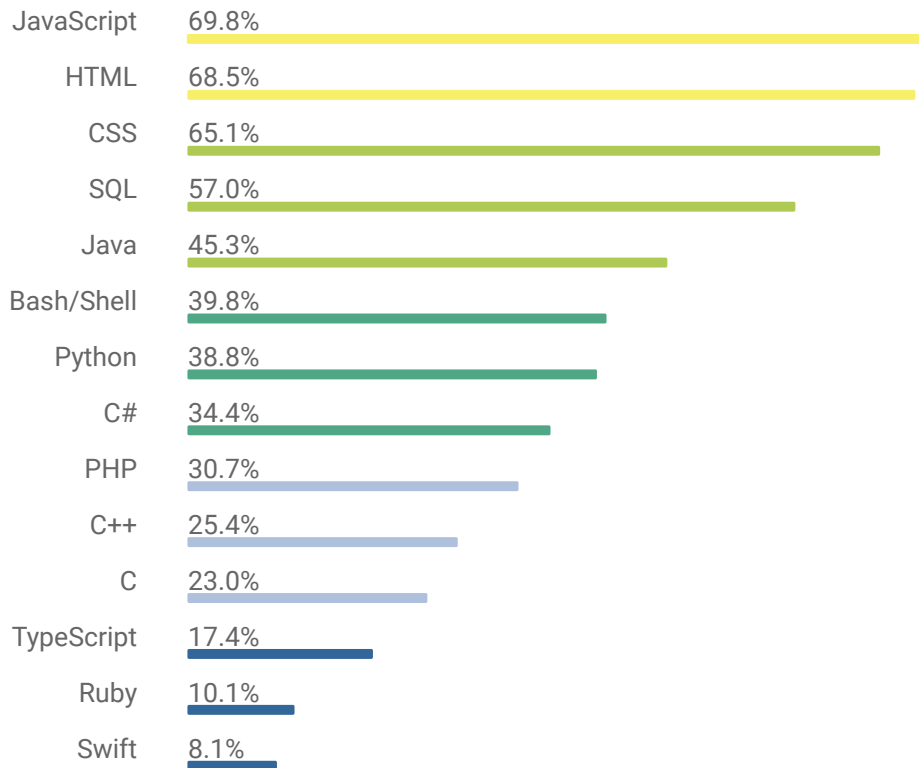
# Why SQL?

**Structured Query Language (SQL)** is one of the main query languages used to access data within relational databases.

**SQL** is designed to efficiently handle large amounts of data, resulting in high value to organizations.

Experienced **SQL** programmers are in high demand.

## Programming, Scripting, and Markup Languages *(all respondents)*



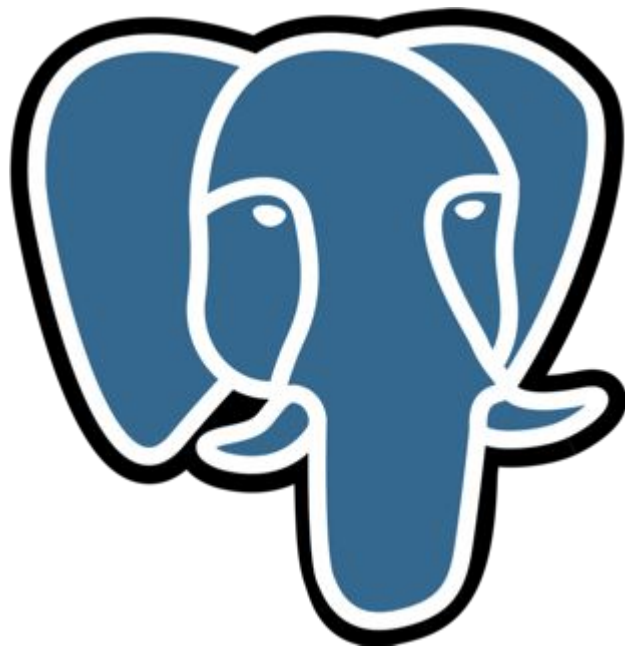
# Postgres and pgAdmin

# Postgres

---

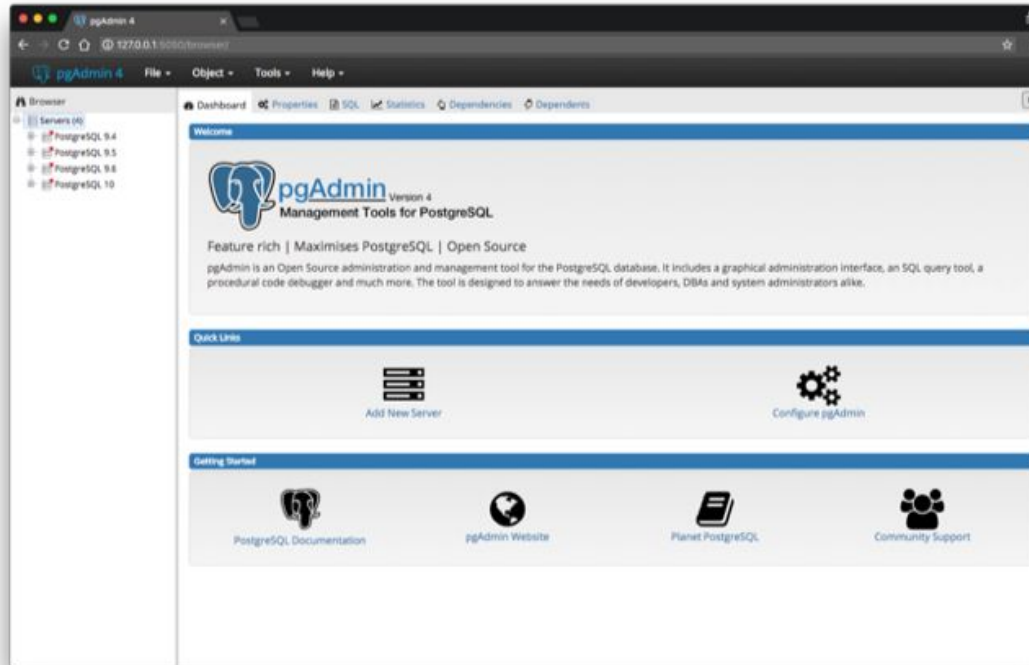
PostgreSQL (usually referred to as "Postgres") is an object-relational database system that uses the SQL language.

- Database engine
- Open source
- Great functionality



# pgAdmin

**pgAdmin** is the management tool used for working with Postgres. It simplifies creation, maintenance, and use of database objects.



# <Time to Code>





# Take a Break!

---



# CRUD

# CRUD Operations

---

**Create Read Update Delete** is a set of operations used with persistent storage.

<b>Create</b>	INSERT INTO table (column1, column2, column3)
<b>Read</b>	SELECT * FROM table
<b>Update</b>	UPDATE table SET column1 = VALUE WHERE id = 1
<b>Delete</b>	DELETE FROM table WHERE id = 5

These tools are fundamental to all programming languages, not just SQL.

# Wildcards

# Wildcards: % and \_

---

Wildcards are used to substitute zero, one, or multiple characters in a string. The keyword **LIKE** indicates the use of a wildcard.

```
SELECT *  
FROM actor  
WHERE last_name LIKE 'Will%';
```

The **%** will substitute **zero, one, or multiple** characters in a query.

For example, all of the following will match: **Will**, **Willa**, and **Willows**.

```
SELECT *  
FROM actor  
WHERE first_name LIKE '_AN';
```

The **\_** will substitute one, and only one, character in a query.

**\_AN** returns all actors whose first name contains three letters, the second and third of which are **AN**.

# <Time to Code>

