Project-based learning is a great way to quickly grasp **PostgreSQL** in a hands-on way. To make the most of a five-day learning sprint, focus on a single, well-defined project. This method helps solidify concepts by immediately applying them to a real-world problem.

Here's a possible five-day roadmap to follow:

Day 1: Setup and Fundamentals

- **Install PostgreSQL**: Download and install PostgreSQL and its client tool, **pgAdmin**. This is the graphical interface for managing your databases.
- **Database and Table Creation**: Create your first database and a simple table. A good starting point is a **"Student Management System"** or **"Library Management System."** Define columns and choose appropriate data types (e.g., VARCHAR for names, INTEGER for IDs).
- **Basic SQL Commands**: Learn the core CRUD operations (Create, Read, Update, Delete) using SQL commands. Practice INSERT, SELECT, UPDATE, and DELETE queries to add, retrieve, modify, and remove data from your table.

Day 2: Data Relationships

- **Primary and Foreign Keys**: Understand how to establish relationships between tables using **primary keys** and **foreign keys**. For your project, create an additional table (e.g., a "Courses" table for the Student Management System) and link it to your main table using a foreign key.
- **Joins**: Learn about different types of joins, such as INNER JOIN, LEFT JOIN, and RIGHT JOIN. Practice writing queries that combine data from both tables you created. For example, find all students enrolled in a specific course.

Day 3: Data Aggregation and Analysis

- **Aggregate Functions**: Master functions like COUNT, SUM, AVG, MIN, and MAX. Use these to perform calculations on your data. For instance, count the number of students, find the average age, or get the total number of books in the library.
- **GROUP BY and HAVING**: Learn how to group data for more meaningful analysis. Use GROUP BY to perform aggregate functions on specific groups (e.g., finding the number

of students in each course) and HAVING to filter those groups.

Day 4: Advanced Queries and Data Manipulation

- **Subqueries and CTEs (Common Table Expressions)**: Explore more complex queries using subqueries and CTEs. These are essential for breaking down complex problems into smaller, more manageable parts.
- **Views**: Create **views** to simplify complex queries. A view is a virtual table based on the result of a SQL query. This allows you to store a complex query for easy reuse.

Day 5: Project Finalization and Beyond

- **Refine Your Project**: Take your initial project and add more functionality. Consider features like a user system, a system for tracking grades, or a checkout system for books.
- **Connect with a Programming Language**: Learn how to connect your PostgreSQL database to a programming language like Python. Use a library like psycopg2 to write a simple script that interacts with your database, such as a script that adds new students or retrieves a list of all courses. This step bridges the gap between database management and application development.

This video is a beginner-friendly tutorial that covers the fundamentals of PostgreSQL, including installation and basic commands.

Learn PostgreSQL fast! | PostgreSQL for beginners
https://youtu.be/9ZixI8Xy0T0