

# Introduction to Databases

## Goals

- Define what a database is & give examples
- Describe and visualize a relational database
- Introduce PostgreSQL
- Create, list, and drop databases in PostgreSQL

## Databases

### Database

A database is a collection of data organized to be efficiently stored, accessed, and managed.

People will interchangeably use the term “database” to refer to database server software, the data records themselves, or the physical machine that is holding the data.

### Databases in the Web Stack

A typical web stack is a client, server, and database.

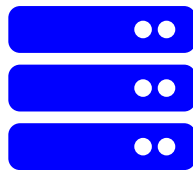
#### Client (Frontend)



[<\\_images/client.svg>](#)

- HTML/JS/CSS
- jQuery
- React
- Angular
- Vue
- ... and more!

#### Server (Backend)



[<\\_images/server.svg>](#)

- Flask
- Django
- Ruby on Rails
- Node.js/Express.js
- Java Spring
- ... and more!

#### Database (Backend)



[<\\_images/database.svg>](#)

- PostgreSQL
- MySQL
- SQLite
- MongoDB
- Cassandra
- ... and more!

## Relational Databases

- Model data as rows and columns of tabular data (like spreadsheets)

- Key terms include:
  - **RDBMS** - *Relational Database Management System*, e.g. PostgreSQL, MySQL, Oracle
  - **schema** - a logical representation of a database including its tables
  - **SQL** - Structured Query Language, a human-readable language for querying
  - **table** (*aka "relation"*) - basically a spreadsheet, a set of columns and rows
  - **column** - a table attribute, e.g. "first\_name", "last\_name", or "company"
  - **row** - a table record with values for set for columns, e.g. "Matt, Lane, Rithm"
- Nearly all relational DBs use SQL to modify/query data

**Note: How Do I Pronounce "SQL"?**

Both "sequel" and "S-Q-L" are valid pronunciations.

## Visualizing Relational Databases

### Example 1: A Books Table

id	title	author	page_count
1	The Design of Everyday Things	Don Norman	368
2	Flatland	Edwin Abbott	96
3	Power-Up: Unlocking Hidden Math in Video Games	Matt Lane	264

### Example 2: A Movies Table

id	title	release_year	runtime	rating
1	Star Wars: The Force Awakens	2015	136	PG-13
2	Avatar	2009	160	PG-13
3	Titanic	1997	194	PG-13

## PostgreSQL

PostgreSQL is the Open Source RDBMS we'll use here

- Powerful
- Popular
- Follows the SQL standard closely

On its website, PostgreSQL describes itself as an ORDBMS, that is, an object relational database management system. However the distinction between these two isn't important for us. If you're curious, you can [read about the difference](https://en.wikipedia.org/wiki/Object-relational_database#Comparison_to_RDBMS) <[https://en.wikipedia.org/wiki/Object-relational\\_database#Comparison\\_to\\_RDBMS](https://en.wikipedia.org/wiki/Object-relational_database#Comparison_to_RDBMS)>.

## Managing Databases

### PostgreSQL Shell

PostgreSQL includes ***psql***, a terminal-based program for issuing database commands.

```
$ psql my_database_name
psql (10.5)
Type "help" for help.

my_database_name=#
```

### Create Databases

Each project will have a separate database

Creating a database:

```
$ createdb my_database_name
```

Good database names are short and straightforward, and in ***lower\_snake\_case***.

### Where Is That Database?

The database **is not** a file in your current directory

It's a bunch of files/folders elsewhere on your computer

These aren't human-readable (*they're optimized for speed!*)

Saving your project in Git won't save your database!

### Seeding a Database

You can feed `.sql` scripts into the program ***psql***:

```
$ psql < my_database.sql
```

This is often used to ***seed*** an empty database by building tables, filling in rows, or both.

### Backing Up a Database

You can also make a backup of your database by “dumping it” to a file:

```
$ pg_dump -C -c -0 my_database_name > backup.sql
```

This makes a file in the current directory, `backup.sql`

It contains all of the commands necessary to re-create the current database when seeding.

## Dropping Databases

A database that is “dropped” is completely deleted (schema & data).

```
$ dropdb my_database_name
```

## Common Commands

- `\l` – List all databases
- `\c DB_NAME` – connect to *DB\_NAME*
- `\dt` -- List all tables (in current db)
- `\d TABLE_NAME` – Get details about *TABLE\_NAME* (in current db)
- `\q` – Quit *psql* (can also type **<Control-D>**)

## SQL

```
SELECT * FROM people WHERE age > 21 AND id IS NOT NULL;
```

- SQL (Structured Query Language) is a human-readable language for relational databases.
- Strings in SQL:
  - case-sensitive: `'Bob'` not same as `'bob'`
  - single-quotes, not double, around strings.
- Commands end with a semicolon.
- SQL keywords are conventionally written in ALL IN CAPS but not required.
  - `SELECT price FROM books` same as `select price from books`

## Types of SQL Statements

### DDL: Data Definition Language

The commands to create, modify, or delete tables, columns & databases.

Examples: **CREATE**, **ALTER**, **DROP**

### DML: Data Manipulation Language

The commands to create, read, modify, or delete rows.

Examples: ***SELECT, INSERT, UPDATE, DELETE***