



# A Case Study of Extract, Transform, Load

Data Boot Camp  
Lesson 13.1





Instructor Demonstration  
Welcome Students



# The Week Ahead!

---

## ▼ Day One (Today)

/ Intro to the ETL process - working through activities

### ▼ Intro to the ETL Project

/ Goals

/ Requirements

### ▼ Working towards a feasible project idea with

/ Instructors

/ TAs

### ▶ Submit Project Proposal

## ▼ Day Two

/ Working on projects

/ Full assistance from Instructors and TAs

## ▼ Day Three

/ Project due date - Presentation!

/ Discussion



# Instructor Demonstration

## Introduction to the Case Study Project

# Data Sources: Introduction to the Case Study Project

---

- It must come from two (minimum) or more sources.
- Recommended sources:
  - Kaggle.
  - Data.world.
  - Google Dataset Search (<https://datasetsearch.research.google.com/>).
  - As an alternatively source you may use APIs.
- Once you datasets are identified - Perform ETL and documentation:
  - Documentation must have:
    - Datasets used and their sources.
    - Types of data wrangling performed - Data cleaning, joining, filtering and aggregating.
    - The schemata used in the final production database.



# Instructor Demonstration

## Introduction to ETL

# Introduction to ETL

---

## ETL

Data integration is an important part of working with data.



# Introduction to ETL

## Extract

Data may come from disparate sources, such as:

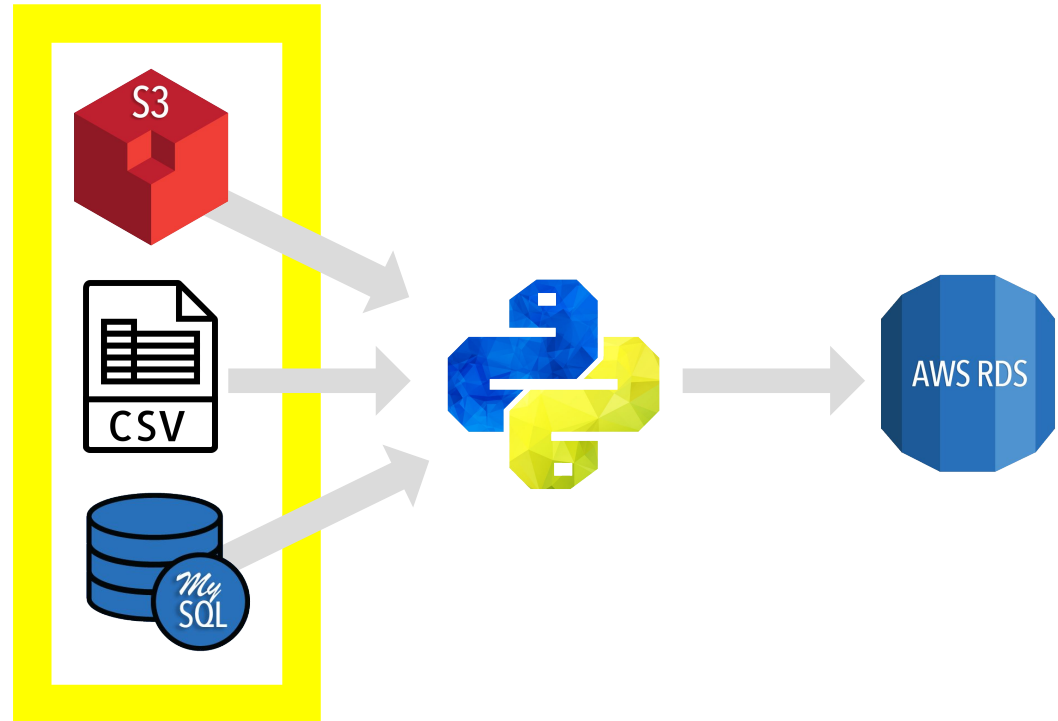
CSV files

JSON files

HTML tables

SQL databases

Spreadsheets



**Extract**

# Introduction to ETL

## Transform

Transform the data to suit business needs.  
This may include:

Data Cleaning

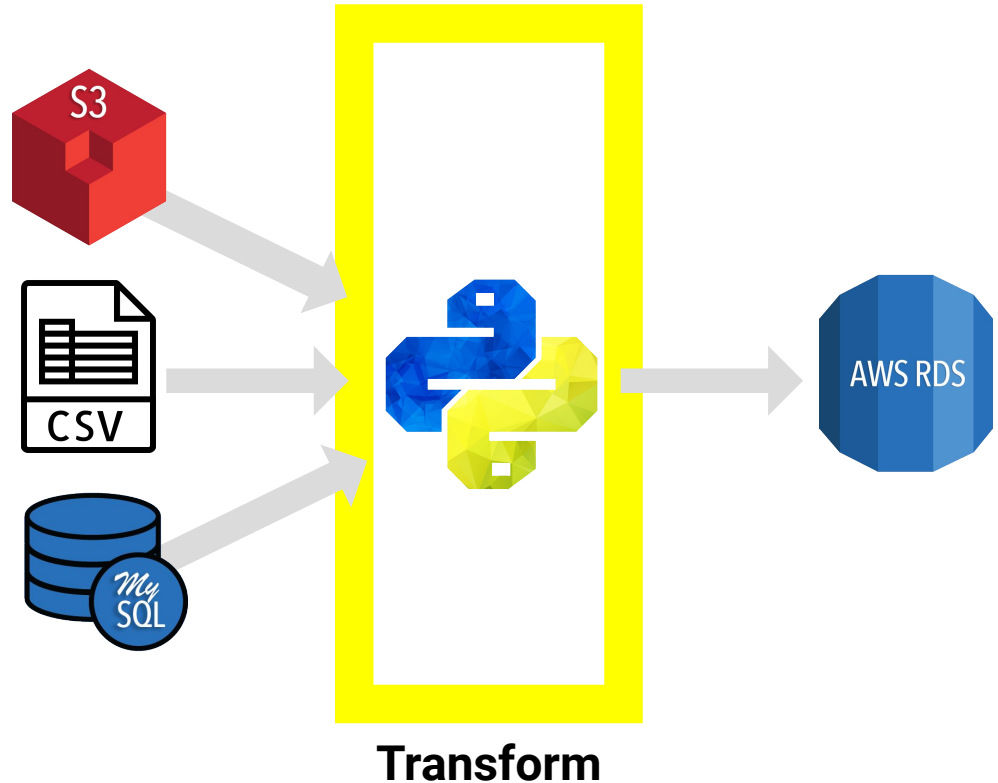
Summarization

Selection

Joining

Filtering

Aggregating





**Note:** We will use Python and pandas for transformation, which can also be done with SQL or a specialized ETL tool.

# Introduction to ETL

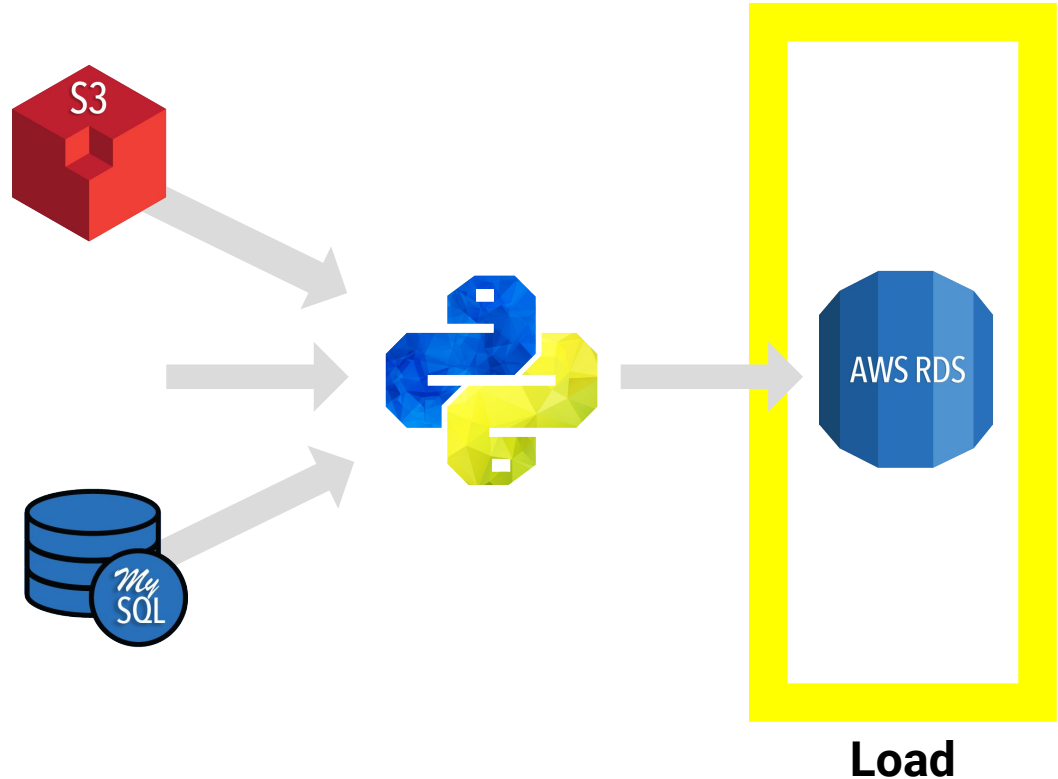
## Load

Load the data into a final database that can be used for future analysis or business use.

Can be a relational or non-relational database

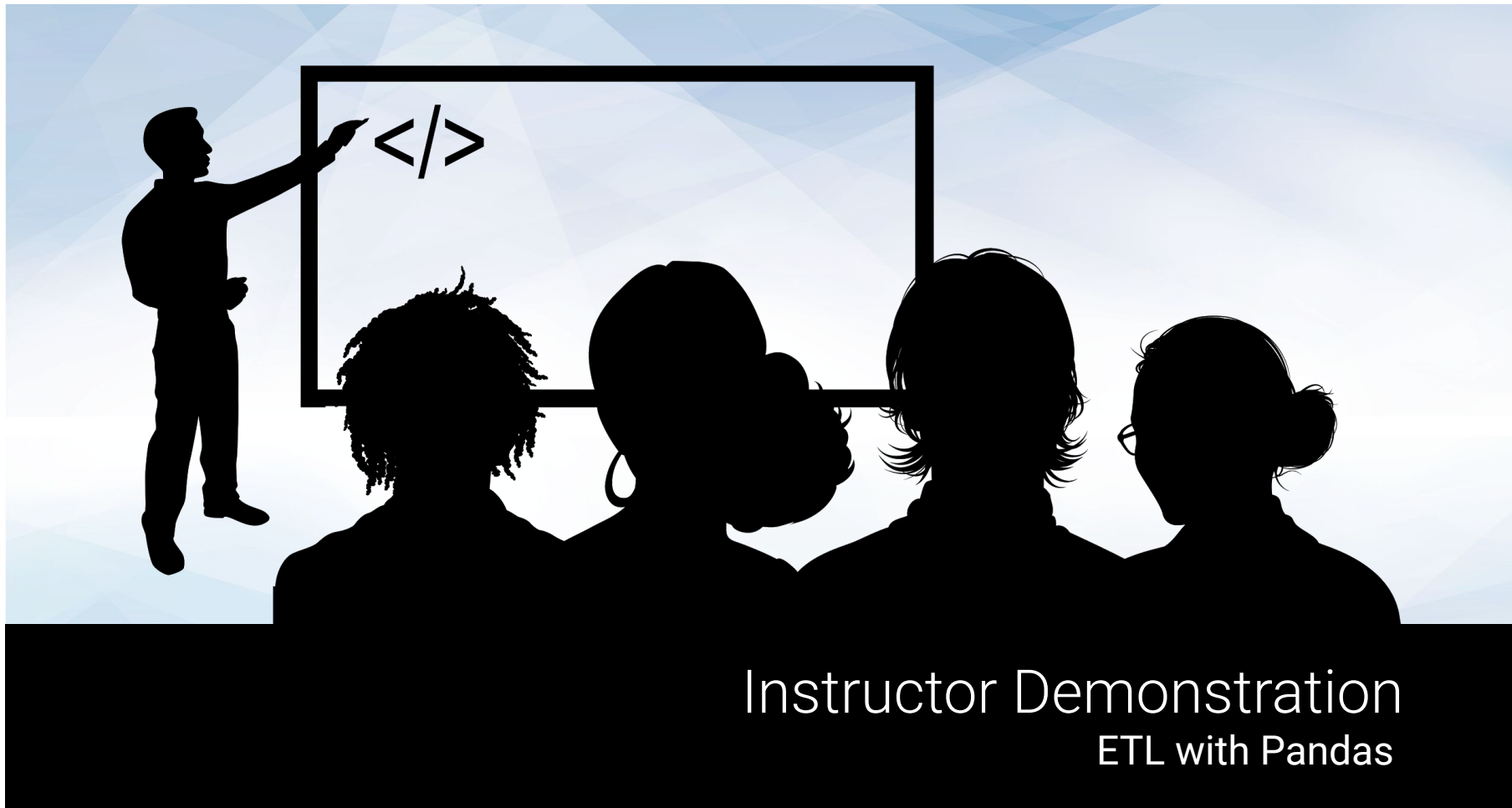
Can be local or in the cloud

Can be a data lake or data warehouse



# Questions?





# Instructor Demonstration

## ETL with Pandas

# ETL with Pandas

---

- Not limited to **Pandas** the **ETL** process is performed with a variety of tools and file formats.
- For this demonstration we will use the following:



 pandas



PostgreSQL



**Couple of things to prepare  
before we move forward!  
Let's find out what it is!**



# ETL with Pandas

---

`pip install psycopg2`

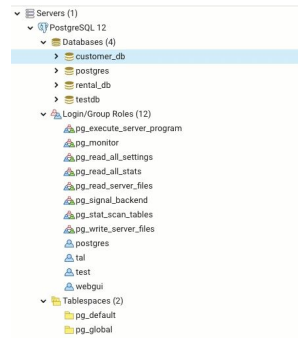
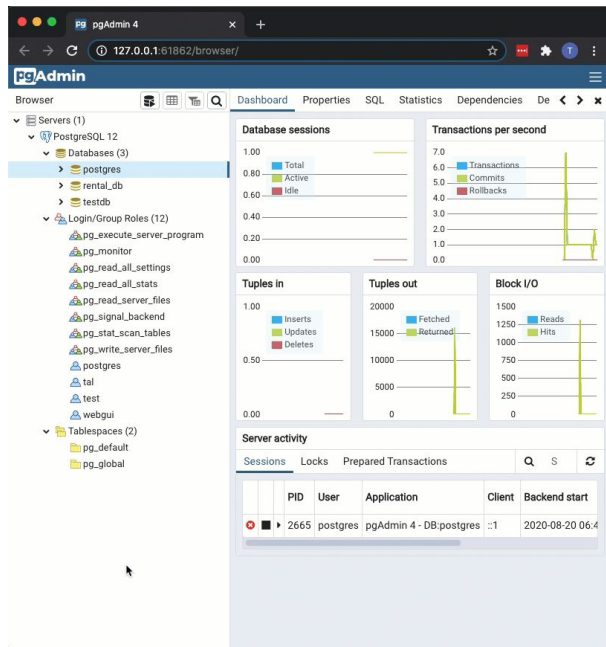
- `pip install psycopg2`
- Psycopg is a package – An adapter for Python that works as a wrapper for **libpq**, which is the official PostgreSQL client library.



# ETL with Pandas

## pgAdmin postgresSQL

- Second, we need to open pgAdmin 4 and connect to a local server. Once connected we need to create a new **database** and **tables** accordingly.



database



tables

```
1 CREATE TABLE customer_name (  
2   id INT PRIMARY KEY,  
3   first_name TEXT,  
4   last_name TEXT  
5 );  
6  
7 CREATE TABLE customer_location (  
8   id INT PRIMARY KEY,  
9   address TEXT,  
10  us_state TEXT  
11 );
```

# ETL with Pandas

## .ipynb (Jupyter Notebook)

- Pandas is the pivotal piece of the ETL process. In Pandas, we are going to extract the data, Transform it and load back into dataframes. Let's follow the code line by line and see how it is done.

### Store CSV into DataFrame

```
In [2]: csv_file = "../Resources/customer_data.csv"
customer_data_df = pd.read_csv(csv_file)
customer_data_df.head()
```

Out[2]:

|   | id | first_name | last_name | email                      | gender | car           |
|---|----|------------|-----------|----------------------------|--------|---------------|
| 0 | 1  | Benetta    | Cancott   | bcancott0@studiopress.com  | Female | Scion         |
| 1 | 2  | Lilyan     | Cherry    | lcherry1@deliciousdays.com | Female | Chrysler      |
| 2 | 3  | Ezekiel    | Benasik   | ebenasik2@wikia.com        | Male   | Mercedes-Benz |
| 3 | 4  | Kennedy    | Atlay     | katlay3@so-net.ne.jp       | Male   | Buick         |
| 4 | 5  | Sanford    | Salmen    | ssalmen4@reuters.com       | Male   | Lincoln       |

→ Cell #2 has data pulled from a csv file and the data is assigned to a variable called `customer_data_df`.

→ Cell #3 is returning a new dataframe with only the necessary columns. The new dataframe is assigned to a new variable as well.

### Create new data with select columns

```
In [3]: new_customer_data_df = customer_data_df[['id', 'first_name', 'last_name']].copy()
new_customer_data_df.head()
```

Out[3]:

|   | id | first_name | last_name |
|---|----|------------|-----------|
| 0 | 1  | Benetta    | Cancott   |
| 1 | 2  | Lilyan     | Cherry    |
| 2 | 3  | Ezekiel    | Benasik   |
| 3 | 4  | Kennedy    | Atlay     |
| 4 | 5  | Sanford    | Salmen    |

# ETL with Pandas

## .ipynb (Jupyter Notebook)

### Store JSON data into a DataFrame

```
In [4]: json_file = "../Resources/customer_location.json"
customer_location_df = pd.read_json(json_file)
customer_location_df.head()
```

```
Out[4]:
```

|   | address               | id | latitude | longitude | us_state |
|---|-----------------------|----|----------|-----------|----------|
| 0 | 043 Mockingbird Place | 1  | 39.1682  | -86.5186  | Indiana  |
| 1 | 4 Prentice Point      | 2  | 41.0938  | -85.0707  | Indiana  |
| 2 | 46 Derek Junction     | 3  | 32.7673  | -96.7776  | Texas    |
| 3 | 11966 Old Shore Place | 4  | 39.0350  | -94.3567  | Missouri |
| 4 | 5 Evergreen Circle    | 5  | 40.7808  | -73.9772  | New York |

### Clean DataFrame

```
In [5]: new_customer_location_df = customer_location_df[["id", "address", "us_state"]].copy()
new_customer_location_df.head()
```

```
Out[5]:
```

|   | id | address               | us_state |
|---|----|-----------------------|----------|
| 0 | 1  | 043 Mockingbird Place | Indiana  |
| 1 | 2  | 4 Prentice Point      | Indiana  |
| 2 | 3  | 46 Derek Junction     | Texas    |
| 3 | 4  | 11966 Old Shore Place | Missouri |
| 4 | 5  | 5 Evergreen Circle    | New York |

→ The same process of extracting and transforming the data is repeated with the JSON file as well.

# ETL with Pandas

## .ipynb (Jupyter Notebook)

### Connect to local database

```
In [6]: rds_connection_string = "<insert user name>:<insert password>@localhost:5432/  
customer_db"  
engine = create_engine(f'postgresql://{rds_connection_string}')
```

### Check for tables

```
In [7]: engine.table_names()
```

```
Out[7]: ['customer_location', 'customer_name']
```

### Use pandas to load csv converted DataFrame into database

```
In [8]: new_customer_data_df.to_sql(name='customer_name', con=engine, if_exists='appe  
nd', index=False)
```

### Use pandas to load json converted DataFrame into database

```
In [9]: new_customer_location_df.to_sql(name='customer_location', con=engine, if_exis  
ts='append', index=False)
```

→ The following step is to connect to the local database. Once connected, we check the tables created earlier in the process.

→ Next, we are dumping the newly created and trimmed dataframes into the database.

# ETL with Pandas

## .ipynb (Jupyter Notebook)

Confirm data has been added by querying the `customer_name` table

- NOTE: can also check using pgAdmin

```
In [10]: pd.read_sql_query('select * from customer_name', con=engine).head()
```

```
Out[10]:
```

|   | id | first_name | last_name |
|---|----|------------|-----------|
| 0 | 1  | Benetta    | Cancott   |
| 1 | 2  | Lilyan     | Cherry    |
| 2 | 3  | Ezekiel    | Benasik   |
| 3 | 4  | Kennedy    | Atlay     |
| 4 | 5  | Sanford    | Salmen    |

Confirm data has been added by querying the `customer_location` table

```
In [11]: pd.read_sql_query('select * from customer_location', con=engine).head()
```

```
Out[11]:
```

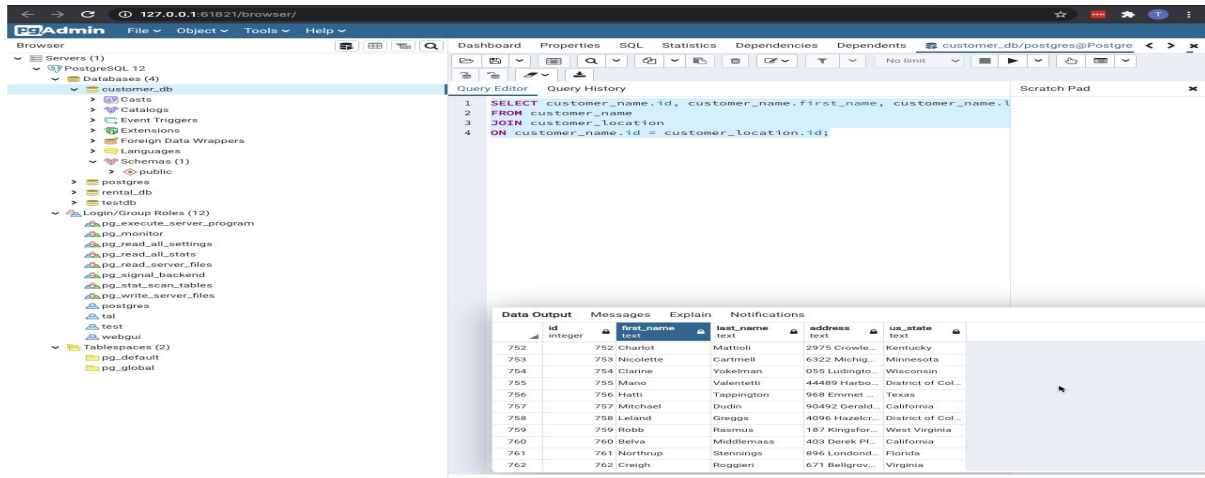
|   | id | address               | us_state |
|---|----|-----------------------|----------|
| 0 | 1  | 043 Mockingbird Place | Indiana  |
| 1 | 2  | 4 Prentice Point      | Indiana  |
| 2 | 3  | 46 Derek Junction     | Texas    |
| 3 | 4  | 11966 Old Shore Place | Missouri |
| 4 | 5  | 5 Evergreen Circle    | New York |

- At this point, all the data that we extracted and transformed is successfully loaded into our PostgreSQL database.
- To double check, as a best practice, we performed queries for both tables at the database.

# ETL with Pandas

## pgAdmin postgresSQL

- The last piece of the process is coming back to pgAdmin to perform the join of the two tables we created.



```
SELECT customer_name.id, customer_name.first_name, customer_name.last_name, customer_location.address,
customer_location.us_state
FROM customer_name
JOIN customer_location
ON customer_name.id = customer_location.id;
```



## Activity: Pandas ETL

In this activity, you will have the opportunity to perform your very first ETL process.

**Suggested Time:**  
20 Minutes





# Activity: Pandas ETL

---

## Instructions:

- Create a `customer_db` database in pgAdmin 4 then create the following two tables within:
  - A premise table that contains the columns `id`, `premise_name` and `county_id`.
  - A county table that contains the columns `id`, `county_name`, `license_count` and `county_id`.
  - Be sure to assign a primary key, as Pandas will not be able to do so.
- In Jupyter Notebook perform all ETL.
  - ➔ **Extraction**
    - ◆ Put each CSV into a pandas DataFrame.
  - ➔ **Transform**
    - ◆ Copy only the columns needed into a new DataFrame.
    - ◆ Rename columns to fit the tables created in the database.
    - ◆ Handle any duplicates. **HINT**: some locations have the same name but each license number is unique.
    - ◆ Set index to the previously created primary key.

# Activity: Pandas ETL

---

## Instructions:

### → Load

- ◆ Create a connection to database.
  - ◆ Check for a successful connection to the database and confirm that the tables have been created.
  - ◆ Append DataFrames to tables. Be sure to use the index set earlier.
- Confirm successful **Load** by querying database.
  - Join the two tables and select the `id` and `premise_name` from the `premise` table and `county_name` from the `county` table.



**Time's Up!** Let's Review.





Countdown timer

**15:00**

(with alarm)



## Activity: Project Proposals

In this activity, you and your fellow group members will be going over the ETL project guidelines.

**Suggested Time:**  
90 Minutes



# Activity: Project Proposals

---

## Team Effort

Due to the short timeline, teamwork will be crucial to the success of this project! Work closely with your team through all phases of the project to ensure that there are no surprises at the end of the week.

Working in a group enables you to tackle more difficult problems than you'd be able to working alone. In other words, working in a group allows you to work smart and dream big. Take advantage of it!

## Project Proposal

Before you start writing any code, remember that you only have one week to complete this project. View this project as a typical assignment from work. Imagine a bunch of data came in and you and your team are tasked with migrating it to a production database.

Take advantage of your Instructor and TA support during office hours and class project work time. They are a valuable resource and can help you stay on track.

# Activity: Project Proposals

---

## Finding Data

Your project must use 2 or more sources of data. We recommend the following sites to use as sources of data:

- [Data.world](#)
- [Kaggle](#)

You can also use APIs or data scraped from the web. However, get approval from your instructor first. Again, there is only a week to complete this!

# Activity: Project Proposals

---

## Data Cleanup & Analysis

Once you have identified your datasets, perform ETL on the data. Make sure to plan and document the following:

- The sources of data that you will extract from.
- The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).
- The type of final production database to load the data into (relational or non-relational).
- The final tables or collections that will be used in the production database.

You will be required to submit a final technical report with the above information and steps required to reproduce your ETL process.



# Activity: Project Proposals

---

## Project Report

At the end of the week, your team will submit a Final Report that describes the following:

- **Extract:** your original data sources and how the data was formatted (CSV, JSON, pgAdmin 4, etc).
- **Transform:** what data cleaning or transformation was required.
- **Load:** the final database, tables/collections, and why this was chosen.

Please upload the report to Github and submit a link to Bootcampspot.

"We set sail on this new sea because there is new knowledge to be gained..."

# We choose to go to the Moon!

We choose to go to the Moon...

...not because they are easy, but because they are  
**HARD...**

Parts of President John F. Kennedy  
address at the Rice University on the  
Nation's Space Effort delivered on September 12, 1962.

