7.2.3

Import Data

Alright, we've created a database. We've written our first SQL code and created tables modeled after our ERD. Bobby is almost ready to begin his analysis. The next step is to import the data from the CSV files. We'll make sure all of the tables we created in pgAdmin appear in the GUI first, because we'll be using the GUI to import data. The import itself should go smoothly, thanks to the thought and careful attention to detail we put into our diagram and table creation.

Now that Bobby's tables are ready, let's begin importing data.

IMPORTANT

SQL is very interactive. Developers are not only importing data and asking it questions through the query language, but they can also update and edit the data stored in the tables as needed.

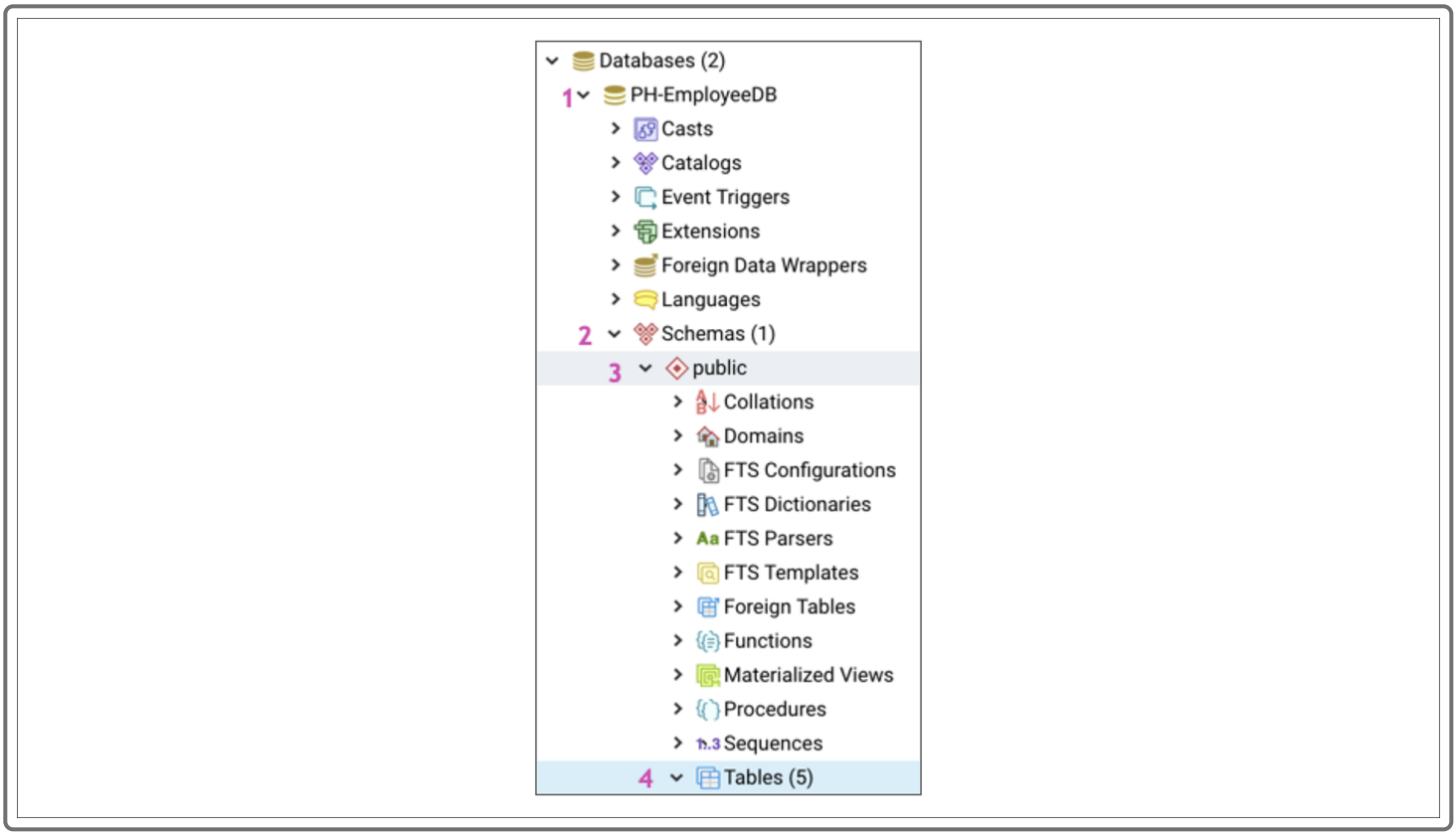
For example, if a single row of data needs to be added to an existing table, a developer can manually add it by using the `INSERT` statement.

If the data in a table is small enough in scale, it can be manually inserted this way completely, instead of importing a CSV file.

Alternately, necessary edits and updates are completed manually as well. We won't be manually editing or uploading data to our tables in this lesson because our datasets are too large.

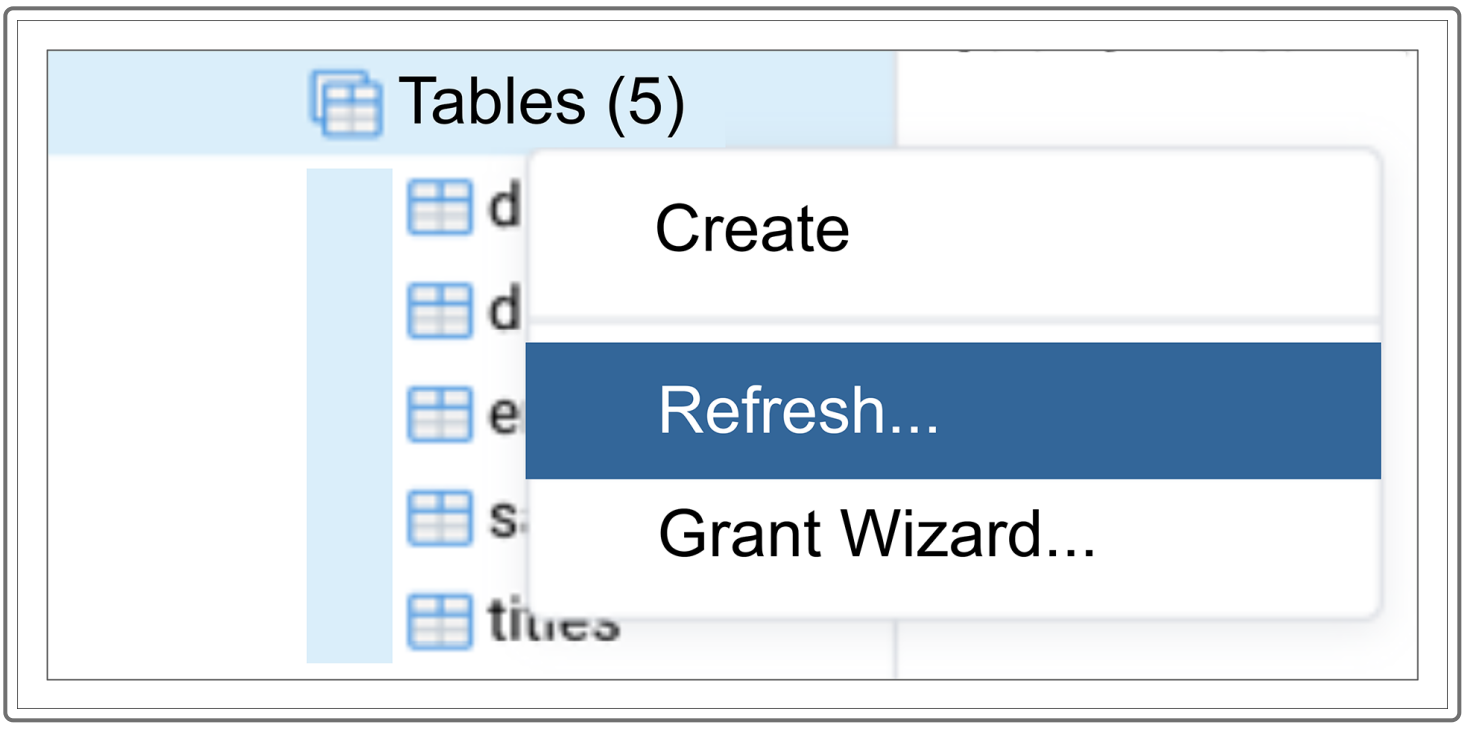
In the pgAdmin window, select the dropdown menu for our PH-EmployeeDB database. To import data into the tables, first confirm all of our tables are listed:

1. Find the PH-EmployeeDB collapsible menu and click it.
2. Scroll down and click "Schemas" to expand the menu.
3. Click "public."
4. Scroll down to "Tables" and note the number in parentheses.



We created six tables, one for each CSV file. If all six don't appear in parentheses, there are two things you can do:

1. Make sure you executed the `CREATE TABLE` statements for each by highlighting the specific code and clicking the lightning bolt on the pgAdmin toolbar.
2. Right-click the "Tables" menu and click "Refresh."

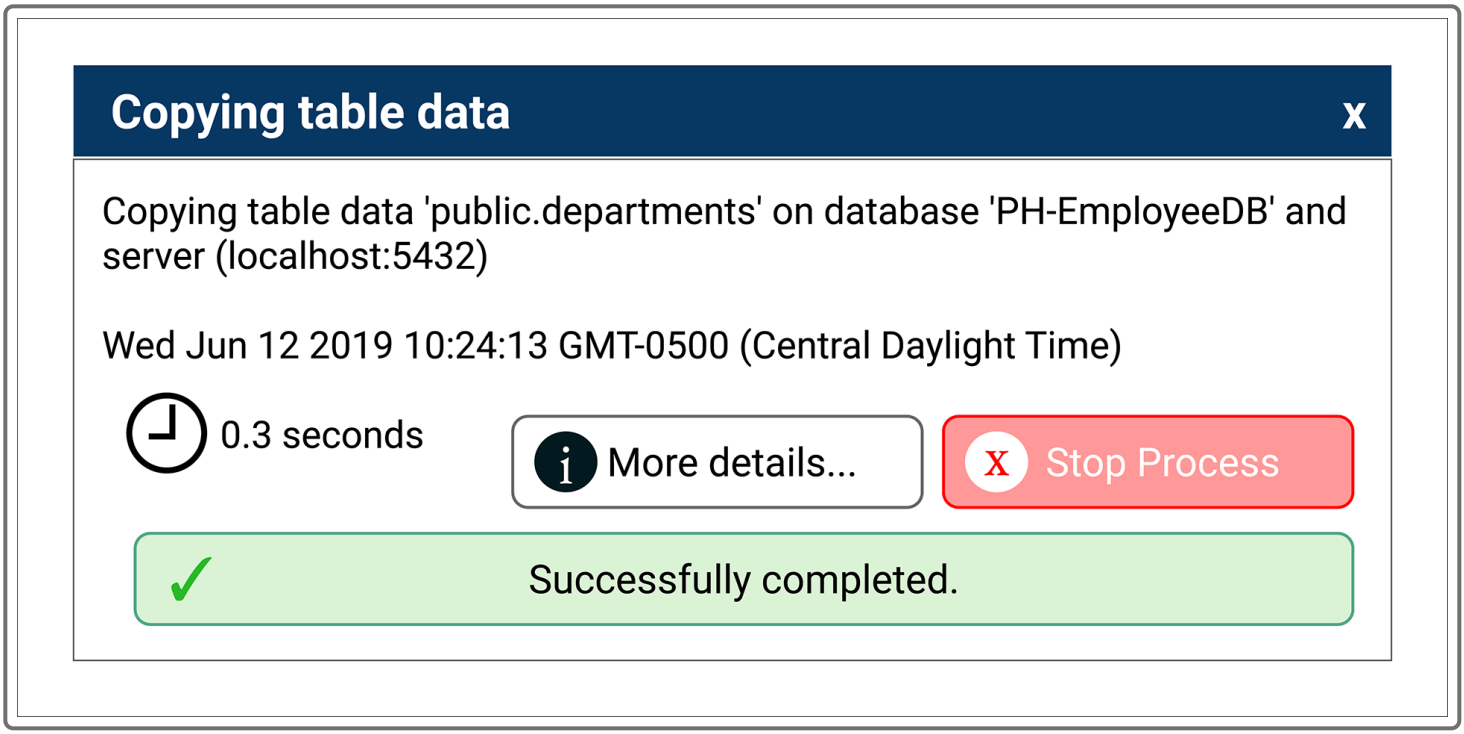


Now that the database is updated, we can insert data. We need each table visible from the dropdown menu because we are importing data directly into each table without using the query editor.

To import a CSV into Postgres with pgAdmin, follow these steps. We'll customize our options to fit our data import, and then check the table to make sure the data has been imported successfully.

1. Right-click the first table, departments.
2. From the menu that pops up, scroll to Import/Export.
3. Toggle the button to show "Import."
4. Click the ellipsis on the Filename field to search for your project folder.
5. Select departments.csv. Make sure Format is set to "csv" and Encoding is blank. **Note:** By default, the Encoding section is blank. If our files were encoded to provide an extra layer of security, we would need to select the type of encoding before importing them to Postgres. We don't have to worry about this, though. Also, if "Encoding" is filled in with an encoding type such as BIG5 or LATIN1, cancel the import and start over.
6. Leave the OID field as is, but toggle the Header field to "Yes" and select the comma as the Delimiter. **Note:** If we don't specify that there is already a header included in the CSV data, then the header will be imported as data. This would result in errors because headers don't always match the data types in the columns.
7. Click OK to begin importing the data.

If the import is successful, a pop-up window will appear at the bottom of your pgAdmin page:



Check the import by typing `SELECT * FROM departments;` at the bottom of the query editor. The resulting table should mirror the CSV file:

```
56  SELECT * FROM departments;
```

Data Output | Explain | Messages | Notifications

	dept_no character varying (4)	dept_name character varying (40)
1	d001	Marketing
2	d002	Finance
3	d003	Human Resources
4	d004	Production
5	d005	Development
6	d006	Quality Management
7	d007	Sales
8	d008	Research

If you see all nine departments, great job. If not, then there is probably an error somewhere in the table creation.

IMPORTANT

If you don't see the same nine tables as shown in the graphic above, then it's time to troubleshoot. Look for error messages in pgAdmin (this will be obvious and will pop up during the import). Also check the data itself. Is the file corrupted? Is anything missing?

Also, search Google for answers. Have a specific, confusing error? Copy and paste the error message in the browser search field to find online forums discussing the same problems.

Developing troubleshooting skills is critical to mastering any coding language, which is a lifelong process. The tools and libraries we use are often updated or changed in minor ways that can produce confounding bugs, so it's good to seek out help when encountering errors. Developers often turn to the Stack Overflow community for help resolving such dilemmas. This community is welcoming to users with all levels of experience—they were new once, too!

We'll also be covering error handling in the next section, so don't worry if you've run into an error here.

Import data into the dept_manager table next. Follow the same steps as before to import data, only this time, right-click on the dept_manager table in the dropdown menu.

Remember to:

1. Find the correct CSV file to upload using the "filepath" field.
2. Toggle the "Header" field to "Yes."
3. Select a comma as the delimiter.

Click OK to upload the CSV file.

SKILL DRILL

Import the data from each CSV into its corresponding table. The remaining tables may have different requirements for primary and foreign keys, so make sure to check out the data first before creating the table. Also, if you need to re-create a table, remember to use the DROP TABLE statement first.