

7.1.2 Identifying Data Relationships

Bobby has all of his tools set up, which is great. The installation wasn't too terrible, and he's ready to start creating databases and importing the data. Except that's a little hasty. A new project is exciting, but let's slow down and take a breath, then look at the data we'll be importing.

By taking a quick look at each CSV, we will have a better understanding of what our data actually looks like. What data types are involved? How many CSV files are there? Is the data all easy to read? By answering these questions early, we'll know if we need to make any adjustments to the data before importing it.

While we're cozying up to the data, let's also look at how the different CSV sheets are connected. Some columns will appear in more than one CSV. We'll take a deeper look into these connections, called primary and foreign keys.

Let's begin the download.

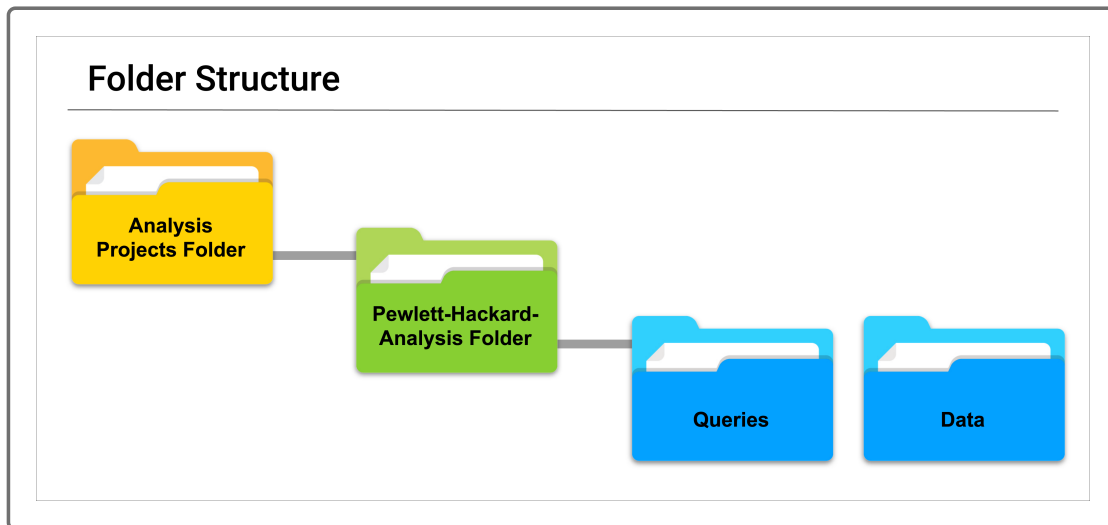
Datasets and Common Columns

Before we can even begin with the actual queries, and before we even load the data into our database, we need to understand what we're looking at.

Download our CSV files to take an initial look.

GITHUB

Create a new GitHub repository named "Pewlett-Hackard-Analysis." Then navigate to your class folder and clone your new repo within the folder. Set up your folder structure as shown below.



For this module, download the following CSV files and save them in the Data folder. Then commit and push these new files to your repo.

[departments.csv](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/departments.csv) (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/departments.csv)

[dept_emp.csv](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/dept_emp.csv) (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/dept_emp.csv)

[dept_manager.csv](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/dept_manager.csv) (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/dept_manager.csv)

[employees.csv](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/employees.csv) (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/employees.csv)

[salaries.csv](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/salaries.csv) (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/salaries.csv)

[titles.csv](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/titles.csv) (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_7/titles.csv)

Now we have six CSVs, each containing different data. Open and review the `departments.csv` file.

dept_no	dept_name
d001	Marketing
d002	Finance
d003	Human Resource
d004	Production
d005	Development
d006	Quality Manager
d007	Sales
d008	Research
d009	Customer Service

There isn't an overwhelming amount of data in this table—only two columns and 10 rows. It's also commonly known as a "lookup table" and is used to organize data. An example is if we would sort revenue, employee counts, and salaries by department.

Let's look at `dept_emp.csv` next.

emp_no	dept_no	from_date	to_date
10001	d005	6/26/86	1/1/99
10002	d007	8/3/96	1/1/99
10003	d004	12/3/95	1/1/99
10004	d004	12/1/86	1/1/99
10005	d003	9/12/89	1/1/99
10006	d005	8/5/90	1/1/99
10007	d008	2/10/89	1/1/99
10008	d005	3/11/98	7/31/00
10009	d006	2/18/85	1/1/99
10010	d004	11/24/96	6/26/00
10010	d006	6/26/00	1/1/99
10011	d009	1/22/90	11/9/96
10012	d005	12/18/92	1/1/99
10013	d003	10/20/85	1/1/99
10014	d005	12/29/93	1/1/99

There are only four columns of data, but considerably more rows in the spreadsheet. Did you notice the common column, dept_no, shared between `departments.csv` and `dept_emp.csv`?

departments.csv

dept_no	dept_name
d001	Marketing
d002	Finance
d003	Human Resource
d004	Production
d005	Development
d006	Quality Manager
d007	Sales
d008	Research
d009	Customer Service

dept_emp.csv

emp_no	dept_no	from_date
10001	d005	6/26/86
10002	d007	8/3/96
10003	d004	12/3/95
10004	d004	12/1/86
10005	d003	9/12/89
10006	d005	8/5/90
10007	d008	2/10/89
10008	d005	3/11/98
10009	d006	2/18/85

Department numbers are listed in both spreadsheets, providing a link between the two. For example, `dept_emp.csv` shows that Employee No. 10009 worked in Department No. 006, and `departments.csv` shows that Department No. 006 is the Quality Management department. We also

know Employee No. 10009 joined the Quality Management department on February 18, 1985.

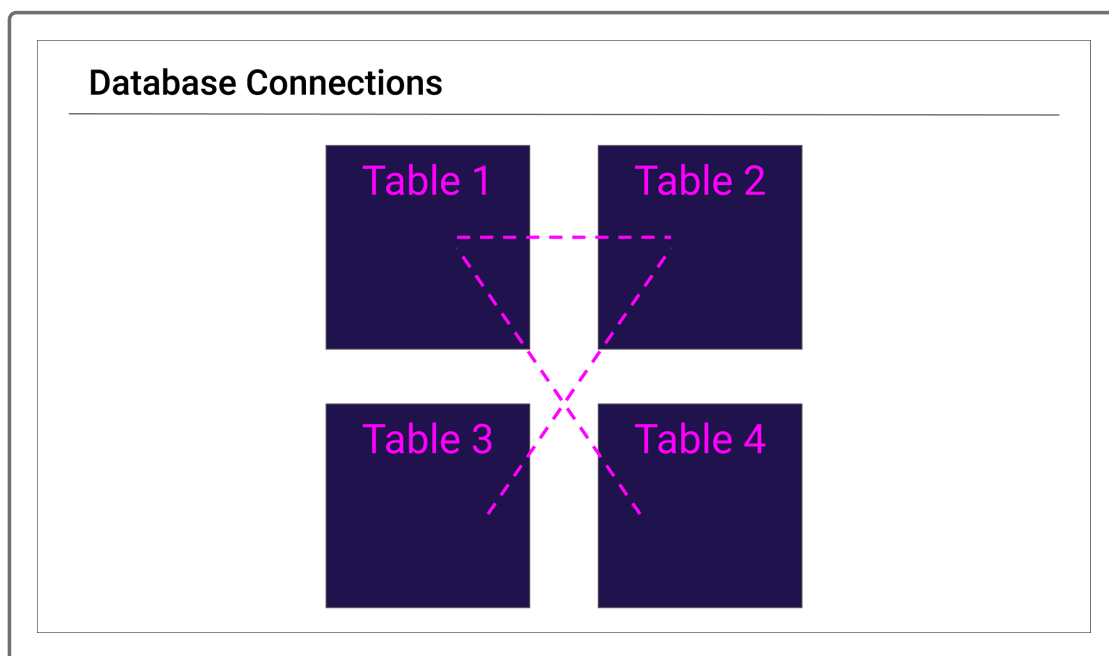
Database Keys

Database keys identify records from tables and establish relationships between tables. There are numerous types of keys. For our purposes, we will focus on primary keys and foreign keys.

Primary Keys

The `departments.csv` file has a dept_no column with unique identifiers for each row (one department number per department). For example, d001 will always reference the Marketing department, across other worksheets. This unique identifier is known as a **primary key**.

Primary keys are an important part of database design. When a database is being created, each table added must include a primary key in the architecture. Primary keys serve as a link between these tables.



In the graphic above, Table 1 has a primary key, or column of unique identifiers in common with Tables 2 and 4. Table 3's primary key is linked only to Table 2. These links trace the relationships between tables. There are times when we'll need to trace two or three links to get the exact data we need. In these cases, we'll pick the data we need from each table. Linking the tables together in this manner is called a **join**, a feature we'll get into later.


In the second CSV file, `dept_emp.csv`, the "emp_no" column contains the primary key.

dept_emp.csv

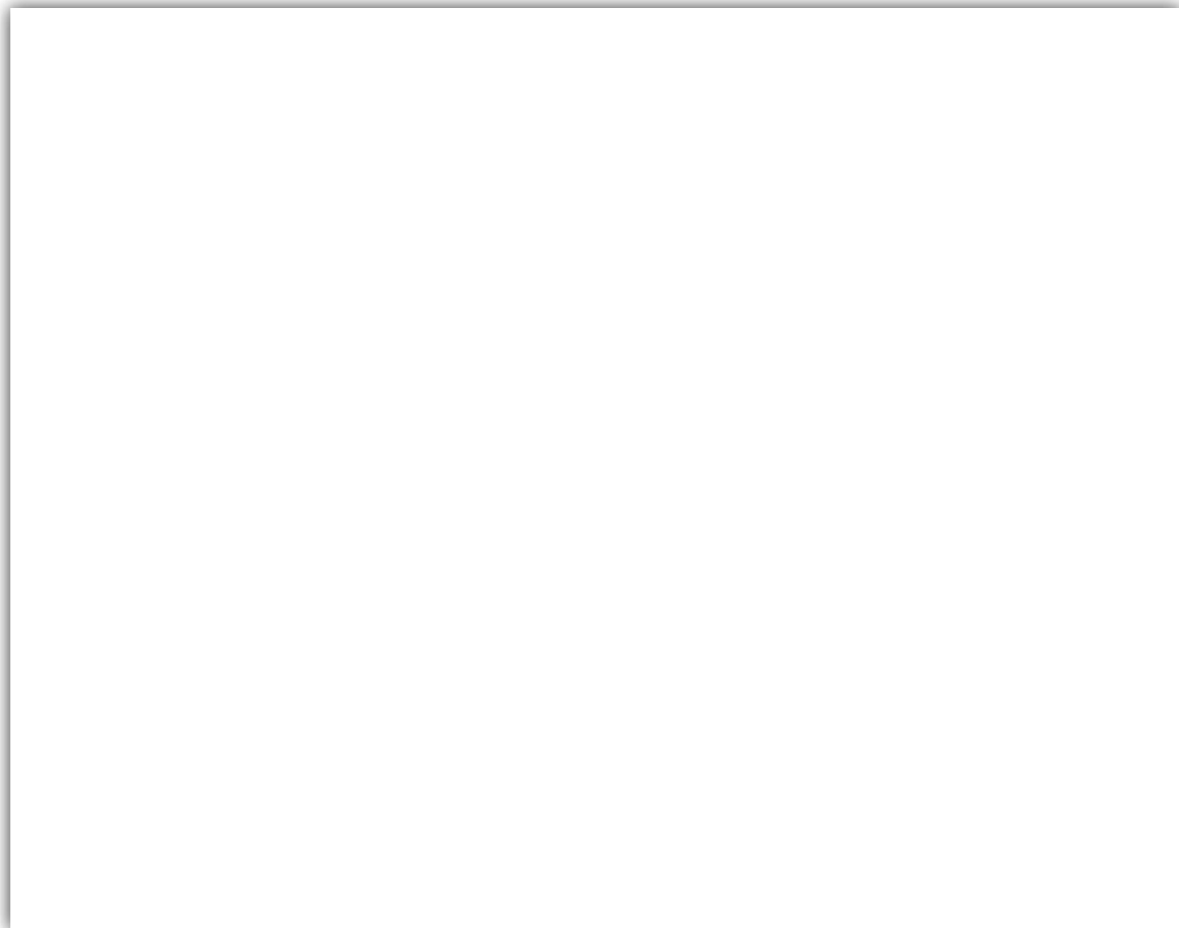
emp_no	dept_no	from_date	to_date
10001	d005	6/26/86	1/1/99
10002	d007	8/3/96	1/1/99
10003	d004	12/3/95	1/1/99
10004	d004	12/1/86	1/1/99
10005	d003	9/12/89	1/1/99
10006	d005	8/5/90	1/1/99
10007	d008	2/10/89	1/1/99

We know this is the primary key because each number is unique. For example, the emp_no column holds employee numbers. Each employee will have only one number, and that number won't be used for any other employee.

dept_emp.csv

Unique Numbers 	emp_no	dept_no	from_date	to_date
	10001	d005	6/26/86	1/1/99
	10002	d007	8/3/96	1/1/99
	10003	d004	12/3/95	1/1/99
	10004	d004	12/1/86	1/1/99
	10005	d003	9/12/89	1/1/99
	10006	d005	8/5/90	1/1/99
	10007	d008	2/10/89	1/1/99

Open that file and take an initial look at the data.



Nice work so far! Now test your skills with the following Skill Drill.

SKILL DRILL

Open the remaining CSVs and identify the primary key and data types in each file. We've already opened

`departments.csv` and `dept_emp.csv`, so there are four more to open:

- `dept_manager.csv`
- `employees.csv`
- `salaries.csv`
- `titles.csv`

Foreign Keys

Foreign keys are just as important as primary keys. While primary keys contain unique identifiers for their dataset, a **foreign key** references another dataset's primary key.

Think about it like a phone number. You have your own number. It's your number, assigned to your phone, and unique to you. This is your primary key. Your friend also has a primary key: his or her own phone number.

When you save your friend's number in your phone, you're creating a reference to that person, also known as a foreign key. Your phone has lots of foreign keys (such as parents, doctors offices, friends, and other family), but only one primary key.

Likewise, when your friend saves your number in their phone, your number is now a foreign key in their phone. Saving these keys connects the devices. They show the relationship between your phone and your friend's phone.

Compare our first two CSVs again by looking at the following image.

departments.csv		dept_emp.csv		
Primary Key		Primary Key	Foreign Key	
dept_no	dept_name	emp_no	dept_no	from_date
d001	Marketing	10001	d005	6/26/86
d002	Finance	10002	d007	8/3/96
d003	Human Resource	10003	d004	12/3/95

In this example, dept_no shows up in both datasets; as an identifier (or primary key) in one and as a reference (or foreign key) in the other. This demonstrates the link between employees and which department they work in.

We could continue to look for connections between the datasets, or we could create a roadmap of the content. Our roadmap would serve as a quick reference diagramming the different datasets and their interconnections. Additionally, it could be used as a reference guide later, when we begin to create queries to access all of the data.