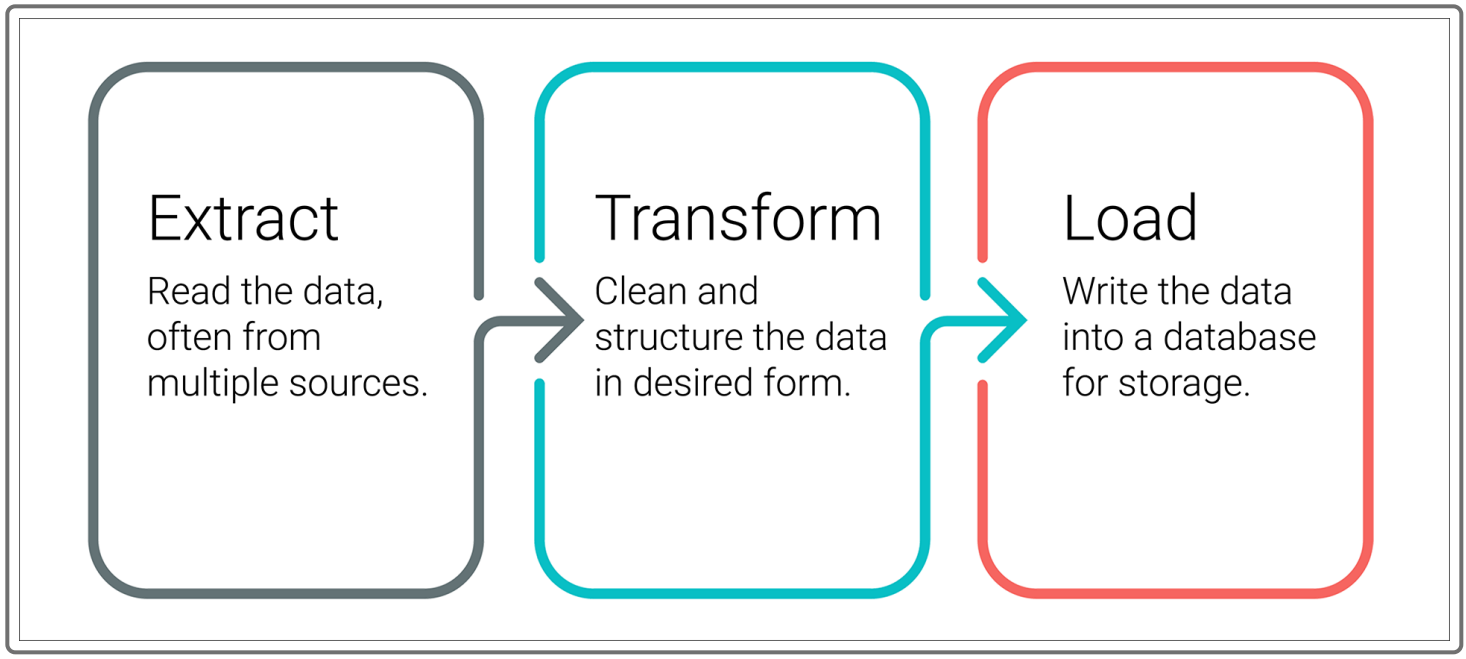
8.1.1

Extract, Transform, and Load the Data

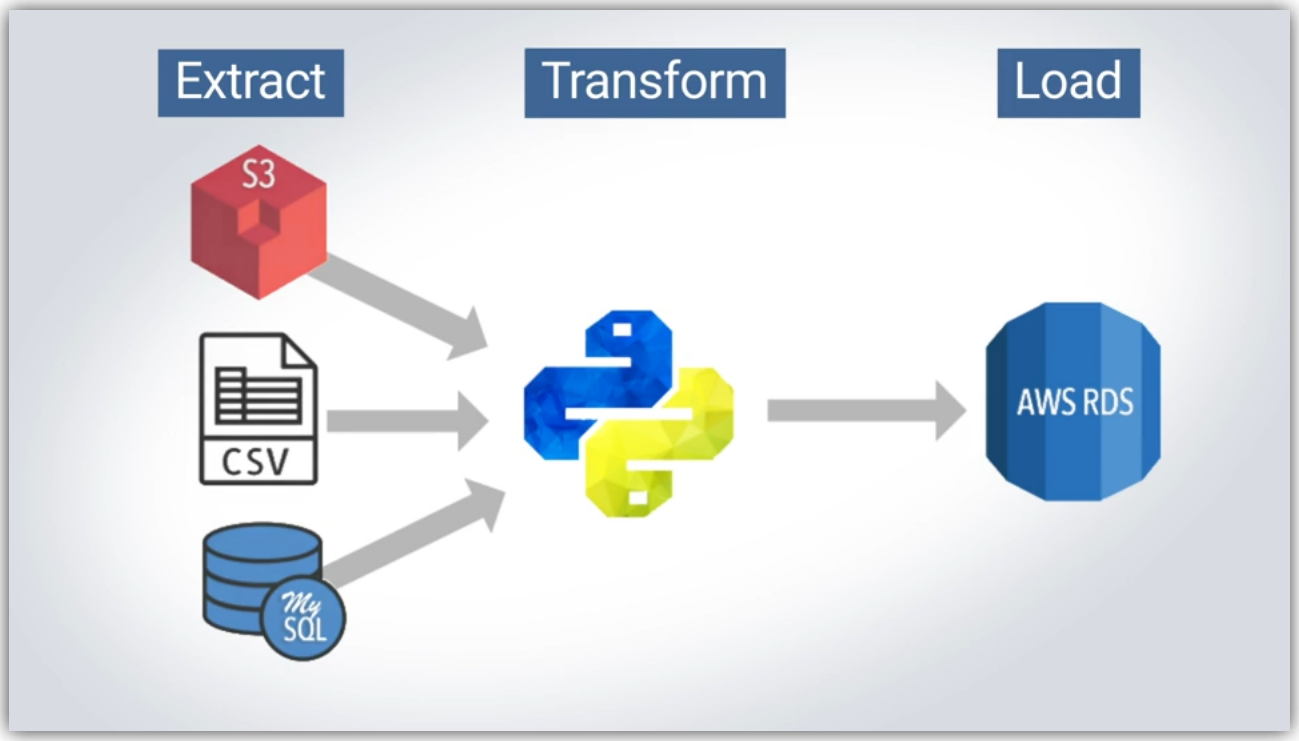
For the ETL process for Independent Funding, you and Britta need to create four new CSV files from a crowdfunding Excel file and upload those files into separate tables in a SQL database. To do so, you'll extract, transform, and load data. First you'll extract the data from two worksheets of an Excel file and place that data in four DataFrames. Then, you'll transform each DataFrame by cleaning, restructuring, formatting, filtering, and splitting the data. Finally, you'll load the cleaned datasets as CSV files into a SQL database.

The idea behind ETL is straightforward. Raw data exists in multiple places, and we need to clean and structure that data before we can analyze it. ETL breaks down this activity into three phases.



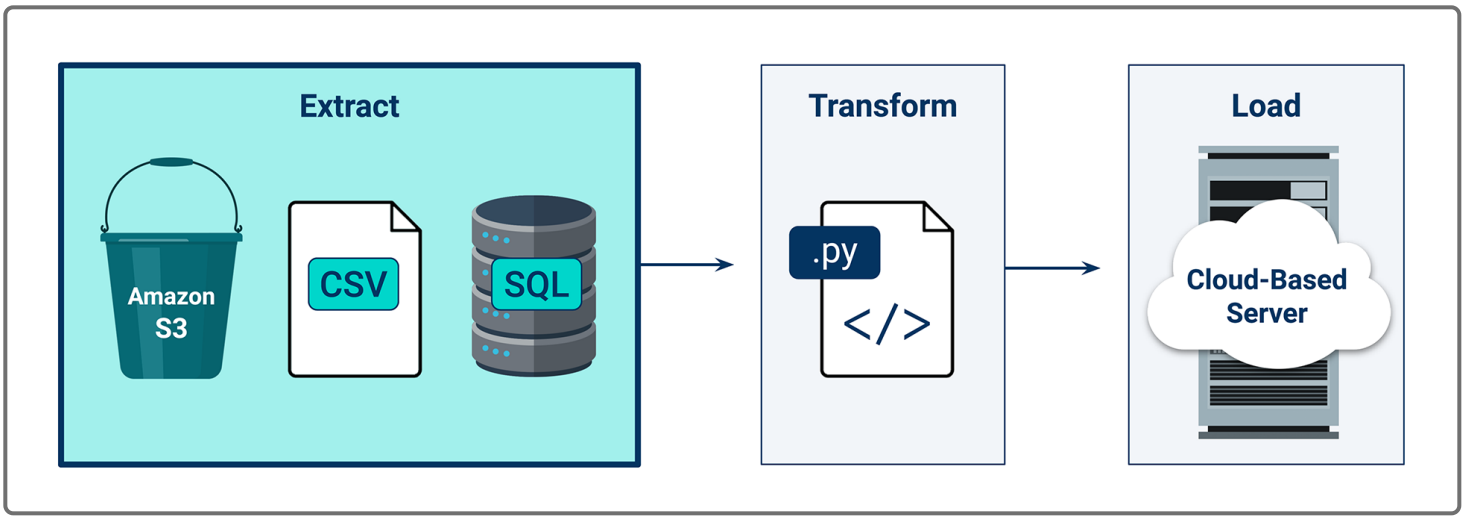
In the extract phase, we read the data, which often comes from multiple sources. In the transform phase, we clean and structure the data into the form that we want. In the load phase, we write the cleaned data to a database for storage.

To get acquainted with ETL, review the following video:



ETL is a flexible process for moving data. It can be as basic as a one-time migration from one database to another. Or, it can be as complex as an ongoing automated collection of messy, real-time data from many sources.

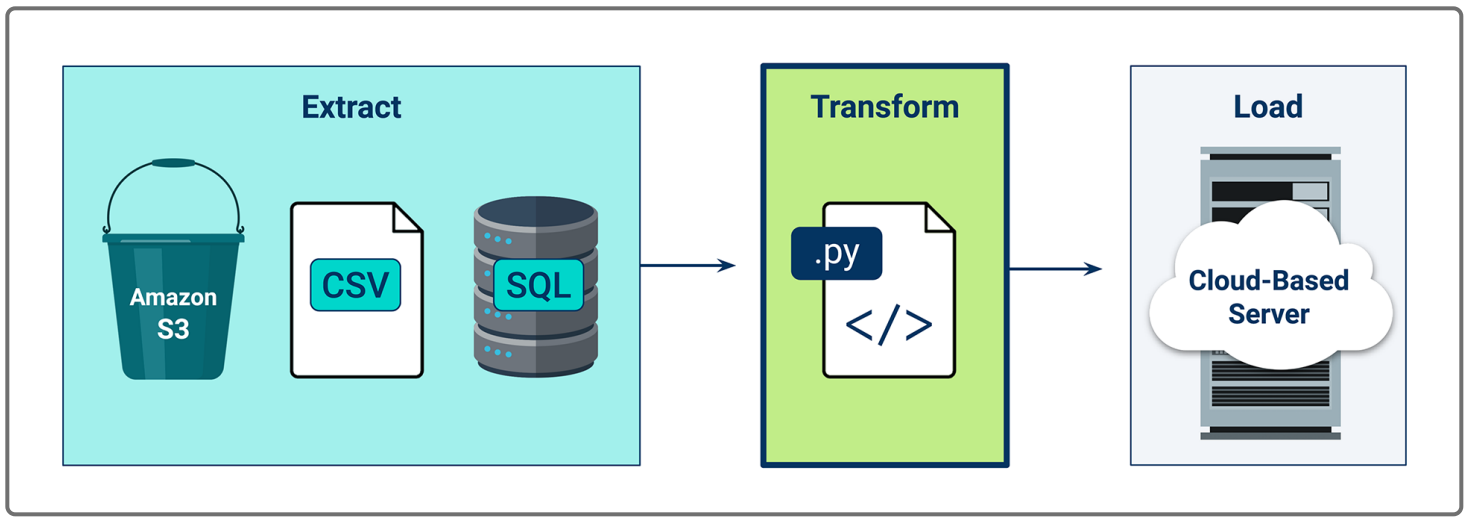
In the extract phase, data gets pulled from possibly diverse sources that are either internal or external. The sources might be flat files, scraped webpages in either HTML or JavaScript Object Notation (JSON) format, SQL tables, or even streams of sensor data stored on a hosted external cloud-based server. The extracted data gets held in a staging area that exists between the data sources and the data targets, like a PostgreSQL database, data warehouse, or data mart.



To help Britta, you'll extract the data that's stored in two worksheets of one large Excel file.

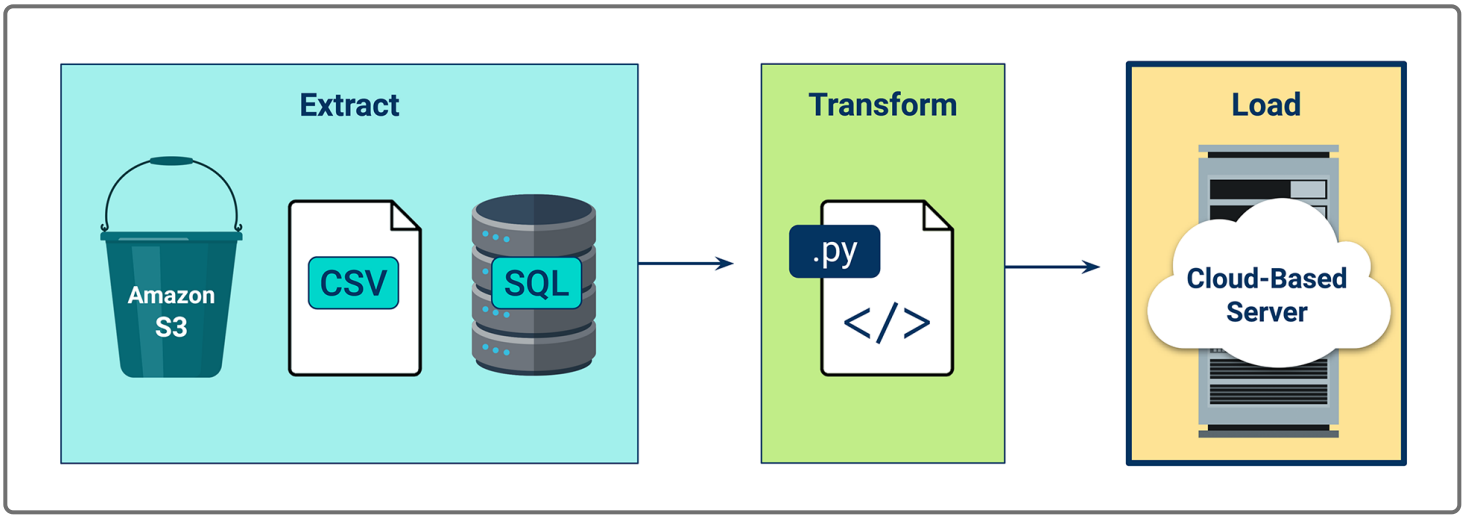
After we've extracted the data, we might need to transform it in many ways. For example, we might need to filter, format, parse, translate, split, sort, interpolate, pivot, summarize, aggregate, or merge the data to restructure it in a way that provides key relationships across tables. The goal is to create a consistent structure in the data. Otherwise, performing any meaningful analysis will be virtually impossible.

We can accomplish the transform phase by using either Python and Pandas, pure SQL, or a specialized ETL tool. Examples of these tools are Apache Airflow and Microsoft SQL Server Integration Services (MSSIS). Python and Pandas are especially good for prototyping an ETL transformation. With **prototyping**, we can test different transformation techniques and approaches to generate an example for users or stakeholders to validate. Python and Pandas in Jupyter Notebook are good sources for prototyping. because they provide flexibility and interactivity without enforcing any complicated frameworks, like Apache Airflow, Microsoft SSIS, or AWS Glue.



To help Britta, you'll use Python and Pandas to explore, document, and perform your data transformation.

Finally, after we've transformed the data into a consistent structure, we load it into the data target. The data target can be a relational database (like a PostgreSQL database), a nonrelational database that stores individual documents (like a MongoDB database), or a data warehouse that optimizes performance specifically for analytics (like Amazon Redshift).



Britta has determined that a SQL database is the best solution for sharing the data in the hackathon. So you'll load your data into a PostgreSQL table. Note that a SQL database is often the target of an ETL process. And because SQL is so ubiquitous, even databases that don't use SQL often have SQL-like interfaces.

Now that we've learned about the phases of the ETL process, let's get started with the first phase and extract some data!