Data Pipelines: ETL vs ELT

Data pipeline is a generic term for moving data from one place to another. For example, it could be moving data from one server to another server.

ETL

An [ETL pipeline](https://en.wikipedia.org/wiki/Extract,_transform,_load) is a specific kind of data pipeline and very common. ETL stands for Extract, Transform, Load. Imagine that you have a database containing web [log data](https://en.wikipedia.org/wiki/Log_file). Each entry contains the IP address of a user, a timestamp, and the link that the user clicked.

What if your company wanted to run an analysis of links clicked by city and by day? You would need another data set that maps IP address to a city, and you would also need to extract the day from the timestamp. With an ETL pipeline, you could run code once per day that would extract the previous day's log data, map IP address to city, aggregate link clicks by city, and then load these results into a new database. That way, a data analyst or scientist would have access to a table of log data by city and day. That is more convenient than always having to run the same complex data transformations on the raw web log data.

Before cloud computing, businesses stored their data on large, expensive, private servers. Running queries on large data sets, like raw web log data, could be expensive both economically and in terms of time. But data analysts might need to query a database multiple times even in the same day; hence, pre-aggregating the data with an ETL pipeline makes sense.

ELT

ELT (extract, load, transform) pipelines have gained traction since the advent of cloud computing. Cloud computing has lowered the cost of storing data and running queries on large, raw data sets. Many of these cloud services, like [Amazon Redshift](https://aws.amazon.com/redshift/), [Google BigQuery](https://cloud.google.com/bigquery/), or [IBM Db2](https://www.ibm.com/cloud/db2-warehouse-on-cloud) can be queried using SQL or a SQL-like language. With these tools, the data gets extracted, then loaded directly, and finally transformed at the end of the pipeline.

However, ETL pipelines are still used even with these cloud tools. Oftentimes, it still makes sense to run ETL pipelines and store data in a more readable or intuitive format. This can help data analysts and scientists work more efficiently as well as help an organization become more data driven.

Outline of the Lesson

1. Extract data from different sources such as:
   * csv files
   * json files
   * APIs
2. Transform data
   * combining data from different sources
   * data cleaning
   * data types
   * parsing dates
   * file encodings
   * missing data
   * duplicate data
   * dummy variables
   * remove outliers
   * scaling features
   * engineering features
3. Load
   * send the transformed data to a database
4. ETL Pipeline
   * code an ETL pipeline

This lesson contains many Jupyter notebook exercises where you can practice the different parts of an ETL pipeline. Some of the exercises are challenging, but they also contain hints to help you get through them. You'll notice that the "transformation" section is relatively long. You'll oftentimes hear data scientists say that cleaning and transforming data is how they spend a majority of their time. This lesson reflects that reality.

Big Data Courses at Udacity

"Big Data" gets a lot of buzz these days, and it is definitely an important part of a data engineer's and, sometimes, a data scientists's work. With "Big Data", you need special tools that can work on distributed computer systems.

This ETL course focuses on the practical fundamentals of ETL. Hence, you'll be working with a local data set so that you do not need to worry about learning a new tool. Udacity has other courses where the primary focus is on tools used for distributed data sets.

Here are links to other big data courses at Udacity:

* [Intro to Hadoop and MapReduce](https://www.udacity.com/course/intro-to-hadoop-and-mapreduce--ud617)
* [Deploying a Hadoop Cluster](https://www.udacity.com/course/deploying-a-hadoop-cluster--ud1000)
* [Real-time Analytics with Apache Storm](https://www.udacity.com/course/real-time-analytics-with-apache-storm--ud381)
* [Big Data Analytics in Health Care](https://www.udacity.com/course/big-data-analytics-in-healthcare--ud758)