ETL versus ELT

So far in this course, you have learned about ETL pipelines that extract, transform, and load data between database storage systems. You have also started learning about newer pipeline systems like ELT pipelines that extract, load, and then transform data. In this reading, you are going to learn more about the differences between these two systems and the ways different types of database storage fit into those systems. Understanding these differences will help you make key decisions that promote performance and optimization to ensure that your organization's systems are efficient and effective.

The primary difference between these two pipeline systems is the order in which they transform and load data. There are also some other key differences in how they are constructed and used:

Differences	ETL	ELT
The order of extraction, transformation, and loading data	Data is extracted, transformed in a staging area, and loaded into the target system	Data is extracted, loaded into the target system, and transformed as needed for analysis
Location of transformations	Data is moved to a staging area where it is transformed before delivery	Data is transformed in the destination system, so no staging area is required
Age of the technology	ETL has been used for over 20 years, and many tools have been developed to support ETL pipeline systems	ELT is a newer technology with fewer support tools built-in to existing technology
Access to data within the system	ETL systems only transform and load the data designated when the warehouse and pipeline are constructed	ELT systems load all of the data, allowing users to choose which data to analyze at any time
Calculations	Calculations executed in an ETL system replace or revise existing columns in order to push the results to the target table	Calculations are added directly to the existing dataset
Compatible storage systems	ETL systems are typically integrated with structured, relational data warehouses	ELT systems can ingest unstructured data from sources like data lakes
Security and compliance	Sensitive information can be redacted or anonymized before loading it into the data warehouse, which protects data	Data has to be uploaded before data can be anonymized, making it more vulnerable
Data size	ETL is great for dealing with smaller datasets that need to undergo complex transformations	ELT is well-suited to systems using large amounts of both structured and unstructured data
Wait times	ETL systems have longer load times, but analysis is faster because data has already been transformed when users access it	Data loading is very fast in ELT systems because data can be ingested without waiting for transformations to occur, but analysis is slower

Data storage systems

Because ETL and ELT systems deal with data in slightly different ways, they are optimized to work with different data storage systems. Specifically, you might encounter data warehouses and data lakes. As a refresher, a data warehouse is a type of database that consolidates data from multiple source systems for data consistency, accuracy, and efficient access. And a data lake is a database system that stores large amounts of raw data in its original format until it's needed. While these two systems perform the same basic function, there are some key differences:

Data warehouse	Data lake
Data has already been processed and stored in a relational system	Data is raw and unprocessed until it is needed for analysis; additionally, it can have a copy of the entire OLTP or relational database
The data's purpose has already been assigned, and the data is currently in use	The data's purpose has not been determined yet
Making changes to the system can be complicated and require a lot of work	Systems are highly accessible and easy to update

There is also a specific type of data warehouse you might use as a data source: data marts. Data marts are very similar to data warehouses in how they are designed, except that they are much smaller. Usually, a data mart is a single subset of a data warehouse that covers data about a single subject.

Key takeaways