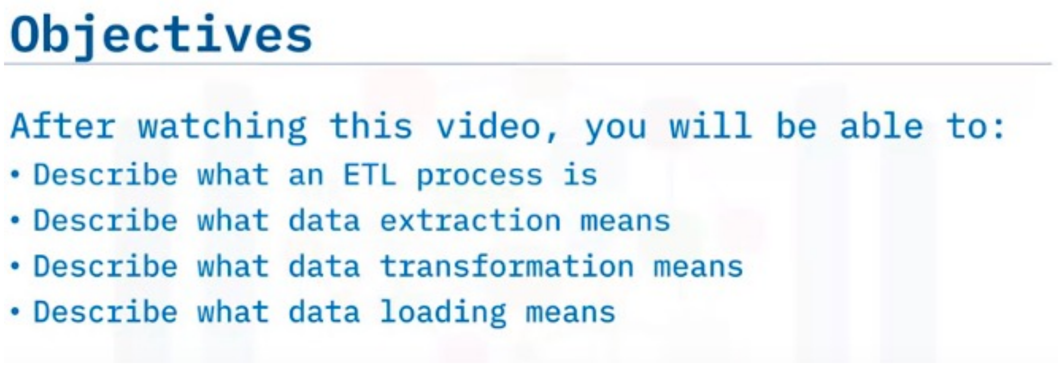
ETL & ELT Techniques

- Describe what an ETL process is
- Explain what data loading means
- Describe why ELT is an emergent trend
- Describe the trending shift from ETL to ELT
- Summarize data extraction techniques
- Name data transformation techniques
- List ways information can be “lost in transformation”
- Summarize data loading techniques
- Differentiate batch loading from stream loading
- Contrast ETL and ELT.

ETL Fundamentals



Objectives

After watching this video, you will be able to:

- Describe what an ETL process is
- Describe what data extraction means
- Describe what data transformation means
- Describe what data loading means

ETL stands for **Extract, Transform, and Load.**

- ETL is an automated data pipeline engineering methodology, whereby data is acquired and prepared for subsequent use in an analytics environment, such as a data warehouse or data mart.
- ETL refers to the process of curating data from multiple sources, conforming it to a unified data format or structure, and then loading the transformed data into its new environment.

- The Extraction process obtains or reads the data from one or more sources.
- The Transformation process wrangles the data into a format that is suitable for its destination and its intended use.
- The final Loading process takes the transformed data and loads it into its new environment, ready for visualization, exploration, further transformation, and modelling.
- The curated data may also be utilized to support automation and decision-making.

What is Extraction?

- To extract data is to configure access to it and read it into an application. Normally this is an automated process.

Some common methods include:

- Web scraping, where data is extracted from web pages using applications such as Python or R to parse the underlying HTML code, and Using APIs to programmatically connect to data and query it.
- The source data may be relatively static, such as a data archive, in which case the extraction step would be a stage within a batch process.
- On the other hand, the data could be streaming live, and from many locations.
- Examples include weather station data, social networking feeds, and IoT devices.

What is data transformation?

- Data transformation, also known as data wrangling, means processing data to make it conform to the requirements of both the target system and the intended use case for the curated data.
- Transformation can include any of the following kinds of processes:

Cleaning: fixing errors or missing values.

Filtering: selecting only what is needed.

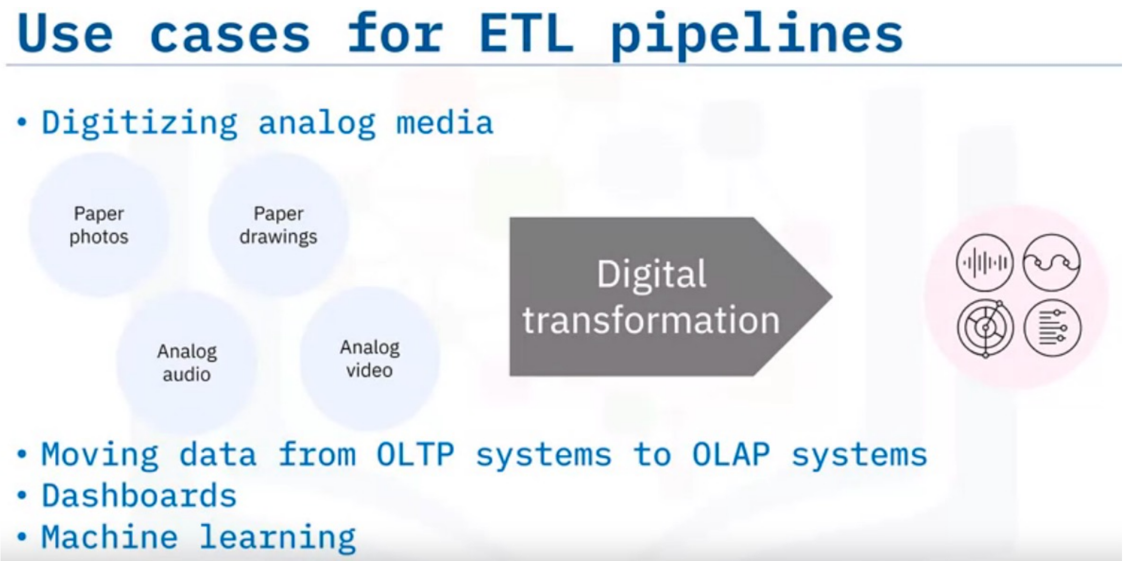
Joining disparate data sources: merging related data.

Feature engineering: such as creating KPIs for dashboards or machine learning.

Formatting and data typing: making the data compatible with its destination.

What is data loading?

- Generally this just means writing data to some new destination environment.
- Typical destinations include **databases, data warehouses, and data marts.**
- The key goal of data loading is to make the data readily available for ingestion by analytics applications so that end users can gain value from it.
- Applications include dashboards, reports, and advanced analytics such as forecasting and classification.



Use cases for ETL pipelines

- Digitizing analog media

Paper photos

Paper drawings

Analog audio

Analog video

Digital transformation

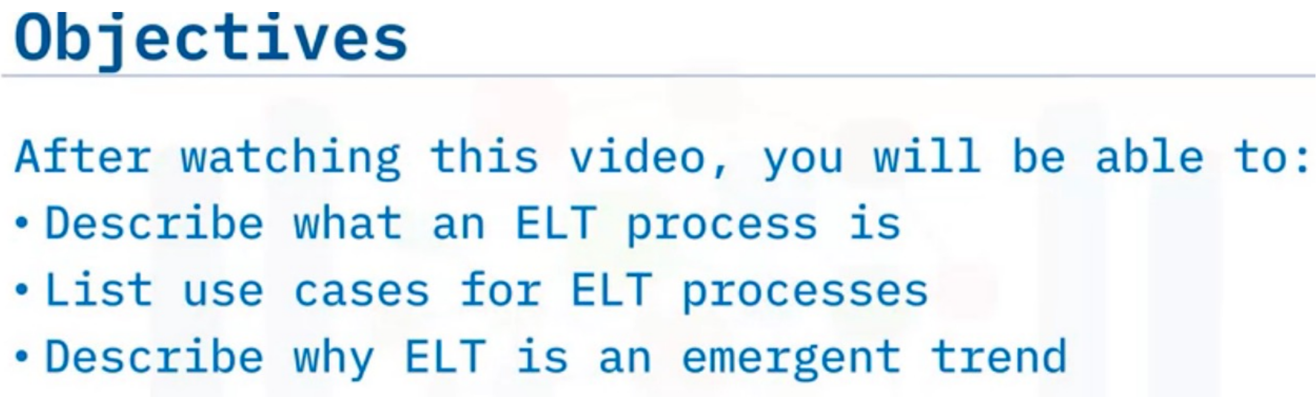
- Moving data from OLTP systems to OLAP systems
- Dashboards
- Machine learning

There are many use cases for ETL pipelines.

- A very large amount of information is either already recorded or being generated, but is not yet captured, or accessible, as a digital file.
- Examples include paper documents, photos and illustrations, and analog audio and video tapes.
- Digitizing analog data includes extraction by some form of scanning, analog-to-digital transformation, and, finally, storage into a repository.
- Online transaction processing (OLTP) systems don’t save historical data.
- Accordingly, ETL processes capture the transaction history and prepare it for subsequent analysis in an online analytical processing (OLAP) system.
- Other use cases include engineering ‘features’, or KPIs, from data sources, as preparation for:

1. Ingestion by dashboards used by operations, sales and marketing, customers, and executives.
2. Training and deploying machine learning models for prediction and augmented decision making.

ELT Basics



Objectives

After watching this video, you will be able to:

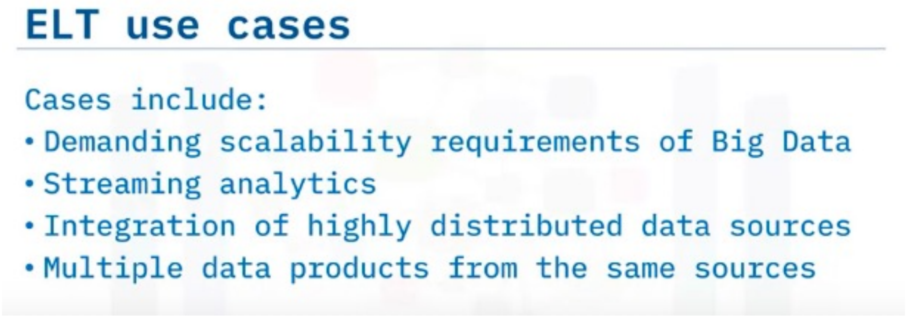
- Describe what an ELT process is
- List use cases for ELT processes
- Describe why ELT is an emergent trend

ELT stands for**: Extract, Load, and Transform.**

- ELT is an acronym for a specific automated data pipeline engineering methodology.
- ELT is similar to ETL in that similar stages are involved but the order in which they are performed is different.

ELT

- For ELT processes, data is acquired and directly loaded, as-is, into its destination environment.
- From its new home, usually a sophisticated analytics platform such as a data lake, it can be transformed on demand and however users wish.
- Like ETL, the first stage in the ELT process is **Extraction**. The Extraction process obtains the data from all sources and reads the data, often in an asynchronous fashion, into an application.
- The **Loading** process takes the raw data as-is , and loads it into its new environment, where modern analytics tools can then be used directly.
- The **Transformation** process for ELT is much more dynamic than it is for conventional ETL.
- Modern analytics tools in the destination environment enable interactive, on-demand exploration and visualization of your data, including advanced analytics such as modelling and prediction.



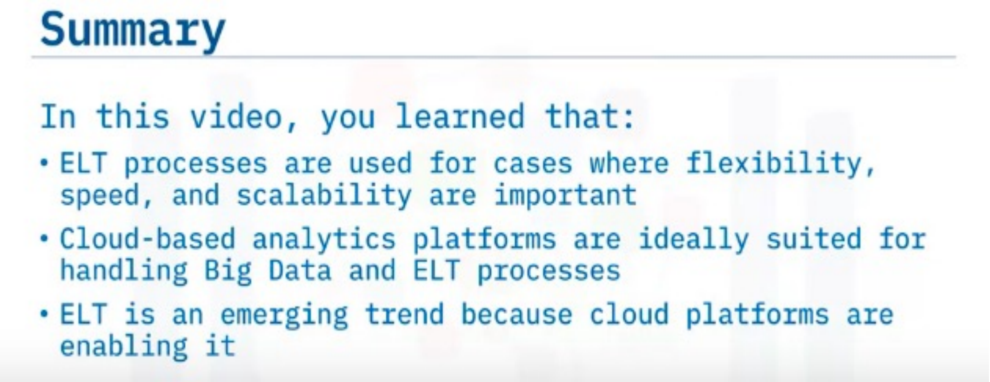
ELT use cases

Cases include:

- Demanding scalability requirements of Big Data
- Streaming analytics
- Integration of highly distributed data sources
- Multiple data products from the same sources

Why is ELT emerging?

- cloud computing solutions are evolving at tremendous rates due to the demands of **Big Data.**
- They can easily **handle huge amounts of asynchronous data** which can be highly distributed around the world.
- **Cloud computing** resources are practically unlimited, and they **can scale on demand**.
- Unlike traditional on-premises hardware, you only pay for the computing resources you use.
- You don't have to worry about underutilizing resources, that is, overspending on equipment.
- With ELT, you have a clean **separation between moving data and processing data**.
- cloud computing is equally prepared to handle the most challenging cases for either of these two tasks.
- There may be many reasons to transform your data and just as many ways to do it.
- Thus, **ELT is a flexible option** that enables a variety of applications from the same source of data.
- Because you are **working with a replica of the source data**, there **is no information loss.**
- Many kinds of transformations can lead to information loss, and if these happen somewhere upstream in the pipeline, it may be a long time before you can have a change request met.
- Worse yet, the information may be forever lost if the raw data is not stored.

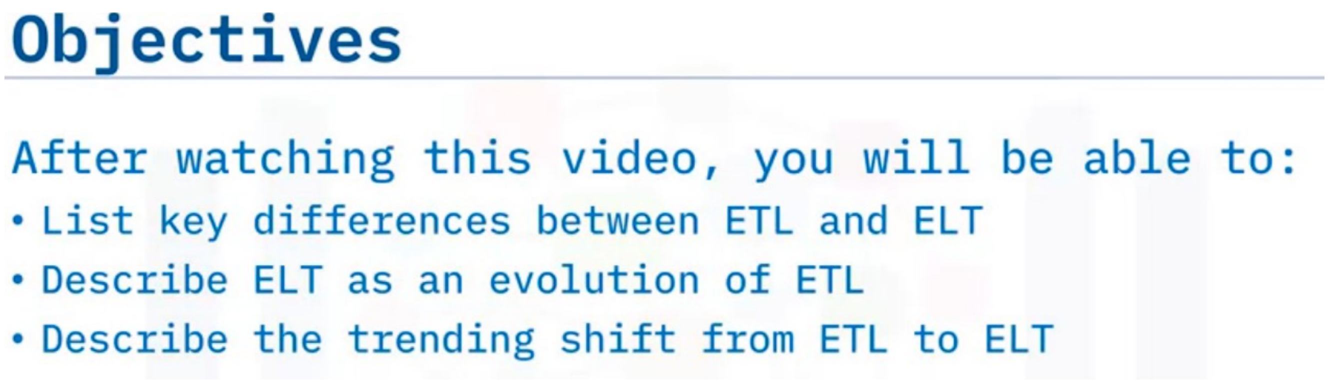


Summary

In this video, you learned that:

- ELT processes are used for cases where flexibility, speed, and scalability are important
- Cloud-based analytics platforms are ideally suited for handling Big Data and ELT processes
- ELT is an emerging trend because cloud platforms are enabling it

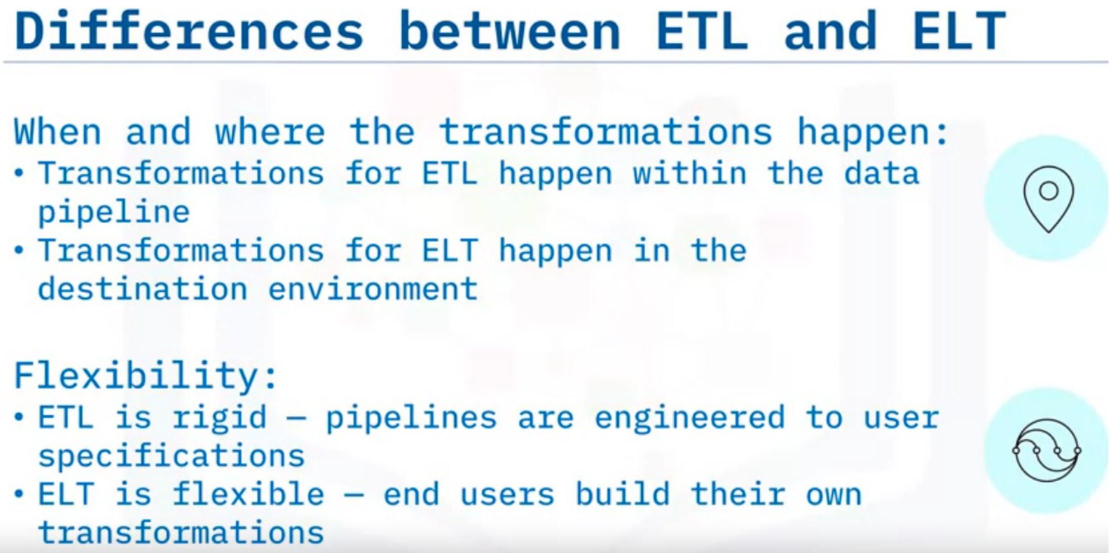
ETL vs ELT comparison



Objectives

After watching this video, you will be able to:

- List key differences between ETL and ELT
- Describe ELT as an evolution of ETL
- Describe the trending shift from ETL to ELT



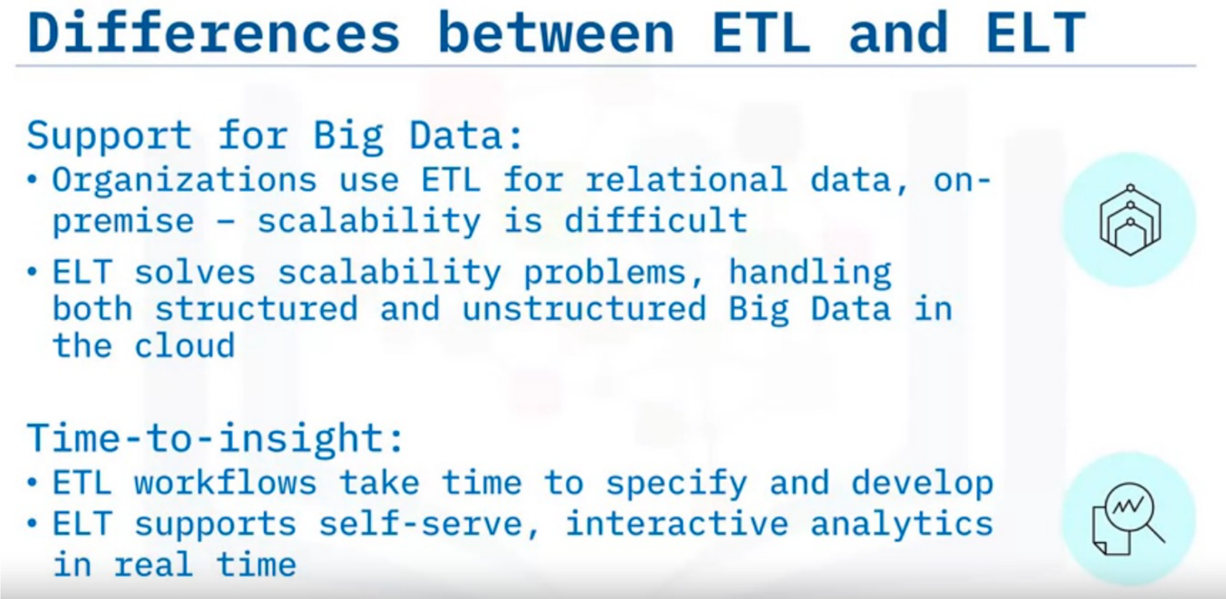
Differences between ETL and ELT

When and where the transformations happen:

- Transformations for ETL happen within the data pipeline
- Transformations for ELT happen in the destination environment

Flexibility:

- ETL is rigid – pipelines are engineered to user specifications
- ELT is flexible – end users build their own transformations



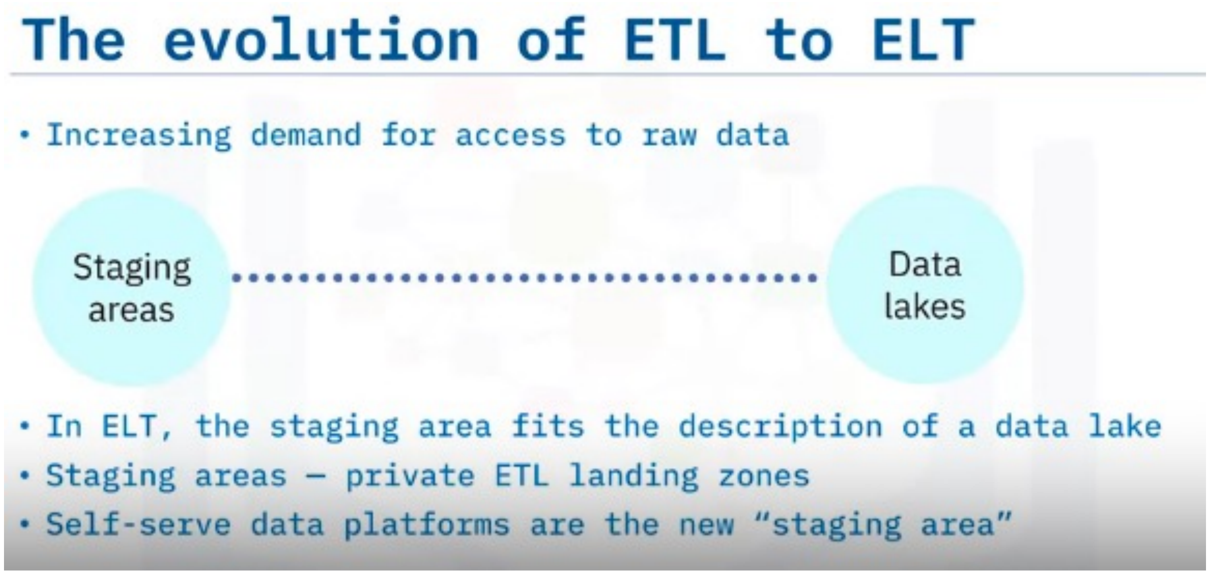
Differences between ETL and ELT

Support for Big Data:

- Organizations use ETL for relational data, on-premise – scalability is difficult
- ELT solves scalability problems, handling both structured and unstructured Big Data in the cloud

Time-to-insight:

- ETL workflows take time to specify and develop
- ELT supports self-serve, interactive analytics in real time



The evolution of ETL to ELT

- Increasing demand for access to raw data

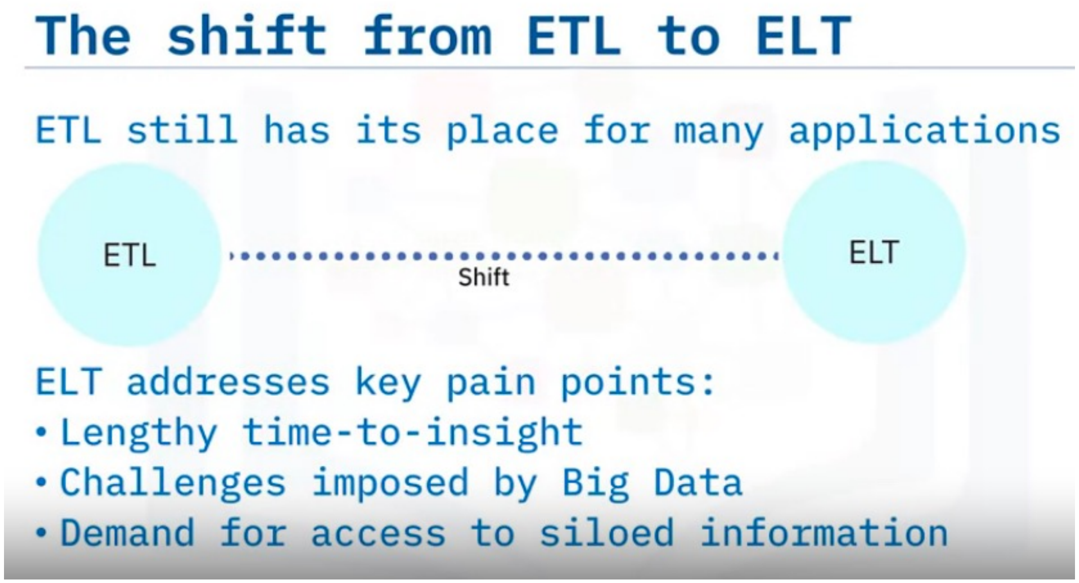
Staging areas

Data lakes

- In ELT, the staging area fits the description of a data lake
- Staging areas – private ETL landing zones
- Self-serve data platforms are the new "staging area"

ELT is a natural evolution of ETL.

- One of the factors driving that evolution is the demand to **release raw data to a wider user base for the enterprise**.
- Traditionally, **ETL p**rocesses include an **intermediate storage facility** called a **staging area.**
- This is a holding area for **raw extracted data**, where you can run processes prior to loading the resulting transformed data into a data warehouse or a data mart.
- This sounds a lot like an ELT process, and the staging area fits the description of a **data lake**, which is a **modern self-serve repository for storing and manipulating raw data.**
- A traditional staging area, however, is not something that is usually shared across the company.
- It's a private, siloed area set aside for developing, monitoring, and performance tuning the data pipeline and its **built-in transformations.**
- Along with the ever-increasing ease-of-use and connection capabilities of analytics tools, raw data sources have become much more accessible to less technical end users. Accordingly, the paradigm is shifting to self-service data platforms.



The shift from ETL to ELT

ETL still has its place for many applications

ETL

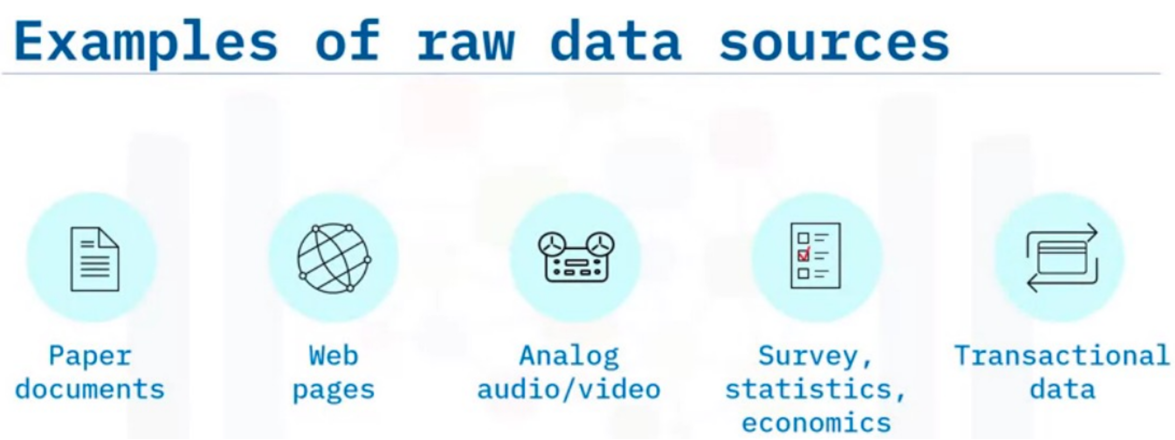
Shift

ELT

ELT addresses key pain points:

- Lengthy time-to-insight
- Challenges imposed by Big Data
- Demand for access to siloed information

Data Extraction Techniques



Examples of raw data sources

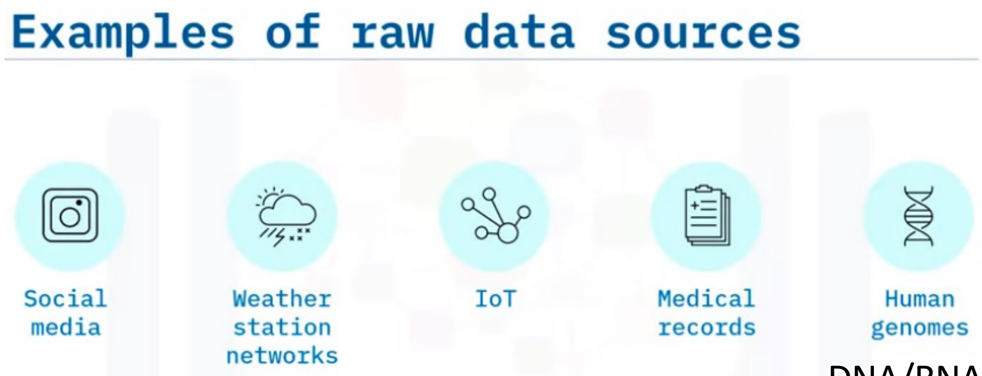
Paper documents

Web pages

Analog audio/video

Survey, statistics, economics

Transactional data



Examples of raw data sources

Social media

Weather station networks

IoT

Medical records

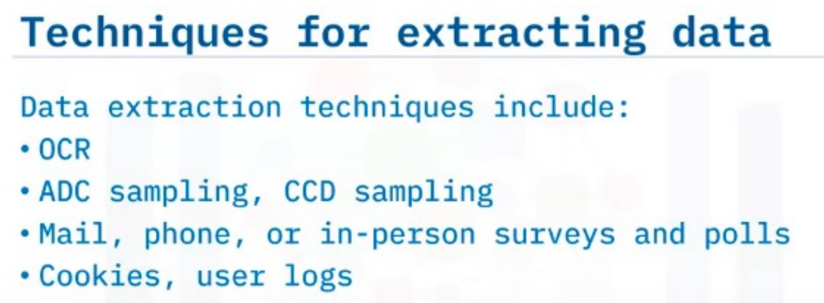
Human genomes

DNA/RNA

There are many techniques for extracting data,

depending on the kind of data source and the intended use of the data. Examples include:

1. Optical character recognition (OCR), which is used to interpret and digitize text scanned from pa documents so it can be stored as a computer-readable file
2. Analog-to-digital converters (ADCs), which can digitize analog audio recordings and signals, and charge-coupled devices (CCDs) that capture and digitize images Opinions, questionnaires, and
3. vital statistical data obtained through polling and census methods
4. Cookies, user logs, and other methods used for tracking human or system behavior

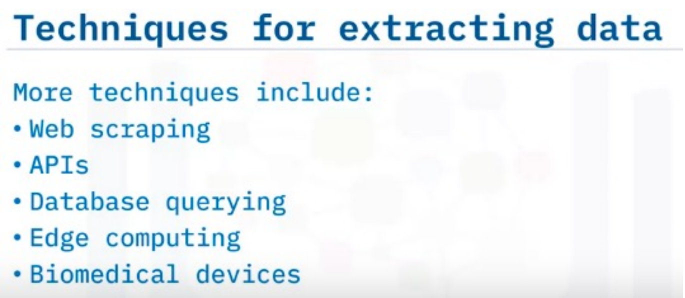


Techniques for extracting data

Data extraction techniques include:
- OCR
- ADC sampling, CCD sampling
- Mail, phone, or in-person surveys and polls
- Cookies, user logs

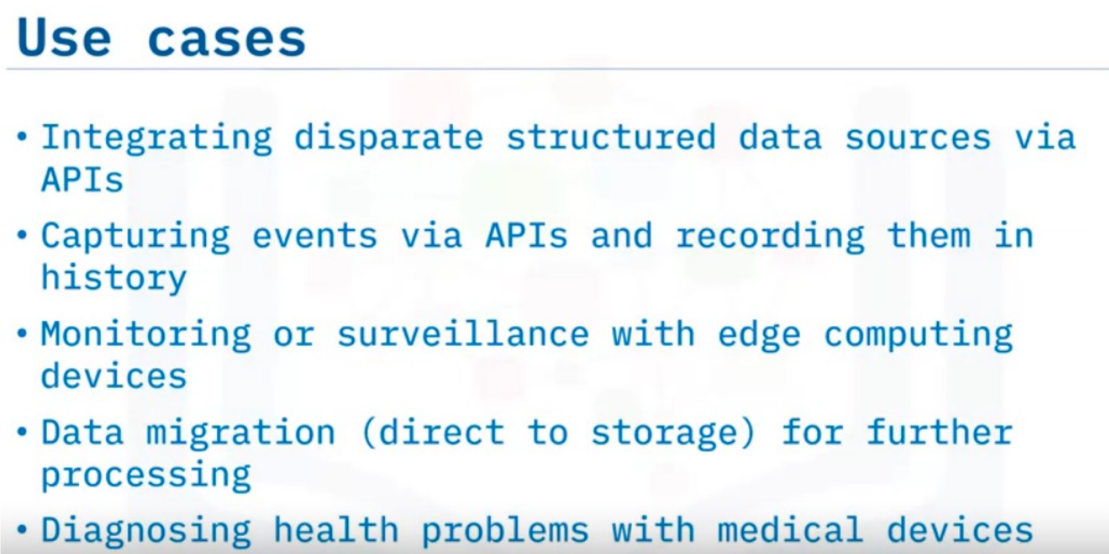
More techniques include:

1. Web scraping, used to crawl web pages in search of text, images, tables, and hyperlinks.
2. **APIs**, which are readily available for **extracting data from all sorts of online data repositor**ies and feeds, such as government bureaus of statistics, libraries, weather networks, online shopping, and social networks.
3. SQL languages for querying relational databases, and NoSQL for querying document, key-value, graph or other non-structured data repositories.
4. **Edge computing devices**, such as video cameras that have built-in processing that **can extract features from raw data**
5. devices, such as **microfluidic arrays** that can **extract DNA sequences**



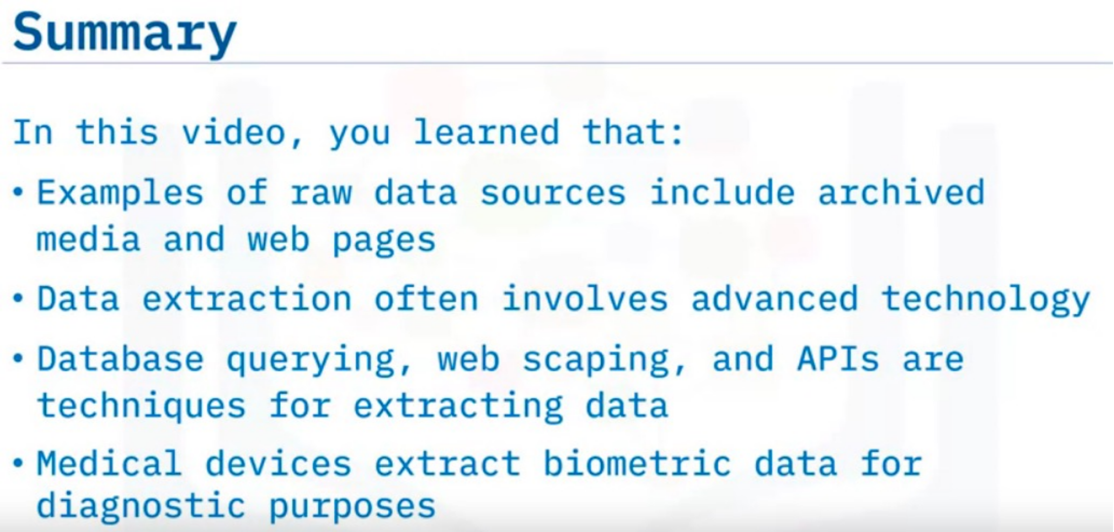
Techniques for extracting data

More techniques include:
- Web scraping
- APIs
- Database querying
- Edge computing
- Biomedical devices



Use cases

- Integrating disparate structured data sources via APIs
- Capturing events via APIs and recording them in history
- Monitoring or surveillance with edge computing devices
- Data migration (direct to storage) for further processing
- Diagnosing health problems with medical devices

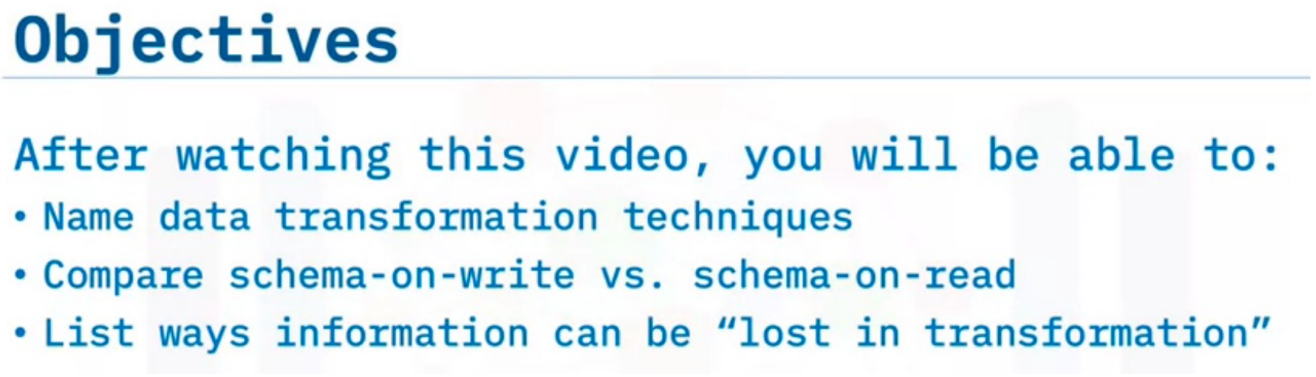


Summary

In this video, you learned that:

- Examples of raw data sources include archived media and web pages
- Data extraction often involves advanced technology
- Database querying, web scaping, and APIs are techniques for extracting data
- Medical devices extract biometric data for diagnostic purposes

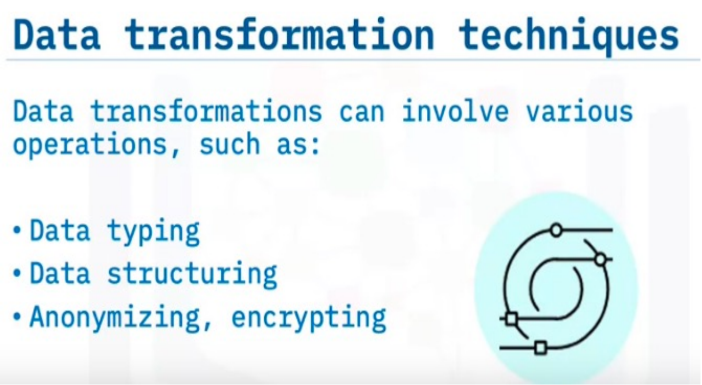
Introduction to Data Transformation Techniques



Objectives

After watching this video, you will be able to:

- Name data transformation techniques
- Compare schema-on-write vs. schema-on-read
- List ways information can be "lost in transformation"



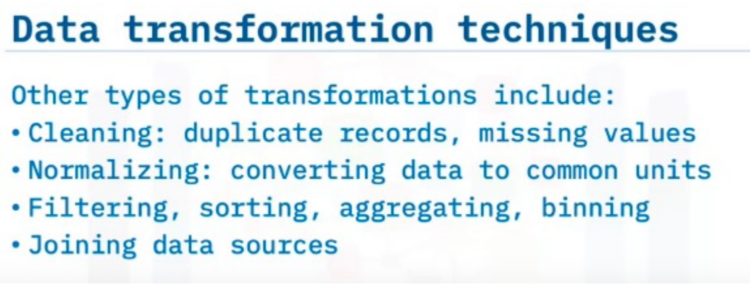
Data transformation techniques

Data transformations can involve various operations, such as:

- Data typing
- Data structuring
- Anonymizing, encrypting

- Data transformation is mainly about formatting the data to suit the application.
- This can involve many kinds of operations, such as:

1. **Data typing,** which involves casting data to appropriate types, such as integer, float, string, object, and category.
2. **Data structuring,** which includes converting one data format to another, such as JSON, XML, or CSV to database tables.
3. Anonymizing and encrypting transformations to help ensure privacy and security



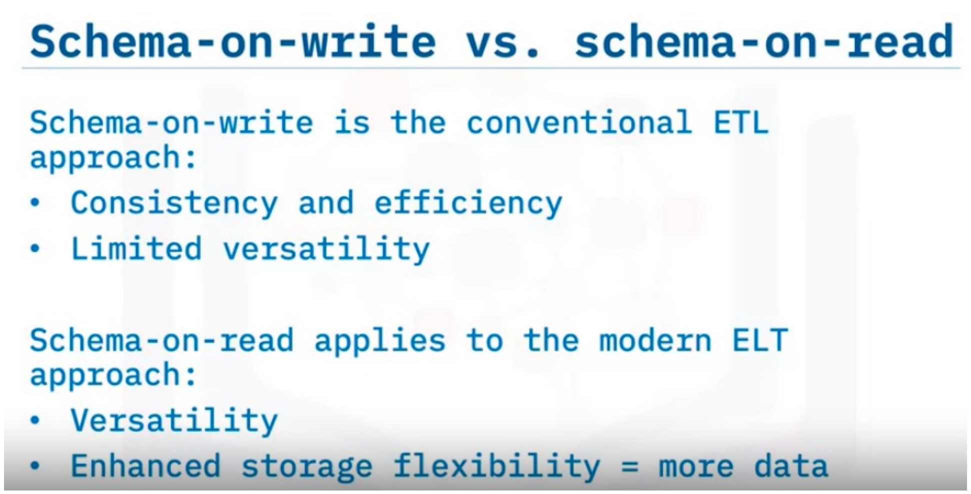
Data transformation techniques

Other types of transformations include:

- Cleaning: duplicate records, missing values
- Normalizing: converting data to common units
- Filtering, sorting, aggregating, binning
- Joining data sources

Other types of transformations include:

- Cleaning operations for removing duplicate records and filling missing values.
- Normalizing data to ensure units are comparable, for example, using a common currency.
- Filtering, sorting, aggregating, and binning operations for accessing the right data at a suitable level of detail and in a sensible order.
- Joining, or merging, **disparate data sources.**



Schema-on-write vs. schema-on-read

Schema-on-write is the conventional ETL approach:

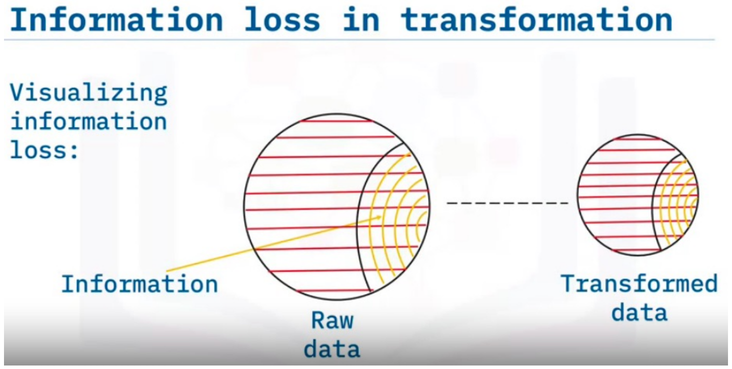
- Consistency and efficiency
- Limited versatility

Schema-on-read applies to the modern ELT approach:

- Versatility
- Enhanced storage flexibility = more data

- **Schema-on-write** is the conventional approach used in ETL pipelines, where the data must be conformed to a defined schema prior to loading to a destination, such as a relational database.
- The idea is to have the **data consistently** structured for stability and for making subsequent queries much faster, but this comes at the **cost of limiting the versatility of the data.**
- **Schema-on-read** relates to the modern ELT approach, where the schema is applied to the raw data after reading it from the raw data storage.
- This approach is versatile since it can obtain multiple views of the same source data using ad-hoc schemas.
- Users potentially have access to more data since it doesn't need to go through a rigorous pre-processing step.

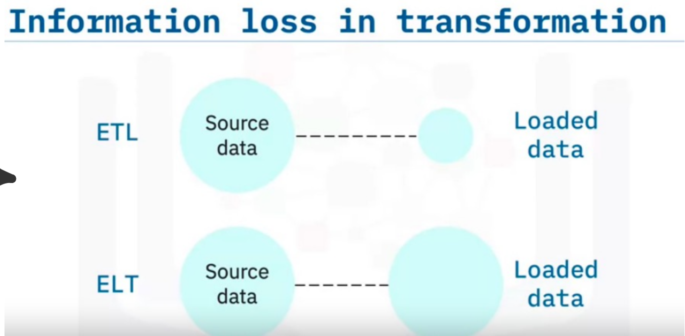
- Whether intentional or accidental, there are many ways in which information can be 'lost in transformation'.
- We can visualize this loss as follows.
- Raw data is normally much bigger than transformed data. Since data usually contains noise and redundancy, we can illustrate the 'information content' of data as a proper subset of the data.
- Correspondingly, we can see that shrinking the 'data volume' can also mean shrinking the 'information content'.



Information loss in transformation

Visualizing information loss:

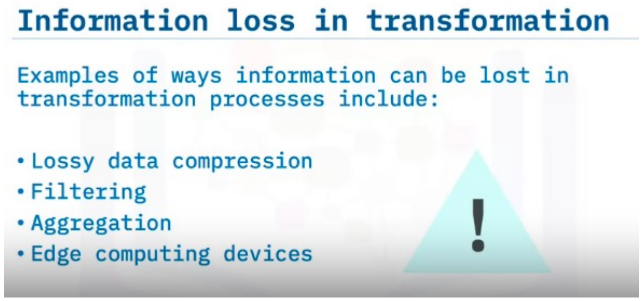
- Either way, for ETL processes, any lost information may or may not be recoverable, whereas **with ELT, all the original information content is left intact because the data is simply copied over as-is.**



Information loss in transformation

Examples of ways information can be lost in transformation processes include:

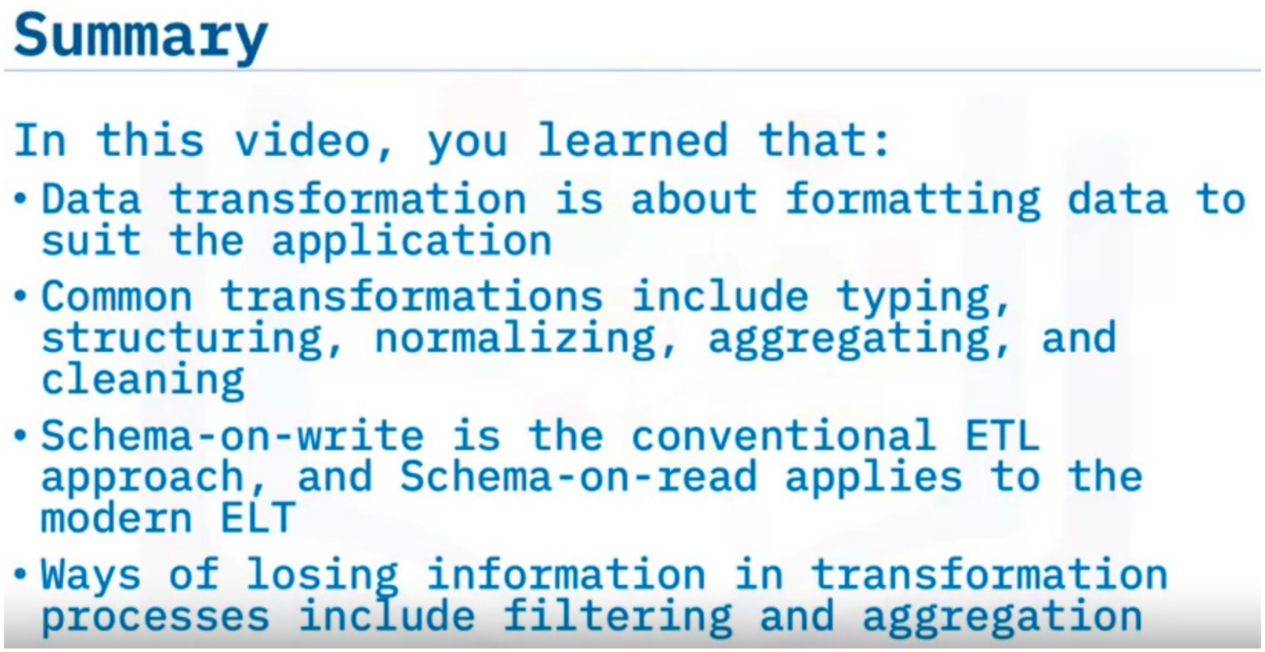
- **Lossy data compression**. For example, converting floating point values to integers, reducing bitrates on audio or video.
- **Filtering**. For example, filtering is usually a **temporary selection of a subset of data**, but when it is permanent, information can easily be discarded.
- Aggregation. For example, average yearly sales vs. daily or monthly average sales.
- **Edge computing** devices. For example, **false negatives in surveillance devices** designed to only stream alarm signals, not the raw data.



Information loss in transformation

Examples of ways information can be lost in transformation processes include:

- Lossy data compression
- Filtering
- Aggregation
- Edge computing devices

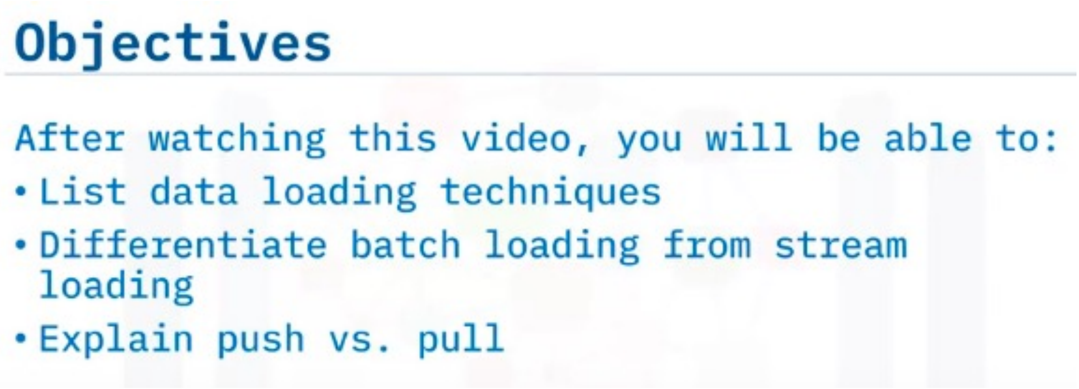


Summary

In this video, you learned that:

- Data transformation is about formatting data to suit the application
- Common transformations include typing, structuring, normalizing, aggregating, and cleaning
- Schema-on-write is the conventional ETL approach, and Schema-on-read applies to the modern ELT
- Ways of losing information in transformation processes include filtering and aggregation

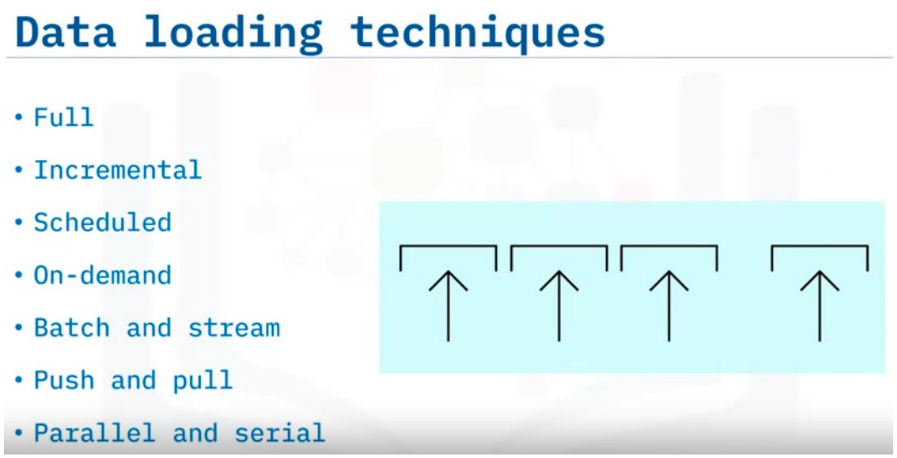
Data Loading Techniques



Objectives

After watching this video, you will be able to:

- List data loading techniques
- Differentiate batch loading from stream loading
- Explain push vs. pull



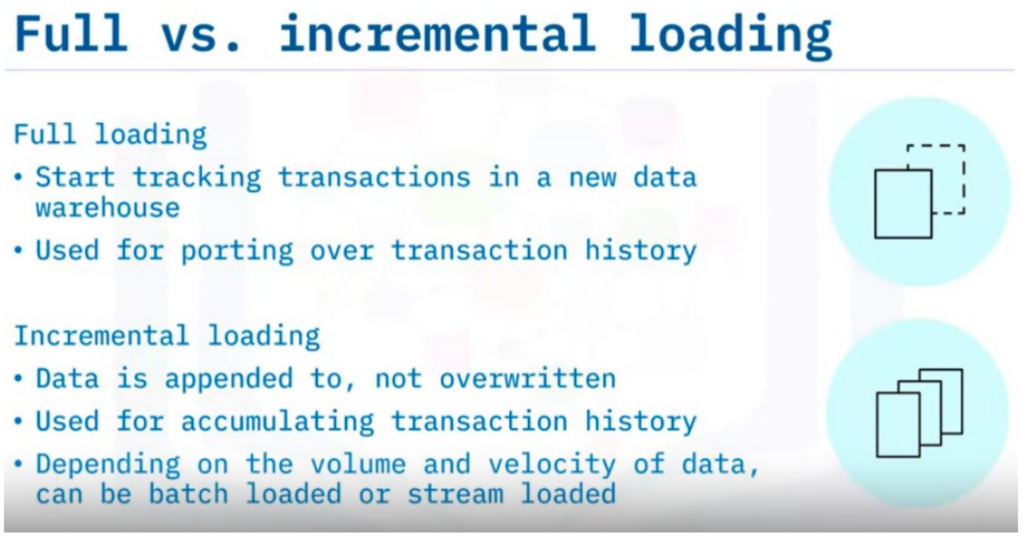
Data loading techniques

- Full
- Incremental
- Scheduled
- On-demand
- Batch and stream
- Push and pull
- Parallel and serial

There are many techniques for loading data, some of which are:

- **“Full loading”:** You can load an initial history into a database, after which
- “**incremental loading”** is applied to insert new data or to update already loaded data.
- You can **schedule data** loading to occur on a periodic basis,
- or you can load it **as required, on demand**.
- Data can be loaded **in batches, or it can be streaming** continuously to its destination.
- The data can be either **pushed to a server or pushed to clients by a server**.
- Data is usually **loaded serially**, but it can also **be loaded in parallel.**

- Full loading refers to loading data in one large batch.
- This is used, for example, when organizations want to start tracking transactions in a new data warehouse and they copy the existing transaction history from the old to the new system.
- Then it's a matter of incrementally loading transactions as they arise, thus ensuring the transaction history is tracked.
- With incremental loading, the target data store is appended to, such that only the changes are loaded.
- This is useful for accumulating historical data such as transactions, weather, and browsing history.
- The volume, velocity, and demand for the data determine whether the data is loaded in batches or streamed live



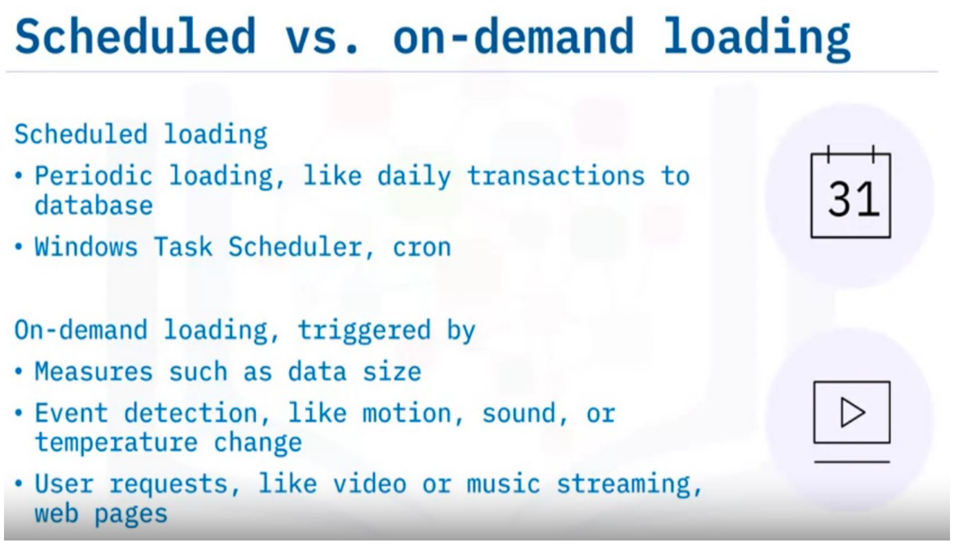
Full vs. incremental loading

Full loading

- Start tracking transactions in a new data warehouse
- Used for porting over transaction history

Incremental loading

- Data is appended to, not overwritten
- Used for accumulating transaction history
- Depending on the volume and velocity of data, can be batch loaded or stream loaded



Scheduled vs. on-demand loading

Scheduled loading

- Periodic loading, like daily transactions to database
- Windows Task Scheduler, cron

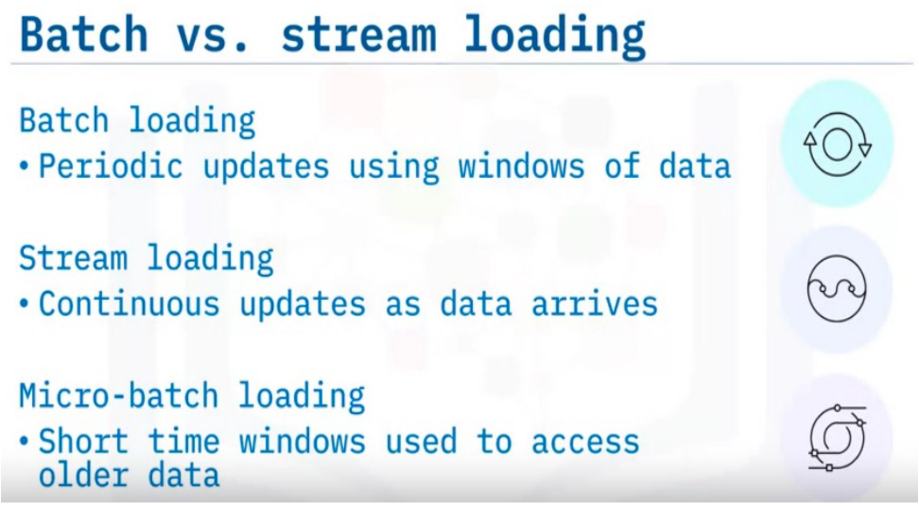
On-demand loading, triggered by

- Measures such as data size
- Event detection, like motion, sound, or temperature change
- User requests, like video or music streaming, web pages

- Data is often loaded on a schedule. For example: Daily point-of-sale transactions can be loaded into a database at the end of each day, during off-peak hours.
- Loading tasks can be s**cheduled** with tools such as **Windows Task Scheduler**, or with **cron** on Unix-like systems.
- On-demand loading is also very common, and relies on triggering mechanisms such as:

1. when the source data reaches a specified size.
2. when an **event is detected** by the source system, such as motion, sounds or temperature changes,
3. when a **user requests data**, such as online videos, music, or web pages.

- Batch and stream data loading are two ends of a spectrum of loading methods.
- **Batch loading** refers to loading data in chunks defined by some time windows of data accumulated by the data source, usually on the order of hours to days.
- **stream loading**, which loads data in real time as it becomes available.
- In between batch and stream loading, we have **micro-batch loadi**ng. This is used when imminent processes **need access to a small window of recent data.**



Batch vs. stream loading

Batch loading

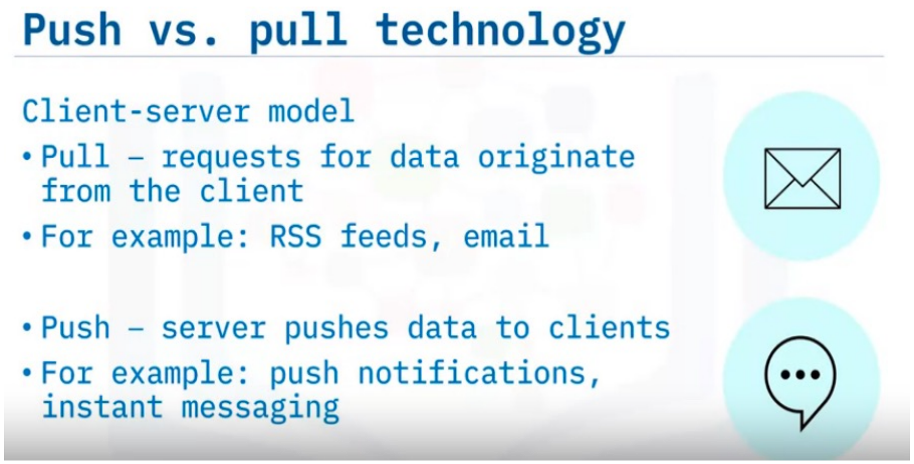
- Periodic updates using windows of data

Stream loading

- Continuous updates as data arrives

Micro-batch loading

- Short time windows used to access older data



Push vs. pull technology

Client-server model

- Pull – requests for data originate from the client
- For example: RSS feeds, email

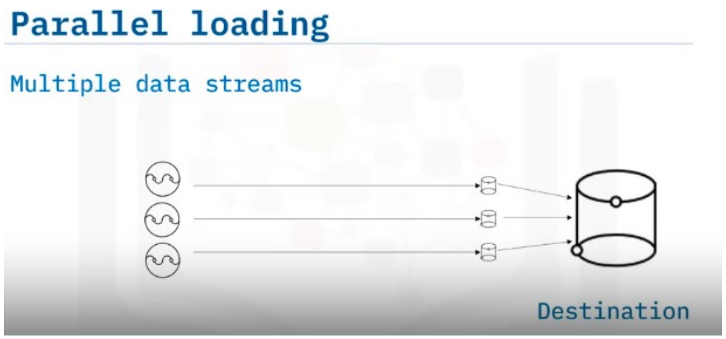
- Push – server pushes data to clients
- For example: push notifications, instant messaging

- **Push and pull** data-loading methods are based on a **client-server model.**
- A “pull” refers to a client initiating a request for data from a server.
- The server then responds to the client’s request and delivers the data.

Ex. **RSS feeds and email**.

- With “push” technology, the **client subscribes** to a service provided by a server, so that the server can then push data to the client as it becomes available.

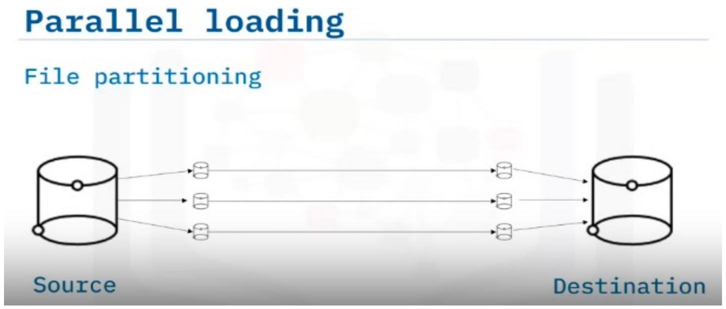
Ex. push notifications and instant messaging services.



Parallel loading

Multiple data streams

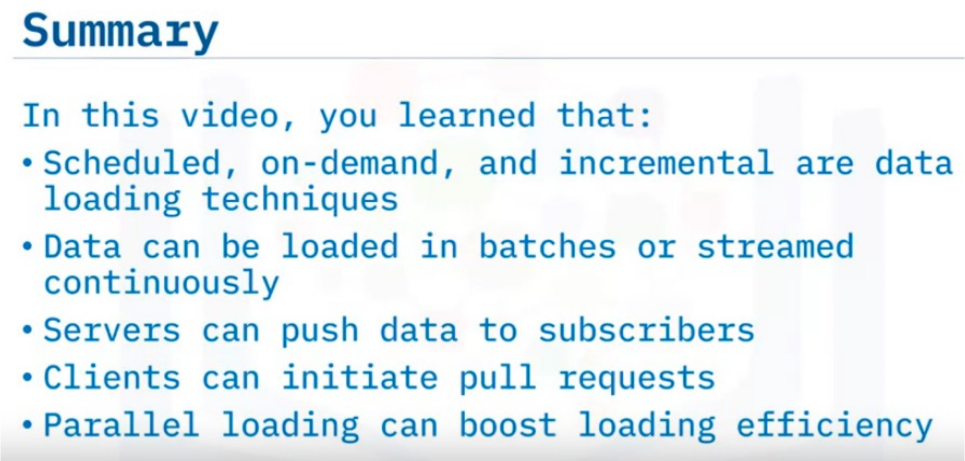
Parallel loading can be employed on multiple data streams to boost loading efficiency, particularly when the **data is big or has to travel long distances**.



Parallel loading

File partitioning

Similarly, by **splitting a single file** into smaller chunks, **the chunks** can be loaded simultaneously.



Summary

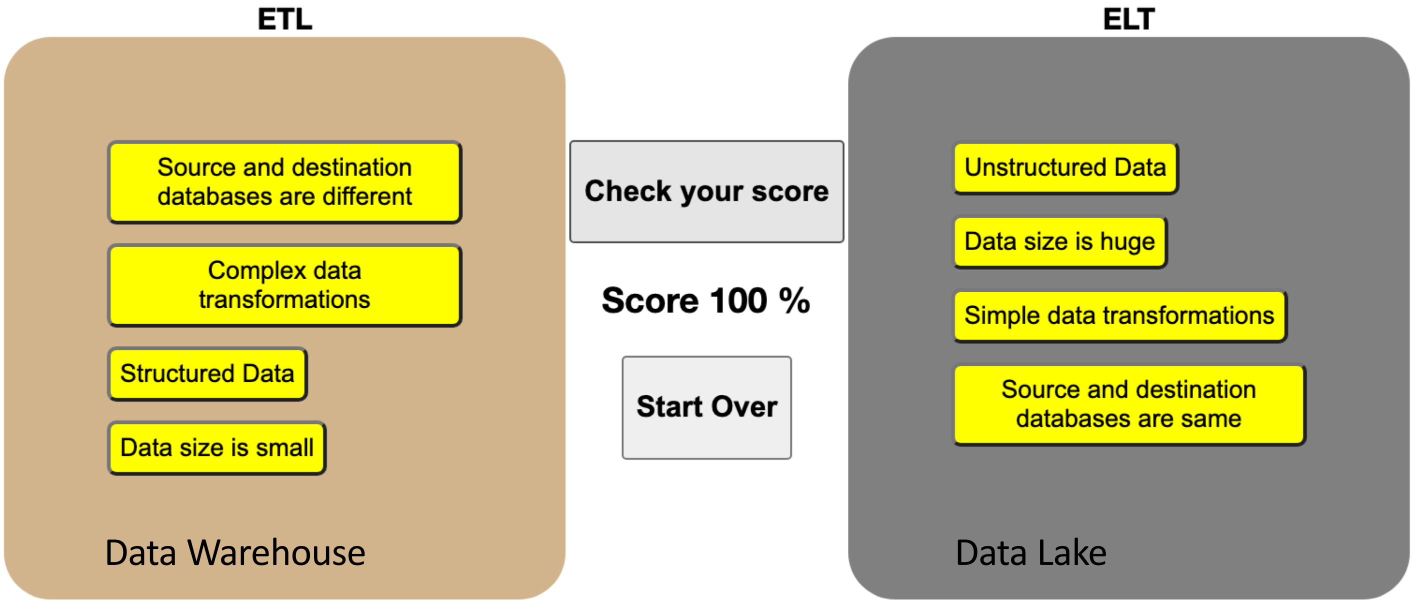
In this video, you learned that:

- Scheduled, on-demand, and incremental are data loading techniques
- Data can be loaded in batches or streamed continuously
- Servers can push data to subscribers
- Clients can initiate pull requests
- Parallel loading can boost loading efficiency

cognitiveclass.ai logo

Can you differentiate between ETL and ELT?

Drag and drop the below terms into the correct bins



ETL

- Source and destination databases are different
- Complex data transformations
- Structured Data
- Data size is small

Data Warehouse

Check your score

Score 100 %

Start Over

ELT

- Unstructured Data
- Data size is huge
- Simple data transformations
- Source and destination databases are same

Data Lake

Aspect	Data Lake	Data Warehouse
Data Structure	Raw, unprocessed, varied (structured, semi-structured, unstructured)	Structured, predefined schema
Data Processing	Processing at the time of analysis	Predefined transformations during ETL
Schema Flexibility	Schema-on-read	Predefined schema and structure
Data Integration	Wide range of data types and sources	Primarily structured data integration
Use Cases	Flexible analysis, exploratory analytics, machine learning	Reporting, querying, business intelligence
Storage Approach	Flat architecture	Relational database system

Summary & Highlights

Congratulations! You have completed this module. At this point, you know:

•ETL stands for Extract, Transform, and Load

•Loading means writing the data to its destination environment

•Cloud platforms are enabling ELT to become an emerging trend

•The key differences between ETL and ELT include the place of transformation, flexibility, Big Data support, and time-to-insight

•There is an increasing demand for access to raw data that drives the evolution from ETL, which is still used, to ELT, which enables ad-hoc, self-serve analytics

•Data extraction often involves advanced technology including database querying, web scraping, and APIs

•Data transformation, such as typing, structuring, normalizing, aggregating, and cleaning, is about formatting data to suit the application

•Information can be lost in transformation processes through filtering and aggregation

•Data loading techniques include scheduled, on-demand, and incremental

•Data can be loaded in batches or streamed continuously