ETL Fundamentals

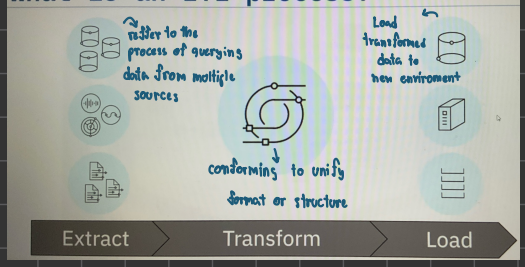
- ETL isn't automated data pipeline

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
- List use cases for ETL processes

What is an ETL process?



What is an ETL process?

E=Extract: Extracting data from a source

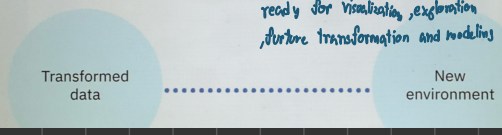
What is an ETL process?

T=Transformation:

Transforming data into the format for the output

What is an ETL process?

L=Load: Loading data into a database, data warehouse or other storage



What is Extraction?

- Configuring access to data and reading it into an application:
  - Web scraping
  - Connecting programmatically via APIs
- The data may be static or streaming online

What is data transformation?

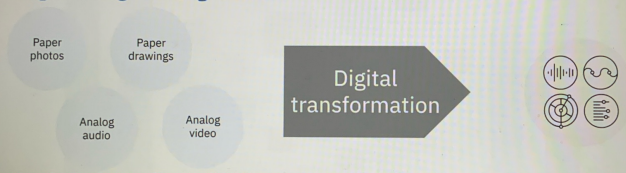
- Processing data
- Conforming to target systems and use cases
- Cleaning
- Filtering
- Joining
- Feature engineering
- Formatting and data typing

What is data loading?

- Moving data into a new environment
- Examples: a database, data warehouse, or data mart
- Making the data readily available for analytics, dashboards, reports

Use cases for ETL pipelines

- Digitizing analog media



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

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

What is an ELT process?

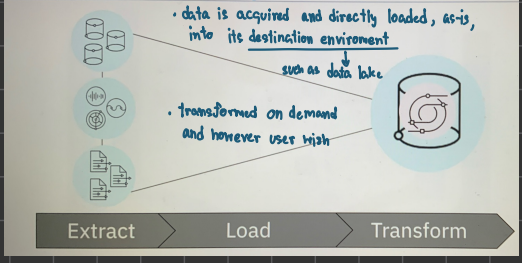
↳ Automated data pipeline

Extract → Load → Transform

Note:
- The ETL process has similar stages involved
- Order of performing ETL stages is different

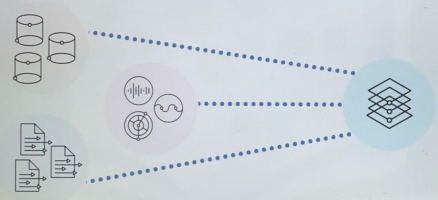
What is an ELT process?

- data is acquired and directly loaded, as-is, into its destination environment
  - such as data lake
- transformed on demand and however user wish



What is the Extraction process?

E=Extract: Extracting data from sources



What is the Loading process?

L=Load: Loading data as-is into destination system
↳ raw data as is

What is the Transform process?

T=Transform: Transforming data on demand
↳ more dynamic

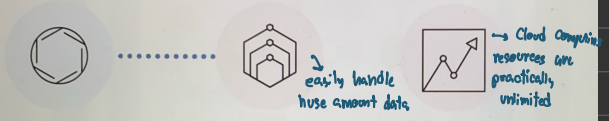
ELT use cases

Typically fall within the high-performance computing and Big Data realms

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

ELT is an emerging trend

- Big Data -> Cloud computing ← This solutions are evolving at tremendous rate due to demand Big data



- ELT separates the data pipeline from the processing
- More flexibility
  - because we are working on a replica of the source data
- No information loss

ETL vs ELT

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 before data reached to destination
- Transformations for ELT happen in the destination environment

Flexibility:

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

Differences between ETL and ELT

Support for Big Data:

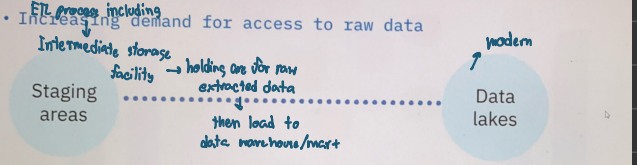
- Organizations use ETL for relational data, on-premise – scalability is difficult
- ELT solves scalability problems, handling both structured and unstructured Big Data in the cloud

Time-to-insight: (user must wait for changes)

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

The evolution of ETL to ELT

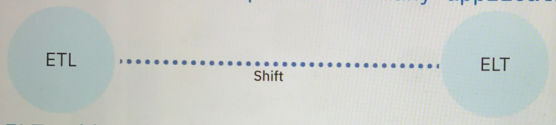
- Increasing demand for access to raw data



- 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"

The shift from ETL to ELT

ETL still has its place for many applications



ELT addresses key pain points:

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

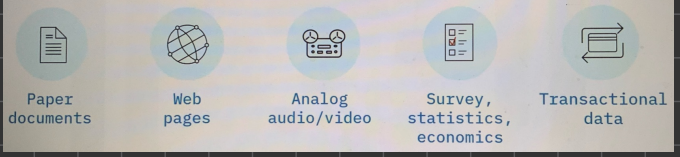
Data Extraction

Objectives

After watching this video, you will be able to:

- List examples of raw data sources
- Describe data extraction techniques
- Relate use cases with data sources and extraction techniques

Examples of raw data sources



Examples of raw data sources

- Data is everywhere

Techniques for extracting data

Data extraction techniques include:

- OCR Optical Character Recognition : used for text scanned from paper
- ADC sampling, CCD sampling → charge-couple devices → capture and digitize images
- Mail, phone, or in-person surveys and polls
- Cookies, user logs for tracking human or system behavior

Analog → Digital

Techniques for extracting data

More techniques include:

- Web scraping
- APIs
- Database querying SQL, NoSQL
- Edge computing
- Biomedical devices

Use cases

- Integrating disparate structured data sources via APIs
- Capturing events → to capture periodic or asynchronous events to store 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

Data Transformation

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

↳ mainly about formatting the data to suit the application

Data transformations can involve various operations, such as:

- Data typing
- Data structuring ↗ converting one data format to another such as json, .csv to table
- Anonymizing, encrypting

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

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

Schema-on-write is the conventional ETL approach:

- Consistency and efficiency
- Limited versatility

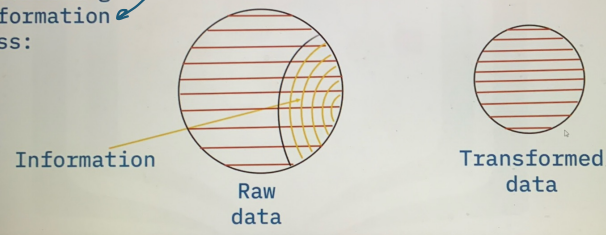
Schema-on-read applies to the modern ELT approach:

- Versatility
- Enhanced storage flexibility = more data

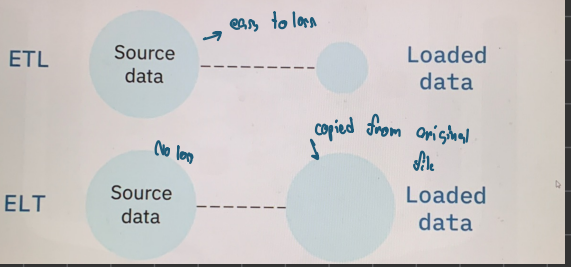
Information loss in transformation

there are many ways in which information can be 'lost' in transformation

Visualizing information loss:



Information loss in transformation



Information loss in transformation

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

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

Data Loading

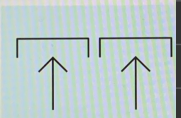
Objectives

After watching this video, you will be able to:

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

Data loading techniques

- Full — load an initial history into database
- Incremental — is applied to insert new data or to update already loaded data
- Scheduled — schedule data loading
- On-demand — load it as required
- Batch and stream
- Push and pull — push to server or clients by server
- Parallel and serial



Full vs. incremental loading

Full loading → loading data in one large batch

- Ex: Start tracking transactions in a new data warehouse
- Used for porting over transaction history — copy the existing transaction history from the old to the new system

Incremental loading → to ensure the transaction history is tracked

- 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 — data is often loaded on schedule

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

On-demand loading, triggered by ex:

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

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 (based on)

- Pull – requests for data originate from the client — then server responds to client's request and delivers the data
- For example: RSS feeds, email — email box

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

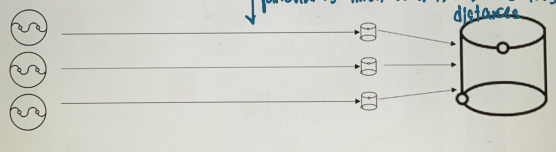
Parallel loading

employed to

Multiple data streams

→ to boost loading efficiency

↓ particularly when data is big and long distance



Parallel loading

File partitioning

↘ splitting file into smaller chunks, the chunks can be loaded simultaneously