Introduction

A diagram of a computer

Description automatically generated with medium confidence

M1 – Data Processing Techniques

A screenshot of a computer

Description automatically generated

M2 – ETL and Data Pipelines Tools and Techniques

A screenshot of a computer

Description automatically generated

M3 – Building Data Pipelines Using Airflow

A screenshot of a computer

Description automatically generated

M4 – Building Streaming Pipelines Using Kafka

A screenshot of a computer

Description automatically generated

M5 Final ProjectA screenshot of a computer

Description automatically generated

**ETL Fundamentals**

**Objectives**

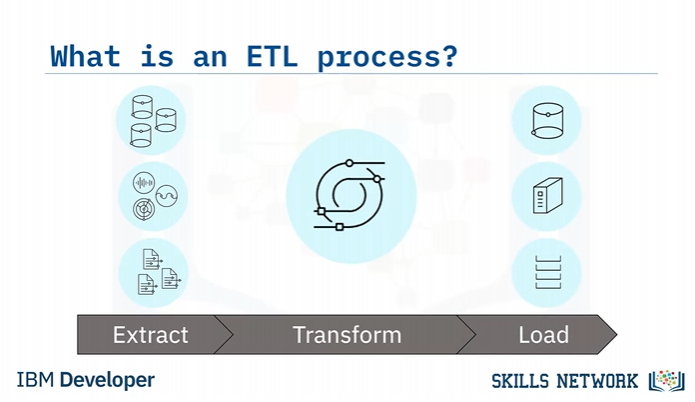
A white and blue screen with blue text

Description automatically generated

**What is an ETL process**

A screen shot of a computer screen

Description automatically generated



**Extraction**

A diagram of data processing

Description automatically generated

The Extraction process obtains or reads the data from one or more sources.

A screen shot of a computer

Description automatically generated

A diagram of data processing

Description automatically generated

The transformation process wrangles the data into

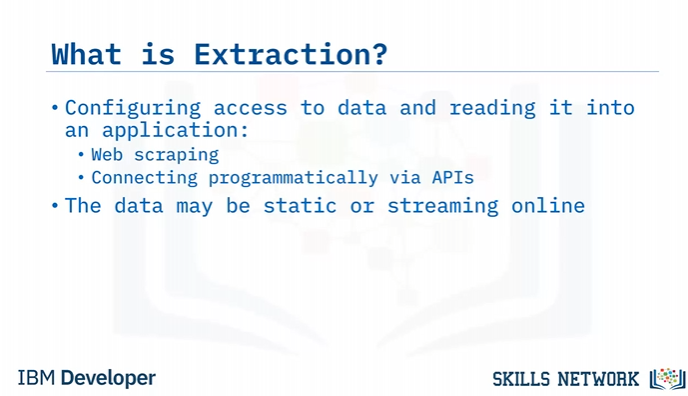
a format that is suitable for its destination and its intended use.

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**



**What is data transformation/ data wrangling**

A white text on a blue background

Description automatically generated

**What is data loading**

A screen shot of a computer

Description automatically generated

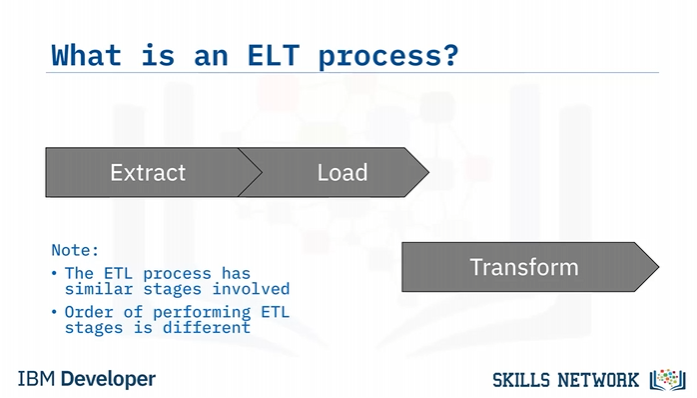
**Use cases for ETL Pipeline**

A white background with blue text

Description automatically generated

**ELT Basics**

**What is ELT Process**



A screenshot of a computer

Description automatically generated

**Extraction**

A diagram of data extraction process

Description automatically generated

The Extraction process obtains the data from

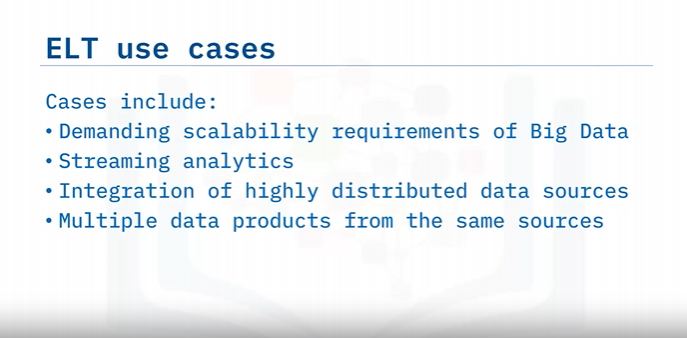
all sources and reads the data, often in an asynchronous fashion, into an application.​

The Loading process takes the raw data as-is,and

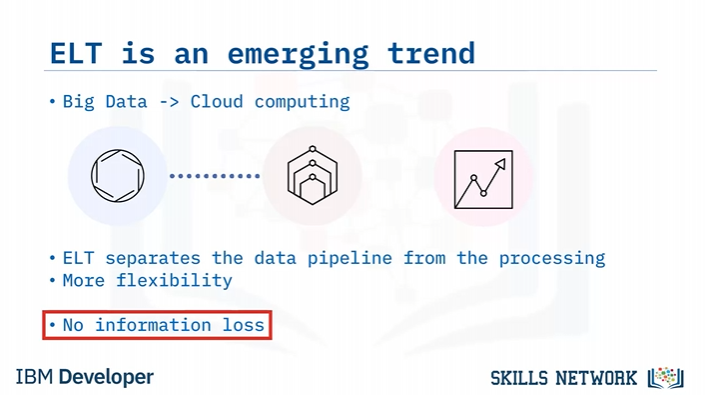
loads it into its new environment, where modern analytics tools can then be used directly.

The Transformation process for ELT is much more dynamic than it is for conventional ETL.

**ELT Use cases**



**Why is ELT is an emerging trend**



Why is ELT emerging?

Firstly, 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.

Of course, 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.

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 in a cost-efficient manner.

ELT is an emerging trend mainly because cloud platform technologies are enabling it.

**Comparing ETL to ELT**

**Difference between ETL and ELT**

**When and where the transformations happen**

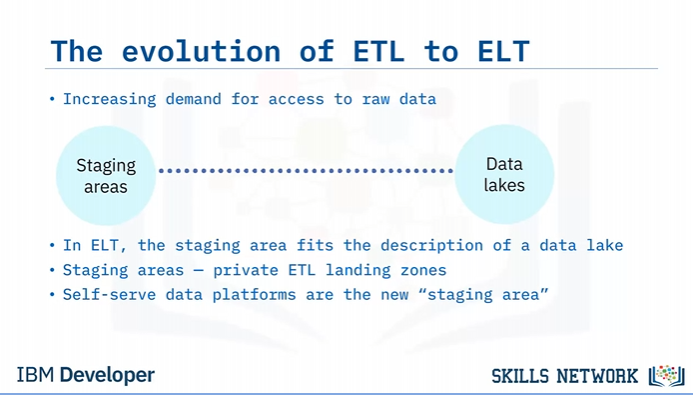


**Support for big data**

A blue and white text on a white background

Description automatically generated

**The evolution of ETL and ELT**



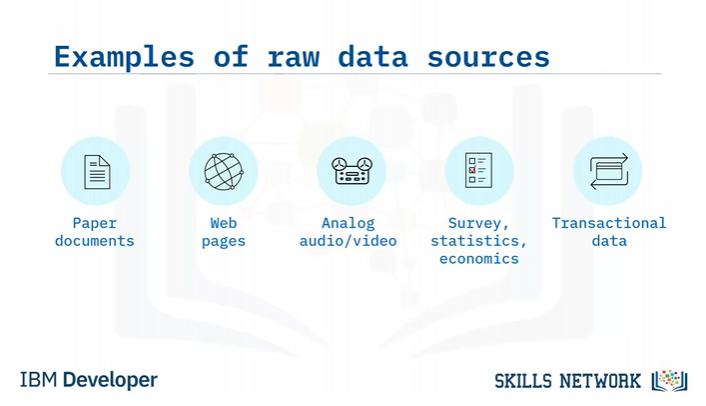
**The shift from ETL and ELT**

A close-up of a form

Description automatically generated

**Data Extraction Techniques**

**Example of raw data sources**



**Examples of raw data sources**

A screenshot of a computer

Description automatically generated

**Techniques for extracting data**

A screenshot of a computer

Description automatically generated

A white screen with blue text

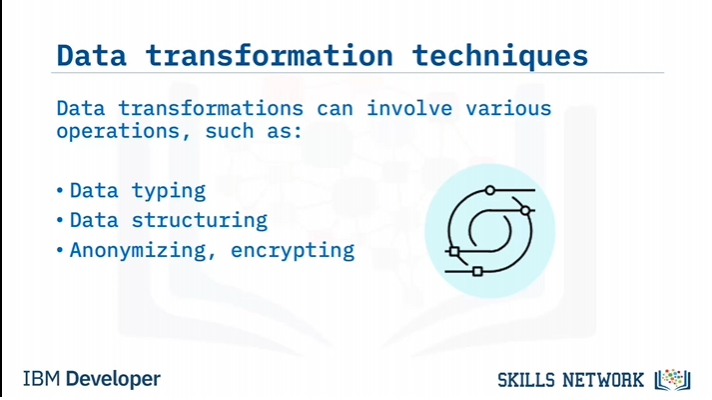
Description automatically generated

**Use cases**

A white and blue text on a white background

Description automatically generated

**Introduction to Data Transformation Techniques**



Data transformation is mainly about formatting the data to suit the application.

This can involve many kinds of operations, such as:

Data typing, which involves casting data to appropriate types, such as integer, float,

string, object, and category.

Data structuring, which includes converting one data format to another, such as JSON,

XML, or CSV to database tables.

Anonymizing and encrypting transformations to help ensure privacy and security.

A blue text on a white background

Description automatically generated

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**

A white text on a white background

Description automatically generated

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.

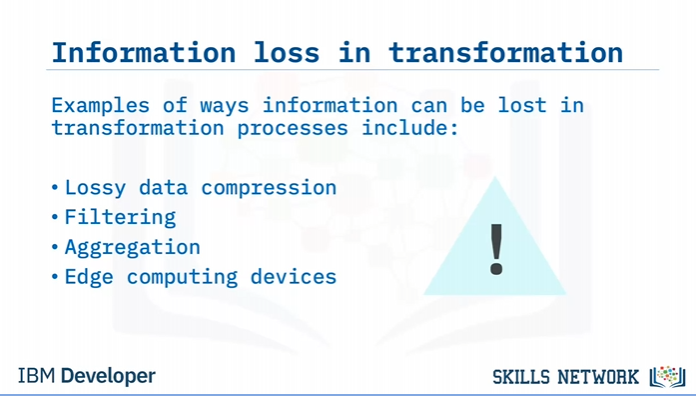
**Information loss in transformation**

A diagram of a diagram of a circular object

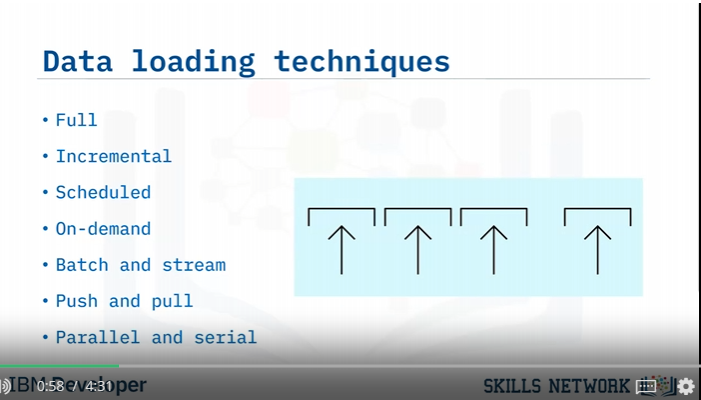
Description automatically generated

A diagram of data loss

Description automatically generated



**Data Loading Techniques**



**Full loading vs incremental loading**

A screen shot of a computer

Description automatically generated

**Scheduled vs on-demand loading**

A screen shot of a computer

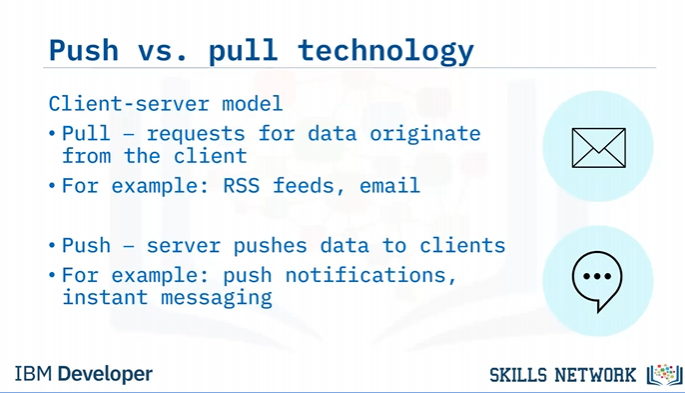
Description automatically generated

**Batch vs stream loading**

A screen shot of a computer

Description automatically generated

**Push vs pull technology**



**Parallel Loading**

A screen shot of a computer

Description automatically generated

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.

A diagram of a loading process

Description automatically generated

Similarly, by splitting a single file into smaller

chunks, the chunks can be loaded simultaneously.