Week 2

Thursday, April 22, 2021 5:08 PM

Overview of the Data Analyst Ecosystem

A data analyst's ecosystem includes the infrastructure, software, tools, frameworks, and processes used to gather, clean, analyze, mine, and visualize data.

Types of Data

- Structured
  - Data that follows a rigid format and can be organized into rows and columns. (Spreadsheet)
- Semi-structured
  - Mix of data that has consistent characteristics and data that does not conform to a rigid structure. (Email)
- Unstructured
  - Data that is complex and mostly qualitative information that cannot be structured into rows and columns. (Photos & videos)

Data comes in a variety of file formats , such as:

- Relational Database
- Non-Relational Database
- API's
- Web Services
- Data Streams
- Social Platforms
- Sensor Devices

Data Repositories

The type, format, and sources of data influence the type of data repositories that you can use to collect, store, clean, analyze, and mine the data for analysis.

- Databases
- Data Warehouses
- Data Marts
- Data Lakes
- Big Data Stores

If you're working **with big data, for example**, you will need big data warehouses, that allow you to store and process large-volume high-velocity data and also frameworks that allow you to perform complex analytics in real-time on big data.

Languages

- **Query Languages**
  - For example, SQL for querying and manipulating data

- **Programming languages**
  - For example, Python for developing data applications

- **Shell and Scripting languages**
  - For repetitive operational tasks

Automated tools, frameworks, and processes for all stages of the analytics process are part of the Data Analysts ecosystem. From tools used for gathering, extracting, transforming, and loading data into data repositories, to tools for data wrangling, data cleaning, data mining, analysis, and data visualization — it's a very diverse and rich ecosystem.

What is Data?

Generally, data comprises of facts, observations, perceptions, numbers, characters, symbols,
and images that can be interpreted to derive meaning. One of the ways in which data can be categorized is by its structure.

Data can be:

- **Structured**
  - SQL Databases
  - Online Transaction Processing
  - Spreadsheets
  - Online forms
  - Sensors GPS and RFID
  - Network and Web server logs

- **Semi-structured**
  - Has some organizational properties but lacks a fixed or rigid scheme
  - Cannot be stored in the forms of rows and columns as in databases
  - Contains tags and elements, or metadata, which is used to group data and organize it in a hierarchy

  - XML and other markup languages
  - Binary executables
  - TCP/IP packets
  - Zipped files
  - Integration of data

- **Unstructured**
  - Does not have an easily identifiable structure. Cannot be organized in a mainstream relational database in the form of rows and columns. Does not follow any particular format, sequence, semantics, or rules.

  - Web pages
  - Social media feeds
  - Images in varied file formats
  - Videos and Audio files
  - Documents and PDF files
  - PowerPoint presentations
  - Surveys

What is Data?

Common Sources of data:

- Relational Databases
- Flat files and XML Datasets
- API's (Application Program Interfaces) and Web Services
- Web Scraping
- Data Streams and Feeds

Languages for Data Professionals

Common languages

- Query languages
    - Are designed for accessing and manipulating data in a database (SQL)
- Programming languages
    - Are designed for developing applications and controlling application behavior (Python, R, Java)
- Shell scripting
    - Are designed for repetitive and time-consuming operational tasks (Unix/Linux Shell, PowerShell)

SQL - Structured Query Language - is a querying language designed for accessing and manipulating information from, mostly, though not exclusively, relational databases.

- Is portable and platform independent
- Can be used for querying data in a wide variety of databases and data repositories
- Has a simple syntax that is similar to the English language
- Its syntax allows developers to write programs with fewer lines of code using basic keywords
- Can retrieve large amounts of data quickly and efficiently
- Runs on an interpreter system

Python - is a widely used open-source, general-purpose, high-level programming language.

- Its syntax allows programmers to express their concepts in fewer lines of code
- An ideal tool for beginning programmers because of its focus on simplicity and readability
- Great for performing high-computational tasks in large volumes of data
- Has in-built functions for frequently used concepts
- Support multiple programming paradigms (object-oriented, imperative, functional and procedural)

    - **Functionalities**
        - **Pandas** for data cleaning and analysis
        - **Numpy** and **Scipy**, for statistical analysis
        - **Beautifulsoup** and **Scrapy** for web scraping
        - **Matplotlib** and **Seaborn** to visually represent data in the form of bar graphs, histogram, and pie-charts

R - programming - R is an open-source programming language and environment for data analysis, data visualization, machine learning, and statistics.

Widely used for:

- Developing statistical software
- Performing data analytics
- Creating compelling visualizations

Key Benefits:

- Open-source
- Platform-independent
- Can be paired with many programming languages
- Highly extensible
- Facilitate the handling of structure and unstructured data
- Includes libraries such as **Ggplot2** and **Plotly** that offer aesthetic graphical plots to its users
- Allows data and scripts to be embedded in reports
- Allows creation of interactive web apps
- Can be used for developing statistical tools

Java - Is an object-oriented, class-based and platform-independent programming language originally developed by Sun Microsystems.

- One of the top-ranked programming languages used today
- Used in a number of data analytics processes - cleaning data, importing and exporting data, statistical analysis, data visualization
- Used in the development of big data frameworks and tools - Hadoop, Hive, Spark

Unix / Linux Shell - is a computer program written for the UNIX shell. It is a series of UNIX commands written in a plain text file to accomplish a specific task.

Typical operations performed by shell scripts include:

- file manipulation
- program execution
- system administration tasks such as disk backups and evaluating system logs
- installation scripts for complex programs
- executing routine backups
- running batches

PowerShell - is a cross-platform automation tool and configuration framework by Microsoft that is optimized for working with structured data formats, such as JSON, CSV, XML, and REST APIs, websites, and office applications.

- Consists of command-line shell and scripting language
- It is object-based and can be used to filter, sort, measure, group and compare objects as they pass through a data pipeline
- Used for data mining, building GUIs, creating charts, dashboards, and interactive reports

Overview of Data Repositories

Types of data repositories include:
BDMS - Database Management System

- **Databases**
  - Collection of data for input, storage, search, retrieval, and modification of data.
  - **DBMS** - is a set of programs that creates and maintains the database. It allows you to store, modify, and extract information from the database using a function called **querying**.

Relational vs Non-Relational Databases

- Relational
  - Data is organized into a tabular format with rows and columns
  - Well-defined structure and schema
  - Optimized for data operations and querying
  - Use SQL as the standard querying language

- Non-Relational
  - Emerged in response to the volume, diversity and speed at which data is being generated today
  - Built for speed, flexibility and scale
  - Data can be stored in a schema-less form
  - Widely used for processing big data

- **Data Warehouses**

  - Consolidates data through the extract, transform, and load process, also known as the **ETL** process, into one

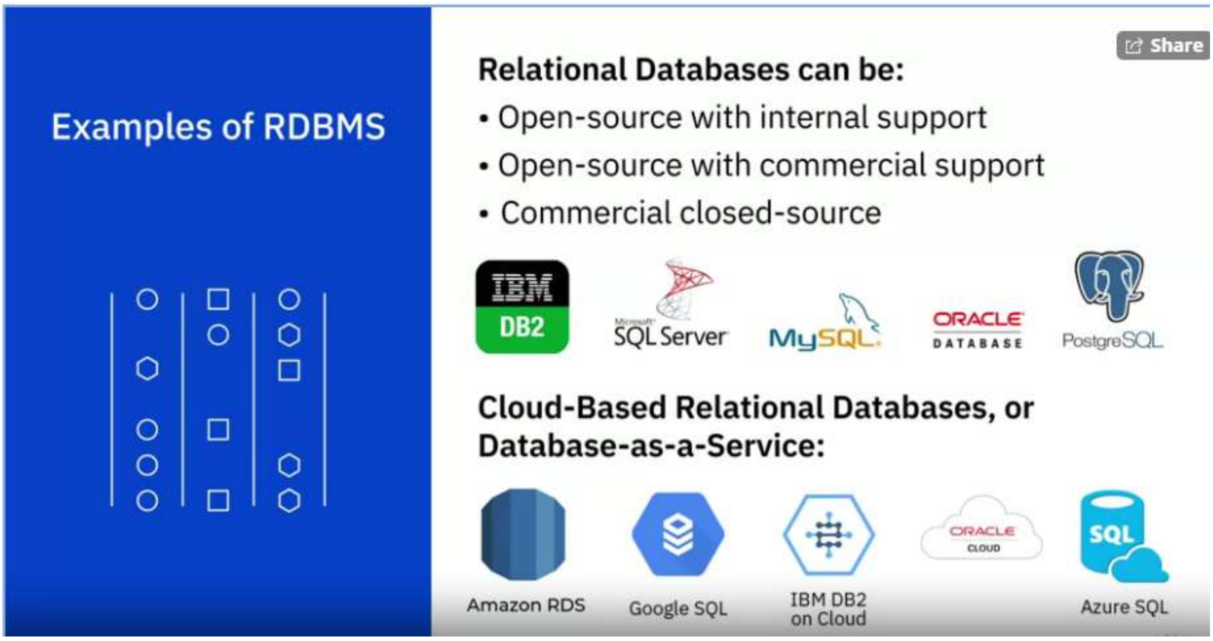
    comprehensive database for analytics and business intelligence.
  - **ETL process -** Extract data from different data sources. Transform the data into a clean and usable state. Load the data into data repository.

- **Big Data Stores** - Distributed computational and storage infrastructure to store, scale and process very large data sets.

RDBMS - Relational Database Management Systems

RBDMS - A relational database is a collection of data organized into a table structure, where the tables can be linked, or related, based on data common to each. (Spreadsheet like data)

- Ideal for the optimized storage, retrieval, and processing of data for large volumes of data
- Each table has a unique set of rows and columns
- Relationships can be defined between tables
- Fields can be restricted to specific data types and values
- Can retrieve millions of records in seconds using SQL for querying data
- Security architecture of relational databases provides greater access control and governance



Examples of RDBMS

Relational Databases can be:

- Open-source with internal support
- Open-source with commercial support
- Commercial closed-source

Cloud-Based Relational Databases, or Database-as-a-Service:

Advantages of Relational Databases:

- Create meaningful information by joining tables
- Flexibility to make changes while the database is in use
- Minimize data redundancy by allowing relationships to be defined between tables
- Offer export and import options that provide ease of backup and disaster recovery
- Are ACID compliant, ensuring accuracy and reliability in database transactions

Relational Databases are well suited for:

- **Online Transaction Processing application (OLTP)**
  - Can support transaction-oriented tasks that run at high rates
    - Accommodate large number of users
    - Manage small amounts of data
    - Support frequent queries and fast response times

- **Data Warehouses**
    - Can be optimized for analytical processing (OLAP)

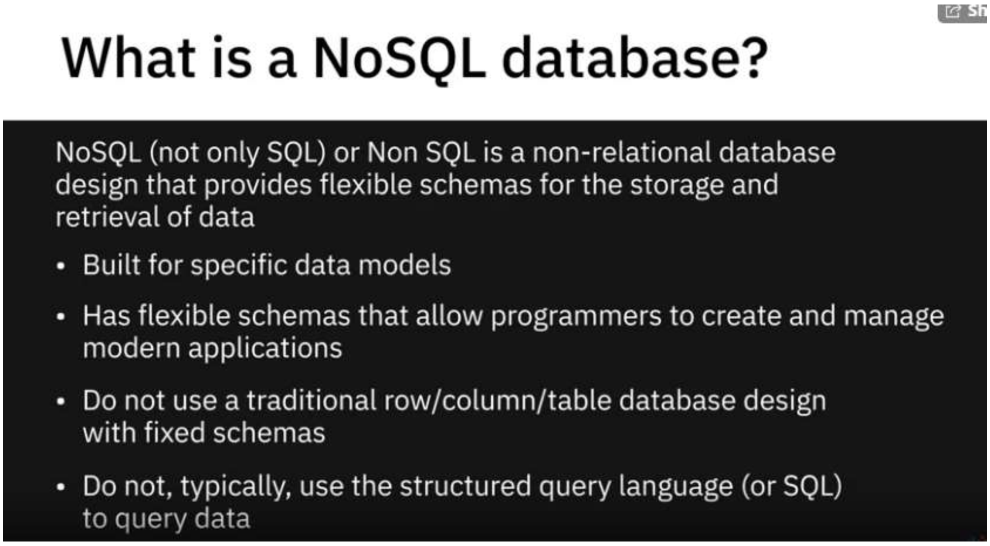
- **IoT Solutions**
    - Provide the speed and ability to collect and process data from edge devices

Limitations of RDBMS

- Does not work well with semi-structured and unstructured data
- Migration between two RDBMS's is possible only when the source and destination tables have identical schemas and data types
- Entering a value greater than the defined length of a data field results in loss of information

NoSQL - "Not only SQL"

NoSQL -is a non-relational database design that provides flexible schemas for the storage and retrieval of data.
- Allows data to be stored in a schema-less or free-form fashion.



What is a NoSQL database?

NoSQL (not only SQL) or Non SQL is a non-relational database design that provides flexible schemas for the storage and retrieval of data

- Built for specific data models
- Has flexible schemas that allow programmers to create and manage modern applications
- Do not use a traditional row/column/table database design with fixed schemas
- Do not, typically, use the structured query language (or SQL) to query data

There are four common types of NoSQL databases:

- **Key-value store (Redis; Memcache, DynamoDB)**
    - Data in a key-value database is stored as a collection of key-value pairs.
    - The key represents an attribute of the data and is a unique identifier.
    - Both keys and values can be anything from simple integers or strings to complex JSON documents.
    - Great for storing user session data and user preferences, making real-time recommendations and targeted advertising, and in-memory data caching
        - **Not great if you want to: Query data on specific data value; Need relationships between data values; Need multiple unique keys**

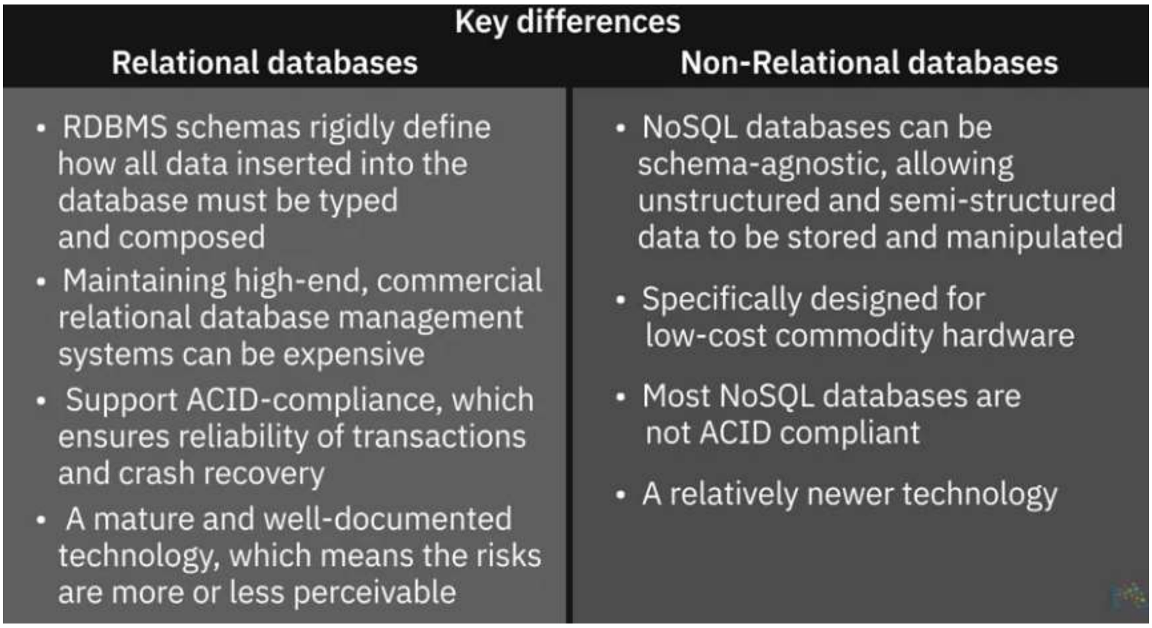
- **Document Based (MongoDB, Document DB, CouchDB, Cloudant)**
    - Document databases store each record and its associated data within a single document.
    - They enable flexible indexing, powerful ad hoc queries, and analytics over collections of documents.
    - Document databases are preferable for eCommerce platforms, medical records storage, CRM platforms, and analytics platforms.

        - **Not great if you want to: Run complex search queries; Perform multi-operation transactions**
    -
- **Column Based (Cassandra, Hbase)**
    - Data is stored in cells grouped as columns of data instead of rows
    - A logical grouping of columns is referred to as a column family
    - All cells corresponding to a column are saved as a continuous disk entry, making access and search easier and faster.
    - Great for systems that require heavy write requests, storing time-series data, weather data, and IoT data.
        - **Not great fit if you want to: Run complex queries; Change querying patterns frequently**

- **Graph Based**
    - Graph-based databases use a graphical model to represent and store data.
    - Useful for visualizing, analyzing and finding connections between different pieces of data.
    - An excellent choice for working with connected data
    - Great for Social Networks, Product recommendations, Network diagrams, Fraud detection, Access management
        - **Not great fit if you want to: Process high volumes of transactions**

Advantages of NoSQL

- Its ability to handle large volumes of structured, semi-structured, and unstructured data
- Its ability to run as a distributed system scaled across multiple data centers
- An efficient and cost-effective scale-out architecture that provides additional capacity and performance with the addition of new nodes
- Simples design, better control over availability, and improved scalability that makes it agile, flexible, and support quick iterations

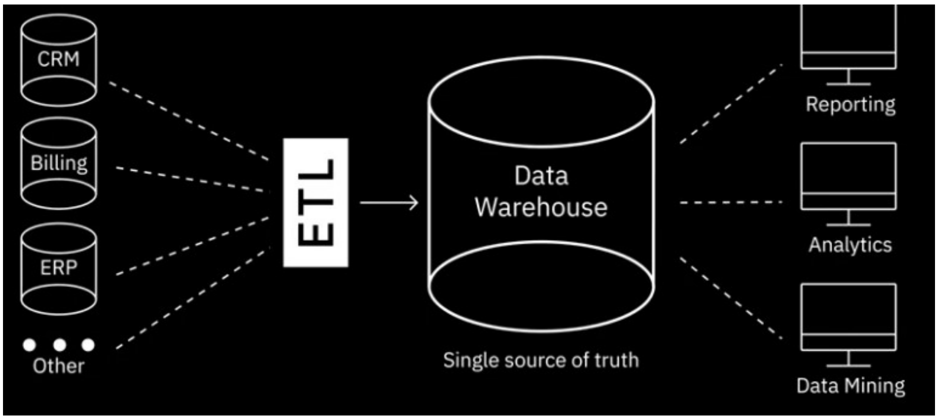


Key differences

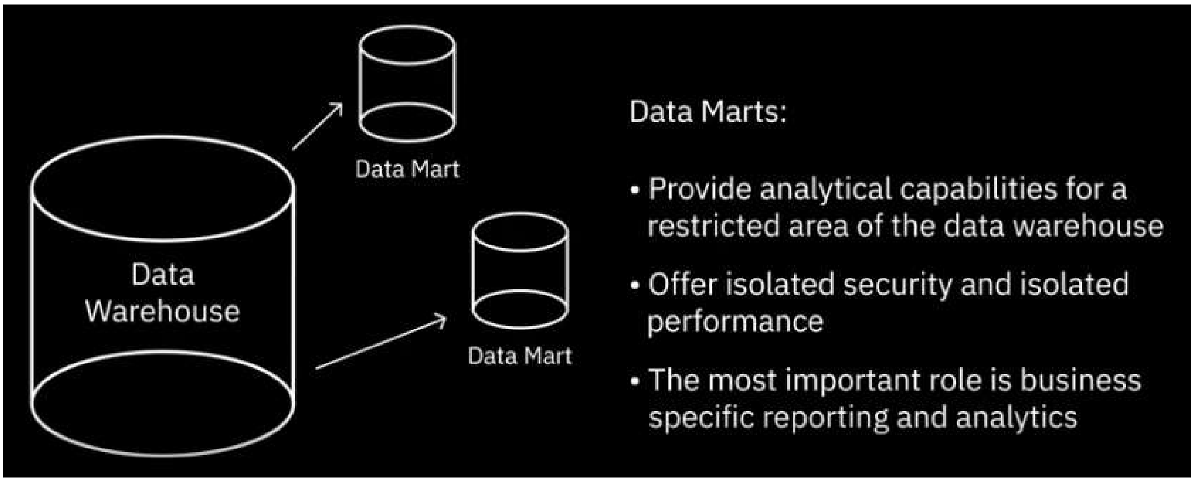
Relational databases	Non-Relational databases
• RDBMS schemas rigidly define how all data inserted into the database must be typed and composed • Maintaining high-end, commercial relational database management systems can be expensive • Support ACID-compliance, which ensures reliability of transactions and crash recovery • A mature and well-documented technology, which means the risks are more or less perceivable	• NoSQL databases can be schema-agnostic, allowing unstructured and semi-structured data to be stored and manipulated • Specifically designed for low-cost commodity hardware • Most NoSQL databases are not ACID compliant • A relatively newer technology

Data Marts, Data Lakes, ETL and Data Pipelines

Data Warehouses - Is a multi-purpose enabler of operational and performance analytics.



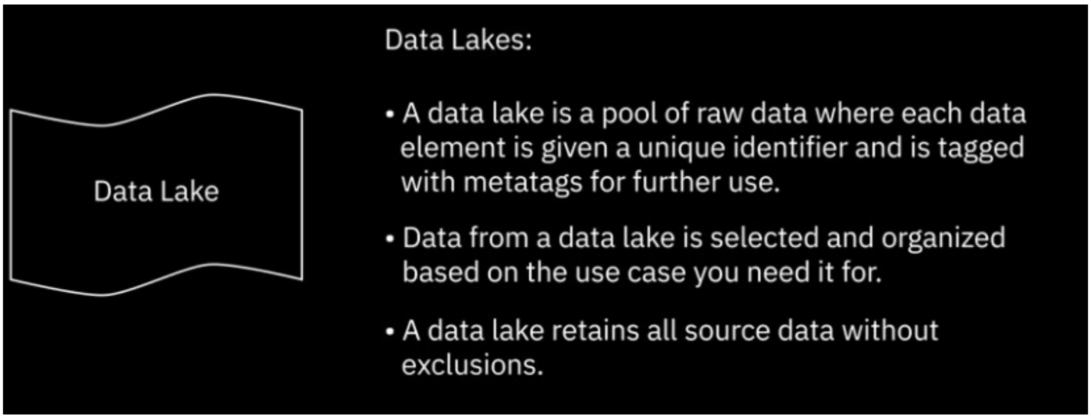
Data Mart- is a sub-section of the data warehouse, built specifically for a particular business function, purpose, or community of users.



Data Marts:

- Provide analytical capabilities for a restricted area of the data warehouse
- Offer isolated security and isolated performance
- The most important role is business specific reporting and analytics

Data Lake - is a storage repository that can store large amounts of structured, semi-structured, and unstructured data in their native format, classified and tagged with metadata. The most important role of a data lake is in predictive and advanced analytics.



Data Lakes:

- A data lake is a pool of raw data where each data element is given a unique identifier and is tagged with metatags for further use.
- Data from a data lake is selected and organized based on the use case you need it for.
- A data lake retains all source data without exclusions.

Extract, Transform, and Load Process (ETL) - automated process which includes
ETL has historically been used for batch workloads on a large scale. However, with the emergence of streaming ETL tools, they are increasingly being used for real-time streaming event data as well.

- Gathering raw data
- Extracting information needed for reporting and analysis
- Cleaning, standardizing, and transforming data into usable format
- Loading data into a data repository

Extraction can be through:

- **Batch processing**
    - Graph-based databases use a graphical model to represent and store data.
- **Stream processing**
    - Data pulled in real-time from source, transformed in transit, and loaded into data repository

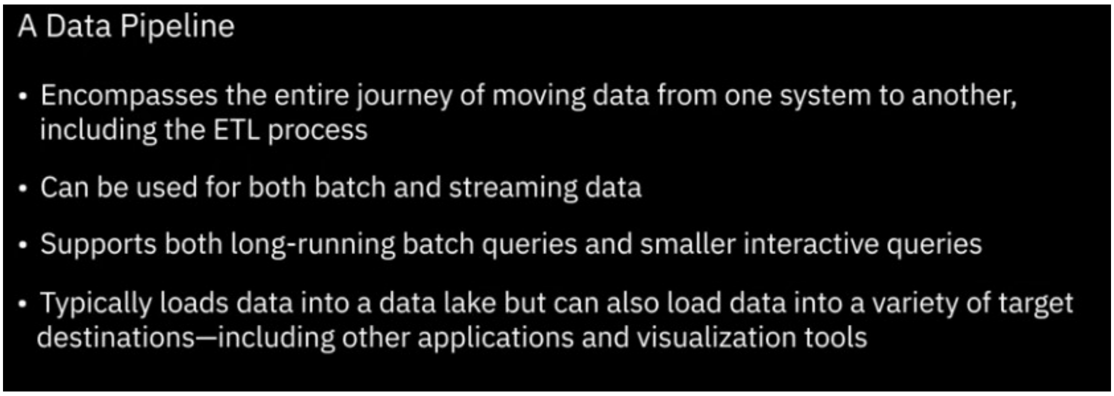
Transform - involves the execution of rules and functions that converts raw data into data that can be used for analysis.

- Standardizing date formats and units of measurement
- Removing duplicate data
- Filtering out data that is not required
- Enriching data
- Establishing key relationships across tables

Loading - is the transportation of processed data into a data repository. It can be:

- Initial loading - populating all of the data in the repository
- Incremental loading - applying updates and modifications periodically
- Full refresh - erasing a data table and reloading fresh data

Data Pipeline



A Data Pipeline

- Encompasses the entire journey of moving data from one system to another, including the ETL process
- Can be used for both batch and streaming data
- Supports both long-running batch queries and smaller interactive queries
- Typically loads data into a data lake but can also load data into a variety of target destinations—including other applications and visualization tools

Foundations of Big Data

Ernst and Young offers the following definition: big data refers to the dynamic, large, and disparate volumes of data being created by people, tools, and machines. It requires new, innovative and scalable technology to collect, host, and analytically process the vast amount of data gathered in order to drive real-time business insights that relate to consumers,
risk, profit, performance, productivity management, and enhanced shareholder value.

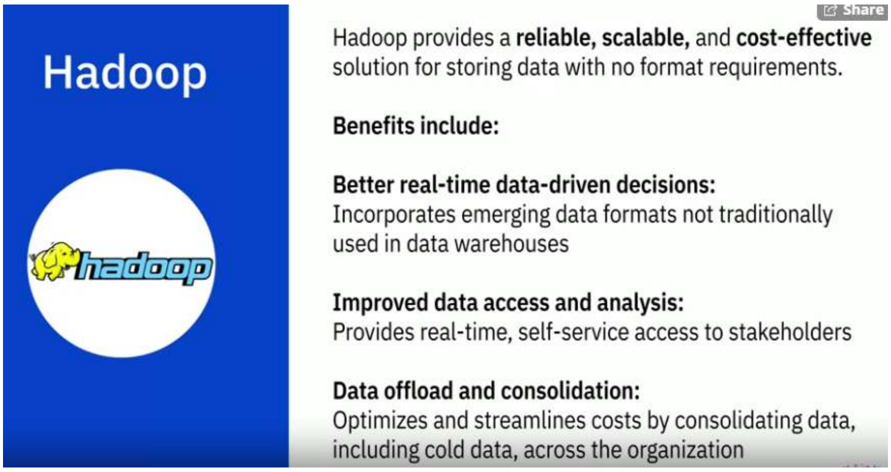
The V's of Big Data

- **Velocity**
- **Volume**
- **Variety**
- **Veracity**
- **Value**

Big Data Processing Tools

The Big Data processing technologies provide ways to work with large sets of structured, semi-structured, and unstructured data so that value can be derived from big data.

- **Apache Hadoop -** A Collection of tools that provides distributed storage and processing of big data.



Hadoop

Hadoop provides a **reliable, scalable,** and **cost-effective** solution for storing data with no format requirements.

Benefits include:

Better real-time data-driven decisions:
Incorporates emerging data formats not traditionally used in data warehouses

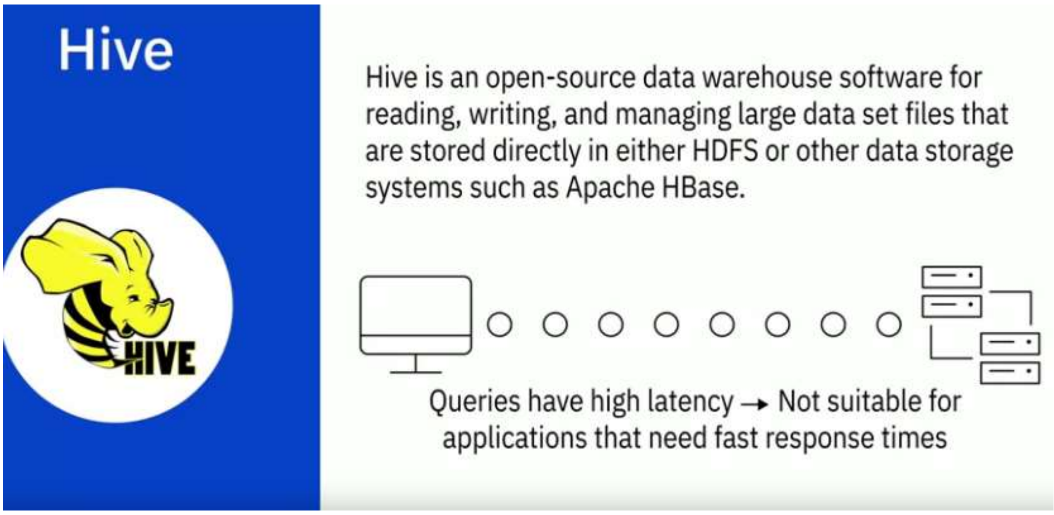
Improved data access and analysis:
Provides real-time, self-service access to stakeholders

Data offload and consolidation:
Optimizes and streamlines costs by consolidating data, including cold data, across the organization

Hadoop Distributed File System, or HDFS is a storage system for big data that runs on multiple commodity hardware connected through a network.

- Provides scalable and reliable big data storage by partitioning files over multiples nodes
- Splits large files across multiple computers, allowing parallel access to them
- Replicates file blocks on different nodes to prevent data loss

- **Apache Hive -** A data warehouse for data query and analysis



Hive

Hive is an open-source data warehouse software for reading, writing, and managing large data set files that are stored directly in either HDFS or other data storage systems such as Apache HBase.

Queries have high latency → Not suitable for applications that need fast response times

Not suitable for:

- Transaction processing that involves a high percentage of write operations

Better suited for:

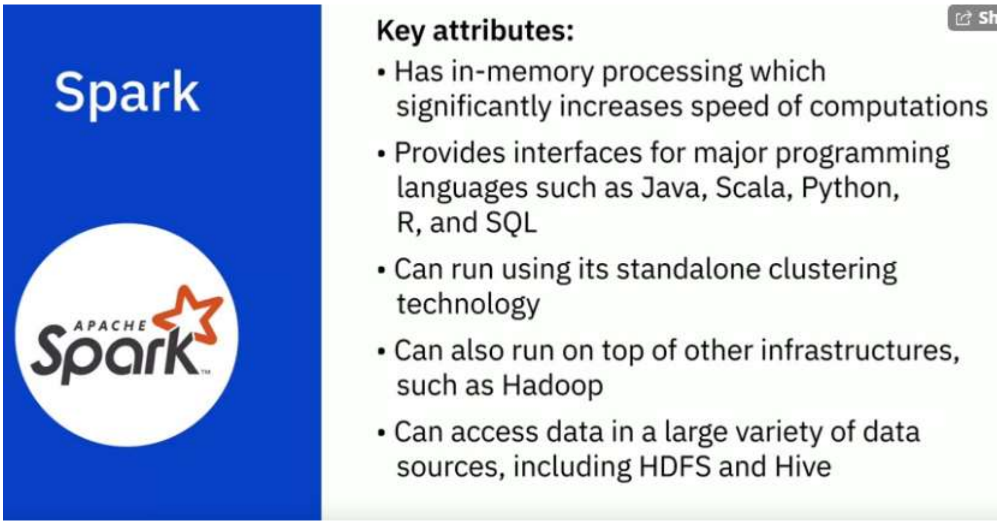
- Data warehousing tasks such as ETL, reporting, and data analysis
- Easy access to data via SQL

- **Apache Spark -** A distributed analytics framework for complex, real-time data analytics

  Sparks is a general-purpose data processing engine designed to extract and process large volumes of data for a wide range of applications.

  - Interactive analytics
  - Streams Processing

- Machine Learning
- Data integration
- ETL



Key attributes:

- Has in-memory processing which significantly increases speed of computations
- Provides interfaces for major programming languages such as Java, Scala, Python, R, and SQL
- Can run using its standalone clustering technology
- Can also run on top of other infrastructures, such as Hadoop
- Can access data in a large variety of data sources, including HDFS and Hive

Summary and Highlights

In this lesson, you have learned the following information:

A Data Repository is a general term that refers to data that has been collected, organized, and isolated so that it can be used for reporting, analytics, and also for archival purposes.

The different types of Data Repositories include:

- Databases, which can be relational or non-relational, each following a set of organizational principles, the types of data they can store, and the tools that can be used to query, organize, and retrieve data.
- Data Warehouses, that consolidate incoming data into one comprehensive storehouse.
- Data Marts, that are essentially sub-sections of a data warehouse, built to isolate data for a particular business function or use case.
- Data Lakes, that serve as storage repositories for large amounts of structured, semi-structured, and unstructured data in their native format.
- Big Data Stores, that provide distributed computational and storage infrastructure to store, scale, and process very large data sets.

ETL, or Extract Transform and Load, Process is an automated process that converts raw data into analysis-ready data by:

- Extracting data from source locations.
- Transforming raw data by cleaning, enriching, standardizing, and validating it.
- Loading the processed data into a destination system or data repository.

Data Pipeline, sometimes used interchangeably with ETL, encompasses the entire journey of moving data from the source to a destination data lake or application, using the ETL process.

Big Data refers to the vast amounts of data that is being produced each moment of every day, by people, tools, and machines. The sheer velocity, volume, and variety of data challenge the tools and systems used for conventional data. These challenges led to the emergence of processing tools and platforms designed specifically for Big Data, such as Apache Hadoop, Apache Hive, and Apache Spark.