Understanding Data Repositories and Big Data Platforms

Overview of Data Repositories

Data Repository is a general term used to refer to data that has been collected, organized, and isolated so that it can be used for business operations or minded for reporting and data

1. Databases

- Databases: Collection of data for input, storage, search, retrieval, and modification of data
    - ✓ Databases Management System (DMS) is a set of programs that creates and maintains the databases
    - ✓ It allows you to store, modify, and extract information from the databases using a function called querying
    - ✓ Ex: if you want to fine customers who have been inactive for six months using the query function, the databases management system will retrieve data of all customers from databases
- Factors governing choice of database include:
    - ✓ Data type
    - ✓ Data structure
    - ✓ Querying mechanisms
    - ✓ Latency requirements transaction speed
    - ✓ Intended use of data
- Relational (RDBMSes), build on the organizational principles of flat files, with data organized into a tabular format with rows and columns following well defined structure and schema, optimized for data operations and querying, use SQL as the standard querying language
- Non-relational, also known as NoSQL (Not Only SQL).
    - ✓ 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

2. Data Warehouse

- Consolidates data through the extract, transform, and load process, also known as the ELT process, into one comprehensive databases for analytics, and business intelligence
    - ✓ Extract data from different data sources
    - ✓ Transform the data into a clean and usable state
    - ✓ Load the data into data repository

3. Big Data Stores

- Distributed computational and storage infrastructure to store, scale, and process very large data sets

Summary

Data repositories help to isolate data and make reporting and analytics more efficient and credible while also serving as a data archive

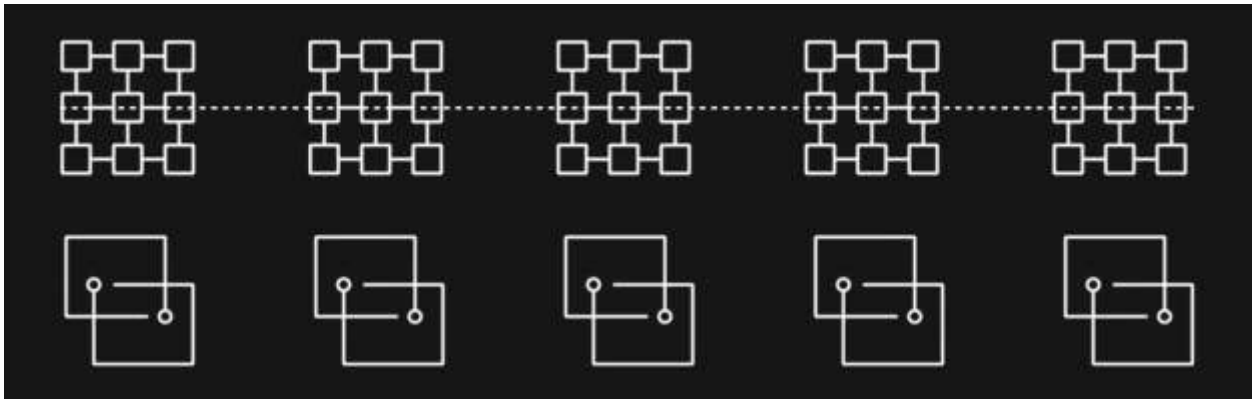
RDMS (Relational Databases Management Systems)

- 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 reach.
    - ✓ Rows: record
    - ✓ Attribute

Customer ID	Customer Name	Customer Address	Customer Phone
01234	Jim	Xxxx	Xxxxx
02345	Paul	Xxxx	Xxxx

Transaction Date	Customer ID	Transaction Amount	Payment method
Xxxx	**01234**	Xxxx	Xxxxx
Xxxx	**02345**	Xxxx	Xxxx

- ✓ The customer table and the transaction table can be related based on the common customer ID field.
- ✓ You can query the customer table to produce reports such as a customer statement that consolidates all transactions in a given period



- ✓ This capability of relating tables based on common data enables you to retrieve an entirely new table from data in one or more tables with a single query
- ✓ It also allows you to understand the relationships among all available data and gain new insights for making better decisions
- ✓ Relational databases use structured query language, or SQL, for querying data
- ✓ Similarities between relational databases and spreadsheets:
    - Relational databases build on the organizational principles of flat files such as spreadsheets, with data organized into rows and columns following a well-defined structure and schema
- ✓ Relational databases (vs spreadsheet)
    - 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
    - Relationship can be defined between tables
    - Field 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

✓ Relational databases can be:
- Open-source with internal support
- Open-source with commercial support
- Commercial closed-source
- Ex: IBM DB2, Microsoft SQL server, MYSQL, Oracle Databases, and PostgreSQL

✓ Cloud-Based Relational Databases, or Database-as-a-service:
- Are gaining wide use as they have access to the limitless compute and storage capabilities offered by the cloud
- Ex: Amazon Relational Databases Service (RDS), Google Cloud SQL, IBM DB2 on Cloud, Oracle Cloud, and SQL Azure
- RDBMS is a mature and well-documented technology, making it easy to learn and find qualified talent
- Advantages of Relational Databases:
  - Create meaningful information by joining tables
  - Flexibility to make changes while the database is in use, or using SQL can add new columns, tables, rename relations, make other changes while the database is running and queries are happening.
  - Minimize data redundancy by allowing relationships to be defined between tables. Ex the information of a customer appears in a single entry in the customer table, and the transaction table pertaining to the customer stores a link to the customer table
  - Offer export and import options that provide ease of backup and disaster recovery. Relational databases offer easy export and import options, making backup and restore easy. Export can happen while the databases running, making restore on failure easy. Cloud-based relational databases do continuous mirroring, which means the loss of data on restore can be measured in seconds or less
  - Are ACID (Atomicity, Consistency, Isolation, and Durability) compliant, ensuring accuracy and reliability in database transactions

- ✓ Relational databases are well suited for:
    - **Online Transaction Processing (OLTP)do** application. Can support transaction-oriented tasks that run at high rates and accommodate large number of users, manage small amounts of data, support frequent queries and fast response times.
    - **Data Warehouses** can be optimized for online analytical processing (OLAP)
    - **IoT Solutions** Provide the speed and ability to collect and process data from edge devices
- ✓ Limitation 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

- NoSQL (Not only SQL) or Non SQL is a non-relational database design that provides flexible schemas for the storage and retrieval of data
    - ✓ Gained greater popularity due to the emergence of cloud computing, big data, and high-volume web and mobile applications
    - ✓ Chosen for their attributes around scale, performance and ease of use
    - ✓ 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
- NoSQL allows data to be stored in a schema-less or free-form fashion
- There are four common types of NoSQL databases:

1. Key-value store

- Data in a key-value database is stored as a collection of key-value pairs
- A key represents an attribute of the data and is a unique identifier
- Both keys and values can be anything from simple integers and strings to complex JSON documents

- Great for storing user session data, user preferences, real-time recommendations, targeted advertising, in-memory data caching
- Not a great fit if you want to:
    - ✓ Query data on specific data value
    - ✓ Need relationships between data values
    - ✓ Need multiple unique keys
- Ex: Redis, Memcached, DynamoDB

2. Document Based

- 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
- Preferred for eCommerce platforms, medical records storage, CRM platforms, and analytics platforms
- Not a great fit if you want to:
    - ✓ Run complex search queries
    - ✓ Perform multi-operation transactions
- Ex: MongoDB, DocumentDB, CouchDB, Cloudant

3. Column Based

- 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
- For example, a customer's name and profile information will most likely be accessed together but not their purchase history. So, customer's name and profile information data can be grouped into a column family
- Columns based:
    - ✓ 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 a great fit if you want to:
    - ✓ Run complex queries
    - ✓ Change querying patterns frequently

- Ex: Cassandra, HBase

4. 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, real-time product recommendations, network diagrams, fraud detection, and access management
- Not a great fit if you want to
    - ✓ Process high volumes of transactions, because graph databases are not optimized for large-volume analytics queries
- Ex: Neo4J & CosmosDB

NoSQL was created in response to the limitations of traditional relational database technology

Advantages of NoSQL

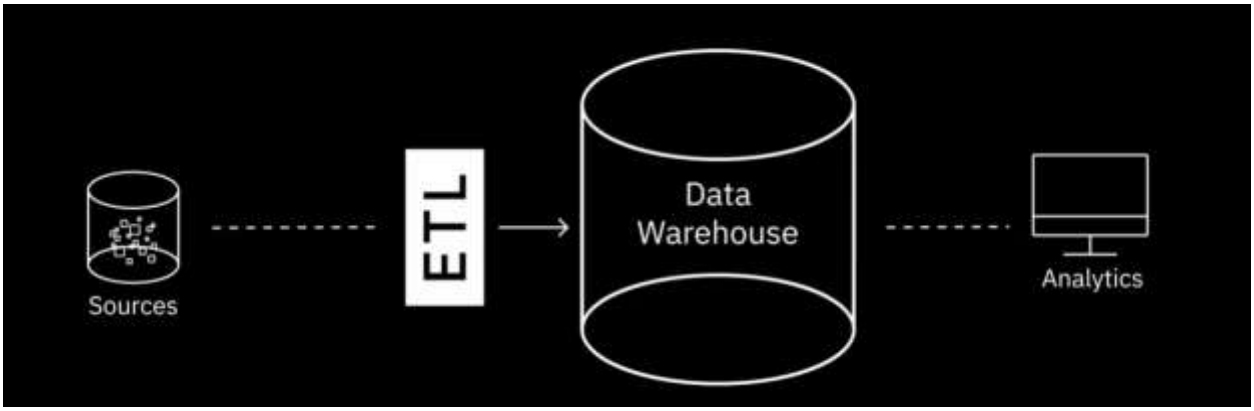
- Its ability to handle large volumes of structured, semi-structured, and unstructured data
- Its ability to run as a distributed system scaled across multiple data centers
- An efficient and cost-effective scale-out architecture that provides additional capacity and performance with the addition of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

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

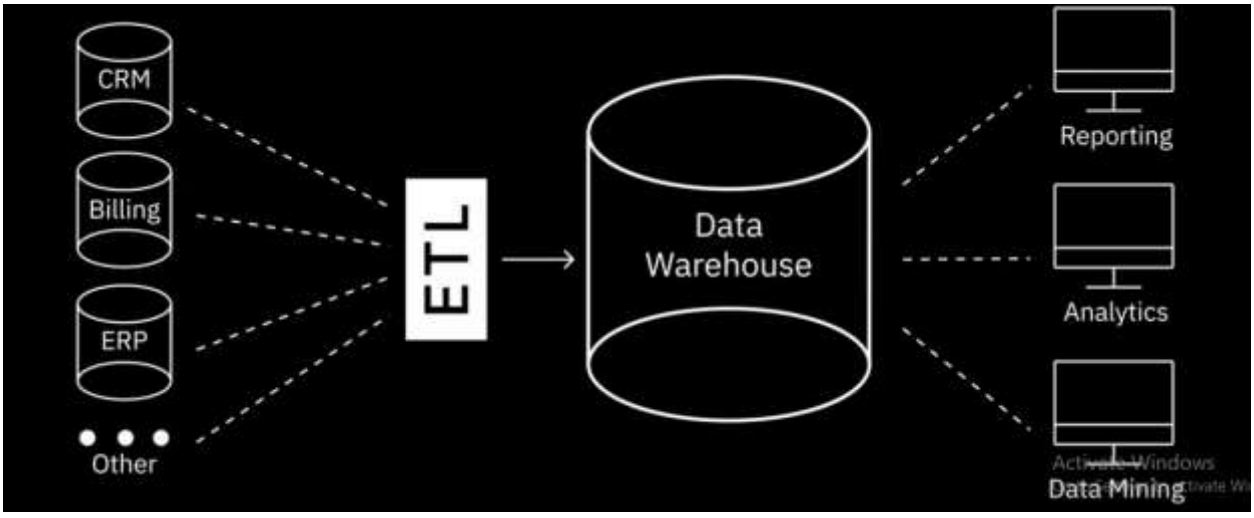
• A mature and well-documented technology, which means the risks are more or less perceivable	• A relatively newer technology

Data Marts, Data Lakes, ETL, and Data Pipelines

Data warehouse

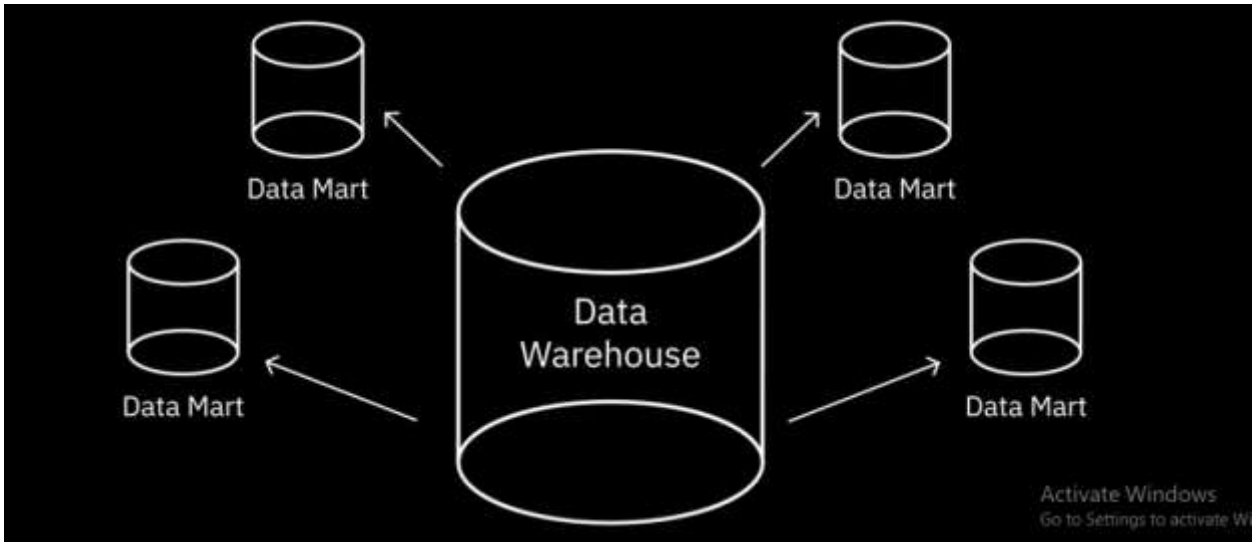


- A data warehouse works like a multi-purpose storage for different use cases. By the time the data comes into the warehouse, it has already been modeled and structured for a specific purpose, meaning it is analysis ready.
- As an organization, you would opt for a data warehouse when you have massive amounts of data from your operational systems that needs to be readily available for reporting and analysis

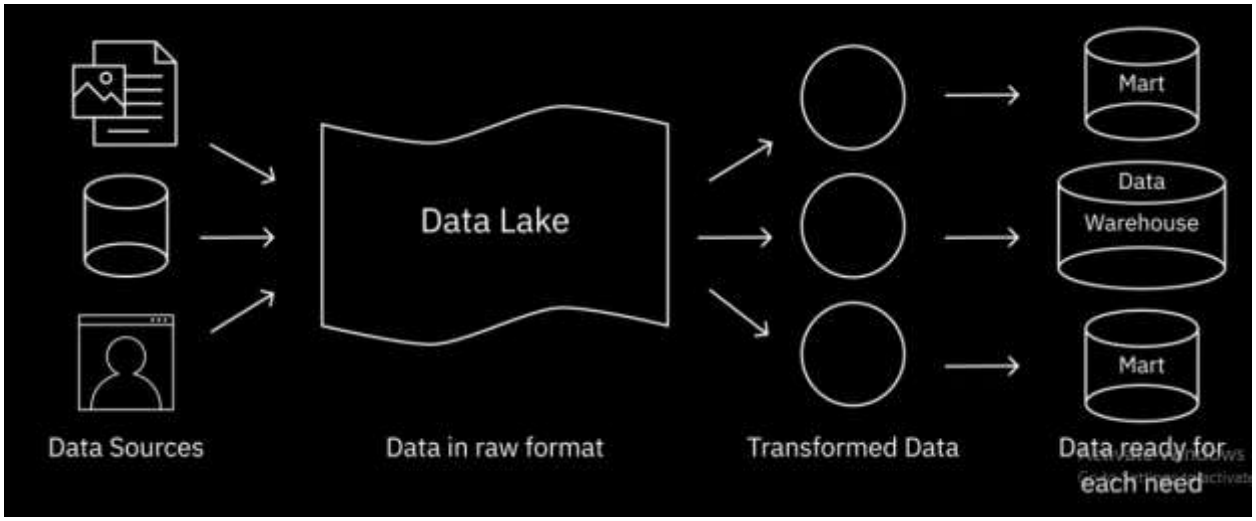


- Data warehouses serve as the single source of truth-storing current and historical data that has been cleansed, conformed, and categorized.
- A data warehouse is a multi-purpose enabler of operational and performance analytics

- A data mart is a sub-section of the data warehouse, built specifically for a particular business function, purpose, or community of users.
- The idea is to provide stakeholders data that is most relevant to them, when they need it



- For example, the sales or finance teams accessing, data for their quarterly reporting and projections
- Data mart:
  - ✓ Provide analytical capabilities for restricted area of the data warehouse
  - ✓ Offer isolated security and isolated performance
  - ✓ The most important role is business specific reporting and analytics
- A 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



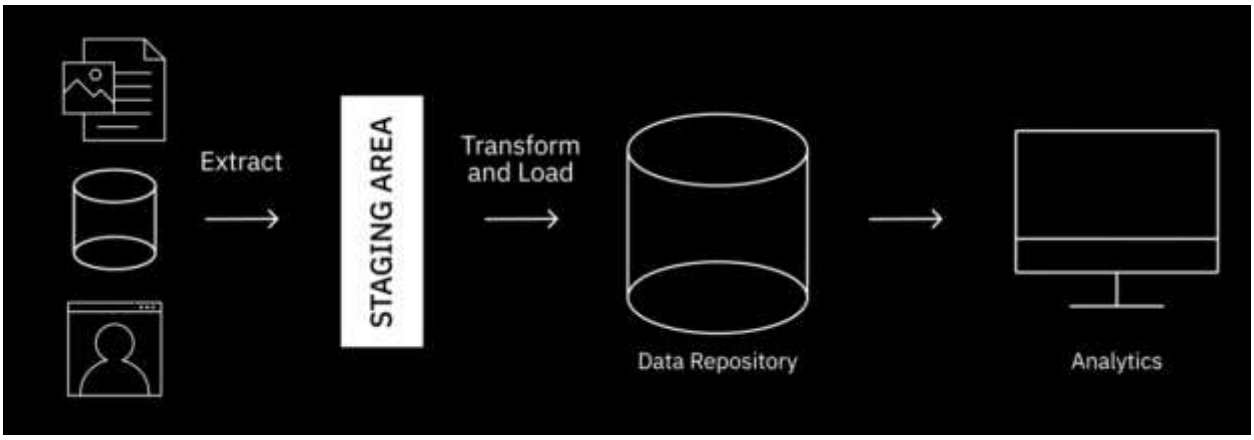
- A data lake is pool of raw data where each data element is given a unique identifier and is tagged with metatags for further use.
- Data from a data lake is selected and organized based on the use case you need it for.
- Unlike data warehouses, a data lake would retain all source data, without any exclusions. And the data could include all types of data sources and types.
- Data lakes are sometimes also used as a staging area of a data warehouse.
- Data lakes:
    - ✓ The most important role of a data lake is in predictive and advanced analytics

The process that is at the heart of gaining data value from data-the extract, transform, and load process, load process (ETL) is an automated process which includes:

- ETL is how raw data is converted into analysis-ready data
    - ✓ Gathering raw data
    - ✓ Extracting information needed for reporting and analysis
    - ✓ Cleaning standardizing, and transforming data into usable format
    - ✓ Loading data into a data repository
- While ETL is a generic process, the actual job can be very different in usage, utility, and complexity
- Extract is the step where data from source locations is collected for transformation
    - ✓ Batch processing-large chunks of data moved from source to destination at scheduled intervals. Tools: Stitch & Blendo
    - ✓ Stream processing – data pulled in real-time from source, transformed in transit, and loaded into data repository. Tools: Samza, Apache Storm, and Apache Kafka
- Transform involves the execution of rules and function that converts raw data into data that can be used for analysis. For Example:
    - ✓ Standardizing date formats and units of measurement
    - ✓ Removing duplicate data
    - ✓ Filtering out data that is not required
    - ✓ Enriching data
    - ✓ Establishing key relationships across tables
    - ✓ Applying business rules and data validation

- Transform & Load: Loading is the transportation of processed data in to a data repository. It can be:
    - ✓ Initial loading – population all of the data in the repository
    - ✓ Incremental loading – applying updates and modifications periodically
    - ✓ Full refresh – erasing a data table and reloading fresh data

    Load Verification includes checks for:
    - ✓ Missing or null values
    - ✓ Server performance
    - ✓ Load failures



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.

It's common to see the terms ETL and data pipelines used interchangeably. And although both move data from sources to destination.

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 visualizations tools.
- Tools: Beam, Data Flow, Kafka

Foundation of Big Data

- 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 and collect, host, and analytically process the vast amount of data gathered in order to derive real-time business insights that relate to consumers, risk, profit, performance, productivity management, and enhanced shareholder value.
- There are certain elements (Big Data), such as velocity, volume, variety, veracity, and value.
- Velocity: is the speed at which data accumulates. Data is being generated extremely fast in a process that never stops. Near or real-time streaming, local, and cloud-based technology can process information very quickly.
- Volume: is the scale of the data or increase in the amount of data stored. Drives of volume are the increase in data sources, higher resolution sensors, and scalable infrastructure.
- Variety: is the diversity of the data. Structured data fits neatly into rows and columns in relational databases, while unstructured data is not organized in a predefined way like tweets, blog posts, pictures, numbers, and video. Variety also reflects that data comes from different sources: machines, people, and processes, both internal and external to organization. Drivers are mobile technologies social media, wearable technologies, geo technologies video, and many, many more.
- Veracity: is the quality and origin of data and its conformity to facts and accuracy. Attributes include consistency, completeness, integrity and ambiguity. Drivers include cost and the need for traceability. With the large amount of data available, the debate rages on about the accuracy of data in the digital age. Is the information real or is it false?
- Value: is our ability and need to turn data into value. Value isn't just profit. It may have medical or social benefits, as well as customer, employee or personal satisfaction. The main reason that people invest time to understand big data is to derive value from it.

Let's look at some examples of the V's in action.

- Velocity: every 60 seconds, hours of footage are uploaded to YouTube, which is generating data. Think about how quickly data accumulates over hours, days, and years.
- Volume: the world population is approximately 7 billion people and the vast majority are now using digital devices. Mobile phones, desktop and laptop computers, wearable

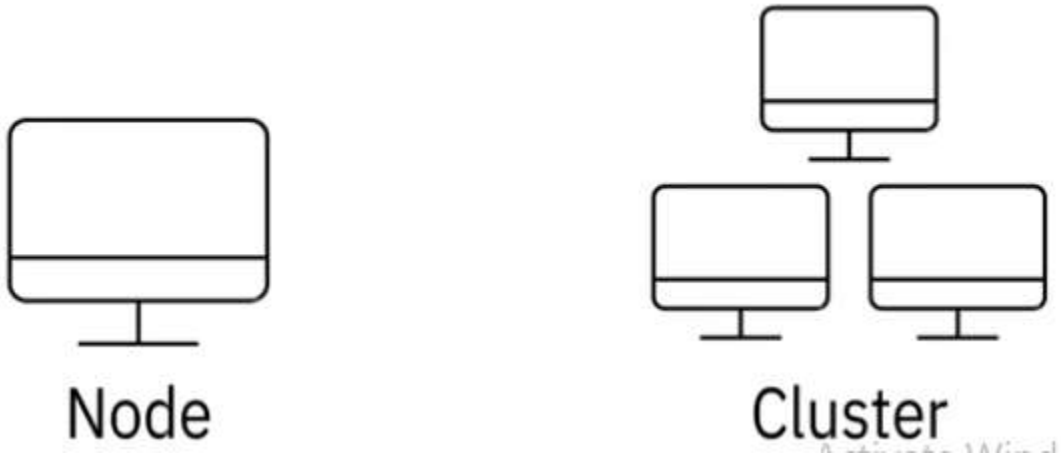
devices, and so on. These devices all generate, capture, and store data approximately 2.5 quintillion bytes every day. That's the equivalent of 10 million blue-ray DVDs.

- Variety: let's think about the different types of data. Text, pictures, film, sound, health data from wearable devices, and many different types of data from devices connected to the internet of things.
- Veracity: eighty percent of data is considered to be unstructured and we must devise ways to produce reliable and accurate insights. The data must be categorized, analyzed, and visualized.
- Data scientist today, derive insights from big data and cope with the challenges that these massive data sets present. The scale of the data being collected means that it's not feasible to use conventional data analysis tools, however, alternative tools that leverages, distributed computing power can overcome this problem. Tools such as Apache Spark, Hadoop, and its Ecosystem. Provides ways to extract, load, analyze, and process the data across distributed compute resources, providing new insights and knowledge. This gives organizations more ways to connect with their customers and enrich the services they offer. So next time you strap on your smartwatch, unlock your smartphone, or track your workout, remember your data is starting a journey that might take it all the way around the world, through big data analysis and back to you.

Big Data Processing Tools

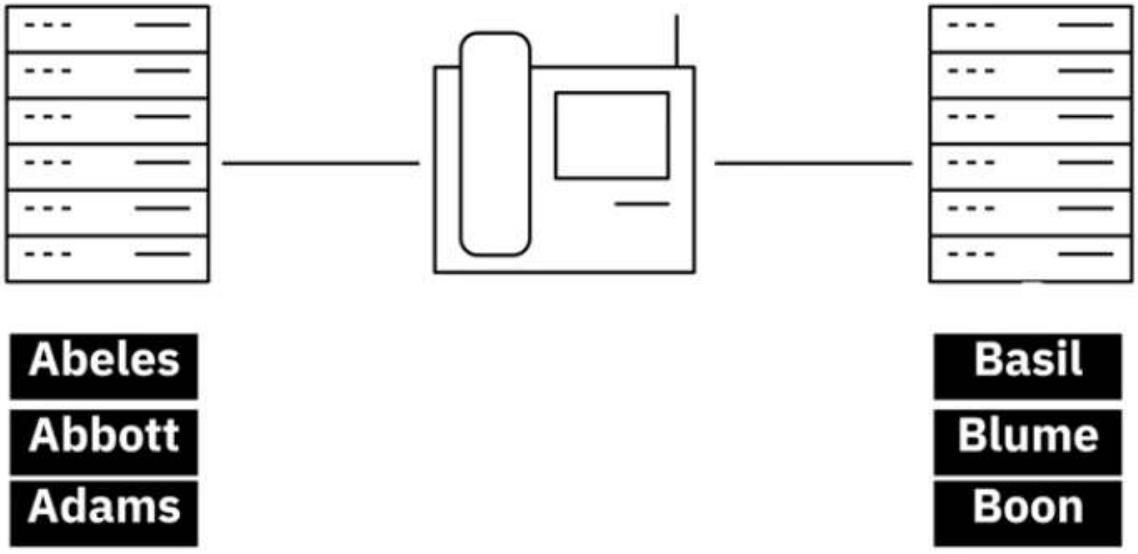
- The Big Data processing technologies provide ways to work with large sets of structured, semi-structured, and unstructured data so that value can be derived from big data.
- Big Data technologies such as NoSQL databases and Data Lakes.
- There are three open source technologies and the role they play in big data analytics – Apache Hadoop, Apache Hive, and Apache Spark.
- Apache Hadoop: a collection of tools that provides distributed storage and processing of Big Data.
- Apache Hive: a data warehouse for data query and analysis built on top of Hadoop.
- Apache Spark: a distributed analytics framework for complex, real-time data analytics in real time.
-

- Hadoop, a java-based open-source framework, allows distributed storage and processing of large datasets across clusters of computers.
- In Hadoop distributed system, a node is a single computer, and a collection of nodes forms a cluster.

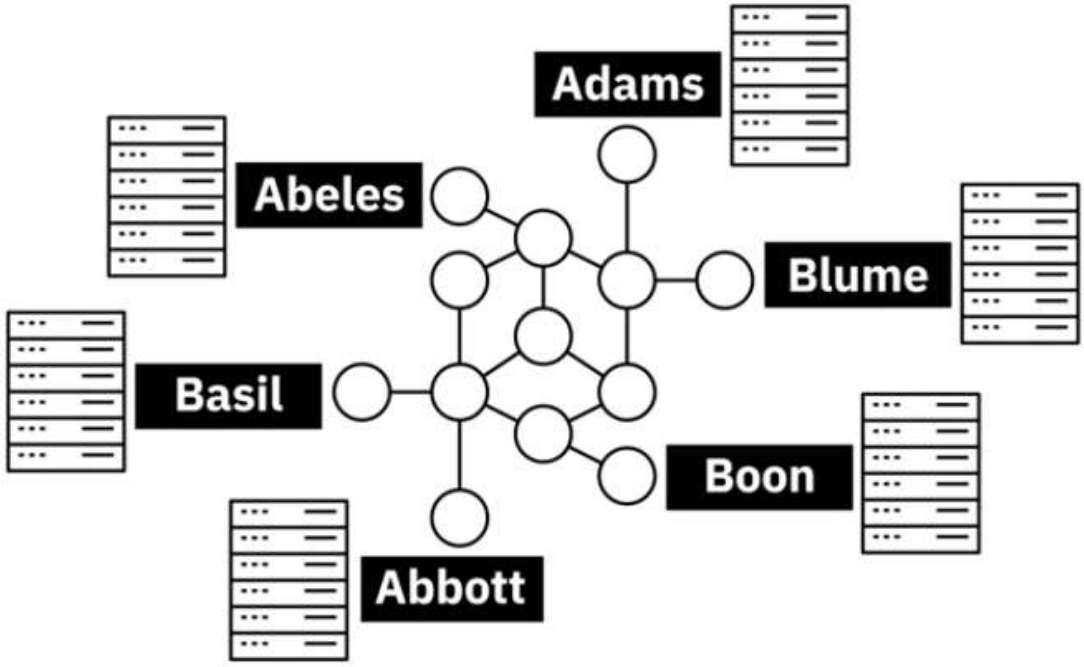


- Hadoop can scale up from single node to any number of nodes, each offering local storage and computation.
- Hadoop provides a reliable, scalable, and cost-effective solution for storing data with no format requirements.
- Using Hadoop, you can: incorporate emerging data formats, such as streaming audio, video, social media sentiment, and clickstream data, along with structured, semi-structured, and unstructured data not traditionally used in a data warehouse. Provide real-time, self-services access for all stakeholders. Optimize and streamline costs in your enterprise data warehouse by consolidating data across the organization and moving “cold” data that is, data that is not in frequent use, to a Hadoop based system.
- One of the four main components of Hadoop is Hadoop Distributed File System, or HDFS, which is the storage system for big data that runs on multiple commodity hardware connected through a network
- HDFS provides scalable and reliable big data storage by partitioning files over multiple nodes. It splits large files across multiple computers, allowing parallel access to them. Computation can, therefore, run in parallel on each node where data is stored.
- Replicates file blocks on different nodes to prevent data loss, making it fault tolerant. Example: consider a file that includes phone numbers for everyone in the United States:

the numbers for people with last name starting with A might be stored on server 1, B on server 2, and so on.

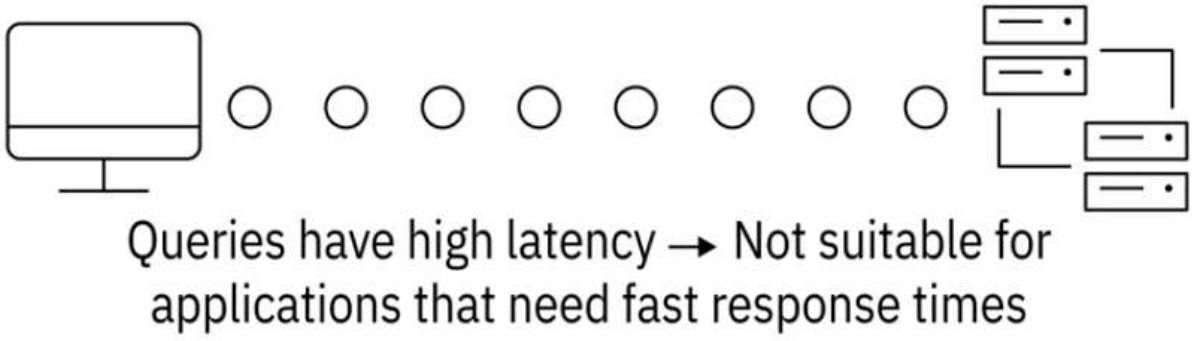


- With Hadoop, pieces of this phonebook would be stored across the cluster. To reconstruct the entire phonebook, your program would need the block from every server in the cluster.



- HDFS also replicates these smaller pieces onto two additional servers by default, ensuring availability when a server fails. In addition to higher availability, this offers multiple benefits. It allows to Hadoop cluster to break up work into smaller chunks and run those jobs on all servers in the cluster for better scalability. Finally, you gain the benefit of data locality, which is the process of the moving the computation closer to the node which the data resides. This is critical when working with large data sets because it minimizes network congestion and increases throughput.
- Some of the other benefits that come from using HDFS include:
    - Fast recovery from hardware failures, because HDFS is built to detect faults and automatically recover.
    - Access to streaming data, because HDFS supports high data throughput rates.
    - Accommodation of large data sets, because HDFS can scale to hundreds of nodes, or computers, in a single cluster.
    - Portability, because HDFS is portable across multiple hardware platforms and compatible with a variety of underlying operating systems.

- Hives is an open-sources 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.



  - Also, Hives is read-based, and therefore not suitable for transaction processing that typically involves a high percentage of write operations
  - Hives is better suited for data warehousing tasks such as ETL, reporting, and data analysis and includes tools that enable easy access to data via SQL.
- Spark 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
  - Process streaming data fast
  - Performs complex analytics in real-time

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 processix`ng tools and platforms designed specifically for Big Data, such as Apache Hadoop, Apache Hive, and Apache Spark.