FAKULTA ELEKTROTECHNICKÁ

České vysoké učení technické v Praze

B4M36DS2 – Database Systems 2

MSc. **Yuliia Prokop**, Ph.D.

prokoyul@fel.cvut.cz

Telegram **@Yulia_Prokop**

ČVUT FEL CourseWare Wiki **https://cw.fel.cvut.cz/b241/courses/b4m36ds2/start**

Basic course information

Lectures: Monday, 9:15 – 10:45

Practical classes: Monday, 12:45 – 14:15, 14:30 – 16:00, 16:15 – 17:45

Homework – maximum 32 points

Course credit – minimum 20 points

Exam – maximum 70 points

- **written exam** (mandatory) + **oral exam** (optional)

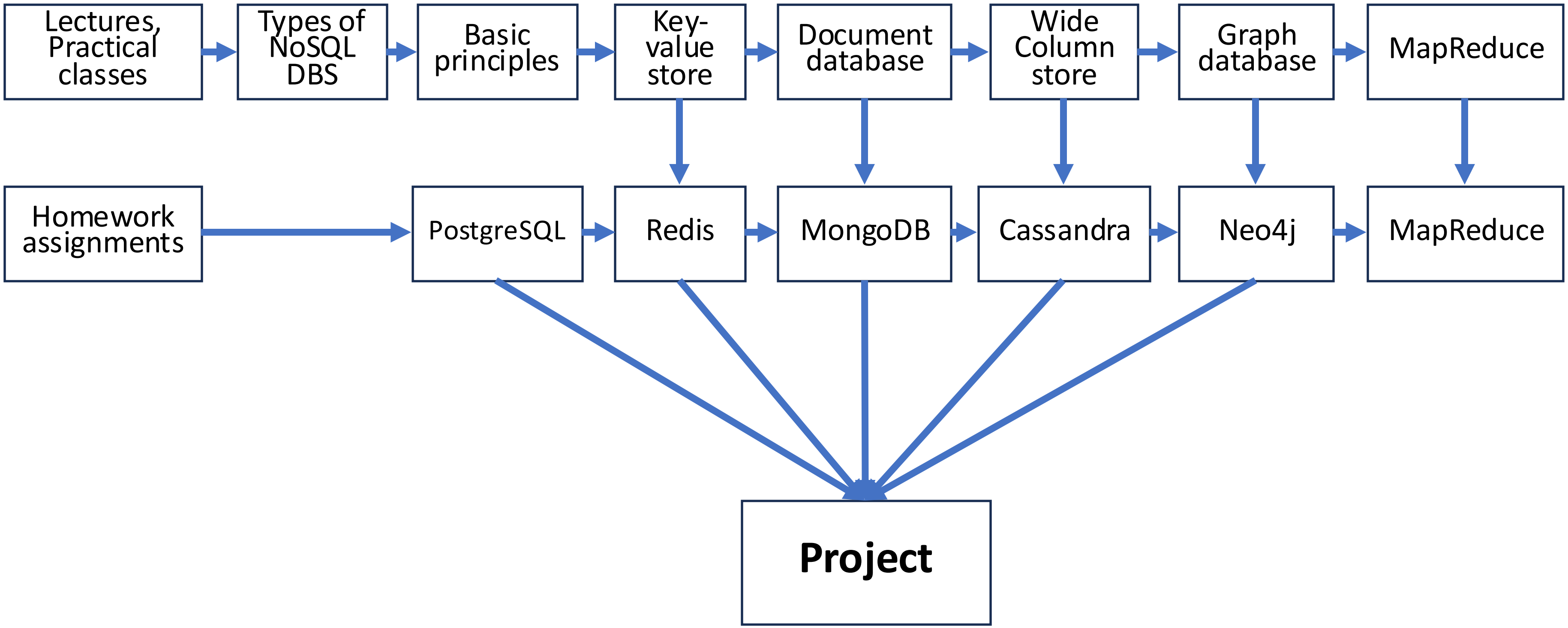
CourseWare Wiki **– course materials**

BRUTE – upload reports on the homework

NoSQL Server – submit and execute homework

Consultation – email me

Basic course information



FAKULTA ELEKTROTECHNICKÁ

České vysoké učení technické v Praze

B4M36DS2 – Database Systems 2

Lecture 1 - Introduction: Big Data, RDBS vs NoSQL DBS

23. 9. 2024

Yuliia Prokop

prokoyul@fel.cvut.cz, Telegram **@Yulia_Prokop**

Based on **Martin Svoboda**'s materials (https://www.ksi.mff.cuni.cz/~svoboda/courses/211-B4M36DS2/)

ČVUT FEL CourseWare Wiki **https://cw.fel.cvut.cz/b241/courses/b4m36ds2/start**

Lecture Outline

- ✓ History of database models
- ✓ DBMS ranking 2024
- ✓ Big Data and its characteristics
- ✓ Relational DBS features
- ✓ NoSQL DBS features

Historical trends of Database Management System

A database management system (DBMS) allows a person to organize, store, and retrieve data from a computer.

2008 NoSQL, Big Data

2000s Relational database model

1990s Object database model

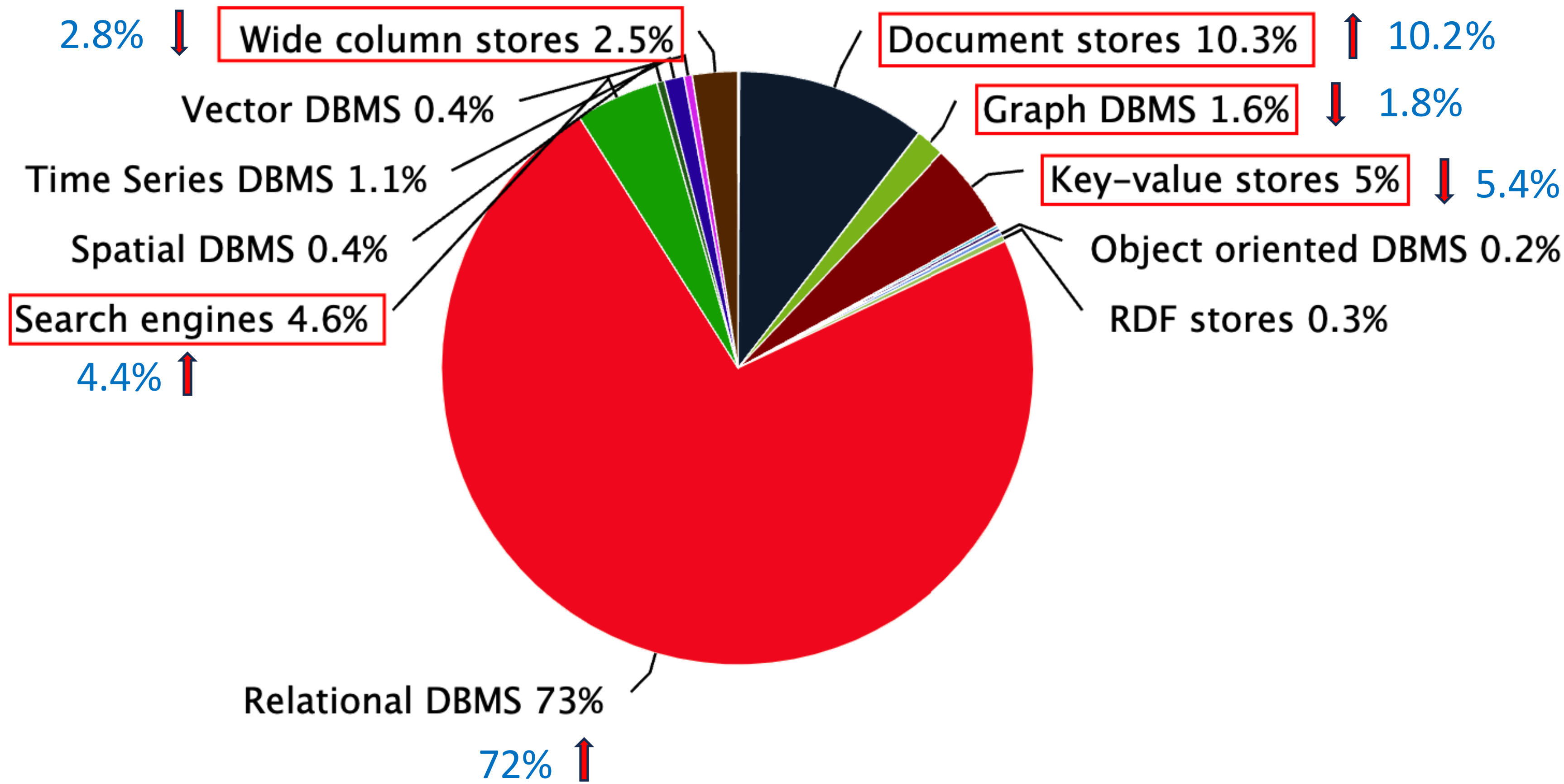
1980s Structured Query Language (SQL)

1970s Relational database model

1960s Network and hierarchical models

Database ranking, August 2024

DB-Engines Ranking



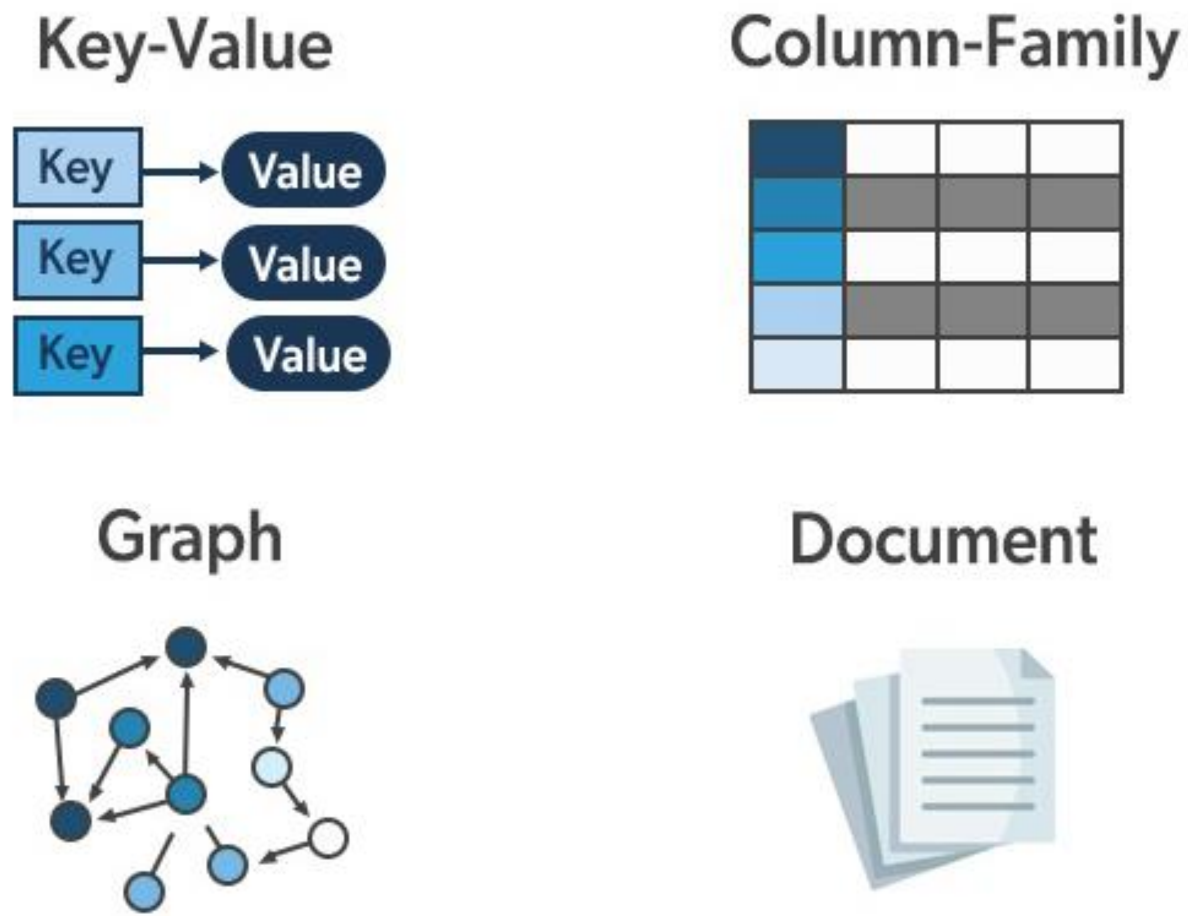
Source: https://db-engines.com/en/ranking_categories

Top Database Management Systems In August, 2024

Database Management System	Rank Now	Jul 2024	
Oracle	1	1	< 0
MySQL	2	2	< 0
MongoDB	3	8	^ 5
Cassandra	4	7	^ 3
Microsoft SQL Server	5	6	^ 1
PostgreSQL	6	8	^ 2
Redis	7	11	^ 4
IBM Db2	8	10	^ 2
SQLite	9	11	^ 2
MariaDB	10	11	^ 1
Elasticsearch	11	12	^ 1
Neo4j	17	20	^ 3

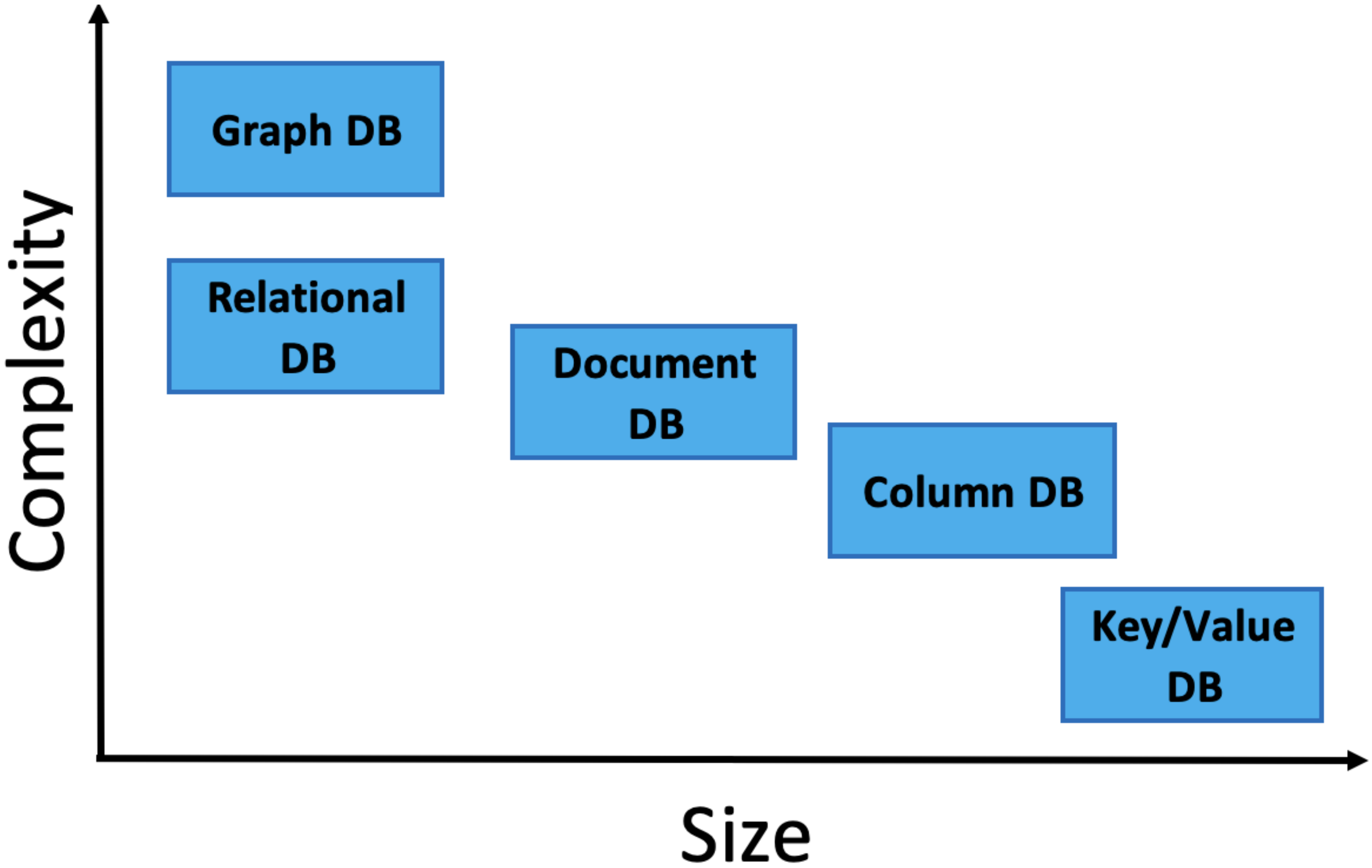
Source: https://red9.com/database-popularity-ranking/

Types of NoSQL Databases



Source: https://www.geeksforgeeks.org/types-of-nosql-databases/

Size / Complexity of data stores



NoSQL DBS in the course

- **Document** stores (**MongoDB**)
- **Key-value** stores (**Redis**)
- **Wide column** stores (**Cassandra**)
- **Grapf** DBMS (**Neo4j**)
- **Search** engines (**Elasticsearch**)
- Hybrid systems (**HADOOP, Mapreduce**)

Big Data

What is Big Data?

Big Data primarily refers to data sets that are **too large** or complex to handle by traditional data-processing application software. It is characterized by the three Vs: volume, variety, and velocity

Where is Big Data?

- **Social media and networks**
  …all of us are generating data
- **Scientific instruments**
  …collecting all sorts of data
- **Mobile devices**
  …tracking all objects all the time
- **Sensor technology and networks**
  …measuring all kinds of data

Application of Big Data

- **Business and Marketing**
  - Companies use Big Data to analyze consumer behavior, personalize offerings, optimize supply chains, and predict demand.
- **Healthcare**
  - Doctors and researchers analyze medical data to improve diagnostics, personalize treatments, and predict the spread of epidemics.
- **Financial Sector**
  - Big Data helps banks and insurance companies assess credit risks, detect fraud, and develop new financial products.
- **Social Media**
  - Analyzing user activity on social media platforms helps companies understand trends, conduct targeted advertising campaigns, and gauge public opinion.
- **Science and Research**
  - Big Data is used in genetics, climatology, astronomy, and other scientific fields to analyze massive amounts of information and identify patterns.

Data generated per minute on the Internet

97 million

15 million

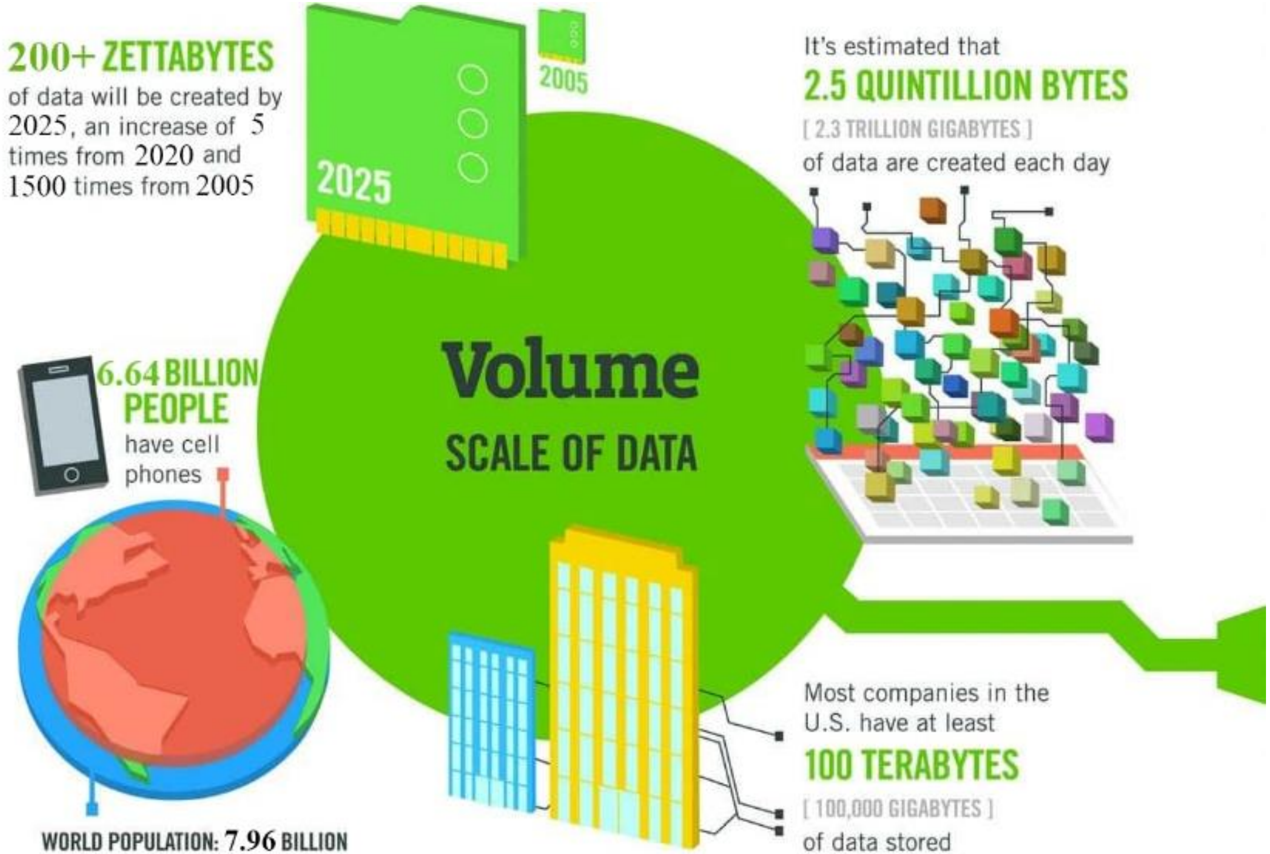
2.52 million

694,000 hours of video content

188 million

6.3 million

Volume refers to the scale of the data



Source: http://www.ibmbigdatahub.com/; https://www.bankmycell.com; https://techjury.net

Key Big Data Statistics

- **Google receives more than 9 billion searches every day.**
  - **3.3 trillion** searches are conducted annually, and over **105,000** search queries are made every second
  - **100 billion messages are exchanged on WhatsApp every day.**
- **Facebook has around 2.9 billion active users monthly**
  - Facebook has over 1.8 billion daily users, and this data includes the Facebook app, Instagram app, Messenger app, and WhatsApp.
- **Internet users generate 2.5 quintillion bytes of data each day.**
  - In 2020, each internet user generated 1.7MB of data per second.

Source: https://earthweb.com/big-data-statistics/

Key Big Data Statistics 2024

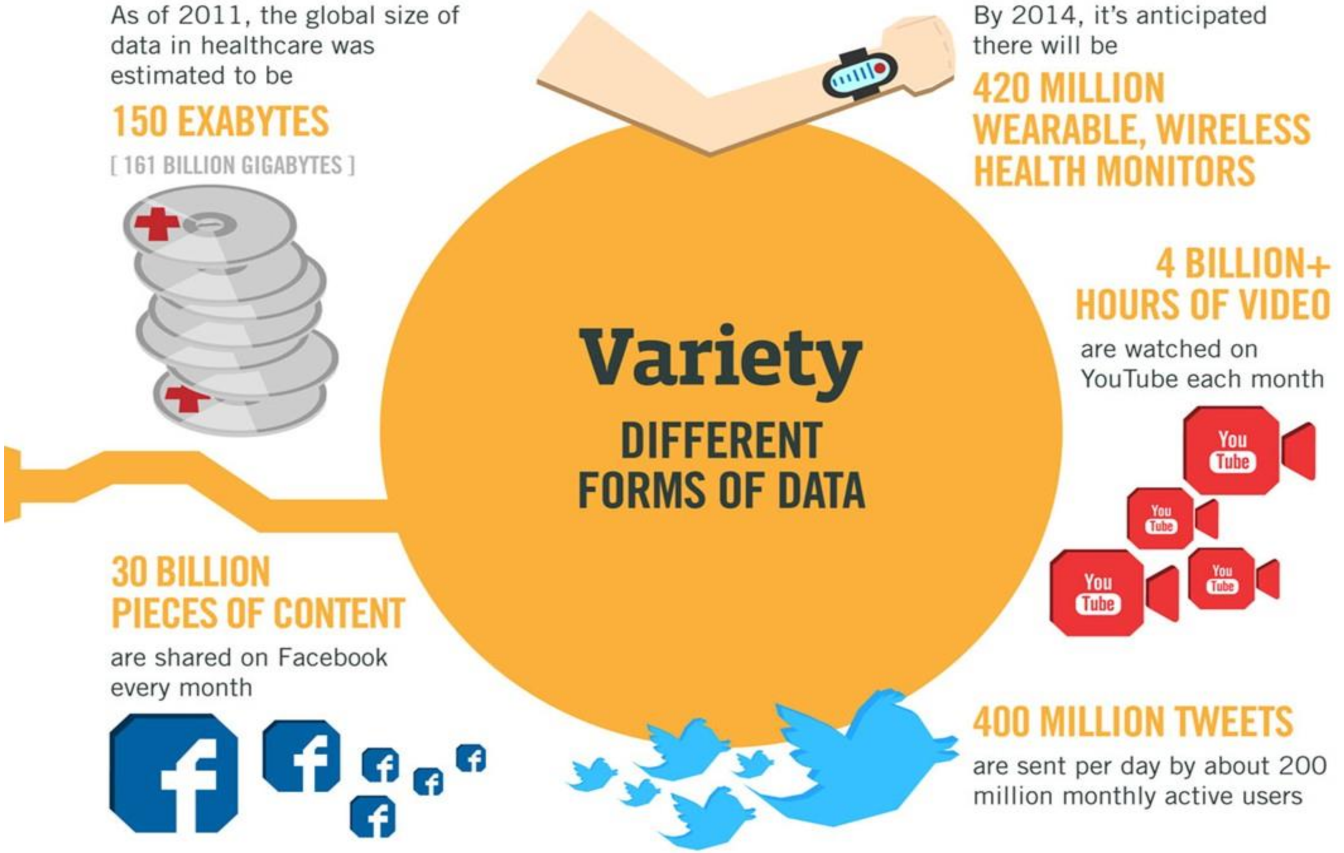
- **All the data available on the Internet can be downloaded in 181 million years.**
- **With each second, more and more IoT devices begin to connect.**
  - These devices produce around 5 quintillion bytes if dated daily.
  - This date can amount to up to 79.4 ZB by 2025.

- **Over 80 % of data generated today is unstructured.**
- **Over 95 % of businesses think that managing unstructured data is their primary problem.**

Source: https://earthweb.com/big-data-statistics/

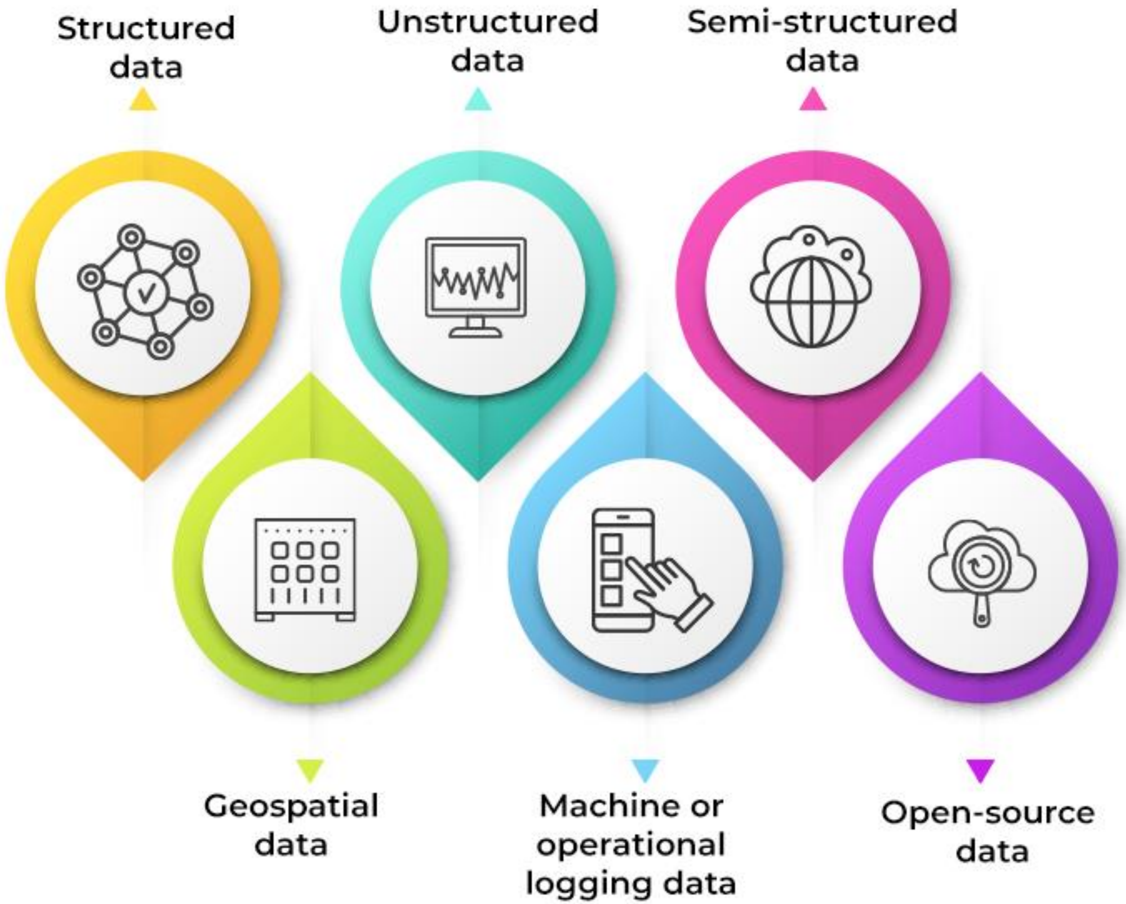
Big Data Characteristics : Variety

Variety refers to the different formats that data comes in



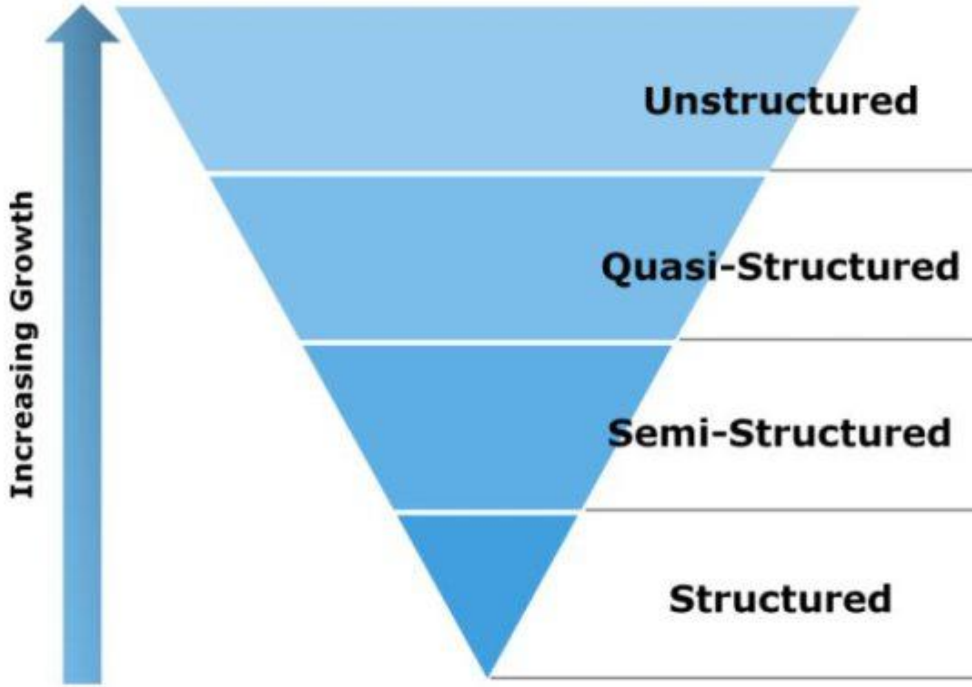
Source: http://www.ibmbigdatahub.com/

Types of Big Data



Source: https://www.spiceworks.com/tech/big-data/articles/what-is-big-data/

Types of Data



- Data that has no inherent structure and is usually stored as different types of files.
- E.g. Text documents, PDFs, images, and videos

- Textual data with erratic formats that can be formatted with effort and software tools
- E.g. Clickstream data

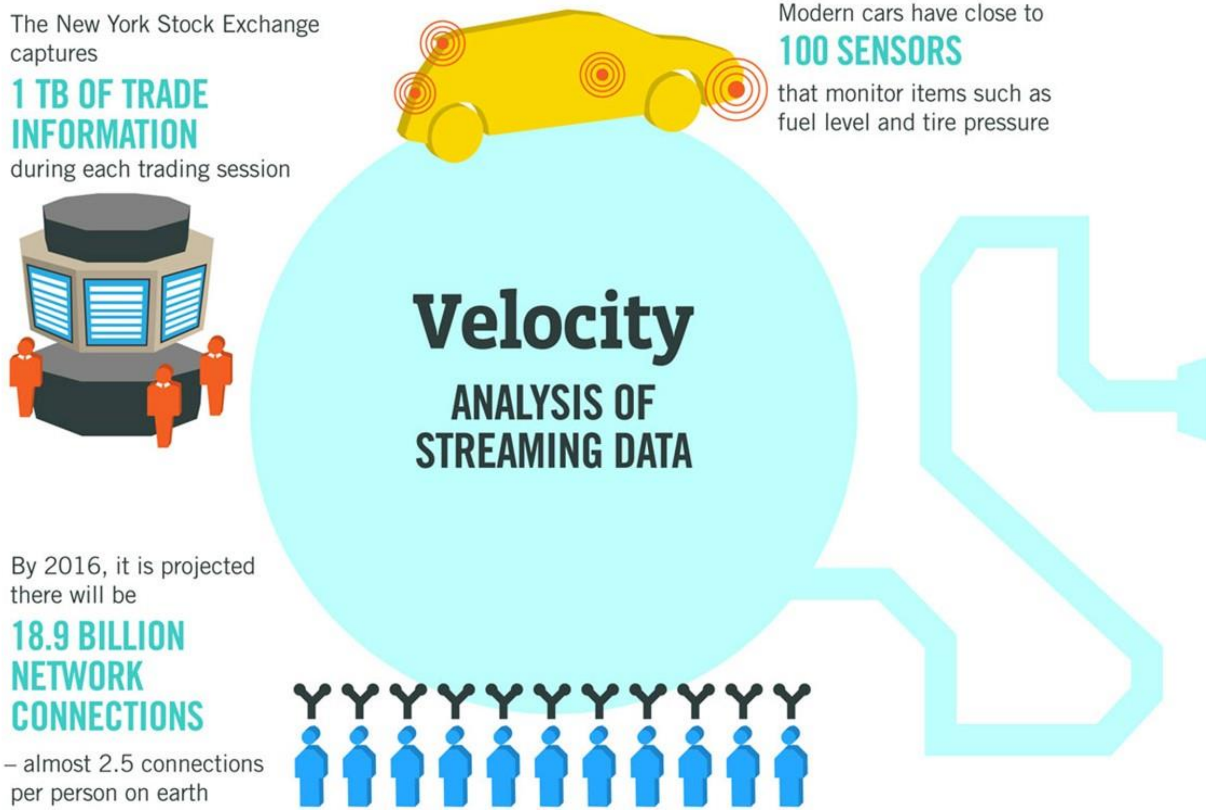
- Textual data files with an apparent pattern, enabling analysis
- E.g. Spreadsheets and XML files

- Data having a defined data model, format, structure
- E.g. Database

Source: https://www.mycloudwiki.com/san/data-and-information-basics/

Big Data Characteristics : Velocity (Speed)

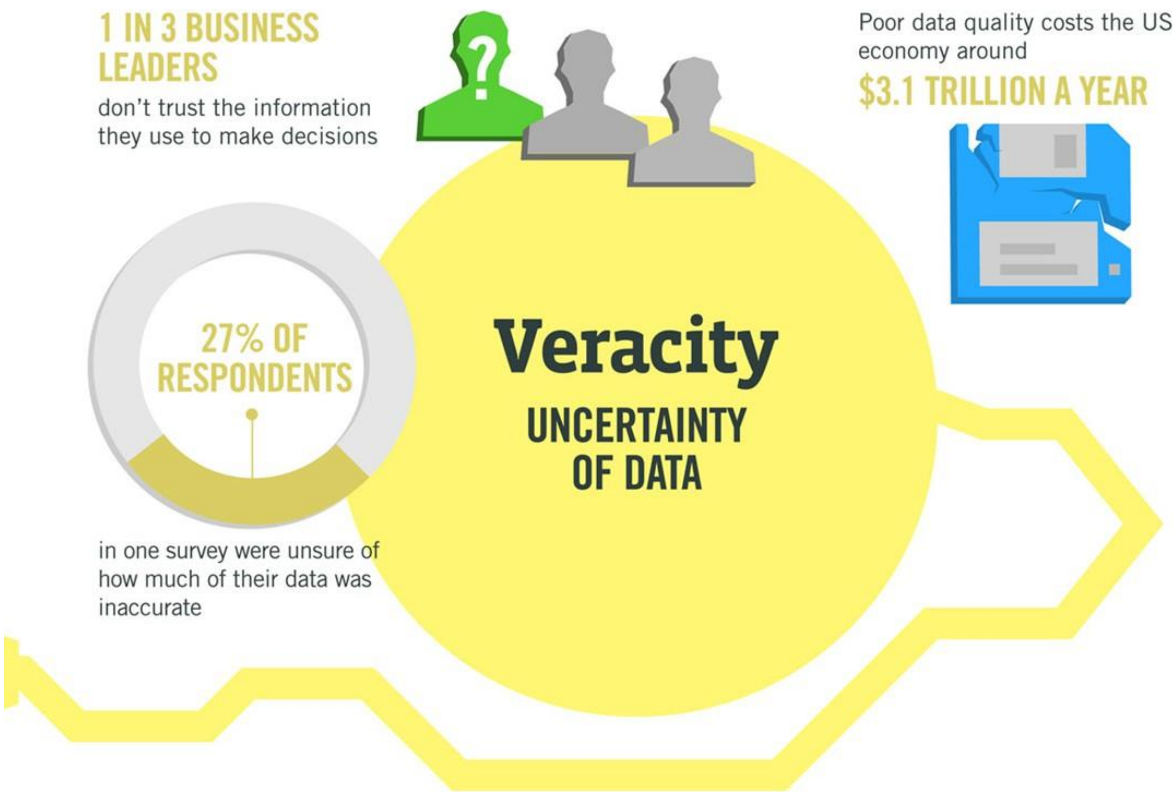
Velocity refers to the speed at which large datasets are acquired, processed, and accessed



Source: http://www.ibmbigdatahub.com/

Big Data Characteristics : Veracity (Uncertainty)

Veracity refers to the quality and accuracy of data. Big data can be noisy and uncertain, full of biases, abnormalities, and imprecision. Low veracity can greatly damage the accuracy of your results.



Source: http://www.ibmbigdatahub.com/

Big Data Characteristics : Additional V

- **Value**

  The business value of the data (needs to be revealed)

- **Validity**

  Data correctness and accuracy with respect to the intended use

- **Volatility**

  Period of time the data is valid and should be maintained

Big Data Characteristics : Additional V

Source: https://www.xenonstack.com/blog/big-data-engineering/ingestion-processing-big-data-iot-stream/

Big Data Characteristics : Three C

- **Cardinality**
  - the number of records in the dynamically growing dataset at a particular instance
- **Continuity**
  - two characteristics and they are: (i) representation of data by continuous functions, and (ii) continuous growth of data size with respect to time
- **Complexity**
  - three characteristics and they are: (i) large varieties of data types, (ii) high dimensional dataset; and (iii) the speed of data processing is very high

Structured Data

- Types of data: **structured**, **unstructured** and **semi-structured**

- **Structured** data can be stored, accessed, and processed in a fixed format.

Employee_ID	Employee_Name	Gender	Department	Salary_In_lacs
2365	Rajesh Kulkarni	Male	Finance	650000
3398	Pratibha Joshi	Female	Admin	650000
7465	Shushil Roy	Male	Admin	500000
7500	Shubhojit Das	Male	Finance	500000
7699	Priya Sane	Female	Finance	550000

Semi-structured Data (no or little schema)


```
[
 {
  "Employee_ID": 2365,
  "Employee_Name": "Jiří Novák",
  "Department": "Finance",
  "Phone": "666555444",
  "Address": {
      "Street": "Václavské náměstí 123",
      "City": "Praha",
  },
  "Skills": [
   "Účetnictví", "Finanční analýza", "Rozpočtování"
  ]
 },
 {
  "Employee_ID": 3398,
  "Employee_Name": "Kateřina Svobodová",
  "Department": "Admin",
  "Projects": [
    {
     "Name": "Renovace kanceláří",
     "Duration": "6 měsíců"
    },
    {
     "Name": "Aktualizace HR systému",
     "Duration": "3 měsíce"
    }
  ]
 }
]
```

Challenges with Semi-structured Data

- **Parsing**

  Although semi-structured data is more organized than unstructured data, it still requires parsing and transformation before it can be effectively used in analysis or querying.

- **Integration**

  Integrating semi-structured data with structured data systems can be challenging, often requiring data transformation processes.

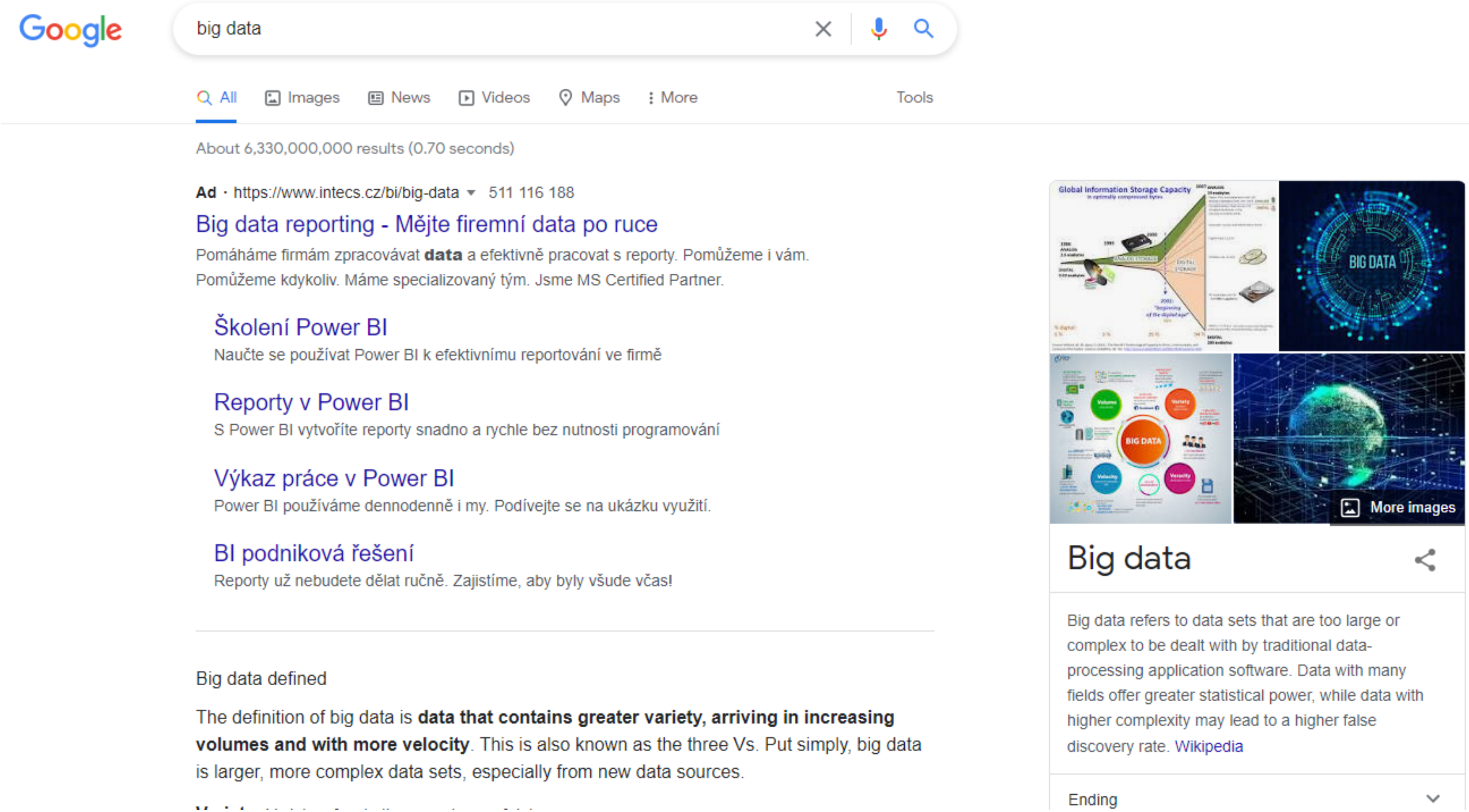
- **Storage**

  While semi-structured data can be stored in NoSQL databases designed to handle flexible schema, it still needs careful organization to ensure efficient access and use.

- **Querying**

  Semi-structured data often requires specialized querying languages, such as XPath for XML or SQL with JSON functions, making it more complex than querying structured data.

- Any data with unknown form or structure is classified as unstructured data.



Examples of Unstructured Data

- **Text Documents**

  Emails, word documents, and PDFs are typical examples. They contain valuable information but lack a structured format that can be easily queried.

- **Social Media Content**

  Posts, tweets, comments, and blog articles often contain insights into public sentiment, but the data is unstructured.

- **Multimedia Files**

  Photos, videos, and audio recordings are complex data types requiring specialized processing and analysis tools.

- **Web Pages**

  Web pages contain a mix of text, images, and other elements that don’t conform to a strict structure.

- **Sensor Data**

  Data from IoT devices that might generate logs in varying formats, which are not uniform or consistent.

Challenges with Unstructured Data

- **Storage**

  Unstructured data requires flexible storage solutions like NoSQL databases, data lakes, or cloud storage systems.

- **Processing**

  Traditional SQL-based tools are ineffective for querying or analyzing unstructured data, necessitating advanced techniques like natural language processing (NLP), machine learning, and AI.

- **Search and Retrieval**

  Extracting relevant information from unstructured data requires specialized algorithms capable of understanding the content's context and meaning.

- **Analysis**

  Analyzing unstructured data can be resource-intensive and requires specific expertise to interpret and extract valuable insights.

Tools and Technologies for Big Data

- **Hadoop**
  - A platform for distributed storage and processing of Big Data. It includes a distributed file system (**HDFS**) and a framework for parallel data processing (**MapReduce**).
- **Apache Spark**
  - A framework for fast data processing in real-time. It offers high performance by processing data in memory.
- **NoSQL Databases**
  - Databases like **Cassandra**, **MongoDB**, and **HBase** excel at storing unstructured and semi-structured data.
- **Data Lakes**
  - These repositories store data in raw form without prior processing, making them ideal for storing large volumes of diverse data for future analysis.
- **Visualization Tools**
  - Tools like Tableau, Power BI, and QlikView enable the creation of visualizations and reports based on Big Data analysis.

Comparison between traditional data and big data

	Traditional data	Big data
Volume	In GBs	TBs and PBs
Data generation rate	Per hour; per day	More rapid
Data structure	Structured	Semi-structured or Unstructured
Data source	Centralized	Fully distributed
Data integration	Easy	Difficult
Data store	RDBMS	HDFS, NoSQL
Data access	Interactive	Batch or near real-time

Source: Furht, Borko, and Flavio Villanustre. "Introduction to big data."

Relational databases : data model

✓ **Relational model**: Structured data is stored in **tables** with **rows** and **columns**

- Each row represents a record with a **unique key**
- Columns hold attributes of data.

Students

ID	Name	Phone	DateOfBirth	Sex
500	Alexander	666-555-444	06.03.2000	M
501	Roman	777-666-555	23.08.1999	M
502	Tereza	555-666-777	14.05.2000	F

All data must follow this schema.

Relational databases : relationships

- ✓ Relational databases allow you to define **relationships** between different data sets.
- ✓ **Foreign keys** are used to define the relationships among the tables.

Students

ID	Name	Phone	DateOfBirth	Sex
500	Alexander	666-555-444	06.03.2000	M
501	Roman	777-666-555	23.08.1999	M
502	Tereza	555-666-777	14.05.2000	F

ID	CourseID	Grade
500	1001	B
501	1002	A
501	1003	B
502	1002	C

Takes_Course

CourseID	Title	Credits
1001	Data Mining	5
1002	Artificial Intelligence	6
1003	Database systems	5

Courses

Relational databases : SQL

Relational databases use **Structured Query Language (SQL)** as the standard interface for querying and manipulating data.

SELECT id, name, price **FROM** products

Representatives

- Oracle Database, Microsoft SQL Server, IBM DB2
- MySQL, PostgreSQL

Relational databases provide powerful tools for querying and analysing data, which can be used to generate reports, discover trends, and make informed decisions.

Selection is based on complex conditions, **projection**, **joins**, **aggregation**, derivation of new values, recursive queries, …

Relational databases : Normal forms

Model

- Functional dependencies
- 1NF, 2NF, 3NF, BCNF (Boyce-Codd normal form)

Objective

- **Normalization of database schema** to BCNF or 3NF
- Algorithms: decomposition or synthesis

Motivation

- Diminish **data redundancy**, prevent update anomalies
- However:

  Data is scattered into small pieces (high granularity), and so these pieces have to be joined back together when querying!

Relational databases : Transactions

- **Transaction** = flat sequence of database operations (READ, WRITE, COMMIT, ABORT)

Objectives

- Enforcement of ACID properties
- **Efficient parallel / concurrent execution** (slow hard drives, …)

ACID properties

- Atomicity – partial execution is not allowed (all or nothing)
- Consistency – transactions turn one valid database state into another
- Isolation – uncommitted effects are concealed among transactions
- Durability – effects of committed transactions are permanent

Relational databases : Limitations

- Handling large volumes of **unstructured data**
  - Relational databases struggle significantly with unstructured or semi-structured data.
- **Scalability** challenges
  - Relational databases often face difficulties when scaling horizontally across multiple servers. This becomes a significant issue for applications that handle massive data or traffic.
- **Schema flexibility**
  - Relational databases require predefined schemas. It can be problematic in case of rapidly changing data requirements or when the nature of the data isn't fully known in advance.
- **High-velocity data**
  - Relational databases may struggle to keep up with the incoming data rate in extremely high-speed data ingestion scenarios.

Current Trends

Big Data

- **Volume**: terabytes → zettabytes
- **Variety**: structured → semi-structured and unstructured data
- **Velocity**: batch processing → streaming data

Big users

- Population online, hours spent online, devices online, …
- Rapidly growing companies / web applications
    - Even millions of users within a few months

Current Trends

Everything is in the **cloud**

- **SaaS:** Software as a Service
- **PaaS:** Platform as a Service
- **IaaS:** Infrastructure as a Service

Processing paradigms

- **OLTP:** Online Transaction Processing
- **OLAP:** Online Analytical Processing
- *…but also…*
- **RTAP:** **Real-Time Analytical Processing**

Current Trends

Data assumptions

- **Data format** is becoming unknown or inconsistent
- Linear growth → **unpredictable exponential growth**
- **Read requests** often prevail **write requests**
- Data updates are no longer frequent
- Data is expected to be replaced
- Strong **consistency** is no longer mission-critical

Current Trends

⇒ **New approach is required**

- Relational databases simply do not follow the current trends

Key technologies

- Distributed **file systems**
- **MapReduce** and other programming models
- Grid computing, cloud computing
- **NoSQL databases**
- Data warehouses
- Large scale machine learning

NoSQL Databases

What does **NoSQL** actually mean?

- Not: *no to SQL*
- Not: *not only SQL*
- NoSQL is an **accidental term with no precise definition**

What does **NoSQL** actually mean?

NoSQL movement = The whole point of **seeking alternatives** is that you need to solve a problem that **relational databases are a bad fit for**

NoSQL databases = Next generation databases mostly addressing some of the points: being

- ✓ **non-relational**,
- ✓ **distributed**,
- ✓ **open-source,**
- ✓ **horizontally scalable**.

The original intention has been modern web-scale databases. Often more characteristics apply as: **schema-free**, **easy replication support**, **simple API**, **eventually consistent**, a **huge data amount**, and more.

Source: http://nosql-database.org/

NoSQL: typical applications

Some typical applications that use NoSQL:

- **Social media** (Facebook, etc.)
- **Web links** (Google search)
- **Marketing and sales** (Amazon, etc.)
- **Interactive maps** (Google maps, etc.)
- **Email** (Gmail, etc.)
- **Ontologies and Knowledge Graphs** (Equinor, Bosch, etc.)

Types of NoSQL Databases

Core types

- **Key-value** stores
- **Wide column** (column family, column oriented, …) stores
- **Document** stores
- **Graph** databases

Non-core types

- **Object** databases
- Native **XML** databases
- **RDF** stores
- …

Features of NoSQL Databases

Data model

- Traditional approach: relational model
- (New) possibilities:
  - **Key-value**, **document**, **wide column**, **graph**
  - Object, XML, RDF, …
- Goal
  - Respect the real-world nature of data
    (i.e. data structure and mutual relationships)

NoSQL Databases: Aggregate structure

- Aggregate definition
  - Data unit with a complex structure
  - **Collection of related data pieces we wish to treat as a unit** (with respect to data manipulation and data consistency)
- Examples
  - **Value** part of key-value pairs in key-value stores
  - **Document** in document stores
  - **Row** of a **column family** in wide column stores
- Types of systems
  - **Aggregate-ignorant**: relational, graph
    - ➢ It is not a bad thing, it is a feature
  - **Aggregate-oriented**: key-value, document, wide column
- Design notes
  - No universal strategy how to draw **aggregate boundaries Atomicity** of database operations: just a <u>single aggregate at a time</u>

Features of NoSQL Databases

Elastic scaling

- Traditional approach: scaling-up
  - Buying bigger servers as database load increases
- New approach: scaling-out
  - Distributing database data across multiple hosts
    - Graph databases (unfortunately): difficult or impossible at all

Data distribution

- Sharding
  - Particular ways how database data is split into separate groups
- Replication
  - Maintaining several data copies (performance, recovery)

Features of NoSQL Databases

Automated processes

- Traditional approach

  Expensive and highly trained database administrators

- New approach: **automatic recovery, distribution, tuning, …**

Relaxed consistency

- Traditional approach

  Strong consistency (ACID properties and transactions)

- New approach

  Eventual consistency only (BASE properties)

  I.e. we have to make trade-offs because of the data distribution

Features of NoSQL Databases

Schemalessness

- Relational databases
  - Database schema present and **strictly enforced**
- NoSQL databases
  - **Relaxed schema** or **completely missing**
  - Consequences: **higher flexibility**
    - Dealing with **non-uniform data**
    - **Structural changes** cause no overhead
  - However: there is (usually) an **implicit schema**
    - We must know the data structure at the application level anyway

Open source

- Often community and enterprise versions (with extended features or extent of support)

Simple APIs

- Often state-less application interfaces (HTTP)

Current State: Five advantages

- **Scaling**
  - Horizontal distribution of data among hosts
- **Volume**
  - High volumes of data that cannot be handled by RDBMS
- **Administrators**
  - No longer needed because of the automated maintenance
- **Economics**
  - Usage of cheap commodity servers, lower overall costs
- **Flexibility**
  - Relaxed or missing data schema, easier design changes

Current State: Five challenges

- **Maturity**
  - Often still in the pre-production phase with key features missing
- **Support**
  - Mostly open source, limited sources of credibility
- **Administration**
  - Sometimes , it is relatively difficult to install and maintain
- **Analytics**
  - Missing support for business intelligence and ad-hoc querying
- **Expertise**
  - Still a low number of NoSQL experts available in the market

Conclusion

The end of relational databases?

- Certainly no
  - They are still suitable for most projects
  - Familiarity, stability, feature set, available support, …
- However, we should also consider different database models and systems
  - Polyglot persistence = **usage of different data stores in different circumstances**

Lecture Conclusion

Big Data

- 4V characteristics: volume, variety, velocity, veracity

NoSQL databases

- (New) **logical models**
  - Core: key-value, wide column, document, graph Non-core: XML, RDF, …
- (New) **principles and features**
  - Horizontal scaling, data sharding and replication, eventual consistency, …