

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

MODULE 1

Data: Definition, Types, and Characteristics

Definition

- Data: individual facts (numbers, words, measurements, observations, descriptions).

Types

- Quantitative: numeric (weight, volume, cost).
- Qualitative: descriptive, non-numeric (name, gender, eye color).

Core Characteristics

- Composition: sources, granularity, types, static vs streaming.
- Condition: readiness for analysis; need for cleansing/enrichment.
- Context: where/why/how generated; sensitivity; related events.

Data Quality Dimensions

- Accuracy: sufficiently correct for intended use; capture at point of activity.
- Validity: follows rules/definitions; measures what's intended.
- Reliability: stable, consistent processes; identifiable source data.
- Timeliness: captured and available quickly enough for decisions.
- Relevance: suited to current purposes; review as needs change.
- Completeness: requirements clear; collection matched to needs.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

2) Classification of Data

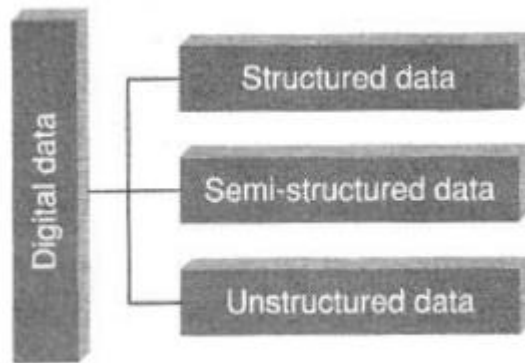


Figure 1.1 Classification of digital data.

Structured Data

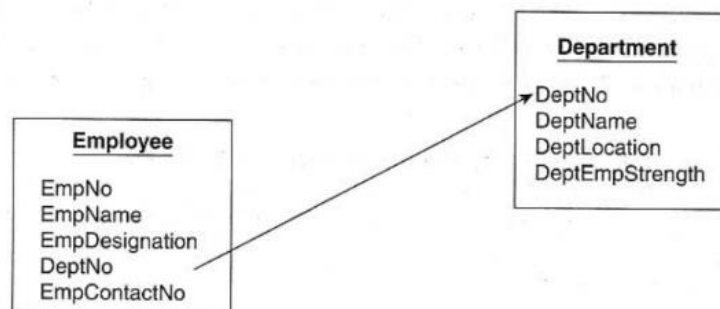


Figure 1.3 Relationship between "Employee" and "Department" tables.

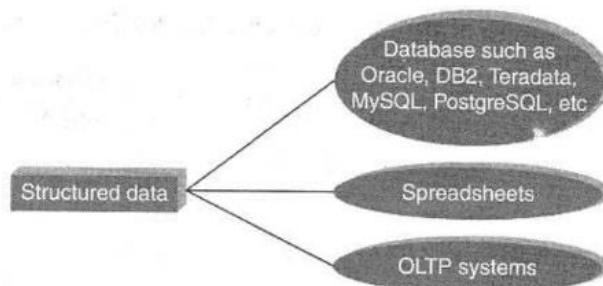


Figure 1.4 Sources of structured data.

- Fixed fields, rows/columns; SQL-managed; easy CRUD, indexing, security, scaling up; ACID transactions.
- Sources: SQL DBs, spreadsheets, OLTP, forms, sensors, logs, medical devices.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

Unstructured Data

- No fixed model; examples: images, video, audio, PDFs, PPT, web pages, emails, blogs.
- Handling: data mining (association, regression, collaborative filtering), text analytics, NLP, noisy text analytics.

Semi-Structured Data

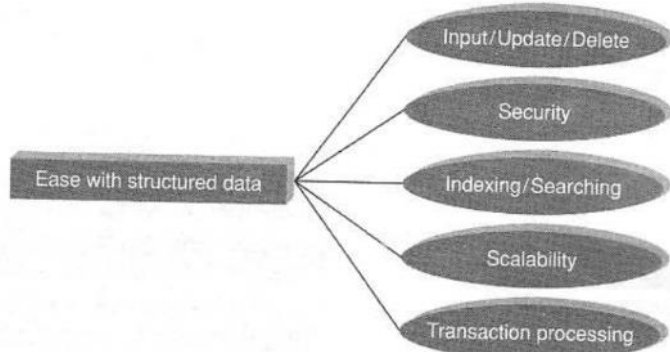


Figure 1.5 Ease of working with structured data.

- Self-describing with tags/metadata (XML, JSON, emails); flexible schema-on-read; same class may have different attributes.

3) Evolution and Definition of Big Data

Timeline

- Mainframes (pre-1980): primitive, structured.
- RDBMS (1980s–1990s): data-intensive apps.
- Web + IoT: explosion of structured, semi-structured, unstructured, multimedia.

Definition

- High-volume, high-velocity, high-variety information assets requiring cost-effective, innovative processing for enhanced insight and decisions.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

3 Vs (+extras)

- Volume: massive amounts from many sources.
- Velocity: batch → real-time/streaming.
- Variety: structured, semi-structured, unstructured.
- Also: Veracity (noise/bias), Validity (accuracy), Volatility (retention time), Variability (inconsistent peaks).

4) Why Big Data? Benefits and Challenges

Benefits

- Understand customer behavior; loyalty uplift; cross/upsell; targeted promos.
- Optimize operations/workforce/supply chain.
- Predict trends and needs; innovate; discover revenue; social listening.

Importance

- Cost/time savings (Hadoop/Spark; in-memory).
- Market insights; customer acquisition/retention; better ads; product innovation.

Key Challenges

- Massive data management → cloud scaling.
- Integrating diverse sources → ETL/ELT, data integration, BI.
- Ensuring data quality → governance, data quality tools.
- Security → encryption, IAM, endpoint protection, monitoring.
- Tool selection → expertise/consultants.
- Efficient scaling and cost control → plan schemas, purge policies.
- Skills gap → hire/train/self-service analytics.
- Organizational resistance → start small, prove value.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

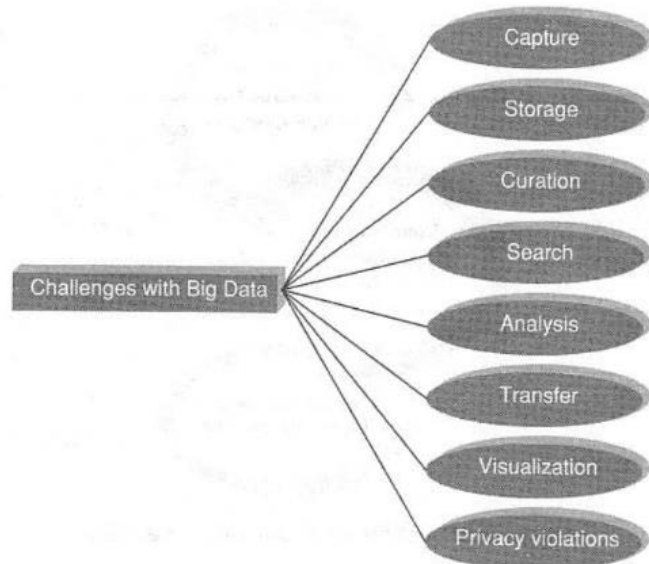


Figure 2.4 Challenges with big data.

5) Traditional BI vs Big Data

Traditional BI

- Centralized data warehouse, schema-on-write, structured data, offline analysis, vertical scaling; move data to code.

Big Data

- Distributed file systems, horizontal scaling, real-time + batch, handles all data types, move code to data, MPP.

6) Typical Data Warehouse vs Hadoop Environment

Data Warehouse (DW)

- ETL → DW → BI tools; consolidation, analysis, reporting at aggregates.

Hadoop Environment

- HDFS + MapReduce/YARN; scales on commodity hardware; fault-tolerant; batch-oriented; complements OLTP/OLAP.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

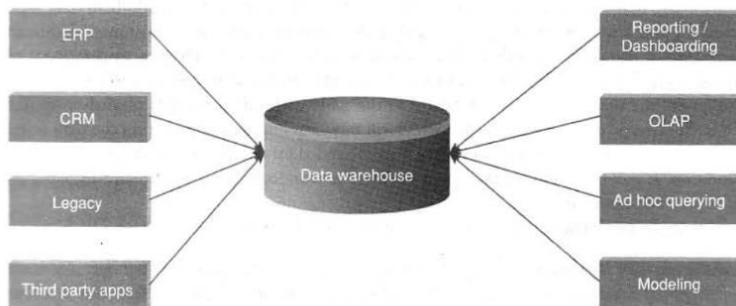


Figure 2.9 A typical data warehouse environment.

7) Analytics Types and Generations

By Maturity

- Basic: reporting, slicing/dicing, historical viz.
- Operationalized: embedded in processes.
- Advanced: predictive/prescriptive models.
- Monetized: direct revenue via analytics.

By Era

- Analytics 1.0 (1950s–2009): descriptive; internal structured data.
- Analytics 2.0 (2005–2012): descriptive + predictive; big data, Hadoop, external sources.
- Analytics 3.0 (2012–present): descriptive + predictive + prescriptive; ML, in-memory, mixed sources.

Descriptive vs Predictive vs Prescriptive

- Descriptive: what happened; aggregation/mining.
- Predictive: what will happen; ML, forecasting.
- Prescriptive: what should we do; recommendations, constraints.

8) Technologies for Big Data

Performance Approaches

- In-memory analytics: RAM-resident datasets; faster insights.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- In-database processing: run analytics inside DB/EDW.
- SMP: shared memory among CPUs.
- MPP: many nodes with own OS/memory; message passing.

Architectures

- Parallel (tightly coupled) vs Distributed (loosely coupled).
- Shared-nothing: neither memory nor disk shared; fault isolation; scalability.

CAP Theorem

- In distributed systems, choose at most two: Consistency, Availability, Partition tolerance.
 - AP, CP, CA (practically AP/CP under partitions).

9) Analytical Tools

Proprietary

- Excel, SAS, IBM SPSS Modeler, Statistica, Salford Systems, WPS.

Open Source

- R, Weka.

10) NoSQL and NewSQL

NoSQL

- Not Only SQL; non-relational, distributed, schema-flexible; typically favor availability/partition tolerance; eventual consistency; auto-sharding, replication.

Types

- Document (MongoDB), Key-Value (Redis, Coherence), Column-family (Bigtable/Cassandra/HBase), Graph (Amazon Neptune).

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

Why NoSQL

- Horizontal scale-out; large multi-structure data; dynamic schema; replication; sharding; cost-effective.

NewSQL

- Modern RDBMS with shared-nothing and SQL interface; aims for NoSQL-like scale with ACID guarantees.

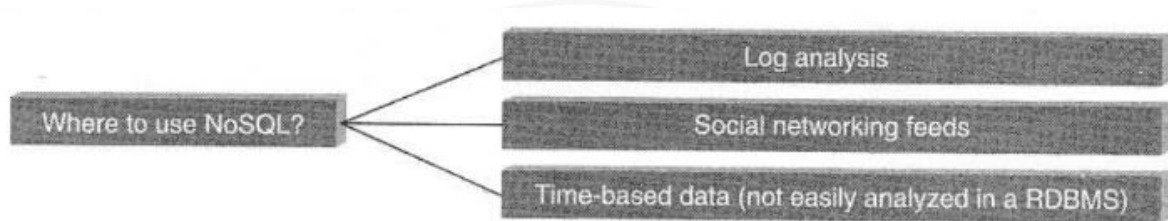


Figure 4.1 Where to use NoSQL?

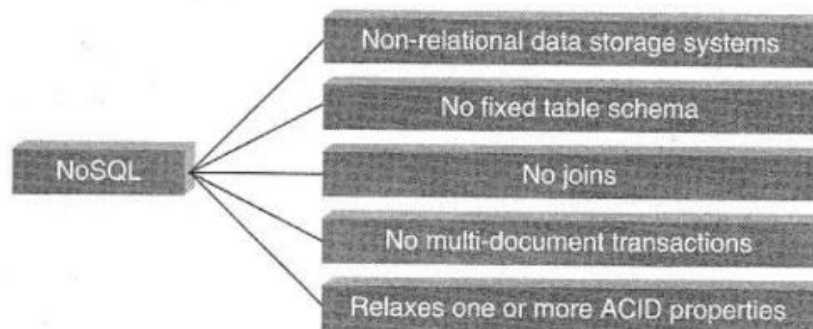


Figure 4.2 What is NoSQL?

11) Hadoop Deep Dive

Basics

- Apache open source; Java; inspired by Google GFS and MapReduce.
- Core: HDFS (storage), YARN (resource management), MapReduce/Spark (processing).

Features

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- Schema-on-read; high throughput; fault-tolerant replication; cost-effective; parallel; best for large files and batch.

Hadoop 1.0 vs 2.0

- 1.0: HDFS + MapReduce tightly coupled; batch-only; MapReduce expertise required.
- 2.0: YARN generalizes resource management; supports multiple processing models (batch + real-time); less dependence on MapReduce.

Ecosystem

- Ingestion: Sqoop (RDBMS↔Hadoop), Flume (logs/events → HDFS/HBase).
- Processing: MapReduce, Spark (in-memory; SQL, Streaming, MLlib, GraphX).
- Analysis: Pig (Pig Latin), Hive (HiveQL, schema-on-read), Impala (low-latency SQL).
- Ops/Utilities: HBase (NoSQL on HDFS), Zookeeper (coordination), Oozie (workflow), Mahout (ML), Chukwa (collection), Ambari (management).

HDFS vs HBase

- HDFS: file system, WORM, high-latency, batch, MapReduce access.
- HBase: NoSQL, random read/write, low-latency, API access, real-time analytics

12) Hadoop vs SQL (Relational)

Table 4.6 Hadoop versus SQL

Hadoop	SQL
Scale out	Scale up
Key-Value pairs	Relational table
Functional Programming	Declarative Queries
Offline batch processing	Online transaction processing

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- Volume: PB vs GB-scale.
- Storage model: distributed files/columns/kv vs tabular fixed schema.
- Scaling: horizontal vs mostly vertical.
- Access: batch vs interactive/batch.
- Cost: open-source/commodity vs licensed/proprietary.
- Schema: dynamic/schema-on-read vs schema-on-write.
- Updates: write-once/read-many vs frequent read/write.
- Integrity: lower vs high; ACID typical in SQL.

13) Distributions and Cloud

Core Apache Hadoop

- Hadoop Common, HDFS, YARN, MapReduce.

Vendor Distributions/Services

- Cloudera, Hortonworks, IBM, Microsoft, Teradata, AWS, etc.
- Cloud connectors (e.g., Google Cloud Storage connector to run jobs on cloud data directly).