

Big Data

NoSQL	Not Only SQL
3 Vs of Big Data	Volume (size), Velocity (speed), Variety (types).
Volume	Massive amount of data generated and stored.
Velocity	Speed at which data is produced, transferred, and processed.
Variety	Diversity of data formats and sources (text, images, videos, sensors, logs, etc.).
Data Modelling	Defining structure, relationships & rules for data.
Relational DB query language	SQL (Structured Query Language)
ACID	Atomicity, Consistency, Isolation, Durability
BASE	Basically Available, Soft state, Eventual consistency

Scaling

Vertical Scaling (up)	Add power to one server.
Horizontal Scaling (up)	Cluster of low-cost servers (preferred for Big Data).
Normalisation	Organising data to reduce redundancy, improve integrity.
Denormalisation	Trade redundancy for faster queries.

Data Model Components

Entity	Object / Class (e.g Student)
Entity Instance	Record of a entity
Attribute	Property / Field (e.g Age)

Data Model Components (cont)

Relationship	Connection (e.g Student -> EnrolledIn -> Course)
MongoDB	
Document Oriented Database	Subclass of key-value databases.
Document	Encoded in format such as XML, JSON, BSON.
Schema-less	No predefined structure on the stored data & each document can have its own structure.
Collection	Group of documents.

SQL VS MongoDB

SQL TERM	MONGODB TERM
database	database
table	collection
index	index
row	document
column	field
joining	embedding & Linking

NoSQL Databases

Key-Value	Stores simple key/value pairs.	DynamoDB, Redis
Document	Stores JSON-like documents.	MongoDB, CouchDB
Column	Stores data by columns instead of rows.	Cassandra, BigTables
Graphs	vStores data as nodes + relationships.	Neo4j, ArangoDB
Time Series	(not part of the unit but whatever it still is a type of NoSQL database)	Prometheus, Timescale (postgreSQL fork (it's great tbh))

NoSQL - Quick Overview

- Non-relational and schema-less.
- Supports distributed database architectures.
- Provides high scalability, and high availability.
- Able to support very large amounts of sparse data.
- Designed mainly towards performance rather than transaction consistency.

NoSQL - Pros & Cons

Advantages	Disadvantages
1. High scalability, and availability	1. Complex programming is required.
2. It uses low-cost commodity hardware.	2. There is no relationship support—only by application code.
3. It supports Big Data.	3. There is no transaction integrity support.
4. It typically improves storage efficiency.	4. In terms of data consistency, it provides an eventually consistent model.

SQL - Pros & Cons

Advantages	Disadvantage
1. Data Integrity and Accuracy	1. Scalability Issues: Horizontal scalability is challenging, making it less suitable for very large datasets or high-traffic applications.
2. Structured Data: Suitable for structured data with predefined schema, making it easier to organize and query.	2. Complexity in Schema Design: Requires complex schema design and management, which can be inflexible to changes.
3. Relationship Support: Strong support for data relationships using foreign keys and join operations.	3. Performance Bottlenecks: Can face performance bottlenecks with high-volume read and write operations.
4. SQL Language: Uses a powerful and standardized query language (SQL) for database management and querying.	4. Cost: Often requires expensive hardware and software licenses, and maintenance can be costly.