Debasmita Bagchi (UG/02/BTCSEAIML/2023/065) → CA

16/2/26

1) Define an unstructured DB with example.

↳ An unstructured DB refers to a DB system designed to store and manage unstructured data (information that lacks a predefined, fixed format or schema, unlike rows/columns in traditional tables.)

Eg:) MongoDB → It excels at handling unstructured and semi-structured data and stores data in flexible, JSON-like documents (BSON format), allowing each document in a collection to have completely different fields and structures without requiring a fixed schema.

Ex of unstructured data (commonly stored in DBs like MongoDB):

i) Text documents (eg: Word files, emails, PDFs, social media posts, articles etc.)

ii) Images (JPEG, PNG photos – can be stored as binary data or referenced)

iii) Audio files (recordings, music, podcasts)

iv) Video files.

v) Web pages / HTML content.

vi) Sensor / IoT data logs.

vii) Emails (body content)

viii) Social media feeds.

(Q2) Explain why traditional RDBMS are not suitable for handling unstructured data.

Ans. Traditional RDBMS follow a rigid, tabular structure with predefined schemas, making them unsuitable for unstructured data for the following key reasons:

i) Fixed Schema Requirement: RDBMS require tables with fixed columns and datatypes defined in advance. Unstructured data (eg, documents, images etc.) has no consistent structure and thus, forcing it into tables results in either extremely wide tables with many NULL values or numerous auxiliary tables, both of which are inefficient and hard to maintain.

ii) Inefficient Storage of Variable & Large Formats: Binary Large Objects (BLOBs) can store images/videos, but RDBMS offer no built-in way to query or search inside them (eg: 'find all documents containing the word AI' or 'images with faces'). This makes content-based retrieval practically impossible without external tools.

iii) Poor Scalability for Volume and Variety: Unstructured data grows rapidly and comes in huge volumes (terabytes to petabytes). RDBMS typically scale vertically (bigger servers), which is expensive and has limits. They struggle with the "variety" dimension of big data, as jobs become complex and slow when handling highly variable schemas.

iv) Querying & Performance limitations: SQL is designed for structured, predictable queries with joins & aggregations. Unstructured data requires flexible, schema-agnostic searches (eg: full-text search, nested fields, geospatial, vector similarity). RDBMS perform poorly on such queries due to lack of native support and overhead of normalization/relationships.

v) ACID Overhead Becomes a Bottleneck:
Strict ACID compliance (especially strong consistency) is costly for write-heavy, high-velocity unstructured workloads (eg: real-time logs, social feeds). This makes RDBMs slow and resource-intensive compared to systems optimized for eventual consistency and massive scale.
In conclusion, RDBMs excel at structured, transactional data with fixed schemas (eg: banking, inventory), but they are fundamentally mismatched for the schema-less, diverse, and high-volume nature of unstructured data.

Q3) Discuss the role of NoSQL DB in manging unstructured data.

↳ No SQL databases play a central and increasingly dominant role in manging unstructured data due to their design principles tailored for flexibility, scale, and performance in modern big data environments.

i) Schema-less / Flexible Schema Design: Unlike RDBMs, NoSQL DBs (especially document oriented ones like MongoDB) do not enforce a fixed schema. Each document/record can have completely different fields and structures. This allows easy ingestion of unstructured data (JSON like documents, text, nested objects, binary) without preprocessing or schema redesign.

ii) Support for Multiple Data Models: No SQL offers various types suited to unstructured needs:
a) Document stores (MongoDB) → ideal for JSON/BSON documents, semi-structured content.
b) Key-value stores (Redis) → fast caching of blobs.
c) Wide-column stores (Cassandra) → massive time-series or log data.
d) Graph DBs (Neo4j) → relationships in social / unstructured networks.

iii) Horizontal scalability & High Volume Handling :

No SQL DBs are designed to scale out across many commodity servers (sharding + replication). This makes them ideal for petabyte-scale unstructured data with high write/read throughput - critical for social media, IOT logs, content platforms etc.

iv) Native Support for Diverse Formats & Fast Ingestion :

Data is stored close to its original form (eg, embedded arrays, binary data, nested JSON). Features like full-text indexing, geospatial indexing, and vector search (in modern NoSQL) enable powerful querying of unstructured content without heavy ETL.

v) Integration with Modern Ecosystems :

NoSQL DBs integrate seamlessly with big data tools (Hadoop, Spark), real-time streaming (Kafka), AI/ML pipelines (vector embeddings for images/text), and data lakes. They support the 3Vs of big data (Volume, Variety, Velocity) far better than traditional RDBMs.