Database Systems: Relational (SQL) & NoSQL (MongoDB)

Comprehensive Technical Study Notes & Architectural Overview

---

1. Relational Databases & SQL Overview

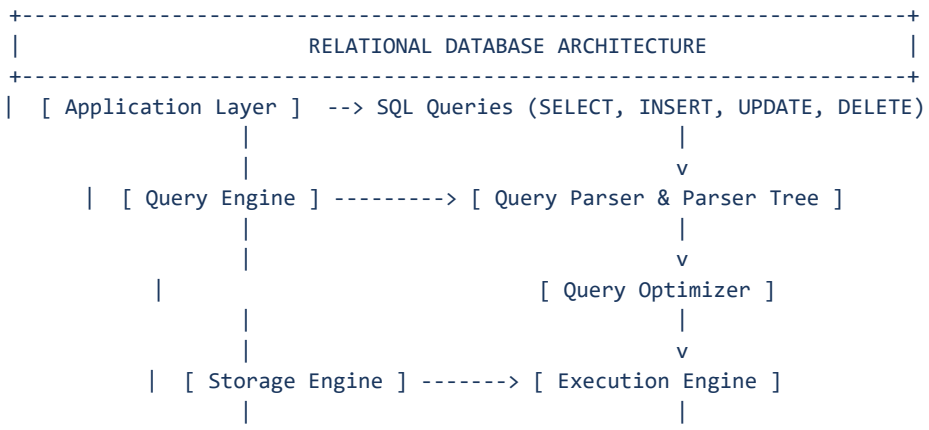
A **Relational Database Management System (RDBMS)** stores data in structured, tabular formats consisting of rows (tuples) and columns (attributes). Data relationships are defined using schema structures, primary keys, and foreign keys.

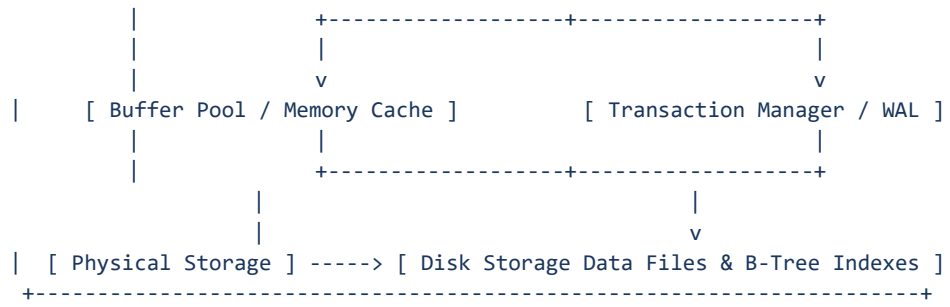
1.1 Core Principles & ACID Properties

- **Atomicity:** Transactions are all-or-nothing. If a single statement fails, the entire transaction is rolled back.
- **Consistency:** Ensures data moves from one valid state to another, strictly respecting constraints, keys, and rules.
- **Isolation:** Concurrent execution of transactions leaves the database in the same state as sequential execution.
- **Durability:** Committed data is saved permanently, even in the event of a system failure or power loss.

> □ **RDBMS USE CASE:** Relational databases prioritize immediate consistency and strict structure, making them ideal for financial applications, ERP systems, and transactional domains.

1.2 Relational Database Architecture Diagram





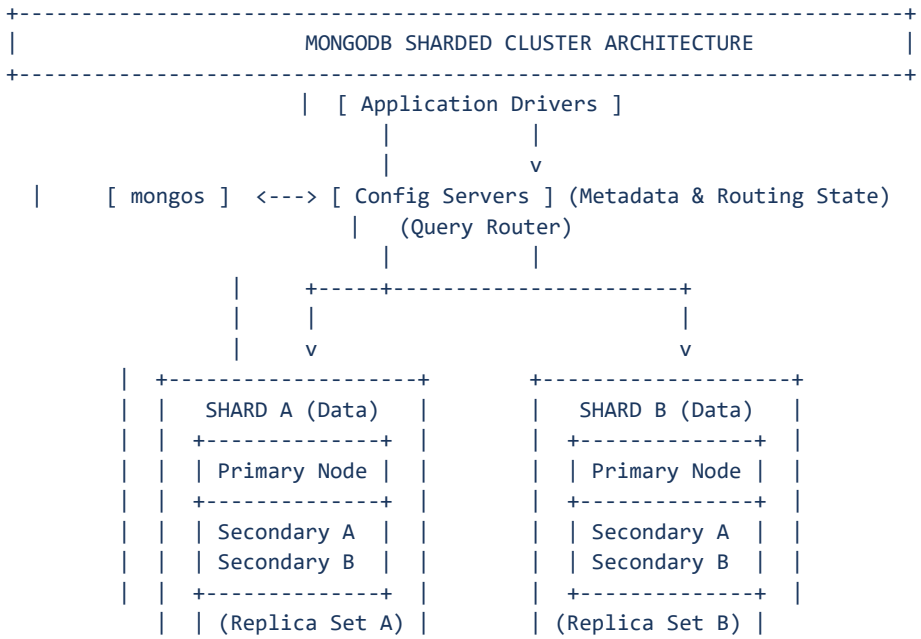
2. NoSQL Databases & MongoDB Overview

NoSQL ('Not Only SQL') databases provide a dynamic, non-tabular mechanism for storage and retrieval. **MongoDB** is a leading document-oriented NoSQL database that stores data in flexible, JSON-like documents (BSON format).

2.1 Key Features of MongoDB & BASE Model

- **Basically Available:** Guarantees availability of the data through distribution and replication across nodes.
- **Soft State:** State of the system can change over time without explicit user interaction due to eventual consistency.
- **Eventual Consistency:** The data will become consistent over time across all replica nodes.
- **Dynamic Schema:** Documents in the same collection do not need to have the same fields or structure.
- **Horizontal Scalability:** Scales out seamlessly using Sharding (distributing data across multiple clusters).

2.2 MongoDB Architecture Diagram





3. Detailed Comparison: SQL vs. NoSQL (MongoDB)

Feature	Relational Databases (SQL)	MongoDB (NoSQL)
Data Structure	Tables (Rows & Columns)	JSON/BSON Documents
Schema Type	Rigid, Predefined Schema	Dynamic / Schema-flexible
Scalability	Vertical (Scale Up)	Horizontal (Scale Out via Sharding)
Consistency Model	ACID (Immediate Consistency)	BASE (Eventual Consistency / Configurable)
Query Language	Structured Query Language (SQL)	MongoDB Query Language (MQL)
Relationships	Foreign Keys, JOIN operations	Embedded Documents & $lookup
Primary Use Case	Financial, ERP, High Integrity Data	Big Data, Real-time Web, Content Management

> □ **SUMMARY VERDICT:** Choose Relational (SQL) when strict transactional consistency and normalized structures are required. Choose MongoDB (NoSQL) when handling high-velocity data, dynamic content structures, or rapid horizontal scaling.