

Week 9: Introduction to MongoDB and NoSQL Databases

Day 1: Understanding MongoDB and the NoSQL Revolution

Theoretical Overview

MongoDB is a cross-platform, document-oriented database classified as a NoSQL (Not Only SQL) database. Unlike traditional relational databases (like MySQL or Oracle) that use tables and rows, MongoDB uses JSON-like documents with dynamic schemas.

- **Positioning of MongoDB:** It is designed for high availability, horizontal scalability, and ease of development.
- **MongoDB Features:** Key features include ad-hoc queries, indexing, replication for high availability, and auto-sharding for horizontal scale.
- **Data Model:** MongoDB stores data in "Collections" (equivalent to tables) which contain "Documents" (equivalent to rows). These documents use the BSON format (Binary JSON) to support a wide variety of data types.

Day 2: MongoDB Use Cases and Limitations

When to Use MongoDB

MongoDB is ideal for applications requiring rapid growth and flexible data structures, such as:

- **Content Management Systems (CMS):** Managing diverse media types.
- **E-commerce Platforms:** Handling product catalogs with varying attributes (essential for your **GadgetShop** project).
- **Real-time Analytics:** Storing high-velocity data streams.

Limitations and Challenges

While powerful, MongoDB has specific limitations:

- **Memory Usage:** MongoDB consumes more memory as it stores field names within every document.
- **Joins:** Unlike SQL, complex joins are not the primary focus, encouraging "data embedding" or "referencing" instead.
- **Document Size:** There is a hard limit of 16MB per BSON document.

Day 3: MongoDB Query Language (QL) and CRUD Operations

The Foundation of Data Manipulation

CRUD stands for Create, Read, Update, and Delete. These are the four basic functions of persistent storage.

- **Inserts (Create):** Documents are added to a collection using methods like `insertOne()` or `insertMany()`.
- **Find (Read):** The `find()` method is used to retrieve documents from a collection. It can take an optional "query filter" to narrow down results.
- **FindOne:** This method specifically returns only the first document that matches the query criteria, which is highly efficient for fetching specific user or product IDs.

Day 4: Advanced Querying and Database Operations

Querying with Filters

MongoDB Query Language (QL) allows for complex filtering using operators:

- **Comparison Operators:** \$gt (greater than), \$lt (less than), and \$in (matches any value in an array).
- **Logical Operators:** \$and, \$or, and \$not to combine multiple query conditions.
- **Sorting and Projection:** The ability to sort results and select only specific fields (projection) to return to the application.

Performing Database Operations

Beyond simple reads, we explored operational commands:

- **Updating Documents:** Using \$set to modify specific fields without overwriting the entire document.
- **Deleting Documents:** Using deleteOne() or deleteMany() to remove data.