**Introduction to MongoDB Data Modeling**

MongoDB uses a flexible schema design, allowing developers to model data according to application needs. This guide covers best practices for structuring MongoDB documents and collections effectively.

**2. Data Modeling Approaches**

**2.1 Embedded Documents (Denormalization)**

Embedded documents store related data within a single document, reducing the need for joins.

**Example: Storing Order and Customer Information Together**

**Advantages:**

* Faster read operations.
* Less need for joins.

**When to use:**

* When related data is frequently read together.
* When data does not grow indefinitely.

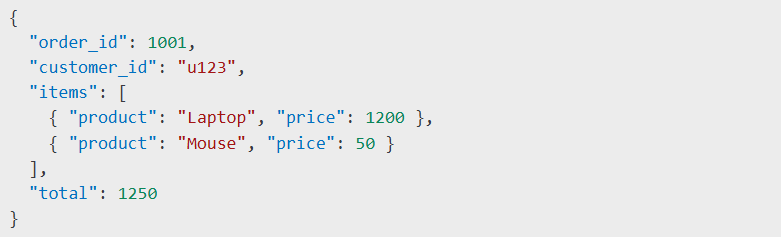
**2.2 Referencing (Normalization)**

Referencing uses separate collections with document references (similar to foreign keys in relational databases).

**Example: Storing Users and Orders Separately**

**Users Collection:**

**Orders Collection:**

**Advantages:**

* Better scalability for large datasets.
* Easier updates when shared data changes.

**When to use:**

* When data relationships are complex.
* When related data changes frequently.

**3. Schema Design Patterns**

**3.1 Single Table Inheritance**

Useful when multiple types of objects share common attributes.

**Example: Users with Different Roles**

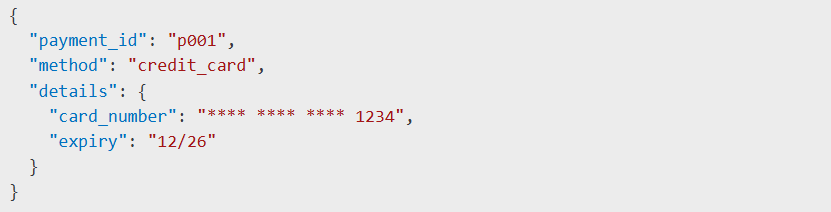
**When to use:**

* When entities share common fields but differ slightly.

**3.2 Polymorphic Pattern**

Allows different document structures within a single collection.

**Example: Different Payment Methods**

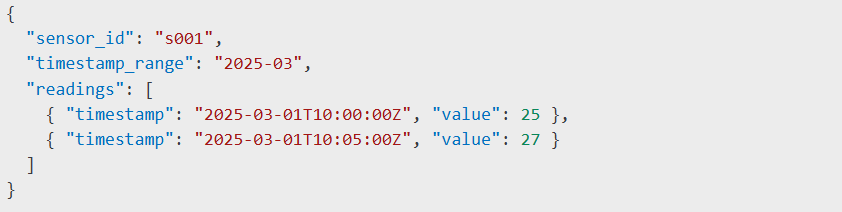
**When to use:**

* When multiple document structures share similar attributes.

**3.3 Bucket Pattern**

Groups time-series data into buckets to improve query performance.

**Example: Sensor Data**

**When to use:**

* When handling large volumes of time-series data.

**4. Indexing Strategies**

Indexing improves query performance by optimizing lookups.

**4.1 Single-Field Index**

**Use Case:** Fast lookup by email.

**4.2 Compound Index**

**Use Case:** Queries filtering by customer\_id and sorting by order\_date.

**5. Aggregation Framework**

MongoDB’s aggregation framework enables powerful data transformations.

**Example: Calculate Total Sales Per Customer**







