# MongoDB Data Modelling: StoryVerse

## What Is Data Modelling?

In MongoDB, **data modeling** is the process of designing the **structure of your documents and collections** to match your app's requirements and performance needs.

## Project Context: StoryVerse - *Verseify*

### Identify Collections

|  |  |
| --- | --- |
| **Collection** | **Purpose** |
| users | Store user details including profile info, roles (reader/writer/editor), preferences, badges, and subscription details. |
| stories | Main record of each collaborative story: title, genre, mood, summary, contributors, status, and story prompt link. |
| chapters | Chapters contributed to stories: content, order, contributor ID, attached media, vote count, and approval status. |
| characters | Store custom characters created by users for stories: names, traits, story linkage, and timeline references. |
| maps | Interactive world-building maps with markers, story association, and clickable location metadata. |
| comments | Comments and threaded discussions under stories or chapters, linked with user ID and timestamps. |
| endorsements | Track endorsements (tokens or virtual gifts) given by users to writers; includes sender, receiver, and value. |
| votes | User voting records for selecting canon chapters, branching paths, or challenge winners. |
| prompts | Reader/AI-created prompts to inspire stories: includes prompt text, tags, and claimed status. |
| timelines | Chronological event tracking for stories or characters to maintain consistency. |
| notifications | Alerts and updates for users: new comments, replies, votes, story approvals, endorsements, etc. |
| media | Stores multimedia (image/audio/video) attachments linked to chapters or characters. |
| roles | Defines roles like co-author, editor, or illustrator assigned to users per story/chapter with permissions. |
| subscriptions | Manages user subscriptions for monetization (e.g., premium stories or access tiers). |
| badges | Achievements for users based on participation, contributions, challenges, and reading habits. |
| leaderboards | Displays rankings of top writers or active users based on votes, tokens earned, or badges. |
| branch\_choices | Stores branching narrative options for chapters, along with reader vote tallies and selected outcomes. |
| challenges | Writing contests with themes, rules, submissions, deadlines, and result tracking. |
| audio\_stories | Narrated story episodes (AI or user-recorded) with chapter mapping and sound effect metadata. |
| ar\_vr\_assets | Metadata and links to 3D/VR/AR elements for immersive story environments (e.g., fantasy worlds). |

### Define Relationships

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Relationship** | **Type** | **Description** | **Field Reference** | **Modeling Approach** | **Justification** |
| User ↔ Stories | One-to-Many | A user can create many stories. | stories.creatorId → users.\_id | Reference | Keeps stories scalable; avoids user duplication. |
| Story ↔ Chapters | One-to-Many | Each story contains multiple chapters. | chapters.storyId → stories.\_id | Reference | Chapters grow over time; separate collection helps versioning. |
| Chapter ↔ Contributors | Many-to-Many | A chapter can have multiple contributors (writer, editor, etc.). | chapters.contributors[] → users.\_id | Reference (array) | Enables collaboration without data duplication. |
| User ↔ Comments | One-to-Many | A user can post many comments. | comments.userId → users.\_id | Reference | For tracking discussion authorship. |
| Chapter/Story ↔ Comments | One-to-Many | A story/chapter can have many comments. | comments.chapterId/storyId → chapters/stories.\_id | Reference | Needed for threaded discussion on content. |
| Story ↔ Characters | One-to-Many | Stories have multiple user-created characters. | characters.storyId → stories.\_id | Reference | Reusability and timeline integration. |
| Story ↔ Maps | One-to-One (Optional) | Stories can have linked world maps. | maps.storyId → stories.\_id | Reference | Interactive data best managed independently. |
| User ↔ Endorsements | One-to-Many | Users can endorse (gift) other users. | endorsements.fromUserId/toUserId → users.\_id | Reference | Tracking user-to-user engagement. |
| Story/Chapter ↔ Media | One-to-Many | Chapters or stories can have multiple media files. | media.chapterId/storyId → chapters/stories.\_id | Reference | Media files carry metadata; better stored separately. |
| Story ↔ Timeline | One-to-One (Optional) | Each story may have a character/story timeline. | timelines.storyId → stories.\_id | Reference | Useful for continuity across chapters. |
| User ↔ Prompts | One-to-Many | Users can submit prompts for stories. | prompts.userId → users.\_id | 🔗Reference | Needed for story-seed tracking and challenges. |
| Prompt ↔ Story | One-to-One (Optional) | A story may originate from a prompt. | stories.promptId → prompts.\_id | Reference | Useful for tracking source idea of a story. |
| Story ↔ Votes | One-to-Many | Each story can have multiple votes (e.g., for branching). | votes.storyId → stories.\_id | Reference | Enables fair canonization of branches. |
| User ↔ Wishlist (Prompt) | Many-to-Many | Users can wishlist story prompts. | users.wishlist[] → prompts.\_id | Reference (array) | Fast access to saved prompts. |
| Story ↔ Tags | One-to-Many (Embedded) | Stories can include multiple genre/mood tags. | stories.tags[] | Embed | Small, static strings; ideal for embedding. |
| Story ↔ Branch Choices | One-to-Many | Stories may branch into alternative paths. | branch\_choices.storyId → stories.\_id | Reference | Needed for branching narrative structure. |
| User ↔ Notifications | One-to-Many (Embedded) | Users receive notifications for updates, replies, votes, etc. | users.notifications[] | Embed | Frequently accessed, short-lived data; embedding is optimal. |
| Story/Chapter ↔ Audio Stories | One-to-One / Many | A chapter or story may have an audio narration. | audio\_stories.chapterId/storyId → chapters/stories.\_id | Reference | Audio with metadata needs a separate model. |
| Story ↔ AR/VR Assets | One-to-One / Optional | Stories may include immersive 3D/VR scenes. | ar\_vr\_assets.storyId → stories.\_id | Reference | Interactive experiences are heavy; better stored separately. |
| User ↔ Badges | One-to-Many (Embedded) | Users earn badges for milestones and challenges. | users.badges[] | Embed | Badges are small and non-relational; embedding improves access speed. |
| User ↔ Challenges | Many-to-Many | Users can join many challenges; each challenge has many participants. | challenges.participants[] → users.\_id | Reference (array) | Gamification and participation tracking. |

### 3. Embed vs Reference

### 

|  |  |
| --- | --- |
| **Use Case** | **Recommended Modeling** |
| Chapters inside a story | Reference storyId |
| Contributors of a chapter | Reference userIds (array) |
| Characters created for a story | Reference storyId |
| Tags of a story | Embed |
| Comments on chapters or stories | Reference storyId/chapterId & userId |
| Media attached to chapters or stories | Reference chapterId/storyId |
| User notifications | Embed |
| User badges | Embed |
| Votes on story branches | Reference storyId & userId |
| Wishlist of prompts by user | Embed promptIds in user |
| Timeline for a story | Reference storyId |
| Audio version of a chapter | Reference chapterId |
| AR/VR assets for a story | Reference storyId |
| Prompts created by users | Reference userId |
| Endorsements sent between users | Reference fromUserId & toUserId |
| Challenge participation | Reference userIds in challenge |
| Cart items (if implemented for book/story access) | Embed with story snapshot |

### 4. Sample Schemas

**a. users**

json

Copy code

{

"\_id": ObjectId("..."),

"name": "Alice Writer",

"email": "alice@example.com",

"roles": ["writer", "reader"],

"wishlist": [ObjectId("prompt1"), ObjectId("prompt2")],

"badges": ["StreakMaster", "TopVoted"],

"notifications": [

{

"message": "Your chapter was approved!",

"read": false,

"timestamp": ISODate("2025-07-10T14:00:00Z")

}

],

"createdAt": ISODate("2025-07-01T10:00:00Z"),

"updatedAt": ISODate("2025-07-08T15:00:00Z")

}

**b. stories**

json

Copy code

{

"\_id": ObjectId("..."),

"title": "Tales of Andromeda",

"summary": "A space opera featuring galactic kingdoms and rebellion.",

"creatorId": ObjectId("user1"),

"genre": "Sci-Fi",

"mood": "Suspenseful",

"tags": ["space", "rebellion", "alien"],

"status": "ongoing",

"promptId": ObjectId("prompt1"),

"createdAt": ISODate("2025-07-02T12:30:00Z"),

"updatedAt": ISODate("2025-07-09T10:15:00Z")

}

**c. chapters**

json

Copy code

{

"\_id": ObjectId("..."),

"storyId": ObjectId("story1"),

"title": "Chapter 1: The Uprising",

"content": "The resistance fighters broke into the imperial vault...",

"contributors": [ObjectId("user1"), ObjectId("user2")],

"media": [ObjectId("media1")],

"order": 1,

"votes": 135,

"approved": true,

"createdAt": ISODate("2025-07-03T08:00:00Z"),

"updatedAt": ISODate("2025-07-08T18:30:00Z")

}

**d. prompts**

json

Copy code

{

"\_id": ObjectId("prompt1"),

"text": "What if Earth had two suns and no night?",

"createdBy": ObjectId("user2"),

"claimedBy": ObjectId("user1"),

"tags": ["sci-fi", "alternate universe"],

"createdAt": ISODate("2025-07-01T09:00:00Z")

}

**e. comments**

json

Copy code

{

"\_id": ObjectId("..."),

"chapterId": ObjectId("chapter1"),

"userId": ObjectId("user3"),

"content": "Loved this twist! Can’t wait for the next one.",

"timestamp": ISODate("2025-07-08T17:45:00Z")

}

**f. media**

json

Copy code

{

"\_id": ObjectId("media1"),

"type": "audio",

"url": "https://cdn.storyverse.com/audio/scene1.mp3",

"description": "Soundtrack for Chapter 1 battle",

"uploadedBy": ObjectId("user2"),

"createdAt": ISODate("2025-07-02T16:00:00Z")

}

**g. endorsements**

json

Copy code

{

"\_id": ObjectId("..."),

"fromUserId": ObjectId("user3"),

"toUserId": ObjectId("user1"),

"tokens": 5,

"message": "Your world-building is amazing!",

"timestamp": ISODate("2025-07-07T19:30:00Z")

}

**h. votes**

json

Copy code

{

"\_id": ObjectId("..."),

"storyId": ObjectId("story1"),

"chapterId": ObjectId("chapter2"),

"userId": ObjectId("user4"),

"vote": true,

"timestamp": ISODate("2025-07-08T13:20:00Z")

}

**i. characters**

json

Copy code

{

"\_id": ObjectId("..."),

"storyId": ObjectId("story1"),

"name": "Captain Zara Voss",

"role": "Protagonist",

"description": "A fearless rebel leader with a mysterious past.",

"traits": ["brave", "strategic", "enigmatic"],

"imageUrl": "https://cdn.storyverse.com/images/zara.png",

"createdBy": ObjectId("user1"),

"createdAt": ISODate("2025-07-03T11:00:00Z"),

"updatedAt": ISODate("2025-07-08T16:45:00Z")

}

**j. maps**

json

Copy code

{

"\_id": ObjectId("..."),

"storyId": ObjectId("story1"),

"title": "The Andromeda Sector",

"mapUrl": "https://cdn.storyverse.com/maps/andromeda.png",

"locations": [

{

"name": "Nova Citadel",

"description": "Capital of the resistance.",

"coordinates": { "x": 120, "y": 75 }

},

{

"name": "Void Barrens",

"description": "A dangerous uncharted region of space.",

"coordinates": { "x": 240, "y": 160 }

}

],

"createdAt": ISODate("2025-07-04T09:15:00Z"),

"updatedAt": ISODate("2025-07-08T18:10:00Z")

}

**k. timelines**

json

Copy code

{

"\_id": ObjectId("..."),

"storyId": ObjectId("story1"),

"events": [

{

"title": "Zara Joins the Rebellion",

"description": "Captain Zara joins the rebellion at age 18.",

"timestamp": "Y2135"

},

{

"title": "Battle of Nova Citadel",

"description": "Major turning point in the rebellion.",

"timestamp": "Y2137"

}

],

"createdBy": ObjectId("user1"),

"createdAt": ISODate("2025-07-05T10:00:00Z"),

"updatedAt": ISODate("2025-07-08T12:00:00Z")

}

**l. audio\_stories**

json

Copy code

{

"\_id": ObjectId("..."),

"chapterId": ObjectId("chapter1"),

"narrator": "AI Voice - Luna",

"duration": 210, // in seconds

"audioUrl": "https://cdn.storyverse.com/audio/andromeda\_ch1.mp3",

"description": "Narrated audio version of Chapter 1.",

"createdBy": ObjectId("user2"),

"createdAt": ISODate("2025-07-06T14:30:00Z"),

"updatedAt": ISODate("2025-07-08T15:00:00Z")

}

#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### ­­­­­­­­­­­­­­­­­­­­­