PostgreSQL and MongoDB Comparison

Ali Emre Pamuk

April 2025

Contents

1 What is MongoDB? 2

2 MongoDB vs PostgreSQL 2

3 Recreating a PostgreSQL Schema in MongoDB 2

1 What is MongoDB?

What is MongoDB?

MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON. It uses collections and documents instead of tables and rows. It’s schema-less, scalable, and optimized for handling large volumes of data with high read/write performance.

- **Document-oriented:** Stores data in JSON-like documents.
- **Schema-less:** Flexible data modeling.
- **Scalable:** Supports horizontal scaling.
- **Querying:** Powerful and expressive queries.

Example in SQL:

```
INSERT INTO users (name, age) VALUES (’Alice’, 25);
```

Example in MongoDB:

```
db.users.insertOne({ name: ’Alice’, age: 25 });
```

2 MongoDB vs PostgreSQL

Feature	MongoDB	PostgreSQL
Type	NoSQL (Document-based)	SQL (Relational)
Schema	Flexible / Schema-less	Strict / Schema-based
Query Language	JSON-like syntax	SQL
Joins	Limited support via $lookup	Full support
Transactions	ACID support since v4.0	Full ACID compliance
Indexing	Yes, including geospatial and text	Advanced indexing options
Scalability	Horizontal (sharding)	Mostly vertical
Best Use Case	Unstructured / rapidly evolving data	Structured / relational data

3 Recreating a PostgreSQL Schema in MongoDB

PostgreSQL Tables:

- `users`
- `books`
- `reading_list`

- reading_list_books
- reading_goals
- goal_progress

MongoDB Equivalents

users Collection

```
{
  "_id": ObjectId("..."),
  "first_name": "John",
  "last_name": "Doe",
  "email": "john@example.com",
  "password_hash": "...",
  "created_at": ISODate(),
  "updated_at": ISODate()
}
```

books Collection

```
{
  "_id": ObjectId("..."),
  "book_name": "Atomic␣Habits",
  "author_name": "James␣Clear",
  "publish_date": ISODate("2018-10-16"),
  "isbn": "9780735211292"
}
```

reading_lists with Embedded Books

```
{
  "_id": ObjectId("..."),
  "user_id": ObjectId("..."),
  "name": "Spring␣Reads",
  "books": [
    { "book_id": ObjectId("..."), "added_at": ISODate() }
  ],
  "created_at": ISODate(),
  "updated_at": ISODate()
}
```

reading_goals with Embedded Progress

```
{
  "_id": ObjectId("..."),
  "user_id": ObjectId("..."),
  "reading_list_id": ObjectId("..."),
  "duration_days": 30,
  "start_date": ISODate(),
  "deadline": ISODate("2025-05-10"),
  "created_at": ISODate(),
  "progress": [
    {
      "book_id": ObjectId("..."),
      "is_completed": false,
      "marked_at": ISODate()
    }
  ]
}
```

- We can use PostgreSQL for: users, books because they are shared, consistent, and benefit from strong constraints.

- We can use MongoDB for: reading_lists, reading_goals, goal_progress because they re user-scoped and can be embedded for fast access.