MongoDB

mongoDB®

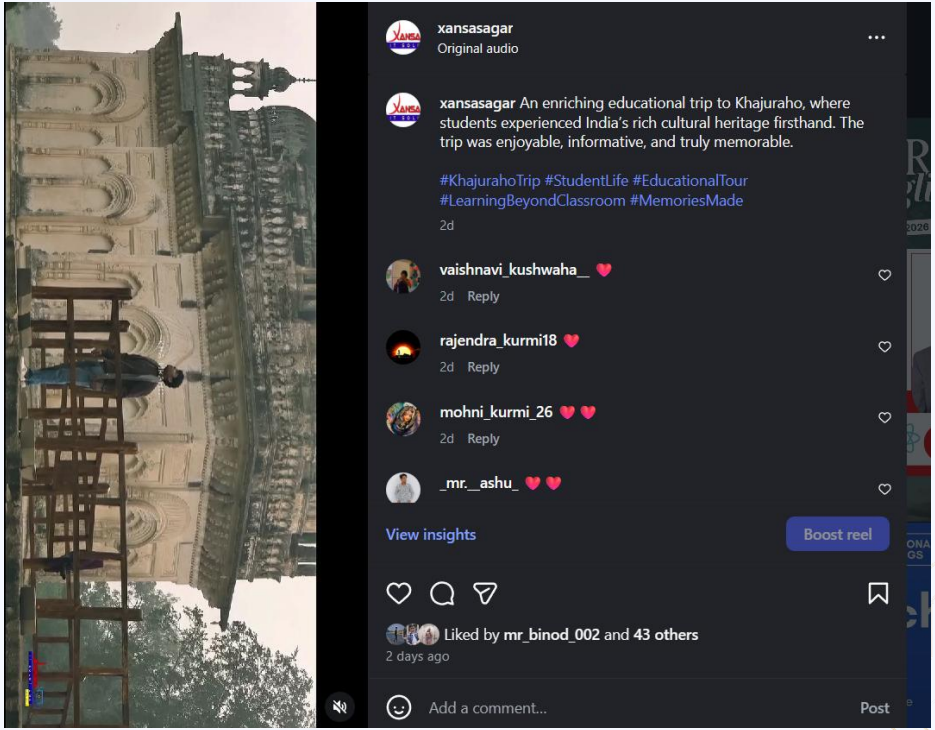
Session 1 – MongoDB Architecture & Core Concepts

XANSA IT SOLUTIONS
simplifying IT

Understanding How MongoDB Stores and Manages Data

The  use case

Consider this post



How RDBMS reads it?

```
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    display_name VARCHAR(100),
    profile_image VARCHAR(255)
);
```

```
CREATE TABLE media (
    media_id INT PRIMARY KEY AUTO_INCREMENT,
    post_id INT,
    media_type VARCHAR(20),
    media_url VARCHAR(255),
    audio VARCHAR(100),
    orientation VARCHAR(20),
    FOREIGN KEY (post_id) REFERENCES posts(post_id)
);
```

```
    post_id INT,
    hashtag_id INT,
    PRIMARY KEY (post_id, hashtag_id),
    FOREIGN KEY (post_id) REFERENCES posts(post_id),
    FOREIGN KEY (hashtag_id) REFERENCES hashtags(hashtag_id
```

```
CREATE TABLE posts (
    post_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    platform VARCHAR(20),
    post_type VARCHAR(20),
    caption TEXT,
    visibility VARCHAR(20),
    created_at DATETIME,
    updated_at DATETIME,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

```
CREATE TABLE hashtags (
    hashtag_id INT PRIMARY KEY AUTO_INCREMENT,
    hashtag VARCHAR(50) UNIQUE
);
```

How RDBMS reads it?

```sql
SELECT
    p.post_id,
    p.platform,
    p.post_type,
    p.visibility,
    p.created_at,

    -- Post Owner
    u.username
    u.display_name,

    -- Caption
    p.caption,

    -- Media
    m.media_type,
    m.media_url,
    m.audio,
    m.orientation,

    -- Hashtags
    h.hashtag,

    -- Likes
    l.username
    l.liked_at,

    -- Comments
    c.comment_id,
    c.username
    c.comment,
    c.commented_at
```

```sql
JOIN users u
    ON p.user_id = u.user_id

-- Media attached to post
JOIN media m
    ON p.post_id = m.post_id

-- Hashtags (many-to-many)
LEFT JOIN post_hashtags ph
    ON p.post_id = ph.post_id

LEFT JOIN hashtags h
    ON ph.hashtag_id = h.hashtag_id

-- Likes (one-to-many)
LEFT JOIN likes l
    ON p.post_id = l.post_id

-- Comments (one-to-many)
LEFT JOIN comments c
    ON p.post_id = c.post_id

WHERE p.post_id = 1

ORDER BY
    p.created_at DESC,
    c.commented_at ASC;
```

Enter MongoDB

How MongoDB reads it?


```
{
  "_id": ObjectId("65b8f2c9e1a9a00123abcd01"),

  "platform": "Instagram",
  "postType": "reel",

  "author": {
    "username": "xansasagar",
    "displayName": "Xansa IT Solutions",
    "profileImage": "xansa_logo.png"
  },

  "caption": "An enriching educational trip to Khajuraho, where students experienced India's rich

  "hashtags": [
    "KhajurahoTrip",
    "StudentLife",
    "EducationalTour",
    "LearningBeyondClassroom",
    "MemoriesMade"
  ],

  "media": {
    "type": "video",
    "url": "khajuraho_reel.mp4",
    "audio": "Original audio",
    "orientation": "vertical"
  },
```

MongoDB Ecosystem

- MongoDB Server
    - Core database engine responsible for storing, retrieving, and managing data securely
- MongoDB Shell (mongosh)
    - Command-line interface used to execute queries, manage databases, and perform administrative tasks
- MongoDB Compass
    - Graphical tool that allows users to visually explore databases, collections, documents, and indexes

MongoDB Ecosystem

- MongoDB Atlas
  - Fully managed cloud MongoDB service providing scalability, security, backups, and global access
- MongoDB Drivers
  - Language-specific libraries (Java, Node.js, PHP, Python, etc.) that enable applications to communicate with MongoDB

Database in MongoDB

- A database is a logical container used to organize collections.
- Each MongoDB server can host multiple databases.
- A database groups related data for a specific application.
- Databases do not store data directly
- Data is stored inside collections within a database

Collection in MongoDB

- A collection is a group of related documents
- Collections are similar to tables in relational databases
- A collection exists inside a database
- Collections do not require a fixed structure or predefined schema
- Documents within a collection can have different fields

Document in MongoDB

- A document is the basic unit of data in MongoDB
- Documents are stored in BSON format
- Data is represented as key–value pairs
- Each document represents a single record
- Documents within the same collection can have different structures

```
{
"_id": ObjectId("..."),
"name": "Rahul",
"course": "MongoDB",
"fees": 4000
}
```

Example of a document:

Fields and _id

- Fields are individual data elements within a document
- Each field consists of a key and a value
- The _id field is a unique identifier for every document
- MongoDB automatically generates the _id field if not provided
- The _id field acts as the primary key of a document
- No two documents in a collection can have the same _id

Schema Flexibility in MongoDB

- MongoDB is schema-less, which means documents in a collection can have different structures
- Fields can be added, removed, or modified without affecting other documents
- This flexibility allows easy adaptation to changing requirements
- Example:
  - Student A → { "name", "course" }
  - Student B → { "name", "course", "email", "phone" }
- Flexible schema eliminates the need for complex ALTER TABLE operations
- Ideal for applications where data structure evolves over time

Embedded Documents and Arrays

- Documents in MongoDB can contain nested documents (embedded documents) and arrays
- Embedded documents allow storing related data within a single document
- Arrays can store multiple values for a single field

Embedded Documents and Arrays

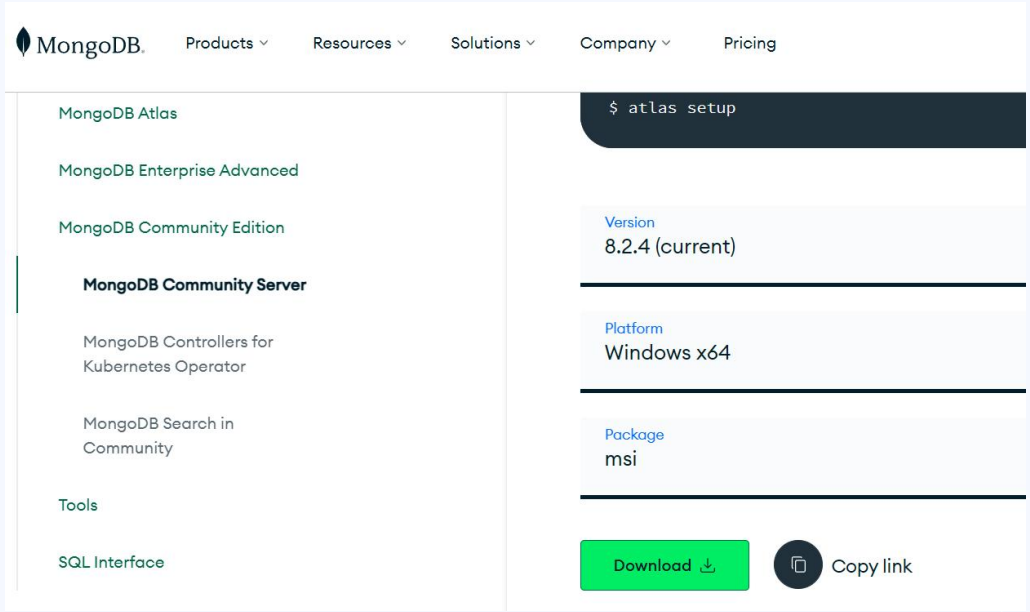
- Example:
- Benefits
  - Related data is stored together
  - Reduces the need for joins
  - Improves read performance

```
{
  "name": "Rahul",
  "courses": ["MongoDB", "Java", "Python"],
  "address": {
    "city": "Delhi",
    "zip": "110001"
  }
}
```

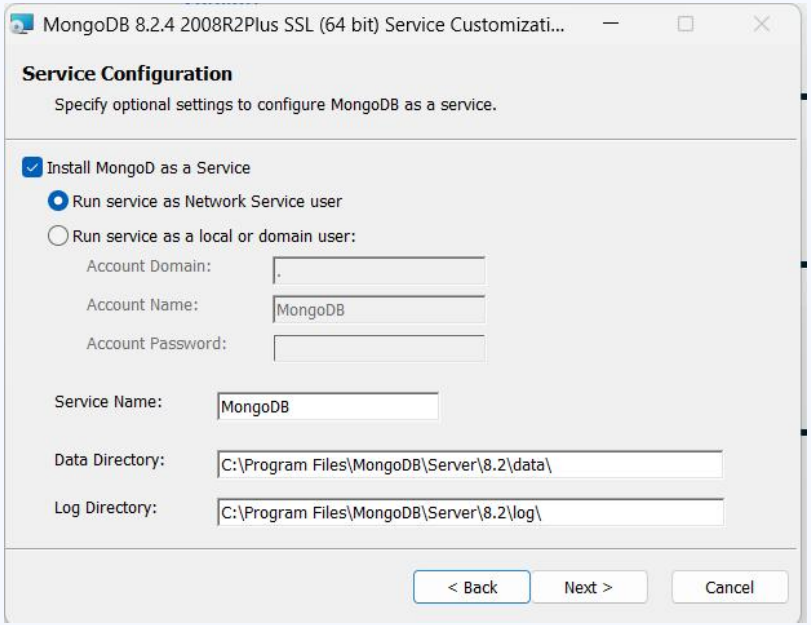
Installation

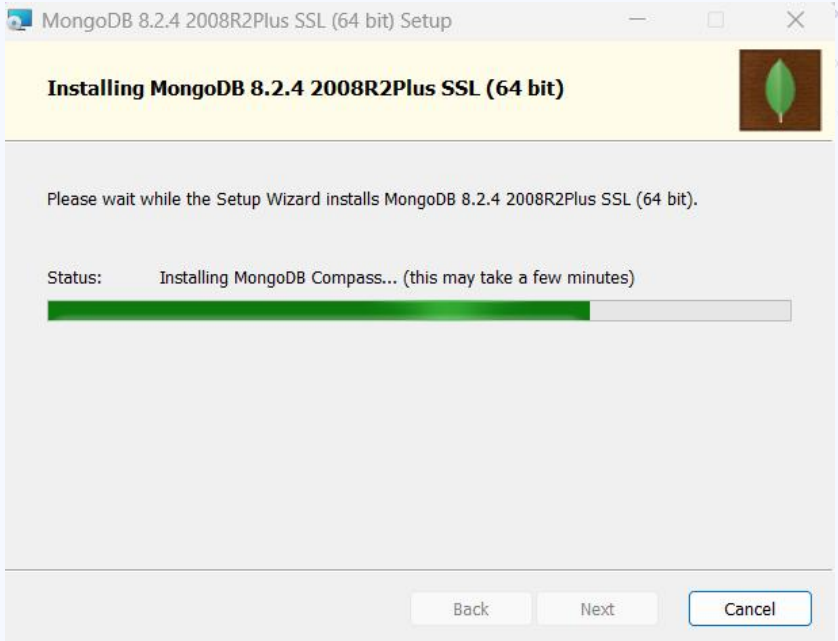
Download & Install MongoDB Community Edition

- 1. Open https://www.mongodb.com/try/download/community



Download & Install MongoDB Community Edition

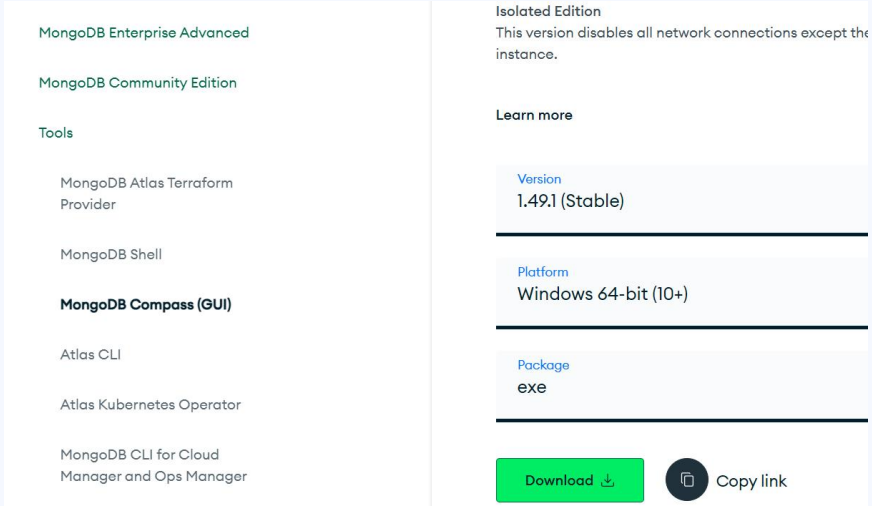




XANSA IT SOLUTIONS
simplifying IT

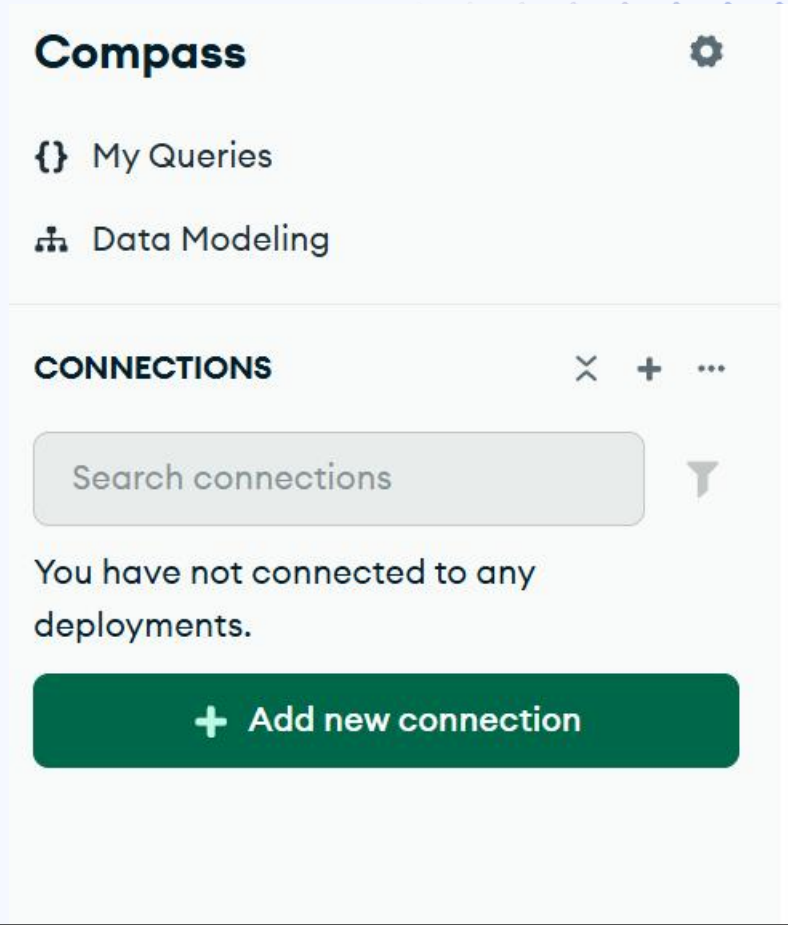
Download & Install MongoDB Compass (If Required)

- 1. Open https://www.mongodb.com/try/download/compass



Compass

- Create a new connection



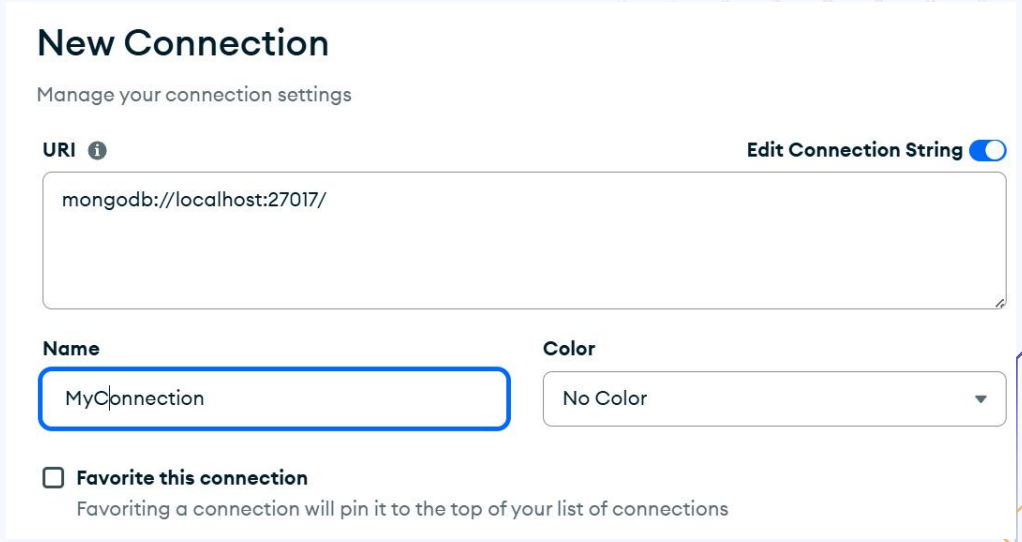
Create a connection

mongodb:// → MongoDB protocol

localhost → MongoDB is running on your own computer

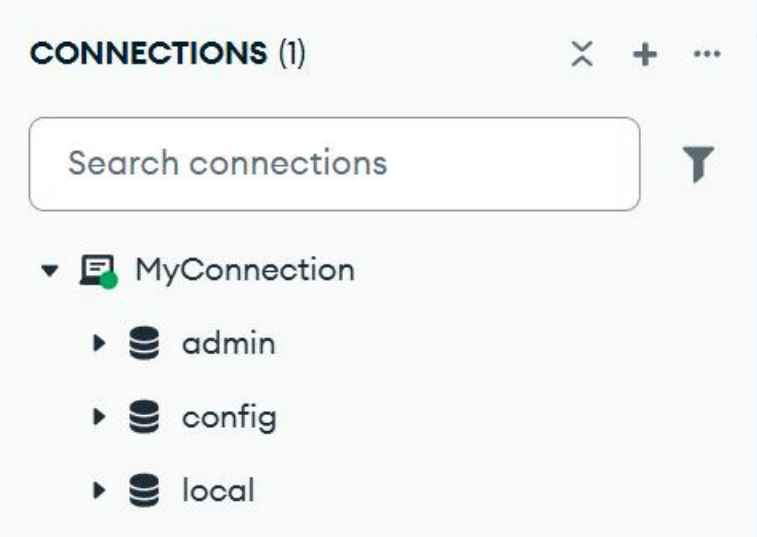
27017 → Default MongoDB port

Connection name



Inbuilt/Default Databases

- admin
  - Authentication & authorization
  - User and role management
  - Administrative commands
- local
  - Stores instance-specific data
  - Data is NOT replicated
- config
  - Stores sharding and cluster metadata
  - Used mainly in sharded clusters



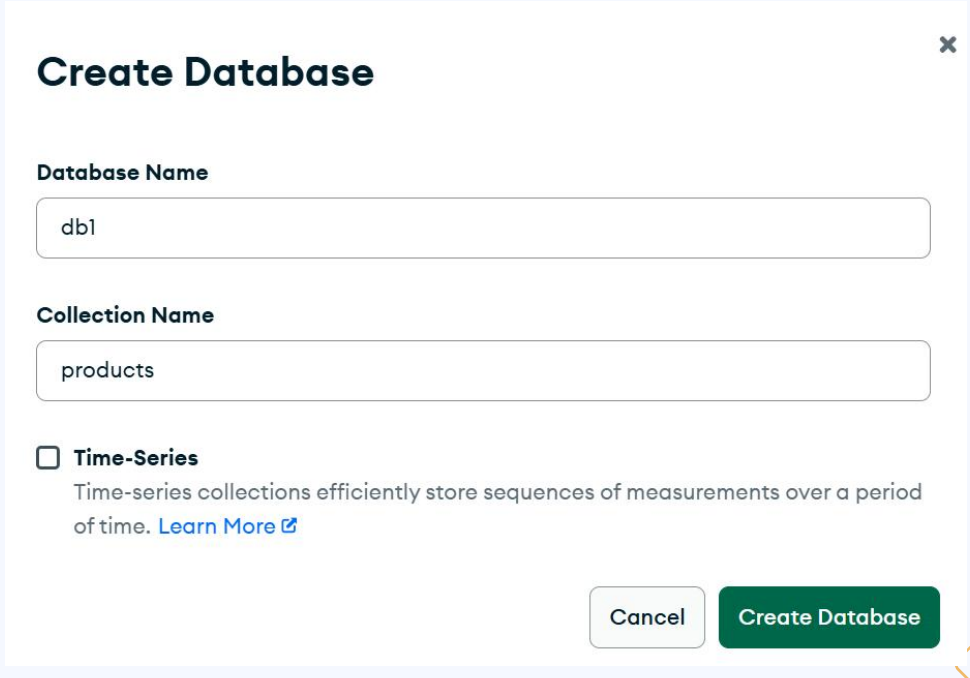
These databases:

- Are created automatically
- Should NOT be modified manually
- Are managed internally by MongoDB

Create Database

- Name of the database
- Name of the collection

(MongoDB requires at least one collection to create a database)



Thank You