hochschule hof
University of Applied Sciences

Non-Relational Databases

MongoDB --- 1

Prof. Dr. Jürgen Heym

Hof University of Applied Sciences

1 – MongoDB – Introduction

hochschule hof
University of Applied Sciences

- Important questions

  Which database/datastore or which combination of databases/datastores may solve a given problem?

- Q 1: *What is the type of the datastore?*
- Q 2: *What was the motor for the development of a given datastore?*
- Q 3: *How do we talk to the datastore?*
- Q 4: *What makes a datastore unique?*
- Q 5: *What kind of solutions provides a datastore?*
- Q 6: *How scales a datastore?*

1 – MongoDB – Introduction

hochschule hof
University of Applied Sciences

- Q 1: *What is the type of the datastore?*

• relational	(PostgreSQL)
• Key / value	(Riak, Redis, Hazelcast)
• column-oriented	(Hbase)
• document-oriented	(MongoDB, CouchDB)
• graph-based	(Neo4J)

1 – MongoDB – Introduction

hochschule hof
University of Applied Sciences

- Q 6: *How scales a datastore?*

  - Scaling features are in tight connection to performance of the datastore!
  - Scaling features need a context!

  - Horizontal scaling features (MongoDB, Hbase, Riak)
  - Vertical scaling features (PostgreSQL, Neo4J, Redis, HazelCast)
  - Mixed scaling features

1 – MongoDB – Introduction

hochschule hof
University of Applied Sciences

- DBMS Types
  - Document oriented Datastore
    - Document == Hash
    - Documents may contain nested structures
    - Flexibility permits variable spaces within documents
    - Differences in indexing strategies, ad-hoc queries, replication, consistency, …
    - Examples: MongoDB, CouchDB, …

1 – MongoDB – Introduction

hochschule hof
University of Applied Sciences

- MongoDB --- Characteristics
  - Derived from „humongous“ == enormous, giant, vast
  - Programming language C++
  - Leading NRD-Datastore
  - Open-Source
  - Document oriented

- MongoDB --- Features
  - Document-Oriented Storage
    - JSON-style documents with dynamic schemas offer simplicity and power.
  - Full Index Support
    - Index on any attribute.
  - Replication & High Availablity
    - Mirror across LANs and WANs for scale and peace of mind.

1 – MongoDB – Introduction

hochschule hof
University of Applied Sciences

- MongoDB --- Features
  - Auto-Sharding (Auto-Splitting)
    - Scale horizontally without compromising functionality.
  - Querying
    - Rich, document-based queries.
  - Fast In-Place Updates
    - Atomic modifiers for contention-free performance.
  - Map / Reduce
    - Flexible aggregation and data processing.
  - GridFS
    - Store files of any size without complicating your stack.
  - MongoDB Management Service (MMS)
    - The best way to run MongoDB in production. Secured. Supported. Certified.
  - Production Support

2 – MongoDB – Use-Cases

hochschule hof
University of Applied Sciences

- Use-Case: catalog based applications
  - Catalogs change permanently. For RDBMSs this is a nightmare!
  - Example: Online-Shops

Catalog	MongoDB
Stuck. You need to add new data and metadata to your catalogs, including unstructured and semi-structured data. Relational databases can't easily handle this data, which leaves you hamstrung.	**Do the Impossible.** MongoDB can incorporate any product, entity, attribute or any other type of data, no matter what it looks like. And It does so while preserving all the functionality needed to build a powerful application, including advanced features like recommendations.
Stagnant. RDBMS technology leaves teams impotent to do what the business asks. Instead, they wrestle with data modeling, transformation and schema problems for months, or even years.	**Faster.** Your teams move faster with MongoDB because its dynamic schemas let them iterate. They spend less time figuring out how to accommodate new entities or attributes, and instead ship new products in weeks.
$$$$. Large teams tied up for long periods of time make RDBMS catalog applications expensive to build and maintain. Proprietary software and hardware add to the cost. The business case becomes hard for you to justify.	**$$.** More productive teams, plus commodity hardware make your projects cost 10% what they would with a relational database.

3 – MongoDB – Installation

hochschule hof
University of Applied Sciences

- Ubuntu (22.04 LTS)
  - Official MongoDB packages are more recent than Ubuntu packages!
  - Needed packages
    - Mongodb-org — Meta-Package, installs all other packages
    - mongodb-org-database — MongoDB server binaries, e.g. amd64
    - mongodb-org-server — MongoDB database server
    - mongodb-org-shell — MongoDB shell client
    - mongodb-org-tools — MongoDB tools
    - Mongodb-org-mongos — MongoDB shared cluster query router
    - … and some more tools packages ….
  - Configuration files and scripts (Upstart-Systematic)
    - /etc/mongodb.conf — configuration options
      https://www.mongodb.com/docs/manual/reference/configuration-options/
    - sudo systemctl [start|stop|status|restart|enable|disable] mongod

3 – MongoDB – Installation

hochschule hof
University of Applied Sciences

- MongoDB installation

In order to install MongoDB on Ubuntu 22.04 we follow the official MongoDB how-to:

https://www.mongodb.com/community/forums/t/how-to-install-mongodb-6-0-on-ubuntu-22-04/

4 – MongoDB – Definitions

hochschule hof
University of Applied Sciences

- ## Database

  Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

- ## Collection

  Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

- ## Document

  A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

4 – MongoDB – Definitions

hochschule hof
University of Applied Sciences

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)
Database Server and Client	
mysqld/Oracle	mongod
mysql/sqlplus	mongo

4 – MongoDB – Sample Document

hochschule hof
University of Applied Sciences

Sample Blog document and ist structure: komma seperated key value pairs

```
{
   _id: ObjectId(7df78ad8902c)
   title: 'MongoDB Overview',
   description: 'MongoDB is no sql database',
   by: 'tutorials point',
   url: 'http://www.tutorialspoint.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 100,
   comments: [
      {
         user:'user1',
         message: 'My first comment',
         dateCreated: new Date(2011,1,20,2,15),
         like: 0
      },
      {
         user:'user2',
         message: 'My second comments',
         dateCreated: new Date(2011,1,25,7,45),
         like: 5
      }
   ]
}
```

5 – MongoDB – Data-Modelling

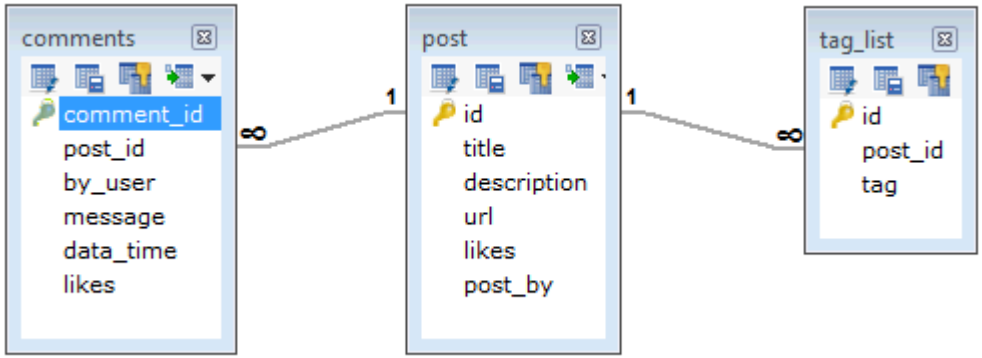
hochschule hof
University of Applied Sciences

- MongoDB Schemas
  - Data have within MongoDB a flexible schema!
  - Documents of a collection do not need the same set of fields or the same structure!
  - Same fields in documents of a collection may take different data types!
- MongoDB Design aspects
  - Develop the MongoDB design based on user and system requirements!
  - Combine object within a document, if you want to use them together. Otherwise separate them! Make sure that you don‘t need joins!
  - If necessary duplicate data in a limited way. RAM is cheap compared to CPU time!
  - Consider joins at write operations, not at read operations!
  - Optimize your schema concerning you key application!
  - Consider complex aggregations already within your schema!

5 – MongoDB – Data-Modelling

hochschule hof
University of Applied Sciences

- Example (RDBMS vs. MongoDB)
  - Your client needs a datastore for his blog website.
  - Requirements are:
    - Each blog entry gets a unique title, a description and an URL.
    - Each blog entry may be classified in one or more categories (tags).
    - Each blog entry has an author and a likes counter.
    - Each blog entry may have comments or not that are stored together with their author, timestamp and likes counter.
  - RDBMS schema



3 – MongoDB – Data-Modelling

hochschule hof
University of Applied Sciences

- Example (RDBMS vs. MongoDB)
  - MongoDB schema holds only one collection!

```
{
        _id: POST_ID
        title: TITLE_OF_POST,
        description: POST_DESCRIPTION,
        by: POST_BY,
        url: URL_OF_POST,
        tags: [TAG1, TAG2, TAG3],
        likes: TOTAL_LIKES,
        comments: [
           {
              user:'COMMENT_BY',
              message: TEXT,
              dateCreated: DATE_TIME,
              like: LIKES

           },
           {
              user:'COMMENT_BY',
              message: TEXT,
              dateCreated: DATE_TIME,
              like: LIKES

           }
        ]
}
```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DDL == Data Definition Language
  - Databases
    - Create database `use DATABASE_NAME`
    - Current database `db`
    - List databases `show dbs`
    - Databases are created as soon as you insert the first document.
    - Delete a database `db.dropDatabase()`

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DDL == Data Definition Language

  - Collections
    - Create a collection `db.createCollection(NAME, OPTIONS)`
    - Example:
```
db.createCollection("mycol",
    { capped : true, size : 6142800, max : 10000 })
```

    - In mongodb you don't need to create collection.
      MongoDB creates collections automatically, when you insert the first document.

    - Drop a collection `db.COLLECTION_NAME.drop()`
    - The drop() method will return true, if the selected collection is dropped successfully otherwise it will return false.

Parameter	Type	Description
Name	String	Name of the collection to be created
Options	Document	(Optional) Specify options about memory size and indexing

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DDL == Data Definition Language
  - Collection Options

Field	Type	Description
capped	Boolean	(Optional) If true, enables a capped collection. A capped collection is a size limited collection that automatically overwrites its oldest entries when it reaches its maximum size. **If you specify true, you need to specify the size parameter also.**
autoIndexID	Boolean	(Optional) If true, automatically create index on _id fields. Default is false.
size	number	(Optional) Specifies a maximum size in bytes for a capped collection.
max	number	(Optional) Specifies the maximum number of documents allowed in the capped collection.

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DDL == Data Definition Language

  - Indexes
    - Indexes support the efficient resolution of queries. Without indexes, MongoDB must scan every document of a collection to select those documents that match the query statement. This scan is highly inefficient and require the mongod to process eventually a large volume of data.
    - Indexes are special data structures, that store a small portion of the data set in an easy to traverse form. The index stores the value of a specific field or set of fields, ordered by the value of the field as specified in index.
    - To create an index you need to use ensureIndex() method of mongodb.
    - Ascending index: 1, descending index: -1
    - Syntax

      ```
      db.col.ensureIndex({KEY:1})
      ```

    - Example

      ```
      db.col.ensureIndex({"title":1,"description":-1})
      ```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DDL == Data Definition Language

  - Index Options (1)

Parameter	Type	Description
background	boolean	Builds the index in the background so that building an index does not block other database activities. Specify true to build in the background. The default value is **false**.
unique	boolean	Creates a unique index so that the collection will not accept insertion of documents where the index key or keys match an existing value in the index. Specify true to create a unique index. The default value is **false**.
name	string	The name of the index. If unspecified, MongoDB generates an index name by concatenating the names of the indexed fields and the sort order.
dropDups	boolean	Creates a unique index on a field that may have duplicates. MongoDB indexes only the first occurrence of a key and removes all documents from the collection that contain subsequent occurrences of that key. Specify true to create unique index. The default value is **false**.
sparse	boolean	If true, the index only references documents with the specified field. These indexes use less space but behave differently in some situations (particularly sorts). The default value is **false**.

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DDL == Data Definition Language

  - Index Options (2)

Parameter	Type	Description
expireAfterSeconds	integer	Specifies a value, in seconds, as a TTL to control how long MongoDB retains documents in this collection.
v	index version	The index version number. The default index version depends on the version of mongod running when creating the index.
weights	document	The weight is a number ranging from 1 to 99,999 and denotes the significance of the field relative to the other indexed fields in terms of the score.
default_language	string	For a text index, the language that determines the list of stop words and the rules for the stemmer and tokenizer. The default value is **english**.
language_override	string	For a text index, specify the name of the field in the document that contains, the language to override the default language. The default value is language.

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- MongoDB Data Types

Datatype	Description
String	This is most commonly used data type to store the data. String in mongodb must be UTF-8 valid.
Integer	This type is used to store a numerical value. Integer can be 32 bit or 64 bit depending upon your server.
Boolean	This type is used to store a boolean (true/ false) value.
Double	This type is used to store floating point values.
Min / Max keys	This type is used to compare a value against the lowest and highest BSON elements.
Arrays	This type is used to store arrays or list or multiple values into one key.
Timestamp	Ctimestamp. This can be handy for recording when a document has been modified or added.
Object	This data type is used for embedded documents.
Null	This type is used to store a Null value.

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- MongoDB Data Types

Datatype	Description
Date	This data type is used to store the current date or time in UNIX time format. You can specify your own date time by creating an object of Date and passing day, month, year into it.
Object ID	This data type is used to store the document's ID.
Binary data	This data type is used to store binary data.
Code	This data type is used to store JavaScript code into a document.
Regular expression	This data type is used to store regular expressions.

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DML == Data Manipulation Language

  - Insert Data
    - To insert data into a MongoDB collection, you need MongoDB's **insert()** or **save()** method:

```
db.COLLECTION_NAME.insertOne(document)
db.COLLECTION_NAME.insertMany([doc1,doc2, ...])
db.COLLECTION_NAME.replaceOne(filter,document)
```

    - If the collection doesn't exist in the database, MongoDB will create this collection and then insert the document the collection.
    - If we don't specify the _id parameter, then MongoDB assigns an unique ObjectId for this document.
    - The ObjectId (_id) is a 12 bytes hexadecimal number, unique for every document in a collection: 4 bytes timestamp, 3 bytes machine id, 2 bytes process id, 3 bytes incrementer.
    - If you specify the _id-Parameter and you use the method **replaceOne()**, then the document will be replaced.

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DML == Data Manipulation Language
  - Replace a document
    - Example

```
db.books.replaceOne(
{
        _id: ObjectId(7df78ad8902c)
},
{
        _id: ObjectId(7df78ad8902c),
        title: 'MongoDB Overview',
        description: 'MongoDB is no sql database',
        tags: ['mongodb', 'database', 'NoSQL'],
        likes: 100
})
```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- DML == Data Manipulation Language
  - Insert multiple documents
    - Example

```
db.books.insertMany([
{
        title: 'MongoDB Overview',
        description: 'MongoDB is no sql database',
        tags: ['mongodb', 'database', 'NoSQL'],
        likes: 100
},{
        title: 'NoSQL Database',
        description: 'NoSQL databases don't have tables',
        tags: ['mongodb', 'database', 'NoSQL'],
        likes: 20
}
])
```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Querying the database

  - Basic syntax of method **find()**

    ```
    db.COLLECTION_NAME.find()
    ```

  - Basic syntax for formatting the output with method pretty(). This changes the output only in mongo not in mongosh!

    ```
    db.COLLECTION_NAME.find().pretty()
    ```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Querying the database
  - Comparison Operators

Operation	Syntax	Example	RDBMS Equivalent
Equality	{<key>:<value>}	`db.col.find({"by":"tutorials"})`	where by = 'tutorials'
Less Than	{<key>:{$lt:<value>}}	`db.col.find({"likes":{$lt:50}})`	where likes < 50
Less Than Equals	{<key>:{$lte:<value>}}	`db.col.find({"likes":{$lte:50}})`	where likes <= 50
Greater Than	{<key>:{$gt:<value>}}	`db.col.find({"likes":{$gt:50}})`	where likes > 50
Greater Than Equals	{<key>:{$gte:<value>}}	`db.col.find({"likes":{$gte:50}})`	where likes >= 50
Not Equals	{<key>:{$ne:<value>}}	`db.col.find({"likes":{$ne:50}})`	where likes != 50

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Querying the database

  - Logical AND
    - If you pass in the **find()** method multiple keys by separating them by ',' then MongoDB treats it as an **AND** condition.
    - Example

```
db.col.find({key1:value1, key2:value2})
```

  - Logical OR
    - Syntax:

```
db.col.find( { $or: [ {key1: value1}, {key2:value2} ] } )
```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Exercise

1. Create a MongoDB database <<your Username>>_library
2. Create within your database a collection books.
3. Insert three documents in your collection.
4. Use the find() method to search your documents.
5. Develop different use cases that use all the comparison operators and logical operators.

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Updating Documents

  - MongoDB's **findOneAndUpdate()** method isused to update documents in a collection.

    Syntax:

    ```
    db.col.findOneAndUpdate(filter,update,options)
    ```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Updating Documents

  - Example

```
db.col.find()
{ "_id" : ObjectId(5983548781331adf45ec5), "title":"MongoDB Overview"}
{ "_id" : ObjectId(5983548781331adf45ec6), "title":"NoSQL Overview"}
{ "_id" : ObjectId(5983548781331adf45ec7), "title":"Tutorials Point Overview"}

db.books.findOneAndUpdate({_id:
ObjectId("5983548781331adf45ec5")},{$set:{likes:5}});

db.col.find()
{ "_id" : ObjectId(5983548781331adf45ec5),
  "likes":5, "title":"MongoDB Overview"}
{ "_id" : ObjectId(5983548781331adf45ec6), "title":"NoSQL Overview"}
{ "_id" : ObjectId(5983548781331adf45ec7), "title":"Tutorials Point Overview"}
```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Replacing Documents

  - The **findOneAndReplace()** method replaces the existing document with the new document passed in the method.

    Basic syntax of mongodb **findOneAndReplace()** method is shown below:

    - Syntax:

      ```
      db.col.findOneAndReplace(filter,replacement,options)
      ```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Deleting Documents

  - MongoDB's **deleteOne()** and **deleteMany()** methods are used to delete documents from a collection. Both methods accept the same parameters. One is the deletion criteria and the second is the optional write-concern.

    - Syntax:

      `db.col.deleteMany(filter,options)` (delete multiple docs)

      or

      `db.col.deleteOne(filter,options)` (delete one doc)

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Querying the database (continued)

  - Projections

    Basic Syntax: show key1, key2, … but don't show the ObjectId.

    ```
    db.col.find({SELECTION_CRITERIA},{KEY1:1,...,_id:0})
    ```

    Example

    ```
    {"_id" : ObjectId(5983548781331adf45ec5), "title":"MongoDB Overview"}
    {"_id" : ObjectId(5983548781331adf45ec6), "title":"NoSQL Overview"}
    {"_id" : ObjectId(5983548781331adf45ec7), "title":"Tutorials Point Overview"}

    db.col.find({},{"title":1,_id:0})

    {"title":"New MongoDB Tutorial"}
    {"title":"NoSQL Overview"}
    {"title":"Tutorials Point Point Overview "}
    ```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Querying the database (continued)
  - Limitations

    Use the limit() method to limit the number of shown documents.

    ```
    db.col.find().limit(NUMBER)
    ```

  - Skipping documents

    Use the skip() method to skip a certain number of documents in the result set.

    ```
    db.col.find().limit(NUMBER).skip(NUMBER2)
    ```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Querying the database (continued)

  - Sorting the result set

    To sort documents in MongoDB, you need to use **sort()** method.

    The **sort()** method accepts a document containing a list of fields along with their sorting order. To specify sorting order 1 and -1 are used. 1 is used for ascending order while -1 is used for descending order.

```
db.col.find({},{"title":1,_id:0}).sort({"title":-1})
```

5 – MongoDB – DDL / DML

hochschule hof
University of Applied Sciences

- Exercise (part 2)

1. Change individual attributes in different documents.
2. Change one attribute in multiple documents at once.
3. Replace a document with a new one.
4. Verify the outcome of all different methods for deleting a document.
5. Practice the projection of different attributes.
6. Limit the output using method limit().
7. Verfy the influence of method skip().
8. Change the sort sequence of search queries.

Literature

hochschule hof
University of Applied Sciences

- Seven NoSQL Datbases in a Week
  Sudarshan Kadambi; Xun Wu; Aaron Ploetz; Devram Kandhare, Oreilly®
  ISBN 978-1-78728-886-7
- MongoDB Inc.
  https://www.mongodb.com/
- MongoDB ORG
  https://www.mongodb.org/
- MongoDB Tutorial
  http://www.tutorialspoint.com/mongodb/index.htm
- MongoDB Configuration options
  http://docs.mongodb.org/manual/reference/configuration-options/