Lecture 9

Type	Lecture

KUTAISI INTERNATIONAL UNIVERSITY

Databases II

Example of a MongoDB document

```
{
  "_id" : ObjectId("53e3663ccb3bd259f9252f67"),
  "type" : [“accelerator", “maker-space“, “co-lab”],
  "name" : “Munich Urban Colab",
  "tags" : “munich colab start-up founder space",
  "desc" : “In the Munich Urban Colab starters and founders find the help they need.
                Equipped with …..",
  "address" : [
{ "street": "123 Fake Street", ”floor”: “15th”,  “pCode": "12345" ,"city": "Faketon"},
{ "street": "1 Some Other Street", “pCode": "12345" ,"city": "Boston"},
{   "str" : “Freddy-Mercury-Street 5",  “postalCode" : 80932,   “city" : “Munich"}
  ],
  "location" : {    "type" : "Point",
          "coordinates" : [ 7.0075, 51.45902 ]
  }
}
```

What structural elements of the document would not be possible in a relational (normalizd) table?

7

knirsch@htw-berlin.de

Relationship cardinalities:

1:1 - yes

1:N - yes

N:1 - No

N:M - No

KUTAISI INTERNATIONAL UNIVERSITY

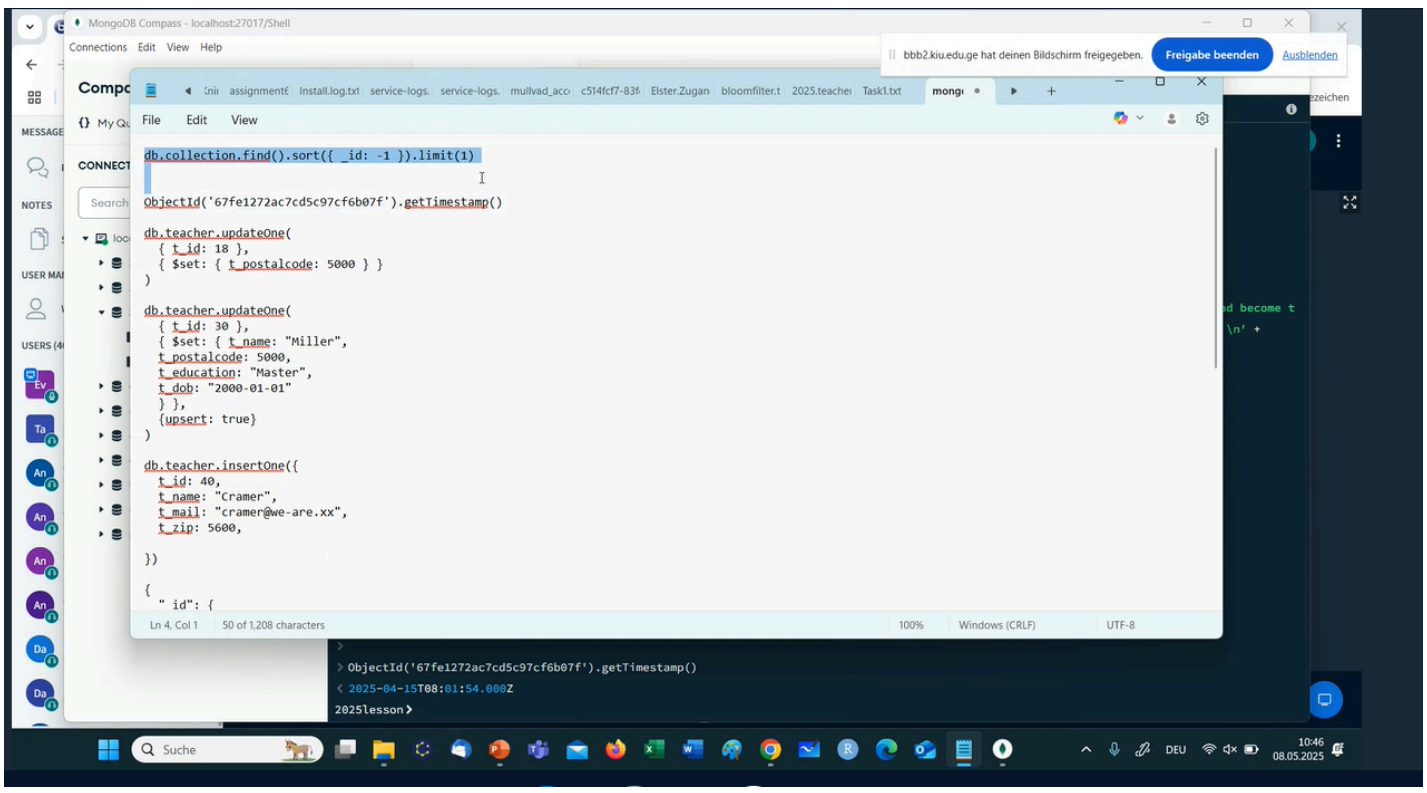
Databases II

Document databases

- Via tags the contents of the documents can be queried.
  → sophisticated and more complex queries are possible
- The "tags" provide the documents with a structure. However, there is no predefined schema. One could call it a flexible or dynamic schema, since each document can have different tags and each document can introduce new tags.
- The responsibility for modeling a data structure does not come with the database design but with the application.
- Unlike RDBMS, where data is distributed across different relations (to avoid anomalies and to map relationships), documents of a document database are self-contained and independent in terms of information content.
- The documents are not connected among each other, that is there is no relationship between documents.
- 1:1 and 1:N relationships are usually implemented within the document itself.

12

knirsch@htw-berlin.de



Embedded or Referenced Documents

Embedded documents
are stored inside a parent document. Whenever a parent document is retrieved, all its embedded documents are also retrieved.

student collection –
-> reference tacher collection

Referenced documents

- are stored in a separate collection. One can retrieve them separately.
- They reference each other with the help of the $ref notation:
  {"$ref": "collectionName1", "$id": ObjectId("89aba98c00a") }
- This reference is only notational!

MongoDB does not support the join concept of FK-PK links. So, if we want to retrieve a referenced document, we must effectively run 2 queries:

1. A first query reads the referenced ID value from one document and with this retrieved ID
2. A second query is needed to retrieve the data from the referenced document.

SQL versus Document Queries

Given: Relational PostgreSQL course database lessons and MongoDB collection teacher with embedded student documents

Return all teachers:
SQL: select * from teacher
MongoDB: db.teacher.find()

Return all students of teacher 1:
SQL: select distinct(username) from lesson where t_id = 1
MongoDB: db.teacher.find({t_id: 1}, {student.username: 1}) well supported

Return all teachers of student Donald:
SQL: select distinct(t_id) from lesson where usrrname= "Donald"
MongoDB not well supported

Syntax

1. Commands start with: db.
2. Followed by the name of the collection (to which the operation refers).

3. Then followed by the method()

Mongo DB basic commands

Command	Explanation
use <database name>	Switches to specified database, makes the specified database the current database or creates the database if it does not exist.
db	displays the name of the current database
show dbs	shows all databases that have at least one collection
db.help()	shows all commannds of MongoDB
db.createCollection(”Name”)	creates a collection (table) with thhe name “Name”
db.<collectionName>.isCapped()	shows if a collection is capped
show collections	Displays the collections of the current database
db.dropDatabase()	Deletes the current database (=deletes all collections from the database)
db.<collectionname>.drop()	deletes the collection
db.<collectionname>.countDocuments()	Count documents of a collection
db.<collectionname>.find({})	Display all documents of a collection

❗ Capped collections **have maximum size or document counts that prevent them from growing beyond maximum thresholds**. MongoDB removes older documents if a collection reaches the maximum size limit before it reaches the maximum document count.