

Non-Relational Databases

MongoDB --- 3

Prof. Dr. Jürgen Heym

Hof University of Applied Sciences

7 – MongoDB – DML – Update

■ Exercise (part 1)

1. Create a new database „TelecomContracts_ddmmyy“, where „ddmmyy“ is your birthday.
2. Create a collection „customers“ that will store address information about our customers.
3. Create a JavaScript function that allows for creating new customers comfortably: insertCustomer(cid,lastname,firstname,street,postcode,city).
4. Save function insertCustomer in file insertCustomer.js and load the file.
5. Insert three customer documents in your collection:

```
{
  "cid"       : "221-0159",
  "lastname"  : "Iakimenkova",
  "firstname" : "Irina",
  "street"    : "352, Birdland Road",
  "postcode"  : "95028",
  "city"      : "Paradise City"
}
```

7 – MongoDB – DML – Update

■ Updating Documents (Mongo version 3)

- MongoDB's version 3 **update()** method is used to update documents in a collection.

The update() method updates values in existing documents.

- Syntax:

```
db.col.update (SELECTION_CRITERIA, UPDATED_DATA)
```

- By default mongodb will update only a single document, to update multiple documents you need to set a parameter 'multi' to true.

Syntax:

```
db.col.update (SELECTION_CRITERIA, UPDATED_DATA, {multi:true})
```

7 – MongoDB – DML – Update

- Example (Mongo version 3)

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

db.col.update({'title':'Overview'},{$set: {'title':'New MongoDB Tutorial'}})

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

7 – MongoDB – DML – Update

- Updating Documents (Mongo version 6)
 - MongoDB's version 6 **updateOne()** method is used to update exactly one document in a collection. To update several documents use **updateMany()**.

The update() method updates values in existing documents.

- Syntax:

```
db.col.updateOne (SELECTION_CRITERIA, UPDATED_DATA)
```

- By default mongodb will update only a single document, to update multiple documents you need to set a parameter 'multi' to true.

Syntax:

```
db.col.updateMany (SELECTION_CRITERIA, UPDATED_DATA)
```

7 – MongoDB – DML – Update

■ Example (Mongo version 6)

```
db.inventory.updateOne(  
  { cid:"221-0159" },  
  {  
    $set: { address2: "Upper Floor" },  
    $currentDate: { lastModified: true }  
  }  
)
```

This update operation:

- uses the \$set operator to update the value of the size.uom field to "cm" and the value of the status field to "P", and
- uses the \$currentDate operator to update the value of the lastModified field to the current date. If lastModified field does not exist, \$currentDate will create the field. See \$currentDate for details.

7 – MongoDB – DML – Update

■ Example (Mongo version 6)

```
db.inventory.updateMany(  
  { "qty": { $lt: 50 } },  
  {  
    $set: { "size.uom": "in", status: "P" },  
    $currentDate: { lastModified: true }  
  }  
)
```

This update operation:

- Finds all documents with key “qty” lower than 50,
- uses the \$set operator to update the value of the size.uom field to "in" and the value of the status field to "P", and
- uses the \$currentDate operator to update the value of the lastModified field to the current date. If lastModified field does not exist, \$currentDate will create the field. See \$currentDate for details.

7 – MongoDB – DML – Update

- Exercise (part 2)
 1. Read the page about update of MongoDB v6 reference manual:
<https://www.mongodb.com/docs/manual/tutorial/update-documents/>
 2. Change the lastname attribute in one of your address documents.
Use the „_id“ attribute as the selection criterium.
 3. Control the result using the find() method. The output should show only the following attributes: cid, lastname, firstname.
 4. One of your addresses should have a second address attribute (address2) with an apartment information „Apartment 5-22“. Verify the updated document.
 5. Delete attribute address of one document and add an attribute „postbox“ with value 1602 to this document. Verify the updated document.

7 – MongoDB – DML – Update

- Solution (part2 for Mongo version 3)

2. `db.contracts.update({"_id":ObjectId(59...c5)},{ $set:{lastname:"Fox"}})`

3. `db.contracts.find({"_id":ObjectId(59...c5)},{
_id:0,cid:1,lastname:1,firstname:1})`

4. `db.contracts.update({"_id":ObjectId(59...c5)},{
$set:{address2:"Appartment 5-22"}})
db.contracts.find()`

5. `db.contracts.update({"_id":ObjectId(59...)},{ $unset:{address:1}})
db.contracts.find()
db.contracts.update({"_id":ObjectId(59...)},{ $set:{postbox:"1602"}})
db.contracts.find()`

7 – MongoDB – DML – Update

- Exercise (part 3)

Each customer may have zero, one or several contracts with different tariff and options.

For each tariff we create a new contract that may be canceled separately.

Contracts will be numbered sequentially and get a leading „C“, e.g. „C-001“.

Valid tariffs with their options are

Phone [Flat-DE, Flat-EU]

TV [HD-PLUS]

Internet [100, 64, 16, WLAN]

1. Use the update method to store a new contract for a customer with tariff phone, option FLAT-EU, and contract date 12/15/2016.
2. The same customer now signs a second contract with tariff Internet including several options Internet 100 & WLAN and today's date.
Use directive \$push !

7 – MongoDB – DML – Update

- Solution for Mongo version 3

```
1. db.contracts.update(
  {"_id":ObjectId(59...c5)},
  {$set:{ contracts:[
    {cnr:"C-001",tariffs:["Phone"],options:["Flat-DE"]}
  ]}
)

db.contracts.find().pretty()

2. db.contracts.update(
  {"_id":ObjectId(59...c5)},
  {$push:{ contracts:
    {cnr:"C-002",tariffs:["Internet"],options:["100","WLAN"]}
  }
)

db.contracts.find()
```

7 – MongoDB – DML – Update

- Challenge (part 4)

3. The third customer now signs two contracts including several tariffs Phone, TV and Internet with options Internet 100 & WLAN and today's date. Use directive \$push in connection with \$each!

Read the MongoDB reference pages for method update and the update operators:

<https://www.mongodb.com/docs/manual/tutorial/update-documents/>
<https://www.mongodb.com/docs/manual/reference/operator/update/>

4. Search and show all customers that have a contract with WLAN-Option.
5. For marketing purposes search and show all customers that have a contract with tariff Internet but without option WLAN.

Literature

- Sieben Wochen, sieben Datenbanken
Moderne Datenbanken und die NoSQL-Bewegung
E. Redmon & J. R. Wilson, O'Reilly®
ISBN 978-3-86899-791-0
- MongoDB Update-Methode
<http://docs.mongodb.org/manual/reference/method/db.collection.update/>
- MongoDB Update-Operatoren und –Modifikatoren
<http://docs.mongodb.org/manual/reference/operator/update/>