

Aufgabe 7a

Daten importieren

--db in diese Datenbank (falls nicht vorhanden wird erstellt).
--collection in diese Collection (falls nicht vorhanden wird erstellt).
--drop falls Collection besteht, wird diese entfernt.
--file Pfad der zu importierenden Datei

```
mongoimport --db inf-wpp-c2 --collection aufgabe7 --drop --file  
~/Desktop/plz.data.txt
```

Falls in dem Dokument eine ID `_id` angegeben ist, wird diese genutzt. Ansonsten wird eine ObjectID beim importieren erzeugt.

Aufgabe 7b und 7c

Wir haben unseren Code in Python geschrieben. Um die Vor- und Nachteile der unterschiedlichen Datenbanken in puncto Geschwindigkeit für den Nutzerbesser darzustellen (und ein wenig dazuzulernen) haben wir mittels Flask (ein Python web framework) eine GUI gebaut. Bei diesen Quellcode-Schnippseln handelt nur um den Verbindungsaufbau zur Datenbank und die Abfragen. Die Darstellung wird in den Schnippseln nicht berücksichtigt.

Verbindungsaufbau zur MongoDB

```
from pymongo import MongoClient  
  
client = MongoClient('mongo', 27017)  
db = client['inf-wpp-c2']  
collection = db['aufgabe7']
```

Aufgabe 7b

```
result = collection.find_one({'_id': postcode})  
city = result['city']  
state = result['state']
```

Aufgabe 7c

```
city = city.upper()  
postcodes = collection.find({'city': city})
```

Die LoC sind nicht relevant unterschiedlich. Die Arbeitszeit des Programmieres, geht man vom identischen Kenntnisstand aus, ist nahezu gleich. Die Abfragezeit nach dem Key ist sehr ähnlich. Die

Abfrage nach dem Inhalt des Values ist in MongoDB wesentlich schneller.

Aufgabe 8

Aufgabe 8 a

Anpassungen: - Syntax des Strings im Date Objekt - Nike j/n als boolean - ' ' bei Array hinzufügen
Korrekte insert Befehle:

```
db.fussball.insert({name: 'HSV', gruendung: new Date("1887-09-29"), farben:
['weiss', 'rot'], Tabellenplatz: 17, nike: false});
db.fussball.insert({name: 'Dortmund', gruendung: new Date("1909-12-19"), farben:
['gelb', 'schwarz'], Tabellenplatz: 16, nike: false});
db.fussball.insert({name: 'Schalke', gruendung: new Date("1904-05-04"), farben:
['blau'], Tabellenplatz: 15, nike: false});
db.fussball.insert({name: 'Paderborn', gruendung: new Date("1907-08-14"),
farben:['blau', 'weiss', ], Tabellenplatz:14, nike:false, });
db.fussball.insert({name: 'Hertha', gruendung: new Date("1892-07-25"), farben:
['blau', 'weiss'], Tabellenplatz: 13, nike: true});
db.fussball.insert({name: 'Augsburg', gruendung: new Date("1907-08-08"), farben:
['rot', 'weiss'], Tabellenplatz: 12, nike: true});
db.fussball.insert({name: 'Pauli', gruendung: new Date("1910-05-15"), farben:
['braun', 'weiss'], Tabellenplatz: 11, nike: false});
db.fussball.insert({name: 'Gladbach', gruendung: new Date("1900-08-01"), farben:
['schwarz', 'weiss', 'gruen'], Tabellenplatz: 10, nike: false});
db.fussball.insert({name: 'Frankfurt', gruendung: new Date("1899-03-08"), farben:
['rot', 'schwarz', 'weiss'], Tabellenplatz: 9, nike: true});
db.fussball.insert({name: 'Leverkusen', gruendung: new
Date("1904-11-20T16:15:00Z"), farben: ['rot', 'schwarz'], Tabellenplatz: 8, nike:
false});
db.fussball.insert({name: 'Stuttgart', gruendung: new Date("1893-09-09"), farben:
['rot', 'weiss'], Tabellenplatz: 7, nike: false});
db.fussball.insert({name: 'Werder', gruendung: new Date("1899-02-04"), farben:
['gruen','weiss'], Tabellenplatz: 6, nike: true});
```

Befehl zur Überprüfung:

```
db.fussball.count()
```

Sollte 12 ergeben.

Aufgabe 8 b)

1. mit Namen ,Augsburg' ``shell db.fussball.find({name:'Augsburg'})

```
2. alle Nike-Vereine, welche schwarz als mindestens eine Vereinsfarbe haben
``shell
db.fussball.find({ $and: [{farben: {$all:['schwarz']}},{nike:true}] })
```

```
db.fussball.find({ $and: [{farben: 'schwarz'}, {nike:true}] })
```

1. alle Nike-Vereine, welche weiss und grün als Vereinsfarbe haben

```
db.fussball.find({ $and: [{farben :['gruen','weiss']},{nike:true}] })
```

1. alle Nike-Vereine, welche weiss oder grün als Vereinsfarbe haben

```
db.fussball.find({ $and: [{$or: [{farben: 'weiss'}, {farben: 'gruen'}]},{nike:true}] })
```

1. den Verein mit dem höchsten Tabellenplatz

```
db.fussball.find().sort({Tabellenplatz:1}).limit(1)
```

1. alle Vereine, die nicht auf einem Abstiegsplatz stehen

```
db.fussball.find({Tabellenplatz: {$lt: 16}}).sort({Tabellenplatz:1})
```

Aufgabe 8 c)

```
db.fussball.find({name: 'HSV'}, {_id:0})
```

Aufgabe 8 d)

Augsburg verliert alle seine Attribute bis auf den Tabellenplatz 1. Geschieht dadurch, dass Parameter 2 das komplette Dokumente ersetzt.

Aufgabe 8 e)

Änderungen mittels Modifier. 1. Ändern sie den Tabellenplatz von Leverkusen auf 2

```
db.fussball.update({name:'Leverkusen'}, {$set:{Tabellenplatz:2}})
```

1. Werder soll um einen Tabellenplatz nach vorne gebracht werden

```
db.fussball.update({name:'Werder'}, {$inc:{Tabellenplatz:-1}})
```

1. Ergänzen sie für den HSV ein Attribut „abgestiegen“ mit einem sinnvollen Wert

```
db.fussball.update({name:'HSV'}, {$set:{abgestiegen: true}})
```

1. Ergänzen sie für alle Vereine, deren Vereinsfarbe weiss enthält, ein Attribut „Waschtemperatur“ mit dem Wert 90.

```
db.fussball.update({farben: 'weiss'}, {$set:{waschtemperatur: 90}}, false, true)
```

Aufgabe 9

Anleitung: <http://bbouille.github.io/2017/02/15/HBase-Docker-OSx/>

Starten des Docker Containers

```
id=$(docker run --name=hbase-docker --rm -h hbase-docker -d -v $PWD/data:/data  
dajobe/hbase)
```

Prüfen ob Container läuft

```
docker ps
```