WP Big Data

Aufgabe 7

Aufwand: geringer, da python code teilweise übernommen werden konnte
Das Laden von Daten dauert länger als bei einer Key-Value Datenbank. Im ganzen ist die Benutzung von redis und MogoDB angenehm leicht.

Zeitmessung:

	redis	MongoDB
insert	~0.2ms	~0.4ms
Get PLZ	~0.4ms	~20ms
Get City	~5.5ms	~3.3ms

Code:

Daten importieren:

```
import bson
from bson.json_util import loads
import pymongo
import time

try:
    client = pymongo.MongoClient("141.22.29.77",27017)
    print ("Connected")
except (pymongo.errors.ConnectionFailure):
    print ("ERROR")

fileData = open('plz.data','r')
logData = open('log.log','a')

def writeToFile(s):
    logData.write(str(s) + '\n')

db = client.plz_db
collection = db.plz_collection
collection.drop()
for line in fileData:
    jsonData = loads(line)
    start = time.time()
    collection.insert_one(jsonData)
    stop = time.time()
    writeToFile('insert: ' + str(stop-start))

print(client.name)
```

Anfragen verarbeiten:

```
import pymongo
import json
import time
```

```
client = pymongo.MongoClient("141.22.29.77",27017)

fileData = open('log.log','a')

def writeToFile(s):
    fileData.write(str(s) + '\n')

def getZipByCity(com):
    start = time.time()
    for item in client.plz_db.plz_collection.find( { "city" : com } ):
        stop = time.time()
        writeToFile('find_city: ' + str(stop-start))
        print (item.get('_id'))

def getCityStateByZip(com):
    start = time.time()
    for item in client.plz_db.plz_collection.find({"_id": com}):
        stop = time.time()
        writeToFile('find_plz: ' + str(stop-start))
        print(item.get('city') +', '+item.get('state'))

wFlag = True

def work():
    com = input('WONACH SOLL GESUCHT WERDEN? [ZIP/CITY] ')
    if com == 'ZIP':
        com = input('BITTE ZIP-NUMMER EINGEBEN: ')
        getCityStateByZip(com)
        print()
    elif com == 'CITY':
        com = input('BITTE EINEN STADT ANGEBEN: ')
        getZipByCity(com)


while wFlag:
    work()
```

Aufgabe 8

Code:

Daten imprtieren:

```
import pymongo
import bson
from bson.json_util import loads

try:
    client = pymongo.MongoClient("141.22.29.77",27017)
    print ("Connected")
except (pymongo.errors.ConnectionFailure):
    print ("ERROR")


db = client.vereine_db
db.fussball.drop()
fileData = open('sinndeslebens.data','r')
for line in fileData:
    print(line)
```

```
    jsonData = loads(line)
    db.fussball.insert_one(jsonData)
```

Abfragen:

```
import pymongo
from bson.json_util import loads

client = pymongo.MongoClient("141.22.29.77",27017)

def augsburg():
    for item in client.vereine_db.fussball.find({'name': "Augsburg"}):
        print(item)

#alle Nike-Vereine, welche schwarz als mindestens eine Vereinsfarbe haben
def nikeSchwarz():
    for item in client.vereine_db.fussball.find( {"nike" : 'j', "farben": {"$in":
['schwarz']}}):
        print(item)

#alle Nike-Vereine, welche weiss und gruen als Vereinsfarbe haben
def  nikeWeissGruen():
    for item in client.vereine_db.fussball.find( {'nike' : 'j', "$and":[{'farben'
: 'weiss'}, {'farben':'gruen'}]}):
        print(item)

# alle Nike-Vereine, welche weiss oder gruen als Vereinsfarbe haben
def nikeWeissOrgruen():
    for item in client.vereine_db.fussball.find({"nike": 'j', "$or": [{'farben' :
'weiss'}, {'farben' : 'gruen'}]}):
        print(item)

#den Verein mit dem hoechsten Tabellenplatz
def highestPlace():
        print(client.vereine_db.fussball.find_one(sort=[("tabellenplatz",+1)]))

#alle Vereine, die nicht auf einem Abstiegsplatz stehen
def keinAbstieg():
    for item in client.vereine_db.fussball.find({"tabellenplatz": {'$not' :
{"$gt": 15}}}):
        print(item)

def beliebigeAnfrage():
    print(client.vereine_db.fussball.find_one({'name': "HSV"},{'_id': False}))

def anderung():
    print(client.vereine_db.fussball.update({'name':"Augsburg"}, {'tabellenplatz'
: 1}))


def rettung():
    fileData = open('sinndeslebens.data','r')
    for line in fileData:
        jsonData = loads(line)
        if(jsonData["name"] == 'Augsburg'):
            client.vereine_db.fussball.update({'tabellenplatz' : 1},jsonData)

def aenderungLeverkusen():
    client.vereine_db.fussball.update({'name':"Leverkusen"}, {'$set'
:{'tabellenplatz' : 2}})
```

```
def aenderungWerder():
    client.vereine_db.fussball.update({'name' : "Werder"}, {'$inc' :
{'tabellenplatz' : -1}})

def aenderungHSV():
    client.vereine_db.fussball.update({'name':"HSV"}, {'$set' :{'abgestiegen' :
"j"}})

def aenderungTemp90():
    client.vereine_db.fussball.update({'farben':"weiss"}, {'$set'
:{'waschtemperatur' : 90}},multi=True)

print('Anfrage 1')
augsburg()
print('Anfrage 2')
nikeSchwarz()
print('Anfrage 3')
nikeWeissGruen()
print('Anfrage 4')
nikeWeissOrgruen()
print('Anfrage 5')
highestPlace()
print('Anfrage 6')
keinAbstieg()
print('Anfrage 7')
beliebigeAnfrage()
print('Aenderung')
anderung()
print('Rettung')
rettung()
print('Aenderung 1')
aenderungLeverkusen()
print('Aenderung 2')
aenderungWerder()
print('Aenderung 3')
aenderungHSV()
print('Aenderung 4')
aenderungTemp90()
for item in client.vereine_db.fussball.find():
    print(item)

Aenderung
{'updatedExisting': True, 'ok': 1, 'n': 1, 'nModified': 1}
```

Außerdem wird der Augsburg-Eintrag komplett überschrieben. Es steht nur noch:

```
{'_id': ObjectId('55674284a19901177043eec4'), 'tabellenplatz': 1}
```

Nach Ausführung aller Anfragen:

```
{'name': 'HSV', 'gruendung': '1887, 09, 29', 'farben': ['weiss', 'rot'],
'abgestiegen': 'j', 'nike': 'n', '_id': ObjectId('55674a1fa199011eb844037f'),
'tabellenplatz': 17, 'waschtemperatur': 90}

{'name': 'Dortmund', 'gruendung': '1909, 12, 19', 'farben': ['gelb', 'schwarz'],
'nike': 'n', '_id': ObjectId('55674a1fa199011eb8440380'), 'tabellenplatz': 16}
```

{'name': 'Schalke', 'gruendung': '1904, 5, 4', 'farben': ['blau'], 'nike': 'n',
'_id': ObjectId('55674a1fa199011eb8440381'), 'tabellenplatz': 15}

{'name': 'Paderborn', 'gruendung': '1907, 8, 14', 'farben': ['blau', 'weiss'],
'waschtemperatur': 90, 'nike': 'n', '_id': ObjectId('55674a1fa199011eb8440382'),
'tabellenplatz': 14}

{'name': 'Hertha', 'gruendung': '1892, 7, 25', 'farben': ['blau', 'weiss'],
'waschtemperatur': 90, 'nike': 'j', '_id': ObjectId('55674a1fa199011eb8440383'),
'tabellenplatz': 13}

{'name': 'Augsburg', 'gruendung': '1907, 8, 8', 'farben': ['rot', 'weiss'],
'waschtemperatur': 90, 'nike': 'j', '_id': ObjectId('55674a1fa199011eb8440384'),
'tabellenplatz': 12}

{'name': 'Pauli', 'gruendung': '1910, 5, 15', 'farben': ['braun', 'weiss'],
'waschtemperatur': 90, 'nike': 'n', '_id': ObjectId('55674a1fa199011eb8440385'),
'tabellenplatz': 11}

{'name': 'Gladbach', 'gruendung': '1900, 8,1', 'farben': ['schwarz', 'weiss',
'gruen'], 'waschtemperatur': 90, 'nike': 'n', '_id':
ObjectId('55674a1fa199011eb8440386'), 'tabellenplatz': 10}

{'name': 'Frankfurt', 'gruendung': '1899, 3, 8', 'farben': ['rot', 'schwarz',
'weiss'], 'waschtemperatur': 90, 'nike': 'j', '_id':
ObjectId('55674a1fa199011eb8440387'), 'tabellenplatz': 9}

{'name': 'Leverkusen', 'gruendung': '1904, 11, 20', 'farben': ['rot', 'schwarz'],
'nike': 'n', '_id': ObjectId('55674a1fa199011eb8440388'), 'tabellenplatz': 2}
{'name': 'Stuttgart', 'gruendung': '1893, 9, 9', 'farben': ['rot', 'weiss'],
'waschtemperatur': 90, 'nike': 'n', '_id': ObjectId('55674a1fa199011eb8440389'),
'tabellenplatz': 7}

{'name': 'Werder', 'gruendung': '1899,2,4', 'farben': ['gruen', 'weiss'],
'waschtemperatur': 90, 'nike': 'j', '_id': ObjectId('55674a1fa199011eb844038a'),
'tabellenplatz': 5}