

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:

	<u>redis</u>	<u>MongoDB</u>
<u>insert</u>	<u>~0.2ms</u>	<u>~0.4ms</u>
<u>Get PLZ</u>	<u>~0.4ms</u>	<u>~20ms</u>
<u>Get City</u>	<u>~5.5ms</u>	<u>~3.3ms</u>

Aufgabe 8

Abfragen:

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

> {'gruendung': '1907, 8, 8', 'tabellenplatz': 12, 'name': 'Augsburg', 'nike': 'j', 'farben': ['rot', 'weiss'], '_id':
ObjectId('55683dd1a199018114e8253a')}
```

#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)
```

```
> {'gruendung': '1899, 3, 8', 'tabellenplatz': 9, 'name': 'Frankfurt', 'nike': 'j', 'farben': ['rot', 'schwarz', 'weiss'], '_id':  
ObjectId('55683dd1a199018114e8253d')}
```

#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)
```

```
> {'gruendung': '1899,2,4', 'tabellenplatz': 6, 'name': 'Werder', 'nike': 'j', 'farben': ['gruen', 'weiss'], '_id':  
ObjectId('55683dd1a199018114e82540')}
```

alle Nike-Vereine, welche weiss oder gruen als Vereinsfarbe haben

```
def nikeWeissOgruen():  
    for item in client.vereine_db.fussball.find({"nike": 'j', "$or": [{"farben" : 'weiss'}, {'farben' : 'gruen'}]}):  
        print(item)
```

```
> {'gruendung': '1892, 7, 25', 'tabellenplatz': 13, 'name': 'Hertha', 'nike': 'j', 'farben': ['blau', 'weiss'], '_id':  
ObjectId('55683dd1a199018114e82539')}
```

```
{'gruendung': '1907, 8, 8', 'tabellenplatz': 12, 'name': 'Augsburg', 'nike': 'j', 'farben': ['rot', 'weiss'], '_id':  
ObjectId('55683dd1a199018114e8253a')}
```

```
{'gruendung': '1899, 3, 8', 'tabellenplatz': 9, 'name': 'Frankfurt', 'nike': 'j', 'farben': ['rot', 'schwarz', 'weiss'], '_id':  
ObjectId('55683dd1a199018114e8253d')}
```

```
{'gruendung': '1899,2,4', 'tabellenplatz': 6, 'name': 'Werder', 'nike': 'j', 'farben': ['gruen', 'weiss'], '_id':  
ObjectId('55683dd1a199018114e82540')}
```

#den Verein mit dem hoechsten Tabellenplatz

```
def highestPlace():  
    print(client.vereine_db.fussball.find_one(sort=[("tabellenplatz",+1)]))
```

```
> {'gruendung': '1899,2,4', 'tabellenplatz': 6, 'name': 'Werder', 'nike': 'j', 'farben': ['gruen', 'weiss'], '_id':  
ObjectId('55683dd1a199018114e82540')}
```

#alle Vereine, die nicht auf einem Abstiegsplatz stehen

```
def keinAbstieg():  
    for item in client.vereine_db.fussball.find({"tabellenplatz": {'$not' : {"$gt": 15}}}):  
        print(item)
```

```

> {'gruendung': '1904, 5, 4', 'tabellenplatz': 15, 'name': 'Schalke', 'nike': 'n', 'farben': ['blau'], '_id':
ObjectId('55683dd1a199018114e82537')}
{'gruendung': '1907, 8, 14', 'tabellenplatz': 14, 'name': 'Paderborn', 'nike': 'n', 'farben': ['blau', 'weiss'], '_id':
ObjectId('55683dd1a199018114e82538')}
{'gruendung': '1892, 7, 25', 'tabellenplatz': 13, 'name': 'Hertha', 'nike': 'j', 'farben': ['blau', 'weiss'], '_id':
ObjectId('55683dd1a199018114e82539')}
{'gruendung': '1907, 8, 8', 'tabellenplatz': 12, 'name': 'Augsburg', 'nike': 'j', 'farben': ['rot', 'weiss'], '_id':
ObjectId('55683dd1a199018114e8253a')}
{'gruendung': '1910, 5, 15', 'tabellenplatz': 11, 'name': 'Pauli', 'nike': 'n', 'farben': ['braun', 'weiss'], '_id':
ObjectId('55683dd1a199018114e8253b')}
{'gruendung': '1900, 8,1', 'tabellenplatz': 10, 'name': 'Gladbach', 'nike': 'n', 'farben': ['schwarz', 'weiss', 'gruen'], '_id':
ObjectId('55683dd1a199018114e8253c')}
{'gruendung': '1899, 3, 8', 'tabellenplatz': 9, 'name': 'Frankfurt', 'nike': 'j', 'farben': ['rot', 'schwarz', 'weiss'], '_id':
ObjectId('55683dd1a199018114e8253d')}
{'gruendung': '1904, 11, 20', 'tabellenplatz': 8, 'name': 'Leverkusen', 'nike': 'n', 'farben': ['rot', 'schwarz'], '_id':
ObjectId('55683dd1a199018114e8253e')}
{'gruendung': '1893, 9, 9', 'tabellenplatz': 7, 'name': 'Stuttgart', 'nike': 'n', 'farben': ['rot', 'weiss'], '_id':
ObjectId('55683dd1a199018114e8253f')}
{'gruendung': '1899,2,4', 'tabellenplatz': 6, 'name': 'Werder', 'nike': 'j', 'farben': ['gruen', 'weiss'], '_id':
ObjectId('55683dd1a199018114e82540')}

```

```

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

```

```

> {'name': 'HSV', 'nike': 'n', 'farben': ['weiss', 'rot'], 'tabellenplatz': 17, 'gruendung': '1887, 09, 29'}

```

```

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

```

```

>{'_id': ObjectId('55674284a19901177043eec4'), 'tabellenplatz': 1}
Anmerkung: Eintrag für Augsburg wurde komplett überschrieben.

```

```

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)
```

```
{'gruendung': '1887, 09, 29', 'tabellenplatz': 17, 'name': 'HSV', 'nike': 'n', 'farben': ['weiss', 'rot'], 'abgestiegen': 'j',
'waschtemperatur': 90, '_id': ObjectId('55683dd1a199018114e82535')}]
{'gruendung': '1909, 12, 19', 'tabellenplatz': 16, 'name': 'Dortmund', 'nike': 'n', 'farben': ['gelb', 'schwarz'], '_id':
ObjectId('55683dd1a199018114e82536')}]
{'gruendung': '1904, 5, 4', 'tabellenplatz': 15, 'name': 'Schalke', 'nike': 'n', 'farben': ['blau'], '_id':
ObjectId('55683dd1a199018114e82537')}]
{'gruendung': '1907, 8, 14', 'tabellenplatz': 14, 'name': 'Paderborn', 'nike': 'n', 'farben': ['blau', 'weiss'],
'waschtemperatur': 90, '_id': ObjectId('55683dd1a199018114e82538')}]
{'gruendung': '1892, 7, 25', 'tabellenplatz': 13, 'name': 'Hertha', 'nike': 'j', 'farben': ['blau', 'weiss'], 'waschtemperatur':
90, '_id': ObjectId('55683dd1a199018114e82539')}]
{'gruendung': '1907, 8, 8', 'tabellenplatz': 12, 'name': 'Augsburg', 'nike': 'j', 'farben': ['rot', 'weiss'], 'waschtemperatur':
90, '_id': ObjectId('55683dd1a199018114e8253a')}]
{'gruendung': '1910, 5, 15', 'tabellenplatz': 11, 'name': 'Pauli', 'nike': 'n', 'farben': ['braun', 'weiss'], 'waschtemperatur':
90, '_id': ObjectId('55683dd1a199018114e8253b')}]
{'gruendung': '1900, 8,1', 'tabellenplatz': 10, 'name': 'Gladbach', 'nike': 'n', 'farben': ['schwarz', 'weiss', 'gruen'],
'waschtemperatur': 90, '_id': ObjectId('55683dd1a199018114e8253c')}]
{'gruendung': '1899, 3, 8', 'tabellenplatz': 9, 'name': 'Frankfurt', 'nike': 'j', 'farben': ['rot', 'schwarz', 'weiss'],
'waschtemperatur': 90, '_id': ObjectId('55683dd1a199018114e8253d')}]
{'gruendung': '1904, 11, 20', 'tabellenplatz': 2, 'name': 'Leverkusen', 'nike': 'n', 'farben': ['rot', 'schwarz'], '_id':
ObjectId('55683dd1a199018114e8253e')}]
{'gruendung': '1893, 9, 9', 'tabellenplatz': 7, 'name': 'Stuttgart', 'nike': 'n', 'farben': ['rot', 'weiss'], 'waschtemperatur':
90, '_id': ObjectId('55683dd1a199018114e8253f')}]
{'gruendung': '1899,2,4', 'tabellenplatz': 5, 'name': 'Werder', 'nike': 'j', 'farben': ['gruen', 'weiss'], 'waschtemperatur': 90,
'_id': ObjectId('55683dd1a199018114e82540')}]
```

Aufgabe 9

Aufwand: Im Vergleich zu redis, MongoDB oder Neo4j ist HBase viel komplexer zu bedienen. Die Installation ist nicht intuitiv und die vorhandenen Anleitungen sind unterschiedlich, wenig aussagend, nicht aktuell. Die Verwendung von HBase gestaltet sich äußerst schwierig. Man muss schon ein hohes Maß an HBase Kenntnissen mitbringen.

	<u>redis(VServer)</u>	<u>MongoDB(VServer)</u>	HBase(in VM)
<u>insert</u>	<u>~0.2ms</u>	<u>~0.4ms</u>	~2ms
<u>Get PLZ</u>	<u>~0.4ms</u>	<u>~20ms</u>	~75ms
<u>Get City</u>	<u>~5.5ms</u>	<u>~3.3ms</u>	~8ms

Empfehlungen: redis – ist gut einsetzbar, wenn man einfache Daten, die sich leicht in Key-Value unterteilen lassen, hat. Eventuell würde sich empfehlen, redis als eine Art Cache für die Suche zu verwenden. Alle Schlüssel-Parameter werden dabei in redis gespeichert, sowie ein Verweis auf die Rest-Daten.

Neo4J – ist gut einsetzbar, wenn man Beziehungen zwischen Daten abbilden und bearbeiten will.

MongoDB – ist gut einsetzbar, wenn man komplexe und viele unterschiedlichen Daten hat, zwischen denen nicht unbedingt eine Verbindung besteht.

HBase – ist vielleicht gut, wenn man extrem viele Daten hat, mit denen man arbeiten will. Der schwere Einstieg macht diese Datenbank, aber sehr unattraktiv.