

# Musterlösung zum Aufgabenkatalog

## Übungsaufgaben auf der Länderdatenbank:

Zunächst ist es hilfreich, einen Scan auf die gesamte Tabelle auszuführen:

```
$ hbase shell
```

```
> scan 'countries'
```

### 1. Welche Region wird mit dem Länderkürzel AQ abgekürzt. Wie viele Menschen leben dort?

```
> get 'countries', 'AQ'
```

Name: Antarctica

Einwohner: 0

### 2. Bei welchen Ländern fängt das Länderkürzel mit X an?

```
> scan 'countries', { ROWPREFIXFILTER => 'X' }
```

Gefundene Länder: XK -> Kosovo

### 3. Bei welchen Ländern fängt das Länderkürzel mit Y an?

```
> scan 'countries', { ROWPREFIXFILTER => 'Y' }
```

Gefundene Länder: YE -> Yemen

YT -> Mayotte

### 4. Welches Land hat das dreistellige Länderkürzel TON ?

```
> scan 'countries', {FILTER => "ValueFilter(=,'binary:TON')"
```

```
> get 'countries', 'TO'
```

Gefundenes Land: TO -> Togo

### 5. Wie groß ist die Weltbevölkerung?

```
> scan 'countries', {COLUMN => 'metainformation:population' }
```

Summieren nicht möglich, hierfür wird ein SQL-Queryserver wie Apache Phoenix benötigt.

### 6. Wie groß ist das Welt-GDP?

```
> scan 'countries', {COLUMN => 'metainformation:gdp' }
```

Daten sind als String eingespeichert, auch mit SQL ist eine Abfrage nicht ohne weiteres möglich. Die Daten müssen zunächst transformiert werden.

### 7. Wann wurden die Daten angelegt? Stimmt die Zeitzone?

```
> scan 'countries'
```

Die Einträge wurden zum Unix-Timestamp 1605698363910 angelegt. Das ergibt einen lesbaren Timestamp von **Wednesday, 18-Nov-20 11:19:23 UTC +910 ms**.

### 8. Was sind die Top-10-Länder nach BIP?

```
> scan 'countries', {COLUMN => 'metainformation:gdp' }
```

Daten sind als String eingespeichert, auch mit SQL ist eine Abfrage nicht ohne weiteres möglich.  
Die Daten müssen zunächst transformiert werden.

## Textaufgaben

1. Ein Eisenbahnunternehmen möchte ihre Zugfahrzeiten in einer Tabelle des Data Lakes speichern. Es soll für jede Teilstrecke ein eigener Eintrag mit Startzeit und Ankunftszeit angelegt werden. Die Daten sollen nach folgendem Modell aufgeschlüsselt werden:

Zugnummer -> Datum -> Teilstrecke -> Abfahrtszeit  
Ankunftszeit

Key	cf:Abfahrt	cf:Ankunft
ICE513/2020-10-21/Hamburg-Hannover	1603267320	1603270040
ICE513/2020-10-21/Hannover-Göttingen	1603270140	1603274040
ICE513/2020-10-21/Göttingen-Kassel	1603274100	1603278000
.....		
ICE513/2020-10-22/Hannover-Göttingen	1603353780	1603358400

2. Ein Versandhaus möchte seine Kundenstammdaten und Kundenbestellungen in einer Tabelle hinterlegen. Dabei sollen mehrere Bestellungen und Bestellpositionen pro Kunde angelegt werden können. Die Adresse und die Bezahltdaten des Kunden sollen zudem nicht in einem einzelnen String gespeichert werden. Die Daten sollen nach folgendem Modell aufgeschlüsselt werden:

Kundennummer -> Stammdaten -> Adresse -> Zeile 1  
Zeile 2  
Zeile....

Kundennummer -> Stammdaten -> Bezahltdaten -> Zeile 1  
Zeile 2  
Zeile...

Kundennummer -> Bestellung 1 -> Allgemeine Daten

Kundennummer -> Bestellung 1 -> Bestellposition 1

Bestellposition 2

Key	cf
10040040/Stammdaten/Adresse	cf:strasse=Lumpenweg; cf:hausnummer=1 cf:plz=12211; cf:ort=Berlin
10040040/Zahlungsdaten/Kreditkarte	cf:betreiber=MasterCard; cf:kreditkartennr=44445555333344441111
10040040/4500400	cf:aufgabedatum=1603353780; cf:preis=100.04EUR
10040040/4500400/001	cf:artikelnr=12123121231; cf:stück=12
10040040/4500400/002	cf:artikelnr=12123111111; cf:stück=3

3. Ein Unternehmen möchte seine Lieferantenstammsätze gemeinsam mit den Materialstammdaten in einer HBase-Tabelle sammeln. Dabei sollen die Adressen der

Lieferanten in mehreren Zellen gespeichert werden. Die Daten sollen nach folgendem Modell aufgeschlüsselt werden:

Lieferantennummer -> Materialart -> Materialnummer1

Materialnummer2....

Lieferantennummer -> Lieferantenstammdaten

Key	cf
555666/Stammdaten/Adresse	cf:strasse=Lumpenweg; cf:hausnummer=1 cf:plz=12211; cf:ort=Berlin
555666/Stammdaten/Zahlungen	cf:zahlungsziel=30; cf:iban=DE4544445555333344441111
10040040/Besen/32423423	cf:name=Spezial100; cf:preis=100.04EUR
10040040/Besen/33434344	cf:name=Spezial200; cf:preis=200.04EUR
10040040/Besen/43443344	cf:name=Spezial300; cf:preis=300.04EUR

## Übungsaufgaben auf der Google Cloud:

1. Legen Sie eine eigene Tabelle an, die alle Vorlesungen des Studiengangs mit Räumen, Dozenten sowie die entsprechenden Noten beinhalten.