

Name:  
 Klasse:  
 Datum:  
 Schuljahr: 2021/22

## Daten importieren

Oft ist es notwendig, Daten von anderen Quellen zu importieren. Hat man einen großen Kundenkreis, ist es sehr aufwendig, jedes mal bei einem neuen Kunden die PLZ bzw. den Ort einzugeben. Auf der Seite [www.suche-postleitzahl.org](http://www.suche-postleitzahl.org) kann man sich die Daten von Deutschland herunterladen. Die Daten stehen unter der Open Database Licence frei zur Verfügung.

```
osm_id,ags,ort,plz,landkreis,bundesland
1104550,08335001,Aach,78267,Landkreis Konstanz,Baden-Württemberg
1255910,07235001,Aach,54298,Landkreis Trier-Saarburg,Rheinland-Pfalz
62564,05334002,Aachen,52062,Städteregion Aachen,Nordrhein-Westfalen
62564,05334002,Aachen,52064,Städteregion Aachen,Nordrhein-Westfalen
62564,05334002,Aachen,52066,Städteregion Aachen,Nordrhein-Westfalen
62564,05334002,Aachen,52068,Städteregion Aachen,Nordrhein-Westfalen
62564,05334002,Aachen,52070,Städteregion Aachen,Nordrhein-Westfalen
```

Das Format entspricht nicht unserem Datenschema - es muss also schrittweise angepasst werden.

### 1. Temporäre Tabelle für die CSV-Daten anlegen.

Die Struktur entspricht dabei genau der CSV-Datei. Die Spaltenbreite wurde hier großzügig bemessen. Entspricht die Spaltenreihenfolge der Reihenfolge der CSV-Datei, so muss beim Import bzgl. der Spalten nichts zusätzlich angegeben werden. Für die Tabelle ist kein PK notwendig, da diese dann später sequentiell komplett gelesen wird (full table scan).

```
CREATE TABLE csv_ort (
  osm_id      VARCHAR(20),
  ags         VARCHAR(20),
  ort         VARCHAR(50),
  plz         VARCHAR(10),
  landkreis   VARCHAR(50),
  bundesland  VARCHAR(50)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

### 2. Datenimport (für Text-Dateien) wird über den Befehl LOAD DATA INFILE realisiert.

```
LOAD DATA LOCAL INFILE 'src/csv/zuordnung_plz_ort_landkreis.csv'
  INTO TABLE csv_ort
  CHARACTER SET utf8
  FIELDS TERMINATED BY ','
  LINES TERMINATED BY '\n'
  IGNORE 1 ROWS;
```

Die Ausgabe ist dabei:

```
Query OK, 12937 rows affected, 12937 warnings (0.221 sec)
Records: 12937 Deleted: 0 Skipped: 0 Warnings: 12937
```

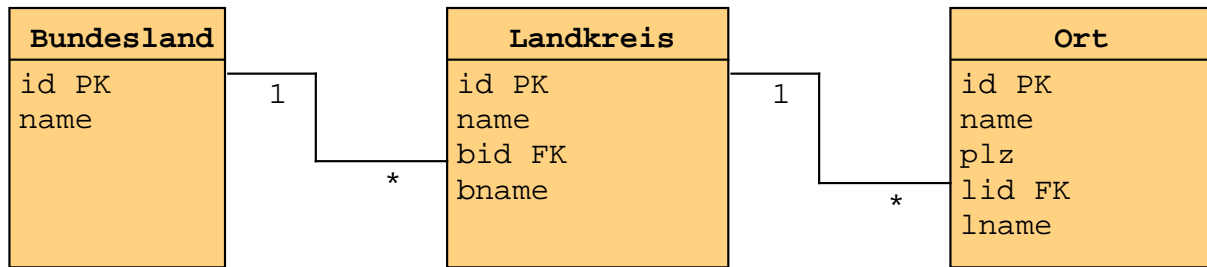
Die Tabelle hat folgenden Inhalt:

```
select ort,plz,landkreis,bundesland from csv_ort limit 4;
```

ort	plz	landkreis	bundesland
Aach	78267	Landkreis Konstanz	Baden-Württemberg
Aach	54298	Landkreis Trier-Saarburg	Rheinland-Pfalz
Aachen	52062	Städteregion Aachen	Nordrhein-Westfalen
Aachen	52064	Städteregion Aachen	Nordrhein-Westfalen

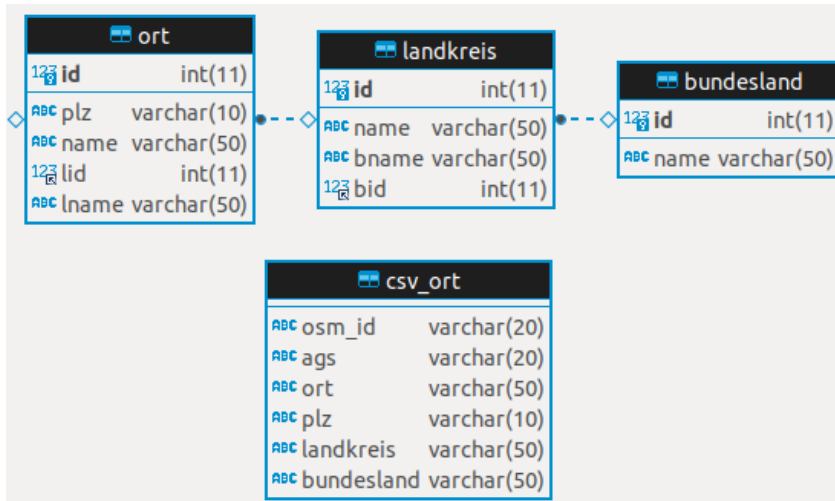


### 3. Tabellen ergänzen bzw. anpassen



Damit im nächsten Schritt die bid ermittelt werden kann, wird temporär der Bundeslandname (bname) mit gespeichert (genauso beim ort).

Erstellen Sie für die Tabellen entsprechende create table bzw. alter table Befehle.



### 4. Bundesländer anzeigen und einfügen

```
select distinct bundesland from csv_orst;
```

```

+-----+
| bundesland |
+-----+
| Baden-Württemberg |
| Rheinland-Pfalz |
| Nordrhein-Westfalen |
| Hessen |
| Schleswig-Holstein |
| Bayern |
| Thüringen |
| Niedersachsen |
| Mecklenburg-Vorpommern |
| Sachsen |
| Sachsen-Anhalt |
| Brandenburg |
| Saarland |
| Berlin |
| Bremen |
| Hamburg |
+-----+
  
```

```
INSERT INTO bundesland
(name) select distinct bundesland from csv_orst;
```

```
Query OK, 16 rows affected (0.013 sec)
Records: 16 Duplicates: 0 Warnings: 0
```



```
select * from bundesland;
```

id	name
1	Baden-Württemberg
2	Rheinland-Pfalz
3	Nordrhein-Westfalen
4	Hessen
5	Schleswig-Holstein
6	Bayern
7	Thüringen
8	Niedersachsen
9	Mecklenburg-Vorpommern
10	Sachsen
11	Sachsen-Anhalt
12	Brandenburg
13	Saarland
14	Berlin
15	Bremen
16	Hamburg

## 5. Landkreise einfügen<sup>1</sup>

```
INSERT INTO landkreis (name,bname)
```

```
  select
    distinct landkreis,bundesland
  from csv_ort
  where landkreis > ''
  order by landkreis;
```

```
# bid anpassen
```

```
update landkreis set bid=(select id from bundesland where bname=name);
```

```
# landkreise anzeigen
```

```
select * from landkreis limit 10;
```

id	name	bname	bid
1	Alb-Donau-Kreis	Baden-Württemberg	1
2	Altmarkkreis Salzwedel	Sachsen-Anhalt	14
3	Bodenseekreis	Baden-Württemberg	1
4	Burgenlandkreis	Sachsen-Anhalt	14
5	Donnersbergkreis	Rheinland-Pfalz	11
6	Eifelkreis Bitburg-Prüm	Rheinland-Pfalz	11
7	Ennepe-Ruhr-Kreis	Nordrhein-Westfalen	10
8	Enzkreis	Baden-Württemberg	1
9	Erzgebirgskreis	Sachsen	13
10	Hochsauerlandkreis	Nordrhein-Westfalen	10

<sup>1</sup> Achtung: kreisfreie Städte wie z. B. München haben keinen Landkreis!



## 6. Orte einfügen

```
# alte Orte löschen
TRUNCATE ort;

INSERT INTO ort (plz,name,lname)
select
    plz, ort,landkreis
from csv_ort
order by ort;

# lid anpassen
update ort set lid=(select id from landkreis where lname=name);

# Orte anzeigen
select * from ort limit 10;
```

id	plz	name	lid	lname
1	78267	Aach	143	Landkreis Konstanz
2	54298	Aach	235	Landkreis Trier-Saarburg
3	52080	Aachen	286	Städteregion Aachen
4	52078	Aachen	286	Städteregion Aachen
5	52076	Aachen	286	Städteregion Aachen
6	52074	Aachen	286	Städteregion Aachen
7	52072	Aachen	286	Städteregion Aachen
8	52070	Aachen	286	Städteregion Aachen
9	52068	Aachen	286	Städteregion Aachen
10	52066	Aachen	286	Städteregion Aachen

## 7. Aufräumen, Mitarbeiter anpassen und FKs setzen

```
ALTER table ort drop column lname;
ALTER table landkreis drop column bname;
DROP table csv_ort;

# Mitarbeiter: oid neu setzen
update mitarbeiter set oid=(
    select id from ort where plz='85662' AND name='Hohenbrunn')
where id in (8,9);
update mitarbeiter set oid=(
    select id from ort where plz='85653' AND name='Aying')
where id in (6,7);
update mitarbeiter set oid=(
    select id from ort where plz='82061' AND name='Neuried')
where id in (4,5);
update mitarbeiter set oid=(
    select id from ort where plz='85521' AND name='Hohenbrunn')
where id=3;
update mitarbeiter set oid=(
    select id from ort where plz='85653' AND name='Aying')
where id=2;
update mitarbeiter set oid=(
    select id from ort where plz='85609' AND name='Aschheim')
where id=1;

# FKs
ALTER table ort ADD CONSTRAINT fk_ort_landkreis
Foreign key (lid) References landkreis (id);

ALTER table landkreis ADD CONSTRAINT fk_landkreis_bundesland
Foreign key (bid) References bundesland (id);
```



## 8. Prüfen...

```
desc ort;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
plz	varchar(10)	YES		NULL	
name	varchar(50)	YES		NULL	
lid	int(11)	YES	MUL	NULL	

```
desc landkreis;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
bid	int(11)	YES	MUL	NULL	

```
select m.name,m.vorname, o.plz, o.name from mitarbeiter m left join  
ort o on m.oid=o.id;
```

name	vorname	plz	name
Walker	Jonny	85609	Aschheim
Schlau	Susi	85653	Aying
Ratlos	Rudi	85521	Hohenbrunn
Keller	Brigitte	82061	Neuried
Keller	Josef	82061	Neuried
Huber	Sepp	85653	Aying
Meier	Siglinde	85653	Aying
Mair	Hans	85662	Hohenbrunn
Maier	Peter	85662	Hohenbrunn

Sie sehen, mit SQL solch einen Datenimport zu machen, ist nicht immer ganz so einfach. Verwendet man dagegen eine Programmiersprache wie Java, ist dies deutlich einfacher (siehe extra Übung).

Führen Sie schrittweise den Import selber durch!

**Notiz:**



## Problem: kreisfreie Städte

Bei dem jetzigen Datenmodell gibt es ein Problem bei kreisfreien Städten, da diese keinem Landkreis zugeordnet sind.

```
select * from ort where lid is null limit 5;
```

id	plz	name	lid
287	92224	Amberg	NULL
331	91522	Ansbach	NULL
393	63739	Aschaffenburg	NULL
394	63741	Aschaffenburg	NULL
395	63743	Aschaffenburg	NULL

Somit lässt sich nicht ermitteln, in welchem Bundesland die Stadt liegt. Überlegen Sie sich dazu eine Lösung.

**Notiz:**

