

1)

Wir wollen eine Deutschlandkarte erstellen, in welcher für jedes Bundesland die Anzahl der Studierenden angegeben ist. Mit den zusätzlichen Daten aus der Bevölkerungsdatenbank, kann so z.B. der Anteil der Studies an der Gesamtbevölkerung ausgerechnet werden.

2)

a

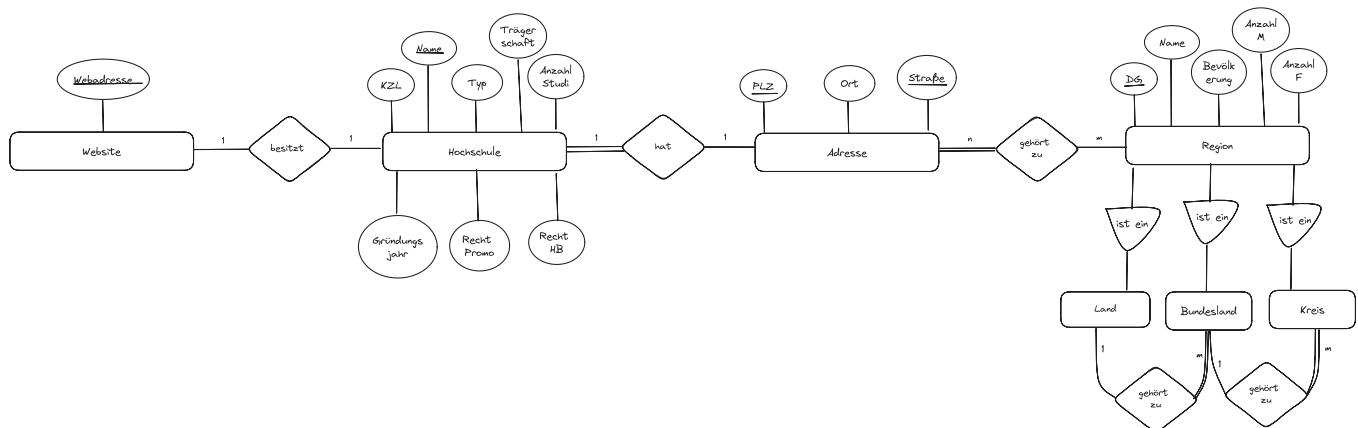
Relevant:

Für Verknüpfung: Bundesland, Name

Für Datenerhebung: Hochschulname, Hochschultyp, Trägerschaft, Anzahl Studierende, Promotionsrecht, Habilitationsrecht, Insgesamt, männlich, weiblich

b

Annahme/Änderung: Deutschland hat DG 0



c

Website (Webadresse)

Hochschule (KZL, Name, Typ, Trägerschaft, AnzahlStudi, Gründungsjahr, RechtPromo, RechtHB)

Adresse (PLZ, Ort, Straße)

Region (DG, Name, Bevölkerung, AnzahlIM, AnzahlIF)

Land (DG)

Bundesland (DG)

Kreis (DG)

besitzt (Webadresse, Name)

hat (Name, PLZ, Straße)

gehörtzuAR(PLZ, Straße, DG)

gehörtzuLB(DGL, DGB)

gehörtzuBK(DGB, DGK)

d

```
PS C:\Users\jurek> psql -U postgres
Passwort für Benutzer postgres:

psql (16.3)
Warnung: Konsolencodeseite (850) unterscheidet sich von der Windows-
Codeseite (1252). 8-Bit-Zeichen funktionieren möglicherweise nicht
richtig. Einzelheiten finden Sie auf der psql-Handbuchseite unter
»Notes for Windows users«.
Geben Sie »help« für Hilfe ein.

postgres=# \l

```

Liste der Datenbanken								
Name	Eigent'mer	Kodierung	Locale-Provider	Sortierfolge	Zeichentyp	ICU-Locale	ICU-Regeln	Zugriffsprivilegien
postgres	postgres	UTF8	libc	German_Germany.1252	German_Germany.1252			
template0	postgres	UTF8	libc	German_Germany.1252	German_Germany.1252			=c/postgres +
template1	postgres	UTF8	libc	German_Germany.1252	German_Germany.1252			=c/postgres +
test	postgres	UTF8	libc	German_Germany.1252	German_Germany.1252			postgres=Ct/postgres

```
(4 Zeilen)

postgres=# \c test
Sie sind jetzt verbunden mit der Datenbank »test« als Benutzer »postgres«.
test=# \d

```

Liste der Relationen			
Schema	Name	Typ	Eigent'mer
public	bevölkerung	Tabelle	postgres
public	bevölkerung_dg_seq	Sequenz	postgres
public	hochschule	Tabelle	postgres
public	wahl	Tabelle	postgres

```
(4 Zeilen)
```

3)

a

Siehe Datensätze

b

test=# SELECT * FROM bevölkerung

test-# ;

dg	name	insgesamt	mannlich	weiblich
0	Deutschland	84358845	41559052	42799793
1	Schleswig-Holstein	2953270	1443269	1510001
1001	Flensburg, kreisfreieStadt	92550	45874	46676
1002	Kiel, kreisfreieStadt	247717	120149	127568
1003	Lubeck, kreisfreieStadt, Hansestadt	218095	104723	113372
1004	Neumunster, kreisfreieStadt	79502	39256	40246
1051	Dithmarschen, Kreis	135252	66561	68691
1053	HerzogtumLauenburg, Kreis	203712	99689	104023
1054	Nordfriesland, Kreis	169043	82669	86374
1055	Ostholstein, Kreis	203606	97822	105784
1056	Pinneberg, Kreis	322130	157663	164467
1057	Plön, Kreis	131266	63357	67909
1058	Rendsburg-Eckernförde, Kreis	278979	137154	141825
1059	Schleswig-Flensburg, Kreis	206038	101614	104424
1060	Segeberg, Kreis	284988	140654	144334
1061	Steinburg, Kreis	132419	65268	67151
1062	Stormarn, Kreis	207973	120816	127157
2	Hamburg	1892122	925616	966506
3	Niedersachsen	8140242	4009822	4130420
31	Braunschweig, StatistischeRegion	1610865	796058	814807
3101	Braunschweig, kreisfreieStadt	251804	124770	127034
3102	Salzgitter, kreisfreieStadt	104548	51829	52719
3103	Wolfsburg, kreisfreieStadt	125961	62441	63520
3151	Gifhorn, Landkreis	180232	89675	90557
3152	Göttingen, Landkreis			
3153	Goslar, Landkreis	134672	66690	67982
3154	Helmstedt, Landkreis	92470	45648	46822
3155	Northeim, Landkreis	133258	65576	67682
3156	OsterodeamHarz, Landkreis			
3157	Peine, Landkreis	138931	68750	70181
3158	Wolfenbüttel, Landkreis	120531	59652	60879
3159	Göttingen, Landkreis	328458	161027	167431
32	Hannover, StatistischeRegion	2180058	1067013	1113045
3241	RegionHannover, Landkreis	1173891	573513	600378
3241001	Hannover, Landeshauptstadt	545045	266093	278952
3251	Diepholz, Landkreis	222816	110415	112401
3252	Hamelu-Pyrmont, Landkreis	150640	72781	77859
3254	Hildesheim, Landkreis	278371	135757	142614
3255	Holzminden, Landkreis	70911	35112	35799
3256	Nienburg (Weser), Landkreis	123469	61060	62409
3257	Schaumburg, Landkreis	159960	78375	81585
33	Lüneburg, StatistischeRegion	1757157	864122	893035
3351	Celle, Landkreis	181994	89138	92856
3352	Cuxhaven, Landkreis	201413	98150	103263
3353	Harburg, Landkreis	261890	128854	133036
3354	Lüchow-Dannenberg, Landkreis	40178	23981	25197
3355	Lüneburg, Landkreis	187552	91180	96372
3356	Osterholz, Landkreis	116551	56813	59738
3357	Rotenburg (Wumme), Landkreis	167604	83960	83644
3358	Heidekreis	145394	72697	72697
3359	Stade, Landkreis	210306	104302	106004
3360	Uelzen, Landkreis	94362	45906	48456
3361	Verden, Landkreis	140913	69141	71772
34	Weser-Ems, StatistischeRegion	2592162	1282629	1309533
3401	Delmenhorst, kreisfreieStadt	78385	38557	39828
3402	Emden, kreisfreieStadt	50535	25071	25464
3403	Oldenburg (Oldenburg), kreisfreieStadt	172830	82742	90088
3404	Osnabrück, kreisfreieStadt	167366	80755	86611
3405	Wilhelmshaven, kreisfreieStadt	76089	37268	38821
3451	Ammerland, Landkreis	128253	62498	65755
3452	Aurich, Landkreis	192672	94234	97838
3453	Cloppenburg, Landkreis	177701	89720	87981

[illegible]

4)

Zahl aller Hochschulen in Berlin

```
SELECT COUNT(hochschulname)
FROM hochschule
WHERE bundesland='Berlin';
```

Bundesländer nach Studierendenanzahl geordnet

```
SELECT bundesland, SUM(studis) AS studierendenanzahl
FROM hochschule
GROUP BY bundesland
ORDER BY SUM(studis) DESC;
```

Hochschulen mit Gesamtbevölkerung des Bundeslands

```
SELECT hochschulname, insgesamt AS Gesamtbevölkerung
FROM hochschule, bevölkerung
WHERE name = bundesland;
```

Hochschulen mit Bundesland geordnet nach Einwohner des Bundeslands

```
SELECT hochschulname AS Name, name AS Bundesland
FROM bevölkerung
RIGHT JOIN hochschule ON name = bundesland
Order By insgesamt DESC;
```

ACHTUNG!: Hochschulen ohne zugeordnete Bevölkerungszahl fallen raus.

Bundesland mit Frauenquote, absoluter Studentenzahl und relativer Studentenzahl

```
SELECT bundesland AS Bundesland, SUM(studis) AS Studierendenzahl,
(MAX(weiblich)*100)/MAX(insgesamt) AS Frauenquote,
(MAX(studis)*10000)/MAX(insgesamt) AS
Studierendenquote_pro_10000_Einwohner
FROM bevölkerung
JOIN hochschule ON name = bundesland
GROUP BY bundesland
ORDER BY SUM(studis) DESC;
```

5)

Siehe Code (CSV-Dateien müssen manuell importiert werden)

6)

7)

Herausforderungen & Entscheidungen:

Wir entscheiden, Land als Entity Typ zu erstellen, damit bei einer evtl. Erweiterung Daten aus anderen Ländern hinzugefügt werden können. Deswegen weisen wir Deutschland die DG 0 zu.

Adresse/ Hochschule kann nicht genau auf Kreis abgebildet werden, da manche Städte aus der Hochschule Datenbank nicht in den Kreisen vorkommen. Deswegen verbinden wir die Datenbanken nur über Bundesland.

Umlaute bereiten Probleme, weshalb wir sie in den Daten durch u,o,a ersetzen.

Der Pfad beim Import der json-Datei wurde nicht erkannt. Wir hatten vergessen, den Pfad in raw umzuwandeln.

Bei der Bereinigung der Tabelle sind uns die Leerzeichen vor und nach den eigentlichen Daten entgangen, wodurch es Probleme beim Vergleichen der Daten gab.

Neue Befehle im Umgang mit Postgresql.

Wesentliche Schritte

1. Postgresql installieren
2. Datenbank erstellen
3. Tabellen erstellen
4. CSV-Tabellen bereinigen
5. Daten aus csv in Tabellen kopieren
6. SQL-Abfragen erstellen und testen
7. Python code für Datenbankabfragen und Datenvisualisierung

8)

a

Wir haben den Datensatz um die jeweilige Wahlbeteiligung in den Bundesländern ergänzt [Quelle](#). Besonders der Vergleich zwischen Studenten pro 10.000 Einwohner und Wahlbeteiligung ist interessant, da dadurch evtl. Zusammenhänge aus Bildungsgrad und Wahlbeteiligung abgeleitet bzw. überprüft werden können.

b

Siehe Code