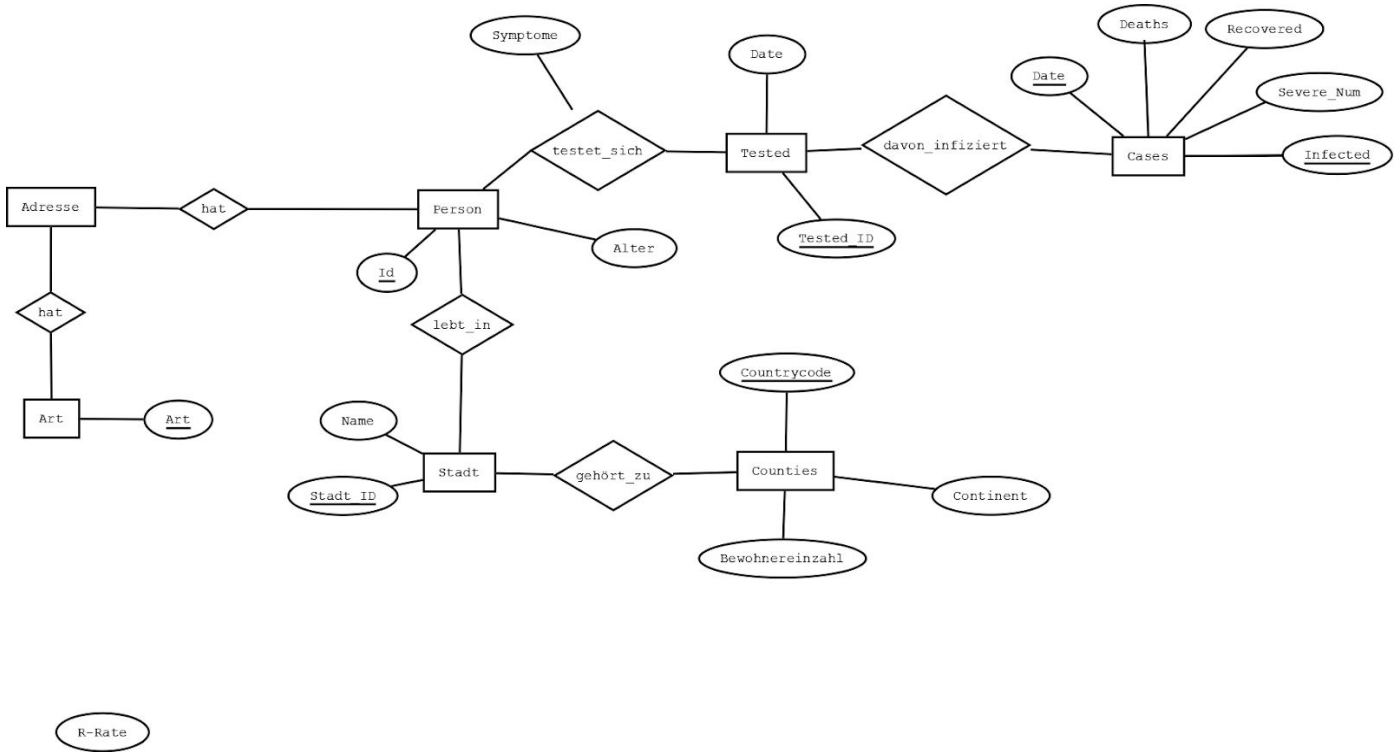
Datenbanksysteme
Projekt: COVID-19

Phase 1: Modellierung

Erste Gedanke

Bei der Modellierung hatten wir uns Anfangs überlegt eine Wissenschaftliche Datenbank zu erstellen und stießen auf einige Probleme zu denen wir später kommen werden. Unsere erstes ER-Modell sah folgendermaßen aus:



Hierbei stießen wir auf das erste Problem, und zwar die Daten die uns zur Verfügung gestellt wurden genügen nicht. Unsere Vorstellung war es in Städten die Zahl der Infizierten Menschen darstellen zu können und auf eventuelle Hotspots hinweisen zu können. Desweiteren haben wir uns ein Beispiel an der Internetseite der WHO genommen und sahen, dass selbst die Anzahl der getesteten Personen Angezeigt wurden. Im ER-Modell soll jeder Person Symptome haben auf die sie sich testen lassen. Jeder Person hat eine Personen_ID und dadurch kann sich auch die Bewohneranzahl berechnen lassen. Dadurch kann man auch den prozentualen Ansatz der infizierten Personen im Vergleich zur Einwohnerzahl berechnen. Desweiteren haben Personen Adressen und Orten an denen sie waren um ein Backtracking durchführen zu können. Bei der Entität Cases gibt es die

Gruppe: Thore Brehmer, Jonny Lam, David Ly
Prof. Dr. A. Voisaard, N. Lehmann
Datum: 27.07.2020

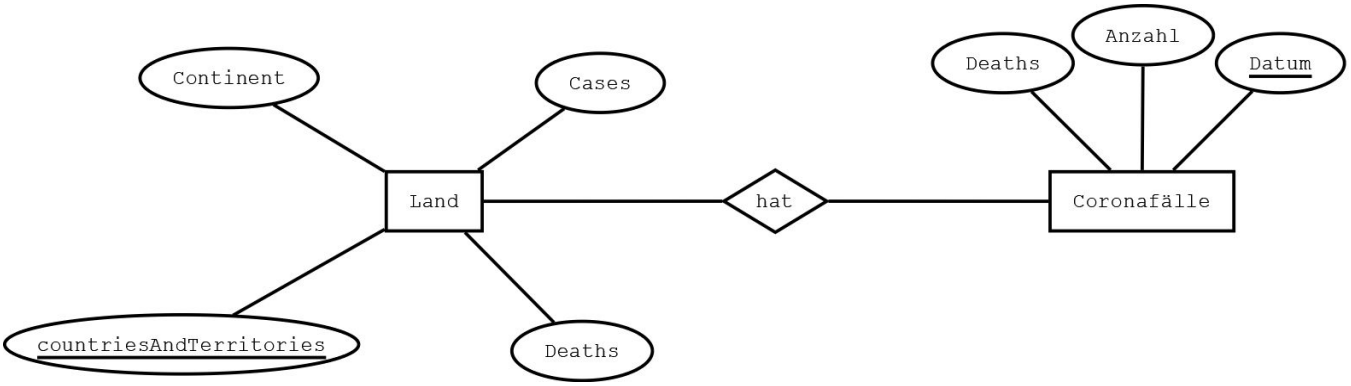
Attribute schwere Fälle, Tote, genesende und Infizierte Fälle. Mit den Informationen kann man die Reproduktionszahl und Prognosen berechnen.

Problem:

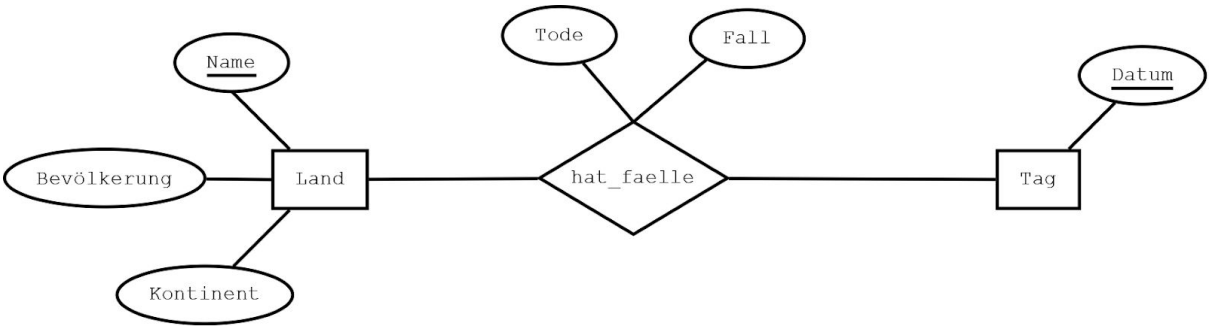
- Nicht ausreichende Daten zur Umsetzung
- Sehr hoher Aufwand und viel Recherche Aufwand für jedes Land

Zweiter Gedanke:

Nun da wir festgestellt haben das die Umsetzung unseres ersten ER-Modells nicht umsetzbar ist, haben wir uns auf die von der Universität gestellten Daten bezogen. Hierbei kann man erkennen, dass die Daten Inkonsistent sind also sich Widersprechen. Beispielsweise gibt es ein “Datum” und “Day”, “Month” und “Year”. Day, Month und Year ist überflüssig und kann gelöscht werden. “CountriesAndTerritories”, “geoId”, “countryterritoryCode” und “popData2018” Widersprechen sich auch.
“CountriesAndTerritories” geben den Name des Landes aus, wobei countryterritoryCode und geoId Abkürzungen sind.
“PopData2018” wussten wir Anfangs nicht was das Bedeuten soll und durch Beobachtung kann man sehen, dass die Zahlen eines Landes den Einträgen des selben Landes sich ähneln, also schließt sich daraus das es möglicherweise ein Zahlencode ist für das jeweilige Land. Bereinigung von Inkonsistenzen haben wir hier wieder ein ER-Modell erstellt.



Probleme:
Hierbei haben wir uns Gedanken darüber gemacht wenn wir Beispielsweise die Anzahl an Infizierten in Deutschland an einem bestimmten Tag haben wollen, dann würden wir die Zahl der COVID-19 aus allen Einträgen zusammen bekommen. Dadurch haben noch ein ER-Modell erstellt.



Unser Finales ER-Modell sieht folgendermaßen aus.

Land(Name, Kontinent,Bevoelkerung)
Tag(Datum)
hat_faelle(Name, Datum, Tode, Fall)

Phase 2: Datenverarbeitung

Idee: Da JSon Datei geeignetes Format für die Datenbank ist, müssen wir die Datei erst in eine für die Datenbank vorgesehen Datei konvertieren.
Problem 1: Die Datei ist fehlerhaft, wir müssen den letzten Datensatz, der nicht vollständig ist löschen und noch richtig Klammern.
Unser **erster Ansatz** war es die JSon Datei einfach in Excel zu öffnen und dann in das richtige Format zu konvertieren:

	Name	Value.dateRep	Value.day	Value.month	Value.year	Value.cases	Value.deaths	Value.countriesAndTerritories	Value.geoId	Value.…
1	records	14/06/2020	14	6	2020	556	5	Afghanistan	AF	AFG
2	records	13/06/2020	13	6	2020	656	20	Afghanistan	AF	AFG
3	records	12/06/2020	12	6	2020	747	21	Afghanistan	AF	AFG
4	records	11/06/2020	11	6	2020	684	21	Afghanistan	AF	AFG
5	records	10/06/2020	10	6	2020	542	15	Afghanistan	AF	AFG
6	records	09/06/2020	9	6	2020	575	12	Afghanistan	AF	AFG
7	records	08/06/2020	8	6	2020	791	30	Afghanistan	AF	AFG
8	records	07/06/2020	7	6	2020	582	18	Afghanistan	AF	AFG
9	records	06/06/2020	6	6	2020	915	9	Afghanistan	AF	AFG
10	records	05/06/2020	5	6	2020	787	6	Afghanistan	AF	AFG
11	records	04/06/2020	4	6	2020	758	24	Afghanistan	AF	AFG
12	records	03/06/2020	3	6	2020	759	5	Afghanistan	AF	AFG
13	records	02/06/2020	2	6	2020	545	8	Afghanistan	AF	AFG
14	records	01/06/2020	1	6	2020	680	8	Afghanistan	AF	AFG
15	records	31/05/2020	31	5	2020	866	3	Afghanistan	AF	AFG
16	records	30/05/2020	30	5	2020	623	11	Afghanistan	AF	AFG
17	records	29/05/2020	29	5	2020	580	8	Afghanistan	AF	AFG
18	records	28/05/2020	28	5	2020	625	7	Afghanistan	AF	AFG
19	records	27/05/2020	27	5	2020	658	1	Afghanistan	AF	AFG
20	records	26/05/2020	26	5	2020	591	1	Afghanistan	AF	AFG
21	records	25/05/2020	25	5	2020	584	2	Afghanistan	AF	AFG
22	records	24/05/2020	24	5	2020	782	11	Afghanistan	AF	AFG
23	records	23/05/2020	23	5	2020	540	12	Afghanistan	AF	AFG
24	records	22/05/2020	22	5	2020	531	6	Afghanistan	AF	AFG
25	records	21/05/2020	21	5	2020	492	9	Afghanistan	AF	AFG
26	records	20/05/2020	20	5	2020	581	5	Afghanistan	AF	AFG
27	records	19/05/2020	19	5	2020	408	4	Afghanistan	AF	AFG
28	records	18/05/2020	18	5	2020	262	1	Afghanistan	AF	AFG
29	records	17/05/2020	17	5	2020	0	0	Afghanistan	AF	AFG
30	records	16/05/2020	16	5	2020	1063	32	Afghanistan	AF	AFG
31	records	15/05/2020	15	5	2020	113	6	Afghanistan	AF	AFG
32	records	14/05/2020	14	5	2020	259	3	Afghanistan	AF	AFG

Hier haben wir dann die Tabellen für unser Modell erstellt, in dem wir ggf. Spalten gelöscht haben.
Da wir für die Tabelle “land” z.B. jeden Land Namen einzeln brauchen, müssen wir die, die Doppelt sind entfernen.
Mit Excel geht das sehr einfach, dafür gibt es schon eine integrierte Funktion:

Beim importieren der Datenbank gab es aber Probleme:

```
ERROR:  extra data after last expected column
CONTEXT:  COPY test, line 26: "Bonaire, Saint Eustatius and Saba"
```

```
ERROR:  invalid byte sequence for encoding "UTF8": 0xe7 0x61 0x6f
CONTEXT:  COPY test, line 51
```

Die Namen sind nicht “UTF8” kodiert, bzw. werden die Datensätze immer nach einem Komma getrennt, doch Land hat ein Komma in seinem Namen drinnen. Daraus folgt, dass eine extra Spalte erstellt wird die “Saint Eustatius and Saba” enthält.

Curaçao

Der Name ist z.B. auch nicht “UTF8” kodiert, weshalb es auch ein Error gab.

Lösung(?):

Unser naiver Ansatz für das erste Problem war, dass wir die Datensätze mit einem Semikolon trennen, wodurch das Komma kein Problem mehr war.
Beim 2. Problem haben wir mithilfe von Excel alle “ç” gesucht und mit einem “c” ersetzt.
Das alles funktionierte und wir konnten die Datenbank problemlos importieren.
Zufällig haben wir in der Datenbank negative Einträge von “cases” gefunden und haben die in positive Zahlen umgewandelt.
Nach einer langen Überlegung sind uns noch mehr Fälle eingefallen z.B. doppelte Datensätze, wo das Datum und das Land gleich sind.

Sicher hätte man das mit Excel lösen können, doch nach einer langen Überlegung sind wir zum Schluss gekommen, dass es nicht der Sinn der Aufgabe war, alles “manuell” zu bearbeiten.
Unser **zweiter Ansatz** war es ein Python Programm zu schreiben, der die JSon Datei in eine csv konvertiert und gleichzeitig noch die Fälle beachtet, die wir “manuell” gelöst haben.
Fall 1: UTF-8 Kodierung
Wir wissen, dass nicht alle Namen UTF-8 kodiert sind und müssen die Namen nun kodieren. Python bietet eine Funktion an, die das für uns macht und zwar *.encode(“utf-8”)* .
Fall 2: Negative Zahlen
Die Fälle, Tode und Bevölkerung dürfen ja nur positive Zahlen erhalten, es gibt keine negative Anzahl von Fälle oder Tode(also in unseren Kontext). Dafür haben wir in Python, die Zahlen ausgelesen, die negativ waren und haben diese Zahlen mit -1 multipliziert, damit die positiv wird. Doch man muss beachten, dass man die Werte erstmal in einem Integer bzw. Long umwandeln musste, damit man überhaupt sehen kann ob die negativ ist bzw. damit man die Zahl überhaupt multiplizieren kann.
Fall 3: Doppelte Einträge
Hier muss man beachten, dass es jeden Datensatz nur einmal gibt. Beziehungsweise, muss man darauf achten, dass es keine Datensätze gibt wo das Datum und der Landname doppelt gibt.
Das haben wir gelöst, indem wir die csv Datei öffnen und die ganzen Datensätze von oben nach unten durchgehen und die zwischenspeichern. Wenn etwas vorkommt, was wir schon hatten, dann wird es nicht übernommen. Die Datensätze die wir zwischengespeichert haben, sind die Daten ohne doppelte Einträge.

Datum
SET datestyle to SQL,DMY;
```
ALTER DATABASE "database" SET datestyle TO SQL,DMY;
```

Phase 3: Visualisierung & Datenanalyse

<-->

<-->
<-->
<-->
<-->