**XML Projekt**

**für Datenbanksysteme ILV 2020**

**an der FH Wiener Neustadt**

Gruppe 4 (Brandstätter, Forsthuber, Jernej)

**Aufgabenstellung:**

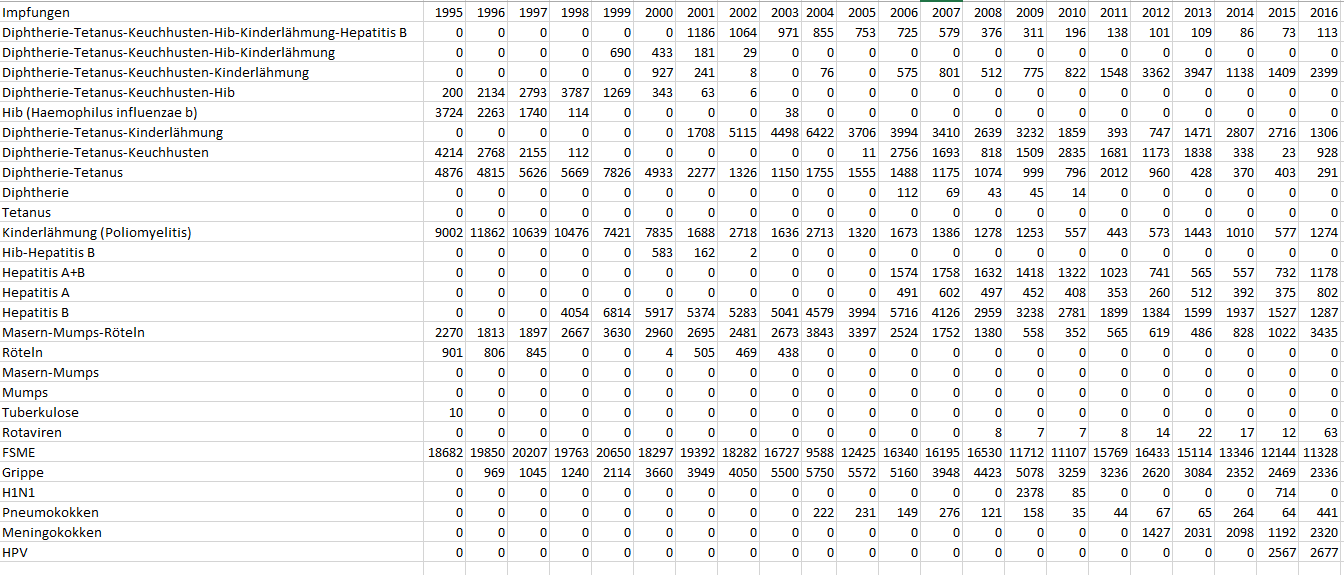
**Suche ein geeignetes Dataset über den Datenprovider Open Data Austria unter -** [**https://www.data.gv.at**](https://www.data.gv.at/)**. Formuliere und definiere auf den Daten dementsprechende Use Cases, welches du mit Queries beantworten sollst.**

Unser Team hat sich einen Datenset über das Impfverhalten der Stadt Linz ausgesucht. Dieses Datenset bietet eine Aufzeichnung von 20 Jahren von über 28 möglichen Impfungen. In Zuge der Analyse der Datenstruktur sowie des Dateninhalts haben wir folgende Use Cases definiert die es gilt zu beantworten:

1. Wie hat sich das Impfverhalten über alle Impfmöglichkeiten generell entwickelt?
2. Wie hat sich das Impfverhalten von klassischen Kindererkrankungen wie z.B. Masern, Mumps etc. entwickelt?
3. Wie hat sich das Impfverhalten die von Ärzten empfohlen werden wie z.B. Grippe, FSME, Hepatitis entwickelt?
4. Hat eine bestimmte Impfung in den letzten Jahren extrem zu genommen oder abgenommen?
5. Wo liegt der absolute Höchstwert von Impfungen & wann begann eine mögliche Trendwende nach unten oder umgekehrt?

**Task 1: Erstellung einer XML fähigen Datenbank – Auswahl Postgres\_**

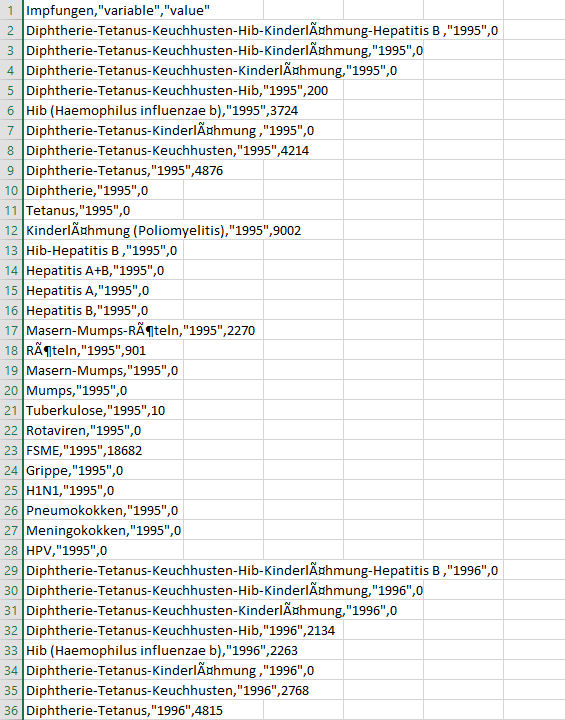
Auszug aus dem Datenset:



Transformieren mittels R-Code des Datenset für das Einlesen in Postgres:

|  |  |
| --- | --- |
| # R script |  |
|  | # File: impf\_recode.R |
|  | # Author: Christian Brandstätter |
|  | # Contact: bran.chri@gmail.com |
|  | # Date: 3.03.2020 |
|  | # Copyright (C) 2020 |
|  | # Description: R-Script zum Einlesen und transformieren der Impfdaten |
|  |  |
|  | library(reshape2) |
|  |  |
|  | # Einlesen |
|  | impfdat <- read.csv2("./Impf.csv", fileEncoding = "Windows-1252") |
|  | # von wide auf long |
|  | impfdat\_long <- melt(impfdat) |
|  |  |
|  | # entfernen von X |
|  | impfdat\_long$variable <- as.character(impfdat\_long$variable) |
|  | impfdat\_long$variable <- gsub("X", "", impfdat\_long$variable) |
|  |  |
|  | # Schreiben ohne Index |
|  | write.csv(impfdat\_long, "./Impfung\_Recode.csv", row.names = FALSE) |

Output nach Transformation des Datensets in ein entsprechenden Dataframe:



Nach erfolgreichem Setup der Installationsdatei von Postgres(pgAdmin4) wurde über die Console eine entsprechende Database und ein Table erstellt:

* Formularbeginn
* Formularende

|  |  |
| --- | --- |
|  | #Einloggen mit Benutzer 'postgres' |
|  | psql -U postgres  (Für Zugriff mit der Console muss die Systemsumgebungsvariable Path für PSQL erweitert werden. (C:/Programfiles/Postgres/bin) |
|  |  |
|  | #Passwort für Benutzer |
|  | Test |
|  |  |
|  | #Datenbank anlegen |
|  | CREATE DATABASE XML WITH ENCODING 'UTF8' LC\_COLLATE='German\_Germany' LC\_CTYPE='German\_Germany';  (Bei Fehlermeldung LC\_COLLATE='German-Austria & LC\_CTYPE='German\_Austria) |
|  |  |
|  | #scope auf erstellte db legen |
|  | \c xml |
|  |  |
|  | #Tabelle impf anlegen |
|  | CREATE TABLE impf (id serial PRIMARY KEY, impfung character varying(100), jahr smallint, anzahl Integer); (Für Zugriff von Postgres auf Datei muss der Zugriff unter Dateieigenschaften/Sicherheit ein neuer Benutzer "jeder" angelegt werden mit Schreib & Leserechte) |
|  |  |
|  | #csv file in die tabelle impf laden |
|  | COPY impf(impfung, jahr, anzahl) FROM 'C:/Users/Dell/Github/XML\_Projekt/Impfung\_Recode.csv' DELIMITER ',' CSV Header; |
|  |  |
|  | #alle daten aus impf anzeigen |
|  | SELECT \* FROM impf  Darstellung der Daten in pgAdmin4 unter Query Editor: |

**Task 2: Definition eines XML-Schema für die Daten**

<?xml version = "1.0" encoding = "UTF-8"?>

<!DOCTYPE impf [

<!ELEMENT impf (impfung, jahr, anzahl)\*>

<!ELEMENT impfung (#PCDATA)>

<!ELEMENT jahr (#PCDATA)>

<!ELEMENT anzahl (#PCDATA)>

]>

**Task 3: Erstelle über den SQL/XML-Support der RDBMS eine XML-Version der Daten die dem XML-Schema entspricht.**

Query über Query Editor pgAdmin4:

select xmlelement(name impf, xmlelement(name impfung, impfung), xmlelement(name jahr, jahr), xmlelement(name anzahl, anzahl)) FROM impf;

Ausgabe der XML-Datei:

\copy (select table\_to\_xml('impfung', true, false, '')) TO '/home/christian/wd/fhwn/db\_systeme/XML\_Project\_Open\_Data\_Austria/result\_neu.xml';

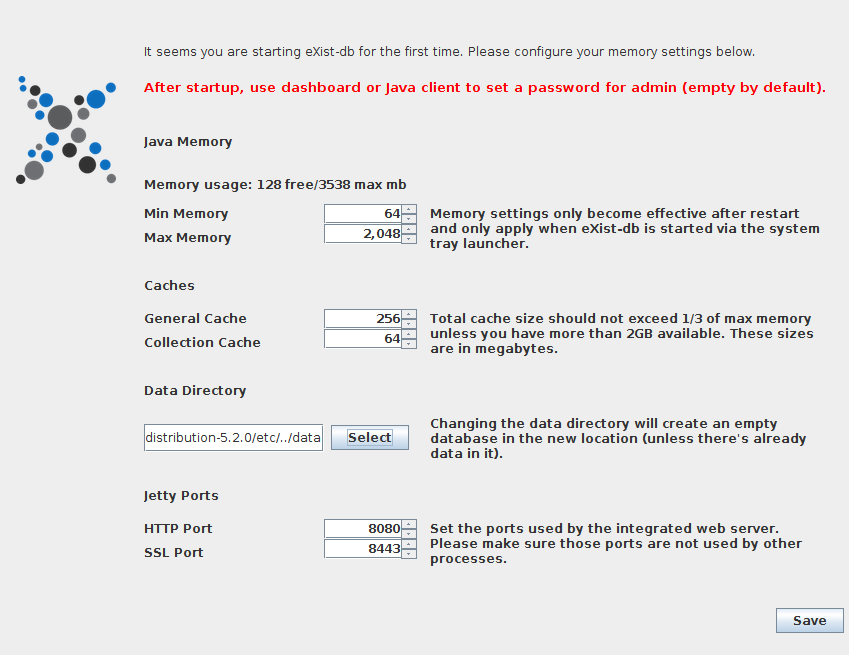
Output der XML-Datei (mit etwas manueller Nachbesserung):

Da es immer noch encoding-Probleme gab, wurden kurzerhand die Umlaute in ASCI-taugliche Formate ersetzt. Da die Validierung in der existdb auf „auto“ gesetzt wurde, musste das Validierungs-Schema mitgegeben werden.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE root SYSTEM "Validierung\_Datenschema.dtd">  <root> |
| <impf><impfung>Diphtherie-Tetanus-Keuchhusten-Hib-Kinderlaehmung-Hepatitis B </impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Diphtherie-Tetanus-Keuchhusten-Hib-Kinderlaehmung</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Diphtherie-Tetanus-Keuchhusten-Kinderlaehmung</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Diphtherie-Tetanus-Keuchhusten-Hib</impfung><jahr>1995</jahr><anzahl>200</anzahl></impf> |
| <impf><impfung>Hib (Haemophilus influenzae b)</impfung><jahr>1995</jahr><anzahl>3724</anzahl></impf> |
| <impf><impfung>Diphtherie-Tetanus-Kinderlaehmung </impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Diphtherie-Tetanus-Keuchhusten</impfung><jahr>1995</jahr><anzahl>4214</anzahl></impf> |
| <impf><impfung>Diphtherie-Tetanus</impfung><jahr>1995</jahr><anzahl>4876</anzahl></impf> |
| <impf><impfung>Diphtherie</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Tetanus</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Kinderlaehmung (Poliomyelitis)</impfung><jahr>1995</jahr><anzahl>9002</anzahl></impf> |
| <impf><impfung>Hib-Hepatitis B </impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Hepatitis A+B</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Hepatitis A</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Hepatitis B</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Masern-Mumps Roeteln</impfung><jahr>1995</jahr><anzahl>2270</anzahl></impf> |
| <impf><impfung>Roeteln</impfung><jahr>1995</jahr><anzahl>901</anzahl></impf> |
| <impf><impfung>Masern-Mumps</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Mumps</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>Tuberkulose</impfung><jahr>1995</jahr><anzahl>10</anzahl></impf> |
| <impf><impfung>Rotaviren</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
| <impf><impfung>FSME</impfung><jahr>1995</jahr><anzahl>18682</anzahl></impf> |
| <impf><impfung>Grippe</impfung><jahr>1995</jahr><anzahl>0</anzahl></impf> |
|  |
| Coding Thema…….. |

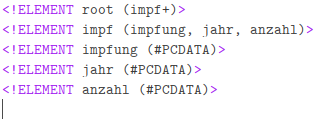
Task 4: Register your XML schema and afterwards store (validate) the related data in the native XML database exist - <http://exist-db.org/>

Installation Datenbank (Unix/Windows)

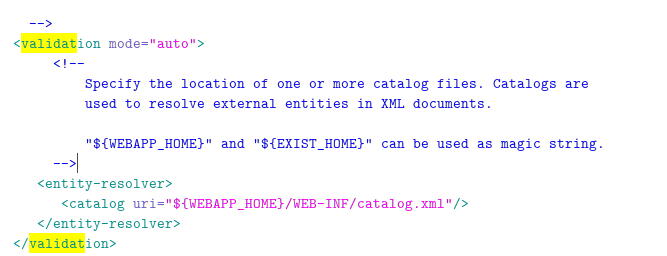
Alle Einstellungen wurden auf Standard gesetzt.

Um das bearbeitete XML-File hochladen und validieren zu können, war es notwendig:

1. ein Dtd-Schema zu erstellen:



2. die Datei conf.xml im Verzeichnis /etc zu ändern (validation mode = „auto“)

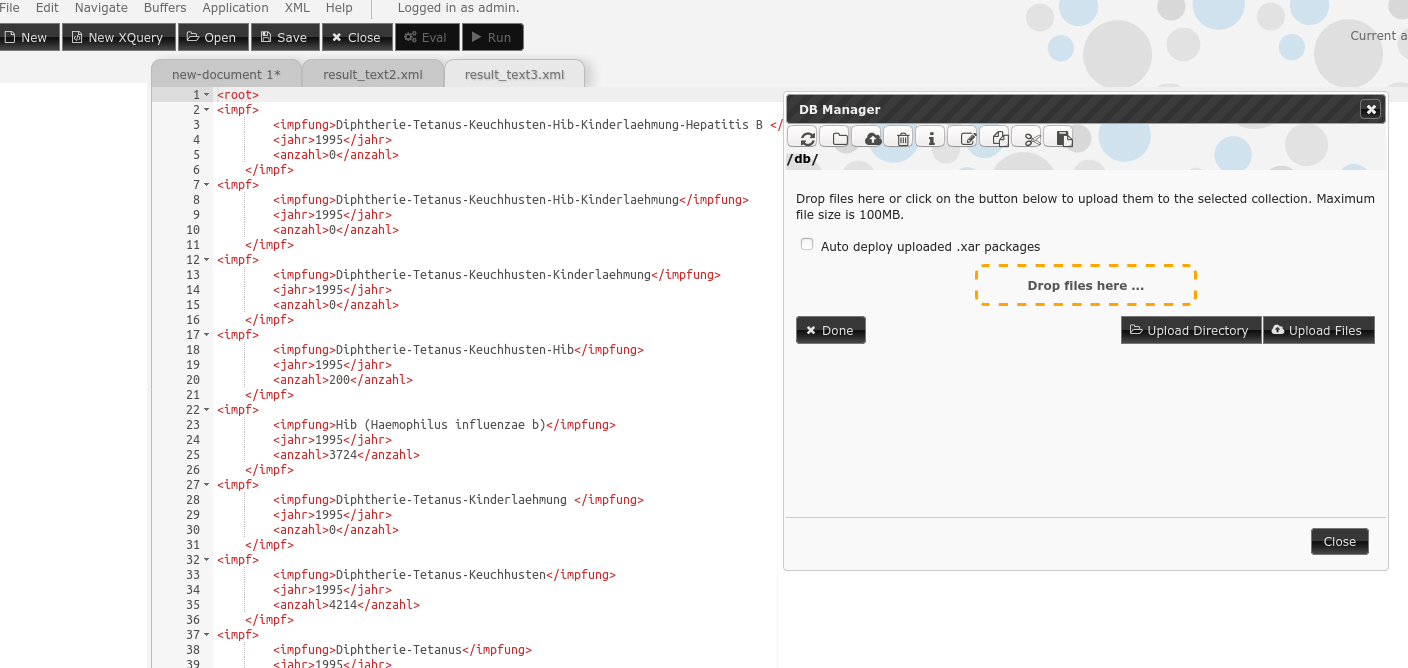
3. das hochzuladende xml zu modifizieren:

2.1. <!DOCTYPE root SYSTEM "Validierung\_Datenschema.dtd">

2.2. Änderung Encoding

4. die Datei catalog.xml im Verzeichnis ./*etc/*webapp/WEB-INF zu editieren, damit das Schema gefunden wird:

5. den eigentlichen Upload auszuführen.

Das ganze Prozedere war etwas Trial und Error, die Log-Datei exist.loc im Verzeichnis ./logs war dabei sehr hilfreich. Das Encoding-Problem erschwerte die Sache etwas.

Task 5: answer your data based question using XQuery via API usage in a supported programming language Upload your documentation (PDF between 10 and 15 pages) including • detailed documentation of each task so somebody else could re-do it • issues and solutions on the way and source code all together in one zip file onto moodle!