

Aufgabe-1: XML

Christoph Jungbauer, Daria Liakhovets, Stefan Kostecky

Dataset

Zum Einsatz kommt das Dataset der Erwerbstätigkeit in der Steiermark 2017:

<https://www.data.gv.at/katalog/dataset/e81dc3c6-cf98-43ca-848d-66a8eb1b61ef>

Die Daten stammen von Bundesanstalt Statistik Österreich, die Datei kann als CSV Datei heruntergeladen werden. Folgende Attribute sind darin enthalten:

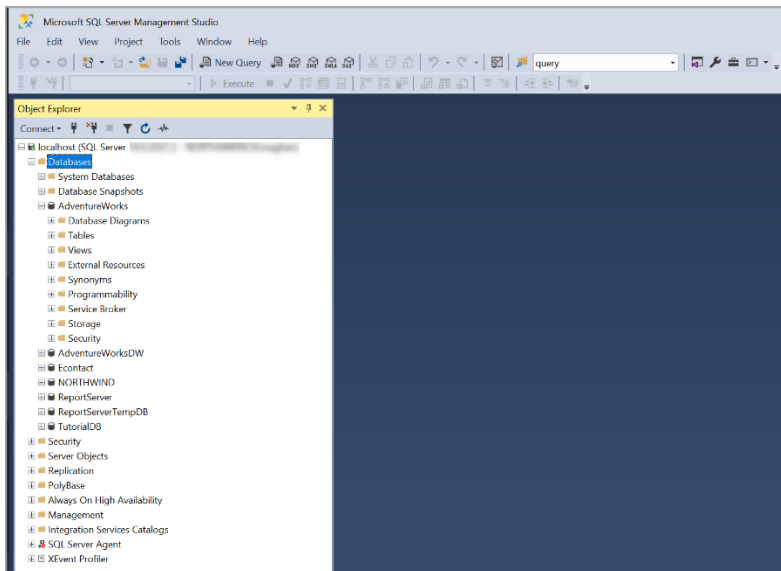
| | |
|---------------------|---|
| NUTS1 | NUTS1-Region (AT2=Südösterreich) |
| NUTS2 | NUTS2-Region (AT22=Steiermark) |
| | NUTS3-Region |
| NUTS3 | (AT221=Graz; AT222=Liezen; AT223=Östliche Obersteiermark; AT224=Oststeiermark; AT225=West- und Südsteiermark; AT226=Westliche Obersteiermark) |
| DISTRICT_CODE | Bezirkskennzahl |
| DISTRICT_NAME | Name des Bezirks |
| LAU_CODE | Gemeindekennzahl |
| LAU_NAME | Name der Gemeinde |
| NON_SELF_EMPL_M | unselbständig Beschäftigte Männer |
| SELF_EMPL_M | Selbständige und mithelfende Familienangehörige Männer |
| TEMP_ABSENT_M | Temporär von der Arbeit abwesend Männer |
| EMPL_TOTAL_M | Erwerbstätige Gesamt Männer |
| NON_SELF_EMPL_W | unselbständig Beschäftigte Frauen |
| SELF_EMPL_W | Selbständige und mithelfende Familienangehörige Frauen |
| TEMP_ABSENT_W | Temporär von der Arbeit abwesend Frauen |
| EMPL_TOTAL_W | Erwerbstätige Gesamt Frauen |
| NON_SELF_EMPL_TOTAL | unselbständig Beschäftigte Gesamt |
| SELF_EMPL_TOTAL | Selbständige und mithelfende Familienangehörige Gesamt |
| TEMP_ABSENT_TOTAL | Temporär von der Arbeit abwesend Gesamt |
| EMPL_TOTAL | Erwerbstätige Gesamt REF_DATE |

Die Daten werden jährlich aktualisiert, der Character Set Code ist ISO-8859-1.

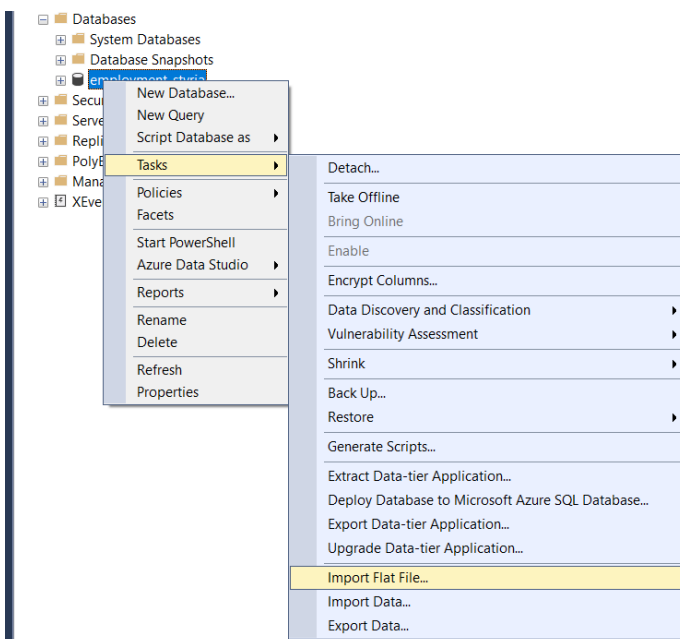
Struktur der XML-Datei definieren und mit SQL/XML Abfrage in einer RDBMS eine XML-Datei herstellen

Als nächster Schritt erfolgt die Installation von MSSQL für die Konvertierung in XML. SQL/XML ist Teil 14 der Structured Query Language (SQL)-Spezifikation. Zusätzlich zu den traditionellen vordefinierten SQL-Datentypen wie NUMERIC, CHAR, TIMESTAMP, ... führt sie den vordefinierten Datentyp XML mit Konstruktoren, ... zusammen und erlaubt Daten in xml zu transformieren.sql[1]

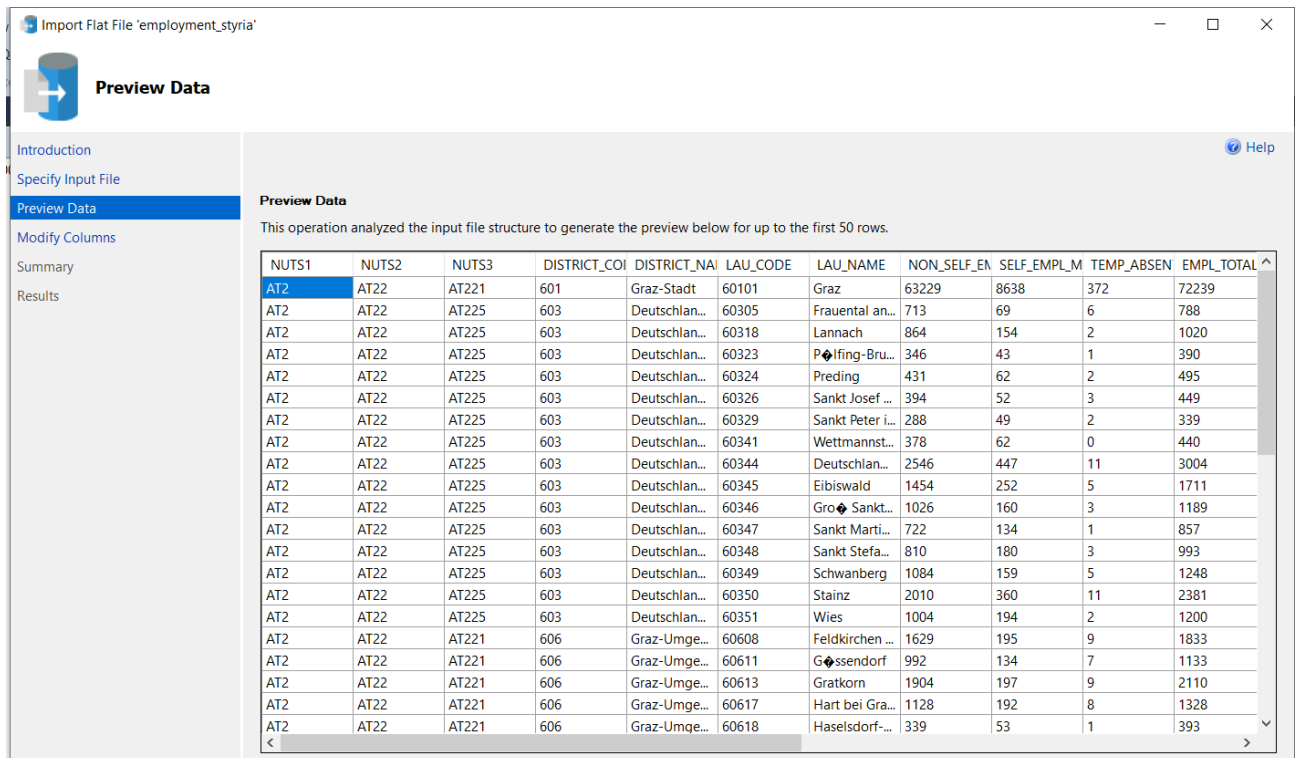
Die Administration erfolgt über das SQL Server Management Studio:[2]



In einem ersten Schritt wurden Daten in die DB (aus einer CSV Tabelle) hochgeladen (import Flat File):



Dabei wurde nach der Empfehlung für Microsoft für den SQL Wizard vorgegangen.[3] Man sieht das Preview von Daten:

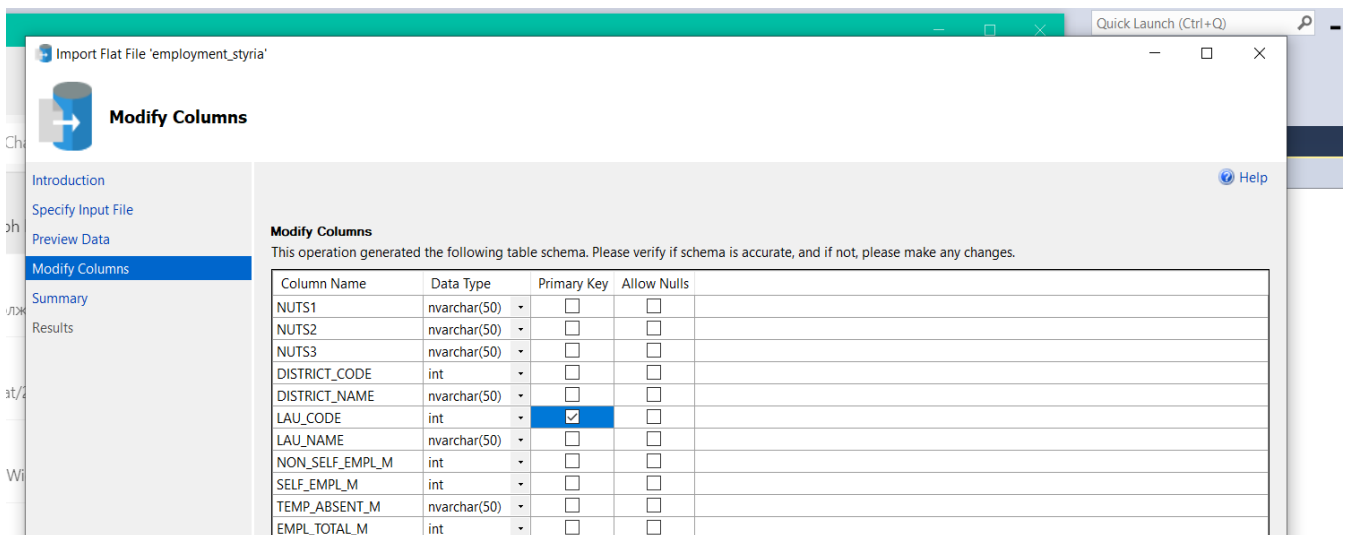


Preview Data

This operation analyzed the input file structure to generate the preview below for up to the first 50 rows.

| NUTS1 | NUTS2 | NUTS3 | DISTRICT_COI | DISTRICT_NA | LAU_CODE | LAU_NAME | NON_SELF_EN | SELF_EMPL_M | TEMP_ABSEN | EMPL_TOTAL |
|-------|-------|-------|--------------|---------------|----------|------------------|-------------|-------------|------------|------------|
| AT2 | AT22 | AT221 | 601 | Graz-Stadt | 60101 | Graz | 63229 | 8638 | 372 | 72239 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60305 | Frauental an... | 713 | 69 | 6 | 788 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60318 | Lannach | 864 | 154 | 2 | 1020 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60323 | Pöfing-Bru... | 346 | 43 | 1 | 390 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60324 | Preding | 431 | 62 | 2 | 495 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60326 | Sankt Josef ... | 394 | 52 | 3 | 449 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60329 | Sankt Peter i... | 288 | 49 | 2 | 339 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60341 | Wettmannst... | 378 | 62 | 0 | 440 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60344 | Deutschlan... | 2546 | 447 | 11 | 3004 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60345 | Eibiswald | 1454 | 252 | 5 | 1711 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60346 | Groß Sankt... | 1026 | 160 | 3 | 1189 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60347 | Sankt Marti... | 722 | 134 | 1 | 857 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60348 | Sankt Stefa... | 810 | 180 | 3 | 993 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60349 | Schwanberg | 1084 | 159 | 5 | 1248 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60350 | Stainz | 2010 | 360 | 11 | 2381 |
| AT2 | AT22 | AT225 | 603 | Deutschlan... | 60351 | Wies | 1004 | 194 | 2 | 1200 |
| AT2 | AT22 | AT221 | 606 | Graz-Umge... | 60608 | Feldkirchen ... | 1629 | 195 | 9 | 1833 |
| AT2 | AT22 | AT221 | 606 | Graz-Umge... | 60611 | Gössendorf | 992 | 134 | 7 | 1133 |
| AT2 | AT22 | AT221 | 606 | Graz-Umge... | 60613 | Gratkorn | 1904 | 197 | 9 | 2110 |
| AT2 | AT22 | AT221 | 606 | Graz-Umge... | 60617 | Hart bei Gra... | 1128 | 192 | 8 | 1328 |
| AT2 | AT22 | AT221 | 606 | Graz-Umge... | 60618 | Haselsdorf-... | 339 | 53 | 1 | 393 |

Es gibt Möglichkeit, Einstellungen für Spalten zu ändern. Wir haben LAU_CODE von Gemeinden als Primärschlüssel gewählt:



Modify Columns

This operation generated the following table schema. Please verify if schema is accurate, and if not, please make any changes.

| Column Name | Data Type | Primary Key | Allow Nulls |
|-----------------|--------------|-------------------------------------|--------------------------|
| NUTS1 | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| NUTS2 | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| NUTS3 | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| DISTRICT_CODE | int | <input type="checkbox"/> | <input type="checkbox"/> |
| DISTRICT_NAME | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| LAU_CODE | int | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| LAU_NAME | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| NON_SELF_EMPL_M | int | <input type="checkbox"/> | <input type="checkbox"/> |
| SELF_EMPL_M | int | <input type="checkbox"/> | <input type="checkbox"/> |
| TEMP_ABSENT_M | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| EMPL_TOTAL_M | int | <input type="checkbox"/> | <input type="checkbox"/> |

Nachdem die Daten erfolgreich importiert worden sind, kann man mit xml/sql query eine Ausgabe im XML-Format bekommen, dabei lässt sich die Struktur von der XML-Datei etwa mit verschachtelten Abfragen („nested query“) definieren.

Ein einfaches Beispiel xml/sql query dient zum Test der Funktionalität:[4]

```
1  use employment_styria;
2  SELECT NUTS3, LAU_CODE,
3         (SELECT EMPL_TOTAL, LAU_CODE
4          FROM   dbo.ERW_ST_BERUF statistiken
5          WHERE  gemeinde.LAU_CODE = statistiken.LAU_CODE
6          FOR XML AUTO, TYPE, ELEMENTS
7         )
8  FROM   dbo.ERW_ST_BERUF gemeinde
9  FOR XML AUTO, TYPE;
```

Wir haben uns für folgende Struktur der XML-Datei entschieden:

- Jede Gemeinde ist ein komplexes Element;
- Das „gemeinde“-Element besitzt folgende Attribute: LAU_CODE, LAU_NAME, REF_DATE (wobei das letzte für das gesamte Dataset konstant und somit nicht unbedingt notwendig ist);
- Das „gemeinde“-Element besteht wiederum aus vier komplexen Elementen: info, total, women, men;
- Das Element „info“ beinhaltet als einfache Elemente NUTS3, DISTRICT_CODE und DISTRICT_NAME;
- Elemente „total“, „women“, „men“ bestehen aus einfachen Elementen, die entsprechende Informationen über Anzahl von erwerbstätigen Personen in jeweiligen Kategorien beinhalten.

Anschließend wurden die Daten als XML-Datei exportiert. Grundlage hierfür war erneut eine Basis-Anleitung von Microsoft.[5] Nachfolgend der Quellcode der Abfrage.

```
1  use employment_styria;
2  SELECT LAU_CODE, LAU_NAME, REF_DATE,
3         (SELECT NUTS3, DISTRICT_CODE, DISTRICT_NAME
4          FROM   dbo.ERW_ST_BERUF info
5          WHERE  gemeinde.LAU_CODE = info.LAU_CODE
6          FOR XML AUTO, TYPE, ELEMENTS
7         ),
8         (SELECT NON_SELF_EEMPL_TOTAL, SELF_EEMPL_TOTAL, TEMP_ABSENT_TOTAL, EEMPL_TOTAL, LAU_CODE
9          FROM   dbo.ERW_ST_BERUF total
10         WHERE  gemeinde.LAU_CODE = total.LAU_CODE
11         FOR XML AUTO, TYPE, ELEMENTS
12        ),
13        (SELECT NON_SELF_EEMPL_W, SELF_EEMPL_W, TEMP_ABSENT_W, EEMPL_TOTAL_W, LAU_CODE
14         FROM   dbo.ERW_ST_BERUF women
15         WHERE  gemeinde.LAU_CODE = women.LAU_CODE
16         FOR XML AUTO, TYPE, ELEMENTS
17        ),
18        (SELECT NON_SELF_EEMPL_M, SELF_EEMPL_M, TEMP_ABSENT_M, EEMPL_TOTAL_M, LAU_CODE
19         FROM   dbo.ERW_ST_BERUF men
20         WHERE  gemeinde.LAU_CODE = men.LAU_CODE
21         FOR XML AUTO, TYPE, ELEMENTS
22        )
23  FROM   dbo.ERW_ST_BERUF gemeinde
24  FOR XML AUTO, TYPE;
```

Die eXist-db installieren und die XML-Datei mit dem Schema validieren

Im nächsten Schritt wurde die eXist-db installiert. Leider stellte sich der Teil als etwas komplizierter dar, die Dokumentation ist relativ rudimentär.[6]

Zum Einsatz kam die Version 5.2.0



Um die implizite Validierung zu aktivieren ist es nötig in der conf.xml der Datenbank den Eintrag zu Validierung auf „yes“ zu setzen:

`<validation mode="yes">`

```
809 <!--
810     Settings for XML validation
811     - mode
812         should XML source files be validated against a schema or DTD before
813         storing them? The setting is passed to the XML parser. The actual
814         effects depend on the parser you use. eXist comes with Xerces which
815         can validate against both: schemas and DTDs.
816
817         Possible values: "yes", "no", "auto". "auto" will leave validation
818         to the parser.
819     -->
820
821 <validation mode="yes">
822     <entity-resolver>
823         <catalog uri="{WEBAPP_HOME}/WEB-INF/catalog.xml"/>
824     </entity-resolver>
825 </validation>
```

Das Schema wurde in der Datenbank gespeichert: C:/eXist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd.

Die referenzierte catalog.xml Datei wurde am Ende um den entsprechenden Eintrag für das Schema der Uebung1 ergänzt:

```
<uri name="file:///C:/eXist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd" uri="entities/Uebung1.xsd" />
```

```
175
176     <uri name="file:///C:/eXist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd" uri="entities/Uebung1.xsd" />
177
178 </catalog>
```

Die entsprechende XSD Datei hat folgende Struktur:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="file:///C:/eXist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="gemeinden">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="gemeinde" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="info">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="NUTS3"
type="xs:string"></xs:element>
                    <xs:element
name="DISTRICT_CODE" type="xs:int"></xs:element>
                    <xs:element
name="DISTRICT_NAME" type="xs:string"></xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="total">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element
name="NON_SELF_EMPL_TOTAL" type="xs:int"></xs:element>
                    <xs:element
name="SELF_EMPL_TOTAL" type="xs:int"></xs:element>
                    <xs:element
name="TEMP_ABSENT_TOTAL" type="xs:int"></xs:element>
                    <xs:element name="EMPL_TOTAL"
type="xs:int"></xs:element>
                    <xs:element name="LAU_CODE"
type="xs:int"></xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="women">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element
name="NON_SELF_EMPL_W" type="xs:int"></xs:element>
                    <xs:element name="SELF_EMPL_W"
type="xs:int"></xs:element>
                    <xs:element
name="TEMP_ABSENT_W" type="xs:int"></xs:element>
                    <xs:element
name="EMPL_TOTAL_W" type="xs:int"></xs:element>
                    <xs:element name="LAU_CODE"
type="xs:int"></xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="men">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element
name="NON_SELF_EMPL_M" type="xs:int"></xs:element>
                    <xs:element name="SELF_EMPL_M"
type="xs:int"></xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

name="TEMP_ABSENT_M" type="xs:int"></xs:element>
name="EMPL_TOTAL_M" type="xs:int"></xs:element>
type="xs:int"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="LAU_CODE"
type="xs:int"></xs:attribute>
<xs:attribute name="LAU_NAME"
type="xs:string"></xs:attribute>
<xs:attribute name="REF_DATE"
type="xs:dateTime"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Ein Auszug der XML-Datei sieht wie folgt aus:

```

<?xml version="1.0" encoding="UTF-8"?>
<gemeinden xmlns="file:///C:/exist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3.org
file:///C:/exist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd">
<gemeinde LAU_CODE="60101" LAU_NAME="Graz " REF_DATE="2017-10-31T00:00:00">
<info>
<NUTS3>AT221</NUTS3>
<DISTRICT_CODE>601</DISTRICT_CODE>
<DISTRICT_NAME>Graz-Stadt</DISTRICT_NAME>
</info>
<total>
<NON_SELF_EMPL_TOTAL>120826</NON_SELF_EMPL_TOTAL>
<SELF_EMPL_TOTAL>13177</SELF_EMPL_TOTAL>
<TEMP_ABSENT_TOTAL>2958</TEMP_ABSENT_TOTAL>
<EMPL_TOTAL>136961</EMPL_TOTAL>
<LAU_CODE>60101</LAU_CODE>
</total>
<women>
<NON_SELF_EMPL_W>57597</NON_SELF_EMPL_W>
<SELF_EMPL_W>4539</SELF_EMPL_W>
<TEMP_ABSENT_W>2586</TEMP_ABSENT_W>
<EMPL_TOTAL_W>64722</EMPL_TOTAL_W>
<LAU_CODE>60101</LAU_CODE>
</women>
<men>
<NON_SELF_EMPL_M>63229</NON_SELF_EMPL_M>
<SELF_EMPL_M>8638</SELF_EMPL_M>
<TEMP_ABSENT_M>372</TEMP_ABSENT_M>
<EMPL_TOTAL_M>72239</EMPL_TOTAL_M>
<LAU_CODE>60101</LAU_CODE>
</men>
</gemeinde>
<gemeinde LAU_CODE="60305" LAU_NAME="Frauental an der La nitz " REF_DATE="2017-10-31T00:00:00">
<info>
<NUTS3>AT225</NUTS3>
<DISTRICT_CODE>603</DISTRICT_CODE>
<DISTRICT_NAME>Deutschlandsberg</DISTRICT_NAME>
</info>
<total>
<NON_SELF_EMPL_TOTAL>1317</NON_SELF_EMPL_TOTAL>
<SELF_EMPL_TOTAL>122</SELF_EMPL_TOTAL>
<TEMP_ABSENT_TOTAL>31</TEMP_ABSENT_TOTAL>
<EMPL_TOTAL>1470</EMPL_TOTAL>
<LAU_CODE>60305</LAU_CODE>
</total>
<women>
<NON_SELF_EMPL_W>604</NON_SELF_EMPL_W>
<SELF_EMPL_W>53</SELF_EMPL_W>
<TEMP_ABSENT_W>25</TEMP_ABSENT_W>
<EMPL_TOTAL_W>682</EMPL_TOTAL_W>
<LAU_CODE>60305</LAU_CODE>

```

```
</women>
<men>
  <NON_SELF_EMPL_M>713</NON_SELF_EMPL_M>
  <SELF_EMPL_M>69</SELF_EMPL_M>
  <TEMP_ABSENT_M>6</TEMP_ABSENT_M>
  <EMPL_TOTAL_M>788</EMPL_TOTAL_M>
  <LAU_CODE>60305</LAU_CODE>
</men>
</gemeinde>
```

Dabei wird explizit auf das zugehörige Schema verwiesen, gegen das Validiert wird.

Wenn nun die XML-Datei in die eXist DB hochgeladen werden kann, so ist die implizite Validierung erfolgreich.

Die datenbasierte Frage mit XQuery mithilfe von API beantworten

Zum Einsatz kommt die Python API „pyexistdb“.

```
In [1]: import sys

In [2]: !{sys.executable} -m pip install pyexistdb
```

Es erfolgt die Referenz auf die lokale eXist Datenbank (die IP erklärt sich aus einem Virtual Box IP Adressbereich).

```
In [3]: from pyexistdb import db

In [4]: urlexist = "http://192.168.38.2:8080/exist/"

In [5]: db = db.ExistDB(urlexist)
```

Um nun erfolgreich mit xquery arbeiten zu können ist es wichtig den korrekten Namespace anzugeben.

```
In [6]: # namespace definieren
xquery = '''
declare namespace ns = "file:///C:/eXist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd";
let $gemeinden := doc('/db/employment/Uebung1.xml')/*
for $x in $gemeinden/ns:gemeinde
return (data($x/@LAU_NAME), data($x/ns:total/ns:EMPL_TOTAL))'''
```

Die Fragestellung lautet:

Wie lauten die Zahlen der temporär von der Arbeit abwesenden Frauen (z.B. Mutterschutz, Elternkarenz) absteigend nach deren Anzahl sortiert und kumuliert nach Bezirken?

Die dazu nötige xquery lautet:

```
# namespace definieren
# women temp absent by district
xquery = '''
declare namespace ns = "file:///C:/exist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd";
let $gemeinden := doc('/db/employment/Uebung1.xml')/*
for $x in $gemeinden/ns:gemeinde

let $d := $x/ns:info/ns:DISTRICT_NAME
let $w := $x/ns:women/ns:TEMP_ABSENT_W

group by $d
order by sum($w) descending
return (data($d), data(sum($w)))'''
```

```
In [132]: # namespace definieren
# women temp absent by district
xquery = '''
declare namespace ns = "file:///C:/eXist-db/etc/webapp/WEB-INF/entities/Uebung1.xsd";
let $gemeinden := doc('/db/employment/Uebung1.xml')/*
for $x in $gemeinden/ns:gemeinde

let $d := $x/ns:info/ns:DISTRICT_NAME
let $w := $x/ns:women/ns:TEMP_ABSENT_W

group by $d
order by sum($w) descending
return (data($d), data(sum($w)))'''
```

Das Ergebnis der Abfrage ist dann:

```
In [133]: res = db.executeQuery(xquery)
hits = db.getHits(res)
```

```
In [134]: for i in range(hits):
print(str(db.retrieve(res,i)))
```

```
Graz-Stadt
2586
Graz-Umgebung
1579
Weiz
999
Hartberg-F rstenfeld
921
S doststeiermark
835
Leibnitz
832
Bruck-M rzzuschlag
794
Liezen
756
Murtal
567
Deutschlandsberg
543
Leoben
445
Voitsberg
402
Murau
252
```

Quellangaben:

- [1] „SQL/XML“, *Wikipedia*. 24-Nov-2019.
- [2] markingmyname, „SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS)“. [Online]. Verfügbar unter: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms>. [Zugegriffen: 19-März-2020].
- [3] yualan, „Import Flat File to SQL - SQL Server“. [Online]. Verfügbar unter: <https://docs.microsoft.com/en-us/sql/relational-databases/import-export/import-flat-file-wizard>. [Zugegriffen: 19-März-2020].
- [4] MightyPen, „FOR XML (SQL Server) - SQL Server“. [Online]. Verfügbar unter: <https://docs.microsoft.com/en-us/sql/relational-databases/xml/for-xml-sql-server>. [Zugegriffen: 19-März-2020].
- [5] MightyPen, „FOR XML Query Compared to Nested FOR XML Query - SQL Server“. [Online]. Verfügbar unter: <https://docs.microsoft.com/en-us/sql/relational-databases/xml/for-xml-query-compared-to-nested-for-xml-query>. [Zugegriffen: 19-März-2020].
- [6] „eXist-db Documentation“. [Online]. Verfügbar unter: <http://exist-db.org/exist/apps/doc/>. [Zugegriffen: 19-März-2020].