**Aufgabe-1: XML**

Christoph Jungbauer, Daria Liakhovets, Stefan Kostelecky

**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:

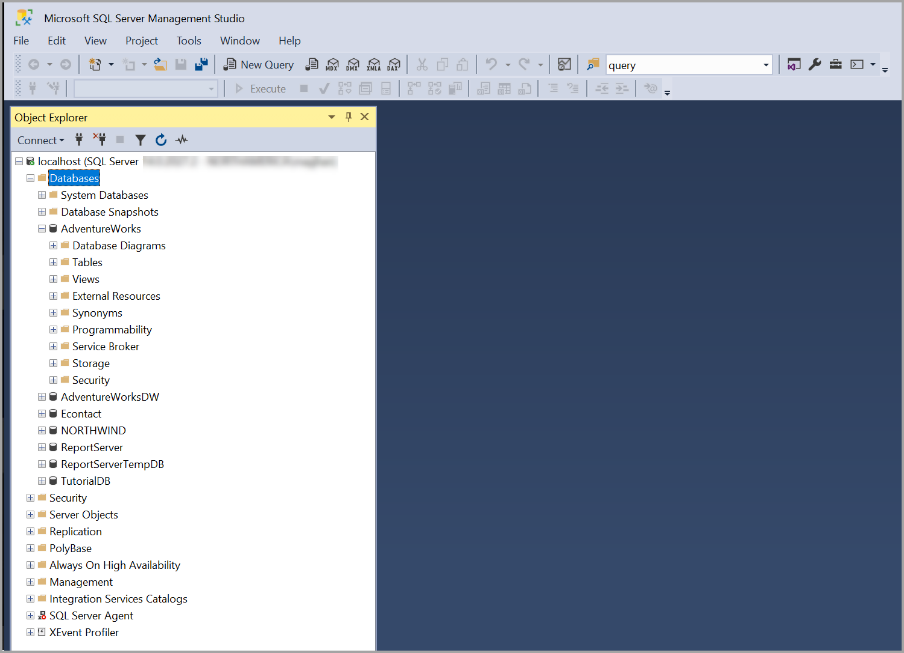
|  |  |
| --- | --- |
| NUTS1 | NUTS1-Region (AT2=Südösterreich) |
| NUTS2 | NUTS2-Region (AT22=Steiermark) |
| NUTS3 | NUTS3-Region  (AT221=Graz; AT222=Liezen; AT223=Östliche Obersteiermark;  AT224=Oststeiermark; AT225=West- und Südsteiermark;  AT226=Westliche Obersteiermark) |
| DISTRICT\_CODE | Bezirkskennzahl |
| 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):

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

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

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

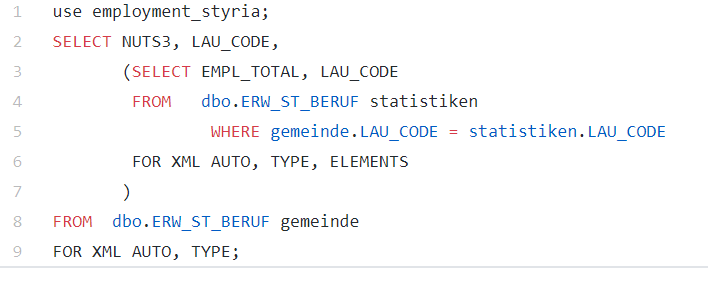
Es gibt Möglichkeit, Einstellungen für Spalten zu ändern. Wir haben LAU\_CODE von Gemeinden als Primärschlüssel gewählt:

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

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]



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.



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

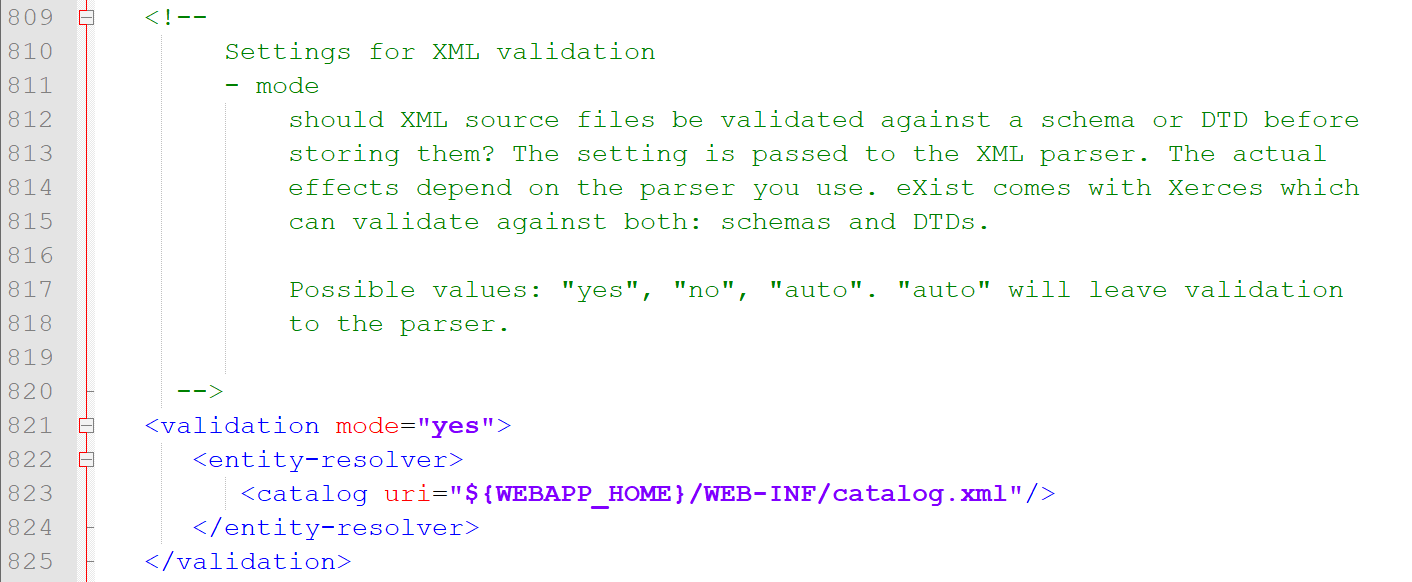
Im nächsten Schritt wurde die eXist-db installiert. Leider stellte sich der Teil als etwas komplizierter dar, die Dokumentation ist relativ rudimentär.[6]

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:

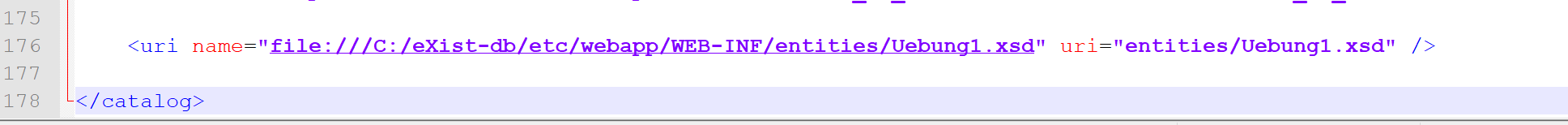
<validation mode="yes">



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" />



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:element name="TEMP\_ABSENT\_M" type="xs:int"></xs:element>

<xs:element name="EMPL\_TOTAL\_M" type="xs:int"></xs:element>

<xs:element name="LAU\_CODE" 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.

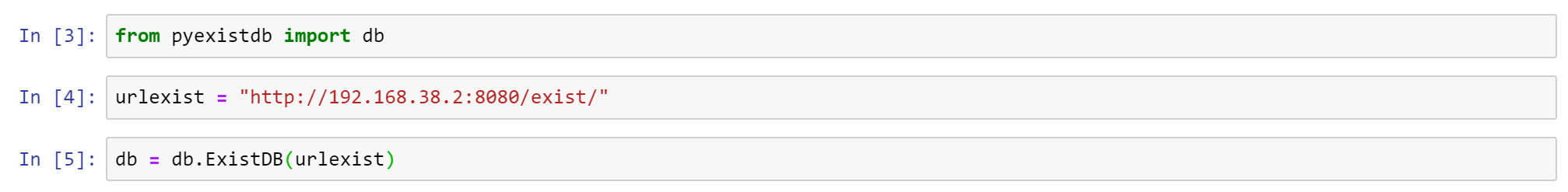
Wenn nun die XML-Datei in die eXist DB hochgeladen werden kann, so ist die implizite Validierung erfolgreich.

**Die datenbasierte Frage mit XQuery mithilfe von API beantworten**

Zum Einsatz kommt die Python API „pyexistdb“.



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



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



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)))'''



Das Ergebnis der Abfrage ist dann:





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].