Hörbe Rainer (7901955)
Utz Wilfrid (9902268)

Aufgabe 2-1a: Edge-Ansatz – Datenbasis XML -> Relationales Schema

Steps Performed

Based on results from 2-0 the following steps where performed:

1. XSL Transformation from original base XML to value SQL entries. The following 3 XSL sheets where used to create the integer and string value tables:
   a. 1 createDBValueTables.xsl: to create a flat list of values from the XML using attribute values and text node values. Using the generate-id function in XSL, IDs for each node are assigned.
   OUTPUT: value_output_step1_pretty.xml
   b. 1a createUniqueDBValueTables.xsl: to eliminate dublicates in the result from a), keys are introduced
   OUTPUT: value_output_step2_pretty.xml
   c. 1b createUniqueDB_INSERTINTO.xsl Based on the output of b) the respective INSERT INTO statements are created.
   OUTPUT: importSQLValueTables.sql
   https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/importSQLValueTables.sql

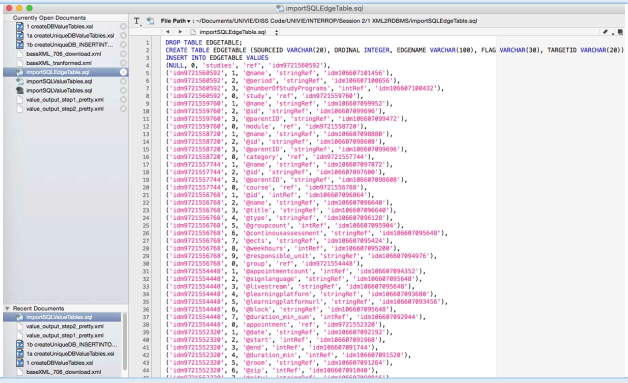
   d. Execution of the SQL statement in DB2. Necessary DROP/CREATE statements are available at the beginning of the script, written in step c.

   All transformations and formatting are performed using xsltproc and xmllint statements.

2. Based on the results of 1) the actual EDGE table is created using 2 createDBFromXML.xsl, the hierarchy is transformed into the necessary INSERT INTO statements (as bulk) and for values a matching to the results of b) is performed using the document() function in XSL. As a clean-up action, the SQL statement is manually modified to allow import (since 90.000 lines as bulk is too large).

   OUTPUT: importSQLEdgeTable.sql
   https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/importSQLEdgeTable.sql

Solution


importSQLEdgeTable.sql
Currently Open Documents
1 createDBValueTables.xsl
1a createUniqueDBValueTables.xsl
1b createUniqueDB_INSERTINT...
baseXML_706_download.xml
baseXML_tranformed.xml
importSQLEdgeTable.sql
importSQLValueTables.sql
importSQLValueTables.sql
value_output_step1_pretty.xml
value_output_step2_pretty.xml
Recent Documents
importSQLValueTables.sql
value_output_step2_pretty.xml
value_output_step1_pretty.xml
1b createUniqueDB_INSERTINTO...
1a createUniqueDBValueTables.xsl
1 createDBValueTables.xsl
baseXML_706_download.xml
File Path ▾ : ~/Documents/UNIVIE/DISS Code/UNIVIE/INTERROP/Session 2/1 XML2RDBMS/importSQLEdgeTable.sql
importSQLEdgeTable.sql

```
DROP TABLE EDGETABLE;
CREATE TABLE EDGETABLE (SOURCEID VARCHAR(20), ORDINAL INTEGER, EDGENAME VARCHAR(100), FLAG VARCHAR(30), TARGETID VARCHAR(20))
INSERT INTO EDGETABLE VALUES
(NULL, 0, 'studies', 'ref', 'idm9721560592'),
('idm9721560592', 1, '@name', 'stringRef', 'idm106607101456'),
('idm9721560592', 2, '@period', 'stringRef', 'idm106607100656'),
('idm9721560592', 3, '@numberOfStudyPrograms', 'intRef', 'idm106607100432'),
('idm9721560592', 0, 'study', 'ref', 'idm9721559760'),
('idm9721559760', 1, '@name', 'stringRef', 'idm106607099952'),
('idm9721559760', 2, '@id', 'stringRef', 'idm106607099696'),
('idm9721559760', 3, '@parentID', 'stringRef', 'idm106607099472'),
('idm9721559760', 0, 'module', 'ref', 'idm9721558720'),
('idm9721558720', 1, '@name', 'stringRef', 'idm106607098880'),
('idm9721558720', 2, '@id', 'stringRef', 'idm106607098608'),
('idm9721558720', 3, '@parentID', 'stringRef', 'idm106607099696'),
('idm9721558720', 0, 'category', 'ref', 'idm9721557744'),
('idm9721557744', 1, '@name', 'stringRef', 'idm106607097872'),
('idm9721557744', 2, '@id', 'stringRef', 'idm106607097600'),
('idm9721557744', 3, '@parentID', 'stringRef', 'idm106607098608'),
('idm9721557744', 0, 'course', 'ref', 'idm9721556768'),
('idm9721556768', 1, '@id', 'intRef', 'idm106607096864'),
('idm9721556768', 2, '@name', 'stringRef', 'idm106607096640'),
('idm9721556768', 3, '@title', 'stringRef', 'idm106607096640'),
('idm9721556768', 4, '@type', 'stringRef', 'idm106607096128'),
('idm9721556768', 5, '@groupcount', 'intRef', 'idm106607095904'),
('idm9721556768', 6, '@continousassessment', 'stringRef', 'idm106607095648'),
('idm9721556768', 7, '@ects', 'stringRef', 'idm106607095424'),
('idm9721556768', 8, '@weekhours', 'intRef', 'idm106607095200'),
('idm9721556768', 9, '@responsible_unit', 'stringRef', 'idm106607094976'),
('idm9721556768', 0, 'group', 'ref', 'idm9721554448'),
('idm9721554448', 1, '@appointmentcount', 'intRef', 'idm106607094352'),
('idm9721554448', 2, '@signlanguage', 'stringRef', 'idm106607095648'),
('idm9721554448', 3, '@livestream', 'stringRef', 'idm106607095648'),
('idm9721554448', 4, '@learningplatform', 'stringRef', 'idm106607093680'),
('idm9721554448', 5, '@learningplatformurl', 'stringRef', 'idm106607093456'),
('idm9721554448', 6, '@block', 'stringRef', 'idm106607095648'),
('idm9721554448', 7, '@duration_min_sum', 'intRef', 'idm106607092944'),
('idm9721554448', 0, 'appointment', 'ref', 'idm9721552320'),
('idm9721552320', 1, '@date', 'stringRef', 'idm106607092192'),
('idm9721552320', 2, '@start', 'intRef', 'idm106607091968'),
('idm9721552320', 3, '@end', 'intRef', 'idm106607091744'),
('idm9721552320', 4, '@duration_min', 'intRef', 'idm106607091520'),
('idm9721552320', 5, '@room', 'stringRef', 'idm106607091264'),
('idm9721552320', 6, '@zip', 'intRef', 'idm106607091040'),
```

Source Files

DEV ENVIRONMENT:
https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/1%20XML2RDBMS

Aufgabe 2-1b 1: SQL Query: Die am längsten pro Einheit dauernde(n) Lehrveranstaltunge(n) (Stunden, Gruppe, LV-Leiter)

Steps Performed

Since it is not clear what the query means semantically, the following assumptions have been made:

a) Query the course that has the longest appointment/lecture in the set of lectures
b) Since for the dataset, no further information on the group is available (only single group courses are offered), the group information is not considered, since similar as with the other attributes.
c) Stunden: the weekhours of the course are returend
d) LV-Leiter: since multiple “Leiter” (in role of V_Verantwortlich”) are possible, all lecturers are returned, resulting in dublicate entries (in case a course has 2 lecturers)

Solution

```
SELECT DISTINCT COURSEIDATTRVAL.INTEGERVALUE AS COURSEID,
                DURATION_MINATTR.INTEGERVALUE AS MAXAPPOINTMENTMIN,
                COURSENAMEVAL.STRINGVALUE AS COURSETITLE,
                COURSEWEEKHOURS.INTEGERVALUE AS COURSEWEEKHOURS,
                LECTURERNAMEVAL.STRINGVALUE AS LECTURERNAME
FROM EDGETABLE COURSE,
     EDGETABLE GROUP,
     EDGETABLE APPOINTMENT,
     EDGETABLE LECTURER,
     EDGETABLE LECTURERNAMEATTR,
     EDGETABLE DURATION,
     EDGETABLE COURSEIDATTR,
     EDGETABLE COURSENAMEATTR,
     EDGETABLE COURSEWEEKHOURSATTR,
     EDGEVALUE_TABLE_INT DURATION_MINATTR,
     EDGEVALUE_TABLE_INT COURSEIDATTRVAL,
     EDGEVALUE_TABLE_INT COURSEWEEKHOURS,
     EDGEVALUE_TABLE_STRING COURSENAMEVAL,
     EDGEVALUE_TABLE_STRING LECTURERNAMEVAL

WHERE
      -- get only starting from course tag
      COURSE.EDGENAME = 'course'
      -- join in the attribute @id
      AND COURSE.TARGETID = COURSEIDATTR.SOURCEID
      AND COURSEIDATTR.TARGETID = COURSEIDATTRVAL.INTEGERVALUEID
      AND COURSEIDATTR.EDGENAME = '@id'
      -- join in the attribute @title
      AND COURSE.TARGETID = COURSENAMEATTR.SOURCEID
      AND COURSENAMEATTR.TARGETID = COURSENAMEVAL.STRINGVALUEID
      AND COURSENAMEATTR.EDGENAME = '@title'
      -- join in the attribute @weekhours
      AND COURSE.TARGETID = COURSEWEEKHOURSATTR.SOURCEID
      AND COURSEWEEKHOURSATTR.TARGETID =
COURSEWEEKHOURS.INTEGERVALUEID
      AND COURSEWEEKHOURSATTR.EDGENAME = '@weekhours'
      -- join in the group tag
      AND COURSE.TARGETID = GROUP.SOURCEID
      AND GROUP.EDGENAME = 'group'
      -- join in the appointment
      AND GROUP.TARGETID = APPOINTMENT.SOURCEID
      AND APPOINTMENT.EDGENAME = 'appointment'
      -- join in the attribute @duration_min
      AND APPOINTMENT.TARGETID = DURATION.SOURCEID
```

<table>
<tr>
<td></td>
<td>

```
      AND DURATION.EDGENAME = '@duration_min'
      AND DURATION.TARGETID = DURATION_MINATTR.INTEGERVALUEID
      AND DURATION_MINATTR.INTEGERVALUE = (
         -- subquery to get the max value for the duration of an
appointment
         SELECT MAX(EDGEVALUE_TABLE_INT.INTEGERVALUE) FROM EDGETABLE
                INNER JOIN EDGEVALUE_TABLE_INT
                ON EDGETABLE.TARGETID =
EDGEVALUE_TABLE_INT.INTEGERVALUEID
                WHERE EDGETABLE.EDGENAME = '@duration_min')
      -- join in the lecturer
      AND GROUP.TARGETID = LECTURER.SOURCEID
      AND LECTURER.EDGENAME = 'lecturer'
      -- join in the attribute @name
      AND LECTURER.TARGETID = LECTURERNAMEATTR.SOURCEID
      AND LECTURERNAMEATTR.EDGENAME = '@name'
      AND LECTURERNAMEATTR.TARGETID = LECTURERNAMEVAL.STRINGVALUEID

;
```

</td>
</tr>
<tr>
<td>Result</td>
<td>The max length for of appointments in the basexml is 615min.
2 courses match, details below.

COURSEID: 70197
MAXAPPOINTMENTMIN: 615
COURSETITLE: Cultural history: Practice of methods in cultural history primary sources
COURSEWEEKHOURS: 20
LECTURERNAME: Diesenberger AND Schmale

COURSEID: 70280
MAXAPPOINTMENTMIN: 615
COURSETITLE: Workshop - Practicing Digital Methods in Historical Research
COURSEWEEKHOURS: 20
LECTURERNAME: König</td>
</tr>
<tr>
<td>Source Files</td>
<td>DEV ENVIRONMENT: https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/1%20XML2RDBMS
SQL:
https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_1%20SQL%20SELECT%20STATEMENT.sql
ResultSet (TXT):
https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_1%20RESULTSET.txt</td>
</tr>
</table>

Aufgabe 2-1b 2: SQL Query: Die am längsten pro Semester dauernde(n) Lehrveranstaltunge(n) (Stunden, Gruppe, LV-Leiter)

Steps Performed

Since it is not clear what the query means semantically, the following assumptions have been made:

a) Query the course that has the longest overall durations (summed up)
b) Group information, see abve.
c) Stunden: the weekhours of the course are returend
d) LV-Leiter: since multiple “Leiter” (in role of V_Verantwortlich”) are possible, all lecturers are returned, resulting in dublicate entries (in case a course has 2 lecturers)

Solution

```
SELECT DISTINCT COURSEIDATTRVAL.INTEGERVALUE AS COURSEID,
                DURATION_MINATTR.INTEGERVALUE AS MAXAPPOINTMENTMIN,
                COURSENAMEVAL.STRINGVALUE AS COURSETITLE,
                COURSEWEEKHOURS.INTEGERVALUE AS COURSEWEEKHOURS,
                LECTURERNAMEVAL.STRINGVALUE AS LECTURERNAME
FROM EDGETABLE COURSE,
     EDGETABLE GROUP,
     EDGETABLE LECTURER,
     EDGETABLE LECTURERNAMEATTR,
     EDGETABLE DURATION,
     EDGETABLE COURSEIDATTR,
     EDGETABLE COURSENAMEATTR,
     EDGETABLE COURSEWEEKHOURSATTR,
     EDGEVALUE_TABLE_INT DURATION_MINATTR,
     EDGEVALUE_TABLE_INT COURSEIDATTRVAL,
     EDGEVALUE_TABLE_INT COURSEWEEKHOURS,
     EDGEVALUE_TABLE_STRING COURSENAMEVAL,
     EDGEVALUE_TABLE_STRING LECTURERNAMEVAL

WHERE
      -- get only starting from course tag
      COURSE.EDGENAME = 'course'
      -- join in the attribute @id
      AND COURSE.TARGETID = COURSEIDATTR.SOURCEID
      AND COURSEIDATTR.TARGETID = COURSEIDATTRVAL.INTEGERVALUEID
      AND COURSEIDATTR.EDGENAME = '@id'
      -- join in the attribute @title
      AND COURSE.TARGETID = COURSENAMEATTR.SOURCEID
      AND COURSENAMEATTR.TARGETID = COURSENAMEVAL.STRINGVALUEID
      AND COURSENAMEATTR.EDGENAME = '@title'
      -- join in the attribute @weekhours
      AND COURSE.TARGETID = COURSEWEEKHOURSATTR.SOURCEID
      AND COURSEWEEKHOURSATTR.TARGETID =
COURSEWEEKHOURS.INTEGERVALUEID
      AND COURSEWEEKHOURSATTR.EDGENAME = '@weekhours'
      -- join in the group tag
      AND COURSE.TARGETID = GROUP.SOURCEID
      AND GROUP.EDGENAME = 'group'
      -- join in the attribute @duration_min_sum
      AND GROUP.TARGETID = DURATION.SOURCEID
      AND DURATION.EDGENAME = '@duration_min_sum'
      AND DURATION.TARGETID = DURATION_MINATTR.INTEGERVALUEID
      AND DURATION_MINATTR.INTEGERVALUE = (
         -- subquery to get the max value for the duration of an
appointment
         SELECT MAX(EDGEVALUE_TABLE_INT.INTEGERVALUE) FROM EDGETABLE
```

```
                INNER JOIN EDGEVALUE_TABLE_INT
                ON EDGETABLE.TARGETID =
EDGEVALUE_TABLE_INT.INTEGERVALUEID
                WHERE EDGETABLE.EDGENAME = '@duration_min_sum')
      -- join in the lecturer
      AND GROUP.TARGETID = LECTURER.SOURCEID
      AND LECTURER.EDGENAME = 'lecturer'
      -- join in the attribute @name
      AND LECTURER.TARGETID = LECTURERNAMEATTR.SOURCEID
      AND LECTURERNAMEATTR.EDGENAME = '@name'
      AND LECTURERNAMEATTR.TARGETID = LECTURERNAMEVAL.STRINGVALUEID

;
```

Result	The max length for of appointments in the basexml is 4680min (this is the sum of all appointments). 1 course matches, details below. COURSEID: 70240 MAXCOURSEMIN: 4680 COURSETITLE: Diplomatics and Chronology COURSEWEEKHOURS: 60 LECTURERNAME: Lackner, Schwarcz, Scharer
Source Files	DEV ENVIRONMENT: https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/1%20XML2RDBMS SQL: https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_2%20SQL%20SELECT%20STATEMENT.sql ResultSet (TXT): https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_2%20RESULTSET.txt

Aufgabe 2-1b 3: SQL Query:
Alle nach 18 Uhr stattfindenden LVs, die pruefungsimmanent sind, und nur einen LV-Leiter haben.

Steps Performed

Since no specific information is requested, the left join needed to also show all other values as above is skipped.

Solution

```
SELECT COURSEID, COUNT(COURSEID) AS LECTURERCOUNT FROM (

SELECT DISTINCT COURSEIDATTRVAL.INTEGERVALUE AS COURSEID,
                STARTATTR.INTEGERVALUE AS STARTTIME,
                COURSENAMEVAL.STRINGVALUE AS COURSETITLE,
                ASSESSMENTATTRVAL.STRINGVALUE AS ASSESSMENTTYPE,
                COURSEWEEKHOURS.INTEGERVALUE AS COURSEWEEKHOURS,
                LECTURERNAMEVAL.STRINGVALUE AS LECTURERNAME


FROM EDGETABLE COURSE,
     EDGETABLE GROUP,
     EDGETABLE APPOINTMENT,
     EDGETABLE LECTURER,
     EDGETABLE LECTURERNAMEATTR,
     EDGETABLE START,
     EDGETABLE COURSEIDATTR,
     EDGETABLE COURSENAMEATTR,
     EDGETABLE ASSESSMENTATTR,
     EDGETABLE COURSEWEEKHOURSATTR,
     EDGEVALUE_TABLE_INT STARTATTR,
     EDGEVALUE_TABLE_INT COURSEIDATTRVAL,
     EDGEVALUE_TABLE_INT COURSEWEEKHOURS,
     EDGEVALUE_TABLE_STRING COURSENAMEVAL,
     EDGEVALUE_TABLE_STRING ASSESSMENTATTRVAL,
     EDGEVALUE_TABLE_STRING LECTURERNAMEVAL

WHERE
      -- get only starting from course tag
      COURSE.EDGENAME = 'course'
      -- join in the attribute @id
      AND COURSE.TARGETID = COURSEIDATTR.SOURCEID
      AND COURSEIDATTR.TARGETID = COURSEIDATTRVAL.INTEGERVALUEID
      AND COURSEIDATTR.EDGENAME = '@id'
      -- join in the attribute @title
      AND COURSE.TARGETID = COURSENAMEATTR.SOURCEID
      AND COURSENAMEATTR.TARGETID = COURSENAMEVAL.STRINGVALUEID
      AND COURSENAMEATTR.EDGENAME = '@title'
      -- join in the attribute @continousassessment
      AND COURSE.TARGETID = ASSESSMENTATTR.SOURCEID
      AND ASSESSMENTATTR.TARGETID = ASSESSMENTATTRVAL.STRINGVALUEID
      AND ASSESSMENTATTR.EDGENAME = '@continousassessment'
      AND ASSESSMENTATTRVAL.STRINGVALUE = 'Yes'

      -- join in the attribute @weekhours
      AND COURSE.TARGETID = COURSEWEEKHOURSATTR.SOURCEID
      AND COURSEWEEKHOURSATTR.TARGETID =
COURSEWEEKHOURS.INTEGERVALUEID
      AND COURSEWEEKHOURSATTR.EDGENAME = '@weekhours'
      -- join in the group tag
      AND COURSE.TARGETID = GROUP.SOURCEID
      AND GROUP.EDGENAME = 'group'
      -- join in the appointment
      AND GROUP.TARGETID = APPOINTMENT.SOURCEID
      AND APPOINTMENT.EDGENAME = 'appointment'
      -- join in the attribute @START_min
      AND APPOINTMENT.TARGETID = START.SOURCEID
      AND START.EDGENAME = '@start'
      AND START.TARGETID = STARTATTR.INTEGERVALUEID
      -- only get those that start later then 18:00
      AND STARTATTR.INTEGERVALUE > 1800
      -- join in the lecturer
      AND GROUP.TARGETID = LECTURER.SOURCEID
      AND LECTURER.EDGENAME = 'lecturer'
      -- join in the attribute @name
      AND LECTURER.TARGETID = LECTURERNAMEATTR.SOURCEID
      AND LECTURERNAMEATTR.EDGENAME = '@name'
      AND LECTURERNAMEATTR.TARGETID = LECTURERNAMEVAL.STRINGVALUEID
)
GROUP BY COURSEID
HAVING COUNT(COURSEID) = 1
;
```

Result	3 courses match the above request: COURSEID: 70068 COURSEID: 70086 COURSEID: 70117
Source Files	DEV ENVIRONMENT: https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/1%20XML2RDBMS SQL: https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_3%20SQL%20SELECT%20STATEMENT.sql ResultSet (TXT): https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_3%20RESULTSET.txt