|  |  |
| --- | --- |
| 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 |  |
| 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:   1. Query the course that has the longest appointment/lecture in the set of lectures 2. 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. 3. Stunden: the weekhours of the course are returend 4. 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  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  ; |
| Result | 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 |
| 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_1%20SQL%20SELECT%20STATEMENT.sql>  ResultSet (TXT): <https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_1%20RESULTSET.txt> |

|  |  |
| --- | --- |
| 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:   1. Query the course that has the longest overall durations (summed up) 2. Group information, see abve. 3. Stunden: the weekhours of the course are returend 4. 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> |