

## Aufgabe 2-2: Anfragen mit XQuery mit BaseX

Steps Performed	<ol style="list-style-type: none"> <li>1. Installation von BaseX</li> <li>2. Create a new database based on baseXML_tranformed.xml</li> <li>3. Queries created and saved as *.xq, Resultset as xml, whereas the result is not valid xml since no root node defined.</li> </ol>
-----------------	--

Solution	<p>2-2a</p> <pre>for \$course in collection("Z:\INTERROP\Session 2\2 BaseX\baseXML_tranformed.xml")//course let \$courseTitle := \$course/@title let \$moduleName := \$course/ancestor::module[1]/@name let \$appointments := \$course//appointment return &lt;course&gt;{\$courseTitle}{\$moduleName}{\$appointments}&lt;/course&gt;</pre> <p>2-2b</p> <pre>for \$lecturer in collection("Z:\INTERROP\Session 2\2 BaseX\baseXML_tranformed.xml")//lecturer  return &lt;lecturer&gt;{\$lecturer}&lt;/lecturer&gt;</pre> <p>2-2c</p> <pre>for \$studies in collection("Z:\INTERROP\Session 2\2 BaseX\baseXML_tranformed.xml")//studies let \$studyCount := count (\$studies//study) let \$moduleCount :=count(\$studies//module) return &lt;result studyCount="{ \$studyCount}" moduleCount="{ \$moduleCount}"&gt;&lt;/result&gt;</pre> <p>2-2d</p> <pre>for \$course in collection("Z:\INTERROP\Session 2\2 BaseX\baseXML_tranformed.xml")//course let \$courseTitle := \$course/@title let \$id := \$course/@id where every \$lecturerName in \$course//lecturer satisfies starts- with(\$lecturerName/@name, "M")  return &lt;course&gt;{\$courseTitle}{\$id}&lt;/course&gt;</pre> <p>2-2e</p> <pre>for \$course in collection("Z:\INTERROP\Session 2\2 BaseX\baseXML_tranformed.xml")//course let \$courseTitle := \$course/@title let \$id := \$course/@id where some \$appointment in \$course//appointment satisfies \$appointment/@start &lt; 1200  return &lt;course&gt;{\$courseTitle}{\$id}&lt;/course&gt;</pre> <p>2-2f</p> <pre>for \$group in collection("Z:\INTERROP\Session 2\2 BaseX\baseXML_tranformed.xml")//group</pre>
----------	--

	<pre>let \$courseTitle := \$group/../../@title let \$id := \$group/../../@id let \$languages := \$group//courselanguage where data(\$languages) != "Englisch" return &lt;course&gt;{\$courseTitle}{\$id}{\$languages}&lt;/course&gt;</pre>
Source Files	DEV ENVIRONMENT: All files available at: <a href="https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/2%20BaseX">https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/2%20BaseX</a>

### Aufgabe 2-1b 1: SQL Query:

Die am längsten pro Einheit dauernde(n) Lehrveranstaltungen(n) (Stunden, Gruppe, LV-Leiter)

Steps Performed	<p>Since it is not clear what the query means semantically, the following assumptions have been made:</p> <ul style="list-style-type: none"> <li>a) Query the course that has the longest appointment/lecture in the set of lectures</li> <li>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.</li> <li>c) Stunden: the weekhours of the course are returned</li> <li>d) LV-Leiter: since multiple "Leiter" (in role of V_Verantwortlich) are possible, all lecturers are returned, resulting in duplicate entries (in case a course has 2 lecturers)</li> </ul>
Solution	<pre> 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 </pre>

	<pre> 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 ; </pre>
Result	<p>The max length for of appointments in the basexml is 615min.  2 courses match, details below.</p> <p>COURSEID: 70197  MAXAPPOINTMENTMIN: 615  COURSETITLE: Cultural history: Practice of methods in cultural history primary sources  COURSEWEEKHOURS: 20  LECTURERNAME: Diesenberger AND Schmale</p> <p>COURSEID: 70280  MAXAPPOINTMENTMIN: 615  COURSETITLE: Workshop - Practicing Digital Methods in Historical Research  COURSEWEEKHOURS: 20  LECTURERNAME: König</p>
Source Files	<p>DEV ENVIRONMENT: <a href="https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/1%20XML2RDBMS">https://github.com/fridlutz/UNIVIE/tree/master/INTERROP/Session%202/1%20XML2RDBMS</a></p> <p>SQL: <a href="https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_1%20SQL%20SELECT%20STATEMENT.sql">https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_1%20SQL%20SELECT%20STATEMENT.sql</a></p> <p>ResultSet (TXT): <a href="https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_1%20RESULTSET.txt">https://github.com/fridlutz/UNIVIE/blob/master/INTERROP/Session%202/1%20XML2RDBMS/2_1b_1%20RESULTSET.txt</a></p>

### Aufgabe 2-1b 2: SQL Query:

Die am längsten pro Semester dauernde(n) Lehrveranstaltungen(n) (Stunden, Gruppe, LV-Leiter)

Steps Performed	<p>Since it is not clear what the query means semantically, the following assumptions have been made:</p> <ul style="list-style-type: none"> <li>a) Query the course that has the longest overall durations (summed up)</li> <li>b) Group information, see above.</li> <li>c) Stunden: the weekhours of the course are returned</li> <li>d) LV-Leiter: since multiple "Leiter" (in role of V_Verantwortlich) are possible, all lecturers are returned, resulting in duplicate entries (in case a course has 2 lecturers)</li> </ul>
-----------------	---

Solution	<pre> 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 </pre>
----------	--

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

### 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 @continuousassessment
    AND COURSE.TARGETID = ASSESSMENTATTR.SOURCEID
    AND ASSESSMENTATTR.TARGETID = ASSESSMENTATTRVAL.STRINGVALUEID
    AND ASSESSMENTATTR.EDGENAME = '@continuousassessment'
    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

```

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