gkgf37

For this assignment I used BaseX 9.1.2

a. Find all Mary Poppins books which were published before WWII.

//book[@publicationDate<"1939"]

- **i.** Since all the books in the given xml were Mary Poppins books we did not worry about making sure that a given book is Mary Poppins or not. Therefore we only needed to check any books with publication dates before 1939. The publication dates were listed as attributes, not elements.
- **ii.** Since we only have Mary Poppins books in the xml file, we did not need to specify that we are looking for these. However in a larger library, where more various books are present, we would also need to specify that we are looking for Mary Poppins book (e.g. it appears in the title) according to the task.
- **iii.**

```
<book publicationDate="1934">
    <title> Mary Poppins </title>
    <author> P. L. Travers </author>
    <illustrator> Mary Shepard </illustrator>
</book>
<book publicationDate="1935">
    <title> Mary Poppins Comes Back </title>
    <author> P. L. Travers </author>
    <illustrator> Mary Shepard </illustrator>
</book>
```

b. Find all movies that have academy nominations

//movie[academyNominations]

- **i.** First of all, we are only looking for movies, while theatre and broadcast type of media is present in the original xml. Next, we are only in the existence of some/any academy notations for the given movies, but doesn't matter how many.
- **ii.** Could not find much limitations. However, if there was a movie with element <academyNominations> and value 0, then we would need to add an extra constraint that academyNominations > 0.
- **iii.**

```
<movie year="1964" type="film" language="En">
    <title> Mary Poppins </title>
    <actress> Julie Andrews </actress>
    <actor> Dick Van Dyke </actor>
    <actor> David Tomlison </actor>
    <actress> Glynis Johns </actress>
    <producer> Walt Disney </producer>
    <director> Robert Stevenson </director>
    <academyNominations> 13 </academyNominations>
</movie>
<movie year="2013" type="film" language="En">
    <title> Saving Mr. Banks </title>
```

```
            <actress> Emma Thompson </actress>
            <actor> Tom Hanks </actor>
            <academyNominations> 1 </academyNominations>
        </movie>
```

c. How many books have there been published about Mary Poppins?

```
let $titles := doc("books.xml")/books/book[contains(title,"Mary Poppins")]
return <result>{count($titles)}</result>
```

or shorter:

```
count(//book/title[contains(.,"Mary Poppins")])
```

i. Simply first we need to filter out those books that are not about Mary Poppins. To do this, we consider books about Mary Poppins which contain this in their title. Then simply count how many books we collected.

ii. The method would work with an extended database as well, we did not find any big limitations on our approach. The only one would be if there was a book about Mary Poppins but it didn't say in the title. But other than the title, there is no more information on what a book is about.

iii. <result>8</result>

iv. XPath: //book[contains(title,"Mary Poppins")]
This only gives us the books which are about Mary Poppins, however in BaseX on the top right corner it shows us the number of answers. Thus it counts the books which are about Mary Poppins: 8 results.
Note that the count() function did not work. This perhaps is due to the version of BaseX or the implemented XPath within BaseX.

d. List the books that have been published before the second Mary Poppins movie appeared.

```
let $result :=
  let $year :=
    let $x :=
      for $movies in doc("media.xml")/media/movie[contains(title,"Mary Poppins")]
      order by $movies/@year
      return $movies
    return  $x[2]/@year/data()
  for $book in doc("books.xml")/books/book[@publicationDate < $year]
  return $book
return <result>{$result}</result>
```

or shorter:

```
doc("books.xml")//book[@publicationDate<(sort(doc("media.xml")//movie[contains(titl
e,"Mary Poppins")]/@year/data())[2])]
```

i. First we needed to find the *year* at which the second Mary Poppins movie appeared. To do this we have to make an order amongst the relevant movies and choose the second oldest. Then compare each book's publication date with this *year* and list those which have lower number than that.

ii. The method works regardless the size of the database, however again as before, if the title does not contain the name Mary Poppins, then there is no way of figuring out a movie is about Mary Poppins.

iii.
```
<result>
  <book publicationDate="1934">
    <title>Mary Poppins</title>
    <author>P. L. Travers</author>
    <illustrator>Mary Shepard</illustrator>
  </book>
  <book publicationDate="1935">
    <title>Mary Poppins Comes Back</title>
    <author>P. L. Travers</author>
    <illustrator>Mary Shepard</illustrator>
  </book>
  <book publicationDate="1943">
    <title>Mary Poppins Opens the Door</title>
    <author>P. L. Travers</author>
    <illustrator>Mary Shepard</illustrator>
  </book>
  <book publicationDate="1952">
    <title>Mary Poppins in the Park</title>
    <author>P. L. Travers</author>
    <illustrator>Mary Shepard</illustrator>
  </book>
  <book publicationDate="1962">
    <title>Mary Poppins From A to Z</title>
    <author>P. L. Travers</author>
    <illustrator>Mary Shepard</illustrator>
  </book>
</result>
```

iv. First we need to put books.xml and media.xml in the same directory and then create a new database called "databases" in BaseX by selecting this directory.

```
//book[@publicationDate < db:open("databases")//movie[contains(.,"Mary Poppins")][position()=2]/@year/data() ]
```

e. How much later has the most recent Mary Poppins movie appeared, compared to the second book about Mary Poppins?

```
let $movie_year :=
  let $x :=
    for $movies in doc("media.xml")/media/movie[contains(title,"Mary Poppins")]
    order by $movies/@year descending
    return $movies
  return  $x[1]/@year/data()
```

gkgf37

```
let $book_year :=
  let $x :=
    for $book in doc("books.xml")/books/book[contains(title,"Mary Poppins")]
    order by $book/@publicationDate ascending
    return $book
  return  $x[2]/@publicationDate/data()
let $difference := $movie_year - $book_year
return <result>{$difference} years later.</result>
```

or shorter:

```
(sort(doc("media.xml")//movie[contains(title,"Mary Poppins")]/@year/data())[last()]-
sort(doc("books.xml")//book[contains(title,"Mary
Poppins")]/@publicationDate/data())[2])
```

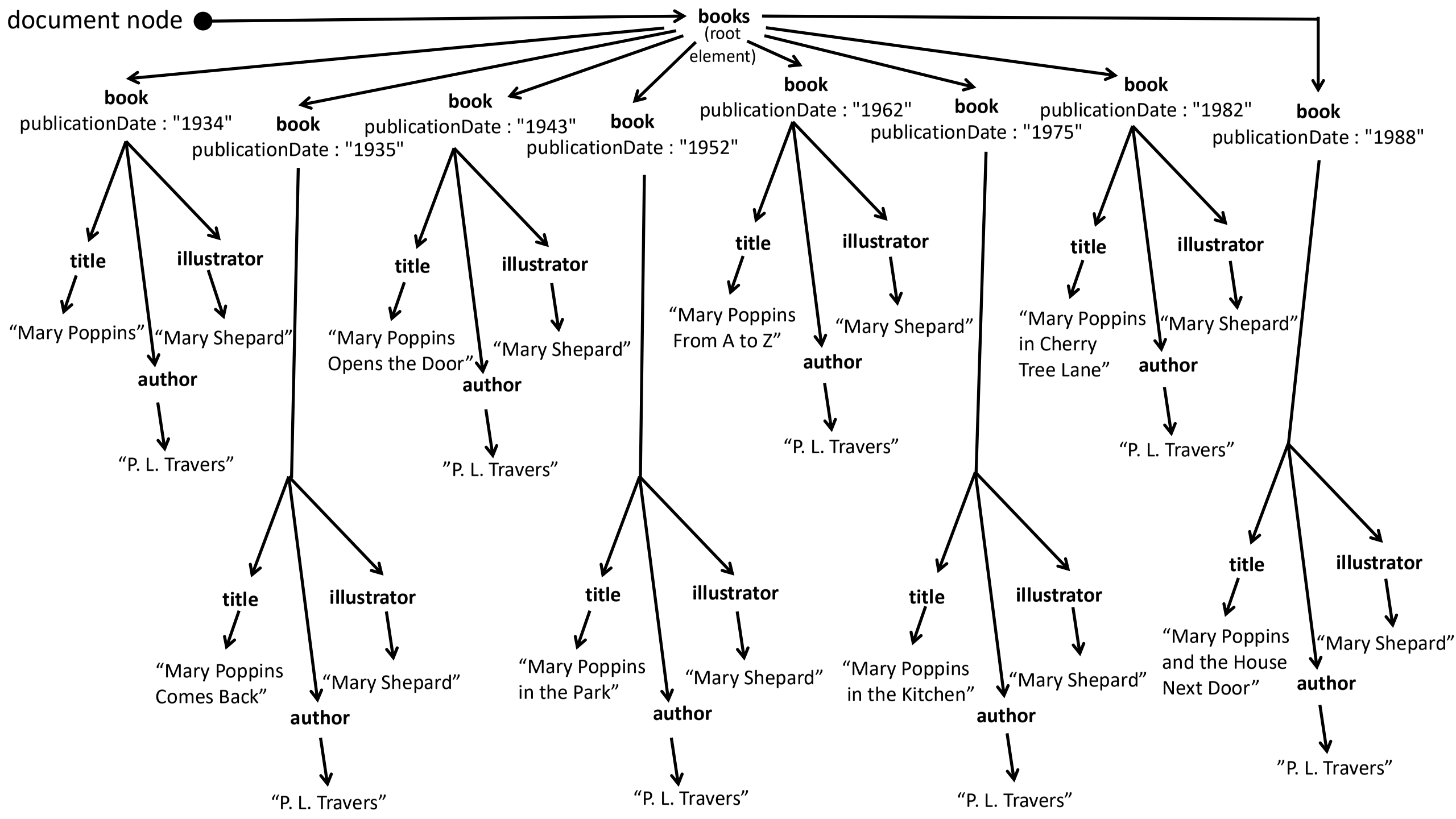
i. To get the answer we first have to find the years the latest movie and the second book were published/released, and then simply subtract the year of the book from the year of the movie.

ii. The query works with arbitrary dataset, thus with an extended version as well. No limitations found other than the ones mention before regarding whether or not the given movie or book is actually about Mary Poppins. The year and publicationDate attributes has to be given in years like in the given xml, and not in any other format.

iii. <result>83 years later.</result>

iv. First we need to put books.xml and media.xml in the same directory and then create a new database called “databases” in BaseX by selecting this directory.

```
//movie[contains(.,"Mary Poppins")][position()=last()]/@year/data() -
//book[contains(.,"Mary Poppins")][position()=2]/@publicationDate/data()
```


document node
books
(root element)
book
publicationDate : "1934"
title
"Mary Poppins"
illustrator
"Mary Shepard"
author
"P. L. Travers"
book
publicationDate : "1935"
title
"Mary Poppins Comes Back"
illustrator
"Mary Shepard"
author
"P. L. Travers"
book
publicationDate : "1943"
title
"Mary Poppins Opens the Door"
illustrator
"Mary Shepard"
author
"P. L. Travers"
book
publicationDate : "1952"
title
"Mary Poppins in the Park"
illustrator
"Mary Shepard"
author
"P. L. Travers"
book
publicationDate : "1962"
title
"Mary Poppins From A to Z"
illustrator
"Mary Shepard"
author
"P. L. Travers"
book
publicationDate : "1975"
title
"Mary Poppins in the Kitchen"
illustrator
"Mary Shepard"
author
"P. L. Travers"
book
publicationDate : "1982"
title
"Mary Poppins in Cherry Tree Lane"
illustrator
"Mary Shepard"
author
"P. L. Travers"
book
publicationDate : "1988"
title
"Mary Poppins and the House Next Door"
illustrator
"Mary Shepard"
author
"P. L. Travers"