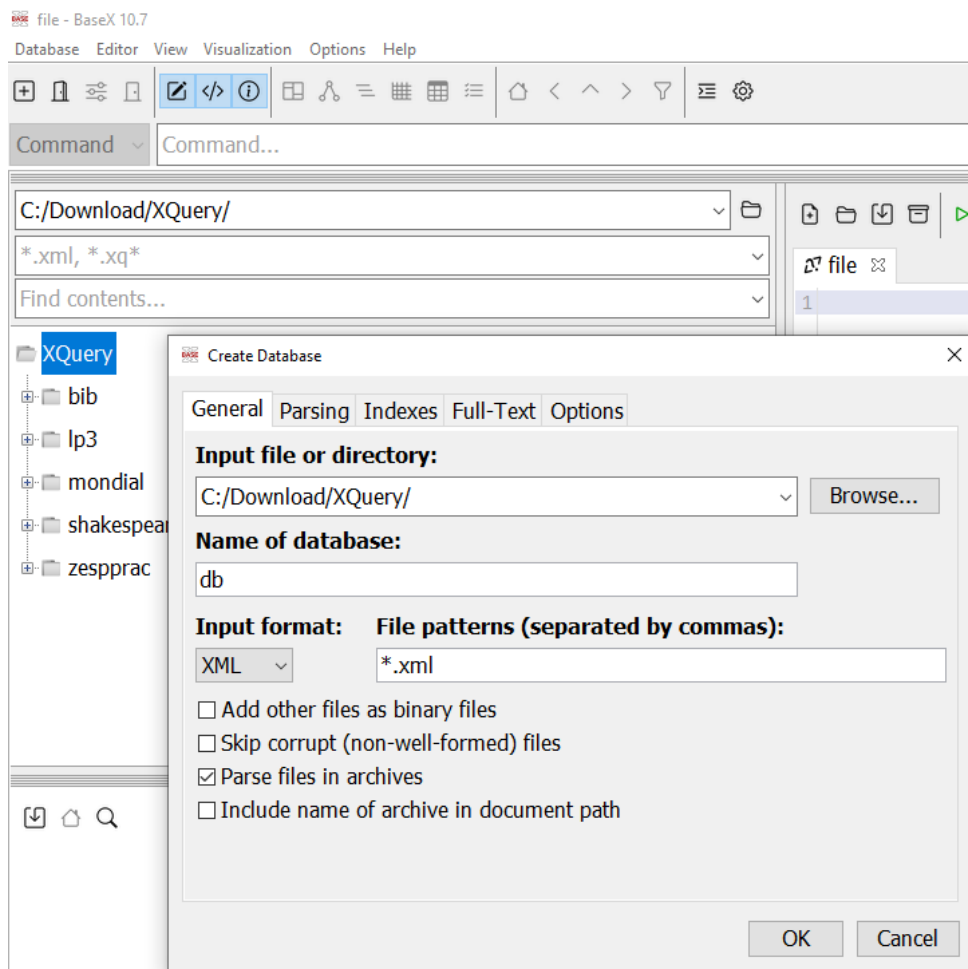


XQuery

Uruchomienie BaseX i utworzenie bazy danych

1. Pobierz oprogramowanie BaseX jako archiwum JAR ze strony <https://basex.org/download/>.
2. Uruchom serwer bazy danych BaseX z linii poleceń komendą (nazwa pliku zależy od wersji):
`java -jar BaseX107.jar`
3. Pobierz ze strony kursu archiwum XQuery.zip i rozpakuj je.
4. Wróć do GUI BaseX i utwórz nową bazę danych opcją Database->New. Powiąż ją z katalogiem, do którego zostało rozpakowane pobrane archiwum. Nazwij ją „db”, a pozostałe opcje pozostaw domyślne.



Zapytania do dokumentu z bibliografią

Na początku naszym źródłem dla zapytań będzie coś prostego. Dokument `/db/bib/bib.xml` zawiera informacje dotyczące kilku książek. Jego schemat jest następujący:

```
<!ELEMENT bib          (book* )>
<!ELEMENT book         (title, (author+ | editor+ ), publisher, price )>
<!ATTLIST book         year CDATA  #REQUIRED >
<!ELEMENT author       (last, first )>
<!ELEMENT editor       (last, first, affiliation )>
<!ELEMENT title        (#PCDATA )>
<!ELEMENT last         (#PCDATA )>
<!ELEMENT first        (#PCDATA )>
<!ELEMENT affiliation   (#PCDATA )>
<!ELEMENT publisher    (#PCDATA )>
<!ELEMENT price        (#PCDATA )>
```

5. Pobierz nazwiska wszystkich autorów książek.

Wskazówka: Dokument wskaż funkcją `doc("/db/bib/bib.xml")`.

```
<last>Stevens</last>
<last>Stevens</last>
<last>Abiteboul</last>
<last>Buneman</last>
<last>Suciu</last>
<last>Cagle</last>
<last>Fussell</last>
<last>Lopez</last>
<last>Mahany</last>
```

6. Dla każdej pary elementów **title**, **author** istniejących w jednym elemencie **book** stwórz element **książka** posiadający w sobie dwa podelementy **title**, **author**. Zwróć uwagę, że jeśli książka ma trzech autorów i jeden tytuł chcemy uzyskać 3 pary (**title**, **author**), a w przypadku książki z dwoma autora i trzema tytułami tych par powinno być sześć.

```
<książka>
  <author>
    <last>Stevens</last>
    <first>W.</first>
  </author>
  <title>TCP/IP Illustrated</title>
</książka>
<książka>
  <author>
    <last>Stevens</last>
    <first>W.</first>
  </author>
  <title>Advanced Programming in the Unix environment</title>
</książka>
<książka>
  <author>
    <last>Abiteboul</last>
    <first>Serge</first>
  </author>
  <title>Data on the Web</title>
</książka>
. . .
```

7. Dla każdej pary **title**, **author** istniejącej w ramach tego samego elementu **book** stwórz element **książka** posiadający w sobie dwa podelementy **tytuł**, **autor**. Element **autor** powinien zawierać konkatenację nazwiska i imienia autora.

```
<książka>
  <autor>StevensW.</autor>
  <tytuł>TCP/IP Illustrated</tytuł>
</książka>
<książka>
  <autor>StevensW.</autor>
  <tytuł>Advanced Programming in the Unix environment</tytuł>
</książka>
<książka>
  <autor>AbiteboulSerge</autor>
  <tytuł>Data on the Web</tytuł>
</książka>
<książka>
  <autor>BunemanPeter</autor>
  <tytuł>Data on the Web</tytuł>
</książka>
. . .
```

8. Popraw zapytanie, tak aby nazwisko i imię były rozdzielone spacją. Wskazówka: Użyj funkcji `concat()` lub operatora `||`.

```
<książka>
  <autor>Stevens W.</autor>
  <tytuł>TCP/IP Illustrated</tytuł>
</książka>
<książka>
  <autor>Stevens W.</autor>
  <tytuł>Advanced Programming in the Unix environment</tytuł>
</książka>
<książka>
  <autor>Abiteboul Serge</autor>
  <tytuł>Data on the Web</tytuł>
</książka>
<książka>
  <autor>Buneman Peter</autor>
  <tytuł>Data on the Web</tytuł>
</książka>
. . .
```

9. Zmodyfikuj poprzednie zapytanie tak aby wynik był zawarty w elemencie **wynik**. W przypadku zastosowania podzapytania pamiętaj o objęciu podzapytania klamrami `{ }`.

```
<wynik>
  <książka>
    <autor>Stevens W.</autor>
    <tytuł>TCP/IP Illustrated</tytuł>
  </książka>
  <książka>
    <autor>Stevens W.</autor>
    <tytuł>Advanced Programming in the Unix environment</tytuł>
  </książka>
  . . .
</wynik>
```

10. Wyświetl imiona autorów książki "Data on the Web".

```
<imiona>
  <imie>Serge</imie>
  <imie>Peter</imie>
  <imie>Dan</imie>
</imiona>
```

11. Uzyskaj informację o całym elemencie – książce, posiadającej tytuł "Data on the Web". Uzyskaj ten wynik na dwa sposoby: za pomocą odpowiedniego wyrażenia ścieżkowego oraz za pomocą klauzuli WHERE.

```
<DataOnTheWeb>
  <book year="2000">
    <title>Data on the Web</title>
    <author>
      <last>Abiteboul</last>
      <first>Serge</first>
    </author>
    <author>
      <last>Buneman</last>
      <first>Peter</first>
    </author>
    <author>
      <last>Suciu</last>
      <first>Dan</first>
    </author>
    <publisher>Morgan Kaufmann Publishers</publisher>
    <price>39.95</price>
    <photo>data.jpg</photo>
  </book>
</DataOnTheWeb>
```

12. Pobierz elementy nazwiska autorów książek, które w tytule posiadają ciąg znaków „Data”. Zastosuj funkcję contains().

```
<Data>
  <nazwisko>Abiteboul</nazwisko>
  <nazwisko>Buneman</nazwisko>
  <nazwisko>Suciu</nazwisko>
</Data>
```

13. Dołącz do nazwisk tytuł książki, której tytuł posiada ciąg znaków „Data”.

```
<Data>
  <title>Data on the Web</title>
  <nazwisko>Abiteboul</nazwisko>
  <nazwisko>Buneman</nazwisko>
  <nazwisko>Suciu</nazwisko>
</Data>
```

14. Wyświetl te tytuły książek, które mają nie więcej niż dwóch autorów (<=2).

```
<title>TCP/IP Illustrated</title>
<title>Advanced Programming in the Unix environment</title>
<title>The Economics of Technology and Content for Digital TV</title>
```

15. Podaj tytuły książek i liczbę ich autorów.

```
<ksiazka>
  <title>TCP/IP Illustrated</title>
  <autorow>1</autorow>
</ksiazka>
<ksiazka>
  <title>Advanced Programming in the Unix environment</title>
  <autorow>1</autorow>
</ksiazka>
<ksiazka>
  <title>Data on the Web</title>
  <autorow>3</autorow>
</ksiazka>
<ksiazka>
  <title>The Economics of Technology and Content for Digital TV</title>
  <autorow>0</autorow>
</ksiazka>
<ksiazka>
  <title>Early Adopter XQuery</title>
  <autorow>4</autorow>
</ksiazka>
```

16. Podaj przedział lat, jaki obejmują książki zawarte w kolekcji dokumentów.

```
<przedział>1992 - 2002</przedział>
```

17. Sprawdź jaka jest różnica cen pomiędzy najdroższą i najtańszą książką.

```
<różnica>94.95999999999998</różnica>
```

18. Znajdź tytuły i autorów (jeśli istnieją) najtańszych książek.

```
<najtańsze>
  <najtańsza>
    <title>Early Adopter XQuery</title>
    <author><last>Cagle</last><first>Kurt</first></author>
    <author><last>Fussell</last><first>Mark</first></author>
    <author><last>Lopez</last><first>Nalleli</first></author>
    <author><last>Mahany</last><first>Dan</first></author>
  </najtańsza>
</najtańsze>
```

19. Dla każdego autora wyświetl jego nazwisko oraz tytuły jego książek. Przyjmij, że nie ma dwóch różnych autorów o tym samym nazwisku.

```
<autor>
  <last>Stevens</last>
  <title>TCP/IP Illustrated</title>
  <title>Advanced Programming in the Unix environment</title>
</autor>
<autor>
  <last>Abiteboul</last>
  <title>Data on the Web</title>
</autor>
<autor>
  <last>Buneman</last>
  <title>Data on the Web</title>
</autor>
<autor>
  <last>Suciu</last>
  <title>Data on the Web</title>
</autor>
<autor>
  <last>Cagle</last>
  <title>Early Adopter XQuery</title>
</autor>
<autor>
  <last>Fussell</last>
  <title>Early Adopter XQuery</title>
</autor>
<autor>
  <last>Lopez</last>
  <title>Early Adopter XQuery</title>
</autor>
<autor>
  <last>Mahany</last>
  <title>Early Adopter XQuery</title>
</autor>
```

Zapytania do kolekcji dokumentów ze sztukami Szekspira

Budowa do tej pory analizowanych dokumentów była dość prosta. Spróbujmy czegoś odrobinę bardziej skomplikowanego – kolekcja /db/shakespeare zawiera dokumenty XML ze sztukami Szekspira. Dokumenty w kolekcji są zgodne z poniższym DTD.

```
<!ELEMENT PLAY (TITLE, FM, PERSONAE, SCNDESCR, PLAYSUBT, INDUCT?, PROLOGUE?, ACT+, EPILOGUE?)>
<!ELEMENT FM (P+)>
<!ELEMENT PERSONAE (TITLE, (PERSONA | PGROUP)+)>
<!ELEMENT PGROUP (PERSONA+, GRPDESCR)>
<!ELEMENT INDUCT (TITLE, SUBTITLE*, (SCENE+|(SPEECH|STAGEDIR|SUBHEAD)+))>
<!ELEMENT ACT (TITLE, SUBTITLE*, PROLOGUE?, SCENE+, EPILOGUE?)>
<!ELEMENT SCENE (TITLE, SUBTITLE*, (SPEECH | STAGEDIR | SUBHEAD)+)>
<!ELEMENT PROLOGUE (TITLE, SUBTITLE*, (STAGEDIR | SPEECH)+)>
<!ELEMENT EPILOGUE (TITLE, SUBTITLE*, (STAGEDIR | SPEECH)+)>
<!ELEMENT SPEECH (SPEAKER+, (LINE | STAGEDIR | SUBHEAD)+)>
<!ELEMENT LINE (#PCDATA | STAGEDIR)*>
```

Wszystkie elementy bez definicji są proste (#PCDATA).

20. Wyświetl tytuły wszystkich sztuk.

Wskazówka: Kolekcję dokumentów wskaż funkcją collection("db/shakespeare").

```
<wynik>
  <TITLE>The Tragedy of Hamlet, Prince of Denmark</TITLE>
  <TITLE>The Tragedy of Macbeth</TITLE>
  <TITLE>The Tragedy of Othello, the Moor of Venice</TITLE>
  <TITLE>The Tragedy of Romeo and Juliet</TITLE>
</wynik>
```

21. Wyświetl tytuły sztuk, w których pada fraza “or not to be” (tj. jest zawarta w którymś z elementów line).

```
<TITLE>The Tragedy of Hamlet, Prince of Denmark</TITLE>
```

22. Dla każdej sztuki podaj jej tytuł oraz krótką statystykę (liczba postaci, liczba aktów, liczba scen).

```
<wynik>
  <sztuka tytuł="The Tragedy of Hamlet, Prince of Denmark">
    <postaci>26</postaci>
    <aktow>5</aktow>
    <scen>20</scen>
  </sztuka>
  <sztuka tytuł="The Tragedy of Macbeth">
    <postaci>28</postaci>
    <aktow>5</aktow>
    <scen>28</scen>
  </sztuka>
  ...
</wynik>
```