XML Query Language/Beispiele

Wechseln zu:Navigation, Suche

1 Beispiele

Die folgenden XQuery-Beispiele wurden mit http://ahnen.multimedia.fh-augsburg.de:8080/exist/xmldb/db/ getestet.

Finde alle Personen in der Datenbank:

```
(: Alle Personen :)
//person
```

Finde die Person Wolfgang in der Datenbank:

```
(: Wolfgang :)
//person[pid="1"]

(: Wolfgang :)
//person[lastname="Kowarschick" and firstname="Wolfgang"]
```

Finde die Wolfgangs Mutter in der Datenbank:

```
(: pid von Wolfgangs Mutter :)
//person[lastname="Kowarschick" and firstname="Wolfgang"]/mother/text()

(: Wolfgangs Mutter:)
//person[pid=//person[lastname="Kowarschick" and
firstname="Wolfgang"]/mother/text()]
```

Eine einfache Let-Anweisung:

```
(: let-Anweisung (Konstanten-Deklaration) :)
let $data := <person><pid>wk</pid></person>
return $data
```

Eine String-Transformation:

```
(: transformiere String in XML :)
declare namespace util="http://exist-db.org/xquery/util";
let $data as xs:string := "<person><pid>wk</pid></person>"
return util:eval($data)
```

Eine einfache For-Anweisung:

```
(: for-Anweisung :)
for $person in collection('/db/ahnen')//person
return string($person/lastname)
```

Erzeugung eines XML-Dokuments:

```
(: Erzeuge ein neues XML-Dokument. :)
let $person := //person[pid="1"]
let $lastname  as xs:string := $person/lastname/text()
let $firstname as xs:string := $person/firstname/text()
let $year      as xs:string := $person/birthday/year/text()
let $month     as xs:string := $person/birthday/month/text()
let $day       as xs:string := $person/birthday/day/text()
return
  <Person id="{$person/pid/text()}">
    <Name> {$firstname} {" "} {$lastname} </Name>
    <Geburtstag>{$day}. {$month}. {$year}</Geburtstag>
  </Person>
```

Erzeugung eines komplexeren XML-Dokuments:

```
(: Erzeuge ein neues XML-Dokument mit for-Schleife.
   $person/lastname/text() liefert Fehler bei leeren Tags
=> string($person/lastname) funktioniert immer
:)
<Personen>
{
  for $person in collection('/db/ahnen')//person  (:
[lastname="Kowarschick"] :)
  let $lastname  as xs:string := string($person/lastname)
  let $firstname as xs:string := string($person/firstname)
  let $year      as xs:string := string($person/birthday/year)
  let $month     as xs:string := string($person/birthday/month)
  let $day       as xs:string := string($person/birthday/day)
  return
    <Person id="{$person/pid/text()}">
      <Name> {$firstname} {" "} {$lastname} </Name>
      <Geburtstag>{$day}. {$month}. {$year}</Geburtstag>
    </Person>
}
</Personen>
```

Berechne die Vorfahren von Sibylle:

```
(: Berechne die Vorfahren einer oder mehrerer Personen :)
declare namespace gen="http://www.fh-augsburg.de/xmlns/genealogy";
declare function gen:ancestors($pid as xs:string) as node()*
{
  <person>
  {
    //person[pid=$pid]/*
      [not(local-name()='mother') and not(local-name()='father')]
  }
    <mother>
    {
      let $mid as xs:string := //person[pid=$pid]/mother
      return
      if ($mid) then gen:ancestors(string($mid)) else ()
    }
    </mother>
    <father>
    {
      let $fid as xs:string := //person[pid=$pid]/father
      return
      if ($fid) then gen:ancestors(string($fid)) else ()
    }
    </father>
  </person>
};
<ancestors>
  {
    for $person in //person[lastname='Kowarschick'][firstname='Sibylle']
    return gen:ancestors(string($person/pid))
  }
</ancestors>
```

Berechne die Vorfahren von bestimmten Personen:

```
(: Berechne die Vorfahren einer oder mehrerer Personen :)
declare namespace gen="http://www.fh-augsburg.de/xmlns/genealogy";
declare namespace util="http://exist-db.org/xquery/util";
declare function gen:ancestors($pid as xs:string) as node()*
{
  <person>
  {
    //person[pid=$pid]/*
      [not(local-name()='mother') and not(local-name()='father')]
  }
    <mother>
    {
      let $mid as xs:string := //person[pid=$pid]/mother
      return
      if ($mid) then gen:ancestors(string($mid)) else ()
    }
    </mother>
    <father>
    {
      let $fid as xs:string := //person[pid=$pid]/father
      return
      if ($fid) then gen:ancestors(string($fid)) else ()
    }
    </father>
  </person>
};
let $firstname as xs:string := ""      (: gesuchter Vorname  :)
let $lastname  as xs:string := ""      (: gesuchter Nachname :)
let $year      as xs:string := "1987"  (: gesuchtes Geburtsjahr :)
(:
let $firstname as xs:string := "Wolfgang"
let $lastname  as xs:string := "Kowarschick"
let $year      as xs:string := ""
:)
(:
let $firstname as xs:string := ""
let $lastname  as xs:string := ""
let $year      as xs:string := "1961"
:)
(:
let $firstname as xs:string := ""
let $lastname  as xs:string := "Kowarschick"
let $year      as xs:string := ""
:)
let $lastname_condition as xs:string :=
  if ($lastname != "" and $lastname != "-")
  then concat("[lastname='",      $lastname,  "']") else ""
let $firstname_condition as xs:string :=
  if ($firstname != "" and $firstname != "-")
```

```
  then concat("[firstname='",     $firstname, "']") else ""
let $year as xs:string :=
  if ($year != "" and $year != "-")
  then concat("[birthday/year='", $year,      "']") else ""
let $query as xs:string :=
  concat("//person", $lastname_condition, $firstname_condition, $year,
"")
return
<ancestors>
  <!-- <query>{$query}</query> -->
  {
    for $person in util:eval($query)
    return gen:ancestors(string($person/pid))
  }
</ancestors>
```

2 Quelle

Die Beispiele wurden von Wolfgang Kowarschick entwickelt.

3 Siehe auch

http://www.datadirect.com/developer/xml/xquery/docs/katz_c01.pdf
http://www.w3schools.com/xquery/default.asp
http://www.research.ibm.com/journal/sj/414/chamberlin.pdf

Dieser Artikel ist GlossarWiki-konform.

Kategorien:
Seiten mit Syntaxhervorhebungsfehlern
XML
Beispiel