

## XQuery における FAQ

XML コンソーシアム XMLDB 勉強会  
凸版印刷株式会社 伊藤勝広

### Q 1 . XQuery を使うと何が出来ますか？

A 1 .

“ XQuery ” とは、W3C にて勧告（2007年1月23日）された XML 文書に対する問い合わせ言語（クエリー言語）です。

リレーショナルデータベース（RDB）に対する SQL のように、XML 文書に対して検索 / 抽出を行う事が出来ます。

加えて、XQuery は高度な処理機能も備えており、異種かつ複数の XML 文書群から抽出した結果に対して、統合、操作、変換、フィルタリング等の処理を行う事も出来ます。

そのため XQuery は、現在では多くの XML データベース（XMLDB）に対して実装が進んでおり、今後 XMLDB における標準の問い合わせ言語になる事が予想されます。

### Q 2 . XQuery の構文はどのようなものですか？

A 2 .

XQuery は「表現式（Expression）」という単位で構成されており、この表現式を組み合わせる事で XML 文書群に対して問い合わせを行う事が出来る関数型言語です。

表現式には下記に示すようなものがあります。

- 1 . FLWOR 表現式
- 2 . XPath 式
- 3 . Unordered 表現式
- 4 . Conditional 表現式
- 5 . Quantified 表現式
- 6 . Typeswitch 表現式
- 7 . Treat 表現式
- 8 . Validate 表現式

中でも FLWOR 表現式は、SQL における SELECT、FROM、WHERE 句のようなものに該当し、FOR、LET、WHERE、ORDER BY、RETURN 句にて、複数の XML 文書に対して“選択”、“射影”、“直積”、“結合”等を実行する事が出来ます。

また XQuery 1.0 は XPath 2.0 をサブセット（部分集合）として含んでおり、各要素や属性、テキストを指定したり、条件を指定して該当するデータを取り出したりするには Xpath 式を使用します。

### Q3 . XQuery を学ぶにはどのような方法がありますか？

A3 .

書籍や Web サイトから知識を得る方法の他に、実際に XQuery の操作を体験してみる方法があります。

参考として、その中の幾つかをご紹介します。

#### 【 書籍 】

- ・改訂版 標準 XML 完全解説 (下)  
著者：奥井康弘、中山幹敏  
出版：技術評論社
- ・XML データベース入門  
NeoCore/Xpiori で XMLDB を極める！  
著者：山田祥寛  
出版：翔泳社
- ・XML データベース入門以前  
著者：糸魚川茂夫  
出版：毎日コミュニケーションズ
- ・XQuery+XML データベース入門  
DB2 9 無償版で実感する XML DB 用標準クエリ言語の基本  
著者：菅原香代子、米持幸寿  
出版：日経 BP 出版センター

#### 【 Web サイト 】

- ・XQuery 1.0、XSLT 2.0、XPath 2.0 の各仕様群の公開について (W3C 勧告)(日本語)  
<http://www.w3.org/2007/01/qt-pressrelease>
- ・XML Query (XQuery) Requirements (英語)  
<http://www.w3.org/TR/xquery-requirements/>
- ・XML Query Use Cases (英語)  
<http://www.w3.org/TR/xquery-use-cases/>
- ・XML Query Use Cases (日本語)  
2002年11月15日ワーキングドラフト時の和訳  
<http://www.xmlconsortium.org/wg/tech/WD-xmlquery-use-cases-20021115-Japan-clean.htm>
- ・XMLDB 勉強会 技術系サブグループ ドキュメンテーションチーム サイト (日本語)  
<http://xmldb-study.discypus.jp/trac/>
- ・アットマーク・アイティ XQuery チュートリアル (日本語)  
<http://www.atmarkit.co.jp/fxml/tanpatsu/19quip/quip01.html>

【 XQuery を体験出来るフリーツール 】

- ・ eXist  
オープンソースのネイティブ XML データベース  
Java で書かれている為、多くのプラットフォームで動かす事が可能  
<http://exist.sourceforge.net/>
- ・ Qizx/open  
Java で記述された、XML クエリに準拠したオープンソースツール  
スキーマのインポートを除き、XQuery の機能をほぼインプリメンテーションしている  
<http://www.axyana.com/qizxopen/>
- ・ XQuisitor  
Java で動く XQuery を試せる GUI クエリー・ツール  
<http://www.cafeconleche.org/xquisitor/>

ライセンスフリーのツールは上記の他にもまだあります。  
また各ベンダーからも、XQuery を体験出来る無償版や試用版の XMLDB , ツール等が提供されています。

ご興味のある方はインターネット等で探してみてください。

**Q 4 . XQuery にて処理を行うのと、XPath で処理を行うのでは何が違いますか？**

A 4 .

XPath は単体の XML 文書しか扱う事が出来ない為、例えば、複数の XML 文書から抽出した結果から別のデータ構造を構築する場合には、抽出した結果をその後 XSLT や DOM などに渡して別途処理（結合、統合、変換）しなければなりません。

XQuery ならば複数の XML 文書にアクセスし、必要な情報の抽出から、結合、統合、変換、フィルタリングまでを一括でまかなう事が可能です。

**Q 5 . XMLDB 製品毎に XQuery の実装に差（方言）はありますか？**

A 5 .

XMLDB 製品の中には、ワーキングドラフト（仕様勧告前）の段階で既に XQuery をサポートしていたものがあります。

その為、実装されている関数や表現式、及びその表し方が製品毎で若干異なる場合があります。

しかし XQuery 仕様が W3C から正式に勧告された事により、今後は各製品ともその仕様に則った形に統一されていく事が予想されます。

## Q 6 . FLWOR 式について詳しく教えてください。

A 6 .

FLWOR 式は「フラワー式」と呼びます。

基本構文は以下のようになります。

for \$変数名 in Path 表現式	}	何れかの内の1つ以上が必須
let \$変数名 := 式		
where 条件式	.....	省略可
order by 式	.....	省略可
return 式	.....	必須

### 【 for 句 】

変数に複数のシーケンス<sup>1</sup>を1つずつ代入し同じ処理を複数回繰り返す場合に使用

for 句は、in 句の Path 表現式が示すシーケンスからノードを順番に取り出し、変数にそのノードを格納します。

ノードが格納された変数はタプルを作成し、作成されたタプルは順番を持ったシーケンスとなります。

また for 句に複数の Path 表現式を列挙した場合には、それぞれのシーケンスに含まれるノード同士の全ての組み合わせ（直積）が作られます。

(例) for \$val1 in (<A/>,<B/>) , \$val2 in (<C/>,<D/>)

上記の例の場合は変数 \$val1、\$val2 の組み合わせで、(<A/>,<C/>)、(<A/>,<D/>)、(<B/>,<C/>)、(<B/>,<D/>)として4回繰り返されます。

更に for 句は、他の FLWOR 式を入れ子にする事も出来ます。

次の例は「XML 出版」から発刊されている本の著者名 (author) を出力します。

(例)

```
for $x in let $y := doc("books.xml")//books/book[publisher/text()="XML 出版"]
return
distinct-values($y/author/text())
return $x
```

「distinct-values() 関数」は、重複した値を取り除く時に用いる関数です。

「doc() 関数」は、外部ドキュメントを取得する時に使用する関数です。

## 【 let 句 】

変数にシーケンスを代入する際に使用

次の例では、books.xml という名前の XML 文書を取り出し、その中に記されている本の題名 (title) を \$x に、本の価格 (price) を \$y にそれぞれ代入します。

```
(例) let $x := doc("books.xml")//books/book/title/text() ,
      $y := doc("books.xml")//books/book/price/text()
```

## 【 where 句 】

for 句、let 句で抽出された内容に対してフィルタリングするのに使用

Path 表現式がオリジナルの XML 文書に対してフィルタをかけるのに対して、where 句は for 句及び let 句で作成されたタプルに対してフィルタをかけます。

対象のノードに対して条件を適用し、真 (true) であった場合には対象ノードを残し、偽 (false) であった場合には対象ノードを除外します。

条件式には、"and" や "or" と書いた論理演算子や、">" や "!=" と書いた比較演算子を用いて、複数の条件を指定する事が出来ます。

次の例では、価格が 500 円以上 1000 円未満の本のタイトルを出力します。

```
(例) for $x in /books/book
      where $x/price >= 500 and $x/price < 1000
      return $x/title
```

また where 句においても、入れ子を表現する事が可能です。

where 句の場合の入れ子は「そのサブクエリの結果が空シーケンスであるかどうか？」を判断してフィルタをかけるだけです。

次の例では、本の題名 (/books/book/title) 要素 (\$y) が存在すれば true となり、存在しなければ (空ならば) false となります。

つまり \$y が存在すれば \$x が出力されます。

```
(例) for $x in /books/book
      where for $y in $x/title return $y
      return $x
```

ところで where 句は、あくまで for 句や let 句でいったん抽出された内容に対してのフィルタリングという事になります。

つまり、いったん不要なデータも含めて取得した後に、再度条件判断処理を行う事になる為パフォーマンス的に劣る事になります。

可能であれば、for 句内の Path 表現式のみで条件を絞り込むのが好ましいと言えるでしょう。

**【 order by 句 】**

処理する順番を指定する場合に使用

order by 句には、ソート（並べ替え）対象となるノードを指定します。

ソートキーとして **ascending** を指定すれば昇順に、**descending** を指定すれば降順にソートされます。

ソートキーを省略した場合には、昇順にソートされます。

ソート条件はカンマで区切って複数記述する事が出来ます。

第一引数、第二引数、.....の順でソートの優先順位が決定されます。

次の例では、まず本の価格（**price**）でソート（昇順）され、価格（**price**）が同じものについては、更に題名（**title**）でソート（降順）されます。

```
(例) for $x in /books/book
      order by $x/price, $x/title descending
      return $x
```

**【 return 句 】**

処理結果の生成に使用

return 句は、for 句や let 句で作成された 1 タプルに対して 1 回呼び出されます。

出力結果はコンストラクタ式<sup>2</sup>で記述します。

コンストラクタ式では、変数を用いて要素、属性、要素の値、属性の値をバインドする事が出来ます。

加えて、直接定数を埋め込む事も出来ます。

変数は { } で括り埋め込みます。

また { } の中には、for 句や let 句を埋め込みネストする事も出来ます。

次の例では、\$x と \$y の値が等しくなる (<B/>, <B/>) の組み合わせを除外し、[ 出力結果 ] の構造で出力されます。

```
(例) for $x in (<A/>, <B/>)
      for $y in (<B/>, <C/>)
      where $x != $y
      return
        <output>{$x}{$y}</output>
```

[ 出力結果 ]

```
<output><A/><B/></output>
```

```
<output><A/><C/></output>
```

```
<output><B/><C/></output>
```

ここまで FLWOR 式について簡単に記してきましたが、より詳しい事がお知りになりたい場合には、先にご紹介させて頂いた「XML Query Use Cases」に記載されている例題等が参考になると思います。

- 1 シーケンス :  
0 個以上のノード (要素や属性) もしくは値の集合。 順序が保持されている。
- 2 コンストラクタ :  
データとそれを操作する為の手続きが一体化したオブジェクト (データ構造)。

### Q 7 . XMLDB においてデータ抽出 (検索) は XQuery が標準になりつつあるようですが、データ投入や更新に対する標準の言語はありますか？

A 7 .

データ更新や新規投入については、現在のところ標準と呼べるものではありません。製品毎に、XPath の機能を拡張させたり、オリジナルの方法で対応したりしているのが現状です。

その一方で、XQuery から「XQuery Update Facility」というタイトルで、データ更新系処理に関するワーキングドラフトの草案が提示されております。

XQuery Update Facility (英語)

<http://www.w3.org/TR/xqupdate/>

「XQuery Update Facility」ではデータを操作する事を目的とした、INSERT、DELETE、REPLACE、RENAME、TRANSFORM と言う 5 つの構文と機能が定義されております。

この内、最初の 4 つは既存の XML 文書を更新 (書き換え) する為の機能として規定されており、TRANSFORM は新しい XML 文書を作成する為の機能として規定されています。

各機能の詳細については上記の草案の本文を参考にして頂くとして、ここでは簡単な例による紹介だけをさせていただきます。

#### 【 INSERT 】

既存の XML 文書に新たなノードを挿入する為の機能

次の例では、5 番目の/school/student 要素の中の name 要素の後に number 要素を挿入する事を表しています。

```
(例) do insert <number>25</number>
      after /school/student[5]/name
```

**【 DELETE 】**

XML 文書からノードを削除する為の機能

次の例では、最初の/books/book 要素の中の subtitle 要素を削除する事を表しています。

```
(例) do delete /books/book[1]/subtitle
```

**【 REPLACE 】**

XML 文書中のノードもしくは値の置き換えを行う為の機能

次の例では、最初の/books/book 要素の中の publisher 要素以下のノードを 2 番目の/books/book 要素の中の publisher 要素以下のノードで置き換えます。

```
(例) do replace /books/book[1]/publisher  
with /books/book[2]/publisher
```

**【 RENAME 】**

XML 文書中の要素名 ( タグ名 ) の書き換えを行う為の機能

次の例では、2 番目の/books/book 要素の中の publisher 要素の名前を “ new-publisher ” に置き換えます。

```
(例) do rename /books/book[2]/publisher  
as “ new-publisher ”
```

**【 TRANSFORM 】**

XML 文書中の一部のノードをコピーし、それに対して変換を行い新しいノードを作成する為の機能

次の例は、book 要素から price が 1000 未満のものだけを一時的に \$org\_node にコピーし、その中から discount 要素を削除し結果を返します。

```
(例) for $org_node in //book[price < 1000]  
return  
transform  
copy $new_node := $org_node  
modify do delete $new_node/discount  
return $new_node
```

**Q 8 . XQuery の他に、XMLDB に関連した W3C 規格にはどのようなものがありますか？**

A 8 .

現在、以下のような規格が勧告されたり策定が進められていたりしています。

**【 XPath 】**

XML 文書の中から、条件に適合した部分の集合を取り出すための仕様です。

1999年11月16日にバージョン 1.0 が、2007年1月23日にバージョン 2.0 が、それぞれ W3C より勧告されております。

XML Path Language ( XPath ) Version 1.0 ( 英語 )

<http://www.w3.org/TR/1999/REC-xpath-19991116>

XML Path Language ( XPath ) バージョン 1.0 ( 日本語 )

<http://www.infoteria.com/jp/contents/xml-data/REC-xpath-19991116-jpn.htm>

XML Path Language ( XPath ) 2.0 ( 英語 )

<http://www.w3.org/TR/2007/REC-xpath20-20070123>

**【 XUpdate 】**

データ更新専用の言語。

XML を用いてデータベースを更新するリクエストを記述する標準言語として、「The XML:DB Initiative」が開発中です。

XML:DB Initiative ( 英語 )

<http://www.xmldb.org/xupdate/index.html>

<http://www.xmldb.org/xupdate/xupdate-wd.html>

**【 XQuery 1.0 and XPath 2.0 Full-Text 】**

XML 文書に対して全文検索を行う事を目的とした、XQuery と XPath を拡張する言語の定義がなされています。

現在仕様の草案がリリースされています。

XQuery 1.0 and XPath 2.0 Full-Text ( 英語 )

<http://www.w3.org/TR/2006/WD-xquery-full-text-20060501>

XQuery 1.0 and XPath 2.0 Full-Text Use Cases ( 英語 )

<http://www.w3.org/TR/2006/WD-xmlquery-full-text-use-cases-20060501>

## 参考文献

- ・山田祥寛著 『XML データベース入門 NeoCore/Xpiori で XMLDB を極める!』  
翔泳社 (2006年)
- ・糸魚川茂夫著 『XML データベース入門以前』  
毎日コミュニケーションズ (2006年)
- ・W3C 『XML Query Use Cases』  
<http://www.w3.org/TR/xquery-use-cases>  
(アクセス: 2007/4/23)
- ・W3C 『XQuery Update Facility』  
<http://www.w3.org/TR/xqupdate>  
(アクセス: 2007/4/23)
- ・日本 IBM 『XQuery の神話と誤解を暴く』  
[http://www-06.ibm.com/jp/developerworks/xml/050603/j\\_x-xqmyth.html](http://www-06.ibm.com/jp/developerworks/xml/050603/j_x-xqmyth.html)  
(アクセス: 2007/4/23)
- ・日本 IBM 『11/22 DB2 9 新機能概要 & XML ハンズオンセミナー(ViperPOT)資料  
4.XQuery 解説.pdf』  
<http://www-06.ibm.com/jp/domino01/mkt/dminfo.nsf/doc/001E250E>  
(アクセス: 2007/4/23)
- ・日本 BEA システムズ 『XML クエリ言語』  
<http://edocs.beasys.co.jp/e-docs/liquiddata/docs11/faq/faq.html#831918>  
(アクセス: 2007/4/23)
- ・SE 虎の巻 - XML HTML CSS リファレンス 『XML 入門 XQuery』  
<http://itref.fc2web.com/xml/xquery.html>  
(アクセス: 2007/4/23)
- ・XML コンソーシアム 『XML Query Use Cases ワーキングドラフト翻訳』(2002/11/15時点)  
<http://www.xmlconsortium.org/wg/tech/WD-xmlquery-use-cases-20021115-Japan-clean.htm>  
(アクセス: 2007/4/23)