

# DB2 XQuery

[XML][XQuery]

- ・ DB2 XML DB (pureXML)
- ・ IBM のマニュアル からのメモ
- ・ World Wide Web Consortium (W3C) によって設計された機能プログラミング言語

## XML データ

柔軟

- ・ XML データは柔軟で、予測不能で、散在しており、自己記述的

XQuery を使うしかない局面

- ・ 階層のどのレベルにあるか不明なオブジェクトの XML を検索
- ・ データにタイして構造変換を実行
- ・ タイプが混合した結果を戻す
- ・ 既存の XML を更新

## 照会のコンポーネント

プロローグと照会本体からなる

- ・ プロローグには、照会処理環境を定義する一連の宣言

```
<!-- プロローグ -->
xquery version "1.0";
declare default element namespace "http://posample.org";
<!-- 照会本体 -->
<price_list>{for $prod in db2-fn:xmlcolumn("PRODUCT.DESCRPTION")/product/description
order by xs:decimal($prod/price) descending
return <product>{$prod/name, $prod/price}</product>}
</price_list>
```

## XQuery と SQL/XML

- ・ XQuery を直接呼び出す照会は、キーワード XQUERY で始まる。
- ・ SQL/XML では、以下の関数と述部を指定して SQL から XQuery を呼び出す

キーワード / 関数 / 述部	内容
XQUERY	<u>XQuery</u> が使用されることを示し、大 / 小文字を区別する規則をサーバーに指示
<u>XMLQUERY</u>	引数として <u>XQuery</u> 式を使用し、 <u>XML</u> シーケンスを戻す関数
<u>XMLTABLE</u>	<u>XQuery</u> 式を使用して <u>XML</u> データから <u>SQL</u> の表を生成関数

XML EXISTS	XQuery 式が 1 つ以上の項目のシーケンスを戻すか判別する述部
------------	------------------------------------

## XQuery 関数を使用した DB2 データの検索

- ・ XQuery では以下のいずれかの関数で DB2 から XML データを取得できる

関数	概要	例
db2-fn:xmlcolumn	XML 列を識別するリテラル引数を使用	db2 -fn:xmlcolumn("BUSINESS.ORDERS.PURCH
db2-fn:sqlquery	fullselect を表す引数を使用。 SQL の機能を使用できる	db2-fn:sqlquery("SELECT purchase_order FROM business.orders WHERE ship_date = '2005-06-15' ")/shipping_address/city

SQL は大文字、小文字を区別しないが、XML は区別する。SQL は通常、自動的に大文字に変換されるが、XML と混在する場合は明示的に大文字として記述する必要がある。ただし、二重引用符で囲むことで、SQL でも大文字小文字を区別するようにすることもできる。

## XQuery および XPath のデータ・モデル

### XDM

- ・ XQuery 式は、XQuery および XPath のデータモデル XDM のインスタンスに対して作動し、XDM のインスタンスを戻す。
- ・ XDM はアトミック値およびノードのシーケンスで説明される

### シーケンスおよび項目

- ・ シーケンスとは 0 個以上の項目の順序づけられたコレクション
- ・ 項目はアトミック値またはノード

### シーケンスの記法

- ・ 各項目はカンマで区切る
- ・ 全体は括弧で囲む
- ・ 空の括弧は、空のシーケンス
- ・ 単独の項目は、一つの項目を含むシーケンスと等価
- ・ ネストできない

### 例

- ・ 36
- ・ <dog/>
- ・ (2,3,4)
- ・ (36,<dog/>,"cat")
- ・ ()

## アトミック値

- ・組み込みアトミックデータ・タイプの1つのインスタンス
- ・ストリング、整数、10進数、日付などこれ以上分割できない

## ノード階層

- ・シーケンスのノードは、1つ以上の階層を形成する。
- ・DB2 は、文書、エレメント、属性、テキスト、処理命令、コメントノードをサポート

```
<products>
  <product pid="10">
    <description>
      <name>Fleece jacket</name>
      <price>19.99</price>
    </description>
  </product>
  <product pid="11">
    <description>
      <name>Nylon pants</name>
      <price>9.99</price>
    </description>
  </product>
</products>
```

- ・products というルートエレメントが含まれる
- ・ルートエレメントに product エレメントが含まれる
- ・product エレメントには、pid という属性と、description という子エレメントを持つ

## ノードのプロパティ

- ・各ノードは以下の1つ以上のプロパティを持つことができる

プロパティ	内容
node-name	QName として表現されるノードの名前
parent	現行ノードの親ノード
type-name	ノードの動的タイプ
children	現行ノードの子であるノードのシーケンス
attributes	現行ノードに属する属性ノードのセット
string-value	ノードから抽出可能なストリング値
typed-value	ノードから抽出可能な0個以上のアトミック値のシーケンス
in-scope namespaces	ノードに関連づけられた範囲のネーム・スペース
content	ノードの内容

## DB2 がサポートする6つのノード種類

ノード種類	内容
文章ノード	<u>XML 文章</u> をカプセル化

エレメントノード	<u>XML</u> エレメントをカプセル化
属性ノード	<u>XML</u> 属性を意味する
テキストノード	<u>XML</u> の文字内容をカプセル化
処理命令ノード	<u>XML</u> 処理命令をカプセル化
コメントノード	<u>XML</u> コメントをカプセル化

## XDM のシリアライゼーション

- ・ XQuery 式の結果は XDM のインスタンスだが、シリアライゼーションを行い、XML 表記に変換可能
- ・ XMLSERIALIZE 関数でも実行できる

## XML ネームスペースと QName

- ・ XML ネームスペースは、命令の衝突を回避
- ・ XML ネームスペースは、URI によって識別される名前のコレクション
- ・ XQuery のエレメント、属性、データタイプ、関数に使用される名前の修飾方法が提供される

ネームスペース接頭部で修飾された名前が修飾名 (QName)

### QName の構成

- ・ オプションのネームスペース接頭部 + ローカル名 (: で区切る) で構成
- ・ 照会の処理時に、ネームスペース接頭部にバインドされている URI を解決

QName は、W3C 勧告 Namespace in XML で定義される構文に準拠

静的に既知のネームスペース

### 事前宣言されたネームスペース

接頭部	<u>XML</u>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
xsi	<a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a>
fn	<a href="http://www.w3.org/2005/xpath-functions">http://www.w3.org/2005/xpath-functions</a>
xd	<a href="http://www.w3.org/2005/xpath-datatypes">http://www.w3.org/2005/xpath-datatypes</a>
db2-fn	<a href="http://www.ibm.com/xmlns/prod/db2/functions">http://www.ibm.com/xmlns/prod/db2/functions</a>

静的に既知のネームスペースセットを提供

- ・ ネーム・スペース宣言またはデフォルト・ネーム・スペース宣言を使用

```
declare namespace ns1 = "http://mycompany.com";
```

接頭部 ns1 が URI <http://mycompany.com> と関連付けられる

- ・ エレメント・コンストラクターのネーム・スペース宣言属性によって宣言

```
<ns2:price xmlns:ns2="http://mycompany.com">14.99</ns2:price>
```

エレメントの有効範囲内で接頭部 ns2 を URI <http://mycompany.com> にバインド

- ・ SQL/XML によって提供
  - ・ SQL/XML の事前宣言されたネーム・スペース
  - ・ SQL/XML コンストラクターおよびその他の SQL/XML 式で宣言されるネーム・スペース

## タイプ・システム

- ・ XQuery は強く型付けされた言語
- ・ XQuery のタイプ・システムには、XML スキーマの組み込みタイプおよび、XQuery の事前定義タイプが含まれる
- ・ XML スキーマの組み込みタイプは、<http://www.w3.org/2001/XMLSchema> にあり、接頭部 xs を持つ
  - ・ xs:integer、xs:string など
- ・ XQuery 事前定義タイプは、<http://www.w3.org/2005/xpath-datatypes> にあり、接頭部 xdt を持つ
  - ・ xdt:untypedAtomic、xdt:yearMonthDuration など

## タイプ階層

- ・ すべてのアトミック・タイプは、データ・タイプ xdt:anyAtomicType から派生

## プロローグ

- ・ 照会の処理環境を定義する一連の宣言

### 宣言

宣言	内容	例
バージョン宣言	<u>XQuery</u> の構文およびセマンティクスのバージョン	xquery version "1.0";
境界スペース宣言	エレメント・コンストラクターによる境界空白の処理方法	declare boundary-space preserve;
構成宣言	構成モードを設定	declare construction strip;
Copy-namespaces 宣言	ネーム・スペースのバインディングを制御	declare copy-namespaces preserve, inherit;
デフォルトのエレメント / タイプのネーム・スペース宣言	接頭部なしの QNames に使用するネーム・スペースを指定	declare default element namespace "http://posample.org";

デフォルトの関数ネーム・スペース宣言	関数呼び出しの接頭部が付かない関数に使用されるネーム・スペース URI を指定	declare default function namespace "http://www.ibm.com/xmlns/prod/db2/function"
空の順序宣言	FLWOR 式の order by 節が処理されるときに空のシーケンスまたは NaN の最大最小としての解釈	declare default order empty greatest;
順序付けモード宣言	照会の順序付けモード	declare ordering unordered;
ネーム・スペース宣言	ネーム・スペース接頭部を宣言	declare namespace ns1 = "http://posample.org";

## 式

- ・照会の基本的なビルディング・ブロック
- ・単独で使用することも、他の式と組み合わせて複合照会を形成することも可能
- ・いくつかの操作は、式の処理によく含まれる。
- ・更新式は変換式の modify 節内でのみ使用できる

### 動的コンテキストおよびフォーカス

フォーカスの構成要素	内容
コンテキスト・アイテム	現在処理中のアトミック値またはノード。ドット "." でアイテムを検索可能
コンテキストの位置	コンテキスト・アイテムの位置。fn:position() 関数によって検索可
コンテキスト・サイズ	アイテム数。fn:last() 関数によって検索可

## 基本式

### リテラル

- ・アトミック値の直接構文表記

リテラル	タイプ	内容
数値	xs:integer、xs:decimal、xs:double	数値、e または E、小数点からなる
ストリング	xs:string	単一引用符 (') または二重引用符 (") に囲まれたアトミック値。単一引用符で区切られたストリング・リテラル内に単一引用符を含めるには、2 つの連続する単一引用符を指定

## 変数参照

- ・変数参照は、ドル記号 (\$) が先頭に付加された NCName

#### 変数を有効範囲内に追加

- ・ホスト言語環境、SQL/XML、XMLQUERY 関数、XMLTABLE 関数、または XML EXISTS 述部によって有効範囲内変数に追加
- ・XQuery 式によって変数を値にバインド
  - ・変数をバインドできる式は、FLWOR 式および量化式
  - ・関数呼び出しも、関数本体を実行する前に、関数の仮パラメーターに値をバインド
  - ・バインドされた式全体を通して有効範囲内

#### 例

- ・FLWOR 式が、変数 \$seq をシーケンス (10, 20, 30) にバインド

```
let $seq := (10, 20, 30)
return $seq[2];
```

戻り値は 20

#### コンテキスト・アイテム式

- ・1 個のピリオド文字 (.) から構成
- ・現在処理中の項目 (コンテキスト・アイテム) に評価

#### 例

- ・範囲式 1 to 100 で戻されるシーケンスのすべての項目に係数演算子を呼び出すコンテキスト・アイテム式

```
(1 to 100)[. mod 5 eq 0]
```

1 から 100 までの数字で、5 で均等に割り切れる整数のシーケンス

#### 関数呼び出し

- ・QName と、その後続く括弧で囲んだ 0 個以上の式のリスト (これを引数と呼びます) で構成
  - ・組み込み XQuery 関数、DB2 固有の関数の呼び出しをサポート

#### 例

##### ストリング引数を使用した関数呼び出し

- ・引数を取り、すべての文字が大文字である xs:string タイプの値を戻

```
fn:upper-case($ns1_customer info/ns1:addr/@country)
```

##### シーケンス引数を使用した関数呼び出し

- ・単一引数として、シーケンス (1, 2, 3) を取ります

fn:max((1, 2, 3))

## パス式

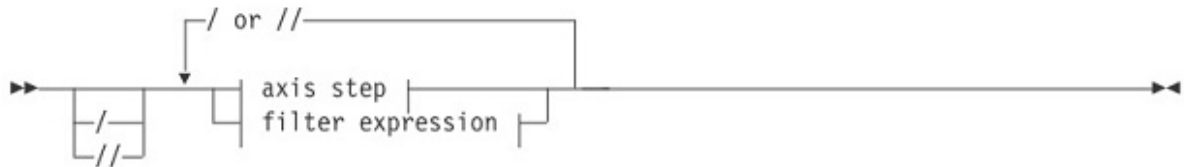
- ・XML ツリー内のノードを識別
- ・XPath 2.0 の構文に基づく
- ・パス式は、スラッシュ (/) またはダブルスラッシュ (//) 文字で区切られた 1 つ以上のステップで構成

## 構文

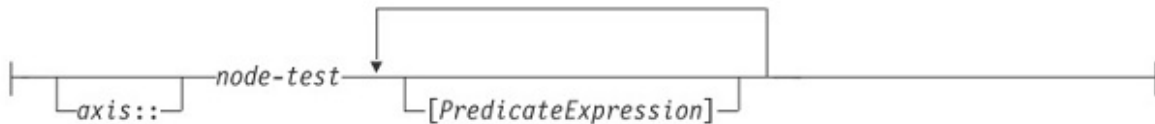
- ・各ステップは、軸ステップまたはフィルター式のいずれか

パス式	内容
軸ステップ	指定された軸によってコンテキスト・ノードから到達できるノードのシーケンスを戻す
フィルター式	基本式と、それに続く 0 個以上の述部で構成

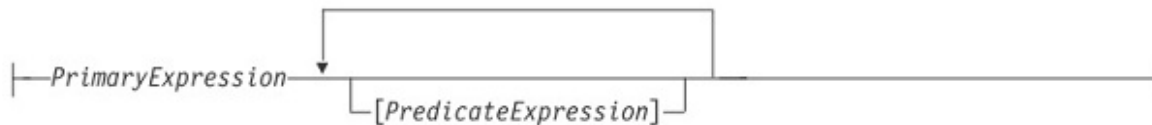
### パス式



### 軸ステップ axis step



### フィルター式 filter expression



記号	説明
/	パスが、コンテキスト・ノードを含むツリーのルート・ノードから開始されることを示す。パス式内のスラッシュ文字は、ステップを分離



//	パスが、コンテキスト・ノードを含むツリーのルート・ノードと、そのルート・ノードのすべての子孫で構成される初期ノード・シーケンスから開始されることを示す。ステップ間のダブルスラッシュ文字は、省略構文参照
axis	<u>XML</u> 文書またはフラグメントの移動方向。child、descendant、attribute、self、descendant-or-self、parentなどをサポート
node-test	軸ステップによって選択される各ノードについて true でなければならない条件
PrimaryExpression	基本式
PredicateExpression	シーケンスの項目が保持されているか、廃棄されているかを判別する式

## 例

### 2つの述部を含む軸ステップ

- ・ secretary 子エレメントおよび assistant 子エレメントの両方を持つ、コンテキスト・ノードの子であるすべての employee を選択

```
child::employee[secretary][assistant]
```

### パス式内のステップとしてフィルター式を使用

- ・ 指定された book 内で複数の footnote を含むすべての chapter または appendix を戻す

```
$book/(chapter | appendix)[fn:count(footnote)> 1]
```

## 軸ステップ

### 構成

- ・ 軸、ノード・テスト、述部の3つの部分から構成
- ・ オプションの軸は、移動の方向を指定
- ・ ノード・テストは、ノードの選択に使用する基準を指定
- ・ 軸ステップの結果は、常にゼロ以上のノードのシーケンス

### フォワード・ステップとリバース・ステップ

- ・ 軸ステップは、フォワード・ステップまたはリバース・ステップに分けられる

ステップ	内容
------	----

フォワード・ステップ	コンテキスト・ノードから開始し、文書の順序で <u>XML</u> ツリーを移動
リバース・ステップ	コンテキスト・ノードから開始し、文書の順序の逆で <u>XML</u> ツリーを移動

例

- ・ child が軸の名前、para がこの軸上で選択されるエレメント・ノードの名前

child::para

この例の軸ステップでは、コンテキスト・ノードの子であるすべての para エレメントを選択

軸

- ・ XML 文書内を移動する方向を指定する軸ステップの一部
- ・ 軸は、フォワード軸またはリバース軸のいずれかに分類

DB2 XQuery でサポートされる軸

軸	方向	説明
child	順	コンテキスト・ノードの子
descendant	順	コンテキスト・ノードの子孫
属性	順方	コンテキスト・ノードの属性
self	順	コンテキスト・ノードのみ
descendant-or-self	順	コンテキスト・ノードおよびコンテキスト・ノードの子孫
parent	逆	コンテキスト・ノードの親

ノード・テスト

- ・ 軸ステップにより選択される各ノードについて true でなければならない条件
- ・ 名前テストまたは種類テスト

名前テスト

- ・ QName またはワイルドカードで構成
- ・ 軸ステップで名前テストが指定されると、ステップは、QName またはワイルドカードに一致する指定された軸上のノードを選択

DB2 XQuery でサポートされる名前テスト

テスト	説明	例	例の説明
QName	QName に等しい(指定された軸上の)任意のノードに一致	child::para	名前テスト para は、子軸上のすべての para エレメントを選択

*	指定された軸上のすべてのノードに一致	child::*	子軸上のすべてのエレメントに一致
NCName:*	QName の接頭部部分を表す NCName を指定	child::ns1:*	接頭部 ns1 にバインドされているネーム・スペースに関連付けられた子軸上のすべてのエレメントに一致
*:NCName	QName のローカル部分を表す NCName を指定	child::*:customerinfo	エレメント名に関連付けられたネーム・スペースには関係なく、ローカル名 customerinfo を持つ、子軸上のすべてのエレメントに一致

## 種類テスト

- ・種類テストに一致する指定された軸上のノードのみを選択

### DB2 XQuery でサポートされる種類テスト

テスト	説明	例	例説明
node()	指定された軸上の任意のノード	child::node()	子軸上の任意のノードを選択
text()	指定された軸上の任意のテキスト・ノードに一致	child::text()	子軸上の任意のテキスト・ノードを選択
comment()	指定された軸上の任意のコメント・ノードに一致	child::comment()	子軸上の任意のコメント・ノードを選択
processing-instruction()	指定された軸上の任意の処理命令ノードに一致	child::processing-instruction()	子軸上の任意の処理命令ノードを選択
element() または element(*)	指定された軸上の任意のエレメント・ノードに一致し	child::element()	子軸上の任意のエレメント・ノードを選択
attribute() または attribute(*)	指定された軸上の任意の属性ノードに一致	child::attribute()	子軸上の任意の属性ノードを選択
document-node()	指定された軸上の任意の文書ノードに一致	self::document-node()	コンテキスト・ノードである文書ノードを選択

## パス式の省略構文

省略構文	説明	省略例	元構文
軸の指定なし	ノード・テストとして軸ステップが attribute() を指定したとき以外は、child:: の省略形。軸ステップが属性テストを指定するときは、省略される軸は、attribute:: の省略形です。	section/para	child::section/child::para
@	attribute:: の省略形	section/@id	child::section/attribute::id
//	パス式の先頭にあるとき以外は、 /descendant-or-self::node()/ の省略形	div1//para の省略形	child::div1 /descendant-or-self::node()/child::p
..	parent::node() の省略形	../title	parent::node()/child::title