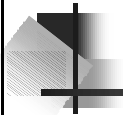




XML Consortium



XML DB活用の最近の動向

日本オラクル株式会社
金井 盛隆



Agenda

XML Consortium

- RDBとXMLの連携
- SQL/XML
- XML DBの分類
- XMLストレージ
- 実装例
- まとめ



Agenda

 RDBとXMLの連携

- SQL/XML
- XML DBの分類
- XMLストレージ
- 実装例
- まとめ



RDB-XML連携の目的

- 旧来のRDBデータとXMLデータを同時に利用・管理する

「連携」の意味するもの

- XMLデータをRDBデータとして参照する機能
 - [Oracle] Extract関数、ExtractValue関数など
- RDBデータをXMLデータとして参照する機能
 - XMLElement関数、XMLForest関数、XMLAgg関数など (SQLX関数)

Agenda

- RDBとXMLの連携
- ⇒ SQL/XML
- XML DBの分類
- XMLストレージ
- 実装例
- まとめ

SQL標準

ANSI (米国規格協会)

American National Standard Institute

ISO (国際標準化機構)

International Organization for Standardization

- SQL92
 - RDB完成形の追求
 - 新たなニーズへの対応
- SQL99
 - オブジェクト・パラダイムへの対応
- SQL2003
 - Java/OLAP/XML連携
 - 外部データ連携

SQL/XML (9075-14)

- XMLの格納・取り出しに必要な仕様
- RDBの利点を活かしつつXMLが持つ利点も利用できる内容
- SQL/XMLで規定された機能
 - XML型
 - XML型の値
 - SQLとXMLの間の対応付け
 - SQLデータからXMLを生成する関数群

SQL/XMLに 適合するための機能項目

- XML型
 - XML型の指定
 - CASTオペランドにXML型を指定できる
 - XML型による式
 - XML型の関数
- XMLElement関数
- XMLForest関数
- XMLAgg関数

Agenda

- RDBとXMLの連携
- SQL/XML
- ⇒ XML DBの分類
- XMLストレージ
- 実装例
- まとめ

XML DBの分類

- ネイティブXMLデータベース
- XML Enabled データベース
- ハイブリッドXML データベース

XML:DB Initiative
<http://www.xmldb.org/>

ネイティブXMLデータベース

- ネイティブXMLデータベースは、以下の2つの条件を満たすものとし、物理的な格納モデルに制限はない
 - XML文書を明示的な変換・マッピング・操作をすることなく、単純にXML文書として格納・取り出しできる
 - XML文書を論理的なモデルとして定義し、格納し検索することができる。要素、属性、PCDATA、ドキュメント順序を最低限サポートしなければならない

XML Enabled データベース

- XML Enabled データベースは、データベースの上位レイヤーにマッピング層が存在しXMLデータとマッピングを行い、XML特有のXPath式などあるいはSQLなどで格納したデータが操作可能なもの

ハイブリッドXMLデータベース

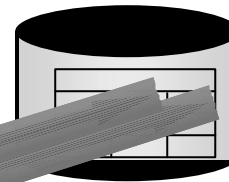
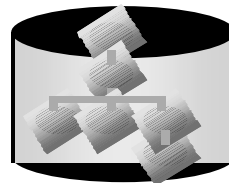
- ハイブリッドXMLデータベースは、ネイティブXMLデータベースとXML Enabled データベースの両方の機能を持つもの

Agenda

- RDBとXMLの連携
- SQL/XML
- XML DBの分類
- ⇒ XMLストレージ
- 実装例
- まとめ

構造化ストレージ vs. 非構造化ストレージ

- 非構造化ストレージ
 - XML文書をそのままの形で表のLOB列などに格納
- 構造化ストレージ (構造化マッピング)
 - XML文書のテキスト値や属性値を表の列にマッピングし格納



格納方法の違い 非構造化 vs. 構造化

非構造化ストレージ

XMLType (CLOB型)

```

<empno>1100</empno>
<ename>SCOTT</ename>
<sal>300000</sal>
</emp>

<emp>
  <empno>1200</empno>
  <ename>SMITH</ename>
  <sal>350000</sal>
</emp>
  
```

構造化ストレージ

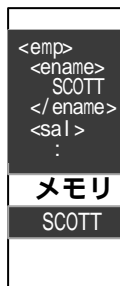
XMLType (オブジェクト型)		
empno (NUMBER)	ename (VARCHAR2)	sal (NUMBER)
1100	SCOTT	300000
1200	SMITH	350000

検索パフォーマンスの違い ~ 非構造化 vs. 構造化

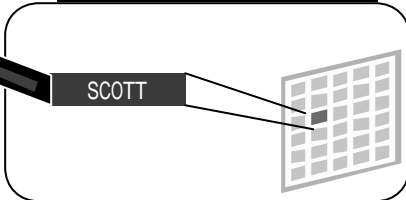
ename要素を
検索
(/emp/ename)

全部を読み込み、
解析(Parse)する

非構造化ストレージ



構造化ストレージ



一部を読み込む

Agenda

- RDBとXMLの連携
- SQL/XML
- XML DBの分類
- XMLストレージ
- 実装例
 - ➡ XML文書の格納と取り出し
 - XML文書の生成
- まとめ

サンプル XML スキーマ (1)

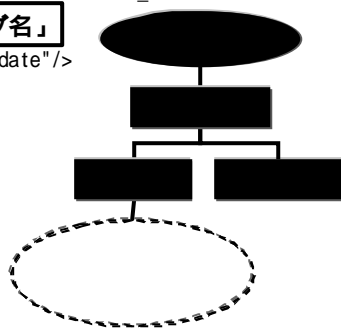
```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ora="http://xmlns.oracle.com/xdb"
  xmlns="http://xmlns.oracle.co.jp/orablog"
  targetNamespace="http://xmlns.oracle.co.jp/orablog"
  elementFormDefault="qualified">
  <xsd:element name="DayEntry" ora:defaultTable="DEFAULT_DAYENTRY">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Date" type="xsd:date"/>
        <xsd:element name="Entries">
          <xsd:complexType>
            <xsd:sequence>
              ...次のページ...
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

対象名前空間

いわゆる「タグ名」



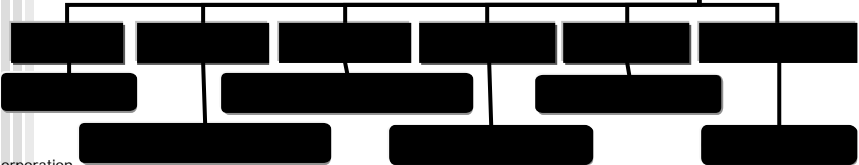
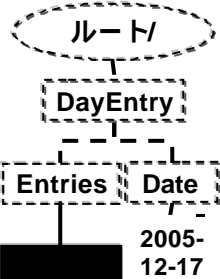
サンプル XML スキーマ (2)

XML Consortium

```

<xsd:element name="Entry" maxOccurs="unbounded"
  ora:SQLInline="false">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="EntryId" type="xsd:string" />
      <xsd:element name="Created" type="xsd:dateTime" />
      <xsd:element name="Modified" type="xsd:dateTime" />
      <xsd:element name="Title" type="xsd:string" />
      <xsd:element name="Content" type="xsd:string"
        ora:SQLType="CLOB" />
      <xsd:element name="Categories" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
  
```

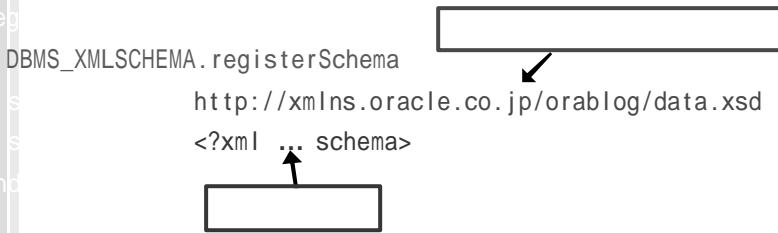
どんな種類のデータが入るか



Oracle XML DB への XML スキーマの登録

XML Consortium

- XML スキーマとXML スキーマを識別する URL を対応づけて登録



XMLスキーマの登録で Oracle XML DB に何が起きるのか

```

<xsd:element name="Entry" ...>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="EntryId"
        type="xsd:string"/>
      <xsd:element name="Created"
        type="xsd:dateTime"/>
      <xsd:element name="Modified"
        type="xsd:dateTime"/>
      <xsd:element name="Title"
        type="xsd:string"/>
      <xsd:element name="Content"
        type="xsd:string"
        ora:SQLType="CLOB"/>
      <xsd:element name="Categories"
        type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
  
```

```

CREATE OR REPLACE TYPE
"SCOTT"."Entry183_T" AS OBJECT (
  "SYS_XDBPD$" "XDB"."XDB$RAW_LIST_T",
  "EntryId" VARCHAR2(4000 CHAR),
  "Created" TIMESTAMP,
  "Modified" TIMESTAMP,
  "Title" VARCHAR2(4000 CHAR),
  "Content" CLOB,
  "Categories" VARCHAR2(4000 CHAR)
) FINAL INSTANTIABLE
  
```

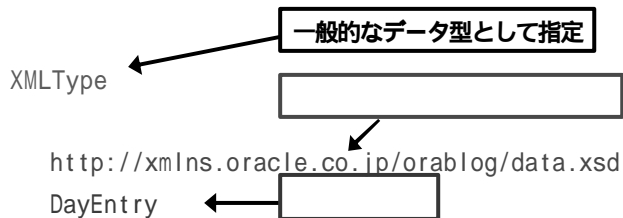
**自動的な XML Schema から
Oracle データ型 (オブジェクト型)
への構造のマッピング**

単体のマッピング

スライドには一部のマッピング
を掲載しています。

XMLType データ型を列に持つ表

- CREATE TABLE 文で作成
- XML スキーマの物理的な位置とルート要素を指定



XML 文書の一部を取得するための関数およびメソッド

- SQL 関数
 - ExistsNode
 - XML 文書を横断して該当するノードの有無を検索
 - Extract
 - ExtractValue
- XMLType データ型のメソッド
 - existsNode : ExistsNode 関数と同様の機能
 - extract :Extract 関数と同様の機能
 - get [Clob | Blob | String | Number] Val
 - 該当するテキストノードをそれぞれのデータ型で取得

Extract 関数

- XML 文書をXPath 式で検索 (ノードセット)
- 引数は XMLType データ型、XPath 式と名前空間
- 名前空間を省略すると 対象となるXMLType データ型が従うXML スキーマの targetNamespace 属性で指定された名前空間が使用される

xmlns="http://xmlns.oracle.co.jp/orablog"

ExtractValue 関数

- XML 文書のスカラー値をXPath 式で検索
- 引数などはExtract 関数と同様
- 構造化マッピングの場合には、対象になるスカラー値のデータ型に合わせた戻り値となる

ORDER BY

xmlns="http://xmlns.oracle.co.jp/orablog"

XSLT プロセッサ

- PL/SQL XSLT Processor API for XMLType
 - DBMS_XSLPROCESSOR パッケージ

XQuery

- Oracle Database 次期バージョンで製品版として対応予定

Agenda

- RDBとXMLの連携
- SQL/XML
- XML DBの分類
- XMLストレージ
- 実装例
 - XML文書の格納と取り出し
- ➡ XML文書の生成
- まとめ

XML生成関数

- 既存のリレーショナルデータから多様なXMLインスタンスを作成
 - 独自のSQL関数
 - PL/SQLパッケージ
 - SQLX関数

SQLX関数概要

XML

- SQLX標準に準拠
- 複雑な構造のXML文書をSQL文で作成

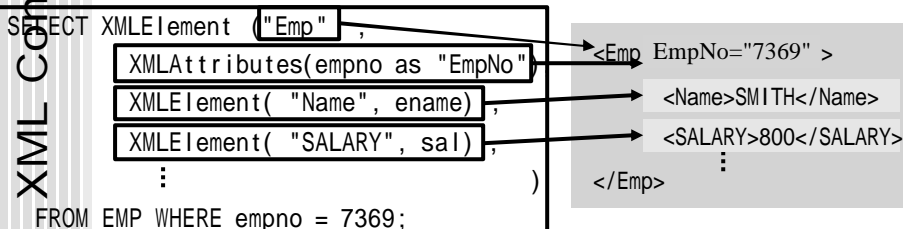
分類	関数	詳細
SQLXで定義されている関数	XMLElement	XML文書の要素を生成する。
	XMLForest	引数に与えられた列ごとにXML文書の要素を生成する。
	XMLConcat	与えられた引数をまとめて一つのXMLフラグメントを生成する。
	XMLAgg	与えられた引数が生成する要素をグループごとにまとめる。
SQLXでの定義をオラクル独自に拡張した関数	XMLColAttVal	引数に与えられた列名を属性名にしたXML文書の要素を生成する。
その他の関数	XMLSequence	REFカーソルやXMLTypeのインスタンスからVARRAY型の配列を生成する。

XMLElement関数

XML Consortium

- XML文書の要素を生成
- 入れ子構造をもったXML要素も生成可能
- XMLAttributes句で属性名を付与

XMLElement(要素名, [XMLAttributes(列名[, as 別名]), [子ノード...])



XMLForest関数

■ XML フラグメントを作成

使用例

```
SQL> SELECT XMLForest(ename as "NAME", empno, sal as "SALARY")
2 AS result FROM emp;
```

RESULT

```
-----
<NAME>SMITH</NAME>
<EMPNO>7369</EMPNO>
<SALARY>800</SALARY>
<NAME>ALLEN</NAME>
<EMPNO>7499</EMPNO>
<SALARY>1600</SALARY>
  ⋮
```

XMLForest関数

XMLForestを使用しない例

```
SELECT
  XMLElement("Emp",
    XMLElement("Name", ename),
    XMLElement("SALARY", sal),
    ...
  )
FROM EMP WHERE empno = 7369;
```

XMLForestを使用した例

```
SELECT
  XMLElement("Emp",
    XMLForest(ename as "Name",
      sal as "SALARY",
      ...
    )
  )
FROM EMP WHERE empno = 7369;
```

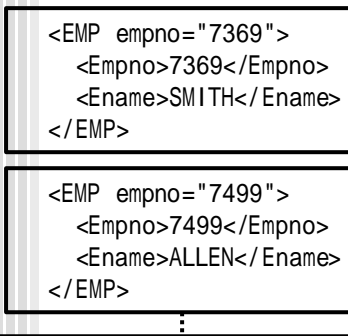
```
<Emp>
  <Name>SMITH</Name>
  <SALARY>800</SALARY>
  ⋮
</Emp>
```

XMLTypeビュー

- XMLType を行オブジェクトを持つビュー
 - 既存の表に格納されたデータをXML文書として表示
- XMLType ビューに対する検索、DML操作が可能
 - XMLType ビューに対するDML操作で、元表を操作可能
- XMLスキーマをサポート
 - XMLスキーマによりデータの妥当性を検証

XMLTypeビュー

ビューのイメージ



EMPNO	ENAME	...
7369	SMITH	...
7499	ALLEN	...
7521	WARD	...
7566	JONES	...
7654	MARTIN	...
7698	BLAKE	...
7782	CLARK	...
7788	SCOTT	...

XMLType型

XMLTypeビューの作成 (XMLスキーマを使用しない場合)



- XML生成関数またはオブジェクト型を使用
- XMLスキーマを使用可能

XMLスキーマを使用しない例

```
CREATE OR REPLACE VIEW emp_view
OF XMLTYPE WITH OBJECT ID
EXTRACT(sys_nc_rowinfo$, '/EMP/@empno').getNumberVal())
AS SELECT
XMLElement("EMP", XMLAttributes(empno as "EmpNo"),
XMLForest(empno AS "Empno", ename AS "Ename"));
FROM emp;
```

オブジェクト
識別子の指定

XMLType
の指定

XMLTypeビューの作成 (XMLスキーマを使用した場合)



XMLスキーマ作成・登録

XMLスキーマを使用した例

```
CREATE OR REPLACE VIEW dept_xml OF XMLTYPE
XMLSCHEMA "http://www.oracle.com/dept.xsd"
ELEMENT "DEPT"
WITH OBJECT ID
EXTRACT(sys_nc_rowinfo$, '/DEPT/DEPTNO').getNumberVal())
AS SELECT XMLElement("dept:DEPT",
XMLAttributes("http://www.oracle.com/DEPT" AS "xmlns:dept",
'http://www.w3.org/2001/XMLSchema-instance' AS "xmlns:xsi",
'http://www.oracle.com/DEPT'
'http://www.oracle.com/dept.xsd' AS "xsi:schemaLocation"),
XMLForest(deptno "DEPTNO", d.dname "DNAME", d.loc "LOC"))
FROM dept d;
```

XMLスキーマ、
ルート要素の
指定

XMLスキーマ、
名前空間
の宣言

XMLTypeビューに対する検索

- XPathを利用した検索が可能
- クエリー・リライトによる高速アクセスを実現

```
SQL> SELECT value(d) FROM dept_xml2 d WHERE
2   ExtractValue(value(d), '/dept:DEPT/LOC',
3     'xmlns:dept="http://www.oracle.com/DEPT"')
4   LIKE '%BOSTON%';
```

VALUE(D)

```
<DEPT xmlns="http://www.oracle.com/DEPT">
  <DEPTNO>40</DEPTNO>
  <DNAME>OPERATIONS</DNAME>
  <LOC>BOSTON</LOC>
</DEPT>
```

XMLTypeビューに対するDML操作

- 元表に対して挿入・更新可能

Insert文の例

```
INSERT INTO dept_xml2 VALUES(
XMLType('<?xml version="1.0"?>
dept:DEPT xmlns:dept="http://www.oracle.com/DEPT"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oracle.com/DEP
http://www.oracle.com/dept2.xsd">
<DEPTNO>90</DEPTNO>
<DNAME>TEST</DNAME>
<LOC>TOKYO</LOC>
dept:DEPT>'));)
```

dept_xml
ビュー

DEPTNO	DNAME	LOC
.	.	.
90	TEST	TOKYO

Agenda

- RDBとXMLの連携
- SQL/XML
- XML DBの分類
- XMLストレージ
- 実装例

⇒ まとめ

XML DBに求められる機能 (1)

- ハイブリッドXMLデータベース方式
 - DB自体がXMLに対応
 - XML専用のデータ型にXML文書を格納
 - 外部のユーティリティを使用する必要がない
- 各種標準への準拠
 - XML
 - XML Schema
 - XPath
 - XSLT
 - XQuery

XML DBに求められる機能 (2)

- 構造化ストレージに対応可能であること
 - XML文書のテキスト値や属性値を表の列にマッピングして格納
- XMLタグと表の列のマッピングを自動的に行う機能を備えていること
 - XML Schema の登録によってマッピング先の表を自動的に生成

XML DBに求められる機能[補足]

- 全文検索機能を有すること
 - XMLタグ内に記述された自然文の部分文字列による検索を高速に行えること
- RDBの標準機能と組み合わせ可能であること
 - トランザクションの概念
 - 高可用性
 - 管理性

参考文献 (1)

- 中山幹敏、奥井康弘 改訂版 『標準XML完全解説』上・下 (技術評論社、2001年)
- 土田正士、小寺孝 『SQL2003ハンドブック』 (ソフト・リサーチ・センター、2004年)
- 『DBマガジン 2004年9月号 特集3 マッピング いらずでXMLデータを格納！ 今日からできる RDB-XML連携術 (中川貴申)』 (翔泳社、2004年)

参考文献 (2)

- OTN (英語)
Oracle Technology Network
<http://otn.oracle.com/tech/xml/xmlldb>
- OTN-J (日本語)
Oracle Technology Network Japan
<http://otn.oracle.co.jp/tech/xml/xmlldb>

ORACLE®

日本オラクル株式会社

無断転載を禁ず

この文書はあくまでも参考資料であり、掲載されている情報は予告なしに変更されることがあります。日本オラクル社は本書の内容に関していかなる保証もいたしません。また、本書の内容に関連したいかなる損害についても責任を負いかねます。

Oracleは米国Oracle Corporationの登録商標です。文中に参照されている各製品名及びサービス名は米国Oracle Corporationの商標または登録商標です。その他の製品名及びサービス名はそれぞれの所有者の商標または登録商標の可能性がありま