

Что дает репозиторий XML DB и как с ним работать

Владимир Пржиялковский

Преподаватель технологий Oracle

prz@yandex.ru,

www.ccas.ru/prz/

*... Зажгу свечу пред каждым сундуком,
И все их отопру, и стану сам
Средь них глядеть на блещущие груды.*

А. С. Пушкин. Скупой рыцарь

Введение

XML DB, созданная в рамках БД Oracle, дает возможность следующего:

- работать с репозитарием;
- регистрировать схему XML для ее использования работе с данными XML в БД;
- создавать внутри СУБД *сервлеты* для доступа к БД через интернет.

Репозиторий XML DB, фактически является древовидно-организованной файловой системой внутри БД, элементами которой выступают *ресурсы*: каталоги (папки) и произвольные (не обязательно текстовые) файлы. Папки можно заводить и удалять, а файлы – заводить, удалять и извлекать.

Технически ресурсы репозитория XML DB суть документы XML. Элемент Contents каждого такого документа представляет содержание ресурса, а все остальные элементы являются метаданными, описывающими ресурс.

Средствами доступа к ресурсам, составляющим репозиторий, могут служить:

- системные таблицы RESOURCE_VIEW и PATH_VIEW,
- программы на PL/SQL (пакет DBMS_XDB) и на Java,
- протоколы HTTP, HTTPS, FTP, WebDAV.

Здесь рассматриваются две первые категории средств, а последняя оставлена для самостоятельных упражнений.

Доступ к ресурсам средствами SQL

Две производные таблицы (из исходных, в схеме XDB) позволяют узнать информацию о ресурсах XML DB: PATH_VIEW и RESOURCE_VIEW:

```
SQL> DESCRIBE resource_view
Name          Null?      Type
-----
RES           SYS.XMLTYPE (XMLSchema
              "http://xmlns.oracle.com/xdb/XDBResource.xsd"
              Element "Resource")
ANY_PATH      VARCHAR2 (4000)
RESID        RAW (16)
```

```
SQL> DESCRIBE path_view
Name          Null?      Type
```

```

-----
PATH          VARCHAR2(1024)
RES           SYS.XMLTYPE(XMLSchema
              "http://xmlns.oracle.com/xdb/XDBResource.xsd"
              Element "Resource")
LINK         SYS.XMLTYPE
RESID        RAW(16)

```

Обе таблицы хранят список ресурсов с путями доступа, однако благодаря возможности определять связи (links), подобно как в файловой системе, путей доступа к одному ресурсу может оказаться несколько; их-то все и покажет таблица PATH_VIEW, в отличие от RESOURCE_VIEW.

Примеры запросов. Список ресурсов в репозитории:

```
SELECT any_path FROM resource_view;
```

Описание первого попавшегося ресурса:

```
SELECT res FROM resource_view WHERE ROWNUM = 1;
```

Для просмотра дерева ресурсов, помимо обычных, существуют специальные функции:

- UNDER_PATH
- EQUALS_PATH
- PATH
- DEPTH

Выдать описания ресурсов, имеющихся в папке */sys/acls*:

```
SELECT res
FROM resource_view
WHERE UNDER_PATH ( res, '/sys/acls' ) = 1
;
```

Выдать относительные имена ресурсов, имеющихся в папке */sys/acls* (в данном случае это будут имена файлов), и их полные имена:

```
SELECT path ( 1 ), any_path
FROM resource_view
WHERE UNDER_PATH ( res, '/sys/acls', 1 ) = 1
;
```

Описание ресурса-папки */sys/acls*:

```
SELECT res
FROM resource_view
WHERE EQUALS_PATH ( res, '/sys/acls' ) = 1
;
```

Полное описание ресурса-файла */sys/acls/all_all_acl.xml*:

```
SELECT r.res.GETCLOBVAL ( )
FROM resource_view r
WHERE EQUALS_PATH ( res, '/sys/acls/all_all_acl.xml' ) = 1
;
```

С запросами последнего типа следует соблюдать осторожность, так как элемент Contents результирующего документа XML хранит для ресурса-файла его содержимое, а оно может оказаться очень объемистым («большой файл»).

Содержание ресурса-файла */sys/acls/all_all_acl.xml* с содержимым в формате XML:

```
SELECT r.res.EXTRACT ( '//Contents' )
FROM resource_view r
WHERE EQUALS_PATH ( res, '/sys/acls/all_all_acl.xml' ) = 1
```

;

Другие типы файлов могут хранить описание содержимого в элементе /Resource/Contents/text или /Resource/Contents/binary.

Работа с ресурсами в программе

Заводить, изменять свойства и удалять ресурсы в репозитории XML DB можно с помощью пакета PL/SQL DBMS_XDB.

Примеры:

```
CONNECT scott/tiger
```

```
DECLARE retb BOOLEAN;
BEGIN
retb := DBMS_XDB.CREATEFOLDER ( '/public/myfolder' );

retb := DBMS_XDB.CREATERESOURCE (
  '/public/myfolder/file1.txt'
, 'First line' || CHR ( 10 ) || 'Second line'
);

retb := DBMS_XDB.CREATERESOURCE (
  '/public/myfolder/file2.xml'
, '<doc><line>First line</line><line>Second line</line></doc>'
);
END;
/
```

Проверка:

```
SQL> SELECT r.res.EXTRACT ( 'Resource/Contents' ) AS xml
       2 FROM resource_view r
       3 WHERE EQUALS_PATH ( res, '/public/myfolder/file2.xml' ) = 1;
```

XML

```
-----
<Contents xmlns="http://xmlns.oracle.com/xdb/XDBResource.xsd">
  <doc xmlns="">
    <line>First line</line>
    <line>Second line</line>
  </doc>
</Contents>
```

```
SQL> SELECT r.res.EXTRACT ( 'Resource/Contents/text/text()' ) AS text
       2 FROM resource_view r
       3 WHERE EQUALS_PATH ( res, '/public/myfolder/file1.txt' ) = 1;
```

TEXT

```
-----
First line
Second line
```

Обратите внимание, что XML DB по-разному хранит файл в зависимости от его расширения (*упражнение*: убедитесь, что XML DB интерпретирует содержимое именно по расширению файла, а не оттого, что внутри). Соответствие расширений файлов типам MIME устанавливается и выясняется в файле-ресурсе /xdbconfig.xml.

Создание связи (link):

```
BEGIN
DBMS_XDB.LINK (
```

```

    '/public/myfolder/file1.txt'
, '/public'
, 'myfolderfile1.txt'
);
END;
/

```

Проверка:

```

SQL> SELECT r.res.EXTRACT ( 'Resource/Contents/text/text()' ) AS text
      2 FROM resource_view r
      3 WHERE EQUALS_PATH ( res, '/public/myfolderfile1.txt' ) = 1
SQL> /

```

TEXT

```

-----
First line
Second line

```

```

SQL> SELECT
      2   p.path AS path
      3   , p.link.extract ('/LINK/ChildName/text()') AS link
      4 FROM path_view p
      5* WHERE UNDER_PATH ( p.res, '/public' ) = 1
SQL> /

```

PATH	LINK
/public/myfolder	myfolder
/public/myfolderfile1.txt	file1.txt
/public/myfolder/file1.txt	file1.txt
/public/myfolder/file2.xml	file2.xml

Обратите внимание на два пути доступа в репозитории к одному и тому же файлу (ресурсу).

Удаление:

```

CALL DBMS_XDB.DELETERESOURCE ( '/public/myfolderfile1.txt' );
CALL DBMS_XDB.DELETERESOURCE ( '/public/myfolder/file1.txt' );
CALL DBMS_XDB.DELETERESOURCE ( '/public/myfolder/file2.xml' );
CALL DBMS_XDB.DELETERESOURCE ( '/public/myfolder' );

```

Упражнение. Проверьте реакцию XML DB на попытку удалить несуществующий файл или непустую папку.

В отличие от производных таблиц (view) словаря-справочника в Oracle, производные таблицы RESOURCE_VIEW и PATH_VIEW обновляемы (на деле это «объектно-реляционные» таблицы). Это позволяет, например, удалить связь также и командой DELETE, или переместить существующий ресурс в другую папку обычной операцией UPDATE:

```

UPDATE path_view
SET путь = НОВЫЙ_ПУТЬ
WHERE equals_path ( res, путь ) = 1
;

```

Пример помещения в репозиторий файла ОС:

```

CONNECT / AS SYSDBA

CREATE DIRECTORY courses AS 'c:\crs';

GRANT READ ON DIRECTORY courses TO scott;

CONNECT scott/tiger

```

```

DECLARE retb BOOLEAN;
BEGIN
retb :=
DBMS_XDB.CREATERESOURCE (
'/public/OracleXML.doc'
, BFILENAME ( 'COURSES', 'OracleXML.doc' )
);
END;
/

```

Проверка:

```

SQL> SELECT res AS resource_description
2 FROM resource_view
3 WHERE EQUALS_PATH ( res, '/public/OracleXML.doc' ) = 1
4 ;

```

RESOURCE_DESCRIPTION

```

-----
<Resource xmlns="http://xmlns.oracle.com/xdb/XDBResource.xsd">
  <CreationDate>2006-09-05T12:47:57.547000</CreationDate>
  <ModificationDate>2006-09-05T12:47:57.547000</ModificationDate>
  <DisplayName>OracleXML.doc</DisplayName>
  <Language>en-US</Language>
  <CharacterSet>WINDOWS-1251</CharacterSet>
  <ContentType>application/msword</ContentType>
  <RefCount>1</RefCount>
</Resource>

```

Технически файлы репозитория размещаются служебных таблицах БД (документы XML) или в объектах LOB (файлы всех остальных типов). Использование формата MIME для хранения двоичных файлов не является самым экономным, что относится к издержкам метода.

Разграничение доступа

Репозиторий XML DB в БД Oracle использует собственную схему защиты доступа, access control list (ACL), созданную в рамках модели ACL для WebDAV (<http://greenbytes.de/tech/webdav/rfc3744.html>). Основными понятиями ACL являются:

Участник безопасности (principal). В XML DB это пользователь БД, роль БД или пользователь/роль справочника каталогов LDAP.

Привилегия. Может быть атомарной (atomic; например read-contents, update или dav:lock) и составной (aggregate, состоящей из других привилегий; например all, dav:all или dav:read-acl). (Полный перечень имеющихся в XML DB привилегий имеется в документации).

Access control entry (ACE). Запись о предоставлении или запрету привилегии участнику. Делается в тексте ACL.

В XML DB имеются несколько встроенных ACL, заданных следующими ресурсами:

```

/sys/acls/all_all_acl.xml
/sys/acls/all_owner_acl.xml
/sys/acls/bootstrap_acl.xml
/sys/acls/ro_all_acl.xml

```

ACL файла */public/OracleXML.doc* в программе можно узнать так:

```

SELECT r.res.EXTRACT ( 'Resource/ACL' ) AS text
FROM resource_view r
WHERE EQUALS_PATH ( res, '/public/OracleXML.doc' ) = 1;

```

Пример замены ACL ресурса:

```

BEGIN DBMS_XDB.SETACL (

```

```

    '/public/OracleXML.doc'
, '/sys/acls/all_all_acl.xml'
);
END;
/

```

Создадим в БД роль и создадим в XML DB соответствующий ей файл ACL:

```

CONNECT / AS SYSDBA

CREATE ROLE mygroup;

CONNECT xdb/xdb

DECLARE
aclxml VARCHAR2 ( 4000 ) :=
'
<acl description="All privileges to MYGROUP, no to others"
  xmlns="http://xmlns.oracle.com/xdb/acl.xsd"
  xmlns:dav="DAV:"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.oracle.com/xdb/acl.xsd
                    http://xmlns.oracle.com/xdb/acl.xsd"
>
  <ace>
    <principal>MYGROUP</principal>
    <grant>true</grant>
    <privilege>
      <all/>
    </privilege>
  </ace>
</acl>
'
;
retb BOOLEAN;

BEGIN
retb :=
DBMS_XDB.CREATERESOURCE ( '/sys/acls/all_mygroup_acl.xml', aclxml )
;
END;
/

```

Защитим файл */public/OracleXML.doc* созданным ACL:

```

CONNECT scott/tiger

BEGIN
DBMS_XDB.SETACL (
  '/public/OracleXML.doc'
, '/sys/acls/all_mygroup_acl.xml'
);
END;
/

COMMIT;

```

Проверим видимость ресурса пользователем SCOTT. Для удобства создадим сначала файл с запросом:

```

SELECT any_path
FROM resource_view
WHERE UNDER_PATH ( res, '/public' ) = 1
.
SAVE publicpaths

```

Проверка:

```
SQL> CONNECT scott/tiger
Connected.
SQL> @publicpaths
```

no rows selected

```
SQL> CONNECT / as sysdba
Connected.
SQL> GRANT mygroup TO scott;
```

Grant succeeded.

```
SQL> CONNECT scott/tiger
Connected.
SQL> @publicpaths
```

ANY_PATH

/public/OracleXML.doc

При включении пользователя в группу ранее невидимый ресурс стал виден.

Тип XDBURITYPE для работы с ресурсами

Для работы с ресурсами репозитория можно использовать системный подтип XDBURITYPE абстрактного типа URITYPE. В частности, методы типа XDBURITYPE позволяют извлекать из репозитория содержимое ресурсов.

Пример определения длины файла */public/OracleXML.doc*:

```
SELECT
  DBMS_LOB.GETLENGTH (
    XDBURITYPE ( '/public/OracleXML.doc' ).GETBLOB ( )
  ) AS bytes
FROM dual
;
```

Результат:

```
      BYTES
-----
      504320
```

Таким же образом можно извлечь большой файл (содержимое, а не длину) в переменную программы.

Одно из применений типа XDBURITYPE – дать возможность ссылаться на данные в репозитории XBM DB из полей обычных таблиц. Пример:

```
CREATE TABLE projects AS
  SELECT
    1 AS pid
    , XDBURITYPE ( '/public/OracleXML.doc' ) AS description
  FROM dual
;

SELECT
  DBMS_LOB.GETLENGTH ( p.description.GETBLOB ( ) ) AS bytes
FROM projects p;
```

Получим:

```
      BYTES
```

Обратите внимание, что сослаться на файл в репозитории из БД можно и через тип HTTPURITYPE, однако в этом случае в ссылке появится имя компьютера и номер порта – признаки, внешние по отношению к содержимому БД, неконтролируемые средствами БД и, в отличие от ссылок извне, по сути ненужные.

Другие возможности

Любой ресурс репозитория можно перевести в режим версионного доступа (version control resource, VCR). С этой целью, и с целью самого доступа, следует использовать особый пакет DBMS_XDB_VERSION.

Любой ресурс репозитория можно снабдить собственным описанием (метаданными; в дополнение к «системным» метаданным), добавив в его описание XML («системное») свои элементы. Для этой цели можно использовать разные средства:

- в PL/SQL – процедуры пакета DBMS_XDB: **APPENDRESOURCEMETADATA**, **UPDATERESOURCEMETADATA**, **DELETERESOURCEMETADATA**, **PURGERESOURCEMETADATA**;
- в SQL – операции INSERT, UPDATE, DELETE применительно к полю RES (производной) таблицы RESOURCE_VIEW;
- методом PROPPATCH протокола WebDAV.

Подробности имеются в документации по Oracle.