

## TRABAJO DE INVESTIGACION – BASES DE DATOS RELACIONALES

Juan Daniel Forner Garriga

1 DAW Mañanas

### 1. Crear tablas, registros, campos

Para crear campos que contengan información almacenada en XML, Oracle nos proporciona el tipo XMLType que asignaremos al campo en que necesitemos almacenar información XML.

### 2. Procesar el XML

Para seleccionar un elemento específico dentro de un campo XMLType se puede utilizar la función `extractvalue()` que tiene la siguiente sintaxis: `EXTRACTVALUE(campoXMLType, XPath_String)`

Para seleccionar varios elementos dentro del XML, se puede utilizar la función `XMLTable` junto con la sintaxis `XPath` para crear una tabla virtual que contiene los elementos seleccionados. Un ejemplo sería:

```
SELECT x.* FROM mi_tabla_xml t, XMLTable('/biblioteca/libro' PASSING t.mi_campo_xml
COLUMNS titulo VARCHAR2(100) PATH 'titulo', autor VARCHAR2(100) PATH 'autor') x;
```

Como podemos observar la consulta se hace a las tablas "mi\_tabla\_xml" (que llamaremos "t") que es la que contiene el campo XMLType, y la que generamos con la función `XMLTABLE` con la información que necesitemos del XML. En este caso a partir del espacio de nombres "/biblioteca/libro" del campo "t.mi\_campo\_xml" creamos las columnas `titulo` y `autor` de tipo `VARCHAR(100)` a partir de los campos `titulo` y `autor` respectivamente de cada "/biblioteca/libro". A esta tabla generada se le refiere como `x` y como podemos ver en la cláusula `SELECT` se seleccionan todos los campos de esta tabla `x`.

### 3. Insertar, borrar, modificar datos en las tablas mediante XML

Para insertar el XML correspondiente dentro de ese campo deberemos de convertir una cadena que contenga el XML a XMLType usando la función `XMLTYPE()`.

Para eliminar datos de una tabla XML de Oracle, se puede utilizar la cláusula `DELETE` y la función `extractvalue()` o `deleteXML()` para seleccionar y eliminar elementos y atributos específicos.

Para modificar datos en una tabla XML de Oracle, se puede utilizar la cláusula `UPDATE` y la función `updateXML()` para actualizar elementos y atributos específicos.

## 1. Crear tablas, registros o campos

El tipo de dato destinado a la integración de XML en PostgreSQL se llama XML. Puedes crear una tabla que contenga un campo de tipo XML para almacenar un documento XML y poder ejecutar consultas sobre el mismo, utilizando algunas funciones integradas que tiene PostgreSQL.

## 2. Procesar XML

Para realizar una consulta al documento XML almacenado en la base de datos podemos utilizar la función `XPATH()` con la siguiente sintaxis: `XPATH(xpath_string, campo_xml)`. Todo esto dentro de la cláusula `SELECT` de nuestra consulta SQL.

También existe la función `XPATH_EXISTS()`, la cual devuelve un booleano dependiendo de si la expresión `XPATH` ha sido satisfecha o no.

También podemos utilizar la función `XMLTABLE()` de forma similar a Oracle para crear una tabla virtual a partir del documento XML a la cual poder hacer consultas.

## 3. Predicados y funciones XML

PostgreSQL nos proporciona predicados para poder evaluar las propiedades de documentos XML como los predicados `IS DOCUMENT` y `IS NOT DOCUMENT`.

De la misma manera proporciona la función `XMLEXIST()`, con la siguiente sintaxis: `XMLEXISTS ( text PASSING [BY {REF|VALUE}] xml [BY {REF|VALUE}] )`. `XMLEXIST` evalúa una expresión `XPATH` sobre el XML pasado como contexto. La función devuelve falso si el resultado de la evaluación está vacío, true si es cualquier otro valor.

También encontramos las funciones `XML_IS_WELL_FORMED( text )`, `XML_IS_WELL_FORMED_DOCUMENT( text )`, `XML_IS_WELL_FORMED_CONTENT( text )`. Estas funciones evalúan si el documento XML está bien formado o no.

## 4. Insertar, borrar, modificar datos mediante XML

En la documentación de PostgreSQL no se proporciona ningún tipo de información sobre cómo manipular campos con contenido XML.

## 1. Crear tablas, registros o campos

Microsoft SQL Server nos proporciona un tipo de dato llamado *XML* para manejar documentos XML en nuestra base de datos. Dicho tipo puede ser asignado a las columnas necesarias para almacenar el mismo. Se pueden utilizar las funciones y métodos proporcionados por Microsoft SQL Server para manejar XML siempre y cuando sea del tipo dicho.

## 2. Procesar XML

Uno de los métodos que podemos utilizar para procesar el XML que tenemos almacenado en nuestra base de datos es *VALUE()*, con la siguiente sintaxis: *campo.VALUE(XQuery, SQLType)* la cual realiza la consulta XQuery sobre el campo y retorna un valor de tipo SQLType.

Otro método disponible es el de *NODES()* con la sintaxis: *campo.NODES(XQuery) as TABLE(Column)*. Este método hace una consulta xQuery sobre el campo XML y devuelve un conjunto de registros que se almacenan en la tabla *TABLE* con la columna de nombre *Column*. Sobre un resultado de esta consulta se pueden aplicar los demás métodos que actúan frente a XML.

## 3. Insertar, modificar o borrar datos en XML

Existe un método llamado *MODIFY()* el cual nos permite insertar, modificar o borrar datos en las tablas XML que tenemos almacenadas en la base de datos.

Este método nos permite utilizar DML (Data Manipulation Language), el cual es un subconjunto de SQL que se utiliza para gestionar y manipular datos en una base de datos relacional.

La sintaxis del metodo es la siguiente:

```
campo.MODIFY(XML_DML)
```

En XML\_DML se pueden utilizar las declaraciones *insert*, *delete* o *replace* que corresponden al lenguaje DML para modificar las tablas XML.

## 1. Crear tablas, registros o campos

Para crear una columna XML en una tabla de IBM Db2, es necesario especificar que el tipo de dato es XML. De esta forma, se puede almacenar un documento XML completo o un fragmento XML en la base de datos.

```
CREATE TABLE tabla_ejemplo (  
  id INTEGER,  
  xml_data XML  
)
```

## 2. Procesar XML

IBM Db2 ofrece diferentes funciones para procesar XML, como XMLPARSE para analizar una cadena XML y convertirla en un valor XML que se pueda almacenar en la base de datos. También se pueden utilizar funciones como XPATH o XQUERY para recuperar datos específicos de un documento XML almacenado en la base de datos.

Por ejemplo, la siguiente consulta utiliza la función XPATH para recuperar el valor del nodo "name" de un documento XML almacenado en una columna de una tabla de IBM Db2:

```
SELECT xml_data.XPATH('/person/name/text()')  
FROM tabla_ejemplo
```

## 3. Insertar, modificar o borrar datos en XML

IBM Db2 también ofrece métodos para insertar, modificar o borrar datos en documentos XML almacenados en la base de datos. Estos métodos se llaman XMLMODIFY y XMLQUERY, y se utilizan de la siguiente manera:

Para insertar un nuevo nodo en un documento XML almacenado en una columna de IBM Db2, se puede utilizar la función XMLMODIFY de la siguiente manera:

```
UPDATE tabla_ejemplo  
SET xml_data = xml_data.XMLMODIFY('insert node <age>30</age> as last into /person')
```

Para borrar un nodo de un documento XML, se puede utilizar la función XMLMODIFY de la siguiente manera:

```
UPDATE tabla_ejemplo  
SET xml_data = xml_data.XMLMODIFY('delete /person/age')
```

Sistema Gestor de Bases de Datos	Características de Integración de XML	Ventajas de Integración de XML	Desventajas de Integración de XML
Oracle	<ul style="list-style-type: none"> <li>- Soporte nativo para almacenamiento y procesamiento de documentos XML - XML DB para gestión de documentos XML - Soporte para XPath, XQuery, y XSLT - Integración con Java y .NET - Soporte para XML Schema y validación de documentos XML</li> </ul>	<ul style="list-style-type: none"> <li>- Capacidad de almacenar y consultar datos XML - Facilidad de integración con aplicaciones Java y .NET - Soporte para estándares XML - Potente motor de búsqueda y análisis de documentos XML</li> </ul>	<ul style="list-style-type: none"> <li>- Requiere conocimientos especializados para la configuración y administración - Mayor complejidad en la gestión de datos relacionales y XML en la misma base de datos - Limitaciones en el soporte de algunos estándares XML</li> </ul>
PostgreSQL	<ul style="list-style-type: none"> <li>- Soporte nativo para almacenamiento y procesamiento de documentos XML - Soporte para XPath y XQuery - Integración con lenguajes de programación como Python y Java - Soporte para XML Schema y validación de documentos XML</li> </ul>	<ul style="list-style-type: none"> <li>- Capacidad de almacenar y consultar datos XML - Facilidad de integración con aplicaciones Python y Java - Soporte para estándares XML</li> </ul>	<ul style="list-style-type: none"> <li>- Limitaciones en el soporte de algunos estándares XML - Requiere conocimientos especializados para la configuración y administración - Menor rendimiento en la gestión de datos XML en comparación con datos relacionales</li> </ul>
Microsoft SQL Server	<ul style="list-style-type: none"> <li>- Soporte nativo para almacenamiento y procesamiento de documentos XML - XML Data Type para gestión de documentos XML - Soporte para XPath, XQuery y XSLT - Integración con .NET y Visual Studio - Soporte para XML Schema y validación de documentos XML</li> </ul>	<ul style="list-style-type: none"> <li>- Capacidad de almacenar y consultar datos XML - Facilidad de integración con aplicaciones .NET - Soporte para estándares XML</li> </ul>	<ul style="list-style-type: none"> <li>- Requiere conocimientos especializados para la configuración y administración - Limitaciones en el soporte de algunos estándares XML - Mayor complejidad en la gestión de datos relacionales y XML en la misma base de datos</li> </ul>
IBM Db2	<ul style="list-style-type: none"> <li>- Soporte nativo para almacenamiento y procesamiento de documentos XML - Soporte para XPath y XQuery - Integración con Java y .NET - Soporte para XML Schema y validación de documentos XML</li> </ul>	<ul style="list-style-type: none"> <li>- Capacidad de almacenar y consultar datos XML - Facilidad de integración con aplicaciones Java y .NET - Soporte para estándares XML</li> </ul>	<ul style="list-style-type: none"> <li>- Requiere conocimientos especializados para la configuración y administración - Limitaciones en el soporte de algunos estándares XML - Mayor complejidad en la gestión de datos relacionales y XML en la misma base de datos</li> </ul>