



Linkia FP

Formación Profesional Oficial a Distancia



DAM – M06 – Clase 10

Acceso a datos

BBDD nativas en XML

Contenido del módulo

- UF3: Bases de datos nativas en XML.
 - eXist.

Tema 5
3 Clases
1 Actividad

Plan de trabajo

Plan de trabajo - Curso 2021/2022

DAM - M06: Acceso a datos

UF	FECHAS	TEMAS	Recursos complementarios: Video clases	EVALUACIONES				EXAMEN	
				ACTIVIDADES DE EVALUACIÓN CONTÍNUA					
				Entregas o participaciones 70%	Clases 10%*	Test 20%			
UF1: Persistencia en ficheros	02/02/2022 - 14/02/2022	Tema 1. Persistencia en ficheros	VC01	Actividad 1	C01 C02	Test UF1	Examen UF1		
	14/02/2022 - 28/02/2022	Tema 2. Persistencia en BDR con JDBC	VC02		C03 C04	Test UF2	Examen UF2		
	28/02/2022 - 14/03/2022	Tema 3. Persistencia BDR con ORM	VC03	Actividad 2	C05 C06				
	14/03/2022 - 28/03/2022	Tema 4. Persistencia en BDOO – BDOR	VC04	Actividad 3	C07 C08				
UF3: Persistencia en BD Nativas XML	28/03/2022 - 19/04/2022	Tema 5. Persistencia en BBDD nativas XML	VC05	Actividad 4	C09 C10 C11	Test UF3	Examen UF3		
UF4: Componentes de acceso a datos	19/04/2022 - 02/05/2022	Tema 6. Componentes da acceso a datos	VC06	Actividad 5	C12 C13	Test UF4	Examen UF4		
	02/05/2022 - 16/05/2022				C14				

Horario de las clases

NÚMERO CLASE	DÍA DE LA SEMANA	FECHA	HORA INICIO
C01	MARTES	08/02/2022	15:15
C02	VIERNES	11/02/2022	14:30
C03	VIERNES	18/02/2022	14:30
C04	VIERNES	25/02/2022	14:30
C05	VIERNES	04/03/2022	14:30
C06	VIERNES	11/03/2022	14:30
C07	VIERNES	18/03/2022	14:30
C08	VIERNES	25/03/2022	14:30
C09	VIERNES	01/04/2022	14:30
C10	MARTES	05/04/2022	15:15
C11	VIERNES	08/04/2022	14:30
C12	VIERNES	22/04/2022	14:30
C13	VIERNES	29/04/2022	14:30
C14	VIERNES	06/05/2022	14:30

Contenido clase

UF3: Bases de datos nativas en XML.

- Creación y borrado de colecciones: clases y métodos.
- Modificación de contenidos con XQuery.
- Modificación de contenidos con XUpdate.



Creación de colecciones con XML:DB

- Los métodos necesarios para crear colecciones son:
 - La clase *Collection* y su método *getService*.
 - La clase *CollectionManagementService* y su método *createCollection*.

Creación de colecciones con XML:DB

```
public Collection anadirColeccion(Collection contexto,
    String newColec) throws ExcepcionGestorBD {
Collection newCollection = null;
try {
    //Se crea un nuevo servicio desde el contexto
    CollectionManagementService mgtService
        = (CollectionManagementService) contexto.getService(
            "CollectionManagementService", "1.0");
    //Se crea una nueva colección con el nombre newColec codificado UTF8.
    //La colección nueva se devuelve en newCollection(Collection)
    newCollection = mgtService.createCollection(
        new String(UTF8.encode(newColec)));
} catch (XMLDBException e) {
    throw new ExcepcionGestorBD("Error añadiendo colección: " + e.getMessage());
}
return newCollection;
}
```

Borrando colecciones con XML:DB

- Vuelve a ser necesario partir de una colección que hará de padre y sobre la que crearemos un nuevo *CollectionManagementService*.
- Esta *CollectionManagementService* es la clase que tiene el método *removeCollection(String name)* que elimina la colección de nombre name.

CLASE

Borrando colecciones con XML:DB

```
public void borrarColeccion(Collection contexto,  
    String antColecc) throws ExcepcionGestorBD {  
    try {  
        CollectionManagementService mgtService  
            = (CollectionManagementService) contexto.getService(  
                "CollectionManagementService", "1.0");  
        mgtService.removeCollection(antColecc);  
    } catch (XMLDBException e) {  
        throw new ExcepcionGestorBD(  
            "Error eliminando colección: " + e.getMessage());  
    }  
}
```

CLASE

Modificación de contenidos XML

- Es necesario poder consultar y actualizar los datos.
- Para realizar actualizaciones existen alternativas:
 - El uso de DOM o SAX.
 - XUpdate.
 - XQuery y su extensión XQueryUpdateFacility3.0.
 - Lenguajes declarativos propietarios que ofertan los diferentes sistemas XML nativos.

CLASE

Uso de DOM o SAX

- Posibilidad más básica y menos recomendable para su uso generalizado.
- Recuperamos el recurso, lo modificamos y los volvemos a guardar en la colección original.
- Solución costosa, que puede implicar trabajar con archivos grandes o distribuidos entre varias colecciones.
- Solución poco flexible, tenemos que generar una aplicación propia para cada consulta.

XUpdate

- Lenguaje declarativo para la actualización de documentos XML.
- Sintaxis similar a XML y compatible con estándares W3C.
- Es soportado por XML:DB.
- No todas las implementaciones de XML:DB incluyen XUpdate.
 - eXist sí lo incluye.

XQuery Update Facility

- La principal carencia de XQuery es la falta de definición de sentencias para la actualización de datos.
 - El diseño de XQuery se ha realizado teniendo en cuenta ese necesidad.
- XQuery Update Facility es un borrador de la W3C que oferta una extensión de XQuery para cubrirla.
- Al ser una recomendación y no un estándar, no todas las implementaciones de XML:DB o XQJ la soporta.
 - eXist NO lo contempla.

XQuery Update Extension

- Otra opción es utilizar un lenguaje declarativo-propietario que ofertan los diferentes sistemas XML nativos.
- Mientras no haya un estándar para la modificación de contenidos XML, la compatibilidad entre aplicaciones clientes XML será complicada.
 - eXist ofrece XQuery Update Extension, que es una extensión para incluir la modificación con sintaxis similar a XQuery.

Introducción a XQuery

Update Extension

- Introducir datos:

update insert origen (into|following|preceding) destino

- Ejemplos:

*update insert <nacionalidad> Rusa </nacionalidad> into
/Libros/Libro[Autor='Nicolai Gogol']*

*update insert <nacionalidad> Española </nacionalidad>
preceding /Libros/Libro[Titulo="El Sanador de Caballos"]/Autor*

Introducción a XQuery Update Extension

- Modificar datos:

update replace / value destino with nuevo_valor

- Ejemplos:

update replace /Libros/Libro[Titulo="El Sanador de Caballos"]/Autor with <Autor>Gonzalito S. Giner </Autor>

update value /Libros/Libro[Titulo="El Sanador de Caballos"]/Autor with "Gonzalito S. Giner"

Introducción a XQuery

Update Extension

- Borrar datos:

update delete expresion

- Ejemplo:

update delete /Libros/Libro[Titulo='El Resplandor']/Traductor

CLASE

Introducción a XQuery

Update Extension

- Renombrar elementos y atributos:

update rename destino as nuevo_nombre

- Ejemplo:

update rename /Libros/Libro/Autor as "Escritor"

CLASE

Modificación de datos con XML:DB y XQuery Update Extension

- Las clases que intervienen a la hora de hacer una consulta de modificación con XQuery Update Extension son *Collection* y *XQueryService*:
 - De la clase *Collection* es necesario el método *getService*

Modificación de datos con XML:DB y XQuery Update Extension

- *XQueryService* es una extensión de Service que proporciona mecanismos para consultar colecciones con XQuery.
 - Los métodos más destacados son compile y execute.

Modificación de datos con XML:DB y XQuery Update Extension

```
public void ejecutarQuery(String consulta, String contexto) throws ExpcionGestorBD{  
    ResourceSet resultado=null;  
    Collection col;  
    try {  
        //Si el context está vacío quiere decir que no se ha establecido la ruta de la colección que se  
        //tomará como base para hacer la consulta.  
        If (contexto==null){  
            col = DatabaseManager.getCollection(URI +DBBroker.ROOT_COLLECTION);  
            System.out.print(URI + DBBroker.ROOT_COLLECTION);  
        } else {  
            // Si el contexto está definido entonces se obtiene la colección que especifica  
            col = DatabaseManager.getCollection(URI +contexto);  
        }  
        //cont ...  
    }
```

Modificación de datos con XML:DB y XQuery Update Extension

```
//cont..  
//Se crea un XQueryService con la colección obtenida.  
XQueryService service = (XQueryService)col.getService( "XQueryService", "1.0" );  
//Se asignan propiedades del document: indentación y codificación  
service.setProperty( OutputKeys.INDENT, "yes" );  
service.setProperty( OutputKeys.ENCODING, "UTF-8" );  
//Se compila la consulta  
CompiledExpression compiled = service.compile( consulta );  
//La consulta compilada se ejecuta. Cuando la consulta es de tipo XQuery Update Extension no  
//devuelve ningún valor por lo que resultado estará vacío.  
resultado = service.execute(compiled);  
}  
catch (XMLDBException e) {  
    throw new ExpcionGestorBD("Error ejecutando query: " + e.getMessage());  
}  
}
```

Introducción a XUpdate

- XUpdate es actualmente la alternativa más estándar para modificar contenidos de documentos XML almacenados en sistemas XML nativos.
- XUpdate utiliza la propia sintaxis para expresar modificaciones en documentos XML. Una modificación se representa como un modelo del tipo `<xupdate:modifications>`

CLASE

Introducción a XUpdate

- Dentro de <xupdate:modifications>, para efectuar las distintas operaciones de actualización, podemos incluir las siguientes sentencias:
 - xupdate:insert-before
 - xupdate:insert-after
 - xupdate:append
 - xupdate:update
 - xupdate:remove
 - xupdate:rename

CLASE

Introducción a XUpdate

- Insertar datos:

```
<xupdate:modifications version='1.0'  
xmlns:xupdate="http://www.xmldb.org/xupdate">  
    <xupdate:insert-after select="/Libros/Libro/Autor[.=‘Nikolai Gogol’]">  
        <xupdate:element name='nacionalidad’>Rusa</xupdate:element>  
    </xupdate:insert-after>  
</xupdate:modifications>
```

Introducción a XUpdate

- Modificar datos:

```
<xupdate:modifications version='1.0'  
xmlns:xupdate="http://www.xmldb.org/xupdate">  
    <xupdate:update select="/Libros/Libro[Titulo='El  
Resplandor']/Autor">Stephen King</xupdate:update>  
</xupdate:modifications>
```

Introducción a XUpdate

- Borrar datos:

```
<xupdate:modifications version='1.0'  
xmlns:xupdate="http://www.xmldb.org/xupdate">  
    <xupdate:remove select='Libros/Libro[2]' />  
</xupdate:modifications>
```

Introducción a XUpdate

- Renombrar elementos y atributos:

```
<xupdate:modifications version='1.0'  
xmlns:xupdate="http://www.xmldb.org/xupdate">  
    <xupdate:rename select="/Libros/Libro/Autor">Escritor</xupdate:rename>  
</xupdate:modifications>
```

CLASE

Modificación de datos con XML:DB y XUpdate

- Las operaciones expresadas con *XUpdate* se ejecutan en eXist de una forma muy similar a cómo las hacemos con las consultas *XQuery*.
- Las clases que intervienen a la hora de hacer una consulta son *Collection* y *XUpdateQueryService*.
- XUpdateService es una clase de tipo Service. Los métodos más destacados de esta clase son:
 - setCollection(Collection col)
 - update(String expression)

CLASE

Actividad 04

Base de datos con XND.

- Fecha de entrega: 19/04/2022
- Proyecto creado en NetBeans
comprimido en .zip

CLASE



Linkia FP

Formación Profesional Oficial a Distancia