

## 1. Utilización de XML para almacenamiento de información.

Gran parte de las bases de datos están basadas en un modelo de datos entidad-relación que es un sistema que sigue funcionando.

Las BD orientadas a objetos permiten simplificar el trabajo de integración con los lenguajes de programación orientados a objetos. Existen modelos híbridos con extensiones hacia el modelo relacional.

Como solución para compatibilizar ambos modelos se pensó en XML.

XML permite definir de manera rápida e intuitiva una representación de la información.

En el caso de emplear desde el principio XML se deben utilizar BD XML Nativas.

Las **Bases de Datos Nativas XML** (NXD -Native XML Database) definen el modelo lógico de un documento XML, y almacena y recupera los documentos según ese modelo.

Estas bases de datos tienen como **unidad fundamental de almacenamiento lógico el documento XML**, tal como una base de datos relacional tiene una fila en una tabla como su unidad fundamental de almacenamiento lógico.

Bases de datos nativas XML:

- eXcelon XIL Lite (comercial).
- TEXTML (comercial).
- dbXML (*OpenSource*).
- eXist (*OpenSource*).

Cuando se habla de BD XML nativas hay que distinguir dos maneras de almacenar la información:

- Modelo **centrado en almacenamiento de datos**.

Similar a las BD relacionales (usando tuplas)

- Modelo **centrado en el documento**.

No hay campos ni datos. Se guardan documentos XML

- **El modelo relacional** es “Well Organized”. Más estricto. Totalmente estructurado.  
El esquema relacional (tablas y restricciones de integridad) están predefinidas.
- **Los datos en XML** son semiestructurados (más flexibles).  
Los esquemas no tienen porque ser fijos.  
Ideal para integrar aplicaciones e intercambio de datos.

## 2. Bases de datos relacionales

La adaptación del modelo BD Relacional a XML es sencillo.

Será necesario crear una DTD y un documento XML bien formado.

Cuando se usan modelos centrados en el almacenamiento de los datos la referencia son las BD relacionales.

## 2. Bases de datos relacionales

### PRIMERA VERSIÓN BÁSICA (NO ÓPTIMA y SIN ID)

TABLA LIBROS

Cód_Libro	Titulo	Autor	Editorial	Edición	ISBN	NumPáginas
1	Don Quijote de la Mancha	Miguel de Cervantes Saavedra	Juan de la Cuesta	3	9788466745840	176
2	La Celestina	Fernando de Rojas	Maxtor	1	9788471664938	320
3	Leyendas	Gustavo Adolfo Bécquer	Cátedra	21	9788437620244	416

Usar una única tabla no es operativa ni cumple los condicionantes de una base de datos relacional.

## Modelo de una tabla XML (con DTD)

### PRIMERA VERSIÓN BÁSICA (NO ÓPTIMA y SIN ID)

1

<!ELEMENT Libros (libro)\*>

Tendremos una tabla llamada Libros que almacenará tuplas de tipo libro (cero o más libros).

## Modelo de una tabla a XML (con DTD)

2

Hay que indicar que campos componen la tupla.

```
<!ELEMENT libro (Cod_Libro, Titulo, Editorial, Edicion, ISBN,  
NumPaginas, Autores) >
```



## Modelo de una tabla a XML (con DTD)

3

Cada columna de la tabla deberá establecerse como un tipo de dato almacenable (char, integer, etc → #PCDATA) si es simple.

```
<!ELEMENT Cod_Libro (#PCDATA)>
```

```
<!ELEMENT Titulo (#PCDATA)>
```

```
<!ELEMENT Editorial (#PCDATA)>
```

```
<!ELEMENT Edicion (#PCDATA)>
```

```
<!ELEMENT ISBN (#PCDATA)>
```

```
<!ELEMENT NumPaginas (#PCDATA)>
```

## Modelo de una tabla a XML (con DTD)

4

Si existe una columna compleja (relacionada con otra tabla) debemos crear un nuevo elemento similar al del paso2, p.e. que un libro pueda tener uno o más autores con sus campos correspondientes. Es decir los campos relacionados.

```
<!ELEMENT Autores (autor)+>
```

```
  <!ELEMENT autor (Cod_Autor, Nombre, Apellidos, FechaNacimiento)>
```

```
    <!ELEMENT Cod_Autor (#PCDATA)>
```

```
    <!ELEMENT Nombre (#PCDATA)>
```

```
    <!ELEMENT Apellidos (#PCDATA)>
```

```
    <!ELEMENT FechaNacimiento(#PCDATA)>
```

## Modelo de una tabla a XML (con DTD)

```
<!ELEMENT Libros (libro)*>
```

```
<!ELEMENT libro (Cod_Libro, Titulo, Editorial, Edicion, ISBN, NumPaginas, Autores) >
```

```
<!ELEMENT Cod_Libro (#PCDATA)>
```

```
<!ELEMENT Titulo (#PCDATA)>
```

```
<!ELEMENT Editorial (#PCDATA)>
```

```
<!ELEMENT Edicion (#PCDATA)>
```

```
<!ELEMENT ISBN (#PCDATA)>
```

```
<!ELEMENT NumPaginas (#PCDATA)>
```

```
<!ELEMENT Autores (autor)+>
```

```
<!ELEMENT (autor Cod_Autor, Nombre, Apellidos, FechaNacimiento)>
```

```
<!ELEMENT Cod_Autor (#PCDATA)>
```

```
<!ELEMENT Nombre (#PCDATA)>
```

```
<!ELEMENT Apellidos (#PCDATA)>
```

```
<!ELEMENT FechaNacimiento (#PCDATA)>
```

01ejemplo.xml

01ejemplo.dtd

## Bases de datos relacionales

### VERSIÓN ÓPTIMA. USO DE ID y IDREF

TABLA LIBROS

Cód_Libro	Título	Cód_Autor	Editorial	Edición	ISBN	NumPáginas
1	Don Quijote de la Mancha	1	Juan de la Cuesta	3	9788466745840	176
2	La Celestina	2	Maxtor	1	9788471664938	320
3	Leyendas	3	Cátedra	21	9788437620244	416

Relación



TABLA AUTORES

Cód_Autor	Nombre	Apellidos	Fecha Nacimiento
1	Miguel	de Cervantes Saavedra	29/09/1547
2	Fernando	de Rojas	01/01/1470
3	Gustavo	Adolfo Bécquer	17/02/1836

Usamos dos tablas para vincular y evitar repeticiones de información.

# Modelo relacional a XML (con DTD)

## VERSIÓN ÓPTIMA. USO DE ID y IDREF

La adaptación del modelo BD Relacional a XML es sencillo.

01OPTejemplo.xml  
01OPTejemplo.dtd

**Atributo ID único para cada libro**

```
<libro Cod_Libro="L2">  
  <Titulo>La Celestina</Titulo>  
  <Editorial>Maxtor</Editorial>  
  <Edicion>1</Edicion>  
  <ISBN>9788471664938</ISBN>  
  <NumPaginas>320</NumPaginas>  
  <autoria Cod_Autor="A2"/>  
</libro>
```

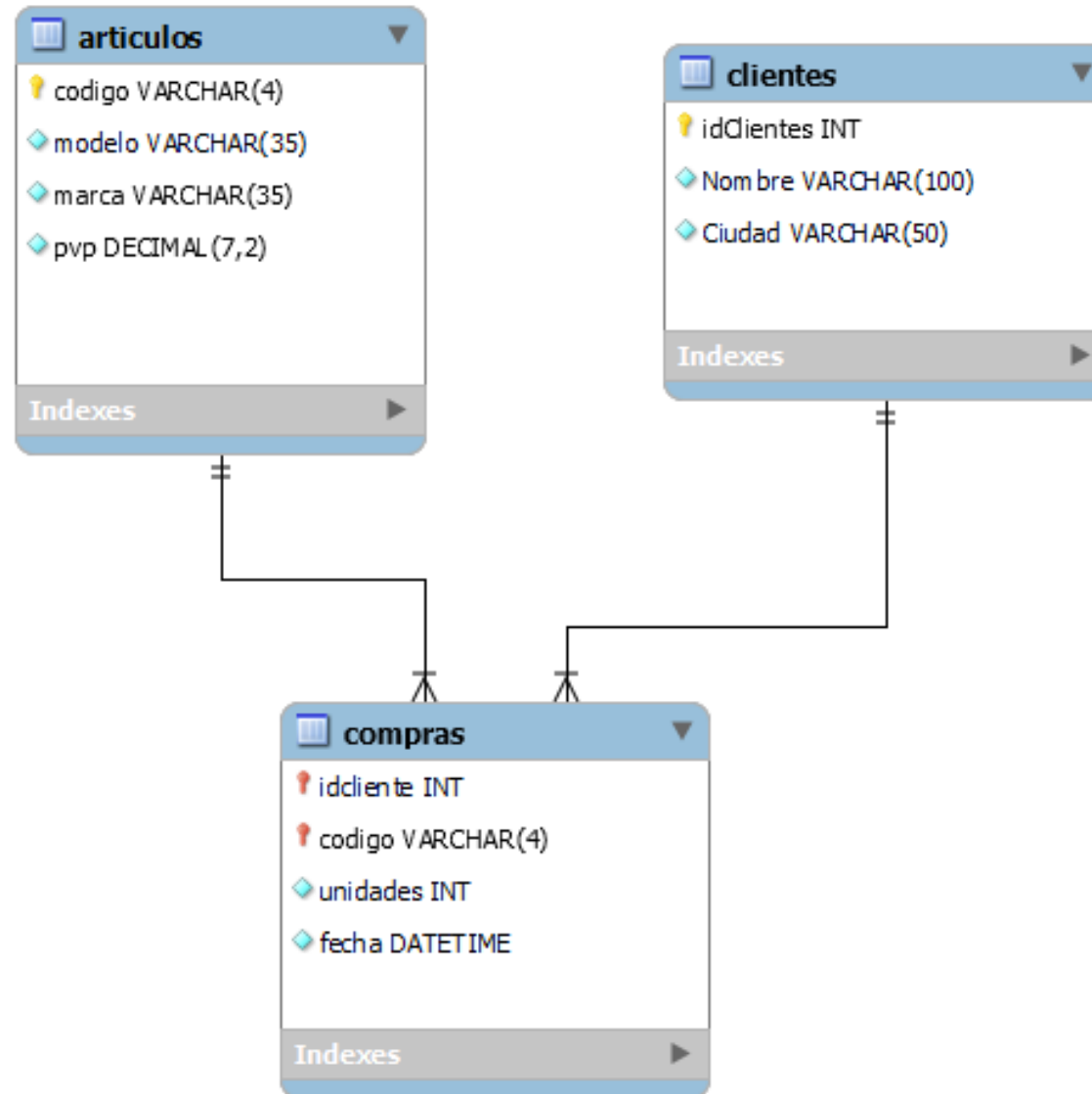
**Atributo ID único para cada autor**

```
<autor IdAutor="A2">  
  <Nombre>Fernando</Nombre>  
  <Apellidos>de Rojas</Apellidos>  
  <FechaNacimiento>01/01/1470</FechaNacimiento>  
</autor>
```

**Atributo IDREF que debe coincidir con uno de ID**

Puede afinarse más con una lista enumerada de posibles códigos de autores

# Modelo relacional a XML (con XML Schema)



# Modelo relacional a XML (con XML Schema)

```
<xs:element ref="clientes"/>
```

```
  <xs:element ref="cliente" minOccurs="0" maxOccurs="unbounded"/>
```

```
    <xs:element name="nombre" type="tipoNombre"/>
```

```
    <xs:element name="ciudad" type="tipoCiudad"/>
```

```
    <xs:attribute name="idcliente" use="required" type="xs:ID"/>
```

02ejemplo.xml

02ejemplo.xsd

```
<xs:element ref="articulos"/>
```

```
  <xs:element ref="articulo" minOccurs="0" maxOccurs="unbounded"/>
```

```
    <xs:element name="modelo" type="tipoModelo"/>
```

```
    <xs:element name="marca" type="tipoMarca"/>
```

```
    <xs:element name="pvp" type="tipoPvp"/>
```

```
    <xs:attribute name="codigo" use="required" type="xs:ID"/>
```

```
<xs:element ref="compras"/>
```

```
  <xs:element ref="compra" minOccurs="0" maxOccurs="unbounded"/>
```

```
    <xs:element name="unidades" type="xs:integer"/>
```

```
    <xs:element name="fecha" type="xs:dateTime"/>
```

```
    <xs:attribute name="idcliente" use="required" type="xs:IDREF"/> <!-- clave ajena -->
```

```
    <xs:attribute name="codigo" use="required" type="xs:IDREF"/> <!-- clave ajena -->
```