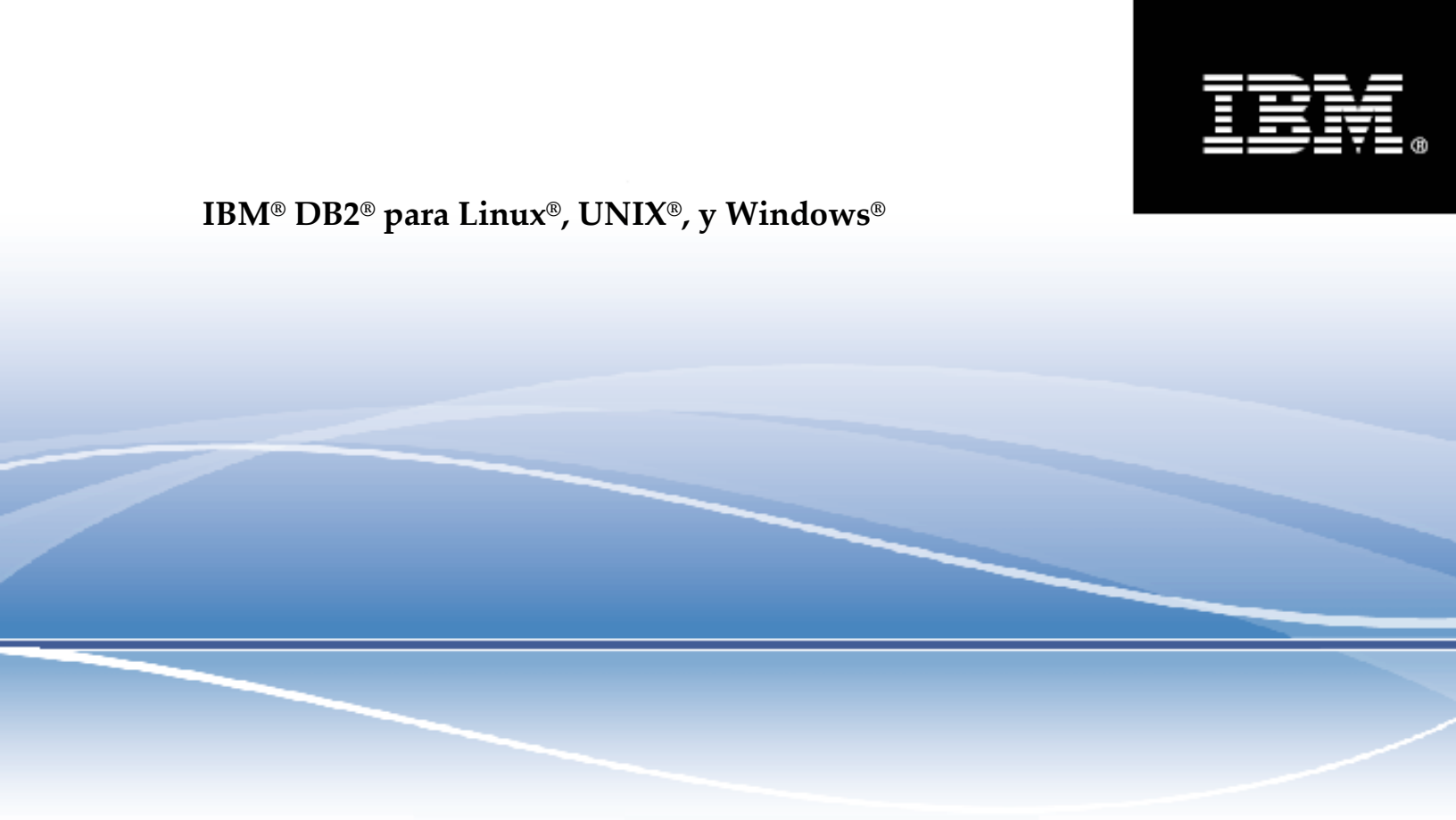

IBM® DB2® para Linux®, UNIX®, y Windows®

# Las mejores prácticas

## Administración de datos XML

Matthias Nicola
*Laboratorio Silicon Valley de IBM*

Susanne Englert
*Laboratorio Silicon Valley de IBM*

Resumen ejecutivo ..... 3

El por qué XML ..... 4

Ventajas y desventajas de los datos XML y de los datos relacionales ..... 4

Soluciones XML para los problemas de modelos de datos relacionales ..... 5

Beneficios de DB2 pureXML con respecto a otras opciones de almacenamiento ..... 7

# Resumen ejecutivo

Este documento presenta los principios y los lineamientos para utilizar DB2 pureXML™ para resolver con eficacia problemas de negocios y para lograr obtener un alto desempeño en la administración de datos XML en aplicaciones empresariales. Los ejemplos que ilustran las mejores prácticas se basan en escenarios de aplicaciones financieras reales y muestran cómo implementar los lineamientos. Los ejemplos pueden adaptarse con facilidad a otros tipos de aplicaciones XML. Este documento abarca las siguientes áreas:

- Opciones de almacenamiento para datos XML para mejorar el desempeño y la eficacia del almacenamiento.
- Técnicas para agregar datos XML a una base de datos DB2.
- Técnicas para realizar consultas y actualizar documentos XML con eficacia.
- Técnicas para utilizar índices en datos XML para realizar consultas con eficacia.
- Técnicas para el mantenimiento y el monitoreo eficaz en una base de datos XML.
- Técnicas para el desarrollo eficaz de aplicaciones pureXML.

# El por qué XML

XML ofrece una forma neutral y flexible para intercambiar datos entre sistemas, aplicaciones y organizaciones diferentes. Con XML, los datos se mantienen en un formato autodescriptivo y que puede ampliarse para satisfacer las necesidades siempre cambiantes de los negocios. Los documentos XML utilizan etiquetas para describir los valores de los datos que contienen, así como el anidado de etiquetas para expresar las relaciones jerárquicas que existen entre los elementos de los datos. XML puede describir datos muy estructurados e imponer la estructura a través de esquemas XML; sin embargo, también puede describir datos semiestructurados que prevalecen en aplicaciones orientadas en el contenido.

Las arquitecturas orientadas al servicio (SOA), la integración de aplicaciones empresariales (EAI), la integración de información empresarial (EII), los servicios web, el bus de mensajes empresarial (ESB) y los esfuerzos de estandarización en muchas industrias todos se basan en XML como la tecnología subyacente para el intercambio de datos.

Las organizaciones, al igual que todas las industrias, han estandarizado los esquemas XML para promover el intercambio de datos y están desarrollando esos esquemas para satisfacer las cambiantes necesidades de los negocios. Estos esfuerzos incluye ACORD en la industria de seguros, FpML® y FIXML en la industria financiera, RosettaNet en la administración de la cadena de suministros, ARTS en el negocio minorista, HL7 en el sector de servicios médicos, XBRL para la presentación de información empresarial y DITA para crear, administrar y publicar documentación impresa y en la web.

Esas iniciativas específicas por industria, así como los requerimientos normativos, son lo que impulsa el despliegue de XML. Conforme más transacciones de negocios se realicen a través de interfases basadas en la web y en formatos electrónicos, las dependencias gubernamentales y las empresas comerciales asumen una mayor responsabilidad respecto de la conservación de los pedidos, las solicitudes, las reclamaciones, las operaciones o los envíos originales. XML ofrece un medio directo para capturar y mantener los datos relacionados con esas transacciones electrónicas. De hecho, los documentos XML con frecuencia representan registros de transacciones en sistemas de procesamiento de transacciones basados en mensajes.

## *Ventajas y desventajas de los datos XML y de los datos relacionales*

Como un formato de datos autodescriptivo, XML permite que diversos datos (con o sin esquema XML) se almacenen en un solo documento o fila sin sacrificar la capacidad de investigar o de agregar partes de esos datos. Las aplicaciones pueden desarrollar sus esquemas XML sin provocar daños al esquema subyacente de la base de datos. Mientras que la flexibilidad de XML significa que el examinar e interpretar datos XML puede consumir más recursos del procesador y/o de entrada/salida que si los mismos datos se almacenaran en forma relacional, factores como la complejidad del esquema podrían hacer que el almacenamiento de los datos resultase poco práctico.

Con definiciones de esquemas más rígidas, el modelo relacional requiere considerablemente menos interpretación y permite operaciones de datos más optimizadas. De ese modo, esto puede ofrecer un muy alto desempeño pero podría no satisfacer los requerimientos de las aplicaciones respecto de la flexibilidad del esquema. El modelo de los datos relacionales se adecua muy bien a las aplicaciones con estructuras de datos estables y patrones de acceso predecibles. Con frecuencia, XML se adecua más a aplicaciones con estructuras de datos variables y complejas es ideal para la combinación de información estructurada y no estructurada.

En algunos casos, XML ofrece beneficios de desempeño con respecto a los modelos relacionales precisamente debido a su flexibilidad. Con frecuencia, los datos relacionales requieren de normalización para ajustar los datos de negocios en estructuras planas y tabulares. Esta normalización de datos de negocios complejos requiere de una transformación cuando los datos se almacenan y recuperan y, con frecuencia, da lugar a consultas conjuntas multidireccionales en bases de datos relacionales. XML puede constituir una representación más natural de los objetos de negocios complejos con todas las relaciones relevantes representadas en un documento único. Las jerarquías dentro de un documento XML son esencialmente uniones precalculadas entre los elementos de los datos relacionados.

Otra consideración en la elección de un modelo de datos es la aplicación que utiliza los datos. Incluso si los datos se originan en XML, si el procesamiento posterior de esos datos depende de que los datos se almacenan en un formato tabular —por ejemplo, cuando se aplica un procesamiento analítico en línea (OLAP) relacional a los datos en un almacén de datos— entonces, el almacenar los datos en un formato relacional en vez de hacerlo en XML sería la elección correcta.

## *Soluciones XML para los problemas de modelos de datos relacionales*

El modelo de datos de almacenamiento debe concordar, en la medida más amplia posible, con el modelo de uso de valor más elevado y más crítico para sus datos. Si los datos que se modelan son tabulares de manera natural, por lo regular, es mejor representarlos en un formato relacional que hacerlo con el formato XML. Sin embargo, existen algunos casos en los que el modelo relacional no es necesariamente la mejor elección y, en ocasiones, incluso se trata de una mala elección para manejar sus datos. Las siguientes son algunas situaciones en las que la representación XML tiende a ser más benéfica que el formato relacional.

### Cuando el esquema es volátil.

**Problema con los datos relacionales:** Si el esquema de los datos cambia con frecuencia, entonces la representación de los datos en forma relacional tiene como resultado que se incurre en costos y en una sobrecarga por la modificación del esquema relacional. Mientras que algunas formas de modificación al esquema son relativamente sencillas en las bases de datos relacionales, como el agregar una nueva columna a una tabla, otras formas son más complicadas, como el eliminar una columna o el modificar el tipo de una columna. Aún así, hay otras formas de modificaciones de esquemas que son absolutamente difíciles, como la normalización de una tabla en múltiples tablas. La modificación de las tablas significa entonces que las aplicaciones necesitan modificar las instrucciones SQL que tienen acceso a éstas.

**Solución con datos XML:** Las partes del esquema que son volátiles pueden expresarse como una única columna XML. La naturaleza autodescriptiva y susceptible de ampliarse de XML permite el manejo a la perfección de la variabilidad y la evolución de los esquemas. Las modificaciones en el formato del documento XML tienen lugar sin necesidad de modificar tablas o columnas en la base de datos y, por lo general, sin desarticular las consultas XML existentes.

### Cuando los datos son jerárquicos de manera intrínseca por su naturaleza.

**Problema con los datos relacionales:** Los datos que son jerárquicos o recursivos de manera intrínseca con frecuencia son difíciles de representar en esquemas relacionales. Como ejemplos de esto se incluyen listas de materiales, objetos de ingeniería o datos biológicos. La explosión de una lista de materiales puede almacenarse en una base de datos relacional pero el reconstruirla en partes o en su totalidad podría requerir el uso recursivo de SQL.

**Solución con datos XML:** Dado que XML es un modelo de datos jerárquico, éste es una representación mucho más natural para los datos de negocios jerárquicos de manera intrínseca. El utilizar XML permite un acceso a los datos simple y de navegación para reemplazar un conjunto de operaciones complejo si éste mismo se representara en un formato tabular.

## Cuando los datos representan objetos de negocios.

**Problema con los datos relacionales:** Si los datos de la aplicación representan objetos de negocios, como formularios de reclamación de un seguro, entonces, con frecuencia es benéfico mantener juntos los elementos de datos que integran una reclamación en particular, en lugar de esparcirlos en un conjunto de tablas. Esto es particularmente cierto cuando los elementos de los datos individuales de un formulario de reclamación no tienen significado comercial válido por sí mismos y sólo pueden interpretarse en el contexto del formulario completo. La normalización de reclamaciones a lo largo de docenas de tablas relacionales significa que la aplicación tiene que lidiar con una fragmentación compleja y poco natural de sus datos de negocios. Esto incrementa la complejidad y la probabilidad de que se presenten errores.

**Solución con datos XML:** XML le permite representar incluso objetos de negocios complejos como documentos cohesivos y definidos al mismo tiempo que sigue capturando todas las relaciones entre los elementos de los datos que integran el objeto de negocios. La representación de cada uno de los formularios de reclamación (el objeto de negocios) como un documento XML único en una única fila de una tabla ofrece un modelo de almacenamiento muy intuitivo para el desarrollador de aplicaciones y permite un rápido desarrollo de las aplicaciones.

## Cuando los objetos tienen atributos escasos

**Problema con los datos relacionales:** Algunas aplicaciones tienen un gran número de posibles atributos, la mayoría de los cuales son escasos, esto es, los atributos son aplicables a muy pocos objetos. Un ejemplo clásico es un catálogo de productos en donde el número de atributos de los diferentes productos es enorme, lo que incluye: tamaño, color, peso, longitud, altura, estilo, tipo de tejido, voltaje, resolución, resistencia al agua y una lista casi interminable de otras propiedades. Para cualquier producto dado, sólo un subconjunto de estos atributos es relevante. Un enfoque relacional posible es almacenar estos datos para tener una columna por atributo, lo que significa que un gran porcentaje de las celdas en la tabla contenga valores NULOS. Esto no es deseable y puede resultar ineficaz. Un enfoque relacional diferente para esos datos escasos es una tabla con tres columnas que almacene varios pares de nombres/valores para cada identificador de producto. Esto significa que los nombres de los atributos no son los nombres de las columnas sino valores en una columna del tipo VARCHAR (de longitud variable). Esto evita que los sistemas de bases de datos relacionales estimen con exactitud una selectividad de restricciones y generen planes de consulta eficaces. Asimismo, el definir e imponer restricciones, como la singularidad de un cierto atributo, es extremadamente complejo.

**Solución con datos XML:** La belleza de XML es que los elementos y los atributos pueden ser opcionales, de modo que simplemente se omiten si no se aplican a un producto específico. Ni los valores NULOS ni los pares de nombres/valores son necesarios. El esquema XML puede definir un gran número de elementos opcionales; sin embargo, sólo unos cuantos de éstos se utilizan para cualquier objeto dado. Mientras que en una tabla relacional cada fila debe tener el número exacto de columnas, los documentos XML en columnas XML pueden tener diferentes elementos de una fila a la otra. Asimismo, un índice XML para un elemento opcional será muy pequeño si este elemento aparece sólo en un porcentaje pequeño de documentos (filas). Ésta es una clara ventaja con respecto a los índices relacionales que tienen exactamente una entrada por fila.

## Cuando los datos necesitan intercambiarse

**Problema con los datos relacionales:** Si usted exporta un conjunto de filas de una tabla relacional y lo envía a una aplicación u organización distinta, el destinatario no puede interpretar los datos sin los metadatos adicionales que describan las columnas. Esto es particularmente cierto si su esquema relacional se ha modificado desde la última vez que envió los datos.

**Solución con datos XML:** Los datos XML son autodescriptivos. Las etiquetas XML son metadatos que describen los valores que les acompañan.

## *Beneficios de DB2 pureXML con respecto a otras opciones de almacenamiento.*

Dado que XML se ha vuelto cada vez más crítico para las operaciones de una empresa, los documentos XML son activos que necesitan compartirse, investigarse, protegerse y actualizarse con total congruencia transaccional. En función de su uso, los datos XML podrían también necesitar transformarse, auditarse e integrarse con otros datos. Para satisfacer estos requerimientos, el almacenamiento de datos XML en su formato jerárquico nativo en una base de datos DB2 tiene diversas ventajas, entre las que se incluyen:

- La retención del conocimiento sobre la estructura interna de los datos XML. Esto tiene la ventaja con respecto al almacenamiento de documentos XML como objetos de caracteres de gran tamaño (CLOBs) u objetos binarios de gran tamaño (BLOBs) en la base de datos. En particular, usted puede realizar consultas con facilidad en los datos XML utilizando XQuery, XPath y SQL/XML para aprovechar la estructura XML y puede mejorar el desempeño de las consultas creando índices con respecto a los datos XML. Además, puede actualizar, transformar y publicar con facilidad datos XML utilizando SQL, XQuery y XSLT.

- El mantenimiento de la naturaleza jerárquica y flexible de los datos XML. Esto tiene ventaja con respecto a la descomposición (fragmentación) de los documentos XML en tablas relacionales en donde un administrador mapea los elementos XML y los atributos hacia columna relacionales. Después de la fragmentación, los valores de los documentos XML se almacenan en estas tablas sin sus etiquetas originales. Con frecuencia, la fragmentación requiere un gran número de tablas y, por lo regular, esto resulta demasiado complejo para ser práctico. Las consultas en documentos XML descompuestos pueden requerir uniones SQL complejas que tienden a ser difíciles de desarrollar y afinar. Los cambios en el esquema XML con frecuencia rompen el mapeo al esquema de la base de datos relacional. Esto conlleva incurrir en mantenimiento costoso y que conlleva mucho tiempo lo que anula la flexibilidad por la que por lo regular se elige XML. Es por esto que DB2 pureXML le permite utilizar una columna XML única para almacenar y realizar consultar en documentos XML que estén basados en diferentes esquemas XML o en versiones diferentes de un esquema XML en evolución.

- La integración de documentos XML con datos relacionales en una base de datos única. Esto tiene ventajas con respecto al almacenamiento de datos relacionales en una base de datos y de documentos XML en otra base de datos exclusiva para XML por separado. Este enfoque requiere habilidades y personal para operar y mantener dos sistemas de bases de datos en lugar de sólo un sistema. Asimismo, la combinación de datos de dos bases de datos por lo general requiere lógica extra en la aplicación, lo que con frecuencia es difícil e ineficaz. Cuando usted almacena tanto datos XML como

relacionales en una base de datos DB2 única, usted puede combinar ambos tipos de datos en consultas y realizar uniones entre éstos e incluso convertirlos de uno a otro conforme necesite. Esto puede resultar potencialmente más rentable y ofrece un mejor desempeño que el utilizar dos bases de datos por separado.