RESIGeT On-Line
Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica
Vol. 02 - Nr. 01 - 2012
Uni-FACEF CENTRO UNIVERSITÁRIO DE FRANCA

SISTEMA GERENCIADOR DE BANCO DE DADOS: SGBD eXist XML

TANAKA, Luís Carlos[1]
tanaka@tanaka.pro.br

CAMARGO, Felipe Melo[2]
felipemelo.si@gmail.com

GOTARDO, Reginaldo[3]
reggotardo@gmail.com

RESUMO
O objetivo deste artigo é apresentar uma revisão do Sistema Gerenciador de Banco de Dados (SGBD) eXist XLM. O estudo está em andamento e tem sido feito essencialmente a partir de pesquisa bibliográfica e documental e pesquisa de caráter exploratório sobre Banco de Dados com uso da linguagem de marcação XML e a integração com um SGBD, o eXiste XML. Como resultado desta revisão, apontam-se as características deste BD, propondo uma reflexão sobre o tema objeto da pesquisa, visto a diversidade de aplicações baseadas em WEB que se utilizam da tecnologia XML para troca de dados, e permitindo assim o debate acerca do armazenamento dos dados também com o uso desta tecnologia, baseada em documentos XML.

Palavras-chave : eXist; Banco de Dados XML; XML nativo;

ABSTRACT
The aim of this paper is to review the System Manager Database (DBMS) XLM eXist. The study is ongoing and has been done mainly from bibliographical and documentary research and exploratory research on the database using the XML markup language and integration with a DBMS, the XML exists. As a result of this review, indicate the characteristics of BD, proposing a reflection on the theme object of research, given the diversity of Web-based applications that use XML technology to exchange data, and thus the debate concerning the storage the data also using this technology, based on XML documents.

Keywords: eXist; XML database, native XML;

[1] [2], - *Discente do Curso de Bacharelado em Sistemas de Informação do Centro Universitário de Franca Uni-FACEF*
[3] - *Docente do Curso de Bacharelado em Sistemas de Informação do Centro Universitário de Franca Uni-FACEF*

INTRODUÇÃO

XML é atualmente o padrão mais utilizado na internet para transferência de dados, em vários tipos de aplicações e sistemas onde exista a necessidade de troca de informação por meios digitais. Na visão de Banco de Dados um conjunto de documentos XML que persistem e podem ser manipulados pode ser considerado um Banco de Dados XML. O Sistema Gerenciador de Banco de Dados (SGBD) eXiste é o objeto de pesquisa deste estudo.

O trabalho proposto será realizado primeiramente por meio de análise bibliográfica e documental, principalmente, com busca na internet referente ao tema. O artigo apresenta alguns conceitos já obtidos na pesquisa bibliográfica. São abordados temas pertinentes ao objeto de pesquisa, como: História do Armazenamento de Dados, o que e Banco de Dados, o que é XML, diferença entre XML e HTML, Banco de Dados eXist XLM, Armazenamento de Dados em XML, SGBD eXist XML, Características dos SGBS XML nativos e por fim uma demonstração das operações básicas de CRUD no SGBD eXiste.

Banco de Dados XML segundo Duarte (2006, p.40), não têm a intenção de marcar história como uma nova geração de SGBD, uma vez que Banco de Dados Relacionais continuam sendo adequados a aplicações que lidam com dados estruturados.

1. HISTÓRIA DO ARMAZENAMENTO DE DADOS

O processamento de dados, que já existia antes mesmo da invenção dos computadores, mas conforme Silberschatz et al. (2006 p.19), foi o principal fator de impulsão do crescimento destas máquinas. No início do século XX os cartões perfurados inventados por Herman Hollerith para o censo dos Estados Unidos eram processados por sistemas mecânicos para tabular os resultados, e foram mais tarde, amplamente utilizados como meio de se inserir dados nos computadores eletrônicos.

As técnicas para armazenamento e processamento de dados evoluíram ao logo dos anos: nas décadas de 50 e 60, fitas magnéticas eram usadas para armazenamento; na década de 70, os discos rígidos e surgem as definições do modelo relacional para banco de dados; na década de 80, surgem o banco de dados relacional, SQL e as redes; na década de 90, explode a Word Wide Web; e em 2000

o surgimento da XML e da linguagem xQuery. (Adaptado de SILBERSCHATZ et al. 2006. p.19 e 20).

Sobre a importância do XML para o armazenamento de dados, “assim como SQL é a linguagem dominante para a consulta de dados relacionais, XML tornou-se o formato dominante para a troca de dados” (SILBERSCHATZ et al. 2006. p.266), .

1.1. O que é Banco de Dados

Segundo Duarte (2006 p.24) apult Graves, um Banco de Dados é um conjunto de dados armazenados de maneira que persistam e possam ser manipulados. Entendendo que persistência é a permanência dos dados em seus locais depois que o trabalho que os utilizada for encerrado e o computador desligado. Sendo assim a maioria dos arquivos são persistentes, como arquivos textos, planilhas e as figuras. Na visão de Banco de Dados um conjunto de documentos XML que persistem e podem ser manipulados pode ser considerado um Banco de Dados XML.

Os elementos XML apresentam alguns recursos que os objetos não possuem como a serialização, ou seja, a habilidade de ser representados como uma *string*. Os objetos possuem alguns recursos que a XML também não tem como os métodos e a herança. Os objetos e os elementos XML ocupam posições opostas quando se trata de encapsulamento, com os objetos ocultando toda a estrutura interna, sendo que a XML a expõe transparentemente. Por causa destes recursos oferecidos pelos objetos, o tornam melhor para a programação, porém faz com que a XML seja uma ferramenta mais adequada para representação de dados, com isto é muito mais fácil trabalhar com a XML do que com os objetos para transferência de dados. (DUARTE, 2006. p.14 e 15).

2. O QUE É XML

XML do inglês *e**X**tensible **M**arkup **L**anguage* é uma linguagem de marcação recomendada pela W3C (*World Wide Web Consortium)*, um consórcio de empresas de tecnologia que visa padronizar a criação e interpretação de conteúdos para *websites*, para a criação de documentos com dados organizados hierarquicamente. Conforme a W3Schools (2012), o XML é uma linguagem de marcação parecida com HTML projetado para transportar dados.

A linguagem XML possui uma maneira especial e muito fácil de tratar os dados e informações, é uma linguagem que utiliza TAG, estas podem ser lidas tanto por homem quanto por máquina. (DUARTE, 2006. p.15).

As TAG XML não são predefinidas, projetado para ser auto descritivo e recomendado pela W3C, permite definir suas próprias TAG, personalizando-as conforme a necessidade.

2.1. Diferença entre XML e HTML

XML não é um substituto para HTML, destaca a W3Schools (2012). XML e HTML foram projetados com objetivos diferentes: O XML foi projetado para transportar e armazenar dados, com foco sobre o que dados são; e o HTML foi projetado para exibir dados, com foco em como os dados aparecem. Assim, cada um dispõe sobre uma característica fundamental, o HTML é sobre a exibição de informações, enquanto que XML é sobre a transmissão de informação. (W3SCHOOLS, 2012).

3. BANCO DE DADOS EXIST XML

A figura 1 apresenta o logo oficial do Banco de Dados eXist, um banco de dados *Open Source* nativo XML, com características de processamento baseado em índices XQuery, suporte XUpdate. Pasquali (2012, p.3), define o eXist como sendo “um banco de dados XML de código aberto, completamente desenvolvida em Java, que pode ser integrado em aplicações que utilizam XML”.

Figura 1: Logo do BD eXist
Fonte**:** http://www.exist-db.org (2012)

Segundo Pasquali (2012, p.3), o eXist oferece armazenamento de documentos XML sem esquemas através de coleções hierárquicas. Através da sintaxe XPath, usuários podem recuperar partes distintas da coleção ou todos os documentos contidos no banco de dados.

Conceito de coleções e recursos adotado pelo eXist: O termo *Collection* é utilizado para se referenciar a um diretório de armazenamento de documentos XML; e Resources: são os próprios documentos XML que são armazenados nas coleções. (PASQUALI, 2012, p.3). Para Bourret (2012), “uma coleção é um conjunto de documentos relacionados e desempenha um papel semelhante ao de uma tabela na base de dados relacional ou um diretório em um sistema de arquivo”.

O módulo de consulta do eXist processa de forma eficiente consultas baseadas em índice. O esquema de indexação suporta a identificação rápida das relações estruturais entre os nós das árvores que representam o documento XML. (PASQUALI, 2012, p.3).

O banco de dados XML é mais apropriado para aplicações que utilizam de pequenas à grandes coleções que são ocasionalmente atualizadas. O eXist oferece extensões para o padrão XPath para processar consultas *fulltext*, incluindo por palavras chaves, por proximidade ou expressões regulares. (PASQUALI, 2012, p.3).

3.1. Armazenamento de Dados em XML

Existem duas maneiras diferentes de "armazenar" documentos XML em um banco de dados. O primeiro é mapear esquema do documento para um esquema de base de dados e os dados de transferência de acordo com o mapeamento. A segunda é usar um conjunto fixo de estruturas que podem armazenar qualquer documento XML. (BOURRET, 2012).

Em um banco de dados XML nativo, um documento é a unidade fundamental de armazenamento, o equivalente a uma linha em um banco de dados relacional. (BOURRET, 2012).

Os documentos XML não possuem uma estrutura rígida que defina os dados, por este motivo os dados em um documento XML são denominados dado semi-estruturados. Os dados semi-estruturados, em sua maioria, possuem como características principais uma estrutura irregular, dinâmica e bastante heterogênea. (DUARTE, 2006. p.25).

4. SGBD EXIST XML

Um fator importante na escolha de um Banco de Dados é se ele será usado para armazenar dados ou documentos: Para dados, é necessário um SGBD desenvolvido para o armazenamento de dados, como um Banco de Dados

Relacional; Para documentos, a escolha é um sistema de gerenciamento de conteúdo, que seja projetado para armazenar documentos, como um Banco de Dados XML Nativo. (DUARTE, 2006. p.26).

Bases de dados XML nativos são bancos de dados projetados especialmente para armazenar documentos XML. Como outros bancos de dados, que suportam funcionalidades como transações, segurança, acesso multi-usuário, API de programação, linguagens de consulta, e assim por diante. A única diferença de outros bancos de dados é que o seu modelo interno é baseado em XML e não outra coisa, como o modelo relacional. (BOURRET, 2012). A figura 2 apresenta a tela do Cliente de Administração do SGBD eXiste XLM.

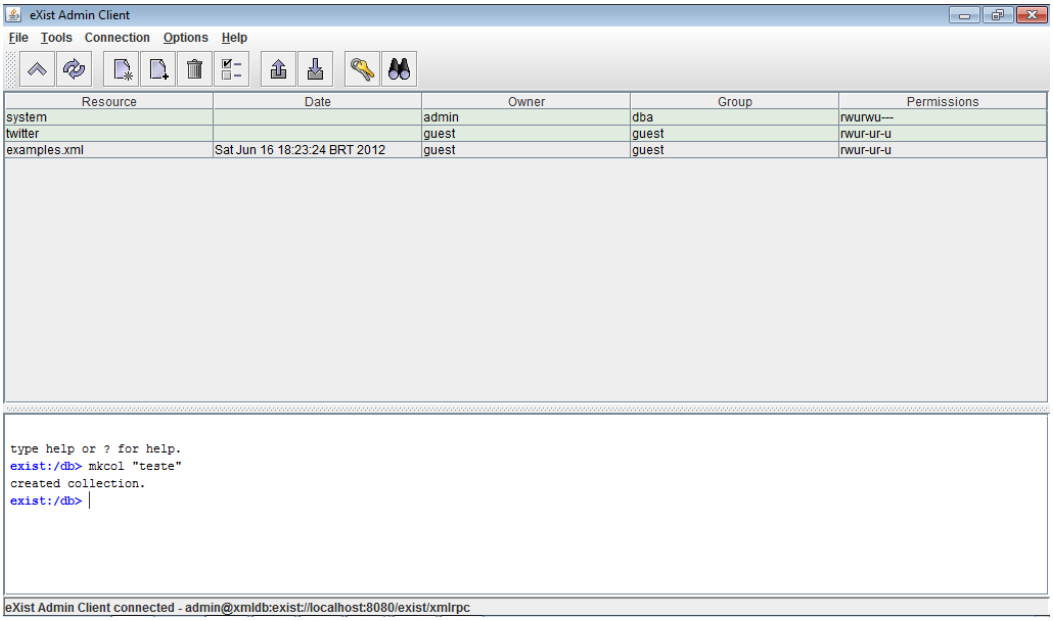

Figura 2: Cliente de Administração do eXist
Fonte: SGBD eXist XML

Bourret (2012), considera com propriedade que o eXist indexa automaticamente todos os elementos e a estrutura do atributo. Por padrão, cria índices de texto completo sobre todo o texto e os valores dos atributos, mas os usuários podem desligar isso por partes selecionadas de um documento. Ele suporta simultânea leitura/gravação de acesso para vários usuários. A segurança é fornecida por meio de permissões de acesso do Unix de estilo para usuários e grupos, que podem ser aplicadas a ambas as coleções e documentos individuais. XQuery controle de acesso é suportada através do *eXtensible Access Control Markup Language* (XACML). As transações são suportadas, mas "limitado à funcionalidade

necessária para a recuperação de falhas", não são visíveis para aplicativos. A figura 3 apresenta a tela principal de do SGBD eXist XML com interface WEB, executando em modo local (localhost:8080/exist/index.xml).

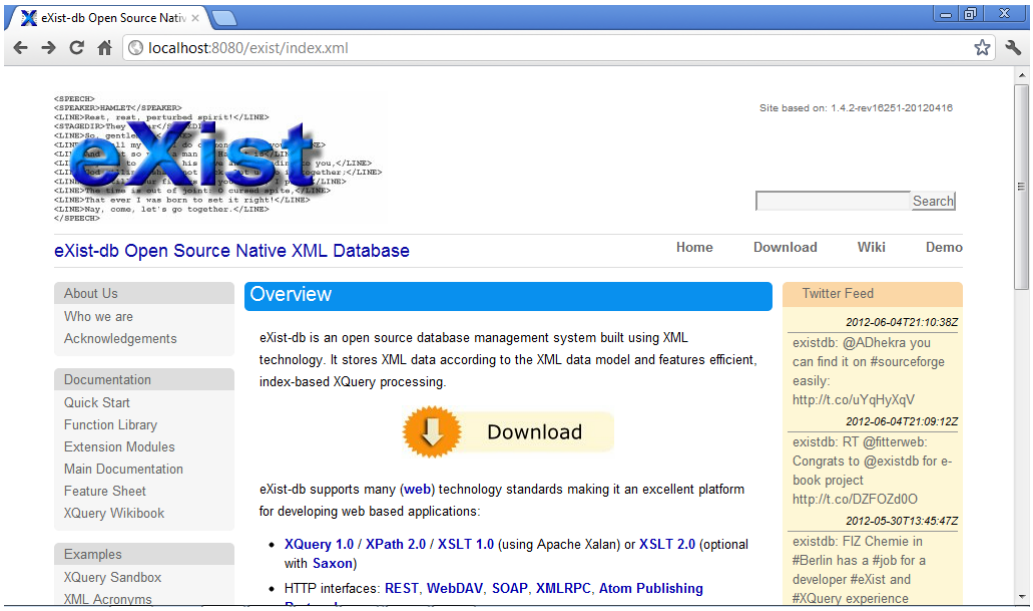

Figura 3: Tela Principal do SGBD eXist XML
Fonte: SGBD eXist XML

4.1. Características dos SGBS XML Nativos

4.1.1. Coleções de documentos

Muitos SGBD suportam a noção de coleção. Elas têm papel similar às tabelas em bancos relacionais ou diretórios em um sistema de arquivos. (DUARTE, 2006. p.36).

4.1.2. Linguagens de consulta

Quase todos os SGBD suportam uma ou mais linguagens de consulta, sendo que as mais populares são a XPath e a XQuery. DUARTE, 2006. p.36).

4.1.3. Transações, travamento e concorrência

Todos os SGBD XML nativos suportam transações. Entretanto, o travamento é feito em nível de documento e não em nível de fragmentos de documento; consequentemente, concorrência entre usuários deve ser baixa. Obviamente que isto depende da aplicação e o que é entendido como um documento, ou seja, como ele é constituído. DUARTE, 2006. p.36).

4.1.4. Dados remotos

Alguns SGBD XML nativos podem incluir dados remotos em documentos armazenados no banco. Geralmente esses dados são recuperados de um banco relacional usando ODBC, OLE DB ou JDBC, por exemplo. DUARTE, 2006. p.37).

4.1.5. Índices

Assim como em qualquer Banco de Dados, os índices são usados para aumentar a performance em consultas. Quase todos os SGBD XML nativos suportam a indexação de elementos e atributos. DUARTE, 2006. p.37).

4.1.6. Normalização

A normalização de dados em um banco XML nativo é basicamente idêntica a normalização de um banco relacional. Uma boa modelagem dos documentos deve garantir que nenhum dado seja repetido, causando inconsistências. Uma vantagem da normalização em bancos XML nativos é que eles suportam propriedades multivaloradas, enquanto que a maioria dos relacionais não suporta. Isso torna possível normalizar os dados de uma forma mais simples e intuitiva, já que pela própria natureza hierárquica de XML um elemento pai pode ter vários elementos filhos. DUARTE, 2006. p.37).

4.1.7. Integridade referencial

Assim como para normalização, a integridade referencial também é semelhante a bancos relacionais em banco XML nativos. Em resumo, serve para garantir a validade de ponteiros entre dados de diferentes tabelas. Em bancos relacionais, a integridade referencial garante que chaves estrangeiras apontem para chaves primárias válidas. Em bancos XML nativos, a integridade referencial garante que mecanismos de "*linkagem*" apontem para documentos ou fragmentos de documentos válidos. DUARTE, 2006. p.37).

4.1.8. Segurança

O W3C *(World Wide Web Consortium)* define padrões de segurança para a linguagem XML, padrões *Encryption Syntax and Processing e Decryption Transform for XML Signature*.

A utilização de modelos XML para obtenção de autenticação, autorização, não repúdio, sigilo e integridade devido o aumento do uso de dados no formato XML. A segurança pode ser inserida dentro dos próprios documentos, e de forma padronizada, para que um documento possa ser entendido por qualquer sistema compatível, sendo assim intercambiável, ou seja, não é necessária uma análise prévia da estrutura do documento, já que está é conhecida pelo padrão XML. (KATO e MORAES, 2005 p.91)

4.1.9. Escalabilidade

Escalabilidade é a capacidade de manter a alta performance mesmo com um aumento significativo do tamanho ou carga do banco. Assim como bancos hierárquicos e relacionais, os XML nativos usam índices para procurar dados. Isto significa que localizar documentos e fragmentos de documentos está relacionado com o tamanho do índice e não com tamanho ou quantidade de documentos. Dado isso, a performance dos bancos XML nativos é a mesma comparada a outros tipos de bancos. Conseqüentemente, a escalabilidade desses bancos é semelhante a dos outros. DUARTE, 2006. p.38).

Ao contrário dos bancos relacionais, os bancos XML nativos "*linkam*" dados relacionados fisicamente, os quais "*linkam*" usando links lógicos. Por esse motivo, destaca Duarte (2006, p.38), eles deveriam ter uma boa escalabilidade com relação à consulta de dados. No entanto, esta escalabilidade é limitada; essa ligação física somente se aplica a uma hierarquia particular, assim a recuperação de dados em uma hierarquia diferente não é tão rápida quanto à recuperação em uma mesma hierarquia.

Segundo Duarte (2006, p.38), para minimizar esse problema, os bancos XML nativos usam largamente índices, algumas vezes até indexando todos os elementos e atributos. Isto pode ser uma solução para bancos que são mais usados para consultas, mas quando muitas atualizações são feitas, essa performance cai drasticamente.

Quando um banco XML nativo faz consultas por dados não indexados, eles se comportam muito mal, bem pior que os bancos relacionais, principalmente devido à normalização não tão boa. Por exemplo, em um banco relacional, para procurar todos os filmes vendidos em determinada data, o banco deve ler todas as colunas "dataVenda". No caso de um banco XML nativo, o banco deve percorrer todos os documentos de vendas por completo e procurar pelo elemento <dataVenda>. (DUARTE, 2006. p.38).

5. CRUD EM BANCO DE DADOS EXIST

CRUD é o acrônimo de *Create, Read, Update e Delete* em língua inglesa para as quatro operações básicas utilizadas em bancos de dados que são: *Create*: Criar ou adicionar novas entradas; *Read* (*Retrieve*): Ler, recuperar ou ver entradas existentes; *Update:* Atualizar ou editar entradas existentes; e *Delete* (*Destroy*): Remover entradas existentes.

A figura 4 ilustra a tela incial do *eXist Client Shell*, presente na pasta de instalação do eXist para inicializar o SGBD.

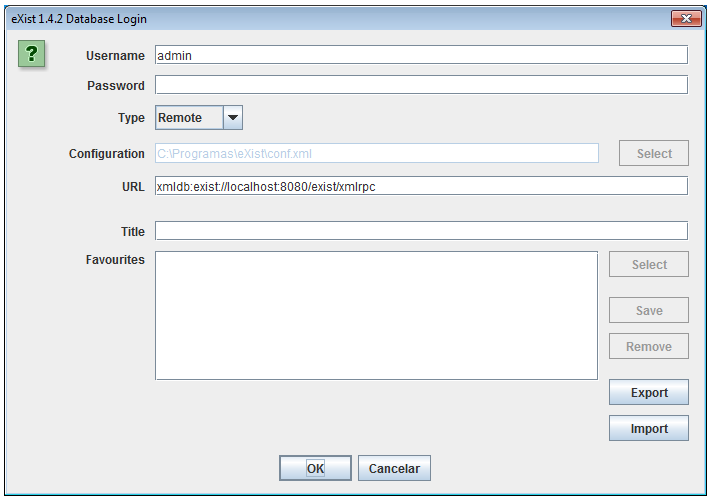

Figura 4: Tela de Login
Fonte: Client eXist Shell

Após o *login* no *Client eXist Shell* é apresentada a tela principal do SGBD conforme figura 5.

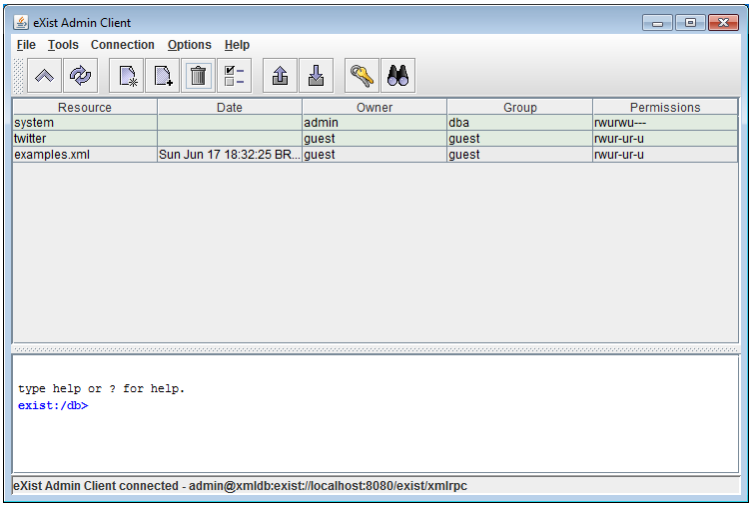

Figura 5: Tela de Principal
Fonte: Client eXist Shell

5.1. Create

Para demostrar a operação de *Create*, é utilizado o editor de textos *Notepad* para criar um arquivo XML, salvo com o nome teste.xml na pasta de instalação do eXist, contendo o código abaixo:

```
<facef>
    <curso>Sistemas de Informacao</curso>
</facef>
```

É criado um arquivo XML com a TAG pai <facef> e TAG filho <curso> com o conteúdo “Sistemas de Informacao”.

Em seguida, cria-se de uma nova coleção, no ícone *Create new collection*, de nome “teste”, para armazenamento de arquivos XML, onde será inserido o arquivo XML teste.xml do passo anterior e ilustrado na figura 6.

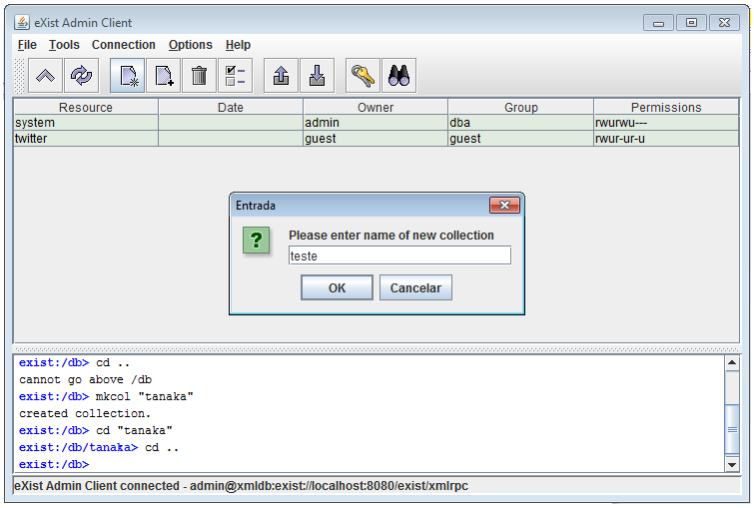

Figura 6: Criando uma Collection
Fonte: Client eXist Shell

Na sequência, adiciona-se o arquivo teste.xml na coleção criada anteriormente e ilustrada na figura 7, seguindo quatro passos básicos: Passo 1: seleciona a coleção "teste"; Passo 2: clica no ícone "*Store one or more files to the database*", Passo 3: na janela de seleção que se abre, selecionar o arquivo teste.xlm; e Passo 4: Clique no botão "*Select files or directories to store*", e na janela seguinte clique no botão "*Close*" para concluir o processo.

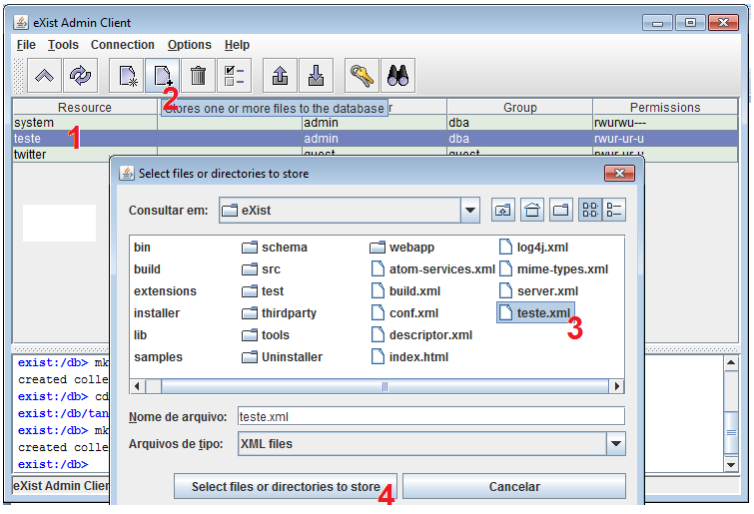

Figura 7: Adicionando arquivo XML à Coleção
Fonte: Client eXist Shell

Criado o arquivo XML e a coleção, para as operações de *Insert, Update e Delete*, é utilizada a interface WEB e no menu a opção "*xQuery Sandbox*" já ilustrada na figura 3, através de um browser no endereço:

http://localhost:8080/exist/index.xml, ilustrado a figura 8. A *Sandbox* permite: 1: Selecionar a coleção; 2: Digitar comandos em linguagem xQuery; 3: O envio dos comando digitados para serem processados pelo eXist; e 4: Exibir resultados dos comandos enviados, através de consultas.

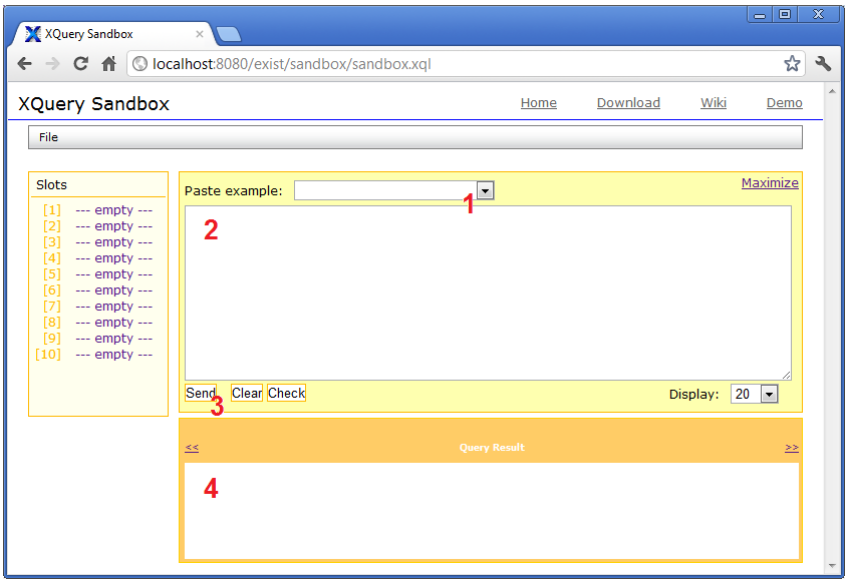

Figura 8: Tela da SandBox
Fonte: eXist SandBox

5.2. Select

A operação de *Select* pode retornar dois modos: Todo conteúdo do arquivo teste.xml; ou apenas os dados do arquivo XML ou o conteúdo das TAG.

Ilustrado na figura 9 o resultado do *Select* com o retorno de todo conteúdo do arquivo teste.xml, na seguinte seguência: Passo 1: Digita-se os comandos em linguagem xQuery; Passo 2: Clica no botão "*Send"* para envio ao SGBD; e Passo 3: É exibido o retorno. Para que o *Select* retorne todo XML é utilizado os comandos abaixo:

```
For $v in /facef
Return $v
```

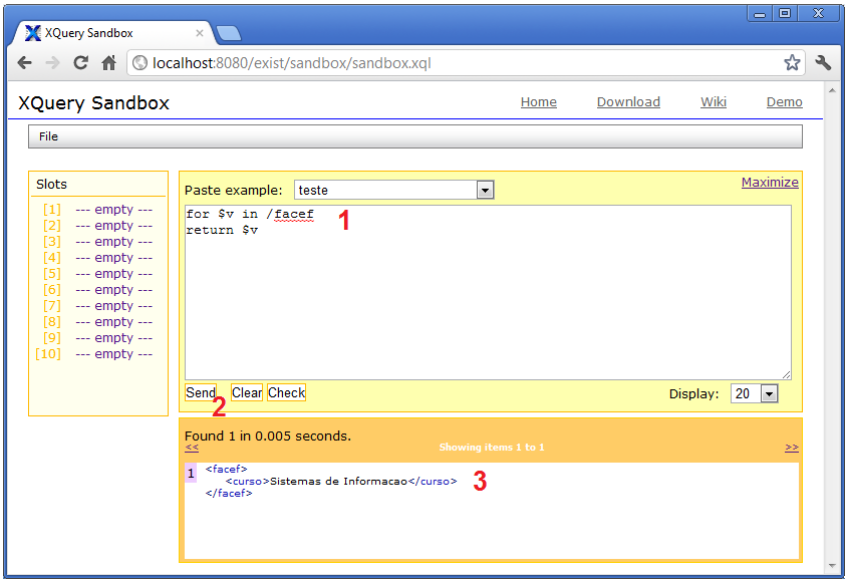

Figura 9: Select retornando todo XML
Fonte: eXist SandBox

Da mesma forma, e na mesma sequência, para o exemplo a seguir a operação de *Select* retorna apenas os dados, ou seja, o conteúdo das TAG, do arquivo teste.xml, ilustrado na figura 10, sendo executado o seguinte comando em linguagem xQuery:

```
For $v in /facef
Return data($v)
```

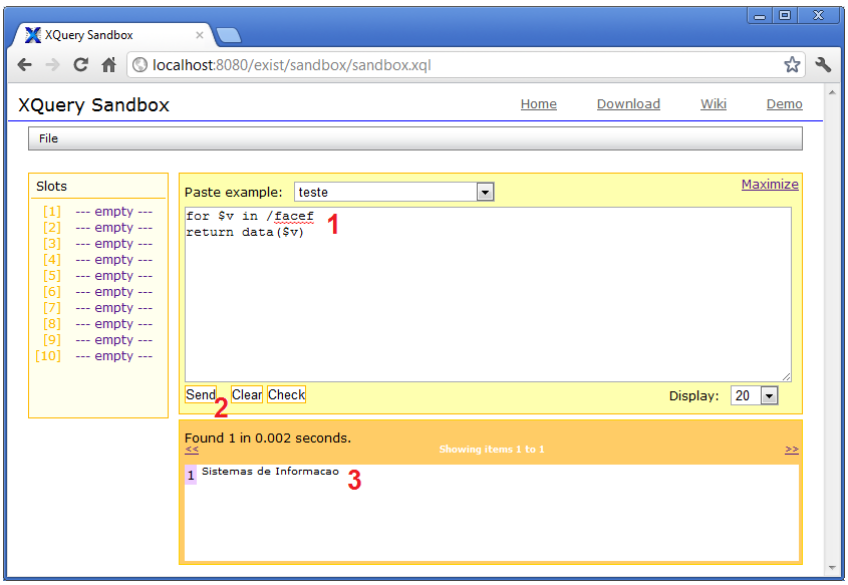

Figura 10: Select retornando somente os dados
Fonte: eXist SandBox

5.3. Insert

Para operação de *Insert*, que acrescentará uma nova TAG no arquivo teste.xml conforme ilustrado na figura 11, o comando abaixo resultará na inserção de uma nova TAG <disciplina> com o conteúdo “Banco de Dados” na TAG pai “facef” e TAG filho “curso” onde o conteúdo desta for “Sistemas de Informacao”.

```
update insert <disciplina>Banco de Dados</disciplina> into //facef[curso="Sistemas de Informacao"]
```

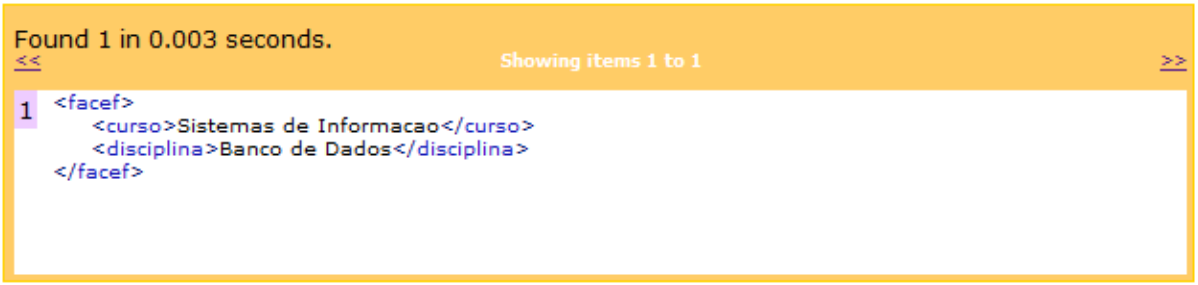

Figura 11: Resultado após Insert
Fonte: eXist SandBox

5.4. Update

Para a operação de *Update*, o comando abaixo em linguagem xQuery efetua a troca dos dados da TAG <disciplina>, e altera o conteúdo de “Banco de Dados” para “Rede de Computadores” atualizando-a, conforme exibido na figura 12.

```
update replace //disciplina[. = "Banco de Dados"] with <disciplina>Rede de Computadores</disciplina>
```

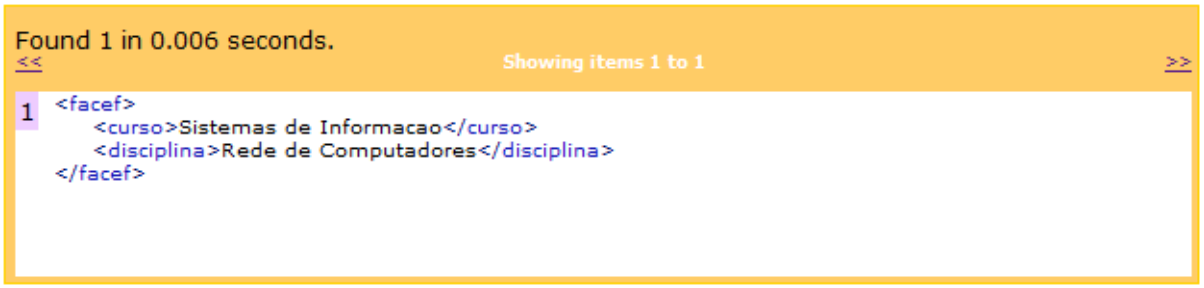

Figura 12: Resultado após Update
Fonte: eXist SandBox

5.5. Delete

Finalmente, a operação de *Delete* exibida na figura 13, o comando em linguagem xQuery exclui a TAG <disciplina> cujo conteúdo “Rede de Computadores” descrito no comando deverá ser excluída.

```
for $x in //facef/disciplina[. = "Rede de Computadores"] return update delete $x
```

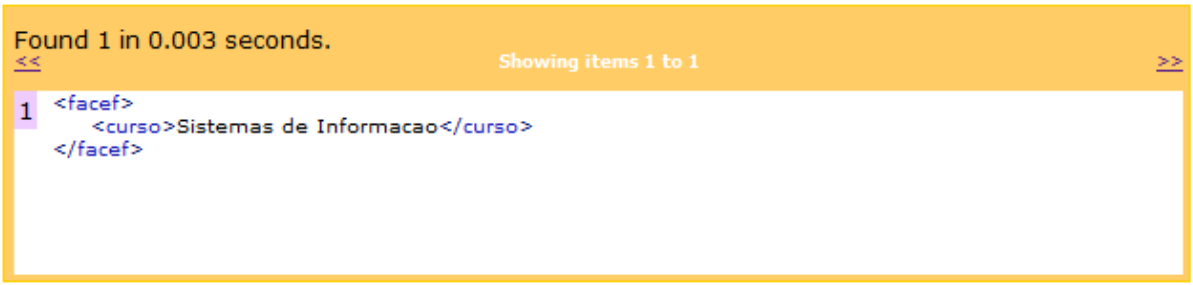

Figura 13: Resultado após o Delete
Fonte: eXist SandBox

CONSIDEREAÇÕES FINAIS

A linguagem de marcação XML vem sendo fortemente utilizada em aplicações na Web. Conforme Duarte (2006, p.60), esta sendo muito utilizada como uma forma de intercâmbio de dados, já que é uma linguagem fácil, portátil que não depende de Sistema Operacional ou linguagem de programação.

Então, devem-se levar em conta diversos fatores antes de se decidir por um banco XML nativo, como se a aplicação vai recuperar os dados na ordem que eles foram armazenados, assim não haverá problemas e deverá escalar bem, caso contrário escalabilidade pode ser um problema. (DUARTE, 2006. p.39).

Banco de Dados XML nativos deveriam trabalhar de maneira mais eficiente do que os Bancos de Dados Relacionais quando o assunto fosse armazenamento de documentos XML. Segundo Duarte (2006, p.60), isto não acontece, pois este tipo de Banco de Dados, ainda se encontra no estágio de desenvolvimento e possui algumas limitações, muitas destas já supridas por Banco de Dados Relacionais como: o Oracle, MS-SQL e DB2.

REFERÊNCIAS

BOURRET, Ronald. Consulting, writing, and research in XML and databases. Acesso em 16 de junho de 2012. Disponível em: http://www.rpbourret.com/xml/ProdsNative.htm#exist

DUARTE, Wellington Silva. O Estado da Arte em Armazenamento de Dados XML em Banco de Dados. 2006. Monografia (Bacharelado em Ciência da Computação) – Curso de Ciência da Computação da Faculdade de Jaguariúna, Jaguariúna - SP.

KATO, Luiz Henrique Wiggers; MORAES, Marcos Hideki Watanabe de. PADRÕES DE SEGURANÇA APLICADOS À LINGUAGEM XML. 2005. 118p. Trabalho de Conclusão de Curso, Curso de Bacharelado em Sistemas de Informação, Universidade Federal de Santa Catarina, Florianópolis (SC), 2005.

PASQUALI, Fábio; DUARTE, Dênio. Estudo Comparativo dos BDXML eXist e Xindice. 2012. Universidade Comunitária Regional de Chapecó - Unichapecó - Centro Tecnológico - CETEC. Chapecó – SC.

SILBERSCHATZ, Abraham; KORTH, Henry F.: SUDARSHAN, S. Sistema de Banco de Dados. 5ª edição. Tradução de Daniel Vieira. Rio de Janeiro: Elseviver, 2006.

W3Schools. Introduction to XML. Acesso em 16 de junho de 2012. Disponível em: http://www.w3schools.com/xml/xml_whatis.asp