cabecalho_timbrado_cin_producao

Aluno: Tiago Cordeiro de Melo Nascimento

Big Data

Desafios e Oportunidades

Recife, 15 de novembro de 2012

**Índice**

1. Introdução 3

2. Motivação 3

3. SGBD 3

4. XML 4

5. SGBD compatível com XML 4

6. SQL Server 5

7. Conclusão 7

8. Bibliografia 8

# 

# Motivação

O volume de dados gerados pelo mundo está aumentando. Um estudo elaborado pela IDC [1] em 2011 mostra o volume de dados estão mais do que dobrando a cada dois anos e deve atingir 11,8 zettabytes (1,8 trilhões de gigabytes). Podemos dizer que estamos vivendo numa explosão de dados.

Segundo a IBM [2], 90% dos dados armazenados no mundo hoje foram criados nos últimos dois anos. Outros estudos [3] [4] revelam que 30 bilhões publicações são compartilhadas no Facebook por mês. Um milhão de transações de clientes são geradas por hora no Wal-Mart. E em 2020, as empresas terão que administrar 10 vezes mais servidores, 50 vezes mais dados, 75 vezes mais arquivos, com apenas 1,5 vezes mais pessoas.

Podemos perceber ao longo da história uma evolução no valor dos dados. Durante a década de 50 e 60 os dados eram vistos como produto. Nas duas décadas seguintes, os dados eram vistos como subproduto. Já nas décadas de 90 e 2000 os dados já começaram a ter um valor, auxiliando as organizações no processo de tomada de decisão. E na presente década, em diante, os dados são tratados como substrato, ou seja, algo essencial para o dia-a-dia das organizações.

Dados estão sendo coletados em grande escala. Eles são provenientes das mais diversas fontes, como: dados gerados por sistemas transacionais, sensores, câmeras, satélites, logs, redes sociais, etc.

Diante de toda essa inundação e evolução no valor dos dados, surge uma questão: O que podemos fazer com toda essa quantidade de informação? No mundo corporativo, decisões que antes eram baseadas em suposições, ou em modelos construídos por especialistas, agora podem ser feita com base nos dados coletados.

# Introdução

O conceito de Big Data não surge do nada. Na verdade ele não é novo, mas está recebendo uma grande atenção por algumas razões, como o barateamento do armazenamento de dados, a proliferação de sensores e tecnologia de captura de dados.

Não existe uma definição precisa sobre o termo Big Data, existindo várias definições na literatura. O IDC define da seguinte maneira: “*As tecnologias Big Data descrevem uma nova geração de tecnologias e arquiteturas, projetadas economicamente para extrair valor de volumes muito grandes de uma larga variedade de dados, permitindo alta velocidade de captura, descoberta e análise*”.

Outra definição, feita pelo *McKinsey Global Institute*: “*Big Data refere-se aos conjuntos de dados cujo tamanho está além da capacidade de ferramentas típicas de software de banco de dados para capturar, armazenar, gerenciar e analisar*”.

Percebemos, então, que não se define Big Data apenas em função do tamanho do volume de dados, e sim na capacidade de manipulá-los.

O Big Data pode ser resumido em 5 características, mais conhecidas como os 5 Vs.

* **Volume** – Grandes volumes de dados são produzidos e são coletados. Um exemplo desse aspecto são os experimentos científicos do CERN, que geram cerca de 40 TB de dados por segundo.
* **Velocidade** – Rapidez com que os dados são produzidos e precisam ser analisados. Muitas aplicações necessitam de resposta em tempo real, como detecções em fraudes, recomendações baseadas em redes sociais.
* **Variedade** – Existe uma grande variedade de dados, proveniente de várias fontes, podendo ser estruturados ou não. Alguns exemplos são dados gerados por sistemas transacionais, sensores, câmeras, satélites, logs, redes sociais, etc.

# Oportunidades

Sistema Gerenciador de Banco de Dados é um conjunto de [softwares](http://pt.wikipedia.org/wiki/Software) responsáveis pelo gerenciamento de uma [base de dados](http://pt.wikipedia.org/wiki/Banco_de_dados). Ele armazena, modifica e extrai informações de um banco de dados. Tem como funções a integridade semântica e a segurança de acesso. O principal objetivo é que a aplicação cliente não tenha a responsabilidade do gerenciamento de acesso, da manipulação e da organização dos dados. Ele possui uma interface onde os usuários podem incluir, alterar ou consultar dados, sendo os comandos executados na linguagem [SQL](http://pt.wikipedia.org/wiki/SQL) [1].

Qualquer empresa, independente de seu porte, possui dados e informações que precisam ser armazenados. E com o aumento acelerado dessas informações, tornou-se fundamental que esses dados fossem guardados em uma base de dados informatizada. Isso fez com que o acesso as informações fosse mais ágil, prático e com menos falhas na recuperação dos dados [1].

Existem tipos diferentes de SGBD, desde aqueles sistemas de grande porte que estão presentes em grandes corporações até os sistemas pequenos que funcionam em computadores pessoais. Os SGBDS mais conhecidos são: PostgreSQL, MySQL, Oracle e o SQL Server [1].

# Desafios

XML (*eXtensible Markup Language*) é uma linguagem de marcação que foi desenvolvida pela [W3C](http://pt.wikipedia.org/wiki/W3C), a partir da [SGML](http://pt.wikipedia.org/wiki/SGML) (*Standard Generalized Markup Language*) [2]. Criada com a finalidade para descrever diversos tipos de dados, para incorporar poder de representação de informações semi-estruturadas, não se preocupando com a estrutura de apresentação. Seu principal objetivo é a facilidade de compartilhamento de informações através da [internet](http://pt.wikipedia.org/wiki/Internet) [3].

XML possui uma característica importante que é criar uma estrutura exclusiva para várias outras linguagens. “Ela combina a flexibilidade da [SGML](http://pt.wikipedia.org/wiki/SGML) com a simplicidade da [HTML](http://pt.wikipedia.org/wiki/HTML)” [3]. O intuito do projeto era criar uma linguagem que pudesse ser lida por software, e pudesse se integrar com as outras linguagens [4].

A simplicidade e legibilidade estão presentes porque a linguagem tem como fundamento ser entendida tanto por humanos quanto por computadores. Além do mais, ela não se preocupa apenas com a estrutura dos dados, mas também com o seu significado. Isso é devido à liberdade que o usuário tem de poder criar tags sem limites [5].

Por causa de sua portabilidade, pois ela não depende de plataformas, seja de software ou hardware, bancos de dados distintos podem através de alguma aplicação se comunicar usando o XML [3].

# SGBD compatível com XML

Documentos XML são um conjunto de dados, assim como os bancos de dados, porém diferem em suas estruturas, na forma do armazenamento. XML possui os dados muito irregulares e pouco estruturados. Outra diferença é que XML não possui gerenciamento de dados, como os bancos de dados que possuem gerenciamento de transações, controle de integridade ou indexação [4].

Com essas divergências, especialistas da área foram divididos em duas partes: aqueles que queriam estender os SGBDs já existentes; e aqueles que queriam criar uma nova solução, os SGBDs XML nativos [4]. Com isso, surgiram três formatos de SBGDs que são apresentados posteriormente.

Antes disso, vale ressaltar que esses SGBDs se baseiam em dois tipos de documentos XML definido em relação sa organização de seus dados: os documentos XML DaCD – *Data-Centric Documents*, são documentos fracamente semi-estruturados que não possuem uma estrutura muito irregular. Esses documentos são utilizados em transferências de dados em aplicações na web, como o envio de formulários; e documentos XML DoCD – *Document-Centric Document*s, são documentos fortemente semi-estruturados que possuem uma estrutura irregular e particular. Eles são usados para descrever a linguagem natural [4].

Os documentos DaCD são mais fáceis de serem mapeados para os SGBDs, por isso são usados pelos SGBDs Relacionais e Objeto-Relacionais que dão suporte a XML. Em contrapartida, os documentos DoCD possuem uma estrutura mais complexa, e por essa razão são mais difíceis de serem mapeados para banco de dados. Nesse caso, para a persistência desses documentos, foram criados os SGBDs XML Nativos [4].

**Tipos de SGBD**

* SGBDs XML habilitados buscam usar os consolidados SGBDs Objeto-Relacionais existentes. Este recurso tem sido muito usado para o armazenamento de dados XML em que o objetivo da aplicação são os dados e não a estrutura do documento. Geralmente, um middleware é empregado para fazer a relação entre os dados XML e o SGBD, para que a estrutura do SBGD não seja alterada. Essa ferramenta é responsável por mapear o documento XML para uma tabela do banco de dados. Quando o administrador do banco utiliza a linguagem SQL para consultas [5].
* SGBDs XML habilitados com suporte a XML faz a persistência dos documentos XML na base de dados, além de armazenar partes da estrutura do documento, e por isso, ao serem novamente construídos, eles não ficam diferentes do documento original e as buscas aos dados podem ser feitas por meio de linguagens, como XPath, porque obedecem a estrutura do documento XML [4].
* “SGBDs XML nativo definem um modelo lógico específico para documentos XML” [4]. O modelo do documento possui a definição dos elementos, atributos e a ordem dos dados. Os SGBDs armazenam e recuperam os dados de acordo com o modelo definido. Suas aplicações suportam documentos DoCD, em razão de serem fortemente semi-estruturados. Os SGBDs XML nativos manipulam conjuntos de documentos como semânticas parecidas sem a necessidade de esquemas, por isso, podem armazenar qualquer documento [4].

# SQL Server

O SQL Server é um SGBD desenvolvido pela [Microsoft](http://pt.wikipedia.org/wiki/Microsoft) e sua versão atual é o SQL Server 2012. Conhecido por ser “uma plataforma de dados confiável, produtiva e inteligente” que permite executar “aplicações críticas, reduzir o tempo e o custo com o desenvolvimento e o gerenciamento das aplicações” [6]. Ele gerencia banco de dados para “soluções de comércio eletrônico, linha de negócios e data warehouse” [6].

A partir de 2000, devido a necessidade de novas aplicações, o SQL Server ofereceu suporte à plataforma XML. E em 2005 foi introduzido o tipo de dado XML que permite armazenar documentos ou partes de dados XML no banco de dados [6].

O tipo de dado XML é um tipo interno assim como o tipo *int* ou *varchar*. Ele pode ser usado como um tipo de variável, parâmetro, tipo de retorno de função ou tipo de coluna. Com esse tipo de dado, os valores XML podem ser guardados em uma coluna de tipo XML e ser classificado em relação aos esquemas XML. Quando o tipo de dado XML possui esquema, ele é classificado como XML com tipo, quando não tem, é dito, XML sem tipo [6].

Apesar do tipo de dado XML possuir representação de caracteres, ele não é uma cadeia de caracteres, mas permite conversão de tipos.

Uma coluna de tipo XML pode ser criada em uma tabela que possua outras colunas de dados relacionais ou pode ser criada em uma tabela separada com uma chave estrangeira referenciando a tabela principal.

A coluna do tipo XML pode ser indexada. A indexação melhora o desempenho das consultas aos dados XML. Quando em uma aplicação é comum consultas a dados XML e esses dados que são retornados são pequenos, usar índices pode ser uma boa alternativa para otimizar a buscar das consultas, porque a indexação evita que todos os dados sejam analisados em tempo de execução. Porém, a manutenção dos índices deve ser considerada durante a modificação dos dados [6].

Instâncias de tipos XML são armazenados como BLOBs, objetos binários grandes, e como eles podem ser muito grandes, sem um índice, essas instâncias são quebradas em tempo de execução para que a consulta seja analisada. Isso pode ser muito custoso, por isso, a indexação de colunas é uma boa solução para minimizar o tempo das consultas.

O SQL Server disponibiliza maneiras para que consultas aos dados XML sejam retornados em forma de tabela e também consultas aos dados relacionais sejam devolvidos como dados XML, são respectivamente: A função OPENXML e a cláusula FOR XML [6].

A cláusula FOR XML que recupera os dados relacionais em formato XML é usado junto com a consulta SELECT, retornando um conjunto de linhas. Também pode ser usada em subconsultas com as instruções INSERT, UPDATE e DELETE [6].

Para determinar a forma como o XML é retornado, a cláusula FOR XML possui quatro modos: RAW, AUTO, EXPLICIT, PATH.

O modo RAW retorna em cada linha um único elemento. Para gerar XML com hierarquia, é preciso que consultas FOR XML estejam aninhadas.

O modo AUTO retorna o XML aninhado usando heurísticas em relação a consulta especificada, porém o controle do usuário é pequeno sobre o formato do XML.

O modo EXPLICIT, diferente do modo AUTO, permitindo ao usuário um maior controle sobre a forma do XML gerado. É consentido usar atributos e elementos como quiser.

E por fim, o modo PATH, usado em subconsultas fornece a flexibilidade do modo EXPLICIT, mas com simplicidade [6].

A função OPENXML devolve um conjunto de linhas de um documento XML semelhante a uma tabela. Esse retorno pode ser armazenado em tabelas no banco de dados. Ele pode ser utilizado em instruções SELECT ou SELECT INTO.

A consulta a um documento XML é feita primeiramente, com o procedimento “sp\_xml\_preparedocument”, que analisa o documento e retorna uma representação da árvore DOM (*Document Object Model*). Depois o OPENXML exibi o conjunto de linhas do documento que foram passados como parâmetros para ele e armazena no banco de dados [6]. A figura 1 ilustra o procedimento do OPENXML.

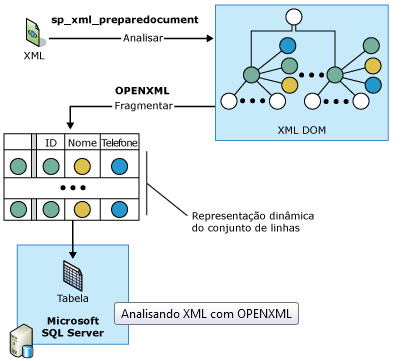


Figura 1. Procedimento da função OPENXML.

# Conclusão

A linguagem XML é absolutamente bem sucedida e atrelada as funcionalidades dos bancos de dados tornou-se ainda mais forte em sua aplicabilidade. Ela oferece uma maneira eficiente e fácil de compartilhar e modificar dados em diferentes aplicações.

Por meio de documentos XML, bancos de dados, como o SQL Server, podem se comunicar, transferindo dados em massa, por XML ser uma linguagem flexível e também pelo fato de que documentos XML são autodescritivos, facilitando a leitura e modificação dos arquivos.

# Bibliografia

[1] GANTZ, John; REINSEL, David. Extracting value from chaos. , 2011. Disponível em: <http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf>. Acesso em: 13 nov. 2012.

[2] IBM. What is big data?. , 1. Disponível em: <http://www-01.ibm.com/software/data/bigdata/>. Acesso em: 13 nov. 2012.

[3] <http://blog.kissmetrics.com/facebook-statistics/>. Acesso em 13 de nov. 2012.

[4] < http://www.economist.com/node/15557443 >. Acesso em 13 de nov. 2012.

[5] The McKinsey Global Institute, *Big data: The next frontier for innovation, competition, and productivity* - 2011 Disponível em: < www.mckinsey.com/~/media/McKinsey/dotcom/Insights%20and%20pubs/MGI/Research/Technology%20and%20Innovation/Big%20Data/MGI\_big\_data\_full\_report.ashx >. Acesso em 13 de nov. 2012.

[6] SQL Server. Disponível em: <[http://msdn.microsoft.com](http://msdn.microsoft.com/)>. Acesso em: 19 de out. 2012.