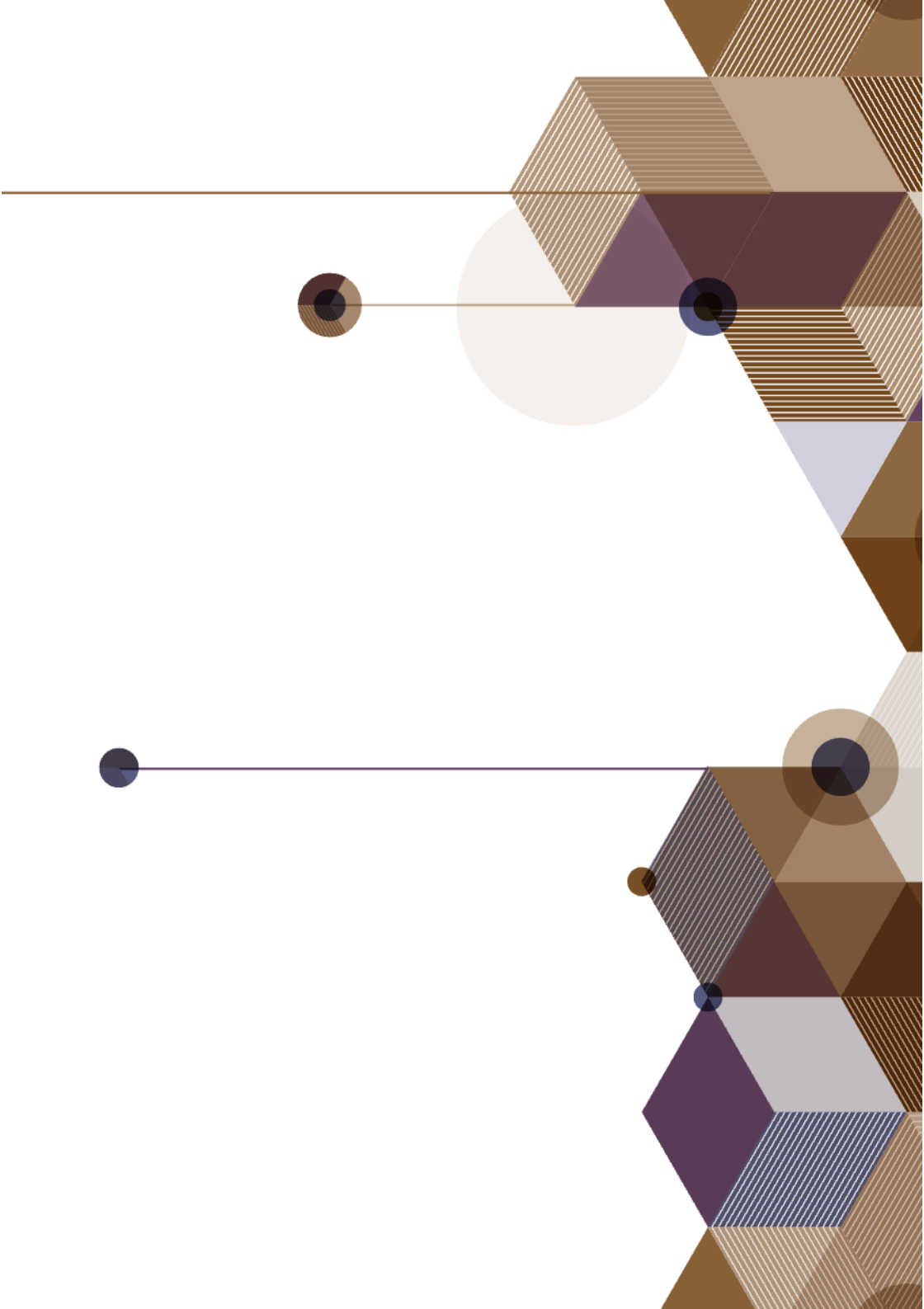





ADMINISTRAÇÃO DE BANCO DE DADOS

Jorge Brandão
de Campos





ANALISAR E COMPREENDER A UTILIZAÇÃO DO BANCO DE DADOS NoSQL

OBJETIVOS DE APRENDIZAGEM

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever o que são bancos de dados NoSQL.
- Listar os principais tipos e características dos bancos de dados NoSQL.
- Diferenciar o modelo relacional (SQL) e o NoSQL.

INTRODUÇÃO

O volume de dados está aumentando em proporções exponenciais no mundo. Os Estados Unidos é o país em que mais circulam dados, ficando a China em segundo lugar. Também merece destaque em tráfego de dados a América Latina, o Oriente Médio e a África.

E como lidar com tanta informação? Os bancos de dados de modelos relacionais estão preparados?

Neste texto, você vai estudar um novo conceito de banco de dados para trabalhar com grandes volumes de dados, o NoSQL.

BANCOS DE DADOS

Antes de falarmos em bancos de dados, devemos conhecer alguns conceitos importantes. O que são dados, informação e conhecimento?

Podemos conceituar dados como sendo a forma bruta da informação, ou a parte indivisível da informação. Vejamos um exemplo: abra um bloco de notas e insira seu nome, "Jorge". Neste instante, faça uma pergunta: o que significa "Jorge"? A resposta é simples, é apenas um dado, não tem significado. Perfeito! É realmente isso, o dado sozinho não tem significado. Portanto, podemos dizer que dado é uma parte da informação, ou seja, a forma mais bruta, sem qualquer semântica. Entretanto, temos a necessidade de entender o conceito de informação, com isso, faremos mais uma atividade: com seu bloco de notas aberto, após a palavra "Jorge", digite "aluno". Neste instante, temos um significado: sabemos que Jorge é um aluno. Portanto,

podemos afirmar que informações são significados que buscamos e isso acontece pelo processamento de dados (Figura 1), em que o conjunto deles gera a informação.

Figura nº 1 – Processamento dos dados

Um retângulo laranja contendo o texto "Jorge + aluno = Jorge é um aluno".

Jorge + aluno = Jorge é um aluno

Fonte: Autor

Outro conceito importante é o conhecimento. O conhecimento é obtido com a extração, organização e cruzamento de informações, ou seja, o conjunto de informações organizadas gera conhecimentos.

Um dos maiores ativos das empresas é o que chamamos de ativo intangível. Podemos colocar nesse grupo o conhecimento na fabricação de seus produtos e execução de serviços; e, em uma visão analítica, o conhecimento em todos os processos que levam a empresa a alcançar seus objetivos. Traçando uma linha cronológica decrescente (Figura 2), temos que o conhecimento vem de informações, ao passo que informações são extraídas do processamento de dados.

Figura nº 2 – Linha do tempo, conhecimento, informação e dado.

Um diagrama de linha do tempo dentro de um retângulo laranja. No topo, há uma linha vertical. Abaixo dela, há uma caixa retangular com o texto "Conhecimento < informação < Dados".

Conhecimento < informação < Dados

Fonte: Autor

Agora podemos entender por que desde os primeiros computadores, uma das grandes preocupações é quanto ao armazenamento e manipulação de dados. Com o passar dos anos, o volume de dados foi aumentando e surgindo modelos para organizar e manipular esses dados. A seguir, veremos uma breve descrição sobre os modelos de dados trabalhados no decorrer dos anos.

MODELOS DE DADOS

Modelo hierárquico

O modelo hierárquico é um tipo de gerenciador de banco de dados que conecta os registros em forma de árvore e cada tipo de registro tem apenas uma raiz.

Modelo em rede

O modelo em rede é uma extensão do modelo hierárquico. A representação dos dados é feita por meio de uma coleção de registros. Os relacionamentos são feitos por links.

Modelo relacional

O modelo relacional é o modelo mais aceito pelas empresas nos dias atuais. Sua característica é a organização dos dados em tabelas. As tabelas se relacionam entre si por meio de chaves primárias e estrangeiras.

Modelo orientado a objetos

Neste modelo os dados são armazenados na forma de objetos, por meio de estruturas chamadas classes. Os campos são instâncias dessas classes.

SGBD

Sistema Gerenciador de Banco de Dados ou SGBD é um conjunto de ferramentas baseados em uma linguagem, usadas para gerir a criação da estrutura, exclusão e alteração de bases de dados, tabelas e manipulação de dados.

BIG DATA

A cada dia é gerada uma grande quantidade e variedade de dados. Podemos citar como algumas das empresas que convivem diariamente com esse dilúvio de informações as companhias aéreas, operadoras de telefonia, redes sociais, busca on-line e redes varejistas. No entanto, não basta ter as informações armazenadas, é necessário gerar conhecimento para nortear processos e

tomadas de decisões, ou seja, saber fazer o melhor uso dos dados. Com isso, entramos no conceito de Big Data.

De acordo com Alecrim (2013), inicialmente podemos definir o conceito de Big Data como sendo conjuntos de dados extremamente amplos e que, por este motivo, necessitam de ferramentas especialmente preparadas para lidar com grandes volumes, de modo que toda e qualquer informação nestes meios possa ser encontrada, analisada e aproveitada em tempo hábil. Simplificando, a análise de grandes quantidades de dados para a geração de resultados importantes que, em volumes menores, dificilmente seriam alcançados.

Diariamente são feitas milhões de transações bancárias; milhões de e-mails são enviados ao redor do mundo; redes sociais com um volume cada vez maior de publicações. Portanto, podemos assumir que Big Data é um problema de um imenso volume e variedade de dados. Os bancos de dados relacionais, por características são normalizados, suas consultas possuem muitos joins pelo grande número de tabelas relacionadas, influenciando diretamente na performance da aplicação. Não estou apresentando o fim dos bancos de dados relacionais, apenas estou relacionando ao problema Big Data. Muitas empresas usam e continuarão usando suas bases relacionais. Buscamos, aqui, uma alternativa melhor para o trabalho com grandes volumes de dados. Os bancos de dados tradicionais, principalmente os relacionais, não possuem soluções adequadas a este problema. Ainda, segundo Alecrim (2013), isso acontece porque bancos de dados relacionais normalmente se baseiam em quatro propriedades que tornam a sua adoção segura e eficiente, razão pela qual soluções do tipo são tão populares: atomicidade, consistência, isolamento e durabilidade. Esta combinação é conhecida como ACID (do inglês, Atomicity, Consistency, Isolation e Durability). Neste ponto entra em cena o conceito de NoSQL.

NoSQL

NoSQL (não somente SQL) não faz referência a movimentos contra SQL, refere-se à solução de banco de dados que possibilita o armazenamento de várias formas não se limitando ao modelo relacional. Embora não seja um conceito novo, surgiu em 1998, ainda não se tornou bem conhecido dos profissionais de TI.

De acordo com Oliveira (2013):

“As análises em grandes massas de dados podem trazer respostas que antes não seriam possíveis de ser obtidas, por isso é grande o potencial de mercado do Big Data, bem como o desejo das empresas de adotá-lo.

Para tratar uma imensa quantidade de dados e aproveitá-los da melhor maneira possível, estão sendo criadas tecnologias que sustentam o Big Data como o NoSQL para infraestrutura de banco de dados, Stream Computing como novo paradigma e Hadoop e MapReduce voltado para análise de dados.

O NoSQL (*Not only Structured Query Language*) é um termo genérico para uma classe definida de bancos de dados não-relacionais, que tem uma propriedade chamada BASE (*Basically Available, Soft state, Eventual consistency*), que distribui os dados em diferentes repositórios tornando-os sempre disponíveis, não se preocupa com a consistência de uma transação, delegando essa função para a aplicação, porém sempre garante a consistência dos dados em algum momento futuro à transação.”

NoSQL é uma forma de organizar os dados diferente do modelo relacional. Até pouco tempo atrás só existiam bancos de dados relacionais; se você tivesse qualquer problema teria que buscar uma solução relacional.

Existe uma nova onda de aplicações de bancos de dados, a qual você não olha mais para o passado, você olha para o presente, ou seja, os dados estão chegando, estão sendo armazenados, processados, extraídos os conhecimentos para que possamos recomendar ao cliente.

Veja, na Tabela 1, o comparativo entre bancos de dados SQL e NoSQL.

Tabela nº 1 – Comparativo entre bancos de dados SQL e NoSql.

	Banco de dados SQL	Bancos de dados NoSQL
Modelo de dados	O modelo relacional normaliza dados em estruturas tabulares conhecidas como tabelas, que consistem em linhas e colunas. Um schema define estritamente as tabelas, colunas, índices, relações entre tabelas e outros elementos do banco de dados.	Bancos de dados não relacionais (NoSQL) normalmente não aplicam um schema. Geralmente, uma chave de partição é usada para recuperar valores, conjuntos de colunas ou documentos semiestruturados JSON, XML ou outros que contenham atributos de itens relacionados.

	Banco de dados SQL	Bancos de dados NoSQL
Propriedades ACID	<p>Sistemas de gerenciamento de bancos de dados relacionais (RDBMS) tradicionais são compatíveis com um conjunto de propriedades definido pela sigla ACID: Atomicidade, Consistência, Isolamento e Durabilidade. Atomicidade significa “tudo ou nada”, ou seja, uma transação é concluída integralmente ou não. Consistência significa que quando uma transação é realizada, os dados devem estar em conformidade com o schema do banco de dados. Isolamento exige que as transações simultâneas sejam executadas separadas uma da outra. Durabilidade é a capacidade de se recuperar de uma falha do sistema ou falta de energia inesperada para o último estado conhecido.</p>	<p>Bancos de dados NoSQL normalmente trocam algumas propriedades ACID de sistemas de gerenciamento de bancos de dados relacionais (RDBMS) por um modelo de dados mais flexível que escala horizontalmente. Essas características fazem dos bancos de dados NoSQL uma excelente opção em situações em que os RDBMS deparam com desafios de arquitetura e precisam solucionar uma combinação de gargalos de desempenho, escalabilidade, complexidade operacional e custos crescentes de administração e suporte.</p>
Desempenho	<p>O desempenho normalmente depende do subsistema do disco. A otimização de consultas, índices e estrutura de tabela é necessária para alcançar máximo desempenho.</p>	<p>Desempenho geralmente é uma função do tamanho do cluster do hardware subjacente, da latência de rede e da aplicação que faz a chamada.</p>
Escala	<p>Mais fácil de aumentar a escala “verticalmente” com hardware mais rápido. Outros investimentos são necessários para tabelas relacionais para abranger um sistema distribuído.</p>	<p>Projetado para aumentar a escala “horizontalmente” usando clusters distribuídos de hardware de baixo custo para aumentar a transferência sem aumentar a latência.</p>
APIs	<p>As solicitações para armazenar e recuperar dados são comunicadas usando consultas compatíveis com structured query language (SQL). Essas consultas são analisadas e executadas por sistemas de gerenciamento de bancos de dados relacionais (RDBMS).</p>	<p>APIs baseadas em objetos permitem que desenvolvedores de aplicações armazenem e restaurem facilmente estruturas de dados na memória. As chaves de partição permitem que os aplicativos procurem pares de chave-valor, conjuntos de colunas ou documentos semiestruturados contendo objetos e atributos de aplicativos serializados.</p>
Ferramentas	<p>Os bancos de dados SQL normalmente oferecem um rico conjunto de ferramentas para simplificar o desenvolvimento de aplicações orientadas ao banco de dados.</p>	<p>Oferecem ferramentas para gerenciar clusters e escalabilidade. As aplicações são a interface principal com os dados subjacentes.</p>

Fonte: <https://aws.amazon.com/pt/nosql/>

Principais características dos bancos de dados NoSQL

Os bancos de dados NoSQL possuem características importantes que os diferenciam dos bancos de dados relacionais. De acordo com Cavalcante (2012), as características dos bancos de dados relacionais são:

- **Escalabilidade Horizontal:** na medida em que o volume de dados cresce, aumenta-se a necessidade de escalabilidade e melhoria do desempenho. Dentre todas as possibilidades para esta solução, a escalabilidade horizontal se torna a mais viável, porém requer diversas threads ou que processos de um tarefa sejam criadas e distribuídas. Dessa forma, o uso de um banco de dados relacional poderia ser muito complexo. Não queremos dizer que os bancos de dados relacionais não escalam, a verdade é que eles não escalam facilmente. Isto por que no momento em que diversos processos se conectam simultaneamente em um mesmo conjunto de dados há uma geração de uma alta concorrência aumentando assim o tempo de acesso às tabelas. Neste contexto, uma grande vantagem dos bancos NoSQL é justamente a ausência de bloqueios, o que permite a escalabilidade horizontal com uma maior facilidade e eficiência (ele não é afetado pelo aumento da concorrência). Uma alternativa muito utilizada para alcançar a escalabilidade horizontal é o Sharding, que divide os dados em múltiplas tabelas a serem armazenadas ao longo de diversos nós na rede. O que esta técnica faz, na realidade, é romper a cadeia de relacionamentos, que é uma forte característica nos bancos relacionais. É possível realizar o Sharding em banco de dados relacionais de forma manual. Entretanto, esta não é uma tarefa simples e demonstra complexidade de implementação para a equipe que está desenvolvendo.
- **Ausência de esquema (Schema-free) ou esquema flexível:** Outra característica notável em bancos de dados NoSQL é a ausência parcial ou total de esquema que define a estrutura de dados. É justamente essa ausência de esquema que facilita uma alta escalabilidade e alta disponibilidade, mas em contrapartida não há a garantia de integridade dos dados, fato este que não ocorre no Modelo Relacional.
- **Suporte nativo a replicação:** Esta é outra forma de prover a escalabilidade, pois, no momento em que permitimos a replicação de forma nativa o tempo gasto para recuperar informações é reduzido.
- **API simples para acessar o banco de dados:** Em banco de dados NoSQL, o foco não está no armazenamento dos dados e sim como recuperar estes dados de forma eficiente. Pensando nisso, é fundamental APIs

desenvolvidas para facilitar o acesso às devidas informações para que se possa usar o banco de dados de forma rápida e eficiente.

- **Consistência eventual:** Outra característica particular de bancos NoSQL é que nem sempre a consistência dos dados é mantida. Esta característica tem embasamento no teorema CAP (Consistency, Availability e Partition tolerance) que afirma que em um dado momento só é possível garantir duas destas três propriedades, que seriam consistência, disponibilidade e tolerância à partição. No mundo real, normalmente estas duas últimas são privilegiadas. Como consequência disto, as propriedades do ACID não são respeitadas simultaneamente, ao contrário disto, temos outro conjunto de projetos denominado BASE (Basicamente disponível, estado leve e consistente em momento indeterminado). Ou seja, é necessário haver um planejamento para que o sistema possa tolerar inconsistências temporárias com o objetivo de priorizar a disponibilidade.

Agora que falamos brevemente sobre as principais características nos bancos de dados NoSQL, é importante ressaltar algumas técnicas utilizadas para a implementação de suas funcionalidades. Entre elas estão:

- **Map/reduce:** permite a manipulação de enormes volumes de dados ao longo de nós em uma rede. Funciona da seguinte forma: na fase map, os problemas são particionados em pequenos problemas que são distribuídos em outros nós na rede. Quando chegam à fase reduce, esses pequenos problemas são resolvidos em cada nó filho e o resultado é passado para o pai, que sendo ele consequentemente filho, repassaria para o seu, até chegar à raiz do problema.

- **Consistent hashing:** suporta mecanismos de armazenamento e recuperação, onde a quantidade de sites está em constante mudança. É interessante usar essa técnica, pois ela evita que haja uma grande migração de dados entre estes sites, que podem ser alocados ou desalocados para a distribuição dos dados.

- **MVCC (Multiversion concurrency control):** Oferece suporte a transações paralelas em banco de dados. Por não fazer uso de locks para controle de concorrência, faz com que transações de escrita e leitura sejam feitas simultaneamente.

- **Vector clocks:** Ordenam eventos que ocorreram em um sistema. Como existe a possibilidade de várias operações estarem acontecendo simultaneamente, o uso de um log de operações informando suas datas se faz importante para informar qual versão de um dado é a mais atual.

Tipos de bancos de dados NoSQL

- Sistemas baseados em armazenamento chave valor
- Sistemas baseados em grafos
- Sistemas orientados a documentos
- Sistemas orientados a colunas

A Tabela 2 apresenta uma descrição dos tipos de bancos de dados NoSQL.

Tabela nº 2 – Comparativo entre bancos de dados SQL e NoSql.

Tipos de bancos de dados NoSQL	
Bancos de dados colunares	Os bancos de dados colunares são otimizados para colunas de leitura e gravação, ao contrário das linhas de dados. O armazenamento orientado a colunas para tabelas do banco de dados é um fator importante no desempenho de consulta analítica, pois ele reduz drasticamente os requisitos gerais de E/S e diminui a quantidade de dados que você precisa carregar do disco.
Bancos de dados de documentos	Os bancos de dados de documentos são projetados para armazenar dados como documentos, geralmente em formato JSON ou XML. Diferentemente dos bancos de dados relacionais tradicionais, o esquema de cada documento não relacional (NoSQL) pode variar, dando a você mais flexibilidade ao organizar e armazenar dados do aplicativo e ao reduzir o armazenamento exigido para valores opcionais.
Bancos de dados de gráficos	Os bancos de dados de gráficos armazenam vértices e links direcionados chamados de bordas. Gráficos podem ser construídos em bancos de dados relacionais (SQL) e não relacionais (NoSQL). Vértices e bordas podem ter propriedades associadas a eles.
Armazéns em memória de chave-valor	As solicitações para armazenar e recuperar dados são comunicadas usando Os armazéns em memória de chave-valor são bancos de dados NoSQL otimizados para cargas de trabalho de aplicativos de leitura pesada (como redes sociais, jogos, compartilhamento de mídia, e portais de P e R) ou cargas de trabalho com uso intenso da computação (como um mecanismo de recomendação). O armazenamento em cache na memória melhora o desempenho do aplicativo ao armazenar pedaços críticos de dados na memória para acesso de baixa latência.

Fonte: <https://aws.amazon.com/pt/nosql/>

NoSQL database exemplos:

- Google bigtable
- Amazon Dynamo
- Facebook Cassandra
- Apache HBASE
- LinkedIn Valdemort

REFERÊNCIAS

ALECRIM, Emerson. **O que é Big Data?** Infowester, 2013. Disponível em: <<https://www.infowester.com/big-data.php>>. Acesso em: 20 mar. 2017.

OLIVEIRA, Claudio Alves de. **Bancos de Dados NoSQL**. IBM, 2013. Disponível em: <https://www.ibm.com/developerworks/community/blogs/tlcbr/entry/bancos_de_dados_nosql?lang=en>. Acesso em: 20 mar. 2017.

AMAZON WEBSERVICE. **O que é NoSQL?** [S.l.]. Disponível em: <<https://aws.amazon.com/pt/nosql/>>. Acesso em: 20 mar. 2017.

NoSQL. [S.l.]. Disponível em: <<http://nosql-database.org/>>. Acesso em: 20 mar. 2017.

CAVALCANTE, Marília. **Banco de dados NoSQL**: Um novo paradigma - Revista SQL Magazine 102.Devmedia, a.9,n. 07, ago. 2012. Disponível em: <<http://www.devmedia.com.br/banco-de-dados-nosql-um-novo-paradigma-revista-sql-magazine-102/25918>>. Acesso em: 20 mar. 2017.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS