



BANCO DE DADOS RELACIONAL E BIG DATA



Cassio Rodolfo Aveiro da Silva
Clarissa Fernanda Correia Lima Loureiro

BANCO DE DADOS RELACIONAL E BIG DATA

1ª edição

Londrina
Editora e Distribuidora Educacional S.A.
2020

© 2020 por Editora e Distribuidora Educacional S.A.

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Editora e Distribuidora Educacional S.A.

Presidente

Rodrigo Galindo

Vice-Presidente de Pós-Graduação e Educação Continuada

Paulo de Tarso Pires de Moraes

Conselho Acadêmico

Carlos Roberto Pagani Junior
Camila Braga de Oliveira Higa
Carolina Yaly
Giani Vendramel de Oliveira
Henrique Salustiano Silva
Mariana Gerardi Mello
Nirse Ruscheinsky Breternitz
Priscila Pereira Silva
Tayra Carolina Nascimento Aleixo

Coordenador

Tayra Carolina Nascimento Aleixo

Revisor

Flavio Kaue Fiuza de Moura

Editorial

Alessandra Cristina Fahl
Beatriz Meloni Montefusco
Gilvânia Honório dos Santos
Mariana de Campos Barroso
Paola Andressa Machado Leal

Dados Internacionais de Catalogação na Publicação (CIP)

S586b Silva, Cassio Rodolfo Aveiro da
Banco de dados relacional e big data/ Cassio Rodolfo
educação profissional/ Cristiane Higuera Simó, Carolina
Aveiro da Silva, Clarissa Fernanda Correia Lima Loureiro –
Londrina: Editora e Distribuidora Educacional S.A. 2020.
42 p.

ISBN 978-65-5903-090-3

1. Banco de dados relacionais. 2. SQL (Linguagem de
programação de computador). 3. Big data. I. Título.

CDD 005

Raquel Torres – CRB: 6/2786

2020
Editora e Distribuidora Educacional S.A.
Avenida Paris, 675 – Parque Residencial João Piza
CEP: 86041-100 — Londrina — PR
e-mail: editora.educacional@kroton.com.br
Homepage: <http://www.kroton.com.br/>

SUMÁRIO

Banco de Dados e Big Data: União Necessária _____	05
ACID e Benefícios do Banco de Dados Relacional _____	19
A SQL e os Sistemas de Gestão de Banco de Dados _____	31
Big Data e bancos de dados não relacionais _____	47

Banco de Dados e Big Data: União Necessária

Autoria: Clarissa Loureiro

Leitura crítica: Flavio Fiuza



Objetivos

- Aprender sobre o que é um Banco de Dados Relacional, a estruturação do Modelo Relacional e a utilização dos bancos relacionais e blocos de código do banco relacional.
- Entender sobre as características dos bancos relacionais, simultaneidade e bloqueio, os benefícios dos bancos de dados relacionais, o futuro e banco de dados relacional e autômato.
- Iniciar a aprendizagem sobre Big Data e os projetos, os pilares dos tratamentos de dados e o armazenamento de informação.
- Aprender sobre os repositórios de dados e Big Data, Data Lake, a importância da nuvem e seus tipos em Big Data.



1. Banco de dados relacional

Nesta disciplina, abordaremos sobre um assunto importante para os dias atuais e para as profissões do futuro, os bancos de dados. Eles são uma ótima solução para um novo produto muito valioso e que está sendo chamado de novo petróleo, os dados.

Nesse sentido, detalharemos sobre o banco de dados relacional, que é um dos tipos de banco de dados que armazena como outros bancos, no entanto, a diferença está nos dados relacionados entre si. Esse tipo de banco é baseado no modelo relacional, ou seja, a representação dos dados ocorre por tabelas, em que cada linha é um registro com o seu endereço, que também pode ser chamado de *ID* ou de chave. Na tabela, as colunas têm as características, que são chamadas de atributos dos dados. Por sua vez, as relações entre os dados são a razão pela qual esse tipo de banco tem o seu nome, pois, para cada atributo haverá um valor registrado que o relaciona com os respectivos dados. Por exemplo, o usuário pode ter duas tabelas com uma coluna comum em ambas. Dessa forma, existe uma relação entre as tabelas em que o banco de dados relacional pode estabelecer uma *ID*, que, nesse caso, é uma chave.

Por causa da relação entre as tabelas e, conseqüentemente, a relação entre os dados das duas tabelas do exemplo apresentado, podemos ter um processamento de dados mais rápido e confiável. Assim, quando o usuário procurar uma informação, ele conseguirá chegar na solução de maneira mais eficiente.

1.1 Modelo relacional

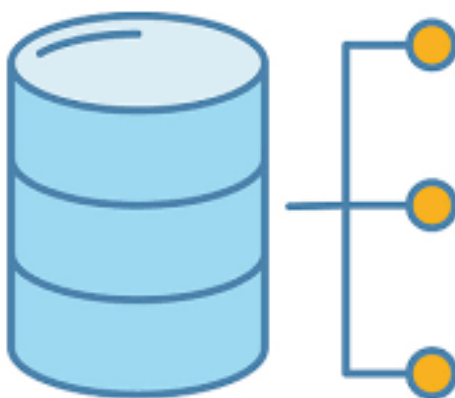
O ponto fundamental, em que você poderá diferenciar o tipo de banco de dados estudado, está no modelo desenvolvido. Quem conhece banco de dados desde os seus primeiros anos sabe que havia um problema: os desenvolvedores que organizavam os bancos de dados eram as únicas pessoas que sabiam onde poderiam encontrar as informações que alguém precisasse. Então, era necessário um banco de dados em que

vários desenvolvedores pudessem encontrar facilmente as informações que procuravam, ainda que não fossem eles que tivessem organizado os dados. Dessa forma, houve a necessidade de estruturar um modelo padrão de consultar os dados, a fim de que ele pudesse ser usado por qualquer profissional ou, até mesmo, aplicativos. Logo, entender a lógica desse modelo é a chave para o entendimento desta disciplina.

O modelo de banco de dados relacional tem a sua característica principal na utilização de tabelas para armazenar, de forma eficiente, e acessar as informações procuradas. Assim, um modelo mais eficiente de acesso aos dados foi criado.

Com o desenvolvimento de diferentes linguagens de programação, houve a ideia de utilizar a linguagem chamada *SQL*, em português, linguagem de consulta estruturada. Essa é uma boa opção de linguagem para criar e consultar informações em um banco. Já para quem precisa trabalhar com grande quantidade de dados, é importante ter familiaridade com essa linguagem, que é baseada na álgebra relacional. Nesse caso, o desempenho é melhor, por se tratar de uma linguagem matemática que é mais consistente internamente.

Figura 1 – Exemplificação do banco de dados relacional



Fonte: bsd555/iStock.com.

1.2 Utilização dos bancos de dados relacionais

Por ser eficiente e, de certa forma, simples, organizações, pequenas e grandes empresas utilizam o modelo relacional. Atualmente, existe uma grande necessidade de guardar informações e depois buscá-las da forma mais otimizada possível. Então, em uma gama ampla de busca de informações, como em cartórios, hospitais, delegacias e, até mesmo, na administração das cidades, exige uma rapidez e confiabilidade na busca dos dados. Logo, um modelo que relaciona tabelas por meio de colunas em comum e otimizar todo o processo foi uma forma sagaz e rápida de ter os clientes satisfeitos com a consistência das informações.

A data da existência desses bancos com esse modelo é de 1970. Até os dias atuais, esse é um dos modelos mais amplamente utilizados e aprovados para banco de dados.

1.3 Blocos de código no banco relacional

As consultas para obter determinadas informações podem envolver ações bem repetitivas. Algumas consultas simples, se necessário, podem ser feitas centenas ou milhares de vezes. Por isso, alguns códigos de acesso à bancos de dados para as funções de acesso à dados foram desenvolvidos. Nesse sentido, criar códigos novos a cada consulta seria impossível em termos de tempo, então, o que era necessário eram blocos de código que funcionassem para vários usuários ou, até mesmo, que fossem utilizados por aplicativos diferentes. A boa notícia é que os bancos de dados relacionais permitem essas ações. Dessa forma, vários aplicativos podem ter acesso, ao mesmo tempo, aos registros de forma eficiente a uma grande variedade de dados e bancos.

No entanto, é possível que existam conflitos quando vários usuários estão tentando alterar os mesmos dados simultaneamente. Por esse motivo, técnicas de bloqueio foram criadas para reduzir esses conflitos e manter a confiabilidade dos dados.

1.4 Bloqueio e simultaneidade

Como dito anteriormente, o bloqueio pode impedir que usuários e diferentes aplicativos possam alterar os dados ao mesmo tempo. Por exemplo, no *Edudrive*, que é bastante utilizado por órgãos governamentais no Brasil, é impossível que vários usuários alterem os dados ao mesmo tempo. O bloqueio pode ser aplicado a uma tabela inteira ou no nível de registro; se forem aplicados bloqueios no registro, isso significa que outros desses poderão ser utilizados dentro da mesma tabela.

Entretanto, também, pode acontecer algo diferente, que é o caso da simultaneidade, a qual gerencia a atividade quando existe uma utilização, como o próprio nome diz, simultânea no mesmo banco de dados. Nesse caso, quem define todas as regras de alterações pelos usuários ou aplicativos é gestor do banco. Dessa forma, o desempenho e controle dos dados é dado por meio de prioridades e políticas que são desenvolvidas de acordo com as necessidades de cada empresa.

Figura 2 – Representação de banco de dados relacional e segurança



Fonte: Bedrin-Alexander/iStock.com.

1.5 Próximos passos do banco relacional – o futuro

A área de programação, desenvolvimento de linguagem e a própria tecnologia, de forma geral, muda muito rápido. Por esse motivo, é preciso estar sempre atualizado sobre novidades e tendências futurísticas do que se está trabalhando. Portanto, saber qual o próximo passo para o modelo de dados relacional é importante para entender qual será a direção da mudança e preparar as empresas para o futuro.

1.6 Bancos relacionais e autônomos

Em um mundo conectado, imagine a quantidade de dados que seu smartphone e diversos aparelhos controlam em sua casa por meio da internet das coisas. O volume e complexidade de dados aumenta em uma velocidade grande demais para o armazenamento de tantas informações, assim, é possível entender que administrar uma quantidade de dados muito grande tem se tornado um desafio extremamente complexo. Então, a gerência e o desempenho dos bancos de dados precisam de otimização.

O tempo é precioso quando se busca informações. Nesse sentido, uma das saídas almejadas é obter processos que sejam menos dependentes de pessoas e que sejam mais rápidos, como alguns robôs. Dessa forma, surgiu a ideia do desenvolvimento do banco de dados autônomo, e esse seria o passo seguinte do banco de dados relacional. A independência e automação foram combinadas para melhorar e tornar o processo de busca de dados rápido e eficiente. Por sua vez, a inteligência artificial e o aprendizado de máquina são utilizados no banco de dados autônomo, a consequência dessa característica é que o banco melhora de forma contínua os processos de busca para tornar o desempenho otimizado. Desse modo, o homem fica menos dependente de monitorar e interferir em cada aplicação. Portanto, o tempo é gasto com outros desenvolvimentos de produtos para as empresas e clientes.



2. Big Data

Uma grande quantidade de informação precisa ser armazenada de forma segura e seus dados devem ser acessados de forma eficiente. Big Data, expressão em inglês que, ao pé da letra, significa grandes dados, é a área do conhecimento responsável pela análise e tratamento de uma quantidade muito grande de dados.

Um volume muito grande de dados é gerado todos os dias e a cada minuto, atualmente, são gerados mais dados do que em anos e décadas passadas. Nesse sentido, saber sobre como analisar os dados e armazená-los de forma eficiente é o grande desafio e necessidade dos dias atuais. Por isso, a junção das disciplinas banco de dados relacional e big data trazem uma fundamental urgência das profissões do presente e do futuro, que são: o cientista, o analista e o engenheiro de dados.


Portanto, conhecer as ferramentas necessárias para manipular, analisar, e tratar os dados pode estar entre as maiores revoluções da atualidade. Atualmente, fala-se em mineração de dados, em que se pode também encontrar padrões dentro de imensos conjuntos de dados.

O processamento e a modelagem dos dados são divididos em três pilares: tecnologias de Big Data, ferramentas de Analytics e ferramentas de BI.

2.1 Início de projeto de Big Data

O projeto de Big Data se inicia na determinação do objetivo e nas reais necessidades da pesquisa ou produto. Assim, como em todo bom projeto, é necessário estabelecer:

1. Quais tipos de dados serão guardados, armazenados.
2. Onde será feita a busca e a aquisição dos dados.

- 
3. Deve-se desenvolver um lago de dados, chamado *Data Lake*.
 4. Estabelecer as formas de capturas de dados, como as APIs (*Application Programming Interface*).
 5. Pesquisar quais fontes disponibilizam as APIs.
 6. Analisar os custos e o tempo de operação.

Com o objetivo de processar o volume de dados do Big Data são necessárias tecnologias que consigam disponibilizar escalabilidade, disponibilidade e flexibilidade.

2.2 Pilares do tratamento de dados

Para processar e modelar os dados, existem alguns pilares divididos em:

- Tecnologias de grandes dados.
- Ferramentas de inteligência de negócios.
- Ferramentas de análises.

Esses três pilares são essenciais para quem trabalha com análise e tratamento de dados. Ao longo da disciplina, passaremos por esses pontos e detalharemos as tecnologias e ferramentas utilizadas nas aplicações.

Quando se trabalha para efetivar e executar um projeto, a organização de algumas etapas é necessária, como: analisar e interpretar os dados para obtenção dos resultados que se deseja, a fim para que a ferramenta desenvolvida transforme os dados em informação útil e valiosa, gerando conhecimento e produto para os clientes. Um projeto bem executado faz toda a diferença na aplicação. Nesse sentido, governos de vários países têm projetos com tratamento de dados, eles trabalham com grandes painéis e centrais de informações que unem e mapeiam um grande volume de informação para um mesmo lugar, com

diversos profissionais de várias áreas atentos. Dentre os profissionais envolvidos, a área de análise de dados torna-se uma das mais importantes para aplicações em segurança, saúde, eficiência de serviços públicos de qualidade, como saneamento e, até mesmo, na própria educação da população nas cidades. Por isso, é tão importante ter bem estabelecido o tratamento dos dados, seus pilares e suas etapas.


Figura 3 – Big Data e suas várias aplicações



Fonte: Devrimb/iStock.com.

2.3 Ambiente e armazenamento de informação

Depois de começar o projeto, subentende-se que as perguntas principais já foram feitas e respondidas, como custo de implantação, valor agregado a empresa e ao negócio, retorno financeiro, valor de mercado, entre outras perguntas que responderão se o projeto e sua implantação são viáveis em um tempo de execução esperado. Tendo o objetivo a ser atingido e suas metas estabelecidas dentro do projeto e as perguntas



principais respondidas, o alcance para o sucesso do projeto de Big Data fica mais próximo.

Nesse momento, alguns fatores devem ser pensados e planejados dentro do projeto com relação a:

- Escalabilidade.
- Alta disponibilidade.
- Flexibilidade.

A escalabilidade diz respeito a quantidade de usuários que acessam o banco de dados, pois, no mundo atual, a rapidez em que isso pode acontecer é enorme, tanto para o crescimento quanto para a diminuição do número de acessos dos usuários. Portanto, são necessários a projeção e o planejamento para casos de picos de acessos ou o decréscimo deles.

Por sua vez, a alta disponibilidade é necessária para garantir o acesso aos usuários sempre que eles precisarem. Nesse sentido, a informação precisa estar disponível em tempo real, independentemente do horário e localização do cliente. Alguns serviços, como redes sociais, bancos e bibliotecas virtuais, precisam estar com sistemas sempre disponíveis.

A flexibilidade trata da maneira em que os dados são armazenados. Por que é importante saber sobre isso? Para que os dados possam ser acessados por diversas tecnologias, aplicativos e clientes é necessário que o armazenamento garanta uma flexibilização e possa armazenar dados estruturados e não estruturados, garantindo a utilização de vários tipos de tecnologias diferentes.

2.4 Repositório de Big Data

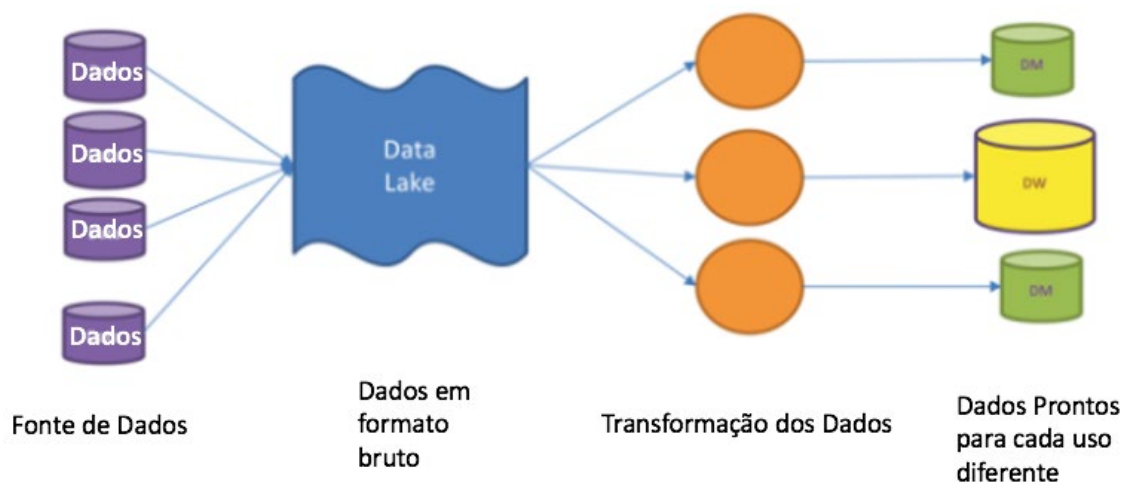
A estrutura do banco de dados, seu custo, gerenciamento e segurança da informação são aspectos fundamentais para o armazenamento de

um volume muito grande de dados. Por este motivo, é importante se pensar no tipo de repositório onde os dados serão armazenados.

Normalmente, para que as informações tenham uma alta disponibilidade, é ideal criar um único repositório. Hoje em dia está sendo usado o que se chama de lago de dados, na tradução ao nome em inglês, *Data Lake*.

Nesse *Data Lake*, os dados são armazenados da mesma forma em que houve a aquisição, de maneira bruta. Esse tipo de repositório pode ser criado na nuvem, em empresas que trabalham com serviços de nuvem, como Microsoft, Amazon, Apple, que são grandes empresas que trabalham com esses tipos de serviços. A utilização de soluções e serviços de software para *Big Data* estão crescendo em uma velocidade muito grande. Logo, toda a área de conhecimento relacionado a nuvem precisa ser bem entendida e relacionada as áreas de banco de dados e *Big Data*. No processo de criação do *Data Lake*, é importante que todos os membros da equipe, desde desenvolvedores até os diretores, participem e entendam como estão sendo organizados e armazenados os dados, para que seja eficiente a busca e resgate dos dados quando necessário.

Figura 4 – Representação didática e simples de *Data Lake*



Fonte: adaptada de <https://medium.com/data-ops/the-data-lake-is-a-design-pattern-888323323c66>. Acesso em: 11 jan. 2021.

2.5 Nuvem em Big Data

Há serviços de nuvem que podem ser privados, públicos ou híbridos. A escolha de umas das três opções ocorre de acordo com o cliente, a empresa e o que ela prioriza em seus produtos.

Quando utilizada para uso público, a nuvem é como o nome diz, pública. Por exemplo, se uma empresa tem todo o domínio da infraestrutura dos dados e vende os seus serviços, ela precisa de uma nuvem pública, como é o caso da Amazon e da Microsoft, que tem o seu produto chamado Azure. No caso da empresa que deseja ter uma nuvem, também chamada *cloud*, em inglês, que quer oferecer ao público produtos em épocas de grandes vendas, como: natal, promoções, vendas de produtos em grandes feiras, a nuvem pública, pode ter a quantidade de servidores que precisar e desejar. Por sua vez, o pagamento na nuvem pública é feito a medida da utilização dos serviços.

A nuvem privada é para o uso exclusivo de uma determinada empresa. Quando a confidencialidade dos dados e sua segurança é a maior prioridade, e não existe, por exemplo, serviços de venda para o público em geral, as empresas optam por ter uma nuvem privada, *cloud* privada, como é chamada popularmente entre os desenvolvedores.

O armazenamento feito em nuvem híbrida é a combinação das características das nuvens públicas e privadas, chamado em inglês, de *Hybrid Cloud*. Os termos nas áreas tecnológicas são muito utilizados em inglês, pois, várias linguagens de programação têm seus comandos parecidos com a língua inglesa. Então, ao longo da disciplina, será comum utilizarmos alguns termos em inglês, para que você se familiarize com a linguagem utilizada no Brasil e no mundo nos dias atuais.

Figura 5 – Diversos serviços do dia a dia, e dados são armazenados na nuvem



Fonte: Liubov Khutter-Kukkonin/iStock.com.

Na área de banco de dados, armazenamento e tratamento, de grande volume de dados é muito conhecida a área de computação nas nuvens. Pela definição de *Cloud Computing*, termo utilizado em inglês, de acordo com o NIST (*National Institute of Standards and Technology*):

Cloud computing é um modelo que permite um acesso sob demanda via redes de computadores a um conjunto compartilhado de recursos computacionais que podem ser rapidamente provisionados e liberados com um mínimo de esforço administrativo ou interação com o provedor de serviços. (NIST, 2011, p. 2, tradução nossa)

A temática abordada nesta Leitura Digital trouxe os assuntos mais atuais e procurados no mundo todo nas áreas tecnológicas. Os dados são o novo petróleo e, portanto, é de fundamental importância o entendimento sobre banco de dados relacionais, tratamento, armazenamento e eficiência das informações que estão hoje em dia com um volume tão grande, que o termo Big Data está entre as maiores empresas do mundo.



Referências Bibliográficas

AMARAL, F. **Introdução à ciência de dados, mineração de dados e Big Data**. Rio de Janeiro: Alta Books, 2016.

GRUS, J. **Data Science do zero**: primeiras regras com Python. Rio de Janeiro: Alta Books, 2016.

HEUSER, C. A. **Projeto de bancos de dados**. Porto Alegre: Bookman, 2009.

NIST. National Institute of Standards and Technology. **The NIST definition of cloud computing**. Gaithersburg: NIST, 2011.

NOGARE, D. **Do banco de dados relacional à tomada de decisão**. São Paulo: Mvtech Livro, 2014.

ACID e Benefícios do Banco de Dados Relacional



Autoria: Clarissa Loureiro

Leitura crítica: Flavio Fiuza



Objetivos

- Aprender sobre ACID (atomicidade, consistência, isolamento e durabilidade).
- Entender um pouco mais sobre cardinalidade.
- Aprender sobre confiabilidade e benefícios do banco de dados relacional.
- Refletir sobre aplicações e o futuro do banco de dados relacional.



1. ACID

O Banco de Dados Relacional é considerado uma das formas mais seguras e eficientes de guardar os dados. Para entender esses conceitos e o porquê de o Banco de Dados ter todas essas qualidades, é importante entender diversas propriedades e compromissos que o Banco de Dados possui.

A garantia de qualidade e eficiência do Banco de Dados é dada quando temos uma série de propriedades para estabelecer antes de escolher qual a forma de aquisição, manutenção e durabilidade dos dados. Além disso, é necessário saber se a aplicação dos dados depende de uma grande segurança dos dados. Então, nesta leitura serão explicitados diversos conceitos importantes sobre o tipo de banco de dados em que se quer trabalhar para cada aplicação ou empresa.

Um conceito bem conhecido sobre Banco de Dados e algumas propriedades é chamado pela seguinte sigla: ACID, que reúne as iniciais dos conceitos de Atomicidade, Consistência, Isolamento e Durabilidade.

Nesse sentido, é comum usar algumas palavras quando tratamos de banco de dados, como transações, que se refere a uma certa sequência de operações. Com essa definição, podemos falar mais detalhadamente sobre o conceito das propriedades ACID que são relativas às transações nos bancos de dados, como:

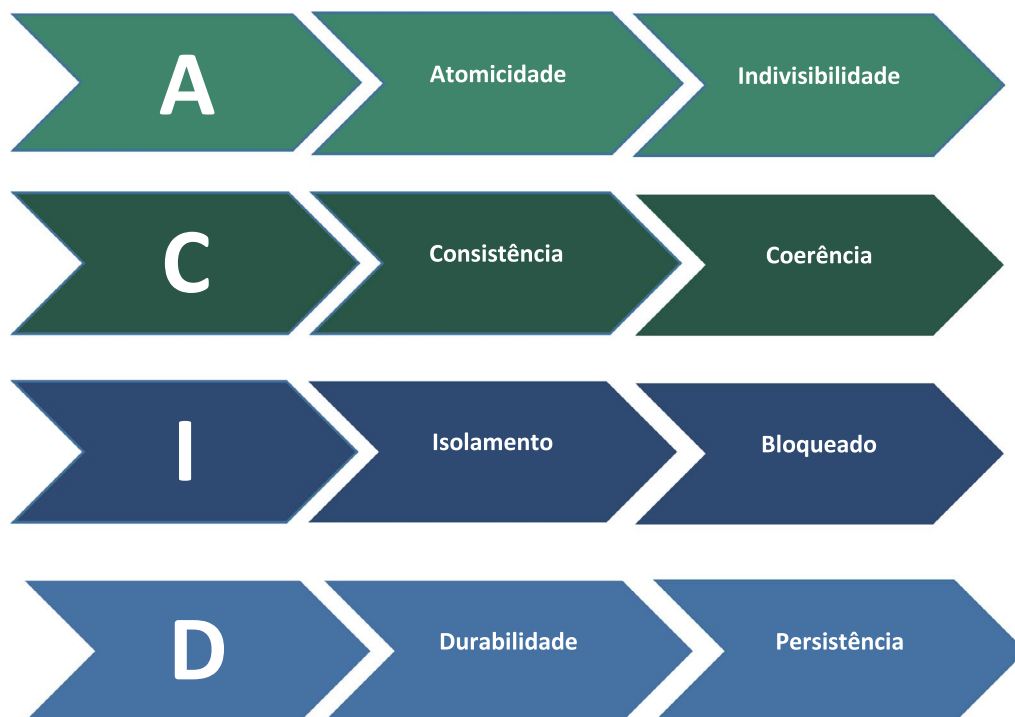
- **Atomicidade:** é o conceito que dá a ideia de indivisibilidade, ou seja, as transações que acontecem com partes de informações discretas devem ser totalmente executadas, caso contrário, se forem divididas, não serão executadas. Dessa forma, esse conceito, assim como o átomo, garante a indivisibilidade de suas partes.
- **Consistência:** garante que se manterá o estado inicial dos dados, caso ocorra alguma falha, ou cria um outro estado dos dados, que

também é válido. Portanto, esse conceito é simples de entender, pois garante a consistência dos dados, como o nome diz.


- **Isolamento:** esse conceito é como se fosse uma proteção quando uma transação não foi validada, mas ela está em andamento, então, o conceito garante que deve permanecer isolada, ou seja, não haverá interferência por outra ação que acontece concomitantemente.
- **Durabilidade:** esse conceito permite que os dados que foram validados pelo sistema fiquem disponíveis, independentemente de falha ou de reinício do sistema.

Os conceitos dessas transações deverão ser sempre lembrados por quem trabalha com bancos de dados, a fim de garantir a qualidade dos dados e das aplicações que forem criadas e que precisarem ser executadas.

Figura 1 – ACID



Fonte: elaborada pela autora.



ACID é um acrônimo para atomicidade, consistência, isolamento e durabilidade. Essas são quatro propriedades de transação de um sistema de banco de dados. A seguir, listamos os detalhes sobre cada uma delas:

- **Atomicidade:** a atomização se encontra a nível de transação. Isso quer dizer que, em uma transação envolvendo duas ou mais partes de informações discretas, a transação poderá ser ou não executada, garantindo que as transações sejam atômicas.
- **Consistência:** a consistência se refere à integridade do dado, ou seja, a transação cria um novo estado válido dos dados para que, em caso de falha, possa retornar todos os dados ao seu estado anterior.
- **Isolamento:** tal característica garante o isolamento de transações concorrentes. Em outras palavras, uma transação em andamento que ainda não esteja validada, deve permanecer isolada de qualquer outra operação, garantindo que a transação não será interferida por nenhuma outra.
- **Durabilidade:** após validação dos dados, eles são registrados de modo a não haver perda em casos de falhas ou quando reiniciamos o sistema. Dessa forma, garantimos que os dados estejam disponíveis em seu estado correto.

2. Cardinalidade

A cardinalidade é uma propriedade importante que pode ser entendida como um número que é expresso por meio de um comportamento, como o número de ocorrências de uma certa entidade. Desse modo, a cardinalidade acontece por meio de um conjunto de associações entre entidades, esse conjunto pode ser chamado de relacionamento. Por sua vez, podemos dizer que entidade é algo ou uma “coisa” em que se deseja guardar as informações no banco de dados.

Dentro da cardinalidade podemos falar que existem dois tipos, a mínima e a máxima.

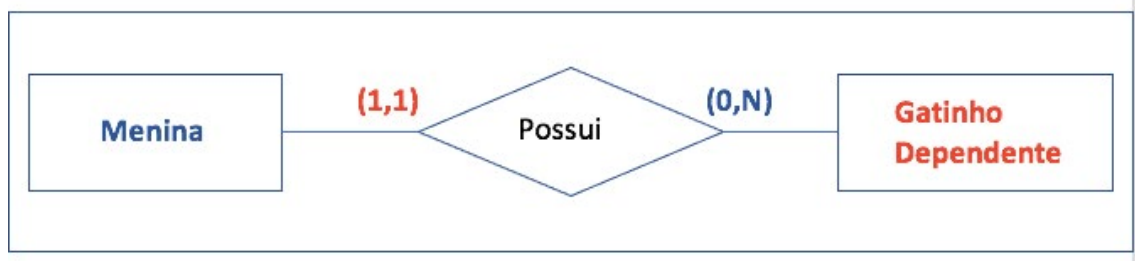
- Cardinalidade Máxima: diz respeito ao máximo número de ocorrências de uma entidade em que se esteja relacionado.
- Cardinalidade Mínima: representa o número mínimo de ocorrências de uma entidade por meio do relacionamento.

Já para os dois tipos das cardinalidades temos outras formas de distinção:

- Cardinalidade máxima, dois tipos $\rightarrow 1$ e N , que são cardinalidades maiores que um.
- Cardinalidade mínima, dois tipos $\rightarrow 0$ e 1 .

A seguir, apresentamos um modelo de exemplo para explicar de forma simples a cardinalidade. Neste diagrama, observe como estão representadas as cardinalidades, máxima e mínima.

Figura 2 – Exemplo de cardinalidade



Fonte: elaborada pela autora.

Nesse modelo, a cardinalidade é representada por $(0:N)$, que faz referência a Menina, e a cardinalidade $(1:1)$ se refere a Gatinho Dependente. Logo:

Uma ocorrência de Menina pode estar associada à várias ocorrências de Gatinho Dependente ou pode não estar associada a uma ocorrência de

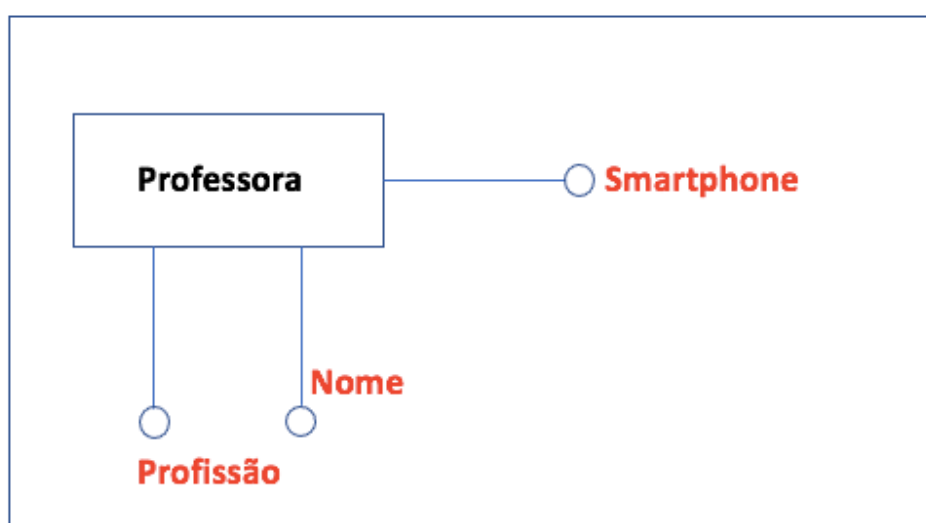
Gatinho Dependente. Note que é possível que uma certa Menina possa ter vários Gatinhos Dependentes e pode não possuir Gatinho.

Uma ocorrência de Gatinho Dependente, por sua vez, está associada a apenas uma ocorrência de Menina. Então, determinado Gatinho Dependente possui apenas uma Menina responsável por ele.

2.1 Cardinalidade do atributo

O Atributo é uma característica importante que se associa a cada tipo de ocorrência ou do que chamamos anteriormente de relacionamento. Para exemplificar e tornar mais claro esse conceito, observe o modelo a seguir, em que temos os atributos designados a determinada entidade.

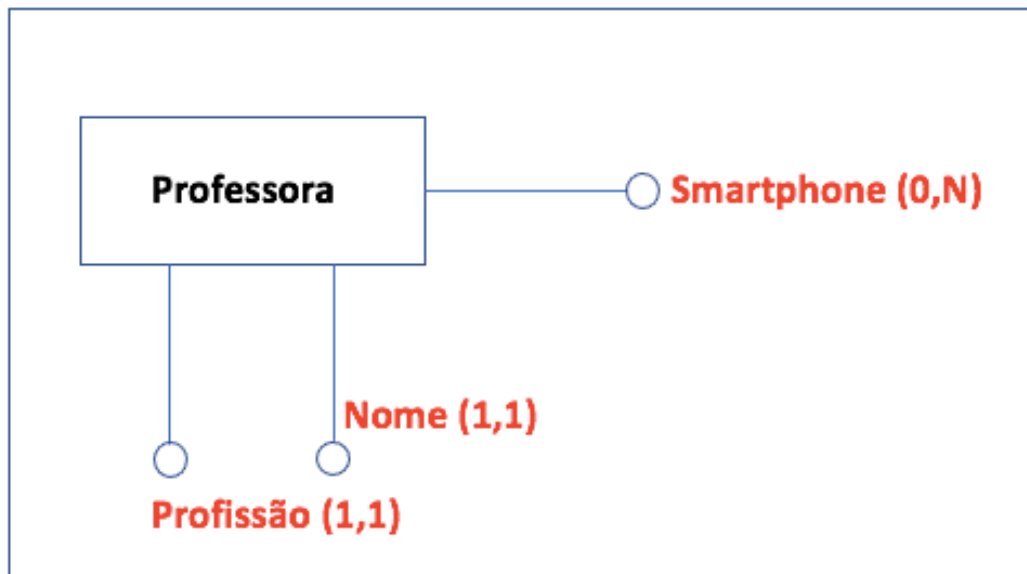
Figura 3 – Exemplo de atributo



Fonte: elaborada pela autora.

Observe o modelo anterior e note que não é informado quantos smartphones a Professora pode ter, ou se a professora pode não ter smartphone. Então, com o objetivo de deixar o modelo mais informativo e preciso, é comum expressar a cardinalidade para os atributos. Assim, é possível perceber que podemos ter a cardinalidade do atributo smartphone no modelo muito parecido, mas com um detalhe importante a mais em um dos atributos.

Figura 4 – Detalhamento dos atributos do exemplo anterior



Fonte: elaborada pela autora.

Observe a Figura 4 que apresenta um modelo muito parecido com o anterior e tem a cardinalidade informando que determinada Professora pode não ter smartphone, veja que o número 0 (zero) representa a cardinalidade mínima zero ou pode ter vários, em que a letra N representa a cardinalidade máxima. Já nos atributos profissão e nome, a cardinalidade é (1,1).

Quando falamos de atributos e possuímos a cardinalidade mínima é igual a 1, isso significa que o atributo é obrigatório, quando a cardinalidade máxima é igual a 1, o atributo é monovalorado. Já para a cardinalidade 0 (zero), que é mínima, significa que o atributo é opcional, quando a cardinalidade máxima é igual a N, o atributo é multivalorado.

➤ 3. Confiabilidade e benefícios do banco de dados relacional

Bancos de dados relacionais são fundamentados no paradigma da orientação a conjuntos. Os seus dados, por sua vez, são armazenados

em estruturas denominadas tabelas, e cada tabela é composta por colunas (atributos e linhas), tuplas ou registros.

Eles costumam ser mais utilizados para dados tabulares, sendo de fácil inserção e recuperação. Sua linguagem é o SQL (*Structured Query Language*) e seus principais representantes são Oracle, SQL Server, MySQL e PostgreSQL.

Os bancos relacionais são a opção ideal para sistemas ERP, CRM ou de gerenciamento financeiro, em que é necessária uma grande consistência de dados. Criado em 1970, por Edgar Frank Codd, esse modelo é o sucessor dos modelos hierárquico e em rede.

Para os sistemas desenvolvidos com o objetivo de criar, alterar, excluir e consultar informações que tenham um padrão de formato regular, o banco de dados mais adequado é o relacional. Já as soluções baseadas em nuvem e dados não tabulados com características heterogêneas exigem a utilização de um banco não relacional.

Figura 5 – Banco de dados



Fonte: Kayocci/iStock.com.

O meio de armazenamento considerado mais seguro é o Banco de Dados Relacional. Ele já possui várias implementações de segurança prontas, sem precisar que o programador da aplicação se preocupe demais neste aspecto. O Banco de Dados também se destaca por todas suas transações terem as propriedades ACID. Estas propriedades garantem que todo dado armazenado possa gerar uma informação segura quando necessário. Elas são essenciais pela natureza do aplicativo de base dos testes e, portanto, o tempo gasto na execução é o preço a se pagar pelas características benéficas adicionadas ao projeto.

Outro ponto de destaque do Banco de Dados Relacional é que ele é utilizado por diversas pessoas, o que resulta em formas de uso diferentes, que pode ser considerado como forma de testes do software.

Do ponto de vista de velocidade, o Banco de Dados Relacional foi considerado como a melhor opção de armazenamento, pois na maioria das aplicações das empresas no Brasil e no mundo este se mostrou mais veloz em termos de armazenamento do que qualquer outro sistema de arquivos, por melhor que fosse.

O Banco de Dados Relacional em que se relaciona com vários outros bancos de dados, por isso se torna muito mais eficiente. A figura abaixo ilustra esta característica.

Figura 6 – Banco de Dados Relacional



Fonte: nabihariahi/iStock.com.



4. Aplicações e futuro do banco de dados relacional

O Banco de Dados relacional foi e, ainda, é bastante utilizado devido a várias vantagens como densidade, velocidade, atualidade e proteção. O banco de dados relacional passou a ser usado em todos os tipos de aplicações, e os desenvolvedores passaram a adequar suas aplicações ao banco de dados relacional.

Com o avanço tecnológico, a ideia de se armazenar todo tipo de dado no banco de dados relacional fez com que diversos profissionais da área enfrentassem vários problemas, principalmente quando se tratava de armazenamento de dados complexos e de grande volume.

Nos últimos anos, por sua vez, ocorreu um crescimento exponencial no número de dados coletados e armazenados pelos mais variados dispositivos e sistemas de informação. O advento da internet tem revolucionado a forma como criamos conteúdos e trocamos informações, além de criar um ambiente favorável para o surgimento de uma série de aplicações com demanda de requisitos diferenciados como um alto grau de disponibilidade e escalabilidade sobre demanda, o que contribuiu para o surgimento de novos paradigmas de banco de dados e novas tecnologias de Sistemas de Gerenciamento de Banco de Dados (SGBD).

O termo Big Data é utilizado para definir este grande volume de dados de alta velocidade, complexos e variáveis que requerem técnicas e tecnologias avançadas para permitir a captura, armazenamento, distribuição, gerenciamento, e análise da informação.

A explosão no uso de Big Data fez com que grandes empresas demandassem por SGBDs capazes de gerenciar grandes volumes de dados de forma eficaz e com um alto desempenho. Nesse contexto, os

tradicionais SGBDs relacionais apresentam algumas limitações quanto a alta concorrência em operações de leitura e escrita, armazenamento de forma eficiente, suporte a escalabilidade horizontal, além da garantia de um serviço rápido e uma alta disponibilidade. Levando em consideração essas necessidades, uma variedade de novos SGBDs surgiram, focados principalmente no baixo custo de operação e manutenção.


Em aplicações denominadas de aplicações de alto risco, como aplicações hospitalares e controle de tráfego aéreo há demanda de maior integridade de dados, e por este motivo é tão importante seguir a propriedade ACID, que garante a integridade de seus dados e prioriza a segurança dos dados entre outras propriedades.

Devido a importantes propriedades encontradas nos Bancos de Dados Relacional, que implementam o paradigma relacional, as aplicações para as empresas são muito boas e os ambientes denominados de Ambientes Corporativos e Aplicações de Alto Risco, o desenvolvedor adiciona pontos mais valiosos na escola do tipo de banco de dados relacional. Para aplicações de baixo risco, a integridade de dados não é um fator limitante.

Figura 7-Proteção de dados



Fonte: djvstock/iStock.com.



Em suma, nesta unidade, tivemos a oportunidade de aprender diversos conceitos muito importantes no Banco de Dados Relacional que também utiliza Big Data dependendo das aplicações. Dessa forma, entendemos que em aplicações de alto risco, em que os dados precisam estar muito seguros, é importante garantir as propriedades ACID que são parte do conceito do tipo de Banco de Dados Relacional.

Referências Bibliográficas

LEAL, L. C. G. **Linguagem, programação e banco de dados**. Curitiba: InterSaberes, 2015.

MEDEIROS, F. L. **Banco de dados: princípios e práticas**. Curitiba: InterSaberes, 2013.

NAVATHE, E. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.

TAURION, C. **Big data**. São Paulo: Brasport, 2013.

A SQL e os Sistemas de Gestão de Banco de Dados

Autoria: Cassio Rodolfo Aveiro da Silva

Leitura crítica: Flavio Fiuza



Objetivos

- Introduzir o conceito e aplicabilidade dos *Database Management System* (DBMS).
- Compreender a importância do *Structure Query Language* (SQL).
- Descobrir os diferentes papéis da SQL na gestão dos bancos de dados.
- Aprender sobre os princípios das linguagens *Data Definition Language* (DDL) e *Data Manipulation Language* (DML) dentro da SQL.



1. Os Sistemas de gestão do banco de dados

A partir da premissa de que já há um conhecimento prévio sobre os princípios básicos de um banco de dados relacional e suas transações de dados e informações necessário à administração e gestão empresarial, você deve compreender e identificar agora, a necessidade do uso de um Sistema de Gestão de Banco de Dados (DBMS–*Data Base Management System*).

Os DBMS ou também conhecidos pela sigla na língua portuguesa como SGBD, constituem-se essencialmente de um software específico destinado a criação, armazenamento, organização e acesso a dados, a partir de um determinado banco de dados (LAUDON; LAUDON, 2014).

Historicamente, os pacotes comerciais DBMSs evoluíram, de seus antepassados e ultrapassados sistemas monolíticos (integrados por meio de um bloco único de softwares) para os contemporâneos sistemas modulares voltados ao projeto a ser desenvolvido, promovendo maior flexibilidade e com característica amplamente centradas na relação cliente/servidor (ELMASRI; NAVATHE, 2009).

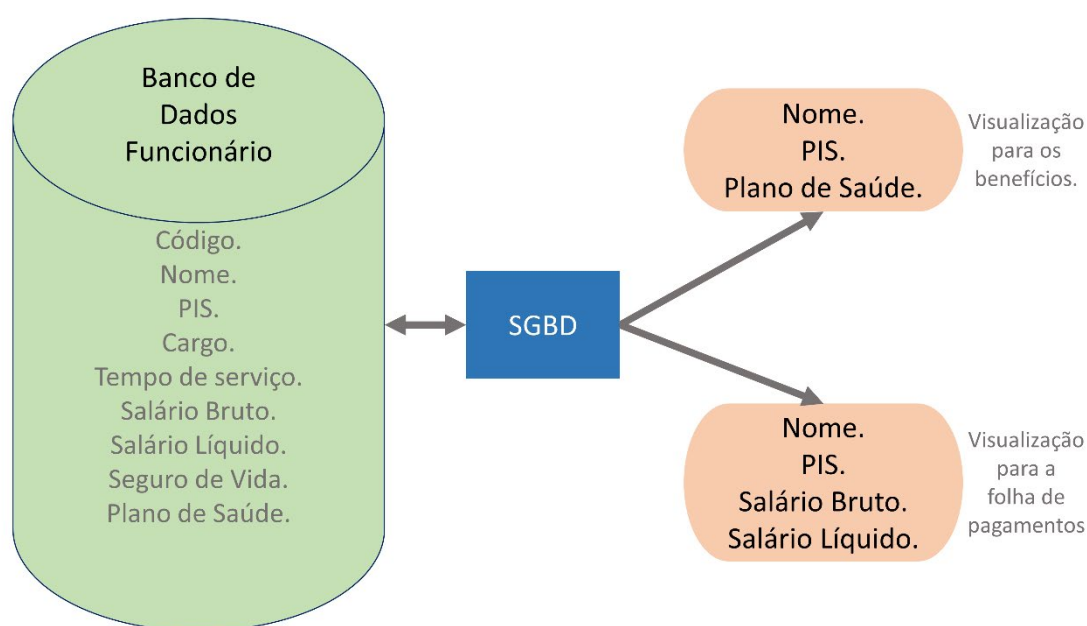
Segundo Elmasri e Navathe (2009), a evolução dos DBMSs nada mais é do que um reflexo do desenvolvimento tecnológico que substituiu os computadores demasiadamente grandes e centralizados, por PCs (*Personal Computer*–computador pessoal), *workstations* (estações de trabalho) remotas e atualmente por *notebooks* e dispositivos móveis. Além disso, os DBMSs foram impulsionados pelo desenvolvimento dos servidores de Web, de Banco de Dados e de aplicações.

Laudon e Laudon (2014) acreditam que os DBMSs eliminam a responsabilidade de usuários e programadores em identificar em que lugar e de que maneira são armazenados os dados. Os sistemas de gestão criam duas concepções sobre os dados: *lógica* e *física*. Na

questão lógica, os dados são apresentados da maneira como usuários e especialistas necessitam vê-los, enquanto que na parte física, os dados são apresentados como realmente são estruturados, organizados e armazenados, como por exemplo em um disco rígido.


Na verdade, há uma relação entre essas concepções. Os DBMSs promovem diferentes visualizações lógicas dos dados para um único banco de dados físico. Na Figura 1 apresentamos um exemplo dessa relação.

Figura 1 – Ação do DBMS (SGBD) no banco de dados de Recursos Humanos



Fonte: adaptada de Laudon e Laudon (2014).

No exemplo apresentado, o Banco de Dados de Recursos Humanos (RH) possui um DBMS. Esse sistema identifica cada um dos dados físicos referentes a cada um dos funcionários de uma empresa e então os processa para gerar visualizações lógicas para o especialista em benefícios que necessita do nome e os números da Previdência Social e do Plano de Saúde de um dado funcionário. O DBMS gera também uma visualização para o encarregado de confeccionar a folha de pagamento apresentando nome, número do PIS e os valores dos salários bruto e



líquido. Como dito anteriormente, esse exemplo ilustra a relação de várias visões lógicas para um único banco de dados (LAUDON; LAUDON, 2014).

1.1 DBMS: estruturação e interfaces

Com relação à estrutura dos sistemas de gestão de banco de dados, cada fabricante evidentemente insere as características que refletem à suas respectivas marcas, mas de um modo geral, pode-se analisar a estrutura desses sistemas baseada na relação cliente/servidor e distribuída em dois grupos de módulos funcionais: *módulo servidor* e *modulo cliente* (ELMASRI; NAVATHE, 2009).

O Módulo Cliente, segundo Machado (2020), deve ser utilizado em um PC ou estação de trabalho, uma vez que se apresenta com interface amigável e intuitiva direcionada ao usuário e compostas por exemplo, pela chamada GUI (*Graphical User Interface*–Interface Gráfica para Usuários). Este módulo promove, portanto, a interação com os clientes (usuários).

Já o Módulo Servidor, trabalha com as questões mais técnicas voltadas ao armazenamento de dados, bem como com a liberação de acessos e pesquisas (ELMASRI; NAVATHE, 2009).


Para autores como Machado (2020), Laudon e Laudon (2014) e Elmasri e Navathe (2009), os sistemas de gestão de banco de dados apresentam basicamente seis tipos de interfaces:

- Interface Baseada em Menus (MBI–*Menu Based Interface*): para clientes Web e de navegação. Apresenta opções de ação em forma lista de botões (menu), eliminando a necessidade de *decorar* comandos e sintaxes de programação da linguagem de consulta. Possibilitam pesquisas exploratórias sem estruturação.

- Interface Baseada em Formulários (FBI–*Forms Based Interface*): Gera um formulário para cada usuário. Empregado na atividade de inserção de dados em um banco. Se o dado for novo, o formulário obrigatoriamente deve ser integralmente preenchido. Já para acessar dados, apenas alguns campos do formulário precisam ser preenchidos, enquanto os demais são auto preenchidos pelo DBMS.
- Interface Gráfica para Usuários (GUI): Pesquisa de dados estruturada na manipulação, por parte do usuário, de formulários ou menus diagramáticos com o auxílio de um indicador como o mouse de um PC, por exemplo.
- Interface de Linguagem Natural (NLI–*Natural Language Interface*): Permite a realização de uma consulta por meio de palavras escritas em inglês ou em outro idioma. A interface então, tenta *entender* o que foi solicitado por meio de um esquema de palavras-padrão iniciando uma espécie de *diálogo* com o usuário (linguagem de alto nível) interagindo com ele para refinar a solicitação de consulta exigida.
- Interface para Usuários Parametrizáveis (PUI–*Parameterizable User Interface*): para usuários que realizam operações de forma repetitiva, como o grupo dos *caixas de bancos*, por exemplo. O objetivo é reduzir o número de *teclas* e *clicks* realizados em cada consulta ao banco de dados.
- Interface para os Administradores dos Bancos de Dados (DBAI–*Data Base Administrator Interface*): Comandos especiais utilizados apenas pelos Administradores dos Bancos de Dados (DBA) com acesso restrito.

1.2 DBMS relacional

De maneira semelhante ao exemplo do banco de dados de RH, em um Banco de Dados Relacional (BDR), os sistemas de gestão de banco de




dados utilizam-se das combinações de tabelas (por meio das chaves estrangeiras) presentes nesse tipo de banco, para apresentar ao usuário os dados solicitados. Para tal, os DBMSs utilizam-se de três operações básicas: *Select*, *Project* e *Join* (selecionar, projetar e vincular). A primeira cria um conjunto de todas as *Tuplas* (registros, linhas) de uma entidade (tabela) que atendem à uma especificação. A função *Join* combina então tabelas relacionais para obter mais informações do que a análise das entidades individualmente pode fornecer. Por fim, a função *Project* gera um conjunto de atributos (colunas) permitindo o usuário confeccionar uma nova tabela contendo apenas as informações necessárias e requisitadas (LAUDON; LAUDON, 2014).

Segundo Laudon; Laudon (2014) para obterem uma gestão adequada de um banco de dados relacional, os DBMSs possuem recursos e ferramentas para definição, manipulação de dados e dicionário próprio. O objetivo é especificar a estrutura do conteúdo de um banco de dados, criando tabelas específicas com determinados atributos. As definições são então armazenadas em uma espécie de dicionário de dados, registrado em um arquivo, digital ou até mesmo manual.

Neste contexto, você verá a seguir a linguagem por trás dos DBMSs responsáveis pela definição e manipulação dos dados baseados em comandos.

2. O surgimento e a necessidade pela SQL

O meio empresarial, de acordo com Machado (2020), imerso em Tecnologia de Informação e repleto de vários Sistemas de Informação, tem o Banco de Dados Relacional como a ferramenta de maior destaque em efetividade no armazenamento de dados nos últimos anos. Assim, grande parte desse sucesso do BDR se deve à utilização da *Structure*




Query Language (SQL–Linguagem Estruturada de Consulta), segundo Elmasri (2009), em função de sua facilidade de compreensão e utilização.

Conforme apresentado por Machado (2020), a Linguagem Estruturada de Consulta (ou pesquisa), comumente chamada apenas de SQL, teve sua origem nos estudos sobre modelos relacionais de Codd por volta de 1970. Contudo, sua primeira versão aplicável foi desenvolvida em 1974 nas instalações da *International Business Machines* (IBM) por Chamberlin, a qual recebeu o nome de SEQUEL (*Structured English Query Language*–Linguagem de consulta estruturada em inglês). Claramente, a SEQUEL foi ampliada, revisada e renomeada para SQL entre 1976 e 1977 (COSTA, 2006).

A empresa IBM, desenvolveu a SQL com o propósito de utilizar uma interface amigável para seu sistema experimental de banco de dados relacional que ficou conhecido como *System R* (Sistema R). De acordo com Machado (2020), foi por essa utilização destinada à pesquisa e manipulação de dados dentro de um banco de dados relacional, que a SQL foi criando destaque em meio as linguagens disponíveis e aumentando drasticamente sua aplicação como linguagem básica de Sistemas de Gestão de Banco de Dados.

Seu sucesso foi tão grande que em 1982, aliado à necessidade de padronização referente aos bancos de dados, a *American National Standard Institute* (ANSI–Instituto Nacional Americano de Padrões) tornou a SQL a linguagem oficial padrão no âmbito de sistemas relacionais. O que em 1986 também foi adotado pela *International Organization for Standardization* (ISO–Organização Internacional de Normalização). Atualmente, segundo Machado (2020), essa padronização não é tão difundida, surgindo cada vez mais linguagens distintas de SQL, competindo para se tornarem as mais utilizadas mundialmente.

Comumente, você verá conceitos e definições referentes aos padrões ISO e ANSI, o essencial para lidar com os DBMS relacionais ou não.



Contudo, é interessante que você pesquise sobre a evolução desse padrão SQL (ANSI e ISO) que passou por versões conhecidas como SQL-86, SQL-92 e a mais atual, SQL-99.

2.1 A SQL como linguagem padrão de DBMS

Como uma linguagem destinada sobretudo aos bancos de dados relacionais, a SQL permite basicamente definir, consultar e atualizar dados, além de possuir funcionalidades para definições no banco de dados que visam garantir a segurança, controle e integridade das transações (MACHADO, 2020).

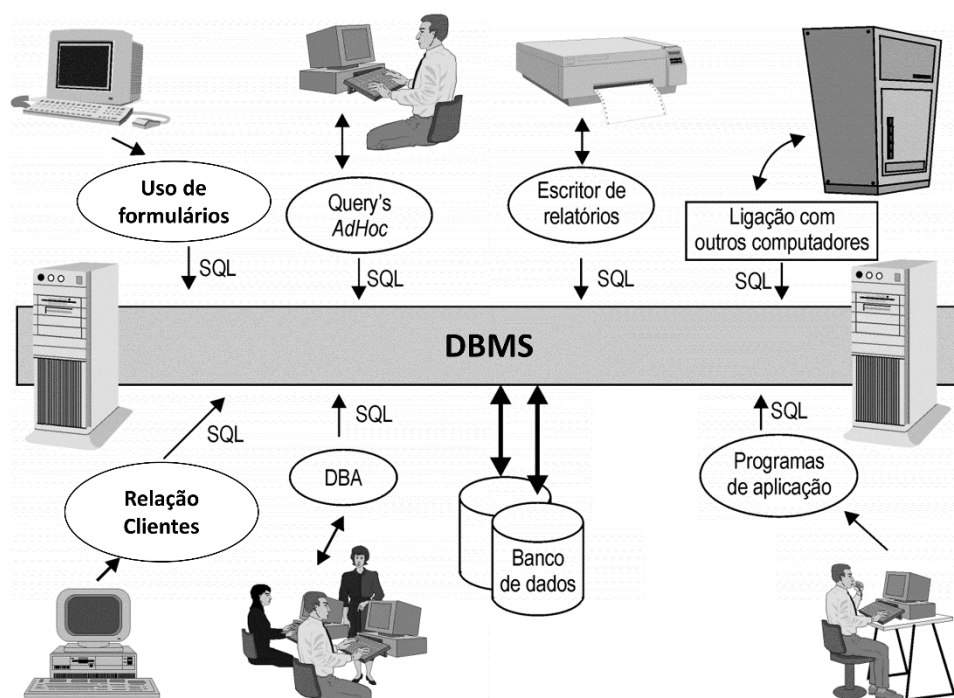
Aliado a tudo isso, a SQL permite ser utilizada por meio de linguagens ditas *hospedeiras*, como a Linguagens “C” e *Fortran*, bem conhecidas no meio empresarial e acadêmico (ELMASRI; NAVATHE, 2009; MACHADO, 2020).

Segundo Elmasri e Navathe (2009), em função de sua grande expansão frente aos bancos relacionais, os clientes e ou usuários da SQL não precisam se preocupar com a migração de dados entre sistemas, uma vez que é uma linguagem padrão desses sistemas, apresentado apenas pequenas diferenças de comandos entre um e outro, nada que gere uma preocupação ou prejuízos. Para Costa (2006), Elmasri e Navathe (2009), os DBMS como um todo, sempre terão a mesma essência advinda do uso da SQL, ou seja, as conversões de dados quando necessárias são simples e não requerem um aprofundamento, bem como uma grande disponibilidade de tempo e dinheiro.

Date (2015), ratificado por Machado (2020), acredita que a SQL pode ter um papel diversificado e de extrema importância em um sistema de gestão de banco de dados. A linguagem age como um facilitador/mediador na troca de informações entre diferentes finalidades destinadas ao uso de um DBMS. Trata-se de um processo de integração entre

software, sistemas, máquinas e pessoas, possibilitada graças ao uso da SQL. A figura a seguir traz um contexto da utilização diversificada da SQL.

Figura 2–Multifaces de utilização da SQL em um DBMS



Fonte: adaptada de Machado (2020).

Conforme já mencionado e observado na Figura 1, segundo Machado (2020), a SQL permite:

- *Consultas Interativas (Query AdHoc)*–Os usuários de SQL podem desenvolver poderosas pesquisas de dados e informações, baseadas em formulários ou ferramentas de geração de relatórios.
- *Acessibilidade a banco de dados*–Auxiliam os programadores a obter acesso a banco de dados durante a programação de suas aplicações.
- *Otimização da gestão*–Os DBA realizam suas tarefas de forma otimizada por comandos SQL.

- *Relacionamento Servidor/Cliente*–Os sites e programas acessados pelos clientes em seus computadores pessoais podem por meio da SQL, utilizar-se de uma rede local para compartilhar dados de um mesmo servidor, minimizando assim o tráfego desses dados e possível travamentos nos sistemas.
- *Transações entre bancos de dados*–A SQL facilita a distribuição de dados entre diferentes bancos de dados e sistemas.
- *Comunicação sistemas/máquinas*–Possibilita o uso de sistemas semelhantes, como por exemplo os sistemas CAM (*Computer Aided Manufacturing* – Manufatura Auxiliada por Computador), por meio de uma eficiente conversão de dados na comunicação entre computadores e máquinas.

Dentro de um DBMS, a SQL em virtude das inúmeras aplicabilidades apresentadas acima, pode alterar diferentes funções ou tarefas desse sistema. Ao mesmo tempo, protege contra a alteração de dados não autorizada (MACHADO, 2020).

Nesse sentido, Laudon, Laudon (2014) e Machado (2020), também, apresentam a SQL como uma ferramenta organizacional de compartilhamento de dados entre usuários (empresas) concorrentes, sem o risco de interferir nas atividades empresariais de cada um ou fornecer dados sigilosos por engano. Além disso, otimiza a função dos bancos de dados relacionais, de manter a integridade dos dados frente a possíveis falhas, corrupções ou inconsistências nos sistemas de gerenciamento.

Com um banco de dados definido e um sistema de gestão adequado, é necessário estabelecer como será utilizada a SQL para especificar

relações entre modelos visuais lógicos e físicos a serem estruturados (LAUDON; LAUDON, 2014).

Se o DBMS escolhido não exigir uma divisão específica referente a níveis dentro do banco de dados, pode ser utilizado a SQL por meio da *Data Definition Language* (DDL–Linguagem de Definição de Dados). Contudo, se houve uma exigência para uma clara separação entre visualização lógica e estruturação interna, a DDL deve ser utilizada apenas para o modelo lógico, sendo necessário outra linguagem para o tratamento físico dos dados. Nesse caso, o DBMS contará com um compilador DDL responsável por ler e escrever os comandos da linguagem DDL (ELMASRI; NAVATHE, 2009).

Nem sempre os modelos apresentados por um banco de dados estarão estruturados e esquematizados de forma a facilitar o acesso e inserção de informações. De modo geral, eles podem estar compilados, o que exige uma manipulação do banco de dados por parte dos usuários, como alterações, recuperações, remoções ou realocações. Faz-se necessário a utilização da SQL baseada na *Data Manipulation Language* (DML–Linguagem de Manipulação de Dados) específica para essa finalidade (ELMASRI; NAVATHE, 2009).

Fundamentalmente, a DDL e a DML são denominadas *linguagens não distintas*, que constituem a linguagem integrada e relacional de banco de dados, SQL. Destas, a DML é a que exige um pouco mais de compreensão.

No entanto, há dois principais tipos de DMLs: a de *nível baixo* também conhecida por *processual*; e, a de *alto nível* ou *não processual*. Machado (2020) e Elmasri, Navathe (2009) trazem uma conceituação sobre essa classificação:

- *DML de Alto Nível*: focados em especificar suas próprias operações dentro de um banco de dados, isto é, por meio de um terminal, monitor ou até por uma linguagem de programação geral, os comandos do DML são identificados e extraído por pré-compiladores e posteriormente processados pelo DBMS.
- *DML de Baixo Nível*: necessita ser embutida em uma linguagem de programação geral, por meio de códigos de programação como *loop*. O foco é recuperar e processar registros do banco de dados de forma individual e um por vez.

Você deve perceber que a diferença entre os tipos de DML é que a de Alto Nível identifica *quais* dados devem ser recuperados enquanto que a de baixo nível especificam *como* recuperá-los (ELMASRI; NAVATHE, 2009).

Um fato interessante é que se as DMLs estiverem inseridas dentro de uma linguagem de programação geral, essa linguagem recebe o nome de *linguagem hospedeira* já mencionada anteriormente, na qual a DML é uma *sub linguagem* (ELMASRI; NAVATHE, 2009).

2.2 Comandos e operadores lógicos da DDL e DML

Costa (2006) traz uma abordagem direta e conceitual sobre os principais comandos utilizados em DBMS constituído de DDL:

- CREATE: para geração de um banco de dados ou tabelas para um banco de dados relacional. Um exemplo de criação de tabela:

```
CREATE TABLE <nome_tabela>
```

- SHOW: pesquisa de um banco de dados ou tabela. Um exemplo para listar bancos de dados existentes:

SHOW DATABASES;

- DROP: para a exclusão de tabelas ou bancos inteiros de dados. Um exemplo de exclusão de banco de dados (BD):

DROP DATABASE <nome_BD>;

- ALTER: para modificar a estrutura de uma tabela do banco de dados. Um exemplo de adição de coluna:

AFTER TABLE <nome_tabela> ADD
<nome_campo> <tipo_campo>(tamanho);

- TRUNCATE: para eliminar registros de uma base de dados. Um exemplo de exclusão de registros dentro de uma tabela:

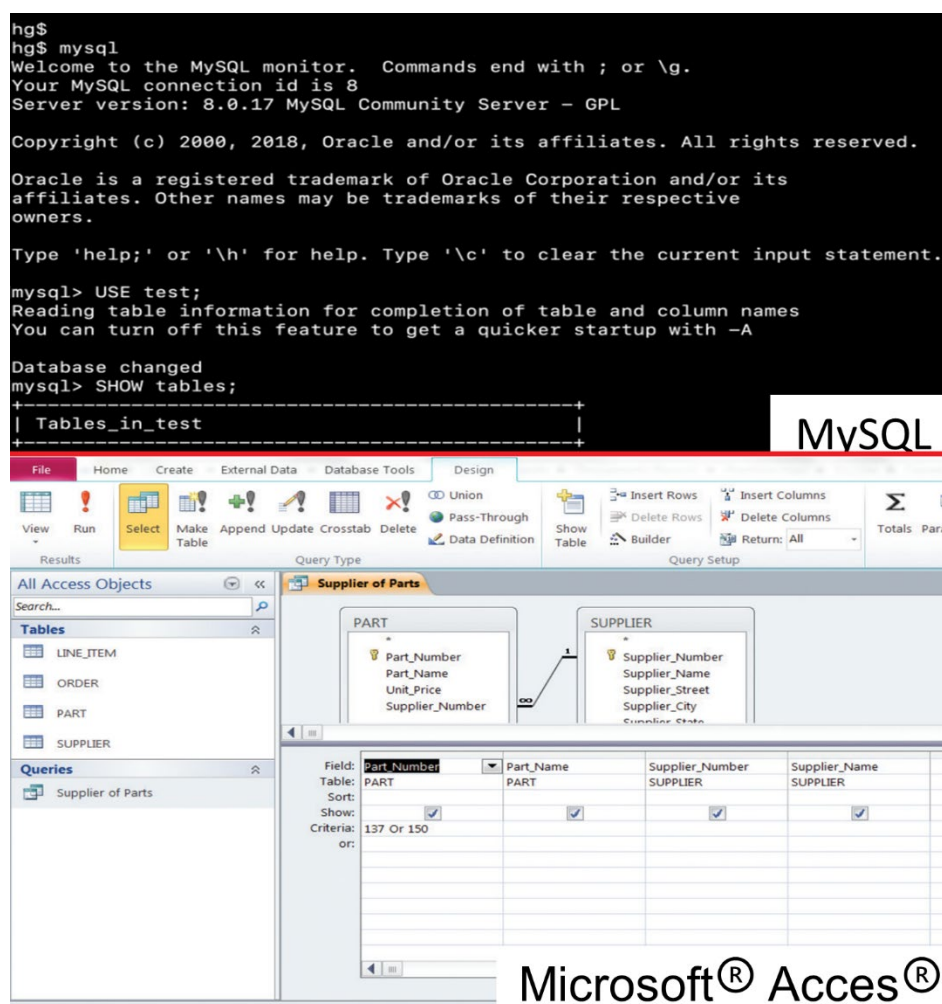
TRUNCATE <nome_tabela>;

- RENAME: para renomeação de tabelas e campos de tabelas de um banco de dados. Um exemplo para renomear uma tabela de um BD:

AFTER TABLE <nome_tabela> RENAME
<novo_nome_tabela>;

Esses comandos podem ser utilizados no MySQL, um dos softwares de gestão de banco de dados mais populares e de código aberto. Podem ser aplicado também no pacote comercial Microsoft® Acces®. Ambos tem potencial para realizarem uma excelente gestão de banco de dados, mas por ser pago, o Acces® apresenta-se ao usuário de forma mais amigável, enquanto que o MySQL é apresentado em forma de terminal de comandos na hora em que torna-se necessário manipular dados (Figura 3). Apenas na hora de consulta de dados é que o MySQL apresenta uma interface mais intuitiva.

Figura 3-Interfaces de manipulação de dados do MySQL e do Microsoft® Access®



Fonte: adaptada de Laudon e Laudon (2014).

Com os comandos DDL cria-se o banco de dados e, a partir deste momento, torna-se necessário inserir dados nas tabelas e manipulá-los posteriormente. Para isso, Elmasri e Navathe (2009) apresentam os operadores (Quadro 1) e os comandos básicos do DML:

Quadro 1-Operadores relacionais DML

Operador	Descrição
=	Igual
!=	Diferente
>	Maior
>=	Maior que

<	Menor
<=	Menor que
&&	E
	OU
!	NÃO
true	verdadeiro
false	falso

Fonte: elaborado pelo autor.

- INSERT: para inserir dados nas tabelas. Exemplo:

```
INSERT <nome_tabela> VALUES
(<Valor_campo_1>,<Valor_campo_2>,...);
```

- SELECT: seleção de dados a serem apresentados. Pode ser utilizado para selecionar todos ou apenas alguns dados. Um exemplo de retornar toda a tabela:

```
SELECT*FROM <nome_tabela>;
```

- UPDATE: para a atualização de dados. Um exemplo de alteração de valor:

```
UPDATE <nome_tabela> SET <coluna>
<operador> <novo_valor>;
```

- DELETE: para apagar dados. Um exemplo de exclusão de um ou mais registros:

```
DELETE FROM <nome_tabela> WHERE <condição>;
```

Vale ressaltar que os comandos apresentados são apenas exemplos aplicados de forma simplista a DBMS específicos. Na prática, durante o desenvolvimento de um banco, definição e manipulação de dados, haverá inúmeras combinações de comandos, condições e operadores, bem como a utilização de comandos não apresentados. Logo, é recomendado a você que se aprofunde mais sobre a linguagem de

programação SQL, uma vez que é bem semelhante às linguagens como C, C++, entre outras de alto nível e de fácil compreensão. Um conhecimento a mais, que impulsionará sua carreira profissional no âmbito da gestão de banco de dados.

Portanto, os sistemas de gestão de banco de dados permitem transformar os dados físicos armazenados em um banco específico, em dados visuais direcionado a determinados usuários como DBAs, *Query AdHocs*, gestores da relação cliente/servidor, programadores de aplicativos, geradores de relatórios, entre muitos outros. Tudo isso graças a linguagem padrão, a SQL com suas sub linguagens DDL e DML, que facilitam a definição, manipulação, inserção e remoção de dados no referido banco de dados relacional.

Referências Bibliográficas

COSTA, R. L. **SQL: guia prático**. 2. ed. Rio de Janeiro: Brasport, 2006.

DATE, C. J. **SQL e teoria relacional: como escrever códigos SQL precisos**. Tradução: Rafael Zanolli. São Paulo: Novatec, 2015.

FERRARI, F. C. **Crie banco de dados em MySQL**. São Paulo: Digerati Books, 2007.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. Tradução: Marília G. Pinheiro. 4. ed. São Paulo: Pearson, 2009.

LAUDON, K.; LAUDON, J. **Sistemas de informações gerenciais**. Tradução: Célia Taniwaki. 11. ed. São Paulo: Pearson, 2014.

MACHADO, F. N. R. **Projeto e implementação de banco de dados**. 4. ed. São Paulo: Érica, 2020.

Big Data e bancos de dados não relacionais

Autoria: Cassio Rodolfo Aveiro da Silva

Leitura crítica: Flavio Fiuza



Objetivos

- Compreender o conceito e aplicabilidade de Big Data.
- Conhecer as características e principais ferramentas de Big Data.
- Entender o gerenciamento de dados com Hadoop.
- Aprender de forma concisa o que é NoSQL.



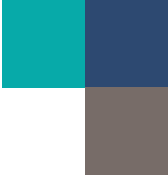
1. A necessidade pela Big Data

Na era digital atual, o fluxo de informações cresceu exponencialmente e empresas em todo o mundo passaram a utilizar o banco de dados como para gerir suas transações internas e armazenar dados importantes. Nesse contexto, não apenas dados internos, as empresas necessitaram expandir seu alcance de informação utilizando os bancos de dados externos, para tomar as melhores decisões e administrar seus processos de negócios de maneira mais eficiente (LAUDON; LAUDON, 2014).

Até meados dos anos 2000, os bancos de dados relacionais eram mais que suficientes para os dados transacionais de uma empresa. Porém, segundo Machado (2020), cada vez mais a exposição a uma enorme quantidade de dados promovida pelo uso da Web, permite aos gestores e administradores ampliar suas capacidades empresariais. Contudo, esse fenômeno tecnológico de informações, traz consigo dados geralmente não estruturados ou no máximo semiestruturados, pois suas origens encontram-se, por exemplo, em mídias sociais e dados automáticos gerados por sensores dos mais variados tipos. Para tais dados, as linhas e colunas típicas de um banco de dados relacional não são adequadas, além de necessitarem de análise em tempo real (LAUDON; LAUDON, 2014).

Para esse enorme volume dados, que exigem uma capacidade de processamento além da oferecida pelos típicos Sistemas de Gestão de Banco de Dados (DBMS–*Data Base Management System*), você deverá utilizar o conceito de *Big Data* (LAUDON; LAUDON, 2014).

Autores como Taurion (2013) apresentam a Big Data como um conceito relacionado ao gerenciamento e processamento de dados extremos excedendo a habilidade de coletar, armazenar e analisar, das tradicionais tecnologias. Assim sendo, Big Data é uma prática contemporânea geradora de novas oportunidades de processos de



negócios. É a ampliação da capacidade de visualização alternativa para coisas que já existem, mas que não se havia conhecimento.

Segundo Goldschmidt (2015), Big Data se refere ao armazenamento de dados na faixa de exabytes, ou simplesmente, trilhões de registros das mais variadas fontes, utilizado para obter informações importantes. Para as empresas, essa mineração de dados (*Data Mining*) com o intuito de adquirir informações valiosas, significa maiores possibilidades de obter soluções e novas percepções sobre seus processos de negócios, por meio da aquisição de mais padrões e propriedades interessantes em meio ao grande volume de dados, além do aumento significativamente elevado da precisão das informações geradas (LAUDON; LAUDON, 2014).

O mais interessante no âmbito de Big Data, é que não é necessário contar com *big servidores* ou conjunto de hardwares e softwares de grande poder computacional. Taurion (2013), por sua vez, apresenta a ocorrência de Big Data por meio da computação em nuvem, ampliando as possibilidades e reduzindo-se os custos ao utilizar nuvens públicas, apresentando despesas relacionadas apenas aos recursos utilizados para análise dos dados.

A realidade imposta às organizações, de modo geral, é que hoje gerenciar por meio de planilhas é uma prática considerada primitiva, ultrapassada, incapaz de atender a velocidade e complexidade das tomadas de decisões e acima de tudo, acompanhar a geração de informações advindas de poderosos dispositivos móveis utilizados pelos usuários desta geração (TAURION, 2013).

Logo, é extremamente importante você compreender de maneira eficaz o conceito de Big Data, para ter a habilidade de saber utilizá-lo, ou seja, saber como obter informações valiosas dessa avalanche de dados oferecida e quais suas limitações. Além disso, você deve levar em consideração que coletar, armazenar e analisar por Big Data, muitas vezes é um trabalho caro sem a garantia de que cem por cento das

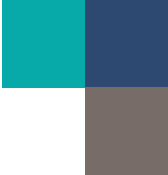
informações obtidas ajudaram a tomar uma decisão assertiva (LAUDON; LAUDON, 2014).

2. Compreendendo o que é Big Data

Alguns autores, como Barbieri (2011) e Taurion (2013), conceituam Big Data em 5 (cinco) “V”:

- I. **Volume:** Trilhões e trilhões de dados coletados diariamente, das mais variadas fontes possíveis, sobretudo de redes sociais, navegadores de internet e *e-commerces*. Quanto mais dados, melhor.
- II. **Velocidade:** Reação em tempo real frente ao imenso volume de dados que devem ser analisados e tratados de forma adequada.
- III. **Veracidade:** Garantia de que as fontes de onde os dados foram obtidos são confiáveis e imparciais, principalmente para que as informações geradas não se tornem tendenciosas.
- IV. **Variedade:** dados advindos de sistemas estruturados (atualmente pouco utilizados) e não estruturados. Diversas fontes de dados podem gerar informações de extrema importância ao serem combinadas, mesmo se em um primeiro momento pareçam sem relação entre si. Auxilia no processo de análises preditivas dentro das empresas.
- V. **Valor:** os dados são hoje o *petróleo* das empresas. Deve-se verificar o poder e importância que os dados trarão à organização, bem como analisar o retorno financeiro com implementação de projetos em Big Data.

Os 5V são as características básicas que definem Big Data. Para não haver dualidades, você deve compreender também o que *não é* Big Data. Evidentemente, não é um produto em forma de software ou hardware. Segundo Taurion (2013), esse é um combinado de




tecnologias, práticas e processos que possibilitam às empresas, o acesso e análise de dados que antes eram indetectáveis, além de promover uma gestão muito mais eficiente e de alta performance.

Big Data não é também uma teoria ou utopia. Todos os dias nós mesmos geramos inúmeros dados e fornecemos nossos padrões de vida e consumo a sistemas de informação. Esse montante de dados, se bem processado dará origem a informações de valor inestimável não apenas às empresas, mas para a própria sociedade como um todo, até mesmo para que governos possam compreender demandas e necessidades da população (TAURION, 2013).

Muitas empresas segundo Taurion (2013) e Laudon e Laudon (2014), acreditam que basta adquirir pacotes comerciais de tecnologia voltados apenas à análise e organização de dados para que Big Data já faça todo o resto. Pensamento errôneo. Não basta entender como promover uma análise de dados eficiente, é preciso atentar-se a algumas etapas que podem facilitar a total compreensão de Big Data.

Nesse momento é importante que você guarde em sua mente, que Big Data significa realizar transformações consideráveis nos processos de negócio, na infraestrutura da Tecnologia de Informação (TI), nas fontes atuais e futuras de obtenção de dados, nos treinamentos e capacitações de funcionários e nas estruturas organizacionais de uma empresa (LAUDON; LAUDON, 2014).

Primeiramente Big Data é um processo de coleta de dados, onde variedade e volume são suas propriedades. Neste ponto é que se percebe a complexidade envolvida. Segundo Taurion (2013), os dados devem ser coletados de inúmeras fontes como redes sociais, sistemas de informação tradicionais, sensores, câmeras de vigilância, serviços de atendimento ao cliente, entre muitas outras. Ou seja, tudo ao seu redor pode ser uma fonte de dados e isso representa um trabalho árduo e extenso.



Na sequência, é preciso que os dados obtidos das mais variadas fontes sejam tratados, como forma de garantir agregação e integração a estes. Erros ou corrupções em dados devem ser eliminados. Outro ponto a se pensar é sobre a segurança dos dados, que irá variar de acordo com suas fontes de origem. A maneira mais fácil de realizar esse processamento é criando categorias, critérios de avaliação e requisitos de aceitação de dados (TAURION, 2013).

O próximo passo para Laudon e Laudon (2014) são as atividades analíticas, para uma correta interpretação dos dados. O problema está em analisar os trilhões de dados. O foco nesta etapa é então identificar quais perguntas devem ser feitas para que padrões e relações agreguem um maior valor para os processos de negócio. Geralmente neste momento, muitas empresas recorrem a profissionais multifuncionais com capacidade e experiência para dar um rumo neste processo, o que requer investimento em pesquisas e novas soluções em visualização de dados para uma melhor interpretação, ao invés do uso dos tradicionais gráficos e planilhas (TAURION, 2013).

De acordo com Barbieri (2011), para uma plena operação de Big Data, é exigido um alto poder computacional, o que é suprido, como já mencionado, pelo armazenamento em nuvem.

3. Big Data e seus impactos

Assim, deve ficar claro que Big Data agrega valor empresarial ao identificar padrões e relações entre dados na Web por meio das mídias sociais e daqueles dados anteriormente perdidos ou descreditados. São vários os impactos da utilização da Big Data e que serão apresentados a você agora.

3.1 Transparência

Para Taurion (2013), Big Data, sobretudo no setor público, garante uma maior transparência em virtude de aumentar o acesso a dados inacessíveis até então, além de promover cruzamento de dados e consequentemente um aumento das chances de se visualizar desvios na gestão de órgãos públicos, por exemplo.


Ainda, segundo Laudon e Laudon (2014), ratificando o exposto por Taurion (2013), essa transparência de dados gera soluções para inúmeros problemas governamentais no que tange compreender o que sua população anseia. Assim, surgem cada vez mais, as chamadas *cidades inteligentes* que se utilizam da tecnologia de informação de maneira otimizada com o intuito de tornarem-se mais seguras e com maior mobilidade urbana.

3.2 Segmentação precisa da sociedade

Com Big Data, é totalmente possível segmentar a sociedade até chegar ao nível do indivíduo. Os já conhecidos dados de rastreamento quando o usuário navega pela internet, permitem por exemplo, discretizar padrões de compra, hábitos de leitura, padrões de interações nas redes sociais, entre muitos outros. Tudo para romper a barreira entre clientes e empresas, que por sua vez, podem agora influenciar consumidores e até mesmo fazer com que comprem algo que nem queriam de verdade. Ponto que chama a atenção para questões éticas e de privacidade do usuário (LAUDON; LAUDON, 2014; TAURION, 2013).

3.3 Ações preditivas

Outro impacto advindo de Big Data, é a previsibilidade. Segundo Taurion (2013), há estudos que comprovam que ao se analisar os padrões comportamentais de dados, pode-se prever por exemplo o surgimento



de uma pandemia, tempestade, seca ou até mesmo de uma crise financeira.

3.4 Decisões automatizadas

Conforme apresentado por Laudon e Laudon (2014), um imenso volume de dados permite o desenvolvimento de sistemas de informação automatizados, os quais por meio de algoritmos específicos, possibilitam controlar de forma autônoma diversas funções. Esse fenômeno cria uma conexão entre o mundo digital e físico, impulsionada pela coleta e análise de dados em tempo real.

3.5 Novos planos de negócios

Com base em informações analisadas e armazenada em Big Data, as empresas podem desenvolver planos de negócios mais eficientes e atualizados, sobretudo em função das ações preditivas, evitando desperdícios, direcionando campanhas de *marketing*, entre muitos outros benefícios que são possibilitados (TAURION, 2013).

De modo geral, Laudon e Laudon (2014) acreditam que o conceito de Big Data melhora muitas funções dentro dos processos de negócios de uma empresa:

- *Otimização da venda cruzada*: também conhecida por *cross-selling*, é uma prática pela qual, na compra de um produto, o usuário se depara com *sugestões* para incrementar sua compra. Essas sugestões são geradas por meio de algoritmos de identificação de padrão comportamental de consumidores.
- *Análise comportamental*: redução de tempo perdido pelos consumidores dentro de uma loja sem comprarem nada. Os clientes geralmente circulam pelo interior da loja na busca pelo produto que querem comprar. Analisar esse comportamento

e criar recombinações de *layouts* por exemplo, podem resolver essa situação. Um típico exemplo, é manter a seção de carvão e produtos de churrasco, bem ao lado de um açougue dentro de um supermercado.


- *Marketing locacional*: rastreamento geográfico dos consumidores permite lhes oferecer produtos ou lojas específicas nas suas proximidades. É uma prática que, de acordo com Taurion (2013), faz com que em média 80% de clientes se tornam mais propensos a conhecer ou visitar uma loja, graças aos alertas de sugestões recebidos.

4. Infraestrutura para uso de Big Data

Laudon e Laudon (2014), ainda, trazem a infraestrutura contemporânea para o uso de Big Data, também conhecida por alguns autores como a inteligência empresarial (BI-Business Intelligence), como um conjunto de ferramentas que devem permitir a aquisição de informações a partir de todo e qualquer dado utilizado por uma empresa em Big Data semiestruturado ou não estruturado, tais como *data warehouses*, *data marts*, plataformas analíticas, computação em memória e *Hadoop*.

4.1 Datas Warehouses e Marts

A ferramenta Data Warehouse, tem se destacado nas últimas décadas no que diz respeito às análises de dados. Nada mais é do que um banco de dados recorrente potencial para as tomadas de decisões empresariais. A origem desses dados está nas transações operacionais da empresa (vendas, manufatura, produção, etc.) e, também, em fontes externas (LAUDON; LAUDON, 2014).



A data warehouse gera relatórios de gerenciamento por meio da combinação entre os dados internos e externos. Apenas após essa combinação, na qual dados imprecisos são corrigidos, é que ocorre o armazenamento. Todos esses dados (internos e sobretudo externos) vêm da Big Data (LAUDON; LAUDON, 2014). Segundo Taurion (2013), as características da data warehouse descrevem um sistema no qual os dados não podem ser alterados, acessáveis sim, mas editáveis não.

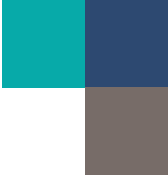
Há também a data mart, uma espécie de versão menor da data warehouse com um volume resumido ou selecionado de dados, colocado em um banco específico destinado a atender um setor ou departamento em especial (LAUDON; LAUDON, 2014).

4.2 Plataformas analíticas

Baseadas em tecnologia de informação relacional e não relacional (NoSQL), tem a função de analisar grandes grupos de dados e promover consultas, que segundo Laudon e Laudon (2014) são capazes de oferecer um processamento de 10 a 100 vezes mais rápido em comparação com os tradicionais sistemas de informação. As principais plataformas do mercado são a IBM Netezza® e a Oracle Big Data Appliance®.

4.3 Computação em memória

Outra ferramenta constituinte da infraestrutura de utilização de Big Data é a *in-memory* (computação em memória), baseada como o próprio nome sugere, na memória principal de um computador, a memória RAM, responsável por realizar o armazenamento de dados. O acesso aos dados da memória RAM, é muito mais rápido do que o acesso em armazenamento de discos rígidos dos tradicionais sistemas (o que está mudando com a crescente utilização de hardwares de armazenamento como o SSD–*Solid State Drive*) (LAUDON; LAUDON, 2014).



O interessante no armazenamento em memória é que sua capacidade é equivalente a um data mart ou até mesmo de um data warehouse de pequeno porte, mas com a vantagem de processar cálculos que antes demoravam horas, e que agora necessitam de poucos segundos (LAUDON; LAUDON, 2014).


Segundo Taurion (2013), Laudon e Laudon (2014), os dois principais produtos comerciais são o Oracle Exalytics® e a *High Performance Analytics Appliance* (HANA–Ferramenta de Alto Desempenho) da SAP. Ambos contam com banco de dados em memória que deve ser executado por hardwares otimizados.

4.4 Hadoop

Sendo uma tecnologia de destaque em meio ao uso de Big Data, o Hadoop necessita de um estudo maior. Originalmente criado em 2005 pela Yahoo para lidar com volumes de dados massivos e não estruturados, segundo Taurion (2013), o Hadoop é uma plataforma de código aberto voltada ao processamento paralelo de numerosos dados sem a necessidade de computadores caros. Basicamente ele trata um dado de Big Data e o fragmenta gerando nós de processamento os quais são combinados posteriormente para criar um conjunto de dados menor e de análise facilitada (LAUDON; LAUDON, 2014).

De forma prática, o Hadoop de acordo com Taurion (2013), é uma combinação de dois serviços básicos: o *Hadoop MapReduce* (HMR) para o processamento paralelo de alto nível; e, o *Hadoop Distributed File System* (HDFS) para o armazenamento dos dados.

Como exposto por Laudon e Laudon (2014), o HDFS é baseado em um serviço do Google denominado Google File System®. Este componente do Hadoop divide os dados em pequenos blocos e os

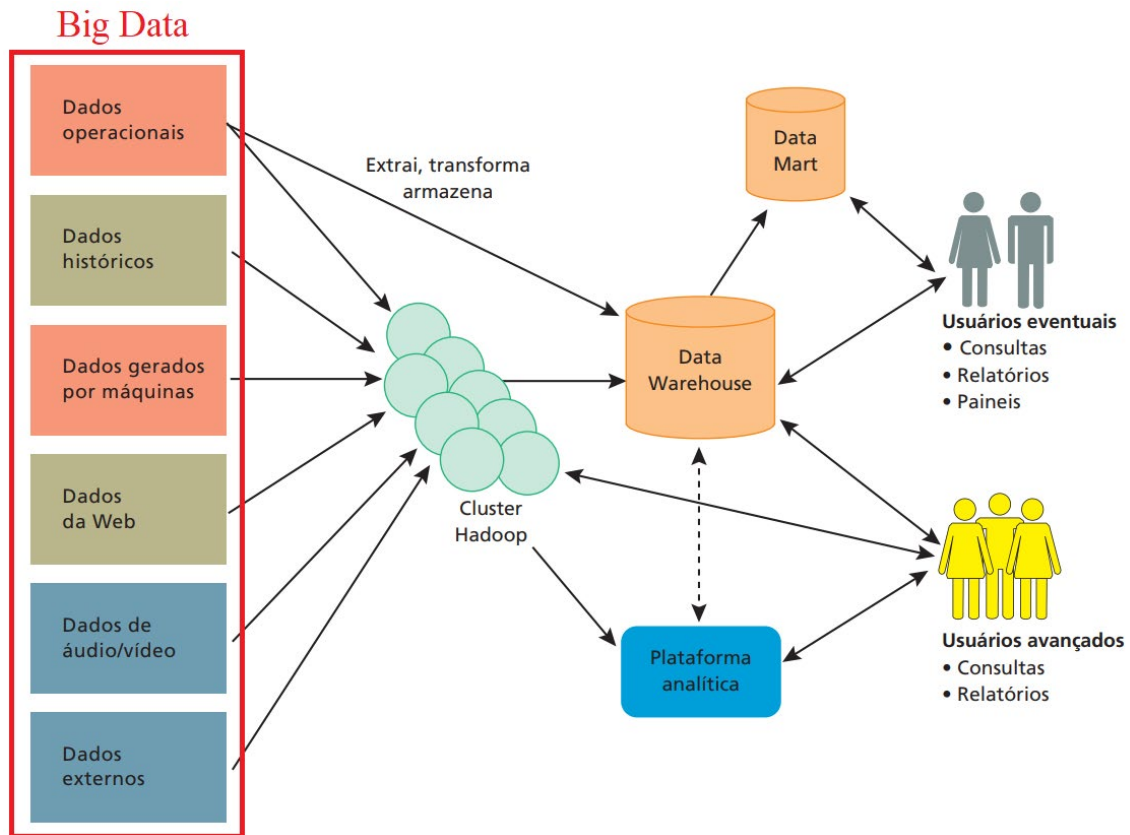


distribui para variados servidores. Por serem pequenos blocos de dados, seu processamento é de altíssima velocidade, e assim, ao invés dos dados de Big Data serem processados sequencialmente, eles passam a ser tratados paralelamente, ao mesmo tempo. Consequentemente não se exige uma grande capacidade computacional de servidores que podem então ser adquiridos por um custo menor. Segundo Taurion (2013), de forma técnica, cada bloco de dados geralmente é copiado para dois lugares diferentes, havendo então três lugares e uma garantia de que se um ou dois servidores falharem, haverá ainda um terceiro repositório. A critério de curiosidade, embora menciona-se que o Hadoop divide em *pequenos blocos*, esses blocos ainda são maiores do que os utilizados em bancos de dados relacionais, lembrando que o Hadoop trabalha com grandes volumes de dados (LAUDON; LAUDON, 2014).

O HMR também é um componente inspirado em um serviço do Google, no caso, o Google MapReduce®. Portanto, trata-se de uma ferramenta para mapear os dados convertendo-os em outros conjuntos de elementos fragmentados em Tuplas. O resultado desse mapeamento, segundo Taurion (2013), é então combinado em outros e menores conjuntos de Tuplas, obtendo o resultado desejado em uma consulta solicitada.

O Hadoop é utilizado geralmente pelas empresas para analisar Big Datas e promover um armazenamento intermediário por meio de seus *clusters* (conjunto de computadores) de servidores, antes dos dados serem armazenados em plataformas analíticas ou data warehouses. A Figura 1 apresenta as relações do Hadoop com as demais ferramentas constituintes da infraestrutura responsável em permitir ou auxiliar o uso de Big Datas.


Figura 1 – Infraestrutura de utilização de Big Data



Fonte: adaptada de Laudon e Laudon (2014).

Como observado na imagem anteriormente, Big Data compreende altos volumes de dados das mais variadas fontes: operacionais, históricas, de máquinas, na Web, de áudios e vídeos e de ambientes externos a uma empresa. Todos esses dados são encaminhados ao Hadoop para tratamento prévio antes de irem para outros sistemas/ferramentas (apenas os dados operacionais podem ser acessados diretamente pela data warehouse).

Por ser de código aberto, Taurion (2013) acredita que o Hadoop tem se proliferado por diversas empresas, assim como também muitas organizações passaram a desenvolvê-lo internamente. Então, mesmo que o código base seja o mesmo, haverá Hadoops diferentes em distintas distribuições (semelhante ao que acontece com sistemas operacionais baseados em Linux). Logo, é de extrema importância que você avalie essas distribuições afim de ser capaz de escolher a que lhe



trará benefícios e sucesso. Ainda, Taurion (2013) apresenta três critérios de seleção de um Hadoop:

- a. *Presença comercial*: verificar se a empresa responsável pela distribuição Hadoop possui condições de se manter operando. Vale a pena solicitar recomendações do uso da distribuição por outras empresas.
- b. *Oferta disponível*: saber quais recursos de arquitetura e funcionalidades a distribuição analisada oferece, bem como seu nível de integração, modelagem, performance, gerenciamento de *cluster* e suportes fornecidos. Importante observar a capacidade técnica oferecida pela equipe da empresa fornecedora.
- c. *Estratégia Principal*: deve-se primeiramente saber qual a finalidade a que será empregada a distribuição Hadoop e seu grau de importância nas estratégias empresariais de seus fornecedores. Prefere-se empresas cujo produto Hadoop é seu *carro chefe*.

4.4.1 Projeto baseado em Hadoop

Iniciar um projeto Hadoop no âmbito empresarial não é um processo simples e requer a passar por estágios de desenvolvimento para que não se transforme em um fracasso, descreditando a importância de Big Data para os processos de negócio de uma empresa. Portanto, apresento a você três etapas, que, segundo Taurion (2013), são fundamentais para confeccionar um projeto Hadoop de sucesso:

- I. Objetivos reais – identificar uma aplicação real para o Hadoop que agregará de fato, valor à empresa. Encontrar setores que irão tirar proveito verdadeiro de Big Data e consequentemente do Hadoop.
- II. Capacitação Requerida – Deve-se averiguar se na empresa há uma qualificação funcional para uso de Big Data. Esse profissional ou equipe profissional capacitada pode ser interna ou externa com ampla visão de negócios e de conhecimentos de Hadoop. Uma

equipe capacitada deve conter profissionais administradores de sistemas, desenvolvedores ou cientistas de dados.

- III. Distribuição – Conforme já mencionado, é importante selecionar uma distribuição Hadoop adequada, que garanta sobretudo suporte e qualificação necessária ao desenvolvimento do projeto.

5. E os Bancos de dados NoSQL?

Trata-se de um movimento que surgiu de forma paralela à Big Data, pela necessidade de resolver problemas não solucionados pelos tradicionais bancos de dados relacionais (TIWARI, 2011).

Segundo Tiwari (2011), o NoSQL é um banco de dados não relacional que também pode ser utilizado como pós relacional, com alta interatividade com Big Data. Conta com ferramentas próprias como MapReduce, de clusterização, de tolerância a falhas, de *schema-free* (organização de dados sem estruturação) e de *sharding* (fragmentação de dados). Além disso, segundo o autor, de forma a facilitar a seleção de qual tipo de banco NoSQL utilizar, esses bancos não relacionais podem ser divididos pela função que desempenham no tratamento dos dados. Os principais tipos são:

- *Key/Value Store* – Chave com valor responsável em carregar dados.
- *Wide Columns Store* – Foco no suporte de elevado número de Atributos (colunas) e Tuplas (linhas) de uma entidade (tabela) do banco de dados.
- *Document Store* – Direcionado à localização de arquivos XML ou JSON por meio de identificadores próprios.
- *Graph Store* – Com a função de armazenar objetos que são utilizados para buscar itens (dados).

- *Column Oriented Store* – Bancos relacionais, mas com características NoSQL para melhorar o armazenamento de dados em colunas.

Os Bancos NoSQL possuem diferentes formas e tamanhos dependendo das peculiaridades e necessidades exigidas por eventuais problemas empresariais. Ou seja, nada mais são do que alternativas para gerenciar volumes consideráveis de dados em computadores dispersos, de forma a possibilitar um aumento ou redução de máquinas, como foco principal na aceleração de consultas simples (LAUDON; LAUDON, 2014).

Portanto, de modo geral, utilizar-se de Big Data é garantir o armazenamento e gestão de enormes volumes de dados que são disponibilizados de forma simultânea e que precisam ser analisados em tempo real, frente aos processos e planos de negócio de uma empresa. Essa prática evita a ocorrência de dados imprecisos ou incompletos que podem criar problemas graves e decisões equivocadas; garantem a veracidade e alta velocidade de processamento de dados oriundo das mais variadas fontes possíveis e que agregaram alto valor às organizações. Tudo isso gerou mudanças nas estruturas organizacionais de empresas que passaram a desenvolver infraestruturas tecnológicas voltadas à Big Data, como a utilização de Hadoops. Dessa forma, o sucesso e os benefícios esperados com a utilização de Big Data que se faz necessária atualmente, tornam-se alcançáveis, desde que haja um planejamento de projeto adequado para esta finalidade.

Referências Bibliográficas

BARBIERI, C. **BI2-Business intelligence**: modelagem e qualidade. Rio de Janeiro: Digeratti, 2011

GOLDSCHMIDT, R. **Data mining**: conceitos, técnicas, algoritmos, orientações e aplicações. Rio de Janeiro: Elsevier, 2015.

LAUDON, K.; LAUDON, J. **Sistemas de informações gerenciais**. Tradução: Célia Taniwaki. 11. ed. São Paulo: Pearson, 2014.



MACHADO, F. N. R. **Projeto e implementação de banco de dados**. 4. ed. São Paulo: Érica, 2020.

TAURION, C. **Big Data**: velocidade, volume, variedade, veracidade, valor. Rio de Janeiro: Brasport, 2013.

TIWARI, S. **Professional NoSQL**. Indianapolis: John Wiley & Sons, 2011.



BONS ESTUDOS!