

Banco de Dados Relacional e Big Data

Banco de Dados Relacional e Big Data

Autor: Sergio Eduardo Nunes

Como citar este documento: NUNES, Sergio. *Banco de Dados*. Valinhos: 2016.

Sumário

Apresentação da Disciplina	03
Unidade 1: MER (Modelo Entidade Relacionamento)	05
Unidade 2: Chaves em Banco de Dados	29
Unidade 3: Cardinalidades	57
Unidade 4: Normalização (Formas Normais)	82
Unidade 5: Comandos DDL (<i>Data Definition Language</i>)	108
Unidade 6: Comandos DML (<i>Data Manipulation Language</i>)	132
Unidade 7: Comandos DCL (<i>Data Control Language</i>)	154
Unidade 8: Big DATA	173



Apresentação da Disciplina

Os bancos de dados estão presentes em grande parte do cotidiano das pessoas, no caixa do supermercado, no *menu* da TV a cabo, no *check in* do aeroporto ou hotel, entre diversos outros serviços que utilizamos. Por esses motivos, o entendimento das técnicas e conceitos acerca desse tema se torna tão relevante no tocante ao desenvolvimento web.


Tal é o grau de importância do banco de dados no desenvolvimento web que as empresas de hospedagem de sites disponibilizam de forma nativa um SGBD (Sistema de Gerenciamento de Banco de Dados). Porém, o planejamento e modelagem de um BD é um desafio!

Esta disciplina visa estruturar os conhecimentos acerca dos conceitos,

modelagens e instruções SQL para banco de dados do tipo relacional. Com isso, possibilitará que os desenvolvimentos web armazenem os dados, para a geração de consultas.

No primeiro tema, serão estudados os conceitos e aplicações dos componentes que estruturam o BD, além da modelagem de BD, a fim de que seja desenvolvida a habilidade para o planejamento de sistemas com a utilização de uma base de dados do tipo relacional.

A utilização das técnicas para modelagem faz com que o planejamento da estrutura do BD possa ser entendido de forma visual por meio do DER (Diagrama Entidade Relacionamento), possibilitando que os erros e ajustes possam ocorrer



antes da implementação do BD no sistema, além de permitir um efetivo entendimento estrutural pelos analistas em caso de manutenção ou ampliação das funcionalidades do sistema web.



Unidade 1

MER (Modelo Entidade Relacionamento)

Objetivos

1. Entender os conceitos e aplicação do BD em sistemas de desenvolvimento web.
2. Identificar as partes que compõem a estrutura do BD, como: entidade, relacionamento e atributos.
3. Modelar banco de dados por meio do DER (Diagrama de Entidade Relacionamento).
4. Utilizar o software brModelo, para planejar a estrutura e relacionamento das tabelas no BD.



Introdução

Devido ao aumento do acesso da população aos serviços de internet, por meio de computadores, *smartphones*, *tablets* e *smartvs*, os serviços e aplicações computacionais exigem que os profissionais de tecnologia da informação necessitem ter conhecimento em diversas áreas computacionais.

Para que os sistemas possam “abrigar” um banco de dados, é necessário que tenhamos um Sistema de Gerenciamento de Banco de Dados. Silberschatz et al. (2010, p. 1) definem que:



Os sistemas de banco de dados são projetados para gerenciar grandes blocos de informações. O gerenciamento de dados envolve definir estruturas para armazenamento de informação e fornecer mecanismos para manipulação de informações. Além disso, o sistema de gerenciamento de banco de dados precisa garantir que a segurança das informações armazenadas, apesar das falhas de sistemas ou acesso não autorizado.

Atualmente, o mercado oferece, de forma gratuita ou com licença paga, plataformas com uma gama de recursos que visam otimizar as funcionalidades de um sistema. Observe o quadro a seguir:




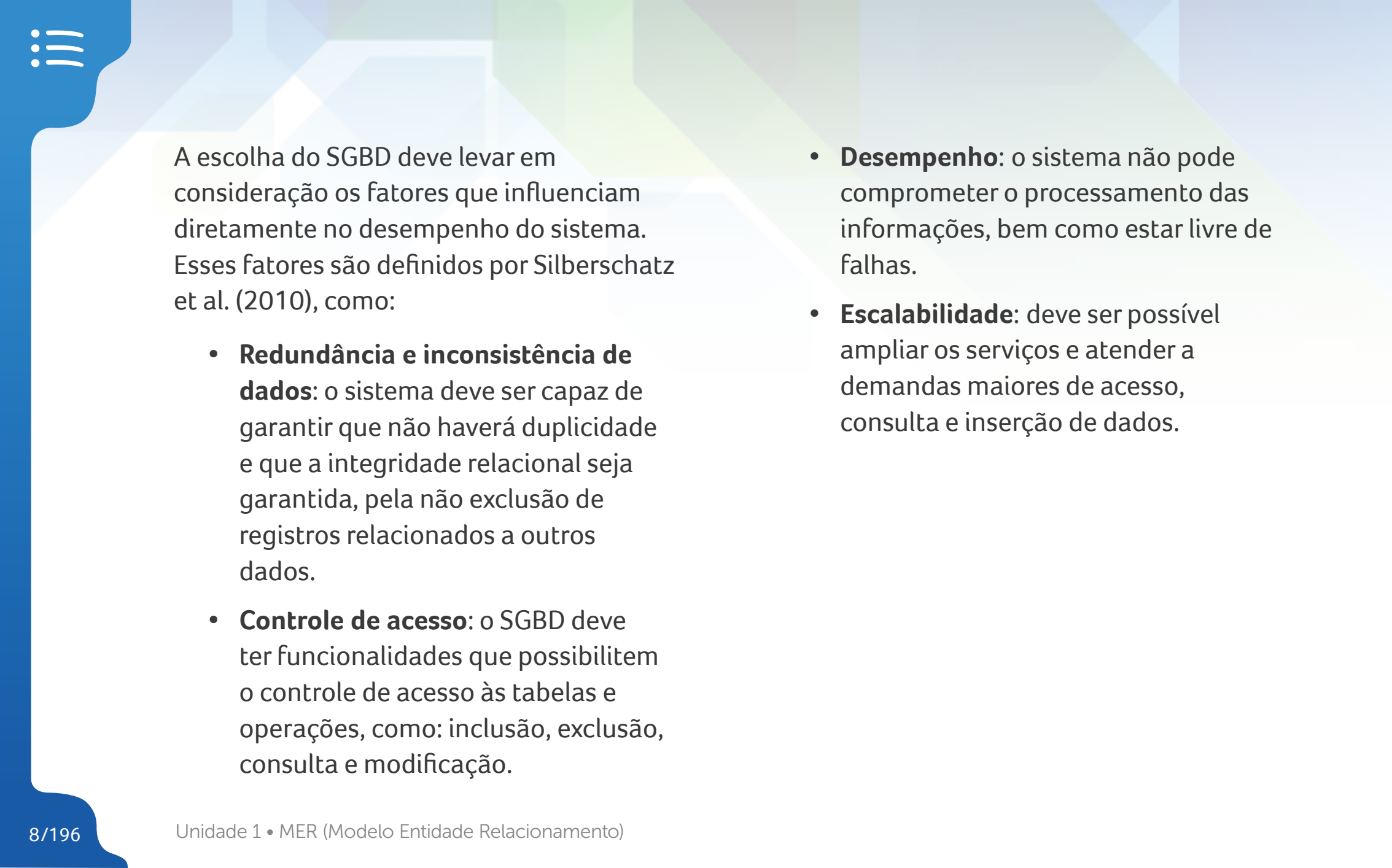
Quadro 1 – SGBDs e licenças

SGBD	LICENÇA
Oracle	Proprietária/Gratuita
SQL Server	Proprietária
Access	Proprietária
MySQL	Gratuita
IBM DB2	Proprietária
Firebird	Gratuita
PostgreSQL	Gratuita

Fonte: Elaborado pelo autor.

Para saber mais

As certificações mais procuradas pelos administradores de banco de dados são MCSA SQL Server (Nível básico Microsoft) e OCA (Nível básico MySQL), respectivamente referentes aos Sistemas de Gerenciamento de Banco de Dados: SQL Server e Oracle.



A escolha do SGBD deve levar em consideração os fatores que influenciam diretamente no desempenho do sistema. Esses fatores são definidos por Silberschatz et al. (2010), como:

- **Redundância e inconsistência de dados:** o sistema deve ser capaz de garantir que não haverá duplicidade e que a integridade relacional seja garantida, pela não exclusão de registros relacionados a outros dados.
- **Controle de acesso:** o SGBD deve ter funcionalidades que possibilitem o controle de acesso às tabelas e operações, como: inclusão, exclusão, consulta e modificação.
- **Desempenho:** o sistema não pode comprometer o processamento das informações, bem como estar livre de falhas.
- **Escalabilidade:** deve ser possível ampliar os serviços e atender a demandas maiores de acesso, consulta e inserção de dados.



1. MER (Modelo de Entidade Relacionamento)

Segundo Pichiliani (2012, p. 219):



O MER (Modelo Entidade Relacionamento), também conhecido apenas como modelo de dados, é a principal documentação de um banco de dados. São relacionadas às entidades, relacionamentos e atributos.

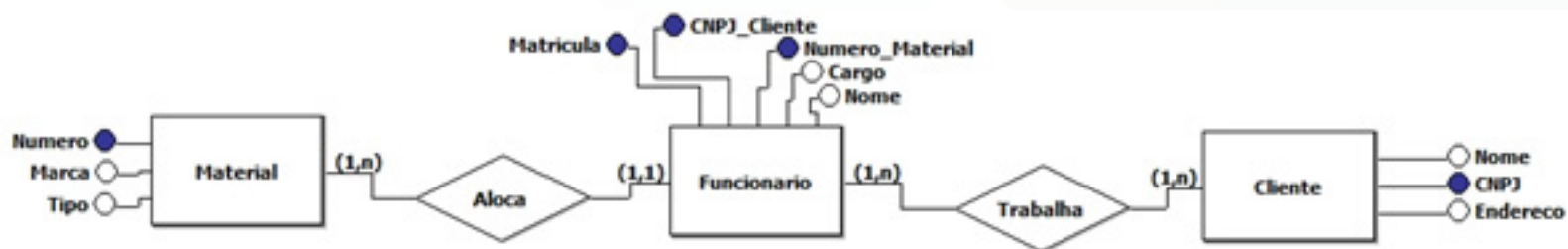
O Modelo de Entidade Relacionamento pode ser representado por meio das seguintes técnicas:

Para saber mais

Dentro de um projeto de desenvolvimentos de sistemas, a estrutura analítica do projeto (EAP), efetua o levantamento de requisitos, a fim de se conhecer as necessidades do cliente. Para documentar a estrutura e funcionamento do banco de dados, deve-se antecipadamente compreender as necessidades e funcionalidades que o sistema deve possuir.

- **DER (Diagrama Entidade Relacionamento):** esse diagrama representa de maneira visual os relacionamentos entre as entidades por meio de formas geométricas para representar as tabelas, campos e chaves. Observe a figura a seguir:

Figura 1 – Exemplo de DER



Fonte: Elaborada pelo autor.

O DER será o objeto de estudo desta aula, tratado com maiores detalhes mais à frente.

- **Mapeamento de banco de dados:** outra forma de modelar o BD é utilizar a técnica de mapeamento. Tal técnica permite que, de maneira descritiva, a modelagem possa ser compreendida pelos analistas e desenvolvedores. Observe o mapeamento do mesmo BD representado na Figura 1:

Material (Numero, Marca, Tipo)

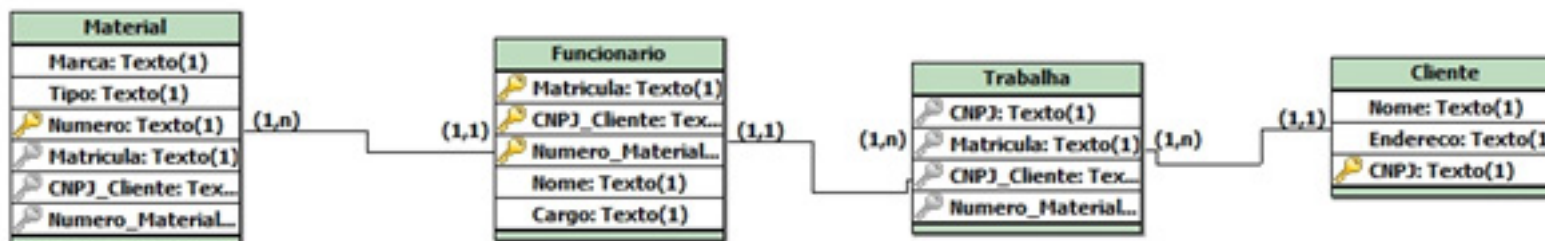
Cliente (CNPJ, Nome, Endereço)

Funcionario (Matricula, CNPJ_Cliente, Numero_Material, Nome, Cargo)


Onde as tabelas são representadas pelos nomes fora dos parênteses, e as colunas são representadas pelos nomes no interior dos parênteses, separados por vírgula.

- Modelo lógico: o modelo lógico permite maior nível de abstração do banco de dados a ser modelado. Essa técnica demonstra os relacionamentos entre as tabelas, os campos e seus respectivos tipos de dados. Observe a figura a seguir, na qual a Figura 2 é representada no modelo lógico:

Figura 2 – Exemplo de modelo lógico



Fonte: Elaborada pelo autor.




Os níveis de abstração fornecem aos desenvolvedores uma visão geral do BD, assim como permitem verificar os relacionamentos entre as tabelas e as suas respectivas chaves.

As modelagens desenvolvidas nesta aula devem ser feitas com o software brModelo. Para isso, siga os passos:

- Faça o download do software BRModelo;
- descompacte o arquivo;
- siga processo de instalação, da mesma forma como é feito com qualquer programa.

2. DER (Diagrama Entidade Relacionamento)

Conforme Silberschatz et al (2008), o DER foi desenvolvido pelo Dr. Peter Pin-Shan Chen, para facilitar o projeto de banco de dados, permitindo especificar um esquema que representa a estrutura lógica geral de um banco de dados, mostrando-se, assim, uma importante ferramenta para



modelagem e planejamento de banco de dados do tipo relacional.

Para saber mais

Para os gerentes de projeto, o DER serve para documentar a estrutura do BD. Muitos gerentes preferem utilizar a técnica de mapeamento de BD para descrever a estrutura e relacionamentos. Já para os desenvolvedores, o DER serve para planejar e estruturar as tabelas, os campos, chaves, cardinalidade e relacionamentos.

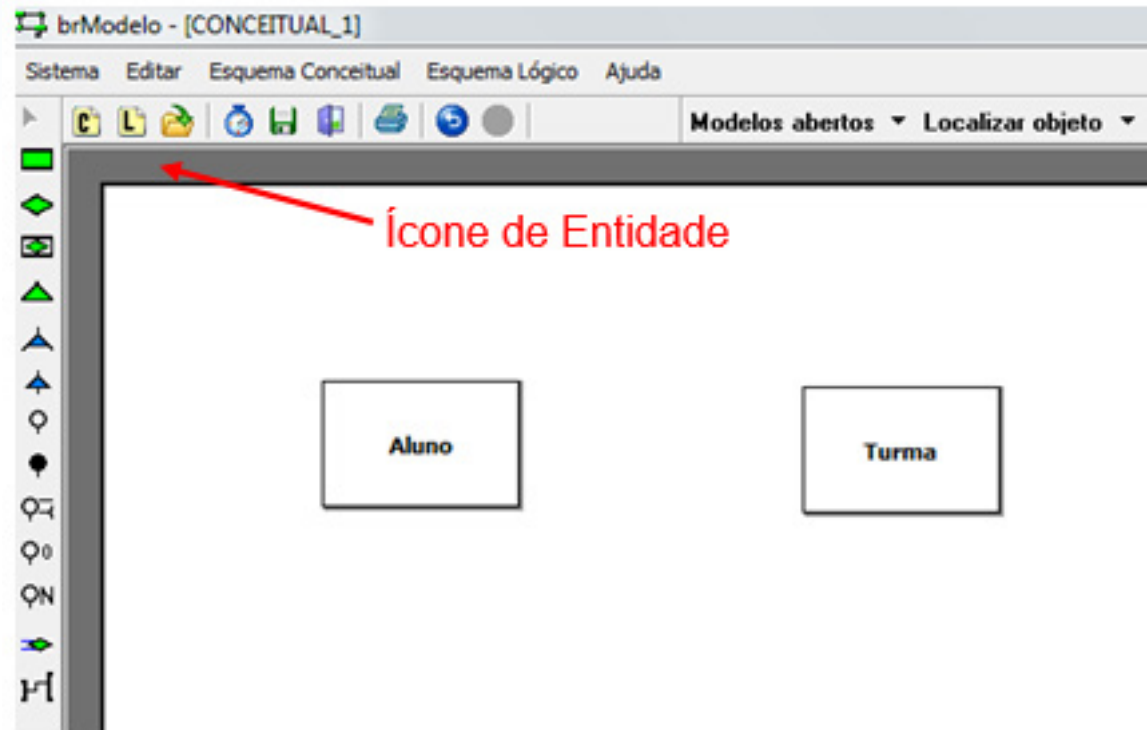
2.1 Entidade

As tabelas nos bancos de dados são chamadas de entidades. Segundo Heuser (2001, p. 34), “Entidade é um conjunto de

objetos da realidade modelada sobre os quais se deseja manter informações no banco de dados”.

No software brModelo, basta clicar no retângulo em verde na barra lateral esquerda e colocá-la na “área de modelagem”. Após isso, clique na entidade e renomeie com o nome que melhor representa o domínio da tabela. Observe a figura a seguir:

Figura 3 – Entidade no brModelo




Fonte: Elaborada pelo autor.

2.2 Relacionamento

Após a modelagem e definição das entidades que compõem o banco de dados serem definidas, é necessário planejar os relacionamentos das tabelas. Compreender essa técnica é parte importante na modelagem, pois isso pode ter impacto significativo quando forem geradas as consultas (Select no MySQL). Heuser (2008, p. 36) define que: “Relacionamento em banco de dados é o conjunto de associações entre as entidades”.

Para relacionar as tabelas no brModelo, siga os passos:

- Clique no símbolo , que se encontra na barra lateral esquerda;

- clique em cima das duas entidades, na qual se deseja manter um relacionamento no BD;
- clique no losango para renomeie o relacionamento; para isso, utilize verbos como: possui, contém, atrela, relaciona, está etc.

Observe a figura a seguir que representa o relacionamento entre duas entidades:

Figura 4 – Relacionamento no brModelo



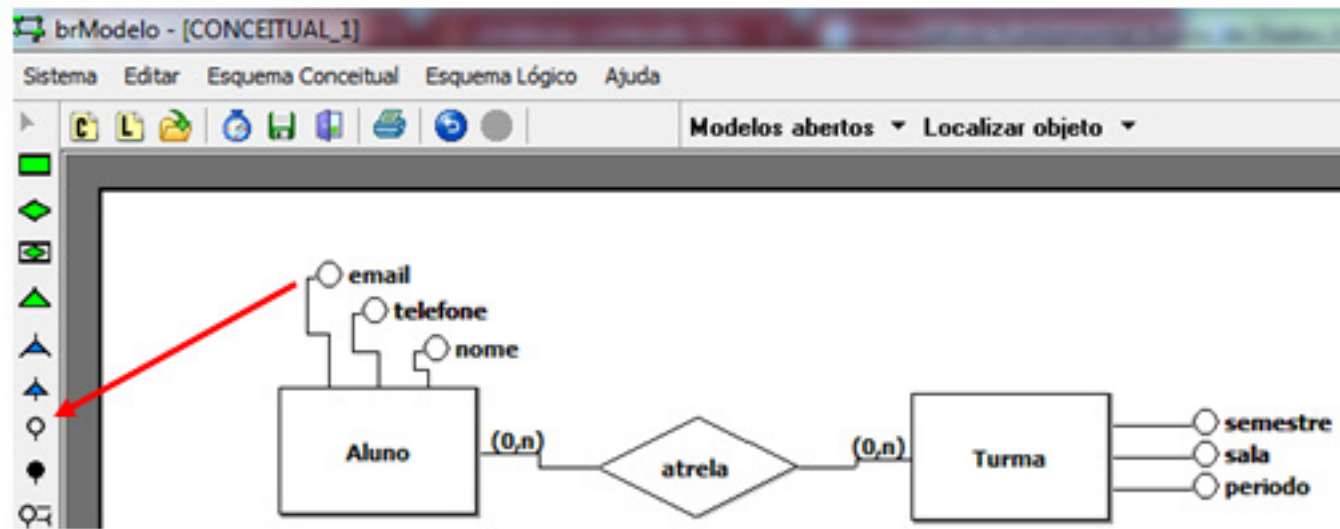
Fonte: Elaborada pelo autor.

2.3 Atributo

Os campos existentes em uma entidade são denominados atributo, representado nas tabelas pelas colunas. Para Heuser (2008, p. 55), “Dado que é associado a cada ocorrência de uma entidade ou de um relacionamento”.

Para adicionar os atributos no brModelo, clique no local indicado na Figura 5 e, em seguida, nomeie o atributo, conforme segue:


Figura 5 – Atributo no brModelo



Fonte: Elaborada pelo autor.

Os nomes dos atributos não podem conter:

- Nomes compostos: data nascimento, nome do funcionário, valor produto etc.
- Começar com números: 1nota, 3medida, 1ano etc.
- Começar com caracteres: @nome, \$idade, &altura etc.



Uma boa prática é começar com letras minúsculas, utilizar *underline* nas palavras compostas (nota_1) e utilizar nomes curtos que descrevam o dado que o campo vai receber (data_nasc).



Glossário

MER: Modelo de Entidade Relacionamento, técnica utilizada para representar a estrutura e planejamento dos bancos de dados.

DER: Diagrama de Entidade Relacionamento, representação do banco de dados, por meio de figuras geométricas.

Modelagem de BD: Técnica utilizada para auxiliar os profissionais de banco de dados no planejamento.

brModelo: programa utilizado para fazer a modelagem do banco de dados.



Questão para reflexão

Uma empresa personaliza camisetas, que são comercializadas somente no balcão da loja. O gerenciamento dos pedidos, fornecedores e clientes é feito em uma caderneta. Devido ao aumento de pedidos, o proprietário deseja um sistema de gerenciamento computacional. Como o poderia ser modelado o DER?





Considerações Finais

- Os conceitos e aplicações do Modelo de Entidade Relacionamento;
- Os tipos de modelagens para representação e planejamento de banco de dados;
- O desenvolvimento do DER (Diagrama Entidade Relacionamento) com o software brModelo.
- Compreensão dos componentes que compõem um banco de dados (Entidade, atributo e relacionamento).



Referências

HEUSER, C. A. **Projeto de Banco de Dados**. 4. ed. Porto Alegre: Instituto de Informática da UFRGS; Sagra Luzzato, 2008. p. 34; 36; 55.

PICHILIANI, M. **Conversando sobre Banco de Dados**. São Paulo: Clube dos Autores, p. 219, 2012.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. 6. ed. Rio de Janeiro: Campos: Elsevier, 2012.



Questão 1

1. Observe as afirmativas a seguir:

I – O MER (Modelo Entidade Relacionamento) é uma forma documental que permite, por meio de suas técnicas, modelar um BD.

II – As entidades e atributos definem a estrutura do Sistema de Gerenciamento de Banco de Dados (SGBD).

III – Os relacionamentos no DER devem vir acompanhados por números e caracteres.

Assinale a alternativa correta:

- a) Está correta apenas a alternativa II.
- b) Estão corretas as alternativas I e II.
- c) Está correta apenas a alternativa I.
- d) Estão corretas as alternativas I, II e III.
- e) Nenhuma alternativa está correta.



Questão 2

2. Assinale a alternativa correta:

- a) Banco de dados guarda informações.
- b) Os atributos são as tabelas do BD.
- c) Podem-se utilizar quantos atributos forem necessários.
- d) Os relacionamentos entre as tabelas são opcionais.
- e) A redundância é benéfica, pois garante os dados em mais de um lugar.



Questão 3

3. Observe os nomes dos atributos e assinale (C) para correto e (E) errado, nas afirmações a seguir:

() nome do dependente.

() nota1

() media_2

() %sobrenome

() cod_postal

Assinale a alternativa CORRETA:

a) F – V – F – V – F.

b) V – V – V – F – V.

c) F – F – F – V – V.

d) V – F – V – F – V.

e) F – V – V – F – V.



Questão 4

4. Observe a citação de Silberschatz et al (2010, p. 1), que define Sistema de Gerenciamento de Banco de Dados como:

Os sistemas de banco de dados são projetados para gerenciar grandes blocos de _____. O gerenciamento de dados envolve definir estruturas para _____ de informação e fornecer mecanismos para manipulação de informações. Além disso, o sistema de gerenciamento de banco de dados precisa garantir que a _____ das informações armazenadas, apesar das falhas de sistemas ou acesso não autorizado.

Assinale as alternativas que complete as lacunas:

- a) informações / manipulação / segurança
- b) dados / manipulação / coluna
- c) informações / armazenamento / segurança
- d) tabelas / BD / coluna
- e) dados / gerenciamento / segurança



Questão 5

5. Assinale a afirmação correta:

- a) Os bancos de dados são utilizados somente para o desenvolvimento web, pois necessitam de maior disponibilidade.
- b) Os SGBDs só podem ser instalados em servidores locais.
- c) Os bancos de dados estão presentes em vários serviços utilizados no cotidiano das pessoas.
- d) Somente os SGBDs de licenças pagas possuem um gerenciamento consistente.
- e) Os bancos de dados geram dados para consultas posteriores.



Gabarito

1. Resposta: C.

As entidades e atributos modelam a estrutura do banco de dados; o nome dos atributos segue regras, não podendo ter nomes compostos ou começar com números e caracteres.

2. Resposta: C.

Banco de dados guarda dados; as entidades são as tabelas; os relacionamentos são obrigatórios para se ter um BD do tipo relacional; redundância é a causa principal de inconsistência em base de dados.

3. Resposta: E.

Não se deve utilizar nomes compostos, iniciados com números ou caracteres.

4. Resposta: C.

A afirmação de Silberschatz et al diz que “Os sistemas de banco de dados são projetados para gerenciar grandes blocos de informações. O gerenciamento de dados envolve definir estruturas para armazenamento de informação e fornecer mecanismos para manipulação de informações. Além disso, o sistema de gerenciamento de banco de dados precisa garantir que a segurança das informações armazenadas, apesar das falhas de sistemas ou acesso não autorizado

5. Resposta: C.

Os SGBDs e BDs podem ser utilizados tanto em servidores local, web e cloud; BD geram informações.



Unidade 2

Chaves em Banco de Dados

Objetivos

1. Compreender os conceitos e utilidade das chaves dentro de um banco de dados do tipo relacional.
2. Identificar quais chaves podem ser utilizadas como chaves primária ou alternativa dentro de uma entidade.
3. Entender como se utiliza e nomeia uma chave estrangeira nas entidades.
4. Modelar banco de dados do tipo relacional no software brModelo.



Introdução

Na aula passada, você estudou os conceitos fundamentais, aplicações e os componentes básicos de banco de dados. Mas você deve ter percebido que as técnicas utilizadas para modelagem não relacionaram as tabelas entre si.

O MER (Modelo de Entidade Relacionamento) descreve como as entidades irão se relacionar nos bancos de dados, seja por meio do DER (Diagrama de Entidade Relacionamento), pelo modelo lógico ou, ainda, pelo mapeamento. Essas técnicas permitem que de maneira visual seja planejado os nomes das chaves e o relacionamento das tabelas, para posterior desenvolvimento no SGBD.


Grande parte dos problemas encontrados ao se desejar obter informações dos

bancos de dados está relacionada à forma de planejamento, estruturação e relacionamento das entidades em um BD, utilizado no desenvolvimento web.

Para saber mais

O DBA (Data Base Administrator), administrador de banco de dados, é o profissional responsável pela instalação, planejamento, backup e gerenciamento do banco de dados. Nos sites de vagas para empregos, sempre figura entre os profissionais de T.I. mais requisitados.

Por isso, o entendimento dos conceitos e aplicações das chaves nos BDs faz-se necessário, para que seja garantido o relacionamento entre as tabelas. As



técnicas para determinar as chaves (primária, estrangeira ou alternativa) serão no DER, por meio do software brModelo, a fim de se estruturar a modelagem de forma visual, facilitando, assim, a compreensão do domínio em que se deseja desenvolver o sistema e gerar consultas.

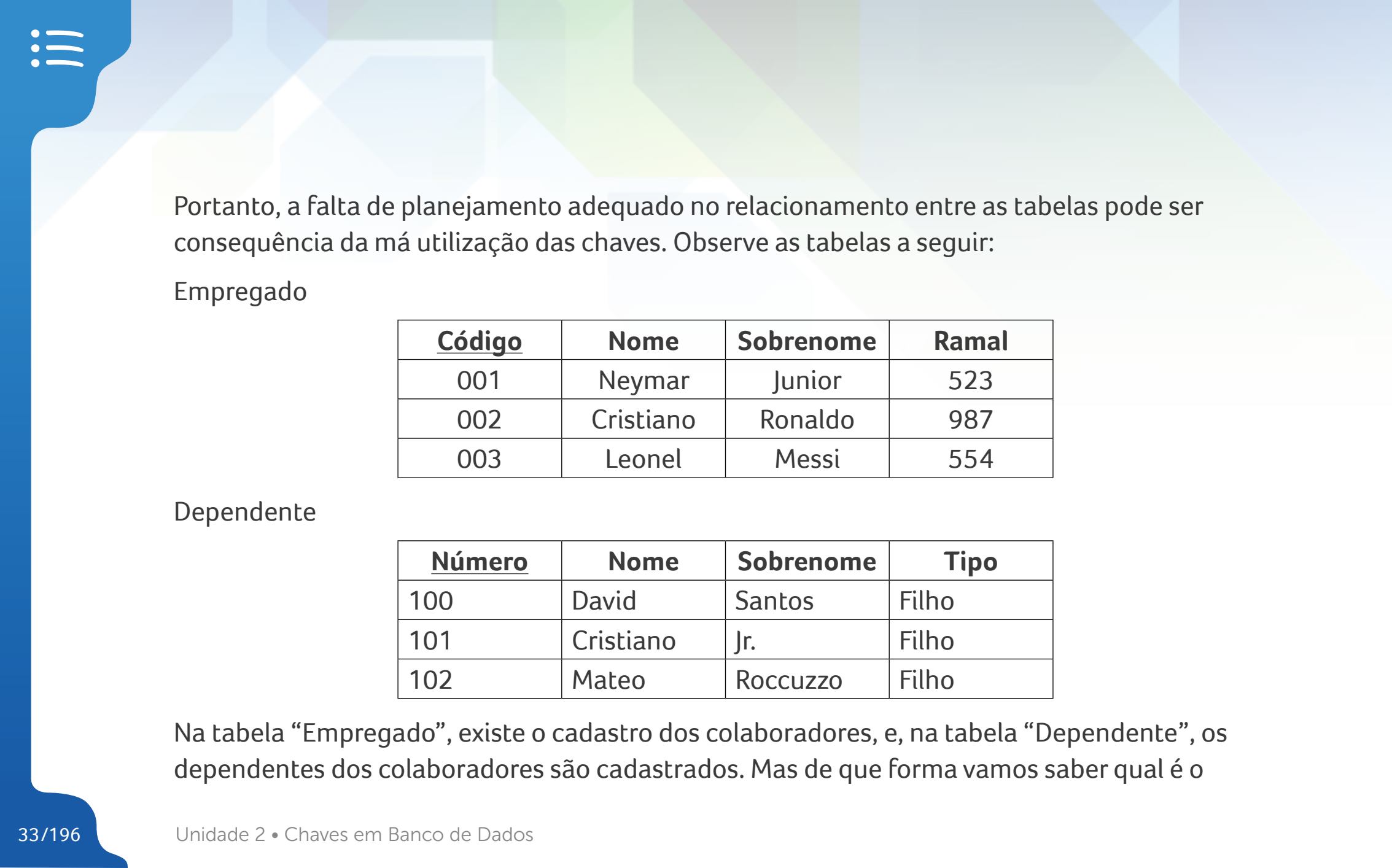
As chaves estão entre os conceitos básicos de banco de dados, a fim de se definir a estrutura das tabelas e os seus relacionamentos. Na fase de modelagem e planejamento dos BDs, as definições desses campos requerem que o desenvolvedor tenha compreensão dos requisitos necessários no sistema. Pois, segundo Ferrari (2007, p. 24):



Cada registro dentro de uma tabela deve ser unívoco, isto é, cada linha da tabela deve ser diferente, em pelo menos uma coluna das demais. Registros idênticos dentro de uma tabela causariam ambiguidade e redundância inúteis e, portanto, devem ser evitados.

Para saber mais

Todas as fases da modelagem e planejamento para implementação do banco de dados devem ser documentadas. Isso permite que, no futuro, as atividades e alterações necessárias possam ser realizadas de forma mais organizada e estruturada.



Portanto, a falta de planejamento adequado no relacionamento entre as tabelas pode ser consequência da má utilização das chaves. Observe as tabelas a seguir:

Empregado

<u>Código</u>	Nome	Sobrenome	Ramal
001	Neymar	Junior	523
002	Cristiano	Ronaldo	987
003	Leonel	Messi	554

Dependente

<u>Número</u>	Nome	Sobrenome	Tipo
100	David	Santos	Filho
101	Cristiano	Jr.	Filho
102	Mateo	Roccuzzo	Filho

Na tabela “Empregado”, existe o cadastro dos colaboradores, e, na tabela “Dependente”, os dependentes dos colaboradores são cadastrados. Mas de que forma vamos saber qual é o



dependente de cada colaborador?

Para resolver esse problema, será necessário entender os conceitos e aplicações de:

- chave primária;
- chave estrangeira;
- chave alternativa.


Sem esses conceitos das chaves no banco de dados relacional, não seria possível que pudéssemos relacionar as tabelas.

1. Chave Primária (Primary Key)

Você já deve ter percebido que alguns números utilizados no nosso dia a dia servem como chave para nos identificar, sendo os mais comuns: RA, RG, CPF, título eleitoral, habilitação, entre outros utilizados em cadastros em sites ou comércio.

Para saber mais

Sempre coloque as chaves primárias como primeiro campo das suas tabelas. Essa prática permite que você e outros desenvolvedores identifiquem de imediato qual é a primeira opção de consulta aos registros.



Nas tabelas, existe uma chave que deve ser a principal dentro da tabela, a chave primária (PK). Conforme Souza (2015, p. 74):



A chave primária (PK) é utilizada para especificar que cada linha é única, ou seja, esse campo não pode ter dois ou mais registros de mesmo valor e também não pode conter registros nulos.

Para o exemplo das tabelas utilizadas na situação-problema entre empregado e dependente, temos que:

Empregado

Código	Nome	Sobrenome	Ramal
001	Neymar	Junior	523
002	Cristiano	Ronaldo	987
003	Leonel	Messi	554

Dependente

<u>Número</u>	Nome	Sobrenome	Tipo
100	David	Santos	Filho
101	Cristiano	Jr.	Filho
102	Mateo	Roccuzzo	Filho

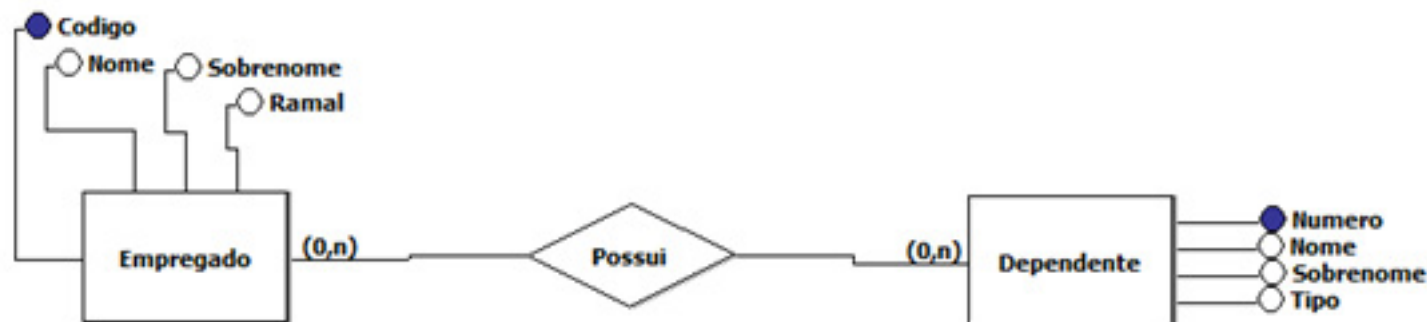
Com isso, temos que as tabelas:

- Empregado: utilizou o atributo “Código” para chave primária.
- Dependente: utilizou o atributo “Número” para chave primária.

Atenção! Para determinar as chaves primárias, utilize aqueles campos que receba números inteiros e/ou caracteres, como, por exemplo: matrícula CPF, RG, número, Id, código etc.

Para modelar as chaves primária no brModelo, observe a figura a seguir:

Figura 1 – Chave primária na modelagem do brModelo.

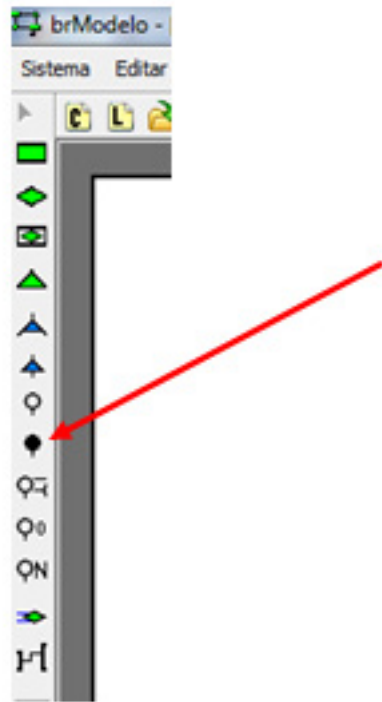


Fonte: Elaborada pelo autor.

Para colocar a chave primária no brModelo, deve-se seguir as seguintes etapas:

- Clique no ícone de atributo com bolinha pintada, na barra lateral. Conforme demonstrado na Figura 2:

Figura 2 – Ícone de chave primária no brModelo



Fonte: Elaborada pelo autor.

Clique na entidade que deseja adicionar a chave primária;

clique na chave estrangeira adicionada e dê nome à chave.

2. Chave Alternativa

Para você entender onde encontramos a chave alternativa, imagine a seguinte situação: no dia da eleição, você foi votar, mas para a sua surpresa o seu título eleitoral não está na sua carteira. E, agora, você não vai poder votar?

Vai sim, sabe por quê? Por causa da chave alternativa.

No caso supracitado, a chave primária é o número do título eleitoral e o número do RG (Registro Geral) é a chave alternativa. Essa chave possibilita que o mesário

identifique qual a zona e seção você deve se dirigir para votar e qual o número do seu título de eleitor.

Para saber mais

Embora em alguns casos a chave alternativa seja indispensável, para que sejam encontrados os registros nos bancos de dados, essa chave deve ser utilizada SE E SOMENTE SE for necessária para garantir o acesso às informações. Não utilize como redundância.

Segundo Heuser (2001), as chaves alternativas são uma segunda opção para distinguir das demais linhas de uma tabela de forma unívoca.

Para o exemplo das tabelas utilizadas na situação-problema, é conveniente que na tabela empregado tenhamos uma chave alternativa. Observe:

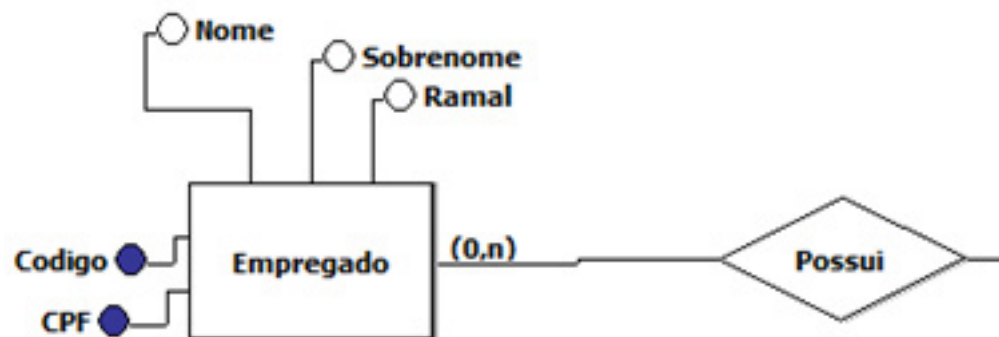
Empregado

<u>Código</u>	Nome	Sobrenome	Ramal	<u>CPF</u>
001	Neymar	Junior	523	123.456.789-10
002	Cristiano	Ronaldo	987	987.654.321-99
003	Leonel	Messi	554	555.444.333-22

Repare que ao final da tabela foi adicionada a coluna CPF como chave alternativa. Primariamente, para se localizar um registro, usa-se a chave primária, nos demais casos, tem-se a possibilidade de utilizar a chave alternativa.

Para representar o DER no brModelo, deve ser utilizado o mesmo ícone (atributo com bolinha pintada), como representado a seguir:

Figura 3 – Chave alternativa na modelagem do brModelo



Fonte: Elaborada pelo autor.

3. Chave Estrangeira

Observe que, até o momento, as tabelas não foram relacionadas por meio de alguma chave. No exemplo utilizado, não seria possível saber qual funcionário é responsável por um dado dependente.

Heuser (2001) define que a chave estrangeira é uma coluna em que o valor obrigatoriamente deve estar presente como chave primária em outra tabela, permitindo, assim, que ocorra o relacionamento entre as tabelas.

No exemplo que utilizamos anteriormente, as tabelas ficam da seguinte forma:

Empregado

<u>Código</u>	Nome	Sobrenome	Ramal	<u>CPF</u>
001	Neymar	Junior	523	123.456.789-10
002	Cristiano	Ronaldo	987	987.654.321-99
003	Leonel	Messi	554	555.444.333-22

Dependente

<u>Número</u>	Nome	Sobrenome	<u>Código_Empregado</u>	Tipo
100	David	Santos	001	Filho
101	Cristiano	Jr.	002	Filho
102	Mateo	Roccuzzo	003	Filho

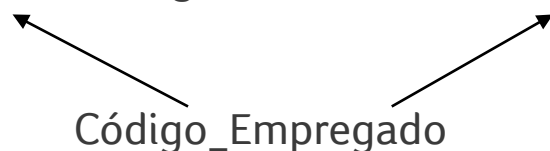
Na tabela “Dependente”, foi adicionada a coluna “Código_Empregado”, sendo que isso possibilitou o relacionamento entre as duas colunas. Para utilizar a chave estrangeira:

- identifique a chave primária da tabela que se deseja efetuar o relacionamento;
- adicione uma coluna para utilizar como relacionamento;
- nomeie a coluna como:

NomeChaveEstrangeira_NomeTabelaChaveEstrangeira.

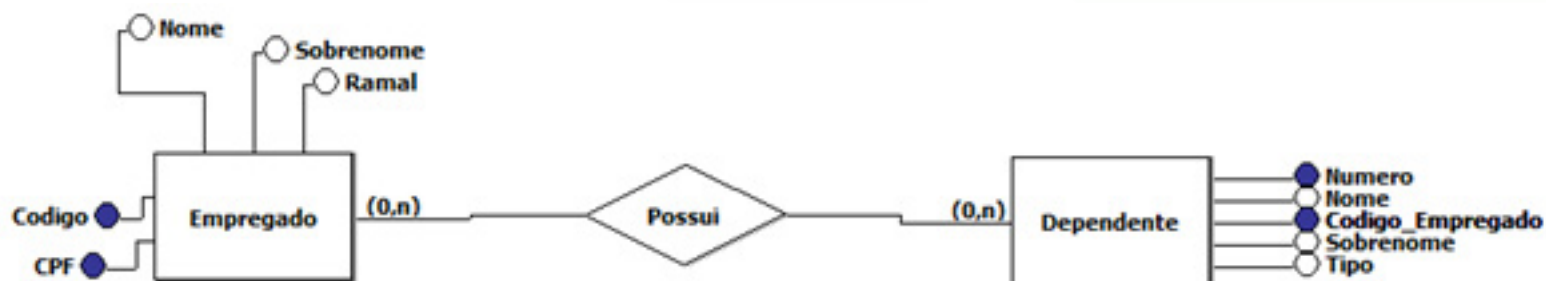
No exemplo anterior, tem-se que:

NomeChaveEstrangeira_NomeTabelaChaveEstrangeira



Para representar as tabelas no brModelo, você deve utilizar o mesmo ícone da chave primária (atributo com bolinha pintada) . Na modelagem, o que diferencia a chave primária da chave estrangeira é a forma como a chave estrangeira é nomeada. Conforme pode ser observado na modelagem a seguir:

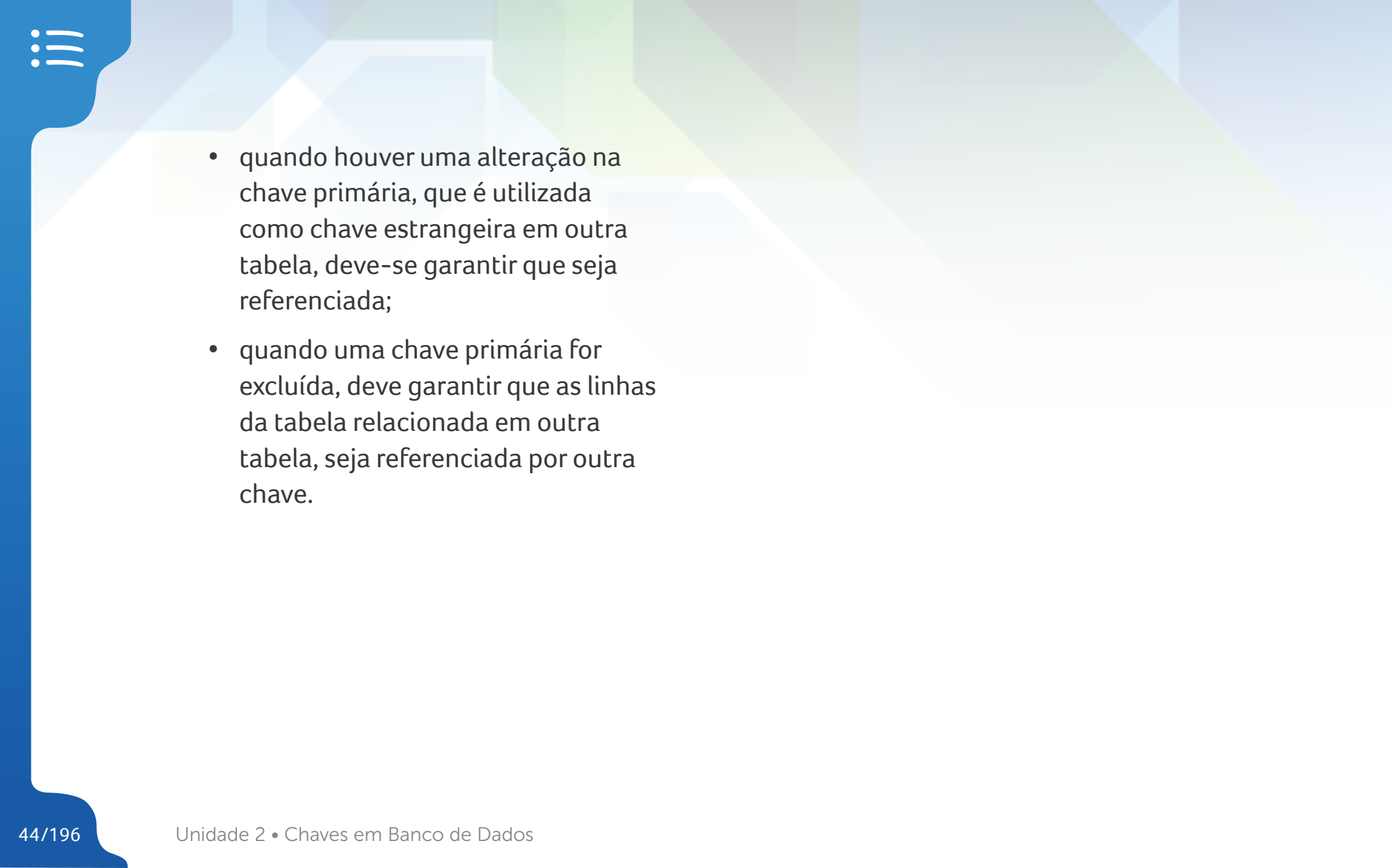
Figura 4 – Chave estrangeira na modelagem do brModelo



Fonte: Elaborada pelo autor.

Para garantia da integridade da utilização das chaves nos bancos de dados, conforme Heuser (2001):

- quando adicionada uma linha que contenha uma chave estrangeira, deve-se ter a garantia que ela esteja na chave primária referente;

- 
- quando houver uma alteração na chave primária, que é utilizada como chave estrangeira em outra tabela, deve-se garantir que seja referenciada;
 - quando uma chave primária for excluída, deve garantir que as linhas da tabela relacionada em outra tabela, seja referenciada por outra chave.



Glossário

Chave primária: atributo utilizado para identificar de maneira unívoca os registros de uma tabela.

Chave estrangeira: atributo utilizado para efetuar o relacionamento entre duas ou mais tabelas.

Chave alternativa: atributo secundário utilizado para identificar os registros de uma tabela.

Modelagem de banco de dados: técnica utilizada para efetuar o planejamento do banco de dados.



Questão para reflexão

Uma pizzaria encomendou um sistema para gerenciamento de suas vendas, e o gerente de projetos designou você para fazer o desenvolvimento da modelagem do banco de dados. O banco de dados deve possuir os seguintes requisitos:

- Possibilidade de cadastrar e consultar os dados dos clientes e pizzas do cardápio;
- Possibilidade de efetuar as vendas e cadastrá-las;
- Escolher os atributos adequados para as tabelas;
- Definir as chaves (primária, estrangeira e, se precisar, alternativa).





Considerações Finais

- Os conceitos e aplicações das chaves no banco de dados;
- O método para relacionar as tabelas, com a utilização da chave primária;
- O entendimento da nomeação da chave estrangeira, para que ocorra o relacionamento entre as tabelas;
- O planejamento do banco para garantir a integridade.



Referências

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

HEUSER, C. A. **Projeto de Banco de Dados**. 4. ed. Porto Alegre: Instituto de Informática da UFRGS; Sagra Luzzato, 2008.

SOUZA, T. H. SQL: **Avançado e Teoria Relacional**. Rio de Janeiro: Ciência Moderna, 2013.



Questão 1

1. Assinale a alternativa correta quanto à função das chaves no banco de dados:

- a) A chave primária deve ser utilizada para destacar a tabela mais importante dentro do banco de dados.
- b) A chave alternativa pode servir como uma opção para encontrar um registro dentro de um SGBD.
- c) O relacionamento entre duas tabelas ocorre quando qualquer um dos atributos é colocado na outra tabela.
- d) O SGBD deve garantir que as tabelas se relacionem, sem a necessidade de se utilizar chaves.
- e) O relacionamento entre as tabelas é feito com a chave primária de uma tabela com a chave estrangeira de outra tabela.



Questão 2

2. Assinale verdadeiro ou falso nas afirmações a seguir:

- () Quanto mais chaves primárias a tabela possuir, melhor os dados são identificados.
 - () As chaves alternativas podem ser relacionadas a outras tabelas, servindo como chave estrangeira.
 - () A chave alternativa é obrigatória nas tabelas.
 - () A chave estrangeira é obrigatória em uma tabela.
 - () Para que haja o relacionamento entre as tabelas, é necessária a utilização de chaves.
- a) F – F – V – F – F.
 - b) V – F – V – F – V.
 - c) F – V – V – F – V.
 - d) F – V – F – F – V.
 - e) V – V – V – F – F.



Questão 3

3. Observe a frase a seguir e assinale a alternativa que complete as lacunas:

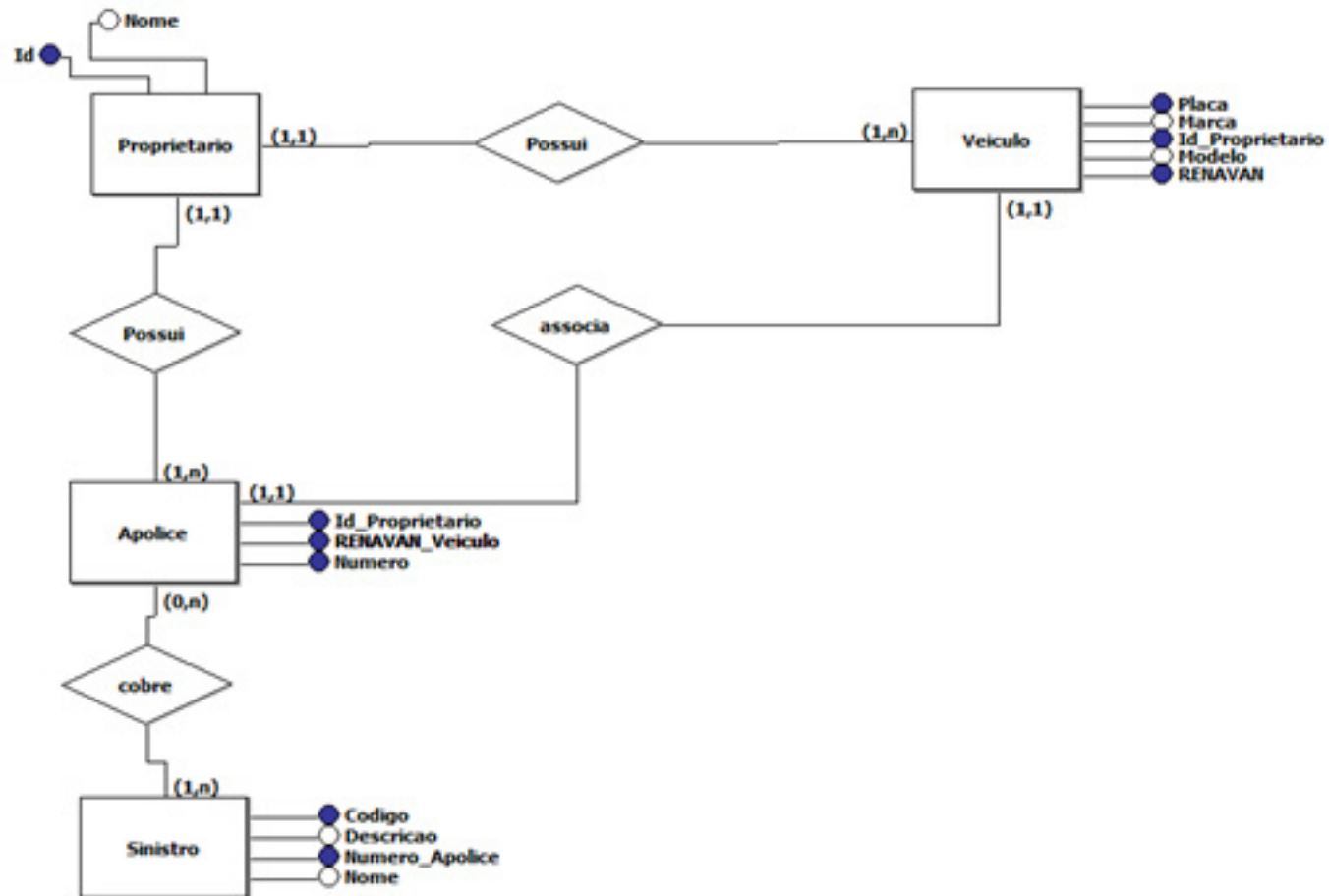
“[...] a chave _____ é uma coluna, em que o valor obrigatoriamente tem que estar presente como chave _____ em outra tabela, permitindo assim que ocorra o relacionamento entre as _____.”

- a) alternativa / estrangeira / tabelas
- b) estrangeira / alternativa / chaves
- c) primária / alternativa / chaves
- d) estrangeira / primária / tabelas
- e) primária / relacionada / chaves



Questão 4

4. Observe o DER a seguir e assinale alternativa CORRETA:





Questão 4

- a) A chave “Numero_Apolice” na tabela “Sinistro” é uma chave primária.
- b) A chave “Id_Proprietario” na tabela “Apolice” é uma chave estrangeira da tabela.
- c) A tabela “veiculo” não possui chave primária.
- d) A chave “Id” na tabela “Proprietário” é uma chave alternativa.
- e) A chave “Codigo” na tabela “Sinistro” é uma chave estrangeira.



Questão 5

5. Analise as afirmativas a seguir e assinale a alternativa CORRETA:

- I. Não existe padrão para nomear as chaves dentro de uma tabela, e sim boas práticas.
 - II. As chaves alternativas são formas de se encontrar um registro, quando não se conseguiu utilizar a chave primária por algum motivo.
 - III. As chaves primária, estrangeira e alternativa não são obrigatórias e não comprometem o desempenho no banco de dados.
- a) Estão corretas as alternativas I e II.
 - b) Somente a alternativa II está correta.
 - c) As alternativas I e III estão corretas.
 - d) Somente a alternativa III está correta.
 - e) Todas as alternativas estão corretas.



Gabarito

1. Resposta: E.

A chave primária é utilizada para encontrar um registro de maneira unívoca; a chave alternativa é uma opção para encontrar um registro no banco de dados; o relacionamento entre as tabelas deve ser feita com a chave estrangeira.

2. Resposta: C.

As tabelas possuem apenas uma chave primária; as chaves alternativas não são obrigatórias; a chave estrangeira é obrigatória somente nas tabelas que necessitam serem relacionadas.

3. Resposta: D.

Heuser (2001) define que a chave estrangeira é uma coluna, em que o valor obrigatoriamente deve estar presente como chave primária em outra tabela, permitindo, assim, que ocorra o relacionamento entre as tabelas.

4. Resposta: B.

A principal característica das chaves estrangeiras no DER e banco de dados é serem nomeadas como: NomeChaveEstrangeira_NomeTabelaChaveEstrangeira.



5. Resposta: A.

A utilização da chave primária garante a integridade do BD; a chave estrangeira só é obrigatória no relacionamento de tabelas; a chave alternativa não é obrigatória no BD.



Unidade 3

Cardinalidades

Objetivos

1. Compreender os conceitos e aplicações dos relacionamentos, mínimo, máximo, opcional e obrigatório entre as tabelas;
2. identificar quais tabelas devem receber as chaves estrangeiras, nos relacionamentos;
3. modelar banco de dados do tipo relacional no software brModelo.



Introdução

Na aula passada, as técnicas para relacionar as tabelas dentro do banco de dados possibilitaram o desenvolvimento de BDs do tipo relacional. Porém, neste momento, surge mais uma dúvida: como fazemos para determinar qual a tabela que vai receber a chave estrangeira?

Pense bem: se temos duas tabelas “Empregado” e “Dependente”, qual das duas tabelas deve receber a chave estrangeira para haver o relacionamento? Qual é a regra?

Para resolver essa questão, nesta aula você vai estudar os conceitos e aplicações da cardinalidade em banco de dados relacional. Todos esses conceitos visam fazer com que o desenvolvimento do BD nas aplicações web seja funcional e

atenda às necessidades dos clientes.

A modelagem do banco no brModelo permite que a cardinalidade possa ser configurada. Além disso, possibilita o planejamento antes do desenvolvimento do BD propriamente dito. Faça da modelagem do banco de dados um hábito nos seus projetos, seja com DER, mapeamento, modelo lógico ou físico.



Para saber mais

Entre as diversas atribuições que o DBA possui, está também a modelagem de banco de dados. A utilização do brModelo fornece algumas vantagens na fase de modelagem, tais como: ser software *free*, fácil instalação, multiplataformas de S.O. (Sistema Operacional), fácil utilização com sistema “arrasta e solta”, entre outras. UTILIZE A MODELAGEM EM SEUS PROJETOS!

Vamos estudar cardinalidade e aplicá-la em suas modelagens?

Após você ter efetuado o relacionamento entre as tabelas nos bancos de dados na aula passada, sempre surge aquela dúvida no momento da modelagem:

Qual a tabela eu devo colocar a chave estrangeira para que haja o relacionamento?

Pois bem, a cardinalidade, segundo, Leite (2007, p. 14):



[...] é a quantificação do relacionamento entre duas entidades, e pode ser entendida como sendo, o número de ocorrências de determinada entidade, associado a uma ocorrência da outra entidade relacionada.

Para que você compreenda melhor uma aplicação da cardinalidade, observe o DER a seguir:

Figura 1 – Exemplo aplicação da cardinalidade



Fonte: Elaborada pelo autor.

Repare que, ao lado das entidades, existem os parênteses com algumas informações em seu interior, os quais representam a cardinalidade no brModelo.

Algumas perguntas são respondidas com essas informações, tais como:

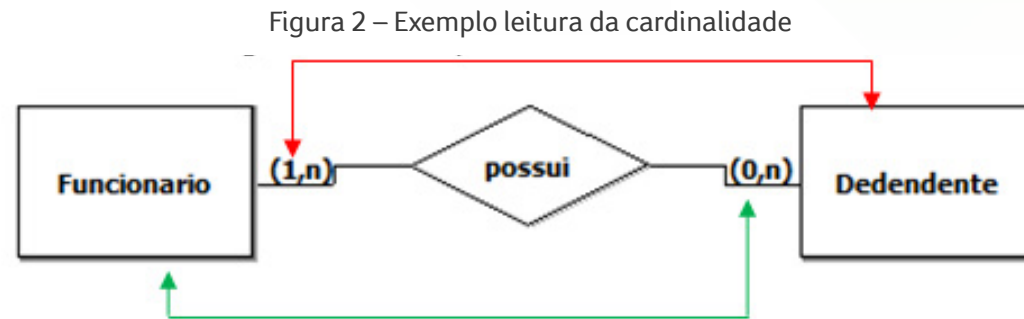
- Qual a quantidade de dependentes que cada funcionário poderá possuir em seu registro no banco de dados?
- Um dependente pode ter quantos funcionários responsáveis por ele?

A cardinalidade pode ser definida como:

- Mínima: opcional ou obrigatória.
- Máxima: podendo ter uma ou várias ocorrências.

Por meio da definição das cardinalidades no relacionamento entre as tabelas, será possível determinar as tabelas que devem receber a chave estrangeira, para a garantia dos relacionamentos.

Para a leitura das cardinalidades na modelagem de banco de dados, deve-se seguir uma regra. Observe a figura a seguir e entenda:



Fonte: Elaborada pelo autor.

- A cardinalidade entre parênteses em frente à entidade “Funcionário” refere-se às ocorrências da entidade “Dependente”;
- a cardinalidade entre parênteses em frente à entidade “Dependente” refere-se às ocorrências da entidade “Funcionario”;

- o primeiro valor entre os parênteses representa a cardinalidade mínima;
- o segundo valor entre os parênteses representa a cardinalidade máxima.

Ou seja, dessa forma a leitura da cardinalidade deve ser feita inversa do posicionamento no DER.

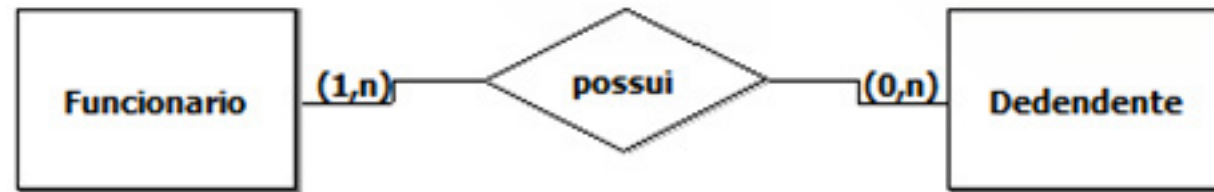
1. Cardinalidade Mínima

De acordo com Heuser (2008), a cardinalidade mínima pode apresentar duas situações:

- Opcional: é representado pelo número zero (0) e indica que o relacionamento não é obrigatório.
- Obrigatória: é representado pelo número um (1) e indica que obrigatoriamente o relacionamento deve apresentar no mínimo uma ocorrência.

Para melhor exemplificar, observe o DER a seguir:

Figura 3 – Exemplo cardinalidade mínima



Fonte: Elaborada pelo autor

No DER, pode-se observar que:

- A cardinalidade mínima é representado pelo primeiro número entre parênteses ao lado das entidades;
- No mínimo um DEPENDENTE deve possuir um FUNCIONÁRIO responsável por ele;
- No mínimo um FUNCIONÁRIO pode possuir nenhum DEPENDENTE.



Para saber mais

Em alguns projetos, a única forma para definir as cardinalidades é efetuar o levantamento de requisitos de projeto. Nos demais casos, o desenvolvedor consegue definir a quantidade de relacionamentos.

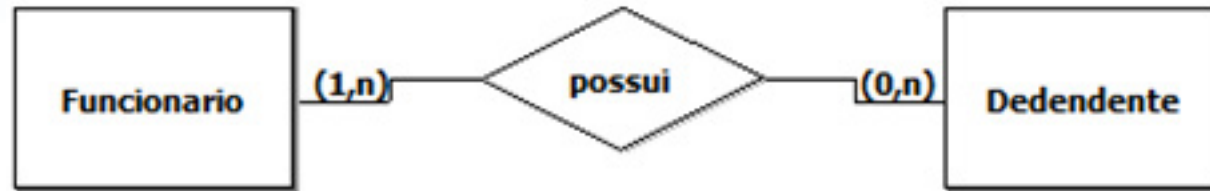
2. Cardinalidade Máxima

Conforme Heuser (2008), a cardinalidade máxima também apresenta com duas possibilidades, sendo elas:

- Máximo de um: é representado pelo número um (1) e indica que obrigatoriamente o relacionamento deve apresentar uma ocorrência.
- Máximo de muitos: é representado pela letra “n” e indica que o relacionamento pode apresentar várias ocorrências.

Observe o DER a seguir:

Figura 4 – Exemplo cardinalidade máxima



Fonte: Elaborada pelo autor.

No DER, pode-se observar que:

- A cardinalidade mínima é representado pelo segundo número entre parênteses ao lado das entidades;
- no máximo um DEPENDENTE pode possuir vários FUNCIONÁRIOS responsáveis por ele;
- no máximo um FUNCIONÁRIO pode possuir vários DEPENDENTES.

Agora você já deve ter notado que a cardinalidade representada no DER apresenta duas informações. A combinação das cardinalidades mínima e máxima é que vai determinar as possibilidades de relacionamentos.

3. Cardinalidade Binária

Para determinar a quantidade de relacionamentos nas cardinalidades binárias, existem quatro possibilidades:

(1,1)

Figura 5 – Exemplo cardinalidade (1,1)



Fonte: Elaborada pelo autor.

Ao lado na tabela “Dependente”, está a cardinalidade (1,1), representando que:
Um FUNCIONÁRIO pode ter UM e somente UM DEPENDENTE.

(0,1)

Figura 6 – Exemplo cardinalidade (0,1)



Fonte: Elaborada pelo autor.

Ao lado, na tabela “Dependente”, está a cardinalidade (0,1), representando que: Um FUNCIONÁRIO pode ter NENHUM ou UM DEPENDENTE.

(1,n)

Figura 7 – Exemplo cardinalidade (1,n)



Fonte: Elaborada pelo autor

Ao lado, na tabela “Dependente”, está a cardinalidade (1,n), representando que:
Um FUNCIONÁRIO pode ter UM ou VÁRIOS DEPENDENTES.

(0,n)

Figura 8 – Exemplo cardinalidade (0,n)



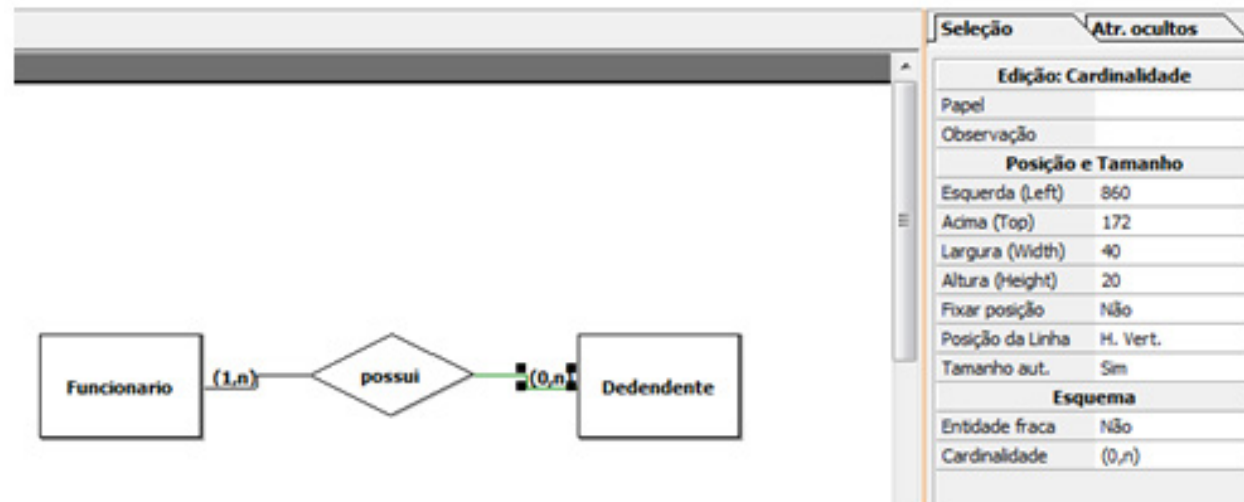
Fonte: Elaborada pelo autor.

Ao lado, na tabela “Dependente”, está a cardinalidade (0,n), representando que:
Um FUNCIONÁRIO pode NÃO ter NENHUM ou VÁRIOS DEPENDENTES.

4. Cardinalidade no brModelo

Para configurar as cardinalidades no brModelo, clique em cima da cardinalidade ao lado da entidade. Na barra lateral direita, um *menu* vai surgir na aba “Seleção”.

Figura 9 – Exemplo cardinalidade no brModelo



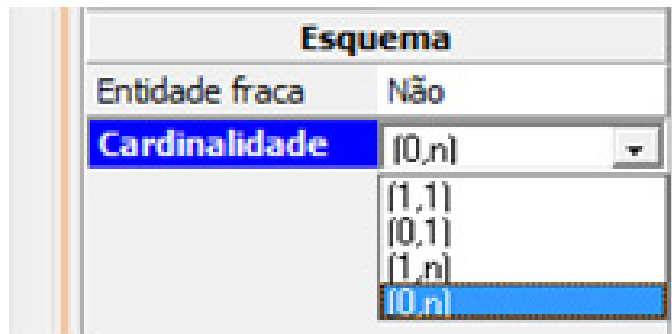
Fonte: Elaborada pelo autor.

Após isso, clique ao lado da ultima opção do *menu* “cardinalidade” e escolha a opção desejada. Observe a figura a seguir:

5. Cardinalidade para a Definição da Chave Estrangeira

Com as cardinalidades definidas nos relacionamentos, agora você pode definir as chaves estrangeiras com base nas regras definidas por Costa (2006), onde:

Figura 10 – Exemplo seleção da cardinalidade no brModelo



Fonte: Elaborada pelo autor.

- Quando a cardinalidade máxima de um (1) e a mínima for zero (0) em ambos os lados do relacionamento, fica a critério do analista decidir qual a tabela que vai migrar a chave estrangeira;
- Quando a cardinalidade máxima for um (1) em ambos os lados e a mínima for um (1) em pelo menos um dos lados, a entidade que possui a cardinalidade (1,1) é que vai receber a chave estrangeira;

- Quando em ambos os lados a cardinalidade máxima e mínima for (1,1), fica a critério do analista decidir qual a tabela que vai migrar a chave estrangeira;
- Quando a cardinalidade máxima for “n” em um dos lados e no outro for um (1), a tabela que possuir a cardinalidade máxima “n” deve receber a chave estrangeira da outra tabela;
- Quando a cardinalidade máxima for “n” em ambos os lados, deve ser criada uma nova tabela para representar o relacionamento. A nova tabela deve receber as chaves estrangeiras das duas tabelas.

Para saber mais

Não confunda o DER (Diagrama Entidade Relacionamento) com fluxograma! Apesar de as duas técnicas utilizarem formas geométricas para representar algum objeto no qual se deseja abstrair informações, as suas aplicações são distintas. O fluxograma descreve a sequência que as execuções devem ocorrer em um programa. Já o DER demonstra a estrutura do banco de dados e o relacionamento entre as tabelas.



Glossário

Cardinalidade mínima: número mínimo de associações entre as entidades.

Cardinalidade máxima: número máximo de associações entre as entidades.

Cardinalidade binária: número mínimo e máximo de associações entre as entidades.



Questão para reflexão

A empresa Shing Ling Ltda deseja manter informações em um BD sobre as entidades: funcionários, clientes, produtos e vendas.

A cardinalidade descreve que:

- Um funcionário pode estar associado em muitas vendas ou nenhuma;
- Uma venda tem que estar associada em um funcionário.
- Um produto pode estar em muitas vendas ou nenhuma;
- Uma venda deve possuir um produto;
- Um cliente pode possuir uma ou muitas vendas (compra);
- Uma venda deve possuir um cliente.

Defina os atributos e as chaves estrangeiras apropriadas.





Considerações Finais

- Conceitos de cardinalidades mínima nos relacionamentos;
- Conceitos de cardinalidades máxima nos relacionamentos;
- Conceitos e aplicações da cardinalidades binária entre as tabelas;
- Definição da chave estrangeira, dado uma cardinalidade;
- Modelagem no brModelo.



Referências

COSTA, R. L. **SQL**: Guia Prático. 2. ed. Rio de Janeiro: Brasport, 2006.

HEUSER, C. A. **Projeto de Banco de Dados**. 4. ed. Porto Alegre: Instituto de Informática da UFRGS; Sagra Luzzato, 2008.

LEITE, M. **Acessando Banco de Dados e com ferramentas RAD**: Aplicações em Delphi. 2. ed. Rio de Janeiro: Brasport, 2007.



Questão 1

1. Observe a citação de Leite (2007, p. 14):

“... é a quantificação do relacionamento entre duas _____, e pode ser entendida como sendo, o número de _____ de determinada entidade, associado a uma ocorrência da outra entidade _____”.

Assinale a alternativa que complete as lacunas corretamente:

- a) cardinalidades / tabelas / principal
- b) entidades / tabelas / correspondente
- c) entidades / ocorrências / relacionada
- d) cardinalidades / ocorrências / correspondente
- e) cardinalidades / associações / relacionada



Questão 2

2. As cardinalidades nos relacionamentos entre as tabelas definem as chaves estrangeiras. Assinale (V) verdadeiro ou (F) falso nas afirmativas a seguir:

- () As cardinalidades podem ser definidas como mínima e máxima.
 - () A cardinalidade mínima pode ser opcional ou obrigatória.
 - () As chaves primárias são definidas conforme a cardinalidade.
 - () A cardinalidade mínima em alguns casos pode apresentar várias ocorrências.
 - () A cardinalidade pode ser utilizada para definir a quantidade de atributos.
- a) V – F – V – F – V.
 - b) V – F – F – F – V.
 - c) F – V – F – V – F.
 - d) F – V – F – F – V.
 - e) V – V – F – F – V.

3. Observe o DER a seguir:



A partir do DER, observe as afirmativas:

I – Um aluno pode estar em uma ou várias turmas associadas.

II – Uma turma pode ter um ou vários alunos associados.

III – Um aluno pode estar atrelado em uma e somente uma turma.

Assinale a alternativa correta:

- a) Somente a alternativa III está correta.
- b) As alternativas II e III estão corretas.
- c) As alternativas I e III estão corretas.
- d) Somente a alternativa I está correta.
- e) Todas estão corretas.



Questão 4

4. As cardinalidades binárias podem ser representadas nos bancos de dados como:

- a) $(0,1); (0,n); (1,1); (1,n)$.
- b) $(n,1); (0,n); (1,1); (1,n)$.
- c) $(0,1); (0,n); (1,1); (n,n)$.
- d) $(0,1); (n,n); (1,1); (1,n)$.
- e) $(0,1); (0,n); (n,1); (1,n)$.



Questão 5

5. Observe as afirmativas a seguir:

I – Quando a cardinalidade $(0,1)$ está em ambas as tabelas, a chave estrangeira deve ser colocada na tabela principal.

II – Quando a cardinalidade for $(1,1)$ em ambas as tabelas, a chave estrangeira fica a critério do analista.

III. Quando uma tabela tem a cardinalidade $(1,n)$ e a outra $(1,1)$, a tabela que possui $(1,n)$ deve receber a chave estrangeira.

Assinale a alternativa correta:

- a) As alternativas I e III estão corretas.
- b) Somente a alternativa I está correta.
- c) As alternativas II e III estão corretas.
- d) Somente a alternativa III está correta.
- e) As alternativas I, II e III estão corretas.



Gabarito

1. Resposta: C.

“[...] é a quantificação do relacionamento entre duas entidades, e pode ser entendida como sendo o número de ocorrências de determinada entidade, associado a uma ocorrência da outra entidade relacionada” (LEITE, 2007, p. 14).

2. Resposta: E.

A cardinalidade define a chave estrangeira. A cardinalidade mínima pode ser opcional ou obrigatória, ou seja, 0 e 1.

3. Resposta: B.

A cardinalidade do DER descreve que uma turma pode ter um ou vários alunos

associados; um aluno pode estar atrelado a uma e somente uma turma.

4. Resposta: A.

A cardinalidade “n”, ou seja, muitas ocorrências não podem ser cardinalidade mínimas. Portanto, somente é possível utilizar (0,1); (0,n); (1,1); (1,n).

5. Resposta: C.

Quando a cardinalidade (0,1) está em ambas as tabelas, a chave estrangeira fica a critério do analista.



Unidade 4

Normalização (Formas Normais)

Objetivos

1. Compreender os conceitos e aplicações das formas normais nas tabelas.
2. Aplicar a 1FN, 2FN e 3FN, a fim de se normalizar o banco de dados.
3. Modelar banco de dados do tipo relacional no software brModelo.



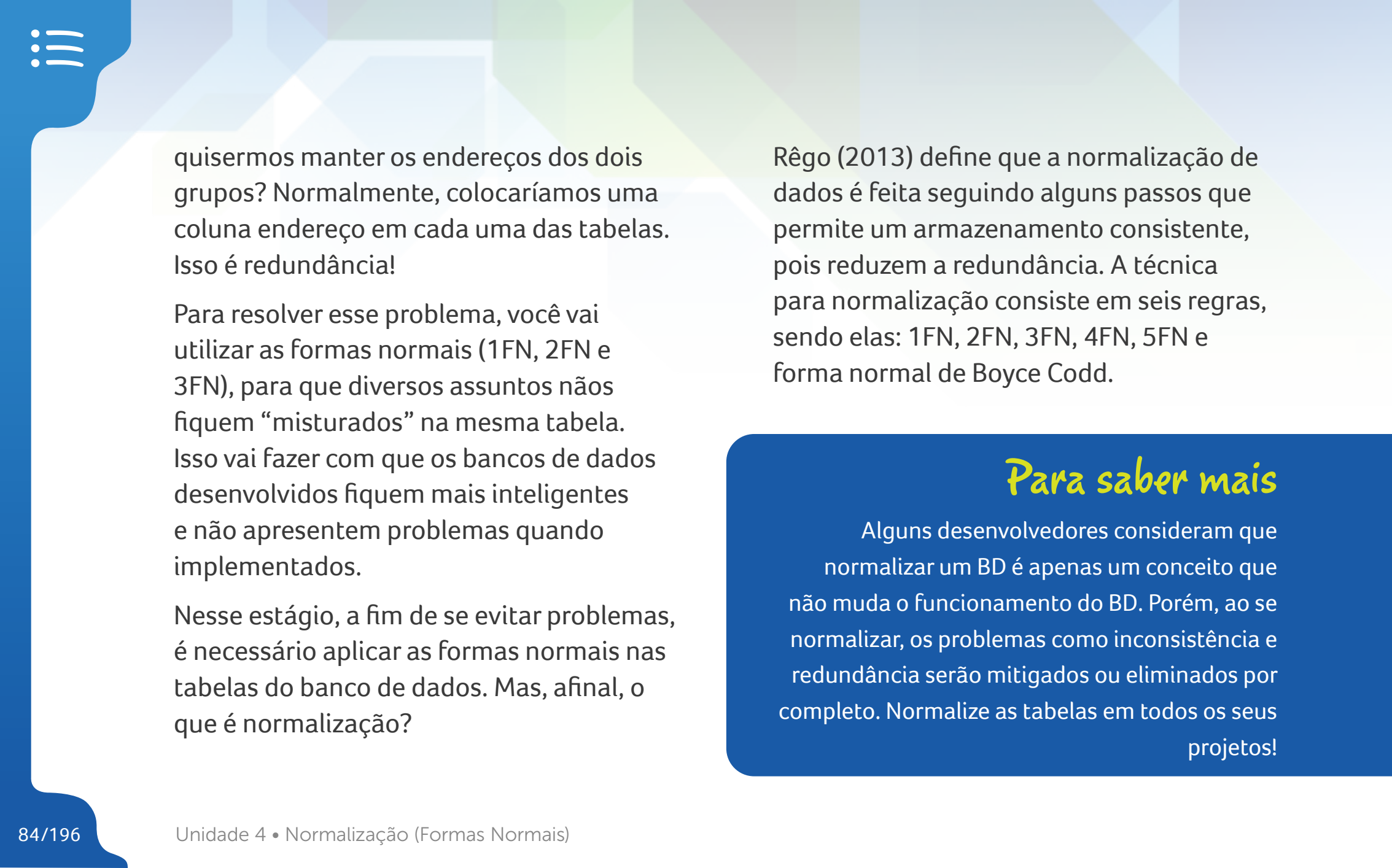
Introdução

Nas modelagens que você desenvolveu nas seções anteriores, não foi utilizada nenhuma norma para definição da estruturação dos campos existentes nas tabelas nos bancos de dados. Dessa forma, os bancos que foram desenvolvidos apresentariam problemas de redundância.

Você sabe o que é redundância? Segundo o dicionário Aurélio, “Repetição de palavras, de expressões; persistência nas mesmas ideias. Qualidade do que é repetitivo e supérfluo”. Você deve estar se perguntando: o que isso tem haver com banco de dados?

Desde o começo dos estudos, os bancos de dados que você desenvolveu foram do tipo relacional. Isso significa que em todos os planejamentos e modelagens foram utilizadas as técnicas de chaves (primária e estrangeira) para que as tabelas se relacionassem.

Porém, algumas informações foram colocadas de forma redundante em mais de uma tabela. Por exemplo: se tivermos as entidades FUNCIONÁRIO e CLIENTE, e



quisermos manter os endereços dos dois grupos? Normalmente, colocaríamos uma coluna endereço em cada uma das tabelas. Isso é redundância!

Para resolver esse problema, você vai utilizar as formas normais (1FN, 2FN e 3FN), para que diversos assuntos não fiquem “misturados” na mesma tabela. Isso vai fazer com que os bancos de dados desenvolvidos fiquem mais inteligentes e não apresentem problemas quando implementados.

Nesse estágio, a fim de se evitar problemas, é necessário aplicar as formas normais nas tabelas do banco de dados. Mas, afinal, o que é normalização?

Rêgo (2013) define que a normalização de dados é feita seguindo alguns passos que permite um armazenamento consistente, pois reduzem a redundância. A técnica para normalização consiste em seis regras, sendo elas: 1FN, 2FN, 3FN, 4FN, 5FN e forma normal de Boyce Codd.

Para saber mais

Alguns desenvolvedores consideram que normalizar um BD é apenas um conceito que não muda o funcionamento do BD. Porém, ao se normalizar, os problemas como inconsistência e redundância serão mitigados ou eliminados por completo. Normalize as tabelas em todos os seus projetos!

Para os desenvolvedores, quando um banco de dados se encontra na 3FN, é considerado normalizado. Sendo assim, os problemas como redundância e inconsistência não afetarão o desempenho dos BDs.

Caro aluno, observe o DER apresentado a seguir, para a compreensão de uma tabela não normalizada:

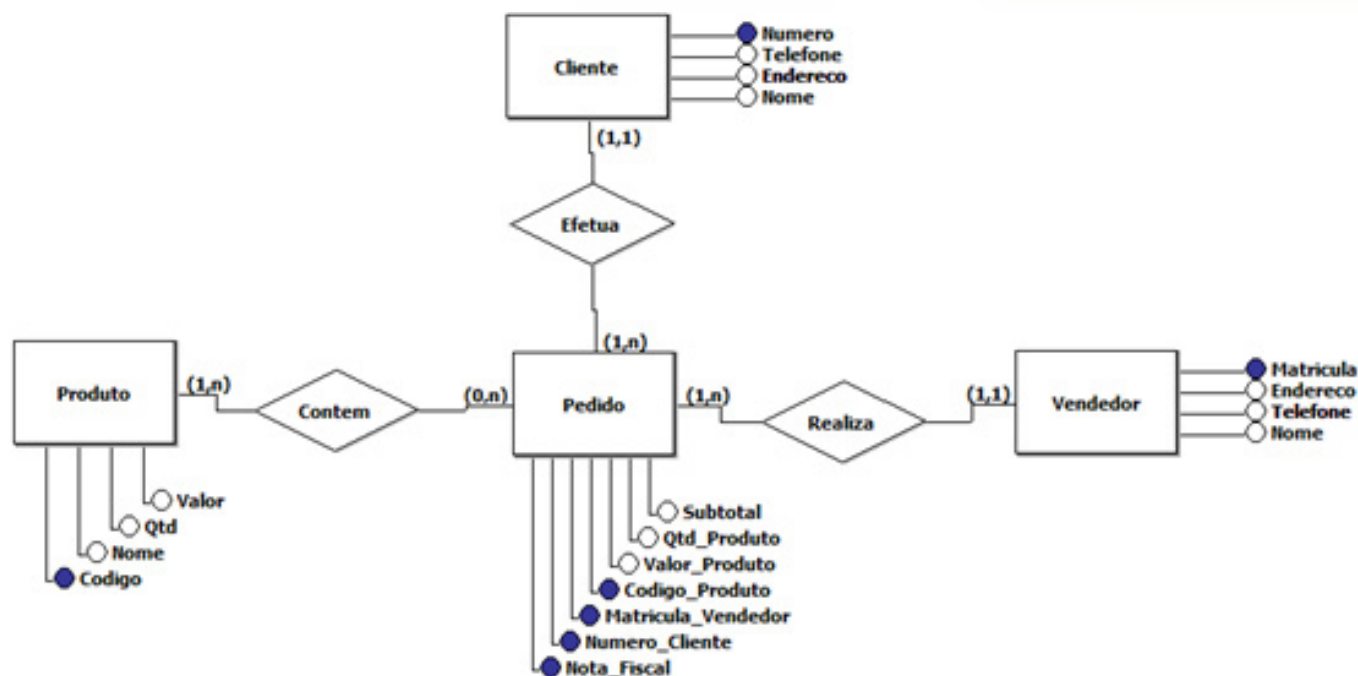
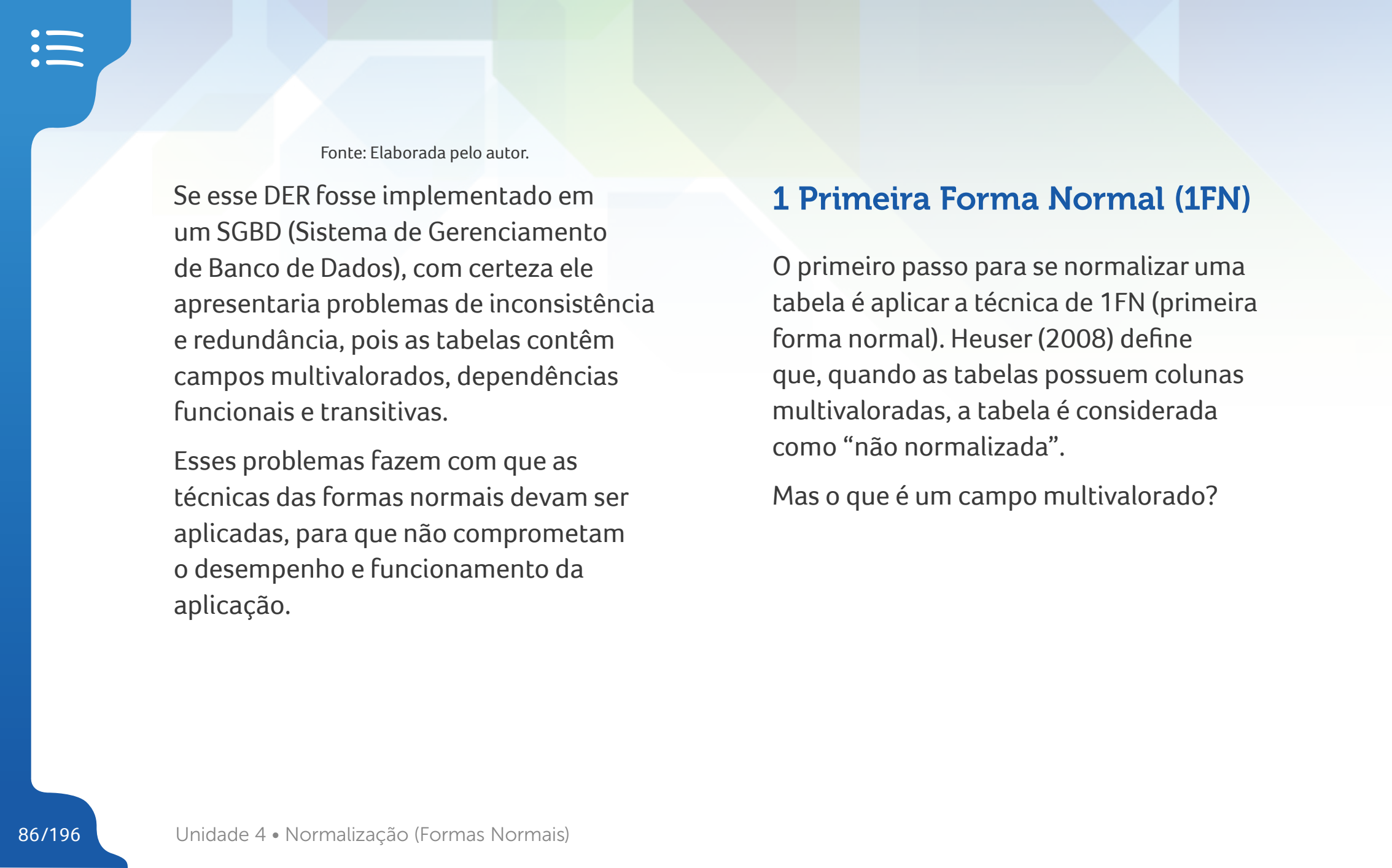


Figura 1 – Exemplo de DER para normalização



Fonte: Elaborada pelo autor.

Se esse DER fosse implementado em um SGBD (Sistema de Gerenciamento de Banco de Dados), com certeza ele apresentaria problemas de inconsistência e redundância, pois as tabelas contêm campos multivalorados, dependências funcionais e transitivas.

Esses problemas fazem com que as técnicas das formas normais devam ser aplicadas, para que não comprometam o desempenho e funcionamento da aplicação.

1 Primeira Forma Normal (1FN)

O primeiro passo para se normalizar uma tabela é aplicar a técnica de 1FN (primeira forma normal). Heuser (2008) define que, quando as tabelas possuem colunas multivaloradas, a tabela é considerada como “não normalizada”.

Mas o que é um campo multivalorado?

Para que você entenda o que são campos multivalorados, observe a tabela a seguir:

Cliente

Id	Nome	Telefone
0125	Jhonny Deep	98123-4567
4896	Serj Tankian	97557-1122 99112-3387
5866	Jackie Chan	3258-8744 99887-9124

Caro aluno, repare que no campo telefone pode receber mais de um valor. Sendo assim, a tabela “Cliente” não está normalizada.

Para uma tabela ser colocada na 1FN, Heuser (2008) define que:

- A chave primária da tabela que contenha campo multivalorado deve ser identificada;
- os campos multivalorados devem ser identificados e removidos;
- deve-se criar uma nova tabela com o campo removido;

- utilize a chave primária da tabela como chave estrangeira da tabela que possuía campo multivalorado.

No exemplo anterior, temos que:

Cliente

Id	Nome
0125	Jhonny Deep
4896	James Hetfield
5866	Jackie Chan

Contato

Id_Cliente	Telefone
0125	98123-4567
4896	97557-1122
4896	99112-3387
5866	3258-8744
5866	99887-9124

Repare que foi removida a coluna “Telefone” da tabela “Cliente”. Após isso, foi criada a tabela “Contato” com os números de telefone dos clientes. O relacionamento entre as tabelas só é possível porque a chave primária da tabela “Cliente” foi utilizada como chave estrangeira na tabela “Contato”.

Existem alguns casos que é possível colocar uma tabela na 1FN, sem que se tenha de fazer a remoção da coluna e a criação de uma nova tabela. Observe o exemplo a seguir:


Cliente

Id	Nome	Endereço
0125	Jhonny Deep	Rua Pirata, 55 – Mar Grande. CEP 01528-335
4896	James Hetfield	Rua do Rock, 148 – Barulho Bom. CEP 85687-836
5866	Jackie Chan	Rua do Ator, 556 – Artes Marciais. CEP 98551-228

Na tabela “Cliente”, o campo “Endereço” é multivalorado, pois está recebendo a rua, o número, o bairro e o CEP. Na primeira forma normal, não é necessário, nesse caso, criar uma nova tabela. Observe a seguir:

Cliente

Id	Nome	Rua	Número	Bairro	CEP
0125	Johnny Deep	Rua Pirata	55	Mar Grande	01528-335
4896	James Hetfield	Rua do Rock	148	Barulho Bom	85687-836
5866	Jackie Chan	Rua do Ator	556	Artes Marciais.	98551-228



O campo “Endereço” sumiu e em seu lugar foram criados os campos: “Rua”, “Número”, “Bairro” e CEP. Dessa forma, a tabela também se encontra na 1FN.

Com isso, podemos definir que, ao se colocar as tabelas na primeira forma normal, os campos que podem receber mais de um valor (campo multivalorado) são:

- Removidos da tabela e criado uma nova tabela, OU,
- Eliminado da tabela e criado diversos outros campos para guardar os valores.

2. Segunda Forma Normal (2FN)

Atenção, aluno, os estudos de normalização de banco de dados não permitem que etapas sejam puladas ou ignoradas. As técnicas exigem que apenas nos casos em que a tabela não tenha nenhum campo multivalorado se pode aplicar a 2FN.

Heuser (2008) define que:

- para se normalizar uma tabela na 2FN, ela obrigatoriamente tem de se encontrar na 1FN;
- as chaves devem depender unicamente da chave primária (dependência funcional).

Observe a tabela a seguir:

Aluno

RA	Nome	Curso	Período	Descrição
0123	Bob Esponja	Biologia	Noturno	Biologia marinha
5896	Johnny Bravo	Ed. Física	Noturno	Null
7785	Batman	Direito	Noturno	Null

Na tabela acima, repare que nenhuma coluna é possível entrar com mais de um valor (ou seja, não existe campo multivalorado), visto que:

- Id: chave primária, por isso os seus números não se repetem, nem recebem mais de um valor;
- Nome: cada aluno só pode ter um nome;
- Curso: só é possível colocar um nome de curso; porém, se o estudante fizer mais de um curso, cria-se um novo RA;
- Período: são limitados em matutino, vespertino, noturno ou integral; não é possível entrar com mais de um período por curso;

- Descrição: é possível entrar com mais de uma descrição, porém, embora possa ter mais de um assunto na descrição, é tido como um texto único.

Assim sendo, os campos “Curso”, “Período” e “Descrição” não possuem dependência funcional da chave primária “Id”, por isso é necessário colocá-la na 2FN.

Para isso, siga os seguintes passos:

- Identifique a chave primária da tabela;
- identifique as chaves que não contêm dependência funcional da chave primária;

- remova as colunas sem dependência funcional;
- crie uma nova tabela com os campos removidos;
- utilize a chave estrangeira para relacionar as tabelas.

Na tabela apresentada anteriormente, para colocá-la na 2FN, temos que:

Aluno

RA	Nome	Id_Curso
0123	Bob Esponja	111
5896	Johnny Bravo	222
7785	Batman	333

Curso

Id	Curso	Período	Descrição
111	Biologia	Noturno	Biologia marinha
222	Ed. Física	Noturno	Null
333	Direito	Noturno	Null

Caro aluno, os campos “Curso”, “Período” e “Descrição” foram retirados da tabela “Aluno” e passaram a fazer parte da tabela “Curso”. Na nova tabela, o campo “Id” é a chave primária, que serviu como chave estrangeira na tabela “Aluno” com a chave “Id_Curso”, fazendo então com que as duas tabelas se relacionassem. Com isso, ambas as tabelas podem ser consideradas na 2FN.

3. Terceira Forma Normal (3FN)

É considerada normalizada pelos DBAs e desenvolvedores de banco de dados, quando o BD se encontra na 3FN. Dessa forma, Heuser (2008) entende que estando normalizado o banco de dados estará “bem projetado”, livre de redundância e outras anomalias que possam comprometer a performance do sistema.

Heuser (2008) define que:


- uma tabela só estará na 3FN se estiver na 2FN;
- as colunas das tabelas não contêm dependência transitiva.

Mas o que é dependência transitiva?

Em banco de dados, é considerada dependência transitiva toda coluna que pode ser obtida utilizando-se por meio de equações matemáticas. Observe a tabela a seguir:

Produto

Código	Nome	Qtd	Valor	Total
M0102	Mouse	20	15,00	300,00
T5520	Teclado	15	25,00	375,00
C2520	Celular	4	750,00	3.000,00



Na tabela “Produto”, a coluna “Total” é obtida pela multiplicação dos valores das colunas “Qty” e “Valor”. Não necessariamente, a dependência transitiva é dada pela multiplicação, mas sim por qualquer operação lógica e/ou matemática.

Para normalizar a tabela na 3FN, siga os seguintes passos:

- Identificar as colunas que contém dependência funcional;
- remover a coluna com dependência funcional da tabela.

Na tabela apresentada anteriormente, na 3FN, temos que:

Produto

Código	Nome	Qty	Valor
M0102	Mouse	20	15,00
T5520	Teclado	15	25,00
C2520	Celular	4	750,00

Observe que a coluna “Total” foi retirada da tabela “Produto”, a fim de se eliminar a dependência transitiva, e assim se encontrar na 3FN.



Para saber mais

Atenção, aluno! Na fase de normalização, não é necessário passar pelas três etapas linearmente. Observe se a tabela não “fere” uma forma normal e faça a próxima. Por exemplo: uma tabela não tem campo multivalorado, então aplique as regras da 2FN.

Caro aluno, normalizar os bancos de dados não está entre as tarefas mais fáceis de fazer. Os exemplos utilizados nessa etapa demonstraram algumas vantagens para os desenvolvimentos, entre os quais estão:

Redução na duplicação dos dados: ao se eliminar os campos multivalorados, evitou-se que um mesmo dado pudesse estar presente em mais de um campo ou tabela.

Agrupamento dos dados: quando os campos que não continham dependência funcional da chave primária foram removidos da tabela e formaram uma nova tabela, isso fez com que somente campos com o mesmo “assunto” compusessem a mesma tabela.

Eliminação de campos inúteis: ao se eliminar das tabelas os campos com dependência transitivas, isso irá impactar diretamente no tempo de resposta das consultas (Query).



Para saber mais

Ao normalizar um banco de dados, elimina-se a inconsistência, redundância e outras anomalias que impactam negativamente no desempenho. O link a seguir demonstra algumas vantagens na normalização: <<http://www.sinfic.pt/SinficWeb/displayconteudo.do2?numero=34156>>. Acesso em: 26 maio 2016.



Glossário

Normalização de banco de dados: regras que permitem que as bases de dados não contenham redundância e/ou anomalias que comprometam o banco de dados.

1FN: remoção da coluna que contém campo multivalorado e criação de uma nova tabela e o seu respectivo relacionamento.

2FN: remoção da coluna que contém campo com dependência funcional e criação de uma nova tabela e o seu respectivo relacionamento.

3FN: remoção da coluna que contém campo com dependência transitiva.



Questão para reflexão

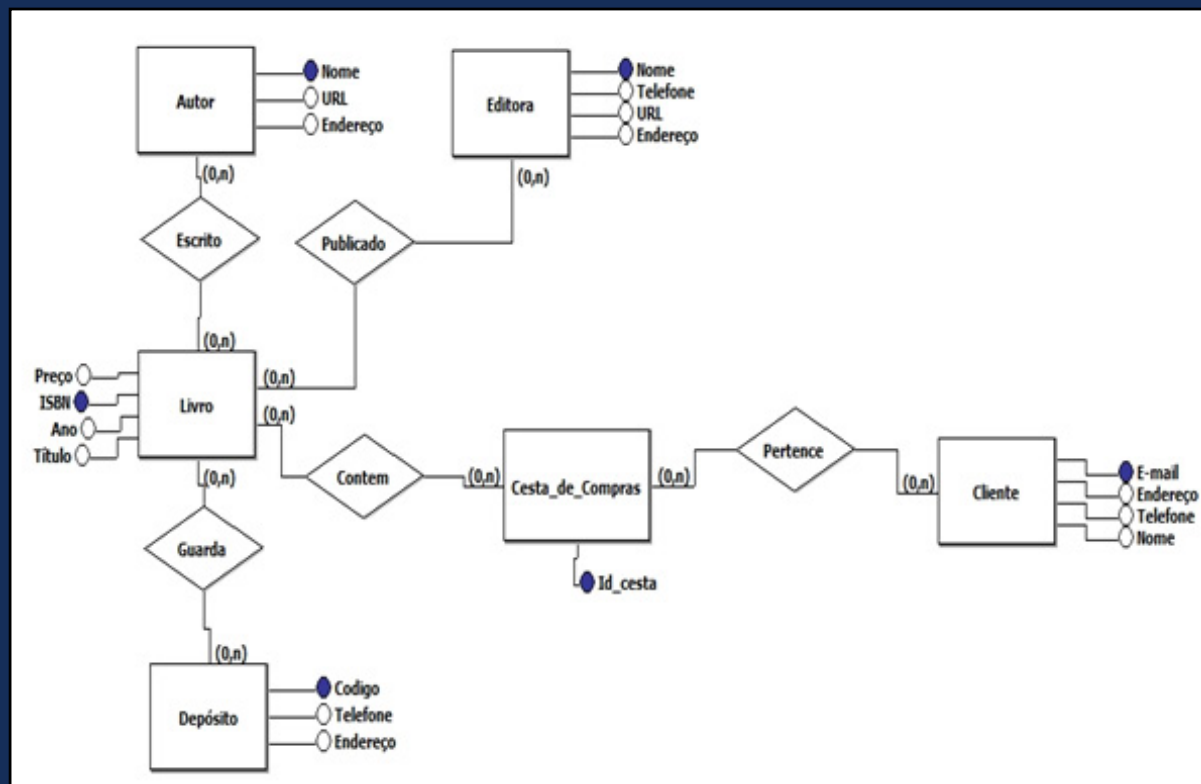
Uma livraria on-line possui um banco de dados no sistema de desenvolvimento web. Porém, ao colocar o sistema em funcionamento, alguns clientes começaram a registrar reclamações a respeito da dificuldade para consultar os livros e efetuarem registro, compras e demais atividades.

O DBA identificou que o problema é a falta de normalização do BD. Com base no DER a seguir, normalize o banco de dados da livraria.





Questão para reflexão





Considerações Finais

- Os conceitos e necessidades da normalização dos bancos de dados, a fim de se reduzir redundância e outras anomalias, que possam comprometer o funcionamento e/ou desempenho do banco de dados;
- A identificação dos campos multivalorados nas tabelas que compõem os bancos de dados. Para colocar a tabela na 1FN, a fim de se evitar redundância e comprometer as futuras consultas na base de dados;
- A identificação dos campos que não possuem dependências funcionais nas tabelas, para, assim, colocar a tabela na 2FN, para que “assuntos” distintos não sejam colocados na mesma tabela;
- A identificação dos campos que são gerados por dependência transitiva, para que sejam eliminados, fazendo com que o tempo de resposta das consultas seja mais rápido.



Referências

HEUSER, C. A. **Projeto de Banco de Dados**. 4. ed. Porto Alegre: Instituto de Informática da UFRGS; Sagra Luzzato, 2008.



Questão 1

1. A respeito de normalização, analise as afirmativas a seguir:

- I – Os processos de normalização são regras opcionais para minimizar a estrutura de um BD.
- II – As formas normais estão divididas em: 1FN, 2FN, 3FN, 4FN, 5FN e forma normal de Alan Turing.
- III – A normalização dos bancos de dados visa reduzir a redundância e outras anomalias que possam comprometer o desempenho do BD.

Assinale a alternativa correta:

- a) Somente a alternativa II está correta.
- b) As alternativas I e III estão corretas.
- c) Somente a alternativa I está correta.
- d) Somente a alternativa III está correta.
- e) As alternativas II e III estão corretas.



Questão 2

2. Relacione as colunas conforme a forma normal e a sua função:

- | | |
|---------|---|
| (1) 1FN | () Eliminar as dependências transitivas. |
| (2) 2FN | () Eliminar os campos multivalorados. |
| (3) 3FN | () Eliminar as dependências funcionais. |

Assinale a sequência correta:

- a) 1 – 2 – 3.
- b) 3 – 2 – 1.
- c) 3 – 1 – 2.
- d) 1 – 3 – 2.
- e) 2 – 1 – 3.



Questão 3

3. Observe o DER a seguir:



O esquema para o banco de dados acima tem problema de:

- a) 1FN na tabela “Curso”;
- b) 2FN na tabela “Aluno” e 2FN na tabela “Curso”;
- c) 1FN na tabela “Aluno”;
- d) 1FN na tabela “Aluno” e 3FN na tabela “Curso”;
- e) 3FN na tabela “Aluno” e 1FN na tabela “Curso”.



Questão 4

4. Uma tabela em um banco de dados possui o mapeamento a seguir:

Celular (Modelo, Nome, Marca, Qtd, Valor, Total)

Para normalizar a tabela, é necessário:

- a) eliminar o campo “Marca” e criar uma nova tabela; eliminar o campo “Total”;
- b) eliminar o campo o campo “Total” e criar uma nova tabela;
- c) eliminar o campo “Nome” e o campo “Qtd”;
- d) eliminar o campo “Valor” e criar uma nova tabela; eliminar o campo “Modelo”;
- e) eliminar o campo “Modelo” e criar uma nova tabela; eliminar o campo “Valor”.



Gabarito

1. Resposta: D.

A normalização são regras obrigatórias para se reduzir a redundância e outras anomalias; as formas normais estão divididas em: 1FN, 2FN, 3FN, 4FN, 5FN e Boyce Codd.

2. Resposta: C.

A 1FN visa eliminar os campos multivalorados; a 2FN visa eliminar as dependências funcionais; a 3FN visa eliminar as dependências transitivas.

3. Resposta: C.

O campo “Telefone” da tabela “Aluno” não está na 1FN.

4. Resposta: A.

Eliminar o campo “Marca” e criar uma nova tabela para colocar na 2FN; eliminar o campo “Total” para colocar na 3FN.



Unidade 5

Comandos DDL (*Data Definition Language*)

Objetivos

1. Compreender o surgimento, necessidades e aplicações do SQL (Structure Query Language).
2. Compreender os conceitos e aplicações dos comandos DDL.
3. Aplicar os comandos: CREATE, ALTER, TRUNCATE e RENAME.
4. Definir a estrutura dos bancos de dados do tipo relacional.



Introdução

A primeira linguagem para acesso aos gerenciadores de banco de dados relacional foi desenvolvida pela IBM. Porém, em 1986, devido à necessidade de padronização, a ISO e a ANSI lançaram a primeira versão do SQL (Structure Query Language), conforme afirma Costa (2007), sendo elas divididas em:

- DDL (Data Definition Language): utilizada para definir a estrutura do banco de dados;
- DML (Data Manipulation Language): utilizados para gerenciamento, seleção e manipulação dos objetos dentro do banco de dados;

- DCL (Data Control Language): relacionados à segurança e permissões dos usuários no banco de dados.

Os desenvolvedores dos Sistemas de Gerenciamento de Banco de Dados (SGBD) utilizam a linguagem SQL com algumas variações conforme o sistema. Porém, as características principais do padrão SQL são mantidas.

O sistema utilizado para os estudos é o MySQL. O SGBD foi desenvolvido na Suécia em 1980. Em 2008, foi adquirida pela Microsystems e, em 2009, adquirida pela Oracle (Fonte: <http://softwarelivre.org/mysql-brasil>)

Para saber mais

Uma alternativa para se utilizar o MySQL é a plataforma de desenvolvimento WAMP Server. O produto possui o Apache, PHP e MySQL nativo. O software está disponível em: <<http://www.wampserver.com/en/>>. Acesso em: 29 maio 2016.

Entre as suas principais características, estão:

- Compatibilidade com diversas linguagens de programação, como: C, Python, PHP, ASP, Ruby etc.;
- não exige grande capacidade de hardware;

- desempenho e estabilidade;
- facilidade de uso e instalação.

Ao instalar o SGBD, existem duas possibilidades de acesso à interface de utilização:

- Gráfico: são interfaces utilizadas para visualizar, criar e efetuar consultas nos bancos de dados;
- Command line: interface do *prompt* do *dos* (Disk Operating System).

Nos estudos dessa etapa, serão utilizados o CRUD (acrônimo de Create, Read, Update e Delete) SQL e o Command Line como ferramenta. Observe a interface a seguir:

Figura 1 – Interface MySQL

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

Fonte: Elaborada pelo autor.

A DB-Engines é uma entidade que efetua pesquisas relacionadas a problemas e utilização de bancos de dados. Um ranking analisa mensalmente a quantidade de empresas que utiliza os SGBDs, conforme apresentado a seguir:

Figura 2 – Interface MySQL

Rank			DBMS	Database Model	Score		
May 2016	Apr 2016	May 2015			May 2016	Apr 2016	May 2015
1.	1.	1.	Oracle	Relational DBMS	1462.02	-5.51	+19.93
2.	2.	2.	MySQL +	Relational DBMS	1371.83	+1.72	+77.56
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1142.82	+7.77	+11.79
4.	4.	4.	MongoDB +	Document store	320.22	+7.78	+42.90
5.	5.	5.	PostgreSQL	Relational DBMS	307.61	+3.89	+34.09
6.	6.	6.	DB2	Relational DBMS	185.96	+1.87	-15.09
7.	↑ 8.	↑ 8.	Cassandra +	Wide column store	134.50	+4.83	+27.95
8.	↓ 7.	↓ 7.	Microsoft Access	Relational DBMS	131.58	-0.39	-14.00
9.	9.	↑ 10.	Redis +	Key-value store	108.24	-3.00	+13.51
10.	10.	↓ 9.	SQLite	Relational DBMS	107.26	-0.70	+2.10

Fonte: DB-Engines. Disponível em: <<http://db-engines.com/en/ranking>>. Acesso em: 29 Mai 2016.

A pesquisa leva em consideração a análise de 305 sistemas de gerenciamento de banco de dados. Como se pode perceber, em segundo lugar no ranking se apresenta o MySQL. Por esses motivos e alegações, os estudos da linguagem atendem às necessidades do mercado.



1. DDL (Data Definition Language)

Os primeiros objetos de estudo da linguagem MySQL estão no grupo de comandos DDL. Segundo Costa (2007), estão:

- CREATE: são comandos utilizados para criar banco de dados e tabelas;
- SHOW: são comandos utilizados para consultar banco de dados e tabelas;
- DROP: são comandos utilizados para excluir banco de dados e tabelas;
- ALTER: são comandos utilizados para alterar a estrutura de uma tabela;
- TRUNCATE: comando para apagar os registros inseridos em uma base de dados;
- RENAME: são comandos utilizados para renomear tabelas e campos dentro de um banco de dados.

Para saber mais

Uma boa prática é escrever os códigos MySQL em editores. O Notepad++ disponibiliza nativamente a personalização com cores para SQL. O software está disponível em: <<https://notepad-plus-plus.org/>>. Acesso em: 29 maio 2016.

Caro aluno, para fins do entendimento das sintaxes utilizadas no MySQL:

- Os símbolos [] ou <> não devem ser digitados; eles são utilizados para organizar a forma como as sintaxes devem ser utilizadas;
- os comandos serão utilizados em letras maiúsculas nos scripts,

contudo também funciona se forem usadas letras minúsculas;

- não são utilizados acentos ou cedilha nos nomes;
- para que o SGBD execute os comandos digitados, OBRIGATORIAMENTE é necessário um ponto e vírgula (;) no final de cada comando.

2.1 Comando CREATE

De acordo com Ferrari (2007) são comandos utilizados para criar objetos dentro do SGBD. Os comandos estão divididos em:

- Para criar um banco de dados no SGBD utilize:

```
CREATE DATABASE <Nome_do_BD>;
```

- Com o BD criado, é necessário entrar nele, para ter permissão de criar as tabelas. Para isso, utilize o comando:

```
USE <Nome_do_BD>;
```

Antes de criar as tabelas dentro do BD, é necessário conhecer os tipos de dados, que é possível utilizar nas estruturas. Assim como nas linguagens de programação como: C, C++, Java entre outras, o MySQL necessita ser declarado o tipo e tamanho dos campos das tabelas. Os principais tipos são mostrados a seguir:

Quadro 1 – Tipos de dados em BD MySQL

TIPO	DESCRIÇÃO
CHAR(n)	Caracteres de tamanho fixo, de 1 a 255.
VARCHAR (n)	Sequência alfanumérica com ate 255 caracteres.
INT (n)	Números inteiros.
FLOAT, DOUBLE	Números fracionários com ponto flutuante.
DATE	Datas no formato ano-mês-dia.

TIPO	DESCRIÇÃO
not null	Não aceita valor nulo.
auto_increment	Adiciona um valor automaticamente.

Fonte: Elaborada pelo autor

- A sintaxe para criar as tabelas nos bancos de dados, deve ser utilizada:

CREATE TABLE <nome_da_tabela> (

→ <nome_da_coluna1> <tipo_da_coluna1> (tamanho1),

→ <nome_da_coluna2> <tipo_da_coluna2> (tamanho2));

Repare que, após declarar a primeira coluna, é colocada uma vírgula e apertada a tecla “ENTER” para ir para a próxima linha. Ao final, é necessário encerrar com ponto e vírgula (;).

- Para declarar os campos que não podem receber valores nulos, utilize:

CREATE TABLE <nome_da_tabela> (

→ <nome_da_coluna> <tipo_da_coluna> (tamanho) NOT NULL);

- Quando é necessário que o número seja gerado automaticamente, utilize:

```
CREATE TABLE <nome_da_tabela> (
```

```
→ <nome_da_coluna> <tipo_da_coluna>  
(tamanho) auto_increment);
```

- Para declarar um dos campos como chave primária, utilize:

```
CREATE TABLE <nome_da_tabela> (
```

```
→ <nome_da_coluna1> <tipo_da_coluna1>  
(tamanho1) primary key,
```

```
→ <nome_da_coluna2> <tipo_da_coluna2>  
(tamanho2));
```

- Para declarar um dos campos como chave estrangeira, utilize:

```
CREATE TABLE <nome_da_tabela> (
```

```
→ <nome_da_coluna1> <tipo_da_coluna1>  
(tamanho1) primary key,
```

```
→ <nome_da_coluna_FK> <tipo_da_coluna_FK>  
(tamanho_FK),
```

```
→ FOREIGN KEY (<nome_da_coluna1>)  
REFERENCES <Nome_da_Tabela_PK>  
(<Nome_da_Coluna_PK>));
```

- Para que o SGBD retorne a estrutura de uma tabela, utilize:

```
DESCRIBE <Nome_da_Tabela>;
```

2.2 Comando SHOW

Segundo Ferrari (2007), são comandos utilizados para listar os objetos dentro do SGBD. Os comandos estão divididos em:

- Para listar os BDs criados dentro do SGBD, utilize:

SHOW DATABASES;

- Para listar as tabelas dentro do BD, utilize:

SHOW TABLES;

2.3 Comando DROP

Conforme Ferrari (2007), são comandos utilizados para excluir os objetos dentro do SGBD (menos registros). Os comandos estão divididos em:

- Para excluir um banco de dados, utilize:

DROP DATABASE <Nome_do_BD>;

- Para excluir uma tabela, utilize:

DROP TABLE <Nome_da_Tabela>;

2.4 Comando TRUNCATE

Embora nessa etapa não sejam estudados os comandos para inserção de dados nas tabelas, o comando TRUNCATE faz parte dos comandos DDL. Segundo Ferrari (2007), é o comando utilizado para excluir todos os registros dentro de uma tabela.

- Para excluir todos os registros no interior de uma tabela, utilize:

TRUNCATE <Nome_da_Tabela>;

Para saber mais

Uma outra alternativa para se utilizar o MySQL é a plataforma de desenvolvimento XAMP. O software possui, de forma nativa, os servidores: PHP, MariaDB e Perl. O software está disponível em: <https://www.apachefriends.org/pt_br/download.html>. Acesso em: 30 maio 2016.

2.5 Comando ALTER

De acordo com Ferrari (2007), são comandos utilizados para alterar a estrutura de uma tabela. Os comandos estão divididos em:

- Para adicionar uma coluna em uma tabela já desenvolvida, utilize:

```
ALTER TABLE <Nome_da_Tabela> ADD  
<Nome_do_Campo> <Tipo_do_Campo>  
(Tamanho);
```

- Para excluir uma coluna em uma tabela já desenvolvida, utilize:

```
ALTER TABLE <Nome_da_Tabela> DROP  
COLUMN <Nome_do_Campo>;
```

- Para modificar a estrutura de uma coluna em uma tabela já desenvolvida, utilize:

```
ALTER TABLE <Nome_da_Tabela> MODIFY  
<Nome_do_Campo> <Novo_tipo>  
(Tamanho_do_Novo_tipo);
```

2.5 Comando RENAME

Segundo Ferrari (2007), são comandos utilizados para renomear os objetos dentro do SGBD (menos BD). Os comandos estão divididos em:

- Para renomear um banco de dados:

O comando utilizado para renomear os BDs foram descontinuados desde 2010, para que a integridade das bases de dados fosse garantida.

- Para renomear uma tabela dentro de um BD, utilize:

```
ALTER TABLE <nome_tabela> RENAME  
<novo_nome_tabela>;
```

- Para se renomear uma coluna, utilize:

```
ALTER TABLE <nome_tabela> CHANGE  
<nome_da_coluna> <novo_nome_coluna>  
<tipo> (Tamanho).
```




Glossário

SQL: linguagem utilizada para criação, alteração e consulta em banco de dados.

DDL: linguagem SQL utilizada para definir a estrutura do banco de dados.

Sintaxe DDL: comandos utilizados para criar, alterar, excluir e listar os objetos dentro do SGBD.



Questão para reflexão

Uma agência bancaria necessita desenvolver um banco de dados para controle e registro de empréstimos e depósitos. Com base no mapeamento a seguir, desenvolva o BD utilizando o MySQL.

Agencias (Codigo, Nome, Cidade)

Clientes (Codigo, Codigo_Agencias, Nome, Rua, Cidade)

Depositos (Numero, Codigo_Clientes, Codigo_Agencias, valor)

Emprestimos (Numero, Codigo_Clientes, Codigo_Agencias, valor, parcelas)





Questão para reflexão

ATENÇÃO:

- Defina os tipos de dados e tamanhos dos campos;
- As palavras em **negrito** são os nomes da tabela;
- Os campos com simples sublinhado são as chaves primárias;
- Os campos com duplo sublinhado são as chaves primárias.





Considerações Finais

- O surgimento e aplicações da linguagem SQL (Structure Query Language) e a sua padronização por parte das empresas desenvolvedoras de SGBD;
- A identificação dos comandos DDL (Data Definition Language) e o entendimento e aplicação da sintaxe;
- O entendimento dos tipos de dados que podem ser utilizados nos campos das tabelas dos bancos de dados;
- O desenvolvimento e estruturação de banco de dados utilizando a sintaxe MySQL.



Referências

COSTA, R. L. **SQL**: Guia Prático. 2. ed. Rio de Janeiro: Brasport, 2006.

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



Questão 1

1. A linguagem de programação de banco de dados SQL apresenta três grupos de comandos, para que sejam realizadas as funções como: criação, manipulação e segurança de banco de dados. Assinale a alternativa que demonstre os três grupos de comandos SQL:

- a) DML – DOS – TIC.
- b) DNS – DML – DOS.
- c) DOS – NIC – DNS.
- d) Server – DML – DLD.
- e) DDL – DML – DCL.



Questão 2

2. Após desenvolver um BD para uma biblioteca, o desenvolvedor deseja alterar o tipo da coluna localização de "int (10)" para "varchar (15)". Assinale a alternativa que descreva o comando que deve ser utilizado no MySQL.

- a) ALTER TABLE livros MODIFY prateleira varchar (15).
- b) ALTER TABLE livros CHANGE prateleira varchar (15).
- c) ALTER TABLE livros DROP prateleira varchar (15).
- d) ALTER TABLE CHANGE livros MODIFY prateleira varchar (15).
- e) ALTER TABLE livros RENAME prateleira varchar (15).



Questão 3

3. Ao se realizar o levantamento de requisitos para o desenvolvimento de uma loja de conveniência, o analista notou a possibilidade do reuso de requisitos de outro projeto. Mas, para utilizar as tabelas, é necessário que todos os dados contidos no interior das tabelas sejam apagados. Assinale a alternativa que descreva a sintaxe utilizada para excluir os registros.

- a) DROP <Nome_da_Tabela>;
- b) TRUNCATE <Nome_do_BD>;
- c) TRUNCATE <Nome_da_Tabela>;
- d) DROP <Nome_do_BD>;
- e) DELETE ALL <Nome_da_Tabela>;



Questão 4

4. O grupo de comandos SQL RENAME é utilizado para renomear objetos no SGBD. Observe as afirmativas a seguir:

- I – É possível renomear um banco de dados no SGBD.
- II – É possível renomear uma tabela dentro de um banco de dados.
- III – É possível renomear uma coluna dentro de uma tabela.

Assinale a alternativa correta:

- a) Somente a alternativa I está correta.
- b) Somente a alternativa II está correta.
- c) Somente as alternativas II e III estão corretas.
- d) Somente as alternativas I e III estão corretas.
- e) Somente a alternativa III está correta.



Questão 5

5. Análise as sintaxes a seguir e assinale (V) verdadeiro ou (F) falso nas sintaxes SQL:

- () ALTER TABLE <Nome_da_Tabela> ADD <Nome_do_Campo> <Tipo_do_Campo> (Tamanho);
- () USE <Nome_da_Tabela>;
- () CREATE DATABASE <Nome_da_Tabela>;
- () ALTER TABLE <nome_tabela> RENAME <novo_nome_tabela>;
- () ALTER TABLE <Nome_da_Tabela> DROP COLUMN <Nome_do_Campo>;

Assinale a sequência correta:

- a) V – F – V – F – V.
- b) F – V – V – V – F.
- c) F – F – V – F – V.
- d) V – F – F – V – V.
- e) F – F – F – V – V.



Gabarito

1. Resposta: E.

O MySQL é dividido nos grupos de comandos: DDL (Data Definition Language), DML (Data Manipulation Language) e DCL (Data Control Language).

2. Resposta: A.

A sintaxe utilizada para alterar a estrutura de uma coluna em uma tabela deve ser: ALTER TABLE <Nome_da_Tabela> MODIFY <Nome_do_Campo> <Novo_tipo> (Tamanho_do_Novo_tipo).

3. Resposta: C.

Para excluir todos os registros no interior de uma tabela, deve ser utilizada a sintaxe: TRUNCATE <Nome_da_Tabela>

4. Resposta: C.

O comando RENAME é utilizado para renomear os atributos ou entidades. A renomeação de BD foi descontinuada para não comprometer a integridade dos BDs.

5. Resposta: D.

2ª Sintaxe: USE <Nome_do_BD>;

3ª Sintaxe: CREATE DATABASE <Nome_do_BD>;



Unidade 6

Comandos DML (*Data Manipulation Language*)

Objetivos

1. Compreender os conceitos e aplicações dos comandos DML.
2. Aplicar os comandos SELECT, INSERT, UPDATE e DELETE, na manipulação de dados.
3. Compreender a necessidade de utilização das condicionais para complementação dos comandos DML.



Introdução

Os comandos DDL (*Data Definition Language*) permitiram que você desenvolvesse um banco de dados e estruturasse as tabelas nele contido. Neste momento, será necessário fazer a inserção dos dados nas tabelas para, posteriormente, fazer a manipulação dos dados contidos nas tabelas.

Segundo Date (2003), o SGBD deve ter em suas funcionalidades técnicas que permitam ao usuário fazer a inserção, a busca, a atualização ou exclusão dos dados. Ainda, o SGBD deve permitir o processamento ou compilador para a linguagem de manipulação de dados.

As técnicas que serão desenvolvidas e aplicadas nas bases de dados, nesse capítulo, vão permitir que você:

- Insira dados nas tabelas;
- Apague os dados inseridos nas tabelas;
- Modifique os dados inseridos nas tabelas;
- Consulte os dados inseridos nas tabelas.

Para personalizar a utilização e/ou aplicação de todas as técnicas que serão estudadas neste capítulo, será necessária a compreensão dos operadores lógicos e aritméticos, para que as funções citadas anteriormente possam ser direcionadas aos objetivos e necessidades do administrador do BD.

Você se lembra no primeiro tema, quando foi questionado se os BDs armazenavam

dados ou informações? Pois bem, neste momento tudo ficará mais claro. Os dados inseridos nas tabelas do banco de dados irão gerar informações por meio das consultas nos comandos DML.

1. Operadores Lógicos e Relacionais

Assim como nas linguagens de programação como C, C++, Java, entre outras, o MySQL necessita dos operadores aritmético e relacional para obter os resultados desejados. Galassi (2015) define que os operadores são utilizados em consultas SQL para realizar tarefas como: agrupar valores, comparar dados, entre outros.

Silberschatz (2010) define os operadores relacionais conforme o quadro a seguir:

Quadro 1 – Operadores relacionais

Operador Relacional	Descrição
=	Igual
!=	Diferente
>	Maior
>=	Maior ou igual
<	Menor
<=	Menor ou igual

Fonte: Adaptado Silberschatz (2010).



Já os operadores lógicos estão divididos em:

Quadro 2 – Operadores lógico

Operador Lógico	Descrição
&&	AND (E)
	OR (OU)
!	NOT (NÃO)

Fonte: Adaptado Silberschatz (2010).

Para saber mais

As mesmas técnicas utilizadas nas linguagens de programação como C, C++, Java, entre outras, para operadores booleanos são utilizados nas sintaxes SQL. O artigo da Revista SQL demonstra algumas dessas aplicações, disponível em: <<http://www.devmedia.com.br/condicoes-e-operadores-em-consultas-sql-revista-sql-magazine-90/22001>>. Acesso em: 01 jun. 2016.

Utilizado como operador booleano (true ou false), as suas comparações devem seguir a tabela verdade. Nunes (2016) exemplifica as utilizações de operadores, conforme pode ser observado nos quadros a seguir:

Quadro 3 – Representação do operador lógico “E”


Nota >= 6,0	Operador	Presença >= 75%	Aprovação
Aprovado	E	Aprovado	Aprovado
Aprovado	E	Reprovado	Reprovado
Reprovado	E	Aprovado	Reprovado
Reprovado	E	Reprovado	Reprovado

Fonte: Adaptado Nunes (2016, p. 38).

Quadro 4 – Representação do operador lógico “OU”

Nota >= 6,0	Operador	Presença >= 75%	Aprovação
Aprovado	OU	Aprovado	Aprovado
Aprovado	OU	Reprovado	Aprovado
Reprovado	OU	Aprovado	Aprovado
Reprovado	OU	Reprovado	Reprovado

Fonte: Adaptado Nunes (2016, p. 38)



A aplicação da técnica correta, conforme demonstrado nos quadros acima, pode influenciar nos resultados das consultas nos bancos de dados.

2. Comando INSERT

Após o desenvolvimento das tabelas, é chegada a hora de você começar a inserir

os dados nas tabelas. Para isso, será necessário utilizar o comando INSERT. Segundo Ferrari (2007), a sintaxe utilizada para fazer a inserção de dados deve ser utilizada:

```
INSERT <Nome_da_tabela> VALUES  
(<Valor_Campo1>, <Valor_Campo2>, ...);
```

Porém, a sintaxe obriga o desenvolvedor a entrar com valores em todos os campos. Observe o código a seguir:

```
CREATE DATABASE Funcionário;
```

```
USE cadastro;
```

```
CREATE TABLE Cadastro (
```

```
→ Id INT (3) not null primary key auto_  
increment,
```

- Nome VARCHAR (35) not null,
- Data_Nasc DATE not null,
- Sexo VARCHAR (8),
- Escolaridade VARCHAR (25));

Para representar, tem-se a tabela:

Cadastro

Id	Nome	Data_Nasc	Sexo	Escolaridade

Com base na tabela acima, o comando utilizado será:

```
INSERT Cadastro VALUES ("1", "Bruce Dickinson", "1958/08/07", "M", "Superior");
```

Porém, nos campos não obrigatórios como: Id (auto_increment), Sexo e Escolaridade, se não forem declarados, é necessário utilizar a seguinte sintaxe:

```
INSERT <Nome_da_tabela> (<Campo1>, <Campo2>) VALUES (<Valor_Campo1>, <Valor_Campo2>, ...);
```



Para o exemplo da tabela acima, para entrar com o cadastro sem os campos obrigatórios, temos que:

```
INSERT Cadastro (Nome, Data_Nasc)  
VALUES ("Bruce Dickinson", "1958/08/07");
```

Quando é necessário inserir dados em algumas colunas não obrigatórias, devem ser declarados primeiramente os campos em que se queira entrar com os dados.

3. Comando SELECT

Uma vez que as tabelas contenham dados no seu interior, é possível extrair informações, por meio da seleção dos dados. Segundo Miyagusku (p. 40, 2008), “O comando SELECT é um dos principais da linguagem SQL e retorna as informações das tabelas de um banco de dados”.

Para saber mais

Para o entendimento dos comandos SELECT, UPDATE e DELETE, é necessário que você entenda o significado da expressão condição nas sintaxes. Por exemplo: para selecionar os registros que a condição são os salários maiores que 5000, temos que: `salário > 5000`. Ou seja, um campo que apresenta a condição (Operadores relacionais) dado um valor.

Os comandos estão divididos em:

- Para selecionar todos os registros de uma tabela:

```
SELECT * FROM <Nome_da_Tabela>;
```

Para personalizar as seleções (Query), é necessário combinar o comando

SELECT com algum *alias*. Segundo Miyagusku (2010), os comandos *alias* são especificações utilizadas para distinguir os campos de uma tabelas, quando pesquisado, onde temos que:


- SELECT + WHERE: utilizado para especificar critérios de seleção:

```
SELECT * FROM <Nome_da_Tabela>  
WHERE <Condição>;
```

- Para especificar um campo no SELECT, utilize:

```
SELECT <Campo1>, <Campo2> FROM  
<Nome_da_Tabela> WHERE <Condição>;
```

- Quando for necessário especificar mais de um critério de seleção, deve ser utilizado o comando AND, onde:



```
SELECT * FROM <Nome_da_Tabela> WHERE  
<Condição1> AND <Condição2>;
```

- Quando for necessário especificar um critério OU outro de seleção, deve ser utilizado o comando OR, onde:

```
SELECT * FROM <Nome_da_Tabela> WHERE  
<Condição1> AND <Condição2>;
```

- Para efetuar uma consulta e especificar a letra ou número que a pesquisa se inicia, utilize:

```
SELECT * FROM <Nome_da_Tabela> WHERE  
<Campo> LIKE (letra_ou_numero_inicial%);
```

- Para ordenar a pesquisa de uma seleção utilize “asc” (para crescente) e “desc” (para decrescente), em que:

```
SELECT * FROM <Nome_da_Tabela> ORDER  
BY <Campo> asc;
```

```
SELECT * FROM <Nome_da_Tabela> ORDER  
BY <Campo> desc;
```

4. Comando UPDATE

Em certas situações, é necessário alterar um ou mais registros dentro de uma tabela. Para isso, você vai utilizar o comando UPDATE. De acordo com Miyagusku (p. 42, 2008), “O comando UPDATE é utilizado para atualizar/alterar os dados de uma tabela”.

Os comandos estão divididos em:

- Para alterar um ou mais registros dentro de uma tabela, dada uma condição, utilize:

```
UPDATE <Nome_da_Tabela> SET <Coluna>  
<operador> <Novo_Valor> WHERE  
<Condição>;
```

- Para alterar mais de uma coluna, dada mais de uma condição, utilize:

```
UPDATE <Nome_da_Tabela> SET  
<Coluna1> <operador> <Novo_Valor1>,  
<Coluna2> <operador> <Novo_Valor2>  
WHERE <Condição1> AND <Condição2>;
```

5. Comando DELETE

Para apagar os registros em uma tabela dada uma condição, é necessário utilizar o comando DELETE. Segundo Miyagusku (p. 43, 2008), “O comando DELETE é utilizado para excluir os registros de uma tabela”.

- Para excluir um ou mais registros dentro de uma tabela, dada uma condição, utilize:

```
DELETE FROM <Nome_da_Tabela> WHERE  
<Condição>;
```



Glossário

Sintaxe DML: comandos utilizados para a manipulação dos dados dentro de uma tabela.

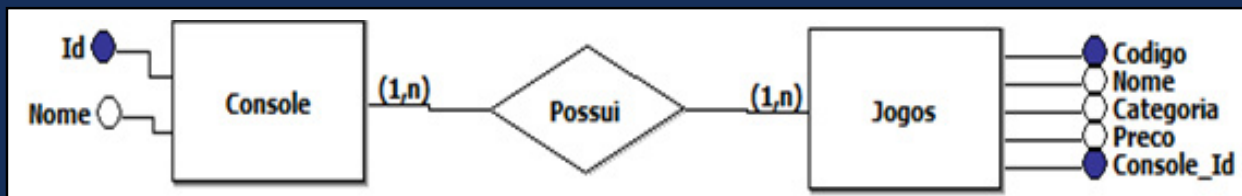
QUERY: comandos utilizados para a seleção de dados nas tabelas.

Alias: são complementos utilizados em conjunto com os comandos DML.



Questão para reflexão

Dado DER abaixo, construa um banco de dados.



- Cadastre os consoles: Xbox One e PS4;
- cadastre os seguintes jogos:
 - Fifa 16, Esporte, R\$ 150,00, XBOX One;
 - Call of Duty, Guerra, R\$ 100,00, PS4;





Questão para reflexão

- Batman, Aventura, R\$ 180,00, XBOX One;
- God of War, Guerra, R\$ 200,00, PS4.
- Selecione os jogos do Xbox One acima de R\$ 150,00;
- Selecione os jogos do PS4 em forma decrescente pelos valores;
- Modifique a categoria do jogo God of War para Aventura;
- Exclua todos os jogos de esporte da tabela.





Considerações Finais

- A identificação dos comandos DML (data Manipulation Language) e o entendimento e aplicação da sintaxe;
- A inserção de dados no bando de dados para “popular” as tabelas;
- A utilização da Query para realizar consultas no banco de dados, a fim de se abstrair informações;
- A utilização do comando UPDATE para modificar um ou mais registros em uma tabela;
- A utilização do comando DELETE para excluir um ou mais registros em uma tabela;
- A utilização dos *alias* para complementação dos comandos DML.



Referências

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

MIYAGUSKU, R. **Curso Prático de SQL**. São Paulo: Digerati Books, 2008.



Questão 1

1. Observe a afirmativa a seguir:

Os operadores _____ comparam os valores dentro de um _____ com um valor de busca, _____, modificação ou exclusão, que for atribuído na operação.

Assinale a sequência que completa as lacunas corretamente:

- a) lógicos / campo / seleção
- b) binários / SGBD / inserção
- c) lógicos / SGBD / inserção
- d) relacionais / tupla / seleção
- e) relacionais / campo / seleção



Questão 2

2. Assinale a sintaxe utilizada para inserção de dados em tabelas com campos não obrigatórios, que não se deseja entrar com dados:

- a) `INSERT <Nome_do_Campo> (<Nome_da_Tabela) VALUES (<Valor_Campo1>, <Valor_Campo2>, ...);`
- b) `INSERT <Nome_da_tabela> (<Campo1>, <Campo2>) ALTER VALUES TO (<Valor_Campo1>, <Valor_Campo2>, ...);`
- c) `INSERT <Nome_da_tabela> (<Campo1>, <Campo2>) VALUES (<Valor_Campo1>, <Valor_Campo2>, ...);`
- d) `INSERT <Nome_do_Campo> (<Nome_da_Tabela) TO VALUES (<Valor_Campo1>, <Valor_Campo2>, ...);`
- e) `INSERT <Nome_do_Campo> (<Nome_da_Tabela) TO (<Valor_Campo1>, <Valor_Campo2>, ...);`



Questão 3

3. Observe as afirmativas a seguir:

I – O comando INSERT quando combinado com alias pode proporcionar mais opções de utilização.

II – O comando SELECT pode ser combinado com WHERE, ORDER BY e alias.

III – O comando DELETE pode ser combinado com WHERE e alias.

Assinale a alternativa CORRETA:

- a) Somente a alternativa II está correta.
- b) Somente as alternativas II e III estão corretas.
- c) Somente as alternativas I e II estão corretas.
- d) Somente a alternativa I está correta.
- e) Todas as alternativas estão corretas.



Questão 4

4. Observe a tabela a seguir:

Veiculo

Placa	Carro	Cor	Valor
BZE 1012	Ferrari	Vermelha	R\$ 900.000,00
FHC 1515	Buggati	Preto	R\$ 600.000,00
MGM 5588	Camaro	Branco	R\$ 250.000,00

Um administrador de banco de dados modificar o valor do Camaro para R\$ 300.000,00. Assinale a alternativa com o comando correto:

- a) UPDATE Placa SET Valor = 300.000,00 WHERE VALUES = "MGM 5588";
- b) UPDATE Veiculo SET VALUES Valor = 300.000,00 WHERE Veiculo = "MGM 5588";
- c) UPDATE Placa WHERE Valor = 300.000,00 SET Placa = "MGM 5588";
- d) UPDATE Veiculo SET Valor = 300.000,00 WHERE Placa = "MGM 5588";
- e) UPDATE Veiculo VALUES SET Valor = 300.000,00 WHERE VALUES Placa = "MGM 5588";



Questão 5

5. Observe a tabela a seguir:

Veículo

Placa	Carro	Cor	Valor
BZE 1012	Ferrari	Vermelha	R\$ 900.000,00
FHC 1515	Buggati	Preto	R\$ 600.000,00
MGM 5588	Camaro	Branco	R\$ 250.000,00

Um administrador de banco de dados deve excluir o veículo Ferrari do banco de dados, devido à venda. Assinale a alternativa com o comando correto:

- a) DROP FROM Veiculo WHERE carro = “Ferrari”;
- b) DELETE FROM Veiculo ALTER columns WHERE carro = “Ferrari”;
- c) DELETE Veiculo WHERE carro = “Ferrari”;
- d) DROP Veiculo WHERE carro = “Ferrari”;
- e) DELETE FROM Veiculo WHERE carro = “Ferrari”;



Gabarito

1. Resposta: E.

Os operadores relacionais comparam os valores dentro de um campo com um valor de busca, seleção, modificação ou exclusão, que for atribuído na operação.

2. Resposta: C.

Para inserção em tabelas que não se deseja entrar obrigatoriamente com dados em todos os campos, a sintaxe SQL exige que os campos sejam declarados.

3. Resposta: B.

O comando INSERT não utiliza *alias*.

4. Resposta: D.

A sintaxe correta para modificar o preço deve ser utilizada, UPDATE <Nome_da_Tabela> SET <Coluna> <operador> <Novo_Valor> WHERE <Condição>;

5. Resposta: E.

A sintaxe correta para excluir um registro deve ser utilizada: DELETE FROM <Nome_da_Tabela> WHERE <Condição>.



Unidade 7

Comandos DCL (*Data Control Language*)

Objetivos

1. Compreender os conceitos e aplicações das autorizações de licenças por usuários.
2. Aplicar os comandos GRANT e REVOKE, no controle dos dados.
3. Compreender a necessidade de se obter o controle dos dados nos BDs.



Introdução

O terceiro e não menos importante, os comandos DCL preocupam-se com o acesso ao usuário e os direitos de permissão. Segundo Jobstraibizer (2009), esse grupo da linguagem SQL efetua o controle de autorização de dados e as licenças que os usuários para seleção, visualização ou manipulação dos dados nos BDs. Entre os dois comandos, desse grupo estão:

- Grant: utilizado para as autorizações dos usuários.
- Revoke: utilizado para remover ou restringir as permissões dos usuários.

Caro aluno, nesse tema vai ser agregado mais um assunto com o banco de dados, a segurança da informação, pois, ao se estruturar um BD para gerenciar um sistema, tem-se a necessidade e preocupação na garantia da integridade dos dados.

Para saber mais

A organização cert.br é um centro de estudos, respostas e tratamento de incidentes de segurança no Brasil. Um dos seus projetos se chama “Honeypots”, que são “iscas” deixados em diversos servidores, para efetuar análises de segurança. Disponível em: <<http://www.cert.br/projetos/>>. Acesso em: 05 jun. 2016.a

Segundo Mannino (2008), a proteção de um banco de dados está relacionada aos acessos não autorizados e as ações mal-intencionadas. Estando divididos em:

- Controle de acesso: é o controle dos acessos ao BD, por meio das regras de restrição as contas do usuário.

- Usuários: abrangem as pessoas que irão poder ter acesso a uma determinada base de dados.
- Domínio de segurança: é um conjunto de privilégios disponíveis para os usuários.
- Privilégios: são as permissões concedidas para um usuário ou um determinado grupo.

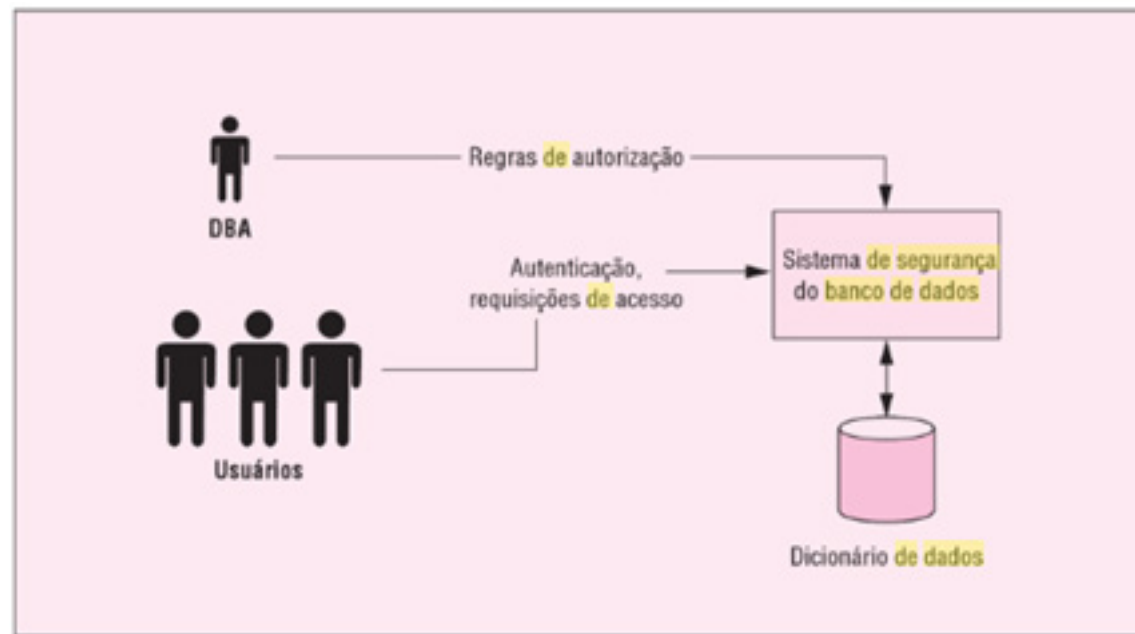
O SGBD deve permitir que o administrador do BD possa atribuir níveis de atribuição de privilégios, em que:

- permita estabelecer privilégios específicos por conta de usuário;

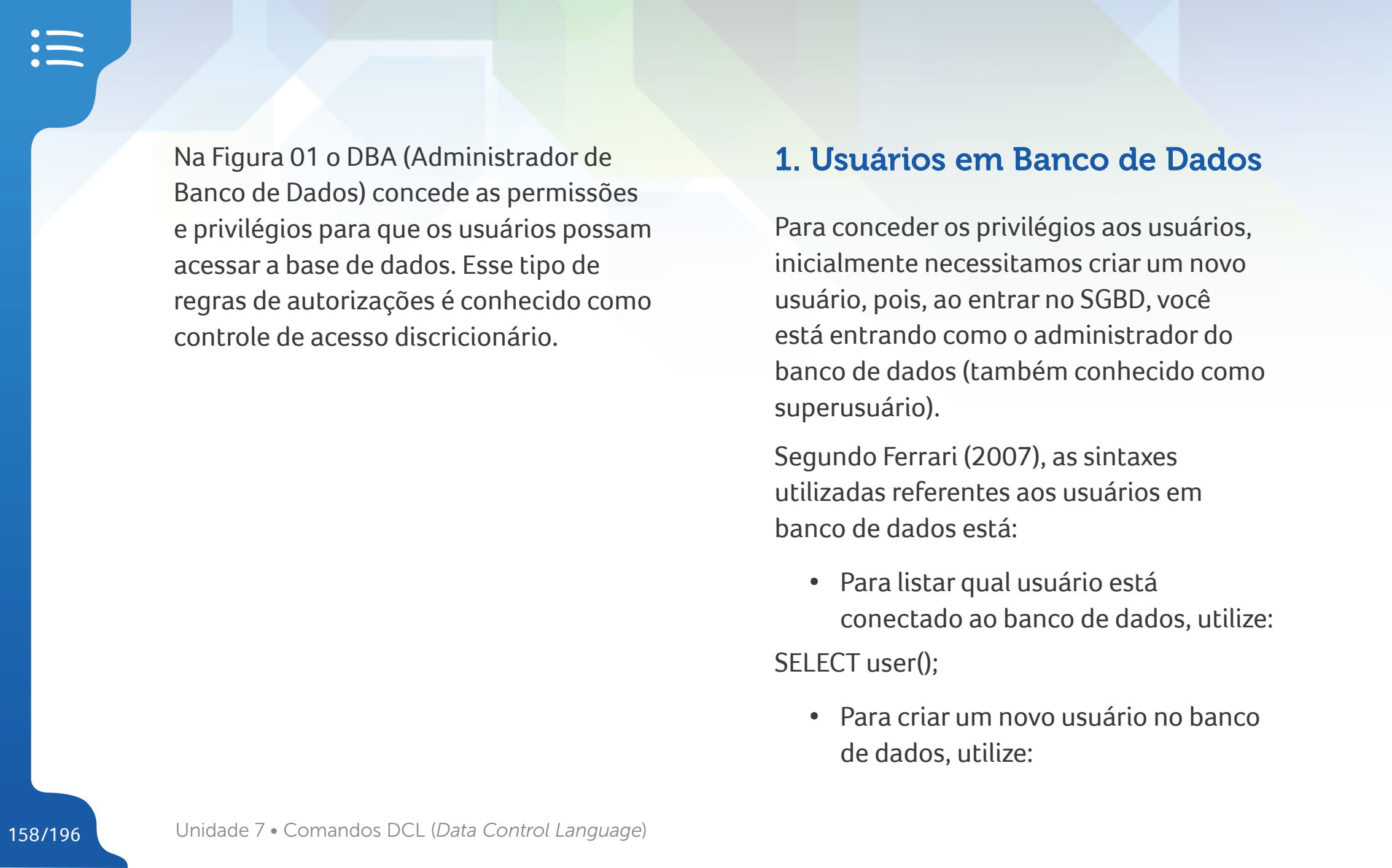
- permita controlar os privilégios de acesso ao banco de dados ou as tabelas;
- permita a revogação dos privilégios.

Esses três tipos de funções de segurança em banco de dados são representados na figura a seguir:

Figura 1 – Sistemas de segurança em Banco de dados



Fonte: Mannino (2008, p. 486)



Na Figura 01 o DBA (Administrador de Banco de Dados) concede as permissões e privilégios para que os usuários possam acessar a base de dados. Esse tipo de regras de autorizações é conhecido como controle de acesso discricionário.

1. Usuários em Banco de Dados


Para conceder os privilégios aos usuários, inicialmente necessitamos criar um novo usuário, pois, ao entrar no SGBD, você está entrando como o administrador do banco de dados (também conhecido como superusuário).

Segundo Ferrari (2007), as sintaxes utilizadas referentes aos usuários em banco de dados está:

- Para listar qual usuário está conectado ao banco de dados, utilize:

```
SELECT user();
```

- Para criar um novo usuário no banco de dados, utilize:



```
CREATE USER 'Nome_do_
Usuário'@'localhost' IDENTIFIED BY
“Senha_de_acesso”;
```

- Para listar todos os usuários e os seus respectivos privilégios, utilize:

```
SELECT * FROM mysql.user;
```

- Para especificar as colunas que irão retornar os usuários e os seus respectivos privilégios, utilize:

```
SELECT host, user, password FROM mysql.
user;
```

- Para excluir um usuário, utilize:

```
DROP USER 'Nome_do_
Usuário'@'localhost';
```

Para saber mais

A organização cert.br é um centro de estudos, respostas e tratamento de incidentes de segurança no Brasil. E disponibiliza uma cartilha de segurança para internet disponível em:<<http://cartilha.cert.br/livro/cartilha-seguranca-internet.pdf>>.

Acesso em: 06 jun. 2016.

2. Permissões dos Usuários

Antes de conceder as autorizações aos usuários, por meio do comando GRANT, será necessário você conhecer os tipos de privilégios. Segundo Mannino (2008), nas instruções GRANT é necessário especificar os privilégios, o objeto (tabela, coluna ou visão) e o usuário. Para compreender os privilégios, observe nas tabelas a seguir:

Quadro 1 – Privilégios comum em SQL

PRIVILÉGIO	DESCRIÇÃO
CREATE	Permite criar tabelas e BDs.
DROP	Permite deletar tabelas e BDs.
DELETE	Permite deletar registros das tabelas.
INSERT	Permite inserir registros nas tabelas.
SELECT	Permite selecionar os dados.
UPDATE	Permite alterar os registros nas tabelas.
ALL PRIVILEGES	Concede todos os privilégios.

Fonte: Mannino (2008, p. 487).

Repare que todos os privilégios foram estudados nos temas anteriores, e para conceder todos os privilégios deve ser utilizado “ALL PRIVILEGES”.

3. Comando GRANT

Para que o usuário criado nos bancos de dados possa ter direito dentro de uma base de dados, é necessário utilizar o comando GRANT. Segundo Mannino (2008), as instruções GRANT são utilizadas para autorizar um usuário a realizar determinadas tarefas dentro do BD. Ferrari (2007) define que,

- Para conceder privilégio para um usuário, utilize:

```
GRANT <privilégio> ON <Nome_do_
BD>.<Nome_da_Tabela> TO '<Nome_do_
usuário>'@'<localhost>';
```

Para dar acesso a todos os bancos de dados ou em determinada tabela, deve

ser utilizado o símbolo de asterisco (*), nos campos <Nome_do_BD> e <Nome_da_Tabela>.

- Para carregar os privilégios concedidos para um usuário, utilize:


```
FLUSH PRIVILEGES;
```

- Para listar os privilégios de um usuário, utilize:

```
SHOW GRANTS FOR <Nome_do_usuario>;
```

4. Comando REVOKE

Uma vez que os usuários foram criados e concedemos as permissões, em certas situações é necessário revogar esses direitos, e, para isso, deve ser utilizado o comando REVOKE. Segundo Mannino



(2008), nas instruções REVOKE são utilizadas para revogar as autorizações dentro de um BD.

Ferrari (2007) define que,

- Para revogar as permissões de um usuário, utilize:

```
REVOKE <privilégio> ON <Nome_do_
BD>.<Nome_da_Tabela> TO '<Nome_do_
usuário>'@'<localhost>';
```

Com isso, todos os privilégios e controle de segurança no bancos de dados podem ser gerenciados pelo administrador.

Para saber mais

A organização cert.br é um centro de estudos, respostas e tratamento de incidentes de segurança no Brasil. E disponibiliza uma cartilha com dicas de senhas disponível em: <<http://cartilha.cert.br/fasciculos/senhas/fasciculo-senhas.pdf>>. Acesso em: 06 jun. 2016.



Glossário

DBA: sigla que referencia o administrador do banco de dados.

Permissões: são as autorizações que os usuários podem possuir.

Revogação de permissões: as permissões de um usuário são retiradas.



Questão para reflexão

Um mercado necessita de um sistema de controle de funcionários. Para isso, é necessário desenvolver o banco de dados, conforme o mapeamento a seguir:

Funcionario (Matricula, Nome, Data_nasc, Cargo)

- a) Adicione 5 funcionários no banco de dados, sendo dois deles com cargo de gerentes.
- b) Crie um usuário para cada gerente e conceda todos os privilégios para ambos.
- c) Revogue o privilégio de um dos gerentes.





Considerações Finais

- A identificação dos comandos DCL (data Control Language) e o entendimento e aplicação das sintaxes;
- A compreensão dos privilégios para utilização dentro da sintaxe GRANT;
- A compreensão da utilização do comando GRANT para conceder privilégios aos usuários;
- A compreensão da utilização do comando REVOKE para revogar privilégios aos usuários.



Referências

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

JOBSTRAIBIZER, F. **Criação de Banco de Dados com MySQL**. São Paulo: Digerati Books, 2009.

MANNINO, M. V. **Projeto, Desenvolvimento de Aplicações & Administração de Banco de Dados**. São Paulo: McGraw-Hill Brasil, 2008.



Questão 1

1. Relacione as colunas conforme os níveis de atribuição de privilégios:

(1) Controle de acesso

(2) Usuários

(3) Domínio de segurança

(4) Privilégios

() É um conjunto de privilégios disponíveis para os usuários.

() Abrange as pessoas que irão poder ter acesso a uma determinada base de dados.

() É o controle utilizado em BD, por meio das regras de restrição às contas dos usuários.

() São as permissões concedidas para um usuário ou um determinado grupo.

Assinale a sequência correta:

a) 1 – 4 – 2 – 3.

b) 3 – 2 – 1 – 4.

c) 2 – 3 – 1 – 4.

d) 4 – 1 – 3 – 2.

e) 3 – 1 – 4 – 2.



Questão 2

2. Entre as permissões SQL utilizadas nos comandos, estão:

I – CREATE, DROP, ALTER e ALL PRIVILEGES.

II – SELECT, INSERT, UPDATE e ALL PRIVILEGES.

III – REVOKE, INSERT, CREATE e ALL PRIVILEGES.

Assinale a alternativa correta:

- a) Somente as alternativas I e II estão corretas.
- b) Somente as alternativas II e III estão corretas.
- c) Somente a alternativa I está correta.
- d) Somente a alternativa II está correta.
- e) Somente a alternativa III está correta.



Questão 3

3. Um administrador de banco deseja inserir um novo usuário para auxiliá-lo no gerenciamento. Assinale a alternativa com a sintaxe correta:

- a) `CREATE DATABASE 'Nome_do_Usuário'@'localhost' ADD "Senha_de_acesso";`
- b) `CREATE USER 'Nome_do_BD'@'localhost' IDENTIFIED BY "Nome_do_Usuário";`
- c) `CREATE USER 'Nome_do_Usuário'@'localhost' IDENTIFIED BY "Senha_de_acesso";`
- d) `CREATE TABLE 'Nome_do_BD'@'localhost' IDENTIFIED BY "Senha_de_acesso";`
- e) `CREATE 'Nome_do_Usuário'@'localhost' IDENTIFIED TO "Senha_de_acesso";`



Questão 4

4. Observe as afirmativas a seguir:

“[...] a instrução _____ é utilizada dentro do BD, para autorizar um usuário em realizar determinadas tarefas.”

“[...] a instrução _____ é utilizada dentro do BD, para revogar as autorizações de um usuário.”

Assinale a alternativa que complete as lacunas corretamente:

- a) GRANT / REVOKE
- b) ADD / CREATE
- c) GRANT / CREATE
- d) ADD / REVOKE
- e) ADD / DROP



Questão 5

5. O funcionário "Patrick" deve ter os seus privilégios revogados no banco de banco que gerencia o restaurante "Siri Cascudo". Assinale a alternativa com o comando correto:

- a) REVOKE CREATE, SELECT ON *.* TO 'Patrick'@'localhost';
- b) REVOKE ALL PRIVILEGES ON *.* TO 'Patrick'@'localhost';
- c) DROP ALL PRIVILEGES ON *.* TO 'Patrick'@'localhost';
- d) GRANT ALL PRIVILEGES ON *.* TO 'Patrick'@'localhost';
- e) DROP CREATE, SELECT ON *.* TO 'Patrick'@'localhost';



Gabarito

1. Resposta: B.

O SGBD deve permitir que o administrador do BD possa atribuir níveis de atribuição de privilégios, em que: permita estabelecer privilégios específicos por conta de usuário; permita controlar os privilégios de acesso ao banco de dados ou as tabelas; permita a revogação dos privilégios.

2. Resposta: E.

O comando ALTER não faz parte das permições

3. Resposta: C.

A sintaxe utilizada para criar um novo usuário deve ser: CREATE USER 'Nome_

do_Usuário'@'localhost' IDENTIFIED BY "Senha_de_acesso";

4. Resposta: A.

"[...] a instrução GRANT é utilizada dentro do BD para autorizar um usuário em realizar determinadas tarefas."

"[...] a instrução REVOKE é utilizada dentro do BD para revogar as autorizações de um usuário."

5. Resposta: B.

A sintaxe para revogar as permissões deve ser utilizada: REVOKE <privilégio> ON <Nome_do_BD>.<Nome_da_Tabela> TO '<Nome_do_usuario>'@'<localhost>';



Unidade 8

Big DATA

Objetivos

1. Compreender os conceitos e aplicações de mercado do Big Data.
2. Entender o papel do Big Data Analytics e a carência desse profissional no mercado.
3. Compreender as necessidades e aplicações do Hadoop no Big Data.



Introdução

Você já percebeu a quantidade de informações que nos cercam? Observe as questões:

- Qual a palavra mais pesquisada nos motores de busca pela internet?
- Qual o produto mais vendido no dia de hoje?
- Qual a comida mais vendida nos finais de semana?
- Qual o roteiro mais procurado nos feriados prolongados? Qual o poder aquisitivo desses turistas?


Todo esse volume de informação carrega consigo uma base de dados valiosíssima. O problema está em como organizar e utilizar todas essas informações. Para

resolver essa subárea de banco de dados, tem-se o Big Data.

Segundo Davenport (2014), o termo Big Data é considerado grande base de dados contidos nos repositórios, podendo estar alocado em diversos servidores, não necessariamente organizados de forma estruturada.

Para auxiliar na compreensão, conceitos e aplicações do Big Data, será necessário estudar a ferramenta Hadoop, considerada uma plataforma open source utilizada para processamento de grandes volumes de dados.

Esse tipo de ferramenta fornece dados para análise de negócios. As informações geradas na mineração de dados (*data*



mining) são utilizadas por grandes empresas, tais como: Walmart, Nike e Marcy's. Nesses casos, a análise é direcionada à pesquisa de mercado e perfil de consumo. O MIT (*Massachusetts Institute of Technology*) efetuou uma pesquisa para calcular o número de pessoas em um evento.

Para saber mais

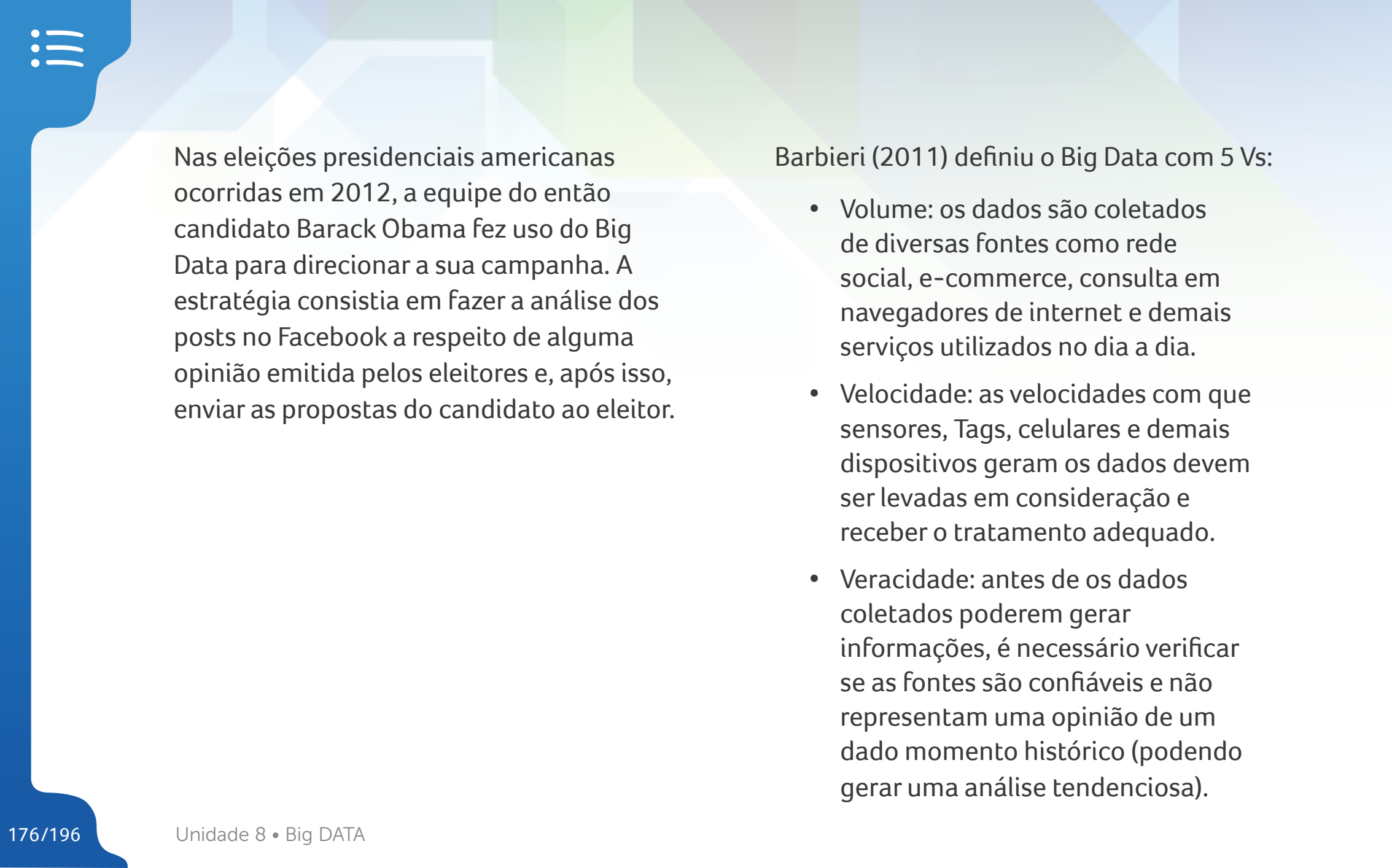
Os cases de sucesso na utilização do Big Data, são relatados no site Big Data Business no artigo “Você sabe o que é Big Data Analytics?”. Disponível em: <<http://www.bigdatabusiness.com.br/voce-sabe-o-que-e-big-data-analytics/>>. Acesso em: 07 jun. 2016.

Nesse tema, além das partes técnicas estudadas, você vai compreender como os conhecimentos técnicos em banco de dados podem ser direcionados ao negócio.

1. BIG DATA

A utilização de redes sociais, jogos on-line e pesquisa de produtos em buscadores de internet são apenas alguns serviços que utilizamos no dia a dia. Tais informações eram armazenadas, muitas vezes excluídas para não ocupar espaço nos servidores.

Segundo Dumbill (2013), os dados do Big Data são em grandes volumes e circulam com uma velocidade alta; a sua captação e tratativa se mostram como o maior desafio, para que as empresas possam tirar proveito dessas informações.



Nas eleições presidenciais americanas ocorridas em 2012, a equipe do então candidato Barack Obama fez uso do Big Data para direcionar a sua campanha. A estratégia consistia em fazer a análise dos posts no Facebook a respeito de alguma opinião emitida pelos eleitores e, após isso, enviar as propostas do candidato ao eleitor.

Barbieri (2011) definiu o Big Data com 5 Vs:

- Volume: os dados são coletados de diversas fontes como rede social, e-commerce, consulta em navegadores de internet e demais serviços utilizados no dia a dia.
- Velocidade: as velocidades com que sensores, Tags, celulares e demais dispositivos geram os dados devem ser levadas em consideração e receber o tratamento adequado.
- Veracidade: antes de os dados coletados poderem gerar informações, é necessário verificar se as fontes são confiáveis e não representam uma opinião de um dado momento histórico (podendo gerar uma análise tendenciosa).

- Variedade: a coleta e mineração dos dados podem ser estruturadas ou não, aproveitando informações como: e-mail, áudio, vídeo, cartões e demais fontes de consultas.
- Valor: identifica o quanto os dados coletados e tratados são importantes para as organizações, podendo agregar informações estratégicas para as empresas.

Porém, tais técnicas e conceitos relacionados ao Big Data não podem ser confundidos com espionagem digital. Em alguns casos, os dados coletados em grandes volumes por infraestruturas tidas como Big Data são utilizados de forma ilícita.

Para saber mais

Em 2013, o ex-funcionário da NSA, Edward Snowden, levou a público as espionagens feitas pelo governo americano, inclusive contra o Brasil. Disponível em: <<http://oglobo.globo.com/mundo/o-escandalo-da-espionagem-dos-eua-10191175>>. Acesso em: 07 jun. 2016.

Disponível em: <<http://www.tecmundo.com.br/politica/41800-espionagem-americana-no-brasil-o-que-e-preciso-saber-.htm>>. Acesso em: 07 jun. 2016.

Dois grandes problemas que as empresas de tecnologia da informação que utilizam o Big Data estão em:

- Processamento: o poder de processamento dos servidores deve permitir o tratamento dos dados.
- Armazenamento: a capacidade de “guardar” os dados.

Para driblar tais problemas ou limitações, são utilizadas tecnologias como cluster, servidores cloud e computadores com maiores capacidades para atender às necessidades de análises.

2. Analista de Dados

Os profissionais de análise de dados, conhecidos também como Big Data Analytics, são considerados os cientistas de Big Data. São profissionais que têm a capacidade técnica de receber, analisar

e disponibilizar os dados. Não basta somente capacidade técnica, mas também esse profissional deve ter criatividade e percepção aguçada.

Os profissionais de análise de dados são cada vez mais raros no mercado. Isso acontece porque o conhecimento técnico necessário para desempenhar a função requer os conhecimentos em:

- Linguagem de programação em banco de dados, como: SQL, SQLServer, Postgree, DB2 etc.;
- Matemática e estatística;
- Regras e tendência de negócios;
- Tecnologia Hadoop.



Por esses motivos e alegações, o profissional de análise de dados é um dos profissionais mais procurados de tecnologia da informação.

3. HADOOP

Goldschmidt (2015) define que o Hadoop é uma das tecnologias mais conhecidas e utilizadas no Big Data, sendo um projeto de código aberto desenvolvido pela Apache.

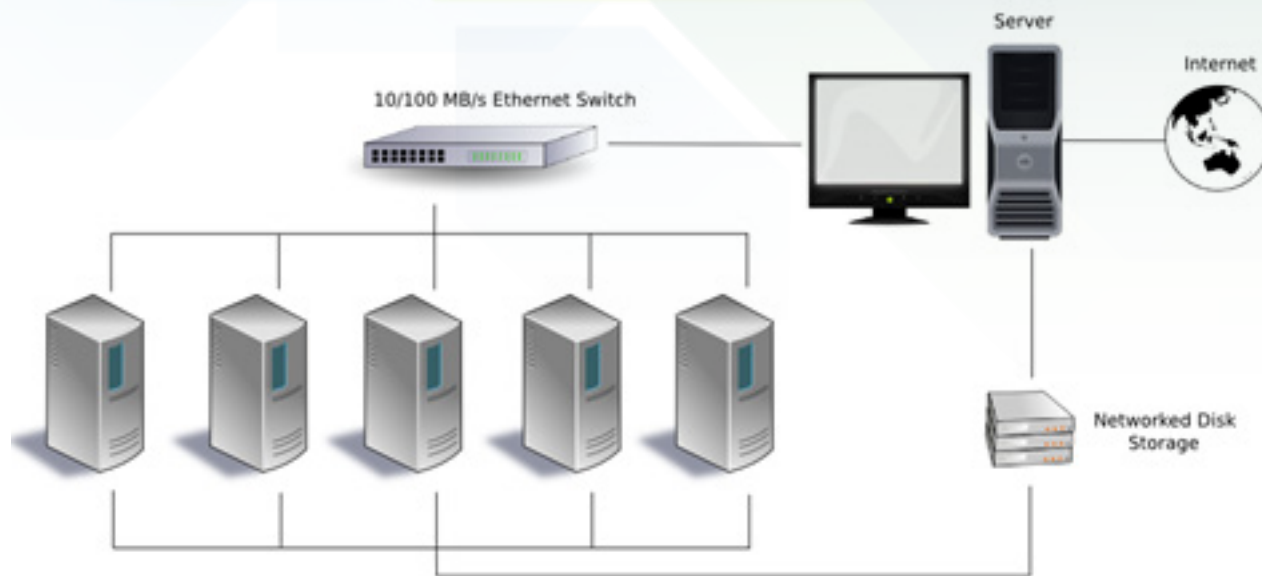
Figura 1 – Logotipo Hadoop



Fonte: <<https://hadoop.apache.org/>>

O seu funcionamento efetua o processamento em servidores geograficamente distribuídos, trabalhando sempre com grandes volumes de dados. Isso só é possível porque a sua escalabilidade permite estruturar a arquitetura em cluster.

Figura 2 – Exemplo de arquitetura de redes em cluster



Fonte: <https://en.wikipedia.org/wiki/Computer_cluster>

Conforme observado na Figura 2, no cluster os computadores, servidores e dispositivos de armazenamento são “interligados”, com compartilhamento dos seus recursos para aumento da capacidade de captura dos dados, processamento e arquivamento.



Para saber mais

No vídeo do programa “Olhar Digital”, são demonstrados os conceitos, aplicações e vantagens na utilização de supercomputadores (Cluster). Disponível em: <<https://www.youtube.com/watch?v=016umo07Rog>>.

Acesso em: 07 jun. 2016.



Glossário

Big Data: conceito de coleta e análise de grandes volumes de dados, para fins de negócios.

Big Data Analytics: profissional responsável por aplicar as tecnologias disponíveis para promover a coleta e análise de dados.

Hadoop: software utilizado para aplicação do Big Data.



Questão para reflexão

Uma empresa produtora de cerveja artesanal deseja desenvolver um novo produto que leve em seu ingrediente algum sabor típico do Brasil. Porém, os mestres cervejeiros não sabem qual sabor poderia representar um país com tantas diversidades.

Para descobrir qual o sabor que representa o Brasil, o gerente de projetos, solicitou que o departamento de marketing fizesse uma pesquisa de campo e o departamento de tecnologia da informação, uma pesquisa na internet. De que forma o Big Data poderia contribuir na pesquisa?





Considerações Finais

- A compreensão dos conceitos do Big Data nas bases de dados, utilizadas em diversos domínios;
- O entendimento e papel do Big Data Analytics frente às necessidades e demanda do mercado;
- A aplicação do Hadoop no Big Data para captura, tratamento e disponibilização das informações;
- A conceituação de como a tecnologia da informação pode fornecer dados para um diferencial competitivo de mercado.



Referências

BARBIERI, C. **BI2-Business Intelligence**: Modelagem e Qualidade. Rio de Janeiro: Digeratti, 2011.

DAVENPORT, H. T. **Big Data at Work**. Harvard Business Review Press, 2014.

DUMBILL, E. Making Sense of Big Data. Disponível em:<<http://online.liebertpub.com/doi/pdf/10.1089/big.2012.1503>>. Acessado em: 08 Jun 2016.

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



Questão 1

1. Os dados são gerados a todo o momento por diversos meios como: redes sociais, e-mail, sites de compras, buscas pela internet entre diversos outros. Assinale a alternativa que descreva os tipos de dados que possibilitam análises:

- a) Pessoas, lugares e jornais.
- b) Históricos de navegação, pessoas, sensores.
- c) Lugares, pessoas e rótulos de produtos.
- d) E-mail, vídeo e sensores.
- e) Sensores, RFID e rótulos de produtos.



Questão 2

2. Segundo Barbieri (2013), os conceitos do Big Data podem ser definidos em 5 Vs. Relacione as colunas quanto aos 5 Vs e os seus respectivos tipos:

(A) Volume

(B) Velocidade

(C) Veracidade

(D) Variedade

(E) Valor

() A coleta e mineração dos dados estruturadas ou não, por meio de diversas fontes.

() Verificação de fontes de informações confiáveis.

() Os dados são coletados de diversas fontes de serviços utilizados no dia a dia.

() Identifica o quanto os dados coletados e tratados são importantes para as organizações.

() As velocidades com que os dados são coletados e tratados.



Questão 2

Assinale a alternativa com a sequência correta:

- a) D – C – A – E – B .
- b) B – A – C – D – E.
- c) A – B – E – D – C.
- d) D – B – A – C – E.
- e) E – C – D – A – B.



Questão 3

3. Para trabalhar com Big Data, os profissionais de tecnologia da informação precisam estar cada vez mais dominando novas técnicas e se atualizando. Observe as afirmativas a seguir:

- I – O Big Data Analytics deve dominar diversas linguagens de programação de banco de dados.
- II – O Big Data Analytics deve dominar diversos idiomas para ampliar as fontes de análises.
- III – O Big Data Analytics deve possuir uma boa network para possibilitar o acesso aos dados das empresas.

Assinale a alternativa correta:

- a) Somente as alternativas I e II estão corretas.
- b) Somente as alternativas I e III estão corretas.
- c) Somente a alternativa I está correta.
- d) Somente a alternativa II está correta.
- e) Somente a alternativa III está correta.



Questão 4

4. Observe as afirmativas a seguir e marque (V) verdadeiro ou (F) falso:

- () O Big Data Analytics é um profissional que faz análise de grandes volumes de dados, em escassez no mercado de tecnologia da informação.
- () O software Hadoop efetua análise de dados, possibilitando estruturar o seu funcionamento em cluster.
- () Os dados capturados podem possuir diversas fontes, como: e-commerce, e-mail, vídeos e sensores.
- () Os dados devem ser acumulados em um único computador para que permita a análise.
- () Após a mineração dos dados, as informações são disponibilizadas em um repositório de domínio público.

Assinale a alternativa correta:

- a) V – F – V – F – F.
- b) F – V – F – V – V.
- c) F – F – V – V – F.
- d) F – F – F – V – V.
- e) V – V – V – F – F.

5. Observe a afirmativa a seguir:

O software Hadoop efetua o processamento em _____ geograficamente distribuídos, trabalhando sempre com grandes volumes de _____. Isso só é possível porque a sua escalabilidade permite estruturar a arquitetura em _____.

Figura 1 – Logotipo Hadoop



Fonte: <<https://hadoop.apache.org/>>

Assinale a alternativa que complete as lacunas:

- a) cloud / informações / rede
- b) servidores / dados / cluster
- c) computadores / informações / centralizadas
- d) cloud / dados / cluster
- e) servidores / informações / rede



Gabarito

1. Resposta: D.

O Big Data utiliza dados extraídos de redes sociais, jogos on-line, pesquisa de produtos em buscadores de internet, entre outros serviços que utilizamos no dia a dia.

2. Resposta: A.

Volume: os dados são coletados de diversas fontes como rede social, e-commerce, consulta em navegadores de internet e demais serviços utilizados no dia a dia. Velocidade: as velocidades com que sensores, Tags, celulares e demais dispositivos geram os dados devem ser levadas em consideração e receber o tratamento adequado. Veracidade: antes

de os dados coletados poderem gerar informações, é necessário verificar se as fontes são confiáveis e não representam uma opinião de um dado momento histórico (podendo gerar uma análise tendenciosa). Variedade: a coleta e mineração dos dados podem ser estruturadas ou não, aproveitando informações como: e-mail, áudio, vídeo, cartões e demais fontes de consultas. Valor: identifica o quanto os dados coletados e tratados são importantes para as organizações, podendo agregar informações estratégicas para as empresas.

3. Resposta: C.

Para ser um Big Data Analytics, deve-se ter conhecimentos em linguagem de programação em banco de dados,



como: SQL, SQLServer, Postgree, DB2 etc.; matemática e estatística; regras e tendência de negócios; tecnologia Hadoop;

escalabilidade permite estruturar a arquitetura em cluster.

4. Resposta: E.

As técnicas permitem que grande volume de dados seja capturado, processado e analisado, para que gerem informações que são utilizadas pela empresa no diferencial competitivo.

5. Resposta: B.

O software Hadoop efetua o processamento em servidores geograficamente distribuídos, trabalhando sempre com grandes volumes de dados. Isso só é possível porque a sua

