



Fundamentos de banco de dados

Fundamentos de banco de dados

Nathalia dos Santos Silva
Gisele Alves Santana

© 2018 por Editora e Distribuidora Educacional S.A.

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

Presidente

Rodrigo Galindo

Vice-Presidente Acadêmico de Graduação

Mário Ghio Júnior

Conselho Acadêmico

Alberto S. Santana

Ana Lucia Jankovic Barduchi

Camila Cardoso Rotella

Cristiane Lisandra Danna

Danielly Nunes Andrade Noé

Emanuel Santana

Grasiele Aparecida Lourenço

Lidiâne Cristina Vivaldini Olo

Paulo Heraldo Costa do Valle

Thatiane Cristina dos Santos de Carvalho Ribeiro

Revisão Técnica

Sérgio Eduardo Nunes

Editorial

Adilson Braga Fontes

André Augusto de Andrade Ramos

Cristiane Lisandra Danna

Diogo Ribeiro Garcia

Emanuel Santana

Erick Silva Griep

Lidiâne Cristina Vivaldini Olo

Dados Internacionais de Catalogação na Publicação (CIP)

Silva, Nathalia dos Santos

S586 Fundamentos de banco de dados / Nathalia dos Santos

Silva, Gisele Alves Santana. Londrina : Editora e

Distribuidora Educacional S.A., 2018.

136 p.

ISBN 978-85-522-0293-6

1. Banco de dados. I. Santana, Gisele Alves. II. Título.

CDD 005.74

2018

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

Sumário

Unidade 1 Modelagem de Banco de Dados	7
Seção 1 - Conceitos e Terminologias de um Banco de Dados	11
1.1 Dados e Informações	11
1.2 Elementos de um Banco de Dados	13
1.2.1 Campo	13
1.2.2 Registro	14
1.2.3 Tabela	14
1.3 Índices	16
1.4 Chaves	16
1.4.1 Chave primária	17
1.4.2 Chave estrangeira	17
1.5 Sistema de Gerenciamento de Banco de Dados (SGBD) e Sistema de Banco de Dados	19
Seção 2 - Modelagem de um Banco de Dados	21
2.1 Abstração de dados	21
2.2 Modelos de Banco De Dados	22
2.2.1 Modelo Conceitual	23
2.2.2 Modelo Lógico	24
2.2.3 Modelo Físico	25
2.2.4 Modelo Relacional	26
2.2.1 Restrições de Integridade	28
2.2.2 Operações em Relações	29
2.2.3 Inserção	29
2.2.4 Remoção	30
2.2.5 Modificação	31
2.3 Álgebra Relacional	32
2.3.1 Operação de Seleção (σ)	32
2.3.2 Projeção (π)	34
2.3.3 União	34
2.3.4 Interseção	36
Unidade 2 Projeto de banco de dados	43
Seção 1 - Entidades e Atributos	48
1.1 Entidades	48
1.2 Atributos	49
1.2.1 Atributo Simples x Composto	50
1.2.2 Atributo de Valor único x Multivvalorado	51
1.2.3 Armazenados x Derivados	51
Seção 2 - Relacionamentos	54
2.1 Relacionamentos	54
2.1.1 Relacionamentos Unários	54
2.1.2 Relacionamentos Binários	56
2.1.4 Relacionamentos Ternários	56
2.2 Cardinalidade	57
2.2.1 Cardinalidade em atributos	57
2.2.2 Cardinalidade em relacionamentos	58
2.2.2.1 Cardinalidade Mínima	58
2.2.2.2 Cardinalidade Máxima	58
2.2.3 Cardinalidades e suas possíveis combinações	58

2.2.3.1 Um para um	59
2.2.3.2 Um para muitos	60
2.2.3.3 Muitos para um	61
2.2.3.4 Muitos para muitos	62
2.2.3.5 Cardinalidade em relacionamentos ternários	63
Seção 3 - Normalização	66
3.1 Primeira Forma Normal (1FN)	66
3.1.1 Dependência parcial ou total	68
3.2 Segunda Forma Normal (2FN)	69
3.3 Terceira Forma Normal (3FN)	70
Unidade 3 Sistemas de gerenciamentos de bancos de dados	79
Seção 1 - POSTGRESQL: Conceitos, instalação e configuração	82
1.1 Sistemas de Gerenciamento de Banco de Dados (SGBDs)	82
1.2 Arquitetura Cliente Servidor	83
1.3 PostgreSQL	86
1.3.1 Instalação	86
Seção 2 - pgAdmin: Criação de um banco de dados e tabelas	93
2.1 PgAdmin	93
2.1.1 Criando um banco de dados com o pgAdmin	94
2.1.2 Criando tabelas com o pgAdmin	97
Unidade 4 Banco de dados não convencionais	113
Seção 1 - Dados Avançados	116
1.1 Dados temporais	116
1.2 Dados espaciais	117
1.3 Dados de multimídia	120
Seção 2 - OLAP e Data Ware Housee	124
2.1 OLAP	124
2.2.1 Data Mining	126
2.2.2 Criando tabelas com o pgAdmin	128

Apresentação

Olá aluno! Tudo bem?

Você deve ter observado o quanto as informações são essenciais nos dias de hoje. Antes, a riqueza de uma empresa era pautada em seus estoques e maquinários, atualmente, é o conhecimento que essa empresa detém! As informações sobre vendas, clientes, histórico de mercado, entre outros, são dados que a empresa vai adquirindo ao longo do tempo, porém, se ela não tiver um processo organizado de armazenamento e disponibilização desses dados eles só serão um agrupamento de dados. As empresas estão interessadas na riqueza que pode estar atrelada àquelas informações, elas querem conhecer seus clientes.

Como um bom vendedor faz para ser lembrado e fidelizar o cliente, o banco de dados e suas ferramentas devem fazê-lo para milhares de clientes. Os conceitos para garantir um projeto que atenda com satisfação as demandas dos clientes deve passar por um bom projeto e modelagem, e por consultas adequadas. Adicionalmente, com o avanço da tecnologia, os projetos de banco de dados estão cada vez mais diferenciados. É importante lembrar que quando falamos de informação, usuário e gestão, estão subentendidos os conceitos de rapidez, robustez, segurança, flexibilidade e principalmente integridade. Neste livro, você é levado a conhecer esse assunto tão consolidado e ao mesmo tempo tão atual na área da computação.

Modelagem de Banco de Dados

Gisele Alves Santana

Objetivos de aprendizagem

Nesta unidade, você será levado a compreender a diferença entre dados e informações, assim como conhecer os principais conceitos relacionados aos bancos de dados, como: campo, registro, tabela e chave. Você também irá compreender como são organizados os dados em um banco de dados, além de conhecer os tipos de modelos de dados utilizados para a modelagem de um banco de dados.

Seção 1 | Conceitos e terminologias de um Banco de Dados

Nesta seção você estudará os principais conceitos relacionados aos Bancos de Dados, conhecendo seus elementos básicos e as terminologias mais importantes. Também serão apresentadas as diferenças entre banco de dados, sistemas gerenciadores de banco de dados e sistemas de banco de dados, explicando o enfoque de cada definição.

Seção 2 | Modelagem de um Banco de Dados

Nesta seção, você vai aprender a importância da modelagem de dados, bem como os principais tipos de modelagem. Será apresentado e exemplificado o modelo relacional, muito utilizado para a modelagem de relações em um sistema de gerenciamento de banco de dados relacional. Nesta seção, é explicada a importância e os tipos de integridade de dados, assim como as operações de inserção, remoção e modificação dos dados. Para finalizar, é brevemente apresentada a álgebra relacional e suas principais operações.

Introdução à unidade

Existem vários tipos de banco de dados e eles estão presentes na nossa vida há muito tempo. Por exemplo, a lista telefônica pode ser considerada um banco de dados. Antigamente, as empresas armazenavam suas informações em arquivos físicos, porém o surgimento e a evolução dos computadores possibilitaram o armazenamento de dados de modo digital. Assim, os bancos de dados evoluíram e se tornaram imprescindíveis para muitos sistemas de informação. Atualmente, os Bancos de Dados são amplamente utilizados, representando uma parte fundamental de quase todas as empresas, como: bancos, universidades, indústrias etc.

Um banco de dados (BD) pode ser entendido como um conjunto de dados interrelacionados que representam informações sobre um domínio específico (SILBERSCHATZ; KORTH, 2006). Um BD consolida registros previamente armazenados em arquivos separados em uma fonte comum de registros de dados. As informações armazenadas em um BD são independentes dos programas aplicativos que os utilizam e do tipo de dispositivo de armazenamento nos quais estão armazenados.

Em um BD relacional os dados são armazenados de forma estruturada e com a menor redundância possível. Um banco de dados permite colocar dados à disposição de usuários para uma consulta, inserção, atualização ou exclusão. Uma das grandes vantagens da utilização dos BDs é a possibilidade de poderem ser acessados por vários usuários, ao mesmo tempo.

Todo sistema de banco de dados deve apresentar uma boa modelagem e projeto, que visam a descrição e organização das informações com o objetivo de que o sistema obtenha desempenho satisfatório e possua fácil manutenção. Dessa maneira, nesta unidade, você será levado a analisar os tipos fundamentais de modelos de bancos de dados. Esses modelos têm o propósito de descrever os tipos de informações armazenadas em um banco de dados e os relacionamentos estabelecidos entre eles.

Os bancos de dados podem ser modelados de acordo com três perspectivas: conceitual, lógica e física. A modelagem conceitual é utilizada como uma representação de alto nível, considerando

exclusivamente o ponto de vista do usuário. Já a modelagem lógica agrupa alguns detalhes de implementação das estruturas de dados. Por último, a modelagem física demonstra como os dados são fisicamente armazenados.

Assim, você irá verificar que é possível descrever os modelos em diferentes níveis de abstração e com diferentes objetivos. Também será estudado o modelo relacional, suas terminologias e exemplos de relações. Agregado ao modelo relacional é apresentado algumas operações básicas da álgebra relacional, que realiza a manipulação de relações utilizando operadores específicos.

Na Seção 1, você vai estudar os principais conceitos relacionados aos Bancos de Dados, conhecendo seus elementos básicos e as terminologias mais importantes. Na Seção 2, serão apresentados os principais modelos de dados, principalmente o Modelo Relacional.

Seção 1

Conceitos e terminologias de um Banco de Dados

Introdução à seção

Nesta seção, você vai estudar os conceitos principais relacionados aos Bancos de Dados, assim como conhecer seus elementos básicos e as terminologias. Também são apresentadas as diferenças entre banco de dados, sistemas gerenciadores de banco de dados e sistemas de banco de dados.

1.1 Dados e Informações

Antes de começarmos nossos estudos sobre Banco de Dados é necessário entender a diferença entre dados e informações. Muitas pessoas consideram esses dois conceitos como palavras sinônimas, porém existem diferenças fundamentais entre elas.

Um **dado** pode ser definido como um fato bruto, em sua forma primária, que em muitas vezes não faz sentido sozinho. Algumas características que podem ser citadas em relação a um dado são:

- Um dado é um registro.
- Um dado é físico.
- Um dado necessita de processamento.

Já a **informação** consistem em um agrupamento de dados de forma organizada, apresentando o significado dos dados de maneira que possa ser interpretada pelas pessoas e gerar conhecimento. A Figura 1.1 ilustra a hierarquia entre esses três conceitos.

Figura 1.1 | Dados e informações

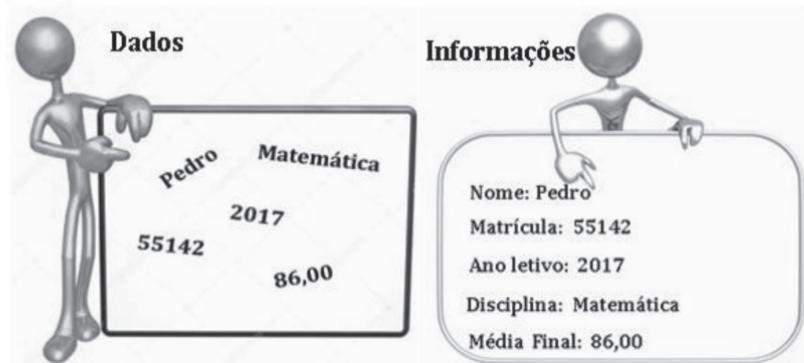


Fonte: elaborada pela autora.

Por exemplo, o número 1500 isoladamente não faz nenhum sentido, pois se trata de um dado. Porém, a seguinte afirmação: "Ano do descobrimento do Brasil: 1500". A afirmação tem sentido, então se trata de uma informação.

Outro exemplo que ilustra a diferença entre dados e informações é apresentado na Figura 1.2. Percebe-se que os dados são fatos isolados e sem sentido. Já a informação é a organização dos dados de forma que se possa ter entendimento dos mesmos. Nesse sentido, um banco de dados é uma estrutura de dados organizada que permite a extração de informações.

Figura 1.2 | Exemplo de dados e informações



Fonte: elaborada pela autora.

Para esclarecer ainda mais a diferença entre dados e informações, considere o cenário de um cliente que realiza uma compra em um supermercado. Para levantar os dados referentes a esse cenário, algumas perguntas podem ser feitas, como:

- Quando? (dia/hora)
- Quanto? (valor)
- Como pagou? (cash, cartão, cheque, vale)
- Quem? (identificação do cliente)

Para se descobrir as informações referentes ao cenário, podem ser utilizadas as seguintes perguntas:

- Porque o cliente escolheu o supermercado?
- O cliente é assíduo?

- O cliente tem crédito?

Com essas estratégias, fica mais fácil entender a diferença entre esses dois conceitos muito importantes e separar o que é “dado” do que é “informação”.

1.2 Elementos de um Banco de Dados

Os elementos principais de um BD são os campos, os registros, as tabelas e os relacionamentos. A seguir, cada elemento é explicado detalhadamente.

1.2.1 Campo

Um campo pode ser definido como a unidade mínima de armazenamento para os valores de uma tabela. Um campo possui um tipo de dado, que pode ser a característica de um lugar, evento, objeto ou pessoa (SILBERSCHATZ; KORTH, 2006). Suponha que se deseja armazenar as informações de um aluno em um BD. Para armazenar essas informações é necessário dividir em algumas partes, nas quais cada campo deve receber um nome de identificação, o tamanho máximo e o tipo de dado (real, caractere, inteiro etc.) que poderá armazenar.

A Figura 1.3 ilustra um exemplo de como os campos são organizados em um BD. Observa-se que cada campo é associado a um tipo de dado que pode armazenar. O campo “Nome” armazena apenas dados do tipo caractere, já o campo “Matrícula” armazena dados do tipo inteiro. Os campos podem ser identificados nas colunas de uma tabela.

Figura 1.3 | Exemplo de campos

Campo				
Nome	Matrícula	Curso	Ano letivo	Coeficiente de rendimento
Pedro Alencar	55143	Computação	2017	9,2

Fonte: elaborada pela autora.

Uma das tarefas mais importantes para a implementação de um BD consiste na definição dos campos que farão parte de uma tabela. Existe a possibilidade de ser criar campos com uma propriedade chamada de autoincremento, que aumenta automaticamente o valor quando

novos registros são inseridos. Os campos podem conter expressões matemáticas, como:

*valor_total(quantidade*preco);*

1.2.2 Registro

Um registro é uma coleção de campos inter-relacionados e identifica a entrada de um item exclusivo de informação em uma tabela. Os registros também são chamados de tuplas e podem ser associados às linhas de uma tabela (SILBERSCHATZ; KORTH, 2006). Suponha uma tabela com 100 linhas, pode-se dizer que a mesma possui 100 registros ou 100 tuplas. Os registros possuem o mesmo tipo, por exemplo, um registro que armazena informações sobre “alunos” não pode armazenar informações sobre “cliente”. A Figura 1.4 ilustra o exemplo de um registro.

Figura 1.4 | Exemplo de registros

Nome	Matrícula	Curso	Ano letivo	Coeficiente de rendimento
Pedro Alencar	55143	Computação	2017	9,2
Julia Busquim	55097	Biologia	2017	8,5
Sandra Oliveira	55199	Pedagogia	2017	9,4

Fonte: elaborada pela autora.

1.2.3 Tabela

Uma tabela representa um objeto e contém todos os dados de um BD. Os dados são organizados de maneira lógica em um formato de linhas e colunas, análogo a uma planilha. Cada linha representa um registro exclusivo e cada coluna representa um campo. Uma tabela que contém dados dos alunos de uma universidade pode conter uma linha para cada aluno e colunas representando as informações sobre o aluno, como: nome, matrícula, curso e coeficiente de rendimento.

As linhas de uma tabela são chamadas de registros. As colunas são tipicamente consideradas como os campos da tabela, caracterizado os tipos de dados que são armazenados na tabela (numéricos, alfanuméricos, datas, coordenadas etc.). A Figura 1.5 mostra o exemplo de uma tabela.

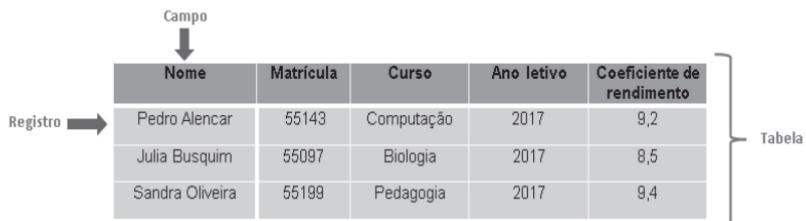
Figura 1.5 | Exemplo de tabela

Nome	Matrícula	Curso	Ano letivo	Coeficiente de rendimento
Pedro Alencar	55143	Computação	2017	9,2
Julia Busquim	55097	Biologia	2017	8,5
Sandra Oliveira	55199	Pedagogia	2017	9,4

Fonte: elaborada pela autora.

Uma aplicação apenas consegue acessar os dados se conhecer o nome da tabela e do campo de onde se deseja recuperar a informação. A Figura 1.6 ilustra todos os principais elementos de um banco de dados.

Figura 1.6 | Principais elementos de um Banco de Dados



Fonte: elaborada pela autora.

Dessa maneira, esses conceitos podem ser hierarquizados de acordo com a Figura 1.7.

Figura 1.7 | Hierarquia dos elementos de um banco de dados



Fonte: elaborada pela autora.

1.3 Índices

Os índices de um banco de dados podem ser comparados ao índice de um livro. Para procurar um determinado assunto em um livro, utiliza-se o índice, que indica a página na qual o assunto desejado se encontra. Nos BDs, os índices possuem a mesma finalidade, proporcionando rapidez na busca de um dado. Os índices também fornecem uma maneira alternativa de acesso aos dados, não modificando a posição física dos mesmos. A Figura 1.8 mostra um exemplo de aplicação dos índices em um BD.

Existem dois tipos de índices: simples e compostos. Um índice simples possui apenas um campo, já o índice composto tem vários campos.

Os campos utilizados para a definição dos índices são chamados de campos de indexação. Os índices possuem somente o valor do campo de indexação e ponteiros que direcionam para o registro adequado dentro da tabela. No exemplo da Figura 1.8 percebe-se que os índices são os “nomes” dos alunos, que apontam para o registro da tabela que contém a informação específica para cada um destes alunos.

Figura 1.8 | Exemplo de índices

Nome	Nome	Matrícula	Curso	Ano letivo	Coeficiente de rendimento
Ana	Pedro Alencar	55143	Computação	2017	9,2
Guilherme	Julia Busquim	55097	Biologia	2017	8,5
Julia	Sandra Oliveira	55199	Pedagogia	2017	9,4
Pedro	Guilherme Dias	55013	Engenharia	2017	7,8
Sandra	Ana Silva	55002	Marketing	2017	8,3

Fonte: elaborada pela autora.

1.4 Chaves

As chaves são responsáveis por distinguir um registro dos demais registros, definindo que cada registro é único dentro de uma tabela. As chaves também são responsáveis pelos relacionamentos entre as tabelas de um BD.

As chaves possuem grande importância para a modelagem de dados, pois definem restrições que garantem a integridade dos dados. Os dois tipos mais importantes e utilizados de chaves são: chave

primária e chave estrangeira.

1.4.1 Chave primária

Uma chave primária é um atributo ou a combinação de vários atributos que possuem a característica de identificar cada linha de uma tabela de maneira única (SILBERSCHATZ; KORTH, 2006). Cada tabela deve conter uma chave primária, capaz de identificar os registros de forma exclusiva. O principal objetivo de uma chave primária é evitar que existam registros duplicados em uma tabela.

Assim, com essa chave é criada uma identificação única para cada registro, fornecendo segurança para que as aplicações possam acessar, excluir ou alterar dados sem correr o risco de apagar ou alterar dois campos da tabela ao mesmo tempo.

Para a escolha dos campos que irão definir uma chave primária devem ser considerados os seguintes aspectos:

- Modificação – o campo chave não deve ser alterado.
- Tamanho do campo – campos menores são atualizados mais rapidamente.
- Usar preferencialmente valores que são calculados pelo próprio SGBD (autoincremento).

1.4.2 Chave estrangeira

A chave estrangeira é responsável por criar o elo físico entre as tabelas. A chave estrangeira ocorre quando um atributo de uma relação for chave primária em outra relação. Em outras palavras, sempre que houver o relacionamento 1:N entre duas tabelas, a tabela 1 receberá a chave primária e a tabela N receberá a chave estrangeira.

No exemplo apresentado na Figura 1.9, observa-se duas tabelas: “Livro” e “Pedidos do Livro”. Observa-se que o campo “Codigo_livro” existe nas duas tabelas. Na tabela “Livros” esse campo é a chave primária, já que cada livro deve ser único.

Figura 1.9 | Exemplo de chaves

Tabela Livro	Tabela Pedidos do Livro
<ul style="list-style-type: none">- Código_livro- Livro- Escritor- Título- Quantidade_paginas- Resumo	<ul style="list-style-type: none">- Numero_pedido- Código_livro- Quantidade

Fonte: elaborada pela autora.

Na tabela “Pedidos do Livro”, o campo “Código_livro” pode constar várias vezes, pois se pode ter vários pedidos de clientes diferentes para um mesmo livro. Como esse campo já é chave primária na tabela “Livro”, então ele será uma chave estrangeira na tabela “Pedidos do Livro”.

Dessa maneira, a Figura 1.10 ilustra as duas tabelas com suas chaves primária e secundária.

Figura 1.10 | Chaves primária e estrangeira

Tabela Livro	Tabela Pedidos do Livro
<ul style="list-style-type: none">- Código_livro (chave primária)- Livro- Escritor- Título- Quantidade_paginas- Resumo	<ul style="list-style-type: none">- Numero_pedido- Código_livro (chave estrangeira)- Quantidade

Fonte: elaborada pela autora.



Questão para reflexão

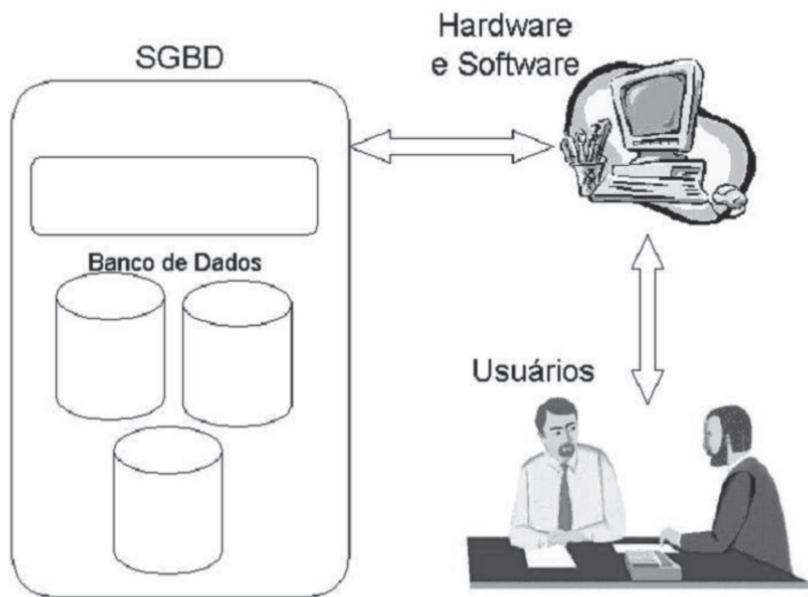
Você consegue imaginar a implicação de não se definir uma chave primária na tabela “Livro”?

1.5 Sistema de Gerenciamento de Banco de Dados (SGBD) e Sistema de Banco de Dados

Um Sistema de Gerenciamento de Banco de Dados (SGBD) pode ser entendido como um conjunto de softwares (serviços) que tem a função de gerenciar um banco de dados, permitindo o acesso aos dados de forma simples (SILBERSCHATZ; KORTH, 2006). Os SGBDs também permitem gerenciar o acesso aos dados para múltiplos usuários, definindo permissões de acesso para cada perfil de usuário. Atualmente, existem vários SGBDs presentes no mercado, os principais são: PostgreSQL, MySQL, Oracle, entre outros.

Por outro lado, um sistema de banco de dados possui maior abrangência, incluindo até mesmo o SGBD. Um sistema de banco de dados possui quatro componentes básicos: hardware, dados, software e usuários. A Figura 1.11 ilustra os componentes básicos de um sistema de banco de dados.

Figura 1.11 | Componentes de um sistema de banco de dados



Fonte: <http://www.devmedia.com.br/imagens/sqlmagazine/mar2006/ORA_RR_01.JPG>. Acesso em: 14 ago. 2017.

Entre os principais objetivos de um sistema de banco de dados, pode-se citar:

- Isolar os usuários dos detalhes internos do banco de dados (abstração de dados);
- Promover a independência dos dados em relação às aplicações (tornar independente da aplicação, a estratégia de acesso e a forma de armazenamento).



Para saber mais

Existem diversos tipos de SGBDs utilizados atualmente. No link a seguir, você encontra um artigo que realiza um estudo comparativo entre os principais tipos de sistemas de gerenciamento de bancos de dados:

<<http://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbds/30788>>. Acesso em: 14 ago. 2017.

Atividades de aprendizagem

1. Existem várias terminologias relacionadas aos bancos de dados e a maneira como sua arquitetura é organizada. Nesse contexto, explique as diferenças entre os conceitos de Banco de Dados, Sistemas de Gerenciamento de Banco de Dados e Sistemas de Banco de Dados.

2. Um dado é um fato bruto, que não agrega em nada ao conhecimento. Já uma informação é o agrupamento dos dados, dando um significado aos mesmos. Imagine um cenário no qual um cliente realiza uma compra em uma loja virtual de livros. Cite alguns dados e informações relacionadas a esse cenário.

Seção 2

Modelagem de um Banco de Dados

Introdução à seção

Os modelos de dados têm o objetivo de explicar como são construídas as estruturas de dados, a organização desses dados e os relacionamentos entre eles. A modelagem dos dados de um BD pode ser realizada conforme três principais tipos de modelos. Um dos modelos mais utilizados para a modelagem de um BD é denominado de modelo relacional, que utiliza conceitos como relações, atributos e tuplas. Um sistema de banco de dados deve proporcionar uma visão abstrata dos dados para os usuários. Dessa maneira, a abstração é representada em três níveis. Devido à sua importância para o desenvolvimento de um BD, iniciamos essa seção discutindo o conceito e a importância da abstração de dados.

2.1 Abstração de dados

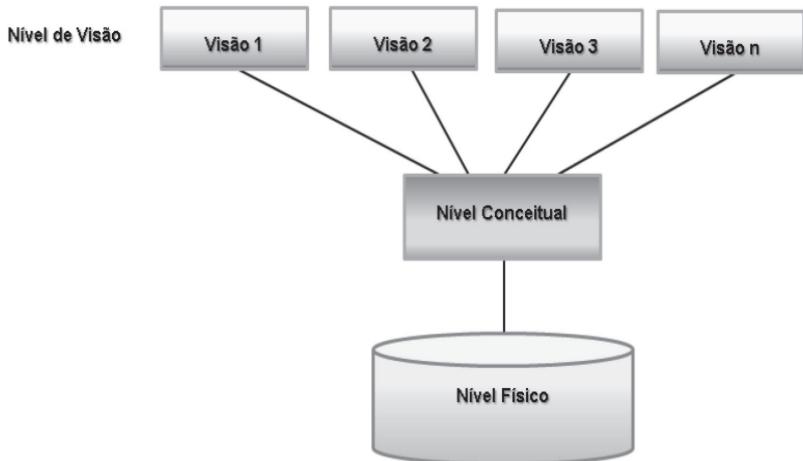
Para os usuários de um BD não é importante ter conhecimento sobre quais unidades de armazenamento os dados estão guardados, desde que esses dados estejam disponíveis quando necessário. Dessa maneira, um sistema de BD deve proporcionar uma visão abstrata do banco de dados para os usuários. A abstração pode ser representada em três níveis: físico, conceitual e visão.

O nível físico possui baixa abstração, descrevendo a maneira como os dados são armazenados. Nesse nível, são descritos detalhadamente as estruturas de dados de baixo nível.

O nível conceitual, também chamado de nível lógico, define quais dados estão armazenados no BD e o tipo de relação entre eles.

O nível de abstração mais alto é chamado de nível de visão do usuário, que descreve apenas partes do banco de dados. Esse nível existe para simplificar a interação do usuário com o sistema. A Figura 1.12 ilustra os três níveis de abstração.

Figura 1.12 | Níveis de abstração



Fonte: elaborada pela autora.

2.2 Modelos de Banco De Dados

Os modelos de banco de dados apresentam os tipos de informações armazenadas em um banco de dados. Os modelos não informam quais são os alunos armazenados, apenas o tipo de informações que eles possuem. Os modelos podem informar, por exemplo, que o BD guarda informações sobre "alunos" e que para cada "aluno" são armazenadas informações, como: matrícula, nome, curso e coeficiente de rendimento.

Para a construção de um modelo é utilizada uma linguagem para a modelagem de dados, que pode ser textual ou gráfica. Os modelos podem ser descritos em diferentes níveis de abstração, sendo que cada nível possui um objetivo específico.

A modelagem de dados é o processo que engloba o entendimento, especificação e validação do modelo de dados proposto, seguindo as diretrizes e padrões. A modelagem de um BD pode ser realizada de acordo com três aspectos:

- Modelagem Conceitual: possui alto nível de representação e leva em consideração a perspectiva do usuário que criou os dados. Este modelo demonstra todas as entidades e relações entre as mesmas, assim como seus atributos.

- Modelagem Lógica: acrescenta detalhes de implementação do banco de dados. Este modelo mostra as ligações entre as tabelas de um BD e detalhes como: chaves primárias, chaves estrangeiras etc.
- Modelagem Física: mostra a maneira de armazenamento físico dos dados. Este modelo analisa os recursos e características necessárias para o armazenamento e manipulação dos dados. Nessa modelagem, comandos em SQL (*Structured Query Language* ou Linguagem de Consulta Estruturada) são executados a fim de se criar tabelas e relacionamentos.



Para saber mais

A linguagem SQL é muito utilizada para a definição e manipulação de dados de um modelo relacional. Com essa linguagem é possível realizar operações de manipulação de dados de um BD, como: criação de tabelas, seleção de atributos, alteração de dados de uma tabela etc. No link a seguir, você pode aprofundar seus estudos sobre SQL, conhecendo a sintaxe das principais operações dessa linguagem: <<http://adrianoribeiro.orgfree.com/downloads/SQL.pdf>>. Acesso em: 18 ago. 2017.

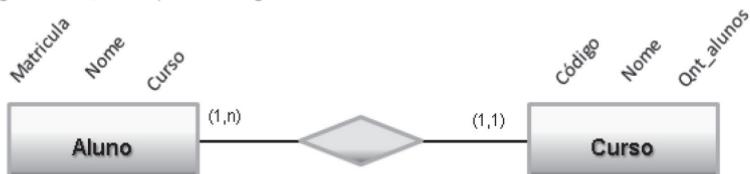
2.1.1 Modelo Conceitual

Este modelo representa uma descrição de um BD de maneira que não depende do sistema de gerenciamento de banco de dados a ser utilizado. Dessa maneira, o modelo conceitual tem a função de definir os dados presentes em um BD, não se importando com a implementação, ou seja, existe uma abstração em nível de SGBD. O modelo conceitual possui uma representação com alto nível de abstração, modelando fatos do mundo real, assim como seus relacionamentos, de maneira natural.

Uma das práticas mais utilizadas no modelo conceitual é o enfoque entidade-relacionamento (ER), que tem seu modelo graficamente representado pelo diagrama entidade-relacionamento (DER). Basicamente, o DER identifica todas as entidades e relacionamentos de uma maneira global. Na Figura 1.13 é apresentado um modelo de DER, que possui informações sobre alunos e cursos. Para cada "Aluno"

são armazenadas as informações sobre: número de matrícula, nome e curso. Cada "Curso" possui um código, nome e quantidade de alunos.

Figura 1.13 | Exemplo de diagrama ER



Fonte: elaborada pela autora.

2.1.2 Modelo Lógico

Este modelo tem por finalidade descrever o banco de dados no nível do sistema de gerenciamento de banco de dados utilizado. Este modelo realiza a representação dos dados em uma estrutura lógica de armazenamento. Após a seleção do SGBD utilizado, o modelo conceitual é mapeado para um modelo lógico. Dessa forma, o modelo lógico depende do SGBD. Assim, qualquer alteração realizada no sistema de gerenciamento de banco de dados exige que o modelo seja alterado para se adequar às características de implementação desse SGBD. Mesmo sendo dependente do software (SGBD), este modelo não depende do hardware, ou seja, a escolha de um novo computador ou sistema operacional não afetará o modelo lógico.

O modelo lógico mais conhecido e utilizado em um SGBD relacional é o modelo relacional, que organiza os dados em forma de tabelas ou relações. Porém, existem outros tipos de modelos lógicos, como: modelo hierárquico e modelo orientado a objetos.

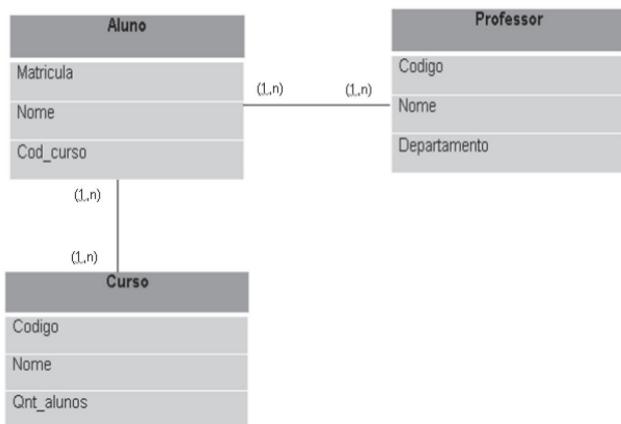
Para o exemplo anteriormente utilizado no DER, o modelo lógico pode ser definido, de forma textual, como:

Aluno(Matricula, Nome, Curso)

Curso(Código, Nome, Qnt_alunos)

No modelo lógico, não são descritos os detalhes internos de armazenamento dos dados, já que estas informações fazem parte do modelo físico. O modelo lógico também pode ser definido graficamente, conforme ilustrado na Figura 1.14.

Figura 1.14 | Exemplo de modelo lógico



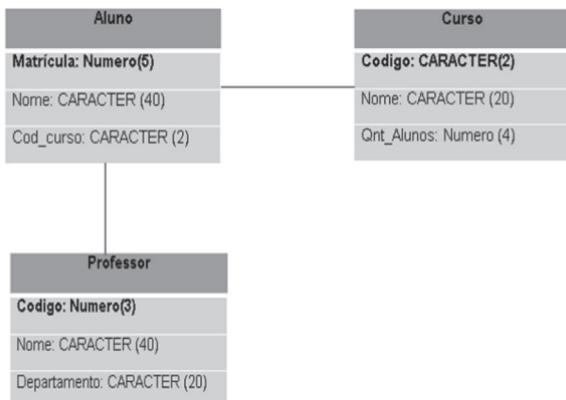
Fonte: elaborada pela autora.

2.1.3 Modelo Físico

O modelo físico representa uma descrição de um BD no nível de abstração do usuário do SGBD. Dessa maneira, este modelo depende do sistema de gerenciamento de banco de dados utilizado para a implementação do BD.

Neste modelo, são delineados os componentes da estrutura física do BD, como: campos, tabelas, tipos de dados, índices etc. A Figura 1.15 ilustra um exemplo de modelo físico de um banco de dados.

Figura 1.15 | Exemplo de modelo físico



Fonte: elaborada pela autora.

Um modelo físico também pode ser representado de forma textual, utilizando a linguagem SQL para a definição de comandos, conforme observado na Figura 1.16.

Figura 1.16 | Exemplo de criação de tabela utilizando SQL

```
CREATE TABLE Aluno (
    MATRICULA Integer(5) PRIMARY KEY NOT NULL,
    NOME Text(40),
    COD_CURSO Text(2),
)
```

Fonte: elaborada pela autora.



Questão para reflexão

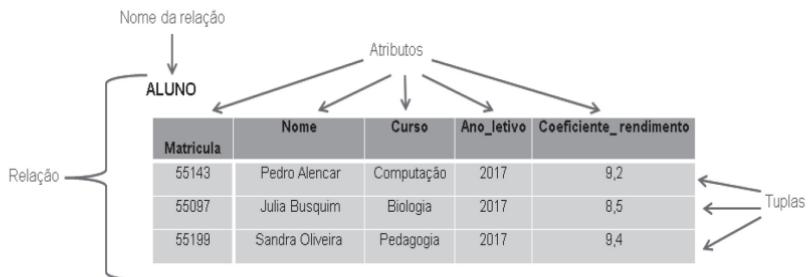
Você consegue explicar a importância da modelagem de dados e citar as vantagens associadas à utilização da mesma para o desenvolvimento de um BD?

2.2 Modelo Relacional

O modelo relacional basicamente representa os dados de um banco de dados como uma coleção de tabelas, também chamadas de relações (SILBERSCHATZ; KORTH, 2006). Cada relação possui um nome exclusivo e um conjunto de atributos que possuem nomes e domínios. Como já estudado, os valores de uma coluna são sempre de um mesmo tipo de dados.

Algumas terminologias são empregadas nos modelos relacionais. Vimos que os elementos básicos de um BD são: campo, registro e tabela. Para o modelo relacional, um registro ou linha é denominado de tupla. Um campo (nome de uma coluna) é chamado de atributo. Já uma tabela é definida como uma relação. As terminologias mais importantes do modelo relacional estão ilustradas na Figura 1.17.

Figura 1.17 | Terminologias do modelo relacional



Fonte: elaborada pela autora.

Uma relação pode ser definida como um conjunto não ordenado de tuplas. Um atributo é um nome que identifica uma característica ou propriedade de uma relação. O domínio é um conjunto de valores que caracterizam um atributo. Uma tupla é uma sequência ordenada de valores. Todas as tuplas de uma relação são diferentes, pois representam entidades ou relacionamentos específicos da base de dados. Suponha uma relação chamada "Aluno" que possui como um de seus atributos "Matrícula", conforme ilustrado na Figura 1.18.

Figura 1.18 | Exemplo de domínio

Matrícula	Nome	Curso	Carga_horaria
55143	Pedro Alencar	Computação	16
55097	Julia Busquim	Biologia	20
55199	Sandra Oliveira	Pedagogia	20
55021	Gustavo Ferreira	Biologia	12
55012	Daniel Silva	Computação	16

Fonte: elaborada pela autora.

Para o atributo "Matrícula" é permitido um conjunto de valores de cinco dígitos. A esse conjunto de valores permitidos para um atributo dá-se o nome de domínio. Demaneira análoga, o domínio do atributo "Curso" é o conjunto de cursos de uma universidade. Cada domínio está associado a um tipo ou formato de dados. Suponha um atributo chamado "telefone", o domínio para este atributo poderia ser:

(ddd) dddd-dddd, onde d = {0,1,2,...,9}.

Um esquema pode ser definido como o conjunto de atributos de uma relação. Para a relação "Aluno", o esquema é: {Matricula, Nome, Curso, Ano_letivo, Coeficiente_rendimento}.

O grau de uma relação refere-se ao número de atributos que seu esquema possui. Por exemplo, o esquema "Aluno" possui cinco atributos: "Matricula", "Nome", "Curso", "Ano_letivo" e "Coeficiente_rendimento", então o grau dessa relação é igual a cinco.

As relações possuem algumas características:

- A ordem das tuplas não é importante;
- Todo o atributo deve possuir um valor atômico;
- Todas as tuplas devem ser únicas;
- Cada atributo possui um nome único dentro da relação.

Nós já estudamos na Unidade 1 os principais conceitos relacionados às chaves de um banco de dados. Para o modelo relacional, usamos as mesmas definições já vistas, apenas agregando algumas convenções.

É convencionado sublinhar os atributos que representam as chaves primárias de uma relação. Por exemplo, na relação "Aluno" a chave primária é o número de matrícula, pois este atributo é único para cada aluno de uma universidade. Então, temos:

Aluno (Matricula, Nome, Curso, Ano_letivo, Coeficiente_rendimento)

2.2.1 Restrições de Integridade

As restrições de integridade reduzem a redundância das informações armazenadas em um banco de dados e diminuem o espaço total de armazenamento.

No modelo relacional são definidas algumas restrições de integridade, principalmente relacionadas às chaves, entidades, atributos e relações.

A integridade de uma chave está relacionada à maneira como cada tupla é identificada unicamente por um conjunto de atributos dentro de uma relação, ou seja, em uma relação não podem existir duas tuplas com valores iguais na chave primária.

Na integridade de atributos, os valores de um atributo devem pertencer ao domínio do mesmo.

Na integridade de uma entidade, os valores das chaves primárias

não podem ser nulos.

Na integridade Referencial, uma relação pode ter um conjunto de atributos que possui valores com o mesmo domínio de um conjunto de atributos que forma a chave primária de outra relação. Como já estudado, esse conjunto é chamado chave estrangeira. Assim, uma tupla que referencia outra relação deve referenciar um tupla existente nesta outra relação.

2.2.2 Operações em Relações

Uma base de dados deve sempre satisfazer as restrições de integridade. Existem alguns tipos de operações para a manipulação das relações, principalmente inserção e remoção. Deve ser observado sempre que essas operações não violem as regras de integridade dos dados.

2.2.3 Inserção

A operação de inserção permite inserir novas tuplas em uma relação. A inserção pode violar as restrições de integridade nos seguintes casos:

- **Integridade de Domínio:** quando um dos valores não pertence ao domínio do atributo respectivo.
- **Integridade de Chave:** quando o valor da chave já existe em outra tupla da relação.
- **Integridade de Entidade:** quando o valor da chave é nulo.
- **Integridade Referencial:** quando uma chave estrangeira referencia uma tupla não existente na relação referenciada.

Quando uma operação de inserção viola uma dessas restrições, o Sistema de Gerenciamento de Banco de Dados pode:

- Rejeitar a inserção e, se possível, indicar o tipo de restrição violada;
- Tentar corrigir as razões pelas quais ocorreu uma violação das restrições de integridade.

Considere a Figura 1.19, que possui duas tabelas. A tabela “Aluno” tem como chave primária o atributo “Matricula” e a tabela “Curso” possui como chave primária o atributo “Codigo”. Percebe-se que o atributo “Cod_curso” é chave primária da tabela “Curso” e chave estrangeira da tabela “Aluno”.

Figura 1.19 | Exemplo de inserção

Aluno				
Matricula	Nome	Cod_curso	Ano_letivo	Coeficiente_rendimento
55143	Pedro Alencar	C1	2017	9,2
55097	Julia Busquim	C2	2017	8,5
55199	Sandra Oliveira	C3	2017	9,4
55021	Gustavo Ferreira	C2	2017	8,9
55012	Daniel Silva	C1	2017	7,5

Curso		
Código	Nome	Qnt_alunos
C1	Computação	287
C2	Biologia	179
C3	Pedagogia	200

Fonte: elaborada pela autora.

No caso da inserção de novas tuplas na relação “Aluno”, pode ocorrer algumas situações, por exemplo:

Inserção da tupla: <'55010','Camila Vieira','C1', '2017', '9,0'>

- Essa inserção é aceita e não viola nenhuma regra de integridade.

Inserção da tupla: <'55012','Camila Vieira','C1', '2017', '9,0'>

- Essa inserção viola a restrição de chave, pois já existe outra tupla com esse mesmo valor (tupla com o valor “Daniel” para o atributo “Nome”).

Inserção da tupla: <NULL,'Camila Vieira','C1', '2017', '9,0'>

- Nesse caso, viola restrição de integridade de entidade, pois a chave primária de uma relação nunca deve ser nula.

Inserção da tupla: <'55010','Camila Vieira','Z5', '2017', '9,0'>

- Nessa inserção, é violada a restrição de integridade referencial, pois a chave estrangeira “Z5” que referencia a tupla não existente na relação “Curso”.

2.2.4 Remoção

A operação de remoção permite remover (excluir) tuplas de uma

relação. Nesta operação, é necessário indicar uma condição sobre os atributos que serão removidos.

No caso da remoção de tuplas na relação "Aluno" e "Curso", podem ocorrer algumas situações, por exemplo:

Remover a tupla com matrícula = '55143'.

- Essa remoção é aceita e não viola nenhuma regra de integridade.

Remover da relação "Curso" a tupla com Código = 'C1'.

- Essa remoção não é aceita, pois viola a regra de integridade referencial, já que existem alunos que estão alocados neste curso.

Se ocorrer uma violação em uma remoção, algumas ações são necessárias, por exemplo:

- Rejeitar a remoção;
- Remover todas as tuplas referenciadas pela tupla que está sendo removida;
- Modificar os atributos referenciados para novos valores ou valores nulos (se não fizerem parte da chave primária).

2.2.5 Modificação

Essa operação permite alterar as tuplas de uma relação. No caso da modificação de tuplas nas relações "Aluno" e "Curso", podem ocorrer algumas situações, como:

Modificar o coeficiente de rendimento do aluno com matrícula = '55143':

- Nesse caso, a operação de modificação é aceita sem problemas, pois não viola nenhuma regra de integridade.

Modificar o código do curso do aluno com matrícula '55021' para 'C3'.

- Essa operação de modificação é aceita, não violando nenhuma regra de integridade.

Modificar o código do curso do aluno com matrícula '55199' para 'C5':

- Operação de modificação não aceita, pois viola a integridade referencial.

Modificar a matrícula do aluno '55097' para '55199':

- Modificação não aceita, porque viola a regra de integridade de chave.

2.3 Álgebra Relacional

A álgebra relacional é uma linguagem formal e procedural de banco de dados. A linguagem de consulta formal é aquela em que o usuário solicita informações à base de dados. Já a linguagem procedural é aquela em que o usuário fornece as instruções ao sistema para que o mesmo realize uma sequência de operações na base de dados para calcular o resultado desejado.

A Álgebra Relacional define operadores para atuar nas relações. Essa linguagem utiliza uma ou mais relações como entrada de dados para produzir uma nova relação como resultado das operações.

Para exemplificar as operações da álgebra relacional, a tabela apresentada na Figura 1.20 será utilizada. Os códigos C1, C2 e C3 correspondem, respectivamente, aos cursos de: Computação, Biologia e Pedagogia.

Figura 1.20 | Exemplo de relação

Matricula	Nome	Cod_curso	Ano_letivo	Coeficiente_rendimento
55143	Pedro Alencar	C1	2017	9,2
55097	Julia Busquim	C2	2017	8,5
55199	Sandra Oliveira	C3	2017	9,4
55021	Gustavo Ferreira	C2	2017	8,9
55012	Daniel Silva	C1	2017	7,5

Fonte: elaborada pela autora.

A seguir, são apresentadas as principais operações da álgebra relacional.

2.3.1 Operação de Seleção (σ)

A operação de seleção escolhe um subconjunto de tuplas de uma relação, de acordo com uma condição. A sintaxe utilização para a seleção é composta pelo operador e predicado:

$\sigma <\text{predicado}> (\text{Relação})$

Onde, σ é o operador de seleção

No predicado, pode-se utilizar operadores relacionais ($=, <, >, \geq, \leq, \neq$) e operadores booleanos (AND, OR, NOT).

Como exemplo, considere o esquema relacional de uma universidade hipotética, com os seguintes dados:

Aluno (Matricula, Nome, Cod_curso, Ano_letivo, Coeficiente_rendimento)

Curso(Codigo, Nome, Qnt_alunos)

Para exemplificar a operação de seleção, considere a situação de selecionar os alunos que estão matriculados no curso de Computação. Na álgebra relacional, essa operação de seleção é expressa da seguinte maneira:

σ Curso = "Computação" (Aluno)

Essa expressão tem como resultado um conjunto dos elementos da relação "Aluno" que atendem ao predicado [Curso = "Computação"], representando um subconjunto dos alunos para o qual a condição é avaliada como verdadeira. A relação resultante dessa operação é ilustrada na Figura 1.21.

Figura 1.21 | Exemplo 1 de seleção

Matricula	Nome	Cod_curso	Ano_letivo	Coeficiente rendimento
55143	Pedro Alencar	C1	2017	9,2
55012	Daniel Silva	C1	2017	7,5

Fonte: elaborada pela autora.

Para selecionar os alunos que possuem coeficiente de rendimento maior que 9,0, pode-se usar a seguinte expressão:

σ Coeficiente_ren dim entro > 9,0 (Aluno)

A relação resultante dessa operação é ilustrada na Figura 1.22.

Figura 1.22 | Exemplo 2 de seleção

Matricula	Nome	Cod_curso	Ano_letivo	Coeficiente rendimento
55143	Pedro Alencar	C1	2017	9,2
55199	Sandra Oliveira	C3	2017	9,4

Fonte: elaborada pela autora.

O operador de seleção é unário, ou seja, apenas aplicado a uma única relação. O número de tuplas da relação resultante é menor ou igual ao número de tuplas da relação original.

2.3.2 Projeção (π)

A operação de projeção é responsável por selecionar um subconjunto de atributos de uma relação. Se a lista de atributos contiver apenas atributos que não são chaves, pode aparecer nos resultados algumas tuplas duplicadas. Entretanto, a operação de projeção elimina essa duplicação. A sintaxe dessa operação é a seguinte:

$\pi < \text{listadeatributos} > (\text{Aluno})$

Como exemplo, considere a expressão: $\pi \text{ Nome } (\text{Aluno})$

Nesse caso, a operação de projeção filtra apenas a coluna da relação que possui como atributo "Nome". A Figura 1.23 ilustra a relação resultante da expressão.

Figura 1.23 | Exemplo de projeção

Nome
Pedro Alencar
Sandra Oliveira
Gustavo Ferreira
Daniel Silva

Fonte: elaborada pela autora.

2.3.3 União

A operação de união realiza a união das tuplas de duas relações, eliminando automaticamente a duplicação de tuplas. Na união, as relações devem possuir o mesmo número de atributos. A sintaxe utilizada nessa operação é a seguinte:

Relação 1 \cup Relação 2

Considere as duas relações, "Aluno" e "Professor", ilustradas na Figura 1.24.

Figura 1.24 | Exemplo de união

ALUNO

Matricula	Nome	Curso	Carga_horaria
55143	Pedro Alencar	C1	16
55097	Julia Busquim	C2	20
55199	Sandra Oliveira	C3	20
55021	Gustavo Ferreira	C2	12
55012	Daniel Silva	C1	16

PROFESSOR

Matricula	Nome	Curso	Carga_horaria
123	Robson Andrade	Química	40
124	Amanda Silva	Matemática	20
125	Sergio Moraes	Letras	20
126	Regina Dias	Matemática	40
127	Paulo Souza	Física	20

Fonte: elaborada pela autora.

Como resultado da união, a Figura 1.25 mostra a relação resultante entre a união de todos os alunos com todos os professores.

Figura 1.25 | Relação resultante da união

Matricula	Nome	Curso	Carga_horaria
55143	Pedro Alencar	C1	16
55097	Julia Busquim	C2	20
55199	Sandra Oliveira	C3	20
55021	Gustavo Ferreira	C2	12
55012	Daniel Silva	C1	16
123	Robson Andrade	Química	40
124	Amanda Silva	Matemática	20
125	Sergio Moraes	Letras	20
126	Regina Dias	Matemática	40
127	Paulo Souza	Física	20

Fonte: elaborada pela autora.

2.3.4 Interseção

A intersecção fornece como resultado as tuplas comuns em duas relações. A sintaxe dessa operação é a seguinte:

Relação 1 \cap Relação 2

Considere as duas relações da Figura 1.26. Se a operação de intersecção for utilizada nessas relações, apenas os alunos que também são professores farão parte da relação resultante.

Figura 1.26 | Exemplo de intersecção

ALUNO

Matricula	Nome	Curso
55143	Pedro Alencar	Computação
55097	Julia Busquim	Biologia
55199	Sandra Oliveira	Pedagogia
55021	Gustavo Ferreira	Biologia
55012	Daniel Silva	Computação

PROFESSOR

Matricula	Nome	Curso
123	Robson Andrade	Química
124	Pedro Alencar	Computação
125	Sergio Moraes	Letras
126	Regina Dias	Matemática
127	Paulo Souza	Física

Fonte: elaborada pela autora.

Como resultante, tem-se a relação apresentada na Figura 1.27.

Figura 1.27 | Relação resultante da intersecção

Matricula	Nome	Curso
55143	Pedro Alencar	Computação

Fonte: elaborada pela autora.



Para saber mais

Existem outras operações da álgebra relacional, como: diferença e produto cartesiano. No link a seguir você pode conferir com detalhes todas as operações da álgebra relacional:

<<http://wiki.icmc.usp.br/images/2/2c/SCC578920131-algebraSQL.pdf>>. Acesso em: 14 ago. 2017.

Atividades de aprendizagem

1. O modelo conceitual não se preocupa com detalhes de implementação dos dados em um BD, apenas descreve as principais informações que o mesmo possui. Imagine que você seja o responsável por desenvolver um banco de dados para uma biblioteca. A primeira etapa do desenvolvimento é a criação do modelo conceitual, que é principalmente representada pelo diagrama ER. Assim, desenhe um DER para o sistema de biblioteca, focando apenas no caso de uso “aluno empresta um livro”.

2. No Modelo Relacional as tabelas são consideradas relações e cada registro é visto como uma tupla. Imagine que você é o responsável pela construção de um modelo relacional para uma loja virtual que vende livros de várias editoras. Desenvolva o modelo relacional para essa loja, considerando as principais relações (tabelas) envolvidas no processo de compra de um livro.

Fique ligado

Nesta unidade, você começou seu estudo aprendendo sobre os principais conceitos relacionados aos Bancos de Dados, assim como as terminologias mais utilizadas. Foram apresentadas as definições de tabela, campo e registro. Você aprendeu que um banco de dados é um conjunto de dados inter-relacionados, ao passo que um sistema de gerenciamento de banco de dados (SGBD) é um programa que fornece uma interface ao usuário para a manipulação dos dados, assim como gerencia o acesso aos dados. Por último, um sistema de banco de dados engloba os dados, os usuários e o SGBD. Esta unidade também demonstrou a importância da modelagem de dados, e os principais tipos utilizados, especialmente o modelo relacional. Foi explicada a importância da integridade dos dados de um BD e as operações de manipulação dos dados, assim como as mesmas podem violar as restrições de integridade. Para finalizar esta unidade foi apresentado a você as principais operações da álgebra relacional.

Para concluir o estudo da unidade

Na primeira seção, você aprendeu os conceitos básicos relacionados aos bancos de dados, a diferença entre dados e informações. Também foram apresentados os elementos básicos de um banco de dados.

Os bancos de dados representam uma parte muito importante de qualquer sistema. Com um BD bem estruturado as consultas e operações sobre os dados possuem maior rapidez e eficiência. Além disso, dados relacionados podem se transformar em informações valiosas para qualquer empresa ou organização. Atualmente, existem algumas técnicas inteligentes que são empregadas nos bancos de dados, como o *DataMining*. Essa técnica realiza a mineração de dados à procura de padrões entre os mesmos, sendo muito utilizada pelas maiores empresas do mundo na intenção de aumentarem suas vendas e produtividade. Como você pode perceber, informações corretas e

bem estruturadas podem favorecer qualquer organização, seja ela com ou sem fins lucrativos. Assim, o estudo sobre banco de dados é fundamental para os profissionais da área de Computação.

Atividades de aprendizagem da unidade

1. O modelo relacional basicamente representa os dados de um banco de dados como uma coleção de tabelas, também chamadas de relações. Considere o seguinte modelo relacional de uma “Empresa”:

Empregado (PrimeiroNome, InicialMeio, UltimoNome, NumEmpregado, DataNascimento, Endereco, Sexo, Salario, NumSupervisor, NumDept)

Departamento (NomeDept, NumDept, NumGerente, DataInicioGerencia)

Localizacao_Depto (NumDepart, Localizacao)

Projeto (NomeProj, NumProj, Localizacao, NumDept)

Trabalha_em (NumEmpregado, NumProj, Horas) Trabalha_em
(NumEmpregado, NumProj, Horas)

Dependente (NumEmpregado, NomeDependent)

Utilizando os operadores da álgebra relacional, recuperar o nome (primeiro e último nome) e o endereço dos empregados que trabalham para o departamento de número 5. Assinale a alternativa que contém a expressão relacional correta.

- a) π PrimeiroNome, UltimoNome, Endereco (σ NumDept=5 (Empregado))
- b) π PrimeiroNome, UltimoNome, Endereco (Empregado)
- c) π Empregado (σ NumDept=5)
- d) π Empregado=5
- e) σ NumDept=5 (Empregado)

2. O modelo lógico, quando se utiliza de um SGBD relacional, pode ser representando pelo modelo relacional. Esse modelo pode ser representado de forma textual. Assim, faça o modelo relacional das relações “Produto” e “Tipo_produto”, escolhendo alguns atributos para as mesmas. Assinale a alternativa correta.

- a) Item (Produto, Tipo_produto) Produto (Item)
- b) Produtos (Produto, Tipo_produto)
- c) Tipo_produto (Produto)
- d) Produto (Tipo_produto)
- e) Tipo_produto (CodTipoProd, DescrTipoProd) Produto (CodProd, DescrProd, PrecoProd, CodTipoProd)

3. A álgebra relacional é uma linguagem formal e procedural de banco de dados. Suponha a tabela “Docente”, conforme especificada a seguir:

Docente (Codigo, CPF, Nome, Endereço, Departamento)

Utilizando a álgebra relacional, liste todos os docentes que trabalham no departamento de Física. Assinale a alternativa correta.

- a) σ Departamento = "Docente = Física"
- b) σ Docente (Departamento)
- c) σ Departamento = "Física" (Docente)
- d) σ Departamento = "Física"
- e) σ Física = (Docente)

4. O modelo conceitual não se preocupa com detalhes de implementação dos dados em um BD, apenas descreve as principais informações que o mesmo possui. Imagine que você seja o responsável por desenvolver um banco de dados para um restaurante. A primeira etapa do desenvolvimento é a criação do modelo conceitual, que é principalmente representada pelo diagrama ER. Assim, assinale a alternativa que possui o DER para o sistema de restaurante, focando apenas no caso de uso “cliente realiza um pedido”.

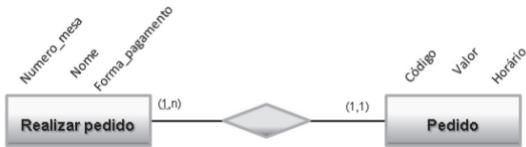
a)



b)



c)



d)



e)



5. A modelagem de dados é fundamental para o desenvolvimento de um banco de dados, sendo os três principais tipos de modelagem: conceitual, lógica e física. Assinale a alternativa que contém a correta definição dos principais tipos de modelagem.

- a) A modelagem conceitual possui baixo nível de representação e leva em consideração armazenamento físico dos dados de um banco de dados.
- b) A modelagem lógica leva em consideração a perspectiva do usuário criador dos dados, demonstrando todas as entidades e relações entre as mesmas, assim como seus atributos.
- c) A modelagem lógica se preocupa com armazenamento físico dos dados de um banco de dados, gerenciando os usuários e criando perfis para restringir o acesso aos dados.
- d) A modelagem física demonstra a maneira de armazenamento físico dos dados e analisa os recursos e características necessárias para o armazenamento e manipulação dos dados.
- e) A modelagem física possui alto nível de abstração dos dados, demonstrando todas as entidades e relações entre as mesmas, sendo principalmente representada pelo diagrama E-R.

Referências

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 4. ed. Rio de Janeiro: Editora Campus, 1991.

SILBERSCHATZ, A.; KORTH, H. Sudarshan. **Sistema de Banco de Dados**. 5. ed. Editora Makron Books, 2006.

Projeto de banco de dados

Nathalia dos Santos Silva

Objetivos de aprendizagem

Nesta unidade, você será apresentado aos conceitos que envolvem o Modelo Entidade Relacionamento, tais como as entidades, os atributos, as condições de cardinalidade e os tipos envolvendo relacionamentos.

Assim, a partir do conhecimento das estruturas que compõem o Diagrama Entidade Relacionamento você terá condições de aplicá-las de maneira consistente durante a fase de projeto do banco de dados.

Introdução à unidade

O Modelo de Dados Relacional é amplamente utilizado devido à sua simplicidade e base matemática, que envolve teoria de conjuntos e lógica de predicados (ELMASRI e NAVATHE, 2011), e este conceito você já estudou na Unidade 1. Ao mesmo tempo, você deve concordar comigo, é necessário um modelo mais dinâmico e visual para descrever de forma mais intuitiva as relações reais entre os envolvidos.

O Modelo Entidade Relacionamento (MER) consegue traduzir de forma mais intuitiva e dinâmica o Modelo de Dados Relacional através de sua representação gráfica, o Diagrama Entidade Relacionamento (DER). Além de apresentar a importância do DER esta unidade traz conceitos e exercícios relacionados à implementação do banco de dados em sua fase de projeto.

A proposta desta Modelagem de Dados por meio do DER é de Peter Chen, desde 1976, e a partir de então é considerada uma poderosa ferramenta de Engenharia de Software para o Projeto de Bancos de Dados. E a principal consolidação desta técnica na área é o fato de ela não estar atrelada à alguma ferramenta tecnológica, pelo contrário, ela é relacionada à visão da informação presente no conhecimento de uma pessoa (MACHADO, 2014). Além disso, a notação do DER é muito semelhante aos diagramas de classes da Unified Modeling Language (UML), o que reforça a sua usabilidade.

O Projeto de Banco de Dados possui as mesmas etapas de projeto de um Software Tradicional, ou seja, partimos de um modelo de mais alto nível, com grande abstração e detalhamento de funcionalidades e procuramos derivá-lo sucessivamente até obter um modelo de baixo nível, com detalhes de implementação e poucos aspectos do contexto do negócio(ELMASRI e NAVATHE, 2011).

Um dos problemas que temos provenientes de um projeto mal elaborado é a redundância (SILBERSCHATZ, KORTH e SUDARSHAN, 2012). Sabemos que agora sabemos que agora você deve estar se perguntando porque isso seria um problema, já que parece interessante possuir redundância nos dados, além do mais, em outras áreas da Computação a redundância é sinal de robustez.

Você já estudou o Modelo Relacional, então lembre que as tabelas

podem reunir informações de outras tabelas. O problema surge da necessidade de se alterar os dados de um dos atributos e o cuidado em garantir a alteração em todas as referências, ou teríamos cópias inconsistentes (SILBERSCHATZ, KORTH e SUDARSHAN, 2012).

Pelo exposto, não é indicado que seja armazenado nestas tabelas os dados em duplicidade e sem o seu devido mapeamento, e mais ainda, não é necessário. Por exemplo, podemos ter um banco de dados que tenha a entidade produto, com atributos como código do produto e nome. E estes atributos poderiam ser referenciados em diversas outras tabelas, mas ao realizar uma alteração no nome, todas as tabelas que contêm essa informação precisariam ser atualizadas. Considera-se então que não é necessário, e nem adequado, construir tabelas com esta característica, já que poderíamos recuperar o nome do produto pelo seu código. Estes aspectos devem ser observados com precaução na elaboração do projeto, especificamente na fase que antecede a implementação.

É indiscutível a importância de boas práticas de desenvolvimento de software para que obtenhamos um projeto coerente e correto, e nesse aspecto o esquema visual do DER consegue traduzir as necessidades do mundo real para a implementação do banco de dados (CARDOSO e CARDOSO, 2012), por isso essa seção discute cada conceito envolvido nesta etapa de projeto.

O capítulo está definido da seguinte maneira:

Na Seção 1, veremos como as entidades representam as instituições que queremos modelar, e como definimos quem serão as entidades e os atributos no projeto de banco de dados.

Na Seção 2, discutiremos sobre os relacionamentos que definimos entre as entidades, e quais as características que devem estar presentes no DER.

Na Seção 3, aprenderemos como a Normalização deixa o nosso projeto de Banco de Dados mais consistente e como é realizado o processo de adequação às Formas Normais.

Seção 1 | Entidades e atributos

A Seção 1 trata das entidades e suas características, porque é através da definição das mesmas que o projeto de banco de dados é iniciado. Nesta seção, também são descritos os atributos das entidades e suas variações quanto às restrições de valores que eles podem assumir.

Seção 2 | Relacionamentos

A Seção 2 apresenta os tipos de relacionamentos, ou seja, se são unários, binários ou terciários, e ainda, a cardinalidade dos relacionamentos. Esta última garantirá algumas exigências de implementação no banco de dados oriundas das relações reais.

Seção 3 | Normalização

A Seção 3 explica o conceito de normalização, apresenta as formas normais, as anomalias que queremos evitar e discute procedimentos para deixar o projeto adequado conforme cada uma das regras de normalização.

Seção 1

Entidades e atributos

Introdução à seção

As tabelas são as estruturas características e explicativas no modelo relacional, já no modelo entidade-relacionamento são as entidades que representam a descrição da situação de interesse, em conjunto com o que chamamos de relacionamento, que basicamente é a associatividade entre as entidades. Devido à sua importância para o processo de desenvolvimento do banco de dados, iniciamos a nossa unidade discutindo como trabalhamos as entidades para que as mesmas tenham significado no nosso projeto.

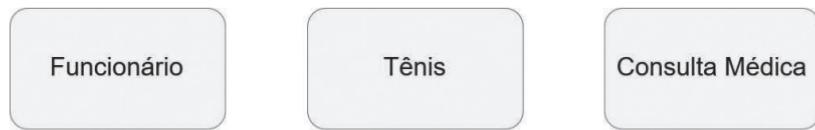
O processo de definição das entidades segue exemplos intuitivos, que fazem parte do nosso contexto, e em seguida aplicaremos a mesma técnica nos dados das tabelas que criamos com base no modelo relacional.

1.1 Entidades

Em geral, as **entidades** são objetos, pessoas ou acontecimentos que possuem significado no contexto a ser modelado, e que se distinguem de outros objetos (SILBERSCHATZ, KORTH e SUDARSHAN, 2012). As entidades representam um conjunto de dados com aquelas características que definimos para classificá-las, porém, as entidades não são o dado em si, a este último damos o nome de **ocorrência**. Podemos pensar também em Classe e Instância, respectivamente, pela similaridade à Entidade e Ocorrência, para os mais familiarizados com a notação de Orientação a Objetos.

Um exemplo de entidade pode ser Funcionário, Tênis ou Consulta Médica, sendo que estas definições permitem que posteriormente existam os dados dos Funcionários, dos Tênis, e das Consultas Médicas. Quanto à sua representação no DER, a entidade é um retângulo com seu nome inscrito, como pode ser observado na Figura 2.1:

Figura 2.1 | Entidades



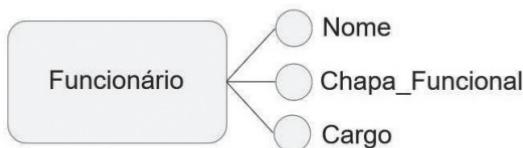
Fonte: elaborada pelo autor.

Observe que as entidades podem ser concretas ou abstratas, depende da finalidade no nosso sistema de banco de dados, em geral elas são descritas em uma ou duas palavras que as define, como curso, departamento, pessoa, aluno, filme, pedido, ingrediente, depósito, casa, locador, clínica ou máquina, tudo depende da aplicação e do foco do nosso sistema.

1.2 Atributos

Os **atributos** organizam as informações de interesse das entidades, ou seja, as características de significado dos dados, bem como as propriedades daquela entidade. Cada entidade tem seu próprio **valor** para cada um de seus atributos, que pode ter uma faixa de valores ou de domínio, e preferencialmente deve ser simples (MEDEIROS, 2013), o suficiente para ser autoexplicativo. Mantendo nosso exemplo Funcionário, seus atributos podem ser: Nome, Chapa Funcional, Data de Nascimento, Cargo, entre outros, como pode ser visto na Figura 2.2:

Figura 2.2 | Atributos



Fonte: elaborada pelo autor.

Na necessidade de se identificar e diferenciar, de maneira única, uma entidade existe o **atributo-chave ou identificador** (MEDEIROS, 2013), como o próprio nome representa. Podemos entender o atributo identificador como aquele que leva a chave primária identificada ao Modelo Relacional. Esse atributo é identificado de maneira diferenciada, como ilustrado na Figura 2.3:

Figura 2.3 | Atributo Identificador



Fonte: elaborada pelo autor.

Precisamos definir as especificações ou restrições, esperadas e possíveis, para o valor que pode assumir cada atributo, pois será a partir da descrição formal destas restrições que traduziremos na implementação do banco de dados as particularidades reais do projeto.

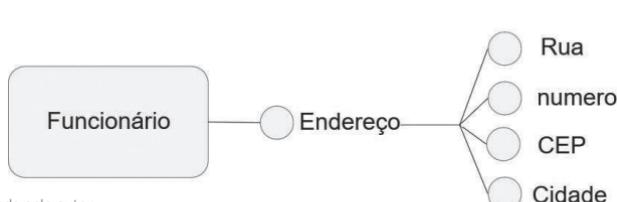
Ao caracterizar o atributo como simples ou composto, multivalorado ou de valor único, armazenado ou derivado, estamos realizando um tratamento para inserção de dados, pensando nas funcionalidades do Banco de Dados. As características que os atributos podem assumir estão detalhadas a seguir.

1.2.1 Atributo Simples x Composto

Dizemos que um atributo é simples quando ele é indivisível, ou atômico, por exemplo, Tamanho é um atributo simples da entidade Tênis, pois sua especificação é uma informação indivisível, como 37 ou 41.

Entretanto, quando o atributo pode ser dividido em outros atributos ele é chamado de composto, por exemplo, Endereço é um atributo composto da entidade Funcionário, já que ele pode ser dividido em atributos como Rua, Número e CEP. Quanto à representação gráfica dos atributos simples e compostos, podemos observar na Figura 2.4:

Figura 2.4 | Atributo Simples e Atributo Composto



Fonte: elaborada pelo autor.

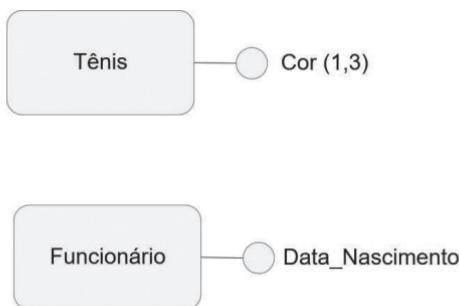
Essa representação é útil quando precisamos dos dados internos do atributo, por exemplo, ao utilizar só o número do CEP para cálculo do frete ou busca do endereço.

1.2.2 Atributo de Valor único x Multivalorado

Um outro aspecto que caracteriza os atributos é a quantidade de valores que um atributo pode assumir. Por exemplo, **cor**, é um atributo multivalorado da entidade Tênis, pois o tênis pode ter mais de uma cor.

Já o atributo Data de Nascimento da entidade Funcionário é um atributo de valor único, por motivos naturais. A representação destes atributos está ilustrada na Figura 2.5:

Figura 2.5 | Atributo Multivalorado e Atributo de Valor único



Fonte: elaborada pelo autor.

Quando definimos que o atributo é do tipo multivalorado devemos informar qual a quantidade mínima e máxima, ou seja, quantos valores ele poderá assumir, sendo que para o atributo **cor** definimos entre 1 e 3.

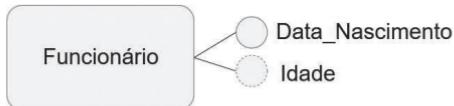
1.2.3 Armazenados x Derivados

Ainda quanto às características dos atributos, estes podem ser inerentes àquelas entidades, isto é, seus valores passaram a existir quando demos vida ao objeto, e eles precisam estar armazenados, como o atributo Data de Nascimento da entidade Funcionário.

Entretanto, se tivermos em nossa entidade, além do atributo Data de Nascimento, a necessidade de manipular um atributo Idade, então não se faz necessário armazenar, de fato, o valor com a idade do funcionário. Devemos armazenar... sim armazenar somente a Data de Nascimento e a partir dela calcular a Idade somente quando esta informação precisar ser utilizada.

Quanto à ilustração dos atributos apresentados, é um círculo com seu contorno tracejado, como observado na Figura 2.6:

Figura 2.6 | Atributo Armazenado e Atributo Derivado



Fonte: elaborada pelo autor.



Questão para reflexão

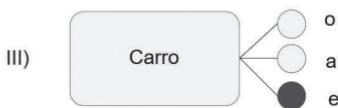
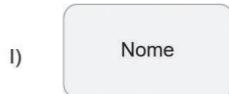
O processo de escolha de entidades e atributos é uma questão em que cabe mais o bom sendo ou a técnica?

Acertou se você falou um pouco dos dois! A técnica é essencial, porém devemos analisar bem todos os aspectos de interesse do mundo a ser descrito no projeto de Banco de Dados!

Para saber mais: Além de estudar mais sobre esses conceitos, podemos também criar as entidades com seus atributos, como se já fossemos desenvolvedores em atividade, para isso é legal ter ferramentas que facilitem a criação dos diagramas. Você pode encontrar algumas sugestões de ferramentas em: <<https://becode.com.br/diagramas-er-ferramentas/>>. Acesso em: 4 set. 2017.

Atividades de aprendizagem

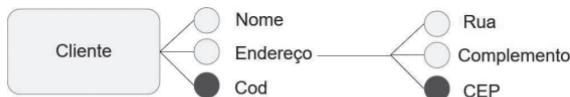
- Analise a construção das Entidades e dos Atributos a seguir:



A partir das construções I, II e III apresentadas, assinale a alternativa correta:

- a) Em I não temos entidade.
- b) Em II temos 3 entidades.
- c) Em III os atributos existem, porém não é indicado nomeá-los sem palavras de sentido real.
- d) Em II Documento e Cidade são atributos opcionais.
- e) Em II Nome é um atributo multivalorado.

2. Sabe-se que as entidades e os atributos são estruturas básicas no projeto e modelagem de banco de dados, por isso estes conceitos são essenciais para o início de um bom projeto de banco de dados. Avalie o esquema gráfico a seguir para responder à questão:



Fonte: elaborada pelo autor.

Assinale a alternativa que apresenta corretamente os conceitos envolvendo as entidades e os atributos:

- a) A entidade Cliente possui 6 atributos.
- b) O atributo Cliente possui 6 entidades.
- c) O atributo Endereço é um atributo composto de 3 outros atributos.
- d) O atributo Endereço possui 3 atributos derivados.
- e) Uma ocorrência de 'Nathalia Silva' é Nome, pois nome é a chave da entidade.

Seção 2

Relacionamentos

Introdução à seção

Ao passo que definimos quem são as entidades no nosso contexto devemos representar o relacionamento que essas entidades apresentam entre si, que basicamente descrevermos qual é a associação entre as entidades envolvidas.

Ao estabelecer os relacionamentos no nosso modelo, estamos aproximando o nosso projeto do conceito de banco de dados, com organização de informações, diferindo-se de um simples apanhado de dados.

2.1 Relacionamentos

O **relacionamento** é a única forma de associarmos uma entidade à outra, pois ele reflete o comportamento real que estamos modelando no projeto do Banco de Dados (MEDEIROS, 2013). Ele costuma ser atribuído como um verbo, sendo esse verbo escolhido como aquele que melhor representa a relação entre as entidades envolvidas.

Quando criamos um relacionamento devemos analisar se essa associação é válida para ambos os lados, e se faz sentido no Banco de Dados como representação fiel da realidade.

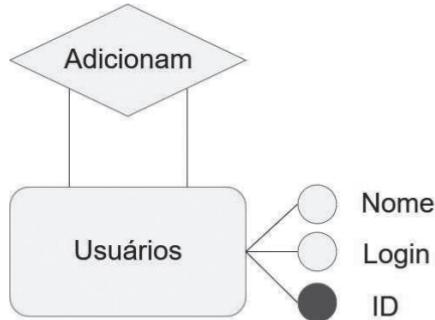
Os relacionamentos podem associar uma, duas ou mais entidades, conforme a necessidade de representação para casos reais. E nós vamos apresentar, discutir e representar os tipos mais comuns nesta seção.

2.1.1 Relacionamentos Unários

No caso de associação entre uma entidade somente, dizemos que o relacionamento é unário, recursivo ou é um autorrelacionamento. Esse tipo de relacionamento é estabelecido entre Usuários em uma rede social ou aplicativo de comunicação, onde usuários contatam outros usuários.

Assim, a representação do relacionamento "Adicionam" é um losango e temos somente uma entidade, a de Usuários, como ilustrado na Figura 2.7:

Figura 2.7 | Relacionamento Unário ou autorrelacionamento

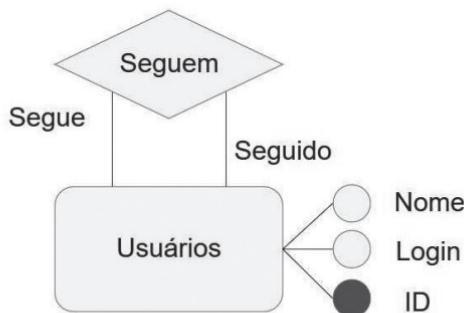


Fonte: elaborada pelo autor.

A representação deste relacionamento está clara, já que a relação é bidirecional, porém nos relacionamentos unários pode haver relações unidirecionais, como ocorre em algumas redes sociais .

Quando o significado implícito do relacionamento não é facilmente representado, ou para evitar ambiguidades, podemos detalhar o **papel** das entidades, que é adicionar informações textuais no diagrama daquele relacionamento (SILBERSCHATZ, KORTH e SUDARSHAN, 2012). Logo, a descriminação dos papéis não é obrigatória no DER, porém é recomendado na necessidade de explicitar o sentido da relação, como é representado na Figura 2.8:

Figura 2.8 | Relacionamento Unário com Papéis



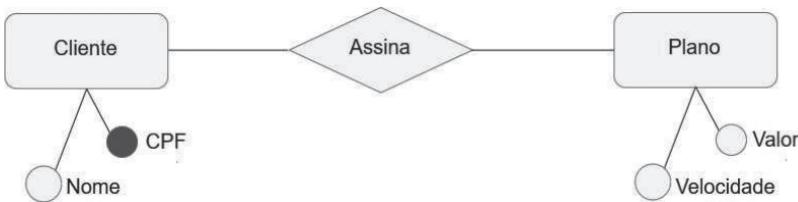
Fonte: elaborada pelo autor.

Podemos entender este exemplo unidirecional como o Usuário A que segue um outro Usuário B, ou este Usuário B que é seguido pelo Usuário A.

2.1.2 Relacionamentos Binários

Estendendo a quantidade de entidades envolvidas, temos o relacionamento binário. O mais comumente comumente encontrado, ocorre quando há a associação de duas entidades. É representado por um losango, entre as duas entidades, como podemos observar na Figura 2.9:

Figura 2.9 | Relacionamento binário entre Cliente e Plano:



Fonte: elaborada pelo autor.

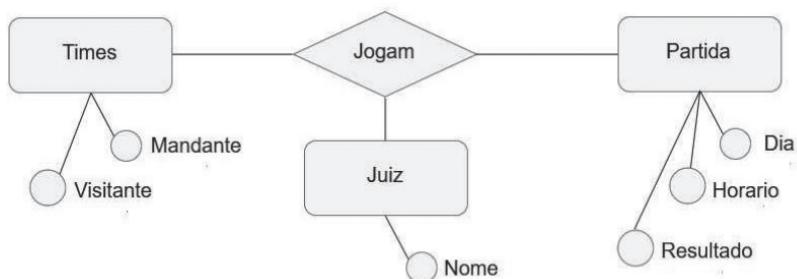
O relacionamento demonstrado na Figura 2.9 é de um cliente que assina um plano de internet, ou de maneira equivalente, que um plano é assinado por um cliente. Poderíamos ter mais atributos relacionados às duas entidades, entretanto, por simplificação foram colocados somente dois atributos, pois o foco desta seção são os relacionamentos.

2.1.4 Relacionamentos Ternários

Ainda quanto à quantidade de entidades envolvidas, podemos ter um relacionamento ternário, que descreve como três entidades são associadas, como é ilustrado na Figura 2.10:

¹Em redes sociais como o Instagram® é possível que um Usuário A siga ou acompanhe outro Usuário B, sem que o Usuário A seja seguido ou acompanhado pelo Usuário B. Em outras palavras, é como se eu fosse fã de uma pessoa famosa, e a seguisse, porém, a pessoa famosa não sabe da minha existência ou não tem interesse em acompanhar minhas atualizações na rede.

Figura 2.10 | Relacionamento ternário representando dados de um campeonato de futebol



Fonte: elaborada pelo autor.

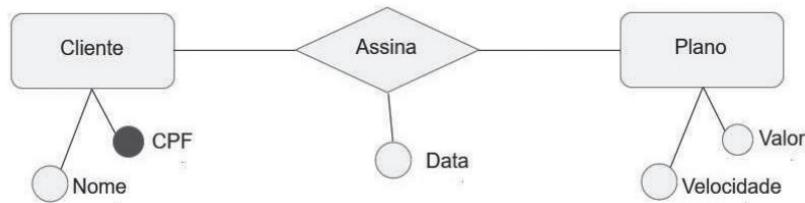
Estudamos relacionamentos de grau um (únário), grau dois (binário) e grau três (ternário), caso seja interessante estabelecer um relacionamento de grau quarto, denominado quaternário, ou envolvendo mais entidades, deve-se estender as características apresentadas nesta seção.

2.2 Cardinalidade

2.2.1 Cardinalidade em atributos

Em casos onde há a necessidade de adicionarmos informações pertinentes para o banco de dados, porém a determinada informação não é um atributo de nenhuma das entidades envolvidas, embora tenha significado, podemos atrelar um atributo ao relacionamento. Por exemplo, se é interessante registrar a data em que houve a assinatura do plano, no exemplo descrito pela Figura 2.9, observamos que a data não é uma característica do Cliente, e nem mesmo do Plano, portanto é um atributo do relacionamento Assina. Observe, na Figura 2.11, como seria o caso se a data de assinatura do plano da Figura 2.9 fosse necessário:

Figura 2.11 | Relacionamento binário com atributo no relacionamento



Fonte: elaborada pelo autor.

Observe que este atributo é monovalorado, ou seja, no mínimo um e no máximo um, porém podemos ter atributos em que seu mínimo é zero e seu máximo é n , ou seja, multivalorado.

2.2.2 Cardinalidade em relacionamentos

Um aspecto importante na associação de entidades através de relacionamentos é a cardinalidade dessa relação, que impõe limites inferiores e superiores para ocorrências de cada entidade envolvida no relacionamento ou mesmo atribui um número fixo para tais limites. A cardinalidade descrita no Projeto de Banco de Dados deve refletir as especificações do modelo real que desejamos implementar. Este dado tinha sido omitido até então para que você conhecesse os tipos de relacionamentos com uma ilustração mais limpa e simples. Agora que você já conhece é interessante que possamos discutir a cardinalidade dos relacionamentos e como representá-las graficamente.

2.2.2.1 Cardinalidade Mínima

Indica a mínima ocorrência para o relacionamento. Costuma ser 1 ou 0, isto é, podemos não ter aquela ocorrência ou ter no mínimo uma.

2.2.2.2 Cardinalidade Máxima

Indica a quantidade máxima de ocorrências daquela entidade no relacionamento. Costuma ser 1 ou n , isto é, podemos ter somente uma ocorrência ou muitas.

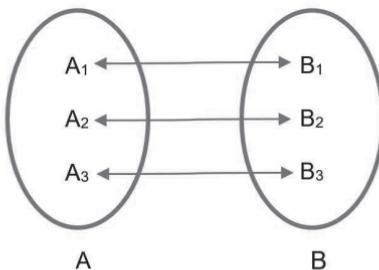
2.2.3 Cardinalidades e suas possíveis combinações

Podemos ilustrar o conceito de cardinalidade como a representação matemática da associação entre conjuntos, a partir das cardinalidades mínima e máxima, descritas pelo (\min, \max), temos quatro possibilidades de mapeamentos (SILBERSCHATZ, KORTH e SUDARSHAN, 2012):

2.2.3.1 Um para um

É quando uma entidade em A é associada a, no máximo, uma entidade em B, e uma entidade em B é associada a, no máximo, uma entidade em A. Esta associação é ilustrada na Figura 2.12:

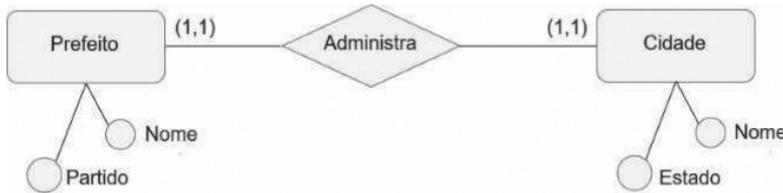
Figura 2.12 | Associação um para um



Fonte: elaborada pelo autor.

Podemos exemplificar esse caso com o cargo de prefeito das cidades no Brasil, onde somente uma pessoa exercer o cargo de prefeito de uma cidade, e uma cidade só pode ter um prefeito, isto pode ser representado pela Figura 2.13:

Figura 2.13 | Relacionamento um para um



Fonte: elaborada pelo autor.

O par (1,1) da esquerda representa que só faz sentido para o nosso cadastro se a pessoa for prefeito (mínimo 1 cidade) e que ela só pode ser prefeito de uma cidade (máximo 1). Já o par (1,1) da esquerda diz que a cidade existirá no cadastro se tiver um prefeito (mínimo 1) e que ela só está associada a um prefeito (máximo 1).

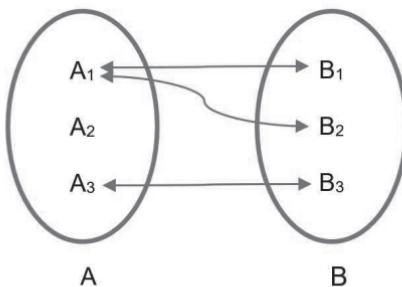
Em alguns diagramas entidade relacionamento também encontramos uma representação resumida, ou seja, somente 1:1, com

1 ao lado de prefeito e 1 ao lado de cidade.

2.2.3.2 Um para muitos

Descreve o caso de uma entidade em A ser associada a nenhuma ou várias entidades em B, e uma entidade em B ser associada a, no máximo, uma entidade em A. Esta associação é ilustrada na Figura 2.14:

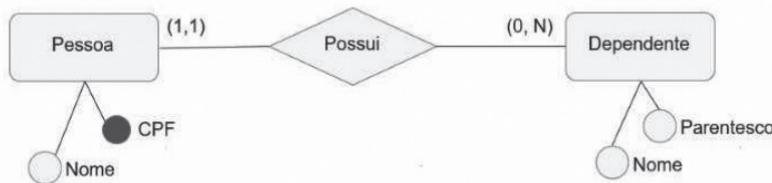
Figura 2.14 | Associação um para muitos



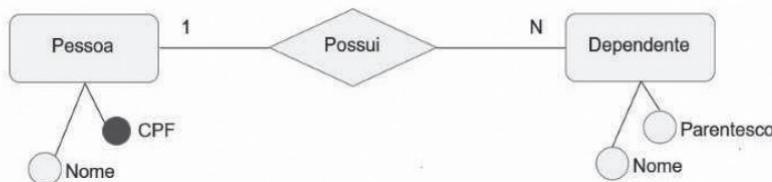
Fonte: elaborada pelo autor.

Na declaração do imposto de renda existe a possibilidade de incluir dependentes, dentre as regras desta relação podemos destacar que uma pessoa pode não incluir nenhum dependente (mínimo 0), ou incluir mais de um dependente (máximo n). Além disso, o dependente deve estar vinculado a uma pessoa (mínimo 1 e máximo 1), como representado pela Figura 2.15:

Figura 2.15 | Relacionamento um para muitos



a) Representação de cardinalidade mínima e máxima, entre ().



b) Representação de cardinalidade resumida.

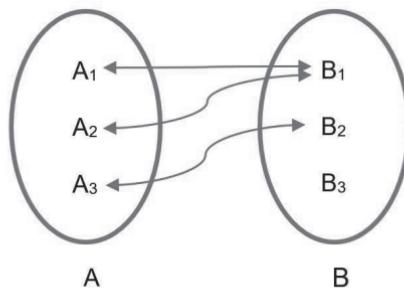
Fonte: elaborada pelo autor.

A característica descrita para a relação de pessoa e seus dependentes é exatamente a definição do relacionamento um para muitos, mantendo nossa representação de cardinalidade (min, max), lê-se que temos 1 pessoa para n dependentes, sendo que n também pode ser zero (0) e que todo dependente tem de estar relacionado a uma única pessoa.

2.2.3.3 Muitos para um

Ocorre quando uma entidade em A é associada a uma entidade em B, e nenhuma ou várias entidades em B são associadas a uma entidade em A. Esta associação é ilustrada na Figura 2.16:

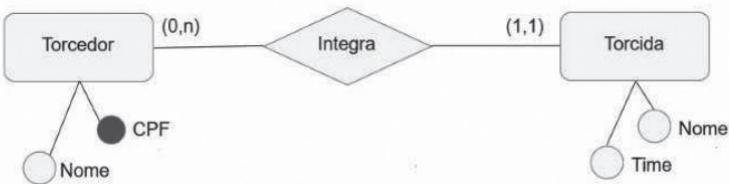
Figura 2.16 | Associação muitos para um



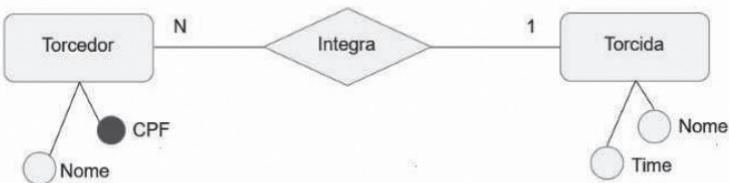
Fonte: elaborada pelo autor.

Para entender o relacionamento muitos para um podemos pensar em Torcedores que se cadastram para ter descontos em eventos esportivos, sendo que vários torcedores integram uma torcida ou ainda que nenhum torcedor faz parte daquela torcida, e espera-se que quando um torcedor se cadastre ele esteja vinculado à somente uma torcida, o que é representado pela Figura 2.17:

Figura 2.17 | Relacionamento muitos para um



a) Representação de cardinalidade mínima e máxima, entre ().



b) Representação de cardinalidade resumida.

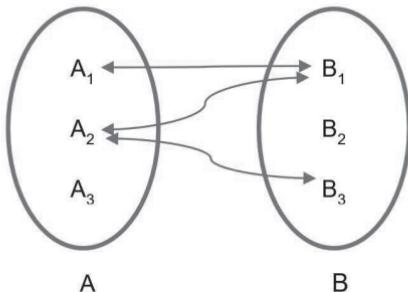
Fonte: elaborada pelo autor.

A característica descrita para a relação de torcida é análoga à definição do relacionamento um para muitos, porém a leitura é realizada no sentido contrário.

2.2.3.4 Muitos para muitos

Ocorre quando uma entidade em A é associada a qualquer quantidade (zero ou mais) de entidade em B, e uma entidade em B é associada a qualquer quantidade (zero ou mais) entidade em A. Esta associação é ilustrada na Figura 2.18:

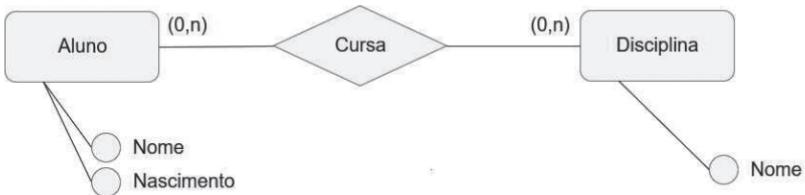
Figura 2.18 | Associação muitos para muitos



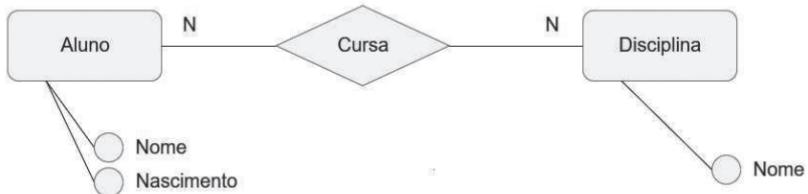
Fonte: elaborada pelo autor.

O relacionamento muitos para muitos pode ser ilustrado por meio de alunos e disciplinas de um curso de formação extracurricular, sendo que qualquer quantidade de alunos pode cursar qualquer quantidade de disciplinas, e ainda, pode haver alunos cadastrados sem disciplinas vinculadas e disciplinas disponíveis, mesmo que sem alunos, e isso tudo é demonstrado pelo diagrama na Figura 2.19:

Figura 2.19 | Relacionamento muitos para muitos



a) Com representação de cardinalidade mínima e máxima entre ().



b) Com representação de cardinalidade resumida.

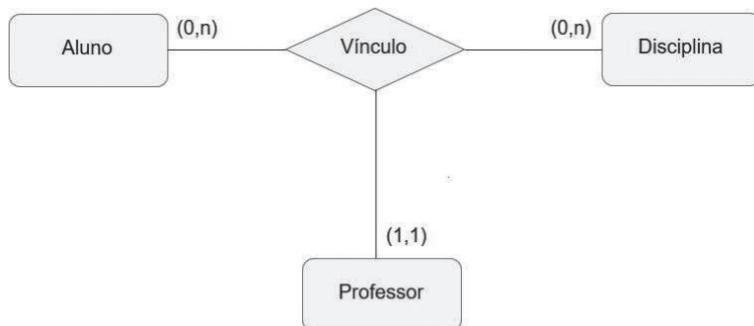
Fonte: elaborada pelo autor.

Em alguns livros também encontramos a representação N:M para salientar que Aluno e Disciplina não necessitam possuir a mesma quantidade de ocorrências.

2.2.3.5 Cardinalidade em relacionamentos ternários

Da mesma forma que caracterizamos a cardinalidade dos relacionamentos binários, podemos fazê-lo com relacionamentos ternários (MACHADO, 2014), vamos analisar alguns exemplos, começando pela Figura 2.20:

Figura 2.20 | Cardinalidade em relacionamentos ternários



Fonte: elaborada pelo autor.

Podemos entender esse relacionamento a partir de algumas considerações:

- Um aluno em uma disciplina sempre tem um professor;
- Um aluno pode estar vinculado a diversas disciplinas;
- Uma disciplina pode ter vários alunos e um professor;

Nesta seção, vimos as cardinalidades possíveis e como são descritas, durante o projeto devemos adotar a representação adequada à realidade que queremos modelar.

Quando modelamos casos de aplicações reais, esses diagramas costumam ficar mais elaborados e completos, conforme detalhamos o projeto e vemos a necessidade de adicionar entidade. Portanto, devemos nos atentar às propriedades de cada entidade e cada relacionamento, pois ao final veremos um projeto com muitas informações importantes relacionadas.



Questão para reflexão

Podemos sempre substituir um relacionamento N:N por dois relacionamentos 1:N e N:1?

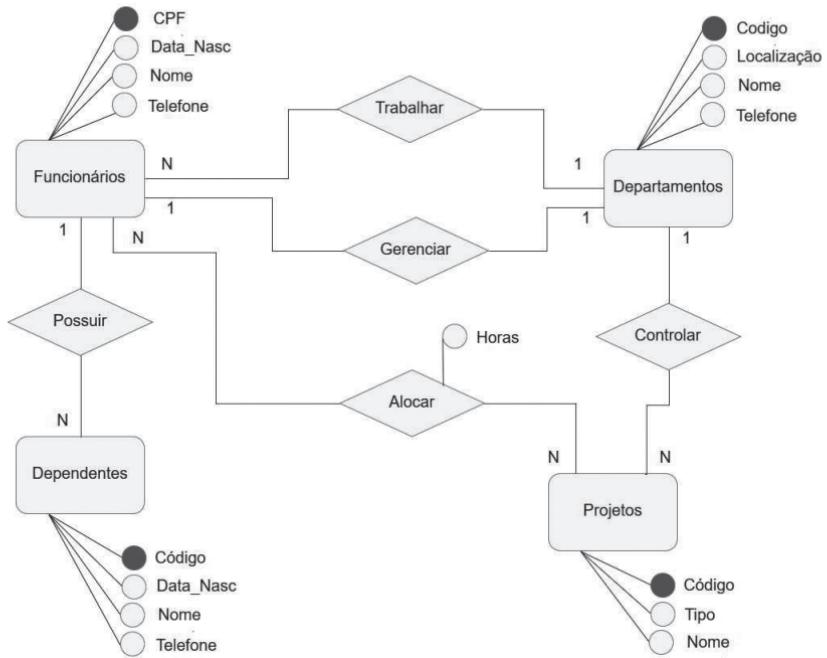


Para saber mais

Se você ficou interessado em continuar o estudo e aprofundar seus conhecimentos, acesse: <<https://www.youtube.com/watch?v=8fxKJWJcRTwl>>. Acesso em: 29 jul. 2017.

Atividades de aprendizagem

Analise o DER descrito a seguir para responder às questões:



Fonte: Diagrama Entidade Relacionamento (CARDOSO e CARDOSO, 2012)

1. Assinale a única alternativa que apresenta a informação correta sobre o diagrama proposto:

- a) O relacionamento trabalhar é ternário.
- b) O relacionamento possuir é do tipo N:N porque o dependente pode estar relacionado à somente um funcionário.
- c) Horas é um atributo do relacionamento.
- d) Gerenciar e Trabalhar é um autorrelacionamento entre entidades.
- e) Dependentes é um atributo da entidade possuir.

2. Com base nesse diagrama e nos seus conhecimentos em entidades e relacionamentos, assinale a alternativa que apresenta corretamente a descrição do sistema.

- a) Um funcionário pode trabalhar em mais de um departamento.
- b) Em um departamento pode trabalhar mais de um funcionário.
- c) Os funcionários têm que possuir no mínimo um dependente.
- d) Um projeto pode ser controlado por mais de um departamento.
- e) Vários funcionários estão associados a vários projetos.

Seção 3

Normalização

Introdução à seção

A fase de projeto de banco de dados também conta com a etapa de Normalização, um processo matemático formal, de E. F. Codd, datado de 1970, que tem por objetivo evitar anomalias de atualização no banco de dados por meio de teoria dos conjuntos (MACHADO, 2014).

Dentre os principais problemas que podemos encontrar nos bancos de dados sem Normalização, ao aplicarmos operações de inclusão, alteração e exclusão destacamos:

- Repetição de grupos de dados (atributos multivalorados);
- Dependências parciais de chave concatenada;
- Redundância desnecessária e indesejável de dados;
- Perdas de informações de interesse;
- Dependências transitivas de atributos;
- Dificuldade de representação consistente.

O processo de Normalização não garante que o projeto seja isento de falhas, até porque isso depende de um processo extenso de verificação que envolve diversas variáveis, porém os conceitos abordados ao longo desta seção visam minimizar a ocorrência de problemas conhecidos e tornar o projeto de banco de dados mais estável.

Pense na Normalização como um checklist de consistência, e que, com o cuidado de aplicarmos as Formas Normais, o nosso projeto com certeza estará melhor do que no início.

3.1 Primeira Forma Normal (1FN)

A Primeira Forma Normal (1FN) nos diz que:

Cada ocorrência da chave primária deve corresponder a uma e somente uma informação de cada atributo, ou seja, a entidade não deve conter grupos repetidos (multivalorados)(MACHADO, 2014).

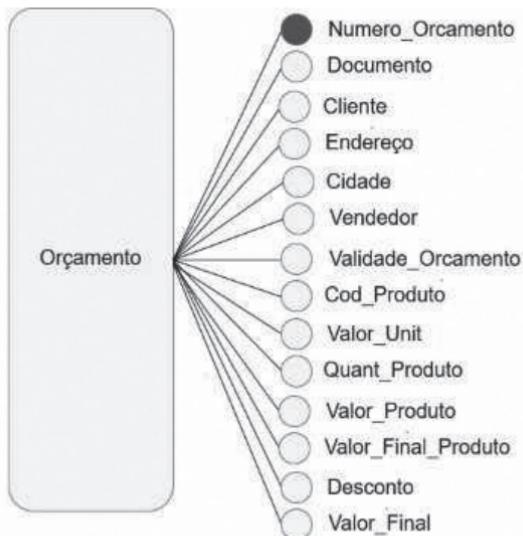
Para aplicar a 1FN devemos:

1. Identificar a entidade não normalizada e seguir os passos de 2 a 4;
2. Decompor (a entidade não normalizada) em n entidades, sendo

- n o número de conjuntos de atributos repetitivos;
3. Identificar a chave primária nas entidades criadas: chave primária da entidade original + chave primária do grupo (atributo do grupo repetitivo), o que chamamos de chave **concatenada**;
 4. Verificar se ainda há entidade não normalizada;

Vamos aplicar a 1FN em um exemplo prático para melhor visualização e entendimento, temos um projeto elaborado para a confecção de orçamentos, descrito na Figura 2.20:

Figura 2.21 | Diagrama Entidade Relacionamento para Orçamentos:



Fonte: elaborada pelo autor.

Analizando o projeto é possível perceber uma certa repetição em alguns itens ao colocarmos todos estes atributos na Entidade Orçamento, até porque o atributo Valor Unitário (Valor_unit) não diz respeito ao Orçamento ou Desc_Produto, a descrição de cada produto não deve estar sob responsabilidade da Entidade Orçamento, na necessidade de alterar esse valor precisaríamos acessar todos os Orçamentos que contém aqueles dados, ou teríamos inconsistência entre nossas tabelas.

Descrição seguindo os passos da 1FN:

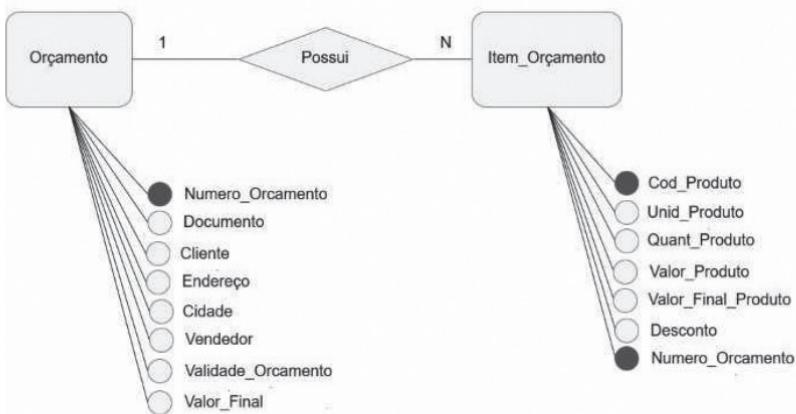
Aplicando o passo 1 identificamos que Orçamento não está na Primeira Forma Normal.

Para aplicar o passo 2, identificamos o **grupo repetitivo** da Entidade

original, que é o grupo que diz respeito ao cliente e aos produtos, portanto criaremos **novas entidades** para agrupar tais dados.

Aplicando o passo 3, criamos a **chave concatenada** para a nova entidade e estabelecemos o relacionamento entre as entidades. O resultado da aplicação da 1FN está descrito na Figura 2.21:

Figura 2.22 | Diagrama Entidade Relacionamento para Orçamentos na 1FN:



Fonte: elaborada pelo autor.

Vamos aproveitar os detalhes deste DER para discutir sua consistência:

Por meio da cardinalidade concluímos que um orçamento pode possuir um ou mais itens, e que cada item lançado no sistema estará vinculado a um orçamento. Ainda, é possível afirmar que cada item poderá ser recuperado individualmente através da chave concatenada, que é a resultante das chaves Num_Orcamento e Cod_Produto.

Estas ações tornaram nosso sistema mais confiável, principalmente no que concerne à controle de alterações, porém ainda podemos identificar problemas neste projeto, como é o caso de atributos que estão **dependentes** de outras chaves.

3.1.1 Dependência parcial ou total

O conceito de **dependência parcial** possui a seguinte definição:

Um atributo ou conjunto de atributos A é dependente funcional de B se, a cada valor de B, existir um único valor de A, estando A e B na mesma entidade (MACHADO, 2014).

Repare que A possivelmente é uma chave na entidade, ocorrendo então a **dependência parcial**. Da mesma forma, o conceito de dependência total ocorre quando esta chave é concatenada.

Isso quer dizer que os atributos Desc_Produto ou Valor_Unit possuem dependência parcial sobre a chave Cod_Produto e possuem dependência total dos atributos que formam a chave concatenada como Cod_Produto e Num_Orçamento. Para resolver essa situação temos a Segunda Forma Normal, apresentada na próxima seção.

3.2 Segunda Forma Normal (2FN)

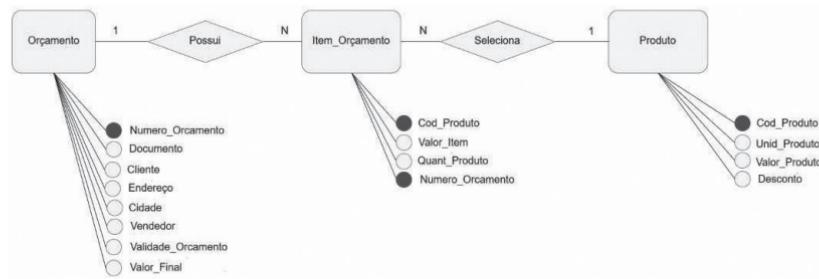
A Segunda Forma Normal (2FN) procura aumentar a estabilidade de projeto de banco de dados através da análise de chaves concatenadas e dependências parciais, e elaborar estratégias para que elas não estejam mais presentes na modelagem (MACHADO, 2014). Para isso vamos procurar decompor, com cautela e segurança, as entidades não normalizadas seguindo esse critério da 2FN, após verificar que ela já está na 1FN.

Para aplicar a 2FN devemos:

1. Identificar a entidade não normalizada e seguir os passos de 2 a 4;
2. Criar entidades que resolvam o problema da dependência funcional parcial;
3. Identificar as chaves nas entidades;
4. Verificar se ainda há entidades não normalizadas;

Vamos aplicar a 2FN no nosso exemplo do orçamento, afinal já discutimos que alguns atributos da Entidade Item_Orçamento possuem dependência parcial. O nosso diagrama revisado com a aplicação da 2FN está descrito na Figura 2.22:

Figura 2.23 | Diagrama Entidade Relacionamento para Orçamentos na 2FN



Fonte: elaborada pelo autor.

Observe que os atributos Valor_Item e Quant_Produto da entidade Item_Orçamento não dependem parcialmente de Cod_Produto porque não haverá repetição, da mesma forma, na entidade Produto, os atributos Desc_Produto e Unid_Produto também não aparecerão de forma repetida. Então nosso projeto está normalizado segundo a 2FN.

3.3 Terceira Forma Normal (3FN)

A Terceira Forma Normal (3FN) procura tornar o projeto de banco de dados mais consistente com a eliminação das dependências transitivas (MACHADO, 2014).

O conceito de dependência transitiva é quando um atributo ou conjunto de atributos A dependem de um atributo B sem que haja significado na relação entre eles, ou seja, os atributos que aparecem em uma entidade, todavia ficariam melhor posicionados se fossem definidos em outra entidade.

Você deve ter notado neste último parágrafo uma certa semelhança com os conceitos de Encapsulamento de Orientação a Objeto. Lhe informo que, felizmente, essas tarefas seguem as mesmas premissas: a de que devemos organizar as estruturas dos nossos sistemas conforme as responsabilidades inerentes à cada Entidade/ Objeto.

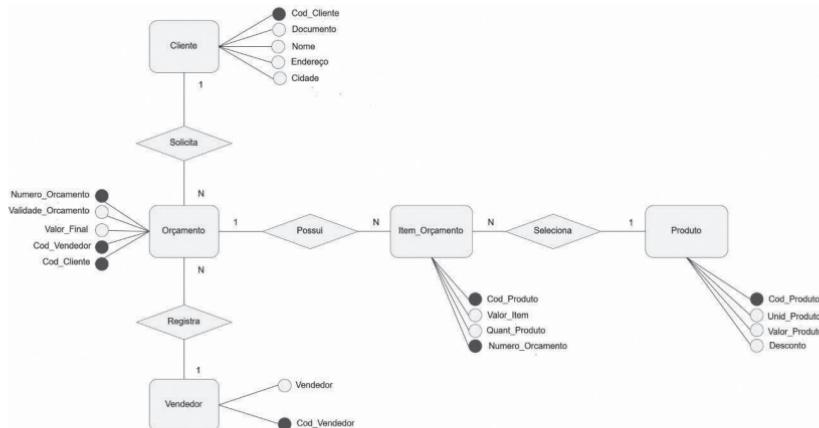
Então vamos procurar decompor, de maneira coerente e coesa, as entidades não normalizadas seguindo esse critério da 3FN, após confirmar que o projeto já estava na 1FN e na 2FN.

Para aplicar a 3FN devemos:

1. Identificar a entidade não normalizada e seguir os passos de 2 a 4;
2. Criar entidades que resolvam o problema da dependência transitiva;
3. Identificar as chaves nas entidades;
4. Verificar se ainda há entidades não normalizadas;

Ao aplicar a 3FN, no nosso exemplo do orçamento, identificamos que os dados de Cliente e de Vendedor podem ser descritos em entidades específicas para eles, e isso elimina a dependência transitiva. O nosso diagrama atualizado com a aplicação da 3FN está descrito na Figura 2.23:

Figura 2.24 | Diagrama Entidade Relacionamento para Orçamentos na 3FN



Fonte: elaborada pelo autor.

Com estas adaptações o nosso projeto finalmente está normalizado segundo a 3FN.

A literatura da área ainda apresenta uma continuação das Formas Normais, que são a Forma Normal de Boyce/Codd (FNBC), a Quarta Forma Normal (4FN) e a Quinta Forma Normal (5FN), e em geral, a decomposição até a 3FN é suficiente para atender à maioria dos projetos de bancos de dados (HEUSER, 2009).



Questão para reflexão

As formas normais precisam ser aplicadas na ordem?

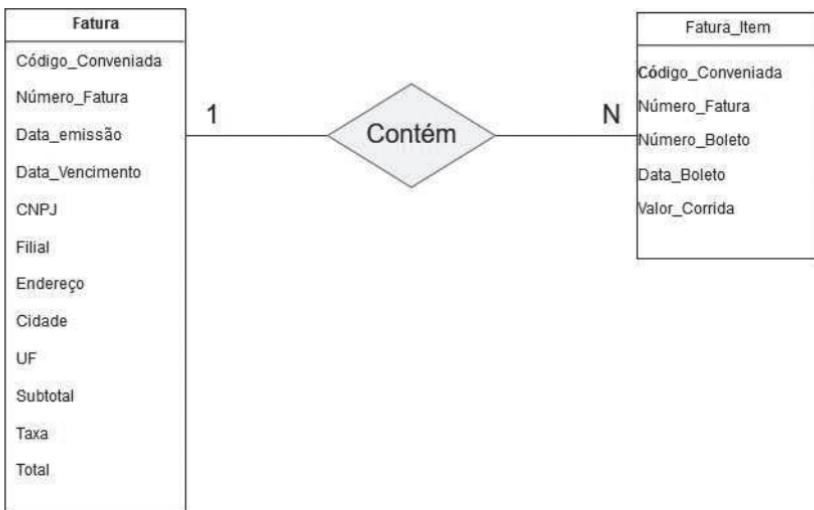


Para saber mais

Você quer conhecer mais sobre os conceitos de aplicação das formas normais? Disponível em: <<http://video.devmedia.com.br/video/images/websysimg/SQL/87/Normalizacao/Normalizacao.html>>. Acesso em: 29 jul. 2017.

Atividades de aprendizagem

1. Analise o Diagrama Entidade Relacionamento a seguir:



Fonte: Diagrama Entidade Relacionamento (PUGA, 2013)

Assinale a alternativa correta sobre a aplicação das formas normais no diagrama analisado:

- a) O diagrama está normalizado segundo todas as formas normais.
- b) O diagrama está normalizado até a 3FN.
- c) O diagrama está normalizado até a 2FN.
- d) O diagrama está normalizado até a 1FN.
- e) O diagrama não está normalizado.

2. Considerando o DER do exercício anterior, o que deve ser feito para colocar o projeto nas próximas formas normais:

Fique ligado

Você viu neste capítulo que descrevemos nosso sistema de interesse através do relacionamento entre entidades, e que o diagrama que representa estas relações carrega uma grande quantidade de informações. A cardinalidade e o grau do relacionamento interferem diretamente no entendimento das funcionalidades e interações do sistema, e uma boa escolha das entidades e dos atributos é o início de um projeto coeso. Você viu como aplicar as formas normais através de um passo a passo e de um exemplo.

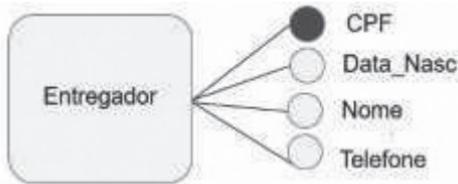
Para concluir o estudo da unidade

O projeto de banco de dados não é uma tarefa trivial, entretanto também não é um objetivo inalcançável. Como um bom profissional da área é interessante que você conheça os conceitos envolvidos e saiba aplicá-los nas situações práticas. Ah, e não se esqueça, a resolução de exercícios e desenvolvimento de diagramas é altamente recomendado!

Este é um assunto de grande abrangência e nível de detalhes, e tem muito campo de atuação profissional!

Atividades de aprendizagem da unidade

Considere o diagrama a seguir para responder às questões 1 e 2:



Fonte: elaborada pelo autor.

- 1.** Considere que o entregador pode ter diversos números de telefone celular, a entidade passaria a não estar normalizada segundo a _____, assim, deve-se decompor a entidade Entregador em outras duas entidades normalizadas.

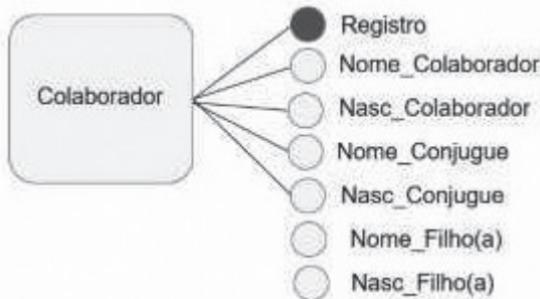
Assinale a alternativa que completa corretamente a lacuna:
a) 1FN.
b) 2FN.

- c) 3FN.
- d) 4FN.
- e) 5FN.

2. Analise as alternativas a seguir e assinale aquela que apresenta a informação correta:

- a) O atributo CPF de Entregador é derivado.
- b) O atributo Nome de Entregador é multivalorado.
- c) A entidade Entregador possui quatro relacionamentos.
- d) O atributo Telefone pode ser composto.
- e) O atributo Entregador possui quatro entidades.

3. Observe a seguinte entidade Colaborador:



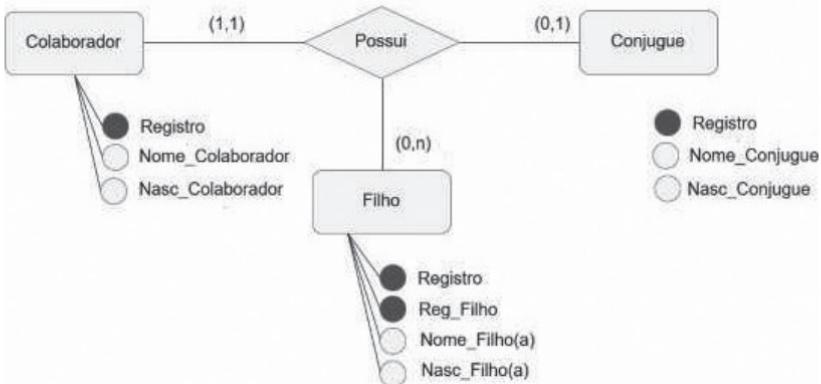
Fonte: elaborada pelo autor.

O diagrama representado não está na forma normal, aplique os conceitos estudados para deixá-lo na forma normal.

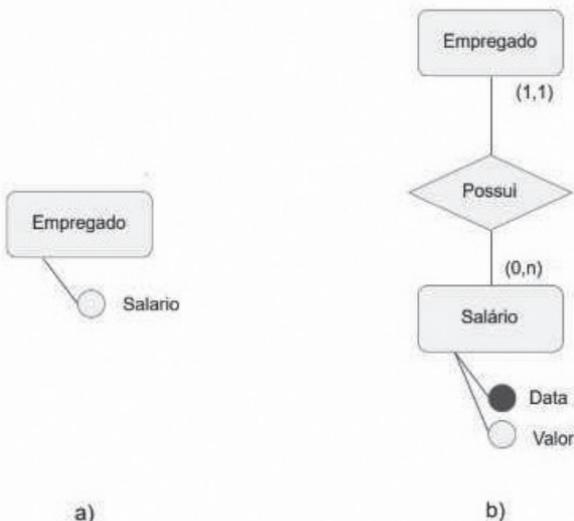
Assinale a alternativa correta sobre o diagrama decomposto na forma normal.

- a) O número de entidades continuaria o mesmo.
- b) O número total de atributos cairia para quatro.
- c) Poderíamos estabelecer o projeto com três entidades: Colaborador, Filho e Número de Registro.
- d) Poderíamos retirar os dados de filhos do projeto.
- e) O ideal seria que a entidade Colaborador só tivesse atributos pertinentes ao próprio colaborador, e o restante em entidades como Filho e Conjugue.

Ilustração do Gabarito da atividade 3 (opcional aparecer):



4. Observe o diagrama a seguir:



Fonte: (HEUSER, 2009)

Assinale a alternativa correta sobre a representação apresentada no diagrama:

- a) As duas representações possuem o mesmo significado.
- b) A cardinalidade da representação em (b) está invertida.
- c) Em (a) somente o salário atual é armazenado e em (b) possui o histórico.
- d) Em (a) não há entidade.
- e) Em (b) o relacionamento é do tipo 0:1.

5. 5 – Relacione a primeira coluna com a segunda coluna, conforme o tipo de relacionamento encontrado:

- (A) Um para um
 - (B) Um para muitos
 - (C) Muitos para um
 - (D) Muitos para muitos
 - () Cantores e músicas
 - () A pessoa e seu documento de habilitação.
 - () (0,N) : (1,1)
 - () A cachorra e seus filhotes
- a) DACB.
 - b) ADCB.
 - c) ADBC.
 - d) DCBA.
 - e) BACD.

Referências

- CARDOSO, Virgínia; CARDOSO, Giselle. **Sistema de banco de dados**: uma abordagem introdutória e aplicada. São Paulo: Saraiva, 2012. 144 p.
- ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011. 790 p.
- HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. 281 p.
- MACHADO, Felipe Nery Rodrigues. **Projeto e implementação de banco de dados**. São Paulo: Érica, 2014. 401 p.
- MEDEIROS, Luciano Frontino de. **Banco de dados**: princípios e prática. Curitiba: InterSaber, 2013. 183 p.
- PUGA, Sandra; FRANÇA, Edson; GOYA, Milton. **Banco de Dados**: implementação em SQL, PL/SQL e Oracle 11g. São Paulo: Pearson Education do Brasil, 2013. 332 p.
- RAMARKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de gerenciamento de banco de dados**. 3. ed. Porto Alegre: McGraw-Hill, 2011. 884 p.
- SILBERSCHATZ, Abrahan; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro: Elsevier, 2012. 961 p.

Sistemas de gerenciamento de banco de dados: PostgreSQL

Gisele Alves Santana

Objetivos de aprendizagem

Nesta unidade, você será levado a entender a importância dos sistemas de gerenciamento de banco de dados (SGBDs), assim como os principais conceitos relacionados à arquitetura cliente servidor. Você irá exercitar conceitos aprendidos através de um estudo de caso envolvendo o SGBD chamado PostgreSQL. Você aprenderá os passos para a instalação do SGBD, assim como as etapas para a criação de um novo banco de dados, entendendo a maneira de criação de tabelas e relacionamentos.

Seção 1 | PostgreSQL: conceitos, instalação e configuração

Nesta seção, você estudará sobre os principais conceitos relacionados aos sistemas de gerenciamento de banco de dados e aprenderá sobre o funcionamento da arquitetura cliente servidor. Esta seção também apresenta detalhadamente os passos para a instalação e configuração do SGBD chamado PostgreSQL.

Seção 2 | PgAdmin: criação de um banco de dados e tabelas

Nesta seção, você vai aprender através de exemplos práticos as etapas para a criação de um novo banco de dados utilizando a ferramenta pgAdmin, que é instalada automaticamente e em conjunto com o PostgreSQL. Serão criadas tabelas para esse novo banco de dados e inseridos vários campos nas mesmas, o que possibilitará que você exercente os conceitos aprendidos e se familiarize com a ferramenta gráfica disponibilizada pelo PostgreSQL.

Introdução à unidade

Nesta unidade, você aprenderá sobre os sistemas de gerenciamento de banco de dados (SGBDs), que como o próprio nome diz, são aplicativos responsáveis por gerenciar um grande volume de informações armazenadas em diversos bancos de dados.

Dentre os vários SGBDs disponíveis atualmente, destaca-se o PostgreSQL. Esse SBGD é relacional e suporta grande parte do padrão SQL, oferecendo características modernas, como: integridade transacional, chaves estrangeiras, controle de integridade etc.

O PostgreSQL utiliza a arquitetura cliente servidor. Esta unidade apresenta as características que envolvem esse tipo de arquitetura, assim como a maneira que o PostgreSQL utiliza esse recurso.

Esta unidade também apresenta um estudo prático envolvendo o PostgreSQL. É apresentado um simples passo a passo para a instalação e configuração do SGBD.

Para finalizar, esta unidade apresenta todas as etapas para a criação de um novo banco de dados através de uma interface gráfica disponibilizada pelo PostgreSQL, chamada de pgAdmin III. Nessa interface, serão criadas tabelas com campos, relacionamentos e chaves.

Seção 1

PostgreSQL: conceitos, instalação e configuração

Introdução à seção

Esta seção apresenta uma pequena introdução sobre conceitos relacionados aos sistemas de gerenciamento de bancos de dados. Primeiramente, são apresentados conceitos fundamentais sobre SGBDs e o tipo de arquitetura cliente servidor. Em seguida, é realizado um estudo de caso utilizando o SGBD chamado PostgreSQL. Esta seção também apresenta o passo a passo para a instalação e configuração do PostgreSQL para que mais adiante seja possível a criação de um banco de dados e manipulação de tabelas.

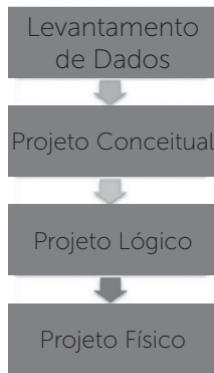
1.1 Sistemas de Gerenciamento de Banco de Dados (SGBDs)

Os Sistemas de Gerenciamento de Banco de Dados (SGBD), como já estudado na Unidade 1, são responsáveis por gerenciar uma grande quantidade de informações. Os SGBDs evitam a redundância de dados e inconsistência, facilitando o acesso e também proporcionando segurança no acesso aos dados de um banco de dados.

Os SGBDs ocultam do usuário detalhes de armazenamento dos dados, utilizando o conceito de abstração. Na abstração física, esses sistemas ocultam a maneira de armazenamento dos dados na memória, na abstração conceitual detalhes e relacionamento entre os dados são ocultados. Já na abstração de visão, os SGBDs expõem somente a parte do banco de dados que os usuários precisam enxergar.

Relembrando que para um bom projeto de um banco de dados quatro etapas são necessárias, conforme ilustrado na Figura 3.1.

Figura 3.1 | Etapas do projeto de um BD



Fonte: elaborada pela autora.

Na etapa de levantamento de dados podem ser realizadas reuniões com o cliente a fim de se levantar os requisitos. No projeto conceitual é realizada a organização dos dados através de um diagrama conceitual. No projeto lógico, o SGBD a ser utilizado já deve ser definido. No projeto físico é realizada a geração de scripts para a criação do banco de dados.

Nesta seção, estudaremos os conceitos, arquitetura e instalação do SGBD chamado PostgreSQL.



Para saber mais

Existem vários tipos de SGBDs disponíveis atualmente. Você pode conferir os mais utilizados no link a seguir.

Disponível em: <<https://www.opservices.com.br/principais-diferencias-entre-banco-de-dados/>>. Acesso em: 17 ago. 2017.

1.2 Arquitetura cliente servidor

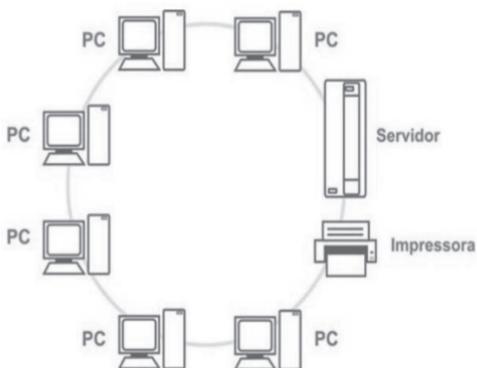
É importante conhecer a arquitetura básica do PostgreSQL e entender como as partes deste SGBD interagem. Como o PostgreSQL utiliza o modelo cliente servidor, é necessário entender como essa tecnologia funciona.

Basicamente, um sistema de gerenciamento de banco de dados cliente servidor divide o processamento em dois sistemas: o cliente e o

servidor. O cliente é geralmente um computador pessoal que executa a aplicação do BD. O servidor é responsável por executar todo o SGBD, ou parte dele. Dessa maneira, são utilizados computadores pessoais e servidores conectados em uma rede local em que o processamento é dividido entre clientes e servidores.

Uma rede local, também conhecida como LAN (Local Area Network), pode ser definida como o conjunto de computadores que pertence a uma mesma organização, conectados por uma rede, em uma pequena área geográfica (TANENBAUM, 2003), conforme exemplo ilustrado na Figura 3.2.

Figura 3.2 | Exemplo de LAN



Fonte: elaborada pela autora.

Um SGBD que utiliza a arquitetura cliente servidor realiza todo o processamento no computador cliente, responsável por rodar o aplicativo do banco de dados. O computador servidor é responsável por hospedar o SGBD propriamente dito, ou parte dele. A maioria das estruturas que utilizam a arquitetura cliente servidor dedica um computador exclusivo para rodar o SGBD.

O aplicativo de banco de dados existente no computador cliente é chamado de front-end, sendo responsável pelo processamento de entrada e saída dos dados dos usuários. O sistema *back-end*, existente no servidor, é responsável por manipular o processamento dos dados e acesso ao disco.

Como exemplo de um processamento entre cliente e servidor imagine um usuário localizado no computador cliente que gera uma

consulta de dados para o servidor de BD. O servidor de BD executa a pesquisa e retorna somente os dados que foram solicitados pelo usuário.

Na arquitetura cliente servidor, o cliente é sempre responsável por iniciar uma comunicação, sendo que o servidor trabalha de forma reativa, respondendo às requisições do cliente.

A grande vantagem desse tipo de arquitetura é a divisão de processamento entre dois sistemas, o que aproveita muito bem as características de cada máquina para cada aplicação específica.

A maior desvantagem dos sistemas de banco de dados cliente servidor é que eles exigem que os dados sejam armazenados em um único sistema. Esse fato pode ser um problema para as grandes empresas, que talvez precisem suportar usuários espalhados por uma ampla região geográfica.

No caso do PostgreSQL, uma sessão possui os seguintes processos:

- Processo servidor (*postmaster*): responsável por gerenciar os arquivos de BD e receber conexões dos aplicativos cliente, executando ações em nome desses clientes.
- Aplicativo cliente do usuário (*frontend*): responsável por executar operações de BD.

O servidor PostgreSQL pode abordar várias conexões de cliente simultaneamente. Para que isso seja possível, para cada conexão, é iniciado um novo processo. Em seguida, o novo processo servidor e o cliente se comunicam sem a interferência do processo *postmaster* original. Dessa maneira, o *postmaster* sempre está executando e aguardando por novas conexões dos clientes, enquanto os clientes e seus processos agregados aparecem e desaparecem.



Para saber mais

No link a seguir você encontra mais explicações sobre redes locais, assim como os principais tipos de redes existentes.

Disponível em: <<https://canaltech.com.br/infra/lan-wlan-man-wan-pan-conheca-os-principais-tipos-de-redes/>>. Acesso em: 17 ago. 2017.



Questão para reflexão

Para que você consiga abrir páginas web, você utiliza um navegador. Assim, podemos dizer que o navegador é o cliente e o site é o servidor, e as redes que utilizam servidores são chamadas do tipo Cliente Servidor?

1.3 PostgreSQL

O PostgreSQL é um dos mais avançados, poderosos e populares SGBDs com código aberto (*opensource*), sendo desenvolvido pela *PostgreSQL Global Development Group*. Esse SGBD possui todos os recursos de um grande banco de dados e é capaz de administrar uma vasta quantidade de informações.

A maneira mais útil de se aprender como funciona um SGBD é colocando os conceitos em prática. Para isso, nós vamos primeiramente instalar o PostgreSQL.

1.3.1 Instalação

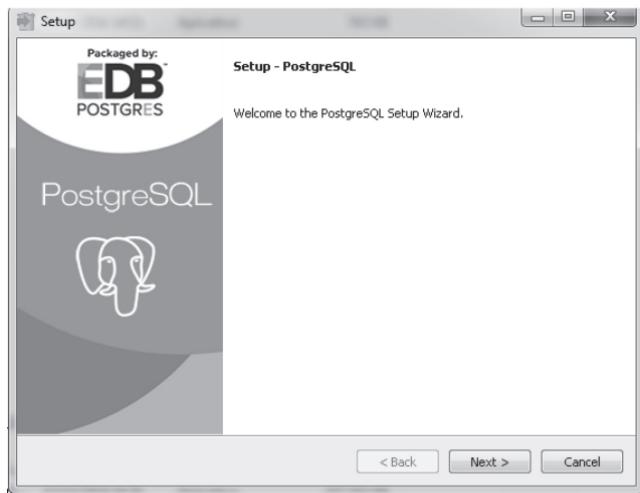
Antes de aprendermos a manipular e criar tabelas no PostgreSQL, é necessária a instalação do SGBD. Existem várias versões disponíveis para a instalação em diferentes sistemas operacionais (Windows, Linux, Solaris etc.).

O download do PostgreSQL pode ser realizado através do seguinte endereço eletrônico:

<<https://www.postgresql.org/download/>>. Nós iremos instalar a versão 9.5.8, pois é a atual e mais recente desse SGBD.

Depois de efetuado o download, execute o instalador “postgresql-9.5.8.exe”. Ao executar o arquivo, aparecerá a tela inicial de instalação, ilustrada na Figura 3.3.

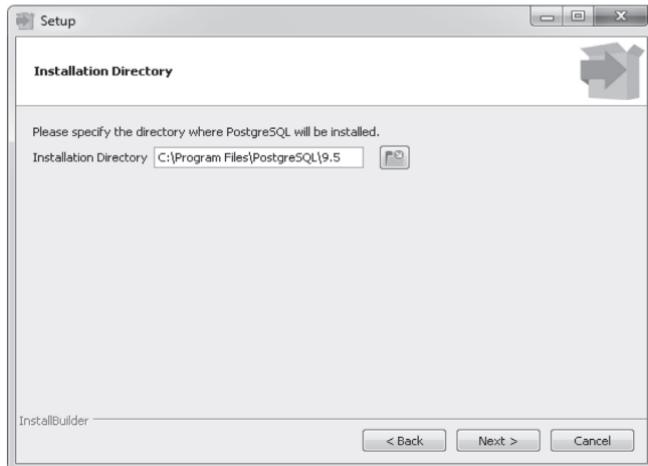
Figura 3.3 | Tela inicial de instalação do PostgreSQL



Fonte: elaborada pela autora.

Ao clicar no botão “Next” será solicitado o diretório para a instalação do SGBD. Por padrão, o instalador cria um novo diretório, conforme Figura 3.4. Se você quiser instalar em outro caminho, apenas troque o diretório padrão pelo local desejado.

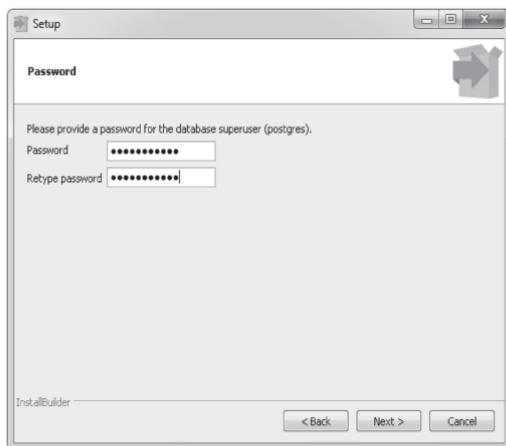
Figura 3.4 | Seleção do diretório de instalação



Fonte: elaborada pela autora.

No próximo passo de instalação será solicitada uma senha, conforme ilustrado na Figura 3.5. Você deve memorizar essa senha, pois precisará da mesma toda vez que quiser acessar o SGBD já instalado. Por padrão, o usuário do PostgreSQL é definido como “postgres”.

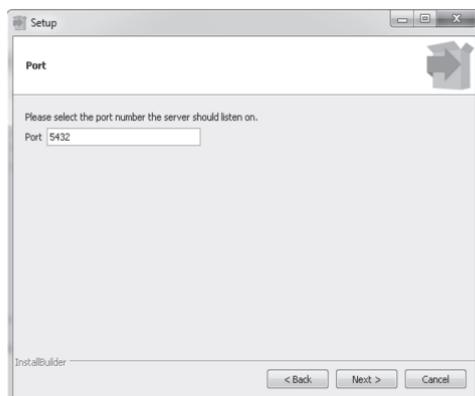
Figura 3.5 | Escolha de senha



Fonte: elaborada pela autora.

O próximo passo é a escolha da porta que executará o serviço do SGBD. Por padrão, o PostgreSQL utiliza a porta 5432, de acordo com a Figura 3.6.

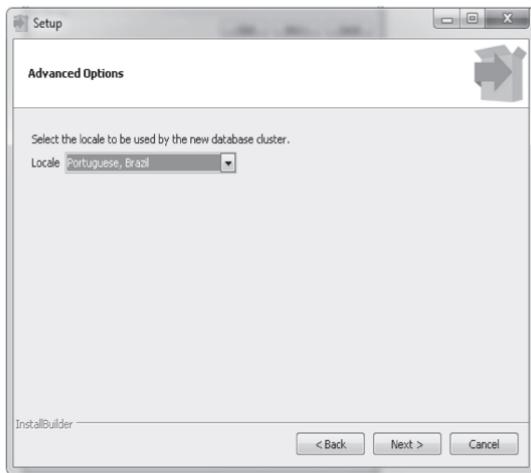
Figura 3.6 | Porta do PostgreSQL



Fonte: elaborada pela autora.

Na próxima tela deve ser feita a escolha pelo idioma do SGBD. Selecione o idioma que você deseja, como mostra a Figura 3.7. Ao clicar em “Next” a instalação é iniciada.

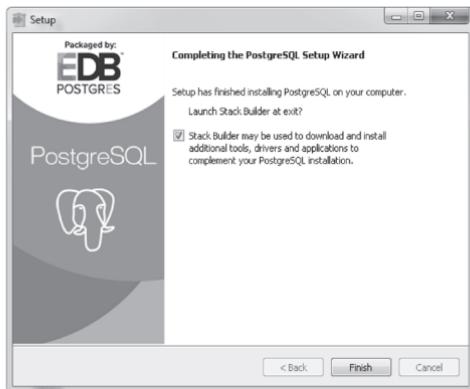
Figura 3.7 | Escolha de idiomas



Fonte: elaborada pela autora.

A tela que aparecerá em seguida é mostrada na Figura 3.8, que indica a finalização da instalação. Deixe selecionado o campo “Stack Builder” se você quiser que ferramentas adicionais sejam instaladas. Após, clique em “Finish”.

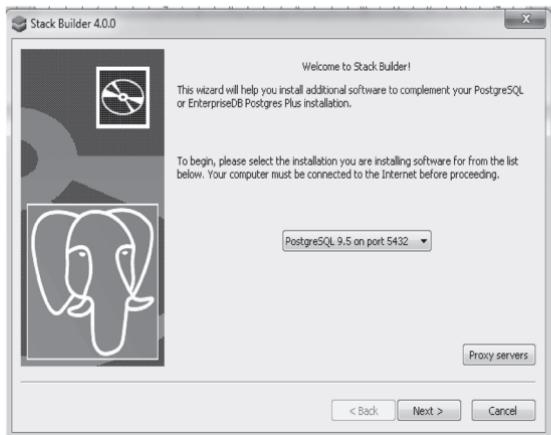
Figura 3.8 | Finalizando a instalação



Fonte: elaborada pela autora.

Se você preferiu baixar e instalar ferramentas adicionais, a tela mostrada na Figura 3.9 irá aparecer. Selecione a versão e a porta do SGBD escolhidas e clique em "Next".

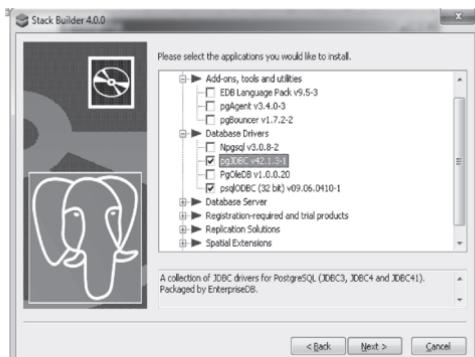
Figura 3.9 | Ferramentas adicionais



Fonte: elaborada pela autora.

Na tela seguinte, marque as opções "pgJDBC" e "psqlODBC", conforme Figura 3.10. Essas opções correspondem aos drivers para conexão com o banco de dados. O Driver JDBC (Java Database Connectivity) permite a conexão de uma aplicação em Java com o Postgresql. O Driver ODBC (Open Data Base Connectivity) permite a conexão com qualquer outra aplicação.

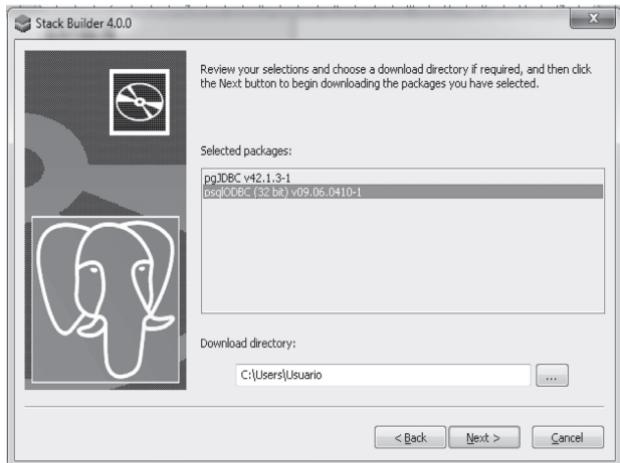
Figura 3.10 | Seleção dos drivers



Fonte: elaborada pela autora.

Na tela seguinte, ilustrada na Figura 3.11, você deve selecionar os drivers, um de cada vez, para que a instalação dos mesmos seja iniciada. Você pode mudar o diretório selecionado para a instalação dos drivers.

Figura 3.11 | Instalação dos drivers



Fonte: elaborada pela autora.

Ao final da instalação, clique em “Concluir” e o PostgreSQL está pronto para ser utilizado. Você pode iniciar esse SGBD através do caminho: “Iniciar/ Programas/ PostgreSQL 9.5”.

Para saber mais

A documentação completa do PostgreSQL pode ser encontrada no site oficial, disponível em: <www.postgresql.org> e <<http://www.postgresql.org/docs/>>. Acesso em: 17 ago. 2017.

Atividades de aprendizagem

- 1.** Um sistema de gerenciamento de banco de dados cliente servidor divide o processamento em dois sistemas: o cliente e o servidor. O cliente é geralmente um computador pessoal que executa a aplicação do BD. Nesse contexto, analise as seguintes afirmativas:
 - I) Na arquitetura cliente servidor existe uma divisão de processamento entre dois sistemas.

II) A maior desvantagem dos SGBDs cliente servidor é que eles exigem que os dados sejam armazenados em um único sistema.

III) A grande vantagem da arquitetura cliente servidor está relacionada a não existência de uma rede para conexão entre as máquinas.

Assinale a alternativa correta:

- a) Apenas a afirmativa I está correta.
- b) Apenas as afirmativas I e II estão corretas.
- c) Apenas as afirmativas I e III estão corretas.
- d) Apenas as afirmativas II e III estão corretas.
- e) As afirmativas I, II e III estão corretas.

2. O PostgreSQL é um SGBD relacional e popular que possui código aberto (opensource). Esse SGBD possui todos os recursos de um grande banco de dados e é capaz de administrar uma grande quantidade de informações. No processo de instalação desse SGBD é aconselhável instalar alguns drivers, por exemplo:

- a) Oracle.
- b) Java.
- c) JDBC.
- d) C++.
- e) SQL

Seção 2

PgAdmin: criação de um banco de dados e tabela

Introdução à seção

Nesta seção, você vai aprender através de exemplos práticos as etapas para a criação de um banco de dados utilizando a ferramenta pgAdmin, que é instalada automaticamente e em conjunto ao PostgreSQL. Serão criadas tabelas para esse novo banco de dados e inseridos vários campos nas mesmas, o que possibilitará que você exercente os conceitos aprendidos e se familiarize com a ferramenta gráfica disponibilizada pelo PostgreSQL.

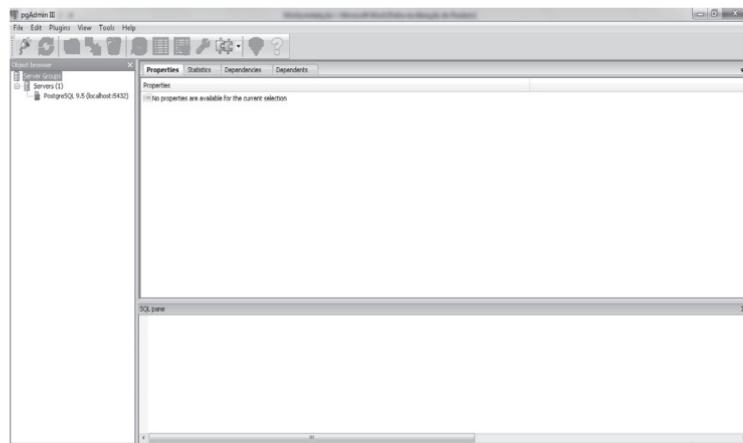
2.1 PgAdmin

Como já estudado na Unidade 1, um banco de dados relacional pode ser definido como um conjunto de tabelas que estão interligadas entre si. Existem duas maneiras para se criar tabelas no PostgreSQL: por meio de comandos SQL ou utilizando o pgAdmin III. Nossa foco é a criação de um banco de dados, tabelas e *constraints* utilizando a interface gráfica do pgAdmin III. O pgAdmin é uma ótima ferramenta para se administrar o PostgreSQL, pois permite a prática de todas as tarefas necessárias para a administração de um banco de dados.

O pgAdmin é instalado de maneira automática em conjunto com o PostgreSQL. A ferramenta pode ser encontrada no seguinte caminho: Iniciar > Programas > PostgreSQL > pgAdmin III.

Na abertura do pgAdmin tem-se a tela ilustrada na Figura 3.12.

Figura 3.12 | Tela principal do pgAdmin

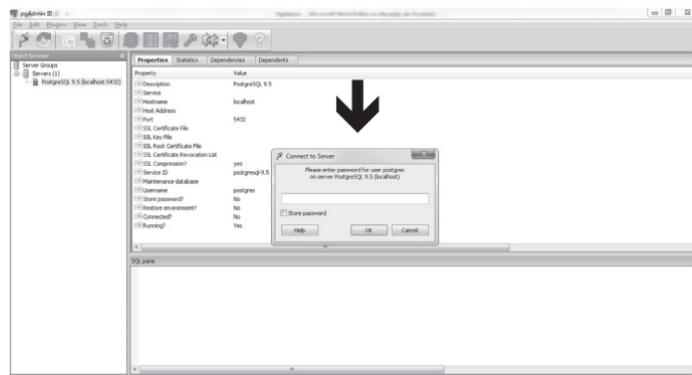


Fonte: elaborada pela autora.

2.1.1 Criando um banco de dados com o pgAdmin

À esquerda da tela principal do pgAdmin são encontradas as conexões com o servidor de BD. Nesse caso, tem-se apenas uma, porém outras conexões podem ser criadas através do menu "File" e "Add Server". Na instalação do PostgreSQL já foi configurado o acesso ao servidor local. Para acessar esse servidor, dê um clique duplo em "PostgreSQL Database Server". Ao clicar no servidor, uma janela será aberta, sendo necessária a digitação da senha definida no processo de instalação do SGBD, conforme ilustrado na Figura 3.13.

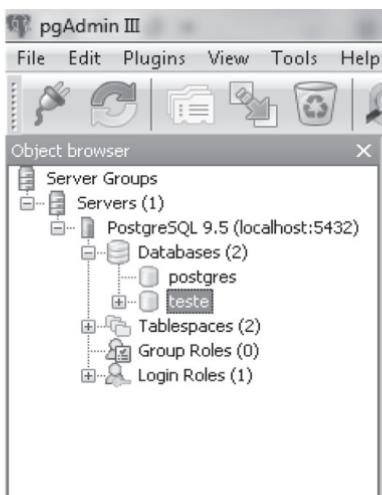
Figura 3.13 | Acesso ao pgAdmin



Fonte: elaborada pela autora.

Assim que a conexão com o servidor for estabelecida com sucesso, pode-se observar no lado esquerdo uma árvore de diretórios dos bancos de dados, ilustrada na Figura 3.14. Em “Databases” tem-se todos os bancos de dados configurados no servidor. Em “Group Rules” é definido o grupo de usuários do BD. Em “Login Rules” são definidas as regras de acesso dos usuários do BD. Clicando na opção “Databases” e em “+” encontramos o banco de dados “postgres”, que foi criado automaticamente no processo de instalação. Esse BD não será utilizado, porém não pode ser excluído, já que é utilizado para a administração do PostgreSQL.

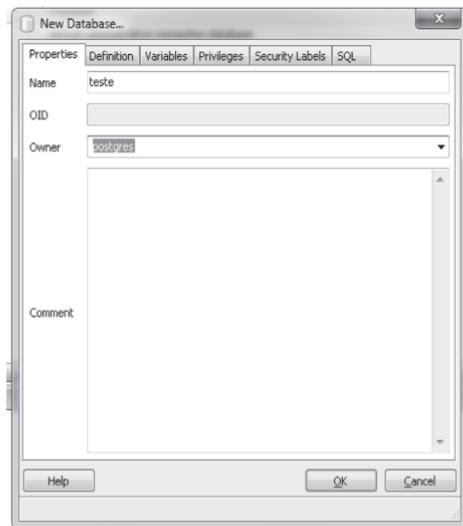
Figura 3.14 | Árvore de diretórios do pgAdmin



Fonte: elaborada pela autora.

Nesta etapa, começaremos com a criação de um novo banco de dados. Para isso, clique com o botão direito do mouse em “Databases”, escolhendo a opção “New Database”. Uma janela com alguns campos, conforme ilustrado na Figura 3.15, será aberta. No campo “Name” deve ser definido o nome do banco de dados, neste caso foi escolhido o nome “teste”. O campo “Owner” identifica o usuário proprietário do banco de dados, que no caso é o usuário “postgres”.

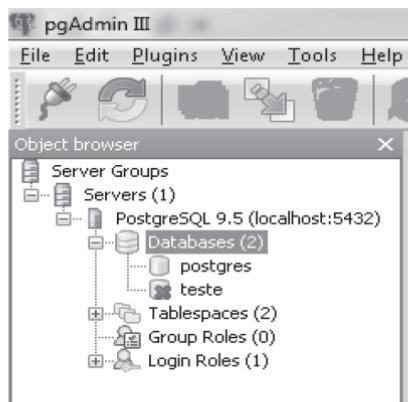
Figura 3.15 | Criação de novo BD



Fonte: elaborada pela autora.

Após configurados os campos e clicado em "OK" podemos visualizar o banco de dados criado na árvore de diretórios, conforme Figura 3.16.

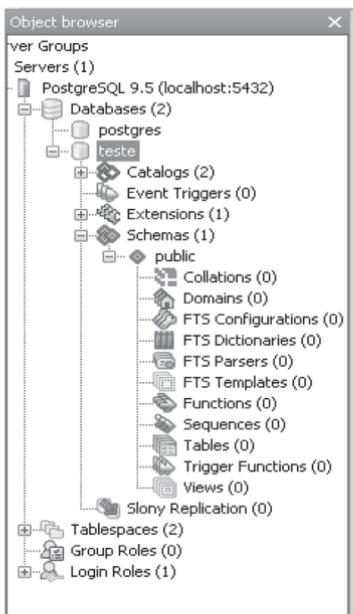
Figura 3.16 | Banco de dados criado



Fonte: elaborada pela autora.

Para acessar o banco criado, clique no nome do banco "teste", depois no "+", em seguida em "Schemas" e "public", de acordo com a Figura 3.17 que ilustra uma visão geral do BD criado.

Figura 3.17 | Atributos do BD criado



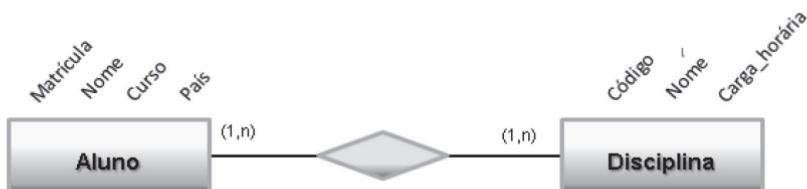
Fonte: elaborada pela autora.

A etapa para a criação de um novo banco de dados foi concluída, o próximo passo é a inserção de tabelas neste BD.

2.1.2 Criando tabelas com o pgAdmin

Serão criadas duas tabelas no pgAdmin para ilustrar o funcionamento dessa ferramenta. As duas tabelas podem ser observadas no diagrama entidade relacionamento da Figura 3.18.

Figura 3.18 | Diagrama entidade relacionamento

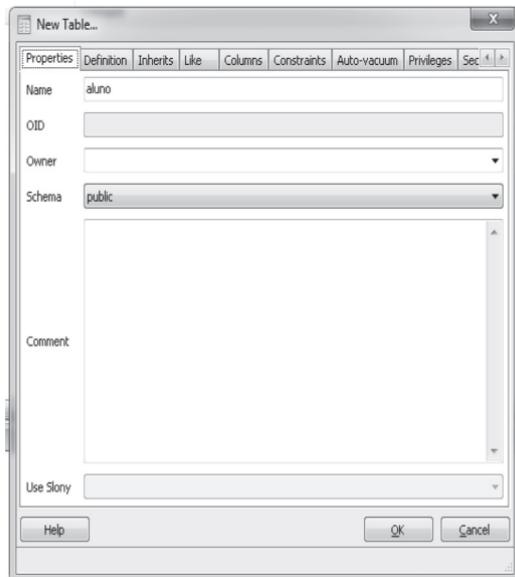


Fonte: elaborada pela autora.

Para a criação de tabelas com o pgAdmin, clique com o botão direito do mouse na opção “Tables” e selecione “New Table”. A janela aberta contém alguns campos que devem ser informados. No campo “Name” deve-se digitar o nome da tabela. A opção “Comment” não é de digitação obrigatória, pois se trata apenas de um comentário sobre a tabela.

Como exemplo, vamos criar uma tabela chamada “aluno”, conforme Figura 3.19.

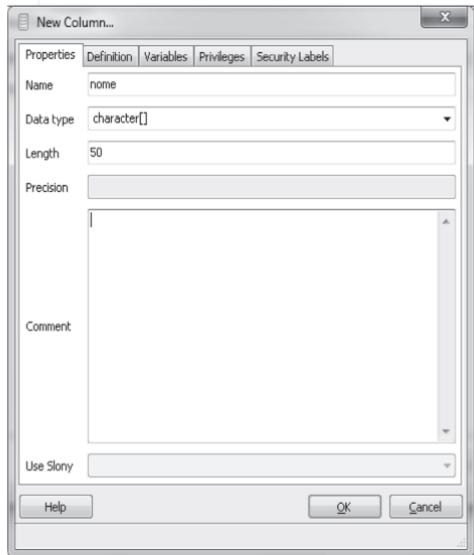
Figura 3.19 | Criação de uma tabela



Fonte: elaborada pela autora.

Após a definição do nome da tabela, clique na aba “Columns” para a adição das colunas ou campos da tabela, que possuem várias opções, como: nome, tipo, precisão e tamanho do novo campo. A opção “Name” determina o nome do campo de dados da tabela. A opção “Data Type” define qual tipo de dados que será aceito para esse campo. Os tipos de dados mais utilizados são: caracteres, data/hora, lógicos e numéricos. Na opção “Length” você pode definir o tamanho máximo suportado pelo campo. Um exemplo de inserção do campo “nome” na tabela “aluno” é ilustrada na Figura 3.20.

Figura 3.20 | Inserção de campos



Fonte: elaborada pela autora.



Para saber mais

Existem vários tipos de dados que podem ser utilizados no pgAdmin. Confira no link a seguir alguns tipos mais importantes.

Disponível em: <<http://www.devmedia.com.br/tipos-de-dados-no-postgresql-e-sql-server/23362>>. Acesso em: 17 ago. 2017.

Na aba “Definition” você encontra a opção de “Default Value”, na qual pode ser configurado um valor padrão para o campo, que será determinado quando nenhum valor for digitado no momento da inserção de um registro. Nessa aba também encontramos a opção “Not NULL”, que indica que o campo não pode ser nulo, sendo obrigatório o seu preenchimento.

Para o nosso exemplo, vamos criar quatro campos: matrícula, nome, curso e país, conforme ilustrado na Figura 3.21.

Figura 3.21 | Campos de uma tabela

Column name	Definition	Inherit...
matrícula	integer NOT NULL	
nome	character varying(50)	
curso	character varying(30)	
país	character varying(20)	

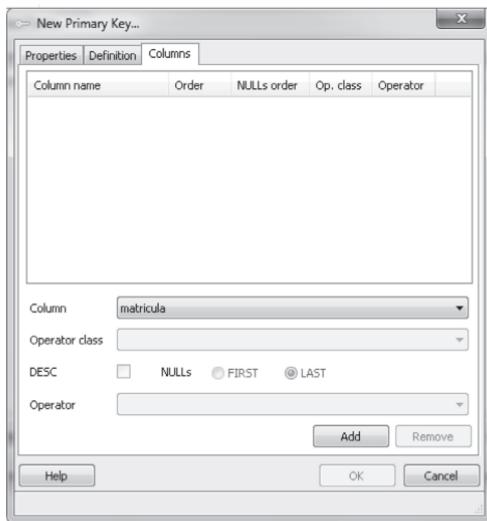
Buttons at the bottom: Change, Add, Remove, Help, OK, Cancel.

Fonte: elaborada pela autora.

Na aba “Constraint” são definidas as regras para uma determinada coluna. As regras mais utilizadas são: *Primary Key* (chave primária) e *Foreign Key* (chave estrangeira). Quando uma *constraint* é utilizada para definir uma chave primária é criada uma regra que não permite que os dados se repitam em uma mesma tabela. Geralmente, os campos definidos como chave primária são usados como identificadores únicos e servem para a ligação entre as tabelas no banco. Já a criação de uma chave estrangeira em uma coluna significa que ela está relacionada com uma coluna específica de outra tabela do BD. Outra *constraint* bastante utilizada é a “Check Constraint”, onde é possível definir uma regra para aceitação do campo. Por exemplo: “ano > 2017”, somente serão aceitos valores que sejam maiores que 2017.

Para a definição de uma chave primária, seleciona “Primary Key” e clique em “Add”. No campo “Name”, digite o nome escolhido para a chave. Por convenção, todos os nomes das chaves primárias possuem o prefixo “pk_”. Escolha como a chave primária da tabela o campo “matrícula”. Então a chave primária deverá se chamar “pk_matrícula”, conforme Figura 3.22.

Figura 3.22 | Definição da chave primária



Fonte: elaborada pela autora.

Na aba “SQL” você pode encontrar o código SQL gerado pelo pgAdmin com a criação da tabela “aluno”, ilustrado na Figura 3.23.

Figura 3.23 | Script SQL da tabela criada

The screenshot shows the 'New Table...' dialog box with the 'SQL' tab selected. The 'Read only' checkbox is checked. The SQL code generated for creating the 'aluno' table is displayed:

```
CREATE TABLE public.aluno
(
    matricula integer NOT NULL,
    nome character(50)[],
    curso character(30)[],
    pais character(20)[] DEFAULT Brasil,
    PRIMARY KEY (matricula)
)
WITH (
    OIDS = FALSE
);
```

Fonte: elaborada pela autora.



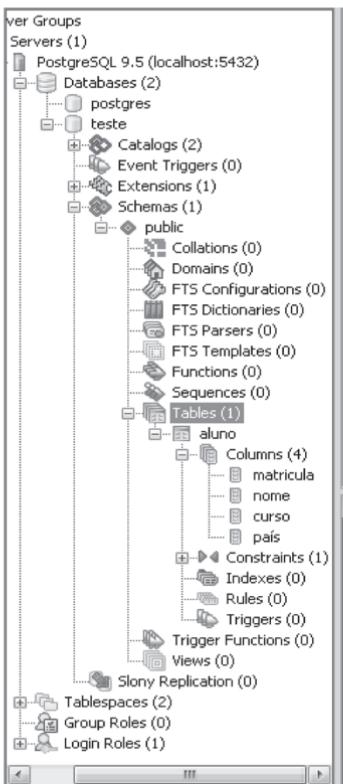
Para saber mais

Existem diversos comandos para manipulação de bancos de dados e tabelas na linguagem SQL. No link a seguir você confere alguns comandos mais importantes dessa linguagem sendo utilizada no pgAdmin.

Disponível em: <<http://www.devmedia.com.br/principais-comandos-sql/23366>>. Acesso em: 17 ago. 2017.

Nós já temos nossa primeira tabela criada. Você também pode conferir essa tabela no diretório do pgAdmin, conforme Figura 3.24.

Figura 3.24 | Tabela “aluno”



Fonte: elaborada pela autora.

Para visualizar, editar, inserir ou excluir os dados da tabela, selecione a tabela “aluno” e clique no botão “View the data in the selected object”, conforme demonstrado na Figura 3.25.

Figura 3.25 | Editando uma tabela

The screenshot shows the pgAdmin III interface. On the left, the Object browser displays a tree structure of database objects. In the center, a properties dialog box is open for the 'aluno' table. An arrow points to the 'View the data in the selected object.' button at the top of the dialog. The 'Properties' tab is selected, showing the following table:

Property	Value
Name	aluno
OID	16409
Owner	postgres
Tablespace	pg_default
ACL	
Of type	
Primary key	matricula
Rows (estimated)	0
Fill factor	
Rows (counted)	0
Inherits tables	No
Inherited tables count	0
Unlogged?	No
Has OIDs?	No
System table?	No
Comment	

At the bottom of the dialog, it says "SQL pane -- Table: public.aluno".

Fonte: elaborada pela autora.

Com o clique no botão indicado anteriormente, uma nova tela irá aparecer, na qual é possível inserir, alterar ou excluir dados da tabela. Para isso, clique na primeira coluna e digite os dados desejados, conforme Figura 3.26. Ao teclar “Enter” os dados digitados serão salvos.

Figura 3.26 | Inserção de dados em uma tabela

The screenshot shows the PostgreSQL Edit Data interface. At the top, the title bar reads "Edit Data - PostgreSQL 9.5 (localhost:5432) - teste - public.aluno". Below the title bar is a menu bar with File, Edit, View, Tools, Help, and a dropdown set to "No limit". The main area contains a table with four columns: matrícula [PK] integer, nome character varying(50), curso character varying(30), and país character varying(20). There are two rows of data: row 1 with matrícula 123, nome Andresa Silva, curso Computação, and país Brasil; and row 2 with matrícula 124, nome Marcelo Souza, curso Biologia, and país Brasil. A third row is present but empty. Below the table is a "Scratch pad" window which is currently empty. At the bottom left, it says "2 rows."

matrícula [PK] integer	nome character varying(50)	curso character varying(30)	país character varying(20)
1	123	Andresa Silva	Computação
2	124	Marcelo Souza	Biologia
*			

Fonte: elaborada pela autora.

Para verificar a funcionalidade da chave primária na tabela, tente inserir os dados de um aluno sem ter digitado o número de sua matrícula. Conforme Figura 3.27, é mostrada uma mensagem de erro, pois o campo “matrícula” necessita de um valor. Da mesma maneira, é mostrada uma mensagem de erro no caso de se tentar inserir um número de matrícula já existente, pois esse campo deve ser único.

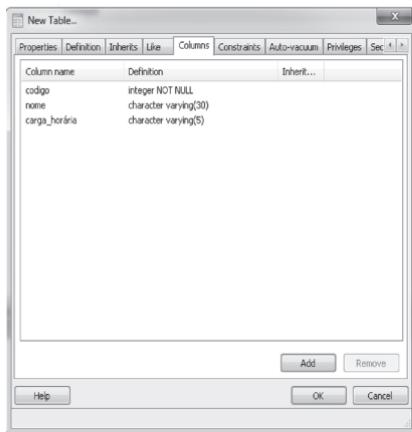
Figura 3.27 | Funcionalidade da chave primária

The screenshot shows the pgAdmin III interface. In the foreground, a modal dialog box titled "pgAdmin III" displays an error message: "An error has occurred: ERROR: null value in column "matrícula" violates not-null constraint DETAIL: Failing row contains (null, Alexandre Vieira, Computação, Brasil.)." At the bottom right of the dialog box is an "OK" button. Behind the dialog box, the PostgreSQL Edit Data interface is visible, showing the same table structure and data as in Figure 3.26. The table now has three rows: row 1 with matrícula 123, nome Andresa Silva, curso Computação, and país Brasil; row 2 with matrícula 124, nome Marcelo Souza, curso Biologia, and país Brasil; and a new row 3 with matrícula null, nome Alexandre Vieira, curso Computação, and país Brasil. The "Scratch pad" window is also visible.

matrícula [PK] integer	nome character varying(50)	curso character varying(30)	país character varying(20)
1	123	Andresa Silva	Computação
2	124	Marcelo Souza	Biologia
*	Alexandre Vieira	Computação	Brasil

Fonte: elaborada pela autora.

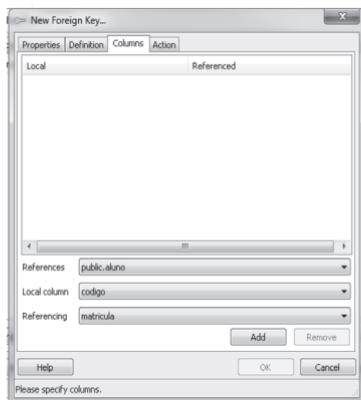
Figura 3.28 | Campos da tabela disciplina



Fonte: elaborada pela autora.

O próximo passo é a criação de uma chave estrangeira, que irá relacionar a tabela “disciplina” e uma coluna “matrícula” da tabela “aluno”. Para a criação dessa chave clique em “Constraint”, selecione “Foreign Key” e clique em “Add”. Defina o nome da chave estrangeira, que por padrão deve ser iniciado com o prefixo “fk”. A tela que será aberta está ilustrada na Figura 3.29. No campo “References” selecione a tabela “aluno”, no campo “Local column” selecione “código” da tabela “disciplina” e no campo “Referencing” selecione “matricula”, que é o campo que será referenciado na tabela “aluno”.

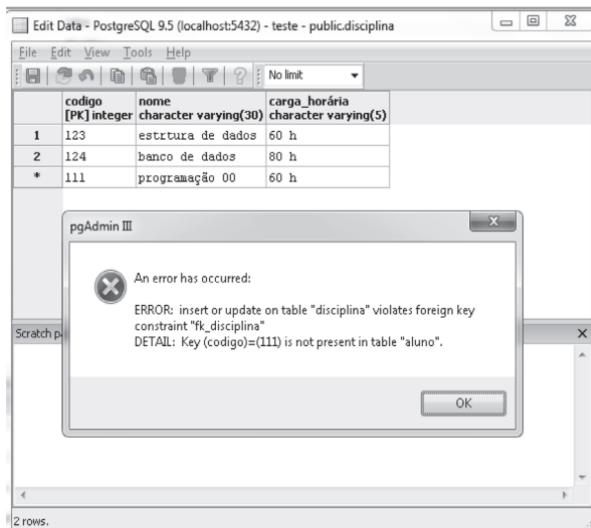
Figura 3.29 | Criação de uma chave estrangeira



Fonte: elaborada pela autora.

Nós criamos duas tabelas no banco de dados. Experimente inserir novos dados na tabela “disciplina”. Agora, tente inserir um novo dado, digitando no campo “código” um valor não existente de matrícula na tabela “aluno”. Nesse caso, vai aparecer uma mensagem de erro, pois ocorre uma tentativa de inserção de um registro que não tem relação com a tabela “aluno”, conforme Figura 3.30.

Figura 3.30 | Inserção de valores na tabela “disciplina”



Fonte: elaborada pela autora.

Tente também informar um código que já foi cadastrado na tabela “disciplina”. Nesse caso, a inserção é aceita, pois é possível que um aluno esteja matriculado em mais de uma disciplina.

Os passos para a criação de uma tabela podem ser resumidos em:

1. Criar o banco de dados: Após a conexão com o banco, clicar com o botão direito em ‘Databases’ e escolher a opção ‘New Database’.
2. Criar as tabelas: Clicar com o botão direito na opção “Tables” e escolher “New Table”.
3. Criar as colunas (Campos): definir os nomes, tipos e outros aspectos dos campos da tabela.
4. Adicionar os *Constraints*: definir as chaves primárias e estrangeiras.



Em que sentido a definição de uma chave estrangeira é importante para um banco de dados?

Atividades de aprendizagem

1. Um banco de dados relacional pode ser definido como um conjunto de tabelas que estão interligadas entre si. Existem duas maneiras para se criar tabelas no PostgreSQL: por meio de comandos SQL ou utilizando o pgAdmin III. Analise as seguintes afirmativas sobre o pgAdmin:

- I) O pgAdmin é uma ótima ferramenta para se administrar o PostgreSQL.
- II) O pgAdmin permite a prática de todas as tarefas necessárias para a administração de um banco de dados.
- III) O pgAdmin é instalado de maneira automática em conjunto com o PostgreSQL.

Assinale a alternativa correta.

- a) Apenas a afirmativa I está correta.
- b) Apenas as afirmativas I e II estão corretas.
- c) Apenas as afirmativas I e III estão corretas.
- d) Apenas as afirmativas II e III estão corretas.
- e) As afirmativas I, II e III estão corretas.

2. Após a definição do nome da tabela, a aba “Columns” é utilizada para a adição das colunas ou campos da tabela. Esses campos possuem várias opções e funcionalidade, como: nome, tipo, precisão e tamanho do novo campo. Assinale a alternativa correta em relação às funcionalidades de cada campo.

- a) A opção “Name” determina o nome do campo de dados da tabela.
- b) A opção “Data Type” define o tamanho dos dados que serão aceitos para esse campo.
- c) A opção “Length” define qual tipo de dado é aceito para esse campo.
- d) A opção “Name” determina o nome do banco de dados que a tabela pertence.
- e) A opção “Name” determina o nome do usuário do banco de dados.

Fique ligado

Nesta unidade, você começou seu estudo aprendendo sobre os principais conceitos relacionados aos SGBDs, você aprendeu sobre os sistemas de gerenciamento de banco de dados (SGBDs). Dentre os vários SGBDs disponíveis atualmente, foi apresentado o PostgreSQL. Esse SBGD é relacional e suporta grande parte do padrão SQL, oferecendo características modernas, como: integridade transacional, chaves estrangeiras, controle de integridade etc. O PostgreSQL utiliza a arquitetura cliente servidor. Assim, esta unidade apresentou as características que envolvem esse tipo de arquitetura, assim como a maneira que o PostgreSQL utiliza esse recurso.

Esta unidade também apresentou um estudo prático envolvendo o PostgreSQL, mostrando um simples passo a passo para a instalação e configuração do SGBD.

Para finalizar, esta unidade apresentou todas as etapas para a criação de um novo banco de dados através de uma interface gráfica disponibilizada pelo PostgreSQL, chamada pgAdmin III. Nessa interface, foram criadas tabelas com campos, relacionamentos e chaves.

Para concluir o estudo da unidade

Existem vários tipos de banco de dados e eles estão presentes na nossa vida há muito tempo. Por exemplo, a lista telefônica pode ser considerada um banco de dados. Antigamente, as empresas armazenavam suas informações em arquivos físicos, porém o surgimento e a evolução dos computadores possibilitaram o armazenamento de dados de modo digital. Assim, os bancos de dados evoluíram e se tornaram imprescindíveis para muitos sistemas de informação. Atualmente, os Bancos de Dados são amplamente utilizados, representando uma parte fundamental de quase todas as empresas, como: bancos, universidades, indústrias etc.

Um Sistema de Gerenciamento de Banco de Dados (SGBD) pode ser entendido como um conjunto de softwares (serviços) que tem a função de gerenciar um banco de dados, permitindo o acesso aos dados de forma simples. Os SGBDs também permitem gerenciar o acesso aos dados para múltiplos usuários, definindo permissões de acesso para cada perfil de usuário. Atualmente, existem vários SGBDs

presentes no mercado, os principais são: PostgreSQL, MySQL, Oracle, entre outros.

Nesta unidade, foi apresentado um exemplo prático de criação de um banco de dados e tabelas utilizando o PostgreSQL. Porém, é fundamental que você pratique os conceitos estudados e principalmente tente exercitar e replicar os exemplos que foram apresentados, assim como criar novos bancos de dados utilizando as dicas e etapas apresentadas nesta unidade.

Atividades de aprendizagem da unidade

1. A arquitetura cliente servidor é muito utilizada no contexto das redes locais e em relação ao acesso à Internet nas empresas. Sobre esta arquitetura, analise as seguintes afirmativas.

- I) É uma arquitetura composta por, pelo menos, dois equipamentos interligados em rede.
- II) O cliente envia a solicitação ao servidor que executa o que foi solicitado ou procura a informação solicitada e retorna ao cliente.
- III) O servidor é sempre responsável por iniciar uma comunicação, sendo que o cliente trabalha de forma reativa, respondendo às requisições do servidor.

Assinale a alternativa correta:

- a) Apenas a afirmativa I está correta.
- b) Apenas as afirmativas I e II estão corretas.
- c) Apenas as afirmativas I e III estão corretas.
- d) Apenas as afirmativas II e III estão corretas.
- e) As afirmativas I, II e III estão corretas.

2. O banco de dados “postgres” é criado automaticamente no processo de instalação. Esse BD não é utilizado, mas não pode ser excluído, já que é utilizado para a administração do PostgreSQL. Para a criação de um novo banco de dados é necessário selecionar quais seguintes opções?

- a) Databases/ New Database
- b) Tables/ New Table
- c) Databases/ AddTable
- d) PostgreSQL/ New Table
- e) Table/ New Database

3. Um banco de dados relacional pode ser definido como um conjunto de tabelas que estão interligadas entre si. Existem duas maneiras de se criar tabelas no PostgreSQL: por meio de comandos SQL ou utilizando uma interface gráfica. Qual o nome dessa interface?

- a) JDBC.
- b) SQL.
- c) Database.
- d) pgAdmin.
- e) Constraint.

4. Após a criação de um novo banco de dados, o próximo passo é a inserção de tabelas neste BD. Em relação à criação de tabelas, analise as seguintes afirmativas:

- I) Para a criação de tabelas com o pgAdmin, clique com o botão direito do mouse na opção "Tables" e selecione "New Table".
 - II) No campo "Name" deve-se digitar o nome da tabela, mas não é obrigatório que toda a tabela possua um nome.
 - III) A opção "Comment" não é de digitação obrigatória, pois se trata apenas de um comentário sobre a tabela.
- Assinale a alternativa correta.
- a) Apenas a afirmativa I está correta.
 - b) Apenas as afirmativas I e II estão corretas.
 - c) Apenas as afirmativas I e III estão corretas.
 - d) Apenas as afirmativas II e III estão corretas.
 - e) As afirmativas I, II e III estão corretas.

5. Na inserção de campos em uma tabela, tem-se a opção "Constraint" onde são definidas as regras para uma determinada coluna. As regras mais utilizadas são: Primary Key e Foreign Key. Qual a função da regra "Check Constraint"?

- a) Definir uma chave primária.
- b) Definir uma chave estrangeira.
- c) Definir um campo não nulo.
- d) Definir uma regra para aceitação do campo.
- e) Definir uma tabela sem campos.

Referências

- DATE, C.J. **Introdução a Sistemas de Bancos de Dados**. 4. ed. São Paulo: Campus, 1991.
- KORTH, H. F.; SILBERSCHATZ, A. **Sistemas de Bancos de Dados**. 2. ed. São Paulo: Makron Books, 1994.
- TANENBAUM, Andrew. S. **Redes de Computadores**. São Paulo: Campus, 2003.

Banco de dados não convencionais

Nathalia dos Santos Silva

Objetivos de aprendizagem

Nesta unidade, você entenderá como os Bancos de Dados precisam se manter evoluídos conforme as demandas encontradas com o avanço da tecnologia.

Você será apresentado aos tipos de dados atuais, conhecidos como não convencionais, e como estes dados são tratados pelos bancos de dados.

Você conhecerá as principais ferramentas de análise de dados e como elas podem auxiliar diversas áreas de um processo de negócio.

Seção 1 | Dados Avançados

Na Seção 1, estudaremos os dados que surgiram nos últimos anos e que, por serem diferenciados, precisam de suporte adequado para serem armazenados. Eles podem ser dados temporais, de mídia e espaciais.

Seção 2 | OLAP e Data Warehouse

Na Seção 2, veremos os tratamentos avançados de dados em um banco por meio dos conceitos de OLAP e *Data Warehouse*, e quais são as características principais de cada um.

Introdução à unidade

Esta unidade traz os conceitos sobre os bancos de dados não convencionais, passando por dados de tipos e análises, ambos explorados por serem uma extensão dos modelos convencionais.

As tecnologias atuais mudaram a forma das pessoas comprar, estudar, trabalhar, se distrair e se comunicar. Consequentemente, isto propiciou uma mudança na maneira com que os desenvolvedores armazenam e disponibilizam os dados envolvidos nessas ações.

Se antes os dados se resumiam a números, tabelas, arquivos e ocorrências bem definidas e conhecidas, hoje temos dados provenientes de diversos sistemas, que carregam diferentes informações, com tamanhos e formatos variados.

Para lidar com esses tipos de dados é necessário uma infraestrutura física e lógica coerente, ou seja, precisamos projetar bancos de dados capazes de receber, tratar, armazenar e disponibilizar esses dados, mantendo rapidez, segurança e controle de acesso.

Além do mais, é muito interessante que as organizações possam obter inteligência a partir dos dados, principalmente se for possível considerar um grande volume de dados, com referência ao histórico da organização e oriundo de fontes diferentes. Uma análise sistematizada, considerando diferentes aspectos, é muito favorável às empresas em seus setores de marketing, controle de estoque, logística ou outros (HEUSER, 2009).

Os sistemas de apoio à decisão, também chamados de sistemas baseados em análise, fazem parte do conjunto de técnicas chamado de *Business Intelligence* – BI, inteligência de negócios, e dentre estes sistemas os mais comuns são o OLAP e o *Data mining*. Cada um com seu nível de abrangência buscam comportamentos diferenciados dos clientes através dos dados armazenados no banco (RAMARKRISHNAN; GEHRKE, 2011).

Esta unidade passará por todos esses aspectos para que você conheça as necessidades futuras mínimas nos projetos de bancos de dados.

Seção 1

Dados Avançados

Introdução à seção

Observe em seu dia a dia como são os dados que você produz, agora pense na quantidade de tipos de dados que existem e que os bancos de dados têm de estar preparados para recebê-los. É uma variedade imensa, você concorda? Você tem conversas e vídeos em seu celular, e em seu computador você tem projetos, códigos-fonte e histórico de versionamento (controle de versões) de programas.

O computador de um engenheiro civil tem dados de georreferenciamento e projetos do tipo CAD¹, e assim temos uma variedade de dados altamente acessados atualmente. Nesta seção, veremos como o banco de dados lida com esses dados.

1.1 Dados temporais

Ao observarmos a maioria dos bancos de dados existentes, percebemos que os dados são armazenados conforme um retrato instantâneo do cenário que queremos modelar, ou seja, as ocorrências naquele intervalo de tempo. Muitas vezes, precisamos armazenar e obter informação sobre estados anteriores, e embora um esquema possa ter essas informações incorporadas manualmente, existe um banco de dados que trata especialmente esses casos (SILBERSCHATZ; KORTH; SUDARSHAN, 2012; ELMASRI; NAVATHE, 2011).

Uma fábrica, por exemplo, deve gravar em seu banco de dados os estados dos sensores ao longo do tempo, e não só aqueles medidos no momento atual. Da mesma forma, um consultório médico precisa armazenar informações que registrem o histórico de consultas, exames e estados de saúde do paciente.

Observe que os tempos em que há atualização de dados nos dois exemplos são distintos, e são diferentes ainda do tempo que cronometramos, assim, existem dois conceitos primordiais e serem retirados do contexto que queremos modelar: o tempo válido e o tempo de transação.

¹ Computer Aided Design

O **tempo da transação** é uma proporção do intervalo de tempo do mundo real (minutos, segundos) para o qual os fatos registrados nos dados do banco são atuais, isto é, pode ser o tempo desde a última atualização, ou o tempo em que determinada atualização anterior durou. Como é um tempo que varia cada vez que sofre mudança em seu valor, é calculado pelo próprio banco de dados, conforme os registros de atualização (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Já o **tempo válido** para um fato é o conjunto de intervalos de tempo durante o qual o fato é verdadeiro no cenário modelado, então é um dado que precisa ser fornecido ao sistema, pois tem relação com o mundo real (SILBERSCHATZ; KORTH; SUDARSHAN, 2012; ELMASRI; NAVATHE, 2011).

Para associarmos uma tupla ao seu tempo referido utilizamos uma relação temporal, contendo o tempo válido ou o tempo de transação, e caso seja necessário armazenar os dois tempos temos uma relação bitemporal.

Uma relação temporal é ilustrada na Tabela 4.1, o campo tempo válido é representado pelos atributos *Desde* e *Até* de maneira conjunta:

Tabela 4.1 | Relação temporal que descreve quem são os capitães dos times

Nome	Registro	Naturalidade	Posição	Desde	Até
José Carlos	1234567	Campinas	Atacante	02/08/2003	10/05/2005
João Silva	3455432	São Paulo	Volante	11/05/2005	17/12/2008
Bruno César	9807865	Salvador	Centroavante	18/12/2008	29/04/2011
Caio Souza	4590235	Florianópolis	Volante	30/04/2011	04/11/2014
Alex Pereira	5382743	Londrina	Goleiro	05/11/2014	atual

Fonte: elaborada pela autora.

Outro aspecto a ser analisado é o fuso-horário, quando os dados são na magnitude de horas, minutos e segundos.

Além disso, é importante falarmos sobre a consulta, esta pode ser instantânea, com atributos atuais que descrevem o cenário dos dados naquele momento, e ela também pode ser temporal, ou seja, envolve os atributos de tempo.

1.2 Dados espaciais

Outra classe de dados que teve um aumento substancial em sua

utilização e em sua demanda por melhoria no acesso é a de dados espaciais. Eles são utilizados para planejamento de cidades, análises agrícolas, ambientais, entre outros, e ainda estão sendo mais requeridos desde que o GPS se tornou popular em celulares e automóveis.

Os sistemas que gerenciam dados geográficos e aplicações relacionadas são conhecidos como sistemas de informações geográficas, do inglês *Geographical Information Systems* – GIS, em geral eles armazenam objetos que são descritos pelas suas características espaciais, e que possuem relacionamentos espaciais entre eles (ELMASRI; NAVATHE, 2011).

Dados espaciais variam em escala de espaço, podendo ser dados geográficos e incluir informações do mundo todo em diversos níveis de precisão e granularidade, ou dados de projetos de construções e urbanização, ou ainda podendo ser projetos de circuitos de computador, este último em escala microscópica.

Durante muito tempo estes dados eram tratados como arquivos em um sistema de arquivos, porém à medida que a complexidade, o volume dos dados e o número de usuários cresceram, as técnicas existentes se tornaram insuficientes para garantir atualizações, durabilidade e controle de concorrência, e fez-se necessário bancos de dados com características especiais para atender a essas exigências (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

A próxima geração terá uma densa rede de sensores, que deverá cobrir a terra completamente, com alta resolução e fidelidade por um longo período de tempo. O volume de dados necessário para entender e modelar essas interações é tão grande (TB, terabytes e crescendo continuamente) que não há solução pronta que permita aos cientistas gerir e analisar facilmente esses dados (LEHNING et al., 2011).

Os dados espaciais, como já descritos, em geral se dividem entre dados geográficos e dados de projeto auxiliado por computador, o CAD como conhecemos. Esses últimos armazenam informações sobre os diversos aspectos das construções, como edifícios, automóveis, motores e outros, e as informações costumam ser do ponto de vista estrutural, elétrico, mecânico, de design, por isso uma indexação comum não é suficiente para mapear e organizar esses dados.

Além destes exemplos temos também os projetos de circuitos integrados e dispositivos eletrônicos, que possuem ainda outros níveis de detalhes. O que há de comum entre todos esses modelos

É que suas informações gráficas são armazenadas como polígonos e agrupamento de pontos, por suas coordenadas cartesianas.

Foi a partir da inviabilidade de carregar um grande volume de dados da memória ao abrir um arquivo de uma aeronave, por exemplo, que bancos de dados orientado a objetos foram desenvolvidos (SILBERSCHATZ; KORTH; SUDARSHAN, 2012). Além disso, os bancos são diferenciados pelas restrições e verificações que precisam fazer nas informações, por exemplo, que não poderiam existir duas trilhas de circuito em um mesmo espaço. A isso isso damos o nome de restrições de integridade espacial.

Paralelo ao avanço desses dados tivemos uma crescente demanda nos dados geográficos, com informações de mapas rodoviários, de levantamento topográfico, ou com delimitações de fronteiras políticas.

São armazenados os dados espaciais dos mapas e, adicionalmente, os projetos de banco destes dados precisam preparar a infraestrutura para receber dados complementares, como índices demográficos, de produção agrícola, de bacias hidrográficas e diversos outros dados que são solicitados posteriormente.

Para que esse banco de dados tenha as funcionalidades desejáveis ele conta com alguns operadores, sendo estes responsáveis por gerenciar informações geométricas relevantes sobre os dados. Eles são: operadores topológicos, operadores projetivos e operadores métricos e ainda operadores espaciais dinâmicos. Podem ser entendidos como atributos complexos e específicos, e possuem operações de captura e retorno de informações (ELMASRI; NAVATHE, 2011).

Quanto às consultas, estas são divididas em níveis de precisão e abrangência de localizações, como consultas de proximidade, consultas por regiões específicas ou intersecção de dados de interesse que resultem em regiões específicas, até então não buscadas.

Por exemplo, podemos, através do celular, acessar e fazer uma busca sobre quais caixas eletrônicos há próximo àquele local, podendo refinar a pesquisa por um raio de distância máxima, pelo que você conhece se SQL percebe que esta não é uma consulta tradicional, já que precisam ser mapeadas e cruzadas informações adicionais.

A estrutura de dados para estes dados são árvores k-d, árvores quadráticas (quadtrees) ou árvores R, que são evoluções das árvores de busca tradicionais.



Para saber mais

Dados Geográficos, como os tratados pelo INPE no Brasil, demandam algumas características diferenciadas no seu desenvolvimento e, com o intuito de disseminar essa informação, o próprio INPE possui em sua página algumas boas referências sobre os aspectos dessa modelagem e infraestrutura. Acesse: <<http://www.dpi.inpe.br/livros/bdados/capitulos.html>>. Acesso em: 25 jul. 2017.

1.3 Dados de multimídia

Outro tipo de banco de dados que lidamos diariamente e, possivelmente, o carregamos em nosso bolso, é o banco de dados de multimídia, responsável por armazenar dados como fotos, vídeos e áudios. Como você já deve imaginar, esses bancos de dados também surgiram da necessidade de manter esses dados por parte dos usuários, bem como as plataformas de redes sociais, ou mesmo sistemas de segurança e monitoramento.

Além dos dados multimídia em si, faz-se necessário armazenar a descrição destes dados, como os atributos de sua criação e autoria. Uma alternativa encontrada é armazenar estas descrições em um banco de dados e registrar os arquivos em que os objetos de multimídia estão armazenados.

Dentre os principais formatos que encontramos nos dados multimídia são JPEG, PNG, GIF, BMP, FLAC, WAV, MP3, MP4, AVI, WMA e MKV, você já deve ter tido contato com a maioria deles, e em geral o que os diferencia é a compactação que cada formato consegue atingir. Este aspecto é importante ser analisado, pois está diretamente relacionado ao tamanho dos dados armazenados, e dados multimídia costumam necessitar de grandes quantidades de bytes para serem alocados.

Pensando nas questões que envolvem a disponibilização de dados de áudio, como existem atualmente diversos aplicativos em rede para acesso a músicas, além dos aspectos legais os projetistas de bancos de dados devem estar atentos às questões técnicas, como a velocidade de disponibilização dos dados. Se a velocidade a velocidade for muito baixa a música vai apresentar falhas e causar a impressão de “cortes”, ou ainda, se a velocidade for acima da capacidade do sistema pode estourar os *buffers*.

Atualmente, buscadores comerciais implementaram e disponibilizaram a busca por imagens, isto foi possível devido aos avanços em bancos de dados de multimídias, inclusive utilizando o que chamamos de recuperação baseada na semelhança (SILBERSCHATZ; KORTH; SUDARSHAN, 2012). Este mesmo princípio é utilizado em aplicativos que conseguem retornar o artista e o título da música a partir da transmissão de trechos de alguns segundos de uma música tocando no ambiente.

Podemos também pensar em buscar todos os clipes musicais em que aparecem cenas de guerra, então nossa pesquisa se baseará em atividades contidas no clipe, este tipo de consulta é chamado de recuperação baseada em conteúdo (ELMASRI; NAVATHE, 2011). Isso é possível com a identificação do conteúdo, que pode ser uma análise automática ou manual.

As análises das fontes de multimídia podem ser realizadas de maneira automática. Por exemplo, exemplo, a cor é uma característica de diferencial em imagens, além do mais, faz-se também o cálculo do histograma das imagens para armazenar como um atributo dela. A textura também pode ser identificada na imagem, por exemplo, uma lã, um piso ou uma grama, e estas informações também são dados armazenados de maneira conjunta com as imagens. Além da cor, consideramos também os segmentos que delimitam a borda do objeto.

A acessibilidade conseguiu avanços consideráveis quando os comandos por voz se tornaram mais populares, é necessário que bancos de dados que tratem voz em diferentes idiomas consigam responder de forma coerente o usuário. Mesmo em sistemas que não necessariamente são voltados para acessibilidade, estão embutindo a opção do comando por voz para melhorar a experiência e promover conforto ao usuário.

Ainda discutindo os desafios para os desenvolvedores de banco de dados multimídia, temos os celulares e notebooks, cada vez mais utilizados para as diversas atividades profissionais e de entretenimento e, consequentemente, gerando dados diversos. Como estes equipamentos possuem tamanho e energia limitados, manter desempenho em banco de dados com estas características são assuntos de pesquisa e constante melhora.



Questão para reflexão

Quando falamos em melhorar os aspectos dos bancos de dados estamos adicionando uma maior complexidade ao desenvolvimento do projeto, mas e se o projetista do banco de dados não tiver tempo hábil ou equipe suficiente para implementar todos esses avanços? Será que esse tipo de serviço já pode ser adquirido em nuvem?

Atividades de aprendizagem

1. Banco de dados espaciais são uma classe de banco de dados especiais, porque lida com informações _____, e estas informações requerem consultas diferenciadas e formas de busca e armazenamento também diferenciadas.

Assinale a alternativa que completa a lacuna:

- a) Geográficas.
- b) Planetárias.
- c) Em Nuvem.
- d) Espalhadas.
- e) Internacionais.

2. Bancos de dados temporais são aqueles que retratam o cenário ao longo de um tempo contínuo, e não somente através de informações de valor instantâneo.

Analise as alternativas e assinale a correta:

- a) Os bancos de dados temporais têm esse nome porque não trabalham com o tempo.
- b) O tempo é a relação neste banco de dados.
- c) O tempo é um atributo adicional e estratégico neste banco de dados.
- d) As consultas deste banco só retornam atributos temporais.
- e) As consultas deste banco não são otimizadas para dados temporais.

Seção 2

OLAP e *Data warehouse*

Introdução à seção

Você já observou que as empresas, principalmente as de comércio eletrônico, nos mostram aquilo que queremos comprar? Ou que elas nos contatam com anúncios direcionados? Isso é possível através da análise de dados sobre nosso consumo, passando por nosso histórico de compras, de buscas e pelo histórico de outros consumidores. Nesta seção, veremos o que significam os termos OLAP e *data warehouse* e como estão relacionados com as mais utilizadas ferramentas de apoio às empresas nas políticas de comunicação, precificação, distribuição e promoção.

2.1 OLAP

O termo OLAP é a abreviação para *On-line Analytical Processing*, ou seja, um Sistema de Processamento Analítico On-line, e podemos descrevê-lo como um sistema que permite ao analista ver diferentes resumos de dados de forma multidimensional, com uma resposta de no máximo alguns segundos (SILBERSCHATZ; KORTH; SUDARSHAN, 2012; HEUSER, 2009). Isto é, praticamente em tempo real o analista pode solicitar novos resumos e consultas acerca daqueles dados.

Um exemplo de OLAP em uma pequena quantidade de dados é o Excel, entretanto, para maiores quantidades de dados é necessário um banco de dados, e que o mesmo forneça suporte ao processamento e consulta, em tempo hábil, destes dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Ao utilizar uma ferramenta OLAP podemos obter como resultado informações inteligentes e interessantes daquele grande conjunto de dados, que seriam inviáveis ou infactíveis de obter com consultas simples. O OLAP então estratifica o conjunto de dados, buscando informações que servirão de base para tomadas de decisão na empresa (RAMARKRISHNAN; GEHRKE, 2011).

Tudo isso é realizado através de análises de múltiplas dimensões, os dados são vistos pelos usuários por perspectivas diferentes e, como cada atributo atua como uma dimensão, combiná-los de maneira multidimensional gera resultados variados.

Por exemplo, um supermercado quer saber quais são seus produtos mais vendidos em determinadas épocas do mês, e quer filtrá-los de acordo com os que retornam um maior lucro. Poderíamos ainda ter uma empresa de telefonia que quer mapear o uso de serviços de internet conforme a localidade no país, para então investir em infraestrutura adequada.

Considere a Tabela 4.2, onde vamos descrever um exemplo de como resulta um conjunto de dados após passar pelo OLAP.

Tabela 4.2 | Tabela com vendas de produtos

Nome_Item	Cor	Tamanho	Quantidade
Camiseta	Azul	P	2
Camiseta	Azul	M	6
Camiseta	Azul	G	8
Camiseta	Preta	P	12
Camiseta	Preta	M	15
Camiseta	Preta	G	23
Camiseta	Amarela	P	7
Camiseta	Amarela	M	8
Camiseta	Amarela	G	5
Camisa Polo	Vermelha	P	5
Camisa Polo	Vermelha	M	8
Camisa Polo	Vermelha	G	10
Camisa Polo	Preta	P	13
Camisa Polo	Preta	M	9
Camisa Polo	Preta	G	30
Camisa Polo	Branca	P	19
Camisa Polo	Branca	M	8
Camisa Polo	Branca	G	6

Fonte: adaptada de Silberschatz, Korth e Sudarshan (2012, p. 121).

Podemos estar interessados em resumir estes dados agrupando por Cor e Tamanho, como demonstrado na Tabela 4.3.

Tabela 4.3 | Tabulação cruzada de vendas por Cor e Tamanho

Cor	P	M	G
Azul	2	6	8
Preta	12	15	23
Amarela	7	8	5
Vermelha	5	8	10
Preta	13	9	30
Branca	19	8	6

Fonte: elaborada pela autora.

Esta tabulação cruzada também é chamada de tabela pivô, e ela reúne dados conforme o interesse do gestor da empresa, e pode ser variada quando necessário. Neste exemplo, o responsável pelo estoque consegue ter uma dimensão maior de quais as cores e tamanhos são mais vendidos, e então se planejar melhor para evitar peças paradas. Se esse gestor também estivesse interessado em mapear o desempenho das vendas conforme os modelos, isso também seria possível alterando a linha ou coluna da tabela. Ainda pensando na melhora do modelo, é possível criar estruturas multidimensionais, como **cubo de dados** (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Os mecanismos para consulta e organização dos dados podem ser realizados através dos comandos em SQL que você estudou na unidade anterior.



Para saber mais

Você viu que esse assunto é muito atual, não é mesmo? Sempre estamos falando de uma nova forma de utilizar a tecnologia a favor dos negócios, e relacionado a esse assunto ainda temos o conceito de Business Intelligence (BI), disponível em: <<https://endeavor.org.br/business-intelligence/>>. Acesso em: 16 ago. 2017.

2.2 Data Warehouse

O termo *data warehouse* diz respeito a um depósito de dados que reúne informações de diversas fontes em um mesmo local, com acesso, interface e gerenciamento unificado (SILBERSCHATZ; KORTH; SUDARSHAN, 2012). Em geral, esses depósitos armazenam

um longínquo histórico de dados, o que aumenta sua necessidade de robustez.

Para que o depósito consiga reunir os dados de diferentes fontes em um depósito unificado, ele deve ter um tratamento para entradas inválidas, técnicas eficientes de armazenamento e indexação de grandes quantidades de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Independentemente da sua origem, os dados passam por tratamentos em diversos níveis e análises estatísticas para fornecer bons resumos para tomada de decisão. Podemos afirmar também que o depósito de dados é uma estrutura estendida do OLAP, pois além das funções de consultas multidimensionais do OLAP, esta estrutura de armazenamento do depósito possibilita que sejam extraídas ainda mais informações dos dados, o que chamamos de mineração de dados, conforme sejam interessantes às organizações.

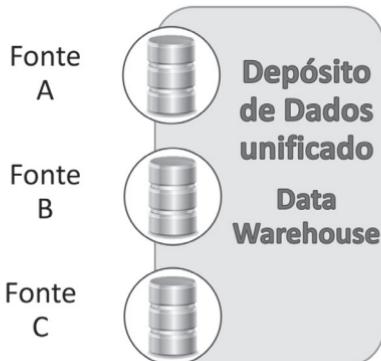
Embora os conceitos de armazenamento, tratamento e disponibilização em banco de dados sejam estudados há décadas, a internet e o consumo impulsionaram o avanço nas pesquisas e implementações nesta área devido à maioria das tecnologias depender de demanda comercial e popularização para investimento em melhoria.

As informações são a grande riqueza das empresas, e nos últimos anos muita informação tem sido armazenada, porém estas informações desconectadas nem sempre fornecem as respostas que as empresas precisam. Por isso os depósitos de dados também necessitam de mecanismos que permitam relacionar as ocorrências dos dados e delas extrair características valiosas, sob o ponto de vista estratégico.

Isso é realizado através da busca por padrões envolvidos nas ocorrências, e são estes padrões que identificam comportamentos de interesse para as empresas, sejam estes comportamentos de consumo, de desempenho ou outros. Basicamente, estamos interessados em uma ferramenta conceitual que mapeia um grande volume de dados, e disto consegue formular estatísticas, gráficos, inferências e conclusões que servem de base para as empresas tomarem decisões estratégicas (HEUSER, 2009).

Por exemplo, uma empresa observa o perfil da sua maioria de clientes em determinada linha de produtos, e isso pode servir como estratégia de marketing para lançamento de novos produtos ou mesmo fidelização dos atuais clientes.

Figura 4.1 | Formação de um depósito de dados



Fonte: elaborada pela autora.

A Figura 4.1 ilustra como é a formação do depósito de dados, ele pode ser diversificado internamente, porém para ser usado como ferramenta de análise ele é visto como um banco de dados comum, com estrutura tradicional de SGBD.

2.2.1 Data Mining

Boa parte do conceito do *data warehouse* se faz possível através de data mining, a **mineração dos dados**, que é o processo de analisar de forma semiautomática um grande volume de dados, a fim de extrair padrões úteis de ocorrência nos mesmos, como chamado por Silberschatz, Korth e Sudarshan (2012, p. 563), é uma “descoberta de conhecimento nos bancos de dados”.

O objetivo da mineração é encontrar **padrões** por meio do uso de algoritmos eficientes e estruturas de dados adequadas para pesquisa. É realizada uma varredura no banco de dados e os padrões encontrados, assim como os resultados, são mostrados ao analista e aos gestores, ou a quem interessar (RAMARKRISHNAN; GEHRKE, 2011).

A busca por padrões pode envolver associações, ou seja, um grupo de atributos que possui ocorrências associadas entre si. Por exemplo, equipamentos que são comprados juntos em uma loja de eletrônicos, ou acessórios que costumam ser adquiridos em conjunto.

As atividades envolvendo comércio estão à procura de padrões conhecidos como classificação, ou seja, elas têm interesse de classificar seus clientes como leais, com pressa, os que buscam descontos, os que têm preferência por marcas conhecidas, entre outros. Isso é

descoberto a partir da combinação de parâmetros, sendo que esses parâmetros podem ser os horários das compras, os itens comprados, as datas e a regularidade das compras, o principal grupo de produtos, entre outros.

Você deve estar se perguntando como é possível identificar o consumidor, a resposta é que cada vez os dados estão interligados, por exemplo, o consumidor pode ser reconhecido pelo cartão de crédito, ou mesmo pelo CPF quando ele pede nota fiscal identificada.

Tabela 4.4 | Exemplo de transações de um mercado

Nota fiscal	Data	Cartão de crédito	Itens comprados
124532	21/02/2016	4356 9964	Pao; carne; queijo
345653	22/05/2016	1256 3453	Refrigerante, chocolate, fralda
345438	07/12/2015	9304 9367	Arroz, detergente, tempero pronto
984902	05/06/2014	3467 7754	Agua, suco, bolacha
382859	19/05/2017	2348 4562	pilha

Fonte: adaptada de Elmasri e Navathe (2011, p. 701)

A Tabela 4.4 exemplifica como pode ser a relação entre os atributos de um banco de dados de um mercado, para posteriormente, em um *data warehouse* ser aplicado um data mining.

Outro padrão buscado nos bancos de dados é o de identificação, ele procura a existência de um item, ou evento. Elmasri e Navathe(2011) exemplifica estes padrões de identificação: podem ser identificados genes no ramo da genética a partir de certas sequências de DNA analisadas, ou ainda, em uma aplicação mais popular, a autenticação de pessoas autorizadas pela íris ou impressão digital.

Existem padrões que são buscados com a finalidade de atuar na previsão de acontecimentos, estes são definidos com base em um histórico de similaridade de ocorrências, e assim predizer certo tipo de comportamentos.

Figura 4.2 | Exemplo de fluxo de informação com o uso de mineração de dados



Fonte: elaborada pela autora.

A Figura 4.2 descreve uma visão conceitual, do ponto de vista de Sistemas de Gerenciamento de informação, sobre o uso da mineração de dados, sendo o termo “Aspectos de interesse” as informações filtradas pela mineração.

Um termo utilizado na área para descrever o processo de conhecimento de dados é o *Knowledge Discovery in Databases* – KDD, ou descoberta de conhecimento nos bancos de dados, que é composta por etapas de preparação de dados, transformação de dados, mineração de dados e exibição das informações encontradas (ELMASRI; NAVATHE, 2011).

Após a apresentação dos padrões encontrados como resultado, estes são analisados e avaliados por humanos, já que parte destes padrões podem derivar de outros, ou não são interessantes, daí o termo semiautomático.

Você, como um estudante da área de Computação, pode pensar: “Por que não colocam técnicas de inteligência artificial para aprender a analisar esses padrões?”. Parece uma alternativa razoável, entretanto, isso é um assunto para um próximo livro!

Árvores de decisão, ajustes de curvas com regressão, programação linear e não linear, programação dinâmica, inteira e binária, simulação de sistemas, entre outros são algumas das técnicas utilizadas na mineração de dados, além de algoritmos de busca, ordenação, seleção e outros comuns a bancos de dados (RAMAKRISHNAN; GEHRKE, 2011).



Questão para reflexão

Todas essas ferramentas em Banco de Dados são relativamente recentes, e os exemplos descritos são na área de serviços e comércios em geral, mas você já imaginou quais avanços poderíamos ter com tamanha capacidade de análise de dados em setores como saúde?

Atividades de aprendizagem

1. Imagine que sua empresa venda quatro produtos diferentes – porcas, pinos, arruelas e parafusos – nas regiões leste, oeste e central. Se você perguntasse: “Quantas arruelas foram vendidas durante o último trimestre?”, poderia facilmente achar a resposta consultando seu banco de dados de vendas. E se você quisesse saber quantas arruelas foram vendidas em cada região e também comparar os resultados reais com a projeção de vendas? (Adaptado de ENADE, 2012).

Perguntas dessa natureza são respondidas com o apoio de ferramentas de análise em banco de dados, como OLAP, *data warehouse* e *data mining*.

Sobre esse assunto, analise as informações a seguir e assinale a verdadeira:

- a) Se as vendas fossem de pontos distintos, como franqueados, on-line, lojas físicas e feiras de demonstração, a ferramenta ideal seria o OLAP.
- b) Ao passo que a pergunta pode ser respondida em algumas horas, com solicitação antecipada, uma infraestrutura cliente servidor dedicada seria a ideal.
- c) Uma tabela cruzada, característica de um OLAP consegue realizar essa consulta em tempo hábil.
- d) Um data mining consegue responder às perguntas somente por região, e não por tempo.
- e) Um *data warehouse* é a ferramenta ideal para esta análise, pois trata os dados como dimensões a serem combinadas.

2. As ferramentas que atuam em conjunto com os bancos de dados têm permitido que as empresas vejam de forma diferente o processo de armazenar informações. As informações mais do que nunca estão envolvidas nos processos de tomada de decisão, porém se não forem tratadas e disponibilizadas de forma adequada continuariam somente armazenando um grande volume de dados. Acerca deste tema, analise as asserções a seguir:

I – O *Data Warehouse* é responsável pelo armazenamento físico dos dados, através de uma plataforma única, mesmo que os dados sejam recebidos de

outros sistemas.

PORQUE

II – Uma análise para reconhecer padrões necessita de um conjunto mais completo possível sobre as informações pertinentes àquele processo, fomentando a etapa do *Data Mining*.

Assinale a alternativa correta acerca das asserções apresentadas:

- a) As duas asserções são corretas, e a segunda justifica a primeira.
- b) As duas asserções são corretas, porém a segunda não justifica a primeira.
- c) As duas asserções são incorretas.
- d) Somente a primeira asserção é correta.
- e) Somente a segunda asserção é correta.

Fique ligado

Quando falamos em Bancos de Dados não convencionais podemos pensar em dois grandes aspectos: operações não convencionais em bancos de dados e dados diferenciados, em ambos os casos estamos estendendo nossa visão sobre o que é Banco de Dados. Isso porque existe um desafio maior ao manipular dados mais complexos, que envolvem indexação, diferentes tamanhos e necessidades de busca que não se resolvem com consultas simples.

Para concluir o estudo da unidade

O assunto de Banco de Dados se torna mais atraente quando visualizamos suas aplicações no nosso dia a dia, e ele ainda se torna mais desafiador quando pensamos que a tecnologia não para de evoluir, e que certamente durante nossa vida profissional o conceito de banco de dados ainda tomará rumos que ainda não conhecemos.

Para atender essa demanda precisamos de profissionais que estejam sempre dispostos a aprender, e cientes que virão mais desafios por aí!

Atividades de aprendizagem da unidade

1. Acerca dos conceitos sobre tratamento avançado de dados e informações para tomada de decisão, complete a lacuna:

_____ tem o objetivo de identificar as relações existentes entre as informações do _____.

Assinale a alternativa que completa corretamente as lacunas:

- a) SQL; ERP.
- b) Banco de dados; Programa.
- c) Consultas; Atributo.
- d) SQL; SGBD.
- e) *Data mining; Data warehouse.*

2. José tem uma empresa de logística, a RECEBEX, ela conta com muitos colaboradores espalhados no Brasil inteiro, carregando e transportando mercadorias sobre rodas. Eles traçam rotas, registram horários de saída e chegada, gastos com pedágios, combustíveis e manutenção. A empresa RECEBEX gera uma quantidade enorme de informações por dia, mês, viagem, trajeto, meio de transporte e assim por diante, porém nem sempre José e seus engenheiros têm tempo de analisar todos os conjuntos de dados, perdendo vantagem para a concorrência por não fazer sempre o serviço de maneira mais otimizada. Os engenheiros que lideram a logística poderiam traçar estratégias de preço, de captação de novos clientes em rotas já existentes ou mesmo de melhor escolha de meios de transporte, mas isso só seria possível com uma análise avançada e multidimensional dos dados, e não consultas tradicionais. A ferramenta ideal para auxiliar os engenheiros da RECEBEX é:

- a) OLAP.
- b) Data center em nuvem.
- c) RAID 0.
- d) *Business Intelligence.*
- e) SQL server

3. Com base em seus conhecimentos em banco de dados e dados avançados, analise as asserções a seguir:

I – OLAP e *Data Mining* são exemplos de ferramentas usadas em banco de dados espaciais ou de multimídia.

PORQUE

II – Dados avançados requerem bancos de dados com funções específicas, pois necessitam de tratamento diferenciado em seus conteúdos.

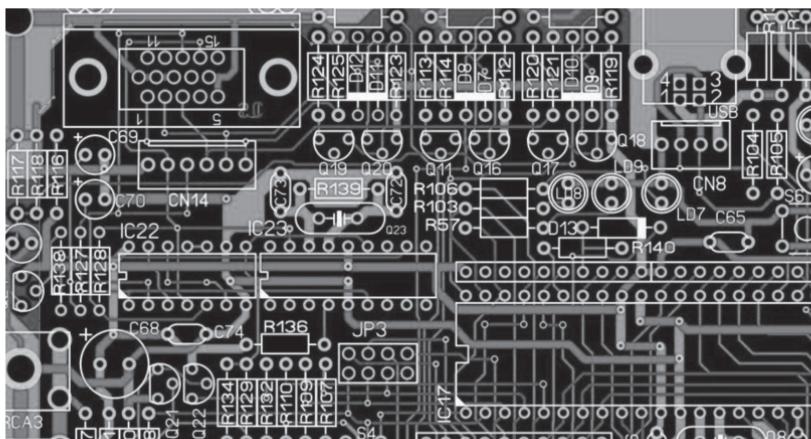
Assinale a alternativa correta:

- a) As duas asserções são corretas, e a segunda justifica a primeira.
- b) As duas asserções são corretas, porém a segunda não justifica a primeira.
- c) As duas asserções são incorretas.
- d) Somente a primeira asserção é correta.
- e) Somente a segunda asserção é correta.

4. Quando precisamos de análises mais aprimoradas de dados, isto é, precisamos descobrir informações sobre os dados que não estariam programadas em uma consulta, devemos recorrer a estruturas mais robustas. Inclusive, o termo Inteligência em Negócios é justamente isso, procurar informações não tão óbvias acerca do conjunto de dados, preferencialmente de maneira automática. Sobre este assunto, do ponto de vista de Banco de Dados, assinale a alternativa correta:

- a) OLAP e Data mining são ambas ferramentas de modelagem de dados e isso garante bons resultados para os gestores das empresas.
 - b) Banco de dados orientado a objetos é a solução para estes casos.
 - c) A análise estatística de um grande volume de dados é feita mesmo que os dados estejam armazenados de forma desacoplada, e resolve estas situações.
 - d) As ferramentas indicadas são OLAP e Oracle.
 - e) Data mining é a ferramenta ideal para descoberta de relações entre os dados.

5. A figura a seguir é de um projeto de circuito desenvolvido no computador:



Fonte: <http://www.circuitoimpresso.pro.br/uploads/8/8/4/6/8846737/7402776_orig.ipa>. Acesso em: 27 set. 2017.

Considere um projeto de layout de placa de circuito impresso, onde temos milhares de trilhas e componentes diversos em escala milimétrica. Além disso, poderíamos armazenar detalhes em cada posição dos componentes, como o material a ser utilizado. Para obter uma maior robustez no armazenamento e disponibilização de projetos uma empresa que possui uma grande quantidade de projetos está procurando uma solução de

banco de dados ideal, e a solução correta para esta implementação é:

- a) Banco de dados multimídia.
- b) Banco de dados espacial.
- c) Banco de dados de projetos.
- d) Banco de dados eletrônicos.
- e) Banco de dados convencional.

Referências

CARDOSO, Virgínia; CARDOSO, Giselle. **Sistema de banco de dados:** uma abordagem introdutória e aplicada. São Paulo: Saraiva, 2012. 144 p.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados.** 6. ed. São Paulo: Pearson Addison Wesley, 2011. 790 p.

HEUSER, Carlos Alberto. **Projeto de banco de dados.** Porto Alegre: Bookman, 2009. 281 p.

LEHNING, Michael et al. **O quarto paradigma:** descobertas científicas na era da eScience; Instrumentalizando a Terra: a Ciência Ambiental e a próxima geração de redes de sensores. São Paulo: Oficina de textos, 2011.

MACHADO, Felipe Nery Rodrigues. **Projeto e implementação de banco de dados.** São Paulo: Érica, 2014. 401 p.

MEDEIROS, Luciano Frontino de. **Banco de dados:** princípios e prática. Curitiba: InterSaber, 2013. 183 p.

PUGA, Sandra; FRANÇA, Edson; GOYA, Milton. **Banco de Dados:** implementação em SQL, PL/SQL e Oracle 11g. São Paulo: Pearson Education do Brasil, 2013. 332 p.

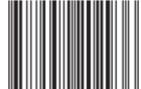
RAMARKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de gerenciamento de banco de dados.** 3. ed. Porto Alegre: McGraw-Hill, 2011. 884 p.

SILBERSCHATZ, Abrahan; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados.** Rio de Janeiro: Elsevier, 2012. 861 p.



unopar

ISBN 978-85-522-0293-6



9 788552 202936 >