

Implementação de banco de dados

Aula 1: Álgebra relacional

Apresentação

Nesta aula, você vai estudar o Modelo Relacional e a Álgebra Relacional. No mundo digital, nossas vidas são regidas por diversos sistemas. Para funcionar, esses sistemas necessitam de grande volume de dados. Os sistemas informatizados evoluíram ao longo do tempo de Sistemas Baseados em Arquivos para Sistemas Baseados em Banco de Dados.

Não é possível conceber, atualmente, qualquer sistema que possa prescindir de um Banco de Dados. Os Sistemas de Banco de Dados atuais são, majoritariamente, baseados no Modelo Relacional.

Entretanto, de que adiantaria termos uma estrutura de armazenamento se não conseguíssemos manipular os dados ali armazenados? Como o modelo relacional teve sua origem na matemática, essa mesma ciência nos fornece a ferramenta para manipulá-lo, a Álgebra Relacional.

Objetivos

- Conhecer as principais características de um Sistema de Banco de Dados;
- Identificar os componentes do modelo relacional;
- Conceituar Álgebra Relacional, suas operações e expressões.

Vamos dar início com um vídeo que orienta como baixar e instalar o PostGreSQL, que é um SGBD utilizado em toda a disciplina, possibilitando o acompanhamento prático e a realização das atividades propostas, fundamentais para que os objetivos da disciplina sejam alcançados.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Sistemas de banco de dados

Podemos iniciar o estudo com a apresentação da definição de FERREIRA et al. (1999, p. 62) para dados:

Dado é o “princípio em que se assenta uma discussão” ou o “elemento ou base para a formação de um juízo”. Ainda, assumindo-se um ponto de vista mais filosófico, dado é “o que se apresenta à consciência como imediato, não construído ou não elaborado”, FERREIRA et al. (1999, p. 602).

Essas definições são úteis para exemplificar o termo "dado" e situar sua definição de um ponto de vista mais humanizado.

Computacionalmente falando, pode-se definir **dado** como um valor armazenado e que por si só não quer dizer muita coisa. Uma lista de números - por exemplo, 10, 12, 20, 21, 23, 38 - não nos fornece nenhum significado, mas, por assim dizer, são os dados obtidos e anotados de alguma forma.

A necessidade de armazenar os dados precede a criação do conceito de Sistemas de Banco de dados.

Atenção

Antigamente, os dados eram armazenados em arquivos. Normalmente são associados a um único sistema de aplicação, trazendo, por consequência, redundância de dados quando os mesmos se faziam necessários em outros sistemas.

A maioria das organizações desenvolve os seus Sistemas de Informação um por vez, à medida que se tornam necessários, utilizando cada um o seu próprio conjunto de programas, arquivos e usuários.

A figura 1 mostra a estrutura de um sistema baseado em arquivo. Nela podemos notar que:

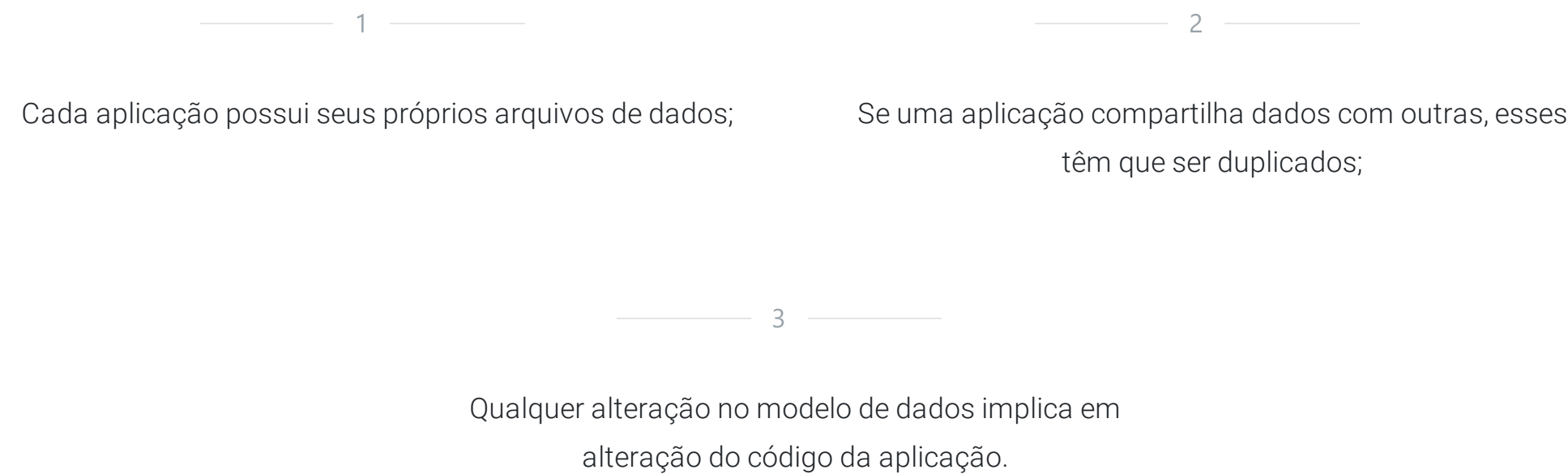


Figura 1: Sistemas baseados em arquivos.

O esquema apresentado acaba acarretando em:

Redundância de dados

Trata-se da duplicação dos mesmos dados em dois ou mais arquivos. O problema com a redundância é que as mudanças, ao serem feitas no arquivo de uma aplicação, não são automaticamente realizadas nos arquivos das outras aplicações, gerando a falta de integridade dos dados.

Dependência entre programas e dados

Os aplicativos tradicionais de banco de dados de arquivos são caracterizados pela dependência entre programas e dados, isso é, programas e dados desenvolvidos e organizados para uma aplicação são incompatíveis com os programas e dados organizados diferentemente para um outro aplicativo.

Custo excessivo em software

Resulta da criação, documentação e acompanhamento de muitos arquivos e aplicações diferentes.

Fonte: SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S., 2006.

A evolução da tecnologia associada ao grande aumento do volume de dados a ser armazenado e processado levou à criação de uma nova abordagem para o problema, o Sistema de Banco de Dados.

Um banco de dados é uma coleção de dados organizados de tal forma que possam ser acessados e utilizados por muitas aplicações diferentes. Ou seja, em vez de armazenar dados em arquivos separados para utilização, os dados são armazenados e organizados em um só local, permitindo compartilhamento e inter-relacionamento por múltiplos usuários.

A fim de usar a abordagem de banco de dados para o gerenciamento de dados, foi necessário um software adicional – um sistema de gerenciamento de banco de dados (SGBD). O SGBD consiste em um grupo de programas que podem ser usados como uma interface entre o banco de dados e um usuário ou um banco de dados e um programa aplicativo.(Fig 2)

 Figura 2: Sistemas de banco de dados.

Modelo relacional

Nos maiores SGBD comerciais, o modelo de dados utilizado é o Modelo Relacional criado com base na Teoria Matemática dos Conjuntos e na Álgebra Relacional, O Modelo de Dados Relacional tem as seguintes vantagens:

- É independente das linguagens de programação;
- É independente dos sistemas de gestão de bases de dados;
- É independente dos Sistemas Operacionais.

“O modelo relacional representa o banco de dados como uma coleção de relações.”

ELMASRI, R.; NAVATHE, S., 2015.

Da afirmativa acima, pode-se depreender uma relação como uma tabela de valores onde cada linha representa uma coleção de dados ou valores relacionados. Cada linha da tabela representa uma realidade ligada ao mundo real. O nome da tabela e o nome das colunas são definidos de forma que representem essa realidade. Pode-se ter uma tabela chamada ALUNO onde cada linha venha a ser o Nome de um Aluno a ser armazenado.

Podemos melhorar ainda mais essa representação, criando uma tabela com as colunas NOME, NÚMERO DE MATRÍCULA e CLASSE. Cada linha da tabela representa várias informações de um ALUNO e cada coluna isoladamente, representa uma informação específica desse aluno. Assim o nome da tabela e os nomes das colunas são capazes de dizer o que cada linha representa e também o que o conjunto representa.

No modelo relacional formal (Figura 3), cada linha é chamada de tupla, o nome da coluna é conhecido como atributo ou variável, e a tabela, relação. (FONSECA, 2016)

Regras de Integridade relacional

As regras de Integridade Relacional visam garantir a fidelidade de informações em um banco de dados. Basicamente, são três as formas mais comuns:


	 Clique nos botões para ver as informações.
Integridade de Domínio	✓
<p>Diz respeito aos dados que são permitidos nas colunas da relação (tabela). Um domínio é um conjunto de valores do mesmo tipo. Os domínios são, portanto, conjuntos/faixas de valores a partir dos quais os valores reais são adicionados às colunas de uma tabel.</p>	
Integridade de Entidade	✓
<p>Diz respeito à unicidade de linhas da relação. Para tal, cada tabela deve ter uma chave primária (<i>Primary Key</i> - PK). Dessa forma., as colunas escolhidas para PK devem ser únicas e de preenchimento obrigatório (não nulas), como, por exemplo, as colunas Código na tabela Emp e Codigodepto na tabela Depto (Fig 3);</p>	
Integridade Referencial	✓
<p>Diz respeito à consistência entre as tuplas de relações. Para tal, as tabelas devem possuir chaves estrangeiras (<i>Foreign Key</i> – FK). A FK de uma tabela com, por exemplo, CodigoDepto na tabela EMP (Fig3) deve referenciar uma PK de outra tabela, no caso CodigoDepto da tabela Depto.(Fig3). Os valores possíveis em uma FK são limitados aos existentes na PK referenciada. Se em um banco de dados você tentar incluir uma linha com valor de FK não existente na PK da outra tabela, o SGBD gerará um erro e não permitirá a operação, garantindo assim a consistência dos dados.</p>	

 Figura 3: Componentes do modelo relacional.

Álgebra relacional

É o conjunto básico de operações para o modelo relacional. Essas operações permitem a recuperação de tuplas mediante instruções de consulta aplicadas ao banco de dados. O resultado dessa recuperação também será uma relação, que pode ser usada em outras operações de consulta.

A importância da álgebra relacional:

- Provê fundamento formal para operações do modelo relacional;
- Alguns de seus conceitos são incorporados na linguagem SQL padrão.
- E o mais importante: é usada como base para implementar e otimizar as consultas em sistemas de banco de dados relacional. (FONSECA, 2016, p. 15)

As principais operações da álgebra relacional são:

———— 1 ————
Seleção

———— 2 ————
Projeção

———— 3 ————
Produto Cartesiano

———— 4 ————
Operações de Conjunto
(União, Intersecção, Diferença)

As operações de SELECT (SELEÇÃO) E PROJECT (PROJEÇÃO) são ditas operações unárias, pois atuam em relações únicas (ELMASRI, R.; NAVATHE, S., 2015).

Operação de seleção

A operação de Seleção recupera os dados de uma ou mais tabelas, selecionando um subconjunto de tuplas que satisfaça determinada condição de seleção. Essa condição de seleção é usada para dividir horizontalmente uma relação em dois conjuntos de tuplas – as tuplas que satisfazem a condição e são retornadas e as tuplas que não satisfazem a condição e são ignoradas. (FONSECA, 2016, p.16)

A forma geral de uma operação de seleção é:

$$(\text{nome da relação}) \langle \text{condição de seleção} \rangle \sigma$$

Onde:

• **<condição de seleção>** é a condição que a linha deve atender para ser selecionada.

Constituindo-se em uma expressão lógica, é construída a partir de cláusulas da forma:

<nome de atributo> <operador de comparação> <valor constante >, ou
<nome de atributo> <operador de comparação> <nome de atributo >

Onde:

<nome de atributo> é um atributo da relação definida em
<nome da relação>

<operador de comparação> é normalmente um dos
operadores relacionais **{=, <, ≤, ≥, ≠}** e

<valor constante> é um literal.

As cláusulas podem ser utilizadas em conjunto com os
operadores lógicos {AND, OR NOT}, seguindo a Lógica
Booleana, para formar uma condição de seleção
composta.

<nome da relação> é o nome da relação sobre a qual será
aplicada a operação de **Seleção**

A relação resultante da operação tem os mesmos atributos
da relação especificada em **<nome da relação>**.

Exemplo

Considere a Tabela Disciplina_Nota (Fig 4)

	NOME_ALUNO	NOME_DISCIPLINA	NOTA
1	José Geraldo	Álgebra	8
2	Eduardo Tomaz	Álgebra	10
3	Cleber Dutra	Álgebra	9
4	Hernesto Paula	Álgebra	9
5	Josué José	Álgebra	10

Figura 4: Tabela DISCIPLINA_NOTA

Para selecionar os alunos da disciplina ‘álgebra’ cuja nota foi integral, pode-se especificar cada uma dessas condições em uma operação de SELEÇÃO:

NOME_DISCIPLINA = ‘ALGEBRA’ AND NOTA = 10 (Disciplina_Nota)

NOME_DISCIPLINA = ‘ALGEBRA’ AND NOTA = 10

Resultando a nova relação:

	NOME_ALUNO	NOME_DISCIPLINA	NOTA
1	Eduardo Tomaz	Álgebra	10
2	Josué José	Álgebra	10

Figura 5: Resultado de uma operação de Seleção | Fonte: FONSECA, 2016.

Operação de Projeção

A operação de Projeção recupera os dados de certas colunas de uma tabela e descarta outras. Se existir a necessidade de mostrar apenas alguns atributos de uma tabela em detrimento de outros, usa-se a operação PROJEÇÃO. (FONSECA, 2016, p. 17)

A forma geral da operação de projeção é:

$$\pi \text{ <lista de atributos> (<nome da relação>)}$$

Onde:

<lista de atributos> representa a lista de atributos que o usuário deseja selecionar e

<nome da relação> representa a relação sobre a qual a operação **projeção** será aplicada.

Exemplo

Para selecionar as disciplinas e as ocorrências de notas nas disciplinas na tabela da fig 4, teremos:

$\pi \langle \text{Nome_Disciplina, Nota} \rangle (\langle \text{Disciplina_Nota} \rangle)$

Resultando na relação:

	NOME_DISCIPLINA	NOTA
1	Álgebra	8
2	Álgebra	9
3	Álgebra	10
4	Álgebra	10

Figura 6: Resultado de uma operação projeção.

Sequencialidade de Operações

A Projeção e a Seleção podem ser combinadas, de forma que apenas algumas linhas e algumas colunas retornem na operação. A forma geral de uma operação sequencializada é:

$\pi \langle \text{lista de atributos} \rangle (\sigma \langle \text{condição de seleção} \rangle (\langle \text{nome da relação} \rangle))$

Onde:

π representa a operação de projeção;

$\langle \text{lista de atributos} \rangle$ representa a lista de atributos que o usuário deseja selecionar;

σ representa a operação de seleção;

$\langle \text{condição de seleção} \rangle$ é a condição que a linha deve atender para ser selecionada;

$\langle \text{nome da relação} \rangle$ é o nome da relação sobre a qual será aplicada a operação de Seleção e Projeção

Exemplo

Para projetar a partir da Tabela Disciplina_Nota, o nome das disciplinas e a nota para alunos com nota diferente de 10 a expressão seria:

π Nome_Disciplina, Nota (σ <> 10 (Disciplina_Nota))

	NOME_DISCIPLINA	NOTA
1	Álgebra	8
2	Álgebra	9

Figura 7: Resultado de uma operação projeção após uma seleção.

Operação Produto Cartesiano

O produto cartesiano é uma operação binária que combina todas as tuplas de duas tabelas. O resultado de um produto cartesiano é uma nova tabela formada pela combinação das tuplas das tabelas sobre as quais aplicou-se a operação.

O formato geral do produto cartesiano entre duas tabelas A e B é:

Operação Junção

A operação Join (Junção) é usada para combinar as tuplas relacionadas em duas relações dentro de uma única tupla. Essa operação é uma das mais importantes em um banco de dados relacional, pois ela nos permite retornar os relacionamentos entre as relações (tabelas).

A forma geral da operação junção entre duas tabelas A e B é a seguinte:

A X B
<condição de junção>

Onde: <condição de junção> é uma expressão relacional, normalmente utilizando o operador =, que determina qual coluna da tabela A deve ser comparada com qual coluna da tabela B.

Observação: normalmente as colunas envolvidas na condição de junção são a Chave Primária de uma tabela e a Chave Estrangeira na outra.

Exemplo

Considere o banco de dados composto pela tabela Disciplina_Nota (Fig 4).

Imagine que nosso modelo de banco de dados definido anteriormente com a tabela DISCIPLINA_NOTA (Fig4) seja redefinido para um modelo mais trabalhado com as tabelas:

DISCIPLINA_NOTA, DISCIPLINA E PESSOA, inserindo uma chave primária (PK) que identifica unicamente um nome, a tabela DISCIPLINA_NOTA com a chave estrangeira (FK) FK_NOME_ALUNO relacionando com a tabela PESSOA e a FK_NOME_DISCIPLINA relacionando com a tabela DISCIPLINA. Temos, então, a seguinte melhoria em nosso banco de dados (FIG 8):

π NOME_DISCIPLINA, NOME_ALUNO, NOTA(PESSOA **X** DISCIPLINA_NOTA **X** Disciplina)
PK_NOME_PESSOA = FK_NOME_ALUNO FK_NOME_DISCIPLINA = PK_DISCIPLINA

	NOME_ALUNO	NOME_DISCIPLINA	NOTA
1	José Geraldo	Álgebra	8
2	Eduardo Tomaz	Álgebra	10
3	Cleber Dutra	Álgebra	9
4	Hernesto Paula	Álgebra	10
5	Josué José	Álgebra	10

Figura 9: Operação JOIN.

Operação de Conjuntos

São operações derivadas das operações matemáticas padrão definidas a partir da teoria dos conjuntos. (FONSECA, 2016, p. 18)

São elas:

📄 UNION (UNIÃO), INTERSECTION (INTERSEÇÃO) e MINUS (SUBTRAÇÃO)]

UNION

UNION é a operação de UNIÃO da teoria de conjuntos. Se temos as relações $R(A_1, A_2, \dots, A_n)$ e $S(B_1, B_2, \dots, B_n)$ para haver a operação União, os atributos de cada relação devem ser compatíveis entre si, ou seja, devem ter o mesmo grau (n) e os domínios de cada atributo devem ser iguais ao domínio do outro atributo, ou seja, $\text{dom}(A_1) = \text{dom}(B_1)$. Significando que cada relação possui o mesmo grau e que cada par de atributos possuem o mesmo domínio. Sendo assim, a operação UNION pode ser aplicada. (FONSECA, 2016, p. 18)

A forma geral da operação junção entre duas tabelas A e B é a seguinte:

$$A \cup B$$

A Operação de União é comutativa, ou seja, $A \cup B$ produz o mesmo resultado de $B \cup A$.

Exemplo: Temos a tabela DISCIPLINA_NOTA (Fig 4) e a nova tabela PROFESSOR (Fig10). Podemos fazer a união dos domínios 'NOME_PESSOA' com os nomes dos alunos e os nomes dos professores.

NOME_PROFESSOR	
1	Fagundes Teles
2	Ferreira
3	Elmasri
4	Navathe
5	Cleber Dutra

Figura 10: TABELA PROFESSOR.

$$\pi \text{ NOME_ALUNO (DISCIPLINA_NOTA)} \cup \text{PROFESSOR}$$

NOME_PESSOA	
1	Cleber Dutra
2	Eduardo Tomaz
3	Elmasri
4	Fagundes Teles
5	Ferreira
6	Hernesto Paula
7	José Geraldo
8	Josué José
9	Navathe

Figura 11: OPERAÇÃO UNION.

Atenção: Repare que a operação UNION não trouxe o nome repetido do Professor e aluno Cleber Dutra, essa operação garante tuplas únicas.

INTERSECTION

Da mesma forma como foi apresentada a operação UNION, pode-se usar a definição matemática de interseção para definirmos a operação de interseção entre as relações. As observações feitas para a operação UNION, no que diz respeito ao domínio dos atributos e ao grau da relação, também devem ser seguidas para a operação de INTERSECTION. (FONSECA, 2016, p. 20)

$$A \cap B$$

Essa operação é comutativa, ou seja, $A \cap B$ produz o mesmo resultado de $B \cap A$.

Exemplo: De acordo com Fonseca (2016), para a interseção entre os nomes de professores e alunos, temos:

π NOME_ALUNO (DISCIPLINA_NOTA) U PROFESSOR	
NOME_PESSOA	
1	Cleber Dutra

Figura 12: OPERAÇÃO INTERSECTION.

MINUS

Por fim, apresentamos a operação MINUS (SUBTRAÇÃO), que representa a diferença de conjunto. O resultado dessa operação, tomando-se nossas relações S e R apresentadas anteriormente, é uma relação que inclui todas as tuplas que estão em R, mas não estão em S. As observações feitas para a operação UNION e INTERSECTION, no que diz respeito ao domínio dos atributos e ao grau da relação, também devem ser seguidas para a operação MINUS. (FONSECA, 2016, p. 20)

A forma geral da operação Diferença entre duas tabelas A e B é a seguinte:

$$A - B$$

Essa operação não é comutativa, ou seja, $A - B$ produz um resultado diferente de $B - A$.

Para obtermos o nome dos Professores que não são homônimos de alunos (FONSECA, 2016, p. 20), a expressão seria:

$$\text{PROFESSOR} - \pi \text{ NOME_ALUNO (DISCIPLINA_NOTA)}$$

Resultando:

NOME_PESSOA	
1	Elmasri
2	Cleber Dutra
3	Fagundes Teles
4	Navathe

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Atividade

1. Defina chave primária, chave estrangeira e qual a importância desses atributos em um modelo relacional.

2. Na tabela a seguir, qual seria uma possível chave primária? Diga os motivos que levaram à sua escolha.

Nome da Coluna	Tipo de Dados	Permitir Nul...
NOME	NCHAR(50)	<input type="checkbox"/>
SOBRENOME	NCHAR(50)	<input type="checkbox"/>
TELEFONE	NCHAR(8)	<input checked="" type="checkbox"/>
ENDEREÇO	NCHAR(100)	<input checked="" type="checkbox"/>
CPFPAI	NCHAR(11)	<input type="checkbox"/>
CPFMAE	NCHAR(11)	<input type="checkbox"/>
RG	NCHAR(15)	<input type="checkbox"/>
CPF	NCHAR(10)	<input type="checkbox"/>

3. Defina: Tupla, Relação, Entidade, Atributo e Domínio.

4. Por que tuplas repetidas não são permitidas em uma relação?

5. Considerando a tabela a seguir, marque todas as opções que atendam à consulta solicitada como **correta** ou **errada** e justifique.

Listar o nome e o salário dos empregados que ganham mais de R\$ 2.000,00 .

Tabela Empregado				
ID	Nome	Cargo	Salário	Id_Depto
101	Carlos Antunes	Gerente de Vendas	5000,00	10
102	Pedro Leitão	Vendedor	1100,00	10
103	Antônio Ventura	Almoxarife	1200,00	20
104	Marco Aurélio	Vendedor	1500,00	10
105	Carla da Silva	Secretária	1000,00	30
106	Ivo Pereira	Contador	2000,00	40

- a) π Nome, Salário (σ salario \geq 2000,00 (Empregado))
- b) π Nome, Salário (σ salario = 2000,00 (Empregado))
- c) π Nome, Salário (σ salario $>$ 2000,00 (Empregado))
- d) π RG ,Nome, Salário (σ salario $>$ 2000,00 (Empregado))
- e) π RG ,Nome, Salário (σ salario \neq 2000,00 (Empregado))
- f) π salario $>$ 2000,00(σ Nome, Salário (Empregado))

6. Considerando as tabelas a seguir, marque todas as opções que atendam à consulta solicitada como correta ou errada e justifique.

Listar o Nome do Departamento e o Nome da Região onde ele está localizado.

Tabela Departamento		
ID	NOME	ID_REGIÃO
10	VENDAS	1
20	OPERAÇÕES	1
30	ADMINISTRATIVO	2
40	FINANCEIRO	3

Tabela Região	
ID	NOME
1	NORTE
2	CENTRO
3	SUL

- a) π Nome, Nome (DEPARTAMENTO Id_Regiao = Id REGIAO)
- b) π Departamento.Nome, Regiao.Nome (DEPARTAMENTO Id_Regiao = Id REGIAO)
- c) π Departamento.Nome, Regiao.Nome (DEPARTAMENTO REGIAO)
- d) π Departamento.Nome, Regiao.Nome (DEPARTAMENTO X REGIAO)
- e) π Departamento.Nome, Regiao.Nome (σ ID_REGIAO= REGIAO.ID (DEPARTAMENTO X REGIAO))
- f) π Departamento.Nome, Regiao.Nome (DEPARTAMENTO ID = ID_REGIAO REGIAO)

DATE, C. J. **Introdução a sistemas de banco de dados**. 7. ed. Rio de Janeiro: Campus, 2000.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 7. ed. São Paulo: Pearson Addison Wesley, 2015.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de banco de dados**. 5. ed. Rio de Janeiro: Campus, 2006.

Próxima aula

- Linguagem SQL.

Explore mais

Leia os textos:

- [Modelagem Relacional](#).

Acessibilidade

- [O Modelo Relacional de Dados - Parte I](#).

A Estácio, sempre preocupada com a necessidade de levar seus serviços a um maior número de pessoas, mudou seu site. Atenta às inovações de tecnologia e às principais tendências mundiais, está trazendo um site com recursos de acessibilidade, seguindo as recomendações do WCAG (Web Content Accessibility Guidelines), do W3C (World Wide Web Consortium), principal organização de padronização da World Wide Web, que desenvolve especificações técnicas e orientações para o mercado.

Com isso, pessoas com deficiência visual, baixa visão, daltonismo e mobilidade reduzida podem navegar em nosso site por meio de recursos que foram implementados para garantir este acesso, tais como alto contraste, aumento de fonte, teclas de atalho e navegação por teclado.

Para aumentar a fonte, é só clicar no símbolo de A+ em nossa barra de acessibilidade. Caso queira voltar ao tamanho de fonte original, é só clicar em A-.

Se for necessário, você também pode usar o zoom nativo do seu navegador, pressionando as teclas “Ctrl” e “+” para aumentar todo o site e “Ctrl” e “-” para diminuir. Para voltar ao padrão, pressione “Ctrl” e “0”.

Este site tem melhor acessibilidade quando acessado nas versões mais atualizadas do seu navegador web. Utilize sempre a versão mais recente de seu software.

Navegação por tabulação

Use a tecla Tab para navegar por elementos que recebem ação do usuário no site, tais como links, botões, campos de formulário e outros na ordem em que eles são apresentados na página, e Shift + Tab para retornar. Use as setas direcionais para acessar as informações textuais.

Sugestões de programas disponíveis para pessoas com deficiência

- Nitrous Voice Flux: controla o computador por voz. Gratuito;
- NVDA: software livre para ler tela – vários idiomas (Windows);
- YeoSoft Text: leitor de tela em inglês e português;

- Jaws for Windows: leitor de tela – vários idiomas;
 - Virtual Vision: leitor de telas em português do Brasil;
 - DOSVOX: sistema para deficientes visuais (Windows ou Linux).
-



Vídeo.