

MSc Sandir Rodrigues Campos

Banco de Dados 2

Aula 6

Exercícios com joins

Exercício 1: **INNER JOIN**

Crie duas tabelas, clientes e pedidos, com as seguintes estruturas:

- `CREATE TABLE clientes (id INT PRIMARY KEY, nome VARCHAR(255));`
- `CREATE TABLE pedidos (id INT PRIMARY KEY, cliente_id INT, valor DECIMAL(10, 2), FOREIGN KEY (cliente_id) REFERENCES clientes(id));`

Insira alguns dados:

- `INSERT INTO clientes (id, nome) VALUES (1, 'João'), (2, 'Maria'), (3, 'Pedro');`
- `INSERT INTO pedidos (id, cliente_id, valor) VALUES (1, 1, 100.00), (2, 1, 200.00), (3, 2, 50.00);`

Escreva uma consulta que retorne o nome do cliente e o valor do pedido, usando **INNER JOIN**.

Exercícios com joins

Exercício 2: **LEFT JOIN**

Crie duas tabelas, funcionarios e departamentos, com as seguintes estruturas:

- CREATE TABLE funcionarios (id INT PRIMARY KEY, nome VARCHAR(255));
- CREATE TABLE departamentos (id INT PRIMARY KEY, nome VARCHAR(255), funcionario_id INT, FOREIGN KEY (funcionario_id) REFERENCES funcionarios(id));

Insira alguns dados:

- INSERT INTO funcionarios (id, nome) VALUES (1, 'João'), (2, 'Maria'), (3, 'Pedro');
- INSERT INTO departamentos (id, nome, funcionario_id) VALUES (1, 'Vendas', 1), (2, 'Marketing', 2);

Escreva uma consulta que retorne o nome do funcionário e o nome do departamento, usando LEFT JOIN.

Exercícios com joins

Exercício 3: **RIGHT JOIN**

Crie duas tabelas, produtos e categorias, com as seguintes estruturas:

- `CREATE TABLE produtos (id INT PRIMARY KEY, nome VARCHAR(255));`
- `CREATE TABLE categorias (id INT PRIMARY KEY, nome VARCHAR(255), produto_id INT, FOREIGN KEY (produto_id) REFERENCES produtos(id));`

Insira alguns dados:

- `INSERT INTO produtos (id, nome) VALUES (1, 'Produto 1'), (2, 'Produto 2'), (3, 'Produto 3');`
- `INSERT INTO categorias (id, nome, produto_id) VALUES (1, 'Categoria 1', 1), (2, 'Categoria 2', 2);`

Escreva uma consulta que retorne o nome do produto e o nome da categoria, usando **RIGHT JOIN**.

Exercícios com joins

Exercício 4: **FULL OUTER JOIN**

Crie duas tabelas, clientes e pedidos, com as seguintes estruturas:

- `CREATE TABLE clientes (id INT PRIMARY KEY, nome VARCHAR(255));`
- `CREATE TABLE pedidos (id INT PRIMARY KEY, cliente_id INT, valor DECIMAL(10, 2), FOREIGN KEY (cliente_id) REFERENCES clientes(id));`

Insira alguns dados:

- `INSERT INTO clientes (id, nome) VALUES (1, 'João'), (2, 'Maria'), (3, 'Pedro');`
- `INSERT INTO pedidos (id, cliente_id, valor) VALUES (1, 1, 100.00), (2, 1, 200.00), (3, 2, 50.00);`

Escreva uma consulta que retorne o nome do cliente e o valor do pedido, usando **FULL OUTER JOIN**.

Exercícios com joins

Exercício 5: CROSS JOIN

Crie duas tabelas, produtos e categorias, com as seguintes estruturas:

- `CREATE TABLE produtos (id INT PRIMARY KEY, nome VARCHAR(255));`
- `CREATE TABLE categorias (id INT PRIMARY KEY, nome VARCHAR(255));`

Insira alguns dados:

- `INSERT INTO produtos (id, nome) VALUES (1, 'Produto 1'), (2, 'Produto 2'), (3, 'Produto 3');`
- `INSERT INTO categorias (id, nome) VALUES (1, 'Categoria 1'), (2, 'Categoria 2'), (3, 'Categoria 3');`

Escreva uma consulta que retorne o nome do produto e o nome da categoria, usando `CROSS JOIN`.

Exercícios com joins

Exercício 6: JOIN com subconsultas e agregação

Crie as seguintes tabelas:

- CREATE TABLE clientes (id INT PRIMARY KEY, nome VARCHAR(255));
- CREATE TABLE pedidos (id INT PRIMARY KEY, cliente_id INT, valor DECIMAL(10, 2), FOREIGN KEY (cliente_id) REFERENCES clientes(id));
- CREATE TABLE produtos (id INT PRIMARY KEY, nome VARCHAR(255), preco DECIMAL(10, 2));
- CREATE TABLE itens_pedidos (id INT PRIMARY KEY, pedido_id INT, produto_id INT, quantidade INT, FOREIGN KEY (pedido_id) REFERENCES pedidos(id), FOREIGN KEY (produto_id) REFERENCES produtos(id));

Insira alguns dados:

- INSERT INTO clientes (id, nome) VALUES (1, 'João'), (2, 'Maria'), (3, 'Pedro');
- INSERT INTO pedidos (id, cliente_id, valor) VALUES (1, 1, 100.00), (2, 1, 200.00), (3, 2, 50.00);
- INSERT INTO produtos (id, nome, preco) VALUES (1, 'Produto 1', 10.00), (2, 'Produto 2', 20.00), (3, 'Produto 3', 30.00);
- INSERT INTO itens_pedidos (id, pedido_id, produto_id, quantidade) VALUES (1, 1, 1, 2), (2, 1, 2, 1), (3, 2, 3, 3), (4, 3, 1, 1);

Escreva uma consulta que retorne o nome do cliente, o valor total dos pedidos e a quantidade total de produtos comprados, agrupando por cliente.

Exercícios com joins

Exercício7: Suponha que você tenha as seguintes tabelas:

- CREATE TABLE clientes (id INT PRIMARY KEY, nome VARCHAR(255), email VARCHAR(255));
- CREATE TABLE pedidos (id INT PRIMARY KEY, cliente_id INT, data_pedido DATE, FOREIGN KEY (cliente_id) REFERENCES clientes(id));
- CREATE TABLE produtos (id INT PRIMARY KEY, nome VARCHAR(255), preco DECIMAL(10, 2));
- CREATE TABLE itens_pedidos (id INT PRIMARY KEY, pedido_id INT, produto_id INT, quantidade INT, FOREIGN KEY (pedido_id) REFERENCES pedidos(id), FOREIGN KEY (produto_id) REFERENCES produtos(id));

Insira alguns dados:

- INSERT INTO clientes (id, nome, email) VALUES (1, 'João', 'joao@example.com'), (2, 'Maria', 'maria@example.com'), (3, 'Pedro', 'pedro@example.com');
- INSERT INTO pedidos (id, cliente_id, data_pedido) VALUES (1, 1, '2022-01-01'), (2, 1, '2022-01-15'), (3, 2, '2022-02-01'), (4, 3, '2022-03-01');
- INSERT INTO produtos (id, nome, preco) VALUES (1, 'Produto 1', 10.00), (2, 'Produto 2', 20.00), (3, 'Produto 3', 30.00), (4, 'Produto 4', 40.00);
- INSERT INTO itens_pedidos (id, pedido_id, produto_id, quantidade) VALUES (1, 1, 1, 2), (2, 1, 2, 1), (3, 2, 3, 3), (4, 3, 1, 1), (5, 4, 4, 2);

Faça uma consulta que retorne os seguintes dados:- Nome do cliente- Data do pedido- Nome do produto- Quantidade do produto- Preço total do produto (quantidade x preço)

Criação de índices

A criação de índices em MySQL é uma técnica importante para melhorar a performance de consultas em um banco de dados.

Tipos de índices

Existem dois tipos principais de índices em MySQL:

1. *Índice de chave primária (PRIMARY KEY): É um índice único que identifica cada linha em uma tabela.*
2. Índice de chave secundária (SECONDARY KEY): É um índice que pode ser usado para melhorar a performance de consultas.

Criação de índices

Criação de índice de chave secundária:

- `CREATE TABLE tabela (coluna1 INT, coluna2 VARCHAR(255), INDEX (coluna1));`

Criação de índice único

- `CREATE TABLE tabela (coluna1 INT, coluna2 VARCHAR(255), UNIQUE INDEX (coluna1));`

Criação de índice composto

- `CREATE TABLE tabela (coluna1 INT, coluna2 VARCHAR(255), INDEX (coluna1, coluna2));`

Adicionando Índices

Adicionando índices a uma tabela existente

Se você já tem uma tabela criada e deseja adicionar um índice, você pode usar a seguinte sintaxe:

- ALTER TABLE tabela ADD INDEX (coluna1);

Removendo índices

Se você deseja remover um índice, você pode usar a seguinte sintaxe:

- ALTER TABLE tabela DROP INDEX coluna1;

Exercícios de índices

Exercício 1: Criar índice em uma tabela existente

Crie uma tabela chamada clientes com as seguintes colunas:

```
CREATE TABLE clientes ( id INT PRIMARY KEY, nome  
VARCHAR(255), email VARCHAR(255));
```

Insira alguns dados:

```
INSERT INTO clientes (id, nome, email) VALUES (1, 'João',  
'joao@example.com'), (2, 'Maria', 'maria@example.com'), (3,  
'Pedro', 'pedro@example.com');
```

Crie um índice na coluna email:

Exercícios de índices

Exercício 2: Criar índice composto em uma tabela existente

Crie uma tabela chamada pedidos com as seguintes colunas:

- `CREATE TABLE pedidos (id INT PRIMARY KEY, cliente_id INT, data_pedido DATE, valor DECIMAL(10, 2));`

Insira alguns dados:

- `INSERT INTO pedidos (id, cliente_id, data_pedido, valor) VALUES (1, 1, '2022-01-01', 100.00), (2, 1, '2022-01-15', 200.00), (3, 2, '2022-02-01', 50.00);`

Crie um índice composto nas colunas cliente_id e data_pedido:

Exercícios de índices

Exercício 3: Criar índice único em uma tabela existente

Crie uma tabela chamada produtos com as seguintes colunas:

- `CREATE TABLE produtos (id INT PRIMARY KEY, nome VARCHAR(255), codigo_barras VARCHAR(255));`

Insira alguns dados:

- `INSERT INTO produtos (id, nome, codigo_barras) VALUES (1, 'Produto 1', '1234567890123'), (2, 'Produto 2', '2345678901234'), (3, 'Produto 3', '3456789012345');`

Crie um índice único na coluna `codigo_barras`:

Fim da aula 6