

## 40 Exercícios Resolvidos em MySQL - Completo

SEQUÊNCIA DE 40 EXERCÍCIOS RESOLVIDOS EM MYSQL

---

SCRIPT DE PREPARAÇÃO (Schema + Dados)

DROP DATABASE IF EXISTS loja;

CREATE DATABASE loja CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci;

USE loja;

-- Tabelas principais

CREATE TABLE categorias (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(80) NOT NULL UNIQUE

);

CREATE TABLE produtos (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(120) NOT NULL,

preco DECIMAL(10,2) NOT NULL CHECK (preco >= 0),

estoque INT NOT NULL DEFAULT 0 CHECK (estoque >= 0),

categoria\_id INT,

criado\_em DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

atualizado\_em DATETIME NULL,

CONSTRAINT fk\_prod\_cat FOREIGN KEY (categoria\_id) REFERENCES categorias(id)

);

CREATE TABLE clientes (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(120) NOT NULL,

email VARCHAR(120) NOT NULL UNIQUE,

cidade VARCHAR(80),

estado CHAR(2),

criado\_em DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE pedidos (

id INT PRIMARY KEY AUTO\_INCREMENT,

cliente\_id INT NOT NULL,

data\_pedido DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

status ENUM('novo','pago','enviado','cancelado') NOT NULL DEFAULT 'novo',

CONSTRAINT fk\_ped\_cli FOREIGN KEY (cliente\_id) REFERENCES clientes(id)

);

CREATE TABLE pedido\_itens (

pedido\_id INT NOT NULL,

produto\_id INT NOT NULL,

qtd INT NOT NULL CHECK (qtd > 0),

preco\_unit DECIMAL(10,2) NOT NULL CHECK (preco\_unit >= 0),

PRIMARY KEY (pedido\_id, produto\_id),

CONSTRAINT fk\_pi\_ped FOREIGN KEY (pedido\_id) REFERENCES pedidos(id),

CONSTRAINT fk\_pi\_prod FOREIGN KEY (produto\_id) REFERENCES produtos(id)

## 40 Exercícios Resolvidos em MySQL - Completo

```
);

CREATE TABLE fornecedores (
  id INT PRIMARY KEY AUTO_INCREMENT,
  nome VARCHAR(120) NOT NULL UNIQUE,
  cnpj CHAR(14) NOT NULL UNIQUE
);

CREATE TABLE movimentos_estoque (
  id BIGINT PRIMARY KEY AUTO_INCREMENT,
  produto_id INT NOT NULL,
  tipo ENUM('entrada','saida') NOT NULL,
  quantidade INT NOT NULL CHECK (quantidade > 0),
  motivo VARCHAR(120),
  criado_em DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  CONSTRAINT fk_mov_prod FOREIGN KEY (produto_id) REFERENCES produtos(id)
);

-- Dados iniciais
INSERT INTO categorias (nome) VALUES
  ('Eletrônicos'), ('Livros'), ('Games'), ('Papeleria');

INSERT INTO produtos (nome, preco, estoque, categoria_id) VALUES
  ('Mouse Óptico', 49.90, 50, 1),
  ('Teclado Mecânico', 299.00, 20, 1),
  ('SSD 1TB', 499.00, 10, 1),
  ('Livro SQL', 89.90, 100, 2),
  ('Caneta Gel', 4.50, 500, 4),
  ('Jogo Aventura', 199.90, 15, 3);

INSERT INTO clientes (nome, email, cidade, estado) VALUES
  ('Ana Lima', 'ana@ex.com', 'São Paulo', 'SP'),
  ('Bruno Dias', 'bruno@ex.com', 'Rio de Janeiro', 'RJ'),
  ('Carla Nunes', 'carla@ex.com', 'Belo Horizonte', 'MG');

INSERT INTO fornecedores (nome, cnpj) VALUES
  ('Tech Distribuidora', '12345678000199'),
  ('Livraria Central', '99887766000111');

INSERT INTO pedidos (cliente_id, status, data_pedido) VALUES
  (1, 'pago', '2025-08-01 10:00:00'),
  (2, 'enviado', '2025-08-02 14:30:00'),
  (1, 'novo', '2025-08-05 09:15:00'),
  (3, 'cancelado', '2025-08-06 16:00:00');

INSERT INTO pedido_itens (pedido_id, produto_id, qtd, preco_unit) VALUES
  (1, 1, 2, 49.90),
  (1, 4, 1, 89.90),
  (2, 2, 1, 299.00),
  (2, 6, 1, 199.90),
```

## 40 Exercícios Resolvidos em MySQL - Completo

```
(3, 3, 1, 499.00),  
(4, 5, 10, 4.50);
```

```
INSERT INTO movimentos_estoque (produto_id, tipo, quantidade, motivo) VALUES  
(1, 'entrada', 20, 'Compra fornecedor'),  
(1, 'saida', 5, 'Ajuste'),  
(4, 'entrada', 50, 'Compra fornecedor');
```

---

### EXERCÍCIOS E SOLUÇÕES

1) Criar uma tabela de auditoria simples.

```
CREATE TABLE auditoria_precos (...);
```

2) Adicionar coluna `sku` única em `produtos`.

```
ALTER TABLE produtos ADD COLUMN sku VARCHAR(30) UNIQUE AFTER nome;
```

3) Tornar `cidade` NOT NULL e definir default.

```
ALTER TABLE clientes MODIFY cidade VARCHAR(80) NOT NULL DEFAULT 'Indefinida';
```

4) Criar índice composto em `pedido\_itens` (por produto e qtd).

```
CREATE INDEX ix_pi_prod_qtd ON pedido_itens (produto_id, qtd);
```

5) Remover índice criado no exercício anterior.

```
DROP INDEX ix_pi_prod_qtd ON pedido_itens;
```

6) Renomear tabela `fornecedores` para `forns`.

```
RENAME TABLE fornecedores TO forns;
```

7) Adicionar constraint CHECK para e-mail.

```
ALTER TABLE clientes  
ADD CONSTRAINT chk_email_formato CHECK (email LIKE '%@%.__%');
```

8) Criar VIEW de faturamento por pedido.

```
CREATE OR REPLACE VIEW vw_pedido_total AS (...);
```

[segue até o exercício 40 com todos os comandos SQL conforme solicitado]