Nome: Gabriel Oliveira Silva de Pietro  
RA: 0125048  
  
ATIVIDADE CORREÇÃO DE ERROS – BANCO DE DADOS  
  
CREATE DATABASE erros;

USE erros;

-- EXERCICIO 01 --

CREATE TABLE usuarios (

id INT PRIMARY KEY, -- ERRO NA SINTAXE (PRIMARI KEY X)

nome VARCHAR(100),

email VARCHAR(100), -- ERRO NA SINTAXE (VCHAR X)

data\_nascimento DATE

);

CREATE TABLE pedidos (

id INT PRIMARY KEY,

id\_usuario INT,

data\_pedido DATETIME,

valor DECIMAL(10,2),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id) -- ERRO NA REFERÊNCIA DA FOREIGN KEY PARA OUTRA TABELA

);

-- Inserção de dados

INSERT INTO usuarios (id, nome, email, data\_nascimento)

VALUES (1, 'João Silva', 'joao@email.com', '1988-05-12'),

(2, 'Maria Souza', 'maria@email', '1992-08-23');

INSERT INTO pedidos (id, id\_usuario, data\_pedido, valor)

VALUES (1, 1, '2023-03-10 14:30', 259.90),

(2, 2, '2023-04-02 10:45', '399.50');

-- Atualização de dados

UPDATE usuarios

SET email = 'joao.silva@email.com'

WHERE id = 1; -- ERRO, POIS UPDATE SÓ PODE SER COM PRIMARY KEY

-- Remoção de pedidos com valor menor que 300

DELETE FROM pedidos

WHERE id = 1 AND id\_usuario = 1; -- ERRO, POIS DELETE É COM FROM

-- Consulta com JOIN

SELECT u.nome, p.data\_pedido, p.valor

FROM usuarios AS u

JOIN pedidos AS p ON u.id = p.id\_usuario; -- ERRO AO REFERENCIAR OUTRA TABELA NO ON

-- Consulta com LEFT JOIN

SELECT u.nome, p.data\_pedido

FROM usuarios AS u

LEFT JOIN pedidos AS p ON u.id = p.id\_usuario; -- ERRO NA SINTAXE POIS O ON P.USUARIO\_ID ESTAVA INVERTIDO

-- Consulta com RIGHT JOIN

SELECT u.nome, p.data\_pedido

FROM usuarios AS u

RIGHT JOIN pedidos AS p ON u.id = p.id\_usuario; -- ERRO NA SINTAXE POIS O ON ESTAVA INVERTIDO

-- Alteração da tabela

ALTER TABLE usuarios

MODIFY email VARCHAR(150),

ADD COLUMN telefone VARCHAR(20); -- ERRO POIS FALTOU A VIRGULA

-- EXERCICIO 02 --

CREATE TABLE clientes (

id INT PRIMARY KEY,

nome VARCHAR(100), -- ERRO NA SINTAXE (VARCH X)

telefone VARCHAR(20),

email VARCHAR(100)

);

CREATE TABLE vendas (

id INT PRIMARY KEY,

id\_cliente INT,

data\_venda DATE,

valor\_total DECIMAL(10,2), -- ERRO NA SINTAXE (DECIML X)

FOREIGN KEY (id\_cliente) REFERENCES clientes(id) -- ERRO NA REFERÊNCIA DA FOREIGN KEY E ADICIONAR PARENTESES

);

-- Inserção de dados

INSERT INTO clientes (id, nome, telefone, email)

VALUES (1, 'Carlos Lima', '1199999999', 'carlos@email.com'),

(2, 'Fernanda Dias', NULL, 'fernanda@email.com');

INSERT INTO vendas (id, id\_cliente, data\_venda, valor\_total)

VALUES (1, 1, '2023-02-15', 150.00),

(2, 2, '2023-03-01', NULL);

-- Consulta com WHERE e ORDER BY

SELECT nome, valor\_total

FROM vendas AS v JOIN clientes AS c ON v.id\_cliente = c.id

WHERE valor\_total >= 100

ORDER BY valor\_total DESC; -- ERRO NA SINTAXE (ORDER SEM O BY)

-- Uso de IFNULL

SELECT nome, IFNULL(telefone, 'Sem telefone') AS 'Possui Telefone'

FROM clientes;

-- Uso de CONCAT

SELECT CONCAT(nome, ' - ', email) AS contato -- ERRO NA SINTAXE (CONCAT NÃO É COM + E SIM COM VIRGULA)

FROM clientes;

-- Uso de CASE

SELECT nome, valor\_total,

CASE

WHEN valor\_total > 200 THEN 'Alto' -- ERRO NA SINTAXE (SEM WHEN)

WHEN valor\_total BETWEEN 100 AND 200 THEN 'Médio'-- ERRO NA SINTAXE (SEM WHEN)

ELSE 'Baixo'

END AS categoria

FROM vendas AS v

JOIN clientes AS c ON c.id = v.id\_cliente;

-- EXERCICIO 03 --

CREATE TABLE livros (

id INT PRIMARY KEY, -- ERRO NA SINTAXE (PRIMARY X)

titulo VARCHAR(150),

autor VARCHAR(100),

ano\_publicacao INT,

disponivel BOOL

);

CREATE TABLE emprestimos (

id INT PRIMARY KEY,

livro\_id INT,

nome\_leitor VARCHAR(100),

data\_emprestimo DATE,

data\_devolucao DATE,

FOREIGN KEY (livro\_id) REFERENCES livros(id)

);

-- Inserção de dados

INSERT INTO livros (id, titulo, autor, ano\_publicacao, disponivel)

VALUES (1, '1984', 'George Orwell', 1949, true),

(2, 'Dom Casmurro', 'Machado de Assis', 1899, false);

INSERT INTO emprestimos (id, livro\_id, nome\_leitor, data\_emprestimo, data\_devolucao)

VALUES (1, 2, 'Ana Paula', '2023-01-10', NULL),

(2, 1, 'Carlos Alberto', '2023-02-05', '2023-02-20');

-- Consulta com WHERE e ORDER BY

SELECT titulo, ano\_publicacao

FROM livros

WHERE disponivel = 'true'

ORDER BY ano\_publicacao DESC; -- ERRO NA SINTAXE (ORDER SEM O BY)

-- Uso de IFNULL

SELECT nome\_leitor, IFNULL(data\_devolucao, 'Não devolvido') AS 'Devolução'

FROM emprestimos;

-- Uso de CONCAT

SELECT CONCAT(nome\_leitor, ' - ', titulo) AS leitura -- ERRO NA SINTAXE (CONCAT NÃO É COM + E SIM COM VIRGULA)

FROM emprestimos AS e

JOIN livros AS l ON e.livro\_id = l.id;

-- Uso de CASE

SELECT titulo, disponivel,

CASE

WHEN disponivel = true THEN 'Disponível' -- ERRO NA SINTAXE (SEM WHEN)

WHEN disponivel = false THEN 'Emprestado' -- ERRO NA SINTAXE (SEM WHEN)

ELSE 'Desconhecido'

END AS status

FROM livros;

-- EXERCICIO 04 --

CREATE TABLE alunos (

id INT PRIMARY KEY, -- ERRO NA SINTAXE (PRIMERY KEY X)

nome VARCHAR(100),

data\_nascimento DATE,

peso FLOAT,

altura FLOAT,

telefone VARCHAR(15)

);

CREATE TABLE treinos (

id INT PRIMARY KEY,

aluno\_id INT,

tipo VARCHAR(50),

duracao INT, -- em minutos

data DATE,

FOREIGN KEY (aluno\_id) REFERENCES alunos(id) -- ERRO NA SINTAXE (FALTOU OS PARENTESES NO ALUNO\_ID)

);

-- Inserção de dados

INSERT INTO alunos (id, nome, data\_nascimento, peso, altura, telefone)

VALUES (1, 'Lucas Nogueira', '1995-09-12', 78.5, 1.75, '11988776655'),

(2, 'Patrícia Alves', '1988-03-22', 65.3, NULL, '11999887766');

INSERT INTO treinos (id, aluno\_id, tipo, duracao, data)

VALUES (1, 1, 'Cardio', '45', '2023-05-10'), -- ERRO NA FORMATAÇÃO DA DURAÇÃO (ESPERA SE RECEBER EM STRING, NÃO INT)

(2, 2, 'Musculação', '60', '2023-05-12'); -- valor numérico como string

-- Consulta com WHERE e ORDER BY

SELECT nome, peso, altura

FROM alunos

WHERE peso > 70 AND altura IS NOT NULL

ORDER BY altura DESC; -- ERRO NA SINTAXE (ORDER SEM O BY)

-- IFNULL para altura

SELECT nome, IFNULL(altura, 0) AS altura

FROM alunos;

-- CONCAT para mensagem personalizada

SELECT CONCAT('Aluno: ', nome, ' - Telefone: ', telefone) AS 'Informações Completas dos Alunos'

FROM alunos;

-- CASE para avaliação do treino

SELECT tipo, duracao,

CASE

WHEN duracao < 30 THEN 'Curto' -- ERRO NA SINTAXE (CASE SEM WHEN)

WHEN duracao BETWEEN 30 AND 60 THEN 'Moderado' -- ERRO NA SINTAXE (CASE SEM WHEN)

ELSE 'Longo'

END AS intensidade

FROM treinos;

-- EXERCICIO 05 --

CREATE TABLE jogadores (

id INT PRIMARY KEY,

nome VARCHAR(100),

nickname VARCHAR(50),

pais\_origem VARCHAR(50)

);

CREATE TABLE torneios (

id INT PRIMARY KEY,

nome VARCHAR(100),

premiacao DECIMAL(8,2),

data\_torneio DATE

);

CREATE TABLE inscricoes (

jogador\_id INT,

torneio\_id INT,

data\_inscricao DATE,

status VARCHAR(20),

CONSTRAINT pksTabela PRIMARY KEY (jogador\_id, torneio\_id),

FOREIGN KEY (jogador\_id) REFERENCES jogadores(id), -- ERRO NA SINTAXE (REFERENCE X)

FOREIGN KEY (torneio\_id) REFERENCES torneios(id)

);

-- Inserção de dados

INSERT INTO jogadores (id, nome, nickname, pais\_origem)

VALUES (1, 'Lucas Pereira', 'Lukao', 'Brasil'),

(2, 'Emily Chan', 'ShadowQueen', 'China');

INSERT INTO torneios (id, nome, premiacao, data\_torneio)

VALUES (1, 'Summer Cup', 5000, '2023-07-10'),

(2, 'Winter Clash', 7500.00, '2023-12-15');

INSERT INTO inscricoes (jogador\_id, torneio\_id, data\_inscricao, status)

VALUES (1, 1, '2023-06-01', 'confirmado'),

(2, 2, '2023-11-20', NULL);

-- Atualização

UPDATE inscricoes

SET status = 'confirmado' -- ERRO NA SINTAXE (CONFIRMADO TEM QUE SER STRING COLOCANDO '')

WHERE jogador\_id = 2 AND torneio\_id = 2;

-- Exclusão

DELETE FROM inscricoes -- ERRO NA SINTAXE (FALTOU FROM EM DELETE)

WHERE status IS NULL;

-- Consulta com JOIN

SELECT j.nome, t.nome, i.status

FROM jogadores AS j

JOIN inscricoes AS i ON i.jogador\_id = j.id

JOIN torneios AS t ON t.id = i.torneio\_id;

-- LEFT JOIN

SELECT j.nome, t.nome

FROM jogadores AS j

LEFT JOIN inscricoes AS i ON j.id = i.jogador\_id

LEFT JOIN torneios AS t ON i.torneio\_id = t.id;

-- RIGHT JOIN

SELECT j.nome, t.nome

FROM jogadores AS j

RIGHT JOIN inscricoes AS i ON j.id = i.jogador\_id

RIGHT JOIN torneios AS t ON i.torneio\_id = t.id;

-- WHERE e ORDER BY

SELECT nickname, pais\_origem

FROM jogadores

WHERE pais\_origem = 'Brasil'

ORDER BY pais\_origem; -- ERRO NA SINTAXE (ORDER SEM O BY)

-- IFNULL e CONCAT

SELECT CONCAT(nome, ' (', nickname, ')') AS Jogador,

IFNULL(pais\_origem, 'Não informado') AS Pais

FROM jogadores;

-- CASE

SELECT j.nome, i.status,

CASE

WHEN i.status = 'confirmado' THEN 'Participante' -- ERRO NA SINTAXE (SEM WHEN)

WHEN i.status IS NULL THEN 'Aguardando' -- ERRO NA SINTAXE (SEM WHEN)

ELSE 'Outro'

END AS situacao

FROM jogadores AS j

JOIN inscricoes i ON j.id = i.jogador\_id;

-- ALTER TABLE

ALTER TABLE jogadores

ADD nacionalidade VARCHAR(50),

MODIFY nickname VARCHAR(100);