

UEMG - Universidade do Estado de Minas Gerais
Curso - Sistemas de Informação - 3º período, noturno
Docente: Uemerson Pinheiro Junior
Matéria: Banco de Dados 1
Data: 05/08/2024

Discentes: Igor Angelotti Marques e Gabriel Machado da Silva

AULA 06

Exercício A)

SCRIPTS:

-- criação de tabelas

```
CREATE TABLE Categorias (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Livros (  
  id SERIAL PRIMARY KEY,  
  titulo VARCHAR(150) NOT NULL,  
  isbn VARCHAR(20) UNIQUE NOT NULL,  
  data_publicacao DATE NOT NULL,  
  categoria VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Autores (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  biografia TEXT  
);
```

```
CREATE TABLE LivrosAutores (  
  livro_id INTEGER NOT NULL,  
  autor_id INTEGER NOT NULL,  
  PRIMARY KEY (livro_id, autor_id),  
  FOREIGN KEY (livro_id) REFERENCES Livros(id),  
  FOREIGN KEY (autor_id) REFERENCES Autores(id)  
);
```

```
CREATE TABLE Membros (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  endereco TEXT,  
  telefone VARCHAR(20) UNIQUE NOT NULL,
```

```
data_adesao DATE NOT NULL
);
```

```
CREATE TABLE Emprestimos (
  id SERIAL PRIMARY KEY,
  data_inicio TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
  data_devolucao DATE,
  livro_id INTEGER NOT NULL,
  membro_id INTEGER NOT NULL,
  FOREIGN KEY (livro_id) REFERENCES Livros(id),
  FOREIGN KEY (membro_id) REFERENCES Membros(id)
);
```

```
INSERT INTO Categorias (nome)
```

```
VALUES
```

```
('Ficção'),
('Romance'),
('Aventura'),
('Mistério'),
('Autoajuda'),
('História'),
('Ciência'),
('Fantasia'),
('Biografia'),
('Poesia');
```

```
INSERT INTO Membros (nome, endereco, telefone, data_adesao)
```

```
VALUES
```

```
('João Silva', 'Rua 1, 123', '11987654321', '2020-01-01'),
('Maria Rodrigues', 'Rua 2, 456', '11991234567', '2020-02-01'),
('Pedro Oliveira', 'Rua 3, 789', '11998765432', '2020-03-01'),
('Ana Souza', 'Rua 4, 101', '11987654322', '2020-04-01'),
('Luiz Carlos', 'Rua 5, 202', '11991234568', '2020-05-01'),
('Julia Martins', 'Rua 6, 303', '11998765433', '2020-06-01'),
('Rafael Santos', 'Rua 7, 404', '11987654323', '2020-07-01'),
('Gabriela Lima', 'Rua 8, 505', '11991234569', '2020-08-01'),
('Leonardo Costa', 'Rua 9, 606', '11998765434', '2020-09-01'),
('Beatriz Almeida', 'Rua 10, 707', '11987654324', '2020-10-01'),
('Carlos Eduardo', 'Rua 11, 808', '11991234570', '2020-11-01'),
('Patricia Mendes', 'Rua 12, 909', '11998765435', '2020-12-01'),
('Ricardo Dias', 'Rua 13, 1010', '11987654325', '2021-01-01'),
('Sandra Silva', 'Rua 14, 1111', '11991234571', '2021-02-01'),
('Thiago Oliveira', 'Rua 15, 1212', '11998765436', '2021-03-01'),
('Vanessa Rodrigues', 'Rua 16, 1313', '11987654326', '2021-04-01'),
('Wagner Souza', 'Rua 17, 1414', '11991234572', '2021-05-01'),
('Ximena Martins', 'Rua 18, 1515', '11998765437', '2021-06-01');
```

```
INSERT INTO Livros (titulo, isbn, data_publicacao, categoria_id)
VALUES
```

```
('O Senhor dos Anéis', '9788571640011', '1954-07-29', 1),
('Harry Potter e a Pedra Filosofal', '9788571640028', '1997-06-26', 8),
('O Código Da Vinci', '9788571640035', '2003-03-18', 3),
('A Menina que Roubava Livros', '9788571640042', '2005-03-14', 2),
('O Poder do Agora', '9788571640059', '1997-08-01', 5),
('A Arte da Guerra', '9788571640066', '2001-09-11', 6),
('O Alquimista', '9788571640073', '1988-05-01', 9),
('O Diário de Anne Frank', '9788571640080', '1947-06-25', 10),
('A Casa dos Espíritos', '9788571640097', '1982-01-01', 1),
('O Nome da Rosa', '9788571640103', '1980-10-01', 6);
```

```
INSERT INTO Autores (nome, biografia)
```

```
VALUES
```

```
('J.K. Rowling', 'Escritora britânica, autora da série Harry Potter'),
('J.R.R. Tolkien', 'Escritor britânico, autor da série O Senhor dos Anéis'),
('George R.R. Martin', 'Escritor americano, autor da série As Crônicas de Gelo e Fogo'),
('Stephen King', 'Escritor americano, autor de romances de terror e suspense'),
('Jane Austen', 'Escritora britânica, autora de romances clássicos como Orgulho e Preconceito');
```

```
INSERT INTO Emprestimos (data_inicio, data_devolucao, livro_id, membro_id)
```

```
VALUES
```

```
('2022-01-01', '2022-01-15', 11, 15),
('2022-01-05', '2022-01-20', 12, 16),
('2022-01-10', '2022-01-25', 13, 18),
('2022-01-15', '2022-02-01', 18, 24),
('2022-01-20', '2022-02-05', 15, 22);
```

```
UPDATE Membros
```

```
SET endereco = 'Rua 1, 456', telefone = '11990004567'
```

```
WHERE id = 16;
```

```
UPDATE Membros
```

```
SET data_adesao = '2024-01-01'
```

```
WHERE data_adesao = '2021-01-01';
```

```
DELETE FROM Membros
```

```
WHERE id = 20;
```

```
DELETE FROM Membros
```

```
WHERE id = 17;
```

```
-- Quais livros foram emprestados no último mês?
```

```
SELECT m.nome, COUNT(e.livro_id) AS quantidade_livros
FROM Membros m
LEFT JOIN Emprestimos e ON m.id = e.membro_id
WHERE e.data_devolucao IS NULL
GROUP BY m.nome;
```

-- Qual é a lista de todos os membros e quantos livros eles têm emprestados atualmente?

```
SELECT m.nome, COUNT(e.livro_id) AS quantidade_livros
FROM Membros m
LEFT JOIN Emprestimos e ON m.id = e.membro_id
WHERE e.data_devolucao IS NULL
GROUP BY m.nome;
```

-- Quais autores têm mais de 5 livros registrados na biblioteca?

```
SELECT a.nome, COUNT(la.livro_id) AS quantidade_livros
FROM Autores a
JOIN LivrosAutores la ON a.id = la.autor_id
GROUP BY a.nome
HAVING COUNT(la.livro_id) > 5;
```

-- Quais livros estão disponíveis para empréstimo?

```
SELECT l.titulo
FROM Livros l
LEFT JOIN Emprestimos e ON l.id = e.livro_id
WHERE e.livro_id IS NULL;
```

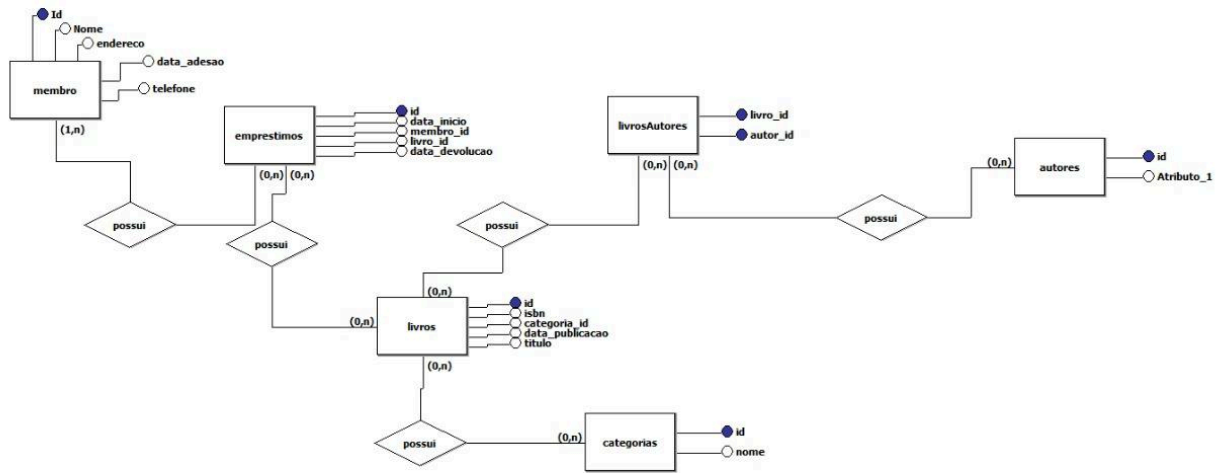
-- Quais membros não devolveram livros há mais de um mês?

```
SELECT m.nome
FROM Membros m
JOIN Emprestimos e ON m.id = e.membro_id
WHERE e.data_devolucao IS NULL AND e.data_inicio <= DATE_SUB(CURRENT_DATE,
INTERVAL 1 MONTH);
```

-- Quantos livros de cada categoria a biblioteca possui?

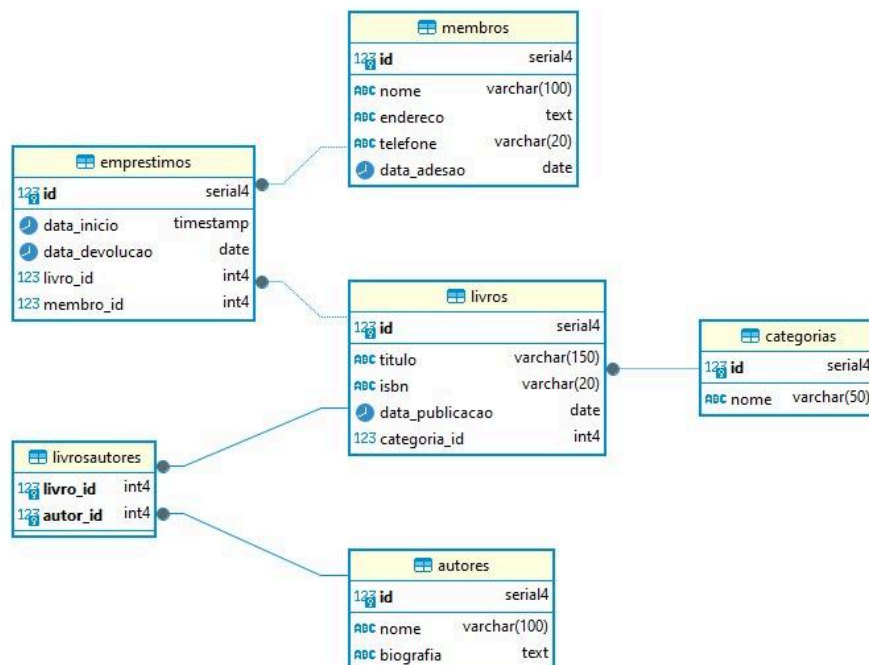
```
SELECT c.nome, COUNT(l.id) AS quantidade_livros
FROM Livros l
JOIN Categorias c ON l.categoria_id = c.id
GROUP BY c.nome;
```

MER:



DER:

<postgres 2> Script-2 membros livros empréstimos autores biblioteca2.erd X



Exercício B)

SCRIPTS:

-- Criação das Tabelas

```
CREATE TABLE Pacientes (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  CPF VARCHAR(11) UNIQUE NOT NULL,  
  data_nasc DATE NOT NULL,  
  endereco VARCHAR(100) NOT NULL,  
  telefone VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE Médicos (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  CRM VARCHAR(10) UNIQUE NOT NULL,  
  especialidade VARCHAR(50) NOT NULL,  
  telefone VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE Consultas (  
  id SERIAL PRIMARY KEY,  
  data_consulta TIMESTAMP NOT NULL,  
  paciente_id INTEGER NOT NULL,  
  medico_id INTEGER NOT NULL,  
  diagnostico TEXT,  
  FOREIGN KEY (paciente_id) REFERENCES Pacientes(id),  
  FOREIGN KEY (medico_id) REFERENCES Médicos(id)  
);
```

```
CREATE TABLE Tratamentos (  
  id SERIAL PRIMARY KEY,  
  descricao TEXT NOT NULL,  
  data_inicio DATE NOT NULL,  
  data_termino DATE,  
  paciente_id INTEGER NOT NULL,  
  medico_id INTEGER NOT NULL,  
  FOREIGN KEY (paciente_id) REFERENCES Pacientes(id),  
  FOREIGN KEY (medico_id) REFERENCES Médicos(id)  
);
```

-- inserts

```
INSERT INTO Pacientes (nome, CPF, data_nasc, endereco, telefone) VALUES
```

```
( 'João Silva', '12345678901', '1990-01-01', 'Rua 1, 123', '123456789'),
( 'Maria Souza', '98765432109', '1980-06-15', 'Rua 2, 456', '987654321'),
( 'Pedro Oliveira', '11122233344', '1995-03-20', 'Rua 3, 789', '111222333'),
( 'Ana Costa', '44455566677', '1985-09-10', 'Rua 4, 901', '444555666'),
( 'Luiz Santos', '77788899900', '1992-02-25', 'Rua 5, 234', '777888999'),
( 'Beatriz Lima', '33344455566', '1988-11-05', 'Rua 6, 567', '333444555'),
( 'Carlos Dias', '66677788899', '1991-04-15', 'Rua 7, 890', '666777888'),
( 'Eduardo Silva', '99900011122', '1986-07-20', 'Rua 8, 345', '999000111'),
( 'Francisco Oliveira', '22233344455', '1993-01-10', 'Rua 9, 678', '222333444'),
( 'Gabriela Costa', '55566677788', '1989-05-25', 'Rua 10, 901', '555666777');
```

```
INSERT INTO Médicos (nome, CRM, especialidade, telefone) VALUES
```

```
( 'Dr. João', '1234567890', 'Clínico Geral', '123456789'),
( 'Dr. Maria', '9876543210', 'Pediatra', '987654321'),
( 'Dr. Pedro', '11122233344', 'Cardiologista', '111222333'),
( 'Dr. Ana', '44455566677', 'Ginecologista', '444555666'),
( 'Dr. Luiz', '77788899900', 'Ortopedista', '777888999');
```

```
INSERT INTO Consultas (data_consulta, paciente_id, medico_id, diagnostico) VALUES
```

```
( '2022-01-01 10:00:00', 1, 1, 'Resfriado'),
( '2022-01-02 11:00:00', 2, 2, 'Gripe'),
( '2022-01-03 12:00:00', 3, 3, 'Dor de cabeça'),
( '2022-01-04 13:00:00', 4, 4, 'Problema de pele'),
( '2022-01-05 14:00:00', 5, 5, 'Lesão esportiva'),
( '2022-01-06 15:00:00', 6, 1, 'Infecção urinária'),
( '2022-01-07 16:00:00', 7, 2, 'Problema de estômago'),
( '2022-01-08 17:00:00', 8, 3, 'Dor de dente'),
( '2022-01-09 18:00:00', 9, 4, 'Problema de ouvido'),
( '2022-01-10 19:00:00', 10, 5, 'Lesão muscular'),
( '2022-01-11 20:00:00', 1, 1, 'Revisão'),
( '2022-01-12 21:00:00', 2, 2, 'Revisão'),
( '2022-01-13 22:00:00', 3, 3, 'Revisão'),
( '2022-01-14 23:00:00', 4, 4, 'Revisão'),
( '2022-01-15 00:00:00', 5, 5, 'Revisão'),
( '2022-01-16 01:00:00', 6, 1, 'Revisão'),
( '2022-01-17 02:00:00', 7, 2, 'Revisão'),
( '2022-01-18 03:00:00', 8, 3, 'Revisão'),
( '2022-01-19 04:00:00', 9, 4, 'Revisão'),
( '2022-01-20 05:00:00', 10, 5, 'Revisão');
```

```
-- Atualização e Remoção
```

```
UPDATE Pacientes SET telefone = '999999999' WHERE id = 1;
```

```
DELETE FROM Pacientes WHERE id = 10;
```

```
-- Quais consultas foram realizadas no último mês?
```

```
SELECT * FROM Consultas WHERE data_consulta >= NOW() - INTERVAL '1 month';
```

-- Qual é a lista de todos os médicos e quantos pacientes cada um atendeu atualmente?

```
SELECT m.nome, COUNT(c.paciente_id) AS quantidade_pacientes
FROM Médicos m
JOIN Consultas c ON m.id = c.medico_id
GROUP BY m.nome;
```

-- Quais médicos têm mais de 3 consultas registradas?

```
SELECT m.nome, COUNT(c.id) AS quantidade_consultas
FROM Médicos m
JOIN Consultas c ON m.id = c.medico_id
GROUP BY m.nome
HAVING COUNT(c.id) > 3;
```

-- Quais tratamentos estão em andamento?

```
SELECT * FROM Tratamentos WHERE data_termino IS NULL;
```

-- Qual é a lista de todos os pacientes e quantos tratamentos cada um está recebendo atualmente?

```
SELECT p.nome, COUNT(t.id) AS quantidade_tratamentos
FROM Pacientes p
JOIN Tratamentos t ON p.id = t.paciente_id
GROUP BY p.nome;
```

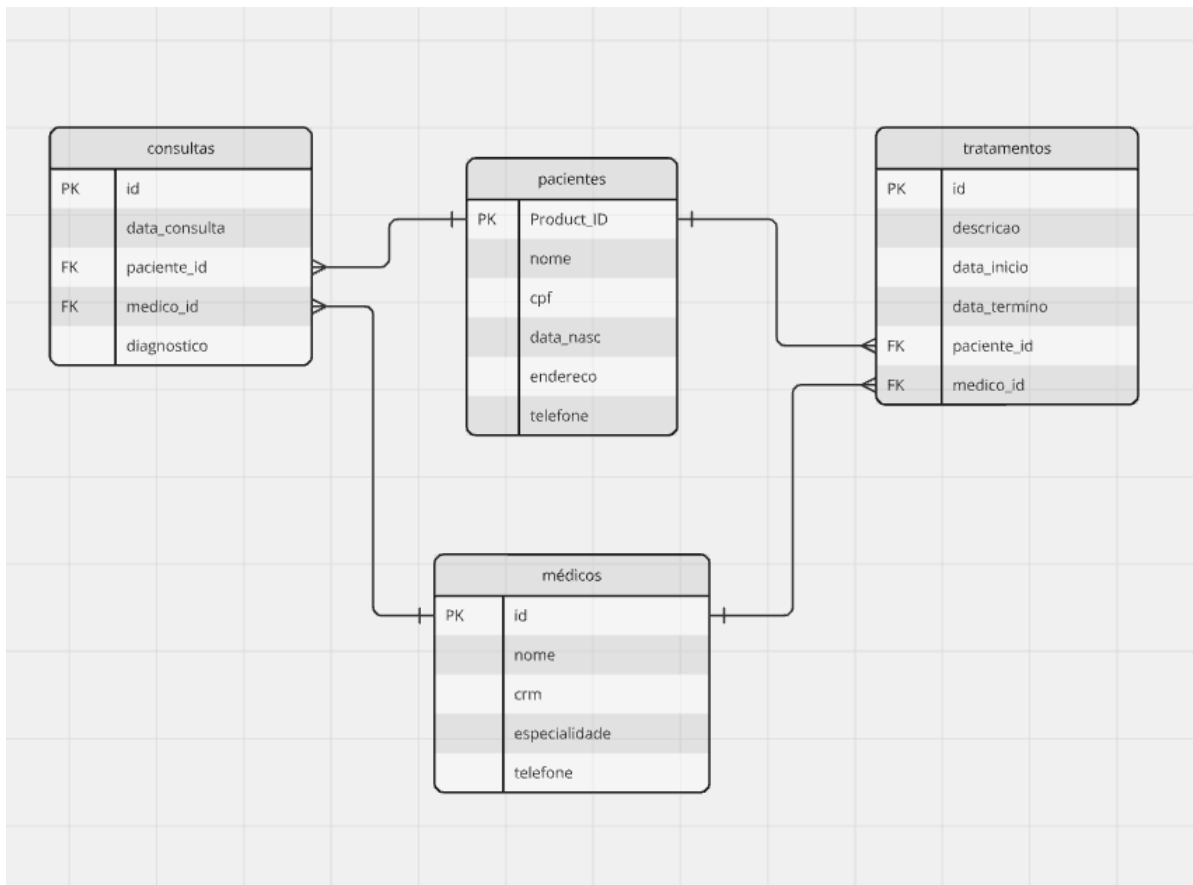
-- Quais médicos prescreveram tratamentos no último mês?

```
SELECT m.nome, t.descricao
FROM Médicos m
JOIN Tratamentos t ON m.id = t.medico_id
WHERE t.data_inicio >= NOW() - INTERVAL '1 month';
```

-- Quais pacientes têm consultas marcadas para a próxima semana?

```
SELECT p.nome, c.data_consulta
FROM Pacientes p
JOIN Consultas c ON p.id = c.paciente_id
WHERE c.data_consulta >= NOW() AND c.data_consulta <= NOW() + INTERVAL '1 week';
```


MER:



DER:

