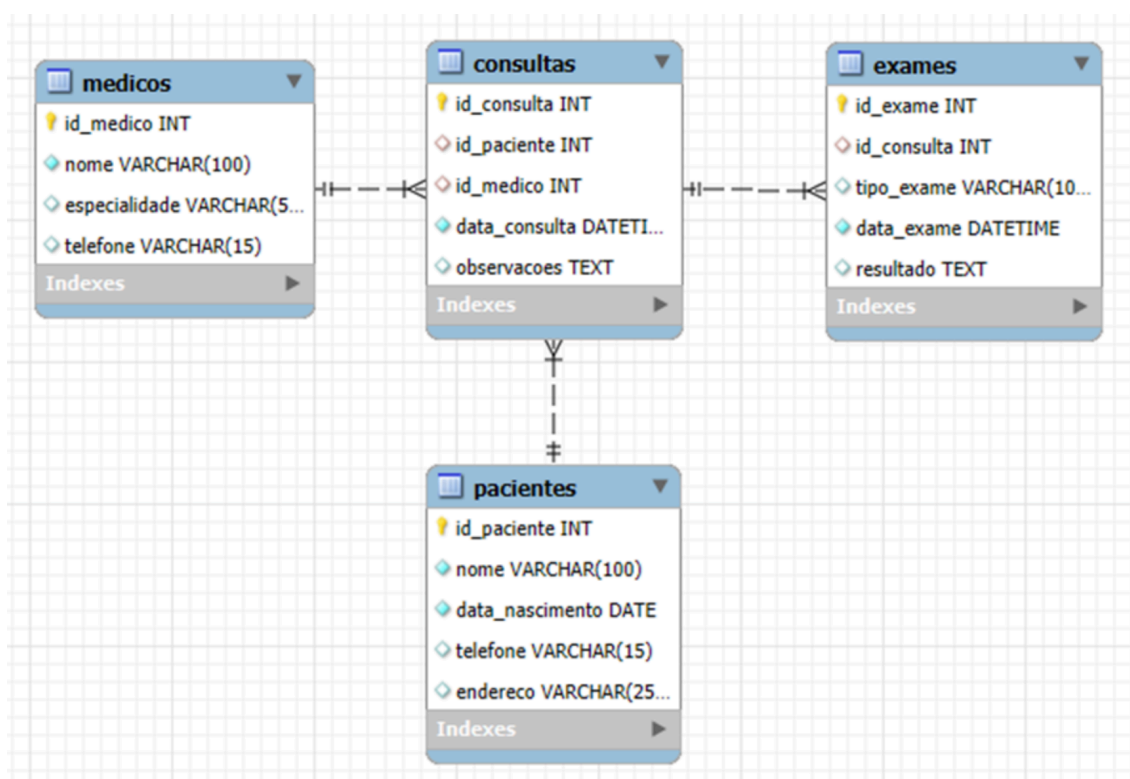


Relatório Sistema de Gerenciamento Hospitalar

Este relatório descreve a criação de um banco de dados para um hospital, incluindo a estrutura das tabelas, a inserção de dados e a realização de consultas. O objetivo é demonstrar a aplicação de comandos DDL, DML e DQL, com ênfase especial no uso de JOINS, incluindo LEFT JOIN e RIGHT JOIN.

1. Design do Banco de Dados

O banco de dados foi criado com o nome `hospital`. Dentro dele, foram definidas quatro tabelas principais: `Pacientes`, `Medicos`, `Consultas` e `Exames`.



- **Pacientes:** Armazena informações sobre os pacientes. Cada paciente tem um `id_paciente` único.
- **Medicos:** Contém dados sobre médicos, incluindo especialidade e telefone. Cada médico tem um `id_medico` único.
- **Consultas:** Relaciona médicos a pacientes em consultas. Inclui data e observações. As chaves estrangeiras `id_paciente` e `id_medico` conectam-se às tabelas correspondentes.
- **Exames:** Contém dados sobre os exames realizados durante as consultas, com uma chave estrangeira `id_consulta` referenciando a tabela de consultas.

2. Inserção de Dados

Foram inseridos pelo menos cinco registros em cada uma das tabelas, garantindo a integridade referencial ao utilizar IDs válidos em tabelas relacionadas. Por exemplo:

```
-- Inserir Pacientes
INSERT INTO Pacientes (nome, data_nascimento, telefone, endereco) VALUES
('Maria Silva', '1985-07-20', '999999999', 'Rua A, 123'),
('João Pereira', '1990-11-11', '988888888', 'Rua B, 456'),
('Ana Costa', '1995-04-10', '977777777', 'Rua C, 789'),
('Carlos Oliveira', '1982-06-25', '966666666', 'Rua D, 101'),
('Luciana Souza', '1992-08-30', '955555555', 'Rua E, 202');

-- Inserir Medicos
INSERT INTO Medicos (nome, especialidade, telefone) VALUES
('Dr. Paulo Almeida', 'Cardiologia', '934567890'),
('Dr. Rafael Costa', 'Pediatria', '934567891'),
('Dr. Fernanda Lima', 'Ortopedia', '934567892'),
('Dr. João Batista', 'Cardiologia', '934567893'),
('Dr. Beatriz Souza', 'Pediatria', '934567894');

-- Inserir Consultas
INSERT INTO Consultas (id_paciente, id_medico, data_consulta, observacoes) VALUES
(1, 1, '2024-11-25 10:00:00', 'Consulta de rotina'),
(2, 2, '2024-11-26 14:00:00', 'Consulta para acompanhamento'),
(3, 3, '2024-11-27 09:00:00', 'Consulta para dor nas costas'),
(4, 4, '2024-11-28 15:00:00', 'Consulta de urgência'),
(5, 5, '2024-11-29 11:00:00', 'Consulta pediátrica');

-- Inserir Exames
INSERT INTO Exames (id_consulta, tipo_exame, data_exame, resultado) VALUES
(1, 'Eletrocardiograma', '2024-11-26 10:00:00', 'Normal'),
(2, 'Exame de sangue', '2024-11-27 10:00:00', 'Normal'),
(3, 'Raio-X', '2024-11-28 10:00:00', 'Fratura detectada'),
(4, 'Ultrassonografia', '2024-11-29 13:00:00', 'Normal'),
(5, 'Exame físico', '2024-11-30 09:00:00', 'Normal');
```

3. Consultas (DQL)

Foi realizado diversas consultas para extrair informações valiosas do banco de dados.

Abaixo, destacamos as principais consultas realizadas e a necessidade de usar LEFT JOIN e RIGHT JOIN.

Consultas básicas:

```
-- Listar todos os pacientes e suas respectivas consultas
SELECT P.nome AS paciente, C.data_consulta, C.observacoes
FROM Consultas C
JOIN Pacientes P ON C.id_paciente = P.id_paciente;

-- Listar todos os pacientes sem números de telefone registrados
SELECT * FROM Pacientes
WHERE telefone IS NULL;
```

Usando operadores:

```
-- Encontre todos os médicos com especialidades específicas usando IN
SELECT * FROM Medicos
WHERE especialidade IN ('Cardiologia', 'Pediatria');

-- Mostrar todas as consultas agendadas em um intervalo de datas usando BETWEEN
SELECT * FROM Consultas
WHERE data_consulta BETWEEN '2024-11-25' AND '2024-11-30';

-- Filtrar pacientes cujo nome começa com "A"
SELECT * FROM Pacientes
WHERE nome LIKE 'A%';

-- Usar DISTINCT para listar as especialidades únicas dos médicos
SELECT DISTINCT especialidade FROM Medicos;
```

Funções de Agregação:

```
-- Calcular a idade média dos pacientes
SELECT AVG(DATEDIFF(CURDATE(), data_nascimento)/365) AS idade_media FROM Pacientes;

-- Contar o número total de consultas realizadas
SELECT COUNT(*) AS total_consultas FROM Consultas;

-- Encontrar o valor máximo e mínimo de datas de consulta
SELECT MAX(data_consulta) AS ultima_consulta, MIN(data_consulta) AS primeira_consulta FROM Consultas;

-- Somar o total de exames realizados
SELECT COUNT(*) AS total_exames FROM Exames;
```

Cláusulas Avançadas:

```
-- Listar todos os pacientes ordenados por data de nascimento
SELECT * FROM Pacientes
ORDER BY data_nascimento;

-- Listar o número de consultas por médico (apenas médicos que realizaram mais de uma consulta)
SELECT M.nome, COUNT(*) AS total_consultas
FROM Consultas C
JOIN Medicos M ON C.id_medico = M.id_medico
GROUP BY M.id_medico
HAVING COUNT(*) > 1;
```

5. LEFT JOIN e RIGHT JOIN

```
-- INNER JOIN para obter uma lista de exames com os dados dos paciente
SELECT E.id_exame, E.tipo_exame, E.resultado, P.nome AS paciente, M.nome AS medico
FROM Exames E
JOIN Consultas C ON E.id_consulta = C.id_consulta
JOIN Pacientes P ON C.id_paciente = P.id_paciente
JOIN Medicos M ON C.id_medico = M.id_medico;

-- UNION para listar todos os médicos e pacientes em uma única lista
SELECT nome FROM Medicos
UNION
SELECT nome FROM Pacientes;

-- LEFT JOIN para listar todos os pacientes e suas consultas, incluindo os que não têm consultas
SELECT P.nome AS paciente, C.data_consulta
FROM Pacientes P
LEFT JOIN Consultas C ON P.id_paciente = C.id_paciente;

-- RIGHT JOIN para listar todos os médicos e as consultas realizadas, incluindo os que não realizaram consultas
SELECT M.nome AS medico, C.data_consulta
FROM Medicos M
RIGHT JOIN Consultas C ON M.id_medico = C.id_medico;

-- LEFT JOIN para listar todas as consultas e os exames correspondentes, incluindo as que não têm exames
SELECT C.data_consulta, E.tipo_exame
FROM Consultas C
LEFT JOIN Exames E ON C.id_consulta = E.id_consulta;

-- RIGHT JOIN para listar todos os médicos e os pacientes que eles atendem, incluindo os que não têm pacientes
SELECT M.nome AS medico, P.nome AS paciente
FROM Medicos M
RIGHT JOIN Consultas C ON M.id_medico = C.id_medico
RIGHT JOIN Pacientes P ON C.id_paciente = P.id_paciente;
```

A utilização de JOINS foi essencial para associar dados entre as tabelas, especialmente LEFT JOIN e RIGHT JOIN, que permitiram incluir todos os registros de uma tabela, mesmo quando não há correspondência na outra tabela.

LEFT JOIN: Foi necessário para listar todos os pacientes, incluindo aqueles sem consultas, e todas as consultas, incluindo aquelas sem exames associados.

RIGHT JOIN: Permitiu listar todos os médicos e as consultas realizadas, incluindo médicos que não realizaram consultas.