



FACULDADE SENAI DE TECNOLOGIA GASPAR RICARDO JÚNIOR

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JULIANA FERNANDES DO NASCIMENTO

MILENNA VICTÓRIA ASSIS PORTELLA

BANCO DE DADOS – HOSPITAL VITALIS

PROJETO FINAL – BANCO DE DADOS RELACIONAL

PROFESSOR: ANDRÉ CASSULINO ARAÚJO SOUZA

DISCIPLINA: BANCO DE DADOS

DATA DA ENTREGA: 20/06/2025

SOROCABA-SP

2025

Sumário

1. Introdução
2. Modelagem Conceitual
3. Modelagem Lógica
4. Estrutura do Banco de Dados
5. Manipulação de Dados
6. Controle de Acesso (DCL)
7. Controle de Transações (DTL)
8. Conclusão
9. Referências

1. Introdução

Este projeto tem como objetivo o desenvolvimento de um banco de dados relacional para o **Hospital Vitalis**, um sistema de gestão hospitalar fictício. A proposta visa aplicar conceitos fundamentais de banco de dados, como modelagem, manipulação e segurança dos dados, simulando cenários reais da rotina de uma instituição de saúde.

O sistema foi projetado para atender às demandas de um pronto-socorro 24 horas, abordando desafios como gerenciamento de pacientes, consultas, exames, prescrições e ocupação de leitos.

Objetivo do Projeto:

- Demonstrar como um banco de dados relacional pode gerenciar informações críticas de saúde, oferecendo confiabilidade, integridade e segurança.

Descrição Geral do Banco de Dados:

O banco de dados do Hospital Vitalis contempla entidades como Paciente, Médico, Consulta, Exame, Prontuário, Leito e Departamento, organizando os processos hospitalares de forma eficiente e segura.

2. Modelagem Conceitual

Descrição das Entidades, Atributos e Relacionamentos:

As principais entidades do banco são:

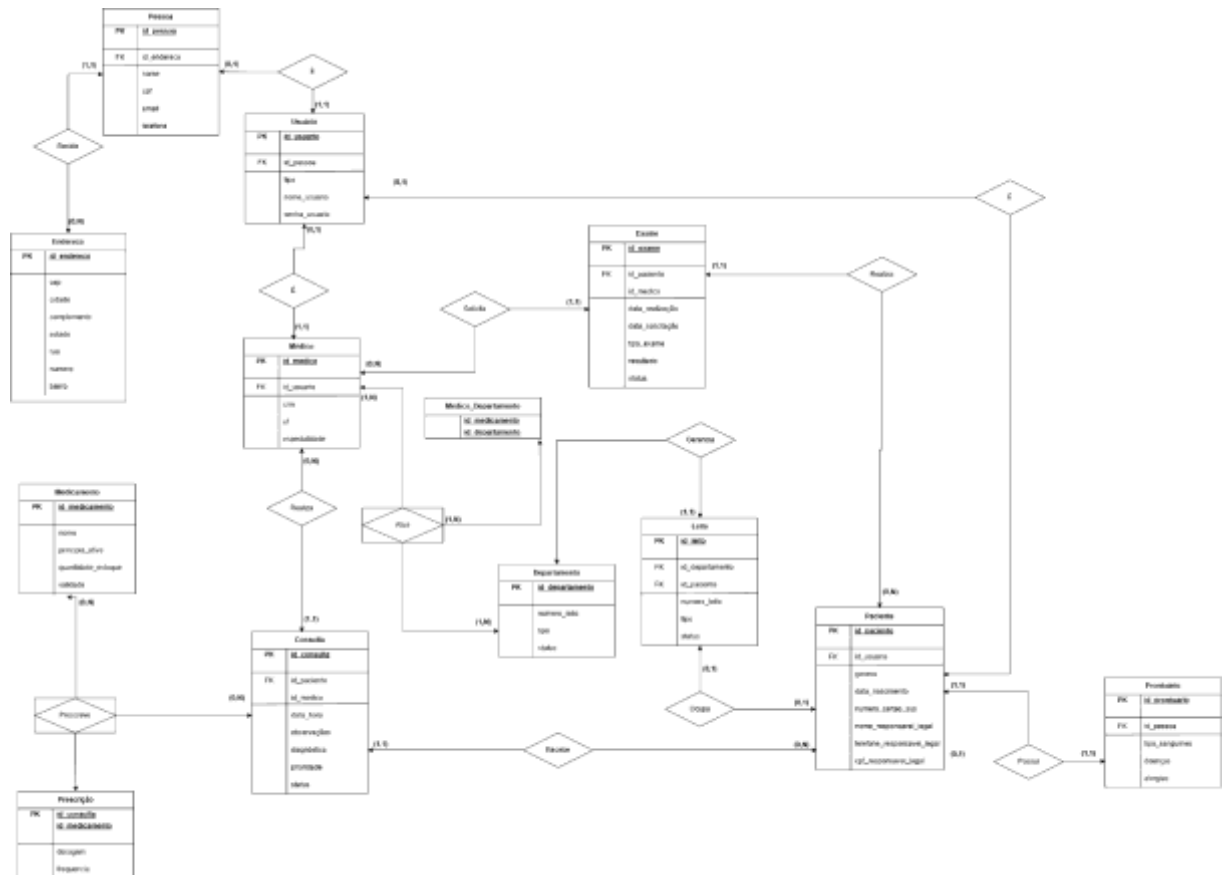
Entidade	Descrição	Atributos
Pessoa	Representa qualquer indivíduo cadastrado no sistema.	id_pessoa, id_endereco, nome, CPF, telefone, e-mail.
Endereço	Informações de localização associadas a uma pessoa.	id_endereco, rua, número, bairro, cidade, estado, CEP, complemento.
Médico	Subtipo de Pessoa que exerce atividades médicas	id_medico, id_pessoa, CRM, UF, especialidade.
Paciente	Subtipo de Pessoa que recebe atendimento médico.	id_paciente, id_pessoa, data_nascimento, gênero, celular, número_cartão_sus, nome_responsável, telefone_responsável, CPF_responsável.
Prontuário	Registro clínico do paciente.	id_prontuario, id_paciente, alergias, doenças, tipo_sanguíneo.
Consulta	Registro de atendimento médico.	id_consulta, id_medico, id_paciente, data_hora, diagnóstico, observações, prioridade, status.
Leito	Controle dos leitos disponíveis.	id_leito, id_departamento, id_paciente, número_leito, tipo, status.

Exame	Exames solicitados e realizados para pacientes.	id_exame, id_paciente, id_medico, data_realizacao, data_solicitacao, tipo_exame, resultado, status.
Medicamento	Controle de estoque e prescrição.	id_medicamento, nome, princípio_ativo, quantidade_estoque, validade.
Usuário	Controle de acesso ao sistema.	: id_usuario, id_pessoa, tipo, nome_usuario, senha_usuario.
Departamento	Setores internos do hospital.	id_departamento, nome, tipo, status.

Tabela de Relacionamentos:

- Associação Médico_Departamento (M:N)
- Associação Prescrição (Consulta x Medicamento) (M:N)

Diagrama Entidade-Relacionamento (DER):



3. Modelagem Lógica

Transformações do DER:

- Conversão de entidades em tabelas com definição de chaves primárias (PK).
- Implementação de chaves estrangeiras (FK) para assegurar integridade referencial.
- Relacionamentos muitos-para-muitos foram desmembrados em tabelas associativas, como Médico_Departamento e Prescrição.

Discussão sobre Normalização:

O banco foi normalizado até a **3ª Forma Normal (3FN)**:

- **1FN:** Eliminação de grupos repetitivos.
- **2FN:** Eliminação de dependências parciais.
- **3FN:** Eliminação de dependências transitivas.

Exemplo:

- Antes: A tabela Consulta tinha um campo “medicamentos_prescritos” como lista.
- Depois: Criação da tabela Prescrição, com uma linha por medicamento vinculado à consulta.

4. Estrutura do Banco de Dados

SGBD Utilizado:

MySQL 8.0+

Exemplo de Script DDL:

```
1  CREATE TABLE Endereco (  
2      Id_endereco INT PRIMARY KEY,  
3      Cep VARCHAR(9),  
4      Complemento VARCHAR(100),  
5      Estado VARCHAR(2),  
6      Rua VARCHAR(100),  
7      Numero VARCHAR(10),  
8      Bairro VARCHAR(50),  
9      Cidade VARCHAR(50)  
10 );  
11  
12 CREATE TABLE Pessoa (  
13     Id_pessoa INT PRIMARY KEY,  
14     Id_endereco INT,  
15     Nome VARCHAR(100),  
16     CPF VARCHAR(14),  
17     Telefone VARCHAR(15),  
18     Email VARCHAR(100),  
19     FOREIGN KEY (Id_endereco) REFERENCES Endereco(Id_endereco)  
20 );
```

Descrição das Tabelas:

Cada tabela representa uma entidade do DER, com campos definidos, tipos de dados, chaves primárias e estrangeiras, além de regras de integridade.

Relacionamentos:

- Tabelas associativas para relacionamentos M:N.
- FK para assegurar vínculo entre registros.

5. Manipulação de Dados (DML e DQL)

Exemplos de Scripts DML:

```
1  #INSERE DADOS
2  INSERT INTO Endereco (Id_endereco, Cep, Complemento, Estado, Rua, Numero, Bairro, Cidade) VALUES
3  (1, '12345-678', 'Apto 101', 'SP', 'Rua das Flores', '100', 'Centro', 'São Paulo'),
4  (2, '23456-789', 'Casa 2', 'RJ', 'Av. Atlântica', '200', 'Copacabana', 'Rio de Janeiro'),
5  (3, '34567-890', 'Bloco B', 'MG', 'Rua da Paz', '300', 'Savassi', 'Belo Horizonte'),
6  (4, '45678-901', '', 'PR', 'Rua das Palmeiras', '400', 'Batel', 'Curitiba'),
7  (5, '56789-012', 'Fundos', 'RS', 'Av. Farrapos', '500', 'Centro', 'Porto Alegre'),
8  (6, '67890-123', 'Apto 305', 'SC', 'Rua das Araucárias', '600', 'Trindade', 'Florianópolis'),
9  (7, '78901-234', '', 'BA', 'Rua do Sol', '700', 'Pelourinho', 'Salvador'),
10 (8, '89012-345', 'Casa 15', 'PE', 'Av. Boa Viagem', '800', 'Boa Viagem', 'Recife'),
11 (9, '90123-456', 'Apto 12', 'DF', 'Setor Comercial Sul', '900', 'Asa Sul', 'Brasília'),
12 (10, '01234-567', '', 'ES', 'Rua da Praia', '1000', 'Praia do Canto', 'Vitória');
13
14 INSERT INTO Pessoa (Id_pessoa, Id_endereco, Nome, CPF, Telefone, Email) VALUES
15 (1, 1, 'João Silva', '123.456.789-00', '(11) 98765-4321', 'joao.silva@email.com'),
16 (2, 2, 'Maria Oliveira', '234.567.890-11', '(21) 98765-4322', 'maria.oliveira@email.com'),
17 (3, 3, 'Pedro Souza', '345.678.901-22', '(31) 98765-4323', 'pedro.souza@email.com'),
18 (4, 4, 'Ana Costa', '456.789.012-33', '(41) 98765-4324', 'ana.costa@email.com'),
19 (5, 5, 'Carlos Lima', '567.890.123-44', '(51) 98765-4325', 'carlos.lima@email.com'),
20 (6, 6, 'Fernanda Alves', '678.901.234-55', '(47) 98765-4326', 'fernanda.alves@email.com'),
21 (7, 7, 'Rafael Santos', '789.012.345-66', '(71) 98765-4327', 'rafael.santos@email.com'),
22 (8, 8, 'Juliana Pereira', '890.123.456-77', '(81) 98765-4328', 'juliana.pereira@email.com'),
23 (9, 9, 'Lucas Mendes', '901.234.567-88', '(61) 98765-4329', 'lucas.mendes@email.com'),
24 (10, 10, 'Beatriz Rocha', '012.345.678-99', '(27) 98765-4330', 'beatriz.rocha@email.com');
```

Consultas (DQL):

```
1  #REVISA O HISTÓRICO DE CONSULTAS DE "JOÃO SILVA"
2  SELECT
3      p.Nome AS Paciente,
4      c.Data_hora,
5      pm.Nome AS Medico,
6      c.Diagnostico,
7      c.Status
8  FROM Pessoa p
9  JOIN Usuario u_p ON p.Id_pessoa = u_p.Id_pessoa
10 JOIN Paciente pa ON u_p.Id_usuario = pa.Id_usuario
11 JOIN Consulta c ON pa.Id_paciente = c.Id_paciente
12 JOIN Medico m ON c.Id_medico = m.Id_medico
13 JOIN Usuario u_m ON m.Id_usuario = u_m.Id_usuario
14 JOIN Pessoa pm ON u_m.Id_pessoa = pm.Id_pessoa
15 WHERE p.Nome = 'João Silva'
16 AND DATE(c.Data_hora) BETWEEN '2025-06-05' AND '2025-06-12'
17 AND (HOUR(c.Data_hora) <= 20 OR DATE(c.Data_hora) < '2025-06-12')
18 ORDER BY c.Data_hora DESC;
```

6. Controle de Acesso (DCL)

Exemplos:

```
1  #UM NOVO MÉDICO, "RAFAEL SANTOS", FOI CONTRATADO E PRECISA DE ACESSO AO SISTEMA PARA REGISTRAR CONSULTAS E EXAMES
2  CREATE USER 'rafael.santos'@'localhost' IDENTIFIED BY 'senha_segura_789';
3  GRANT SELECT, INSERT, UPDATE ON hospital_vitalis.Consulta TO 'rafael.santos'@'localhost';
4  GRANT SELECT, INSERT, UPDATE ON hospital_vitalis.Exame TO 'rafael.santos'@'localhost';
5  GRANT SELECT ON hospital_vitalis.Prontuario TO 'rafael.santos'@'localhost';
6
7  #UMA ENFERMEIRA, "JULIANA PEREIRA" PRECISA GERENCIAR LEITOS E PACIENTES. ASSIM, RECEBER ACESSO PARA ATUALIZAR E CONSULTAR LEITO E PACIENTE.
8  CREATE USER 'juliana.pereira'@'localhost' IDENTIFIED BY 'senha_segura_565';
9  GRANT SELECT, UPDATE ON hospital_vitalis.Leito TO 'juliana.pereira'@'localhost';
10 GRANT SELECT, UPDATE ON hospital_vitalis.Paciente TO 'juliana.pereira'@'localhost';
11
12 #UMA RECEPCIONISTA, "ANA.USER", FOI DEMITIDA E SEU ACESSO AO SISTEMA DEVE SER REVOGADO
13 REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'ana.user'@'localhost';
14 DROP USER 'ana.user'@'localhost';
15
16 #UM MÉDICO, "Pedro Souza", FOI PROMOVIDO A ESPECIALISTA EM PEDIATRIA E AGORA PODE ACESSAR E MODIFICAR DADOS DE PRONTUÁRIOS DE PACIENTES PEDIÁTRICOS
17 CREATE USER 'pedro.souza'@'localhost' IDENTIFIED BY 'pediatra_789';
18 GRANT SELECT, INSERT, UPDATE ON hospital_vitalis.Prontuario TO 'pedro.souza'@'localhost';
19 GRANT SELECT, INSERT, UPDATE ON hospital_vitalis.Consulta TO 'pedro.souza'@'localhost';
20 GRANT SELECT ON hospital_vitalis.Paciente TO 'pedro.souza'@'localhost';
21 FLUSH PRIVILEGES;
```

7. Controle de Transações (DTL)

Exemplos:

```
1  --UM MÉDICO REGISTRA UMA CONSULTA COM PRESCRIÇÃO DE "PARACETAMOL". SE O ESTOQUE FOR INSUFICIENTE, A TRANSAÇÃO É REVERTIDA
2  DELIMITER //
3  CREATE PROCEDURE RegistrarConsultaPresricao()
4  BEGIN
5      START TRANSACTION;
6      INSERT INTO Consulta (Id_consulta, Id_paciente, Id_medico, Data_hora, Diagnostico, Observacoes, Prioridade, Status)
7      VALUES (12, 1, 2, '2025-06-13 19:20:00', 'Febre alta', 'Monitorar temperatura', 'Média', 'Agendada');
8      INSERT INTO Presricao (Id_presricao, Id_consulta, Id_medimento, Dosagem, Frequencia, Duracao)
9      VALUES (13, 12, 2, '500mg', '4x ao dia', '5 dias');
10     IF (SELECT Quantidade_estoque FROM Medicamento WHERE Id_medimento = 2) > 0 THEN
11         COMMIT;
12     ELSE
13         ROLLBACK;
14     END IF;
15 END //
16 DELIMITER ;
17 CALL RegistrarConsultaPresricao();
```

8. Conclusão

O desenvolvimento do banco de dados do **Hospital Vitalis** proporcionou a aplicação prática dos conceitos teóricos de banco de dados, abordando desde a modelagem até o controle de transações e acessos.

Lições Aprendidas:

- Importância da normalização para garantir integridade.
- Validação da integridade referencial com uso de FK.
- Aplicação de DCL e DTL em cenários reais.

Possíveis Melhorias:

- Implementação de procedures e triggers.
- Relatórios via views para melhorar consultas.
- Backup e replicação dos dados.

9. Referências

- DATE, C. J. *Introdução a Sistemas de Banco de Dados*. 8. ed. São Paulo: Pearson, 2004.
- HEUSER, Carlos Alberto. *Projeto de Banco de Dados*. 6. ed. Porto Alegre: Bookman, 2009.
- Documentação oficial MySQL: <https://dev.mysql.com/doc/>
- Material didático da disciplina Banco de Dados.

10. Apêndices

Diagrama Lógico:

