

Portfólio



Database Modeling & SQL

Nome: Polliana Silva

Curso: Análise e Desenvolvimento de Sistemas

Turma: ADS - EAD/2024

Atividade: Portfólio

Sistema de uma Faculdade

Entrega: 04/01/2025

Sumário

1. Descrição do Problema.....	03
2. Objetivos de Aprendizagem.....	03
3. Levantamento de Requisitos.....	03
4. Modelo Conceitual.....	04
5. Modelo Lógico.....	05
6. Modelo Físico.....	05-10
a. Criação do banco e tabelas.....	05-07
b. Inserção de dados.....	08-09
c. Consulta simples.....	09
d. Consultas relacionadas.....	09
e. Atualizações.....	10
f. Exclusões.....	10
7. Conclusão.....	11

1. Descrição do Problema

Um sistema acadêmico para uma faculdade que precisa gerenciar informações sobre alunos, cursos, matérias e professores. O sistema permitirá registrar dados detalhados das matrículas realizadas, armazenando informações dos alunos matriculados, professores e suas respectivas disciplinas, além de controlar as turmas e notas dos alunos.

2. Objetivos de Aprendizagem

Com este estudo de caso, os alunos deverão ser capazes de:

Analisar Requisitos: Trabalhar em conjunto para levantar os requisitos necessários para o sistema.

Modelar Banco de Dados: Criar um modelo *conceitual*, *lógico* e *físico* para armazenar todas as informações necessárias para o sistema.

3. Levantamento de Requisitos

1. Principais necessidades dos clientes:

- Armazenar informações de alunos, professores, cursos e matérias
- Realizar controles básicos, como montar turmas e armazenar notas dos alunos

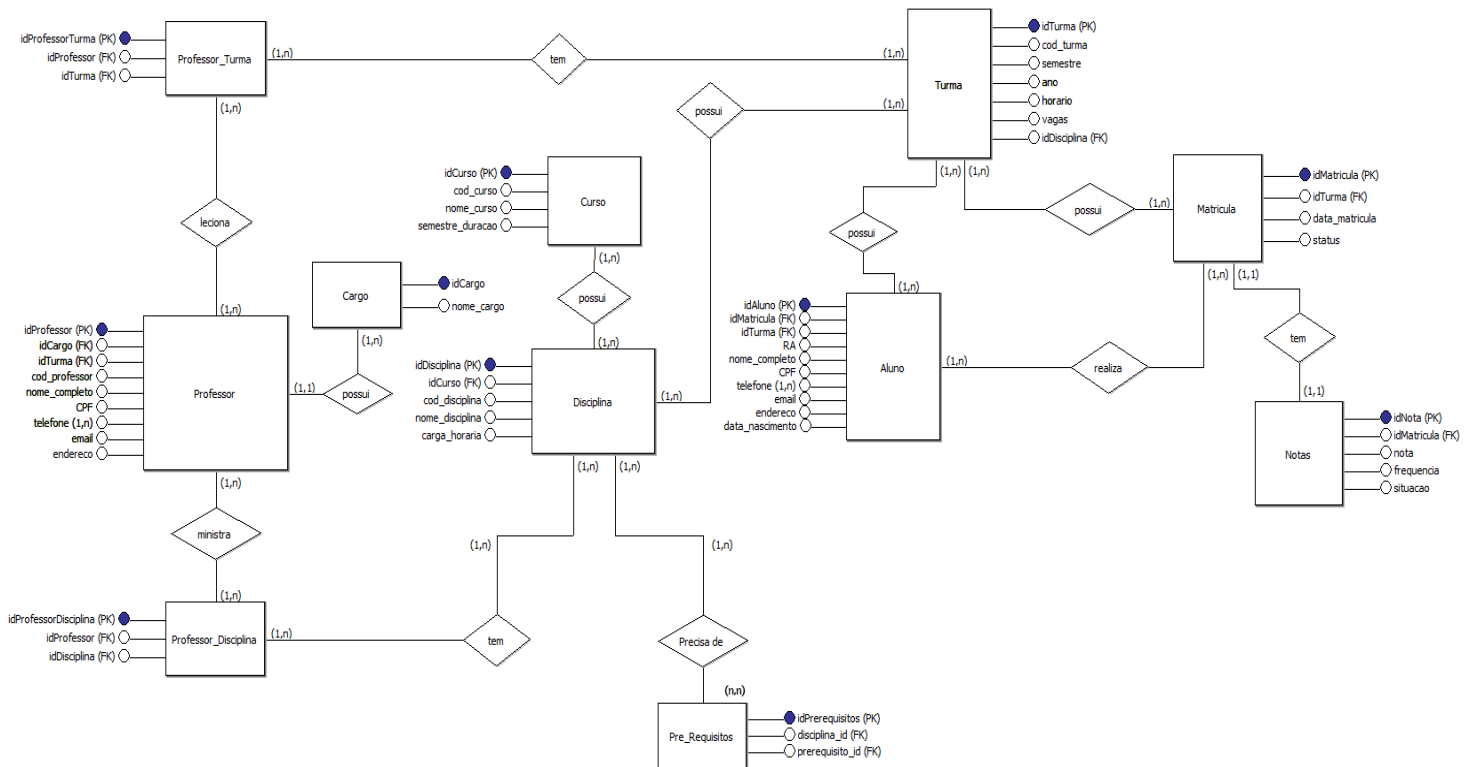
2. Informações a serem armazenadas:

- Dados pessoais de alunos e professores
- Informações sobre cursos, matérias e turmas
- Notas e desempenho dos alunos em cada matéria

3. Utilização dos dados:

- Consultar informações acadêmicas
- Gerenciar a matrícula de alunos em turmas
- Registrar e consultar notas

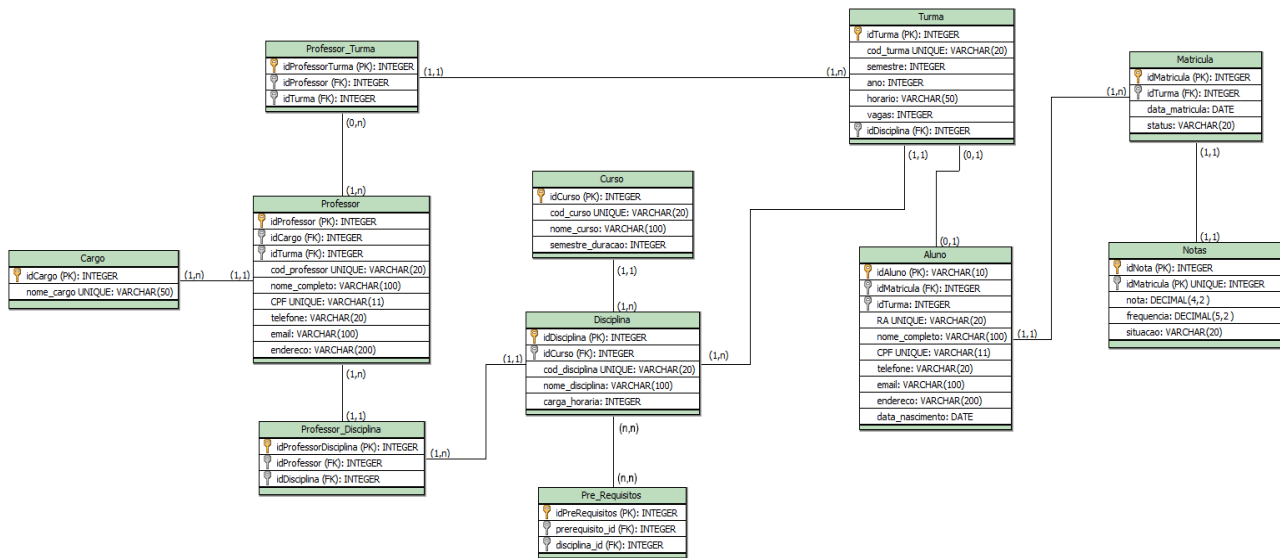
4. Modelo Conceitual



O modelo conceitual representa a estrutura do sistema através de 9 entidades principais (Professor, Aluno, Curso, Disciplina, Turma, Matrícula, Notas, Cargo e Pré-Requisitos) e seus relacionamentos:

- Professores podem lecionar várias turmas e ministrar diferentes disciplinas
- Alunos podem realizar matrículas em diversas turmas
- Cada disciplina pertence a um curso e pode ter pré-requisitos
- Para cada matrícula é registrado um conjunto de notas

5. Modelo Lógico



O modelo lógico define a estrutura das tabelas com seus campos e tipos de dados:

- Chaves primárias (PK) utilizam tipo INTEGER
- Dados textuais utilizam VARCHAR com tamanhos adequados
- Campos únicos (CPF, RA, códigos) possuem constraint UNIQUE
- Relacionamentos implementados via chaves estrangeiras (FK)
- Dados numéricos (notas, frequência) utilizam DECIMAL para precisão

6. Modelo Físico

a. Criação do banco e tabelas:

```

-- Criar e usar o banco de dados
CREATE DATABASE sistema_academico;
USE sistema_academico;
-- Criar tabela de Cargo
CREATE TABLE Cargo (
    idCargo INT AUTO_INCREMENT PRIMARY KEY,
    nome_cargo VARCHAR(50)
);
-- Criar tabela de Professor
CREATE TABLE Professor (

```

```

    idProfessor INT AUTO_INCREMENT PRIMARY KEY,
    idCargo INT,
    cod_professor VARCHAR(20),
    nome_completo VARCHAR(100),
    CPF VARCHAR(11),
    telefone VARCHAR(20),
    email VARCHAR(100),
    endereco VARCHAR(200),
    FOREIGN KEY (idCargo) REFERENCES Cargo(idCargo)
);
-- Criar tabela de Curso
CREATE TABLE Curso (
    idCurso INT AUTO_INCREMENT PRIMARY KEY,
    cod_curso VARCHAR(20),
    nome_curso VARCHAR(100),
    semestre_duracao INT
);
-- Criar tabela de Disciplina
CREATE TABLE Disciplina (
    idDisciplina INT AUTO_INCREMENT PRIMARY KEY,
    idCurso INT,
    cod_disciplina VARCHAR(20),
    nome_disciplina VARCHAR(100),
    carga_horaria INT,
    FOREIGN KEY (idCurso) REFERENCES Curso(idCurso)
);
-- Criar tabela de Turma
CREATE TABLE Turma (
    idTurma INT AUTO_INCREMENT PRIMARY KEY,
    idDisciplina INT,
    cod_turma VARCHAR(20),
    semestre INT,
    ano INT,
    horario VARCHAR(50),
    vagas INT,
    FOREIGN KEY (idDisciplina) REFERENCES Disciplina(idDisciplina)
);
-- Criar tabela de Aluno
CREATE TABLE Aluno (
    idAluno INT AUTO_INCREMENT PRIMARY KEY,
    RA VARCHAR(20),
    nome_completo VARCHAR(100),
    CPF VARCHAR(11),
    telefone VARCHAR(20),
    email VARCHAR(100),
    endereco VARCHAR(200),
    data_nascimento DATE
);

```

```

-- Criar tabela de Matricula
CREATE TABLE Matricula (
    idMatricula INT AUTO_INCREMENT PRIMARY KEY,
    idAluno INT,
    idTurma INT,
    data_matricula DATE,
    status VARCHAR(20),
    FOREIGN KEY (idAluno) REFERENCES Aluno(idAluno),
    FOREIGN KEY (idTurma) REFERENCES Turma(idTurma)
);

-- Criar tabela de Notas
CREATE TABLE Notas (
    idNota INT AUTO_INCREMENT PRIMARY KEY,
    idMatricula INT,
    nota DECIMAL(4,2),
    frequencia DECIMAL(5,2),
    situacao VARCHAR(20),
    FOREIGN KEY (idMatricula) REFERENCES Matricula(idMatricula)
);

-- Criar tabela de Pré-Requisitos
CREATE TABLE Pre_Requisitos (
    idPreRequisitos INT AUTO_INCREMENT PRIMARY KEY,
    disciplina_id INT,
    prerequisite_id INT,
    FOREIGN KEY (disciplina_id) REFERENCES Disciplina(idDisciplina),
    FOREIGN KEY (prerequisite_id) REFERENCES Disciplina(idDisciplina)
);

-- Criar tabela de Professor_Turma
CREATE TABLE Professor_Turma (
    idProfessorTurma INT AUTO_INCREMENT PRIMARY KEY,
    idProfessor INT,
    idTurma INT,
    FOREIGN KEY (idProfessor) REFERENCES Professor(idProfessor),
    FOREIGN KEY (idTurma) REFERENCES Turma(idTurma)
);

-- Criar tabela de Professor_Disciplina
CREATE TABLE Professor_Disciplina (
    idProfessorDisciplina INT AUTO_INCREMENT PRIMARY KEY,
    idProfessor INT,
    idDisciplina INT,
    FOREIGN KEY (idProfessor) REFERENCES Professor(idProfessor),
    FOREIGN KEY (idDisciplina) REFERENCES Disciplina(idDisciplina)
);

-- Ver todas as tabelas
SHOW TABLES;

-- Ver estrutura de uma tabela específica
DESCRIBE Professor;

```

b. Inserção de Dados:

```
-- Inserir cargos
INSERT INTO Cargo (nome_cargo) VALUES
('Docente'),
('Coordenador'),
('Tutor');

-- Inserir professores
INSERT INTO Professor (idCargo, cod_professor, nome_completo, CPF, telefone,
email, endereco) VALUES
(1, 'PROF001', 'João Silva', '12345678901', '11999999991', 'joao@email.com', 'Rua
A, 123'),
(2, 'PROF002', 'Maria Santos', '23456789012', '11999999992', 'maria@email.com',
'Rua B, 456'),
(1, 'PROF003', 'Pedro Souza', '34567890123', '11999999993', 'pedro@email.com',
'Rua C, 789');

-- Inserir cursos
INSERT INTO Curso (cod_curso, nome_curso, semestre_duracao) VALUES
('SI', 'Sistemas de Informação', 8),
('CC', 'Ciência da Computação', 8),
('ADS', 'Análise e Desenvolvimento de Sistemas', 6);

-- Inserir disciplinas
INSERT INTO Disciplina (idCurso, cod_disciplina, nome_disciplina, carga_horaria)
VALUES
(1, 'BD001', 'Banco de Dados', 80),
(1, 'PROG001', 'Programação I', 60),
(2, 'ALG001', 'Algoritmos', 60);

-- Inserir turmas
INSERT INTO Turma (idDisciplina, cod_turma, semestre, ano, horario, vagas) VALUES
(1, 'BD2024A', 1, 2024, 'Segunda 19h-22h', 40),
(2, 'PROG2024A', 1, 2024, 'Terça 19h-22h', 35),
(3, 'ALG2024A', 1, 2024, 'Quarta 19h-22h', 45);

-- Inserir alunos
INSERT INTO Aluno (RA, nome_completo, CPF, telefone, email, endereco,
data_nascimento) VALUES
('RA001', 'Ana Oliveira', '45678901234', '11988888881', 'ana@email.com', 'Rua X,
100', '2000-03-15'),
('RA002', 'Carlos Lima', '56789012345', '11988888882', 'carlos@email.com', 'Rua Y,
200', '2001-06-20'),
('RA003', 'Julia Costa', '67890123456', '11988888883', 'julia@email.com', 'Rua Z,
300', '2002-09-10');

-- Inserir matrículas
INSERT INTO Matricula (idAluno, idTurma, data_matricula, status) VALUES
(1, 1, '2024-01-15', 'Ativa'),
(2, 2, '2024-01-15', 'Ativa'),
(3, 3, '2024-01-15', 'Ativa');

-- Inserir notas
```



```

INSERT INTO Notas (idMatricula, nota, frequencia, situacao) VALUES
(1, 8.5, 90.0, 'Aprovado'),
(2, 7.0, 85.0, 'Aprovado'),
(3, 9.0, 95.0, 'Aprovado');
-- Inserir pré-requisitos
INSERT INTO Pre_Requisitos (disciplina_id, prerequisite_id) VALUES
(2, 1),
(3, 2);
-- Inserir professor_turma
INSERT INTO Professor_Turma (idProfessor, idTurma) VALUES
(1, 1),
(2, 2),
(3, 3);
-- Inserir professor_disciplina
INSERT INTO Professor_Disciplina (idProfessor, idDisciplina) VALUES
(1, 1),
(2, 2),
(3, 3);

```

c. Consulta Simples:

```

-- Ver todos os professores
SELECT * FROM Professor;

-- Ver todas as disciplinas
SELECT * FROM Disciplina;

-- Ver todos os alunos
SELECT * FROM Aluno;

```

d. Consultas Relacionadas:

```

-- Ver alunos e suas notas
SELECT a.nome_completo, d.nome_disciplina, n.nota, n.frequencia
FROM Aluno a
JOIN Matricula m ON a.idAluno = m.idAluno
JOIN Notas n ON m.idMatricula = n.idMatricula
JOIN Turma t ON m.idTurma = t.idTurma
JOIN Disciplina d ON t.idDisciplina = d.idDisciplina;
-- Ver professores e suas disciplinas
SELECT p.nome_completo, d.nome_disciplina
FROM Professor p
JOIN Professor_Disciplina pd ON p.idProfessor = pd.idProfessor
JOIN Disciplina d ON pd.idDisciplina = d.idDisciplina;

```

e. Atualizações:

```
-- Atualizar telefone de um professor
UPDATE Professor
SET telefone = '11999997777'
WHERE idProfessor = 1;

-- Atualizar status de uma matrícula
UPDATE Matricula
SET status = 'Trancada'
WHERE idAluno = 1;

-- Atualizar nota de um aluno
UPDATE Notas
SET nota = 9.5
WHERE idMatricula = 1;
```

f. Exclusões:

```
-- Deletar nota:
DELETE FROM Notas WHERE idNota = 1;

-- Podemos deletar um vínculo de professor com turma
DELETE FROM Professor_Turma
WHERE idProfessorTurma = 1;

-- Ou um vínculo de professor com disciplina
DELETE FROM Professor_Disciplina
WHERE idProfessorDisciplina = 1;

-- Quando tem relacionamento entre tabelas, exemplo:
-- 1º Deletar as notas deste aluno
DELETE FROM Notas
WHERE idMatricula IN (SELECT idMatricula FROM Matricula WHERE idAluno = 3);

-- 2º Deletar as matrículas do aluno
DELETE FROM Matricula
WHERE idAluno = 3;

-- 3º Finalmente, deletar o aluno
DELETE FROM Aluno
WHERE idAluno = 3;
```

7. Conclusão

O script SQL foi desenvolvido para criar a estrutura do banco de dados do sistema acadêmico, seguindo a sequência:

1. Criação do Banco de Dados
 - Criação do banco `sistema_academico`
 - Definição para uso imediato do banco
2. Criação das Tabelas
 - As tabelas são criadas em ordem específica para respeitar as dependências de chaves estrangeiras
 - Cada tabela possui um identificador único (chave primária) com `AUTO_INCREMENT`
 - Campos que não podem ter duplicatas são marcados como `UNIQUE`
 - Relacionamentos são estabelecidos através de chaves estrangeiras (`FOREIGN KEY`)
3. Estrutura das Principais Tabelas:
 - Cargo: Armazena os tipos de cargos dos professores
 - Professor: Dados completos dos professores, vinculado ao cargo
 - Curso: Informações dos cursos oferecidos
 - Disciplina: Cadastro das disciplinas, vinculadas aos cursos
 - Turma: Registro das turmas, vinculadas às disciplinas
 - Aluno: Dados completos dos alunos
 - Matricula: Registro de matrículas, vinculando alunos e turmas
 - Notas: Controle de notas e frequências por matrícula
 - Pre_Requisitos: Gerenciamento de pré-requisitos entre disciplinas
 - Professor_Turma e Professor_Disciplina: Tabelas de relacionamento
4. Exemplos de Dados
 - Inserção de registros básicos nas tabelas Cargo e Curso
 - Demonstração do funcionamento da estrutura do banco

Esta estrutura permite o armazenamento e gerenciamento completo das informações acadêmicas, mantendo a integridade dos dados e os relacionamentos necessários entre as entidades.