

Script SQL - Universidade (Com Comentários)

```
-- CRIAÇÃO DO BANCO DE DADOS
CREATE DATABASE Universidade;
GO
USE Universidade;
GO

-- TABELA DEPARTAMENTO: armazena os departamentos acadêmicos
CREATE TABLE Departamento (
    cod_departamento INT PRIMARY KEY IDENTITY(1,1), -- chave primária auto-incrementável
    nome_departamento VARCHAR(255) NOT NULL -- nome do departamento
);

-- TABELA ENDEREÇO: armazena endereços genéricos para alunos e professores
CREATE TABLE Endereco (
    cod_endereco INT PRIMARY KEY IDENTITY(1,1),
    nome_ rua VARCHAR(255) NOT NULL,
    numero INT NOT NULL,
    complemento VARCHAR(50),
    bairro VARCHAR(100) NOT NULL,
    cidade VARCHAR(100) NOT NULL,
    estado VARCHAR(100) NOT NULL,
    cep VARCHAR(10) NOT NULL
);

-- TABELA CURSO: armazena os cursos da universidade
CREATE TABLE Curso (
    cod_curso INT PRIMARY KEY IDENTITY(1,1),
    nome_curso VARCHAR(255) NOT NULL,
    tipo VARCHAR(255) NOT NULL, -- Bacharelado, Licenciatura, Tecnólogo
    duracao_meses INT NOT NULL,
    cod_departamento INT NOT NULL,
    turno VARCHAR(50) NOT NULL,
    carga_horaria_total INT NOT NULL,
    modalidade VARCHAR(50) NOT NULL,
    FOREIGN KEY (cod_departamento) REFERENCES Departamento(cod_departamento)
);

-- TABELA DISCIPLINA: armazena as disciplinas oferecidas
CREATE TABLE Disciplina (
    cod_disciplina INT PRIMARY KEY IDENTITY(1,1),
    nome_disciplina VARCHAR(255) NOT NULL,
    ementa TEXT NOT NULL,
    cod_departamento INT NOT NULL,
    carga_horaria_total INT NOT NULL,
    FOREIGN KEY (cod_departamento) REFERENCES Departamento(cod_departamento)
);

-- Associativa CURSO_DISCIPLINA: define quais disciplinas pertencem a quais cursos
CREATE TABLE Curso_Disciplina (
    cod_curso INT NOT NULL,
    cod_disciplina INT NOT NULL,
    tipo_disciplina VARCHAR(20) NOT NULL, -- obrigatória, optativa
    PRIMARY KEY (cod_curso, cod_disciplina),
    FOREIGN KEY (cod_curso) REFERENCES Curso(cod_curso),
    FOREIGN KEY (cod_disciplina) REFERENCES Disciplina(cod_disciplina)
);
```

Script SQL - Universidade (Com Comentários)

```
-- Pré-requisitos de disciplinas
CREATE TABLE Disciplina_PreRequisito (
    cod_disciplina INT NOT NULL,
    cod_pre_requisito INT NOT NULL,
    PRIMARY KEY (cod_disciplina, cod_pre_requisito),
    FOREIGN KEY (cod_disciplina) REFERENCES Disciplina(cod_disciplina),
    FOREIGN KEY (cod_pre_requisito) REFERENCES Disciplina(cod_disciplina)
);

-- ALUNO: cadastro completo dos alunos
CREATE TABLE Aluno (
    ra_aluno INT PRIMARY KEY,
    nome_aluno VARCHAR(255) NOT NULL,
    sobrenome_aluno VARCHAR(255) NOT NULL,
    data_nascimento DATE NOT NULL,
    identificacao_genero VARCHAR(50),
    cpf VARCHAR(14) NOT NULL UNIQUE,
    rg VARCHAR(20) NOT NULL UNIQUE,
    nacionalidade VARCHAR(100) NOT NULL,
    estado_civil VARCHAR(50) NOT NULL,
    naturalidade VARCHAR(100) NOT NULL,
    deficiencia VARCHAR(255) NOT NULL,
    data_ingresso DATE NOT NULL,
    email_pessoal VARCHAR(255),
    email_corporativo VARCHAR(255),
    status_aluno VARCHAR(50) NOT NULL,
    nome_mae VARCHAR(255) NOT NULL,
    nome_pai VARCHAR(255),
    cod_endereco INT NOT NULL,
    cod_curso INT NOT NULL,
    FOREIGN KEY (cod_endereco) REFERENCES Endereco(cod_endereco),
    FOREIGN KEY (cod_curso) REFERENCES Curso(cod_curso)
);

-- PROFESSOR: cadastro completo dos professores
CREATE TABLE Professor (
    cod_professor INT PRIMARY KEY IDENTITY(1,1),
    nome_professor VARCHAR(255) NOT NULL,
    sobrenome_professor VARCHAR(255) NOT NULL,
    data_nascimento DATE NOT NULL,
    identificacao_genero VARCHAR(50),
    data_contratacao DATE NOT NULL,
    tipo_vinculo VARCHAR(50) NOT NULL,
    cpf VARCHAR(14) NOT NULL UNIQUE,
    rg VARCHAR(20) NOT NULL UNIQUE,
    titulacao VARCHAR(50) NOT NULL,
    email_pessoal VARCHAR(255),
    email_corporativo VARCHAR(255),
    cod_departamento INT NOT NULL,
    status_professor VARCHAR(50) NOT NULL,
    cod_endereco INT NOT NULL,
    FOREIGN KEY (cod_departamento) REFERENCES Departamento(cod_departamento),
    FOREIGN KEY (cod_endereco) REFERENCES Endereco(cod_endereco)
);

-- TELEFONES ALUNO
```

Script SQL - Universidade (Com Comentários)

```
CREATE TABLE Telefone_Aluno (
    cod_telefone INT PRIMARY KEY,
    ra_aluno INT NOT NULL,
    num_telefone VARCHAR(15) NOT NULL,
    tipo_telefone VARCHAR(30) NOT NULL,
    FOREIGN KEY (ra_aluno) REFERENCES Aluno(ra_aluno)
);

-- TELEFONES PROFESSOR
CREATE TABLE Telefone_Professor (
    cod_telefone_professor INT PRIMARY KEY,
    cod_professor INT NOT NULL,
    num_telefone VARCHAR(15) NOT NULL,
    tipo_telefone VARCHAR(30) NOT NULL,
    FOREIGN KEY (cod_professor) REFERENCES Professor(cod_professor)
);

-- PROFESSOR_DISCIPLINA: associativa
CREATE TABLE Professor_Disciplina (
    cod_professor INT NOT NULL,
    cod_disciplina INT NOT NULL,
    PRIMARY KEY (cod_professor, cod_disciplina),
    FOREIGN KEY (cod_professor) REFERENCES Professor(cod_professor),
    FOREIGN KEY (cod_disciplina) REFERENCES Disciplina(cod_disciplina)
);

-- TURMAS: define as turmas
CREATE TABLE Turma (
    cod_turma INT PRIMARY KEY IDENTITY(1,1),
    cod_curso INT NOT NULL,
    sala VARCHAR(50) NOT NULL,
    ano INT NOT NULL,
    semestre INT NOT NULL,
    cod_disciplina INT NOT NULL,
    cod_professor INT NOT NULL,
    periodo VARCHAR(50) NOT NULL,
    numero_alunos INT NOT NULL,
    data_inicio DATE NOT NULL,
    data_fim DATE,
    FOREIGN KEY (cod_curso) REFERENCES Curso(cod_curso),
    FOREIGN KEY (cod_disciplina) REFERENCES Disciplina(cod_disciplina),
    FOREIGN KEY (cod_professor) REFERENCES Professor(cod_professor)
);

-- MATRÍCULA (atualizada)
CREATE TABLE Matricula (
    cod_matricula INT PRIMARY KEY IDENTITY(1,1),
    ra_aluno INT NOT NULL,
    cod_turma INT NOT NULL,
    data_matricula DATE NOT NULL,
    data_cancelamento DATE,
    motivo_cancelamento TEXT,
    status_matricula VARCHAR(50) NOT NULL,
    FOREIGN KEY (ra_aluno) REFERENCES Aluno(ra_aluno),
    FOREIGN KEY (cod_turma) REFERENCES Turma(cod_turma)
);
```

Script SQL - Universidade (Com Comentários)

```
-- HISTÓRICO ESCOLAR
CREATE TABLE Historico (
    cod_disciplina INT NOT NULL,
    ra_aluno INT NOT NULL,
    ano INT NOT NULL,
    semestre INT NOT NULL,
    nota DECIMAL(4,2) NOT NULL,
    frequencia INT NOT NULL,
    situacao_historico VARCHAR(50) NOT NULL,
    PRIMARY KEY (cod_disciplina, ra_aluno, ano, semestre),
    FOREIGN KEY (cod_disciplina) REFERENCES Disciplina(cod_disciplina),
    FOREIGN KEY (ra_aluno) REFERENCES Aluno(ra_aluno)
);
```