

PROJETO FINAL DE MODELAGEM DE DADO

ENTENDENDO O PROJETO

Um aluno só pode estar matriculado em um curso por vez. Alunos possuem código de identificação (RA). Cursos são compostos por disciplinas. Cada disciplina terá no máximo 30 alunos por turma. As disciplinas podem ser obrigatórias ou optativas. Disciplinas pertencem a departamentos específicos. Cada disciplina possui um código de identificação.

Um aluno pode trancar a matrícula, não estando matriculado em nenhuma disciplina no semestre. Um aluno pode se matricular em no máximo 9 disciplinas por semestre. Um aluno pode ser reprovado no máximo 3 vezes. A faculdade terá no máximo 3.000 alunos em 10 cursos distintos. Entram 300 alunos por ano. Existem no máximo 90 disciplinas.

O histórico escolar deve armazenar as disciplinas cursadas, a nota final, a frequência e o período do curso realizado. Professores podem ser cadastrados mesmo sem lecionar disciplinas. A escola possui 40 professores. Um professor leciona no máximo 4 disciplinas. Cada professor pertence a no máximo um departamento. Professores possuem código de identificação.

Identificando Entidades, Atributos e Relacionamentos

A) Identificando Entidades

- Aluno - Professor
- Disciplina
- Curso
- Departamento

B) Identificando Relacionamentos

- Aluno está matriculado em curso
- Aluno cursa disciplina
- Aluno realizou disciplina
- Disciplina pertence a curso
- Professor ministra disciplina
- Professor pertence a departamento
- Departamento é responsável por disciplina
- Departamento controla curso
- Disciplina depende de disciplina

C) Identificando Atributos

- Aluno
- Número da Matrícula

- Nome
- Sobrenome
- Endereço : Rua Número Bairro CEP Cidade Estado
- Código do Curso

D) Identificando Atributos - Professor

- Código do Professor
- Nome
- Sobrenome
- Código do departamento

E) Identificando Atributos

- Disciplina
- Código da Disciplina
- Nome da Disciplina
- Descrição Curricular
- Código do departamento
- Número de Alunos
- Código do Departamento

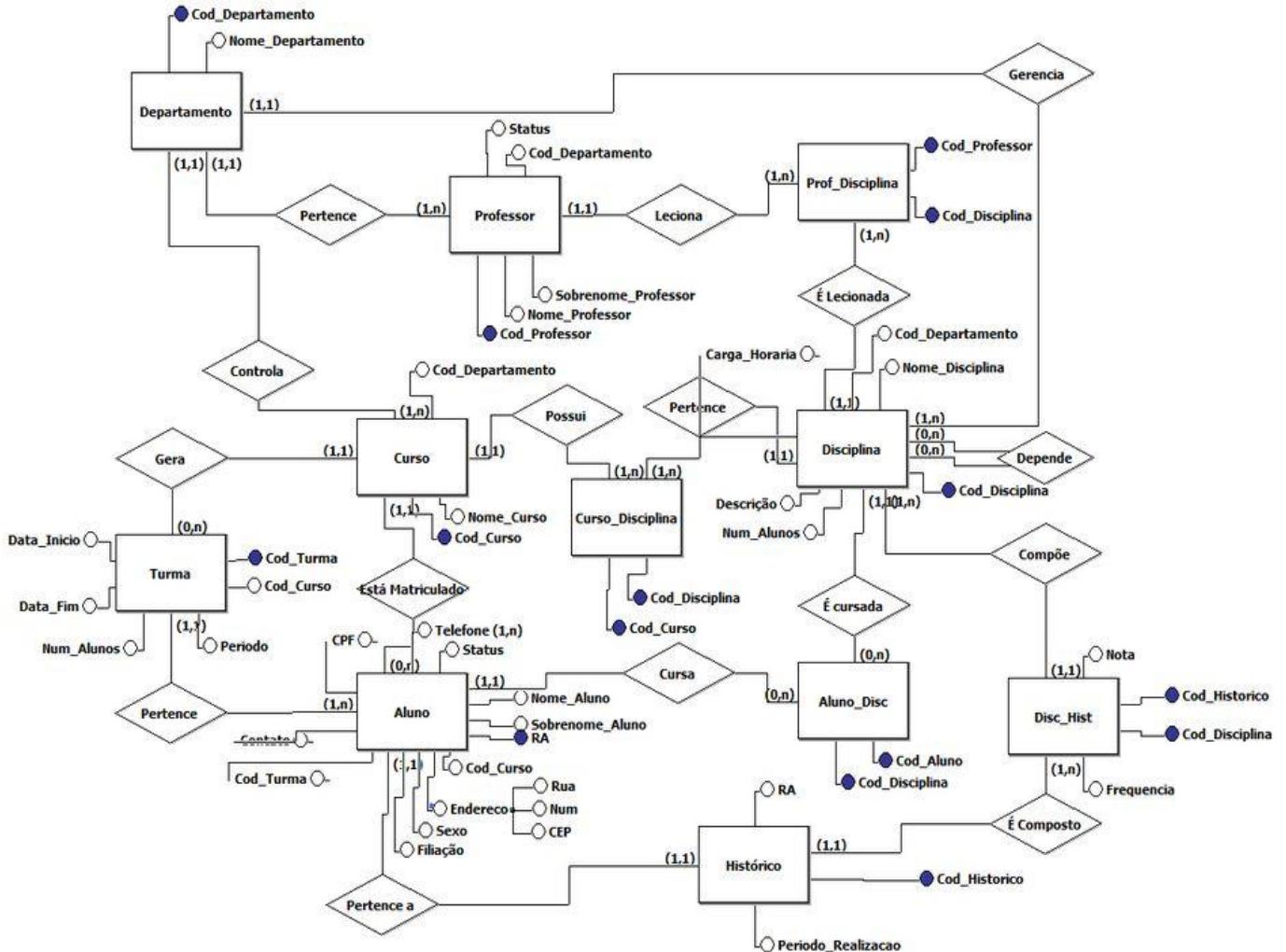
F) Identificando Atributos

- Curso
- Código do Curso
- Nome do Curso
- Código do Departamento

G) Identificando Atributos

- Departamento
- Código do departamento
- Nome do Departamento

Diagrama Entidade-Relacionamento



```
-- Criar Banco CREATE DATABASE db_Faculdade; USE db_Faculdade;
```

```
-- Criar Tabelas CREATE TABLE Departamento ( Cod_Departamento INT PRIMARY KEY AUTO_INCREMENT, Nome_departamento VARCHAR(20) NOT NULL );
```

```
CREATE TABLE Professor ( Cod_Professor INT PRIMARY KEY AUTO_INCREMENT, Nome_Professor VARCHAR(20) NOT NULL, Sobrenome_Professor VARCHAR(50) NOT NULL, Status_Professor TINYINT, Cod_Departamento INT, CONSTRAINT fk_Codigo_departamento FOREIGN KEY(Cod_Departamento) REFERENCES Departamento (Cod_Departamento) );
```

```
CREATE TABLE Curso ( Cod_Curso INT PRIMARY KEY AUTO_INCREMENT, Nome_Curso VARCHAR(30), Cod_Departamento INT, CONSTRAINT fk_Cod_Departamento FOREIGN KEY (Cod_Departamento) REFERENCES Departamento (Cod_Departamento) );
```

```
CREATE TABLE Turma ( Cod_Turma INT PRIMARY KEY AUTO_INCREMENT, Cod_Curso INT, Periodo VARCHAR(8), Num_Alunos INT, Data_Inicio DATE, Data_Fim DATE, CONSTRAINT fk_Cod_Curso FOREIGN KEY (Cod_Curso) REFERENCES Curso (Cod_Curso) );
```

```
CREATE TABLE Disciplina ( Cod_Disciplina INT PRIMARY KEY AUTO_INCREMENT, Cod_Disciplina_Depende INT NULL /* Auto-relacionamento */, Nome_Disciplina VARCHAR(30), Cod_Departamento INT NOT NULL, Carga_Horaria INT NOT NULL, Descrição VARCHAR(80),
```

Num_Alunos INT NOT NULL, CONSTRAINT fk_Cod_Departamento_Disciplina FOREIGN KEY
(Cod_Departamento) REFERENCES Departamento (Cod_Departamento), CONSTRAINT
fk_Cod_Disciplina FOREIGN KEY (Cod_Disciplina_Depende) REFERENCES Disciplina
(Cod_Disciplina);

CREATE TABLE Prof_Disciplina (Cod_Professor INT NOT NULL, Cod_Disciplina INT NOT NULL,
PRIMARY KEY (Cod_Professor, Cod_Disciplina), CONSTRAINT fk_Cod_Professor_Prof FOREIGN KEY
(Cod_Professor) REFERENCES Professor (Cod_Professor), CONSTRAINT fk_Cod_Disciplina_Prof
FOREIGN KEY (Cod_Disciplina) REFERENCES Disciplina (Cod_Disciplina));

CREATE TABLE Curso_Disciplina (Cod_Curso INT NOT NULL, Cod_Disciplina INT NOT NULL,
PRIMARY KEY (Cod_Curso, Cod_Disciplina), CONSTRAINT fk_Cod_Curso_Disci FOREIGN KEY
(Cod_Curso) REFERENCES Curso (Cod_Curso), CONSTRAINT fk_Cod_Disciplina_Disci FOREIGN KEY
(Cod_Disciplina) REFERENCES Disciplina (Cod_Disciplina));

CREATE TABLE Aluno (RA INT PRIMARY KEY AUTO_INCREMENT, Nome_Aluno VARCHAR(20) NOT
NULL, Sobrenome_Aluno VARCHAR(20) NOT NULL, CPF VARCHAR(11) NOT NULL, Status_Aluno
TINYINT(1) NOT NULL, Cod_Turma INT, Sexo VARCHAR(1), Cod_Curso INT, Nome_Pai VARCHAR(50)
NOT NULL, Nome_Mae VARCHAR(50) NOT NULL, Email VARCHAR(50) NOT NULL, Whatsapp
VARCHAR(20) NOT NULL, CONSTRAINT fk_Cod_Turma_Aluno FOREIGN KEY(Cod_Turma)
REFERENCES Turma (Cod_Turma), CONSTRAINT fk_Cod_Curso_Aluno FOREIGN KEY(Cod_Curso)
REFERENCES Curso (Cod_Curso));

CREATE TABLE Aluno_Disc (RA INT NOT NULL, Cod_Disciplina INT NOT NULL, PRIMARY KEY (RA,
Cod_Disciplina), CONSTRAINT fk_RA_Aluno FOREIGN KEY (RA) REFERENCES Aluno (RA),
CONSTRAINT fk_Cod_Disciplina_Aluno FOREIGN KEY (Cod_Disciplina) REFERENCES Disciplina
(Cod_Disciplina));

CREATE TABLE Historico (Cod_Historico INT PRIMARY KEY AUTO_INCREMENT, RA INT NOT NULL,
Data_Inicio DATE NOT NULL, Data_Final DATE, CONSTRAINT fk_Cod_RA FOREIGN KEY (RA)
REFERENCES Aluno (RA));

CREATE TABLE Disc_Hist (Cod_Historico INT NOT NULL, Cod_Disciplina INT NOT NULL, Nota INT,
Frequencia INT, PRIMARY KEY (Cod_Historico, Cod_Disciplina), CONSTRAINT fk_Cod_Historico
FOREIGN KEY (Cod_Historico) REFERENCES Historico (Cod_Historico), CONSTRAINT
fk_Cod_Disciplina_Hist FOREIGN KEY (Cod_Disciplina) REFERENCES Disciplina (Cod_Disciplina));

CREATE TABLE Tipo_Telefone(Cod.Tipo_Telefone INT PRIMARY KEY AUTO_INCREMENT,
Tipo_Telefone VARCHAR(8));

CREATE TABLE Telefones_Aluno (Cod_Telefones_Aluno INT PRIMARY KEY AUTO_INCREMENT, RA INT
NOT NULL, Cod.Tipo_Telefone INT NOT NULL, Num_Telefone VARCHAR(20) NOT NULL,
CONSTRAINT fk_Cod_RA_Tel FOREIGN KEY (RA) REFERENCES Aluno (RA), CONSTRAINT
fk_Cod_Tipo_Telefone FOREIGN KEY (Cod.Tipo_Telefone) REFERENCES Tipo_Telefone
(Cod.Tipo_Telefone));

CREATE TABLE Tipo_Logradouro (Cod.Tipo_Logradouro INT PRIMARY KEY AUTO_INCREMENT,
Tipo_Logradouro VARCHAR(11));

```
CREATE TABLE Endereco_Aluno ( Cod_Endereco_Aluno INT PRIMARY KEY AUTO_INCREMENT, RA INT NOT NULL, Cod_Tipo_Logradouro INT NOT NULL, Nome_Rua VARCHAR(50) NOT NULL, Num_Rua INT NOT NULL, Complemento VARCHAR(20) NULL, CEP VARCHAR(8) NOT NULL, CONSTRAINT fk_Cod_RA_End FOREIGN KEY(RA) REFERENCES Aluno (RA), CONSTRAINT fk_Cod_Tipo_Lougradouro FOREIGN KEY(Cod_Tipo_Logradouro) REFERENCES Tipo_Logradouro (Cod_Tipo_Logradouro );
```

```
-- Carga de Dados para Testes INSERT INTO Departamento (Nome_Departamento) VALUES ('Ciências Humanas'), ('Matemática'), ('Biológicas'), ('Estágio');
```

```
INSERT INTO Professor (Nome_Professor, Sobrenome_Professor, Status_Professor, Cod_Departamento) VALUES ('Fábio', 'dos Reis', 0, 2), ('Sophie', 'Allemand', 1, 1), ('Monica', 'Barroso', 1, 3);
```

```
INSERT INTO Curso (Nome_Curso, Cod_Departamento) VALUES ('Matemática', 2), ('Psicologia', 1), ('Análise de Sistemas', 2), ('Biologia', 3), ('História', 1), ('Engenharia', 2);
```

```
INSERT INTO Turma (Cod_Curso, Periodo, Num_Alunos, Data_Inicio, Data_Fim) VALUES (2, 'Manhã', 20, '2016-05-12', '2017-10-15'), (1, 'Noite', 10, '2014-05-12', '2020-03-05'), (3, 'Tarde', 15, '2012-05-12', '2014-05-10');
```

```
INSERT INTO Disciplina (Nome_Disciplina, Cod_Departamento, Carga_Horaria, Descrição, Num_Alunos) VALUES ('Raciocínio Lógico', 2, 1200, 'Desenvolver o raciocínio lógico', 50), ('Psicologia Cognitiva', 1, 1400, 'Entender o funcionamento do aprendizado', 30), ('Programação em C', 2, 1200, 'Aprender uma linguagem de programação', 20), ('Eletrônica Digital', 2, 300, 'Funcionamento de circuitos digitais', 30);
```

```
INSERT INTO Aluno (Nome_Aluno, Sobrenome_Aluno, CPF, Status_Aluno, Cod_Turma, Sexo, Cod_Curso, Nome_Pai, Nome_Mae, Email, Whatsapp) VALUES ('Marcos', 'Aurelio Martins', 14278914536, 1, 2, 'M', 3, 'Marcio Aurelio', 'Maria Aparecida', 'marcosaurelio@gmail.com', 946231249), ('Gabriel', 'Fernando de Almeida', 14470954536, 1, 1, 'M', 1, 'Adão Almeida', 'Fernanda Almeida', 'gabrielalmeida@yahoo.com', 941741247), ('Beatriz', 'Sonia Meneguel', 1520984537, 1, 3, 'F', 3, 'Samuel Meneguel', 'Gabriella Meneguel', 'batrizmene@hotmail.com', 945781412), ('Jorge', 'Soares', 14223651562, 1, 3, 'M', 4, 'João Soares', 'Maria Richter', 'jorgesoares@gmail.com', 925637857), ('Ana Paula', 'Ferretti', 32968914522, 1, 3, 'F', 5, 'Marcio Ferretti', 'Ana Hoffbahn', 'anapaulaferretti@hotmail.com', 974267423), ('Mônica', 'Yamaguti', 32988914510, 1, 2, 'F', 6, 'Wilson Oliveira', 'Fernanda Yamaguti', 'monyamaguti@outlook.com', 932619560);
```

```
INSERT INTO Aluno_Disc (RA, Cod_Disciplina) VALUES (3, 1), (1, 2), (2, 3), (4, 3), (5, 4), (6, 1);
```

```
INSERT INTO Curso_Disciplina(Cod_Curso, Cod_Disciplina) VALUES (1, 1), (2, 2), (3, 3), (6, 4);
```

```
INSERT INTO Prof_Disciplina(Cod_Professor, Cod_Disciplina) VALUES (2, 1), (1, 2), (3, 3), (2, 4);
```

```
INSERT INTO Historico (RA, Data_Inicio, Data_Final) VALUES (2, '2016-05-12', '2017-10-15'), (3, '2014-05-12', '2020-03-05'), (1, '2010-05-12', '2012-05-10');
```

```
INSERT INTO Tipo_Logradouro (Tipo_Logradouro) VALUES ('Rua'), ('Avenida'), ('Alameda'), ('Travessa');
```

```
INSERT INTO Endereco_Aluno (RA, Cod_Tipo_Logradouro, Nome_Rua, Num_Rua, Complemento, CEP) VALUES (2, 1, 'das Giestas', 255, 'Casa 02', 02854000), (3, 3, 'Lorena', 10, 'Apto 15', 02945000), (1, 2, 'do Cursino', 1248, "", 0851040), (4, 1, 'das Heras', 495, "", 03563142), (5, 3, 'Santos', 1856, "", 04523963), (6, 4, 'Matão', 206, "", 04213650);
```

-- CONSULTAS PARA MANIPULAÇÃO DOS DADOS

-- 1Listar todos os alunos SELECT * FROM Aluno;

-- Listar alunos com seus respectivos cursos SELECT A.Nome_Aluno, A.Sobrenome_Aluno, C.Nome_Curso FROM Aluno A JOIN Curso C ON A.Cod_Curso = C.Cod_Curso;

-- Listar professores e seus departamentos SELECT P.Nome_Professor, P.Sobrenome_Professor, D.Nome_Departamento FROM Professor P JOIN Departamento D ON P.Cod_Departamento = D.Cod_Departamento;

-- Listar disciplinas com seus departamentos SELECT Di.Nome_Disciplina, D.Nome_Departamento FROM Disciplina Di JOIN Departamento D ON Di.Cod_Departamento = D.Cod_Departamento;

-- Mostrar quais alunos fazem quais disciplinas SELECT A.Nome_Aluno, Di.Nome_Disciplina FROM Aluno A JOIN Aluno_Disc AD ON A.RA = AD.RA JOIN Disciplina Di ON AD.Cod_Disciplina = Di.Cod_Disciplina;

-- Mostrar professores e as disciplinas que lecionam SELECT P.Nome_Professor, Di.Nome_Disciplina FROM Professor P JOIN Prof_Disciplina PD ON P.Cod_Professor = PD.Cod_Professor JOIN Disciplina Di ON PD.Cod_Disciplina = Di.Cod_Disciplina;

-- Mostrar cursos e suas disciplinas SELECT C.Nome_Curso, Di.Nome_Disciplina FROM Curso C JOIN Curso_Disciplina CD ON C.Cod_Curso = CD.Cod_Curso JOIN Disciplina Di ON CD.Cod_Disciplina = Di.Cod_Disciplina;

-- Mostrar alunos com seus endereços completos SELECT A.Nome_Aluno, TL.Tipo_Logradouro, E.Nome_Rua, E.Num_Rua, E.CEP FROM Aluno A JOIN Endereco_Aluno E ON A.RA = E.RA JOIN Tipo_Logradouro TL ON E.Cod_Tipo_Logradouro = TL.Cod_Tipo_Logradouro;

-- Mostrar histórico dos alunos SELECT A.Nome_Aluno, H.Data_Inicio, H.Data_Final FROM Aluno A JOIN Historico H ON A.RA = H.RA;

-- Contar quantos alunos existem por curso SELECT C.Nome_Curso, COUNT(A.RA) AS Total_Alunos FROM Curso C LEFT JOIN Aluno A ON C.Cod_Curso = A.Cod_Curso GROUP BY C.Nome_Curso;