

# 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

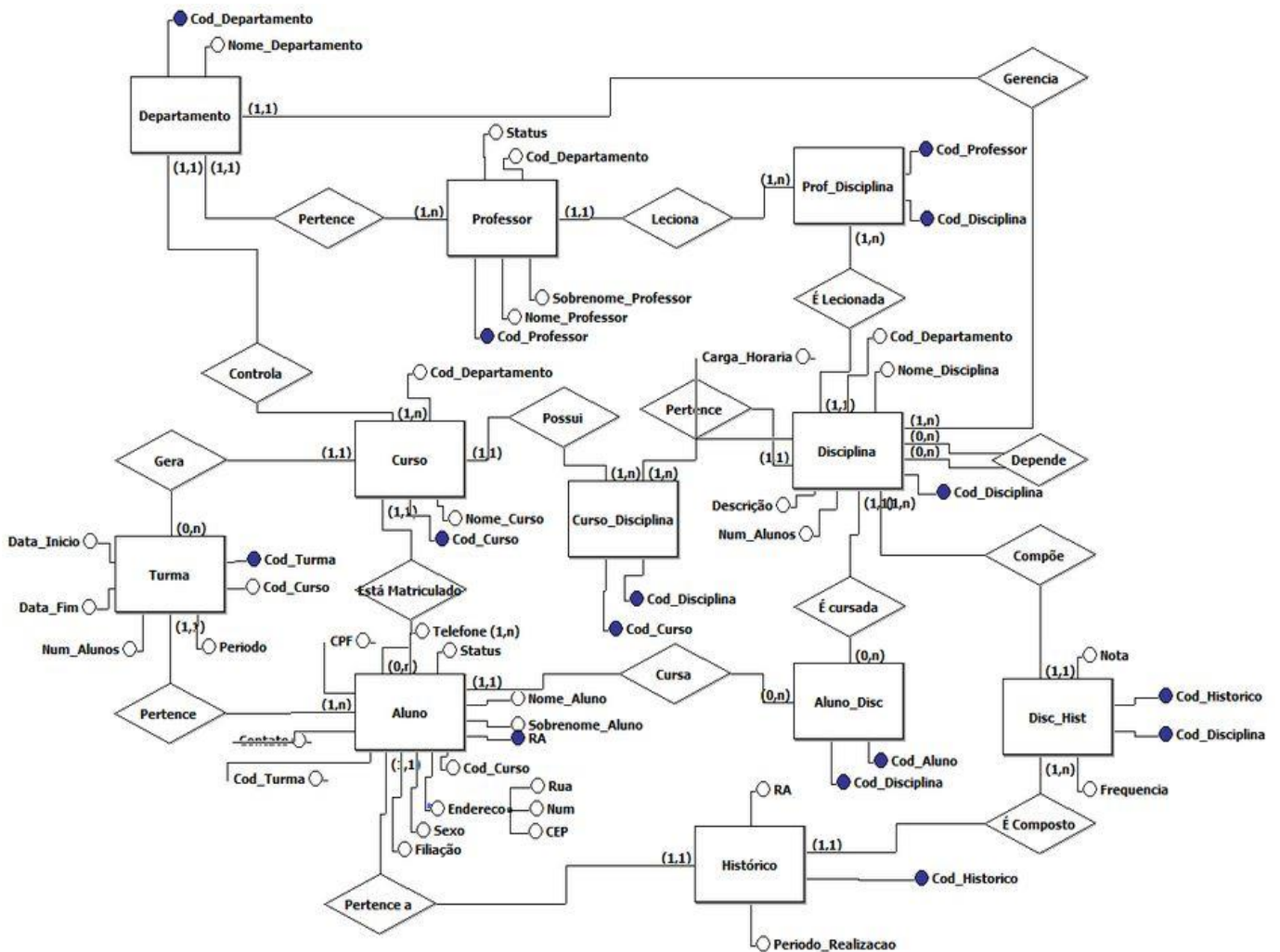
#### F) Identificando Atributos

- 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, Período 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, Frequência 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_Logradouro 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

-- 1 Listar 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;