BANCO DE DADOS

Trabalho – Relatório

Curso:	Ciência de Dados
Aluno(a):	Ryan Pablo Parode Máximo
RU:	3746648

1. 1ª Etapa – Modelagem

Pontuação: 25 pontos.

Dado o estudo de caso abaixo, elabore o Modelo Entidade-Relacionamento (MER), isto é, o modelo conceitual.

O Modelo Entidade-Relacionamento (MER) deve contemplar os seguintes itens:

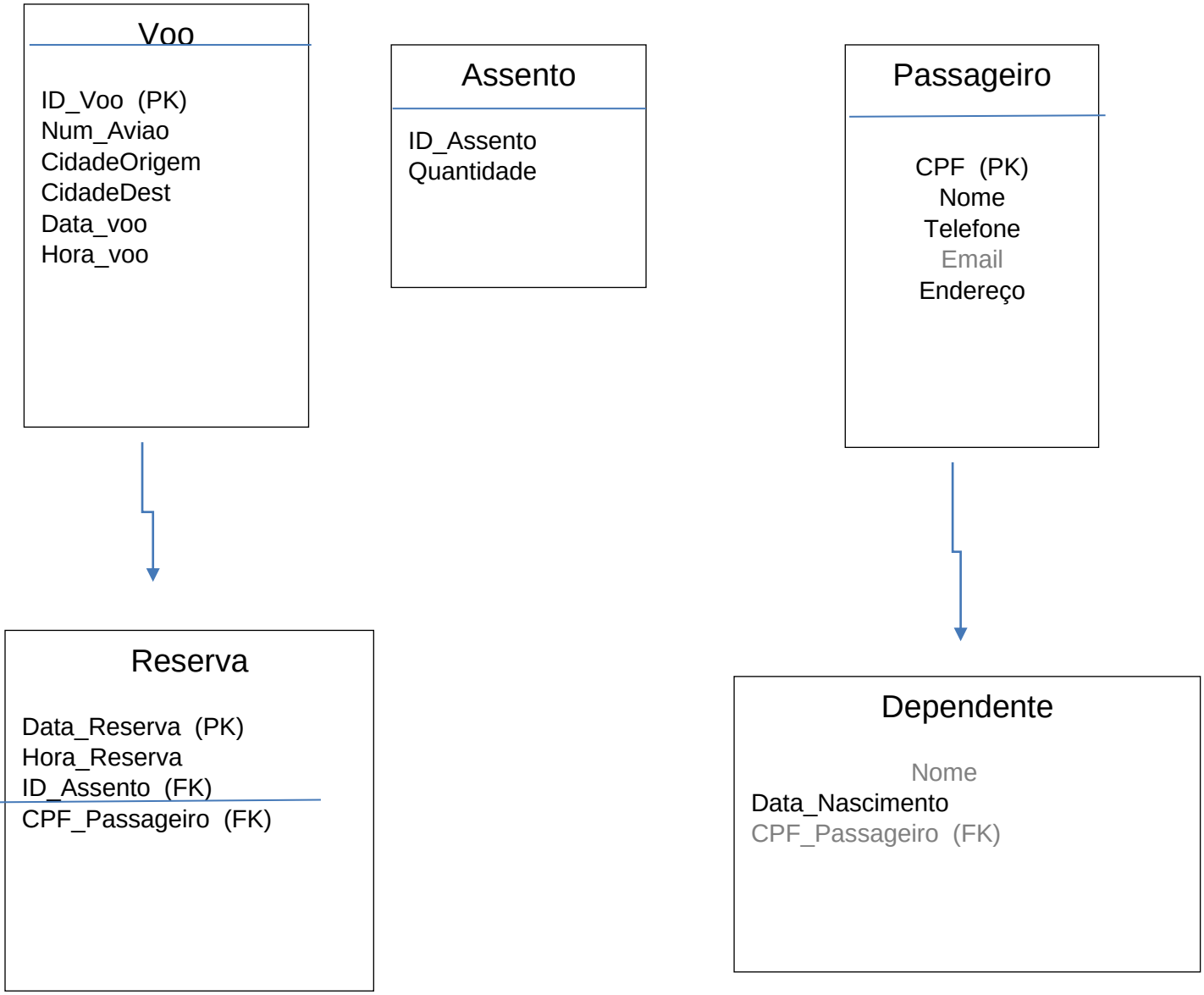
- Entidades;
- Atributos;
- Relacionamentos;
- Cardinalidades.

Uma companhia aérea necessita controlar os dados de seus voos. Para isso, contratou um profissional de Banco de Dados, a fim de modelar o Banco de Dados que armazenará os dados dos voos.

As regras de negócio são:

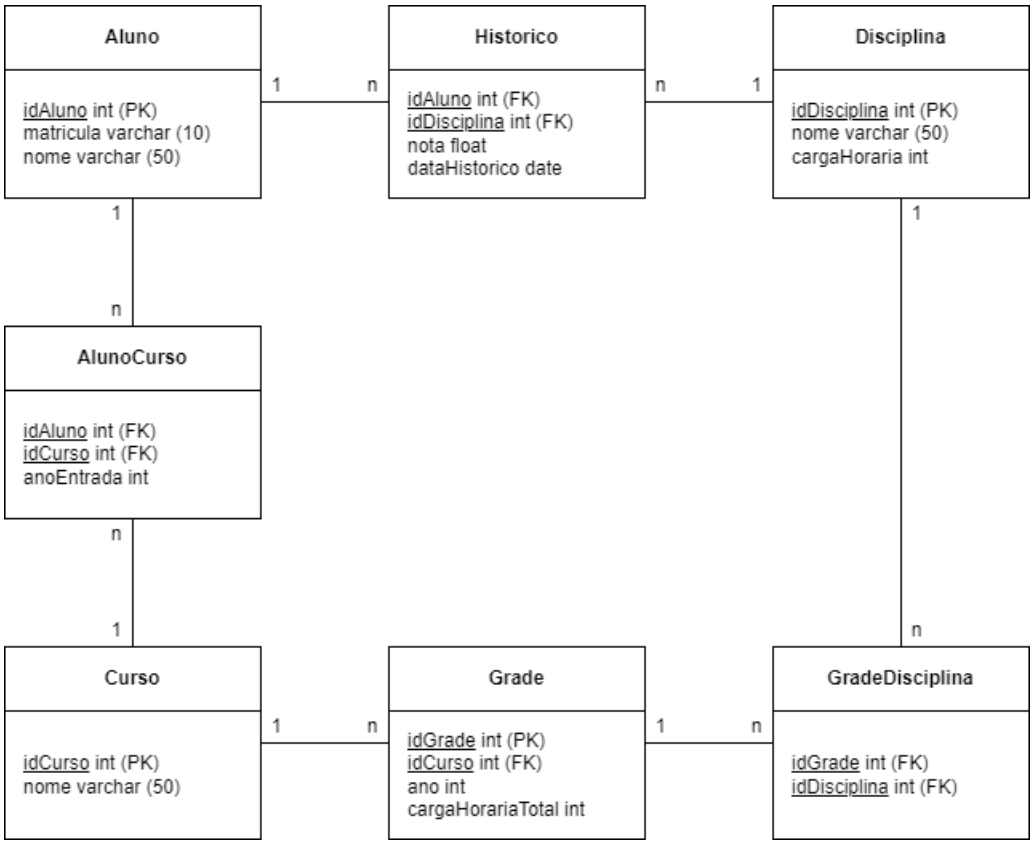
- Voo – Deverão ser armazenados os seguintes dados: identificação do voo, número do avião, cidade de origem, cidade destino, data do voo e hora do voo;
- Assentos – Deverão ser armazenados os seguintes dados: identificação do assento e quantidade;
- Passageiro – Deverão ser armazenados os seguintes dados: CPF, nome, telefone, e-mail e endereço (rua, número, complemento, bairro, CEP, cidade e estado);

- Dependentes – Deverão ser armazenados os seguintes dados: nome e data de nascimento;
- Um voo pode ter zero ou vários assentos, assim como zero ou vários assentos pertencem a um voo;
- Um passageiro pode ter zero ou várias reservas de assentos, assim como zero ou várias reservas de assentos pertencem a um passageiro;
- Um passageiro pode ter zero ou vários dependentes, assim como zero ou vários dependentes são de um passageiro;
- Da reserva, deverão ser armazenados os seguintes dados: data da reserva e hora da reserva.



2. 2ª Etapa – Implementação

Considere o seguinte Modelo Relacional (lógico):



Com base no Modelo Relacional dado e utilizando a *Structured Query Language* (SQL), no MySQL Workbench, implemente o que se pede.

Observação: Para testar o Banco de Dados após a criação, utilize os comandos contidos no arquivo “Trabalho – Populando o Banco de Dados”, o qual contém todos os comandos de inserção de dados (fictícios) necessários para a realização dos testes.

Pontuação: 25 pontos.

1. Implemente um Banco de Dados chamado “Faculdade”. Após, crie as tabelas, conforme o Modelo Relacional dado, observando as chaves primárias e as chaves estrangeiras. Todos os campos, de todas as tabelas, não podem ser nulos.

```
CREATE DATABASE faculdade;


-- Tabela Aluno
CREATE TABLE Aluno (
  id_aluno INT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  data_nascimento DATE NOT NULL,
  endereco VARCHAR(200) NOT NULL
);

-- Tabela Histórico
CREATE TABLE Histórico (
  id_aluno INT,
  id_disciplina INT,
  nota DECIMAL(4,2) NOT NULL,
  PRIMARY KEY (id_aluno, id_disciplina),
  FOREIGN KEY (id_aluno) REFERENCES Aluno(id_aluno),
  FOREIGN KEY (id_disciplina) REFERENCES Disciplina(id_disciplina)
);

-- Tabela Disciplina
CREATE TABLE Disciplina (
  id_disciplina INT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  carga_horaria INT NOT NULL,
  carahorariatotal INT NOT NULL
);

-- Tabela AlunoCurso
CREATE TABLE AlunoCurso (
  id_aluno INT,
  id_curso INT,
  ano_entrada INT NOT NULL,
  PRIMARY KEY (id_aluno, id_curso),
  FOREIGN KEY (id_aluno) REFERENCES Aluno(id_aluno),
  FOREIGN KEY (id_curso) REFERENCES Curso(id_curso)
);
```

```
-- Tabela Curso
CREATE TABLE Curso (
  id_curso INT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  duracao INT NOT NULL
);

-- Tabela Grade
CREATE TABLE Grade (
  id_grade INT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  id_curso INT,
  ano INT NOT NULL,
  cargahorariatotal INT NOT NULL,
  FOREIGN KEY (id_curso) REFERENCES Curso(id_curso)
);

-- Tabela GradeDisciplina
CREATE TABLE GradeDisciplina (
  id_grade INT,
  id_disciplina INT,
  semestre INT NOT NULL,
  PRIMARY KEY (id_grade, id_disciplina),
  FOREIGN KEY (id_grade) REFERENCES Grade(id_grade),
  FOREIGN KEY (id_disciplina) REFERENCES Disciplina(id_disciplina)
);
```

Pontuação: 10 pontos.

2. Implemente uma consulta para listar o quantitativo de cursos existentes.

```
SELECT COUNT(*) AS quantitativo_cursos
FROM Curso;
```

Pontuação: 10 pontos.

3. Implemente uma consulta para listar o nome das disciplinas existentes.

```
SELECT nome
FROM Disciplina;
```

Pontuação: 10 pontos.

4. Implemente uma consulta para listar o nome de todos os cursos e seus respectivos alunos. A listagem deve ser mostrada em ordem decrescente pelo nome dos cursos.

```
SELECT c.nome AS nome_curso, a.nome AS nome_aluno
FROM Curso c
JOIN AlunoCurso ac ON c.id_curso = ac.id_curso
JOIN Aluno a ON ac.id_aluno = a.id_aluno
ORDER BY c.nome DESC;
```

Pontuação: 10 pontos.

5. Implemente uma consulta para listar a média das notas das disciplinas de todos os cursos. Para isso, utilize o comando *group by*.

```
SELECT c.nome AS nome_curso, AVG(h.nota) AS media_notas
FROM Curso c
JOIN GradeDisciplina gd ON c.id_curso = gd.id_curso
JOIN Disciplina d ON gd.id_disciplina = d.id_disciplina
LEFT JOIN Histórico h ON gd.id_disciplina = h.id_disciplina
GROUP BY c.nome;
```

Pontuação: 10 pontos.

6. Implemente uma consulta para listar o nome de todos os cursos e a quantidade de alunos em cada curso. Para isso, utilize os comandos *join* e *group by*.

```
SELECT c.nome AS nome_curso, COUNT(ac.id_aluno) AS quantidade_alunos
FROM Curso c
LEFT JOIN AlunoCurso ac ON c.id_curso = ac.id_curso
GROUP BY c.nome;
```