

BANCO DE DADOS

Trabalho – Relatório

Curso:	Bacharelado em Engenharia de Software
Aluno(a):	Luiz Fernando de Sousa Carvalho
RU:	4394500

1. 1ª Etapa – Modelagem

Pontuação: 25 pontos.

Dadas as regras de negócio abaixo listadas, referentes ao estudo de caso de uma companhia aérea, elabore o Modelo Entidade-Relacionamento (MER), isto é, o modelo conceitual.

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

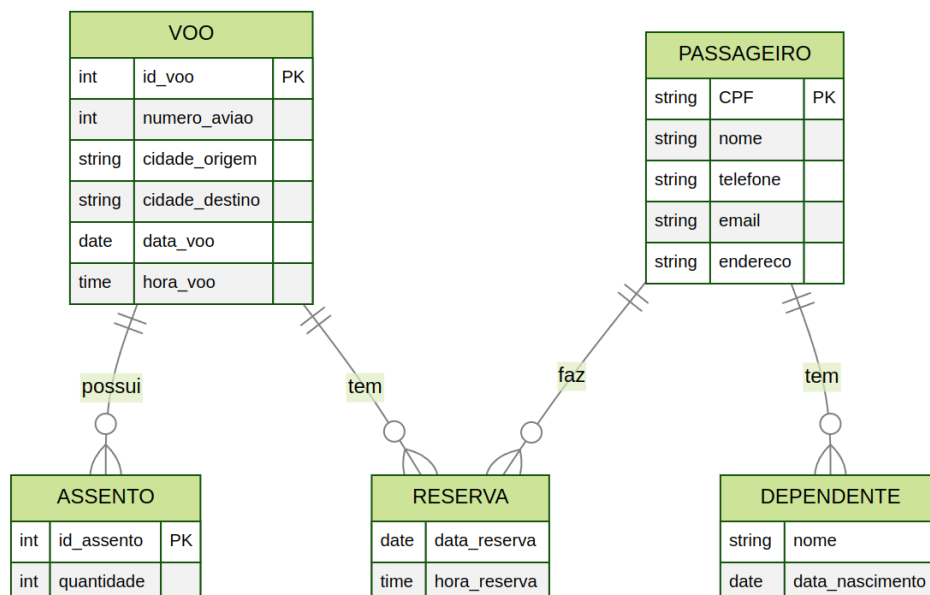
- Entidades;
- Atributos;
- Relacionamentos;
- Cardinalidades;
- Chaves primárias;
- Chaves estrangeiras.

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:

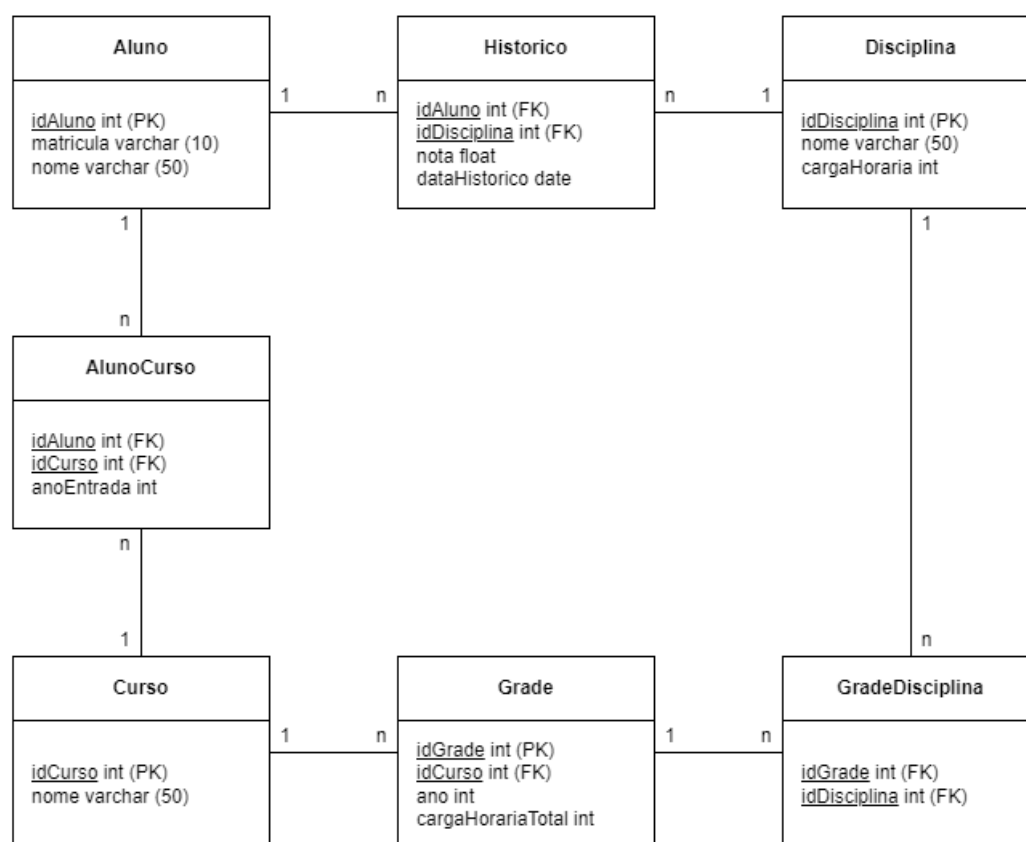
- Voo – Deverão ser armazenados os seguintes dados: identificação do voo, número do avião, cidade de origem, cidade de destino, data do voo e hora do voo;

- Assento – 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);
- Dependente – 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 (modelo lógico), referente ao estudo de caso de uma faculdade:



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 implementação, utilize os comandos contidos no arquivo “Trabalho – Populando o Banco de Dados” para popular as tabelas. Tal arquivo contém todos os comandos de inserção dos dados (fictícios) necessários para a realização dos testes.

Pontuação: 25 pontos.

1. Implemente um Banco de Dados chamado “Faculdade”. Após, implemente 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 (*not null*).

CREATE DATABASE Faculdade;

USE Faculdade;

**CREATE TABLE Aluno (
 idAluno INT PRIMARY KEY,
 matricula VARCHAR(10) NOT NULL,
 nome VARCHAR(255) NOT NULL
);**

**CREATE TABLE Disciplina (
 idDisciplina INT PRIMARY KEY,
 nome VARCHAR(255) NOT NULL,
 cargaHoraria INT NOT NULL
);**

**CREATE TABLE Curso (
 idCurso INT PRIMARY KEY,
 nome VARCHAR(255) NOT NULL
);**

**CREATE TABLE Historico (
 idAluno INT,
 idDisciplina INT,**

```
nota DECIMAL(5,2) NOT NULL,  
dataHistorico DATE NOT NULL,  
FOREIGN KEY (idAluno) REFERENCES Aluno(idAluno),  
FOREIGN KEY (idDisciplina) REFERENCES Disciplina(idDisciplina)  
);
```

```
CREATE TABLE AlunoCurso (  
    idAluno INT,  
    idCurso INT,  
    anoEntrada INT NOT NULL,  
    FOREIGN KEY (idAluno) REFERENCES Aluno(idAluno),  
    FOREIGN KEY (idCurso) REFERENCES Curso(idCurso)  
);
```

```
CREATE TABLE Grade (  
    idGrade INT PRIMARY KEY,  
    idCurso INT,  
    ano INT NOT NULL,  
    cargaHorariaTotal INT NOT NULL,  
    FOREIGN KEY (idCurso) REFERENCES Curso(idCurso)  
);
```

```
CREATE TABLE GradeDisciplina (  
    idGrade INT,  
    idDisciplina INT,  
    FOREIGN KEY (idGrade) REFERENCES Grade(idGrade),  
    FOREIGN KEY (idDisciplina) REFERENCES Disciplina(idDisciplina)  
);
```

Pontuação: 10 pontos.

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

SELECT COUNT(*) AS QuantidadeDeCursos FROM Curso;

The screenshot shows a database query interface. At the top, the SQL query is entered: `SELECT COUNT(*) AS QuantidadeDeCursos FROM Curso;`. Below the query, the 'Result Grid' displays a single row with the value 9 under the column header 'QuantidadeDeCurso'. Below the result grid, the 'Action Output' section shows a log of the query execution, indicating that 1 row(s) were returned.

#	QuantidadeDeCurso
1	9

#	Time	Action	Message
1	12:38:24	SELECT COUNT(*) AS QuantidadeDeCursos FROM Curso ...	1 row(s) returned

Pontuação: 10 pontos.

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

SELECT nome AS NomeDasDisciplinas FROM Disciplina;

The screenshot shows a database query interface. At the top, the SQL query is entered: `SELECT nome AS NomeDasDisciplinas FROM Disciplina;`. Below the query, the 'Result Grid' displays a list of 15 discipline names. Below the result grid, the 'Action Output' section shows a log of the query execution, indicating that 15 row(s) were returned.

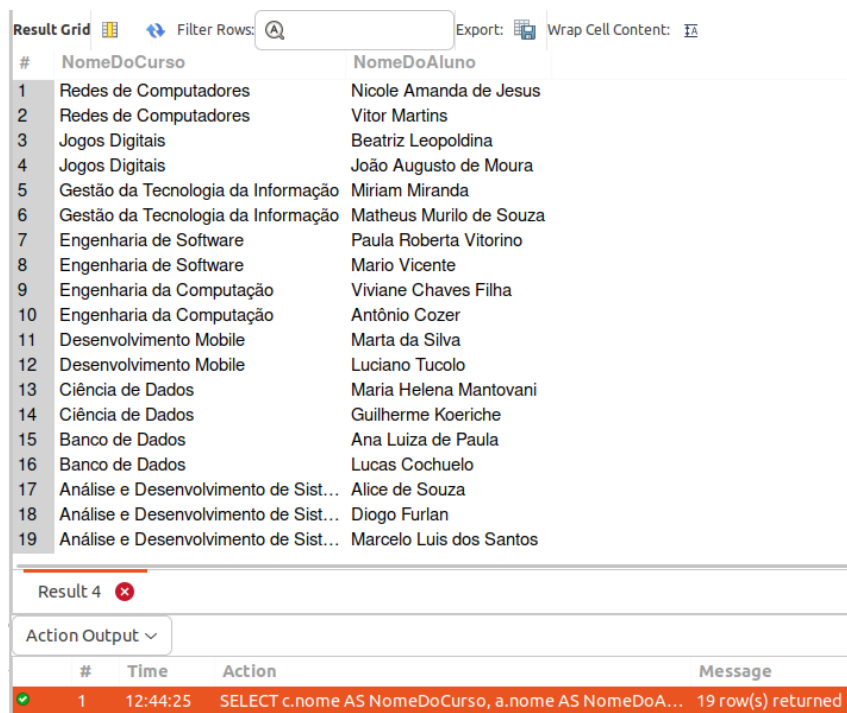
#	NomeDasDisciplinas
1	Análise de Sistemas
2	Arquitetura de Computadores
3	Atividade Extensionista I
4	Atividade Extensionista II
5	Banco de Dados
6	Empreendedorismo
7	Engenharia de Software
8	Fundamentos de Sistemas de Infor...
9	Gestão de Projetos de Software
10	Lógica de Programação e Algoritmos
11	Matemática Computacional
12	Programação de Computadores
13	Programação Orientada a Objetos
14	Sistema Gerenciador de Banco de ...
15	Sistemas Operacionais

#	Time	Action	Message
1	12:42:50	SELECT nome AS NomeDasDisciplinas FROM Disciplina L...	15 row(s) returned

Pontuação: 10 pontos.

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

```
SELECT c.nome AS NomeDoCurso, a.nome AS NomeDoAluno  
FROM Curso c  
JOIN AlunoCurso ac ON c.idCurso = ac.idCurso  
JOIN Aluno a ON ac.idAluno = a.idAluno  
ORDER BY c.nome DESC;
```



The screenshot shows a database query result grid with two columns: 'NomeDoCurso' and 'NomeDoAluno'. The results are ordered by course name in descending order. Below the grid, there is a status bar indicating 'Result 4' and 'Action Output'.

#	NomeDoCurso	NomeDoAluno
1	Redes de Computadores	Nicole Amanda de Jesus
2	Redes de Computadores	Vitor Martins
3	Jogos Digitais	Beatriz Leopoldina
4	Jogos Digitais	João Augusto de Moura
5	Gestão da Tecnologia da Informação	Miriam Miranda
6	Gestão da Tecnologia da Informação	Matheus Murilo de Souza
7	Engenharia de Software	Paula Roberta Vitorino
8	Engenharia de Software	Mario Vicente
9	Engenharia da Computação	Viviane Chaves Filha
10	Engenharia da Computação	Antônio Cozer
11	Desenvolvimento Mobile	Marta da Silva
12	Desenvolvimento Mobile	Luciano Tucolo
13	Ciência de Dados	Maria Helena Mantovani
14	Ciência de Dados	Guilherme Koeriche
15	Banco de Dados	Ana Luiza de Paula
16	Banco de Dados	Lucas Cochuelo
17	Análise e Desenvolvimento de Sist...	Alice de Souza
18	Análise e Desenvolvimento de Sist...	Diogo Furlan
19	Análise e Desenvolvimento de Sist...	Marcelo Luis dos Santos

Result 4 ✖

Action Output ▾

#	Time	Action	Message
✓ 1	12:44:25	SELECT c.nome AS NomeDoCurso, a.nome AS NomeDoA...	19 row(s) returned

Pontuação: 10 pontos.

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

```
SELECT d.nome AS NomeDaDisciplina, AVG(h.nota) AS MediaDasNotas  
FROM Disciplina d  
JOIN Historico h ON d.idDisciplina = h.idDisciplina  
GROUP BY d.nome;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
#	NomeDaDisciplina	MediaDasNotas			
1	Análise de Sistemas	85.000000			
2	Atividade Extensionista I	75.000000			
3	Banco de Dados	85.000000			
4	Gestão de Projetos de Software	75.000000			
5	Matemática Computacional	70.000000			
6	Programação de Computadores	70.000000			
7	Programação Orientada a Objetos	70.000000			
8	Sistema Gerenciador de Banco de ...	82.000000			
9	Arquitetura de Computadores	76.000000			
10	Atividade Extensionista II	80.000000			
11	Empreendedorismo	89.000000			

Result 5

Action Output

#	Time	Action	Message
1	12:45:33	SELECT d.nome AS NomeDaDisciplina, AVG(h.nota) AS M...	11 row(s) returned

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    NomeDoCurso,    COUNT(ac.idAluno)    AS
QuantidadeDeAlunos
FROM Curso c
JOIN AlunoCurso ac ON c.idCurso = ac.idCurso
GROUP BY c.nome;
```


Result Grid

Filter Rows:

Export:

Wrap Cell Content:

#	NomeDoCurso	QuantidadeDeAluno
1	Análise e Desenvolvimento de Sist...	3
2	Banco de Dados	2
3	Ciência de Dados	2
4	Desenvolvimento Mobile	2
5	Engenharia da Computação	2
6	Engenharia de Software	2
7	Gestão da Tecnologia da Informação	2
8	Jogos Digitais	2
9	Redes de Computadores	2

Result 6

Action Output

#	Time	Action	Message
1	12:46:37	SELECT c.nome AS NomeDoCurso, COUNT(ac.idAluno) A...	9 row(s) returned