

BANCO DE DADOS

Trabalho – Relatório

Curso:	ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
Aluno(a):	Vitor de Oliveira Fonseca
RU:	4462897

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:

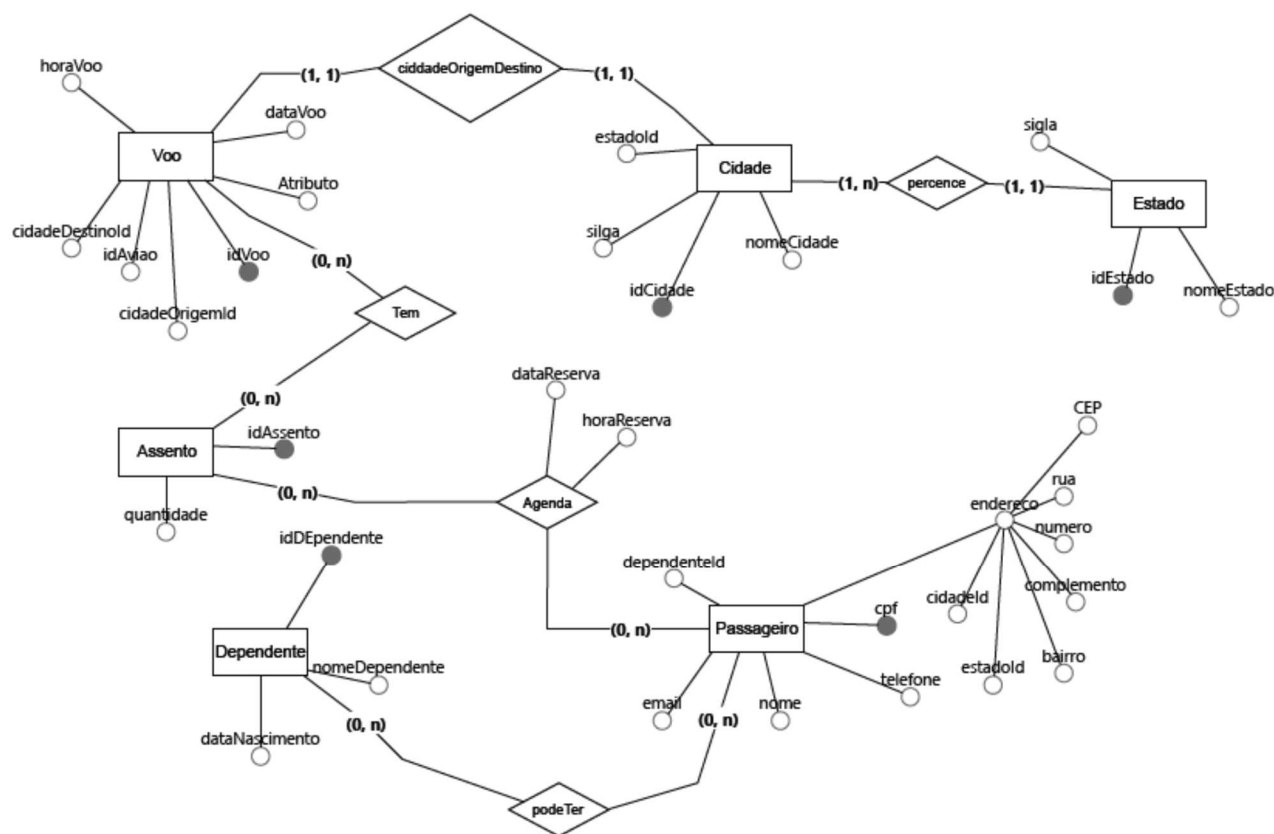
- Entidades;
- Atributos;
- Relacionamentos;
- Cardinalidades;
- Chaves primárias;
- Chaves estrangeiras.

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:

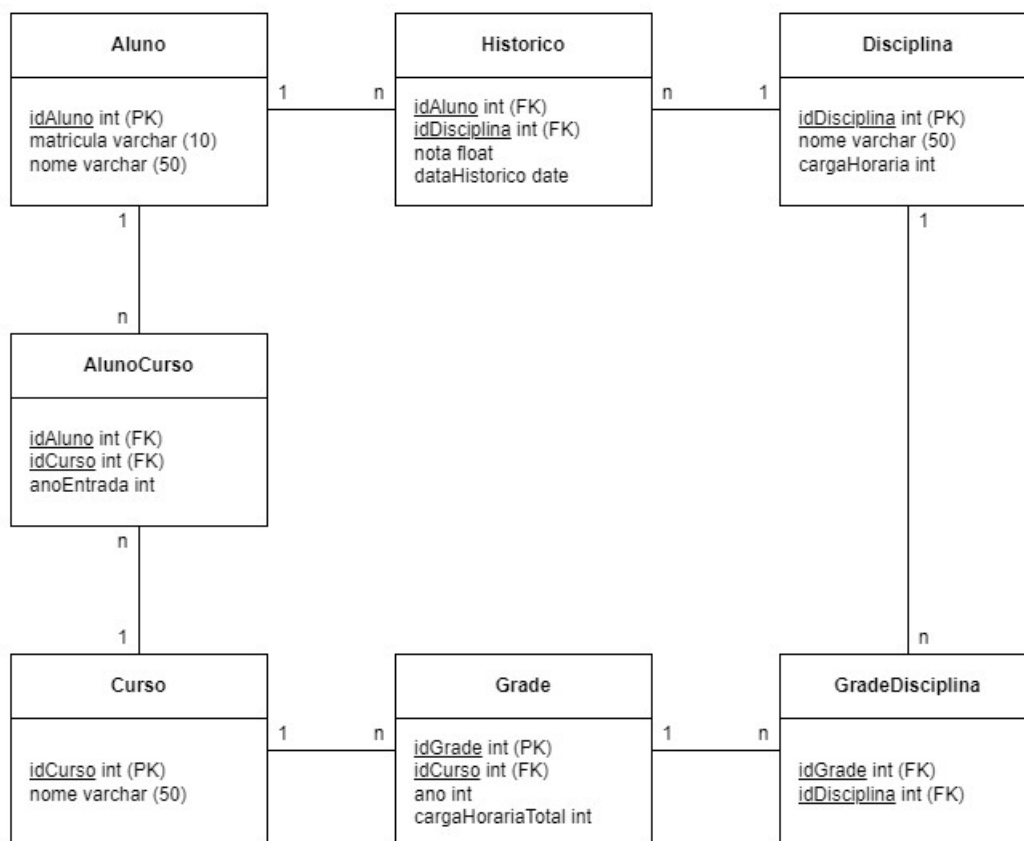
- Voo – Deverão ser armazenados os seguintes dados: identificação do voo, número do avião, cidade de origem, cidade de destino, data do voo e hora do voo;
- Assento – Deverão ser armazenados os seguintes dados: identificação do assento e quantidade;

- 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.



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(50)  
);
```

```
create table Disciplina  
(  
    idDisciplina int primary key,  
    nome varchar(50) not null,  
    cargaHoraria int not null  
  
);
```

```
create table AlunoCurso  
(  
    idAluno int not null,  
    idCurso int not null,  
    anoEntrada int not null  
);
```

```
create table Curso  
(  
    idCurso int primary key,  
    nome varchar(50) not null  
);
```

```
create table Grade
```

```
(  
    idGrade int primary key,  
    idCurso int not null,  
    ano int not null,  
    cargaHorariaTotal int not null  
);
```

```
create table Historico
```

```
(  
    idAluno int not null,  
    idDisciplina int not null,  
    nota float not null,  
    dataHistorico date not null  
);
```

```
create table GradeDisciplina
```

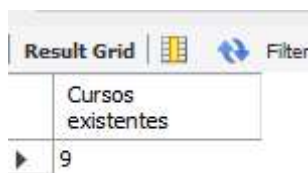
```
(  
    idGrade int not null,  
    idDisciplina int not null  
);
```

```
alter table AlunoCurso add foreign key(idAluno) references Aluno (idAluno);  
alter table AlunoCurso add foreign key(idCurso) references Curso (idCurso);  
alter table Grade add foreign key(idCurso) references Curso (idCurso);  
alter table Historico add foreign key(idAluno) references Aluno (idAluno);  
alter table Historico add foreign key(idDisciplina) references Disciplina (idDisciplina);  
alter table GradeDisciplina add foreign key(idGrade) references Grade (idGrade);  
alter table GradeDisciplina add foreign key(idDisciplina) references Disciplina (idDisciplina);
```

Pontuação: 10 pontos.

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

```
select  
    count(idCurso) as "Cursos existentes"  
from  
    Curso;
```

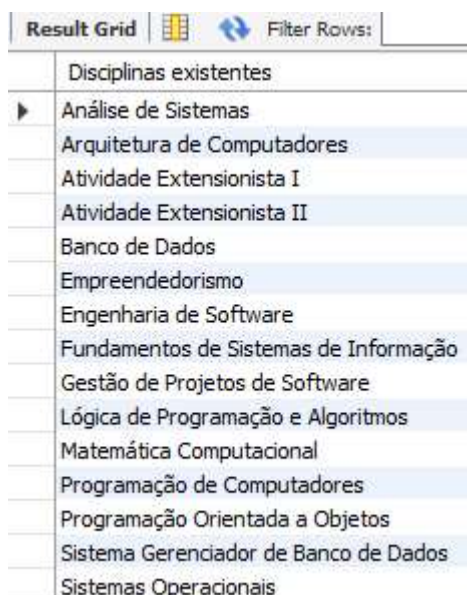


Cursos existentes
9

Pontuação: 10 pontos.

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

```
select  
    nome as "Disciplinas existentes"  
from  
    Disciplina;
```



Disciplinas existentes
Análise de Sistemas
Arquitetura de Computadores
Atividade Extensionista I
Atividade Extensionista II
Banco de Dados
Empreendedorismo
Engenharia de Software
Fundamentos de Sistemas de Informação
Gestão de Projetos de Software
Lógica de Programação e Algoritmos
Matemática Computacional
Programação de Computadores
Programação Orientada a Objetos
Sistema Gerenciador de Banco de Dados
Sistemas Operacionais

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 curso.nome as Cursos, aluno.nome as Alunos from curso  
join alunocurso  
on curso.idCurso = alunocurso.idCurso  
join aluno  
on aluno.idAluno = alunocurso.idAluno  
order by curso.nome;
```

Cursos	Alunos
Análise e Desenvolvimento de Sistemas	Alice de Souza
Análise e Desenvolvimento de Sistemas	Diogo Furlan
Análise e Desenvolvimento de Sistemas	Marcelo Luis dos Santos
Banco de Dados	Ana Luiza de Paula
Banco de Dados	Lucas Cochuelo
Ciência de Dados	Maria Helena Mantovani
Ciência de Dados	Guilherme Koeriche
Desenvolvimento Mobile	Marta da Silva
Desenvolvimento Mobile	Luciano Tucolo
Engenharia da Computação	Viviane Chaves Filha
Engenharia da Computação	Antônio Cozer
Engenharia de Software	Paula Roberta Vitorino
Engenharia de Software	Mario Vicente
Gestão da Tecnologia da Informação	Miriam Miranda
Gestão da Tecnologia da Informação	Matheus Murilo de Souza
Jogos Digitais	Beatriz Leopoldina
Jogos Digitais	João Augusto de Moura
Redes de Computadores	Nicole Amanda de Jesus
Redes de Computadores	Vitor Martins

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`

`disciplina.nome as Disciplinas,`

`avg(historico.nota) as "Média das notas"`

`from`

`disciplina, historico`

`where disciplina.idDisciplina= historico.idDisciplina`

`group by disciplina.idDisciplina;`

Disciplinas	Média das notas
Análise de Sistemas	85
Arquitetura de Computadores	76
Atividade Extensionista I	75
Atividade Extensionista II	80
Banco de Dados	85
Empreendedorismo	89
Gestão de Projetos de Software	75
Matemática Computacional	70
Programação de Computadores	70
Programação Orientada a Objetos	70
Sistema Gerenciador de Banco de...	82

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 curso.nome as nome_do_curso, COUNT(alunocurso.idCurso) as
quantidade_de_alunos
from curso
left join alunocurso on curso.idCurso = alunocurso.idCurso
group by curso.nome;
```

| nome_do_curso                         | quantidade_de_alunos |
|---------------------------------------|----------------------|
| Análise e Desenvolvimento de Sistemas | 3                    |
| Banco de Dados                        | 2                    |
| Ciência de Dados                      | 2                    |
| Desenvolvimento Mobile                | 2                    |
| Engenharia da Computação              | 2                    |
| Engenharia de Software                | 2                    |
| Gestão da Tecnologia da Informação    | 2                    |
| Jogos Digitais                        | 2                    |
| Redes de Computadores                 | 2                    |