

## BANCO DE DADOS

### Trabalho – Relatório

<b>Curso:</b>	Bacharelado Em Engenharia de Software - Distância
<b>Aluno(a):</b>	João Vitor Figueira Prado
<b>RU:</b>	4543960

#### 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:

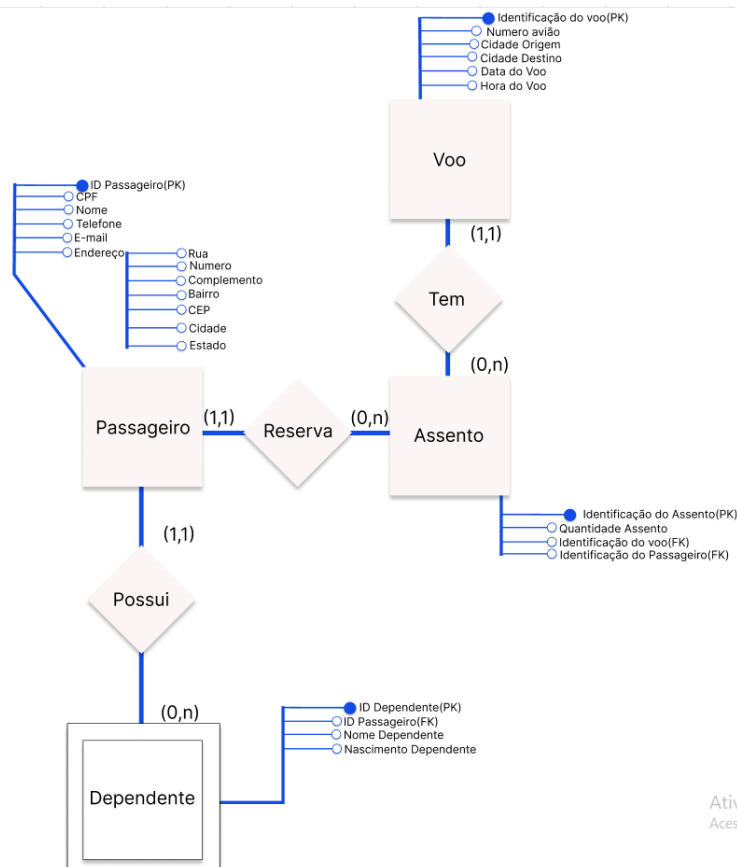
- Entidades;
- Atributos;
- Relacionamentos;
- Cardinalidades;
- Chaves primárias;
- Chaves estrangeiras.

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;

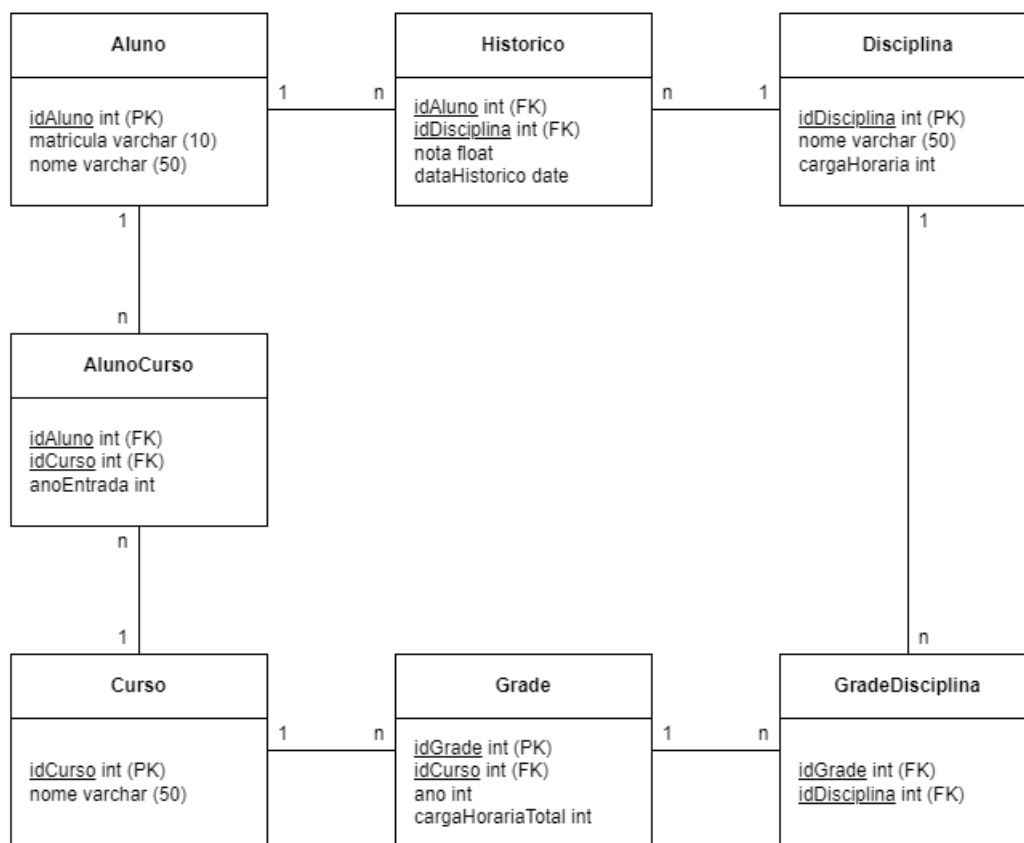
- Passageiro – Deverão ser armazenados os seguintes dados: CPF, nome, telefone, e-mail e endereço (rua, número, complemento, bairro, CEP, cidade e estado);
- Dependente – Deverão ser armazenados os seguintes dados: nome e data de nascimento;
- Um voo pode ter zero ou vários assentos, assim como zero 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.



Ativar o W  
Acesse Config

## **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*).

Código usado:

```
create database Faculdade;  
use faculdade;
```

```
create table Aluno(  
idAluno int primary key not null,  
matrícula varchar(10) not null,  
nomeAluno varchar(50) not null);
```

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

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

```
create table Grade(  
idGrade int primary key not null,  
idCurso int not null,  
constraint fkcursograde foreign key (idCurso) references Curso(idCurso),  
ano int not null,  
cargaHorariaTotal int not null);
```

```
create table GradeDisciplina(  
idGrade int not null,  
constraint fkgradedisciplina foreign key (idGrade) references Grade(idGrade),  
idDisciplina int not null,  
constraint fkdisciplinagrade foreign key (idDisciplina) references Disciplina(idDisciplina));
```

```
create table AlunoCurso(  
idAluno int not null,  
constraint fkalunocurso foreign key (idAluno) references Aluno(idAluno),
```

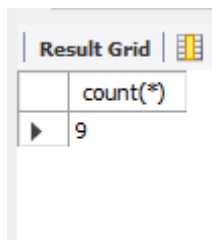
```
idCurso int not null,  
constraint fkcursoaluno foreign key (idCurso) references Curso(idCurso),  
anoEntrada int not null);
```

```
create table Historico(  
idAluno int not null,  
constraint fkalunohistorico foreign key (idAluno) references Aluno(idAluno),  
idDisciplina int not null,  
constraint fkdisciplinahistorico foreign key (idDisciplina) references Disciplina(idDisciplina),  
nota float not null,  
dataHistorico date not null);
```

**Pontuação:** 10 pontos.

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

```
select count(*) from Curso;
```



Result Grid	
	count(*)
▶	9

**Pontuação:** 10 pontos.

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

```
select nome from disciplina;
```

Result Grid		Filter Rows:
	nome	
►	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 c.nome,a.nomealuno from curso as c inner join alunocurso as ac on c.idcurso = ac.idcurso  
inner join aluno as a on ac.idaluno = a.idaluno order by c.nome desc;
```

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

**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,avg(nota) from historico as h inner join disciplina as d on d.iddisciplina=h.iddisciplina  
group by h.iddisciplina;
```



Result Grid	Filter Rows:
nome	avg(nota)
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 c.nome,count(idaluno) from curso as c inner join alunocurso as ac on  
c.idcurso=ac.idcurso group by ac.idcurso;
```

Result Grid	Filter Rows:	Export:
nome	count(idaluno)	
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	