

Matéria Banco de Dados Relacional – 2º Semestre

Data da Aula: 10/02/2022

Construir o Modelo Entidade- Relacionamento e em seguida o lógico para a questão a seguir:

1) Elaborar um diagrama E-R para uma seguradora de automóveis:

Entidades: Cliente, Apólice, Carro e Acidentes.

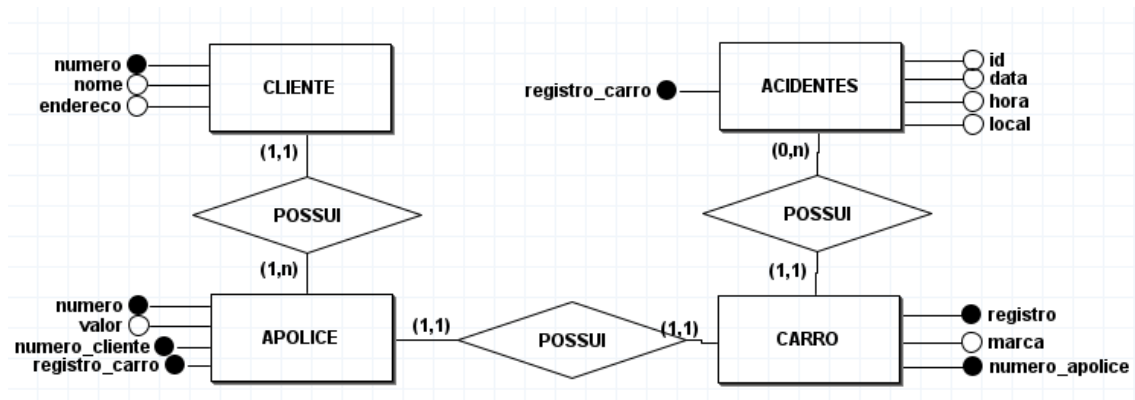
Requisitos:

- a) Um cliente pode ter várias apólices (no mínimo uma);
- b) Cada apólice somente dá cobertura a um carro;
- c) Um carro pode ter zero ou n registros de acidentes a ele;

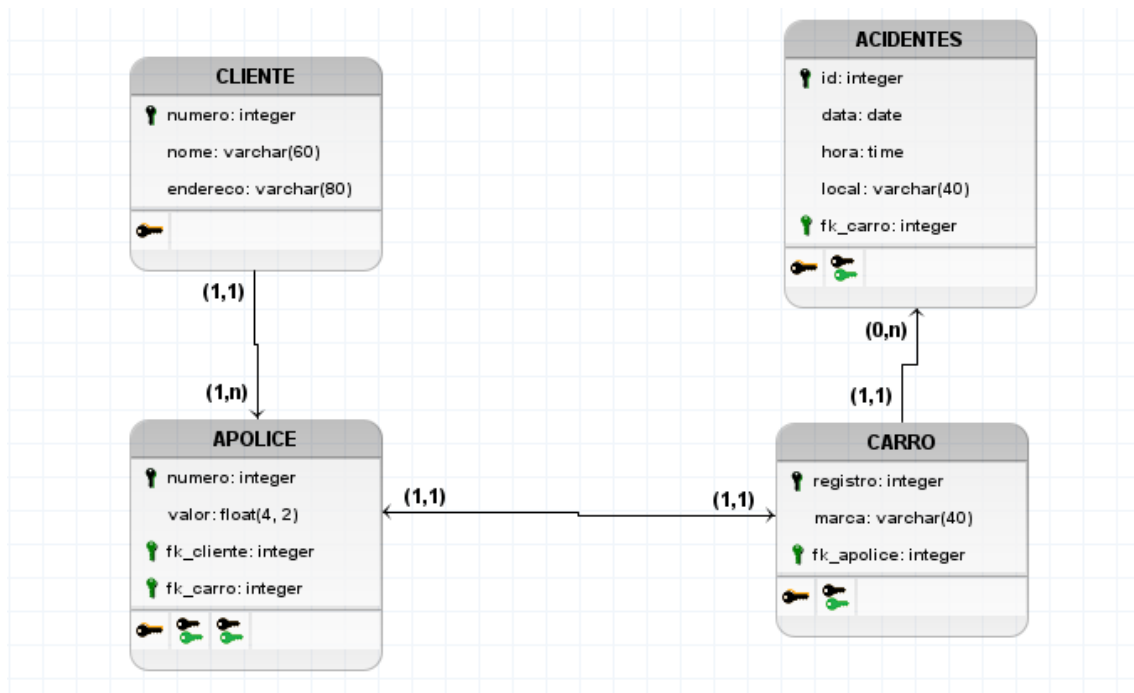
Atributos:

- a) **Cliente:** Número, Nome e Endereço;
- b) **Apólice:** Número e Valor;
- c) **Carro:** Registro e Marca;
- d) **Acidente:** Data, Hora e Local;

Modelo Conceitual:



Modelo Lógico:



Modelo Físico:

-- CRIANDO O BANCO DE DADOS

```
CREATE DATABASE seguradora;
```

-- SELECIONAND QUAL BANCO DE DADOS IREI UTILIZAR

```
USE seguradora;
```

-- CRIANDO A TABELA 'cliente'

```
CREATE TABLE cliente (  
    numero INTEGER AUTO_INCREMENT,  
    nome VARCHAR(60) NOT NULL,  
    endereco VARCHAR(80) NOT NULL,  
    PRIMARY KEY(numero)  
);  
  
SELECT * FROM cliente;  
  
DESC cliente;
```

-- CRIANDO A TABELA 'apolice'

```
CREATE TABLE apolice (  
    numero INTEGER AUTO_INCREMENT,  
    valor FLOAT(4, 2) NOT NULL,  
    PRIMARY KEY(numero),  
    fk_cliente INTEGER,  
    fk_carro INTEGER  
);
```

-- CRIANDO A FK (FOREIGN KEY) COM Nome 'fk_cliente'

```
ALTER TABLE apolice ADD CONSTRAINT fk_apolice_cliente  
FOREIGN KEY(fk_cliente) REFERENCES cliente(numero);
```

```
SELECT * FROM apolice;  
DESC apolice;
```

-- CRIANDO A TABELA 'carro'

```
CREATE TABLE carro (  
    registro INTEGER AUTO_INCREMENT,  
    marca VARCHAR(40) NOT NULL,  
    PRIMARY KEY(registro),  
    fk_apolice INTEGER  
);
```

-- CRIANDO A FK (FOREIGN KEY) COM Nome 'fk_apolice'

```
ALTER TABLE carro ADD CONSTRAINT fk_carro_apolice  
FOREIGN KEY(fk_apolice) REFERENCES apolice(numero);
```

-- CRIANDO A FK (FOREIGN KEY) COM Nome 'fk_carro'

```
ALTER TABLE apolice ADD CONSTRAINT fk_apolice_carro  
FOREIGN KEY(fk_carro) REFERENCES carro(registro);
```

```
SELECT * FROM carro;  
DESC carro;
```

-- CRIANDO A TABELA 'acidentes'

```
CREATE TABLE acidentes (  
    id INTEGER AUTO_INCREMENT,  
    `data` DATE NOT NULL,  
    hora TIME NOT NULL,  
    `local` VARCHAR(40) NOT NULL,  
    PRIMARY KEY(id),  
    fk_carro INTEGER  
);
```

-- CRIANDO A FK (FOREIGN KEY) COM Nome 'fk_carro'

```
ALTER TABLE acidentes ADD CONSTRAINT fk_acidentes_carro  
FOREIGN KEY(fk_carro) REFERENCES carro(registro);
```

```
SELECT * FROM acidentes;  
DESC acidentes;
```

Baseado no Modelo abaixo, resolva os exercícios de SQL.

Criação da Tabela:

-- CRIANDO O BANCO DE DADOS

```
CREATE DATABASE empresa;
```

-- SELECIONAND QUAL BANCO DE DADOS IREI UTILIZAR

```
USE empresa;
```

-- CRIANDO A TABELA 'cargos'

```
CREATE TABLE cargos (  
    id_cargos INTEGER AUTO_INCREMENT,  
    descricao VARCHAR(60) NOT NULL,  
    salario DECIMAL(10, 3) NOT NULL,  
    PRIMARY KEY(id_cargos)  
);  
  
SELECT * FROM cargos;  
  
DESC cargos;
```

-- INSERINDO REGISTROS TABELA 'cargos'

```
INSERT INTO cargos(descricao, salario)  
VALUES('PROGRAMADOR', 1.800);  
  
INSERT INTO cargos(descricao, salario)  
VALUES('DBA', 5.800);  
  
INSERT INTO cargos(descricao, salario)  
VALUES('GERENTE', 4.800);
```

-- CRIANDO A TABELA 'funcionario'

```
CREATE TABLE funcionario (  
    id INTEGER AUTO_INCREMENT,  
    nome VARCHAR(60) NOT NULL,  
    primary key(id),  
    fk_cargos INTEGER  
);
```

-- CRIANDO A FK (FOREIGN KEY) COM Nome 'fk_cargos'

```
ALTER TABLE funcionario ADD CONSTRAINT fk_funcionario_cargos  
FOREIGN KEY(fk_cargos) REFERENCES cargos(id_cargos);
```

-- INSERINDO REGISTROS TABELA 'funcionario'

```
INSERT INTO funcionario(nome, fk_cargos)  
VALUES('João', 1);  
INSERT INTO funcionario(nome, fk_cargos)  
VALUES('Maria', 2);  
INSERT INTO funcionario(nome, fk_cargos)  
VALUES('Fatima', 3);  
INSERT INTO funcionario(nome, fk_cargos)  
VALUES('Jose', 3);  
INSERT INTO funcionario(nome, fk_cargos)  
VALUES('Joana', 2);
```

```
SELECT * FROM funcionario;  
DESC funcionario;
```

1- Mostre todos os Empregados (funcionarios):

R: SELECT * FROM funcionario;

2- Mostre todos os Empregados (funcionarios) que pertençam ao cargo de GERENTE:

R: SELECT f.id, f.nome, c.descricao
FROM funcionario f, cargos c
WHERE f.fk_cargos = c.id_cargos AND c.descricao = 'GERENTE';

3- Mostre todos os cargos:

R: SELECT * FROM cargos;

4- Mostre o salário referente ao Empregado de código 5:

R: SELECT f.id, f.nome, c.salario
FROM funcionario f, cargos c
WHERE f.id = 5 AND f.fk_cargos = c.id_cargos;

5- Mostre a soma dos salários de todos os Empregados:

R: SELECT SUM(c.salario) AS soma_salarios
FROM funcionario f, cargos c
WHERE f.fk_cargos = c.id_cargos;

6- Mostre a média salarial dos Empregados:

R: SELECT AVG(c.salario) AS media_salarios
FROM funcionario f, cargos c
WHERE f.fk_cargos = c.id_cargos;

7- Mostre o Maior e o Menor salário dos Empregados:

R: SELECT MAX(c.salario) AS maior_salario

```
FROM funcionario f, cargos c  
WHERE f.fk_cargos = c.id_cargos;
```

```
SELECT MIN(c.salario) AS menor_salario  
FROM funcionario f, cargos c  
WHERE f.fk_cargos = c.id_cargos;
```

8- Mostre a quantidade de Empregados cadastrados:

```
R: SELECT COUNT(c.salario) AS qtd_funcionarios  
FROM funcionario f, cargos c  
WHERE f.fk_cargos = c.id_cargos;
```

9- Mostre a soma dos salários, agrupados por cargo:

```
R: SELECT c.descricao,  
        SUM(c.salario) AS soma_salarios,  
        COUNT(c.salario) AS qtd_salarios  
FROM funcionario f, cargos c  
WHERE f.fk_cargos = c.id_cargos  
GROUP BY c.descricao;
```