**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:**

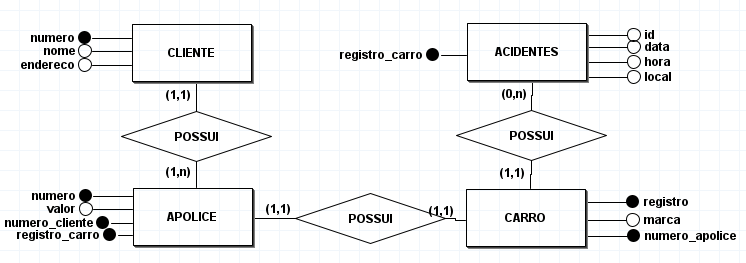
**a) Cliente:** Número, Nome e Endereço;

**b)** **Apólice:** Número e Valor;

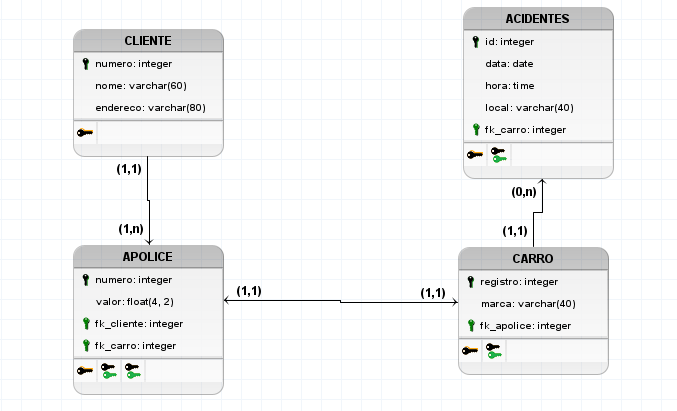
**c)** **Carro:** Registro e Marca;

**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\_cliente\_apolice

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\_apolice\_carro

FOREIGN KEY(fk\_apolice) REFERENCES apolice(numero);

*-- CRIANDO A FK (FOREIGN KEY) COM Nome 'fk\_carro'*

ALTER TABLE apolice ADD CONSTRAINT fk\_carro\_apolice

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\_carro\_acidentes

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\_cargos\_funcionario

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;