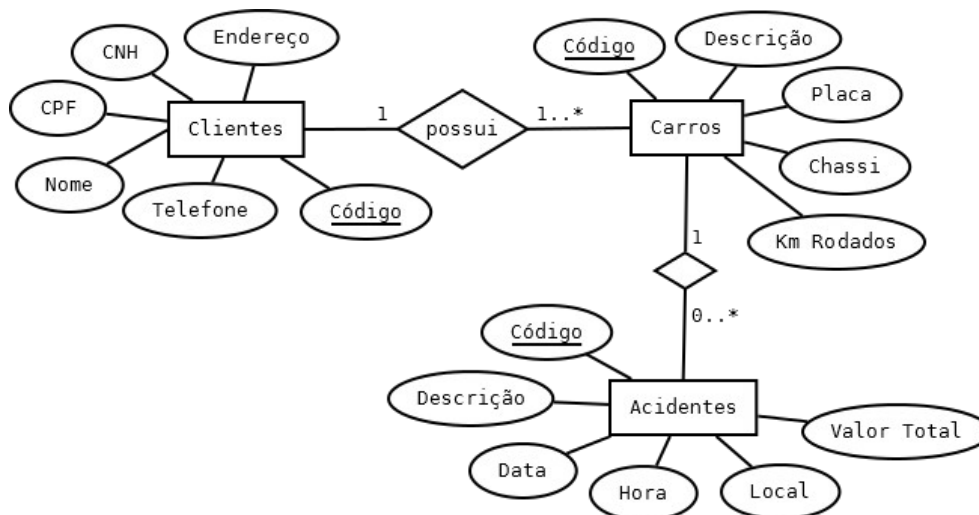


BANCO DE DADOS – LISTA DE EXERCÍCIOS 8

SOLUÇÕES

Exercício 1.

Modelo ER:



Modelo Relacional:

clientes (codigo, nome, cpf, cnh, endereco, telefone)
carros (codigo, descricao, placa, chassi, km_rodados, cod_cliente)
 cod_cliente referencia clientes
acidentes (codigo, descricao, data, hora, local, valor_total, cod_carro)
 cod_carro referencia carros

Criação do banco de dados em SQL:

```
CREATE DATABASE exercicio1;
```

```
\c exercicio1
```

```
CREATE TABLE clientes (codigo serial primary key,  
    nome varchar(60) not null,  
    cpf bigint not null unique,  
    cnh bigint not null unique,  
    endereco varchar(100),  
    telefone int);
```

```
CREATE TABLE carros (codigo serial primary key,  
    descricao varchar(100) not null,  
    placa char(7) not null unique,  
    chassi varchar(100) not null unique,  
    km_rodados int,  
    cod_cliente int not null references clientes(codigo));
```

```
CREATE TABLE acidentes (codigo serial primary key,  
    descricao varchar(100) not null,  
    data date not null,  
    hora time not null,  
    local varchar(100) not null,  
    valor_total numeric(20,2) not null,
```

```
cod_carro int not null references carros(codigo));

INSERT INTO clientes VALUES (default,
                              'Jose da Silva',
                              25869585350,
                              11111111111,
                              'Rua ABC, 123',
                              32323232);
INSERT INTO clientes VALUES (default,
                              'Fulano de Tal',
                              17257643180,
                              62202610545,
                              'Rua DEF, 456',
                              32322323);
INSERT INTO clientes VALUES (default,
                              'Ciclana das Neves',
                              15713147177,
                              50354476505,
                              'Rua GHI, 789',
                              32320000);

INSERT INTO carros VALUES (default,
                             'Fiat Palio 2008',
                             'IOM2867',
                             '123345ABC',
                             60000,
                             1);
INSERT INTO carros VALUES (default,
                             'Ford Fiesta Sedan 2010',
                             'IMG1234',
                             '123345DEF',
                             30000,
                             1);
INSERT INTO carros VALUES (default,
                             'Honda Civic',
                             'IKW1243',
                             '34566DEF',
                             45678,
                             2);
INSERT INTO carros VALUES (default,
                             'Uno Mille',
                             'IDQ4321',
                             '456321FG',
                             23000,
                             3);

INSERT INTO acidentes VALUES (default,
                               'Batida em poste',
                               '2008-02-20',
                               '14:45',
                               'Rua General Neto',
                               10000,
                               1);
INSERT INTO acidentes VALUES (default,
                               'Capotagem',
                               '2013-05-13',
                               '16:30',
                               'Estrada BR-116',
                               8000,
```

```
1);  
INSERT INTO acidentes VALUES (default,  
                                'Vidro trincado por pedra',  
                                '2014-01-17',  
                                '20:14',  
                                'Estrada RS-734',  
                                300,  
                                3);  
INSERT INTO acidentes VALUES (default,  
                                'Batida frontal',  
                                '2013-04-17',  
                                '10:00',  
                                'Estrada RS-734',  
                                49000,  
                                2);
```

Exercício 2.

```
SELECT acidentes.*  
FROM acidentes  
INNER JOIN carros ON (acidentes.cod_carro = carros.codigo)  
INNER JOIN clientes ON (carros.cod_cliente = clientes.codigo)  
WHERE clientes.nome = 'Jose da Silva';
```

Exercício 3.

```
SELECT clientes.nome, count(acidentes.codigo) as num_acidentes  
FROM acidentes  
INNER JOIN carros ON (acidentes.cod_carro = carros.codigo)  
INNER JOIN clientes ON (carros.cod_cliente = clientes.codigo)  
WHERE acidentes.data BETWEEN '2014-01-01' AND '2014-02-28'  
GROUP BY clientes.nome;
```

Exercício 4.

```
DELETE FROM acidentes WHERE data < '2009-01-01';
```

Exercício 5.

```
SELECT clientes.nome, sum(acidentes.valor_total) as total_danos  
FROM acidentes  
INNER JOIN carros ON (acidentes.cod_carro = carros.codigo)  
INNER JOIN clientes ON (carros.cod_cliente = clientes.codigo)  
WHERE acidentes.data BETWEEN '2013-01-01' AND '2013-12-31'  
GROUP BY clientes.nome;
```

Exercício 6.

```
SELECT clientes.nome  
FROM acidentes  
INNER JOIN carros ON (acidentes.cod_carro = carros.codigo)  
INNER JOIN clientes ON (carros.cod_cliente = clientes.codigo)  
WHERE acidentes.data BETWEEN '2012-01-01' AND '2013-12-31'  
GROUP BY clientes.nome
```

```
HAVING SUM(acidentes.valor_total) > 50000;
```

Exercício 7.

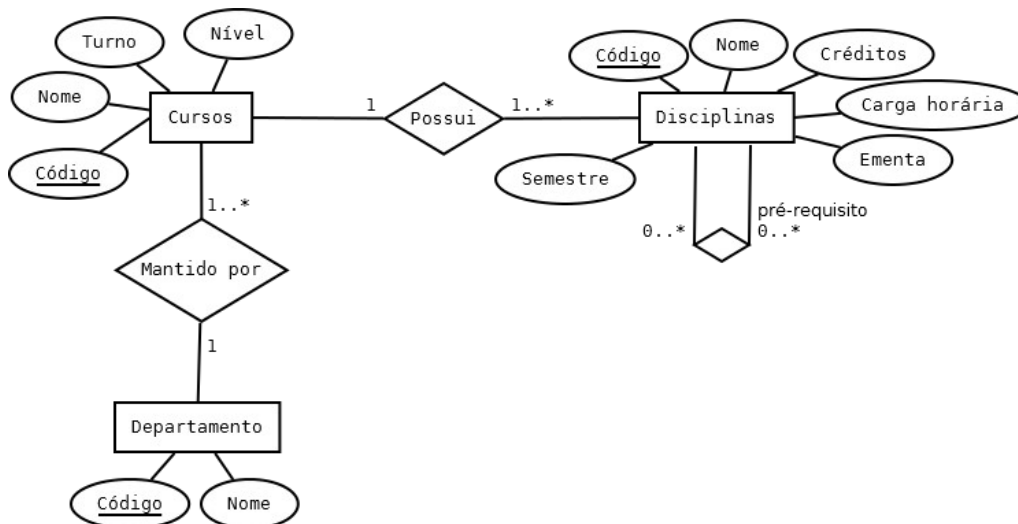
Anulada, pois pelo enunciado não há como fazer com o que vimos até aqui.

Exercício 8.

Anulada, pois pelo enunciado não há como fazer com o que vimos até aqui.

Exercício 9.

Modelo ER:



Modelo Relacional:

departamentos (codigo, nome)
cursos (codigo, nome, turno, nivel, cod_departamento)
 cod_departamento referencia departamentos
disciplinas (codigo, nome, credits, carga_horaria, ementa, semestre, cod_curso)
 cod_curso referencia cursos
pre-requisitos (cod_disciplina, cod_pre_requisito)
 cod_disciplina referencia disciplinas
 cod_pre_requisito referencia disciplinas

Criação do banco de dados em SQL:

```
CREATE DATABASE exercicio9;

\c exercicio9

CREATE TABLE departamentos (codigo serial primary key,
                             nome varchar(60) not null);

CREATE TABLE cursos (codigo serial primary key,
                      nome varchar(60) not null,
                      turno char(1) not null,
                      nivel char(1) not null,
```

```
        cod_departamento int not null references departamentos(codigo),
        check (turno in ('D', 'N')),
        check (nivel in ('G', 'E', 'M', 'D')));

CREATE TABLE disciplinas (codigo serial primary key,
                           nome varchar(60) not null,
                           creditos int not null,
                           carga_horaria int not null,
                           ementa text,
                           semestre int not null,
                           cod_curso int not null references cursos(codigo));

CREATE TABLE pre_requisitos (
    cod_disciplina int not null references disciplinas (codigo),
    cod_pre_requisito int not null references disciplinas (codigo),
    primary key (cod_disciplina, cod_pre_requisito));

INSERT INTO departamentos (nome) VALUES ('Ciencia da Computacao');
INSERT INTO departamentos (nome) VALUES ('Engenharia');
INSERT INTO departamentos (nome) VALUES ('Matematica');
INSERT INTO departamentos (nome) VALUES ('Fisica');
INSERT INTO departamentos (nome) VALUES ('Letras');
INSERT INTO departamentos (nome) VALUES ('Direito');

INSERT INTO cursos VALUES (
    default,
    'Tecnologia em Analise e Desenvolvimento de Sistemas',
    'N',
    'G',
    1);
INSERT INTO cursos VALUES (
    default,
    'Engenharia Civil',
    'D',
    'G',
    2);
INSERT INTO cursos VALUES (
    default,
    'Direito',
    'D',
    'G',
    6);
INSERT INTO cursos VALUES (
    default,
    'Letras Portugues/Ingles',
    'D',
    'G',
    5);
INSERT INTO cursos VALUES (
    default,
    'Mestrado em Ciencia da Computacao',
    'D',
    'M',
    1);

INSERT INTO disciplinas VALUES (
    default,
    'Logica de Programacao',
    9,
```

```
135,  
null,  
1,  
1);  
INSERT INTO disciplinas VALUES (  
default,  
'Fundamentos de Computacao',  
4,  
60,  
null,  
1,  
1);  
INSERT INTO disciplinas VALUES (  
default,  
'Banco de Dados',  
7,  
105,  
null,  
2,  
1);  
INSERT INTO disciplinas VALUES (  
default,  
'Engenharia de Software',  
6,  
90,  
null,  
3,  
1);  
INSERT INTO disciplinas VALUES (  
default,  
'Calculo Diferencial e Integral I',  
6,  
90,  
null,  
1,  
2);  
INSERT INTO disciplinas VALUES (  
default,  
'Resistencia dos Materiais',  
4,  
60,  
null,  
4,  
2);  
INSERT INTO disciplinas VALUES (  
default,  
'Filosofia do Direito',  
2,  
30,  
null,  
1,  
3);  
INSERT INTO disciplinas VALUES (  
default,  
'Historia do Direito',  
4,  
60,  
null,  
1,
```

```
3);  
INSERT INTO disciplinas VALUES (  
    default,  
    'Direito Civil I',  
    6,  
    90,  
    null,  
    2,  
    3);  
INSERT INTO disciplinas VALUES (  
    default,  
    'Direito Civil II',  
    6,  
    90,  
    null,  
    3,  
    3);  
INSERT INTO disciplinas VALUES (  
    default,  
    'Direito Penal I',  
    9,  
    135,  
    null,  
    4,  
    3);  
INSERT INTO disciplinas VALUES (  
    default,  
    'Direito Penal II',  
    6,  
    90,  
    null,  
    5,  
    3);  
INSERT INTO disciplinas VALUES (  
    default,  
    'Direito do Trabalho',  
    6,  
    90,  
    null,  
    6,  
    3);
```

Exercício 10.

```
SELECT d.nome, d.creditos, d.carga_horaria  
FROM disciplinas as d  
INNER JOIN cursos as c ON (d.cod_curso = c.codigo)  
WHERE c.nome = 'Tecnologia em Analise e Desenvolvimento de Sistemas'  
ORDER BY d.semestre, d.nome;
```

Exercício 11.

```
SELECT SUM(d.carga_horaria) as carga_horaria_total  
FROM disciplinas as d  
INNER JOIN cursos as c ON (d.cod_curso = c.codigo)  
WHERE c.nome = 'Engenharia Civil';
```

Exercício 12.

Anulada, pois pelo enunciado não há como fazer com o que vimos até aqui.

Exercício 13.

```
SELECT c.nome
FROM cursos as c
INNER JOIN departamentos as d ON (c.cod_departamento = d.codigo)
WHERE d.nome = 'Ciencia da Computacao';
```

Exercício 14.

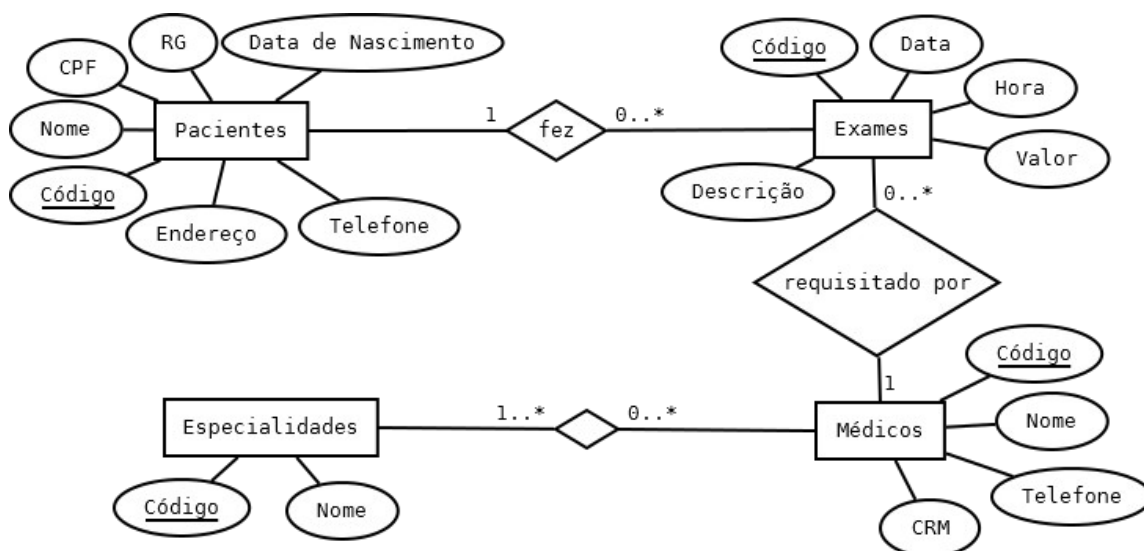
```
SELECT d.nome, COUNT(c.codigo) as num_cursos
FROM cursos as c
INNER JOIN departamentos as d ON (c.cod_departamento = d.codigo)
GROUP BY d.nome;
```

Exercício 15.

```
SELECT d.semestre, SUM(d.creditos) as num_creditos
FROM disciplinas as d
INNER JOIN cursos as c ON (d.cod_curso = c.codigo)
WHERE c.nome = 'Direito'
GROUP BY d.semestre
ORDER BY d.semestre;
```

Exercício 16.

Modelo ER:



Modelo Relacional:

pacientes (codigo, nome, cpf, rg, data_nascimento, endereco, telefone)
especialidades (codigo, nome)
medicos (codigo, nome, telefone, crm, cod_especialidade)
cod_especialidade referencia especialidades
exames (codigo, descricao, data, hora, valor, cod_medico, cod_paciente)

cod_medico referencia medicos
cod_paciente referencia pacientes

Criação do banco de dados em SQL:

```
CREATE DATABASE exercicio16;

\c exercicio16

CREATE TABLE pacientes (codigo serial primary key,
                        nome varchar(60) not null,
                        cpf bigint not null unique,
                        rg bigint not null,
                        data_nascimento date,
                        endereco varchar(100),
                        telefone int);

CREATE TABLE especialidades (codigo serial primary key,
                             nome varchar(40) not null);

CREATE TABLE medicos (codigo serial primary key,
                       nome varchar(60) not null,
                       telefone int not null,
                       crm bigint not null,
                       cod_especialidade int not null references especialidades(codigo));

CREATE TABLE exames (codigo serial primary key,
                      descricao varchar(100) not null,
                      data date not null,
                      hora time not null,
                      valor numeric(20,2) not null,
                      cod_medico int not null references medicos(codigo),
                      cod_paciente int not null references pacientes(codigo));

INSERT INTO especialidades (nome) VALUES ('Pediatria');
INSERT INTO especialidades (nome) VALUES ('Obstetricia');
INSERT INTO especialidades (nome) VALUES ('Clinica Geral');

INSERT INTO pacientes VALUES (
    default,
    'Fulano da Silva',
    17257643180,
    1234567890,
    '1976-03-20',
    'Rua ABC, 123',
    32323232);

INSERT INTO pacientes VALUES (
    default,
    'Jose das Neves',
    62202610545,
    0987654321,
    '1985-10-08',
    'Rua DEF, 456',
    32320000);

INSERT INTO pacientes VALUES (
    default,
    'Maria da Conceicao',
    15713147177,
    5035447650,
```

```
        '1970-12-04',  
        'Rua GHI, 789',  
        32322323);  
  
INSERT INTO medicos VALUES (  
    default,  
    'Ciclana das Neves',  
    32324545,  
    123456789,  
    1);  
INSERT INTO medicos VALUES (  
    default,  
    'Joao Seilayevski',  
    32339000,  
    987654321,  
    2);  
INSERT INTO medicos VALUES (  
    default,  
    'Josicleidson da Silva',  
    32367777,  
    111222333,  
    3);  
  
INSERT INTO exames VALUES (  
    default,  
    'Hemograma',  
    '2013-07-29',  
    '07:00',  
    40.00,  
    3,  
    1);  
INSERT INTO exames VALUES (  
    default,  
    'Radiografia da face',  
    '2013-08-10',  
    '16:00',  
    60.00,  
    3,  
    1);  
INSERT INTO exames VALUES (  
    default,  
    'Tomografia',  
    '2014-02-10',  
    '17:00',  
    1000.00,  
    3,  
    1);  
INSERT INTO exames VALUES (  
    default,  
    'Hemograma',  
    '2012-05-27',  
    '07:00',  
    40.00,  
    3,  
    2);  
INSERT INTO exames VALUES (  
    default,  
    'Radigrafia da Face',  
    '2012-01-27',
```

```
        '14:00',  
        60.00,  
        1,  
        3);  
INSERT INTO exames VALUES (  
    default,  
    'Hemograma',  
    '2012-02-01',  
    '07:00',  
    40.00,  
    1,  
    3);  
INSERT INTO exames VALUES (  
    default,  
    'Ultrassonografia',  
    '2012-03-01',  
    '11:00',  
    100.00,  
    2,  
    2);
```

Exercício 17.

```
SELECT exames.*  
FROM exames  
INNER JOIN pacientes ON (exames.cod_paciente = pacientes.codigo)  
WHERE pacientes.nome = 'Fulano da Silva' AND  
    exames.data BETWEEN '2013-01-01' AND '2013-12-31';
```

Exercício 18.

```
SELECT m.nome  
FROM medicos as m  
INNER JOIN especialidades as e ON (m.cod_especialidade = e.codigo)  
WHERE e.nome = 'Pediatria';
```

Exercício 19.

```
SELECT medicos.nome, COUNT(exames.codigo) as numero_exames_requisitados  
FROM exames  
INNER JOIN medicos ON (exames.cod_medico = medicos.codigo)  
WHERE exames.data BETWEEN '2012-01-01' AND '2012-12-31'  
GROUP BY medicos.nome;
```

Exercício 20.

```
SELECT pacientes.nome, SUM(exames.valor) as valor_pago  
FROM pacientes  
INNER JOIN exames ON (pacientes.codigo = exames.cod_paciente)  
WHERE exames.data BETWEEN '2013-01-01' AND '2013-12-31'  
GROUP BY pacientes.nome;
```

Exercício 21.

```
SELECT DISTINCT pacientes.nome  
FROM pacientes  
INNER JOIN exames ON (pacientes.codigo = exames.cod_paciente)  
INNER JOIN medicos ON (exames.cod_medico = medicos.codigo)  
WHERE medicos.nome = 'Ciclana das Neves';
```

Exercício 22.

```
SELECT especialidades.nome, COUNT(DISTINCT exames.cod_paciente)  
FROM especialidades  
INNER JOIN medicos ON (especialidades.codigo = medicos.cod_especialidade)  
INNER JOIN exames ON (medicos.codigo = exames.cod_medico)  
WHERE exames.data BETWEEN '2013-01-01' AND '2013-12-31'  
GROUP BY especialidades.nome;
```