

**Universidade Federal Rural do Rio de Janeiro - INSTITUTO DE CIÊNCIAS EXATAS****Departamento de Computação**

Prof. Sergio Serra

Prova - 2021.1

PROVA SOMENTE A CANETA AZUL OU PRETA (SEM RASURAS)

Aluno:	Matrícula:	VALOR TOTAL: 10,00 pts NOTA:
--------	------------	---------------------------------

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO**2ª PROVA BD I - 10-12-2021****PROFESSOR SERGIO SERRA**

OBSERVAÇÃO 1 - Esta prova é teórico-prática, pode ser feita em casa, você pode consultar softwares a vontade. NÃO COPIE E COLE de outras fontes (Internet e colegas). As provas serão corrigidas por comparação. Provas assemelhadas serão zeradas. Essa prova vale 10 pontos.

OBSERVAÇÃO 2 - Você tem várias horas para fazer a prova (**até dia 13/11/2021 as 13:59:00**). Ela deve ser enviada em resposta ao e-mail. É responsabilidade EXCLUSIVA do aluno garantir que a prova foi submetida e recebida pelo professor. Certifique-se disso. A cada HORA DE ATRASO será atribuído 1,0 ponto de desconto da nota final da P1. Para isso usarei o relógio do Google. Não deixa para a última hora. Faça a sua prova com calma. Além disso, você deverá entregar a prova ASSINADA! Para isso basta salvar sua assinatura junto ao arquivo da prova enviada por e-mail. NÃO ENVIE VERSÕES DISTINTAS.

OBSERVAÇÃO 3 - O complemento da frequência do dia da prova será atribuído quando o professor este receber o material enviado pelo aluno (caderno de perguntas e respostas da prova). Caso não emita resposta, a frequência não será atribuída. Caracterizará uma falta na avaliação

OBSERVAÇÃO 4 - NÃO COPIE DA INTERNET. Seja criativo, a prova não é pesada nem difícil, descreva em detalhes as soluções encaminhadas para cada questão.

OBSERVAÇÃO 5 - Enviar a prova para o email **COMPUTACAOUFRRJ@GMAIL.COM** com assunto **[P2 2021.2 BDI] seu nome completo + matrícula**. Isso será usado para a correção, não fuja da regra. Exemplo: **[P1 2021.1 BD1] Abelardo Barbosa 2020134567**.

QUESTÕES

- 1) REFAÇA TODOS os modelo Entidades-Relacionamentos e os modelos lógicos da P1 (muitas estavam parcialmente errados!) a partir das descrições abaixo e responda as questões associadas. Use a ferramenta BrModelo para o modelo relacional e as heurísticas apresentadas em sala de aula para fazer o modelo conceitual (não use e evite usar o BR modelo para fazer isso de modo automático, será descontado da nota), entregue ao professor o relatório e os diagramas conceitual e lógico com os tipos de dados de cada atributo.
- 2) NORMALIZE todos os 4 Casos, use a até a 4FN, explique o passo a passo de cada normalização de cada tabela
- 3) IMPLEMENTE Todos os modelos FÍSICOS no SQL FIDDLE (banco MySQL ou POSTGRESQL)
- 4) MOSTRE TODOS OS COMANDOS usados para criar e popular e consultar os bancos
- 5) INSIRA 15 registros distintos em cada uma das tabelas de cada um dos casos abaixo. A inserção dos dados é de sua escolha, faça com atenção para que as consultas funcionem corretamente

- 6) RESPONDA COM SQL de cada caso. Apresente a consulta e captura a tela com os resultados

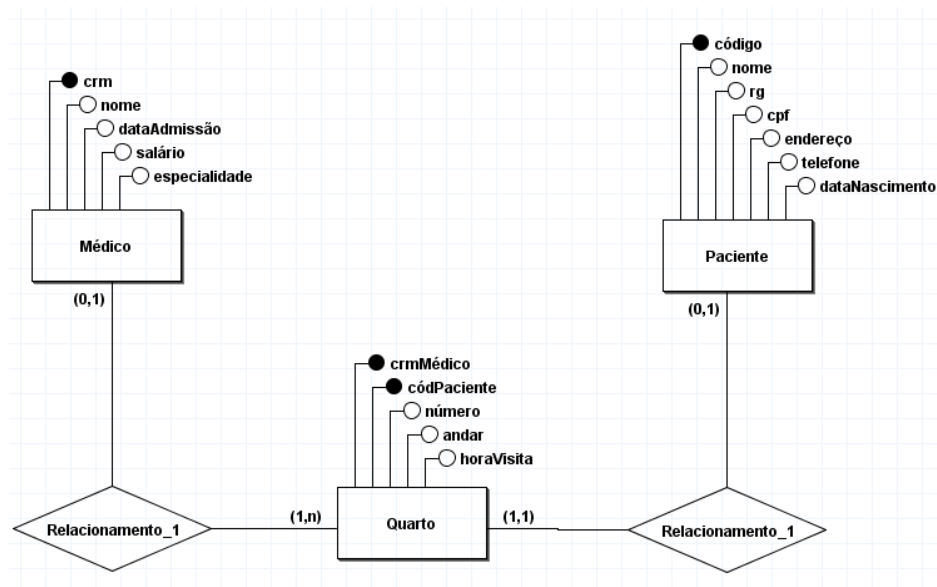
NÃO COPIE OS MODELOS, DADOS E NEM AS CONSULTAS DO COLEGUINHA É Zero!

Cada caso correto (todos os itens) vale até 2.5 pontos.

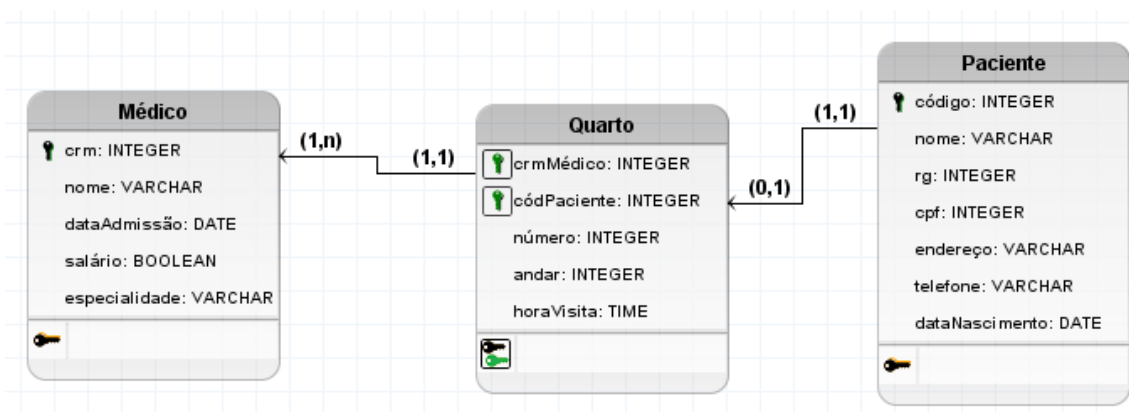
Caso1

Cada médico que trabalha na clínica é identificado pelo seu CRM, possui um nome, uma data de admissão na clínica e um salário. Um médico tem formação em diversas especialidades (ortopedia, traumatologia, etc), mas só exerce uma delas na clínica. Para todo paciente internado na clínica são cadastrados alguns dados pessoais: código (único), nome, RG, CPF, endereço, telefone(s) para contato e data de nascimento. Um paciente tem sempre um determinado médico como responsável, com um horário de visita diário predeterminado. Pacientes estão sempre internados em quartos individuais, que são identificados por um número e estão em um andar da clínica.

Modelo Conceitual:



Modelo Lógico:



Normalização:

No modelo lógico proposto para esse caso, não havia necessidade de realizar a normalização 1FN, 3FN e 4FN. O sistema só permite que seja inserido um telefone e um endereço por paciente, assim não foi necessário realizar a 1FN na parte do paciente, sendo feita apenas a normalização 2FN onde o quarto ganhou uma chave primária e foi criada uma nova entidade para agrupar a conexão de chaves estrangeiras das outras 3 entidades.

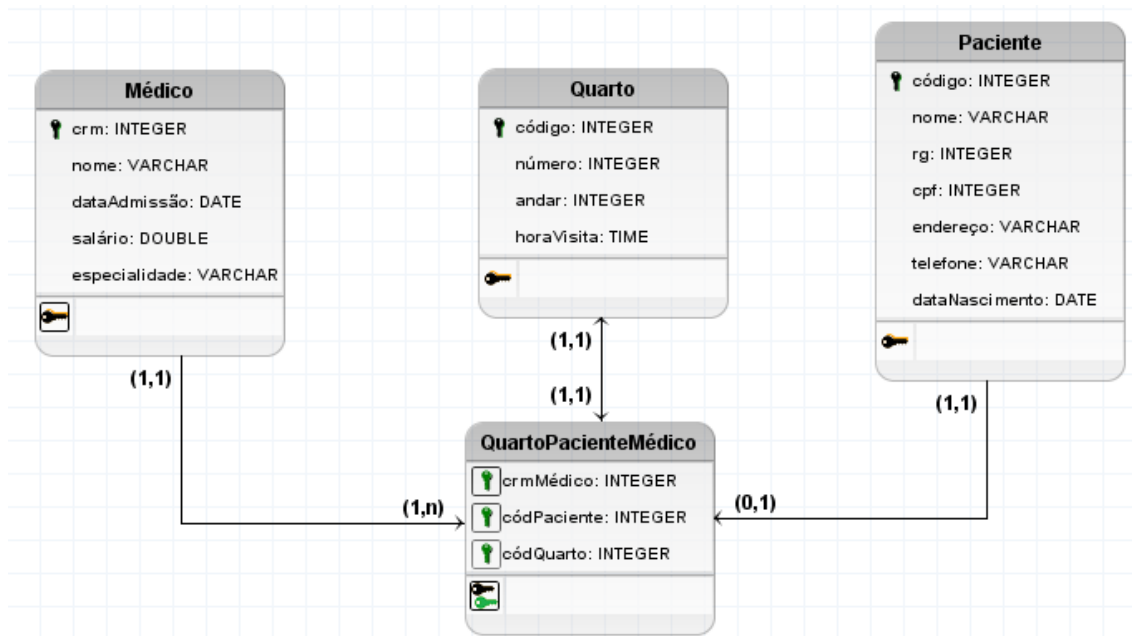


Tabela SQL:

```
CREATE TABLE Médico
(
  crm int,
  nome varchar(255),
  dataAdmissão date,
  salário float,
  especialidade varchar(255),
  PRIMARY KEY (crm)
);
```

```
INSERT INTO Médico
VALUES
(123456, "Roberto", "1995-10-10", 3000.00, "ortopedista"),
(123457, "Paulo", "1995-10-10", 3800.00, "ortopedista"),
(123458, "João", "1995-10-10", 6500.00, "ortopedista"),
(123459, "Paulo", "1995-10-10", 7200.00, "pediatra"),
(123460, "Daniel", "1995-10-10", 2000.00, "ortopedista"),
(123461, "Rafael", "1995-10-10", 8500.00, "pediatra"),
(123462, "Gabriel", "1995-10-10", 2500.00, "ortopedista"),
(123463, "Thiago", "1995-10-10", 3000.00, "ortopedista"),
(123464, "Lucas", "1995-10-10", 8000.00, "pediatra"),
```

```
(123465, "Ayrton", "1995-10-10", 4000.00, "pediatra"),
(123466, "Jeferson", "1995-10-10", 5800.00, "ortopedista"),
(123467, "Adilson", "1995-10-10", 2600.00, "ortopedista"),
(123468, "Gustavo", "1995-10-10", 6300.00, "pediatra"),
(123469, "Felipe", "1995-10-10", 6900.00, "pediatra"),
(123470, "José", "1995-10-10", 7100.00, "ortopedista");
```

```
CREATE TABLE Quarto
(
    código int,
    número int,
    andar int,
    horaVisita time,
    PRIMARY KEY (código)
);
```

```
INSERT INTO Quarto
VALUES
(1, 1, 1, "01:00:00"),
(2, 2, 1, "02:00:00"),
(3, 3, 1, "03:00:00"),
(4, 4, 1, "04:00:00"),
(5, 5, 1, "05:00:00"),
(6, 1, 2, "06:00:00"),
(7, 2, 2, "07:00:00"),
(8, 3, 2, "08:00:00"),
(9, 4, 2, "09:00:00"),
(10, 5, 2, "10:00:00"),
(11, 1, 3, "11:00:00"),
(12, 2, 3, "12:00:00"),
(13, 3, 3, "13:00:00"),
(14, 4, 3, "14:00:00"),
(15, 5, 3, "15:00:00");
```

```
CREATE TABLE Paciente
(
    código int,
    nome varchar(255),
    rg varchar(255),
    cpf varchar(255),
    endereço varchar(255),
    telefone varchar(255),
    dataNascimento date,
    PRIMARY KEY (código)
);
```

INSERT INTO Paciente

VALUES

(1, "João da Silva", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"1995-10-08"),
(2, "João da Silva", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"1995-10-08"),
(3, "Daniel", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342", "2004-10-08"),
(4, "João da Silva", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"1995-10-08"),
(5, "Daniel", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342", "2004-10-08"),
(6, "Daniel", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342", "2004-10-08"),
(7, "Rinaldo", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342", "1993-10-08"),
(8, "Jósue", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342", "2007-10-08"),
(9, "Rinaldo", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342", "1993-10-08"),
(10, "Jósue", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342", "2007-10-08"),
(11, "João da Silva", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"1995-10-08"),
(12, "Rinaldo", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"1993-10-08"),
(13, "Maurício", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"2008-10-08"),
(14, "Rinaldo", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"1993-10-08"),
(15, "Maurício", "27.438.543-9", "156.427.929-23", "Bairro do Caju", "2847-8342",
"2008-10-08");

CREATE TABLE QuartoPacienteMédico

(
 crmMédico int,
 códPaciente int,
 códQuarto int,
 FOREIGN KEY (crmMédico) REFERENCES Médico(crm),
 FOREIGN KEY (códPaciente) REFERENCES Quarto(código),
 FOREIGN KEY (códQuarto) REFERENCES Paciente(código)
);

INSERT INTO QuartoPacienteMédico

VALUES

(123456, 1, 1),
(123457, 2, 2),
(123458, 4, 3),
(123456, 4, 4),
(123460, 2, 5),
(123461, 3, 6),
(123462, 4, 7),
(123463, 5, 8),
(123464, 6, 9),

(123465, 7, 10),
(123466, 2, 11),
(123467, 3, 12),
(123468, 4, 13),
(123469, 5, 14),
(123470, 6, 15);

- a) Faça uma consulta que mostre os médicos ortopedistas que nunca atenderam pacientes com menos de 18 anos.

```
SELECT médico.nome
FROM médico
WHERE médico.especialidade = "ortopedista" AND
médico.crm IN (SELECT quartoPacienteMédico.crmMédico
FROM quartoPacienteMédico
WHERE quartoPacienteMédico.códPaciente IN (SELECT Paciente.código
FROM Paciente
WHERE Paciente.dataNascimento < "2003__%"))
```

- b) Faça uma consulta mostre todas as consultas realizadas pelo paciente João da Silva com médicos que ganham mais de 5000 nos últimos 3 anos.

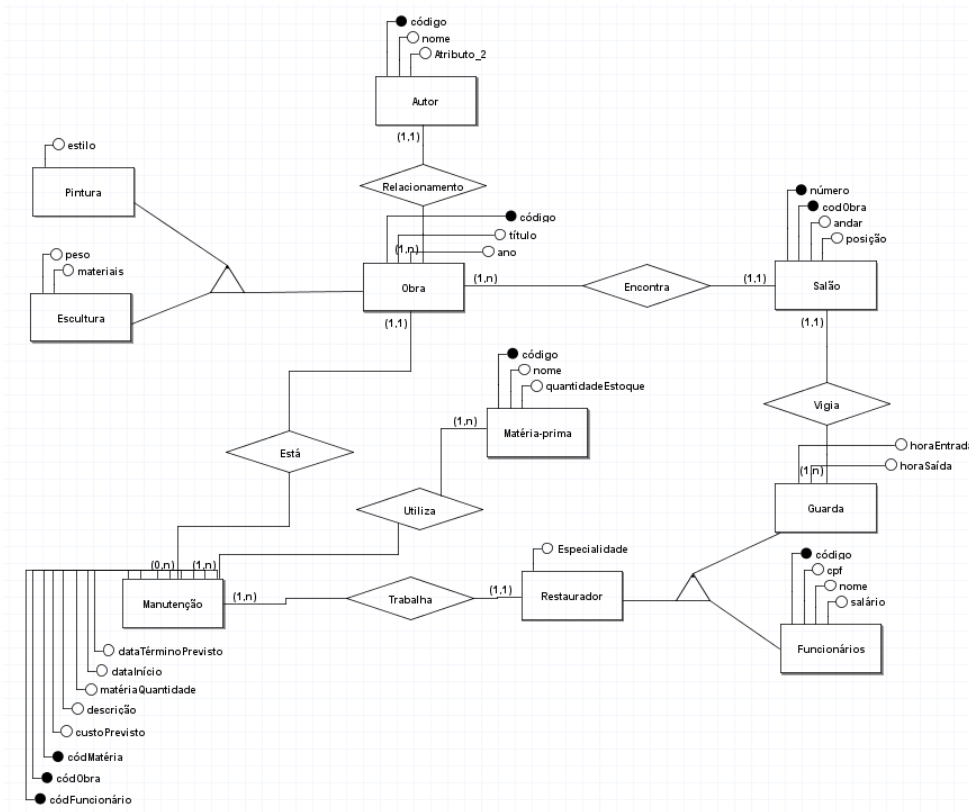
```
SELECT * FROM Paciente
WHERE Paciente.nome = "João da Silva" AND
Paciente.dataNascimento < "2018__%" AND
Paciente.código IN (SELECT QuartoPacienteMédico.códPaciente
FROM QuartoPacienteMédico
WHERE QuartoPacienteMédico.crmMédico IN (SELECT Médico.crm
FROM Médico
WHERE Médico.salário > 5000))
```

Caso2

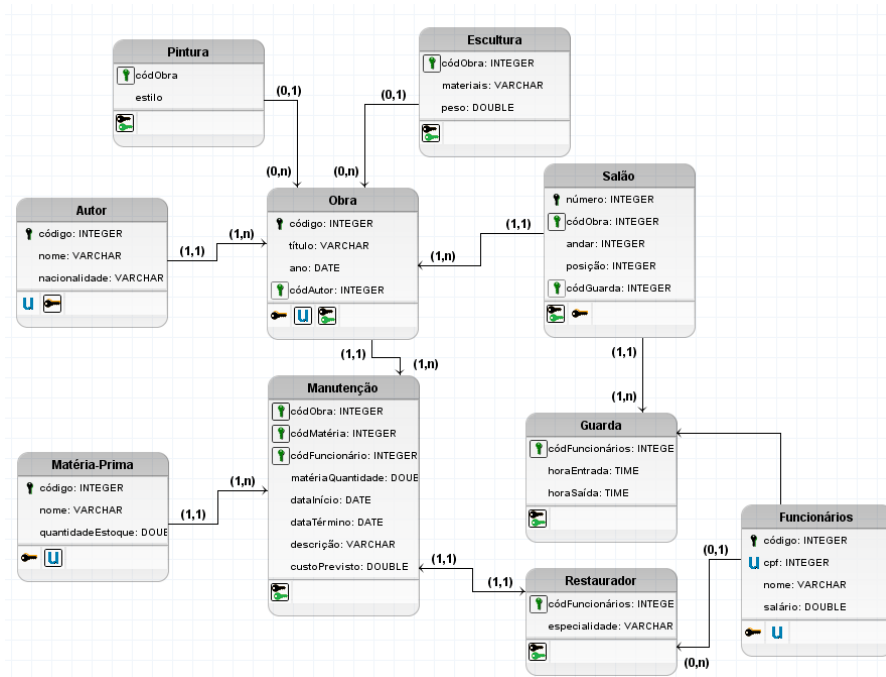
Cada obra no museu possui um código, um título e um ano. Obras ou são pinturas ou são esculturas. No primeiro caso, são dados importantes o estilo (por exemplo, impressionista). No caso de esculturas, são importantes o peso e os materiais de que é feita (por exemplo, argila, madeira, etc). Uma obra pode estar exposta em um único salão, em uma determinada posição neste salão. Um salão, que geralmente abriga várias obras, é identificado por um número e está em um andar do museu. Certos dados a respeito dos autores de cada obra também são relevantes: código, nome e nacionalidade. Uma obra é produzida por apenas um autor, porém, pode existir mais de uma obra de um mesmo autor no museu. No museu trabalham funcionários, cada um possuindo um ID, CPF, um nome e um salário. Funcionários ou são guardas ou são restauradores de obras. No primeiro caso, mantêm-se dados sobre a hora de entrada e hora de saída. No caso de restauradores, qual a sua especialidade. Um guarda é responsável pela segurança de um único salão, que pode ser vigiado por vários guardas. Um restaurador pode estar realizando a manutenção de várias obras. Uma obra, caso esteja em manutenção, está nas mãos de apenas um restaurador. Para cada manutenção deve-se

registrar a data de início e a data prevista de término do trabalho, uma descrição do serviço a ser feito e um custo previsto para realizar a manutenção. Uma manutenção pode estar utilizando uma ou mais matérias-primas. Uma matéria-prima possui um código, um nome e uma quantidade em estoque. Uma matéria-prima pode estar sendo utilizada em várias manutenções, em uma certa quantidade.

Modelo Conceitual:



Modelo Lógico:



Naturalização:

Nesse caso não foi realizada 3FN e 4FN. 1FN foi utilizado na entidade salão que possuía a posição de todas as obras, sendo assim, foi criada uma nova entidade para receber a chave primária do salão, a chave primária da obra e através desses dois ele indica em qual posição a obra se encontra. Fora isso foi feito a 2FN para Autor e obra ligando uma a outra através de suas chaves primárias, guarda e salão também foi realizado o mesmo, e antes, a entidade manutenção guardava a chave estrangeiras das outras relações da manutenção, com isso, se criou a entidade ManutençãoRestauradorObraMatéria, para guardar as chaves estrangeiras e a entidade manutenção ganhou uma chave primária própria.

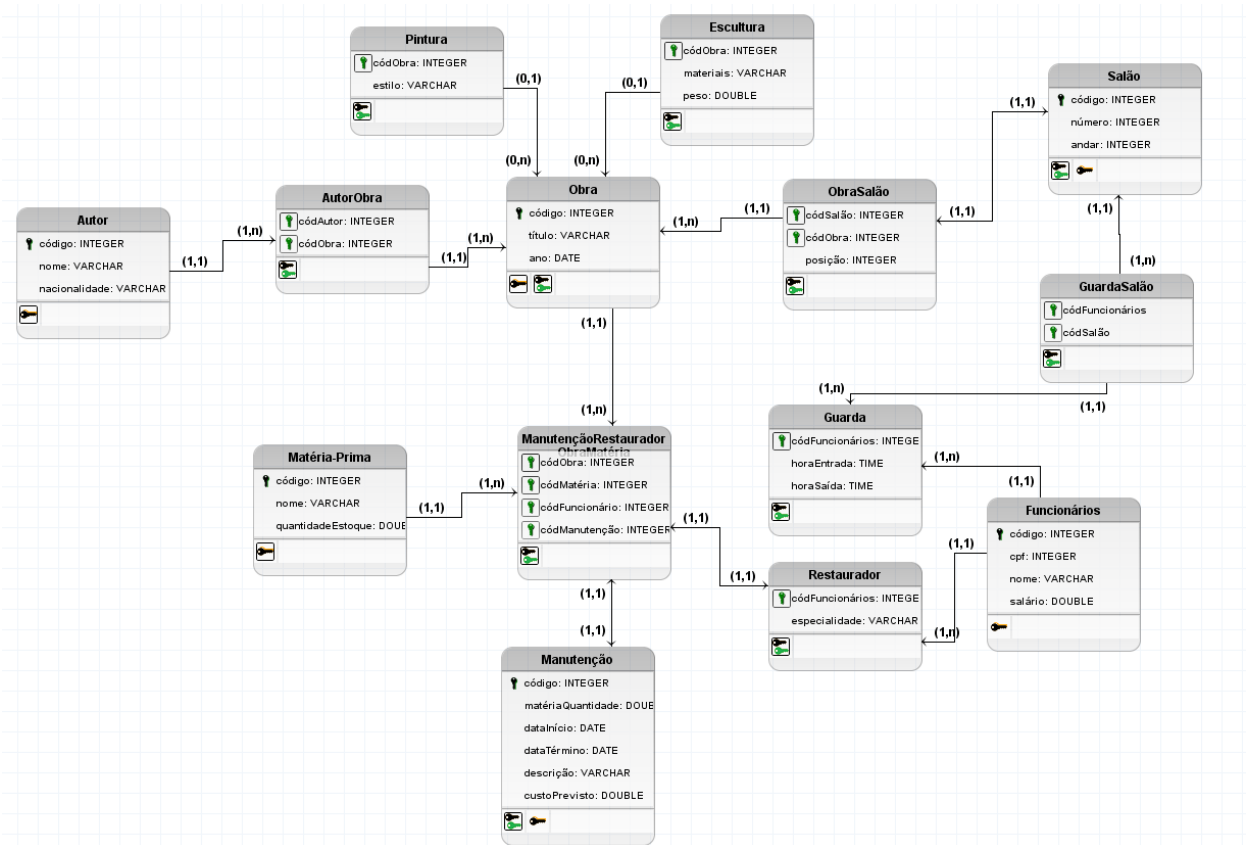


Tabela SQL:

```

CREATE TABLE Autor
(
    código int,
    nome varchar(255),
    nacionalidade varchar(255),
    PRIMARY KEY (código)
);
  
```

```

INSERT INTO Autor
VALUES
(1, 'Roberto', 'chileno'),
(2, 'Ricardo', 'argentino'),
(3, 'João', 'brasileiro'),
(4, 'Pedro', 'francês'),
(5, 'Pedro', 'brasileiro'),
(6, 'Renato', 'alemão'),
(7, 'Sifa', 'brasileiro'),
(8, 'Pedro', 'argentino'),
(9, 'Sofia', 'alemão'),
(10, 'Rebeca', 'libanês'),
(11, 'Ruan', 'brasileiro'),
(12, 'Flávio', 'sírio'),
(13, 'Reinaldo', 'alemão'),
  
```

```
(14, 'Carlos', 'francês'),  
(15, 'Lupan', 'francês');
```

```
CREATE TABLE Obra  
(  
    código int,  
    título varchar(255),  
    ano date,  
    PRIMARY KEY (código)  
);
```

```
INSERT INTO Obra  
VALUES  
(1, 'olinda', "1847-01-01"),  
(2, 'ceu estrelado', "1882-01-01"),  
(3, 'mãe', "1889-01-01"),  
(4, 'mulher na janela', "1875-01-01"),  
(5, 'subterfúgio largata', "1885-01-01"),  
(6, 'lestrade', "1975-01-02"),  
(7, 'de outra', "1997-01-03"),  
(8, 'silencio', "1995-01-02"),  
(9, 'súbito', "1995-01-04"),  
(10, 'alem mar', "1995-01-03"),  
(11, 'o subalterno', "1815-01-01"),  
(12, 'sentido', "1795-01-09"),  
(13, 'orgulho', "1797-01-20"),  
(14, 'sabedoria', "1800-01-09"),  
(15, 'vestigios', "1995-01-09");
```

```
CREATE TABLE AutorObra  
(  
    códAutor int,  
    códObra int,  
    FOREIGN KEY (códAutor) REFERENCES Autor(código),  
    FOREIGN KEY (códObra) REFERENCES Obra(código)  
);
```

```
INSERT INTO AutorObra  
VALUES  
(1, 1),  
(2, 2),  
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),
```

```
(8, 8),
(9, 9),
(10,10),
(11,11),
(12,12),
(13,13),
(14,14),
(15,15);
```

```
CREATE TABLE Pintura
(
    códObra int,
    estilo varchar(255),
    FOREIGN KEY (códObra) REFERENCES Obra(código)
);
```

```
INSERT INTO Pintura
VALUES
(1, 'expressionista'),
(2, 'classica'),
(3, 'figurativa'),
(4, 'impressionista'),
(5, 'cubista'),
(6, 'abstrata'),
(7, 'fotorrealista');
```

```
CREATE TABLE Escultura
(
    códObra int,
    materiais varchar(255),
    peso float,
    FOREIGN KEY (códObra) REFERENCES Obra(código)
);
```

```
INSERT INTO Escultura
VALUES
(8, 'aço', 8.2),
(9, 'cobre', 9.5),
(10, 'aço', 10.3),
(11, 'cobre', 11.9),
(12, 'ferro', 12.1),
(13, 'aço', 13.4),
(14, 'cobre', 14.2),
(15, 'ferro', 15.3);
```

```
CREATE TABLE Salão
```

```
(  
    código int,  
    número int,  
    andar float,  
    PRIMARY KEY (código)  
);
```

INSERT INTO Salão

VALUES

```
(1, 1, 1),  
(2, 2, 1),  
(3, 1, 2),  
(4, 1, 4),  
(5, 2, 4),  
(6, 3, 4),  
(7, 1, 5),  
(8, 2, 5),  
(9, 1, 6),  
(10, 1, 3),  
(11, 2, 3),  
(12, 3, 3),  
(13, 4, 3),  
(14, 5, 3),  
(15, 6, 3);
```

CREATE TABLE ObraSalão

```
(  
    códSalão int,  
    códObra int,  
    posição int,  
    FOREIGN KEY (códSalão) REFERENCES Salão(código),  
    FOREIGN KEY (códObra) REFERENCES Obra(código)  
);
```

INSERT INTO ObraSalão

VALUES

```
(1, 2, 2),  
(2, 1, 5),  
(3, 3, 7),  
(4, 8, 1),  
(5, 4, 14),  
(6, 7, 8),  
(7, 12, 3),  
(8, 9, 12),  
(9, 10, 6),
```

(10, 12, 9),
(11, 6, 10),
(12, 15, 11),
(13, 14, 20),
(14, 11, 4),
(15, 5, 14);

CREATE TABLE Funcionários

(
 código int,
 cpf varchar(255),
 nome varchar(255),
 salário float,
 PRIMARY KEY (código)
);

INSERT INTO Funcionários

VALUES

(1, "156.476.975-58", "Roberto", 1001),
(2, "156.476.975-58", "Sifão", 1002),
(3, "156.476.975-58", "Loto", 1003),
(4, "156.476.975-58", "Rubens", 1000),
(5, "156.476.975-58", "Luiz", 1004),
(6, "156.476.975-58", "Ricardo", 1005),
(7, "156.476.975-58", "Veríssimo", 1010),
(8, "156.476.975-58", "Letícia", 1000),
(9, "156.476.975-58", "Isabel", 1010),
(10, "156.476.975-58", "Daniel", 1100),
(11, "156.476.975-58", "Luan", 1230),
(12, "156.476.975-58", "Cláudia", 1470),
(13, "156.476.975-58", "Letícia", 1450),
(14, "156.476.975-58", "Ana", 1950),
(15, "156.476.975-58", "Guilherme", 1000);

CREATE TABLE Guarda

(
 códFuncionários int,
 horaEntrada time,
 horaSaída time,
 FOREIGN KEY (códFuncionários) REFERENCES Funcionários(código)
);

INSERT INTO Guarda

VALUES

(8, '01:00:00', '10:00:00'),
(9, '02:00:00', '11:00:00'),

```
(10, '03:00:00', '12:00:00'),  
(11, '04:00:00', '13:00:00'),  
(12, '05:00:00', '14:00:00'),  
(13, '06:00:00', '15:00:00'),  
(14, '07:00:00', '16:00:00'),  
(15, '08:00:00', '17:00:00');
```

```
CREATE TABLE GuardaSalão  
(  
    códFuncionários int,  
    códSalão int,  
    FOREIGN KEY (códFuncionários) REFERENCES Funcionários(código),  
    FOREIGN KEY (códSalão) REFERENCES Salão(código)  
);
```

```
INSERT INTO GuardaSalão  
VALUES  
(1, 2),  
(2, 4),  
(3, 2),  
(4, 1),  
(5, 4),  
(6, 7),  
(7, 12),  
(8, 7),  
(9, 10),  
(10, 12),  
(11, 6),  
(12, 1),  
(13, 2),  
(14, 3),  
(15, 9);
```

```
CREATE TABLE Restaurador  
(  
    códFuncionários int,  
    especialidade varchar(255),  
    FOREIGN KEY (códFuncionários) REFERENCES Funcionários(código)  
);
```

```
INSERT INTO Restaurador  
VALUES  
(1, 'livros raros'),  
(2, 'livros raros'),  
(3, 'livros raros'),  
(4, 'livros raros'),
```

```
(5, 'livros raros'),  
(6, 'livros gregos'),  
(7, 'livros gregos');
```

```
CREATE TABLE MatériaPrima  
(  
    código int,  
    nome varchar(255),  
    quantidade float,  
    PRIMARY KEY (código)  
);
```

```
INSERT INTO MatériaPrima  
VALUES
```

```
(1, 'tecido', 2.2),  
(2, 'escova', 1.4),  
(3, 'tecido', 3.1),  
(4, 'escova', 8.3),  
(5, 'tecido', 4.6),  
(6, 'escova', 7.3),  
(7, 'tecido', 12.6),  
(8, 'escova', 9.8),  
(9, 'tecido', 10.1),  
(10, 'tecido', 12.2),  
(11, 'pano', 6.4),  
(12, 'tecido', 15.5),  
(13, 'pano', 14.2),  
(14, 'escova', 11.7),  
(15, 'pano', 5.8);
```

```
CREATE TABLE Manutenção  
(  
    código int,  
    matériaQuantidade float,  
    dataInício date,  
    dataTérmino date,  
    descrição varchar(255),  
    custoPrevisto float,  
    PRIMARY KEY (código)  
);
```

```
INSERT INTO Manutenção  
VALUES
```

```
(1, 2, "2020-01-09", "2021-01-09", 'limpar', 1000),  
(2, 1, "2020-02-09", "2021-02-09", 'limpar', 1000),  
(3, 3, "2020-03-09", "2021-03-09", 'limpar', 1000),
```

```
(4, 8, "2020-04-09", "2021-04-09", 'limpar', 1000),
(5, 4, "2020-05-12", "2021-05-01", 'limpar', 1000),
(6, 7, "2020-01-10", "2021-01-02", 'limpar', 1000),
(7, 12, "2020-02-10", "2021-02-05", 'limpar', 1000),
(8, 9, "2020-03-10", "2021-03-05", 'restaurar', 1000 ),
(9, 10, "2020-04-10", "2021-04-04", 'restaurar', 1000),
(10,12, "2020-05-12", "2021-05-05", 'restaurar', 1000),
(11,6, "2020-01-11", "2021-01-06", 'restaurar', 1000),
(12,15, "2020-02-15", "2021-02-07", 'restaurar', 1000),
(13,14, "2020-03-12", "2021-04-08", 'restaurar', 1000),
(14, 11, "2020-04-15", "2021-05-09", 'restaurar', 1000),
(15,5, "2020-05-01", "2021-01-01", 'restaurar', 1000);
```

```
CREATE TABLE ManutençãoRestauradorObraMatéria
(
    códObra int,
    códMatéria int,
    códFuncionário int,
    códManutenção int,
    FOREIGN KEY (códObra) REFERENCES Obra(código),
    FOREIGN KEY (códMatéria) REFERENCES MatériaPrima(código),
    FOREIGN KEY (códFuncionário) REFERENCES Funcionários(código),
    FOREIGN KEY (códManutenção) REFERENCES Manutenção(código)
);
```

```
INSERT INTO ManutençãoRestauradorObraMatéria
VALUES
(1, 1, 15, 1),
(2, 2, 14, 2),
(3, 3, 13, 3),
(4, 4, 12, 4),
(5, 5, 11, 5),
(6, 6, 10, 6),
(7, 7, 9, 7),
(8, 8, 8, 8),
(9, 9, 7, 9),
(10, 10, 6, 10),
(11, 11, 5, 11),
(12, 12, 4, 12),
(13, 13, 3, 13),
(14, 14, 2, 14),
(15, 15, 1, 15);
```

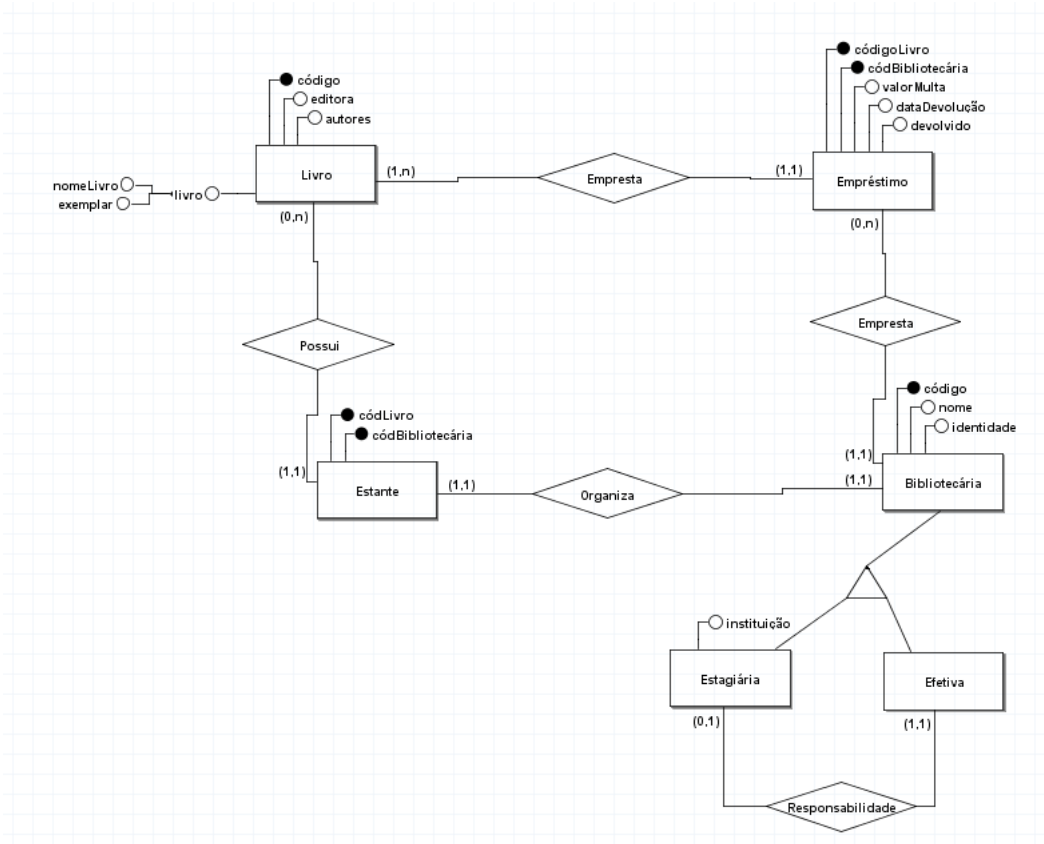

- a) Faça uma consulta que exiba todas as obras do terceiro andar do museu que tenham entre 200 a 300 anos de idade e foram restauradas no ano de 2020 por restauradores especialistas em livros raros que tenham salários menores ou iguais a de um segurança.

```
SELECT *
FROM Obra
WHERE ano < "1821%" AND ano > "1720%" AND código IN
(SELECT ManutençãoRestauradorObraMatéria.códObra
FROM ManutençãoRestauradorObraMatéria
WHERE códFuncionário IN (SELECT Funcionários.código
FROM Restaurador, Funcionários
WHERE Restaurador.códFuncionários = Funcionários.código and
Restaurador.especialidade = "livros raros" and
Funcionários.salário < ANY (SELECT Funcionários.salário
FROM Guarda, Funcionários
WHERE Guarda.códFuncionários = Funcionários.código)) AND
ManutençãoRestauradorObraMatéria.códManutenção IN
(SELECT Manutenção.código
FROM Manutenção
WHERE dataInício > "2020-01-01" and dataInício < "2021-01-01"))
AND Obra.código IN
(SELECT ObraSalão.códObra
FROM ObraSalão
WHERE códObra IN (SELECT Salão.código
FROM Salão
WHERE andar = 3))
```

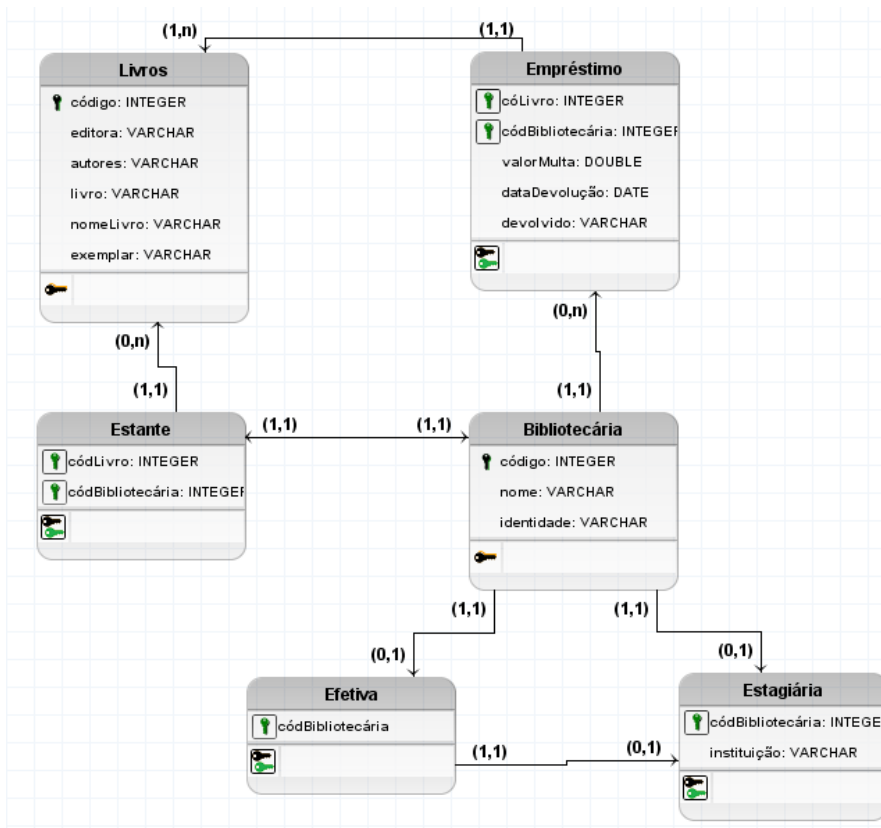
Caso 3

Um livro tem vários exemplares na biblioteca, como por exemplo, Liv1-Ex1, Liv1-Ex2, Liv2-Ex1. Exemplares estão dispostos em estantes. São mantidos dados detalhados sobre autores e editoras dos livros para fins de consulta. Na biblioteca trabalham bibliotecárias. Cada estante é periodicamente organizada por uma única bibliotecária. Bibliotecárias também realizam empréstimos de exemplares para clientes. Empréstimos cadastrados no BD devem conter a data da devolução e o valor diário da multa, permanecendo no BD até o cliente entregar o exemplar. A bibliotecária que realizou o empréstimo também é relevante de ser mantido no BD. Algumas bibliotecárias são estagiárias. Uma bibliotecária estagiária está sempre sob a responsabilidade de uma bibliotecária efetiva. Deve-se saber também a instituição de ensino da qual a estagiária vem. Defina os atributos que julga relevantes para os fatos identificados.

Modelo conceitual:



Modelo lógico:



Naturalização:

Nesse modelo não foi necessário utilizar a 1FN, 3FN e 4FN, sendo feito apenas a 2FN onde a estante ganhou um código próprio e agora ela grava qual livro pertence a ela. O empréstimo recebeu uma chave primária própria e foi criada uma entidade para registrar a relação de EmpréstimoBibliotecáriaLivro. E foi criada uma nova entidade para registrar a relação de EstagiáriaEfetiva.

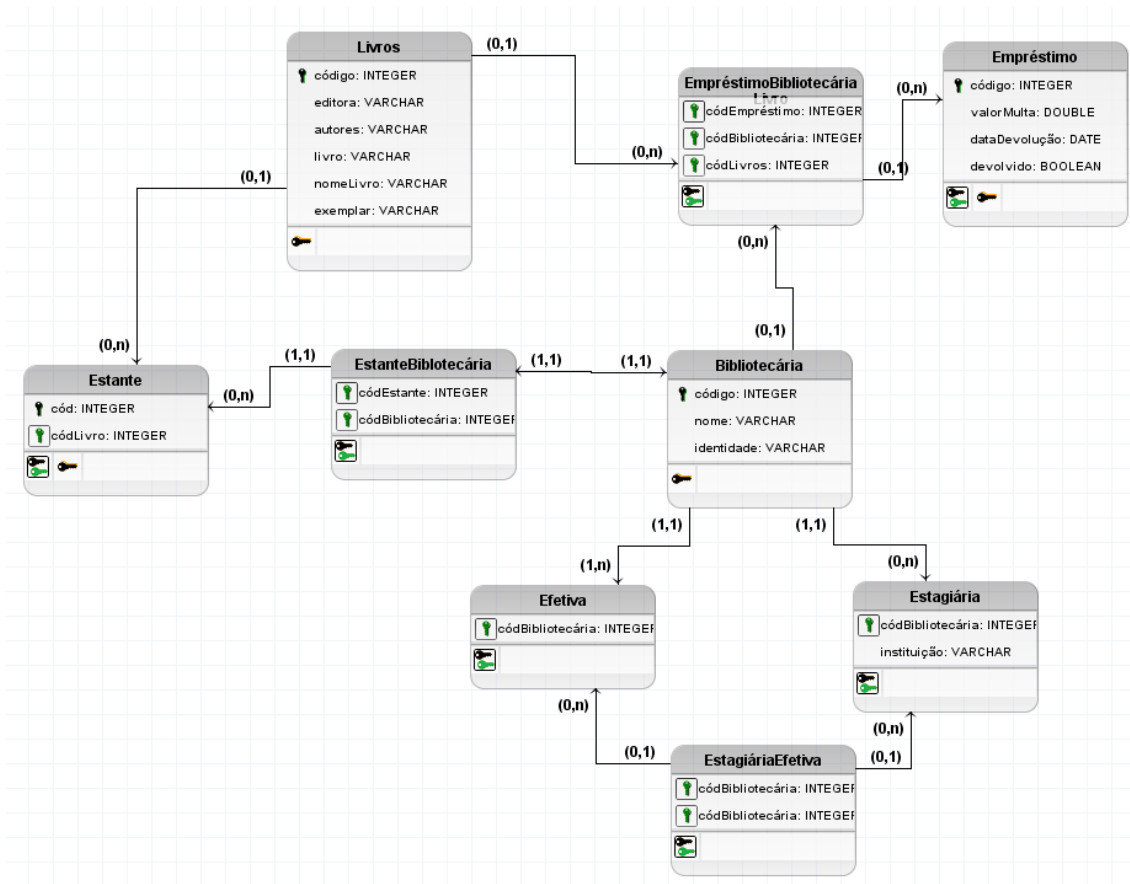


Tabela SQL:

CREATE TABLE Livros

```

(
  código int,
  editora varchar(255),
  autores varchar(255),
  livro varchar(255),
  nomeLivro varchar(255),
  exemplar varchar(255),
  PRIMARY KEY (código)
);
  
```

INSERT INTO Livros

VALUES

```

(1, "moderna", "Lucas", "Matemática 1", "Matemática", "1"),
(2, "moderna", "Roberto", "Matemática 2", "Matemática", "2"),
(3, "moderna", "Leonardo", "Matemática 3", "Matemática", "3"),
(4, "moderna", "Rogério", "Matemática 4", "Matemática", "4"),
(5, "moderna", "Arthur", "Matemática 5", "Matemática", "5"),
(6, "moderna", "Reinaldo", "Matemática 6", "Matemática", "6"),
(7, "moderna", "Lucas", "Matemática 7", "Matemática", "7"),
(8, "moderna", "Roberto", "Matemática 8", "Matemática", "8"),
(9, "moderna", "Rogério", "Filosofia 1", "Filosofia", "1"),
  
```

```
(10, "moderna", "Arthur", "Filosofia 2", "Filosofia", "2"),  
(11, "moderna", "Reinaldo", "Filosofia 3", "Filosofia", "3"),  
(12, "moderna", "Lucas", "Filosofia 4", "Filosofia", "4"),  
(13, "moderna", "Rogério", "Filosofia 5", "Filosofia", "5"),  
(14, "moderna", "Roberto", "Filosofia 6", "Filosofia", "6"),  
(15, "moderna", "Leonardo", "Filosofia 7", "Filosofia", "7");
```

```
CREATE TABLE Estante  
(  
    código int,  
    códLivro int,  
    PRIMARY KEY (código),  
    FOREIGN KEY (códLivro) REFERENCES Livros(código)  
);
```

```
INSERT INTO Estante
```

```
VALUES
```

```
(1, 1),  
(2, 2),  
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),  
(8, 8),  
(9, 9),  
(10, 10),  
(11, 11),  
(12, 12),  
(13, 13),  
(14, 14),  
(15, 15);
```

```
CREATE TABLE Bibliotecária
```

```
(  
    código int,  
    nome varchar(255),  
    identidade varchar(255),  
    PRIMARY KEY (código)  
);
```

```
INSERT INTO Bibliotecária
```

```
VALUES
```

```
(1, "Roberta", "27.498.679-5"),  
(2, "Letícia", "27.498.679-5"),  
(3, "Filomena", "27.498.679-5"),
```

```
(4, "Juçara", "27.498.679-5"),
(5, "Laís", "27.498.679-5"),
(6, "Isabel", "27.498.679-5"),
(7, "Leia", "27.498.679-5"),
(8, "Gaida", "27.498.679-5"),
(9, "Irani", "27.498.679-5"),
(10, "Rosimeri", "27.498.679-5"),
(11, "Ariel", "27.498.679-5"),
(12, "Jéssica", "27.498.679-5"),
(13, "Helena", "27.498.679-5"),
(14, "Alice", "27.498.679-5"),
(15, "Marcia", "27.498.679-5");
```

```
CREATE TABLE EstanteBibliotecária
(
    códEstante int,
    códBibliotecária int,
    FOREIGN KEY (códEstante) REFERENCES Estante(código),
    FOREIGN KEY (códBibliotecária) REFERENCES Bibliotecária(código)
);
```

```
INSERT INTO EstanteBibliotecária
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13),
(14, 14),
(15, 15);
```

```
CREATE TABLE Empréstimo
(
    código int,
    ValorMulta float,
    dataDevolução date,
    devolvido varchar(5),
    PRIMARY KEY (código)
```

);

INSERT INTO Empréstimo

VALUES

(1, 0.5, "2021-12-09", "true"),
(2, 0.6, "2021-12-19", "false"),
(3, 0.1, "2021-10-08", "true"),
(4, 0.2, "2021-11-11", "true"),
(5, 0.4, "2021-10-10", "true"),
(6, 0.7, "2021-09-19", "true"),
(7, 0.4, "2021-12-16", "false"),
(8, 0.5, "2018-12-11", "false"),
(9, 0.7, "2021-07-11", "true"),
(10, 0.8, "2021-06-11", "true"),
(11, 0.4, "2021-11-11", "true"),
(12, 0.1, "2021-12-20", "false"),
(13, 0.4, "2021-12-28", "false"),
(14, 0.8, "2021-12-29", "false"),
(15, 0.5, "2021-12-25", "false");

CREATE TABLE EmpréstimoBibliotecáriaLivros

(
 códEmpréstimo int,
 códBibliotecária int,
 códLivros int,
 FOREIGN KEY (códEmpréstimo) REFERENCES Empréstimo(código),
 FOREIGN KEY (códBibliotecária) REFERENCES Bibliotecária(código),
 FOREIGN KEY (códLivros) REFERENCES Livros(código)
);

INSERT INTO EmpréstimoBibliotecáriaLivros

VALUES

(1, 1, 1),
(2, 1, 2),
(3, 1, 3),
(4, 1, 4),
(5, 2, 5),
(6, 2, 6),
(7, 3, 7),
(8, 4, 8),
(9, 5, 9),
(10, 2, 10),
(11, 3, 11),
(12, 4, 12),
(13, 5, 13),
(14, 6, 14),

```
(15, 7, 15);
```

```
CREATE TABLE Efetiva  
(  
    códBibliotecária int,  
    FOREIGN KEY (códBibliotecária) REFERENCES Bibliotecária(código)  
);
```

```
INSERT INTO Efetiva
```

```
VALUES
```

```
(1),
```

```
(2),
```

```
(3),
```

```
(4),
```

```
(5),
```

```
(6),
```

```
(7),
```

```
(8);
```

```
CREATE TABLE Estagiária
```

```
(  
    códBibliotecária int,  
    instituição varchar(255),  
    FOREIGN KEY (códBibliotecária) REFERENCES Bibliotecária(código)  
);
```

```
INSERT INTO Estagiária
```

```
VALUES
```

```
(9, "ufrj"),
```

```
(10, "ufrj"),
```

```
(11, "ufrj"),
```

```
(12, "ufrj"),
```

```
(13, "ufrj"),
```

```
(14, "ufrj"),
```

```
(15, "ufrj");
```

```
CREATE TABLE EstagiáriaEfetiva
```

```
(  
    códEfetiva int,  
    códEstagiária int,  
    FOREIGN KEY (códEfetiva) REFERENCES Efetiva(códBibliotecária),  
    FOREIGN KEY (códEstagiária) REFERENCES Estagiária(códBibliotecária)  
);
```

```
INSERT INTO EstagiáriaEfetiva
```

```
VALUES
```


(1, 9),
(2, 10),
(3, 11),
(4, 12),
(5, 13),
(6, 14),
(7, 15);

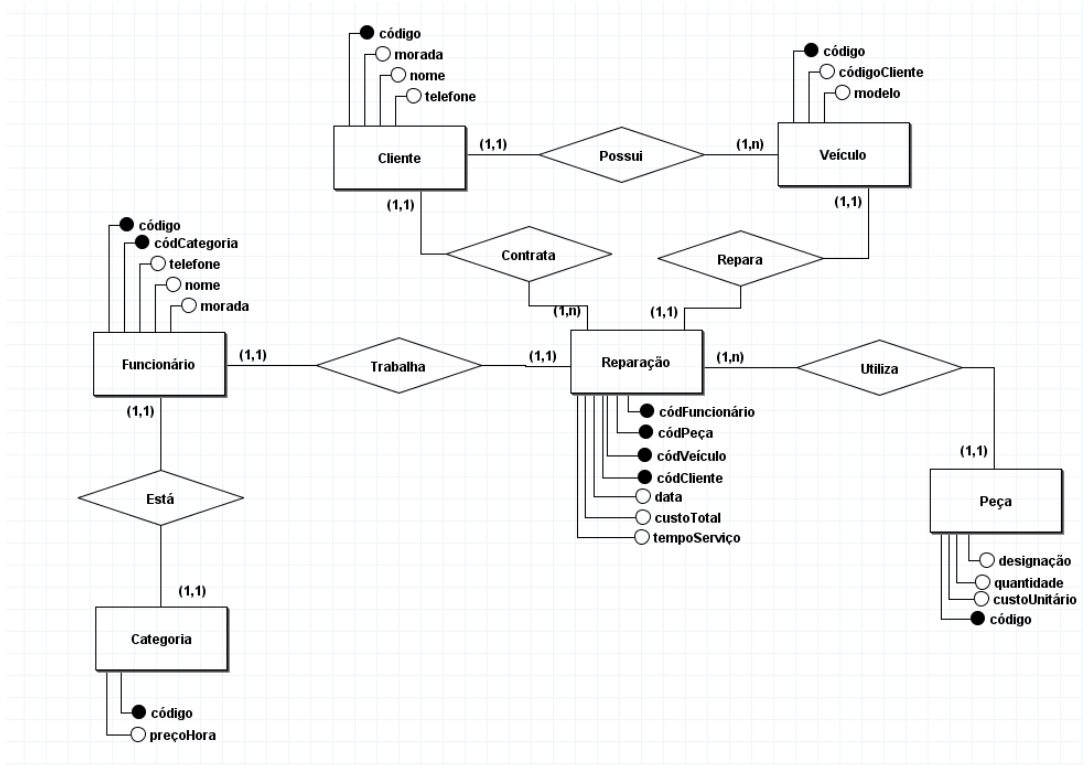
- a) Faça um consulte o banco e liste todos os empréstimos cujos livros não foram devolvidos

```
SELECT *  
FROM Empréstimo  
WHERE (Empréstimo.dataDevolução < "2021-12-13" AND Empréstimo.devolvido =  
"false") OR (Empréstimo.dataDevolução > "2021-12-13")
```

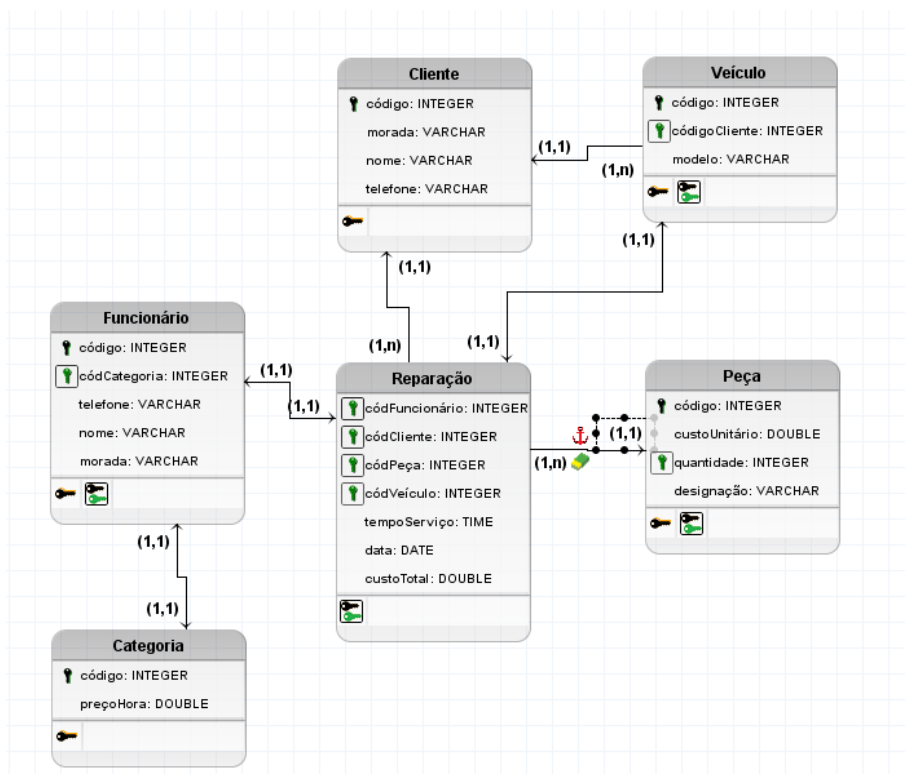
Caso 4

Uma empresa de venda de automóveis pretende implementar um sistema para gerir a informação relativa às operações de oficina efetuadas nos veículos dos seus clientes. O sistema de informação deverá permitir manter um registo histórico de todas as reparações efetuadas. A empresa pretende manter dos clientes a seguinte informação: código de identificação, nome, morada e telefone. A empresa pretende também saber os veículos que um dado cliente teve ou tem e as datas em que foram adquiridos. Dos funcionários da oficina a empresa pretende manter a seguinte informação: código de identificação, nome, morada, telefone e categoria. O custo horário de mão de obra depende da categoria do funcionário e é definido através de uma tabela que é atualizada regularmente. A empresa pretende saber para cada reparação: qual o veículo, qual o cliente, a data em que foi efetuada e o custo total da reparação. A empresa pretende também armazenar para cada reparação as peças que foram utilizadas e o seu preço, bem como o tempo de mão de obra gasto por cada funcionário e o respectivo custo. A empresa pretende manter para as peças em armazém a seguinte informação: código de identificação, designação, custo unitário e quantidade em armazém.

Modelo conceitual:



Modelo lógica:



Normalização:

Nessa normalização, não foi utilizada 1FN, 3FN e 4FN. 1FN não foi utilizado pois só é permitido cadastrar 1 endereço por cliente, e a relação entre cliente e vários veículos, cada veículo estará relacionado com o seu cliente, não criando nenhuma cédula multivalorada. Já para 2FN, foi criada uma nova entidade para a relação ClienteVeículo, para CategoriaFuncionário, com categoria ganhando sua própria chave primária, e para manutenção, que também ganhou sua própria chave primária e agora existe uma entidade para a relação ReparaçãoClienteFuncionárioPeça.

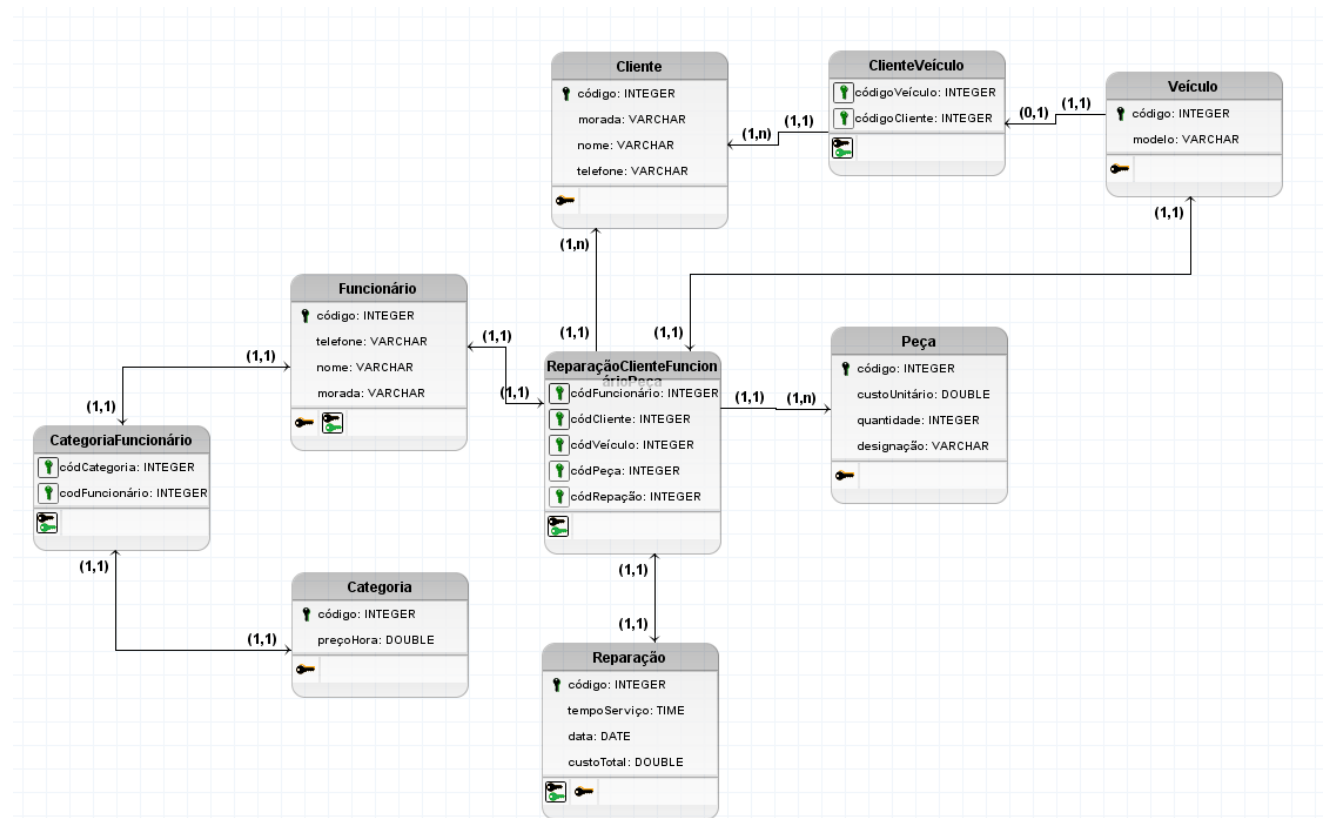


Tabela SQL:

CREATE TABLE Funcionário

```
(
  código int,
  telefone int,
  nome varchar(255),
  morada varchar(255),
  PRIMARY KEY (código)
);
```

INSERT INTO Funcionário

VALUE

```
(1, "26919946", "Lucas", "Rua Jequitiba"),
(2, "26919946", "Caio", "Rua Jequitiba"),
(3, "26919946", "Ayrton", "Rua Jequitiba"),
(4, "26919946", "Daniel", "Rua Jequitiba"),
```

```
(5, "26919946", "luan", "Rua Jequitiba"),
(6, "26919946", "isabel", "Rua Jequitiba"),
(7, "26919946", "leticia", "Rua Jequitiba"),
(8, "26919946", "claudia", "Rua Jequitiba"),
(9, "26919946", "ana", "Rua Jequitiba"),
(10, "26919946", "guilherme", "Rua Jequitiba"),
(11, "26919946", "gustavo", "Rua Jequitiba"),
(12, "26919946", "marcelo", "Rua Jequitiba"),
(13, "26919946", "roberto", "Rua Jequitiba"),
(14, "26919946", "paulo", "Rua Jequitiba"),
(15, "26919946", "gabriel", "Rua Jequitiba");
```

```
CREATE TABLE Categoria
```

```
(
    código int,
    preçoHora float,
    PRIMARY KEY (código)
);
```

```
INSERT INTO Categoria
```

```
VALUE
(1, 10.5),
(2, 7.5),
(3, 4.5),
(4, 8.5),
(5, 9.5),
(6, 11.5),
(7, 12.5),
(8, 9.5),
(9, 1.5),
(10, 9.5),
(11, 4.5),
(12, 2.5),
(13, 7.5),
(14, 4.5),
(15, 5.5);
```

```
CREATE TABLE CategoriaFuncionário
```

```
(
    códCategoria int,
    códFuncionário int,
    FOREIGN KEY (códCategoria) REFERENCES Categoria(código),
    FOREIGN KEY (códFuncionário) REFERENCES Funcionário(código)
);
```

```
INSERT INTO CategoriaFuncionário
```

VALUE

(1, 1),
(2, 2),
(3, 1),
(4, 4),
(5, 1),
(6, 6),
(7, 7),
(8, 2),
(9, 6),
(10, 10),
(11, 2),
(12, 8),
(13, 3),
(14, 9),
(15, 1);

CREATE TABLE Cliente

(
 código int,
 nome varchar(255),
 morada varchar(255),
 telefone varchar(255),
 PRIMARY KEY (código)
);

INSERT INTO Cliente

VALUE

(1, "lucas", "Rua Jequitiba", "26919946"),
(2, "caio", "Rua Jequitiba", "26919946"),
(3, "ayrton", "Rua Jequitiba", "26919946"),
(4, "daniel", "Rua Jequitiba", "26919946"),
(5, "luan", "Rua Jequitiba", "26919946"),
(6, "isabel", "Rua Jequitiba", "26919946"),
(7, "leticia", "Rua Jequitiba", "26919946"),
(8, "claudia", "Rua Jequitiba", "26919946"),
(9, "ana", "Rua Jequitiba", "26919946"),
(10, "guilherme", "Rua Jequitiba", "26919946"),
(11, "gustavo", "Rua Jequitiba", "26919946"),
(12, "marcelo", "Rua Jequitiba", "26919946"),
(13, "roberto", "Rua Jequitiba", "26919946"),
(14, "paulo", "Rua Jequitiba", "26919946"),
(15, "gabriel", "Rua Jequitiba", "26919946");

CREATE TABLE Veículo

(

```
código int,  
modelo varchar(255),  
PRIMARY KEY (código)  
);
```

```
INSERT INTO Veículo
```

```
VALUE
```

```
(1, "Gol"),  
(2, "Uno"),  
(3, "Palio"),  
(4, "Fox"),  
(5, "Siena"),  
(6, "Celta"),  
(7, "Voyage"),  
(8, "HB20"),  
(9, "Corsa Sedan"),  
(10, "Onix"),  
(11, "Sanderó"),  
(12, "Fiesta"),  
(13, "Cobalt"),  
(14, "Ka"),  
(15, "Corolla");
```

```
CREATE TABLE ClienteVeículo
```

```
(  
  códCliente int,  
  códVeículo int,  
  FOREIGN KEY (códCliente) REFERENCES Cliente(código),  
  FOREIGN KEY (códVeículo) REFERENCES Veículo(código)  
);
```

```
INSERT INTO ClienteVeículo
```

```
VALUE
```

```
(1, 1),  
(2, 2),  
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),  
(8, 8),  
(9, 9),  
(10, 10),  
(11, 11),  
(12, 12),  
(13, 13),
```

```
(14, 14),  
(15, 15);
```

```
CREATE TABLE Peça  
(  
    código int,  
    custoUnitário float,  
    quantidade int,  
    designação varchar(255),  
    PRIMARY KEY (código)  
);
```

```
INSERT INTO Peça  
VALUE  
(1, 7.8, 10, "Filtro de óleo"),  
(2, 6.2, 10, "Velas de ignição"),  
(3, 3.8, 10, "Pistão"),  
(4, 5.2, 10, "Caixa de câmbio"),  
(5, 8.8, 10, "Suspensão"),  
(6, 1.8, 10, "Filtro de óleo"),  
(7, 2.4, 10, "Velas de ignição"),  
(8, 9.3, 10, "Pistão"),  
(9, 1.4, 10, "Caixa de câmbio"),  
(10, 2.9, 10, "Suspensão"),  
(11, 2.1, 10, "Filtro de óleo"),  
(12, 3.5, 10, "Velas de ignição"),  
(13, 1.2, 10, "Pistão"),  
(14, 2.7, 10, "Caixa de câmbio"),  
(15, 1.2, 10, "Suspensão");
```

```
CREATE TABLE Reparação  
(  
    código int,  
    tempoServiço time,  
    data date,  
    custoTotal float,  
    PRIMARY KEY (código)  
);
```

```
INSERT INTO Reparação  
VALUES  
(1, "01:00:00", "2021-01-01", 5500.00),  
(2, "01:00:00", "2021-01-01", 7800.00),  
(3, "01:00:00", "2021-01-01", 3200.00),  
(4, "01:00:00", "2021-01-01", 9000.00),  
(5, "01:00:00", "2021-01-01", 1000.00),
```

```
(6, "01:00:00", "2021-01-01", 3000.00),
(7, "01:00:00", "2021-01-01", 6000.00),
(8, "01:00:00", "2021-01-01", 8000.00),
(9, "01:00:00", "2021-01-01", 5900.00),
(10, "01:00:00", "2021-01-01", 5000.00),
(11, "01:00:00", "2021-01-01", 5200.00),
(12, "01:00:00", "2021-01-01", 5300.00),
(13, "01:00:00", "2021-01-01", 3000.00),
(14, "01:00:00", "2021-01-01", 2000.00),
(15, "01:00:00", "2021-01-01", 9000.00);
```

```
CREATE TABLE ReparaçãoClienteFuncionárioPeça
(
    códReparação int,
    códCliente int,
    códFuncionário int,
    códPeça int,
    códVeículo int,
    FOREIGN KEY (códReparação) REFERENCES Reparação(código),
    FOREIGN KEY (códCliente) REFERENCES Cliente(código),
    FOREIGN KEY (códFuncionário) REFERENCES Funcionário(código),
    FOREIGN KEY (códPeça) REFERENCES Peça(código),
    FOREIGN KEY (códVeículo) REFERENCES Veículo(código)
);
```

```
INSERT INTO ReparaçãoClienteFuncionárioPeça
VALUE
(1, 1, 1, 1, 1),
(2, 2, 1, 1, 3),
(3, 1, 1, 1, 2),
(4, 3, 1, 1, 5),
(5, 2, 1, 1, 3),
(6, 2, 1, 1, 3),
(7, 1, 1, 1, 1),
(8, 3, 1, 1, 5),
(9, 2, 1, 1, 4);
```

- b) Faça uma consulta sobre todos os automóveis que foram recuperados mais de uma vez?

```
SELECT *
FROM Veículo
WHERE Veículo.código =
(SELECT códVeículo
FROM ReparaçãoClienteFuncionárioPeça
GROUP BY códVeículo)
```



```
ORDER BY COUNT(códVeículo) DESC  
LIMIT 1)
```

- c) Faça uma consulta que liste em ordem crescente as 10 recuperações de automóveis mais caras realizadas pela oficina

```
SELECT *  
FROM Reparação  
ORDER BY custoTotal DESC  
LIMIT 10
```