Sistema Fiep

FIEP SESI SENAI IEL

Banco de Dados

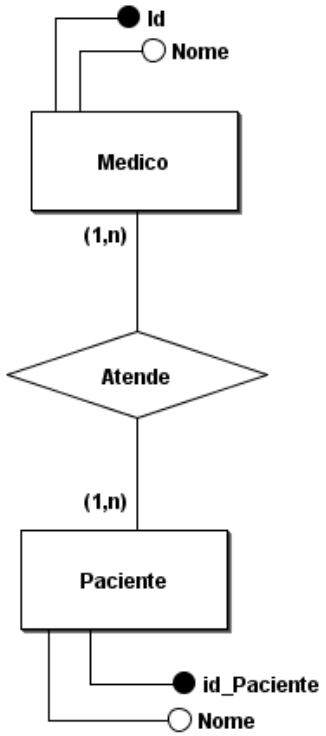
Modelo Físico de Banco de Dados

Conteúdo

- Resumo e definição em estrutura física.
- Características modelos: OO, UML e ER.
- Modelo Físico.
- *Structured Query Language – SQL*.
- Linguagem de Definição de Dados - DDL
  - Criação de Tabelas;
  - Alteração de Tabelas.
- Script de criação e alteração das tabelas.

Modelos de Dados

Conceitual (DER)



Lógico (MER)

Físico (depende do SGBD)

```
CREATE TABLE Paciente
(
    IDPACIENTE      INT IDENTITY PRIMARY KEY NOT NULL,
    NOMEPACIENTE    VARCHAR(50),
    TELEFONE        VARCHAR(10)
)

CREATE TABLE Consulta
(
    IDCONSULTA      INT IDENTITY PRIMARY KEY NOT NULL,
    IDMEDICO        INT NOT NULL,
    IDPACIENTE      INT NOT NULL,
    DATACONSULTA    DATETIME,
    HORAINICIO      DATETIME,
    HORAFIM         DATETIME,
    OBSERVACOES     VARCHAR(MAX),
    ATIVO           BIT
)
```

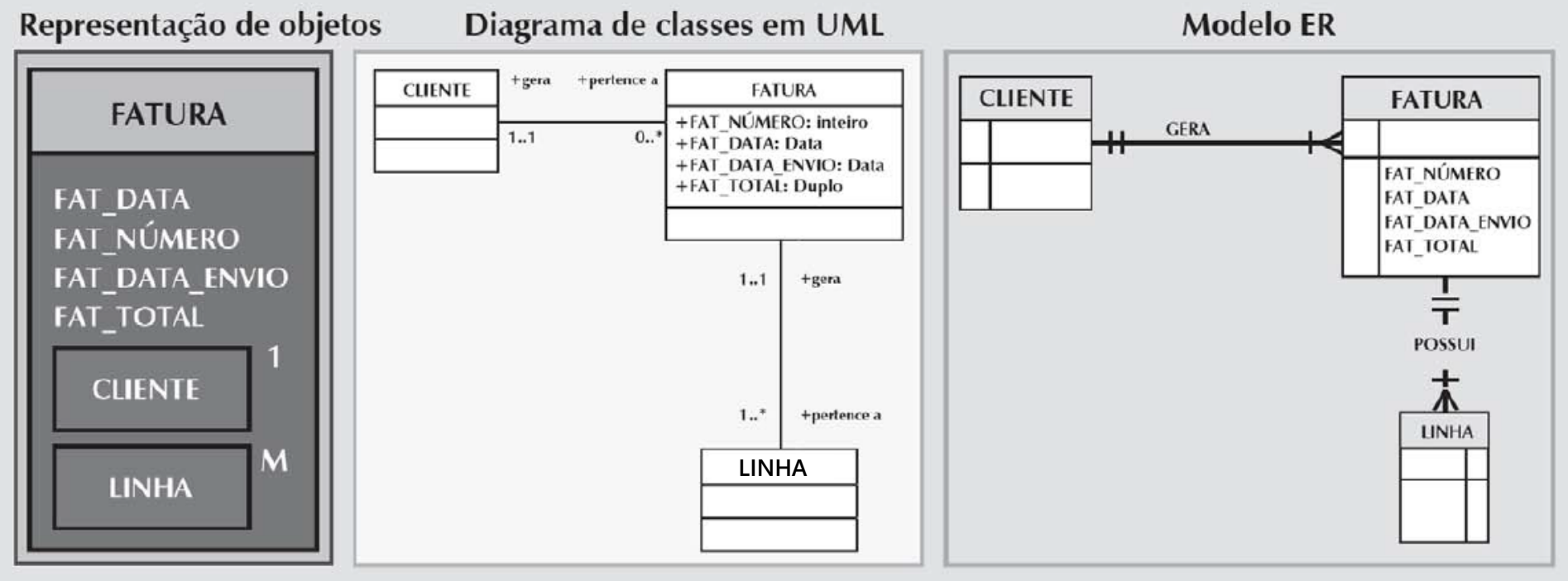
Resumo de Definições em Estrutura Física

Termo	Definição
Entidade	Algo de interesse para analistas, projetistas e usuários do banco de dados. Alguns exemplos: clientes, pessoas, localizações geográficas etc.
Coluna	Para definição de tipo e armazenamento de dado em uma tabela.
Linha	Um conjunto de colunas que, coletivamente, descreve de forma completa uma entidade ou alguma ação em uma entidade. Também chamada de registro.
Tabela	Conjunto de linhas, mantidas em memória volátil (não-persistente) ou em armazenamento permanente (persistente).
Conjunto-resultado	Outro nome para uma tabela não-persistente, geralmente o resultado de uma consulta SQL.
Chave primária	Uma ou mais colunas (neste caso, chave primária composta) que podem ser usadas como identificador único de cada linha em uma tabela.
Chave estrangeira	Uma ou mais colunas que podem ser usadas em conjunto para identificar uma única linha em outra tabela.

Adaptado de: Beaulieu, Alan. Aprendendo SQL . Novatec Editora.

Características modelos: OO, UML e ER

- A representação de objetos da FATURA inclui todos os objetos relacionados na mesma “caixa”.
- O diagrama de classes em UML utiliza três classes de objeto distintas (CLIENTE, FATURA e LINHA) e dois relacionamentos.
- O modelo de entidade e relacionamento utiliza três entidades separadas e dois relacionamentos para representar a modelagem do problema.



Modelo Físico

O Modelo Físico opera nos níveis mais baixos de abstração, descrevendo o modo como os dados são salvos em meios de armazenamento (discos).

Este modelo exige a definição tanto de dispositivos de armazenamento físico como dos métodos de acesso necessários para chegar aos dados armazenados nesses dispositivos, ou seja, isso torna o processo dependente tanto do software como do hardware.

As estruturas de armazenamento utilizadas são dependentes do software (SGBD e Sistema Operacional) e dos tipos de dispositivos de armazenamentos, por exemplo: *storage.*

Structured Query Language (SQL)

Quando o sistema R (https://en.wikipedia.org/wiki/IBM_System_R) estava sendo desenvolvido na IBM, pesquisadores da IBM desenvolveram a linguagem SEQUEL, primeira linguagem de acesso para Sistemas Gerenciadores de Banco de Dados Relacionais.

- O ANSI *(American National Standards Institute)*, em 1986, lançou o padrão da linguagem SQL, o SQL-86.
- Teve sua evolução passando por alterações em 1989 e, em 1992, foi lançada a SQL-92 ou SQL2.
- Um novo padrão, chamado de SQL-99 ou SQL3 foi lançado no ano de 2000. Foi o primeiro padrão a estender a linguagem para permitir a utilização de tipos de dados complexos e a incorporar características da orientação a objetos.
- Depois de uma considerável revisão do padrão SQL3, foi lançada a SQL-2003. Nesta versão foi adicionada uma nova parte ligada ao tratamento de XML.

Structured Query Language (SQL)

A SQL (Structured Query Language – Linguagem Estruturada de Consulta) envolve três partes:

1. Interface do usuário – Permite que o usuário final interaja com os dados.
2. Conjunto de tabelas armazenadas no banco de dados – i) cada tabela é independente uma da outra; ii) as linhas de tabelas diferentes são relacionadas com base em valores comuns de atributos comuns.
3. Mecanismo de SQL – executa todas as consultas ou solicitações de dados.

Structured Query Language (SQL)

A linguagem SQL é dividida nos seguintes tipos de grupos de comandos:

DDL ***(Data Definition Language)*** – permite a criação de componentes no banco de dados, como tabelas, índices, etc. Por exemplo: **`CREATE TABLE; ALTER TABLE; DROP TABLE; CREATE INDEX; ALTER INDEX; DROP INDEX.`**

DML ***(Data Manipulation Language)*** – permite a manipulação dos dados armazenados em uma banco de dados. Por exemplo: **`INSERT; DELETE; UPDATE.`**

DQL ***(Data Query Language)*** – permite recuperar (consultar) dados do banco de dados. Por exemplo: **`SELECT.`**

DCL ***(Data Control Language)*** – provê a segurança interna do banco de dados. Por exemplo: **`CREATE USER; ALTER USER; GRANT; REVOKE; CREATE SCHEMA.`**

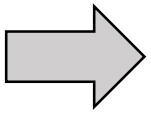
DTL ***(Data Transaction Language)*** – integra a parte de controle de transações do banco de dados. Por exemplo: **`BEGIN TRANSACTION; COMMIT; ROLLBACK.`**

DDL (Data Definition Language)

Sistema Fiep

Analisando um domínio (Pessoa) e identificando seus atributos definimos como serão armazenados os dados de uma Pessoa, os dado são: Nome, Sexo, Data de nascimento, Endereço e Comidas favoritas.

As colunas Nome, Endereco e ComidaFavorita são do tipo **varchar** e permitem a entrada de dados de forma livre. A coluna Sexo permite um único **char**, que deve ser igual a M ou F. A coluna DataNasc é do tipo **date**, já que um componente de horário não é necessário.



Nome	Sexo	DataNasc	Endereco	ComidaFavorita

DDL (Data Definition Language)

Ao olhar para as colunas da tabela **Pessoa** mais uma vez, as seguintes questões surgem:

- A coluna **Nome** é, na verdade, um atributo composto consistindo em um **primeiro** e ultimo **nome**.
- Como várias pessoas podem ter o mesmo nome, sexo, data de nascimento e assim por diante, não há colunas na tabela **Pessoa** que garantam unicidade.
- A coluna **Endereco** também é um atributo composto, consistindo em rua, cidade, estado e país.
- A coluna **ComidaFavorita** é uma lista, contendo nenhum (0), um (1) ou mais itens independentes. É melhor criar uma tabela separada para que esse dado inclua uma chave estrangeira da tabela **Pessoa**, para podermos saber a qual pessoa uma comida determinada deve ser atribuída.

Depois considerar todas essas questões, a tabela **Pessoa** deve ser normalizada.

DDL (Data Definition Language)

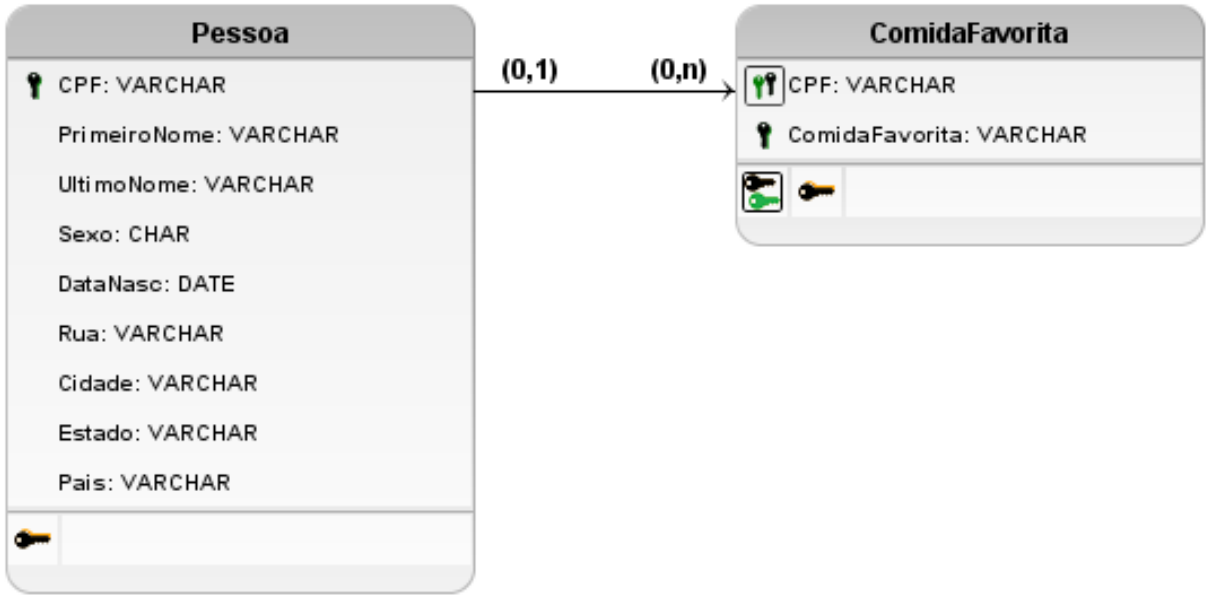
Nome	Sexo	DataNasc	Endereco	ComiFavorita

CPF	PrimeiroNome	UletimoNome	Sexo	DataNasc	Rua	Cidade	Estado	Pais	ComidaFavorita

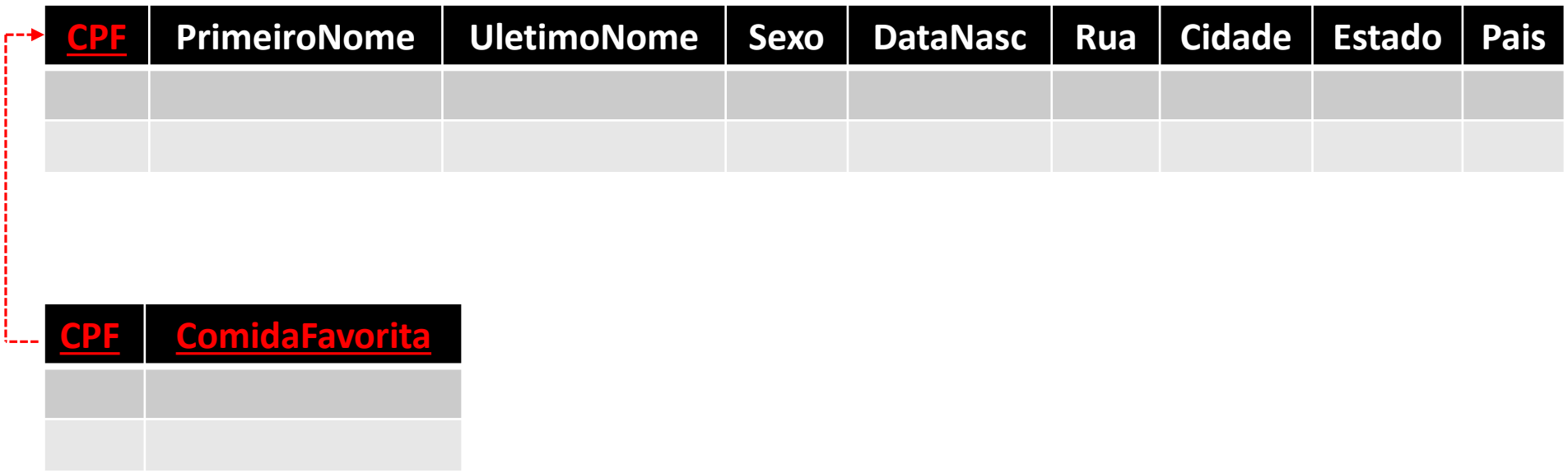
DDL (Data Definition Language)

Agora que a tabela **Pessoa** tem uma chave primária **CPF** para garantir a unicidade, o próximo passo é construir uma tabela **ComidaFavorita** que inclua uma chave estrangeira referenciando a tabela **Pessoa**.

As colunas **CPF** e **ComidaFavorita** formam a chave primária da tabela **ComidaFavorita**, e a coluna **CPF** também é uma chave estrangeira para a tabela **Pessoa**.



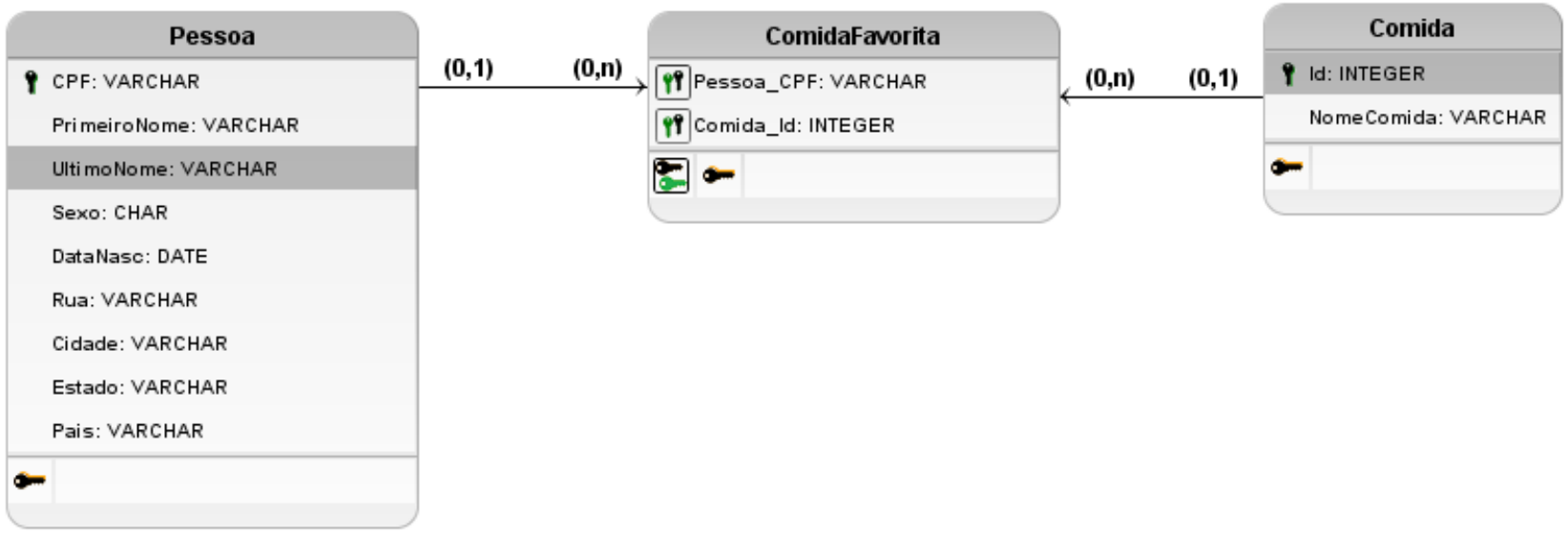
DDL (Data Definition Language)



DDL (Data Definition Language)

Retirar a coluna **ComidaFavorita** da tabela **Pessoa** foi correto. Entretanto, podemos ajustar ainda mais esse modelo, por exemplo, se uma pessoa cadastra “massa” como uma comida favorita enquanto outra lista “espaguete”, ambos são massas. Para evitar esse problema, podemos definir que as pessoas escolham suas comidas favoritas de uma lista de opções.

Nesse caso, é necessário criar uma tabela **Comida** com as colunas **Id** e **NomeComida** e, então, alterar a tabela **ComidaFavorita** para conter uma chave estrangeira referenciando da tabela **Comida**.



DDL (Data Definition Language)

Pessoa

CPF	PrimeiroNome	UletimoNome	Sexo	DataNasc	Rua	Cidade	Estado	Pais

ComidaFavorita

Pessoa_CPF	Comida_Id

Comida

Id	NomeComida

DDL (Data Definition Language)

- Criação de Tabelas:

```
CREATE TABLE NOME_TABELA(
Col1 TIPO_COLUNA [Not Null],
Col2 TIPO_COLUNA [Not Null],
Col3 TIPO_COLUNA [Not Null])
```

- Na criação de tabelas é possível especificar vários tipos de restrições:
  - Chave Primária: PRIMARY KEY;
  - Chave Estrangeira: FOREIGN KEY;
  - Restrição de Unicidade: UNIQUE;
  - Restrição de Domínio: CHECK.

DDL (Data Definition Language)

- Alteração de Tabelas:
  - Incluir novas colunas em uma tabela;
  - Excluir colunas existentes em uma tabela;
  - Adicionar a definição de uma restrição em uma tabela;
  - Excluir a definição de uma restrição existente em uma tabela;
  - Modificar uma coluna.

```
ALTER TABLE Pessoa
ADD COLUMN nova_coluna VARCHAR(10) NOT NULL;

ALTER TABLE Comida
ADD COLUMN nova_couma VARCHAR(50) NOT NULL;
```

DDL (Data Definition Language)

```
USE TESTE

GO
--Criar de tabelas no SQL Server
--Criar tabela Comida
CREATE TABLE Comida (
  Id INTEGER IDENTITY,
  NomeComida VARCHAR(50) NULL,
  PRIMARY KEY(Id)
);

--Criar tabela associativa ComidaFavorita
CREATE TABLE ComidaFavorita (
  Pessoa_CPF VARCHAR(11) NOT NULL,
  Comida_Id INTEGER NOT NULL,
  PRIMARY KEY(Pessoa_CPF, Comida_Id),
);
```

DDL (Data Definition Language)

Sistema Fiep

```
--Criar tabela Pessoa
CREATE TABLE Pessoa (
  CPF VARCHAR(11) NOT NULL,
  PrimeiroNome VARCHAR(20) NULL,
  UltimoNome VARCHAR(20) NULL,
  Sexo CHAR(1) NULL,
  DataNasc DATE NULL,
  Rua VARCHAR(50) NULL,
  Cidade VARCHAR(50) NULL,
  Estado CHAR(2) NULL,
  Pais VARCHAR(30) NULL,
  PRIMARY KEY(CPF)
);
```

DDL (Data Definition Language)


```
--Criar constraints

ALTER TABLE ComidaFavorita
ADD CONSTRAINT FK_Pessoa_CPF FOREIGN KEY
(Pessoa_CPF)
REFERENCES Pessoa (CPF);

ALTER TABLE ComidaFavorita
ADD CONSTRAINT FK_ComidaId FOREIGN KEY
(Comida_Id)
REFERENCES Comida (Id);
```

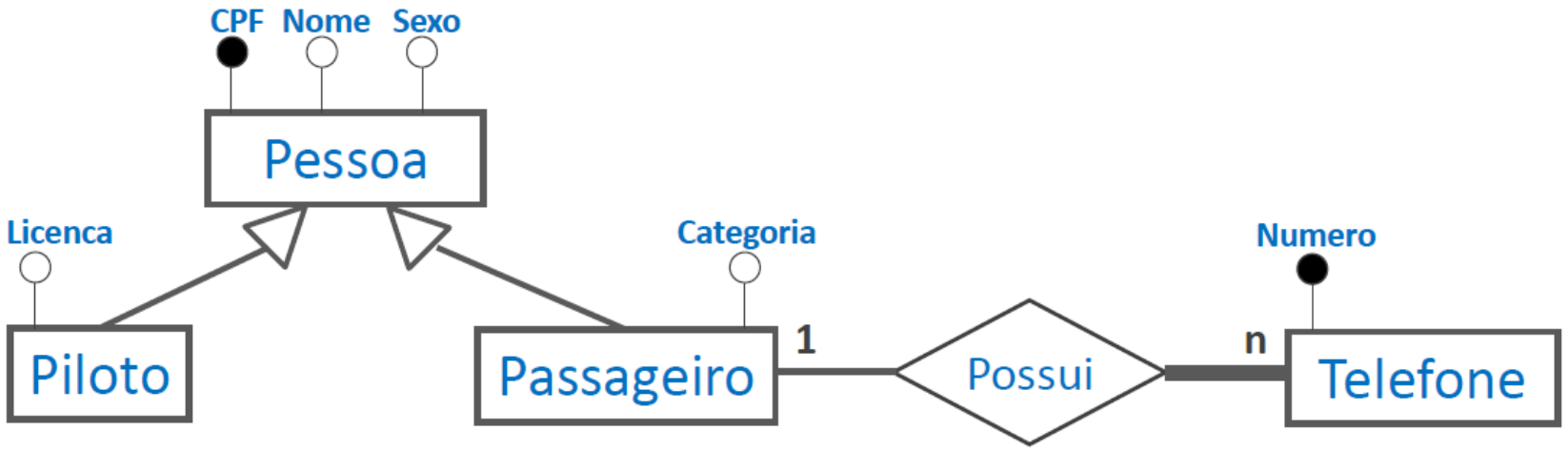
Descrição de um domínio

Será desenvolvido um sistema de informação para uma companhia aérea com as seguintes regras de negócio:

- Toda pessoa cadastrada no sistema deve possuir CPF e nome, e quando possível deve-se cadastrar o sexo também, cujos valores só podem ser ‘M’, para masculino, e ‘F’, para feminino.
- Os pilotos devem cadastrar sua licença de voo, e os passageiros, a sua categoria.
- Deve-se cadastrar também os telefones dos passageiros.
- Deve-se cadastrar os dados de voo, número e data, e o respectivo piloto.
- Deve-se identificar também os passageiros dos voos.

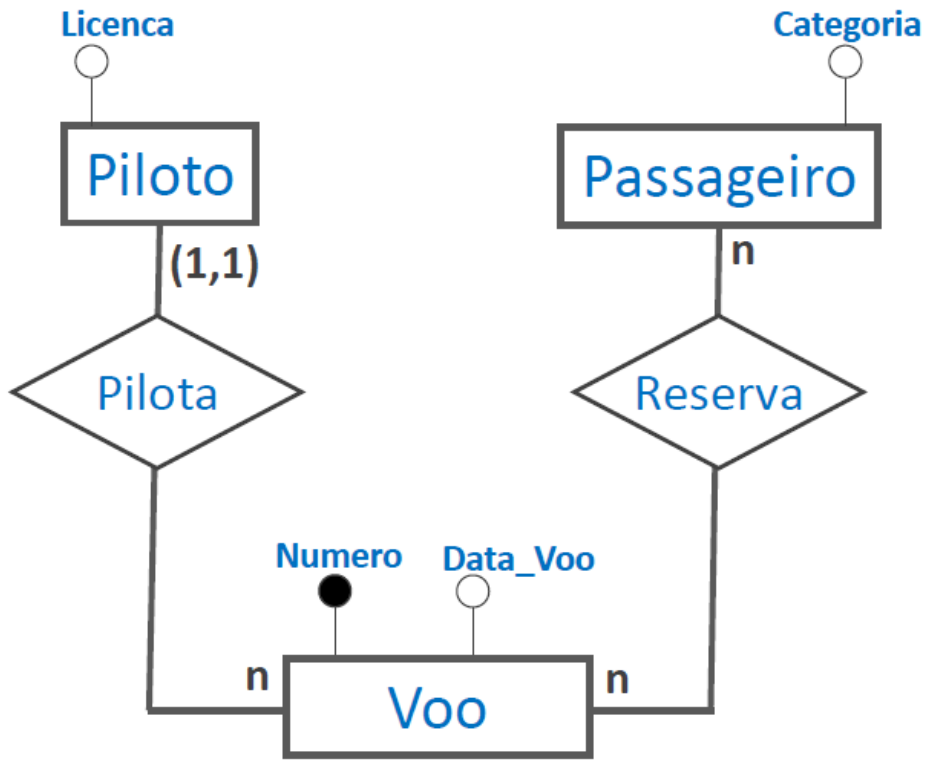
Descrição de um domínio

- Toda pessoa cadastrada no sistema deve possuir CPF e nome, e quando possível deve-se cadastrar o sexo também, cujos valores só podem ser ‘M’, para masculino, e ‘F’, para feminino.
- Os pilotos devem cadastrar sua licença de voo, e os passageiros, a sua categoria.
- Deve-se cadastrar também os telefones dos passageiros.

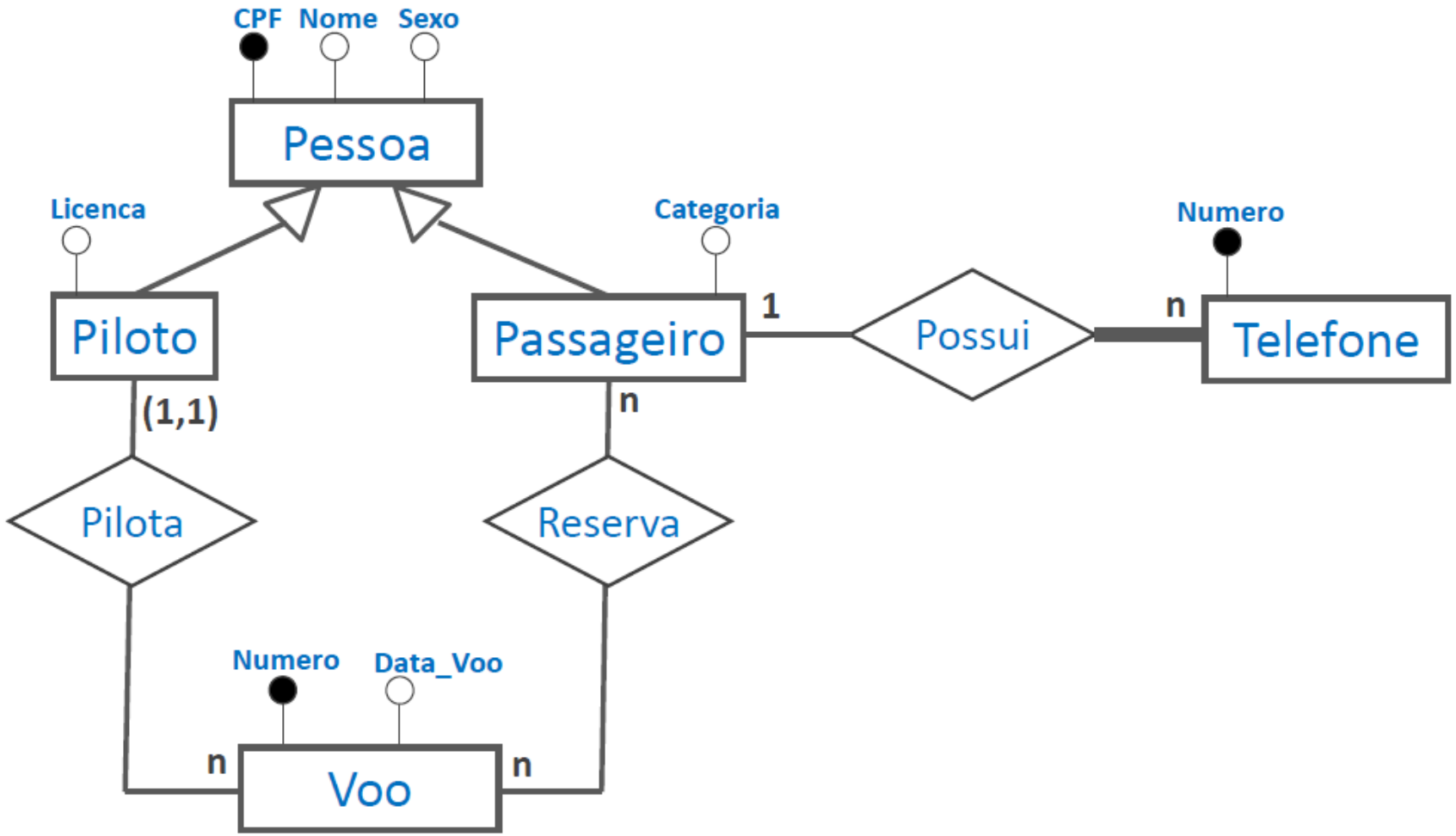


Descrição de um domínio

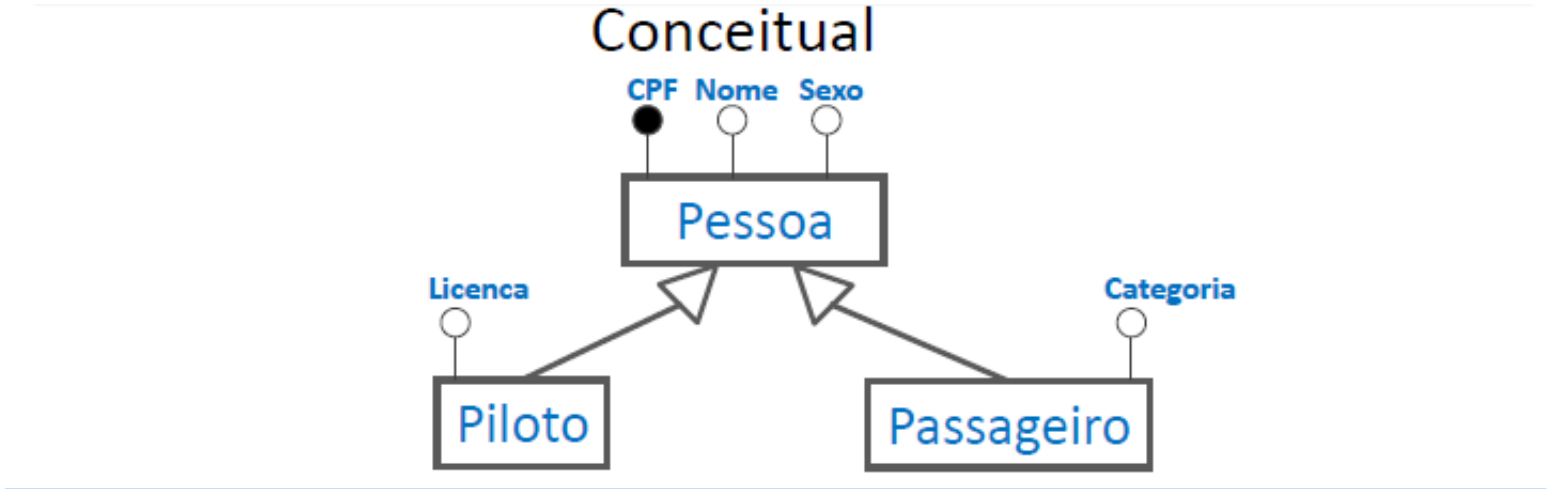
- Deve-se cadastrar os dados de voo, número e data, e o respectivo piloto.
- Deve-se identificar também passageiros dos voos.



Modelo Conceitual



Modelo Conceitual – Modelo Lógico



Lógico

Piloto

P_CPF	Licenca
222	LC01
555	LC02

Pessoa

CPF	Nome	Sexo
111	Ana	F
222	Beto	M
333	Caio	M
444	Igor	M
555	Edu	M

Passageiro

P_CPF	Categoria
333	Ouro
444	Normal

Modelo Conceitual – Especificação do BD

- Pessoa (**CPF**, Nome, Sexo).

- Piloto (**P_CPF**, Licenca)
  P_CPF Referencia **Pessoa(CPF**).

- Passageiro(**P_CPF**, Categoria)
  P_CPF Referencia **Pessoa(CPF**).

Script DDL

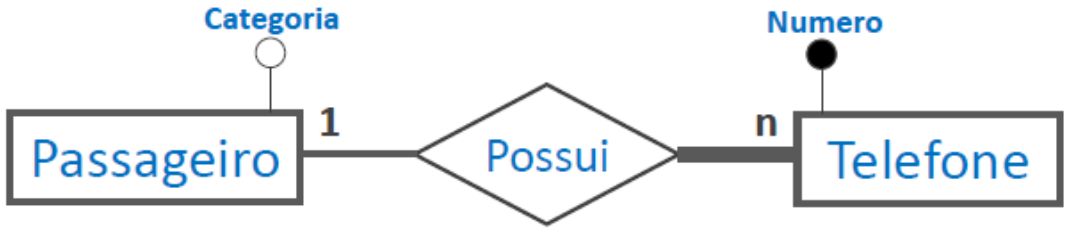
```
CREATE TABLE Pessoa(CPF VARCHAR(11) NOT NULL,
Nome VARCHAR(30) NOT NULL,
Sexo CHAR(1) CHECK (Sexo IN('M', 'F')),
PRIMARY KEY (CPF));

CREATE TABLE Piloto(P_CPF VARCHAR(11) NOT NULL,
Licenca VARCHAR(5) NOT NULL UNIQUE,
PRIMARY KEY (P_CPF),
FOREIGN KEY (P_CPF)
REFERENCES Pessoa (CPF));

CREATE TABLE Passageiro(P_CPF VARCHAR(11) NOT NULL,
Categoria VARCHAR(10) ,
PRIMARY KEY (P_CPF),
FOREIGN KEY (P_CPF)
REFERENCES Pessoa (CPF));
```

Modelo Conceitual – Modelo Lógico

Conceitual



Lógico

Passageiro

P_CPF	Categoria
333	Ouro
444	Normal

Telefone

Numero	Pas_P_CPF
999555	333
222444	444
888555	333

Modelo Conceitual – Especificação do BD

- Passageiro(**P_CPF**, Categoria)
  P_CPF Referencia Pessoa(**CPF**).

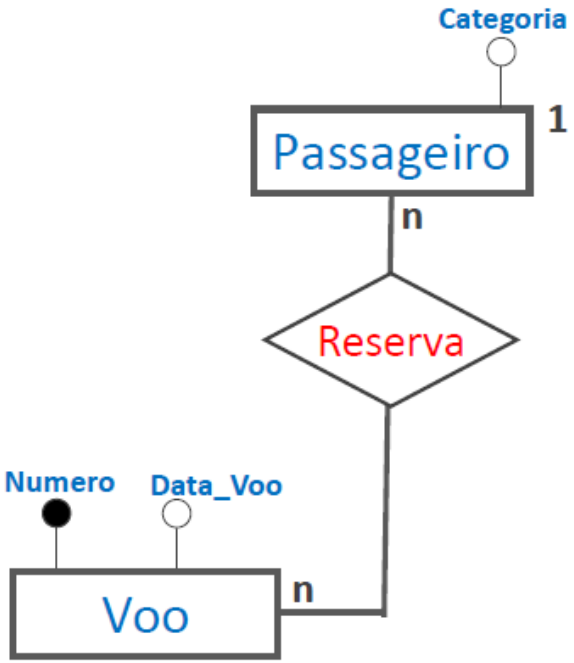
- Telefone(**Numero**, **Pas_P_CPF**)
  Pas_P_CPF Referencia Passageiro(**P_CPF**).

Script DDL

```
CREATE TABLE Telefone(Numero VARCHAR(10) NOT NULL,
Pas_P_CPF VARCHAR(11) NOT NULL,
PRIMARY KEY (Numero, Pas_P_CPF),
FOREIGN KEY (Pas_P_CPF)
REFERENCES Passageiro(P_CPF));
```

Modelo Conceitual – Modelo Lógico

Conceitual



Lógico

Passageiro

P_CPF	Categoria
333	Ouro
444	Normal

Reserva

Pas_P_CPF	V_Numero
333	101
444	101
333	202
333	404

Voo

Numero	Data_Voo
G101	10/10/2010
G202	11/11/2011
G303	12/12/2012
G404	13/01/2013

Modelo Conceitual – Especificação do BD

- Passageiro(**P_CPF**, Categoria)
  P_CPF Referencia Pessoa(**CPF**).

- Voo(**Numero**, Data_Voo).

- Reserva(**Pas_P_CPF**, **V_Numero**)
  Pas_P_CPF Referencia Passageiro(**P_CPF**)
  V_Numero Referencia Voo(**Numero**).

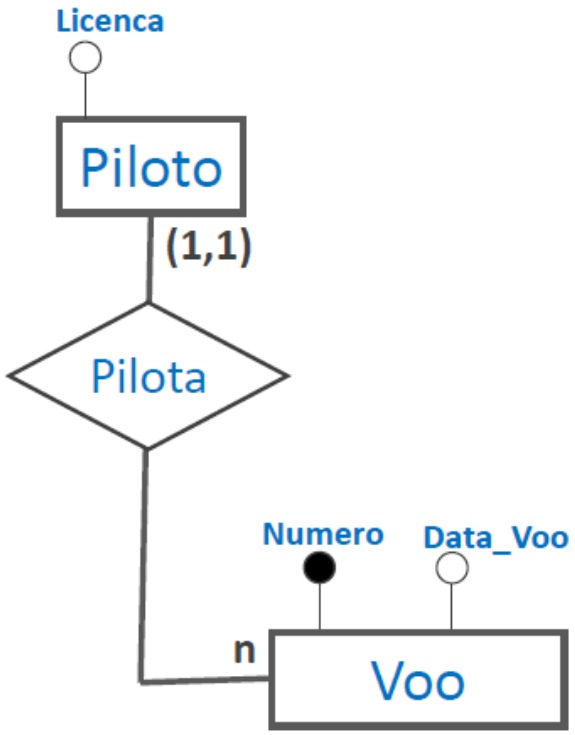
Script DDL

```
CREATE TABLE Voo(Numero Integer NOT NULL,
Data_Voo Date NOT NULL,
PRIMARY KEY (Numero));

CREATE TABLE Reserva(PAS_P_CPF VARCHAR(11) NOT NULL,
V_Numero Integer NOT NULL,
PRIMARY KEY (Pas_P_CPF, V_Numero),
FOREIGN KEY (Pas_P_CPF)
REFERENCES Passageiro(P_CPF),
FOREIGN KEY (V_Numero)
REFERENCES Voo(Numero));
```

Modelo Conceitual – Modelo Lógico

Conceitual



Lógico

Piloto

P_CPF	Licenca
222	LC01
555	LC02

Voo

Numero	Data_Voo	Pi_P_CPF
101	10/10/2010	222
202	11/11/2011	555
303	12/12/2012	555
404	13/01/2013	555

Modelo Conceitual – Especificação do BB

- Piloto(**P_CPF**, Licenca)
  P_CPF Referencia Pessoa(**CPF**).

- Voo(**Numero**, Data_Voo, **Pi_P_CPF**)
  Pi_P_CPF Referencia Piloto(**P_CPF**).

Script DDL

```
ALTER TABLE Voo
ADD Pi_P_CPF VARCHAR(11) NOT NULL;

ALTER TABLE Voo
ADD CONSTRAINT FK_Piloto_Voo FOREIGN KEY (Pi_P_CPF)
REFERENCES Piloto (P_CPF);
```

Script DML Inserção


```
--Insert Pessoa
Insert Into Pessoa(CPF, Nome, Sexo)
Values('111', 'Ana', 'F' );
Insert Into Pessoa(CPF, Nome, Sexo)
Values('222', 'Beto', 'M' );
Insert Into Pessoa(CPF, Nome, Sexo)
Values('333', 'Caio', 'M' );
Insert Into Pessoa(CPF, Nome, Sexo)
Values('444', 'Igor', 'M' );
Insert Into Pessoa(CPF, Nome, Sexo)
Values('555', 'Edu', 'M' );
```

Script DML Inserção


```
--Insert Piloto
Insert Into Piloto(P_CPF, Licenca)
Values('222', 'LC01' );
Insert Into Piloto(P_CPF, Licenca)
Values('555', 'LC02' );
```

Script DML Inserção


```
--Insert Passageiro
Insert Into Passageiro(P_CPF, Categoria)
Values('333', 'Ouro' );
Insert Into Passageiro(P_CPF, Categoria)
Values('444', 'Normal' );
```

Script DML Inserção

```
-- Insert Telefone
Insert Into Telefone(Numero, Pas_P_CPF)
Values(999555, '333' );
Insert Into Telefone(Numero, Pas_P_CPF)
Values(222444, '444' );
Insert Into Telefone(Numero, Pas_P_CPF)
Values(888555, '333' );
```

Script DML Inserção

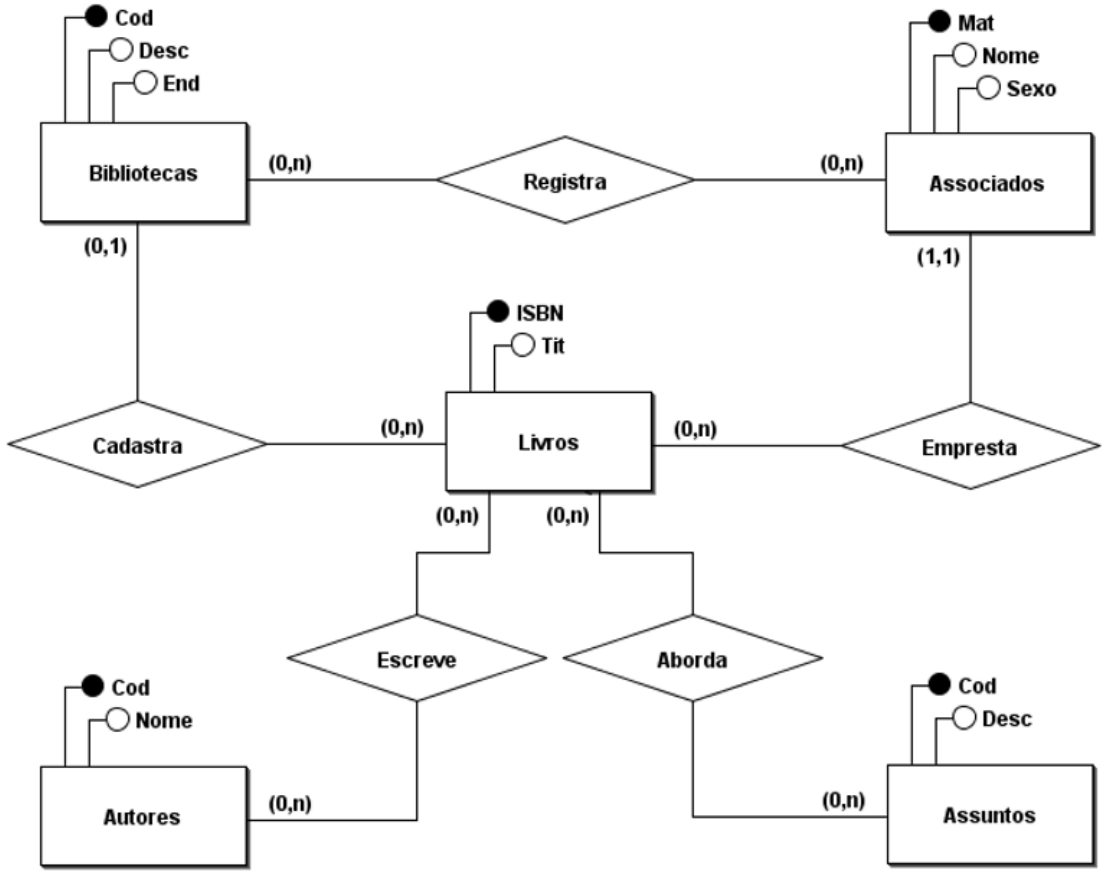

```
--Insert Voo
Insert Into Voo(Numero, Data_Voo, Pi_P_CPF)
Values(101,'2010-10-10','222');
Insert Into Voo(Numero, Data_Voo, Pi_P_CPF)
Values(202,'2011-11-11','555');
Insert Into Voo(Numero, Data_Voo, Pi_P_CPF)
Values(303,'2012-12-12','555');
Insert Into Voo(Numero, Data_Voo, Pi_P_CPF)
Values(404,'2013-01-13','555');
```

Script DML Inserção

```
--Insert Reserva
Insert Into Reserva(Pas_P_CPF, V_Numero)
Values('333', 101);
Insert Into Reserva(Pas_P_CPF, V_Numero)
Values('444', 101);
Insert Into Reserva(Pas_P_CPF, V_Numero)
Values('333', 202);
Insert Into Reserva(Pas_P_CPF, V_Numero)
Values('333', 404);
```

Atividade

Descrever a especificação formal do modelo conceitual abaixo e escrever os códigos DDLs para criação do modelo físico.



Sistema Fiep

FIEP
SESI
SENAI
IEL

nosso i é de indústria.