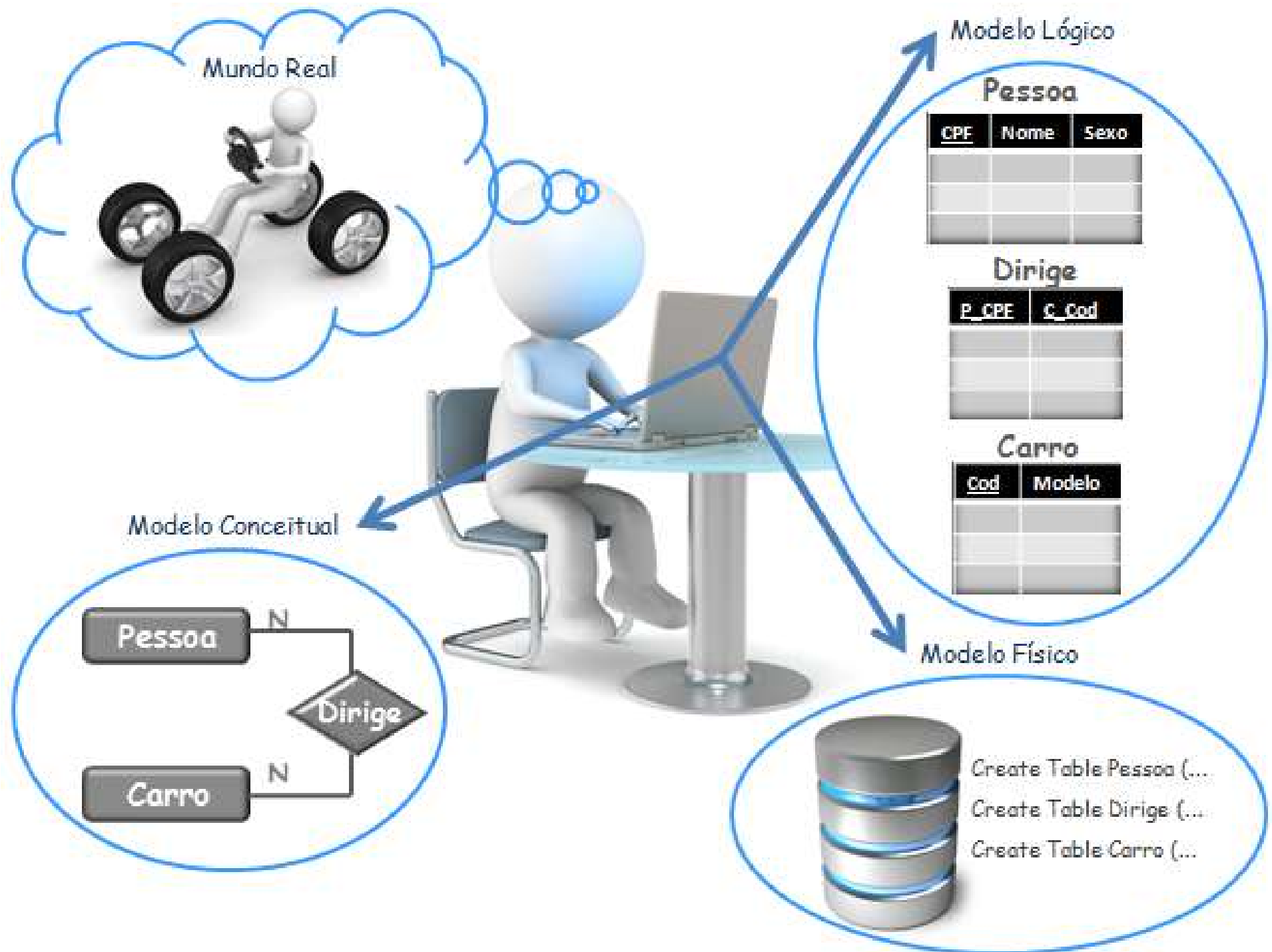


Modelagem de Dados e Linguagem
SQL Descomplicadas usando o MySQL



<https://www.youtube.com/c/ProgramarIsCool>



Agenda

- ❑ Structured Query Language - SQL
- ❑ Linguagem de Definição de Dados - DDL
 - ❑ Criação de Tabelas; e
 - ❑ Alteração de Tabelas.
- ❑ Script de Criação das Tabelas

Modelo Físico

Modelo Físico

- ❑ Consiste de uma representação gráfica do detalhamento físico das estruturas dos dados que o banco de dados irá armazenar. Este modelo é dependente do SGBD em que será implementado.

Structured Query Language - SQL

- ❑ Durante o desenvolvimento do sistema R, pesquisadores da IBM desenvolveram a linguagem SEQUEL, primeira linguagem de acesso para Sistemas Gerenciadores de Banco de Dados Relacionais.
- ❑ ISO e ANSI lançou em 1986 a primeira versão do padrão da linguagem SQL, o SQL-86.
- ❑ A linguagem passou por aperfeiçoamentos 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.
- ❑ Após uma grande revisão do padrão SQL3 foi lançada a SQL:2003. Nesta versão foi adicionada uma nova parte ligada ao tratamento de XML.

Linguagem de Definição de Dados - DDL

❑ 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.

Linguagem de Definição de Dados - DDL

❑ 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.

Linguagem de Definição de Dados - DDL

❑ Alteração de Tabelas:

❑ ALTER TABLE Telefone

```
ADD COLUMN nome_coluna VARCHAR2(5) DEFAULT  
'FIXO' NOT NULL;
```

❑ ALTER TABLE Telefone

```
DROP COLUMN nome_coluna;
```

Script de Criação de Tabelas

Estudo de Caso

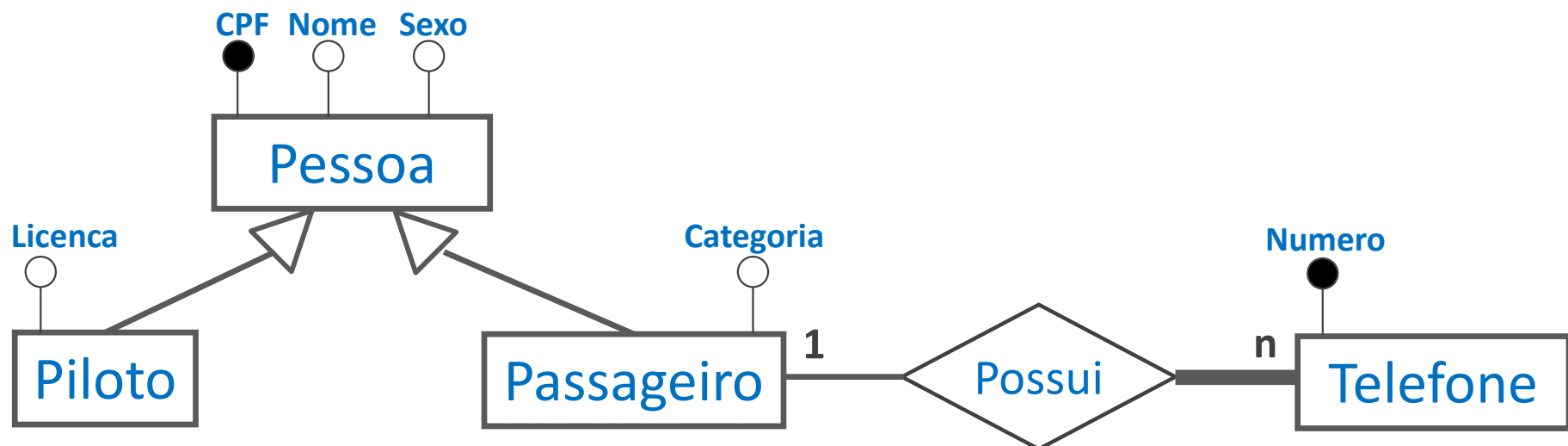
Descrição do Domínio

Deseja-se desenvolver um sistema de informação computadorizado 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 passageiros dos voos.

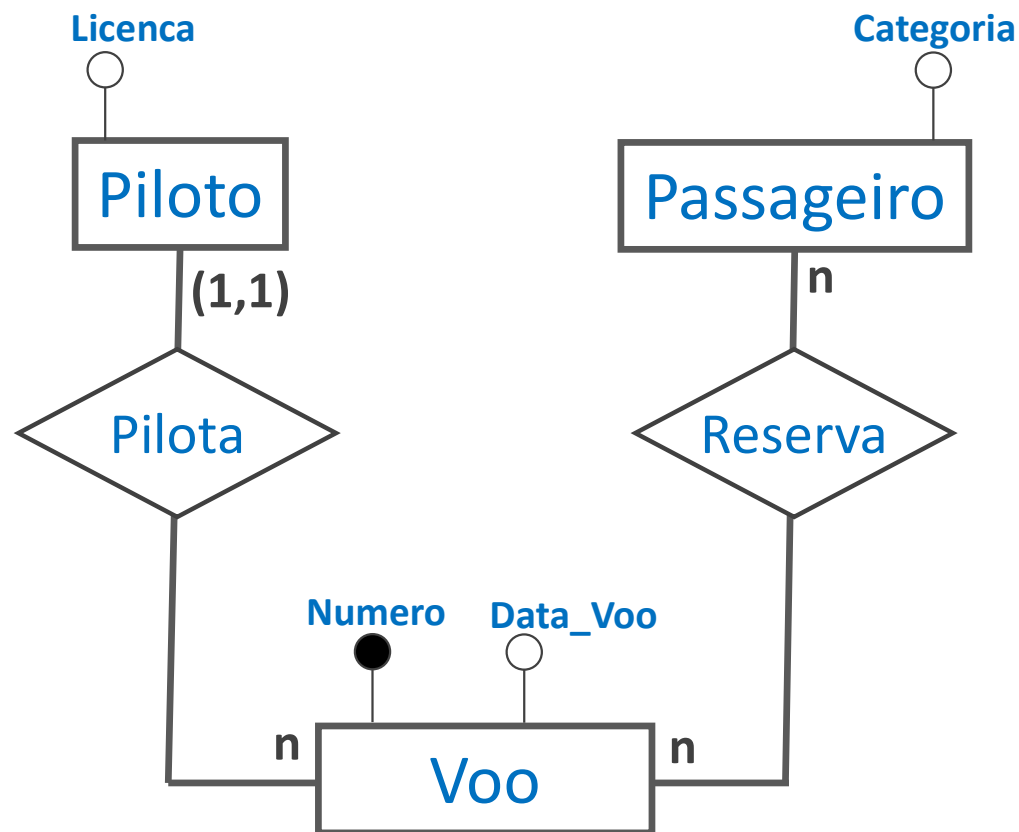
Descrição do Domínio

- ❑ 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.

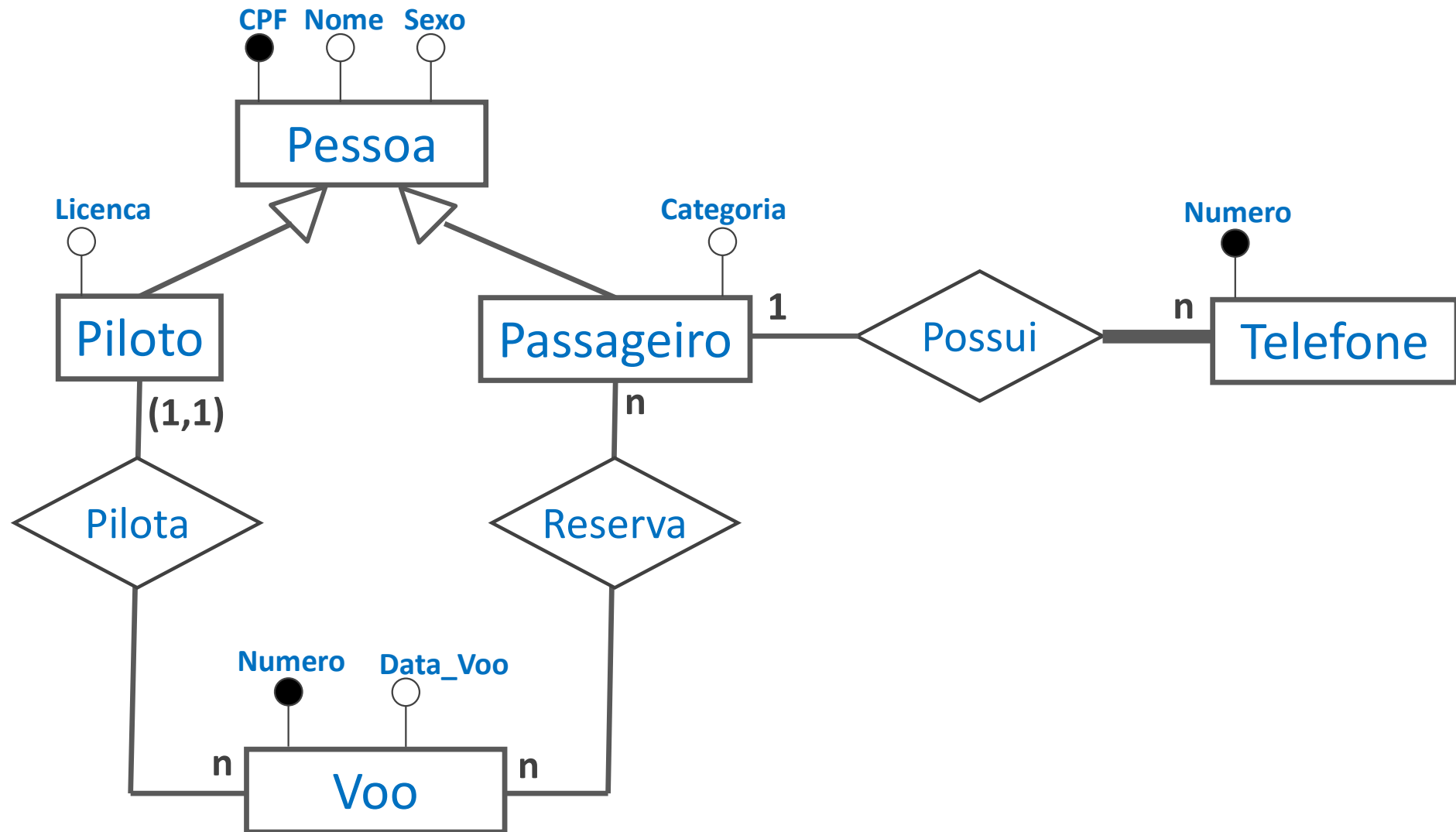


Descrição do Domínio

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

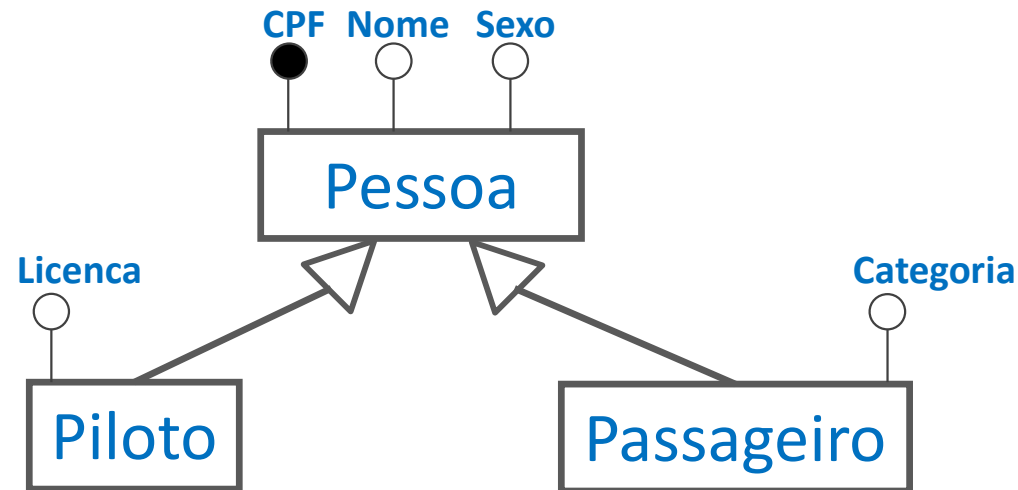


Modelo Conceitual

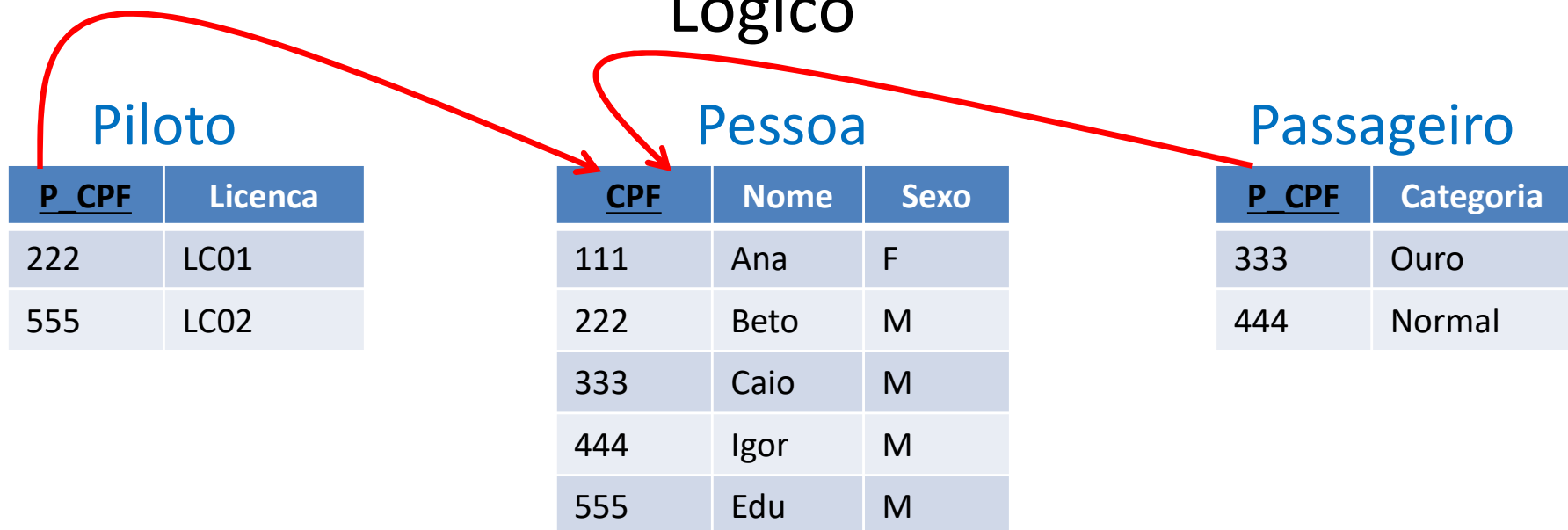


Modelo Conceitual – Modelo Lógico

Conceitual



Lógico



Modelo Lógico – Especificação do BD

❑ Pessoa (CPF, Nome, Sexo).

❑ Piloto (P_CPF, Licenca)

P_CPF Referencia Pessoa(CPF).

❑ Passageiro(P_CPF, Categoria)

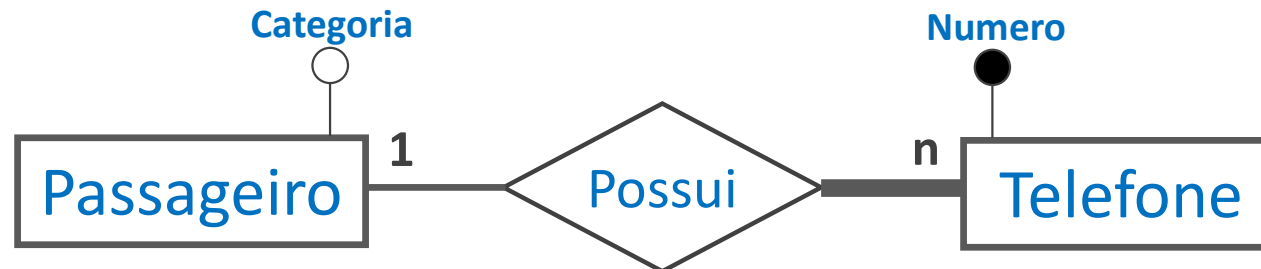
P_CPF Referencia Pessoa(CPF).

DDL

- ❑ CREATE TABLE Pessoa(CPF VARCHAR(11) NOT NULL,
 Nome VARCHAR(30) NOT NULL UNIQUE,
 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		Telefone	
<u>P_CPF</u>	Categoria	<u>Numero</u>	Pas P_CPF
333	Ouro	999555	333
444	Normal	222444	444
		888555	333

Modelo Lógico – Especificação do BD

❑ Passageiro(P_CPF, Categoria)

P_CPF Referencia Pessoa(CPF).

❑ Telefone(Numero, Pas_P_CPF)

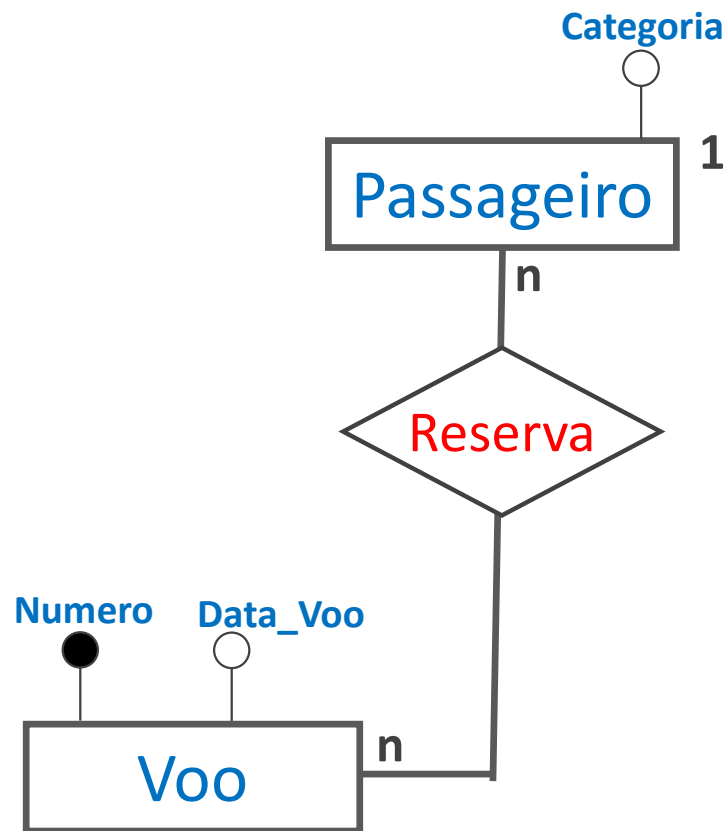
Pas_P_CPF Referencia Passageiro(P_CPF).

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

<u>P_CPF</u>	Categoria
333	Ouro
444	Normal

Reserva

<u>Pas_P_CPF</u>	<u>V_Numero</u>
333	101
444	101
333	202
333	404

Voo

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

Modelo Lógico – 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)

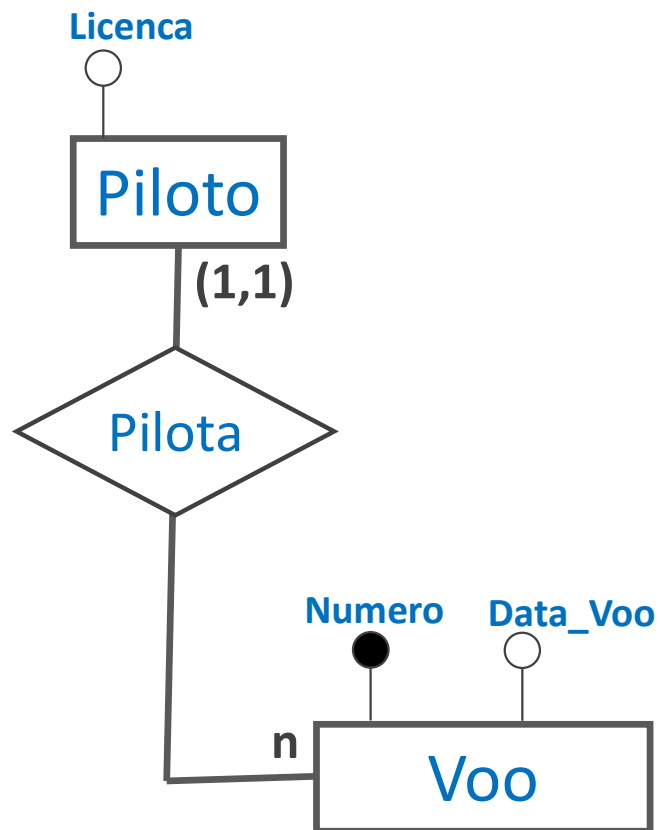
V_Numero Referencia Voo(Numero).

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



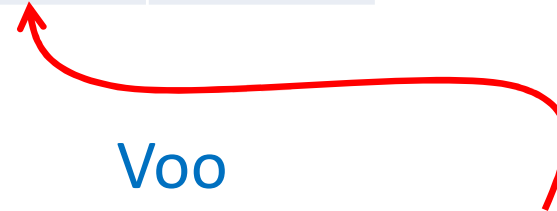
Lógico

Piloto

<u>P_CPF</u>	Licenca
222	LC01
555	LC02

Voo

<u>Numero</u>	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 Lógico – Especificação do BD

- ❑ Piloto(P_CPF, Licenca)

P_CPF Referencia Pessoa(CPF).

- ❑ Voo(Numero, Data_Voo, Pi_P_CPF)

Pi_P_CPF Referencia Piloto(P_CPF).

DDL

- ❑ ALTER TABLE Voo

ADD COLUMN Pi_P_CPF VARCHAR(11) NOT NULL;

- ❑ ALTER TABLE Voo

ADD CONSTRAINT FK_Piloto_Voo FOREIGN KEY (Pi_P_CPF)
REFERENCES Piloto (P_CPF);

Inserção de dados

❑ Tabela Pessoa:

```
Insert Into Pessoa(CPF, Nome, Sexo)
```

```
Values('111', 'Ana', 'F' );
```

```
Insert Into Pessoa(CPF, Nome, Sexo)
```

```
Values('222', 'Beto', 'F' );
```

```
Insert Into Pessoa(CPF, Nome, Sexo)
```

```
Values('333', 'Caio', 'F' );
```

```
Insert Into Pessoa(CPF, Nome, Sexo)
```

```
Values('444', 'Igor', 'F' );
```

```
Insert Into Pessoa(CPF, Nome, Sexo)
```

```
Values('555', 'Edu', 'F' );
```

Inserção de dados

❑ Tabela Piloto:

```
Insert Into Piloto(P_CPF, Licenca)
```

```
Values('222', 'LC01' );
```

```
Insert Into Piloto(P_CPF, Licenca)
```

```
Values('555', 'LC02' );
```

❑ Tabela Passageiro:

```
Insert Into Passageiro(P_CPF, Categoria)
```

```
Values('333', 'Ouro' );
```

```
Insert Into Passageiro(P_CPF, Categoria)
```

```
Values('444', 'Normal' );
```

Inserção de dados

❑ Tabela 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' );
```

Inserção de dados

❑ Tabela 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' );;
```

Inserção de dados

❑ Tabela 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 );
```

Fim