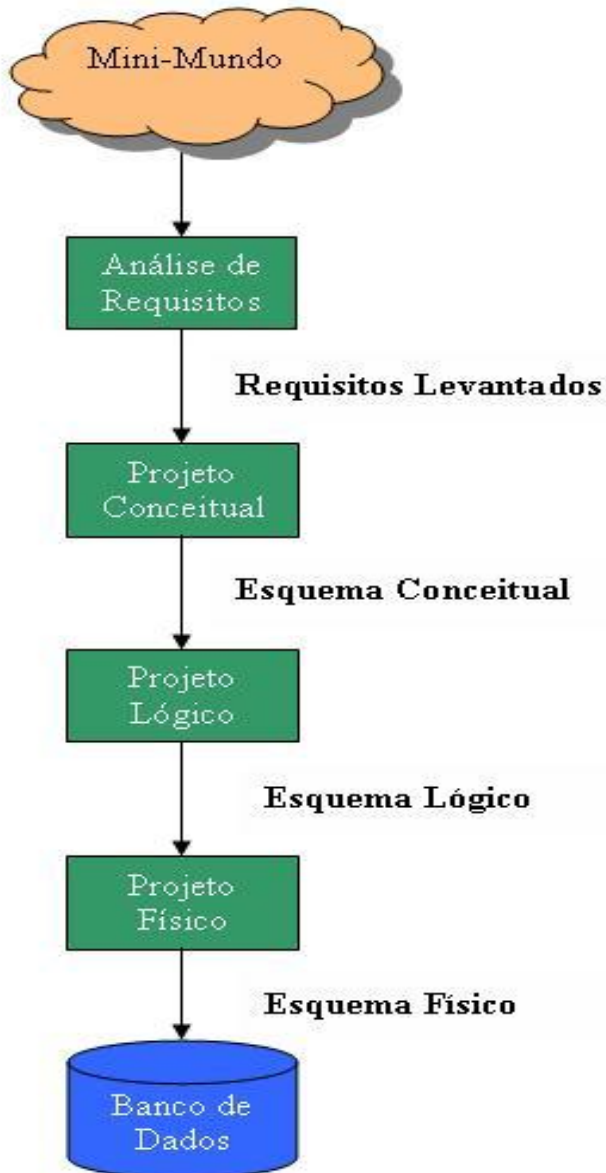


**[Projeto de Banco de Dados]**  
**Aula 02: Revisão: Modelagem Lógica**

# Etapas da Modelagem de BD

---



<- o problema real

<- o que o usuário quer

<- 1ª etapa modelagem

<- 2ª etapa modelagem

<- 3ª etapa modelagem

# Estudo de Caso: Projeto Conceitual

---

- A UFRPE deseja melhorar o sistema da biblioteca da UAST. Os alunos da disciplina de PBD foram convocados para esta função!
- Exercício:
  - Dados os requisitos seguintes, desenvolver o DER.
  - Utilizar o BrModelo.

# Estudo de Caso: desenvolver o DER

---

- Requisitos levantados:
  - 1) Devem ser cadastradas as obras do acervo da biblioteca, que representam os livros, periódicos (revistas, jornais) ou qualquer outro elemento do acervo. As obras possuem um código único, título, autor principal, ano de publicação e editora. As editoras, por sua vez, possuem um código, nome e cidade. *Uma obra sempre é de uma editora e uma editora pode possuir diversas obras.*
  - 2) *Cada obra contém um conjunto de exemplares e cada exemplar contém um código único e uma situação (disponível ou emprestada).*

# Estudo de Caso: desenvolver o DER

---

- Requisitos levantados:

3)O sistema deve guardar os dados sobre usuários da biblioteca. Cada usuário tem uma **identificação única, nome, endereço completo, telefone de contato e CPF.**

4)Os funcionários da biblioteca também devem ser cadastrados com **número de matrícula, seu nome completo e departamento** em que trabalha. Os departamentos, por sua vez, possuem **código e nome**. *Todo funcionário obrigatoriamente é vinculado a um departamento, que pode ter vários funcionários.*

# Estudo de Caso: desenvolver o DER

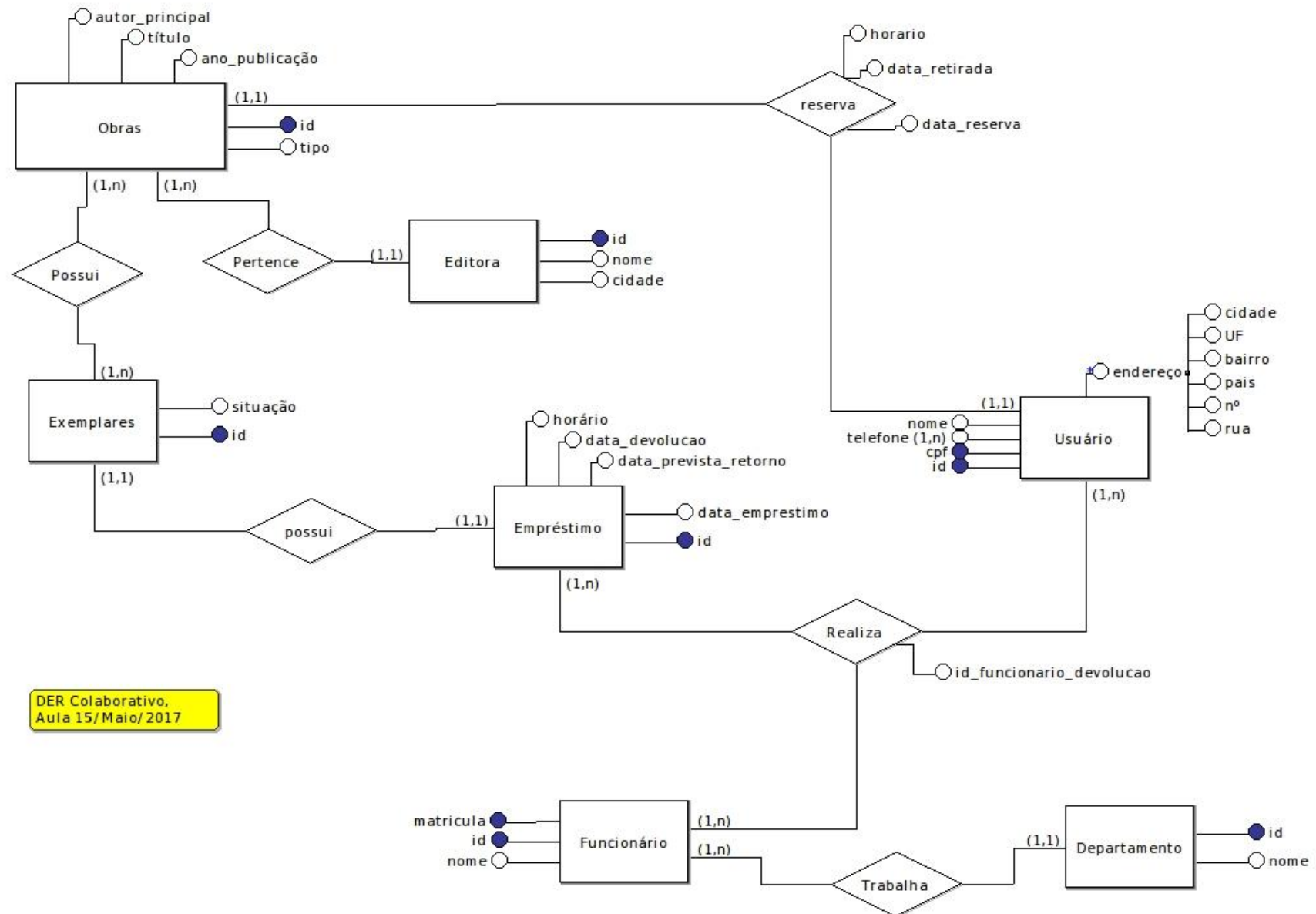
---

- Requisitos levantados:

5) Os usuários podem realizar empréstimos de exemplares. *Cada empréstimo contém um único exemplar obrigatoriamente.* Os empréstimos ainda devem registrar a **data e horário do empréstimo, data prevista de retorno, bem como o funcionário** que o realizou. Quando da **devolução** de um exemplar emprestado, deve-se registrar **a data e horário da devolução,** bem como o **funcionário** responsável pela devolução.

6) Os usuários ainda podem realizar reservas de obras. *Uma reserva deve conter uma única obra obrigatoriamente.* As reservas devem registrar **a data e horário da reserva e data na qual a obra será retirada.**

# Estudo de Caso: DER Resultante



# Modelagem Lógica



A modelagem lógica é feita observando 07 regras



# Modelagem Lógica

---

- Após o projeto conceitual do banco de dados, passamos para o projeto lógico.
- Objetivo: converter um modelo conceitual de entrada em projeto de implementação a ser praticado.
  - Por exemplo, o Modelo Relacional
- Perguntas:
  - Posso ignorar a etapa de modelagem conceitual, e implementar o projeto direto? OU
  - Não é mais “interessante” modelar logo no relacional?
    - Resposta: Não é **aconselhável**. É muito mais fácil para o cliente entender o MER, assim evitamos “ruídos” na comunicação, ou seja, problemas de entendimento.

# Modelagem Lógica

---

- **Regra 1:** Mapeamento de Entidades Fortes
  - Cada entidade forte deve ser transformada em uma relação
  - Todos os atributos simples da entidade devem ser incluídos na relação;
  - Apenas os componentes simples dos atributos compostos devem ser incluídos na relação;
  - Um dos atributos chaves da entidade deve ser escolhido como chave primária da relação.
- Sintaxe:

**NomeDaRelação(Atributo1, Atributo2, ..., AtributoN);**

# Modelagem Lógica

---

- **Regra 1:** Mapeamento de Entidades Fortes
  - Aplicando a Regra 1:
    - Obras (id, titulo, tipo, autor\_principal, ano\_publicacao)
    - Exemplares (id, situacao)
    - Editora (id, nome, cidade)
    - Emprestimo (id, data, horario, previsao\_retorno, devolucao)
    - Funcionario (id, matricula, nome)
    - Departamento (id, nome)
    - Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)

*(continua...)*

# Modelagem Lógica

---

- **Regra 2:** Mapeamento de Entidades Fracas
  - Cada entidade fraca deve ser transformada em uma relação, seguindo as mesmas restrições da Regra 1 para os seus atributos simples e compostos;
  - Incluir os atributos da chave primária da tabela dominante como chave estrangeira da relação;
  - A chave primária da relação deve ser a combinação dos atributos da chave primária da relação dominante e da chave da entidade fraca;

# Modelagem Lógica

---

- **Regra 2:** Mapeamento de Entidades Fracas
  - Aplicando a Regra 2:
    - não se aplica no Estudo de Caso.

# Modelagem Lógica

---

- **Regra 3:** Mapeamento de Relacionamentos Binários Um para Um (1:1)
  - Deve-se identificar as entidades que participam do relacionamento;
  - Existem três soluções possíveis:
    - Escolha da chave estrangeira;
    - Relacionamento incorporado;
    - Relação de relacionamento;

# Modelagem Lógica

---

- **Regra 3:** Mapeamento de Relacionamentos Binários Um para Um (1:1)
  - Deve-se identificar as entidades que participam do relacionamento;
  - Existem três soluções possíveis:
    - **Escolha da chave estrangeira;**
      - Deve-se escolher uma das relações e inserir nela a chave estrangeira da outra relação;
        - Geralmente, as entidades com participação total no relacionamento exercem este papel;
      - Incluir também todos os atributos do relacionamento como atributos da tabela;
    - Relacionamento incorporado;
    - Relação de relacionamento;

# Modelagem Lógica

---

- **Regra 3:** Mapeamento de Relacionamentos Binários Um para Um (1:1)
  - Aplicando a Regra 3:
    - Obras (id, titulo, tipo, autor\_principal, ano\_publicacao, horario, data\_reserva, data\_retirada, #id\_usuario)
    - Exemplares (id, situacao)
    - Editora (id, nome, cidade)
    - Emprestimo (id, data, horario, previsao\_retorno, devolucao, #id\_exemplar)
    - Funcionario (id, matricula, nome)
    - Departamento (id, nome)
    - Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)

*(continua...)*



# Modelagem Lógica

---

- **Regra 4:** Mapeamento de Relacionamentos Binários Um para Muitos (1:N)
  - Deve-se incluir a chave primária da relação que representa a entidade que aparece do lado “1” do relacionamento, como chave estrangeira na outra relação;
  - Isto acontece porque cada instância de uma entidade está relacionada a apenas uma instância da outra entidade;
  - Incluir também os atributos do relacionamento na relação que contém a chave estrangeira;

# Modelagem Lógica

---

- **Regra 4:** Mapeamento de Relacionamentos Binários Um para Muitos (1:N)
  - Aplicando a Regra 4.
    - Obras (id, titulo, tipo, autor\_principal, ano\_publicacao, horario, data\_reserva, data\_retirada, #id\_editora #id\_usuario)
    - Exemplares (id, situacao)
    - Editora (id, nome, cidade)
    - Emprestimo (id, data, horario, previsao\_retorno, devolucao, #id\_exemplar)
    - Funcionario (id, matricula, nome, #id\_departamento)
    - Departamento (id, nome)
    - Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)

*(continua...)*

# Modelagem Lógica

---

- **Regra 5:** Mapeamento de Relacionamentos Binários Muitos para Muitos (N:M)
  - Deve-se criar uma nova relação para o relacionamento;
  - Incluir as chaves primárias das duas entidades que participam do relacionamento na relação;
    - A combinação destas chaves formará a chave primária da relação;
  - Incluir também na relação os atributos do relacionamento;

# Modelagem Lógica

---

- **Regra 5:** Mapeamento de Relacionamentos Binários Muitos para Muitos (N:M)
    - Aplicando a Regra 5
      - Obras (id, titulo, tipo, autor\_principal, ano\_publicacao, horario, data\_reserva, data\_retirada, *#id\_editora* *#id\_usuario*)
      - Exemplares (id, situacao)
      - Editora (id, nome, cidade)
      - Emprestimo (id, data, horario, previsao\_retorno, devolucao, *#id\_exemplar*)
      - Funcionario (id, matricula, nome, *#id\_departamento*)
      - Departamento (id, nome)
      - Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)
      - Obras\_Exemplares (*#id\_obras*, *#id\_exemplares*)
- (continua...)

# Modelagem Lógica

---

- **Regra 6:** Mapeamento de Atributos Multivalorados
  - Deve-se criar uma nova relação para o atributo multivalorado;
  - Incluir na relação o atributo multivalorado;
  - Incluir a chave primária da relação que representa a entidade ao qual o atributo está associado como chave estrangeira;
  - A chave primária será a combinação da chave estrangeira e do atributo multivalorado;

# Modelagem Lógica

---

- **Regra 6:** Mapeamento de Atributos Multivalorados
    - Aplicando a Regra 6:
      - Obras (id, titulo, tipo, autor\_principal, ano\_publicacao, horario, data\_reserva, data\_retirada, *#id\_editora* *#id\_usuario*)
      - Exemplares (id, situacao)
      - Editora (id, nome, cidade)
      - Emprestimo (id, data, horario, previsao\_retorno, devolucao, *#id\_exemplar*)
      - Funcionario (id, matricula, nome, *#id\_departamento*)
      - Departamento (id, nome)
      - Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)
      - Obras\_Exemplares (*#id\_obras*, *#id\_exemplares*)
      - Telefone(*#id\_usuario*, telefone)
- (continua...)*

# Modelagem Lógica

---

- **Regra 7:** Mapeamento de Relacionamentos *N*-ários
  - Para relacionamentos *n*-ários ( $n > 2$ ), deve-se criar uma nova relação para representar o relacionamento;
  - As chaves primárias de cada relação que representa uma entidade participante do relacionamento devem ser inseridas na relação;
  - Os atributos do relacionamento também devem ser incluídos na relação;
  - A chave primária da relação será a combinação das chaves primárias das relações;

# Modelagem Lógica

---

- **Regra 7:** Mapeamento de Relacionamentos *N-ários*
  - Aplicando a Regra 7:
    - Obras (id, titulo, tipo, autor\_principal, ano\_publicacao, horario, data\_reserva, data\_retirada, #id\_editora #id\_usuario)
    - Exemplares (id, situacao)
    - Editora (id, nome, cidade)
    - Emprestimo (id, data, horario, previsao\_retorno, devolucao, #id\_exemplar)
    - Funcionario (id, matricula, nome, #id\_departamento)
    - Departamento (id, nome)
    - Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)
    - Obras\_Exemplares (#id\_obras, #id\_exemplares)
    - Telefone(#id\_usuario, telefone)
    - Realizar\_Emprestimo (#id\_emprestimo, #id\_usuario, #id\_funcionario, id\_funcionario\_devolucao)

(continua...)



# Modelagem Lógica

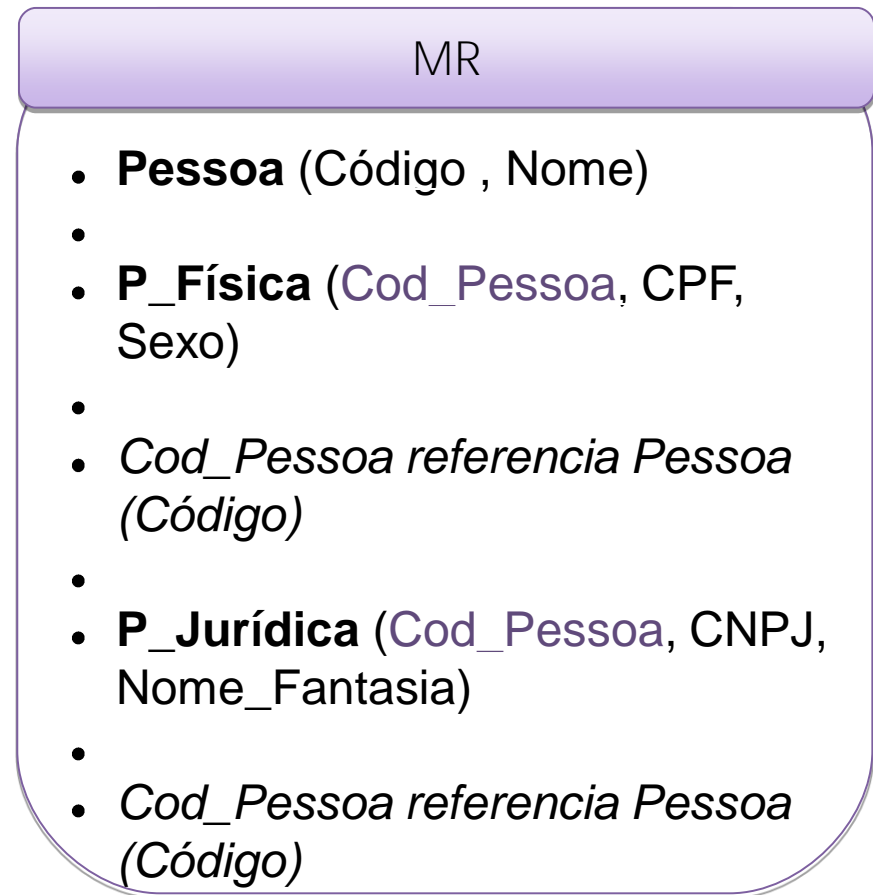
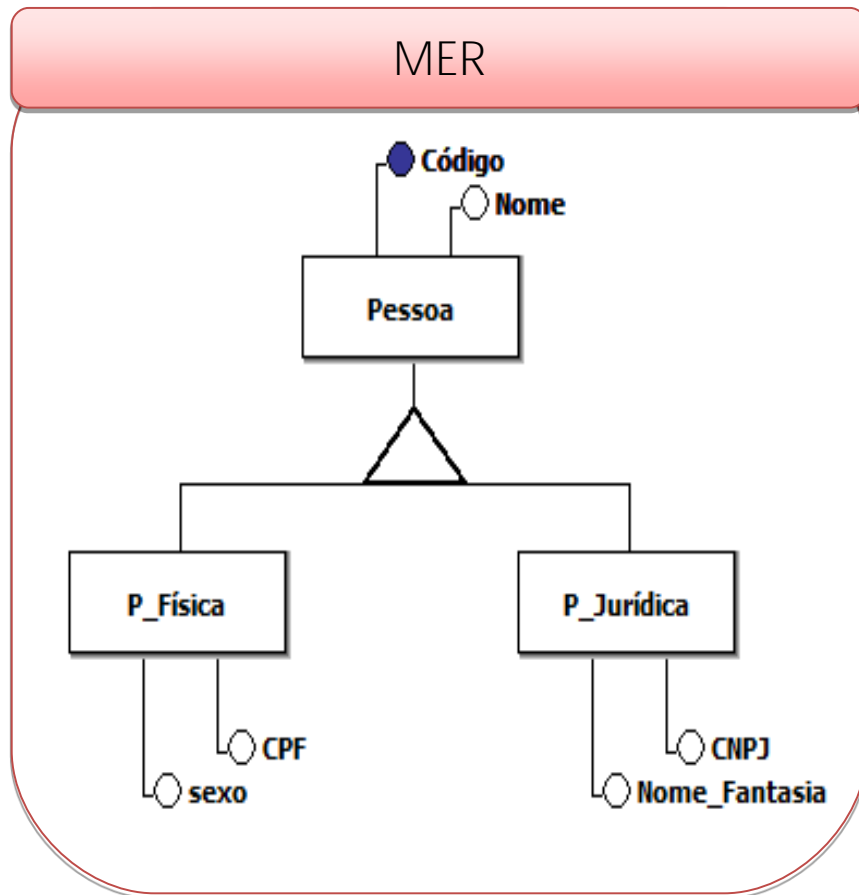
---

- Generalização/Especialização:

Solução	Ligação
1) Uma tabela para cada entidade	As tabelas especializadas são ligadas com a tabela genérica através de chave estrangeira (chave primária da tabela genérica)
2) Tabela única para toda hierarquia	É acrescentada uma coluna para determinar o tipo da entidade.
3) Tabela apenas para as entidades especializadas	Os atributos da tabela genérica são duplicados nas tabelas especializadas e não existe ligação entre as tabelas especializadas.

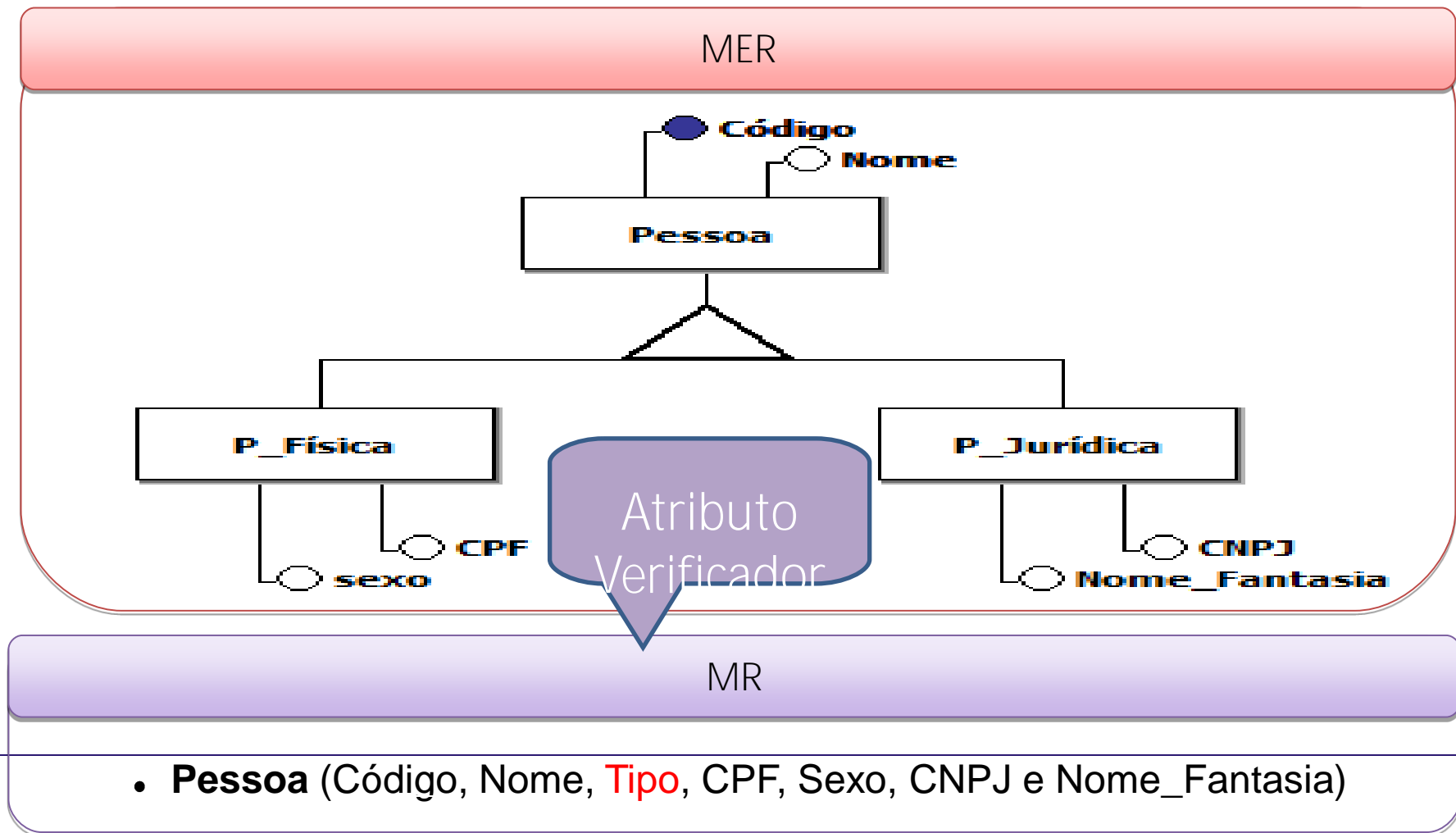
# Modelagem Lógica

- Generalização/Especialização:
  - Solução 1: Uma tabela para cada entidade.
    - Ex:



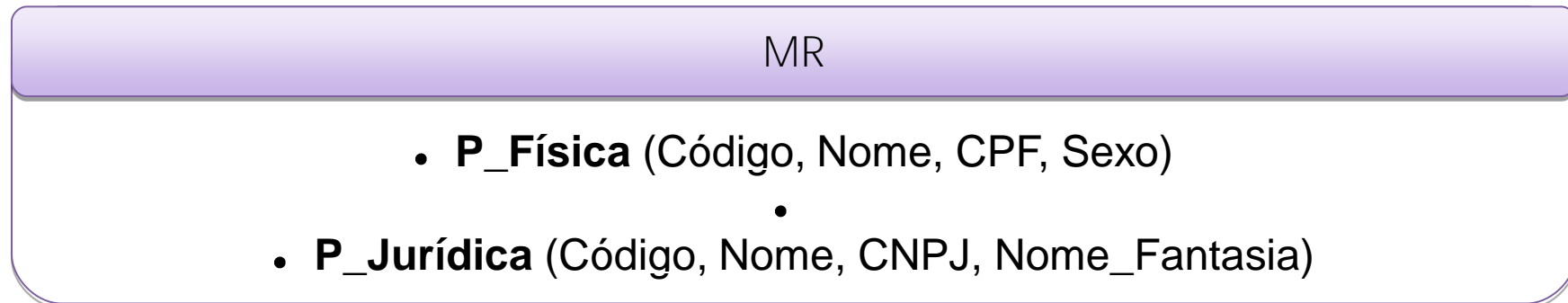
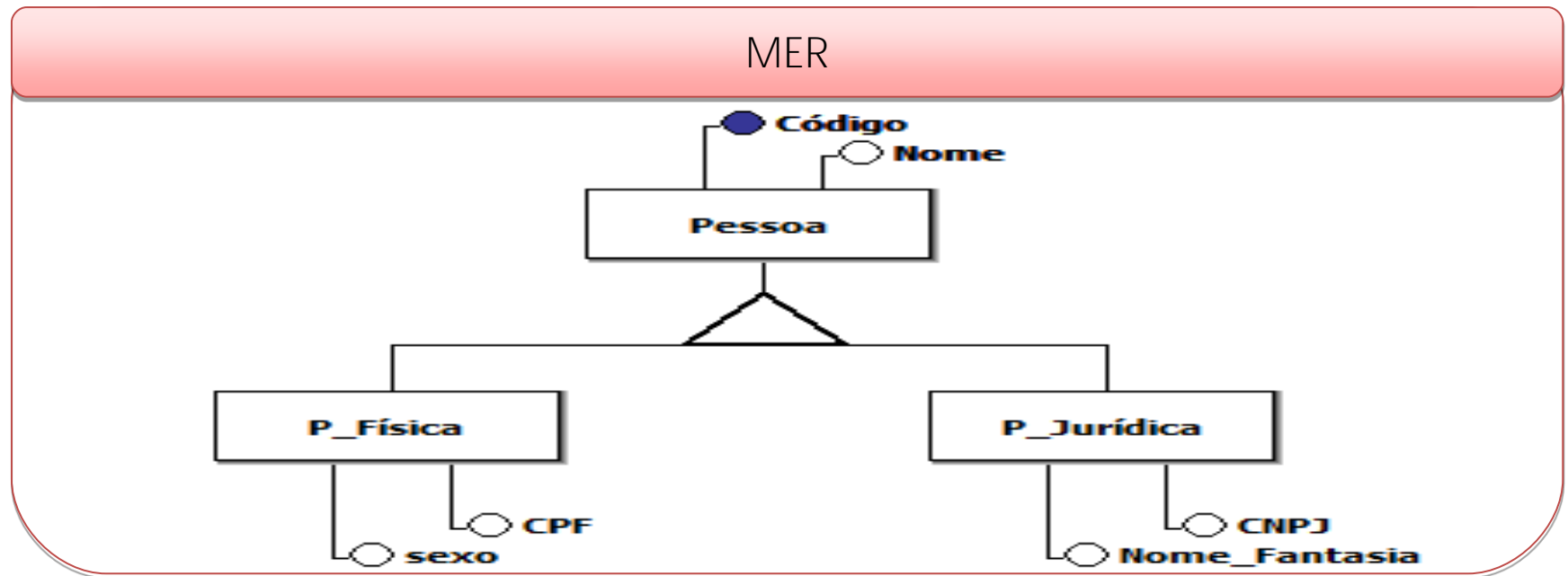
# Modelagem Lógica

- Generalização/Especialização:
  - Solução 2: tabela única para toda hierarquia
    - Ex.:



# Modelagem Lógica

- Generalização/Especialização:
  - Solução 3: tabela apenas para entidades especializadas. Ex:



# Modelagem Lógica

---

- Generalização/Especialização: **Comparando soluções:**
  - S1: Tabela para cada entidade:
    - **Vantagens**
      - É a que melhor reflete o modelo conceitual
      - As colunas desta relação são correspondentes aos atributos específicos da entidade.
    - **Desvantagens**
      - Necessidade de realizar junções entre as tabelas.
      - Baixo desempenho da manipulação das relações
        - Inserções e remoções e junções.

# Modelagem Lógica

---

- Generalização/Especialização: **Comparando soluções:**
  - S2: Tabela única:
    - **Vantagens**
      - Consulta rápida;
      - Todos os dados estão em uma única tupla, não sendo necessário efetuar junções;
      - A chave primária é armazenada uma única vez..
    - **Desvantagens**
      - Existe a ocorrência de muitos campos nulos;
      - Quando uma instância é modificada na superentidade, cada uma das relações correspondentes as suas entidades filhas devem ser modificadas.

# Modelagem Lógica

---

- Generalização/Especialização: **Comparando soluções:**
  - S3: Tabela para entidades especializadas:
    - **Vantagens**
      - A implementação é simples;
      - Há uma facilidade nas situações em que instâncias mudam de entidade.
    - **Desvantagens**
      - Há muitas junções e não pode ser implementada se possuir relacionamento envolvendo a entidade genérica.
      - Alteração de esquema na adição ou remoção de atributos.
      - Tem o potencial de desperdiçar bastante espaço de armazenamento:
      - hierarquia com várias entidades “irmãs”
      - instâncias pertencem a uma, e somente uma, entidade da hierarquia.

# Modelagem Lógica

---

- Generalização/Especialização: **Comparando soluções:**
  - S3: Tabela para entidades especializadas:
    - **Atenção !!**
      - Se existir um relacionamento com a entidade **mais genérica** **NÃO** pode ser aplicada a
      - **Solução 3**



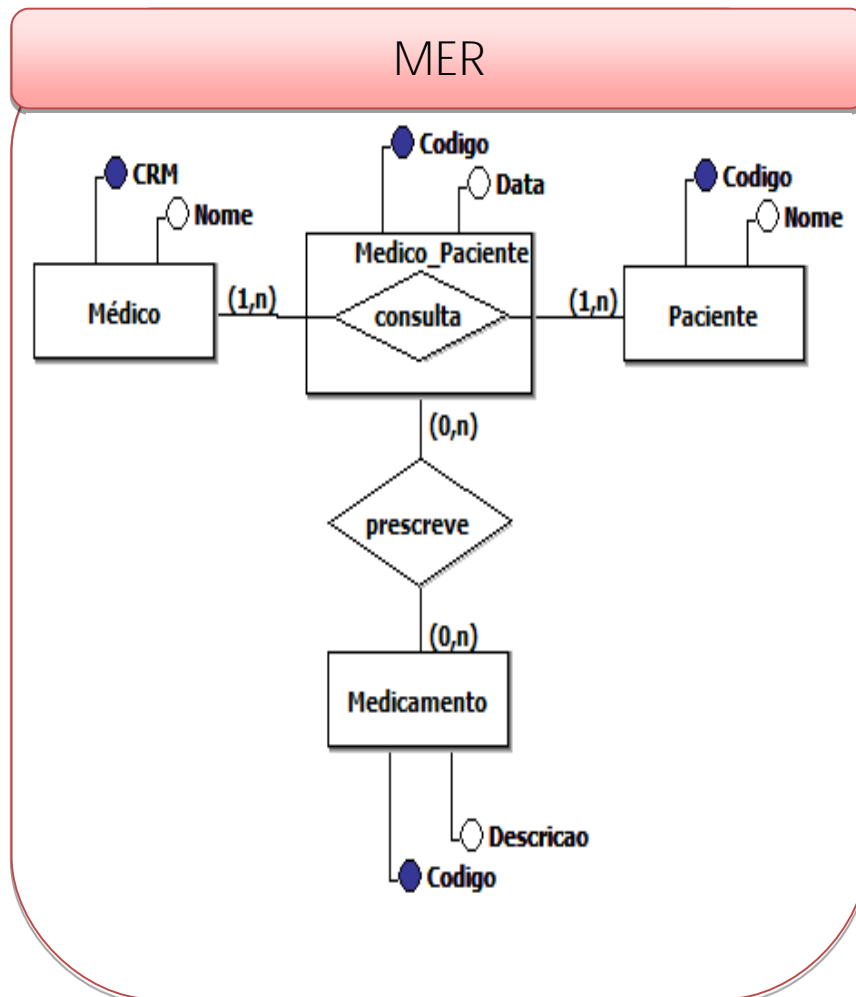
# Modelagem Lógica

---

- Agregação:
  - Solução:
    - Criar tabela para cada entidade;
    - A agregação deve ser vista como dois relacionamentos e deve criar tabela de acordo com as regras aplicadas aos relacionamentos.

# Modelagem Lógica

- Agregação:
  - Ex.:



MR

- **Médico** (CRM, Nome)
- **Paciente** (Codigo, Nome)
- **Medicamento** (Codigo, Descricao)
- 
- **Medico\_Pac** (Codigo, CRM, Cod\_Paciente, Data)
- 
- *CRM referencia Médico (CRM);*
- *Cod\_Paciente referencia Paciente (Codigo)*
- 
- **Cons\_Medicamento** (Cod\_Consulta, Cod\_Medicamento)
- 
- *Cod\_Consulta referencia Medico\_Pac (Codigo);*
- *Cod\_Medicamento referencia Medicamento (Codigo)*

# Modelagem Lógica

---

Qual das situações deverá ser adotada no mapeamento de especialização/generalização?

A **prioridade da escolha** deve ser adequada à necessidade do cliente e à estratégia de ter ou manter uma base de dados consistente, eficiente e rápida!

# Modelagem Lógica

---

Como no DER gerado no EC1 não houveram casos de Generalização/Especialização/Agregação, não houve necessidade de alteração no MR gerado anteriormente.

# Modelagem Lógica

---

- Após aplicação das regras, temos o seguinte Modelo Relacional:
  - Obras (id, titulo, tipo, autor\_principal, ano\_publicacao, data\_reserva, horario, data\_retirada, #id\_editora, #id\_usuario)
  - Exemplares (id, situacao)
  - Editora (id, nome, cidade)
  - Emprestimo (id, data, horario, previsao\_retorno, devolucao, #id\_exemplar)
  - Funcionario (id, matricula, nome, #id\_departamento)
  - Departamento (id, nome)
  - Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)
  - Obras\_Exemplares (id, #id\_obras, #id\_exemplares) //inserindo chave cega
  - Telefone(id, telefone, #id\_usuario) //inserindo chave cega
  - Realizar\_Emprestimo (id, id\_funcionario\_devolucao, #id\_emprestimo, #id\_usuario, #id\_funcionario) //inserindo chave cega

# Dicionário de Dados

---

- Para facilitar o seu entendimento do banco de dados e, principalmente, a comunicação entre as partes interessadas (cliente e setores da empresa), o esquema relacional gerado deve ser descrito em um dicionário de dados;
- 
- Este dicionário deve conter as seguintes informações:
  - Descrição de todas as relações;
  - Descrição de cada atributo das relações;
    - Tipo de dado, restrições, etc;

# Dicionário de Dados

- Ex.:

Nome_da_Tabela (descrição da tabela, se necessário)			
Atributo	Descrição	Tipo	Restrições
id	Atributo que representa o código de identificação do departamento	Texto	■ Chave Primária
cpf	Atributo que representa o CPF do presidente	Texto	■ Não Nulo ■ Atributo único
---	---	---	---
---	---	---	---

Os dicionários de dados são muito importantes, porque será a partir deles que deverão ser gerados os Scripts SQL DDL

# Dicionário de Dados

- Ex.:

Obras			
Atributo	Descrição	Tipo	Restrições
id	Atributo que representa o código da obra.	Inteiro	■ Chave Primária ■ Autoincremento
titulo	Título ou nome da obra	Texto	■ Não Nulo
tipo	Tipo da Obra	Inteiro	1. Livro, 2. Periódicos, 3.CD, 4. DVD, 5. Outros
autor_principal	Nome do autor	Texto	Não-Nulo
ano_publicacao	Ano da publicação	Inteiro	Não-Nulo
data_reserva	Data da reserva	Data	--
Horario	Horario da reserva	Hora	---
data_retirada	Data da retira	Data	---
#id_editora	Editora	Inteiro	FK (tabela Editora) Não-Nulo
#Id_usuario	Usuario que faz a reserva	Inteiro	FK (tabela Usuario)

- As demais tabelas devem ser preenchidos pelos alunos.



# SQL, DDL, EC1

---

- Editora (id, nome, cidade)

```
create table editora (  
    id serial, nome varchar(50) not null,  
    cidade varchar (50), primary key (id)  
);
```

- Usuario (id, cpf, nome, rua, numero, bairro, cidade, uf, pais)

```
create table usuario (  
    id serial, cpf varchar(11) not null unique,  
    nome varchar(50) not null, rua varchar(50),  
    numero varchar(10), bairro varchar(50),  
    cidade varchar (50), uf varchar(2),  
    pais varchar(50), primary key (id)  
);
```

# SQL, DDL, EC1

---

- Obras (id, titulo, tipo, autor\_principal, ano\_publicacao, data\_reserva, horario, data\_retirada, #id\_editora, #id\_usuario)

```
create table obras (  
  id serial, titulo varchar(50) not null,  
  tipo int not null,  
  autor_principal varchar(50) not null,  
  ano_publicacao int not null,  
  data_reserva date, horario time,  
  data_retirada date, id_editora int not null,  
  id_usuario int,  
  primary key (id),  
  foreign key (id_editora) references  
    editora(id),  
  foreign key (id_usuario) references  
    usuario(id) );
```

# SQL, DDL, EC1

---

- Exemplares (id, situacao)

```
create table exemplares (  
    id serial, situacao boolean,  
    primary key (id)  
);
```

- Obras\_Exemplares (id, #id\_obras, #id\_exemplares)

```
create table obras_exemplares (  
    id serial, /*chave cega*/  
    id_obras int, id_exemplar int,  
    primary key (id),  
    foreign key (id_obras) references obras(id),  
    foreign key (id_exemplar) references  
        exemplares(id)  
);
```

# SQL, DDL, EC1

---

- Emprestimo (id, data, horario, previsao\_retorno, devolucao, *#id\_exemplar*)

```
create table emprestimo (  
    id serial, data date not null,  
    horario time not null,  
    previsao_retorno date not null,  
    devolucao date, id_exemplar int,  
    primary key (id),  
    foreign key (id_exemplar) references  
        exemplares(id)  
);
```

- Departamento (id, nome)

```
create table departamento (  
    id serial, nome varchar(50) not null,  
    primary key (id) );
```

# SQL, DDL, EC1

---

- **Funcionario (id, matricula, nome, #*id\_departamento*)**

```
create table funcionario (  
    id serial, matricula varchar(20) not null,  
    nome varchar(50) not null,  
    id_departamento int,  
    primary key (id),  
    foreign key (id_departamento) references  
departamento(id)  
);
```

- **Telefone(id, telefone, #*id\_usuario*)**

```
create table telefone (  
    id serial, telefone varchar(50) not null,  
    id_usuario int,  
    primary key (id),  
    foreign key (id_usuario) references  
usuario(id));
```

# SQL, DDL, EC1

---

- Realizar\_Emprestimo (id, id\_funcionario\_devolucao, *#id\_emprestimo, #id\_usuario, #id\_funcionario*)  
create table realizar\_emprestimo (  
    id serial, /\*chave cega\*/  
    id\_funcionario\_devolucao int,  
    id\_emprestimo int, id\_usuario int,  
    id\_funcionario int,  
    primary key (id),  
    foreign key (id\_usuario) references  
        usuario(id),  
    foreign key (id\_emprestimo) references  
        emprestimo(id),  
    foreign key (id\_funcionario) references  
        funcionario(id)  
);

# Quadro Resumo

Item	MER	MR
Informação a ser persistida	Entidade	Tabela
Características	Atributo	Coluna (campo) ou tabela
Vínculo entre as informações	Relacionamento	Tabela ou chave
Herança de Informação	Generalização / Especialização	Tabelas
Conjunto de informações	Agregação	Tabelas e relacionamentos

# Exercícios de Modelagem

---

- Para prática na disciplina, desenvolva o DER, MR e o Dicionário de Dados dos casos abaixo:

1) Em Novembro de 2016 será realizado o II-JATI (II Jornada de Atualização em Tecnologia da Informação), na UAST-UFRPE. Para a realização deste evento foi necessário a criação de um sistema para organizá-lo.

- O evento possui diversos minicursos que serão ministradas em diversas salas. Os minicursos possuem um código, título e tema. Cada minicurso é ministrado por um professor em uma determinada sala.
- As salas possuem um número e a quantidade máxima de lugares. Também é necessário armazenar o horário e o dia de cada minicurso ministrado. Pois os minicursos podem acontecer uma ou mais vezes durante o evento.
- É necessário armazenar os dados de todos os professores. Para cada professor devemos guardar o seu nome, cpf, endereço (rua, bairro, cidade, UF), telefone, e-mail e a quantidade de minicursos que está ministrando.
- Cada minicurso pode ter vários participantes que foram previamente inscritos. Um participante pode se inscrever em vários cursos. Cada participante possui nome, cpf, endereço, curso, telefones e e-mails.
- Cada minicurso possui um ou mais patrocinadores. Um mesmo patrocinador pode patrocinar vários minicursos. Devem ser cadastradas as informações do patrocinador: Nome, CNPJ e o tipo (patrocinador Ouro, Prata ou Bronze).



# Exercícios de Modelagem

---

- Para prática na disciplina, desenvolva o DER, MR e o Dicionário de Dados dos casos abaixo:

2) Em uma seguradora de automóveis, um cliente tem pelo menos um carro e um carro pertence a um único cliente. Cada carro pode ter um número de acidentes associados a ele, devendo ser armazenados um código deste acidente, a data, o local e uma descrição do acidente. O acidente pode ser com vítima ou sem vítima. Se for com vítima, devem ser armazenados um histórico (contendo os nomes das vítimas e o tipo de lesão sofrida) e o valor gasto com indenização das vítimas. Se for sem vítima deve ser armazenado o valor gasto com danos materiais.

3) Em um hospital, um paciente pode realizar consultas com vários médicos que atendem vários pacientes. Em cada consulta podem ser requisitados exames a serem realizados. Cada exame corresponde a uma consulta. Devem ser armazenados os dados do paciente (CPF, Nome, Data de Nascimento e Endereço), os dados dos médicos (CRM, Nome, Especialidade) os dados dos exames (Número do prontuário, descrição e resultado).

4) Em uma biblioteca há vários tipos de materiais (livros, revistas e audiovisual). Cada material possui um único código de identificação e o ano. Para os livros são armazenados o autor e a editora; as revistas têm a marca, volume, data e material de destaque; os materiais audiovisuais têm o nome do diretor e o tempo de duração. Um cliente pode retirar vários materiais e um material pode ser retirado por vários clientes. Para toda retirada devem ser armazenadas a data de retirada e a data de devolução.

# Exercícios

---

- Para prática na disciplina, desenvolva o DER, MR e o Dicionário de Dados dos casos abaixo:
  - 5) Utilize o MER Estendido para modelar informações sobre PEÇAS de uma fábrica que possuem as seguintes informações: número da peça e descrição. Toda peça é manufaturada ou comprada ou então toda peça pode ser vendida ou trocada. As peças manufaturadas têm nº do lote e a data de fabricação. As peças compradas têm o nome do fabricante e o preço registrados. As peças vendidas possuem um código de venda e um valor. As peças trocadas possuem um código de troca e um valor que é a diferença da troca. Um cliente pode comprar as peças vendidas. Um cliente também pode trocar as peças que são disponíveis para a troca. Para um cliente são cadastrados um código único, o seu nome, RG, endereço e telefones.
  - 6) Um programador pode trabalhar em vários programas e cada programa pode ser elaborado por um conjunto de programadores.
    - Acrescente ao diagrama um tipo de entidade Analista, sabendo-se que:
      - Um analista avalia o trabalho dos programadores em programas e atribui uma nota.
      - Todo trabalho de um programador em um programa é sempre avaliado por apenas um analista.
    - Altere a última suposição considerando que todo trabalho de um programador em um programa é sempre avaliado por mais que um analista.

# Próxima aula:

---

- JDBC

# Referências

---

- ROB, P. Coronel, C. (2011) Sistemas de Banco de Dados. 8ºEd. Editora Cengage Learning.
- ELMASRI, R. Navathe, B. S. (2011) Sistemas de Banco de Dados. 6º Ed. Editora Pearson.
- HEUSER, Carlos A. Projeto de Banco de Dados. (1991)