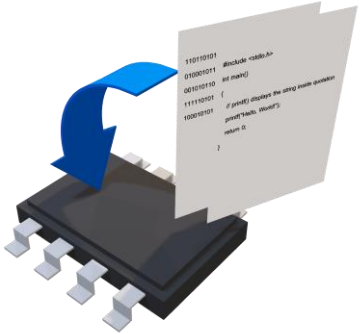
UNISINOS

DESAFIE O AMANHÃ.

BANCO DE DADOS PARA ENGENHARIA

Prof. Armando L. Keller



PROJETO LÓGICO: ER PARA RELACIONAL

PROJETO LÓGICO: ER PARA RELACIONAL

Depois de realizar a modelagem do bancos em alto nível com o diagrama ER é necessário “traduzi-lo” para SQL para que possa ser implementado.

Para fazer esta conversão existem algumas estratégias padrão para gerar um esquema de banco de dados relacional muito próximo do projeto ER.

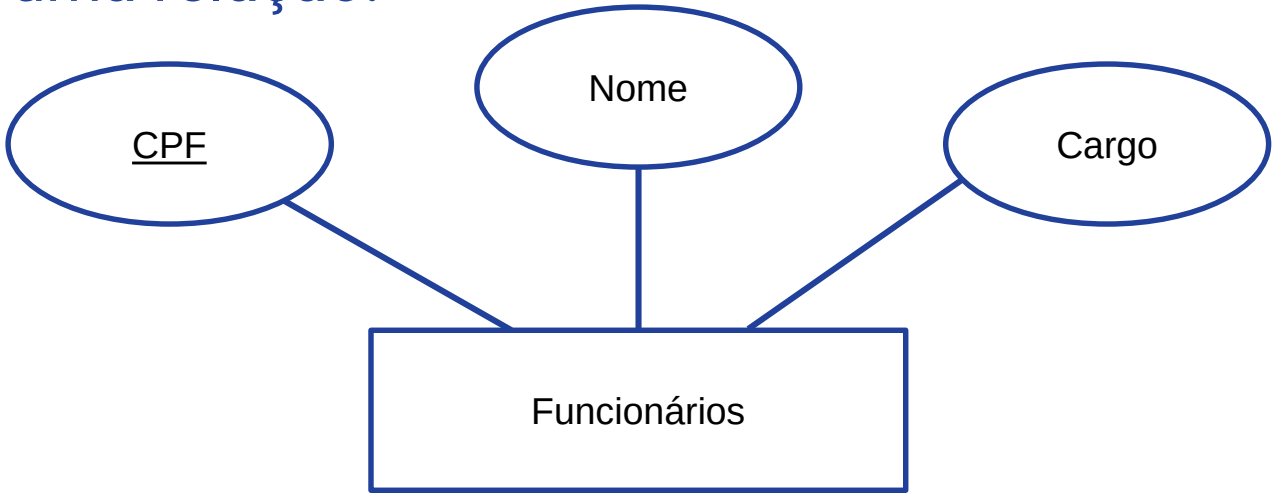
Conjunto de entidades para tabelas

Conjunto de entidades para tabelas

A transformação de um conjunto de entidades em uma relação é realizado de maneira bastante direta, onde cada atributo do conjunto de entidades será um atributo (coluna) da tabela.

Conjunto de entidades para tabelas

Exemplo: Converter o conjunto de entidades Funcionarios para uma relação.



Conjunto de entidades para tabelas

Neste caso o nome da nossa relação será Funcionários, e as colunas serão cpf, nome e vaga. Sendo que CPF será a chave primária.
A instrução SQL para a criação dessa tabela pode ser:

CREATE TABLE Funcionarios (cpf CHAR(11), nome CHAR(30), vaga INTEGER, **PRIMARY KEY**(cpf));

Conjunto de entidades para tabelas

Representação de uma instância do conjunto de entidades Funcionários:

cpf	*nome*	*vaga*
123-22-3666	Attishoo	48
231-31-5368	Smiley	22
131-24-3650	Smethurst	35

Conjunto de relacionamentos (sem restrições) para tabelas

Conjunto de relacionamentos (sem restrições) para tabelas

Os conjuntos de relacionamentos, assim como os conjuntos de entidades também é mapeado em uma relação no modelo relacional.

Para realizar a representação e um relacionamento, deve-se identificar cada entidade participante e fornecer valores para os atributos descritivos do relacionamento.

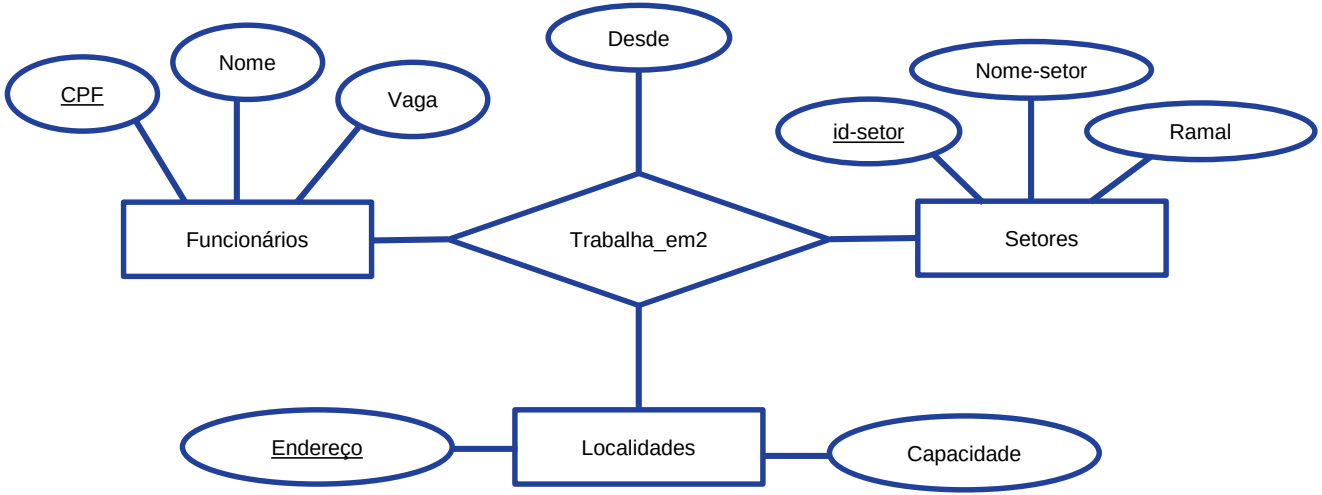
Conjunto de relacionamentos (sem restrições) para tabelas

Estes atributos incluem os atributos de chave primária de cada conjunto de entidades participante do relacionamento, através de campos de chave estrangeiras. Assim como os atributos descritivos do relacionamento.

O conjunto de atributos não descritivos (chaves primarias das entidades participantes) formará uma superchave para a relação.

Conjunto de relacionamentos (sem restrições) para tabelas

Exemplo: Converter o conjunto de relacionamentos Trabalha_em2 em uma tabela.



Conjunto de relacionamentos (sem restrições) para tabelas

Exemplo: Comando SQL para criação da tabela que representa o relacionamento Trabalha_em2.

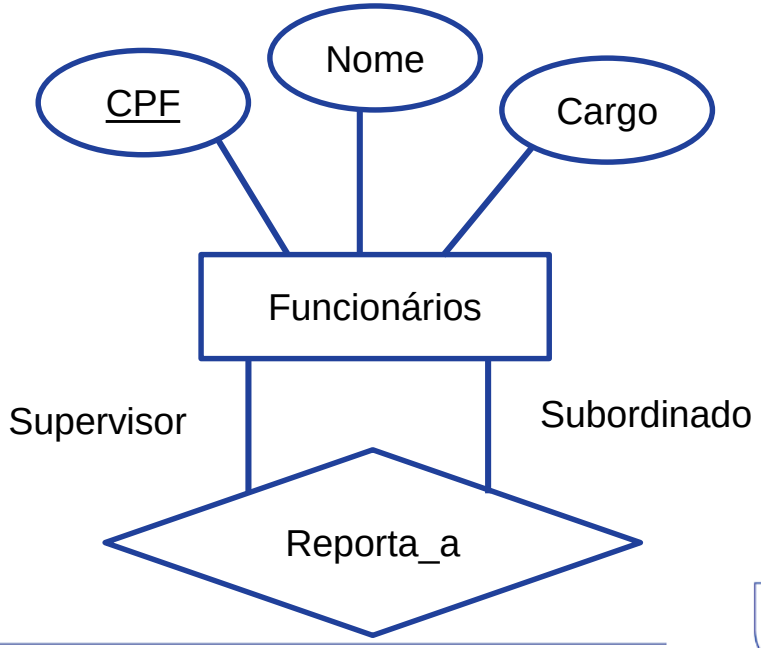
CREATE TABLE Trabalha_em2(cpf char(11), id_setor integer, endereco char(20), desde DATE, **PRIMARY KEY** (cpf, id_setor, endereco), **FOREIGN KEY** (cpf) **REFERENCES** Funcionarios, **FOREIGN KEY** (endereco) **REFERENCES** Localidades, **FOREIGN KEY** (id_setor) **REFERENCES** Setores);

Conjunto de relacionamentos (sem restrições) para tabelas

No caso de um auto-relacionamento, os indicadores de papel são utilizados para criar nomes de campos significativos.

Conjunto de relacionamentos (sem restrições) para tabelas

Exemplo: Converter o conjunto de relacionamentos Reporta_a em uma tabela.



Conjunto de relacionamentos (sem restrições) para tabelas

```
CREATE TABLE Reporta_a(
supervisor_cpf CHAR (11),
Subordinado_cpf CHAR(11),
PRIMARY KEY (supervisor_cpf, subordinado_cpf),
FOREIGN KEY (supervisor_cpf) REFERENCES Funcionarios (cpf),
FOREIGN KEY (subordinado_cpf) REFERENCES Funcionarios(cpf));
```

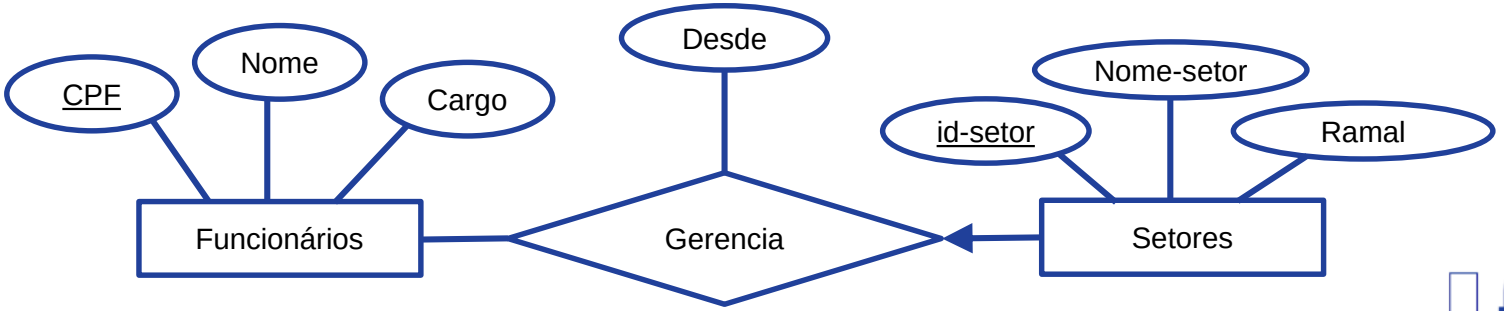
Conjunto de relacionamentos com restrições de chave

Conjunto de relacionamentos com restrições de chave

Caso um conjunto de entidades esteja vinculado a um conjunto de relacionamentos através de uma seta, temos uma restrição e chave. Existe mais de uma estratégia para representar estas restrições de chave.

Conjunto de relacionamentos com restrições de chave

Exemplo: Considerando o diagrama ER abaixo, pelo que foi visto pelo caso sem restrições de chave, a tabela gerencia terá os campos *cpf*, *id_setor* e *desde*.



Conjunto de relacionamentos com restrições de chave

Neste caso a escolha da chave primaria garantirá que somente possa ter um gerente por setor.

A escolha da chave primaria {cpf,id_setor} não garante a restrição pois aceitaria mais de um cpf por setor, no entanto utilizar somente o campo id_setor como chave primária, limita a apena um cpf por setor.

Conjunto de relacionamentos com restrições de chave

A instrução SQL para a criação da tabela pode ser:

```
CREATE TABLE  Gerencia (cpf CHAR(11),
 id_setor INTEGER,
 desde DATE,
 PRIMARY KEY (id_setor),
 FOREIGN KEY (cpf) REFERENCES Funcionarios,
 FOREIGN KEY (id_setor) REFERENCES Departamentos);
```

Conjunto de relacionamentos com restrições de chave

A outra estratégia disponível é levar os campos do conjunto de relacionamentos para o conjunto de entidades que possui a chave da restrição.

Esta abordagem evita a criação de uma nova tabela.

Conjunto de relacionamentos com restrições de chave

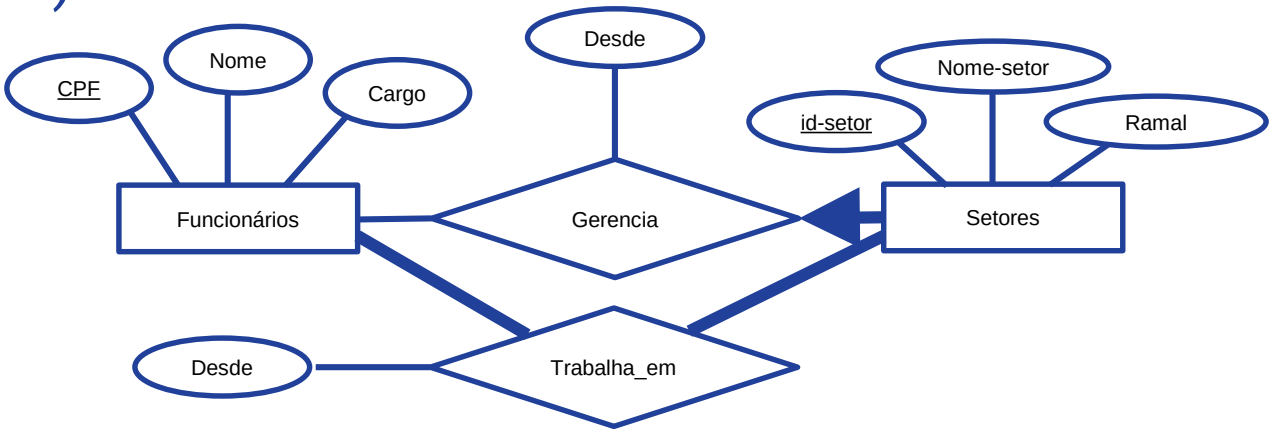
A instrução SQL para criar a tabela Setores com as informações de gerência pode ser:

CREATE TABLE Setores (id_setor INTEGER,
nome_setor CHAR(20),
orcamento REAL,
gerente_cpf char(11),
gerente_desde DATE,
PRIMARY KEY (id_setor),
FOREIGN KEY (gerente_cpf) **REFERENCES** Funcionarios(cpf));

Conjuntos de relacionamentos com restrições de participação

Conjunto de relacionamentos com restrições de participação

A restrição do vinculo entre setores e gerencia exige que todo o departamento possua um gerente (restrição de participação total), e que seja apenas um gerente (restrição de chave).



Conjunto de relacionamentos com restrições de participação

Para garantir estas restrições, deve-se utilizar a estratégia de levar os campos do conjunto de relacionamento para o conjunto de entidades, isto garantirá a restrição de chave. No entanto o comando SQL apresentado anteriormente permitia que o campo gerente_cpf aceitasse valores nulos ,ou seja, um setor não necessariamente teria um gerente. Para corrigir isto podemos utilizar alterar o comando para que não sejam aceitos valores nulos.

Conjunto de relacionamentos com restrições de participação

Exemplo de instrução SQL para criar a nova tabela Setores:

```
CREATE TABLE Setores (id_setor INTEGER,
nome_setor CHAR(20),
orcamento REAL,
gerente_cpf char(11) NOT NULL,
gerente_desde DATE NOT NULL,
PRIMARY KEY (id_setor),
FOREIGN KEY (gerente_cpf) REFERENCES
Funcionarios(cpf));
```

Conjunto de relacionamentos com restrições de participação

Já para as restrições de participação no conjunto de relacionamentos Trabalha_em, utilizando as restrições de chave e chave estrangeira, só é possível garantir a participação total de Funcionários, levando os campos para a tabela Funcionários e exigindo valores não nulos.

Mapeando conjuntos de entidades fracas

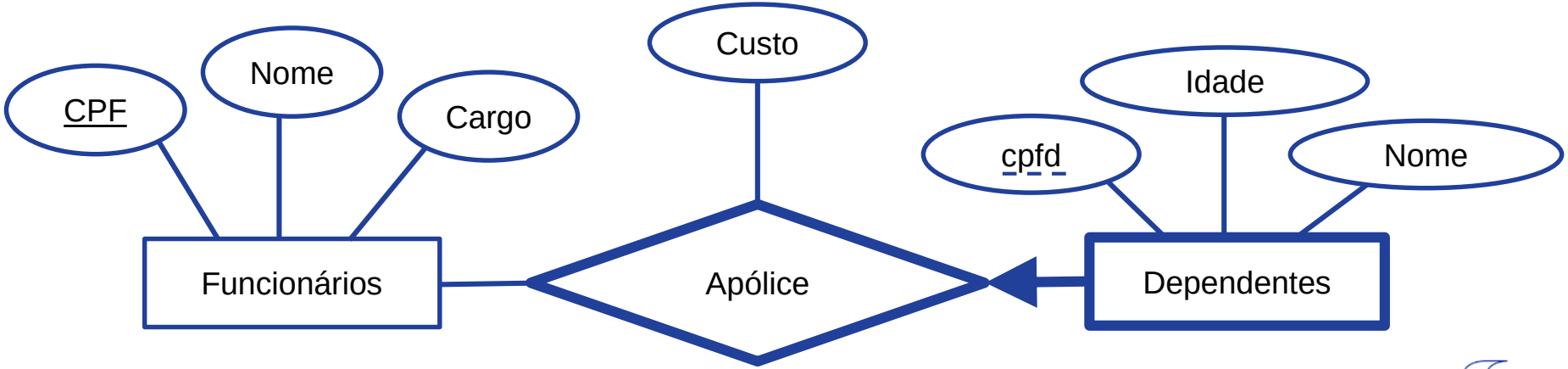
Mapeando conjuntos de entidades fracas

Os conjuntos de entidades fracas sempre participam de um relacionamento binário de um-para-muitos, e possuem uma restrição de chave e participação total.

Novamente utilizaremos a estratégia de levar os campos do relacionamento para o conjunto de entidades. E além disto, devemos garantir que quando a entidade proprietária for excluída, todas as entidades fracas associadas sejam excluídas também.

Mapeando conjuntos de entidades fracas

Exemplo: Criando a tabela Dependentes para o diagrama abaixo.



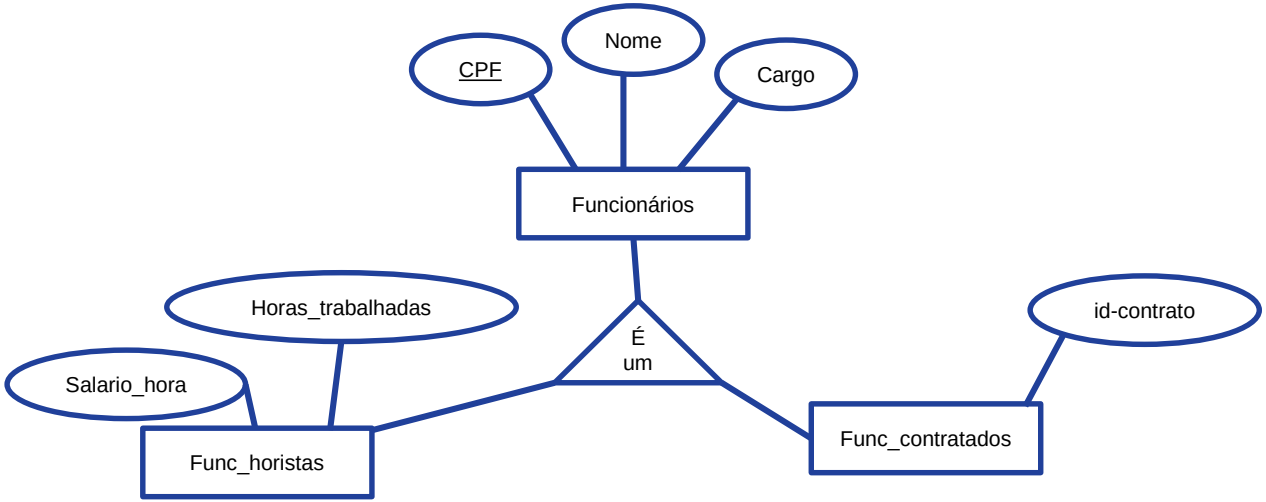
Mapeando conjuntos de entidades fracas

CREATE TABLE Dependentes (nomed CHAR(20),
idade INTEGER,
custo REAL,
funcionario_cpf CHAR(11),
PRIMARY KEY (nomded, funcionario_cpf),
FOREIGN KEY (funcionario_cpf) **REFERENCES** Funcionarios(cpf) **ON DELETE CASCADE**);

Mapeando hierarquias de classe

Mapeando hierarquias de classe

As hierarquias É-UM como a do exemplo abaixo podem ser implementadas de mais de uma maneira.



Mapeando hierarquias de classe

No entanto a maneira mais geral e comumente aplicada é criar uma tabela para a superclasse (Funcionarios) e outras tabelas para os demais conjuntos de entidades, neste caso Func_horistas e Func_contratados, utilizando a chave estrangeira para criar o vinculo.

Mapeando hierarquias de classe

Exemplo:
CREATE TABLE Func_contratados (cpf CHAR(11),
horas_trabalhadas INTEGER,
salario_hora REAL,
FOREIGN KEY (cpf) **REFERENCES** Funcionarios(cpf) **ON DELETE CASCADE**);

OBRIGADO.

UNISINOS

DESAFIE
O AMANHÃ.