Banco de Dados

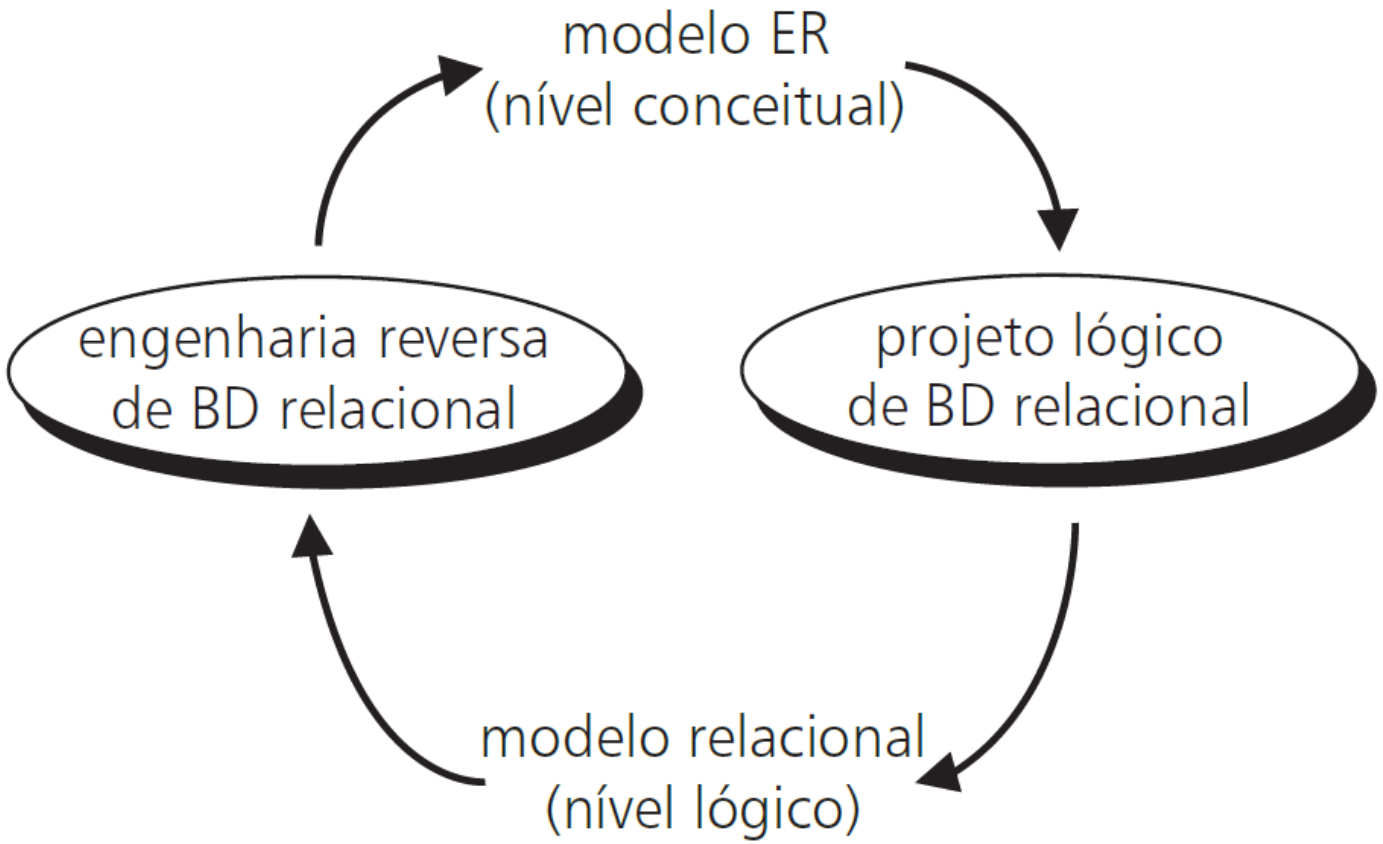
Prof. Thiago Cassio Krug

thiago.krug@iffarroupilha.edu.br

Transformação de ER para Relacional

- Durante a aula, foi mostrado duas formas de modelagem de dados:
  - a abordagem **entidade-relacionamento** (**ER**) e a abordagem **relacional**.
- Estas abordagens propõem modelar os dados em diferentes níveis de abstração.
- A abordagem ER é voltada à modelagem de dados de forma independente do SGBD, sendo adequada para a construção do **modelo conceitual**.
- Já a abordagem relacional modela os dados no nível de SGBD relacional.
- Um modelo neste nível de abstração é chamado de **modelo lógico**

Transformação de ER para Relacional



Transformação de ER para Relacional

- Há algumas regras para a transformação de um modelo ER para um modelo relacional.

- As regras foram definidas tendo em vista os dois objetivos básicos do projeto de BD:
  - Obter um banco de dados que permita **boa performance** de instruções de consulta e alteração do banco de dados.
  - Obter um banco de dados que **simplifique o desenvolvimento** e a manutenção de aplicações.

Transformação de ER para Relacional

- A fim de alcançar os dois objetivos citados, as regras de tradução foram definidas tendo por base, entre outros, os seguintes **princípios**:
  - Evitar junções
  - Diminuir o número de chaves
  - Evitar campos opcionais

Evitar Junções

- Evitar junções: Ter os dados necessários a uma consulta em uma única linha
- Um SGBD relacional normalmente armazena os dados de uma linha de uma tabela contiguamente em disco.
- Com isto, todos os dados de uma linha tabela são trazidos para a memória em uma única operação de acesso a disco.

Evitar Junções

- Isto significa que, uma vez encontrada uma linha de uma tabela, seus campos estão todos disponíveis, sem necessidade de acessos adicionais a disco.
  - Aumento de velocidade

- Quando for necessário buscar em disco dados de diversas linhas associadas pela igualdade de campos (por exemplo, buscar os dados de um empregado e os dados de seu departamento) é necessário usar a operação de **junção**.

Dept

CodigoDepto	NomeDepto
D1	Compras
D2	Engenharia
D3	Vendas

Emp

CodEmp	Nome	CodigoDepto	CategFuncional	CPF
E1	Souza	D1	—	132.121.331-20
E2	Santos	D2	C5	891.221.111-11
E3	Silva	D2	C5	341.511.775-45
E5	Soares	D1	C2	631.692.754-88

Evitar Junções

- Os SGBDs procuram implementar a junção de forma eficiente, já que ela é uma operação executada muito frequentemente.
- Mesmo assim, a junção envolve diversos acessos a disco.
- Assim, quando for possível, é preferível ter os dados necessários a uma consulta em uma única linha, ao invés de tê-los distribuídos em diversas linhas, exigindo a sua junção.

Diminuir o Número de Chaves

- Diminuir o número de chaves: Para a implementação eficiente do controle da unicidade da chave primária ou chave alternativa, o SGBD usa normalmente uma estrutura de acesso auxiliar, um **índice**.

- O índice permite que o SGBD teste rapidamente a existência do valor de uma chave primária sendo incluída, sem que seja necessário fazer uma leitura exaustiva de toda tabela.

Diminuir o Número de Chaves

- Também para o controle de chaves estrangeiras são utilizados índices.
- O índice é usado pelo SGBD quando uma chave primária estiver sendo excluída ou alterada.
- Neste caso, o índice permite o acesso rápido à tabela que contém a chave estrangeira, para verificação da existência do antigo valor da chave primária.

Diminuir o Número de Chaves

- Exemplificando, no caso do banco de dados abaixo, o índice para a chave estrangeira seria um índice na tabela Emp pela coluna `CodigoDepto`.

- Este índice seria usado, por exemplo, quando uma linha de Dept fosse excluída, para garantir a ausência na tabela Emp de linhas referenciando a linha de Dept excluída.

Dept

CodigoDepto	NomeDepto
D1	Compras
D2	Engenharia
D3	Vendas

Emp

CodEmp	Nome	CodigoDepto	CategFuncional	CPF
E1	Souza	D1	—	132.121.331-20
E2	Santos	D2	C5	891.221.111-11
E3	Silva	D2	C5	341.511.775-45
E5	Soares	D1	C2	631.692.754-88

Diminuir o Número de Chaves

- A regra geral de projeto físico de banco de dados é que, para cada chave primária, alternativa ou estrangeira, é necessário definir um índice.
- Pela forma em que são implementados, índices tendem a ocupar espaço considerável em disco.
- Além disso, a inserção ou remoção de entradas em um índice podem exigir múltiplos acessos a disco.
- Assim sendo, quando for necessário escolher entre duas alternativas de implementação, deve ser preferida aquela que exige o menor número de chaves e, consequentemente, o menor número de índices.

Diminuir o Número de Chaves

- Para exemplificar, vamos considerar que se deseja armazenar dados sobre clientes em um banco de dados relacional.
- Deseja-se armazenar, para cada cliente, seu código, seu nome, o nome da pessoa de contato, o endereço e o telefone.
- Estes dados poderiam ser implementados através de uma das seguintes alternativas:

```
Cliente(CodCliente,Nome,NomeContato,Endereço,Telefone)
```

ou

```
Cliente (CodCliente,Nome,NomeContato)
ClienteEnder (CodCliente,Endereço,Telefone)
  CodCliente referencia Cliente
```

Diminuir o Número de Chaves

- Na primeira alternativa, o SGBD cria apenas um índice por código de cliente, a chave primária da tabela.
- Na segunda alternativa, o SGBD cria, para cada tabela, um índice por código de cliente.
- Como cada cliente aparece nas duas tabelas, os dois índices possuem exatamente as mesmas entradas, resultando em armazenamento e processamento dobrados.
- A primeira alternativa é a preferida se considerarmos este princípio de projeto.

Evitar Campos Opcionais

- Evitar campos opcionais: Campos opcionais são campos que podem assumir o valor nulo (NULL em SQL).
- Os SGBDs relacionais usualmente não desperdiçam espaço ao armazenar um campo nulo, pois usam técnicas de compressão de dados e registros de tamanho variável no armazenamento interno de linhas.
- Assim, em princípio, não há problemas em usar este tipo de campo.

Evitar Campos Opcionais

- Uma situação que pode gerar problemas é aquela na qual a obrigatoriedade ou não do preenchimento de um campo depende do valor de outros campos.

- Neste caso, em alguns SGBDs, o controle da obrigatoriedade deve ser feito pelos programas que acessam o banco de dados, o que deve ser evitado.

Transformação de ER para Relacional

- Nos próximos slides é explicado o processo de projeto lógico, ou seja, os passos envolvidos na transformação de um modelo conceitual ER em um modelo lógico relacional.

- Neste processo, são aplicados os princípios descritos anteriormente.

- O processo de projeto lógico consta dos seguintes passos:
  1. Implementação inicial de entidades e respectivos atributos
  2. Implementação de relacionamentos e respectivos atributos
  3. Implementação de generalizações/especializações

Implementação Inicial de Entidades

- Cada entidade é traduzida para uma tabela.
- Neste processo, cada atributo da entidade define uma coluna desta tabela.
- Os atributos identificadores da entidade definem as colunas que compõem a chave primária da tabela.

Implementação Inicial de Entidades

- A figura abaixo apresenta um exemplo da transformação de uma entidade em uma tabela.
- A figura mostra o DER e o esquema relacional correspondente.
- A entidade PESSOA com seus atributos código, nome, endereço, data de admissão e data de nascimento é transformada na tabela denominada Pessoa com colunas denominadas `CodigoPess`, `Nome`, `Endereço`, `DataNasc` e `DataAdm`.

Implementação Inicial de Entidades

- Como o atributo código é identificador da entidade, a coluna correspondente a este atributo é a chave primária da tabela.

`Pessoa (CodigoPess,Nome,Endereço,DataNasc,DataAdm)`

Nomes de Atributos e Nomes de Colunas

- Não é aconselhável simplesmente transcrever os nomes de atributos para nomes de colunas.
- Nomes de colunas serão referenciados frequentemente em programas e outras formas de texto em computador.
- Assim, para diminuir o trabalho dos programadores é conveniente manter os nomes de colunas curtos.
  - Discordo.

Nomes de Atributos e Nomes de Colunas

- Além disso, em um SGBD relacional, o nome de uma coluna não pode conter brancos, nem hifens.
- Assim, nomes de atributos compostos de diversas palavras devem ser abreviados.
- Com base nestas considerações, os nomes de atributos data de nascimento e data de admissão foram traduzidos para os nomes de colunas `DataNasc` e `DataAdm` respectivamente.

Nomes de Atributos e Nomes de Colunas

- Nas linguagens de banco de dados, o nome da tabela é muitas vezes usado como qualificador do nome da coluna.
- Exemplificando, para referenciar a coluna Nome da tabela Pessoa muitas vezes é usado um termo na forma `Pessoa.Nome`.
- Por isso, não é recomendado incluir, no nome de uma coluna, o nome da tabela em que ela aparece.
- Assim, é preferível usar o nome de coluna `Nome`, a usar os nomes de coluna `NomePess` ou `NomePessoa`.

Nomes de Atributos e Nomes de Colunas

- A exceção a esta regra é a coluna chave primária da entidade.
- Como esta coluna pode aparecer em outras tabelas, na forma de chave estrangeira, é recomendável que os nomes das colunas que compõem a chave primária sejam sufixados ou prefixados com o nome ou sigla da tabela na qual aparecem como chave primária.
- Por este motivo, a coluna chave primária da tabela do exemplo recebeu a denominação de `CodigoPess`.

Nomes de Atributos e Nomes de Colunas

- Outra recomendação quanto à nomeação de colunas é relativa ao uso de abreviaturas.

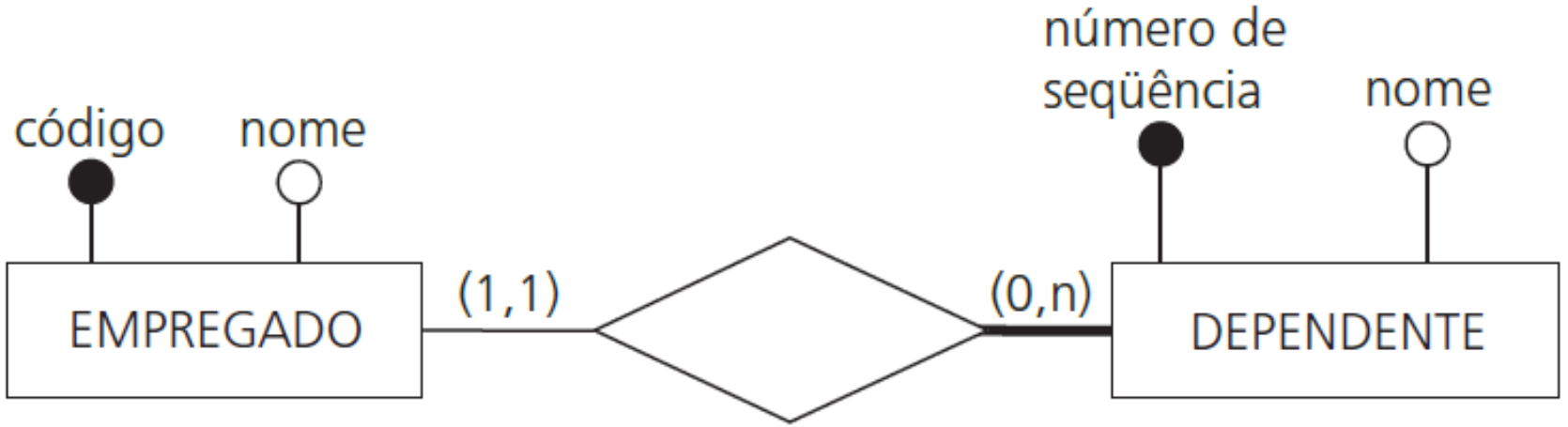
- Muitas vezes usa-se determinadas abreviaturas para tipos de campos que se repetem, como `Cod` para um código e `No` ou `Num` para um número.

- A recomendação é que se use sempre a mesma abreviatura em todo o banco de dados.

Relacionamento Identificador

- Vamos considerar como exemplo a entidade DEPENDENTE.

- Segundo o modelo abaixo, um dependente é identificado pelo código do empregado ao qual ele está vinculado e por um número de sequência que distingue os diversos dependentes de um mesmo empregado.



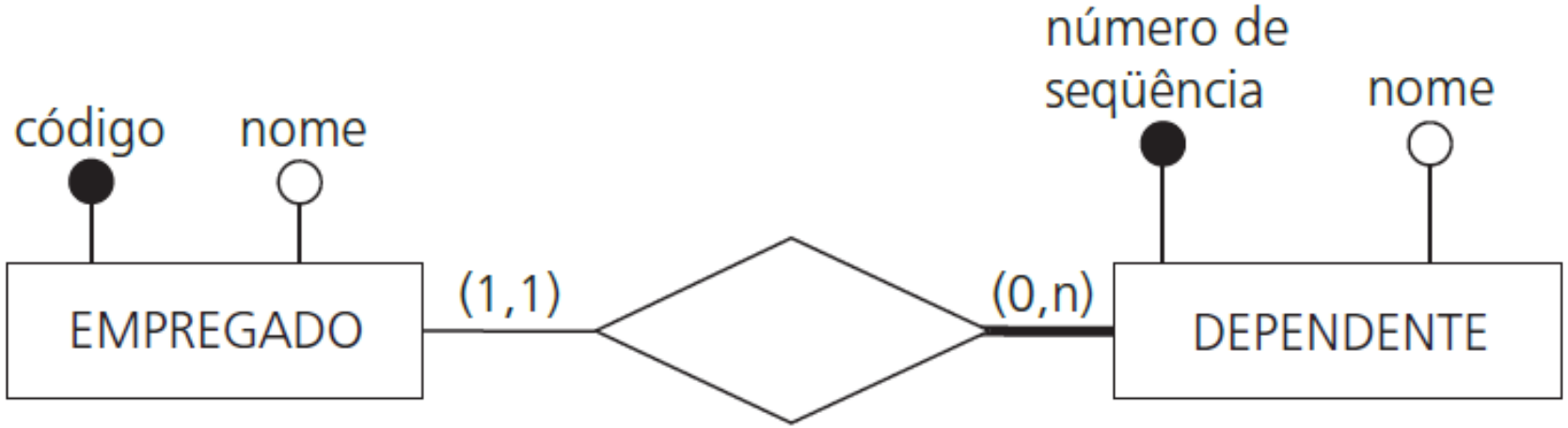
Relacionamento Identificador

- Para cada relacionamento identificador, é criada uma chave estrangeira na tabela que implementa a entidade identificada pelo relacionamento identificador.

- Esta chave estrangeira é formada pelas colunas da chave primária da tabela referenciada pela chave estrangeira.

- Isto significa que uma coluna `CodigoEmp` (chave primária da tabela correspondente à entidade EMPREGADO) deve ser adicionada à tabela Dependente.

Relacionamento Identificador




```
Empregado(CodigoEmp,Nome)
Dependente(CodigoEmp,NoSeq,Nome)
   CodigoEmp referencia Empregado
```

Relacionamento Identificador

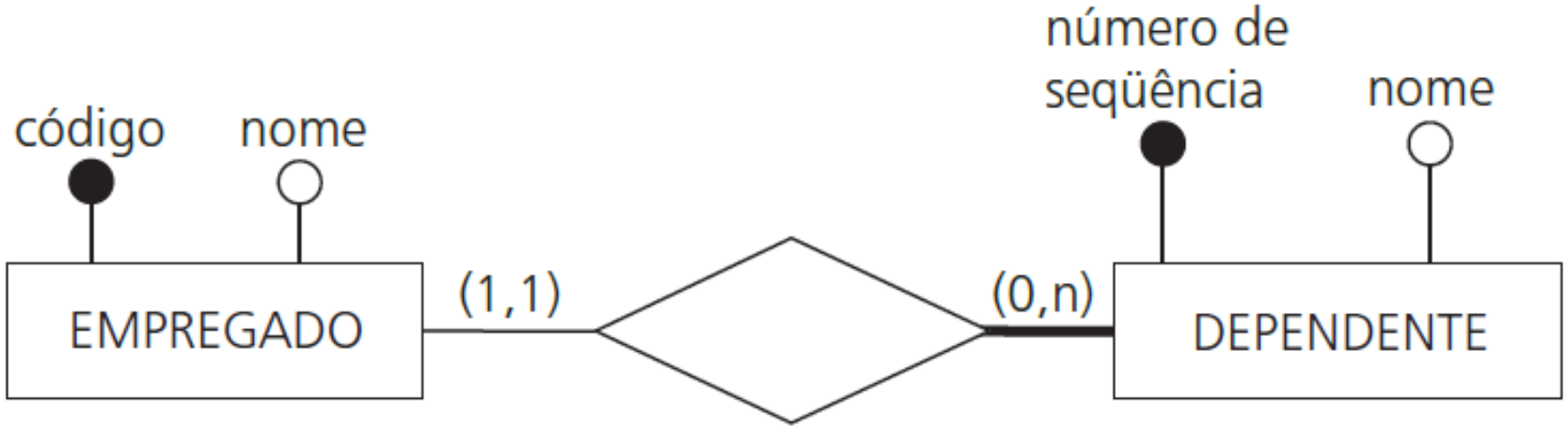
- A chave primária da tabela que implementa a entidade identificada pelo relacionamento identificador é formada por:

- Colunas correspondentes aos atributos identificadores da entidade, se existirem, e
- Chaves estrangeiras que implementam os relacionamentos identificadores.

Relacionamento Identificador

- Isto significa que a chave primária da tabela Dependente é composta pelas colunas `CodigoEmp` (chave estrangeira que implementa o relacionamento identificador) e `NoSeq` (coluna que implementa o atributo identificador número de sequência da entidade DEPENDENTE).

Relacionamento Identificador

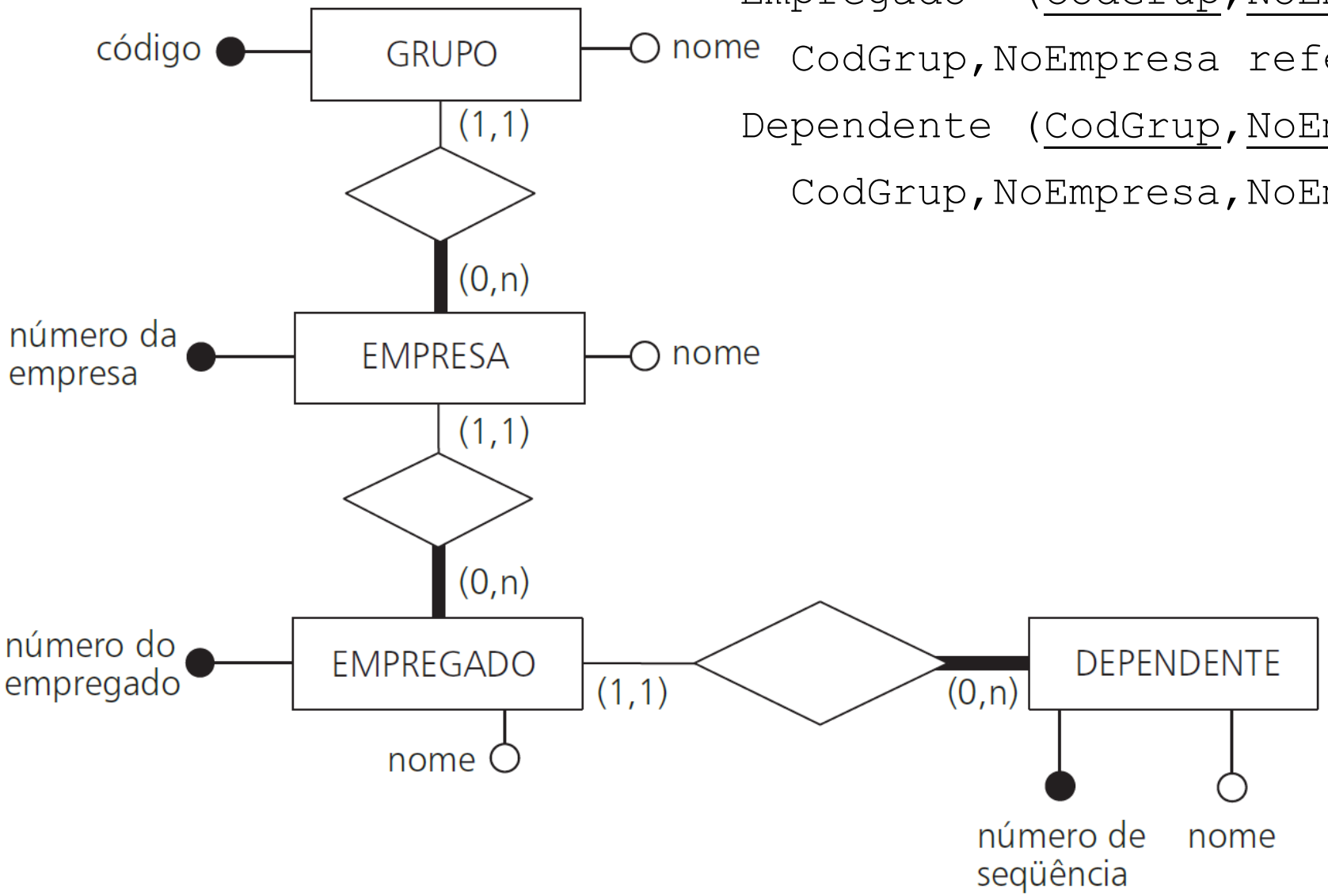



```
Empregado(CodigoEmp,Nome)
Dependente(CodigoEmp,NoSeq,Nome)
   CodigoEmp referencia Empregado
```

Relacionamento Identificador

- Cabe observar que, quando a entidade que possui o relacionamento identificador referencia uma entidade, que, por sua vez, também tem um relacionamento identificador, é necessário propagar por vários níveis a importação de chaves estrangeiras para a chave primária.

```
Grupo       (CodGrup,Nome)
Empresa     (CodGrup,NoEmpresa,Nome)
  CodGrup referencia Grupo
Empregado   (CodGrup,NoEmpresa,NoEmpreg,Nome)
  CodGrup,NoEmpresa referencia Empresa
Dependente  (CodGrup,NoEmpresa,NoEmpreg,NoSeq,Nome)
  CodGrup,NoEmpresa,NoEmpreg referencia Empregado
```



Referências

- HEUSER, C. A.; **Projeto de Banco de Dados**. 6ª edição. Editora Artmed, 2009.
- SILBERCHATZ, A.; KORTH, H. F.; SUDARSHA, S.; **Sistema de Banco de Dados**. 6ª edição. Editora Campus, 2012.
- AGELOTTI, E. S. **Banco de Dados**. Curitiba: Editora do Livro Técnico, 2010.
- RAMAKRISHNAN, R.; GEHRKE, J.; **Sistemas de Gerenciamento de Banco de Dados**. 3ª edição. Editora Mc Graw-Hill, 2008.
- DATE, C. J.; **Introdução a Sistemas de Bancos de Dados**. 8ª edição. Editora Campus, 2004.
- ELMASRI, R.; NAVATHE S. B.; **Sistemas de Banco de Dados**. 4ª edição. Editora Pearson, 2005.