

CURSO TÉCNICO EM INFORMÁTICA



BANCO DE DADOS



CURSO TÉCNICO EM INFORMÁTICA

O que é um banco de dados?

Um **banco de dados** é uma coleção de dados relacionados. Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significado implícito.

Por exemplo, considere os nomes, números de telefone e endereços das pessoas que você conhece. Você pode ter registrado esses dados em uma agenda ou, talvez, os tenha armazenado em um disco rígido, usando um computador pessoal e um software como **Microsoft Access** ou **Excel**. Essa coleção de dados relacionados, com um significado implícito, é um banco de dados.



CURSO TÉCNICO EM INFORMÁTICA

O que é um banco de dados?

Um **banco de dados** pode ser gerado e mantido manualmente, ou pode ser computadorizado.

Por exemplo, um **catálogo** de cartão de biblioteca é um banco de dados que pode ser criado e mantido manualmente.

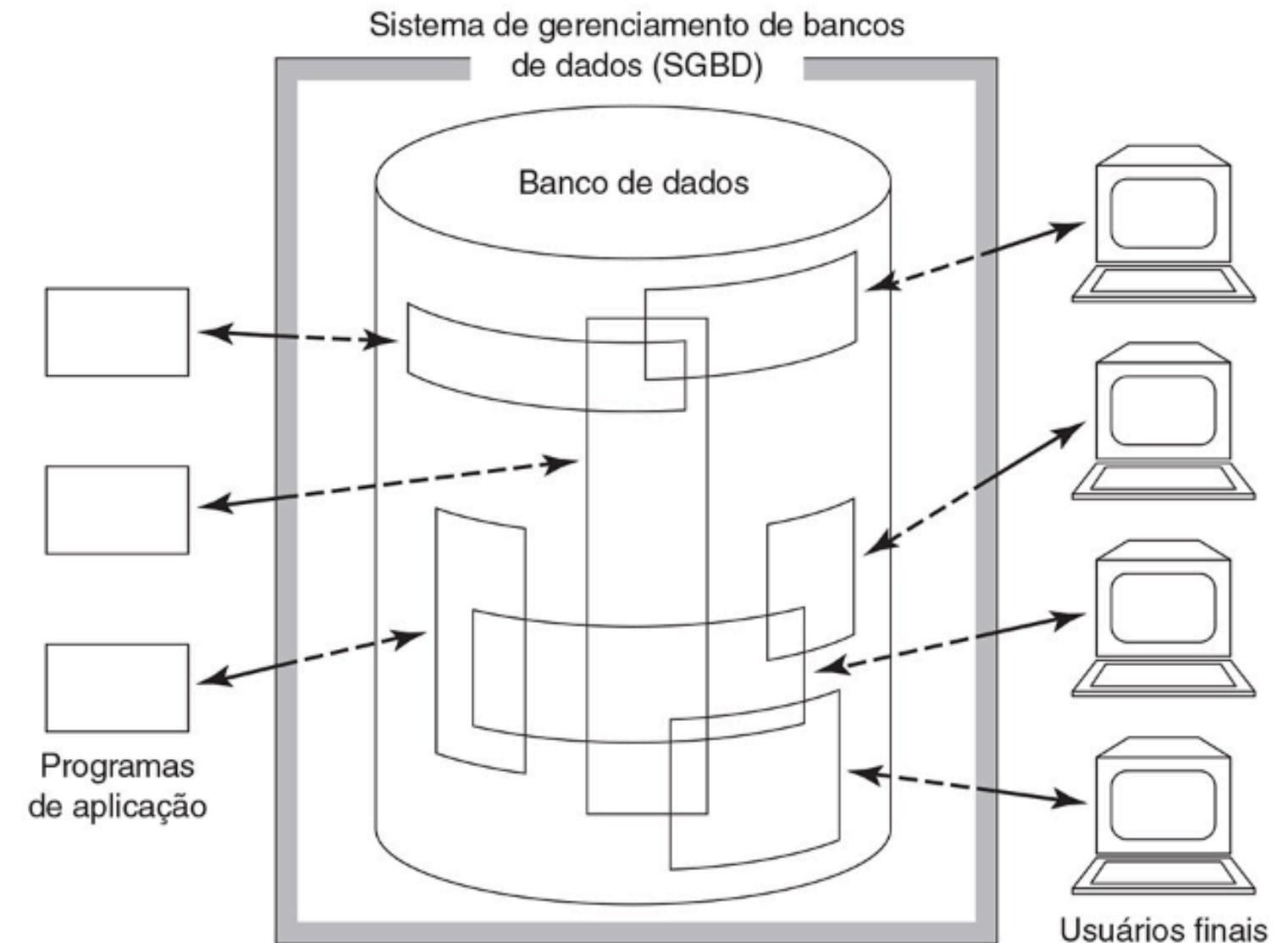
Um banco de dados computadorizado pode ser criado e mantido por um grupo de programas de aplicação escritos especificamente para essa tarefa ou por um **sistema gerenciador de banco de dados (SGBD)**.



Sistema gerenciador de banco de dados (SGBD — Database Management System)

É uma coleção de programas que permite aos usuários criar e manter um banco de dados.

O **SGBD** é um **sistema de software** de uso geral que facilita o processo de definição, construção, manipulação e compartilhamento de bancos de dados entre diversos usuários e aplicações.



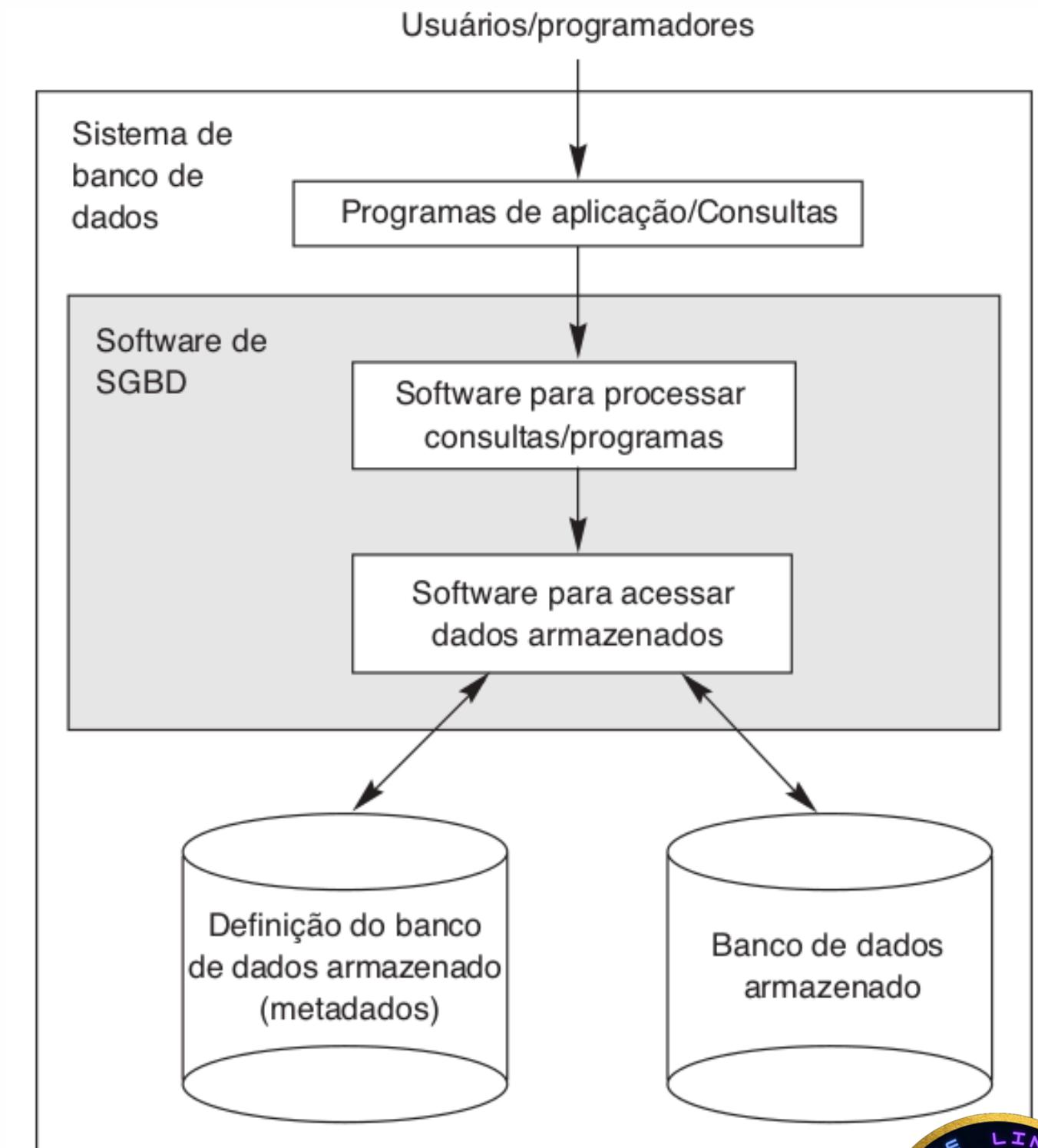
CURSO TÉCNICO EM INFORMÁTICA

Sistema gerenciador de banco de dados (SGBD)

Criar um **banco de dados** significa organizar e definir como as informações serão armazenadas e gerenciadas. Para isso, é necessário especificar:

- **Tipos de dados** (como números, textos, datas, etc.).
- **Estruturas** (como tabelas e relações entre os dados).
- **Restrições** (regras que garantem a integridade das informações).

Essa definição é registrada pelo Sistema de Gerenciamento de Banco de Dados (SGBD) em um catálogo chamado **metadados**, que funciona como um "manual" do banco de dados.



Etapas do Banco de Dados:

1. **Definição:** O banco de dados é planejado e configurado conforme as necessidades da aplicação.
2. **Construção:** Os dados começam a ser inseridos e armazenados pelo SGBD em um meio apropriado, como um **servidor**.
3. **Manipulação:**
Inclui ações como:
 - a. **Consultar:** Buscar informações específicas.
 - b. **Atualizar:** Modificar os dados conforme mudanças no mundo real.
 - c. **Gerar relatórios:** Criar documentos organizados a partir dos dados.
4. **Compartilhamento:** Permite que vários usuários ou sistemas accessem o banco de dados ao mesmo tempo, garantindo que as informações estejam disponíveis quando necessário.



Acesso e Gestão de Banco de Dados

Os **softwares** precisam acessar informações armazenadas em um **banco de dados**.

Isso acontece por meio de consultas e transações enviadas ao **Sistema de Gerenciamento de Banco de Dados (SGBD)**.

- **Consultas:** Servem para buscar informações específicas no banco de dados.
- **Transações:** Podem tanto recuperar dados quanto salvar novas informações.



Funções Importantes do SGBD

O **SGBD** não apenas gerencia o acesso aos dados, mas também desempenha funções essenciais para garantir a integridade e segurança do banco:

- **Proteção contra falhas:** Evita a perda de dados devido a problemas no hardware (computador, servidor) ou no software (erros no programa).
- **Segurança:** Controla o acesso aos dados, impedindo acessos não autorizados.
- **Manutenção a longo prazo:** Um banco de dados pode ser usado por muitos anos e deve ser capaz de se adaptar a novas necessidades.



Implementação de Banco de Dados

O SGBD não apenas gerencia o acesso aos dados, mas também desempenha funções essenciais para garantir a integridade e segurança do banco:

- Podemos usar um **SGBD de uso geral**, que é um software já pronto para gerenciar bancos de dados (como MySQL, PostgreSQL, SQL Server).
- Ou podemos criar nosso próprio sistema específico para armazenar e processar os dados.

Independentemente da escolha, gerenciar um banco de dados exige a criação de programas e processos sofisticados para garantir o seu funcionamento eficiente.



Sistema de Banco de Dados

Chamamos de sistema de banco de dados a união entre:

- **O banco de dados** (onde os dados são armazenados).
- **O SGBD** (software que gerencia o banco de dados).
- **Os programas de aplicação** (softwares que acessam e utilizam os dados).



CURSO TÉCNICO EM INFORMÁTICA

Exemplo: Banco de Dados de uma Universidade

Para entender melhor, imagine um banco de dados que armazena informações de uma universidade, incluindo:

- **Alunos** (dados pessoais, matrícula, curso).
- **Disciplinas** (código, nome, carga horária).
- **Turmas** (aulas específicas de cada disciplina em um semestre).
- **Histórico Escolar** (notas e aprovações dos alunos).
- **Pré-requisitos** (disciplinas que devem ser concluídas antes de cursar outras).

Essas informações são organizadas em arquivos estruturados, onde cada registro contém dados de um mesmo tipo. O SGBD gerencia todas essas informações, garantindo que professores, alunos e a administração possam acessá-las e atualizá-las de maneira segura e eficiente.



CURSO TÉCNICO EM INFORMÁTICA

Exemplo: Banco de Dados de uma Universidade

ALUNO			
Nome	Numero_aluno	Tipo_aluno	Curso
Silva	17	1	CC
Braga	8	2	CC



CURSO TÉCNICO EM INFORMÁTICA

Exemplo: Banco de Dados de uma Universidade

DISCIPLINA			
Nome_disciplina	Numero_disciplina	Creditos	Departamento
Introd. à ciência da computação	CC1310	4	CC
Estruturas de dados	CC3320	4	CC
Matemática discreta	MAT2410	3	MAT
Banco de dados	CC3380	3	CC



CURSO TÉCNICO EM INFORMÁTICA

Exemplo: Banco de Dados de uma Universidade

TURMA				
Identificacao_turma	Numero_disciplina	Semestre	Ano	Professor
85	MAT2410	Segundo	07	Kleber
92	CC1310	Segundo	07	Anderson
102	CC3320	Primeiro	08	Carlos
112	MAT2410	Segundo	08	Chang
119	CC1310	Segundo	08	Anderson
135	CC3380	Segundo	08	Santos



CURSO TÉCNICO EM INFORMÁTICA

Exemplo: Banco de Dados de uma Universidade

HISTORICO_ESCOLAR

Numero_aluno	Identificacao_turma	Nota
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A



CURSO TÉCNICO EM INFORMÁTICA

Exemplo: Banco de Dados de uma Universidade

PRE_REQUISITO	
Numero_disciplina	Numero_pre_requisito
CC3380	CC3320
CC3380	MAT2410
CC3320	CC1310



CURSO TÉCNICO EM INFORMÁTICA

Exemplo: Banco de Dados de uma Universidade

DISCIPLINA			
Nome_disciplina	Numero_disciplina	Creditos	Departamento
Introd. à ciência da computação	CC1310	4	CC
Estruturas de dados	CC3320	4	CC
Matemática discreta	MAT2410	3	MAT
Banco de dados	CC3380	3	CC

PRE_REQUISITO

Numero_disciplina	Numero_pre_requisito
CC3380	CC3320
CC3380	MAT2410
CC3320	CC1310



CURSO TÉCNICO EM INFORMÁTICA

Manipulação do Banco de Dados

O banco pode ser acessado por meio de consultas (para buscar informações) e atualizações (para modificar dados).

Exemplos de consultas:

- Listar todas as disciplinas e notas de um aluno.
- Buscar os alunos que cursaram "Banco de Dados" em um semestre específico.
- Listar os pré-requisitos de um curso.

Exemplos de atualizações:

- Alterar o tipo de aluno (por exemplo, de novato para segundo ano).
- Criar uma nova turma para uma disciplina.
- Inserir uma nova nota para um aluno.

Essas operações precisam ser definidas corretamente na linguagem de consulta do SGBD para serem executadas corretamente. O banco de dados faz parte de um sistema de informação, essencial para a organização e gestão acadêmica.



CURSO TÉCNICO EM INFORMÁTICA

Manipulação do Banco de Dados

O banco pode ser acessado por meio de consultas (para buscar informações) e atualizações (para modificar dados).

Exemplos de consultas:

- Listar todas as disciplinas e notas de um aluno.
- Buscar os alunos que cursaram "Banco de Dados" em um semestre específico.
- Listar os pré-requisitos de um curso.

Exemplos de atualizações:

- Alterar o tipo de aluno (por exemplo, de novato para segundo ano).
- Criar uma nova turma para uma disciplina.
- Inserir uma nova nota para um aluno.

Essas operações precisam ser definidas corretamente na linguagem de consulta do SGBD para serem executadas corretamente. O banco de dados faz parte de um sistema de informação, essencial para a organização e gestão acadêmica.



CURSO TÉCNICO EM INFORMÁTICA

Como é Criado um Banco de Dados?

A construção de um banco de dados passa por três etapas principais:

Especificação e Análise de Requisitos

- Identificar quais informações serão armazenadas.
- Documentar detalhadamente essas necessidades.

Implementação e Manutenção

O banco de dados é criado, preenchido com informações reais e atualizado continuamente para refletir mudanças no mundo real.

Projeto do Banco de Dados

- **Projeto Conceitual:** Criar um modelo visual dos dados e suas relações (como o modelo Entidade-Relacionamento).
- **Projeto Lógico:** Traduzir esse modelo para um formato compatível com um Sistema de Gerenciamento de Banco de Dados (SGBD).
- **Projeto Físico:** Definir como os dados serão armazenados e acessados no sistema.



Diferença entre Bancos de Dados e Arquivos Simples

Antes dos bancos de dados modernos, as empresas armazenavam informações em arquivos separados, cada um sendo gerenciado por um setor específico. Isso gerava problemas como:

Redundância de Dados: As mesmas informações eram armazenadas em lugares diferentes, desperdiçando espaço.

Dificuldade de Atualização: Modificar um dado exigia atualização manual em vários arquivos.

Falta de Integração: Diferentes setores da empresa tinham dificuldade em acessar as mesmas informações.



CURSO TÉCNICO EM INFORMÁTICA

Diferença entre Bancos de Dados e Arquivos Simples

Antes dos bancos de dados modernos, as empresas armazenavam informações em arquivos separados, cada um sendo gerenciado por um setor específico. Isso gerava problemas como:

Redundância de Dados: As mesmas informações eram armazenadas em lugares diferentes, desperdiçando espaço.

Dificuldade de Atualização: Modificar um dado exigia atualização manual em vários arquivos.

Falta de Integração: Diferentes setores da empresa tinham dificuldade em acessar as mesmas informações.

Solução com Bancos de Dados

Um banco de dados centralizado permite que todos os setores acessem e compartilhem as mesmas informações de maneira organizada, eliminando redundâncias e garantindo a consistência dos dados.



Características Principais de um Banco de Dados

Autodescrição do Sistema

- O banco de dados contém não apenas os dados reais, mas também uma descrição detalhada da sua estrutura e regras (chamadas de metadados).
- Essas informações são armazenadas em um catálogo dentro do SGBD.

Isolamento entre Programas e Dados

- Nos arquivos simples, os dados são fixos e vinculados ao programa que os usa.
- No banco de dados, as informações podem ser modificadas sem afetar os programas que as acessam, tornando o sistema mais flexível.



Características Principais de um Banco de Dados

Suporte para Múltiplas Visões

- Cada usuário pode ver apenas os dados que lhe interessam, sem precisar acessar todo o banco.
- Exemplo: Um setor financeiro pode acessar apenas informações de pagamentos, enquanto o setor acadêmico vê apenas notas dos alunos.

Compartilhamento e Controle de Acesso

- Múltiplos usuários podem acessar e modificar os dados simultaneamente.
- O SGBD gerencia o acesso para evitar conflitos. Por exemplo:
 - Dois funcionários não podem reservar o mesmo assento em um avião ao mesmo tempo.



CURSO TÉCNICO EM INFORMÁTICA

Exemplo Prático: Banco de Dados de uma Universidade

Imagine um banco de dados que gerencia informações acadêmicas.
Ele pode conter:

- **Alunos** (nome, número de matrícula, curso).
- **Disciplinas** (código, nome, carga horária).
- **Turmas** (professor, sala, horário).
- **Histórico Escolar** (notas e aprovações).
- **Pré-requisitos** (disciplinas que precisam ser feitas antes de outras).



CURSO TÉCNICO EM INFORMÁTICA

Exemplo Prático: Banco de Dados de uma Universidade

Exemplo de Consultas:

- Listar todas as disciplinas e notas de um aluno.
- Verificar quais alunos fizeram a disciplina "Banco de Dados" em um determinado semestre.
- Atualizar a nota de um aluno em uma disciplina.

Essas operações são feitas por meio de linguagens de consulta, como **SQL**, dentro do SGBD.



modelo de dados

Um **modelo de dados** é uma forma de descrever os tipos de informações que estão guardadas em um banco de dados. Por exemplo, em uma indústria, o modelo de dados pode indicar que o banco de dados armazena informações sobre produtos. Para cada produto, são registradas informações como o código, o preço e a descrição.

É importante notar que o modelo de dados não mostra quais produtos específicos estão no banco de dados — ele apenas define que tipo de informação o banco é capaz de armazenar.

modelo de dados

=

descrição formal da estrutura de um banco
de dados



CURSO TÉCNICO EM INFORMÁTICA

modelo de dados

Para construir um **modelo de dados**, usa-se uma linguagem de modelagem de dados. Linguagens de modelagem de dados podem ser classificadas conforme a forma de apresentar modelos, em linguagens textuais ou linguagens gráficas.

Como veremos adiante, um mesmo modelo de dados pode ser apresentado de várias formas. Cada apresentação do modelo recebe a denominação esquema de banco de dados.

Um **banco de dados** pode ser modelado em diferentes níveis de abstração, dependendo da intenção do modelador. Um modelo feito para usuários tende a ser mais simples e sem detalhes técnicos, enquanto um modelo para técnicos inclui informações mais detalhadas sobre a estrutura interna. No **projeto de banco de dados**, geralmente se consideram dois níveis principais: **o modelo conceitual e o modelo lógico**.



Modelo Conceitual

É a visão geral do banco de dados, focada no que será armazenado, sem se preocupar com detalhes técnicos.

- Feito para ser entendido por usuários e analistas.
- Mostra **entidades** (ex: Produto, Cliente), **atributos** (ex: nome, preço) e **relacionamentos** (ex: Cliente compra Produto).
- Usa ferramentas como o diagrama entidade-relacionamento (DER).

modelo conceitual
=

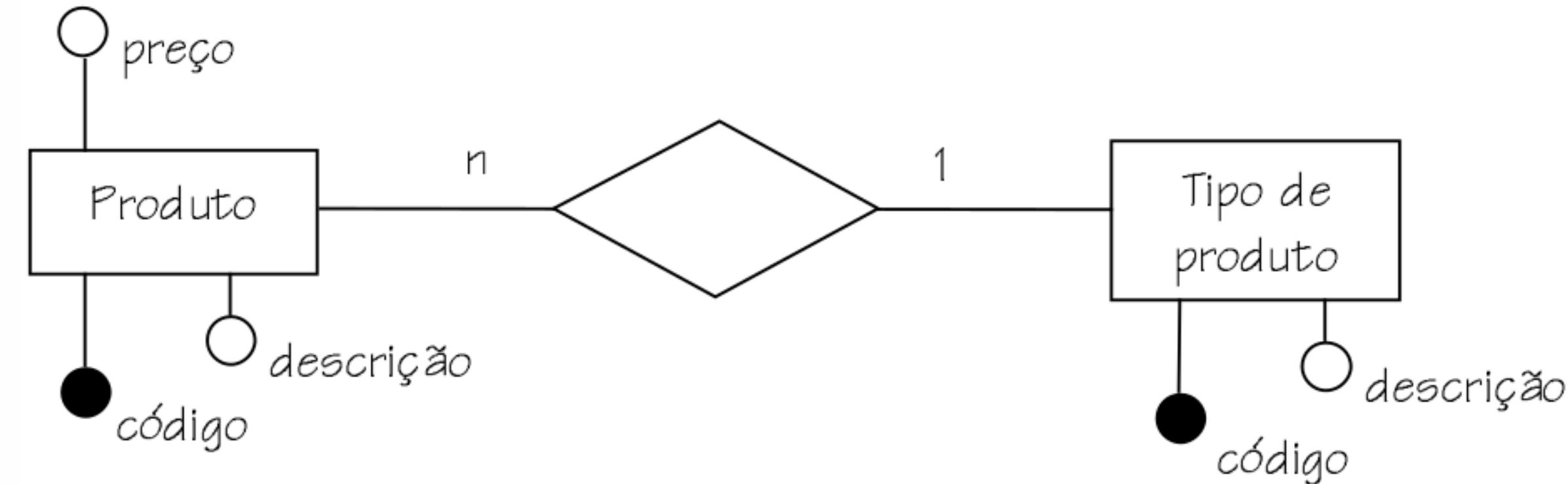
modelo de dados abstrato, que descreve a estrutura de um banco de dados de forma independente de um SGBD particular



Modelo Conceitual

A técnica mais difundida de modelagem conceitual é a abordagem **entidade-relacionamento** (ER). Nesta técnica, um modelo conceitual é usualmente

- representado através de um diagrama, chamado diagrama entidade-relacionamento (DER).



Modelo Lógico

É uma visão mais técnica, feita a partir do modelo conceitual, que descreve como os dados serão organizados.

- Indica **tabelas, colunas, tipos de dados** e **relacionamentos** em forma de chaves primárias e estrangeiras.
- Já está mais próximo de como o banco será implementado em um SGBD (Sistema de Gerenciamento de Banco de Dados), como MySQL ou PostgreSQL.



CURSO TÉCNICO EM INFORMÁTICA

Modelo Lógico

TipoDeProduto(CodTipoProd,DescrTipoProd)

Produto(CodProd,DescrProd,PrecoProd,CodTipoProd)

CodTipoProd referencia TipoDeProduto

TipoDeProduto

CodTipoProd	DescrTipoProd
1	Computador
2	Impressora

Produto

CodProd	DescrProd	PrecoProd	CodTipoProd
1	PC desktop modelo X	2.500	1
2	PC notebook ABC	3.500	1
3	Impressora jato de tinta	600	2
4	Impressora laser	800	2



Modelo Lógico

O **modelo lógico** descreve a estrutura do banco de dados do ponto de vista do usuário do **SGBD**, sem incluir detalhes de armazenamento interno.

Esses detalhes pertencem ao **modelo físico**, que é usado por especialistas para otimizar o desempenho do banco. O modelo físico varia entre produtos e, em sistemas mais modernos, essa parte é gerenciada automaticamente pelo SGBD.

modelo lógico

=

modelo de dados que representa a estrutura
de dados de um banco de dados conforme
vista pelo usuário do SGBD



Projeto de Banco de Dados

O projeto de um novo banco de dados ocorre em duas fases principais:

1. **Modelagem conceitual**: cria-se um diagrama entidade-relacionamento para representar as necessidades da organização, sem se preocupar com detalhes técnicos.
2. **Projeto lógico**: transforma-se o modelo conceitual em um modelo lógico, que já considera como o banco será implementado em um SGBD específico.



CURSO TÉCNICO EM INFORMÁTICA

ABORDAGEM ENTIDADE RELACIONAMENTO

Uma entidade representa um objeto ou conceito do mundo real que possui significado dentro do sistema e sobre o qual se deseja armazenar informações.

Exemplos de entidades:

- Pessoa
- Produto
- Aluno
- Livro

Cada entidade possui atributos, que são as informações que descrevem essa entidade. Por exemplo, a entidade Aluno pode ter os atributos nome, matrícula, idade.

entidade
=

conjunto de objetos¹ da realidade modelada
sobre os quais deseja-se manter informações
no banco de dados



CURSO TÉCNICO EM INFORMÁTICA

ABORDAGEM ENTIDADE RELACIONAMENTO

Em um DER, uma entidade é representada por meio de um retângulo que contém o nome da entidade.
Alguns exemplos são mostrados na Figura

PESSOA

DEPARTAMENTO

Representação gráfica de entidades



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO

relacionamento

=

conjunto de associações entre entidades

Em um DER, um **relacionamento** é representado através de um losango, ligado por linhas aos retângulos representativos das entidades que participam do relacionamento.



RELACIONAMENTO



- **Pessoas:** representa um conjunto de dados sobre indivíduos (relacionamento chamado PESSOA).
- **Departamentos:** armazena informações sobre os setores ou áreas da instituição (relacionamento chamado DEPARTAMENTO).
- **Lotação:** mostra a ligação entre pessoas e departamentos, ou seja, indica em qual departamento cada pessoa está alocada (relacionamento chamado LOTAÇÃO).



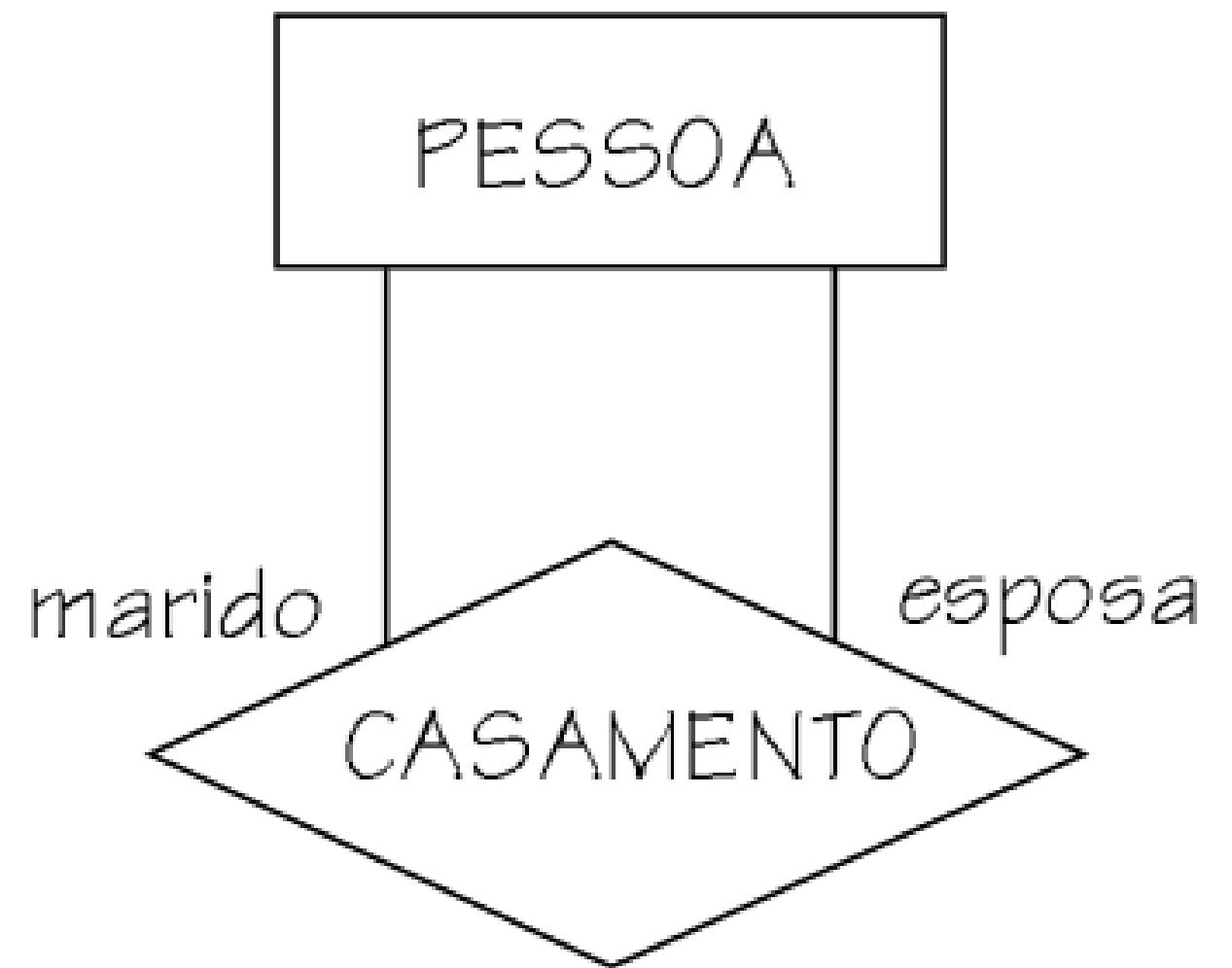
CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO

Não necessariamente um relacionamento associa entidades diferentes. A Figura abaixo mostra um DER que contém um auto-relacionamento, isto é, um relacionamento entre ocorrências de uma mesma entidade.

Neste caso, é necessário um conceito adicional, o de **papel da entidade** no relacionamento. No caso do relacionamento de casamento, uma ocorrência de pessoa exerce o papel de **marido** e a outra ocorrência de pessoa exerce o papel de **esposa**.

Papéis são anotados no **DER** como mostrado na figura, o caso de relacionamentos entre entidades diferentes, como o de LOTAÇÃO mostrado anteriormente, não é necessário indicar os papéis das entidades, já que eles são óbvios.



CURSO TÉCNICO EM INFORMÁTICA

CARDINALIDADE

Para fins de **projeto de banco de dados**, uma propriedade importante de um relacionamento é a de quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência através do relacionamento. Esta propriedade é chamada de cardinalidade de uma entidade em um relacionamento.

Há duas cardinalidades a considerar: a **cardinalidade máxima** e a **cardinalidade mínima**.

cardinalidade (mínima, máxima) de entidade
em relacionamento
=

número (mínimo, máximo) de ocorrências
de entidade associadas a uma ocorrência da
entidade em questão através do
relacionamento



CURSO TÉCNICO EM INFORMÁTICA

CARDINALIDADE MÁXIMA



No relacionamento **LOTAÇÃO**, temos as seguintes regras:

- A entidade **EMPREGADO** tem cardinalidade máxima 1:
Isso quer dizer que cada empregado pode estar lotado em no máximo um departamento. Ou seja, um mesmo empregado não pode estar ligado a vários departamentos ao mesmo tempo.
- A entidade **DEPARTAMENTO** tem cardinalidade máxima 120:
Isso significa que cada departamento pode ter, no máximo, 120 empregados lotados nele. Ou seja, um departamento pode estar associado a até 120 empregados diferentes.



CURSO TÉCNICO EM INFORMÁTICA

CARDINALIDADE MÁXIMA



Para facilitar o entendimento e a prática, não é necessário diferenciar exatamente os números quando a cardinalidade máxima for maior que 1.

Por isso, usamos apenas duas representações:

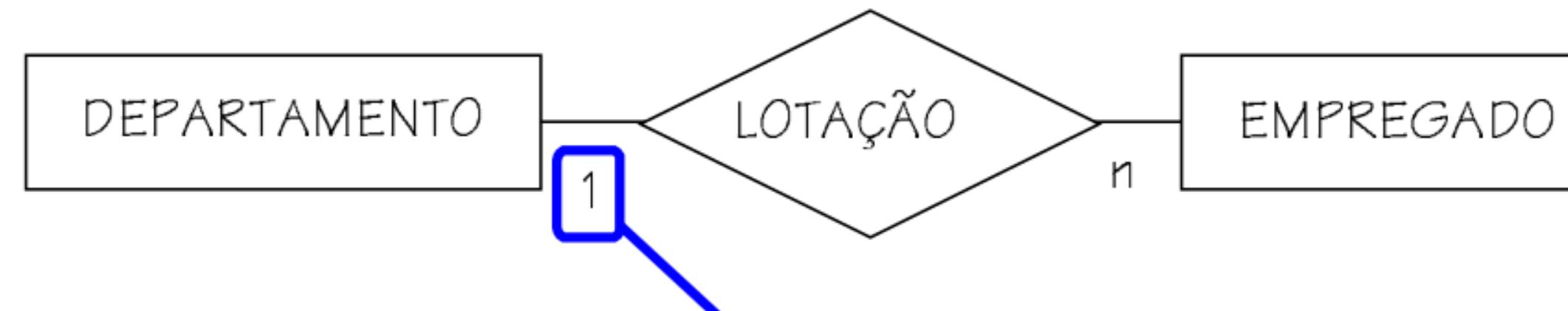
- **1**: quando uma entidade pode se relacionar com no máximo uma ocorrência da outra;
- **n**: quando uma entidade pode se relacionar com várias ocorrências da outra (independente do número exato).

No exemplo anterior, como um departamento pode estar ligado a até 120 empregados, consideramos essa cardinalidade como "n" — ou seja, "muitos".



CURSO TÉCNICO EM INFORMÁTICA

CARDINALIDADE MÁXIMA



expressa que a uma ocorrência de EMPREGADO
(entidade do lado oposto da anotação) pode estar associada
ao máximo uma (“1”) ocorrência de DEPARTAMENTO

expressa que a uma ocorrência de DEPARTAMENTO
(entidade ao lado oposto da anotação) podem estar associadas
muitas (“n”) ocorrências de EMPREGADO



CURSO TÉCNICO EM INFORMÁTICA

CARDINALIDADE MÁXIMA

A cardinalidade **máxima** é útil para classificar relacionamentos binários — ou seja, relacionamentos que envolvem duas entidades.

Esses relacionamentos podem ser classificados em três tipos principais:

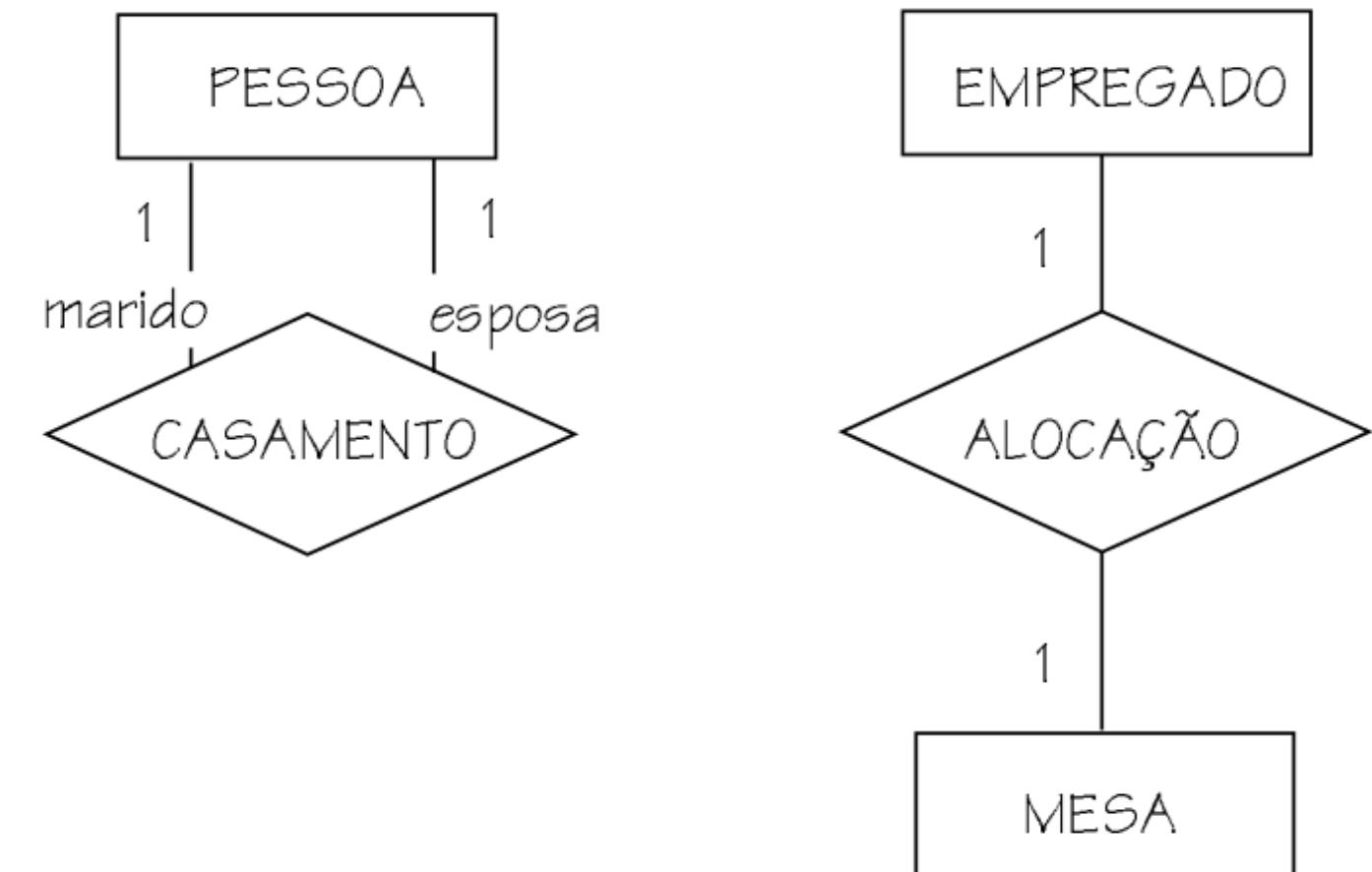
- **1:1** (um para um): cada ocorrência de uma entidade se relaciona com no máximo uma ocorrência da outra.
- **1:n** (um para muitos): uma ocorrência de uma entidade se relaciona com várias da outra, mas o contrário não acontece.
- **n:n** (muitos para muitos): várias ocorrências de uma entidade podem se relacionar com várias da outra.



RELACIONAMENTO 1:1

O relacionamento **Alocação**, mostrado na figura, representa a ligação entre Empregado de uma **mesa**.

- 1 Empregado → 1 mesa
- 1 mesa → 1 Empregado (no máximo)



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO 1:N

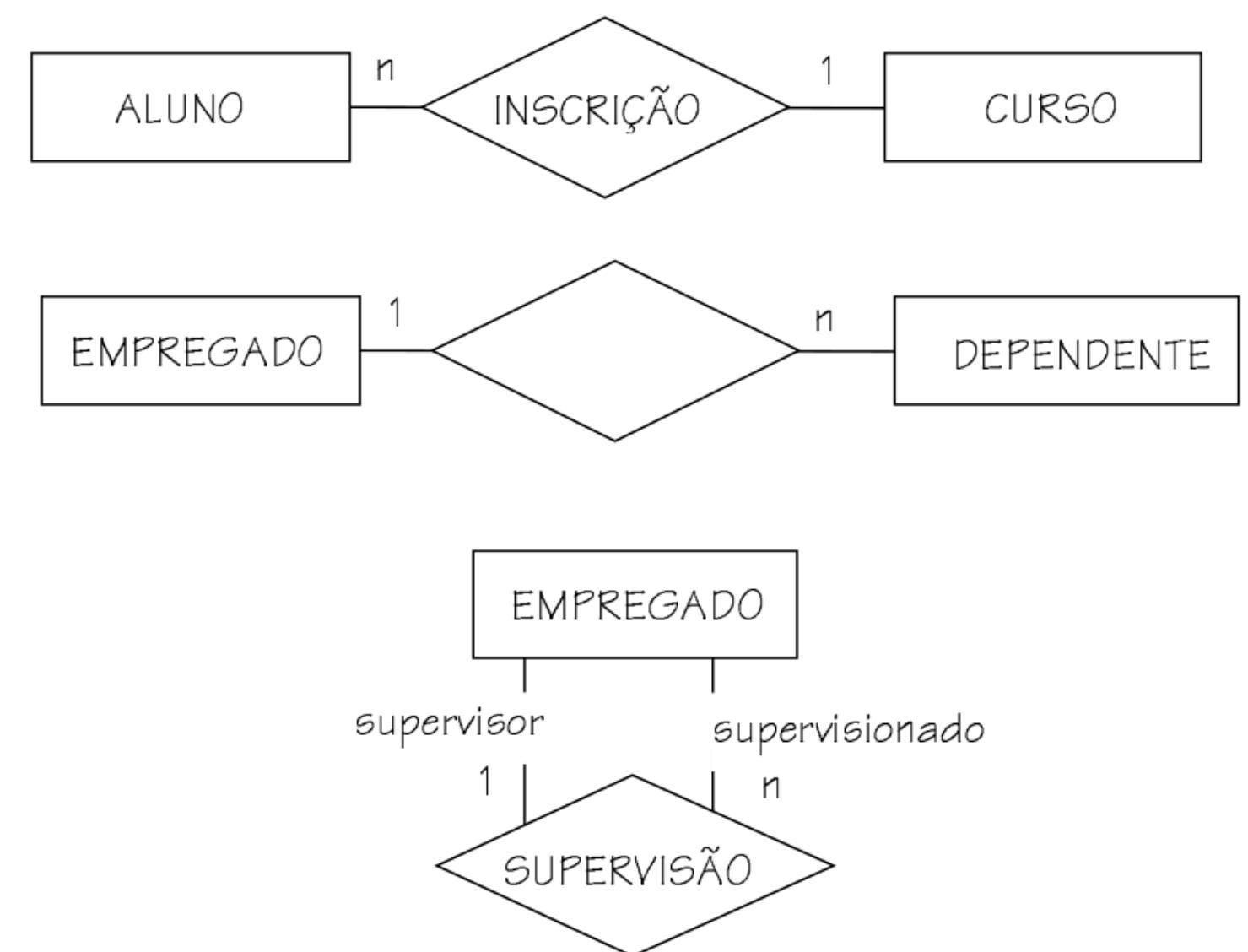
O relacionamento **INSCRIÇÃO**, mostrado na figura, representa a ligação entre cursos de uma universidade pública e seus alunos.

Como se trata de uma universidade pública, cada aluno pode estar inscrito em no máximo um curso.

Ou seja, um curso pode ter vários alunos, mas um aluno só pode estar associado a um único curso.

Esse tipo de relação é classificado como 1:n (um para muitos):

- 1 curso → n alunos
- 1 aluno → 1 curso (no máximo)

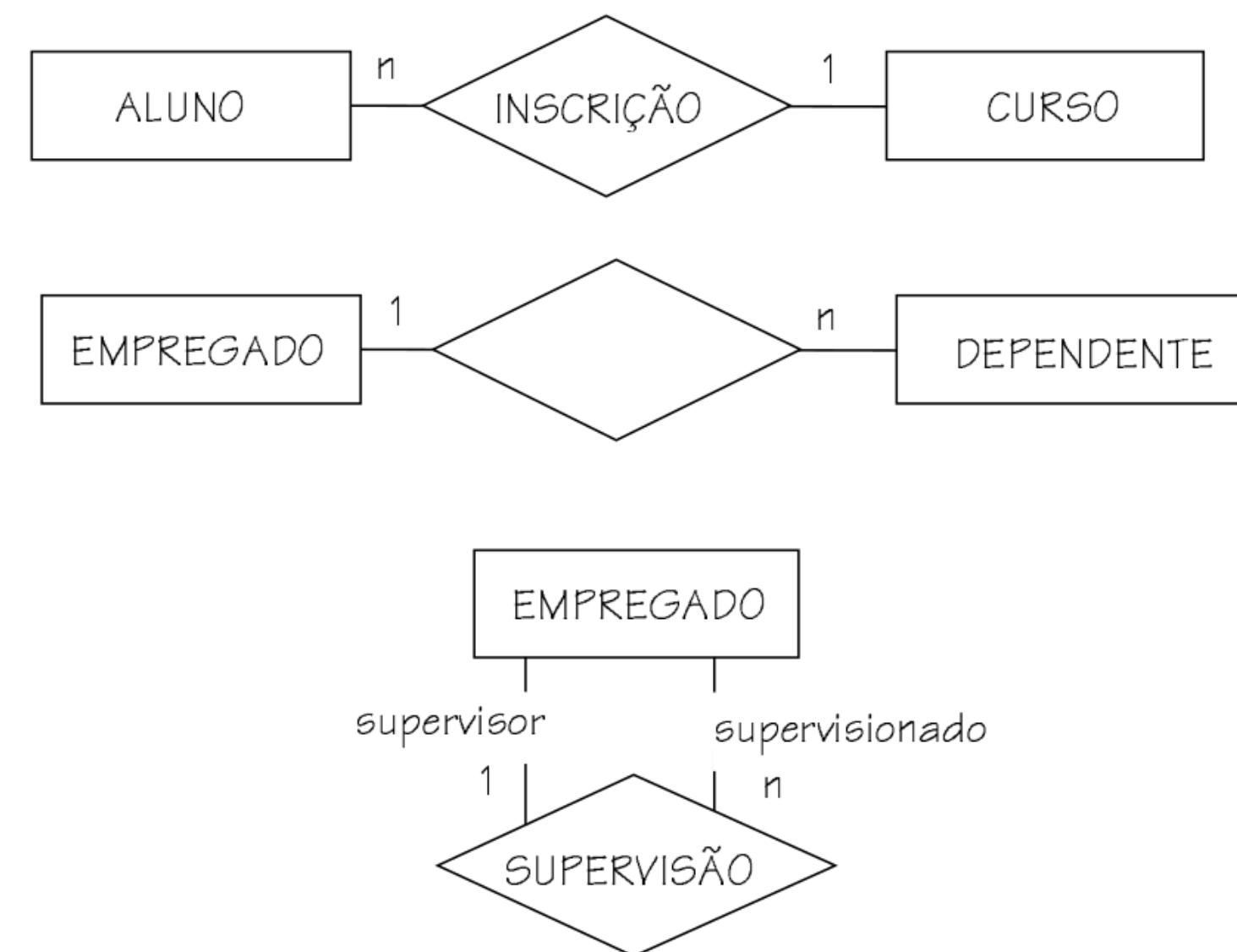


CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO 1:N

O relacionamento entre as entidades EMPREGADO e DEPENDENTE representa a ligação entre um empregado e seus dependentes, utilizada, por exemplo, para fins de imposto de renda.

- Nesse caso, cada dependente pode estar vinculado a no máximo um empregado.
- No **DER (Diagrama Entidade-Relacionamento)**, não foi indicado um nome para o relacionamento.
- Quando isso acontece, o relacionamento recebe como nome a junção dos nomes das entidades envolvidas.
- Assim, neste exemplo, o relacionamento é chamado de **EMPREGADO-DEPENDENTE**.

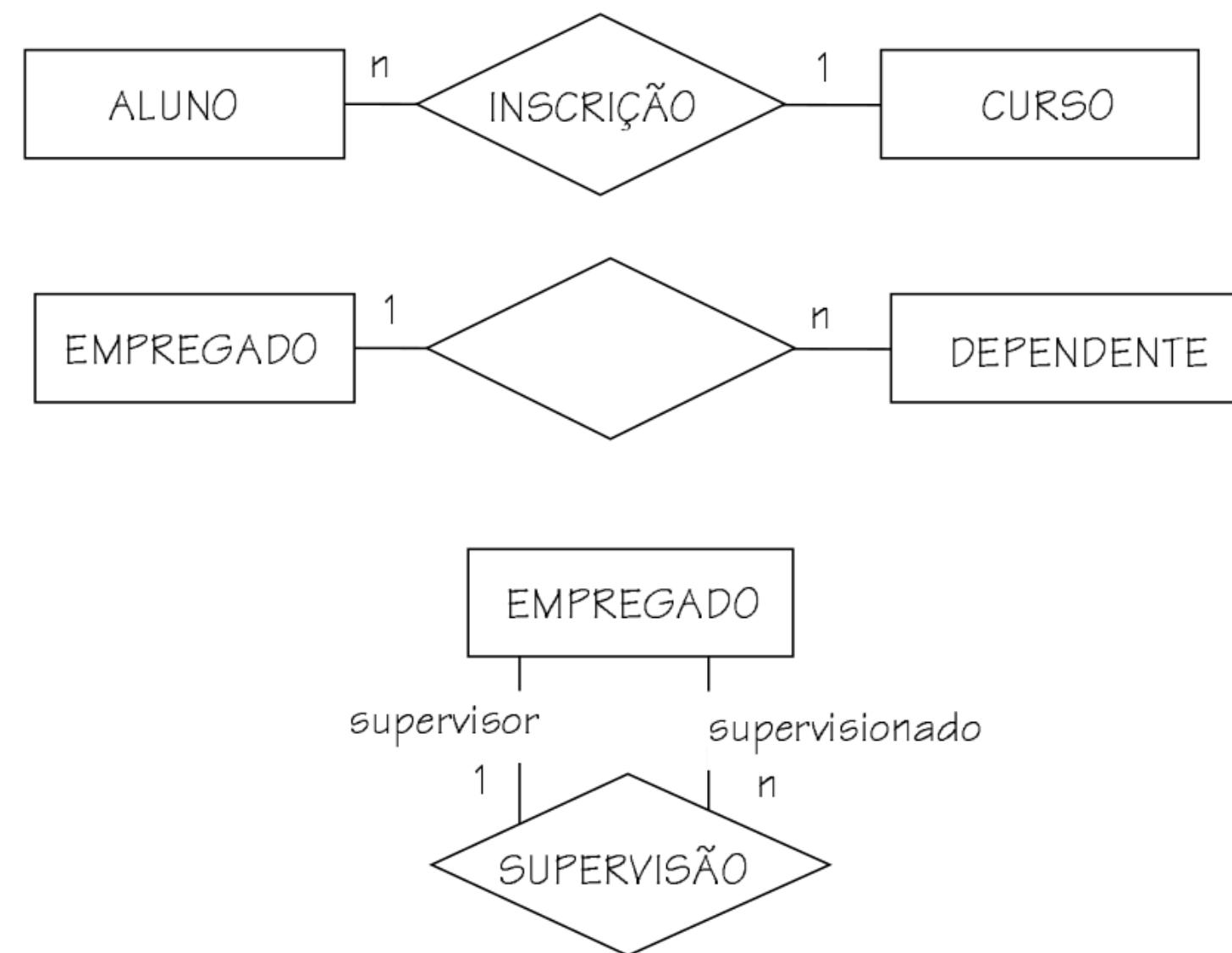


CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO 1:N

O relacionamento SUPERVISÃO é um exemplo de auto-relacionamento 1:n, ou seja, um relacionamento que liga a entidade EMPREGADO a ela mesma.

- Ele representa a relação entre um supervisor e seus supervisionados imediatos.
- A cardinalidade mostra que:
 - Cada empregado pode ter no máximo um supervisor;
 - Mas um supervisor pode estar ligado a vários supervisionados.



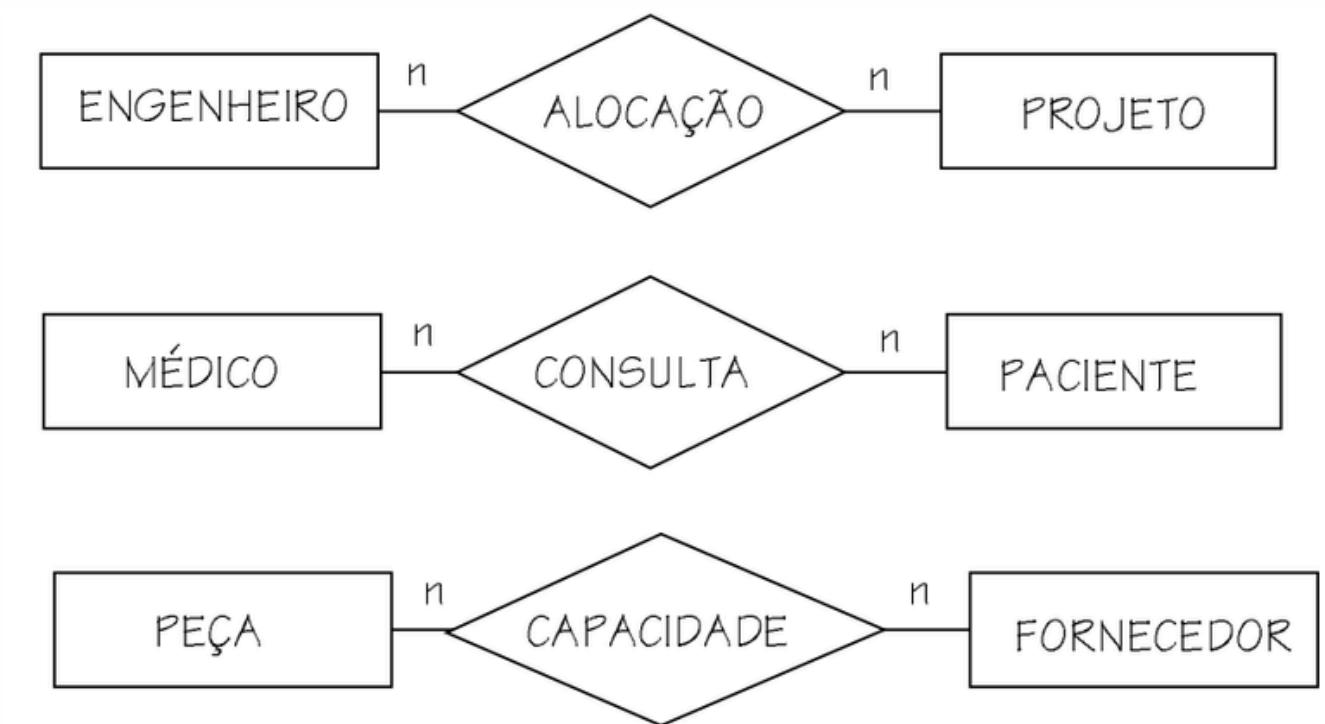
CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO N:N

Um relacionamento com cardinalidade **n:n** indica que muitas ocorrências de uma entidade podem estar associadas a múltiplas ocorrências da outra entidade.

Ou seja:

- Uma entidade A pode estar ligada a várias entidades B,
- E cada entidade B também pode estar ligada a várias entidades A.



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO N:N

ENGENHEIRO – ALOCAÇÃO – PROJETO

Cada engenheiro pode estar alocado em vários projetos, e um projeto pode ter vários engenheiros trabalhando nele.

◆ MÉDICO – CONSULTA – PACIENTE

Um médico pode atender vários pacientes, e um paciente pode consultar com vários médicos diferentes.

◆ PEÇA – CAPACIDADE – FORNECEDOR

Uma peça pode ser fornecida por diversos fornecedores, e um fornecedor pode fornecer várias peças diferentes.



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO N:N

ENGENHEIRO – ALOCAÇÃO – PROJETO

Cada engenheiro pode estar alocado em vários projetos, e um projeto pode ter vários engenheiros trabalhando nele.

◆ MÉDICO – CONSULTA – PACIENTE

Um médico pode atender vários pacientes, e um paciente pode consultar com vários médicos diferentes.

◆ PEÇA – CAPACIDADE – FORNECEDOR

Uma peça pode ser fornecida por diversos fornecedores, e um fornecedor pode fornecer várias peças diferentes.



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO N:N

Relacionamento COMPOSIÇÃO (Autorelacionamento n:n)

A imagem mostra que a entidade PRODUTO está se relacionando com ela mesma por meio do relacionamento COMPOSIÇÃO.

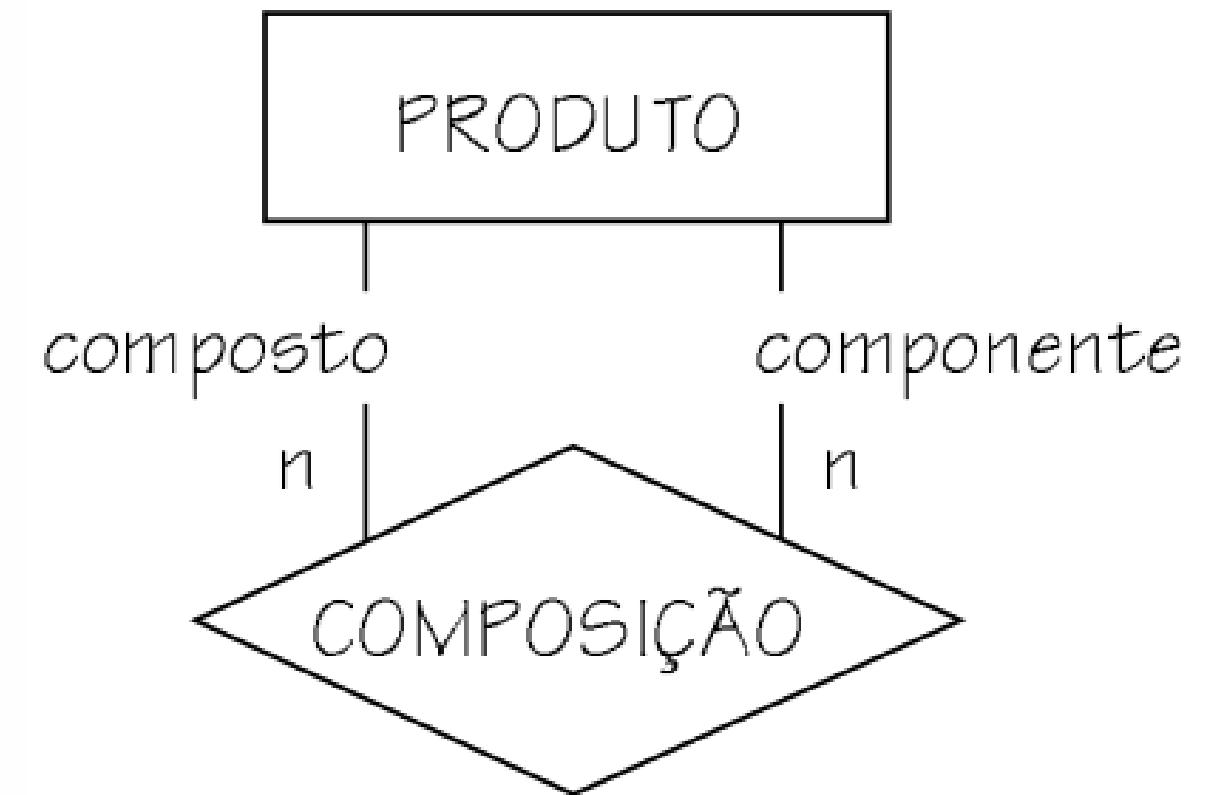
Esse é um caso de autorelacionamento, pois um produto está associado a outros produtos da mesma entidade.

Como interpretar:

Um produto composto pode ser formado por vários produtos componentes.

E um produto componente pode fazer parte da composição de vários produtos compostos.

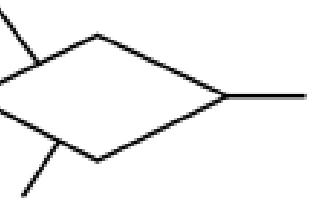
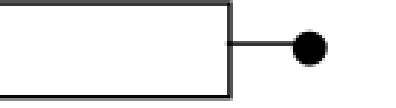
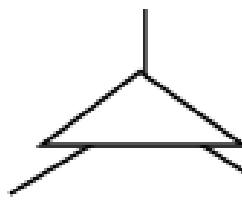
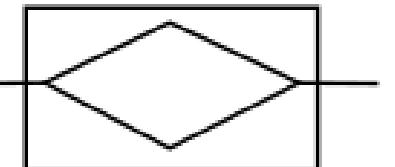
Ou seja, temos uma relação muitos para muitos (n:n) entre produtos



CURSO TÉCNICO EM INFORMÁTICA

Em banco de dados, a descrição de um modelo é chamada de esquema.

Tradicionalmente, os esquemas **ER** (Entidade-Relacionamento) são representados por diagramas gráficos, utilizando símbolos específicos (como mostrado na Figura 2.33).

Conceito	Símbolo
Entidade	
Relacionamento	
Atributo	
Atributo identificador	 
Relacionamento identificador	
Generalização/ especialização	
Entidade associativa	



CURSO TÉCNICO EM INFORMÁTICA

EXERCÍCIOS

Contexto: Cada pessoa pode ter no máximo um passaporte, e cada passaporte está vinculado a uma única pessoa.

Tarefa:

- Crie as entidades: PESSOA e PASSAPORTE
- Crie o relacionamento POSSE com cardinalidade 1:1
- Adicione atributos como nome em PESSOA e número em PASSAPORTE



CURSO TÉCNICO EM INFORMÁTICA

EXERCÍCIOS

Contexto: Um professor pode lecionar várias disciplinas, mas cada disciplina é ministrada por apenas um professor.

Tarefa:

- Crie as entidades: PROFESSOR e DISCIPLINA
- Crie o relacionamento LECIONA com cardinalidade 1:n
- Adicione atributos como nome em ambas as entidades



CURSO TÉCNICO EM INFORMÁTICA

EXERCÍCIOS

Contexto: Um aluno pode se matricular em várias disciplinas, e cada disciplina pode ter vários alunos.

Tarefa:

- Crie as entidades: ALUNO e DISCIPLINA
- Crie o relacionamento MATRÍCULA com cardinalidade n:n
- Adicione atributos como nota no relacionamento



CURSO TÉCNICO EM INFORMÁTICA

EXERCÍCIOS

Contexto: Em uma empresa, um empregado pode supervisionar outros empregados. Cada empregado tem no máximo um supervisor, mas um supervisor pode supervisionar vários.

Tarefa:

- Crie a entidade: EMPREGADO
- Crie o relacionamento SUPERVISÃO (auto-relacionamento) com cardinalidade 1:n
- Adicione o atributo nome em EMPREGADO



CURSO TÉCNICO EM INFORMÁTICA

EXERCÍCIOS

Contexto: Um produto pode ser formado por vários componentes, que também são produtos.

Tarefa:

- Crie a entidade: PRODUTO
- Crie o relacionamento COMPOSIÇÃO (auto-relacionamento) com cardinalidade n:n
- Use os papéis composto e componente
- Adicione atributos como nome e código em PRODUTO



CURSO TÉCNICO EM INFORMÁTICA

EXERCÍCIOS

Contexto: Um produto pode ser formado por vários componentes, que também são produtos.

Tarefa:

- Crie a entidade: PRODUTO
- Crie o relacionamento COMPOSIÇÃO (auto-relacionamento) com cardinalidade n:n
- Use os papéis composto e componente
- Adicione atributos como nome e código em PRODUTO



CURSO TÉCNICO EM INFORMÁTICA

COMPOSIÇÃO DE UM BANCO DE DADOS RELACIONAL

Um **banco de dados relacional** é formado por **tabelas**, que também podem ser chamadas de **relações**.

O termo **tabela** é o mais usado no dia a dia, especialmente em softwares e sistemas comerciais.

Já o termo **relação** vem dos estudos acadêmicos e da teoria que deu origem aos bancos relacionais — por isso o nome "relacional".



CURSO TÉCNICO EM INFORMÁTICA

COMPOSIÇÃO DE UM BANCO DE DADOS RELACIONAL

Tabela

Uma tabela é formada por um conjunto de linhas, que não possuem uma ordem específica.

Na linguagem acadêmica, essas **linhas** são chamadas de **tuplas**.

Cada **linha** contém vários campos, ou seja, os valores que representam as informações.

Na terminologia acadêmica, esses campos são chamados de **valores de atributo**.

Cada campo é identificado por um **nome**, que chamamos de nome do campo — ou nome do atributo, no meio acadêmico.

Os campos com o mesmo nome, que se repetem em todas as linhas da tabela, formam o que chamamos de **coluna**.



CURSO TÉCNICO EM INFORMÁTICA

COMPOSIÇÃO DE UM BANCO DE DADOS RELACIONAL

Tabela

Emp		coluna (atributo)	nome do campo (nome do atributo)	
	CódigoEmp	Nome	CodigoDept	CategFuncional
	E5	Souza	D1	C5
	E3	Santos	D2	C5
	E2	Silva	D1	C2
	E1	Soares	D1	—

Diagrama explicativo da estrutura de uma tabela relacional:

- Coluna (atributo):** Indicada por um retângulo rosa que engloba as colunas "Nome" e "CodigoDept".
- nome do campo (nome do atributo):** Indicado por um círculo azul ao redor da coluna "CategFuncional".
- linha (tupla):** Indicada por um elipse rosa que engloba as linhas de dados.
- valor de campo (valor de atributo):** Indicado por um círculo azul ao redor do valor "C2" na célula da linha E2 e da coluna CategFuncional.



CURSO TÉCNICO EM INFORMÁTICA

COMPOSIÇÃO DE UM BANCO DE DADOS RELACIONAL

Tabela

Ao comparar uma tabela de um **banco de dados relacional** com um **arquivo tradicional** armazenado no sistema de arquivos de um computador, podemos perceber algumas diferenças importantes:

- **Ordem das linhas:** Nas tabelas de um banco de dados, as linhas não seguem uma ordem específica. Quem controla a ordem de apresentação dos dados é o próprio **SGBD** (Sistema Gerenciador de Banco de Dados), e ela só será organizada se você pedir isso explicitamente na consulta (por exemplo, com ORDER BY). Não é possível acessar uma linha pelo número da posição (como “linha 1”, “linha 2”). Já nos arquivos tradicionais, o programador decide a ordem em que os dados são salvos, e é possível acessar registros pela posição em que foram gravados.



CURSO TÉCNICO EM INFORMÁTICA

COMPOSIÇÃO DE UM BANCO DE DADOS RELACIONAL

Formato dos campos

- **Nas tabelas**, cada campo armazena apenas um único valor atômico (indivisível). Em **arquivos tradicionais**, um campo pode ser composto (ou seja, conter subcampos) ou multi-valorado (como um vetor em Pascal ou um grupo repetido em COBOL).
- **Forma de buscar informações:** Em um **banco de dados relacional**, é possível fazer consultas flexíveis, usando qualquer campo (ou combinação de campos) para localizar os dados. Por exemplo, você pode buscar todos os registros onde a idade seja maior que 30 sem precisar saber a ordem de armazenamento. Nos arquivos tradicionais, para fazer buscas rápidas por valores de campos, é comum ter que criar estruturas extras, como índices manuais, ou programar o processo de busca com cuidado.



CURSO TÉCNICO EM INFORMÁTICA

COMPOSIÇÃO DE UM BANCO DE DADOS RELACIONAL

As linguagens de consulta usadas em **bancos de dados relacionais**, como o **SQL**, permitem acessar informações com base em qualquer critério, usando os campos de uma ou mais linhas da tabela.

Você pode, por exemplo, buscar todos os alunos com nota maior que 8 ou todos os pedidos feitos em determinada data, sem se preocupar com como os dados estão organizados fisicamente.

Já nos arquivos convencionais, fazer esse tipo de busca pode ser mais trabalhoso. Para encontrar rapidamente os registros com base em valores de campo, normalmente é necessário existir algum tipo de caminho de acesso.

Um caminho de acesso é uma estrutura auxiliar, como:

- um **índice** (semelhante ao índice de um livro, que aponta diretamente para a página certa),
- ou uma **cadeia de ponteiros** (ligações entre os registros que ajudam a chegar mais rápido ao dado desejado).



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS



Chaves

Para relacionar as linhas das tabelas em um banco de dados relacional, usamos um conceito fundamental: o de chave.

Em geral, há três tipos principais de chaves a considerar:

- **Chave primária**
- **Chave alternativa**
- **Chave estrangeira**



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

➔ CHAVE PRIMÁRIA

A **chave primária** é uma **coluna** ou um **conjunto de colunas** que identifica de forma única cada linha da tabela.

Por exemplo, na tabela Empregados, a coluna **CódigoEmp** pode ser usada como **chave primária**, pois cada empregado tem um código único.

Em alguns casos, duas colunas juntas são necessárias para identificar uma linha. Isso é chamado de **chave primária composta**.

Veja o exemplo da tabela Dependente, que usa as colunas **CódigoEmp** e **NoDepen** como chave primária.

Isso acontece porque:

- O mesmo código de empregado pode aparecer em várias linhas (um empregado pode ter mais de um dependente).
- O mesmo número de dependente (NoDepen, como 1, 2, 3...) pode aparecer em diferentes empregados.

Portanto, apenas a combinação das duas colunas é capaz de identificar de forma única um dependente.



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

🔑 CHAVE PRIMÁRIA

Dependente

CódigoEmp	NoDepen	Nome	Tipo	DataNasc
E1	01	João	Filho	12/12/91
E1	02	Maria	Esposa	01/01/50
E2	01	Ana	Esposa	05/11/55
E5	01	Paula	Esposa	04/07/60
E5	02	José	Filho	03/02/85

Tabela com chave primária composta



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

⚠ MINIMALIDADE DA CHAVE

Uma **chave primária** deve ser mínima, ou seja, deve conter apenas as colunas necessárias para garantir que não haja duplicidade.

Por exemplo, se alguém sugerir que a chave primária seja formada por **CódigoEmp**, **NoDepen** e **Tipo**, isso não está correto, pois **Tipo** é desnecessário para garantir a **unicidade**.

Retirando **Tipo**, a combinação **CódigoEmp + NoDepen** continua sendo suficiente, e por isso é essa a chave primária correta.



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

⟳ CHAVE ALTERNATIVA – O QUE É E COMO FUNCIONA

Em um banco de dados relacional, uma **chave alternativa** é qualquer outra coluna (ou conjunto de colunas) que também poderia ser usada para identificar unicamente uma linha em uma tabela, mas não foi escolhida como **chave primária**.

Ou seja, ela é uma “**alternativa**” válida à chave primária, mas o projetista optou por não usá-la como a principal.

Código Emp (PK)	CPF	Email	Nome
101	123.456.789-00	joao@email.com	João Silva
102	987.654.321-00	maria@email.com	Maria Souza
103	456.789.123-00	jose@email.com	José Pereira



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

⟳ CHAVE ALTERNATIVA – O QUE É E COMO FUNCIONA

CódigoEmp (PK)	CPF	Email	Nome
101	123.456.789-00	joao@email.com	João Silva
102	987.654.321-00	maria@email.com	Maria Souza
103	456.789.123-00	jose@email.com	José Pereira

- A **chave primária** dessa tabela é **CódigoEmp**, pois é o identificador interno do sistema.
- Porém, tanto o **CPF** quanto o **Email** também são valores únicos para cada funcionário.

Assim, podemos dizer que:

CPF e Email são **chaves alternativas** da tabela Empregado.

Eles poderiam ter sido usados como chave primária, mas o projetista escolheu CódigoEmp.



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

CHAVE ESTRANGEIRA – LIGANDO TABELAS

Uma **chave estrangeira** é uma coluna ou combinação de colunas de uma tabela cujos valores devem existir obrigatoriamente na **chave primária** de outra tabela (ou, em alguns casos, na própria tabela).
Ela é o mecanismo principal que permite criar relacionamentos entre tabelas em um banco de dados relacional.

Dept

CodigoDept	NomeDept
D1	Compras
D2	Engenharia
D3	Vendas

Emp

CodigoEmp	Nome	CodigoDept	CategFuncional	CIC
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



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

CHAVE ESTRANGEIRA – LIGANDO TABELAS

Regras de Integridade Referencial

A existência de uma chave estrangeira impõe regras que devem ser respeitadas sempre que o banco de dados for alterado:

1. Ao inserir dados na tabela que possui a chave estrangeira

→ **Exemplo:** ao cadastrar um novo empregado, o valor de **CodigoDept** precisa já existir na tabela **Dept**.

Ou seja, não é permitido atribuir um departamento que ainda não existe.

2. Ao alterar o valor da **chave estrangeira**

→ Se você modificar o departamento de um **empregado**, o novo valor também precisa existir na tabela **Dept**.

3. Ao excluir dados da tabela referenciada (a que tem a **chave primária**)

→ Um departamento não pode ser excluído se ainda houver empregados vinculados a ele.

Antes de excluir, você precisa remover ou transferir todos os empregados daquele departamento.



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

🌐 CHAVE ESTRANGEIRA – LIGANDO TABELAS

Em um **banco de dados relacional**, uma chave estrangeira (ou **foreign key**) é usada para criar uma ligação entre duas tabelas.

Ela representa uma referência à **chave primária** de outra tabela, garantindo a integridade dos dados entre as tabelas relacionadas.

Tabela Empregado		
CódigoEmp (PK)	Nome	Cargo
101	João Silva	Analista
102	Maria Souza	Gerente
103	José Pereira	Técnico

Tabela Dependente		
CódigoEmp (FK)	NoDepen	NomeDependente
101	1	Lucas Silva
101	2	Ana Silva
102	1	Pedro Souza

Na tabela **Dependente**, a coluna **CódigoEmp** é uma chave estrangeira que faz referência à chave primária da tabela **Empregado**.

Isso quer dizer que um dependente só pode existir se estiver vinculado a um empregado já cadastrado.



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

CHAVE ESTRANGEIRA – LIGANDO TABELAS

Regras garantidas pela chave estrangeira:

1. **Integridade referencial:**

- a. O valor da chave estrangeira deve existir como chave primária na tabela referenciada.
- b. **Exemplo:** não é permitido inserir um **dependente** com **CódigoEmp** = 999 se esse código não existir na tabela Empregado.

2. **Relacionamentos entre tabelas:**

- a. A **chave estrangeira** permite relacionar os dados de uma tabela com outra — por exemplo, para consultar todos os dependentes de um empregado específico.



CURSO TÉCNICO EM INFORMÁTICA

CHAVES EM BANCOS DE DADOS RELACIONAIS

🌐 CHAVE ESTRANGEIRA NA MESMA TABELA

Apesar do nome “**estrangeira**”, esse tipo de chave não precisa sempre fazer referência a outra tabela. Ela pode também referenciar a chave primária da própria tabela.

Exemplo: Tabela `Emp` com coluna de gerente

CodigoEmp (PK)	Nome	CodigoEmpGerente (FK)
101	João Silva	—
102	Maria Souza	101
103	Pedro Santos	102

Neste caso, a coluna **CodigoEmpGerente** é uma chave estrangeira que aponta para a própria tabela **Emp**.

Isso representa, por exemplo, que:

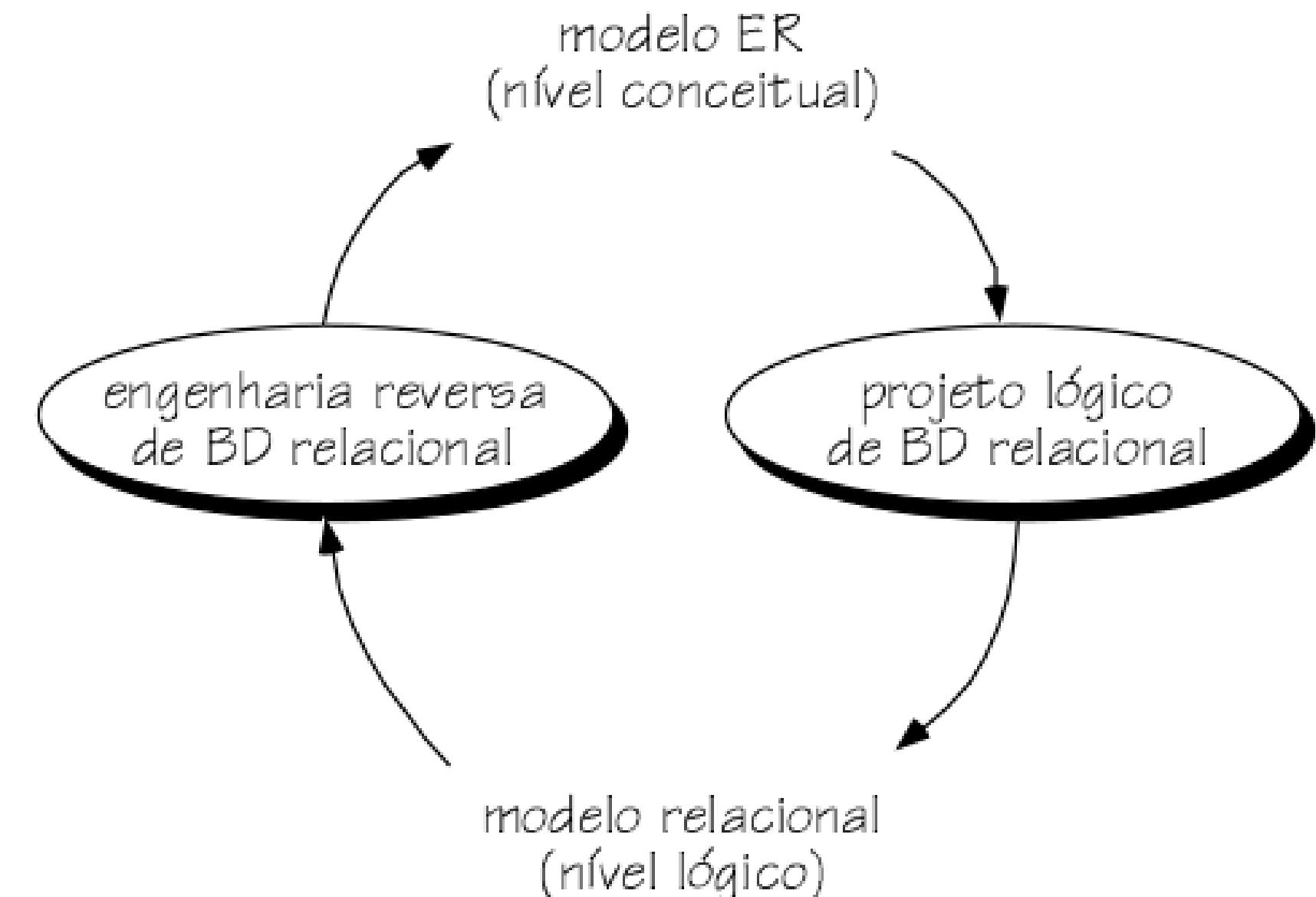
- João Silva não tem gerente (é o chefe).
- Maria Souza é gerenciada por João Silva.
- Pedro Santos é gerenciado por Maria Souza.

Assim, sempre que **CodigoEmpGerente** estiver preenchido, seu valor deve existir na coluna **CodigoEmp** da própria tabela.



CURSO TÉCNICO EM INFORMÁTICA

COMO TRANSFORMAR UM DER EM MODELO LÓGICO RELACIONAL



CURSO TÉCNICO EM INFORMÁTICA

DO DER AO MODELO LÓGICO (RELACIONAL)

O QUE É O MODELO LÓGICO

O modelo lógico é a representação do esquema de dados já no formato do paradigma escolhido (ex.: **relacional**), pronta para virar código SQL, mas sem detalhes de armazenamento físico. Ele fica entre o **modelo conceitual (DER)** e o **modelo físico**.

O que ele descreve

- **Estruturas:** tabelas (no relacional), colunas, tipos de dados.
- **Chaves:** PK (primária) e FK (estrangeira).
- **Restrições:** NOT NULL, UNIQUE, CHECK, regras de cardinalidade via FKS.
- **Regras de integridade:** domínios válidos, participação total/parcial (traduzidas em NOT NULL, UNIQUE, etc.).



CURSO TÉCNICO EM INFORMÁTICA

COMO TRANSFORMAR UM DER EM MODELO LÓGICO RELACIONAL

Entidades viram **Tabelas** (e escolher a chave certa)

- Crie uma **tabela** por **entidade** forte.
- **Atributos simples** → **colunas**.
- **Identificador do DER** → **PK**.
- Escolha a **PK**: **natural ou surrogate?**
- **Natural** (ex.: ra, cod_produto) quando for curta, estável e única.
- **Surrogate** (id inteiro/UUID) quando a natural for longa, instável ou externa (ex.: e-mail, CPF pode mudar por retificação, código vindo de outro sistema).
- Se usar surrogate, mantenha a natural com **UNIQUE**.



CURSO TÉCNICO EM INFORMÁTICA

COMO TRANSFORMAR UM DER EM MODELO LÓGICO RELACIONAL

Entidades viram **Tabelas** (e escolher a chave certa)

- Crie uma **tabela** por **entidade** forte.
- **Atributos simples** → **colunas**.
- **Identificador do DER** → **PK**.
- Escolha a **PK**: **natural ou surrogate?**
- **Natural** (ex.: ra, cod_produto) quando for curta, estável e única.
- **Surrogate** (id inteiro/UUID) quando a natural for longa, instável ou externa (ex.: e-mail, CPF pode mudar por retificação, código vindo de outro sistema).
- Se usar surrogate, mantenha a natural com **UNIQUE**.



CURSO TÉCNICO EM INFORMÁTICA

COMO TRANSFORMAR UM DER EM MODELO LÓGICO RELACIONAL



O que o DER mostra

- **Entidade:** PESSOA
- **Atributos:**
 - à direita: código (**círculo preto = identificador**), nome, endereço
 - à esquerda: data de admissão, data de nascimento
- Não há relacionamentos ou atributos multivalorados/compostos explícitos no desenho.



CURSO TÉCNICO EM INFORMÁTICA

COMO TRANSFORMAR UM DER EM MODELO LÓGICO RELACIONAL



Mapeamento para o modelo lógico (relacional)

PESSOA vira a tabela **Pessoa**.

Identificador → **PK (Primary Key)**

- O atributo código é o identificador no DER → vira PRIMARY KEY no lógico.

Atributos → **Colunas**

- nome, endereço, data_admissao, data_nascimento viram colunas.
- Defina tipos e regras coerentes (ex.: datas válidas, obrigatoriedade).



CURSO TÉCNICO EM INFORMÁTICA

NOMEAR COLUNAS AO PASSAR DO DER PARA O MODELO LÓGICO.

1) Mantenha nomes curtos e sem espaços Por quê? Serão digitados o tempo todo em consultas e código.

Como fazer? Abrevie atributos com várias palavras:

data de nascimento → **DataNasc**

data de admissão → **DataAdm**

2) Não repita o nome da tabela na coluna

- Por quê? Em SQL usamos a qualificação Tabela.Coluna (ex.: Pessoa.Nome).
- **Errado:** NomePessoa, NomePess
- **Certo:** Nome (e use Pessoa.Nome quando precisar qualificar)



CURSO TÉCNICO EM INFORMÁTICA

NOMEAR COLUNAS AO PASSAR DO DER PARA O MODELO LÓGICO.

3) Trate a chave primária como exceção

- Por quê? Ela aparece como **FK** em outras tabelas; incluir a sigla da tabela evita ambiguidade.
- Exemplo: **PK** de Pessoa → **CodigoPess**
 - Em outra tabela: ... **FOREIGN KEY (CodigoPess)** REFERENCES Pessoa(**CodigoPess**)

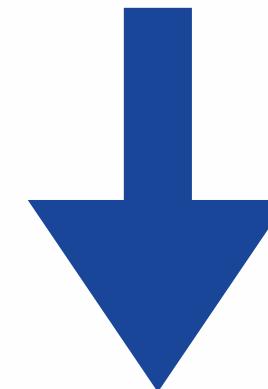
4) Padronize abreviações em todo o banco

- Por quê? Consistência facilita leitura e manutenção.
- Escolha e mantenha:
 - **Cod** para código → **CodigoPess** ou **CodProduto**
 - **Num** (ou **No**) para número → **NumCasa**, **NumPedido**



CURSO TÉCNICO EM INFORMÁTICA

COMO TRANSFORMAR UM DER EM MODELO LÓGICO RELACIONAL

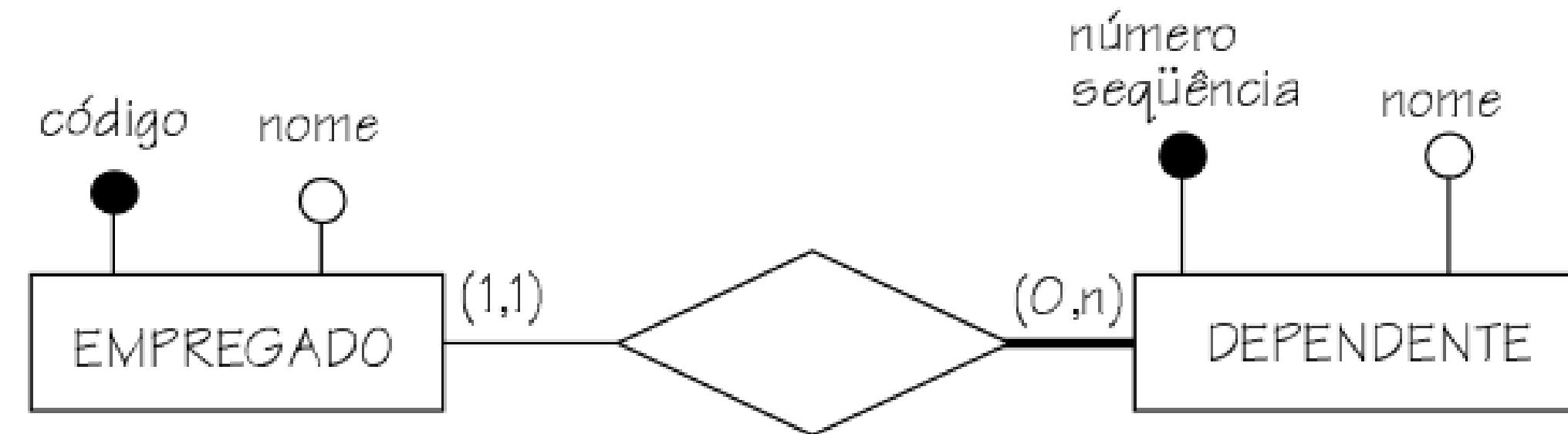


Pessoa (CodigoPess,Nome,Endereço,DataNasc,DataAdm)



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO IDENTIFICADOR



Em **DER**, relacionamento identificador é aquele em que a chave da entidade “**pai**” participa da identificação da entidade “**filha**”. Em outras palavras, a filha depende do pai para existir e usa a chave do pai como parte da sua PK.

Geralmente isso acontece com entidades **fracas**.

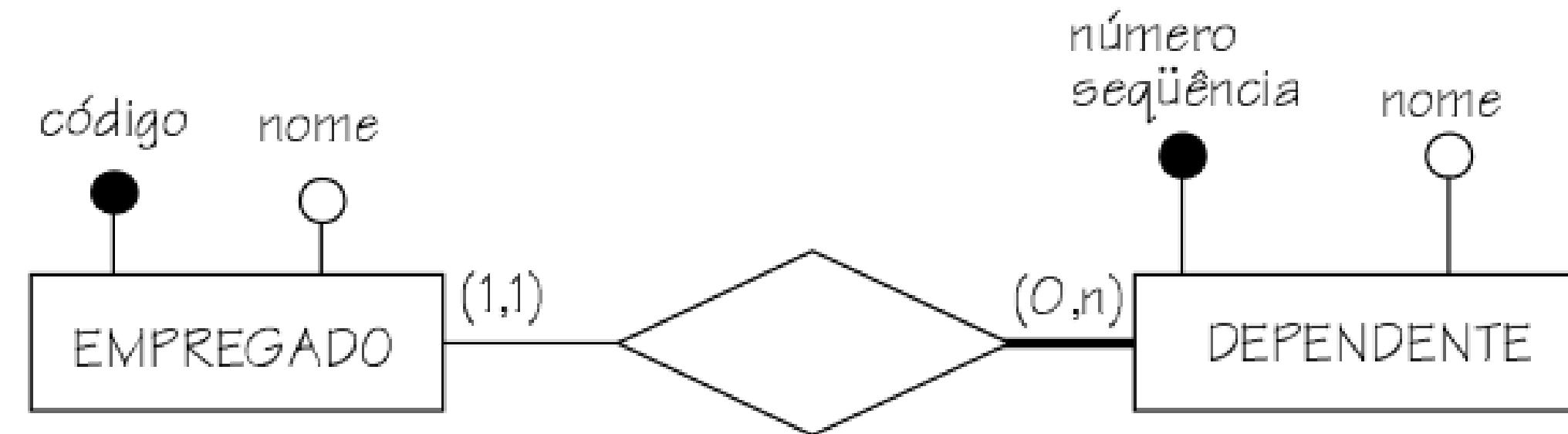
Regras práticas

- Participação total do lado da **filha** (ela não existe sem o pai).
- **PK** da filha = **FK** do pai + **identificador parcial da filha**.
- **Normalmente é 1:N** (um pai, muitos filhos).
- **FK** costuma usar **ON DELETE CASCADE** (faz sentido a filha sumir com o pai).



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO IDENTIFICADOR



Em **DER**, relacionamento identificador é aquele em que a chave da entidade “**pai**” participa da identificação da entidade “**filha**”. Em outras palavras, a filha depende do pai para existir e usa a chave do pai como parte da sua PK.

Geralmente isso acontece com entidades **fracas**.

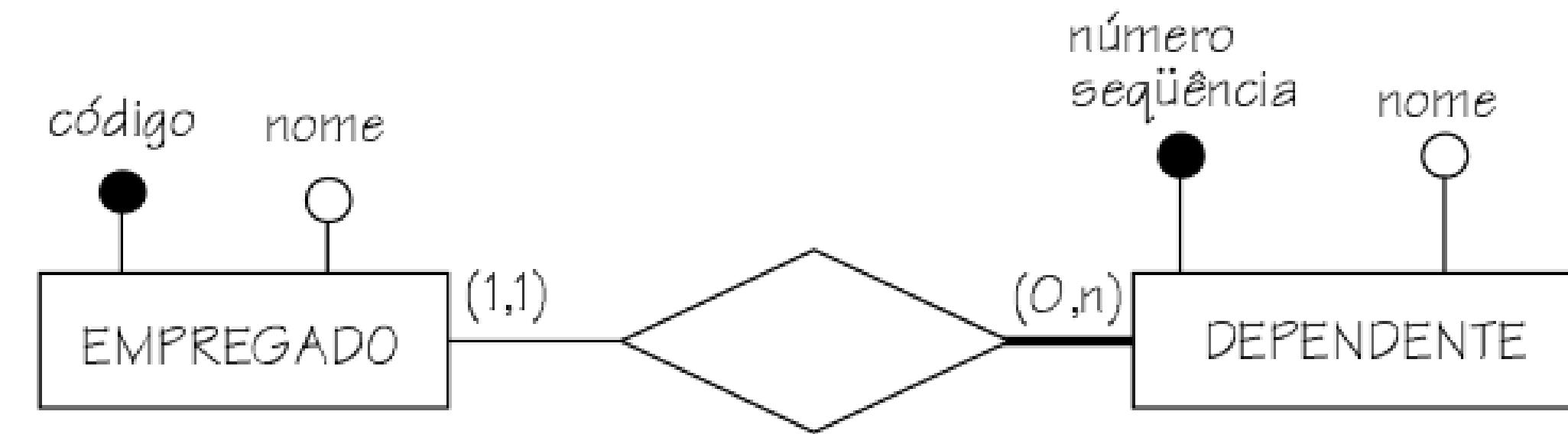
Regras práticas

- Participação total do lado da **filha** (ela não existe sem o pai).
- **PK** da filha = **FK** do pai + **identificador parcial da filha**.
- **Normalmente é 1:N** (um pai, muitos filhos).
- **FK** costuma usar **ON DELETE CASCADE** (faz sentido a filha sumir com o pai).



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO IDENTIFICADOR (1:N)



Modelo lógico

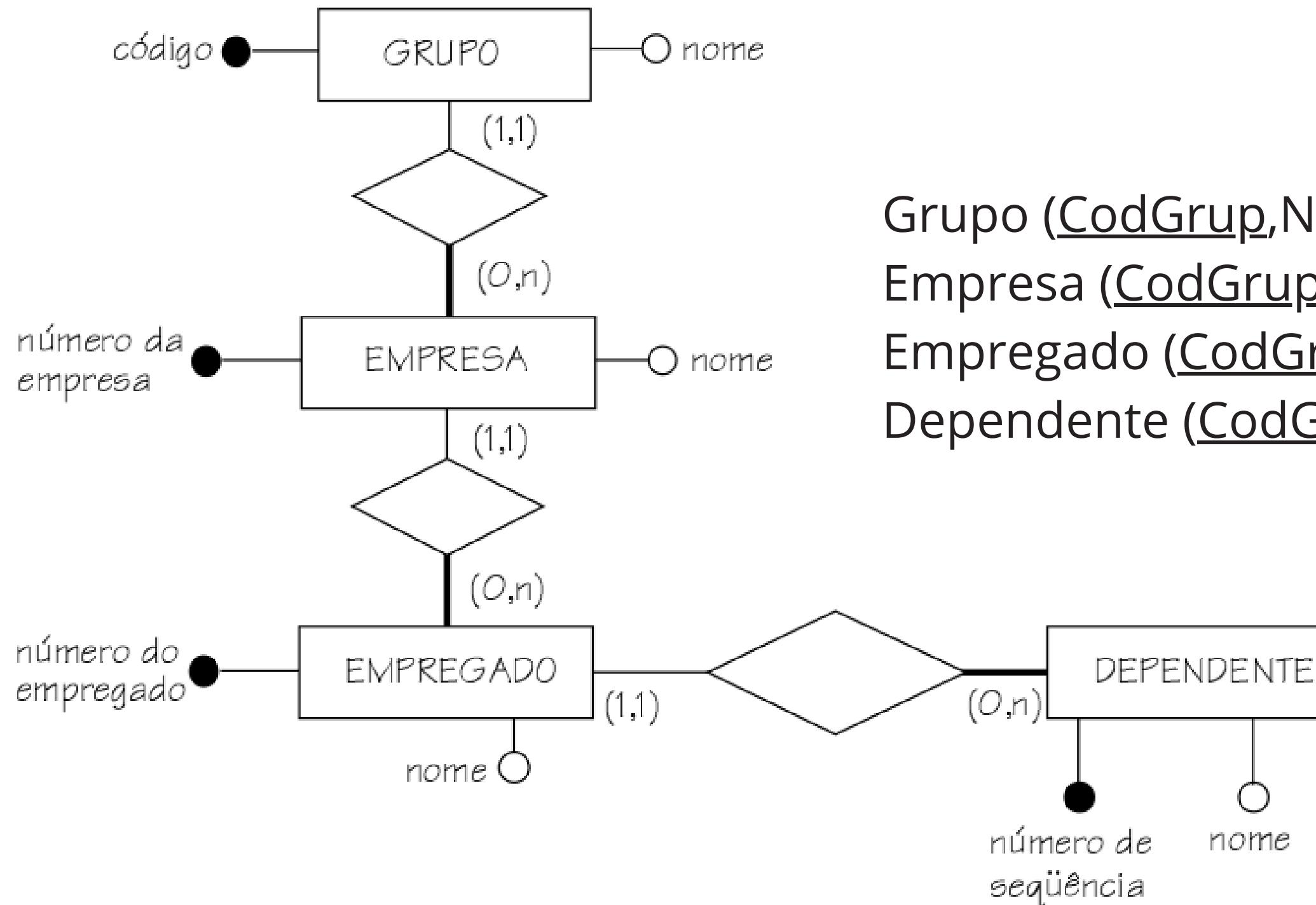
EMPREGADO(codigo_emp, nome)

DEPENDENTE(codigo_emp, num_seq, nome)



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO IDENTIFICADOR (1:N)



Grupo (CodGrup,Nome)

Empresa (CodGrup,NoEmpresa,Nome)

Empregado (CodGrup,NoEmpresa,NoEmpreg,Nome)

Dependente (CodGrup,NoEmpresa,NoEmpreg,NoSeq,Nome)



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO IDENTIFICADOR

Grupo (CodGrup,Nome)

Empresa (CodGrup,NoEmpresa,Nome)

Empregado (CodGrup,NoEmpresa,NoEmpreg,Nome)

Dependente (CodGrup,NoEmpresa,NoEmpreg,NoSeq,Nome)

No caso do exemplo, para compor a chave primária da tabela **Dependente**, é necessário, usar além do número de seqüência deste dependente, também o **identificador do empregado**. Entretanto, um empregado é identificado por seu número e pelo identificador da empresa a qual ele está vinculado.

Por sua vez, a empresa é identificada por um número e pelo identificador do grupo ao qual ela pertence.

Em outros termos, um dependente é identificado pela combinação das seguintes informações:

- código do grupo** da empresa à qual seu empregado está vinculado
- número da empresa** à qual seu empregado está vinculado
- número de seu empregado**
- seu número de seqüência.**

Essa linha de raciocínio nos leva à chave primária da tabela Dependente



CURSO TÉCNICO EM INFORMÁTICA

RELACIONAMENTO IDENTIFICADOR

Grupo (CodGrup,Nome)

Empresa (CodGrup,NoEmpresa,Nome)

Empregado (CodGrup,NoEmpresa,NoEmpreg,Nome)

Dependente (CodGrup,NoEmpresa,NoEmpreg,NoSeq,Nome)

No caso do exemplo, para compor a chave primária da tabela **Dependente**, é necessário, usar além do número de seqüência deste dependente, também o **identificador do empregado**. Entretanto, um empregado é identificado por seu número e pelo identificador da empresa a qual ele está vinculado.

Por sua vez, a empresa é identificada por um número e pelo identificador do grupo ao qual ela pertence.

Em outros termos, um dependente é identificado pela combinação das seguintes informações:

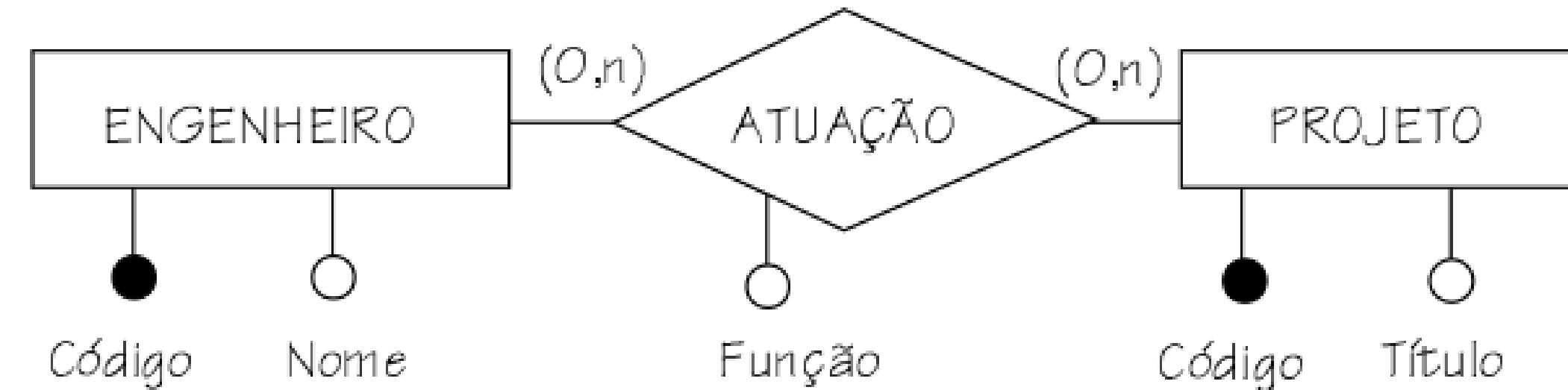
- código do grupo** da empresa à qual seu empregado está vinculado
- número da empresa** à qual seu empregado está vinculado
- número de seu empregado**
- seu número de seqüência.**

Essa linha de raciocínio nos leva à chave primária da tabela Dependente



CURSO TÉCNICO EM INFORMÁTICA

TABELA PRÓPRIA(N:N)



Quando traduzimos um relacionamento do **modelo conceitual** para o **modelo relacional**, criamos uma tabela específica para representá-lo.

Essa tabela vai conter:

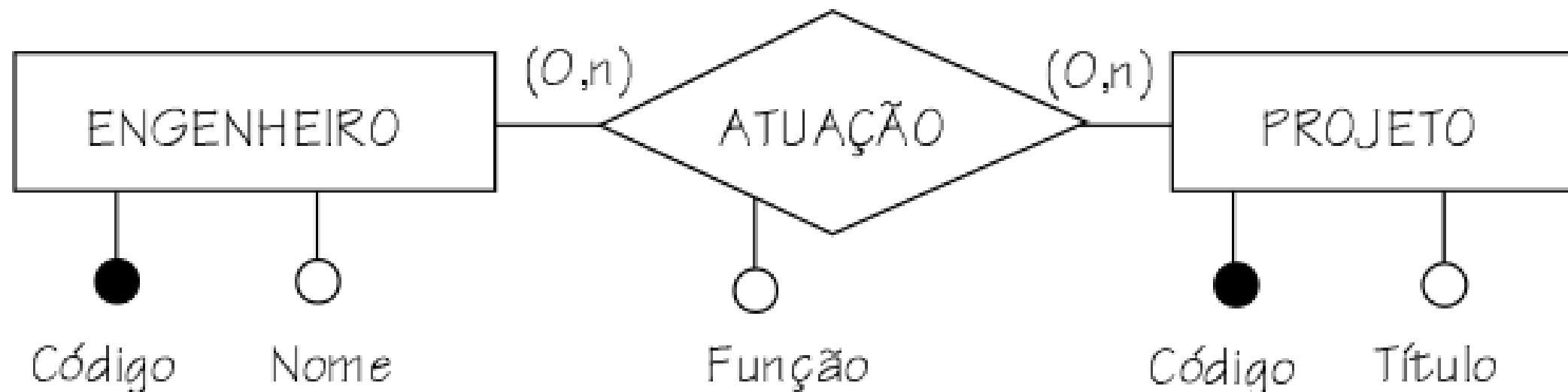
- **Colunas com os identificadores (chaves primárias)** das entidades que participam do relacionamento.
- **Colunas com os atributos próprios do relacionamento (caso existam).**

A **chave primária** dessa nova tabela será formada pela combinação das colunas que identificam as entidades relacionadas. Além disso, cada conjunto de colunas que representa o **identificador** de uma entidade será definido como chave estrangeira, apontando para a **tabela da entidade** correspondente.



CURSO TÉCNICO EM INFORMÁTICA

TABELA PRÓPRIA(N:N)



Engenheiro (CodEng,Nome)
Projeto (CodProj,Título)
Atuacao(CodEng,CodProj,Funcao)
CodEng referencia Engenheiro
CodProj referencia Projeto

A tabela **Atuação** implementa o relacionamento **ATUAÇÃO**. A chave primária da tabela é formada pelas colunas CodEng e CodProj, que correspondem aos identificadores das entidades relacionadas (ENGENHEIRO e PROJETO). Cada uma destas colunas é chaves estrangeira das tabelas que implementa a entidade relacionada. A coluna Função corresponde ao atributo do relacionamento.



CURSO TÉCNICO EM INFORMÁTICA

COMO UM RELACIONAMENTO ENTRE TABELAS PODE SER IMPLEMENTADO?

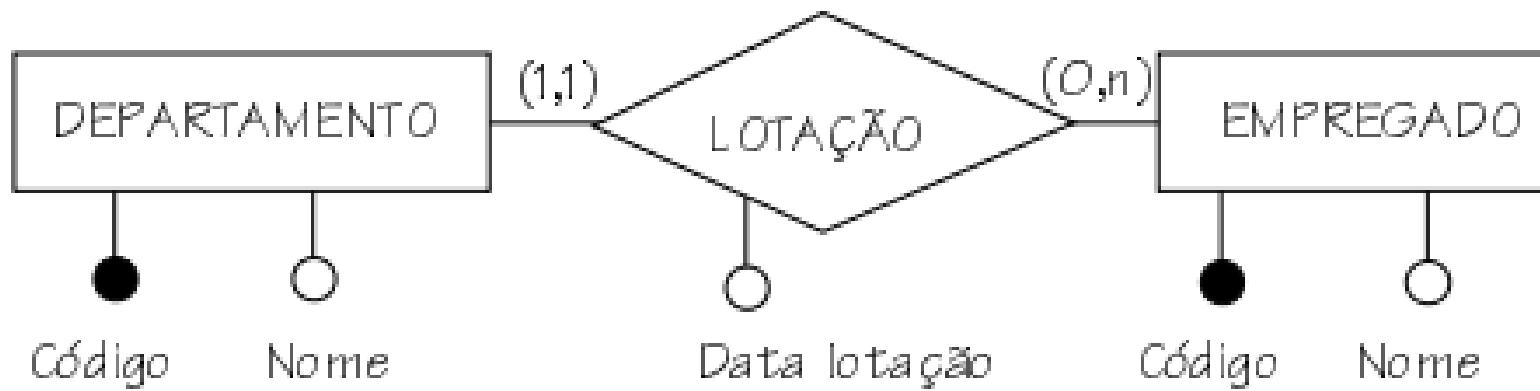
Uma forma de **conectar** duas tabelas é adicionando colunas a uma delas. Essa técnica só funciona se uma das entidades do relacionamento tem cardinalidade máxima de **1**.

O que isso significa?

Imagine a relação entre um **EMPREGADO** e um **DEPARTAMENTO**. Se cada empregado só pode trabalhar em um departamento (cardinalidade máxima de 1), podemos simplesmente adicionar informações do **departamento** à tabela de **empregados**.

Para a tabela **EMPREGADO**, você adicionaria duas novas colunas:

- 1. Código do Departamento (CodDept):** Esta coluna identifica a qual departamento o empregado pertence. Ela funciona como uma **chave estrangeira**, fazendo um link para a tabela de departamentos.
- 2. Dados do Relacionamento (DataLota):** Esta coluna guarda informações extras sobre a relação, como a data em que o empregado foi alocado.



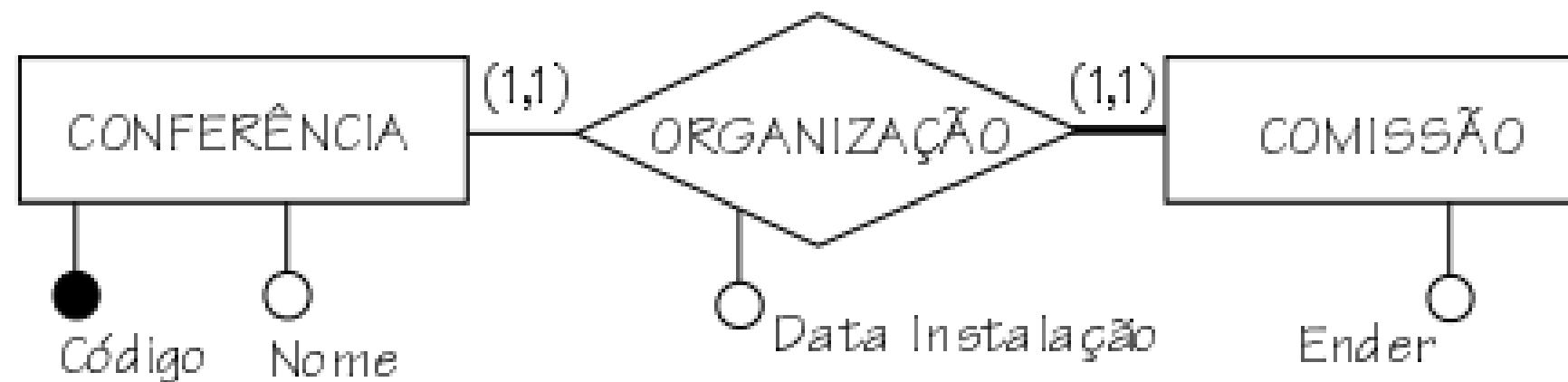
Esquema relacional correspondente:
Departamento (CodDept, Nome)
Empregado (CodEmp, Nome, CodDept, DataLota)
CodDept referencia Departamento



CURSO TÉCNICO EM INFORMÁTICA

COMO FUNDIR TABELAS PARA CRIAR UM RELACIONAMENTO 1:1?

Em um banco de dados, a melhor maneira de implementar um relacionamento 1 para 1 (como o da imagem, onde cada Conferência tem exatamente uma Comissão organizadora, e vice-versa) é fundindo **as tabelas** das duas entidades.



O que isso significa? Na imagem, o relacionamento entre uma **Conferência** e uma **Comissão** é **1:1**. Isso quer dizer que cada **conferência** tem apenas uma **comissão** organizadora, e cada comissão é responsável por apenas uma conferência.



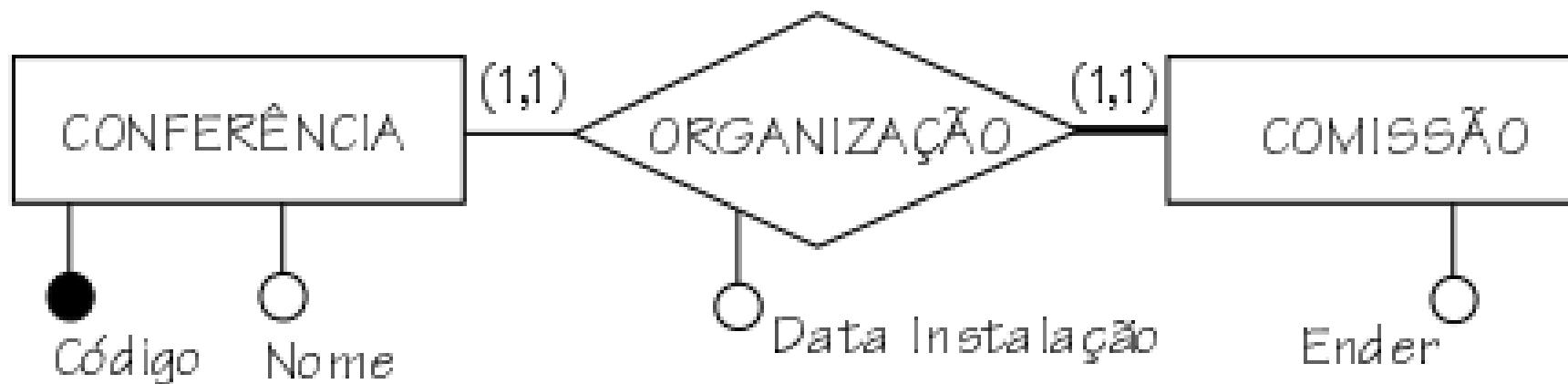
CURSO TÉCNICO EM INFORMÁTICA

COMO FUNDIR TABELAS PARA CRIAR UM RELACIONAMENTO 1:1?

Como fazer isso na prática?

Em vez de criar três tabelas separadas (CONFERÊNCIA, COMISSÃO e ORGANIZAÇÃO), você pode criar uma **única** tabela que combine todas as informações. Essa tabela unificada teria:

- Atributos da Entidade **CONFERÊNCIA**: Incluindo o **Código** (que é o identificador da entidade) e o **Nome**.
- Atributos da Entidade **COMISSÃO**: Incluindo o **Ender** (endereço da comissão).
- Atributos do Relacionamento **ORGANIZAÇÃO**: Incluindo a **Data_Instalação**.



Esquema relacional correspondente:

Conferência (CodConf,Nome,DataInstComOrg,EnderComOrg)



CURSO TÉCNICO EM INFORMÁTICA

COMO FUNDIR TABELAS PARA CRIAR UM RELACIONAMENTO 1:1?

Como escolher a melhor forma de traduzir um **relacionamento**?

A forma ideal de transformar um relacionamento em tabelas de banco de dados depende das cardinalidades (**mínima e máxima**) das entidades envolvidas.

A regra é simples: para cada tipo de relacionamento, existe uma forma de tradução preferida. Em alguns casos, há uma segunda alternativa, e em outros, uma que deve ser evitada a todo custo.

Resumo das regras:

- **Alternativa preferida (✓):** É a forma mais comum e recomendada.
- **Segunda alternativa (±):** Uma opção aceitável se a preferida não se encaixar bem.
- **Alternativa a ser evitada (✗):** Uma abordagem que pode causar problemas no banco de dados.

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
Relacionamentos 1:1			
(0,1) (0,1)	±	✓	✗
(0,1) (1,1)	✗	±	✓
(1,1) (1,1)	✗	±	✓
Relacionamentos 1:n			
(0,1) (0,n)	±	✓	✗
(0,1) (1,n)	±	✓	✗
(1,1) (0,n)	✗	✓	✗
(1,1) (1,n)	✗	✓	✗
Relacionamentos n:n			
(0,n) (0,n)	✓	✗	✗
(0,n) (1,n)	✓	✗	✗
(1,n) (1,n)	✓	✗	✗

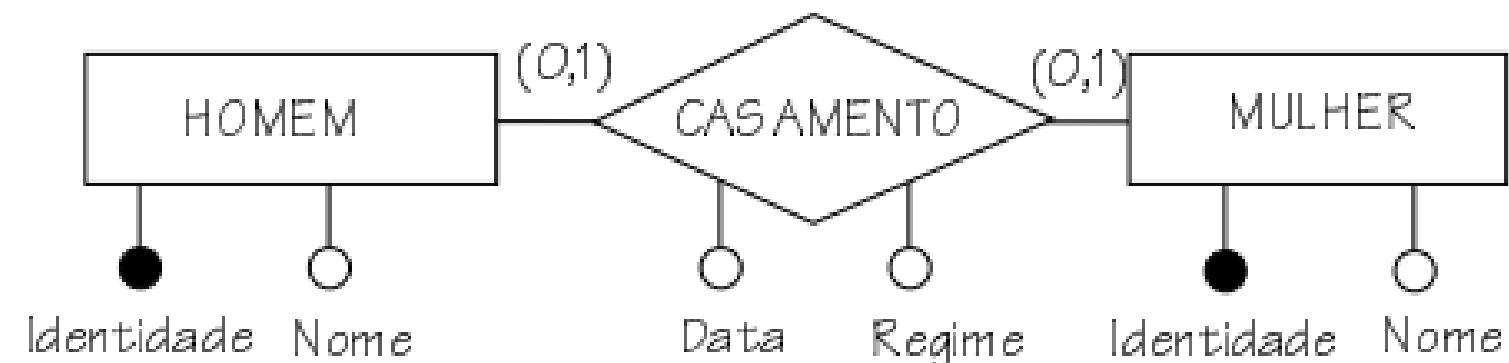
✓ Alternativa preferida ± Pode ser usada ✗ Não usar



CURSO TÉCNICO EM INFORMÁTICA

COMO TRADUZIR UM RELACIONAMENTO 1:1 COM PARTICIPAÇÃO OPCIONAL?

Quando um relacionamento é do tipo **1:1** e a participação de ambas as entidades é opcional (cardinalidade mínima **0**), a forma mais recomendada de traduzi-lo para o banco de dados é adicionar colunas em uma das tabelas.



O que isso significa? No exemplo, um HOMEM pode ou não ser casado, e uma MULHER pode ou não ser casada. A participação de ambos no relacionamento de **CASAMENTO** é opcional, pois nem todo homem ou mulher estará em um relacionamento de casamento.

Como fazer isso na prática?

Nesse caso, a regra é simples: você pode escolher qualquer uma das duas tabelas (HOMEM ou MULHER) para receber a chave estrangeira. Ambas as opções funcionam bem porque o relacionamento é simétrico (**1:1**).

- **Opção 1:** Adicionar colunas na tabela MULHER que apontem para a tabela HOMEM. Essas colunas seriam a identidade do homem (chave estrangeira), a data do casamento e o regime do casamento.
- **Opção 2:** Adicionar colunas na tabela HOMEM que apontem para a tabela MULHER. Essas colunas seriam a identidade da mulher (chave estrangeira), a data do casamento e o regime do casamento.



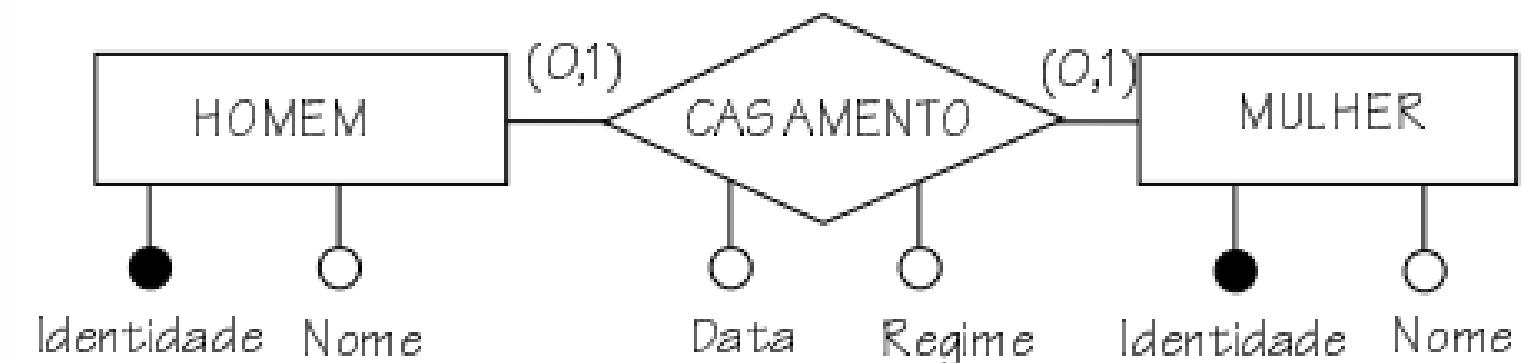
CURSO TÉCNICO EM INFORMÁTICA

COMO TRADUZIR UM RELACIONAMENTO 1:1 COM PARTICIPAÇÃO OPCIONAL?

Neste esquema, as colunas relacionadas ao casamento estão em negrito.

Elas incluem tanto os atributos do relacionamento (**Data**, **Regime**) quanto a chave estrangeira (**IdentH**), que é o identificador do homem.

Neste exemplo, a escolha foi adicionar essas colunas à tabela **Mulher**. No entanto, a decisão é arbitrária. A mesma implementação poderia ter sido feita adicionando as colunas (identificador da mulher, data e regime de casamento) à tabela Homem.



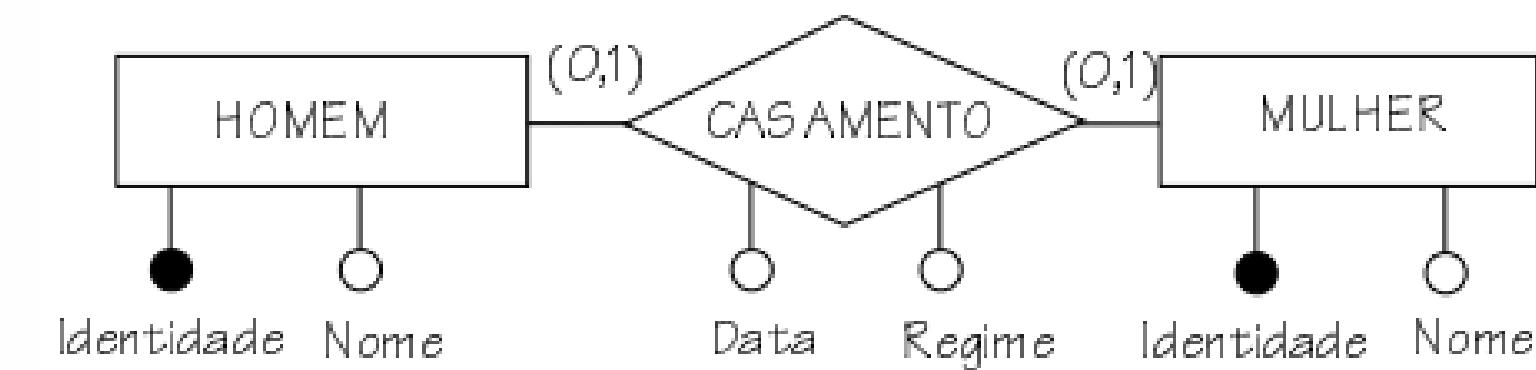
Uma solução poderia ser a apresentada abaixo:
Mulher (IdentM,Nome,IdentH,Data,Regime)
IdentH referencia Homem
Homem (IdentH,Nome)



CURSO TÉCNICO EM INFORMÁTICA

COMO TRADUZIR UM RELACIONAMENTO 1:1 COM PARTICIPAÇÃO OPCIONAL?

A outra alternativa seria a de gerar uma tabela própria para o relacionamento, conforme o esquema abaixo:



Mulher (IdentM, Nome)

Homem (IdentH, Nome)

Casamento (IdentM, IdentH, Data, Regime)

IdentM referencia Mulher

IdentH referencia Homem

Implementação com Tabela de Relacionamento

A segunda alternativa para implementar um relacionamento 1:1, como no exemplo do **casamento**, é usar uma tabela específica para o relacionamento, a tabela **Casamento**.

Nessa tabela:

- As colunas **IdentH** e **IdentM** são ambas chaves estrangeiras, conectando cada linha de casamento a um Homem e uma Mulher.
- Como o relacionamento é **1:1**, qualquer uma das duas colunas (IdentH ou IdentM) pode ser a chave primária.
- No exemplo, IdentM foi escolhida arbitrariamente como chave primária, e IdentH se torna uma chave alternativa.

