

Modelagem de dados

Prof. Daniel S. Kaster

Departamento de Computação
Universidade Estadual de Londrina
dskaster@uel.br

Modelo de dados

- Um modelo de dados é uma coleção de conceitos que podem ser usados para
 - **Representar** a estrutura de um conjunto de dados
 - Tipos de dados, relacionamentos e restrições
 - **Manipular** os dados
 - Atualizações e consultas
- Através da modelagem o projetista do banco de dados abstrai os elementos do mundo real e representa-os de forma organizada e interrelacionada, para serem depositados no banco de dados

Etapas do desenvolvimento de um Banco de Dados

Análise do problema

(Regras do negócio e requisitos)



Modelo conceitual

(Modelo Entidade-Relacionamento e extensões)



Modelo lógico

(Modelo Relacional, Objeto-Relacional, Semiestruturado, ...)



Implementação

(SQL: Definição de dados, funções, atualizações e consultas)

Modelo conceitual Entidade-Relacionamento

Entidades

- Entidade: que é uma “coisa” no mundo real que tem uma existência independente.
 - objeto com uma existência física: pessoa, casa
 - objeto conceitual: companhia, conta bancária
- Tipo de entidade: agrupa entidades de mesma natureza

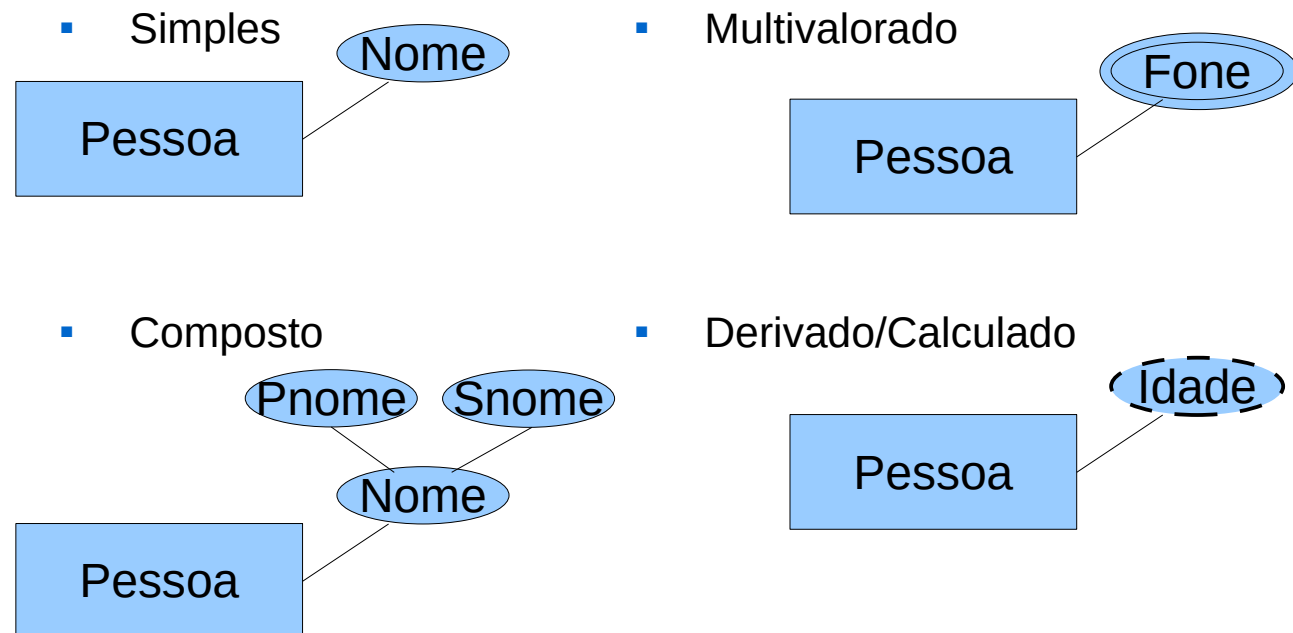


Pessoa

A light blue rectangular box with a thin black border, containing the word 'Pessoa' in black text, representing an entity type.

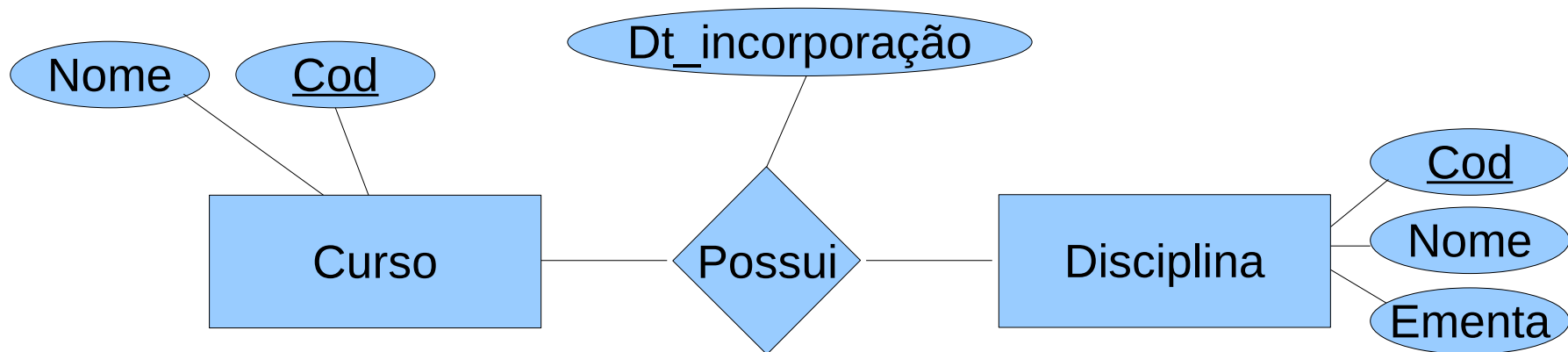
Atributos

- Propriedades particulares que descrevem uma entidade (ou relacionamento)
- Chave: um atributo (ou conjunto de atributos) é chave quando não existe no banco de dados nenhuma duplicação deste valor (ou conjunto de valores), garantindo a identidade de cada entidade



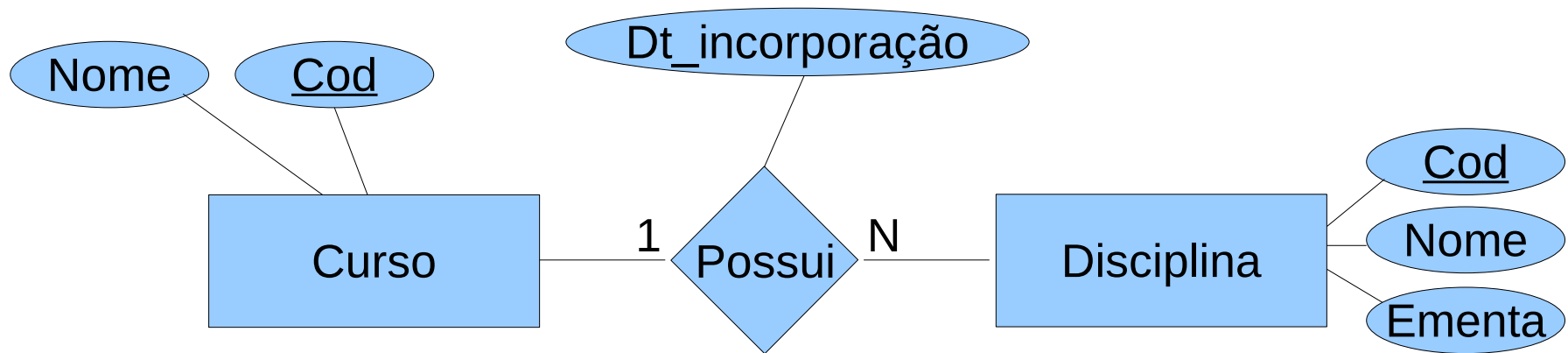
Relacionamentos

- Relacionamentos: são associações implícitas entre entidades
- Tipo de relacionamento: agrupa relacionamentos entre entidades de mesma natureza
 - Um relacionamento pode ter atributos associados, que fazem sentido apenas na associação entre as entidades



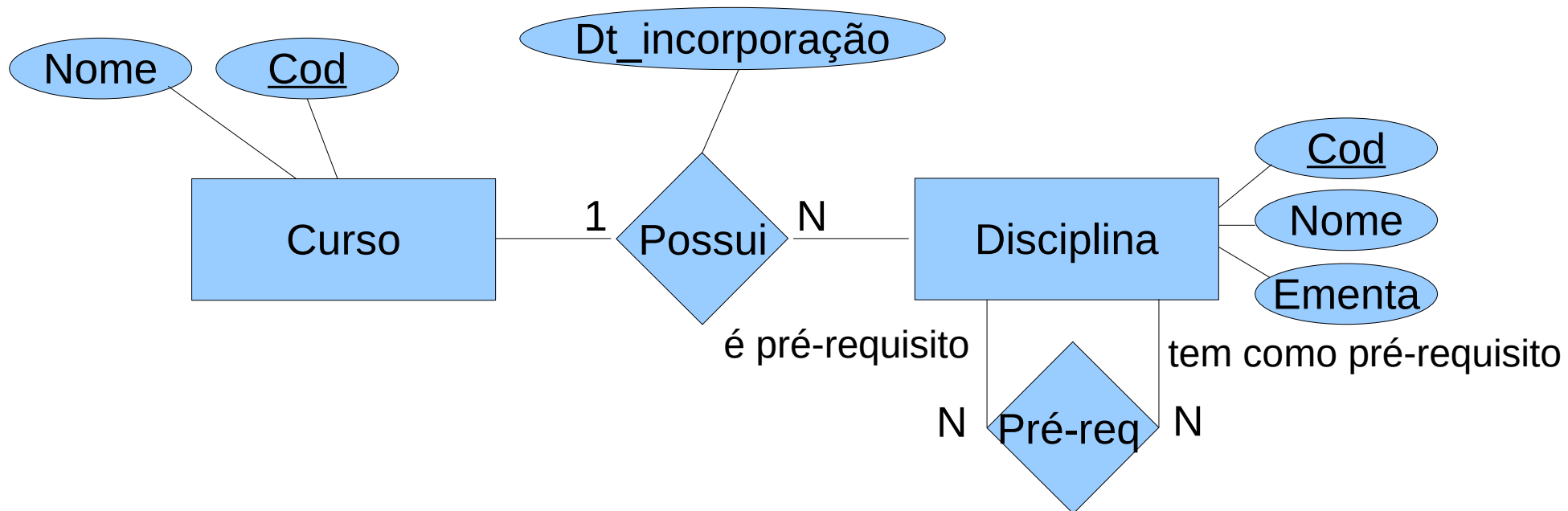
Classificação de relacionamentos

- Grau: quantidade de tipos de entidade participantes (binário, ternário, etc.)
- Cardinalidade: quantidade de entidades participantes (1:1, 1:N, N:M)



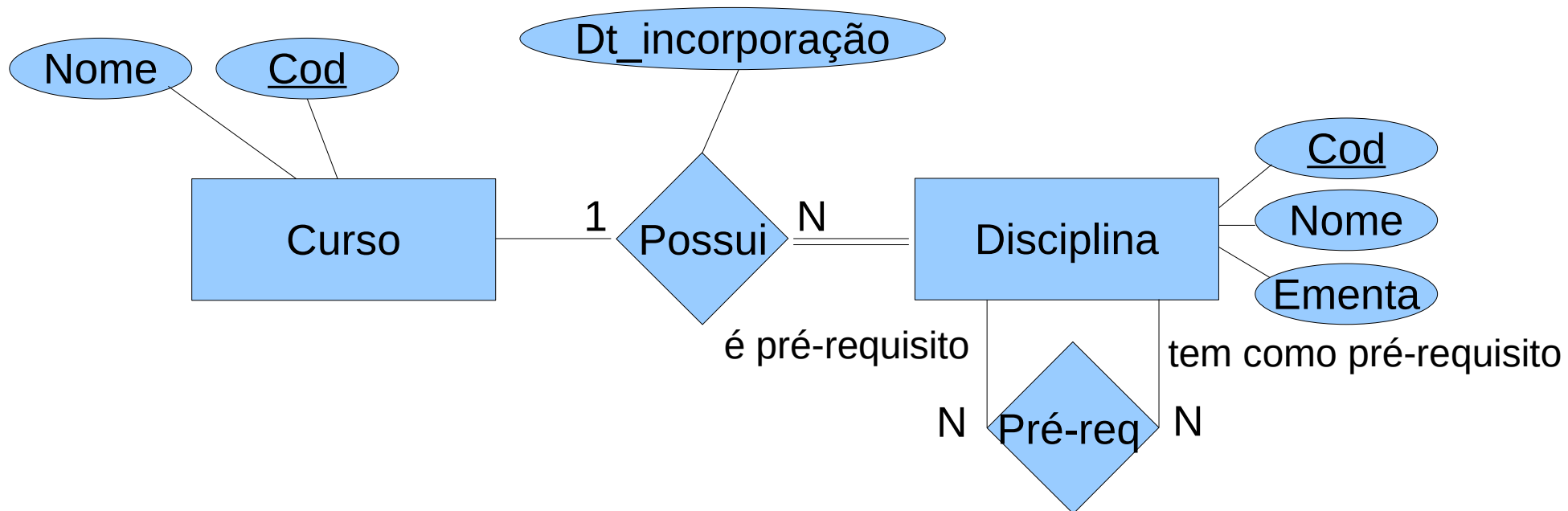
Relacionamentos recursivos

- Relacionamentos recursivos (ou auto-relacionamento) são aqueles que envolvem o mesmo tipo de entidade
 - É necessário explicitar as regras/papéis em que a entidade participa



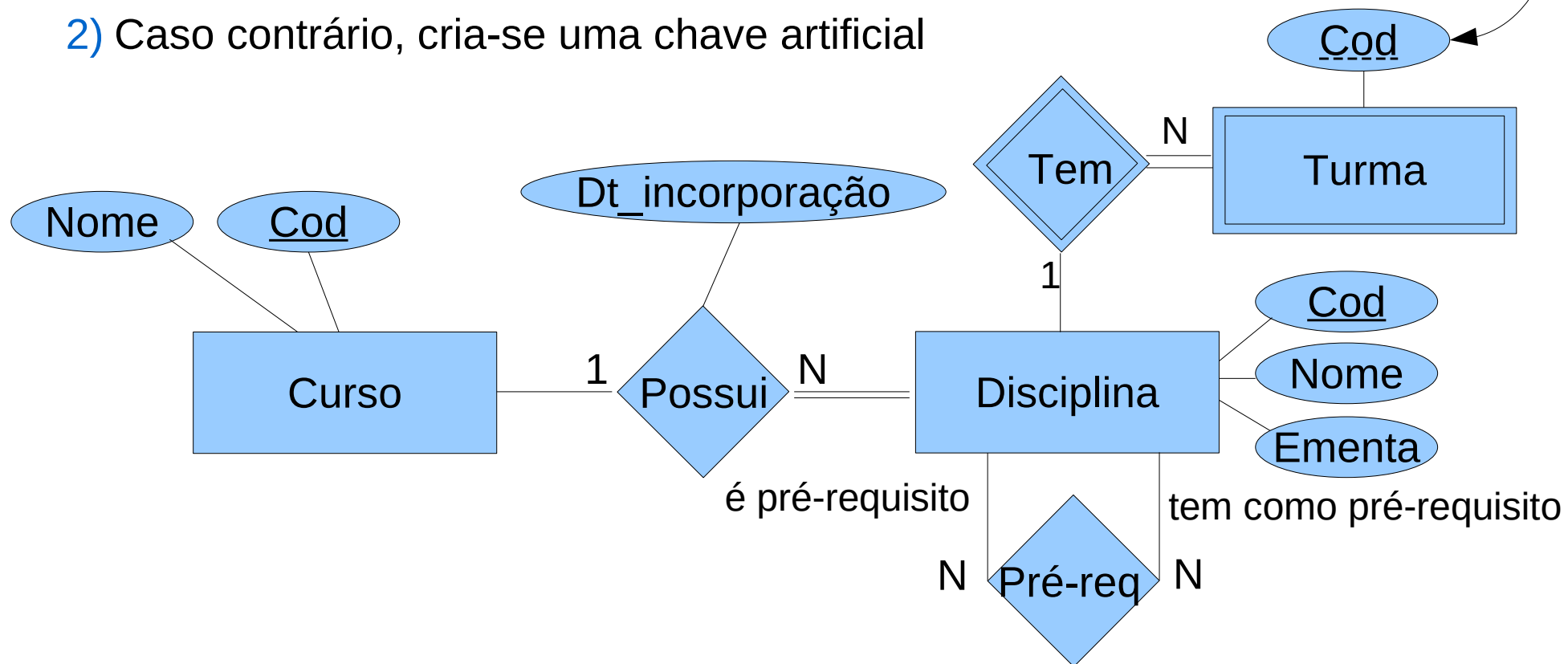
Participação em relacionamentos

- Participação total (=): para que uma entidade exista, ela deve estar relacionada a outra entidade
- Participação parcial/regular (—): uma entidade pode existir por si só



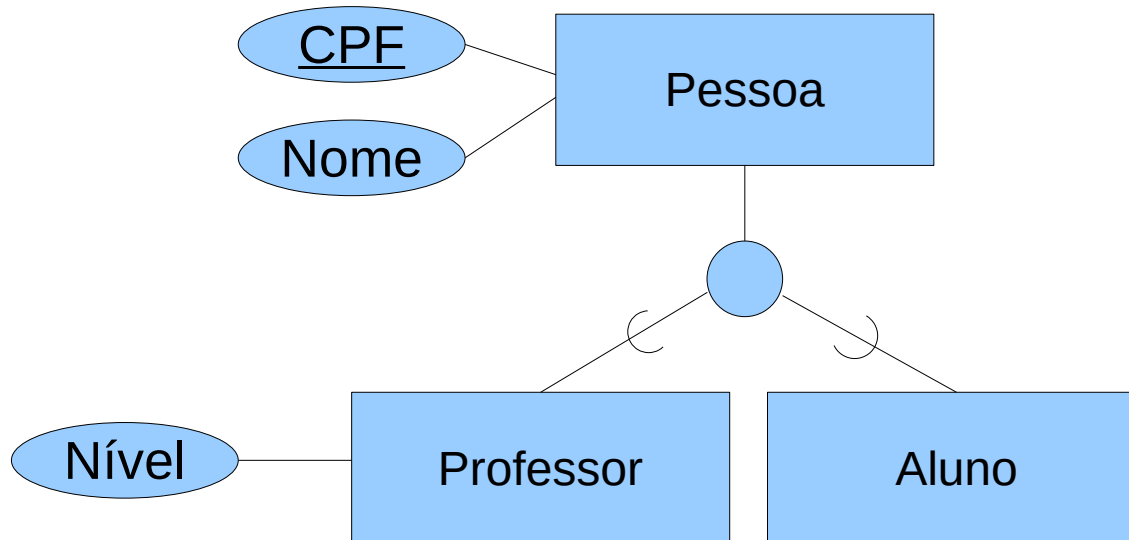
Entidades fracas

- Se uma entidade não possui atributos que a identificam:
 - Se pode ser identificada através de sua associação com outra(s) entidade(s) (entidade(s) identificadora(s)) é uma entidade fraca
 - Caso contrário, cria-se uma chave artificial



Herança

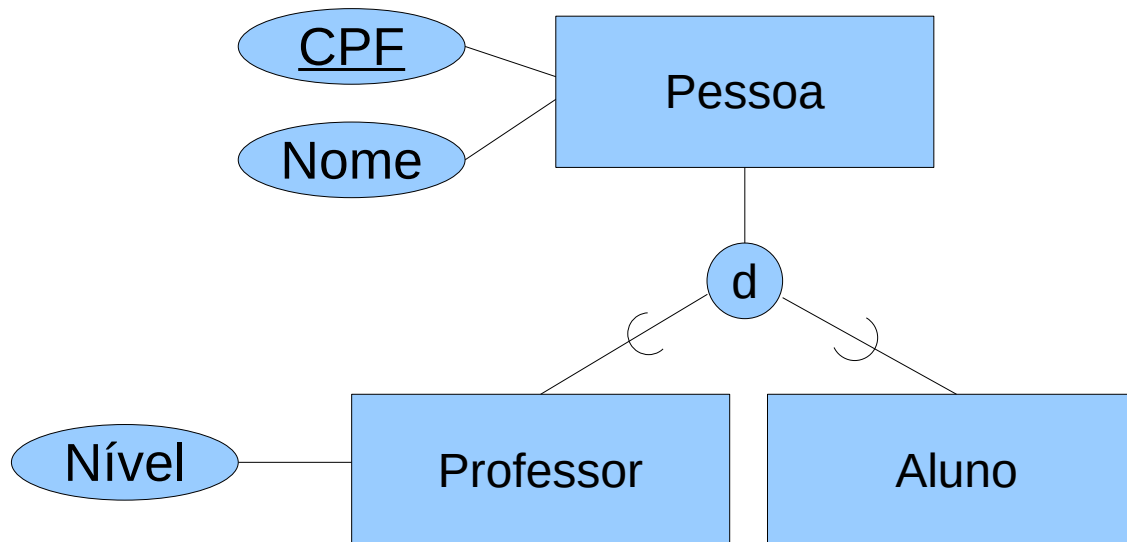
- Usada para expressar as noções de especialização ou generalização de tipos de entidades
 - Restrição elementar: toda entidade que ocorre em uma subclasse tem que ocorrer na superclasse



Restrições de herança

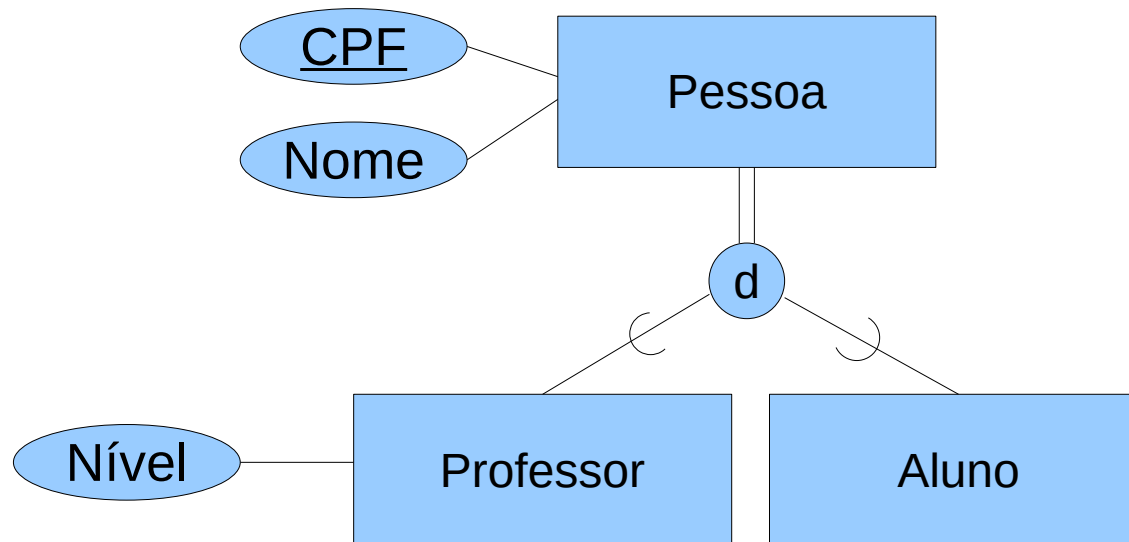
- Subclasses

- Disjunção (d): cada entidade pertence a apenas uma subclasse
- Sobreposição ou overlap (o): uma entidade pode pertencer a mais de uma subclasse



Restrições de herança(2)

- Participação da superclasse
 - Total (\equiv): toda entidade que existir na superclasse tem que existir em uma subclasse
 - Parcial (\dashv): a entidade pode existir apenas na superclasse



Modelo Relacional

Introdução

- O modelo relacional representa o banco de dados como um conjunto de relações, que podem ser entendidas como tabelas de valores com nome, atributos (colunas) e tuplas (linhas)
- Proposta: uma visão dos dados independente de aspectos de implementação
 - Edgar Frank "Ted" Codd. A relational model of data for large shared data banks. Communications of the ACM (CACM), volume 13, issue 6, pages 377-387, June 1970
- Possui uma forte fundamentação matemática, principalmente teoria de conjuntos e lógica de predicados

Aspectos que compõem o modelo relacional

- Representação
 - Definições e restrições de integridade
- Definição de dados
 - Normalização
 - Mapeamento do modelo E-R para o modelo Relacional
- Manipulação de dados
 - Álgebra relacional
 - Cálculo relacional (de tuplas e de domínio)

Relação

- Uma relação é uma instância de um esquema
- Seja $R(A_1, A_2, \dots, A_n)$ um esquema, uma relação $r(R)$ é um conjunto de n -tuplas $r = \{t_1, t_2, \dots, t_m\}$, onde cada tupla é uma lista de valores $t_i = \langle v_1, v_2, \dots, v_n \rangle$
- Cada $t_i.v_j$ é o valor do atributo A_j na tupla t_i e pertence ao domínio deste atributo ou é um valor nulo (null), quando inexistente ou desconhecido
- Um domínio de dados é um conjunto de valores atômicos, indivisíveis e monovalorados

Relações são conjuntos

- Como uma relação é um conjunto, não permitem repetições
- Também não existe a ideia de ordem, nem de tuplas e nem de atributos
- Assim, importa apenas indicar-se que cada valor corresponde ao respectivo atributo
- Obs: nos SGBDs que implementam o modelo relacional existe ordem de atributos e tuplas, embora isso seja apenas uma questão de implementação e não uma restrição do modelo

Restrições de integridade do modelo

1) Restrição de domínio

O valor de cada atributo A deve ser um valor atômico do domínio $\text{dom}(A)$, satisfazendo o tipo de dados e o formato especificado

2) Restrição de chave

Um mesmo valor para os atributos de uma chave (primária ou candidata) não pode ocorrer em duas tuplas diferentes de uma relação

3) Integridade de entidade

Estabelece que nenhum atributo que compõe uma chave pode ser nulo

4) Integridade referencial

É especificada entre duas relações, para manter a consistência entre as tuplas das duas relações (chaves estrangeiras)

Notação – chaves primárias e candidatas

- O conjunto de atributos que forma a chave primária aparece sublinhado, indicando que a combinação de atributos forma a chave

Empregado(rg_num, rg_est_exp, pnome, snome, ...)

- Opcionalmente, outras chaves candidatas podem ser destacadas, utilizando uma notação diferenciada (e.g., sublinhado duplo)

Departamento(num, nome, ...)

- Não há diferença entre sublinhado contínuo ou não; por exemplo, todos os exemplos a seguir indicam a mesma coisa

Empregado(rg_num, rg_est_exp, pnome, snome, ...)

Empregado(rg_num, rg_est_exp, pnome, snome, ...)

Empregado(rg_num, pnome, snome, rg_est_exp, ...)

Notação – chaves estrangeiras

- Chaves estrangeiras podem ser denotadas usando notações alternativas

Departamento(num, nome, ...)

Projeto(num, nome, ..., dept_num_controlado^{Departamento})

- Esta notação é uma proposta do professor =)
- Em geral, não é necessário indicar os atributos referenciados, pois a notação implica a referência aos atributos que formam a chave primária da relação referenciada
 - Se houver ambiguidade de interpretação (i.e., não fica claro qual atributo da relação referenciadora referencia qual atributo da relação referenciada), é preciso indicar explicitamente os atributos

Notação – chaves estrangeiras combinadas

- Uma chave estrangeira combinada não é equivalente a múltiplas chaves separadas

Empregado(rg_num, rg_estado, pnome, snome, ...)

Departamento(num, nome, ger_rg_num^{Emp.rg_num},
ger_rg_estado^{Emp.rg_estado})

- Neste exemplo, há duas chaves estrangeiras separadas, o que garante a ocorrência de um e outro valor nos respectivos atributos referenciados, mas não a combinação dos valores
- Isto implica que, na notação adotada, a continuidade da sobrelinha faz diferença, pois indica uma chave combinada

Empregado(rg_num, rg_estado, pnome, snome, ...)

Departamento(num, nome, ger_rg_num, ger_rg_estado^{Empregado})

Mapeamento do modelo Entidade-Relacionamento para o modelo Relacional

Introdução

- O modelo relacional é um modelo lógico que define um conjunto de definições para representar a estrutura de um banco de dados e técnicas para eliminação de redundâncias (normalização)
- Entretanto, em geral, é mais natural para o projetista conceber um modelo em um nível conceitual, utilizando um modelo semântico como o modelo Entidade-Relacionamento
- Desta forma, desenvolveu-se um algoritmo para efetuar o mapeamento das construções do modelo E-R para o modelo relacional, respeitando todas as definições impostas pelo modelo relacional, gerando um esquema relacional de banco de dados normalizado

Mapeamento de entidades regulares

- 1) Para cada entidade regular E no diagrama E-R, crie uma relação R que inclui os seus atributos simples
 - São incluídos apenas os componentes simples de um atributo composto
- 2) Escolha uma chave candidata e defina-a como chave primária da relação
 - Se a chave candidata é composta, seus atributos simples formarão juntos a chave primária de R

Mapeamento de atributos multivalorados

- 1) Para cada atributo multivalorado, crie uma nova relação S
- 2) Inclua o atributo multivalorado e a chave primária da relação de origem do atributo multivalorado como chave estrangeira
- 3) Defina a chave primária da nova relação como a combinação destes atributos
 - Obs: se o atributo multivalorado é composto, inclui-se os seus componentes simples

Mapeamento de relacionamentos binários 1:1 e 1:N

- 1) Para cada relacionamento binário 1:N, identifique a relação S que representa a entidade participante do lado N no relacionamento
- 2) Inclua, como chave estrangeira de S, a chave primária de T, que representa a outra entidade participante no relacionamento
- 3) Inclua todos os atributos deste relacionamento como atributos de S

OBS: para relacionamentos 1:1

- 1) Tem-se a opção de escolher o lado mais conveniente para definir a chave estrangeira
- 2) É preciso garantir a unicidade da chave estrangeira

Mapeamento de relacionamentos binários N:M e de grau maior que 2

- 1) Para cada relacionamento binário N:M, crie uma nova relação U para representar este relacionamento
 - 2) Inclua, como chaves estrangeiras de U as chaves primárias das relações das entidades participantes
 - 3) Inclua todos os atributos do relacionamento N:M como atributos de U
 - 4) Defina a chave primária de U como a combinação das chaves estrangeiras das entidades participantes
- Obs: Esta abordagem também pode ser usada para relacionamentos 1:1 e 1:N, ajustando-se a chave da relação
 - 1:N – apenas a chave estrangeira do lado N compõe a chave primária de U
 - 1:1 – as duas chaves estrangeiras são chaves candidatas de U; elege-se uma delas como chave primária

Mapeamento de entidades fracas

- 1) Para cada entidade fraca E_f no diagrama E-R, com entidade identificadora E , crie uma relação R e inclua todos os atributos simples de E_f como atributos de R
- 2) Inclua, também, como chave estrangeira, os atributos da chave primária de E , para estabelecer o vínculo de existência entre elas
 - A chave primária da nova relação R é formada pelos atributos chave da entidade identificadora combinados com a chave parcial de E_f (se existirem)

Mapeamento de hierarquias

- Para cada especialização, contendo uma superclasse Super, com os atributos $\{K, A_1, A_2, \dots, A_n\}$, onde K é o subconjunto de atributos que formam a chave da superclasse ($PK(\text{Super}) = K$) e m subclasses $\{\text{Sub}_1, \text{Sub}_2, \dots, \text{Sub}_m\}$, converta em esquemas relacionais usando uma das alternativas a seguir

A) Mapeamento de superclasse e subclasses separadamente

- 1) Crie um esquema relacional $SP(K, A_1, A_2, \dots, A_n)$ para Super e faça $PK(SP) = K$
 - 2) Crie um esquema relacional SB_i para cada subclasse Sub_i , com os atributos de Sub_i mais o conjunto de atributos K e faça $PK(SB_i) = K$
- Opção usada para qualquer restrição: disjunta, overlap, parcial ou total.

Mapeamento de hierarquias(2)

B) Mapeamento apenas das subclasses

- 1) Crie um esquema relacional SB_i para cada subclasse Sub_i , com os atributos de Sub_i mais os atributos de Super, e faça $PK(SB_i) = K$
- Opção usada para especialização disjunta e total
 - Se a especialização possui overlap são geradas redundâncias e, se for parcial, uma entidade que não aparece em nenhuma subclasse é perdida

Mapeamento de hierarquias(3)

C) Mapeamento da superclasse e das subclasses em uma única relação

- 1) Crie um único esquema relacional SP contendo os atributos da superclasse Super e os atributos de todas as subclasses Sub_i e faça $PK(SP) = K$
 - 2) Acrescente a esta relação um atributo discriminador t que indica a subclasse à qual a entidade pertence
 - Se a especialização é disjunta, t é um atributo simples
 - Se a especialização é com overlap, t é um conjunto de m atributos booleanos (ou um atributo binário com m bits)
- Opção de mapeamento que tende a gerar nulos, mas pode agilizar a execução de algumas consultas

SQL para Definição de Dados (SQL-DDL)

Implementação relacional

- Instrução CREATE TABLE
 - Definição do nome da relação (dentro de um esquema)
 - Definição dos atributos, com seus tipos de dados e restrições de nulidade
 - Definição das restrições de tabela (PRIMARY KEY, UNIQUE, CHECK)
 - Definição das chaves estrangeiras (FOREIGN KEY)
- Instrução ALTER TABLE
 - Permite a evolução do esquema
 - Adicionar/remover/alterar atributos e restrições, renomeação, etc.
- Instrução DROP TABLE
 - Remove relações

Exemplo de tabela

```
CREATE TABLE rh.empregado (  
    cpf INT,  
    pnome CHAR(40) NOT NULL,  
    snome CHAR(60) NOT NULL,  
    endereco VARCHAR(100) DEFAULT 'Campus UEL',  
    dnum INT,  
    CONSTRAINT pk_empr PRIMARY KEY(cpf),  
    CONSTRAINT un_empr UNIQUE (pnome, snome),  
    CONSTRAINT fk_empr_dept FOREIGN KEY(dnum)  
        REFERENCES rh.departamento(dnum),  
    CONSTRAINT ck_empr_cpf CHECK (cpf > 0)  
);
```

Sequências

- Sequências são úteis para a definição de valores para chaves artificiais (autoincrementadas)
- São atualizadas fora do contexto de transações, pois são pontos de gargalo
 - Como consequência, uma falha em uma atualização faz com que o valor da sequência seja "inutilizado"
- Ex:

```
CREATE SEQUENCE departamento_num_seq  
START 10 INCREMENT 2;
```

```
ALTER TABLE departamento  
ALTER num SET DEFAULT nextval('departamento_num_seq');
```

Propagação de atualizações

- Define como as atualizações de um atributo/tupla referenciado devem ser propagadas para os atributos/tuplas que o referenciam
- Cláusula ON DELETE na definição de FKs
- Opções
 - Restrita (RESTRICT; NO ACTION)
 - Cascadeamento (CASCADE)
 - Atribuir nulo (SET NULL)
 - Atribuir valor default (SET DEFAULT)

Definição do momento da verificação de restrições

- Restrições de integridade podem ser
 - DEFERRABLE: sua verificação pode ser atrasada para o fim da transação
 - NOT DEFERRABLE: verificação imediata (padrão)
- No caso de ser DEFERRABLE, há 2 opções
 - INITIALLY IMMEDIATE: por padrão, a verificação é imediata
 - INITIALLY DEFERRED: por padrão, a verificação é atrasada
- Neste caso, o momento da verificação das restrições de uma transação pode ser alterado com a instrução SET CONSTRAINTS