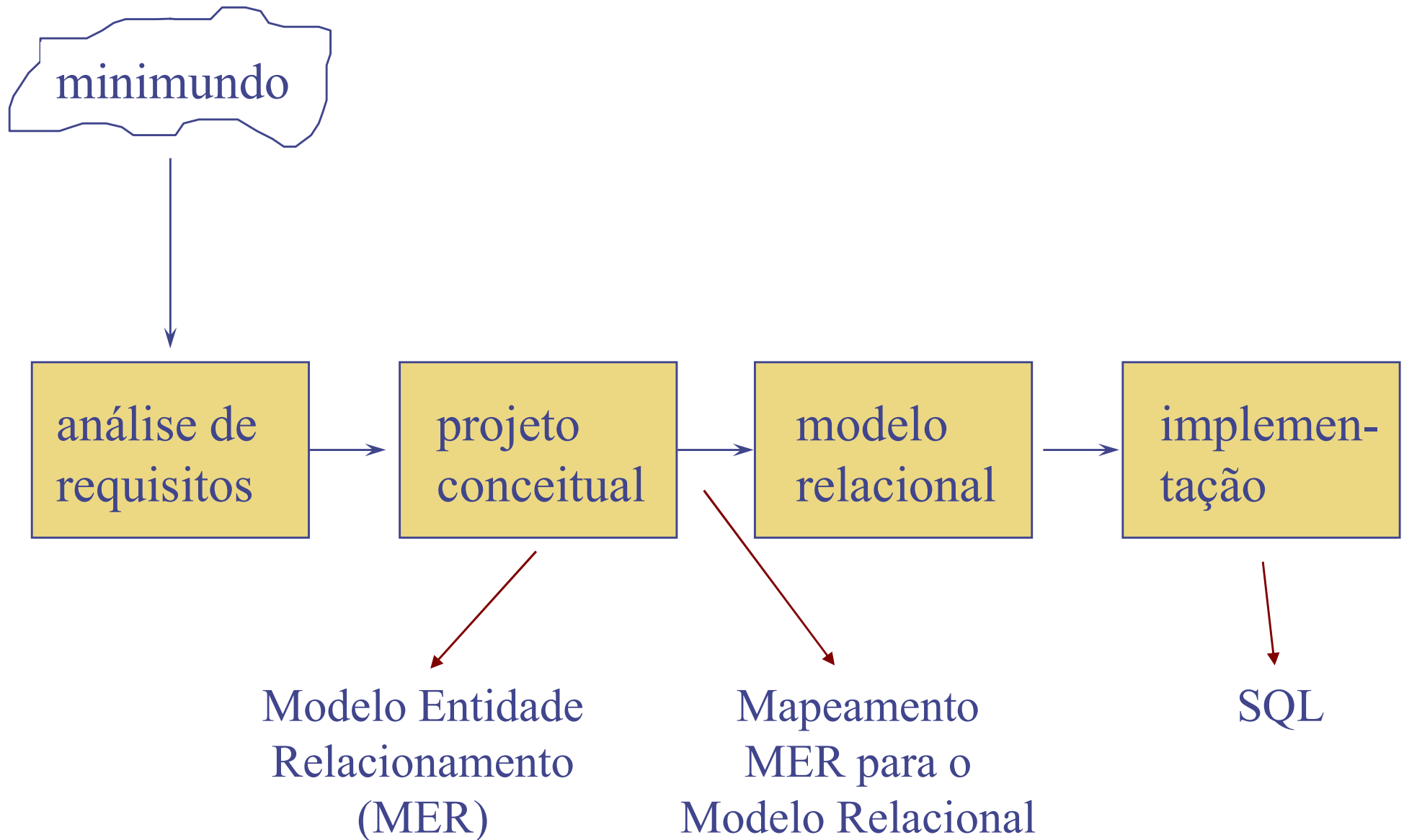




Introdução ao Modelo Relacional

Prof. Humberto Luiz Razente
Bloco B - sala 1B144

Modelo de Dados e o Projeto de BD



Modelo Relacional

- ◆ Introduzido por Edgar (Ted) Codd da IBM em 1970
 - Simplicidade e base matemática
 - Base teórica na teoria de conjuntos e na lógica de predicados de primeira ordem

- ◆ Primeiras implementações: final 70's
 - Projetos: System R e Ingres
 - ◆ System R → SQL/DS, DB2, Oracle
 - ◆ Ingres → Informix, Sybase, SQLServer, Ingres

Modelo Relacional

◆ BD

- representado como uma coleção de relações

◆ Relação

- possui um nome que é único em um BD
- é uma tabela bi-dimensional

Tabela Bi-Dimensional

◆ Características

- cada coluna tem um nome distinto e representa um **atributo**
- cada atributo possui um **domínio** de valores
- cada domínio possui **VALORES ATÔMICOS**
 - ◆ por atômico entendemos que o valor é indivisível no domínio
- todos os valores de uma coluna são valores do **mesmo** atributo

Tabela Bi-Dimensional

◆ Características

- cada **linha** da tabela representa o relacionamento entre um conjunto de valores
- cada linha é distinta e representa uma **tupla**
- uma **n-tupla** representa uma tupla que possui **n** valores
 - ◆ grau da relação: número **n** de atributos de sua relação esquema

Exemplo: Tabela Aluno

relação esquema: Aluno (nmat, nome, endereço, idade)

nome do atributo

atributo

nmat	nome	endereço	idade
935639	Adriana Zagalo	Rua Floriano Peixoto, 1	
935632	Beatriz da Silva	Rua Itambé, 124 apto 62 bloco B	22
933219	Carlos Alberto Bozato	Rua Sucupira, 3452 apto 125	19
938904	Antônio Nascimento	Av. Castro Alves, 57	18
934789	Roberto Antonione	Av. Sunab Jatab, 3467 apto 32	32

Intenção do BD
Esquema

Extensão do BD
Banco de dados

valor

Definições Formais

◆ Relação esquema R:

- utilizada para descrever uma relação
- denotada por $R(A_1, A_2, \dots, A_n)$
- formada por
 - ◆ um nome de relação R
 - ◆ uma lista de atributos A_1, A_2, \dots, A_n
- para cada atributo A_i ($1 \leq i \leq n$)
 - ◆ $\text{dom}(A_i)$: domínio de A_i
 - ◆ domínio: conjunto de valores atômicos
- caracteriza a **intenção** do BD

Definições Formais

◆ Exemplos de domínios para

- Aluno (nmat, nome, telefone, celular, idade)
 - ◆ Números de telefone
 - ◆ Nomes de aluno
 - ◆ Idade
- Um método comum para especificar o domínio compreende
 - ◆ Definição lógica
 - ◆ Definição do tipo de dado ou formato

Definições Formais

◆ Aluno (nmat, nome, telefone, celular, idade)

- Definição lógica

1. **Números de matrícula:** conjunto de dígitos válidos para matrícula de alunos
2. **Números de telefone:** conjunto de números de telefone válido no Brasil
3. **Nomes de aluno:** conjunto de todos os nomes possíveis para pessoas
4. **Idade:** conjunto de idades possíveis para alunos

- Definição do tipo de dado ou formato

1. **Números de matrícula:** inteiro com 8 dígitos
2. **Números de telefone:** inteiro com 10 dígitos
3. **Nomes de aluno:** string de 60 caracteres
4. **Idade:** inteiro entre 0 e 127

Definições Formais

◆ Exemplos de domínios para

- Aluno (nmat, nome, telefone, celular, idade)
 - ◆ $\text{dom}(\text{nmat}) = \text{Números de matrícula}$
 - ◆ $\text{dom}(\text{nome}) = \text{Nomes de aluno}$
 - ◆ $\text{dom}(\text{telefone}) = \text{Números de telefone}$
 - ◆ $\text{dom}(\text{celular}) = \text{Números de telefone}$
 - ◆ $\text{dom}(\text{idade}) = \text{Idade}$

Definições Formais

- ◆ Relação r da relação esquema $R(A_1, A_2, \dots, A_n)$
 - representa a instância da relação
 - denotada por $r(R)$
 - formada por um conjunto de n -tuplas
 $r = \{t_1, t_2, \dots, t_m\}$
 - ◆ cada n -tupla t é uma lista de n valores
 $t = \langle v_1, v_2, \dots, v_n \rangle$
 - ◆ v_i ($1 \leq i \leq n$) é um elemento de $\text{dom}(A_i)$ ou um valor nulo (i.e., null)
 - caracteriza a **extensão** do BD

Definições Formais

◆ Exemplo de possível relação do esquema

◆ Aluno (nmat, nome, telefone, celular, idade)

$r(\text{Aluno}) = \{ \langle 22222222 \rangle, \text{Júlia}, 1134343434, 1126262626, 21 \rangle, \\ \langle 11111111 \rangle, \text{Pedro}, 1965656565, 1977777777, 18 \rangle, \\ \langle 99999999 \rangle, \text{Cecília}, 1144443333, 1165658888, 23 \rangle \}$

Características das relações

- ◆ Ordenação de tuplas em uma relação (nível abstrato)
 - matematicamente, não há ordem entre os elementos de um conjunto
 - na implementação de um SGBDR existe uma ordem física de armazenamento das tuplas na memória externa
 - ◆ pode determinar uma ordem na recuperação das informações

- ◆ Ordenação de tuplas em uma relação (nível lógico)
 - muitas ordens lógicas podem ser especificadas para uma relação
 - ◆ relação ALUNO pode ser ordenada pelos atributos NOME, DATANASCIMENTO, CPF, etc.

Características das relações

◆ Ordenação de valores dentro de uma tupla

- uma tupla é uma lista de n valores dispostos em uma ordem determinada de acordo com a disposição dos atributos no esquema da relação

◆ Valores nas tuplas

- são atômicos (monovalorados)
 - ◆ relações não permitem atributos multivalorados
- o valor **null** deve ser utilizado quando um atributo não possui valor ou seu valor não é conhecido

Restrições sobre uma Relação

◆ Domínio

- dentro de cada tupla, o valor de cada atributo A deve ser um valor **atômico** de $\text{dom}(A)$

◆ Unicidade de chave

- Chave primária
 - ◆ identifica de forma única cada tupla da relação

Restrições sobre uma Relação

◆ Valor nulo

- permitido: **null** (padrão)
- não permitido: **not null**

◆ Integridade de entidade

- **nenhum** valor de chave primária pode ser nulo
 - ◆ permitir valores **null** para a chave primária implica que não podemos identificar algumas tuplas
 - não seria possível diferenciá-las ao tentar referenciá-las por outras relações

Restrições sobre uma Relação

Chaves e Superchaves

- ◆ Uma superchave de uma relação R é um conjunto de atributos S contido em R
 - no qual não haverá duas tuplas t_1 e t_2 cujo $t_1[S] = t_2[S]$

Restrições sobre uma Relação

Chaves e Superchaves

- ◆ Uma chave K é uma superchave com a propriedade adicional de que a remoção de qualquer atributo da chave fará com que K não identifique mais unicamente cada tupla da relação
 - a diferença é que uma chave tem que ser mínima

Restrições sobre uma Relação

Chaves e Superchaves

Aluno (nmat, nome, telefone, celular, idade)

◆ Exemplo:

- {nmat} é uma chave de aluno
- Superchaves
 - ◆ {nmat, nome}
 - ◆ {nmat, nome, telefone}
 - ◆ {nmat, nome, telefone, celular}
 - ◆ {nmat, nome, telefone, celular, idade}
 - ◆ ...

Restrições sobre uma Relação

Chaves e Superchaves

◆ Chave candidata:

- se um esquema de relação tiver mais de uma chave, cada uma delas é chamada chave candidata
- uma delas é arbitrariamente designada para ser chave primária

◆ Um atributo de um esquema de relação R é chamado atributo primário se for membro de alguma chave candidata

Restrições sobre uma Relação

Chaves e Superchaves

Aluno (nmat, nome, telefone, celular, idade)

◆ Exemplo:

- {nmat} é a única chave candidata de aluno, portanto também é a chave primária

Restrições sobre uma Relação

Chave Primária

◆ Resumindo:

- chave primária para um esquema de relação R satisfaz duas restrições
 - ◆ duas tuplas distintas não podem ter valores idênticos para os atributos da chave
 - ◆ ela é uma superchave mínima

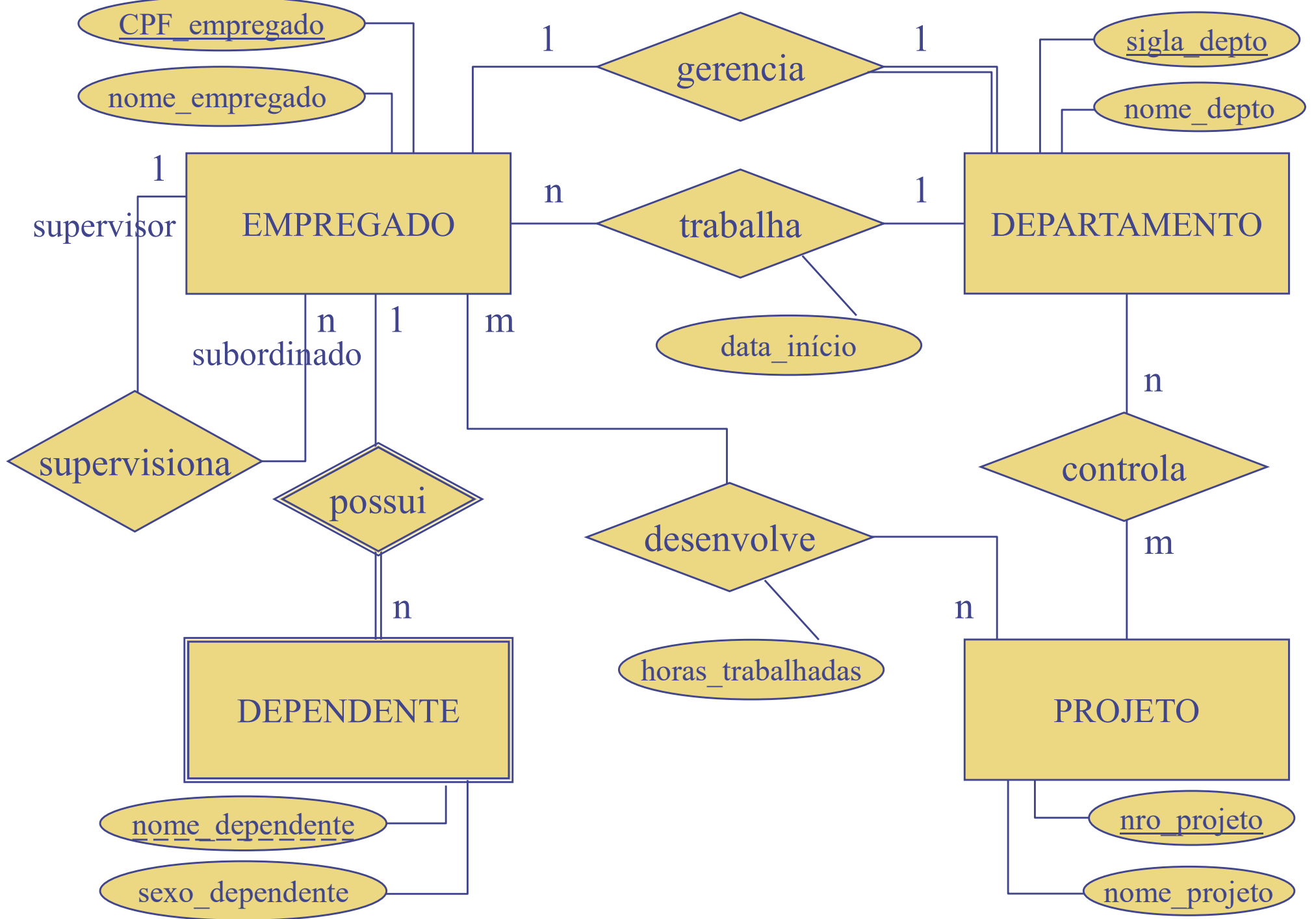
Definições

◆ Esquema de banco de dados S

- conjunto de relações esquema
 $S = \{R_1, R_2, \dots, R_m\}$
- conjunto de restrições de integridade IC

◆ Estado do banco de dados DB

- conjunto de estados da relação
 $DB = \{r_1, r_2, \dots, r_m\},$
onde cada r_i é um estado de R_i
- os estados de r_i devem satisfazer às restrições de integridade especificadas em IC



Esquema do BD Relacional

empregado (CPF empregado, nome_empregado,

cod_supervisor, sigla_depto, data_início)

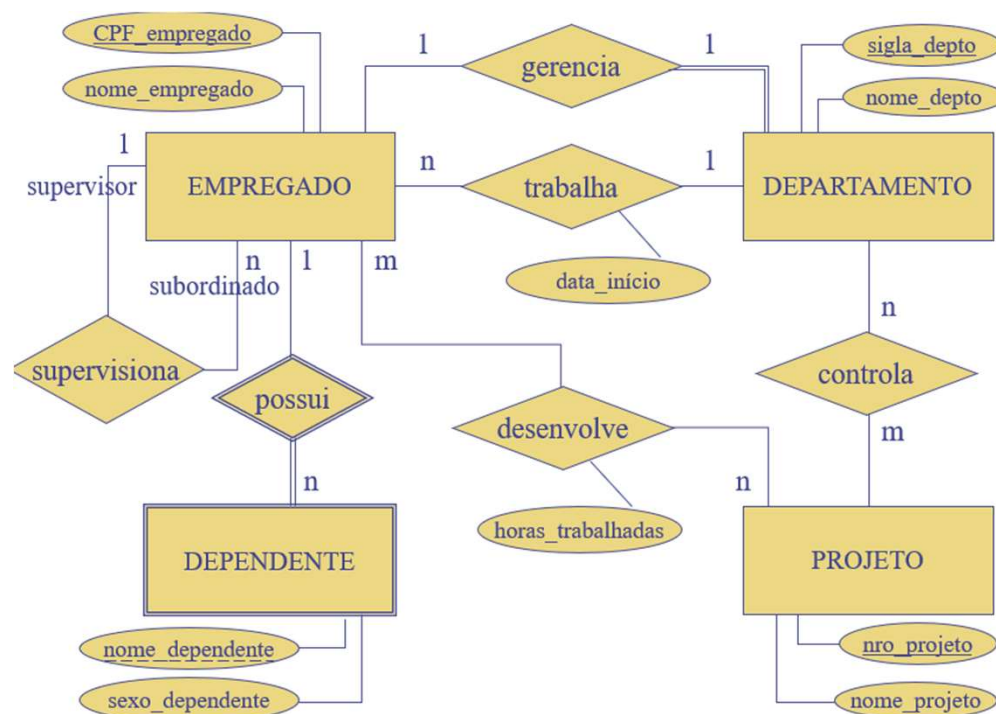
dependente (CPF empregado, nome dependente, sexo_dependente)

departamento (sigla_depto, nome_depto, CPF_empregado)

projeto (nro_projeto, nome_projeto)

controla (sigla_depto, nro_projeto)

desenvolve (CPF empregado, nro_projeto, horas_trabalhadas)



Restrições entre duas Relações

◆ Integridade referencial

- mantém a consistência entre as tuplas nas duas relações
- declara que uma tupla em uma relação, a qual faz referência a uma outra relação, deve se referir a uma tupla existente nessa segunda relação
- definida entre a **chave estrangeira** (FK) de uma relação esquema R_1 e a **chave primária** (PK) de uma relação esquema R_2

Restrições entre duas Relações

- ◆ FK de R_1 é **chave estrangeira** de R_1 , que faz referência à PK de R_2 , se:
 - os atributos de FK têm os mesmos domínios que os atributos de PK

Integridade Referencial

empregado (CPF empregado, nome_empregado,
cod_supervisor, sigla_depto, data_início)

dependente (CPF empregado, nome dependente,
sexo_dependente)

departamento (sigla_depto, nome_depto,
CPF_empregado)

projeto (nro_projeto, nome_projeto)

controla (sigla_depto, nro_projeto)

desenvolve (CPF empregado, nro_projeto,
horas_trabalhadas)

Restrições *versus* Operações

◆ Operações de modificação

- insert → inserção
- delete → remoção
- update → atualização

➤ Quando estas operações são aplicadas, as restrições de integridade especificadas no esquema do banco de dados relacional **não** devem ser violadas

Operação Insert

◆ Característica

- fornece uma lista de valores de atributos para uma nova tupla t , que é inserida em uma relação R

◆ Pode violar as seguintes restrições

- domínio
- unicidade de chave
- integridade de entidade
 - ◆ (chave primária *null*)
- integridade referencial

Solução:

- rejeitar a inserção
- enviar mensagem de erro ao usuário

Operação Delete

◆ Característica

- remove uma ou mais tuplas

◆ Pode violar a integridade referencial

- quando as tuplas removidas forem referidas por chaves estrangeiras de outras tuplas

◆ Soluções

- rejeitar a remoção
- remover em cascata
- modificar valores dos atributos de referência

Operação Update

◆ Característica

- altera valores de alguns atributos em tuplas

◆ Pode violar as seguintes restrições

- domínio
- unicidade de chave (se atributo é PK)
- integridade de entidade (se atributo é PK)
- integridade referencial (se atributo é FK)

◆ Soluções

- idem anteriores (para insert e delete)

Exercício

- ◆ A administradora trabalha com administração de condomínios e de aluguéis
 - ◆ Uma entrevista com o gerente resultou nas seguintes informações:
 - São administrados condomínios formados por unidades condominiais. Para cada condomínio a administradora precisa saber seu nome (que é único) e endereço (nome da rua, número, bairro e cep). Para cada unidade condominal seu número único e o andar
 - Cada unidade condominial é de propriedade de uma ou mais pessoas. Uma pessoa pode possuir diversas unidades
 - Cada unidade pode estar alugada para no máximo uma pessoa. Uma pessoa pode alugar diversas unidades. Uma data determina quando a pessoa alugou uma unidade condominal
 - Para cada pessoa são armazenadas informações como CPF, nome e telefones (residencial, comercial e celular)
-
1. Definir os domínios (definição lógica, tipo de dado e formato) necessários para a criação dos esquemas de relação para essa especificação
 - ex: nomes de pessoas: conjunto de todos os nomes possíveis para pessoas – strings de 60 caracteresD
 2. Definir os esquemas de relação para a especificação
 3. Instancie relações a partir dos esquemas criados exemplificando situações em que não são satisfeitas as seguintes restrições de integridade:
 - unicidade de chave, integridade de entidade e integridade referencial

Bibliografia e leitura complementar

- ◆ Elmasri, Ramez; Navathe, Shamkant B.
Sistemas de banco de dados:
 - 6ª edição: capítulo 3, O modelo de dados relacional e as restrições em bancos de dados relacionais