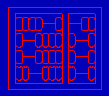
Modelos lógicos históricos



Modelo de redes:

definido pelo comitê “Codasyl Data Base Task Group”

Modelo hierárquico:

desenvolvido pela IBM para o SGBD IMS (mainframes IBM)

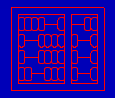
não desacoplavam modelagem lógica da física,

suplantados pelo modelo relacional,

são de interesse apenas histórico.

UNICAMP

O modelo relacional (MR)



Introduzido por E. Codd em 1970,

um único objeto básico para representar dados: tabela ou relação

uma tabela no MR é uma coleção não ordenada de linhas,

cada linha corresponde a uma entidade,

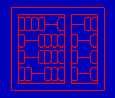
coleção de linhas corresponde a um CE do MER (ou a um relacionamento)

Criado seis anos antes do MER, é independente do mesmo,

porém analogia com o MER simplifica a compreensão dos conceitos e do mapeamento do MER para o MR

UNICAMP

O conceito de tabela (MR)



Uma linha é composta por uma lista de valores,

$\approx$ valores dos atributos da entidade,

número de valores em cada linha é constante,

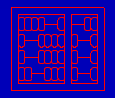
$\approx$ número de atributos da entidade.

O número de linhas é variável com o tempo.

Cada coluna corresponde a um atributo do CE.

Dado um nome para a tabela e um nome para cada coluna: $\approx$ esquema de uma tabela e esquema de um CE.

O conceito de tabela (MR)



MR: tabela $\equiv$ relação, linha $\equiv$ tupla.

Relação é um conjunto não ordenado de tuplas

cada tupla é um conjunto não ordenado de n pares $(A_1, v_1), (A_2, v_2), \ldots, (A_n, v_n)$, onde A_i = nome de atributo e v_i = valor de atributo

caracterizção: quais valores possíveis para cada atributo,

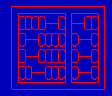
i. é, tipo de dados de cada atributo = (domínio)

um domínio especifica um conjunto de valores indivisíveis do mesmo tipo.

Cada atributo A_i corresponde um domínio D_i.

UNICAMP

O conceito de tabela ou relação



Definição matemática de relação R:

$$\mathrm{R} \subset D_1 \times D_2 \times \ldots \times D_n$$

subconjunto do produto cartesiano dos domínios $D_1 \ldots D_n$

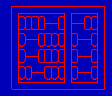
Base de dados relacional = conjunto de tabelas ou relações, cada tabela associada a uma lista de domínios

esquema de uma relação $=$
nome da relação(lista de nomes dos atributos):

Funcionários(numf, RG, CPF, nome, endereço, salário)

UNICAMP

Visualização de tabelas



No mundo ocidental: linhas de cima para baixo,colunas da esquerda para a direita,

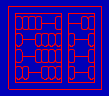
não corresponde à organização interna ou lógica da tabela!

economia de notação: ordenar atributos de acordo com o esquema da tabela

atributos ficariam implícitos ao se escrever o conteúdo da tabela

ao escrever instância, rotular colunas com os nomes dos atributos.

Visualização de tabelas



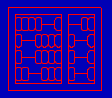
Exemplo: instância da tabela Funcionários:

numf	RG	CPF	nome	ender.	sal.
1234	98765	537865443-67	Ana Paula	Rua da Lapa 133	9200
4321	12312	987654321-99	Pedro Silva	Rua Cabral 321	920

Analogias úteis:

- Relação $\approx$ Tabela $\approx$ Arquivo
- Tupla $\approx$ Linha $\approx$ Registro
- Atributo $\approx$ Coluna $\approx$ Campo
- Domínio $\approx$ Tipo de dados da coluna $\approx$ Tipo de dados do campo

Chaves de tabelas

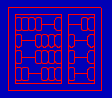


Chave é um atributo (ou lista de atributos) satisfazendo:

(i) identifica de forma única cada linha da tabela: dadas duas linhas o valor da chave nunca é o mesmo para as duas linhas

(ii) caso possua mais de um atributo, deve ser *minimal*: se retirarmos qualquer atributo da chave ela deixa de identificar univocamente as linhas da tabela.

Chaves de tabelas



Toda tabela deve ter pelo menos uma chave. Se tiver mais de uma, uma delas deve ser escolhida (projetada) como chave primária,

as outras chaves são denominadas chaves alternativas.

Exemplo:

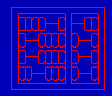
a tabela Funcionários tem três chaves, numf, RG e CPF

numf foi escolhida como chave primária (sublinhada),

RG e CPF são chaves alternativas.

UNICAMP

Conceitos de esquema e instância



Esquema de uma tabela: definição do nome e atributos,

em geral imutável, reflete projeto da tabela

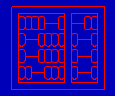
instância: conjunto de linhas na tabela num determinado instante (varia com o tempo).

Esquema de uma BD relacional: conjunto dos esquemas de todas as tabelas da BD

Instância da BD: conteúdo instantâneo de todas as tabelas da BD.

Não confundir: *relacionamento* (entre entidades do MER) com *relação* (tabela do MR)!

Valores nulos



O mundo real é imperfeito:

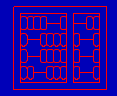
ao se preencher tabelas é comum faltarem informações sobre atributos de tuplas:

informação não existe ou é desconhecida, ou não está disponível no momento,

o valor do atributo não se aplica ou não faz sentido.

Solução paliativa: valor especial e genérico: *vazio* ou *nulo* (*null value*), compatível com qualquer tipo de dados.

Valores nulos



Problemas:

vários significados distintos para este valor: temporariamente vazio, inexistente, desconhecido, não se aplica

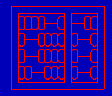
operações com esse valor em geral não fazem sentido,

o termo *nulo* é particularmente infeliz:

dá conotação errônea de “zero” para atributo numérico, ou de “cadeia vazia”, se atributo tipo cadeia de caracteres.

UNICAMP

Restrições de integridade



Restrições de integridade ou regras de consistência sobre valores de atributos

Regra de integridade existencial:

chave primária não pode tomar o valor nulo
razão óbvia: violaria definição de chave

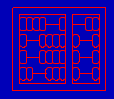
Regra de integridade referencial:

inter-relacionamento entre uma tabela mestre e outra dita detalhe ou auxiliar. Exemplo:

dados sobre dependentes de cada funcionário

“dependentes”: exemplo típico de atributo multivalorado, não suportado pelo MR

Regra de integridade referencial



Solução: tabela auxiliar:

Dependentes($\underline{numf, numdep}$, nomedep, parentesco, data_nasc)

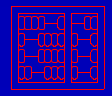
uma linha para cada dependente identificando funcionário e dependente

se existe linha em Dependentes onde Dependentes.numf = x deve existir em Funcionários uma linha onde Funcionários.numf = x

Definição: o atributo numf de Dependentes é uma chave estrangeira que "referencia" a chave primária numf da tabela Funcionários.

UNICAMP

Restrições sobre atributos



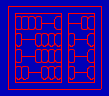
- atributo pode/não pode tomar o valor *nulo*;
- atributo deve satisfazer restrições de domínio: ≥ 10 e < 100.

formas para manter integridade referencial:

- abortar inserções e atualizações em tabela com chave estrangeira (como Dependentes) que violem a integridade referencial
- abortar remoção de tupla em tabela mestre referenciada por chave estrangeira;

UNICAMP

Restrições sobre atributos



Remoção em cascata:

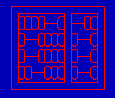
remoção de tupla em tabela mestre referenciada por chave estrangeira: remover todas as tuplas que a referenciam em tabelas auxiliares

Exemplo: se funcionário é removido, então todos os seus dependentes são removidos dde Dependentes

Todos os recursos acima estão disponíveis na linguagem SQL.

UNICAMP

Mapeamento do MER para o MR



O mapeamento de CEs é direto:

a cada CE corresponde uma relação com o mesmo nome e os mesmos atributos do CE,

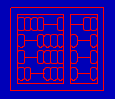
os tipos dos atributos (domínios) devem ser especificados

a chave determinante do CE passa a ser a chave primária da tabela.

chaves estrangeiras devem também ser especificadas, assim como outras restrições sobre atributos.

SQL: restrições definidas no esquema da relação

Mapeamento de relacionamentos



Relacionamentos N : 1

embutidos na relação do lado N do relacionamento como novo atributo (e que será chave estrangeira).

Exemplo: “Lotações”: departamento em que cada funcionário está lotado:

colocanr o número do departamento como atributo de Funcionários, referenciando a chave primária de Departamentos.

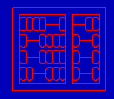
Atributos do relacionamento:

inserir como novos atributos de Funcionários:

Funcionários(numf, RG, CPF, nome, endereço, salário, num_depto, data_lotação)

UNICAMP

Mapeamento de relacionamento 1 : 1



Relacionamento entre “Correntistas” e “Contas_Correntes”

pode ser embutido em qualquer um dos lados,

natural dentro de Contas_Correntes, por ser um CE *fraco*, isto é, sua existência é dependente de Correntistas:

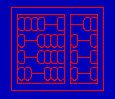
Correntistas(num_correntista, nome, end, . . .

Contas_Correntes(num_conta, num_correntista, limite, . . .),

onde *num_correntista* é uma chave estrangeira com a restrição *not null*, referenciando a chave primária de Correntistas.

UNICAMP

Auto-relacionamentos N : 1



Podem ser representados via chave estrangeira na própria tabela mestre

Exemplo, o auto-relacionamento Gerenciamentos entre Funcionários pode ser expresso assim:

em Funcionários(numf, RG, CPF, nome, endereço, salário, num_depto, num_gerente

num_depto not null referencia Departamentos.num_depto

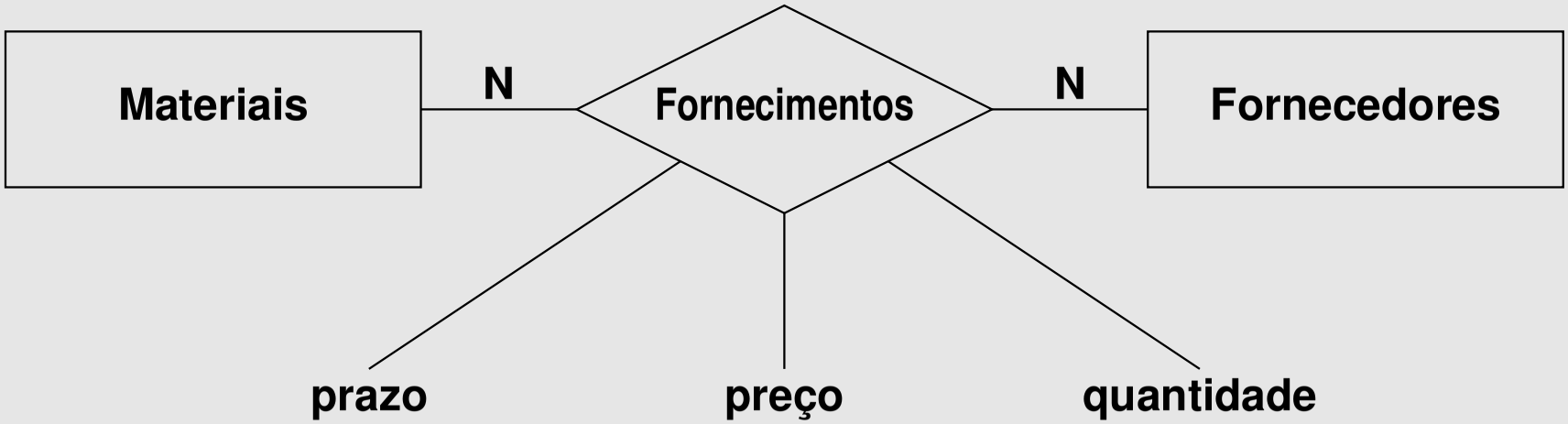
num_gerente referencia Funcionários.numf)

UNICAMP

Relacionamentos N : N

Devem ser representados via tabela auxiliar.

Ex: Fornecimentos entre Materiais e Fornecedores:

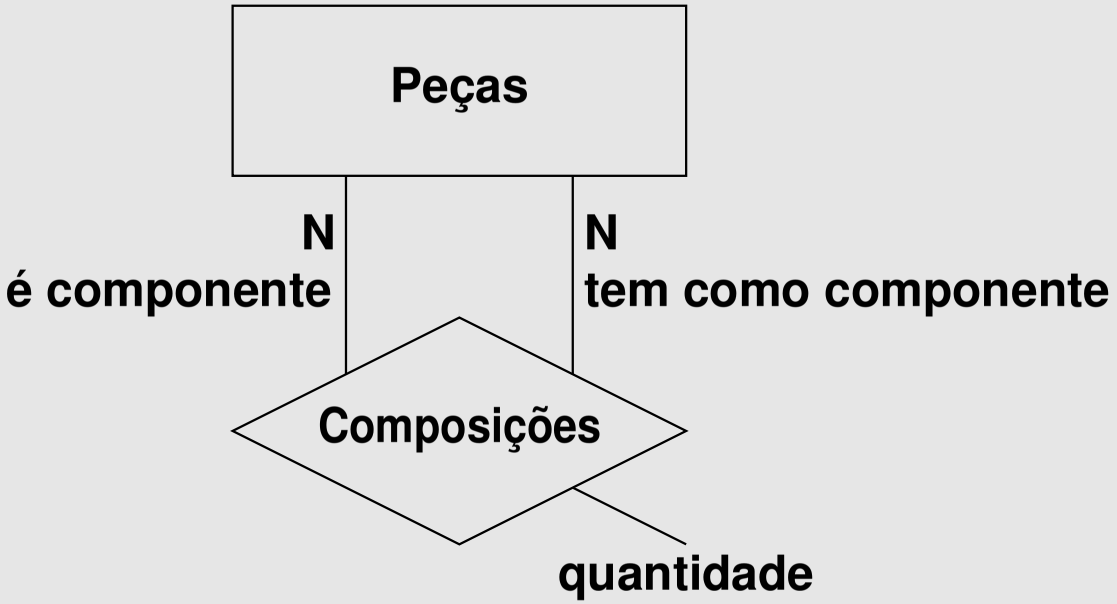


representado por tabela com esquema:

Fornecimentos(num_mat, num_fornecedor, prazo, preço, quantidade)

Auto-relacionamentos N : N

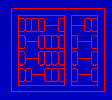
O auto-relacionamento N : N, “Composições” (mais preciamente, o papel *tem como componente*), na Figura 3.5:



será representado pela tabela:

Composições(num_peça,num_componente, quantidade)

Mapeamento de atributos multivalorados



Serão representados por uma tabela auxiliar

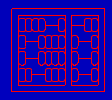
Exemplo: Dependentes de funcionários

Se um CE possui mais de um atributo multivalorado, deve-se criar uma tabela auxiliar para cada atributo multivalorado.

razões: quando estudarmos a 4ª forma normal

UNICAMP

Mapeamento de relacionamentos triplos



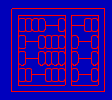
Também serão representados por tabelas auxiliares.

Exemplo: o relacionamento triplo PAD, visto na Figura 2.12, entre Professores, Alunos e Disciplinas, de classe 1 : N : N, seria representado pela relação:

PAD (num_aluno, num_disciplina, num_prof)

a chave primária é o par de atributos (num_aluno, num_disciplina), pois dado um aluno e uma disciplina só há um professor.

Mapeamento de relacionamentos triplos



O relacionamento triplo N : N : N entre Materiais, Requisições e Pedidos, visto na Figura 2.13, seria representado pelo esquema:

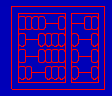
MRP (num_mat, num_req, num_pedido, quant_requisitada, quant_pedida)

Agora os três primeiros atributos constituem a chave primária,

os atributos do relacionamento passam a ser atributos da relação criada: simples e direto!

UNICAMP

Mapeamento de agregações



Exemplo: Figura 2.15 do capítulo anterior:

Materiais(num_mat,descrição,unidade, . . . , etc)

Requisições (num_req, emitente, data, local)

Pedidos(num_pedido, num_comprador, num_fornecedor, data)

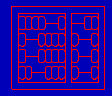
Itens_de_requisições(num_mat, num_req, quan_requisitada)

Itens_de_pedidos(num_pedido, num_mat, num_req, quant_pedida)

chaves primárias devem ficar claras ao analisarmos as classe dos relacionamentos, assim como as chaves estrangeiras

UNICAMP

Mapeamento de especialização



Exemplo: Figura 2.15:

Secretárias, Técnicos, Engenheiros, Gerentes são especializações do CE Funcionários

uma tabela auxiliar para cada:

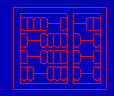
nome, chave primária = *numf*, ao mesmo tempo chave estrangeira referenciando a chave primária *numf* de Funcionários.

Funcionários(numf, RG, CPF, nome, endereço, salário)

Engenheiros(numf, especializ, anos-formado, . . . numf referencia Funcionários.numf)

UNICAMP

Mapeamento da BD Torneios de Tênis



MER da figura 2.14:

mapeamento praticamente direto:

Jogadores(numj, nome, pnome, país, ano_n, ano_p, cid_n, cid_res, tit_s, tit_d, vits, derrs, natp)

Lista_torneios(numt, nomet, país, cat, quadra, nparts)

Torneios(numt, ano, numj, prêmio)

relacionamento “Vencs_torneios” : representado pela relação Torneios

CE “Anos_realização” : desneceesário mapear (apenas seu atributo determinante)

UNICAMP