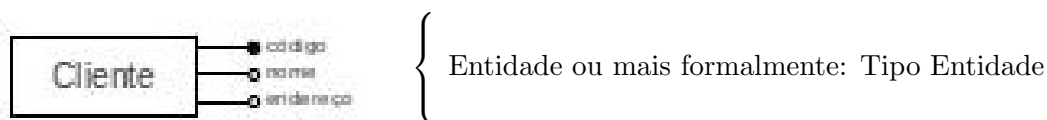


Universidade Federal da Fronteira Sul - UFFS
Campus Chapecó
Ciência da Computação
Banco de Dados I
Prof.: Denio Duarte

Notas de Aula - Modelo Relacional - Cap. 03 - Elmasri 6ª Edição

O modelo relacional representa um banco de dados (BD) como um conjunto de relações. Cada relação se assemelha a uma tabela de valores. Se enxergarmos uma relação como uma tabela e, baseado no modelo entidade relacionamento (ER), imaginarmos a relação representando uma entidade, temos uma linha de uma tabela representando um fato que corresponde a uma instância (ou uma ocorrência) de uma entidade (ou relacionamento).

Ex.:



Uma instância ou ocorrência da entidade Cliente poderia ser:

045 Fulano R. Joinville, 20

Na representação tabela seria:

Codigo	Nome	Ender
045	Fulano	R. Joinville, 20

Assim, no modelo relacional temos:

(Tipo) Entidade → Relação ou Tabela

Instância ou ocorrência de (tipo) entidade → Tupla ou Linha

Atributo → atributo ou coluna

Domínios, Atributos, Tuplas e Relações

- Um domínio \mathcal{D} é um conjunto de valores atômicos.

Ex.: um *int* é um domínio de inteiros.

CPF pode ter como domínio uma sequência de 11 dígitos de 0 à 9 (nnnnnnnnnnn)

- Um esquema de relação \mathcal{R} , indicado por $\mathcal{R}(A_1, A_2, \dots, A_n)$, é composto pelo nome da relação \mathcal{R} e um conjunto de atributos A_1, A_2, \dots, A_n . Cada atributo A_i ($0 < i \leq n$) é o nome de uma papel desempenhado por algum domínio \mathcal{D} em \mathcal{R} . \mathcal{D} é chamado de domínio de A_i e é indicado como $dom(A_i)$.

- O grau (ou aridade) de \mathcal{R} é o número de atributos de \mathcal{R} , denota-se $grau(\mathcal{R})$ ou $arity(\mathcal{R})$

- A cardinalidade de uma relação \mathcal{R} é o número de tuplas de \mathcal{R} , denotado por $card(\mathcal{R})$

Ex.: $Aluno(matric, cpf, nome, ender)$ ou $Aluno(matric : string, cpf : long int, nome : string, ender : string)$ representa o esquema da relação (ou tabela) *Aluno* que possui 4 atributos: *matric*, *cpf*, *nome* e *ender*. A primeira representação, que será utilizada neste curso, não indica o domínio de cada atributo, já a segunda indica.

$grau(Aluno) = 4$ ou $arity(Aluno) = 4$

- Uma relação (ou instância da relação) r do esquema de relação $\mathcal{R}(A_1, A_2, \dots, A_n)$, indicado por $r(\mathcal{R})$, e o conjunto de n -tuplas $r = \{t_1, t_2, \dots, t_n\}$. Cada n -tupla t é uma lista ordenada de n valores ($grau(\mathcal{R})$) $t = \langle v_1, v_2, \dots, v_n \rangle$ onde cada valor v_i ($0 < i \leq n$) é um elemento de $dom(A_i)$ ou um valor nulo (*null*) - valor especial que indica valor inexistente/vazio. O i -ésimo

valor de t , que corresponde ao atributo A_i , e denotado por $t[A_i]$ (ou $t[i]$ se utilizarmos a notação sem nome ou posicional).

Ex.: $t_2 = \langle 046, 66677733344, Ciclano, R.Itajaí \rangle$, $t_2[nome] = Ciclano$, $t_2[1] = 046$

A tabela a seguir representa uma instância da relação *Aluno* cuja cardinalidade é 3 ($card(Aluno) = 3$), ou seja, possui 3 tuplas (ou linhas).

Aluno			
matric	cpf	nome	ender
045	11122233344	Fulano	R. Joinville, 20
046	66677733344	Ciclano	R. Itajaí, 120
047	77788833344	Beltrano	R. Gaspar, 220

Formalmente

A instância da relação $r(\mathcal{R})$ é uma relação matemática de grau n nos domínios $dom(A_1)$, $dom(A_2)$, ..., $dom(A_n)$ que é um subconjunto do produto cartesiano dos domínios que definem \mathcal{R} :

$$r(\mathcal{R}) \subseteq (dom(A_1) \times dom(A_2) \times \dots \times dom(A_n))$$

Produto cartesiano: todas as possíveis combinações dos domínios.

- O produto cartesiano acima indica todas as combinações possíveis do $r(\mathcal{R})$.
- O estado da relação corrente reflete apenas as tuplas válidas (subconjunto do produto cartesiano dos domínios) que representam um estado particular do mundo real de \mathcal{R}
- É possível que várias atributos tenham o mesmo domínio. Os nomes dos atributos, então, indicam diferentes papéis ou interpretações do domínio. Ex.: o atributo *idade* e *num_amigos* pertencem ao domínio dos inteiros positivos mas o papel do primeiro é indicar a idade de uma determinada entidade e o segundo o número de amigos que uma determinada entidade tem.
- As tuplas em uma relação não tem ordem. Porém, fisicamente, uma tabela tem ordem pois as tuplas (linhas) estão armazenadas no disco.

Notação

- Um esquema de uma relação \mathcal{R} de grau n é indicado por $\mathcal{R}(A_1, A_2, \dots, A_n)$
- As letras maiúsculas Q , R e S indicam nomes de relação.
- As letras minúsculas q , r e s indicam estados de relação.
- As letras t , u , v indicam tuplas.
- Quando for utilizado o nome da relação, por exemplo *ALUNO*, também indicará o conjunto atual de tuplas nessa relação (ou seja, o *estado de relação atual*, enquanto *ALUNO* (*Nome*, *CPF*, ...) se refere apenas ao esquema de relação.
- Um atributo A pode ser qualificado como o nome da relação R a qual pertence utilizando a notação $R.A$. Isso pois duas relação diferentes podem ter nome de atributos iguais.
- Uma n -tupla t em uma relação $r(\mathcal{R})$ é indicado por $t = \langle v_1, v_2, \dots, v_n \rangle$, onde v_i é o valor correspondente ao atributo A_i . Os valores componentes de uma tupla podem ser representados da seguinte maneira:
 - Tanto $t[A_i]$ quanto $t.A_i$ (e às vezes $t[i]$) se referem ao valor v_i em t para o atributo A_i .

- Tanto $t[A_u, A_w, \dots, A_z]$ quanto $t.(A_u, A_w, \dots, A_z)$, onde A_u, A_w, \dots, A_z é uma lista de atributos de R , se referem à sub tupla de valores $\langle v_u, v_w, \dots, v_z \rangle$ de t correspondentes aos atributos especificados na lista.
- Ex.: dada a tupla $t = \langle 045, 11122233344, \text{'Fulano'}, \text{'R. Joinville, 20'} \rangle$ da relação *CLIENTE* (apresentada anteriormente), temos $t[Nome] = \langle \text{'Fulano'} \rangle$ e $t[Codigo, Ender] = \langle 045, \text{'R. Joinville, 20'} \rangle$

Restrições

Restrições são derivadas das regras do minimundo que o banco de dados representa. Representam alguma restrição para o armazenamento e manutenção dos dados. O banco de dados que respeita as restrições impostas em um determinado estado é dito que tal banco está em um *estado válido*. Principais tipos de restrições:

- Restrições implícitas (modeladas no próprio modelo de dados).
- Restrições explícitas (expressas no próprio esquema da relação).
- Regras de negócio (restrições implementadas pela aplicação).
- Dependências funcionais (restrições de dependência de dados).

Restrições baseadas em esquema: restrições de domínio (domain constraints), restrições de chaves (key constraints), restrições sobre valores *nulos* (null constraint), restrições de integridade de entidade (integrity entity constraint) e restrições de integridade referencial (referential integrity constraint).

- Domínio: dentro de cada tupla, o valor de cada atributo A deve ser um valor indivisível de $dom(A)$.
- Valores *nulo*: indica se um valor *NULL* pode ser permitido para um atributo ou não. Geralmente, os atributos opcionais de uma relação aceita valores *nulos*.
- Chave (key): uma relação é um conjunto de tuplas e, neste contexto, as tuplas são distintas entre si. Isso significa que duas tuplas não podem ter a mesma combinação de valores para todos os seus atributos. Normalmente, existe um subconjunto de atributos de um esquema de relação R com a propriedade de que duas tuplas em qualquer estado r de R não deverão ter a mesma combinação de valores. Suponha que SSa é um subconjunto de atributos da relação R que respeitam a restrição de chave, assim, não existirão duas tuplas t e u tal que $t[SSa] = u[SSa]$, ou seja, para qualquer estado r de R e dadas duas tuplas quaisquer t e u , a propriedade $t[SSa] \neq u[SSa]$ deve ser respeitada. O subconjunto SSa é chamado de superchave (superkey) de R . Por definição, todos os atributos de uma relação R compõem a superchave de R . Porém, dado o conjunto de atributos de R chamado de $ABCD$, se encontramos um subconjunto de atributos ABC tal que para qualquer t e u a propriedade $t[ABC] \neq u[ABC]$, ABC é uma superchave menor que $ABCD$. Esse raciocínio pode ser continuado até encontrarmos um subconjunto onde a propriedade acima não é mais respeitada. Neste caso, o último subconjunto (que pode ser unitário) que respeitou a propriedade de distinção é chamado de *chave* (key) ou *superchave mínima* (minimal superkey). Em uma relação pode haver várias *chaves* porém apenas uma delas deve ser escolhida para representar unicamente uma tupla, esta chave será chamada de *chave primária* (primary key) e as outras de *chaves candidatas* (candidate key). Observe que uma *chave* deve valer para qualquer estado em qualquer momento de uma relação R . Por exemplo, o atributo *Nome* da relação cliente pode se configurar com *chave* em um

determinado estado da relação *CLIENTE* porém, pelo conhecimento anterior, é sabido que em um momento futuro, este atributo poderá ter valores repetidos na relação.

As restrições vistas anteriormente são aplicadas dentro de uma relação. A seguir, apresentaremos as restrições que ocorrem interrelação. Para tal, definiremos banco de dados relacional e esquema de banco de dados relacional.

Um esquema de banco de dados relacional S é um conjunto de esquemas de relação $S = \{R_1, R_2, \dots, R_n\}$ e um conjunto de restrições de integridade (RI) - integrity constraints (IC). Um estado de banco de dados relacional DB de S é um conjunto de estados de relação $DB = \{r_1, r_2, \dots, r_m\}$, tal que cada r_i é um estado de R_i e tal que os estados da relação r_i satisfaçam as restrições de integridade especificadas em RI . Um estado de um banco de dados relacional que obedece às RI é chamado de *estado válido* (estado consistente), caso contrário é chamado de *estado inválido* (estado inconsistente).

Integridade referencial: essa restrição é especificada entre duas relações e é utilizada para manter a consistência entre tuplas de duas relações. Informalmente, a restrição de integridade referencial afirma que uma tupla t em uma relação R que referencia outra relação S precisa se referir a uma *tupla existente* em S .

A integridade referencial é implementada com o conceito de *chave estrangeira* (foreign key). Uma chave estrangeira é uma restrição de integridade entre dois esquemas de relação R_1 e R_2 . Um conjunto de atributos FK de R_1 (tal que $FK \subseteq R_1$) é uma chave estrangeira de R_1 que referencia R_2 , se e somente se:

1. FK tem os mesmo domínio (ou domínios) dos atributos que FK referencia em R_2 . É dito que FK referencia ou refere-se à relação R_2 .
2. FK na tupla t_1 do estado corrente $r_1(R_1)$ ou
 - (a) Casa com um conjunto de atributos PK (possivelmente a chave primária) de alguma tupla t' de um estado corrente $r_2(R_2)$ ou
 - (b) É nulo.

Dados esquemas de relação $Aluno(mat, nome, ender, codcurso)$ e $Curso(cod, nome, carga)$ sendo mat e cod as chaves primárias de $Aluno$ e $Curso$, respectivamente, e $codcurso$ a chave estrangeira em $Aluno$ que referencia cod em $Curso$. Abaixo dois exemplos de estados do banco de dados dessas relações.

Instância válida

Aluno				Curso		
mat	nome	ender	codcurso	cod	nome	carga
0123	João da Silva	R. XII, 10	5	1	Corte e Costura	2900
0124	Maria Ia	R. Xis, 110	1	5	Ballet Clássico	1000
0130	Julieta	R. Poá, 40	5			
0142	Pedro	R. Pia, 320	5			

Instância inválida

Aluno				Curso		
mat	nome	ender	codcurso	cod	nome	carga
0123	João da Silva	R. XII, 10	5	1	Corte e Costura	2900
0124	Maria Ia	R. Xis, 110	1	5	Ballet Clássico	1000
0130	Julieta	R. Poá, 40	15			
0142	Pedro	R. Pia, 320	5			

Notação utilizada em sala de aula para representar os esquemas de relação:

- Atributos obrigatórios: coloca-se apenas o nome sem sinais. particulares
- Atributos opcionais: sublinha-se com uma linha tracejada.
- Chave primária: sublinha-se com uma linha contínua.
- chave estrangeira: Colocar, entre parênteses, a relação referenciada.

Acertando a nomenclatura das relações *Aluno* e *Curso* conforme as regras acima:

Aluno(mat, nome, _e_n_d_e_r_, codcurso(*Curso*))

Curso(cod, nome, carga)