

Capítulo 3: Modelo Relacional

Algebra Racional

OPERAÇÃO	DESCRIÇÃO
<u>Seleccção</u> $\sigma p(r)$	Definida como: $\sigma p(r) = \{t \mid t \in r \text{ e } p(t)\}$, onde p é uma formula do calculo proposicional constituída por termos ligados por: $\wedge$ (e), $\vee$ (ou), $\neg$ (não) p é designado por predicado de selecção. Cada termo é da forma: <atributo> op <atributo> ou <constante>, onde op pode ser: =, $\neq$, >, $\geq$, <, $\leq$ Exemplo de selecção: $\sigma \text{branch-name} = \text{'Perryridge'} (\text{account})$
<u>Projecção</u> $\pi A_1, A_2, \dots, A_k$	$\pi A_1, A_2, \dots, A_k$, onde $A_1, \dots, A_k$ são nomes de atributos e r é uma relação. O resultado é a relação com as k colunas obtidas eliminando as colunas que não estão listadas. São removidas as linhas duplicadas no resultado, dado que as relações são conjuntos. E.g. eliminar o atributo branch-name de account: $\pi \text{account-number, balance} (\text{account})$
<u>União</u> $r \cup s$	Definida como: $r \cup s = \{t \mid t \in r \text{ ou } t \in s\}$ Para $r \cup s$ ser valida: - r, s devem ter a mesma aridade (igual número de atributos) - Os domínios dos atributos têm de ser compatíveis (e.g., os valores da 2a coluna de r são do mesmo tipo dos valores da 2a coluna de s) E.g. determinar quais os clientes que tem uma conta ou um empréstimo: $\pi \text{customer-name} (\text{depositor}) \cup \pi \text{customer-name} (\text{borrower})$
<u>Diferença de Conjuntos</u> $r - s$	Definida como: $r - s = \{t \mid t \in r \text{ e } t \notin s\}$ As diferenças de conjuntos só podem ser efectuadas entre relações compatíveis. - r e s devem ter a mesma aridade - os domínios dos atributos de r e s devem ser compatíveis
<u>Produto Cartesiano</u> $r \times s$	Definida como: $r \times s = \{t \mid t \in r \text{ e } q \in s\}$ Assume que os atributos de $r(R)$ e $s(S)$ são disjuntos. (Ou seja, $R \cap S = \emptyset$). Se os atributos de $r(R)$ e $s(S)$ não são disjuntos, então tem que se utilizar renomeações.
<u>Composição de Operações</u>	Pode-se construir expressões combinando várias operações. Exemplo: $\sigma A = C(r \times s)$
<u>Renomeação</u> $\rho x (E)$	A expressão: $\rho x (E)$, devolve a expressão E com o nome X. Permite que uma relação seja referida por mais de um nome. Se uma expressão de algebra relacional E tem aridade n, então $\rho x (A_1, A_2, \dots, A_n) (E)$ devolve a expressão E com o nome X, e com os atributos renomeados para $A_1, A_2, \dots, A_n$.
<u>Intersecção de Conjuntos</u> $r \cap s$	Definida por: $r \cap s = \{t \mid t \in r \text{ e } t \in s\}$ Assume: r, s tem a mesma aridade - Os atributos de r e s são compatíveis Notar que: $r \cap s = r - (r - s) = s - (s - r)$
<u>Junção Natural</u> $r \bowtie s$	Sejam r e s relações nos esquemas R e S respectivamente. O resultado é uma relação no esquema $R \cup S$ que é obtido considerando cada par de tuplos t_r de r e t_s de s. Se t_r e t_s tem o mesmo valor em cada um dos atributos em $R \cap S$, um tuplo t é adicionado ao resultado, em que: t tem os mesmos valores que t_r em r t tem os mesmos valores que t_s em s Exemplo: $R = (A, B, C, D) \quad S = (E, B, D)$

	Esquema resultado: (A, B, C, D, E) $r \bowtie s$ define-se por: $\prod_{r:A, r:B, r:C, r:D, s:E} (\sigma_{r:B=s:B \wedge r:D=s:D} (r \times s))$
<u>Divisão</u> $r \div s$	Adequada para consultas que incluam a frase “para todo”. Sejam r e s relações nos esquemas R e S respectivamente com $R = (A1, \dots, Am, B1, \dots, Bn)$ $S = (B1, \dots, Bn)$ O resultado de $r \div s$ é uma relação no esquema $R - S = (A1, \dots, Am)$ e $r \div s = \{ t \mid t \in \prod R - S(r) \wedge \forall u \in s (tu \in r) \}, \text{ onde } tu \text{ é o tuplo resultante da concatenação dos tuplos } t \text{ e } u.$ Propriedade: Seja $q = r \div s$ Então q é a maior relação satisfazendo $q \times s \subseteq r$ Definição em termos de operações básicas da algebra real: Sejam $r(R)$ e $s(S)$ relações, com $S \subseteq R$ $r \div s = \prod_{R-S}(r) - \prod_{R-S}((\prod_{R-S}(r) \times s) - \prod_{R-S,S}(r))$ Porquê? $\prod_{R-S}(r) \times s$ da os elementos de r com todos os valores de S $\prod_{R-S,S}(r)$ reordena os atributos de r $\prod_{R-S}(\prod_{R-S}(r) \times s) - \prod_{R-S,S}(r)$ da os tuplos t em $\prod_{R-S}(r)$ tal que para algum tuplo $u \in s$, $tu \notin r$.
<u>Atribuição</u> $\leftarrow$	A operação de atribuição ($\leftarrow$) permite-nos expressar consultas complexas de uma forma muito conveniente. Escreve-se a consulta como um programa sequencial constituído por uma sequência de atribuições terminada com uma expressão cujo valor é o resultado da consulta. A atribuição é sempre efectuada para uma variavel de relação temporaria. Exemplo: escrever $r \div s$ como $\begin{aligned} \text{temp1} &\leftarrow \prod_{R-S}(r) \\ \text{temp2} &\leftarrow \prod_{R-S}((\text{temp1} \times s) - \prod_{R-S,S}(r)) \\ \text{resultado} &\leftarrow \text{temp1} - \text{temp2} \end{aligned}$ O resultado à direita de $\leftarrow$ é atribuído à variável que se encontra à esquerda de $\leftarrow$. Pode-se utilizar a variável nas expressões seguintes.
<u>Propriedades</u>	A selecção é comutativa: $\sigma_{p1}(\sigma_{p2}(r)) = \sigma_{p2}(\sigma_{p1}(r))$ A projecção é comutativa com a selecção: $\begin{aligned} \prod_X(\sigma_{a=v}(r)) &= \sigma_{a=v}(\prod_X(r)) && \text{se } a \in X \\ \prod_X(\prod_Y(r)) &= \prod_X(r) && \text{se } X \subseteq Y \end{aligned}$ A junção é comutativa e associativa. Outras: Se $R \cap S = \{ \}$ então $r(R) \bowtie s(S) = r(R) \times s(S)$ $s \bowtie (\sigma_{a=v}(r)) = \sigma_{a=v}(s \bowtie r)$ $(r \cup r') \bowtie s = (r \bowtie s) \cup (r' \bowtie s)$ $\sigma_{a=v}(r \text{ op } s) = \sigma_{a=v}(r) \text{ op } \sigma_{a=v}(s)$ para $\text{op} \in \{ \cup, \cap, - \}$
<u>Projecção Generalizada</u>	Permite a utilização de funções aritmeticas na lista de projecção. $\prod F1, F2, \dots, Fn(E)$ E é uma expressão arbitrária de algebra relacional. Cada uma das expressões $F1, F2, \dots, Fn$ é uma expressão aritmetica envolvendo constantes e atributos no esquema de E. Dada a relação credit-info(customer-name, limit, credit-balance), encontrar quanto cada cliente pode ainda gastar: $\prod \text{customer-name, limit} - \text{credit-balance}(\text{credit-info})$ Ha’ quantos dias foi cada uma das consultas: $\prod \text{InConsulta, today} - \text{data}(\text{consultas})$

<u>Funções de Agregação e Operações</u>	Funções de Agregação aplicam-se a uma colecção de valores e devolvem um único valor como resultado: avg: média dos valores min: mínimo dos valores max: máximo dos valores sum: soma dos valores count: número dos valores Operação de Agregação na álgebra relacional $G_1, G_2, \dots, G_n \quad G \quad F_1(A_1), F_2(A_2), \dots, F_n(A_n) (E)$ E - uma expressão de algebra relacional; G1, G2 ..., Gn - uma lista de atributos de agrupamento (pode ser vazia) Cada Fi - uma função de agregação Cada Ai - um nome de um atributo O resultado da agregação não tem um nome: Pode-se recorrer à operação de renomeação para lhe dar um nome; Por conveniência, permite-se a renomeação de atributos na operação de agregação: branch-name $G \quad sum(balance) \quad as \quad sum_balance \quad (account)$ Qual a media de idades dos pacientes de cada um dos medicos? $nEmpr \quad G \quad avg(idade) \quad as \quad media \quad (\prod nB1, nEmpr, idade(consultas \mid x \mid pacientes))$
<u>Junção Externa (ou exterior)</u>	Uma extensão da operação de junção que evita a perda de informação. Calcula a junção e depois adiciona ao resultado os tuplos de uma relação que não estão relacionados com a outra relação najunção. Utiliza valores nulos: null significa que o valor é desconhecido ou que não existe; Simplificadamente, todas as comparações com null são falsas por definição.
<u>Valores Nulos</u>	- É possível que um tuplo tenha um valor nulo, denotado por null, para algum dos seus atributos. - null significa um valor desconhecido ou que não existe. - O resultado de qualquer expressão aritmetica envolvendo um null é null. - As funções de agregação ignoram os valores nulos: - Na eliminação de duplicados e agrupamento, um null é tratado como um outro valor qualquer, assumindo-se que dois nulls são o mesmo - Comparações com valores nulos devolvem o valor de verdade unknown. Se false fosse usado em vez unknown, então $not (A < 5)$ não seria equivalente a $A \geq 5$ - Logica a três valores com o valor de verdade unknown: - OR: (unknown or true) = true, (unknown or false) = unknown (unknown or unknown) = unknown - AND: (true and unknown) = unknown, (false and unknown) = false, (unknown and unknown) = unknown - NOT: (not unknown) = unknown - Em SQL “P is unknown” é verdade se o predicado P tem valor de verdade unknown. - Resultado do predicado de selecção é tratado como false se tiver valor de verdade unknown.

Modificação da Base de Dados

O conteúdo da base de dados pode ser modificado através das seguintes operações:

Remoção

Inserção

Actualização

Todas estas operações são expressas por intermédio do operador de atribuição.

<u>Remoção</u>	
Uma operação de remoção é expressa de uma maneira semelhante a uma consulta, excepto que os tuplos seleccionados são removidos da base de dados. So se podem remover tuplos integralmente; não se podem apagar valores de determinados atributos Uma remoção é expressa em algebra relacional por: $r \leftarrow r - E$ em que r é uma relação e E é uma operação de algebra relacional.	Exemplos: Apagar todas as contas na agência de Perryridge. $\text{account} \leftarrow \text{account} - \sigma_{\text{branch-name} = \text{'Perryridge'}}(\text{account})$ Apagar todos os registos de empréstimos de montante entre 0 e 50 $\text{loan} \leftarrow \text{loan} - \sigma_{\text{amount} \geq 0 \text{ and } \text{amount} \leq 50}(\text{loan})$ Apagar todas as contas de balcões localizados em Needham. $r1 \leftarrow \sigma_{\text{branch-city} = \text{'Needham'}}(\text{account } 1 \text{ branch})$ $r2 \leftarrow \prod_{\text{branch-name, account-number, balance}}(r1)$ $r3 \leftarrow \prod_{\text{customer-name, account-number}}(r2 \text{ } 1 \text{ depositor})$ $\text{account} \leftarrow \text{account} - r2$ $\text{depositor} \leftarrow \text{depositor} - r3$ Apagar toda a informação relativa a consultas anteriores a 2000: $r1 \leftarrow \sigma_{\text{data} < 01-01-2000}(\text{consultas})$ $r2 \leftarrow \prod_{\text{codF, nConsulta, quant}}(\text{receitas } 1 \text{ } r1)$ $\text{consultas} \leftarrow \text{consultas} - r1$ $\text{receitas} \leftarrow \text{receitas} - r2$
<u>Inserção</u>	
Para inserir informação numa relação podemos: especificar um tuplo a ser inserido escrever uma consulta cujo resultado é um conjunto de tuplos a inserir Na algebra relacional, uma inserção é expressa por: $r \leftarrow r \cup E$ em que r é uma relação e E é uma expr. de algebra relacional. A inserção de um único tuplo é efectuada quando a expressão E é a relação constante contendo esse tuplo.	Exemplos: Inserir informação na base de dados especificando que o cliente Smith tem \$1200 na conta A-973 na agência de Perryridge. $\text{account} \leftarrow \text{account} \cup \{(\text{'Perryridge'}, \text{A-973}, 1200)\}$ $\text{depositor} \leftarrow \text{depositor} \cup \{(\text{'Smith'}, \text{A-973})\}$ Dar um bonus a todos os mutuários na agência de Perryridge: uma conta de poupança de \$200. O número do empréstimo é utilizado para número da conta de poupança. $r1 \leftarrow (\sigma_{\text{branch-name} = \text{'Perryridge'}}(\text{borrower } 1 \text{ loan}))$ $\text{account} \leftarrow \text{account} \cup \prod_{\text{branch-name, account-number, 200}}(r1)$ $\text{depositor} \leftarrow \text{depositor} \cup \prod_{\text{customer-name, loan-number}}(r1)$ Inserir informação na base de dados especificando que um novo paciente, com BI nº 10000000 e nome Paulo, teve uma consulta (nº1000) no dia 30-09-2003 com o médico João (assumindo que só há um médico com esse nome) $\text{pacientes} \leftarrow \text{pacientes} \cup \{(10000000, \text{'Paulo'}, \text{null}, \text{null}, \text{null})\}$ $\text{consultas} \leftarrow \text{consultas} \cup \prod_{1000, 30-09-2003, 1000, nEmpr}(\sigma_{\text{nomeM} = \text{'João'}}(\text{medicos}))$
<u>Actualização</u>	
Um mecanismo para alterar um valor de um tuplo sem alterar todos os valores do tuplo. Recorre-se ao operador de projecção generalizada para efectuar este tipo de tarefa $r \leftarrow \prod_{F1, F2, \dots, Fi}(r)$ Cada F_i, ou é o i-ésimo atributo de r, se o i-ésimo atributo não for alterado, ou uma expressão F_i, envolvendo apenas constantes e atributos de r, que permite calcular o novo valor do atributo.	Exemplos: Pague juros de 5% em todas as contas: $\text{account} \leftarrow \prod_{\text{AN, BN, BAL}}(\text{BAL} * 1.05(\text{account}))$ em que AN, BN e BAL são abreviaturas para account-number, branch-name e balance, respectivamente. Pague 6% de juros em todas as contas com saldo superior a \$10,000 e 5% às restantes contas. $\text{account} \leftarrow \prod_{\text{AN, BN, BAL}}(\sigma_{\text{BAL} > 10000}(\text{account})) \cup \prod_{\text{AN, BN, BAL}}(\sigma_{\text{BAL} \leq 10000}(\text{account}))$