

Modelo Relacional

Database System Concepts, 5th Ed.
©Silberschatz, Korth and Sudarshan

20018/2019

- Estrutura das Bases de Dados Relacionais
- Operações fundamentais da Álgebra Relacional
- Operações adicionais da Álgebra Relacional
- Operações estendidas da Álgebra Relacional
- Valores Nulos
- Modificação da base de dados

Estrutura das Bases de Dados Relacionais

Exemplo de uma Relação no domínio da Banca:

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

- *account_number* - numero da conta
- *branch_name* - nome da agência
- *balance* - saldo

Uma **relação** é um conjunto de tuplos de aridade n ,
 (a_1, a_2, \dots, a_n) com $a_i \in D_i$

- Dados os conjuntos D_1, D_2, \dots, D_n , a **relação** r é um subconjunto de $D_1 \times D_2 \times \dots \times D_n$
- $r \subseteq D_1 \times D_2 \times \dots \times D_n$

Estrutura básica das Bases de Dados Relacionais

Exemplo:

- $\text{customer_name} = \{\text{Jones, Smith, Curry, Lindsay, ...}\}$ /* Conjunto de todos os nomes de clientes */
- $\text{customer_street} = \{\text{Main, North, Park, ...}\}$ /* Conjunto de todas as ruas de clientes*/
- $\text{customer_city} = \{\text{Harrison, Rye, Pittsfield, ...}\}$ /* Conjunto de todas as cidades de clientes */

$\text{customer} = \{$
 (Jones, Main, Harrison),
 (Smith, North, Rye),
 (Curry, North, Rye), (Lindsay, Park, Pittsfield) $\}$

customer é uma relação de aridade 3 sobre *customer_name* x *customer_street* x *customer_city*

Tipos de Atributos

- Cada atributo de uma relação tem um nome
- Ao conjunto dos valores permitidos para cada atributo chama-se domínio do atributo
- Os valores dos atributos são normalmente atómicos; isto é, indivisíveis
E.g. O valor de um atributo pode ser o número de uma conta (account number), mas não pode ser um conjunto de números de conta
- O domínio é atómico se todos os seus membros são atómicos
- O valor especial null (nulo) é membro de todos os domínios
- O valor null complica a definição de muitas operações
Na apresentação inicial dos operadores vamos ignorar os casos em que o valor é null

Esquema de uma relação

- A_1, A_2, \dots, A_n são atributos
- $R = (A_1, A_2, \dots, A_n)$ é o esquema da relação

Exemplo:

Customer_schema =

(customer_name, customer_street, customer_city)

- $r(R)$ denota a relação r no esquema de relação R

Exemplo:

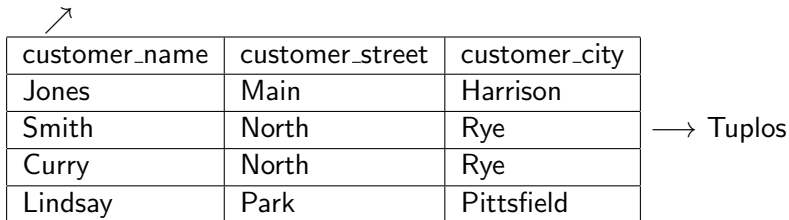
customer (Customer_schema)

Estrutura básica das Bases de Dados Relacionais

Instância de uma relação

- Os valores actuais (instância da relação) da relação são especificados numa tabela
- Um elemento t de r é um tuplo, representado numa linha da tabela

Atributos



The diagram illustrates the relationship between attributes and tuples in a relational database. An arrow points from the word 'Atributos' to the first column of a table. The table has three columns: 'customer_name', 'customer_street', and 'customer_city'. To the right of the table, an arrow points from the word 'Tuplos' to the first row of the table, which contains the values 'Jones', 'Main', and 'Harrison'.

customer_name	customer_street	customer_city
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

Estrutura básica das Bases de Dados Relacionais

As relações não têm ordem

A ordem dos tuplos não é relevante (os tuplos podem ser guardados por uma ordem arbitrária)

Exemplo:

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

Estrutura básica das Bases de Dados Relacionais

Base de Dados relacional

Uma base de dados é um conjunto de relações

A informação de uma empresa é separada em várias partes e cada relação guarda uma parte da informação

account : (conta) guarda a informação sobre contas

depositor : (depositante) guarda a informação sobre que cliente tem q

customer : (cliente) guarda a informação sobre clientes

Guardar toda a informação numa única relação como por exemplo:

bank(account_number, balance, customer_name, ..) resulta na:

- Repetição de informação
e.g., se dois clientes têm a mesma conta (o que é que se repete?)
- Necessidade de usar valores null
e.g., para representar um cliente que não tem conta

No capítulo 7 vamos lidar com o desenho de esquemas relacionais usando a teorias de normalização.

Estrutura básica das Bases de Dados Relacionais

A relação cliente (customer)

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Estrutura básica das Bases de Dados Relacionais

A relação depositante (depositor)

<i>customer_name</i>	<i>account_number</i>
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

Chaves

Seja K um conjunto de atributos do esquema R , $K \in R$

- K é uma **super chave** de R se os valores de K são suficientes para identificar um tuplo único de cada relação possível $r(R)$

“relação possível r ” é a relação r que poderá existir na empresa que estamos a modelar.

Exemplo:

$\{\text{customer_name}, \text{customer_street}\}$ e $\{\text{customer_name}\}$ são ambas super chaves de Customer se não é possível que dois clientes tenham o mesmo nome

Na vida real, seria necessário usar um atributo `customer_id` em vez de `customer_name` para identificar univocamente um cliente, mas nos exemplos para ficarem mais pequenos assumimos que os nomes dos clientes são únicos.

K é uma chave candidata se K é minimal

Exemplo:

{customer_name} é uma chave candidata para Customer, pois é uma super chave e não tem nenhum subconjunto que seja super chave.

Chave primária: uma chave candidata escolhida como a forma principal para identificar os tuplos de uma relação

Deve-se escolher uma atributo que nunca ou raramente muda. E.g. o endereço de email é único, mas pode mudar, o CC não muda.

Estrutura básica das Bases de Dados Relacionais

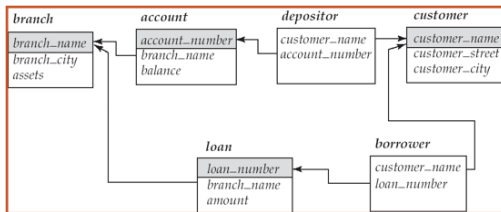
Chaves estrangeiras

O esquema de uma relação pode ter um atributo que é a chave primária de outra relação. A este atributo nesta relação chama-se chave estrangeira.

E.g. os atributos `customer_name` e `account_number` de `depositor` são chaves estrangeiras de `customer` e `account` respectivamente.

Só os valores que ocorrem no atributo chave primária da relação referenciada podem ocorrer no atributo chave estrangeira da relação que referencia.

Diagrama do esquema



Linguagem de interrogação (Query Language)

- Linguagem em que o utilizador pede a informação à base de dados.
- Categorias das Linguagens
 - Procedimental
 - Não-procedimental, ou declarativa
- Linguagens “Puras”:
 - **Álgebra Relacional**
 - Tuple relational calculus
 - Domain relational calculus

As linguagens puras estão na origem das linguagens de interrogação implementadas e usadas nos sistemas de gestão de bases de dados

Seis operadores básicos:

- Selecciona (select): σ
- Projecta (project): π
- União (union): \cup
- Diferença conjuntos (set difference): $-$
- Produto cartesiano (Cartesian product): \times
- Renomear (rename): ρ

Os operadores tem uma ou duas relações como argumentos e retornam uma nova relação como resultado.

Álgebra Relacional - Operação seleciona (select) σ

Operação seleciona (select) σ

Exemplo:

$r =$

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

$\sigma_{A=B \wedge D > 5}(r) =$

A	B	C	D
α	α	1	7
β	β	23	10

Álgebra Relacional - Operação seleciona (select) σ

Operação seleciona (select) σ

- Notação: $\sigma_p(r)$
- A p chama-se o predicado de selecção
- $\sigma_p(r) = \{t \mid t \in r \wedge p(t)\}$
 p - é uma fórmula do cálculo de proposições que consiste em termos ligados com : \wedge (e), \vee (ou), \neg (negação)
Cada termo é da forma:
 $\langle \text{atributo} \rangle \text{ op } \langle \text{atributo} \rangle \text{ ou } \langle \text{constante} \rangle$
onde op é: $=, <, >, \leq \text{ ou } \geq$

Exemplo de uma selecção:

$\sigma_{\text{branch_name}=\text{Perryridge}}(\text{account})$

Operação projeção (project) π

Exemplo:

$r =$

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$\pi_{A,C}(r) =$

A	C
α	1
α	1
β	1
β	2

$=$

A	C
α	1
β	1
β	2

Operação projeção (project) π

- Notação: $\pi_{A_1, A_2, \dots, A_k}(r)$
onde A_1, A_2, \dots são nome de atributos de r
- O resultado é uma relação de k colunas que se obtém eliminando as colunas que não foram listadas
As linhas duplicadas são removidas, uma vez que as relações são conjuntos

Exemplo: Para eliminar o atributo *branch_name* da relação conta
 $\pi_{account_number, balance}(account)$

Álgebra Relacional - Operação união (union) \cup

Operação união (union) \cup

Exemplo:

$r=$	A	B	$s=$	A	B
	α	1		α	2
	α	2		β	3
	β	1			

$r \cup s =$	<table><tr><th>A</th><th>C</th></tr><tr><td>α</td><td>1</td></tr><tr><td>α</td><td>2</td></tr><tr><td>β</td><td>1</td></tr><tr><td>β</td><td>3</td></tr></table>	A	C	α	1	α	2	β	1	β	3
A	C										
α	1										
α	2										
β	1										
β	3										

Álgebra Relacional - Operação união (union) \cup

Operação união (union) \cup

- Notação: $r \cup s$
- Definição:
$$r \cup s = \{t \mid t \in r \vee t \in s\}$$
- Para que $r \cup s$ seja uma operação válida:
 - r e s devem ter a mesma aridade (o mesmo número de atributos)
 - O domínio dos atributos deve ser compatível (exemplo: a 2ª coluna de r tem o mesmo tipo de valores que a 2ª coluna de s)

Exemplo: para encontrar todos os clientes que têm uma conta ou um empréstimo

$$\pi_{customer_name}(depositor) \cup \pi_{customer_name}(borrower)$$

Operação diferença de conjuntos —
Exemplo:

$r =$	A	B	$s =$	A	B
	α	1		α	2
	α	2		β	3
	β	1			

$r - s =$	A	C
	α	1
	β	1

Operação diferença de conjuntos —

- Notação: $r - s$
- Definição:
$$r - s = \{t \mid t \in r \wedge t \notin s\}$$
- Para que $r - s$ seja uma operação válida:
 - r e s devem ter a mesma aridade (o mesmo número de atributos)
 - O domínio dos atributos deve ser compatível (exemplo: a 2ª coluna de r tem o mesmo tipo de valores que a 2ª coluna de s)

Álgebra Relacional - Operação produto cartesiano \times

Exemplo da Operação produto cartesiano \times

$r =$	A	B	$s =$	C	D	E
	α	1		α	10	a
	β	2		β	10	a
				β	20	b
				γ	20	b

$r \times s =$	A	B	C	D	E
	α	1	α	10	a
	α	1	β	10	a
	α	1	β	20	b
	α	1	γ	20	b
	β	2	α	10	a
	β	2	β	10	a
	β	2	β	20	b
	β	2	γ	20	b

Operação produto cartesiano \times

- Notação: $r \times s$

- Definição:

$$r \times s = \{tq \mid t \in r \wedge q \in s\}$$

- Para que $r \times s$ seja uma operação válida:
 - Os atributos de $r(R)$ e $s(S)$ são disjuntos. (Isto é, $R \cap S = \emptyset$).
Se os atributos de $r(R)$ e $s(S)$ não são disjuntos é necessário usar a operação renomear.

Álgebra Relacional - Composição de operações

Podem-se construir expressões usando vários operadores

Exemplo: $\sigma_{A=C}(r \times s)$

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

$r \times s =$	A	B	C	D	E	$\sigma_{A=C}(r \times s) =$	A	B	C	D	E
	α	1	α	10	a		α	1	α	10	a
	α	1	β	10	a		β	2	β	10	a
	α	1	β	20	b		β	2	β	20	b
	α	1	γ	20	b						
	β	2	α	10	a						
	β	2	β	10	a						
	β	2	β	20	b						
	β	2	γ	20	b						

Álgebra Relacional - Operação renomear ρ

- Permite dar um nome aos resultados das expressões da álgebra relacional.
- Permite a referência a uma relação com mais do que um nome
- Exemplo: $\rho_X(E) \leftarrow$ retorna E com o nome X
- Se a expressão da álgebra relacional, E , tem aridade n , então $\rho_{X(A_1, A_2, \dots, A_n)}(E) \leftarrow$ retorna o resultado da expressão E com o nome X , e com o nome dos atributos renomeados A_1, A_2, \dots, A_n .

Tabelas

- branch (branch_name, branch_city, assets)
agencia(agencia_nome, agencia_cidade, agencia_negocios)
- customer (customer_name, customer_street, customer_city)
cliente(cliente_nome, cliente_rua, cliente_cidade)
- account (account_number, branch_name, balance)
conta(conta_numero, agencia_nome, saldo)
- loan (loan_number, branch_name, amount)
emprestimo(emprestimo_numero, agencia_nome, valor)
- depositor (customer_name, account_number)
depositante(cliente_nome, conta_numero)
- borrower (customer_name, loan_number)
credito(cliente_nome, emprestimo_numero)

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar todos os empréstimos maiores de \$1200

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar todos os empréstimos maiores de \$1200
- loan (loan_number, branch_name, amount) ou
emprestimo(emprestimo_numero, agencia_nome, valor)

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar todos os empréstimos maiores de \$1200
- `loan (loan_number, branch_name, amount)` ou `emprestimo(emprestimo_numero, agencia_nome, valor)`
- $\sigma_{amount > 1200}(loan)$ ou $\sigma_{valor > 1200}(emprestimo)$

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar todos os empréstimos maiores de \$1200
- $\text{loan}(\text{loan_number}, \text{branch_name}, \text{amount})$ ou $\text{emprestimo}(\text{emprestimo_numero}, \text{agencia_nome}, \text{valor})$
- $\sigma_{\text{amount} > 1200}(\text{loan})$ ou $\sigma_{\text{valor} > 1200}(\text{emprestimo})$
- $\sigma_{\text{amount} > 1200}(\text{loan}) =$

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os números dos empréstimos maiores de \$1200

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os números dos empréstimos maiores de \$1200
- loan (loan_number, branch_name, amount) ou
emprestimo(emprestimo_numero, agencia_nome, valor)

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os números dos empréstimos maiores de \$1200
- `loan (loan_number, branch_name, amount)` ou
`emprestimo(emprestimo_numero, agencia_nome, valor)`
- $\pi_{loan_number}(\sigma_{amount > 1200}(loan))$

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os números dos empréstimos maiores de \$1200
- `loan (loan_number, branch_name, amount)` ou `emprestimo(emprestimo_numero, agencia_nome, valor)`
- $\pi_{loan_number}(\sigma_{amount > 1200}(loan))$
- $\pi_{loan_number}(\sigma_{amount > 1200}(loan)) =$

<i>loan_number</i>
L-11
L-14
L-15
L-16
L-17
L-23
L-93

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes dos cliente que têm um empréstimo uma conta ou ambos.

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes dos cliente que têm um empréstimo uma conta ou ambos.
- depositor (customer_name, account_number)
ou depositante(cliente_nome, conta_numero)
e borrower (customer_name, loan_number)
ou credito(cliente_nome, emprestimo_numero)

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes dos cliente que têm um empréstimo uma conta ou ambos.
- depositor (customer_name, account_number)
ou depositante(cliente_nome, conta_numero)
e borrower (customer_name, loan_number)
ou credito(cliente_nome, emprestimo_numero)
- $\pi_{customer_name}(depositor) \cup \pi_{customer_name}(borrower)$

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes dos cliente que têm um empréstimo uma conta ou ambos.
- depositor (customer_name, account_number)
ou depositante(cliente_nome, conta_numero)
e borrower (customer_name, loan_number)
ou credito(cliente_nome, emprestimo_numero)
- $\pi_{customer_name}(depositor) \cup \pi_{customer_name}(borrower)$
- depositor =

<i>customer_name</i>	<i>account_number</i>
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

Álgebra Relacional - Exemplos

borrower=

<i>customer_name</i>	<i>loan_number</i>
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Álgebra Relacional - Exemplos

$$\pi_{customer_name}(depositor) \cup \pi_{customer_name}(borrower) =$$

<i>customer_name</i>
Adams
Curry
Hayes
Jackson
Jones
Smith
Williams
Lindsay
Johnson
Turner

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge.

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge.
- loan (loan_number, branch_name, amount)
ou emprestimo(emprestimo_numero, agencia_nome, valor)
e borrower (customer_name, loan_number)
ou credito(cliente_nome, emprestimo_numero)

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge.
- $\text{loan}(\text{loan_number}, \text{branch_name}, \text{amount})$
ou $\text{emprestimo}(\text{emprestimo_numero}, \text{agencia_nome}, \text{valor})$
e $\text{borrower}(\text{customer_name}, \text{loan_number})$
ou $\text{credito}(\text{cliente_nome}, \text{emprestimo_numero})$
-

$$\pi_{\text{customer_name}}(\sigma_{\text{loan.number} = \text{borrower.number} \wedge \left(\begin{smallmatrix} \text{loan} \times \\ \text{borrower} \end{smallmatrix} \right) \text{branch_name} = \text{Perryridge}})$$

Álgebra Relacional - Exemplos

$loan \times borrower =$

<i>customer_name</i>	<i>borrower_name</i>	<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>
Adams	L-16	L-11	Round Hill	900
Adams	L-16	L-14	Downtown	1500
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Adams	L-16	L-17	Downtown	1000
Adams	L-16	L-23	Redwood	2000
Adams	L-16	L-93	Mianus	500
Curry	L-93	L-11	Round Hill	900
Curry	L-93	L-14	Downtown	1500
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Curry	L-93	L-17	Downtown	1000
Curry	L-93	L-23	Redwood	2000
Curry	L-93	L-93	Mianus	500
Hayes	L-15	L-11		900
Hayes	L-15	L-14		1500
Hayes	L-15	L-15		1500
Hayes	L-15	L-16		1300
Hayes	L-15	L-17		1000
Hayes	L-15	L-23		2000
Hayes	L-15	L-93		500
...
...
...
Smith	L-23	L-11	Round Hill	900
Smith	L-23	L-14	Downtown	1500
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Smith	L-23	L-17	Downtown	1000
Smith	L-23	L-23	Redwood	2000
Smith	L-23	L-93	Mianus	500
Williams	L-17	L-11	Round Hill	900
Williams	L-17	L-14	Downtown	1500
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300
Williams	L-17	L-17	Downtown	1000
Williams	L-17	L-23	Redwood	2000
Williams	L-17	L-93	Mianus	500

Álgebra Relacional - Exemplos

$$\sigma_{loan.number=borrower.number}(loan \times borrower) =$$

customer_name	borrower. loan_number	loan. loan_number	branch_name	amount
Adams	L-16	L-11	Round Hill	900
Adams	L-16	L-14	Downtown	1500
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Adams	L-16	L-17	Downtown	1000
Adams	L-16	L-23	Redwood	2000
Adams	L-16	L-93	Mianus	500
Curry	L-93	L-11	Round Hill	900
Curry	L-93	L-14	Downtown	1500
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Curry	L-93	L-17	Downtown	1000
Curry	L-93	L-23	Redwood	2000
Curry	L-93	L-93	Mianus	500
Hayes	L-15	L-11		900
Hayes	L-15	L-14		1500
Hayes	L-15	L-15	Perryridge	1500
Hayes	L-15	L-16		1300
Hayes	L-15	L-17		1000
Hayes	L-15	L-23		2000
Hayes	L-15	L-93		500
...
...
...
Smith	L-23	L-11	Round Hill	900
Smith	L-23	L-14	Downtown	1500
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Smith	L-23	L-17	Downtown	1000
Smith	L-23	L-23	Redwood	2000
Smith	L-23	L-93	Mianus	500
Williams	L-17	L-11	Round Hill	900
Williams	L-17	L-14	Downtown	1500
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300
Williams	L-17	L-17	Downtown	1000
Williams	L-17	L-23	Redwood	2000
Williams	L-17	L-93	Mianus	500

Álgebra Relacional - Exemplos

$$\sigma_{\substack{\text{loan.number} = \text{borrower.number} \wedge \\ \text{branch_name} = \text{Perryridge}}} (\text{loan} \times \text{borrower})$$

customer_name	borrower. loan_number	loan. loan_number	branch_name	amount
Adams	L-16	L-11	Round Hill	900
Adams	L-16	L-14	Downtown	1500
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Adams	L-16	L-17	Downtown	1000
Adams	L-16	L-23	Redwood	2000
Adams	L-16	L-93	Mianus	500
Curry	L-93	L-11	Round Hill	900
Curry	L-93	L-14	Downtown	1500
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Curry	L-93	L-17	Downtown	1000
Curry	L-93	L-23	Redwood	2000
Curry	L-93	L-93	Mianus	500
Hayes	L-15	L-11		900
Hayes	L-15	L-14		1500
Hayes	L-15	L-15	Perryridge	1500
Hayes	L-15	L-17		1000
Hayes	L-15	L-23		2000
Hayes	L-15	L-93		500
...
...
...
Smith	L-23	L-11	Round Hill	900
Smith	L-23	L-14	Downtown	1500
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Smith	L-23	L-17	Downtown	1000
Smith	L-23	L-23	Redwood	2000
Smith	L-23	L-93	Mianus	500
Williams	L-17	L-11	Round Hill	900
Williams	L-17	L-14	Downtown	1500
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300
Williams	L-17	L-17	Downtown	1000
Williams	L-17	L-23	Redwood	2000
Williams	L-17	L-93	Mianus	500

Álgebra Relacional - Exemplos

$$\pi_{customer_name}(\sigma_{loan.number = borrower.number \wedge \begin{pmatrix} loan \times \\ borrower \end{pmatrix} \\ branch_name = Perryridge})$$

<i>customer_name</i>
Adams
Hayes

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge mas não têm nenhuma conta em nenhuma agência do banco.

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge mas não têm nenhuma conta em nenhuma agência do banco.
- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge mas não têm nenhuma conta em nenhuma agência do banco.
- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge
-

$$\pi_{customer_name}(\sigma_{loan.number = borrower.number \wedge \left(\begin{matrix} loan \times \\ borrower \end{matrix} \right) \\ branch_name = Perryridge})$$

Álgebra Relacional - Exemplos

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge mas não têm nenhuma conta em nenhuma agência do banco.
- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge
-

$$\pi_{customer_name}(\sigma_{loan.number = borrower.number \wedge \left(\begin{smallmatrix} loan \times \\ borrower \end{smallmatrix} \right) \\ branch_name = Perryridge})$$

- Encontrar os nomes de todos os clientes que têm uma conta numa agência do banco.

Exemplos de perguntas:

- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge mas não têm nenhuma conta em nenhuma agência do banco.
- Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge
-

$$\pi_{customer_name}(\sigma_{loan.number = borrower.number \wedge \begin{pmatrix} loan \times \\ borrower \end{pmatrix} \\ branch_name = Perryridge})$$

- Encontrar os nomes de todos os clientes que têm uma conta numa agência do banco.
- $\pi_{customer_name}(depositor)$

Encontrar os nomes de todos os clientes que têm um empréstimo na agência Perryridge mas não têm nenhuma conta em nenhuma agência do banco.

$$\pi_{cust_name}(\sigma_{loan.num = borrower.num \wedge \begin{pmatrix} loan \times \\ borrower \end{pmatrix} - \pi_{cust_name}(depositor)} \wedge branch_name = Perryridge)$$

- **Encontrar o maior saldo das contas do banco**

Álgebra Relacional - Exemplo de maior

- **Encontrar o maior saldo das contas do banco**
- **Estratégia:**

Álgebra Relacional - Exemplo de maior

- Encontrar o maior saldo das contas do banco
- Estratégia:
- Encontrar os saldos que não são o maior

- Encontrar o maior saldo das contas do banco
- Estratégia:
- Encontrar os saldos que não são o maior
- Renomear a relação account por d para que se possa compara cada conta com as outras

- Encontrar o maior saldo das contas do banco
- Estratégia:
- Encontrar os saldos que não são o maior
- Renomear a relação account por d para que se possa compara cada conta com as outras
- Usar a operação diferença de conjuntos para encontrar as contas que não estão no passo anterior.

- Encontrar o maior saldo das contas do banco
- Estratégia:
- Encontrar os saldos que não são o maior
- Renomear a relação *account* por *d* para que se possa compara cada conta com as outras
- Usar a operação diferença de conjuntos para encontrar as contas que não estão no passo anterior.
- $\pi_{balance}(account) -$
 $\pi_{account.balance}(\sigma_{account.balance < b.balance}(account \times \rho_d(account)))$

Álgebra Relacional - Exemplo de maior

$$r = \begin{array}{|c|c|} \hline A & B \\ \hline a & 1 \\ b & 2 \\ c & 3 \\ \hline \end{array} \quad r \times \rho_d(r) =$$

A	B	d.A	d.B
a	1	a	1
a	1	b	2
a	1	c	3
b	2	a	1
b	2	b	2
b	2	c	3
c	3	a	1
c	3	b	2
c	3	c	3

Álgebra Relacional - Exemplo de maior

$$r \times \rho_d(r) =$$

A	B	d.A	d.B
a	1	a	1
a	1	b	2
a	1	c	3
b	2	a	1
b	2	b	2
b	2	c	3
c	3	a	1
c	3	b	2
c	3	c	3

$$\sigma_{B < d.B}(r \times \rho_d(r)) =$$

A	B	d.A	d.B
a	1	b	2
a	1	c	3
b	2	c	3

Álgebra Relacional - Exemplo de maior

$$\sigma_{B < d.B}(r \times \rho_d(r)) =$$

A	B	d.A	d.B
a	1	b	2
a	1	c	3
b	2	c	3

$$\pi_{\sigma_{B < d.B}(r \times \rho_d(r))} =$$

B
1
2

$$\pi_B(r) =$$

B
1
2
3

$$\pi_B(r) - \pi_{\sigma_{B < d.B}(r \times \rho_d(r))} =$$

B
3

Álgebra Relacional - Definição formal

- Uma expressão básica da álgebra relacional é:
 - Uma relação da base de dados
 - Ou uma relação constante
- Se e_1 e e_2 são expressões da álgebra relacional; as expressões seguintes também são:
 - $e_1 \cup e_2$
 - $e_1 - e_2$
 - $e_1 \times e_2$
 - $\sigma_P(e_1)$, P é um predicado sobre atributos de e_1
 - $\pi_{S_1, S_2, \dots, S_n}(e_1)$, S_1, S_2, \dots, S_n , são atributos de e_1
 - $\rho_X(e_1)$, X é um nome para o resultado de e_1

Álgebra Relacional - Operações adicionais

- Intersecção de conjuntos (Set intersection) \cap
- Junção natural (Natural join) \bowtie
- Divisão (Division) \div
- Afectação (Assignment) \leftarrow

Álgebra Relacional - Intersecção de conjuntos (Set intersection) \cap

- Notação: $r \cap s$
- Definição:
$$r \cap s = \{t \mid t \in r \wedge t \in s\}$$
- Para que a operação possa ser definida é necessário que:
 - r, s tenham a mesma aridade
 - Os atributos de r e s sejam compatíveis
- Nota: $r \cap s = r - (r - s)$

Álgebra Relacional - Intersecção de conjuntos (Set intersection) \cap

$r =$	A	B	$s =$	A	B
	α	1		α	2
	α	2		β	3
	β	1			

$r \cap s =$	<table><tr><th>A</th><th>B</th></tr><tr><td>α</td><td>2</td></tr></table>	A	B	α	2
A	B				
α	2				

Álgebra Relacional - Operação Junção natural (join) \bowtie

- Notação: $r \bowtie s$
- Sejam r e s relações nos esquemas R e S respectivamente. $r \bowtie s$ é uma relação no esquema $R \cup S$ obtida da seguinte forma:
 - Considere os pares de tuplos t_r da relação r e t_s da relação s .
 - Se t_r e t_s tem o mesmo valor em cada um dos atributos de $R \cup S$, adiciona-se o tuplo t ao resultado, com
- Exemplo:
 $R = (A, B, C, D)$
 $S = (E, B, D)$
 - Esquema resultado = (A, B, C, D, E)
 - $r \bowtie s$ é definido como:
$$\pi_{(r.A, r.B, r.C, r.D, s.E)}(\sigma_{r.B=s.B}(\sigma_{r.D=s.D}(r \times s)))$$

Álgebra Relacional - Operação Junção natural (join) \bowtie

• $r =$

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

$s =$

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

• $r \bowtie s =$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

- Notação: $r \div s$
- Usa-se para perguntas com a expressão “para todos”
- Sejam r e s relações nos esquemas R e S respectivamente.
 - $R = (A_1, \dots, A_m, B_1, \dots, B_n)$
 - $S = (B_1, \dots, B_n)$
- O resultado de $r \div s$ é uma relação no esquema $R - S = (B_1, \dots, B_n)$
- $r \div s = \{t \mid t \in \pi_{R-S}(r) \wedge \forall u \in s, tu \in r\}$
 tu é a concatenação dos tuplos t e u .

Álgebra Relacional - Operação Divisão \div

Exemplo

• $r =$

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ϵ	6
ϵ	1
β	2

$s =$

B
1
2

• $r \div s =$

A
α
β

Álgebra Relacional - Operação Divisão \div

• $r =$

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

$s =$

D	E
a	1
b	1

$r \div s =$

A	B	C
α	a	γ
γ	a	γ

- Propriedade

- Seja $q = r \div s$
- Então q é a maior relação que satisfaz $q \times s \subseteq r$

- Definição usando os operadores básicos:

Sejam $r(R)$ e $s(S)$ relações, e seja $S \subseteq R$

$$r \div s = \pi_{R-S}(r) - \pi_{R-S}((\pi_{R-S}(r) \times s) - \pi_{R-S,S}(r))$$

- $\pi_{R-S,S}(r)$ reordena os atributos de r
- $\pi_{R-S}(\pi_{R-S}(r) \times s) - \pi_{R-S,S}(r)$ tem os tuplos t pertencentes a $\pi_{R-S}(r)$ tal que para algum tuplo $u \in s$, $tu \notin r$.

Álgebra Relacional - Operação Afecção \leftarrow

A operação afecção ($\text{temp} \leftarrow e_1$) permite representar as perguntas mais complexas de forma conveniente

- Escrever uma pergunta com uma série de afecções
- Seguida de uma expressão cujo valor é apresentado como resultado da pergunta.
- A Afecção é sempre feita para uma variável de relação temporária.
- O resultado à direita de \leftarrow é atribuído à variável de relação do lado esquerdo de \leftarrow .
- Pode-se usar a variável em expressões subsequentes.

Exemplo:

- Escrever $r \div s$ como
- $\text{temp}_1 \leftarrow \pi_{R-S}(r)$
- $\text{temp}_2 \leftarrow \pi_{R-S}(\text{temp}_1 \times s) - \pi_{R-S,S}(r)$
- $\text{resultado} = \text{temp}_1 - \text{temp}_2$

- Encontrar os nomes de todos os cliente que têm um empréstimo e uma conta no banco

- Encontrar os nomes de todos os cliente que têm um empréstimo e uma conta no banco
- depositor (customer_name, account_number)
ou depositante(cliente_nome, conta_numero)
e borrower (customer_name, loan_number)
ou credito(cliente_nome, emprestimo_numero)

- Encontrar os nomes de todos os cliente que têm um empréstimo e uma conta no banco
- depositor (customer_name, account_number)
ou depositante(cliente_nome, conta_numero)
e borrower (customer_name, loan_number)
ou credito(cliente_nome, emprestimo_numero)
- $\pi_{customer_name}(borrower) \cap \pi_{customer_name}(depositor)$

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar o nome de todos os clientes que têm um empréstimo e o valor do empréstimo

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar o nome de todos os clientes que têm um empréstimo e o valor do empréstimo
- loan (loan_number, branch_name, amount)
ou emprestimo(emprestimo_numero, agencia_nome, valor)
e borrower (customer_name, loan_number)
ou credito(cliente_nome, emprestimo_numero)

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar o nome de todos os clientes que têm um empréstimo e o valor do empréstimo
- `loan` (`loan_number`, `branch_name`, `amount`)
ou `emprestimo`(`emprestimo_numero`, `agencia_nome`,`valor`)
e `borrower` (`customer_name`, `loan_number`)
ou `credito`(`cliente_nome`, `emprestimo_numero`)
- $\pi_{customer_name, amount}(borrower \bowtie loan)$

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar o nome de todos os clientes que têm um empréstimo e o valor do empréstimo
- $\text{loan}(\text{loan_number}, \text{branch_name}, \text{amount})$
ou $\text{emprestimo}(\text{emprestimo_numero}, \text{agencia_nome}, \text{valor})$
e $\text{borrower}(\text{customer_name}, \text{loan_number})$
ou $\text{credito}(\text{cliente_nome}, \text{emprestimo_numero})$
- $\pi_{\text{customer_name}, \text{amount}}(\text{borrower} \bowtie \text{loan})$
- $\pi_{\text{customer_name}, \text{amount}}(\sigma_{\text{borrower.number}=\text{loan.number}}(\text{borrower} \times \text{loan}))$

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar todos os clientes que têm pelo menos uma conta nas agências de Downtown e Uptwon

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar todos os clientes que têm pelo menos uma conta nas agências de Downtown e Uptwon
- account (account_number, branch_name, balance)
conta(conta_numero, agencia_nome, saldo)
e depositor (customer_name, account_number)
ou depositante(cliente_nome, conta_numero)

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar todos os clientes que têm pelo menos uma conta nas agências de Downtown e Uptown
- $\text{account}(\text{account_number}, \text{branch_name}, \text{balance})$
 $\text{conta}(\text{conta_numero}, \text{agencia_nome}, \text{saldo})$
e $\text{depositor}(\text{customer_name}, \text{account_number})$
ou $\text{depositante}(\text{cliente_nome}, \text{conta_numero})$
- $\pi_{\text{customer_name}}(\sigma_{\text{branch_name}=\text{Downtown}}(\text{account} \bowtie \text{depositor})) \cap \pi_{\text{customer_name}}(\sigma_{\text{branch_name}=\text{Uptown}}(\text{account} \bowtie \text{depositor}))$

Álgebra Relacional - - Exemplos de Perguntas sobre o Banco

- Encontrar todos os clientes que têm pelo menos uma conta nas agências de Downtown e Uptown
- $\text{account}(\text{account_number}, \text{branch_name}, \text{balance})$
 $\text{conta}(\text{conta_numero}, \text{agencia_nome}, \text{saldo})$
e $\text{depositor}(\text{customer_name}, \text{account_number})$
ou $\text{depositante}(\text{cliente_nome}, \text{conta_numero})$
- $\pi_{\text{customer_name}}(\sigma_{\text{branch_name}=\text{Downtown}}(\text{account} \bowtie \text{depositor})) \cap \pi_{\text{customer_name}}(\sigma_{\text{branch_name}=\text{Uptown}}(\text{account} \bowtie \text{depositor}))$
- $\pi_{\text{customer_name}, \text{branch_name}}(\text{account} \bowtie \text{depositor}) \div \rho_t(\text{branch_name}(\{("Downtown"), ("Uptown")\}))$
Nota: $\{("Downtown"), ("Uptown")\}$ é uma relação constante

- Encontrar todos os clientes que têm uma conta em todas as agências na cidade de Brooklyn.

- Encontrar todos os clientes que têm uma conta em todas as agências na cidade de Brooklyn.
- account (account_number, branch_name, balance)
depositor (customer_name, account_number)
branch (branch_name, branch_city, assets)

- Encontrar todos os clientes que têm uma conta em todas as agências na cidade de Brooklyn.
- $\text{account}(\text{account_number}, \text{branch_name}, \text{balance})$
 $\text{depositor}(\text{customer_name}, \text{account_number})$
 $\text{branch}(\text{branch_name}, \text{branch_city}, \text{assets})$
- $\pi_{\text{customer_name}, \text{branch_name}}(\text{account} \bowtie \text{depositor}) \div \pi_{\text{branch_name}}(\sigma_{\text{branch_city}=\text{Brooklyn}}(\text{branch}))$

Álgebra Relacional - - Operações da Algebra Relacional estendidas

- Operação Projectção Generalizada (Generalized Projection) Π
- Funções de Agregação (Aggregate Functions) \mathcal{F}
- Operação Agregação \mathcal{G}
- Operações Junção exterior (Outer Join) (\bowtie , \ltimes , $\ltimes\ltimes$)

Álgebra Relacional - Projectão generalizada Π

- Estende a operação projecção permitindo o uso de funções aritméticas na lista de atributos da projecção.
 - $\Pi_{F_1, F_2, \dots, F_n}(e)$
 - e é uma expressão da Álgebra relacional
 - F_1, F_2, \dots, F_n são expressões aritméticas com constantes e atributos do esquema de e , E
- Exemplo:
 - Considere a relação:
`credit_info(customer_name, limit, credit_balance)` ou
`info_credito(nome_cliente, limite, balanço_credito)`
 - encontrar o valor que cada pessoa pode gastar:
 $\Pi_{customer_name, limite - credit_balance}(credit_info)$

- Funções de agregação recebem uma colecção de valores e retornam um único valor
 - avg: valor médio
 $avg(\{1, 2, 3, 4, 5\}) = 3$
 - min: valor mínimo
 $min(\{1, 2, 3, 4, 5\}) = 1$
 - max: valor máximo
 $max(\{1, 2, 3, 4, 5\}) = 5$
 - sum: soma dos valores
 $sum(\{1, 2, 3, 4, 5\}) = 15$
 - count: numero de valores
 $count(\{1, 2, 3, 4, 5\}) = 5$

- operação agregação \mathcal{G}

$$g_1, g_2, \dots, g_n \mathcal{G}_{F_1(A_1), F_2(A_2), \dots, F_m(A_m)}(e)$$

- e é uma expressão da álgebra relacional
- g_1, g_2, \dots, g_n é a lista de atributos onde se pretende agrupar (pode ser vazia)
- F_i é uma função de agregação
- A_i é o nome de um atributo do esquema E

Álgebra Relacional - Operação Agregação Exemplo

• $r =$

A	B	C
α	α	7
α	β	7
β	β	3
β	β	10

• $\mathcal{G}_{sum(C)}(r) =$

$sum(C)$
27

Álgebra Relacional - Operação Agregação Exemplo

- Para cada agência indique valor total dos depósitos

- $account =$

Account_number	Branch_name	Balance
Perryridge	A – 102	400
Perryridge	A – 201	900
Brighton	A – 217	750
Brighton	A – 215	750
Redwood	A – 222	700

- $Branch_name \mathcal{G}_{sum(Balance)}(account) =$

Branch_name	$sum(Balance)$
Perryridge	1300
Brighton	1500
Redwood	700

Álgebra Relacional - Operação Agregação Exemplo

- O resultado da agregação não tem nome
 - Pode-se dar um nome ao resultado
 - i.e Pode-se renomear o resultado na operação agregação

$Branch_name \mathcal{G}_{sum(Balance)} as Sum_balance(account) =$

Branch_name	Sum_balance
Perryridge	1300
Brighton	1500
Redwood	700

Álgebra Relacional - Junção exterior (Outer Join)

- A Junção exterior é uma extensão da operação junção para evitar perder informação.
 - Calcula a junção e adiciona tuplos de uma relação que não encaixam na outra relação
 - Usa valores nulos (null values):
Nulo (null) significa que o valor é desconhecido ou não existe
 - Todas as comparações que envolvem valores nulos retornam *Falso* por definição

Álgebra Relacional - Junção exterior: Exemplo

• $loan =$

Loan_number	Branch_name	Amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

• $borrower =$

Customer_name	Loan_number
Jones	L-170
Smith	L-230
Hayes	L-155

• $loan \bowtie borrower =$

Loan_number	Branch_name	Amount	Customer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

Álgebra Relacional - Junção exterior: Exemplo

• *loan* =

Loan_number	Branch_name	Amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

• *borrower* =

Customer_name	Loan_number
Jones	L-170
Smith	L-230
Hayes	L-155

• Left outer join *loan* \bowtie *borrower* =

Loan_number	Branch_name	Amount	Customer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	<i>null</i>

Álgebra Relacional - Junção exterior: Exemplo

• $loan =$

Loan_number	Branch_name	Amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

• $borrower =$

Customer_name	Loan_number
Jones	L-170
Smith	L-230
Hayes	L-155

• Right outer join $loan \bowtie_r borrower =$

Loan_number	Branch_name	Amount	Customer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	<i>null</i>	<i>null</i>	Hayes

Álgebra Relacional - Junção exterior: Exemplo

• $loan =$

Loan_number	Branch_name	Amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

• $borrower =$

Customer_name	Loan_number
Jones	L-170
Smith	L-230
Hayes	L-155

• Full outer join $loan \bowtie_{full} borrower =$

Loan_number	Branch_name	Amount	Customer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	<i>null</i>
L-155	<i>null</i>	<i>null</i>	Hayes

Valores Nulos (Null Values)

- É possível os tuplos terem valores nulos (null) em alguns dos seus atributos
null significa que o valor é desconhecido ou que não existe
- O resultado de qualquer operação aritmética que envolva valores *null* é *null*.
- As funções de agregação ignoram os valores *null* (no SQL também)
- Para eliminar duplicados e agrupar o *null* é tratado como outro valor qualquer (no SQL também)

Álgebra Relacional -Valores Nulos (Null Values)

- As comparações com valores nulos retornam o valor: *unknown*
Se *false* fosse usado em vez de *unknown*, então $(A < 5)$ não seria equivalente a $A \geq 5$
- Lógica a três valores o valor *unknown*:
 - $(\text{unknown or true}) = \text{true},$
 - OR: $(\text{unknown or false}) = \text{unknown}$
 $(\text{unknown or unknown}) = \text{unknown}$
 - $(\text{true and unknown}) = \text{unknown},$
 - AND: $(\text{false and unknown}) = \text{false},$
 $(\text{unknown and unknown}) = \text{unknown}$
 - NOT: $(\text{not unknown}) = \text{unknown}$
- Em SQL “P is unknown” é verdade (true) se o predicado P é *unknown*
- Na selecção (σ) quando o resultado do predicado é *unknown* o efeito é igual ao caso em que é *false*.

- O conteúdo da base de dados pode ser alterado usando as seguintes operações:
 - Deletion (remover ou apagar)
 - Insertion (inserir)
 - Updating (actualizar)
- Estas operações são representadas usando o operador afectação \leftarrow .

Álgebra Relacional - Remover (Deletion)

- Um pedido de remoção representa-se como uma pergunta, mas o seu resultado é a remoção de um conjunto de tuplos de uma tabela da base de dados.
- Só se podem remover tuplos inteiros, não se podem remover valores de alguns atributos
- A remoção representa-se com a seguinte expressão da álgebra relacional:
 - $r \leftarrow r - e$
 - onde r é uma relação e e é uma expressão da álgebra relacional.

Álgebra Relacional - Remover: Exemplos

- Remover todas as contas da agência Perryridge.
 - $account \leftarrow account - \sigma_{Branch_name=Perryridge}(account)$
- Remover todos empréstimos com o valor entre 0 e 50
 - $loan \leftarrow loan - \sigma_{Amount \geq 0 \wedge \leq 50}(loan)$
- Remover todas as contas da agência de Needham.
 - $r_1 \leftarrow \sigma_{Branch_city=Needham}(account \bowtie branch)$
 - $r_2 \leftarrow \pi_{Account_number, Branch_name, Balance}(r_1)$
 - $r_3 \leftarrow \pi_{Customer_name, Account_number}(r_2 \bowtie depositor)$
 - $account \leftarrow account - r_2$
 - $depositor \leftarrow depositor - r_3$

Álgebra Relacional - Inserção (Insert)

- Para inserir dados numa relação podemos:
 - Especificar o tuplo
 - Ou escrever uma pergunta cujo resultado é o conjunto de tuplos a ser inserido
- Na álgebra relacional, uma inserção exprime-se com a expressão:
 - $r \leftarrow r \cup e$
 - onde r é uma relação e e é uma expressão da álgebra relacional.
- A inserção de um único tuplo obtém-se quando e é uma relação constante com um só tuplo.

Álgebra Relacional - Inserção: Exemplos

- Inserir na base de dados a informação de que o Sr Smith tem 1200 dólares na conta A-973 na agência Perryridge .
 - $account \leftarrow account \cup \{("A - 937", "Perryridge", 1200)\}$
 - $depositor \leftarrow depositor \cup \{("Smith", "A - 937")\}$
- Ofereça a todos os clientes com empréstimos na agência Perryridge, 200 dólares numa nova conta poupança. Considere que o número do empréstimo será o número da nova conta.
 - $r_1 \leftarrow (\sigma_{Branch_name=Perryridge}(borrower \bowtie loan))$
 - $account \leftarrow account \cup \Pi_{Loan_number, Branch_name, 200}(r_1)$
 - $depositor \leftarrow depositor \cup \Pi_{Customer_name, Loan_number}(r_1)$

- A actualização é um mecanismo para alterar um valor num tuplo sem alterar todos os valores do tuplo.
- Usa-se o operador projecção generalizada para a actualização:
 - $r \leftarrow \Pi_{F_1, F_2, \dots, F_m}(r)$
 - Cada F_i é:
 - o i -ésimo atributo de r se o i -ésimo atributo não é actualizado
 - ou é uma expressão com constantes e atributos de r se o i -ésimo atributo é actualizado

Álgebra Relacional - Atualização: Exemplos

- Faça o pagamento dos juros aumentando 5% em todos os saldos de todas as contas.
 - $account \leftarrow \Pi_{Account_number, Branch_name, Balance * 1.05}(r)$
- Pague 6% de juro às contas com um saldo superior a 10.000 e 5% de juro às outras contas.
 - $account \leftarrow$
 $\Pi_{Account_number, Branch_name, Balance * 1.06}(\sigma_{Balance > 10000}(r)) \cup$
 $\Pi_{Account_number, Branch_name, Balance * 1.05}(\sigma_{Balance \leq 10000}(r))$