

Álgebra Relacional

Base de Dados - 2020/21
Carlos Costa

1

Introdução

Linguagem de Consulta/Interrogação de BD

- Álgebra Relacional
 - Linguagem formal do Modelo Relacional
 - Um conjunto básico de operações
- Outras linguagem formais: *relational calculus*
- As linguagens formais oferecem uma base teórica para a linguagem de consulta utilizada na prática.
- Linguagem prática do Modelo Relacional
 - SQL

2

2

Álgebra Relacional



Questões?

- Como deve ser uma linguagem de interrogação da BD?
- Que tipo de interrogações existem?
- Como é que são os resultados?
- Expressões de álgebra relacional (linguagem).
 - Sequência de operações de álgebra relacional.
 - Permitem formular pedidos básicos de recuperação de informação sobre uma ou mais relações.
- Formulação da interrogação:
 - conjunto de operadores que operam sobre as relações
 - devolvem uma nova relação
- Vamos estudar um conjunto de operações...

3

3

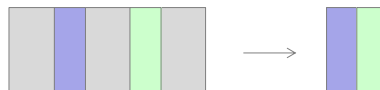
Álgebra Relacional - Operações Básicas



• Seleção



• Projeção



• União



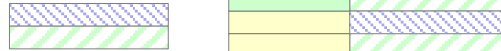
• Diferença



• Produto Cartesiano




• Renomeação



4

deti

Seleção



- Notação: $\sigma_{\langle \text{selection condition} \rangle}(R)$
 - Utilizada para selecionar um subconjunto de tuplos da relação ($t \in R$) que satisfazem os critérios de seleção.
 - “selection condition” é uma expressão booleana.

Relation2 $\leftarrow \sigma_{\langle \text{selection condition} \rangle}(\text{Relation1})$

- O resultado é uma nova relação (Relation2) que tem um esquema relacional igual à original (Relation1).

5

5

deti

Seleção - Predicado

- Operadores de Comparação
 - Permitem comparar dois atributos ou um atributo com um valor.
 - Operandos: Nomes dos atributos e constantes.
 - Operadores: =, \neq , \leq , \geq , $<$, $>$
 - Exemplos:
 - $\sigma_{\text{Dno}=4}(\text{EMPLOYEE})$
 - $\sigma_{\text{Salary}>30000}(\text{EMPLOYEE})$
- Condições Booleanas
 - Utilização de AND, OR e NOT.
 - Exemplo:
 - $\sigma_{(\text{Dno}=4 \text{ AND } \text{Salary}>25000) \text{ OR } (\text{Dno}=5 \text{ AND } \text{Salary}>30000)}(\text{EMPLOYEE})$

6

6

Seleção - Exemplo

EMPLOYEE

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

$\sigma_{(Dno=4 \text{ AND } Salary>25000) \text{ OR } (Dno=5 \text{ AND } Salary>30000)} (EMPLOYEE)$

SQL query (próxima aula...)

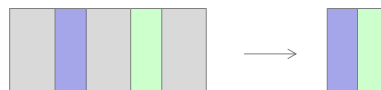
SELECT * FROM EMPLOYEE
WHERE Dno=4 AND Salary>25000
OR Dno=5 AND Salary>30000;

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |

7

7

Projeção



- Notação: $\Pi_{\langle \text{attribute list} \rangle}(R)$
 - $\langle \text{attribute list} \rangle = A_1, A_2, \dots, A_k$
 - $A_1 \dots A_k$ são nomes dos atributos da relação R
- O resultado é uma nova relação só com os k atributos selecionados.
- São removidas as linhas duplicadas do resultado.
 - Condição de conjunto (set)

8

8



Projeção - Exemplo

EMPLOYEE

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

$\Pi_{\text{Lname, Fname, Salary}}(\text{EMPLOYEE})$

SQL query:

```
SELECT DISTINCT Lname, Fname, Salary
FROM EMPLOYEE;
```

| Lname | Fname | Salary |
|---------|----------|--------|
| Smith | John | 30000 |
| Wong | Franklin | 40000 |
| Zelaya | Alicia | 25000 |
| Wallace | Jennifer | 43000 |
| Narayan | Ramesh | 38000 |
| English | Joyce | 25000 |
| Jabbar | Ahmad | 25000 |
| Borg | James | 55000 |

9

9



Encadeamento de Operações

- $\Pi_{\text{Fname, Lname, Salary}}(\sigma_{\text{Dno}=5}(\text{EMPLOYEE}))$
- Se quisermos renomear os atributos e a relação:

$\text{TEMP} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEE})$

$R(\text{First_name, Last_name, Salary}) \leftarrow \Pi_{\text{Fname, Lname, Salary}}(\text{TEMP})$

TEMP

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |

R

| First_name | Last_name | Salary |
|------------|-----------|--------|
| John | Smith | 30000 |
| Franklin | Wong | 40000 |
| Ramesh | Narayan | 38000 |
| Joyce | English | 25000 |

10

10

Renomeação

- Notação: $\rho_{R_2(B_1, B_2, \dots, B_n)}(R_1)$ ou $\rho_{R_2}(R_1)$

ou $\rho_{(B_1, B_2, \dots, B_n)}(R_1)$

- No primeiro caso o resultado é uma nova relação R2 com os atributos renomeados (B1, B2, ..., Bn).
- No segundo caso só renomeamos a relação.
- No terceiro só renomeamos os atributos.

SQL query:

```
SELECT E.Fname AS First_name, E.Lname AS Last_name, E.Salary AS Salary
FROM EMPLOYEE AS E
WHERE E.Dno=5;
```

Seleção

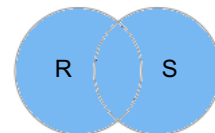
R1: EMPLOYEE
R2: E
Fname -> First_name
Lname -> Last_Name
...

11

11

União

- Notação: $R \cup S = \{t: t \in R \vee t \in S\}$
- As tabelas têm de ser compatíveis
 - Mesmo número de atributos
 - Atributos com domínios compatíveis
- O resultado é uma relação que inclui todos os tuplos de R e de S
 - Os tuplos duplicados são eliminados



| STUDENT | |
|---------|---------|
| Fname | Lname |
| Susan | Yao |
| Ramesh | Shah |
| Johnny | Kohler |
| Barbara | Jones |
| Amy | Ford |
| Jimmy | Wang |
| Ernest | Gilbert |

U

| INSTRUCTOR | |
|------------|---------|
| Fname | Lname |
| John | Smith |
| Ricardo | Browne |
| Susan | Yao |
| Francis | Johnson |
| Ramesh | Shah |



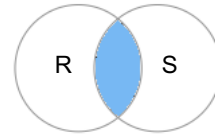
| Fname | Lname |
|---------|---------|
| Susan | Yao |
| Ramesh | Shah |
| Johnny | Kohler |
| Barbara | Jones |
| Amy | Ford |
| Jimmy | Wang |
| Ernest | Gilbert |
| John | Smith |
| Ricardo | Browne |
| Francis | Johnson |

12

12

Intersecção

- Notação: $R \cap S = \{t: t \in R \wedge t \in S\}$
- As tabelas têm de ser compatíveis
 - Mesmo número de atributos
 - Atributos com domínios compatíveis
- O resultado é uma relação que inclui os tuplos que existem simultaneamente em R e S
 - Os tuplos duplicados são eliminados



STUDENT

| Fn | Ln |
|---------|---------|
| Susan | Yao |
| Ramesh | Shah |
| Johnny | Kohler |
| Barbara | Jones |
| Amy | Ford |
| Jimmy | Wang |
| Ernest | Gilbert |



INSTRUCTOR

| Fname | Lname |
|---------|---------|
| John | Smith |
| Ricardo | Browne |
| Susan | Yao |
| Francis | Johnson |
| Ramesh | Shah |



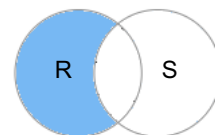
| Fn | Ln |
|--------|------|
| Susan | Yao |
| Ramesh | Shah |

13

13

Diferença

- Notação: $R - S = \{t: t \in R \wedge t \notin S\}$
- As tabelas têm de ser compatíveis
 - Mesmo número de atributos
 - Atributos com domínios compatíveis
- O resultado é uma relação que inclui os tuplos de R que não existem em S



STUDENT

| Fn | Ln |
|---------|---------|
| Susan | Yao |
| Ramesh | Shah |
| Johnny | Kohler |
| Barbara | Jones |
| Amy | Ford |
| Jimmy | Wang |
| Ernest | Gilbert |



INSTRUCTOR

| Fname | Lname |
|---------|---------|
| John | Smith |
| Ricardo | Browne |
| Susan | Yao |
| Francis | Johnson |
| Ramesh | Shah |



| Fn | Ln |
|---------|---------|
| Johnny | Kohler |
| Barbara | Jones |
| Amy | Ford |
| Jimmy | Wang |
| Ernest | Gilbert |

14

14

União, Intersecção e Diferença

- Em SQL existem os seguintes comandos
 - UNION (ALL), INTERSECT (ALL) e EXCEPT (ALL)

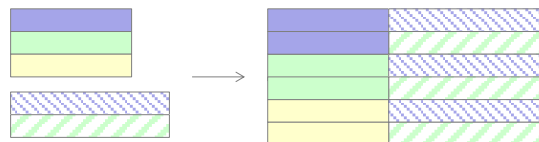
Propriedades:

- União e Intersecção são operações comutativas:
 - $R \cup S = S \cup R$ e $R \cap S = S \cap R$
- A diferença não é comutativa:
 - $R - S \neq S - R$
- União e Intersecção são operações associativas:
 - $R \cup (S \cap T) = (R \cup S) \cap T$ e $(R \cap S) \cup T = R \cap (S \cup T)$

15

15

Produto Cartesiano



- Notação: $R \times S$
- Permite-nos combinar tuplos de relações diferentes.
 - O resultado é uma nova relação (Q) que combina cada elemento (tuplo) de uma relação (R) com um elemento (tuplo) da outra relação (S):

$$Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m) = R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$$
 - O número de tuplos de Q é $n * m$.
- UK: "CROSS JOIN"

16

16

deti

Produto Cartesiano - Exemplo

| Fname | Lname | Ssn |
|----------|---------|-----------|
| Alicia | Zelaya | 999887777 |
| Jennifer | Wallace | 987654321 |
| Joyce | English | 453453453 |

| Fname | Lname | Ssn | Essn | Dependent_name | Sex | Bdate | ... |
|----------|---------|-----------|-----------|----------------|-----|------------|-----|
| Alicia | Zelaya | 999887777 | 333445555 | Alice | F | 1986-04-05 | ... |
| Alicia | Zelaya | 999887777 | 333445555 | Theodore | M | 1983-10-25 | ... |
| Alicia | Zelaya | 999887777 | 333445555 | Joy | F | 1958-05-03 | ... |
| Alicia | Zelaya | 999887777 | 987654321 | Abner | M | 1942-02-28 | ... |
| | | 999887777 | 123456789 | Michael | M | 1988-01-04 | ... |
| | | 999887777 | 123456789 | Alice | F | 1988-12-30 | ... |
| | | 999887777 | 123456789 | Elizabeth | F | 1967-05-05 | ... |
| | | 987654321 | 333445555 | Alice | F | 1986-04-05 | ... |
| | | 987654321 | 333445555 | Theodore | M | 1983-10-25 | ... |
| | | 987654321 | 333445555 | Joy | F | 1958-05-03 | ... |
| | | 987654321 | 987654321 | Abner | M | 1942-02-28 | ... |
| | | 987654321 | 123456789 | Michael | M | 1988-01-04 | ... |
| | | 987654321 | 123456789 | Alice | F | 1988-12-30 | ... |
| Jennifer | Wallace | 987654321 | 123456789 | Elizabeth | F | 1967-05-05 | ... |
| Joyce | English | 453453453 | 333445555 | Alice | F | 1986-04-05 | ... |
| Joyce | English | 453453453 | 333445555 | Theodore | M | 1983-10-25 | ... |
| Joyce | English | 453453453 | 333445555 | Joy | F | 1958-05-03 | ... |
| Joyce | English | 453453453 | 987654321 | Abner | M | 1942-02-28 | ... |
| Joyce | English | 453453453 | 123456789 | Michael | M | 1988-01-04 | ... |
| Joyce | English | 453453453 | 123456789 | Alice | F | 1988-12-30 | ... |
| Joyce | English | 453453453 | 123456789 | Elizabeth | F | 1967-05-05 | ... |

DEPENDENT

| Essn | Dependent_name | Sex | Bdate | Relationship |
|-----------|----------------|-----|------------|--------------|
| 333445555 | Alice | F | 1986-04-05 | Daughter |
| 333445555 | Theodore | M | 1983-10-25 | Son |
| 333445555 | Joy | F | 1958-05-03 | Spouse |
| 987654321 | Abner | M | 1942-02-28 | Spouse |
| 123456789 | Michael | M | 1988-01-04 | Son |
| 123456789 | Alice | F | 1988-12-30 | Daughter |
| 123456789 | Elizabeth | F | 1967-05-05 | Spouse |

EMPNames X DEPENDENT

17

17

deti

Junção θ (THETA JOIN)

- Notação: $R \bowtie_C S$
 - Pode ser visto como o resultado das seguintes operações:

$R3 \leftarrow R1 \times R2$
 $\sigma_C(R3)$

(produto cartesiano)
 (seleção com condição c)
 - C é <join condition> que pode tomar a seguinte forma:

$\langle \text{condition} \rangle \text{ AND } \langle \text{condition} \rangle \text{ AND } \dots \text{ AND } \langle \text{condition} \rangle$
 - Em cada <condition> podemos aplicar operadores de comparação:

$=, <, \leq, >, \geq, \neq$

18

18

Junção θ (THETA JOIN) - Exemplo

- Pretendemos saber os nomes dos funcionários gestores de departamentos

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 90000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 668894444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

| Dname | Dnumber | Mgr_ssn | Mgr_start_date |
|----------------|---------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

Para obter o nome dos gestores temos de combinar cada tuplo do departamento (Department) com um tuplo dos funcionários (Employee) cujo Ssn é igual ao Mgr_ssn.

$DEPT_MGR \leftarrow DEPARTMENT \bowtie_{Mgr_ssn=Ssn} EMPLOYEE$

| Dname | Dnumber | Mgr_ssn | ... | Fname | Minit | Lname | Ssn | ... |
|----------------|---------|-----------|-----|----------|-------|---------|-----------|-----|
| Research | 5 | 333445555 | ... | Franklin | T | Wong | 333445555 | ... |
| Administration | 4 | 987654321 | ... | Jennifer | S | Wallace | 987654321 | ... |
| Headquarters | 1 | 888665555 | ... | James | E | Borg | 888665555 | ... |

Depois só temos de utilizar projeção para obter os atributos desejados:

$RESULT \leftarrow \pi_{Dname, Lname, Fname}(DEPT_MGR)$

19

19

Junção - Variações da Junção θ

- Equi-Junção (EquiJoin)
 - É utilizado o operador = na condição de junção.
 - Exemplo anterior: $DEPARTMENT \bowtie_{Mgr_ssn=Ssn} EMPLOYEE$.
 - Vamos ter sempre duas colunas repetidas.
- Junção Natural (Natural Join): $R \bowtie S$
 - Condição implícita: igualdade dos atributos com o mesmo nome.
 - Os atributos repetidos são removidos.
 - Nota: Muitas vezes opta-se por renomear colunas de modo a facilitar junções naturais.

| R | | | S | | |
|---|---|--|---|---|--|
| X | Y | | Y | Z | |
| a | c | | d | g | |
| b | d | | e | h | |

 \bowtie

| X | Y | Z |
|---|---|---|
| b | d | g |

20

20

deti

Junção Natural - Exemplo

| Pname | Pnumber | Plocation | Dnum |
|-----------------|---------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

| Dname | Dnumber | Mgr_ssn | Mgr_start_date |
|----------------|---------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

$\text{PROJECT} \bowtie \rho_{(Dname, Dnum, Mgr_ssn, Mgr_start_date)}(\text{DEPARTMENT})$

| Pname | Pnumber | Plocation | Dnum | Dname | Mgr_ssn | Mgr_start_date |
|-----------------|---------|-----------|------|----------------|-----------|----------------|
| ProductX | 1 | Bellaire | 5 | Research | 333445555 | 1988-05-22 |
| ProductY | 2 | Sugarland | 5 | Research | 333445555 | 1988-05-22 |
| ProductZ | 3 | Houston | 5 | Research | 333445555 | 1988-05-22 |
| Computerization | 10 | Stafford | 4 | Administration | 987654321 | 1995-01-01 |
| Reorganization | 20 | Houston | 1 | Headquarters | 888665555 | 1981-06-19 |
| Newbenefits | 30 | Stafford | 4 | Administration | 987654321 | 1995-01-01 |

21

21

deti

Divisão

- Notação: $R \div S$
 - Dadas as relações $R(A_1, \dots, A_r, B_1, \dots, B_k)$ e $S(B_1, \dots, B_k)$
 - O resultado incluirá todos os tuplos de $R(A_1, \dots, A_r)$ que tenham correspondência com todos os tuplos de S em $R(B_1, \dots, B_k)$.
 - R_1 e R_2 são projeções de R
 - número de atributos de $R >$ número de atributos de S .
- Em SQL não existe um operador que implemente a divisão. Temos de recorrer a operadores básicos:
 - $R \div S = \pi_{R-S}(R) - \pi_{R-S}((\pi_{R-S}(R) \times S) - R)$
 - onde $\pi_{R-S} \rightarrow \pi_{(A_1, \dots, A_r)}$

22

22

deti

Divisão - Exemplos

| A | B |
|---|---|
| a | c |
| b | c |
| a | d |
| b | e |
| a | a |
| a | e |

÷

| B |
|---|
| c |
| d |

→

| A |
|---|
| a |

Department

| Dno | Name | Location |
|-----|----------------|----------|
| 1 | Research | Houston |
| 2 | Commercial | Bellaire |
| 3 | Administration | LA |
| 2 | Commercial | Houston |
| 4 | Headquarters | Bellaire |
| 2 | Commercial | LA |

Location

| Location |
|----------|
| Houston |
| Bellaire |
| LA |

→

| Dno | Name |
|-----|------------|
| 2 | Commercial |

Departamentos que existem em todas as localizações?

23

23

deti

Operações Álgebra Relacional - Resumo

| OPERATION | PURPOSE | NOTATION |
|-------------------|--|--|
| SELECT | Selects all tuples that satisfy the selection condition from a relation R . | $\sigma_{\langle \text{selection condition} \rangle}(R)$ |
| PROJECT | Produces a new relation with only some of the attributes of R , and removes duplicate tuples. | $\pi_{\langle \text{attribute list} \rangle}(R)$ |
| THETA JOIN | Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition. | $R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$ |
| EQUIJOIN | Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons. | $R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$, OR $R_1 \bowtie_{\langle \text{join attributes } 1 \rangle, \langle \text{join attributes } 2 \rangle} R_2$ |
| NATURAL JOIN | Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all. | $R_1 \star_{\langle \text{join condition} \rangle} R_2$, OR $R_1 \star_{\langle \text{join attributes } 1 \rangle, \langle \text{join attributes } 2 \rangle} R_2$ OR $R_1 \star R_2$ |
| UNION | Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2 ; R_1 and R_2 must be union compatible. | $R_1 \cup R_2$ |
| INTERSECTION | Produces a relation that includes all the tuples in both R_1 and R_2 ; R_1 and R_2 must be union compatible. | $R_1 \cap R_2$ |
| DIFFERENCE | Produces a relation that includes all the tuples in R_1 that are not in R_2 ; R_1 and R_2 must be union compatible. | $R_1 - R_2$ |
| CARTESIAN PRODUCT | Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2 . | $R_1 \times R_2$ |
| DIVISION | Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$. | $R_1(Z) \div R_2(Y)$ |

24

24

Álgebra Relacional - Operações Estendidas

- Semi-Join (Semi Junção)
 - Left Semi Join
 - Right Semi Join
- Outer Join (Junção Externa)
 - Left Outer Join
 - Right Outer Join
 - Full Outer Join
- Agregação
 - Funções de Agregação

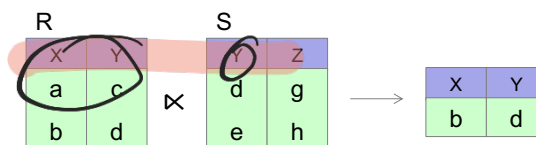
25

25

Semi Join

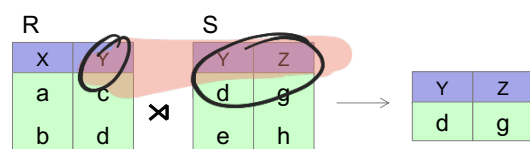
- **Left** Semi Join: $R \ltimes S = \pi_R (R \bowtie S)$

Projeção dos atributos de R na junção natural de R com S



- **Right** Semi Join: $R \rtimes S = \pi_S (R \bowtie S)$

Projeção dos atributos de S na junção natural de R com S



26

26



Inner Join vs Outer Join

Inner Join

- As operações de junção anteriores combinam dados de duas tabelas para que estes possam ser apresentados na forma de uma única tabela.
- Os tuplos que não estão relacionados (*matching*) são descartados.
 - Incluindo os tuplos com valores Null nos atributos de junção.

Outer Join

- Incluimos no resultado todos os tuplos de uma (ou de ambas) das relações componentes.
- Os atributos que não fazem *matching* são preenchidos com *Null*.

27

27



Outer Join

- Left Outer Join: $R \bowtie_{A2=B1} S$

| R | | | S | | | | | | |
|----|----|-------------------|----|----|---------------|----|----|------|------|
| A1 | A2 | | B1 | B2 | | A1 | A2 | B1 | B2 |
| a | c | $\bowtie_{A2=B1}$ | d | g | \rightarrow | a | c | null | null |
| b | d | | e | h | | b | d | d | g |

- Right Outer Join: $R \ltimes_{A2=B1} S$

| R | | | S | | | | | | |
|----|----|-------------------|----|----|---------------|------|------|----|----|
| A1 | A2 | | B1 | B2 | | A1 | A2 | B1 | B2 |
| a | c | $\ltimes_{A2=B1}$ | d | g | \rightarrow | b | d | d | g |
| b | d | | e | h | | null | null | e | h |

- Full Outer Join: $R \Join_{A2=B1} S$

| R | | | S | | | | | | |
|----|----|-----------------|----|----|---------------|------|------|------|------|
| A1 | A2 | | B1 | B2 | | A1 | A2 | B1 | B2 |
| a | c | $\Join_{A2=B1}$ | d | g | \rightarrow | a | c | null | null |
| b | d | | e | h | | b | d | d | g |
| | | | | | | null | null | e | h |

28

28

Left Outer Join - Exemplo

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1985-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

| Dname | Dnumber | Mgr_ssn | Mgr_start_date |
|----------------|---------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

$\Pi_{Fname, Minit, Lname, Dname} (EMPLOYEE \bowtie_{Ssn=Mgr_ssn} DEPARTMENT)$

| Fname | Minit | Lname | Dname |
|----------|-------|---------|----------------|
| John | B | Smith | NULL |
| Franklin | T | Wong | Research |
| Alicia | J | Zelaya | NULL |
| Jennifer | S | Wallace | Administration |
| Ramesh | K | Narayan | NULL |
| Joyce | A | English | NULL |
| Ahmad | V | Jabbar | NULL |
| James | E | Borg | Headquarters |

29

Join - Quadro Resumo

- Natural | Left Outer | Right Outer | Full Outer

R

| X | Y |
|---|---|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |

S

| Y | Z |
|---|---|
| | a |
| | b |
| | c |
| | d |

All Joins

| X | Y | Z |
|------|---|------|
| 1 | | null |
| 2 | | a |
| 3 | | null |
| 4 | | b |
| 4 | | c |
| 5 | | null |
| null | | d |

R ... S

| \bowtie | \bowtie | \bowtie | \bowtie |
|-----------|-----------|-----------|-----------|
| x | v | x | v |
| v | v | v | v |
| x | v | x | v |
| v | v | v | v |
| v | v | v | v |
| x | v | x | v |
| x | x | v | v |

30

Agregação

- Operação de Agregação

<grouping attributes> Σ <function list> (R)

Σ - Script F
symbol

- Operações sobre vários tuplos da relação

- Lista de Funções de Agregação:

- avg: média dos valores
- min: mínimo dos valores
- max: máximo dos valores
- sum: soma dos valores
- count: número dos valores

31

31

Funções de Agregação

- Também podem ser usadas em projeções

- criar atributos agregados
- os atributos não agregados são agrupados de forma a não haver valores repetidos.

- Exemplos:

$\Pi_{A1, A2, M = \text{avg}(A3)}(R)$

EMPLOYEE

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

$\Pi_{Dno, \text{Avg_Salary}=\text{avg}(\text{Salary})}(\text{EMPLOYEE})$

| Dno | Avg_Salary |
|-----|------------|
| 1 | 55000 |
| 4 | 31000 |
| 5 | 33250 |

32

32

Agregação (*Grouping*) - Exemplos

EMPLOYEE

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|-----------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

 $\Sigma \text{count(Ssn), avg(Salary)}(\text{EMPLOYEE})$

| Count_ssn | Average_salary |
|-----------|----------------|
| 8 | 35125 |

 $\text{Dno } \Sigma \text{count(Ssn), avg(Salary)}(\text{EMPLOYEE})$

| Dno | Count_ssn | Average_salary |
|-----|-----------|----------------|
| 5 | 4 | 33250 |
| 4 | 3 | 31000 |
| 1 | 1 | 55000 |

 $\rho_R(\text{Dno, No_of_employees, Average_sal}) (\text{Dno } \Sigma \text{count(Ssn), avg(Salary)}(\text{EMPLOYEE}))$

| Dno | No_of_employees | Average_sal |
|-----|-----------------|-------------|
| 5 | 4 | 33250 |
| 4 | 3 | 31000 |
| 1 | 1 | 55000 |

33

33

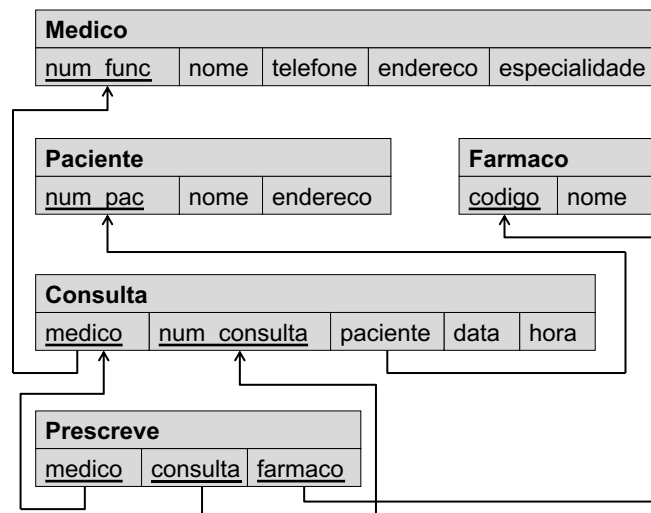
Álgebra Relacional - Queries Caso de Estudo

Clínica Médica

34

34

Clínica - Esquema Relacional da BD

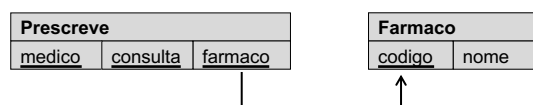


35

35

Clínica - Problema 1

- Nome dos fármacos que nunca foram prescritos



$$\Pi_{\text{nome}}(\sigma_{\text{farmaco}=\text{null}}(\text{Prescreve} \bowtie_{\text{farmaco}=\text{codigo}} \text{Farmaco}))$$

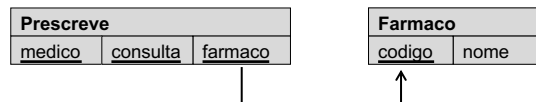
36

36



Clínica - Problema 2

- O número de fármacos prescritos em cada consulta



$\pi_{\text{medico, consulta, num_farm=count(farmaco)}}(\text{Prescreve})$

Ou

$\text{medico, consulta } \mathfrak{S} \text{ count(farmaco)}(\text{Prescreve})$

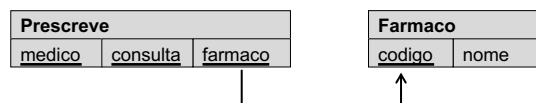
37

37



Clínica - Problema 3

- Para cada médico, a quantidade média de fármacos receitados por consulta



$\text{temp} \leftarrow \pi_{\text{medico, consulta, num_farm=count(farmaco)}}(\text{Prescreve})$

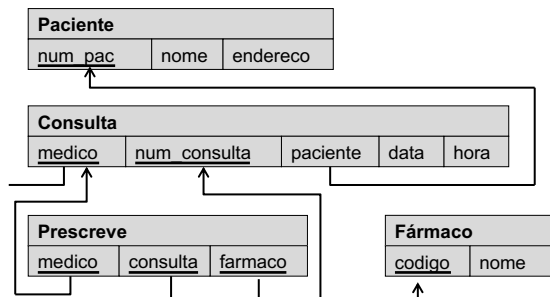
$\pi_{\text{medico, avg_farmaco=avg(num_farm)}}(\text{temp})$

38

38

Clínica - Problema 4

- O nome de todos os fármacos prescritos, incluindo a quantidade, para o paciente número 35312161



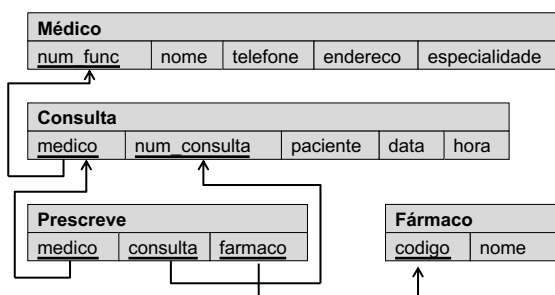
```

temp ←  $\pi_{\text{medico}, \text{num\_consulta}} (\sigma_{\text{paciente}=35312161} (\text{Consulta}))$ 
temp2 ←  $\pi_{\text{farmaco}, \text{quantidade}=\text{count}(\text{farmaco})} (\text{temp} \bowtie_{\text{medico}=\text{medico} \text{ AND } \text{num\_consulta}=\text{consulta}} \text{Prescreve})$ 
 $\pi_{\text{nome}, \text{quantidade}} (\text{temp2} \bowtie_{\text{farmaco}=\text{codigo}} \text{Farmaco})$ 
  
```

39

Clínica - Problema 5

- O nome dos fármacos que já foram prescritos por todos os médicos da clínica



```

temp ←  $(\pi_{\text{farmaco}, \text{medico}} (\text{Prescreve})) \div (\rho_{\text{medico}} (\pi_{\text{num\_func}} (\text{Medico})))$ 
 $\pi_{\text{nome}} (\rho_{\text{codigo}, \text{medico}} (\text{temp}) \bowtie \text{Farmaco})$ 
  
```

40

40

deti

A Seguir?

Data Operations – Relational Algebra

| A | B |
|----------|---|
| α | 1 |
| α | 2 |
| β | 1 |

| A | B |
|----------|---|
| α | 2 |
| β | 3 |

| A | B |
|----------|---|
| α | 1 |
| α | 2 |
| β | 1 |
| β | 3 |

Query syntax

```

SELECT <desired attributes>
FROM <one or more tables>
WHERE <predicate holds for selected tuple>
GROUP BY <key columns, aggregations>
HAVING <predicate holds for selected group>
ORDER BY <columns to sort>

```

SQL – Data Manipulation

SQL query:

```

SELECT  Pnumber, Pname, COUNT (*)
FROM    PROJECT, WORKS_ON
WHERE   Pnumber=Pno
GROUP BY Pnumber, Pname;

```

SQL query:

```

INSERT INTO EMPLOYEE (Fname,
Lname, Ssn, Dno) VALUES ('Robert',
'Hatcher', '980760540', 2);

```

SQL – Describe Database Schema

```

CREATE TABLE DEPARTMENT
( Dname          VARCHAR(15)    NOT NULL,
  Dnumber        INT            NOT NULL,
  Mgr_ssn        CHAR(9)       NOT NULL,
  Mgr_start_date DATE,
  PRIMARY KEY (Dnumber),
  UNIQUE (Dname),
  FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );

```

The Relational Schema

Part (Name,Description,Part#)

Supplier (Name, Addr)

Customer (Name, Addr)

Supplies (Name,Part#, Date)

Orders (Name,Part#)

41

41

deti

Resumo

Álgebra Relacional:

- Operações Básicas
- Operações Estendidas
- Caso de Estudo - Queries

42

42