

Álgebra Relacional

Base de Dados - 2023/24

Carlos Costa

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

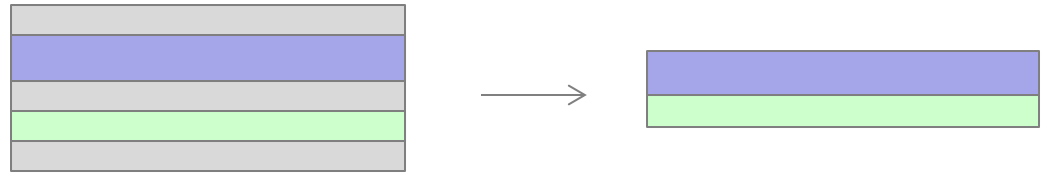
Álgebra Relacional

Questões?

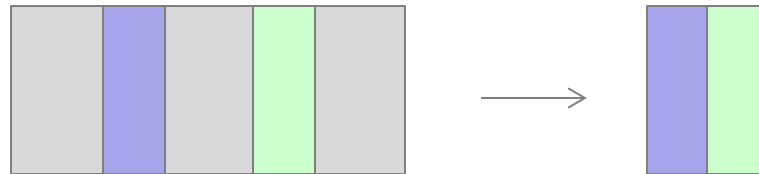
- Como deve ser uma linguagem de interrogação da BD?
- Que tipo de interrogações existem?
- Como é que são os resultados?
- Expressões de álgebra relacional (linguagem).
 - Sequência de operações de álgebra relacional.
 - Permitem formular pedidos básicos de recuperação de informação sobre uma ou mais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

Álgebra Relacional - Operações Básicas

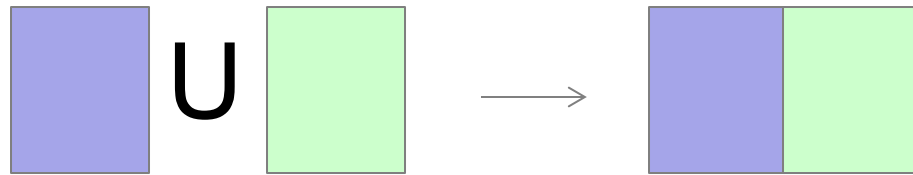
- Seleção



- Projeção



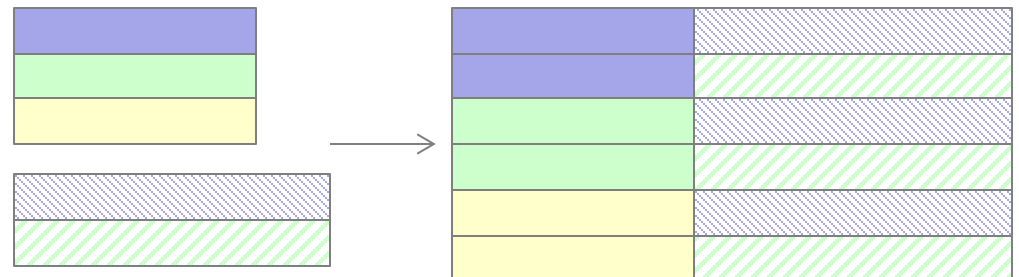
- União



- Diferença

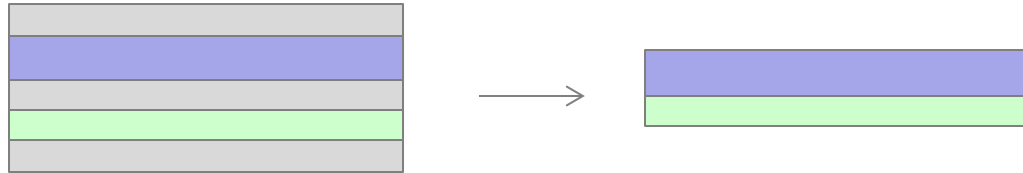


- Produto Cartesiano



- Renomeação

Seleção



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

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

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

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:

$\sigma_{Dno=4} (EMPLOYEE)$

$\sigma_{Salary>30000} (EMPLOYEE)$

- Condições Booleanas

- Utilização de AND, OR e NOT.
- Exemplo:

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

Seleção - Exemplo

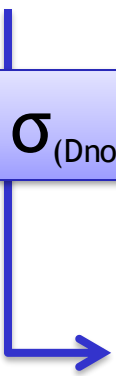
EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | 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 |

SQL query (próxima aula...)

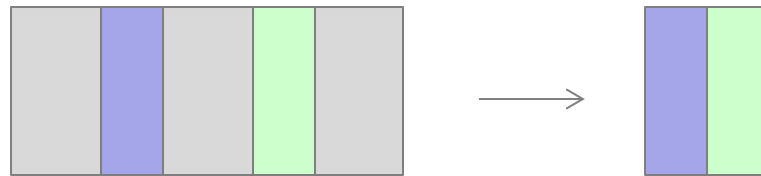
$\sigma_{(Dno=4 \text{ AND } Salary>25000) \text{ OR } (Dno=5 \text{ AND } Salary>30000)} (EMPLOYEE)$

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



| Fname | Minit | Lname | <u>Ssn</u> | 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 |

Projeção



- Notação: $\Pi_{\langle \text{attribute list} \rangle}(R)$
 - $\langle \text{attribute list} \rangle = A1, A2, \dots Ak$
 - $A1 \dots Ak$ 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*)

Projeção - Exemplo

EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | 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 |

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

| 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 |

SQL query:

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

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 |

Renomeação

- Notação: $\rho_{R2(B1, B2, \dots, Bn)}(R1)$ ou $\rho_{R2}(R1)$

ou $\rho_{(B1, B2, \dots, Bn)}(R1)$

- 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

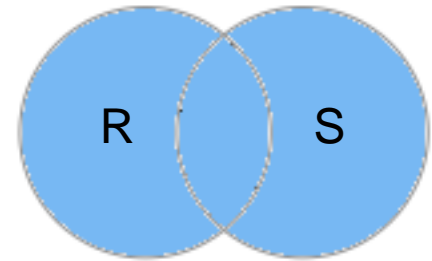
Fname -> First_name

Lname -> Last_Name

...

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

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

U

INSTRUCTOR

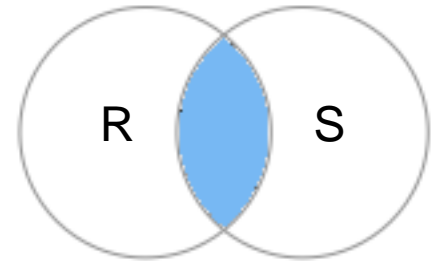
| Fname | Lname |
|---------|---------|
| John | Smith |
| Ricardo | Browne |
| Susan | Yao |
| Francis | Johnson |
| Ramesh | Shah |



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

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 |

\cap

INSTRUCTOR

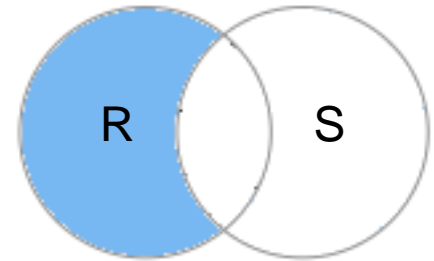
| Fname | Lname |
|---------|---------|
| John | Smith |
| Ricardo | Browne |
| Susan | Yao |
| Francis | Johnson |
| Ramesh | Shah |



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

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 |

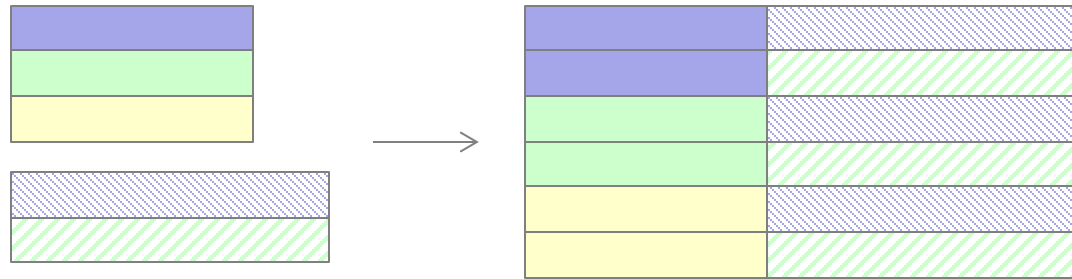
União, Intersecção e Diferença

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

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 $(R \cap S) \cup T = R \cap (S \cup T)$

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”

Produto Cartesiano - Exemplo

EMPNames

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

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 |

| 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 | ... |

EMPNames X DEPENDENT

Junção θ (THETA JOIN)

- Notação: $R \bowtie_c S$

- Pode ser visto como o resultado das seguintes operações:

$R3 \leftarrow R1 \times R2$ (produto cartesiano)

$\sigma_c (R3)$ (seleção com condição c)

- C é <join condition> que pode tomar a seguinte forma:
<condition> AND <condition> AND ... AND <condition>
- Em cada <condition> podemos aplicar operadores de comparação:
 $=, <, \leq, >, \geq, \neq$

Junção θ (THETA JOIN) - Exemplo

- Pretendemos saber os nomes dos funcionários gestores de departamentos

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 |

DEPARTMENT

| 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.

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

DEPT_MGR

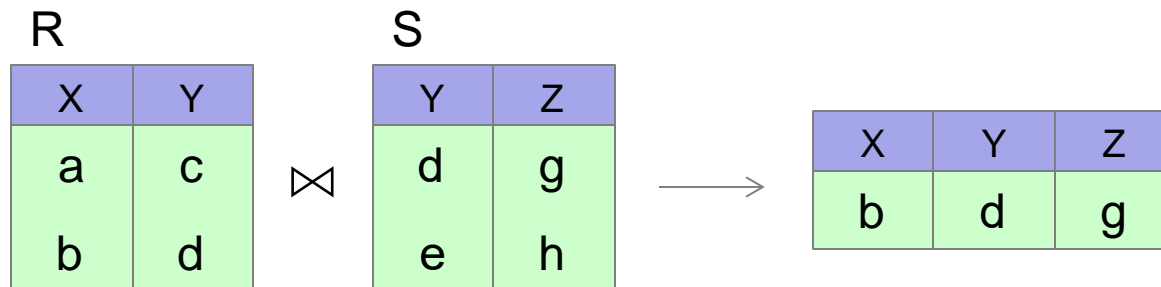
| Dname | Dnumber | Mgr_ssn | ... | Fname | Minit | Lname | Ssn | ... |
|----------------|---------|-----------|-----|----------|-------|---------|-----------|-----|
| Research | 5 | 333445555 | ... | Franklin | T | Wong | 333445555 | ... |
| Administration | 4 | 987654321 | ... | Jennifer | S | Wallace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

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

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

Junção - Variações da Junção θ

- Equi-Junção (EquiJoin)
 - É utilizado o operador $=$ na condição de junção.
 - Exemplo anterior: $\text{DEPARTMENT} \bowtie_{\text{Mgr_ssn}=\text{Ssn}} \text{EMPLOYEE}$.
 - 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.



Junção Natural - Exemplo

PROJECT

| Pname | <u>Pnumber</u> | 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 |

DEPARTMENT

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



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



| Pname | <u>Pnumber</u> | 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 |

Divisão

- Notação: $R \div S$

Dadas as relações $R(A_1, \dots, A_r, B_1, \dots, B_k)$ e $S(B_1, \dots, B_k)$

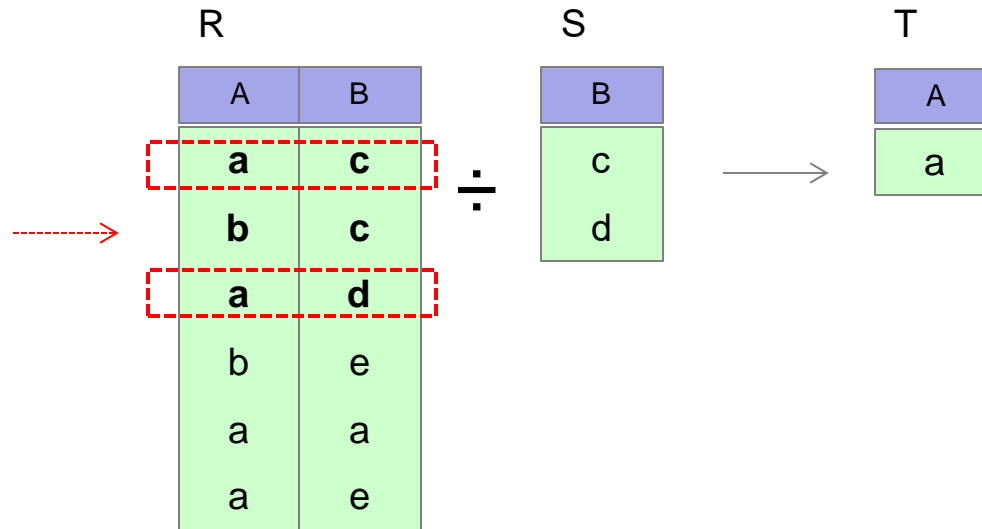
- O resultado incluirá todos os tuplos de $R(A_1, \dots, A_r)$ que tenham correspondência com todos os tuplos de S em $R(B_1, \dots, B_k)$.
 - R_1 e R_2 são projeções de R
- número de atributos de $R >$ número de atributos de S .

- 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 $\pi_{R-S} \rightarrow \pi_{(A_1, \dots, A_r)}$

Divisão - Exemplos



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?

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 $Z = X \cup Y$. | $R_1(Z) \div R_2(Y)$ |

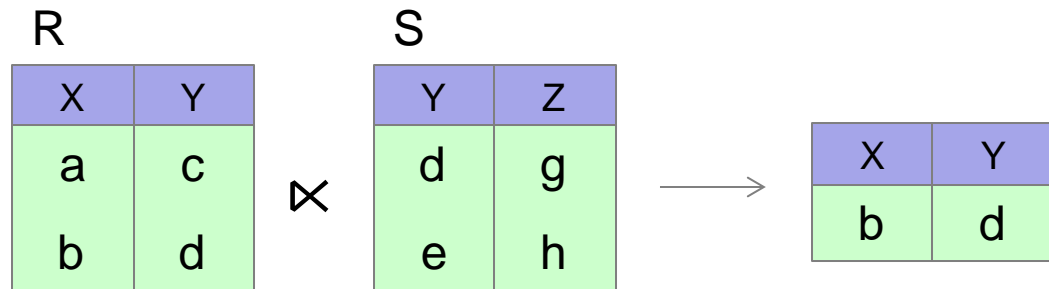
Álgebra Relacional - Operações Estendidas

- 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

Semi Join

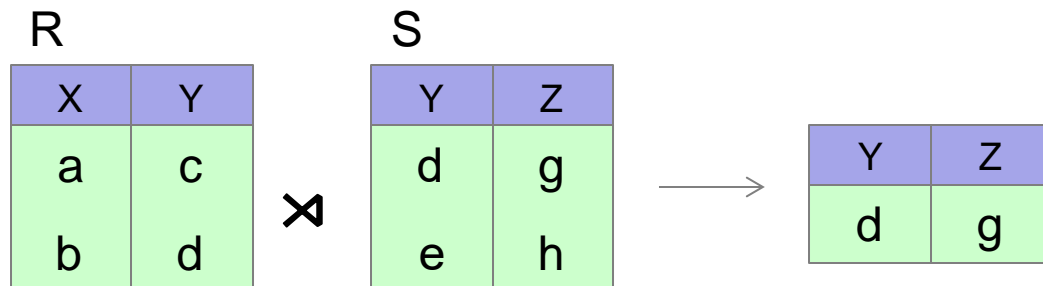
- **Left** Semi Join: $R \ltimes S = \pi_R (R \bowtie S)$

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



Inner Join vs Outer Join

Inner Join

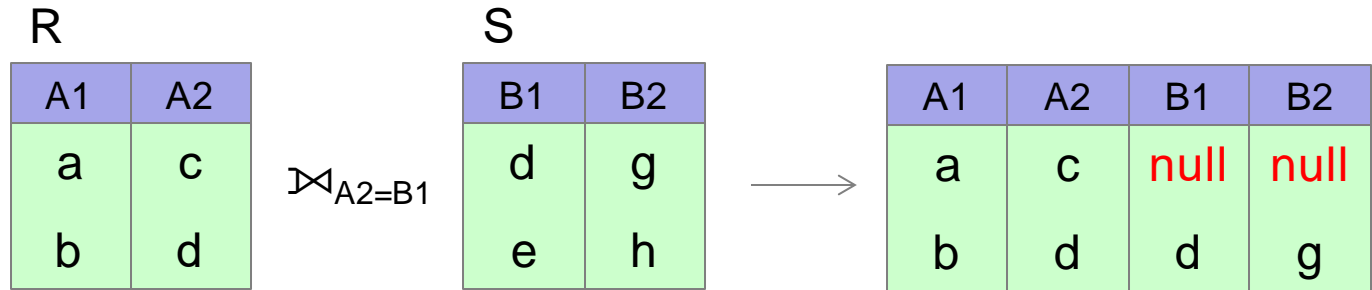
- As operações de junção anteriores combinam dados de duas tabelas para que estes possam ser apresentados na forma de uma única tabela.
- Os tuplos que não estão relacionados (*matching*) são descartados.
 - Incluindo os tuplos com valores Null nos atributos de junção.

Outer Join

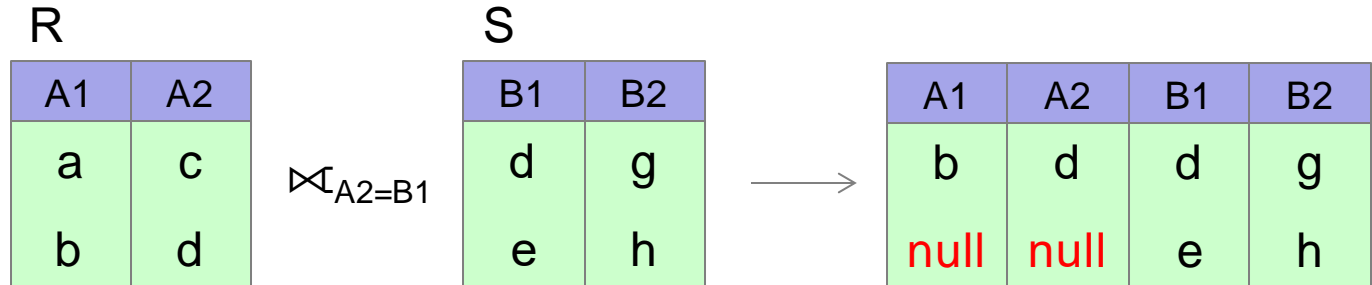
- Incluimos no resultado todos os tuplos de uma (ou de ambas) das relações componentes.
- Os atributos que não fazem *matching* são *preenchidos com Null*.

Outer Join

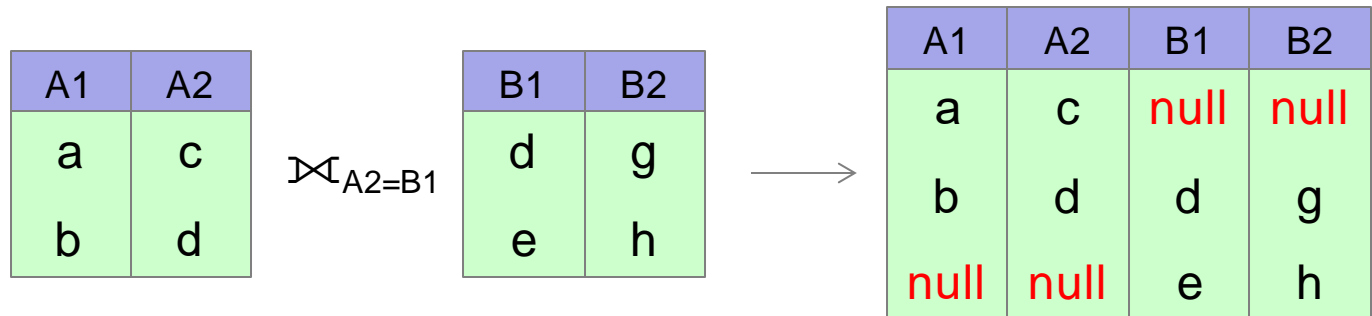
- Left Outer Join: $R \bowtie S$



- Right Outer Join: $R \bowtie S$



- Full Outer Join: $R \bowtie S$



Left Outer Join - Exemplo

EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | 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 |

DEPARTMENT

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



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



| 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 |

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 |

Agregação

- Operação de Agregação

<grouping attributes> \mathfrak{F} <function list> (R)

\mathfrak{F} - 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

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 | <u>Ssn</u> | 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 |

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 |

Σ count(Ssn), avg(Salary) (EMPLOYEE)

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

Dno Σ count(Ssn), avg(Salary) (EMPLOYEE)

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

$\rho_{R(Dno, No_of_employees, Average_sal)}$ (Dno Σ count(Ssn), avg(Salary) (EMPLOYEE))

R

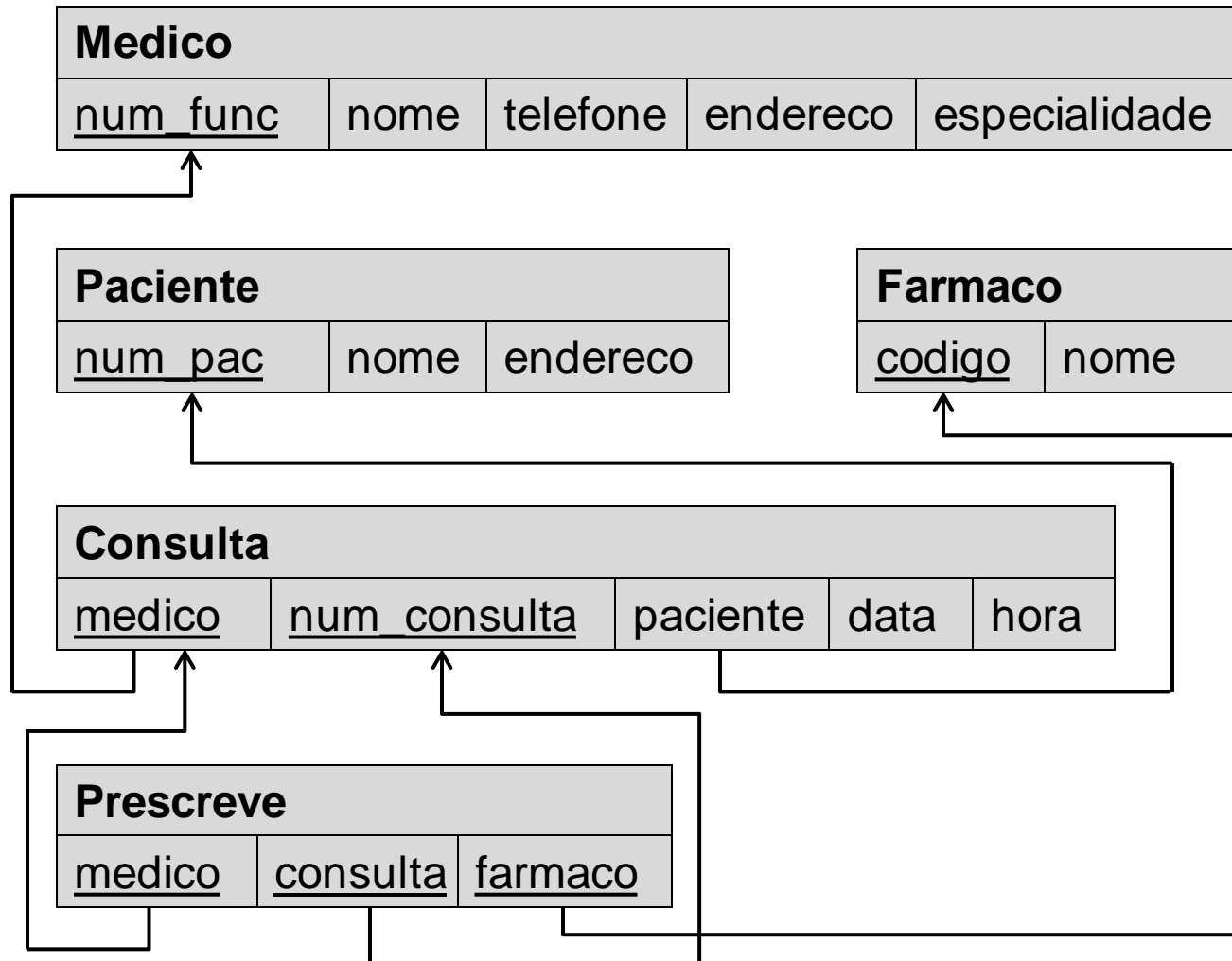
| Dno | No_of_employees | Average_sal |
|-----|-----------------|-------------|
| 5 | 4 | 33250 |
| 4 | 3 | 31000 |
| 1 | 1 | 55000 |

Álgebra Relacional - Queries

Caso de Estudio

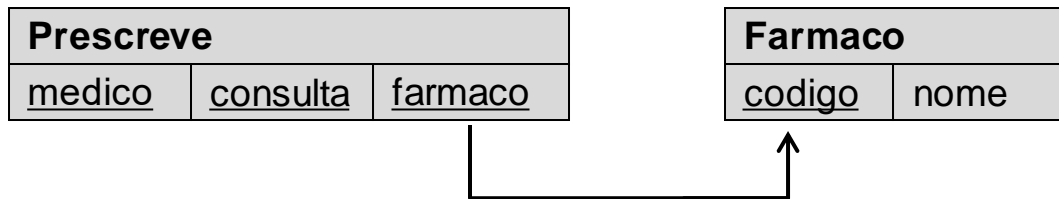
Clínica Médica

Clínica - Esquema Relacional da BD



Clínica - Problema 1

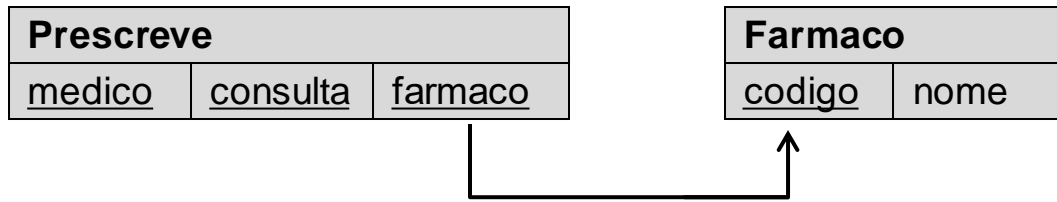
- Nome dos fármacos que nunca foram prescritos



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

Clínica - Problema 2

- O número de fármacos prescritos em cada consulta



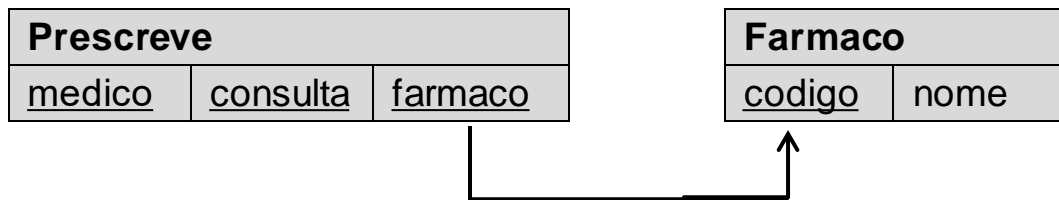
$\Pi_{\text{medico, consulta, num_farm=count(farmaco)}} (\text{Prescreve})$

Ou

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

Clínica - Problema 3

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

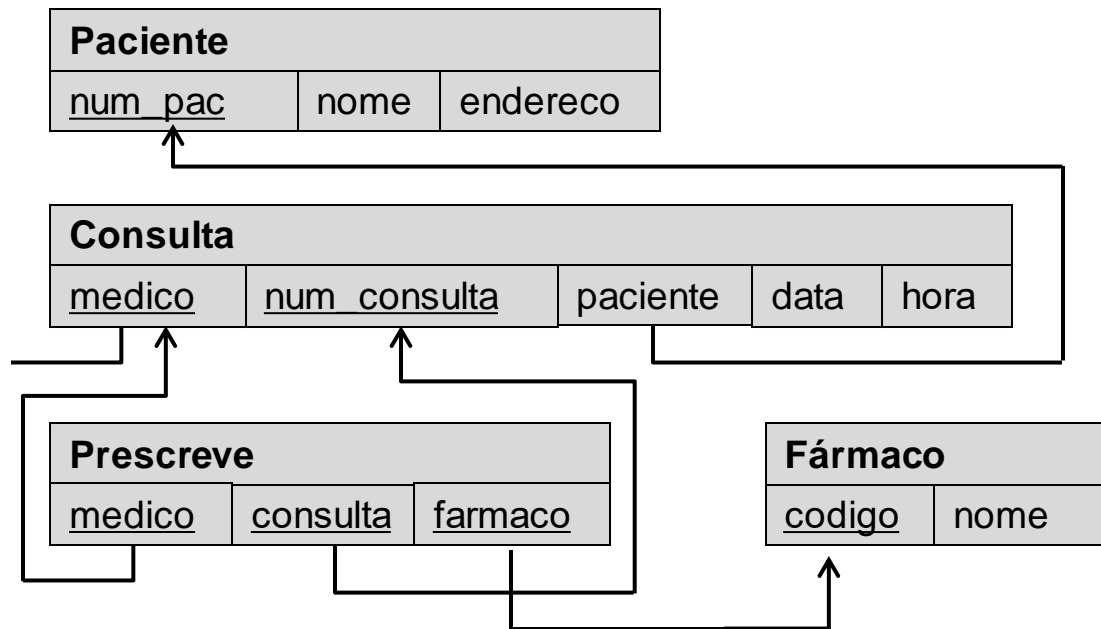


$\text{temp} \leftarrow \Pi_{\text{medico}, \text{consulta}, \text{num_farm}=\text{count}(\text{farmaco})} (\text{Prescreve})$

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

Clínica - Problema 4

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



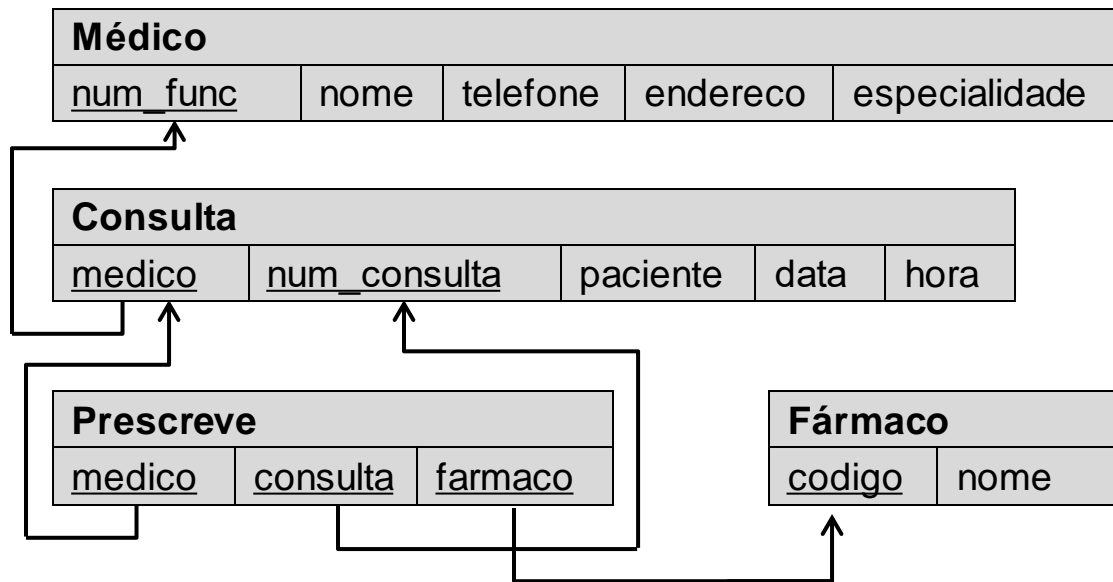
$\text{temp} \leftarrow \pi_{\text{medico}, \text{num_consulta}} (\sigma_{\text{paciente}=35312161} (\text{Consulta}))$

$\text{temp2} \leftarrow \pi_{\text{farmaco}, \text{quantidade}=\text{count}(\text{farmaco})} (\text{temp} \bowtie_{\text{medico}=\text{medico} \text{ AND } \text{num_consulta}=\text{consulta}} \text{Prescreve})$

$\pi_{\text{nome}, \text{quantidade}} (\text{temp2} \bowtie_{\text{farmaco}=\text{codigo}} \text{Farmaco})$

Clínica - Problema 5

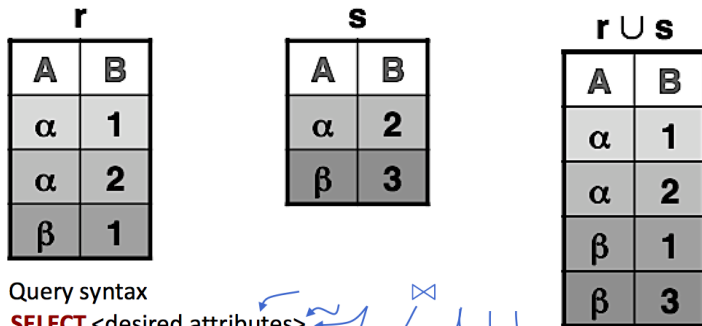
- O nome dos fármacos que já foram prescritos por todos os médicos da clínica



$$\begin{aligned}
 \text{temp} &\leftarrow (\pi_{\text{farmaco}, \text{medico}}(\text{Prescreve})) \div (\rho_{\text{medico}}(\pi_{\text{num_func}}(\text{Medico}))) \\
 &\pi_{\text{nome}}(\rho_{\text{codigo}, \text{medico}}(\text{temp}) \bowtie \text{Farmaco})
 \end{aligned}$$

A Seguir?

Data Operations – Relational Algebra



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#)

Resumo

Álgebra Relacional:

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