

Introdução a Banco de Dados

Revisão

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br

Álgebra relacional

Operações unárias

- Seleção (σ)
- Projeção (π)
- Renomeação (ρ)

Operações binárias

- União (\cup)
- Interseção (\cap)
- Diferença ($-$) *aditivas*
- Prod. cartesiano (\times)
- Junção (\bowtie)
- Divisão (\div) *multiplicativas*

Seleção (σ)

Recupera as tuplas de uma relação R que satisfazem uma determinada condição lógica [$cond$]

- Notação: $\sigma_{[cond]}(R)$

Exemplo:

- $\sigma_{(Dept=28 \wedge Salário>600) \vee (Dept=21)}(Empregado)$

Projeção (π)

Projeta as tuplas de uma relação R sobre um determinado conjunto de atributos [$atrib$]

- Notação: $\pi_{[atrib]}(R)$

Exemplos:

- $\pi_{\text{Nome, Salário}}(\text{Empregado})$
- $\pi_{\text{Dept}}(\text{Empregado})$

Renomeação (ρ)

Cria um novo esquema $S(A_1, \dots, A_n)$ para a relação R

- Notação 1: $\rho_{S(A_1, \dots, A_n)}(R)$
- Notação 2: $S(A_1, \dots, A_n) \leftarrow (R)$

Exemplo:

- $\rho_{\text{Funcionário}(\text{ID}, \text{Nome}, \text{Renda}, \text{Dept})}(\text{Empregado})$

Operações aditivas vs. multiplicativas

Operações aditivas vs. multiplicativas

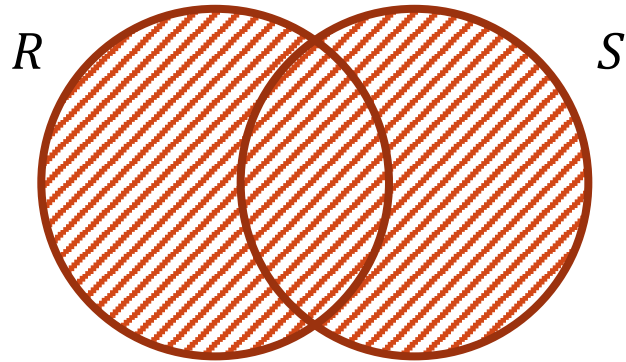
Operações aditivas

- União (\cup)
- Interseção (\cap)
- Diferença ($-$)

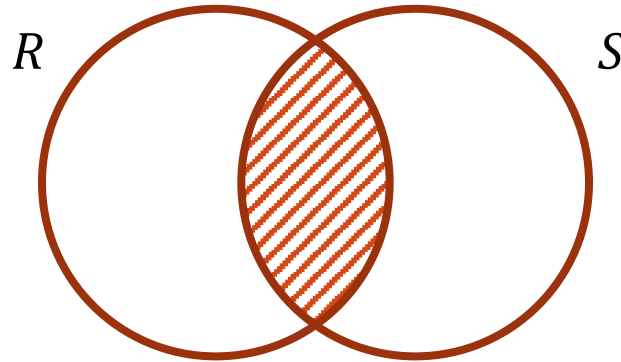
Esquema preservado!

Operações aditivas

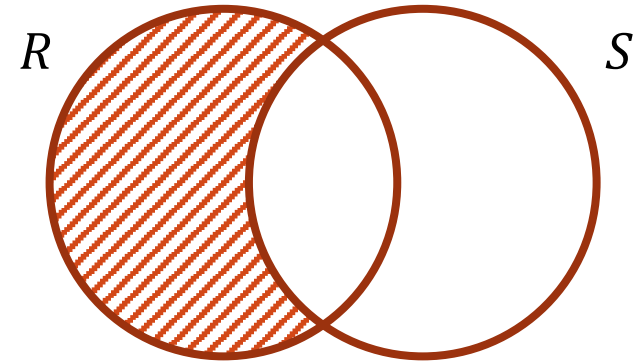
União (\cup)



Interseção (\cap)



Diferença ($-$)



Operações aditivas vs. multiplicativas

Operações aditivas vs. multiplicativas

Operações multiplicativas

- Prod. cartesiano (\times)
- Junção (\bowtie)
- Divisão (\div)

Esquema alterado!

Produto cartesiano (\times)

Dadas duas relações R e S quaisquer, combina cada tupla de R com cada tupla de S

- Notação: $R \times S$

Características

- $|R \times S| = |R| |S|$
- $\text{grau}(R \times S) = \text{grau}(R) + \text{grau}(S)$

Junção (\bowtie)

Dadas duas relações R e S quaisquer, combina tuplas de R e S que satisfazem uma condição lógica $[cond]$

- Notação: $R \bowtie_{[cond]} S$

Características

- Equivalente a $\sigma_{[cond]}(R \times S)$

Junção natural (\bowtie)

Condição de junção implícita

- Teste de igualdade em atributos de mesmo nome
- Não necessariamente possuem a mesma semântica

Atributos de mesmo nome retornados uma única vez

- $\text{grau}(R \bowtie S) = \text{grau}(R) + \text{grau}(S) - k$
- k : número de atributos de mesmo nome em R e S

Junção externa

Dadas duas relações R e S quaisquer, combina tuplas de R e S que satisfazem uma condição lógica $[cond]$, e possivelmente tuplas de R ou S que não a satisfazem

- Junção externa à esquerda: $R \bowtie_{[cond]} S$
- Junção externa à direita: $R \bowtie_{[cond]} S$
- Junção externa completa: $R \bowtie_{[cond]} S$

Divisão

Dadas duas relações $R(A_1, \dots, A_m)$ e $S(B_1, \dots, B_n)$, tais que $\{A_1, \dots, A_m\} \supset \{B_1, \dots, B_n\}$, retorna uma projeção Q de R sobre $\{A_1, \dots, A_m\} - \{B_1, \dots, B_n\}$, tal que cada tupla de Q se relaciona a todas as tuplas de S

- Notação: $Q = R \div S$

Esquemas resultantes

Operações unárias

- **Seleção:** esquema preservado
- **Projeção:** esquema modificado pela operação
- **Renomeação:** esquema redefinido pela operação

Operações binárias aditivas

- **União, interseção, diferença:** esquema preservado

Esquemas resultantes

Operações binárias multiplicativas

- **Produto cartesiano:** esquema combina os esquemas das relações participantes do produto
- **Junção:** idem ao produto cartesiano
- **Junção natural:** idem à junção, exceto pelo fato de que atributos de mesmo nome só aparecem uma vez

SQL

LDD – Linguagem Definição de Dados

- CREATE SCHEMA / TABLE / VIEW
- DROP SCHEMA / TABLE / VIEW
- ALTER TABLE

LMD – Linguagem Manipulação de Dados

- INSERT, UPDATE, DELETE

LC – Linguagem de Consulta

- SELECT

Criação de esquemas

```
CREATE SCHEMA COMPANY AUTHORIZATION JS;
```

```
CREATE TABLE <nome da tabela>
```

```
    (<definições de colunas>
```

```
        <definição da chave primária>
```

```
        <definições de chaves alternativas>
```

```
        <definições de chaves estrangeiras>);
```

Criação de esquemas

Opções de remoção (cláusula **ON DELETE**):

- **CASCADE** (propagação)
- **SET NULL** (substituição por nulos)
- **SET DEFAULT** (substituição por um valor default)
- **RESTRICT** (bloqueio – opção default)

As mesmas opções se aplicam à cláusula **ON UPDATE**

Ateração de esquemas

DROP SCHEMA COMPANY CASCADE (RESTRICT);

-- RESTRICT: se não tem elementos

DROP TABLE DEPENDENT CASCADE (RESTRICT);

-- RESTRICT: se não é referenciada

Alteração de esquemas

```
ALTER TABLE COMPANY.EMPLOYEE  
ADD JOB VARCHAR(12);
```

-- Inicialmente **NULL** para todas as tuplas

```
ALTER TABLE COMPANY.EMPLOYEE  
DROP ADDRESS CASCADE (RESTRICT);
```

-- **RESTRICT**: se não é referenciada

Consultas básicas

```
SELECT PNUMBER, DNUM, LNAME, ADDRESS, BDATE  
FROM PROJECT, DEPARTMENT, EMPLOYEE  
WHERE PLOCATION='Stafford'  
      AND DNUM=DNUMBER AND MGRSSN=SSN;
```

```
SELECT SSN, LNAME, SALARY  
FROM EMPLOYEE;
```



sem seleção

```
SELECT LNAME, DNAME  
FROM EMPLOYEE, DEPARTMENT;
```



prod. cartesiano

```
SELECT * FROM EMPLOYEE;
```



sem projeção

```
SELECT DISTINCT SALARY FROM EMPLOYEE;
```

Consultas complexas

```
SELECT FNAME, LNAME, ADDRESS
FROM EMPLOYEE
WHERE DNO IN
    (SELECT DNUMBER FROM DEPARTMENT WHERE DNAME='Research');
```

```
SELECT LNAME, FNAME
FROM EMPLOYEE
WHERE SALARY > ALL
    (SELECT SALARY FROM EMPLOYEE WHERE DNO=5);
```

```
SELECT E.FNAME, E.LNAME
FROM EMPLOYEE AS E
WHERE EXISTS
    (SELECT * FROM DEPENDENT
     WHERE E.SSN=ESSN AND E.FNAME=DEPENDENT_NAME);
```

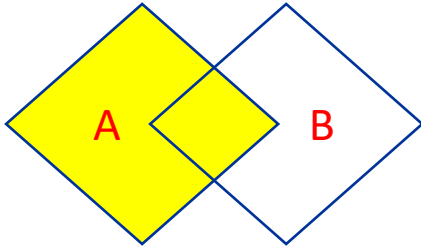

Junções

```
SELECT FNAME, LNAME, ADDRESS  
FROM EMPLOYEE JOIN DEPARTMENT ON DNO=DNUMEBR  
WHERE DNAME='Research';
```

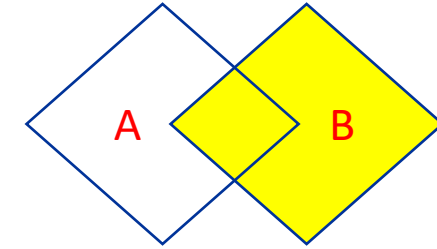
```
SELECT DNAME, DLOCATION  
FROM DEPARTMENT NATURAL JOIN DEPT_LOCATIONS;
```

```
SELECT fname, lname, dependent_name  
FROM EMPLOYEE LEFT JOIN DEPENDENT ON ssn=essn;
```

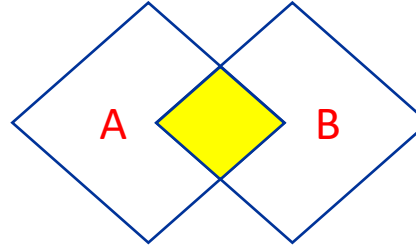
SQL JOINS



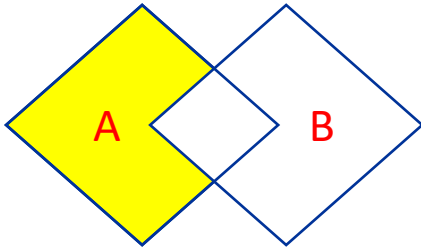
SELECT ...
FROM A LEFT JOIN B
ON A.key = B.key



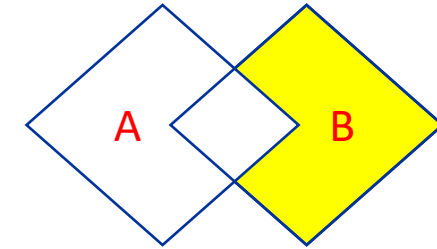
SELECT ...
FROM A RIGHT JOIN B
ON A.key = B.key



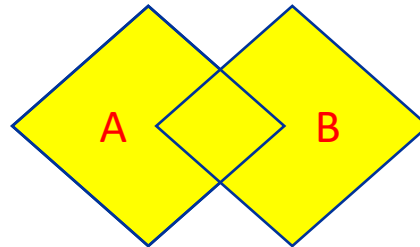
SELECT ...
FROM A INNER JOIN B
ON A.key = B.key



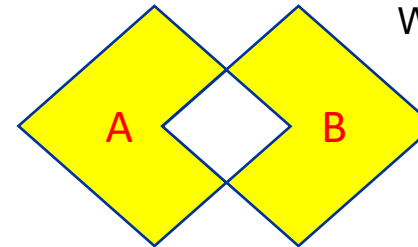
SELECT ...
FROM A LEFT JOIN B
ON A.key = B.key
WHERE B.key IS NULL



SELECT ...
FROM A RIGHT JOIN B
ON A.key = B.key
WHERE A.key IS NULL



SELECT ...
FROM A FULL OUTER JOIN B
ON A.key = B.key



SELECT ...
FROM A FULL OUTER JOIN B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL

Agrupamento

Exemplo: média de salário em cada departamento

**SELECT DNO, COUNT(*), AVG(SALARY)
FROM EMPLOYEE GROUP BY DNO;**

(a)



| PNOME | MINICIAL | LNOME | <u>SSN</u> | • • • | SALARIO | SUPERSSN | DNO |
|----------|----------|---------|------------|-------|---------|-----------|-----|
| John | B | Smith | 123456789 | • • • | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | | 40000 | 888665555 | 5 |
| Ramesh | K | Narayan | 666884444 | | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | | 25000 | 333445555 | 5 |
| Alicia | J | Zelaya | 999887777 | | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | | 43000 | 888665555 | 4 |
| Ahmad | V | Jabbar | 987987987 | | 25000 | 987654321 | 4 |
| James | E | Bong | 888665555 | | 55000 | null | 1 |

| DNO | COUNT (*) | AVG (SALARIO) |
|-----|-----------|---------------|
| 5 | 4 | 33250 |
| 4 | 3 | 31000 |
| 1 | 1 | 55000 |

Resultado da Q24

Agrupamento das tuplas EMPREGADO por meio do valor de DNO

Agrupamento condicional

```
SELECT PNUMBER, PNAME, COUNT(*)  
FROM PROJECT, WORKS_ON  
WHERE PNUMBER=PNO  tuplas  
GROUP BY PNUMBER, PNAME  
HAVING COUNT(*) > 2;  grupos
```

Modificação de dados

```
INSERT INTO EMPLOYEE  
VALUES ('Richard', 'K', 'Marini', '653258653',  
'1962-12-30', '98 Oak Forest, Katy, TX',  
37000, '987654321', 4);
```

```
DELETE FROM EMPLOYEE  
WHERE LNAME='Brown';
```

```
UPDATE PROJECT  
SET PLOCATION='Bellaire', DNUM=5  
WHERE PNUMBER=10;
```