

Bases de Dados

SQL: structured query language

A linguagem do modelo relacional

Introdução

Pedro Furtado

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia da Universidade de Coimbra
2021/2022

Para que serve o SQL

- A ideia é conseguirmos fazer tudo o que nos possamos lembrar sobre os dados, operações e utilizadores com comandos simples

Manipulação e utilização de dados relacionais

- **Definição de dados** (criar e alterar tabelas, vistas, etc)
- **Integridade** (definir regras de integridade dos dados)
- **Interrogação** (pesquisar dados em tabelas)
- **Manipulação de dados** (inserir, alterar, apagar dados)
- **Autorizações e segurança** (definir privilégios, perfis, etc)
- **Controlo de transacções** (iniciar e terminar transacções)

SQL
(Structured Query Language)

SQL é um standard ANSI e ISO desde 1986

Manipulação e utilização de dados relacionais

- **Interrogação** (pesquisar dados em tabelas)
 - **Select * from emp where sal>2000**
- **Manipulação de dados** (inserir, alterar, apagar dados)
 - **Insert into tableX, delete from tableX, update tableX set x=y where, ...**
- **Definição de dados** (criar e alterar tabelas, vistas, etc)
 - **Create table, alter table, drop column,**
- **Integridade** (definir regras de integridade dos dados)
 - **Primary key, foreign key, references, not null, check(sal>2000), ...**

SQL é um standard ANSI e ISO desde 1986

Manipulação e utilização de dados relacionais

....

- **Autorizações e segurança** (definir privilégios, perfis, etc)
 - **Grant login to userX, Grant all to userX, revoke all from userX, Grant select, update, delete to userR**
- **Controlo de transacções** (iniciar e terminar transacções)
 - **Start transaction commit**

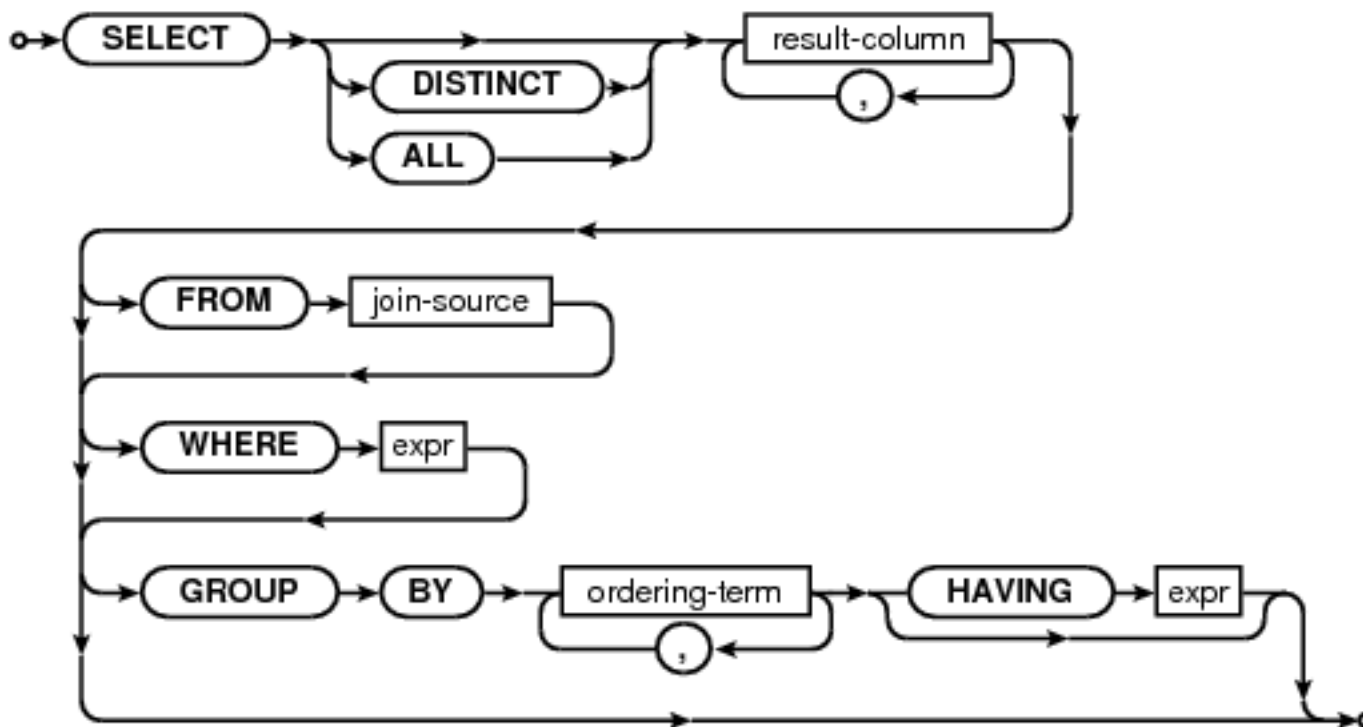
Porque o SQL é tão útil?



- Note que a “entrada” são tabelas, mas a “saída” também são tabelas
- Então eu posso encadear operações!!!!
- Posso fazer até dezenas ou centenas encadeadas

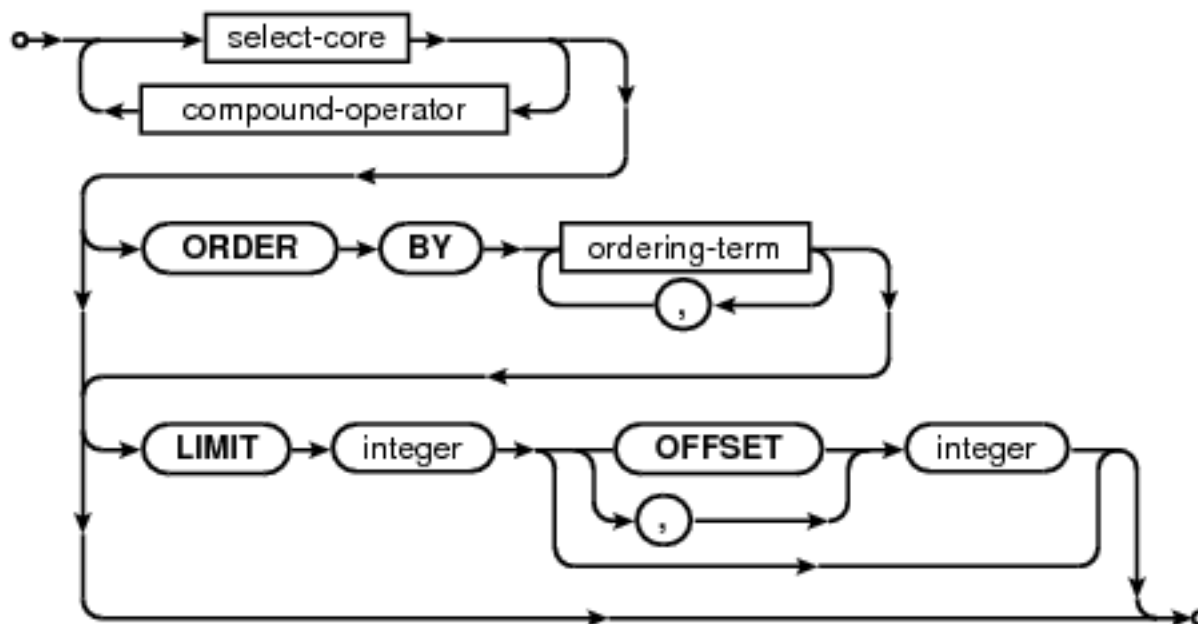
Vamos ver primeiro a interrogação

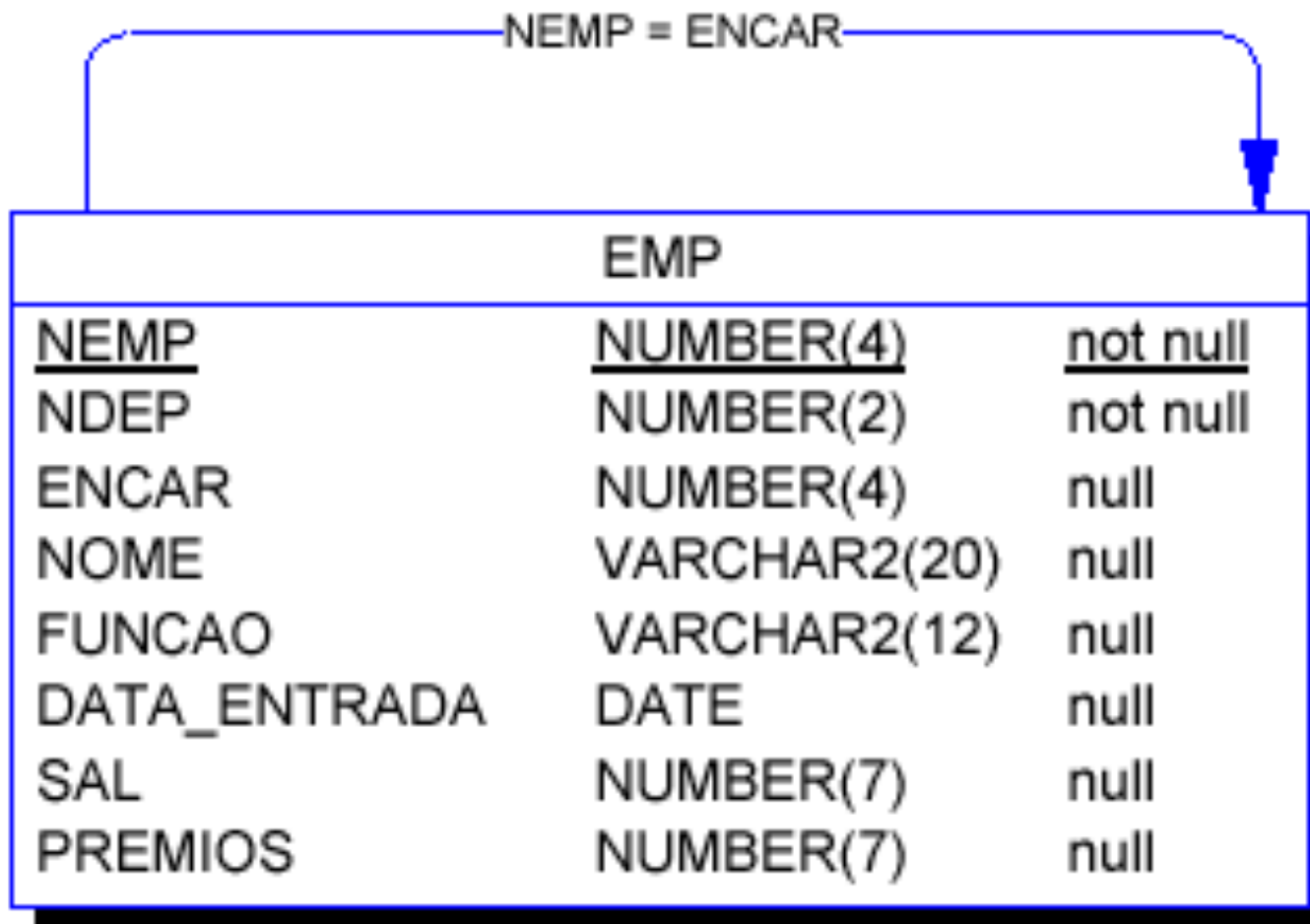
• **Interrogação** (pesquisar dados em tabelas)



Vamos ver primeiro a interrogação

• **Interrogação** (pesquisar dados em tabelas)

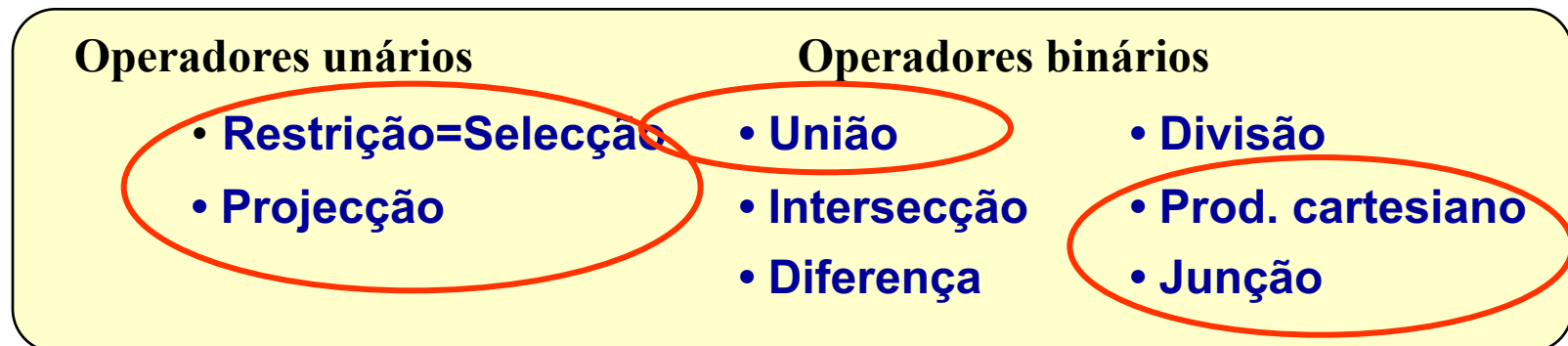




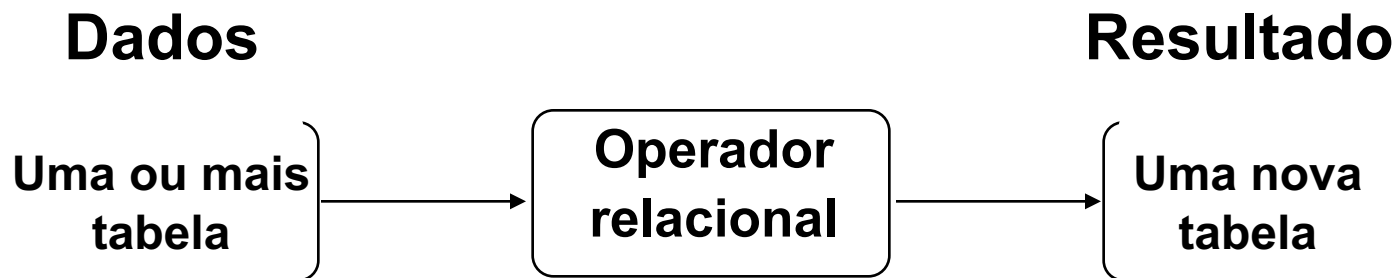
Operações relacionais (breve introdução)



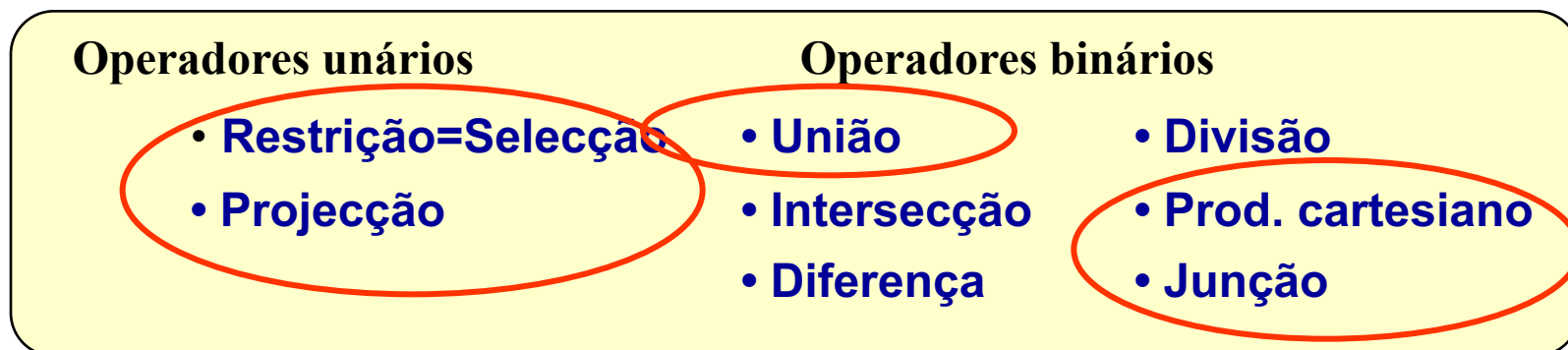
Operadores relacionais



Operações relacionais (breve introdução)



Operadores relacionais



Exemplo simples do comando SELECT

Ver todos os dados de uma tabela

```
SQL> select * from emp;
```

NEMP	NOME	FUNCAO	ENCAR	DATA_ENT	SAL	PREMIOS	NDEP
1839	Jorge Sampaio	Presidente		84.02.11	890000		10
1566	Augusto Reis	Encarregado	1839	85.02.13	450975		20
1698	Duarte Guedes	Encarregado	1839	91.11.25	380850		30
1782	Silvia Teles	Encarregado	1839	86.11.03	279450		10
1788	Maria Dias	Analista	1566	82.11.07	565000		20
1902	Catarina Silva	Analista	1566	93.04.13	435000		20
1499	Joana Mendes	Vendedor	1698	84.10.04	145600	56300	30
1521	Nelson Neves	Vendedor	1698	83.02.27	212250	98500	30
1654	Ana Rodrigues	Vendedor	1698	90.12.17	221250	81400	30
1844	Manuel Madeira	Vendedor	1698	85.04.21	157800	0	30
1900	Tome Ribeiro	Continuo	1698	94.03.05	56950		30
1876	Rita Pereira	Continuo	1788	96.02.07	65100		20
1934	Olga Costa	Continuo	1782	86.06.22	68300		10
1369	Antonio Silva	Continuo	1902	96.12.22	70800		20

14 rows selected.

Restrição ou Selecção

Operação que permite seleccionar registo(s) de uma tabela que satisfazem uma dada condição.

```
SELECT *
FROM emp
WHERE ndep = 10;
```

Mostra todos os atributos da tabela **emp**

Mas apenas para os registos que verificam a condição de **ndep = 10**

Resultado:

NEMP	NOME	FUNCAO	ENCAR	DATA_ENT	SAL	PREMIOS	NDEP
1839	Jorge Sampaio	Presidente		84.02.11	890000		10
1782	Silvia Teles	Encarregado	1839	86.11.03	279450		10
1934	Olga Costa	Continuo	1782	86.06.22	68300		10

Podem ser criadas condições múltiplas usando os operadores AND, OR e NOT.

Projecção

Operação que permite seleccionar uma ou mais colunas de uma tabela e criar uma nova tabela (resultado).

```
SELECT nome, funcao  
FROM emp;
```

Resultado:

NOME	FUNCAO
-----	-----
Jorge Sampaio	Presidente
Augusto Reis	Encarregado
Duarte Guedes	Encarregado
Silvia Teles	Encarregado
Maria Dias	Analista
Catarina Silva	Analista
Joana Mendes	Vendedor
Nelson Neves	Vendedor
Ana Rodrigues	Vendedor
Manuel Madeira	Vendedor
Tome Ribeiro	Continuo
Rita Pereira	Continuo
Olga Costa	Continuo
Antonio Silva	Continuo
14 rows selected.	

Mostra apenas os atributos
nome e funcao da tabela **emp**

Mostra todos os registos
existentes na tabela pois não
há qualquer restrição.

Restrição múltipla e projecção

Pode-se combinar as operações de restrição e de projecção.

```
SELECT nome, funcao, sal  
FROM emp  
WHERE ndep = 10 AND sal > 200;
```

Projecção: mostra apenas os atributos **nome, funcao e sal** da tabela **emp**

Restrição: Mostra apenas os registos que satisfazem a condição de ser do departamento 10 e ter salário maior do que 200.

Resultado:

NOME	FUNCAO	SAL
-----	-----	-----
Jorge Sampaio	Presidente	890000
Silvia Teles	Encarregado	279450
Olga Costa	Continuo	68300

AND, OR e precedência

Operação que permite seleccionar registo(s) de uma tabela que satisfazem um dada condição.

```
SELECT *  
FROM emp  
WHERE ndep = 10 OR  
      data_entrada > '80.01.01' AND sal > 60000;
```

```
SELECT *  
FROM emp  
WHERE ndep = 10 AND  
      (data_entrada > '80.01.01' OR sal > 60000 );
```

PERGUNTA: a primeira inclui indivíduos fora do departamento 10? E a segunda?

Introdução

SET THEORY AND RELATIONAL ALGEBRA

Where does SQL come from?

- SQL is an interface language for humans to write set operations
- Its maths basis is set theory, where sets are “relations” and relational operators

Set Theory (relational algebra)

- Set = collection of objects (members or elements).
- All mathematical objects can be put into sets.

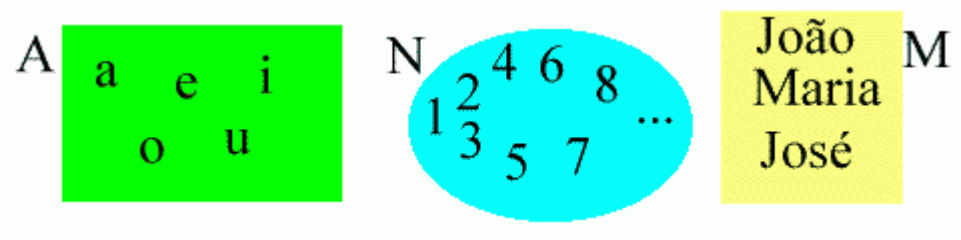
Notação

- Pertença (belongs): $1 \in \mathbb{N}$, $0 \notin \mathbb{N}$
- Diagrama de Venn-Euler: (lê-se: "Ven-óiler") Os conjuntos são mostrados graficamente.

$A = \{a, e, i, o, u\}$

$N = \{1, 2, 3, 4, \dots\}$

$M = \{\text{João}, \text{Maria}, \text{José}\}$

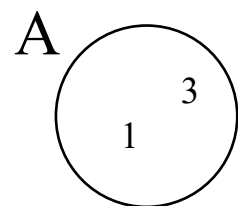


Set Theory (relational algebra)

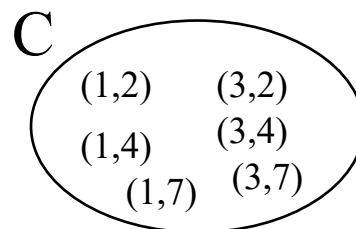
Set = collection of objects (members or elements).

Given **object O** (the object can be a set itself) and an **S set**, the object is a set element if the following child relationship occurs:

$$O \in S$$

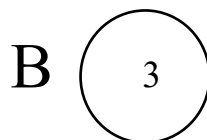
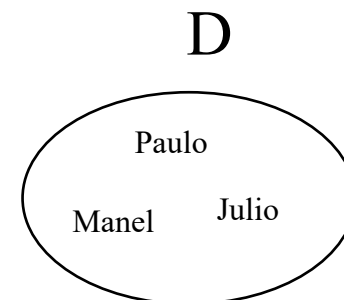


1 e 3 são elementos de A



(1,2) é elemento de C

1 e 2 são elementos de (1,2)



Given sets B and A, the inclusion relationship is $B \subseteq A$

Alguns conjuntos especiais

- **Conjunto vazio:**

É um conjunto que não possui elementos
 $\{ \}$ ou por \emptyset .

Exemplo com intersecção:

Se $A=\{a,e,i,o,u\}$ e $B=\{1,2,3,4\}$ então **$A \cap B = \emptyset$**

- **Conjunto universo:**

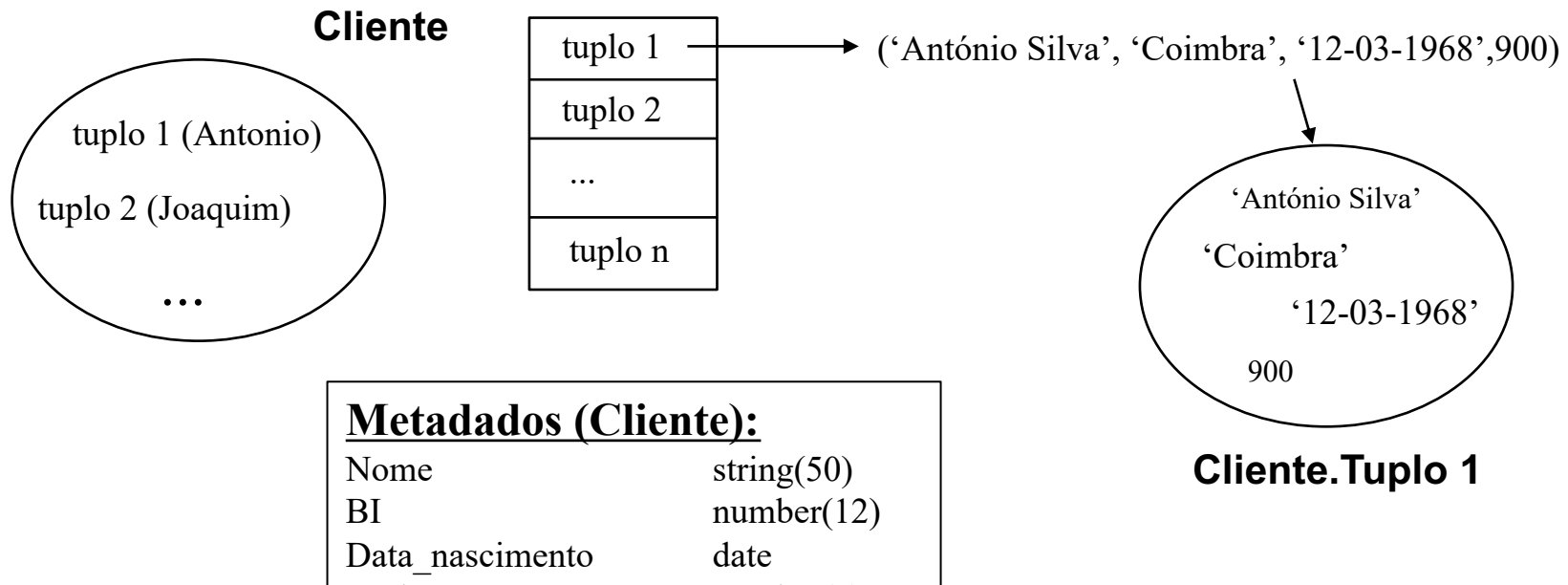
É um conjunto que contém todos os elementos = U

The relation=table is a set of tuples...

A tuple is a set of attributes

Cliente

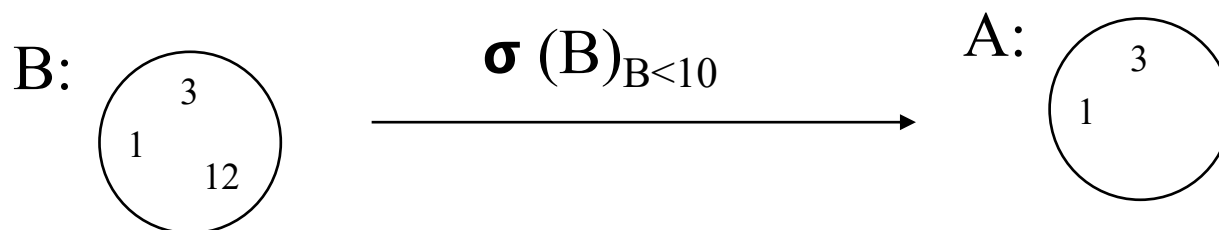
Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
...				



Selection

SELECTION= operation that selects sub-set of B based on a condition

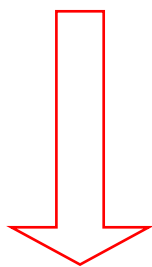
$$A = \sigma(B)_{B < 10}$$



σ = SELECTION

Operações mais básicas: **SELECÇÃO** σ

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
Maria Teresa Horta	Lisboa	1275432	15-09-1978	900
Joana Antunes	Lisboa	1022634	23-07-1988	801
Luísa Saraiva	Lisboa	1342664	22-04-1989	647



σ (Clientes)_{Nome like 'Jo%'}

Nome	Cidade	BI	Data_nascimento	N_de_conta
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
Joana Antunes	Lisboa	1022634	23-07-1988	801

UC

bd

Select * from clientes where nome like 'Jo%'

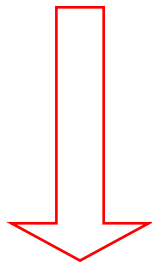
Projecção

$$\pi_{C.2} \left(C \begin{array}{cc} (1,2) & (3,5) \\ (1,4) & (3,4) \\ (1,7) & (3,7) \end{array} \right) = C \begin{array}{cc} (2) & (5) \\ (4) & (4) \\ (7) & (7) \end{array}$$

π = PROJECTION

Operações mais básicas: PROJEÇÃO π

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
Maria Teresa Horta	Lisboa	1275432	15-09-1978	900
Joana Antunes	Lisboa	1022634	23-07-1988	801
Luísa Saraiva	Lisboa	1342664	22-04-1989	647



π (Clientes)_{Nome, BI}

Nome	BI
António Silva	1234343
Joaquim Alves Dias	1256673
Maria Teresa Horta	1275432
Joana Antunes	1022634
Luísa Saraiva	1342664

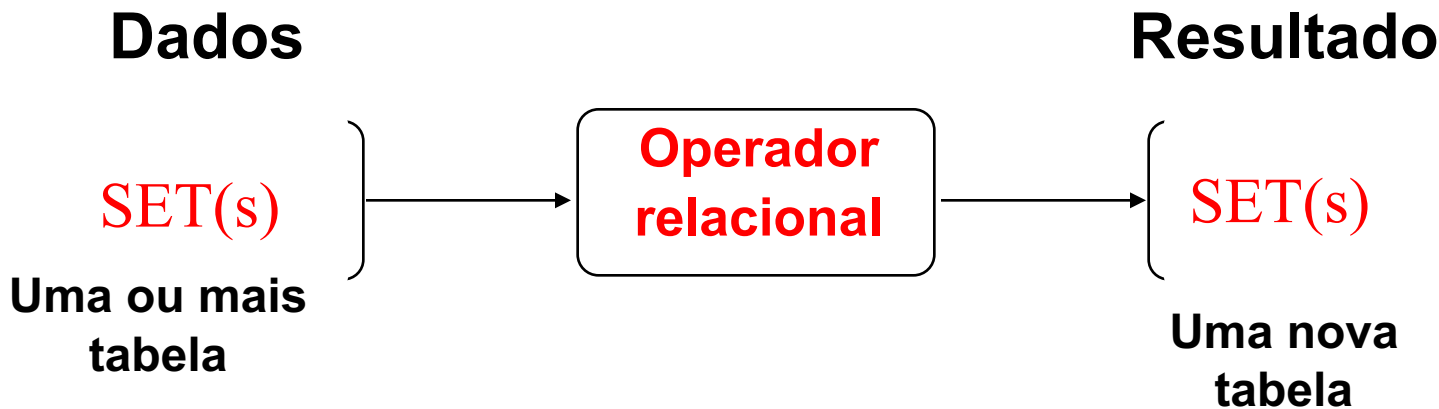
Select Nome, BI
from clientes

How SLQ relates to maths...

Select emp.nome, sal,funcao,dep.nome
From emp join dep on (emp.ndep=dep.ndep)
Where emp.sal>200 and dep.ndep=10;

$\pi_{\text{emp.nome, sal, funcao, dep.nome}} [\sigma_{\text{emp.sal} > 200} (\sigma_{\text{dep.ndep} = 10} (\text{emp} \bowtie_{\text{ndep} = \text{ndep}} \text{dep}))]$

Porque o SQL é tão útil?



- Note que a “entrada” são tabelas, mas a “saída” também são tabelas
- Então eu posso encadear operações!!!!
- Posso fazer até dezenas ou centenas encadeadas

How SLQ relates to maths...

Select emp.nome, sal,funcao
From emp

Where emp.sal>200 and funcao='Vendedor';

$$\pi_{\text{emp.nome, sal, funcao}} [\sigma_{\text{emp.sal} > 200} (\sigma_{\text{emp.funcao} = \text{'Vendedor'}} (\text{ emp}))]$$

PAUSE...

Set operations

Union, Intersect, difference

Set Theory (relational algebra)

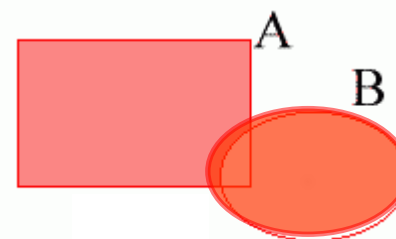
- Set = collection of objects (members or elements).
- All mathematical objects can be put into sets.

Operações sobre conjuntos

- **Reunião de conjuntos (UNION)**

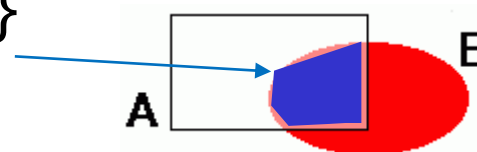
A reunião dos conjuntos A e B é o conjunto de todos os elementos que pertencem ao conjunto A ou ao conjunto B.

$$A \cup B = \{ x: x \in A \vee x \in B \}$$



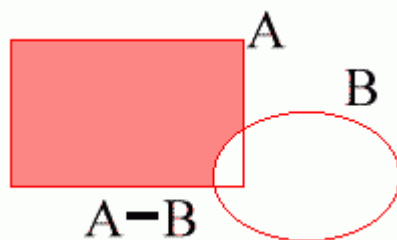
- **Interseção de conjuntos (intersection)**

$$A \cap B = \{ x: x \in A \wedge x \in B \}$$



- **Diferença de conjuntos (difference)**

A diferença entre os conjuntos A e B é o conjunto de todos os elementos que pertencem ao conjunto A e *não* pertencem ao conjunto B.



Operations on sets

Set = collection of objects (members or elements).

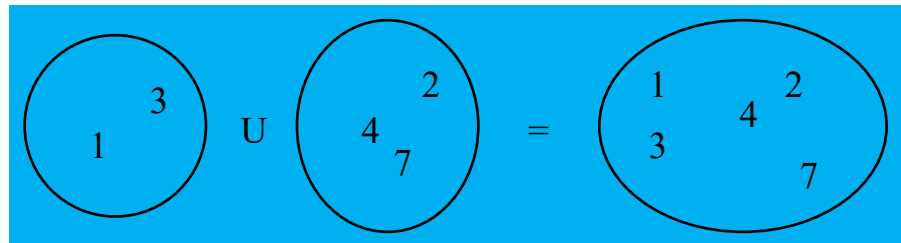
Binary operations on sets:

Intersection, Union,

Difference of sets, Cartesian product.

UC

UNION:




bd

União


A união de duas tabelas resulta numa nova tabela que contém todos os registos existentes nas duas tabelas.

```
SELECT nome, funcao, sal
FROM emp
WHERE ndep = 30 AND sal > 220000;
```



NOME	FUNCAO	SAL
-----	-----	-----
Duarte Guedes	Encarregado	380850
Ana Rodrigues	Vendedor	221250

```
SELECT nome, funcao, sal
FROM emp
WHERE funcao = 'Vendedor';
```




NOME	FUNCAO	SAL
-----	-----	-----
Joana Mendes	Vendedor	145600
Nelson Neves	Vendedor	212250
Ana Rodrigues	Vendedor	221250
Manuel Madeira	Vendedor	157800

```
SELECT nome, funcao, sal
FROM emp
WHERE ndep = 30 AND sal > 220000
```

UNION

```
SELECT nome, funcao, sal
FROM emp
WHERE funcao = 'Vendedor';
```



NOME	FUNCAO	SAL
-----	-----	-----
Ana Rodrigues	Vendedor	221250
Duarte Guedes	Encarregado	380850
Joana Mendes	Vendedor	145600
Manuel Madeira	Vendedor	157800
Nelson Neves	Vendedor	212250

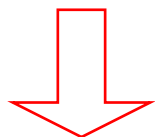
Operações mais básicas: **UNIÃO U**

Clientes

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
Maria Teresa Horta	Lisboa	1275432	15-09-1978	900

Clientes1

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556



Clientes **U** Clientes1

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
Maria Teresa Horta	Lisboa	1275432	15-09-1978	900

UC

bd

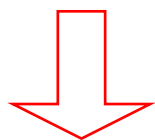
Operações mais básicas: INTERSECÇÃO \cap

Clientes

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
Maria Teresa Horta	Lisboa	1275432	15-09-1978	900
Joana Antunes	Lisboa	1022634	23-07-1988	801
Luísa Saraiva	Lisboa	1342664	22-04-1989	647

Clientes1

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556



Clientes \cap Clientes1

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556

UC

bd

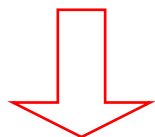
Operações básicas: **Diferença** \

Clientes

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556
Maria Teresa Horta	Lisboa	1275432	15-09-1978	900

Clientes1

Nome	Cidade	BI	Data_nascimento	N_de_conta
António Silva	Coimbra	1234343	12-03-1968	900
Joaquim Alves Dias	Guarda	1256673	18-01-1978	556



Clientes \ Clientes1

Nome	Cidade	BI	Data_nascimento	N_de_conta
Maria Teresa Horta	Lisboa	1275432	15-09-1978	900

Joins

Operações relacionais (breve introdução)



Operadores relacionais

Operadores unários

- **Restrição=Seleção**
- **Projecção**

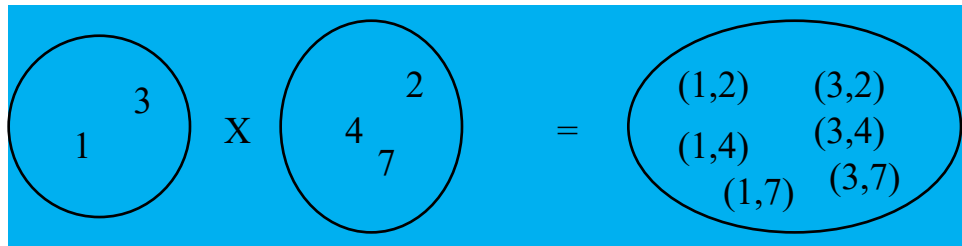
Operadores binários

- **União**
- **Intersecção**
- **Diferença**
- **Divisão**
- **Prod. cartesiano**
- **Junção**

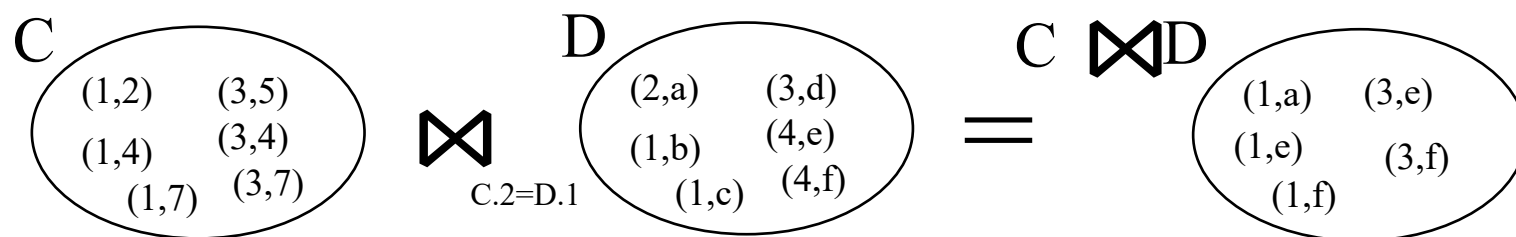
Operations on sets

The Cartesian product is a set whose members are all possible ordered pairs of members of the two sets participating in the operation.

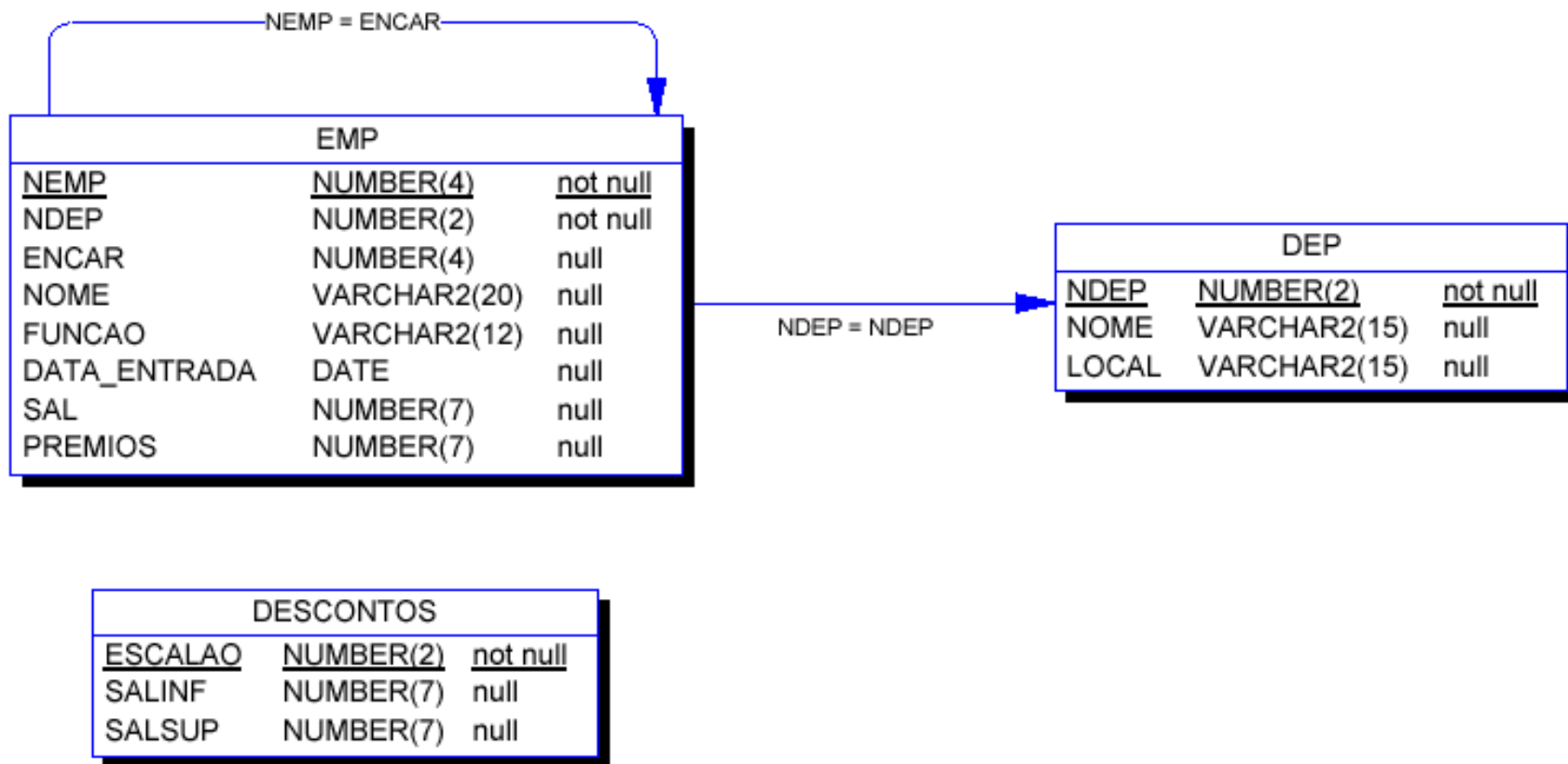
PRODUCT:



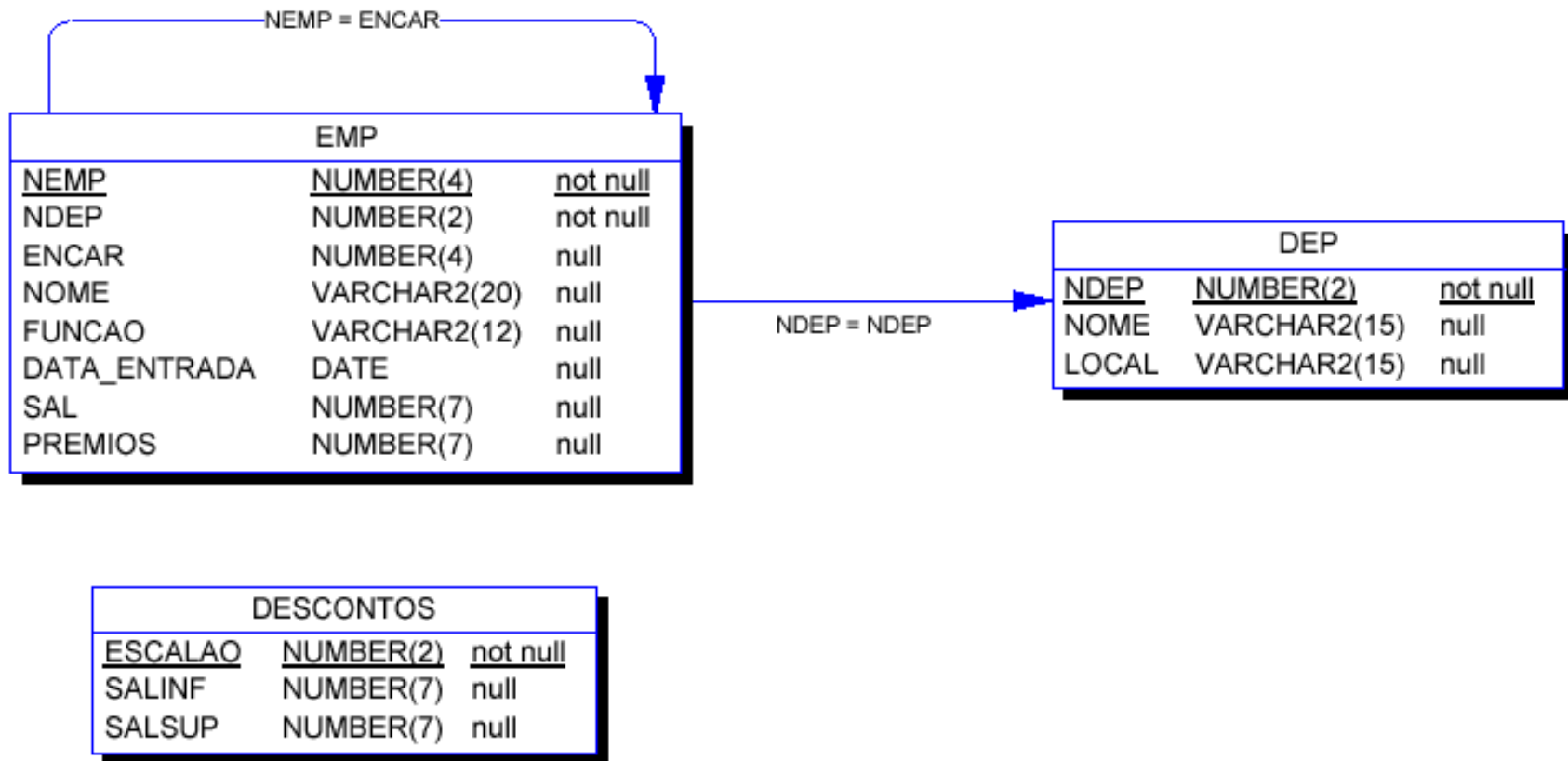
Join



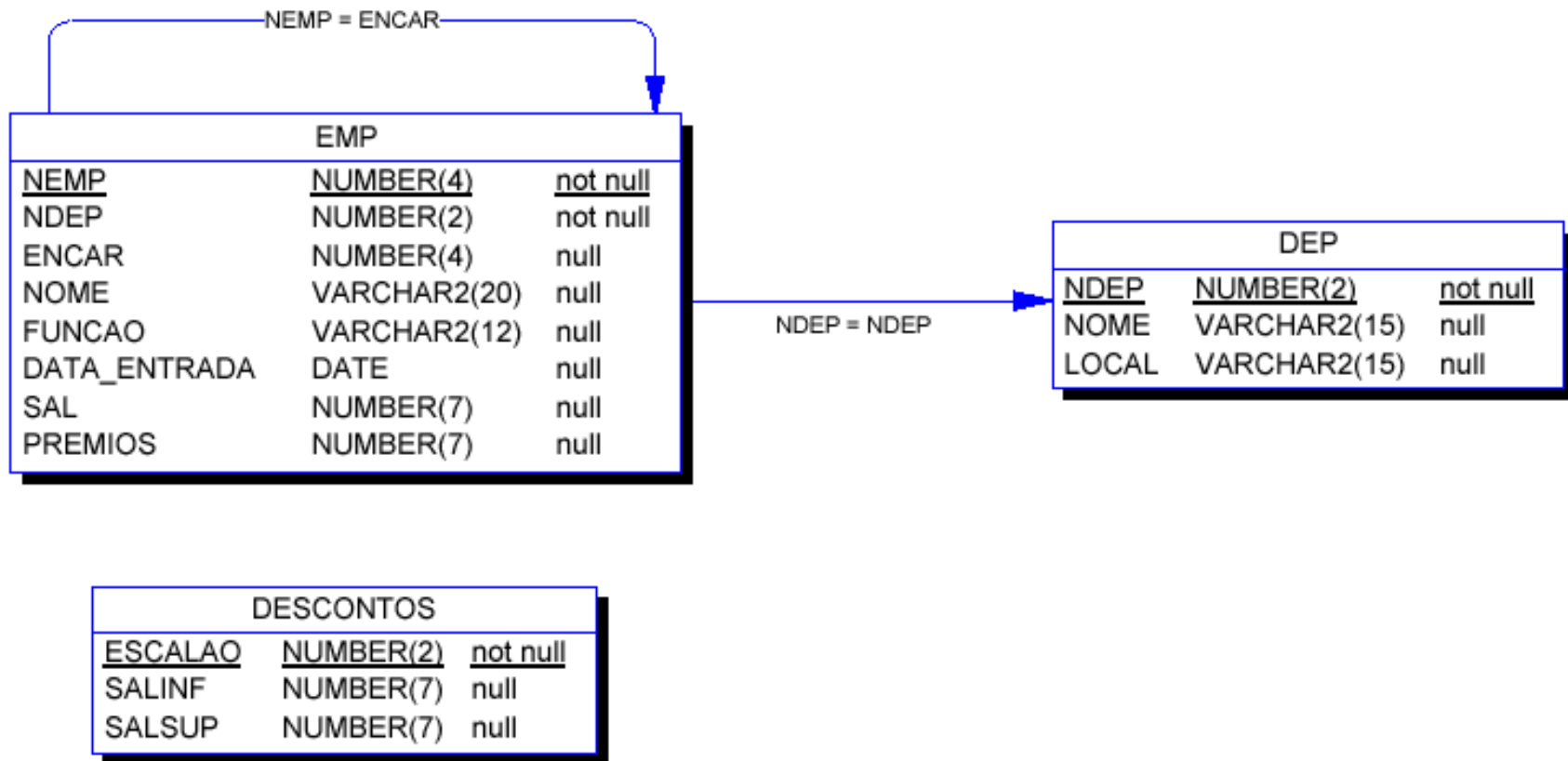
Como se relacionam as duas tabelas?



Como vejo o departamento de um empregado?



Como vejo o empregado e o departamento num só resultado?



Junção interna (*inner join*)

A junção interna (*inner join*) é tipo de junção por defeito. Apenas os registos que satisfazem a condição de junção aparecem no resultado.

```
SELECT emp.nome, emp.ndep, dep.ndep, dep.nome
FROM emp, dep
WHERE emp.ndep = dep.ndep;
```

Neste exemplo pode ver-se que só os registos de ambas as tabelas que têm o mesmo valor de **ndep** é que aparecem no resultado.

NOME	NDEP	NDEP	NOME
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
Duarte Guedes	30	30	Vendas
Joana Mendes	30	30	Vendas
Nelson Neves	30	30	Vendas
Ana Rodrigues	30	30	Vendas
Manuel Madeira	30	30	Vendas
Tome Ribeiro	30	30	Vendas
14 rows selected.			

How SLQ relates to maths...

Select emp.nome, sal,funcao,dep.nome
From emp join dep on (emp.ndep=dep.ndep)
Where emp.sal>200 and dep.ndep=10;

$$\pi_{\text{emp.nome, sal, funcao, dep.nome}} [\sigma_{\text{emp.sal} > 200} (\sigma_{\text{dep.ndep} = 10} (\text{emp} \bowtie_{\text{ndep} = \text{ndep}} \text{dep}))]$$

Junção (equi-junção) Agora sem mostrar ndep:

Exemplo:

```
SELECT emp.nome, funcao, dep.nome
FROM emp, dep
WHERE emp.ndep = dep.ndep;
```

Condição de junção. Só os registos que satisfazem esta condição de junção é que aparecem no resultado.

Os dados destes dois atributos vêm da tabela **emp**.

Os dados deste atributo vêm da tabela **dep**.

NOME	FUNCAO	NOME
-----	-----	-----
Jorge Sampaio	Presidente	Contabilidade
Silvia Teles	Encarregado	Contabilidade
Olga Costa	Continuo	Contabilidade
Augusto Reis	Encarregado	Investigação
Rita Pereira	Continuo	Investigação

O que faz a Junção interna

Vai tentar juntar cada linha de uma tabela com cada linha da outra

```
SELECT emp.nome, emp.ndep, dep.ndep, dep.nome  
FROM emp, dep  
WHERE emp.ndep = dep.ndep;
```

Mas depois, baseado na condição de junção, fica apenas com as linhas que estão em ambas as tabelas

E se eu selecionar tudo?

```
select *
from emp,dep
where emp.ndep=dep.ndep;
```

nemp	nome	funcao	encar	data_entrada	sal	premios	ndep	ndep	nome	local
1839	Jorge Sampaio	Presidente		1984-02-11	890000		10	10	Contabilidade	Condeixa
1566	Augusto Reis	Encarregado	1839	1985-02-13	450975		20	20	InvestigaÁ„o	Mealhada
1698	Duarte Guedes	Encarregado	1839	1991-11-25	380850		30	30	Vendas	Coimbra
1782	Silvia Teles	Encarregado	1839	1986-11-03	279450		10	10	Contabilidade	Condeixa
1788	Maria Dias	Analista	1566	1982-11-07	565000		20	20	InvestigaÁ„o	Mealhada
1902	Catarina Silva	Analista	1566	1993-04-13	435000		20	20	InvestigaÁ„o	Mealhada
1499	Joana Mendes	Vendedor	1698	1984-10-04	145600	56300	30	30	Vendas	Coimbra
1521	Nelson Neves	Vendedor	1698	1983-02-27	212250	98500	30	30	Vendas	Coimbra
1654	Ana Rodrigues	Vendedor	1698	1990-12-17	221250	81400	30	30	Vendas	Coimbra
1844	Manuel Madeira	Vendedor	1698	1985-04-21	157800	0	30	30	Vendas	Coimbra
1900	Tome Ribeiro	Continuo	1698	1994-03-05	56950		30	30	Vendas	Coimbra
1876	Rita Pereira	Continuo	1788	1996-02-07	65100		20	20	InvestigaÁ„o	Mealhada
1934	Olga Costa	Continuo	1782	1986-06-22	68300		10	10	Contabilidade	Condeixa
1369	Antonio Silva	Continuo	1902	1996-12-22	70800		20	20	InvestigaÁ„o	Mealhada

(14 rows)

Porque tenho de por a condição?
E se não pusesse?

```
select * from emp,dep  
where emp.ndep=dep.ndep;
```

Se não pusesse a condição: Prod cartesiano:
 select * from emp,dep;

nemp	nome	funcao	encar	data_entrada	sal	premios	ndep	ndep	nome	local
1839	Jorge Sampaio	Presidente		1984-02-11	890000		10	10	Contabilidade	Condeixa
1566	Augusto Reis	Encarregado	1839	1985-02-13	450975		20	10	Contabilidade	Condeixa
1698	Duarte Guedes	Encarregado	1839	1991-11-25	380850		30	10	Contabilidade	Condeixa
1782	Silvia Teles	Encarregado	1839	1986-11-03	279450		10	10	Contabilidade	Condeixa
1788	Maria Dias	Analista	1566	1982-11-07	565000		20	10	Contabilidade	Condeixa
1902	Catarina Silva	Analista	1566	1993-04-13	435000		20	10	Contabilidade	Condeixa
1499	Joana Mendes	Vendedor	1698	1984-10-04	145600	56300	30	10	Contabilidade	Condeixa
1521	Nelson Neves	Vendedor	1698	1983-02-27	212250	98500	30	10	Contabilidade	Condeixa
1654	Ana Rodrigues	Vendedor	1698	1990-12-17	221250	81400	30	10	Contabilidade	Condeixa
1844	Manuel Madeira	Vendedor	1698	1985-04-21	157800	0	30	10	Contabilidade	Condeixa
1900	Tome Ribeiro	Continuo	1698	1994-03-05	56950		30	10	Contabilidade	Condeixa
1876	Rita Pereira	Continuo	1788	1996-02-07	65100		20	10	Contabilidade	Condeixa
1934	Olga Costa	Continuo	1782	1986-06-22	68300		10	10	Contabilidade	Condeixa
1369	Antonio Silva	Continuo	1902	1996-12-22	70800		20	10	Contabilidade	Condeixa
1839	Jorge Sampaio	Presidente		1984-02-11	890000		10	20	Investigaç„o	Mealhada
1566	Augusto Reis	Encarregado	1839	1985-02-13	450975		20	20	Investigaç„o	Mealhada
1698	Duarte Guedes	Encarregado	1839	1991-11-25	380850		30	20	Investigaç„o	Mealhada
1782	Silvia Teles	Encarregado	1839	1986-11-03	279450		10	20	Investigaç„o	Mealhada
1788	Maria Dias	Analista	1566	1982-11-07	565000		20	20	Investigaç„o	Mealhada
1902	Catarina Silva	Analista	1566	1993-04-13	435000		20	20	Investigaç„o	Mealhada
1499	Joana Mendes	Vendedor	1698	1984-10-04	145600	56300	30	20	Investigaç„o	Mealhada

56 rows = 14 emps x 4 deps

A junção e o produto cartesiano: como não fazer

junção:

```
SELECT emp.NOME, emp.NDEP,  
       dep.NDEP, dep.NOME  
FROM emp, dep  
Where emp.NDEP = dep.NDEP;
```



produto cartesiano:

```
SELECT emp.NOME, emp.NDEP,  
       dep.NDEP, dep.NOME  
FROM emp, dep
```



selecção:

```
Where emp.NDEP = dep.NDEP;
```

todos os pares de
registos de ambas

restringe a alguns

Problema:

E se forem
10M emp e
10 M dep?

como
Resolver
isso?

Explain de prod cartesiano

```
explain select e.*, d.*  
from emp as e, dep as d
```

```
explain select *  
from emp, dep
```

```
"Nested Loop (cost=0.00..6796.90 rows=540800 width=178)"
```

```
"  -> Seq Scan on dep (cost=0.00..20.40 rows=1040 width=50)"
```

```
"  -> Materialize (cost=0.00..17.80 rows=520 width=128)"
```

```
"    -> Seq Scan on emp (cost=0.00..15.20 rows=520 width=128)"
```

O problema do prod cartesiano

- Qual o problema potencial, se as tabelas forem grandes (e.g. imagine EMP e DEP ambas com 10 M de registos)?=

produto cartesiano:

```
SELECT emp.NOME, emp.NDEP,  
       dep.NDEP, dep.NOME  
FROM emp, dep
```

todos os pares de
registos de ambas



selecção:

```
Where emp.NDEP = dep.NDEP;
```

restringe a alguns

Juncao e prod cartesiano

```
Select e.*,d.*  
From  
    (select e.*,d.*  
     from emp e, dep d)
```

Where e.ndep=d.ndep

PROBLEMA: 10M x 10M = 10 MM!!!

Como faz o sistema para evitar o problema?

Uma hipótese: ordena cada uma das tabelas pelo attr de junção

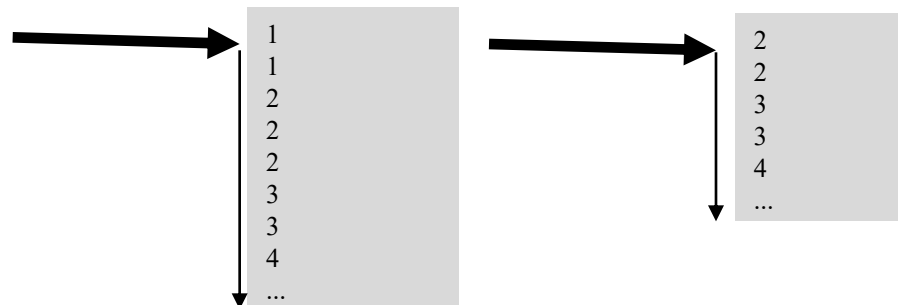
Agora percorre so uma vez ambas 10M+10M

MAS CLARO QUE ORDENAR TB É caro....

Evita prod cartesiano:

```
SELECT emp.NOME, emp.NDEP,  
       dep.NDEP, dep.NOME  
FROM emp, dep  
Where emp.NDEP = dep.NDEP;
```

1. Identify that this is a join
2. Sort table 1 and table 2 based on the where attrs
3. Ordered matching of rows:



Explain de junção (sort join)

```
explain select e.*, d.*  
from emp as e, dep as d  
where d.ndep=e.ndep
```

```
explain select /*+ USE_MERGE */ *  
from emp, dep
```

```
"Merge Join (cost=23.50..42.29 rows=370 width=296)"  
"  
  Merge Cond: (e.ndep = d.ndep)"  
"  
  -> Sort (cost=16.00..16.00 rows=600 width=108)"  
"  
    -> Seq Scan on emp e (cost=0.00..13.70 rows=370 width=188)"  
"  
      -> Sort (cost=16.00..16.00 rows=600 width=108)"  
"  
        -> Seq Scan on dep d (cost=0.00..16.00 rows=600 width=108)"  
"
```

UC

bdai

Explain de junção

(hash é outro algoritmo como o sort join).

```
explain select e.*, d.*  
from emp as e, dep as d  
where d.ndep=e.ndep
```

```
"Hash Join (cost=23.50..42.29 rows=370 width=296)"
```

```
"  Hash Cond: (e.ndep = d.ndep)"
```

```
"  -> Seq Scan on emp e (cost=0.00..13.70 rows=370 width=188)"
```

```
"  -> Hash (cost=16.00..16.00 rows=600 width=108)"
```

```
"    -> Seq Scan on dep d (cost=0.00..16.00 rows=600 width=108)"
```

Mas o join nested loop tb pode ser eficiente...

Se uma das tabelas for pequena

```
explain select e.*, d.*  
from emp as e, dep as d  
where emp.ndep=dep.ndep
```

```
explain select *  
from emp, dep
```

"**Nested Loop** (cost=0.00..6796.90 rows=540800 width=178)"

" Filter Cond: (e.ndep = d.ndep)"

" -> Seq Scan on dep (cost=0.00..20.40 rows=1040 width=50)"

" -> Materialize (cost=0.00..17.80 rows=520 width=128)"

" -> Seq Scan on emp (cost=0.00..15.20 rows=520 width=128)"

Conclusion

- (at least 3) Different ways for the DBMS to execute a JOIN
- It is fully optimized to be eficiente even with huge tables
- That way it only has difficulties when both tables are extremely huge (e.g. 10GB x 2GB)
- In those cases it takes a lot of time and uses disk (external memory=paging), which is slower, but DOES NOT CRASH...

Distinguir colunas com o mesmo nome em tabelas diferentes

```
SELECT emp.NOME, FUNCAO, dep.NOME  
FROM emp, dep  
WHERE dep.NDEP = emp.ndep;
```

Como o atributo nome aparece nas duas tabelas, para os distinguir é necessário inserir o nome da tabela antes.

```
SELECT e.NOME, e.FUNCAO, d.NOME  
FROM emp AS e, dep AS d  
WHERE d.NDEP = e.ndep;
```

Para facilitar, podemos dar um **pseudónimo ou cognome** ao nome da tabela de modo a ser mais fácil escrever o comando. Por exemplo, a tabela **emp** passa a ser designada por **e** e **dep** por **d**.

Junção, projecção e restrição

Vulgarmente a junção usa-se em conjunto com a projecção e a restrição, de modo a eliminar colunas e registos desnecessários para o resultado.

```
SELECT emp.nome, funcao, dep.nome
FROM emp, dep
WHERE emp.ndep = dep.ndep
And sal > 220000;
```

Junção: mostra dados das duas tabelas, emp e dep

Projecção: só mostra alguns atributos.

Restrição: só mostra alguns registos.

NOME	FUNCAO	NOME
-----	-----	-----
Jorge Sampaio	Presidente	Contabilidade
Silvia Teles	Encarregado	Contabilidade
Augusto Reis	Encarregado	Investigação
Catarina Silva	Analista	Investigação
Maria Dias	Analista	Investigação
Duarte Guedes	Encarregado	Vendas
Ana Rodrigues	Vendedor	Vendas
7 rows selected.		


```
explain SELECT emp.nome, funcao,  
dep.nome  
FROM emp, dep  
WHERE emp.ndep = dep.ndep  
And sal > 220000;
```

```
"Hash Join (cost=33.40..50.35 rows=173 width=59)"  
" Hash Cond: (emp.ndep = dep.ndep)"  
" -> Seq Scan on emp (cost=0.00..16.50 rows=173 width=52)"  
"   Filter: (sal > '220000'::numeric)"  
" -> Hash (cost=20.40..20.40 rows=1040 width=31)"  
"   -> Seq Scan on dep (cost=0.00..20.40 rows=1040 width=31)"
```

Junção: outra sintaxe (SQL standard)

Outra sintaxe mais explícita quanto à operação de junção:

```
SELECT emp.nome, emp.ndep, dep.ndep, dep.nome
FROM emp JOIN dep on emp.ndep = dep.ndep
WHERE dep.nome = 'Contabilidade' or dep.nome = 'Investigação' ;
```

NOME	NDEP	NDEP	NOME
-----	-----	-----	-----
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
8 rows selected.			

UC

bd

Junção: outra sintaxe (SQL standard)

Outra sintaxe mais explícita quanto à operação de junção:

```
SELECT emp.nome, emp.ndep, dep.ndep, dep.nome  
FROM emp JOIN dep on emp.ndep = dep.ndep  
WHERE dep.nome = 'Contabilidade' or dep.nome = 'Investigação' ;
```

UC

```
SELECT emp.nome, funcao, dep.nome  
FROM emp, dep  
WHERE emp.ndep = dep.ndep
```

```
And dep.nome = 'Contabilidade' or dep.nome =  
'Investigação' ;
```

bd

Junção externa

```
SELECT emp.nome, emp.ndep, dep.ndep, dep.nome
FROM emp, dep
WHERE emp.ndep = dep.ndep;
```

Como fazer
para aparecer isto?

NOME	NDEP	NDEP	NOME
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
Duarte Guedes	30	30	Vendas
Joana Mendes	30	30	Vendas
Nelson Neves	30	30	Vendas
Ana Rodrigues	30	30	Vendas
Manuel Madeira	30	30	Vendas
Tome Ribeiro	30	30	Vendas

14 rows selected.

O departamento 40 não tem
empregados mas aparece
também no resultado da
junção externa.

NOME	NDEP	NDEP	NOME
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
Duarte Guedes	30	30	Vendas
Joana Mendes	30	30	Vendas
Nelson Neves	30	30	Vendas
Ana Rodrigues	30	30	Vendas
Manuel Madeira	30	30	Vendas
Tome Ribeiro	30	30	Vendas
		40	Planeamento

15 rows selected.

NOME	NDEP	NDEP	NOME
-----	-----	-----	-----
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
Duarte Guedes	30	30	Vendas
Joana Mendes	30	30	Vendas
Nelson Neves	30	30	Vendas
Ana Rodrigues	30	30	Vendas
Manuel Madeira	30	30	Vendas
Tome Ribeiro	30	30	Vendas
		40	Planeamento
15 rows selected.			

É um departamento sem
empregados!

Exemplo de junção externa (à direita)

ORACLE:

```
SELECT emp.NOME, emp.NDEP,
       dep.NDEP, dep.NOME
FROM emp, dep
Where emp.NDEP (+) = dep.NDEP;
```

QQ SGBD, inclui Oracle:

```
SELECT emp.NOME, emp.NDEP,
       dep.NDEP, dep.NOME
FROM emp RIGHT JOIN dep
ON emp.ndep=dep.ndep;
```

O departamento 40 não tem empregados mas aparece também no resultado da junção externa.

NOME	NDEP	NDEP	NOME
-----	-----	-----	-----
Jorge Sampaio	10	10	Contabilidade
Silvia Teles	10	10	Contabilidade
Olga Costa	10	10	Contabilidade
Augusto Reis	20	20	Investigação
Rita Pereira	20	20	Investigação
Catarina Silva	20	20	Investigação
Maria Dias	20	20	Investigação
Antonio Silva	20	20	Investigação
Duarte Guedes	30	30	Vendas
Joana Mendes	30	30	Vendas
Nelson Neves	30	30	Vendas
Ana Rodrigues	30	30	Vendas
Manuel Madeira	30	30	Vendas
Tome Ribeiro	30	30	Vendas
		40	Planeamento

15 rows selected.

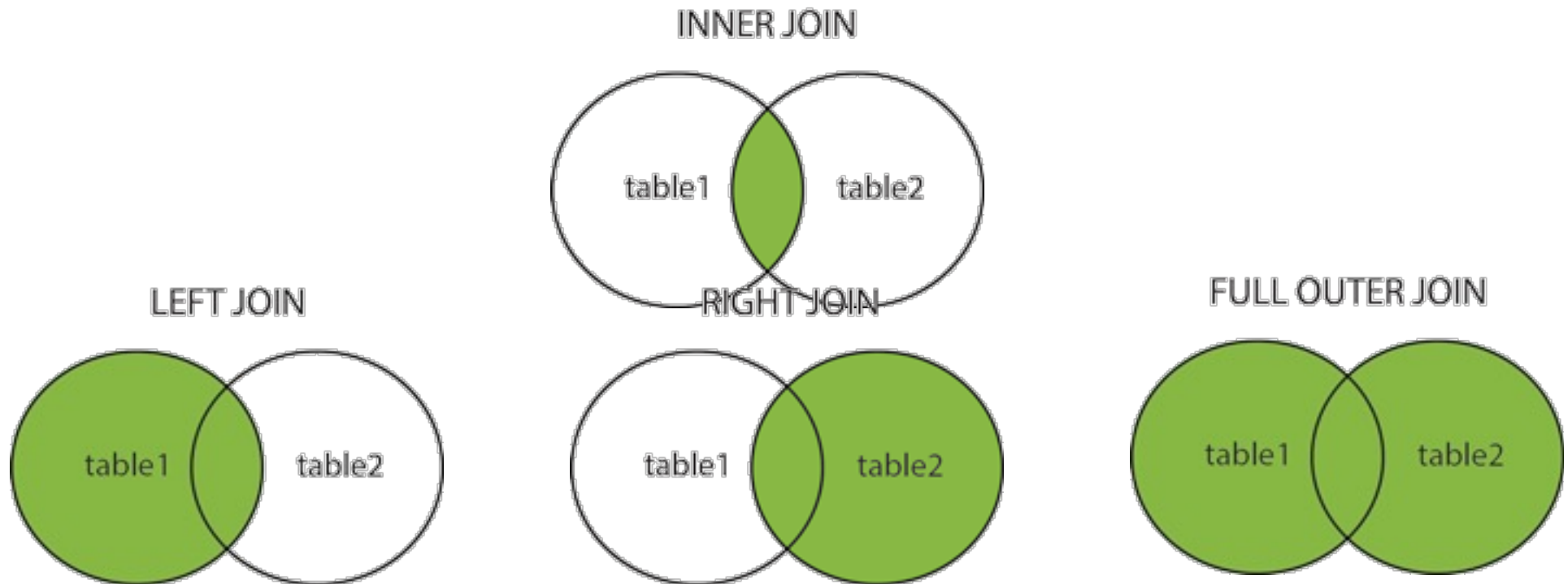
Junção interna e externa - SINTAXE

- Keywords:

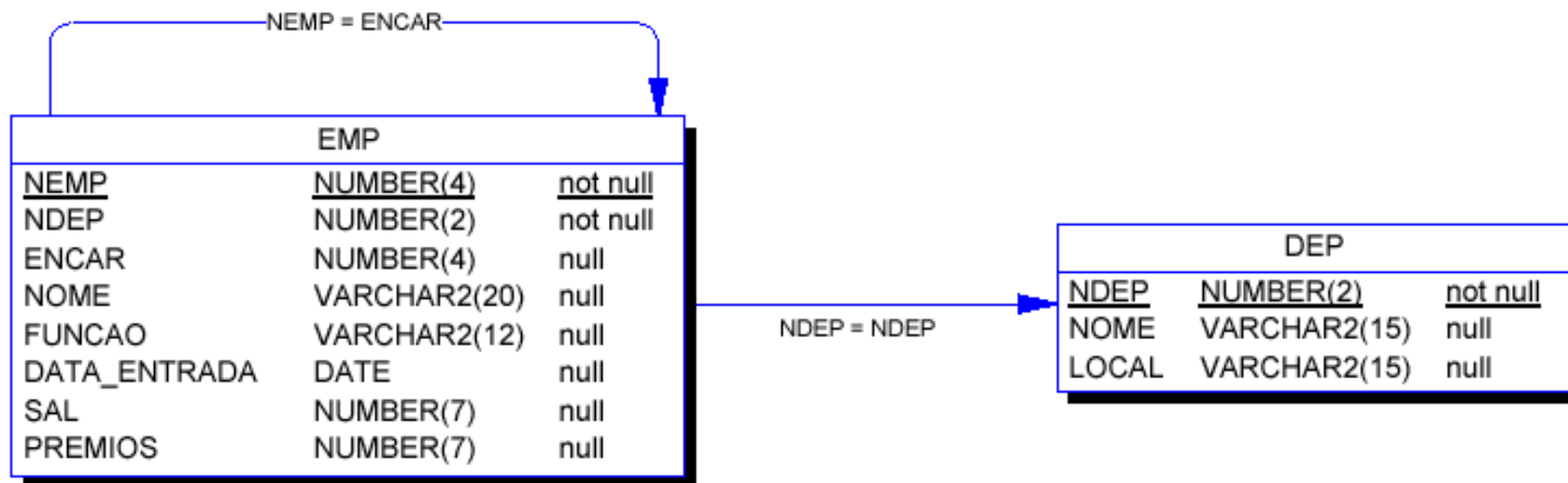
FULL INNER JOIN = JOIN = INNER JOIN

FULL OUTER JOIN = OUTER JOIN

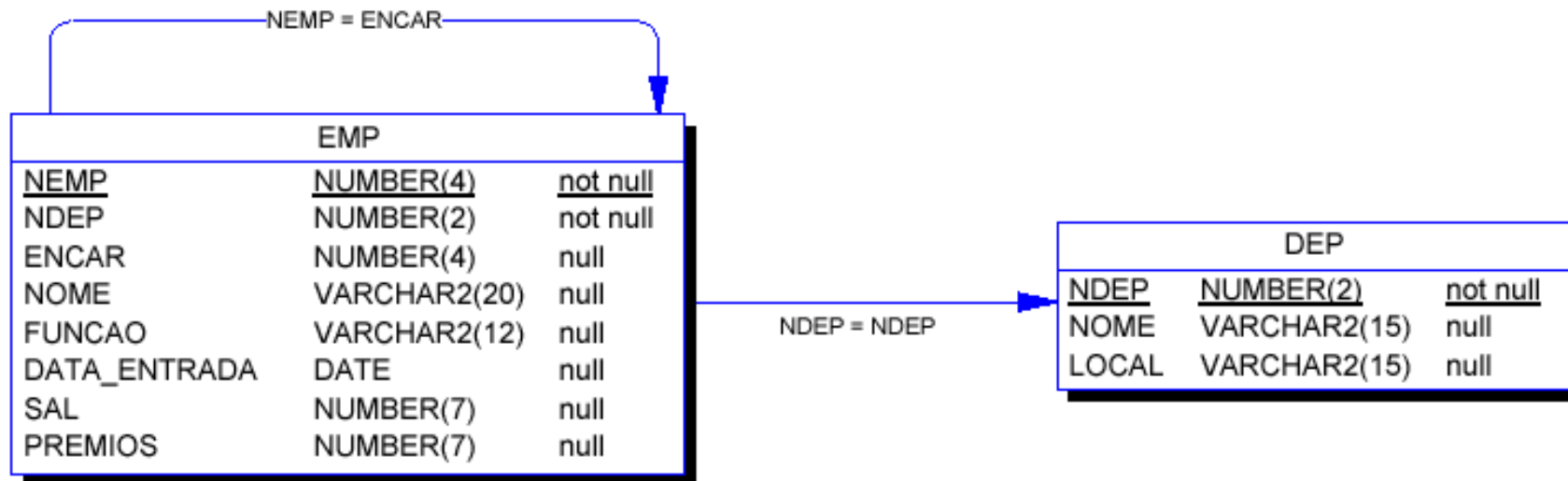
RIGHT JOIN, LEFT JOIN



Nao equi-junção



NOME	SAL	ESCALAO
-----	-----	-----
Tome Ribeiro	56950	1
Rita Pereira	65100	1
Olga Costa	68300	1
Antonio Silva	70800	1
Joana Mendes	145600	2
Manuel Madeira	157800	2
Nelson Neves	212250	3
Ana Rodrigues	221250	3
Silvia Teles	279450	3
Duarte Guedes	380850	4
Catarina Silva	435000	4
Augusto Reis	450975	4
Maria Dias	565000	5
Jorge Sampaio	890000	5
14 rows selected.		



If sal in [20000, 30000] escalão = 1

Else if sal in [30000, 40000] escalão = 2

...

descontos

Escalao	salinf	salsup
"1"	"55000"	"99999"
"2"	"100000"	"210000"
"3"	"210001"	"350000"
"4"	"350001"	"550000"
"5"	"550001"	"9999999"

If sal in [20000, 30000] escalão = 1

Else if sal in [20000, 30000] escalão = 2

...

Não equi-junção

```
SELECT NOME, SAL, ESCALAO  
FROM emp, descontos  
WHERE sal between salinf and salsup;
```

NOME	SAL	ESCALAO
Tome Ribeiro	56950	1
Rita Pereira	65100	1
Olga Costa	68300	1
Antonio Silva	70800	1
Joana Mendes	145600	2
Manuel Madeira	157800	2
Nelson Neves	212250	3
Ana Rodrigues	221250	3
Silvia Teles	279450	3
Duarte Guedes	380850	4
Catarina Silva	435000	4
Augusto Reis	450975	4
Maria Dias	565000	5
Jorge Sampaio	890000	5

14 rows selected.

Neste caso a condição de junção não é uma igualdade (como quando é feita através das chaves primária + estrangeira) mas sim um intervalo.

Aggregations

Agregações

- Estatísticas
 - exemplo: qual o salário médio?
 - exemplo: qual o percentil 75 do salário?
 - qual o percentil 75 do salário de cada função na empresa?
- Principais funções de grupo incluem:
 - AVG, COUNT, MIN, MAX e SUM
 - STDDEV
 - PERCENTILE_cont
 - ...

Agregações

```
SELECT AVG(sal)
FROM emp;
```

```
      avg
-----
285666.071428571429
(1 row)
```

```
SELECT funcao, AVG(sal) , stddev(sal), max(sal), min(sal)
FROM emp
GROUP BY funcao
ORDER BY funcao ASC;
```

funcao	avg	stddev	max	min
Analista	500000.000000000000	91923.88155425	565000	435000
Continuo	65287.500000000000	6028.042108899594	70800	56950
Encarregado	370425.000000000000	86236.40110185	450975	279450
Presidente	890000.000000000000		890000	890000
Vendedor	184225.000000000000	38063.20401648	221250	145600

(5 rows)

Agregações

```
SELECT funcao, round(AVG(sal),1) ,  
        round(stddev(sal),1), max(sal), min(sal)  
FROM emp  
GROUP BY funcao  
ORDER BY funcao ASC;
```

funcao	round	round	max	min
Analista	500000.0	91923.9	565000	435000
Continuo	65287.5	6028.0	70800	56950
Encarregado	370425.0	86236.4	450975	279450
Presidente	890000.0		890000	890000
Vendedor	184225.0	38063.2	221250	145600

(5 rows)

COUNT

- O que faz o seguinte?

```
SELECT funcao, count(*)  
FROM emp  
GROUP BY funcao  
ORDER BY funcao ASC;
```

- E o que muda o seguinte?

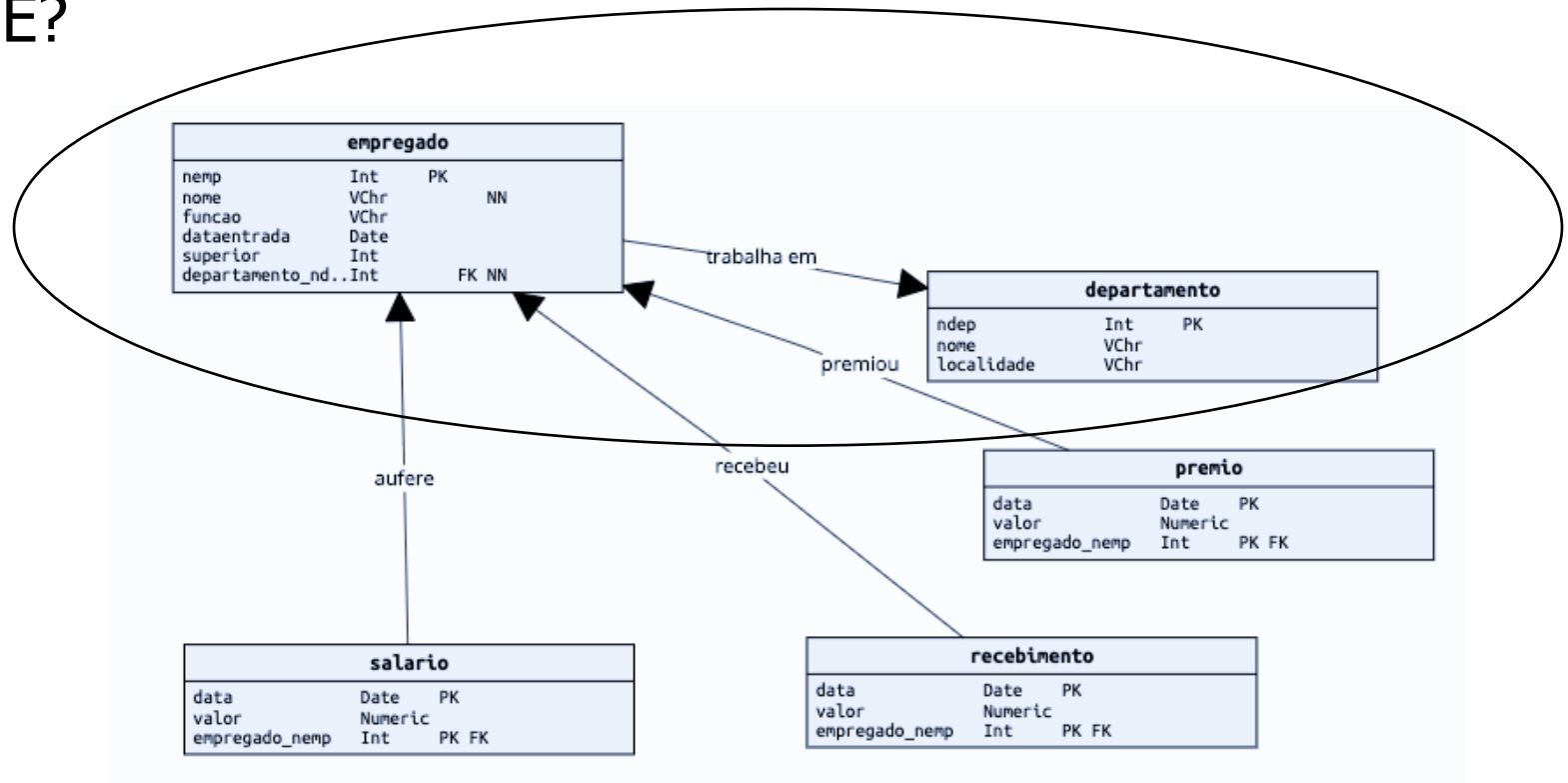
```
SELECT funcao, count(premios)  
FROM emp  
GROUP BY funcao  
ORDER BY funcao ASC;
```

Representação e análise ao longo do tempo?

Como armazenar e analisar o salário de empregados ao longo do tempo?

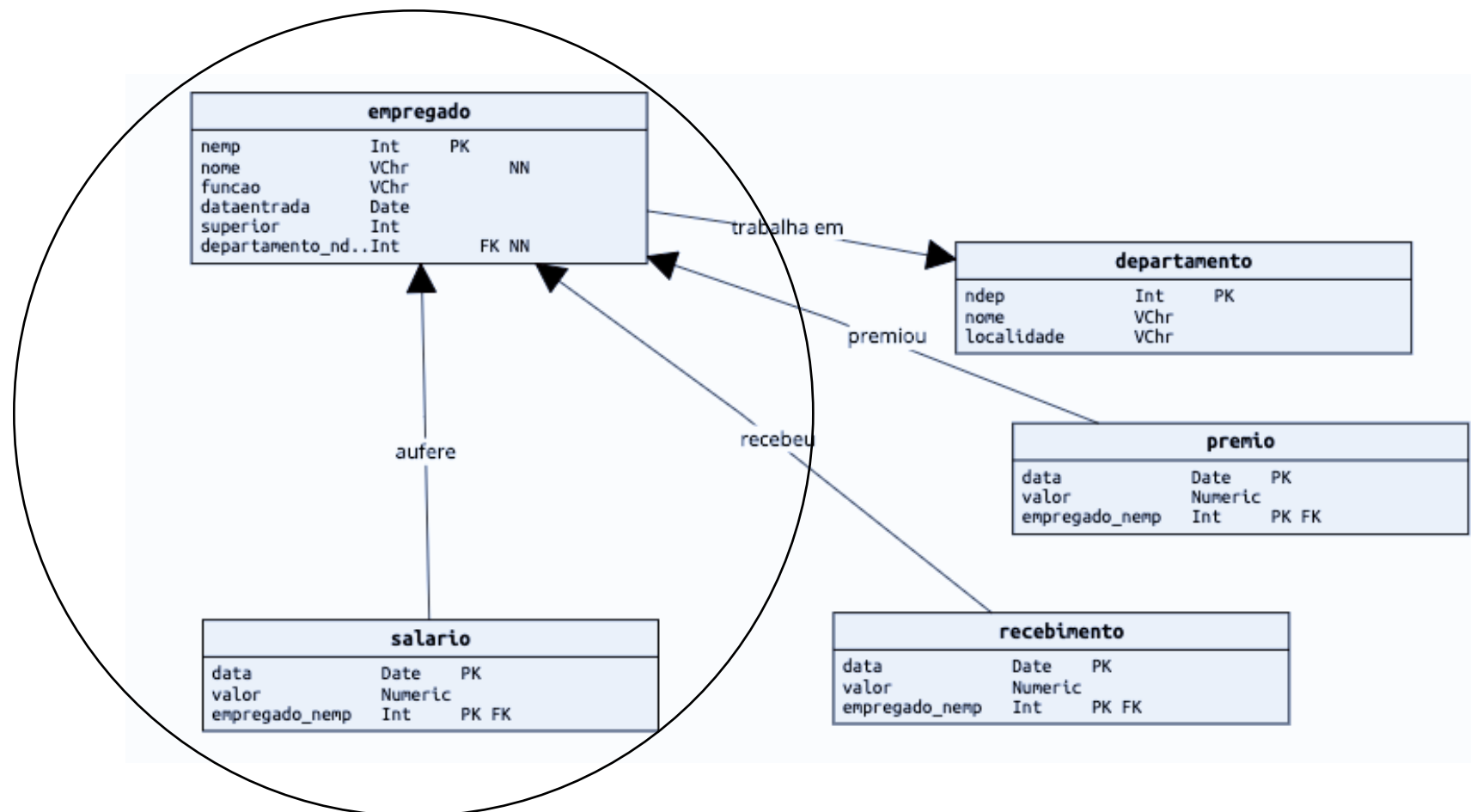
Agrupar ao longo do tempo

- Porque há a tabela salario e a tabela recebimento?
- Porque há a tabela premio?
- Qual a chave primária de salario, recebimento e premio?
PORQUÊ?



Agrupar ao longo do tempo

- Como analisar a evolução dos salários dos analistas ao longo dos anos?



Agrupar ao longo do tempo

- Como analisar a evolução dos salários dos analistas ao longo dos anos?

```
Select avg(valor), max(valor), min(valor) , to_char(data,'yyyy') as ano  
from salario, empregado  
Where salario.empregado_nemp=empregado.nemp  
and função='Analista'  
Group by to_char(data,'yyyy');
```

1. E se eu quiser a evolução de salários de cada função?

E agora comparar a evolução entre funções tb

- Como analisar a evolução dos salários de cada função ao longo dos anos?

```
Select avg(valor), max(valor), min(valor) , to_char(data,'yyyy') as  
ano, funcao  
from salario, empregado  
Where salario.empregado_nemp=empregado.nemp  
Group by to_char(data,'yyyy'), função  
Order by função, to_char(data,'yyyy');
```


1. E se for ao longo dos meses de todos os anos na BD?
(evolução de salários de cada função?)

```
Select avg(valor), max(valor), min(valor) , to_char(data,'yyyy') as  
ano, funcao  
from salario, empregado  
Where salario.empregado_nemp=empregado.nemp  
Group by to_char(data,'yyyy'), função  
Order by função, to_char(data,'yyyy');
```

Ao longo dos meses ...

- Como analisar a evolução dos salários de cada função ao longo dos anos?

```
Select avg(valor), max(valor), min(valor) , to_char(data,'yyyy') as  
ano, to_char(data,'mm') as mes, funcao  
from salario, empregado  
Where salario.empregado_nemp=empregado.nemp  
Group by to_char(data,'yyyy'), to_char(data,'mm'), função  
Order by função, 4, 5;
```

1. E se eu quiser saber o salário que representa o percentil 25 dos salários? O que quer isso dizer?
2. E se eu quiser saber a percentagem de empregados que teem salario acima ou abaixo de um dado valor?

PERCENTIL

Agregações: percentil

```
select percentile_cont(0.25)
within group (order by sal asc) as percentile_25 from emp;
```

percentile_25
89500

```
select percentile_cont(0.75)
within group (order by sal asc) as percentile_75 from emp;
```

percentile_75
421462.5

Agregações: percentil

```
SELECT funcao, round(AVG(sal),1) ,
       round(stddev(sal),1), max(sal), min(sal),
       percentile_cont(0.75) within group (order by sal asc)
       as perc75,
       percentile_cont(0.25) within group (order by sal asc)
       as perc25
FROM emp
GROUP BY funcao
ORDER BY funcao ASC;
```

funcao	round	round	max	min	perc75	perc25
-----+-----+-----+-----+-----						
Analista	500000.0	91923.9	565000	435000	532500	467500
Continuo	65287.5	6028.0	70800	56950	68925	63062.5
Encarregado	370425.0	86236.4	450975	279450	415912.5	330150
Presidente	890000.0		890000	890000	890000	890000
Vendedor	184225.0	38063.2	221250	145600	214500	154750
(5 rows)						

Having

1. E se eu só quiser ver as funções que têm salário médio acima de x ?

Having

```
SELECT funcao, round(AVG(sal),1) ,
       round(stddev(sal),1), max(sal), min(sal)
FROM emp
GROUP BY funcao
HAVING round(AVG(sal),1) >100000
ORDER BY funcao ASC;
```

UC

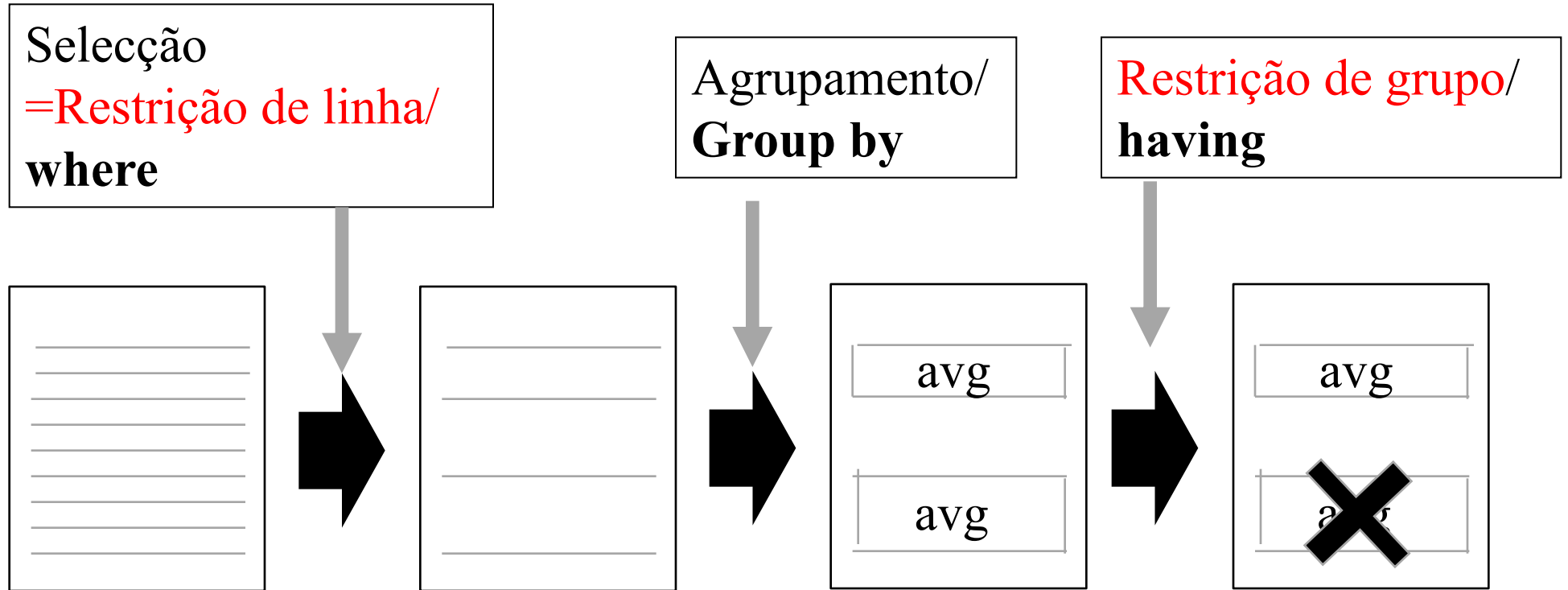
funcao	round	round	max	min
-----+-----+-----+-----				
Analista	500000.0	91923.9	565000	435000
Encarregado	370425.0	86236.4	450975	279450
Presidente	890000.0		890000	890000
(3 rows)				



funcao	round	round	max	min
-----+-----+-----+-----				
Analista	500000.0	91923.9	565000	435000
Contínuo	65287.5	6028.0	70800	56950
Encarregado	370425.0	86236.4	450975	279450
Presidente	890000.0		890000	890000
Vendedor	184225.0	38063.2	221250	145600
(5 rows)				

bdai

O pipeline de restrições



Agregação – group by - having

Select função, ndep, avg(14*sal), sum(sal)
From emp
Group by função, ndep

$\gamma_{\text{função, ndep}}(\text{função, ndep, avg(14*sal), sum(sal)}) (\text{emp})$

$\pi_{\text{função, ndep, avg(14*sal), sum(sal)}} \{$
 $\quad (\gamma_{\text{função, ndep}}(\text{função, ndep, avg(14*sal), sum(sal)}) (\text{emp}))$
 $\quad \}$

Agregação – group by - having

```
Select função, ndep, avg(14*sal), sum(sal)
From emp
Group by função, ndep
Having sum(sal)<20000
```

$$\pi_{\text{função, ndep, avg(14*sal), sum(sal)}} \{$$
$$H_{\text{sum(sal)<20000}} (\gamma_{\text{função, ndep}} (\text{função, ndep, avg(14*sal), sum(sal)) (emp))$$
$$\}$$

Equivalent in Relational Algebra

```
SELECT emp.nome, sal, funcao, dep.nome  
FROM emp, dep  
WHERE  
emp.ndep=dep.ndep AND ndep = 10 AND sal > 200;
```

UC

$$\pi_{\text{emp.nome, sal, funcao, dep.nome}} [\sigma_{\text{emp.sal} > 200} (\sigma_{\text{dep.ndep} = 10} (\text{emp} \bowtie_{\text{ndep} = \text{ndep}} \text{dep}))]$$

bd

Write equivalent in Relational Algebra

```
SELECT emp.nome, sal, funcao, dep.nome  
FROM emp, dep  
WHERE  
emp.ndep=dep.ndep AND ndep = 10 AND sal > 200;
```

FAZER: Equivalente em Algebra Relacional

```
SELECT emp.nome, escalao, dep.nome  
FROM emp, dep, descontos  
WHERE  
emp.ndep=dep.ndep AND dep.nome='Contabilidade' AND sal <250  
and sal between salinf and salsup;
```

UC

```
SELECT emp.nome, funcao, sal  
FROM emp INTERSECT (select * from emp where nemp<3000)  
where sal < 250  
and funcao='Contabilista';
```

E como simplificar = otimizar esta ultima pesquisa?

bd

RESPOSTA: Equivalente em Algebra Relacional

```
SELECT emp.nome, escalao, dep.nome
FROM emp, dep, descontos
WHERE
emp.ndep=dep.ndep AND dep.nome='Contabilidade' AND sal <250
and sal between salinf and salsup;
```

$$\pi_{emp.nome, escalao, dep.nome} \left(\sigma_{dep.nome='Contabilidade'} \left(\sigma_{sal < 250} \left(dep \bowtie_{emp.ndep=dep.ndep} (emp \bowtie_{sal \text{ between } salinf \text{ and } salsup} descontos) \right) \right) \right)$$

```
SELECT emp.nome, funcao, sal
FROM emp INTERSECT (select * from emp where nemp<3000)
where sal < 250
and funcao='Contabilista';
```

E como simplificar = otimizar esta pesquisa?

```
SELECT emp.nome, dep.nome, avg(sal)
FROM emp JOIN dep ON
(emp.ndep=dep.ndep)
where sal < 250
Group by emp.nome, dep.nome
```


Subconsultas

Sub-consultas

- Mostra o nome e salário do empregado que recebe menos:

nome	sal
Tome Ribeiro	56950

(1 row)

- Como faria? (intuitivo)
- Porque está mal?

Sub-consultas

- Mostra o nome e salário do empregado que recebe menos:

nome	sal
Tome Ribeiro	56950

(1 row)

- Porque nao funciona o seguinte?

(Forma ERRADA habitual)

```
select nome, min(sal)  
from emp
```

← ERRO

Sub-consultas

Mostra o nome e salário do empregado que recebe menos:

```
select nome, sal
from emp
where sal=(
    select min(sal)
    from emp
);
```

nome	sal
-----+	
Tome Ribeiro	56950

(1 row)

Sub-consultas

Mostra o empreg de salário mínimo de cada função: porque está mal?

O que pode mostrar que nao era suposto?

```
select nome, função, sal
from emp
where sal in (  select min(sal)
                  from emp
                  group by funcao
                );
```

Sub-consultas

Mostra o salário mínimo de cada função:

```
select funcao, sal
from emp
where (função,sal)=(
    select função, min(sal)
    from emp
    group by funcao
);
```

Sobre sub-consultas do from

For each person... show his name, job and avg sal of that job

nome	funcao	avgsal
Jorge Sampaio	Presidente	890000.000000000000
Augusto Reis	Encarregado	370425.000000000000
Duarte Guedes	Encarregado	370425.000000000000
Silvia Teles	Encarregado	370425.000000000000
Maria Dias	Analista	500000.000000000000
Catarina Silva	Analista	500000.000000000000
Joana Mendes	Vendedor	184225.000000000000
Nelson Neves	Vendedor	184225.000000000000
Ana Rodrigues	Vendedor	184225.000000000000
Manuel Madeira	Vendedor	184225.000000000000
Tome Ribeiro	Continuo	65287.500000000000
Rita Pereira	Continuo	65287.500000000000
Olga Costa	Continuo	65287.500000000000
Antonio Silva	Continuo	65287.500000000000
(14 rows)		

O que faz isto?

```
select funcao, avg(sal) as avgsal  
from emp  
group by funcao
```

O que faz isto?

```
select funcao, avg(sal) as avgsal  
from emp  
group by funcao
```

Subconsulta do from permite...

Definir com select este conteúdo como “Tabela temporária”
em subconsulta ...

para depois ser usado na consulta de fora...

For each person... show his name, job and avg sal of that job

- Show name and job;
- Get another dataset with avg salaries of jobs
- Use it joining by job

nome	funcao	avgsal
Jorge Sampaio	Presidente	890000.000000000000
Augusto Reis	Encarregado	370425.000000000000
Duarte Guedes	Encarregado	370425.000000000000
Silvia Teles	Encarregado	370425.000000000000
Maria Dias	Analista	500000.000000000000
Catarina Silva	Analista	500000.000000000000
Joana Mendes	Vendedor	184225.000000000000
Nelson Neves	Vendedor	184225.000000000000
Ana Rodrigues	Vendedor	184225.000000000000
Manuel Madeira	Vendedor	184225.000000000000
Tome Ribeiro	Continuo	65287.500000000000
Rita Pereira	Continuo	65287.500000000000
Olga Costa	Continuo	65287.500000000000
Antonio Silva	Continuo	65287.500000000000
(14 rows)		

Sub-consultas do from: o que faz o seguinte?

SELECT nome, e.funcao, **nometabela**.attr, sal
FROM emp as e,
(**COMO SE DEFINISSE TABELA (temp) COM SELECT(atributos)**) as **nometabela**
where e.funcao= **nometabela**.funcao; ➡ Juncao das duas tabelas

UC

Define com select
O conteúdo de
“Tabela temporária”
em subconsulta



Usa
“Tabela temporária”
Na pesquisa externa

bdai

Sub-consultas do from: o que faz o seguinte?

```

SELECT nome, e.função, e.sal, funcaosal.avgsal
FROM emp as e,      (select funcao, avg(sal) as avgsal
                     from emp
                     group by funcao
                     ) as funcaosal
where e.funcao=funcaosal.funcao;

```

nome	funcao	avgsal
Jorge Sampaio	Presidente	890000.000000000000
Augusto Reis	Encarregado	370425.000000000000
Duarte Guedes	Encarregado	370425.000000000000
Silvia Teles	Encarregado	370425.000000000000
Maria Dias	Analista	500000.000000000000
Catarina Silva	Analista	500000.000000000000
Joana Mendes	Vendedor	184225.000000000000
Nelson Neves	Vendedor	184225.000000000000
Ana Rodrigues	Vendedor	184225.000000000000
Manuel Madeira	Vendedor	184225.000000000000
Tome Ribeiro	Continuo	65287.500000000000
Rita Pereira	Continuo	65287.500000000000
Olga Costa	Continuo	65287.500000000000
Antonio Silva	Continuo	65287.500000000000

(14 rows)

Other SQL details

Como obter uma amostra aleatória de linhas de tabela?

Amostra aleatória... 20%

select *

from emp

where random() < 0.2;

Outras operações de SQL

- 1. Design database**
- 2. Create tables**
3. Use tables

Data definition language (DDL)

- **CREATE TABLE**
- **DROP TABLE**
- **ALTER TABLE**

Examples: creating tables

```
CREATE TABLE DEP
(NDEP NUMBER PRIMARY KEY,
NOME VARCHAR2(15) NOT NULL UNIQUE);
```

```
CREATE TABLE EMP
(NEMP NUMBER,
NDEP NUMBER REFERENCES DEP(NDEP),
NOME VARCHAR2(20),
. . .
. . . ,
PRIMARY KEY (NEMP));
```

Creating a table from another table

```
CREATE TABLE <nome da tabela>  
    [(Nome_coluna, ...)]  
AS SELECT <instrução>;
```

Exemplo:

```
CREATE TABLE AMOSTRA  
AS  
SELECT NOME, N_ELEITOR  
FROM   ELEITOR  
WHERE  N_ELEITOR <= 2000;
```

You may specify new
names for columns

Restrictions are lost

Eliminar uma tabela

DROP TABLE <nome da tabela>;

Exemplo:

DROP TABLE ELEITOR;

Add columns to a table

ALTER TABLE *<nome da tabela>*

ADD (Nome_coluna tipo(tamanho) [*<restrições de integridade>*],
.....);

Exemplo:

ALTER TABLE ELEITOR

ADD (NOME_FREG CHAR(60) NOT NULL);

Exemplo: adicionar coluna NOT NULL

- A tabela eleitor já tem 7 Milhoes de eleitores
- Preciso de adicionar a coluna 'NomeFreguesia', NOT NULL

Exemplo: adicionar coluna NOT NULL

- A tabela eleitor já tem 7 Milhoes de eleitores
- Preciso de adicionar a coluna 'NomeFreguesia', NOT NULL

alter table eleitor, add column NomeFreguesia varchar(100);

update eleitor set NomeFreguesia='NAO_PREENCHIDO';

alter table eleitor modify NomeFreguesia add constraint NOT NULL;

Change the type of a column

```
ALTER TABLE <nome da tabela>  
    MODIFY ( Nome_coluna    novo_tipo(tamanho),  
            .....);
```

Exemplo:

```
ALTER TABLE ELEITOR  
    MODIFY N_ELEITOR    NUMBER(8);
```

Add, change, delete a constraint

ALTER TABLE ELEITOR

ADD CONSTRAINT ELEITOR_CP PRIMARY KEY (N_ELEITOR);

ALTER TABLE ELEITOR

MODIFY N_ELEITOR NULL;

ALTER TABLE ELEITOR

DROP CONSTRAINT ELEITOR_CP;

Eliminate a constraint

ALTER TABLE *<nome da tabela>*

DROP CONSTRAINT *<nome_restrição>;*

Exemplo:

ALTER TABLE ELEITOR

DROP CONSTRAINT ELEITOR_CP;

Manipulação de dados

Posso manipular dados em tabelas

Uso: o dia-a-dia de qualquer aplicação/uso de bases de dados

1. Projecta
2. Cria tabelas
- 3. Programa / Usa**

Data manipulation language (DML)

Comandos DML da linguagem SQL

- **INSERT**
- **UPDATE**
- **DELETE**

Insert new records into a table

```
INSERT INTO <nome da tabela> [(coluna1, coluna2,...)]  
VALUES (valor1, valor2, ...);
```

Exemplo:

```
INSERT INTO ELEITOR (BI, NOME, END, N_ELEITOR)  
VALUES (4537687, 'António Silva', 'R. Carlos seixas, 29, 1º, Esq', 2075);
```

Remove headings...

```
INSERT INTO ELEITOR (BI, NOME, END, N_ELEITOR)  
VALUES (4537687, 'António Silva', 'R. Carlos seixas, 29, 1º, Esq', 2075);
```

```
INSERT INTO ELEITOR  
VALUES (4537687, 'António Silva', 'R. Carlos seixas, 29, 1º, Esq', 2075);
```


Insert multiple records into a table

```
INSERT INTO <nome da tabela> [(coluna1, coluna2,...)]
```

```
VALUES (valor1, valor2, ...),
```

```
(valor21, valor22, ...);
```

Exemplo:

```
INSERT INTO ELEITOR (BI, NOME, END, N_ELEITOR)
```

```
VALUES (4537687, 'António Silva', 'R. Carlos seixas, 29, 1º, Esq', 2075),
```

```
VALUES (4537688, 'Manuel Silva', 'R. Carlos seixas, 39, 3º, Esq', 2074);
```

Insert records from another table

```
INSERT INTO <nome da tabela> [(coluna1, coluna2,...)]
```

```
SELECT (lista)
```

```
FROM <tabela(s)>
```

```
WHERE <condição>;
```

Exemplos:

```
INSERT INTO AMOSTRA
```

```
SELECT NOME, N_ELEITOR
```

```
FROM ELEITOR
```

```
WHERE N_ELEITOR <= 2000;
```

```
INSERT INTO AMOSTRA (NOMEFamilia, ELEITOR)
```

```
SELECT NOME, N_ELEITOR
```

```
FROM ELEITOR
```

```
WHERE N_ELEITOR <= 2000;
```

Atualizar registos

```
UPDATE <nome da tabela> [pseudónimo]  
    SET      coluna = (valor, expressão,..),  
              coluna = (valor, expressão,..), ....  
    WHERE <condição>;
```

Exemplo:

```
UPDATE ELEITOR  
    SET      NOME = 'António Dias da Silva',  
              BI = 4537687  
    WHERE N_ELEITOR = 2075;
```

Examples: modify data

```
UPDATE DEP  
SET LOCAL='Maputo'  
WHERE NDEP=5;
```

If the **WHERE** clause is omitted, what happens?

. . .

Examples: modify data

```
UPDATE DEP  
SET LOCAL='Maputo'  
WHERE NDEP=5;
```

**Write query: update salary by +10%
For department 'Vendas'**

. . .

Examples: erase data

```
DELETE FROM DEP  
WHERE NDEP=5;
```

```
DELETE FROM DEP  
WHERE NDEP IN  
    (SELECT NDEP  
     FROM DEP  
     WHERE NOME= 'Vendas' );
```

. . .

DELETE – dilema...

- Why is DELETE an infrequent operation in databases?
-

What happens if i use the rollback?

begin

delete from emp1;

rollback;

select * from emp1;

... But be careful with autocommit...