

SQL

Select

José R. Paramá



Guión

Introducción

Sqlplus

Táboas usadas nos exemplos

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Introdución

Parte do SEQUEL (*Structured English QUery Language*), desenvolvido en laboratorios de IBM para o SYSTEM R.

Baséase na álgebra relacional e o cálculo relacional.

Esta linguaxe evolucionou cara á SQL (*Structured Query Language*).

ANSI decidiu estandarizala; chamoulle SQL-86 ou SQL1. ISO aceptouna en 1987.

Despois apareceron outros estándares: SQL-92 (SQL2), SQL:1999 (SQL3), SQL:2003, SQL:2006, SQL:2008 e SQL:2011

Introdución

A meirande parte é seguida polos fabricantes, pero hai pequenas diverxencias, por iso a SQL que segue cada produto se denomina *dialecto*.

A pesar do seu nome, non é só unha linguaxe de consulta, senón que cobre:

- DDL ou LDD (linguaxe de definición de datos).

- DML ou LMD (linguaxe de manipulación de datos).

Con SQL pódese realizar calquera tarefa dentro dun SXBD (crear usuarios, dar permisos, control de concorrencia, creación de estruturas de almacenamento e acceso aos datos etc.).

Introdución

Outras características:

É posible inserir SQL dentro de programas escritos con linguaxes de propósito xeral.

É unha linguaxe non procedemental (especificácase o *que* queremos, e non especificamos o *como*).

Non manipula conxuntos de filas (como o modelo relacional teórico); manexa *coleccións* de filas (non hai orde, pero pode haber filas repetidas).

Cambia tamén a terminoloxía: *táboas, filas e columnas* en lugar de *relacións, tuplas e atributos*.

SQL*Plus de Oracle

Programa para executar sentenzas de SQL nun contorno Oracle.

Ten unha serie de comandos propios.

Entrar:

```
$ sqlplus
```

Nome de usuario: nome de usuario da UDC entre comiñas dobres; ex.: "brais.souto"

Contrasinal: DNI máis letra en maiúscula. Ex.: 41222444R

Se entras antepoñendo *rlwrap*:

```
$ rlwrap sqlplus
```

Coas teclas ↑↓ podes acceder ás últimas consultas

Cambiar contrasinal:

```
password
```

SQL*Plus

Os comandos SQL*plus escríbense nunha única liña e non necesitan ;

```
DESCRIBE emp
```

As sentenzas SQL pódense escribir en varias filas e acaban en ;

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

SQL*Plus

Comandos:

DESC[RIBE] <tabla> mostra a estrutura (columnas e tipos de datos) da táboa.

```
SQL> DESCRIBE emp
```

Nome	Nulo?	Tipo
-----	-----	-----
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

EXIT saír

SQL*Plus

SQL*Plus usa un búfer para almacenar a última sentenza SQL executada (non afecta aos comandos SQL*Plus). Incorpora un editor de liñas elemental para realizar cambios mínimos nesa sentenza SQL:

Comandos para usar o búfer:

`L[IST]`: Visualiza o contido do búfer

`ED[IT]`: Abre o contido do búfer nun editor (modo texto).

`R[UN]`: Executa o contido do búfer.

`SAV[E] fichero[.sql]`: Salva o contido do búfer nun ficheiro con nome `fichero[.sql]`.

`GET fichero[.sql]`: Carga o contido do ficheiro `fichero[.sql]` no búfer.

Para configurar o editor de texto e outros aspectos do contorno, descarga o ficheiro `login.sql` do Moodle, e ponlo nun directorio onde *sempre debes arrancar* o SQL*plus.

Se non tes cargado o `login.sql`, podes escribir `DEFINE_EDITOR=pico` (OU `DEFINE_EDITOR=vi` se o sabes manexar) para configurar o editor.

Táboas usadas nos exemplos

EMP		
Campo	Tipo	Descrición
<u>EMPNO</u>	NUMBER(4) NOT NULL	Número ou código do empregado. É a clave primaria da táboa.
ENAME	VARCHAR2(10)	Nome do empregado
JOB	VARCHAR2(9)	Traballo do empregado
MGR	NUMBER(4)	Código do xefe do empregado. Clave foránea que referencia (ciclicamente) a táboa EMP
HIREDATE	DATE	Data de contratación.
SAL	NUMBER(7, 2)	Salario mensual do empregado
COMM	NUMBER(7, 2)	Comisión
DEPTNO	NUMBER(2)	Código do departamento ao que o empregado está adscrito. Clave foránea que referencia a táboa DEPT

Un nulo en COMM significa que o empregado non traballa a comisión (o valor non procede).

Un nulo en MGR significa que non ten xefe (tamén "non procede")

DEPT		
Campo	Tipo	Descrición
<u>DEPTNO</u>	NUMBER(2) NOT NULL	Número ou código do departamento. É a clave primaria da táboa.
DNAME	VARCHAR2(14)	Nome do departamento.
LOC	VARCHAR2(13)	Localidade (ou cidade) onde o departamento está localizado.

PRO		
Campo	Tipo	Descrición
<u>PRONO</u>	NUMBER(4) NOT NULL	Número ou código do proxecto. É a clave primaria da táboa.
PNAME	VARCHAR2(10)	Nome do proxecto.
LOC	VARCHAR2(13)	Cidade onde se realiza o proxecto.
DEPTNO	NUMBER(2)	Número do departamento controlador do proxecto. Clave foránea que referencia a táboa DEPT

EMPPRO		
Campo	Tipo	Descrición
<u>EMPNO</u>	NUMBER(4) NOT NULL	Número ou código do empregado. Clave foránea que referencia a táboa EMP
<u>PRONO</u>	NUMBER(4) NOT NULL	Número ou código do proxecto. Clave foránea que referencia a táboa PRO
HOURS	NUMBER(2)	Horas que leva traballadas un empregado nun proxecto.

É a clave primaria da táboa

Táboas usadas nos exemplos

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30
7566	JONES	MANAGER	7839	02/04/81	2,975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20
7839	KING	PRESIDENT		17/11/81	5,000		10
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1,100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3,000		20
7934	MILLER	CLERK	7782	23/01/82	1,300		10

14 filas seleccionadas.

```
SQL> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

4 filas seleccionadas.

```
SQL> select * from emppro;
```

EMPNO	PRONO	HOURS
7499	1004	15
7499	1005	12
7521	1004	10
7521	1008	8
7654	1001	16
7654	1006	15
7654	1008	5
7844	1005	6
7934	1001	4

9 filas seleccionadas.

```
SQL> select * from pro;
```

PRONO	PNAME	LOC	DEPTNO
1001	P1	BOSTON	20
1004	P4	CHICAGO	30
1005	P5	CHICAGO	30
1006	P6	LOS ANGELES	30
1008	P8	NEW YORK	30

5 filas seleccionadas.

Guión

Introducción

Conceptos previos

Nulos

Expresións

Senzenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Conceptos previos

Nulos:

O valor nulo **NULL** representa a *ausencia de información*, ou ben por descoñecemento do dato, ou ben porque non procede.

Debe diferenciarse de calquera outro valor, entre eles do valor 0 se se trata dun dato numérico, e da cadea de caracteres baleira, se é un dato de tipo carácter.

Conceptos previos

Expresión¹:

Unha expresión é a formulación dunha secuencia de operacións, ou sexa, unha combinación de operadores, operandos e parénteses, que, cando se executa, *devolve un único valor escalar como resultado*.

Os operandos poden ser constantes, nomes de columna, funcións, outras expresións e outros elementos.

O tipo de dato de cada operando dunha expresión debe ser o mesmo. Se un operando é nulo, o resultado tamén é nulo.

Operadores numéricos: + - * /

Operador alfanumérico: || (concatenación de cadeas de texto)

Exemplos:

- 3
- 'Casa'
- 3+2
- 'A' || 'BC'
- ENAME
- SAL*1.5
- 0.5 * COMM
- SAL + COMM
- (Select COMM FROM EMP WHERE EMPNO = 7499)

Non son expresións:

- SAL < 1500
- (SAL+COMM) >=10

¹Restrinximos a definición de expresión á versión “Core SQL” do estándar.

Conceptos previos

En Oracle, o texto (literal de texto) vai entre comiñas simples e é sensible ás maiúsculas/minúsculas.

‘casa’

‘Casa’

‘Casa bonita’

Os identificadores (de usuario, nomes de columnas...) van entre comiñas dobres. Pódense omitir as comiñas cando o identificador non ten espazos ou símbolos de puntuación.

psanchez

“Pedro Sanchez”

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsulta

Composición de consultas

Sentenza Select

A sentenza SELECT permite seleccionar ou obter datos dunha ou de varias táboas.

Parte dunha ou de varias táboas da BD e o resultado é outra táboa, denominada ás veces *táboa resultado*, pero que non formará parte da BD.

```
SELECT [DISTINCT|ALL] { * | <expr1>[, <expr2>] ... }  
  FROM <tabla1>[[INNER|LEFT|RIGHT|FULL|CROSS] JOIN <tabla2> ...]  
  [WHERE <condicion_where>]  
  [GROUP BY <columna1>[,<columna2>, ... ]]  
  [HAVING <condicion_having>]  
  [ORDER BY <expr_orderby1>[, ... ]]
```

Sentenza Select

```
SELECT [DISTINCT|ALL] { * | <expr1>[, <expr2>] ...}  
      FROM <tabla1>[[INNER|LEFT|RIGHT|FULL|CROSS] JOIN <tabla2> ...]  
      [WHERE <condicion_where>]  
      [GROUP BY <columna1>[,<columna2>,...]  
      [HAVING <condicion_having>]  
      [ORDER BY <expr_orderby1>[,...]]
```

A orde de execución das cláusulas e a función de cada unha é:

1. **FROM**(obrigatoria)

Partindo dunha ou máis táboas *obtén una única táboa* que será procesada polo resto de cláusulas

2. **WHERE** (optativa)

Das filas que lle pasa o FROM, *elimina as filas que NON FAN CERTA a condición* especificada

3. **GROUP BY** (optativa)

4. **HAVING** (optativa)

5. **SELECT** (obrigatoria)

Cada fila que lle chega é usada para obter unha fila do resultado.

Procésanse as filas unha por unha; cando se procesa unha fila, avalíase sobre as expresións, *cada expresión dá lugar a unha columna da táboa resultado*.

Alternativamente un * indica que no resultado se engadan todas as columnas.

Se houbese *filas repetidas*, aparecen de forma predefinida, pero non o fan se se inclúe **DISTINCT**.

6. **ORDER BY** (optativa)

Permite determinar o criterio de ordenación das filas da táboa resultado. Sen ela, obteremos as mesmas filas, pero non hai garantía de en que orde, que será a que dite a estratexia seguida polo SXBD para extraer os datos.

Sentenza Select

```
Select 'Nombre: ' || ename, sal*0.20
```

```
from emp
```

```
where deptno=10;
```

O resultado deste passo é

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30
7566	JONES	MANAGER	7839	02/04/81	2,975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20
7839	KING	PRESIDENT		17/11/81	5,000		10
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1,100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3,000		20
7934	MILLER	CLERK	7782	23/01/82	1,300		10

Sentenza Select

```
Select 'Nombre: ' || ename, sal*0.20
from emp
where deptno=10;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	
7369	SMITH	CLERK	7902	17/12/80	800		20	Falso
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30	Falso
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30	Falso
7566	JONES	MANAGER	7839	02/04/81	2,975		20	Falso
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30	Falso
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30	Falso
7782	CLARK	MANAGER	7839	09/06/81	2,450		10	Certo: continúa
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20	Falso
7839	KING	PRESIDENT		17/11/81	5,000		10	Certo: continúa
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30	Falso
7876	ADAMS	CLERK	7788	23/05/87	1,100		20	Falso
7900	JAMES	CLERK	7698	03/12/81	950		30	Falso
7902	FORD	ANALYST	7566	03/12/81	3,000		20	Falso
7934	MILLER	CLERK	7782	23/01/82	1,300		10	Certo: continúa

O resultado deste passo é

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7839	KING	PRESIDENT		17/11/81	5,000		10
7934	MILLER	CLERK	7782	23/01/82	1,300		10

A sentença Select

```
Select 'Nombre: ' || ename, sal*0.20
```

```
from emp
```

```
where deptno=10;
```

Cada coma separa dúas expresións
e cada expresión dá lugar a unha columna
no resultado

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7839	KING	PRESIDENT		17/11/81	5,000		10
7934	MILLER	CLERK	7782	23/01/82	1,300		10

'Nombre: '	ename	sal*0.20
→ 'Nombre: '	CLARK	2450*0.20
→ 'Nombre: '	KING	5000*0.20
→ 'Nombre: '	MILLER	1300*0.20

O resultado deste paso é

'NOMBRE: '	ENAME	SAL*0.20
Nombre: CLARK		490
Nombre: KING		1000
Nombre: MILLER		260

A sentencia Select

Pódese cambiar o nome dunha columna.

```
SELECT  <expr1> [AS] nuevo_nombre, ...
```

```
select ename as nome, sal salario, sal+comm as "ingresos mensuais", hiredate "fecha contratación"  
from emp;
```

nombre character varying(10)	salario numeric(7,2)	ingresos mensuales numeric	fecha contratación date
KING	5000.00		1981-11-17
JONES	2975.00		1981-04-02
FORD	3000.00		1981-12-03
SMITH	800.00		1980-12-17
BLAKE	2850.00		1981-05-01
ALLEN	1600.00	1900.00	1981-02-20
WARD	1250.00	1750.00	1981-02-22
MARTIN	1250.00	2650.00	1981-09-28
CLARK	2450.00		1981-06-09
SCOTT	3000.00		1982-12-09
TURNER	1500.00	1500.00	1981-09-08
ADAMS	1100.00		1983-01-12
JAMES	950.00		1981-12-03
MILLER	1300.00		1982-01-23

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Distinct

Se se inclúe ***DISTINCT*** antes das expresións na cláusula select, eliminaranse ***FILAS REPETIDAS***.

```
SQL> select job, deptno  
2  from emp  
3  order by job, deptno;
```

JOB	DEPTNO
-----	-----
ANALYST	20
ANALYST	20
CLERK	10
CLERK	20
CLERK	20
CLERK	30
MANAGER	10
MANAGER	20
MANAGER	30
PRESIDENT	10
SALESMAN	30
SALESMAN	30
SALESMAN	30
SALESMAN	30

14 filas seleccionadas.

```
SQL> select distinct job, deptno  
2  from emp  
3  order by job, deptno;
```

JOB	DEPTNO
-----	-----
ANALYST	20
CLERK	10
CLERK	20
CLERK	30
MANAGER	10
MANAGER	20
MANAGER	30
PRESIDENT	10
SALESMAN	30

9 filas seleccionadas.

Distinct

Os nulos, *para o distinct, son iguais*

```
SQL> select deptno, comm  
2  from emp  
3  order by deptno, comm;
```

DEPTNO	COMM
--------	------

10	
10	
10	
20	
20	
20	
20	
20	
30	0
30	300
30	500
30	1,400
30	
30	

14 filas seleccionadas.

```
SQL> select distinct deptno, comm  
2  from emp  
3  order by deptno, comm;
```

DEPTNO	COMM
--------	------

10	
20	
30	0
30	300
30	500
30	1,400
30	

7 filas seleccionadas.

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Order by

ORDER BY <expr_orderby1> [ASC|DESC], <expr_orderby2> [ASC|DESC],...

Ordena as *FILAS* obtidas.

As expresións de order by poden ser *expresións que non sexan un literal* (unha constante).

Tanto a expresión como a columna non teñen que aparecer necesariamente na cláusula select.

Correctas

```
select ename, job  
from emp  
order by hiredate;
```

```
select ename, job  
from emp  
order by sal+comm;
```

Order by

Se non se indica nada, o ordenamento predefinido é ascendente (ASC).

```
select ename, sal  
from emp  
order by sal;
```

ename character varying(10)	sal numeric(7,2)
SMITH	800.00
JAMES	950.00
ADAMS	1100.00
WARD	1250.00
MARTIN	1250.00
MILLER	1300.00
TURNER	1500.00
ALLEN	1600.00
CLARK	2450.00
BLAKE	2850.00
JONES	2975.00
SCOTT	3000.00
FORD	3000.00
KING	5000.00

```
select ename, sal  
from emp  
order by sal DESC;
```

ename character varying(10)	sal numeric(7,2)
KING	5000.00
SCOTT	3000.00
FORD	3000.00
JONES	2975.00
BLAKE	2850.00
CLARK	2450.00
ALLEN	1600.00
TURNER	1500.00
MILLER	1300.00
MARTIN	1250.00
WARD	1250.00
ADAMS	1100.00
JAMES	950.00
SMITH	800.00

Order by

Pódese usar tamén o nome de columna (no canto de usar a expresión que a define).

```
select ename as nombre, sal salario, sal+comm as "ingresos mensuales",  
hiredate "fecha contratación"  
from emp  
order by "ingresos mensuales"
```

Order by

Se hai varias expresións de ordenamento, ordénanse as *filas* primeiro pola primeira expresión de ordenamento; para aquelas filas co mesmo valor na primeira expresión de ordenamento, desempátase pola segunda expresión de ordenamento, e así sucesivamente.

```
select ename, deptno, sal
from emp
order by deptno, sal;
```

ename	deptno	sal
character varying(10)	numeric(2,0)	numeric(7,2)
MILLER	10	1300.00
CLARK	10	2450.00
KING	10	5000.00
SMITH	20	800.00
ADAMS	20	1100.00
JONES	20	2975.00
FORD	20	3000.00
SCOTT	20	3000.00
JAMES	30	950.00
MARTIN	30	1250.00
WARD	30	1250.00
TURNER	30	1500.00
ALLEN	30	1600.00
BLAKE	30	2850.00

Ordénase
pola
segunda

Igual valor na
primeira

Order by

Pódese ordenar ascendentemente nunhas expresións e descendentemente noutras.

```
select ename, deptno, sal  
from emp  
order by deptno, sal desc;
```

ename	deptno	sal
character varying(10)	numeric(2,0)	numeric(7,2)
KING	10	5000.00
CLARK	10	2450.00
MILLER	10	1300.00
FORD	20	3000.00
SCOTT	20	3000.00
JONES	20	2975.00
ADAMS	20	1100.00
SMITH	20	800.00
BLAKE	30	2850.00
ALLEN	30	1600.00
TURNER	30	1500.00
WARD	30	1250.00
MARTIN	30	1250.00
JAMES	30	950.00

Descendente

Ascendente

Order by

Para *order by*, por convención, os nulos considéranse *maiores* que calquera valor.

```
select ename, sal, comm
from emp
order by comm;
```

ename character varying(10)	sal numeric(7,2)	comm numeric(7,2)
TURNER	1500.00	0.00
ALLEN	1600.00	300.00
WARD	1250.00	500.00
MARTIN	1250.00	1400.00
BLAKE	2850.00	
CLARK	2450.00	
SCOTT	3000.00	
ADAMS	1100.00	
JAMES	950.00	
KING	5000.00	
MILLER	1300.00	
JONES	2975.00	
FORD	3000.00	
SMITH	800.00	

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Predicados elementais

Os predicados permitem especificar unha condición.

Pódense usar nas partes *where* e *having*.

<expr1> <op_condición> <expr2>

<op_condición> puede ser: < <= = != <> >= >

O resultado dun predicado dentro dunha cláusula where, como vimos, aplícase a *unha única fila* e o seu resultado pode ser: *certo (true)*, *falso (false)* ou *descoñecido (null)*.

O motivo do terceiro resultado posible é a presenza de nulos.

Cando <expr1> ou <expr2> é un nulo, o resultado é *descoñecido*.

Predicados elementais

```
Select 'Nombre: ' || ename, sal*0.20
from emp
```

```
where comm>1000;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30
7566	JONES	MANAGER	7839	02/04/81	2,975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20
7839	KING	PRESIDENT		17/11/81	5,000		10
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1,100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3,000		20
7934	MILLER	CLERK	7782	23/01/82	1,300		10

- Descoñecido
- Falso
- Falso
- Descoñecido
- Certo: *continúa*
- Descoñecido
- Descoñecido
- Descoñecido
- Descoñecido
- Falso
- Descoñecido
- Descoñecido
- Descoñecido

Resultado final

?column? text	?column? numeric
nombre: MARTIN	250.0000

Predicados elementais

<expre1> <op_condición> <expre2>

Observa que aos dous lados da condición pode haber *expresións*

```
select ename, job, sal, comm, sal+comm  
from emp  
where sal+comm > 2500;
```

```
select ename, job, sal, comm, sal+comm  
from emp  
where 1000 = 1000; ?????
```

```
select ename, job, sal, comm, sal+comm  
from emp  
where null = null; ?????
```

```
select ename, job, sal, comm, sal+comm  
from emp  
where comm= null; ?????
```

Predicados de nulos

Os predicados de comparación non serven para determinar os valores nulos.

Como vimos, non é válido **COMM = NULL** porque sería discernir se un valor (que tamén pode ser descoñecido) é igual a descoñecido.

Requírese un predicado especial, con formato: **<expr> IS [NOT] NULL**

```
select ename, sal, comm, sal+comm total
from emp
where sal+comm is null;
```

ename	sal	comm	total
character varying(10)	numeric(7,2)	numeric(7,2)	numeric
KING	5000.00		
JONES	2975.00		
FORD	3000.00		
SMITH	800.00		
BLAKE	2850.00		
CLARK	2450.00		
SCOTT	3000.00		
ADAMS	1100.00		
JAMES	950.00		
MILLER	1300.00		

```
select ename, sal, comm, sal+comm total
from emp
where sal+comm is not null;
```

ename	sal	comm	total
character varying(10)	numeric(7,2)	numeric(7,2)	numeric
ALLEN	1600.00	300.00	1900.00
WARD	1250.00	500.00	1750.00
MARTIN	1250.00	1400.00	2650.00
TURNER	1500.00	0.00	1500.00

Predicado Between

Predicado de rango ou predicado **BETWEEN**

Compara se os valores dunha expresión están ou non entre os valores doutras dúas (incluíndo os extremos).

Formato: **<expr1> [NOT] BETWEEN <expr2> AND <expr3>**

```
SELECT *  
FROM emp  
WHERE sal BETWEEN 1500 AND 3000;
```

Predicados de pertenza a conxunto

Predicado de pertenza a conxunto (IN)

Comproba se o valor dunha expresión coincide con algún dos valores incluídos nunha listaxe de expresións.

Formato: **<expr1> [NOT] IN (<expr2>[, <expr3>, ...])**

```
SELECT *  
FROM emp  
WHERE deptno IN (10,30,40);
```

```
SELECT ename  
FROM emp  
WHERE job IN ( 'CLERK' , ' SALESMAN' );
```

<expre1> IN (<expre2>,<expre3>) é o mesmo que **<expre1>=<expre2> OR <expre1>=<expre3>**
<expre1> NOT IN (<expre2>,<expre3>) é o mesmo que **<expre1>!=<expre2> AND <expre1>!=<expre3>**

Predicado LIKE

Predicado de correspondencia cun padrón ou modelo.

Comproba se o valor dunha expresión alfanumérica se corresponde cun modelo. O modelo pode incluír dous caracteres que actúan como comodíns:

- Indica un único carácter, incluído o branco.
- % Indica unha cadea de caracteres de calquera lonxitude, incluída a cadea baleira.

Formato: **<expr1> [NOT] LIKE <modelo>**

```
SELECT *  
FROM emp  
WHERE ename LIKE '%NE%'
```

```
SELECT *  
FROM emp  
WHERE ename LIKE '_____'
```


Predicados compostos

Son a unión de dous ou máis predicados mediante os operadores lóxicos **AND**, **OR** e **NOT**.

Ao existir unha lóxica de tres valores, debemos considerar o efecto do valor **NULL**.

X	Y	X AND Y	X OR Y	NOT X
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
TRUE	DESC.	DESC.	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	DESC.	FALSE	DESC.	TRUE
DESC.	TRUE	DESC.	TRUE	DESC.
DESC.	FALSE	FALSE	DESC.	DESC.
DESC.	DESC.	DESC.	DESC.	DESC.

Predicados compostos

```
Select *
from emp
where sal+comm>2500;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30
7566	JONES	MANAGER	7839	02/04/81	2,975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20
7839	KING	PRESIDENT		17/11/81	5,000		10
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1,100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3,000		20
7934	MILLER	CLERK	7782	23/01/82	1,300		10

Descoñecido
Falso
Falso
Descoñecido
Certo
Descoñecido
Descoñecido
Descoñecido
Descoñecido
Falso
Descoñecido
Descoñecido
Descoñecido
Descoñecido

```
Select *
from emp
where sal+comm>2500
      or sal > 2500;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30
7566	JONES	MANAGER	7839	02/04/81	2,975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20
7839	KING	PRESIDENT		17/11/81	5,000		10
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1,100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3,000		20
7934	MILLER	CLERK	7782	23/01/82	1,300		10

Desc.(Desc. OR Falso)
Falso (Falso OR Falso)
Falso
Certo: (Desc. OR Certo)
Certo: (Certo OR Falso)
Certo: (Desc. OR Certo)
Desc (Desc. OR Falso):
Certo:
Certo:
Falso (Falso OR Falso)
Desc.(Desc. OR Falso)
Desc
Certo: (Desc. OR Certo)
Desc.

Exercicios

1. Mostra os postos de traballo que hai en cada departamento (código de dept e nome do posto de traballo). Non deben aparecer repetidos.
2. Mostra os códigos de empregados que son xefes. No resultado non deben aparecer filas con nulos.
3. Mostra as cidades onde se executan proxectos controlados polo departamento 30. Non deben aparecer repetidos no resultado.
4. Mostra empregados que non teñen xefe.
5. Mostra empregados que teñan xefe e que gañen (incluíndo salario e comisión) máis de 2500.
6. Mostra os empregados cuxo nome comeza por 'S'.
7. Mostra os empregados que gañan (incluíndo salario e comisión) entre 1500 e 2500 euros.
8. Mostra os empregados que son 'CLERK', 'SALESMAN' ou 'ANALYST' e gañan (incluíndo salario e comisión) máis de 1250.

Exercicios

1.

```
Select distinct deptno, job
from emp
```
2.

```
Select mgr
from emp
where mgr is not null
```
3.

```
Select distinct loc
from pro
where deptno=30
```
4.

```
Select empno, ename
from emp
where mgr is null
```
5.

```
Select empno, ename
from emp
where mgr is not null
and (sal>2500 or
sal+comm>2500)
```
6.

```
Select empno, ename
from emp
where ename like 'S%'
```
7.

```
Select empno, ename
from emp
where (sal between 1500 and 2500
and comm is null)
or
(sal+comm between 1500 and 2500)
```
8.

```
Select empno, ename, sal, comm, job
from emp
where job in ('ANALYST','CLERK','SALESMAN')
and
(sal+comm > 1250
or sal >1250)
```

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Escalares

Colectivas

Group by

Having

Join

Subconsultas

Composición de consultas

Funcións escalares

Son funcións que se aplican dentro de expresións *normais* e, por tanto, pódense utilizar en *calquera sitio* onde se agarde unha expresión.

É dicir, sobre expresións que se aplican *SOBRE UNHA FILA* e devolven *un valor para esa fila*.

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)
7839	KING	PRESIDENT		1981-11-17	5000.00		10
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7369	SMITH	CLERK	7902	1980-12-17	800.00		20
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20
7900	JAMES	CLERK	7698	1981-12-03	950.00		30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10

```
select sqrt(sal)
from emp;
```

sqrt numeric
70.710678118654752
54.543560573178572
54.772255750516611
28.284271247461901
53.385391260156556
40.000000000000000
35.355339059327376
35.355339059327376
49.497474683058327
54.772255750516611
38.729833462074169
33.166247903553998
30.822070014844882
36.055512754639893

Funcións escalares

Hai moitas, algúns exemplos:

Numéricas ou aritméticas:

SQRT(<exp_numerica>) Raíz cadrada. Ex.: SQRT(81)

ABS(<exp_numerica>) Valor absoluto. Ex.: ABS(-11)

POWER(<exp1> , <exp2>) Potencia. Ex.: POWER(9,2) = 81

Alfanuméricas ou de cadeas de caracteres:

SUBSTR(<exp1> , <exp2> [, <exp3>]) Subcadea de <exp1> comezando na posición <exp2> e de lonxitude <exp3>.

Ej.: SUBSTR('Materia',2,4) = 'ater' SUBSTR('Materia',5) = 'ria'

UPPER(<exp_caracter>) Pasa a maiúsculas. Ex.: UPPER('Materia') = 'MATERIA'

LOWER(<exp_caracter>) Pasa a minúsculas. Ex.: LOWER('Materia') = 'materia'

LENGTH(<exp_carácter>) conta o número de caracteres

De data e tempo:

CURRENT_DATE Data actual do sistema. Ex.: SELECT CURRENT_DATE FROM DUAL

Funcións escalares

Función útil cando hai valores nulos en expresións.

Formato: **COALESCE**(<expr1>, <expr2>, ...)

Funcionamento:

Avalía a <expr1>. Se o seu valor é distinto de NULL, devolve o dito valor. En caso contrario, avalía a <expr2> e devolve o resultado, e así sucesivamente.

```
SELECT COALESCE(sal + comm, sal)
FROM emp;
```

Neste exemplo avalíase a suma do salario e a comisión de cada empregado. Se o resultado é distinto de NULL, devólvese o dito resultado. Se o resultado é NULL (debido a que a comisión del empregado é NULL), avalíase o salario de cada empregado e devólvese ese valor.

COALESCE (<expr1>, <expr2>, ...) é unha expresión e, por tanto, pódese usar en *calquera sitio* onde se agarde unha expresión, por ex.:

```
SELECT COALESCE(sal + comm, sal)
FROM emp
WHERE COALESCE(comm, 0) + sal > 2500
```


Funcións colectivas

As funcións colectivas (ou de agrupamento, ou de conxuntos), son funcións que se aplican sobre unha *COLECCIÓN DE FILAS* e devolve *un valor para esa colección de filas*.

Unha expresión que contén unha función colectiva segue a ser unha expresión, pero agora xa non se pode utilizar en calquera sitio, existen restricións.

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)
7839	KING	PRESIDENT		1981-11-17	5000.00		10
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7369	SMITH	CLERK	7902	1980-12-17	800.00		20
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20
7900	JAMES	CLERK	7698	1981-12-03	950.00		30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10

```
select sum(sal)
from emp;
```

sum numeric
29025.00

Neste exemplo ao non haber *where* chega toda a táboa á cláusula *select*

Funcións colectivas

```
select sum(sal)
from emp
where job='CLERK';
```

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)
7369	SMITH	CLERK	7902	1980-12-17	800.00		20
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20
7900	JAMES	CLERK	7698	1981-12-03	950.00		30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10

sum numeric
4150.00

Como aquí hai *where*
o que chega á cláusula
select é o resultado do
where

Primeiro
execútase o
where

E, a seguir, a
cláusula *select*

Funcións colectivas

Formato: **func** (<expr>)

Moitas permiten ALL e DISTINCT: <func> ([ALL | DISTINCT] <expr>)

Se aparece DISTINCT, elimínanse os valores repetidos do argumento, antes de calcular a función.

As máis frecuentes:

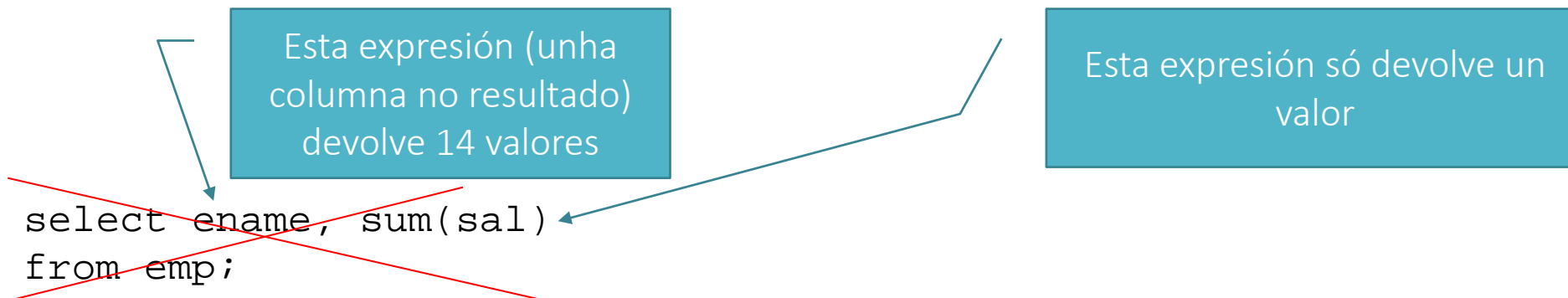
AVG	Media	COUNT	Contar
MAX	Máximo	MIN	Mínimo
SUM	Suma	VAR	Varianza

O estándar indica que a expresión non pode ser unha subconsulta nin unha expresión cunha función colectiva (non se pode aniñar funcións colectivas).

Aínda que algúns SXBD si permiten un nivel de aniñamento.

Funcións colectivas

Se se inclúe unha función de agrupamento na cláusula SELECT todas as expresións na devandita cláusula deben ter un valor único para o conxunto das filas.



Non se pode construír unha táboa resultado cunha columna de 14 valores e outra columna con só un valor. A táboa ten unha *bolsa* de filas, e cada fila ten que ter un valor (ou un nulo) para cada unha das columnas (2 neste exemplo).

O propio SXBD dá un erro:

```
select ename, sum(sal)
      2 from emp;
select ename, sum(sal)
      *
```

ERROR en línea 1:

ORA-00937: la función de grupo no es de grupo único

Funcións colectivas

As funcións colectivas eliminan (case sempre) os nulos antes de realizar a súa operación.

```
select comm, mgr  
from emp;
```

comm numeric(7,2)	mgr numeric(4,0)
	7839
	7566
	7902
	7839
300.00	7698
500.00	7698
1400.00	7698
	7839
	7566
0.00	7698
	7788
	7698
	7782

10
nulos

1 nulo e 13 distintos
de nulo

```
select sum(comm), count(mgr)  
from emp;
```

sum numeric	count bigint
2200.00	13

Suma dos
distintos de
nulo

Conta os
distintos de
nulo

Funcións colectivas

Pódese incluír un *distinct* dentro de cada función colectiva (é o único modo de que poida aparecer máis dun distinct nunha cláusula select).

<func> ([ALL | DISTINCT] <expr>)

```
select deptno  
from emp;
```

deptno numeric(2,0)
10
20
20
20
30
30
30
30
10
20
30
20
30
10

14 valores e 3
distintos (10,20,30)

```
select sum(distinct deptno), count(distinct deptno)  
from emp;
```

sum numeric	count bigint
60	3

Conta (10,20,30)

Suma (10,20,30)

Funcións colectivas

A función `count` ten pequenas diferenzas:

`count ([ALL] <expre>)` conta cantos valores distintos de nulo hai na columna correspondente a `<expre>` na táboa resultante.

`count (DISTINCT <expre>)` conta cantos valores distintos de nulo e distintos entre si hai na columna correspondente a `<expre>` na táboa resultante.

`count (*)` conta cantas filas hai na táboa resultante (aínda cando as filas sexan todo nulos).

```
select *  
from tabla;
```

col1	col2
integer	integer
1	1
1	1
2	2

Dúas filas todo nulos

5 filas, 3 distintos de
nulo e 2 distintos
(1,2)

```
select count(col1), count(distinct col1), count(*)  
from tabla;
```

count	count	count
bigint	bigint	bigint
3	2	5

Exercicios

1. Mostra cantos empregados hai e a canto ascenden os seus ingresos (sumando os de todos e incluíndo salario e comisión) que sexan SALESMAN ou CLERK.
2. Cantos empregados teñen comisión, cantos non teñen comisión, a canto ascende o salario medio, e a canto ascende a comisión media.
3. Empregados cun nome de máis de 5 letras.
4. Cantos empregados traballan para os departamentos 20 e 30, e cantos traballos distintos se desempeñan neses departamentos.
5. Cantos empregados teñen xefe, cantos son xefes e cantos non son xefes.
6. Cantos son os ingresos (salario máis comisión) medios dos empregados contratados despois do 01-08-1981.

Exercicios

1.

```
Select count(*),sum(sal+coalesce(comm,0))  
from emp  
where job in ('SALESMAN','CLERK')
```
2.

```
Select count(comm), count(*)-count(comm),avg(sal), avg(comm)  
from emp
```
3.

```
Select ename  
from emp  
where length(ename)>5
```
4.

```
Select count(*), count(distinct job)  
from emp  
where deptno in (20,30)
```
5.

```
Select count(mgr), count(distinct mgr), count(empno)-count(distinct mgr)  
from emp
```
6.

```
select avg(coalesce(sal+comm,sal))  
from emp  
where hiredate >'01-08-1981'
```

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Group by

Ata agora, as funcións de agrupamento actuaban sobre o conxunto das filas que lle chegan á cláusula select.

En lugar de aplicar as funcións colectivas sobre todas as filas, estas pódense agrupar, formando máis dun grupo de filas, e daquela aplicar as funcións sobre cada un deses grupos.

Eses grupos créanse indicando unha ou máis *columnas de agrupamento*, así os grupos de filas están formados por todas as filas que teñen o mesmo valor nas columnas de agrupamento.

No resultado, haberá unha fila por cada un destes grupos.

```
SELECT ...  
  FROM ...  
  [WHERE ...]  
  [GROUP BY <columna1>[,<columna2>,...]
```

Algúns SXBD (como Oracle) permiten expresións no group by.

Group by

```
SELECT count(*), sum(sal)
FROM emp
GROUP BY job
```

empno numeric(4,0)	ename character varying(10)	sal numeric(7,2)	job character varying(9)	deptno numeric(2,0)
7788	SCOTT	3000.00	ANALYST	20
7902	FORD	3000.00	ANALYST	20
7934	MILLER	1300.00	CLERK	10
7369	SMITH	800.00	CLERK	20
7876	ADAMS	1100.00	CLERK	20
7900	JAMES	950.00	CLERK	30
7782	CLARK	2450.00	MANAGER	10
7566	JONES	2975.00	MANAGER	20
7698	BLAKE	2850.00	MANAGER	30
7839	KING	5000.00	PRESIDENT	10
7844	TURNER	1500.00	SALESMAN	30
7654	MARTIN	1250.00	SALESMAN	30
7521	WARD	1250.00	SALESMAN	30
7499	ALLEN	1600.00	SALESMAN	30

count bigint	sum numeric
2	6000.00
4	4150.00
3	8275.00
1	5000.00
4	5600.00

Por cada grupo de filas co mesmo *job*, xérase *UNHA FILA* no resultado

Group by

```
SELECT job, count(*), sum(sal)
FROM emp
GROUP BY job
```

É posible pór a(s) columna(s) de agrupamento na cláusula select

Pero calquera outra columna **NON** pode aparecer na cláusula select

empno numeric	ename character	sal numeric(7,2)	job character varying(9)	deptno numeric(2,0)
7788	SCOTT	3000.00	ANALYST	20
7902	FORD	3000.00	ANALYST	20
7934	MILLER	1300.00	CLERK	10
7369	SMITH	800.00	CLERK	20
7876	ADAMS	1100.00	CLERK	20
7900	JAMES	950.00	CLERK	30
7782	CLARK	2450.00	MANAGER	10
7566	JONES	2975.00	MANAGER	20
7698	BLAKE	2850.00	MANAGER	30
7839	KING	5000.00	PRESIDENT	10
7844	TURNER	1500.00	SALESMAN	30
7654	MARTIN	1250.00	SALESMAN	30
7521	WARD	1250.00	SALESMAN	30
7499	ALLEN	1600.00	SALESMAN	30

job character varying(9)	count bigint	sum numeric
ANALYST	2	6000.00
CLERK	4	4150.00
MANAGER	3	8275.00
PRESIDENT	1	5000.00
SALESMAN	4	5600.00

Todas as filas do grupo
teñen o mesmo *job*

Por tanto, hai UN
único *job* grupo, que
pode aparecer no
resultado

Group by

```
SELECT job, deptno, count(*), sum(sal)
FROM emp
GROUP BY job
```

É posible pór a(s) columna(s) de agrupamento na cláusula select

Pero calquera outra columna **NON** pode aparecer na cláusula select

Non hai único deptno para este grupo

Nunha fila só pode haber para cada atributo un único valor. Pero temos varios!!!!

empno numeric(4,0)	ename character(10)	sal numeric(7,2)	job character varying(9)	deptno numeric(2,0)
7788	SCOTT	3000.00	ANALYST	20
7902	FORD	3000.00	ANALYST	20
7934	MILLER	1300.00	CLERK	10
7369	SMITH	800.00	CLERK	20
7876	ADAMS	1100.00	CLERK	20
7900	JAMES	950.00	CLERK	30
7782	CLARK	2450.00	MANAGER	10
7566	JONES	2975.00	MANAGER	20
7698	BLAKE	2850.00	MANAGER	30
7839	KING	5000.00	PRESIDENT	10
7844	TURNER	1500.00	SALESMAN	30
7654	MARTIN	1250.00	SALESMAN	30
7521	WARD	1250.00	SALESMAN	30
7499	ALLEN	1600.00	SALESMAN	30

job character varying(9)	count bigint	sum numeric
ANALYST	2	6000.00
CLERK	4	4150.00
MANAGER	3	8275.00
PRESIDENT	1	5000.00
SALESMAN	4	5600.00

```
SQL>
1  SELECT job, deptno, count(*), sum(sal)
2    FROM emp
3    GROUP BY job
4*
SELECT job, deptno, count(*), sum(sal)
      *
ERROR en línea 1:
ORA-00979: no es una expresión GROUP BY

SQL>
```

Group by

```
SELECT job, deptno, count(*), sum(sal)
FROM emp
GROUP BY job, deptno
```

Agora os grupos son formados por filas con igual valor en job e deptno

Por tanto, ao incluír deptno no group by cambian os grupos

empno numeric	ename character	sal numeric(7,2)	job character varying(9)	deptno numeric(2,0)
7788	SCOTT	3000.00	ANALYST	20
7902	FORD	3000.00	ANALYST	20
7934	MILLER	1300.00	CLERK	10
7369	SMITH	800.00	CLERK	20
7876	ADAMS	1100.00	CLERK	20
7900	JAMES	950.00	CLERK	30
7782	CLARK	2450.00	MANAGER	10
7566	JONES	2975.00	MANAGER	20
7698	BLAKE	2850.00	MANAGER	30
7839	KING	5000.00	PRESIDENT	10
7844	TURNER	1500.00	SALESMAN	30
7654	MARTIN	1250.00	SALESMAN	30
7521	WARD	1250.00	SALESMAN	30
7499	ALLEN	1600.00	SALESMAN	30

job character varying(9)	deptno numeric(2,0)	count bigint	sum numeric
ANALYST	20	2	6000.00
CLERK	10	1	1300.00
CLERK	20	2	1900.00
CLERK	30	1	950.00
MANAGER	10	1	2450.00
MANAGER	20	1	2975.00
MANAGER	30	1	2850.00
PRESIDENT	10	1	5000.00
SALESMAN	30	4	5600.00

Group by

```
SELECT comm, count(*)  
FROM emp  
GROUP BY comm
```

Os nulos son “iguais” para o group by

empno numeric	ename character	sal numeric(7,2)	deptno numeric	comm numeric(7,2)
7844	TURNER	1500.00	30	0.00
7499	ALLEN	1600.00	30	300.00
7521	WARD	1250.00	30	500.00
7654	MARTIN	1250.00	30	1400.00
7698	BLAKE	2850.00	30	
7782	CLARK	2450.00	10	
7788	SCOTT	3000.00	20	
7876	ADAMS	1100.00	20	
7900	JAMES	950.00	30	
7839	KING	5000.00	10	
7934	MILLER	1300.00	10	
7566	JONES	2975.00	20	
7902	FORD	3000.00	20	
7369	SMITH	800.00	20	

comm numeric(7,2)	count bigint
0.00	1
300.00	1
500.00	1
1400.00	1
	10

Exercicios

1. Cantos empregados hai en cada departamento, cantos teñen comisión, cantos non teñen comisión e cales son os ingresos medios (incluíndo salario e comisión).
2. Mostra os departamentos que teñen empregados con comisión. Non pode haber valores repetidos.
3. Para cada departamento mostra a comisión media; se non ten empregados con comisión, débese indicar cun 0.
4. Para cada departamento mostra cantos postos de traballo distintos desempeñan os seus traballadores.
5. Para cada departamento mostra cantos empregados hai de cada posto de traballo.
6. Mostra cantos empregados teñen uns ingresos superiores a 2500 € en cada departamento.

Exercicios

1.

```
Select deptno, count(*),count(comm), count(*)-count(comm),avg(coalesce(sal+comm,sal))  
from emp  
group by deptno
```
2.

```
Select distinct deptno  
from emp  
where comm is not null
```
3.

```
Select deptno, coalesce(avg(comm),0)  
from emp  
group by deptno
```
4.

```
Select deptno, count(distinct job)  
from emp  
group by deptno
```
5.

```
Select deptno, job, count(*)  
from emp  
group by deptno, job
```
6.

```
select deptno, count(*)  
from emp  
where coalesce(sal+comm,sal)>2500  
group by deptno
```

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Having

De igual forma que a cláusula *where* permite filtrar *filas*, a cláusula *having* permite filtrar grupos.

A cláusula *having* permite establecer unha condición (predicado) que se avalía sobre cada grupo de filas, e aqueles grupos que fan certa a condición, pasan á cláusula *select*.

```
SELECT job, count(*), sum(sal)
FROM emp
GROUP BY job
HAVING min(sal) > 1500
```

empno numeric(4,0)	ename character varying(9)	sal numeric(7,2)	job character varying(9)	deptno numeric(2,0)
7788	SCOTT	3000.00	ANALYST	20
7902	FORD	3000.00	ANALYST	20
7934	MILLER	1300.00	CLERK	10
7369	SMITH	800.00	CLERK	20
7876	ADAMS	1100.00	CLERK	20
7900	JAMES	950.00	CLERK	30
7782	CLARK	2450.00	MANAGER	10
7566	JONES	2975.00	MANAGER	20
7698	BLAKE	2850.00	MANAGER	30
7839	KING	5000.00	PRESIDENT	10
7844	TURNER	1500.00	SALESMAN	30
7654	MARTIN	1250.00	SALESMAN	30
7521	WARD	1250.00	SALESMAN	30
7499	ALLEN	1600.00	SALESMAN	30

Certo

Falso

Certo

Certo

Falso

job character varying(9)	count bigint	sum numeric
ANALYST	2	6000.00
MANAGER	3	8275.00
PRESIDENT	1	5000.00

Having

No predicado dun having pódese utilizar todas as ferramentas usadas nos predicados where: todos os predicados, subconsultas etc.

Pero hai que ter tino coas expresións que se inclúan: estas só poden conter constantes, funcións (incluíndo as colectivas) aplicadas sobre calquera columna, pero, fóra de funcións colectivas, só poden aparecer as *columnas de agrupamento*.

```
SELECT job, count(*), sum(sal)
FROM emp
GROUP BY job
HAVING sal > 1500
```

empno numeric(4)	ename character varying(10)	sal numeric(7,2)	job character varying(9)	deptno numeric(2,0)
7788	SCOTT	3000.00	ANALYST	20
7902	FORD	3000.00	ANALYST	20
7934	MILLER	1300.00	CLERK	10
7369	SMITH	800.00	CLERK	20
7876	ADAMS	1100.00	CLERK	20
7900	JAMES	950.00	CLERK	30
7782	CLARK	2450.00	MANAGER	10
7566	JONES	2975.00	MANAGER	20
7698	BLAKE	2850.00	MANAGER	30
7839	KING	5000.00	PRESIDENT	10
7844	TURNER	1500.00	SALESMAN	30
7654	MARTIN	1250.00	SALESMAN	30
7521	WARD	1250.00	SALESMAN	30
7499	ALLEN	1600.00	SALESMAN	30

→ ¿sal?>1500

O predicado aplícase UN A UN a cada *UN dos grupos de filas*.

Por tanto, cando se aplique sobre este grupo, por que valor substituímos o nome de columna **SAL**

Having

Pero hai que ter tino, tamén, coas expresións que se inclúan: estas só poden conter constantes, funcións (incluíndo as colectivas) aplicadas sobre calquera columna, pero, fóra de funcións colectivas, só poden aparecer as *columnas de agrupamento*.

```
SELECT job, count(*), sum(sal)
FROM emp
GROUP BY job
HAVING min(sal) > 1500
```

empno numeric(4)	ename character varying(10)	sal numeric(7,2)	job character varying(9)	deptno numeric(2,0)
7788	SCOTT	3000.00	ANALYST	20
7902	FORD	3000.00	ANALYST	20
7934	MILLER	1300.00	CLERK	10
7369	SMITH	800.00	CLERK	20
7876	ADAMS	1100.00	CLERK	20
7900	JAMES	950.00	CLERK	30
7782	CLARK	2450.00	MANAGER	10
7566	JONES	2975.00	MANAGER	20
7698	BLAKE	2850.00	MANAGER	30
7839	KING	5000.00	PRESIDENT	10
7844	TURNER	1500.00	SALESMAN	30
7654	MARTIN	1250.00	SALESMAN	30
7521	WARD	1250.00	SALESMAN	30
7499	ALLEN	1600.00	SALESMAN	30

→ 3000>1500?

→ 800>1500?

→ 2450>1500?

→ 5000>1500?

→ 1250>1500?

O predicado aplícase UN A UN a cada un dos grupos de filas.

Having

Orde de execución na sentenza select:

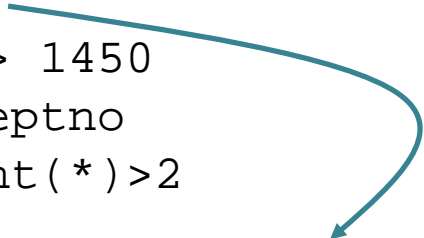
1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY

Having

A condición *where* aplícase a *filas*

A condición *having* aplícase a *grupos de filas*

```
SELECT deptno, count(*), sum(sal)
FROM emp
WHERE sal > 1450
GROUP BY deptno
HAVING count(*) > 2
```



Primeiro paso

empno numeric(4)	ename character	job character var	mgr numeric	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric
7369	SMITH	CLERK	7902	1980-12-17	800.00		20
7900	JAMES	CLERK	7698	1981-12-03	950.00		30
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7839	KING	PRESIDENT		1981-11-17	5000.00		10

Having

A condición *where* aplícase a *filas*

A condición *having* aplícase a *grupos de filas*

```
SELECT deptno, count(*), sum(sal)
FROM emp
WHERE sal > 1450
GROUP BY deptno
HAVING count(*) > 2
```

Segundo paso

empno	ename	job	mgr	hiredate	sal	comm	deptno
numeric(4)	character	character var	numeric	date	numeric(7,2)	numeric(7,2)	numeric
7369	SMITH	CLERK	7902	1980-12-17	800.00		20
7900	JAMES	CLERK	7698	1981-12-03	950.00		30
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7839	KING	PRESIDENT		1981-11-17	5000.00		10

Falso

Certo

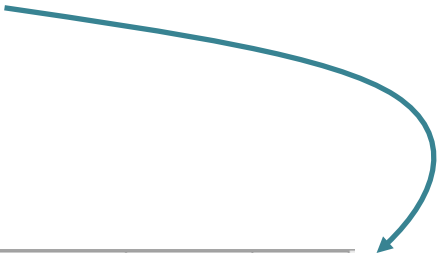
empno	ename	job	mgr	hiredate	sal	comm	deptno
numeric(4)	character	character var	numeric	date	numeric(7,2)	numeric(7,2)	numeric
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7839	KING	PRESIDENT		1981-11-17	5000.00		10
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30

Having

A condição *where* aplícase a *filas*

A condição *having* aplícase a *grupos de filas*

```
SELECT deptno, count(*), sum(sal)
FROM emp
WHERE sal > 1450
GROUP BY deptno
HAVING count(*) > 2
```



Terceiro passo

empno numeric(4)	ename character	job character var	mgr numeric	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7839	KING	PRESIDENT		1981-11-17	5000.00		10
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30

Having

A condición *where* aplícase a *filas*

A condición *having* aplícase a *grupos de filas*

```
SELECT deptno, count(*), sum(sal)
FROM emp
WHERE sal > 1450
GROUP BY deptno
HAVING count(*) > 2
```

Cuarto paso

empno numeric(4)	ename character	job character var	mgr numeric	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7839	KING	PRESIDENT		1981-11-17	5000.00		10
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30

Falso

Certo

Certo

empno numeric(4)	ename character	job character var	mgr numeric	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30

Having

A condición *where* aplícase a *filas*

A condición *having* aplícase a *grupos de filas*

```
SELECT deptno, count(*), sum(sal)
```

```
FROM emp
```

```
WHERE sal > 1450
```

```
GROUP BY deptno
```

```
HAVING count(*) > 2
```

Quinto paso

empno numeric(4)	ename character	job character var	mgr numeric	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30

deptno numeric(2,0)	count bigint	sum numeric
20	3	8975.00
30	3	5950.00

Exercicios

1. Para cada departamento mostra cantos empregados teñen uns ingresos (sal+comm) superiores a 2500 €.
2. Mostra os departamentos cuns ingresos medios superiores aos 2500 €. Mostra para cada un, cantos empregados teñen.
3. Departamentos con, polo menos, dous 'MANAGER'
4. Departamentos con, polo menos, dous empregados con comisión. Para cada departamento mostra cantos empregados ten (en total) e cantos con comisión.
5. Departamentos con, polo menos, dous empregados co mesmo posto de traballo. Non poden aparecer repetidos.

Exercicios

1. `Select deptno, count(*)
from emp
where coalesce(sal+comm,sal)>2500
group by deptno`

2. `Select deptno, count(*)
from emp
group by deptno
having avg(coalesce(sal+comm,sal))>2500`

3. `Select deptno
from emp
where job = 'MANAGER'
group by deptno
having count(*)>=2`

4. `Select deptno, count(*), count(comm)
from emp
group by deptno
having count(comm)>=2`

5. `Select distinct deptno
from emp
group by deptno, job
having count(*)>=2`

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Máis dunha táboa no FROM

Como podo obter unha táboa que para cada *empregado* me mostre o seu *nome* e o *nome do departamento* para o que traballa.

Na cláusula `select` os nomes de columna que aparezan deben estar nalgunha táboa no `from`.

Pero na táboa `emp` só temos o número de departamento.

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30
7566	JONES	MANAGER	7839	02/04/81	2,975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20
7839	KING	PRESIDENT		17/11/81	5,000		10
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1,100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3,000		20
7934	MILLER	CLERK	7782	23/01/82	1,300		10

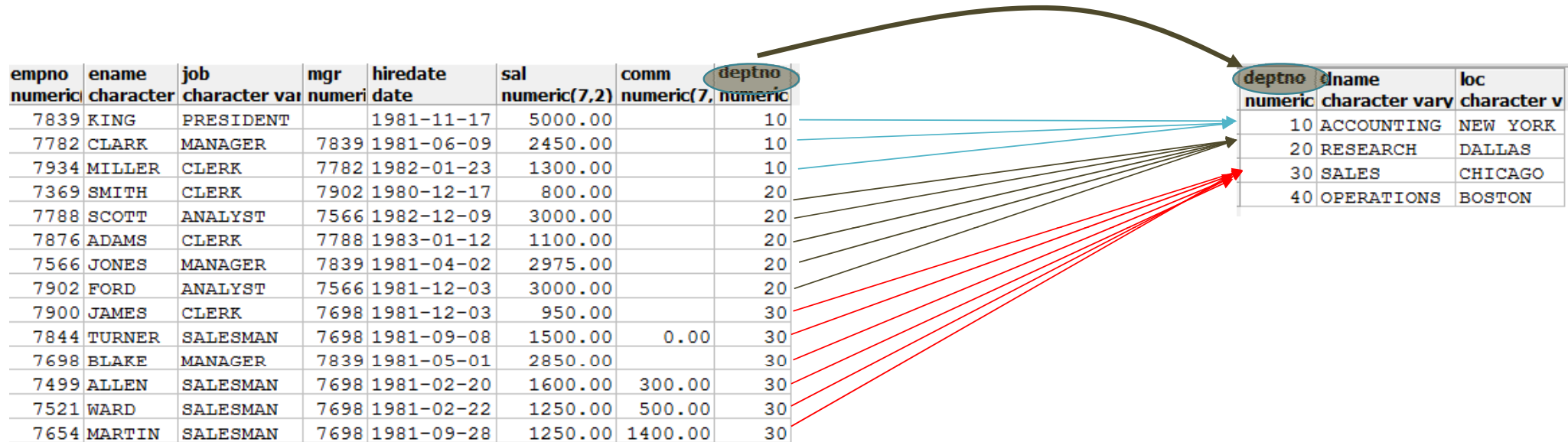
```
SQL> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Necesitamos acceder á táboa `emp` e á táboa `dept`

Máis dunha táboa no FROM

Cando partimos a información en táboas, deixamos claves foráneas que manteñen os vínculos entre a información partida.



Mediante esas claves foráneas, podemos enlazar a información partida.

O mecanismo que se usa en SQL chámase JOIN ('reunir, xuntar' en galego).

JOIN

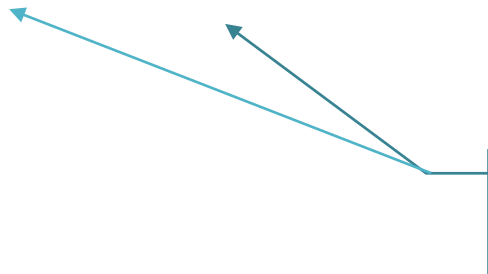
```
SELECT ...  
FROM <tabla1> [INNER|LEFT|RIGHT|FULL] JOIN <tabla2>  
ON <condición de join>
```

O máis normal é este: que obtén, do produto cartesiano, as filas que fan certa a condición de join

```
SELECT ...  
FROM <tabla1> [INNER] JOIN <tabla2>  
ON <condición de join>
```

E o máis normal é que a condición de join sexa unha igualdade entre unha clave foránea e unha clave primaria. Por ex.:

```
SELECT *  
FROM emp JOIN dept  
ON emp.deptno=dept.deptno
```



Hai 2 columnas deptno, así que hai que desambiguar

JOIN

Conceptualmente (non o fai en realidade), primeiro crea o produto cartesiano

Todas as
columnas
das dúas táboas

Todas as filas de
emp pegadas a
todas as de
dept

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)	deptno numeric(2,0)	dname character varying(14)	loc character varying(13)
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-09-01	2975.00		20	10	ACCOUNTING	NEW YORK
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	10	ACCOUNTING	NEW YORK
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	1981-09-08	1600.00	300.00	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	1981-02-12	1250.00	500.00	30	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7698	1981-09-08	1250.00	1400.00	30	10	ACCOUNTING	NEW YORK
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	10	ACCOUNTING	NEW YORK
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	10	ACCOUNTING	NEW YORK
7876	ADAMS	CLERK	7788	1983-01-23	1100.00		20	10	ACCOUNTING	NEW YORK
7900	JAMES	CLERK	7698	1981-12-03	950.00		30	10	ACCOUNTING	NEW YORK
7934	MILLER	CLERK	7782	1982-07-06	1300.00		10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT		1981-11-17	5000.00		10	20	RESEARCH	DALLAS
7566	JONES	MANAGER	7839	1981-09-01	2975.00		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	7698	1981-09-08	1600.00	300.00	30	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7698	1981-02-12	1250.00	500.00	30	20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7698	1981-09-08	1250.00	1400.00	30	20	RESEARCH	DALLAS
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	20	RESEARCH	DALLAS

... continúa con máis filas...

JOIN

```
SELECT *  
FROM emp JOIN dept  
ON emp.deptno=dept.deptno
```

EMP							DEPT			
empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)	deptno numeric(2,0)	dname character varying(14)	loc character varying(13)
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-09-08	2975.00		20	10	ACCOUNTING	NEW YORK
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	10	ACCOUNTING	NEW YORK
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	1981-03-17	1600.00	300.00	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	1981-05-12	1250.00	500.00	30	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7698	1981-09-08	1250.00	1400.00	30	10	ACCOUNTING	NEW YORK
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-07-09	3000.00		20	10	ACCOUNTING	NEW YORK

Sobre o produto cartesiano
selecciónanse as filas
que fan certa a condición
de join

Este tipo de filas
son as que fan
certa a condición
de join

Estas filas non saen
porque non fan certa
a condición de join

JOIN

```
SELECT *  
FROM emp JOIN dept  
ON emp.deptno=dept.deptno
```

EMP							DEPT			
empno numeric	ename character	job character vari	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric	deptno numeric	dname character vari	loc character vari
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	1981-12-03	950.00		30	30	SALES	CHICAGO
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	10	ACCOUNTING	NEW YORK

Pódesse desambiguar usando um alcume.

```
SELECT ename, E.deptno, dname  
FROM emp E JOIN dept D  
ON E.deptno=D.deptno
```

Só é necesario desambiguar nos nomes de columna que aparecen nas dúas táboas.

JOIN

Lembrar os pasos

1. FROM (obrigatoria)

Partindo dunha ou máis táboas *obtén unha única táboa* que será procesada polo resto de cláusulas

2. WHERE (optativa)

3. GROUP BY (optativa)

4. HAVING (optativa)

3. SELECT (obrigatoria)


4. ORDER BY (optativa)

Así pois, todo o visto ata agora funciona igual porque o FROM é o primeiro que se executa, e devolve unha única táboa.

JOIN

```
SELECT ename, E.deptno, dname  
FROM emp E JOIN dept D  
ON E.deptno=D.deptno  
WHERE coalesce(comm,0)+sal>2500
```

Primeiro passo



empno numeric	ename character	job character var	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric	deptno numeric	dname character var	loc character var
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	1981-12-03	950.00		30	30	SALES	CHICAGO
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	10	ACCOUNTING	NEW YORK

JOIN

```
SELECT ename, E.deptno, dname
FROM emp E JOIN dept D
ON E.deptno=D.deptno
WHERE coalesce(comm,0)+sal>2500
```

empno	ename	job	mgr	hiredate	sal	comm	deptno	deptno	dname	loc
numeric	character	character var	numeric(4,0)	date	numeric(7,2)	numeric(7,	numeric(numeric(character var	character v
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	1981-12-03	950.00		30	30	SALES	CHICAGO
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	10	ACCOUNTING	NEW YORK

Segundo paso

empno	ename	job	mgr	hiredate	sal	comm	deptno	deptno	dname	loc
numeric	character	character var	numeric(4,0)	date	numeric(7,2)	numeric(7,	numeric(numeric(character var	character v
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	30	SALES	CHICAGO
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	20	RESEARCH	DALLAS

Terceiro passo

ename	deptno	dname
character varying(10)	numeric(2,0)	character varying(14)
KING	10	ACCOUNTING
JONES	20	RESEARCH
FORD	20	RESEARCH
BLAKE	30	SALES
MARTIN	30	SALES
SCOTT	20	RESEARCH

JOIN

Pódese usar máis dunha copia dunha mesma táboa.

```
SELECT s.ename subordinado, s.mgr, j.empno, j.ename jefe  
FROM emp s JOIN emp j ON s.mgr=j.empno
```

subordinado character varying(10)	mgr numeric(4,0)	empno numeric(4,0)	jefe character varying(10)
JONES	7839	7839	KING
FORD	7566	7566	JONES
SMITH	7902	7902	FORD
BLAKE	7839	7839	KING
ALLEN	7698	7698	BLAKE
WARD	7698	7698	BLAKE
MARTIN	7698	7698	BLAKE
CLARK	7839	7839	KING
SCOTT	7566	7566	JONES
TURNER	7698	7698	BLAKE
ADAMS	7788	7788	SCOTT
JAMES	7698	7698	BLAKE
MILLER	7782	7782	CLARK

JOIN

A condición de join pode ser calquera predicado.

```
SELECT a.ename, a.sal, b.ename, b.sal
FROM emp a JOIN emp b ON a.sal>b.sal
```

ename character varying(10)	sal numeric(7,2)	ename character varying(10)	sal numeric(7,2)
KING	5000.00	JONES	2975.00
KING	5000.00	FORD	3000.00
KING	5000.00	SMITH	800.00
KING	5000.00	BLAKE	2850.00
KING	5000.00	ALLEN	1600.00
KING	5000.00	WARD	1250.00
KING	5000.00	MARTIN	1250.00
KING	5000.00	CLARK	2450.00
KING	5000.00	SCOTT	3000.00
KING	5000.00	TURNER	1500.00
KING	5000.00	ADAMS	1100.00
KING	5000.00	JAMES	950.00
KING	5000.00	MILLER	1300.00
JONES	2975.00	SMITH	800.00
JONES	2975.00	BLAKE	2850.00
JONES	2975.00	ALLEN	1600.00
JONES	2975.00	WARD	1250.00
JONES	2975.00	MARTIN	1250.00
JONES	2975.00	CLARK	2450.00
JONES	2975.00	TURNER	1500.00
JONES	2975.00	ADAMS	1100.00
JONES	2975.00	JAMES	950.00
JONES	2975.00	MILLER	1300.00
FORD	3000.00	JONES	2975.00
FORD	3000.00	SMITH	800.00
FORD	3000.00	BLAKE	2850.00

...

```
SELECT s.ename sub, s.sal, j.ename jefe, j.sal
FROM emp s JOIN emp j
ON s.mgr=j.empno AND s.sal>j.sal
```

sub character varying(10)	sal numeric(7,2)	jefe character varying(10)	sal numeric(7,2)
FORD	3000.00	JONES	2975.00
SCOTT	3000.00	JONES	2975.00

JOIN

Recalcamos que no resultado están as filas (do produto cartesiano) que cumpren a condición de join.

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SELECT *  
FROM emp JOIN dept  
ON emp.deptno=dept.deptno
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30
7566	JONES	MANAGER	7839	02/04/81	2,975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30
7698	BLAKE	MANAGER	7839	01/05/81	2,850		30
7782	CLARK	MANAGER	7839	09/06/81	2,450		10
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20
7839	KING	PRESIDENT		17/11/81	5,000		10
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1,100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3,000		20
7934	MILLER	CLERK	7782	23/01/82	1,300		10

empno	ename	job	mgr	hiredate	sal	comm	deptno	deptno	dname	loc
numeric	character	character var	numeric(4,0)	date	numeric(7,2)	numeric(7,2)	numeric(2)	numeric(2)	character var	character var
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	1981-12-03	950.00		30	30	SALES	CHICAGO
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	10	ACCOUNTING	NEW YORK

No resultado non hai pegada do departamento 40, porque non hai ningún empregado que traballe para ese departamento e, por tanto, nunca se fai certa a condición de join.

JOIN exterior

```
SELECT ...  
FROM <tabla1> [LEFT|RIGHT|FULL] JOIN <tabla2>  
ON <condición de join>
```

Podemos forzar a que as filas dunha (ou das dúas) táboa de entrada que no INNER join non aparecen por non facer certa a condición de join, saian enchendo as columnas do outro lado con nulos.

```
SELECT *
```

```
FROM emp RIGHT JOIN dept  
ON emp.deptno=dept.deptno
```

Imos forzar que saian as filas do lado dereito que non saen no INNER

empno numeric(4,0)	ename character	job character var	mgr numeri	hiredate date	sal numeric(7,2)	comm numeric(7)	deptno numeric	deptno numeric	dname character vary	loc character var
7839	KING	PRESIDENT		1981-11-17	5000.00		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	1980-12-17	800.00		20	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	1981-12-03	950.00		30	30	SALES	CHICAGO
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	10	ACCOUNTING	NEW YORK
								40	OPERATIONS	BOSTON

Nulos

As filas do lado dereito que “emparellan”, saen igual que no INNER join

As filas do lado dereito que “non emparellan”, saen cubrindo as columnas do lado esquerdo con nulos

JOIN exterior

```
SELECT s.ename subordinado, s.mgr, j.empno, j.ename jefe
FROM emp s LEFT JOIN emp j ON s.mgr=j.empno
```

subordinado character varying(10)	mgr numeric(4,0)	empno numeric(4,0)	jefe character varying(10)
KING			
JONES	7839	7839	KING
FORD	7566	7566	JONES
SMITH	7902	7902	FORD
BLAKE	7839	7839	KING
ALLEN	7698	7698	BLAKE
WARD	7698	7698	BLAKE
MARTIN	7698	7698	BLAKE
CLARK	7839	7839	KING
SCOTT	7566	7566	JONES
TURNER	7698	7698	BLAKE
ADAMS	7788	7788	SCOTT
JAMES	7698	7698	BLAKE
MILLER	7782	7782	CLARK

} Forzada pelo left join

} Encaixe "normal"

JOIN exterior

```
SELECT s.ename subordinado, s.mgr, j.empno, j.ename jefe
FROM emp s FULL JOIN emp j ON s.mgr=j.empno
```

subordinado character varying(10)	mgr numeric(4,0)	empno numeric(4,0)	jefe character varying(10)
KING			
JONES	7839	7839	KING
FORD	7566	7566	JONES
SMITH	7902	7902	FORD
BLAKE	7839	7839	KING
ALLEN	7698	7698	BLAKE
WARD	7698	7698	BLAKE
MARTIN	7698	7698	BLAKE
CLARK	7839	7839	KING
SCOTT	7566	7566	JONES
TURNER	7698	7698	BLAKE
ADAMS	7788	7788	SCOTT
JAMES	7698	7698	BLAKE
MILLER	7782	7782	CLARK
		7844	TURNER
		7369	SMITH
		7876	ADAMS
		7934	MILLER
		7521	WARD
		7900	JAMES
		7499	ALLEN
		7654	MARTIN

} Forzada polo left join

} Encaixe "normal"

} Forzada polo right join

Join de máis de dúas táboas

```
SELECT ...  
FROM <tabla1> [INNER|LEFT|RIGHT|FULL] JOIN <tabla2> ON <condición de join12>  
          [INNER|LEFT|RIGHT|FULL] JOIN <tabla3> ON <condición de join123>  
          [INNER|LEFT|RIGHT|FULL] JOIN <tabla4> ON <condición de join1234>  
          ...
```

En primeiro lugar, únense <tabla1> e <tabla2>, coa súa condición de join.

Isto dá lugar a unha táboa.

Esa táboa resultante únese á <tabla3>, coa súa condición de join, o que resulta noutra táboa.

E así sucesivamente.

Join de máis de dúas táboas

```
SELECT ...  
FROM <tabla1> JOIN <tabla2> JOIN <tabla3> JOIN <tabla4>  
ON <condición de join12>  
  and <condición de join123>  
  and <condición de join1234>  
  ...
```

Isto **NON É CORRECTO!!**

Join de máis de dúas táboas

SELECT ename, pname, hours

FROM emp e JOIN emppro ep ON e.empno=ep.empno
JOIN pro p ON ep.prono=p.prono

Primeiro paso

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)
7839	KING	PRESIDENT		1981-11-17	5000.00		10
7566	JONES	MANAGER	7839	1981-04-02	2975.00		20
7902	FORD	ANALYST	7566	1981-12-03	3000.00		20
7369	SMITH	CLERK	7902	1980-12-17	800.00		20
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00		30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00		10
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00		20
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1983-01-12	1100.00		20
7900	JAMES	CLERK	7698	1981-12-03	950.00		30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10

empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7499	1004	15
7499	1005	12
7521	1004	10
7521	1008	8
7654	1001	16
7654	1006	15
7654	1008	5
7844	1005	6
7934	1001	4

emp

emppro

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)	empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1004	15
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1005	12
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1004	10
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1008	8
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1001	16
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1006	15
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1008	5
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	7844	1005	6
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	7934	1001	4

Join de máis de dúas táboas

```
SELECT ename, pname, hours  
FROM emp e JOIN emppro ep ON e.empno=ep.empno  
      JOIN pro p ON ep.prono=p.prono
```

Segundo paso

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)	empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1004	15
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1005	12
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1004	10
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1008	8
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1001	16
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1006	15
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1008	5
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	7844	1005	6
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	7934	1001	4

prono numeric	pname character	loc character varying	deptno numeric
1001	P1	BOSTON	20
1004	P4	CHICAGO	30
1005	P5	CHICAGO	30
1006	P6	LOS ANGELES	30
1008	P8	NEW YORK	30

emp

emppro

pro

empno numeric(4,0)	ename character varying(10)	job character varying(9)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)	empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)	prono numeric(4,0)	pname character varying(10)	loc character varying(13)	deptno numeric(2,0)
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1004	15	1004	P4	CHICAGO	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1005	12	1005	P5	CHICAGO	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1004	10	1004	P4	CHICAGO	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1008	8	1008	P8	NEW YORK	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1001	16	1001	P1	BOSTON	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1006	15	1006	P6	LOS ANGELES	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1008	5	1008	P8	NEW YORK	30
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	7844	1005	6	1005	P5	CHICAGO	30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	7934	1001	4	1001	P1	BOSTON	20

Join de máis de dúas táboas

```
SELECT ename, pname, hours
FROM emp e JOIN emppro ep ON e.empno=ep.empno
      JOIN pro p ON ep.prono=p.prono
```

empno numeric(4,0)	ename character varying(10)	job character varying(10)	mgr numeric(4,0)	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric(2,0)	empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)	prono numeric(4,0)	pname character varying(10)	loc character varying(10)	deptno numeric(2,0)
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1004	15	1004	P4	CHICAGO	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	7499	1005	12	1005	P5	CHICAGO	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1004	10	1004	P4	CHICAGO	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	7521	1008	8	1008	P8	NEW YORK	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1001	16	1001	P1	BOSTON	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1006	15	1006	P6	LOS ANGELES	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	7654	1008	5	1008	P8	NEW YORK	30
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30	7844	1005	6	1005	P5	CHICAGO	30
7934	MILLER	CLERK	7782	1982-01-23	1300.00		10	7934	1001	4	1001	P1	BOSTON	20

ename character varying(10)	pname character varying(10)	hours numeric(2,0)
ALLEN	P4	15
ALLEN	P5	12
WARD	P4	10
WARD	P8	8
MARTIN	P1	16
MARTIN	P6	15
MARTIN	P8	5
TURNER	P5	6
MILLER	P1	4

Exercicios

1. Para cada proxecto mostra o seu nome e o nome do departamento que os controla.
2. Para cada empregado mostra o seu nome e os códigos de proxectos para os que traballa.
3. Para cada empregado mostra o seu nome e os códigos de proxectos para os que traballa. Se hai empregados que non traballan en proxectos, estes deben aparecer co código de proxecto a nulo.
4. Para cada empregado mostra o nome do seu xefe; se non ten xefe, mostra un nulo no nome do xefe.
5. Para cada empregado mostra o seu nome, o nome do seu xefe e o departamento para o que traballa o seu xefe.
6. Devolve os empregados que teñen un salario máis alto que o seu xefe.

Exercicios

1. `Select pname, dname
from pro p join dept d on p.deptno=d.deptno`
2. `Select ename, proname
from emp e join emppro ep on e.empno=ep.empno`
3. `Select ename, proname
from emp e left join emppro ep on e.empno=ep.empno`
4. `Select e.ename, j.ename
from emp e left join emp j on e.mgr=j.empno`
5. `Select e.ename, j.ename, d.dname
from emp e join emp j on e.mgr=j.empno join dept d
on j.deptno=d.deptno`
6. `Select e.ename, e.sal, j.ename, j.sal
from emp e join emp j on e.mgr=j.empno
where e.sal>j.sal`

Exercicios

1. Para cada empregado mostra o seu nome e cantas horas traballou en proxectos.
2. Para cada departamento, mostra o seu nome e cantos empregados ten.
3. Para cada xefe, mostra o seu nome e cantos subordinados ten.
4. Mostra o nome de proxectos onde se traballou (en total, todos os empregados) máis de 15 horas
5. Mostra os departamentos (nome) que controlan máis de dous proxectos.
6. Mostra os departamentos (nome) onde hai polo menos dous empregados co mesmo posto de traballo. Non deben aparecer repetidos.
7. Para cada departamento mostra o seu nome e cantos empregados ten; se non ten ningún, hai que indicalo cun 0.
8. Para cada empregado mostra as horas que traballou en proxectos; se non traballou en ningún, hai que indicalo cun 0.
9. Para cada xefe, cantos subordinados gañan máis ca el; se non o gaña ningún, hai que indicalo cun 0.

Exercicios

1.

```
Select ename, sum(hours)
from emp e join emppro ep on e.empno=ep.empno
group by e.empno, ename
```
2.

```
Select dname, count(ename)
from emp e join dept d on e.deptno=d.deptno
group by d.deptno, dname
```
3.

```
Select j.ename, count(e.empno)
from emp e join emp j on e.mgr=j.empno
group by j.empno, j.ename
```
4.

```
Select pname, sum(hours)
from emppro ep join pro p on ep.prono=p.prono
group by p.prono, pname
having sum(hours)>15
```
5.

```
Select dname, count(prono)
from dept d join pro p on d.deptno=p.deptno
group by d.deptno, dname
having count(*)>2
```
6.

```
Select distinct dname
from emp e join dept d on e.deptno=d.deptno
group by d.deptno, dname, job
having count(*)>=2
```
7.

```
Select dname, count(empno) /*no count(*)*/
from emp e right join dept d on e.deptno=d.deptno
group by d.deptno, dname
```
8.

```
Select ename, coalesce(sum(hours),0)
from emp e left join emppro ep on e.empno=ep.empno
group by e.empno, ename
```
9.

```
Select j.ename, count(e.ename)
from emp e right join emp j on e.mgr=j.empno
and e.sal>j.sal
group by j.empno, j.ename
```

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Subconsultas

Ata agora, todos os nomes de columnas que aparecen en todas as expresións deben ser das táboas que aparecen no `FROM` da sentenza `select`.

Se quixésemos saber os empregados que traballan en 'RESEARCH' (sen `join`) teríamos un problema, porque necesitamos a táboa `EMP` para obter os datos dos empregados.

```
SELECT *  
FROM emp  
WHERE deptno=20
```

Pero necesitamos saber que o departamento denominado `RESEARCH` é o que ten o número 20.

Para isto necesitamos executar unha consulta previa.

```
SELECT deptno  
FROM dept  
WHERE dname='RESEARCH'
```

Necesitamos un mecanismo que permita consultar o número de departamento denominado `RESEARCH` na mesma consulta, sen que o usuario teña que facer unha consulta previa.

Subconsultas

SQL permítenos incluír consultas dentro doutras consultas.

Así, en lugar de escribir o 20, podemos incluír unha consulta que obteña o devandito valor.

```
SELECT *  
FROM emp  
WHERE deptno=(SELECT deptno  
               FROM dept  
               WHERE dname='RESEARCH' )
```

Isto denomínanse *subconsultas*.

Unha consulta pode conter, á súa vez, subconsultas.

A consulta inicial coñécese como consulta principal.

O resultado da consulta utilízao a consulta de nivel superior nun predicado, pero o seu resultado non se pode trasladar ao resultado da consulta.

É dicir, no noso exemplo, non podemos pór ningunha expresión na cláusula SELECT principal contendo nomes de columnas da táboa DEPT.


Subconsultas

A subconsulta execútase (conceptualmente) antes de comezar a execución da principal, e o seu resultado úsase para avaliar o predicado onde está.

O primeiro caso que vimos é unha *subconsulta escalar*, que é un tipo especial de expresión, xa que devolve un único valor escalar.

```
SELECT *  
FROM emp  
WHERE deptno=(
```

```
  SELECT deptno  
  FROM dept  
  WHERE dname='RESEARCH' )
```



deptno
numeric(2,0)
20

Dito doutro xeito, a subconsulta devolve unha táboa cunha fila e unha columna. Por tanto, é equivalente a escribir unha expresión literal (unha constante).

Este é o único tipo de subconsulta válida con predicados elementais.

```
SELECT *  
FROM emp  
WHERE deptno = (SELECT deptno  
                FROM dept  
                WHERE dname IN ('SALES', 'RESEARCH'))
```

ERROR en línea 3:
ORA-01427: la subconsulta
de una sola fila devuelve
más de una fila

Subconsultas

A subconsulta pode usar todas as ferramentas usadas ata o momento. Por exemplo:

```
SELECT *  
FROM emp  
WHERE sal > (SELECT AVG(sal)  
              FROM emp  
              WHERE deptno=10)
```

Se usamos un predicado IN, a subconsulta pode devolver máis dunha fila. En tal caso, a subconsulta xa non se considera unha expresión.

A subconsulta debe devolver unha táboa de só unha columna.

```
SELECT *  
FROM emp  
WHERE deptno IN(SELECT deptno  
                 FROM dept  
                 WHERE loc IN ( 'DALLAS' , 'NEW YORK' ) )
```

Subconsultas

É posible que a subconsulta devolva máis dunha fila usando predicados elementais se os operadores están *cuantificados*.

Os operadores **ANY** ou **SOME** e **ALL** modifican o operador para que se poida comparar un valor escalar cunha listaxe de valores (unha táboa dunha soa columna)

```
... <expre> operador_comparacion { ALL | SOME | ANY } (subconsulta)
```

Operadores de comparacion: < <= = != <> >= >

ANY, SOME: a expresión compárase con cada un dos valores da subconsulta e se para algún é verdadeira, o resultado é verdadeiro.

ALL: a expresión compárase con cada un dos valores da subconsulta e se para todos é verdadeira, o resultado é verdadeiro.

```
SELECT *  
FROM emp  
WHERE sal > ANY(SELECT sal  
                FROM emp  
                WHERE deptno=10)
```

```
SELECT *  
FROM emp  
WHERE sal > ALL(SELECT sal  
                FROM emp  
                WHERE deptno=10)
```

Exercicios

1. Empleados que teñen un salario maior ao salario medio da empresa.
2. Para cada departamento, mostra cantos empregados ten que gañen máis do salario medio da empresa. Mostra o nome do departamento.
3. Empleados que son xefe. Mostra o seu nome.
4. Empleados que non son xefe. Mostra o seu nome.
5. Mostra o(s) empregado(s) (nome) co salario máis alto.
6. Mostra o departamento (nome) coa suma de salarios máis alta.
7. Para os departamentos que teñen empregados con comisión, mostra cantos empregados teñen comisión, e cantos non. Mostra o nome do departamento.

Exercicios

1.

```
Select empno, ename, sal
from emp
where sal > (Select avg(sal) from emp)
```
2.

```
Select dname, count(*)
from emp e join dept d on e.deptno=d.deptno
where sal > (Select avg(sal) from emp)
group by d.deptno, dname
```
3.

```
Select ename
from emp
where empno in (Select mgr from emp)
```
4.

```
Select ename
from emp
where empno not in (Select mgr from emp
                    where mgr is not null)
```
5.

```
Select ename
from emp
where sal= (Select max(sal) from emp)
```
6.

```
Select dname, sum(sal)
from emp e join dept d on e.deptno=d.deptno
group by d.deptno, dname
having sum(sal) >= ALL (select sum(sal)
                       from emp
                       group by deptno)
```
7.

```
Select dname, count(*)-count(comm) "Sin comisión", count(comm) "Con comisión"
from emp e join dept d on e.deptno=d.deptno
where d.deptno in (Select deptno
                  from emp
                  where comm is not null)
group by d.deptno, dname
```

Subconsultas correlacionadas

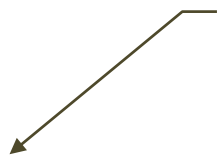
As subconsultas vistas ata agora executábanse unha única vez antes de executar a principal, e o seu resultado utilizábo a consulta principal como un literal ou listaxe de literais.

É dicir, a subconsulta era totalmente independente da principal.

As subconsultas correlacionadas varían con respecto ás normais en:

- Teñen polo menos unha referencia a unha das columnas das táboas no FROM da consulta principal.
- A subconsulta execútase unha vez por cada fila da principal.

```
SELECT *  
FROM emppro a  
WHERE hours = (SELECT MAX(hours)  
                FROM   emppro  
                WHERE  prono=a.prono)
```



Referencia á
columna da
consulta principal

Subconsultas correlacionadas

```
SELECT *
```

```
FROM emppro a
```

```
WHERE hours = (SELECT MAX(hours)  
               FROM emppro  
               WHERE prono=a.prono)
```

Para cada proxecto, cal é o empregado que máis horas traballa?

empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7934	1001	4
7654	1001	16
7499	1004	15
7521	1004	10
7844	1005	6
7499	1005	12
7654	1006	15
7654	1008	5
7521	1008	8

```
SELECT MAX(hours)  
FROM emppro  
WHERE prono=1001
```

Para cada fila proveniente do FROM da consulta principal, execútase a subconsulta, substituíndo as referencias a columnas da consulta principal polo valor da fila procesada.

Subconsultas correlacionadas

```
SELECT *
```

```
FROM emppro a
```

```
WHERE hours = (SELECT MAX(hours)  
                FROM emppro  
                WHERE prono=a.prono)
```

Comprobación da
condición where da
consulta principal

empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7934	1001	4
7654	1001	16
7499	1004	15
7521	1004	10
7844	1005	6
7499	1005	12
7654	1006	15
7654	1008	5
7521	1008	8

WHERE 4 = 16

```
SELECT MAX(hours)  
FROM emppro  
WHERE prono=1001
```

Execución da
subconsulta
para a fila en
cor vermella

7934 non é o que máis horas traballa no proxecto 1001

Subconsultas correlacionadas

```
SELECT *
```

```
FROM emppro a
```

```
WHERE hours = (SELECT MAX(hours)  
                FROM emppro  
                WHERE prono=a.prono)
```

Comprobación da
condición where da
consulta principal

empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7934	1001	4
7654	1001	16
7499	1004	15
7521	1004	10
7844	1005	6
7499	1005	12
7654	1006	15
7654	1008	5
7521	1008	8

```
WHERE 16 = 16
```

```
SELECT MAX(hours)  
FROM emppro  
WHERE prono=1001
```

Execución da
subconsulta
para a fila en
cor vermella

7654 é o que máis horas traballa no proxecto 1001

Subconsultas correlacionadas

```
SELECT *
```

```
FROM emppro a
```

```
WHERE hours = (SELECT MAX(hours)  
                FROM emppro  
                WHERE prono=a.prono)
```

Comprobación da
condición where da
consulta principal

empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7934	1001	4
7654	1001	16
7499	1004	15
7521	1004	10
7844	1005	6
7499	1005	12
7654	1006	15
7654	1008	5
7521	1008	8

```
WHERE 15 = 15
```

```
SELECT MAX(hours)  
FROM emppro  
WHERE prono=1004
```

Execución da
subconsulta
para a fila en
cor vermella

7499 é o que máis horas traballa no proxecto 1004

Subconsultas correlacionadas

```
SELECT *
```

```
FROM emppro a
```

```
WHERE hours = (SELECT MAX(hours)  
                FROM emppro  
                WHERE prono=a.prono)
```

Comprobación da
condición where da
consulta principal

empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7934	1001	4
7654	1001	16
7499	1004	15
7521	1004	10
7844	1005	6
7499	1005	12
7654	1006	15
7654	1008	5
7521	1008	8

```
WHERE 10 = 15
```

```
SELECT MAX(hours)  
FROM emppro  
WHERE prono=1004
```

Execución da
subconsulta
para a fila en
cor vermella

7521 non é o que máis horas traballa no proxecto 1004

Subconsultas correlacionadas

```
SELECT *  
FROM emppro a  
WHERE hours = (SELECT MAX(hours)  
               FROM emppro  
               WHERE prono=a.prono)
```

Comprobación da
condición where da
consulta principal

empno numeric(4,0)	prono numeric(4,0)	hours numeric(2,0)
7934	1001	4
7654	1001	16
7499	1004	15
7521	1004	10
7844	1005	6
7499	1005	12
7654	1006	15
7654	1008	5
7521	1008	8

WHERE 6 = 12

```
SELECT MAX(hours)  
FROM emppro  
WHERE prono=1005
```

Execución da
subconsulta
para a fila en
cor vermella

7844 non é o que máis horas traballa no proxecto 1005

E así ata rematar con todas as filas da consulta principal

Subconsultas correlacionadas con predicado exists

Predicado de existencia (**EXISTS**)

Comproba se a subconsulta devolve ou non filas. Devolve CERTO se a subconsulta devolve filas e FALSO se non ten filas.

Formato: **[NOT] EXISTS (subconsulta)**

A subconsulta pode devolver unha táboa con calquera número de columnas e filas.

```
SELECT dname
FROM dept d
WHERE EXISTS (select *
              from emp
              where deptno=d.deptno)
```

Exercicios

1. Mostra o(s) empregado(s) co salario máis alto de cada departamento.
2. Mostra o código do(s) empregado(s) que máis horas traballa(n) en cada proxecto.
3. Mostra o nome do(s) empregado(s) que máis horas traballa(n) en cada proxecto.
4. Mostra o nome do(s) empregado(s) que máis horas traballa(n) en cada proxecto.
Mostra tamén o nome do proxecto.
5. Para cada departamento mostra o seu nome e cantos empregados dese departamento teñen un salario maior ao salario medio do seu departamento.
6. Para cada departamento mostra o seu nome e cantos empregados gañan máis que o seu xefe.

Exercicios

1.

```
Select ename, deptno, sal
from emp e
where sal= (Select max(sal) from emp
           where deptno=e.deptno)
```
2.

```
Select pronno, empno, hours
from emppro ep
where hours= (Select max(hours) from emppro
             where pronno=ep.pronno)
```
3.

```
Select pronno, ename, hours
from emppro ep join emp e on e.empno=ep.empno
where hours= (Select max(hours) from emppro
             where pronno=ep.pronno)
```
4.

```
Select pname, ename, hours
from emppro ep join emp e on e.empno=ep.empno
              join pro p on ep.pronno=p.pronno
where hours= (Select max(hours) from emppro
             where pronno=ep.pronno)
```
5.

```
Select dname, count(*)
from emp e join dept d on e.deptno=d.deptno
where sal > (Select avg(sal) from emp
            where deptno=d.deptno)
group by d.deptno, dname
```
6.

```
Select dname, count(*)
from emp e join dept d on e.deptno=d.deptno
where sal > (Select sal from emp
            where empno=e.mgr)
group by d.deptno, dname
```

Guión

Introducción

Conceptos previos

Sentenza Select

Distinct

Order by

Predicados

Funcións

Group by

Having

Join

Subconsultas

Composición de consultas

Composición de consultas

SQL dispón de tres operadores de conxuntos: unión (UNION), intersección (INTERSECT) e diferenza (EXCEPT).

Permiten realizar esas operacións coas filas resultantes de dúas sentenzas select.

Formato: **consulta1**

{UNION|INTERSECT|EXCEPT} [ALL|DISTINCT]

consulta2

[order by <expre1>,...]

As dúas consultas deben ser “unión compatibles”:

Deben ter igual número de columnas.

Correspondencia de tipos entre as columnas situadas na mesma posición (contando desde a esquerda).

ALL permite filas duplicadas, DISTINCT elimina filas duplicadas. O predefinido é DISTINCT.

Só pode haber un order by, que se aplicaría sobre o resultado da operación conxuntista.

Composición de consultas

```
SELECT ename, sal+comm AS "Ingresos totales", 'Incluye comisión' AS "Comisión?"  
FROM emp  
WHERE comm is not null  
  
UNION
```

```
SELECT ename, sal, 'No tiene comisión'  
FROM emp  
WHERE comm is null  
ORDER BY "Ingresos totales"
```

ename	Ingresos totales	Comisión?
character varchar2(10)	numeric(8,2)	text
SMITH	800.00	No tiene comisión
JAMES	950.00	No tiene comisión
ADAMS	1100.00	No tiene comisión
MILLER	1300.00	No tiene comisión
TURNER	1500.00	Incluye comisión
WARD	1750.00	Incluye comisión
ALLEN	1900.00	Incluye comisión
CLARK	2450.00	No tiene comisión
MARTIN	2650.00	Incluye comisión
BLAKE	2850.00	No tiene comisión
JONES	2975.00	No tiene comisión
FORD	3000.00	No tiene comisión
SCOTT	3000.00	No tiene comisión
KING	5000.00	No tiene comisión