

# SQL

Select avanzado



UNIVERSIDADE DA CORUÑA



José R. Paramá



# Guión

Introducción

Tablas usadas en los ejemplos

Sentencia Select

Join avanzado

Subconsultas de fila

Expresiones de consulta

# Tablas usadas en los ejemplos

EMP		
Campo	Tipo	Descripción
<u>EMPNO:</u>	NUMBER(4) NOT NULL	Número o código del empleado. Es la clave primaria de la tabla.
ENAME	VARCHAR2(10)	Nombre del empleado
JOB	VARCHAR2(9)	Trabajo del empleado
MGR	NUMBER(4)	Código del jefe del empleado. Clave foránea que referencia (cíclicamente) la tabla EMP
HIREDATE	DATE	Fecha de contratación.
SAL	NUMBER(7, 2)	Salario mensual del empleado
COMM	NUMBER(7, 2)	Comisión
DEPTNO	NUMBER(2)	Código del departamento al que el empleado está adscrito. Clave foránea que referencia la tabla DEPT

Un nulo en COMM significa que el empleado no trabaja a comisión (el valor no procede).

Un nulo en MGR significa que no tiene jefe (también "no procede")

DEPT		
Campo	Tipo	Descripción
<u>DEPTNO</u>	NUMBER(2) NOT NULL	Número o código del departamento. Es la clave primaria de la tabla.
DNAME	VARCHAR2(14)	Nombre del departamento.
LOC	VARCHAR2(13)	Localidad (o ciudad) donde el departamento está ubicado.

PRO		
Campo	Tipo	Descripción
<u>PRONO</u>	NUMBER(4) NOT NULL	Número o código del Proyecto. Es la clave primaria de la tabla.
PNAME	VARCHAR2(10)	Nombre del proyecto.
LOC	VARCHAR2(13)	Ciudad donde se realiza el proyecto.
DEPTNO	NUMBER(2)	Número del departamento controlador del proyecto. Clave foránea que referencia la tabla DEPT

EMPPRO		
Campo	Tipo	Descripción
<u>EMPNO</u>	NUMBER(4) NOT NULL	Número o código del empleado. Clave foránea que referencia la tabla EMP
<u>PRONO</u>	NUMBER(4) NOT NULL	Número o código del proyecto. Clave foránea que referencia la tabla PRO
HOURS	NUMBER(2)	Horas que ha trabajado un empleado en un proyecto.

Es la clave primaria de la tabla

# Tablas usadas en los ejemplos

```
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

Tablas usadas en los ejemplos

Sentencia Select

Join avanzado

Subconsultas de fila

Expresiones de consulta

# Sentencia Select

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

El orden de ejecución de las cláusulas y la función de cada una es:

1. **FROM**(obligatoria)

Partiendo de una o más tablas *obtiene una única tabla* que será procesada por el resto de cláusulas

2. **WHERE** (optativa)

De las filas que le pasa el FROM, *elimina las filas que NO HACEN CIERTA la condición* especificada

3. **GROUP BY** (optativa)

Realiza un agrupamiento de las filas.

3. **HAVING** (optativa)

De los grupos de filas que llegan, *elimina aquéllos que NO HACEN CIERTA la condición* especificada

3. **SELECT** (obligatoria)

Muestra una columna por expresión, o todas las columnas si se indica \*

6. **ORDER BY** (optativa)

Permite determinar el criterio de ordenación de las filas de la tabla resultado. Sin ella obtendremos las mismas filas, pero no hay garantía de en qué orden, que será el que dicte la estrategia seguida por el SGBD para extraer los datos.

# Sentencia Select

Obtén el departamento con la mayor suma de salarios, considerando los empleados contratados después del 1/6/81. Mostrar el nombre del departamento y la suma de los salarios de los empleados de ese departamento que fueron contratados después del 1/6/1981.

```
select dname, sum(sal)
  from emp e join dept d on e.deptno=d.deptno
 where hiredate >= to_date('01/06/1981','dd/mm/yyyy')
 group by d.deptno, dname
 having sum(sal) >= ALL (select sum(sal)
                           from emp
                          where hiredate>='to_date('01/06/1981','dd/mm/yyyy')
                            group by deptno)
```

# Sentencia Select

```
select dname, sum(sal)
  from emp e join dept d on e.deptno=d.deptno
 where hiredate >= to_date('01/06/1981','dd/mm/yyyy')
group by d.deptno, dname
having sum(sal) >= ALL (select sum(sal)
                           from emp
                          where hiredate>= to_date('01/06/1981','dd/mm/yyyy')
                          group by deptno)
```

The diagram illustrates the join operation between the EMP and DEPT tables. A horizontal line labeled 'EMP E' connects the EMP table to the DEPT table. Another horizontal line labeled 'DEPT D' connects the DEPT table back to the EMP table. The columns involved in the join are highlighted in blue: DEPTNO from the EMP table and DEPTNO from the DEPT table.

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

# Sentencia Select

```
select dname, sum(sal)
  from emp e join dept d on e.deptno=d.deptno
 where hiredate >= to_date('01/06/1981','dd/mm/yyyy')
group by d.deptno, dname
having sum(sal) >= ALL (select sum(sal)
                           from emp
                          where hiredate>= to_date('01/06/1981','dd/mm/yyyy')
                           group by deptno)
```

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

# Sentencia Select

```
select dname, sum(sal)
  from emp e join dept d on e.deptno=d.deptno
 where hiredate >= to_date('01/06/1981','dd/mm/yyyy')
 group by d.deptno, dname
having sum(sal) >= ALL (select sum(sal)
                           from emp
                         where hiredate>= to_date('01/06/1981','dd/mm/yyyy')
                           group by deptno)
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7782	CLARK	MANAGER	7839	09/06/81	2,450		10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT		17/11/81	5,000		10	10	ACCOUNTING	NEW YORK
7934	MILLER	CLERK	7782	23/01/82	1,300		10	10	ACCOUNTING	NEW YORK
7902	FORD	ANALYST	7566	03/12/81	3,000		20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	19/04/87	3,000		20	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	23/05/87	1,100		20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	03/12/81	950		30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30	30	SALES	CHICAGO

# Sentencia Select

```
select dname, sum(sal)
  from emp e join dept d on e.deptno=d.deptno
 where hiredate >= to_date('01/06/1981','dd/mm/yyyy')
group by d.deptno, dname
```

```
having sum(sal) >= ALL (select sum(sal)
                           from emp
                           where hiredate >= to_date('01/06/1981','dd/mm/yyyy')
                           group by deptno)
```

EMPNO	ENAME	HIREDATE	SAL	DEPTNO	DNAME
7782	CLARK	09/06/81	2,450	10	ACCOUNTING
7839	KING	17/11/81	5,000	10	ACCOUNTING
7934	MILLER	23/01/82	1,300	10	ACCOUNTING
7902	FORD	03/12/81	3,000	20	RESEARCH
7788	SCOTT	19/04/87	3,000	20	RESEARCH
7876	ADAMS	23/05/87	1,100	20	RESEARCH
7900	JAMES	03/12/81	950	30	SALES
7654	MARTIN	28/09/81	1,250	30	SALES
7844	TURNER	08/09/81	1,500	30	SALES

sum(sal)

8750 ✓ >=ALL

7100 ✗

3700 ✗

SUM(SAL)

3700  
7100  
8750

# Sentencia Select

```
select dname, sum(sal)
  from emp e join dept d on e.deptno=d.deptno
 where hiredate >= to_date('01/06/1981','dd/mm/yyyy')
 group by d.deptno, dname
 having sum(sal) >= ALL (select sum(sal)
                           from emp
                          where hiredate>=to_date('01/06/1981','dd/mm/yyyy')
                            group by deptno)
```

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

dname	sum(sal)
-----	-----
<b>ACCOUNTING</b>	<b>8750</b>

# Sentencia Select

Para cada proyecto muestra el departamento cuyos empleados han dedicado más horas a dicho proyecto. Muestra el número de departamento, número de proyecto y horas dedicadas por los empleados del departamento al proyecto.

```
select e.deptno, ep.prono, sum(hours)
from emp e join emppro ep on e.empno=ep.empno
group by e.deptno, ep.prono
having sum(hours) >=ALL (select sum(hours)
                           from emp a join emppro b on a.empno=b.empno
                           group by a.deptno, b.prono
                           having b.prono=ep.prono)
```

# Sentencia Select

```
select e.deptno, ep.prono, sum(hours)
from emp e join emppro ep on e.empno=ep.empno
group by e.deptno, ep.prono
having sum(hours) >=ALL (select sum(hours)
                           from emp a join emppro b on a.empno=b.empno
                           group by a.deptno, b.prono
                           having b.prono=ep.prono)
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EMPNO	PRONO	HOURS
7934	MILLER	CLERK	7782	23/01/82	1,300		10	7934	1001	4
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30	7654	1001	16
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30	7521	1004	10
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30	7499	1004	15
7844	TURNER	SALESMAN	7698	08/09/81	1,500	0	30	7844	1005	6
7499	ALLEN	SALESMAN	7698	20/02/81	1,600	300	30	7499	1005	12
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30	7654	1006	15
7521	WARD	SALESMAN	7698	22/02/81	1,250	500	30	7521	1008	8
7654	MARTIN	SALESMAN	7698	28/09/81	1,250	1,400	30	7654	1008	5

# Sentencia Select

```
select e.deptno, ep.prono, sum(hours)
from emp e join emppro ep on e.empno=ep.empno
group by e.deptno, ep.prono
```

```
having sum(hours) >=ALL (select sum(hours)
                           from emp a join emppro b on a.empno=b.empno
                           group by a.deptno, b.prono
                           having b.prono=ep.prono)
```

DEPTNO	HOURS	PRONO
10	4	1001
30	16	1001
30	10	1004
30	15	1004
30	6	1005
30	12	1005
30	15	1006
30	8	1008
30	5	1008

**sum(hours)**

-----  
4  
16

>=ALL

select sum(hours)  
from emp a join emppro b on a.empno=b.empno  
group by a.deptno, b.prono  
having b.prono=1001

>=ALL

select sum(hours)  
from emp a join emppro b on a.empno=b.empno  
group by a.deptno, b.prono  
having b.prono=1001

SUM(HOURS)
4
16

# Ejercicios

1. Para cada empleado muestra su nombre y cuántas horas trabajó en proyectos.
2. Para cada empleado muestra su nombre, el nombre de su jefe, y el departamento para el que trabaja su jefe.
3. Para cada jefe, muestra su nombre y cuántos subordinados tiene.
4. Muestra el nombre de proyectos donde se ha trabajado (en total, todos los empleados) más de 15 horas
5. Muestra los departamentos (nombre) donde hay por lo menos dos empleados con el mismo puesto de trabajo. No debe aparecer repetidos.
6. Para cada proyecto muestra el empleado que ha trabajado más horas. Muestra los nombres del proyecto y empleado
7. Muestra el jefe tal que la suma de los salarios de sus subordinados es la más alta. Muestra su nombre y la suma de salarios de sus subordinados.
8. Muestra para cada departamento el proyecto controlado por él donde se han trabajado más horas.

# Ejercicios

1. Select ename, sum(hours)  
from emp e join emppro ep on e.empno=ep.empno  
group by e.empno, ename
2. 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
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 distinct dname  
from emp e join dept d on e.deptno=d.deptno  
group by d.deptno, dname, job  
having count(\*)>=2
6. select pname, ename, hours  
from pro p join emppro ep on ep.prono=p.prono  
join emp e on e.empno=ep.empno  
where hours >= ALL (select hours  
from emppro  
where prono=p.prono)
7. select j.ename, sum(s.sal)  
from emp j join emp s on s.mgr=j.empno  
group by j.empno, j.ename  
having sum(s.sal) >= ALL (select sum(sal)  
from emp  
where mgr is not null  
group by mgr)
8. select sum(hours), pname, deptno  
from pro p join emppro ep on ep.prono=p.prono  
group by p.prono, pname, p.deptno  
having sum(hours) >= ALL (select sum(hours)  
from emppro ep1 join pro p1  
on ep1.prono=p1.prono  
where p1.deptno=p.deptno  
group by p1.prono)

# Guión

Introducción

Tablas usadas en los ejemplos

Sentencia Select

Join avanzado

Subconsultas de fila

Expresiones de consulta

# INNER JOIN

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

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

Se puede desambiguar usando alias.

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

Sólo es necesario desambiguar en los nombres de columna que aparecen en las dos tablas.

# JOIN Natural

La condición de Join es la igualdad en *TODOS* los atributos que se llamen igual.

Las columnas que se llamen igual, por tanto, solo aparecen una vez en el resultado.

## Natural

SQL> select * from emp natural join dept;									
DEPTNO	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DNAME	LOC
10	7782	CLARK	MANAGER	7839	09/06/81	2450		ACCOUNTING	NEW YORK
10	7839	KING	PRESIDENT		17/11/81	5000		ACCOUNTING	NEW YORK
10	7934	MILLER	CLERK	7782	23/01/82	1300		ACCOUNTING	NEW YORK
20	7566	JONES	MANAGER	7839	02/04/81	2975		RESEARCH	DALLAS
20	7902	FORD	ANALYST	7566	03/12/81	3000		RESEARCH	DALLAS
20	7876	ADAMS	CLERK	7788	23/05/87	1100		RESEARCH	DALLAS
20	7369	SMITH	CLERK	7902	17/12/80	800		RESEARCH	DALLAS
20	7788	SCOTT	ANALYST	7566	19/04/87	3000		RESEARCH	DALLAS
30	7521	WARD	SALESMAN	7698	22/02/81	1250	500	SALES	CHICAGO
30	7844	TURNER	SALESMAN	7698	08/09/81	1500	0	SALES	CHICAGO
30	7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	SALES	CHICAGO
30	7900	JAMES	CLERK	7698	03/12/81	950		SALES	CHICAGO
30	7698	BLAKE	MANAGER	7839	01/05/81	2850		SALES	CHICAGO
30	7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	SALES	CHICAGO

Solo  
aparece  
una vez

Vs

## Inner

SQL> select * from emp e join dept d on e.deptno=d.deptno;											
DEPTNO	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
10	7782	CLARK	MANAGER	7839	09/06/81	2450		10	10	ACCOUNTING	NEW YORK
10	7839	KING	PRESIDENT		17/11/81	5000		10	10	ACCOUNTING	NEW YORK
10	7934	MILLER	CLERK	7782	23/01/82	1300		10	10	ACCOUNTING	NEW YORK
20	7566	JONES	MANAGER	7839	02/04/81	2975		20	20	RESEARCH	DALLAS
20	7902	FORD	ANALYST	7566	03/12/81	3000		20	20	RESEARCH	DALLAS
20	7876	ADAMS	CLERK	7788	23/05/87	1100		20	20	RESEARCH	DALLAS
20	7369	SMITH	CLERK	7902	17/12/80	800		20	20	RESEARCH	DALLAS
20	7788	SCOTT	ANALYST	7566	19/04/87	3000		20	20	RESEARCH	DALLAS
30	7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	30	SALES	CHICAGO
30	7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30	30	SALES	CHICAGO
30	7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	30	SALES	CHICAGO
30	7900	JAMES	CLERK	7698	03/12/81	950		30	30	SALES	CHICAGO
30	7698	BLAKE	MANAGER	7839	01/05/81	2850		30	30	SALES	CHICAGO
30	7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	30	SALES	CHICAGO

# JOIN Natural

No hace falta desambiguar en ninguna parte de la sentencia select.

```
SQL> select * from emp natural join dept  
2 where deptno=10;
```

DEPTNO	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DNAME	LOC
10	7782	CLARK	MANAGER	7839	09/06/81	2450		ACCOUNTING	NEW YORK
10	7839	KING	PRESIDENT		17/11/81	5000		ACCOUNTING	NEW YORK
10	7934	MILLER	CLERK	7782	23/01/82	1300		ACCOUNTING	NEW YORK

```
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 pro natural join dept;
```

LOC	DEPTNO	PRONO	PNAME	DNAME
CHICAGO	30	1004	P4	SALES
CHICAGO	30	1005	P5	SALES

```
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.

Se excede en las igualdades, también hace el join por el atributo *LOC*.

# JOIN Natural

Para evitar este problema se puede usar USING, que hace unjoin natural sólo por las columnas indicadas.

**FROM <tabla1> JOIN <tabla2> USING(<col1>[ , <col2> ... ] )**

```
SQL> select * from pro natural join dept;
```

LOC	DEPTNO	PRONO	PNAME	DNAME
CHICAGO	30	1004	P4	SALES
CHICAGO	30	1005	P5	SALES

```
SQL> select * from pro join dept using(deptno);
```

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

# CROSS JOIN

Es un producto cartesiano

```
select * from emp cross join dept
```

Todas columnas  
de las dos tablas

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

... continúa con más filas...

Todas las filas de  
emp pegadas a  
todas las de  
dept

# JOIN exterior

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

Podemos forzar a que las filas de una (o de las dos) tabla de entrada que en el INNER join no aparecen por no hacer cierta la condición de join, salgan rellenando las columnas del otro lado con nulos.

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

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

Nulos

Vamos forzar que salgan las filas del lado derecho que no salen en el INNER

Las filas del lado derecho que “emparejan”, salen igual que en el INNER join

Las filas del lado derecho que “no emparejan”, salen llenando las columnas del lado izquierdo 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
```

SUBORDINAD	MGR	EMPNO	JEFE
FORD	7566	7566	JONES
SCOTT	7566	7566	JONES
JAMES	7698	7698	BLAKE
TURNER	7698	7698	BLAKE
MARTIN	7698	7698	BLAKE
WARD	7698	7698	BLAKE
ALLEN	7698	7698	BLAKE
MILLER	7782	7782	CLARK
ADAMS	7788	7788	SCOTT
CLARK	7839	7839	KING
BLAKE	7839	7839	KING
JONES	7839	7839	KING
SMITH	7902	7902	FORD
KING			

Encaje “normal”

} Forzada por el left join

# 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 por el left join

Encaje “normal”

} Forzada por el right join

# JOIN

La condición de join no tiene porqué ser una condición de igualdad

```
select a.ename, a.sal, b.ename, b.sal  
from emp a join emp b on a.sal>b.sal
```

Y puede ser una combinación

```
1 select e.ename subordinado, e.sal, j.ename jefe, j.sal  
2   from emp e join emp j on e.mgr=j.empno and e.sal>j.sal  
3*
```

SUBORDINAD	SAL	JEFE	SAL
FORD	3000	JONES	2975
SCOTT	3000	JONES	2975

# JOIN exterior

Para cada jefe mostrar cuántos subordinados ganan más que él. Si no hay ninguno, mostrar un 0.

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

ENAME	COUNT(E.ENAME)
JONES	2

```
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
```

ENAME	COUNT(E.ENAME)
WARD	0
JONES	2
TURNER	0
ADAMS	0
ALLEN	0
SMITH	0
CLARK	0
KING	0
BLAKE	0
JAMES	0
FORD	0
SCOTT	0
MARTIN	0
MILLER	0

Join

Filtrar las filas que cumplen esa condición.  
Elimina el efecto de join exterior

SQL:			
1 Select e.ename, e.sal, j.ename, j.sal 2* from emp e right join emp j on e.mgr=j.empno			
ENAME	SAL	ENAME	SAL
		SMITH	800
		ALLEN	1600
		WARD	1250
FORD	3000	JONES	2975
SCOTT	3000	JONES	2975
		MARTIN	1250
TURNER	1500	BLAKE	2850
ALLEN	1600	BLAKE	2850
WARD	1250	BLAKE	2850
JAMES	950	BLAKE	2850
MARTIN	1250	BLAKE	2850
MILLER	1300	CLARK	2450
ADAMS	1100	SCOTT	3000
BLAKE	2850	KING	5000
JONES	2975	KING	5000
CLARK	2450	KING	5000
		TURNER	1500
		ADAMS	1100
		JAMES	950
SMITH	800	FORD	3000
		MILLER	1300

Where

Join

SQL:			
1 Select e.ename, e.sal, j.ename, j.sal 2* from emp e right join emp j on e.mgr=j.empno and e.sal>j.sal			
ENAME	SAL	ENAME	SAL
		SMITH	800
		ALLEN	1600
		WARD	1250
FORD	3000	JONES	2975
SCOTT	3000	JONES	2975
		MARTIN	1250
BLAKE		CLARK	2450
CLARK		SCOTT	3000
SCOTT		KING	5000
KING		TURNER	1500
TURNER		ADAMS	1100
ADAMS		JAMES	950
JAMES		FORD	3000
FORD		MILLER	1300

# JOIN exterior natural

También se pueden hacer joins naturales exteriores.

En este caso, observa que sólo hay una columna *deptno* con el valor 40 (no hay una columna correspondiente a la tabla de empleado con *deptno* nulo).

```
SQL> select * from emp natural right join dept;
```

DEPTNO	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DNAME	LOC
--									
10	7782	CLARK	MANAGER	7839	09/06/81	2450		ACCOUNTING	NEW YORK
10	7839	KING	PRESIDENT		17/11/81	5000		ACCOUNTING	NEW YORK
10	7934	MILLER	CLERK	7782	23/01/82	1300		ACCOUNTING	NEW YORK
20	7566	JONES	MANAGER	7839	02/04/81	2975		RESEARCH	DALLAS
20	7902	FORD	ANALYST	7566	03/12/81	3000		RESEARCH	DALLAS
20	7876	ADAMS	CLERK	7788	23/05/87	1100		RESEARCH	DALLAS
20	7369	SMITH	CLERK	7902	17/12/80	800		RESEARCH	DALLAS
20	7788	SCOTT	ANALYST	7566	19/04/87	3000		RESEARCH	DALLAS
30	7521	WARD	SALESMAN	7698	22/02/81	1250	500	SALES	CHICAGO
30	7844	TURNER	SALESMAN	7698	08/09/81	1500	0	SALES	CHICAGO
30	7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	SALES	CHICAGO
30	7900	JAMES	CLERK	7698	03/12/81	950		SALES	CHICAGO
30	7698	BLAKE	MANAGER	7839	01/05/81	2850		SALES	CHICAGO
30	7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	SALES	CHICAGO
40								OPERATIONS	BOSTON
15 filas seleccionadas.									

# Join de más de dos tablas

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

En primer lugar se unen <tabla1> y <tabla2>, con su condición de join.

Esto da lugar a una tabla.

Esa tabla resultante se une a la <tabla3>, con su condición de join, lo que resulta en otra tabla.

Y así sucesivamente.

# Join de más de dos tablas

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

Esto NO ES CORRECTO!!

# Join de más de dos tablas

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

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

Primer paso

empno	prono	hours
numeric(4,0)	numeric(4,0)	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	ename	job	mgr	hiredate	sal	comm	deptno	empno	prono	hours
numeric(4,0)	character varying(10)	character varying(9)	numeric(4,0)	date	numeric(7,2)	numeric(7,2)	numeric(2,0)	numeric(4,0)	numeric(4,0)	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 más de dos tablas

```
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)	pron umeric(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

pron umeric	pname characte	loc character varyin	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	ename characte	job character v	mgr numer	hiredate date	sal numeric(7,2)	comm numeric(7,2)	deptno numeric	empno numeric	pron umeric	hours numeric	pron umeric(4,0)	pname charact	loc character varyir	deptno numeric
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 más de dos tablas

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

empno	ename	job	mgr	hiredate	sal	comm	deptno	empno	pron	hours	pron	pname	loc	deptno
numeric(4,0)	character varying(10)	character varying(10)	numeric(4,0)	date	numeric(7,2)	numeric(7,2)	numeric	numeric(4,0)	numeric(4,0)	numeric	numeric(4,0)	character varying(10)	character varying(10)	numeric
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	pname	hours
character varying(10)	character	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

# Join de más de dos tablas

Los joins de más de dos tablas pueden no ser tan sencillos.

Cuántos empleados tiene cada departamento:

```
SQL> select dname, count(*)
  2  from emp e join dept d on e.deptno=d.deptno
  3  group by d.deptno, dname;
```

DNAME	COUNT(*)
ACCOUNTING	3
RESEARCH	5
SALES	6

Para cada departamento, mostrar cuántos empleados tiene, y cuántos proyectos controla: ¿Un join más?

```
1  select dname, count(e.empno), count(p.prono)
  2  from emp e join dept d on e.deptno=d.deptno
  3          join pro p on d.deptno=p.deptno
  4* group by d.deptno, dname
```

DNAME	COUNT(E.EMPNO)	COUNT(P.PRONO)
RESEARCH	5	5
SALES	24	24

!Accounting desaparece y Sales pasa a tener 24 empleados!

# Join de más de dos tablas

La desaparición de Accounting se debe a que no controla ningún proyecto

```
SQL> select e.empno, ename, d.deptno, dname
  2  from emp e join dept d on d.deptno=e.deptno;
```

EMPNO	ENAME	DEPTNO	DNAME
7782	CLARK	10	ACCOUNTING
7839	KING	10	ACCOUNTING
7934	MILLER	10	ACCOUNTING
7566	JONES	20	RESEARCH
7902	FORD	20	RESEARCH
7876	ADAMS	20	RESEARCH
7369	SMITH	20	RESEARCH
7788	SCOTT	20	RESEARCH
7521	WARD	30	SALES
7844	TURNER	30	SALES
7499	ALLEN	30	SALES
7900	JAMES	30	SALES
7698	BLAKE	30	SALES
7654	MARTIN	30	SALES

```
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.

Con un join exterior se soluciona

```
1 select d.deptno, dname, count(e.empno), count(p.prono)
2  from emp e join dept d on d.deptno=e.deptno
3      left join pro p on d.deptno=p.deptno
4* group by d.deptno, dname
```

DEPTNO	DNAME	COUNT(E.EMPNO)	COUNT(P.PRONO)
10	ACCOUNTING	3	0
20	RESEARCH	5	5
30	SALES	24	24

Pero los números no cuadran

```
SQL> select dname, count(*)
  2  from emp e join dept d on e.deptno=d.deptno
  3  group by d.deptno, dname;
```

DNAME	COUNT(*)
ACCOUNTING	3
RESEARCH	5
SALES	6

# Join de más de dos tablas

```
1 select d.deptno, dname, count(e.empno), count(p.prono)
2 from emp e join dept d on d.deptno=e.deptno
3      left join pro p on d.deptno=p.deptno
4* group by d.deptno, dname
```

La condición de Join es *on d.deptno=p.deptno*

```
1 select e.empno, ename, dname, d.deptno
2 from emp e join dept d on d.deptno=e.deptno
3* order by d.deptno
```

EMPNO	ENAME	DNAME	DEPTNO
7782	CLARK	ACCOUNTING	10
7839	KING	ACCOUNTING	10
7934	MILLER	ACCOUNTING	10
7566	JONES	RESEARCH	20
7902	FORD	RESEARCH	20
7876	ADAMS	RESEARCH	20
7369	SMITH	RESEARCH	20
7788	SCOTT	RESEARCH	20
7521	WARD	SALES	30
7844	TURNER	SALES	30
7499	ALLEN	SALES	30
7900	JAMES	SALES	30
7698	BLAKE	SALES	30
7654	MARTIN	SALES	30

14 filas seleccionadas.

```
SQL> select deptno, pname, prono
  2  from pro
  3  order by deptno;
```

DEPTNO	PNAME	PRONO
20	P1	1001
30	P8	1008
30	P6	1006
30	P4	1004
30	P5	1005

Todas estas filas  
“emparejan”

```
1 select d.deptno,dname,count(distinct e.empno), count(distinct p.prono)
2 from emp e join dept d on d.deptno=e.deptno
3      left join pro p on d.deptno=p.deptno
4* group by d.deptno, dname
```

DEPTNO	DNAME	COUNT(DISTINCT E.EMPNO)	COUNT(DISTINCT P.PRONO)
10	ACCOUNTING	3	0
20	RESEARCH	5	1
30	SALES	6	4

Se crean  
 $6 \times 4 = 24$   
filas.

```
1 select e.empno, ename, d.deptno,dname, p.prono, pname
2 from emp e join dept d on d.deptno=e.deptno
3      left join pro p on d.deptno=p.deptno
4* order by p.prono
```

EMPNO	ENAME	DEPTNO	DNAME	PRONO	PNAME
7902	FORD	20	RESEARCH	1001	P1
7566	JONES	20	RESEARCH	1001	P1
7369	SMITH	20	RESEARCH	1001	P1
7876	ADAMS	20	RESEARCH	1001	P1
7788	SCOTT	20	RESEARCH	1001	P1
7521	WARD	30	SALES	1004	P4
7499	ALLEN	30	SALES	1004	P4
7698	BLAKE	30	SALES	1004	P4
7900	JAMES	30	SALES	1004	P4
7844	TURNER	30	SALES	1004	P4
7654	MARTIN	30	SALES	1004	P4
7844	TURNER	30	SALES	1005	P5
7900	JAMES	30	SALES	1005	P5
7499	ALLEN	30	SALES	1005	P5
7698	BLAKE	30	SALES	1005	P5
7654	MARTIN	30	SALES	1005	P5
7521	WARD	30	SALES	1005	P5
7698	BLAKE	30	SALES	1006	P6
7521	WARD	30	SALES	1006	P6
7900	JAMES	30	SALES	1006	P6
7654	MARTIN	30	SALES	1006	P6
7499	ALLEN	30	SALES	1006	P6
7844	TURNER	30	SALES	1006	P6
7654	MARTIN	30	SALES	1008	P8
7844	TURNER	30	SALES	1008	P8
7521	WARD	30	SALES	1008	P8
7698	BLAKE	30	SALES	1008	P8
7499	ALLEN	30	SALES	1008	P8
7900	JAMES	30	SALES	1008	P8
7934	MILLER	10	ACCOUNTING		
7839	KING	10	ACCOUNTING		
7782	CLARK	10	ACCOUNTING		

32 filas seleccionadas.

# Join de más de dos tablas exterior

Mostrar las localidades donde ha trabajado cada empleado, si no ha trabajado en ninguna (no ha trabajado en proyectos) mostrar un nulo.

```
SQL> select * from emp e left join emppro ep on e.empno=ep.empno order by ep.empno;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EMPNO	PRONO	HOURS
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1004	15
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1005	12
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1008	8
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1004	10
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1006	15
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1008	5
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1001	16
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30	7844	1005	6
7934	MILLER	CLERK	7782	23/01/82	1300		10	7934	1001	4
7782	CLARK	MANAGER	7839	09/06/81	2450					10
7788	SCOTT	ANALYST	7566	19/04/87	3000					20
7369	SMITH	CLERK	7902	17/12/80	800					20
7876	ADAMS	CLERK	7788	23/05/87	1100					20
7900	JAMES	CLERK	7698	03/12/81	950					30
7566	JONES	MANAGER	7839	02/04/81	2975					20
7698	BLAKE	MANAGER	7839	01/05/81	2850					30
7902	FORD	ANALYST	7566	03/12/81	3000					20
7839	KING	PRESIDENT		17/11/81	5000					10

18 filas seleccionadas.

Desaparecen

```
1 select * from emp e left join emppro ep on e.empno=ep.empno  
2          join pro p on ep.prono=p.prono  
3* order by ep.empno
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EMPNO	PRONO	HOURS	PRONO	PNAME	LOC	DEPTNO
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1004	15	1004	P4	CHICAGO	30
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1005	12	1005	P5	CHICAGO	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1004	10	1004	P4	CHICAGO	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1008	8	1008	P8	NEW YORK	30
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1008	5	1008	P8	NEW YORK	30
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1001	16	1001	P1	BOSTON	20
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1006	15	1006	P6	LOS ANGELES	30
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30	7844	1005	6	1005	P5	CHICAGO	30
7934	MILLER	CLERK	7782	23/01/82	1300		10	7934	1001	4	1001	P1	BOSTON	20

filas seleccionadas.

# Join de más de dos tablas exterior

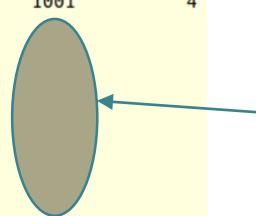
El problema es el segundo (**inner**) join, la condición es: **ON ep.prono=p.prono**

1er join

```
SQL> select * from emp e left join emppro ep on e.empno=ep.empno order by ep.empno;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EMPNO	PRONO	HOURS
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1004	15
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1005	12
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1008	8
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1004	10
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1006	15
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1008	5
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1001	16
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30	7844	1005	6
7934	MILLER	CLERK	7782	23/01/82	1300		10	7934	1001	4
7782	CLARK	MANAGER	7839	09/06/81	2450		10			
7788	SCOTT	ANALYST	7566	19/04/87	3000		20			
7369	SMITH	CLERK	7902	17/12/80	800		20			
7876	ADAMS	CLERK	7788	23/05/87	1100		20			
7900	JAMES	CLERK	7698	03/12/81	950		30			
7566	JONES	MANAGER	7839	02/04/81	2975		20			
7698	BLAKE	MANAGER	7839	01/05/81	2850		30			
7902	FORD	ANALYST	7566	03/12/81	3000		20			
7839	KING	PRESIDENT		17/11/81	5000		10			

18 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.

Pero `ep.prono` es nulo en estas filas, por tanto no hace **innerjoin**

Para solucionarlo hay que hacer los dos join exteriores

```
1 select * from emp e left join emppro ep on e.empno=ep.empno
2 left join pro p on ep.prono=p.prono
3* order by ep.empno
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EMPNO	PRONO	HOURS	PRONO	PNAME	LOC	DEPTNO
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1005	12	1005	P5	CHICAGO	30
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30	7499	1004	15	1004	P4	CHICAGO	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1008	8	1008	P8	NEW YORK	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30	7521	1004	10	1004	P4	CHICAGO	30
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1001	16	1001	P1	BOSTON	20
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1008	5	1008	P8	NEW YORK	30
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30	7654	1006	15	1006	P6	LOS ANGELES	30
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30	7844	1005	6	1005	P5	CHICAGO	30
7934	MILLER	CLERK	7782	23/01/82	1300		10	7934	1001	4	1001	P1	BOSTON	20
7900	JAMES	CLERK	7698	03/12/81	950		30							
7876	ADAMS	CLERK	7788	23/05/87	1100		20							
7788	SCOTT	ANALYST	7566	19/04/87	3000		20							
7782	CLARK	MANAGER	7839	09/06/81	2450		10							
7698	BLAKE	MANAGER	7839	01/05/81	2850		30							
7369	SMITH	CLERK	7902	17/12/80	800		20							
7902	FORD	ANALYST	7566	03/12/81	3000		20							
7566	JONES	MANAGER	7839	02/04/81	2975		20							
7839	KING	PRESIDENT		17/11/81	5000		10							

8 filas seleccionadas.

# Join de más de dos tablas exterior

También se puede solucionar alterando el orden (de izquierda a derecha) usando paréntesis.

```
SQL> select deptno, comm, sal, hiredate, job, ename, empno  
2  from emp  
3  order by empno;
```

DEPTNO	COMM	SAL	HIREDATE	JOB	ENAME	EMPNO
20		800	17/12/80	CLERK	SMITH	7369
30	300	1600	20/02/81	SALESMAN	ALLEN	7499
30	500	1250	22/02/81	SALESMAN	WARD	7499
20		2975	02/04/81	MANAGER	JONES	7521
30	1400	1250	28/09/81	SALESMAN	MARTIN	7566
30		2850	01/05/81	MANAGER	BLAKE	7654
10		2450	09/06/81	MANAGER	CLARK	7698
20		3000	19/04/87	ANALYST	SCOTT	7788
10		5000	17/11/81	PRESIDENT	KING	7839
30	0	1500	08/09/81	SALESMAN	TURNER	7844
20		1100	23/05/87	CLERK	ADAMS	7876
30		950	03/12/81	CLERK	JAMES	7900
20		3000	03/12/81	ANALYST	FORD	7902
10		1300	23/01/82	CLERK	MILLER	7934

14 filas seleccionadas.

```
SQL> r  
1  select * from emppro ep join pro p on ep.prono=p.prono  
2* order by empno
```

EMPNO	PRONO	HOURS	PRONO	PNAME	LOC	DEPTNO
7499	1004	15	1004	P4	CHICAGO	30
7499	1005	12	1005	P5	CHICAGO	30
7521	1004	10	1004	P4	CHICAGO	30
7521	1008	8	1008	P8	NEW YORK	30
7654	1001	16	1001	P1	BOSTON	20
7654	1008	5	1008	P8	NEW YORK	30
7654	1006	15	1006	P6	LOS ANGELES	30
7844	1005	6	1005	P5	CHICAGO	30
7934	1001	4	1001	P1	BOSTON	20

9 filas seleccionadas.

A continuación, se hace el join exterior

```
1  select e.deptno,comm,sal,hiredate,job,ename,e.empno,ep.empno,ep.prono,hours,p.prono,pname,loc,p.deptno  
2  from emp e left join (emppro ep join pro p on ep.prono=p.prono) on e.empno=ep.empno  
3* order by e.empno
```

DEPTNO	COMM	SAL	HIREDATE	JOB	ENAME	EMPNO	EMPNO	PRONO	HOURS	PRONO	PNAME	LOC	DEPTNO
20		800	17/12/80	CLERK	SMITH	7369							
30	300	1600	20/02/81	SALESMAN	ALLEN	7499	7499	1004	15	1004	P4	CHICAGO	30
30	300	1600	20/02/81	SALESMAN	ALLEN	7499	7499	1005	12	1005	P5	CHICAGO	30
30	500	1250	22/02/81	SALESMAN	WARD	7521	7521	1004	10	1004	P4	CHICAGO	30
30	500	1250	22/02/81	SALESMAN	WARD	7521	7521	1008	8	1008	P8	NEW YORK	30
20		2975	02/04/81	MANAGER	JONES	7566							
30	1400	1250	28/09/81	SALESMAN	MARTIN	7654	7654	1001	16	1001	P1	BOSTON	20
30	1400	1250	28/09/81	SALESMAN	MARTIN	7654	7654	1008	5	1008	P8	NEW YORK	30
30	1400	1250	28/09/81	SALESMAN	MARTIN	7654	7654	1006	15	1006	P6	LOS ANGELES	30
30		2850	01/05/81	MANAGER	BLAKE	7698							
10		2450	09/06/81	MANAGER	CLARK	7782							
20		3000	19/04/87	ANALYST	SCOTT	7788							
10		5000	17/11/81	PRESIDENT	KING	7839							
30	0	1500	08/09/81	SALESMAN	TURNER	7844	7844	1005	6	1005	P5	CHICAGO	30
20		1100	23/05/87	CLERK	ADAMS	7876							
30		950	03/12/81	CLERK	JAMES	7900							
20		3000	03/12/81	ANALYST	FORD	7902							
10		1300	23/01/82	CLERK	MILLER	7934	7934	1001	4	1001	P1	BOSTON	20

18 filas seleccionadas.

Primero se realiza este join (indicado por los paréntesis)

Ahora el join ya no es asociativo  
El orden importa.

# Ejercicios

1. Para cada departamento mostrar su nombre y cuántos empleados tiene, si no tiene ninguno, indicarlo con un 0.
2. Para cada empleado mostrar las horas que trabajó en proyectos, si no trabajó en ninguno, indicarlo con un 0.
3. Para cada empleado muestra su nombre, el nombre de su jefe, y el departamento para el que trabaja su jefe. Si el empleado no tiene jefe, debe aparecer con nulos en los datos del jefe.
4. Para cada empleado muestra en cuántas ciudades distintas ha trabajado. Si no ha trabajado en ninguna, debe mostrar un cero. Muestra el nombre del empleado.
5. Considerando los proyectos controlados por los departamentos 30 y 40, muestra cuántos empleados distintos han trabajado en cada departamento. Si no han trabajado empleados, debe mostrar un cero. Muestra el nombre del departamento.

# Ejercicios

1. Select dname, count(empno) /\*no count(\*)\*/  
from emp e right join dept d on e.deptno=d.deptno  
group by d.deptno, dname
2. Select ename, coalesce(sum(hours),0)  
from emp e left join emppro ep on e.empno=ep.empno  
group by e.empno, ename
3. Select e.ename, j.ename, d.dname  
from emp e left join emp j on e.mgr=j.empno  
left join dept d on j.deptno=d.deptno
4. Select e.ename, count (distinct loc)  
from emp e left join emppro ep on e.empno=ep.empno  
left join pro p on ep.prono=p.prono  
group by e.empno, e.ename
5. Select d.dname, count (distinct ep.empno)  
from dept d left join pro p on d.deptno=p.deptno  
left join emppro ep on ep.prono=p.prono  
group by d.deptno, d.dname  
having d.deptno=30 or d.deptno=40

# Ejercicios

1. Muestra cuántos empleados subordinados tiene cada jefe, que fueran contratados el mismo año. Si no tiene ninguno debe mostrar un cero.
2. Muestra los empleados que han trabajado en proyectos ubicados en la misma localidad donde está su departamento.
3. Para cada empleado mostrar cuántas veces trabajó en proyectos ubicados en la misma ciudad donde está su departamento.
4. Idem anterior, pero mostrando un cero cuando nunca ocurrió.

1. Select j.ename, count(e.ename)  
from emp e right join emp j on e.mgr=j.empno  
and to\_char(e.hiredate,'YYYY')=  
to\_char(j.hiredate,'YYYY')  
where j.empno in (select mgr from emp)  
group by j.empno, j.ename
2. Select distinct ename, de.loc departamento  
from pro p join emppro ep on ep.prono=p.prono  
join emp e on ep.empno=e.empno  
join dept de on de.deptno=e.deptno  
where p.loc =de.loc
3. Select ename, count(\*)  
from pro p join emppro ep on ep.prono=p.prono  
join emp e on ep.empno=e.empno  
join dept de on de.deptno=e.deptno  
where p.loc =de.loc  
group by e.empno, e.ename
4. Select ename, count(p.prono)  
from emp e join dept de on e.deptno=de.deptno  
left join emppro ep on e.empno=ep.empno  
left join pro p on ep.prono=p.prono  
and de.loc=p.loc  
group by e.empno, ename

# Guión

Introducción

Tablas usadas en los ejemplos

Sentencia Select

Join avanzado

Subconsultas de fila

Expresiones de consulta

# Subconsultas de fila

Las subconsultas de fila tienen más de una expresión en la cláusula select (más de una columna).

```
SELECT ...
FROM ...
WHERE (<expr1>, <expr2>, [,<expr3>]...) { !=|={ ALL | SOME | ANY } | [NOT] IN} (subconsulta)
```

La subconsulta debe tener tantas expresiones como expresiones haya a la izquierda del operador de comparación

```
select *
from emp
where (sal,deptno) =SOME (select max(sal),deptno
                           from emp
                           group by deptno)
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01/05/81	2850		30
7902	FORD	ANALYST	7566	03/12/81	3000		20
7788	SCOTT	ANALYST	7566	19/04/87	3000		20
7839	KING	PRESIDENT		17/11/81	5000		10

# Guión

Introducción

Tablas usadas en los ejemplos

Sentencia Select

Join avanzado

Subconsultas de fila

Expresiones de consulta

# Expresiones de consulta

Una expresión de consulta es la utilización de sentencias select en las cláusulas select o from.

En la cláusula select sólo puede ser una select escalar.

Las expresiones de consulta en la cláusula from deben ser autosuficientes (no puede estar correlacionada).

```
SQL> select empno, ename, (select max(sal) from emp)-sal "Diferencia con max sal"
  2  from emp;
```

EMPNO	ENAME	Diferencia con max sal
7369	SMITH	4200
7499	ALLEN	3400
7521	WARD	3750
7566	JONES	2025
7654	MARTIN	3750
7698	BLAKE	2150
7782	CLARK	2550
7788	SCOTT	2000
7839	KING	0
7844	TURNER	3500
7876	ADAMS	3900
7900	JAMES	4050
7902	FORD	2000
7934	MILLER	3700

14 filas seleccionadas.

```
1  select ename, e.job, maximo-sal "Diferencia max de trabajo"
  2  from emp e join (select job, max(sal) maximo
  3                      from emp
  4                     group by job) m
  5*          on e.job=m.job
```

ENAME	JOB	Diferencia max de trabajo
SCOTT	ANALYST	0
FORD	ANALYST	0
SMITH	CLERK	500
ADAMS	CLERK	200
JAMES	CLERK	350
MILLER	CLERK	0
JONES	MANAGER	0
BLAKE	MANAGER	125
CLARK	MANAGER	525
KING	PRESIDENT	0
ALLEN	SALESMAN	0
WARD	SALESMAN	350
MARTIN	SALESMAN	350
TURNER	SALESMAN	100

14 filas seleccionadas.

# Ejercicios

1. Muestra los empleados que ganan más del salario medio de la empresa. Muestra también dicho salario medio.
2. Muestra los empleados que ganan más del salario medio de su departamento. Muestra también dicho salario medio.
3. Muestra de cada proyecto, el empleado que más horas trabaja.
4. El empleado que ha trabajado más horas en cada ciudad. Muestra los nombres del empleado y de la ciudad.
5. Muestra la media de horas dedicadas a proyectos (sumando las horas de todos los empleados que trabajan en el proyecto).
6. Muestra los proyectos cuya suma de horas es mayor que la cantidad de horas media dedicadas a proyectos. Debes mostrar dicha suma.

# Ejercicios

1. select ename, (select avg(sal) from emp)  
from emp  
where sal >(select avg(sal) from emp)
2. select ename, sal, deptno, salario\_medio  
from emp e join (select deptno depart, avg(sal) salario\_medio  
from emp  
group by deptno)  
on deptno=depart  
where sal>salario\_medio
3. select prono, ename, hours  
from emp natural join emppro  
where (prono,hours)=SOME(select prono, max(hours)  
from emppro  
group by prono)
4. select sum(hours) horas, loc, e.ename  
from emppro ep join pro p on p.prono=ep.prono join emp e on e.empno=ep.empno  
group by ep.empno,e.ename, p.loc  
having sum(hours) >=ALL (select sum(hours)  
from emppro ep1 join pro p1 on ep1.prono=p1.prono  
where loc=p.loc  
group by ep1.empno)
- 4 bis. SELECT e.empno, e.ename, loc, sum(hours)  
FROM emp e JOIN emppro ep ON e.empno=ep.empno  
JOIN pro p ON ep.prono=p.prono  
GROUP BY empno, loc  
HAVING (loc, sum(hours)) in  
(SELECT loc , max (sumahoras)  
FROM  
(SELECT empno, loc, sum(hours) sumahoras  
FROM emppro ep2 JOIN pro p2 ON ep2.prono=p2.prono  
GROUP BY empno, loc  
ORDER BY loc) horas\_ec  
GROUP BY loc)
5. select avg(horas\_medias)  
from (select sum(hours) horas\_medias  
from emppro  
group by prono)
6. select pname, sum(hours)  
from pro p join emppro ep on ep.prono=p.prono  
group by p.prono, pname  
having sum(hours)> (select avg(horas\_medias)  
from (select sum(hours) horas\_medias  
from emppro  
group by prono))

# Ejercicios

1. Para cada empleado muestra su nombre y cuál es la diferencia de sus ingresos con respecto a los ingresos del empleado que más gana de su departamento.
2. Mostrar para cada departamento la diferencia entre la suma de los salarios de todos sus empleados y la del departamento con la suma más alta.
3. Muestra para cada empleado en qué ciudades trabajó.
4. De entre los empleados que han trabajado en proyectos, muestra para cada uno en qué ciudades no trabajó.
5. Muestra para cada empleado de la empresa en qué ciudades no trabajó. Muestra el nombre del empleado.
6. Muestra las ciudades donde trabajó un empleado con un texto ‘SI’, y aquellas ciudades donde no trabajó con un texto ‘NO’. Ten en cuenta todos los empleados (los que están en emp y no en emppro también).
7. Muestra parejas de empleados que tienen relación jefe-empleado directa.
8. Muestra las parejas de empleados que no tienen relación jefe-empleado directa.

# Ejercicios

1. select ename, maximo-coalesce(sal+comm, sal)  
from emp e join (select max(coalesce(sal+comm, sal)) maximo, deptno  
from emp  
group by deptno) s  
on e.deptno=s.deptno
2. select deptno, (select distinct sum(sal) from emp  
group by deptno  
having sum(sal)>=ALL(select sum(sal)  
from emp  
group by deptno))-sum(sal) diferencia  
from emp  
group by deptno
3. select distinct empno, loc  
from emppro natural join pro
4. select distinct empno,loc  
from emppro ep cross join pro p  
where (empno,loc) not in  
(select empno, loc  
from emppro natural join pro)
5. select distinct e.empno, ename,loc  
from emp e cross join pro p  
where (e.empno,loc) not in (select empno, loc  
from emppro join pro on ep.prono=p.prono)
6. select ename, loc, 'SI' "¿Trabajó?"  
from emp natural join emppro join pro using(prono)  
union  
select ename, loc, 'NO'  
from (select empno,loc  
from emp cross join pro  
minus  
select empno, loc  
from emppro natural join pro)  
natural join emp
7. select e.empno,e.ename,j.empno, j.ename  
from emp e join emp j on e.mgr=j.empno
8. select e.empno,e.ename,j.empno,j.ename  
from emp e cross join emp j  
where e.empno!=j.empno  
and (e.empno,j.empno) not in  
(select e.empno, j.empno  
from emp e join emp j on e.mgr=j.empno)  
union  
select j.empno, e.empno  
from emp e join emp j on e.mgr=j.empno)