

The background is a light blue gradient with several realistic water droplets of various sizes scattered across it. The droplets have highlights and shadows, giving them a 3D appearance.

# SENTENCIAS JOIN

Presentado por: Daniel Alejandro  
Guerrero Suárez

# Función (+)

La forma de conseguir que salgan todos los registros de una tabla aunque no estén relacionados con las de otra es realizar una asociación lateral o unión externa (también llamada *outer join*).

En esas asociaciones, el signo (+) indica que se desean todos los registros de la tabla estén o no relacionados.

```
SELECT tabla1.columna1, tabla1.columna2,....  
        tabla2.columna1, tabla2.columna2,...  
FROM tabla1, tabla2  
WHERE tabla1.columnaRelacionada (+)=tabla2.columnaRelacionada
```

Eso obtiene los registros relacionados entre las tablas y además los registros no relacionados de la tabla2.

# Función (+)

17. Seleccionar los nombres completos y cargos de todos los empleados con sus subalternos  
 mtengan o no tengan subalternos.

```
select j.first_name || ' ' || j.last_name "Jefe", s.first_name || ' ' || s.last_name
```

```
from s_emp j, s_emp s
```

where `j.id=s.manager_id(+)`;

41 Rows Selected

```

Ejecutar Línea de Comandos SQL
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> select j.first_name||' '||j.last_name "Jefe", s.first_name||' '||s.last_name
e from s_emp j, s_emp s where j.id=s.manager_id(+);

Jefe                                S.FIRST_NAME||' '||S.LAST_NAME
-----
Carmen Velasquez                    LaDoris Ngao
Carmen Velasquez                    Midori Nagayama
Carmen Velasquez                    Mark Quick-To-See
Carmen Velasquez                    Audry Ropeburn
LaDoris Ngao                        Molly Urguhart
LaDoris Ngao                        Roberta Menchu
LaDoris Ngao                        Ben Biri
LaDoris Ngao                        Antoinette Catchpole
LaDoris Ngao                        Marta Havel
Midori Nagayama                     Colin Magee
Midori Nagayama                     Henry Giljum

Jefe                                S.FIRST_NAME||' '||S.LAST_NAME
-----
Midori Nagayama                     Yasmin Sedeghi
Midori Nagayama                     Mai Nguyen
Midori Nagayama                     Andre Dumas
Molly Urguhart                      Elena Maduro
Molly Urguhart                      George Smith
Roberta Menchu                      Akira Nozaki
Roberta Menchu                      Vikram Patel
Ben Biri                            Chad Newman
Ben Biri                            Alexander Markarian
Antoinette Catchpole               Eddie Chang
Antoinette Catchpole               Radha Patel

Jefe                                S.FIRST_NAME||' '||S.LAST_NAME
-----
Marta Havel                         Bela Dancs
Marta Havel                         Sylvie Schwartz
Andre Dumas
Alexander Markarian
Akira Nozaki
Mark Quick-To-See
Radha Patel
Sylvie Schwartz
Bela Dancs
Audry Ropeburn
Henry Giljum

Jefe                                S.FIRST_NAME||' '||S.LAST_NAME
-----
Vikram Patel
George Smith
Mai Nguyen
Elena Maduro
Yasmin Sedeghi
Eddie Chang
Chad Newman
Colin Magee
SQL>

```

# Función Minus

Con **MINUS** también se combinan dos consultas SELECT de forma que aparecerán los registros del primer SELECT que no estén presentes en el segundo.

```
(SELECT....  
....  
UNION  
SELECT....  
...  
)  
MINUS  
SELECT.... /* Primero se hace la unión y luego la diferencia*/
```

Se podrían hacer varias combinaciones anidadas (una unión cuyo resultado se intersectara con otro SELECT por ejemplo), en ese caso es conveniente utilizar paréntesis para indicar qué combinación se hace primero:



# Función Minus

20. Seleccionar los productos que no están en el inventario.

```
select distinct p.id, p.name  
from s_product p, s_inventory i  
where p.id=i.product_id(+)
```

minus

```
select distinct p.id, p.name  
from s_product p, s_inventory i  
where p.id=i.product_id;
```

No Rows Selected

# Función JOIN (Asociar Tablas)

Es una *unión* entre dos tablas por medio de un campo.

JOIN se utiliza para consultar datos de más de una tabla

La condición de JOIN se escribe en la cláusula WHERE y/o cláusula FROM.

Si existen columnas con el mismo nombre en las tablas seleccionadas, se deberán nombrar los campos

# Ejemplo Oracle

Tabla emp

EMPNO	ENAME	JOB	MGR	DEPTNO
7369	HETFIELD	CLERK		
7499	ALLEN	SALESMAN	7698	30
7521	WARD	SALESMAN	7698	30
7566	JONES	MANAGER	7839	20
7654	MARTIN	SALESMAN	7698	30
7698	BLAKE	MANAGER	7839	30
7782	CLARK	MANAGER	7839	10
7788	SCOTT	ANALYST	7566	20
7839	KING	PRESIDENT		
7844	TURNER	SALESMAN	7698	30
7876	ADAMS	CLERK		
7900	JAMES	CLERK		
7902	FORD	ANALYST	7566	20
7934	MILLER	CLERK		
9999	JUAN			
2222	HAMMET	CLERK		10
7777	ULRICH	CLERK		10
1234	HETFIELD	CLERK		10
4321	TRUJILLO	CLERK		10
4444	ARAYA	CLERK		10
5555	LOMBARDO	CLERK		10
6666	ANSELMO	CLERK		10
9421	RAMIREZ	CLERK		10
9090	GONZALEZ	CLERK		10

24 rows selected.

Tabla dept

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

4 rows selected.

# Función JOIN (EquiJoin)

Este tipo de Join, tiene involucrado el igual (=) como comparador entre dos campos de dos tablas distintas. Devolverá los registros que cumplan con la condición de ser iguales en ambas tablas.

```
SQL> select e.empno, e.ename, e.job, d.dname  
2  from   emp e,  
3         dept d  
4  where  e.deptno = d.deptno;
```

EMPNO	ENAME	JOB	DNAME
7782	CLARK	MANAGER	ACCOUNTING
2222	HAMMET	CLERK	ACCOUNTING
9090	GONZALEZ	CLERK	ACCOUNTING
9421	RAMIREZ	CLERK	ACCOUNTING
6666	ANSELMO	CLERK	ACCOUNTING
5555	LOMBARDO	CLERK	ACCOUNTING
4444	ARAYA	CLERK	ACCOUNTING
4321	TRUJILLO	CLERK	ACCOUNTING
1234	HETFIELD	CLERK	ACCOUNTING
7777	ULRICH	CLERK	ACCOUNTING
7902	FORD	ANALYST	RESEARCH
7788	SCOTT	ANALYST	RESEARCH
7566	JONES	MANAGER	RESEARCH
7654	MARTIN	SALESMAN	SALES
7521	WARD	SALESMAN	SALES
7499	ALLEN	SALESMAN	SALES
7698	BLAKE	MANAGER	SALES
7844	TURNER	SALESMAN	SALES

18 rows selected.



# Función JOIN (Self Join)

Es un join de una tabla consigo misma. Se realiza de manera similar a un join normal; la diferencia sustancial, reside en que será comparada y unida por algún campo de la misma tabla.

Ejemplo: Manager\_ID

```
SQL> select emp.empno, emp.ename, emp.job, mgr.ename, mgr.job  
2   from   emp emp,  
3         emp mgr  
4   where  emp.mgr = mgr.empno;
```

EMPNO	ENAME	JOB	ENAME	JOB
7788	SCOTT	ANALYST	JONES	MANAGER
7902	FORD	ANALYST	JONES	MANAGER
7844	TURNER	SALESMAN	BLAKE	MANAGER
7499	ALLEN	SALESMAN	BLAKE	MANAGER
7654	MARTIN	SALESMAN	BLAKE	MANAGER
7521	WARD	SALESMAN	BLAKE	MANAGER
7782	CLARK	MANAGER	KING	PRESIDENT
7566	JONES	MANAGER	KING	PRESIDENT
7698	BLAKE	MANAGER	KING	PRESIDENT

9 rows selected.

# Función JOIN (Producto Cartesiano)

El producto cartesiano, se genera cuando se incluyen dos o más tablas en la parte *From* y no se incluye alguna condición que las una en un join.

Es una de las peores formas de acceder a la información porque por cada registro de una tabla se ponen todos los registros de la otra. Así, si se tienen una tabla de 100 registros y se hace un producto cartesiano con una 10,000, se tendrán 1,000,000 registros.

```
SQL> select e.empno, e.ename, e.job, d.dname
2  from emp e,
3      dept d;
EMPNO ENAME      JOB      DNAME
-----
7369  HETFIELD    CLERK    ACCOUNTING
7499  ALLEN        SALESMAN ACCOUNTING
7521  WARD         SALESMAN ACCOUNTING
7566  JONES        MANAGER  ACCOUNTING
7654  MARTIN       SALESMAN ACCOUNTING
7698  BLAKE        MANAGER  ACCOUNTING
7782  CLARK        MANAGER  ACCOUNTING
7788  SCOTT        ANALYST  ACCOUNTING
7839  KING         PRESIDENT ACCOUNTING
7844  TURNER       SALESMAN ACCOUNTING

...

7369  HETFIELD    CLERK    RESEARCH
7499  ALLEN        SALESMAN RESEARCH
7521  WARD         SALESMAN RESEARCH
7566  JONES        MANAGER  RESEARCH
7654  MARTIN       SALESMAN RESEARCH
7698  BLAKE        MANAGER  RESEARCH
7782  CLARK        MANAGER  RESEARCH
7788  SCOTT        ANALYST  RESEARCH
7839  KING         PRESIDENT RESEARCH
7844  TURNER       SALESMAN RESEARCH

...

7369  HETFIELD    CLERK    SALES
7499  ALLEN        SALESMAN SALES
7521  WARD         SALESMAN SALES
7566  JONES        MANAGER  SALES
7654  MARTIN       SALESMAN SALES
7698  BLAKE        MANAGER  SALES
7782  CLARK        MANAGER  SALES
```

# Función JOIN (Inner Join) (simple Join)

Es la unión entre dos tablas que nos muestra nada más aquellos registros que cumplen con la condición del join.

```
SQL> SELECT e.empno, e.ename, e.job, d.dname  
FROM emp e JOIN dept d ON e.deptno = d.deptno;
```

```
SQL> select e.empno, e.ename, e.job, d.dname  
2 from emp e,  
3 dept d  
4 where e.deptno = d.deptno;
```

EMPNO	ENAME	JOB	DNAME
7782	CLARK	MANAGER	ACCOUNTING
2222	HAMMET	CLERK	ACCOUNTING
9090	GONZALEZ	CLERK	ACCOUNTING
9421	RAMIREZ	CLERK	ACCOUNTING
6666	ANSELMO	CLERK	ACCOUNTING
5555	LOMBARDO	CLERK	ACCOUNTING
4444	ARAYA	CLERK	ACCOUNTING
4321	TRUJILLO	CLERK	ACCOUNTING
1234	HETFIELD	CLERK	ACCOUNTING
7777	ULRICH	CLERK	ACCOUNTING
7902	FORD	ANALYST	RESEARCH
7788	SCOTT	ANALYST	RESEARCH
7566	JONES	MANAGER	RESEARCH
7654	MARTIN	SALESMAN	SALES
7521	WARD	SALESMAN	SALES
7499	ALLEN	SALESMAN	SALES
7698	BLAKE	MANAGER	SALES
7844	TURNER	SALESMAN	SALES

18 rows selected.

# • Función JOIN (Outer Join)

Este tipo de join, nos devuelve los registros que cumplen

- con la condición de join,
- más aquellos registros de una de las dos tablas involucradas que no cumplieron con la condición.

Ejemplos claros de estos podrían ser:

- Mostrar todos los departamentos de la empresa con sus empleados, no importa si no tienen empleados.
- Listar los empleados de la empresa con su departamento asignado, aún cuando no tengan departamento asignado.

Left Outer Join - Right Outer Join - Full Outer Join



# Función JOIN

## (Left outer Join)

## (Left Join)

Es aquel en el que se unen dos tablas A y B. En esta unión se incluyen todos los registros de A aunque no satisfagan la condición.

Para hacer esto, se puede usar el *Left Outer Join* o el operador (+) en todas las columnas de B en la parte *Where*.

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

EMPNO	ENAME	JOB	DNAME
7369	HETFIELD	CLERK	
7499	ALLEN	SALESMAN	SALES
7521	WARD	SALESMAN	SALES
7566	JONES	MANAGER	RESEARCH
7654	MARTIN	SALESMAN	SALES
7698	BLAKE	MANAGER	SALES
7782	CLARK	MANAGER	ACCOUNTING
7788	SCOTT	ANALYST	RESEARCH
7839	KING	PRESIDENT	
7844	TURNER	SALESMAN	SALES
7876	ADAMS	CLERK	
7900	JAMES	CLERK	
7902	FORD	ANALYST	RESEARCH
7934	MILLER	CLERK	
9999	JUAN		
2222	HAMMET	CLERK	ACCOUNTING
7777	ULRICH	CLERK	ACCOUNTING
1234	HETFIELD	CLERK	ACCOUNTING
4321	TRUJILLO	CLERK	ACCOUNTING
4444	ARAYA	CLERK	ACCOUNTING
5555	LOMBARDO	CLERK	ACCOUNTING
6666	ANSELMO	CLERK	ACCOUNTING
9421	RAMIREZ	CLERK	ACCOUNTING
9090	GONZALEZ	CLERK	ACCOUNTING

24 rows selected.

```
SQL> select e.empno, e.ename, e.job, d.dname
2  from   emp e,
3         dept d
4  where  e.deptno = d.deptno (+);
```

# Función JOIN (Right Outer Join) (Right Join)

Es aquel en el que se unen dos tablas A y B. En esta unión se incluyen todos los registros de B aunque no satisfagan la condición.

Para hacer esto, se puede usar el *Right Outer Join* o el operador (+) en todas las columnas de A en la parte *Where*.

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

EMPNO	ENAME	JOB	DNAME
7782	CLARK	MANAGER	ACCOUNTING
2222	HAMMET	CLERK	ACCOUNTING
9090	GONZALEZ	CLERK	ACCOUNTING
9421	RAMIREZ	CLERK	ACCOUNTING
6666	ANSELMO	CLERK	ACCOUNTING
5555	LOMBARDO	CLERK	ACCOUNTING
4444	ARAYA	CLERK	ACCOUNTING
4321	TRUJILLO	CLERK	ACCOUNTING
1234	HETFIELD	CLERK	ACCOUNTING
7777	ULRICH	CLERK	ACCOUNTING
7902	FORD	ANALYST	RESEARCH
7788	SCOTT	ANALYST	RESEARCH
7566	JONES	MANAGER	RESEARCH
7654	MARTIN	SALESMAN	SALES
7521	WARD	SALESMAN	SALES
7499	ALLEN	SALESMAN	SALES
7698	BLAKE	MANAGER	SALES
7844	TURNER	SALESMAN	SALES
			OPERATIONS

19 rows selected.

```
SQL> select e.empno, e.ename, e.job, d.dname  
2   from   emp e,  
3          dept d  
4  where  e.deptno (+) = d.deptno;
```

# Función JOIN (Full Outer Join) (Full Join)

Finalmente, este join es la combinación del right outer join y el left outer join; mostrándonos la información de ambas tablas unidas A y B no importando que haya registros que no satisfagan la condición.

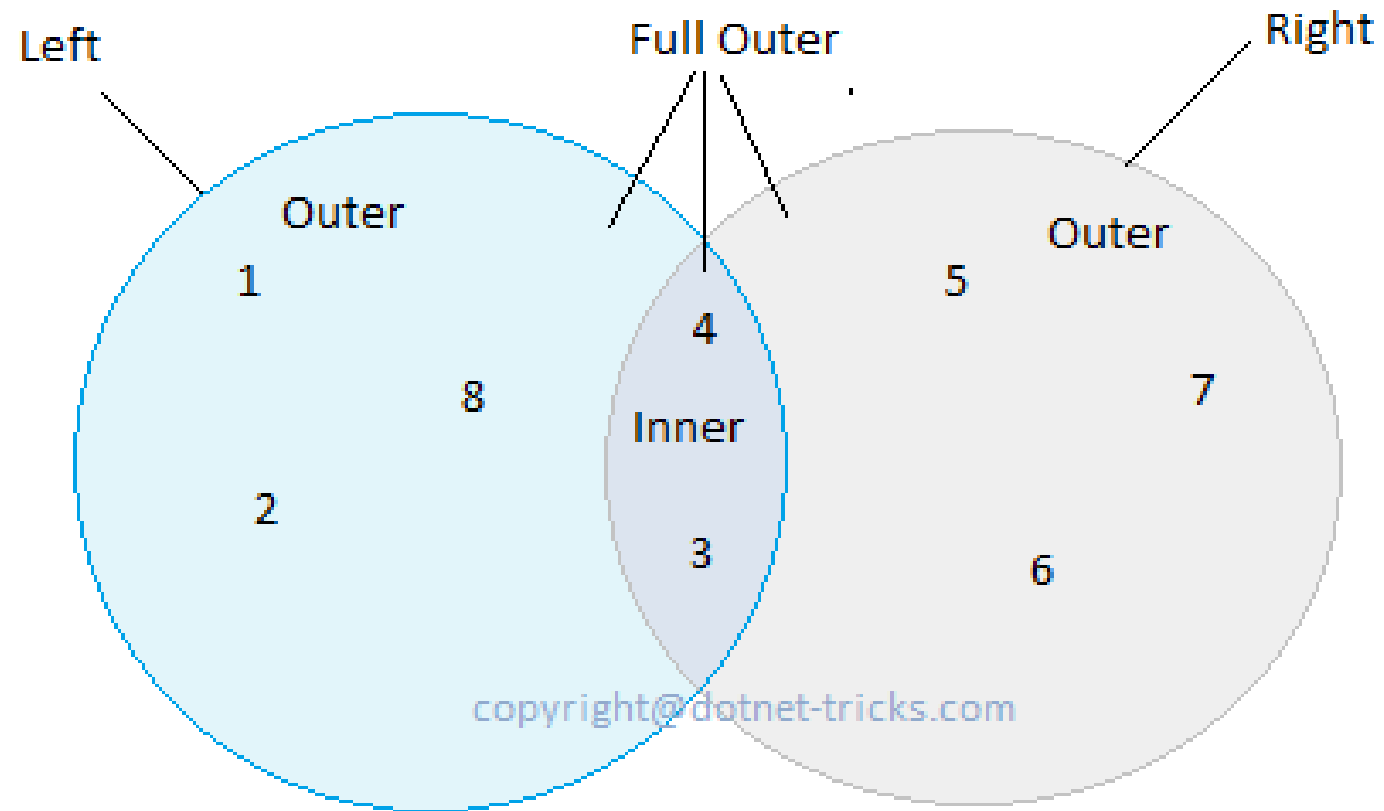
Para realizar esto, se usa la cláusula *Full Outer Join* o un *Union* con los dos queries del operador (+)

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

EMPNO	ENAME	JOB	DNAME
7369	HETFIELD	CLERK	
7499	ALLEN	SALESMAN	SALES
7521	WARD	SALESMAN	SALES
7566	JONES	MANAGER	RESEARCH
7654	MARTIN	SALESMAN	SALES
7698	BLAKE	MANAGER	SALES
7782	CLARK	MANAGER	ACCOUNTING
7788	SCOTT	ANALYST	RESEARCH
7839	KING	PRESIDENT	
7844	TURNER	SALESMAN	SALES
7876	ADAMS	CLERK	
7900	JAMES	CLERK	
7902	FORD	ANALYST	RESEARCH
7934	MILLER	CLERK	
9999	JUAN		
2222	HAMMET	CLERK	ACCOUNTING
7777	ULRICH	CLERK	ACCOUNTING
1234	HETFIELD	CLERK	ACCOUNTING
4321	TRUJILLO	CLERK	ACCOUNTING
4444	ARAYA	CLERK	ACCOUNTING
5555	LOMBARDO	CLERK	ACCOUNTING
6666	ANSELMO	CLERK	ACCOUNTING
9421	RAMIREZ	CLERK	ACCOUNTING
9090	GONZALEZ	CLERK	ACCOUNTING OPERATIONS

25 rows selected.

```
SQL> select e.empno, e.ename, e.job, d.dname  
2  from   emp e,  
3         dept d  
4  where  e.deptno (+) = d.deptno  
5  union  
6  select e.empno, e.ename, e.job, d.dname  
7  from   emp e,  
8         dept d  
9  where  e.deptno = d.deptno (+);
```



Inner Join Result : (4,3)

Left Join Result : (1,2,8,4,3)

Right Join Result : (5,6,7,4,3)

Full Outer Join Result : (1,2,8,4,3,5,6,7)



# Función JOIN (AntiJoin)

Este tipo de join es parecido a la operación *Minus* entre queries.

Nos devuelve una cantidad de registros de la parte izquierda del predicado que no pertenecen a la parte derecha. Esto se logra con un *Not In*.

```
SQL> select empno, ename, job
2  from emp
3  where deptno not in (select deptno
4                        from dept
5                        where dname = 'ACCOUNTING');

```

EMPNO	ENAME	JOB
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7788	SCOTT	ANALYST
7844	TURNER	SALESMAN
7902	FORD	ANALYST

8 rows selected.

# Función JOIN (SemiJoin)

Este tipo de join, muestra los registros que coinciden por medio de un subquery que usa la palabra *Exists*, sin duplicar registros de la parte izquierda del mismo, cuando múltiples registros del lado derecho satisfagan la condición del subquery.

```
SQL> select *
      2  from dept
      3  where exists (select *
      4                  from emp
      5                  where dept.deptno = emp.deptno
      6                  and emp.job      = 'CLERK');
      DEPTNO DNAME          LOC
-----
          10 ACCOUNTING    NEW YORK
1 row selected.
```