

# 2

## Visualizando datos de múltiples tablas

# Objetivos

**Tras completar esta lección seremos capaces de hacer lo siguiente:**

- **Escribir sentencias SELECT para acceder a datos de más de una tabla usando Equal-Join y Outer-Join.**
- **Join de una tabla consigo misma.**

# Obtener datos de múltiples tables

**EMP**

EMPNO	ENAME	...	DEPTNO
-----	-----	...	-----
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

**DEPT**

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



EMPNO	DEPTNO	LOC
-----	-----	-----
7839	10	NEW YORK
7698	30	CHICAGO
7782	10	NEW YORK
7566	20	DALLAS
7654	30	CHICAGO
7499	30	CHICAGO
...		
14 rows selected.		

# Qué es un Join?

Se usa para consultar datos procedentes de más de una tabla.

```
SELECT    table1.column, table2.column
FROM      table1, table2
WHERE     table1.column1 = table2.column2;
```

- Se escribe la condición del join en la cláusula WHERE .
- Cuando el mismo nombre de columna aparece en más de una tabla se pone el nombre de tabla delante del nombre de columna .

# Producto cartesiano

- **Se forma cuando:**
  - Se omite la condición del JOIN.
  - La condición del JOIN es inválida.
  - Todas las filas de la primera tabla se unen con todas las filas de la segunda tabla.
- **Para evitar el producto cartesiano siempre incluiremos una condición válida de join en la cláusula WHERE.**

# Generación de un producto cartesiano

**EMP (14 rows)**

EMPNO	ENAME	...	DEPTNO
-----	-----	...	-----
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

**DEPT (4 rows)**

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

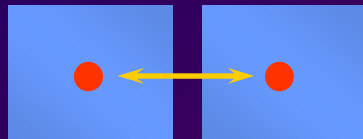


**“Cartesian  
product:  
14\*4=56 rows”** →

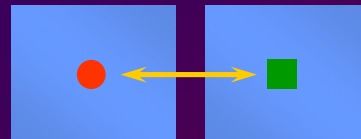
ENAME	DNAME
-----	-----
KING	ACCOUNTING
BLAKE	ACCOUNTING
...	
KING	RESEARCH
BLAKE	RESEARCH
...	
56 rows selected.	

# Tipos de Join

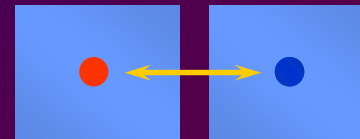
**Equijoin**



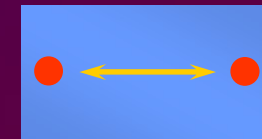
**Non-equijoin**



**Outer join**



**Self join**



# Equijoin

**EMP**

EMPNO	ENAME	DEPTNO
-----	-----	-----
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		
14 rows selected.		

**DEPT**

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		
14 rows selected.		

**Foreign key**

**Primary key**



# Recuperación de registros con Equijoins

```
SQL> SELECT  emp.empno, emp.ename, emp.deptno,  
2           dept.deptno, dept.loc  
3 FROM      emp, dept  
4 WHERE     emp.deptno=dept.deptno;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS
...				

14 rows selected.

## Nombres ambiguos de columnas

- Se usa el nombre de tabla como prefijo para cualificar nombres de columnas que aparecen en múltiples tablas.
- Se gana eficiencia usando nombres de tabla como prefijos.
- Es conveniente distinguir columnas que tienen nombres idénticos pero que residen en diferentes tablas usando alias de columna.

## Definiremos condiciones adicionales de búsqueda usando el operador AND

### EMP

EMPNO	ENAME	DEPTNO
-----	-----	-----
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		
14 rows selected.		

### DEPT

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		
14 rows selected.		

# Uso de Alias de Tablas

Se pueden simplificar las consultas usando alias de tablas.

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
2          dept.deptno, dept.loc  
3 FROM      emp, dept  
4 WHERE     emp.deptno=dept.deptno;
```

```
SQL> SELECT e.empno, e.ename, e.deptno,  
2          d.deptno, d.loc  
3 FROM      emp e, dept d  
4 WHERE     e.deptno=d.deptno;
```

# Unir más de dos tablas

## CUSTOMER

NAME	CUSTID
-----	-----
JOCKSPORTS	100
TKB SPORT SHOP	101
VOLLYRITE	102
JUST TENNIS	103
K+T SPORTS	105
SHAPE UP	106
WOMENS SPORTS	107
...	...
9 rows selected.	

## ORD

CUSTID	ORDID
-----	-----
101	610
102	611
104	612
106	601
102	602
...	...
21 rows selected.	

## ITEM

ORDID	ITEMID
-----	-----
610	3
611	1
612	1
601	1
602	1
...	...
64 rows selected.	

# Non-Equijoins

**EMP**

EMPNO	ENAME	SAL
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
...		
14 rows selected.		

**SALGRADE**

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

El salario en la tabla EMP  
está comprendido entre LOSAL  
y HISAL en la tabla SALGRADE

# Recuperación de Registros con Non-Equijoins

```
SQL>  SELECT    e.ename, e.sal, s.grade
      2  FROM      emp e, salgrade s
      3  WHERE     e.sal
      4  BETWEEN   s.losal AND s.hisal;
```

ENAME	SAL	GRADE
-----	-----	-----
JAMES	950	1
SMITH	800	1
ADAMS	1100	1
...		

14 rows selected.

# Outer Joins

**EMP**

ENAME	DEPTNO
-----	-----
KING	10
BLAKE	30
CLARK	10
JONES	20
...	

**DEPT**

DEPTNO	DNAME
-----	-----
10	ACCOUNTING
30	SALES
10	ACCOUNTING
20	RESEARCH
...	
40	OPERATIONS



**No hay empleados en el  
Departamento OPERATIONS**



# Outer Joins

- Se pueden usar también para obtener filas que no cumplen la condición del JOIN.
- El operador Outer join es el signo (+).

```
SELECT table1.column, table2.column  
FROM   table1, table2  
WHERE  table1.column (+) = table2.column;
```

```
SELECT table1.column, table2.column  
FROM   table1, table2  
WHERE  table1.column = table2.column (+);
```

# Ejemplo de Outer Joins

```
SQL> SELECT    e.ename, d.deptno, d.dname
  2  FROM      emp e, dept d
  3  WHERE     e.deptno(+) = d.deptno
  4  ORDER BY  e.deptno;
```

ENAME	DEPTNO	DNAME
KING	10	ACCOUNTING
CLARK	10	ACCOUNTING
...		
	40	OPERATIONS

15 rows selected.

# Self Joins

**EMP (WORKER)**

EMPNO	ENAME	MGR
-----	-----	-----
7839	KING	
7698	BLAKE	7839
7782	CLARK	7839
7566	JONES	7839
7654	MARTIN	7698
7499	ALLEN	7698

**EMP (MANAGER)**

EMPNO	ENAME
-----	-----
7839	KING
7839	KING
7839	KING
7698	BLAKE
7698	BLAKE



**“MGR en la tabla WORKER es igual a EMPNO en la tabla MANAGER”**

# Join de una Tabla consigo misma

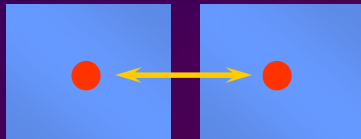
```
SQL> SELECT worker.ename || ' works for ' || manager.ename  
2   FROM    emp worker, emp manager  
3   WHERE   worker.mgr = manager.empno;
```

```
WORKER.ENAME | 'WORKSFOR' | MANAG  
-----  
BLAKE works for KING  
CLARK works for KING  
JONES works for KING  
MARTIN works for BLAKE  
...  
13 rows selected.
```

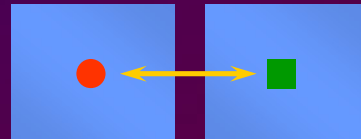
# Resumen

```
SELECT    table1.column, table2.column  
FROM      table1, table2  
WHERE     table1.column1 = table2.column2;
```

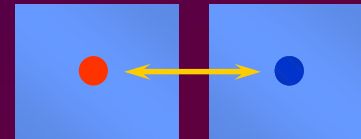
Equijoin



Non-equijoin



Outer join



Self join

