

Consultas multitable:  
mostrar datos  
a partir de varias tablas

JOINS

# Objetivos

- **Al completar esta lección, deberías ser capaz de hacer lo siguiente:**
  - **Escribir sentencias SELECT para acceder a los datos desde una o más tablas.**
  - **Visualizar datos que no se cumplirían normalmente con una condición de join usando outer joins (reuniones externas).**
  - **Combinar (Join) una tabla consigo misma.**

# Obtención de Datos de Múltiples Tablas

**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?

- Utiliza el join para consultar datos de más de una tabla. Es una reunión o composición de las filas de una tabla con las de la otra.
- Dos formas diferentes de realizar lo mismo.

```
SELECT    table1.column, table2.column2
FROM      table1 [INNER] JOIN table2
          ON table1.Clave_Ajena = table2.Clave_primaria;
```

# Tipos de Join

- **Inner join (join bàsic)**
- **Outer join**
  - Left join
  - Right join
  - Full join

## Casos Especiales

- Self join
- Producto cartesiano con restricció

# ¿Qué es un join?

**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 filas mediante joins

```
SELECT    emp.empno, emp.ename, emp.deptno,  
          dept.deptno, dept.loc  
FROM      emp JOIN dept ON 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 de Columna Ambiguos

- Cuando se está realizando una consulta en la que aparecen dos campos que se llaman igual, el sistema gestor da un error de “ambigüedad”.

```
Select empno, ename, deptno, dname  
From emp JOIN dept ON detpno=deptno;
```

- En este caso, el uso del nombre de la tabla como prefijo del campo es obligatorio.

```
Select empno, ename, dept.deptno, dname  
From emp JOIN dept ON emp.detpno=dept.deptno;
```

Así se distinguen las columnas que tienen nombres idénticos pero pertenecen a diferentes tablas.



# Uso de Alias de Tabla

- **Ayudan a simplificar las consultas**

```
SELECT emp.empno, emp.ename, emp.deptno,  
       dept.deptno, dept.loc  
FROM   emp JOIN dept ON emp.deptno=dept.deptno;
```

```
SELECT e.empno, e.ename, e.deptno,  
       d.deptno, d.loc  
FROM   emp e JOIN dept d ON e.deptno=d.deptno;
```

# Outer Joins

EMP		DEPT	
ENAME	DEPTNO	DEPTNO	DNAME
-----	-----	-----	-----
KING	10	10	ACCOUNTING
BLAKE	30	30	SALES
CLARK	10	20	RESEARCH
JONES	20	40	OPERATIONS
...			

**No hay empleados en el departamento OPERATIONS**

# Outer Joins

Se utiliza para ver todas las filas, tanto las que cumplen la condición de join como las que no.

- Left join
- Right join
- Full join

```
SELECT table.column, table.column  
FROM   table1 LEFT [outer] JOIN table2  
       ON table1.clave_ajena = table2.clave_primaria;
```

```
SELECT table.column, table.column  
FROM   table1 RIGHT [outer] JOIN table2  
       ON table1.clave_ajena = table2.clave_primaria;
```

# Uso de Outer Joins

```
SELECT e.ename, d.deptno, d.dname
FROM   emp e RIGHT JOIN dept d ON e.deptno=d.deptno
ORDER BY      e.deptno;
```

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

15 rows selected.

Fuerzo a que se muestre el departamento 40 a pesar de no cumplir la condicion de join

# Self Joins:

Cuando la clave ajena hace referencia a la propia tabla

EMP			JEFE (EMP)		
EMPNO	ENAME...	MGR	EMPNO	ENAME...	MGR
-----	-----	----	-----	-----	----
7839	KING		7839	KING	
7698	BLAKE	7839	7698	BLAKE	7839
7782	CLARK	7839	7782	CLARK	7839
7566	JONES	7839	7566	JONES	7839
7654	MARTIN	7698	7654	MARTIN	7698
7499	ALLEN	7698	7499	ALLEN	7698

“MGR en la tabla EMP es igual a  
EMPNO en la tabla JEFE”

# Self join

Se combina la propia tabla consigo misma

```
SELECT emp.ename Empleat, jefe.ename Cap  
FROM    emp JOIN emp jefe ON emp.mgr = jefe.empno;
```

Empleat	Cap
-----	
BLAKE	KING
CLARK	KING
JONES	KING
MARTIN	BLAKE
...	
13 rows selected.	

# Producto cartesiano con restricción

**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 entre el salario más bajo y el salario más alto de la tabla SALGRADE”

# Producto cartesiano con restricción

```
SELECT e.ename, e.sal, s.grade
FROM   emp e JOIN salgrade s
       ON e.sal BETWEEN s.losal AND s.hisal;
```

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

14 rows selected.



# Producto Cartesiano (cross join)

- **Cuando se hace una consulta multitable, siempre se ha de poner la condición, de lo contrario se estará realizando un producto cartesiano.**
- **Produce un resultado incorrecto, con la aparición de gran cantidad de filas como resultado.**

# Generación de un Producto Cartesiano

**EMP (14 filas)**

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

**DEPT (4 filas)**

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



**“Producto Cartesiano  
14\*4=56 filas”**

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