

5.- Consultas de varias tablas

Temario

5. Consultas de varias tablas.

Uniendo tablas (inner joins)

Tratamiento de nulos (outer joins: left join y right join).

5.1 Uniendo tablas

- ◉ Existen dos tipos de Joins
 - > Equijoins (inner join)
- ◉ Existen dos adicionales
 - > Outer Joins (left | right join)
 - > Self Joins

5.1 Uniendo tablas (INNER JOINS)

- Algunos manejadores no incluyen en su sintaxis las palabras INNER JOIN, JOIN, LEFT OUTER JOIN, por lo que el método tradicional para hacer un join es colocar las tablas afectadas en la cláusula FROM separadas por comas y la condición join colocarla en la cláusula WHERE. Estas dos formas de crear JOINS generan EXACTAMENTE el mismo resultado, solo que el método anterior permite mayor comodidad y claridad

SELECT	tabla1.columna, tabla2.columna
FROM	tabla1, tabla2
WHERE	tabla1.columna1 = tabla2.columna2;

5.1 Uniendo tablas (INNER JOINS)

- ◉ Los inner joins se conocen como joins o equijoins
- ◉ Se utiliza un JOIN para consultara datos en más de una tabla
- ◉ Escribe la condición join en la cláusula FROM utilizando ON
- ◉ Incluye el nombre de la tabla como prefijo del nombre de la columna, cuando el nombre de la columna aparezca en dos o más tablas

```
SELECT      tabla1.columna, tabla2.columna  
FROM  tabla1 INNER JOIN tabla2 ON  tabla1.columna1 = tabla2.columna2
```

5.1 Uniendo tablas (INNER JOINS)

- **JOIN:** Cuando se requieren datos que están en más de una tabla, se requiere utilizar una condición JOIN
- Los renglones de una tabla pueden ser “unidos” a los de otra tabla, solo si existen en ambas tablas un atributo que sea común entre ellas, normalmente, una relación de llave primaria y llave foránea
- Sintaxis:
 - > `Tabla1.columna1=tabla2.columna2`: es la condición que “junta” o relaciona las tablas. Es preciso preceder cada columna con el nombre de su respectiva tabla, por claridad y mejorar el acceso a la BD
 - > Para relacionar n tablas, se necesita realizar el mínimo de n-1 condiciones join

5.1 Uniendo tablas

● Ejemplo INNER JOIN

```
select emp.ename Trabajador, dept.dname Departamento from EMP,DEPT where EMP.DEPTNO=DEPT.DEPTNO;
```

	Trabajador	Departamento
1	SMITH	RESEARCH
2	ALLEN	SALES
3	WARD	SALES
4	JONES	RESEARCH
5	MARTIN	SALES
6	BLAKE	SALES
7	Rosa Isela	RESEARCH
8	CLARK	ACCOUNTING
9	SCOTT	RESEARCH
10	KING	ACCOUNTING
11	TURNER	SALES
12	ADAMS	RESEARCH
13	JAMES	SALES
14	FORD	RESEARCH
15	MILLER	ACCOUNTING

5.1 Uniendo tablas

◉ Inner Join

```
select emp.ename Trabajador, dept.dname Departamento  
from EMP inner join DEPT ON EMP.DEPTNO=DEPT.DEPTNO;
```

	Trabajador	Departamento
1	SMITH	RESEARCH
2	ALLEN	SALES
3	WARD	SALES
4	JONES	RESEARCH
5	MARTIN	SALES
6	BLAKE	SALES
7	Rosa Isela	RESEARCH
8	CLARK	ACCOUNTING
9	SCOTT	RESEARCH
10	KING	ACCOUNTING
11	TURNER	SALES
12	ADAMS	RESEARCH
13	JAMES	SALES
14	FORD	RESEARCH
15	MILLER	ACCOUNTING

5.2 Tratamiento de nulos (Outer Joins)

- Si los renglones no satisfacen la condición JOIN, el renglón no aparecerá en el resultado de la consulta.
- En la siguiente consulta, no aparece el departamento 40 pues no hay renglones que cumplan la condición

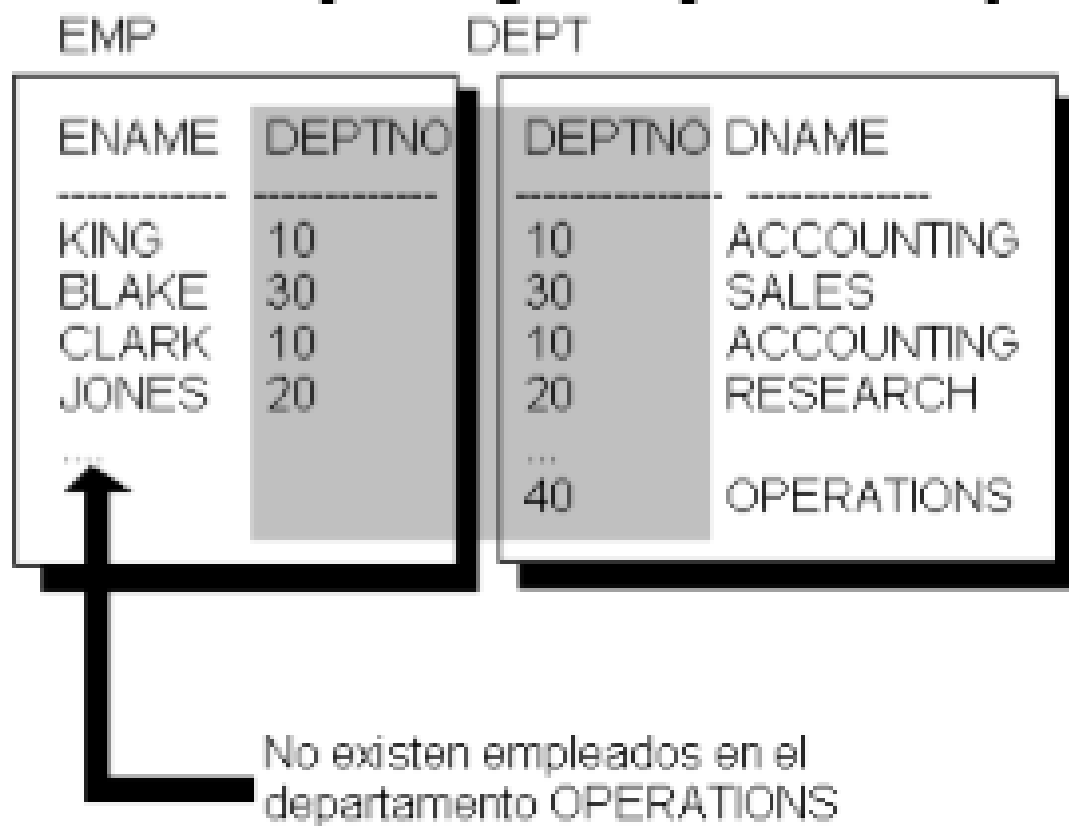
5.2 Tratamiento de nulos (Outer Joins)

```
select e.ENAME,e.DEPTNO,d.DNAME from EMP e, DEPT d where e.DEPTNO=d.DEPTNO;  
select * from DEPT;
```

Results Messages

	ENAME	DEPTNO	DNAME
1	SMITH	20	RESEARCH
2	ALLEN	30	SALES
3	WARD	30	SALES
4	JONES	20	RESEARCH
5	MARTIN	30	SALES
6	BLAKE	30	SALES
7	Rosa Isela	20	RESEARCH
8	CLARK	10	ACCOUNTING
9	SCOTT	20	RESEARCH
10	KING	10	ACCOUNTING
11	TURNER	30	SALES
12	ADAMS	20	RESEARCH
13	JAMES	30	SALES
14	FORD	20	RESEARCH
15	MILLER	10	ACCOUNTING

5.2 Tratamiento de nulos



5.2 Tratamiento de nulos

- Se puede utilizar un OUTER JOIN para ver también aquellos renglones que normalmente no cumplen la condición-join.
- OUTER JOIN puede ser por la izquierda (LEFT) o por la derecha (RIGHT)
- Los renglones que no cumplen la condición pueden ser mostrados dependiendo de que lado se encuentran.
- Se utiliza LEFT o RIGHT dependiendo de que lado se encuentran los renglones que desee aparezcan aun cuando no cumpla la condicion-join.

```
SELECT tabla1.columna, tabla2.columna  
FROM   tabla1 LEFT OUTER JOIN tabla2  
        ON tabla1.columna = tabla2.columna;
```

```
SELECT tabla1.columna, tabla2.columna  
FROM   tabla1 RIGHT OUTER JOIN tabla2  
        ON tabla1.columna = tabla2.columna;
```

5.2 Tratamiento de nulos

```
select e.ENAME,e.DEPTNO,d.DNAME from EMP e RIGHT outer join DEPT d on e.DEPTNO=d.DEPTNO;  
select * from DEPT;
```

Results Messages

	ENAME	DEPTNO	DNAME
1	CLARK	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	MILLER	10	ACCOUNTING
4	SMITH	20	RESEARCH
5	JONES	20	RESEARCH
6	Rosa Isela	20	RESEARCH
7	SCOTT	20	RESEARCH
8	ADAMS	20	RESEARCH
9	FORD	20	RESEARCH
10	ALLEN	30	SALES
11	WARD	30	SALES
12	MARTIN	30	SALES
13	BLAKE	30	SALES
14	TURNER	30	SALES
15	JAMES	30	SALES
16	NULL	NULL	OPERATIONS

5.2 Tratamiento de nulos

```
select e.ENAME,e.DEPTNO,d.DNAME from DEPT d Left outer join EMP e on e.DEPTNO=d.DEPTNO;  
select * from DEPT;
```

Results

Messages

	ENAME	DEPTNO	DNAME
1	CLARK	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	MILLER	10	ACCOUNTING
4	SMITH	20	RESEARCH
5	JONES	20	RESEARCH
6	Rosa Isela	20	RESEARCH
7	SCOTT	20	RESEARCH
8	ADAMS	20	RESEARCH
9	FORD	20	RESEARCH
10	ALLEN	30	SALES
11	WARD	30	SALES
12	MARTIN	30	SALES
13	BLAKE	30	SALES
14	TURNER	30	SALES
15	JAMES	30	SALES
16	NULL	NULL	OPERATIONS

5.3 Tratamiento de nulos

```
select e.ENAME,e.DEPTNO,d.DNAME from EMP e RIGHT outer join DEPT d on e.DEPTNO=d.DEPTNO;
select e.ENAME,e.DEPTNO,d.DNAME from EMP e RIGHT outer join DEPT d on d.DEPTNO=e.DEPTNO;
select e.ENAME,e.DEPTNO,d.DNAME from EMP e left outer join DEPT d on d.DEPTNO=e.DEPTNO;

select e.ENAME,e.DEPTNO,d.DNAME from DEPT d Left outer join EMP e on e.DEPTNO=d.DEPTNO;
select e.ENAME,e.DEPTNO,d.DNAME from DEPT d Left outer join EMP e on d.DEPTNO=e.DEPTNO;
select e.ENAME,e.DEPTNO,d.DNAME from EMP E Left outer join DEPT D on e.DEPTNO=d.DEPTNO;
```

Self JOIN

- ◉ En ocasiones se necesitará un JOIN con la misma tabla
- ◉ Ejemplo: para encontrar al jefe de Blake se necesita:
 - > Encontrar a Blake en la tabla EMP mediante la columna ENAME
 - > Encontrar el número de jefe para Blake en la columna MGR (7839)
 - > Encontrar el nombre del jefe con EMPNO=7839, de tal forma que King es el jefe de Blake

Self JOIN

```
SELECT empleado.ename + ' Trabaja para ' + jefe.ename as 'auto referencia'  
FROM emp empleado, emp jefe WHERE empleado.mgr = jefe.empno;
```

Results Messages

	auto referencia
1	SMITH Trabaja para FORD
2	ALLEN Trabaja para BLAKE
3	WARD Trabaja para BLAKE
4	JONES Trabaja para KING
5	MARTIN Trabaja para BLAKE
6	BLAKE Trabaja para KING
7	Rosa Isela Trabaja para FORD
8	CLARK Trabaja para KING
9	SCOTT Trabaja para JONES
10	TURNER Trabaja para BLAKE
11	ADAMS Trabaja para SCOTT
12	JAMES Trabaja para BLAKE
13	FORD Trabaja para JONES
14	MILLER Trabaja para CLARK

Self JOIN

```
SELECT e.ename + ' Trabaja para ' + j.ename as Autoreferenciada  
FROM emp e JOIN emp j ON e.mgr = j.empno;
```

Results		Messages
Autoreferenciada		
1	SMITH Trabaja para FORD	
2	ALLEN Trabaja para BLAKE	
3	WARD Trabaja para BLAKE	
4	JONES Trabaja para KING	
5	MARTIN Trabaja para BLAKE	
6	BLAKE Trabaja para KING	
7	Rosa Isela Trabaja para FORD	
8	CLARK Trabaja para KING	
9	SCOTT Trabaja para JONES	
10	TURNER Trabaja para BLAKE	
11	ADAMS Trabaja para SCOTT	
12	JAMES Trabaja para BLAKE	
13	FORD Trabaja para JONES	
14	MILLER Trabaja para CLARK	

Reuniendo más de dos tablas

- En ocasiones se necesitará hacer join con mas de dos tablas.
- Por ejemplo para mostrar el nombre, las órdenes y los ítems, el total de cada orden para el cliente TKB SPORT SHOP, se necesitará reunir las tablas CUSTOMER, ORD y ITEM.

Reuniendo más de dos tablas

CUSTOMER		ORD		ITEM
NAME	CUSTID	CUSTID	ORDID	
JOCKSPORTS	100	101	610	
TKB SPORT SHOP	101	102	611	
VOLLYRITE	102	104	612	
JUST TENNIS	103	106	601	
K+T SPORTS	105	102	602	
SHAPE UP	106	106		
WOMENS SPORTS	107	106		
.....				
9 rows selected				

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

21 rows selected

Reuniendo más de dos tablas

```
SELECT c.name, o.ordid, i.itemid, i.itemtot, o.total  
FROM customer c, ord o, item i  
WHERE c.custid = o.custid AND o.ordid = i.ordid AND c.name = 'TKB SPORT SHOP';  
  
select * from CUSTOMER where NAME='TKB SPORT SHOP';
```

Results Messages

	name	ordid	itemid	itemtot	total
1	TKB SPORT SHOP	610	1	35.00	101.40
2	TKB SPORT SHOP	610	2	8.40	101.40
3	TKB SPORT SHOP	610	3	58.00	101.40

Reuniendo más de dos tablas

```
SELECT c.name, o.ordid, i.itemid, i.itemtot, o.total  
FROM customer c, ord o, item i, PRODUCT p  
WHERE c.custid = o.custid AND o.ordid = i.ordid AND c.CUSTID=102 and i.PRODID=p.PRODID
```

Results Messages

	name	ordid	itemid	itemtot	total
1	VOLLYRITE	602	1	56.00	56.00
2	VOLLYRITE	603	2	224.00	224.00
3	VOLLYRITE	611	1	45.00	45.00
4	VOLLYRITE	614	1	15540.00	23940.00
5	VOLLYRITE	614	2	2800.00	23940.00
6	VOLLYRITE	614	3	5600.00	23940.00