Unidad 7. CONSULTAS MULTITABLA

COMBINACIÓN DE TABLAS

- A veces una consulta necesita columnas de varias tablas, en este caso el formato es:
- SELECT columna1, columna2,…
  FROM tabla1, tabla2,…

REGLAS

- Podemos unir tantas tablas como deseemos.(aunque la experiencia aconseja que no sean más de 8)
- En la claúsula SELECT podemos citar columnas de todas las tablas.
- Si hay columnas con el mismo nombre en las distintas tablas de FROM, se debe especificar: *Nombretabla.columna*
- SQL no exige que las columnas de emparejamiento estén relacionadas como clave primaria y clave ajena. Pueden servir cualquier par de columnas de dos tablas, siempre que tengan tipos de datos comparables.
- El criterio que se siga para combinar las tablas debe especificarse en la cláusula WHERE, **si no se especifica ningún criterio,** el resultado de la combinación será un **producto cartesiano**, que emparejará todas las filas de una tabla con todas las filas de la otra

Combinaciones de tablas

- Multiplicación o producto cartesiano
- Composiciones simples o INNER JOIN
- Autocomposiciones
- Composición o combinación externa (OUTER JOIN)

Multiplicación de tablas o producto cartesino

Ejemplo. Obtener el nombre y apellido de los empleados con su nombre de departamento.

SELECT employees.department_id, last_name, first_name, departments.department_id, department_name

FROM employees, departments

El resultado de esta consulta es el producto cartesiano de ambas tablas, es decir si la 1ª tabla tiene 5 filas y la segunda 10, el resultado dará 10x5=50 filas

Composiciones o combinaciones simples

La composición más sencilla es aquella en que la condición de selección se establece con el operador de igualdad entre las columnas que deban coincidir exactamente en tablas diferentes.

SELECT e.department_id, last_name, first_name, d.department_id, department_name
FROM employees e , departments d
WHERE e.department_id=d.department_id

DEPARTMENT_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
90	King	Steven	90	Executive
90	Kochhar	Neena	90	Executive
90	De Haan	Lex	90	Executive
60	Hunold	Alexander	60	IT
60	Ernst	Bruce	60	IT
60	Austin	David	60	IT
60	Pataballa	Valli	60	IT
60	Lorentz	Diana	60	IT
100	Greenberg	Nancy	100	Finance
100	Faviet	Daniel	100	Finance
100	Chen	John	100	Finance
100	Sciarra	Ismael	100	Finance
100	Urman	Jose Manuel	100	Finance
100	Popp	Luis	100	Finance
30	Raphaely	Den	30	Purchasing
30	Khoo	Alexander	30	Purchasing

SQL resuelve la consulta anterior así:

- La cláusula FROM genera todas las combinaciones posibles de filas de la tabla de *empleados* (100 filas) por las de la tabla de *departamentos* (13 filas), resultando una tabla producto de 100x13=1300filas.
- La cláusula WHERE selecciona únicamente aquellas filas de la tabla producto donde coinciden los números de departamento, que necesitan del alias por tener el mismo nombre en ambas tablas. En total se han seleccionado 100 filas y el resto se eliminan.
- La sentencia SELECT visualiza las columnas especificadas de la tablas producto para las filas seleccionadas.

Ejemplos

1. **Obtener los distintos departamentos existentes en la tabla de *empleados*.**

```
SQL> SELECT DISTINCT d.department_id,
  d.department_name
FROM employees e, departaments d
WHERE e.department_id
  =d.deparment_id;
```

2. Seleccionar el código de empleado, apellido y código de oficio y oficio de los empleados que pertenezcan al departamento 100

```
SQL> SELECT employee_id, last_name,
j.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id=j.job_id AND
department_id=100
```

INNER JOIN

- Normalmente emparejamos tablas que están relacionadas entre sí y una de las columnas de emparejamiento es clave principal, pues en este caso, cuando **una** de las **columnas de emparejamiento** tienen un **índice** definido es más eficiente utilizar otro tipo de composición, el **INNER JOIN**.

INNER JOIN

- Permite **emparejar filas** de distintas tablas de forma **más eficiente** que con el producto cartesiano **cuando** una de las **columnas de emparejamiento** está **indexada (por ejemplo es clave primaria)**.Ya que en vez de hacer el producto cartesiano completo y luego seleccionar la filas que cumplen la condición de emparejamiento, para cada fila de una de las tablas **busca directamente** en la otra tabla **las filas que** cumplen la condición, con lo cual **se emparejan** sólo las filas que luego aparecen en el resultado.

SINTAXIS

- La sintaxis es la siguiente:

FROM tabla1 INNER JOIN tabla2 ON tabla1.col1=tabla2.col2

Ejemplo:

- **SELECT ***
  FROM pedidos INNER JOIN clientes ON pedidos.clie = clientes.numclie

Autocomposiciones o SELFJOINS

Las composiciones de una tabla consigo misma reciben el nombre de autocomposiciones

Ejemplo.

1. **Obtener la lista de los empleados con los nombres de sus jefes.**

```
select e.employee_id, e.manager_id, e.last_name "empleado",
    a.employee_id, a.last_name "dire"
from employees e, employees a
where e.manager_id=a.employee_id
```

EMPLOYEE_ID	MANAGER_ID	Empleado	EMPLOYEE_ID	Dire
101	100	Kochhar	100	King
102	100	De Haan	100	King
103	102	Hunold	102	De Haan
104	103	Ernst	103	Hunold
105	103	Austin	103	Hunold
106	103	Pataballa	103	Hunold
107	103	Lorentz	103	Hunold
108	101	Greenberg	101	Kochhar
109	108	Faviet	108	Greenberg
110	108	Chen	108	Greenberg
111	108	Sciarra	108	Greenberg
112	108	Urman	108	Greenberg
113	108	Popp	108	Greenberg
114	100	Raphaely	100	King
115	114	Khoo	114	Raphaely

Ejemplos

2. Obtener los jefes de los empleados cuyo oficio sea el de Jefe de ventas, JOB_ID='SA_MAN'

```
SELECT e.employee_id, e.last_name "empleado",
a.employee_id, a.last_name "dire"
FROM employees e, employees a
WHERE e.manager_id=a.employee_id
AND e.job_id='SA_MAN';
```

Composición o combinación externa (OUTER JOIN)

Es aconsejable que la salida, obtenida por una consulta muestre todas las filas, aunque algunas con falta de información.

Para conseguir este resultado se utiliza la **composición o combinación externa (OUTER JOIN)**.

En ORACLE se coloca (+) detrás de la columna en cuya tabla puede faltar información:

WHERE *tabla1.columna1***(+)** = *tabla2.columna2* o

WHERE *tabla1.columna1* = *tabla2.columna2***(+)**

En este ejemplo se combinan las tablas AUTORES y LIBROS pero la tupla

(Saltor F, Española, FI de UPB) no aparece en la tabla resultante ya que no se combina con ninguna fila de la tabla

Autores

NOMBRE	NACIONALIDAD	INSTITUCION
DATE C.J.	NORTEAM.	RELATIONAL INS.
DE MIGUEL A.	ESPANOLA	F.IM.
SALTOR, F.	ESPANOLA	F.I. DE U.P.B.
CERI, S.	ITALIA	POLITEC. MILAN

Libros

LIBRO	AUTOR	EDITORIAL
DB SYSTEMS	DATE C.J.	ADDISON
BASI DI DATI	CERI, S.	CLUP
SQL STAND.	DATE C.J.	ADDISON
DISENO DB	DE MIGUEL, A.	RAMA

Autores • Libros
(Autores.nombre=Libros.autor)

NOMBRE	NAC.	INSTITUCION	LIBRO	EDITORIAL
DATE C.J.	NORTEAM.	RELATIONAL INS.	DB SYSTEMS	ADDISON
DE MIGUEL, A.	ESPANOLA	F.IM.	DISENO BD	RAMA
DATE C.J.	NORTEAM.	RELATION INS.	SQL STAND.	ADDISON
CERI, S.	ITALIANA	PLITEC. MILAN	BASI DI DATI	CLUP

- En este caso basta con aplicar el operador de combinación externa izquierda (*left outer join)*, ya que es en la segunda relación donde se crean tuplas nulas.

Autores / Libros (Autores.nombre=Libros.autor)

NOMBRE	NACIONALIDAD	INSTITUCIÓN	LIBRO	EDITORIAL
DATE C.J.	USA	RELATIONAL INS.	DB SYSTEMS	ADDISON
DE MIGUEL A.	ESPAÑOLA	F. IM.	DISEÑO BD	RAMA
DATE C.J.	USA	RELATIONAL INS.	SQL STAND.	ADDISON
CERI, S.	ITALIA	POLITEC. MILAN	BASI DI DATI	CLUP
SALTOR F.	ESPAÑOLA	F.I. DE U.P.B.	?	?

Figura 2.12. Ejemplo de operador de combinación externa

EJEMPLO

select first_name, e.department_id, d.department_id, department_name

from employees e, departments d

where e.department_id(+)=d.department_id

Se coloca el + en la tabla **emple** ya que puede haber algún departamento en el que no trabajen empleados por lo que aparecerán filas nulas en la tabla **emple**

Jose Manuel	100	100	Finance
John	100	100	Finance
Daniel	100	100	Finance
Ismael	100	100	Finance
Nancy	100	100	Finance
William	110	110	Accounting
Shelley	110	110	Accounting
-	-	120	Treasury
-	-	130	Corporate Tax
-	-	140	Control And Credit
-	-	150	Shareholder Services
-	-	160	Benefits
-	-	170	Manufacturing
-	-	180	Construction
-	-	190	Contracting
-	-	200	Operations
-	-	210	IT Support
-	-	220	NOC
-	-	230	IT Helpdesk
-	-	240	Government Sales
-	-	250	Retail Sales
-	-	260	Recruiting
-	-	270	Payroll

Composiciones y subconsultas

Hay ocasiones en que una consulta puede resolverse con una composición o combinación (**join**) de tablas, o con una subconsulta. En este caso sería preferible hacerlo con una subconsulta.

En general, si no se necesita visualizar columnas de más de una tabla, se utiliza una subconsulta.

Si se necesita visualizar columnas de más de una tabla, se usará una composición o combinación.

Obtener apellido y oficio de los empleados que tienen el mismo oficio y mismo número de departamento que el de INVESTIGACIÓN.

Con subconsulta:

```
SQL> SELECT apellido,oficio FROM empleados
WHERE oficio IN (SELECT oficio FROM empleados WHERE dep_no IN
(SELECT dep_no FROM departamentos
WHERE dnombre='INVESTIGACION'));
```

Con composición de tablas:

```
SQL> SELECT apellido,oficio FROM empleados
WHERE oficio IN (SELECT e.oficio FROM empleados e, departamentos d
WHERE e.dep_no=d.dep_no AND d.dnombre='INVESTIGACION');
```