

CONSULTAS MULTITABLA

JOIN (INTERNAS-EXTERNAS)

OPERADOR UNION

CONSULTAS REFLEXIVAS

CONSULTAS CON TABLAS DERIVADAS

CONSULTAS MULTITABLA

- Una consulta multitabla es aquella en la que se puede consultar información de más de una tabla (en una sola select).
- Se aprovecha los campos relacionados de las tablas para unirlos (join).
- Para poder realizar este tipo de consultas hay que utilizar la siguiente sintaxis en la cláusula FROM:

```
SELECT [ALL | DISTINCT | DISTINCTROW]
    expresion select,...
FROM referencias de tablas
WHERE condiciones
[GROUP BY {nombre_col | expresion | posicion}
    [ASC | DESC], ... [WITH ROLLUP]]
[HAVING condiciones]
[ORDER BY {nombre_col | expresion | posicion}
    [ASC | DESC] ,...]
[LIMIT {[desplazamiento,] contador | contador OFFSET desplazamiento}]
```

[FROM referencias_tablas]

SINTAXIS

referencia_tabla, [referencia_tabla].....

| referencia_tabla [INNER | CROSS] JOIN referencia_tabla [ON condicion]

| referencia_tabla LEFT [OUTER] JOIN referencia_tabla ON condición

| referencia_tabla LEFT [OUTER] JOIN referencia_tabla USING (lista_col)

| referencia_tabla RIGHT [OUTER] JOIN referencia_tabla ON condición

- JOIN INTERNA:
 - De Equivalencia (INNER JOIN)
 - Natural (NATURAL JOIN)
- PRODUCTO CARTESIANO:
 - CROSS JOIN
- JOIN EXTERNA:
 - De tabla derecha (RIGHT OUTER JOIN)
 - De tabla izquierda (LEFT OUTER JOIN)
 - Completa (FULL OUTER JOIN)

La cláusula FROM

- Mencionar dos o mas tablas porque la consulta necesita traer datos de ellas. Separadas por comas.
- Cuando se combina (se denomina join) información desde múltiples tablas, se necesita indicar qué registro de una tabla se combinará con qué registro de la otra. Esto se realiza a través de los nombre de sus columnas respectivas. En su cláusula WHERE.

JOIN DE DOS TABLAS

- Es una composición entre las mismas basada en la coincidencia exacta (u otro tipo de comparación) de dos columnas, una de cada tabla.
- Join forma parejas de filas haciendo coincidir los contenidos de las columnas relacionadas.

```
SELECT * FROM curso, departamento  
WHERE cdept=dept;
```

JOIN DE DOS TABLAS

```
SELECT * FROM curso, departamento  
WHERE cdept=dept;
```

```
SELECT cnombre, ctarifa, dchfno  
FROM curso, departamento  
WHERE cdept=dept AND ctarifa >100;
```

JOINS DE TRES O MAS TABLAS

- Un Join de tres o mas tablas es una composición entre las mismas basadas en la coincidencia exacta (u otro tipo de comparación) de tres o mas columnas, una de cada tabla.

```
SELECT numopedido, importe, empresa, nombre  
FROM pedidos,clientes,repventas  
WHERE clie=numclie AND rep=numempl AND  
importe > 25000;
```


Alias para tablas

- Dado que la columna name aparece en ambas tablas, se debe especificar a cuál tabla pertenece la columna al hacer referencia a ella.
- Esto se hace anteponiendo el nombre de la tabla al nombre de la columna en la cláusula FROM .

JOINS INTERNOS Y EXTERNOS

- El concepto que conocemos hasta ahora de JOIN corresponde a lo que se conoce como JOIN INTERNO , y es la operación que combina información procedente de dos o mas tablas mediante la formación de parejas de filas relacionadas en las tablas según la condición de búsqueda (WHERE) del JOIN.
- ¿Qué pasa cuando no hay emparejamiento ?

Tabla Empleado

Apellido	IDDepartamento
Andrade	31
Jordán	33
Steinberg	33
Róbinson	34
Zolano	34
Gaspar	36

Tabla Departamento

NombreDepartamento	IDDepartamento
Ventas	31
Ingeniería	33
Producción	34
Mercadeo	35

Existen empleados que tienen asignado un número de departamento que no se encuentra en la tabla **Departamento** (Gaspar).

Igualmente, existen departamentos a los cuales no pertenece ningún empleado (Mercadeo).

<http://www.pvilas.com/2013/02/entender-los-joins-sql-graficamente.html>

JOIN EXTERNO COMPLETO

Es una ampliación de un JOIN INTERNO (o simplemente JOIN) que contiene la siguiente información:

- ☐ El JOIN INTERNO (o JOIN) de las dos tablas.
- ☐ Por cada fila de la primera tabla que no haya correspondido a ninguna fila de la segunda tabla, se añade una fila a los resultados, utilizando los valores de las columnas de la primera tabla y suponiendo un valor NULL para todas las columnas de la segunda tabla correspondientes en el emparejamiento.
- ☐ Por cada fila de la segunda tabla que no haya correspondido a ninguna fila de la primera tabla, se añade una fila a los resultados, utilizando los valores de las columnas de la segunda tabla y suponiendo un valor NULL para todas las columnas de la primera tabla correspondientes en el emparejamiento.

JOIN EXTERNO COMPLETO

SELECT,,

FROM empleado FULL OUTER JOIN departamento
ON empleado.IDDepartamento =
departamento.IDDepartamento;

<http://www.pvilas.com/2013/02/entender-los-joins-sql-graficamente.html>

Empleado.Apellido	Empleado.IDDepartamento	Departamento.NombreDepartamento	Departamento.IDDepartamento
Zolano	34	Producción	34
Jordán	33	Ingeniería	33
Róbinson	34	Producción	34
Gaspar	36	NULL	NULL
Steinberg	33	Ingeniería	33
Andrade	31	Ventas	31
NULL	NULL	Mercadeo	35

- Como se puede notar, en este caso se encuentra el empleado Gaspar con valor nulo en su área correspondiente, y se muestra además el departamento de Mercadeo con valor nulo en los empleados de esa área.
- Algunos sistemas de bases de datos no soportan esta funcionalidad, pero puede ser emulada a través de las combinaciones de tabla izquierda, tabla derecha y de la sentencia de unión [union](#).

JOIN EXTERNO Izquierdo

Se produce entre dos tablas ofreciendo la siguiente información:

- ❑ El JOIN INTERNO (o JOIN) de las dos tablas.
- ❑ Por cada fila de la primera tabla que no haya correspondido a ninguna fila de la segunda tabla, se añade una fila a los resultados, utilizando los valores de las columnas de la primera tabla y suponiendo valor NULL para todas las columnas de la segunda tabla correspondientes en el emparejamiento.

JOIN EXTERNO Izquierdo

LEFT OUTER JOIN o LEFT JOIN

SELECT ...,

FROM empleado LEFT OUTER JOIN departamento
ON empleado.IDDepartamento =
departamento.IDDepartamento

<http://www.pvilas.com/2013/02/entender-los-joins-sql-graficamente.html>

Empleado.Apellido	Empleado.IDDepartamento	Departamento.NombreDepartamento	Departamento.IDDepartamento
Jordán	33	Ingeniería	33
Andrade	31	Ventas	31
Róbinson	34	Producción	34
Zolano	34	Producción	34
Gaspar	36	NULL	NULL
Steinberg	33	Ingeniería	33

JOIN EXTERNO Derecho

- Se produce entre dos tablas ofreciendo la siguiente información:
 - ☐ El JOIN INTERNO (o JOIN) de las dos tablas.
 - ☐ Por cada fila de la segunda tabla que no haya correspondido a ninguna fila de la primera tabla, se añade una fila a los resultados, utilizando los valores de las columnas de la segunda tabla y suponiendo valor NULL para todas las columnas de la primera tabla correspondientes en el emparejamiento.

JOIN EXTERNO Derecho

RIGHT OUTER JOIN o RIGHT JOIN

SELECT ...,

FROM empleado RIGHT OUTER JOIN departamento
ON empleado.IDDepartamento =
departamento.IDDepartamento

<http://www.pvilas.com/2013/02/entender-los-joins-sql-graficamente.html>

Empleado.Apellido	Empleado.IDDepartamento	Departamento.NombreDepartamento	Departamento.IDDepartamento
Zolano	34	Producción	34
Jordán	33	Ingeniería	33
Róbinson	34	Producción	34
Steinberg	33	Ingeniería	33
Andrade	31	Ventas	31
NULL	NULL	Mercadeo	35

Ejemplos: ¿Los mismos resultados?

```
SELECT nombre, ciudad, oficinarep, numempl, dir  
FORM repventas, oficinas  
WHERE numempl=dir;  
[ VENDEDORES QUE SON DIRECTORES]
```

```
SELECT nombre, ciudad, oficinarep, numempl, dir  
FORM repventas LEFT JOIN  oficinas  
ON numempl=dir;  
[VENDEDORES QUE SON DIRECTORES Y QUE NO LO SON y como  
valor para dir pondrá NULL.]
```

```
SELECT nombre, ciudad, oficinarep, numempl, dir  
FORM repventas INNER JOIN  oficinas  
ON numempl=dir;  
[ VENDEDORES QUE SON DIRECTORES]
```

JOINS INTERNOS

De equivalencia: (INNER JOIN)

Natural (NATURAL JOIN)

Hay dos formas diferentes para expresar las INNER JOIN o composiciones internas:

- La primera usa ‘,’ (la coma) para separar las tablas a combinar en la sentencia FROM (es SQL1- realizado hasta ahora).
- La segunda usa la palabra reservada INNER JOIN.

INNER JOIN

(JOIN INTERNO-equivalencia)

- Con la operación INNER JOIN se calcula el producto cartesiano de todos los registros, después cada registro de la primera tabla es combinado con cada registro de la segunda tabla, y solo se seleccionan aquellos registros que satisfacen las condiciones que se especifiquen. Los valores NULOS no se combinan.

Las dos consultas siguientes son similares y se realizan de manera explícita (A) e implícita (B).

A. Ejemplo de la sentencia **INNER JOIN** explícita:

```
SELECT Campos FROM empleado INNER JOIN  
departamento ON empleado.IDDepartamento  
= departamento.IDDepartamento
```

B. Ejemplo de la sentencia **INNER JOIN** implícita:

```
SELECT Campos FROM empleado, departamento  
WHERE empleado.IDDepartamento =  
departamento.IDDepartamento
```

JOIN INTERNO

De equivalencia: (INNER JOIN)

Empleado.Apellido	Empleado.IDDepartamento	departamento.NombreDepartamento	departamento.IDDepartamento
Zolano	34	Producción	34
Jordán	33	Ingeniería	33
Róbinson	34	Producción	34
Steinberg	33	Ingeniería	33
Andrade	31	Ventas	31

El empleado Gaspar y el departamento de Mercadeo no son presentados en los resultados ya que ninguno de éstos tiene registros correspondientes en la otra tabla.

No existe un departamento con número 36 ni existe un empleado con número de departamento 35.

INNER JOIN EJEMPLO sintáxis

```
SELECT ..., ..., FROM animales INNER JOIN  
    propietarios
```

```
ON animales.propietario = propietarios.dni
```

```
SELECT ..., ..., FROM animales, propietarios  
WHERE animales.propietario = propietarios.dni
```

Resultados idénticos. Si no hay coincidencia no se muestran. Si hay animales sin propietario no se muestran.

JOIN INTERNO- Natural

(NATURAL JOIN)

- Es una especialización de la combinación de equivalencia, anteriormente mencionada. En este caso se comparan todas las columnas que tengan el mismo nombre en ambas tablas. La tabla resultante contiene sólo una columna por cada par de columnas con el mismo nombre.

Ejemplo de combinación natural:

```
SELECT campos FROM empleado NATURAL JOIN  
departamento
```

El resultado es un poco diferente al del ejemplo anterior, ya que esta vez la columna IDDepartamento se muestra sola una vez en la tabla resultante.

Empleado.Apellido	IDDepartamento	Departamento.NombreDepartamento
Zolano	34	Producción
Jordán	33	Ingeniería
Róbinson	34	Producción
Steinberg	33	Ingeniería
Andrade	31	Ventas

El uso de esta sentencia NATURAL puede producir resultados ambiguos y generar problemas si la base de datos cambia, porque al añadir, quitar o renombrar las [columnas](#) puede perder el sentido la sentencia; por esta razón es preferible expresar el predicado usando las otras expresiones nombradas anteriormente (ejemplos A y B).

DESCRIBE Empleados;

Field	Type	Null	Key	Default	Extra
CodigoEmpleado	int(11)	NO	PRI	NULL	
Nombre	varchar(50)	NO		NULL	
Apellido1	varchar(50)	NO		NULL	
Apellido2	varchar(50)	YES		NULL	
Extension	varchar(10)	NO		NULL	
Email	varchar(100)	NO		NULL	
CodigoOficina	varchar(10)	NO		NULL	#relación
CodigoJefe	int(11)	YES		NULL	
Puesto	varchar(50)	YES		NULL	

DESCRIBE Oficinas;

Field	Type	Null	Key	Default	Extra
CodigoOficina	varchar(10)	NO	PRI	NULL	#relación
Ciudad	varchar(30)	NO		NULL	
Pais	varchar(50)	NO		NULL	
Region	varchar(50)	YES		NULL	
CodigoPostal	varchar(10)	NO		NULL	
Telefono	varchar(20)	NO		NULL	
LineaDireccion1	varchar(50)	NO		NULL	
LineaDireccion2	varchar(50)	YES		NULL	

NATURAL JOIN coge los mismos nombre de campo, en este caso CodigoOficina

```
SELECT CodigoEmpleado, Empleados, Nombre,  
       CodigoOficina, Oficinas.Ciudad  
FROM Empleados NATURAL JOIN Oficinas;
```

Aunque el CódigoOficina es un campo que está en dos Tablas, esta vez no es necesario precederlo del nombre de la tabla puesto que NATURAL JOIN devuelve un único campo por cada pareja de campos con el mismo nombre.

Producto cartesiano

Cruzada (*Cross join*)

Presenta el producto cartesiano de todos los registros de las dos tablas.

El código SQL para realizar este producto cartesiano enuncia las tablas que serán combinadas, pero no incluye algún predicado que filtre el resultado.

Ejemplo de combinación cruzada explícita:

- `SELECT * FROM empleado CROSS JOIN departamento`

Ejemplo de combinación cruzada implícita:

- `SELECT * FROM empleado, departamento;`

Esta clase de combinaciones son usadas pocas veces; generalmente se les agregan condiciones de filtrado con la sentencia WHERE para hallar resultados específicos.

OPERADOR UNION

- El Operador UNION, añade al conjunto de resultados producidos por una SELECT, los resultados de otra SELECT.

SELECT FROM

UNION [ALL]

SELECT FROM

- El parámetro ALL incluye todos los registros de las dos select, incluyendo los que son iguales. Si no se indica ALL, se excluyen los duplicados.

Ejercicios de NBA con join

Realizar los ejercicios: 23, 25,28,30,35,38,40

23- Nombre de los jugadores de la división
"SouthWest".(DOS TABLAS)

```
SELECT JUGADORES.NOMBRE FROM JUGADORES  
INNER JOIN EQUIPOS  
ON NOMBRE_EQUIPO = EQUIPOS.NOMBRE AND  
DIVISION = 'SOUTHWEST';
```

Ejercicios de NBA con join

Realizar los ejercicios: 23, 25,28,30,35,38,40

25- ¿En qué divisiones juegan jugadores serbios?(DOS TABLAS JOIN)

```
SELECT E.DIVISION  
FROM EQUIPOS E INNER JOIN JUGADORES J  
ON E.NOMBRE= J.NOMBRE_EQUIPO AND  
J.PROCEDENCIA LIKE 'SERBIA%';
```


Ejercicios de NBA con join

Realizar los ejercicios: 23, 25,28,30,35,38,40

28-Equipo y ciudad de los jugadores españoles de la NBA.(DOS TABLAS)

```
SELECT E.NOMBRE, E.CIUDAD  
FROM EQUIPOS E INNER JOIN JUGADORES J  
ON E.NOMBRE = J.NOMBRE_EQUIPO AND  
J.PROCEDENCIA = 'SPAIN';
```

Ejercicios de NBA con join

Realizar los ejercicios: 23, 25,28,30,35,38,40

30-Puntos de PAU GASOL en toda su carrera.(DOS TABLAS)

```
SELECT J.NOMBRE, SUM(E.PUNTOS_POR_PARTIDO)  
AS PUNTOS
```

```
FROM JUGADORES J inner join ESTADISTICAS E
```

```
on J.CODIGO = E.JUGADOR AND J.NOMBRE = 'PAU  
GASOL'
```

```
GROUP BY E.JUGADOR;
```

Ejercicios de NBA con join

Realizar los ejercicios: 23, 25,28,30,35,38,40

35-Número de jugadores que tiene cada equipo de la conferencia oeste.(DOS TABLAS)

```
SELECT COUNT(J.CODIGO) AS 'TOTAL JUGADORES',  
       J.NOMBRE_EQUIPO  
FROM EQUIPOS E inner join JUGADORES J  
on CONFERENCIA = 'WEST' AND E.NOMBRE=  
   J.NOMBRE_EQUIPO  
GROUP BY J.NOMBRE_EQUIPO;
```

Ejercicios de NBA con join

Realizar los ejercicios: 23, 25,28,30,35,38,40

38-Asistencias por partido de "JOSE CALDERON" en la temporada "07/08".(DOS TABLAS)

```
SELECT ASISTENCIAS_POR_PARTIDO, NOMBRE  
FROM ESTADISTICAS inner join JUGADORES  
on CODIGO= JUGADOR AND TEMPORADA = '07/08'  
AND NOMBRE="JOSE CALDERON";
```

Ejercicios de NBA con join

Realizar los ejercicios: 23, 25,28,30,35,38,40

40-Número de jugadores que tiene cada equipo de la conferencia este. (EAST).(DOS TABLAS) (HAY 15 EQUIPOS EN ESA CONFERENCIA)

```
SELECT COUNT(CODIGO) AS 'TOTAL JUGADORES',  
       CONFERENCIA, E.NOMBRE  
FROM   EQUIPOS E INNER JOIN JUGADORES J  
ON E.NOMBRE = NOMBRE_EQUIPO AND CONFERENCIA  
   ='EAST'  
GROUP BY NOMBRE_EQUIPO;
```

Una instrucción JOIN con varias cláusulas ON

SELECT campos

FROM **tabla1** INNER JOIN **tabla2**

ON **tabla1**.campo1 **operadorcomparacion**
tabla2.campo1

AND

ON **tabla1**.campo2 **operadorcomparacion**
tabla2.campo2

OR

ON **tabla1**.campo3 **operadorcomparacion**
tabla2.campo3

Anidamiento de instrucciones JOIN

```
SELECT campos
FROM tabla1 INNER JOIN
(tabla2 INNER JOIN
( tabla3 INNER JOIN
( tabla4 [INNER JOIN [(] INNER JOIN ....]
ON tabla3.campo3 operadorcomparacion tabla4.campo4)
ON tabla2.campo2 operadorcomparacion tabla3.campo3)
ON tabla1.campo1 operadorcomparacion tabla2.campo2);
```

CONSULTAS REFLEXIVAS

Crear una autocombinación con una relación reflexiva

Se puede combinar una tabla consigo misma mediante el uso de una relación reflexiva: una relación en la que las columnas de clave externa que hacen referencia y las columnas de clave principal a las que se hace referencia estén en la misma tabla.

DESCRIBE Empleados;

Field	Type	Null	Key	Default	Extra
CodigoEmpleado	int(11)	NO	PRI	NULL	
Nombre	varchar(50)	NO		NULL	
Apellido1	varchar(50)	NO		NULL	
Apellido2	varchar(50)	YES		NULL	
Extension	varchar(10)	NO		NULL	
Email	varchar(100)	NO		NULL	
CodigoOficina	varchar(10)	NO		NULL	#relación
CodigoJefe	int(11)	YES		NULL	
Puesto	varchar(50)	YES		NULL	

Por ejemplo, supongamos que la tabla empleados contiene una columna adicional **codigojefe** que es una clave externa de **codigojefe** a **codigoempleado**.

De esa forma, en las filas de la tabla empleados, la columna **codigojefe** se referirá al jefe de cada empleado.

Por ser más exactos, se referirá al **codigoempleado** del jefe del empleado.

Empleados(**codigoempleado**, nombre, apellido, **codigojefe**)

Crear una autocombinación con una relación reflexiva

- Mediante la combinación de la tabla consigo misma , por medio de esta relación reflexiva, puede establecerse un conjunto de resultados en el que cada fila contenga el nombre de un jefe y el nombre de uno de los empleados de ese jefe.

REFLEXIVA

- Informe de Nombre y apellidos de empleados junto con el Nombre y apellidos de sus jefe.

```
DESCRIBE Empleados;
```

Field	Type	Null	Key	Default	Extra
CodigoEmpleado	int(11)	NO	PRI	NULL	
Nombre	varchar(50)	NO		NULL	
Apellido1	varchar(50)	NO		NULL	
Apellido2	varchar(50)	YES		NULL	
Extension	varchar(10)	NO		NULL	
Email	varchar(100)	NO		NULL	
CodigoOficina	varchar(10)	NO		NULL	#relación
CodigoJefe	int(11)	YES		NULL	
Puesto	varchar(50)	YES		NULL	

El código SQL resultante puede presentar el siguiente aspecto:

```
SELECT jefes.nombre, empleados.nombre,  
FROM empleados  
INNER JOIN empleados jefes  
ON empleados.jefe_emp_id = jefes.emp_id
```

Para ver su funcionamiento, con NBA no es posible (no hay tiene interrelaciones reflexivas).
Importamos JARDINERIA está en INTRANET.

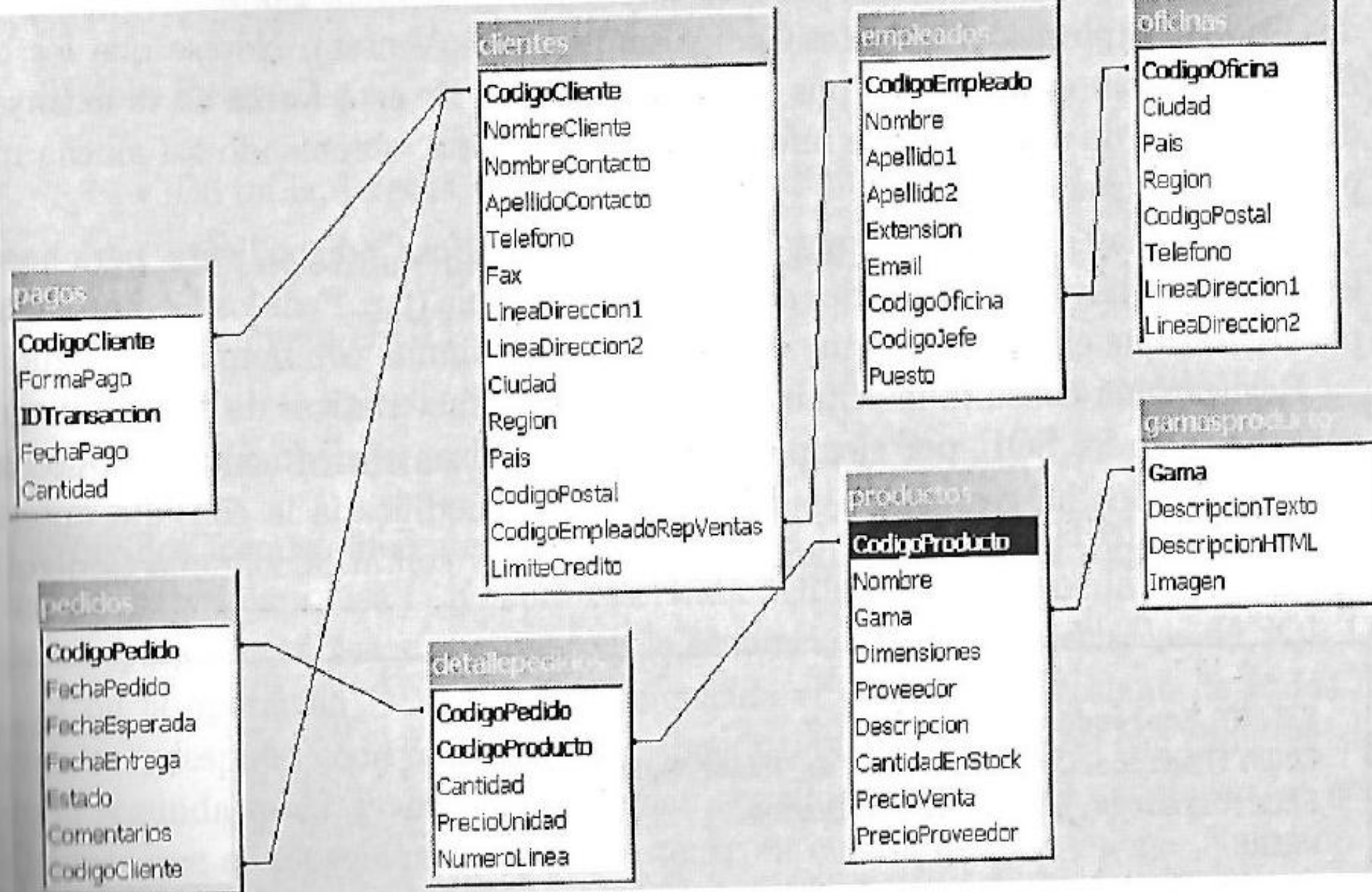


Figura 4.1: Relaciones de la bbdd "jardineria".

empleados

CodigoEmpleado

Nombre

Apellido1

Apellido2

Extension

Email

CodigoOficina

CodigoJefe

Puesto

Reflexiva – probamos-COMPROBAMOS

SELECT

e.Nombre,e.Apellido1,j.nombre, j.apellido1

FROM Empleados e INNER JOIN Empleados j

ON e.CodigoJefe= j.Codigoempleado;

¿Cómo hacer para que aparezcan los empleados sin jefes?

¿Cuál es el resultado si hacemos un ***RIGHT JOIN***?

Modifica la consulta para que sea más clara

Nombre y Apellidos de los empleados
(EMPLEADO) y Nombre y Apellido de su Jefe (JEFE)

Usa Concat y alias para nombrar las columnas

Salen 30 filas

Comprueba que resultado es correcto.

EMPLEADO	JEFE
Ruben López	Marcos Magaña
Alberto Soria	Ruben López
Maria Solís	Ruben López
Felipe Rosas	Alberto Soria
Juan Carlos Ortiz	Alberto Soria
Carlos Soria	Alberto Soria
Mariano López	Carlos Soria
Lucio Campoamor	Carlos Soria
Hilario Rodriguez	Carlos Soria
Emmanuel Magaña	Alberto Soria
José Manuel Martinez	Emmanuel Magaña
David Palma	Emmanuel Magaña
Oscar Palma	Emmanuel Magaña
Francois Fignon	Alberto Soria
Lionel Narvaez	Francois Fignon
Laurent Serra	Francois Fignon
Michael Bolton	Alberto Soria
Walter Santiago Sanchez	Michael Bolton
Hilary Washington	Alberto Soria
Marcus Paxton	Hilary Washington
Lorena Paxton	Hilary Washington
Nei Nishikori	Alberto Soria
Narumi Riko	Nei Nishikori
Takuma Nomura	Nei Nishikori
Amy Johnson	Alberto Soria
Larry Westfalls	Amy Johnson
John Walton	Amy Johnson
Kevin Fallmer	Alberto Soria
Julian Bellinelli	Kevin Fallmer
Mariko Kishi	Kevin Fallmer

30 rows in set (0.00 sec)

Comprobar que:

- Empleado 1 (Marcos Magaña) no debe salir porque no tiene jefe, él es el DIRECTOR GENERAL.
- El empleado nº 7 (Carlos Soria) es jefe de tres empleados (8,9,10)

Mariano	López	Murcia
Lucio	Campoamor	Martín
Hilario	Rodriguez	Huertas

SELECT

Concat (e.Nombre,' ',e.Apellido1) as EMPLEADO,

Concat(j.nombre, ' ', j.apellido1) as JEFE

FROM Empleados e INNER JOIN Empleados j

ON e.CodigoJefe= j.Codigoempleado;

CONSULTAS CON TABLAS DERIVADAS

Son aquellas que utilizan sentencias
SELECT en la cláusula FROM en lugar
de nombres de tablas

EJEMPLO

```
SELECT * FROM  
(SELECT codigoempleado, nombre from empleados  
WHERE codigoOficina = 'TAL-ES') as tabla_derivada;
```

Hacemos :

Sacar un listado con el número de partidos ganados por los equipos de la NBA

Sacar la media de partidos ganados por los equipos del oeste.