CLASE DE REPASO JOINS

Módulo: BASES DE DATOS

Curso 2022/2023. 1° DAM

Ruth Lospitao Ruiz

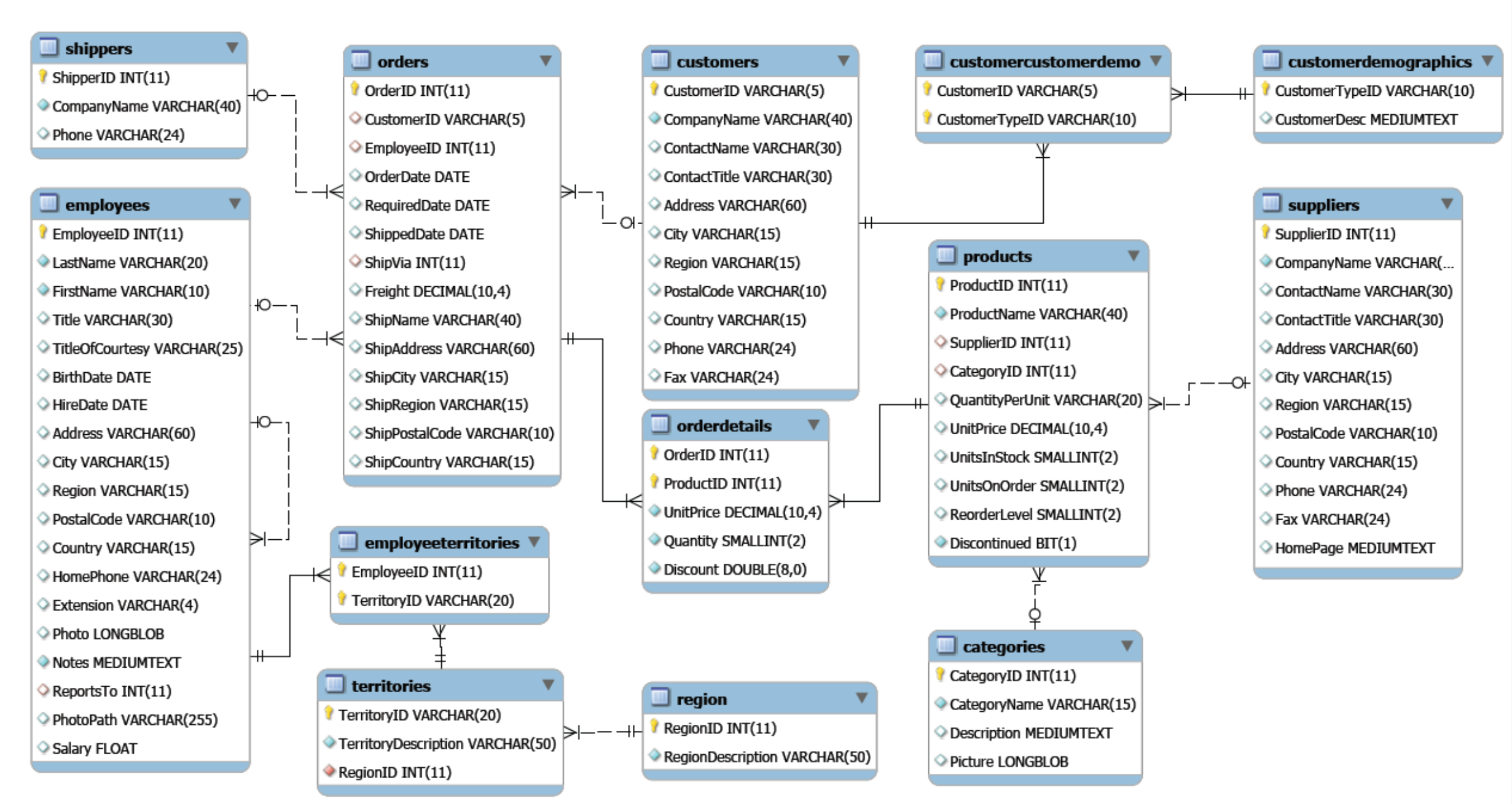
QUÉ SON

- La operación JOIN o combinación permite mostrar columnas de varias tablas como si se tratase de una sola tabla, combinando entre sí los registros relacionados usando para ello claves externas. Las tablas relacionadas se especifican en la cláusula FROM, y además hay que hacer coincidir los valores que relacionan las columnas de las tablas.

```
SELECT [ALL / DISTINCT ] [*] / [LISTADO COLUMNAS O EXPRESIONES]

FROM NOMBRETABLA1 JOIN NOMBRETABLA2 ON CONDICIONES_VINCULOS_TABLAS
```

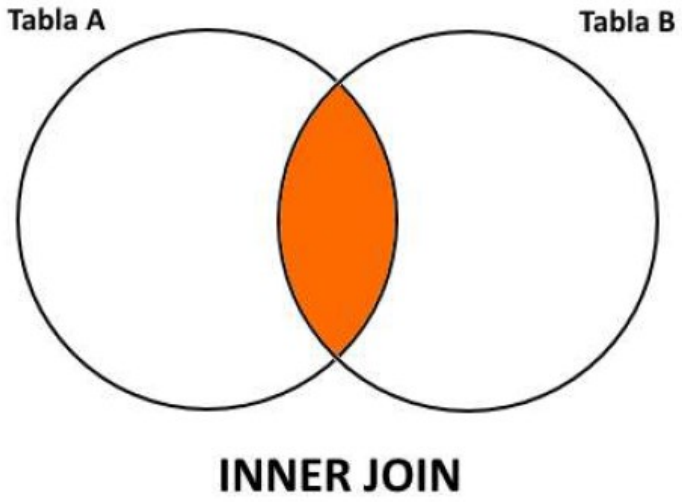
- En SQL hay **tres tipos de JOIN**
  - Combinación interna INNER JOIN
  - Cruzada CROSS JOIN
  - Combinación externa OUTER JOIN
    - LEFT OUTER JOIN o LEFT JOIN
    - RIGHT OUTER JOIN o RIGHT JOIN
    - FULL OUTER JOIN o FULL OUTER JOIN

BASE DE DATOS NORTHWIND



INNER JOIN

- Devuelven únicamente aquellos registros/filas que tienen valores idénticos en los dos campos que se comparan para unir ambas tablas. Es decir, aquellas que tienen elementos en las dos tablas, identificados éstos por el campo de relación



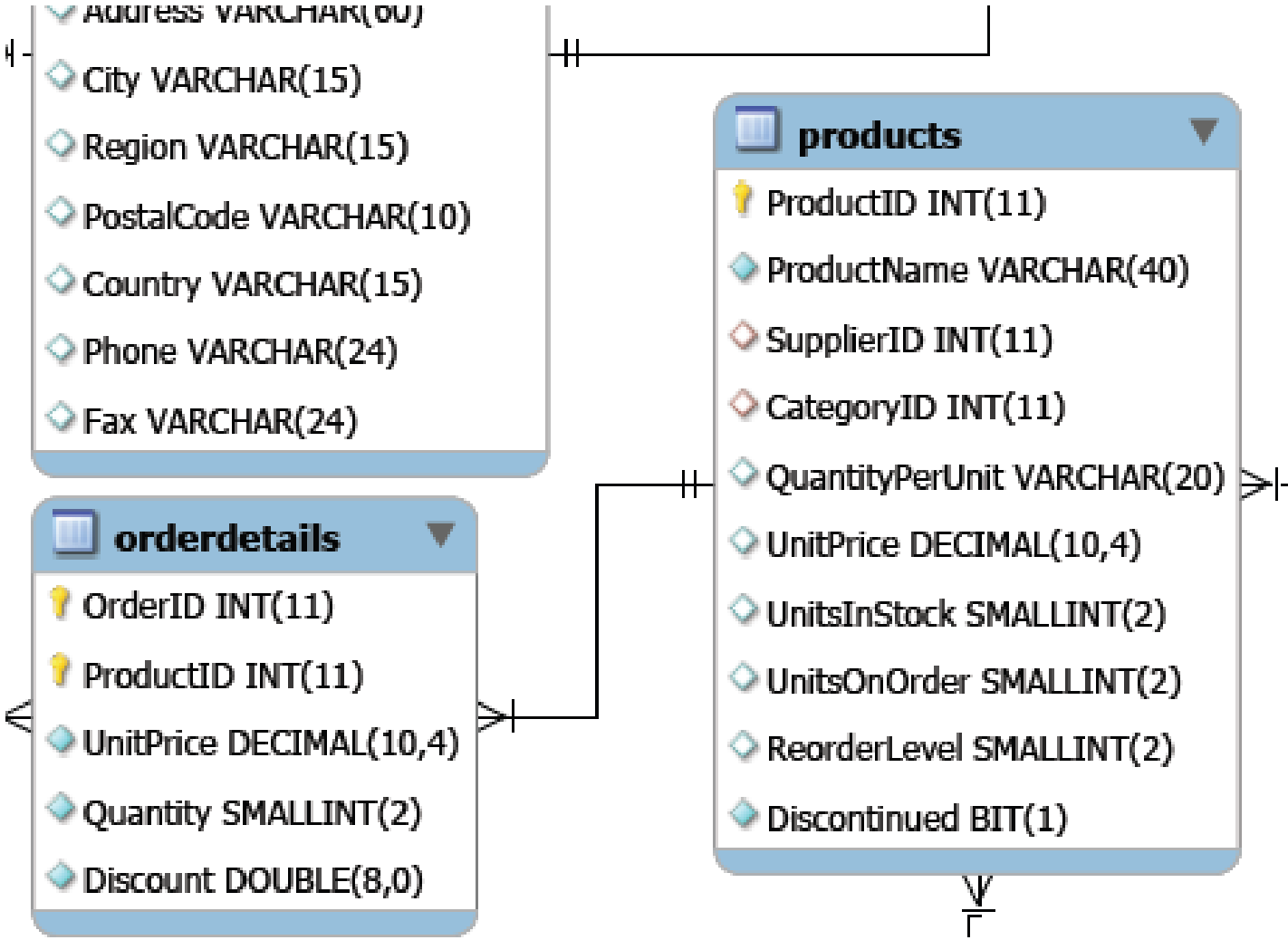
- En este caso se devuelven los registros que tienen nexo de unión en ambas tablas. En realidad, esto ya lo conocíamos puesto que, en las combinaciones internas, el uso de la palabra INNER es opcional así que si simplemente indicamos la palabra JOIN y la combinación de columnas el sistema sobreentiende que estamos haciendo una combinación interna (INNER JOIN).

```
Select <lista_Campos>
From <TablaA A> inner join <TablaB B>
On a.key=b.key
```

```
Select <lista_Campos>
From <TablaA A> join <TablaB B>
On a.key=b.key
```

EJEMPLO INNER JOIN

```
#se desea conocer todos los
productos que se encuentran
en una orden

select p.ProductID,
     p.ProductName,
     o.OrderID
from Products p
inner join order_details o
on p.ProductID=o.productID;
```



NOTA: cuando el campo por el que estamos relacionando tiene el mismo nombre, poner delante nombre de la tabla o su alias

OUTER JOIN

- Devuelven todos los valores de la tabla que hemos puesto a la derecha, los de la tabla que hemos puesto a la izquierda o los de ambas tablas según el caso, devolviendo además valores nulos en las columnas de las tablas que no tengan el valor existente en la otra tabla
- Es decir, que **nos permite seleccionar algunas filas de una tabla, aunque éstas no tengan correspondencia con las filas de la otra tabla con la que se combina.** ASintaxis general

```
SELECT <lista_campos>
FROM <TABLA_A A> [LEFT/RIGHT/FULL] [OUTER] JOIN <TABLA_B B>
ON A.KEY=B.KEY
```

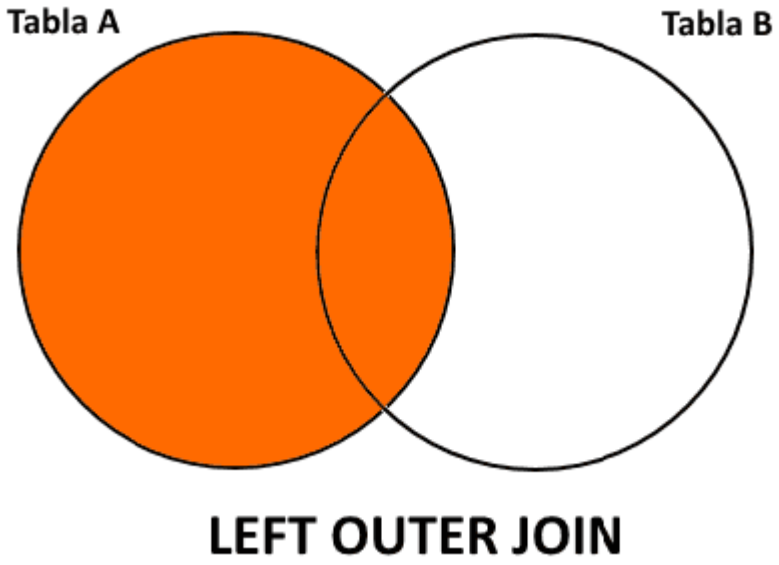
- En todas estas combinaciones externas el **uso de la palabra OUTER es opcional**.

❑ TIPOS OUTER JOIN
  LEFT OUTER JOIN o LEFT JOIN
  RIGHT OUTER JOIN o RIGHT JOIN
  FULL OUTER JOIN o FULL OUTER JOIN

LEFT OUTER JOIN

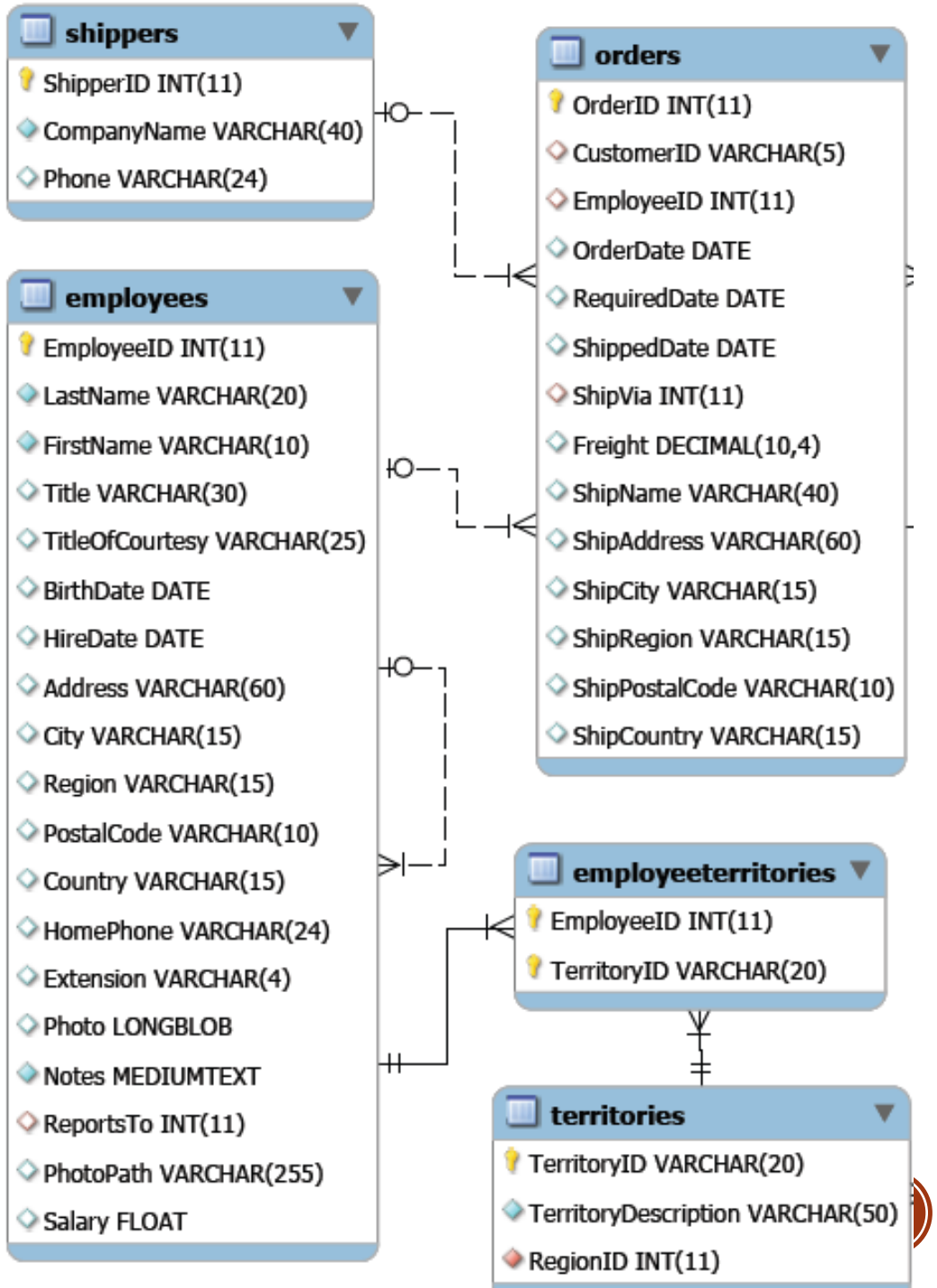
- Se obtienen todas las filas de la tabla colocada a la izquierda, aunque no tengan correspondencia en la tabla de la derecha

```
SELECT <lista_Campos>
FROM <TablaA A> LEFT [OUTER] JOIN <TablaB B> on a.key=b.key
```



EJEMPLO LEFT OUTER JOIN

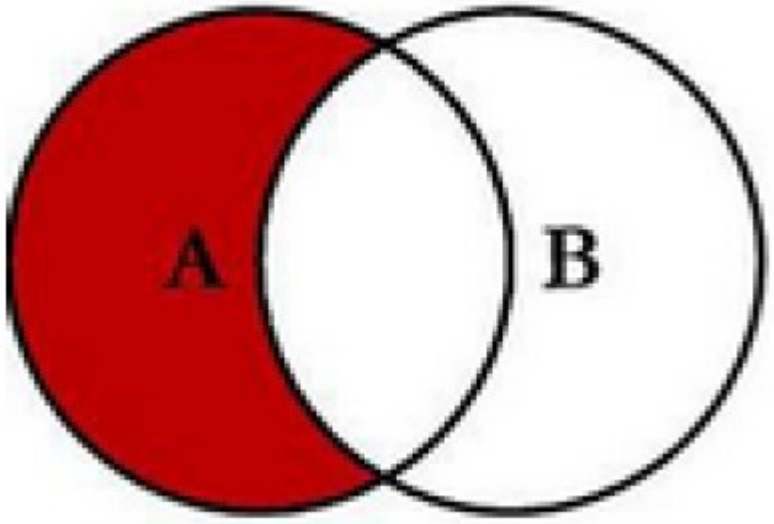
```
#se desea conocer que empleados han
atendido un pedido
independientemente si este lo ha
realizado o no

select OrderID,
     e.EmployeeID,
     LastName, FirstName
from Employees e left join orders o
on e.employeeid=o.employeeid;
```



LEFT OUTER JOIN (IS NULL)

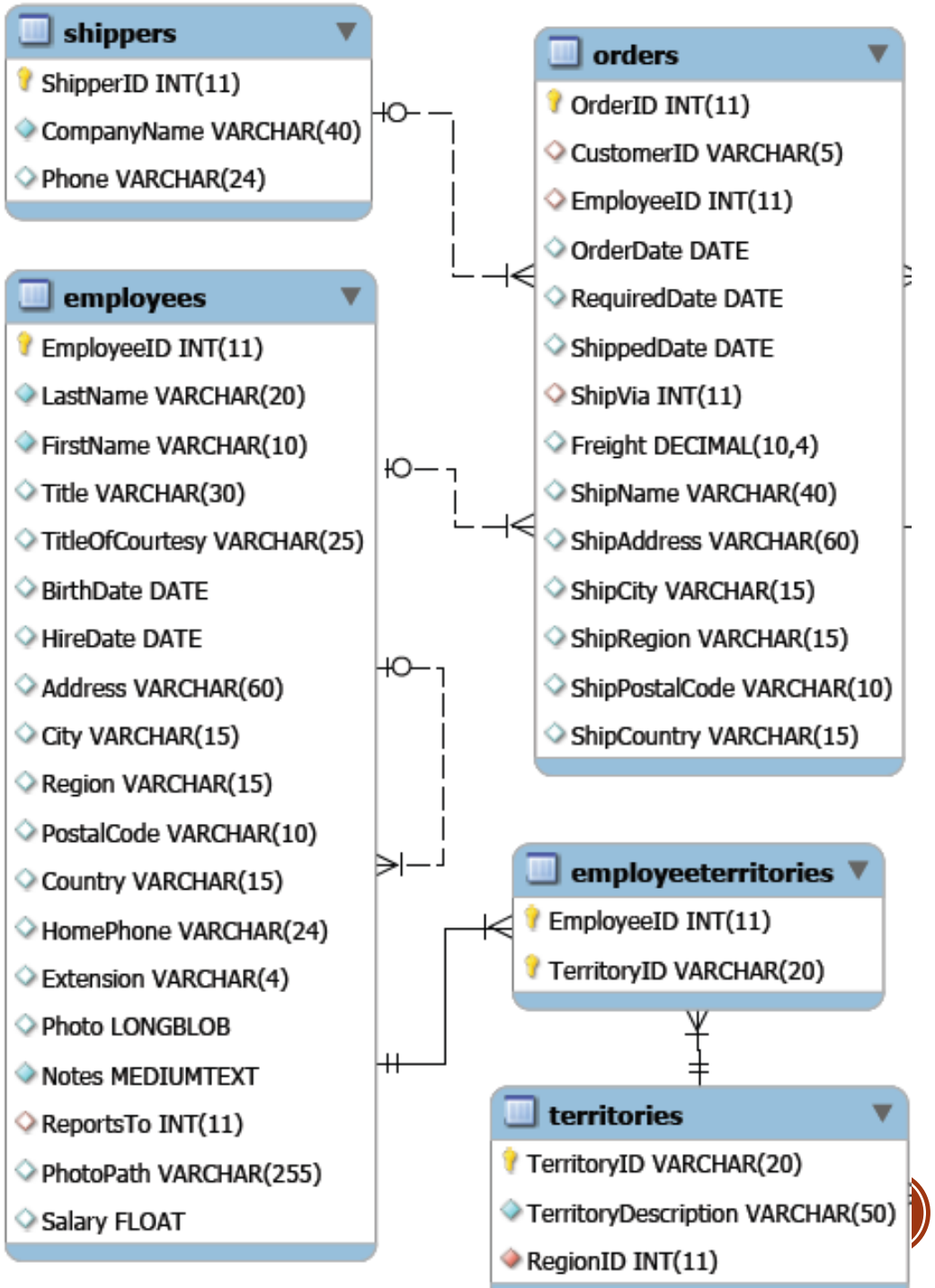
- Muestra los registros de la tabla izquierda menos los registros coincidentes con la tabla derecha

```
SELECT <lista_Campos>

FROM <TablaA A> LEFT [OUTER] JOIN <TablaB B> on a.key=b.key
WHERE B.Key is Null
```



EJEMPLO LEFT OUTER JOIN IS NULL

```
#Se desea conocer los empleados que
no han atendido ningún pedido

Select orderid, e.employeeid,
       firstname, lastname
from employees e left join orders o
on e.employeeid=o.employeeid
where o.employeeid is null;
```

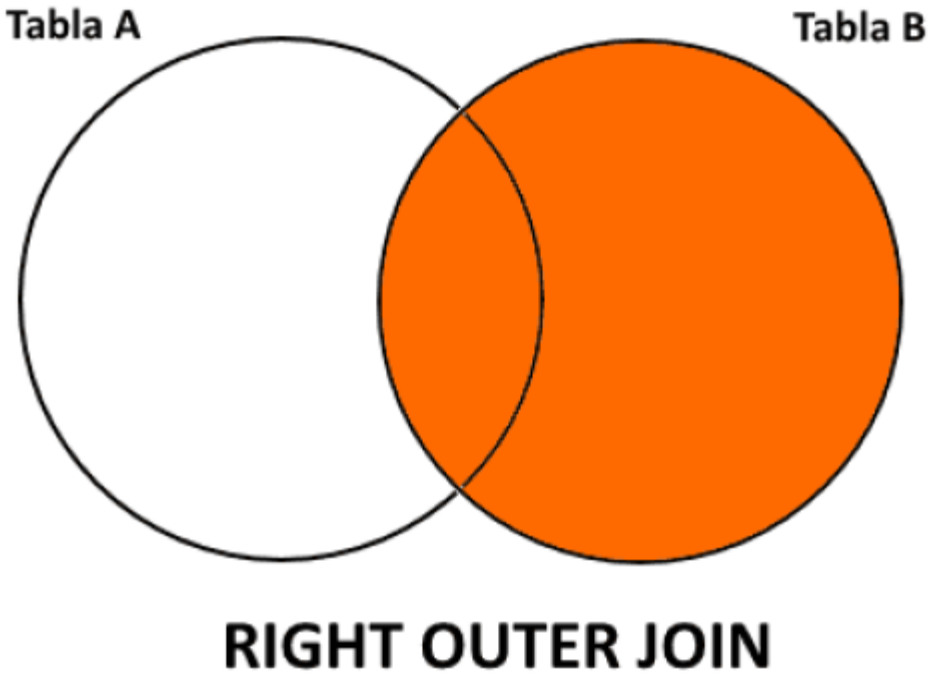


NOTA: incluir un nuevo empleado 10 para que no tenga pedidos

RIGHT OUTER JOIN

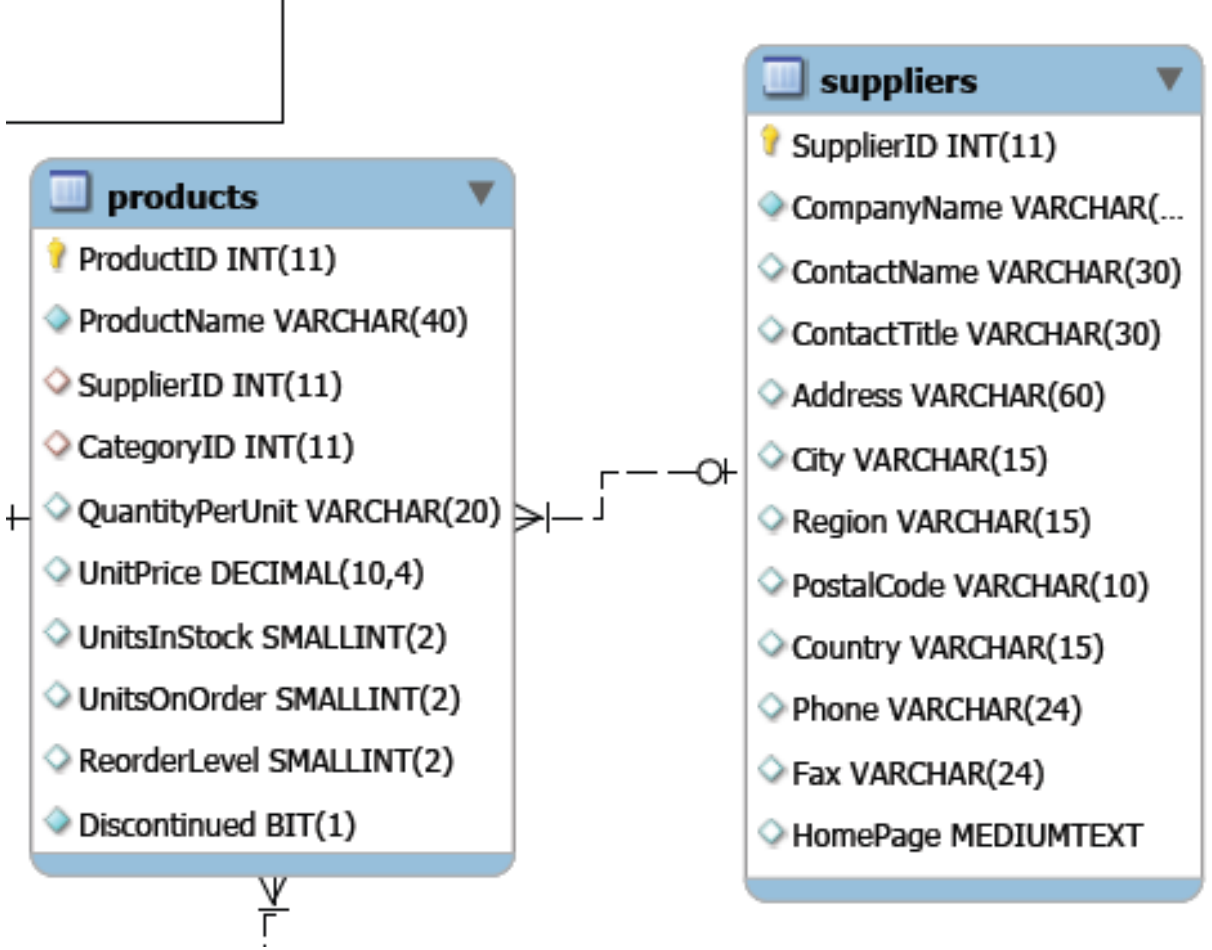
- Muestra los registros de la tabla derecha más los registros coincidentes con la tabla izquierda

```
SELECT <lista_campos>
FROM <TABLA A> RIGHT [OUTER] JOIN <TABLAB B> ON A.key=B.key
```



EJEMPLO RIGHT OUTER JOIN

```
#Mostrar que productos ofrece cada
proveedor

select productname, companyName,
     s.supplierid
from products p right
join suppliers s
on p.supplierid=s.supplierid
order by s.supplierid;
```

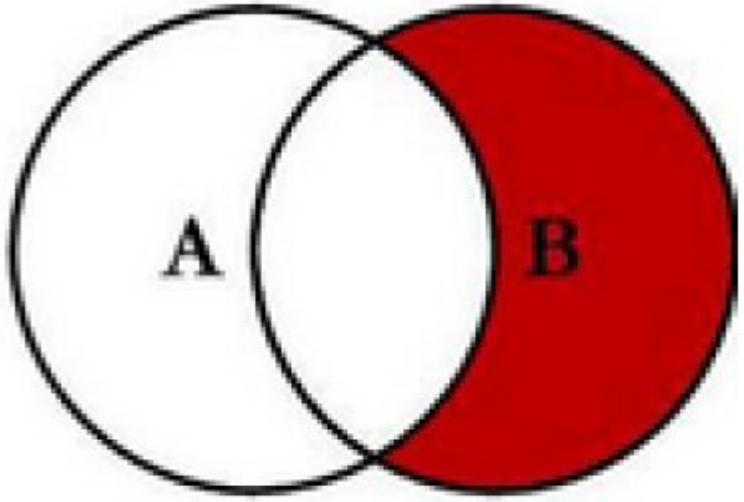


NOTA: incluir un nuevo proveedor 30 Mohou

RIGHT OUTER JOIN (IS NULL)

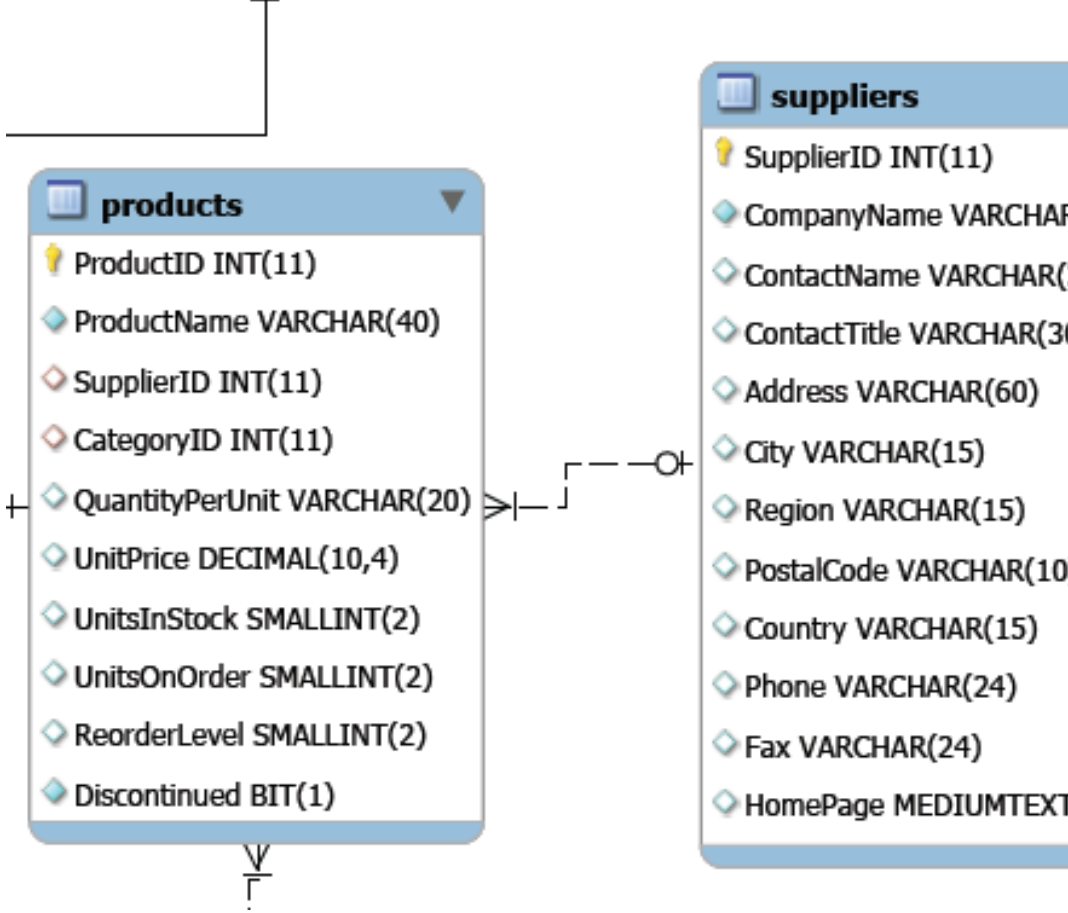
- Muestra los registros de la tabla derecha menos los registros coincidentes con la tabla izquierda

```
SELECT <lista_campos>
FROM <TABLA A> RIGHT [OUTER] JOIN <TABLAB B> ON A.key=B.key
WHERE a.key is null
```



EJEMPLO RIGHT OUTER JOIN (IS NULL)

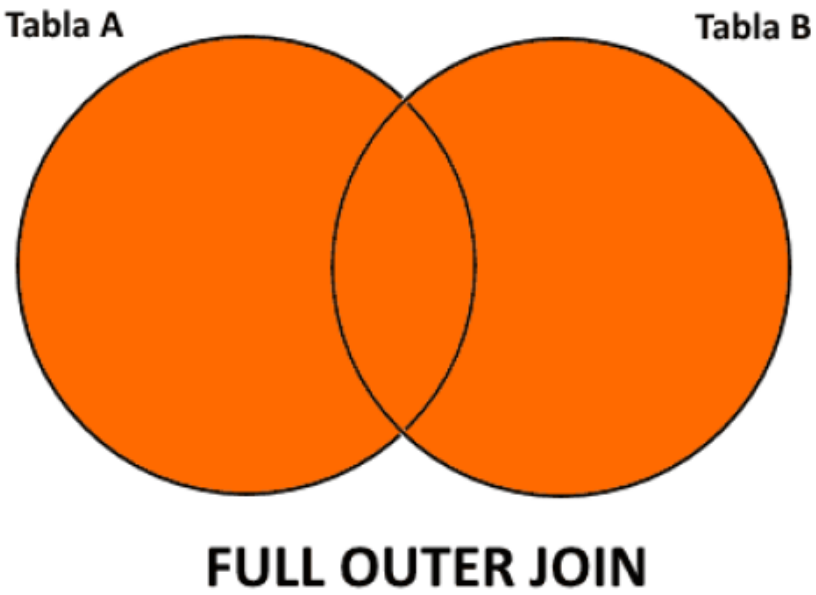
```
#Mostrar que proveedor no ha ofrecido
productos

select productname, companyName,
     s.supplierid
from products p right join suppliers s
on p.supplierid=s.supplierid
where p.supplierid is null
order by s.supplierid;
```



FULL OUTER JOIN

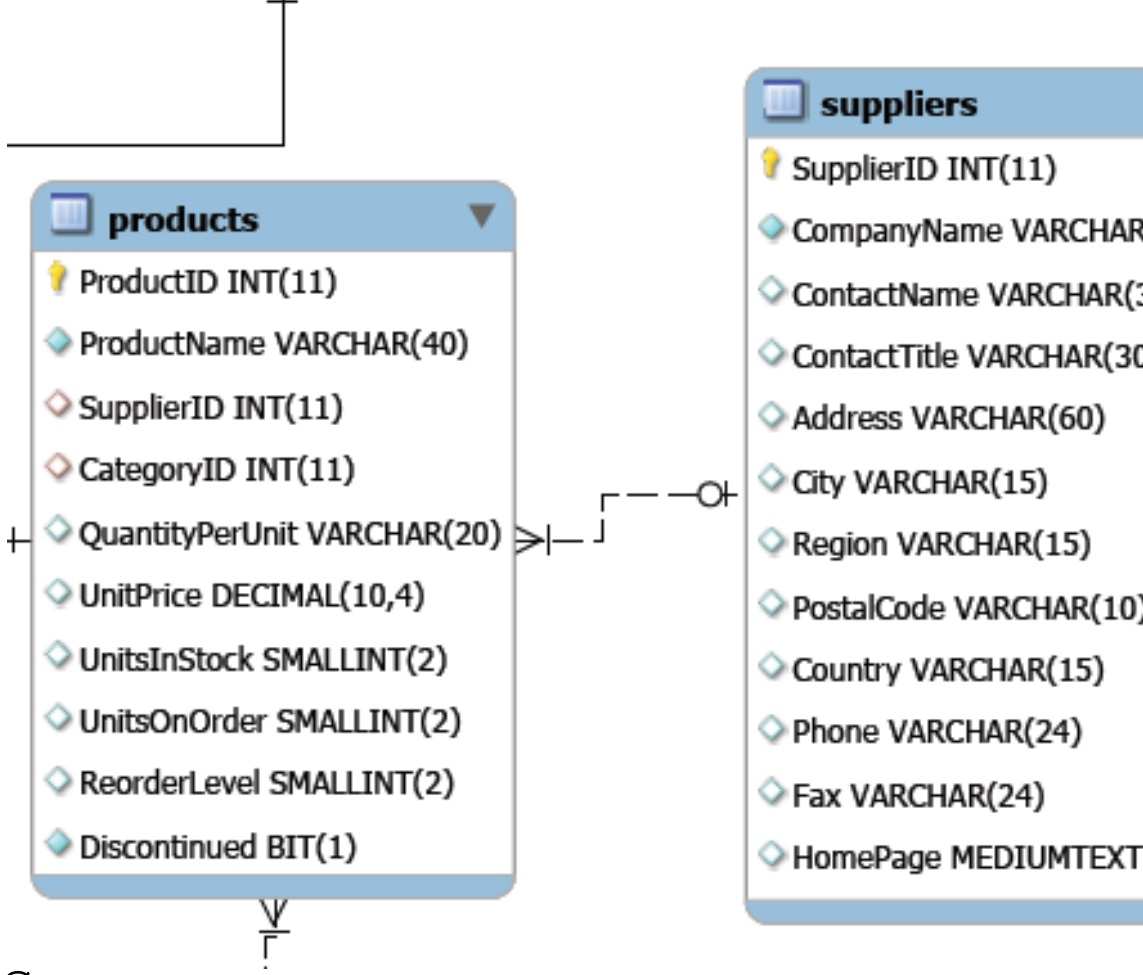
- Se obtienen todas las filas en ambas tablas, aunque no tengan correspondencia en la otra tabla. Es decir, todos los registros de A y de B aunque no haya correspondencia entre ellos, rellenando con nulos los campos que falten

```
SELECT <lista_campos>
FROM <TablaA A> FULL JOIN <TablaB B> ON A.Key=B.Key
```



EJEMPLO FULL OUTER JOIN

```
#Mostrar los productos que tengan o
no asignado un proveedor y los
proveedores independientemente si
estos han ofrecido o no un producto


select productname, companyName,
       s.supplierid
from products p full join suppliers s
on p.supplierid=s.supplierid;
```



OJO NO ES COMPATIBLE EN MYQL

CROSS JOIN

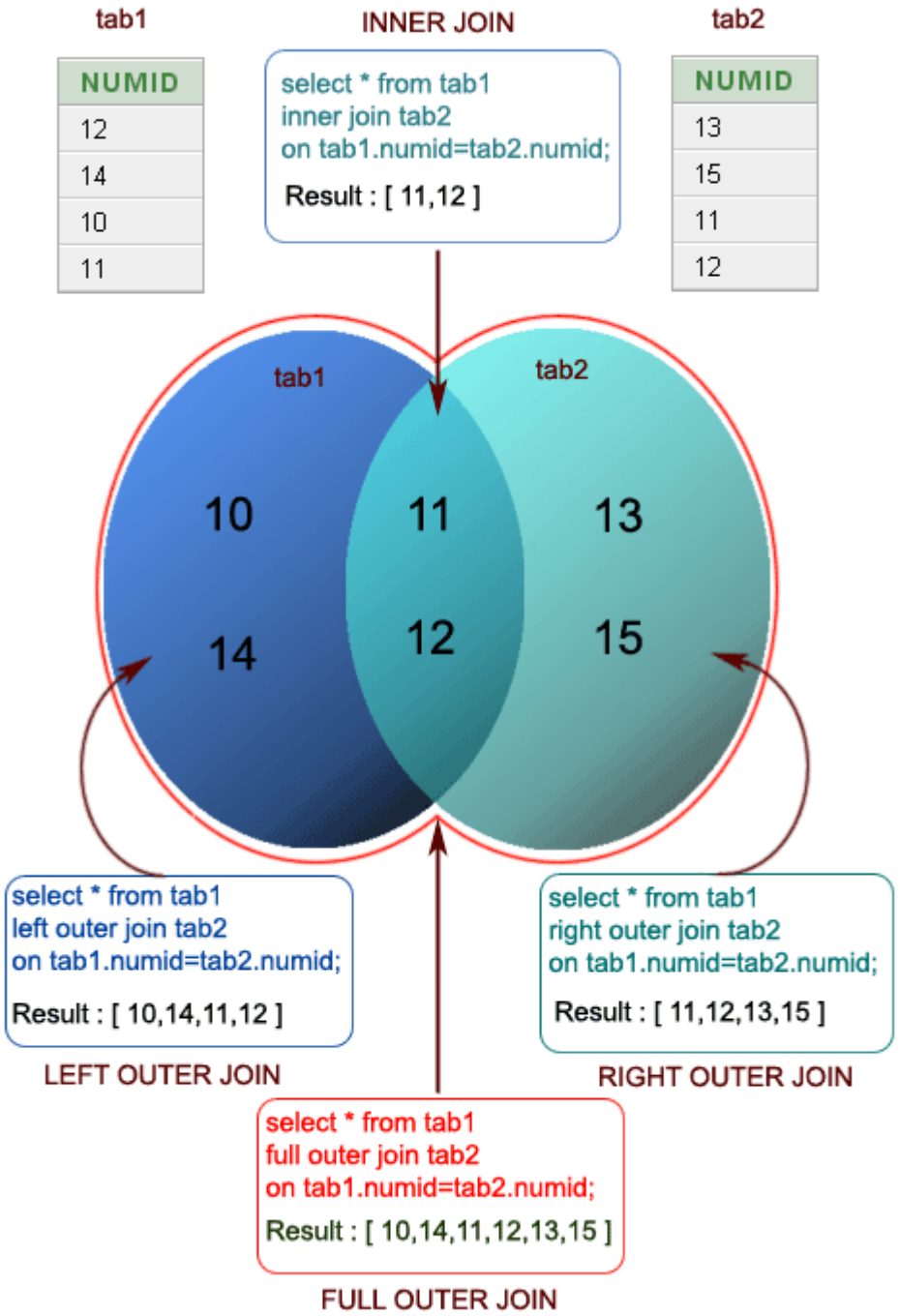
- Se obtienen todas las filas en ambas tablas, aunque no tengan correspondencia en la otra tabla. Es decir, todos los registros de A y de B aunque no haya correspondencia entre ellos, rellenando con nulos los campos que falten

```
SELECT <lista_campos>
FROM <TablaA A> CROSS JOIN <TablaB B>
```

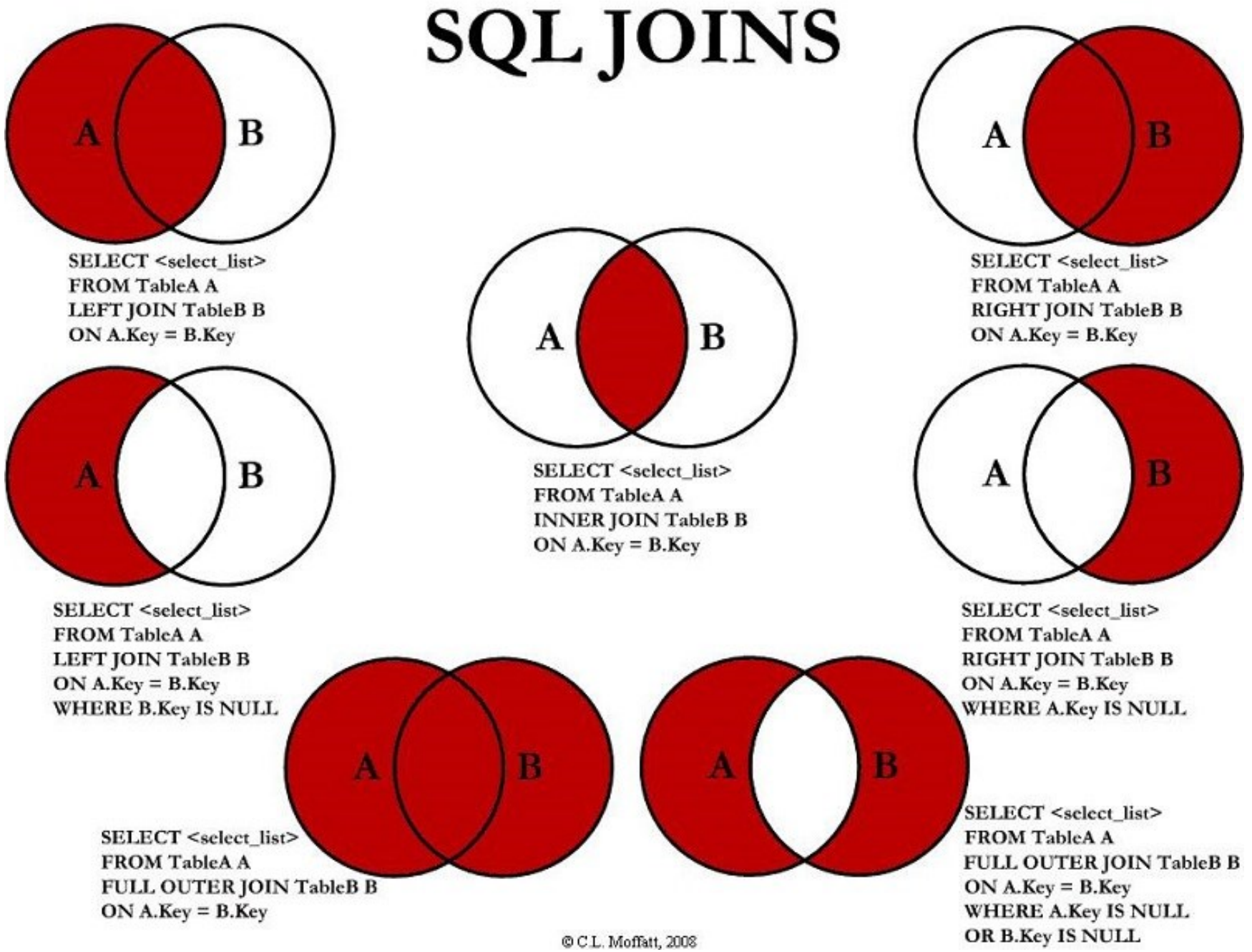
Ejemplo:

```
select productname, companyname
from products p cross join suppliers swhere p.supplierid=s.supplierid;
```

RESUMEN


tab1
NUMID
12
14
10
11
INNER JOIN
select * from tab1
inner join tab2
on tab1.numid=tab2.numid;
Result : [11,12]
tab2
NUMID
13
15
11
12
tab1
tab2
10
14
11
12
13
15
select * from tab1
left outer join tab2
on tab1.numid=tab2.numid;
Result : [10,14,11,12]
LEFT OUTER JOIN
select * from tab1
right outer join tab2
on tab1.numid=tab2.numid;
Result : [11,12,13,15]
RIGHT OUTER JOIN
select * from tab1
full outer join tab2
on tab1.numid=tab2.numid;
Result : [10,14,11,12,13,15]
FULL OUTER JOIN

RESUMEN


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008