```sql
/*        SOLUCIONES EJERCICIOS DML. LAS CONSULTAS MULTITABLA
*/
USE GestionSimples;

-- 2. Listar los códigos y nombres de los empleados de las oficinas del Este con su
oficina y ciudad.

SELECT  numemp, nombre, empleados.oficina, ciudad
FROM oficinas INNER JOIN empleados ON oficinas.oficina = empleados.oficina
WHERE region ='Este';

-- 3. Listar todos los pedidos mostrando su número, importe, nombre de cliente, y el
límite de crédito del cliente correspondiente.

SELECT numpedido, importe, clientes.nombre AS Cliente, limitecredito
FROM pedidos INNER JOIN clientes ON clie=numclie;

-- 4. Listar todos los empleados y la ciudad y región donde trabaja.

SELECT  numemp, nombre, edad, titulo, CONVERT(CHAR(8),contrato,3)AS contrato, jefe,
empleados.oficina,ciudad, region
FROM oficinas RIGHT JOIN empleados ON oficinas.oficina = empleados.oficina;

-- 5. Listar todas las oficinas y los nombres y títulos de sus directores.

SELECT  oficinas.*, nombre AS director, titulo
FROM oficinas LEFT JOIN empleados ON dir = numemp;

-- 6. Listar las oficinas con objetivo superior a 60.000 euros indicando para cada
una el nombre de su director.

SELECT  oficinas.*, nombre AS director
FROM oficinas LEFT JOIN empleados ON dir = numemp
WHERE objetivo > 60000;

-- 7. Listar todos los pedidos, mostrando el precio y  la descripción del producto.

SELECT  pedidos.*, precio, descripcion
FROM pedidos INNER JOIN productos ON fab = idfab AND producto = idproducto;

-- 8. Listar los pedidos superiores a 250 euros, incluyendo el nombre del vendedor
que tomó el pedido y el nombre del cliente que lo solicitó.

SELECT  numpedido, CONVERT(CHAR(8),fechapedido,3) AS fechapedido, clie, rep, fab,
producto, cant, importe, clientes.nombre AS cliente, empleados.nombre AS vendedor
FROM (pedidos INNER JOIN empleados ON rep = numemp)
                    INNER JOIN clientes ON clie = numclie
WHERE importe > 250;

-- 9. Listar los pedidos superiores a 250 euros, mostrando el nombre del cliente que
solicitó el pedido y el nombre del vendedor asignado a ese cliente.

SELECT  pedidos.*, clientes.nombre AS cliente, empleados.nombre AS [vendedor
asignado]
FROM (pedidos INNER JOIN clientes ON clie = numclie)
                    INNER JOIN empleados ON repclie = numemp
WHERE importe > 250;
```

```sql
-- 10. Listar los pedidos superiores a 250 euros, mostrando además el nombre del
cliente que solicitó º  el pedido y el nombre del vendedor asignado a ese cliente y
la ciudad de la oficina donde el vendedor trabaja.

SELECT  numpedido, clie, rep, clientes.nombre AS cliente, repclie, empleados.nombre
AS vendedor, ciudad
FROM ((pedidos INNER JOIN clientes ON clie = numclie)
            INNER JOIN empleados ON repclie = numemp)
            LEFT JOIN oficinas ON empleados.oficina=oficinas.oficina
WHERE importe > 250;

-- 11. Hallar los pedidos recibidos los días en que un nuevo empleado fue
contratado.

SELECT numpedido, fechapedido, rep, numemp, nombre, contrato
FROM pedidos,  empleados
WHERE fechapedido=contrato;

-- Cuando comparamos fechas puede que tengamos problemas si una de ellas tiene hora,
en este caso no hay problema porque tanto en fechapedido como en contrato la hora
está a cero. Si queremos asegurarnos de que la hora no nos dará problemas comparamos
la parte de fecha en vez de todo el campo:
-- SELECT *
-- FROM pedidos,  empleados
-- WHERE CONVERT(CHAR(10),fechapedido,103)=CONVERT(CHAR(10),contrato,103)

-- 12. Hallar los empleados que realizaron su primer pedido el mismo día  que fueron
contratados.

SELECT numemp, nombre, contrato, numpedido, rep, fechapedido
FROM pedidos INNER JOIN empleados ON rep = numemp
WHERE fechapedido = contrato;

-- 13. Listar los empleados que tienen una cuota superior al objetivo de al menos
una oficina. La oficina puede ser cualquiera no tiene por que ser la del empleado.

SELECT numemp, nombre, cuota, empleados.oficina AS [Su oficina], oficinas.oficina,
objetivo
FROM empleados, oficinas
WHERE cuota > objetivo

--14. Mostrar de cada empleado su código, nombre, ventas, oficina y ciudad en la que
está ubicada su oficina.

SELECT numemp, nombre,empleados.ventas,empleados.oficina,ciudad
FROM empleados LEFT JOIN oficinas ON empleados.oficina=oficinas.oficina;

--15. Listar de cada empleado su código, nombre, cuota, y código, nombre y cuota de
su jefe.

SELECT empleados.numemp, empleados.nombre, empleados.cuota, empleados.jefe,
jefes.nombre AS [Nombre jefe], jefes.cuota AS [Cuota jefe]
FROM empleados LEFT JOIN empleados jefes ON empleados.jefe = jefes.numemp;
```

```sql
-- 16. Listar los empleados con una cuota superior a la de su jefe, los campos a
obtener son el número, nombre y cuota del empleado y número, nombre y cuota de su
jefe.

SELECT empleados.numemp, empleados.nombre, empleados.cuota, empleados.jefe,
jefes.nombre, jefes.cuota
FROM empleados INNER JOIN empleados jefes ON empleados.jefe = jefes.numemp
WHERE empleados.cuota > jefes.cuota;

-- 17. Desde el entorno gráfico cambia el empleado 111, asígnale el jefe 110 y la
oficina 21. Después cambia la sentencia anterior para que salgan también
los empleados cuyo jefe no tenga cuota.

SELECT empleados.numemp, empleados.nombre, empleados.cuota, empleados.jefe,
jefes.nombre, jefes.cuota
FROM empleados INNER JOIN empleados jefes ON empleados.jefe = jefes.numemp
WHERE empleados.cuota > jefes.cuota OR
                      (empleados.cuota IS NOT NULL AND jefes.cuota IS NULL)

-- Se podía haber utilizado la función ISNULL:
-- SELECT empleados.numemp, empleados.nombre, empleados.cuota, empleados.jefe,
jefes.nombre, jefes.cuota
-- FROM empleados INNER JOIN empleados jefes ON empleados.jefe = jefes.numemp
-- WHERE empleados.cuota > ISNULL(jefes.cuota,0)
-- ISNULL(expresion, valor) si expresion no es nula devuelve expresion sino valor.

-- 18. Listar los empleados que no están asignados a la misma oficina que su jefe,
queremos ñumero, nombre y número de oficina tanto del empleado como se su jefe.

SELECT  e.numemp, e.nombre, e.oficina, e.jefe, j.nombre as [nombre jefe], j.oficina
AS [oficina jefe]
FROM empleados e INNER JOIN empleados j ON e.jefe = j.numemp
WHERE e.oficina <> j.oficina;

-- 19. En el punto anterior no salen los que no tienen oficina, cambiar la sentencia
-- para que aparezcan.

SELECT  e.numemp, e.nombre, e.oficina, e.jefe, j.nombre as [nombre jefe], j.oficina
AS [oficina jefe]
FROM empleados e INNER JOIN empleados j ON e.jefe = j.numemp
WHERE e.oficina <> j.oficina OR e.oficina IS NULL OR j.oficina IS NULL;

-- 20. Lo mismo que la anterior pero queremos que aparezca también la ciudad de las
oficinas (tanto del empleado como de su jefe).

SELECT e.numemp, e.nombre, e.oficina, ofiemp.ciudad, e.jefe, j.nombre as [nombre
jefe], j.oficina AS [oficina jefe], ofijefe.ciudad
FROM (oficinas ofiemp RIGHT JOIN empleados e ON ofiemp.oficina= e.oficina)
           INNER JOIN
(empleados j LEFT JOIN oficinas ofijefe ON j.oficina = ofijefe.oficina)
                           ON e.jefe = j.numemp
WHERE e.oficina <> j.oficina OR e.oficina IS NULL OR j.oficina IS NULL;
```

```sql
-- 21. Listar las oficinas (código) que no tienen empleados, resolver el problema de
dos formas (con JOIN y sin utilizar la composición de tablas).

SELECT oficinas.oficina
FROM oficinas LEFT JOIN empleados ON oficinas.oficina=empleados.oficina
WHERE numemp IS NULL;

-- o bien sin utilizar la composición de tablas

SELECT oficina
FROM oficinas
EXCEPT
SELECT oficina
FROM empleados;

-- 22. Obtener los productos (código completo) que no aparecen en ningún pedido.

SELECT idfab, idproducto
FROM productos
EXCEPT
SELECT fab, producto
FROM pedidos;

-- 23. Obtener los empleados de GestionSimples que aparecen en GestionA con otra
oficina.

SELECT numemp, oficina
FROM gestionsimples.dbo.empleados
EXCEPT
SELECT numemp, oficina
FROM gestionA.dbo.empleados;

-- 24. Obtener los empleados de GestionSimples que aparecen en GestionA con misma
oficina.

SELECT numemp, oficina
FROM gestionsimples.dbo.empleados
INTERSECT
SELECT numemp, oficina
FROM gestionA.dbo.empleados;
```