

```

/*          SOLUCIONES EJERCICIOS DML. LAS CONSULTAS DE RESUMEN
*/
USE GestionSimples;

--1. ¿Cuál es la cuota media y las ventas medias de los empleados?
SELECT AVG(cuota) AS [Cuota media], AVG(ventas) AS [Ventas medias]
FROM empleados;

-- 1.2
SELECT AVG(cuota) AS [Cuota media], ROUND(AVG(ventas),2) AS [Ventas medias]
FROM empleados;

--2 ¿De media, cuánto superávit obtienen los empleados, considerando superávit la
diferencia entre lo vendido y su cuota?

SELECT AVG(ventas-cuota) AS [Superávit medio]
FROM empleados;

--3. ¿Cuál es el total de cuotas y total de ventas de todos los empleados?

SELECT SUM(cuota) AS [Total cuotas], SUM(ventas) AS [Total ventas]
FROM empleados;

--4. ¿Cuántas oficinas tenemos en Valencia?
SELECT COUNT(*) AS Valencianas
FROM oficinas
WHERE ciudad = 'Valencia';

--5. ¿Cuántas oficinas tienen empleados?
SELECT COUNT(DISTINCT oficina) AS [Oficinas con empleados]
FROM empleados;
-- Si no incluimos DISTINCT obtenemos el nº de valores no nulos que tenemos en la
columna oficina es decir nº de empleados asignados a una oficina.

--6. ¿Cuántos empleados superan su cuota (tienen superávit positivo)?

SELECT count(*) AS [Empleados buenos]
FROM empleados
WHERE ventas > cuota ;

--7. ¿Cuántas oficinas tienen empleados que superan su cuota (tienen superávit
positivo)?

SELECT count(DISTINCT oficina) AS [Oficinas buenas]
FROM empleados
WHERE ventas > cuota ;

--8.¿Cuál es el importe total de los pedidos realizados por el empleado Luis
Antonio?

SELECT SUM(importe) AS [Importe vendido]
FROM pedidos INNER JOIN empleados ON rep = numemp
WHERE nombre = 'Luis Antonio';

```

-- Hay que tener en cuenta que hemos utilizado para seleccionar un campo que admite duplicados (nombre) por lo que si tenemos dos empleados Luis Antonio sus pedidos se agruparán.

--9. Hallar el precio medio de los productos del fabricante aci.

```
SELECT AVG(precio) AS [precio medio]
FROM productos
WHERE idfab = 'aci';
```

--9.2 Repetir la consulta para el fabricante Bic.

```
SELECT AVG(precio) AS [precio medio]
FROM productos
WHERE idfab = 'bic';
```

--10. Calcular el importe medio de los pedidos realizados por el cliente 2103.

```
SELECT AVG(importe) AS [importe medio]
FROM pedidos
WHERE clie = 2103;
```

--11. ¿Entre qué cuotas se mueven los empleados?

```
SELECT MIN(cuota) AS [Cuota mínima], MAX(cuota) AS [Cuota máxima]
FROM empleados;
```

--12. Hallar en qué fecha se realizó el primer pedido.

```
SELECT CONVERT(CHAR(10),MIN(fechapedido),103) AS [Primer pedido]
FROM pedidos;
```

--13. Hallar el mejor superávit de todos los empleados.

```
SELECT MAX(ventas - cuota) AS [Mejor superavit]
FROM empleados;
```

--14. Hallar cuántos pedidos hay de más de 250 euros.

```
SELECT COUNT(*) AS [Superiores a 250]
FROM pedidos
WHERE importe > 250;
```

--15. ¿Cuántos títulos de empleados se usan?

```
SELECT COUNT(DISTINCT titulo) AS [Cuántos títulos]
FROM empleados;
```

--16. ¿Cuántas oficinas tienen empleados cuyas ventas superan el objetivo de su oficina?

```
SELECT COUNT(DISTINCT empleados.oficina) AS Cuántas
FROM empleados INNER JOIN oficinas ON empleados.oficina = oficinas.oficina
WHERE empleados.ventas > objetivo;
```

--17. Hallar el importe medio de pedidos realizados por cada empleado.

```
SELECT rep, AVG(importe) AS [Importe medio]
FROM pedidos
GROUP BY rep;
```

--18. Repetir la consulta pero para que también salgan los empleados que no han vendido nada.

```
SELECT numemp, AVG(importe) AS [Importe medio]
FROM pedidos RIGHT JOIN empleados ON rep = numemp
GROUP BY numemp;
```

--19. ¿Cuál es en cada una de las oficinas, el rango de cuotas asignadas a sus vendedores?

```
SELECT oficinas.oficina, MIN(cuota) AS [Cuota mínima], MAX(cuota) AS [Cuota máxima]
FROM oficinas LEFT JOIN empleados ON empleados.oficina = oficinas.oficina
GROUP BY oficinas.oficina;
```

--20. Listar cuántos empleados están asignados a cada oficina, incluso las que no tienen.

```
SELECT oficinas.oficina, COUNT(numemp) AS Empleados
FROM oficinas LEFT JOIN empleados ON empleados.oficina = oficinas.oficina
GROUP BY oficinas.oficina;
```

--21. Listar cuántos clientes son atendidos por cada empleado (el cliente ha realizado un pedido con él).

```
SELECT rep, COUNT(DISTINCT clie) AS [Clientes atendidos]
FROM pedidos
GROUP BY rep;
```

--22. De cada vendedor (todos) queremos saber su nombre y el importe total vendido.

```
SELECT numemp, nombre, SUM(importe) AS [Importe vendido]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, nombre;
```

--22.2 Rectificar la consulta para que no aparezca la palabra NULL

```
SELECT numemp, nombre, ISNULL(SUM(importe),0) AS [Importe vendido]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, nombre;
```

--23 Para cada empleado cuyos pedidos suman más de 300 euros, hallar el importe medio vendido.

```
SELECT rep, AVG(importe) AS [Importe medio]
FROM pedidos
GROUP BY rep
HAVING SUM(importe) > 300;
```

--23.2. Rectifica la consulta para que aparezca también el nombre del empleado.

```
SELECT rep, nombre, AVG(importe) AS [Importe medio]
FROM empleados INNER JOIN pedidos ON numemp = rep
GROUP BY rep, nombre
```

```
HAVING SUM(importe) > 300;
```

--24. Por cada oficina con 2 o más personas, calcula la cuota total y las ventas totales de todos los empleados que trabajan en la oficina, de la oficina queremos el número y la ciudad.

```
SELECT oficinas.oficina, ciudad, SUM(cuota) AS [Cuota total], SUM(empleados.ventas) AS ventas
FROM empleados INNER JOIN oficinas ON empleados.oficina = oficinas.oficina
GROUP BY oficinas.oficina, ciudad
HAVING COUNT(*) >= 2;
```

--25. De cada empleado, obtener el importe vendido a cada cliente.

```
SELECT rep, clie AS Cliente, SUM(importe) AS [Importe vendido]
FROM pedidos
GROUP BY rep, clie;
```

--26. Repetir la consulta anterior pero ahora deben aparecer también los empleados que no han vendido nada.

```
SELECT numemp, clie, SUM(importe) AS [Importe vendido]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie
ORDER BY numemp, clie;
```

--27. Repetir la consulta pero ahora debe aparecer también el total de cuánto ha vendido cada empleado.

```
SELECT numemp, clie, SUM(importe) AS [Importe vendido]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie WITH ROLLUP
ORDER BY numemp, clie;
```

--27.2. En los resultados anteriores no se distinguen bien las líneas que corresponden a totales. ...

```
SELECT numemp, clie, SUM(importe) AS [Importe vendido], GROUPING(clie) AS [Agrupa clie], GROUPING(numemp) AS [Agrupa numemp]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie WITH ROLLUP
ORDER BY numemp, clie;
```

--27.3 Ahora modifica la consulta para que las filas de totales aparezcan más claras:

```
SELECT numemp, clie, SUM(importe) AS [Importe vendido], CASE GROUPING(clie) WHEN 0 THEN ' ' WHEN 1 THEN 'Total empleado' END AS [Agrupa clie], CASE GROUPING(numemp) WHEN 0 THEN ' ' ELSE 'Total Final' END AS [Agrupa numemp]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie WITH ROLLUP
ORDER BY numemp, clie;
```

--27.4 Ahora queremos colocar las columnas Agrupa delante de las demás columnas:

```
SELECT CASE GROUPING(numemp) WHEN 0 THEN ' ' ELSE 'Total Final' END AS [Agrupa numemp], CASE GROUPING(clie) WHEN 0 THEN ' ' WHEN 1 THEN 'Total empleado' END AS [Agrupa clie], numemp, clie, SUM(importe) AS [Importe vendido]
```

```
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie WITH ROLLUP
ORDER BY numemp, clie;
```

--27.5 Ahora queremos que "Total empleado" aparezca en la columna clie:

```
SELECT CASE GROUPING(numemp) WHEN 0 THEN ' ' ELSE 'Total Final' END AS [Agrupa
numemp], numemp, CASE GROUPING(clie) WHEN 0 THEN CONVERT(CHAR(4),clie) WHEN 1 THEN
'Total empleado' END AS [Clie], SUM(importe) AS [Importe vendido]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie WITH ROLLUP
ORDER BY numemp, clie;
```

--27.6 El empleado 104 (y otros) no ha vendido a nadie y por sale en la columna clie la palabra NULL, queremos que en estos casos no aparezca nada (se deje en blanco).

```
SELECT CASE GROUPING(numemp) WHEN 0 THEN ' ' ELSE 'Total Final' END AS [Agrupa
numemp], numemp, CASE WHEN GROUPING(clie) = 1 THEN 'Total empleado' WHEN clie IS
NULL THEN ' ' ELSE CONVERT(CHAR(4),clie) END AS [Clie], ISNULL(SUM(importe),0) AS
[Importe vendido]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie WITH ROLLUP
ORDER BY numemp, clie;
```

--27.7 Lo rematamos para que el resultado quede así:

```
SELECT CASE GROUPING(numemp) WHEN 0 THEN ' ' ELSE 'Total Final' END AS [ ],
ISNULL(CONVERT(CHAR(3),numemp),'... ') AS [numemp], CASE WHEN GROUPING(clie) = 1 AND
GROUPING(numemp) = 0 THEN 'Total empleado' WHEN GROUPING(clie) = 1 AND
GROUPING(numemp) = 1 THEN '...' WHEN clie IS NULL THEN ' ' ELSE
CONVERT(CHAR(4),clie) END AS [Clie], ISNULL(SUM(importe),0) AS [Importe vendido]
FROM empleados LEFT JOIN pedidos ON numemp = rep
GROUP BY numemp, clie WITH ROLLUP
ORDER BY numemp, clie;
```