

```
/*          SOLUCIONES EJERCICIOS DML. LAS CONSULTAS SIMPLES
*/
```

Una vez hayas descargado y extraído los archivos del zip, abre el Sql Server Management Studio.

1.- Para añadir la base de datos utiliza la opción Adjuntar vista en el tema anterior. Para cambiar el nombre lógico de la base de datos dentro del servidor, en la ventana donde seleccionas la base de datos a adjuntar, agrega la base de datos a adjuntar y cambia el campo *Adjuntar* como escribiendo el nombre lógico que quieres que tenga GestionSimples.

2.- Para copiar el pedido de código 13, abre la tabla pedidos, selecciona la fila haciendo clic en la columna de la izquierda sobre la fila del pedido 13, Ctrl+C (para copiar), haz clic sobre la primera columna de la última fila (la que tiene el asterisco y sirve para añadir nuevas filas), quedará seleccionada, a continuación Ctrl+V (Pegar), y antes de salir de la fila rectifica los campos *codigo* y *numpedido*. Si quieres cancelar la introducción de datos pulsa la tecla ESC.

El resto de los ejercicios se hace creando una nueva consulta. Ojo con la base de datos sobre la que se ejecuta la consulta. Para evitar problemas añade un *USE GestionSimples* al principio de la consulta.

Puedes escribir y guardar todos los ejercicios en la misma consulta. Para ejecutar un determinado ejercicio selecciona primero la(s) instrucción(es) que quieres que se ejecute(n) antes de pulsar el botón Ejecutar, o bien puedes utilizar bloques de comentarios para comentar por ejemplo todas las instrucciones que ya has probado y funcionan.

```
-- 3. Obtener los datos de los empleados.
```

```
SELECT *
FROM empleados;
```

```
-- 4. Listar de cada empleado su código, nombre, oficina, y fecha de
contrato (fecha corta).
```

```
SELECT numemp, nombre, oficina, CONVERT (CHAR(8),contrato,3) AS [Fecha
contrato]
FROM empleados;
```

```
-- 5. Listar de cada pedido, su número, código completo del artículo
vendido y precio unitario al que se ha vendido.
-- precio sin redondear
```

```
SELECT numpedido, fab, producto, importe/cant AS Precio
FROM pedidos;
```

```
-- 6. Lo mismo que la anterior pero con precios redondeados.
```

```
SELECT numpedido, fab, producto, round(importe/cant,2) AS Precio
FROM pedidos;
```

```
-- 7. Obtener una lista de productos por fabricante.
```

```
SELECT idfab, descripcion, idproducto, productos.precio
FROM productos
ORDER BY idfab, descripcion;
```

```

-- 8. De cada producto queremos saber el id de fabricante, id de
producto, su descripción y el valor de sus existencias.

SELECT idfab,idproducto,descripcion,(existencias*precio) AS valoracion
FROM productos;

-- 9. Listar los empleados indicando código, nombre y día de la semana en
que fue contratado.

SELECT numemp, nombre, DATENAME(dw,contrato) AS [Día de contrato]
FROM empleados;

-- 10. Listar de cada empleado su código, nombre y título (sustituir la
abreviatura dir por la palabra completa Director).

SELECT numemp, nombre, REPLACE(titulo, 'Dir ', 'Director ') AS Cargo
FROM empleados;
-- Ojo con el espacio que hay que dejar después de la letra r.

-- 11. Listar los datos de los empleados que no tienen oficina asignada

SELECT *
FROM empleados
WHERE oficina IS NULL;

-- 12. Listar los códigos de las oficinas que tienen empleados asignados.

SELECT DISTINCT oficina
FROM empleados
WHERE oficina IS NOT NULL;

-- 13. Listar las oficinas de manera que las oficinas de mayores ventas
aparezcan en primer lugar.

SELECT ciudad, region, ventas
FROM oficinas
ORDER BY ventas DESC;

-- 14. Listar de cada región las oficinas por orden de mejores ventas.

SELECT region, ciudad, ventas
FROM oficinas
ORDER BY region, ventas DESC;

-- 15. Listar de cada jefe su código y el código y nombre de sus
subordinados ordenados por nombres

SELECT jefe, numemp, nombre
FROM empleados
WHERE jefe IS NOT NULL
ORDER BY jefe, nombre;

-- 16. ¿En qué ciudades tenemos oficinas?

SELECT DISTINCT ciudad
FROM oficinas
ORDER BY ciudad;

```

```

-- 17. Queremos saber el código del empleado(s) que ha realizado el
pedido más caro.
SELECT TOP 1 WITH TIES rep
FROM pedidos
ORDER BY importe DESC;
-- Ordenamos los pedidos de mayor a menor importe, nos quedamos con el
primero (con empates) y sacamos el código del representante.

-- 18. Listar las 3 mejores oficinas en cuanto a ventas.

SELECT TOP 3 oficina, ciudad, ventas
FROM oficinas
ORDER BY ventas DESC;

-- 19. Listar las 2 peores oficinas en cuanto a ventas.

SELECT TOP 2 WITH TIES oficina, ciudad, ventas
FROM oficinas
WHERE ventas IS NOT NULL
ORDER BY ventas;

-- 20. Obtener la mejor oficina en cuanto a ventas teniendo en cuenta su
objetivo,
-- puede tener menos ventas pero que supongan un mayor porcentaje del
objetivo.

SELECT TOP 1 WITH TIES oficina, ciudad, ventas, objetivo
FROM oficinas
ORDER BY ventas/objetivo DESC;

-- 21. Queremos saber los productos que tienen un precio superior o igual
al precio de la mitad de los productos.

SELECT TOP 50 PERCENT WITH TIES *
FROM productos
ORDER BY precio DESC

-- 22. Listar las oficinas que han alcanzado su objetivo.

SELECT *
FROM oficinas
WHERE ventas >= objetivo OR (ventas >0 AND objetivo IS NULL)

-- 23. Listar los pedidos con fechas erróneas (una fecha posterior a
hoy).

SELECT *
FROM pedidos
WHERE fechapedido > getdate();
-- NOTA: El resultado puede variar según la fecha en que se ejecuta la
instrucción.

```

```

-- 24. Listar los pedidos anteriores a 2008.

SELECT *
FROM pedidos
WHERE fechapedido < '01/01/2008';
-- o

SELECT *
FROM pedidos
WHERE year(fechapedido) < 2008;

-- 25. Listar los pedidos de este año.
SELECT *
FROM pedidos
WHERE year(fechapedido) = year(getdate());

-- 26. Listar las oficinas cuyas ventas no alcanzan el 50% de su
objetivo.

SELECT *
FROM oficinas
WHERE ventas < (.5 * objetivo) OR (ventas IS NULL AND objetivo IS NOT
NULL);

-- 27. Hallar las oficinas dirigidas por el empleado 108.

SELECT *
FROM oficinas
WHERE dir = 108;

-- 28. Listar los empleados que no tienen ventas.

SELECT numemp, nombre, ventas, cuota
FROM empleados
WHERE ventas = 0;
-- En la tabla el campo ventas no admite nulos.

-- 29. Listar los empleados que tienen ventas pero que no han alcanzado
su cuota.

SELECT numemp, nombre, ventas, cuota
FROM empleados
WHERE ventas < cuota AND ventas > 0;

-- 30. Hallar los empleados a cargo del empleado 106

SELECT numemp, nombre
FROM empleados
WHERE jefe = 106;

-- 31. Hallar los empleados que no están a cargo del empleado 106

SELECT numemp, nombre
FROM empleados
WHERE jefe <> 106 OR jefe IS NULL;

-- NOTA: si no añadimos la segunda parte de la pregunta los empleados que
no tienen jefe no salen.

```

```

-- 32. Listar las oficinas con ventas comprendidas entre 16.000 euros y
40.000.

SELECT *
FROM oficinas
WHERE ventas BETWEEN 16000 AND 40000;
--o
SELECT *
FROM oficinas
WHERE ventas >= 16000 AND ventas <= 40000;

-- 33. Listar los empleados que trabajan en las oficinas 11, 12 o 13:

SELECT numemp, nombre, oficina
FROM empleados
WHERE oficina IN (11,12,13);
-- o
SELECT numemp, nombre, oficina
FROM empleados
WHERE oficina = 11 OR oficina = 12 OR oficina = 13;
-- o
SELECT numemp, nombre, oficina
FROM empleados
WHERE oficina >= 11 AND oficina <= 13;

-- 34. Listar todos los nombres de empleados que empiecen por ju

SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE 'ju%';

-- 35. Listar los nombres de empleados que acaban en z.

SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE '%z';

-- 36. Listar los empleados cuyo nombre contiene on.

SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE '%on%';

-- 37. Listar los empleados cuyo nombre tiene una u como segunda letra.

SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE '_u%';

-- 38. Listar los empleados cuyo nombre empieza por cualquier letra de la
a a la k.

SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE '[a-k]';

```

-- 39. Listar los empleados cuyo nombre empieza por J

```
SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE 'J%';
```

-- 40. Listar los empleados cuyo nombre no empieza por J

```
SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE '[^J]%';
```

-- 41. Listar de cada empleado su código, nombre y título (si el título es representante deberá aparecer comercial
-- y si es director general deberá aparecer Gerente).

```
SELECT numemp, nombre, CASE titulo WHEN 'representante' THEN 'Comercial'
WHEN 'Director general' THEN 'Gerente' ELSE titulo END AS Cargo
FROM empleados;
```

-- 42. Lo mismo que la anterior pero cambiando el orden de las columnas, colocando la columna de título la primera y ordenando por ella.

```
SELECT CASE titulo WHEN 'representante' THEN 'Comercial' WHEN 'Dir
general' THEN 'Gerente' ELSE titulo END AS Cargo, numemp, nombre
FROM empleados
ORDER BY Cargo;
```

-- 43. Listar código, ciudad y región de cada oficina, si la oficina no tiene región deberá aparecer el texto "No tiene".

```
SELECT oficina, ciudad,
CASE WHEN region IS NULL THEN 'No tiene' ELSE region END AS región
FROM oficinas;
```

En este caso no podemos utilizar el formato utilizado en la sentencia anterior (CASE region WHEN NULL THEN 'No tiene' ELSE region END) porque WHEN NULL equivaldría a WHEN region = NULL y no sirve. Puedo utilizar *región* como alias de columna porque lleva acento y el nombre de la columna no lleva, así que son dos nombres diferentes.

-- 44. Listar código, nombre y oficina de cada empleado, si el empleado no tiene oficina deberá aparecer el texto "No tiene".

```
SELECT numemp, nombre,
CASE WHEN oficina IS NULL THEN 'No tiene' ELSE CAST(oficina AS CHAR(5))
END AS [Cod. Oficina]
FROM empleados;
```

En este caso queremos colocar en la columna oficina o un texto o el código de la oficina, pero este código es numérico, así que lo tenemos que convertir a texto para que todos los valores sean del mismo tipo, si no la función CASE da error.