

## Practicar con la selección de filas

Aquí practicaremos los puntos vistos respecto a la selección de filas que deben aparecer en el resultado final. Para ello existen distintas cláusulas, la cláusulas TOP y DISTINCT que una vez calculado el resultado ejercen un último filtro sobre las filas, y la cláusula WHERE que ejerce un primer filtro sobre las filas del origen.

Puedes probarlos utilizando la base de datos GestionA suministrada en la unidad 2.

Como en los ejemplos anteriores, tienes el enunciado seguido de la respuesta, utiliza una hoja de papel para tapar la solución e intenta redactar la respuesta sin mirarla.

Plantea la consulta como varias preguntas y en este orden:

- ¿Dónde están los datos que necesito? La respuesta me dará la FROM
- ¿Qué columnas quiero visualizar en el resultado? La respuesta me dará la lista de selección.
- ¿Me interesan todas las filas del origen?, si es que no, tengo que añadir un WHERE.
- ¿Tienen que aparecer todas las filas del resultado?
  - Si la respuesta es del tipo sólo las primeras necesito un TOP.
  - Si la respuesta es no quiero que salgan dos filas iguales necesito un DISTINCT.
- ¿El resultado tiene que estar ordenado? Si es que sí añado un ORDER BY.

1. Listar los códigos de empleado de los directores de las oficinas.

```
SELECT dir
FROM oficinas;
```

2. En el ejemplo anterior, si un mismo empleado dirige varias oficinas (por ejemplo el 108), su código aparece repetido en el resultado. Para evitarlo modifica la consulta.

```
SELECT DISTINCT dir
FROM oficinas;
```

3. Si un mismo empleado dirige oficinas en varias regiones quiero que aparezca una vez con cada una de las regiones en las que dirige.

```
SELECT DISTINCT dir, region
FROM oficinas;
```

4. Queremos el código, ciudad y ventas de las 10 mejores oficinas en cuanto a ventas.

Pista: ordenamos las oficinas por ventas de mayor a menor y sacamos las 10 primeras.

```
SELECT TOP 10 oficina, ciudad, ventas
FROM oficinas
ORDER BY ventas DESC;
```

5. En la instrucción anterior si hay varias oficinas que tienen las mismas ventas que la décima, puede que no aparezcan todas, cambia la instrucción para que sí aparezcan todas.

```
SELECT TOP 10 WITH TIES oficina, ciudad, ventas
FROM oficinas
ORDER BY ventas DESC;
```

Con la cláusula WITH TIES sacamos los empates.

6. Queremos poder saber de forma rápida cuáles son los 5 productos de menor precio.

```
SELECT TOP 5 *
FROM productos
ORDER BY precio;
```

Aquí tenemos un problema, hay productos sin precio (precio nulo) por lo que salen primero estos. Pero lo que nos interesa son los cinco precios más bajos, por lo que tenemos que eliminar de la consulta los productos que no tienen precio. Además lo que nos interesan son los precios, no los productos que tienen esos precios.

7. Modifica la consulta anterior para que se solucione el problema.

```
SELECT TOP 5 precio
FROM productos
WHERE precio IS NOT NULL
ORDER BY precio;
```

Se seleccionan los productos con precio (WHERE precio IS NOT NULL), se ordenan por precio (ORDER BY precio) y se sacan los cinco primeros (TOP 5), como nos interesan los precios no los productos asociados a estos precios, no hace falta sacar los empates (no ponemos WITH TIES).

8. Listar los “buenos” vendedores (los que han vendido más que lo que marca su cuota).

```
SELECT numemp, nombre, ventas, cuota
FROM empleados
WHERE ventas > cuota
```

9. Hallar vendedores contratados antes de 1988.

```
SELECT numemp, nombre, contrato
FROM empleados
WHERE contrato < '01/01/1988';
```

O bien

```
SELECT numemp, nombre, contrato
FROM empleados
WHERE YEAR(contrato) < 1988;
```

Observa que en la primera solución comparo la fecha del contrato con una fecha concreta (una constante de tipo fecha) por lo que hay que escribir la fecha entre comillas, mientras que en la segunda comparo el resultado de la función YEAR (un número) con un valor concreto que también tiene que ser un número por eso no lleva las comillas.

Es muy importante, cuando escribimos una comparación comprobar que lo que ponemos de cada lado del operador de comparación es del mismo tipo de datos, nos evitará muchos problemas.

10. Hallar oficinas cuyas ventas estén por debajo del 80% de su objetivo:

```
SELECT oficina
FROM oficinas
WHERE ventas < (.8 * objetivo);
```

Recuerda que  $(\text{objetivo} * 80) / 100$  es lo mismo que  $\text{objetivo} * 0.80$

$\text{Objetivo} * 8 / 100$  es lo mismo que  $\text{objetivo} * 0.08$

$\text{Base} + \text{Base} * 18 / 100$  es lo mismo que  $\text{base} * 1.18$  (siempre  $1 +$  el % que sumas)

$\text{Base} - \text{Base} * 5 / 100$  es lo mismo que  $\text{base} * 0.95$  (siempre  $1 -$  el % que descuentas)

En todos estos casos la segunda opción es mejor porque involucra una sólo operación aritmética (la multiplicación) mientras que en la primera se tiene que realizar una (una división) o dos más (una suma/resta y una división).

11. Hallar las oficinas dirigidas por el empleado 108:

```
SELECT oficina, dir
FROM oficinas
WHERE dir = 108;
```

Sacamos dir en la lista de selección para que aparezca en el resultado y así poder comprobar que funciona pero no es imprescindible, puedo tener el campo en el WHERE y no tenerlo en la lista de selección.

12. Hallar las oficinas no dirigidas por el empleado 108:

```
SELECT oficina, dir
FROM oficinas
WHERE dir <> 108; -- o bien WHERE NOT dir = 108
```

¡Ojo con este tipo de comparaciones! porque el valor NULL no cumple la condición y las oficinas que no tienen director no saldrán. Si queremos que salgan debemos añadir otra condición:

```
SELECT oficina, dir
FROM oficinas
WHERE dir <> 108 OR dir IS NULL;
```

13. Hallar vendedores cuyas ventas estén comprendidas entre 20.000 euros y 50.000.

```
SELECT numemp, nombre, ventas
FROM empleados
WHERE ventas >= 20000 AND ventas <= 50000;
```

¡Ojo con el separador de miles! En SQL NUNCA utilices el separador de miles y para separar la parte decimal de la parte entera utiliza SIEMPRE el punto.

O bien

```
SELECT numemp, nombre, ventas
FROM empleados
WHERE ventas BETWEEN 20000 AND 50000;
```

Esta solución es más fácil de leer e interpretar.

14. Listar el código y nombre de los empleados que no estén asignados a ninguna oficina.

```
SELECT numemp,nombre, oficina
FROM empleados
WHERE oficina IS NULL;
```

La oficina la sacamos para comprobar que salen los que tienen que salir.

15. Obtener los empleados que trabajan en las oficinas 11, 20 o 22:

```
SELECT oficina, numemp, nombre
FROM empleados
WHERE oficina = 11 OR oficina = 20 OR oficina = 22;
```

O bien

```
SELECT oficina, numemp, nombre
FROM empleados
WHERE oficina IN (11,20,22);
```

Esta solución es más fácil de leer e interpretar.

16. Obtener los empleados que NO trabajan en las oficinas 11, 20 o 22:

```
SELECT oficina, numemp, nombre
FROM empleados
WHERE oficina <> 11 AND oficina <> 20 AND oficina <> 22;
```

¡Ojo con el operador AND! Son los empleados que no son de la 11 NI (No y) de la 20 ni de la 22. Cuando se combinan varias condiciones, sobre todo con negaciones (distinto es no igual) es fácil equivocarse.

O bien

```
SELECT oficina, numemp, nombre
FROM empleados
WHERE oficina NOT IN (11,20,22);
```

17. Listar los vendedores asignados a alguna oficina.

```
SELECT numemp, nombre, oficina
FROM empleados
WHERE oficina IS NOT NULL;
```

18. Obtener el código y nombre de los empleados cuyo nombre empiece por An

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

19. Lo mismo pero de los empleados cuyo nombre acabe por ez.

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

20. Lo mismo pero de los empleados cuyo nombre contiene on.

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

21. Obtiene el código y nombre de los empleados cuyo nombre que empiece por cualquier letra de la *a* a la *d*

```
SELECT numemp,nombre
FROM empleados
WHERE nombre LIKE '[a-d]%' ;
```

Es equivalente a escribir:

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

22. Listar los nombres de los empleados que no han alcanzado su cuota y cuyas ventas sean inferiores a 150.000€.

```
SELECT nombre
FROM empleados
WHERE ventas < cuota AND ventas < 150000;
```

23. Hallar las oficinas no dirigidas por el empleado 108

```
SELECT oficina, dir
FROM oficinas
WHERE NOT dir = 108 OR dir IS NULL;
```

Tenemos que añadir la última condición (OR dir IS NULL) si queremos que salgan las oficinas que no tienen director. Para esas oficinas la condición dir = 108 da como resultado NULL, por lo que NOT dir=108 da también NULL