

## T016\_Tareas

<b>Ejercicio 1</b>	<b>2</b>
- 1.1 Obtener los nombres de los artículos de la tienda	2
- 1.2 Obtener los nombres y los precios de los productos de la tienda	2
- 1.3 Obtener el nombre de los productos cuyo precio sea menor o igual a 200€	2
- 1.4 Obtener todos los datos de los artículos cuyo precio esté entre los 60€ y los 120€	3
- 1.5 Obtener el nombre y el precio en pesetas (es decir, el precio en euros multiplicado por 166'386)	3
- 1.6 Seleccionar el precio medio de todos los productos	3
- 1.7 Obtener el precio medio de los artículos cuyo código de fabricante sea 2	3
- 1.8 Obtener el número de artículos cuyo precio sea mayor o igual a 180€	4
- 1.9 Obtener el nombre y precio de los artículos cuyo precio sea mayor o igual a 180€ y ordenarlos descendientemente por precio, y luego ascendientemente por nombre	4
- 1.10 Obtener un listado completo de artículos, incluyendo por cada artículo los datos del artículo y su fabricante	4
- 1.11 Obtener un listado de artículos, incluyendo el nombre del artículo, su precio y el nombre de su fabricante	5
- 1.12 Obtener el precio medio de los productos de cada fabricante, mostrando solo los códigos del fabricante	5
- 1.13 Obtener el precio medio de los productos de cada fabricante, mostrando el nombre del fabricante	6
- 1.14 Obtener los nombres de los fabricantes que ofrezcan productos cuyo precio medio sea mayor o igual a 150€	6
- 1.15 Obtener el nombre y precio del artículo más barato	6
- 1.16 Obtener una lista con el nombre y precio de los artículos más caros de cada proveedor (incluyendo el nombre del proveedor)	7
- 1.17 Añadir un nuevo producto	7
- 1.18 Cambiar el nombre del producto 8 a 'Impresora Laser'	7
- 1.19 Aplicar un descuento del 10% (multiplicar el precio por 0'9) a todos los productos	7
- 1.20 Aplicar un descuento de 10€ a todos los productos cuyo precio sea mayor o igual a 120€	7
<b>Ejercicio 2</b>	<b>8</b>
- 2.1 Obtener los apellidos de los empleados	8
- 2.2 Obtener los apellidos de los empleados sin repeticiones	8
- 2.3 Obtener todos los datos de los empleados que se apellidan 'López'	8
- 2.4 Obtener todos los datos de los empleados que se apellidan 'López' y los que se apellidan 'Pérez'	8
- 2.5 Obtener todos los datos de los empleados que trabajan para el departamento 9	9
- 2.6 Obtener todos los datos de los empleados que trabajan para el departamento 3 y para el departamento 7	9
- 2.7 Obtener todos los datos de los empleados cuyo apellido comience por 'P'	9
- 2.8 Obtener el presupuesto total de todos los departamentos	9
- 2.9 Obtener un listado completo de empleados, incluyendo por cada empleado los datos del empleado y de su departamento	10
- 2.10 Obtener un listado completo de empleados, incluyendo por cada empleado los datos del empleado y de su departamento	10

- 2.11 Obtener un listado completo de empleados, incluyendo el nombre y apellidos del empleado junto al nombre y presupuesto de su departamento.....	10
- 2.12 Obtener los nombres y apellidos de los empleados que trabajen en departamentos cuyo presupuesto sea mayor de 60.000 €.....	11
- 2.13 Obtener los datos de los departamentos cuyo presupuesto es superior al presupuesto medio de todos los departamentos.....	11
- 2.14 Obtener los nombres (únicamente los nombres) de los departamentos que tienen más de dos empleados.....	11
- 2.15 Añadir un nuevo departamento: 'Calidad', con presupuesto de 40.000 € y código11. Añadir un empleado vinculado al departamento recién creado: Esther Vázquez, DNI: 89267109.....	11
- 2.16 Aplicar un recorte presupuestario del 10 % a todos los departamentos.....	12
- 2.17 Reasignar a los empleados del departamento de investigación ( código 77) al departamento de informática ( código 14).....	12
- 2.18 Despedir a todos los empleados que trabajan para el departamento de informática ( código 9).....	12
- 2.19 Despedir a todos los empleados que trabajen para departamentos cuyo presupuesto sea superior a los 60.000 €.....	12
- 2.20 Despedir a todos los empleados.....	12
<b>Ejercicio 3.....</b>	<b>13</b>
- 3.1 Obtener todos los almacenes.....	13
- 3.2 Obtener todas las cajas cuyo contenido tenga un valor superior a 150 €.....	13
- 3.3 Obtener los tipos de contenidos de las cajas.....	13
- 3.4 Obtener el valor medio de todas las cajas.....	13
- 3.5 Obtener el valor medio de las cajas de cada almacén.....	14
- 3.6 Obtener los códigos de los almacenes en los cuales el valor medio de las cajas sea superior a 150 €.....	14
- 3.7 Obtener el número de referencia de cada caja junto con el nombre de la ciudad en el que se encuentra.....	14
- 3.8 Obtener el número de cajas que hay en cada almacén.....	14
- 3.9 Obtener los códigos de los almacenes que están saturados (los almacenes donde el número de cajas es superior a la capacidad).....	15
- 3.10 Obtener los numeras de referencia de las cajas que están en Bilbao.....	15
- 3.11 Insertar un nuevo almacén en Barcelona con capacidad para 3 cajas.....	15
- 3.12 Insertar una nueva caja, con número de referencia 'H5RT', con contenido 'Papel' , valor 200, y situada en el almacén F. Rebajar el valor de todas las cajas un 15 %.....	15
- 3.13 Rebajar el valor de todas las cajas un 15 %.....	15
- 3.14 Rebajar un 20 % el valor de todas las cajas cuyo valor sea superior al valor medio de todas las cajas.....	15
- 3.15 Eliminar todas las cajas cuyo valor sea inferior a 100 €.....	15
- 3.16 Vaciar el contenido de los almacenes que están saturados.....	15
<b>Ejercicio 4.....</b>	<b>16</b>
- 4.1 Mostrar el nombre de todas las películas.....	16
- 4.2 Mostrar las distintas calificaciones de edad que existen.....	16
- 4.3 Mostrar todas las películas que no han sido calificadas.....	16
- 4.4 Mostrar todas las salas que no proyectan ninguna película.....	16
- 4.5 Mostrar la información de todas las salas y, si se proyecta alguna película en la	

sala, mostrar también la información de la película.....	16
- 4.6 Mostrar la información de todas las películas y, si se proyecta en alguna sala, mostrar también la información de la sala.....	17
- 4.7 Mostrar los nombres de las películas que no se proyectan en ninguna sala.....	17
- 4.8 Añadir una nueva película 'Uno, Dos, Tres' , para mayores de 7 años.....	17
- 4.9 Hacer constar que todas las películas no calificadas han sido calificadas 'no recomendables para menores de 13 años'.....	17
- 4.10 Eliminar todas las salas que proyectan películas recomendadas para todos los públicos.....	17

**Ejercicio 1**

- 1.1 Obtener los nombres de los artículos de la tienda

```
MariaDB [tienda_informatica]> SELECT Nombre FROM articulos;
+-----+
| Nombre |
+-----+
| Artículo 1 |
| Artículo 2 |
| Artículo 3 |
| Artículo 4 |
| Artículo 5 |
| Artículo 6 |
| Artículo 7 |
| Artículo 8 |
| Artículo 9 |
| Artículo 10 |
+-----+
10 rows in set (0.007 sec)
```

- 1.2 Obtener los nombres y los precios de los productos de la tienda

```
MariaDB [tienda_informatica]> SELECT Nombre, Precio FROM articulos;
+-----+-----+
| Nombre | Precio |
+-----+-----+
| Artículo 1 | 100 |
| Artículo 2 | 200 |
| Artículo 3 | 150 |
| Artículo 4 | 300 |
| Artículo 5 | 120 |
| Artículo 6 | 180 |
| Artículo 7 | 250 |
| Artículo 8 | 350 |
| Artículo 9 | 220 |
| Artículo 10 | 280 |
+-----+-----+
10 rows in set (0.000 sec)

MariaDB [tienda_informatica]>
```

- 1.3 Obtener el nombre de los productos cuyo precio sea menor o igual a 200€

```
MariaDB [tienda_informatica]> SELECT Nombre FROM articulos WHERE Precio <= 200;
+-----+
| Nombre |
+-----+
| Artículo 1 |
| Artículo 2 |
| Artículo 3 |
| Artículo 5 |
| Artículo 6 |
+-----+
5 rows in set (0.007 sec)

MariaDB [tienda_informatica]>
```

- 1.4 Obtener todos los datos de los artículos cuyo precio esté entre los 60€ y los 120€

```
MariaDB [tienda_informatica]> SELECT * FROM articulos
-> WHERE Precio BETWEEN 60 AND 120;
+-----+-----+-----+-----+
| Codigo | Nombre   | Precio | Fabricante |
+-----+-----+-----+-----+
|      1 | Artículo 1 |    100 |          1 |
|      5 | Artículo 5 |    120 |          5 |
+-----+-----+-----+-----+
2 rows in set (0.007 sec)

MariaDB [tienda_informatica]> _
```

- 1.5 Obtener el nombre y el precio en pesetas (es decir, el precio en euros multiplicado por 166'386)

```
MariaDB [tienda_informatica]> SELECT Nombre, Precio * 166386 AS Precio_pesetas FROM articulos;
+-----+-----+
| Nombre   | Precio_pesetas |
+-----+-----+
| Artículo 1 |    16638600 |
| Artículo 2 |    33277200 |
| Artículo 3 |    24957900 |
| Artículo 4 |    49915800 |
| Artículo 5 |    19966320 |
| Artículo 6 |    29949480 |
| Artículo 7 |    41596500 |
| Artículo 8 |    58235100 |
| Artículo 9 |    36604920 |
| Artículo 10 |    46588080 |
+-----+-----+
10 rows in set (0.001 sec)
```

- 1.6 Seleccionar el precio medio de todos los productos

```
MariaDB [tienda_informatica]> SELECT AVG(Precio) AS Precio_medio FROM articulos;
+-----+
| Precio_medio |
+-----+
|    215.0000 |
+-----+
1 row in set (0.001 sec)

MariaDB [tienda_informatica]> _
```

- 1.7 Obtener el precio medio de los artículos cuyo código de fabricante sea 2

```
MariaDB [tienda_informatica]> SELECT AVG(Precio) AS Precio_medio FROM articulos
-> WHERE Fabricante = 2;
+-----+
| Precio_medio |
+-----+
|    200.0000 |
+-----+
1 row in set (0.008 sec)
```

- 1.8 Obtener el número de artículos cuyo precio sea mayor o igual a 180€

```
MariaDB [tienda_informatica]> SELECT COUNT(*) AS Numero_de_articulos FROM articulos
-> WHERE Precio >= 180;
+-----+
| Numero_de_articulos |
+-----+
|          7          |
+-----+
1 row in set (0.001 sec)
```

- 1.9 Obtener el nombre y precio de los artículos cuyo precio sea mayor o igual a 180€ y ordenarlos descendientemente por precio, y luego ascendientemente por nombre

```
MariaDB [tienda_informatica]> SELECT Nombre, Precio FROM articulos
-> WHERE Precio >= 180
-> ORDER BY Precio DESC, Nombre ASC;
+-----+-----+
| Nombre      | Precio |
+-----+-----+
| Artículo 8  | 350    |
| Artículo 4  | 300    |
| Artículo 10 | 280    |
| Artículo 7  | 250    |
| Artículo 9  | 220    |
| Artículo 2  | 200    |
| Artículo 6  | 180    |
+-----+-----+
7 rows in set (0.001 sec)
```

- 1.10 Obtener un listado completo de artículos, incluyendo por cada artículo los datos del artículo y su fabricante

```
MariaDB [tienda_informatica]> SELECT a.Codigo AS Codigo_articulo,
->      a.Nombre AS Nombre_articulo,
->      a.Precio,
->      f.Codigo AS Codigo_fabricante,
->      f.Nombre AS Nombre_fabricante
-> FROM articulos a
-> JOIN fabricantes f ON a.Fabricante = f.Codigo;
+-----+-----+-----+-----+-----+
| Codigo_articulo | Nombre_articulo | Precio | Codigo_fabricante | Nombre_fabricante |
+-----+-----+-----+-----+-----+
| 1 | Artículo 1 | 100 | 1 | Fabricante A |
| 2 | Artículo 2 | 200 | 2 | Fabricante B |
| 3 | Artículo 3 | 150 | 3 | Fabricante C |
| 4 | Artículo 4 | 300 | 4 | Fabricante D |
| 5 | Artículo 5 | 120 | 5 | Fabricante E |
| 6 | Artículo 6 | 180 | 6 | Fabricante F |
| 7 | Artículo 7 | 250 | 7 | Fabricante G |
| 8 | Artículo 8 | 350 | 8 | Fabricante H |
| 9 | Artículo 9 | 220 | 9 | Fabricante I |
| 10 | Artículo 10 | 280 | 10 | Fabricante J |
+-----+-----+-----+-----+-----+
10 rows in set (0.007 sec)
```

- 1.11 Obtener un listado de artículos, incluyendo el nombre del artículo, su precio y el nombre de su fabricante

```
MariaDB [tienda_informatica]> SELECT a.Nombre AS Nombre_articulo,
->      a.Precio,
->      f.Nombre AS Nombre_fabricante
-> FROM articulos a
-> JOIN fabricantes f ON a.Fabricante = f.Codigo;
```

Nombre_articulo	Precio	Nombre_fabricante
Articulo 1	100	Fabricante A
Articulo 2	200	Fabricante B
Articulo 3	150	Fabricante C
Articulo 4	300	Fabricante D
Articulo 5	120	Fabricante E
Articulo 6	180	Fabricante F
Articulo 7	250	Fabricante G
Articulo 8	350	Fabricante H
Articulo 9	220	Fabricante I
Articulo 10	280	Fabricante J

10 rows in set (0.001 sec)

- 1.12 Obtener el precio medio de los productos de cada fabricante, mostrando solo los códigos del fabricante

```
MariaDB [tienda_informatica]> SELECT Fabricante, AVG(Precio) AS Precio_medio
-> FROM articulos
-> GROUP BY Fabricante;
```

Fabricante	Precio_medio
1	100.0000
2	200.0000
3	150.0000
4	300.0000
5	120.0000
6	180.0000
7	250.0000
8	350.0000
9	220.0000
10	280.0000

10 rows in set (0.001 sec)

- 1.13 Obtener el precio medio de los productos de cada fabricante, mostrando el nombre del fabricante

```
MariaDB [tienda_informatica]> SELECT f.Nombre AS Nombre_fabricante, AVG(a.Precio) AS Precio_medio
-> FROM articulos a
-> JOIN fabricantes f ON a.Fabricante = f.Codigo
-> GROUP BY f.Nombre;
```

Nombre_fabricante	Precio_medio
Fabricante A	100.0000
Fabricante B	200.0000
Fabricante C	150.0000
Fabricante D	300.0000
Fabricante E	120.0000
Fabricante F	180.0000
Fabricante G	250.0000
Fabricante H	350.0000
Fabricante I	220.0000
Fabricante J	280.0000

```
10 rows in set (0.007 sec)
```

- 1.14 Obtener los nombres de los fabricantes que ofrezcan productos cuyo precio medio sea mayor o igual a 150€

```
MariaDB [tienda_informatica]> SELECT f.Nombre
-> FROM fabricantes f
-> JOIN articulos a ON f.Codigo = a.Fabricante
-> GROUP BY f.Nombre
-> HAVING AVG(a.Precio) >= 150;
```

Nombre
Fabricante B
Fabricante C
Fabricante D
Fabricante F
Fabricante G
Fabricante H
Fabricante I
Fabricante J

```
8 rows in set (0.007 sec)
```

- 1.15 Obtener el nombre y precio del artículo más barato

```
MariaDB [tienda_informatica]> SELECT Nombre, Precio
-> FROM articulos
-> WHERE Precio = (SELECT MIN(Precio) FROM articulos);
```

Nombre	Precio
Articulo 1	100

```
1 row in set (0.008 sec)
```



- 1.16 Obtener una lista con el nombre y precio de los artículos más caros de cada proveedor (incluyendo el nombre del proveedor)

```
MariaDB [tienda_informatica]> SELECT f.Nombre AS Nombre_proveedor, a.Nombre AS
Nombre_articulo, a.Precio AS Precio_articulo
-> FROM fabricantes f
-> JOIN articulos a ON f.Codigo = a.Fabricante
-> WHERE (a.Precio, a.Fabricante) IN (
->   SELECT MAX(Precio), Fabricante
->   FROM articulos
->   GROUP BY Fabricante
-> );
```

Nombre_proveedor	Nombre_articulo	Precio_articulo
Fabricante A	Articulo 1	100
Fabricante B	Articulo 2	200
Fabricante C	Articulo 3	150
Fabricante D	Articulo 4	300
Fabricante E	Articulo 5	120
Fabricante F	Articulo 6	180
Fabricante G	Articulo 7	250
Fabricante H	Articulo 8	350
Fabricante I	Articulo 9	220
Fabricante J	Articulo 10	280

10 rows in set (0.002 sec)

- 1.17 Añadir un nuevo producto

```
MariaDB [tienda_informatica]> INSERT INTO articulos (Codigo, Nombre, Precio, Fabricante)
-> VALUES (11, 'Nuevo producto', 150, 3);
Query OK, 1 row affected (0.009 sec)
```

- 1.18 Cambiar el nombre del producto 8 a 'Impresora Laser'

```
MariaDB [tienda_informatica]> UPDATE articulos
-> SET Nombre = 'Impresora Laser'
-> WHERE Codigo = 8;
Query OK, 1 row affected (0.010 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

- 1.19 Aplicar un descuento del 10% (multiplicar el precio por 0.9) a todos los productos

```
MariaDB [tienda_informatica]> UPDATE articulos
-> SET Precio = Precio * 0.9;
Query OK, 11 rows affected (0.008 sec)
Rows matched: 11 Changed: 11 Warnings: 0
```

- 1.20 Aplicar un descuento de 10€ a todos los productos cuyo precio sea mayor o igual a 120€

```
MariaDB [tienda_informatica]> UPDATE articulos
-> SET Precio = Precio - 10
-> WHERE Precio >= 120;
Query OK, 9 rows affected (0.008 sec)
Rows matched: 9 Changed: 9 Warnings: 0
```

**Ejercicio 2**

- 2.1 Obtener los apellidos de los empleados

```
MariaDB [empleados]> SELECT Apellidos FROM empleados;
+-----+
| Apellidos |
+-----+
| Ruiz      |
| González  |
| Martínez  |
| Rodríguez |
| Pérez     |
| López     |
| Sánchez   |
| Gómez     |
| Díaz      |
| Fernández |
+-----+
10 rows in set (0.001 sec)
```

- 2.2 Obtener los apellidos de los empleados sin repeticiones

```
MariaDB [empleados]> SELECT DISTINCT Apellidos FROM empleados;
+-----+
| Apellidos |
+-----+
| Ruiz      |
| González  |
| Martínez  |
| Rodríguez |
| Pérez     |
| López     |
| Sánchez   |
| Gómez     |
| Díaz      |
| Fernández |
+-----+
10 rows in set (0.000 sec)
```

- 2.3 Obtener todos los datos de los empleados que se apellidan 'López'

```
MariaDB [empleados]> SELECT * FROM empleados
-> WHERE Apellidos = 'López';
+-----+-----+-----+-----+
| DNI      | Nombre | Apellidos | Departamento |
+-----+-----+-----+-----+
| 55555555E | José   | López     | 5             |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

- 2.4 Obtener todos los datos de los empleados que se apellidan 'López' y los que se apellidan 'Pérez'.

```
MariaDB [empleados]> SELECT * FROM empleados
-> WHERE Apellidos = 'López' OR Apellidos = 'Pérez';
+-----+-----+-----+-----+
| DNI      | Nombre | Apellidos | Departamento |
+-----+-----+-----+-----+
| 44444444D | Carmen | Pérez     | 4             |
| 55555555E | José   | López     | 5             |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

- 2.5 Obtener todos los datos de los empleados que trabajan para el departamento 9

```
MariaDB [empleados]> SELECT * FROM empleados
-> WHERE Departamento = 9;
+-----+-----+-----+-----+
| DNI      | Nombre | Apellidos | Departamento |
+-----+-----+-----+-----+
| 99999999I | Miguel | Fernández | 9 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

- 2.6 Obtener todos los datos de los empleados que trabajan para el departamento 3 y para el departamento 7

```
MariaDB [empleados]> SELECT * FROM empleados
-> WHERE Departamento IN (3, 7);
+-----+-----+-----+-----+
| DNI      | Nombre | Apellidos | Departamento |
+-----+-----+-----+-----+
| 33333333C | Antonio | Rodríguez | 3 |
| 77777777G | Manuel | Gómez      | 7 |
+-----+-----+-----+-----+
2 rows in set (0.007 sec)
```

- 2.7 Obtener todos los datos de los empleados cuyo apellido comience por 'P'

```
MariaDB [empleados]> SELECT * FROM empleados
-> WHERE Apellidos LIKE 'P%';
+-----+-----+-----+-----+
| DNI      | Nombre | Apellidos | Departamento |
+-----+-----+-----+-----+
| 44444444D | Carmen | Pérez      | 4 |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

- 2.8 Obtener el presupuesto total de todos los departamentos

```
MariaDB [empleados]> SELECT SUM(Presupuesto) AS Presupuesto_total
-> FROM departamentos;
+-----+
| Presupuesto_total |
+-----+
| 167000 |
+-----+
1 row in set (0.000 sec)
```

- 2.9 Obtener un listado completo de empleados, incluyendo por cada empleado los datos del empleado y de su departamento

```
MariaDB [empleados]> SELECT e.*, d.Nombre AS Nombre_departamento, d.Presupuesto
-> FROM empleados e
-> JOIN departamentos d ON e.Departamento = d.Codigo;
```

DNI	Nombre	Apellidos	Departamento	Nombre_departamento	Presupuesto
10101010J	Elena	Ruiz	10	Administración	21000
11111111A	Juan	González	1	Ventas	10000
22222222B	María	Martínez	2	Recursos Humanos	15000
33333333C	Antonio	Rodríguez	3	Marketing	20000
44444444D	Carmen	Pérez	4	Desarrollo	12000
55555555E	José	López	5	Finanzas	18000
66666666F	Ana	Sánchez	6	Producción	22000
77777777G	Manuel	Gómez	7	Logística	13000
88888888H	Laura	Díaz	8	Calidad	19000
99999999I	Miguel	Fernández	9	Tecnología	17000

10 rows in set (0.001 sec)

- 2.10 Obtener un listado completo de empleados, incluyendo por cada empleado los datos del empleado y de su departamento

```
MariaDB [empleados]> SELECT empleados.*, departamentos.Nombre AS Nombre_departamento, departamentos.Presupuesto
-> FROM empleados
-> JOIN departamentos ON empleados.Departamento = departamentos.Codigo;
```

DNI	Nombre	Apellidos	Departamento	Nombre_departamento	Presupuesto
10101010J	Elena	Ruiz	10	Administración	21000
11111111A	Juan	González	1	Ventas	10000
22222222B	María	Martínez	2	Recursos Humanos	15000
33333333C	Antonio	Rodríguez	3	Marketing	20000
44444444D	Carmen	Pérez	4	Desarrollo	12000
55555555E	José	López	5	Finanzas	18000
66666666F	Ana	Sánchez	6	Producción	22000
77777777G	Manuel	Gómez	7	Logística	13000
88888888H	Laura	Díaz	8	Calidad	19000
99999999I	Miguel	Fernández	9	Tecnología	17000

10 rows in set (0.001 sec)

- 2.11 Obtener un listado completo de empleados, incluyendo el nombre y apellidos del empleado junto al nombre y presupuesto de su departamento

```
MariaDB [empleados]> SELECT e.Nombre, e.Apellidos, d.Nombre AS Nombre_departamento, d.Presupuesto
-> FROM empleados e
-> JOIN departamentos d ON e.Departamento = d.Codigo;
```

Nombre	Apellidos	Nombre_departamento	Presupuesto
Elena	Ruiz	Administración	21000
Juan	González	Ventas	10000
María	Martínez	Recursos Humanos	15000
Antonio	Rodríguez	Marketing	20000
Carmen	Pérez	Desarrollo	12000
José	López	Finanzas	18000
Ana	Sánchez	Producción	22000
Manuel	Gómez	Logística	13000
Laura	Díaz	Calidad	19000
Miguel	Fernández	Tecnología	17000

10 rows in set (0.001 sec)

- 2.12 Obtener los nombres y apellidos de los empleados que trabajen en departamentos cuyo presupuesto sea mayor de 60.000 €

```
MariaDB [empleados]> SELECT e.Nombre, e.Apellidos
-> FROM empleados e
-> JOIN departamentos d ON e.Departamento = d.Codigo
-> WHERE d.Presupuesto > 60000;
Empty set (0.007 sec)
```

- 2.13 Obtener los datos de los departamentos cuyo presupuesto es superior al presupuesto medio de todos los departamentos

```
MariaDB [empleados]> SELECT *
-> FROM departamentos
-> WHERE Presupuesto > (
->     SELECT AVG(Presupuesto)
->     FROM departamentos
-> );
```

Codigo	Nombre	Presupuesto
3	Marketing	20000
5	Finanzas	18000
6	Producción	22000
8	Calidad	19000
9	Tecnología	17000
10	Administración	21000

```
6 rows in set (0.001 sec)
```

- 2.14 Obtener los nombres (únicamente los nombres) de los departamentos que tienen más de dos empleados

```
MariaDB [empleados]> SELECT d.Nombre
-> FROM departamentos d
-> JOIN empleados e ON d.Codigo = e.Departamento
-> GROUP BY d.Codigo
-> HAVING COUNT(*) > 2;
Empty set (0.000 sec)
```

- 2.15 Añadir un nuevo departamento: 'Calidad', con presupuesto de 40.000 € y código 11.  
Añadir un empleado vinculado al departamento recién creado: Esther Vázquez, DNI: 89267109

```
MariaDB [empleados]> -- Añadir el nuevo departamento 'Calidad'
MariaDB [empleados]> INSERT INTO departamentos (Codigo, Nombre, Presupuesto)
-> VALUES (11, 'Calidad', 40000);
Query OK, 1 row affected (0.003 sec)

MariaDB [empleados]>
MariaDB [empleados]> -- Añadir el nuevo empleado Esther Vázquez vinculado al departamento 'Calidad'
MariaDB [empleados]> INSERT INTO empleados (DNI, Nombre, Apellidos, Departamento)
-> VALUES ('89267109', 'Esther', 'Vázquez', 11);
Query OK, 1 row affected (0.002 sec)
```

- 2.16 Aplicar un recorte presupuestario del 10 % a todos los departamentos

```
MariaDB [empleados]> UPDATE departamentos
-> SET Presupuesto = Presupuesto * 0.9;
Query OK, 11 rows affected (0.009 sec)
Rows matched: 11 Changed: 11 Warnings: 0
```

- 2.17 Reasignar a los empleados del departamento de investigación ( código 77) al departamento de informática ( código 14)

```
MariaDB [empleados]> UPDATE empleados
-> SET Departamento = 9
-> WHERE Departamento = 4;
Query OK, 1 row affected (0.007 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

- 2.18 Despedir a todos los empleados que trabajan para el departamento de informática ( código 9)

```
MariaDB [empleados]> DELETE FROM empleados
-> WHERE Departamento = 9;
Query OK, 2 rows affected (0.009 sec)
```

- 2.19 Despedir a todos los empleados que trabajen para departamentos cuyo presupuesto sea superior a los 60.000 €

```
MariaDB [empleados]> DELETE FROM empleados
-> WHERE Departamento IN (
-> SELECT Codigo
-> FROM departamentos
-> WHERE Presupuesto > 60000
-> );
Query OK, 0 rows affected (0.001 sec)
```

- 2.20 Despedir a todos los empleados

```
MariaDB [empleados]> DELETE FROM empleados;
Query OK, 9 rows affected (0.007 sec)
```

**Ejercicio 3**

- 3.1 Obtener todos los almacenes

```
MariaDB [almacenes]> SELECT * FROM almacenes;
```

Codigo	Lugar	Capacidad
1	Almacén A	100
2	Almacén B	150
3	Almacén C	200
4	Almacén D	120
5	Almacén E	180
6	Almacén F	220
7	Almacén G	130
8	Almacén H	190
9	Almacén I	170
10	Almacén J	210

```
10 rows in set (0.001 sec)
```

- 3.2 Obtener todas las cajas cuyo contenido tenga un valor superior a 150 €

```
MariaDB [almacenes]> SELECT *
-> FROM cajas
-> WHERE Valor > 150;
```

NumReferencia	Contenido	Valor	Almacen
C9101	Electrodomésticos	200	3
G2122	Muebles	300	7

```
2 rows in set (0.001 sec)
```

- 3.3 Obtener los tipos de contenidos de las cajas

```
MariaDB [almacenes]> SELECT DISTINCT Contenido
-> FROM cajas;
```

Contenido
Libros
Ropa
Electrodomésticos
Juguetes
Herramientas
Electrónica
Muebles
Comida
Artículos deportivos
Productos de limpieza

```
10 rows in set (0.001 sec)
```

- 3.4 Obtener el valor medio de todas las cajas

```
MariaDB [almacenes]> SELECT AVG(Valor) AS Valor_medio
-> FROM cajas;
```

Valor_medio
122.0000

```
1 row in set (0.000 sec)
```

- 3.5 Obtener el valor medio de las cajas de cada almacén

```
MariaDB [almacenes]> SELECT Almacen, AVG(Valor) AS Valor_medio
-> FROM cajas
-> GROUP BY Almacen;
+-----+-----+
| Almacen | Valor_medio |
+-----+-----+
| 1 | 50.0000 |
| 2 | 100.0000 |
| 3 | 200.0000 |
| 4 | 80.0000 |
| 5 | 120.0000 |
| 6 | 150.0000 |
| 7 | 300.0000 |
| 8 | 70.0000 |
| 9 | 90.0000 |
| 10 | 60.0000 |
+-----+-----+
10 rows in set (0.001 sec)
```

- 3.6 Obtener los códigos de los almacenes en los cuales el valor medio de las cajas sea superior a 150 €

```
MariaDB [almacenes]> SELECT Almacen
-> FROM cajas
-> GROUP BY Almacen
-> HAVING AVG(Valor) > 150;
+-----+
| Almacen |
+-----+
| 3 |
| 7 |
+-----+
2 rows in set (0.001 sec)
```

- 3.7 Obtener el número de referencia de cada caja junto con el nombre de la ciudad en el que se encuentra

```
MariaDB [almacenes]> SELECT c.NumReferencia, a.Lugar AS Ciudad
-> FROM cajas c
-> JOIN almacenes a ON c.Almacen = a.Codigo;
+-----+-----+
| NumReferencia | Ciudad |
+-----+-----+
| A1234 | Almacén A |
| B5678 | Almacén B |
| C9101 | Almacén C |
| D1213 | Almacén D |
| E1516 | Almacén E |
| F1819 | Almacén F |
| G2122 | Almacén G |
| H2425 | Almacén H |
| I2728 | Almacén I |
| J3031 | Almacén J |
+-----+-----+
10 rows in set (0.001 sec)
```

- 3.8 Obtener el número de cajas que hay en cada almacén

```
MariaDB [almacenes]> SELECT Almacen, COUNT(*) AS Num_cajas
-> FROM cajas
-> GROUP BY Almacen;
+-----+-----+
| Almacen | Num_cajas |
+-----+-----+
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 1 |
| 7 | 1 |
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |
+-----+-----+
10 rows in set (0.000 sec)
```



- 3.9 Obtener los códigos de los almacenes que están saturados (los almacenes donde el número de cajas es superior a la capacidad)

```
MariaDB [almacenes]> SELECT Almacen
-> FROM (
->   SELECT c.Almacen, COUNT(*) AS Num_cajas, a.Capacidad
->   FROM cajas c
->   JOIN almacenes a ON c.Almacen = a.Codigo
->   GROUP BY c.Almacen
-> ) AS subquery
-> WHERE Num_cajas > Capacidad;
Empty set (0.001 sec)
```

- 3.10 Obtener los numeros de referencia de las cajas que están en Bilbao

```
MariaDB [almacenes]> SELECT c.NumReferencia
-> FROM cajas c
-> JOIN almacenes a ON c.Almacen = a.Codigo
-> WHERE a.Lugar = 'Bilbao';
Empty set (0.000 sec)
```

- 3.11 Insertar un nuevo almacén en Barcelona con capacidad para 3 cajas

```
MariaDB [almacenes]> INSERT INTO almacenes (Codigo, Lugar, Capacidad)
-> VALUES (11, 'Barcelona', 3);
Query OK, 1 row affected (0.007 sec)
```

- 3.12 Insertar una nueva caja, con número de referencia 'H5RT', con contenido 'Papel' , valor 200, y situada en el almacén F. Rebajar el valor de todas las cajas un 15 %

```
MariaDB [almacenes]> INSERT INTO cajas (NumReferencia, Contenido, Valor, Almacen)
-> VALUES ('H5RT', 'Papel', 200, 6);
Query OK, 1 row affected (0.053 sec)

MariaDB [almacenes]> UPDATE cajas
-> SET Valor = Valor * 0.85;
Query OK, 11 rows affected (0.014 sec)
Rows matched: 11  Changed: 11  Warnings: 0
```

- 3.13 Rebajar el valor de todas las cajas un 15 %

```
MariaDB [almacenes]> UPDATE cajas
-> SET Valor = Valor * 0.85;
Query OK, 11 rows affected (0.008 sec)
Rows matched: 11  Changed: 11  Warnings: 0
```

- 3.14 Rebajar un 20 % el valor de todas las cajas cuyo valor sea superior al valor medio de todas las cajas

```
MariaDB [almacenes]> UPDATE cajas
-> SET Valor = Valor * 0.8
-> WHERE Valor > (
->   SELECT AVG(Valor)
->   FROM cajas
-> );
Query OK, 4 rows affected (0.014 sec)
Rows matched: 4  Changed: 4  Warnings: 0
```

- 3.15 Eliminar todas las cajas cuyo valor sea inferior a 100 €

```
MariaDB [almacenes]> DELETE FROM cajas
-> WHERE Valor < 100;
Query OK, 8 rows affected (0.004 sec)
```

- 3.16 Vaciar el contenido de los almacenes que están saturados

```
MariaDB [almacenes]> DELETE FROM cajas
-> WHERE Almacen IN (
->   SELECT Codigo
->   FROM almacenes
->   WHERE (SELECT COUNT(*) FROM cajas WHERE cajas.Almacen = almacenes.Codigo) > Capacidad
-> );
Query OK, 0 rows affected (0.009 sec)
```

**Ejercicio 4**

- 4.1 Mostrar el nombre de todas las películas

```
MariaDB [peliculasysalas]> SELECT Nombre
-> FROM peliculas;
+-----+
| Nombre |
+-----+
| Titanic |
| The Shawshank Redemption |
| The Godfather |
| Forrest Gump |
| Pulp Fiction |
| The Matrix |
| Inception |
| The Silence of the Lambs |
| The Lion King |
| The Dark Knight |
+-----+
10 rows in set (0.000 sec)
```

- 4.2 Mostrar las distintas calificaciones de edad que existen

```
MariaDB [peliculasysalas]> SELECT DISTINCT CalificacionEdad
-> FROM peliculas;
+-----+
| CalificacionEdad |
+-----+
| 13 |
| 16 |
| 18 |
| 12 |
| 0 |
+-----+
5 rows in set (0.000 sec)
```

- 4.3 Mostrar todas las películas que no han sido calificadas

```
MariaDB [peliculasysalas]> SELECT Nombre
-> FROM peliculas
-> WHERE CalificacionEdad IS NULL;
Empty set (0.002 sec)
```

- 4.4 Mostrar todas las salas que no proyectan ninguna película

```
MariaDB [peliculasysalas]> SELECT *
-> FROM salas
-> WHERE Pelicula IS NULL;
Empty set (0.007 sec)
```

- 4.5 Mostrar la información de todas las salas y, si se proyecta alguna película en la sala, mostrar también la información de la película

```
MariaDB [peliculasysalas]> SELECT s.Codigo AS Codigo_sala, s.Nombre AS Nombre_sala, p.Codigo AS Codigo_pelicula, p.Nombre AS Nombre_pelicula
-> FROM salas s
-> LEFT JOIN peliculas p ON s.Pelicula = p.Codigo;
+-----+-----+-----+-----+
| Codigo_sala | Nombre_sala | Codigo_pelicula | Nombre_pelicula |
+-----+-----+-----+-----+
| 1 | Sala 1 | 1 | Titanic |
| 2 | Sala 2 | 2 | The Shawshank Redemption |
| 3 | Sala 3 | 3 | The Godfather |
| 4 | Sala 4 | 4 | Forrest Gump |
| 5 | Sala 5 | 5 | Pulp Fiction |
| 6 | Sala 6 | 6 | The Matrix |
| 7 | Sala 7 | 7 | Inception |
| 8 | Sala 8 | 8 | The Silence of the Lambs |
| 9 | Sala 9 | 9 | The Lion King |
| 10 | Sala 10 | 10 | The Dark Knight |
+-----+-----+-----+-----+
10 rows in set (0.001 sec)
```

- 4.6 Mostrar la información de todas las películas y, si se proyecta en alguna sala, mostrar también la información de la sala

```
MariaDB [peliculasysalas]> SELECT p.Codigo AS Codigo_pelicula, p.Nombre AS Nombre_pelicula, p.CalificacionEdad, s.Codigo
AS Codigo_sala, s.Nombre AS Nombre_sala
-> FROM peliculas p
-> LEFT JOIN salas s ON p.Codigo = s.Pelicula;
```

Codigo_pelicula	Nombre_pelicula	CalificacionEdad	Codigo_sala	Nombre_sala
1	Titanic	13	1	Sala 1
2	The Shawshank Redemption	16	2	Sala 2
3	The Godfather	18	3	Sala 3
4	Forrest Gump	12	4	Sala 4
5	Pulp Fiction	18	5	Sala 5
6	The Matrix	16	6	Sala 6
7	Inception	13	7	Sala 7
8	The Silence of the Lambs	18	8	Sala 8
9	The Lion King	0	9	Sala 9
10	The Dark Knight	16	10	Sala 10

10 rows in set (0.001 sec)

- 4.7 Mostrar los nombres de las películas que no se proyectan en ninguna sala

```
MariaDB [peliculasysalas]> SELECT Nombre
-> FROM peliculas
-> WHERE Codigo NOT IN (SELECT Pelicula FROM salas WHERE Pelicula IS NOT NULL);
Empty set (0.007 sec)
```

- 4.8 Añadir una nueva película 'Uno, Dos, Tres', para mayores de 7 años

```
MariaDB [peliculasysalas]> INSERT INTO peliculas (Nombre, CalificacionEdad)
-> VALUES ('Uno, Dos, Tres', 7);
Query OK, 1 row affected (0.007 sec)
```

- 4.9 Hacer constar que todas las películas no calificadas han sido calificadas 'no recomendables para menores de 13 años'

```
MariaDB [peliculasysalas]> UPDATE peliculas
-> SET CalificacionEdad = 13
-> WHERE CalificacionEdad IS NULL;
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0 Changed: 0 Warnings: 0
```

- 4.10 Eliminar todas las salas que proyectan películas recomendadas para todos los públicos

```
MariaDB [peliculasysalas]> DELETE FROM salas
-> WHERE Pelicula IN (SELECT Codigo FROM peliculas WHERE CalificacionEdad = 0);
Query OK, 1 row affected (0.007 sec)
```