

# BASE DE DATOS EJERCICIOS

---

1. Obtener el listado con todos los pedidos que se han realizado, los pedidos deben estar ordenados por fecha de realización y se deben de mostrar en primer lugar los pedidos más recientes.

```
SELECT *  
FROM pedido  
ORDER BY fecha DESC; # el asterisco lo selecciona todo
```

| # | id | total    | fecha      | id_cliente | id_comercial |
|---|----|----------|------------|------------|--------------|
| 1 | 15 | 370,85   | 2019-03-11 | 1          | 5            |
| 2 | 16 | 2.389,23 | 2019-03-11 | 1          | 5            |
| 3 | 13 | 545,75   | 2019-01-25 | 6          | 1            |
| 4 | 8  | 1.983,43 | 2017-10-10 | 4          | 6            |
| 5 | 1  | 150,5    | 2017-10-05 | 5          | 2            |
| 6 | 3  | 65,26    | 2017-10-05 | 2          | 1            |
| 7 | 5  | 948,5    | 2017-09-10 | 5          | 2            |
| 8 | 12 | 3.045,6  | 2017-04-25 | 2          | 1            |
| 9 | 14 | 145,82   | 2017-02-02 | 6          | 1            |

2. Devolver todos los datos de los dos pedidos de mayor valor.

```
SELECT *  
FROM pedido  
ORDER BY total  
DESC LIMIT 2;
```

| # | id | total   | fecha      | id_cliente | id_comercial |
|---|----|---------|------------|------------|--------------|
| 1 | 7  | 5.760   | 2015-09-10 | 2          | 1            |
| 2 | 12 | 3.045,6 | 2017-04-25 | 2          | 1            |

3. Obtener un listado con los id de los clientes que han realizado algún pedido y no se mostrarán los id que estén repetidos.

```
SELECT DISTINCT (id_cliente)
FROM pedido;
```

| # | id_cliente |
|---|------------|
| 1 | 1          |
| 2 | 2          |
| 3 | 3          |
| 4 | 4          |
| 5 | 5          |
| 6 | 6          |
| 7 | 7          |
| 8 | 8          |

4. Obtener el listado de todos los pedidos que se realizaron en el año 2017 cuya cantidad total sea superior a 500 euros.

```
SELECT *
FROM pedido
WHERE YEAR (fecha) = 2017 AND total > 500;
```

| # | id | total    | fecha      | id_cliente | id_comercial |
|---|----|----------|------------|------------|--------------|
| 1 | 5  | 948,5    | 2017-09-10 | 5          | 2            |
| 2 | 8  | 1.983,43 | 2017-10-10 | 4          | 6            |
| 3 | 12 | 3.045,6  | 2017-04-25 | 2          | 1            |

5. Obtener el listado con el nombre y apellidos de los comerciales que tienen una comisión entre 0,05 y 0,1.

```
SELECT nombre, apellido1, apellido2
FROM comercial
WHERE comision BETWEEN 0.05 AND 0.11;
```

| # | nombre  | apellido1 | apellido2 |
|---|---------|-----------|-----------|
| 1 | Diego   | Flores    | Salas     |
| 2 | Antonio | Vega      | Hernández |
| 3 | Alfredo | Ruiz      | Flores    |

6. Obtener el valor de la comisión de mayor valor que existe en la tabla comercial.

```
SELECT MAX(comision)
FROM comercial; #el max debe ir pegado al paréntesis
```

| # | MAX(comision) |
|---|---------------|
| 1 | 0,15          |

7. Obtener el id nombre y primer apellido de los clientes cuyo segundo apellido no es NULL, el listado tiene que estar ordenado alfabéticamente por nombre y apellido.

```
SELECT id, nombre, apellido1
FROM cliente
WHERE apellido2 IS NOT NULL
ORDER BY nombre, apellido1, apellido2;
```

| # | id | nombre    | apellido1 |
|---|----|-----------|-----------|
| 1 | 1  | Aarón     | Rivero    |
| 2 | 2  | Adela     | Salas     |
| 3 | 3  | Adolfo    | Rubio     |
| 4 | 10 | Daniel    | Santana   |
| 5 | 9  | Guillermo | López     |
| 6 | 5  | Marcos    | Loyola    |
| 7 | 6  | María     | Santana   |
| 8 | 8  | Pepe      | Ruiz      |

8. Obtener el listado con los nombres de los clientes que empiezan por A y terminan por N y también los nombres que empiezan por P el listado deberá estar ordenado alfabéticamente.

```
SELECT nombre
FROM cliente
WHERE nombre LIKE 'a%' AND nombre LIKE '%n'
OR nombre LIKE 'p%';
```

| # | nombre |
|---|--------|
| 1 | Aarón  |
| 2 | Adrián |
| 3 | Pilar  |
| 4 | Pepe   |

9. Obtener el listado de los nombres de los clientes que no empiezan por A y ordenarlos alfabéticamente.

```
SELECT nombre
FROM cliente
WHERE nombre NOT LIKE 'a%'
ORDER BY nombre;
```

| # | nombre    |
|---|-----------|
| 1 | Daniel    |
| 2 | Guillermo |
| 3 | Marcos    |
| 4 | María     |
| 5 | Pepe      |
| 6 | Pilar     |

10. Obtener un listado con los nombres de los comerciales que terminan por "el" u "o" teniendo en cuenta que se deben eliminar los nombres repetidos.

```
SELECT DISTINCT nombre #También se puede usar comercial.nombre
FROM comercial
WHERE nombre LIKE '%el' OR nombre LIKE '%o';
```

| comercial (5r × 1c) |         |
|---------------------|---------|
| #                   | nombre  |
| 1                   | Daniel  |
| 2                   | Diego   |
| 3                   | Antonio |
| 4                   | Manuel  |
| 5                   | Alfredo |

11. De la tabla cliente; el nombre, id, apellido1, apellido2 de la base de datos de VENTAS en la tabla CLIENTE y añadir el "id" de la tabla PEDIDO y CLIENTE.

```
SELECT DISTINCT c.nombre, c.id, c.apellido1, c.apellido2
FROM ventas.cliente c INNER JOIN pedido p ON c.id = p.id
ORDER BY 4 ASC, 3 ASC;
```

| cliente (10r × 4c) |           |    |           |           |
|--------------------|-----------|----|-----------|-----------|
| #                  | nombre    | id | apellido1 | apellido2 |
| 1                  | Pilar     | 7  | Ruiz      | (NULL)    |
| 2                  | Adrián    | 4  | Suárez    | (NULL)    |
| 3                  | Adela     | 2  | Salas     | Díaz      |
| 4                  | Adolfo    | 3  | Rubio     | Flores    |
| 5                  | Guillermo | 9  | López     | Gómez     |
| 6                  | Aarón     | 1  | Rivero    | Gómez     |
| 7                  | Daniel    | 10 | Santana   | Loyola    |
| 8                  | Marcos    | 5  | Loyola    | Méndez    |
| 9                  | María     | 6  | Santana   | Moreno    |
| 10                 | Pepe      | 8  | Ruiz      | Santana   |

# Dado que hay 4 elementos, ordenados por nombre, id, apellido1 y apellido2 al poner el ORDER BY 4 o 3; el 4 sería apellido2 y el 3 apellido1. Nos lo ordena según ese criterio y de forma ascendente, de ahí el ASC en ambos.

11. Obtener el nombre y apellidos de la tabla CLIENTE y la fecha y el total de la tabla PEDIDO Obtener un listado de todos los pedidos de cada cliente y ordenados por el segundo apellido y en orden ascendente.

```
SELECT c.nombre, c.apellido1, c.apellido2, p.fecha, p.total
FROM cliente c INNER JOIN pedido p ON c.id = p.id_cliente;
ORDER BY c.apellido2 ASC;
```

| cliente (16r x 5c) |        |           |           |            |          |
|--------------------|--------|-----------|-----------|------------|----------|
| #                  | nombre | apellido1 | apellido2 | fecha      | total    |
| 1                  | Pilar  | Ruiz      | (NULL)    | 2016-07-27 | 2.400,6  |
| 2                  | Adrián | Suárez    | (NULL)    | 2017-10-10 | 1.983,43 |
| 3                  | Adela  | Salas     | Díaz      | 2017-10-05 | 65,26    |
| 4                  | Adela  | Salas     | Díaz      | 2015-09-10 | 5.760    |
| 5                  | Adela  | Salas     | Díaz      | 2017-04-25 | 3.045,6  |
| 6                  | Adolfo | Rubio     | Flores    | 2016-08-17 | 75,29    |
| 7                  | Aarón  | Rivero    | Gómez     | 2016-09-10 | 270,65   |
| 8                  | Aarón  | Rivero    | Gómez     | 2019-03-11 | 370,85   |
| 9                  | Aarón  | Rivero    | Gómez     | 2019-03-11 | 2.389,23 |
| 10                 | Marcos | Loyola    | Méndez    | 2017-10-05 | 150,5    |
| 11                 | Marcos | Loyola    | Méndez    | 2017-09-10 | 948,5    |
| 12                 | María  | Santana   | Moreno    | 2019-01-25 | 545,75   |
| 13                 | María  | Santana   | Moreno    | 2017-02-02 | 145,82   |
| 14                 | Pepe   | Ruiz      | Santana   | 2016-08-17 | 110,5    |
| 15                 | Pepe   | Ruiz      | Santana   | 2016-10-10 | 2.480,4  |
| 16                 | Pepe   | Ruiz      | Santana   | 2015-06-27 | 250,45   |






**12. Obtener un listado que muestre todos los pedidos en los que ha participado un comercial, el resultado debe mostrar el dato de los pedidos y de los comerciales. Deben estar ordenados por el primer apellido. Como datos queremos nombre, apellidos, id, pedido, fecha y el total de los pedidos.**

```
SELECT c.nombre, c.apellido1, c.apellido2, c.id, p.fecha, p.total
FROM cliente c INNER JOIN pedido p ON c.id = p.id_cliente
ORDER BY c.apellido1 ASC;
```

| cliente (16r x 6c) |        |           |           |    |            |          |
|--------------------|--------|-----------|-----------|----|------------|----------|
| #                  | nombre | apellido1 | apellido2 | id | fecha      | total    |
| 1                  | Marcos | Loyola    | Méndez    | 5  | 2017-10-05 | 150,5    |
| 2                  | Marcos | Loyola    | Méndez    | 5  | 2017-09-10 | 948,5    |
| 3                  | Aarón  | Rivero    | Gómez     | 1  | 2016-09-10 | 270,65   |
| 4                  | Aarón  | Rivero    | Gómez     | 1  | 2019-03-11 | 370,85   |
| 5                  | Aarón  | Rivero    | Gómez     | 1  | 2019-03-11 | 2.389,23 |
| 6                  | Adolfo | Rubio     | Flores    | 3  | 2016-08-17 | 75,29    |
| 7                  | Pepe   | Ruiz      | Santana   | 8  | 2016-08-17 | 110,5    |
| 8                  | Pilar  | Ruiz      | (NULL)    | 7  | 2016-07-27 | 2.400,6  |
| 9                  | Pepe   | Ruiz      | Santana   | 8  | 2016-10-10 | 2.480,4  |
| 10                 | Pepe   | Ruiz      | Santana   | 8  | 2015-06-27 | 250,45   |
| 11                 | Adela  | Salas     | Díaz      | 2  | 2017-10-05 | 65,26    |
| 12                 | Adela  | Salas     | Díaz      | 2  | 2015-09-10 | 5.760    |
| 13                 | Adela  | Salas     | Díaz      | 2  | 2017-04-25 | 3.045,6  |
| 14                 | María  | Santana   | Moreno    | 6  | 2019-01-25 | 545,75   |
| 15                 | María  | Santana   | Moreno    | 6  | 2017-02-02 | 145,82   |
| 16                 | Adrián | Suárez    | (NULL)    | 4  | 2017-10-10 | 1.983,43 |

**13. Obtener un listado que muestre todos los clientes, con todos los pedidos que ha realizado y con todos los datos de los comerciales asociados a cada pedido.**

```
SELECT *
FROM cliente INNER JOIN pedido ON cliente.id = pedido.id_cliente
INNER JOIN comercial ON pedido.id_comercial = comercial.id;
```

| Resultado #1 (16r x 16c) |    |  |           |           |         |           |    |   |            |            |  |  |  |           |           |          |
|--------------------------|----|--|-----------|-----------|---------|-----------|----|---|------------|------------|--|--|--|-----------|-----------|----------|
| #                        | id |  nombre | apellido1 | apellido2 | ciudad  | categoria | id |  total | fecha      | id_cliente |  id_comercial |  id |  nombre | apellido1 | apellido2 | comision |
| 1                        | 5  | Marcos   | Loyola    | Méndez    | Almería | 200       | 1  | 150,5   | 2017-10-05 | 5          | 2  | 2  | Juan   | Gómez     | López     | 0,13     |
| 2                        | 1  | Aarón  | Rivero    | Gómez     | Almería | 100       | 2  | 270,65  | 2016-09-10 | 1          | 5  | 5  | Antonio  | Carretero | Ortega    | 0,12     |
| 3                        | 2  | Adela  | Salas     | Díaz      | Granada | 200       | 3  | 65,26   | 2017-10-05 | 2          | 1  | 1  | Daniel   | Sáez      | Vega      | 0,15     |
| 4                        | 8  | Pepe   | Ruiz      | Santana   | Huelva  | 200       | 4  | 110,5   | 2016-08-17 | 8          | 3  | 3  | Diego  | Flores    | Salas     | 0,11     |
| 5                        | 5  | Marcos   | Loyola    | Méndez    | Almería | 200       | 5  | 948,5   | 2017-09-10 | 5          | 2  | 2  | Juan   | Gómez     | López     | 0,13     |
| 6                        | 7  | Pilar  | Ruiz      | (NULL)    | Sevilla | 300       | 6  | 2.400,6   | 2016-07-27 | 7          | 1  | 1  | Daniel   | Sáez      | Vega      | 0,15     |
| 7                        | 2  | Adela  | Salas     | Díaz      | Granada | 200       | 7  | 5.760   | 2015-09-10 | 2          | 1  | 1  | Daniel   | Sáez      | Vega      | 0,15     |
| 8                        | 4  | Adrián   | Suárez    | (NULL)    | Jaén    | 300       | 8  | 1.983,43  | 2017-10-10 | 4          | 6  | 6  | Manuel   | Dominguez | Hernández | 0,13     |
| 9                        | 8  | Pepe   | Ruiz      | Santana   | Huelva  | 200       | 9  | 2.480,4   | 2016-10-10 | 8          | 3  | 3  | Diego  | Flores    | Salas     | 0,11     |
| 10                       | 8  | Pepe   | Ruiz      | Santana   | Huelva  | 200       | 10 | 250,45  | 2015-06-27 | 8          | 2  | 2  | Juan   | Gómez     | López     | 0,13     |
| 11                       | 3  | Adolfo   | Rubio     | Flores    | Sevilla | (NULL)    | 11 | 75,29   | 2016-08-17 | 3          | 7  | 7  | Antonio  | Vega      | Hernández | 0,11     |
| 12                       | 2  | Adela  | Salas     | Díaz      | Granada | 200       | 12 | 3.045,6   | 2017-04-25 | 2          | 1  | 1  | Daniel   | Sáez      | Vega      | 0,15     |
| 13                       | 6  | María  | Santana   | Moreno    | Cádiz   | 100       | 13 | 545,75  | 2019-01-25 | 6          | 1  | 1  | Daniel   | Sáez      | Vega      | 0,15     |
| 14                       | 6  | María  | Santana   | Moreno    | Cádiz   | 100       | 14 | 145,82  | 2017-02-02 | 6          | 1  | 1  | Daniel   | Sáez      | Vega      | 0,15     |
| 15                       | 1  | Aarón  | Rivero    | Gómez     | Almería | 100       | 15 | 370,85  | 2019-03-11 | 1          | 5  | 5  | Antonio  | Carretero | Ortega    | 0,12     |
| 16                       | 1  | Aarón  | Rivero    | Gómez     | Almería | 100       | 16 | 2.389,23  | 2019-03-11 | 1          | 5  | 5  | Antonio  | Carretero | Ortega    | 0,12     |

## INNER JOIN de "Ventas"

1. Devuelve un listado de todos los clientes que realizaron un pedido durante el año 2017, cuya cantidad esté entre 300 € y 1000 €.

**USE** ventas;

**SELECT** c.id, c.nombre, c.apellido1, c.apellido2, p.fecha, p.total

**FROM** cliente c

**INNER JOIN** pedido p **ON** c.id = p.id\_cliente

**WHERE** YEAR (p.fecha) = 2017

**AND** p.total **BETWEEN** 300 **AND** 1000;

| cliente (1r x 6c) |    |        |           |           |            |       |  |
|-------------------|----|--------|-----------|-----------|------------|-------|--|
| #                 | id | nombre | apellido1 | apellido2 | fecha      | total |  |
| 1                 | 5  | Marcos | Loyola    | Méndez    | 2017-09-10 | 948,5 |  |

2. Devuelve el nombre y los apellidos de todos los comerciales que han participado en algún pedido realizado por María Santana Moreno.

**SELECT DISTINCT** c.nombre, c.apellido1, c.apellido2

**FROM** comercial c

**INNER JOIN** pedido p **ON** c.id = p.id\_comercial

**INNER JOIN** cliente cli **ON** p.id\_cliente = cli.id

**WHERE** cli.nombre = 'María'

**AND** cli.apellido1 = 'Santana'

**AND** cli.apellido2 = 'Moreno';

| comercial (1r x 3c) |        |           |           |
|---------------------|--------|-----------|-----------|
| #                   | nombre | apellido1 | apellido2 |
| 1                   | Daniel | Sáez      | Vega      |



3. Devuelve el nombre de todos los clientes que han realizado algún pedido con el comercial Daniel Sáez Vega.

**USE** ventas;

**SELECT DISTINCT** c.nombre, c.apellido1, c.apellido2

**FROM** cliente c

**INNER JOIN** pedido p **ON** c.id = p.id\_comercial

**INNER JOIN** comercial com **ON** p.id\_cliente = com.id

**WHERE** com.nombre = 'Daniel'

**AND** com.apellido1 = 'Sáez'

**AND** com.apellido2 = 'Vega';

| cliente (1r × 3c) |        |           |           |
|-------------------|--------|-----------|-----------|
| #                 | nombre | apellido1 | apellido2 |
| 1                 | Marcos | Loyola    | Méndez    |

## Consultas simples de la base de datos Empleados

1. Lista el código de los departamentos de los empleados que aparecen en la tabla empleado, eliminando los códigos que aparecen repetidos.

**USE** empleados;

**SELECT DISTINCT** nif

**FROM** empleado;

| empleado (13r × 1c) |           |
|---------------------|-----------|
| #                   | nif       |
| 1                   | 17087203C |
| 2                   | 32481596F |
| 3                   | 38382980M |
| 4                   | 41234836R |
| 5                   | 46384486H |
| 6                   | 56399183D |
| 7                   | 67389283A |
| 8                   | 71651431Z |
| 9                   | 77705545E |
| 10                  | 80576669X |
| 11                  | 82635162B |
| 12                  | R6970642B |
| 13                  | Y5575632D |

2. Lista el nombre y apellidos de los empleados en una única columna.

```
SELECT CONCAT(nombre,' ', apellido1,' ', apellido2) AS EMPLEADOS  
FROM empleado;
```

| #  | EMPLEADOS                  |
|----|----------------------------|
| 1  | Aarón Rivero Gómez         |
| 2  | Adela Salas Díaz           |
| 3  | Adolfo Rubio Flores        |
| 4  | (NULL)                     |
| 5  | Marcos Loyola Méndez       |
| 6  | María Santana Moreno       |
| 7  | (NULL)                     |
| 8  | Pepe Ruiz Santana          |
| 9  | Juan Gómez López           |
| 10 | Diego Flores Salas         |
| 11 | Marta Herrera Gil          |
| 12 | Irene Salas Flores         |
| 13 | Juan Antonio Sáez Guerrero |


3. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula.

```
SELECT UPPER (CONCAT (nombre, ' ', apellido1, ' ', apellido2)) AS  
EMPLEADOS  
FROM empleado;
```

| #  | EMPLEADOS                  |
|----|----------------------------|
| 1  | AARÓN RIVERO GÓMEZ         |
| 2  | ADELA SALAS DÍAZ           |
| 3  | ADOLFO RUBIO FLORES        |
| 4  | (NULL)                     |
| 5  | MARCOS LOYOLA MÉNDEZ       |
| 6  | MARÍA SANTANA MORENO       |
| 7  | (NULL)                     |
| 8  | PEPE RUIZ SANTANA          |
| 9  | JUAN GÓMEZ LÓPEZ           |
| 10 | DIEGO FLORES SALAS         |
| 11 | MARTA HERRERA GIL          |
| 12 | IRENE SALAS FLORES         |
| 13 | JUAN ANTONIO SÁEZ GUERRERO |

4. *Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en minúscula.*

```
SELECT LOWER (CONCAT (nombre, ' ', apellido1, ' ', apellido2)) AS  
EMPLEADOS  
FROM empleado;
```



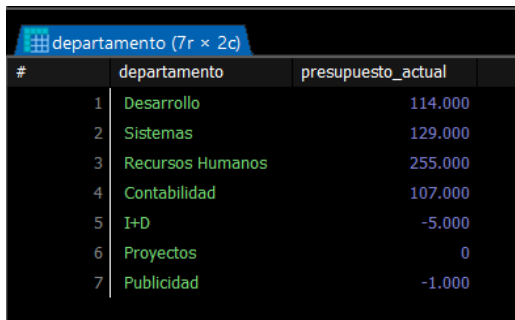
The screenshot shows a database query result in a dark-themed interface. The window title is 'empleado (13r x 1c)'. The table has two columns: '#' and 'EMPLEADOS'. The data is as follows:

| #  | EMPLEADOS                  |
|----|----------------------------|
| 1  | aarón rivero gómez         |
| 2  | adela salas díaz           |
| 3  | adolfo rubio flores        |
| 4  | (NULL)                     |
| 5  | marcos loyola méndez       |
| 6  | maría santana moreno       |
| 7  | (NULL)                     |
| 8  | pepe ruiz santana          |
| 9  | juan gómez lópez           |
| 10 | diego flores salas         |
| 11 | marta herrera gil          |
| 12 | irene salas flores         |
| 13 | juan antonio sáez guerrero |

5. *Lista el código de los empleados junto al nif, pero el nif deberá aparecer en dos columnas, una mostrará únicamente los dígitos del nif y la otra la letra.*

6. Lista el nombre de cada departamento y el valor del presupuesto actual del que dispone. Para calcular este dato tendrá que restar al valor del presupuesto inicial (columna presupuesto) los gastos que se han generado (columna gastos). Tenga en cuenta que en algunos casos pueden existir valores negativos. Utilice un alias apropiado para la nueva columna que está calculando.

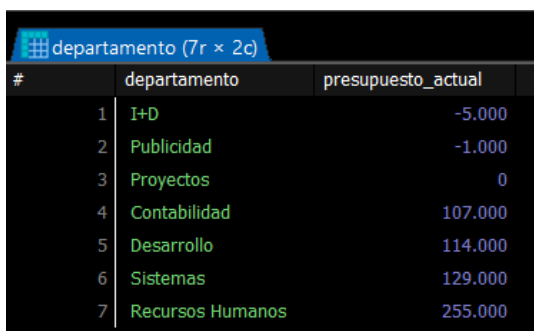
```
SELECT nombre AS departamento,  
(presupuesto - gastos) AS presupuesto_actual  
FROM departamento;
```



| # | departamento     | presupuesto_actual |
|---|------------------|--------------------|
| 1 | Desarrollo       | 114.000            |
| 2 | Sistemas         | 129.000            |
| 3 | Recursos Humanos | 255.000            |
| 4 | Contabilidad     | 107.000            |
| 5 | I+D              | -5.000             |
| 6 | Proyectos        | 0                  |
| 7 | Publicidad       | -1.000             |

7. Lista el nombre de los departamentos y el valor del presupuesto actual ordenado de forma ascendente.

```
SELECT nombre AS departamento, (presupuesto - gastos) AS  
presupuesto_actual  
FROM departamento  
ORDER BY presupuesto_actual ASC;
```



| # | departamento     | presupuesto_actual |
|---|------------------|--------------------|
| 1 | I+D              | -5.000             |
| 2 | Publicidad       | -1.000             |
| 3 | Proyectos        | 0                  |
| 4 | Contabilidad     | 107.000            |
| 5 | Desarrollo       | 114.000            |
| 6 | Sistemas         | 129.000            |
| 7 | Recursos Humanos | 255.000            |

8. Lista el nombre de todos los departamentos ordenados de forma ascendente.

```
SELECT nombre  
FROM departamento  
ORDER BY nombre ASC;
```



A screenshot of a database query result window titled 'departamento (7r x 1c)'. It displays a table with two columns: '#' and 'nombre'. The data is ordered alphabetically from A to Z.

| # | nombre           |
|---|------------------|
| 1 | Contabilidad     |
| 2 | Desarrollo       |
| 3 | I+D              |
| 4 | Proyectos        |
| 5 | Publicidad       |
| 6 | Recursos Humanos |
| 7 | Sistemas         |

9. Lista el nombre de todos los departamentos ordenados de forma descendente.

```
SELECT nombre  
FROM departamento  
ORDER BY nombre DESC;
```



A screenshot of a database query result window titled 'departamento (7r x 1c)'. It displays a table with two columns: '#' and 'nombre'. The data is ordered alphabetically from Z to A.

| # | nombre           |
|---|------------------|
| 1 | Sistemas         |
| 2 | Recursos Humanos |
| 3 | Publicidad       |
| 4 | Proyectos        |
| 5 | I+D              |
| 6 | Desarrollo       |
| 7 | Contabilidad     |

10. Lista los apellidos y el nombre de todos los empleados, ordenados de forma alfabética teniendo en cuenta en primer lugar sus apellidos y luego su nombre.

```
SELECT apellido1, apellido2, nombre  
FROM empleado  
ORDER BY apellido1, apellido2 ASC, nombre ASC;
```

| #  | apellido1 | apellido2 | nombre       |
|----|-----------|-----------|--------------|
| 1  | Flores    | Salas     | Diego        |
| 2  | Gómez     | López     | Juan         |
| 3  | Herrera   | Gil       | Marta        |
| 4  | Loyola    | Méndez    | Marcos       |
| 5  | Rivero    | Gómez     | Aarón        |
| 6  | Rubio     | Flores    | Adolfo       |
| 7  | Ruiz      | (NULL)    | Pilar        |
| 8  | Ruiz      | Santana   | Pepe         |
| 9  | Sáez      | Guerrero  | Juan Antonio |
| 10 | Salas     | Díaz      | Adela        |
| 11 | Salas     | Flores    | Irene        |
| 12 | Santana   | Moreno    | María        |
| 13 | Suárez    | (NULL)    | Adrián       |

11. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen mayor presupuesto.

```
SELECT nombre, presupuesto  
FROM departamento  
ORDER BY presupuesto DESC  
LIMIT 3;
```

| # | nombre           | presupuesto |
|---|------------------|-------------|
| 1 | I+D              | 375.000     |
| 2 | Recursos Humanos | 280.000     |
| 3 | Sistemas         | 150.000     |

12. Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen mayor gasto.

```
SELECT nombre, gastos
FROM departamento
ORDER BY gastos DESC
LIMIT 2;
```

| departamento (2r x 2c) |                  |         |
|------------------------|------------------|---------|
| #                      | nombre           | gastos  |
| 1                      | I+D              | 380.000 |
| 2                      | Recursos Humanos | 25.000  |

13. Devuelve una lista con 5 filas a partir de la tercera fila de la tabla empleado. La tercera fila se debe incluir en la respuesta. La respuesta debe incluir todas las columnas de la tabla empleado.

```
SELECT *
FROM empleado
LIMIT 5 OFFSET 2; #OFFSET OMITTE LAS PRIMERAS 2 FILAS EMPEZANDO POR LA TERCERA
```

| empleado (5r x 6c) |    |           |        |           |           |                 |  |
|--------------------|----|-----------|--------|-----------|-----------|-----------------|--|
| #                  | id | nif       | nombre | apellido1 | apellido2 | id_departamento |  |
| 1                  | 3  | R6970642B | Adolfo | Rubio     | Flores    | 3               |  |
| 2                  | 4  | 77705545E | Adrián | Suárez    | (NULL)    | 4               |  |
| 3                  | 5  | 17087203C | Marcos | Loyola    | Méndez    | 5               |  |
| 4                  | 6  | 38382980M | María  | Santana   | Moreno    | 1               |  |
| 5                  | 7  | 80576669X | Pilar  | Ruiz      | (NULL)    | 2               |  |

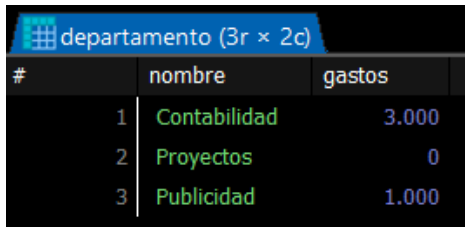
14. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto mayor o igual a 150000 euros.

```
SELECT *
FROM departamento
WHERE presupuesto >= 150000;
```

| departamento (3r x 4c) |    |                  |             |         |  |
|------------------------|----|------------------|-------------|---------|--|
| #                      | id | nombre           | presupuesto | gastos  |  |
| 1                      | 2  | Sistemas         | 150.000     | 21.000  |  |
| 2                      | 3  | Recursos Humanos | 280.000     | 25.000  |  |
| 3                      | 5  | I+D              | 375.000     | 380.000 |  |

15. Devuelve una lista con el nombre de los departamentos y el gasto, de aquellos que tienen menos de 5000 euros de gastos.

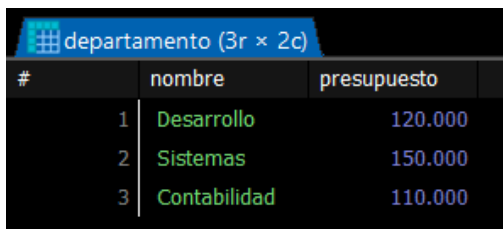
```
SELECT nombre, gastos
FROM departamento
WHERE gastos < 5000;
```



| # | nombre       | gastos |
|---|--------------|--------|
| 1 | Contabilidad | 3.000  |
| 2 | Proyectos    | 0      |
| 3 | Publicidad   | 1.000  |

16. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto entre 100000 y 200000 euros. Sin utilizar el operador BETWEEN.

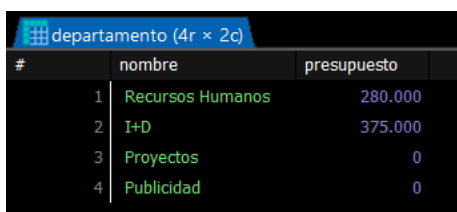
```
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto >= 100000 AND presupuesto <= 200000; #ESTO ES SIN USAR BETWEEN
```



| # | nombre       | presupuesto |
|---|--------------|-------------|
| 1 | Desarrollo   | 120.000     |
| 2 | Sistemas     | 150.000     |
| 3 | Contabilidad | 110.000     |

17. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre 100000 y 200000 euros. Sin utilizar el operador BETWEEN.

```
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto < 100000 OR presupuesto > 200000; #ESTO ES SIN USAR BETWEEN
```



| # | nombre           | presupuesto |
|---|------------------|-------------|
| 1 | Recursos Humanos | 280.000     |
| 2 | I+D              | 375.000     |
| 3 | Proyectos        | 0           |
| 4 | Publicidad       | 0           |



**18. Devuelve una lista con el nombre de los departamentos, gastos y presupuesto, de aquellos departamentos donde los gastos sean mayores que el presupuesto del que disponen.**

```
SELECT *  
FROM departamento  
WHERE gastos > presupuesto;
```

| departamento (2r x 4c) |    |            |             |         |  |
|------------------------|----|------------|-------------|---------|--|
| #                      | id | nombre     | presupuesto | gastos  |  |
| 1                      | 5  | I+D        | 375.000     | 380.000 |  |
| 2                      | 7  | Publicidad | 0           | 1.000   |  |

**19. Devuelve una lista con el nombre de los departamentos, gastos y presupuesto, de aquellos departamentos donde los gastos sean iguales al presupuesto del que disponen.**

```
SELECT *  
FROM departamento  
WHERE gastos = presupuesto;
```

| departamento (1r x 4c) |    |           |             |        |  |
|------------------------|----|-----------|-------------|--------|--|
| #                      | id | nombre    | presupuesto | gastos |  |
| 1                      | 6  | Proyectos | 0           | 0      |  |

20. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Sin utilizar el operador IN.

```
SELECT *  
FROM empleado  
WHERE apellido2 LIKE '%Díaz' OR apellido2 LIKE '%Moreno%';
```

| empleado (2r x 6c) |    |           |        |           |           |                 |  |
|--------------------|----|-----------|--------|-----------|-----------|-----------------|--|
| #                  | id | nif       | nombre | apellido1 | apellido2 | id_departamento |  |
| 1                  | 2  | Y5575632D | Adela  | Salas     | Díaz      | 2               |  |
| 2                  | 6  | 38382980M | María  | Santana   | Moreno    | 1               |  |

21. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Utilizando el operador IN.

```
SELECT *  
FROM empleado  
WHERE apellido2 IN ('Díaz', 'Moreno');
```

| empleado (2r x 6c) |    |           |        |           |           |                 |  |
|--------------------|----|-----------|--------|-----------|-----------|-----------------|--|
| #                  | id | nif       | nombre | apellido1 | apellido2 | id_departamento |  |
| 1                  | 2  | Y5575632D | Adela  | Salas     | Díaz      | 2               |  |
| 2                  | 6  | 38382980M | María  | Santana   | Moreno    | 1               |  |

22. Lista los nombres, apellidos y nif de los empleados que trabajan en los departamentos 2, 4 o 5.

```
SELECT nombre, apellido1, apellido2, nif  
FROM empleado  
WHERE id_departamento IN (2,4,5);
```

| empleado (6r x 4c) |        |           |           |           |
|--------------------|--------|-----------|-----------|-----------|
| #                  | nombre | apellido1 | apellido2 | nif       |
| 1                  | Adela  | Salas     | Díaz      | Y5575632D |
| 2                  | Pilar  | Ruiz      | (NULL)    | 80576669X |
| 3                  | Juan   | Gómez     | López     | 56399183D |
| 4                  | Adrián | Suárez    | (NULL)    | 77705545E |
| 5                  | Marcos | Loyola    | Méndez    | 17087203C |
| 6                  | Diego  | Flores    | Salas     | 46384486H |

## Consultas Multitabla base de datos Empleados

1. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno.

```
SELECT e.nombre, e.apellido1, e.apellido2, d.nombre  
FROM empleado e, departamento d;
```

| empleado (91r x 4c) |        |           |           |                  |
|---------------------|--------|-----------|-----------|------------------|
| #                   | nombre | apellido1 | apellido2 | nombre           |
| 1                   | Aarón  | Rivero    | Gómez     | Publicidad       |
| 2                   | Aarón  | Rivero    | Gómez     | Proyectos        |
| 3                   | Aarón  | Rivero    | Gómez     | I+D              |
| 4                   | Aarón  | Rivero    | Gómez     | Contabilidad     |
| 5                   | Aarón  | Rivero    | Gómez     | Recursos Humanos |
| 6                   | Aarón  | Rivero    | Gómez     | Sistemas         |
| 7                   | Aarón  | Rivero    | Gómez     | Desarrollo       |

2. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno. Ordena el resultado, en primer lugar por el nombre del departamento (en orden alfabético) y en segundo lugar por los apellidos y el nombre de los empleados.

```
SELECT e.nombre, e.apellido1, e.apellido2, d.nombre  
FROM empleado e  
JOIN departamento d ON e.id = d.id  
ORDER BY d.nombre ASC, e.apellido1 ASC, e.apellido2 ASC, e.nombre ASC;
```

| empleado (7r x 4c) |        |           |           |                  |
|--------------------|--------|-----------|-----------|------------------|
| #                  | nombre | apellido1 | apellido2 | nombre           |
| 1                  | Adrián | Suárez    | (NULL)    | Contabilidad     |
| 2                  | Aarón  | Rivero    | Gómez     | Desarrollo       |
| 3                  | Marcos | Loyola    | Méndez    | I+D              |
| 4                  | María  | Santana   | Moreno    | Proyectos        |
| 5                  | Pilar  | Ruiz      | (NULL)    | Publicidad       |
| 6                  | Adolfo | Rubio     | Flores    | Recursos Humanos |
| 7                  | Adela  | Salas     | Díaz      | Sistemas         |

3. Devuelve un listado con el código y el nombre del departamento, solamente de aquellos departamentos que tienen empleados.

```
SELECT DISTINCT d.id, d.nombre  
FROM departamento d  
JOIN empleado e ON d.id = e.id;
```

| departamento (7r x 2c) |    |                  |
|------------------------|----|------------------|
| #                      | id | nombre           |
| 1                      | 1  | Desarrollo       |
| 2                      | 2  | Sistemas         |
| 3                      | 3  | Recursos Humanos |
| 4                      | 4  | Contabilidad     |
| 5                      | 5  | I+D              |
| 6                      | 6  | Proyectos        |
| 7                      | 7  | Publicidad       |

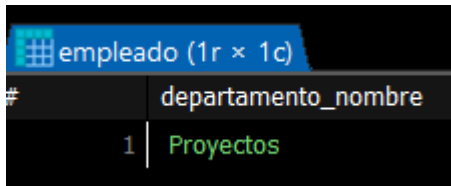
4. Devuelve un listado con el código, el nombre del departamento y el valor del presupuesto actual del que dispone, solamente de aquellos departamentos que tienen empleados. El valor del presupuesto actual lo puede calcular restando al valor del presupuesto inicial (columna presupuesto) el valor de los gastos que ha generado (columna gastos).

```
SELECT DISTINCT d.id, d.nombre AS departamento_nombre, (d.presupuesto -  
d.gastos) AS presupuesto_actual  
FROM departamento d  
JOIN empleado e ON d.id = e.id;
```

| departamento (7r x 3c) |    |                     |                    |
|------------------------|----|---------------------|--------------------|
| #                      | id | departamento_nombre | presupuesto_actual |
| 1                      | 1  | Desarrollo          | 114.000            |
| 2                      | 2  | Sistemas            | 129.000            |
| 3                      | 3  | Recursos Humanos    | 255.000            |
| 4                      | 4  | Contabilidad        | 107.000            |
| 5                      | 5  | I+D                 | -5.000             |
| 6                      | 6  | Proyectos           | 0                  |
| 7                      | 7  | Publicidad          | -1.000             |

5. Devuelve el nombre del departamento donde trabaja el empleado que tiene el nif 38382980M.

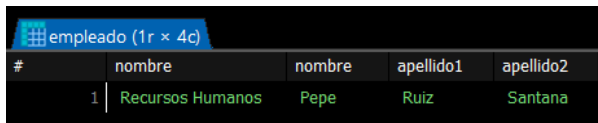
```
SELECT d.nombre AS departamento_nombre  
FROM empleado e  
JOIN departamento d ON e.id = d.id  
WHERE e.nif = '38382980M';
```



| empleado (1r x 1c) |                     |
|--------------------|---------------------|
| #                  | departamento_nombre |
| 1                  | Proyectos           |

6. Devuelve el nombre del departamento donde trabaja el empleado Pepe Ruiz Santana.

```
SELECT d.nombre, e.nombre, e.apellido1, e.apellido2  
FROM empleado e  
JOIN departamento d ON d.id= e.id_departamento  
WHERE e.nombre = 'Pepe' AND e.apellido1 = 'Ruiz' AND e.apellido2 =  
'Santana';
```



| empleado (1r x 4c) |                  |        |           |           |
|--------------------|------------------|--------|-----------|-----------|
| #                  | nombre           | nombre | apellido1 | apellido2 |
| 1                  | Recursos Humanos | Pepe   | Ruiz      | Santana   |

7. Devuelve un listado con los datos de los empleados que trabajan en el departamento de I+D. Ordena el resultado alfabéticamente.
8. Devuelve un listado con los datos de los empleados que trabajan en el departamento de Sistemas, Contabilidad o I+D. Ordena el resultado alfabéticamente.
9. Devuelve una lista con el nombre de los empleados que tienen los departamentos que no tienen un presupuesto entre 100000 y 200000 euros.