

TERCER TALLER SQL.

**PRESENTADO POR:
RUBÉN DARÍO VELASCO BURBANO.**

**PRESENTADO A:
ING. HENRY EDUARDO BASTIDAS**

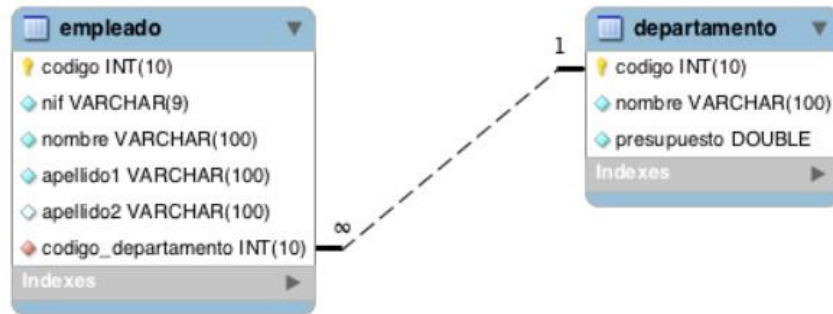
PRESENTACIÓN TERCERA EVIDENCIA.

SERVICIO NACIONAL DE APRENDIZAJE S.E.N.A

POPAYÁN, 2023.

Ejercicios prácticos SQL taller Final

Modelo entidad/relación



Cree el schema de bases de datos en MYSQL e inserte los siguientes valores:

```
(1, 'Desarrollo', 120000, 6000);
(2, 'Sistemas', 150000, 21000);
(3, 'Recursos Humanos', 280000, 25000);
(4, 'Contabilidad', 110000, 3000);
(5, 'I+D', 375000, 380000); (6,
    'Proyectos', 0, 0);
(7, 'Publicidad', 0, 1000);

(1, '32481596F', 'Aarón', 'Rivero', 'Gómez', 1);
(2, 'Y5575632D', 'Adela', 'Salas', 'Díaz', 2);
(3, 'R6970642B', 'Adolfo', 'Rubio', 'Flores', 3);
(4, '77705545E', 'Adrián', 'Suárez', NULL, 4);
(5, '17087203C', 'Marcos', 'Loyola', 'Méndez', 5);
(6, '38382980M', 'María', 'Santana', 'Moreno', 1);
(7, '80576669X', 'Pilar', 'Ruiz', NULL, 2);
(8, '71651431Z', 'Pepe', 'Ruiz', 'Santana', 3);
(9, '56399183D', 'Juan', 'Gómez', 'López', 2);
(10, '46384486H', 'Diego', 'Flores', 'Salas', 5);
(11, '67389283A', 'Marta', 'Herrera', 'Gil', 1);
(12, '41234836R', 'Irene', 'Salas', 'Flores', NULL);
(13, '82635162B', 'Juan Antonio', 'Sáez', 'Guerrero', NULL);
```

A partir del siguiente modelo realice las siguientes operaciones:

1. Cree un usuario nuevo llamado "taller"

```
mysql> CREATE USER 'taller'@'localhost' IDENTIFIED BY 'contraseña';
Query OK, 0 rows affected (0.03 sec)
```

2. Cree un nuevo schema llamado empleado;

```
mysql> create database empleado;
Query OK, 1 row affected (0.01 sec)
```

3. Asigne los privilegios requeridos para el usuario "taller" de tal manera que pueda tener acceso al schema creado.

```
mysql> GRANT ALL PRIVILEGES ON empleado.* TO 'taller'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

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

```
mysql> select empleados.codigo, empleados.nombre as nombre_empleados, departamentos.codigo as codigo_departame
ntos, departamentos.nombre as nombre_departamentos
-> FROM empleados
-> inner join departamentos on empleados.codigo_departamento = departamentos.codigo;
+-----+-----+-----+-----+
| codigo | nombre_empleados | codigo_departamentos | nombre_departamentos |
+-----+-----+-----+-----+
| 1 | Aarón | 1 | Desarrollo |
| 6 | María | 1 | Desarrollo |
| 11 | Marta | 1 | Desarrollo |
| 2 | Adela | 2 | Sistemas |
| 7 | Pilar | 2 | Sistemas |
| 9 | Juan | 2 | Sistemas |
| 3 | Adolfo | 3 | Recursos Humanos |
| 8 | Pepe | 3 | Recursos Humanos |
| 4 | Adrián | 4 | Contabilidad |
| 5 | Marcos | 5 | I+D |
| 10 | Diego | 5 | I+D |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

5. 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.

```
mysql> SELECT e.codigo, e.nombre AS nombre_empleado, e.apellido1, e.apellido2, d.codigo AS codigo_departament
o, d.nombre AS nombre_departamento
-> FROM empleados e
-> JOIN departamentos d ON e.codigo_departamento = d.codigo
-> ORDER BY d.nombre, e.apellido1, e.apellido2, e.nombre;
+-----+-----+-----+-----+-----+-----+
| codigo | nombre_empleado | apellido1 | apellido2 | codigo_departamento | nombre_departamento |
+-----+-----+-----+-----+-----+-----+
| 4 | Adrián | Suárez | NULL | 4 | Contabilidad |
| 11 | Marta | Herrera | Gil | 1 | Desarrollo |
| 1 | Aarón | Rivero | Gómez | 1 | Desarrollo |
| 6 | María | Santana | Moreno | 1 | Desarrollo |
| 10 | Diego | Flores | Salas | 5 | I+D |
| 5 | Marcos | Loyola | Méndez | 5 | I+D |
| 3 | Adolfo | Rubio | Flores | 3 | Recursos Humanos |
| 8 | Pepe | Ruiz | Santana | 3 | Recursos Humanos |
| 9 | Juan | Gómez | López | 2 | Sistemas |
| 7 | Pilar | Ruiz | NULL | 2 | Sistemas |
| 2 | Adela | Salas | Díaz | 2 | Sistemas |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

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

```
mysql> select departamentos.codigo, departamentos.nombre
-> FROM departamentos
-> inner join empleados on empleados.codigo_departamento= departamentos.codigo;
+-----+-----+
| codigo | nombre |
+-----+-----+
| 1      | Desarrollo |
| 1      | Desarrollo |
| 1      | Desarrollo |
| 2      | Sistemas  |
| 2      | Sistemas  |
| 2      | Sistemas  |
| 3      | Recursos Humanos |
| 3      | Recursos Humanos |
| 4      | Contabilidad |
| 5      | I+D       |
| 5      | I+D       |
+-----+-----+
11 rows in set (0.00 sec)
```

7. 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).

```
mysql> SELECT departamentos.codigo, departamentos.nombre, (departamentos.presupuesto - departamentos.gastos
AS presupuesto_actual
-> FROM departamentos
-> inner JOIN empleados ON empleados.codigo_departamento = departamentos.codigo;
+-----+-----+-----+
| codigo | nombre | presupuesto_actual |
+-----+-----+-----+
| 1      | Desarrollo | 114000 |
| 1      | Desarrollo | 114000 |
| 1      | Desarrollo | 114000 |
| 2      | Sistemas  | 129000 |
| 2      | Sistemas  | 129000 |
| 2      | Sistemas  | 129000 |
| 3      | Recursos Humanos | 255000 |
| 3      | Recursos Humanos | 255000 |
| 4      | Contabilidad | 107000 |
| 5      | I+D       | -5000  |
| 5      | I+D       | -5000  |
+-----+-----+-----+
11 rows in set (0.00 sec)
```

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

```
mysql> SELECT d.nombre
-> FROM empleados e
-> JOIN departamentos d ON e.codigo_departamento = d.codigo
-> WHERE e.nif = '38382980M';
+-----+
| nombre |
+-----+
| Desarrollo |
+-----+
1 row in set (0.01 sec)
```

9. Devuelve el nombre del departamento donde trabaja el empleado Pepe Ruiz Santana. 10.

```
mysql> SELECT d.nombre
-> FROM empleados e
-> JOIN departamentos d ON e.codigo_departamento = d.codigo
-> WHERE e.nombre = 'Pepe' AND e.apellido1 = 'Ruiz' AND e.apellido2 = 'Santana';
+-----+
| nombre |
+-----+
| Recursos Humanos |
+-----+
1 row in set (0.00 sec)
```

10.Devuelve un listado con los datos de los empleados que trabajan en el departamento de I+D. Ordena el resultado alfabéticamente.

```
mysql> SELECT e.codigo, e.nif, e.nombre AS nombre_empleado, e.apellido1, e.apellido2
-> FROM empleados e
-> JOIN departamentos d ON e.codigo_departamento = d.codigo
-> WHERE d.nombre = 'I+D'
-> ORDER BY e.apellido1, e.apellido2, e.nombre;
+-----+-----+-----+-----+-----+
| codigo | nif      | nombre_empleado | apellido1 | apellido2 |
+-----+-----+-----+-----+-----+
| 10     | 46384486H | Diego           | Flores    | Salas      |
| 5      | 17087203C | Marcos          | Loyola    | Méndez     |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

11.Devuelve un listado con los datos de los empleados que trabajan en el departamento de Sistemas, Contabilidad o I+D. Ordena el resultado alfabéticamente.

```
mysql> SELECT e.codigo, e.nif, e.nombre AS nombre_empleado, e.apellido1, e.apellido2
-> FROM empleados e
-> JOIN departamentos d ON e.codigo_departamento = d.codigo
-> WHERE d.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
-> ORDER BY e.apellido1, e.apellido2, e.nombre;
+-----+-----+-----+-----+-----+
| codigo | nif      | nombre_empleado | apellido1 | apellido2 |
+-----+-----+-----+-----+-----+
| 10     | 46384486H | Diego           | Flores    | Salas      |
| 9      | 56399183D | Juan            | Gómez     | López      |
| 5      | 17087203C | Marcos          | Loyola    | Méndez     |
| 7      | 80576669X | Pilar           | Ruiz      | NULL       |
| 2      | Y5575632D | Adela           | Salas     | Díaz       |
| 4      | 77705545E | Adrián          | Suárez    | NULL       |
+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

12.Devuelve una lista con el nombre de los empleados que tienen los departamentos que no tienen un presupuesto entre 100000 y 200000 euros.

```
mysql> SELECT e.nombre AS nombre_empleado
-> FROM empleados e
-> JOIN departamentos d ON e.codigo_departamento = d.codigo
-> WHERE d.presupuesto < 100000 OR d.presupuesto > 200000;
+-----+
| nombre_empleado |
+-----+
| Adolfo          |
| Pepe            |
| Marcos          |
| Diego           |
+-----+
4 rows in set (0.00 sec)
```

13.Devuelve un listado con el nombre de los departamentos donde existe algún empleado cuyo segundo apellido sea NULL. Tenga en cuenta que no debe mostrar nombres de departamentos que estén repetidos

```
mysql> SELECT DISTINCT d.nombre AS nombre_departamento
-> FROM empleados e
-> JOIN departamentos d ON e.codigo_departamento = d.codigo
-> WHERE e.apellido2 IS NULL;
+-----+
| nombre_departamento |
+-----+
| Contabilidad         |
| Sistemas             |
+-----+
2 rows in set (0.00 sec)
```


14.Devuelve un listado con todos los empleados junto con los datos de los departamentos donde trabajan. Este listado también debe incluir los empleados que no tienen ningún departamento asociado.

```
mysql> SELECT e.codigo, e.nif, e.nombre AS nombre_empleado, e.apellido1, e.apellido2, d.codigo AS codigo_depa
rtamento, d.nombre AS nombre_departamento
-> FROM empleados e
-> LEFT JOIN departamentos d ON e.codigo_departamento = d.codigo;
```

| | codigo | nif | nombre_empleado | apellido1 | apellido2 | codigo_departamento | nombre_departamento |
|----|-----------|--------------|-----------------|-----------|-----------|---------------------|---------------------|
| 1 | 32481596F | Aarón | Rivero | Gómez | 1 | Desarrollo | |
| 2 | Y5575632D | Adela | Salas | Díaz | 2 | Sistemas | |
| 3 | R6970642B | Adolfo | Rubio | Flores | 3 | Recursos Humanos | |
| 4 | 77705545E | Adrián | Suárez | NULL | 4 | Contabilidad | |
| 5 | 17087203C | Marcos | Loyola | Méndez | 5 | I+D | |
| 6 | 38382980M | María | Santana | Moreno | 1 | Desarrollo | |
| 7 | 80576669X | Pilar | Ruiz | NULL | 2 | Sistemas | |
| 8 | 71651431Z | Pepe | Ruiz | Santana | 3 | Recursos Humanos | |
| 9 | 56399183D | Juan | Gómez | López | 2 | Sistemas | |
| 10 | 46384486H | Diego | Flores | Salas | 5 | I+D | |
| 11 | 67389283A | Marta | Herrera | Gil | 1 | Desarrollo | |
| 12 | 41234836R | Irene | Salas | Flores | NULL | NULL | |
| 13 | 82635162B | Juan Antonio | Sáez | Guerrero | NULL | NULL | |

```
13 rows in set (0.00 sec)
```

15.Devuelve un listado donde sólo aparezcan aquellos empleados que no tienen ningún departamento asociado.

```
mysql> SELECT e.codigo, e.nif, e.nombre AS nombre_empleado, e.apellido1, e.apellido2
-> FROM empleados e
-> LEFT JOIN departamentos d ON e.codigo_departamento = d.codigo
-> WHERE d.codigo IS NULL;
```

| | codigo | nif | nombre_empleado | apellido1 | apellido2 |
|----|-----------|--------------|-----------------|-----------|-----------|
| 12 | 41234836R | Irene | Salas | Flores | |
| 13 | 82635162B | Juan Antonio | Sáez | Guerrero | |

```
2 rows in set (0.00 sec)
```

16.Devuelve un listado donde sólo aparezcan aquellos departamentos que no tienen ningún empleado asociado.

```
mysql> SELECT d.codigo, d.nombre AS nombre_departamento
-> FROM departamentos d
-> LEFT JOIN empleados e ON d.codigo = e.codigo_departamento
-> WHERE e.codigo IS NULL;
```

| | codigo | nombre_departamento |
|---|------------|---------------------|
| 6 | Proyectos | |
| 7 | Publicidad | |

```
2 rows in set (0.00 sec)
```

17.Devuelve un listado con todos los empleados junto con los datos de los departamentos donde trabajan. El listado debe incluir los empleados que no tienen ningún departamento asociado y los departamentos que no tienen ningún empleado asociado. Ordene el listado alfabéticamente por el nombre del departamento.

```
mysql> SELECT e.codigo, e.nif, e.nombre AS nombre_empleado, e.apellido1, e.apellido2, d.codigo AS codigo_dep  
rtamento, d.nombre AS nombre_departamento  
-> FROM empleados e  
-> LEFT JOIN departamentos d ON e.codigo_departamento = d.codigo  
-> UNION  
-> SELECT NULL, NULL, NULL, NULL, NULL, d.codigo, d.nombre AS nombre_departamento  
-> FROM departamentos d  
-> LEFT JOIN empleados e ON d.codigo = e.codigo_departamento  
-> WHERE e.codigo IS NULL  
-> ORDER BY nombre_departamento;
```

| codigo | nif | nombre_empleado | apellido1 | apellido2 | codigo_departamento | nombre_departamento |
|--------|-----------|-----------------|-----------|-----------|---------------------|---------------------|
| 12 | 41234836R | Irene | Salas | Flores | NULL | NULL |
| 13 | 82635162B | Juan Antonio | Sáez | Guerrero | NULL | NULL |
| 4 | 77705545E | Adrián | Suárez | NULL | 4 | Contabilidad |
| 1 | 32481596F | Aarón | Rivero | Gómez | 1 | Desarrollo |
| 6 | 38382980M | María | Santana | Moreno | 1 | Desarrollo |
| 11 | 67389283A | Marta | Herrera | Gil | 1 | Desarrollo |
| 5 | 17087203C | Marcos | Loyola | Méndez | 5 | I+D |
| 10 | 46384486H | Diego | Flores | Salas | 5 | I+D |
| NULL | NULL | NULL | NULL | NULL | 6 | Proyectos |
| NULL | NULL | NULL | NULL | NULL | 7 | Publicidad |
| 3 | R6970642B | Adolfo | Rubio | Flores | 3 | Recursos Humanos |
| 8 | 71651431Z | Pepe | Ruiz | Santana | 3 | Recursos Humanos |
| 2 | Y5575632D | Adela | Salas | Díaz | 2 | Sistemas |
| 7 | 80576669X | Pilar | Ruiz | NULL | 2 | Sistemas |
| 9 | 56399183D | Juan | Gómez | López | 2 | Sistemas |

18.Devuelve un listado con los empleados que no tienen ningún departamento asociado y los departamentos que no tienen ningún empleado asociado. Ordene el listado alfabéticamente por el nombre del departamento.

```
mysql> SELECT e.codigo, e.nif, e.nombre AS nombre_empleado, e.apellido1, e.apellido2, d.codigo AS codigo_dep  
rtamento, d.nombre AS nombre_departamento  
-> FROM empleados e  
-> LEFT JOIN departamentos d ON e.codigo_departamento = d.codigo  
-> WHERE d.codigo IS NULL  
-> UNION  
-> SELECT NULL, NULL, NULL, NULL, NULL, d.codigo, d.nombre AS nombre_departamento  
-> FROM departamentos d  
-> LEFT JOIN empleados e ON d.codigo = e.codigo_departamento  
-> WHERE e.codigo IS NULL  
-> ORDER BY nombre_departamento;
```

| codigo | nif | nombre_empleado | apellido1 | apellido2 | codigo_departamento | nombre_departamento |
|--------|-----------|-----------------|-----------|-----------|---------------------|---------------------|
| 12 | 41234836R | Irene | Salas | Flores | NULL | NULL |
| 13 | 82635162B | Juan Antonio | Sáez | Guerrero | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | 6 | Proyectos |
| NULL | NULL | NULL | NULL | NULL | 7 | Publicidad |

4 rows in set (0.01 sec)

19.Cree una copia de seguridad del schema empleado

```
C:\Program Files\MySQL\MySQL Server 8.0\bin> mysqldump -u root -p empleado > c:/copia_empleado/copia_empleado.sql  
Enter password: ****
```

20.Restaurar la copia de seguridad a un schema nuevo llamado empleadodeparta

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p empleadodeparta < C:/copia_empleado/copia_empleado.sql  
Enter password: ****
```