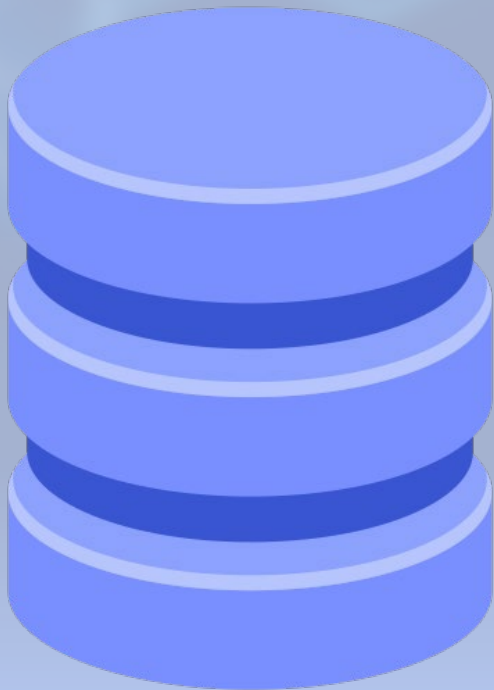


# **MYSQL: INNER JOIN**

---



---

**GESTIÓN DE BASES DE DATOS**  
**JUAN CARLOS NAVIDAD GARCÍA**

- a) Crear un usuario que se llame COCINERO.

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

- b) Asignarle permisos de consulta, actualización e inserción sobre las tablas productos, clientes, empleados, líneas pedido y pedidos.

```
mysql> GRANT SELECT,UPDATE,INSERT ON pizzeria.productos TO cocinero@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT SELECT,UPDATE,INSERT ON pizzeria.clientes TO cocinero@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT SELECT,UPDATE,INSERT ON pizzeria.empleados TO cocinero@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT,UPDATE,INSERT ON pizzeria.lineaspedido TO cocinero@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT,UPDATE,INSERT ON pizzeria.pedidos TO cocinero@localhost;
Query OK, 0 rows affected (0.01 sec)
```

- c) Conéctate con ese usuario y comprueba los permisos que tienes. Sigue conectado con él para resolver las siguientes cuestiones.

```
mysql> SHOW GRANTS FOR cocinero@localhost;
+-----+
| Grants for cocinero@localhost |
+-----+
| GRANT USAGE ON *.* TO 'cocinero'@'localhost' |
| GRANT SELECT, INSERT, UPDATE ON 'pizzeria'.'clientes' TO 'cocinero'@'localhost' |
| GRANT SELECT, INSERT, UPDATE ON 'pizzeria'.'empleados' TO 'cocinero'@'localhost' |
| GRANT SELECT, INSERT, UPDATE ON 'pizzeria'.'lineaspedido' TO 'cocinero'@'localhost' |
| GRANT SELECT, INSERT, UPDATE ON 'pizzeria'.'pedidos' TO 'cocinero'@'localhost' |
| GRANT SELECT, INSERT, UPDATE ON 'pizzeria'.'productos' TO 'cocinero'@'localhost' |
+-----+
6 rows in set (0.00 sec)
```

- d) Obtener el total de un pedido de vuestra elección.

```
SELECT SUM(productos.precio)*lineaspedido.unidades FROM lineaspedido INNER
JOIN productos ON lineaspedido.idproducto=productos.idprod WHERE idped=4;
```

```
+-----+
| SUM(productos.precio)*lineaspedido.unidades |
+-----+
|                                     24.55 |
+-----+
1 row in set (0.00 sec)
```

- e) Crear una función para obtener el total de un pedido dado. (tendrás que revisar que el usuario COCINERO tiene permisos para crear funciones).

```
CREATE FUNCTION act_e(num_ped INT) RETURNS DECIMAL(8,2) DETERMINISTIC
BEGIN
DECLARE total DECIMAL(8,2);
SELECT SUM(productos.precio)*lineaspedido.unidades INTO total FROM
lineaspedido INNER JOIN productos ON lineaspedido.idproducto=productos.idprod
WHERE idped=num_ped;
RETURN total;
END //
```

```
mysql> SELECT act_e(4);//
+-----+
| act_e(4) |
+-----+
|    24.55 |
+-----+
1 row in set (0.00 sec)
```

- f) Obtener la suma total de los pedidos realizados por clientes menores de edad.

```
SELECT SUM(precio*unidades) FROM lineaspedido INNER JOIN productos ON
lineaspedido.idproducto=productos.idprod INNER JOIN pedidos ON
lineaspedido.idped=pedidos.idped INNER JOIN clientes ON
pedidos.idcli=clientes.idcli WHERE TIMESTAMPDIFF(YEAR,fnac,CURDATE())<18;
```

```
+-----+
| SUM(precio*unidades) |
+-----+
|           138.25 |
+-----+
1 row in set (0.00 sec)
```

- g) Obtener la suma total de los pedidos gestionados por el empleado R04.

```
SELECT SUM(precio*unidades) FROM lineaspedido INNER JOIN productos ON
lineaspedido.idproducto=productos.idprod INNER JOIN pedidos ON
lineaspedido.idped=pedidos.idped INNER JOIN empleados ON
pedidos.idemp=empleados.idemp WHERE empleados.idemp="R04";
```

```
+-----+
| SUM(precio*unidades) |
+-----+
|           35.54 |
+-----+
1 row in set (0.00 sec)
```

- h) Obtener un listado de los empleados junto con la descripción de su categoría. Si hubiera dificultades, reasigna los permisos adecuados para poder implementar esta consulta.

```
SELECT empleados.nombre, categorias.descripcion FROM empleados INNER JOIN
categorias ON empleados.categoria=categorias.idcat;
```

nombre	descripcion
Pepe Rodriguez	Cocinero
Jordi Cruz	Cocinero
Belen de la Torre	Cocinero
Carlos Sanchez	Repartidor
Miriam Valero	Repartidor
Susana Alameda	Repartidor
Juan Lopez	Repartidor

7 rows in set (0.00 sec)

- i) Listado de empleados con el sueldo incrementado en 1,5€ por cada pedido que haya gestionado.

```
SELECT empleados.nombre, sueldo+(1.5*COUNT(pedidos.idemp)) AS sueldo_ampliado
FROM empleados INNER JOIN pedidos ON empleados.idemp=pedidos.idemp GROUP BY
nombre;
```

nombre	sueldo_ampliado
Pepe Rodriguez	1503.0
Carlos Sanchez	2504.5
Miriam Valero	4509.0

3 rows in set (0.00 sec)

- j) Crear un evento que se ejecute cada hora para ir calculando la caja e ir almacenando el resultado en una tabla con dos campos, FechaHora y TotalCaja.

```
CREATE PROCEDURE caja()  
BEGIN  
DECLARE xcaja DECIMAL(8,2);  
SELECT SUM(productos.precio)*lineaspedido.unidades INTO xcaja  
FROM lineaspedido  
INNER JOIN productos ON lineaspedido.idproducto=productos.idprod INNER JOIN  
pedidos ON lineaspedido.idped=pedidos.idped  
WHERE pedidos.fecha=CURDATE();  
CREATE TABLE IF NOT EXISTS caja (FechaHora DATETIME PRIMARY KEY, TotalCaja  
DECIMAL(8,2));  
INSERT INTO caja VALUES(NOW(), xcaja);  
END //
```

```
CREATE EVENT IF NOT EXISTS caja  
ON SCHEDULE EVERY 1 HOUR  
ON COMPLETION PRESERVE  
DO  
CALL caja();
```

```
mysql> SELECT * FROM caja; //  
+-----+-----+  
| FechaHora          | TotalCaja |  
+-----+-----+  
| 2023-04-14 12:06:43 |    18.50 |  
+-----+-----+  
1 row in set (0.00 sec)
```