

## Actividad 9. Inner Join.

### Paso 2: Lectura del fichero sql.

1. Entra en SQLite3 a través del siguiente comando:

sqlite3 tarea8.db

```
sqlite> .open tarea8.db
sqlite> █
```

2. Haciendo un .read del fichero sql, de nombre librería-dump.sql, realiza la creación e inserción de información de la BBDD.

```
sqlite> .read librería-dump.sql
sqlite> CREATE TABLE CLIENTE (
(x1...>     ID_CLIENTE INTEGER PRIMARY KEY
(x1...>     NOMBRE TEXT,
(x1...>     APELLIDO TEXT,
(x1...>     DIRECCION TEXT
(x1...> );
sqlite>
sqlite> INSERT INTO CLIENTE (
(x1...>     NOMBRE,
(x1...>     APELLIDO,
(x1...>     DIRECCION
(x1...> ) VALUES (
(x1...>     'Juan',
(x1...>     'González',
(x1...>     'Calle Principal 123'
(x1...> ),
...> (
(x1...>     'María',
(x1...>     'Martínez',
(x1...>     'Avenida Central 456'
(x1...> ),
...> (
(x1...>     'Pedro',
```

### Paso 3: Realizar consultas de datos.

Realiza las siguientes consultas usando inner join, y muestra el resultado obtenido:

1. Listar todos los libros vendidos junto con la cantidad y el precio total de cada venta.

```
SELECT LIBRO.TITULO, VENTA.CANTIDAD, (VENTA.CANTIDAD * LIBRO.PRECIO) AS
PRECIO_TOTAL FROM VENTA JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO;
```

```
sqlite> SELECT LIBRO.TITULO, VENTA.CANTIDAD, (VENTA.CANTIDAD * LIBRO.PRECIO) AS PRECIO_TOTAL FROM VENTA JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO;
```

TITULO	CANTIDAD	PRECIO_TOTAL
El aleph	2	31.0
Harry Potter y la piedra filosofal	1	45.0
Antología poetica	3	117.0
Cervantes y el quijote	1	35.4
Martin Fierro	2	45.8
Harry Potter y la camara secreta	1	46.0

2. Encontrar el nombre completo de los clientes que han comprado libros.

```
SELECT cliente.NOMBRE || ' ' || CLIENTE.APELLIDO AS NOMBRE_COMPLETO from VENTA
JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO JOIN CLIENTE ON VENTA.ID_CLIENTE =
CLIENTE.ID_CLIENTE;
```

```
sqlite> SELECT cliente.NOMBRE || ' ' || cliente.APELLIDO AS NOMBRE_COMPLETO from VENTA JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO JOIN cliente ON VENTA.ID_CLIENTE = cliente.ID_CLIENTE;
```

NOMBRE_COMPLETO
Juan González
Maria Martínez
Pedro Sánchez
Juan González
Maria Martínez
Pedro Sánchez

3. Calcular el total gastado por cada cliente en todas sus compras.

```
SELECT cliente.ID_CLIENTE, cliente.NOMBRE || ' ' || cliente.APELLIDO AS NOMBRE_COMPLETO, SUM(LIBRO.PRECIO * VENTA.CANTIDAD) AS TOTAL_GASTADO
FROM cliente JOIN VENTA ON cliente.ID_CLIENTE = VENTA.ID_CLIENTE
JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO
GROUP BY cliente.ID_CLIENTE, NOMBRE_COMPLETO;
```

```
sqlite> SELECT cliente.ID_CLIENTE, cliente.NOMBRE || ' ' || cliente.APELLIDO AS NOMBRE_COMPLETO, SUM(LIBRO.PRECIO * VENTA.CANTIDAD) AS TOTAL_GASTADO
...> FROM
...> cliente
...> JOIN
...> VENTA ON cliente.ID_CLIENTE = VENTA.ID_CLIENTE
...> JOIN
...> LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO
...> GROUP BY
...> cliente.ID_CLIENTE, NOMBRE_COMPLETO;
```

ID_CLIENTE	NOMBRE_COMPLETO	TOTAL_GASTADO
1	Juan González	66.4
2	Maria Martínez	90.8
3	Pedro Sánchez	163.0

4. Encontrar los títulos de los libros vendidos en una fecha específica.

```
SELECT DISTINCT LIBRO.TITULO
FROM VENTA
JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO
WHERE VENTA.FECHA_VENTA = '2024-02-10';
```

```
sqlite> SELECT DISTINCT LIBRO.TITULO
...> FROM VENTA
...> JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO
...> WHERE VENTA.FECHA_VENTA = '2024-02-10';
```

TITULO
Harry Potter y la piedra filosofal

5. Listar todos los clientes que han comprado el libro "El aleph".

```
SELECT DISTINCT cliente.ID_CLIENTE, cliente.NOMBRE, cliente.APELLIDO
FROM cliente
JOIN VENTA ON cliente.ID_CLIENTE = VENTA.ID_CLIENTE
JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO
WHERE LIBRO.TITULO = 'El aleph';
```

```
sqlite> SELECT DISTINCT cliente.ID_CLIENTE, cliente.NOMBRE, cliente.APELLIDO
...> FROM cliente
...> JOIN VENTA ON cliente.ID_CLIENTE = VENTA.ID_CLIENTE
...> JOIN LIBRO ON VENTA.CODIGO_LIBRO = LIBRO.CODIGO
...> WHERE LIBRO.TITULO = 'El aleph';
```

ID_CLIENTE	NOMBRE	APELLIDO
1	Juan	González

6. Encontrar el autor que ha vendido más libros.

```
SELECT LIBRO.AUTOR, COUNT(*) AS NUMERO_DE_LIBROS_VENDIDOS
FROM LIBRO
JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO
GROUP BY LIBRO.AUTOR ORDER BY NUMERO_DE_LIBROS_VENDIDOS DESC
LIMIT 1;
```

```
sqlite> SELECT LIBRO.AUTOR, COUNT(*) AS NUMERO_DE_LIBROS_VENDIDOS
...> FROM LIBRO
...> JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO
...> GROUP BY LIBRO.AUTOR
...> ORDER BY NUMERO_DE_LIBROS_VENDIDOS DESC
...> LIMIT 1;
```

AUTOR	NUMERO_DE_LIBROS_VENDIDOS
J.K. Rowling	2

7. Obtener el título y el precio de los libros vendidos a un cliente específico.

```
SELECT LIBRO.TITULO, LIBRO.PRECIO
FROM LIBRO JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO
WHERE VENTA.ID_CLIENTE = 1;
```

```
sqlite> SELECT LIBRO.TITULO, LIBRO.PRECIO
...> FROM LIBRO JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO
...> WHERE VENTA.ID_CLIENTE = 1;
```

TITULO	PRECIO
El aleph	15.5
Cervantes y el quijote	35.4

8. Calcular el precio promedio de los libros vendidos.

```
SELECT AVG(LIBRO.PRECIO) AS PRECIO_PROMEDIO
FROM LIBRO JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO;
```

```
sqlite> SELECT AVG(LIBRO.PRECIO) AS PRECIO_PROMEDIO
...> FROM LIBRO JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO;
```

PRECIO_PROMEDIO
33.96666666666667

9. Listar todos los libros que se han vendido más de una vez.

```
SELECT LIBRO.TITULO, COUNT(*) AS NUMERO_DE_VENTAS
FROM LIBRO
JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO
GROUP BY LIBRO.TITULO
HAVING COUNT(*) > 1;
```

10. Encontrar los libros más caros comprados por María Martínez.

```
SELECT LIBRO.TITULO, LIBRO.PRECIO  
FROM LIBRO  
JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO  
JOIN CLIENTE ON VENTA.ID_CLIENTE = CLIENTE.ID_CLIENTE  
WHERE CLIENTE.NOMBRE = 'María' AND CLIENTE.APELLIDO = 'Martínez'  
ORDER BY LIBRO.PRECIO DESC ;
```

```
sqlite> SELECT LIBRO.TITULO, LIBRO.PRECIO  
...> FROM LIBRO  
...> JOIN VENTA ON LIBRO.CODIGO = VENTA.CODIGO_LIBRO  
...> JOIN CLIENTE ON VENTA.ID_CLIENTE = CLIENTE.ID_CLIENTE  
...> WHERE CLIENTE.NOMBRE = 'María' AND CLIENTE.APELLIDO = 'Martínez'  
...> ORDER BY LIBRO.PRECIO DESC ;
```

TITULO	PRECIO
Harry Potter y la piedra filosofal	45.0
Martin Fierro	22.9