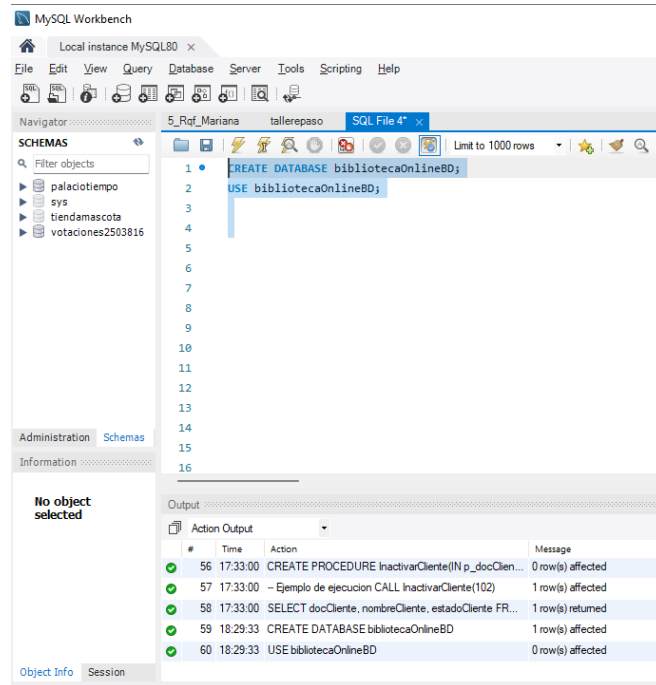


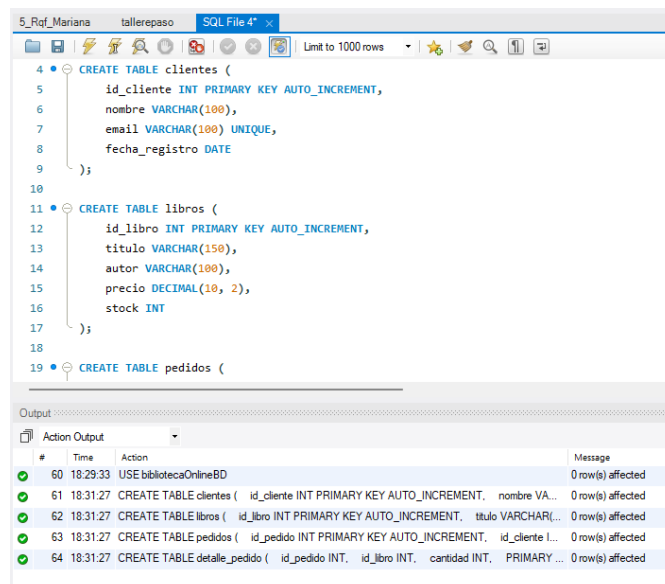
## Taller de Repaso 2

Mariana Álvarez Carvajal

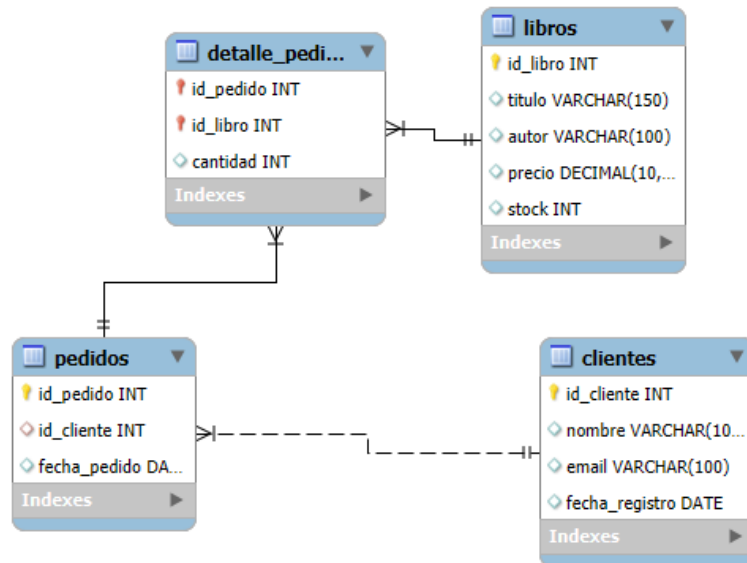
- 1- Crear la base de datos que se llame biblioteca OnlineBD



- 2- Crear las tablas clientes, libros, y pedidos. Aplicar restricciones adecuadas (claves primarias, foráneas, unicidad, etc.).



- 3- Genera el modelo físico desde gestor de base de datos utilizado



#### 4- Insertar datos en las tablas de clientes, libros, y pedidos.

```

56 • select*from clientes;
57 • INSERT INTO clientes (nombre, email, fecha_registro) VALUES('Laura Gómez', 'laura@example.com', '2024-01-10'),('Carlo
58 ('Ana Martínez', 'ana@example.com', '2024-03-05');
59
60 • INSERT INTO libros (titulo, autor, precio, stock) VALUES ('Cien Años de Soledad', 'Gabriel García Márquez', 25.00, 10
61 ('El Principito', 'Antoine de Saint-Exupéry', 15.00, 20),('Don Quijote de la Mancha', 'Miguel de Cervantes', 30.00, 5
62 ('Rayuela', 'Julio Cortázar', 22.00, 8),('Ficciones', 'Jorge Luis Borges', 18.00, 12);
63
64 • INSERT INTO pedidos (id_cliente, fecha_pedido) VALUES(1, '2025-04-01'),(2, '2025-04-02'),(3, '2025-04-03');
65
66
67 • desc detalle_pedido;
68 -- Pedido 1: Laura compró 2 de "Cien Años de Soledad" y 1 de "El Principito"
69 INSERT INTO detalle_pedido VALUES(1, 1, 2),(1, 2, 1);
70 -- Pedido 2: Carlos compró 1 de "Don Quijote" y 3 de "Rayuela"
  
```

Output:

#	Time	Action	Message
85	18:53:22	select from clientes LIMIT 0, 1000	3 row(s) returned
86	18:53:46	INSERT INTO pedidos (id_cliente, fecha_pedido) VALUES(1, '2025-04-01'),(2, '2025-04-02'),(3, '2025-04-03');	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
87	18:53:55	INSERT INTO detalle_pedido VALUES(1, 1, 2),(1, 2, 1);	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
88	18:53:55	INSERT INTO detalle_pedido VALUES (2, 3, 1), (2, 4, 3)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
89	18:53:55	INSERT INTO detalle_pedido VALUES (3, 5, 2)	1 row(s) affected

#### 5- Actualizar el stock de los libros una vez que se realice un pedido

```

76 DELIMITER //
77 • CREATE TRIGGER actualizar_stock
78 AFTER INSERT ON detalle_pedido
79 FOR EACH ROW BEGIN
80 UPDATE libros
81 SET stock = stock - NEW.cantidad
82 WHERE id_libro = NEW.id_libro;
83 END//
84 DELIMITER ;
85 • select*from libros;
86 • INSERT INTO detalle_pedido (id_pedido, id_libro, cantidad)
87 VALUES (2, 2, 3); /*Antes habian 20 de El Principito0, ahora hav 17*/
  
```

Result Grid:

id_libro	titulo	autor	precio	stock
1	Cien Años de Soledad	Gabriel García Márquez	25.00	10
2	El Principito	Antoine de Saint-Exupéry	15.00	17
3	Don Quijote de la Mancha	Miguel de Cervantes	30.00	5
4	Rayuela	Julio Cortázar	22.00	8
5	Ficciones	Jorge Luis Borges	18.00	12
6	...	...	...	...

- 6- listar los pedidos hechos por un cliente específico y obtener detalles de los libros comprados.

```
90 /*listar los pedidos hechos por un cliente específico y obtener detalles de los libros comprados.*/
91 • SELECT clientes.nombre,pedidos.id_pedido,pedidos.fecha_pedido,libros.titulo,libros.precio,detalle_pedido.cantidad
92 FROM clientes
93 JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente
94 JOIN detalle_pedido ON pedidos.id_pedido = detalle_pedido.id_pedido
95 JOIN libros ON detalle_pedido.id_libro = libros.id_libro
96 WHERE clientes.id_cliente = 1;
```

Result Grid

	nombre	id_pedido	fecha_pedido	titulo	precio	cantidad
▶	Laura Gómez	1	2025-04-01	Cien Años de Soledad	25.00	2
	Laura Gómez	1	2025-04-01	El Principito	15.00	1

Result 11 x

Output

#	Time	Action	Message
95	19:05:43	SELECT nombre, id_pedido, fecha_pedido, titulo, precio, cantidad FROM clientes...	Error Code: 1052. Column 'id_pedido' in field list is ambiguous
96	19:06:48	SELECT clientes.nombre, pedidos.id_pedido, pedidos.fecha_pedido, libros.titulo, libr...	2 row(s) returned
97	19:07:09	SELECT clientes.nombre, id_pedido, pedidos.fecha_pedido, libros.titulo, libros.precio...	Error Code: 1052. Column 'id_pedido' in field list is ambiguous
98	19:09:48	SELECT clientes.nombre,pedidos.id_pedido,pedidos.fecha_pedido,libros.titulo,libros.precio,detalle...	2 row(s) returned

- 7- Consultar el cliente que ha realizado el mayor número de pedidos.

```
98
99 /*Consultar el cliente que ha realizado el mayor número de pedidos.*/
100 • SELECT clientes.id_cliente,nombre,COUNT(pedidos.id_pedido) AS total_pedidos
101 FROM clientes JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente
102 GROUP BY clientes.id_cliente, clientes.nombre ORDER BY total_pedidos DESC LIMIT 1; /*EI Iimi!
103
104
105
106
```

Result Grid

	id_cliente	nombre	total_pedidos
▶	1	Laura Gómez	2

Result 16 x

Output

#	Time	Action	Message
103	19:18:24	SELECT COUNT(*) FROM pedidos WHERE id_cliente = 1 LIMIT 0, 1000	1 row(s) returned
104	19:18:29	SELECT COUNT(*) FROM pedidos WHERE id_cliente = 3 LIMIT 0, 1000	1 row(s) returned
105	19:19:25	INSERT INTO pedidos (id_cliente, fecha_pedido) VALUES(1, '2025-04-01')	1 row(s) affected
106	19:19:30	SELECT clientes.id_cliente,nombre,COUNT(pedidos.id_pedido) AS total_pedidos FROM clientes ...	1 row(s) returned

- 8- Crear un procedimiento almacenado que permita registrar un nuevo pedido, actualizando la tabla de pedidos y reduciendo el stock del libro correspondiente.

```

107 DELIMITER //
108 CREATE PROCEDURE registrar_pedido (IN p_id_pedido INT, IN p_id_cliente INT, IN p_fecha DATE, IN p_id_libro INT,
109 IN p_cantidad INT)
110 BEGIN
111     -- Insertar el pedido con el ID y fecha recibidos
112     INSERT INTO pedidos (id_pedido, id_cliente, fecha_pedido)
113     VALUES (p_id_pedido, p_id_cliente, p_fecha);
114
115     -- Insertar el detalle del pedido
116     INSERT INTO detalle_pedido (id_pedido, id_libro, cantidad)
117     VALUES (p_id_pedido, p_id_libro, p_cantidad);
118
119     -- Actualizar el stock del libro
120     UPDATE libros
121     SET stock = stock - p_cantidad
122     WHERE id_libro = p_id_libro;
123 END //

```

Output

#	Time	Action	Message
105	19:19:25	INSERT INTO pedidos (id_cliente, fecha_pedido) VALUES(1, '2025-04-01')	1 row(s) affected
106	19:19:30	SELECT clientes.id_cliente, nombre, COUNT(pedidos.id_pedido) AS total_pedidos FROM clientes ...	1 row(s) returned
107	19:32:31	CREATE PROCEDURE registrar_pedido (IN p_id_pedido INT, IN p_id_cliente INT, IN p_fecha DA...	0 row(s) affected
108	19:32:43	CALL registrar_pedido(10, 2, '2025-04-05', 3, 2)	1 row(s) affected

## 9- Selecciona los libros cuyo precio sea mayor a \$20.00

```

125 CALL registrar_pedido(10, 2, '2025-04-05', 3, 2);
126
127
128 /*Selecciona los libros cuyo precio sea mayor a $20.00 */
129 SELECT * FROM libros WHERE precio > 20.00;
130
131
132
133

```

Result Grid

id_libro	titulo	autor	precio	stock
1	Cien Años de Soledad	Gabriel García Márquez	25.00	10
3	Don Quijote de la Mancha	Miguel de Cervantes	30.00	1
4	Rayuela	Julio Cortázar	22.00	8
NULL	NULL	NULL	NULL	NULL

libros 17 x

Output

#	Time	Action	Message
106	19:19:30	SELECT clientes.id_cliente, nombre, COUNT(pedidos.id_pedido) AS total_pedidos FROM clientes ...	1 row(s) returned
107	19:32:31	CREATE PROCEDURE registrar_pedido (IN p_id_pedido INT, IN p_id_cliente INT, IN p_fecha DA...	0 row(s) affected
108	19:32:43	CALL registrar_pedido(10, 2, '2025-04-05', 3, 2)	1 row(s) affected
109	19:35:15	SELECT * FROM libros WHERE precio > 20.00 LIMIT 0, 1000	3 row(s) returned

## 10- Selecciona los pedidos realizados después del 1 de octubre de 2024.

```

129 SELECT * FROM libros WHERE precio > 20.00;
130
131
132 /*Selecciona los pedidos realizados después del 1 de octubre de 2024.*/
133 SELECT * FROM pedidos WHERE fecha_pedido > '2024-10-01';
134
135
136
137

```

Result Grid

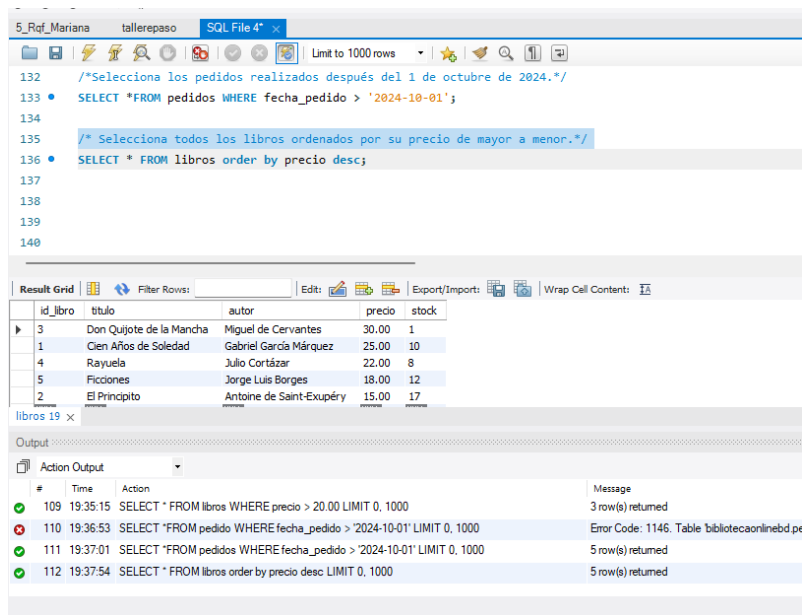
id_pedido	id_cliente	fecha_pedido
1	1	2025-04-01
2	2	2025-04-02
3	3	2025-04-03
4	1	2025-04-01
10	2	2025-04-05

pedidos 18 x

Output

#	Time	Action	Message
108	19:32:43	CALL registrar_pedido(10, 2, '2025-04-05', 3, 2)	1 row(s) affected
109	19:35:15	SELECT * FROM libros WHERE precio > 20.00 LIMIT 0, 1000	3 row(s) returned
110	19:36:53	SELECT * FROM pedidos WHERE fecha_pedido > '2024-10-01' LIMIT 0, 1000	Error Code: 1146. Table 'bibli...
111	19:37:01	SELECT * FROM pedidos WHERE fecha_pedido > '2024-10-01' LIMIT 0, 1000	5 row(s) returned

11-Selecciona todos los libros ordenados por su precio de mayor a menor.



SQL File 4

```
132 /*Selecciona los pedidos realizados después del 1 de octubre de 2024.*/  
133 • SELECT *FROM pedidos WHERE fecha_pedido > '2024-10-01';  
134  
135 /* Selecciona todos los libros ordenados por su precio de mayor a menor.*/  
136 • SELECT * FROM libros order by precio desc;  
137  
138  
139  
140
```

Result Grid

	id_libro	título	autor	precio	stock
▶	3	Don Quijote de la Mancha	Miguel de Cervantes	30.00	1
1	1	Cien Años de Soledad	Gabriel García Márquez	25.00	10
4	4	Rayuela	Julio Cortázar	22.00	8
5	5	Ficciones	Jorge Luis Borges	18.00	12
2	2	El Principito	Antoine de Saint-Exupéry	15.00	17

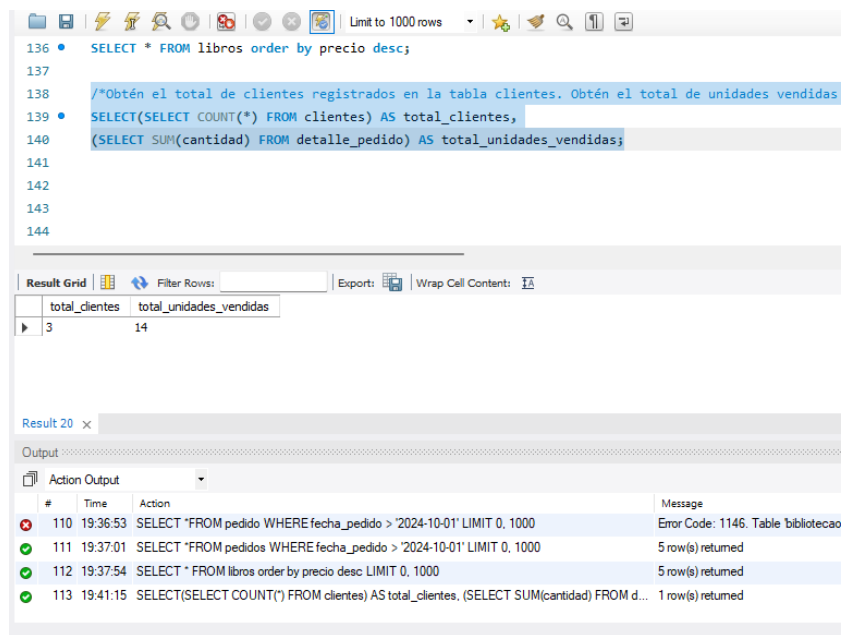
libros 19 x

Output

Action Output

#	Time	Action	Message
✓	109 19:35:15	SELECT * FROM libros WHERE precio > 20.00 LIMIT 0, 1000	3 row(s) returned
✗	110 19:36:53	SELECT *FROM pedido WHERE fecha_pedido > '2024-10-01' LIMIT 0, 1000	Error Code: 1146. Table 'bibliotecaonlinebd.pe
✓	111 19:37:01	SELECT *FROM pedidos WHERE fecha_pedido > '2024-10-01' LIMIT 0, 1000	5 row(s) returned
✓	112 19:37:54	SELECT * FROM libros order by precio desc LIMIT 0, 1000	5 row(s) returned

12-Obtén el total de clientes registrados en la tabla clientes. Obtén el total de unidades vendidas en la tabla pedidos.



SQL File 4

```
136 • SELECT * FROM libros order by precio desc;  
137  
138 /*Obtén el total de clientes registrados en la tabla clientes. Obtén el total de unidades vendidas  
139 • SELECT(SELECT COUNT(*) FROM clientes) AS total_clientes,  
140 (SELECT SUM(cantidad) FROM detalle_pedido) AS total_unidades_vendidas;  
141  
142  
143  
144
```

Result Grid

	total_clientes	total_unidades_vendidas
▶	3	14

Result 20 x

Output

Action Output

#	Time	Action	Message
✗	110 19:36:53	SELECT *FROM pedido WHERE fecha_pedido > '2024-10-01' LIMIT 0, 1000	Error Code: 1146. Table 'biblioteca
✓	111 19:37:01	SELECT *FROM pedidos WHERE fecha_pedido > '2024-10-01' LIMIT 0, 1000	5 row(s) returned
✓	112 19:37:54	SELECT * FROM libros order by precio desc LIMIT 0, 1000	5 row(s) returned
✓	113 19:41:15	SELECT(SELECT COUNT(*) FROM clientes) AS total_clientes, (SELECT SUM(cantidad) FROM d...	1 row(s) returned

13-Muestra el número de pedidos realizados por cada cliente.

Limit to 1000 rows

```

141
142      /*Muestra el número de pedidos realizados por cada cliente.*/
143 •   SELECT clientes.id_cliente, clientes.nombre, COUNT(pedidos.id_pedido) AS total_pedidos
144     FROM clientes LEFT JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente
145     GROUP BY clientes.id_cliente, clientes.nombre;
146
147
148
149

```

Result Grid

	id_cliente	nombre	total_pedidos
▶	1	Laura Gómez	2
	2	Carlos Ruiz	2
	3	Ana Martínez	1

Result 21 x

Output

Action Output

#	Time	Action	Message
✓	111 19:37:01	SELECT * FROM pedidos WHERE fecha_pedido > '2024-10-01' LIMIT 0, 1000	5 row(s) returned
✓	112 19:37:54	SELECT * FROM libros order by precio desc LIMIT 0, 1000	5 row(s) returned
✓	113 19:41:15	SELECT(SELECT COUNT(*) FROM clientes) AS total_clientes, (SELECT SUM(cantidad) FROM d...	1 row(s) returned
✓	114 19:42:44	SELECT clientes.id_cliente, clientes.nombre, COUNT(pedidos.id_pedido) AS total_pedidos FROM ...	3 row(s) returned

14-Muestra el nombre del cliente, el título del libro y la cantidad de cada pedido.

```

146
147      /*Muestra el nombre del cliente, el título del libro y la cantidad de cada pedido.*/
148 •   SELECT clientes.nombre AS nombre_cliente, libros.titulo AS titulo_libro, detalle_pedido.cantidad
149     FROM clientes
150     JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente
151     JOIN detalle_pedido ON pedidos.id_pedido = detalle_pedido.id_pedido
152     JOIN libros ON detalle_pedido.id_libro = libros.id_libro;
153

```

Result Grid

	nombre_cliente	titulo_libro	cantidad
▶	Laura Gómez	Cien Años de Soledad	2
	Laura Gómez	El Principito	1
	Carlos Ruiz	El Principito	3
	Carlos Ruiz	Don Quijote de la Mancha	1
	Carlos Ruiz	Rayuela	3

Result 22 x

Output

Action Output

#	Time	Action	Message
✓	112 19:37:54	SELECT * FROM libros order by precio desc LIMIT 0, 1000	5 row(s) returned
✓	113 19:41:15	SELECT(SELECT COUNT(*) FROM clientes) AS total_clientes, (SELECT SUM(cantidad) FROM d...	1 row(s) returned
✓	114 19:42:44	SELECT clientes.id_cliente, clientes.nombre, COUNT(pedidos.id_pedido) AS total_pedidos FROM ...	3 row(s) returned
✓	115 19:44:51	SELECT clientes.nombre AS nombre_cliente, libros.titulo AS titulo_libro, detalle_pedido.cantidad F...	7 row(s) returned

15-Muestra los títulos de los libros y la cantidad total vendida de cada uno.

Limit to 1000 rows

```

155 •   SELECT libros.titulo, SUM(detalle_pedido.cantidad) AS total_vendido
156     FROM detalle_pedido
157     JOIN libros ON detalle_pedido.id_libro = libros.id_libro
158     GROUP BY libros.titulo;
159
160
161
162
163

```

Result Grid

	titulo	total_vendido
▶	Cien Años de Soledad	2
	El Principito	4
	Don Quijote de la Mancha	3
	Rayuela	3
	Ficciones	2

Result 23 x

Output

Action Output

#	Time	Action	Message
✓	113 19:41:15	SELECT(SELECT COUNT(*) FROM clientes) AS total_clientes, (SELECT SUM(cantidad) FROM d...	1 row(s) returned
✓	114 19:42:44	SELECT clientes.id_cliente, clientes.nombre, COUNT(pedidos.id_pedido) AS total_pedidos FROM ...	3 row(s) returned
✓	115 19:44:51	SELECT clientes.nombre AS nombre_cliente, libros.titulo AS titulo_libro, detalle_pedido.cantidad F...	7 row(s) returned
✓	116 19:48:14	SELECT libros.titulo, SUM(detalle_pedido.cantidad) AS total_vendido FROM detalle_pedido JOIN l...	5 row(s) returned

