

## Pruebas de la implementación de la librería:

- Mostrar datos guardados sobre los todos los libros de la biblioteca.

The screenshot shows the MySQL Workbench interface. The top tab bar has tabs for Create, Insert, SQL File 3\*, SQL File 4, SQL File 5, SQL File 6, and SQL File 7. The SQL File 3\* tab is active, displaying the following SQL code:

```
1 #Mostrar datos guardados sobre los miembros de la biblioteca y todos los libros que este almacene
2
3 • SELECT * FROM libro;
```

Below the code is a Result Grid table with columns: ISBN, título, numero\_de\_páginas, área, estante, categoría.id, tipo\_casa\_o\_sala, and autor.id. The table contains 10 rows of Harry Potter book data. At the bottom of the grid, it says "libro 2 x".

On the right side of the interface, there is an Output pane titled "Action Output" which lists the following log entries:

#	Time	Action	Message
206	01:34:28	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro.ISBN', 'condicion', 'estado_en_uso') VALUES (6, '978847888...', 1 row(s) affected)	1 row(s) affected
207	01:34:28	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro.ISBN', 'condicion', 'estado_en_uso') VALUES (7, '978849838...', 1 row(s) affected)	1 row(s) affected
208	01:34:29	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro.ISBN', 'condicion', 'estado_en_uso') VALUES (8, '978849838...', 1 row(s) affected)	1 row(s) affected
209	01:34:29	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro.ISBN', 'condicion', 'estado_en_uso') VALUES (9, '978846634...', 1 row(s) affected)	1 row(s) affected
210	01:34:29	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro.ISBN', 'condicion', 'estado_en_uso') VALUES (10, '97884010...', 1 row(s) affected)	1 row(s) affected
211	01:34:29	COMMIT	0 row(s) affected
212	01:34:29	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected
213	01:34:29	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected
214	01:34:29	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected
215	01:36:08	SELECT * FROM empleado LIMIT 0, 1000	10 row(s) returned
216	01:36:48	SELECT * FROM libro LIMIT 0, 1000	10 row(s) returned

- Mostrar Los autores de la literatura para un área en particular : Juvenil

Screenshot of MySQL Workbench showing the execution of a SQL query and its results.

```

1 •  SELECT ISBN,titulo,categoría.nombre,autor.name FROM libro INNER JOIN categoría
2   ON categoría.id=libro.categoría_id INNER JOIN autor
3   ON autor.id=libro.autor_id
4   WHERE categoría.nombre='Juvenil'
    
```

**Result Grid**

ISBN	título	nombre	name
1408855917	Harry Potter y el Prisionero de Azkaban	Juvenil	J. K.
1408855925	Harry Potter y el Cáliz de Fuego	Juvenil	J. K.
8478884459	Harry Potter y la Piedra Filosofal	Juvenil	J. K.
8478884955	Harry Potter y la Cámara Secreta	Juvenil	J. K.
8478887423	Harry Potter y la Orden de Felix	Juvenil	J. K.
9788478889907	HArry Potter y el Príncipe Mestizo	Juvenil	J. K.
9788498381405	Harry Potter y las Reliquias de la Muerte	Juvenil	J. K.
9788498387544	Harry Potter y el Legado Maldito	Juvenil	J. K.

**Output**

#	Time	Action	Message
206	01:58:44	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro_ISBN', 'condicion', 'estado_en_uso') VALUES (6, '97884788...', 1 row(s) affected)	
207	01:58:44	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro_ISBN', 'condicion', 'estado_en_uso') VALUES (7, '97884983...', 1 row(s) affected)	
208	01:58:44	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro_ISBN', 'condicion', 'estado_en_uso') VALUES (8, '97884983...', 1 row(s) affected)	
209	01:58:44	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro_ISBN', 'condicion', 'estado_en_uso') VALUES (9, '97884663...', 1 row(s) affected)	
210	01:58:44	INSERT INTO 'mydb'.'Ejemplar_sala' ('id', 'libro_ISBN', 'condicion', 'estado_en_uso') VALUES (10, '9788401...', 1 row(s) affected)	
211	01:58:44	COMMIT	0 row(s) affected
212	01:59:06	SELECT * FROM libro INNER JOIN categoría ON categoría.id=libro.categoría_id LIMIT 0, 1000	10 row(s) returned
213	02:00:11	SELECT * FROM libro INNER JOIN categoría ON categoría.id=libro.categoría_id INNER JOIN autor ON autor.i...	10 row(s) returned
214	02:00:36	SELECT ISBN,título,categoría.nombre,autor.nombre FROM libro INNER JOIN categoría ON categoría.id=libro.c...	Error Code: 1054. Unknown column 'autor.nombre' in field list
215	02:00:52	SELECT ISBN,título,categoría.nombre,autor.name FROM libro INNER JOIN categoría ON categoría.id=libro.cat...	10 row(s) returned
216	02:01:33	SELECT ISBN,título,categoría.nombre,autor.name FROM libro INNER JOIN categoría ON categoría.id=libro.cat...	8 row(s) returned

- Mostrar Los miembros de las bibliotecas y sus direcciones.

1. SELECT \* FROM usuario INNER JOIN direccion  
 2. ON usuario.direccion\_id=direccion.id

**Result Grid**

	id	nombre	apellido	numero_copias_hechas	DNI	direccion_id	id	ciudad	calle	numero	codigo_postal
▶	1	ELISA	PULIDO	0	45e454353	2	2	CABACÉS	GRAN VÍA MAYOR	38	43633
	2	LUIS ALBERTO	ALCAZAR ROVIRA	0	535345435	3	3	VILEÑA	RAMBLA HORNO	51	45591
	4	LOLA	PAVON ROVIRA	0	345346346	3	3	VILEÑA	RAMBLA HORNO	51	45591
	8	ISABEL	DOMINGO LOZANO	0	567564534	3	3	VILEÑA	RAMBLA HORNO	51	45591
	7	JOSE IGNACIO	ARCE GARRIDO	0	435435365	4	4	TOCINA	CORREDERA MAYOR	29	41370
	3	CANDELARIA	CONDE CALLEJA	0	999999999	5	5	CUBILLAS DE CERRATO	CAMINO REAL	37	34410
	6	EDUARD	LAFUENTE CASANOVA	10	554645565	6	6	LANTARÓN	RAMBLA REAL	91	01876
	5	BENITA	ORDÓÑEZ ARENAS	0	345345345	7	7	MACAEL	PASAJE IGLESIA	25	04372
	9	DIONISIO	PONCE VALLS	0	345345343	8	8	GAUCÍN	CALLEJÓN NUEVA	97	29238
	10	MAR	OCAÑA GIL	5	574563454	9	9	PINSEQUÉ	CUESTA MAYOR	63	50410

**Action Output**

#	Time	Action	Message
206	01:03:13	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected
207	01:03:13	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected
208	01:04:05	SELECT * FROM empleado INNER JOIN direccion ON empleado.direccion_id=direccion.id LIMIT 0, 1000	10 row(s) returned
209	01:05:32	SELECT * FROM usuario INNER JOIN colección ON usuario.colección_id=colección.id LIMIT 0, 1000	Error Code: 1054. Unknown column 'usuario.colección_id' in 'on clause'
210	01:06:13	SELECT * FROM usuario INNER JOIN colección ON usuario.id=colección.usuario_id LIMIT 0, 1000	2 row(s) returned
211	01:07:17	SELECT nombre, apellido, colección FROM usuario INNER JOIN colección ON usuario.id=colección.usuario_id ...	2 row(s) returned
212	01:08:49	SELECT * FROM usuario INNER JOIN dirección ON usuario.dirección_id=dirección.id LIMIT 0, 1000	10 row(s) returned

- Mostrar los delitos mencionados en los requisitos, los 3 tipos y a quién corresponden.

Screenshot of a MySQL query execution interface showing the results of a JOIN query.

```

1 •  SELECT * FROM delito INNER JOIN usuario
2   ON usuario.id=delito.usuario_id
3   INNER JOIN reaccion
4   ON delito.usuario_id=reaccion.usuario_id
5
6

```

**Result Grid**

fecha	usuario_id	empleado_seguridad_social	id	nombre	apellido	id	delito_fecha	usuario_id	mensaje	fecha
2019-01-01	1	439389006267	1	ELISA	PULIDO	1	2019-01-01	1	Devuelto tarde	2019-01-01
2019-02-02	2	257202374569	2	LUIS ALBERTO	ALCAZAR ROVIRA	2	2019-02-02	2	Devuelto tarde	2019-02-02
2019-03-03	3	457643396053	3	CANDELARIA	CONDE CALLEJA	3	2019-03-03	3	Solicitud al juez enviada	2019-03-03
2019-04-04	4	353493489677	4	LOLA	PAVON ROVIRA	4	2019-04-04	4	Devuelto tarde	2019-04-04
2019-05-05	5	518995105404	5	BENITA	ORDÓÑEZ ARENAS	5	2019-05-05	5	Multa enviada	2019-05-05
2019-06-06	6	468269319354	6	EDUARD	LAFUENTE CASANOVA	6	2019-06-06	6	Devuelto tarde	2019-06-06
2019-07-07	7	217835147669	7	JOSE IGNACIO	ARCE GARRIDO	7	2019-07-07	7	Solicitud al juez enviada	2019-07-07
2019-08-08	8	088005150489	8	ISABEL	DOMINGO LOZANO	8	2019-08-08	8	Multa enviada	2019-08-08
2019-09-09	9	261355322667	9	DIONISIO	PONCE VALLS	9	2019-09-09	9	Devuelto tarde	2019-09-09
2019-10-10	10	439565855354	10	MAR	OCAÑA GIL	10	2019-10-10	10	Devuelto tarde	2019-10-10

**Output**

#	Time	Action	Message
1	02:04:29	SELECT * FROM delito INNER JOIN usuario ON usuario.id=delito.usuario_id LIMIT 0, 1000	10 row(s) returned
2	02:05:18	SELECT * FROM delito INNER JOIN usuario ON usuario.id=delito.usuario_id INNER JOIN reaccion ON delito.u...	10 row(s) returned

- Mostrar la información del usuario que lee una determinada literatura en la sala de lectura, es decir, quien la reserve.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL query:
 

```

1 • SELECT * FROM reservacion_de_sala INNER JOIN usuario
  ON usuario.id=reservacion_de_sala.usuario_id
  INNER JOIN sala
  ON sala.id=reservacion_de_sala.sala_id
  
```
- Result Grid:** Shows the results of the query, listing 15 rows of data. The columns are: sala\_id, hora\_inicio, hora\_fin, empleado\_seguridad\_social, usuario\_id, id, nombre, apellido, id, and capacidad. The data includes various names like ELISA PULIDO, LUIS ALBERTO ALCAZAR ROVIRA, etc., and room capacities ranging from 10 to 10.
- Action Output:** Shows the history of actions taken in the session, including the execution of the three queries above.

- Mostrar la frecuencia de su uso, en un período de tiempo específico.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL query:
 

```

1 • SELECT COUNT(DISTINCT hora_inicio ) AS 'Numero de uso de la sala despues de las 10:00' FROM reservacion_de_sala INNER JOIN usuario
  ON usuario.id=reservacion_de_sala.usuario_id
  INNER JOIN sala
  ON sala.id=reservacion_de_sala.sala_id
  WHERE hora_inicio>'10:00'
  
```
- Result Grid:** Shows the results of the query, displaying a single row with the value "Numero de uso de la sala despues de las 10:00" followed by a blank cell.
- Action Output:** Shows the history of actions taken in the session, including the execution of the query above and several error messages related to syntax errors in subsequent attempts to run the same or similar queries.

- Mostrar la frecuencia de uso de un tipo particular de literatura y la cantidad que tiene reservados dentro de un período específico.

Screenshot of MySQL Workbench showing a query execution window.

```

1 • SELECT * FROM libro INNER JOIN libro_sala
2   ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ejemplar_sala
3   ON ejemplar_sala.libro.ISBN=libro_sala.libro.ISBN INNER JOIN categoria
4   ON categoria.id=libro.categoria_id
5   WHERE categoria.nombre='Juvenil';
6

```

The Result Grid shows the following data:

ISBN	título	numero_de_paginas	area	estante	categoria_id	tipo_casa_o_sala	autor_id	libro_ISBN	id	libro_ISBN	condicion	estado_en_uso	id	nombre
1408855917	Harry Potter y el Prisionero de Azkaban	450	1	10	1	S	1	1408855917	3	1408855917	Nuevo	1	1	Juvenil
1408855925	Harry Potter y el Cáliz de Fuego	500	1	10	1	S	1	1408855925	4	1408855925	Nuevo	0	1	Juvenil
8478884459	Harry Potter y la Piedra Filosofal	300	1	10	1	C	1	8478884459	1	8478884459	Usado	1	1	Juvenil
8478884955	Harry Potter y la Cámara Secreta	350	1	10	1	C	1	8478884955	2	8478884955	Usado	0	1	Juvenil
8478887423	Harry Potter y la Orden de Felix	600	1	10	1	C	1	8478887423	5	8478887423	Usado	1	1	Juvenil
9788478889907	HARRY POTTER Y EL PRÍNCIPE MESTIZO	600	1	10	1	S	1	9788478889907	6	9788478889907	Usado	0	1	Juvenil
9788498381405	Harry Potter y las Reliquias de la Muerte	800	1	10	1	S	1	9788498381405	7	9788498381405	Nuevo	1	1	Juvenil
9788498387544	Harry Potter y el Legado Maldito	120	1	10	1	C	1	9788498387544	8	9788498387544	Nuevo	0	1	Juvenil

The Output pane shows the following log entries:

#	Time	Action	Message	Duration / f
24	02:20:12	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	10 row(s) returned	0.000 sec /
25	02:20:31	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	Error Code: 1054. Unknown column 'categoria.nombre' in 'where clause'	0.000 sec /
26	02:20:43	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	Error Code: 1054. Unknown column 'nombre' in 'where clause'	0.016 sec /
27	02:20:48	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	10 row(s) returned	0.016 sec /
28	02:20:59	SELECT categoria.nombre FROM libro INNER JOIN libro_sala ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ...	10 row(s) returned	0.000 sec /
29	02:21:14	SELECT categoria.nombre FROM libro INNER JOIN libro_sala ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ...	8 row(s) returned	0.000 sec /
30	02:21:23	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro.ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	8 row(s) returned	0.000 sec /

SQL File 3\* | SQL File 4 | SQL File 5 | SQL File 6 | SQL File 7\*

```

1 •  SELECT COUNT(*) AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala
2   ON libro_sala.libro_ISBN=libro.ISBN INNER JOIN ejemplar_sala
3   ON ejemplar_sala.libro_ISBN=libro_sala.libro_ISBN INNER JOIN categoria
4   ON categoria.id=libro.categoria_id
5   WHERE categoria.nombre='Juvenil';
6

```

Result Grid | Filter Rows: [ ] | Export: | Wrap Cell Content: [ ]

	Frecuencia de uso de libros Juveniles
▶	8

Result 35 X

Output

Action Output	#	Time	Action	Message
✓ 28 02:20:59	SELECT categoria.nombre FROM libro INNER JOIN libro_sala ON libro_sala.libro_ISBN=libro.ISBN INNER JOI...	10 row(s) returned		
✓ 29 02:21:14	SELECT categoria.nombre FROM libro INNER JOIN libro_sala ON libro_sala.libro_ISBN=libro.ISBN INNER JOI...	8 row(s) returned		
✓ 30 02:21:23	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro_ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	8 row(s) returned		
✗ 31 02:22:36	SELECT COUNT(DISTINCT id) AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ...	Error Code: 1052. Column 'id' in field list		
✗ 32 02:22:50	SELECT COUNT AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala.J...	Error Code: 1054. Unknown column 'C'		
✗ 33 02:22:55	SELECT COUNT() AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala...	Error Code: 1064. You have an error in		
✓ 34 02:23:00	SELECT COUNT(*) AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sal...	1 row(s) returned		

- ¿Qué libro fue el más prestado? -  
Mostramos las reservas de libros realizadas :

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query:
 

```
1 •  SELECT * FROM prestamo INNER JOIN libro
2   ON libro.ISBN=prestamo.libro_ISBN
3
```
- Result Grid:** Displays the results of the query, showing multiple rows of data from the joined tables.
- Output Panel:** Shows the execution history (Result 8) with the following log entries:
 

#	Time	Action	Message	Duration
211	01:07:17	SELECT nombre.apellido,coleccion FROM usuario INNER JOIN colección ON usuario.id=colección.usuario_id ...	2 row(s) returned	0.000 se
212	01:08:49	SELECT * FROM usuario INNER JOIN dirección ON usuario.dirección_id=dirección.id LIMIT 0, 1000	10 row(s) returned	0.031 se
213	01:09:43	SELECT * FROM préstamo WHERE libro.ISBN!=NULL LIMIT 0, 1000	Error Code: 1054. Unknown column 'libro.ISBN' in 'where clause'	0.000 se
214	01:10:24	SELECT * FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN WHERE libro.ISBN!=NULL ...	0 row(s) returned	0.000 se
215	01:10:30	SELECT * FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN LIMIT 0, 1000	5 row(s) returned	0.063 se
216	01:11:13	SELECT MAX(título) FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN LIMIT 0, 1000	1 row(s) returned	0.062 se
217	01:11:25	SELECT * FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN LIMIT 0, 1000	5 row(s) returned	0.000 se

Mostramos el libro más prestado:

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query:
 

```
1 •  SELECT MAX(título) FROM préstamo INNER JOIN libro
2   ON libro.ISBN=préstamo.libro_ISBN
3
```
- Result Grid:** Displays the results of the query, showing one row with the title "Harry Potter y el Prisionero de Azkaban".
- Output Panel:** Shows the execution history (Result 9) with the following log entries:
 

#	Time	Action	Message	Duration
212	01:08:49	SELECT * FROM usuario INNER JOIN dirección ON usuario.dirección_id=dirección.id LIMIT 0, 1000	10 row(s) returned	
213	01:09:43	SELECT * FROM préstamo WHERE libro.ISBN!=NULL LIMIT 0, 1000	Error Code: 1054. Unknown column 'libro.ISBN' in 'where clause'	
214	01:10:24	SELECT * FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN WHERE libro.ISBN!=NULL ...	0 row(s) returned	
215	01:10:30	SELECT * FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN LIMIT 0, 1000	5 row(s) returned	
216	01:11:13	SELECT MAX(título) FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN LIMIT 0, 1000	1 row(s) returned	
217	01:11:25	SELECT * FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN LIMIT 0, 1000	5 row(s) returned	
218	01:12:04	SELECT MAX(título) FROM préstamo INNER JOIN libro ON libro.ISBN=préstamo.libro_ISBN LIMIT 0, 1000	1 row(s) returned	

- ¿Cuántos libros ha tomado prestado este año ELISA PULIDO?

The screenshot shows the MySQL Workbench interface with the following details:

**Query Editor:**

```

1 • SELECT * FROM usuario INNER JOIN prestamo
  ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_casa
  ON ejemplar_casa.libro_ISBN=prestamo.ejemplar_casa_ISBN INNER JOIN libro
  ON libro.ISBN=ejemplar_casa.libro_ISBN
  WHERE nombre='ELISA' && apellido='PULIDO'
  
```

**Result Grid:**

id	nombre	apellido	fecha_inicio	fecha_fin	usuario_id	empleado_seguridad_social	ejemplar_casa_ISBN	id	libro_ISBN	estado_prestado	condicion	ISBN	título	numero...
1	ELISA	PULIDO	2019-01-01	2019-02-02	1	439389006267	8478884459	1	8478884459	1	Usado	8478884459	Harry Potter y la Piedra Filosofal	300

**Action Output:**

#	Time	Action	Message	Durati...
49	02:34:17	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_ca...	10 row(s) returned	0.000
50	02:34:43	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_ca...	10 row(s) returned	0.000
51	02:35:04	SELECT nombre.apellido.título FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNE...	10 row(s) returned	0.000
52	02:36:02	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_ca...	10 row(s) returned	0.000
53	02:36:21	SELECT COUNT(título) FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN...	1 row(s) returned	0.000
54	02:36:33	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_ca...	10 row(s) returned	0.000
55	02:37:13	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_ca...	1 row(s) returned	0.016

- ¿Qué libros son de uso exclusivo en la sala de lectura?

The screenshot shows the MySQL Workbench interface with the following details:

**Query Editor:**

```

1 • SELECT * FROM libro WHERE tipo_casa_o_sala='S'
  
```

**Result Grid:**

ISBN	título	numero_de_paginas	area	estante	categoria_id	tipo_casa_o_sala	autor_id
1408855917	Harry Potter y el Prisionero de Azkaban	450	1	10	1	S	1
1408855925	Harry Potter y el Cáliz de Fuego	500	1	10	1	S	1
9788478889907	Harry Potter y el Príncipe Mestizo	600	1	10	1	S	1
9788498381405	Harry Potter y las Reliquias de la Muerte	800	1	10	1	S	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Action Output:**

#	Time	Action	Message	Durati...
34	02:23:00	SELECT COUNT(*) AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala...	1 row(s) returned	
35	02:23:39	SELECT COUNT(*) AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala...	1 row(s) returned	
36	02:23:45	SELECT (*) AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala.libro_...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SELECT (*) AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala.libro_...' at line 1	
37	02:23:49	SELECT * AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala.libro_IS...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SELECT * AS 'Frecuencia de uso de libros Juveniles' FROM libro INNER JOIN libro_sala ON libro_sala.libro_IS...' at line 1	
38	02:23:55	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro_ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	10 row(s) returned	
39	02:24:57	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro_ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	10 row(s) returned	
40	02:26:12	SELECT * FROM libro WHERE tipo_casa_o_sala='S' LIMIT 0, 1000	4 row(s) returned	

- ¿Qué libros están reservados y quién ha hecho la reserva?

SQL Editor:

```

1 •  SELECT nombre,apellido,titulo FROM usuario INNER JOIN prestamo
2   ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_casa
3   ON ejemplar_casa.libro_ISBN=prestamo.ejemplar_casa_ISBN INNER JOIN libro
4   ON libro.ISBN=ejemplar_casa.libro_ISBN;
5
6

```

Result Grid:

nombre	apellido	título
ELISA	PULIDO	Harry Potter y la Piedra Filosofal
LUIS ALBERTO	ALCAZAR ROVIRA	Harry Potter y la Cámara Secreta
CANDELARIA	CONDE CALLEJA	Harry Potter y el Prisionero de Azkaban
LOLA	PAVON ROVIRA	Harry Potter y el Cáñamo Fuego
BENITA	ORDÓÑEZ ARENAS	Harry Potter y la Orden de Felix
EDUARD	LAFUENTE CASANOVA	Harry Potter y el Príncipe Mestizo
JOSE IGNACIO	ARCE GARRIDO	Harry Potter y las Reliquias de la Muerte
ISABEL	DOMINGO LOZANO	Harry Potter y el Legado Maldito
DIONISIO	PONCE VALLS	It
MAR	OCAÑA GIL	El Instituto

Result 48 ×

Output:

#	Time	Action	Message
45	02:31:03	SELECT * FROM empleado INNER JOIN reservacion_de_sala ON empleado.seguridad_social=reservacion_de_sala	Error Code: 1146. Table 'mydb.emplad' doesn't exist
46	02:31:11	SELECT * FROM empleado INNER JOIN reservacion_de_sala ON empleado.seguridad_social=reservacion_de_sala	Error Code: 1054. Unknown column 'seguridad_social' in 'field list'
47	02:31:34	SELECT * FROM empleado INNER JOIN reservacion_de_sala ON empleado.seguridad_social=reservacion_de_sala	10 row(s) returned
48	02:32:40	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id LIMIT 0, 1000	10 row(s) returned
49	02:34:17	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_casa ON prestamo.ejemplar_casa_ISBN=ejemplar_casa.ISBN	10 row(s) returned
50	02:34:43	SELECT * FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_casa ON prestamo.ejemplar_casa_ISBN=ejemplar_casa.ISBN	10 row(s) returned
51	02:35:04	SELECT nombre,apellido,titulo FROM usuario INNER JOIN prestamo ON usuario.id=prestamo.usuario_id INNER JOIN ejemplar_casa ON prestamo.ejemplar_casa_ISBN=ejemplar_casa.ISBN	10 row(s) returned

- Mostrar libros favoritos de una determinada colección. Un ejemplo sería :  
Usuario A : {colección de Programación -> C++,Ruby..etc  
colección de Novelas -> Titanic, Amor pasionado..etc }

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL editor window displays the following code:

```
1 •   SELECT nombre,apellido,colección FROM usuario INNER JOIN colección
2       on usuario.id=colección.usuario_id
```

Below the code, the "Result Grid" section shows a table with three rows of data:

	nombre	apellido	colección
▶	ELISA	PULIDO	BLOB
▶	LUIS ALBERTO	ALCAZAR ROVIRA	BLOB

At the bottom of the interface, there are tabs for "Result 3", "Output", and "Action Output".

Ha habido un problema de seguridad que no permitía mostrar el contenido de la variable de colección por ser de tipo BLOB. En este se insertó un archivo de texto / imagen como ejemplos de colecciones.

- Mostrar las salas a poder usarse por los alumnos de la biblioteca y sobre el personal mostrar sus horarios, turnos, salario, datos de la seguridad social..etc

SQL File 3

```

1 •  SELECT * FROM usuario INNER JOIN reservacion_de_sala
2   ON reservacion_de_sala.usuario_id=usuario.id INNER JOIN sala
3   ON sala.id=reservacion_de_sala.sala_id

```

**Result Grid**

	id	nombre	apellido	sala_id	hora_inicio	hora_fin	empleado_seguridad_social	usuario_id	id	capacidad
▶	1	ELISA	PULIDO	1	09:00:00	10:00:00	439389006267	1	1	10
	2	LUIS ALBERTO	ALCAZAR ROVIRA	2	10:00:00	11:00:00	257202374569	2	2	10
	3	CANDELARIA	CONDE CALLEJA	3	11:00:00	12:00:00	457643396053	3	3	10
	4	LOLA	PAVON ROVIRA	4	12:00:00	13:00:00	353493489677	4	4	10
	5	BENITA	ORDOÑEZ ARENAS	5	13:00:00	14:00:00	518995105404	5	5	10
	6	EDUARD	LAFUENTE CASANOVA	6	14:00:00	15:00:16	468269319354	6	6	10
	7	JOSE IGNACIO	ARCE GARRIDO	7	15:00:16	16:00:00	217835147669	7	7	10
	8	ISABEL	DOMINGO LOZANO	8	16:00:00	17:00:00	088005150489	8	8	10
	9	DIONISIO	PONCE VALLS	9	17:00:00	09:00:00	261355322667	9	9	10
	10	MAR.	OCAÑA GIL	10	09:00:00	10:00:00	439565855354	10	10	10

**Result 43**

**Action Output**

#	Time	Action	Message
38	02:23:55	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro_ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	10 row(s) returned

SQL File 3

```

1 •  SELECT * FROM empleado INNER JOIN reservacion_de_sala
2   ON empleado.seguridad_social=reservacion_de_sala.empleado_seguridad_social

```

**Result Grid**

	seguridad_social	turno	horario	salario	sala_id	hora_inicio	hora_fin	empleado_seguridad_social	usuario_id
▶	088005150489	Tarde	15:00 - 20:00	1200	8	16:00:00	17:00:00	088005150489	8
	217835147669	Mañana	09:00 - 14:00	1200	7	15:00:16	16:00:00	217835147669	7
	257202374569	Tarde	15:00 - 20:00	1200	2	10:00:00	11:00:00	257202374569	2
	261355322667	Mañana	09:00 - 14:00	1200	9	17:00:00	09:00:00	261355322667	9
	353493489677	Tarde	15:00 - 20:00	1200	4	12:00:00	13:00:00	353493489677	4
	439389006267	Mañana	09:00 - 14:00	1200	1	09:00:00	10:00:00	439389006267	1
	439565855354	Tarde	15:00 - 20:00	1200	10	09:00:00	10:00:00	439565855354	10
	457643396053	Mañana	09:00 - 14:00	1200	3	11:00:00	12:00:00	457643396053	3
	468269319354	Tarde	15:00 - 20:00	1200	6	14:00:00	15:00:16	468269319354	6
	518995105404	Mañana	09:00 - 14:00	1200	5	13:00:00	14:00:00	518995105404	5

**Result 44**

**Action Output**

#	Time	Action	Message
41	02:26:40	SELECT * FROM libro INNER JOIN libro_sala ON libro_sala.libro_ISBN=libro.ISBN INNER JOIN ejemplar_sala ...	10 row(s) returned
42	02:28:08	SELECT * FROM sala LIMIT 0, 1000	10 row(s) returned
43	02:29:15	SELECT * FROM usuario INNER JOIN reservacion_de_sala ON reservacion_de_sala.usuario_id=usuario.id Li...	10 row(s) returned
44	02:29:43	SELECT * FROM usuario INNER JOIN reservacion_de_sala ON reservacion_de_sala.usuario_id=usuario.id IN...	10 row(s) returned
45	02:31:03	SELECT * FROM empleado INNER JOIN reservacion_de_sala ON empleado.seguridad_social=reservacion_de...	Error Code: 1146. Table 'mydb.empleado' doesn't exist
46	02:31:11	SELECT * FROM empleado INNER JOIN reservacion_de_sala ON empleado.seguridad_social=reservacion_d...	Error Code: 1054. Unknown column 'reservacion_de_sala.seguridad_social' in 'on clause'
47	02:31:34	SELECT * FROM empleado INNER JOIN reservacion_de_sala ON empleado.seguridad_social=reservacion_d...	10 row(s) returned

- Quienes han reservado un portatil y cuando :

Screenshot of MySQL Workbench showing a query execution window.

```

1 SELECT * FROM recurso INNER JOIN portatil
2 ON recurso.id=portatil.recurso_id
3 INNER JOIN prestamo_portatil
4 INNER JOIN usuario
5 ON usuario.id=prestamo_portatil.usuario_id
    
```

The Result Grid displays the following data:

	id	tipo	recurso_id	marca	modelo	fecha_inicio	fecha_fin	usuario_id	empleado_seguridad_social	portatil_id	id	nombre	apellido	numero_copias_hechas	DNI	direc
▶	3	Portatil	3	Toshiba	FSW	2019-02-02	2019-02-03	1	257202374569	1	1	ELISA	PULIDO	0	45e454353	2
	3	Portatil	3	Asus	SDF	2019-02-02	2019-02-03	1	257202374569	1	1	ELISA	PULIDO	0	45e454353	2
	3	Portatil	3	Dell	WDW	2019-02-02	2019-02-03	1	257202374569	1	1	ELISA	PULIDO	0	45e454353	2
	3	Portatil	3	Toshiba	FSW	2019-02-03	2019-02-04	2	257202374569	2	2	LUIS ALBERTO	ALCAZAR ROVIRA	0	535345435	3
	3	Portatil	3	Asus	SDF	2019-02-03	2019-02-04	2	257202374569	2	2	LUIS ALBERTO	ALCAZAR ROVIRA	0	535345435	3
	3	Portatil	3	Dell	WDW	2019-02-03	2019-02-04	2	257202374569	2	2	LUIS ALBERTO	ALCAZAR ROVIRA	0	535345435	3
	3	Portatil	3	Toshiba	FSW	2019-02-04	2019-02-05	3	518995105404	2	3	CANDELARIA	CONDE CALLEJA	0	999999999	5
	3	Portatil	3	Asus	SDF	2019-02-04	2019-02-05	3	518995105404	2	3	CANDELARIA	CONDE CALLEJA	0	999999999	5
	3	Portatil	3	Dell	WDW	2019-02-04	2019-02-05	3	518995105404	2	3	CANDELARIA	CONDE CALLEJA	0	999999999	5

The Output pane shows the following log entries:

Action	Time	Action	Message
② 221	01:15:44	SELECT * FROM recurso INNER JOIN portatil IN portatil.id=portatil.recurso_id	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se
② 222	01:15:49	SELECT * FROM recurso INNER JOIN portatil ON portatil.id=portatil.recurso_id LIMIT 0, 1000	Error Code: 1054. Unknown column 'portatil.id' in 'on clause'
② 223	01:15:50	SELECT * FROM recurso INNER JOIN portatil ON portatil.id=portatil.recurso_id LIMIT 0, 1000	Error Code: 1054. Unknown column 'portatil.id' in 'on clause'
③ 224	01:16:17	SELECT * FROM recurso INNER JOIN portatil ON recurso.id=portatil.recurso_id LIMIT 0, 1000	3 row(s) returned
③ 225	01:19:22	SELECT * FROM recurso INNER JOIN portatil ON recurso.id=portatil.recurso_id INNER JOIN prestamo_portatil...	0 row(s) returned
③ 226	01:19:30	SELECT * FROM recurso INNER JOIN portatil ON recurso.id=portatil.recurso_id INNER JOIN prestamo_portatil...	9 row(s) returned
③ 227	01:20:14	SELECT * FROM recurso INNER JOIN portatil ON recurso.id=portatil.recurso_id INNER JOIN prestamo_portatil...	9 row(s) returned

## Restricciones de actualización:

Mostramos dos ejemplos de las restricciones de actualización y borrado creadas:

1- CASCADE. Al crear la restricción CASCADE en el foreign key correspondiente al id del usuario, al intentar actualizar el valor de este, automáticamente se actualizará también dicho valor en la tabla Delito. Se muestra en el ejemplo como se cambia el id del usuario “Isabel” que tiene id=8 al valor 11. Ese dato se actualiza en la tabla del usuario y de delito.

```
mysql> SELECT * FROM Delito
-> ;
+-----+-----+-----+
| fecha | usuario_id | empleado_seguridad_social |
+-----+-----+-----+
| 2019-08-08 | 8 | 088005150489
| 2019-07-07 | 7 | 217835147669
| 2019-02-02 | 2 | 257202374569
| 2019-09-09 | 9 | 261355322667
| 2019-04-04 | 4 | 353493489677
| 2019-01-01 | 1 | 439389006267
| 2019-10-10 | 10 | 439565855354
| 2019-03-03 | 3 | 457643396053
| 2019-06-06 | 6 | 468269319354
| 2019-05-05 | 5 | 518995105404
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> SELECT * FROM Usuario;
+-----+-----+-----+-----+-----+-----+
| id | nombre | apellido | numero_copias_hechas | DNI | direccion_id |
+-----+-----+-----+-----+-----+-----+
| 1 | ELISA | PULIDO | 0 | 45e454353 | 2 |
| 2 | LUIS ALBERTO | ALCAZAR ROVIRA | 0 | 535345435 | 3 |
| 3 | CANDELARIA | CONDE CALLEJA | 0 | 999999999 | 5 |
| 4 | LOLA | PAVON ROVIRA | 0 | 345346346 | 3 |
| 5 | BENITA | ORDOÑEZ ARENAS | 0 | 345345345 | 7 |
| 6 | EDUARD | LAFUENTE CASANOVA | 10 | 554645565 | 6 |
| 7 | JOSE IGNACIO | ARCE GARRIDO | 0 | 435435365 | 4 |
| 8 | ISABEL | DOMINGO LOZANO | 0 | 567564534 | 3 |
| 9 | DIONISIO | PONCE VALLS | 0 | 345345343 | 8 |
| 10 | MAR | OCAÑA GIL | 5 | 574563454 | 9 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

```

mysql> UPDATE Usuario SET id=11 WHERE id=8;
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Usuario;
+----+-----+-----+-----+-----+
| id | nombre | apellido | numero_copias_hechas | DNI   | direccion_id |
+----+-----+-----+-----+-----+
| 1  | ELISA  | PULIDO   | 0                | 45e454353 | 2      |
| 2  | LUIS ALBERTO | ALCAZAR ROVIRA | 0                | 535345435 | 3      |
| 3  | CANDELARIA | CONDE CALLEJA | 0                | 999999999 | 5      |
| 4  | LOLA    | PAVON ROVIRA | 0                | 345346346 | 3      |
| 5  | BENITA  | ORDOÑEZ ARENAS | 0                | 345345345 | 7      |
| 6  | EDUARD  | LAFUENTE CASANOVA | 10               | 554645565 | 6      |
| 7  | JOSE IGNACIO | ARCE GARRIDO | 0                | 435435365 | 4      |
| 9  | DIONISIO | PONCE VALLS  | 0                | 345345343 | 8      |
| 10 | MAR     | OCAÑA GIL   | 5                | 574563454 | 9      |
| 11 | ISABEL  | DOMINGO LOZANO | 0                | 567564534 | 3      |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> SELECT * FROM Delito
-> ;
+-----+-----+-----+
| fecha | usuario_id | empleado_seguridad_social |
+-----+-----+-----+
| 2019-08-08 | 11 | 088005150489 |
| 2019-07-07 | 7  | 217835147669 |
| 2019-02-02 | 2  | 257202374569 |
| 2019-09-09 | 9  | 261355322667 |
| 2019-04-04 | 4  | 353493489677 |
| 2019-01-01 | 1  | 439389006267 |
| 2019-10-10 | 10 | 439565855354 |
| 2019-03-03 | 3  | 457643396053 |
| 2019-06-06 | 6  | 468269319354 |
| 2019-05-05 | 5  | 518995105404 |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> _

```

## 2-Delete :

Se intenta eliminar una categoría asociada a un libro, y el sistema no lo permite debido a la restricción

The screenshot shows a MySQL Workbench interface with two panes. The top pane displays a query editor with the following SQL code:

```
1 USE mydb;
2 • SELECT categoria_id FROM Libro WHERE titulo='it'
```

The results grid shows one row:

#	categoria_id
1	2

The bottom pane shows the Action Output history:

#	Time	Action	Message
1	17:55:40	USE mydb	0 row(s) affected
2	17:55:40	DELETE FROM Categoria WHERE id=1	Error Code: 1451. Cannot delete or update a parent
3	17:56:50	USE mydb	0 row(s) affected
4	17:56:50	SELECT * FROM Libro LIMIT 0, 1000	10 row(s) returned
5	17:57:08	USE mydb	0 row(s) affected
6	17:57:08	SELECT * FROM Libro WHERE titulo='it' LIMIT 0, 1000	1 row(s) returned
7	17:57:43	USE mydb	0 row(s) affected
8	17:57:43	SELECT categoria_id FROM Libro WHERE titulo='it' LIMIT...	1 row(s) returned
9	17:57:43	USE mydb	0 row(s) affected
10	17:57:43	SELECT categoria_id FROM Libro WHERE titulo='it' LIMIT...	1 row(s) returned

The second action (DELETE) is highlighted with an orange background, indicating it failed. The message for this action is: "Error Code: 1451. Cannot delete or update a parent".

The bottom-most pane shows the same SQL code and history, with the failed DELETE statement highlighted. A tooltip on the right side of the bottom pane provides the detailed error message:

Automatic context help is disabled. Use the toolbar to manually get help for the current toggle automatic help.

Action: DELETE FROM Categoria WHERE id=1  
Response: Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('mydb'.'Libro', CONSTRAINT 'categoria' FOREIGN KEY ('categoria\_id') REFERENCES 'Categoria' ('id') ON UPDATE CASCADE)  
Duration: 0,258 sec

## **Pruebas de triggers implementados:**

1.- Todos los recursos se han de catalogar en uno de los tres tipos para ello : Libro, revista o portatil. Se crea un trigger para comprobar que se asocia un recurso dentro del dominio correspondiente : libro - 1, revista - 2 o portatil - 3. Ejemplo de trigger para libro:

```
USE mydb;

delimiter //
DROP PROCEDURE IF EXISTS lanzar_error //
CREATE PROCEDURE lanzar_error ()
BEGIN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tipo de recurso mal asociado';
END//


delimiter ;

DELIMITER $$
DROP TRIGGER IF EXISTS assert_catalogacion $$ 
CREATE TRIGGER assert_catalogacion
BEFORE INSERT ON libro FOR EACH ROW
BEGIN

DECLARE _result,_ISBN BOOL DEFAULT FALSE;

IF (NEW.recurso_id>3) THEN
    SET _result=TRUE;
END IF;

IF (_result) THEN
    CALL lanzar_error();
END IF;

IF (NEW.ISBN!=NULL AND NEW.recurso_id!=1) THEN
    SET _ISBN=TRUE;
END IF;

IF (_ISBN) THEN
    CALL lanzar_error();
END IF;

END $$
DELIMITER ;
```

```
1 USE mydb;
2 INSERT INTO `mydb`.`Libro` (`recurso_id`, `ISBN`, `titulo`, `numero_de_paginas`, `area`, `estante`, `categoria_id`, `tipo_recurso_id`)
3
```

Output:

Action	Time	Action	Message	
✓	74	01:56:04	CREATE TRIGGER assert_catalogacion BEFORE INSERT ON libro FOR EACH ROW BEGIN DECLARE _res...	0 row(s) affected
✓	75	01:57:14	DROP PROCEDURE IF EXISTS lanzar_error	0 row(s) affected
✓	76	01:57:14	CREATE PROCEDURE lanzar_error () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tipo de r...	0 row(s) affected
✓	77	01:57:14	DROP TRIGGER IF EXISTS assert_catalogacion	0 row(s) affected
✓	78	01:57:15	CREATE TRIGGER assert_catalogacion BEFORE INSERT ON libro FOR EACH ROW BEGIN DECLARE _res...	0 row(s) affected
✓	79	01:57:18	USE mydb	0 row(s) affected
✗	80	01:57:18	INSERT INTO `mydb`.`Libro` (`recurso_id`, `ISBN`, `titulo`, `numero_de_paginas`, `area`, `estante`, `categoria_id`, `tipo_recurso_id`)	Error Code: 1644. Tipo de recurso mal asociado

2.- Si el usuario devuelve un libro más tarde de los días que tiene permitido, se envía de inmediato un recordatorio como primer aviso; (1 dia mas tarde por poner un ejemplo)

USE mydb;

```
delimiter //
DROP PROCEDURE IF EXISTS enviar_aviso //
CREATE PROCEDURE enviar_aviso (IN candidato INT)
BEGIN

INSERT INTO `mydb`.`Reaccion` (`id`, `delito_fecha`, `usuario_id`,
`mensaje`, `fecha`) VALUES (10, '2019-05-06', candidato, 'Enviado
aviso', '2019-05-08');

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El aviso ha sido
enviado al usuario correspondiente.';

END// 

delimiter ;

DELIMITER $$ 
DROP TRIGGER IF EXISTS enviar_aviso_trigger $$ 
CREATE TRIGGER enviar_aviso_trigger
BEFORE UPDATE ON prestamo FOR EACH ROW
BEGIN

DECLARE _result BOOL DEFAULT FALSE;

    if NEW.fecha_fin>DATE_ADD(NEW.fecha_inicio, INTERVAL 1 DAY)
THEN
    SET _RESULT=TRUE;
END IF;

    if (_result) THEN
CALL enviar_aviso(NEW.usuario_id);

END IF;

END $$
```

```
DELIMITER ;
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a query editor window displays two SQL statements:

```
1 • USE mydb;
2 • UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5;
```

Below the query editor is an "Output" pane titled "Action Output". It contains a table with six rows, each representing an action taken by the MySQL server:

#	Time	Action	Message
1	02:25:40	DROP PROCEDURE IF EXISTS enviar_aviso	0 row(s) affected
2	02:25:40	CREATE PROCEDURE enviar_aviso (IN candidato INT) BEGIN INSERT INTO `mydb`.`Reaccion` (id, `delito_fe...`	0 row(s) affected
3	02:25:40	DROP TRIGGER IF EXISTS enviar_aviso_trigger	0 row(s) affected
4	02:25:41	CREATE TRIGGER enviar_aviso_trigger BEFORE UPDATE ON prestamo FOR EACH ROW BEGIN DECLAR...	0 row(s) affected
5	02:25:43	USE mydb	0 row(s) affected
6	02:25:43	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5	Error Code: 1644. El aviso ha sido enviado al usuario correspondiente.

Vemos que efectivamente se ha creado una alerta con esa fecha:

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, a query editor window displays the command: `SELECT * FROM reaccion;`. The result grid shows the following data:

	id	delito_fecha	usuario_id	mensaje	fecha
5	2019-05-05	5	Enviada multa	2019-05-05	
6	2019-06-06	6	Enviada multa	2019-06-06	
7	2019-07-07	7	Enviada solicitud a juez	2019-07-07	
8	2019-05-06	2	Enviada multa	2019-05-08	
8	2019-05-06	3	Enviada multa	2019-05-08	
8	2019-05-06	4	Enviada multa	2019-05-08	
8	2019-08-08	8	Enviada solicitud a juez	2019-08-08	
9	2019-09-09	9	Devuelto tarde	2019-09-09	
10	2019-10-10	10	Devuelto tarde	2019-10-10	
	HULL	HULL	HULL	HULL	

Below the result grid, the message `reaccion 1` is displayed. Under the "Output" tab, the "Action Output" section shows the following log entries:

#	Time	Action	Message
1	02:25:40	DROP PROCEDURE IF EXISTS enviar_aviso	0 row(s) affected
2	02:25:40	CREATE PROCEDURE enviar_aviso (IN candidato INT) BEGIN INSERT INTO `mydb`.`Reaccion` (`id`, `delito_f...`	0 row(s) affected
3	02:25:40	DROP TRIGGER IF EXISTS enviar_aviso_trigger	0 row(s) affected
4	02:25:41	CREATE TRIGGER enviar_aviso_trigger BEFORE UPDATE ON prestamo FOR EACH ROW BEGIN DECLA...	0 row(s) affected
5	02:25:43	USE mydb	0 row(s) affected
6	02:25:43	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5	Error Code: 1644. El aviso ha sido enviado al usuario correspondiente.
7	02:26:09	SELECT * FROM reaccion LIMIT 0, 1000	13 row(s) returned

### 3.- Una multa financiera como segundo aviso; (2 dias mas tarde)

USE mydb;

```

delimiter //
DROP PROCEDURE IF EXISTS enviar_multa //
CREATE PROCEDURE enviar_multa (IN candidato INT)
BEGIN

INSERT INTO `mydb`.`Reaccion` (`id`, `delito_fecha`, `usuario_id`, `mensaje`, `fecha`)
VALUES (8, '2019-05-06', candidato, 'Enviada multa', '2019-05-08');

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La multa ha sido enviada al usuario correspondiente.';

END// 

delimiter ;

DELIMITER $$ 
DROP TRIGGER IF EXISTS enviar_multa_trigger $$ 
CREATE TRIGGER enviar_multa_trigger 
BEFORE UPDATE ON prestamo FOR EACH ROW 
BEGIN

DECLARE _result BOOL DEFAULT FALSE;

if NEW.fecha_fin>DATE_ADD(NEW.fecha_inicio, INTERVAL 2 DAY) THEN
    SET _RESULT=TRUE;

```

```

END IF;

if (_result) THEN
CALL enviar_multa(NEW.usuario_id);

END IF;

END $$

DELIMITER ;

```

The screenshot shows the MySQL Workbench interface. In the top query editor window, there are two statements:

```

1 • USE mydb;
2 • UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5;

```

In the bottom output pane, there is a table titled "Action Output" showing the results of the executed statements:

#	Time	Action	Message
4	02:20:04	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=4	Error Code: 1062. Duplicate entry '8-2019-05-06-4' for key 'PRIMARY'
5	02:21:08	DROP PROCEDURE IF EXISTS enviar_multa	0 row(s) affected
6	02:21:08	CREATE PROCEDURE enviar_multa (IN candidato INT) BEGIN INSERT INTO `mydb`.`Reaccion` (id, `delto_...`)	0 row(s) affected
7	02:21:08	DROP TRIGGER IF EXISTS enviar_multa_trigger	0 row(s) affected
8	02:21:09	CREATE TRIGGER enviar_multa_trigger BEFORE UPDATE ON prestamo FOR EACH ROW BEGIN DECLA...	0 row(s) affected
9	02:21:19	USE mydb	0 row(s) affected
10	02:21:19	UPDATE prestamo SET fecha_fin="2020-03-14" WHERE usuario_id=5	Error Code: 1644. La multa ha sido enviada al usuario correspondiente.

Observamos el error #4 como demostración , ya que al darle dos veces se había creado ya la primera vez el mensaje y entonces dice que está duplicado.

4.- Parte de la literatura ( de los libros ) de la biblioteca está permitida solo para las salas que se reservan ( salas de lectura), y la literatura que tiene este código especial se considerará por separado.

USE mydb;

```
delimiter //
DROP PROCEDURE IF EXISTS enviar_mensaje //
CREATE PROCEDURE enviar_mensaje ()
BEGIN
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede llevar
este libro a su casa. Lo sentimos';
```

```
END//
```

```
delimiter ;
```

```
DELIMITER $$  
DROP TRIGGER IF EXISTS libro_casa $$  
CREATE TRIGGER libro_casa  
BEFORE INSERT ON prestamo FOR EACH ROW  
BEGIN
```

```
DECLARE _result BOOL DEFAULT FALSE;
```

```
    if ((SELECT tipo_casa_o_sala FROM libro WHERE NEW.Libro_ISBN =
libro.ISBN) ='s' ) THEN
```

```

SET __RESULT=TRUE;

END IF;

if (_result) THEN
CALL enviar_mensaje();

END IF;

END $$

DELIMITER ;

```

The screenshot shows the MySQL Workbench interface. The top part is a query editor with the following SQL code:

```

1  INSERT INTO `Libro_casa`(`Libro_ISBN`, `Revista_titulo`, `Portatil_modelo`, `Portatil_marca`) VALUES ('2019-02-09', '2019-03-03', 2, '9788466347921', 'S', '1408855917', NULL, NULL, NULL)
2
3
]
```

The bottom part is an "Output" window titled "Action Output" showing the execution log:

#	Time	Action	Message	Duration
33	02:36:45	INSERT INTO 'mydb'.'Prestamo' ('fecha_inicio', 'fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_...')	1 row(s) affected	0.062 sec
34	02:37:39	DROP PROCEDURE IF EXISTS enviar_mensaje	0 row(s) affected	0.125 sec
35	02:37:39	CREATE PROCEDURE enviar_mensaje () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No ...'	0 row(s) affected	0.047 sec
36	02:37:39	DROP TRIGGER IF EXISTS libro_casa	0 row(s) affected	0.063 sec
37	02:37:39	CREATE TRIGGER libro_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _resul...	0 row(s) affected	0.109 sec
38	02:37:42	USE mydb	0 row(s) affected	0.000 sec
39	02:37:42	INSERT INTO 'mydb'.'Prestamo' ('fecha_inicio', 'fecha_fin', 'usuario_id', 'empleado_seguridad_social', 'casa_...')	Error Code: 1644. No se puede llevar este libro a su casa. Lo sentimos	0.000 sec

Vemos que efectivamente este libro es solo para la sala y no lo deja reservar:

```

1 •    `io_seguridad_social`, `casa_o_sala`, `Libro_ISBN`, `Revista_título`, `Portatil_modelo`, `Portatil_marca` ) VALUES ('2019-02-09', '2019-0
2 •
3

```

recurso_id	ISBN	título	numero_de_páginas	área	estante	categoría_id	tipo_casa_o_sala	autor_id
1	1408855917	Harry Potter y el Prisionero de Azkaban	450	1	10	1	c	1
1	1408855925	Harry Potter y el Cáliz de Fuego	500	1	10	1	c	1
1	8478884459	Harry Potter y la Piedra Filosofal	300	1	10	1	c	1
1	8478884955	Harry Potter y la Cámara Secreta	350	1	10	1	c	1
1	8478887423	Harry Potter y la Orden de Felix	600	1	10	1	c	1
1	9788401022357	El Instituto	634	2	1	2	s	2
1	9788466347921	It	1020	2	1	2	s	2
1	9788478889907	HARRY Potter y el Príncipe Mestizo	600	1	10	1	c	1
1	9788498381405	Harry Potter y las Reliquias de la Muerte	800	1	10	1	c	1
1	9788498387544	Harry Potter y el Legado Maldito	120	1	10	1	c	1
• NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

libro 1 ×

Output

Action Output

#	Time	Action	Message
34	02:37:39	DROP PROCEDURE IF EXISTS enviar_mensaje	0 row(s) affected
35	02:37:39	CREATE PROCEDURE enviar_mensaje () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No ...	0 row(s) affected
36	02:37:39	DROP TRIGGER IF EXISTS libro_casa	0 row(s) affected
37	02:37:39	CREATE TRIGGER libro_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _resul...	0 row(s) affected
38	02:37:42	USE mydb	0 row(s) affected
39	02:37:42	INSERT INTO `mydb`.`Prestamo`(`fecha_inicio`, `fecha_fin`, `usuario_id`, `empleado_seguridad_social`, `casa_...`)	Error Code: 1644. No se puede llevar este libro a su casa. Lo sentimos
40	02:38:25	SELECT * FROM libro LIMIT 0, 1000	10 row(s) returned

5.- Las revistas se prestarán solo si hay fotocopias realizadas de la misma:

USE mydb;

```

delimiter //
DROP PROCEDURE IF EXISTS enviar_mensaje_revista //
CREATE PROCEDURE enviar_mensaje_revista ()
BEGIN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede llevar
esta revista a su casa. Debe realizar las fotocopias y
llevarselas.. Lo sentimos';

```

END//

delimiter ;

```

DELIMITER $$

DROP TRIGGER IF EXISTS revista_casa $$

CREATE TRIGGER revista_casa
BEFORE INSERT ON prestamo FOR EACH ROW
BEGIN

    DECLARE _result BOOL DEFAULT FALSE;

        if ((SELECT fotocopias_realizadas FROM revista WHERE
NEW.Revista_titulo = revista.titulo) = FALSE ) THEN
            SET _RESULT=TRUE;

        END IF;

        if (_result) THEN
            CALL enviar_mensaje_revista();
        END IF;

END $$

DELIMITER ;

```

The screenshot shows the MySQL Workbench interface. The top part is the SQL editor window with the following content:

```

1 :asa_o_sala', 'Libro_ISBN', 'Revista_titulo', 'Portatil_modelo', 'Portatil_marca') VALUES ('2019-07-08', '2019-08-08', 7, '217835147669', 'C', NULL, 'Revista 2', NU
2

```

The bottom part is the Output pane titled "Action Output". It displays the following log entries:

#	Time	Action	Message	Affected Rows
1	02:52:22	DROP PROCEDURE IF EXISTS enviar_mensaje_revista	0 row(s) affected	0
2	02:52:22	CREATE PROCEDURE enviar_mensaje_revista () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =...	0 row(s) affected	0
3	02:52:23	DROP TRIGGER IF EXISTS revista_casa	0 row(s) affected	0
4	02:52:23	CREATE TRIGGER revista_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _resul...	0 row(s) affected	0
5	02:52:36	INSERT INTO `mydb`.`Prestamo`(`fecha_inicio`, `fecha_fin`, `usuario_id`, `empleado_seguridad_social`, `casa_o_...`)	Error Code: 1644. No se puede llevar esta revista a su casa. Debe realizar las fotocopias y llevarselas.. Lo sentimos	0

Esto ocurre porque la revista 2 no tiene las fotocopias realizadas.

The screenshot shows the MySQL Workbench interface. At the top, there is a query editor window with the following SQL code:

```
1 select * from revista;
```

Below the query editor is a result grid table titled "revista". The table has three columns: "recurso\_id", "titulo", and "fotocopias\_realizadas". The data is as follows:

recurso_id	titulo	fotocopias_realizadas
2	Revista 1	1
2	Revista 2	0
2	Revista 3	1
2	Revista 4	1
*	HULL	HULL

At the bottom of the interface, there is a "revista 3" tab which is currently active. It contains an "Output" section with an "Action Output" table. The table lists the following actions:

#	Time	Action	Message
1	02:52:22	DROP PROCEDURE IF EXISTS enviar_mensaje_revista	0 row(s) affected
2	02:52:22	CREATE PROCEDURE enviar_mensaje_revista () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT ...	0 row(s) affected
3	02:52:23	DROP TRIGGER IF EXISTS revista_casa	0 row(s) affected
4	02:52:23	CREATE TRIGGER revista_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _res...	0 row(s) affected
5	02:52:36	INSERT INTO 'mydb'.`Prestamo` (fecha_inicio, fecha_fin, usuario_id, empleado_segundad_social, casa_o...)	Error Code: 1644. No se puede llevar esta revista a su casa. Debe realizar las fotocopias y llevarselas.. Lo sentim...
6	02:52:57	select * from revista LIMIT 0, 1000	4 row(s) returned

6.- Los portátiles se prestarán directamente pero se deberán almacenar distintos datos al ser un objeto de mayor valor y tener más control sobre él.

USE mydb;

```
delimiter //  
DROP PROCEDURE IF EXISTS enviar_mensaje_portatil //  
CREATE PROCEDURE enviar_mensaje_portatil ()  
BEGIN  
  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Lo sentimos, no  
    podemos darle el portatil, tiene que enviarnos su DNI antes.';  
  
END//  
  
delimiter ;  
  
DELIMITER $$  
DROP TRIGGER IF EXISTS portatil_casa $$
```

```

CREATE TRIGGER portatil_casa
BEFORE INSERT ON prestamo FOR EACH ROW
BEGIN

    DECLARE _result BOOL DEFAULT TRUE;

        IF ((SELECT DNI FROM usuario WHERE NEW.usuario_id = usuario.id)) THEN
            SET _RESULT=false;

        END IF;

        if (_result) THEN
            CALL enviar_mensaje_portatil();
        END IF;

END $$

DELIMITER ;

```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a text editor window displays the SQL code for creating a trigger named 'portatil\_casa'. The code includes logic to check if the inserted user's DNI exists in the 'usuario' table and to call a stored procedure 'enviar\_mensaje\_portatil()' if it does not.

Below the text editor is an 'Output' pane. It has a header 'Action Output' with columns for #, Time, Action, and Message. The output log shows the following entries:

#	Time	Action	Message
47	03:04:31	CREATE TRIGGER portatil_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN	DECLARE _re... 0 row(s) affected
48	03:04:35	INSERT INTO `mydb`.`Prestamo`(`fecha_inicio`, `fecha_fin`, `usuario_id`, `empleado_seguridad_social`, `casa_...`)	VALUES ('2019-10-2d', '2019-11-10', 14, '439565855354', 'S', NULL, NU
49	03:05:01	DROP PROCEDURE IF EXISTS enviar_mensaje_portatil	0 row(s) affected
50	03:05:01	CREATE PROCEDURE enviar_mensaje_portatil () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TE...	0 row(s) affected
51	03:05:02	DROP TRIGGER IF EXISTS portatil_casa	0 row(s) affected
52	03:05:02	CREATE TRIGGER portatil_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN	DECLARE _re... 0 row(s) affected
53	03:05:06	INSERT INTO `mydb`.`Prestamo`(`fecha_inicio`, `fecha_fin`, `usuario_id`, `empleado_seguridad_social`, `casa_...`)	Error Code: 1644. Lo sentimos, no podemos darle el portatil, tiene que enviamos su DNI antes.

Vemos que el usuario 14 no tiene su dni establecido :

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a query editor window displays the SQL command: `select * from usuario;`. The results are shown in a "Result Grid" table:

	id	nombre	apellido	numero_copias_hechas	DNI	direccion_id
3	CANDELARIA	CONDE CALLEJA	0	999999999	5	
4	LOLA	PAVON ROVIRA	0	345346346	3	
5	BENITA	ORDÓÑEZ ARENAS	0	345345345	7	
6	EDUARD	LAFUENTE CASANOVA	10	554645565	6	
7	JOSE IGNACIO	ARCE GARRIDO	0	435435365	4	
8	ISABEL	DOMINGO LOZANO	0	567564534	3	
9	DIONISIO	PONCE VALLS	0	345345343	8	
10	MAR	OCAÑA GIL	5	574563454	9	
14	MARIA	ZAMBRANO	2	HULL	9	
	HULL	HULL	HULL	HULL	HULL	HULL

In the "Output" section, there is a "Action History" table:

#	Time	Action	Message
48	03:04:35	INSERT INTO `mydb`.`Prestamo`(`fecha_inicio`, `fecha_fin`, `usuario_id`, `empleado_seguridad_social`, `casa...`)	1 row(s) affected
49	03:05:01	DROP PROCEDURE IF EXISTS enviar_mensaje_portatil	0 row(s) affected
50	03:05:01	CREATE PROCEDURE enviar_mensaje_portatil () BEGIN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT...	0 row(s) affected
51	03:05:02	DROP TRIGGER IF EXISTS portatil_casa	0 row(s) affected
52	03:05:02	CREATE TRIGGER portatil_casa BEFORE INSERT ON prestamo FOR EACH ROW BEGIN DECLARE _re...	0 row(s) affected
53	03:05:06	INSERT INTO `mydb`.`Prestamo`(`fecha_inicio`, `fecha_fin`, `usuario_id`, `empleado_seguridad_social`, `casa...`)	Error Code: 1644. Lo sentimos, no podemos darle el portatil, tiene que enviamos su DNI antes.
54	03:05:27	select * from usuario LIMIT 0, 1000	11 row(s) returned