

INFORME CONSULTAS Y VISTAS

Actividad 5

C1-2023-QA-BD-05

Entregado A

Juanes Pineda

Presentado

Yeison Ferney Osorio Buitrago

SOFKAU

SAN JOSE DE CUCUTA

## Contenido

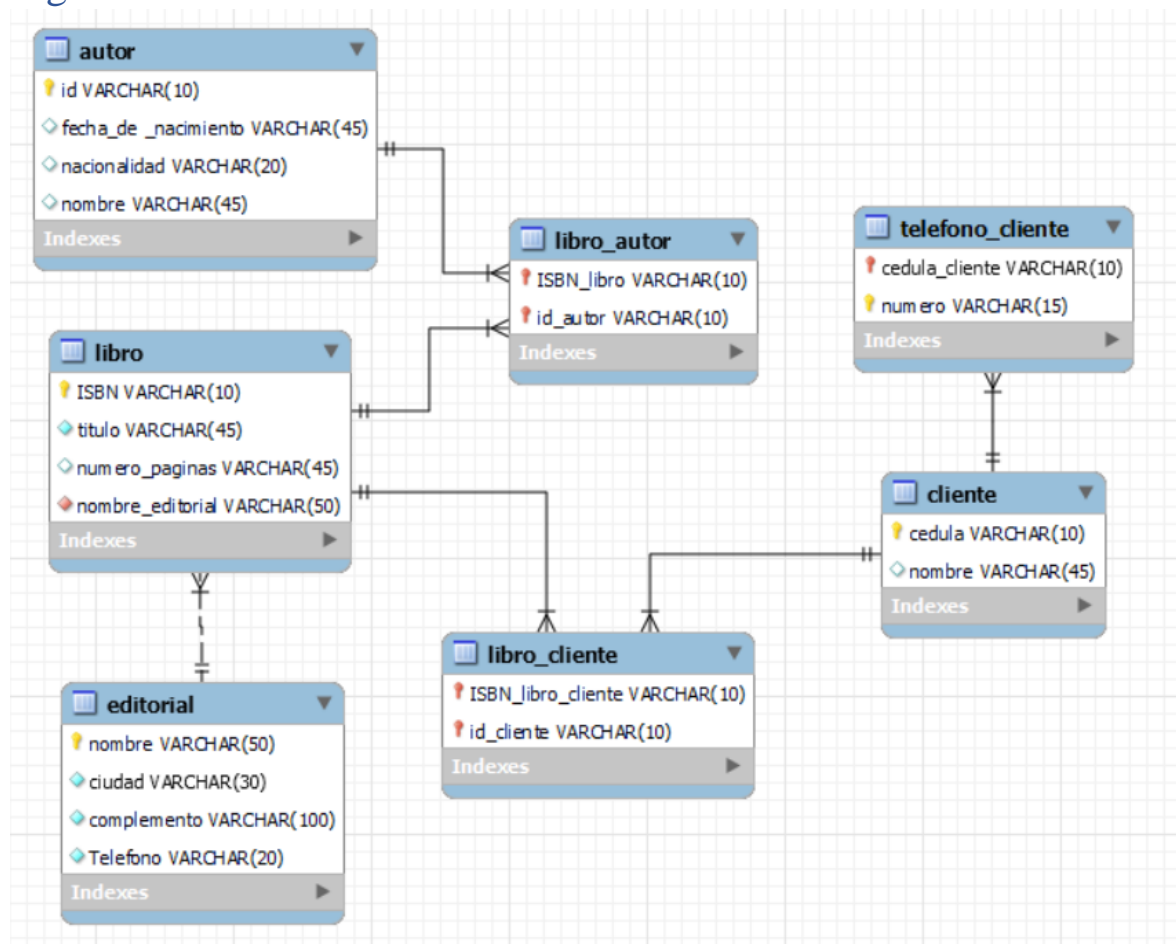
Primera actividad: Utilizando el ejercicio de la Librería realizado en clase (se adjunta script SQL) realice lo siguiente: .....	3
Figura de las Tablas Librería.....	3
Script de los datos insertados SQL a mano .....	4
Pantallazo de los datos insertados .....	6
Script de las consultas y Pantallazo.....	7
Realice las dos vistas que considere sean las más importantes y explique el motivo de su selección. ....	9
Conclusion.....	11
Actividad 2 .....	11
Utilizando el ejercicio del hospital realizado por sus compañeros realice lo siguiente: .....	11
Analizando la Normalización.....	11
Convierta el MR en una base de datos en MySQL utilizando sentencias SQL o el diagrama EER. ....	12
Registros de Datos a la Base de Datos .....	13
Script de los datos insertados a mano .....	14
Realice una consulta que me permita conocer que medicamentos a tomado cada paciente y la dosis suministrada. ....	15
Realice una consulta que me permita conocer que medicamentos a tomado cada paciente y la dosis suministrada. ....	16
Realice una consulta que me permita conocer que enfermeros estuvieron en los procedimientos de los pacientes. ....	16
Análisis de la solución de la consulta. ....	17
Realice las tres vistas que considere sean las más importantes y explique el motivo de su selección. ....	17
¿Después de realizar el trabajo responda ¿Qué le agregaría al modelo para dar más información y esa información cual sería?.....	19
Conclusion:.....	20
Tercera actividad (TALLER 6):.....	21
Elabore 4 procedimientos almacenados que me permitan agregar, actualizar, consultar y borrar, en una de las tablas de la librería .....	21
Elabore una nueva tabla llamada "control_de_cambios_librería" la cual debe contener 3 columnas (usuario, acción, fecha) y guarde utilizando 2 Triggers el nombre del usuario que agrego o elimino un registro en la tabla seleccionada en el punto anterior. ....	22
Elabore 4 procedimientos almacenados que me permitan agregar, actualizar, consultar y borrar, en una de las tablas del Hospital .....	22
Elabore una nueva tabla llamada "control_de_cambios_hospital" la cual debe contener 3 columnas (usuario, acción, fecha) y guarde utilizando 2 Triggers el nombre del usuario que agrego o elimino un registro en la tabla seleccionada en el punto anterior. ....	23
Documentos Anexos .....	24

## Consultas y Vistas

“Cada día trae una nueva oportunidad”

Primera actividad: Utilizando el ejercicio de la Librería realizado en clase (se adjunta script SQL) realice lo siguiente:

Figura de las Tablas Librería



Complete la información para las tablas autor, libro, cliente, editorial, libro\_cliente, libro\_autor y teléfono\_cliente con al menos (5,20,7,4,10,10, 12) registros respectivamente usando únicamente comandos SQL creados por usted.

## Script de los datos insertados SQL a mano

```
1 • use libreriaescalibre;
2
3 • select * from autor;
4
5
6 • INSERT INTO autor (id, `fecha de nacimiento`, nacionalidad, nombre) VALUES
7   (1, '1947-09-21', 'American', 'Stephen King'),
8   (2, '1965-07-31', 'Colombia', 'J.K. Rowling'),
9   (3, '1903-06-25', 'EEUU', 'George Orwell'),
10  (4, '1896-09-24', 'España', 'Gabriel Garcia Marquez'),
11  (5, '1926-04-28', 'American', 'Harper Lee');
12
13 • INSERT INTO editoriales (nombre, ciudad, complemento, telefono) VALUES
14  ('editorial1', 'Londres', 'complemento2', '020 7631'),
15  ('editorial2', 'Colombia', 'complemento3', '212 366'),
16  ('editorial3', 'Nueva York', 'complemento4', '212 632'),
17  ('editorial4', 'londres', 'complemento1', '020 7494');
18
19 • INSERT INTO libro (ISBN, titulo, numero_paginas, nombre_editorial) VALUES
20  ('ISBN001', 'Cien años de soledad', 300, 'editorial1'),
21  ('ISBN002', 'Matar a un ruiseñor', 250, 'editorial2'),
22  ('ISBN003', '1984', 200, 'editorial3'),
23  ('ISBN004', 'El gran Gatsby', 350, 'editorial4'),
24  ('ISBN005', 'Crimen y castigo', 400, 'editorial1'),
25  ('ISBN006', 'La metamorfosis', 300, 'editorial2'),
26  ('ISBN007', 'El extranjero', 250, 'editorial3'),
27  ('ISBN008', 'Momo', 200, 'editorial4'),
28  ('ISBN009', 'La isla del tesoro', 300, 'editorial1'),
29  ('ISBN010', 'La isla de los hombres solos', 250, 'editorial2'),
30  ('ISBN011', 'La divina comedia', 400, 'editorial3'),
31  ('ISBN012', 'Don Quijote de la Mancha', 250, 'editorial4'),
32  ('ISBN013', 'El diario de Ana Frank', 300, 'editorial1'),
33  ('ISBN014', 'El retrato de Dorian Gray', 200, 'editorial2'),
34  ('ISBN015', 'Los tres mosqueteros', 250, 'editorial3'),
35  ('ISBN016', 'El conde de Montecristo', 300, 'editorial4'),
36  ('ISBN017', 'La odisea', 200, 'editorial1'),
37  ('ISBN018', 'Frankenstein', 250, 'editorial2'),
38  ('ISBN019', 'Dracula', 300, 'editorial3'),
39  ('ISBN020', 'La Guerra y la Paz', 400, 'editorial4');
--
```

```

43 • INSERT INTO cliente (cedula, nombre) VALUES
44 ('001', 'Juan Pérez'),
45 ('002', 'María González'),
46 ('003', 'Pedro Rodríguez'),
47 ('004', 'Ana García'),
48 ('005', 'Luis Sánchez'),
49 ('006', 'Isabel Martínez'),
50 ('007', 'José López');
51
52 • INSERT INTO libro_cliente (ISBN_libro_cliente, id_cliente) VALUES
53 ('ISBN001', '001'),
54 ('ISBN002', '002'),
55 ('ISBN003', '003'),
56 ('ISBN004', '004'),
57 ('ISBN005', '005'),
58 ('ISBN006', '006'),
59 ('ISBN007', '007'),
60 ('ISBN008', '001'),
61 ('ISBN009', '002'),
62 ('ISBN010', '003');

65 • INSERT INTO libro_autor (ISBN_libro, id_autor) VALUES
66 ('ISBN001', 1),
67 ('ISBN002', 2),
68 ('ISBN003', 3),
69 ('ISBN004', 4),
70 ('ISBN005', 1),
71 ('ISBN006', 4),
72 ('ISBN007', 1),
73 ('ISBN008', 4),
74 ('ISBN009', 2),
75 ('ISBN010', 4),
76 ('ISBN011', 3),
77 ('ISBN012', 2),
78 ('ISBN013', 1),
79 ('ISBN014', 4),
80 ('ISBN015', 1),
81 ('ISBN016', 3),
82 ('ISBN017', 2),
83 ('ISBN018', 2),
84 ('ISBN019', 1),
85 ('ISBN020', 3);
86
87 • INSERT INTO telefono_cliente (cedula_cliente, numero) VALUES
88 ('001', '587514'),
89 ('001', '558451'),
90 ('002', '558451'),
91 ('003', '584155'),
92 ('004', '515151'),
93 ('005', '148485'),
94 ('006', '515415'),
95 ('007', '561588'),
96 ('007', '554515'),
97 ('006', '522560'),
98 ('005', '555461'),
99 ('004', '571162');

```

## Pantallazo de los datos insertados

### Tabla Libro

```
1 • SELECT * FROM libreriabuscalibre.libro;
```

ISBN	titulo	numero_paginas	nombre_editorial
ISBN001	Cien años de soledad	300	editorial1
ISBN002	Matar a un ruiseñor	250	editorial2
ISBN003	1984	200	editorial3
ISBN004	El gran Gatsby	350	editorial4
ISBN005	Crimen y castigo	400	editorial1
ISBN006	La metamorfosis	300	editorial2
ISBN007	El extranjero	250	editorial3
ISBN008	Momo	200	editorial4
ISBN009	La isla del tesoro	300	editorial1
ISBN010	La isla de los hombres solos	250	editorial2
ISBN011	La divina comedia	400	editorial3
ISBN012	Don Quijote de la Mancha	250	editorial4
ISBN013	El diario de Ana Frank	300	editorial1
ISBN014	El retrato de Dorian Gray	200	editorial2
ISBN015	Los tres mosqueteros	250	editorial3
ISBN016	El conde de Montecristo	300	editorial4
ISBN017	La odisea	200	editorial1
ISBN018	Frankenstein	250	editorial2
ISBN019	Dracula	300	editorial3
ISBN020	La Guerra y la Paz	400	editorial4
NULL	NULL	NULL	NULL

### Tabla Editorial

```
1 • SELECT * FROM libreriabuscalibre.editorial;
```

nombre	ciudad	complemento	Telefono
editorial1	Londres	complemento2	020 7631
editorial2	Colombia	complemento3	212 366
editorial3	Nueva York	complemento4	212 632
editorial4	Londres	complemento1	020 7494
NULL	NULL	NULL	NULL

### Tabla Cliente

```
1 • SELECT * FROM libreriabuscalibre.cliente;
```

cedula	nombre
001	Juan Pérez
002	María González
003	Pedro Rodríguez
004	Ana García
005	Luis Sánchez
006	Isabel Martínez
007	José López
NULL	NULL

**Nota: se tomaron algunos pantallazos para no llenar el documento, los demás están en la base de datos**

Realice 5 consultas que me permitan conocer el nombre y la fecha de nacimiento de cada escritor, la cantidad de libros diferentes vendidos, el nombre de su cliente acompañado de su número telefónico, el nombre del libro acompañado por su autor o sus autores, el nombre de las editoriales que han logrado vender libros.

## Script de las consultas y Pantallazo

use libreriabuscalibre;

1. nombre y la fecha de nacimiento de cada escritor

*select nombre, 'fecha de nacimiento' from autor;*

```

1 • use libreriabuscalibre;
2
3 -- nombre y la fecha de nacimiento de cada escritor
4 • select nombre, 'fecha de nacimiento' from autor;
5

```

nombre	fecha de nacimiento
Stephen King	fecha de nacimiento
J.K. Rowling	fecha de nacimiento
George Orwell	fecha de nacimiento
Gabriel Garcia Marquez	fecha de nacimiento
Harper Lee	fecha de nacimiento

2. la cantidad de libros diferentes vendidos

***SELECT libro.titulo, COUNT(DISTINCT libro\_cliente.ISBN\_libro\_cliente) as Total\_libros\_vendidos  
FROM libro\_cliente  
JOIN libro  
ON libro\_cliente.ISBN\_libro\_cliente = libro.ISBN  
GROUP BY libro.titulo;***

```

6 -- la cantidad de libros diferentes vendidos,
7 • SELECT libro.titulo, COUNT(DISTINCT libro_cliente.ISBN_libro_cliente) as Total_libros_vendidos
8 FROM libro_cliente
9 JOIN libro
10 ON libro_cliente.ISBN_libro_cliente = libro.ISBN
11 GROUP BY libro.titulo;
12

```

titulo	Total_libros_vendidos
1984	1
Cien años de soledad	1
Crimen y castigo	1
El extranjero	1
El gran Gatsby	1
La isla de los hombres solos	1
La isla del tesoro	1
La metamorfosis	1
Matar a un ruiseñor	1
Momo	1

3. el nombre de su cliente acompañado de su número telefónico

```
SELECT cliente.nombre, telefono_cliente.numero
FROM cliente
JOIN telefono_cliente
ON cliente.cedula = telefono_cliente.cedula_cliente;
```

```
14 -- el nombre de su cliente acompañado de su numero telefónico
15 • SELECT cliente.nombre, telefono_cliente.numero
16 FROM cliente
17 JOIN telefono_cliente
18 ON cliente.cedula = telefono_cliente.cedula_cliente;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

nombre	numero
Juan Pérez	558451
Juan Pérez	587514
María González	558451
Pedro Rodríguez	584155
Ana García	515151
Ana García	571162
Luis Sánchez	148485
Luis Sánchez	555461
Isabel Martínez	515415
Isabel Martínez	522560
José López	554515
José López	561588

4. el nombre del libro acompañado por su autor o sus autores

```
SELECT libro.titulo AS libro, GROUP_CONCAT(autor.nombre SEPARATOR ', ') AS autores
FROM libro
JOIN libro_autor ON libro.ISBN = libro_autor.ISBN_libro
JOIN autor ON libro_autor.id_autor = autor.id
GROUP BY libro.titulo;
```

```
20 -- el nombre del libro acompañado por su autor o sus autores
21 • SELECT libro.titulo AS libro, GROUP_CONCAT(autor.nombre SEPARATOR ', ') AS autores
22 FROM libro
23 JOIN libro_autor ON libro.ISBN = libro_autor.ISBN_libro
24 JOIN autor ON libro_autor.id_autor = autor.id
25 GROUP BY libro.titulo;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

libro	autores
1984	George Orwell
Cien años de soledad	Stephen King
Crimen y castigo	Stephen King
Don Quijote de la Mancha	J.K. Rowling
Dracula	Stephen King
El conde de Montecristo	George Orwell
El diario de Ana Frank	Stephen King
El extranjero	Stephen King
El gran Gatsby	Gabriel Garcia Marquez
El retrato de Dorian Gray	Gabriel Garcia Marquez

5. el nombre de las editoriales que han logrado vender libros.



```

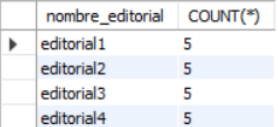
SELECT nombre_editorial, COUNT(*)
FROM libro
GROUP BY nombre_editorial
HAVING COUNT(*) > 0;

```

```

27  -- el nombre de las editoriales que han logrado vender libros.
28  •  SELECT nombre_editorial, COUNT(*)
29      FROM libro
30      GROUP BY nombre_editorial
31      HAVING COUNT(*) > 0;
32

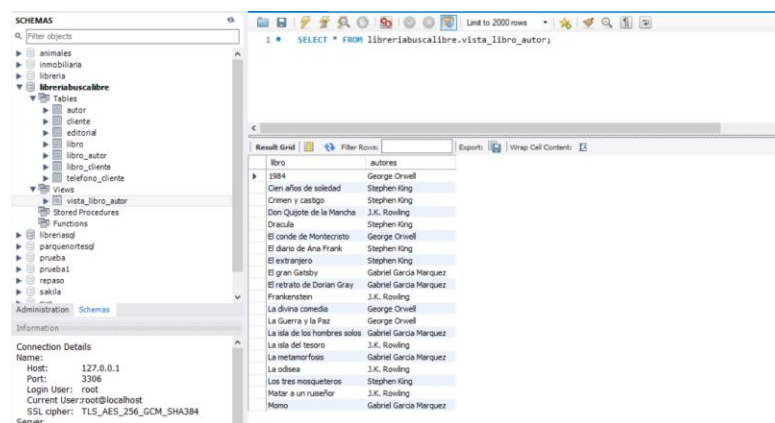
```



nombre_editorial	COUNT(*)
editorial1	5
editorial2	5
editorial3	5
editorial4	5

Realice las dos vistas que considere sean las más importantes y explique el motivo de su selección.

## Vista 1



```

1  •  SELECT * FROM libreria.buscablibre.vista_libro_autor;

```

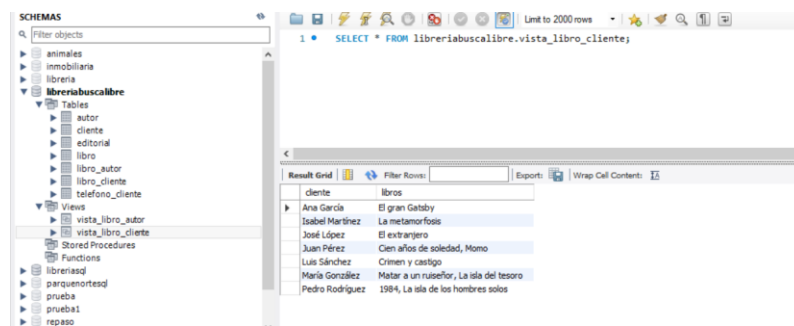
libro	autores
1984	George Orwell
Cien años de soledad	Stephen King
Crimen y castigo	Stephen King
Don Quijote de la Mancha	J.K. Rowling
Dracula	Stephen King
El conde de Montecristo	George Orwell
El diario de Ana Frank	Stephen King
El extranjero	Stephen King
El gran Gatsby	Gabriel Garcia Marquez
El retrato de Dorian Gray	Gabriel Garcia Marquez
Frankenstein	J.K. Rowling
La divina comedia	George Orwell
La Guerra y la Paz	George Orwell
La isla de los hombres solos	Gabriel Garcia Marquez
La isla del tesoro	J.K. Rowling
La metamorfosis	Gabriel Garcia Marquez
La odisea	J.K. Rowling
Los tres mosqueteros	Stephen King
Matar a un ruiseñor	J.K. Rowling
Nomo	Gabriel Garcia Marquez

```
CREATE VIEW vista_libro_autor AS
SELECT libro.titulo AS libro, GROUP_CONCAT(autor.nombre SEPARATOR ', ') AS autores
FROM libro
JOIN libro_autor ON libro.ISBN = libro_autor.ISBN_libro
JOIN autor ON libro_autor.id_autor = autor.id
GROUP BY libro.titulo;
```

La vista es importante porque me permite ver de manera clara la información de tres tablas, los **libro** y sus **autores** en una sola tabla. En lugar de tener que revisar la tabla libro y la tabla **libro autor** para encontrar toda la información que necesitas sobre un libro y sus autores, puedes usar la **vista vista libro autor** y tener toda la información en un solo lugar.

Además, esta consulta utiliza la función GROUP\_CONCAT para agrupar los nombres de los autores en una sola fila, separados por comas. Esto te permite tener una representación más clara.

## Vista 2



cliente	libros
Ana García	El gran Gatsby
Isabel Martínez	La metamorfosis
José López	El extranjero
Juan Pérez	Cien años de soledad, Momo
Luis Sánchez	Crimen y castigo
Maria González	Matar a un ruiseñor, La isla del tesoro
Pedro Rodríguez	1984, La isla de los hombres solos

```
CREATE VIEW vista_libro_cliente AS
SELECT cliente.nombre AS cliente, GROUP_CONCAT(libro.titulo SEPARATOR ', ') AS libros
FROM cliente
JOIN libro_cliente ON cliente.cedula = libro_cliente.id_cliente
JOIN libro ON libro_cliente.ISBN_libro_cliente = libro.ISBN
GROUP BY cliente.nombre;
```

Esta vista es importante porque combina información de tres tablas diferentes en un solo resultado, lo que puede ser útil para tener una visión más completa.

La consulta combina información de la **tabla cliente, libro\_cliente** y **libro**, utilizando los joins para unir las tablas en función de las claves primarias y extranjeras correspondientes. La cláusula "GROUP BY" agrupa los resultados por el nombre del cliente, y la función "GROUP\_CONCAT" combina los títulos de los libros en una sola columna, separándolos con una coma y un espacio.

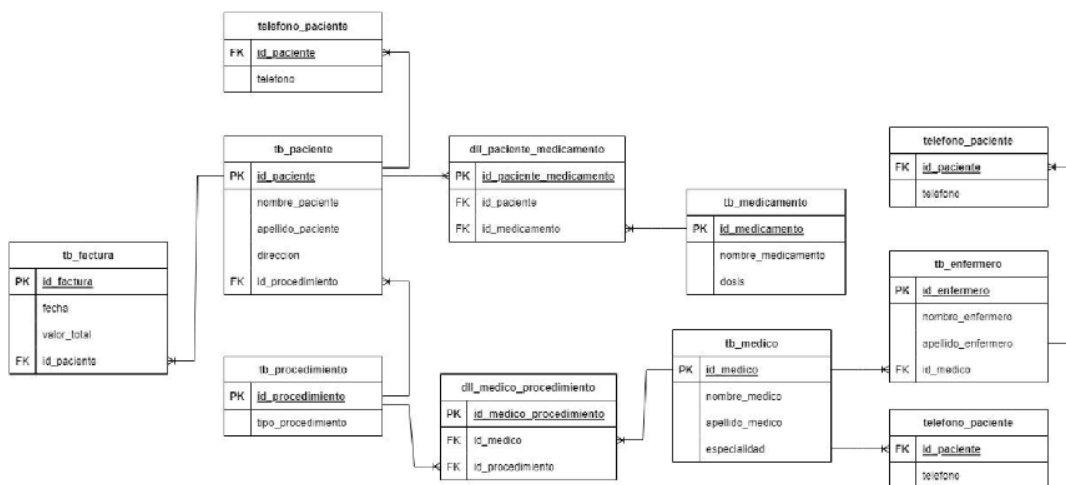
De esta manera, se puede obtener una lista de todos los clientes y los libros que han comprado, tomando como premisa que los libros son vendidos en la librería, para hacer un seguimiento de los clientes y sus preferencias de lectura.

## Conclusion

Las dos vistas creadas nos permiten ver el Core del negocio librería ya que integran la mayor cantidad de datos mas relevantes de la base de datos, como lo son los autores y sus libros correspondientes y los clientes que han alquilado un libro.

## Actividad 2

Utilizando el ejercicio del hospital realizado por sus compañeros realice lo siguiente:



## Analizando la Normalización

### Primera Forma Normal 1FN

- ✓ Para cumplir con la función debimos crear tabla para los valores multivaluados y de esta manera cumplir con los requerimientos.
- ✓ No deben existir registros duplicado.
- ✓ Todas las tablas tienen llave primaria

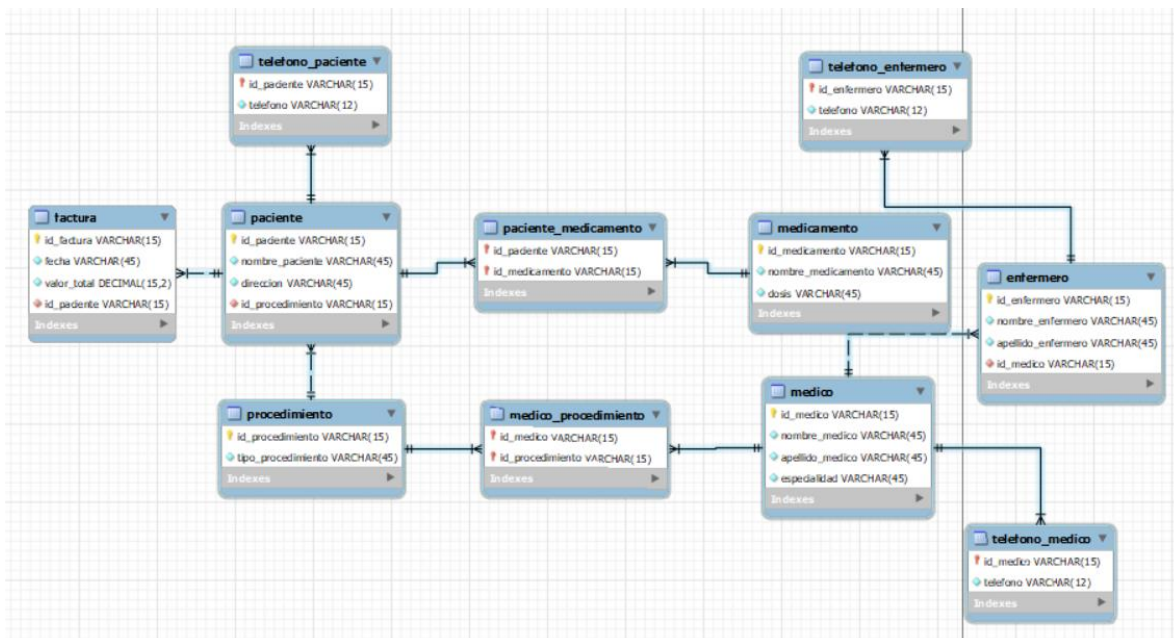
Se cumple con la 2FN.

- ✓ Cumple con 1FN
- ✓ Todos los valores de las columnas dependen únicamente de la llave primaria de la tabla.
- ✓ Las tablas tienen una única llave primaria que identifica a la tabla y sus atributos dependen de ella. Esto significa que, si se conoce la llave primaria, se pueden determinar todos los demás valores en la tabla.

Se cumple con la 3FN.

- ✓ Cumple en 2FN
- ✓ Cada atributo que no está incluido en la clave primaria no depende de la clave primaria. Esto significa que un atributo no debe depender de otro atributo que no forme parte de la clave primaria.

Convierta el MR en una base de datos en MySQL utilizando sentencias SQL o el diagrama EER.

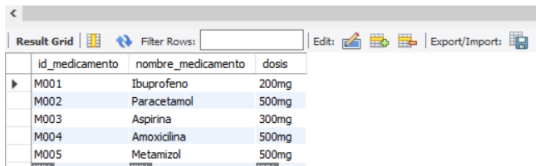


Complete la información para las tablas realizadas con al menos 5 registros por tabla. realice una consulta que me permita conocer que medicamentos a tomado cada paciente y la dosis suministrada.

## Registros de Datos a la Base de Datos

### Tabla Medicamento

```
1 • SELECT * FROM hospital.medimento;
```

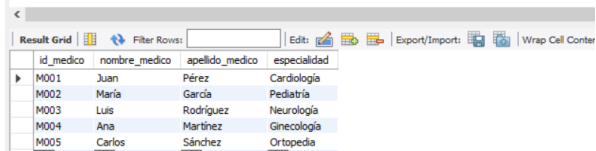


The screenshot shows a database query result grid for the 'Tabla Medicamento' table. The grid has three columns: 'id\_medimento', 'nombre\_medimento', and 'dosis'. There are five rows of data, each with a blue selection bar on the left. The interface includes a 'Filter Rows' field, an 'Edit' button, and an 'Export/Imports' button.

id_medimento	nombre_medimento	dosis
M001	Ibuprofeno	200mg
M002	Paracetamol	500mg
M003	Aspirina	300mg
M004	Amoxicilina	500mg
M005	Metamizol	500mg

### Tabla Medico

```
1 • SELECT * FROM hospital.medico;
```

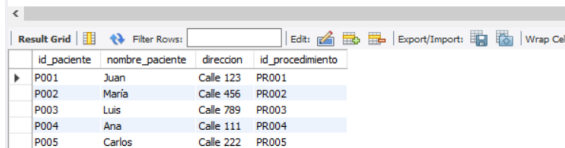


The screenshot shows a database query result grid for the 'Tabla Medico' table. The grid has four columns: 'id\_medico', 'nombre\_medico', 'apellido\_medico', and 'especialidad'. There are five rows of data, each with a blue selection bar on the left. The interface includes a 'Filter Rows' field, an 'Edit' button, an 'Export/Imports' button, and a 'Wrap Cell Content' button.

id_medico	nombre_medico	apellido_medico	especialidad
M001	Juan	Pérez	Cardiología
M002	María	García	Pediatría
M003	Luis	Rodríguez	Neurología
M004	Ana	Martínez	Ginecología
M005	Carlos	Sánchez	Ortopedia

### Tabla Paciente

```
1 • SELECT * FROM hospital.paciente;
```

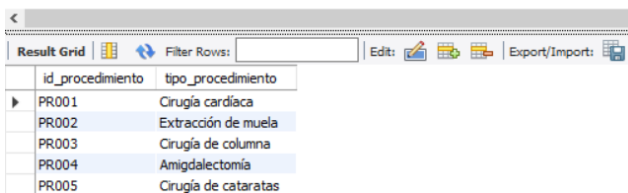


The screenshot shows a database query result grid for the 'Tabla Paciente' table. The grid has four columns: 'id\_paciente', 'nombre\_paciente', 'direccion', and 'id\_procedimiento'. There are five rows of data, each with a blue selection bar on the left. The interface includes a 'Filter Rows' field, an 'Edit' button, an 'Export/Imports' button, and a 'Wrap Cell Content' button.

id_paciente	nombre_paciente	direccion	id_procedimiento
P001	Juan	Calle 123	PR001
P002	María	Calle 456	PR002
P003	Luis	Calle 789	PR003
P004	Ana	Calle 111	PR004
P005	Carlos	Calle 222	PR005

### Tabla Procedimiento

```
1 • SELECT * FROM hospital.procedimiento;
```



The screenshot shows a database query result grid for the 'Tabla Procedimiento' table. The grid has two columns: 'id\_procedimiento' and 'tipo\_procedimiento'. There are five rows of data, each with a blue selection bar on the left. The interface includes a 'Filter Rows' field, an 'Edit' button, and an 'Export/Imports' button.

id_procedimiento	tipo_procedimiento
PR001	Cirugía cardíaca
PR002	Extracción de muela
PR003	Cirugía de columna
PR004	Amigdalectomía
PR005	Cirugía de cataratas

**Nota: se tomaron algunos pantallazos para no llenar el documento, los demás están en la base de datos**

## Script de los datos insertados a mano

```
1 • use hospital;
2
3
4 • INSERT INTO enfermero (id_enfermero, nombre_enfermero, apellido_enfermero, id_medico)
5 VALUES
6 ("E001", "Juan", "Pérez", "M001"),
7 ("E002", "María", "García", "M002"),
8 ("E003", "Luis", "Rodríguez", "M003"),
9 ("E004", "Ana", "Martínez", "M004"),
10 ("E005", "Carlos", "Sánchez", "M005");
11
12 • INSERT INTO medico (id_medico, nombre_medico, apellido_medico, especialidad)
13 VALUES
14 ("M001", "Juan", "Pérez", "Cardiología"),
15 ("M002", "María", "García", "Pediatría"),
16 ("M003", "Luis", "Rodríguez", "Neurología"),
17 ("M004", "Ana", "Martínez", "Ginecología"),
18 ("M005", "Carlos", "Sánchez", "Ortopedia");
19
20 • INSERT INTO telefono_medico (id_medico, telefono)
21 VALUES
22 ("M001", "5551"),
23 ("M002", "5552"),
24 ("M003", "5553"),
25 ("M004", "5554"),
26 ("M005", "5555");
27
28
29 • INSERT INTO telefono_enfermero (id_enfermero, telefono)
30 VALUES
31 ("E001", "1551"),
32 ("E002", "2552"),
33 ("E003", "3553"),
34 ("E004", "4554"),
35 ("E005", "5555");
36
37
38 • INSERT INTO telefono_paciente (id_paciente, telefono)
39 VALUES
40 ("P001", "15510"),
41 ("P002", "25520"),
42 ("P003", "35530"),
43 ("P004", "45540"),
44 ("P005", "55550");
45
46 • INSERT INTO medicamento (id_medicamento, nombre_medicamento, dosis)
47 VALUES
48 ("M001", "Ibuprofeno", "200mg"),
49 ("M002", "Paracetamol", "500mg"),
50 ("M003", "Aspirina", "300mg"),
51 ("M004", "Amoxicilina", "500mg"),
52 ("M005", "Metamizol", "500mg");
--
```

```

54 • INSERT INTO paciente (id_paciente, nombre_paciente, direccion, id_procedimiento)
55 VALUES
56 ("P001", "Juan", "Calle 123", "PR001"),
57 ("P002", "María", "Calle 456", "PR002"),
58 ("P003", "Luis", "Calle 789", "PR003"),
59 ("P004", "Ana", "Calle 111", "PR004"),
60 ("P005", "Carlos", "Calle 222", "PR005");
61
62
63 • INSERT INTO paciente_medicamento (id_paciente, id_medimento)
64 VALUES
65 ("P001", "M001"),
66 ("P002", "M002"),
67 ("P003", "M003"),
68 ("P004", "M004"),
69 ("P005", "M005");
70
71 • INSERT INTO procedimiento (id_procedimiento, tipo_procedimiento)
72 VALUES
73 ("PR001", "Cirugía cardíaca"),
74 ("PR002", "Extracción de muela"),
75 ("PR003", "Cirugía de columna"),
76 ("PR004", "Amigdalectomía"),
77 ("PR005", "Cirugía de cataratas");
78
79 • INSERT INTO medico (id_medico, nombre_medico, apellido_medico, especialidad)
80 VALUES
81 ("M001", "Juan", "Pérez", "Cardiología"),
82 ("M002", "María", "García", "Pediatría"),
83 ("M003", "Luis", "Rodríguez", "Neurología"),
84 ("M004", "Ana", "Martínez", "Ginecología"),
85 ("M005", "Carlos", "Sánchez", "Ortopedia");

88 • INSERT INTO medico_procedimiento (id_medico, id_procedimiento)
89 VALUES
90 ("M001", "PR001"),
91 ("M002", "PR002"),
92 ("M003", "PR003"),
93 ("M004", "PR004"),
94 ("M005", "PR005");
95
96 • INSERT INTO factura (id_factura, fecha, valor_total, id_paciente)
97 VALUES
98 ("F001", "13/01/2023", "500000", "P001"),
99 ("F002", "11/02/2023", "1000000", "P002"),
100 ("F003", "18/05/2023", "80000", "P003"),
101 ("F004", "21/04/2023", "30000", "P004"),
102 ("F005", "30/09/2023", "3005000", "P005")

```

Realice una consulta que me permita conocer que medicamentos a tomado cada paciente y la dosis suministrada.

```
--
52 • SELECT paciente.nombre_paciente, medicamento.nombre_medimento, medicamento.dosis
53 FROM paciente
54 INNER JOIN paciente_medimento ON paciente.id_paciente = paciente_medimento.id_paciente
55 INNER JOIN medicamento ON paciente_medimento.id_medimento= medicamento.id_medimento;
```

nombre_paciente	nombre_medimento	dosis
pepe	Ibuprofeno	200mg
juanito	Paracetamol	500mg
brayam	Aspirina	300mg
karen	Amoxicilina	500mg
armando	Metamizol	500mg

Realice una consulta que me permita conocer que medicamentos a tomado cada paciente y la dosis suministrada.

```
SELECT paciente.nombre_paciente, medicamento.nombre_medimento, medicamento.dosis
FROM paciente
INNER JOIN paciente_medimento ON paciente.id_paciente = paciente_medimento.id_paciente
INNER JOIN medicamento ON paciente_medimento.id_medimento= medicamento.id_medimento;
```

Realice una consulta que me permita conocer que enfermeros estuvieron en los procedimientos de los pacientes.

```
17
18
19 • USE hospital;
20 -- consulta que me permita conocer que enfermeros estuvieron en los procedimientos de los pacientes.
21 • SELECT enfermero.nombre_enfermero, paciente.nombre_paciente
22 FROM enfermero
23 INNER JOIN medico ON enfermero.id_medico = medico.id_medico
24 INNER JOIN medico_procedimiento ON medico.id_medico=medico_procedimiento.id_medico
25 INNER JOIN procedimiento ON medico_procedimiento.id_procedimiento=procedimiento.id_procedimiento
26 INNER JOIN paciente ON procedimiento.id_procedimiento=paciente.id_procedimiento
```

nombre_enfermero	nombre_paciente
yeison	pepe
Maria	juanito
diego	brayam
yulpza	karen
Carlos	armando

```
SELECT enfermero.nombre_enfermero, paciente.nombre_paciente
FROM enfermero
INNER JOIN medico ON enfermero.id_medico = medico.id_medico
INNER JOIN medico_procedimiento ON medico.id_medico=medico_procedimiento.id_medico
INNER JOIN procedimiento ON medico_procedimiento.id_procedimiento=procedimiento.id_procedimiento
INNER JOIN paciente ON procedimiento.id_procedimiento=paciente.id_procedimiento
```



## Análisis de la solución de la consulta.

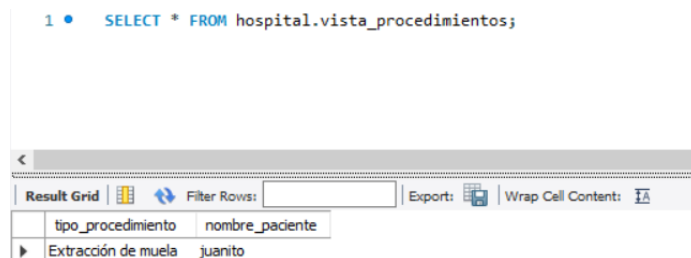
Primero analice como seria el camino de la consulta empezando por la tabla **enfermero** que seria mi tabla principal, donde paso a **medico** comparando la llave foránea de enfermero `id_medico` la comparo `id_medico` y luego al **procedimiento** pasando por la tabla intermedia **medico procedimiento** y por último a la tabla **paciente** para saber que paciente fue atendido por el enfermero

Realice las tres vistas que considere sean las más importantes y explique el motivo de su selección.

### Vista 1

- ✓ Una vista del paciente y que procedimiento le están realizando probando con los alias para practicarlos y saberlos implementar

```
CREATE VIEW vista_procedimientos AS
SELECT p.tipo_procedimiento, pa.nombre_paciente
FROM procedimiento p
INNER JOIN paciente pa ON p.id_procedimiento = pa.id_procedimiento
WHERE pa.id_paciente = "P002";
```



```
1 • SELECT * FROM hospital.vista_procedimientos;
```

tipo_procedimiento	nombre_paciente
Extracción de muela	juanito

### Vista 2

- ✓ consulta que me permita conocer que enfermeros estuvieron en los procedimientos de los pacientes. Me pareció pertinente poner en una vista una consulta tan completa donde agrupa 5 tablas para conocer el enfermero que acompaña a un medico en un procedimiento que con un cambio podríamos ver el medico que trabajo con ese enfermero

```

CREATE VIEW nombre_enfermo_que_atendio_a_un_pacinte AS
SELECT enfermero.nombre_enfermero, paciente.nombre_paciente
FROM enfermero
INNER JOIN medico ON enfermero.id_medico = medico.id_medico
INNER JOIN medico_procedimiento ON medico.id_medico=medico_procedimiento.id_medico
INNER JOIN procedimiento ON
medico_procedimiento.id_procedimiento=procedimiento.id_procedimiento
INNER JOIN paciente ON procedimiento.id_procedimiento=paciente.id_procedimiento;

```

1 • **SELECT \* FROM hospital.nombre\_enfermo\_que\_atendio\_a\_un\_pacinte;**

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

	nombre_enfermero	nombre_paciente
▶	yeison	pepe
	María	juanito
	diego	brayam
	yulipza	karen
	Carlos	armando

- ✓ Consulta me trae en enfermero que acompaño a un médico en el tratamiento a un paciente, variación del anterior

```

SELECT enfermero.nombre_enfermero, paciente.nombre_paciente, medico.nombre_medico
FROM enfermero
INNER JOIN medico ON enfermero.id_medico = medico.id_medico
INNER JOIN medico_procedimiento ON medico.id_medico=medico_procedimiento.id_medico
INNER JOIN procedimiento ON
medico_procedimiento.id_procedimiento=procedimiento.id_procedimiento
INNER JOIN paciente ON procedimiento.id_procedimiento=paciente.id_procedimiento;
Cualquier duda en la documentación remitirse

```

1 • **SELECT \* FROM hospital.vista\_medico\_tratamiento;**

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

	nombre_medico	nombre_enfermero	nombre_paciente
▶	salvador	yeison	pepe
	carme	María	juanito
	yulieth	diego	brayam
	zulay	yulipza	karen
	gabriel	Carlos	armando

### Vista 3

- ✓ Consulta que me permite ver el paciente y su enfermero y que le receto su medico y la dosis del medicamento cerrando con la gran consulta donde se pasa por cada tabla de la base de datos, una consulta muy completa donde le puedo poner ciertos datos más para ver los números de teléfonos y demás adornos que no son necesarios.

```

CREATE VIEW vista_procedimientos_nombre_medico_receta_medica AS
SELECT medico.nombre_medico, enfermero.nombre_enfermero, paciente.nombre_paciente,
medicamento.nombre_medicamento, medicamento.dosis
FROM enfermero
INNER JOIN medico ON enfermero.id_medico = medico.id_medico
INNER JOIN medico_procedimiento ON medico.id_medico=medico_procedimiento.id_medico
INNER JOIN procedimiento ON
medico_procedimiento.id_procedimiento=procedimiento.id_procedimiento
INNER JOIN paciente ON procedimiento.id_procedimiento=paciente.id_procedimiento
INNER JOIN paciente_medicamento ON paciente.id_paciente = paciente_medicamento.id_paciente
INNER JOIN medicamento ON paciente_medicamento.id_medicamento = medicamento.id_medicamento

```

```

1 • SELECT * FROM hospital.vista_procedimientos_nombre_medico_receta_medica;

```

nombre_medico	nombre_enfermero	nombre_paciente	nombre_medicamento	dosis
salvador	yeison	pepe	Ibuprofeno	200mg
carne	María	juanito	Paracetamol	500mg
yulieith	diego	brayam	Aspirina	300mg
zulay	yulipza	karen	Amoxicilina	500mg
gabriel	Carlos	armando	Metamizol	500mg

¿Después de realizar el trabajo responda ¿Qué le agregaría al modelo para dar más información y esa información cual sería?

## Modelo Hospital

Agregar una tabla para almacenar información sobre los resultados de los procedimientos, como los valores de las pruebas de laboratorio o los registros de los resultados, exámenes de todo tipo como imágenes médicas.

Agregar una tabla para almacenar información sobre los seguros eps de los pacientes, como el tipo de seguro, si pago o contributivo el número del seguro y la compañía de seguros y en base a esa eps tendría unos beneficios.

Al agregar estas tablas y relaciones, se podrían realizar consultas más avanzadas que brinden información más detallada sobre los pacientes y su historial del procedimiento, lo que podría ayudar a los médicos a brindar una atención más efectiva y personalizada.

## Modelo Librería

Tabla **Género** Esta tabla podría contener los diferentes géneros de los libros, lo que podría ayudar a los clientes a buscar libros que se ajusten a su género. La tabla podría tener los atributos: id\_genero PRIMARY KEY, nombre del genero

Tabla **Descuento** Esta tabla podría contener información sobre los descuentos que se ofrecen en la librería. La tabla podría tener los siguientes campos: id descuento PRIMARY KEY, " nombre del descuento, el porcentaje de descuento, la fecha en que comienza el descuento, y la fecha en que termina el descuento.

Pienso que estas dos tablas podrían agregar más información respecto a la librería tomando él cuenta el género de los libros y los descuento que son realizados en la librería con sus respectivos atributos que enriquecen las consultas sql

## Conclusion:

Las consultas en SQL son una herramienta importantísima para la extracción de información valiosa a partir de bases de datos. Aprendí que una consulta consta de varias partes, desde la selección de las tablas a las que se desea acceder, la definición de las condiciones de búsqueda, y la unión de las tablas mediante el uso de comandos JOIN. También aprendí que los alias son una forma conveniente de abreviar los nombres de las tablas y los campos, lo que permite una mayor claridad y legibilidad del código.

## Tercera actividad (TALLER 6):

Elabore 4 procedimientos almacenados que me permitan agregar, actualizar, consultar y borrar, en una de las tablas de la librería (primera actividad).

Archivos anexos en el repositorio

```
1 • use libreriaabuscalibre;
2
3 -- PROCEDIMIENTO PARA AGREGAR AUTOR
4 DELIMITER //
5 • CREATE PROCEDURE agregar_autor(IN id_autor VARCHAR(10), fecha_nacimiento VARCHAR(45), IN nacionalidad_autor VARCHAR(20), IN nombre_autor VARCHAR(45))
6 BEGIN
7   INSERT INTO autor (id, `fecha de nacimiento`, nombre, nacionalidad) VALUES (id_autor, fecha_nacimiento, nacionalidad_autor, nombre_autor);
8 END
9 //
10 DELIMITER
11
12
13 call agregar_autor("10", "13/12/2023", "cucuta", "yeison osorio");
14
15 -- PROCEDIMIENTO PARA ELIMINAR AUTOR
16 DELIMITER //
17 • CREATE PROCEDURE eliminar_autor (IN id_autor VARCHAR(10))
18 BEGIN
19   DELETE FROM autor
20   WHERE id = id_autor;
21 END//
22
23 call eliminar_autor("1010");
24
25
26
27
28 -- PROCEDIMIENTO PARA ACTUALIZAR UN AUTOR
29 DELIMITER //
30 • CREATE PROCEDURE actualizar_autor (IN id_autor VARCHAR(10), IN fecha_nac VARCHAR(45),
31                                     IN nacionalidad VARCHAR(20), IN nombre VARCHAR(45))
32 BEGIN
33   UPDATE autor
34   SET `fecha de nacimiento` = fecha_nac, nacionalidad = nacionalidad, nombre = nombre
35   WHERE id = id_autor;
36 END//
37
38 call actualizar_autor('5', '12/12/2021', 'BRASIL', 'YEISON OSORIO');
39
40 -- PROCEDIMIENTO PARA SELECCIONAR UN AUTOR
41 DELIMITER //
42 • CREATE PROCEDURE seleccionar_autor (IN id_autor VARCHAR(10))
43 BEGIN
44   SELECT id, `fecha de nacimiento`, nacionalidad, nombre
45   FROM autor
46   WHERE id = id_autor;
47 END//
48
49 • call seleccionar_autor('5');
--
```

Elabore una nueva tabla llamada "control\_de\_cambios\_librería" la cual debe contener 3 columnas (usuario, acción, fecha) y guarde utilizando 2 Triggers el nombre del usuario que agrego o elimino un registro en la tabla seleccionada en el punto anterior.

Archivos anexos en el repositorio

```
1 • use libreriaescalibre;
2
3 -- creamos la tabla para guardar los cambios
4 • CREATE TABLE control_de_cambios_libreria (
5     usuario VARCHAR(45),
6     accion VARCHAR(20),
7     fecha DATETIME
8 );
9
10 DELIMITER //
11 • CREATE TRIGGER ins_control_libreria_insert
12 AFTER INSERT ON autor
13 FOR EACH ROW
14 BEGIN
15     INSERT INTO control_de_cambios_libreria (usuario, accion, fecha)
16     VALUES (user(), 'insert', NOW());
17 END;
18 //
19
20 call agregar_autor('2020', '01-05-2020', 'colombia', 'choco');
21 SELECT * FROM control_de_cambios_libreria;
--
24 -- eliminar por trigger
25 DELIMITER //
26 • CREATE TRIGGER after_delete_autor
27 AFTER DELETE ON autor
28 FOR EACH ROW
29 BEGIN
30     INSERT INTO control_de_cambios_libreria (usuario, accion, fecha)
31     VALUES (user(), 'delete', NOW());
32 END;
33 //
34
35 • drop trigger after_delete_autor;
36 SELECT * FROM control_de_cambios_libreria;
37 call eliminar_autor("5")
```

Elabore 4 procedimientos almacenados que me permitan agregar, actualizar, consultar y borrar, en una de las tablas del Hospital (segunda actividad).

Archivos anexos en el repositorio

```

1 • use hospital;
2
3 DELIMITER //
4 • CREATE PROCEDURE insertar_medico (IN id VARCHAR(15), nombre VARCHAR(45), IN apellido VARCHAR(45), especialidad_medico VARCHAR(45))
5 BEGIN
6     INSERT INTO medico (id_medico, nombre_medico, apellido_medico, especialidad) VALUES (id, nombre, apellido, especialidad_medico);
7 END//
8 DELIMITER
9
10 call hospital.insertar_medico('M008', 'YEISON FERNEY', 'OSORIO_BUITRAGO', 'ING DE SISTEMAS');
11 DROP TRIGGER control_hospital_insert
12
13 DELIMITER //
14 • CREATE PROCEDURE seleccionar_medico (IN id VARCHAR(15))
15 BEGIN
16     SELECT id_medico, nombre_medico, apellido_medico, especialidad
17     FROM medico
18     WHERE id_medico = id;
19 END//
20
21 call hospital.seleccionar_medico('M003');
22
23 DELIMITER //
24
25 • CREATE PROCEDURE eliminar_medico (IN id VARCHAR(15))
26 BEGIN
27     DELETE FROM medico
28     WHERE id_medico = id;
29 END//
30
31 call hospital.eliminar_medico('M006');
32
33 DELIMITER //
34
35 • CREATE PROCEDURE actualizar_medico (IN id VARCHAR(15), IN nombre_medico VARCHAR(45),
36                                     IN apellido_medico VARCHAR(45), IN especialidad VARCHAR(45))
37 BEGIN
38     UPDATE medico
39     SET nombre_medico = nombre_medico, apellido_medico = apellido_medico, especialidad = especialidad
40     WHERE id_medico = id;
41 END//
42
43
44 • call hospital.actualizar_medico('M008', 'JUAN CHUANDO', 'OSORIO', 'INGENIERO');

```

Elabore una nueva tabla llamada "control\_de\_cambios\_hospital" la cual debe contener 3 columnas (usuario, acción, fecha) y guarde utilizando 2 Triggers el nombre del usuario que agrego o elimino un registro en la tabla seleccionada en el punto anterior.

Archivos anexos en el repositorio

```

1 • use hospital;
2
3 • create table control_de_cambios_hospital(
4     nombre varchar(50),
5     accion varchar(30),
6     fecha datetime default current_timestamp
7 );
8
9 -- trigger insert
10 DELIMITER //
11 • CREATE TRIGGER control_hospital_insert
12 AFTER INSERT ON medico
13 FOR EACH ROW
14 BEGIN
15     INSERT INTO control_de_cambios_hospital (usuario, accion, fecha)
16     VALUES (user(), 'insert', NOW());
17 END;
18 //
19
20 call insertar_medico('M007', 'YEISON', 'OSORIO', 'CORAZONOLOGO');
21 call insertar_medico('M001', 'YEISON', 'BUITRAGO', 'TURISTAS');
22 -- eliminar trigger
23
24
25 DELIMITER //
26 • CREATE TRIGGER control_hospital_delete
27 AFTER DELETE ON medico
28 FOR EACH ROW
29 BEGIN
30     INSERT INTO control_de_cambios_hospital (usuario, accion, fecha)
31     VALUES (user(), 'delete', NOW());
32 END;
33 //
34
35 • call eliminar_medico('M006');

```

## Documentos Anexos

### Carpeta Actividad 1

- ✓ consultasSQL\_Libreria.sql
- ✓ Insert\_Libreria.sql
- ✓ LibreriaBD.sql
- ✓ libreriabuscarlibre.mwb
- ✓ ProcedimientoLibreria
- ✓ TriggerLibreria

### Carpeta Actividad 2

- ✓ Hospital.mwb
- ✓ HospitalBD.sql
- ✓ Insert\_hospital.sql
- ✓ ProcedimientoHospital
- ✓ ConsultasVistas.sql
- ✓ HospitalTrigger