

## DOCUMENTACION

### Consultas y Vistas

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

- 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.

R/: Se agregan los registros a todas las tablas como se muestra en las fotos.

```
1 * use libreriabuscalibre;
2
3 * INSERT INTO autor (id, `fecha de nacimiento`, nacionalidad, nombre)
4 VALUES
5 ("A001", "01/01/1980", "Argentina", "Santiago Ramos"),
6 ("A002", "02/02/1985", "Chile", "Luz Cardona"),
7 ("A003", "03/03/1990", "Colombia", "Carlos Gómez"),
8 ("A004", "04/04/1995", "Perú", "Andrés Rodríguez"),
9 ("A005", "05/05/2000", "México", "Juan Ortiz");
10
11 * INSERT INTO editorial (nombre, ciudad, complemento, Telefono)
12 VALUES
13 ("Editorial 1", "Ciudad 1", "Fundada en 1950", "3189752"),
14 ("Editorial 2", "Ciudad 2", "Fundada en 1955", "3189753"),
15 ("Editorial 3", "Ciudad 3", "Fundada en 1966", "3189754"),
16 ("Editorial 4", "Ciudad 4", "Fundada en 1977", "3189755");
17
18 * INSERT INTO libro (ISBN, titulo, numero_paginas, nombre_editorial)
19 VALUES
20 ("L001", "El Quijote", "500", "Editorial 1"),
21 ("L002", "Don Juan Tenorio", "450", "Editorial 2"),
22 ("L003", "La Celestina", "400", "Editorial 3"),
23 ("L004", "La Isla del Tesoro", "500", "Editorial 4"),
24 ("L005", "El Gran Gatsby", "450", "Editorial 1"),
25 ("L006", "Matar un Ruiseñor", "400", "Editorial 2"),
26 ("L007", "1984", "450", "Editorial 3"),
27 ("L008", "La División del Agua", "450", "Editorial 4"),
28 ("L009", "La Historia Interminable", "400", "Editorial 1"),
29 ("L0010", "El Principito", "350", "Editorial 2"),
30 ("L0011", "Cien Años de Soledad", "450", "Editorial 3"),
31 ("L0012", "El Jardín de los Finzi-Contini", "400", "Editorial 4"),
32 ("L0013", "La Guerra y la Paz", "450", "Editorial 1"),
33 ("L0014", "La Montaña Mágica", "450", "Editorial 2"),
34 ("L0015", "El Amor en los Tiempos del Cólera", "450", "Editorial 3"),
35 ("L0016", "La Metamorfosis", "450", "Editorial 4"),
36 ("L0017", "El Proceso", "450", "Editorial 1"),
37 ("L0018", "La Strada", "450", "Editorial 2"),
38 ("L0019", "La Naranja Mecánica", "450", "Editorial 3"),
```

```
38     ("L0019", "La Naranja Mecánica", "450", "Editorial 3"),
39     ("L0020", "La Noche de los Tiempos", "450", "Editorial 4");
40
41 • INSERT INTO cliente (cedula, nombre)
42 VALUES
43     ("C001", "Luis García"),
44     ("C002", "Ana Martínez"),
45     ("C003", "Juan Pérez"),
46     ("C004", "James Muñoz"),
47     ("C005", "Arley Borja"),
48     ("C006", "Julieta Muñoz"),
49     ("C007", "Ana Velazquez");
50
51 • INSERT INTO libro_autor (id_autor, ISBN_libro)
52 VALUES
53     ("A001", "L001"),
54     ("A002", "L002"),
55     ("A003", "L003"),
56     ("A004", "L004"),
57     ("A005", "L005"),
58     ("A001", "L006"),
59     ("A002", "L007"),
60     ("A003", "L008"),
61     ("A004", "L009"),
62     ("A005", "L0010");
63
64 • INSERT INTO libro_cliente (id_cliente, ISBN_libro_cliente)
65 VALUES
66     ("C001", "L001"),
67     ("C002", "L002"),
68     ("C003", "L003"),
69     ("C004", "L004"),
70     ("C005", "L005"),
71     ("C001", "L006"),
72     ("C002", "L007"),
73     ("C003", "L008"),
74     ("C004", "L009"),
75     ("C005", "L0010");
```

```

76
77 * INSERT INTO telefono_cliente (cedula_cliente, numero)
78 VALUES
79 ("C001", "555-555-551"),
80 ("C002", "555-555-552"),
81 ("C003", "555-555-553"),
82 ("C001", "555-555-554"),
83 ("C005", "555-555-555"),
84 ("C006", "555-555-556"),
85 ("C007", "555-555-557"),
86 ("C002", "555-555-558"),
87 ("C001", "555-555-559"),
88 ("C004", "555-555-560"),
89 ("C002", "555-555-560");
90

```

- 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.

R/: se mostrarán las consultas que se crearon para cumplir con los requerimientos.

**Consulta 1:** Para conocer el nombre y la fecha de nacimiento de cada escritor

```
-- Para conocer el nombre y la fecha de nacimiento de cada escritor:
```

```
SELECT nombre, `fecha de nacimiento` FROM autor;
```

Result Grid			Filter Rows:
	nombre	fecha de nacimiento	
▶	Santiago Ramos	01/01/1980	
	Luz Cardona	02/02/1985	
	Carlos Gómez	03/03/1990	
	Andrés Rodríguez	04/04/1995	
	Juan Ortiz	05/05/2000	

**Consulta 2:** Para conocer la cantidad de libros diferentes vendidos

```
-- Para conocer la cantidad de libros diferentes vendidos:
```

```
SELECT ISBN_libro_cliente, count(*) as num_vendidos
FROM libro_cliente
GROUP BY ISBN_libro_cliente;
```

	ISBN_libro_cliente	num_vendidos
►	L001	1
	L0010	1
	L002	1
	L003	1
	L004	1
	L005	1
	L006	1
	L007	1
	L008	1
	L009	1

**Consulta 3:** Para conocer el nombre de los clientes acompañado de su número de teléfono

```
-- Para conocer el nombre de los clientes acompañado de su número de teléfono:
```

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

Result Grid   Filter Rows:		
	nombre	numero
►	Luis García	555-555-551
	Luis García	555-555-554
	Luis García	555-555-559
	Ana Martínez	555-555-552
	Ana Martínez	555-555-558
	Ana Martínez	555-555-560
	Juan Pérez	555-555-553
	James Muñoz	555-555-550
	Arley Borja	555-555-555
	Julieta Muñoz	555-555-556
	Ana Velazquez	555-555-557

**Consulta 4:** Para conocer el nombre del libro acompañado por su autor o sus autores

```
-- Para conocer el nombre del libro acompañado por su autor o sus autores:
```

```
SELECT libro.titulo, autor.nombre  
FROM libro  
JOIN libro_autor ON libro.ISBN = libro_autor.ISBN_libro  
JOIN autor ON libro_autor.id_autor = autor.id;
```

	titulo	nombre
►	El Quijote	Santiago Ramos
	Matar un Ruiseñor	Santiago Ramos
	Don Juan Tenorio	Luz Cardona
	1984	Luz Cardona
	La Celestina	Carlos Gómez
	La División del Agua	Carlos Gómez
	La Isla del Tesoro	Andrés Rodríguez
	La Historia Interminable	Andrés Rodríguez
	El Principito	Juan Ortiz
	El Gran Gatsby	Juan Ortiz

**Consulta 5:** Para conocer el nombre de las editoriales que han logrado vender libros

```
-- Para conocer el nombre de las editoriales que han logrado vender libros:
```

```
SELECT nombre_editorial, count(*) as libros_vendidos  
FROM libro  
JOIN libro_cliente ON libro.ISBN = libro_cliente.ISBN_libro_cliente  
GROUP BY nombre_editorial;
```

Result Grid	Filter Rows:
nombre_editorial	libros_vendidos
► Editorial 1	3
Editorial 2	3
Editorial 3	2
Editorial 4	2



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

#### R/: VISTA 1:

Se hace la siguiente vista mostrando los datos más importantes a la hora de comprar un libro, en este caso son nombre, páginas y editorial, se descartando el ISBN ya que no nos fijamos mucho en esto.

```
create view Libros as
select titulo, numero_paginas, nombre_editorial
from libro;
```

Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
	titulo	numero_paginas	nombre_editorial
▶	El Quijote	500	Editorial 1
	El Principito	350	Editorial 2
	Cien Años de Soledad	450	Editorial 3
	El Jardín de los Finzi-Contini	400	Editorial 4
	La Guerra y la Paz	450	Editorial 1
	La Montaña Mágica	450	Editorial 2
	El Amor en los Tiempos del Cólera	450	Editorial 3
	La Metamorfosis	450	Editorial 4
	El Proceso	450	Editorial 1
	La Strada	450	Editorial 2
	La Naranja Mecánica	450	Editorial 3
	Don Juan Tenorio	450	Editorial 2
	La Noche de los Tiempos	450	Editorial 4
	La Celestina	400	Editorial 3
	La Isla del Tesoro	500	Editorial 4
	El Gran Gatsby	450	Editorial 1
	Matar un Ruiseñor	400	Editorial 2
	1984	450	Editorial 3
	La División del Agua	450	Editorial 4
	La Historia Interminable	400	Editorial 1

#### VISTA 2:

Se saca la información de los autores en este caso el nombre, fecha de nacimiento y nacionalidad, considerando que la información del autor es importante a la hora de comprar libros.

```
CREATE VIEW informacion_autores AS
SELECT nombre, `fecha de nacimiento`, nacionalidad
FROM autor;
```

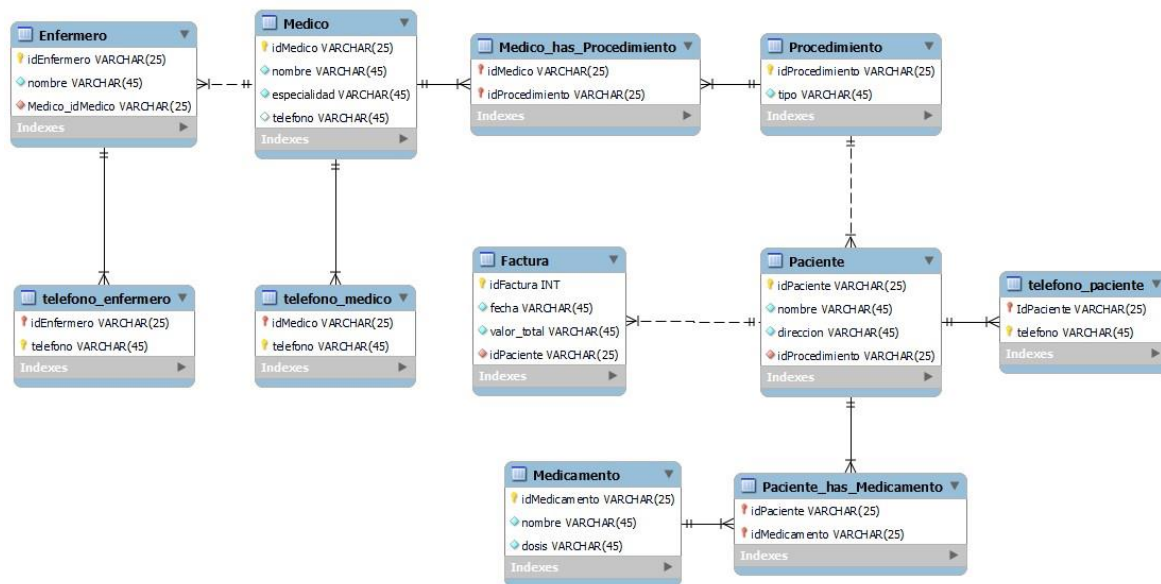
	nombre	fecha de nacimiento	nacionalidad
▶	Santiago Ramos	01/01/1980	Argentina
	Luz Cardona	02/02/1985	Chile
	Carlos Gómez	03/03/1990	Colombia
	Andrés Rodríguez	04/04/1995	Perú
	Juan Ortiz	05/05/2000	México

**Segunda actividad:** <https://github.com/Ococho/Hospital-GNECJ.git>

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

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

R/: Se adjunta foto del MR realizado en workbench, también se adjunta al repositorio



- Complete la información para las tablas realizadas con al menos 5 registros por tabla.

R/: Se agregan los registros a las tablas del hospital.

```
8 * INSERT INTO Medico (idMedico, nombre, especialidad, telefono)
9 VALUES
10 ("M001", "James Muñoz", "Pediatria", "555-555-5555"),
11 ("M002", "Julieta Muñoz", "Cardiología", "555-555-5556"),
12 ("M003", "Sara Mejia", "Neurología", "555-555-5557"),
13 ("M004", "Jams Muñoz", "Oftalmología", "555-555-5558"),
14 ("M005", "Carlos Muñoz", "Ortopedia", "555-555-5559");
15
16 * INSERT INTO Enfermero (idEnfermero, nombre, Medico_idMedico)
17 VALUES
18 ("E001", "Pedro Ramos", "M001"),
19 ("E002", "Javier Muñoz", "M002"),
20 ("E003", "Ara Borja", "M003"),
21 ("E004", "Luz Cardona", "M004"),
22 ("E005", "David Ortiz", "M005");
23
24 * INSERT INTO Procedimiento (idProcedimiento, tipo)
25 VALUES
26 ("P001", "Cirugía"),
27 ("P002", "Terapia"),
28 ("P003", "Radiografía"),
29 ("P004", "Tomografía"),
30 ("P005", "Ecografía");
31
32 * INSERT INTO Paciente (idPaciente, nombre, direccion, idProcedimiento)
33 VALUES
34 ("PA001", "Juan González", "Calle 123", "P001"),
35 ("PA002", "Ana López", "Calle 124", "P002"),
36 ("PA003", "Luis Pérez", "Calle 125", "P003"),
37 ("PA004", "María Rodríguez", "Calle 126", "P004"),
38 ("PA005", "Carlos Martínez", "Calle 127", "P005");
39
```



```
40 • INSERT INTO Factura (idFactura, fecha, valor_total, idPaciente)
41 VALUES
42 ("001", "2023/02/13", "100", "PA001"),
43 ("002", "2023/02/13", "100", "PA002"),
44 ("003", "2023/02/13", "100", "PA003"),
45 ("004", "2023/02/13", "100", "PA004"),
46 ("005", "2023/02/13", "100", "PA005");
47
48 • INSERT INTO Medico_has_Procedimiento (idMedico, idProcedimiento)
49 VALUES
50 ("M001", "P001"),
51 ("M002", "P002"),
52 ("M003", "P003"),
53 ("M004", "P004"),
54 ("M005", "P005");
55
56 • INSERT INTO Medicamento (idMedicamento, nombre, dosis)
57 VALUES
58 ("MD001", "Ibuprofeno", "200mg"),
59 ("MD002", "Paracetamol", "500mg"),
60 ("MD003", "Amlodipino", "10mg"),
61 ("MD004", "Naproxeno", "250mg"),
62 ("MD005", "Loratadina", "100mg");
63
64 • INSERT INTO Paciente_has_Medicamento (idPaciente, idMedicamento)
65 VALUES
66 ("PA001", "MD001"),
67 ("PA002", "MD002"),
68 ("PA003", "MD003"),
69 ("PA004", "MD004"),
70 ("PA005", "MD005");
71
```

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

R/: **Consulta 1:** vemos que medicamentos toma cada paciente y su dosis

```

32 * SELECT
33     Paciente.nombre AS 'Nombre del Paciente',
34     Medicamento.nombre AS 'Nombre del Medicamento',
35     Medicamento.dosis AS 'Dosis Suministrada'
36 FROM
37     Paciente
38 JOIN Paciente_has_Medicamento ON Paciente.idPaciente = Paciente_has_Medicamento.idPaciente
39 JOIN Medicamento ON Paciente_has_Medicamento.idMedicamento = Medicamento.idMedicamento;

```

Nombre del Paciente	Nombre del Medicamento	Dosis Suministrada
Juan González	Ibuprofeno	200mg
Ana López	Paracetamol	500mg
Luis Pérez	Amlodipino	10mg
María Rodríguez	Naproxeno	250mg
Carlos Martínez	Loratadina	100mg

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

R/: **Consulta 2:** Vemos los enfermeros que estuvieron en los procedimientos de los pacientes

```

41 * SELECT
42     Paciente.nombre AS 'Nombre del Paciente',
43     Procedimiento.tipo AS 'Tipo de Procedimiento',
44     Enfermero.nombre AS 'Nombre del Enfermero'
45 FROM
46     Paciente
47 JOIN Procedimiento ON Paciente.idProcedimiento = Procedimiento.idProcedimiento
48 JOIN Medico_has_Procedimiento ON Procedimiento.idProcedimiento = Medico_has_Procedimiento.idProcedimiento
49 JOIN Medico ON Medico_has_Procedimiento.idMedico = Medico.idMedico
50 JOIN Enfermero ON Enfermero.Medico_idMedico = Medico.idMedico

```

Nombre del Paciente	Tipo de Procedimiento	Nombre del Enfermero
Juan González	Cirugía	Pedro Ramos
Ana López	Terapia	Javier Muñoz
Luis Pérez	Radiografía	Ara Borja
María Rodríguez	Tomografía	Luz Cardona
Carlos Martínez	Ecografía	David Ortiz

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

R/: se agrega las vistas y el por qué se hicieron estas mismas.

#### **primera vista: facturas de los pacientes.**

Este vista es importante porque permite ver las facturas emitidas a cada paciente para el manejo financiero del hospital o para verificar que se cobren las cantidades adecuadas a cada paciente.

```
CREATE VIEW pacientes_facturas AS
SELECT Paciente.idPaciente, Paciente.nombre, Factura.fecha, Factura.valor_total
FROM Paciente
JOIN Factura ON Paciente.idPaciente = Factura.idPaciente;
```

	idPaciente	nombre	fecha	valor_total
▶	PA001	Juan González	2023/02/13	100
	PA002	Ana López	2023/02/13	100
	PA003	Luis Pérez	2023/02/13	100
	PA004	María Rodríguez	2023/02/13	100
	PA005	Carlos Martínez	2023/02/13	100

#### **Segunda vista: médico y su especialidad.**

Esta vista es importante ya que permite visualizar de manera clara y sencilla la información sobre los médicos y su especialidad, lo cual puede ser útil las funciones dentro del hospital.

```
-- vista para ver los medicos y su especialidad
CREATE VIEW vista_medicos_especialidad AS
SELECT idMedico, nombre, especialidad
FROM Medico;
```

	idMedico	nombre	especialidad
▶	M001	James Muñoz	Pediatría
	M002	Julieta Muñoz	Cardiología
	M003	Sara Mejia	Neurología
	M004	Jams Muñoz	Oftalmología
	M005	Carlos Muñoz	Ortopedia

#### **Tercera vista: medicamentos por paciente.**

Esta vista es importante porque permite ver qué medicamentos ha tomado cada paciente y su dosis correspondiente, lo cual permite monitorear los tratamientos.

```
-- vista para ver medicamentos asignados a los pacientes

CREATE VIEW MedicamentosPorPaciente AS
SELECT Paciente.idPaciente, Paciente.nombre AS nombrePaciente, Medicamento.nombre AS nombreMedicamento, Medicamento.dosis
FROM Paciente
JOIN Paciente_has_Medicamento ON Paciente.idPaciente = Paciente_has_Medicamento.idPaciente
JOIN Medicamento ON Paciente_has_Medicamento.idMedicamento = Medicamento.idMedicamento;
```

	idPaciente	nombrePaciente	nombreMedicamento	dosis
	PA001	Juan González	Ibuprofeno	200mg
	PA002	Ana López	Paracetamol	500mg
	PA003	Luis Pérez	Amlodipino	10mg
	PA004	María Rodríguez	Naproxeno	250mg
	PA005	Carlos Martínez	Loratadina	100mg

### 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).

### R/: Procedimiento para agregar datos a tabla editorial:

```
-- agregar un nueva editorial

DELIMITER //
• CREATE PROCEDURE agregar_editorial(IN nombre VARCHAR(50), IN ciudad VARCHAR(30), IN complemento VARCHAR(100), IN Telefono VARCHAR(20))
• BEGIN
    INSERT INTO editorial(nombre, ciudad, complemento, Telefono) VALUES(nombre, ciudad, complemento, Telefono);
    INSERT INTO control_de_cambios_librería(usuario, accion, fecha) VALUES(usuario, 'agregar_editorial', NOW());
END //
DELIMITER ;
• call libreriaabuscalibre.agregar_editorial('Editorial 5', 'ciudad 5', 'Fundada en 1978', '3189756');
```

Editorial 5 agregada en la tabla editorial:

	nombre	ciudad	complemento	Telefono
	Editorial 1	Ciudad 1	Fundada en 1950	3189752
	Editorial 2	Ciudad 8	Fundada en 1800	1111111
	Editorial 3	Ciudad 3	Fundada en 1966	3189754
	Editorial 4	Ciudad 4	Fundada en 1977	3189755
	Editorial 5	ciudad 5	Fundada en 1978	3189756
	NULL	NULL	NULL	NULL

### Procedimiento para consultar datos a tabla editorial:

```
177 -- Consultar información de una editorial por su nombre
178
179 DELIMITER //
180 • CREATE PROCEDURE consultar_editorial(IN nombre_editorial VARCHAR(50))
181 • BEGIN
182     SELECT * FROM editorial WHERE nombre = nombre_editorial;
183 • END //
184 DELIMITER ;
185 • call libreriaescalibre.consultar_editorial('Editorial 3');
186
```

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

	nombre	ciudad	complemento	Telefono
▶	Editorial 3	Ciudad 3	Fundada en 1966	3189754

### Procedimiento para actualizar datos a tabla editorial.

```
-- actualizar información de una editorial existente
DELIMITER //
• CREATE PROCEDURE actualizar_editorial(IN nombre_editorial VARCHAR(50), IN ciudad_editorial VARCHAR(30), IN complemento_editorial VARCHAR(100),
• IN Telefono_editorial VARCHAR(20))
• BEGIN
    UPDATE editorial SET ciudad = ciudad_editorial, complemento = complemento_editorial, Telefono = Telefono_editorial WHERE nombre = nombre_editorial;
    INSERT INTO control_de_cambios_libreria(usuario, accion, fecha) VALUES(usuario, 'actualizar_editorial', NOW());
• END //
DELIMITER ;
call libreriaescalibre.actualizar_editorial('Editorial 2', 'Ciudad 8 ', 'Fundada en 1800', '1111111');
```

### Editorial 2 modificada:

Result Grid | Filter Rows: | Edit:

	nombre	ciudad	complemento	Telefono
▶	Editorial 1	Ciudad 1	Fundada en 1950	3189752
	Editorial 2	Ciudad 8	Fundada en 1800	1111111
	Editorial 3	Ciudad 3	Fundada en 1966	3189754
	Editorial 4	Ciudad 4	Fundada en 1977	3189755
	Editorial 5	ciudad 5	Fundada en 1978	3189756

editorial 1 ×



Procedimiento para eliminar datos a tabla editorial:

```
187 -- Eliminar un libro por su nombre
188
189 DELIMITER //
190 • CREATE PROCEDURE borrar_editorial(IN nombre_editorial VARCHAR(50))
191 • BEGIN
192     DELETE FROM editorial WHERE nombre = nombre_editorial;
193 • END //
194 DELIMITER ;
195 • call libreriaescalibre.borrar_editorial('Editorial 5');
196
197 • select * from editorial;
```

Editorial 5 creada anteriormente eliminada:

Result Grid				
Filter Rows: <input type="text"/>				
Edit:				
	nombre	ciudad	complemento	Telefono
▶	Editorial 1	Ciudad 1	Fundada en 1950	3189752
	Editorial 2	Ciudad 8	Fundada en 1800	1111111
	Editorial 3	Ciudad 3	Fundada en 1966	3189754
	Editorial 4	Ciudad 4	Fundada en 1977	3189755
*	NULL	NULL	NULL	NULL
editorial 1 x				

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

R/:

Tabla creada de control\_de\_cambios\_librería con las 3 columnas pedidas.

```
195 -- tabla llamada "control_de_cambios_librería" con las columnas "usuario", "accion" y "fecha"
196
197 • CREATE TABLE control_de_cambios_librería (
198     usuario VARCHAR(50),
199     accion VARCHAR(50),
200     fecha DATETIME
201 );
```

-- Trigger para registrar la inserción de una editorial:

```
203 -- Trigger para registrar la inserción de una editorial:
204 |
205 DELIMITER //
206 • CREATE TRIGGER agregar_editorial_trigger
207 AFTER INSERT ON editorial
208 FOR EACH ROW
209 BEGIN
210     INSERT INTO control_de_cambios_librería (usuario, accion, fecha) VALUES (USER(), 'insertar', NOW());
211 END; //
212 DELIMITER ;
213 • call libreriaescalibre.agregar_editorial("Editorial 2000", "ciudad 20", "Fundada en 2000", "3189799");
214
```

Result Grid	Filter Rows:	Export:
usuario	accion	fecha
root@localhost	insertar	2023-02-14 18:11:49

-- Trigger para registrar la inserción de una editorial:

```
215 -- Trigger para registrar la eliminación de un libro:
216 |
217 DELIMITER //
218 • CREATE TRIGGER eliminar_libro_trigger
219 BEFORE DELETE ON editorial
220 FOR EACH ROW
221 BEGIN
222     INSERT INTO control_de_cambios_librería (usuario, accion, fecha) VALUES (USER(), 'eliminar', NOW());
223 END //
224 DELIMITER ;
225 • call libreriaescalibre.borrar_editorial('Editorial 2000');
226
```

Result Grid	Filter Rows:	Export:
usuario	accion	fecha
root@localhost	insertar	2023-02-14 18:11:49
root@localhost	eliminar	2023-02-14 18:15:51

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

R/:

#### Procedimiento para agregar los datos a la tabla médico:

```

130      -- Agregar un nuevo registro en la tabla Medico
131
132      DELIMITER //
133      • CREATE PROCEDURE agregar_Medico (
134          IN idMedico_medico VARCHAR(25),
135          IN nombre_medico VARCHAR(45),
136          IN especialidad_medico VARCHAR(45),
137          IN telefono_medico VARCHAR(45)
138      )
139      • BEGIN
140          INSERT INTO Medico (idMedico, nombre, especialidad, telefono)
141          VALUES (idMedico_medico, nombre_medico, especialidad_medico, telefono_medico);
142      END //
143      DELIMITER ;
144      • call hospital.agregar_Medico('M006', 'Jeison', 'Ginecologo', '2223311');
145

```

Registro del nuevo médico agregado:

Result Grid				
Filter Rows: <input type="text"/>				
Edit:				
	idMedico	nombre	especialidad	telefono
▶	M001	James Muñoz	Pediatría	555-555-5555
	M002	Julieta Muñoz	Cardiología	555-555-5556
	M003	Sara Mejia	Neurología	555-555-5557
	M004	Jams Muñoz	Oftalmología	555-555-5558
	M005	Carlos Muñoz	Ortopedia	555-555-5559
	M006	Jeison	Ginecologo	2223311
	NULL	NULL	NULL	NULL

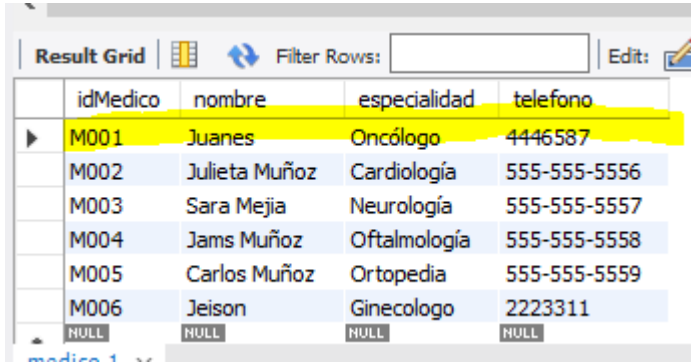
#### Procedimiento para actualizar los datos a la tabla médico:

```

148      DELIMITER //
149      • CREATE PROCEDURE actualizarMedico_Medico (IN idMedico_Medico VARCHAR(10), IN nombre_Medico VARCHAR(50), IN especialidad_Medico VARCHAR(50),
150          IN telefono_Medico VARCHAR(15)
151      )
152      • BEGIN
153          UPDATE Medico
154          SET nombre = nombre_Medico, especialidad = especialidad_Medico, telefono = telefono_Medico
155          WHERE idMedico = idMedico_Medico;
156      END //
157      DELIMITER ;
158      • call hospital.actualizarMedico_Medico('M001', 'Juanes', 'Oncólogo', '4446587');
159

```

Medico con id M001 actualizado:

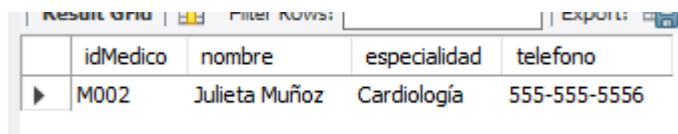


	idMedico	nombre	especialidad	telefono
▶	M001	Juanes	Oncólogo	4446587
	M002	Julieta Muñoz	Cardiología	555-555-5556
	M003	Sara Mejia	Neurología	555-555-5557
	M004	Jams Muñoz	Oftalmología	555-555-5558
	M005	Carlos Muñoz	Ortopedia	555-555-5559
	M006	Jeison	Ginecologo	2223311
	NULL	NULL	NULL	NULL

Procedimiento para consultar los datos a la tabla médico:

```
160      -- Consultar un registro en la tabla Medico
161
162      DELIMITER //
163  • CREATE PROCEDURE consultarMedico_Medico (
164      IN idMedico_Medico VARCHAR(10)
165  )
166  • BEGIN
167      SELECT idMedico, nombre, especialidad, telefono
168      FROM Medico
169      WHERE idMedico = idMedico_Medico;
170  END //
171      DELIMITER ;
172  • call hospital.consultarMedico_Medico('M002');
```

Médico con id M002 actualizado:



	idMedico	nombre	especialidad	telefono
▶	M002	Julieta Muñoz	Cardiología	555-555-5556

Procedimiento para eliminar los datos a la tabla médico:

```
174 -- Eliminar un registro de la tabla Medico
175
176 DELIMITER //
177 CREATE PROCEDURE eliminarMedico_Medico (IN idMedico_Medico VARCHAR(10)
178 )
179 BEGIN
180     DELETE FROM Medico WHERE idMedico = idMedico_Medico;
181 END //
182 DELIMITER ;
183 call hospital.eliminarMedico_Medico('M006');
184
```

Médico con id M006 eliminado:

Result Grid				
		Filter Rows:		
				Edit:
	idMedico	nombre	especialidad	telefono
▶	M001	Juanes	Oncólogo	4446587
	M002	Julieta Muñoz	Cardiología	555-555-5556
	M003	Sara Mejia	Neurología	555-555-5557
	M004	Jams Muñoz	Oftalmología	555-555-5558
	M005	Carlos Muñoz	Ortopedia	555-555-5559
*	NULL	NULL	NULL	NULL

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

R/: Tabla creada control\_de\_cambios\_hospital con las 3 columnas pedidas.

```
194 CREATE TABLE control_de_cambios_hospital (
195     usuario VARCHAR(50),
196     accion VARCHAR(50),
197     fecha DATETIME
198 );
```



Trigger Para agregar registros en la tabla Medico

```
200 -- Trigger Para agregar registros en la tabla Medico
201
202 DELIMITER //
203 • CREATE TRIGGER tr_agregarMedico
204 AFTER INSERT ON Medico
205 FOR EACH ROW
206 BEGIN
207     INSERT INTO control_de_cambios_hospital (usuario, accion, fecha) VALUES (USER(), 'insertar', NOW());
208 END; //
209 DELIMITER ;
210 • call hospital.agregar_Medico('M0010', 'Ledys', 'General', '445511');
211
```

Result Grid | Filter Rows: | Export:

	usuario	accion	fecha
▶	root@localhost	insertar	2023-02-14 18:26:02

-- Trigger Para eliminar registros de la tabla Medico

```
212 -- Trigger Para eliminar registros de la tabla Medico
213
214 DELIMITER //
215 • CREATE TRIGGER tr_eliminarMedico AFTER DELETE ON Medico
216 FOR EACH ROW
217 BEGIN
218     INSERT INTO control_de_cambios_hospital (usuario, accion, fecha) VALUES (USER(), 'eliminar', NOW());
219 END; //
220 DELIMITER ;
221 • call hospital.eliminarMedico_Medico('M0010');
222
```

Result Grid | Filter Rows: | Export:

	usuario	accion	fecha
▶	root@localhost	insertar	2023-02-14 18:26:02
	root@localhost	eliminar	2023-02-14 18:27:18

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

En el modelo de la librería agregaría información del autor, podría agregar una entidad “autor” que incluya información como el nombre, fecha de nacimiento, y su biografía por la importancia de esta misma, lo cual ayudaría a tener una mejor vista de la información del libro igualmente.

También agregaría detalles del libro, como información el género para poder filtrar y escoger más fácilmente.

En el modelo del hospital agregaría una entidad “diagnóstico” para almacenar información sobre los diagnósticos realizados a los pacientes. La relación entre paciente y diagnóstico podría ser de muchos a muchos, ya que un paciente puede tener varios diagnósticos y un diagnóstico puede ser realizado a varios pacientes.