

Taller 5 Bases de datos QA

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 **unicamente** comandos SQL creados por usted.

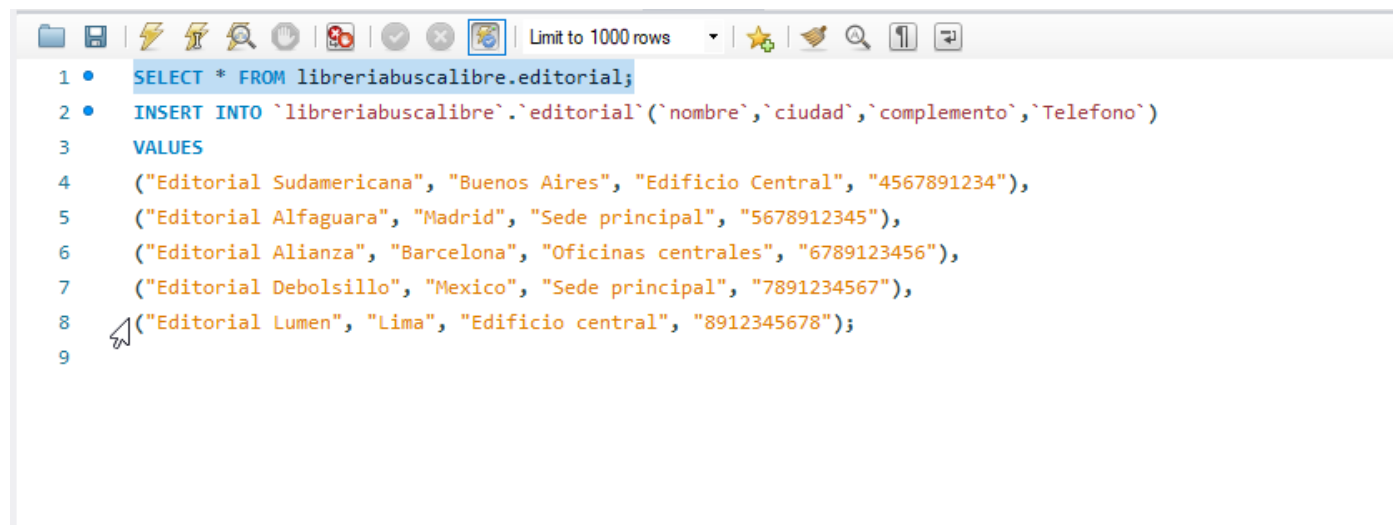
Inicialmente se hace uso del script proporcionado para crear la DB en Workbench utilizando Forward Engineering, una vez finalizada la creación se procede a completar la información de cada tabla utilizando comandos SQL dentro de la aplicación validando los campos a llenar y los registros requeridos.

Primero se completaran los campos de las tablas que contienen PK utilizadas por las otras tablas y posteriormente se completaran las demás tablas.

Para la tabla Editorial se requiere agregar 4 registros:

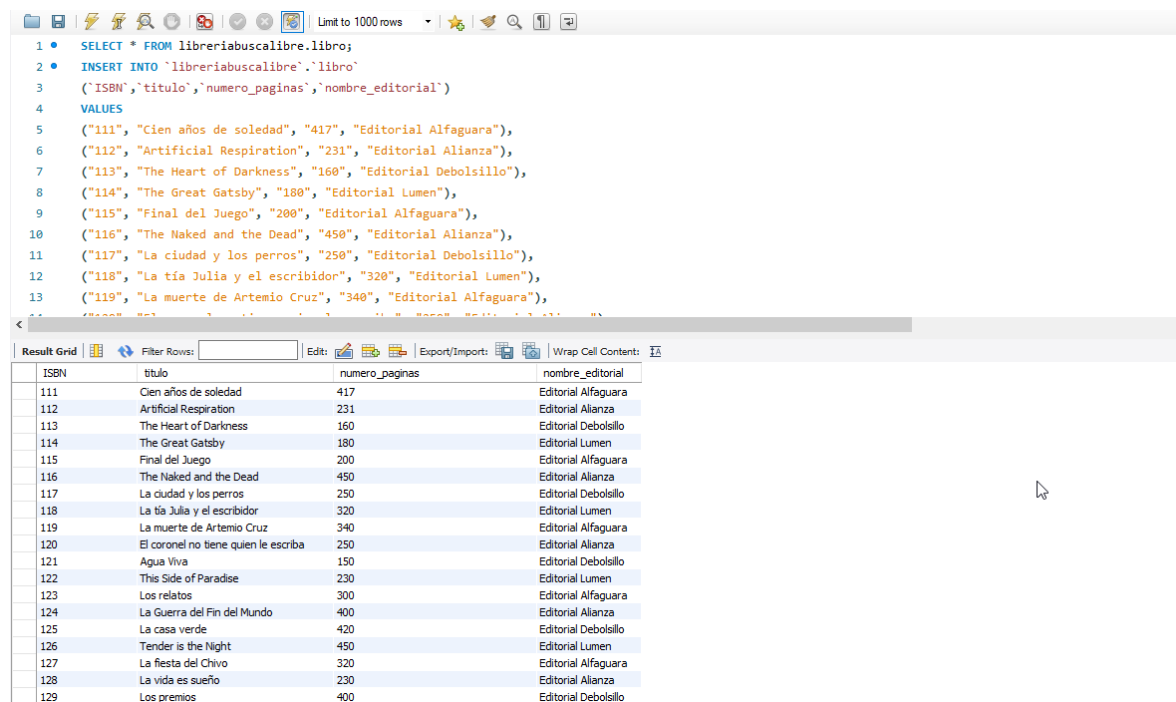
Inicialmente se abre el editor de SQL para ingresar el código necesario para los campos Nombre, Ciudad, Complemento y teléfono.

Para la columna complemento se usará información de complemento de sede y los datos de la ciudad donde estarán ubicadas las editoriales. Los datos de teléfono son números aleatorios. Todos los datos ingresados son de tipo VARCHAR.



```
1 • SELECT * FROM libreriaescalibre.editorial;
2 • INSERT INTO `libreriaescalibre`.`editorial`(`nombre`,`ciudad`,`complemento`,`Telefono`)
3   VALUES
4   ("Editorial Sudamericana", "Buenos Aires", "Edificio Central", "4567891234"),
5   ("Editorial Alfaguara", "Madrid", "Sede principal", "5678912345"),
6   ("Editorial Alianza", "Barcelona", "Oficinas centrales", "6789123456"),
7   ("Editorial Debolsillo", "Mexico", "Sede principal", "7891234567"),
8   ("Editorial Lumen", "Lima", "Edificio central", "8912345678");
9
```

Para la tabla Libro se hace uso de los valores usados previamente en la tabla Editorial en el campo de NOMBRE_EDITORIAL para agregar los libros a la DB y se proporciona la información de los libros teniendo en cuenta los campos requeridos de tipo VARCHAR. Para los ISBN se usaran datos de tipo VARCHAR pero no serán los ISBN reales de los libros.



```
1 • SELECT * FROM libreriaescalibre.libro;
2 • INSERT INTO `libreriaescalibre`.`libro`
3   (`ISBN`,`titulo`,`numero_paginas`,`nombre_editorial`)
4   VALUES
5   ("111", "Cien años de soledad", "417", "Editorial Alfaguara"),
6   ("112", "Artificial Respiration", "231", "Editorial Alianza"),
7   ("113", "The Heart of Darkness", "160", "Editorial Debolsillo"),
8   ("114", "The Great Gatsby", "180", "Editorial Lumen"),
9   ("115", "Final del Juego", "200", "Editorial Alfaguara"),
10  ("116", "The Naked and the Dead", "450", "Editorial Alianza"),
11  ("117", "La ciudad y los perros", "250", "Editorial Debolsillo"),
12  ("118", "La tía Julia y el escribidor", "320", "Editorial Lumen"),
13  ("119", "La muerte de Artemio Cruz", "340", "Editorial Alfaguara"),
14  ("120", "El coronel no tiene quien le escriba", "250", "Editorial Alianza"),
15  ("121", "Agua Viva", "150", "Editorial Debolsillo"),
16  ("122", "This Side of Paradise", "230", "Editorial Lumen"),
17  ("123", "Los relatos", "300", "Editorial Alfaguara"),
18  ("124", "La Guerra del Fin del Mundo", "400", "Editorial Alianza"),
19  ("125", "La casa verde", "420", "Editorial Debolsillo"),
20  ("126", "Tender is the Night", "450", "Editorial Lumen"),
21  ("127", "La fiesta del Chivo", "320", "Editorial Alfaguara"),
22  ("128", "La vida es sueño", "230", "Editorial Alianza"),
23  ("129", "Los premios", "400", "Editorial Debolsillo")
```

ISBN	titulo	numero_paginas	nombre_editorial
111	Cien años de soledad	417	Editorial Alfaguara
112	Artificial Respiration	231	Editorial Alianza
113	The Heart of Darkness	160	Editorial Debolsillo
114	The Great Gatsby	180	Editorial Lumen
115	Final del Juego	200	Editorial Alfaguara
116	The Naked and the Dead	450	Editorial Alianza
117	La ciudad y los perros	250	Editorial Debolsillo
118	La tía Julia y el escribidor	320	Editorial Lumen
119	La muerte de Artemio Cruz	340	Editorial Alfaguara
120	El coronel no tiene quien le escriba	250	Editorial Alianza
121	Agua Viva	150	Editorial Debolsillo
122	This Side of Paradise	230	Editorial Lumen
123	Los relatos	300	Editorial Alfaguara
124	La Guerra del Fin del Mundo	400	Editorial Alianza
125	La casa verde	420	Editorial Debolsillo
126	Tender is the Night	450	Editorial Lumen
127	La fiesta del Chivo	320	Editorial Alfaguara
128	La vida es sueño	230	Editorial Alianza
129	Los premios	400	Editorial Debolsillo

Para la tabla Autor se realiza el registro de 8 autores de distintas nacionalidades, se les asignara un ID numérico registrando el orden en el que cada autor fue ingresado, se investigo las fechas de nacimiento de cada autor y al ser este un valor de tipo DATE se ingresan los datos de fecha de nacimiento utilizando la notación AAAA-MM-DD y su respectiva nacionalidad.

```

1 • SELECT * FROM libreriaescalibre.autor;
2 • INSERT INTO `libreriaescalibre`.`autor`(`id`,`fecha de nacimiento`,`nacionalidad`,`nombre`)
3 VALUES
4 ("1", "1928-03-06", "Colombiana", "Gabriel García Márquez"),
5 ("2", "1947-03-31", "Española", "Ana María Matute"),
6 ("3", "1902-07-03", "Polaca", "Joseph Conrad"),
7 ("4", "1899-02-11", "Estadounidense", "F. Scott Fitzgerald"),
8 ("5", "1927-11-29", "Argentina", "Julio Cortázar"),
9 ("6", "1916-12-07", "Estadounidense", "Norman Mailer"),
10 ("7", "1948-08-25", "Peruana", "Mario Vargas Llosa"),
11 ("8", "1919-05-22", "Mexicana", "Carlos Fuentes");
12

```

id	fecha de nacimiento	nacionalidad	nombre
1	1928-03-06	Colombiana	Gabriel García Márquez
2	1947-03-31	Española	Ana María Matute
3	1902-07-03	Polaca	Joseph Conrad
4	1899-02-11	Estadounidense	F. Scott Fitzgerald
5	1927-11-29	Argentina	Julio Cortázar
6	1916-12-07	Estadounidense	Norman Mailer
7	1948-08-25	Peruana	Mario Vargas Llosa
8	1919-05-22	Mexicana	Carlos Fuentes
NULL	NULL	NULL	NULL

Para la tabla clientes se realiza el registro de 10 clientes con su respectiva Cedula y Nombre, teniendo en cuenta que la tabla no requiere mas información.

```

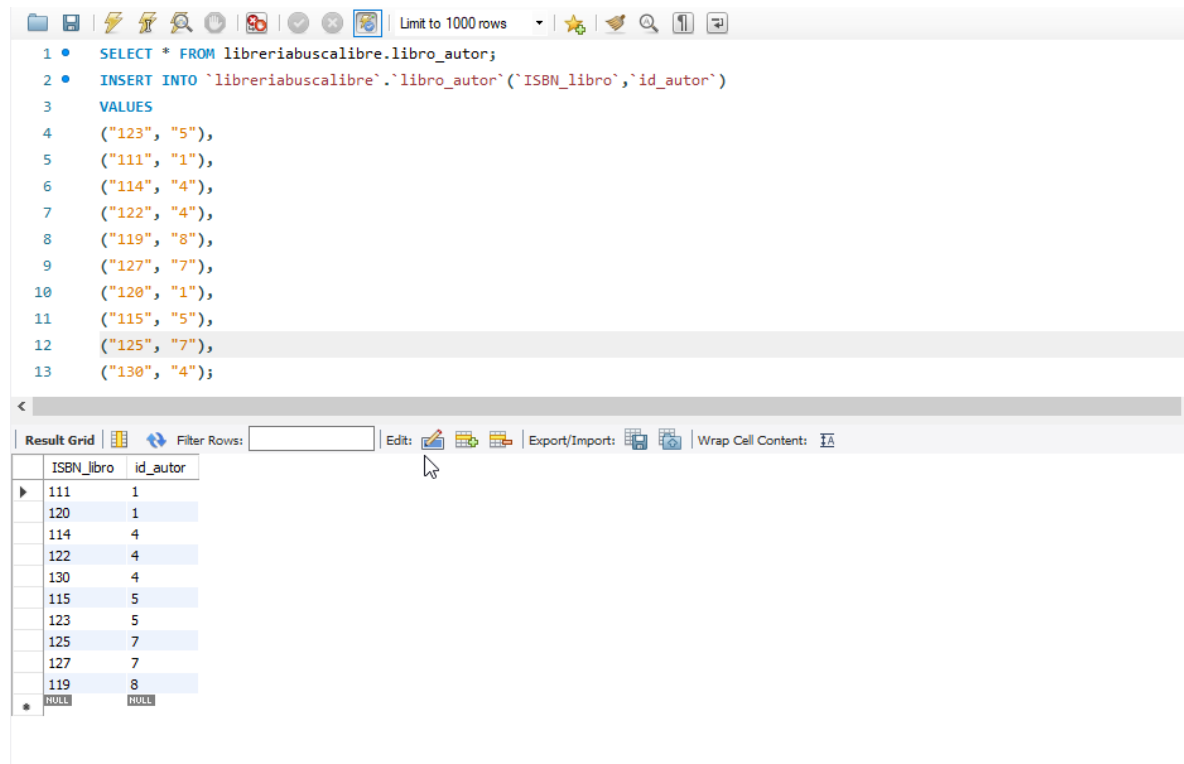
1 • SELECT * FROM libreriaescalibre.cliente;
2 • INSERT INTO `libreriaescalibre`.`cliente`(`cedula`,`nombre`)
3 VALUES
4 ("1234567890", "Andres Lopez"),
5 ("2345678901", "Lina Rodriguez"),
6 ("3456789012", "Edgar Andrade"),
7 ("4567890123", "Maria Gonzalez"),
8 ("5678901234", "Julian Casablanca"),
9 ("6789012345", "Jhon Perez"),
10 ("7890123456", "Lucia Ramirez"),
11 ("8901234567", "Carlos Perdomo"),
12 ("9012345678", "Julio Coronado"),
13 ("0123456789", "Nelson Garzon");

```

cedula	nombre
0123456789	Nelson Garzon
1234567890	Andres Lopez
2345678901	Lina Rodriguez
3456789012	Edgar Andrade

Para las tablas siguientes se empezará a completar los registros utilizando los datos de PK previamente ingresados y completando las tablas con los registros requeridos.

Para la tabla LIBRO_AUTOR se usaran los datos de ISBN de la tabla LIBRO y de ID de la tabla AUTOR, verificando que estos sean efectivamente los correspondientes autores de estos libros.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, search, and execution, along with a 'Limit to 1000 rows' dropdown. The SQL editor contains the following script:

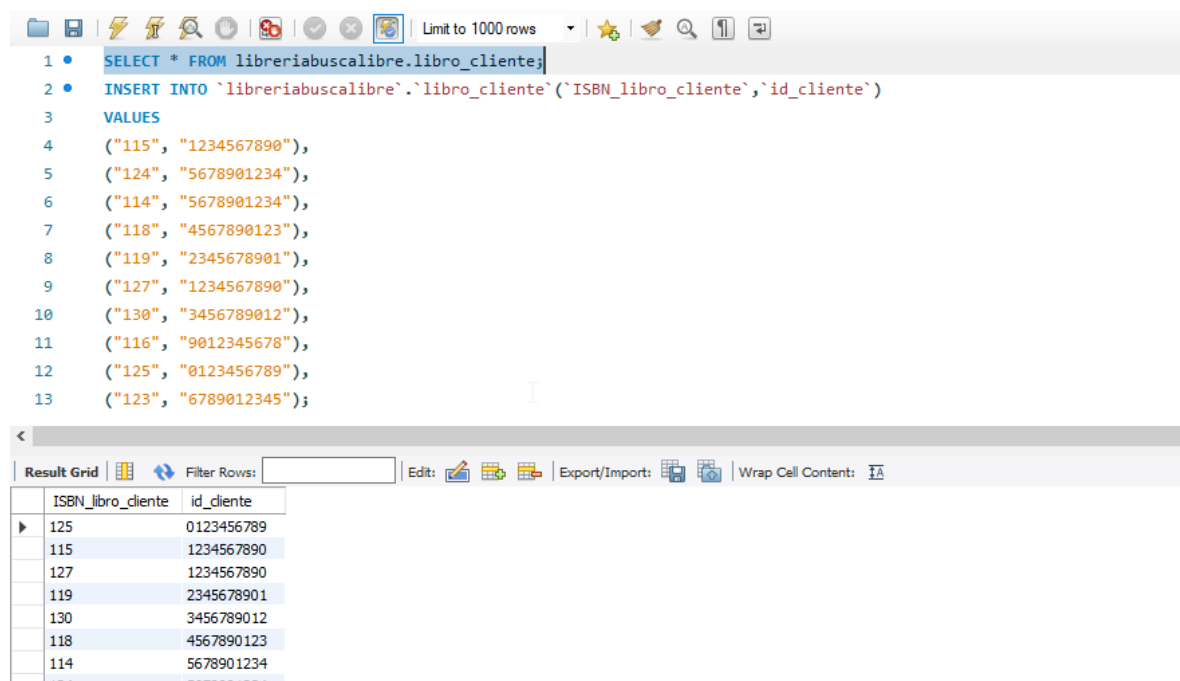
```

1 • SELECT * FROM libreriaescalibre.libro_autor;
2 • INSERT INTO `libreriaescalibre`.`libro_autor`(`ISBN_libro`,`id_autor`)
3   VALUES
4     ("123", "5"),
5     ("111", "1"),
6     ("114", "4"),
7     ("122", "4"),
8     ("119", "8"),
9     ("127", "7"),
10    ("120", "1"),
11    ("115", "5"),
12    ("125", "7"),
13    ("130", "4");
  
```

Below the editor is the 'Result Grid' tab, which displays the data inserted into the `libro_autor` table. The grid has two columns: `ISBN_libro` and `id_autor`.

ISBN_libro	id_autor
111	1
120	1
114	4
122	4
130	4
115	5
123	5
125	7
127	7
119	8
NULL	NULL

Para la tabla LIBRO_CLIENTE se utiliza el dato de ISBN del libro que el cliente compro y el ID del cliente que realizo la compra, estos datos se modificaran posteriormente al momento de realizar la consulta.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, search, and execution, along with a 'Limit to 1000 rows' dropdown. The SQL editor contains the following script:

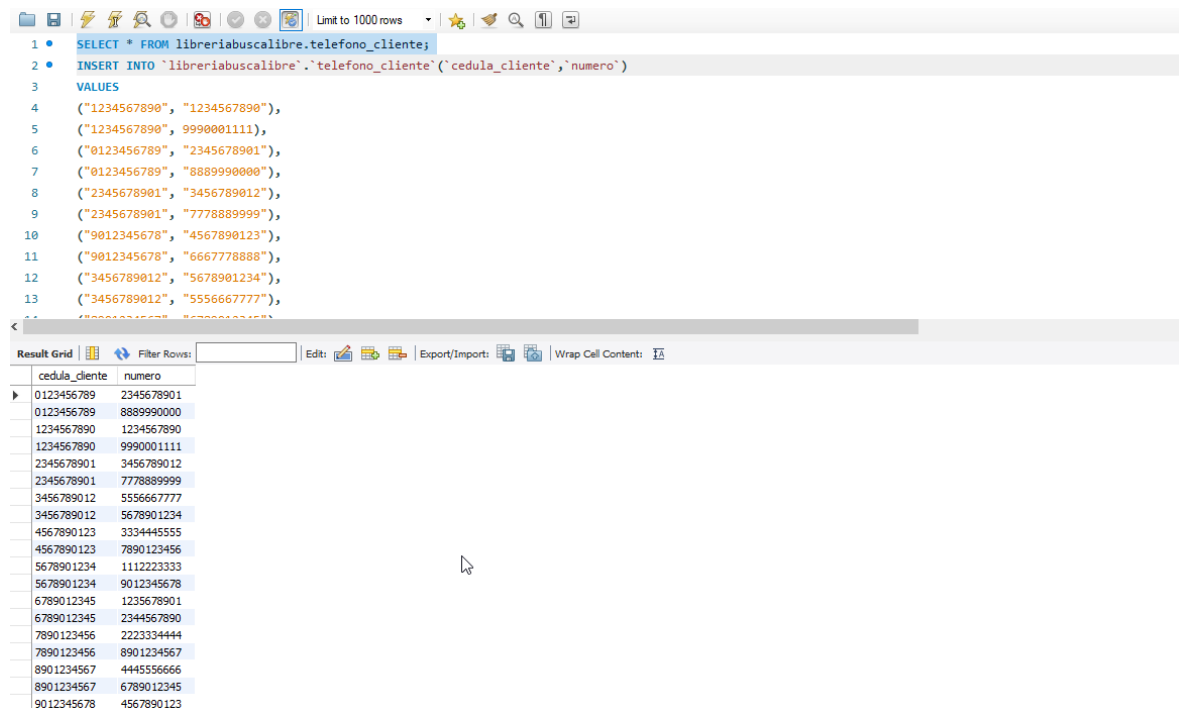
```

1 • SELECT * FROM libreriaescalibre.libro_cliente;
2 • INSERT INTO `libreriaescalibre`.`libro_cliente`(`ISBN_libro_cliente`,`id_cliente`)
3   VALUES
4     ("115", "1234567890"),
5     ("124", "5678901234"),
6     ("114", "5678901234"),
7     ("118", "4567890123"),
8     ("119", "2345678901"),
9     ("127", "1234567890"),
10    ("130", "3456789012"),
11    ("116", "9012345678"),
12    ("125", "0123456789"),
13    ("123", "6789012345");
  
```

Below the editor is the 'Result Grid' tab, which displays the data inserted into the `libro_cliente` table. The grid has two columns: `ISBN_libro_cliente` and `id_cliente`.

ISBN_libro_cliente	id_cliente
125	0123456789
115	1234567890
127	1234567890
119	2345678901
130	3456789012
118	4567890123
114	5678901234

Para la tabla TELEFONO_CLIENTE se ingresan 2 números telefónicos por cada id de cliente generados aleatoriamente de tipo VARCHAR.



The screenshot shows a database management interface. The top pane displays two SQL queries: a SELECT statement to retrieve data from 'libreriaabuscalibre.telefono_cliente' and an INSERT statement to populate it with random phone numbers for each client ID. The bottom pane shows the 'Result Grid' with two columns: 'cedula_cliente' and 'numero'. It contains 20 rows of data, each with a client ID and a corresponding phone number.

```
1 • SELECT * FROM libreriaabuscalibre.telefono_cliente;
2 • INSERT INTO `libreriaabuscalibre`.`telefono_cliente`(`cedula_cliente`,`numero`)
3   VALUES
4     ("1234567890", "1234567890"),
5     ("1234567890", "9990001111"),
6     ("0123456789", "2345678901"),
7     ("0123456789", "8889990000"),
8     ("2345678901", "3456789012"),
9     ("2345678901", "7778889999"),
10    ("9012345678", "4567890123"),
11    ("9012345678", "6667778888"),
12    ("3456789012", "5678901234"),
13    ("3456789012", "5556667777"),
14    ("5556667777", "8889990000"),
15    ("5556667777", "8889990000"),
16    ("5556667777", "8889990000"),
17    ("5556667777", "8889990000"),
18    ("5556667777", "8889990000"),
19    ("5556667777", "8889990000"),
20    ("5556667777", "8889990000");
```

cedula_cliente	numero
0123456789	2345678901
0123456789	8889990000
1234567890	1234567890
1234567890	9990001111
2345678901	3456789012
2345678901	7778889999
3456789012	5556667777
3456789012	5678901234
4567890123	3334445555
4567890123	7890123456
5678901234	1112223333
5678901234	9012345678
6789012345	1235678901
6789012345	2345678901
7890123456	2223334444
7890123456	8901234567
8901234567	4445556666
8901234567	6789012345
9012345678	4567890123

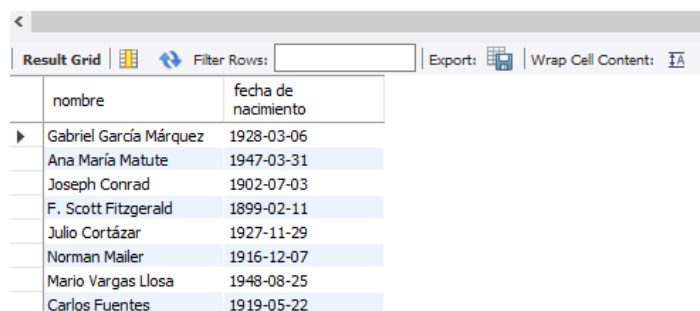
Con los registros en las tablas ya terminados podemos proceder a realizar las consultas requeridas en el siguiente punto.

- 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.

Consulta por nombre y fecha de nacimiento de los autores:

Para esta consulta haremos uso de la sentencia SELECT seleccionando únicamente los campos solicitados(Nombre de autor y fecha de nacimiento) de la tabla AUTOR.

```
1 • SELECT `nombre`, `fecha de nacimiento`  
2 FROM `libreriaescalibre`.`autor`;  
3
```



The screenshot shows a database query result grid with two columns: 'nombre' and 'fecha de nacimiento'. The results are listed in a table with alternating light blue and white rows.

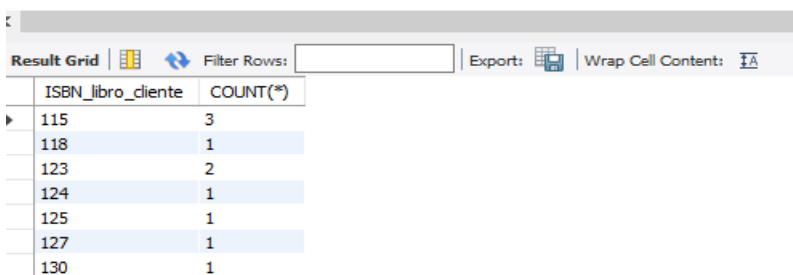
nombre	fecha de nacimiento
Gabriel García Márquez	1928-03-06
Ana María Matute	1947-03-31
Joseph Conrad	1902-07-03
F. Scott Fitzgerald	1899-02-11
Julio Cortázar	1927-11-29
Norman Mailer	1916-12-07
Mario Vargas Llosa	1948-08-25
Carlos Fuentes	1919-05-22

Como resultado obtenemos un listado de todos los nombres presentes en la columna nombre y las fechas de nacimiento de la columna fecha de nacimiento de la tabla AUTOR.

Cantidad de libros vendidos:

Como se indicó anteriormente para esta consulta los datos ingresados previamente fueron modificados con el fin de que hubiera libros que hubieran sido comprados mas de una vez, para esta consulta emplearemos la sentencia SELECT y la función COUNT para contar cuantas veces fue comprado el libro.

```
2 • SELECT `ISBN_libro_cliente`, COUNT(*)  
3 FROM libreriaescalibre.libro_cliente  
4 GROUP BY `ISBN_libro_cliente`;
```



The screenshot shows a database query result grid with two columns: 'ISBN_libro_cliente' and 'COUNT(*)'. The results are listed in a table with alternating light blue and white rows.

ISBN_libro_cliente	COUNT(*)
115	3
118	1
123	2
124	1
125	1
127	1
130	1

Nombre de cliente acompañado de su número telefónico:

Para esta consulta se seleccionan inicialmente las tablas CLIENTE y TELEFONO_CLIENTE de la DB, posteriormente se hace uso de la sentencia JOIN ya que ambas comparten el mismo dato de numero de cedula de CLIENTE por lo cual haremos uso de la cláusula ON para que se tomen específicamente los valores compartidos de “cedula” en ambas tablas.

```
1 • SELECT * FROM libreriaescalibre.cliente;  
2 • SELECT libreriaescalibre.cliente.nombre, libreriaescalibre.telefono_cliente.numero  
3 FROM libreriaescalibre.cliente  
4 JOIN libreriaescalibre.telefono_cliente  
5 ON libreriaescalibre.cliente.cedula = libreriaescalibre.telefono_cliente.cedula_cliente;  
6
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	nombre	numero		
▶	Nelson Garzon	2345678901		
	Nelson Garzon	8889990000		
	Andres Lopez	1234567890		
	Andres Lopez	9990001111		
	Lina Rodriguez	3456789012		
	Lina Rodriguez	7778889999		
	Edgar Andrade	5556667777		
	Edgar Andrade	5678901234		
	Maria Gonzalez	3334445555		
	Maria Gonzalez	7890123456		
	Julian Casabla...	1112223333		
	Julian Casabla...	9012345678		
	Jhon Perez	1235678901		
	Jhon Perez	2344567890		
	Lucia Ramirez	2223334444		
	Lucia Ramirez	8901234567		
	Carlos Perdomo	4445556666		
	Carlos Perdomo	6789012345		
	Julio Coronado	4567890123		

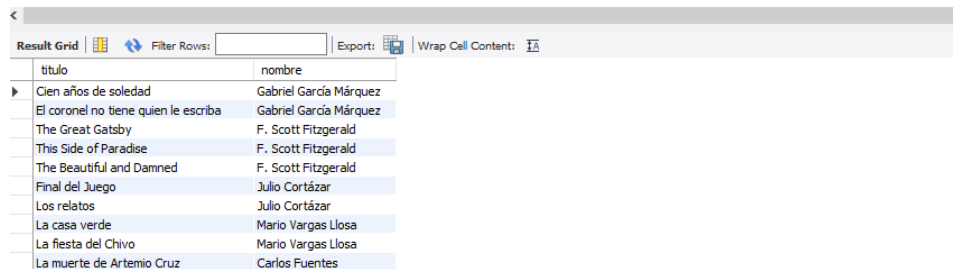
Nombre de libro acompañado de su autor o autores:

Para esta consulta utilizaremos la sentencia JOIN para unir tres tablas, LIBRO, LIBRO_AUTOR y AUTOR. Usaremos la consulta SELECT para seleccionar dos columnas en específico de las tablas título de la tabla LIBRO y nombre de la tabla AUTOR.

El primer JOIN se realizara entre la tabla LIBRO y LIBRO_AUTOR uniendo el campo ISBN de la tabla LIBRO con el campo ISBN_libro de la tabla LIBRO_AUTOR para asociar los libros escritos con el autor correspondiente.

El segundo JOIN se realiza entre la tabla LIBRO_AUTOR y AUTOR, se une en el campo id de la tabla AUTOR con el campo id_autor de la tabla LIBRO_AUTOR para obtener el nombre del autor correspondiente a cada libro. Combinando estas dos sentencias JOIN podemos obtener el nombre del libro y el nombre del autor correspondiente.

```
2 • SELECT libreriabuscalibre.libro.titulo, libreriabuscalibre.autor.nombre
3 FROM libreriabuscalibre.libro
4 JOIN libreriabuscalibre.libro_autor ON libreriabuscalibre.libro.ISBN = libreriabuscalibre.libro_autor.ISBN_libro
5 JOIN libreriabuscalibre.autor ON libreriabuscalibre.libro_autor.id_autor = libreriabuscalibre.autor.id;
```



The screenshot shows a database query result grid with two columns: 'titulo' and 'nombre'. The grid contains 12 rows of data. The first row is 'Cien años de soledad' by 'Gabriel García Márquez'. The second row is 'El coronel no tiene quien le escriba' by 'Gabriel García Márquez'. The third row is 'The Great Gatsby' by 'F. Scott Fitzgerald'. The fourth row is 'This Side of Paradise' by 'F. Scott Fitzgerald'. The fifth row is 'The Beautiful and Damned' by 'F. Scott Fitzgerald'. The sixth row is 'Final del Juego' by 'Julio Cortázar'. The seventh row is 'Los relatos' by 'Julio Cortázar'. The eighth row is 'La casa verde' by 'Mario Vargas Llosa'. The ninth row is 'La fiesta del Chivo' by 'Mario Vargas Llosa'. The tenth row is 'La muerte de Artemio Cruz' by 'Carlos Fuentes'.

titulo	nombre
Cien años de soledad	Gabriel García Márquez
El coronel no tiene quien le escriba	Gabriel García Márquez
The Great Gatsby	F. Scott Fitzgerald
This Side of Paradise	F. Scott Fitzgerald
The Beautiful and Damned	F. Scott Fitzgerald
Final del Juego	Julio Cortázar
Los relatos	Julio Cortázar
La casa verde	Mario Vargas Llosa
La fiesta del Chivo	Mario Vargas Llosa
La muerte de Artemio Cruz	Carlos Fuentes

Nombre editoriales que han vendido libros

Al igual que en la consulta anterior haremos uso de la sentencia JOIN para unir tres tablas diferentes: EDITORIAL, LIBRO y LIBRO_CLIENTES.

Inicialmente haremos uso de la sentencia JOIN para unir la información de las tablas EDITORIAL y LIBRO teniendo en cuenta que el nombre de la editorial es el mismo en ambas tablas, de esta forma obtenemos una tabla intermedia que contiene información sobre la editorial y el libro correspondiente. En el segundo JOIN aplica la misma lógica pero esta vez usando la tabla intermedia que fue creada con la sentencia JOIN previa y la tabla LIBRO_CLIENTE teniendo en cuenta que el ISBN es el campo compartido, así obtenemos una tabla final que contiene información sobre la editorial, el libro y la venta.

Por último, se utiliza la sentencia GROUP BY para agrupar las filas por nombre de editorial junto con la sentencia COUNT(*) para contar el número de filas y obtener una tabla con el nombre de cada editorial y el número de libros vendidos de cada una.

```
2 • SELECT libreriaescalibre.editorial.nombre, COUNT(*)
3 FROM libreriaescalibre.editorial
4 JOIN libreriaescalibre.libro ON libreriaescalibre.editorial.nombre = libreriaescalibre.libro.nombre_editorial
5 JOIN libreriaescalibre.libro_cliente ON libreriaescalibre.libro.ISBN = libro_cliente.ISBN_libro_cliente
6 GROUP BY libreriaescalibre.editorial.nombre;
```

Result Grid	
nombre	COUNT(*)
Editorial Debolsillo	1
Editorial Alfaguara	6
Editorial Lumen	2
Editorial Alianza	1

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

1. Se creo una vista que muestra que libros ha comprado cada cliente siguiendo la lógica de que al ser una librería se puede usar el contacto del cliente para ofrecer descuentos y publicidad del género favorito de libros de esta manera se puede generar mas ingresos gracias a la publicidad.

```
2 • CREATE VIEW vista_libros_comprados_cliente AS
3 SELECT libreriaescalibre.cliente.nombre AS Nombre_Cliente, libreriaescalibre.libro.titulo AS Titulo_Libro
4 FROM libreriaescalibre.cliente
5 JOIN libreriaescalibre.libro_cliente ON libreriaescalibre.cliente.cedula = libro_cliente.id_cliente
6 JOIN libreriaescalibre.libro ON libreriaescalibre.libro_cliente.ISBN_libro_cliente = libro.ISBN;
7 • SELECT * FROM vista_libros_comprados_cliente;
```

Result Grid	
Nombre_Cliente	Titulo_Libro
Nelson Garzon	La casa verde
Andres Lopez	Final del Juego
Andres Lopez	La fiesta del Chivo
Lina Rodriguez	Final del Juego
Edgar Andrade	The Beautiful and Damned
Maria Gonzalez	La tía Julia y el escribidor
Julian Casablanca	Final del Juego
Julian Casablanca	La Guerra del Fin del Mundo
Jhon Perez	Los relatos
Julio Coronado	Los relatos

2. Se creo una vista para saber que libros son los mas vendidos en la librería con el fin de si es el caso solicitar mas a la editorial y validar posibles razones de la popularidad del libro ya que de esta manera la librería se mantendría actualizada en los trends del momento y podría aprovechar para vender mas libros haciendo publicidad de que esta en existencia el libro que todos quieren leer.

```
2 • CREATE VIEW vista_libros_populares AS
3 SELECT libreriaescalibre.libro.titulo AS Titulo_Libro, COUNT(*) AS Numero_Ventas
4 FROM libreriaescalibre.libro
5 JOIN libreriaescalibre.libro_cliente ON libreriaescalibre.libro.ISBN = libreriaescalibre.libro_cliente.ISBN_libro_cliente
6 GROUP BY libro.titulo
7 ORDER BY Numero_Ventas DESC;
8 • SELECT * FROM vista_libros_populares;
```

Result Grid	
Filter Rows: <input type="text"/>	
Export: <input type="button" value=""/>	
Wrap Cell Content: <input type="button" value=""/>	
Titulo_Libro	Numero_Ventas
Final del Juego	3
Los relatos	2
La casa verde	1
La fiesta del Chivo	1
The Beautiful and Damned	1
La tía Julia y el escribidor	1
La Guerra del Fin del Mundo	1

2 Actividad

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

Inicialmente se valida el modelo E-R presente en el repositorio y se realiza la transformación del modelo a tablas teniendo en cuenta las entidades creadas y sus atributos. Adicionalmente se identifican las que serán las llaves primarias de cada una de las tablas.

Enfermero	Medico	Procedimiento
id_enfermero	id_medico	id_procedimiento
nombre	nombre	tipo
telefono	especialidad	
	telefono	
Paciente	Medicamento	Factura
id_paciente	id_medicamento	id_factura
nombre	nombre	fecha
direccion	dosis	valor total
telefono		

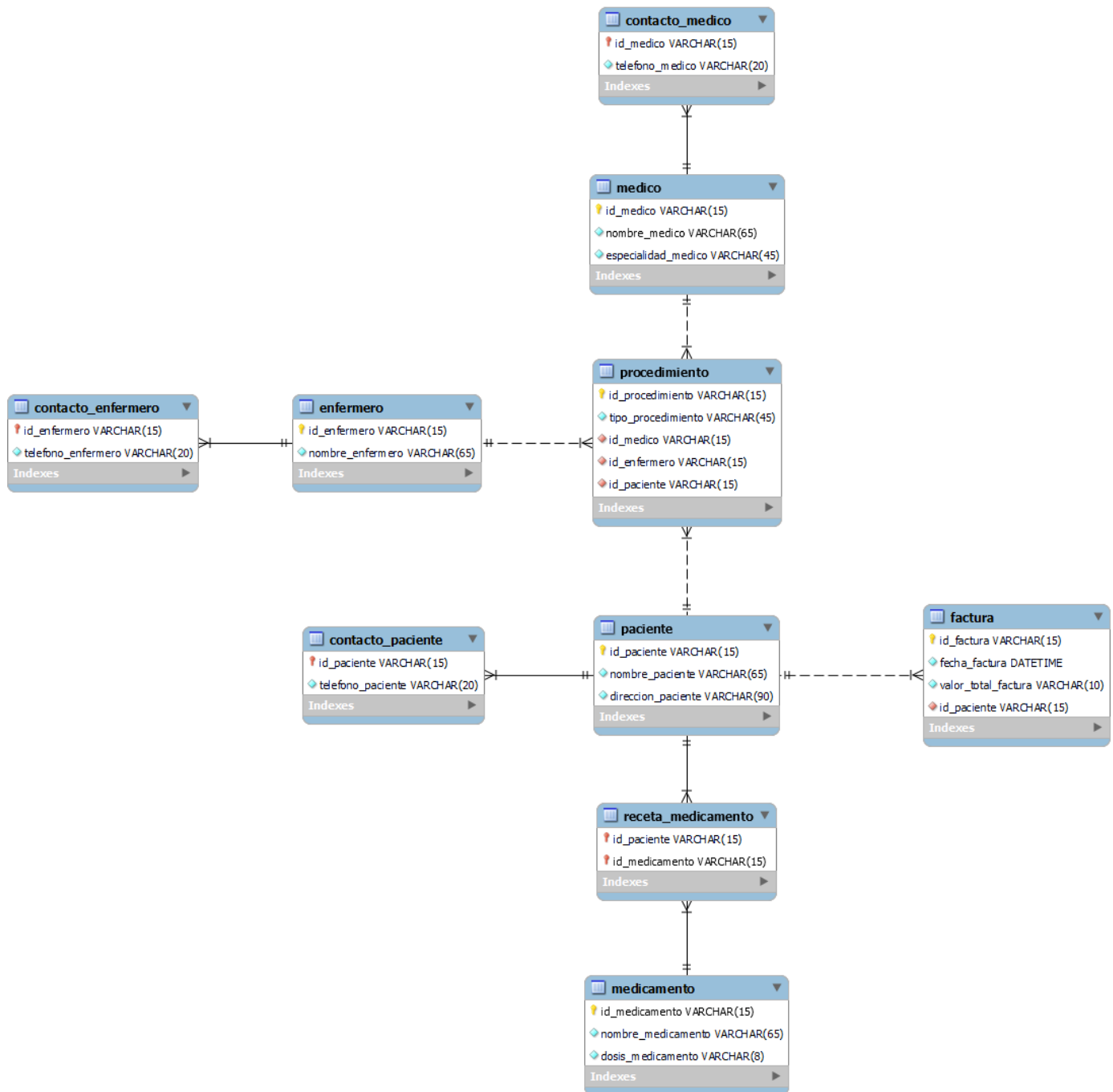
Una vez se tienen las tablas se procede a generar las tablas para los atributos multivalor continuando con la normalización del modelo.

Contacto enfermero	Contacto medico		
id_enfermero	id_medico		
telefono enfermero	telefono medico		
Enfermero	Medico		Procedimiento
id_enfermero	id_medico		id_procedimiento
nombre	nombre		tipo
	especialidad		
Paciente	Medicamento		Factura
id_paciente	id_medicamento		id_factura
nombre	nombre		fecha
direccion	dosis		valor total
Contacto paciente			
id_paciente			
telefono paciente			

Teniendo estas tablas procedemos a iniciar con la creación del modelo en Workbench, como cardinalidades tenemos las siguientes:

- 1-N: Enfermero - contacto_enfermero, ya que cada enfermero/a puede tener 1 o muchos teléfonos pero un teléfono solo puede estar asociado a un enfermero/a.
Medico - contacto_medico, misma situación que con la descripción de Enfermero y contacto_enfermero.
Paciente - contacto_paciente, misma situación que con la descripción de Enfermero y contacto_enfermero.
Medico – Procedimiento, un medico puede realizar muchos procedimientos pero un procedimiento solo puede ser realizado por un medico a la vez
Paciente – Procedimiento, a un paciente se le pueden realizar uno o muchos procedimientos pero solo un procedimiento puede ser realizado al paciente a la vez
Enfermero – Procedimiento, un enfermero puede asistir en uno o muchos procedimientos pero un procedimiento solo puede ser asistido por un enfermero a la vez
Paciente – Factura, un paciente puede tener una o muchas facturas pero una factura solo puede pertenecer a un paciente.
- N-M: Paciente – Medicamento, un paciente puede ser recetado uno o muchos medicamentos y un medicamento puede ser recetado a uno o muchos pacientes, por tanto se crea una tabla intermedia utilizando los atributos de id tanto de medicamento como de paciente que será llamada receta_medicamento.

Posteriormente se realiza el modelado y la creación de la DB usando Workbench y la herramienta de Forward Engineering como se muestra a continuación.



Una vez se crea la DB se procede a crear los registros en cada una de las tablas como se evidencia a continuación:

- **Tabla Enfermeros**

```
2 • INSERT INTO `procedimiento_medico`.`enfermero`(`id_enfermero`,`nombre_enfermero`)
3 VALUES ('E001', 'Juan Reyes'),
4 ('E002', 'Rubi Ochoa'),
5 ('E003', 'Nicandro Mejia'),
6 ('E004', 'Soraya Montenegro'),
7 ('E005', 'Paola Bracho');
8
```

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_enfermero	nombre_enfermero				
▶	E001	Juan Reyes				
	E002	Rubi Ochoa				
	E003	Nicandro Mejia				
	E004	Soraya Montenegro				
	E005	Paola Bracho				
✱	NULL	NULL				

- **Tabla Medico**

```
2 • INSERT INTO `procedimiento_medico`.`medico`(`id_medico`,`nombre_medico`,`especialidad_medico`)
3 VALUES
4 ('M001', 'Freddie Mercury', 'Oncología'),
5 ('M002', 'Paola Jara', 'Cardiología'),
6 ('M003', 'Elton John', 'Pediatria'),
7 ('M004', 'Paulina Rubio', 'Neurología'),
8 ('M005', 'Michael Jackson', 'Cirugía');
9
```

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_medico	nombre_medico	especialidad_medico			
	M001	Freddie Mercury	Oncología			
	M002	Paola Jara	Cardiología			
	M003	Elton John	Pediatria			
	M004	Paulina Rubio	Neurología			
	M005	Michael Jackson	Cirugía			
	NULL	NULL	NULL			

- Tabla Paciente

```
2 • INSERT INTO `procedimiento_medico`.`paciente`(`id_paciente`,`nombre_paciente`,`direccion_paciente`)
3 VALUES
4 ('P001', 'María del Rosario', 'Calle 123 # 45-67'),
5 ('P002', 'Orésteles Mercouri', 'Avenida 54 # 12-34'),
6 ('P003', 'Beatriz Pinzon', 'Carrera 78 # 90-12'),
7 ('P004', 'Catalina Santana', 'Calle 21 # 43-56'),
8 ('P005', 'Fernando Colunga', 'Avenida 87 # 23-45');
9
```

id_paciente	nombre_paciente	direccion_paciente
P001	María del Rosario	Calle 123 # 45-67
P002	Orésteles Mercouri	Avenida 54 # 12-34
P003	Beatriz Pinzon	Carrera 78 # 90-12
P004	Catalina Santana	Calle 21 # 43-56
P005	Fernando Colunga	Avenida 87 # 23-45







- Tabla Procedimiento

```
2 • INSERT INTO `procedimiento_medico`.`procedimiento`
3 (`id_procedimiento`,`tipo_procedimiento`,`id_medico`,`id_enfermero`,`id_paciente`)
4 VALUES
5 ('PR001', 'Cirugía de corazón', 'M002', 'E001', 'P003'),
6 ('PR002', 'Quimioterapia', 'M001', 'E002', 'P001'),
7 ('PR003', 'Extracción de apéndice', 'M005', 'E003', 'P002'),
8 ('PR004', 'Terapia ocupacional', 'M003', 'E004', 'P005'),
9 ('PR005', 'Radiografía de pulmones', 'M004', 'E005', 'P004');
```

id_procedimiento	tipo_procedimiento	id_medico	id_enfermero	id_paciente
PR001	Cirugía de corazón	M002	E001	P003
PR002	Quimioterapia	M001	E002	P001
PR003	Extracción de apéndice	M005	E003	P002
PR004	Terapia ocupacional	M003	E004	P005
PR005	Radiografía de pulmones	M004	E005	P004







- Tabla Factura

```
2 • INSERT INTO `procedimiento_medico`.`factura`(`id_factura`,`fecha_factura`,`valor_total_factura`,`id_paciente`)
3 VALUES ('F001', '2022-01-15 09:23:12', '3000000', 'P001'),
4 ('F002', '2022-01-18 14:11:01', '650000', 'P002'),
5 ('F003', '2022-01-21 11:45:23', '8000000', 'P003'),
6 ('F004', '2022-01-25 08:32:56', '150000', 'P004'),
7 ('F005', '2022-01-29 16:40:18', '90000', 'P005');
```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:   				
Export/Import:  				
Wrap Cell Content: 				
	id_factura	fecha_factura	valor_total_factura	id_paciente
►	F001	2022-01-15 09:23:12	3000000	P001
	F002	2022-01-18 14:11:01	650000	P002
	F003	2022-01-21 11:45:23	8000000	P003
	F004	2022-01-25 08:32:56	150000	P004
	F005	2022-01-29 16:40:18	90000	P005

- Tabla Medicamento

```
2 • INSERT INTO `procedimiento_medico`.`medicamento`(`id_medicamento`,`nombre_medicamento`,`dosis_medicamento`)
3 VALUES
4 ('M001', 'Aspirina', '100mg'),
5 ('M002', 'Paracetamol', '500mg'),
6 ('M003', 'Ibuprofeno', '200mg'),
7 ('M004', 'Amoxicilina', '500mg'),
8 ('M005', 'Omeprazol', '20mg');
9
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit:   			
Export/Import:  			
Wrap Cell Content: 			
	id_medicamento	nombre_medicamento	dosis_medicamento
►	M001	Aspirina	100mg
	M002	Paracetamol	500mg
	M003	Ibuprofeno	200mg
	M004	Amoxicilina	500mg
	M005	Omeprazol	20mg

- **Tabla receta_medicamento**

```

2 • INSERT INTO `procedimiento_medico`.`receta_medicamento`(`id_paciente`,`id_medicamento`)
3 VALUES
4 ('P003', 'M001'),
5 ('P001', 'M002'),
6 ('P002', 'M003'),
7 ('P001', 'M004'),
8 ('P003', 'M005');

```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
id_paciente	id_medicamento				
P003	M001				
P001	M002				
P002	M003				
P001	M004				
P003	M005				

- **Tabla contacto-enfermero**

```

2 • INSERT INTO `procedimiento_medico`.`contacto_enfermero`(`id_enfermero`,`telefono_enfermero`)
3 VALUES
4 ('E001', '1111111'),
5 ('E002', '2222222'),
6 ('E003', '3333333'),
7 ('E004', '4444444'),
8 ('E005', '5555555');
9

```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
id_enfermero	telefono_enfermero				
E001	1111111				
E002	2222222				
E003	3333333				
E004	4444444				
E005	5555555				

- Tabla contacto_medico

```

2 • INSERT INTO `procedimiento_medico`.`contacto_medico`(`id_medico`,`telefono_medico`)
3 VALUES
4 ('M001', '6666666'),
5 ('M002', '7777777'),
6 ('M003', '8888888'),
7 ('M004', '9999999'),
8 ('M005', '0000000');
9

```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_medico	telefono_medico			
▶	M001	6666666			
	M002	7777777			
	M003	8888888			
	M004	9999999			
	M005	0000000			
*	NULL	NULL			

- Tabla contacto_paciente

```

1 • SELECT * FROM procedimiento_medico.contacto_paciente;
2 • INSERT INTO `procedimiento_medico`.`contacto_paciente`(`id_paciente`,`telefono_paciente`)
3 VALUES
4 ('P001', '6789012345'),
5 ('P002', '9012345678'),
6 ('P003', '4567890123'),
7 ('P004', '4445556666'),
8 ('P005', '2345678901');

```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_paciente	telefono_paciente			
▶	P001	6789012345			
	P002	9012345678			
	P003	4567890123			
	P004	4445556666			
	P005	2345678901			

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

Esta consulta recupera información de los pacientes y los medicamentos recetados, y la filtra para asegurar que solo se incluyan los datos que se relacionan entre sí. Usamos SELECT para especificar las columnas que se deben incluir en el resultado. Posteriormente usamos un FROM donde se especifica las tablas de las que se deben recuperar los datos ya que están relacionadas a través de claves foráneas. Finalmente usamos un WHERE para especificar la condición de búsqueda que debe cumplirse para que se incluyan las filas en el resultado obteniendo únicamente los medicamentos recetados a los pacientes y unimos.

```

2 • SELECT procedimiento_medico.paciente.nombre_paciente, procedimiento_medico.medimento.nombre_medimento, procedimiento_medico.medimento.dosis_medimento
3 FROM procedimiento_medico.paciente, procedimiento_medico.receta_medimento, procedimiento_medico.medimento
4 WHERE procedimiento_medico.paciente.id_paciente = procedimiento_medico.receta_medimento.id_paciente
5 AND procedimiento_medico.receta_medimento.id_medimento = procedimiento_medico.medimento.id_medimento;
6

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nombre_paciente	nombre_medimento	dosis_medimento	
▶ María del Rosario	Paracetamol	500mg	
María del Rosario	Amoxicilina	500mg	
Oréste Mercouri	Ibuprofeno	200mg	
Beatriz Pinzon	Aspirina	100mg	
Beatriz Pinzon	Omeprazol	20mg	

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

Utilizamos un SELECT para especificar los campos que se desean obtener debido a que la consulta busca relacionar el nombre del enfermero con el tipo de procedimiento que asistió. Utilizamos una cláusula FROM para especificar las tablas en las que se buscarán los datos y relacionamos estas dos tablas usando una cláusula WHERE para comparar el valor de la columna id_enfermero de la tabla enfermero" con el valor de la columna id_enfermero de la tabla procedimiento así nos aseguramos de que solo se seleccionen los procedimientos realizados por los enfermeros que están registrados en la tabla enfermero.

```

2 • SELECT procedimiento_medico.enfermero.nombre_enfermero, procedimiento_medico.procedimiento.tipo_procedimiento
3 FROM procedimiento_medico.enfermero, procedimiento_medico.procedimiento
4 WHERE procedimiento_medico.enfermero.id_enfermero = procedimiento_medico.procedimiento.id_enfermero
5

```

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

nombre_enfermero	tipo_procedimiento
Juan Reyes	Cirugía de corazón
Rubi Ochoa	Quimioterapia
Nicandro Mejia	Extracción de apéndice
Soraya Montenegro	Terapia ocupacional
Paola Bracho	Radiografía de pulmones

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

1. Vista que muestra el nombre del paciente, el nombre del procedimiento, el nombre del medico que realiza el procedimiento y el nombre del enfermero que asiste.

Esta vista me parece importante ya que si un médico o enfermero está involucrado en varios procedimientos programados al mismo tiempo se podría ajustar su horario para asegurarse de que puedan cumplir con todos los procedimientos.

```

2 • CREATE VIEW vista_procedimientos AS
3 SELECT
4     procedimiento_medico.paciente.nombre_paciente,
5     procedimiento_medico.procedimiento.tipo_procedimiento,
6     procedimiento_medico.medico.nombre_medico,
7     procedimiento_medico.enfermero.nombre_enfermero
8 FROM
9     procedimiento_medico.paciente
10    INNER JOIN procedimiento_medico.procedimiento ON procedimiento_medico.paciente.id_paciente = procedimiento_medico.procedimiento.id_paciente
11    INNER JOIN procedimiento_medico.medico ON procedimiento_medico.procedimiento.id_medico = procedimiento_medico.medico.id_medico
12    INNER JOIN procedimiento_medico.enfermero ON procedimiento_medico.procedimiento.id_enfermero = procedimiento_medico.enfermero.id_enfermero ;
13 • SELECT * FROM vista_procedimientos;
14

```

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

nombre_paciente	tipo_procedimiento	nombre_medico	nombre_enfermero
María del Rosario	Quimioterapia	Freddie Mercury	Rubi Ochoa
Oréstes Mercouri	Extracción de apéndice	Michael Jackson	Nicandro Mejia
Beatriz Pinzon	Cirugía de corazón	Paola Jara	Juan Reyes
Catalina Santana	Radiografía de pulmones	Paulina Rubio	Paola Bracho
Fernando Colunga	Terapia ocupacional	Elton John	Soraya Montenegro

2. Vista que muestra el nombre de paciente, nombre de medicamento, dosis de medicamento y el nombre del doctor que lo receto

Esta vista me parece que información de los medicamentos que se han recetado a cada paciente para garantizar la administración adecuada y la dosificación de los medicamentos recetados a los pacientes por los médicos.

```
2 • CREATE VIEW vista_medicamentos_prescripcion AS
3 SELECT
4     procedimiento_medico.medico.nombre_medico,
5     procedimiento_medico.paciente.nombre_paciente,
6     procedimiento_medico.medificacion.nombre_medificacion,
7     procedimiento_medico.medificacion.dosis_medificacion
8 FROM
9     procedimiento_medico.medico
10    JOIN procedimiento_medico.procedimiento ON procedimiento_medico.medico.id_medico = procedimiento_medico.procedimiento.id_medico
11    JOIN procedimiento_medico.paciente ON procedimiento_medico.procedimiento.id_paciente = procedimiento_medico.paciente.id_paciente
12    JOIN procedimiento_medico.receta_medificacion ON procedimiento_medico.paciente.id_paciente = procedimiento_medico.receta_medificacion.id_paciente
13    JOIN procedimiento_medico.medificacion ON procedimiento_medico.receta_medificacion.id_medificacion = procedimiento_medico.medificacion.id_medificacion;
14 • SELECT * FROM vista_medicamentos_prescripcion;
```

nombre_medico	nombre_paciente	nombre_medificacion	dosis_medificacion
Freddie Mercury	María del Rosario	Paracetamol	500mg
Freddie Mercury	María del Rosario	Amoxicilina	500mg
Paola Jara	Beatriz Pinzon	Aspirina	100mg
Paola Jara	Beatriz Pinzon	Omeprazol	20mg
Michael Jackson	Oréste Mercouri	Ibuprofeno	200mg

3. Vista que muestra el nombre de enfermero y su numero de contacto junto al paciente que tiene a cargo.

Esta vista me parece importante ya que si un paciente o familiar del paciente requiere ponerse en contacto con el enfermero encargado se puede verificar el numero de contacto fácilmente.

```
2 • CREATE VIEW vista_pacientes_enfermeros AS
3 SELECT
4     procedimiento_medico.paciente.nombre_paciente,
5     procedimiento_medico.enfermero.nombre_enfermero,
6     procedimiento_medico.contacto_enfermero.telefono_enfermero
7 FROM
8     procedimiento_medico.paciente
9    INNER JOIN procedimiento_medico.procedimiento ON procedimiento_medico.paciente.id_paciente = procedimiento_medico.procedimiento.id_paciente
10   INNER JOIN procedimiento_medico.enfermero ON procedimiento_medico.procedimiento.id_enfermero = procedimiento_medico.enfermero.id_enfermero
11   INNER JOIN procedimiento_medico.contacto_enfermero ON procedimiento_medico.contacto_enfermero.id_enfermero = procedimiento_medico.enfermero.id_enfermero;
12 • SELECT * FROM vista_pacientes_enfermeros;
```

nombre_paciente	nombre_enfermero	telefono_enfermero
María del Rosario	Rubi Ochoa	2222222
Oréste Mercouri	Nicandro Mejía	3333333
Beatriz Pinzon	Juan Reyes	1111111
Catalina Santana	Paola Bracho	5555555
Fernando Colunga	Soraya Montenegro	4444444

Actividad 03

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

Procedimiento Agregar

Para este procedimiento agregaremos un nuevo registro en la tabla libro correspondiente a la Librería, se crea el procedimiento con los campos requeridos (ISBN, titulo, numero de paginas y editorial).

```
2 DELIMITER //
3 CREATE PROCEDURE agregar_libro(IN isbn VARCHAR(10),IN titulo VARCHAR(45),IN num_paginas VARCHAR(45),IN editorial VARCHAR(50)
4 )
5 BEGIN
6     INSERT INTO libro (ISBN, titulo, numero_paginas, nombre_editorial)
7     VALUES (isbn, titulo, num_paginas, editorial);
8 END
9
10 //
11 DELIMITER ;
12 CALL agregar_libro("131", "100 años de soledad", "365", "Editorial Alfaguara");
13
14
```

Una vez creado el procedimiento agregamos un nuevo registro haciendo un llamado y verificamos que este nuevo registro se encuentre en la tabla Libro.

128	La vida es sueño	230	Editorial Alianza
129	Los premios	400	Editorial Debolsillo
130	The Beautiful and Damned	350	Editorial Lumen
131	100 años de soledad	365	Editorial Alfaguara
* NULL	NULL	NULL	NULL

Procedimiento actualizar

Para este procedimiento vamos a actualizar el ISBN de uno de los libros ya ingresados, así que procedemos con la creación del mismo utilizando el campo de ISBN actual y agregando un nuevo valor de ISBN a actualizar.

```

3 • CREATE PROCEDURE actualizar_libro(IN isbn_actualizar VARCHAR(10),IN isbn_nuevo VARCHAR(10),IN titulo VARCHAR(45),IN num_paginas VARCHAR(45),IN editorial VARCHAR(50)
4 )
5 BEGIN
6     UPDATE libro
7     SET ISBN = isbn_nuevo, titulo = titulo, numero_paginas = num_paginas, nombre_editorial = editorial
8     WHERE ISBN = isbn_actualizar;
9 END
10
11
12 //
13 DELIMITER ;
14 • CALL actualizar_libro("131","132", "100 años de soledad", "365", "Editorial Alfaguara");
15
16

```

Una vez se realiza el llamado del procedimiento verificamos si el ISBN del libro fue actualizado en la tabla.

129	Los premios	400	Editorial Debolsillo
130	The Beautiful and Damned	350	Editorial Lumen
132	100 años de soledad	365	Editorial Alfaguara
NULL	NULL	NULL	NULL



Procedimiento consultar

Para este procedimiento haremos uso del titulo del libro presente en la tabla libro, al recibir el parámetro de la consulta el procedimiento realiza una validación dentro de la tabla y nos arroja la coincidencia si existe dentro de la tabla.

```

2 DELIMITER //
3 • CREATE PROCEDURE consultar_libro(IN buscar_titulo VARCHAR(45))
4 BEGIN
5     SELECT * FROM libro WHERE titulo = buscar_titulo;
6 END
7 //
8 DELIMITER ;
9 • CALL consultar_libro("El coronel no tiene quien le escriba");
10
11

```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
ISBN	titulo	numero_paginas	nombre_editorial
120	El coronel no tiene quien le escriba	250	Editorial Alianza

Procedimiento eliminar

Para este procedimiento se utilizara el valor de ISBN como parámetro para definir el registro a eliminar dentro de la tabla.

```
• CREATE PROCEDURE borrar_libro(IN isbn_borrar VARCHAR(10))
  )
  BEGIN
    DELETE FROM libro
    WHERE ISBN = isbn_borrar;
  END

  //
  DELIMITER ;
• CALL borrar_libro("132");
```

Una vez se realiza el llamado del procedimiento podemos verificar dentro de la tabla que el registro fue eliminado.

128	La vida es sueño	230	Editorial Alianza
129	Los premios	400	Editorial Debolsillo
130	The Beautiful and Damned	350	Editorial Lumen
NULL	NULL	NULL	NULL

pro 4 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.**

Inicialmente realizaremos la creación de la tabla, para ello indicamos la DB en la que vamos a trabajar y realizamos la creación de la tabla control_de_cambios_libreria y se crean las columnas solicitadas, para la fecha se utilizará el tipo de dato DATETIME.

```
1 • USE libreriaescalibre;
2 • CREATE TABLE control_de_cambios_libreria (
3     usuario_cambio VARCHAR(50) NOT NULL,
4     cambio_realizado VARCHAR(50) NOT NULL,
5     fecha DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
6 );
7
```


Posteriormente realizamos la creación de los correspondientes triggers empezando por el trigger que nos generara un nuevo registro en la tabla al agregar datos a la tabla libro.

```
3 • CREATE TRIGGER control_de_cambios_insert
3 AFTER INSERT ON libro
3 FOR EACH ROW
1 INSERT INTO control_de_cambios_libreria()
2 VALUES (USER(), 'insertar un registro',now());
3
4
5
```

Al insertar un nuevo registro en la tabla libro el trigger creara un registro en la tabla de control de cambios con el usuario, la acción realizada e incluirá la fecha y hora cuando se realizo el cambio, para validar su funcionamiento utilizaremos el procedimiento creado de agregar libro para agregar un registro y verificar que el trigger funcione.

```
16
17 • CALL agregar_libro ("131","100 años de soledad", "365", "Editorial Alfaguara");
18
```

Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
	usuario_cambio	cambio_realizado	fecha
▶	root@localhost	insertar un registro	2023-02-14 22:10:16

Al usar el procedimiento se creo un nuevo registro dentro de la tabla de control.

Ahora realizaremos el trigger que generara un registro en la tabla de control al eliminar un registro de la tabla libro.

```
8 • CREATE TRIGGER control_de_cambios_delete
9 AFTER DELETE ON libro
10 FOR EACH ROW
11 INSERT INTO control_de_cambios_libreria()
12 VALUES (USER(), 'eliminar un registro',now());
13
```

Al eliminar un registro en la tabla libro el trigger creara un registro en la tabla de control de cambios con el usuario similar al que se genera al agregar registros pero el mensaje de acción será diferente, para validar su funcionamiento utilizaremos el procedimiento creado de eliminar libro para eliminar un registro y verificar que el trigger funcione.

```

16
17 • CALL borrar_libro ("131");
18
19

```

	usuario_cambio	cambio_realizado	fecha
▶	root@localhost	insertar un registro	2023-02-14 22:10:16
	root@localhost	eliminar un registro	2023-02-14 22:14:50

Una vez realizado el procedimiento podemos validar que se genero satisfactoriamente el registro en la tabla.

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

Procedimiento Agregar

Para este procedimiento agregaremos un nuevo registro en la tabla enfermero correspondiente al Procedimiento Medico, se crea el procedimiento con los campos requeridos (nombre de enfermero y su respectivo ID).

```

DELIMITER //
• CREATE PROCEDURE agregar_enfermero(IN nombre VARCHAR(65), IN id VARCHAR(15))
BEGIN
    INSERT INTO enfermero (nombre_enfermero, id_enfermero) VALUES (nombre, id);
END
//
DELIMITER ;
• CALL agregar_enfermero("Nicolas Mora", "E006");

```

Una vez creado hacemos llamado al procedimiento y validamos que se encuentre un nuevo registro en la tabla.

	id_enfermero	nombre_enfermero
▶	E001	Juan Reyes
	E002	Rubi Ochoa
	E003	Nicandro Mejia
	E004	Soraya Montenegro
	E005	Paola Bracho
	E006	Nicolas Mora
✱	NULL	NULL

Procedimiento actualizar

Para este procedimiento vamos a actualizar el nombre de uno de los enfermeros ya ingresados, así que procedemos con la creación del mismo utilizando el campo de nombre actual y utilizando la sentencia UPDATE y el comando SET para reemplazar el nombre de un ID ya existente.

```
21 DELIMITER //
```

```
22 • CREATE PROCEDURE actualizar_enfermero(IN id VARCHAR(15), IN nombre VARCHAR(65))
```

```
23 BEGIN
```

```
24     UPDATE enfermero SET nombre_enfermero = nombre WHERE id_enfermero = id;
```

```
25 END
```

```
26 //
```

```
27 DELIMITER ;
```

```
28 • CALL actualizar_enfermero("E005", "Pedro Pascal");
```

Una vez hecho el llamado del procedimiento validamos en la tabla que haya sido realizado con éxito.

	id_enfermero	nombre_enfermero
▶	E001	Juan Reyes
	E002	Rubi Ochoa
	E003	Nicandro Mejia
	E004	Soraya Montenegro
	E005	Pedro Pascal
	E006	Nicolas Mora
*	NULL	NULL

Procedimiento consultar

Para este procedimiento haremos uso del ID de enfermero presente en la tabla enfermero, al recibir el parámetro de la consulta el procedimiento realiza una validación dentro de la tabla y nos arroja la coincidencia si existe dentro de la tabla.

```
21 DELIMITER //
```

```
22 • CREATE PROCEDURE consultar_enfermero(IN id VARCHAR(15))
```

```
23 BEGIN
```

```
24     SELECT * FROM enfermero WHERE id_enfermero = id;
```

```
25 END
```

```
26 //
```

```
27 DELIMITER ;
```

```
28 • CALL consultar_enfermero("E002");
```

```
29
```

<	Result Grid	Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
	id_enfermero	nombre_enfermero		
▶	E002	Rubi Ochoa		

Procedimiento eliminar

Para este procedimiento se utilizara el valor de ID como parámetro para definir el registro a eliminar dentro de la tabla.

```
21 DELIMITER //
22 • CREATE PROCEDURE borrar_enfermero(IN id VARCHAR(15))
23 BEGIN
24     DELETE FROM Enfermero WHERE id_enfermero = id;
25 END
26 //
27 DELIMITER ;
28 • CALL borrar_enfermero("E006");
```

Una vez hecho el llamado del procedimiento validamos que el cambio haya sido realizado en la tabla.

	id_enfermero	nombre_enfermero
▶	E001	Juan Reyes
	E002	Rubi Ochoa
	E003	Nicandro Mejia
	E004	Soraya Montenegro
	E005	Pedro Pascal
*	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.

Inicialmente realizaremos la creación de la tabla, para ello indicamos la DB en la que vamos a trabajar y realizamos la creación de la tabla control_de_cambios_procedimiento_medico y se crean las columnas solicitadas, para la fecha se utilizará el tipo de dato DATETIME.

```
1 • USE procedimiento_medico;
2 • CREATE TABLE control_de_cambios_procedimiento_medico (
3     usuario_cambio VARCHAR(65) NOT NULL,
4     cambio_realizado VARCHAR(50) NOT NULL,
5     fecha DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
6 );
7
8
9
```



Posteriormente realizamos la creación de los correspondientes triggers empezando por el trigger que nos generara un nuevo registro en la tabla al agregar datos a la tabla enfermero.

```
1 • CREATE TRIGGER control_de_cambios_insert AFTER INSERT ON enfermero
2   FOR EACH ROW
3   INSERT INTO control_de_cambios_procedimiento_medico ()
4   VALUES (USER(), 'Se inserto un registro', NOW());
5
6 • CALL agregar_enfermero("Armando Mendoza", "E006");
```

Al insertar un nuevo registro en la tabla enfermero el trigger creara un registro en la tabla de control de cambios con el usuario, la acción realizada e incluirá la fecha y hora cuando se realizo el cambio, para validar su funcionamiento utilizaremos el procedimiento creado de agregar un enfermero para agregar un registro y verificar que el trigger funcione.

	id_enfermero	nombre_enfermero
▶	E001	Juan Reyes
	E002	Rubi Ochoa
	E003	Nicandro Mejia
	E004	Soraya Montenegro
	E005	Pedro Pascal
	E006	Armando Mendoza
*	NULL	NULL

```
1 • SELECT * FROM procedimiento_medico.control_de_cambios_procedimiento_medico;
```

<		
Result Grid	Filter Rows: <input type="text"/>	Export:  Wrap Cell Content: 
usuario_cambio	cambio_realizado	fecha
▶ root@localhost	Se inserto un registro	2023-02-14 22:32:19

Al usar el procedimiento se creo un nuevo registro dentro de la tabla de control.

Ahora realizaremos el trigger que generara un registro en la tabla de control al eliminar un registro de la tabla enfermero.

```
8 • CREATE TRIGGER control_de_cambios_borrado AFTER DELETE ON enfermero
9   FOR EACH ROW
10  INSERT INTO control_de_cambios_procedimiento_medico ()
11  VALUES (USER(), 'Se ha borrado un registro', NOW());
12
13 • CALL borrar_enfermero("E006");
14
```

Al eliminar un registro en la tabla enfermero el trigger creara un registro en la tabla de control de cambios con el usuario similar al que se genera al agregar registros pero el mensaje de acción será diferente, para validar su funcionamiento utilizaremos el procedimiento creado de eliminar un enfermero de la tabla eliminando un registro y verificando que el trigger funcione.

	usuario_cambio	cambio_realizado	fecha
▶	root@localhost	Se inserto un registro	2023-02-14 22:41:38
	root@localhost	Se ha borrado un registro	2023-02-14 22:41:46

- Después de realizar el trabajo responda ¿Qué le agregaría al modelo para dar mas información y esa información cuál seria?

Agregaría más atributos a la entidad paciente, donde se pueda tener una información más detallada sobre el paciente que es necesaria para los procedimientos médicos, adicionalmente agregaría una entidad que contenga historial médico del paciente donde se pueda tener la información sobre el historial médico de los pacientes, diagnósticos, tratamientos, cirugías y hospitalizaciones anteriores, alergias y medicamentos recetados.