



PRESENTADO POR:

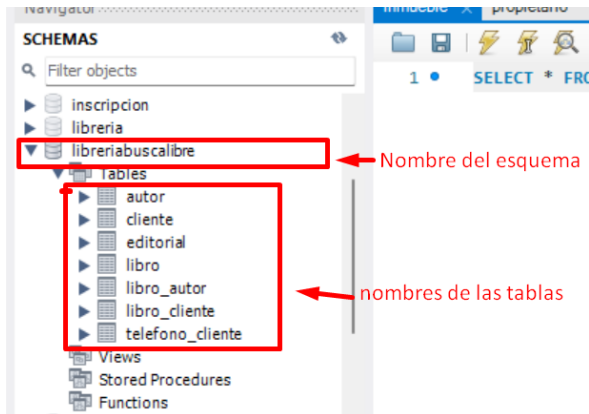
NEVARDO ANTONIO OSPINA Z.

COACH:

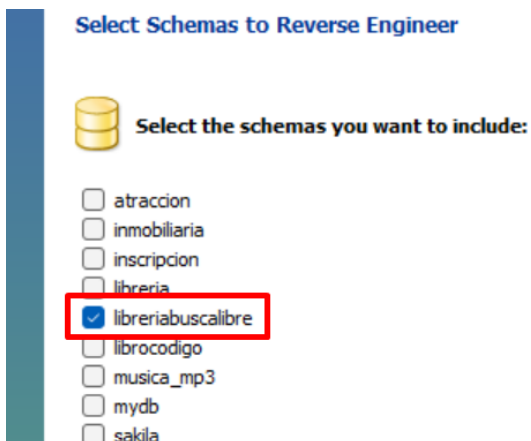
JUAN ESTEBAN PINEDA ANGEL

14/02/2023

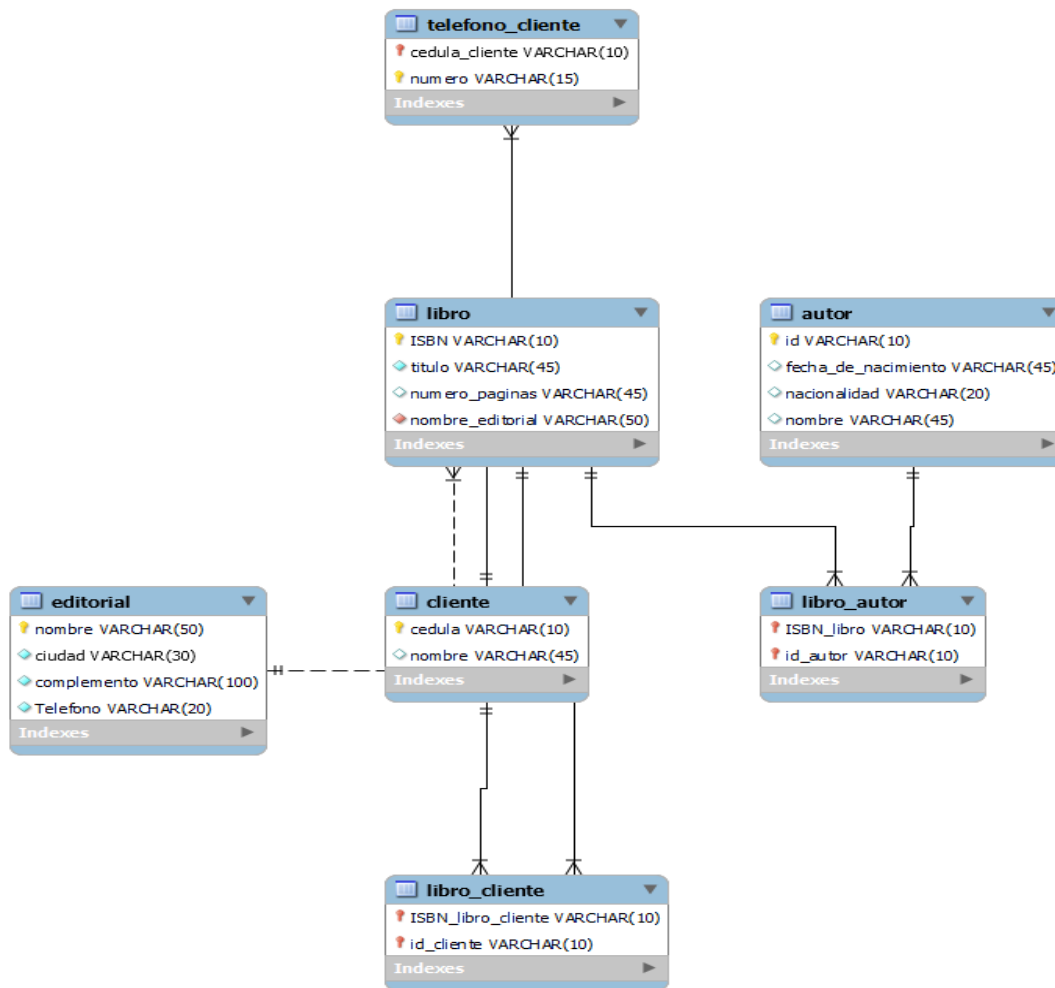
Consultas, Vistas, Procedimientos y Trigger



Seleccionamos el nombre del esquema para generar la tabla modelo relacional través de la ingeniería inversa.



M-R



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

1. 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.

Se insertan los datos de los diferentes campos de la tabla autor

```

2 • INSERT INTO `libreriabuscalibre`.`autor` (`fecha_de_nacimiento`, `nacionalidad`, `nombre`)
3 VALUES ('11/07/2012', 'Mexicano', 'Emilia Martinez');

```

Registros de la tabla Cliente

Result Grid				
Filter Rows:				
Edit:				
	id	fecha_de_nacimiento	nacionalidad	nombre
	01	19/09/2000	Colombiana	Antonio Ospina
	02	12/10/2009	Chileno	Angelica Mendez
	03	20/09/2005	Mexicano	Maria Zuñiga
	04	19/09/1900	Panameño	Luis Miguel Fuente
▶	05	11/07/2012	Mexicano	Emilia Martinez
•	NULL	NULL	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Cliente

```

INSERT INTO `libreriabuscalibre`.`cliente` (`cedula`, `nombre`) VALUES ('1000001', 'Luis M');
INSERT INTO `libreriabuscalibre`.`cliente` (`cedula`, `nombre`) VALUES ('1000002', 'Luisa Morales');

```

Registros de la tabla Cliente

Result Grid		
Filter Rows:		
	cedula	nombre
▶	1000001	Luis M
	10000010	David Montes
	10000011	Samara Berrio
	10000012	Jorge Luis Espitia
	10000013	Diana Campiño
	10000014	Maria Jose Marino
	10000015	Monica Martin
	10000016	Josefina diaz
	10000017	Diover Quintero
	10000018	Juanita Zuñiga
	10000019	Nevardo Antonio...
	1000002	Luisa Morales
	10000020	Anthony Vitola
	1000003	Fernanda Lopez
	1000004	Ana sofia Lopez
	1000005	Juan Manuel diaz
	1000006	Emilia Hernandez

Se insertan los datos de los diferentes campos de la tabla Editorial

```

INSERT INTO `libreriabuscalibre`.`editorial` (`nombre`, `ciudad`, `complemento`, `Telefono`) VALUES
('Acantilado', 'Pereira', 'Calle 27c # 12-50', '3447575');

```

Se inserta información a la tabla editorial

Registros de la tabla editorial

Result Grid

Filter Rows:

Edit:

Export

	nombre	ciudad	complemento	Telefono
▶	Acantilado	Pereira	Calle 27c # 12-50	3447575
	Aguilar	Manizales	Calle 30c # 123-50	76665554
	Alba	Medellin	Calle 666 # 456-150	66633344
	Alfaguara	Manizales	Calle 445 # 553-550	711117776654
	Alianza	Monteria	Calle 80c # 93-502	1112554
	Alkal	Santa Rosa de Cabal	Calle 45 # 823-550	78887764
	Almadia	Cartagena	Carrera 30c # 923-230	6655444
✱	NULL	NULL	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla libro

```
1 • INSERT INTO `libreriaescalibre`.`libro` (`ISBN`, `titulo`, `numero_paginas`, `nombre_editorial`)
2   VALUES ('A001', 'Viaje al fin de la noche', '77', 'Acantilado');
~
```

Registros de la tabla libro

Result Grid				
		Filter Rows:		
		Edit:		
	ISBN	titulo	numero_paginas	nombre_editorial
▶	A000016	Tiempo de migrar al norte	78	Alba
	A000120	Al faro	665	Acantilado
	A00017	El rey Lear	66	Acantilado
	A00018	Rojo y negro	777	Aguilar
	A00019	Memorias de Adriano	66	Alba
	A001	Viaje al fin de la noche	77	Acantilado
	A0010	El proceso	44	Alba
	A0011	Hijos y amantes	56	Aguilar
	A0012	Pippi Calzaslargas	76	Aguilar
	A00120	Al faro	665	Acantilado
	A0013	Beloved	55	Almadia
	A0014	Lolita	44	Acantilado
	A0015	Bostan	45	Alba

Se insertan los datos de los diferentes campos de la tabla libro_cliente

```

1 • INSERT INTO `libreriaescalibre`.`libro_cliente` (`ISBN_libro_cliente`, `id_cliente`)
2   VALUES ('A001', '1000001'),('A002', '1000002'),('A003', '1000003'),('A004', '1000004'),
3   |

```

Registros de la tabla Libro_cliente

	ISBN_libro_cliente	id_cliente
▶	A001	1000001
	A002	10000010
	A003	10000011
	A004	10000012
	A005	10000013
	A006	10000014
	A007	10000015
	A008	10000016
	A009	10000017
	A0010	10000018
	A0011	10000019
	A0013	1000002
	A0012	10000020
	A0014	1000003
	A0015	1000004
	A000016	1000005
	A00017	1000006

Se insertan los datos de los diferentes campos de la tabla libro_autor

```
1 • INSERT INTO `libreriaescalibre`.`libro_autor` (`ISBN_libro`, `id_autor`)
2   VALUES ('A001', '01'),('A002', '02'),
-
```

Registros de la tabla libro_autor

	ISBN_libro	id_autor
▶	A000016	01
	A000120	01
	A00017	01
	A00018	01
	A00019	01
	A001	01
	A0014	01
	A0015	01
	A006	01
	A007	01
	A008	01
	A0010	02

Se insertan los datos de los diferentes campos de la tabla teléfono_cliente

```
1 • INSERT INTO `libreriaescalibre`.`telefono_cliente` (`cedula_cliente`, `numero`)
2   VALUES ('1000001', '66677788'),('1000002', '666770000'),
3
```

Registros de la tabla teléfono_cliente


	cedula_cliente	numero
▶	1000001	66677788
	10000010	98887654
	10000011	988898
	10000012	988556
	10000013	9886674
	10000014	988334
	10000015	9844555
	10000016	9888777
	10000017	98000998
	10000018	9899932
	10000019	7778887
	1000002	666770000

2. 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.

- Condiciones que permite conocer el nombre y la fecha de nacimiento de cada escritor

```
#Condicion que me permitan conocer el nombre y la fecha de nacimiento de cada escritor.
SELECT nombre, fecha_de_nacimiento from autor;
```


Resultado donde muestra el nombre y la fecha de nacimiento de cada escritor



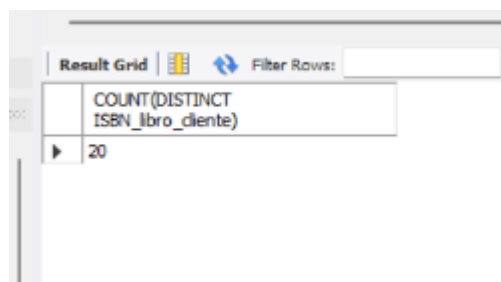
A screenshot of a database application's 'Result Grid'. The grid has two columns: 'nombre' and 'fecha_de_nacimiento'. It contains five rows of data. The first row is highlighted with a mouse cursor. Above the grid, there are icons for 'Result Grid', a grid icon, and a 'Filter Rows' button with a search input field.

nombre	fecha_de_nacimiento
Antonio Ospina	19/09/2000
Angelica Mendez	12/10/2009
Maria Zuñiga	20/09/2005
Luis Miguel Fuente	19/09/1900
Emilia Martinez	11/07/2012

Condición que permita la cantidad de libros diferentes vendidos

```
# la cantidad de libros diferentes vendidos
• SELECT COUNT(DISTINCT ISBN_libro_cliente)
  FROM libro_cliente;
```

Resultado libros diferentes vendidos



A screenshot of a database application's 'Result Grid'. The grid has one column containing the SQL query 'COUNT(DISTINCT ISBN_libro_cliente)'. Below the query, the result '20' is displayed. Above the grid, there are icons for 'Result Grid', a grid icon, and a 'Filter Rows' button with a search input field.

COUNT(DISTINCT ISBN_libro_cliente)
20

Condición que permita tener el nombre de su cliente acompañado de su número telefónico

- ```
El nombre de su cliente acompañado de su numero telefónico
SELECT libreriaescalibre.cliente.nombre, libreriaescalibre.telefono_cliente.numero
FROM libreriaescalibre.cliente
JOIN libreriaescalibre.telefono_cliente
ON libreriaescalibre.cliente.cedula = libreriaescalibre.telefono_cliente.cedula_cliente;
```

## Resultado de la búsqueda

| Result Grid |                    |          | Filter Rows: |
|-------------|--------------------|----------|--------------|
|             | nombre             | numero   |              |
| ▶           | Luis M             | 66677788 |              |
|             | David Montes       | 98887654 |              |
|             | Samara Berrio      | 988898   |              |
|             | Jorge Luis Espitia | 988556   |              |
|             | Diana Campiño      | 9886674  |              |
|             | Maria Jose Marino  | 988334   |              |
|             | Monica Martin      | 9844555  |              |
|             | Josefina diaz      | 9888777  |              |
|             | Diover Quintero    | 98000998 |              |

Condición que permita traer, el nombre del libro acompañado por su autor o sus autores

- ```
#Condición que permita traer, el nombre del libro acompañado por su autor o sus autores
SELECT titulo, GROUP_CONCAT(nombre SEPARATOR ', ') as autores
FROM libro JOIN libro_autor ON libro.ISBN = libro_autor.ISBN_libro
JOIN autor GROUP BY titulo;
```

Resultado de la búsqueda

Result Grid		Filter Rows:	Export
	titulo	autores	
	Los cuentos de Canterbury	Maria Zuñiga	
	Medea	Emilia Martinez	
	Memorias de Adriano	Antonio Ospina	
	Odisea	Antonio Ospina	
	Pippi Calzaslargas	Maria Zuñiga	
	Rojo y negro	Antonio Ospina	
	Tiempo de migrar al norte	Antonio Ospina	
	Ulises	Emilia Martinez	
	Viaje al fin de la noche	Antonio Ospina	

Condición que permita el nombre de las editoriales que han logrado vender libros

```
# Condición que permita el nombre de las editoriales que han logrado vender libros
SELECT DISTINCT titulo
FROM libro JOIN libro_cliente ON libro.ISBN = libro_cliente.ISBN_libro_cliente;
```

Resultado de la búsqueda

Result Grid		Filter F
	nombre_editorial	
►	Acantilado	
	Aguilar	
	Alba	
	Alfaguara	
	Alianza	
	Alkal	
	Almadia	

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

Esta vista contiene la información importante como el número de teléfono y el nombre del cliente que facilitar una pronta reacción al llamar estas personas.

```

1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `libreriaescalibre`.`consultatelefono` AS
6     SELECT
7         `libreriaescalibre`.`cliente`.`nombre` AS `nombre`,
8         `libreriaescalibre`.`telefono_cliente`.`numero` AS `numero`
9     FROM
10        (`libreriaescalibre`.`cliente`
11     JOIN `libreriaescalibre`.`telefono_cliente` ON ((`libreriaescalibre`.`cliente`.`cedula` = `libreriaescalibre`.`tel

```

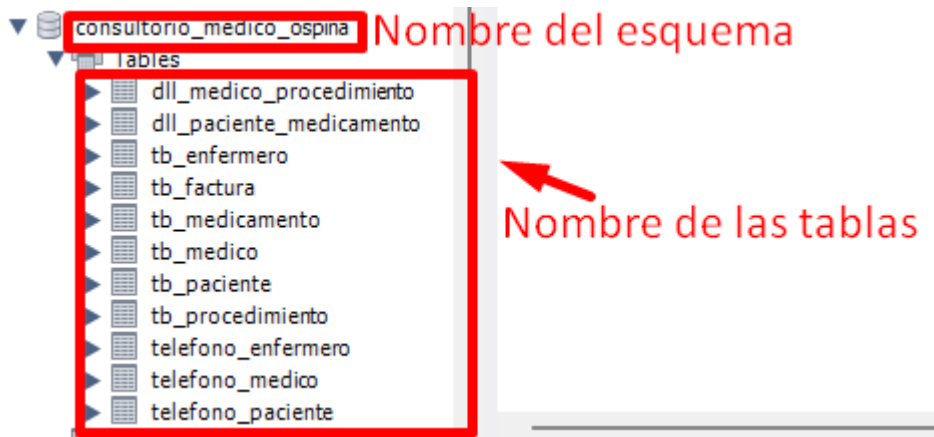
Tabal con los registros.

Result Grid			Filter Rows:	Export:
	nombre	numero		
▶	Luis M	66677788		
	David Montes	98887654		
	Samara Berrio	988898		
	Jorge Luis Espitia	988556		
	Diana Campiño	9886674		

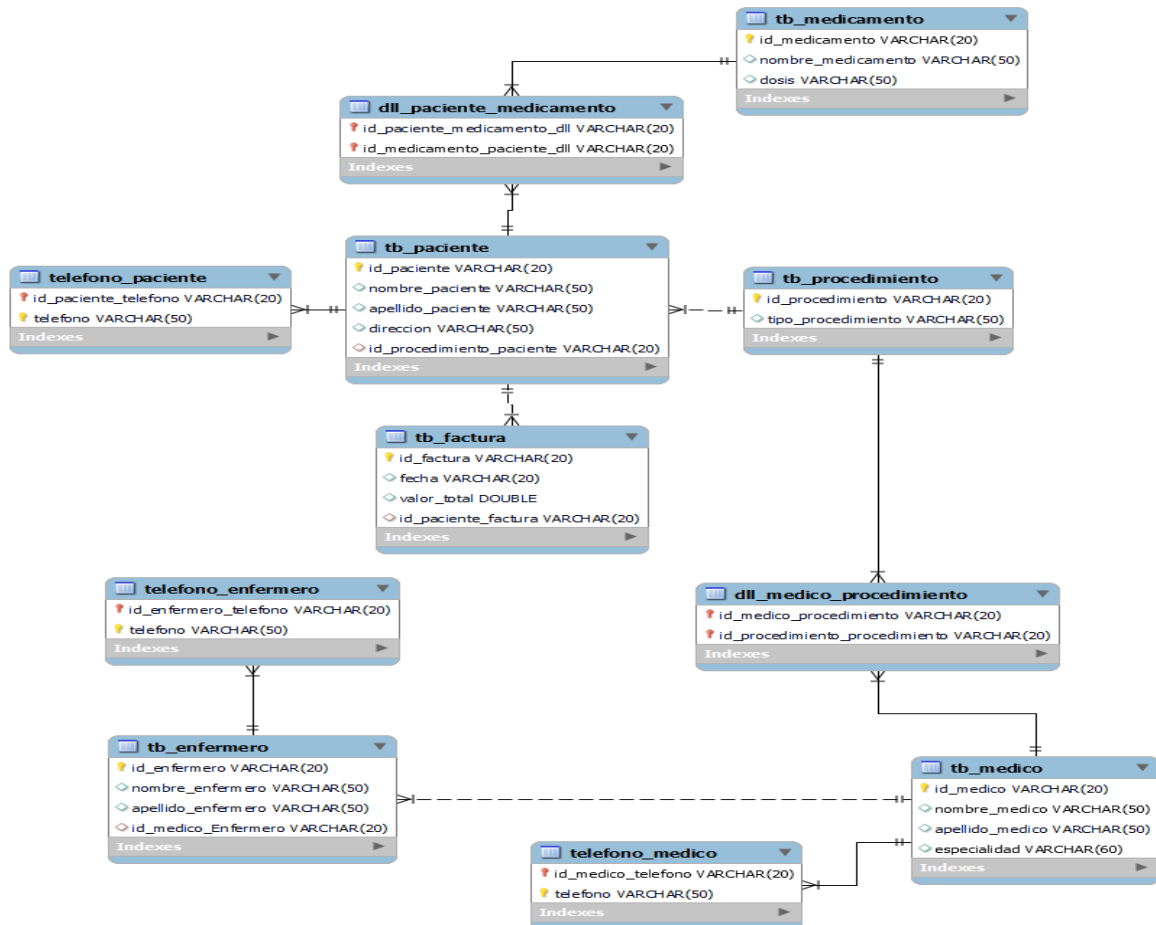
SEGUNDA ACTIVIDAD

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

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



M-R



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

Se insertan los datos de los diferentes campos de la tabla medico

```
SELECT * FROM consultorio_medico_ospina.tb_paciente;  
INSERT INTO tb_medico (`id_medico`,`nombre_medico`,`apellido_medico`,`especialidad`) values  
("100A","Antonio","Ospina","General"),  
("100B","Maria","Lopez","Pidiatra"),  
("100C","Juliana","Marin","Higiene Oral"),  
("100D","Oscar","Higuita","Otorrino"),  
("100E","Emilia","Gonzalez","Cardiologo");
```

Registros de la tabla medico

Result Grid				
Filter Rows: <input type="text"/>				
Edit: 				
	id_medico	nombre_medico	apellido_medico	especialidad
	100B	Maria	Lopez	Pidiatra
	100C	Juliana	Marin	Higiene Oral
	100D	Oscar	Higuita	Otorrino
	100E	Emilia	Gonzalez	Cardiologo
*	NULL	NULL	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla enfermero

```
INSERT INTO `consultorio_medico_ospina`.`tb_enfermero` (`id_enfermero`, `nombre_enfermero`, `apellido_enfermero`, `id_medico_Enfermero`  
("200A", "Laura", "Garcia", "100A"),  
("200B", "Maga", "Oscura", "100B"),  
("200C", "Stuart", "Little", "100C"),  
("200D", "Trunks", "Delfuturo", "100D"),  
("200E", "Power", "Ranger", "100E");
```

Registros de la tabla enfermero

Result Grid  Filter Rows: <input type="text"/> Edit:    Export/Import				
	id_enfermero	nombre_enfermero	apellido_enfermero	id_medico_Enfermero
▶	200A	Laura	Garcia	100A
	200B	Maga	Oscara	100B
	200C	Stuart	Little	100C
	200D	Trunks	Delfuturo	100D
	200E	Power	Ranger	100E
✱	NULL	NULL	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Medicamento

```
INSERT INTO `consultorio_medico_ospina`.`tb_medicamento` (`id_medicamento`, `nombre_medicamento`, `dosis`)  
VALUES ("600A", "Ibuprofeno", "2 al dia"),  
("600B", "Bipirona", "6 cada horas"),  
("600C", "Acetaminofen", "mucho hasta aliviarse"),  
("600D", "Calmidon", "2 veces al dia 5mg"),  
("600E", "Impoio", "Dos veces al dia ");
```

Registros de la tabla medicamento

Result Grid  Filter Rows: <input type="text"/> Edit: 			
	id_medicamento	nombre_medicamento	dosis
▶	600A	Ibuprofeno	2 al ...
	600B	Bipirona	6 ca...
	600C	Acetaminofen	muc...
	600D	Calmidon	2 ve...
	600E	Impoio	Dos ...

medicamento 1 ▼

Se insertan los datos de los diferentes campos de la tabla Procedimiento

```
INSERT INTO `consultorio_medico_ospina`.`tb_procedimiento` (`id_procedimiento`, `tipo_procedimiento`) VALUES  
("300B","Cirugia"),  
("300C","Triage"),  
("300D","Ecografia"),  
("300E","Rx"),  
("300F","Vacuna");
```

Registros de la tabla procedimiento

	id_procedimiento	tipo_procedimiento
▶	300A	Vacuna
	300B	Cirugia
	300C	Triage
	300D	Ecografia
	300E	Rx
•	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Paciente

```
INSERT INTO `consultorio_medico_ospina`.`tb_paciente` (`id_paciente`, `nombre_paciente`, `apellido_paciente`, `direccion`,  
`id_procedimiento_paciente`) VALUES  
("400A","Luis","Gaviria","calle6 #12-121","300A"),  
("400B","Jesus","Diaz","calle8 #124-121","300B"),  
("400C","Maria","Bonilla","calle89 #12-1214","300C"),  
("400D","Patricia","Lopez","calle98 #124-121","300D"),  
("400E","Jose Luis","Loaiza","calle67 #124-121","300E");
```

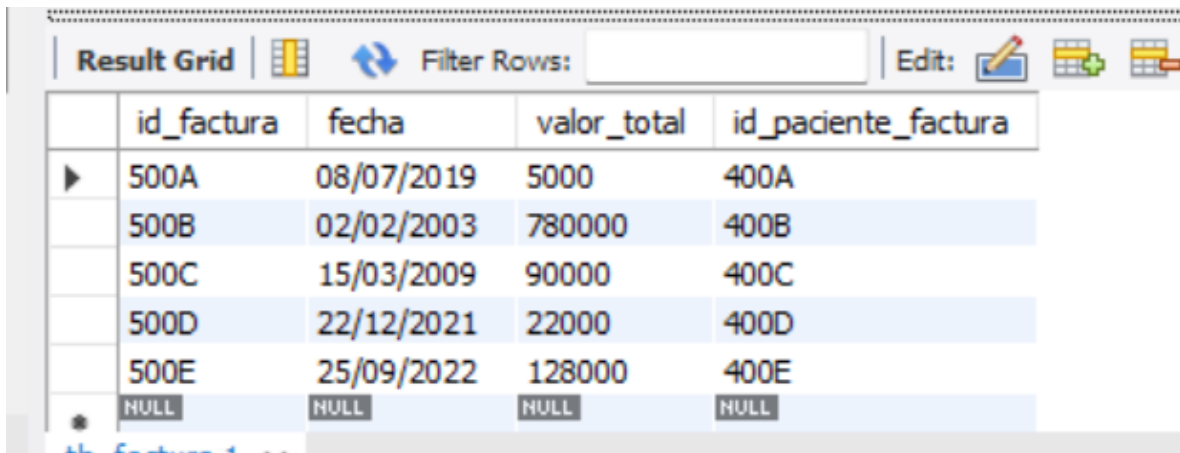
Registros de la tabla paciente

Result Grid Filter Rows: <input type="text"/> Edit: Export/Import: Wrap					
	id_paciente	nombre_paciente	apellido_paciente	direccion	id_procedimiento_paciente
▶	400A	Luis	Gaviria	calle6 #12-121	300A
	400B	Jesus	Diaz	calle8 #124-121	300B
	400C	Maria	Bonilla	calle89 #12-1214	300C
	400D	Patricia	Lopez	calle98 #124-121	300D
	400E	Jose Luis	Loaiza	calle67 #124-121	300E
•	NULL	NULL	NULL	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Factura

```
INSERT INTO `consultorio_medico_ospina`.`tb_factura` (`id_factura`, `fecha`, `valor_total`, `id_paciente_factura`) VALUES  
("500A", "08/07/2019", 5000, "400A"),  
("500B", "02/02/2003", 780000, "400B"),  
("500C", "15/03/2009", 90000, "400C"),  
("500D", "22/12/2021", 22000, "400D"),  
("500E", "25/09/2022", 128000, "400E");
```

Registros de la tabla factura



	id_factura	fecha	valor_total	id_paciente_factura
▶	500A	08/07/2019	5000	400A
	500B	02/02/2003	780000	400B
	500C	15/03/2009	90000	400C
	500D	22/12/2021	22000	400D
	500E	25/09/2022	128000	400E
•	NULL	NULL	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Procedimiento

```
INSERT INTO `consultorio_medico_ospina`.`tb_procedimiento` (`id_procedimiento`) VALUES  
("300A", "Vacuna"),  
("300B", "Cirugia"),  
("300C", "Triage"),  
("300D", "Ecografia"),  
("300E", "Rx");
```

Registros de la tabla procedimiento

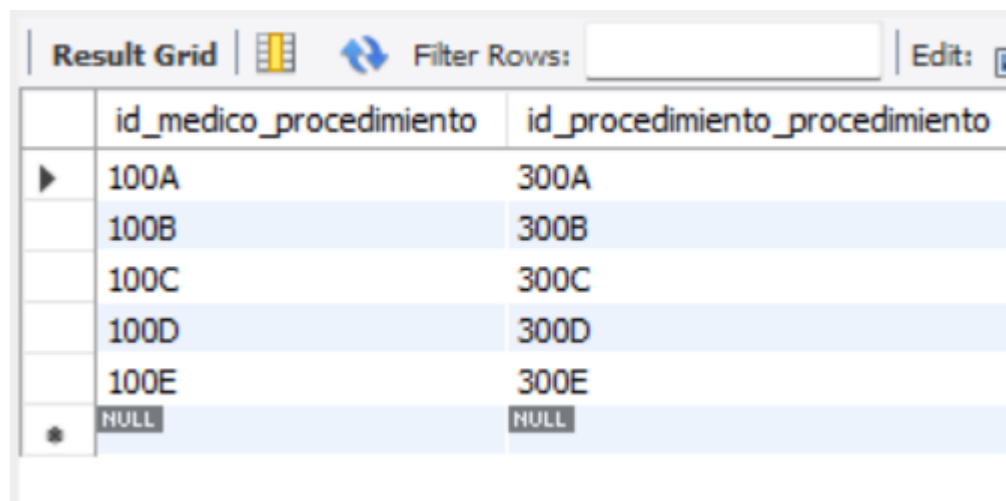


	id_procedimiento	tipo_procedimiento
▶	300A	Vacuna
	300B	Cirugia
	300C	Triage
	300D	Ecografia
	300E	Rx
•	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla
Dll_medico_procedimiento

```
INSERT INTO `consultorio_medico_ospina`.`dll_medico_procedimiento` (`id_medico_procedimiento`,  
`id_procedimiento_procedimiento`) VALUES  
("100A", "300A"),  
("100B", "300B"),  
("100C", "300C"),  
("100D", "300D"),  
("100E", "300E");
```

Registros de la tabla Dll_medico_procedimiento

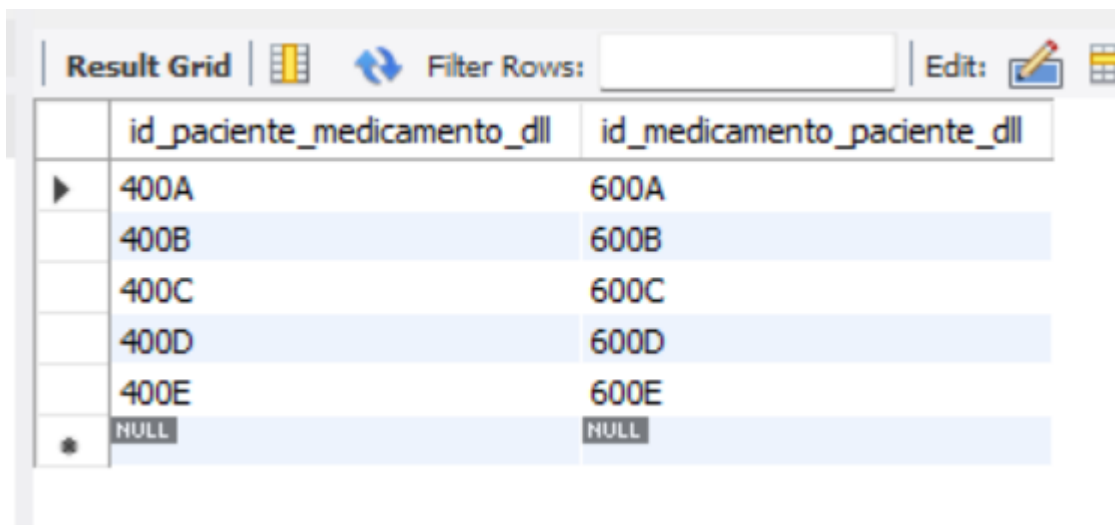


	id_medico_procedimiento	id_procedimiento_procedimiento
▶	100A	300A
	100B	300B
	100C	300C
	100D	300D
	100E	300E
✱	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla
Dll_paciente_medicamento

```
INSERT INTO `consultorio_medico_ospina`.`dll_paciente_medicamento` (`id_paciente_medicamento_dll`,  
`id_medicamento_paciente_dll`) values  
("400A", "600A"),  
("400B", "600B"),  
("400C", "600C"),  
("400D", "600D"),  
("400E", "600E");
```

Registros de la tabla Dll_paciente_medicamento



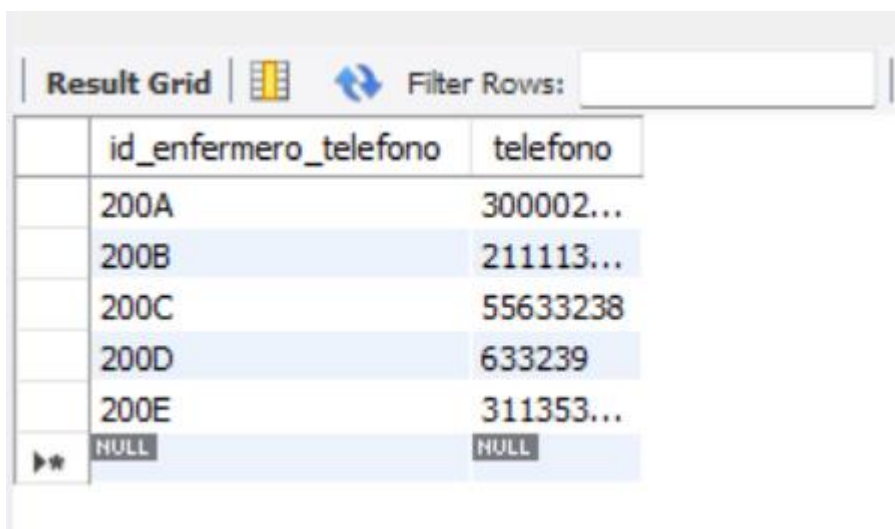
The screenshot shows a database interface with a 'Result Grid' header. Below the header is a table with two columns: 'id_paciente_medicamento_dll' and 'id_medicamento_paciente_dll'. The table contains six rows of data, with the last row showing 'NULL' values. The interface includes a 'Filter Rows' search bar and an 'Edit' button with a pencil icon.

	id_paciente_medicamento_dll	id_medicamento_paciente_dll
▶	400A	600A
	400B	600B
	400C	600C
	400D	600D
	400E	600E
✱	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Teléfono_enfermero

```
INSERT INTO `consultorio_medico_ospina`.`telefono_enfermero` (`id_enfermero_telefono`, `telefono`) VALUES  
("200A", "3000023236"),  
("200B", "2111133237"),  
("200C", "55633238"),  
("200D", "633239"),  
("200E", "311353210");
```

Registros de la tabla Teléfono_enfermero



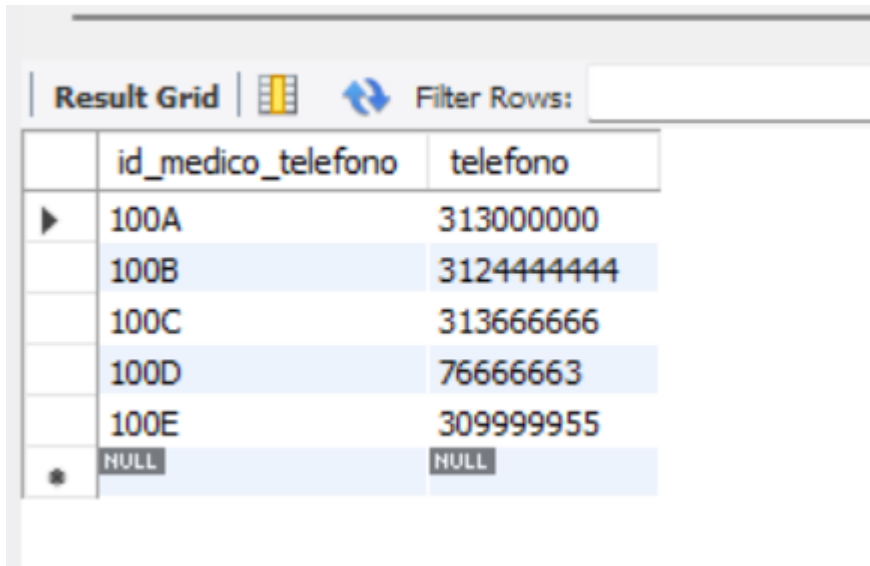
The screenshot shows a database interface with a 'Result Grid' header. Below the header is a table with two columns: 'id_enfermero_telefono' and 'telefono'. The table contains six rows of data, with the last row showing 'NULL' values. The interface includes a 'Filter Rows' search bar.

	id_enfermero_telefono	telefono
	200A	300002...
	200B	211113...
	200C	55633238
	200D	633239
	200E	311353...
▶✱	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Telefono_medico

```
INSERT INTO `consultorio_medico_ospina`.`telefono_medico` (`id_medico_telefono`, `telefono`) VALUES
("100A", "313000000"),
("100B", "3124444444"),
("100C", "313666666"),
("100D", "76666663"),
("100E", "309999955");
```

Registros de la tabla Telefono_medico

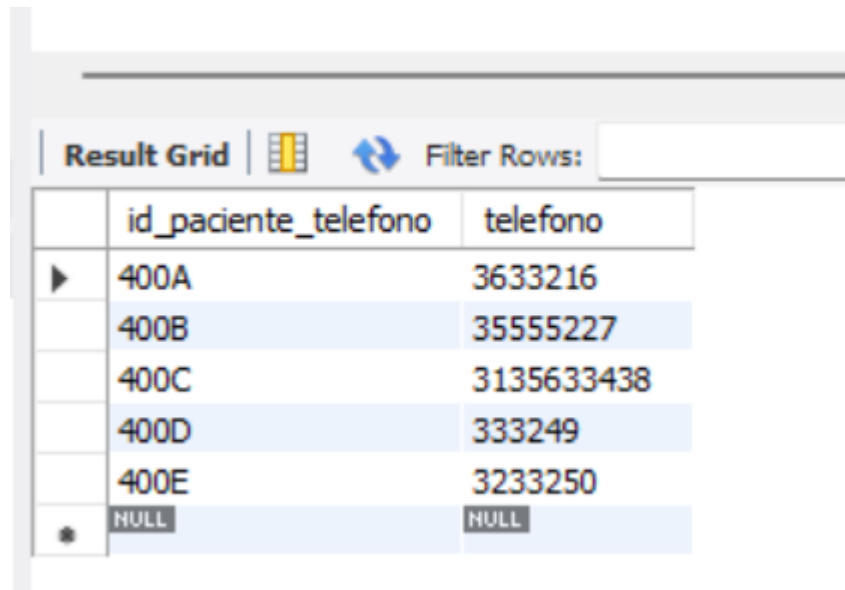


	id_medico_telefono	telefono
▶	100A	313000000
	100B	3124444444
	100C	313666666
	100D	76666663
	100E	309999955
✱	NULL	NULL

Se insertan los datos de los diferentes campos de la tabla Teléfono_paciente

```
INSERT INTO `consultorio_medico_ospina`.`telefono_paciente` (`id_paciente_telefono`, `telefono`) VALUES
("400A", "3633216"),
("400B", "35555227"),
("400C", "3135633438"),
("400D", "333249"),
("400E", "3233250");
```

Registros de la tabla Teléfono_paciente

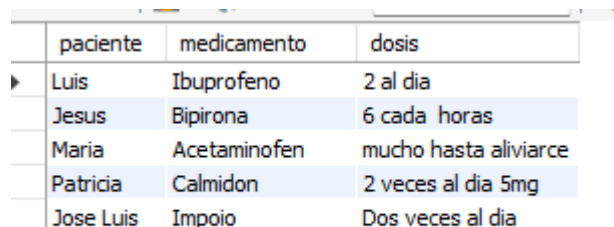


	id_paciente_telefono	telefono
▶	400A	3633216
	400B	35555227
	400C	3135633438
	400D	333249
	400E	3233250
●	NULL	NULL

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

En la siguiente imagen, se muestra la tabla con nombre de los pacientes, medicamentos y dosis, como es requerido por el ejercicio.

```
SELECT tb_paciente.nombre_paciente as "paciente", tb_medicamento.nombre_medicamento as  
"medicamento", tb_medicamento.dosis as "dosis"  
from tb_paciente inner join dll_paciente_medicamento on  
id_paciente=dll_paciente_medicamento.id_paciente_medicamento_dll inner join tb_medicamento on  
id_medicamento = dll_paciente_medicamento.id_medicamento_paciente_dll;
```



	paciente	medicamento	dosis
▶	Luis	Ibuprofeno	2 al dia
	Jesus	Bipirona	6 cada horas
	Maria	Acetaminofen	mucho hasta aliviarse
	Patricia	Calmidon	2 veces al dia 5mg
	Jose Luis	Impoio	Dos veces al dia

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

```
SELECT tb_enfermero.nombre_enfermero as "enfermero", tb_paciente.nombre_paciente as "Paciente",
       tb_procedimiento.tipo_procedimiento as "procedimiento"
from tb_paciente inner join tb_procedimiento on id_procedimiento=tb_procedimiento.id_procedimiento
inner join dll_medico_procedimiento on id_procedimiento_procedimiento=dll_medico_procedimiento.id_procedimiento_procedimiento
inner join tb_medico on id_medico =dll_medico_procedimiento.id_medico_procedimiento
inner join tb_enfermero on id_medico_Enfermero=tb_medico.id_medico;
```

Tabla donde se muestran los resultados solicitados.

Result Grid | Filter Rows:

	enfermero	Paciente	procedimiento
▶	Lina Maria	Luis	Rx
	Lina Maria	Jesus	Rx
	Lina Maria	Maria	Rx
	Lina Maria	Patricia	Rx
	Lina Maria	Jose Luis	Rx
	Lina Maria	Luis	Ecografia
	Lina Maria	Jesus	Ecografia

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

En la siguiente vista tenemos la información completa de los enfermeros, esto con el fin de tener un registro del personal que labora en el consultorio.

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `listado_enfermeros` AS
  SELECT
    `tb_enfermero`.`id_enfermero` AS `id_enfermero`,
    `tb_enfermero`.`nombre_enfermero` AS `nombre_enfermero`,
    `tb_enfermero`.`apellido_enfermero` AS `apellido_enfermero`,
    `tb_enfermero`.`id_medico_Enfermero` AS `id_medico_Enfermero`
  FROM
    `tb_enfermero`;
```

Result Grid			
Filter Rows:			
Export:			
	id_enfermero	nombre_enfermero	apellido_enfermero
▶	200A	Lina Maria	Garcia
	200B	Sifia vergara	Oscuro
	200C	Esperanza Gomez	Little
	200D	Moises	Delfuturo
	200E	Joselit	Ranger

En la siguiente vista, tenemos los registros enfermero, paciente, procedimiento y médico, esto con el fin de tener un control por cada paciente que es atendido en el consultorio y para facilitar la búsqueda de una historia clínica.

```

1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `control de citas medicas` AS
6     SELECT
7         `tb_enfermero`.`nombre_enfermero` AS `Enfermero`,
8         `tb_paciente`.`nombre_paciente` AS `nombre_paciente`,
9         `tb_procedimiento`.`tipo_procedimiento` AS `Procedimiento`,
10        `tb_medico`.`nombre_medico` AS `nombre_medico`
11    FROM
12        ((((`tb_paciente`
13        JOIN `tb_procedimiento` ON ((`tb_paciente`.`id_procedimiento_paciente` = `tb_procedimiento`.`id_procedimiento`)))
14        JOIN `dll_medico_procedimiento` ON ((`tb_procedimiento`.`id_procedimiento` = `dll_medico_procedimiento`.`id_procedimiento_procedimiento`)))
15        JOIN `tb_medico` ON ((`tb_medico`.`id_medico` = `dll_medico_procedimiento`.`id_medico_procedimiento`)))
16        JOIN `tb_enfermero` ON ((`tb_enfermero`.`id_medico_enfermero` = `tb_medico`.`id_medico`)))

```

Result Grid				
Filter Rows:				
Export:				
Wrap C				
	Enfermero	nombre_paciente	Procedimiento	nombre_medico
▶	Lina Maria	Luis	Vacuna	Antonio
	Sifia vergara	Jesus	Cirugia	Maria
	Esperanza Gomez	Maria	Triage	Juliana
	Moises	Patricia	Ecografia	Oscar
	Joselit	Jose Luis	Rx	Emilia

Se crea esta vista solo con el nombre y número telefónico de los médicos que laboran en el hospital, con el fin de tener una pronta comunicación en caso de una emergencia.

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
  VIEW `datos del medico` AS
  SELECT
    `tb_medico`.`nombre_medico` AS `nombre_medico`,
    `telefono_medico`.`telefono` AS `telefono`
  FROM
    (`tb_medico`
  JOIN `telefono_medico` ON ((`tb_medico`.`id_medico` = `telefono_medico`.`id_medico_telefono`)))
```

Result Grid			Filter Rows:
	nombre_medico	telefono	
▶	Antonio	313000000	
	Maria	3124444444	
	Juliana	313666666	
	Oscar	76666663	
	Emilia	309999955	

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).

En la siguiente tabla con el nombre de autor tenemos la información de la consulta.

```
DELIMITER //
```

```
CREATE PROCEDURE listar_autor(IN nombre VARCHAR(10))
```

```
BEGIN
```

```
SELECT * FROM libreriaescalibre.autor WHERE nombre = nombre;
```

```
END //
```

```
DELIMITER ;
```

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
	id	fecha_de_nacimiento	nacionalidad	nombre			
▶	01	19/09/2000	Colombiana	Antonio Ospina			
	02	12/10/2009	Chileno	Angelica Mendez			
	03	20/09/2005	Mexicano	Maria Zuñiga			
	04	19/09/1900	Panameño	Luis Miguel Fuente			
	05	11/07/2012	Mexicano	Emilia Martinez			

En el siguiente procedimiento se crea el registro para ingresar una nueva informacion a la tabla cliente.

```
DELIMITER //
```

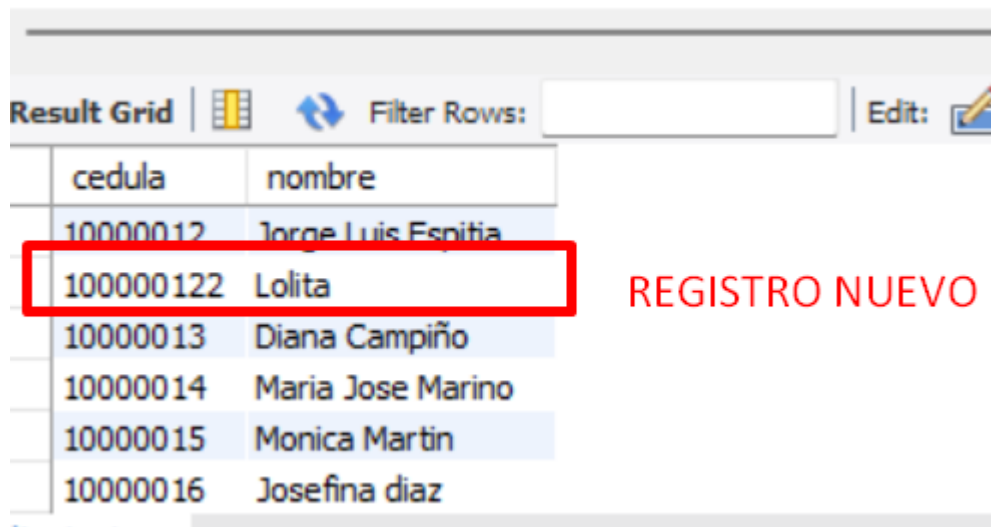
```
CREATE PROCEDURE agregar(IN cedula VARCHAR(10), in nombre VARCHAR(45))
```

```
BEGIN
```

```
INSERT INTO `libreriaescalibre`.`cliente` (`cedula`, `nombre`) VALUES ('100000122', 'Lolita');
```

```
END //
```

```
DELIMITER ;
```



cedula	nombre
10000012	Jorge Luis Espitia
100000122	Lolita
10000013	Diana Campiño
10000014	Maria Jose Marino
10000015	Monica Martin
10000016	Josefina diaz

En el siguiente procedimiento se actualiza los registros a la tabla cliente.

```
DELIMITER //
```

```
CREATE PROCEDURE actualizar(IN cedula VARCHAR(10), in nombre VARCHAR(45))
```

```
BEGIN
```

```
UPDATE `libreriaescalibre`.`cliente` SET `cedula` = '1999977760', `nombre` = 'Flor Maria' WHERE (`cedula` = '100000122');
```

```
END //
```

```
DELIMITER ;
```

cedula	nombre
1000001	Luis M
10000010	David Montes
10000011	Samara Berrio
10000012	Jorge Luis Espitia
100000122	Flor Maria
10000013	Diana Campiño
10000014	Maria Jose Marino

se actualiza el nombre de

Lolita por Flor

En el siguiente procedimiento se elimina los registro para la tabla cliente por medio del numero de cedula.

```

DELIMITER //
CREATE PROCEDURE eliminar(IN cedula VARCHAR(10), in nombre VARCHAR(45))
BEGIN
DELETE FROM `libreriaescalibre`.`cliente` WHERE (`cedula` = '100000122');
END //
DELIMITER ;

```

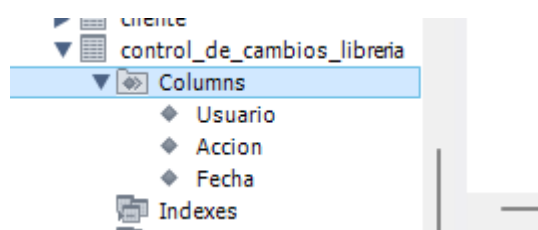
El registro con el numero de cedula 100000122 fue eliminado de la tabla

cedula	nombre
1000001	Luis M
10000010	David Montes
10000011	Samara Berrio
10000012	Jorge Luis Espitia
10000013	Diana Campiño
10000014	Maria Jose Marino
10000015	Monica Martin
10000016	Josefina diaz
10000017	Diover Quintero
10000018	Juanita Zuñiga
10000019	Nevardo Antonio...

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.

En la siguiente imagen se crea la tabla solicitada con el nombre "control_de_cambios_librería".

```
create table control_de_cambios_libreria(  
    Usuario varchar(100), Accion varchar(100), Fecha datetime default current_timestamp  
);
```



```
DELIMITER //  
● CREATE trigger agregar_libro after insert on libro for each row begin  
  insert into control_de_cambios_libreria values  
  (user(), "Agregar un nuevo libro", now());  
end;  
//  
DELIMITER ;  
  
DELIMITER //  
● create trigger eliminar_libro after delete on libro  
  for each row  
  begin insert into control_de_cambios_libreria values (user(), "eliminar libro de la lista", now());  
end;  
//  
DELIMITER ;  
  
● select * from libro;  
● insert into libro values ("A00018", "Rojo y negro", "777", "Aguilar");  
● delete from libro where ISBN="A00125"; select * from control_de_cambios_libreria;
```

Result Grid			
Filter Rows:			
Export:			
Wrap Cell Content:			
	Usuario	Accion	Fecha
▶	root@localhost	Eliminar libro de la lista	2023-02-14 20:22:23

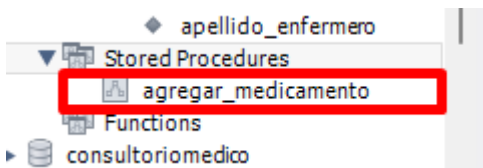
Elabore 4 procedimientos almacenados que me permitan agregar, actualizar, consultar y borrar, en una de las tablas del Hospital (segunda actividad).

Se crea procedimiento para el requerimiento de agregar un medicamento a la tabla medicamento.

```

CREATE DEFINER='root'@'localhost' PROCEDURE `agregar_medicamento`(IN id_medicamento VARCHAR(20),
in nombre_medicamento VARCHAR(50), in dosis VARCHAR(50))
BEGIN
INSERT INTO tb_medicamento values (medicamento,nombre_medicamento,dosis);
END

```



Call stored procedure consultorio_medico_ospina.agregar...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

id_medimento	<input type="text" value="600P"/>	[IN]	VARCHAR(20)
nombre_medimento	<input type="text" value="Naproxeno"/>	[IN]	VARCHAR(50)
dosis	<input type="text" value="de por vida"/>	[IN]	VARCHAR(50)

Execute Cancel

Se muestra el nuevo registro que fue insertado a la tabla.

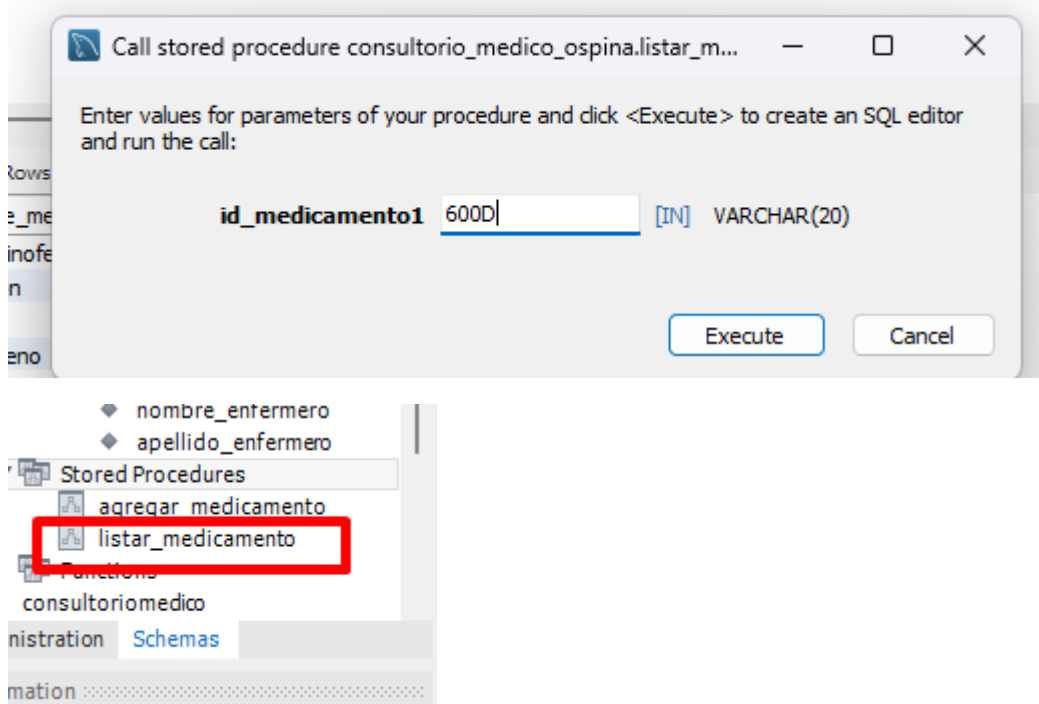
id_medimento	nombre_medimento	dosis
600C	Acetaminofen	mucho hasta aliviarse
600D	Calmidon	2 veces al dia 5mg
600E	Impoio	Dos veces al dia
600P	Naproxeno	de por vida

Se crea procedimiento para el requerimiento de listar medicamentos a la tabla medicamento.

```

1 CREATE PROCEDURE listar_medimento(IN id_medimento1 VARCHAR(20))
2 BEGIN
3     select * from consultorio_medico_ospina.tb_medimento where id_medimento=id_medimento1;
4 END
5

```

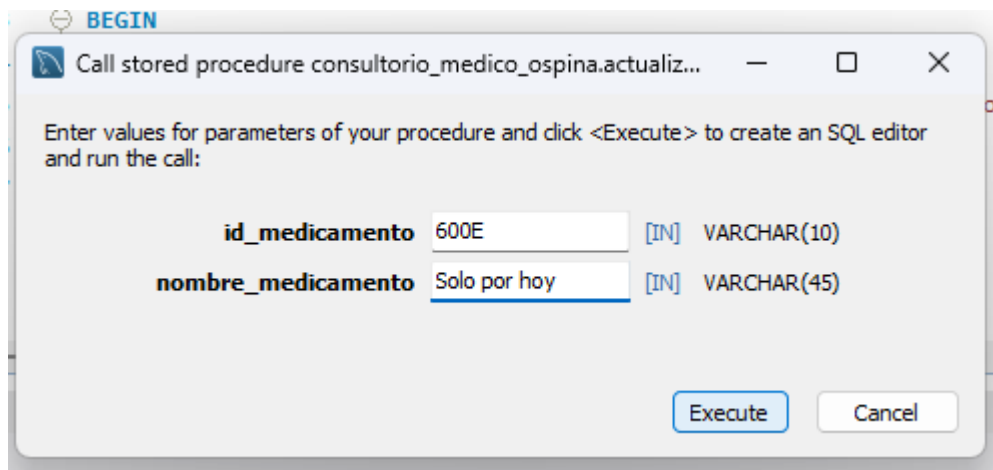
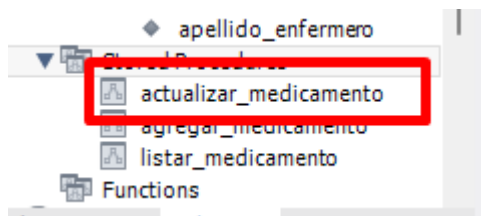


En la imagen se muestra la información del medicamento solicitado por el ID.

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content			
	id_medicamento	nombre_medicamento	dosis
▶	600D	Calmidon	2 veces al dia 5mg

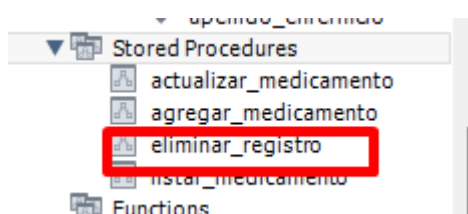
Se crea procedimiento para el requerimiento de actualizar un medicamento a la tabla medicamento.

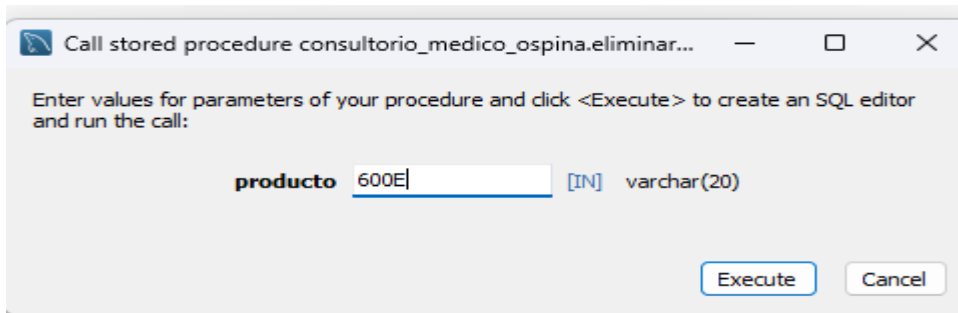
```
CREATE DEFINER='root'@'localhost' PROCEDURE `actualizar_medicamento`(IN id_medicamento VARCHAR(10), in nombre_medicamento
VARCHAR(45))
BEGIN
UPDATE `consultorio_medico_ospina`.`tb_medicamento` SET
`id_medicamento` = '600E6', `nombre_medicamento` = 'Impoio6', `dosis` =
'Dos veces al dia 6' WHERE (`id_medicamento` = '600E');
END
```



Se crea procedimiento para el requerimiento de eliminar un registro de un medicamento a la tabla medicamento.

```
1 • CREATE PROCEDURE eliminar_registro(in producto varchar(20))
2 BEGIN
3     delete from tb_medicamento where id_medicamento = producto;
4 END
5
```





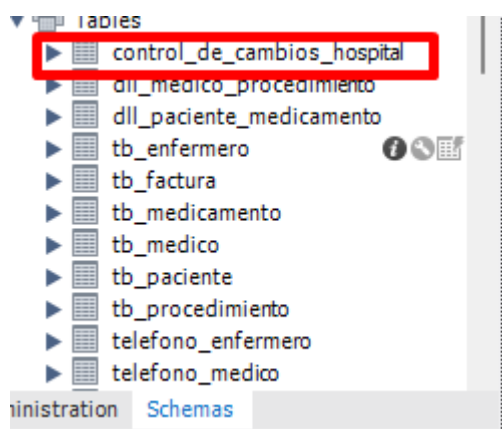
Como se puede observar en la imagen, el ID 600E ya no existe en la tabla.

id_medicamento	nombre_medicamento	dosis
600A	Ibuprofeno	2 al dia
600B	Bipirona	6 cada horas
600C	Acetaminofen	mucho hasta aliviarse
600D	Calmidon	2 veces al dia 5mg
NULL	NULL	NULL

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.

Se crea la tabla control_de_cambios_hospital

```
create table control_de_cambios_hospital(
    Usuario varchar(100), Accion varchar(100), Fecha datetime default current_timestamp
);
```



Se crea el trigger agregar y eliminar.

```
DELIMITER //
CREATE trigger agregar_procedimiento after insert on tb_procedimiento for each row begin
insert into control_de_cambios_hospital values
(user(), "Agregar un nuevo procedimiento", now());
end;
//
DELIMITER ;

DELIMITER //
create trigger eliminar_procedimiento after delete on tb_procedimiento
for each row
begin insert into control_de_cambios_hospital values (user(), "Eliminar un procedimiento", now());
end;

//
DELIMITER ;

insert into tb_procedimiento values ("300W", "Cirugia de ceja");

delete from tb_procedimiento where id_procedimiento="300W";

select * from control_de_cambios_hospital
```

se crea un nuevo registro con el ID 300W.

id_procedimiento	tipo_procedimiento
300C	Triage
300D	Ecografia
300E	Rx
300W	Cirugia de ceja

En esta tabla se muestra el registro que se realizo en la tabla anterior.

Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
Usuario	Accion	Fecha	
root@localhost	Agregar un nuevo procedimiento	2023-02-14 21:48:47	

Se Elimina procedimiento con el ID 300W.

Result Grid		Filter Rows:
id_procedimiento	tipo_procedimiento	
300B	Cirugia	
300C	Triage	
300D	Ecografia	
300E	Rx	
NULL	NULL	

En la siguiente tabla muestra la alerta con la fecha y la acción que fue realizada en la tabla anterior.

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	Usuario	Accion	Fecha			
▶	root@localhost	Agregar un nuevo procedimiento	2023-02-14 21:48:47			
	root@localhost	Eliminar un procedimiento	2023-02-14 21:51:30			

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

Al modelo librería busca libre, le agregaría una tabla llamada registro para saber que libros son los mas vendidos e intervenir de forma más ágil con los libros que se venden poco y de esto que mande un aviso al dueño para que promocióne estos productos.