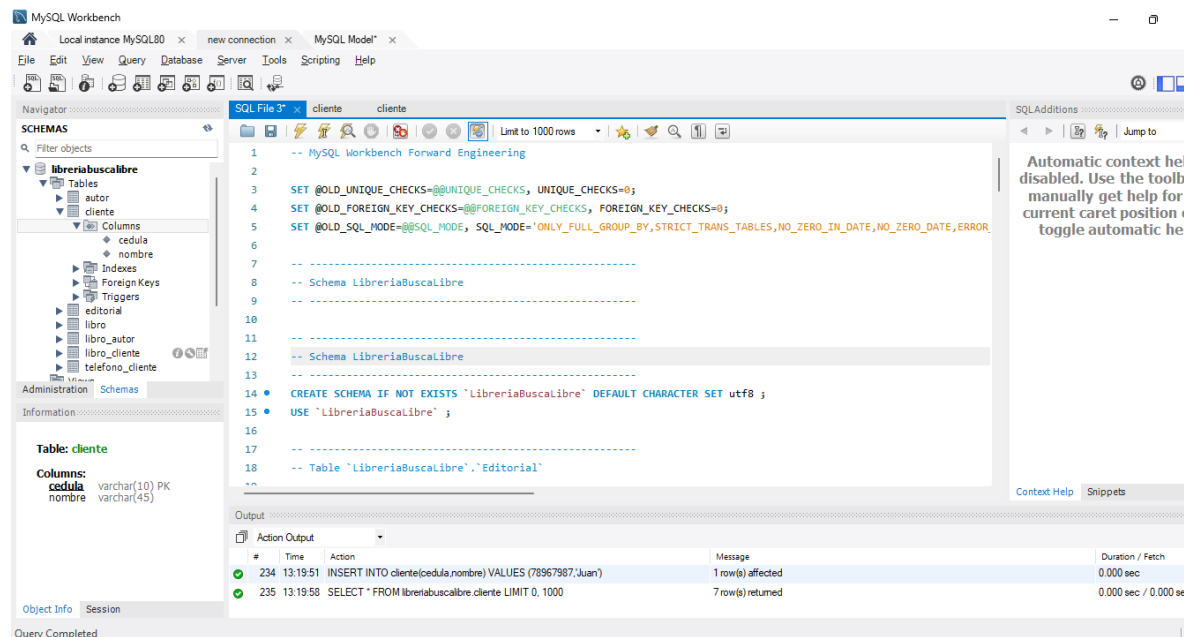


## Taller 5 :

Se inicia el taller ejecutando el script que se adjunta en el README para crear la BD.



Se comienza hacer los registros en las tablas indicadas y queda de esta forma:

### Registro tabla autor:

```
INSERT INTO autor (id,`fecha de nacimiento`,nacionalidad,nombre) VALUES
(1,'25/07/1991','Colombia','Efrain');
INSERT INTO autor (id,`fecha de nacimiento`,nacionalidad,nombre) VALUES
(2,'30/09/1990','España','Andres');
INSERT INTO autor (id,`fecha de nacimiento`,nacionalidad,nombre) VALUES
(4,'1/01/1960','Colombia','Gabriel');
INSERT INTO autor (id,`fecha de nacimiento`,nacionalidad,nombre) VALUES
(5,'2/05/1978','Argentina','Felipe');
INSERT INTO autor (id,`fecha de nacimiento`,nacionalidad,nombre) VALUES
(3,'31/12/1980','Colombia','Mario');
```

### Registro tabla libro:

```
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES
(1,'Satanás',200,'Planeta');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (2,'Cien años
de soledad',1000,'Sudamericana');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (3,'La
hojarasca',40,'Sudamericana');
```

```

INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES
(4,'1984',300,'Debolsillo');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (5,'Crimen y
castigo',500,'Porrúa S.A');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (6,'Melancolia
de los feos',150,'Planeta');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (7,'El pantano
de las mariposas',200,'Destino');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (8,'¿Dónde
está la franja amarilla editorial?',50,'LITERATURA RANDOM HOUSE MONDADORI S.A. ');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (9,'Un mundo
feliz',200,'Planeta');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (10,'Moby
Dick',200,'Editorial Alma');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (11,'el club de
la pelea',200,'Tornamesa');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (12,'CABALLO
DE TROYA',200,'Penguin Libros');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (13,'El código
Da Vinci',200,'Umbriel Editores');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES
(14,'Debolsillo',200,'Debolsillo');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (15,'La milla
verde',200,'Debolsillo');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES
(16,'Dracula',200,'Editorial Porrúa');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES
(17,'Akelarre',200,'Planeta');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (18,'A sangre
fría',200,'Debolsillo');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (19,'Las venas
abiertas de América Latina',200,'Siglo XXI Editores');
INSERT INTO libro (ISBN,titulo,numero_paginas,nombre_editorial) VALUES (20,'El extraño
caso del Dr. Jekyll y Mr. Hyde',200,'Editorial Alma');

```

Registro tabla cliente:

```

INSERT INTO cliente(cedula,nombre) VALUES (43243243,'Andres');
INSERT INTO cliente(cedula,nombre) VALUES (534354,'Jorge');
INSERT INTO cliente(cedula,nombre) VALUES (53453453,'Laura');
INSERT INTO cliente(cedula,nombre) VALUES (31321312,'Leidy');
INSERT INTO cliente(cedula,nombre) VALUES (321312,'Daniela');
INSERT INTO cliente(cedula,nombre) VALUES (1243245,'Jesus');
INSERT INTO cliente(cedula,nombre) VALUES (78967987,'Juan');

```

Registro tabla editorial:

```
INSERT INTO editorial(nombre,ciudad,complemento,Telefono) VALUES
('Debolsillo','España','CL 40 #50', 43267657);
INSERT INTO editorial(nombre,ciudad,complemento,Telefono) VALUES ('Editorial
Alma','Perú','Cr89 # 20 -60', 543543);
INSERT INTO editorial(nombre,ciudad,complemento,Telefono) VALUES
('Destino','Argentina', 'Cr 10 # 30 -50', 654655465);
INSERT INTO editorial(nombre,ciudad,complemento,Telefono) VALUES
('Planeta','Mexico','Cr 50 # 30', 43243324);
```

Registro tabla libro\_cliente:

```
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (1,1243245);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (20,1243245);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (3,1243245);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (6,321312);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (7,321312);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (1,534354);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (7,534354);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (17,321312);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (1,43243243);
INSERT INTO libro_cliente(ISBN_libro_cliente,id_cliente) VALUES (2,43243243);
```

Registro tabla libro\_autor:

```
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (17,3);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (6,3);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (2,1);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (5,1);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (2,4);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (18,4);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (20,4);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (7,5);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (12,2);
INSERT INTO libro_autor(ISBN_libro,id_autor) VALUES (1,3);
```

Registro tabla telefono\_cliente:

```
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (31321312,53543435);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (31321312,54354345);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (321312,6757566765);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (321312,4324324);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (43243243,654867867);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (43243243,97997897);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (534354,767565);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (534354,98798870);
```

```
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (53453453,765765);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (53453453,4537876);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (78967987,3213145);
INSERT INTO telefono_cliente(cedula_cliente,numero) VALUES (78967987,7545343);
```

Consultas SQL en BD:

1. Se realiza la consulta que me permite ver el “nombre” y “fecha\_nacimiento” del escritor:

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

2. Se realiza la consulta que me permite hacer un conteo de los libros vendidos sin que se repitan:

```
SELECT count(distinct ISBN_libro_cliente ) as Libros_vendidos FROM
libreriabuscalibre.libro_cliente;
```

3. Se realizó la consulta que me permite ver el nombre del cliente con su número telefónico:

```
SELECT nombre, numero FROM libreriabuscalibre.cliente join telefono_cliente on
cedula=cedula_cliente ;
```

4. Se realizó la consulta que me permite ver el nombre del libro y el nombre del autor:

```
SELECT titulo,nombre FROM libreriabuscalibre.libro join libro_autor on ISBN_libro =
ISBN join autor on id_autor = id;
```

5. Se realizó la consulta que me permite ver el nombre de las editoriales que vendieron libros:

```
SELECT distinct nombre_editorial FROM libreriabuscalibre.libro join libro_cliente on
ISBN = ISBN_libro_cliente;
```

Vistas:

Se crea una vista para hacer un conteo de los libros más vendidos y los clientes más frecuentes ya que por medio de esta se puede evaluar parte del inventario que se puede tener y hacer un registro de los cliente frecuentes para hacer descuentos, etc.

```
CREATE VIEW libros_mas_vendidos as  
Select titulo,ISBN_libro_cliente from libro join libro_cliente on ISBN = ISBN_libro_cliente
```

Se crea una segunda vista para tener los datos de contacto de los clientes en caso de tener publicidad o mantenerlos informados de todos los cambios en la librería.

```
CREATE VIEW datos_contacto_cliente as  
SELECT nombre, numero FROM libreriabuscalibre.cliente join telefono_cliente on  
cedula=cedula_cliente ;
```

Actividad 2:

Se crea BD clínica sofka con sentencias SQL Y el script queda así:

```
create database clinica_sofka;
```

```
create table enfermero(  
ID_enfermero varchar(15) primary key,  
Nombre_enfermero varchar(100),  
ID_medico_enfermero varchar(15),  
foreign key(ID_medico_enfermero) REFERENCES medico(ID_medico)  
);
```

```
create table telefono_enfermero(  
Telefono varchar(15) primary key,
```

```
ID_enfermero_telefono varchar(100),  
foreign key(ID_enfermero_telefono) REFERENCES enfermero(ID_enfermero)  
);
```

```
create table medico(  
ID_medico varchar(15) primary key,  
Nombre_medico varchar(100),  
Especialidad_medico varchar(100)  
);
```

```
create table telefono_medico(  
Telefono_medico varchar(15) primary key,  
ID_medico_telefono varchar(100),  
foreign key(ID_medico_telefono) REFERENCES medico(ID_medico)  
);
```

```
create table medico_procedimiento(  
ID_procedimiento_medico varchar(15) primary key,  
ID_medico_procedimiento varchar(100),  
foreign key(ID_procedimiento_medico) REFERENCES procedimiento(ID_procedimiento),  
foreign key(ID_medico_procedimiento) REFERENCES medico(ID_medico)  
);
```

```
create table procedimiento(  
ID_procedimiento varchar(15) primary key,  
tipo_procedimiento varchar(100)  
);
```

```
create table paciente(  
ID_paciente varchar(15) primary key,  
Nombre_paciente varchar(100),  
Direccion_paciente varchar(100),  
ID_procedimiento_paciente varchar(15),  
foreign key(ID_procedimiento_paciente) REFERENCES procedimiento(ID_procedimiento)  
);
```

```
create table telefono_paciente(  
Telefono_paciente varchar(15) primary key,  
ID_paciente_telefono varchar(100),  
foreign key(ID_paciente_telefono) REFERENCES paciente(ID_paciente)  
);
```

```
create table medicamento(  
ID_medicamento varchar(15) primary key,  
Nombre_medicamento varchar(100),  
Dosis_medicamento varchar(100)  
);
```

```

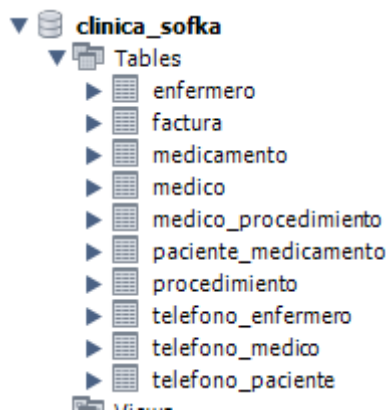
create table paciente_medicamento(
ID_paciente_medicamento varchar(15) primary key,
ID_medicamento_paciente varchar(100),
foreign key(ID_paciente_medicamento) REFERENCES paciente(ID_paciente),
foreign key(ID_medicamento_paciente) REFERENCES medicamento(ID_medicamento)
);

```

```

create table factura(
ID_factura varchar(15) primary key,
Fecha_factura varchar(100),
Valor_factura varchar(100),
ID_paciente_factura varchar(15),
foreign key(ID_paciente_factura) REFERENCES paciente(ID_paciente)
);

```



Se hacen los 5 registros por tabla:

Tabla enfermero:

```

insert into enfermero(ID_enfermero,Nombre_enfermero,ID_medico_enfermero)
values (123,'Gina',432443),
(543543,'Cristina',434238),
(868768,'Hugo',654676),
(865341,'Daniela',654654),
(123789,'Ingrid',654635);

```

Tabla telefono\_enfermero:

```

insert into telefono_enfermero(Telefono,ID_enfermero_telefono)
values (5675,123),
(324567,543543),

```

```
(2545676,868768),  
(678647,865341),  
(56547,123789);
```

Tabla medico:

```
insert into medico(ID_medico,Nombre_medico,Especialidad_medico)  
values (432443, 'Laura', 'Cardiologo'),  
(434238, 'Efrain','oral y maxilofacial'),  
(654676,'Juan','ortopédica y traumatología'),  
(654654, 'Andres','pediátrica'),  
(654635,'Juana','Neurocirugía');
```

Tabla telefono\_medico:

```
insert into telefono_medico(Telefono_medico,ID_medico_telefono)  
values (567567567,1),  
(32456769,432443),  
(254567687,434238),  
(67864736,654654),  
(5654763,654635);
```

Tabla procedimiento:

```
insert into procedimiento(ID_procedimiento,tipo_procedimiento)  
values (7878973127, 'Cardiologia'),  
(654685901,'maxilofacial'),  
(123895430,'ortopédica'),  
(10954756,'pediátrica'),  
(75090978901,'Neurocirugía')
```

Tabla medico\_procedimiento:

```
insert into medico_procedimiento(ID_procedimiento_medico,ID_medico_procedimiento)  
values (7878973127,432443),  
(654685901,434238),  
(123895430,654676),  
(10954756,654654),  
(75090978901,654635);
```

Tabla paciente:

```
insert into  
paciente(ID_paciente,Nombre_paciente,Direccion_paciente,ID_procedimiento_paciente)  
values (1, 'Andrea','cr 55',7878973127),  
(2,'Lucas','cr 60',654685901),  
(3,'Jhon','cr40',123895430),  
(4,'Lisa','cl102',10954756),
```



```
(5,'Ruben','cr30',75090978901);
```

Tabla telefono\_paciente:

```
insert into telefono_paciente(Telefono_paciente,ID_paciente_telefono)
values (567567567,1),
(32456769,2),
(254567687,3),
(67864736,4),
(5654763,5);
```

Tabla paciente\_medicamento:

```
insert into paciente_medicamento(ID_paciente_medicamento,ID_medicamento_paciente)
values (1,60),
(2,70),
(3,80),
(4,90),
(5,100);
```

Tabla medicamento:

```
insert into medicamento(ID_medicamento,Nombre_medicamento,Dosis_medicamento)
values (60, 'Acetaminofén','40 tabletas'),
(70,'Dolex','150 tabletas'),
(80,'Pregabalina','100 tabletas'),
(90,'Celecoxid','50 tabletas'),
(100,'Adorén','10 tabletas');
```

Tabla factura:

```
insert into factura(ID_factura,Fecha_factura,Valor_factura,ID_paciente_factura)
values (606, '01/06/2022',3000000,1),
(707,'02/07/2022',1200000,2),
(808,'20/10/2022',90000,3),
(909,'05/12/2022',5000000,4),
(1001,'30/01/2022',1000000,5);
```

Consultas actividad 2:

Se realiza la consulta donde se ve el medicamento, dosis y el paciente que la necesita:

```
SELECT Nombre_medicamento,Dosis_medicamento, Nombre_paciente FROM  
clinica_sofka.medicamento join paciente_medicamento on ID_medicamento_paciente =  
ID_medicamento join paciente on ID_paciente = ID_paciente_medicamento ;
```

Se realiza la consulta donde se ve el nombre de paciente y el nombre del enfermero que lo acompañó en el procedimiento:

```
SELECT Nombre_paciente,Nombre_enfermero FROM clinica_sofka.paciente  
join procedimiento on ID_procedimiento=ID_procedimiento_paciente  
join medico_procedimiento on ID_procedimiento_medico = ID_procedimiento  
join enfermero on ID_medico_enfermero = ID_medico_procedimiento;
```

Vistas Actividad 2:

Vista 1:

Se realiza vista para evidenciar el paciente, personal que participó en el procedimiento(Médico y enfermeros) y nombre del procedimiento. Esta me parece importante ya que contiene toda la información del procedimiento con sus participantes. Se puede llevar un control de asistencia en la clínica.

```
create view info_procedimiento as  
SELECT Nombre_paciente,Nombre_enfermero,tipo_procedimiento, Nombre_medico FROM  
clinica_sofka.paciente  
join procedimiento on ID_procedimiento=ID_procedimiento_paciente  
join medico_procedimiento on ID_procedimiento_medico = ID_procedimiento  
join enfermero on ID_medico_enfermero = ID_medico_procedimiento  
join medico on ID_medico = ID_medico_procedimiento;
```

Vista 2:

Se realiza vista para evidenciar la factura del paciente donde le muestra el tipo de procedimiento, valor y fecha en que se realizó el procedimiento. Esta vista me parece importante ya que le da toda la información relevante muy detallada para el pago de su factura.

```
create view Costo_factura as
select Valor_factura, Fecha_factura, Nombre_paciente, tipo_procedimiento from factura
join paciente on ID_paciente = ID_paciente_factura
join procedimiento on ID_procedimiento = ID_procedimiento_paciente;
```

Vista 3:

En esta vista muestra el valor total sumando el costo de todas la facturas.  
Es importante ya que se lleva un control de los ingresos de la clínica sofka por los  
procedimiento:

```
create view Total_facturas as
select sum(Valor_factura) as Total from factura;
```

Pregunta final:

¿Qué le agregaría al modelo para dar más información y esa información cuál sería?

Para dar más información le agregaría el salario del enfermero, salario del médico, costo del medicamento para ser más claros con ganancia neta de la clínica.

También le agregaría un número de sala asignado al procedimiento y con el costo por el alquiler de esta sala para llevar un control de la infraestructura del hospital.

Taller 6 actividad 3:

Se elabora procedimiento para agregar registros a la tabla autor y queda de esta forma:

use libreria buscalibre;

```
DELIMITER //
create procedure agregar_registro_autor(in ID_autor varchar(15), in fecha_nacimiento
varchar(50), in nacionalidad_autor varchar(50), in nombre_autor varchar(50))
begin
select * from autor;
insert into autor (id, `fecha de nacimiento`, nacionalidad, nombre) values
(ID_autor, fecha_nacimiento, nacionalidad_autor, nombre_autor);
end//
delimiter ;
```

```
call agregar_registro_autor(12345,'12/03/2000','colombia','Efrain');
```

Se elabora procedimiento para eliminar un registro de la tabla autor y queda de esta forma:

```
DELIMITER //
create procedure eliminar_registro_autor(in ID_autor varchar(15))
begin
select * from autor;
delete from autor where id = ID_autor;
end//
delimiter ;

call eliminar_registro_autor(12345);
```

Se elabora procedimiento para actualizar un registro de la tabla autor y queda de esta forma:

```
DELIMITER //
create procedure actualizar_registro_autor(in ID_autor varchar(15), in fecha_nacimiento
varchar(50),in nacionalidad_autor varchar(50),in nombre_autor varchar(50))
begin
select * from autor;
update autor set `fecha de nacimiento` = fecha_nacimiento, nacionalidad =
nacionalidad_autor, nombre= nombre_autor where id = ID_autor;
end//
delimiter ;

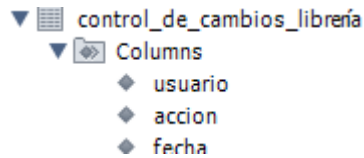
call actualizar_registro_autor(1,'20/12/1990', 'perù', 'Raul');
```

Se elabora procedimiento para consultar un registro de la tabla autor y queda de esta forma:

```
DELIMITER //
create procedure consultar_registro_autor(in ID_autor varchar(15))
begin
select * from autor where id = ID_autor;
end//
delimiter ;
```

```
call consultar_registro_autor(1);
```

Se crea la tabla “control\_de\_cambios\_librería” con sus tres atributos:



```
create table control_de_cambios_librería(  
usuario varchar(50) primary key,  
accion varchar(50),  
fecha datetime default current_timestamp  
);
```

También se crea el trigger que me da un registro del usuario que eliminó un registro de la tabla autor:

```
delimiter //  
create trigger registro_eliminado_autor after delete on autor  
for each row  
begin  
    insert into control_de_cambios_librería values (user,'se elimino registro registro',now());  
end//  
delimiter ;
```

Se crea trigger para usuarios que hagan un registro en la tabla autor:

```
delimiter //  
create trigger registro_autor after insert on autor  
for each row  
begin  
    insert into control_de_cambios_librería values (user,'se hizo registro',now());  
end//  
delimiter ;
```

Los registro en la tabla “control\_de\_cambios\_librería” se ven así:

|   | usuario        | accion           | fecha               |
|---|----------------|------------------|---------------------|
| ▶ | Efrain         | se hizo registro | 2023-02-14 19:36:53 |
|   | root@localhost | se hizo registro | 2023-02-14 19:41:36 |
| * | NULL           | NULL             | NULL                |

Se crea procedimiento para agregar registros a la tabla procedimiento de la clínica sofka

```
delimiter //
create procedure agregar_registro_procedimiento(in ID_procedimiento_procedure varchar(15), in tipo_procedimiento_procedure
begin
select * from procedimiento;
insert into procedimiento (ID_procedimiento,tipo_procedimiento) values (ID_procedimiento_procedure,tipo_procedimiento_proce
end//
delimiter ;
```

Se crea procedimiento para actualizar registros en la tabla procedimiento:

```
delimiter //
create procedure actualizar_registro_procedimiento(in ID_procedimiento_procedure varchar(15), in tipo_procedimiento_procedure varchar(50))
begin
select * from procedimiento;
update procedimiento set tipo_procedimiento = tipo_procedimiento_procedure where ID_procedimiento = ID_procedimiento_procedure;
end//
delimiter ;
```

Se crea procedimiento para eliminar registro de la tabla procedimiento:

```
delimiter //
create procedure eliminar_registro_procedimiento(in ID_procedimiento_procedure varchar(15))
begin
select * from procedimiento;
delete from procedimiento where ID_procedimiento = ID_procedimiento_procedure;
end//
delimiter ;
```

Se crea procedimiento para consultar registros en la tabla procedimiento:

```
delimiter //
create procedure consultar_registro_procedimiento(in ID_procedimiento_procedure varchar(15))
begin
select * from procedimiento where ID_procedimiento = ID_procedimiento_procedure;
end//
delimiter ;
```

Se crea trigger para hacer el registro cuando un usuario agregue un registro de la tabla procedimiento:

```
delimiter //
create trigger registro_procedimiento after insert on procedimiento
for each row
begin
insert into control_de_cambios_hospital values (user(),'se hizo registro',now());
end//
delimiter ;
```

Se crea trigger para hacer el registro cuando un usuario elimine un registro de la tabla procedimiento:

```
delimiter //
create trigger registro_procedimiento_eliminado after delete on procedimiento
for each row
begin
insert into control_de_cambios_hospital values (user(),'se hizo registro',now());
end//
delimiter ;
```

Los registros de los usuarios cuando eliminan o agregan un registro se ve de esta manera en la tabla control\_de\_cambios\_hospital:

|   | Usuario        | Accion           | Fecha               |
|---|----------------|------------------|---------------------|
| ▶ | root@localhost | se hizo registro | 2023-02-14 21:35:09 |
|   | root@localhost | se hizo registro | 2023-02-14 21:49:14 |