

Taller 6 Bases de Datos

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

*Procedimientos Libreria

-Agregar Libro: Se crea este procedimiento para agregar un libro nuevo pasándole los campos necesarios y teniendo en cuenta que el nombre_editorial debe existir en la base de datos

```
DELIMITER //
create procedure agregar_libro(in isbn_libro varchar(10),in titulo_libro varchar(45),n_paginas varchar(45),name_editorial varchar(50))
BEGIN
insert into libro values (isbn_libro,titulo_libro,n_paginas,name_editorial);
END ;
//
call agregar_libro('L21', 'Probando ando', '500', 'Alba');
```

Luego se puede ver que se agrega el registro en la tabla de Libros

	ISBN	titulo	numero_paginas	nombre_editorial
	L2	Design Patterns	1200	Planeta
	L20	El fenix	417	Alba
	L21	Probando ando	500	Alba
	L3	Algoritms Introduct...	320	Alfaguara
	L4	La Juria	250	Alfaguara

-Actualizar Libro: En este procedimiento se le entregan todos los datos que se entregan en el procedimiento de actualizar pero, en este se actualizan los datos dependiendo del id_libro que se entregue como parámetro. El ISBN sigue siendo el mismo pero nos permite cambiar las otras columnas.

```
DELIMITER //
create procedure actualizar_libro(in isbn_libro varchar(10),in titulo_libro varchar(45),n_paginas varchar(45),name_editorial varchar(50))
BEGIN
UPDATE libro
SET titulo = titulo_libro, numero_paginas = n_paginas, nombre_editorial = name_editorial
WHERE ISBN = isbn_libro;
END ;
//
call actualizar_libro('L21', 'Introduccion a Procedimientos', '280','Alfaguara');
```

Al listar los libros podemos evidenciar que el libro con ISBN “L21” se actualiza con los nuevos datos

	ISBN	titulo	numero_paginas	nombre_editorial
	L20	El fenix	417	Alba
	L21	Introduccion a Proced...	280	Alfaguara
	L3	Algoritms Introduction	320	Alfaguara

-Consultar Libros: Para este procedimiento decidí que se le entregué como parámetro el nombre de la editorial para filtrar los libros que pertenecen a determinada editorial.

```
DELIMITER //
create procedure listar_libros(in name_editorial varchar(50))
BEGIN
select * from libro where nombre_editorial = name_editorial;
END ;
//
call listar_libros('Alba');
```

Ahora al llamar el procedimiento muestra los libros pertenecientes a la editorial Alba.

	ISBN	titulo	numero_paginas	nombre_editorial
▶	L1	Clean code	800	Alba
	L12	Codigo da vinci	580	Alba
	L15	Codigo enigma	310	Alba
	L20	El fenix	417	Alba
	L6	El señor de los anillos	650	Alba

-Eliminar Libro: Por ultimo se crea un procedimiento para eliminar un libro y, recibe como parámetro el ISBN del libro que desea eliminar.

```
DELIMITER //
create procedure eliminar_libro(in isbn_libro varchar(10))
BEGIN
delete from libro where ISBN = isbn_libro;
END ;
//
call eliminar_libro('L21');
```

Ahora al listar los libros nos damos cuenta que el registro con ISBN = L21 ya no existe en los registros.

	ISBN	titulo	numero_paginas	nombre_editorial
▶	L1	Clean code	800	Alba
	L10	Algebra de Baldor	840	Planeta
	L11	Divina comedia	260	Alfaguara
	L12	Codigo da vinci	580	Alba
	L13	Angeles y demonios	640	Planeta
	L14	Free soul	280	Planeta
	L15	Codigo enigma	310	Alba
	L16	Logica	745	Planeta
	L17	Hogwarts Trilogy	850	Planeta
	L18	Las musas	214	Alfaguara
	L19	Renacer	360	Libretextos
	L2	Design Patterns	1200	Planeta
	L20	El fenix	417	Alba
	L3	Algoritms Introduction	320	Alfaguara
	L4	La Juria	250	Alfaguara
	L5	Harry Potter	95	Planeta
	L6	El señor de los anillos	650	Alba
	L7	Como hacerse rico	1200	Planeta
	L8	Aprende a programar...	210	Alfaguara
	L9	Maria	1200	Planeta
*	NULL	NULL	NULL	NULL

*Triggers Libreria

-Primero creamos la tabla que se especifica en el taller

```
create table control_de_cambios_libreria(
  usuario varchar(45),
  accion varchar(45),
  fecha datetime default current_timestamp
);
```

Ahora se crea el trigger que se encargara de crear un registro en la tabla control de cambios cada que un usuario de la base de datos agregue un nuevo libro.

```
DELIMITER //
create trigger inserto_libro after insert on libro
for each row
begin
  insert into control_de_cambios_libreria
  values (user(),"AGREGO LIBRO",now());
end;
//
DELIMITER ;
```

Luego se crea un nuevo registro para ver que el trigger funcione.

```
insert into libro values ('L22', 'Triggers', '900', 'Alba');
```

Por último vemos lo que se crea en la tabla de control de cambios.

	usuario	accion	fecha
▶	root@localhost	AGREGO LIBRO	2023-02-14 18:51:36

-Ahora se crea el trigger que dejara constancia cuando un usuario de la base de datos elimine un registro de la tabla libros.

```
DELIMITER //
create trigger elimino_libro after delete on libro
for each row
begin
    insert into control_de_cambios_libreria
    values (user(),"ELIMINO LIBRO",now());
end;
//
DELIMITER ;
```

Luego se elimina un registro para ver que se registra en la tabla control de cambios.

```
delete from libro where ISBN = 'L22';
```

Por último vemos que se agrega el nuevo registro a la tabla de control de cambios.

	usuario	accion	fecha
▶	root@localhost	AGREGO LIBRO	2023-02-14 18:51:36
	root@localhost	ELIMINO LIBRO	2023-02-14 18:59:43

*Procedimientos Hospital

-**Agregar medico:** Se crea un procedimiento que nos permite agregar nuevos médicos a nuestra base de datos.

```
#Procedimiento para agregar un nuevo medico
DELIMITER //
create procedure agregar_medico(in id varchar(10),in nombre varchar(45),apellido varchar(45),espe varchar(45))
BEGIN
    insert into medico values (id,nombre,apellido,espe);
END ;
//
call agregar_medico('M6', 'Juan', 'Valdez','Cirujia Plastica');
```

Luego de llamarlo podemos ver que se ejecuta correctamente³ y el registro se agrega.

M5	Grace	Hurts	Pediatría
M6	Juan	Valdez	Cirujia Plastica

-Actualizar medico: Ahora teniendo en cuenta el id de medico que se ingrese se actualizaran los cambios que se deseen con excepción del id de médico.

```
#Procedimiento para actualizar un medico
DELIMITER //
create procedure actualizar_medico(in id varchar(10),in nombre varchar(45),apellido varchar(45),espe varchar(45))
BEGIN
UPDATE medico
SET nombre_medico = nombre, apellido_medico = apellido, especialidad = espe
WHERE id_medico = id;
END ;
//
call actualizar_medico('M6', 'Maria', 'Valdez', 'Pediatria');
```

Podemos ver como se actualiza el registro para el medico M6.

M5	Grace	Hurts	Pediatrica
M6	Maria	Valdez	Pediatra

-Consultar medico: Ahora creo un procedimiento para consultar los medios de una especialidad en específico entregándole la especialidad como un parámetro.

```
#Procedimiento para listar los medicos de determinada especialidad
DELIMITER //
create procedure listar_medicos(in espe varchar(45))
BEGIN
select * from medico where especialidad = espe;
END ;
//
call listar_medicos('Pediatria');
```

Ahora bien al llamar el procedimiento nos trae el medico M6 que habíamos actualizado anteriormente.

id_medico	nombre_medico	apellido_medico	especialidad
M6	Maria	Valdez	Pediatra

-Eliminar Medico: Ahora se eliminará un médico teniendo en cuenta el id que se ingrese como parámetro.

```
#Procedimiento para eliminar un medico
DELIMITER //
create procedure eliminar_medico(in id varchar(10))
BEGIN
delete from medico where id_medico = id;
END ;
//
call eliminar_medico('M6');
```

Al llamarlo de borra el registro M6 de nuestra tabla de médicos.

	id_medico	nombre_medico	apellido_medico	especialidad
▶	M1	Andrew	Meneses	Traumatismo
	M2	Felipe	Correa	Cirujía
	M3	Marilyn	Velez	Odontologia
	M4	Carlos	Solarte	Oftalmologia
	M5	Grace	Hurts	Pediatrica

*Triggers Hospital:

Primero se crea la tabla para el control de cambios en el hospital con los atributos definidos.

```
create table control_de_cambios_hospital(  
  usuario varchar(45),  
  accion varchar(45),  
  fecha datetime default current_timestamp  
);
```

Ahora se crea el trigger que registrara los datos del usuario que agregue un nuevo medico

```
#Trigger para registrar quien agrega un medico  
DELIMITER //  
create trigger inserto_medico after insert on medico  
for each row  
begin  
  insert into control_de_cambios_hospital  
  values (user(),"AGREGO MEDICO",now());  
end;  
//  
DELIMITER ;
```

Posteriormente se ingresa un nuevo registro a medico para probar el trigger.

```
insert into medico values('M7', 'Juan', 'Valdez','Cirujia Plastica');
```

Vemos que se crea el registro deseado en la tabla de control de cambios

	usuario	accion	fecha
▶	root@localhost	AGREGO MEDICO	2023-02-14 19:30:27

Ahora se crea el trigger que controlara que usuario de la base de datos ha borrado algún registro de la tabla médico.

```
#Trigger para registrar quien elimina un medico
DELIMITER //
create trigger elimino_medico after delete on medico
for each row
begin
    insert into control_de_cambios_hospital
    values (user(),"ELIMINO MEDICO",now());
end;
//
DELIMITER ;
```

Luego se elimina un registro de medico para probar el trigger.

```
delete from medico where id_medico = 'M7';
```

Por ultimo vemos que se agrega el nuevo registro en nuestra tabla de control de cambios.

	usuario	accion	fecha
▶	root@localhost	AGREGO MEDICO	2023-02-14 19:30:27
	root@localhost	ELIMINO MEDICO	2023-02-14 19:35:10