

## TEMA 7 – ACTIVIDADES EVALUABLE

Te he puesto subrayado en amarillo y con comentario los pequeños errores. La nota de la actividad es:

Rellena cada una de las 11 partes de la actividad con la instrucción o instrucciones correctas. Adjunta pantallazos con las actualizaciones y la base de datos como ha quedado al finalizar el ejercicio.

A partir de la base de datos adjuntada para la actividad evaluable:

1. Crea una tabla llamada GESTIÓN que contenga los siguientes campos:

TABLA GESTIÓN	
Campo	Tipos datos
Cod_Registro	Int (4) / Auto_increment / Primary key
Cod_Tratamiento	Int(4)
Descripcion_Ant	Varchar (25)
Descripcion_Nuevo	Varchar (25)
Fecha_Registro	Datetime
Precio_Ant	Decimal (5,2)
Precio_Nuev	Decimal (5,2)
Cantidad_Ant	Int(11)
Cantidad Nueva	Int(11)
Acción	Varchar (20)
Usuario	Varchar (20)

```
CREATE TABLE Gestion (Cod_Registro Int(4) AUTO_INCREMENT PRIMARY KEY, Cod_Tratamiento Int(4),Descripcion_Ant Varchar(25),Descripcion_Nuevo Varchar(25),Fecha_Registro Datetime,Precio_Ant Decimal(5,2),Precio_Nuev Decimal(5,2),Cantidad_Ant Int(11),Cantidad_Nueva Int(11),Acción Varchar(20),Usuario Varchar(20));
```

2. Cada vez que se inserte un nuevo tratamiento en la tabla TRATAMIENTOS, se tendrá que insertar de manera automática en la tabla GESTION, una fila indicando el código del tratamiento, la descripción del tratamiento, la fecha de registro de la acción, el precio del tratamiento, la cantidad que hay de tratamientos, la acción que se ha realizado (en este caso 'NUEVO') y el usuario que ha registrado la consulta.

```
CREATE TRIGGER Trat_Insertar AFTER INSERT ON tratamientos FOR EACH ROW
```

```
INSERT INTO gestion (cod_tratamiento, descripcion_nuevo, fecha_registro,
precio_nuev, cantidad_nueva, acción, usuario)
VALUES (new.cod_tratamiento, new.descripcion, now(),new.precio, new.cantidad,
'NUEVO', CURRENT_USER())
```

3. Cada vez que se modifique un tratamiento en la tabla TRATAMIENTOS, se tendrá que insertar de manera automática en la tabla GESTION, una fila indicando el código de la tratamiento, la descripción del tratamiento anterior y la nueva, la fecha de registro de la acción, el precio del tratamiento anterior y el nuevo, la cantidad de tratamientos anterior y nueva, la acción que se ha realizado (en este caso 'MODIFICACIÓN') y el usuario que ha registrado la consulta.

```
CREATE TRIGGER Trat_Modificar before UPDATE ON tratamientos
FOR EACH ROW
INSERT INTO gestion (cod_tratamiento, Descripcion_Ant, descripcion_nuevo,
fecha_registro, Precio_Ant, precio_nuev, Cantidad_Ant, cantidad_nueva, acción,
usuario)
VALUES (new.cod_tratamiento, old.descripcion, new.descripcion, now(),old.precio,
new.precio, old.cantidad, new.cantidad, 'MODIFICAR', CURRENT_USER())
```

4. Cada vez que se borre un tratamiento en la tabla TRATAMIENTOS, se tendrá que insertar de manera automática en la tabla GESTION, una fila indicando el código de la tratamiento, la descripción del tratamiento, la fecha de registro de la acción, el precio del tratamiento, la cantidad que hay de tratamientos, la acción que se ha realizado (en este caso 'ELIMINACIÓN') y el usuario que ha registrado la consulta.

```
CREATE TRIGGER Trat_Eliminar before DELETE ON tratamientos
FOR EACH ROW
INSERT INTO gestion (cod_tratamiento, Descripcion_Ant, fecha_registro, Precio_Ant,
Cantidad_Ant, acción, usuario)
VALUES (OLD.cod_tratamiento, old.descripcion, now(),old.precio, old.cantidad,
'ELIMINAR', CURRENT_USER())
```

5. Cada vez que se añade una consulta de algún tratamiento, se tiene que actualizar la cantidad de tratamientos que hay. (Cada consulta es un tratamiento menos).

```
CREATE TRIGGER INSERTAR_CONSULTAS AFTER INSERT ON CONSULTAS
FOR EACH ROW
UPDATE TRATAMIENTOS SET CANTIDAD = CANTIDAD - 1 WHERE COD_TRATAMIE
NTO = NEW.COD_TRATAMIENTO
```

6. Crea un procedimiento almacenado para consultar los clientes que tienen menos de 30 años.

```
CREATE PROCEDURE CLIENTES_30() SELECT * FROM CLIENTES WHERE
EDAD < 30
```

7. Crea un procedimiento almacenado de las mascotas que han realizado alguna consulta antes de enero de 2021 (no incluido)

```
CREATE PROCEDURE MASCOTAS2020()
SELECT * FROM MASCOTAS INNER JOIN CONSULTAS ON
CONSULTAS.COD_MASCOTA=MASCOTAS.CODIGO WHERE FECHA < '2021-
01-01'
```

```
CREATE PROCEDURE MASCOTAS2020()
SELECT * FROM mascotas WHERE Codigo IN (SELECT Cod_Mascota FROM
consultas WHERE Fecha < '2021-01-01')
```

8. Crea un procedimiento almacenado para consultar los clientes que tienen mascotas de tipo 'perro'.

```
CREATE PROCEDURE CLIENTESPERRO() SELECT CLIENTES.* FROM
CLIENTES INNER JOIN MASCOTAS ON
CLIENTES.DNI=MASCOTAS.DNI_CLIENTE WHERE RAZA='PERRO'
```

```
CREATE PROCEDURE CLIENTESPERRO()
SELECT * FROM clientes WHERE DNI IN (SELECT Dni_Cliente FROM
mascotas WHERE Raza = 'Perro' AND Raza = 'perro')
```

9. Crea un procedimiento almacenado que indicando el código del tratamiento y el precio nuevo, se actualice el precio del tratamiento.

```
CREATE PROCEDURE ACT_PRECIO(PRECIO_NUE DECIMAL(5,2), CODIGO
INT(11))
UPDATE TRATAMIENTOS SET PRECIO=PRECIO_NUE WHERE
CODIGO=COD_TRATAMIENTO
```

10. Crea un procedimiento almacenado para calcular el precio con descuento del 10% del tratamiento que indique el usuario. Hazlo definiendo variables (descuento=0.10). Se debe mostrar el precio antiguo y el nuevo.

```
DELIMITER //
CREATE PROCEDURE PRECIO_DESCUENTO(CODIGO INT)
BEGIN
    DECLARE PRECIO_NUEVO DECIMAL(5,2);
    DECLARE DESCUENTO DECIMAL(5,2) DEFAULT 0.10;
    SELECT PRECIO INTO PRECIO_NUEVO FROM TRATAMIENTOS
    WHERE TRATAMIENTOS.COD_TRATAMIENTO= CODIGO;
    SELECT PRECIO_NUEVO AS PRECIO_ANTERIOR;
    SET PRECIO_NUEVO=PRECIO_NUEVO*(1-DESCUENTO);
    SELECT PRECIO_NUEVO;
END;
```

*DELIMITER ;*

11. Crea un Trigger para que cada vez que se modifique el precio de los tratamientos, si se quiere poner un número menor de 20 que te ponga un 20 en el precio, si el precio es mayor que 500 que ponga un 500 y si está entre 20 y 500 que ponga el precio que se indica el usuario.

*DELIMITER //*

*CREATE TRIGGER Modificar\_Precio BEFORE UPDATE ON Tratamientos FOR EACH ROW*

*BEGIN*

*IF(New.precio<20) THEN*

*SET New.precio=20;*

*ELSEIF(New.precio>500) THEN*

*SET New.precio=500;*

*END IF;*

*END; //*

*DELIMITER ;*