
	<p>TEMA 9.-PROGRAMACIÓN MySQL- BD LIBRERÍA</p> <p>MÓDULO DAM-DAW</p> <p>HOJA DE EJERCICIOS 3- BD LIBRERIA.</p> <p>MANEJO DE ERRORES-CURSORES-TRIGGERS</p>
	

2. Crea un procedimiento con un CURSOR para actualizar los precios según el valor del campo GENERO LITERARIO, de tal forma que:
 - si género literario es NOVELA, el nuevo precio será 20;
 - si género literario es JUVENIL, el nuevo precio será 10,
 - si género literario es POESÍA y el precio es mayor que 10, aumentar el precio un 25%.

OS INCLUYO LA PRIMERA PARTE DEL CÓDIGO, PARA QUE OS AYUDE A COMENZAR EL EJERCICIO:

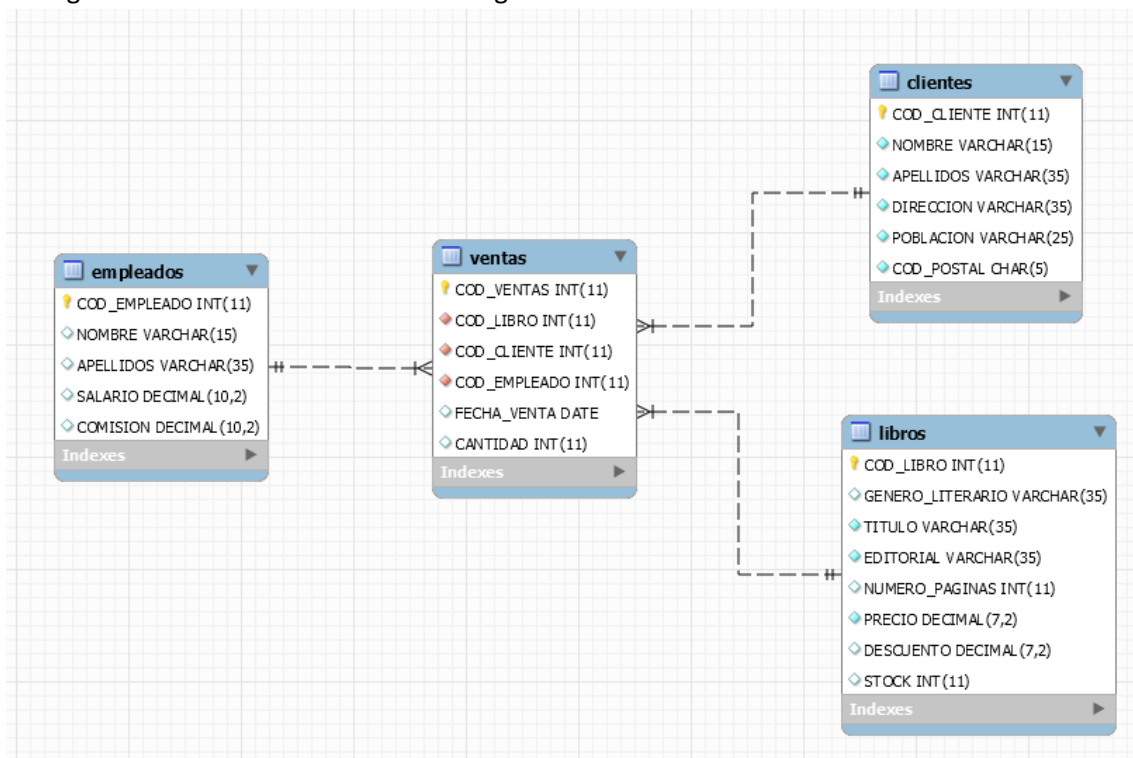
```

DELIMITER //
DROP PROCEDURE IF EXISTS HOJA3_EJER2_SP//
CREATE PROCEDURE HOJA3_EJER2_SP()
BEGIN
    /* DECLARAR VARIABLES LOCALES*/
    DECLARE A INT;
    DECLARE B VARCHAR (35);
    DECLARE C DECIMAL (7,2);
    DECLARE ULTIMAFILA BOOLEAN DEFAULT FALSE; /*VARIABLE QUE CONTROLA FIN DE TABLA*/
    DECLARE C CURSOR FOR SELECT COD_LIBRO, GENERO_LITERARIO, PRECIO FROM LIBROS;
    /*CURSOR PARA LEER DATOS*/
    DECLARE CONTINUE HANDLER FOR SQLSTATE'02000'SET ULTIMAFILA=TRUE;
    OPEN C; /*SE ABRE EL CURSOR*/
    REPEAT /* BUCLE DE LECTURA DEL CURSOR*/
        FETCH C INTO A,B,C; /* GUARDAR CADA DATO DE CADA FILA EN LAS VARIABLES LOCALES*/
        IF NOT ULTIMAFILA THEN
            IF B='NOVELA'THEN UPDATE LIBROS SET PRECIO='20' WHERE COD_LIBRO=A;
        ELSE

```

- Modifica la tabla LIBROS y añade un nuevo campo entero STOCK, que almacenará el número de ejemplares de un determinado libro que está disponible en la librería. Inicialmente, tendremos un stock de 10 ejemplares para cada libro (modifica los registros de la tabla libros con ese dato). Modifica la tabla VENTAS y añade un campo CANTIDAD, que almacenará el número de ejemplares que se venden en cada venta. Este campo inicialmente estará a 0.

El diagrama de la BD modificada será el siguiente:



Después de hacer estas modificaciones, crear un **trigger** que actualice el campo stock cada vez que se realice una venta.

Ejemplos de ejecución de insert en ventas y “disparo” del trigger:



Tengo este registro en la tabla libros:

COD_LIBRO	GENERO_LITERARIO	TITULO	EDITORIAL	NUMERO_PAGINAS	PRECIO	DESCUENTO	STOCK
1	JUVENIL	AY DEL QUE SE RIA	ANAYA	125	10.00	NULO	10

Voy a realizar una venta de 2 unidades de este libro (cantidad=2)

```
5 insert into ventas VALUES (50,1,1,1,'2020/04/20',2);
```

Compruebo que se ha actualizado el stock en la tabla libros:

	<p>TEMA 9.-PROGRAMACIÓN MySQL- BD LIBRERÍA</p> <p>MÓDULO DAM-DAW</p> <p>HOJA DE EJERCICIOS 3- BD LIBRERIA.</p> <p>MANEJO DE ERRORES-CURSORES-TRIGGERS</p>
	

115 `SELECT * FROM LIBROS;`

	COD_LIBRO	GENERO_LITERARIO	TITULO	EDITORIAL	NUMERO_PAGINAS	PRECIO	DESCUENTO	STOCK
1		JUVENIL	AY DEL QUE SE RIA	ANAYA	125	10.00	NULL	8

4. Crear un **trigger** que cuando se borre un registro en la tabla VENTAS, actualice el stock en la tabla libros, simulando una devolución.

Ejemplos de ejecución de delete en ventas y “disparo” del trigger:

Tengo este registro en la tabla libros:

3	TEATRO	LA DAMA DUENDE	PLANETA	150	14.00	NULL	10
---	--------	----------------	---------	-----	-------	------	----

En primer lugar voy a realizar una venta de 5 unidades de este libro: (se disparará el trigger del ejercicio anterior)

134 `INSERT INTO VENTAS VALUES (25,3,1,1,'2019-02-11',5)`
 135 `select * from libros;`



	COD_LIBRO	GENERO_LITERARIO	TITULO	EDITORIAL	NUMERO_PAGINAS	PRECIO	DESCUENTO	STOCK
1		JUVENIL	AY DEL QUE SE RIA	ANAYA	125	10.00	NULL	8
2		JUVENIL	TODO CONTROLADO	ANAYA	200	10.00	NULL	10
3		TEATRO	LA DAMA DUENDE	PLANETA	150	14.00	NULL	5

Ahora voy a borrar en ventas la fila 25:

En la tabla LIBROS se debe de actualizar el stock.

136 `DELETE FROM VENTAS WHERE COD_VENTAS=25`
 137 `select * from libros;`

	COD_LIBRO	GENERO_LITERARIO	TITULO	EDITORIAL	NUMERO_PAGINAS	PRECIO	DESCUENTO	STOCK
1		JUVENIL	AY DEL QUE SE RIA	ANAYA	125	10.00	NULL	8
2		JUVENIL	TODO CONTROLADO	ANAYA	200	10.00	NULL	10
3		TEATRO	LA DAMA DUENDE	PLANETA	150	14.00	NULL	10

	<p>TEMA 9.-PROGRAMACIÓN MySQL- BD LIBRERÍA</p> <p>MÓDULO DAM-DAW</p> <p>HOJA DE EJERCICIOS 3- BD LIBRERÍA.</p> <p>MANEJO DE ERRORES-CURSORES-TRIGGERS</p>
	

5. Crear un trigger de auditoría que almacene en una tabla auxiliar que llamaremos “NUEVOS_CLIENTES”, los siguientes datos de los nuevos clientes que se den de alta en nuestra BD. La tabla NUEVOS_CLIENTES tendrá la siguiente estructura:

Field	Type	Null	Key	Default	Extra
NUM_REGISTRO	int(11)	NO	PRI	NULL	auto_increment
REGISTRO	varchar(200)	YES		NULL	

Sólo tiene dos campos, el primero es la PK y es un entero automático (se rellena automáticamente al insertar cada fila). El campo REGISTRO guarda los datos de la modificación concatenando: el usuario que da de alta el nuevo cliente (USER()), la fecha de alta (CURRENT_DATE()), el nombre, apellido y población del usuario.

Ejemplo de inserción de un nuevo cliente y “disparo” del trigger:

Insertamos un nuevo cliente en la tabla, por ejemplo:

```
INSERT INTO CLIENTES VALUES (20, 'MARTIN', 'MARTINEZ', 'CALLE NUEVA', 'PLASENCIA', '10600')
```

Este es el contenido en la tabla CLIENTES:

COD_CLIENTE	NOMBRE	APELLIDOS	DIRECCION	POBLACION	CODIGO_POSTAL
1	ANA	GARCIA	c SOL	PLASENCIA	10600
2	LUIS	SANCHEZ	c PALMERAS	PLASENCIA	10600
3	ANTONIO	GARCIA	c LUNA	PLASENCIA	10600
4	MARIA	SÁNCHEZ	c MAYOR	PLASENCIA	10600
20	MARTIN	MARTINEZ	CALLE NUEVA	PLASENCIA	10600

Si el trigger funciona, se debe haber insertado una fila en la tabla NUEVOS_CLIENTES:

NUM_REGISTRO	REGISTRO
1	ALTA REALIZADA POR: root@localhost FECHA: 2020-05-06 NUEVO CLIENTE MARTIN MARTINEZ PLASENCIA