Crea una base de datos llamada “TU NOMBRE”. (0.20 PUNTOS)

**CREATE DATABASE alberto\_bolta\_sanmateu**

Partiendo del siguiente diagrama, crea las siguientes tablas con los atributos correspondientes. (0.5 PUNTOS)

Diagrama

Descripción generada automáticamente

Los tipos de datos son:

* Id\_Inmueble 🡪 Entero tamaño 5, auto-incrementado y clave principal.
* Tipo 🡪Varchar tamaño 30.
* Tamaño🡪 Decimal de 5,2
* Precio 🡪 entero de tamaño 5.
* Fecha\_compra🡪Tipo fecha.
* DNI 🡪 Varchar de 9 y clave principal.
* Nombre 🡪Varchar tamaño 30.
* Apellidos 🡪Varchar tamaño 40.
* Teléfono 🡪Varchar tamaño 9.

Crea las relaciones correspondientes. (La eliminación será de tipo cascada). (0.4 PUNTOS)

**CREATE TABLE Cliente (**

**Dni VARCHAR(9) PRIMARY KEY,**

**Nombre VARCHAR(30),**

**Apellidos VARCHAR(40),**

**Telefono VARCHAR(9)**

**);**

**CREATE TABLE Inmueble (**

**Id\_inmueble INT(5) PRIMARY KEY AUTO\_INCREMENT,**

**Tipo VARCHAR(30),**

**Tamaño DECIMAL(5,2),**

**Precio INT(5),**

**Fecha\_compra DATE,**

**Dni\_cliente VARCHAR(9),**

**CONSTRAINT fk\_cliente**

**FOREIGN KEY (Dni\_cliente)**

**REFERENCES Cliente(Dni)**

**ON DELETE CASCADE**

**)**

Modifica la tabla de INMUEBLE, añadiendo el campo de Población como varchar de tamaño 30 y Valencia por defecto. (0.2 PUNTOS)

**ALTER TABLE inmueble**

**ADD COLUMN Poblacion VARCHAR(30) DEFAULT 'valencia'**

Modifica a la tabla de INMUEBLE el campo que se llame PRECIO, que sea de tipo decimal (9,2). (0.25 PUNTOS)

**ALTER TABLE inmueble**

**MODIFY COLUMN Precio DECIMAL(9,2)**

Elimina de la tabla Clientes el campo de Teléfono. (0.25 PUNTOS)

**ALTER TABLE cliente**

**DROP COLUMN Telefono**

Añade a la tabla Clientes un campo llamado Población de Varchar 30 y por defecto Catarroja. (0.25 PUNTOS)

**ALTER TABLE cliente**

**ADD COLUMN Poblacion VARCHAR(30) DEFAULT 'Catarroja'**

Inserta los siguientes datos en la tabla CLIENTES: (0.2 PUNTOS)

*('12345678A', 'MARÍA', 'GARCÍA MÁS', 'PATERNA'), ('02345678A','FRANCISCO', 'RODRÍGUEZ SEVILLA', DEFAULT), ('22345678A','JOSÉ', 'GONZÁLEZ GARCÍA', 'TORRENT'), ('32345678A','MARTÍN', 'REYES CASAS', DEFAULT), ('42345678A','IRENE', 'GÓMEZ MADRID', 'TORRENT'), ('52345678A','FERNANDO', 'CÁMARA GUZMÁN', 'ALBAL' ), ('62345678A','MARÍA', 'SUÁREZ MAS', 'TORRENT'), ('72345678A','LUCIA', 'GIMÉNEZ SORIA', 'TORRENT'), ('82345678A','DAVID', 'MARTÍNEZ CONIA', 'TORRENT'), ('92345678A','ALEJANDRA', 'CAMARA MADRID', 'ALBAL')*

**INSERT INTO cliente VALUES ('12345678A', 'MARÍA', 'GARCÍA MÁS', 'PATERNA'), ('02345678A','FRANCISCO', 'RODRÍGUEZ SEVILLA', DEFAULT), ('22345678A','JOSÉ', 'GONZÁLEZ GARCÍA', 'TORRENT'), ('32345678A','MARTÍN', 'REYES CASAS', DEFAULT), ('42345678A','IRENE', 'GÓMEZ MADRID', 'TORRENT'), ('52345678A','FERNANDO', 'CÁMARA GUZMÁN', 'ALBAL' ), ('62345678A','MARÍA', 'SUÁREZ MAS', 'TORRENT'), ('72345678A','LUCIA', 'GIMÉNEZ SORIA', 'TORRENT'), ('82345678A','DAVID', 'MARTÍNEZ CONIA', 'TORRENT'), ('92345678A','ALEJANDRA', 'CAMARA MADRID', 'ALBAL')**

Inserta los siguientes datos en la tabla de INMUEBLES: (0.2 PUNTOS)

*('ADOSADO', 225, 225000, '12345678A', '2021-02-15', 'VALENCIA'), ('PISO', 85, 120000, '02345678A', '2021-03-15', 'VALENCIA'), ('LOCAL', 100, 25000, '02345678A', '2021-04-13', 'CATARROJA'), ('PISO', 125, 95000, '22345678A', '2021-04-15','VALENCIA'), ('PISO', 105, 105000, '72345678A', '2021-04-25', 'CATARROJA'), ('ADOSADO', 300, 335000, '82345678A', '2022-08-15', 'VALENCIA'), ('ÁTICO', 115, 200000, '62345678A', '2023-06-22', 'ALAQUÀS'), ('ÁTICO', 135, 180000, '52345678A', '2023-02-16', 'CATARROJA'), ('LOCAL', 50, 50000, '52345678A', '2023-03-31', 'ALAQUÀS'), ('PISO', 90, 103000, '32345678A', '2023-08-1', 'ALAQUÀS'), ('ÁTICO', 95, 155000, '32345678A', '2021-12-15', 'CATARROJA'), ('ÁTICO', 100, 175000, '02345678A', '2021-02-09', 'XIRIVELLA')*

**INSERT INTO inmueble(Tipo, Tamaño, Precio, Dni\_cliente, Fecha\_compra, Poblacion) VALUES ('ADOSADO', 225, 225000, '12345678A', '2021-02-15', 'VALENCIA'), ('PISO', 85, 120000, '02345678A', '2021-03-15', 'VALENCIA'), ('LOCAL', 100, 25000, '02345678A', '2021-04-13', 'CATARROJA'), ('PISO', 125, 95000, '22345678A', '2021-04-15','VALENCIA'), ('PISO', 105, 105000, '72345678A', '2021-04-25', 'CATARROJA'), ('ADOSADO', 300, 335000, '82345678A', '2022-08-15', 'VALENCIA'), ('ÁTICO', 115, 200000, '62345678A', '2023-06-22', 'ALAQUÀS'), ('ÁTICO', 135, 180000, '52345678A', '2023-02-16', 'CATARROJA'), ('LOCAL', 50, 50000, '52345678A', '2023-03-31', 'ALAQUÀS'), ('PISO', 90, 103000, '32345678A', '2023-08-1', 'ALAQUÀS'), ('ÁTICO', 95, 155000, '32345678A', '2021-12-15', 'CATARROJA'), ('ÁTICO', 100, 175000, '02345678A', '2021-02-09', 'XIRIVELLA')**

Crea la tabla de AUDITORÍA, con los siguientes campos y sus características: (0.2 PUNTOS)

|  |  |
| --- | --- |
| TABLA AUDITORÍA | |
| Campo | Tipos datos |
| Cod\_Auditoria | Int(5)/Auto-incrementado / Primary Key |
| DNI\_Ant | Varchar(9) |
| DNI\_Nue | Varchar(9) |
| ID\_INMUEBLE\_Ant | Int(5) |
| ID\_INMUEBLE\_Nue | Int (5) |
| Precio\_Ant | Decimal (9,2) |
| Precio\_Nue | Decimal (9,2) |
| Usuario | Varchar (30) |
| Fecha\_REG | Datetime |
| Acción | Varchar (15) |

**CREATE TABLE auditoria (**

**Cod\_auditoria INT(5) AUTO\_INCREMENT PRIMARY KEY,**

**Dni\_ant VARCHAR(9),**

**Dni\_nue VARCHAR(9),**

**Id\_inmueble\_ant INT(5),**

**Id\_inmueble\_nue INT(5),**

**Precio\_ant DECIMAL(9,2),**

**Precio\_nue DECIMAL(9,2),**

**Usuario VARCHAR(30),**

**Fecha\_reg DATETIME,**

**Accion VARCHAR(15)**

**)**

Cada vez que se inserte un nuevo registro en la tabla INMUEBLE, se tendrá que insertar de manera automática en la tabla AUDITORÍA, una fila indicando el DNI del cliente, el ID\_Inmueble, el precio del inmueble, el usuario que ha registrado la consulta, la fecha de registro de la acción y la acción que se ha realizado (en este caso ‘NUEVA’). (0.40 PUNTOS)

**CREATE TRIGGER nuevo\_ingreso**

**AFTER INSERT ON inmueble**

**FOR EACH ROW**

**INSERT INTO auditoria(Dni\_nue, Id\_inmueble\_nue, Precio\_nue, Fecha\_reg, Usuario, Accion)**

**VALUES (NEW.Dni\_cliente, NEW.Id\_inmueble, NEW.Precio, NOW(), CURRENT\_USER(), 'NUEVA')**

Cada vez que se modifique un registro en la tabla INMUEBLE, se tendrá que insertar de manera automática en la tabla AUDITORÍA, una fila indicando el DNI del cliente anterior y nuevo, el ID\_Inmueble anterior y nuevo, el precio del inmueble anterior y nueva, el usuario que ha registrado la consulta, la fecha de registro de la acción y la acción que se ha realizado (en este caso ‘MODIFICACIÓN’). (0.40 PUNTOS)

**CREATE TRIGGER actualizar\_registro**

**BEFORE UPDATE ON inmueble**

**FOR EACH ROW**

**INSERT INTO auditoria(Dni\_ant, Dni\_nue, Id\_inmueble\_ant, Id\_inmueble\_nue, Precio\_ant, Precio\_nue, Fecha\_reg, Usuario, Accion)**

**VALUES (OLD.Dni\_cliente, NEW.Dni\_cliente, OLD.Id\_inmueble, NEW.Id\_inmueble, OLD.Precio, NEW.Precio, NOW(), CURRENT\_USER(), 'MODIFICACION')**

Cada vez que se borre un registro en la tabla INMUEBLE, se tendrá que insertar de manera automática en la tabla AUDITORÍA, una fila indicando el DNI del cliente, el ID\_Inmueble, el precio del inmueble, el usuario que ha registrado la consulta, la fecha de registro de la acción y la acción que se ha realizado (en este caso ‘ELIMINACIÓN’). (0.40 PUNTOS)

**CREATE TRIGGER borrar\_registro**

**AFTER DELETE ON inmueble**

**FOR EACH ROW**

**INSERT INTO auditoria(Dni\_ant, Id\_inmueble\_ant, Precio\_ant, Fecha\_reg, Usuario, Accion)**

**VALUES (OLD.Dni\_cliente, OLD.Id\_inmueble, OLD.Precio, NOW(), CURRENT\_USER(), 'ELIMINACION')**

Realiza una consulta que muestre la cantidad total de clientes de cada población. (0.35 PUNTOS)

**SELECT Poblacion, COUNT(\*) AS TotalClientes**

**FROM cliente**

**GROUP BY Poblacion**

Consulta cual es la media del precio de los INMUEBLES por tipo. (0.35 PUNTOS)

**SELECT AVG(Precio)**

**FROM inmueble**

**GROUP BY Tipo**

Actualiza los inmuebles que son de tipo LOCAL ahora pasarán a ser de tipo BAJO. (0.35 PUNTOS)

**UPDATE inmueble SET Tipo = 'Bajo' WHERE Tipo = 'Local'**

Crea una tabla nueva que se llame INMUEBLESNUEVOS que contenga los campos de la tabla INMUEBLE de (Id\_Inmueble, Tipo, Precio y tamaño) que se hayan sido comprado en el año 2021. (0.35 PUNTOS)

**CREATE TABLE inmuebles\_nuevos SELECT Id\_inmueble, Tipo, Precio, Tamaño FROM inmueble WHERE YEAR(Fecha\_compra) = 2021**

Borra de la tabla INMUEBLESNUEVOS los que tengan un precio menor de 100.000 o mayor de 200.000. (0.35 PUNTOS)

**DELETE FROM inmuebles\_nuevos WHERE Precio < 100000 OR Precio > 200000**

Muestra el id\_inmueble y el tipo de inmueble que están por encima del precio medio de todos los inmuebles. (0.50 PUNTOS)

**SELECT Id\_inmueble, Tipo**

**FROM inmueble**

**WHERE Precio > (**

**SELECT AVG(Precio)**

**FROM inmueble**

**)**

Muestra el nombre y apellidos de los clientes que no han comprado ningún inmueble. (0.50 PUNTOS)

**SELECT Nombre, Apellidos**

**FROM cliente**

**WHERE Dni NOT IN (**

**SELECT Dni\_cliente**

**FROM inmueble**

**)**

Muestra los datos de los inmuebles que han sido comprados por clientes de Catarroja. (0.60 PUNTOS)

**SELECT \***

**FROM inmueble**

**WHERE Dni\_cliente IN (**

**SELECT Dni**

**FROM cliente**

**WHERE Poblacion = 'CATARROJA'**

**)**

Elimina los clientes que son de “Torrent” y han comprado un inmueble en la población de “Alaquàs”. (0.60 PUNTOS)

**DELETE FROM cliente WHERE Dni IN (**

**SELECT Dni\_cliente**

**FROM inmueble**

**WHERE Poblacion = 'Alaquàs'**

**)**

**AND Poblacion = 'Torrent'**

Crea un procedimiento almacenado que, indicando el código del inmueble y el precio nuevo, se actualice el precio del inmueble. (0.60 PUNTOS)

Crea un procedimiento almacenado para calcular el precio con descuento del 5% del inmueble que indique el usuario. Hazlo definiendo variables (descuento=0.05). Se debe mostrar el precio antiguo y el nuevo. (0.80 PUNTOS)

Crea un Trigger para que cada vez que se modifique el precio de los inmuebles, si se quiere poner un precio menor de 20000 que te ponga un 20000 en el precio, si el precio es mayor que 500000 que ponga un 50000 y si está entre 20000 y 50000 que ponga el precio que se indica el usuario. (0.80 PUNTOS)

Examen 2

Crea una base de datos llamada “TU NOMBRE”. (0.20 PUNTOS)

**CREATE DATABASE alberto\_bolta\_2**

Partiendo del siguiente diagrama, crea las siguientes tablas con los atributos correspondientes. (0.5 PUNTOS)

Diagrama

El contenido generado por IA puede ser incorrecto.

Los tipos de datos son:

ISBN 🡪 Entero tamaño 8, auto-incrementado y clave principal.

Título 🡪Varchar tamaño 30.

Autor🡪 Varchar tamaño 30.

Tipo 🡪 Varchar tamaño 15.

Fecha\_publicacion🡪Tipo fecha.

Stock 🡪Entero de tamaño 2.

Cod\_compra 🡪 Entero tamaño 6, auto-incrementado y clave principal.

Cantidad 🡪Entero de tamaño 2.

Cod\_cliente 🡪 Entero tamaño 6, auto-incrementado y clave principal.

Nombre 🡪Varchar tamaño 20.

Apellidos 🡪Varchar tamaño 30.

Teléfono 🡪Varchar tamaño 9.

Población🡪Varchar tamaño 20, por defecto Valencia.

Crea las relaciones correspondientes. (La eliminación será de tipo cascada). (0.4 PUNTOS)

**CREATE TABLE libros (**

**ISBN INT(8) AUTO\_INCREMENT PRIMARY KEY,**

**titulo VARCHAR(30),**

**autor VARCHAR(30),**

**tipo VARCHAR(15),**

**fecha\_publicacion DATE,**

**stock INT(2)**

**);**

**CREATE TABLE cliente (**

**cod\_cliente INT(6) AUTO\_INCREMENT PRIMARY KEY,**

**nombre VARCHAR(20),**

**apellidos VARCHAR(30),**

**telefono VARCHAR(9),**

**poblacion VARCHAR(20) DEFAULT 'Valencia'**

**);**

**CREATE TABLE compras (**

**ISBN\_libros INT(8),**

**CONSTRAINT fk\_libros**

**FOREIGN KEY (ISBN\_libros)**

**REFERENCES libros(ISBN)**

**ON DELETE CASCADE,**

**cod\_cliente INT(6),**

**CONSTRAINT fk\_cliente**

**FOREIGN key (cod\_cliente)**

**REFERENCES cliente(cod\_cliente),**

**cantidad INT(2),**

**cod\_compra INT(6) AUTO\_INCREMENT PRIMARY KEY**

**)**

Modifica la tabla de LIBROS, poniendo el campo de Stock como entero de tamaño 3. (0.2 PUNTOS)

**ALTER TABLE libros**

**MODIFY COLUMN stock INT(3)**

Añade a la tabla de LIBROS un campo que se llame precio, que sea de tipo decimal (5,2) y por defecto que el precio sea 14.99. (0.25 PUNTOS)

**ALTER TABLE libros**

**ADD COLUMN precio DECIMAL(5,2) DEFAULT 14.99**

Elimina de la tabla Clientes el campo de Teléfono. (0.25 PUNTOS)

**ALTER TABLE cliente**

**DROP COLUMN telefono**

Inserta los siguientes datos en la tabla de LIBROS: (0.20 PUNTOS)

('LOBA NEGRA', 'JUAN GÓMEZ-JURADO', 'SUSPENSE', '2019-10-24', 10, '20.99'), ('REINA ROJA', 'JUAN GÓMEZ-JURADO', 'SUSPENSE', '2018-11-08', 5, '19.85'), ('REY BLANCO', 'JUAN GÓMEZ-JURADO', 'SUSPENSE', '2020-11-05', 20, '22.99'), ('UN ÓCEANO PARA LLEGAR A TI', 'SANDRA BARNEDA', 'FICCIÓN', '2020-11-05', 12, '20.99'), ('LOS PILARES DE LA TIERRA', 'KEN FOLLETT', 'HISTÓRICO', '1989-05-05', 4, '12.99'), ('AQUITANIA', 'EVA GARCIA SAENZ DE URTURI', 'HISTÓRICO', '2020-11-05', 8, '20.99'), ('LA CATEDRAL DEL MAR', 'ILDEFONSO FALCONES', 'HISTÓRICO', '2006-01-01', 13, '15.99')

**INSERT INTO libros(titulo, autor, tipo, fecha\_publicacion, stock, precio) VALUES**

**('LOBA NEGRA', 'JUAN GÓMEZ-JURADO', 'SUSPENSE', '2019-10-24', 10, '20.99'),**

**('REINA ROJA', 'JUAN GÓMEZ-JURADO', 'SUSPENSE', '2018-11-08', 5, '19.85'),**

**('REY BLANCO', 'JUAN GÓMEZ-JURADO', 'SUSPENSE', '2020-11-05', 20, '22.99'),**

**('UN ÓCEANO PARA LLEGAR A TI', 'SANDRA BARNEDA', 'FICCIÓN', '2020-11-05', 12, '20.99'),**

**('LOS PILARES DE LA TIERRA', 'KEN FOLLETT', 'HISTÓRICO', '1989-05-05', 4, '12.99'),**

**('AQUITANIA', 'EVA GARCIA SAENZ DE URTURI', 'HISTÓRICO', '2020-11-05', 8, '20.99'),**

**('LA CATEDRAL DEL MAR', 'ILDEFONSO FALCONES', 'HISTÓRICO', '2006-01-01', 13, '15.99')**

Inserta los siguientes datos en la tabla CLIENTES: (0.2 PUNTOS)

('MARÍA', 'GARCÍA MÁS', 'PATERNA'), ('FRANCISCO', 'RODRÍGUEZ SEVILLA', DEFAULT), ('JOSÉ', 'GONZÁLEZ GARCÍA', 'TORRENT'), ('MARTÍN', 'REYES CASAS', DEFAULT), ('IRENE', 'GÓMEZ MADRID', 'TORRENT'), ('FERNANDO', 'CÁMARA GUZMÁN', 'ALBAL'), ('MARÍA', 'SUÁREZ MAS', 'TORRENT'), ('LUCIA', 'GIMÉNEZ SORIA', 'TORRENT'), ('DAVID', 'MARTÍNEZ CONIA', 'TORRENT'), ('ALEJANDRA', 'CAMARA MADRID', 'ALBAL')

**INSERT INTO cliente(nombre, apellidos, poblacion) VALUES**

**('MARÍA', 'GARCÍA MÁS', 'PATERNA'),**

**('FRANCISCO', 'RODRÍGUEZ SEVILLA', DEFAULT),**

**('JOSÉ', 'GONZÁLEZ GARCÍA', 'TORRENT'),**

**('MARTÍN', 'REYES CASAS', DEFAULT),**

**('IRENE', 'GÓMEZ MADRID', 'TORRENT'),**

**('FERNANDO', 'CÁMARA GUZMÁN', 'ALBAL'),**

**('MARÍA', 'SUÁREZ MAS', 'TORRENT'),**

**('LUCIA', 'GIMÉNEZ SORIA', 'TORRENT'),**

**('DAVID', 'MARTÍNEZ CONIA', 'TORRENT'),**

**('ALEJANDRA', 'CAMARA MADRID', 'ALBAL')**

Crea la tabla de AUDITORÍA, con los siguientes campos y sus características: (0.20 PUNTOS)

|  |  |
| --- | --- |
| TABLA AUDITORÍA | |
| Campo | Tipos datos |
| Cod\_Auditoria | Int(5)/Auto-incrementado / Primary Key |
| Cod\_Cliente\_Ant | Int(6) |
| Cod\_Cliente\_Nue | Int (6) |
| ISBN\_Ant | Int(8) |
| ISBN\_Nue | Int (8) |
| Cantidad\_Ant | Int(2) |
| Cantidad\_Nue | Int(2) |
| Usuario | Varchar (30) |
| Fecha\_REG | Datetime |
| Acción | Varchar (15) |

**CREATE TABLE auditoria (**

**cod\_auditoria INT(5) AUTO\_INCREMENT PRIMARY KEY,**

**cod\_cliente\_ant INT(6),**

**cod\_cliente\_nue INT(6),**

**ISBN\_ant INT(8),**

**ISBN\_nue INT(8),**

**cantidad\_ant INT(2),**

**cantidad\_nue INT(2),**

**usuario VARCHAR(30),**

**fecha\_reg DATETIME,**

**accion VARCHAR(15)**

**)**

Cada vez que se inserte un nuevo registro en la tabla COMPRAS, se tendrá que insertar de manera automática en la tabla AUDITORÍA, una fila indicando el código del cliente, el ISBN del libro, la cantidad adquirida, el usuario que ha registrado la consulta, la fecha de registro de la acción y la acción que se ha realizado (en este caso ‘NUEVA’). (0.35 PUNTOS)

**CREATE TRIGGER insertar\_compra**

**AFTER INSERT ON compras**

**FOR EACH ROW**

**INSERT INTO auditoria (cod\_cliente\_nue, ISBN\_nue, cantidad\_nue, usuario, fecha\_reg, accion)**

**VALUES (NEW.cod\_cliente, NEW.ISBN, NEW.cantidad, CURRENT\_USER(), NOW(), 'NUEVA')**

Cada vez que se modifique un registro en la tabla COMPRAS, se tendrá que insertar de manera automática en la tabla AUDITORÍA, una fila indicando el código del cliente anterior y nuevo, el ISBN del libro anterior y nuevo, la cantidad adquirida anterior y nueva, el usuario que ha registrado la consulta, la fecha de registro de la acción y la acción que se ha realizado (en este caso ‘MODIFICACIÓN’). (0.35 PUNTOS)

**CREATE TRIGGER modificar\_compra**

**BEFORE UPDATE ON compras**

**FOR EACH ROW**

**INSERT INTO auditoria (cod\_cliente\_ant, cod\_cliente\_nue, ISBN\_ant, ISBN\_nue, cantidad\_ant, cantidad\_nue, usuario, fecha\_reg, accion)**

**VALUES (OLD.cod\_cliente, NEW.cod\_cliente, OLD.ISBN, NEW.ISBN, OLD.cantidad, NEW.cantidad, CURRENT\_USER(), NOW(), 'MODIFICACION')**

Cada vez que se borre un registro en la tabla COMPRAS, se tendrá que insertar de manera automática en la tabla AUDITORÍA, una fila indicando el código del cliente, el ISBN del libro, la cantidad adquirida, el usuario que ha registrado la consulta, la fecha de registro de la acción y la acción que se ha realizado (en este caso ‘ELIMINACIÓN’). (0.35 PUNTOS)

**CREATE TRIGGER quitar\_compra**

**BEFORE DELETE ON compras**

**FOR EACH ROW**

**INSERT INTO auditoria (cod\_cliente\_ant, ISBN\_ant, cantidad\_ant, usuario, fecha\_reg, accion)**

**VALUES (OLD.cod\_cliente, OLD.ISBN, OLD.cantidad, CURRENT\_USER(), NOW(), 'ELIMINACION')**

Cada vez que se añade una compra, se tiene que actualizar el stock de libros que hay, restando la cantidad que se compra. (0.50 PUNTOS)

Realiza una consulta que muestre la cantidad total de clientes de cada población. (0.40 PUNTOS)

**SELECT COUNT(cod\_cliente), poblacion**

**FROM cliente**

**GROUP BY poblacion**

Consulta cual es la media del precio de los libros por tipo. (0.40 PUNTOS)

**SELECT AVG(precio), tipo**

**FROM libros**

**GROUP BY tipo**

Actualiza la tabla de Libros, todos los libros que valgan más de 20 euros y sean de suspense, van a costar 3 euros menos. (0.40 PUNTOS)

**UPDATE libros**

**SET precio = 3**

**WHERE precio > 20 AND tipo = 'suspense'**

Crea una tabla nueva que se llame LIBROSNUEVOS que contenga todos los campos que se hayan publicado en el año 2020. (0.40 PUNTOS)

**CREATE TABLE libros\_nuevos**

**SELECT \***

**FROM libros**

**WHERE YEAR(fecha\_publicacion) = 2020**

Borra de la tabla LIBROSNUEVOS los que sean de ficción o de Historia. (0.35 PUNTOS)

**DELETE FROM libros\_nuevos**

**WHERE tipo = 'Ficcion' OR tipo = 'Historia'**

Inserta los siguientes datos en la tabla compra (Cod\_cliente, ISBN y cantidad): (0.2 PUNTOS)

(1,1,2), (2,1,1), (2,3,1), (3,5,1), (6,3,2), (4,6,1), (7,1,2), (8,6,5), (7,5,1)

**INSERT INTO compras(cod\_compra, ISBN\_libros, cantidad) VALUES (1,1,2), (2,1,1), (2,3,1), (3,5,1), (6,3,2), (4,6,1), (7,1,2), (8,6,5), (7,5,1)**

Muestra el nombre y apellido de los clientes que han comprado más de un libro en una misma compra. (0.60 PUNTOS)

**SELECT nombre, apellidos**

**FROM cliente**

**WHERE cod\_cliente IN (**

**SELECT cod\_cliente**

**FROM compras**

**WHERE cod\_compra > 1)**

Muestra los datos de los libros que han sido comprados por clientes de Valencia. (0.70 PUNTOS)

**SELECT \***

**FROM libros**

**WHERE ISBN IN (**

**SELECT ISBN\_libros**

**FROM compras**

**WHERE cod\_cliente IN (**

**SELECT cod\_cliente**

**FROM cliente**

**WHERE poblacion = 'Valencia'))**

**\*creo que esta bien pero no me da ningún resultado lo cual es extraño**

Elimina los clientes que son de “Torrent” y han comprado un libro de tipo “Historia”. (Hay que referenciarlo con la tabla de libros) (0.70 PUNTOS)

**DELETE FROM cliente**

**WHERE cod\_cliente IN (**

**SELECT cod\_cliente**

**FROM compras**

**WHERE ISBN\_libros IN (**

**SELECT ISBN**

**FROM libros**

**WHERE tipo = 'HISTÓRICO'**

**)**

**)**

**AND poblacion = 'TORRENT';**

Crea un procedimiento almacenado que, indicando el código del libro y el precio nuevo, se actualice el precio del libro. (0.40 PUNTOS)

Crea un procedimiento almacenado para calcular el precio con descuento del 15% del libro que indique el usuario. Hazlo definiendo variables (descuento=0.15). Se debe mostrar el precio antiguo y el nuevo. (0.70 PUNTOS)

Crea un Trigger para que cada vez que se modifique el precio de los libros, si se quiere poner un número menor de 20 que te ponga un 20 en el precio, si el precio es mayor que 50 que ponga un 50 y si está entre 20 y 50 que ponga el precio que se indica el usuario. (0.80 PUNTOS)

**DELIMITER //**

**CREATE TRIGGER modificar\_precio\_libros**

**BEFORE UPDATE ON libros**

**FOR EACH ROW**

**BEGIN**

**IF NEW.precio < 20 THEN**

**SET NEW.precio = 20;**

**ELSEIF NEW.precio < 50 THEN**

**SET NEW.precio = 50;**

**END IF;**

**END; //**

**DELIMITER ;**