

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

Dado las tablas y campos siguientes que reflejan la base de datos de un Banco denominada BANCODAM creada en MYSQL, resolver las cuestiones:

Create Table Banco (cod_banco int not null, nombre varchar (50) not null, iban varchar (4) not null, num_cuenta_banco varchar(4) not null, direccion varchar (100) not null, telefono varchar (10) not null, constraint PK_Banco primary key (cod_banco));	Create Table Sucursales(cod_sucursal int not null, cod_banco int not null, nombre varchar(50) not null, direccion varchar (100) not null, telefono varchar (10) not null, num_cuenta_sucursal varchar(4) not null, constraint PK_Sucursales primary key (cod_sucursal), constraint FK_Sucursales foreign key (cod_banco) references Banco (cod_banco) on update cascade on delete cascade);
Create Table Movimientos(cod_movimiento int not null, cod_cliente int not null, descripcion varchar(50) not null, importe_abono decimal(8,2) , importe_cargo decimal(8,2) , fecha_mov datetime , constraint PK_Movimientos primary key (cod_movimiento), constraint FK_Movimientos foreign key (cod_cliente) references clientes(cod_cliente) on update cascade on delete cascade);	Create Table Clientes (cod_cliente int not null, nombre varchar(40) not null, apellidos varchar(80) not null, dni varchar(9) not null, saldo decimal(15,2) not null, num_cuenta varchar(20) not null, fecha_nacimiento datetime not null, cod_sucursal int not null, constraint pk_clientes primary key (cod_cliente), constraint fk_clientes foreign key (cod_sucursal) references sucursales(cod_sucursal) on update cascade on delete cascade);

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

1º Rellena varios registros en las tablas para comprobar posteriormente funcionamiento de la base de datos y preguntas posteriores. Mostrar imagen. (0,5 puntos).

2º Basándonos en las tablas propuestas. Crear una **función** que pasando como parámetro de entrada el número de cuenta bancaria del cliente, indique el nombre de la sucursal y el nombre del banco en una línea. Indica además como llamar a la función y muestra por pantalla un ejemplo. Tener en cuenta el formato de un número de cuenta bancaria. (2 puntos)



Ejemplo: **Banco.”Mi Banco” – Sucursal: “1º DAM”**

Datos Ejemplos

```
INSERT INTO Banco (cod_banco, nombre, iban, num_cuenta_banco, direccion, telefono)
VALUES
(1, 'CaixaBank', 'ES21', '2100', 'Madrid', '910001122'),
(2, 'BBVA', 'ES01', '0182', 'Bilbao', '99003344'),
(3, 'Santander', 'ES03', '0030', 'Cantabria', '97008899');

INSERT INTO Sucursales (cod_sucursal, cod_banco, nombre, direccion, telefono,
num_cuenta_sucursal)
VALUES (1, 1, 'Sucursal Las Palmas', 'Calle Mayor 2', '111111111', '0001'),
(2, 2, 'Sucursal Madrid', 'Avenida de la Libertad 6', '222222222', '0002'),
(3, 2, 'Sucursal Tenerife', 'Plaza del Sol 3', '333333333', '0003'),
(4, 3, 'Sucursal Barcelona', 'Calle Real 8', '444444444', '0004'),
(5, 3, 'Sucursal Sevilla', 'Avenida de la Victoria 11', '555555555', '0005');

INSERT INTO Clientes (cod_cliente, nombre, apellidos, dni, saldo, num_cuenta,
fecha_nacimiento, cod_sucursal)
VALUES
(1, 'Juan', 'González Pérez', '12345678A', 2500.00, '981822745628', '1985-04-15', 1),
(2, 'María', 'Sánchez López', '87654321B', 5000.00, '771036165462', '1990-02-20', 2),
(3, 'Pedro', 'García Ruiz', '13579086C', 10000.00, '323537161636', '1978-09-05', 3),
(4, 'Lucía', 'Martínez Gómez', '24680135D', 7500.00, '529865424364', '1982-12-10', 4),
(5, 'Antonio', 'Fernández Castro', '98765432E', 12000.00, '52811131182', '1995-07-23', 1),
(6, 'Sofía', 'López Pérez', '54321678F', 3200.00, '3235371777367', '1997-08-08', 1),
(7, 'Mario', 'Rodríguez García', '98765432F', 500.00, '52811131182', '1999-01-22', 2),
(8, 'Ana', 'Jiménez Martínez', '24681357G', 2500.00, '3235371777367', '1980-06-15', 3),
(9, 'Carlos', 'González Sánchez', '13579753H', 15000.00, '9818227456284', '1975-09-30', 4),
(10, 'Lucas', 'Romero Pérez', '73591068I', 2000.00, '981822745628', '1994-03-12', 2);

INSERT INTO Movimientos (cod_movimiento, cod_cliente, descripcion, importe_abono,
importe_cargo, fecha_mov)
VALUES
```

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

```
(1, 1, 'Ingreso nómina', 1500.00, NULL, '2023-03-01 12:00:00'),  
(2, 1, 'Pago factura', NULL, 50.00, '2023-03-05 15:30:00'),  
(3, 2, 'Transferencia recibida', 100.00, NULL, '2023-03-10 10:00:00'),  
(4, 2, 'Pago seguro', NULL, 30.00, '2023-03-12 18:45:00'),  
(5, 3, 'Ingreso extra', 3000.00, NULL, '2023-03-15 09:00:00'),  
(6, 3, 'Pago recibo', NULL, 100.00, '2023-03-16 11:00:00'),  
(7, 4, 'Transferencia enviada', NULL, 500.00, '2023-03-20 14:00:00'),  
(8, 4, 'Compra online', NULL, 75.00, '2023-03-22 20:30:00'),  
(9, 5, 'Ingreso regalo', 200.00, NULL, '2023-03-25 16:00:00'),  
(10, 5, 'Retirada efectivo', NULL, 1000.00, '2023-03-27 10:45:00');
```

Para hacer esta función se tiene en cuenta que no todo el número de cuenta del cliente almacena todo el formato del número de cuenta. Es decir, hay que consultar la tabla sucursal y banco para saber correctamente la información solicitada en la función.

```
DELIMITER $$  
  
CREATE FUNCTION DatosCuenta (NumCuenta varchar(24))  
RETURNS VARCHAR(50)  
DETERMINISTIC  
BEGIN  
    DECLARE iban VARCHAR(4);  
    DECLARE num_entidad VARCHAR(4);  
    DECLARE num_sucursal VARCHAR(4);  
    DECLARE num_cliente VARCHAR(20);  
    DECLARE NombreBanco varchar(50);  
    DECLARE NombreSucursal varchar(50);  
    DECLARE MensajeFinal VARCHAR(100);  
  
    SET iban = substring(NumCuenta,1,4);  
    SET num_entidad = substring(NumCuenta,5,4);  
    SET num_sucursal = substring(NumCuenta,9,4);  
    SET num_cliente = substring(NumCuenta,13,12);  
  
    Select sucursales.nombre, banco.nombre into NombreSucursal, NombreBanco  
    from clientes join sucursales  
    on clientes.cod_sucursal = sucursales.cod_sucursal  
    join banco on sucursales.cod_banco = banco.cod_banco  
    where clientes.num_cuenta= num_cliente  
    and banco.iban = iban  
    and banco.num_cuenta_banco=num_entidad  
    and sucursales.num_cuenta_sucursal = num_sucursal;  
  
    if NombreBanco is null then  
        SET MensajeFinal = "No existe ese número de cuenta";  
    else  
        SET MensajeFinal = concat("Banco.",NombreBanco,"-Sucursal:",NombreSucursal);  
    end if;  
    return MensajeFinal;  
  
END$$
```

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

```
DELIMITER ;
```

```
select DatosCuenta("ES2121000001981822745628");
```

3º Basándonos en las tablas propuestas y teniendo la tabla de **Movimientos** con registros. Crear un **procedimiento almacenado** que indique el mayor ingreso (campo abono en mi cuenta) que ha tenido un cliente en un mes en concreto de un año en concreto. Pasar por parámetro el código de cliente, mes y año. Muestra por pantalla un ejemplo de la ejecución del procedimiento (2 puntos)

Ejemplo:

Sucursal: 1DAM; Código Cliente: 2537; Mes 1/23: 250000€

Sucursal: 1DAM; Código Cliente: 2537; Mes 2/23: 3250000€

Sucursal: 1DAM; Código Cliente: 2537; Mes 3/23: 250000€

```
DELIMITER $$
CREATE PROCEDURE mayorIngreso (IN codcliente int, IN mes int, IN anio
varchar(4))
BEGIN

DECLARE nombre_cliente VARCHAR(40);
DECLARE nombre_sucursal VARCHAR(20);
DECLARE maximo_importe VARCHAR(50);
DECLARE mensaje VARCHAR(100);

SELECT c.nombre, s.nombre, MAX(m.importe_abono) INTO
nombre_cliente, nombre_sucursal, maximo_importe
FROM Sucursales as s
INNER JOIN Clientes as c ON s.cod_sucursal = c.cod_sucursal
INNER JOIN Movimientos as m ON c.cod_cliente = m.cod_cliente
WHERE c.cod_cliente = codcliente
AND YEAR(m.fecha_mov) = anio
AND MONTH(m.fecha_mov) = mes
GROUP BY c.nombre, s.nombre;

if maximo_importe is not null then
    SET mensaje = CONCAT('Sucursal: ', nombre_sucursal, ' Código cliente:
', codcliente, ' Mes ', mes, '/', anio, ': ', maximo_importe);
else
    SET mensaje = CONCAT("No existen datos para el Mes:", mes, '/', anio, ':0');
end if;
```

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

```
SELECT mensaje;
```

```
END$$
```

```
DELIMITER ;
```

```
call mayorIngreso(5, "3","2023");
```

4º Basándonos en las tablas propuestas y teniendo la tabla de **Cientes** con registros. Crear un **procedimiento almacenado** que disminuya el saldo de todos los clientes un 1,2% que pertenecen a la sucursal 3. También se insertará en la tabla Movimientos un cargo (gasto) por dicho importe con la descripción en dicha tabla de “Impuesto Revolucionario” (1,5 punto)

```
DELIMITER $$
```

```
CREATE PROCEDURE reduccionSaldo()
```

```
BEGIN
```

```
DECLARE cantidadMovimientos int;
```

```
UPDATE clientes
```

```
SET saldo = saldo - (saldo*0.012)
```

```
where cod_sucursal=3;
```

```
SELECT count(*) into cantidadMovimientos from Movimientos;
```

```
INSERT INTO Movimientos(
```

```
Select (cantidadMovimientos+cod_cliente) as cod_movimiento, cod_cliente, ("Importe  
Revolucionario") as descripcion, NULL as importe_abono, round(saldo*0.012,2) as
```

```
importe_cargo, current_date() as fecha_mov
```

```
from clientes
```

```
where cod_sucursal=3);
```

```
END$$
```

```
DELIMITER ;
```

```
Call reduccionSaldo;
```

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

5º Basándonos en las tablas propuestas. Crear un disparador que se produzca al insertar un **Movimiento** de cargo/abono en la cuenta. Dicho disparador debe controlar la no inserción de un movimiento de cargo en cuenta si el número total de cargos en ese mes y año es superior a 10000€. Muestra por pantalla la inserción y el resultado (1,5 punto)

```
DELIMITER //
CREATE TRIGGER ControlSaldo BEFORE INSERT ON Movimientos
FOR EACH ROW
BEGIN

DECLARE cantidad_ingresado INT;
DECLARE mes INT;
DECLARE anio INT;
SET mes = MONTH(NEW.fecha_mov);
SET anio = YEAR(NEW.fecha_mov);

SELECT SUM(importe_cargo) INTO cantidad_ingresado
FROM Movimientos
WHERE MONTH(fecha_mov) = mes
AND YEAR(fecha_mov) = anio;

IF (cantidad_ingresado + NEW.importe_cargo) > 10000 THEN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La columna importe_cargo se
pasa de los 10000€ este mes';
END IF;
END//
DELIMITER ;
```

```
INSERT INTO Movimientos (cod_movimiento, cod_cliente, descripcion,
importe_abono, importe_cargo, fecha_mov)
VALUES (36, 5, 'Ingreso regalo', NULL, 10000.00, '2024-03-25 16:00:00');
```

6º Crear un **procedimiento almacenado** que mediante un **Cursor** recorra la tabla **Clientes** e indique el mayor ingreso de cada cliente del BancoDam en el último trimestre del año pasado. La salida de este procedimiento debe mostrar la sucursal, el código de cliente, mayor ingreso por cliente. Muestra por pantalla un ejemplo de la ejecución del procedimiento (2,5 puntos)

Ejemplo:

Sucursal: 1Dam; Código Cliente: 2537; Mes 10/22: 250000€

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

Sucursal: 1Dam; Código Cliente: 2537; Mes 11/22: 500000€

Sucursal: 1Dam; Código Cliente: 2537; Mes 12/22: 4500000€

Suma Total del Trimestre por cliente: 5250000

Seguir con el siguiente cliente: ...

Antes de crear el procedimiento, creo un procedimiento similar al de la pregunta 3

```
DELIMITER $$
CREATE PROCEDURE mayorIngreso2 (IN codcliente int, IN mes int, IN anio
varchar(4), INOUT maximo_importe decimal(20,2), INOUT mensaje varchar (200))
BEGIN

DECLARE nombre_cliente VARCHAR(40);
DECLARE nombre_sucursal VARCHAR(20);
DECLARE maximporte VARCHAR(24);
SELECT c.nombre, s.nombre, MAX(m.importe_abono) INTO
nombre_cliente, nombre_sucursal, maximporte
FROM Sucursales as s
INNER JOIN Clientes as c ON s.cod_sucursal = c.cod_sucursal
INNER JOIN Movimientos as m ON c.cod_cliente = m.cod_cliente
WHERE c.cod_cliente = codcliente
AND YEAR(m.fecha_mov) = anio
AND MONTH(m.fecha_mov) = mes
GROUP BY c.nombre, s.nombre;

if maximporte is not null then
begin
SET mensaje = CONCAT('Sucursal: ', nombre_sucursal, ' Código cliente:
', codcliente, 'Mes ', mes, '/', anio, ': ', maximporte);
SET maximo_importe= cast(maximporte as decimal (20,2));
end;
else
begin
SET maximo_importe= 0;
SET mensaje = CONCAT("No existen datos para el Mes:", mes, '/', anio, ':0');
end;
end if;

END$$
DELIMITER ;
```

```
DELIMITER $$

CREATE PROCEDURE `CLIENTES_INGRESOS`_Trimestre()
```

ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

```
BEGIN
-- Declaración de variables
DECLARE IDSUCURSAL INT;
DECLARE IDCLIENTE INT;
DECLARE NOMSUCURSAL VARCHAR(50);
DECLARE NOMCLIENTE VARCHAR(50);
DECLARE IMP1 DECIMAL(20,2);
DECLARE IMP2 DECIMAL(20,2);
DECLARE IMP3 DECIMAL(20,2);
DECLARE mensaje1 VARCHAR(200);
DECLARE mensaje2 VARCHAR(200);
DECLARE mensaje3 VARCHAR(200);
DECLARE TOTAL DECIMAL(20,2);
DECLARE done INT DEFAULT 0;
DECLARE current_year VARCHAR(4) DEFAULT '2023';

-- Declaración del cursor
DECLARE mi_cursor CURSOR FOR
SELECT DISTINCT S.COD_SUCURSAL, S.NOMBRE, C.COD_CLIENTE
FROM CLIENTES C
JOIN SUCURSALES S ON C.COD_SUCURSAL = S.COD_SUCURSAL
JOIN MOVIMIENTOS M ON C.COD_CLIENTE = M.COD_CLIENTE
WHERE QUARTER(M.FECHA_MOV) = 4
AND YEAR(M.FECHA_MOV) = current_year;

-- Declaración de manejo de finalización del cursor
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

-- Apertura del cursor
OPEN mi_cursor;

-- Bucle para recorrer los registros
read_loop: LOOP
    FETCH mi_cursor INTO IDSUCURSAL, NOMSUCURSAL, IDCLIENTE;

    IF done THEN
        LEAVE read_loop;
    END IF;

    -- Llamadas a procedimientos almacenados para cada mes del trimestre
    CALL mayorIngreso2 (IDCLIENTE, 10, current_year, @IMP1, @mensaje1);
    CALL mayorIngreso2 (IDCLIENTE, 11, current_year, @IMP2, @mensaje2);
    CALL mayorIngreso2 (IDCLIENTE, 12, current_year, @IMP3, @mensaje3);

    -- Cálculo del total trimestral
    SET TOTAL = @IMP1 + @IMP2 + @IMP3;

    -- Imprimir el resultado
```


ACTIVIDAD PROCEDIMIENTOS ALMACENADOS EN MYSQL

```
SELECT CONCAT(@mensaje1,'',@mensaje2,'', @mensaje3,'','Saldo total
Trimestre: ', CAST(TOTAL AS CHAR)) AS resultado;
END LOOP;

-- Cierre y liberación del cursor
CLOSE mi_cursor;

END$$

DELIMITER ;
```

```
CALL CLIENTES_INGRESOS_Trimestre();
```