

**ITCR – Escuela de Ing. en Computación – Base de Datos 1 – Prof. Quiros –  
Diciembre 2020**

**Primera tarea programada (3era parte)**

**Uno.** Objetivo: Respecto de una base de datos física, escribir el código SQL que permita la actualización de entidades, la realización de una simulación de su operación, así como la realización de consultas.

**Dos. Nuevos requerimientos.**

**Dos.1. Cuenta Objetivo.** El archivo de simulación para esta tarea va a contener nodos para la creación de Cuentas Objetivo, el cual contiene el número de cuenta de la cuenta maestra, el número de cuenta de la cuenta objetivo, fecha final de la Cuenta Objetivo, el monto a ahorrar mensualmente, el día del mes para aplicar el ahorro y la descripción del objetivo; esto en los datos para las operaciones diarias.

De esta forma:

```
<CO CuentaMaestra="12345" CodigoCO="123" MontoAhorrar="5000" FechaFinal="2021-10-01" DiaDelMesAhorro="20"
Description="Vacaciones" />
```

La tasa de interés de la CO depende la cantidad de meses de su vigencia (fecha de creación a fecha de finalización). 0.5% de interés por cada mes, mes entero. El interés es anualizado. Si la suma de los meses es 5, la tasa 2.5% anualizada, por día sería 2.4% dividido entre 365.

El script de simulación debe calcular intereses diarios a las cuentas de CO.

Procesar intereses de CO: los intereses se acumulan, diariamente, hasta la finalización de la CO, debe crearse una tabla de MovimientosInteresCO, con créditos todos los días, el interés en día que corresponde al ahorro, se calcula previo a depósito de CO (el ahorro del mes). Se calcula el interés, se crea crédito y se acumula el monto (intereses acumulados).

En la simulación, su script, cuando procesa un día, antes de realizar el proceso de estados de cuenta, debe agregar código para 'procesar CO', el cual tiene 2 partes: procesar depósitos en CO y procesar redención de CO.

Procesar depósitos en CO: si el día del mes la fecha de operación coincide con el día el día para ahorrar en CO, debe realizar un débito por retiro de la cuenta de ahorro y por ese mismo monto, debe realizar un depósito en la CO, el saldo en cuenta de ahorro se disminuye y el saldo en CO se incrementa. La CO, tendrá su propia tabla de movimientos (MovimientoCO) y su propia tabla de TipoMovimientoCO, con 3 tipos de movimientos 1: Depósito por ahorro, 2: Depósito por redención de intereses y, 3: Redención de la CO. Si el saldo en la cuenta ahorro va a quedar negativo después del depósito en CO, entonces no se realiza.

**Procesar Redención de CO:** si la fecha de proceso es igual a la fecha final de la CO. Se redimen los intereses, esto es, se crea un débito por el monto total de los intereses en la tabla de movimientos de Intereses (de manera que los intereses acumulados quedan en cero) y se genera un crédito por el mismo monto en tabla de movimientosCO, aumentando el saldo. Luego se redime la CO, lo cual consiste en realizar un débito en MovimientosCO por todo el ahorro (más los intereses) de manera que su saldo queda en 0, y por ese mismo monto se realiza un depósito en cuenta de ahorro, por el monto del ahorro total realizado en la CO (más los intereses). La CO se desactiva.

La tabla de movimientosCO tiene la siguiente estructura (id int identity(1,1) primary key, idTipoMovimiento int, fecha date, monto money, descripción varchar (100))

La tabla de movimientosInteresCO tiene la siguiente estructura (id int identity(1,1) primary key, idTipoMovimiento int, fecha date, monto money, descripción varchar (100))

## **Dos.2 Bitácora de cambios en entidades.**

Se llevará bitácora de cambios sólo sobre 2 entidades CuentasObjetivo y Beneficiarios. Se crean 2 nuevas tablas TipoEvento (id PK, nombre) la cual tendrá 6 valores (1 Insertar Beneficiarios, 2: Modificar Beneficiario, 3: Eliminar (desactivar) Beneficiario 4: Insertar CO, 5: Modificar CO y 6: Eliminar (desactivar) CO) y la tabla de Eventos (id PK, idTipoEvento, idUser, IP, Fecha, XMLAntes, XMLDespues). Debe agregar triggers a las entidades de tal forma que, si hay inserción, se crea una fila en eventos, xmlAntes será vacío, y XMLDespues tendrá el contenido de toda la fila insertada en formato XML, para el borrado, el xml antes tendrá el contenido de toda la fila borrada en formato XML y XMLDespues será vacío; para modificación, el XmlAntes tiene los campos antes de la modificación y xmldespues los campos después de la modificación.

**Tres. Consultas para el administrador.** Hará una página para que el administrador corra consultas complejas. Serán 4 consultas. La página, para efectos prácticos no requerirá control de acceso mediante usuario y clave. Las consultas se especificarán posteriormente. Para la realización de las consultas debe crear al menos una vista.

## **Cuatro. Cambios en los xml que contienen datos de prueba.**

Respecto de datos básicos, son necesario 2 xml para llenar las tablas de: TipoMovimientoCO y TipoEvento.

Respecto del archivo para simular la operación: un nuevo nodo para la creación de CO.

La simulación se hará por un rango de 6 meses, no todos los días habrá datos, aunque el proceso debe realizarse para todos los días.

## **Cinco. Qué se pide.**

Código SQL para carga de datos básicos.

Código SQL para la simulación, que debe incluir el proceso de calculo diario de intereses a CO, redención de intereses, depósitos en CO y redención de CO.

Código SQL para las Consultas que realiza el administrador.

Código SQL para la implementación de la bitácora de cambios.

Capa Lógica para ejecutar las consultas para el administrador.

La documentación acostumbrada.

**Seis. Fecha de entrega.** <Por definir>.