

**ITCR – Escuela de Ing. en Computación – Base de Datos 1 – Prof. fquiros –
Octubre 2021**

Tercera tarea programada

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 xml 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 anualizada de la CO depende la cantidad de meses de su vigencia que se determina con base en fecha de creación y fecha final. La cantidad de meses mínima será de 3 meses, habrá una tabla catálogo que asocia la tasa de interés que se aplica según la cantidad de meses. El script de simulación debe calcular intereses mensuales respecto del saldo de las cuentas de CO, que se agregan al principal de la cuenta maestra cuando termina el plazo de la cuenta objetivo.

Procesar intereses de CO: los intereses se acumulan mensualmente en el día del ahorro de la CO y antes de aplicar el ahorro. Debe crearse una tabla de MovimientosInteresCO, con movimiento de crédito por el monto de los intereses que se calcula el día del ahorro y este monto es acumulado en un campo llamado InteresAcumuladoCO, en tabla CO. Entonces cada mes en día del ahorro se calcula el monto del interés respecto del saldo de la CO, se inserta un movimiento de crédito en MovimientosInteresCO y se acumula el monto de intereses acumulados.

Procesar ahorros (depósitos) en CO: si el día de la fecha de operación coincide con el día para ahorrar en CO, debe realizar un débito por retiro de la cuenta de ahorro (cuenta maestra) y por ese mismo monto, debe realizar un depósito (crédito) 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: Crédito por ahorro, 2: Crédito por redención de intereses y, 3: Débito por 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 (CO.InteresAcumuladoCO) en la tabla de MovimientoInteresesCO (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 de la CO (por el monto de los intereses acumulados). Luego se redime la CO, lo cual consiste en realizar un débito en MovimientosCO por todo el ahorro (más los intereses) de manera que el 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.

Resumen movimientos en Día de Ahorro

- Calcular intereses sobre saldoCO
- Agregar movimiento crédito monto de intereses en MovimientoInteresesCO
- Actualizar CO.Intereses acumulados
- Débito por monto de ahorro en CA y actualizar saldo
- Crédito por monto de ahorro en CO y actualizar saldo

Resumen movimientos en finalización de CO

- Débito en MovimientoInteresesCO por el monto de los intereses acumulados, los intereses acumulados quedan en cero.
- Crédito en MovimientosCO por el monto de los intereses acumulados, actualizar saldo CO
- Débito en MovimientosCO por el monto del SaldoCO, saldoCO queda en 0
- Crédito en Movimientos CA por el monto aplicado en SaldoCO

La Tabla TipoMovimientoCO (Id Int PK, NombreMovimiento, Operacion varchar(1)). Operación 1: crédito, 2: débito.

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

La tabla de movimientosInteresesCO tiene la siguiente estructura (id int identity(1,1) primary key, fecha date, monto money, NuevoInteresAcumulado, descripción varchar (100)). Si monto es positivo es crédito por cálculo de intereses mensual, si es negativo es un débito por redención de intereses.

Hay que recordar que, si el día del ahorro es 31, y el mes de la fecha de operación es de 20 días, entonces debe ser aplicado el 1 del siguiente mes. Ejemplo si el día de ahorro es 31 y la fecha de operación es 30 de abril, entonces los intereses deben calcularse el 1 de mayo para las CO que ahorran 31. Esta misma regla aplica para las cuentas de ahorro que cierran o abren EstadoCuenta los 31.

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). En XML de catálogos vendrán estos valores.

Debe agregar triggers o lógica en sus SP de crud de estas 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 necesarios 3 xml para llenar las tablas de: TasaInteresCO, 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 cálculo diario de intereses a CO, redención de intereses, depósitos en CO y redención de CO. Toda la actualización en las tablas de CuentaAhorro, MovimientosCA, EstadoCuenta, CO, MovimientoCO y MovimientoInteresesCO para un cliente en una fecha de operación debe ser transaccional y la transacción de BD NO debe tener flujo de control (No estructura IF).

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 del administrador.

La documentación acostumbrada.

Seis. Fecha de entrega. 22 de noviembre.

Siete. Comentario respecto Proyecto Final. Si el promedio de las 2 primeras tareas programadas es mayor a 80 y la tercera tarea programada tiene una calificación mayor o igual a 85 se eximen del proyecto final, la nota del proyecto final será el promedio de las 3 tareas.

Si no se cumple la condición anterior, el proyecto final será la entrega del proyecto completo (o sea todos los entregables de tarea 1, 2 y 3) el 1 de diciembre con todas las correcciones producto de la revisión de las 3 tareas.

Ocho. Tomar en cuenta. Al evaluar esta tarea o el proyecto final, el profesor calificara que no haya código redundante de SQL, que es código optimo, y que la interfaz del usuario (solo para proyecto final) es usable y elegante.