ITCR. Escuela de Computación. Bases de datos 1. Prof. fquiros.
Segunda tarea programada, versión no final

Software para una Municipalidad.

Cambios:

- Mejora en el ejemplo del calculo del monto del movimiento de consumo de agua (diferencia entre lectura del medidor y M3 acumulados).
- Se agrega atributo de Monto Mínimo de Recibo, en la entidad CCobro_Agua
- Especificación del cálculo del Recibo de agua
- Estado de los recibos
- Aclaración sobre la IP en tabla de bitácora de cambios
- Validaciones y manejo de errores.

1. Objetivos

- Realizar cambios en una base de datos física.
- Generar, leer y desplegar formato json.
- Implementar procesos masivos de actualización utilizando transacciones de BD.
- Mejorar un sitio web funcionando (el de administradores).
- Crear un nuevo sitio web para los usuarios.

2. Descripción

Registro de consumo de agua, emisión de recibos, actividades de corta y reconexión de agua, bitácora de cambios de las entidades.

Registro de consumo de agua

Diariamente se procesa un archivo xml con las lecturas del medidor de agua que se hicieron ese día así como ajustes al consumo debido a reclamos o lecturas incorrectas del medidor de agua, el xml tiene la siguiente estructura <consumo NumeroFinca=”12345” LecturaMedidorM3=”1075”/> o <AjusteConsumo NumeroFinca=”12345” M3=”5” Razón=”lectura erronea”/> o <AjusteConsumo NumeroFinca=”12345” M3=”-8” Razón=”Ajuste por reclamo”/>. Solo se hace una lectura por mes de lectura del medidor de agua a una propiedad.

Para efectos de reclamos, trazabilidad, big data, etc. Se mantiene un registro de todas las lecturas y ajustes de consumo de agua, tal que el valor de M3Acumulados se puede reconstruir en cualquier momento, así como el valor de M3Acumulado luego del registro de cada lectura o ajuste.

Para la propiedad se conoce el acumulado de M3 de agua, por ejemplo: Propiedad=77345, M3Acumulados =12320, y la lectura es: <consumo Numfinca="77345" LecturaM3="12405"> ===> se tiene que generar un movimiento por 85 M3, que es la diferencia en el atributo M3Acumulados y la lectura M3 indicada en el XML

Generación de recibos.

Para efectos de emisión del recibo de agua, agregamos un nuevo atributo en la entidad CCobro_Agua, que es MontoMinimoRecibo (el otro atributo es ValorM3, que es el valor de un m3 de agua necesario para obtener el monto de recibo. A cada propiedad se le agregan 2 atributos que se llamaran M3Acumulados de agua, M3AcumuladosUltimoRecibo, los cuales NO son editables, y se inicializan en 0 una vez que la propiedad es insertada.

Se genera un recibo de cobro de agua, a todas las propiedades asociados con concepto de cobro agua, el monto del recibo se calcula asi:

Monto Recibo Agua =

```
case when (AcumuladoActualM3-AcumuladoUltimoRecibo)*ValorM3>MontoMinimoRecibo
     then (AcumuladoActualM3-AcumuladoUltimoRecibo)*ValorM3
     else MontoMinimoRecibo
end
```

Al procesar fechas del archivo XML operación, llegado el día en que se generan recibos para los otros conceptos de cobro (ej.: recolección de basura, patentes, limpieza de parques, ….), estos se deben generar masivamente y asociar con las propiedades que tengan relación con los conceptos de cobro cuyo día `de corte corresponde a esa fecha. Ejemplo, si el concepto de cobro "recolección de basura" tiene como día de corte los 12 de cada mes, entonces si la fecha de operación es 12 de mayo, se genera un recibo por el monto que corresponda a las propiedades que tienen asociación con concepto de cobro "recolección de basura"; si el monto del recibo para ese concepto de cobro es porcentual, este se aplica al valor de propiedad para determinar el monto del recibo.

Los estados de un recibo son 0: Pendiente de pago (estado default), 1: Pagado, 3: Anulado. En esta tarea programada no se anulan recibos.

Ordenes de corta y reconexión

Todos los días se generan las órdenes de cortes de agua a las propiedades que tengan más de un recibo pendiente de agua, y se genera un recibo del tipo "Reconexión de agua" (si no existe concepto de cobro por esta razón, hay que generar un nuevo nodo xml para ello), el cual tiene un monto fijo. Igual todos los días se genera ordenes de reconexión, para quienes tenga el agua cortada, hayan pagado todos los recibos vencidos de agua y hayan pagado el recibo de reconexión.

Pago de los recibos.

El xml de operación diaria, también incluye nodos de pago de recibo, con este formato <Pago NumFinca="1234" TipoRecibo="3">, cuyo proceso debe generar el pago del recibo pendiente más antiguo cuyo concepto de cobro sea id=3, y si la fecha de operación es superior a la fecha máxima de pago del recibo; se debe calcular el monto de los intereses moratorios, crear un recibo por intereses moratorios y pagarlo.

El monto del recibo de intereses moratorios se calcula así:

FechaMax::=Fecha Máxima Pago del Recibo (por ej: recibo de agua). Es la fecha de emisión del recibo a la cual se le agregan la cantidad de días de gracia para pagarlo (QDiasVencimiento: días para que expire en la base de datos), información que se encuentra en la tabla de Concepto de Cobro.

FechaActual:=Fecha del archivo de operación de XML que se está procesando.

TasaInteresMoratorio: Atributo asociado a cada Concepto de Cobro.

MontoRecibo::=Monto a cobrar en el recibo que se esa pagando.

MontoInteresMoratorio::=

```
case
when FechaActual<=FechaMax then 0 –- no tiene que generarse recibo de int moratorios
else (MontoRecibo*TasaInteres/365)*abs(datedif(dias, FechaMax, FechaActual))
end
```

Si por alguna razón, para una propiedad se pagan uno o varios recibos el mismo día, se genera un solo comprobante de pago, que contendrá la suma del monto de todos los recibos pagados. Pueden suponer que los datos en el archivo XML de operación, los pagos de un cliente en un mismo día, vienen juntos; o para más seguridad el proceso que realiza los pagos los pueden ordenar para que estén juntos al ser procesados.

Todos los procesos masivos tienen que ser atómicos, o se generan todos los recibos, o todos los pagos, o todas las cortas de agua, o no se genera nada.

Cambios de valor de propiedad

Un nuevo nodo xml debe ser procesado <PropiedadCambio Numfinca=”1234” NuevoValor=” 100000”/>, el cual modifica el valor de una propiedad.

Bitácora de cambios de las entidades.

Todo cambio que haga un usuario administrador, sobre las entidades: propietarios, propiedades, usuarios, propiedades vs propietarios o propiedades vs usuarios, debe quedar almacenado en una tabla de bitácora de cambios, debe quedar el registro previo a la modificación y el registro posterior ambos representados en Json. En el caso de cambios en propiedades, ese ingreso a la bitácora debe hacerse mediante un trigger.

EntityType ::= 1: Propiedad, 2: Propietario, 3: User, 4: Propiedad vs Propietario, 5 Propiedad vs Usuario, 6: PropietarioJuridico, 7: Concepto de Cobro

La tabla de bitácora de cambios tendrá el siguiente formato:

```
(id int identity (1, 1) primary key,
IdEntityType int,                    -- referencia a table EntityType
EntityId int,                        -- Id de la entidad siendo actualizada
jsonAntes varchar (500),
jsonDespues varchar (500),
insertedAt datetime,                 -- estampa de tiempo de cuando se hizo la actualización
insertedby varchar (20),             -- usuario persona que hizo la actualización
insertedIn varchar(20)               -- IP desde donde se hizo la actualización, NO la IP del servidor,
sino la del usuario que debe capturarse en capa lógica
)
```

En un insert, el valor de jsonAntes es null o vacío, en un borrado el valor de jsonDespues es Null o vacío.

EntityId: la llave de la entidad que se está procesando.

Validaciones de los parámetros de entrada en SP y manejo de errores.

Los SP deben implementar try cath, para capturar errores, si error sucede dentro de una transacción debe hacer rollback, el manejo de errores puede hacerse mediante un throw, raise error, o devolviendo algún código de error en parámetro de salida, o en el @return_value. El único valor de entrada que se valida en SP que hacen CRUD, es que el @inId, referido a la entidad que debe consultarse, actualizarce o borrarse, debe existir.

3. **Que se pide**:

- Script para proceso de xml de operación, ejecutando día por día.
- Un ambiente web para administradores que actualiza las entidades (primera tarea programada), debe incluir una consulta en la cual se incluye el tipo de entidad, un rango de fechas y se despliegan cronológicamente los cambios en la entidad, fecha, ip desde donde se hizo y su autor, ese despliegue tiene que ser bonito (user friendly, o sea que NO json plano).
- Un ambiente web para usuarios ciudadanos de la municipalidad, que al ingresar verá sus propiedades asignadas, escoge una y en ella puede consultar los recibos pendientes, los recibos pagados y sus comprobantes de pago.
- Ambos ambientes web se ingresan desde la misma página de login.
- Documentación en una bitácora, ojalá con registro de entrada en git hub.

Fecha de entrega: 5 de Julio.