

Migracion de una aplicacion TPS a SQL Server

(Hacer backup de todo primero!)

Resumen

Hay dos cosas a tener en cuenta al crear una base de datos SQL basada en un sistema existente de Archivos TPS: Convertir las definiciones de datos y convertir los datos.
La conversion de datos lo veremos mas adelante...

Convertir las definiciones de datos (Creando un DCT nuevo)

La primer tarea es crear un nuevo DCT

Necesitamos mover la definicion de las tablas TPS existentes al DCT nuevo y convertir las tablas a SQL (cambiar el driver, etc..). Esto se puede hacer de dos maneras: Copiar el viejo DCT al DCT nuevo y cambiar cada tabla una a la vez; o exportar el viejo DCT a TXD, modificar el TXD e importar el TXD en el DCT nuevo.

Utilizaré la segunda técnica, la cual permite hacer cambios masivos. Sin embargo, este sistema es mucho más peligroso, así que hay que hacerlo con mucho cuidado.

Para la modificacion masiva utilizaremos un editor de texto, por ejemplo el www.TextPad.com

Driver Name

Este es el cambio más obvio. Necesitamos cambiar 'TopSpeed' por 'MsSQL'. No lo hagamos todavia... Veamos unos detalles antes.

OWNER

Todas las tablas SQL requieren la propiedad OWNER, que indique cómo conectarse con la base de datos donde estan las tablas. El OWNER debe ser siempre una variable. Si usted ya tiene algunas tablas con la propiedad OWNER, debemos borrar estos OWNER primero.

Luego podemos reemplazar "'TopSpeed') " con
"'MSSQL'),OWNER(GLO:Conexion)"

Campos STRING => CSTRING

En la mayoría de las bases SQL los espacios al final son importantes cuando verificamos si un campo es igual a otro. Por lo tanto 'Perez' ~= 'Perez '. Normalmente esto es importante en los componentes de las claves que se usan para relaciones o filtros. Si usamos CSTRINGs en lugar de STRINGs, entonces el Driver de Clarion trata los campos alfanumericos de la misma manera que el SQL.

Tengamos en cuenta que el Cstring tiene un caracter adicional para fin de cadena.

LONG => DATE and TIME (generalmente)

En los sistemas TPS, generalmente los datos de fecha y de la hora se almacenan como LONG. Estos se deben convertir a campos DATE y TIME, para que los datos sean facilmente accesibles desde herramientas de terceros. O sea si dejamos los campos como LONG @D6 lo vamos a poder seguir trabajando sin problemas desde Clarion, pero no se van a ver correctamente desde el SQLEditor o desde Cristal Report, Reporting Services, etc.

Tambien existen formulas de conversion de fecha clarion a SQL, pero en general se complican bastante las consultas.

RECLAIM

Los drivers SQL no soportan este atributo

Sacar NOCASE

(O sea marcar Case Sensitive en las claves)

La mayoría de las bases SQL solamente usan el concepto del sensible a mayusculas a nivel de toda la base de datos. No se puede especificar una llave como sensible y otra como insensible a mayusculas.

Por esta causa, las bases SQL no requieren este atributo NOCASE. En realidad , usar las claves sin case sensitive puede degradar mucho la performance, ya que las consultas generadas por el driver agregan la funcion UCASE, lo cual evita que se usen correctamente las claves definidas en el motor.

Integridad referencial

Cambiar la Integridad referencial para que la maneje el Server, no Clarion.

Desde el punto de vista del reemplazo en el TXA, seria:

CASCADE por CASCADE_SERVER;

RESTRICT por RESTRICT_SERVER;

CLEAR por CLEAR_SERVER

Importacion del TXD

Luego de estas modificaciones, importamos el TXD.

Modificaciones al DCT

Clave unica (Primaria)

Asegurarnos de que todas las tablas tengan una clave unica. En SQL no existe el concepto de puntero del archivo, o numero de registro.

Entonces para que el Driver pueda actualizar un registro necesitamos indicarle cual campo es el que identifica univocamente cada registro.

Para esto necesitamos tener por lo menos una clave unica por cada archivo.

Tengamos en cuenta que un PUT es convertido por el driver en una sentencia SQL del tipo:

```
UPDATE TABLA SET CAMPOS=VALORES WHERE ClaveUnica=Valor
```

No usar GROUPS en las claves

Se pueden usar campos que formen parte de grupos como componentes de una clave. Pero no se puede usar el campo GROUP como componente de la clave. Si tenemos claves con campos del tipo GROUP, necesitamos cambiar la clave para que use cada campo dentro del group.

Conversion de las definiciones de datos

Crear un SQL Script

Para crear el SQL Script necesitaremos los 2 diccionarios de datos. Uno es el que acabamos de crear. El otro es un diccionario de datos SQL. Un diccionario de datos SQL es ni mas ni menos que una base de datos (o mejor dicho, la definicion de su estructura). Podemos crear una base de datos nueva o usar una existente.

Ejecutar el Dictionary Synchronizer para crear un SQL Script

Ejecutar el Dictionary Synchronizer. El diccionario destino va a ser la base de datos donde crearemos las tablas.

El "Source DCT" va a ser el DCT que acabamos de crear y el "destination DCT" nuestra base SQL.

Una vez que llegamos a la pantalla del synchroniser tenemos que "copiar" todas las tablas a nuestra base SQL. La manera mas facil es seleccionar el inicio del arbol, click con el boton derecho del mouse y elegimos ADD.

Ejecutar el un Utility Template para crear un SQL Script

Existen varios Utility Template para generar Scripts SQL circulando gratis dentro de la comunidad de usuarios.

Ejecutar el Script

Para esto usaremos el SQL Query Analyzer, el osql, winSQL, o cualquier herramienta que nos permita ejecutar scripts SQL en el motor.

Conexion a la base de datos.

Recordemos que le pusimos la variable Global GLO:Conexion a todas nuestras tablas, obviamente deberemos declararla en la aplicacion o en el DCT.

En esa variable vamos a guardar la informacion de Conexion con la base de datos. En el caso del Driver MSSQL en esa variable deberemos asignar un String del tipo server,base,usuario,password.

Esta informacion podriamos leerla desde un INI o pedirla al usuario desde una pantalla que se ejecute antes de que se abra cualquier archivo.

La conexion con el server se realiza la primera vez que se abre un archivo. Necesitaremos abrir un archivo en nuestra pantalla de login, o bien en el Frame. Por mas que ese archivo se cierre inmediatamente, la conexion se mantendra abierta hasta que la aplicacion termine o bien hasta que se llame a PROP:Disconnect.

En el Anexo Strings de Conexion detallamos algunos de los posibles String de conexion dependiendo de diferentes drivers ODBC.