Softland
Lo hacemos fácil

Av. Libertador 6343 – 11° piso
(C1428ARG) - Ciudad Autónoma de Buenos Aires
Tel. +54 11 4789-7600

www.gruposoftland.com.ar

Guía de implementación
Softland en SQL Server 2012

Versión 1.0

Tabla de Contenido

Guía de implementación

1. Introducción

Este documento describe las nuevas características de implementación de las consultas de Outer Join de SQL Server 2012 y cómo deben ser implementadas desde Softland para obtener los mismos datos. También explica cómo deben ser utilizados desde el módulo del Generador de Reportes para obtener resultados consistentes con las implementaciones existentes en versiones de SQL Server 2008 R2 y previas.

2. Migración a SQL Server 2012

2.1 Ausencia de Compatibilidad con versiones anteriores

SQL Server 2012 ya no soporta el modo de compatibilidad de base de datos con SQL Server 2000 por lo que ya no se reconoce la sintaxis clásica de Outer Join "*=", la misma debe ser reemplazada por la cláusula ANSI "LEFT OUTER JOIN" pero existen cambios de criterio de cómo se procesa la información que la consulta devuelve.

Este cambio de comportamiento debe ser revisado en todos los puntos donde se han escrito implementaciones que involucran sentencias SQL de usuario como ser Reportes de Usuario, Jobs del administrador de Tareas, Controllers de usuario, y cualquier otro script SQL que se pueda estar ejecutando en la base de datos.

3. Cambio General Comportamiento de las sentencias de Join

3.1 Introducción al Join de Tablas

Suponga que tenemos dos tablas, EMPLEADO y FAMILIA

EMPLEADO

CODIGO	NOMBRE
1	Juan Perez
2	Ernesto Alvarez
3	Francisco Gonzalez
4	Nahuel Testa
5	Andrea Navarro
6	Ines Moreno
7	Carman Lopez
8	German Bringas
9	Andrea Pizzini

FAMILIA

CODEMP	FAMILIAR	RELACION	EDAD
1	Ernestina Perez	HIJO	7
1	Morena Perez	HIJO	10
1	Silvia Gonzalez	CONYUGUE	41
3	Andrea Lopez	CONYUGUE	32
3	Ignacio Gonzalez	HIJO	12
4	Sandra Romero	CONYUGUE	37
5	Esteban Fels	HIJO	5
5	Lucila Fels	HIJO	8
5	Miguel Fels	CONYUGUE	34
6	Andres Rodriguez	HIJO	3
7	Carlos Ibañez	CONYUGUE	52
7	Francisco Ibañez	HIJO	5

Un Join en la sintaxis clásica de SQL Server se escribía juntando varias tablas en el FROM e igualando las columnas en la sección del WHERE

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO, FAMILIA
WHERE CODIGO = CODEMP
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Ernestina Perez	HIJO	7
1	Juan Perez	Morena Perez	HIJO	10
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
3	Francisco Gonzalez	Ignacio Gonzalez	HIJO	12
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Esteban Fels	HIJO	5
5	Andrea Navarro	Lucila Fels	HIJO	8
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
6	Ines Moreno	Andres Rodriguez	HIJO	3
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52
7	Carman Lopez	Francisco Ibañez	HIJO	5

La sintaxis ANSI para realizar una Join es a través de la cláusula INNER JOIN

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO INNER
JOIN FAMILIA ON (CODIGO = CODEMP)
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD

1	Juan Perez	Ernestina Perez	HIJO	7
1	Juan Perez	Morena Perez	HIJO	10
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
3	Francisco Gonzalez	Ignacio Gonzalez	HIJO	12
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Esteban Fels	HIJO	5
5	Andrea Navarro	Lucila Fels	HIJO	8
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
6	Ines Moreno	Andres Rodriguez	HIJO	3
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52
7	Carman Lopez	Francisco Ibañez	HIJO	5

Este tipo de Join deja fuera de la lista a los empleados que no tienen conyugue o hijos

Para ver a todos los empleados y sus vínculos familiares en caso que corresponda, se debe ejecutar una sentencia conocida como Outer Join

3.2 Outer Join

En la sintaxis clásica se representaba con un "*=" con el "=" junto al campo de la tabla donde se encuentran los valores que pueden ser omitidos.

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO, FAMILIA
WHERE CODIGO *= CODEMP
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Ernestina Perez	HIJO	7
1	Juan Perez	Morena Perez	HIJO	10
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
2	Ernesto Alvarez	NULL	NULL	NULL
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
3	Francisco Gonzalez	Ignacio Gonzalez	HIJO	12
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Esteban Fels	HIJO	5
5	Andrea Navarro	Lucila Fels	HIJO	8
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
6	Ines Moreno	Andres Rodriguez	HIJO	3
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52
7	Carman Lopez	Francisco Ibañez	HIJO	5
8	German Bringas	NULL	NULL	NULL

9	Andrea Pizzini	NULL	NULL	NULL

En la sintaxis ANSI esta sentencia debe ser escrita como:

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO LEFT OUTER JOIN FAMILIA ON (CODIGO = CODEMP)
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Ernestina Perez	HIJO	7
1	Juan Perez	Morena Perez	HIJO	10
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
2	Ernesto Alvarez	NULL	NULL	NULL
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
3	Francisco Gonzalez	Ignacio Gonzalez	HIJO	12
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Esteban Fels	HIJO	5
5	Andrea Navarro	Lucila Fels	HIJO	8
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
6	Ines Moreno	Andres Rodriguez	HIJO	3
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52
7	Carman Lopez	Francisco Ibañez	HIJO	5
8	German Bringas	NULL	NULL	NULL
9	Andrea Pizzini	NULL	NULL	NULL

3.3 Diferencias en el Outer Join

Suponga que nos piden conocer todos los empleados que tienen hijos con edad menor a 12 años

Utilizando la sintaxis clásica escribíamos

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO, FAMILIA WHERE CODIGO *= CODEMP AND RELACION = 'HIJO' AND EDAD <= 12
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Ernestina Perez	HIJO	7
1	Juan Perez	Morena Perez	HIJO	10
2	Ernesto Alvarez	NULL	NULL	NULL
3	Francisco Gonzalez	Ignacio Gonzalez	HIJO	12
4	Nahuel Testa	NULL	NULL	NULL
5	Andrea Navarro	Esteban Fels	HIJO	5
5	Andrea Navarro	Lucila Fels	HIJO	8

6	Ines Moreno	Andres Rodriguez	HIJO	3
7	Carman Lopez	Francisco Ibañez	HIJO	5
8	German Bringas	NULL	NULL	NULL
9	Andrea Pizzini	NULL	NULL	NULL

El resultado es la lista de empleados que tienen hijos menores de 12 años, pero incluye también a los que no tienen hijos. Este es el comportamiento clásico del Join, ya que utilizando la sintaxis clásica PRIMERO se aplica el Filtro, y SEGUNDO se hace el Join de los resultados.

Para cumplir con la misma petición utilizando la sintaxis ANSI, la query se debería escribir

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO LEFT OUTER JOIN FAMILIA ON (CODIGO = CODEMP) WHERE RELACION = 'HIJO' AND EDAD <= 12
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Ernestina Perez	HIJO	7
1	Juan Perez	Morena Perez	HIJO	10
3	Francisco Gonzalez	Ignacio Gonzalez	HIJO	12
5	Andrea Navarro	Esteban Fels	HIJO	5
5	Andrea Navarro	Lucila Fels	HIJO	8
6	Ines Moreno	Andres Rodriguez	HIJO	3
7	Carman Lopez	Francisco Ibañez	HIJO	5

Como se puede observar en el resultado, hay un cambio de comportamiento en cómo se procesan los datos

Esto se debe a que la sintaxis ANSI PRIMERO se realiza el Join y SEGUNDO filtra los resultados a diferencia de la sintaxis clásica PRIMERO se aplica el Filtro, y SEGUNDO se hace el Join de los resultados, con lo que no importa cómo se escriba el filtro siempre se incluirán los registros de la tabla origen (empleados) que no tengan su correlato en la tabla destino (familia)

Sin importar cuál de los dos resultados deba considerarse correcto o mejor la realidad es que existe un cambio de comportamiento que puede afectar el resultado de consultas preexistentes.

Otro Ejemplo

Suponga que nos piden un listado de todos los empleados con el nombre de su conyugue en caso de tenerlo.

Utilizando la sintaxis clásica escribimos:

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO, FAMILIA WHERE CODIGO *= CODEMP AND RELACION = 'CONYUGUE'
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
2	Ernesto Alvarez	NULL	NULL	NULL

3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
6	Ines Moreno	NULL	NULL	NULL
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52
8	German Bringas	NULL	NULL	NULL
9	Andrea Pizzini	NULL	NULL	NULL

Utilizando la sintaxis ANSI debemos escribir

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO LEFT OUTER JOIN FAMILIA ON (CODIGO = CODEMP) WHERE RELACION = 'CONYUGUE'
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52

Al realizar el Join primero, y luego filtrar por la relación, la consulta con la sintaxis ANSI no nos está mostrando los datos que cumplen con la consigna. Si bien en el caso anterior la sintaxis ANSI parecía traer el resultado correcto, en este ejemplo la sintaxis clásica parece ser la más adecuada.

El problema se resuelve entendiendo en cada caso que es lo que se debe realizar primero, el Filtro o el Join.

Dado que la sintaxis clásica ya no está disponible en SQL Server 2012, solo queda la sintaxis ANSI para resolver el problema

Para realizar el filtro ANTES del Join, se debe mover la condición de la cláusula WHERE dentro de la cláusula ON.

Esto se escribe así

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO LEFT OUTER JOIN FAMILIA ON (CODIGO = CODEMP AND RELACION = 'CONYUGUE')
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
2	Ernesto Alvarez	NULL	NULL	NULL
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
6	Ines Moreno	NULL	NULL	NULL

7	Carman Lopez	Carlos Ibañez	CONYUGUE	52
8	German Bringas	NULL	NULL	NULL
9	Andrea Pizzini	NULL	NULL	NULL

De esta forma se le indica al motor de base de datos realizar el filtrado ANTES de realizar el Join

3.4 Resumen

Si necesitamos realizar PRIMERO Join y SEGUNDO Filtrar el resultado debemos escribir la condición del filtro dentro de la cláusula WHERE, porque el WHERE siempre se ejecuta DESPUES de realizar el Join

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO LEFT OUTER JOIN FAMILIA ON (CODIGO = CODEMP) WHERE RELACION = 'CONYUGUE'
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52

Si necesitamos realizar PRIMERO el Filtrado y SEGUNDO el Join con el resultado del filtro ya aplicado, debemos escribir la condición del filtro dentro de la cláusula ON () del OUTER JOIN, porque de esta forma el filtro siempre se ejecuta ANTES de realizar el Join

```
SELECT CODIGO, NOMBRE, FAMILIAR, RELACION, EDAD FROM EMPLEADO LEFT OUTER JOIN FAMILIA ON (CODIGO = CODEMP AND RELACION = 'CONYUGUE')
```

CODIGO	NOMBRE	FAMILIAR	RELACION	EDAD
1	Juan Perez	Silvia Gonzalez	CONYUGUE	41
2	Ernesto Alvarez	NULL	NULL	NULL
3	Francisco Gonzalez	Andrea Lopez	CONYUGUE	32
4	Nahuel Testa	Sandra Romero	CONYUGUE	37
5	Andrea Navarro	Miguel Fels	CONYUGUE	34
6	Ines Moreno	NULL	NULL	NULL
7	Carman Lopez	Carlos Ibañez	CONYUGUE	52
8	German Bringas	NULL	NULL	NULL
9	Andrea Pizzini	NULL	NULL	NULL

Entonces cuando utilizamos la sintaxis ANSI (LEFT OUTER JOIN):

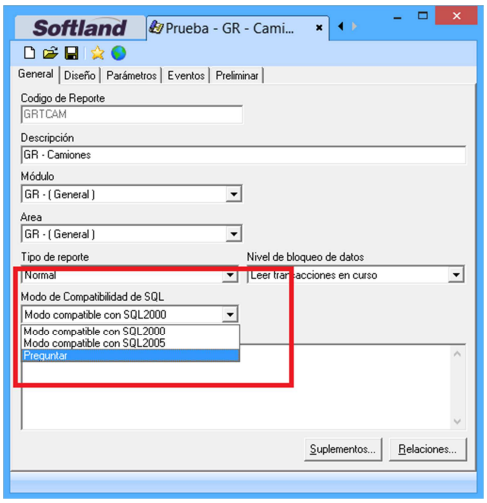
- Cuando se necesita realizar el Filtrado de Datos Después de realizar el Join se debe establecer la condición del filtro dentro de la cláusula WHERE

- Cuando se necesita realizar el Filtrado de Datos Antes de realizar el Join se debe establecer la condición del filtro dentro de la cláusula ON ()

4. MÓDULO DE REPORTES

4.1 Modo de Compatibilidad

El módulo de Reportes (RM) genera automáticamente consultas de SQL para resolver la recuperación de los datos de la base según la definición del reporte.



El módulo puede generar las sentencias utilizando la cláusula de Join Clásica (*=) o la Cláusula de Join ANSI (LEFT OUTER JOIN). A partir de la versión 2.3.9.16 en caso de detectar que la base de datos se encuentra en modalidad compatible con SQL Server 2000 (esto es requerido hasta la versión 2.3.9.15) el editor de definición de reportes permite definir el modo de compatibilidad del reporte, esto significa que las consultas de SQL que el reporte generará serán compatibles con el modo de compatibilidad de la base de datos SQL Server 2000 o modo compatibilidad SQL Server 2005 en adelante. Este comportamiento puede definirse en la sección "General" del editor de reportes, según se muestra en la figura a continuación.

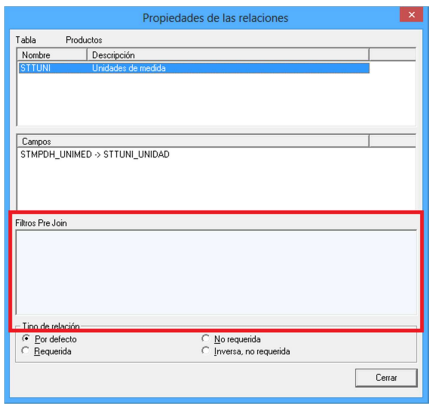
En caso de se selecciona el "Modo compatible con SQL2000" RM generará sentencias con sintaxis clásica (*=).

En caso de se selecciona el "Modo compatible con SQL2005" RM generará sentencias con sintaxis ANSI (LEFT OUTER JOIN).

En caso de se selecciona el "Preguntar" RM generará un parámetro de tipo booleano que permite establecer en el momento de ejecutar el reporte el modo de funcionamiento. Esto está destinado al proceso de validación del reporte para facilitarle al usuario verificar el funcionamiento de un modo y otro sin entrar en el diseño del reporte.

4.2 Filtrar antes de realizar el Join

Cuando se utiliza el modo de compatibilidad con SQL Server 2005 todos las condiciones definidos en la sección Filtros se aplican en la cláusula WHERE, por lo que se filtran los datos luego de haber realizado el Join. En caso de requerir que los datos sean filtrados antes de realizar el Join en la ventana "Propiedades de las relaciones" del editor de reportes puede establecer las condiciones de filtrado que deben establecerse ANTES de realizar el Join. El efecto sobre el conjunto de datos es similar a la sintaxis clásica (*=) con el modo "Compatibilidad SQL Server 2000" activo.



4.3 Prueba y Validación de la instalación

Es recomendable antes de realizar la actualización a SQL Server 2012, activar el módulo de compatibilidad con SQL Server 2012 y modificar la propiedad “Modo de compatibilidad de SQL” de los reportes críticos de la instalación estableciendo su valor a “Modo compatible con SQL 2005” para que comiencen generar sentencias compatibles con las soportadas por SQL Server 2012. Esta tarea puede requerir modificar uno o más reportes para adecuarlos al modo de funcionamiento del SQL Server 2012.

Valide el correcto funcionamiento en el entorno de pruebas y solo cuando esté completamente seguro que funciona correctamente, aplique el mismo cambio en su entorno de producción.

No migre la versión de la base de datos a SQL Server hasta asegurarse que ha revisado todos los reportes que son críticos para su instalación.