http://www.teknodatips.com.ar

Teknoda - Notas técnicas – Tips de AS400 – iSeries - System i

Tip Nro. 45

(Lo nuevo, lo escondido, o simplemente lo de siempre pero bien explicado)

---

*"Notas técnicas de AS/400, iSeries System i" se envía con frecuencia variable y absolutamente **sin cargo** como un servicio a nuestros clientes AS/400. Contiene notas/recursos/artículos técnicos desarrollados en forma totalmente objetiva e independiente. Teknoda es una organización de servicios de tecnología informática y **NO comercializa hardware, software ni otros productos**. Conteste este mail con asunto "REMOVER" si no desea recibir más esta publicación. Si desea suscribir otra dirección de e-mail para que comience a recibir los "Tips", envíe un mensaje **desde esa dirección** a letter400@teknoda.com, aclarando nombre, empresa, cargo y país del suscriptor.*

Triggers "Instead Of": Triggers sobre Vistas SQL

Tema:	Base de Datos, DB2 UDB for i5/OS
Utilidad:	Conocer por medio de un ejemplo los tipos de **Triggers "Instead Of"** como nueva funcionalidad del DB2 UDB, definidos sobre **Vistas SQL multi-tablas**.
Nivel:	Intermedio.
Versión:	V5R3 o superior

Lista de Tips publicados hasta la fecha:

Resumen ejecutivo e Introducción

Uno de los tópicos del sistema operativo IBM i (conocido anteriormente como i5/OS o simplemente OS/400), que más funcionalidad ha incorporado en los últimos años es el gestor de Base de Datos **DB2 UDB**.

Dentro de las mejoras de funcionalidad más importantes en la V5R1, figuran los Triggers SQL, que se agregaron a los ya definidos Triggers Externos (Triggers del Sistema), incorporados al DB2 en la V3R1.

Un **Trigger SQL** es un programa que ejecuta el DB2 **antes** o **después** de que ocurra una operación de Insert, Update o Delete sobre la **TABLA** donde el trigger está definido. El DB2 invoca al trigger que se "dispara" automáticamente realizando una acción determinada, sin tener en cuenta qué aplicación o qué interfase se utilizó para modificar esa tabla.

Los Triggers tanto SQL como Externos, se definen **sólo** sobre TABLAS. La posibilidad de definirlos sobre VISTAS, es ofrecida por los **Triggers "Instead Of**", introducidos por IBM con algunas restricciones en la V5R3 (por medio de algunas PTFs especiales), y liberados con mejoras en V5R4.

Un **Trigger Instead Of** es un tipo especial de trigger, escrito en **lenguaje SQL** que le dice al DB2 qué código específico ejecutar cuando se realiza una operación de Insert, Delete o Update sobre una **VISTA SQL, "en vez de"** permitir al gestor realizar la modificación "él mismo" (lo que muchas veces sería imposible de realizar).

El soporte de **Triggers "Instead Of"** se introdujo en la **V5R3** por medio de una PTF especial y sólo se podía definir sobre una **Vista SQL** que accediera a **una sóla tabla**.

En **V5R4** IBM liberó el soporte completo mejorando este tipo de triggers, pudiendo además realizar operaciones de manipulación de datos con **vistas SQL multi-tablas**, es decir, aquellas que involucran Joins o Unions.

En el desarrollo del presente Tip detallaremos el concepto de **Triggers "Instead Of"** y un ejemplo para entender su funcionamiento.

Qué son los Triggers "Instead Of"?

En determinadas ocasiones, usar sentencias SQL para realizar operaciones simultáneas de Insert, Delete y Updates contra tablas relacionadas, puede ser una tarea engorrosa. Sería muy conveniente que se pudiera tratar de manera simple con las tablas relacionadas, como si fuera una única tabla con el fin de modificar sus datos.

Si bien es relativamente fácil recuperar y modificar datos de tablas usando RPG con subfiles, no es fácil realizar esta tarea con SQL, especialmente en una aplicación cliente/servidor.

Usualmente, hacer ésto podría requerir múltiples sentencias insert, delete y update para cada tabla en una vista múltiple.

Permitir modificaciones sobre tablas combinadas (joined) es sólo un ejemplo de lo "imposible" de realizar, a través de los triggers **Instead Of**.

Además de usarlos sobre vistas que combinan tablas con un JOIN, los **Triggers INSTEAD Of** pueden ser definidos sobre **vistas** que combinan tablas "verticalmente" usando las sentencias UNION o UNION ALL, permitiendo también "convertirlas" en vistas modificables.

Los **"Triggers Instead of"** son triggers escritos en lenguaje SQL que dan la posibilidad al desarrollador de controlar cómo se modifican los datos en una Vista. El código definido como lógica del trigger Instead Of es procesado "instead of" ("en lugar de") la sentencia SQL Update, Delete o Insert que se haya especificado realizar.

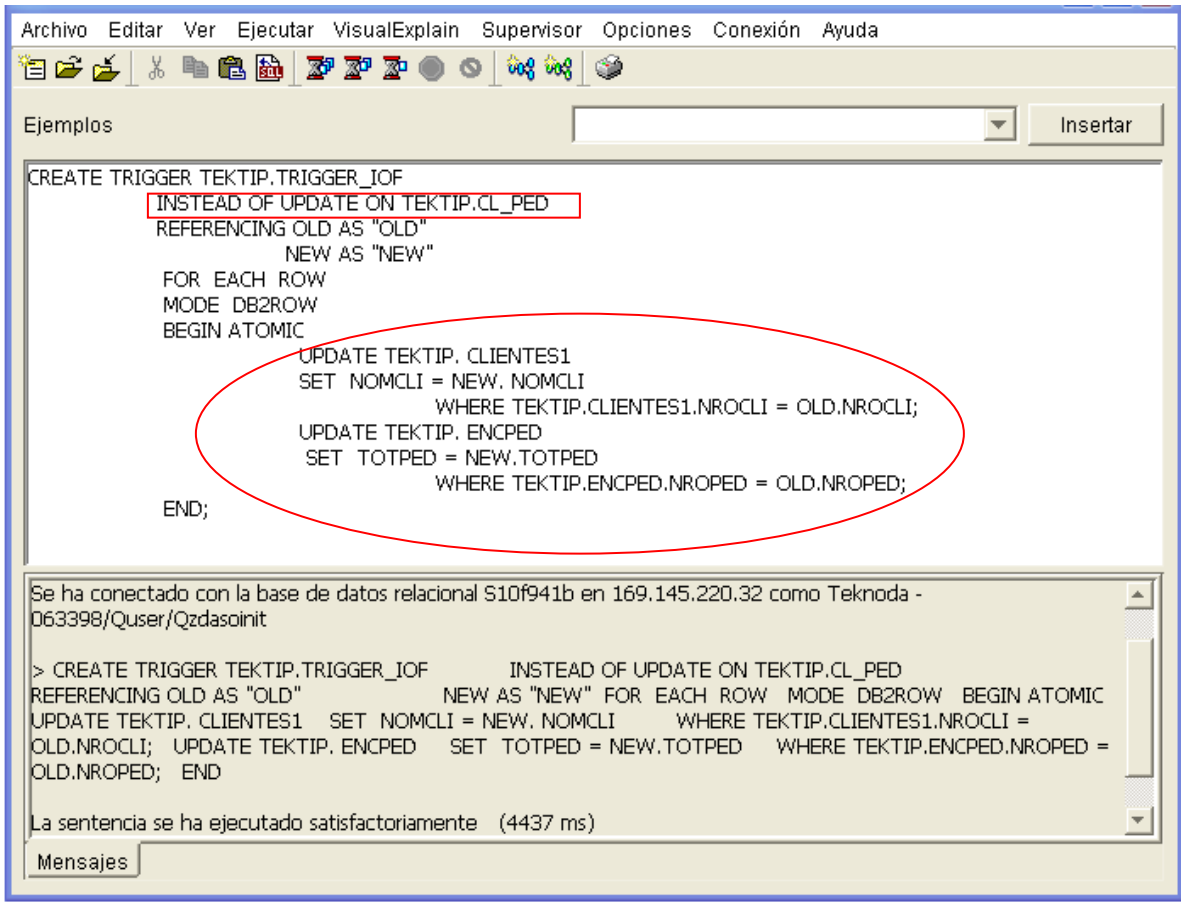
Tradicionalmente, las vistas basadas en una tabla única eran las únicas vistas "modificables", y actualizar una vista, era por lo tanto lo mismo que actualizar la tabla subyacente.

Sin embargo, usando ahora los **triggers Instead Of**, se puede agregar lógica que permita ejecutar operaciones inserts, deletes o updates a **cualquier clase de Vista SQL**

Un ejemplo

Se tiene ya creada una **Vista SQL** que accede a dos tablas: CLIENTES1 y ENCPED. Al crear un trigger INSTEAD OF sobre el evento Update, la vista **CL_PED** pasará a ser **modificable**.

Todo lo que se necesita suministrar en el **código del trigger Instead Of** es la lógica apropiada para modificar las tablas subyacentes como se muestra en el ejemplo de más abajo:



El Script SQL descripto en la figura de arriba, creó un Trigger Instead Of sobre la vista CL_PED de la biblioteca TEKTIP. (*Ver **Tip Nro. 42**: Creación y ejecución de un script SQL desde Windows usando iSeries Navigator*, para más información sobre cómo usar la componente Run SQL Script del iSeries Navigator)

Donde:

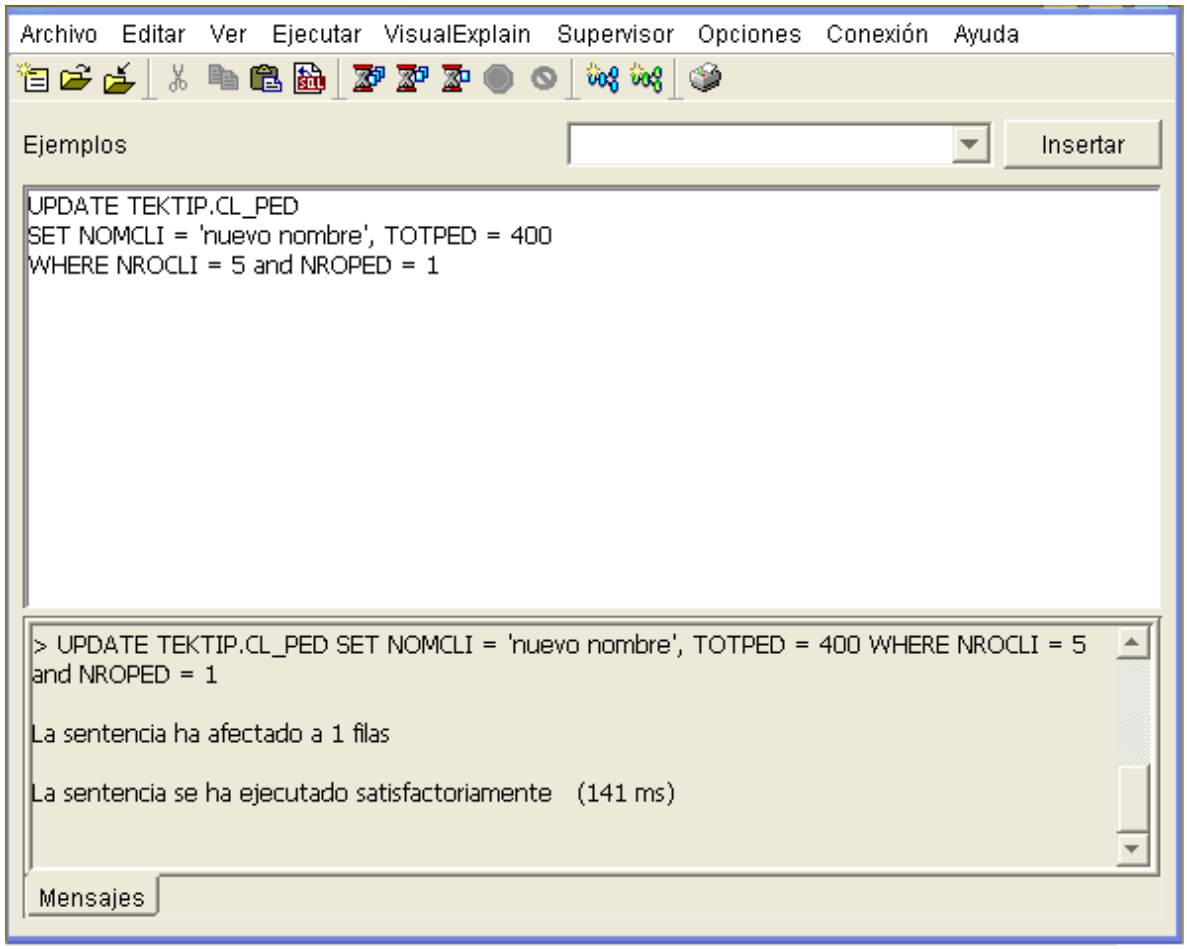
- **TRIGGER_IOF** es el nombre del trigger creado. Si se desea visualizar el objeto creado desde la interfaz tradicional, se lo verá como un objeto de tipo ***PGM CLE**, en la biblioteca TEKTIP. Como el nombre supera los 10 caracteres, se crea con nombre: TRIGGER00001.

- **INSTEAD OF UPDATE ON** TEKTIP.CL_PED, se refiere al **tipo de trigger que se crea,** y especifica que **"en lugar de"** realizar un UPDATE sobre la vista, ejecute el trigger con la lógica especificada.

- **CL_PED** es el nombre de la vista creada en la biblioteca TEKTIP, sobre la cuál se define el trigger Instead Of**.** En este caso es una vista multi-tablas que combina dos tablas: **CLIENTES1** y **ENCPED** de la biblioteca TEKTIP.

- El nombre especificado en las porciones **OLD** y **NEW** (en este caso se llaman OLD y NEW) de la cláusula **REFERENCING** de la sentencia CREATE TRIGGER, se usa para referenciar a las columnas en la fila que está siendo modificada, y representan el contenido anterior (OLD) de un campo y el nuevo contenido (NEW). Se las llama Variables de Transición.

Luego de haber creado el Trigger Instead OF asociado a la vista SQL CL_PED del ejemplo, si se ejecuta **sobre la vista CL_PED** la sentencia SQL especificada en el siguiente script, se realizará el cambio en el o los registros involucrados de **las tablas asociadas con esa vista SQL**, con los valores indicados en la sentencia:



Cuando una sentencia UPDATE sea realizada sobre la vista CL_PED, se ejecutará el código especificado en el trigger Instead Of y actualizará la fila (en este caso es sólo una) involucrada en ambas tablas.

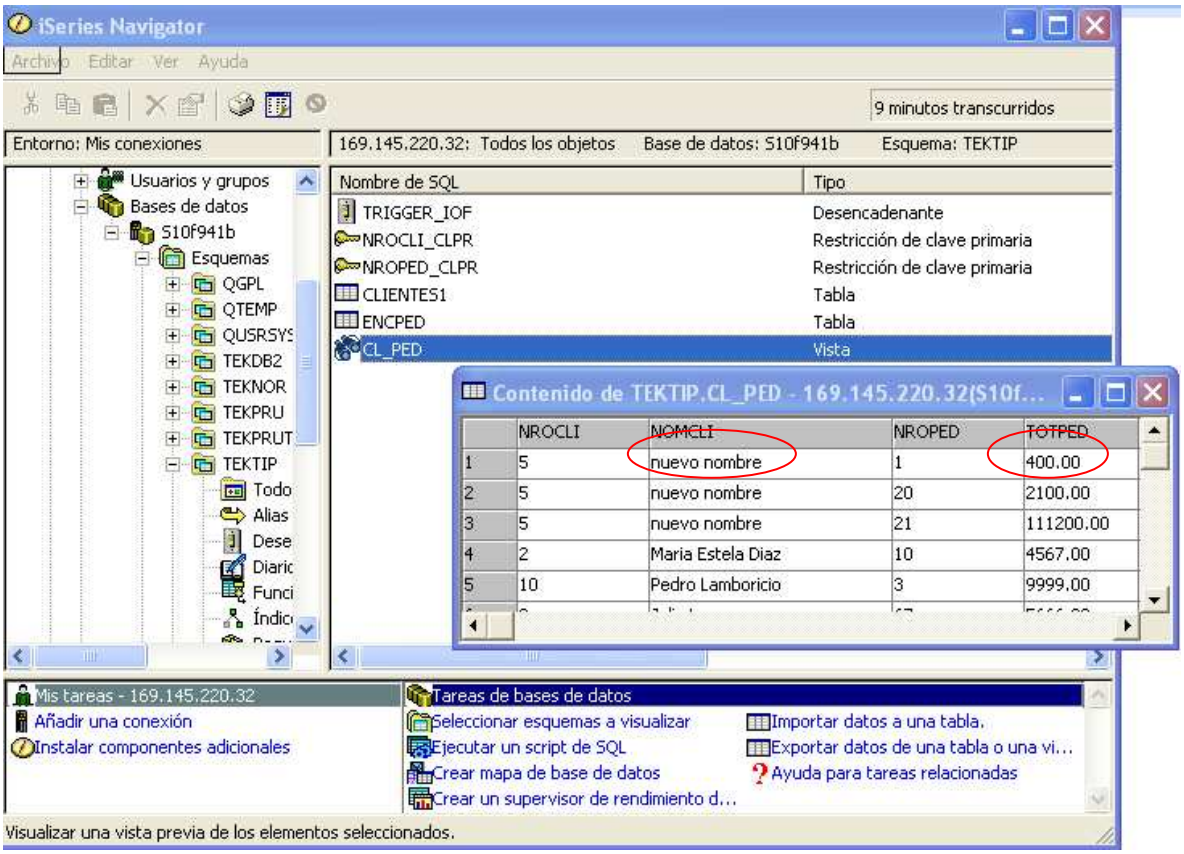
Si bien una sentencia UPDATE normalmente fallaría al realizarse sobre esta vista, en este caso es exitosa debido al código alternativo provisto por el DB2 a través del trigger definido. Más adelante se detalla el error emitido en caso de no tener definido este tipo de trigger sobre una vista que sea sólo de lectura.

La figura de abajo muestra cómo ver el contenido de una vista **desde la interfaz gráfica iSeries Navigator**, es decir, muestra los registros de ambas tablas que satisfacen la definición de la vista.

Se puede observar que después se haber realizado la operación de Update anterior sobre **la vista SQL CL_PED**, si luego se elige **Ver contenido** de la misma, se ven los cambios en la fila afectada (podría haber involucrado más de una fila). (Seleccionar **Ver contenido** sobre la Vista SQL usando la interfaz iSeries Navigator, es como realizar un SELECT sobre la vista)

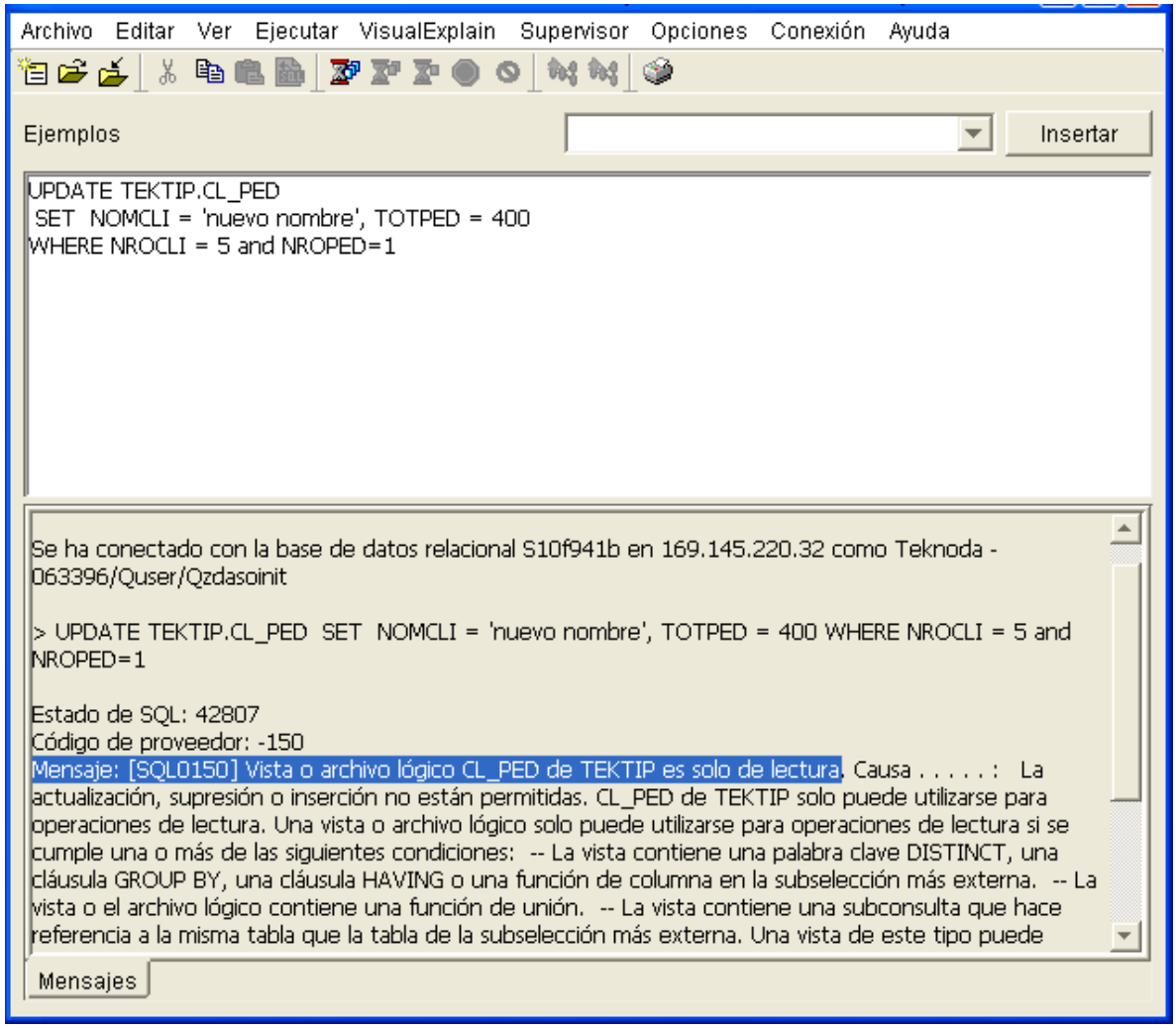
Si además, luego de realizar la operación anterior se visualizaran las tablas involucradas CLIENTES1 y ENCPED se podrán comprobar que los cambios se realizaron efectivamente en cada una de ellas, como si se hubieran establecido directamente sobre las mismas, y no sobre la vista.

En el caso de la tabla CLIENTES1, el campo NOMCLI tendrá el valor '*nuevo nombre'* y en la tabla ENCPED, el campo TOTPED (sólo para el NROPED = 1) tendrá el valor *400* , según lo indicado en la sentencia UPDATE, para los demás nros. de pedido el valor total no cambió.



Si la vista SQL CL_PED no tuviese definido el **trigger Instead Of** , la sentencia UPDATE ejecutada sobre la vista CL_PED usada en el ejemplo, **fallaría** indicando el error como se muestra en la pantalla siguiente.

Es decir, la sentencia UPDATE fallaría al realizarse sobre esta vista, debido a que **no existe el código alternativo provisto por el DB2 a través de un trigger Instead Of**, y por lo tanto es una vista sólo de lectura.



Conclusión

A través del ejemplo detallado en este tip se logró **hacer modificable** una vista SQL, por medio de la definición de un **trigger Instead Of** . Una vista que en principio permitía operaciones sólo de lectura sobre las tablas involucradas, pudo aceptar operaciones que modifiquen sus registros.

Si quisiéramos hacer que la vista SQL CL_PED tomada como ejemplo en este tip, pase a ser totalmente modificable, además del código definido para el trigger Instead OF **Update,** se debería agregar:

- un trigger INSTEAD OF Insert y
- un trigger INSTEAD OF Delete

Restricciones y limitaciones

- Un trigger INSTEAD OF debe ser definido sobre una Vista SQL, y la vista no puede estar distribuida en múltiples sistemas.
- NO se puede definir un trigger INSTEAD OF sobre un **archivo lógico** nativo (creado a partir de un fuente escrito en lenguaje DDS).
- Para una vista dada, puede ser definido un **sólo** trigger INSTEAD OF para cada operación (INSERT, UPDATE, y DELETE). Por lo tanto, una vista tendrá un máximo de tres triggers.

Para tener en cuenta:

Requerimientos de PTFs para V5R3:

- Para habilitar el soporte de los triggers INSTEAD OF, tienen que ser aplicadas en el sistema las siguientes PTFs:

  - PTF de Grupo de Base de Datos de V5R3 – Nivel 4 o mayor
  - SI17399
  - SI17434

- Una vista puede ser usada para controlar el acceso a tablas. Al usar triggers INSTEAD OF el mantenimiento del control de acceso a tablas puede simplificarse.
- Se puede visualizar la información de un trigger Instead Of definido sobre una vista con el comando Display File Description (DSPFD) y especificando en el parámetro **TYPE** el valor *TRG.
- Puede también visualizarse la información de un Trigger Instead OF, desde la interfaz gráfica iSeries Navigator, componente **Base de Datos.**
- Para obtener información de cualquier trigger INSTEAD OF definido sobre una Vista SQL, se puede realizar un query sobre la vista del catálogo SYSTRIGGER en la biblioteca QSYS2.
- La vista del catálogo SYSTRIGGER ubicado en la biblioteca QSYS2, tendrá un nuevo valor **INSTEAD** en la columna **ACTION_TIMING** para un trigger INSTEAD OF.
- Se debe contar con la autorización ALTERAR sobre el objeto para poder definir un trigger de cualquier tipo, ya sean sobre TABLAS o sobre VISTAS. .

- En V6R1 (IBM i 6.1), mejoró el soporte de Triggers, al permitir a los triggers INSTEAD OF poder ser ejecutados ***for each statement***

Dudas o consultas a : nsalmun@teknoda.com