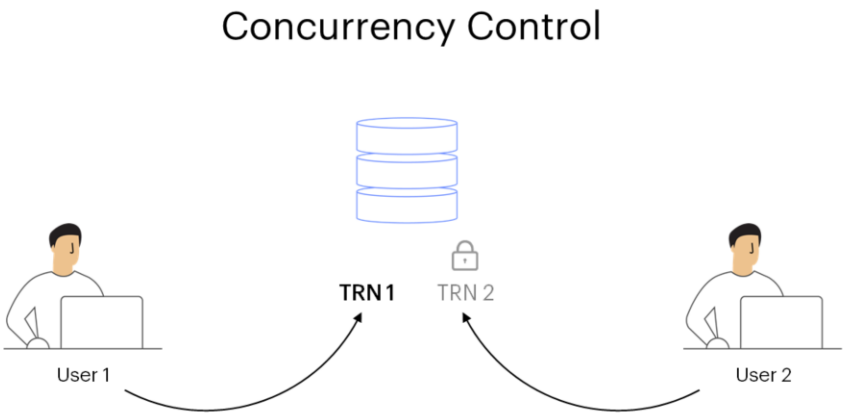
Control de concurrencia

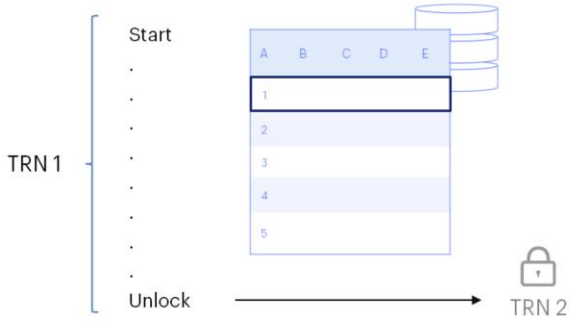
GeneXus™



Cuando hablamos de “Control de concurrencia” nos referimos a un conjunto de controles para evitar posibles inconsistencias en los datos cuando se trabaja en ambientes con múltiples usuarios.

Para controlar la concurrencia es necesario bloquear la información.

Concurrency Control



Además, La ejecución concurrente de los programas no debe causar una mala performance.

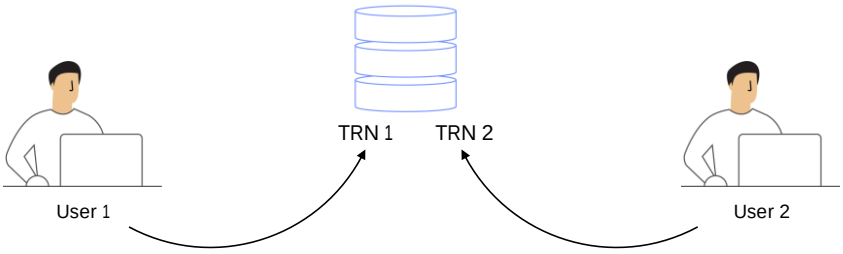
Concurrency Control Optimistic

Transactions

Business Components

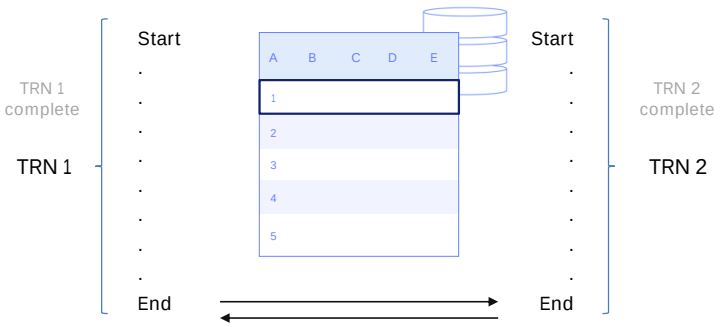
Comencemos entonces viendo cómo es el control de concurrencia en Transacciones y Business Components. Hablemos entonces del "Control de concurrencia optimista"

Optimistic Concurrency Control



Este mecanismo asume que múltiples transacciones de base de datos se pueden completar sin afectarse unas a otras, y que por lo tanto estas transacciones pueden realizarse sin bloquear los recursos de datos que involucran.

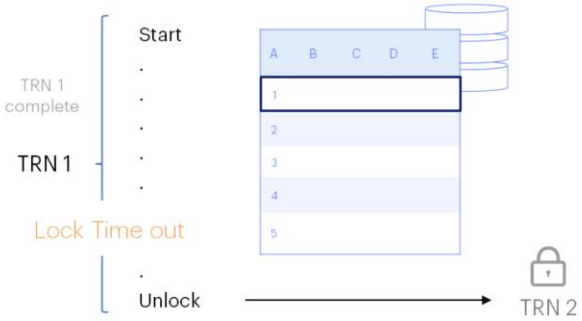
Optimistic Concurrency Control



Antes de realizar una actualización, cada transacción de base de datos controla que ninguna otra haya modificado sus datos.

Esto tiene una relación con la naturaleza globalizada de la web donde varios usuarios pueden acceder a la vez a una misma página. Esto hace que el bloqueo sea inviable para las interfaces de usuario web.

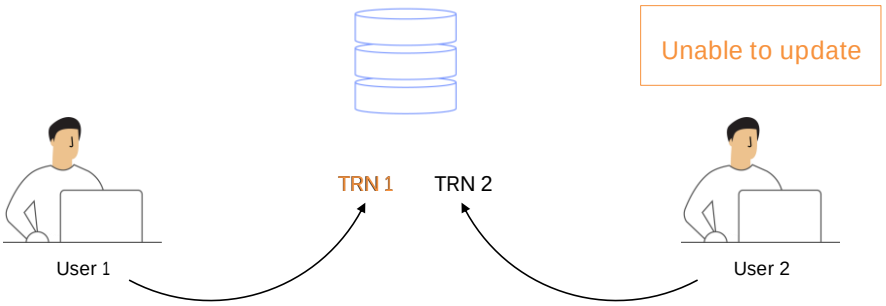
Optimistic Concurrency Control



Es común que un usuario comience a editar un registro y luego se vaya sin seguir el enlace de "cancelar" o de "cerrar la sesión".

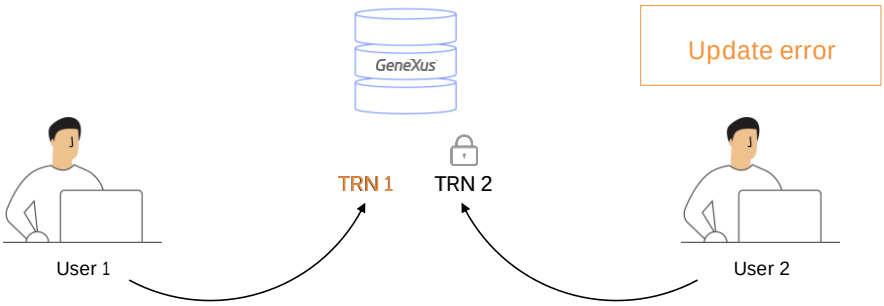
Si se usa el bloqueo, otros usuarios que intenten editar el mismo registro deben esperar hasta que se agote el tiempo de bloqueo del primer usuario.

Optimistic Concurrency Control



En lugar de bloquear cada registro cada vez que se usa, el sistema simplemente busca indicaciones de que dos usuarios realmente intentaron actualizar el mismo registro al mismo tiempo. Si se encuentra esa evidencia, entonces las actualizaciones de un usuario se descartan y se le informa de esto.

Optimistic Concurrency Control

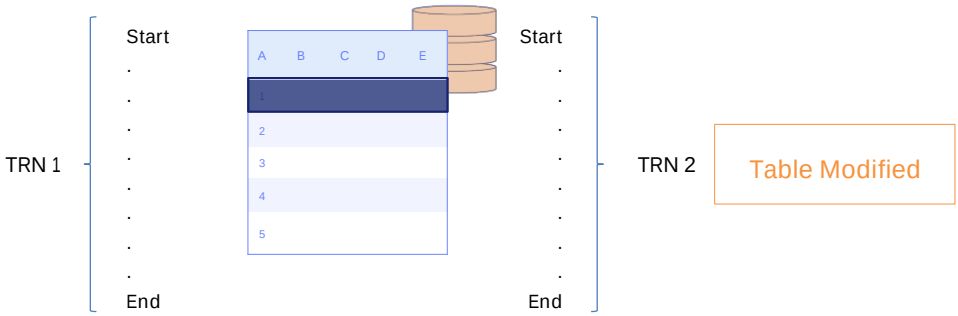


En el contexto de GeneXus, cuando dos o más usuarios quieran actualizar un mismo registro, el primero podrá actualizarlo y los demás obtendrán un error (porque estaban a punto de guardar nuevos datos en base a información desactualizada actualizada por otro usuario).

Entonces, en estos casos, la operación de confirmación se revierte.

Optimistic Concurrency Control

Old Function



IMPORTANT
Only the present attributes in the transaction structure will be verified.

Does not apply to Attributes:

El control de concurrencia optimista se basa en la "Función Old".

Cuando se confirma un registro, los valores "antiguos" de cada uno de los atributos se comparan con los valores actuales de la base de datos. Si un valor no coincide, se muestra un error del tipo: "La tabla fue modificada", lo que significa que otro usuario modificó el registro desde el momento en que obtuvo los valores.

Es importante mencionar que:

- Solo se tienen en cuenta los atributos presentes en la estructura de la transacción. En otras palabras, si hay por ejemplo transacciones paralelas con más atributos, no se verifican.
- Este mecanismo no aplica a atributos de tipo imagen, video, audio, Longvarchar, ni tampoco a aquellos atributos inferidos de la tabla extendida que se actualizan mediante reglas declaradas en la transacción.
- Solo aplica a atributos que no son parte de la clave, porque con ellos se instancia el registro.

Concurrency Control in GeneXus Objects

when Reading Data

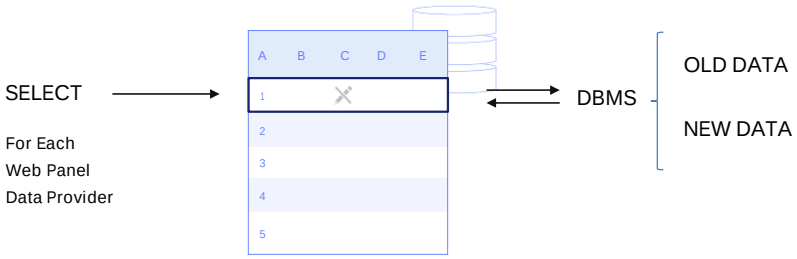
when Reading & Writing Data

Veamos ahora cómo es el control de concurrencia en otros objetos GeneXus.

Veamos cómo GeneXus maneja los bloqueos cuando solo lee datos y cuando lee y escribe datos para controlar la concurrencia.

Concurrency Control in GeneXus Objects

Read Only



En GeneXus:
Propiedad Isolation level

- Read Committed
- Read Uncommitted

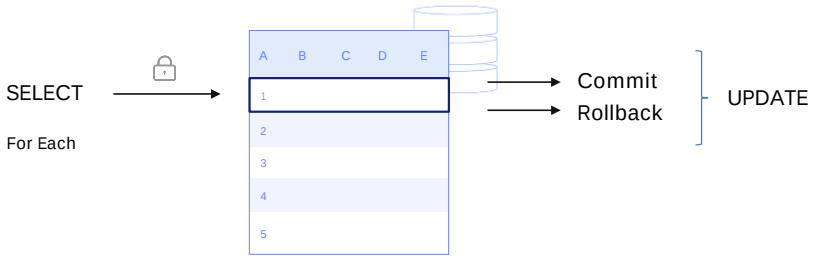
¿Qué sucede cuando solamente se leen datos?

Si es solo para lectura (cuando se usan comandos For Each, objetos de tipo Web Panel, Data Providers, etc.), los SELECT generados NO se bloquean.
Siempre se ven afectados por bloqueos exclusivos. Por ejemplo, en las reorganizaciones las tablas se abren en una forma exclusiva. En este caso, ningún otro proceso podrá abrir la tabla, no importa si es solo para lectura.

Por otro lado, si la información a leer está bloqueada por otro programa de escritura, los valores que se mostrarán dependerán del DBMS. Es el DBMS entonces quien decidirá si mostrar el valor antiguo o el nuevo.

Concurrency Control in GeneXus Objects

Read & Write



IMPORTANT
Certain For Eachs are optimized and won't lock down. They directly update.

¿Y qué sucede cuando se leen y escriben datos?

Un comando For Each que incluye una actualización a la base de datos, realiza un SELECT con un candado al ingresar al For Each y luego, se realiza la actualización.

Como cualquier bloqueo, libera los registros cuando ejecuta el Commit o Rollback

Es importante destacar que ciertos comandos For Each están optimizados y no bloquean, pero actualizan directamente.
Solamente son optimizables los comandos For each que contienen condiciones que van al servidor, y solo asignaciones en su cuerpo.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications