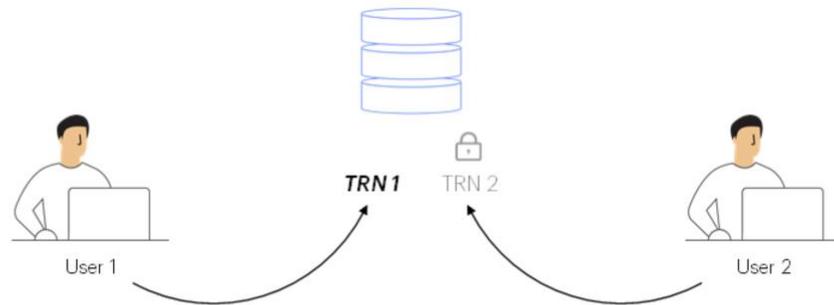


Controle de concorrência

GeneXus™

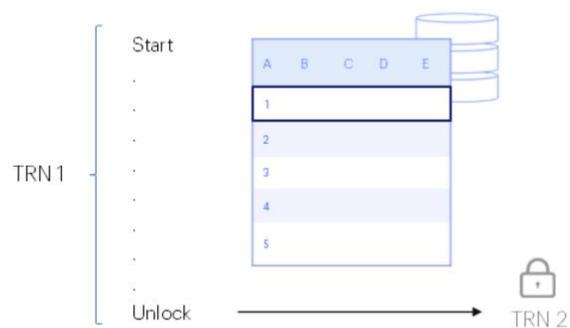
Concurrency Control



Quando falamos de “Controle de concorrência”, nos referimos a um conjunto de controles para evitar possíveis inconsistências nos dados quando se trabalha em ambientes com múltiplos usuários.

Para controlar a concorrência é necessário bloquear a informação.

Concurrency Control



Além disso, a execução concorrente dos programas não deve causar mau desempenho.

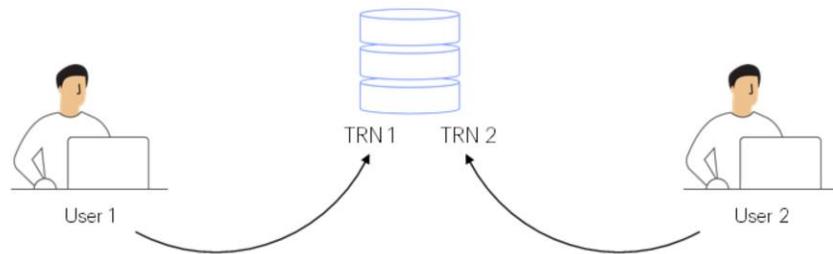
Concurrency Control Optimistic

Transactions

Business Components

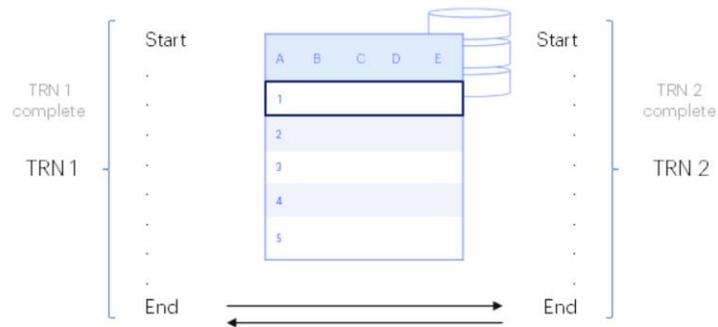
Comecemos então vendo como é o controle de concorrência em Transações e Business Components. Vamos falar então do “Controle de concorrência otimista”

Optimistic Concurrency Control



Este mecanismo assume que múltiplas transações de base de dados podem ser concluídas sem afetar umas às outras e que, portanto, estas transações podem ser realizadas sem bloquear os recursos de dados que envolvem.

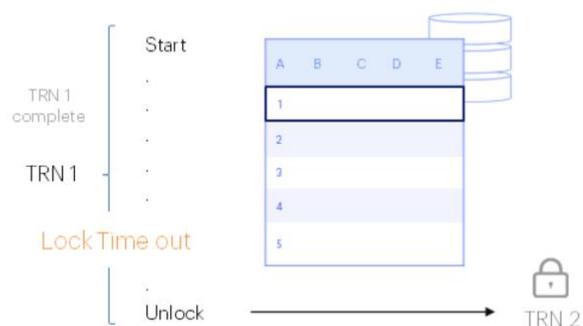
Optimistic Concurrency Control



Antes de realizar uma atualização, cada transação de base de dados controla que nenhuma outra tenha modificado seus dados.

Isto tem uma relação com a natureza globalizada da web, onde vários usuários podem acessar ao mesmo tempo uma mesma página. Isto faz com que o bloqueio seja inviável para interfaces de usuário web.

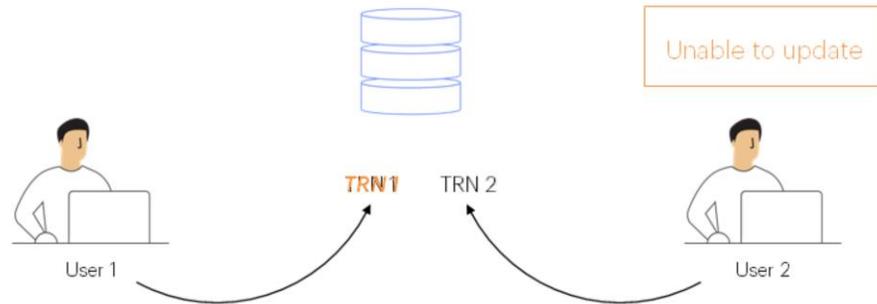
Optimistic Concurrency Control



É comum que um usuário comece a editar um registro e depois saia sem seguir o link de "cancelar" ou de "fechar a sessão".

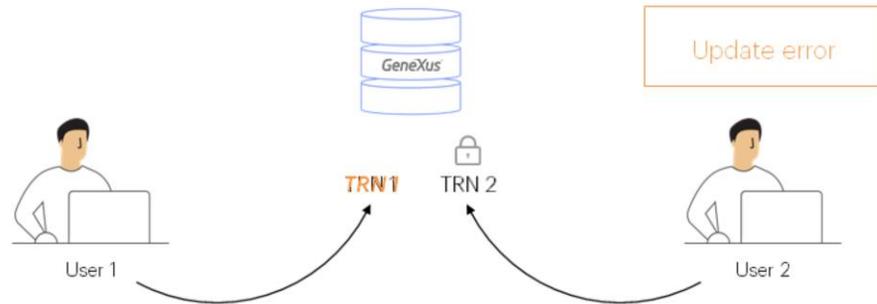
Se é usado o bloqueio, outros usuários que tentem editar o mesmo registro devem esperar até que se esgote o tempo de bloqueio do primeiro usuário.

Optimistic Concurrency Control



Em vez de bloquear cada registro cada vez que é usado, o sistema simplesmente procura indicações de que dois usuários realmente tentaram atualizar o mesmo registro ao mesmo tempo. Se essa evidência for encontrada, então as atualizações de um usuário são descartadas e ele é informado disso.

Optimistic Concurrency Control

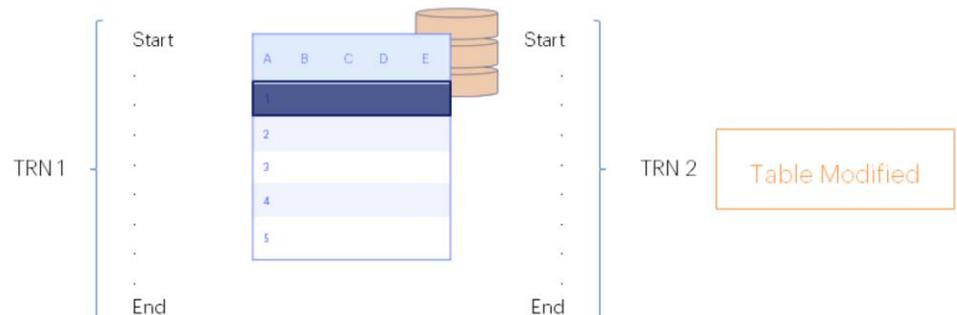


No contexto de GeneXus, quando dois ou mais usuários desejam atualizar um mesmo registro, o primeiro poderá atualizá-lo e os demais obterão um erro (porque estavam prestes a salvar novos dados com base em informações desatualizadas, atualizadas por outro usuário).

Portanto, nestes casos, a operação de confirmação é revertida.

Optimistic Concurrency Control

Old Function



IMPORTANT

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

Does not apply to Attributes:



O controle de concorrência otimista se baseia na "Função Old".

Quando se confirma um registro, os valores "antigos" de cada um dos atributos são comparados com os valores atuais da base de dados. Se um valor não coincide, é exibido um erro do tipo: "A tabela foi modificada", o que significa que outro usuário modificou o registro desde o momento em que obteve os valores.

É importante mencionar que:

- São considerados apenas os atributos presentes na estrutura da transação. Em outras palavras, se houver, por exemplo, transações paralelas com mais atributos, não são verificados.
- Este mecanismo não se aplica aos atributos do tipo imagem, vídeo, áudio, Longvarchar, nem aos atributos inferidos da tabela estendida que são atualizados por regras declaradas na transação.
- Só se aplica a atributos que não fazem parte da chave, porque com eles o registro é instanciado.

Concurrency Control in GeneXus Objects

when Reading Data

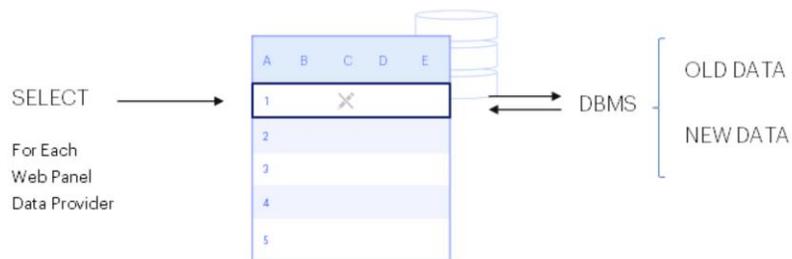
when Reading & Writing Data

Vejamos agora como é o controle de concorrência em outros objetos GeneXus.

Vejamos como GeneXus trata os bloqueios quando só lê dados e quando lê e grava dados para controlar a concorrência.

Concurrency Control in GeneXus Objects

Read Only



Em GeneXus:
Propriedade Isolation level

- Read Committed
- Read Uncommitted



O que acontece quando apenas são lidos os dados?

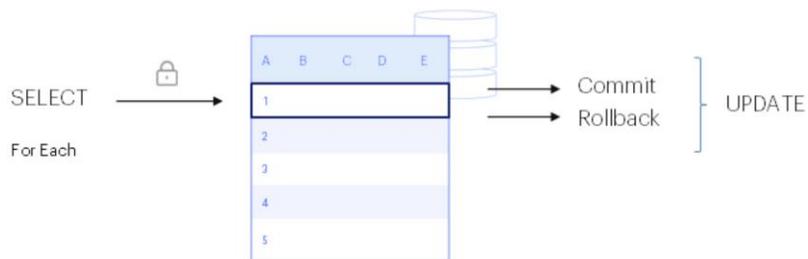
Se for apenas para leitura (quando são usados comandos For Each, objetos do tipo Web Panel, Data Providers, etc.), os SELECTs gerados NÃO são bloqueados.

Sempre são afetados por bloqueios exclusivos. Por exemplo, nas reorganizações, as tabelas são abertas de uma forma exclusiva. Neste caso, nenhum outro processo poderá abrir a tabela, não importa se é apenas para leitura.

Por outro lado, se a informação a ser lida está bloqueada por outro programa de escrita, os valores que serão mostrados dependerão do DBMS. É o DBMS então que decidirá se mostra o valor antigo ou o novo.

Concurrency Control in GeneXus Objects

Read & Write



IMPORTANT

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

E o que acontece quando são lidos e gravados os dados?

Um comando For Each que inclui uma atualização da base de dados, realiza um SELECT com um cadeado ao entrar no For Each e, em seguida, é realizada a atualização.

Como qualquer bloqueio, libera os registros quando executa o Commit ou Rollback

É importante destacar que certos comandos For Each estão otimizados e não bloqueiam, mas atualizam diretamente.

Apenas são otimizáveis os comandos For each que contêm condições que vão ao servidor, e apenas atribuições em seu corpo.

GeneXus™

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