

O que significa LOG, REDO, UNDO e Checkpoint



LOG



- ▶ Registra todas as transações e as modificações de banco de dados feitas por cada transação;
- ▶ Há 3 tipos de LOG usados com maior frequência:
 - ▶ LOG de erros;
 - ▶ LOG de consulta geral;
 - ▶ LOG de consulta lento.
- ▶ Os campos de um registro de atualizações de log se dão por:
 - ▶ Identificador de Transições - T_i
 - ▶ Identificador do item de dado - X_i
 - ▶ Valor antigo e novo valor - v_1 e v_2

LOG



Em um fórum da microsoft, um usuário relatou sobre um servidor SQL possuir alguns técnicos, e precisavam analisar quem realizou login e quais operações foram executadas no banco de dados

A solução para esse problema se encontra no uso do **log de transações**, onde estará gravada cada ação realizada em um ou mais banco de dados e identificado o autor.

UNDO



Em banco de dados, também chamado de NO-REDO, UNDO é usado para reverter ou desfazer, as alterações no banco de dados. Ele é composto por registro de ações de transações que ainda estão em um estado ativo, ou seja, antes de serem confirmadas.

Undo é utilizado para:

- ▶ Reverter transações
- ▶ Reverter transações quando uma instrução ROLLBACK é emitido
- ▶ Recuperar o banco de dados
- ▶ Fornecer consistência de leitura
- ▶ Analisar dados a partir de um ponto anterior no tempo usando o Oracle Flashback Query
- ▶ Recuperar-se de corrupções lógicas usando recursos do Oracle Flashback

UNDO



Suponha que você seja o administrador de banco de dados da sua empresa. Você planejou rotinas para execução no subsistema DB de produção todos os sábados, entre às 22:00h e 4:00h da manhã. Essas execuções incluem tarefas em lote para:

- ▶ Funcionários demitidos
- ▶ Registrar os dias de folga e férias dos funcionários
- ▶ Aumentos salariais
- ▶ Registrar novos funcionários

UNDO



Às 2:00 da manhã, você é alertado sobre erros no trabalho em lotes que registra dias de ausência e férias. Todos os outros trabalhos foram executados sem erros. Você está programado para iniciar o processamento da folha de pagamento. Como os dados de férias e dias de folga não afetam o processamento da folha de pagamento, você permite que o processamento continue conforme planejado.

Neste exemplo, você sabe o intervalo específico de transações a recuperar e o período em que as transações em lote ocorreram. Portanto, você está pronto para usar o Log para gerar UNDO SQL.

O processo geral para executar um procedimento UNDO é o seguinte:

1. Identifique e analise transações com problemas. (Consulte Análise de transações com problemas).

UNDO



2. Gere UNDO SQL:

- a. Defina um filtro para selecionar o script correto.
- b. Especifique os pontos inicial e final da varredura de log.
- c. Especifique atributos para o relatório Backout Integrity e o arquivo de saída SQL. Use o relatório Backout Integrity serve para determinar o impacto da execução de uma operação UNDO ou REDO e pesar os benefícios da aplicação de UNDO SQL versus REDO SQL.
- d. Salve todas as seleções em uma identificação de trabalho.
- e. Gere e execute o script.

3. Execute UNDO SQL.

Você tem a opção de executar as instruções SQL na mesma execução que as gera. Se você optar por não executar o UNDO SQL na mesma execução, deverá criar uma tarefa Apply de alta velocidade para executar o SQL.

REDO



- ▶ É utilizado para **reaplicar** uma **transação** válida após executar um **recovery** da base
- ▶ Omite a transação das operações indesejadas
- ▶ Pode ser usado quando é executado uma alteração indesejada
- ▶ Para resolver um problema envolvendo transações indesejadas:
 1. Utiliza os **LOGS** para definir quais foram as alterações indesejadas
 2. Executa o **UNDO** para recuperar a base de dados até um ponto anterior às operações indesejadas
 3. Executa o **REDO** para reaplicar as alterações após o ponto atual, excluindo as alterações indesejadas

REDO



Continuando o exemplo do caso **UNDO...**

Você descobre que um trabalho em lotes foi executado por engano na hora errada e que o trabalho adicionou impostos desnecessários a todos os registros de faturas da região sul.

Neste exemplo, você sabe o intervalo específico de transações a recuperar e o período em que as transações em lote ocorreram. Como você deseja restaurar os espaços de tabela envolvidos para um estado anterior ao momento em que a tarefa em lote incorreta foi executada, você decide executar um procedimento de recuperação e de REDO.

REDO



Para executar esse procedimento, você deve gerar REDO SQL que exclua as transações com problemas. Em seguida, você recupera os espaços de tabela envolvidos nas transações do problema para um ponto de consistência anterior ao problema. Em seguida, você executa o REDO SQL gerado.

A geração de REDO SQL segue aproximadamente os mesmos procedimentos que a geração de UNDO SQL, com a exceção de que você deve gerar REDO SQL antes de executar uma recuperação.

O processo geral para executar um procedimento REDO é o seguinte:

REDO



1. Identifique e analise transações problemáticas
2. Identifique todos os espaços de tabela envolvidos e, para esses espaços de tabela, localize um ponto de consistência anterior às transações do problema.
3. Coloque os espaços de tabela no status Utility (UT)
4. Gere REDO SQL:
 - a. Defina um filtro para selecionar o trabalho em lotes correto.
 - b. Especifique os pontos inicial e final da varredura de log.
 - c. Especifique o ponto de recuperação.
 - d. Especifique atributos para o relatório Backout Integrity e o arquivo de saída SQL.

REDO



4.
 - e. Salve todas as seleções em uma identificação de trabalho.
 - f. Gere e execute a JCL.
5. Recupere espaços de tabela e índices relacionados para o ponto anterior no tempo.
6. Coloque os espaços de tabela no status de leitura/gravação (read/write, RW).
7. Execute o REDO SQL.

CHECKPOINT



- ▶ O comando **CHECKPOINT** escreve todos os **registros** de dados que estão no **buffer** de escrita no disco rígido
- ▶ Ações seguidas em um processo de checkpoint são:
 1. Suspender temporariamente a execução de transações.
 2. Gravar à força todos os buffers de memória que tenham sido modificados para o disco.
 3. Gravar um registro [checkpoint] no log, forçando a gravação do mesmo no disco.
 4. Retomar as transações em execução.

CHECKPOINT



- ▶ Existem 4 tipos de checkpoint:

1. Automático

Ocorre de intervalos em intervalos.

Ex: `EXEC sp_configure 'recovery interval', 'seconds'`

2. Indireto

Emitido em segundo plano para cumprir um horário de recuperação de destino especificado pelo usuário para um determinado banco de dados.

Ex: `ALTER DATABASE ... SET TARGET_RECOVERY_TIME = target_recovery_time { SECONDS | MINUTES }`

`ALTER DATABASE ... SET TARGET_RECOVERY_TIME = target_recovery_time { SECONDS | MINUTES }`

CHECKPOINT



- ▶ Existem 4 tipos de checkpoint:
- ▶ Manual

Emitido quando você executa um comando Transact-SQL CHECKPOINT.

EX: `CHECKPOINT [checkpoint_duration]`

- ▶ Interno

Emitido por várias operações de servidor, como backup e criação de instantâneo de banco de dados, para garantir que as imagens de disco coincidam com o estado atual do log.