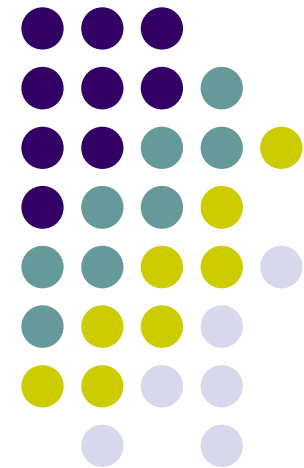
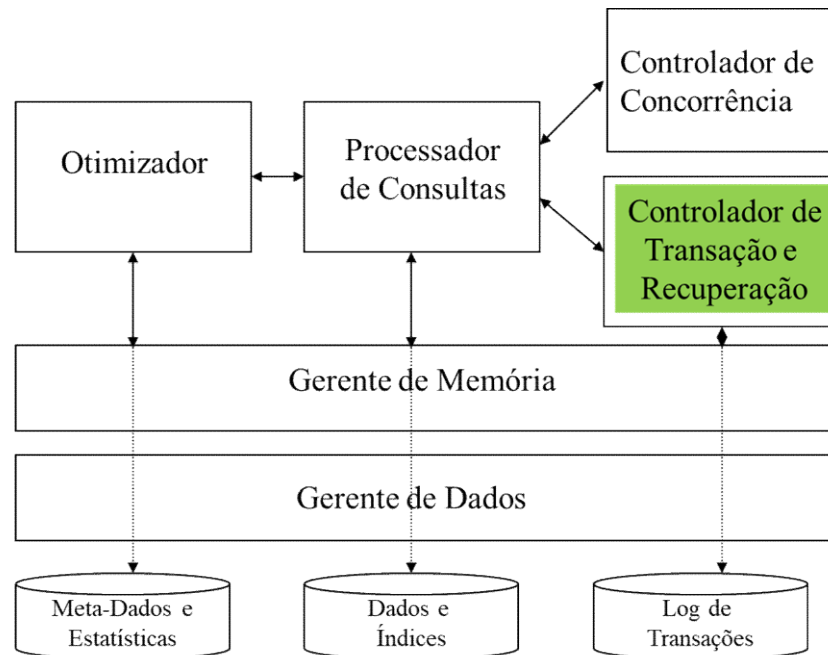
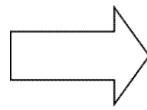


# Bancos de Dados II

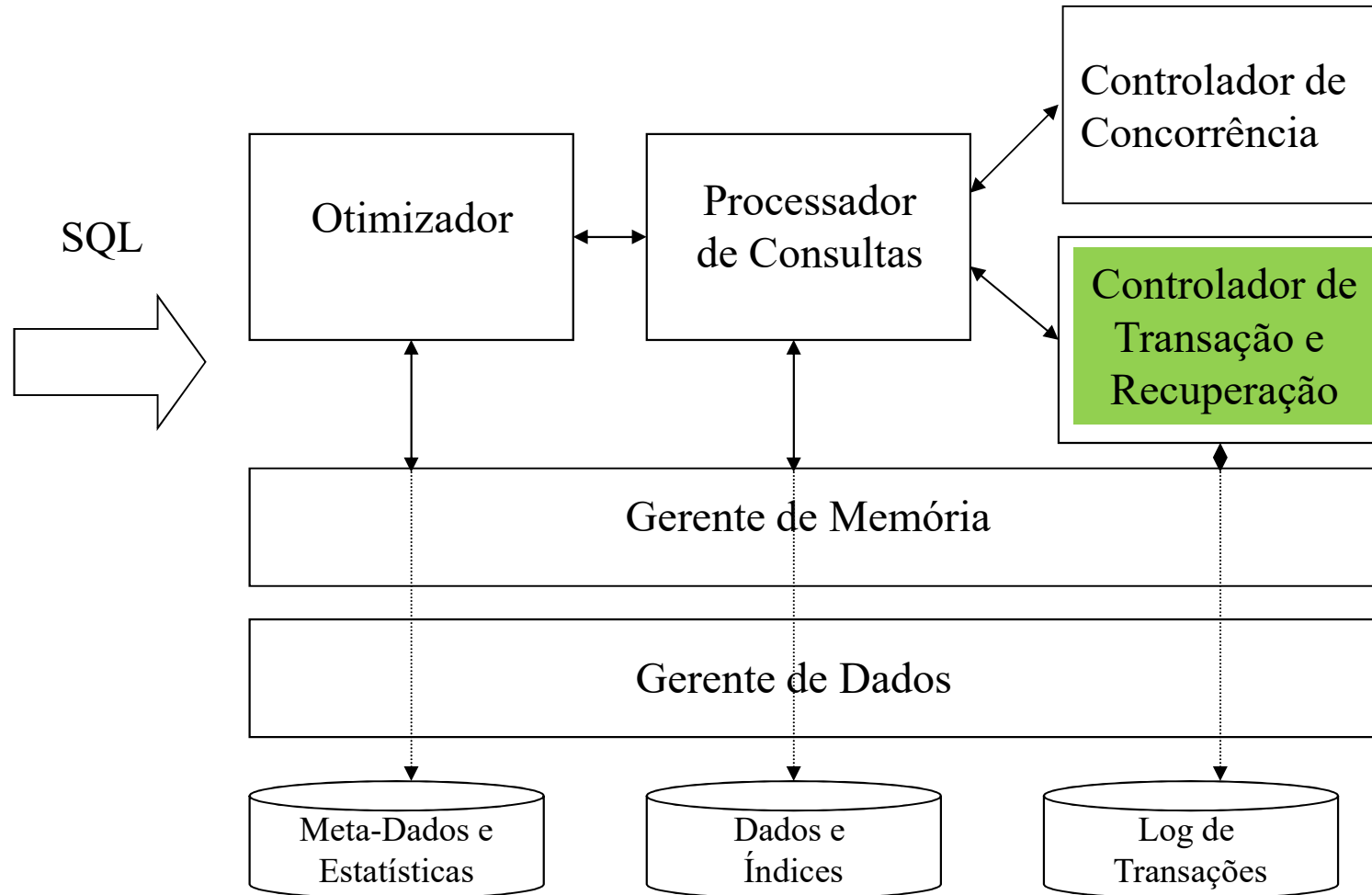
## Técnicas de Recuperação de Banco de Dados

SQL



# Arquitetura Funcional

## Componentes de um SGBD



# Técnicas de Recuperação de Bancos de Dados



## Caching de blocos de disco:

O processo de recuperação está fortemente entrelaçado com as funções do Sistema Operacional.

O disco é dividido em páginas (blocos) e estes blocos são transferidos do disco para a memória (e da memória para o disco em algumas circunstâncias).

Pode ser necessário substituir as páginas do disco para que haja espaço disponível para uma nova página. Neste caso os algoritmos de substituição de páginas do S.O. são usados (ex: LRU – last recently used ou FIFO).

# Técnicas de Recuperação de Bancos de Dados



## Caching de blocos de disco:

É usado um catálogo que indica que blocos de disco que estão em quais posições de memória. É de fato, uma tabela contendo o endereço da página do disco e o seu correspondente endereço de memória.

A cada bloco no disco é associado a um bit de **bloco sujo** (no catálogo), que indica que o bloco foi modificado e precisa ser regravado no disco.

Há ainda outro bit denominado **pin-unpin**, que indica que uma página está fixada na memória e não pode ser transferida para o disco.

# Técnicas de Recuperação de Bancos de Dados



## Checkpoints no log do sistema

Outro tipo de entrada no log é chamado de checkpoint. Um registro do tipo [checkpoint] é escrito periodicamente no log, quando o sistema grava no disco todos os buffers que tiverem sido modificados.

Assim, todas as transações que tiverem entradas no log [commit,T] antes de uma entrada [checkpoint] não necessitarão ter suas operações WRITE refeitas no caso de queda do sistema.

O SGBD decide quando submete um checkpoint. De tempos em tempos (m minutos) ou a cada t transações, m e t são parâmetros do sistema.

# Técnicas de Recuperação de Bancos de Dados



## Rollback de transação

Se uma transação falhar depois de atualizar o BD, pode ser necessário reverter a transação. Se algum valor de item de dados tiver sido alterado pela transação e gravado no BD, ele deverá ser restaurado para seus valores anteriores. As entradas do log do tipo desfazer (UNDO) são usadas para restaurar os dados antigos dos itens a serem revertidos.

Se uma transação T for revertida e outra transação S tenha lido um item de T, tiver lido o valor um item X escrito por T, S também deverá ser revertida. Se S foi revertida, qualquer transação R que tenha lido o valor de um item de dado Y de S também deve ser revertida. Esse fenômeno é chamado **reversão em cascata (cascading rollback)**.

# Técnicas de Recuperação de Bancos de Dados



## Rollback de transação

A reversão em cascata pode ser complexa e consumir muito tempo. Assim, quase todos os mecanismos de recuperação são projetados para que a reversão em cascata nunca seja exigida.

# Técnicas de Recuperação de Bancos de Dados



## Atualização adiada

A ideia é postergar qualquer atualização no BD até que a transação complete sua execução com sucesso e alcance o seu ponto de efetivação.

Durante a execução da transação as atualizações são registradas apenas no log e nos buffers de memória. Quando a transação alcançar seu ponto de efetivação e se forçar à gravação do log no disco, as atualizações serão registradas no banco. Se uma transação falhar antes de alcançar seu ponto de efetivação, não será necessário desfazer nenhuma operação, porque a transação não afetou o banco. A atualização adiada é conhecida como algoritmo **NO-UNDO/REDO**.

Essa técnica é utilizada para transações curtas e que alterem poucos itens do BD.



# Técnicas de Recuperação de Bancos de Dados



## Atualização adiada com execução concorrente

O protocolo pode ser escrito como:

1. Uma transação não pode mudar o BD até que ela alcance seu ponto de efetivação.
2. Uma transação não alcança seu ponto de efetivação até que todas as suas operações de atualização sejam registradas no log e até que seja forçada a gravação do log no disco.

# Técnicas de Recuperação de Bancos de Dados



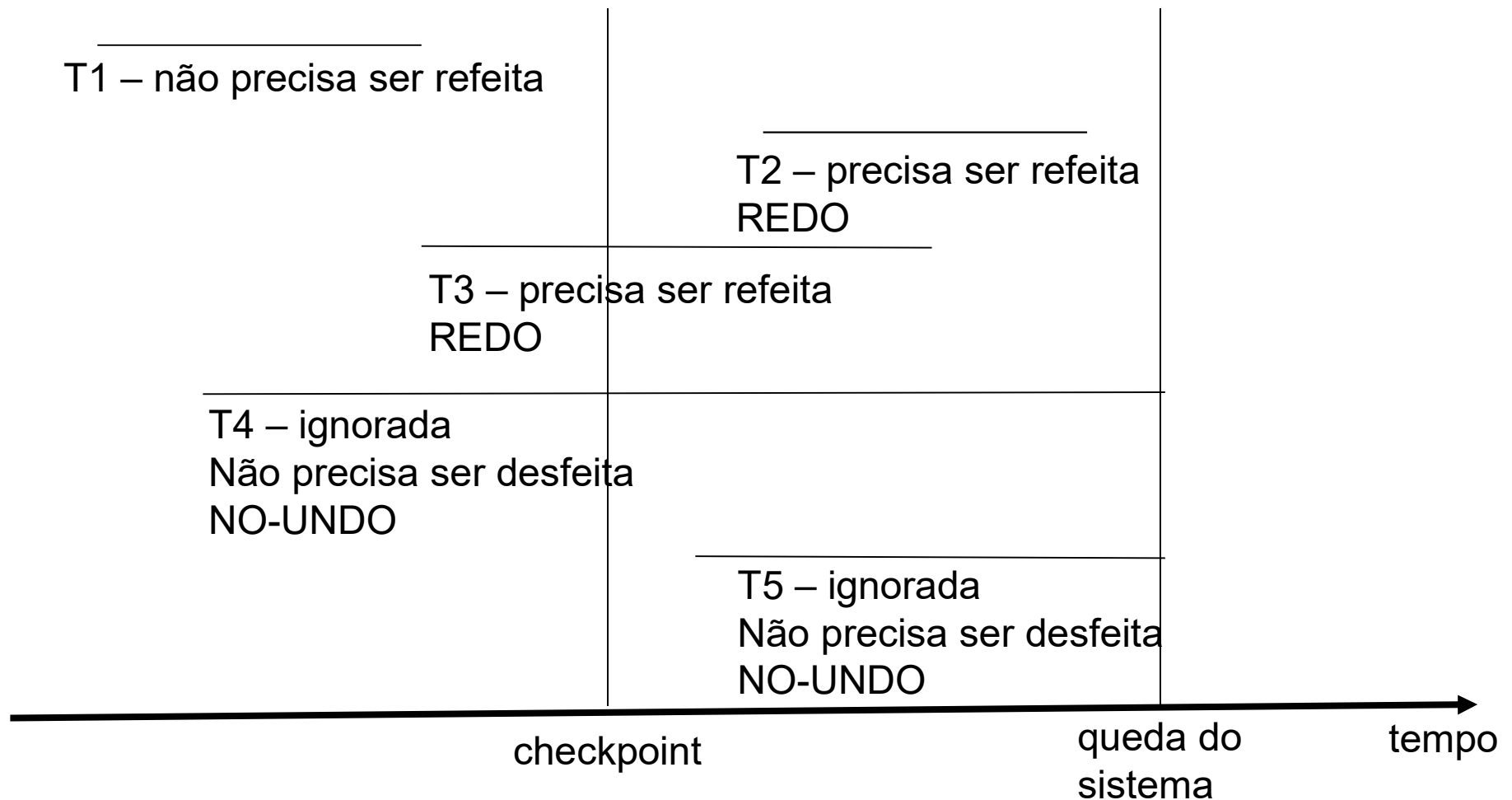
## **Atualização adiada com execução concorrente em um ambiente multiusuário**

Para sistemas multiusuários com controle de concorrência, o processo de recuperação pode ser mais complexo.

Se uma transação for interrompida por qualquer razão (ex. deadlock) ela simplesmente será re-submetida já que não alterou o BD em disco.

Uma desvantagem desse método é a de limitar a concorrência porque todos os itens permanecem bloqueados até que a transação alcance o seu ponto de efetivação.

# Exemplo de recuperação em ambiente multiusuário



# Técnicas de Recuperação de Bancos de Dados



O benefício principal do método é que as operações de transação nunca terão de ser desfeitas por duas razões:

1. Uma transação não registrará nenhuma mudança no BD em disco até que a transação seja efetivada.
2. Uma transação nunca lerá o valor de um item escrito por uma transação não efetivada porque seus itens permanecerão bloqueados até que alcance seu ponto de efetivação. Portanto, não ocorrerá reversão em cascata.

# Técnicas de Recuperação de Bancos de Dados



## Atualização imediata

O log pode ser atualizado por algumas operações de transação **antes** que ela alcance seu ponto de efetivação. No entanto, essas operações estarão registradas no log em disco porque uma gravação forçada no log foi feita antes que elas fossem aplicadas no BD, tornando possível a recuperação.

Se uma operação falhar depois de registrar algumas mudanças no log e antes de atingir seu ponto de efetivação, os efeitos da operação deverão ser desfeitos, ou seja, a transação deve ser revertida. Um exemplo de atualização forçada ocorre quando o buffer do log fica cheio e precisa ser gravado.

Essa técnica é conhecida como **UNDO/REDO** porque exige ambas as operações e é a mais usada na prática.

# Técnicas de Recuperação de Bancos de Dados

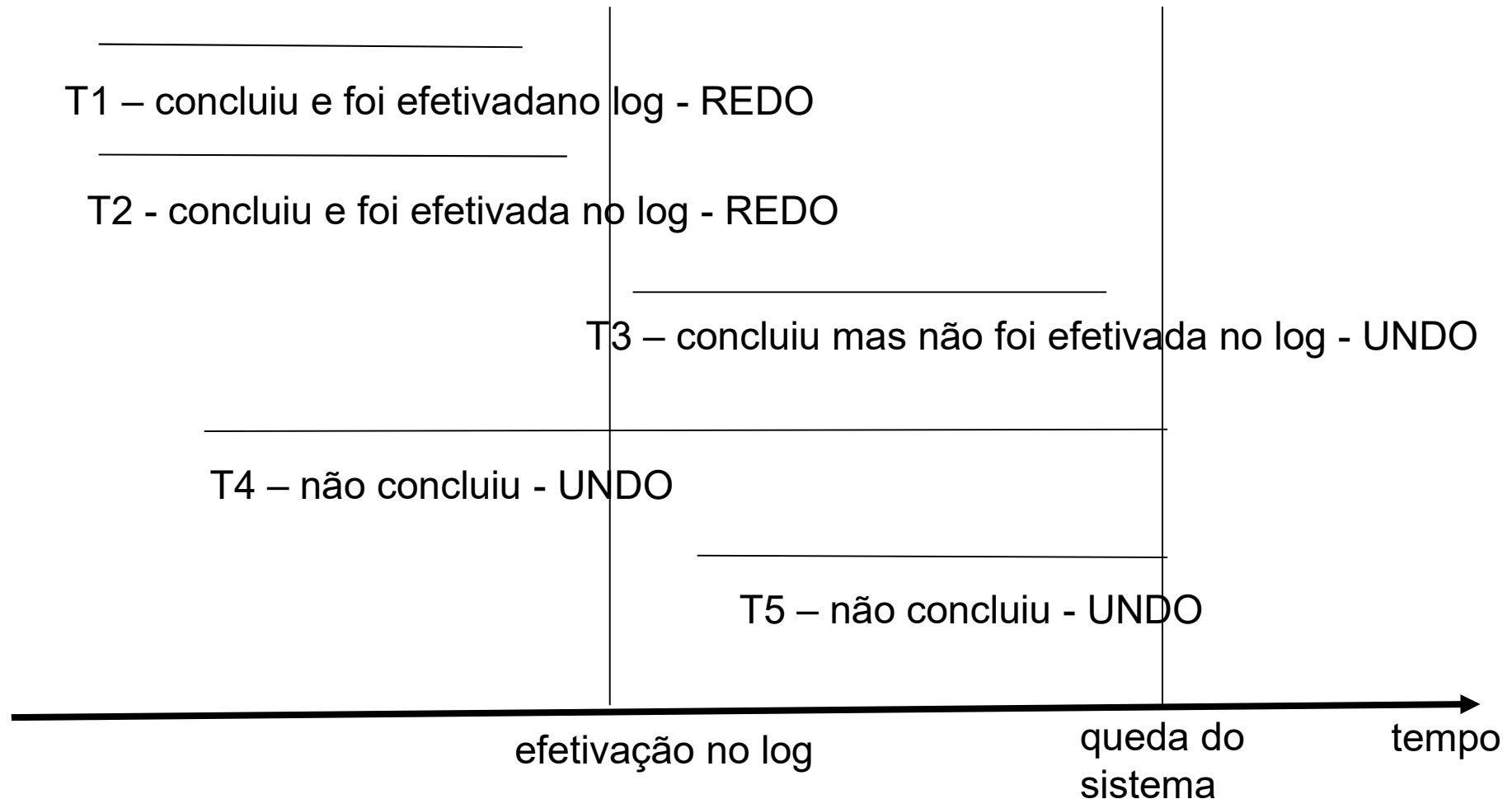


## Atualização imediata com execução concorrente

O protocolo pode ser escrito como:

1. Usar duas listas de transações mantidas pelo sistema: a das transações efetivadas no log desde o último commit e a das transações ativas.
2. Desfazer todas as operações `escrever_item` feitas pelas transações ativas (e não-efetivadas), usando o procedimento UNDO. As operações devem ser desfeitas na ordem inversa em que elas foram gravadas no log.
3. Refazer todas as operações `escrever_item` das transações efetivadas no log, na sequência em que elas foram escritas no log (procedimento REDO).

# Exemplo de recuperação em ambiente multiusuário



# Técnicas de Recuperação de Bancos de Dados



## Atualização imediata

Uma variação do algoritmo registra no BD todas as atualizações antes que a transação seja efetivada. Ela é conhecida como **UNDO/NO-REDO** (quando ocorre um crash do banco, desfaz as transações não efetivadas).



# Técnicas de Recuperação em falhas catastróficas



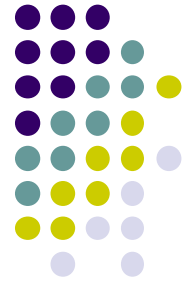
A principal técnica utilizada neste caso é o back-up do BD.

Para evitar a perda dos efeitos das transações executadas desde o último backup é comum salvar o log do sistema em intervalos menores que o usado no backup total do banco.

Essa técnica do backup em intervalos menores é denominada **backup incremental**. É mais comumente usada porque o log é substancialmente menor do que o BD e pode, portanto, ser copiado mais frequentemente. Com isso os usuários não perdem todas as informações que executaram depois do último backup total do banco.

O back-up total do banco é conhecido como **full back-up**.

# Perguntas?



# Perguntas?



1. Suponha que um SGBD utilize atualização imediata. Em que casos ocorrerão operações de UNDO e REDO? Quando essas operações são executadas pelo SGBD? Crie um exemplo para justificar sua resposta.
2. Suponha que um SGBD utilize atualização adiada. Em que casos ocorrerão operações de UNDO e REDO? Quando essas operações são executadas pelo SGBD? Crie um exemplo para justificar sua resposta.
3. Para que serve o log de um banco de dados. Quando ele é utilizado?
4. Qual a vantagem e desvantagem de se utilizar um backup full do banco de dados?