trabalho cap 21

💡 EXCLARECENDO A DUVIDO DO NOSSO COLEGA DE CLASSE RAFAEL MYAUTI → SISU (replica de banco de dados, direcionada ao Taynan)

❌ as frases em cinza não seram apresentadas, são apenas para caso surja alguma duvida sobre o topico em questão.

introdução

Os ***sistema de de processamento de transação*** são sistemas com grandes bancos de dados, ao quais incluem centenas de usuarios simultaneamente que executam transações de banco de dados. são sistemas que exigem alta disponibilidade e um agil tempo de resposta para cada centena de usuario.
ex: reservas de passagens aereas, sistemas bancarios e mercados de ações.

Logo definimos o conceito transação como um processamento logico usado para representar uma unidade logica de processamento de BD ao qual deve ser concluida ou executada por inteiro para que possa garantir uma exatidão.

> obs, caso haja perguntas:
> transação ocorre atraves de um programa de BD onde inclui inserções, exclusções, recuperações e atualizações.

Introdução ao processamento de transações:

aqui discutiremos o conceito de transação e recuperação.

sendo eles:

- topico 1 monousuarios e multiousuarios

topico 2 operações de leitura e gravções de BD
topico 3 controle de concorrencia
topico 4 recuperação, pq ter de lidar com ela em uma falha de transação ??

Sistemas de monousuário versus multiusuário

“Um critério para classificar um sistema de banco de dados é de acordo com o número de usuários que podem usar o sistema simultaneamente.” logo

monousuario: somente um usuário de cada vez pode acessar o SBD
Computadores Pessoais

SGBD Multiusuário: muitos usuários podem acessar o SBD simultaneamente
Reservas de passagens aereas, sistemas bancarios e mercados de ações.

Diferença entre sistemas de banco de dados de usuário único e multiusuário:

monousuario

Um DBMS é de usuário único se, no máximo, um usuário por vez puder usar o sistema.

SGBDs de usuário único são principalmente restritos a sistemas de computadores pessoais.

Os bancos de dados de usuário único não possuem multiprogramação, portanto, uma única CPU pode executar no máximo um processo por vez.

multiusuario

Um SGBD é multiusuário se muitos / vários usuários podem usar o sistema e, portanto, acessar o banco de dados simultaneamente.

A maioria dos DBMSs é multiusuário, como bancos de dados de sistemas de reserva de companhias aéreas, bancos de dados bancários, etc.

Vários usuários podem acessar bancos de dados e usar sistemas de computador simultaneamente devido ao conceito de Multiprogramação.

buscar um video de exemplos de sistema multiusuario para dar de exemplo explicativo e inserir aqui sua explicação! (passagem aerea/mercado de acoes)

Transações, itens de banco de dados, operações de leitura e gravação e buffers do SGBD:

“Uma transação é um programa em execução que forma uma unidade lógica de processamento de banco de dados”.

A delimitação de uma transação vem das instruções begin transaction e and transaction, onde um programa pode conter uma ou mais operações de acesso ao BD, sucedendo as operações de inserção, exclusão, recuperação e modificação. vale ressaltar que um programa de aplicação só podera conter mais transações caso haja varios limites de transação.

”Se as operações de banco de dados em uma transação não atualizarem o banco de dados, mas apenas recu perarem dados, a transação é chamada de transação somente de leitura; caso contrário, ela é conhecida como transação de leitura-gravação”

read_item - write_item → funcoes que serão explicadas posteriormente.

modelo de funcionamento das funcoes:
acha o bloco → copia o bloco(do disco) p/ o buffer na memoria principal →copia o item a ser
lido para uma variavel.
para a funcao de write_item basta acrecentar um armazene o bloco atualizado do buffer de
volta ao disco.

na etapa de armazenamento, em alguns casos ela não ocorre simultaneamente pois ela pode ser modificada conforme as demandas.

Dadas as informações sobre leitura e escrita vale ressaltar que precisamos ter um grande cuidado ao se tratar do mecanismo de controle de concorrencia.

O SGBD manterá na cache do banco de dados uma série de buffers de dados na memória principal. Cada buffer costuma manter o conteúdo de um bloco de disco do banco de dados, que contém alguns dos itens de banco de dados que estão sendo processados. Quando esses buffers estão todos ocupados, e blocos de disco de banco de dados adicionais devem ser copiados para a memória, alguma política de

substituição de buffer é utilizada para escolher quais buffers atuais devem ser substituídos. Se um buffer escolhido tiver de ser modificado, ele precisa ser gravado de volta no disco antes de ser reutilizado.

modelo FIFO → first in first out, o primeiro a entrar é o primeiro a sair

Por que o controle de concorrência é necessário

— exemplo prático para de mecanismos de controle de concorrência em um sistema de banco de dados

Para fins de controle de concorrência, uma transação é uma execução em particular de um programa em uma data, que tecnicamente pega/são/puxa suas caracteristicas unicas PK).

adiante ireia abordar os quatro possiveis problemas que podemos encontrar ao se tratar de controle de concorrencia:

- O problema da atualização perdida.
- O problema da atualização temporária (ou leitura suja)
- O problema do resumo incorreto
- O problema da leitura não repetitiva

Por que a recuperação é necessária

se uma transação for submetida ao SGBD tem que ser concluida com sucesso assim então os dados serão registrados permanentemente

pq voce pode ter varios ripos de impecilho, ue!

- *Uma falha do computador (falha do sistema):* erro no hardware, software ou rede.
- *Um erro de transação ou do sistema:* alguma insconsistencia na codificação, como um inteiro divisivel por zero
- *Erros locais ou condições de exceção detectadas pela transação*: Durante a execução da transação, podem ocorrer certas condições que necessitam de cancelamento da transação.

Por exemplo, os dados da transação podem não ser encontrados

- *Imposição de controle de concorrência:* pode abortar uma transações para resolver um estado de deadlock

- *Falha de disco:* Alguns blocos de disco podem perder seus dados devido a um defeito de leitura, gravação

- *Problemas físicos e catástrofes:* falha de energia ou de ar-condicionado, incêndio, roubo, sabotagem

Suporte para transação em SQL

nada mais é que uma unidade lógica de trabalho e tem garantias de ser atômica (ou indivisível)

em se tratando do SQL, ele não precisa de um inicio implicito como o begin transction, todavia ele pricisa de um fim explicito que seria o COMMIT ou ROLLBACK.

- modo de acesso: pode ser especificado pela leitura ou pela leitura e gravação.
- tamanho da área de diagnóstico: está relacionado à quantidade de informações de feedback que o sistema pode armazenar temporariamente para ajudar os usuários a entender e corrigir problemas durante a execução de instruções SQL.
- nível de isolamento: O nível de isolamento em bancos de dados se refere à capacidade de controlar a visibilidade das transações concorrentes em um ambiente de banco de dados
    - Leitura suja: A leitura suja ocorre quando uma transação lê dados que foram modificados por outra transação, mas ainda não foram confirmados (commit). → READ UNCOMMITTED
    - leitura não repetitiva: A leitura não repetitiva ocorre quando uma transação lê um conjunto de dados e, posteriormente, uma segunda transação modifica ou exclui parte desses dados antes que a primeira transação seja concluída. →REPEATABLE READ

- fantasma: A leitura fantasma ocorre quando uma transação lê um conjunto de dados, outra transação insere novos dados que atendem aos critérios da primeira transação e, em seguida, a primeira transação lê novamente usando os mesmos critérios, obtendo agora um conjunto de dados diferente → SERIALIZABLE

SET TRANSACTION → define características específicas para o comportamento de uma transação

`EXEC SQL WHENEVER SQLERROR GOTO UNDO` → Define uma ação a ser tomada quando ocorre um erro SQL. Neste caso, se houver um erro SQL, o controle será transferido para a etiqueta `UNDO`

`EXEC SQL SET TRANSACTION READ WRITE DIAGNOSTIC SIZE 5 ISOLATION LEVEL SERIALIZABLE;`:

- Configura a transação atual.
- `READ WRITE`: Indica que a transação envolverá operações de leitura e gravação.
- `DIAGNOSTIC SIZE 5`: Define o tamanho do bloco de diagnóstico para 5 (isso pode ser específico da implementação).
- `ISOLATION LEVEL SERIALIZABLE`: Configura o nível de isolamento da transação como SERIALIZABLE, o que significa que a transação é executada de forma isolada de outras transações, garantindo consistência nos dados.

`EXEC SQL INSERT INTO FUNCIONARIO (Pnome, Unome, Cpf, Dnr, Salario) VALUES ('Roberto', 'Silva', '99100432111', 2, 35.000);`:

- Realiza uma operação de inserção na tabela `FUNCIONARIO`.
- Insere um novo registro com os valores fornecidos nas colunas `Pnome`, `Unome`, `Cpf`, `Dnr` e `Salario`.

`EXEC SQL UPDATE FUNCIONARIO SET Salario = Salario * 1.1 WHERE Dnr = 2;`:

- Realiza uma operação de atualização na tabela `FUNCIONARIO`.
- Aumenta o salário em 10% para todos os registros onde o valor da coluna `Dnr` é igual a 2.

`EXEC SQL COMMIT;`:

- Confirma a transação, aplicando as alterações realizadas pela transação ao banco de dados.

`GOTO THE_END;`:

- Um comando de desvio incondicional para a etiqueta `THE_END`. Este é um ponto no código onde a execução deverá continuar após a conclusão bem-sucedida da transação.

`UNDO: EXEC SQL ROLLBACK;`:

- Etiqueta que é referenciada em caso de erro. Se ocorrer um erro SQL, a execução será transferida para esta etiqueta, onde uma instrução `ROLLBACK` será executada, desfazendo todas as alterações feitas pela transação.

`THE_END: ... ;`:

- Ponto onde a execução continua após o final bem-sucedido da transação ou as alterações são abortadas.