

1 - A execução simultânea de transações de banco de dados em um sistema multiusuário refere-se à capacidade de executar várias transações simultaneamente por diferentes usuários. O controle de simultaneidade é necessário para garantir a execução correta e consistente de transações simultâneas, mantendo a integridade dos dados e evitando situações indesejadas, como leituras incorretas, atualizações duplicadas ou atualizações perdidas.

Leitura Suja (Dirty Read): Imagine um sistema bancário onde o cliente A transfere dinheiro para o cliente B. Se a transação A ainda não estiver completa e B ler o saldo, ele poderá ver o saldo errado, causando confusão. O controle concorrente impede que B veja o saldo "sujo" de A antes de concluir a transação.

Escrita Perdida (Lost Update): Suponha que dois usuários, X e Y, atualizem o saldo da mesma conta bancária ao mesmo tempo. Se o sistema não gerenciar a simultaneidade, uma atualização pode substituir outra, resultando em perda de dados. O controle de simultaneidade garante que as atualizações sejam feitas de forma correta e consistente.

Anomalias de Serialização: Considere uma situação em que o usuário C lê o saldo de uma conta enquanto o usuário D faz vários depósitos e saques nesta conta. Se as transações não forem sequenciadas corretamente, as leituras de C podem causar um saldo desequilibrado, porque as transações de D podem afetar o equilíbrio entre leituras e atualizações em C. O controle de simultaneidade evita essas anomalias, garantindo que os eventos sejam sequenciados corretamente.

Dependências entre Transações: Imagine um sistema de gerenciamento de estoque onde duas transações, uma para adicionar um novo produto e outra para atualizar o estoque, são realizadas simultaneamente. Se o evento de atualização de estoque ocorrer antes do evento de adição de produto, isso pode causar inconsistências nos níveis de estoque. O controle de simultaneidade garante que as transações sejam executadas na ordem correta, mantendo a integridade dos dados.

2 - As falhas em sistemas de computação podem ser classificadas de várias maneiras, dependendo dos critérios usados para categorizá-las como Falhas Transientes, Intermitentes, Permanentes, Software, Hardware, Rede, Energia.

Uma falha catastrófica é uma falha extrema com consequências sérias e de longo alcance. Essa falha geralmente resulta em perda significativa de dados, interrupções críticas de serviço ou até mesmo danos físicos. A falha catastrófica pode ser causada por eventos como incêndios, terremotos, ataques cibernéticos maciços ou mau funcionamento de sistemas críticos.

3 - O log do sistema é uma ferramenta essencial em sistemas de computação para registrar informações relevantes sobre as atividades, eventos e estados que ocorrem no sistema.

Podemos citar alguns registros de log, como Monitoramento, Diagnóstico de Problemas, Auditoria e Segurança, Recuperação e Resiliência.

Pontos de transferência de transação (também conhecidos como "pontos de commit") são marcos de transação que indicam quando uma transação foi concluída com sucesso e os resultados podem ser armazenados permanentemente no sistema. Eles são importantes pois garantem as seguintes características:

Atomicidade: Os pontos de confirmação garantem a nuclearidade das transações, o que significa que uma transação é tratada como uma unidade indivisível. Se a transação falhar, as alterações serão descartadas.

Consistência: Os pontos de confirmação garantem que as alterações feitas durante a transação sejam aplicadas de forma consistente em todo o sistema.

Durabilidade: Uma vez que uma transação é confirmada e seu ponto de execução é registrado, as alterações são permanentes e persistem mesmo em caso de falha do sistema.

Recuperação: Em caso de falha do sistema após um ponto de confirmação, o log pode ser usado para recuperar o estado do sistema até o ponto de confirmação mais recente.

Auditoria: Os pontos de validação também podem ser usados para rastrear determinadas ações de transação, o que é importante para auditoria e conformidade.

4 - **Atomicidade:** O recurso de atomicidade garante que uma transação seja tratada como uma unidade indivisível. Isso significa que todas as operações na transação serão concluídas com sucesso ou nenhuma delas será concluída. Se qualquer parte da transação falhar, todas as alterações feitas até aquele ponto serão revertidas, restaurando o sistema a um estado consistente. A atomicidade evita estados intermediários inconsistentes e garante que o banco de dados não seja deixado em um estado incompleto ou inconsistente devido a falhas.

Consistência: A propriedade de consistência garante que um evento altere o banco de dados de um estado válido para outro estado válido. Isso significa que as transações devem manter a integridade dos dados e garantir a conformidade com as restrições e regras de integridade definidas no banco de dados antes e depois da transação. A consistência refere-se à ideia de que as transações não devem violar regras e restrições predeterminadas, o que garante que os dados permaneçam consistentes.

Isolamento: Isolamento refere-se à capacidade de um evento ocorrer independentemente e ser isolado de outros eventos simultâneos. Isso garante que as ações de uma transação não sejam visíveis ou afetem outras transações em andamento. O isolamento evita problemas como leituras sujas, gravações perdidas e erros de serialização, mantendo a integridade dos dados e evitando conflitos entre transações simultâneas.

Durabilidade: O recurso de durabilidade garante que os resultados de uma transação concluída sejam permanentemente armazenados no banco de dados, mesmo após uma falha do sistema. Depois que uma transação é confirmada, suas alterações tornam-se permanentes e não podem ser desfeitas. Isso garante que os dados sejam preservados mesmo em situações de falha, o que garante a confiabilidade do sistema.

5 - Um schedule, também conhecido como plano de execução, é uma sequência ordenada de operações (leitura e gravação) realizadas entre transações em um sistema de banco de dados. Representa a sequência de ações de diferentes eventos em relação ao tempo. O estudo de schedules é fundamental para entender como as transações interagem e como as propriedades ACID são mantidas durante a execução concorrente.

Schedule Recuperável:

Um schedule é considerado recuperável se, para qualquer par de transações T1 e T2, se T2 lê um valor escrito por T1, então T1 termina antes de T2 começar. Isso garante que as operações de leitura de T2 não leiam valores de T1 que ainda não foram confirmados, evitando leituras incorretas. Schedule recuperável ajuda a evitar problemas como leituras sujas.

Schedule Sem Cascata:

Um schedule é considerado sem cascata se uma transação T1 que for revertida não permite que outra transação T2, que já leu valores de T1, seja confirmada. Isso evita a propagação de valores de transações abortadas para outras transações, garantindo que as operações de T2 sejam baseadas em valores consistentes. Isso ajuda a evitar problemas como escritas perdidas.

Schedule Estrito:

Um schedule é considerado estrito se todas as operações de escrita de uma transação T1 precedem todas as operações de leitura de outra transação T2. Isso significa que as operações de leitura nunca veem valores escritos pela mesma transação. O schedule estrito elimina o fenômeno de leituras não consistentes, garantindo que as operações de leitura sejam sempre baseadas em valores confirmados.

6 - Um "schedule serial" é uma sequência de transações em que cada transação é completamente executada antes que a próxima comece. Isso significa que as transações são executadas de forma isolada, uma de cada vez, sem intercalação de suas operações. É considerado correto porque, como mencionado anteriormente, ele garante a consistência e previsibilidade dos resultados.

Um "schedule socializável" é uma sequência de transações em que as operações podem ser reorganizadas ou intercaladas, desde que a ordem de execução preserve as dependências de dados entre as operações. É considerado correto porque mantém a consistência dos dados e preserva as dependências de dados entre as operações, mesmo que as operações sejam reorganizadas em relação à sua ordem original.

7 -

8 - Alternativa D

9 -