TDE 2 - Controle de Acesso e Concorrência de Transações - 06h 45m (Relógio)

Lucas Azevedo Dias

Exercícios TE06

1. Que conta é designada como proprietária de uma relação? Que privilégios o proprietário de uma relação possui?

   Normalmente, a conta usada quando a relação foi criada se torna a proprietária da relação. Assim, ela ganha todos os privilégios sobre aquela relação (que seriam os privilégios de leitura, de modificação e de referência).

2. Como o mecanismo de visão é usado como um mecanismo de autorização?

   Pode-se usar o mecanismo de visão como um mecanismo de autorização quando uma certa conta quer apenas conceder acesso a certos campos de uma relação específica a outra. Dessa forma, a primeira poderia criar uma visão dessa relação com esses atributos e conceder à segunda o direito de leitura sobre ela (o que limita o acesso da segunda a apenas o que a primeira deseja que ela tenha e nada mais).

3. O que significa a concessão de um privilégio? O que significa a revogação de um privilégio?

   Quando há a concessão de um privilégio, aquele usuário-alvo passa a possuir o direito para realizar certos comandos especificados. E, porém, quando há a revogação, o usuário-alvo perde esse direito, assim, não podendo mais realizar esses certos comandos.

4. Liste os tipos de privilégios disponíveis em SQL.

   São os privilégios de leitura (ou *select*), de modificação e de referência.

5. Qual é a diferença entre controle de acesso discricionário e obrigatório?

   No discricionário, há a concessão ou revogação de privilégios aos usuários, enquanto, no obrigatório, há a imposição de certas normas através da classificação em classes dos usuários e dos dados.

6. Quais são os diferentes tipos de ataques de Injeção de SQL?

   Os tipos de ataques de injeção de SQL são: manipulação de SQL (em que há a modificação de uma chamada do próprio programa), injeção de código (onde há a inserção de mais instruções SQL para explorar alguma falha) e injeção de chamada de função (em que uma função ou chamada é inserida nas instruções SQL).

Exercícios TE07

1. Discuta as propriedades de atomicidade, durabilidade, isolamento e preservação da consistência de uma transação de banco de dados.

Toda transação em um banco de dados precisa obedecer a certas propriedades para garantir a integridade dos dados trabalhados. Dessa forma, essas propriedades seriam:

- Atomicidade: todas as operações dentro de uma transação são tratadas como um bloco indivisível (ou são todas realizadas, ou então nenhuma);
- Consistência: após a execução de uma transação, o banco de dados deve continuar consistente;
- Isolamento: uma transação não pode sofrer interferências de outros comandos sendo executados em paralelo;
- Durabilidade: Após uma transação ser completada, as mudanças que ela realizou no banco de dados devem perdurar (mesmo com possíveis falhas no sistema).

2. Discuta como a serialização é usada para impor o controle de concorrência em um sistema de banco de dados. Por que a serialização às vezes é considerada muito restritiva como uma medida da exatidão para os schedules?

   A serialização atua garantindo que não haja interferências entre as operações de cada transação dentro de uma *schedule* resultando em um estado final inconsistente para o banco de dados, fazendo com que o estado final seja equivalente a uma execução em série das transações (em ordem cronológica de requisição). Contudo, para fazer a verificação de uma *schedule* e saber se ela é serializável, seria necessário averiguar todo o cálculo das operações dentro das transações, o que é inviável em termos computacionais. Dessa forma, acaba-se usando métodos mais simples considerando apenas as operações de leitura e de escrita de dados que acabam restringindo mais as *schedules*.

3. Descreva os quatro níveis de isolamento em SQL.

   Há quatro níveis de isolamento dentro do SQL:

   - Serializável: teoricamente garante a serialização, porém, pela forma como é implementado, pode permitir algumas execuções não serializáveis;
   - Leitura reprodutível: apenas dados confirmados podem ser lidos e entre duas leituras o dado não pode ser atualizado por nenhuma outra transação;
   - Leitura confirmada: apenas dados confirmados podem ser lidos (mas entre duas leituras o dado pode ser atualizado por outra transação);
   - Leitura não confirmada: dados não confirmados também podem ser lidos.

4. Defina as violações causadas por cada um dos seguintes itens: leitura suja, leitura não repetitiva e fantasmas.

   - Leitura suja: ocorre quando há a leitura de um dado ainda não confirmado (pode acarretar problemas se haver um *rollback* da transação que não havia confirmado a mudança);
   - Leitura não repetitiva: ocorre quando o dado lido por uma transação não continua constante por causa de uma outra transação ter atualizado o valor no meio-tempo (gerando inconsistência sobre o valor do dado);

- Fantasma: ocorre quando, após a leitura por uma transação, um dado é inserido ou deletado no meio-tempo por outra (gera o aparecimento de um “fantasma” em uma próxima leitura, ou seja, algo que supostamente não deveria estar lá).

5. O que é o protocolo de bloqueio em duas fases? Como ele garante a serialização?

   É um protocolo de bloqueio que estabelece que cada transação pode apenas se encontrar em uma das duas fases: fase de crescimento (adquire bloqueios, mas não libera nenhum) ou fase de encolhimento (libera bloqueios, mas não adquire nenhum). Assim, cada dado só pode ser trabalhado por uma única transação em um momento e apenas será liberado quando as operações que seriam feitas nele forem concluídas, o que, dessa maneira, garante a serialização.