

*"Um SGBD deve ser capaz de garantir que apenas planos de execução serializáveis e recuperáveis sejam permitidos e que nenhuma ação de transações efetivadas seja perdida ao desfazer transações canceladas. Normalmente, um SGBD utiliza um protocolo de bloqueio."*

(Ramakrishnan, Gehrke, Database Management Systems)

Suponha um Sistema Gerenciador de Banco de Dados Relacional que tradicionalmente utiliza o bloqueio de duas fases (*Strict 2PL*)

1. Cite as duas regras utilizadas pelo protocolo de bloqueio em duas fases.
2. Explique o problema de impasse (*deadlock*) citando uma das duas políticas de tratamento.
3. Apresente um ponto negativo em relação à política de tratamento de *deadlock*.

**GABARITO**

A questão vale 40 pontos.

Item 1 - O bloqueio em duas fases tem como regra primeiro, solicitar o bloqueio compartilhado ou exclusivo sobre o objeto de banco de dados e manter bloqueado até que seja finalizada a transação. A segunda regra é quando uma transação é cancelada ou finalizada, ela libera todos os bloqueios atualiza a tabela de bloqueios do gerenciador de bloqueios do objeto e verifica o início da fila de pedidos de bloqueio desse objeto, se puder ser atendido o gerenciador de bloqueio desperta a transação que aguardava ou se for vários pedidos de bloqueio compartilhado, todas as outras transações podem acorrer em conjunto.

Item 2 - O gerenciador de bloqueio organiza os bloqueios dos objeto em uma estrutura de grafo onde os nós correspondem a transações e os bloqueios pelos arcos. Portanto se uma transação  $T_i$  for suspensa porque um bloqueio solicitado e não pode ser concedido, ela tem que esperar até que  $T_j$  que atualmente mantém bloqueios conflitantes os libere. Se esse bloqueio fechar um ciclo no grafo, ou seja,  $T_i$  estiver aguardando  $T_j$  liberar o bloqueio e  $T_j$  estiver aguardando  $T_i$  liberar o bloqueio, então caracteriza um impasse ou *deadlock*. Para solucionar uma das duas transações devem ser canceladas pela política de esperar-morrer se  $T_i$  tem prioridade mais alta ela pode esperar; caso contrário é cancelada ou ferir-esperar se  $T_i$  tem prioridade mais alta, cancela  $T_j$ ; Caso contrário  $T_i$  espera.

Item 3 - Para resolver os impasses, independente das diferentes políticas, a transação de menor prioridade deve ser cancelada. O problema está quando essa menor prioridade não estiver alinhado com o tamanho do rollback que essa transação terá que realizar para garantir o ACID o que levar muito tempo e ocasionar outros bloqueios indesejáveis.

*"Dentre as estratégias de sintonia fina mais comumente utilizadas por administradores de bancos de dados, estão a construção de estruturas de acesso, tais como índices, índices parciais e visões materializadas, a desnormalização e a reescrita de consultas. Por outro lado, o particionamento de tabelas consiste na transformação de uma tabela em dois ou mais fragmentos, onde cada fragmento é, na verdade, um subconjunto dos dados da tabela gerado com base em critério pré-definido."*

(Almeida, Ana, Medeiros, Antony, Costa, Rogério, Lifschitz, Sergio: Particionamento como Ação de Sintonia Fina em Bancos de Dados Relacionais)

Considere o uso de particionamento de tabelas em SGBD Relacional.

1. Explique o que é particionamento horizontal e vertical.
2. Cite três vantagens e três desvantagens do uso de particionamento.
3. Cite e explique as quatro maneiras de particionar uma tabela no SGBD Oracle.

#### GABARITO

A questão vale 40 pontos.

Item 1 - O particionamento de tabelas horizontal divide-se a tabela em subconjuntos disjuntos de tuplas, sendo cada subconjunto, partição, definido como base em um critério, composto por um ou mais atributos e os respectivos valores para cada partição. A fragmentação horizontal será denominada primária se a chave do particionamento for um atributo da tabela que está sendo particionada ou derivada no caso contrário. No particionamento vertical, divide-se a tabela em subconjuntos de colunas, repetindo-se sua chave primária em cada partição.

Item 2 - Vantagens: (i) Desempenho em leitura, ao particionar uma tabela, o índice parcial irá abranger apenas a massa de dados daquela partição. Se os particionamentos estiverem distribuídos em discos distintos, uma consulta na partição atual não influenciará uma consulta a períodos anteriores, o que demonstra ser bastante vantajoso. (ii) Desempenho em gravação, se efetuar a gravação de dados em áreas distintas da partição, como caso da partição horizontal, você terá um aumento de performance, pois o processo será dividido em mais de um meio de gravação, podendo acontecer de forma concorrente. (iii) Expurgo de dados, poderá expurgar um volume dado eliminando a partição. Porém só faz sentido para particionamento vertical, para horizontal não faz sentido manter a chave e eliminar o conteúdo. Desvantagem: (i) Administração, administrar tabelas que possuem muitas partições tende a ser muito complexo, se a definição incorreta do range de dados pode tornar certos dados inacessíveis ou uma alocação excessiva em determinada partição. (ii) Integridade, as regras de partição deve ser bem claras pois podem ocorrer gravações indevidas caso surjam dados diferentes das regras especificadas. (iii) Consultas, cada comando de consulta que fizer acesso a tabela particionada deve conter em sua cláusula (WHERE) a coluna que está sendo usada para partição caso contrário toda a tabela será varrida na busca pelo dado.

Item 3 - (i) Partição por range, intervalo de valores, normalmente datas que é a forma mais comum (ii) Lista, divisões são baseadas em lista de valores normalmente usa a coluna das chaves estrangeiras, por exemplo, País, Estado (iii) Hash, trabalha em um faixa de valores, que não irão se encaixar nem na opção de range nem de lista (iv) Composto, quando utiliza mais de uma forma de particionamento, normalmente ligado a situações de sub-partição.