

Exercício - Questão da 2ª Prova 2018-2

MC536 - Bancos de Dados: Teoria e Prática
 Instituto de Computação
 Universidade Estadual de Campinas

Turmas A, B, C e D
 André Santanchè
 13/11/2018

Questão 3

Um sistema de aplicação na bolsa de valores mantém as seguintes tabelas:

<pre>CREATE TABLE Empresa (codigo_empresa VARCHAR(5), nome_empresa VARCHAR(100) UNIQUE, PRIMARY KEY (codigo_empresa));</pre>	<pre>CREATE TABLE Acao (codigo_acao VARCHAR(5), codigo_empresa VARCHAR(5), valor_unitario FLOAT, PRIMARY KEY (codigo_acao), FOREIGN KEY (codigo_empresa) REFERENCES Empresa(codigo_empresa));</pre>	<pre>CREATE TABLE Operacao (codigo_operacao VARCHAR(5), compra_venda CHAR, codigo_acao VARCHAR(5), quantidade INT, valor_operacao FLOAT, PRIMARY KEY (codigo_operacao), FOREIGN KEY (codigo_acao) REFERENCES Acao (codigo_acao));</pre>
--	---	---

A tabela **Empresa** registra o código e o nome das empresas. A tabela **Acao** mantém as ações disponíveis na bolsa de valores, identificadas pelo seu código, registrando o código da empresa associada à ação e o seu valor. A tabela **Operacao** registra cada operação de compra ou venda de ação, identificada por um código e tipo de operação no campo compra_venda (valores: 'C' - compra ou 'V' - venda); além do código da ação associada à operação, a quantidade de ações envolvidas na operação e o valor total da operação. Na tabela **Acao** existem, pelo menos, os seguintes registros:

<pre>INSERT INTO Acao values ('PB05', 'PETRO', 25); INSERT INTO Acao values ('VR07', 'VALE', 123);</pre>
--

Considere as seguintes duas transações sendo executadas concorrentemente (parte do mesmo plano), com operações de leitura e gravação possivelmente intercaladas. Nos INSERTs de T2, o valor de cada campo valor_operacao será o resultado do respectivo SELECT.

T1	T2
<pre>UPDATE Acao SET valor_unitario = valor_unitario * 1.1 WHERE valor_unitario <= 100;</pre>	<pre>INSERT INTO Operacao VALUES ('1A', 'V', 'PB05', 100, (SELECT valor_unitario * 100 FROM Acao WHERE codigo_acao = 'PB05')); INSERT INTO Operacao VALUES ('2A', 'C', 'PB05', 200, (SELECT valor_unitario * 200 FROM Acao WHERE codigo_acao = 'PB05'));</pre>

Para cada uma das questões a seguir que solicitar um plano de execução, utilize o identificador da transação (T1: ou T2:) antes de cada instrução no plano. As instruções a ser usadas no plano (quando necessário) são: read (leitura), write (gravação), rlock/wlock/unlock (bloqueio), start (início transação), commit (transação recebe commit), abort (transação abortada). Para representar um upgrade de lock, use wlock para um item que a transação já tenha um rlock. Cada instrução que envolver uma tupla da tabela deve indicar a tabela e a tupla pela sua chave primária - e.g.: T1: read(Acao, 'PB'). O plano deve desconsiderar as leituras adicionais realizadas para a localização dos registros no banco. Utilize o mínimo de operações para o que for solicitado e não é necessário representar explicitamente os locks, a menos que sejam fundamentais para a ilustração de um problema ou solução.

- a) Existe algum plano de execução para T1 e T2 com problema de Isolamento, que pode resultar em um resultado inconsistente? Se sim, apresente um plano de execução que tenha este problema ou um fragmento deste plano com foco no problema. Indique de forma sintética qual o problema. Construa o grafo de precedência deste plano.

T1	T2
<pre>UPDATE Acao SET valor_unitario = valor_unitario * 1.1 WHERE valor_unitario <= 100;</pre>	<pre>INSERT INTO Operacao VALUES ('1A', 'V', 'PB05', 100, (SELECT valor_unitario * 100 FROM Acao WHERE codigo_acao = 'PB05')); INSERT INTO Operacao VALUES ('2A', 'C', 'PB05', 200, (SELECT valor_unitario * 200 FROM Acao WHERE codigo_acao = 'PB05'));</pre>

Possível Resposta:

Plano	Grafo de Precedência
<pre>T2: start T2: read(Acao, 'PB05') T2: write(Operacao, '1A') T1: start T1: read(Acao, 'PB05') T1: Acao.valor_unitario = Acao.valor_unitario * 1.1 T1: write(Acao, 'PB05') T1: commit T2: read(Acao, 'PB05') T2: write(Operacao, '1B')</pre>	<p style="text-align: center;">Acao, PB05</p> <pre> graph LR T1((T1)) -- "Acao, PB05" --> T1 T1 -- "Acao, PB05" --> T2((T2)) T2 -- "Acao, PB05" --> T1 </pre> <p style="text-align: center;">Acao, PB05</p>
Justificativa:	Na execução dessa transação o valor da ação para a operação 1A é diferente de 2A. Isso não é equivalente a nenhuma execução serial e portanto é inconsistente.

b) Escreva o trecho de um log referente a um plano de execução restaurável, no qual haja um abort e T1 tenha que ser refeita e T2 tenha que ser desfeita. Indique no log o ponto onde o plano aborta. No espaço abaixo do log explique por que se o protocolo Write-Ahead Logging (WAL) não fosse usado não seria possível a recuperação. Para a indicação do Item lido/modificado (na coluna Item) informe a tabela e a chave primária do Item. Nos campos BFIM e AFIM registre apenas os valores dos campos modificados. Para fins de simplificação, a tabela abaixo representa apenas uma parte das colunas de um log completo. O tamanho do trecho do log deve ser suficiente para ilustrar o cenário.

Possível Resposta:

T1	T2
<pre>UPDATE Acao SET valor_unitario = valor_unitario * 1.1 WHERE valor_unitario <= 100;</pre>	<pre>INSERT INTO Operacao VALUES ('1A', 'V', 'PB05', 100, (SELECT valor_unitario * 100 FROM Acao WHERE codigo_acao = 'PB05')); INSERT INTO Operacao VALUES ('2A', 'C', 'PB05', 200, (SELECT valor_unitario * 200 FROM Acao WHERE codigo_acao = 'PB05'));</pre>

Log				
Transação	Operação	Item	BFIM	AFIM
T1	start			
T1	read	Acao, 'PB05'		
T1	write	Acao, 'PB05'	20	22
T1	commit			
T2	start			
T2	read	Acao, 'PB05'		
T2	write	Operacao, '1A'		1A, V, PB05, 2200
	crash			

Como o WAL sempre grava no log antes de dar um commit e em T1 foi dado um commit, portanto todas as suas foras gravadas no log, será possível restaurar T1 a partir do log. Se não houvesse esta garantia, pode ser que os dados comitados estivessem em um buffer (não no disco nem no log), não havendo registro para recuperação após o crash.