

Prova 2 – Implementação de BD Prof: Sérgio Mergen

Nome do aluno: _____ Matrícula do aluno: _____

1)(1,6 pontos) A Figura 1 apresenta um *schedule* que **não** é equivalente em conflito a um *schedule* serial. Com base nesse *schedule*, responda as seguintes questões.

- (0,8 ponto) Remover **uma** instrução do **final** de **uma** transação faz com que o *schedule* passe a ser equivalente a um *schedule* serial. Que transação é essa?
- (0,8 ponto) Mostre todos *schedules* seriais que são equivalentes ao *schedule* modificado.

T1	T2	T3	T4	T5
		Read(A)		
	Write(F)			
Read(B)				
Read(C)				
				read(C)
			Read(A)	
		Read(F)		
			Read(D)	
			Read(F)	
				Write(C)
	Write(B)			
Write(A)				

Figura 1 – Schedule não serial em conflito

2) (1,5 pontos) A Figura 2 apresenta as instruções de três transações. Com base nessas transações, monte um único *schedule* de execução (com *aborts*) que seja ao mesmo tempo:

- Equivalente a um *schedule* serial
- Recuperável
- **Não** livre de *rollback* em cascata. O *abort* de uma transação deve provocar *abort* em cascata de outras duas transações.

T1	T2	T3
Read(F)	write(D)	Read(C)
read (D)	read(B)	write(B)

Figura 2 – Instruções de três transações

Observação: Para responder as questões três e quatro é necessário simular como o gerenciador de *locks* controla a aquisição e liberação de acesso aos recursos requisitados pelas transações da Figura 3. Na simulação, considere os seguintes aspectos:

- As transações são atendidas pelo gerenciador em ciclos de execuções, onde em cada ciclo todas transações tem o direito de tentar executar uma instrução.
- Dentro de cada ciclo a ordem de execução é determinada pelo número da transação.
- O gerenciador utiliza o protocolo *two fase locking*.

3)(2,5 pontos) Monte um *schedule* de execução usando a estratégia de prevenção de *deadlock* “wound-wait”. As esperas **não** precisam ser indicadas. Os *aborts* e *commits* sim.

4)(2,5 pontos) Para este exercício, considere que nenhuma estratégia de prevenção de *deadlock* seja usada e que tenha ocorrido um *checkpoint* após a segunda instrução de T3. Ainda, considere que todos os itens de dados tenham 0 (zero) como valor inicial. Com base nisso, responda as seguintes questões:

- Mostre o *log* de recuperação com *checkpoints* que seria gerado.
- Quais ações de log teriam que ser realizadas caso ocorresse um erro **imediatamente antes** do *commit* de T2?

T1: read(A); Write(B=1) read(C).	T2: read(B); Write(C=2).	T3: read(C); Write(D=3). Write(E=10)	T4: read(D); Write(E=4).	T5: read(E); read(A).
---	--------------------------------	---	--------------------------------	-----------------------------

Figura 3 – Instruções de cinco transações

5)(1 ponto) Quais são os níveis de isolamento que fazem parte do SQL-92? Responda na ordem do mais forte ao mais fraco.

6)(1 ponto) Existe uma variação do protocolo de *lock* de duas fases que garante que sejam gerados *schedules* recuperáveis. Qual o nome desse protocolo? Como ele se diferencia do original?

7)(1 ponto) O *starvation* é um problema que ocorre quando o gerenciador de transações não é bem projetado. Quando o *starvation* ocorre? Como isso pode ser evitado?