

Resolução e Critérios de Correção

U.C. 21103

Sistemas de Gestão de Bases de Dados

13 de fevereiro de 2014

INSTRUÇÕES

- O tempo de duração da prova de p-fólio é de 90 minutos.
- O estudante deverá responder à prova na folha de ponto e preencher o cabeçalho e todos os espaços reservados à sua identificação, com letra legível.
- Visto que o enunciado da prova não é utilizado para resposta, poderá ficar na posse do mesmo.
- Verifique no momento da entrega das folhas de ponto se todas as páginas estão rubricadas pelo vigilante. Caso necessite de mais do que uma folha de ponto, deverá numerá-las no canto superior direito.
- Em hipótese alguma serão aceites folhas de ponto dobradas ou danificadas.
- Exclui-se, para efeitos de classificação, toda e qualquer resposta apresentada em folhas de rascunho.
- Os telemóveis deverão ser desligados durante toda a prova e os objectos pessoais deixados em local próprio da sala das provas presenciais.
- O enunciado da prova é constituído por 2 páginas e termina com a palavra **FIM**. Verifique o seu exemplar do enunciado e, caso encontre alguma anomalia, dirija-se ao professor vigilante nos primeiros 15 minutos da mesma, pois qualquer reclamação sobre defeitos de formatação e/ou de impressão que dificultem a leitura não será aceite depois deste período.
- Utilize unicamente tinta azul ou preta.
- O p-fólio é sem consulta. A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.

A informação da avaliação do estudante está contida no vetor das cotações:

Questão: 1 2 3 4 5

C: 25 25 25 25 20 décimas

Grupo A – Sistemas de Bases de Dados

1. (2,5 valores) Para sistemas de gestão de bases de dados qual a melhor política de substituição das memórias temporárias: LRU (least recently used) ou MRU (most recently used)? Justifique a sua resposta.

(Resposta: 1 página)

Resposta:

- As políticas de substituição das memórias temporárias (buffers) pretendem minimizar o acesso a disco. As duas mais conhecidas são: a política LRU (least recently used) e a MRU (*most recently used*).

- Nos sistemas operativos geralmente utilizam a política LRU (*least recently used*) de substituição de blocos, i.e., se um bloco tem de ser substituído vamos remover o que foi utilizado menos recentemente.

- Nas bases de dados a estratégia LRU é uma má estratégia. Por exemplo, na junção de duas tabelas (empréstimo |><| cliente) o algoritmo "nested loop join" contém um ciclo interno e um ciclo externo:

```
01 For each tuple in empréstimo
02   For each tuple in cliente
03     If empréstimo.id_cliente=cliente.id_cliente
04       Begin ..... end
05     End for
06 End for
```

- No caso dos blocos com informação do cliente, depois de tratados, podem ser imediatamente removidos. Assim, para as bases de dados, a substituição dos blocos mais recentemente utilizados é a estratégia mais adequada, i.e., MRU (*most recently used*).

Manual: As políticas de substituição dos buffers, página 462 e seguintes.

Critérios de correção:

- 1,0 valores, diferenciar as políticas

- 1,5 valores, justificação da MRU

2. (2,5 valores) Mostre como deriva uma das seguintes expressões relacionais utilizando regras de equivalência para satisfazer a igualdade:

$$\sigma_{\theta_1 \wedge \theta_2} (E1 \bowtie_{\theta_3} E2) = \sigma_{\theta_1} (E1 \bowtie_{\theta_3} (\sigma_{\theta_2} (E2))) \text{ onde } \theta_2 \text{ envolve só atributos de } E2.$$

(Resposta: 1 página)

Resposta:

Utilizando as seguintes regras de equivalência:

Regra A: $\sigma_{\theta_1 \wedge \theta_2} (E) = \sigma_{\theta_1} (\sigma_{\theta_2} (E))$ uma operação de seleção conjuntiva pode ser decomposta por uma sequência de seleções individuais;

Regra B: $\sigma_{\theta_1} (E1 \bowtie_{\theta} E2) = (\sigma_{\theta_1} (E1)) \bowtie_{\theta} E2$ a operação de seleção distribuí sobre uma teta-junção quando todos os atributos de seleção θ_1 envolvem só os atributos da expressão E1.

Podemos encontrar as equivalências com as seguintes justificações:

$$\sigma_{\theta_1 \wedge \theta_2} (E1 \bowtie_{\theta_3} E2) = \text{ dado}$$

$$\sigma_{\theta_1} (\sigma_{\theta_2} (E1 \bowtie_{\theta_3} E2)) = \text{ aplicando a regra A}$$

$$\sigma_{\theta_1} (E1 \bowtie_{\theta_3} (\sigma_{\theta_2} (E2))) = \text{ aplicando a regra B}$$

Manual: Regras de equivalência, página 571 e seguintes.

Critérios de correção:

- 1,0 valores, derivação das equivalências

- 1,5 valores, justificação das regras

3. (2,5 valores) O que entende por ACID num sistema de base de dados transacional? Exemplifique falhas das propriedades de isolamento e durabilidade.

(Resposta: 1 página)

Resposta: ACID é o acrónimo de quatro propriedades: Atomicidade, Consistência, Isolamento e Durabilidade.

(a) Atomicidade - a transação executa todas as ações ou pelo contrário nenhuma delas, isto é, não existem execuções parciais das transações. A ideia de atomicidade está associada à indivisibilidade; as transações são indivisíveis.

(b) Consistência - a transação ao utilizar uma base de dados consistente deve deixá-la consistente no final da mesma transação. Sem o requisito da consistência, as quantidades de dinheiro podiam ser criadas ou destruídas numa transação. A responsabilidade da consistência recai sobre o programador da transação.

(c) Isolamento - a transação, em ambiente de concorrência, devem comportar-se de forma independente, sem qualquer influência das outras transações que são executadas em simultâneo.

Considere duas transações: T1 transfere 50 euros de A para B; T2 transfere 10% de B para A.

Exemplo falha		
	A=100	B=100
0001: T1 writes A=A-50	50	
0002: T2 writes B=B-temp		90
0003: T1 writes B=B+50		140
0004: T2 writes A=A+temp	60	
temp=10% * 100=10	A=60	B=140
Exemplo correcto		
	A=100	B=100
0001: T1 writes A=A-50	50	
0002: T1 writes B=B+50		150
0003: T2 writes B=B-temp		135
0004: T2 writes A=A+temp	65	
temp=10% * 150=15	A=65	B=135

(d) Durabilidade - esta propriedade garante que uma transação que termine com sucesso, todas as alterações se refletem na base de dados de forma persistente, mesmo se existir uma falha do sistema depois de a transação terminar.

Como exemplo de falha de durabilidade, assuma que a transação T transfere 10 euros da conta A para a conta B; depois de executada a transação é enviada a mensagem de sucesso para o utilizador; contudo, os dados dos "buffers" não foram passados para discos; o sistema falha de seguida. Existe falha de durabilidade visto que as alterações foram perdidas e o utilizador ficou convencido que as alterações foram realizadas.

Critérios de correção:

- 1,0 valores, definição das propriedades

- 1,5 valores, exemplos apropriados

4. (2,5 valores) Considere a seguinte sequência do "log" para recuperação de uma base de dados. O que é realizado nas fases de Redo() e Undo().

LSN	LOG
00	Begin checkpoint
10	End checkpoint
20	T1 writes P1
30	T2 writes P2
40	T3 writes P3
50	T2 commit
60	T3 writes P2
70	T2 end
80	T1 writes P5
90	T3 abort
	CRASH, RESTART

(Resposta: 1 página)

Resposta:

O "Algorithms for Recovery and Isolation Exploiting Semantics", ARIES, distingue-se pela utilização de um *Log* com "Log Sequence Number" (LSN), uma "Dirty Page Table" (DPT) e uma "Transaction Table" (TT).

A "Dirty Page Table" (DPT) - contém uma entrada para cada página suja no buffer e o LSN correspondente à primeira actualização dessa página.

"Transaction Table" (TT)- contém uma entrada para cada transação ativa, com informações sobre o ID de transação, o LSN do registo de *log* mais recente para a transação e o status da transação.

No ARIES distinguem-se três fases: análise, *redo* e *undo*.

(0) Na fase prévia de análise, são criadas as DPT e TT. O *log* é varrido desde o *checkpoint* até ao ponto de *crash*. Na DPT é registado o primeiro LSN para cada página e a TT o último LSN para cada transação.

Dirty Page Table		Transaction Table		
Page	first LSN	Trans	last LSN	status
P1	20	T1	80	live
P2	30	T2	70	commit, end
P3	40	T3	60	abort
P5	80			

O ARIES também regista os *rollbacks* das transações através dos "Compensation Log Records" (CLR) i.e. que compensam as alterações de transações incompletas ou abortadas.

Assim no *log* existem comandos de *write*, *commit*, *end*, *abort*, CLR (Compensation Log Record).

(1) O *Redo* inicia com o menor LSN da DPT, LSN=20. O *log* é varrido desde LSN 20 até ao ponto de *crash*, voltando a aplicar as alterações (*writes* ou CLR) do *log*.

LSN 20 Redo write P1
 LSN 30 Redo write P2
 LSN 40 Redo write P3
 LSN 50 No action
 LSN 60 Redo write P2
 LSN 70 No action
 LSN 80 Redo write P5
 LSN 90 No action

(2) O *Undo* é iniciado no maior LSN da TT, LSN=80, o *log* é varrido do fim para o princípio. A instrução de CLR é acrescentada ao *log* para cada ação de desmontagem.

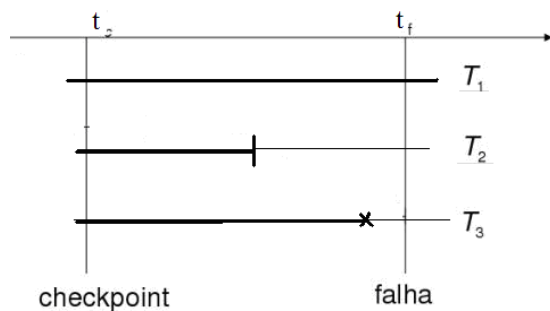
LSN	LOG	prev_LSN
00	begin checkpoint	--
10	end checkpoint	--
20	T1 writes P1	--
30	T2 writes P2	--
40	T3 writes P3	--
50	T2 commit	30
60	T3 writes P2	40
70	T2 end	50
80	T1 writes P5	20
90	T3 abort	60
	crash, restart	
100	CLR T1 LSN 80	80
110	CLR T3 LSN 60	60
120	CLR T3 LSN 40	40
130	T3 end	120
140	CLR T1 LSN 20	20
150	T1 end	140

Em conclusão para cada transação:

T1 fez *rollback* (*redo, undo*)

T2 fez *commit* (*redo*)

T3 fez *rollback* (*redo, undo*)



Critérios de correção:

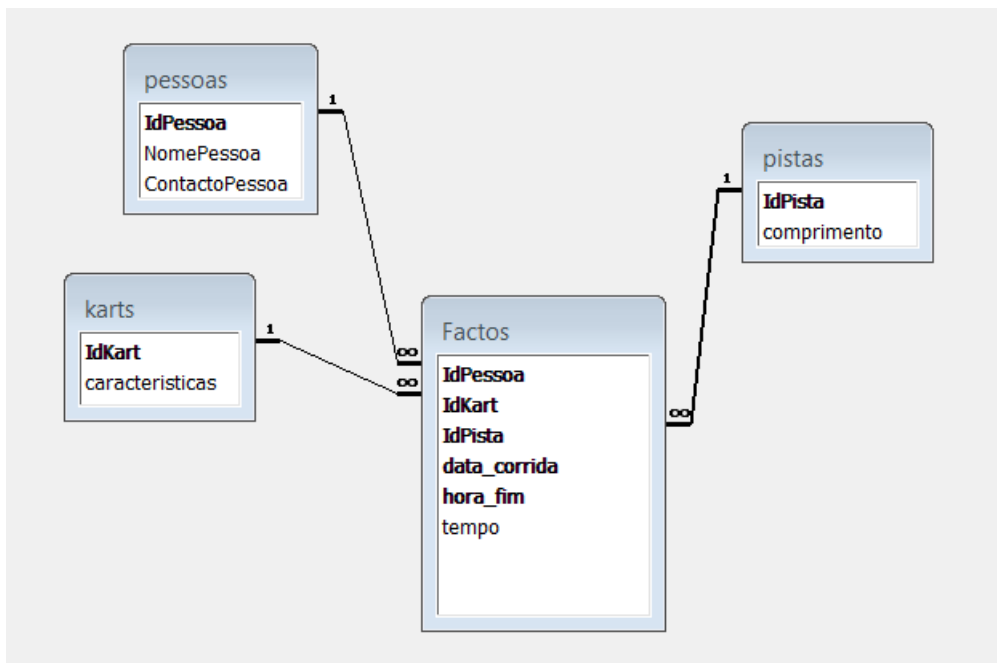
- 1,0 valores, detalhe ARIES-REDO
- 1,0 valores, detalhe ARIES-UNDO
- 0,5 valores, comentários conclusivos

Grupo B – Prática em “Data Warehousing”

5. (2 valores) Pretendemos desenhar um “Data Warehouse” do seguinte sistema. Defina a tabela de factos em primeiro lugar. De seguida, defina três dimensões para o “Data Warehouse” e apresente a tabela de factos associada às três dimensões.

Considere uma base de dados de um clube de karting que possui um kartódromo com várias pistas e uma garagem onde são guardados vários karts. Por razões de segurança (entre outras) cada pessoa apenas pode conduzir certos karts. Assim, pretendemos guardar a informação sobre que karts podem ser conduzidos por que pessoas. Para se saber quem é o melhor condutor, iremos guardar os tempos já obtido por cada pessoa, ao volante de cada um dos karts que ela pode conduzir, em cada uma das pistas onde já o conduziu. Pretende-se saber ainda saber qual é o kart com melhor desempenho.

(Resposta: 1 página)



Critérios de correção:

- criar DW com 1 tabelas de factos e 3 dimensões
- penalização até 50% para esquema mal desenhado
- penalização até 50% atributos desadequados na tabela factos
- penalização até 50% dimensões desadequadas
- penalização até 50% ligações mal estabelecidas

FIM