

Resolução e Critérios de Correção

U.C. 21103

Sistemas de Gestão de Bases de Dados

16 de fevereiro de 2018

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.
- 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 **3** 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: 2.5 2.5 2.5 2.5 2.0

Grupo A – Sistemas de Bases de Dados

1. (2,5 valores) No processamento de consultas considere a junção $R \bowtie S$ com os números de blocos $B_R = B_S = 10.000$. Usando o algoritmo de junção de blocos em ciclo, qual o valor dos blocos em memória M , para o qual não seja necessário realizar mais do que 50.000 leituras. Explique detalhadamente o seu raciocínio.

(Resposta: 1 página)

i) No pior caso de apenas houver espaço em memória para guardar um bloco de R e um bloco de S , teremos o custo de:

$$\#Leituras = B_R + B_R \times B_S$$

ii) Assumindo que R (a tabela/relação mais pequena) cabe na memória com M blocos, e que há um bloco disponível para ler S , obtemos o seguinte algoritmo:

```
Ler todos os blocos  $b_R$  de  $R$  do disco
Para cada bloco  $b_S$  de  $S$ 
  Ler bloco  $b_S$  do disco
  Para cada tuplo  $s$  de  $b_S$ 
    Para cada tuplo  $r$  de todos os  $b_R$ 
      Se os tuplos  $(r, s)$  respeitam a condição:
         $(r, s)$  são adicionados ao resultado
```

iii) Dividimos a memória disponível a seguinte forma: $M-2$ blocos para R , 1 bloco para S e 1 bloco para guardar os resultados parciais da junção.

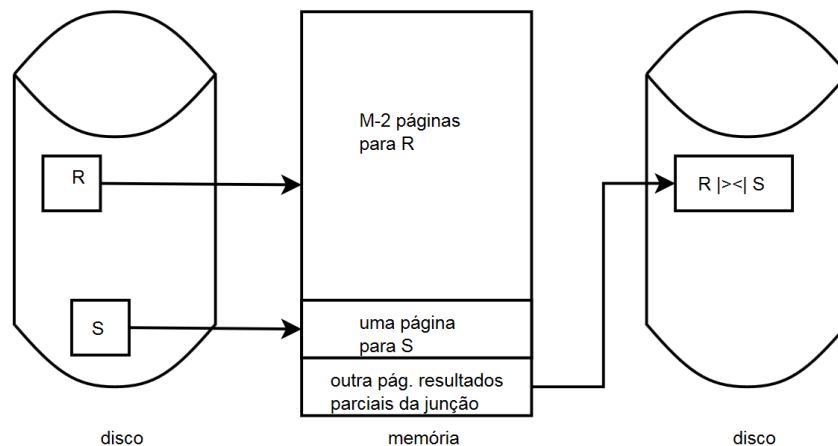


Figura: junção de blocos em ciclo

Como podemos ter simultaneamente $M-2$ blocos de R em memória, terá de ser lida $\lceil B_R / (M-2) \rceil$ vezes e o custo total será:

$$\#Leituras = B_R + (\lceil B_R / (M - 2) \rceil) \times B_S$$

$$50.000 = 10.000 + (\lceil 10.000 / (M - 2) \rceil) \times 10.000$$

logo $M=2.502$

Assim, com 2.502 blocos em memória, não é necessário realizar mais do que 50.000 leituras.

Para este (iii) caso o algoritmo será:

Para cada bloco b_R de R

 Ler os $R / (M-2)$ blocos b_R de do disco

 Para cada bloco b_S de S

 Ler 1 bloco b_S do disco

 Para cada tuplo s de b_S

 Para cada tuplo r de todos os b_R

 Se os tuplos (r, s) respeitam a condição:

(r, s) são adicionados ao resultado

Critério de correção:

- (1,5) fórmula e resultado

- (1,0) explicação de detalhada

- erros, omissões ou redundância: -20% a -100%

2. (2,5 valores) Na otimização de consultas de um SGDB quais as principais técnicas de estimação de resultados?

(Resposta: 1 página)

Resposta:

A escolha de um “bom” plano é essencial na execução de uma consulta SQL, que tem as seguintes fases: análise sintática -> escolha do plano -> execução.

A otimização do plano de execução baseada em custos tem duas tarefas essenciais:

- estimar a cardinalidade do resultado da aplicação de um operador, i.e. o número tuplos (linhas) do resultado;
- escolher a combinação de operadores (seleção, projeção e junção) de menor custo.

As principais técnicas de estimação de resultados de um operador são: amostragem, técnicas paramétricas e histogramas.

- amostragem: obriga a várias leituras, contudo, fornecem geralmente bons resultados
- técnicas paramétricas: obriga que a distribuição dos dados tenha funções conhecidas, ex: Normal (média, desvio padrão), Poisson (lambda)
- histogramas: fornece um resumo dos dados com um grau de aproximação passível de configuração. Existem dois tipos de histogramas: equi-largos e equi-profundos.

Critério de correção:

- (1,0) amostragem, técnicas paramétricas
- (1,5) histogramas
- erros, omissões ou redundância: -20% a -100%

3. (2,5 valores) Defina o protocolo 2-PL. Considere o protocolo 2-PL e explique detalhadamente a execução das seguintes transações.

	T1	T2	Gestão dos 'locks'
1	BEGIN	BEGIN	
2	S-LOCK(A)		
3	R(A)		
4		S-LOCK(A)	
5		R(A)	
6	X-LOCK(A)		
7	W(A)		
8		X-LOCK(A)	
9	UNLOCK(A)		
10	COMMIT		
11		W(A)	
12		UNLOCK(A)	
13		COMMIT	

(Resposta: 1 página)

Resposta:

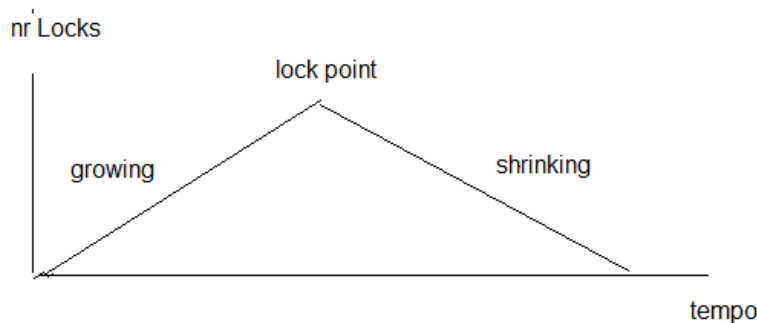
3.a) Defina o protocolo 2-PL.

Existem dois tipos de “locks” para as operações de Read e Write, os partilhados (Shared-lock ou S-lock) e exclusivos (eXclusive-lock ou X-lock).

O protocolo “two-phase locking” (2PL) assegura o controlo da concorrência e garante sequenciamentos conflito - serializáveis. O 2PL é composto, como a própria designação indica, por duas fases:

Fase 1: Fase de Crescimento/Expansão (*growing*)
a transação pode obter *locks* e não pode libertar *locks*

Fase 2: Fase de Encolhimento/Contenção (*shrinking*)
a transação pode libertar *locks* e não pode obter *locks*

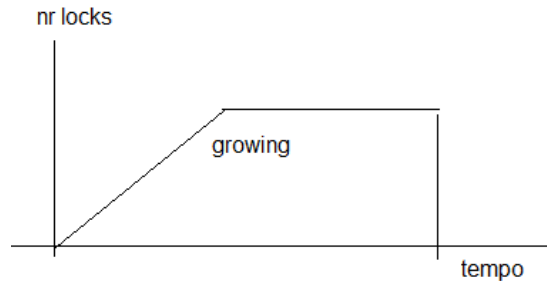


Para a transação T1: R(X), W(X), R(Y), W(Y) teremos um 2PL

T1: S-lock(X), R(X), X-lock(X), W(X), S-lock(Y), R(Y), X-lock(Y), W(Y), Lock-Point, Commit, Unlock(X), Unlock(Y)

O protocolo “two-phase locking”, dado que é muito restritivo, é muito usual a ocorrência de “deadlocks”. No “deadlock” pelo menos uma das transações tem de ser abortada. Se estiverem envolvidas várias transações “cascading rollback (abort)” irá ocorrer.

Para o evitar, pode utilizar uma variante estrita deste protocolo denominado “strict two-phase locking”, onde a transação retém todos os seus *locks* eXclusivos até que efetue *Commit* ou *Abort*.



T1: S-lock(X), R(X), X-lock(X), W(X), S-lock(Y), R(Y), X-lock(Y), W(Y), Lock-Point, Unlock-All, Commit,

3.b) Execução das transações.

A gestão dos *locks* é realizada pelo *Lock Manager* do SGBD. Quando existe um pedido de *lock* verificar o estado do item:

- se não existe *lock* do item, é dada permissão de *lock* (*Grant*)
- caso contrário, existindo um *lock* do item, o pedido fica em fila de espera (*Enqueue*); quando o item é libertado o pedido sai da fila de espera (*Dequeue*)

	T1	T2	Gestão dos locks
1	BEGIN	BEGIN	
2	S-LOCK(A)		Grant (T1, S, A) permite o acesso partilhado da transação T1 ao item A
3	R(A)		
4		S-LOCK(A)	Grant (T2, S, A) permite o acesso partilhado da transação T2 ao item A
5		R(A)	
6	X-LOCK(A)		Grant (T1, X, A) permite o acesso exclusivo da transação T1 ao item A
7	W(A)		
8		X-LOCK(A)	existe lock do item A, pedido vai para fila de espera, Enqueue(A)
9	UNLOCK(A)		lock(A) é libertado
10	COMMIT		
11		W(A)	Dequeue(A), X-lock(A) é permitido
12		UNLOCK(A)	lock(A) é libertado
13		COMMIT	

No instante 8, dado que existe um *lock* do item A, o pedido vai para fila de espera, sendo retirado da fila depois do UNLOCK (A).

Critério de correção:

- (1,0) 3.a) definições
- (1,5) 3.b) execução transações considerando os pedidos em fila de espera
- erros, omissões ou redundância: -20% a -100%

4. (2,5 valores) Explique e exemplifique o objetivo do mecanismo de “checkpoint”. Como é que a frequência do “checkpoint” afeta o desempenho do sistema quando não ocorre a falha? Qual a influência da frequência no tempo que leva para se recuperar de uma falha do sistema? Quantas vezes deve o “checkpoint” ser chamado?

(Resposta: 1 página)

Resposta:

O “checkpoint” (ou ponto de sincronização) é utilizado para reduzir o tempo de recuperação depois de uma falha do SGBD, quando da leitura dos “logs” (registo) do sistema. Se não existissem “checkpoints”, teria de ser feita a leitura de todo o “log” e a operação de “undo” e “redo” das respetivas transações. Se o “checkpoint” for realizado, as operações de escrita das transações passaram para os discos rígidos, sendo uma grande parte do “log” ignorado na fase de recuperação da falha.

Visto que os “checkpoints” degradam o desempenho do SGBD, a sua frequência deve ser reduzida se não for crítica a recuperação rápida. Se for preciso recuperação rápida, a frequência de “checkpoints” deve aumentar. Assim, para as variáveis frequência, desempenho do sistema e tempo de recuperação teremos:

- Sistema fiável (falhas são raras) tem maior intervalo entre checkpoints, com menos degradação do desempenho e uma recuperação mais lenta.
- Sistema menos fiável (falhas são mais frequentes) tem menor intervalo entre checkpoints, com mais degradação do desempenho e uma recuperação mais rápida.

Critério de correção:

- (1,0) definição “checkpoint”
- (1,5) desempenho “checkpoint”
- erros, omissões ou redundância: -20% a -100%

Grupo B – Prática em “Data Warehousing”

5. (2 valores) O principal negócio de uma pizzaria é servir pizzas.
- Cada pizza tem um nome, um tipo de massa (massa fina, massa alta, etc.) e um preço. O preço da pizza é diferente se a pizza for pequena, média, ou grande. Certas pizzas só estão disponíveis em alguns tamanhos e no futuro poderá haver outros tamanhos também (p.ex. familiar). Cada pizza é feita com um conjunto de ingredientes. Cada ingrediente tem um nome e preço.
 - Para além das pizzas, a pizzaria também serve bebidas e sobremesas. Estas são identificadas por um nome. As bebidas têm um nome e preço, enquanto as sobremesas têm um nome, descrição e preço. Além disso, existem vários tipos de bebidas: refrigerantes, cerveja e vinho. Os refrigerantes têm um preço diferente se forem em lata ou garrafa. A cerveja também tem um preço para lata, garrafa e copo. O vinho tem preço para garrafa e 1/2 garrafa.
 - Finalmente, existem menus pré-definidos que incluem uma pizza, uma bebida, e uma sobremesa específicas. Estes menus são conhecidos por um número (Menu 1, Menu 2, etc.) e têm um preço. Os menus são encomendados por um cliente com número de contribuinte, nome, morada e telefone. O cliente também pode encomendar um menu diferente, composto por quaisquer alimentos; nesse caso o preço do menu é calculado com base no preço dos alimentos escolhidos.
 - O gerente da pizzaria pretende ter um registo exato de todas as vendas do seu negócio. Pretende saber o que vendeu a quem vendeu por dia e por hora.

Pretendemos desenhar um “Data Warehouse” do seguinte sistema. Defina as tabelas de factos em primeiro lugar. De seguida, defina três dimensões para cada tabela de factos.

(Resposta: 1 página)

Resposta:

1 tabela de factos (id, data, hora, nr_contribuinte, nome_produto, quantidade) e 3 dimensões: clientes, produtos, tempo

Critérios de correção:

- criar DW com 1 tabelas factos com 3 dimensões
- penalização de 5 décimas na falta da dimensão tempo
- penalização para esquema mal desenhado
- penalização para atributos desadequados na tabela factos
- penalização para dimensões desadequadas
- penalização para ligações mal estabelecidas
- erros, omissões ou redundância: -20% a -100%

FIM