UNIVERSIDADE ABERTA
www.uab.pt

Ministério da Educação e Ciência

p-Fólio

Resolução e Critérios de Correção

U.C. 21103

Sistemas de Gestão de Bases de Dados

12 de julho de 2018

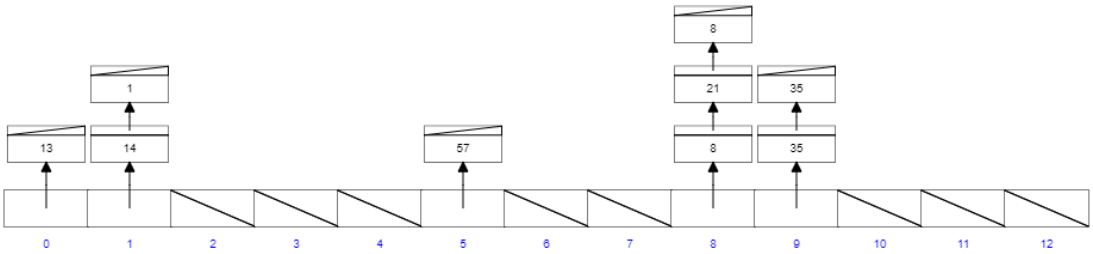
INSTRUÇÕES

- 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
Cotação: 2.5 2.5 2.5 2.5 2.0 valores

Grupo A – Sistemas de Bases de Dados

1. (2,5 valores) Relativamente à indexação de dados considere os algoritmos com 'Hash Tables'. Especifique em pseudo-código o algoritmo apresentado na figura.



(Resposta: 1 página)

Resposta:

```
Function Open_Hashing (int X)
Table_Size =13;
int loc == X % Table_Size;
if (Table[loc]==Empty) Table[loc]=X;
else append(X, Table[loc]);
```

Critério de correção:
(1,0) int loc == X % Table_Size ou equivalente
(1,0) if (Table[loc]==Empty) Table[loc]=X ou equivalente
(0,5) else append(X, Table[loc]) ou equivalente

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

Critério de correção:
(1,0) amostragem
(1,0) técnicas paramétricas
(0,5) histogramas
- penalização se faltarem respetivas definições

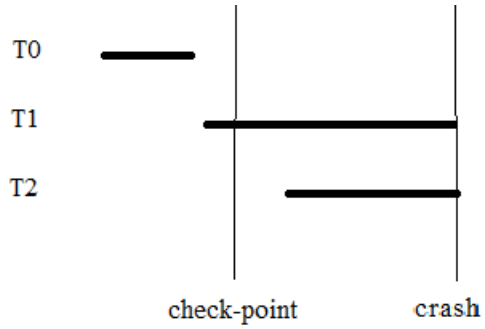
3. (2,5 valores) Na recuperação da falha do seguinte registo (log) que operações se seguem? Explique detalhadamente os passos de Redo e Undo.

<T_0 start>
<T_0, B, 2000, 2050>
<T_0 commit>
<T_1 start>
<T_1, B, 2050, 2100>
<T_1, O_4, operation-begin>
<checkpoint {T_1}>
<T_1, C, 700, 400>
<T_1, O_4, operation-end, (C, +300)>
<T_2 start>
<T_2, O_5, operation-begin>
<T_2, C, 400, 300>

(Resposta: 1 página)

Resposta:

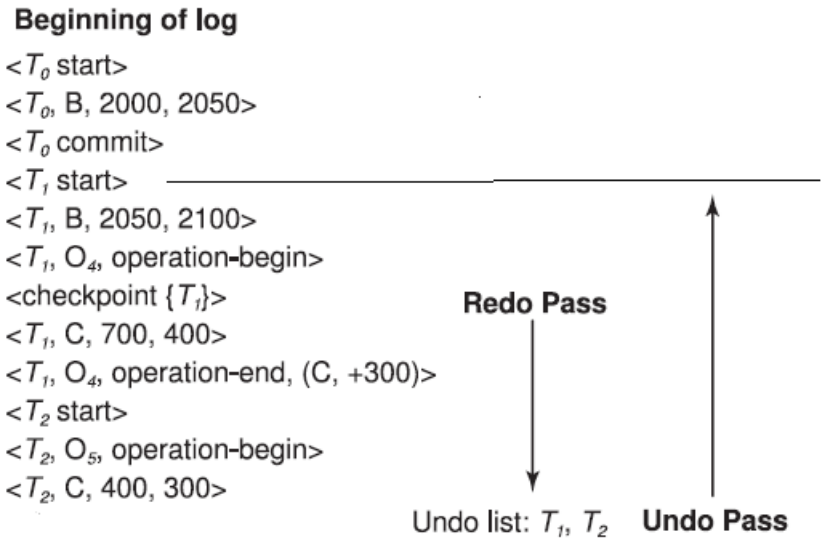
Na recuperação aplicam-se a seguintes regras às transações ativas depois do último *check-point*:
- para todas as transações Tk que não têm registo de <Tk commit> no "log", é executado undo(Tk);
- para todas as transações Tk que têm registo de <Tk commit> no "log", é executado redo(Tk).



T0 terminou antes do *check-point*, tendo toda a informação sido gravada em disco, pelo que nada é preciso fazer.

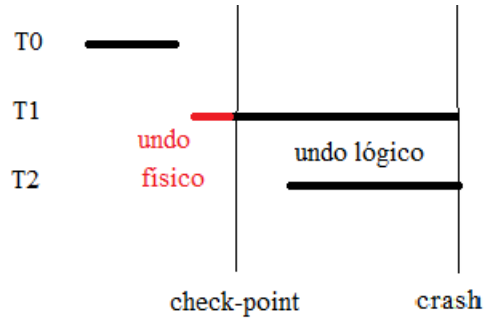
Para as transações T1 e T2 com foram interrompidas pelo *crash*, é necessário proceder ao undo(Tk).

Assim teremos na fase de recuperação da falha:



Fim do Log, Crash
Fase de recuperação são acrescentados os seguintes registos para *undo*(T1) e *undo*(T2)

<T2, C, 400>
<T2 abort>
<T1, C, 400, 700> undo lógico, soma 300 a C
<T1, O4, operation-abort>
<T1, B, 2050> undo físico na DB, repor o valor antes do *chek-point*
<T1, abort>



Para a transação T1 as operações realizadas antes do *check-point*, gravadas em disco, devem set desfeitas fisicamente; as operações depois do *check-point* são desfeitas a o nível lógico.

Critério de correção:
(1,0) explicação geral
(1,5) explicação detalhada diferenciando o undo lógico do físico

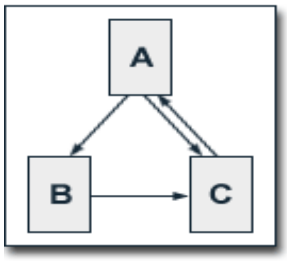
4. (2,5 valores) Em "Information Retrieval" o que entende por algoritmo "PageRank"?
(Resposta: 1 página)

Resposta:
O algoritmo original de PageRank descrito por Lawrence Page and Sergey Brin em 1995 é dado por:

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

onde:
PR(A) é o PageRank da página A,
PR(Ti) é o PageRank das páginas Ti que estão ligadas (apontam) para a página A,
C(Ti) é o número de apontadores ("outbound links") na página Ti
d é o fator de amortecimento que varia em 0 e 1.

Exemplo:



Seja d=0.5,
PR(A) = 0.5 + 0.5 (PR(C) /1)
PR(B) = 0.5 + 0.5 (PR(A) / 2)
PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B) /1)

Resolvendo o sistema de 3 equações e 3 incógnitas obtemos os seguintes PR:
PR(A) = 14/13 = 1.07692308
PR(B) = 10/13 = 0.76923077
PR(C) = 15/13 = 1.15384615

Dada a dimensão da Web, existem métodos iterativos que permitem calcular o PageRank sem recorrer aos sistemas de equações.
Fonte: http://pr.efactory.de/e-pagerank-algorithm.shtml

O PageRank é o algoritmo que permite calcular o "valor" de uma página na Web. O valor da página não depende apenas da quantidade de links apontados para ela, mas do "valor" das páginas que apontam para ela. No exemplo PR(C) depende de PR(A) e PR(B), PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B) /1).

Critérios de Correção:
(1,0) explicação geral
(1,5) exemplo

Grupo B – Prática em "Data Warehousing"

5. (2 valores) Uma livraria deseja manter um registo dos clientes, vendas e 'stock' de livros. Sobre cada cliente, é importante manter a morada com código postal, telefone/telemóvel, 'e-mail' e a lista dos livros que este cliente já comprou. Para cada compra, é importante guardar a data em que esta foi realizada. Um cliente pode comprar muitos livros. Um livro pode ser vendido para mais de um cliente pois geralmente existem vários livros em 'stock'. Um cliente pode ser pessoa física ou empresa. Se for empresa, o seu identificador deve ser o número contribuinte. A livraria compra livros às editoras. Sobre as editoras, a livraria precisa de seu código, morada com código postal, telefone/telemóvel, 'e-mail' e o nome de seu diretor. Cada cliente tem um código único. Deve-se manter um registo sobre cada livro na livraria. Para cada livro, é importante armazenar o nome do autor, assunto, editora, ISBN e a quantidade dos livros em 'stock'. Editoras diferentes não fornecem o mesmo tipo de livro.

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:

Uma tabela de factos (id, data, id-cliente, id_livro, quantidade) e 3 dimensões: clientes, livros, tempo

Critérios de correção:
- criar DW com 1 tabelas factos com 3 dimensões
- 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