



Critérios de Correção

SISTEMAS DE GESTÃO DE BASES DE DADOS | 21103 | ÉPOCA NORMAL

Período de Realização: decorre 26-02-2021 deste 10:00 com 2,5 horas de duração

Data de Limite de Entrega: decorre 26-02-2021 até 12:30 de Portugal Continental

Temática / Tema / Conteúdos: Sistemas de gestão de bases de dados

Objetivos: Reconhecer formas de armazenamento de dados e formas de otimização de consultas; reconhecer o sistema transacional e formas de recuperação de dados; reconhecer ambientes de Data Warehouse, Data Mining e Information Retrieval.

Trabalho a desenvolver: Resolução de um conjunto de exercícios.

Critérios de avaliação e cotação: A cotação deste e-fólio é de 120 pontos = 12 valores, pode encontrar as cotações parciais junto de cada pergunta. A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida. Critérios de avaliação gerais: (i) para a dificuldade de leitura (linhas cruzadas, letras com fontes desadequadas) a penalização é de 20% a 100%; (ii) para erros e omissões a penalização é de 20% a 100%.

Normas a respeitar: Deve redigir o seu E-fólio na Folha de Resolução disponibilizada na turma e preencher todos os dados do cabeçalho. Podem ser incluídas imagens e digitalizações de conteúdos produzido manualmente pelo estudante. Todas as páginas do documento devem ser numeradas. O documento A4 deve ser redigido em Times New Roman, tamanho de letra 12. O espaçamento entre linhas deve corresponder a 1,0 ou 1,5 linhas. Nomeie o ficheiro com o seu número de estudante, seguido da identificação do E-fólio, segundo o exemplo apresentado: 000000efolioGlobal. Finalmente deve gerar um PDF do documento. Deve carregar o referido ficheiro para a plataforma no dispositivo E-fólio Global até à data e hora limite de entrega. Evite a entrega próximo da hora limite para se precaver contra eventuais problemas. O ficheiro a enviar não deve exceder 8 MB. Votos de bom trabalho! Luís Cavique.

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

Questão: 1 2 3 4 5

Cotação: 2 2 2 2 4

Grupo A – Sistemas de Bases de Dados

1. (2 valores) Relativamente ao armazenamento e estrutura de ficheiros, considere o efeito de atomicidade dos blocos num disco RAID nível 5 como o da figura seguinte.

Disk 1	Disk 2	Disk 3	Disk 4
A1	A2	A3	P4
B1	B2	P3	B4
C1	P2	C3	C4
P1	D2	D3	D4

Ocorreu um erro no disco 3. Qual o algoritmo para a recuperação da informação do disco 3? Qual a informação recuperada? Justifique a resposta e exemplifique.

Disk 1	Disk 2	Disk 3	Disk 4
1011 1011	1010 0011		1000 1000
0110 0001	1100 1010		1001 0011
1101 0111	1110 0110		0111 0000
1000 0101	1110 1011		0010 1010

(Resposta: 1 página)

A RAID5 utiliza um algoritmo para recuperação de dados que é baseado em funções XOR entre a informação dos restantes discos para obter a informação perdida.

Para recuperar os dados do Disco 3 temos de fazer as seguintes operações:

$$\begin{aligned}
 A3 &= (A1 \text{ XOR } A2) \text{ XOR } P4 = \\
 &= (1011 \ 1011 \text{ XOR } 1010 \ 0011) \text{ XOR } 1000 \ 1000 = \\
 &= 0001 \ 1000 \text{ XOR } 1000 \ 1000 = 1001 \ 0000
 \end{aligned}$$

Disk 1	Disk 2	Disk 3	Disk 4
1011 1011	1010 0011	1001 0000	1000 1000
0110 0001	1100 1010	0011 1000	1001 0011
1101 0111	1110 0110	0100 0001	0111 0000
1000 0101	1110 1011	0100 0100	0010 1010

Curiosidade: podemos usar a soma aritmética para a operação XOR (a, b, c), substituindo o 2 por 0, e o 3 por 1.

Disk 1	Disk 2	Disk 4	soma Xor
1011.1011	1010.0011	1000.1000	3021.2022
110.0001	1100.1010	1001.0011	2211.1022
1101.0111	1110.0110	111.0000	2322.0221
1000.0101	1110.1011	10.1010	2120.2122

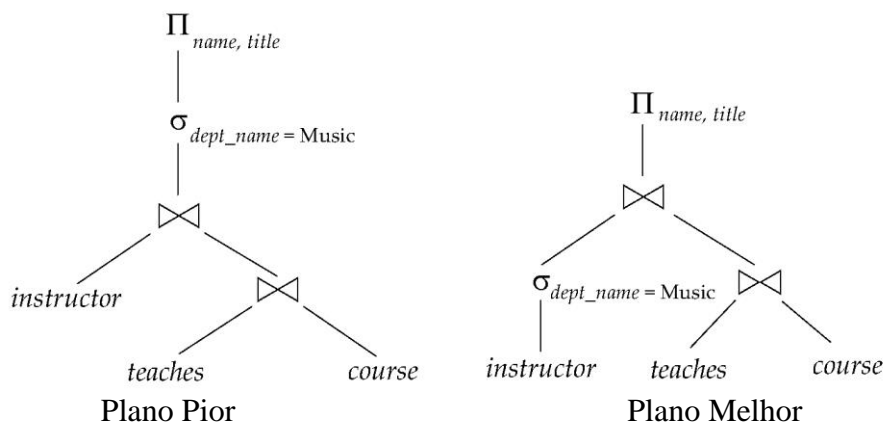
Critério de correção:

- 1,0 valor para o algoritmo recuperação
- 1,0 valor para a informação recuperada
- erros, omissões ou redundâncias: -20% a -100%

2. (2 valores) Na otimização de consultas de um SGDB considere a seguinte consulta. Crie dois planos com qualidade diferente, apresentando as árvores da cada expressão. Calcule o custo de cada um dos planos com base no número de tuplos Nr das tabelas, estimando o número de tuplos distintos $V(A, r)$. Justifique a resposta.

SELECT I.name, I.title FROM course C, teaches T, instructor I WHERE I.dept_name = 'Music' AND C.attribute1 = T.attribute1 AND I.attribute2 = T.attribute2	Course tem 50 tuplos/linhas, Teaches tem 6000 tuplos Instructor tem 100 tuplos Instrutor de música tem 5 tuplos
---	--

(Resposta: 1 página)



São dadas as seguintes dimensões das tabelas:

Tabela	N tuplos
course	50
teaches	6,000
instructor	100
music Instr.	5

Da informação do catálogo sabemos que $V(A, r)$ é o número distinto de tuplos que aparecem na relação r para o atributo A . Este valor é igual a $\Pi_A(r)$. Se A é a chave na relação r , $V(A, r) = n_r$. (pag. 591)

No caso prático, $V(\text{attr2}, \text{course})=50$, visto que é chave da relação. Estimamos que cada 'course' (UC) aparece 4 vezes em 'teaches', $V(\text{attr2}, \text{teaches})= N/4=6000/4=1500$.

Para estimar junção fazemos: custo = $\min(nr*ns/V(A,s), nr*ns/V(A,r))$

Plano Pior				Plano Melhor			
		N	V(attr1)			N	V(attr1)
s	teaches	6,000	1,500	s	teaches	6,000	1,500
r	course	50	50	r	course	50	50
j1	$Nr * Ns / V(A,s)$	200		j1	$Nr * Ns / V(A,s)$	200	
j2	$Nr * Ns / V(A,r)$	6,000		j2	$Nr * Ns / V(A,r)$	6,000	
	Custo = min(j1,j2)	200			Custo = min(j1,j2)	200	
		N	V(attr2)			N	V(attr2)
s	teaches >< course	200	50	s	teaches >< course	200	50
r	instructor	100	100	r	music instructor	5	5
j1	$Nr * Ns / V(A,s)$	400		j1	$Nr * Ns / V(A,s)$	20	
j2	$Nr * Ns / V(A,r)$	200		j2	$Nr * Ns / V(A,r)$	200	
	dimensão = min(j1,j2)	200			dimensão = min(j1,j2)	20	
	Dimensão Plano Pior	400			Dimensão Plano Melhor	220	

Critério de correção:

- 0,5 valor, árvores de planos com qualidade diferente
- 1,5 valor, cálculo do custo dos planos com estimativa e justificação
- erros, omissões, redundâncias: -20% a -100%

3. (2 valores) Relativamente à concorrência em bases de dados:

Considere o protocolo 2-PL e explique detalhadamente a execução das seguintes transações, usando os operadores X-lock(_), S-lock(_) e Unlock(_). Como classifica a concorrência destas duas transações? Justifique a resposta.

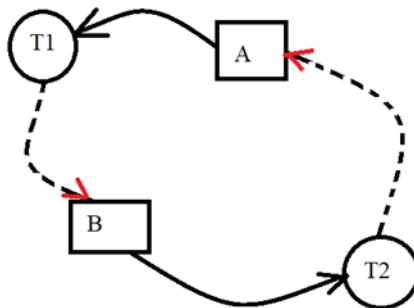
T1	T2
1 Read A	
2	Read A
3 Write A	
4	Read B
5	Write A
6	Write B
7 Read B	
8 Write B	

(Resposta: 1 página)

Ao usar para a transação R(A), W(A) dois fechos, lock-S(A), R(a), lock-X(A), W(A) irá causar ambiguidades desnecessárias, tal como:

T1	T2	Comentários
lock-S(A), R(A)		
	lock-S(A), R(A)	
waiting lock-X(A)		O recurso A tem lock-S de duas transações T1 e T2, não sendo possível T1 fazer lock-X(A).
	waiting lock-X(A)	O recurso A tem lock-S de duas transações T1 e T2, não sendo possível T2 fazer lock-X(A).
		Será que existe 'deadlock'? Claro que não!

O 'deadlock' ocorre quando pelo menos duas transações concorrem para pelo menos dois recursos. T1 tem o recurso A e espera por B, enquanto que T2 tem o recurso B e espera por A.



Desta forma iremos adotar as seguintes regras com um 'lock' por cada recurso:

- Se existe só R(A) na transação, fazer lock-S(A)
- Se existe R(A) e W(A) fazer lock-X (A)

Assim teremos parta T1 e T2:

T1	T2	Comentários
lock-X(A), R(A)		
	lock-X(A) wait	Não é possível T1 fazer X-lock(A), deve aguardar
W(A)		
lock-X(B), R(B)		
W(B)		
Unlock (B)		
Unlock(A)		recurso A é libertado
	lock-X(A), R(A)	
	lock-X(B), R(B)	
	W(A)	
	W(B)	
	Unlock (B)	
	Unlock(A)	

Como prevenção do 'deadlock' utilizou a técnica 'wait-die', T2 requer um recurso de T1, T2 aguardar enquanto T1 termina a transação.

O sequenciamento é serializável e pode ocorrer utilizando o protocolo 2PL, o 'deadlock' não ocorre.

Critério de correção:

- 1,0 valor, tabelas com transações e comentários
- 1,0 valor, resposta final em que o 'deadlock' não ocorre
- erros, omissões, redundâncias: -20% a -100%

4. (2 valores) Considere a seguinte sequência: $r_1(x)$, $w_1(y)$, $w_2(x)$, $w_1(x)$, $r_1(y)$, $r_2(x)$, $w_2(y)$, $w_2(z)$, $w_1(z)$, $w_3(x)$. Desenhe o grafo de precedência e verifique se o escalonamento é serializável a conflitos. Verifique ainda se o escalonamento é serializável a vistas.

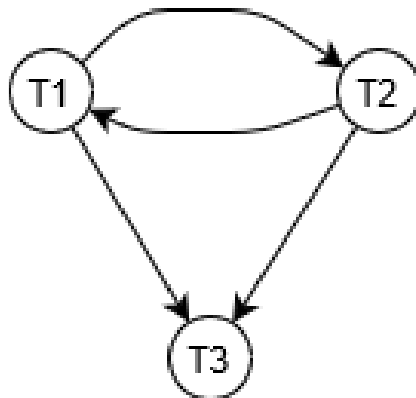
(Resposta: 1 página)

Verificação se o escalonamento é serializável a conflitos

i) dividir por recurso e analisar precedências:

- x) $r_1(x)$, $w_2(x)$, $w_1(x)$, $r_2(x)$, $w_3(x)$: $r_1 \rightarrow w_2$, $r_1 \rightarrow w_3$, $w_2 \rightarrow w_1$, $w_2 \rightarrow w_3$
- y) $w_1(y)$, $r_1(y)$, $w_2(y)$: $r_1 \rightarrow w_2$
- z) $w_2(z)$, $w_1(z)$: $w_2 \rightarrow w_1$

ii) desenhar o grafo



A sequência não é serializável a conflitos, visto existir um ciclo em $T1 \leftrightarrow T2$.

Verificação se o escalonamento é serializável a vistas

O escalonamento é serializável a vistas se existirem dois escalonamentos equivalentes.

Dois escalonamentos S_1 e S_2 são equivalentes se:

- (1) Todos os elementos A têm o mesmo valor inicial
Se S_1 : $r_i(A)$ lê o valor inicial, então S_2 : $r_i(A)$ também lê o mesmo valor inicial
- (2) Um elemento A tem a mesma vista nos dois escalonamentos
Se S_1 : $w_j(A) \Rightarrow r_i(A)$ então em S_2 : $w_j(A) \Rightarrow r_i(A)$; têm a mesma vista
- (3) Todos os elementos A têm a mesma vista final
Se S_1 : T_i termina $w_i(A)$, então S_2 : T_i também termina com $w_i(A)$; os valores finais são os mesmos.

Para as transações T1, T2 e T3, vamos analisar as condições para os escalonamentos serem equivalentes:

T1	T2	T3			x	y	z
r1(x)				inicial R	T1, T2	T1	-
w1(y)				update RW			
	w2(x)			final W	T3	T2	T1, T2
w1(x)					T2->T3	T1->T2	
r1(y)					T1->T3		
	r2(x)						
	w2(y)						
	w2(z)						
w1(z)							
		w3(x)					

Para o recurso x, encontramos as sequências T2->T3 e T1->T3 existindo ainda uma escrita-cega ('blind-write') w2(x).

Para o recurso y encontramos a sequência T1->T2 com uma escrita-cega adicional w1(y).

Para o recurso z existem duas escritas possíveis w1(z) e w2(z) que permitem dois escalonamentos com vistas diferentes.

Assim, o escalonamento não é serializável a vistas.

Critério de correção:

- 1,0 valores, verificação de é serializável a conflitos
- 1,0 valores, verificação de é serializável a vista
- erros, omissões, redundâncias: -20% a -100%

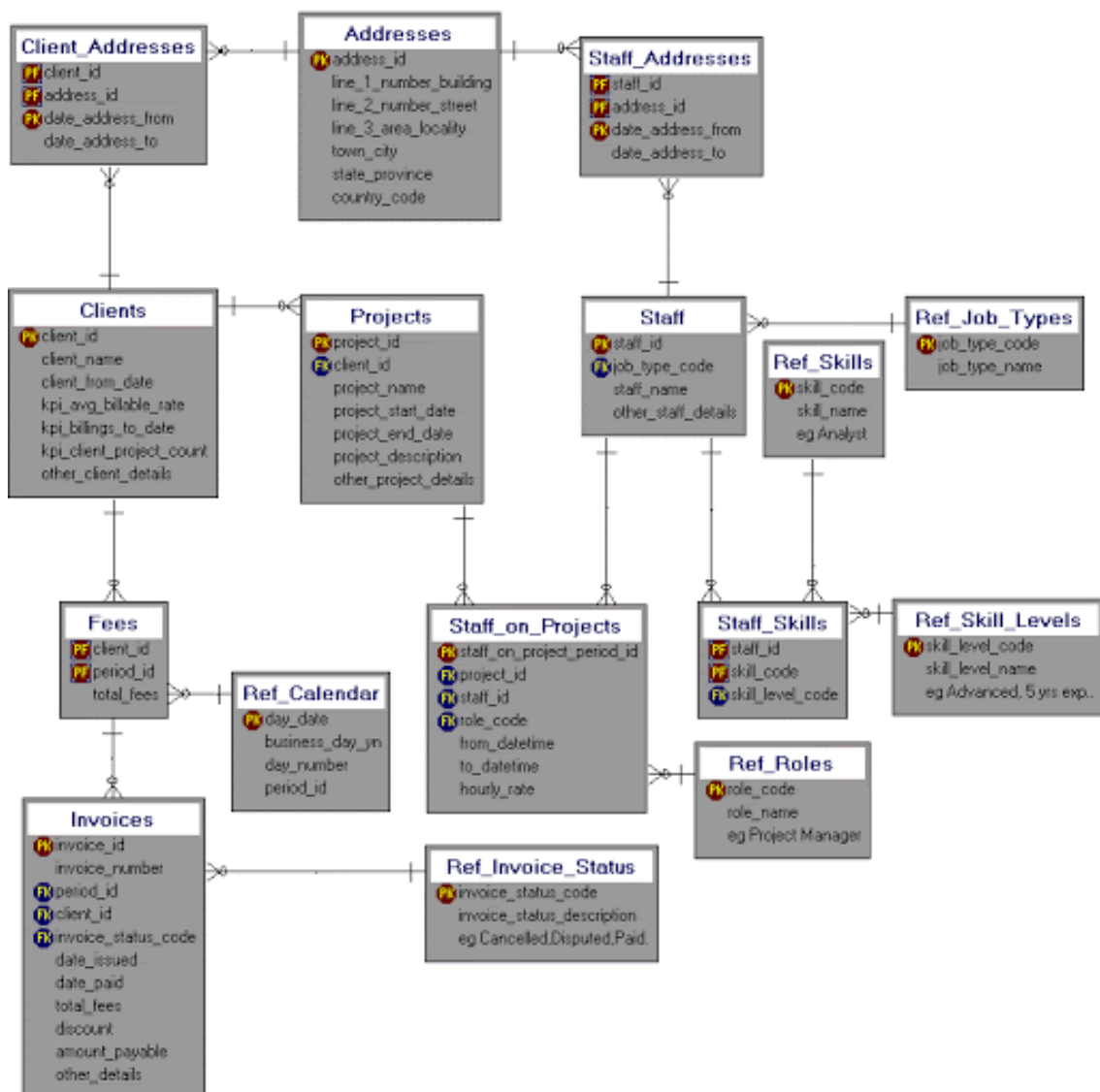
Grupo B – Prática em “Data Warehousing”

5. (4 valores) Considere a seguinte base de dados que vai servir de fonte de dados a um “Data Warehouse”.

a) Defina as tabelas de auxiliares (‘lookup’), intermédias e de factos da base de dados em primeiro lugar. Para as tabelas de factos defina os atributos aditivos, semi-aditivos ou não-aditivos.

b) De seguida, defina um DW, em estrela ou constelação, com pelo menos três dimensões para cada tabela factos.

(Resposta: 1 página)

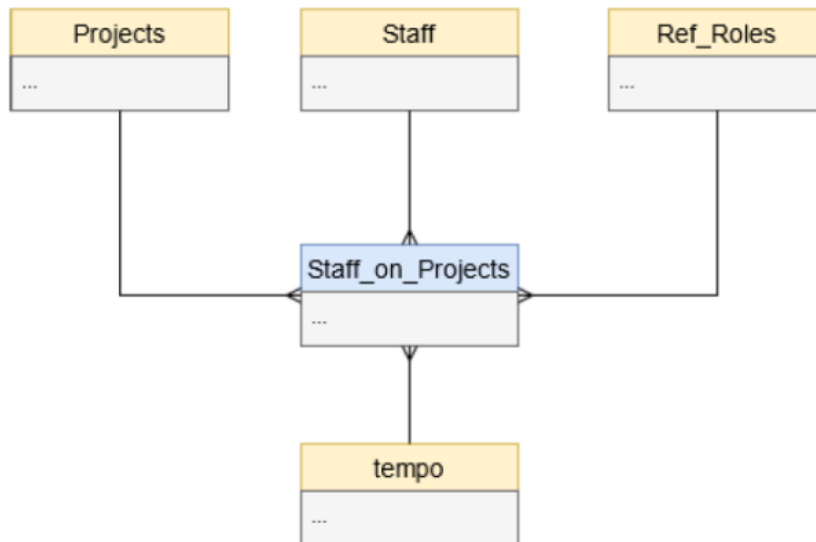
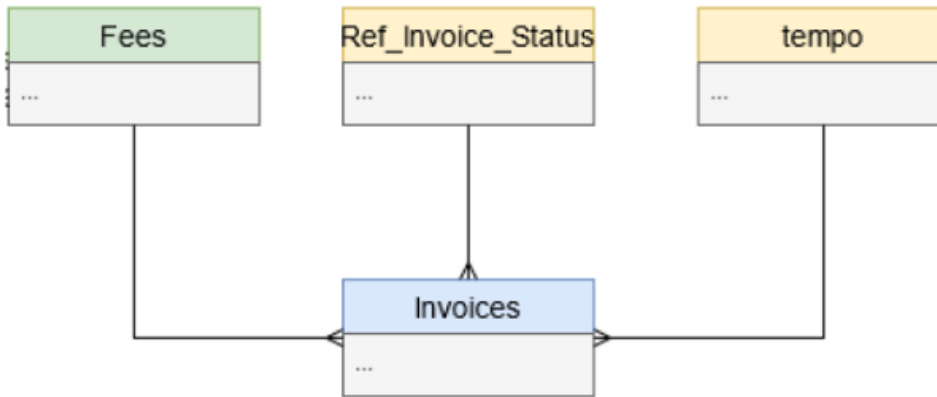


5.a) Primeiro vamos definir para as tabelas de factos defina os atributos aditivos, semi-aditivos, não-aditivos e sem factos.

- Aditivos: são atributos que podem ser agregados (somados) por todas as dimensões, ex: valor da venda (usar Sum() sempre)
- Semi-aditivos: são atributos que podem ser agregados (somados) por algumas as dimensões, ex: quantidade (usar Sum() em condições particulares)
- Não-aditivos: são atributos que não podem ser agregados (somados), ex: preço unitário (usar Average() por exemplo)
- Sem factos: só existem identificadores (usar a função Count() dos identificadores)

Table	type of table	type_facts
Addresses	lookup	
Clients	lookup	
Ref_Calendar	lookup	
Ref_Invoice_Status	lookup	
Ref_Job_Types	lookup	
Ref_Roles	lookup	
Ref_Skill_Levels	lookup	
Ref_Skills	lookup	
Fees	intermediate	
Projects	intermediate	
Staff	intermediate	
Client_Addresses	facts	no facts
Invoices	facts	additive (amount_payable)
Staff_Addresses	facts	no facts
Staff_on_Projects	facts	non-additive (hourly rate)
Staff_Skills	facts	no facts

5.b) Constelações de estrelas com factos aditivos, semi-aditivos ou não-aditivos:



Critério de correção:

- a) 2,0 valores, 3 tipos tabelas e tipos de factos
- b) 2,0 valores, tabelas de factos e dimensões
- erros, omissões, redundâncias: -20% a -100%

FIM