

# ACH 2025 – Laboratório de Banco de Dados

**José de J. Pérez-Alcázar**

## Prova 2 – Turma 04

1) O Clube Globetrotters é organizado em capítulos. O presidente de um capítulo nunca pode servir como o presidente de qualquer outro capítulo, e cada capítulo dá ao seu presidente algum salário. Capítulos continuam se movendo para novos locais e um novo presidente é eleito quando (e somente quando) um capítulo se move. Esses dados são armazenados em uma relação  $G(C, S, L, P)$ , onde os atributos são capítulos (C), salários (S), localizações (L) e presidentes (P). A consulta a seguir é feita com frequência e você deve ser capaz de respondê-la eficientemente: “Quem era o presidente do capítulo X quando estava no local Y?”. Dadas as seguintes dependências funcionais:  $PC \rightarrow S$ ;  $CL \rightarrow P$ ; e  $P \rightarrow C$  (data a primeira e terceira dependência, podemos deduzir que  $P \rightarrow S$ ) 30%

- a. Defina quais são as chaves candidatas para a relação G?
- b. Em qual forma normal está a relação G? Crie um esquema de banco de dados normalizado para o clube. Justifique as suas decisões.
- c. Escreva a consulta descrita acima em SQL.
- d. Quais índices (supondo que pode definir índices de hash e agrupados) e de que tipo você criaria para melhorar a sua consulta? Justifique
- e. Se os índices não são suficientes para melhorar o desempenho da consulta o que você faria?

2) A Figura abaixo, mostra o log correspondente a determinado plano, para quatro transações T1, T2, T3, T4, no ponto da queda do sistema. Suponha que usemos o protocolo de atualização **imediate** com checkpoint. Descreva o processo para recuperação da queda do sistema. Especifique quais transações serão revertidas, quais operações do log serão refeitas e quais (se houver) serão desfeitas, e se poderá ocorrer alguma reversão em cascata. 20%

```
[start_transaction, T1]
[read_item, T1, A]
[write_item, T1, D, 20, 25]
[commit, T1]
[checkpoint]
[start_transaction, T2]
[read_item, T2, B]
[write_item, T2, B, 12, 18]
[start_transaction, T4]
[read_item, T4, D]
[write_item, T4, D, 25, 15]
[start_transaction, T3]
[write_item, T3, C, 30, 40]
[read_item, T4, A]
[write_item, T4, A, 30, 20]
[commit, T4]
[read_item, T2, D]
[write_item, T2, D, 15, 25] ← system crash
```

3) Assinale como verdadeiro e falso as alternativas a seguir (duas erradas anulam uma certa) 50%

- a) No caso da transação falhar sob um esquema de log incremental de atualização adiada, qual das seguintes opções será necessária:
- I. Uma operação desfazer ()
  - II. Uma operação refazer ()
  - III. Uma operação desfazer e refazer ()
  - IV. Nenhuma acima. ()

b) No controle de concorrência baseado em Timestamp:

- I. Ocorrem deadlocks ( )
- II. Se uma transação  $T_i$  tenta executar um  $\text{ler\_item}(X)$  e  $\text{write\_TS}(X) > \text{TS}(T_i)$ , então  $T_i$  lê o item  $X$  ( )
- III. Se uma transação  $T_i$  tenta executar um  $\text{escrever\_item}(X)$  e  $\text{write\_TS}(X) > \text{TS}(T_i)$ , então  $T_i$  aborta ( )
- IV. Pode-se ter a opção de controle de concorrência multi-versão ( )

c) As técnicas de controle de concorrência otimistas:

- I. São conhecidas como de validação ou certificação ( )
- II. Executam alguma ação de verificação antes que uma operação de banco de dados possa ser executada ( )
- III. Assumem que ocorrerá muita interferência em transações ( )
- IV. As atualizações são aplicadas inicialmente a versões dos dados. ( )

d) Se uma abordagem de shadowing for usada:

- I. Para gravar um item de dado de volta no disco, ele será gravado na mesma localização do disco ( )
- II. O mecanismo de controle de concorrência que melhor se adapta é o de multi-versão ( )
- III. Ela é caracterizada como uma técnica NO-UNDO/REDO ( )
- IV. O estado do BD antes da execução da transação está totalmente disponível no catálogo Shadow. ( )

e) Sobre o artigo “Aplicando técnicas de tuning para melhoria de desempenho em bancos de dados PostgreSQL”:

- I. PostgreSQL permite os seguintes métodos de indexação: Btree, Hash, R-tree e o GIST ( )
- II. PostgreSQL usa índice em consultas que tenham LIKE e começam com um caractere ‘%’ ou ‘\_’ ( )
- III. Em PostgreSQL, o comando EXPLAIN é responsável por mostrar o plano de execução que o otimizador gera ( )
- IV. Em PostgreSQL, o administrador pode dirigir a escolha dos algoritmos utilizados pelo otimizador com comandos como `enable_nestloop` ( )