

Trabalho Prático - LOG

(29/11)

Objetivo: implementar o mecanismo de log Redo/Undo com checkpoint **usando o SGBD**

Funcionamento:

O código, que poderá utilizar qualquer linguagem de programação, deverá ser capaz de ler o arquivo de log (entradaLog) e o arquivo de Metadado e validar as informações no banco de dados através do modelo REDO/UNDO.

O código receberá como entrada o arquivo de metadados (dados salvos) e os dados da tabela que irá operar no banco de dados.

Exemplo de tabela do banco de dados:

Id	A	B
1	100	20
2	20	30

Arquivo de log no formato <transação, "id da tupla", "coluna", "valor antigo", "valor novo">. Exemplo:

Arquivo de Metadado:

```
[ { "att": "A", "row"=1, "value"=100 },  
  { "att": "A", "row"=2, "value"=20 },  
  { "att": "B", "row"=1, "value"=20 },  
  { "att": "B", "row"=2, "value"=30 }]
```

Arquivo de Log:

<start T1>
<T1,1, A,20,30>

<start T2>

<commit T1>

<CKPT (T2)>

<T2,2, A,50>

<start T3>

<CKPT>

<start T4>

<T4,1, A,100>

<commit T4>

Saída:

“Transação T2 **não** realizou Redo”

“Transação T3 **não** realizou Redo”

“Transação T4 realizou Redo”

Imprima as variáveis, exemplo:

```
[ { "att": "A", "row"= , "value"=100 },  
  { "att": "A", "row"= , "value"=20 },  
  { "att": "B", "row"= , "value"=20 },  
  { "att": "B", "row"= , "value"=30 }]
```

O checkpoint Redo permite que parte do log já processada seja descartada para evitar o reproprocessamento.

Detalhes:

Funções a serem implementadas:

- 1- carregar o banco de dados com a tabela antes de executar o código do log (para zerar as configurações e dados parciais);
- 2- Carregar o arquivo de log;
- 3- Verifique quais transações devem realizar REDO/UNDO. Imprimir o nome das transações que irão sofrer Redo. Observem a questão do **checkpoint**;
- 4- Checar quais valores estão salvos nas tabelas (com o select) e atualizar valores inconsistentes (update);
- 5- Reportar quais dados foram atualizados;
- 6- Seguir o fluxo de execução conforme o método de REDO, conforme visto em aula;

Execução:

- Pode ser implementado em duplas;
- A nota será individual;
- Deve ser enviado o repositório no GIT (será avaliado a participação dos membros através dos commits). Um único commit com o código pronto será entendido como uma cópia e receberá nota zero. Os commits irão interferir na nota final dos membros da dupla;
- Poderá ser testado com outro arquivo de log a execução do programa;