

Área do Conhecimento de Ciências Exatas

Curso: Bacharelado em Ciência da Computação

Disciplina: Aspectos de Implementação de Banco de Dados - INF0227A

Docente: Helena Graziottin Ribeiro

Alunos: Bruno Miguel Caregnato e Leopoldo Corbellini Reginato

# Simulação de escalonador de transações

# 

**Descrição**

Foi desenvolvido um programa para simular um escalonador de transações para um SGBD, utilizando o protocolo 2PL básico. O trabalho foi desenvolvido como uma aplicação para Windows através Visual Studio 2019 na linguagem C#, com a necessidade de instalação do Runtime do .Net Framework 4.5.

A entrada do algoritmo é uma história seguindo o padrão: **<comando><número da transação>"["<dado>"]"**, utilizando "-" como separador entre cada comando. Para o commit é aceito o padrão: **"c"<número da transação>**. Segue abaixo um exemplo de uma história completa válida:

**w1[x]-r2[y]-w2[y]-c2-w1[y]-c1**

A história deve ser informada corretamente, pois não foi implementada uma solução para validar a sintaxe da história.

Descrição das tags:

* <comando>: deve ser "r": Read ou "w": Write.
* <número da Transação>: número de identificação da transação. Tendo limite de 9 transações.
* <dado>: dado sendo representado por uma letra.

O algoritmo foi desenvolvido para executar todos os comandos de todas as transações, distribuindo os locks para cada dado utilizado em cada transação, tratando dead locks.

No protocolo 2PL, após a transação realizar o unlock de um dado (para poder realizar o unlock, deve ter sido realizado o lock antes) não é possível realizar o lock novamente. Desta forma o algoritmo realiza o lock do dado no momento que a transação fará a leitura ou escrita, então ele realiza a operação necessária e permanece com o lock do dado até a transação realizar o commit, podendo dar unlock do dado tanto antes do commit como depois.

Várias transações podem realizar lock shared na leitura, mas quando uma transação realiza uma escrita esta transação exige o lock exclusive. O algoritmo definido para tratar os dead locks analisa todas as transações que estão em estado de espera, estas sendo consideradas em um ciclo. Quando um ciclo é detectado o programa irá finalizar a transação que realizou menos comandos até o momento do dead lock.

**Execução**

A execução baseia-se em primeiramente informar a história, seguindo o modelo explicado anteriormente. Após, deve-se definir um tempo n (segundos) que é utilizado para definir de quantos em quantos segundos serão executados os comandos. Então o programa irá retornar as informações como: fila de execução, fila de espera, transações em espera, transações abortadas e dados bloqueados. Conforme um exemplo de imagem abaixo: