

## Especificação do Trabalho

O segundo trabalho da disciplina de Banco de Dados I consiste na implementação de um algoritmo para verificação se um escalonamento (*schedule*) é serial. Este trabalho tem por objetivo auxiliar os(as) alunos(as) na compreensão dos gargalos e protocolos (ou regras) existentes no processamento de transações. Cada grupo pode ser composto por até 2 alunos(as) e deve utilizar um dos sistemas de gerenciamento de banco de dados utilizados na disciplina: MySQL versão 5.7 ou PostgreSQL 10.

### Teste de *schedule* serial

Como o nome diz, em um *schedule* serial todas as transações são executadas em série uma após a outra. Em outras palavras, uma transação não inicia até que a transação atualmente em execução termine. Esse tipo de comportamento também é conhecido como execução não intercalada. *Schedules* seriais são sempre recuperáveis, livres de cascata, rigorosos e consistentes. Uma programação serial sempre fornece o resultado correto.

Como ilustração, considere os 2 *schedules* mostrados abaixo, que realizam algumas operações. Se houver intercalação de operações de transações distintas, o escalonamento é não serial (Exemplo 01). Já se não houver intercalação de operações, como no Exemplo 02, em que todas as operações T3 são seguidas por todas as operações T4, o *schedule* é considerado serial. O algoritmo de teste de *schedule* serial deve possuir a assinatura descrita nos arquivos em anexo.

### Entrada:

Considere a existência da tabela **Schedule**, na qual cada linha representa a chegada de uma operação pertencente à transação. A tabela possui 4 colunas: a primeira representa o tempo de chegada (*time*), a segunda o identificador da transação (*#t*), a terceira a operação (*R* : leitura, *W* : escrita, *C* : confirmação ou *A* : aborto) e a quarta o atributo que será lido/escrito (quando aplicável). As linhas da tabela estão ordenadas logicamente pelo valor na primeira coluna, que indica o carimbo de tempo (*timestamp*) de chegada (quanto menor o valor, mais antiga a operação).

### Saída:

A saída deve ser 1, se um dado escalonamento for serial, e 0, caso contrário.

### Exemplos

Exemplo 01 (escalonamento não serial)

time	#t	op	attr
1	1	R	X
2	2	R	X
3	2	W	X
4	1	W	X
5	2	C	-
6	1	C	-

Saída Exemplo 01

0

Exemplo 02 (escalonamento serial)

time	#t	op	attr
7	3	R	X
8	3	R	Y
9	3	W	Y
10	3	C	-
11	4	R	X
12	4	C	-

Saída Exemplo 02

1

## **Submissão**

Cada grupo deve entregar um único arquivo *.sql* contendo o algoritmo, instruções básicas para execução, código comentado e testes realizados. A submissão deve ser realizada pelo Google Classroom (o professor criará uma atividade especificamente para isso).

## **Prazos**

Os prazos de entrega do trabalho podem ser consultados no calendário da página do curso no Google Classroom.

## **Avaliação**

O trabalho será avaliado com base nos seguinte critérios:

- Funciona?
- Cumpre os requisitos?
- Qualidade da solução proposta (modelagem, desempenho, etc.)
- Boas práticas (estrutura e organização do código, documentação, etc.)
- Testes

## **Observações finais**

Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas nas aulas. É responsabilidade do(a) aluno(a) manter-se informado(a), frequentando as aulas ou acompanhando as novidades através das ferramentas utilizadas na disciplina.