

Lista de Exercícios II (Processamento de Transações)

1. Discuta as diferentes medidas de equivalência dos escalonamentos. Qual a diferença entre equivalência de conflito e equivalência de visão?
2. O que é um escalonamento serial? O que é um escalonamento serializável? Por que um escalonamento serial é considerado correto? Por que um escalonamento serializável é considerado correto?
3. Discuta como serialização é utilizada para garantir o controle de concorrência em um sistema de BD.
4. Por que a serialização é considerada, algumas vezes, muito restritiva como medida de correção de escalonamento?
5. Considere as seguintes transações:

| T <sub>1</sub> | T <sub>2</sub> |
|----------------|----------------|
| read_item (x)  | read_item (x)  |
| x := x + 10    | x := x - 10    |
| write_item (x) | write_item (x) |

- a) liste todos os possíveis escalonamentos para estas transações;
- b) quais escalonamentos são serializáveis de acordo com a equivalência de conflito?

6. Considere as seguintes transações:

| T1             | T2             | T3             |
|----------------|----------------|----------------|
| read_item (x)  | read_item (z)  | read_item (y)  |
| write_item (x) | read_item (y)  | read_item (z)  |
| read_item (y)  | write_item (y) | write_item (y) |
| write_item (y) | read_item (x)  | write_item (z) |

- (a) quantos escalonamentos seriais existem para estas três transações?
- (b) escreva um procedimento que gere todos os possíveis escalonamentos seriais para estas três transações

7. Determine se os seguintes escalonamentos são serializáveis de acordo com a equivalência de conflito:

| T <sub>1</sub>   | T <sub>2</sub>  |
|--|---|
| read_item (x)<br>x = x - 50<br>write_item (x)<br><br><br><br>read_item (y)<br>y = y + 50<br>write_item (y) | <br><br>read_item (x)<br>tempo = x * 0.1<br>x = x - tempo<br>write_item (x)<br><br><br><br>read_item (y)<br>y = y + tempo<br>write_item (y) |

| T <sub>1</sub>   | T <sub>2</sub>  |
|--|---|
| read_item (x)<br>x = x - 50<br><br><br><br>write_item (x)<br>read_item (y)<br>y = y + 50<br>write_item (y) | <br><br>read_item (x)<br>tempo = x * 0.1<br>x = x - tempo<br>write_item (x)<br>read_item (y)<br><br><br><br>y = y + tempo<br>write_item (y) |

| T <sub>1</sub>   | T <sub>2</sub>   |
|--|--|
| read_item (a)<br>write_item (a)<br><br>read_item (b)<br><br>write_item (b) | <br><br>read_item (a)<br><br>write_item (a)<br><br><br><br>read_item (b)<br>write_item (b) |

| T <sub>1</sub>                      | T <sub>2</sub>         |
|-------------------------------------|------------------------|
| read_item (q)<br><br>write_item (q) | <br><br>write_item (q) |

| T <sub>1</sub>   | T <sub>2</sub>   |
|--|--|
| read_item (a)<br>a = a – 50<br>write_item (a)<br><br>read_item (b)<br>b = b + 50<br>write_item (b) | read_item (b)<br>b = b – 10<br>write_item (b)<br><br>read_item (a)<br>a = a + 10<br>write_item (a) |

8. Considere as seguintes transações:

| T <sub>1</sub>   | T <sub>2</sub>   |
|--|--|
| read_item (a)<br>read_item (b)<br>if a = 0<br>then b = b + 1<br>write_item (b) | read_item (b)<br>read_item (y)<br>if b = 0<br>then a = a + 1<br>write_item (a) |

- mostre que toda execução serial envolvendo essas duas transações preserva a consistência do BD;
- mostre uma execução concorrente de T<sub>1</sub> e T<sub>2</sub> que produza um escalonamento não-serializável (de acordo com a equivalência de conflito);
- mostre uma execução concorrente de T<sub>1</sub> e T<sub>2</sub> que produza um escalonamento não-serializável (de acordo com a equivalência de visão);
- existe alguma execução concorrente de T<sub>1</sub> e T<sub>2</sub> que produza um escalonamento serializável (de acordo com a equivalência de conflito) ?