**A. Jawablah dengan singkat !**

**Transaction Processing**

1. Jika diketahui suatu Serial Schedule (S) sebagai berikut :

|  |  |
| --- | --- |
| **T1** | **T2** |
| Read(A) |  |
| Write(A) |  |
| Read(B) |  |
| Write(B) |  |
|  | Read(B) |
|  | Write(B) |
|  | Read(A) |
|  | Write(A) |

Jika **B tidak pernah terpengaruh dari nilai A** untuk setiap transaksi, buatlah suatu Schedule lain (S’) **yang ekivalen dengan Schedule S!**

**Jawab:**

|  |  |
| --- | --- |
| **T1** | **T2** |
| Read(A) |  |
| Write(A) |  |
|  | Read(A) |
|  | Write(A) |
| Read(B) |  |
| Write(B) |  |
|  | Read(B) |
|  | Write(B) |

**Concurency Control**

1. Diketahui Schedule S sebagai berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T1** | **T2** | **T3** | **T4** | **T5** |
|  | Read(Y) |  |  |  |
| Read (Z) |  |  |  |  |
|  |  |  |  | Write(W) |
|  |  | Read(Z) |  |  |
|  |  | Write(Z) |  |  |
|  |  | Write(Q) |  |  |
|  |  | Read(Q) |  |  |
|  |  |  | Read(Q) |  |
|  |  |  | Write(X) |  |
|  | Read(X) |  |  |  |
|  |  |  |  | Write(X) |
| Write(X) |  |  |  |  |
|  |  | Read(A) |  |  |
|  | Write(Y) |  |  |  |

1. Buatlah Precedence Graph dari Schedule diatas.

**Jawab:**

1. Apakah Schedule tersebut Conflict Serializable?

**Jawab:**

**Tidak terjadi Conflict Seriazable, karena terjadi circular pada presedence graph.**

**BackUp dan Recovery**

1. Terdapat tiga transaksi yang bekerja secara konkurensi seperti gambar di bawah ini. Mereka melakukan operasi pengurangan atau penambahan terhadap account a, account b, dan account c.

|  |  |  |
| --- | --- | --- |
| **Transaksi 1** | **Transaksi 2** | **Transaksi 3** |
| sum:=0; | read value c | read value b |
| read value a | c := c - 10; | b := b + 5 |
| a := a – 5 | write value c | write value b |
| sum := sum + a; | read value a |  |
| write value a | a := a + 10 |  |
| read value c | write value a |  |
| sum := sum + c; | Commit |  |
| read value b |  |  |
| sum := sum + b; |  |  |
| write value b |  |  |

1. Isilah isi dari tiap-tiap variable pada log variable pada kolom yang tersedia dibawah ini, bila diasumsikan proses yang berlaku adalah **immediately update** (initial value a=10, b=10, c=10).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Transaksi 1 | Transaksi 2 | Transaksi 3 | variable (buffer) | | | |
| a | B | C | Sum |
| 1 | sum:=0; |  |  | - | - | - | 0 |
| 2 | read value a |  |  | 10 | - | - | 0 |
| 3 | a := a - 5 |  |  | 5 | - | - | 0 |
| 4 | sum := sum + a; |  |  | 5 | - | - | 5 |
| 5 | write value a |  |  | 5 | - | - | 5 |
| 6 |  | read value c |  | 5 | - | 10 | 5 |
| 7 |  | c := c - 10; |  | 5 | - | 0 | 5 |
| 8 |  | write value c |  | 5 | - | 0 | 5 |
| 9 |  | read value a |  | 5 | - | 0 | 5 |
| 10 |  | a := a + 10 |  | 15 | - | 0 | 5 |
| 11 |  | write value a |  | 15 | - | 0 | 5 |
| 12 |  | commit |  | 15 | - | 0 | 5 |
| 13 | read value c |  |  | 15 | - | 0 | 5 |
| 14 | sum := sum + c; |  |  | 15 | - | 0 | 5 |
| 15 |  |  | read value b | 15 | 10 | 0 | 5 |
| 16 |  |  | b := b + 5 | 15 | 15 | 0 | 5 |
| 17 |  |  | write value b | 15 | 15 | 0 | 5 |
| 18 | read value b |  |  | 15 | 15 | 0 | 5 |
| 19 | sum := sum + b; |  |  | 15 | 15 | 0 | 20 |
| 20 | failure |  |  |  |  |  |  |

1. Buatlah log entry (buffer) pada tabel dibawah ini untuk keperluan recovery database bila terjadi kesalahan (**immediately update**).

**Jawab:**

setiap terjadi modifikasi langsung disimpan, menggunakan konsep undo dan redo ketika terjadi crash

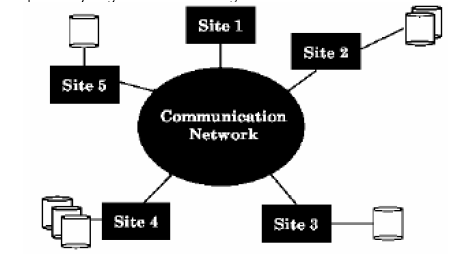
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| no | log entry (buffer) | DBMS (disk) | | |
| A | B | C |
| 1 | <start T1 > | - | - | - |
| 2 | <T1,a,10,5> | 5 | - | - |
| 3 | <start T2> | 5 | - | - |
| 4 | <T2,c,10,0> | 5 | - | 0 |
| 5 | <T2,a,5,15> | 15 | - | 0 |
| 6 | <commit T2> | 15 | - | 0 |
| 7 | <start T3> | 15 | - | 0 |
| 8 | <T3,b,10,15> | 15 | 15 | 0 |
| 9 | <rollback T1> | 15 | 10 | 0 |
| 10 |  |  |  |  |

**Arsitektur dan Basis data terdistribusi**

1. Isilah dengan singkat :
2. Arsitektur di bawah ini merupakan arsitektur share disk



1. Arsitektur di bawah ini merupakan arsitektur distributed systems



1. Sebutkan 2 Karakteristik homogeneous distributed database !
2. Sebutkan 2 keuntungan replikasi basis data !

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Selamat Mengerjakan \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*