CHAPTER 14
DISCUSSIONS 1

Discussion 14-1

Consider the task of transferring $100 from account *A* to account *B*. How would you define the transaction(s) for the task? Why?

Options:

1. as one single transaction.
2. as two separate transactions:
   - one for subtracting $100 from *A* and
   - another for adding $100 to *B*.

Discussion 14-2

Consider the task of increasing every employee’s salary by 5% for a company with 500 employees. How would you define the transaction(s) for the task? Why?

Options:

1. as one single transaction.
2. as ten separate transactions:
   one for every 50 employees.
3. as 500 separate transactions:
   one for each employee.

Discussion 14-3

Can you think of an interesting situation where a concurrent execution of two (correct) transactions produces an incorrect result? Provide the *schedule* of the situation.

Discussion 14-4

Consider the following situation.

① A user makes a reservation over the Web.

② The *database system crashes* just *after* the reservation transaction commits and before sending the result to the application server.

What should the DB system and the application server do to provide a consistent service?

What if the crash occurred *before* the transaction committed?

Discussion 14-5

Justify the following statement:

Concurrent execution of transactions is …

- *more important when data must be fetched from (slow) disk or when transactions are long, and*
- *less important when data are in memory and transactions are short.*