Transactions

• A couple examples
• ACID Properties
  – Atomicity
  – Consistency
  – Isolation
  – Durability
Examples

• Consider a database with one table called Accounts with attributes AcctNum (primary key) and Balance. There is an index on AcctNum
Example 1

Transfer $100 from account 1 to account 2
- Update Accounts Set Balance = Balance - 100 where AcctNum = 1
- Update Accounts Set Balance = Balance + 100 where AcctNum = 2
- Commit

Transfer $200 from account 3 to account 4
- Update Accounts Set Balance = Balance - 200 where AcctNum = 3
- Update Accounts Set Balance = Balance + 200 where AcctNum = 4
- Commit
Example 1

Interleaved Execution 1
• Update Account 1
• Update Account 2
• Commit
• Update Account 3
• Update Account 4
• Commit

Interleaved Execution 2
• Update Account 1
• Update Account 3
• Update Account 4
• Commit
• Update Account 2
• Commit
Example 1

• Serial Schedule
  – Transactions complete in some serial order
  – Execution 1?
  – Execution 2?
Example 2

Transfer $100 from account 1 to account 2

• Update Accounts Set Balance = Balance - 100 where AcctNum = 1
• Update Accounts Set Balance = Balance + 100 where AcctNum = 2
• Commit

Transfer $200 from account 1 to account 4

• Update Accounts Set Balance = Balance - 200 where AcctNum = 1
• Update Accounts Set Balance = Balance + 200 where AcctNum = 4
• Commit
Example 2

Interleaved Execution 1
• Update Account 1
• Update Account 2
• Commit
• Update Account 1
• Update Account 4
• Commit

Interleaved Execution 2
• Update Account 1
• Update Account 1
• Update Account 4
• Commit
• Update Account 2
• Commit
Example 2

• Serial Schedule
  – Transactions complete in some serial order
  – Execution 1?
  – Execution 2?
  – What operations are needed to complete the update?
Subtract 100 from the Balance of Account 1

- Search\textsubscript{1} Index
- Read\textsubscript{1} balance for account 1 into b1
- b1 = b1 - 100
- Write b1 as the new balance for account 1
- Commit\textsubscript{1}
Subtract 200 from the Balance of Account 1

- Search$_2$ Index
- Read$_2$ balance for account 1 into b2
- $b2 = b2 - 200$
- Write $b2$ as the new balance for account 1
- Commit$_2$
Possible Interleaved Schedule

- Search$_1$ Index
- Read$_1$ balance for account 1 into b1
- Search$_2$ Index
- Read$_2$ balance for account 1 into b2
- b2 = b2 - 200
- b1 = b1 - 100
- Write b1 as the new balance for account 1
- Write b2 as the new balance for account 1
Transactions

• The previous slide shows an example of the problem transaction processing is trying to solve.
Transaction Properties

• **ACID**
  – Atomicity
  – Consistency
  – Isolation
  – Durable
Atomicity

• The system must ensure that either the transaction runs to completion or, if it does not complete, it has no effect at all.

• Maintain log of updates

• Rollback
Consistency

• Assume all integrity constraints are satisfied before the transaction starts
• Then all integrity constraints must be satisfied after the transaction completes
• For example money is not created or lost as the result of a transfer
Isolation

• Even though transactions are executed concurrently, the overall effect of the schedule is the same as if the transactions had executed serially in some order

• Locking
  – DeadLock
  – Two Phase Locking
  – Granularity of locks

• Serializability

• Isolation levels less strict are also supported
Durability

- The system must ensure that once the transaction commits, its effects remain in the database even if the computer or the medium on which the database is stored subsequently fails.
- Log files
- Recovery procedure