# What's covered here?

- Transaction Definition
- Data Integrity Issues
- Concurrency Control
- Transaction Throughput

# Transaction

- Logical unit of work on the database.
- Can have one of two outcomes:
  - Success - transaction *commits* and database reaches a new consistent state.
  - Failure - transaction *aborts*, and database must be restored to consistent state before it started.
- Committed transaction cannot be aborted.
- Aborted transaction that is rolled back can be restarted later.

# Properties of Transactions

Four basic *(ACID)* properties of a transaction are:

| | |
|---|---|
| Atomicity | 'All or nothing' property. |
| Consistency | Must transform database from one consistent state to another. |
| Isolation | Partial effects of incomplete transactions should not be visible to other transactions. |
| Durability | Effects of a committed transaction are permanent and must not be lost because of later failure. |

# Data Integrity Issues

- Lost Updates
- Dirty Reads
- Nonepeatable Reads
- Phantoms

# Isolation Levels

- Read Uncommitted (Pessimistic only)
  - Allows Dirty Reads, Nonrepeatable Reads, Phantoms
- Read Committed (Pessimistic & Optimistic)
  - Allows Nonrepeatable Reads, Phantoms
- Repeatable Read (Pessimistic only)
  - Allows Phantoms
- Serializable (Pessimistic only)
  - No data integrity issue
- Snapshot (Optimistic only)

# Concurrency Control

Process of managing simultaneous operations on the database without having them interfere with one another

# Concurrency Control Techniques

- Two basic concurrency control techniques
  - Pessimistic
  - Optimistic

# Locking - Basic Rules
## (Pessimistic Techniques)

- Shared lock can read but not update
- Exclusive lock can both read and update
- Reads cannot conflict, shared locks can coexist on same item
- Exclusive lock gives transaction exclusive access

# Granularity of Locking
## (Pessimistic Techniques)

- Size of data items chosen as unit of protection by concurrency control protocol.
- Row, Page (8KB), Extent(64KB), Table, Database
- Tradeoff:
  - coarser, lower concurrency;
  - finer, more system overhead

# Deadlock
## (Pessimistic Techniques)

- An impasse that may result when two (or more) transactions are each waiting for locks held by the other to be released
- Only one way to break deadlock: abort one or more of the transactions
- Deadlock should be managed by system and transparent to users

# Optimistic Techniques

- Assume conflict is rare and more efficient to let transactions proceed without delays
- At commit, check is made to determine whether conflict has occurred
- If there is a conflict, transaction must be rolled back and restarted
- Potentially allows greater concurrency than traditional protocols