**CS 31 Database Management Programming Lecture 13**

**Transaction** – a group of actions that is performed on the database automatically; either all actions are committed or none are committed

**Concurrent Transactions** – two transactions that are being processed at the same time

**Concurrent Update Problem** – an error condition in which one user's data changes are overwritten by another user's data changes; also known as the lost update problem

**Dirty Read** – reading data that have been changed but not yet committed to the database; such changes may later be rolled back and removed from the database – can occur with ISOLATION LEVEL READ UNCOMMITTED

**Inconsistent Read Problem** – a series of reads of a set of rows in which some of the rows have been updated by a second transaction and some of the rows have not been updated by that second transaction

**Nonrepeatable Read** – the situation that occurs when a transaction reads data it has previously read and finds modifications or deletions caused by a committed transaction – can occur with ISOLATION LEVEL READ COMMITTED

**Phantom Read** – the situation that occurs when a transaction reads data it has previously read and finds new rows that were inserted by a committed transaction – can occur with ISOLATION LEVEL READ COMMITTED

**Lock** – the process of allocating a database resource to a particular transaction in a concurrent-processing system

**Implicit Lock** – a lock that is automatically placed by the DBMS

**Explicit Lock** – a lock requested by command from an application program

**Lock Granularity** – the size of a locked data element

**Exclusive Lock** – a lock on a data resource such that no other transaction can read or update that resource

**Shared Lock** – a lock against a data resource in which only one transaction may update the data but many transactions can concurrently read that data

**Deadlock** – a condition that can occur during concurrent processing in which each of two (or more) transactions is waiting to access data that the other transaction has locked; also called a deadly embrace

**Optimistic Locking** – a locking strategy that assumes no conflict will occur, processes a transaction, and then checks to determine whether conflict did occur; if conflict did occur, no changes are made to the database and the transaction is repeated

**Pessimistic Locking** – a locking strategy that prevents conflict by locking data resources, processing the transaction, and then unlocking the resources