

## Introduction to Lock Management

- The part of the DBMS that keeps track of the locks issued two transactions is called the Lock Manager
- The Lock Manager maintains a lock table which is a hash table with the data object identifier as the key
- The DBMS also maintains a descriptive entry for each transaction in a transaction table
- A lock table entry of an object which can be a page, a record and so on depending on the DBMS contains the following information
  1. The number of transactions currently holding the lock on the object
  2. The nature of the lock
  3. A pointer to a queue of lock requests

## Implementing Lock and Unlock requests

- According to the Strict 2PL protocol before a transaction T reads or writes a database object O it must obtain a shared or exclusive on O and must hold on to the lock until it commits or aborts
- When a transaction needs a lock on an object it issues a lock request to the manager
- The queue of requests is empty and the object is not currently locked in exclusive mode
  1. The Lock Manager grants the lock and updates the lock table entry
  2. And no transaction currently holds a lock on the object lock and updates the lock table entry
  3. Otherwise the requested lock cannot be immediately granted is added to the queue for this object, the transaction requesting the lock is suspended
- When a transaction aborts it releases all its locks

## Specialised locking techniques

- If the collection of database objects is not fixed but can grow and strength to the insertion and deletion of objects we must be with a subtle complication known as the phantom problem
- Although treating a database as an independent collection of objects is adequate for a discussion of serialisability and recoverability much better performance can sometimes be obtained using protocols that recognised and exploit relationships between objects

### **Dynamic Databases and the phantom problem**

- Consider the following the following transaction T1 scans the sailors relation to find the oldest sailor for each of the rating levels 1 and 2
- First T1 identifies and locks all pages and containing sailors with rating one and then finds the age of the oldest sailor which is say 71 next transaction T2 inserts a new sailor with rating 1 and age 96