

Database Management Systems

- Exam Review

Doug Shook

Logistics

- Monday May 7th, 1-3PM
- Open book/notes
- No electronic devices

Topic Outline

■ Transactions

- Issues that can arise with concurrent operations
- Transaction state and log files
- ACID Properties
- Scheduling
 - Conflicts
 - Serializability

■ Locks

- Types of locks
- Two phase locking
- Deadlock detection and avoidance
- Granularity

Topic Outline

- Data Distribution
 - Advantages
 - Partitioning schemes
 - Data transfer costs
 - Three phase commit
- Security
 - User based control
 - Role based control
 - SQL Injection

Topic Outline

- Data Warehousing
 - Design
 - Cube Operations
 - OLTP vs. OLAP
- NoSQL
 - Comparison with relational
 - MongoDB
 - Basic setup and usage

Exercise

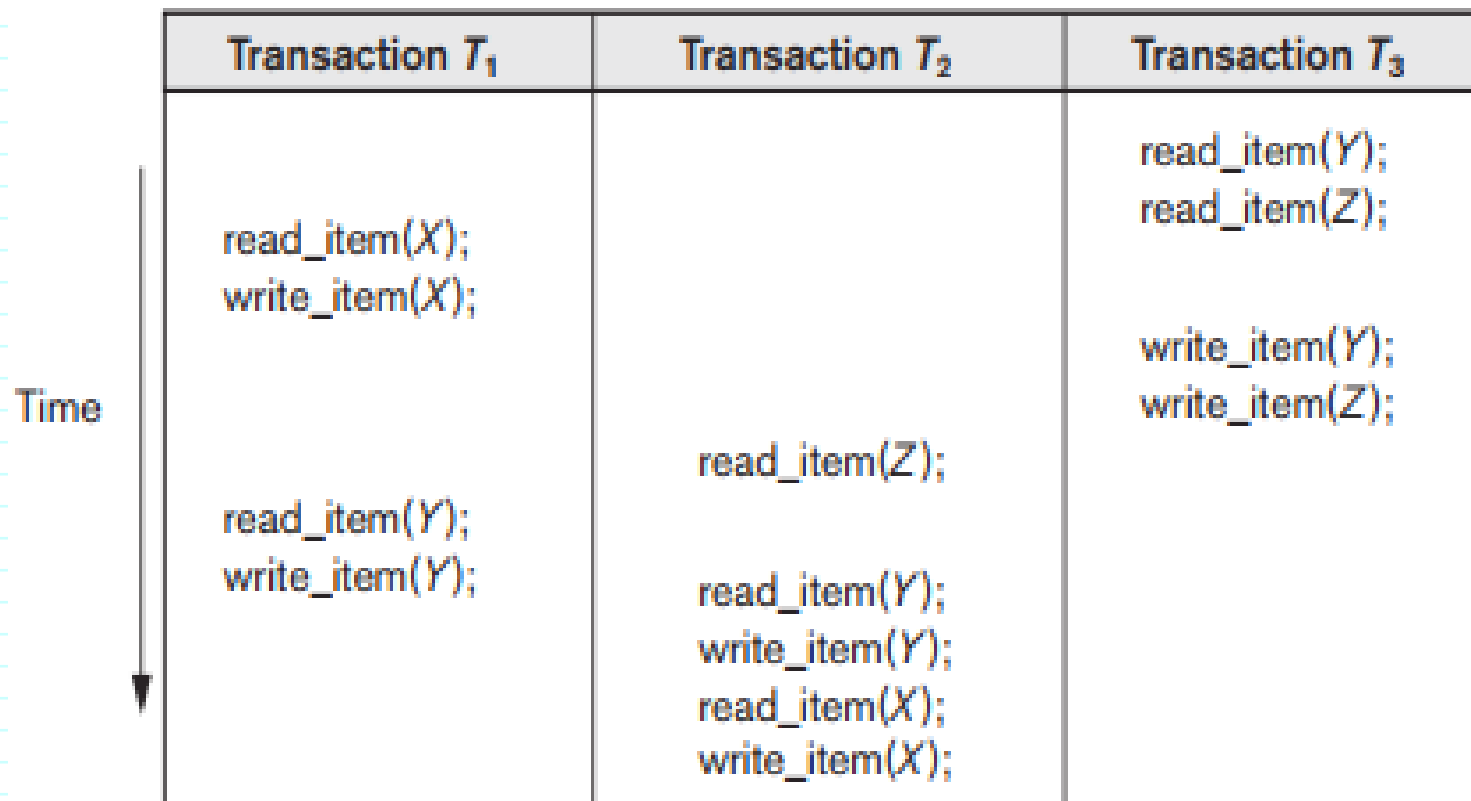
- For the following schedules:
 - What conflicts exist?
 - Is this schedule serializable?
 - Will there be deadlock?

$R1(X), R2(X), W1(Y), W1(Z), R2(Z), R1(Y), R2(Y)$

$R1(Z), W2(Y), R3(Y), W2(Z), R1(Y), W1(X),$
 $R3(Z), W1(Y), W3(Z)$

Exercise

- Could the following schedule suffer from deadlock? Explain how you know. If deadlock is a problem, explain how it could be resolved.



Transaction T_1	Transaction T_2	Transaction T_3
<code>read_item(X);</code> <code>write_item(X);</code>		<code>read_item(Y);</code> <code>read_item(Z);</code>
	<code>read_item(Z);</code>	<code>write_item(Y);</code> <code>write_item(Z);</code>
<code>read_item(Y);</code> <code>write_item(Y);</code>	<code>read_item(Y);</code> <code>write_item(Y);</code> <code>read_item(X);</code> <code>write_item(X);</code>	

Schedule F

Exercise

- We wish to distribute our course tracking database so that each department is considered a separate site.
 - Explain what kinds of partitioning would need to take place to accomplish this.
 - Consider a query that reports all of the courses that a student has taken across all departments
 - What should the data transmission pattern be?
 - Will distribution improve or degrade the performance of this query?

Exercise

- Explain what steps are taken to prevent SQL injection attacks
- Give an example of a second kind of attack on a database (not SQL injection) and describe how it could be prevented

Exercise

- Suppose you have a data warehouse with dimensions of *date*, *doctor*, and *patient*, and the measures of *count* and *charge*.
 - Draw a schema for this warehouse. Include attributes for each dimension.
 - Starting with a cube that has dimensions of day, doctor, and patient, which cube operations would be needed to find the total fees collected by each doctor in 2009?
 - Write a SQL query to perform these operations.

Exercise

- We wish to design a MongoDB database that tracks what movies different users have watched.
- Design the database: what documents should exist? What should those documents contain?
- Write code to insert a few example documents into your database.
- Write a query that will list all of the movies watched by a particular user.