



- As app program is executing, it is "in a transaction"
- Program can execute COMMIT

April 2004

- SQL command to finish the transaction successfully
- The next SQL statement will automatically start a new transaction

Transactions by Alan Fekete

Warning

- The idea of a transaction is hard to see when interacting directly with DBMS, instead of from an app program
- Using an interactive query interface to DBMS, by default each SQL statement is treated as a separate transaction (with implicit COMMIT at end) unless you explicitly say "START TRANSACTION"

Transactions by Alan Fekete

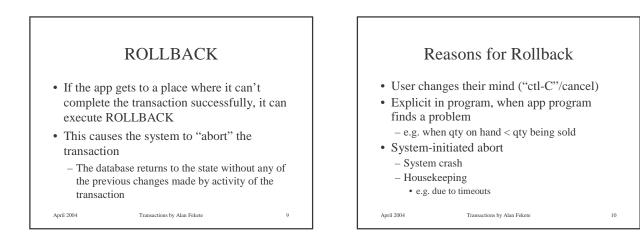
April 2004

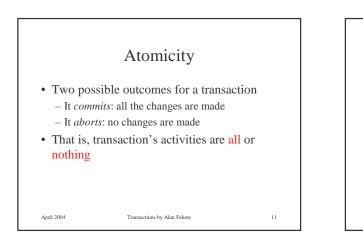
A Limitation

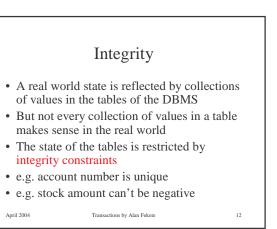
- Some systems rule out having both DML and DDL statements in a single transaction
- i.e., you can change the schema, or change the data, but not both

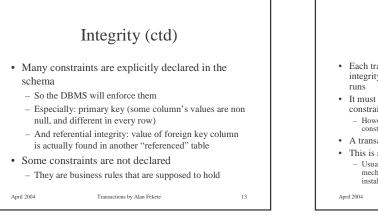
Transactions by Alan Fekete

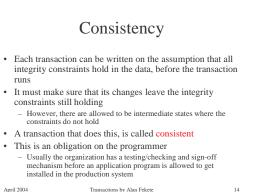
April 2004

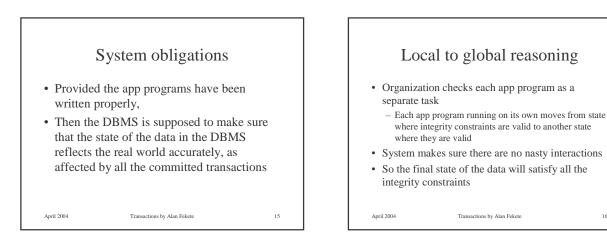


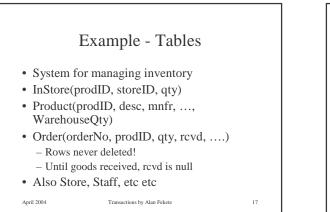


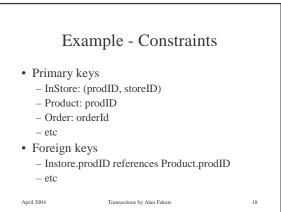


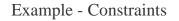










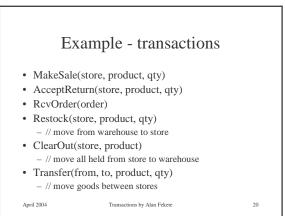


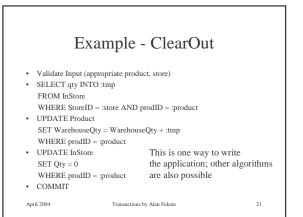
- · Data values
 - Instore.qty >= 0
 - Order.rcvd <= current_date or Order.rcvd is null</p>
- Business rules
 - for each p, (Sum of qty for product p among all stores and warehouse) >= 50
 - for each p, (Sum of qty for product p among all stores and warehouse) >= 70 or there is an outstanding order of product p

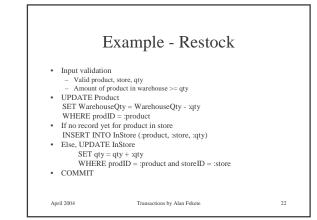
Transactions by Alan Fekete

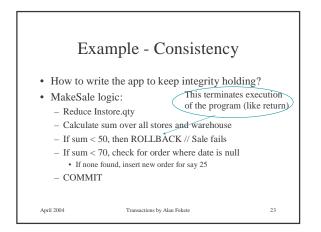
19

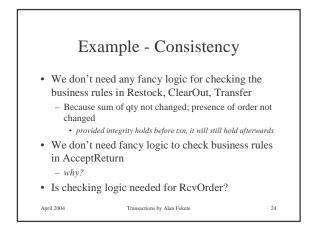
April 2004

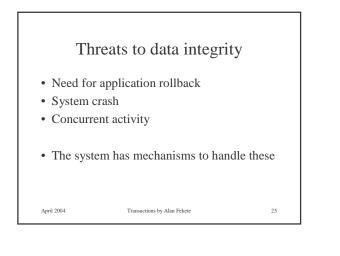


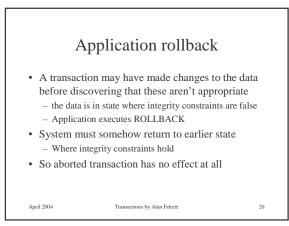


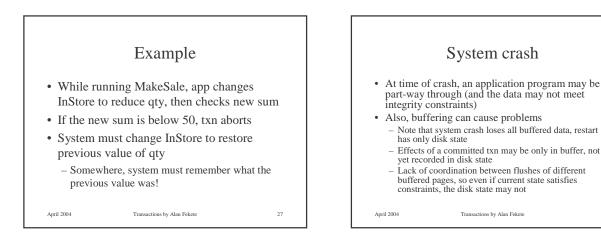


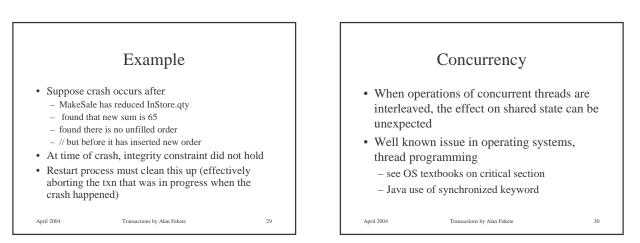


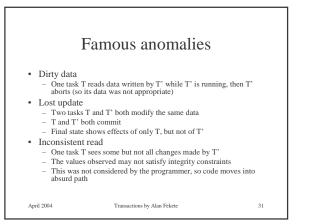


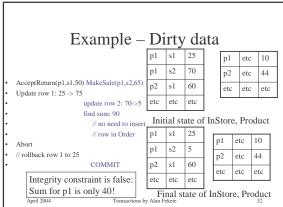


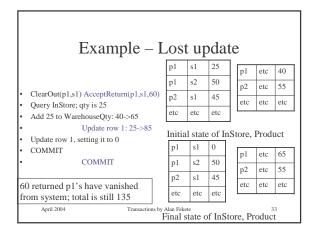


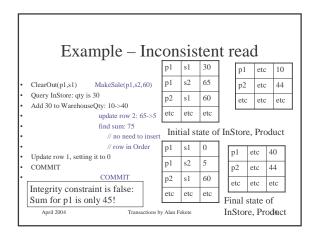


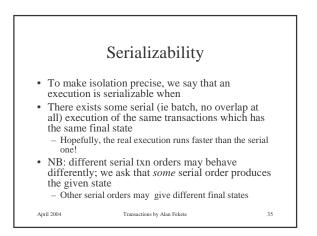


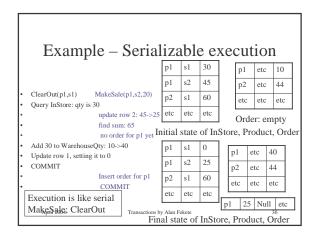


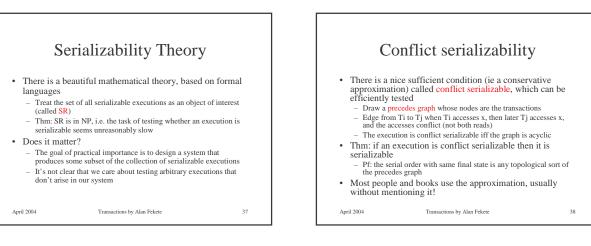


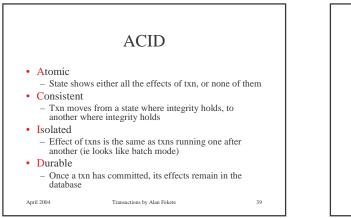


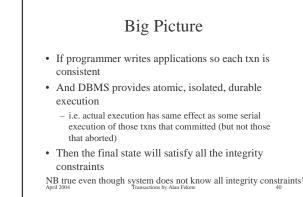


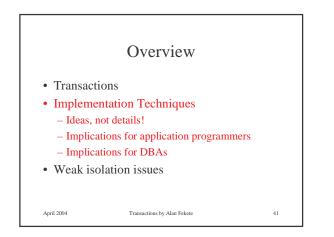


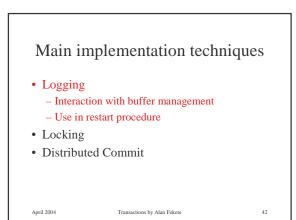


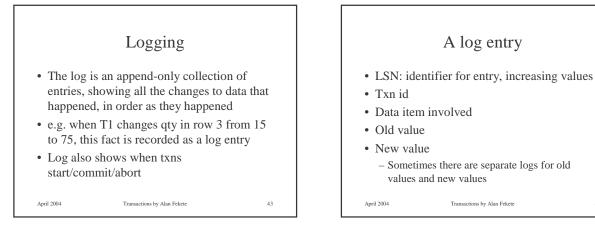


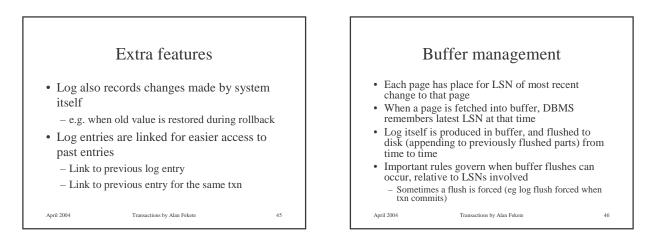


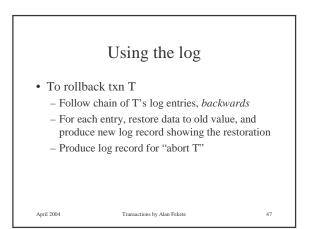


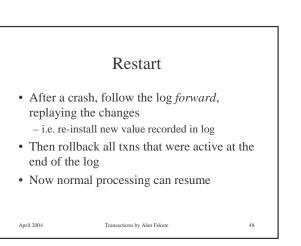


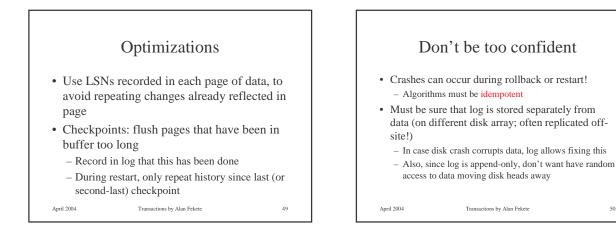


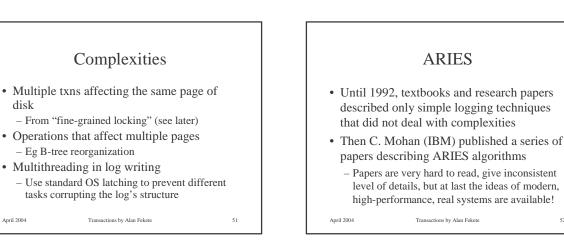


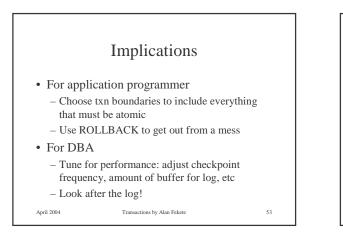


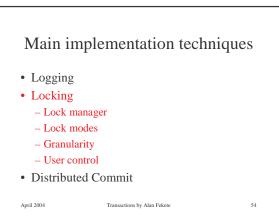


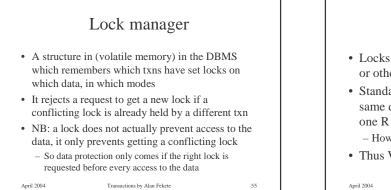










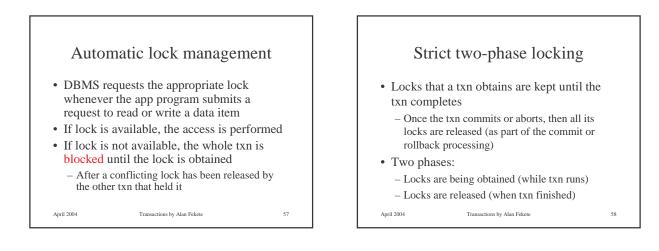


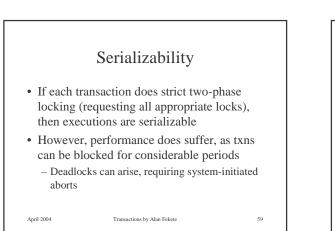


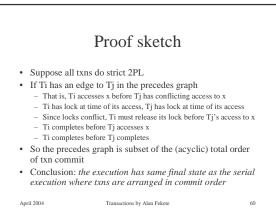
- Locks can be for writing (W), reading (R) or other modes
- Standard conflict rules: two W locks on the same data item conflict, so do one W and one R lock on the same data

Transactions by Alan Fekete

- However, two R locks do not conflict
- Thus W=exclusive, R=shared







	Example N		Dir	4 * 7	da	to				
	Example – N AcceptReturn(p1,s1,50) MakeSale(p1,s2,65)	p1		25	ua	p1	etc	10		
•	Update row 1: 25 -> 75 //t1 W-locks InStore, row 1	p1	s2	70		p2	etc	44		
•	update row 2: 70->5	p2	s1	60		etc	etc	etc		
•	//t2 W-locks Instore.row2 try_find sum:// blocked	etc	etc	etc						
:	// as R-lock on Instore.row1 // can't be obtained Initial state of InStore, Product									
•	User-initiated Abort // rollback row 1 to 35; release lock	p1	s1	25	Гр	1 6	etc	10		
•	// now get locks find sum: 40	p1	s2	70	p	2 6	etc	44		
•	ROLLBACK // row 2 restored to 70	p2	s1	60	e	tc e	etc	etc		
•	Integrity constraint is valid	etc	etc	etc	Final state of					
	April 2004 Transactions by Alan Fekete					InStore, Product				

