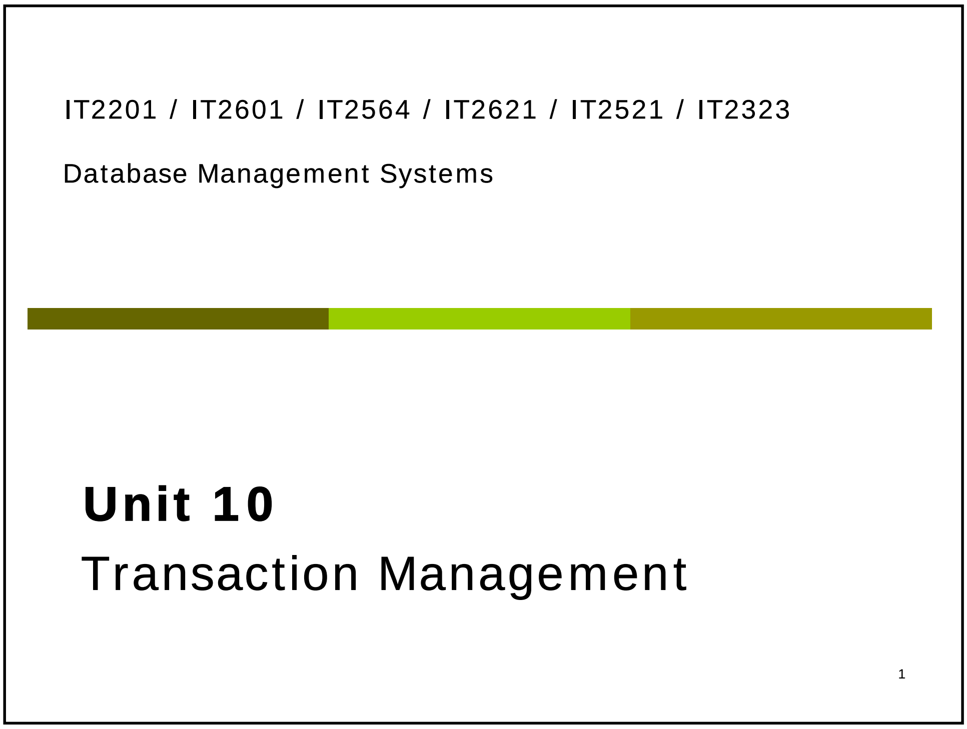
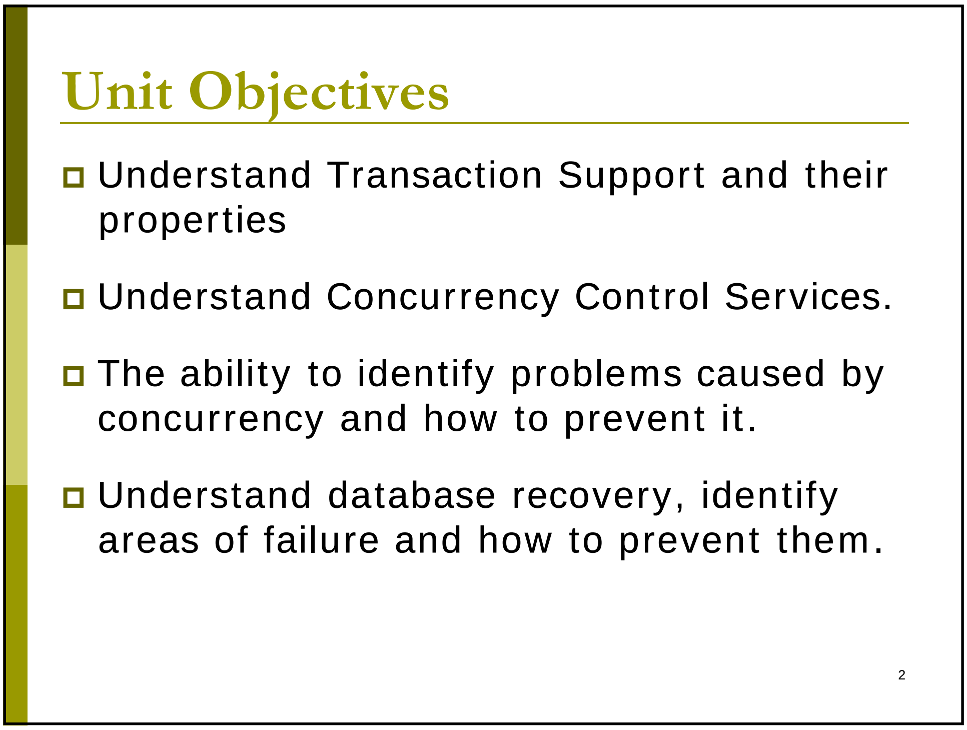


IT2201 / IT2601 / IT2564 / IT2621 / IT2521 / IT2323

Database Management Systems

Unit 10
Transaction Management



Unit Objectives

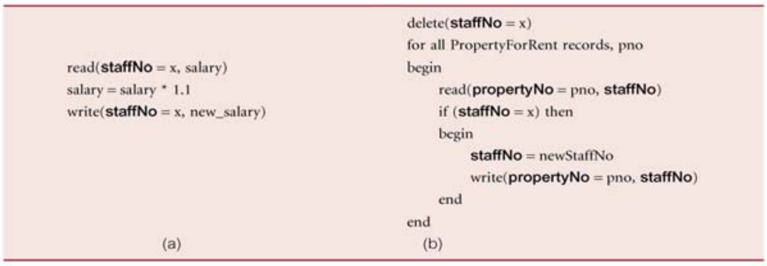
- Understand Transaction Support and their properties
- Understand Concurrency Control Services.
- The ability to identify problems caused by concurrency and how to prevent it.
- Understand database recovery, identify areas of failure and how to prevent them.

Transaction Support

- What is a Transaction
  - Action, or series of actions, carried out by user or application, which accesses or changes contents of database.
- A transaction is a logical unit of work on the database.
- A transaction transforms database from one consistent state to another.
- A transaction can have one of two outcomes:
  - commit
  - rollback

Example of a Transaction

- Transaction (a)
  - To update the salary of a particular member of staff given the staff number, x.
- Transaction (b)
  - To delete the member of staff with a given staff number x, and to reassign the properties managed by this staff to a different member of staff.

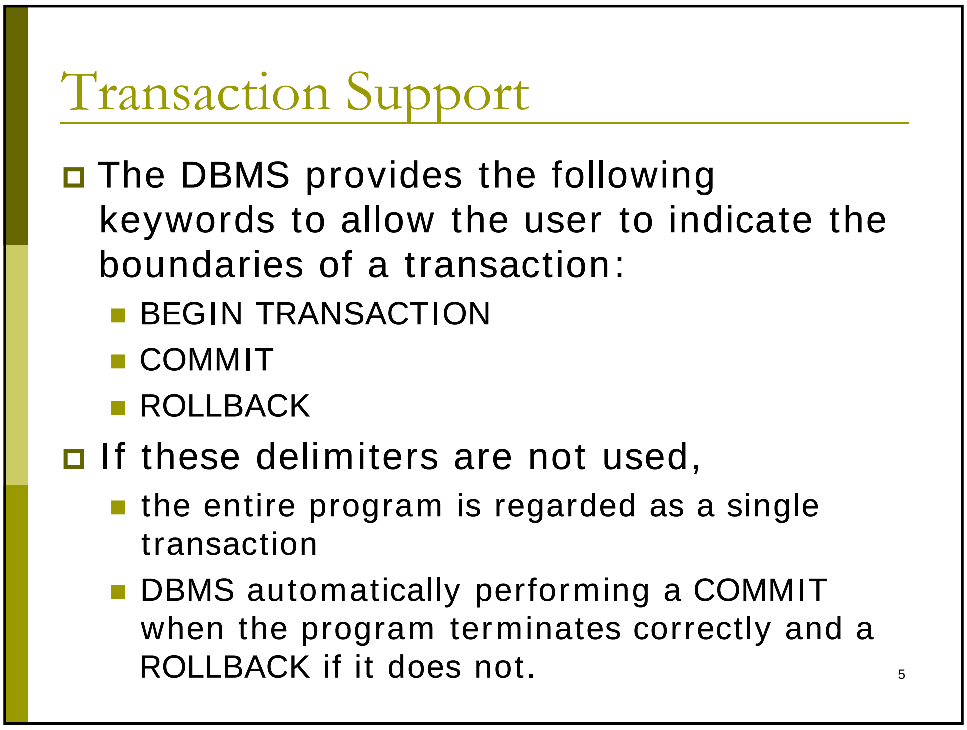


```
read(staffNo = x, salary)
salary = salary * 1.1
write(staffNo = x, new_salary)
```

(a)

```
delete(staffNo = x)
for all PropertyForRent records, pno
begin
    read(propertyNo = pno, staffNo)
    if (staffNo = x) then
    begin
        staffNo = newStaffNo
        write(propertyNo = pno, staffNo)
    end
end
```

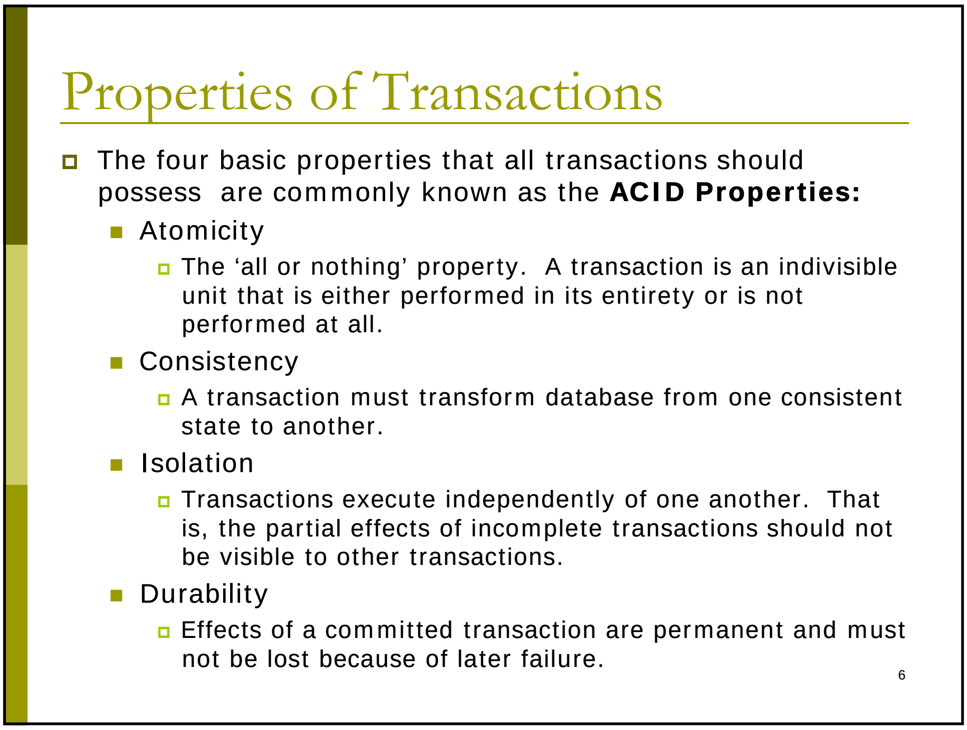
(b)



Transaction Support

- The DBMS provides the following keywords to allow the user to indicate the boundaries of a transaction:
  - BEGIN TRANSACTION
  - COMMIT
  - ROLLBACK
- If these delimiters are not used,
  - the entire program is regarded as a single transaction
  - DBMS automatically performing a COMMIT when the program terminates correctly and a ROLLBACK if it does not.

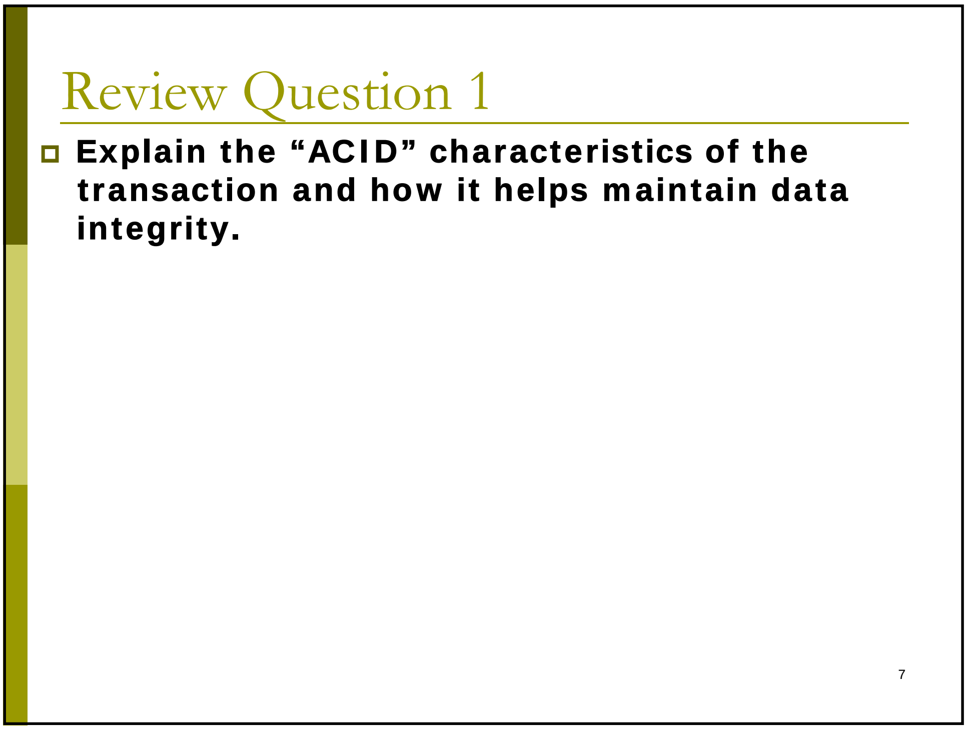
5



Properties of Transactions

- The four basic properties that all transactions should possess are commonly known as the **ACID Properties:**
  - Atomicity
    - The 'all or nothing' property. A transaction is an indivisible unit that is either performed in its entirety or is not performed at all.
  - Consistency
    - A transaction must transform database from one consistent state to another.
  - Isolation
    - Transactions execute independently of one another. That is, the 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.

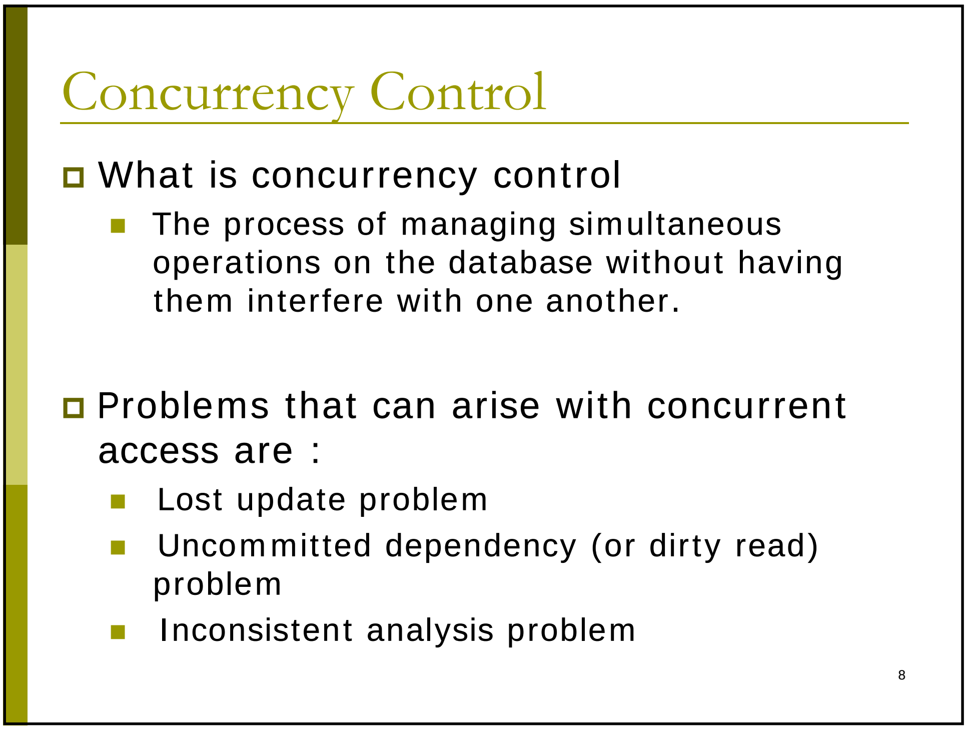
6



Review Question 1

- **Explain the "ACID" characteristics of the transaction and how it helps maintain data integrity.**

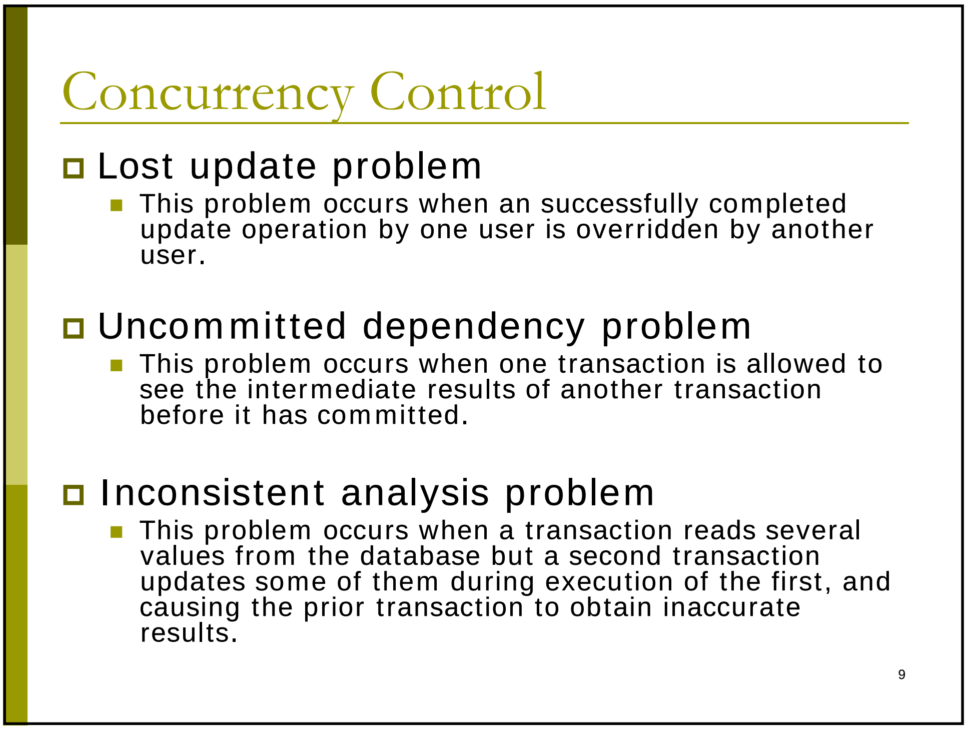
7



Concurrency Control

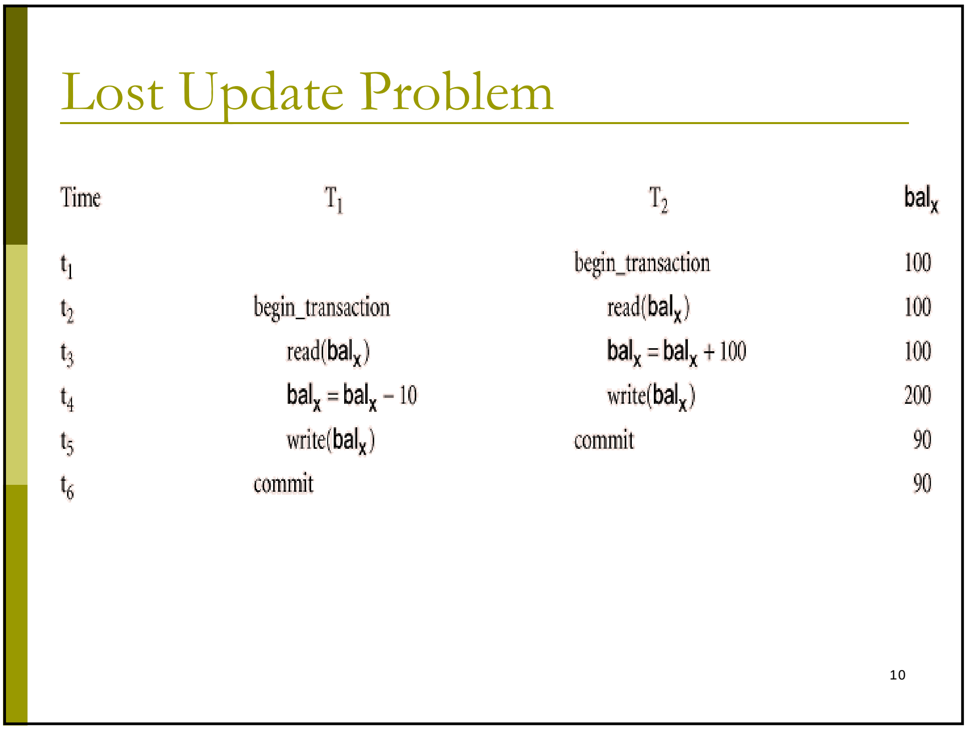
- What is concurrency control
  - The process of managing simultaneous operations on the database without having them interfere with one another.
- Problems that can arise with concurrent access are :
  - Lost update problem
  - Uncommitted dependency (or dirty read) problem
  - Inconsistent analysis problem

8



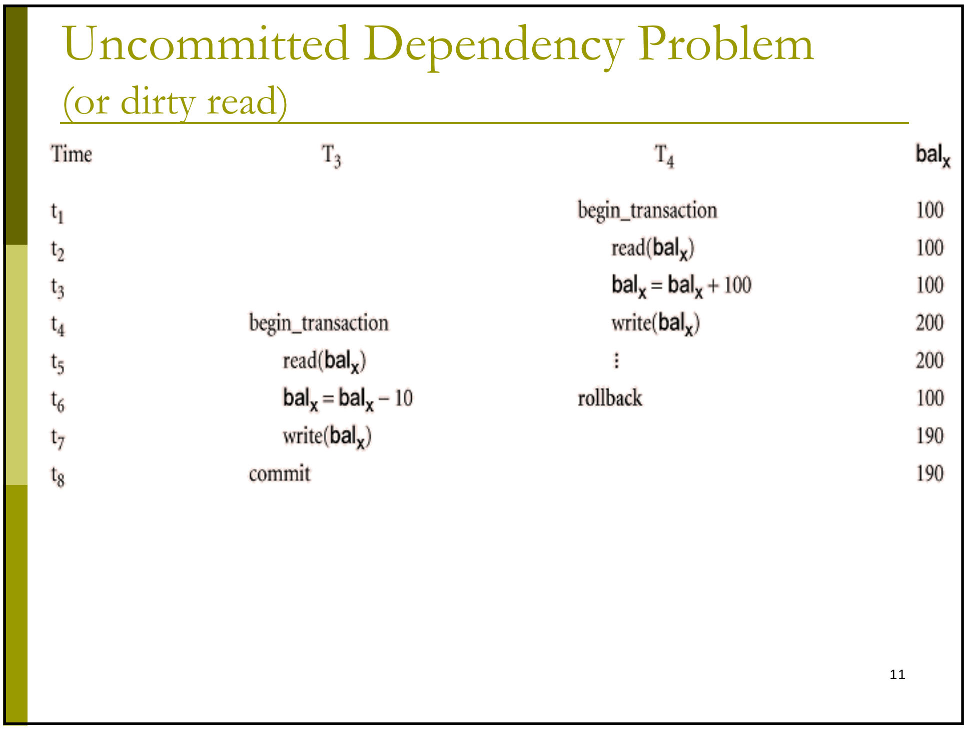
Concurrency Control

- Lost update problem
  - This problem occurs when an successfully completed update operation by one user is overridden by another user.
- Uncommitted dependency problem
  - This problem occurs when one transaction is allowed to see the intermediate results of another transaction before it has committed.
- Inconsistent analysis problem
  - This problem occurs when a transaction reads several values from the database but a second transaction updates some of them during execution of the first, and causing the prior transaction to obtain inaccurate results.



Lost Update Problem

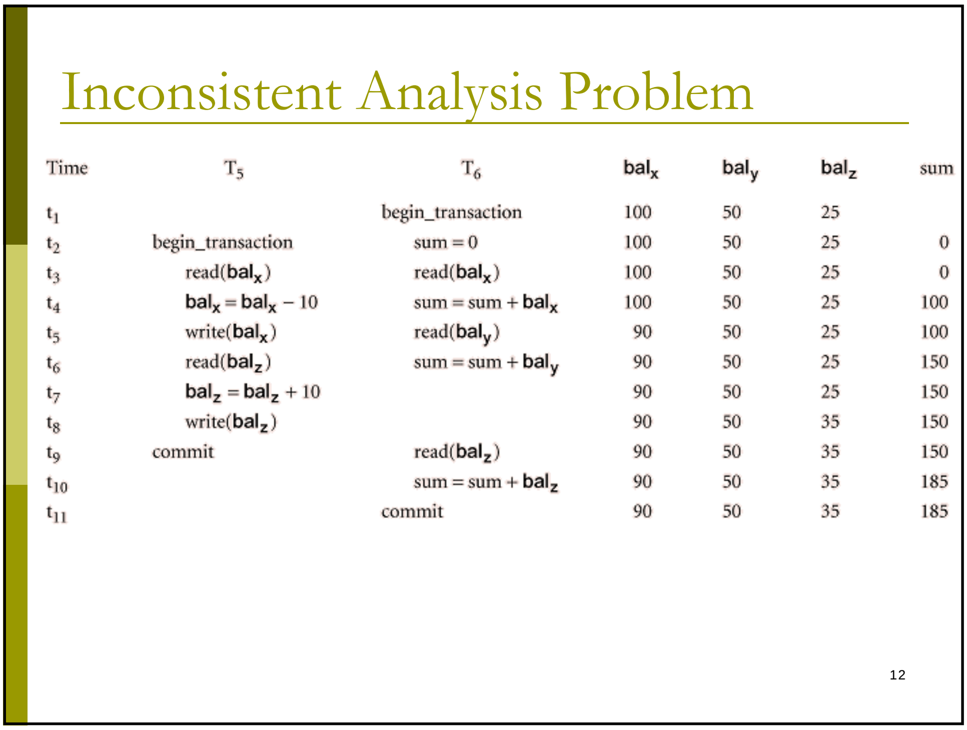
Time	T_1	T_2	bal_x
t_1		begin_transaction	100
t_2	begin_transaction	read(bal_x)	100
t_3	read(bal_x)	$bal_x = bal_x + 100$	100
t_4	$bal_x = bal_x - 10$	write(bal_x)	200
t_5	write(bal_x)	commit	90
t_6	commit		90



Uncommitted Dependency Problem (or dirty read)

Time	T_3	T_4	bal_x
t_1		begin_transaction	100
t_2		read(bal_x)	100
t_3		$\text{bal}_x = \text{bal}_x + 100$	100
t_4	begin_transaction	write(bal_x)	200
t_5	read(bal_x)	⋮	200
t_6	$\text{bal}_x = \text{bal}_x - 10$	rollback	100
t_7	write(bal_x)		190
t_8	commit		190

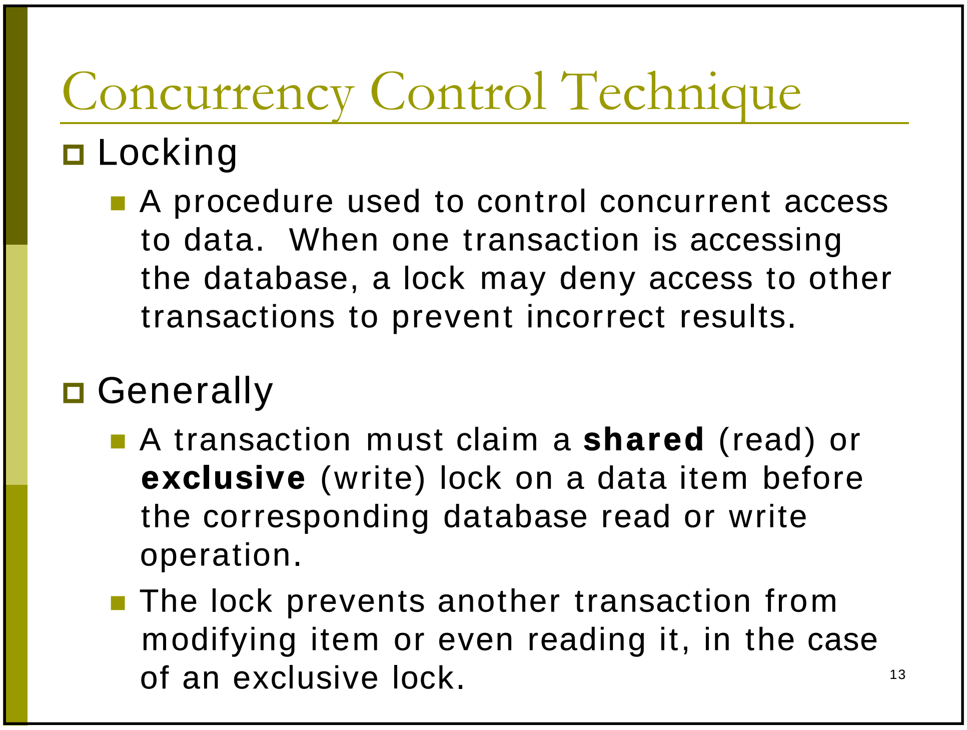
11



Inconsistent Analysis Problem

Time	T_5	T_6	bal_x	bal_y	bal_z	sum
t_1		begin_transaction	100	50	25	
t_2	begin_transaction	sum = 0	100	50	25	0
t_3	read(bal_x)	read(bal_x)	100	50	25	0
t_4	$\text{bal}_x = \text{bal}_x - 10$	sum = sum + bal_x	100	50	25	100
t_5	write(bal_x)	read(bal_y)	90	50	25	100
t_6	read(bal_z)	sum = sum + bal_y	90	50	25	150
t_7	$\text{bal}_z = \text{bal}_z + 10$		90	50	25	150
t_8	write(bal_z)		90	50	35	150
t_9	commit	read(bal_z)	90	50	35	150
t_{10}		sum = sum + bal_z	90	50	35	185
t_{11}		commit	90	50	35	185

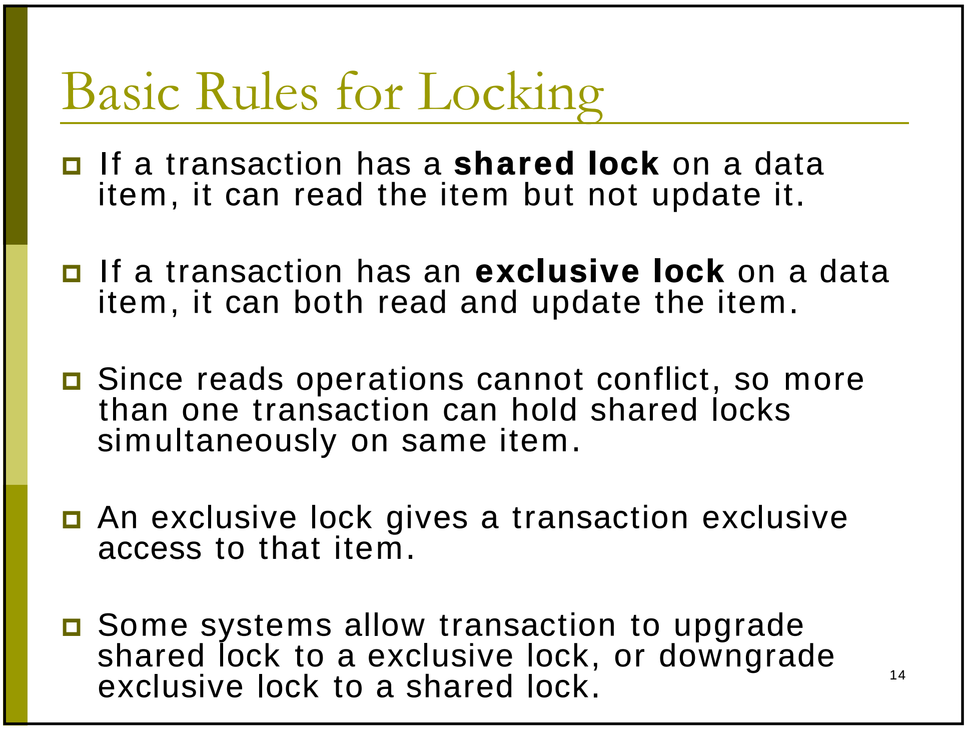
12



Concurrency Control Technique

- Locking
  - A procedure used to control concurrent access to data. When one transaction is accessing the database, a lock may deny access to other transactions to prevent incorrect results.
- Generally
  - A transaction must claim a **shared** (read) or **exclusive** (write) lock on a data item before the corresponding database read or write operation.
  - The lock prevents another transaction from modifying item or even reading it, in the case of an exclusive lock.

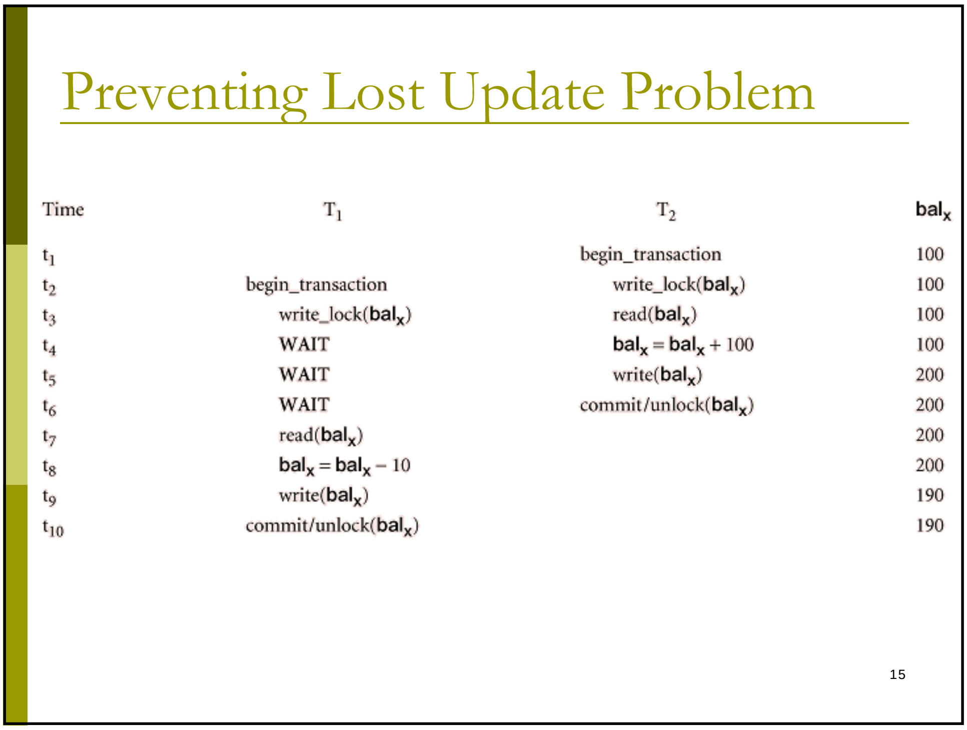
13



Basic Rules for Locking

- If a transaction has a **shared lock** on a data item, it can read the item but not update it.
- If a transaction has an **exclusive lock** on a data item, it can both read and update the item.
- Since reads operations cannot conflict, so more than one transaction can hold shared locks simultaneously on same item.
- An exclusive lock gives a transaction exclusive access to that item.
- Some systems allow transaction to upgrade shared lock to a exclusive lock, or downgrade exclusive lock to a shared lock.

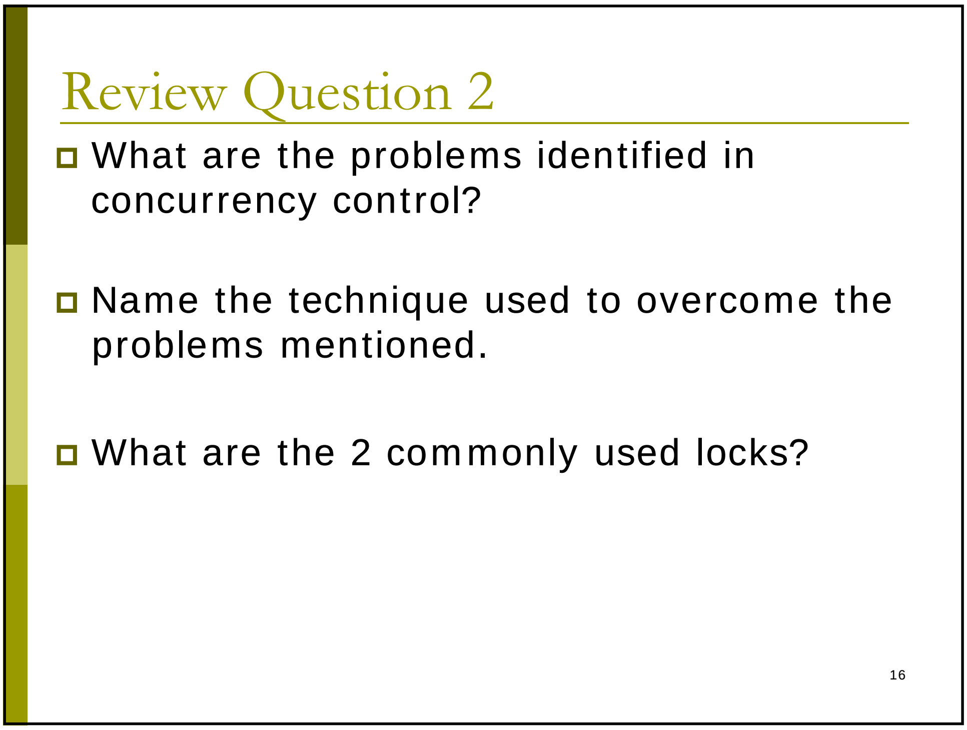
14



Preventing Lost Update Problem

Time	T_1	T_2	bal_x
t_1		begin_transaction	100
t_2	begin_transaction	write_lock(bal_x)	100
t_3	write_lock(bal_x)	read(bal_x)	100
t_4	WAIT	$bal_x = bal_x + 100$	100
t_5	WAIT	write(bal_x)	200
t_6	WAIT	commit/unlock(bal_x)	200
t_7	read(bal_x)		200
t_8	$bal_x = bal_x - 10$		200
t_9	write(bal_x)		190
t_{10}	commit/unlock(bal_x)		190

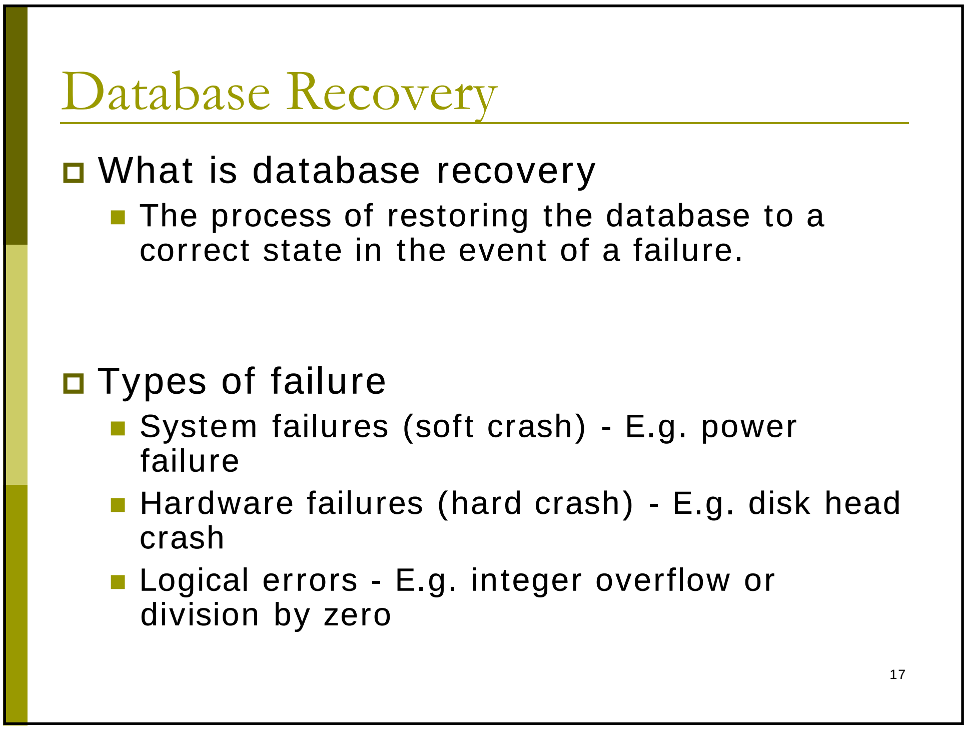
15



Review Question 2

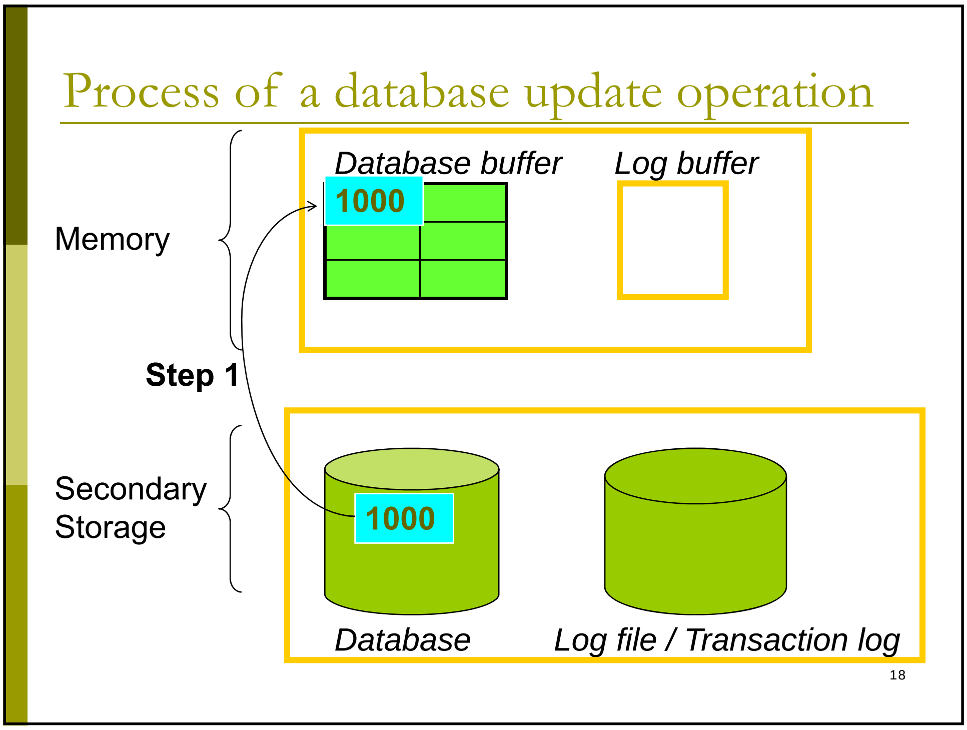
- What are the problems identified in concurrency control?
- Name the technique used to overcome the problems mentioned.
- What are the 2 commonly used locks?

16



Database Recovery

- What is database recovery
  - The process of restoring the database to a correct state in the event of a failure.
- Types of failure
  - System failures (soft crash) - E.g. power failure
  - Hardware failures (hard crash) - E.g. disk head crash
  - Logical errors - E.g. integer overflow or division by zero



Process of a database update operation

Memory

Database buffer

1000

Log buffer

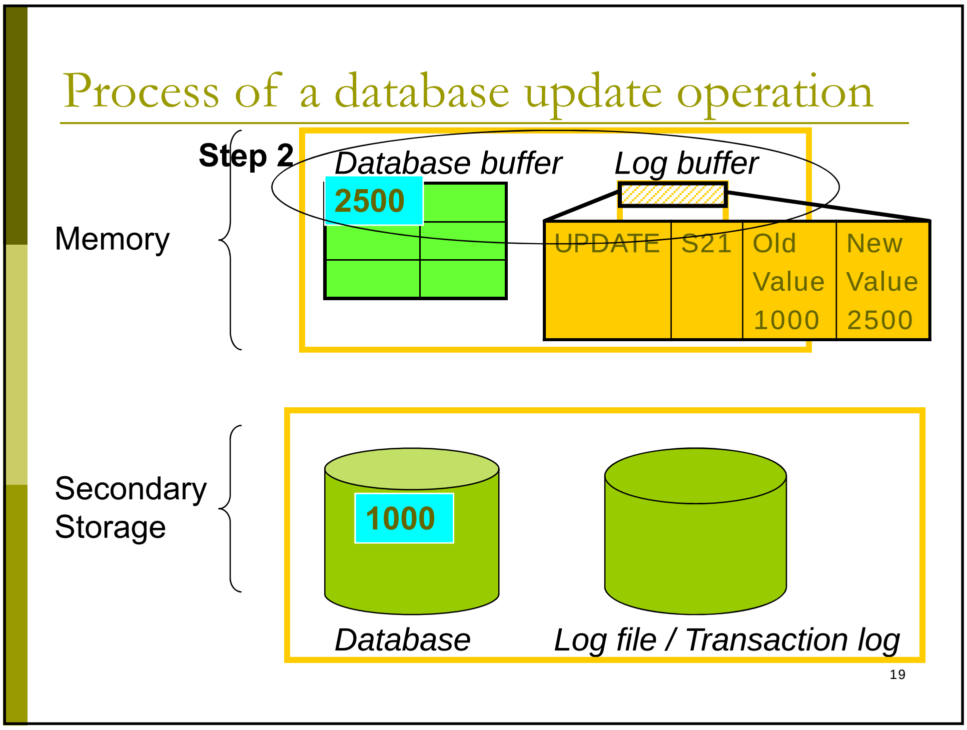
Step 1

Secondary Storage

1000

Database

Log file / Transaction log



Process of a database update operation

Step 2

Memory

Database buffer

2500

Log buffer

UPDATE	S21	Old Value 1000	New Value 2500

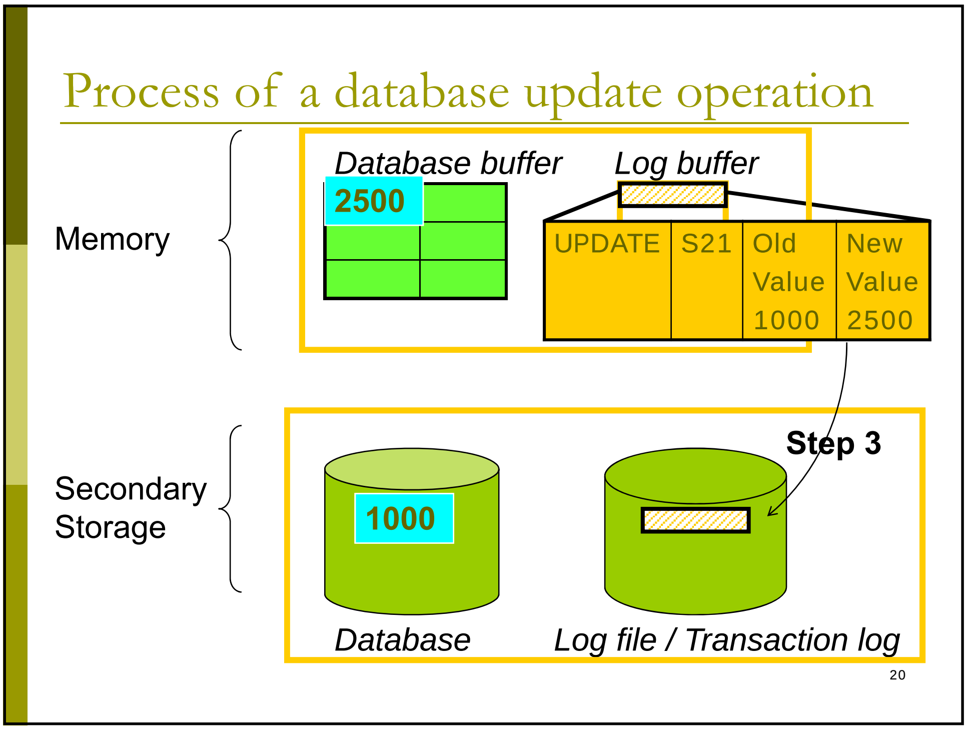
Secondary Storage

1000

Database

Log file / Transaction log

19



Process of a database update operation

Memory

Database buffer

2500

Log buffer

UPDATE	S21	Old Value 1000	New Value 2500

Step 3

Secondary Storage

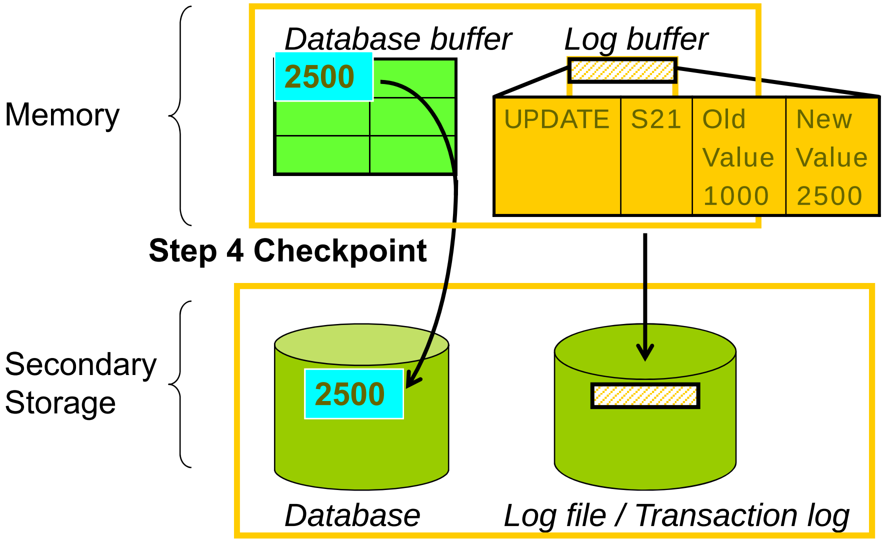
1000

Database

Log file / Transaction log

20

Process of a database update operation



Memory

Database buffer

Log buffer

2500

UPDATE	S21	Old Value 1000	New Value 2500

Step 4 Checkpoint

Secondary Storage

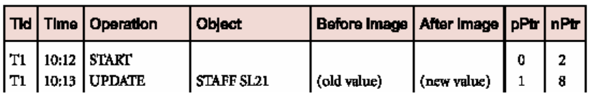
2500

Database

Log file / Transaction log

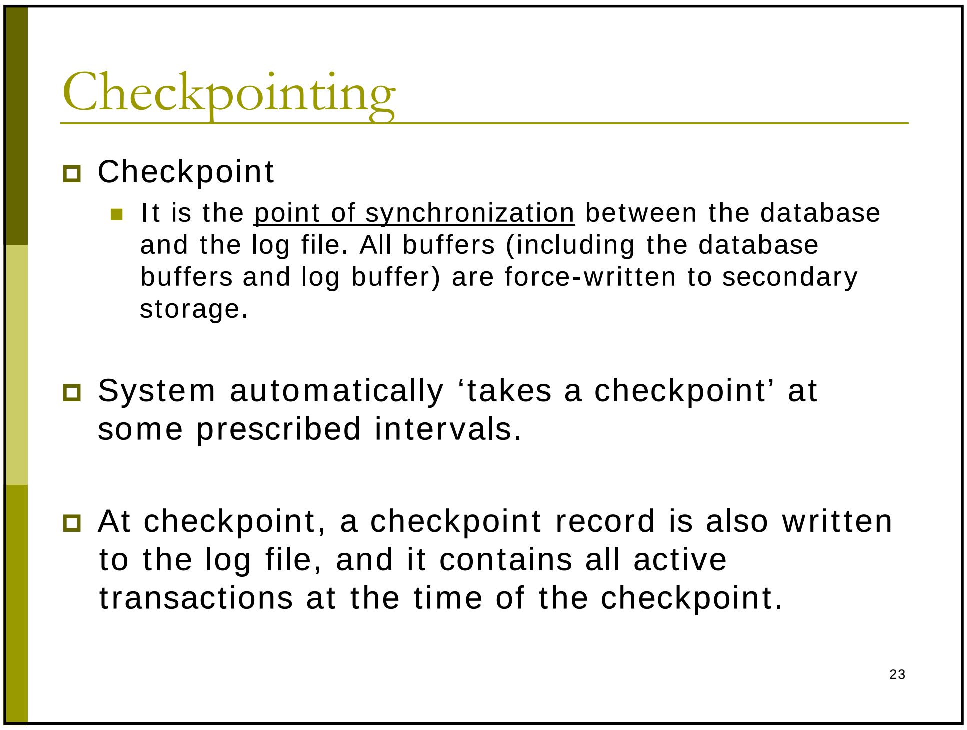
Log File

- A log file is a special file maintained by the DBMS, which contains information about all updates to database.
- E.g. of a update record in the log file :



Tid	Time	Operation	Object	Before Image	After Image	pPtr	nPtr
T1	10:12	START				0	2
T1	10:13	UPDATE	STAFF SL21	(old value)	(new value)	1	8

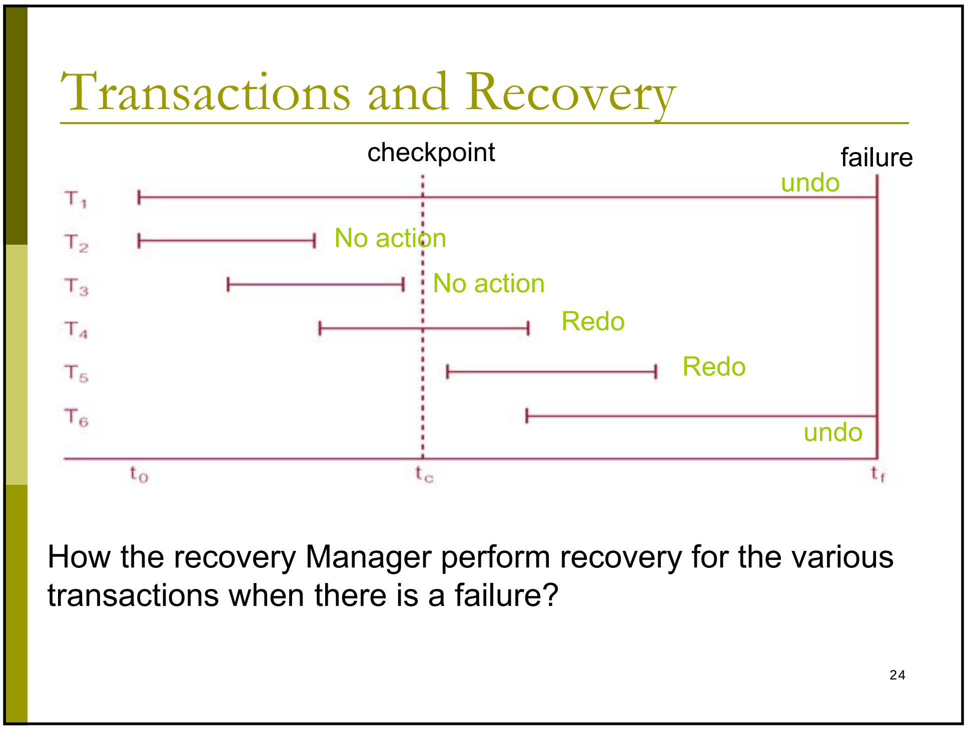
  - Note that Before-image and After-image of data item are stored in the log file.
- The log is used for purposes such as
  - Recovery
  - Performance monitoring
  - Auditing



Checkpointing

- Checkpoint
  - It is the point of synchronization between the database and the log file. All buffers (including the database buffers and log buffer) are force-written to secondary storage.
- System automatically 'takes a checkpoint' at some prescribed intervals.
- At checkpoint, a checkpoint record is also written to the log file, and it contains all active transactions at the time of the checkpoint.

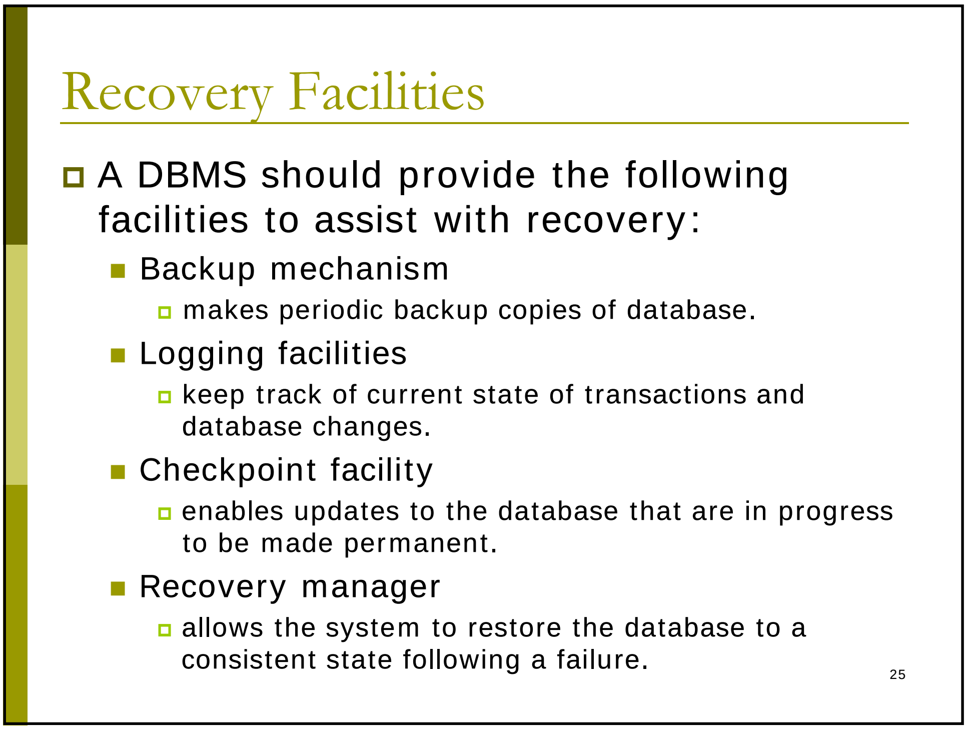
23



Transactions and Recovery

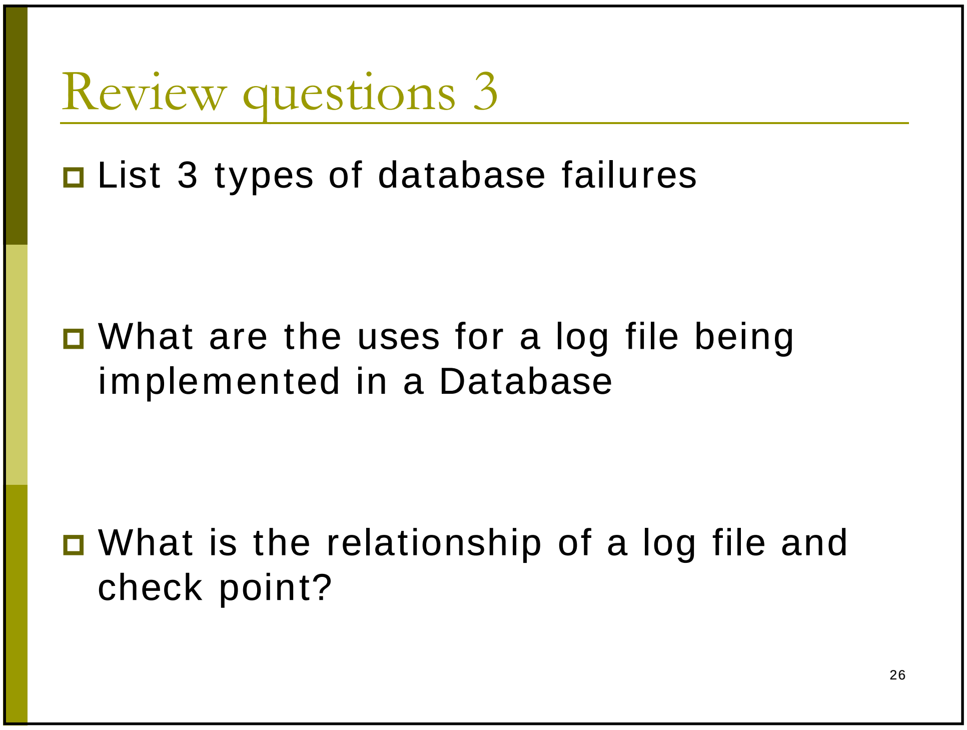
How the recovery Manager perform recovery for the various transactions when there is a failure?

24



Recovery Facilities

- A DBMS should provide the following facilities to assist with recovery:
  - Backup mechanism
    - makes periodic backup copies of database.
  - Logging facilities
    - keep track of current state of transactions and database changes.
  - Checkpoint facility
    - enables updates to the database that are in progress to be made permanent.
  - Recovery manager
    - allows the system to restore the database to a consistent state following a failure.

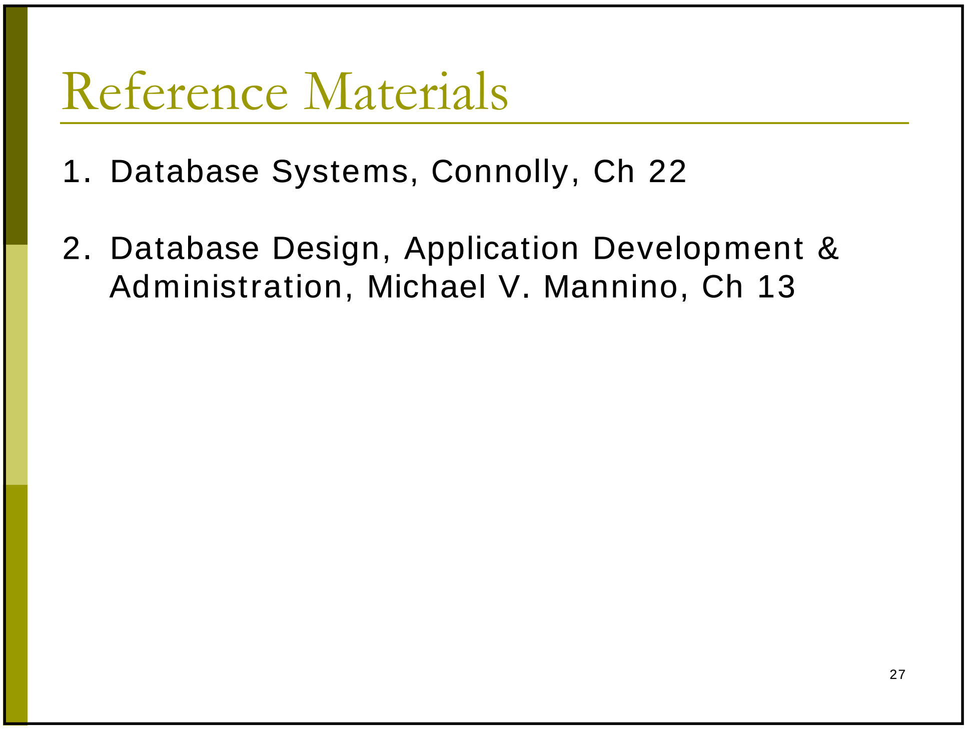


Review questions 3

- List 3 types of database failures

- What are the uses for a log file being implemented in a Database

- What is the relationship of a log file and check point?



Reference Materials

1. Database Systems, Connolly, Ch 22
2. Database Design, Application Development & Administration, Michael V. Mannino, Ch 13