CHAPTER 12: DATA AND DATABASE ADMINISTRATION

Modern Database Management

12th Edition

Jeff Hoffer, Ramesh Venkataraman, Heikki Topi

OBJECTIVES

- Define terms
- List functions and roles of data/database administration
- Describe role of data dictionaries and information repositories
- Compare optimistic and pessimistic concurrency control
- Describe problems and techniques for data security
- Describe problems and facilities for data recovery
- Describe database tuning issues and list areas where changes can be done to tune the database
- Describe importance and measures of data availability

INEFFECTIVE DATA ADMINISTRATION POOR DATA QUALITY

- Multiple data definitions of the same data entity causing data integration problems
- Missing key data elements, causing loss in existing data value.
- Inappropriate data sources and timing of data transfer from one system to another, causing lowered reliability
- Inadequate familiarity, causing ineffective use of data for planning and strategy
- Poor response time and excessive downtime of database control
- Damaged, sabotaged, and stolen data
- Embarrassment to organization because of unauthorized access to data

TRADITIONAL ADMINISTRATION DEFINITIONS

- ***Data Administration:*** A high-level function that is responsible for the overall management of data resources in an organization, including maintaining corporate-wide definitions and standards
- ***Database Administration:*** A technical function that is responsible for physical database design and for dealing with technical issues such as security enforcement, database performance, and backup and recovery

TRADITIONAL DATA ADMINISTRATION FUNCTIONS

- Data policies, procedures, standards
  - Data policies: State goal of data administration
  - Procedure: Outline of actions to perform activity
  - Standard: Convention needs to follow to evaluate database quality. This should be standardized for programmers.
- Planning
  - Require understands of the need for data and lead development of architecture to meet the needs of organizations.
- Data conflict (ownership) resolution
- Managing the information repository

TRADITIONAL DATABASE ADMINISTRATION FUNCTIONS

- Analyzing and designing databases
- Selecting DBMS and software tools
- Installing/upgrading DBMS:
  - Making db faster to cope with requests, and integrate with 3rd – party software easily
  - Ensure applications work properly after upgrade

CONT

- Tuning database performance
  - Query should be monitored constantly.
  - DB design is changed to meet requirements
  - DB is rebuilt, reorganized, re-index to keep performance
- Improving query processing performance
  - Adding index to improve processing
- Managing data security, privacy, and integrity
- Data backup and recovery

TRENDS

- Increased used of procedural logic
- Proliferation of e-business applications
- Increase use of smartphones

DATA WAREHOUSE ADMINISTRATION

- New role, coming with growth in data warehouses
- Similar to DA/DBA roles
- Emphasis on integration and coordination of metadata/data across many data sources
- Specific roles:
  - Support decision support applications
  - Build a stable architecture – corporate information factory
  - Establish service level agreements regarding data warehouses and data marts

OPEN SOURCE DB MANAGEMENT

- Open Source DBMS: an alternative to proprietary packages such as Oracle, Microsoft SQL Server, or DB2
- Examples: MySQL, PostgreSQL
- Advantages:
  - Pool of volunteer developers and testers
  - Less expensive than proprietary packages
  - Source code available, for modification
- Disadvantages
  - Sometimes absence of complete documentation
  - Ambiguous licensing concerns
  - Vendors may not have certification programs

OPEN SOURCE DB MANAGEMENT (CONT.)

- Considerations when selecting an open source DBMS
  - Features: Support capabilities needed ?
  - Support: Are there any support or documentation?
  - Ease of use: Are there any tools to manage well
  - Stability: Does your database behave well overtime
  - Speed: Response time is quick or not
  - Training: Learn the use of RDBMS is easy or not

DATA SECURITY

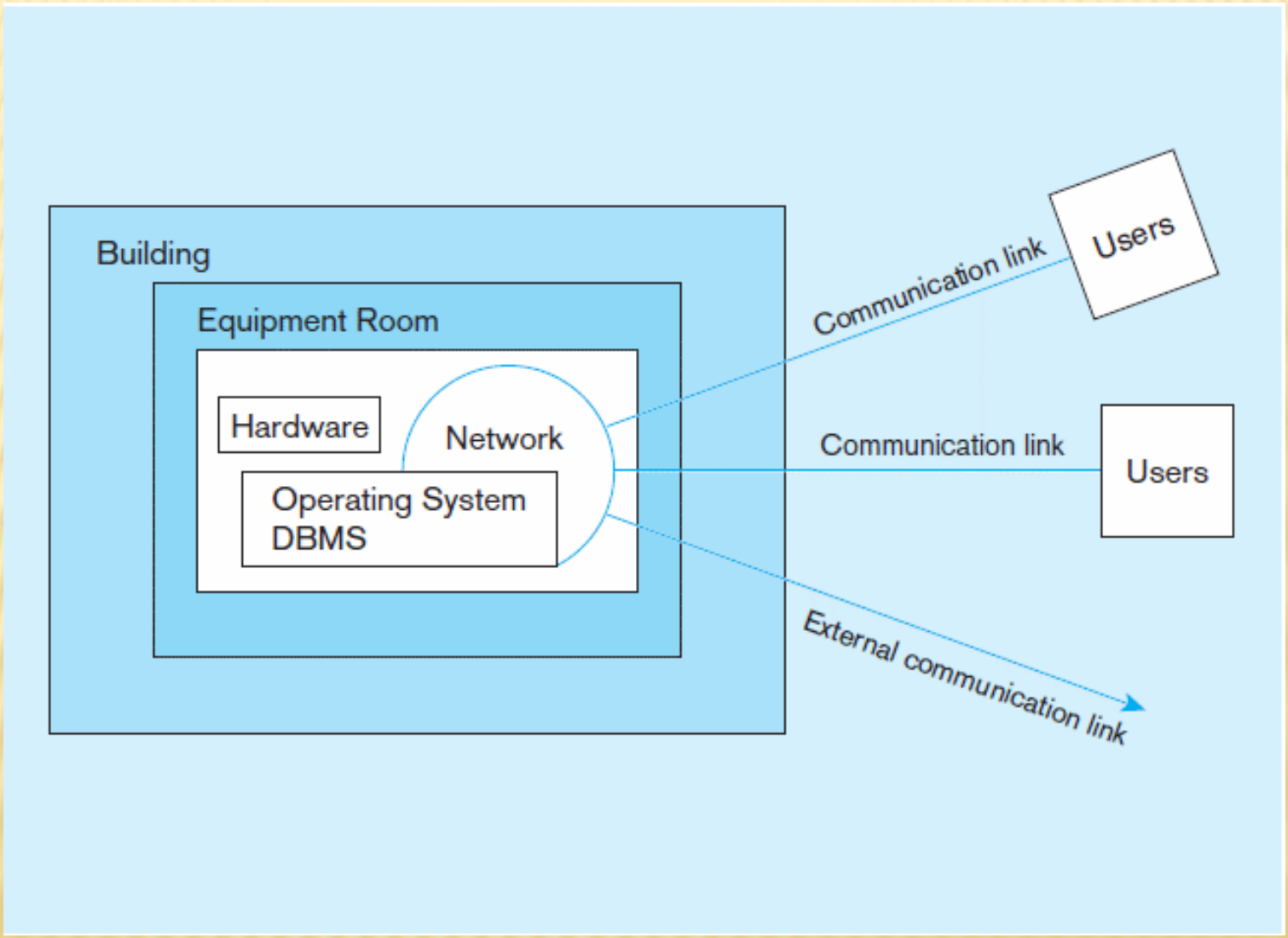
- **Database Security:** Protection of the data against accidental or intentional loss, destruction, or misuse

- Increased difficulty due to Internet access and client/server technologies

QUESTION:

- Provide some methods to protect data against unwanted access

Figure 12-2 Possible locations of data security threats



THREATS TO DATA SECURITY

- Accidental losses attributable to:
  - Human error
  - Software failure
  - Hardware failure
- Theft and fraud
- Loss of privacy or confidentiality
  - Loss of privacy such as stealing password
  - Loss of confidentiality leads to loss of competitiveness.

CONT

- Loss of data integrity:
  - Data is invalid or corrupted or incorrect
  - Hard to make decision based on wrong data.
- Loss of availability (e.g., through destruction)
  - Data is not available to end user when needed.
  - Caused by corrupted software to make system unusable.

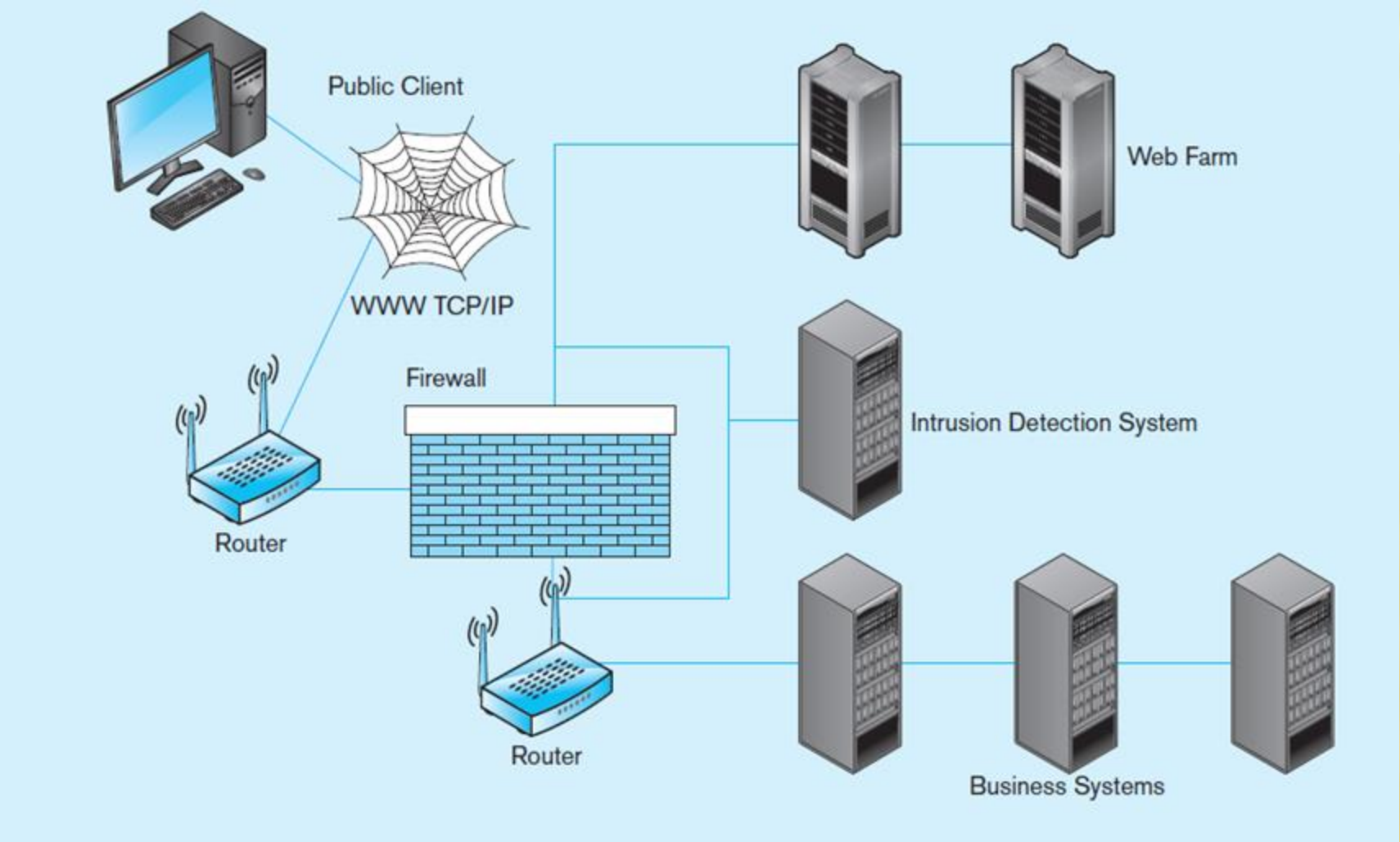
SERVER SECURITY

- Located in secured area
- Accessible only to authorized administrator and supervisors.
- When logged in, authenticated user only performs privilege based on role assigned, that privilege is assigned only to specific tables.

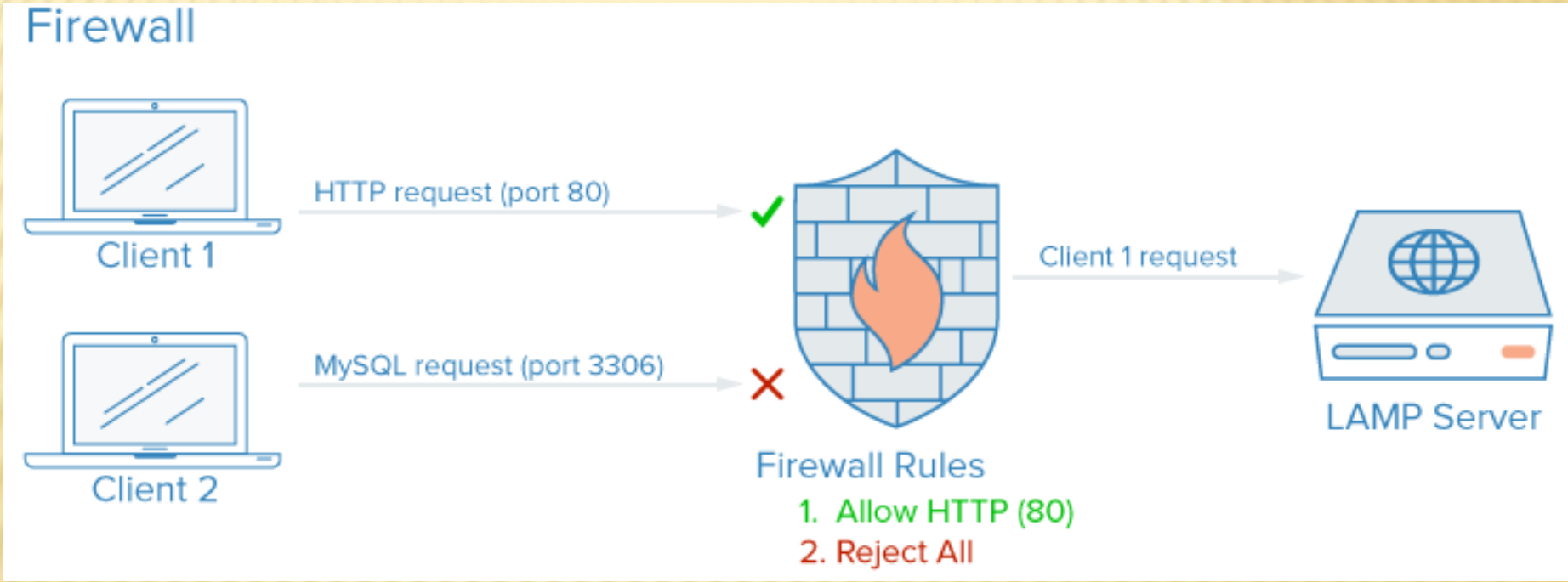
CLIENT – SERVER APPLICATION SECURITY

- Static HTML files are easy to secure
  - Standard database access controls
  - Place Web files in protected directories on server
- Dynamic pages are harder
  - User authentication
  - Session security
  - SSL for encryption
  - Restrict number of users and open ports. Limit number of superuser and admin.
  - Remove unnecessary programs because hacker may use that software to harm database

Figure 12-3 Establishing Internet Security



CLIENT SERVER SECURITY



W3C

- Platform for Privacy Protection (P3P)
- Addresses the following:
  - Who collects data
  - What data is collected and for what purpose
  - Who is data shared with
  - Can users control access to their data
  - How are disputes resolved
  - Policies for retaining data
  - Where are policies kept and how can they be accessed

DATABASE SOFTWARE SECURITY FEATURES

- Views or subschemas
- Integrity controls
- Authorization rules
- User-defined procedures
- Encryption
- Authentication schemes
- Backup, journalizing, and checkpointing

VIEWS AND INTEGRITY CONTROLS

- Views
  - Subset of the database that is presented to one or more users
  - User can be given access privilege to view without allowing access privilege to underlying tables
- Integrity Controls
  - Protect data from unauthorized use
  - Domains–set allowable values
  - Assertions–enforce database conditions
  - Triggers – prevent inappropriate actions, invoke special handling procedures, write to log files

AUTHORIZATION RULES

- Controls incorporated in the data management system
- Restrict:
  - access to data
  - actions that people can take on data
- Authorization matrix for:
  - Subjects
  - Objects
  - Actions
  - Constraints

Subject	Object	Action	Constraint
Sales Dept.	Customer record	Insert	Credit limit LE $5000
Order trans.	Customer record	Read	None
Terminal 12	Customer record	Modify	Balance due only
Acctg. Dept.	Order record	Delete	None
Ann Walker	Order record	Insert	Order aml LT $2000
Program AR4	Order record	Modify	None

Figure 12-4 Authorization matrix

Implementing authorization rules

Figure 12-5a Authorization table for subjects (salespersons)

	Customer records	Order records
Read	Y	Y
Insert	Y	Y
Modify	Y	N
Delete	N	N

Figure 12-5b Authorization table for objects (orders)

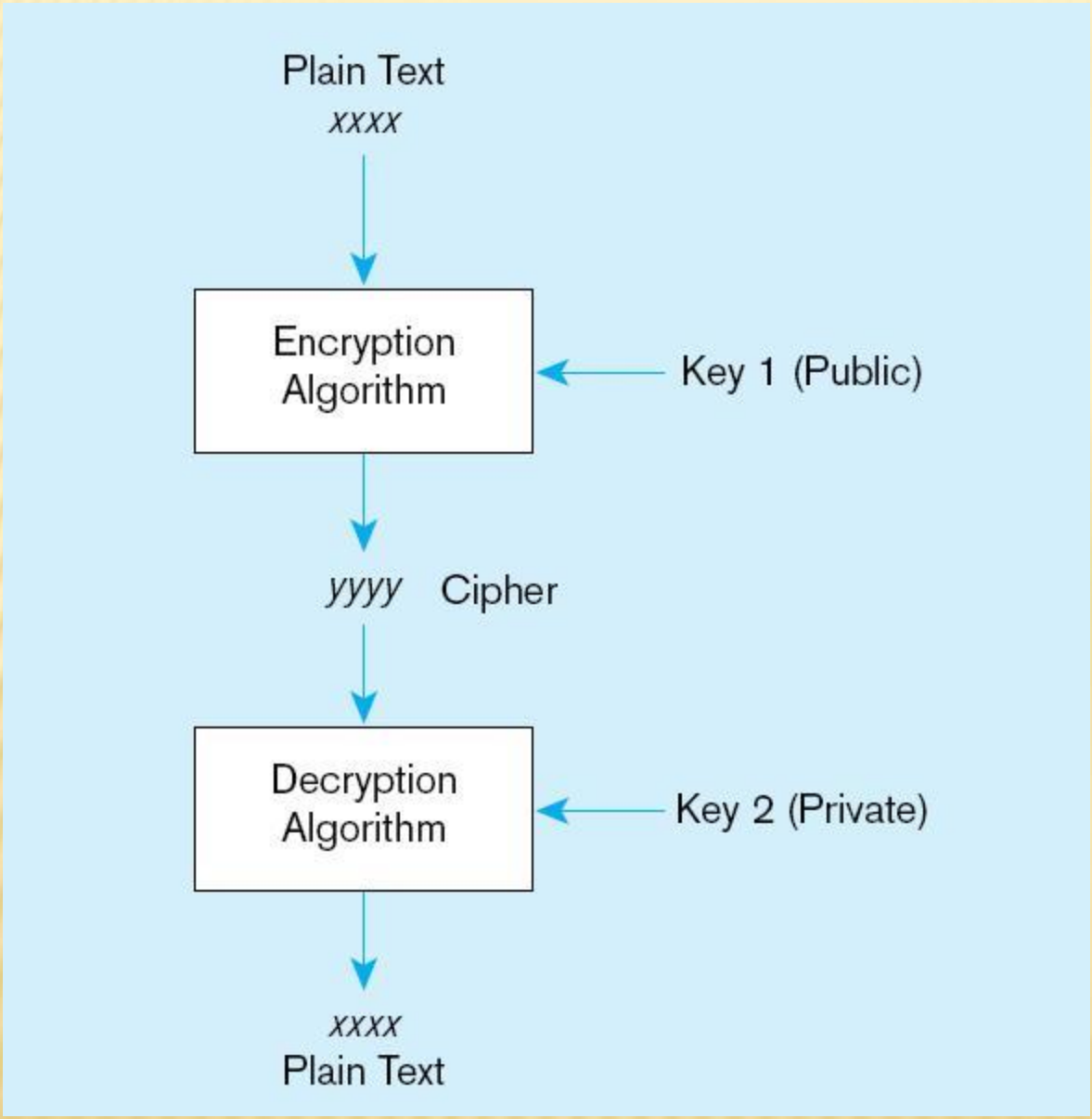
	Salespersons (password BATMAN)	Order entry (password JOKER)	Accounting (password TRACY)
Read	Y	Y	Y
Insert	N	Y	N
Modify	N	Y	Y
Delete	N	N	Y

Figure 11-6 Oracle privileges

Privilege	Capability
SELECT	Query the object.
INSERT	Insert records into the table/view. Can be given for specific columns.
UPDATE	Update records in table/view. Can be given for specific columns.
DELETE	Delete records from table/view.
ALTER	Alter the table.
INDEX	Create indexes on the table.
REFERENCES	Create foreign keys that reference the table.
EXECUTE	Execute the procedure, package, or function.

Some DBMSs also provide capabilities for ***user-defined procedures*** to customize the authorization process.

Figure 11-7 Basic two-key encryption



Encryption – the coding or scrambling of data so that humans cannot read them

Secure Sockets Layer (SSL) is a popular encryption scheme for TCP/IP connections.

AUTHENTICATION SCHEMES

- Goal – obtain a *positive* identification of the user
- Passwords: First line of defense
  - Should be at least 8 characters long
  - Should combine alphabetic and numeric data
  - Should not be complete words or personal information
  - Should be changed frequently

AUTHENTICATION SCHEMES (CONT.)

- Strong Authentication
  - Passwords are flawed:
    - Users share them with each other
    - They get written down, could be copied
    - Automatic logon scripts remove need to explicitly type them in
    - Unencrypted passwords travel the Internet
- Possible solutions:
  - Two factor–e.g., smart card plus PIN
  - Three factor–e.g., smart card, biometric, PIN

IT CHANGE MANAGEMENT

- The process by which changes to operational systems and databases are authorized
- For database, changes to: schema, database configuration, updates to DBMS software
- Segregation of duties: development, test, production

LOGICAL ACCESS TO DATA

- Personnel controls
  - Hiring practices, employee monitoring, security training, separation of duties
- Physical access controls
  - Swipe cards, equipment locking, check-out procedures, screen placement, laptop protection

IT OPERATIONS

- Policies and procedures for day-to-day management of infrastructure, applications, and databases in an organization
- Also involves vendor management
  - Review external maintenance agreements
  - Access source code? (if vendor goes out of business)

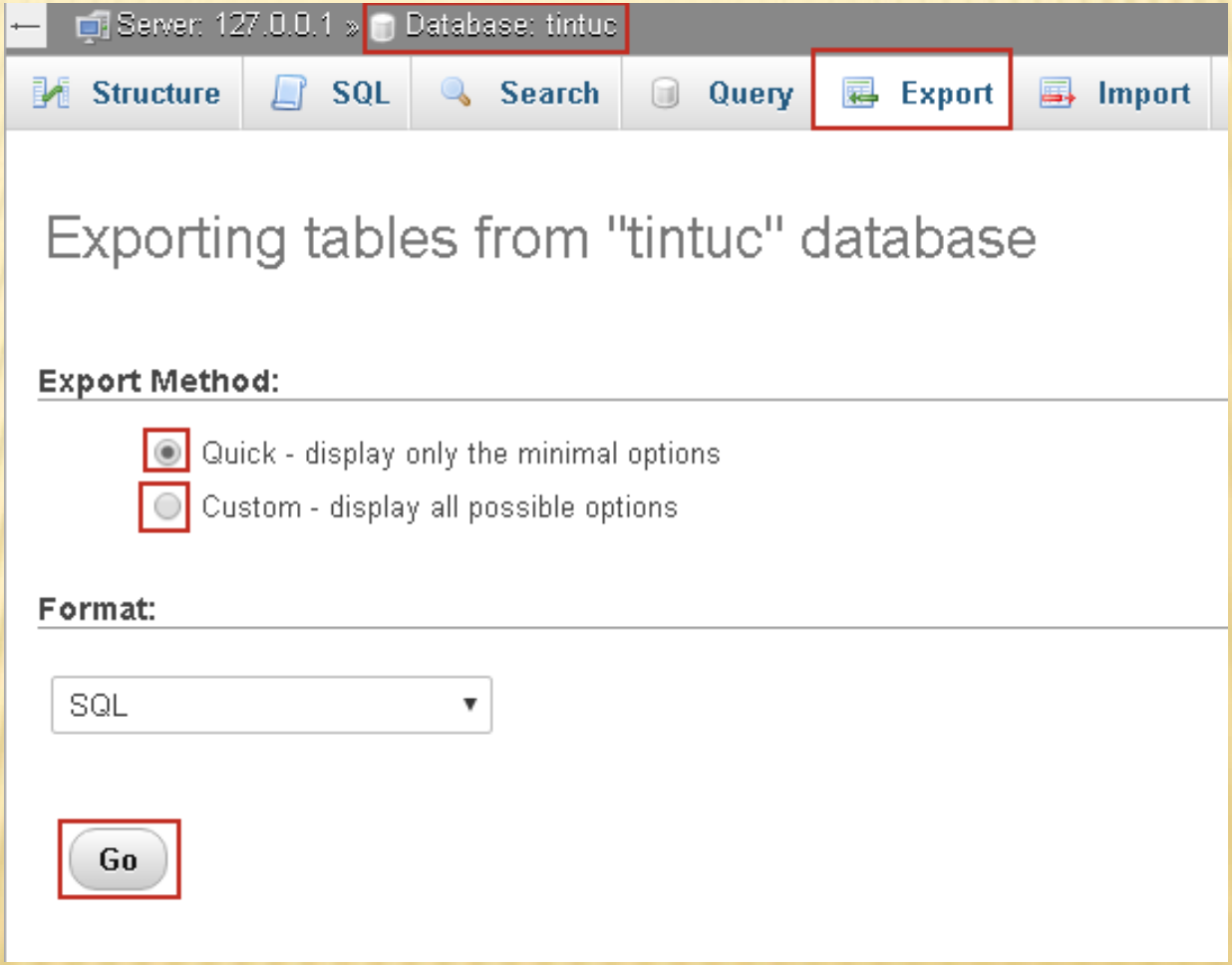
DATABASE RECOVERY

- Mechanism for restoring a database quickly and accurately after loss or damage
- Recovery facilities:
  - Backup Facilities
  - Journalizing Facilities
  - Checkpoint Facility
  - Recovery Manager

BACK-UP FACILITIES

- DBMS copy utility that produces backup copy of the entire database or subset
- Periodic backup (e.g. nightly, weekly)
- Cold backup–database is shut down during backup
- Hot backup–selected portion is shut down and backed up at a given time
- Backups stored in secure, off-site

PHPMYADMIN BACKUP

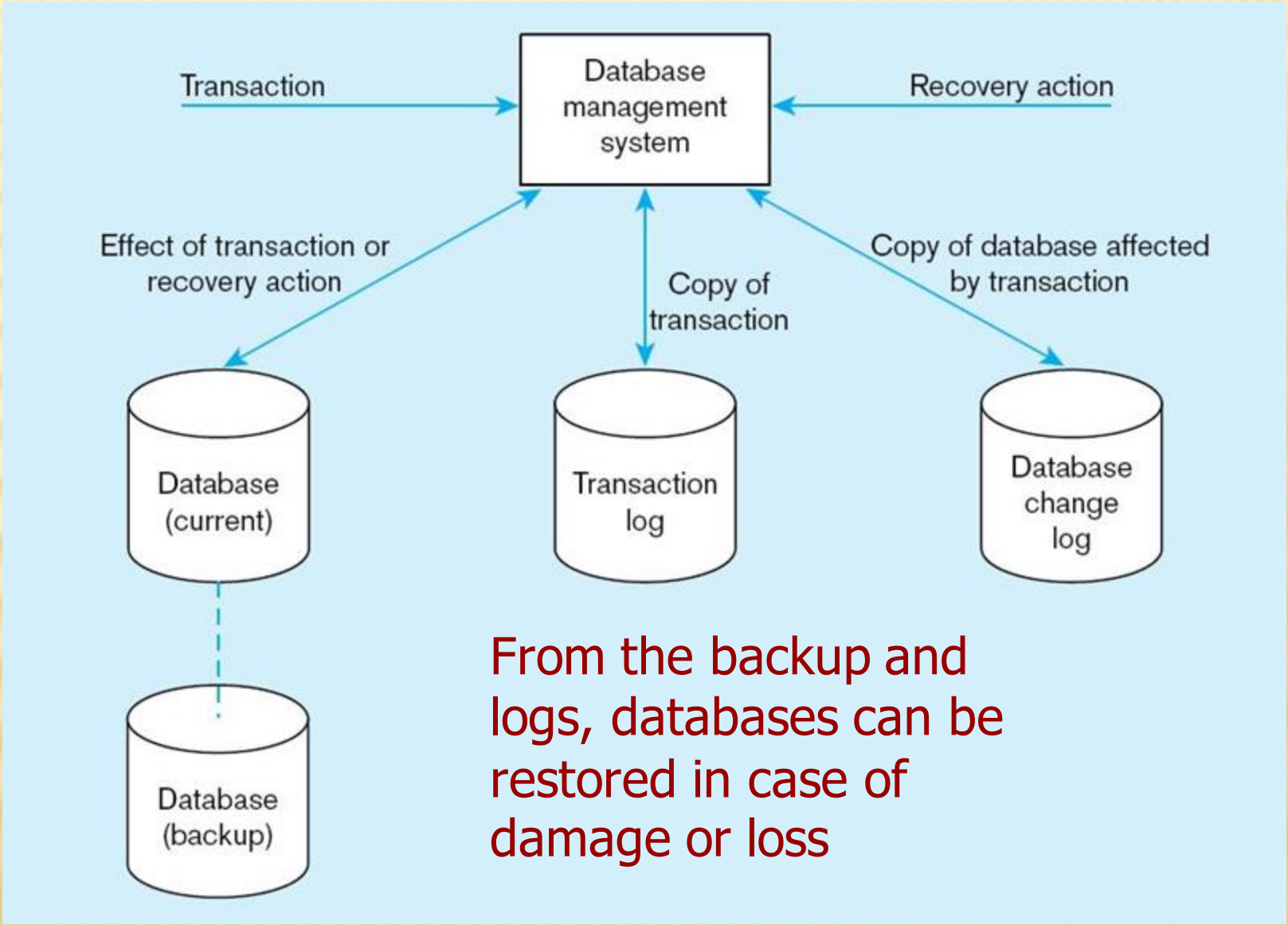


JOURNALIZING FACILITIES

- Audit trail of transactions and database updates
- Transaction log–record of essential data for each transaction processed against the database
- Database change log–images of updated data
  - Before-image–copy before modification
  - After-image–copy after modification

Produces an ***audit trail***

Figure 12-8 Database audit trail



CHECKPOINT FACILITIES

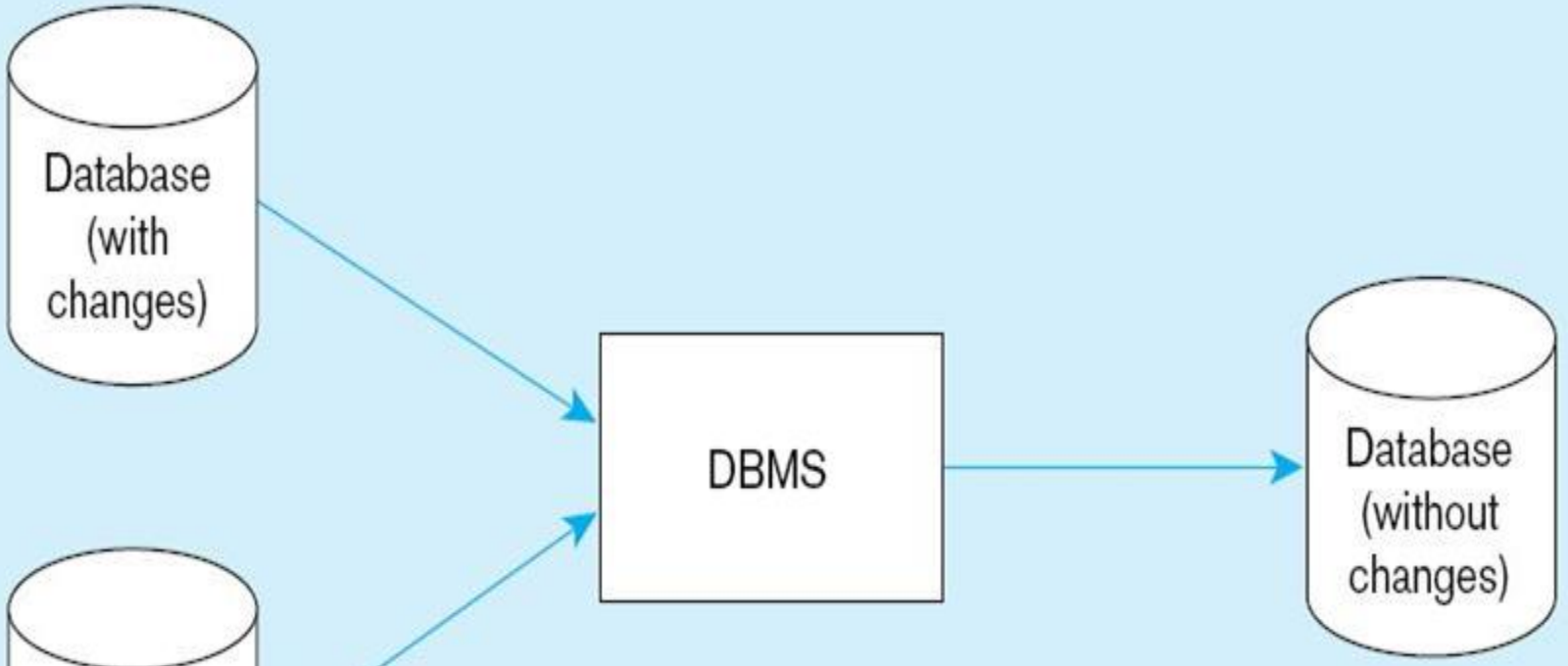
- DBMS periodically refuses to accept new transactions
- system is in a *quiet* state
- Database and transaction logs are synchronized

This allows recovery manager to resume processing from short period, instead of repeating entire day

RECOVER MANAGER

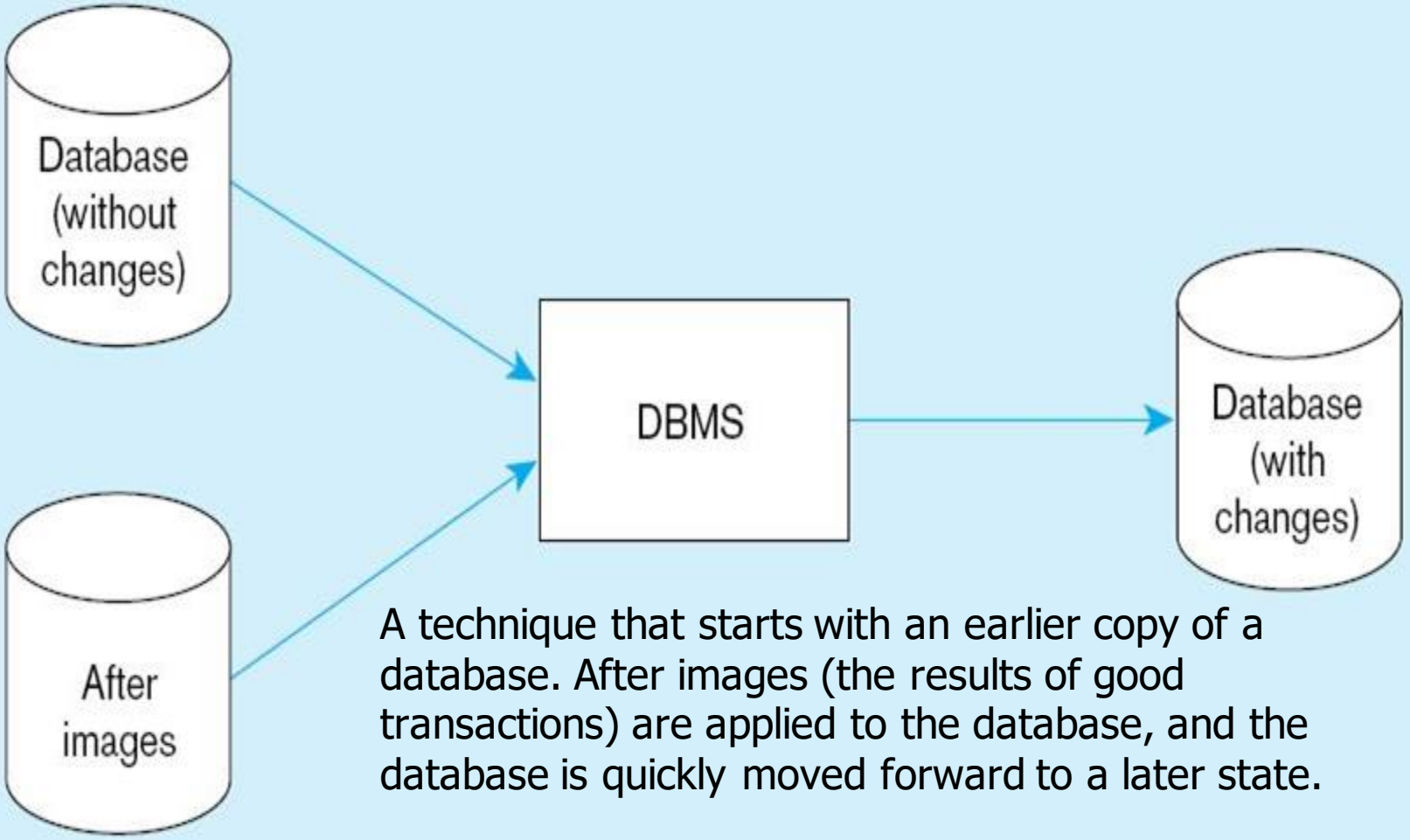
- Recovery Manager – DBMS module that restores the database to a correct condition when a failure occurs and then resumes processing user requests
- Recovery and Restart Procedures
  - Disk Mirroring–switch between identical copies of databases
  - Restore/Rerun–reprocess transactions against the backup (only done as a last resort)
  - Transaction Integrity–commit or abort all transaction changes
  - Backward Recovery (Rollback)–apply before images
  - Forward Recovery (Roll Forward)–apply after images (preferable to restore/rerun)

Figure 12-9 Basic recovery techniques
a) Rollback (backward recovery)



The backout, or undo, of unwanted changes to a database. Before images of the records that have been changed are applied to the database, and the database is returned to an earlier state. Rollback is used to reverse the changes made by transactions that have been aborted, or terminated abnormally.

Figure 12-9 Basic recovery techniques (cont.)
b) Rollforward (forward recovery)



A technique that starts with an earlier copy of a database. After images (the results of good transactions) are applied to the database, and the database is quickly moved forward to a later state.

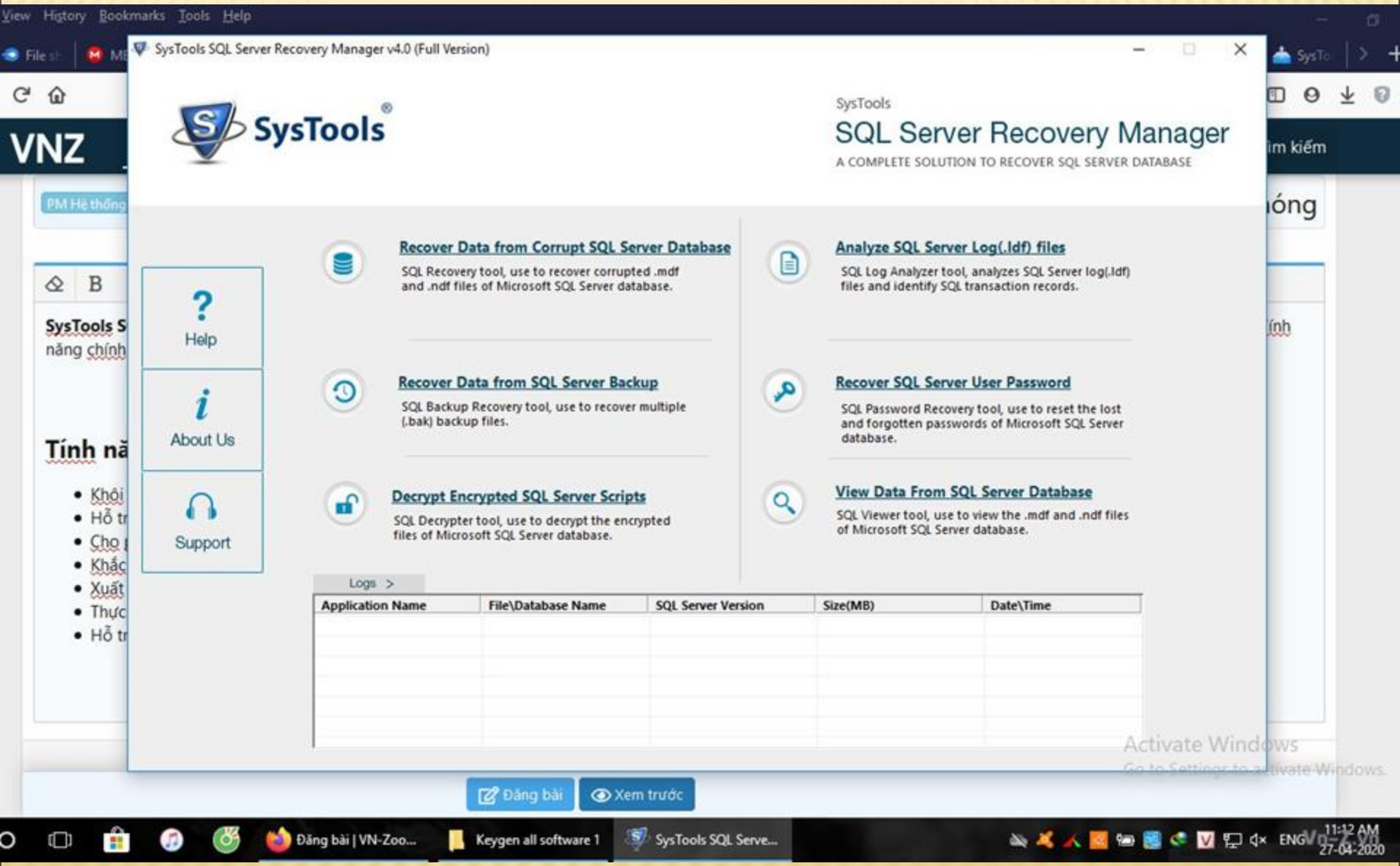


TABLE 12-1 Responses to Database Failure

Type of Failure	Recovery Technique
Aborted transaction	Rollback (preferred)
	Rollforward/return transactions to state just prior to abort
Incorrect data (update inaccurate)	Rollback (preferred)
	Reprocess transactions without inaccurate data updates
	Compensating transactions
System failure (database intact)	Switch to duplicate database (preferred)
	Rollback
	Restart from checkpoint (rollforward)
Database destruction	Switch to duplicate database (preferred)
	Rollforward
	Reprocess transactions

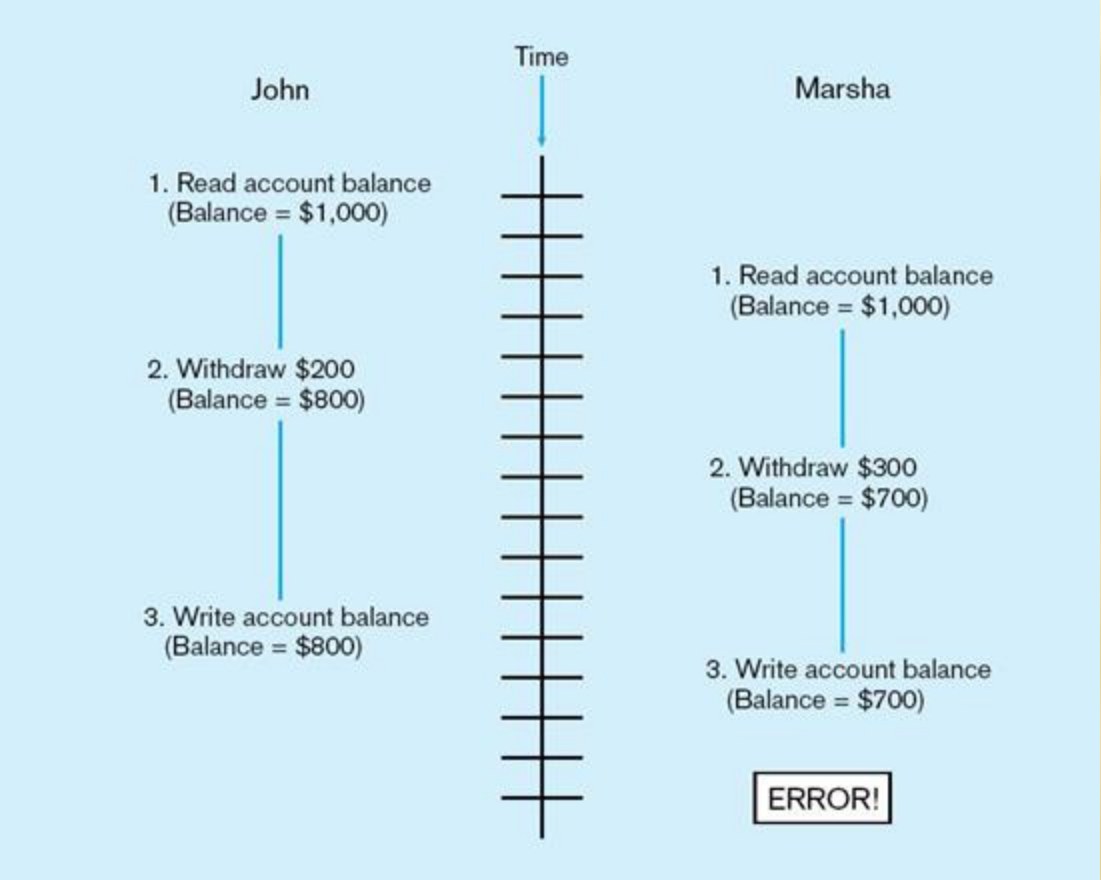
TRANSACTION ACID PROPERTIES

- Atomic
  - Transaction cannot be subdivided
- Consistent
  - Constraints don't change from before transaction to after transaction
- Isolated
  - Database changes not revealed to users until after transaction has completed
- Durable
  - Database changes are permanent

CONTROL CONCURRENT ACCESS

- *Problem*–in a multi-user environment, simultaneous access to data can result in interference and data loss (**lost update** problem)
- ***Solution*–Concurrency Control**
  - The process of managing simultaneous operations against a database so that data integrity is maintained and the operations do not interfere with each other in a multi-user environment

Figure 12-10 Lost update (no concurrency control in effect)

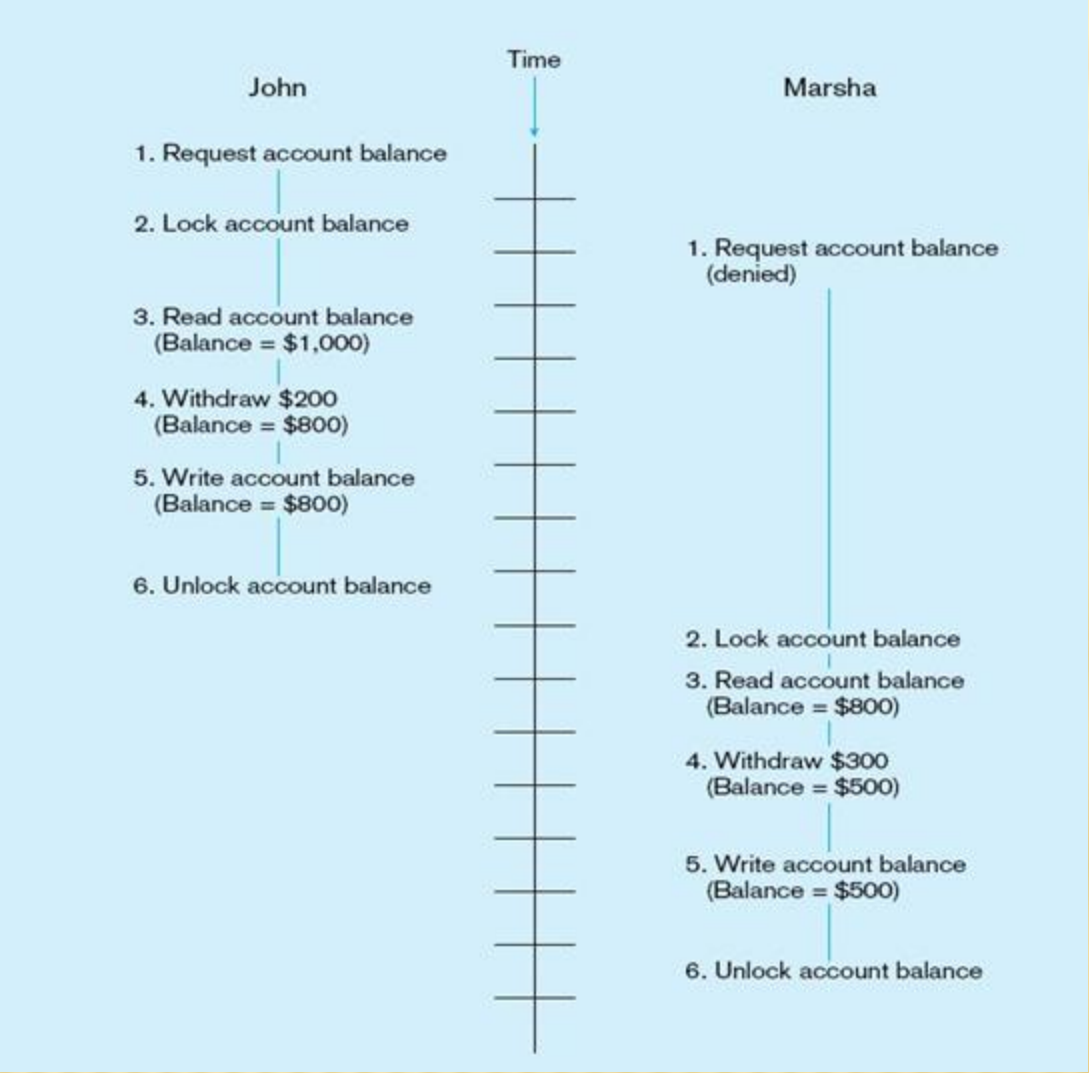


Simultaneous access causes updates to cancel each other. A similar problem is the **inconsistent read** problem.

CONCURRENCY CONTROL TECHNIQUES

- Serializability
  - Finish one transaction before starting another
- Locking Mechanisms
  - The most common way of achieving serialization
  - Data that is retrieved for the purpose of updating is locked for the updater
  - No other user can perform update until unlocked

Figure 12-11: Updates with locking (concurrency control)



This prevents the lost update problem

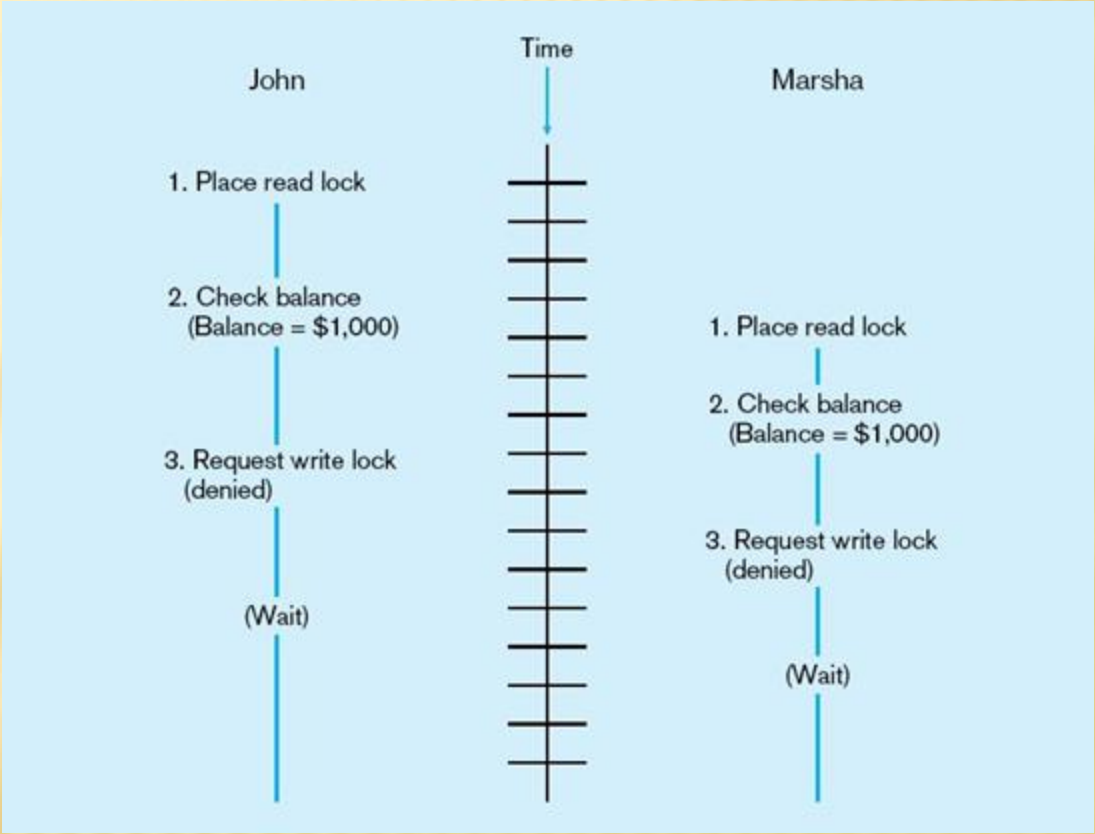
LOCKING MECHANISMS

- Locking level:
  - Database–used during database updates
  - Table–used for bulk updates
  - Block or page–physical storage block contain requested record is blocked
  - Record–only requested row; fairly commonly used
  - Field–requires significant overhead; impractical
- Types of locks:
  - Shared lock (Read lock)–Read but no update permitted. Used when just reading to prevent another user from placing an exclusive lock on the record
  - Exclusive lock (Write lock)–No access permitted. Prevent from reading. Used when

DEADLOCK

- An impasse that results when two or more transactions have locked common resources, and each waits for the other to unlock their resources

Figure 12-12
The problem of deadlock



John and Marsha will wait forever for each other to release their locked resources!

MANAGING DEADLOCK

- Deadlock prevention:
  - Lock all records required at the beginning of a transaction
  - Two-phase locking protocol
    - Growing phase
    - Shrinking phase
  - May be difficult to determine all needed resources in advance
- Deadlock Resolution:
  - Allow deadlocks to occur
  - Mechanisms for detecting and breaking them
    - Resource usage matrix

VERSIONING

- Optimistic approach to concurrency control
- Instead of locking
- Assumption is that simultaneous updates will be infrequent
- Each transaction can attempt an update as it wishes
- The system will create a new version of a record instead of replacing the old one
- When a conflict occurs, accept one user's update and inform the other user that its update needs to be tried again.
- Use of rollback and commit for this

Figure 12-14 The use of versioning

Better performance than locking

DATABASE PERFORMANCE TUNING

- DBMS Installation
  - Setting installation parameters
- Memory and Storage Space Usage
  - Set cache levels
  - Choose background processes
  - Data archiving
- Input/output (I/O) Contention
  - Use striping
  - Distribution of heavily accessed files
- CPU Usage – Monitor CPU load
- Application tuning
  - Modification of SQL code in applications
  - Use of heartbeat queries

COST OF DOWNTIME

Downtime is expensive

TABLE 12-2 Cost of Downtime, by Type of Business

Industry/Type of Business	Approximate Estimated Hourly Cost
Financial services/Brokerage operations	$7 million
Financial services/Electronic transactions (card) processing	$2.5 million
Retail/Tele-sales	$115,000
Travel/Reservation Centers	$90,000
Logistics/Shipping Services	$28,000

Based on: Mullins (2002), p. 226

TABLE 12-3 Cost of Downtime, by Availability

	Downtime Per Year		
Availability	Minutes	Hours	Cost Per Year
99.999%	5	.08	$8,000
99.99%	53	.88	$88,000
99.9%	526	8.77	$877,000
99.5%	2,628	43.8	$4,380,000
99%	5,256	87.6	$8,760,000

Based on: Mullins (2002), p. 226

DATA AVAILABILITY

- How to ensure availability
  - Hardware failures–provide redundancy for fault tolerance
  - Loss of data–database mirroring
  - Human error–standard operating procedures, training, documentation
  - Maintenance downtime–automated and non-disruptive maintenance utilities
  - Network problems–careful traffic monitoring, firewalls, and routers