Chapter 12: Data and Database Administration

Modern Database Management

6th Edition

Jeffrey A. Hoffer, Mary B. Prescott, Fred R. McFadden

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

Data Administration Functions

- Data policies, procedures, standards
- Planning
- Data conflict (ownership) resolution
- Internal marketing of DA concepts
- Managing the data repository

Database Administration Functions

- Selection of hardware and software
- Installing/upgrading DBMS
- Tuning database performance
- Improving query processing performance
- Managing data security, privacy, and integrity
- Data backup and recovery

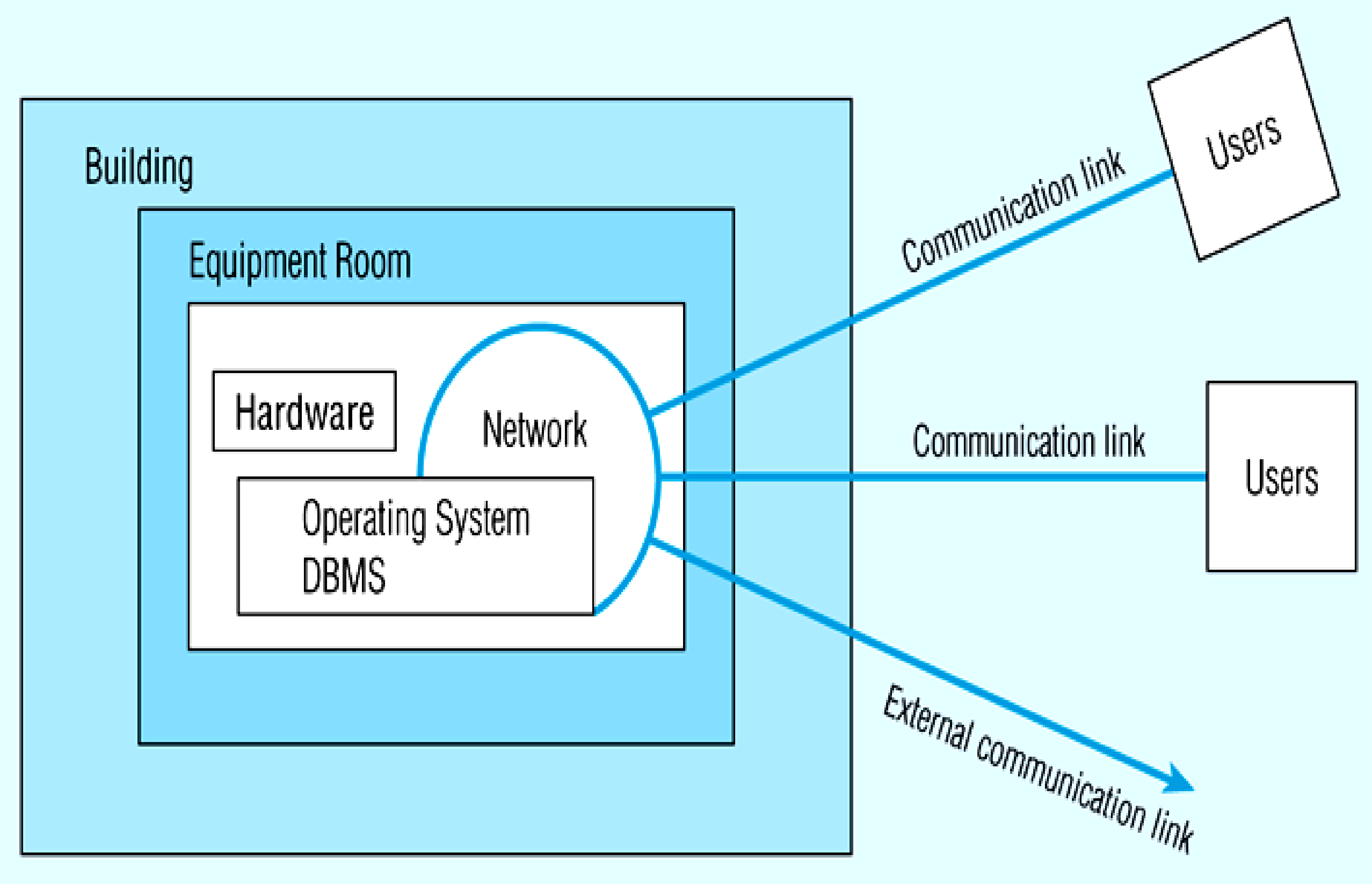
Data Warehouse Administration

- New role, coming with the 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
  - Manage data warehouse growth
  - Establish service level agreements regarding data warehouses and data marts

Database 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

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.

Improper data access:

- Loss of privacy (personal data)
- Loss of confidentiality (corporate data)

Loss of data integrity

Loss of availability (through, e.g. sabotage)

Data Management Software Security Features

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

Views and Integrity Controls

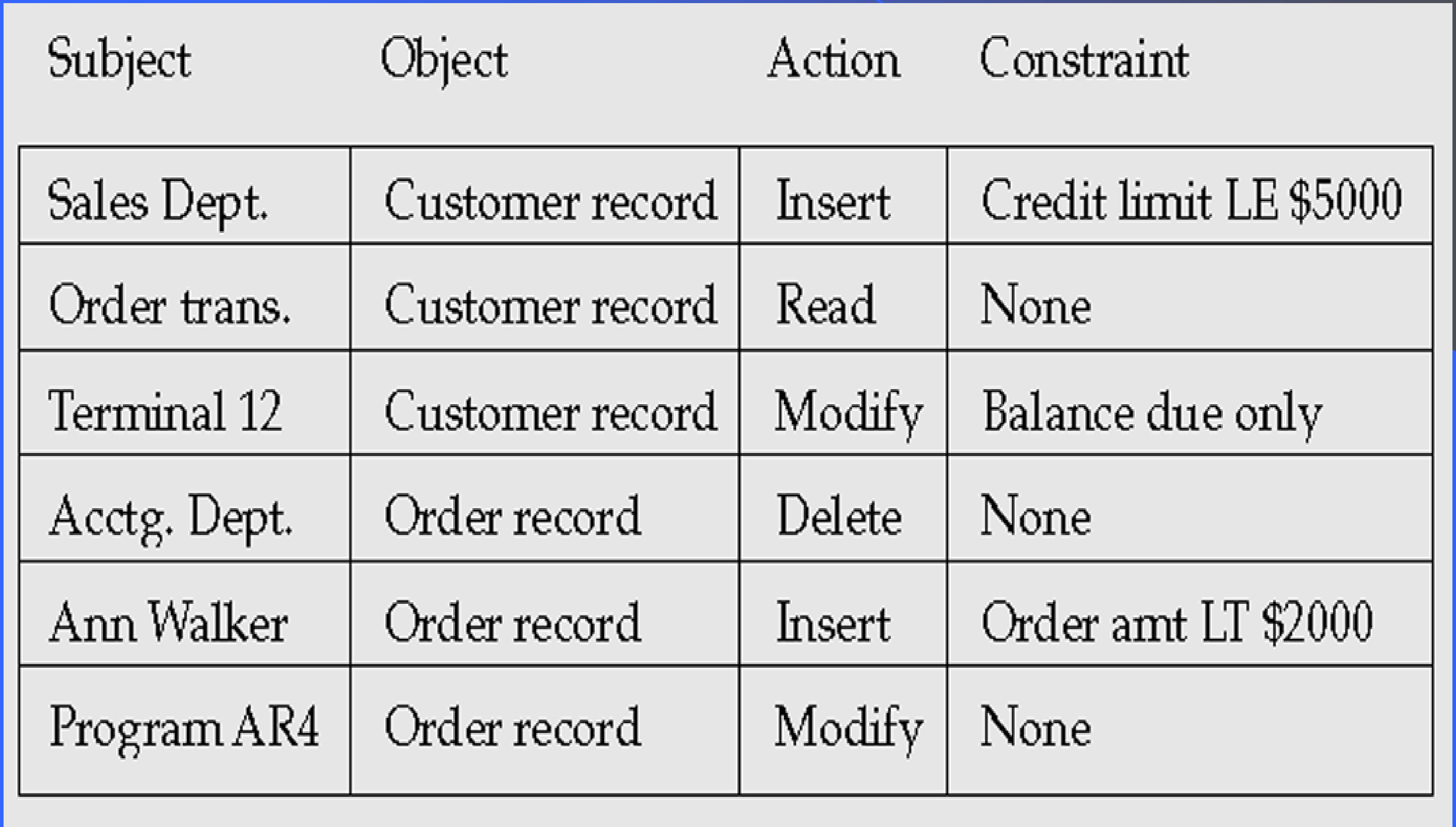
- Views
  - Subset of the database that is presented to one or more users
  - User can be given access privilege to view without allowing access privilege to underlying tables
- Integrity Controls
  - Protect data from unauthorized use
  - Domains – set allowable values
  - Assertions – enforce database conditions

Authorization Rules

Controls incorporated in the data management system

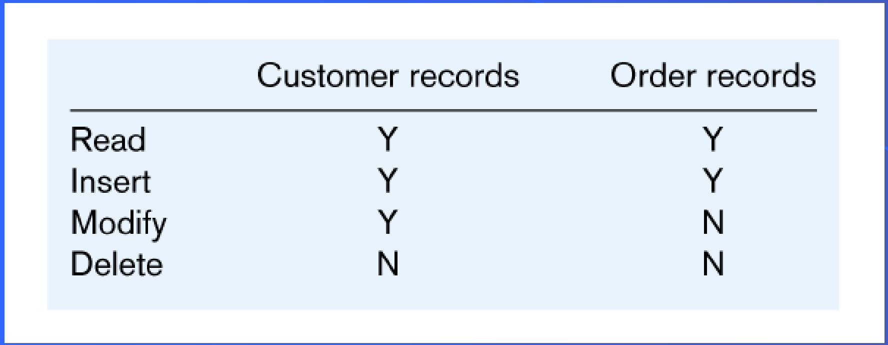
- ➔ Restrict:
  - access to data
  - actions that people can take on data
- ➔ Authorization matrix for:
  - Subjects
  - Objects
  - Actions
  - Constraints

Figure 12-3: Authorization matrix



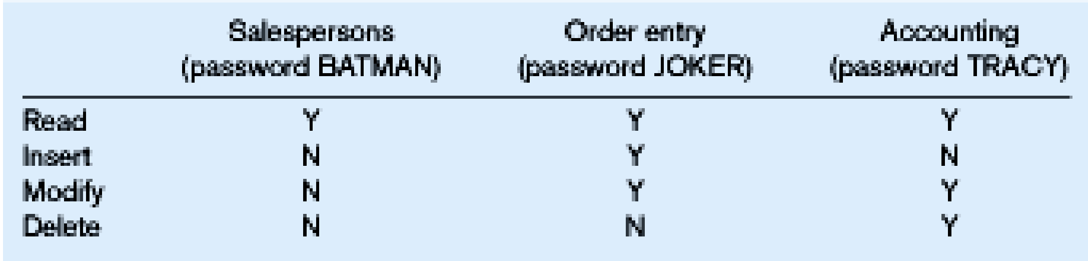
Subject	Object	Action	Constraint
Sales Dept.	Customer record	Insert	Credit limit LE $5000
Order trans.	Customer record	Read	None
Terminal 12	Customer record	Modify	Balance due only
Acctg. Dept.	Order record	Delete	None
Ann Walker	Order record	Insert	Order amt LT $2000
Program AR4	Order record	Modify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

Figure 12-4(a): Authorization table for subjects



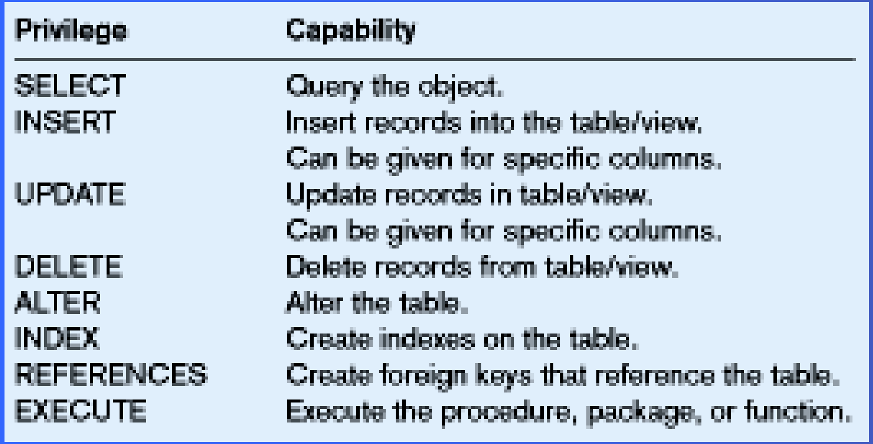
	Customer records	Order records
Read	Y	Y
Insert	Y	Y
Modify	Y	N
Delete	N	N

Figure 12-4(b): Authorization table for objects



	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 12-5: Oracle8i 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

Authentication Schemes

- Goal – obtain a *positive* identification of the user
- 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:
  - Biometric devices – use of fingerprints, retinal scans, etc. for positive ID
  - Third-party authentication – using secret keys, digital certificates

Database Recovery

Mechanism for restoring a database quickly and accurately after loss or damage

Recovery facilities:

- Backup Facilities
- Journalizing Facilities
- Checkpoint Facility
- Recovery Manager

Backup Facilities

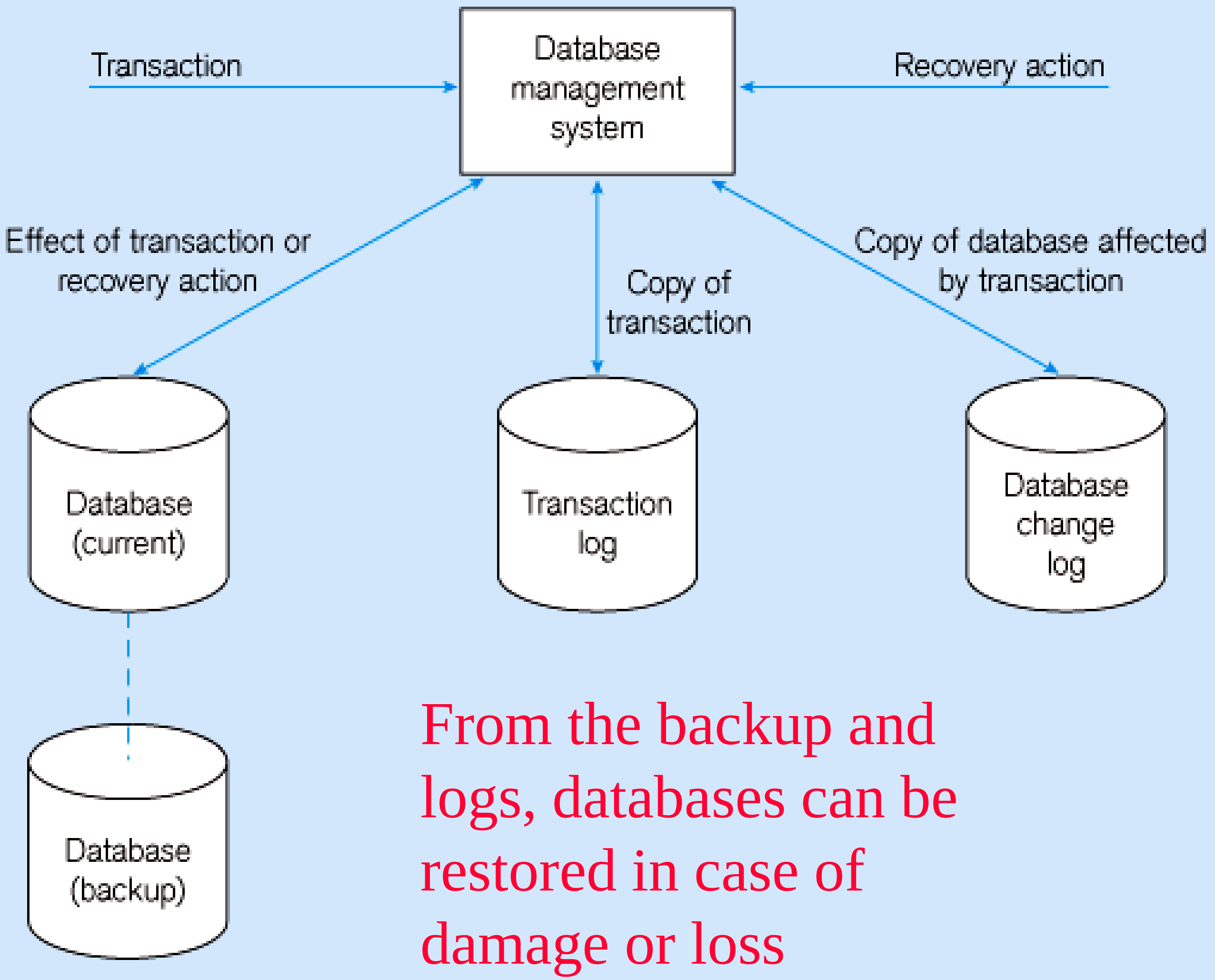
- Automatic dump facility that produces backup copy of the entire database
- Periodic backup (e.g. nightly, weekly)
- Cold backup – database is shut down during backup
- Hot backup – selected portion is shut down and backed up at a given time
- Backups stored in secure, off-site location

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 12-6: Database audit trail



From the backup and logs, databases can be restored in case of damage or loss

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

Recovery and Restart Procedures

Switch - Mirrored databases

Restore/Rerun - Reprocess transactions against the backup

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 12-7: Basic recovery techniques
(a) Rollback

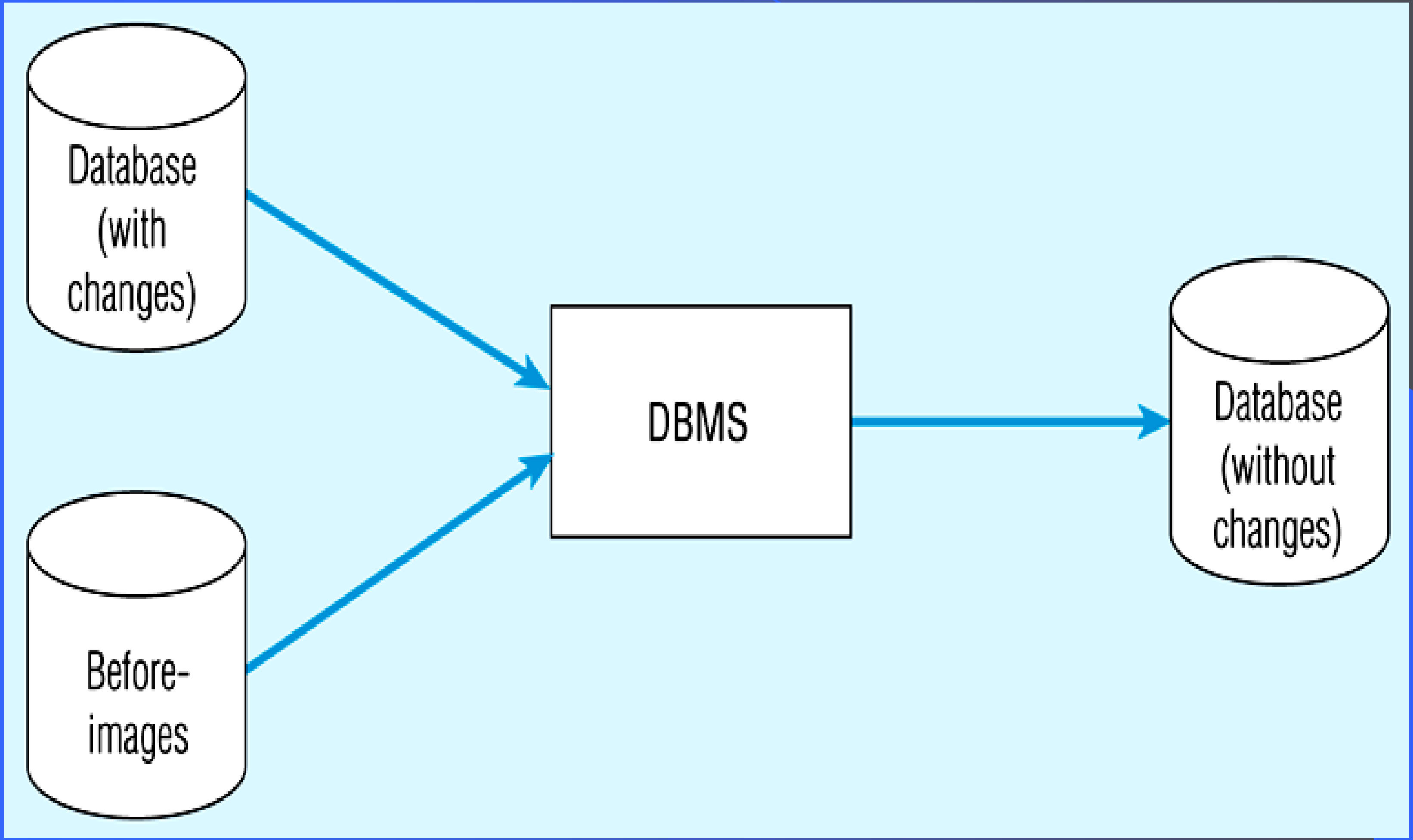
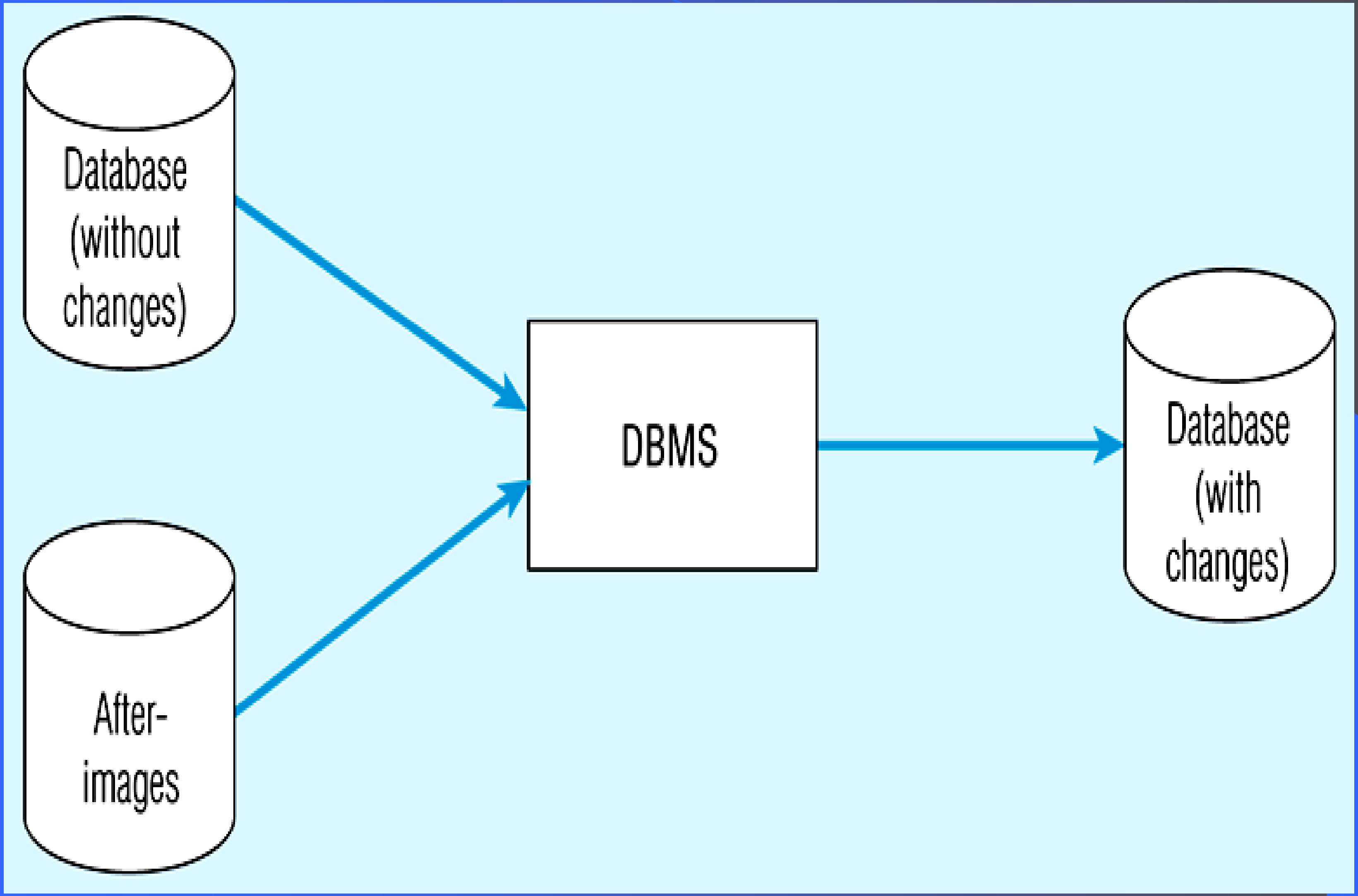


Figure 12-7(b) Rollforward



Database Failure Responses

Aborted transactions

- Preferred recovery: rollback
- Alternative: Rollforward to state just prior to abort

Incorrect data

- Preferred recovery: rollback
- Alternative 1: re-run transactions not including inaccurate data updates
- Alternative 2: compensating transactions

System failure (database intact)

- Preferred recovery: switch to duplicate database
- Alternative 1: rollback
- Alternative 2: restart from checkpoint

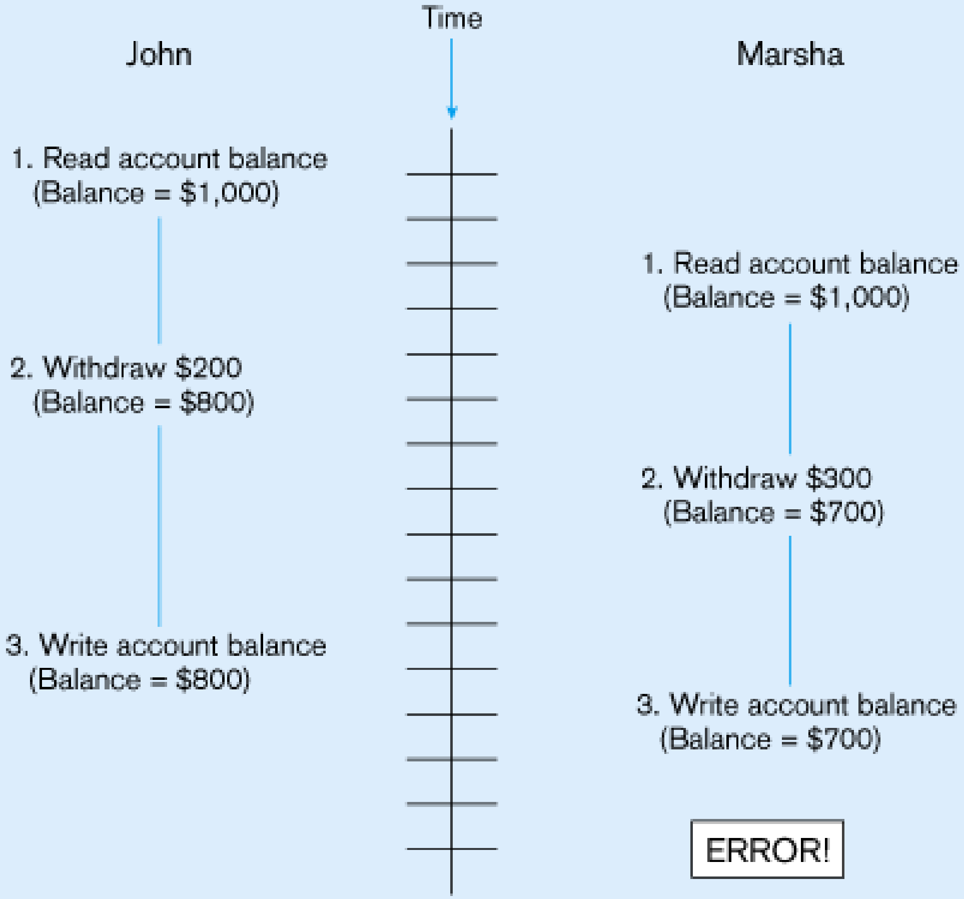
Database destruction

- Preferred recovery: switch to duplicate database
- Alternative 1: rollforward
- Alternative 2: reprocess transactions

Concurrency Control

- *Problem* – in a multi-user environment, simultaneous access to data can result in interference and data loss
- ***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 12-8: **LOST UPDATE**



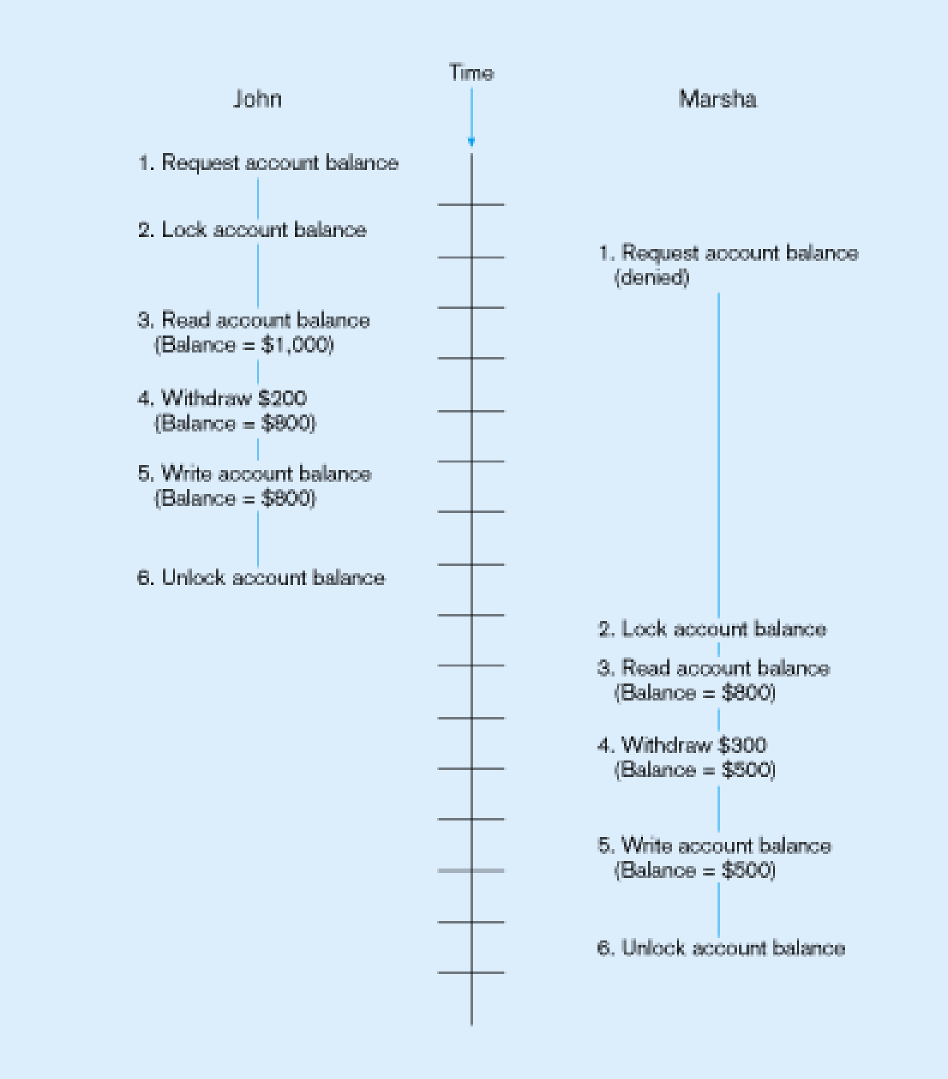
Simultaneous access causes updates to cancel each other

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 12-9: Updates with locking for concurrency control



Time

John

1. Request account balance
2. Lock account balance
3. Read account balance
   (Balance = $1,000)
4. Withdraw $200
   (Balance = $800)
5. Write account balance
   (Balance = $800)
6. Unlock account balance

Marsha

1. Request account balance
   (denied)
2. Lock account balance
3. Read account balance
   (Balance = $800)
4. Withdraw $300
   (Balance = $500)
5. Write account balance
   (Balance = $500)
6. Unlock account balance

This prevents the lost update problem

Locking Mechanisms

Locking level:

- Database – used during database updates
- Table – used for bulk updates
- Block or page – very commonly used
- Record – only requested row; fairly commonly used
- Field – requires significant overhead; impractical

Types of locks:

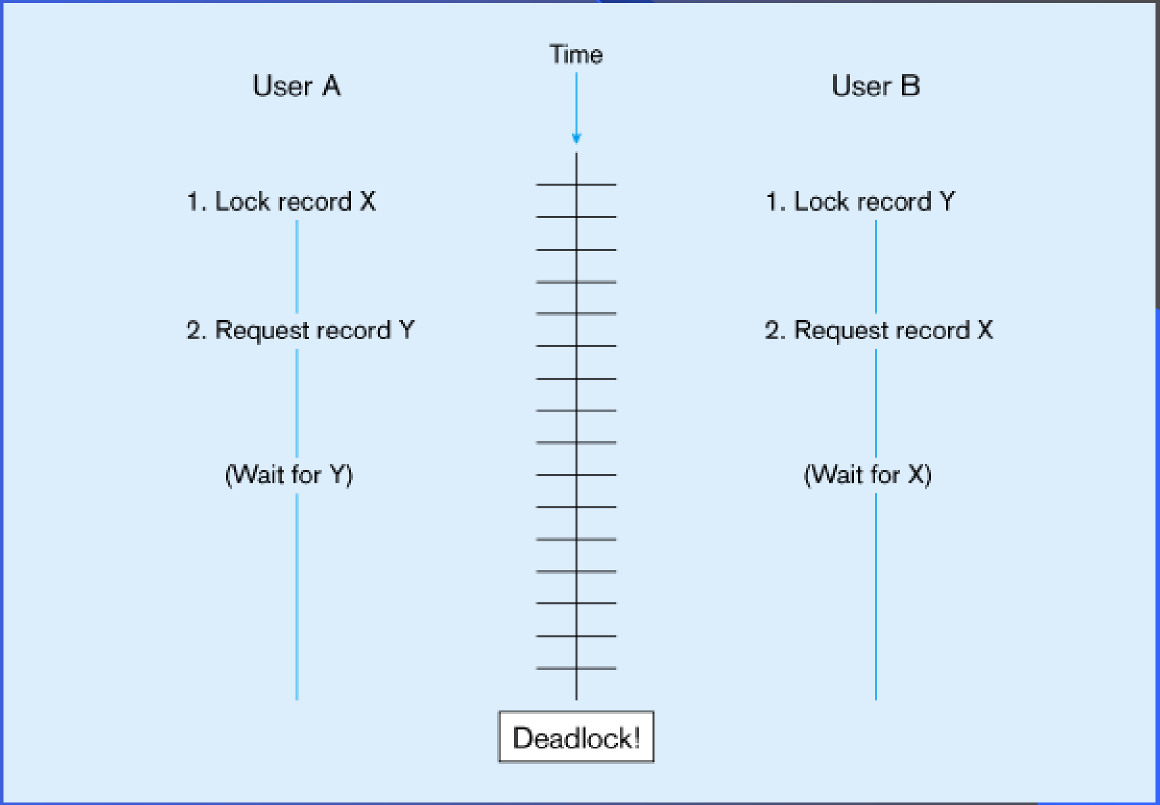
- Shared lock - Read but no update permitted. Used when just reading to prevent another user from placing an exclusive lock on the record
- Exclusive lock - No access permitted. Used when preparing to update

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 12-11
A deadlock situation

UserA and UserB 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

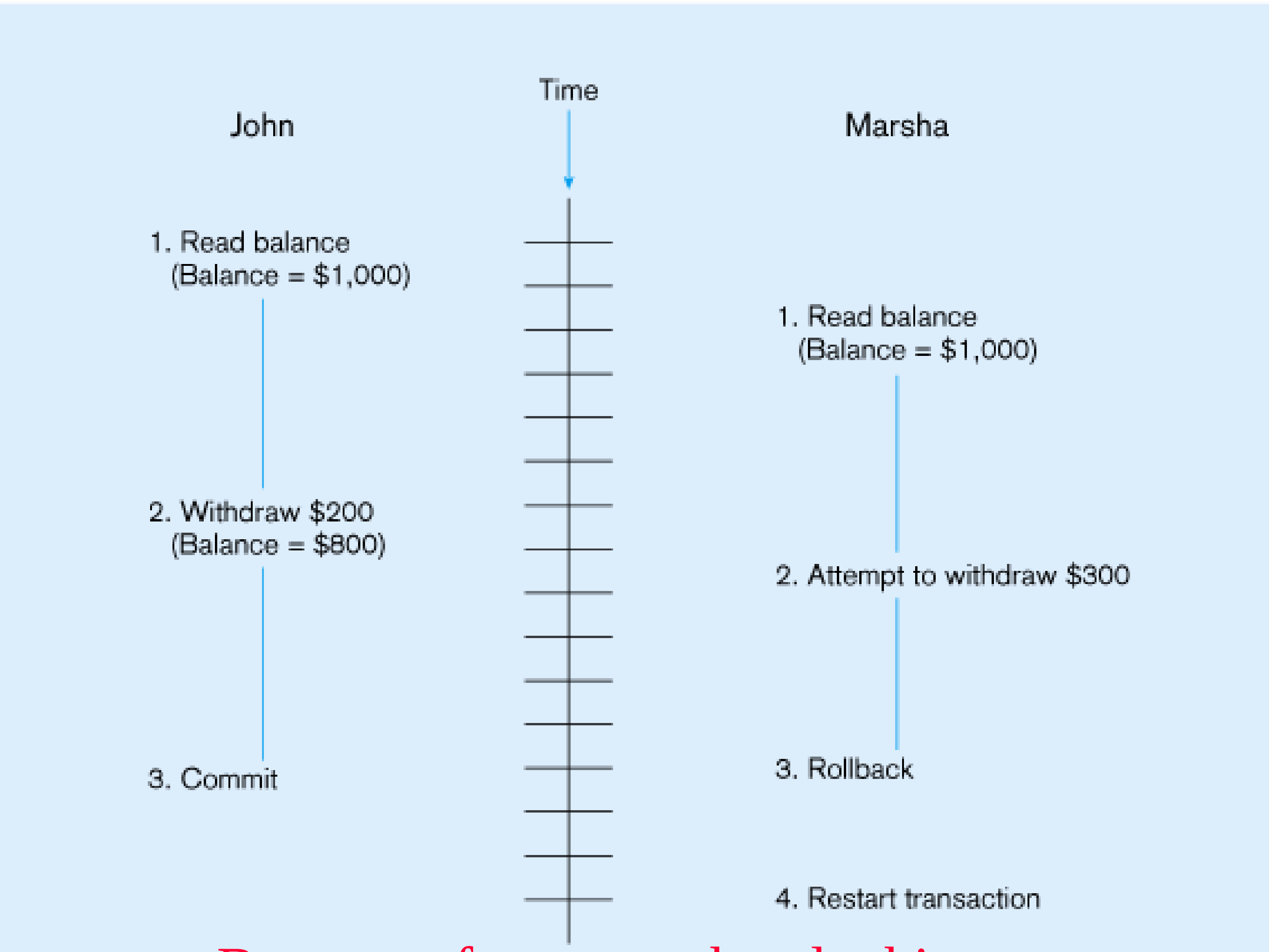
Deadlock Resolution:

- 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 reject an update when it senses a conflict
- Use of rollback and commit for this

Figure 12-12: the use of versioning



Better performance than locking

Managing Data Quality

- ***Data Steward*** - Liaisons between IT and business units

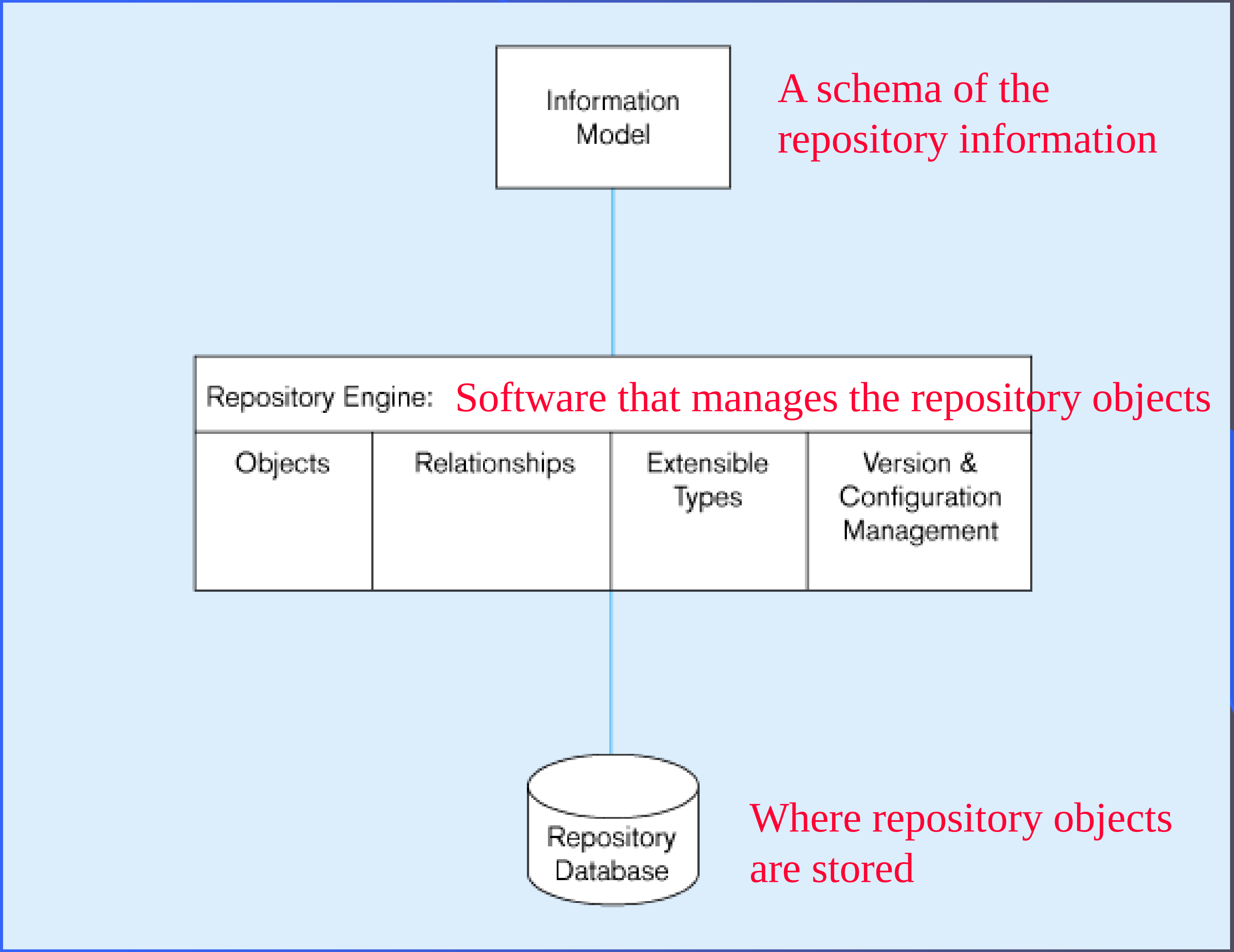
Five Data Quality Issues:

- ✓ Security policy and disaster recovery
- ✓ Personnel controls
- ✓ Physical access controls
- ✓ Maintenance controls (hardware & software)
- ✓ Data protection and privacy

Data Dictionaries and Repositories

- Data dictionary
  - Documents data elements of a database
- System catalog
  - System-created database that describes all database objects
- Information Repository
  - Stores metadata describing data and data processing resources
- Information Repository Dictionary System (IRDS)
  - Software tool managing/controlling access to information repository

Figure 12-13: Three components of the repository system architecture



A schema of the repository information

Software that manages the repository objects

Where repository objects are stored

Source: adapted from Bernstein, 1996.

Database Performance Tuning

- DBMS Installation
  - Setting installation parameters
- Memory Usage
  - Set cache levels
  - Choose background processes
- Input/Output Contention
  - Use striping
  - Distribution of heavily accessed files
- CPU Usage
  - Monitor CPU load
- Application tuning
  - Modification of SQL code in applications