Database Concepts (VI)

Selected Database Issues

Chaokun Wang

School of Software, Tsinghua University
chaokun@tsinghua.edu.cn

May 10, 2021

Outline

- Transaction Management*
- Database Security
- Database Administration

Motivation

- read(bal_x)
- read($item_z$)
- $bal_x = bal_x - 10$
- $item_z = item_z + 1$
- write(bal_x)
- write($item_z$)

Motivation

- read(bal_x) // Mom
- $bal_x = bal_x + 100$ // Mom
- read(bal_x) // Tom
- write(bal_x) // Mom
- $bal_x = bal_x – 10$ // Tom
- write(bal_x) // Tom

这个safe吗？
· 如果用transaction就可以了
· transaction1可以完成
transaction2会显示“由于同步更新而无法完成”

What is a Transaction? 可以保证数据库系统可以正常的运行-并发控制

工作的基本单元，可以由下面的这些东西组成

- A logical unit of work that must be entirely completed or aborted 核心就是，要么这些都干了，要么都没干
  - SELECT statement

e.g.

  - Series of related UPDATE statements
  - Series of INSERT statements
  - Combination of SELECT, UPDATE, and INSERT statements

A Transaction

- Consistent database state
  - All data integrity constraints are satisfied
  - Must begin with the database in a known consistent state to ensure consistency
  - Most are formed by two or more database requests
    - Database requests: equivalent of a single SQL statement in an application program or transaction

Transaction Properties “酸性”

- Atomicity
  - All operations (SQL requests) of a transaction be completed; if not, the transaction is aborted.
- Consistency
  - A transaction takes a database from one consistent state to another consistent state.
  - If any of the transaction parts violates an integrity constraint, the entire transaction is aborted.
- Isolation
  - The data used during the execution of a transaction cannot be used by a second transaction until the first one is completed.
- Durability
  - Once transaction changes are done (committed), they cannot be undone or lost, even in the event of a system failure.

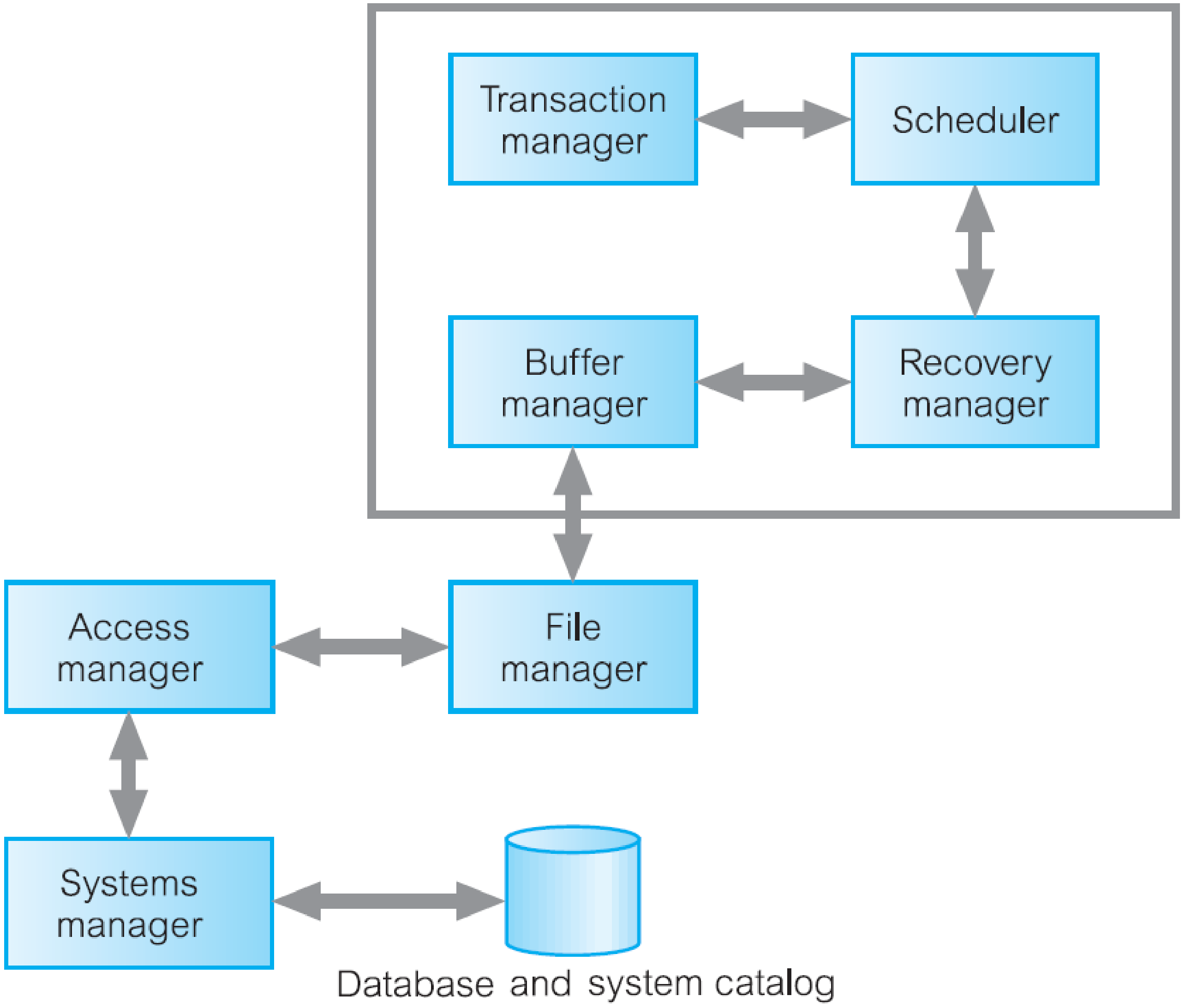
Transaction Management with SQL

- SQL statements that provide transaction support:
  - COMMIT 提交
  - ROLLBACK 后退

TM in PG:
- BEGIN
- SAVEPOINT 当事务非常非常大的时候

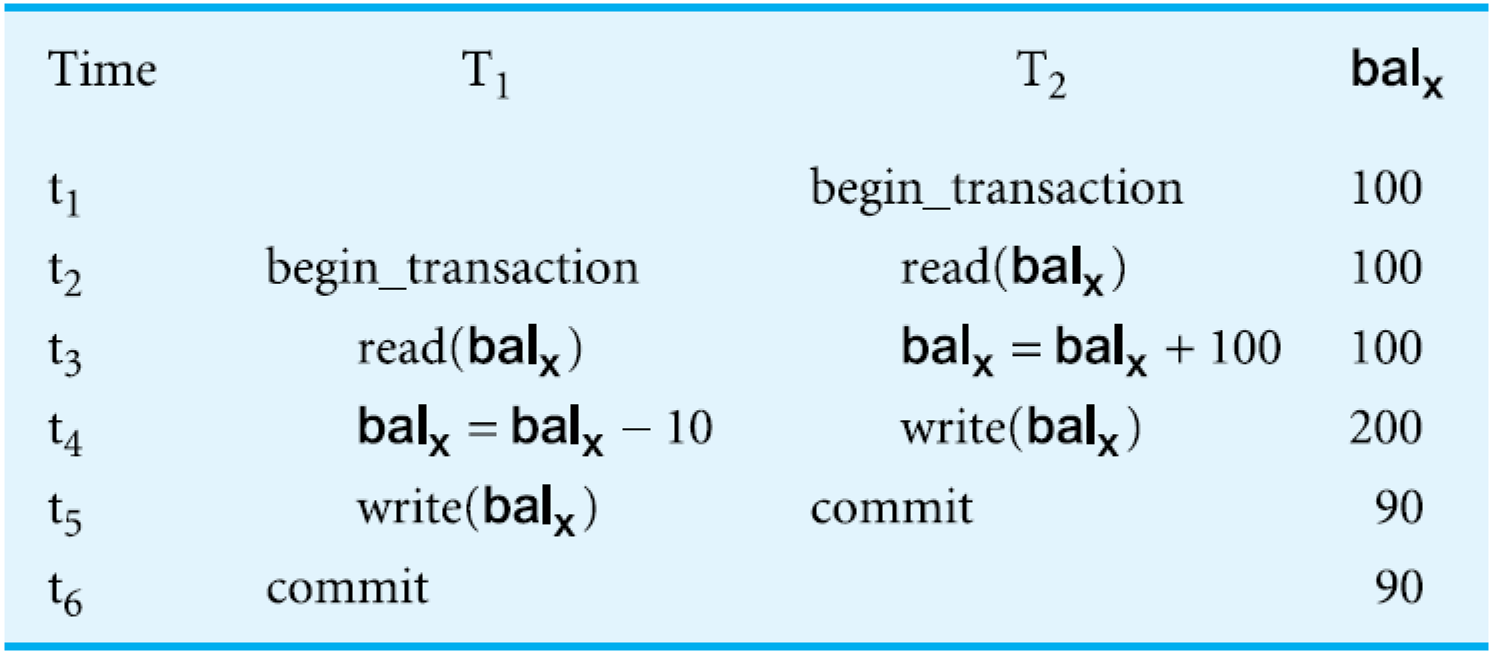
- Transaction sequence must continue until one of four events occur:
  - COMMIT statement is reached
  - ROLLBACK statement is reached
  - End of program is reached
  - Program is abnormally terminated

DBMS Architecture (Part)



Concurrency Control

- The process of managing simultaneous operations on the database without having them interfere with one another.

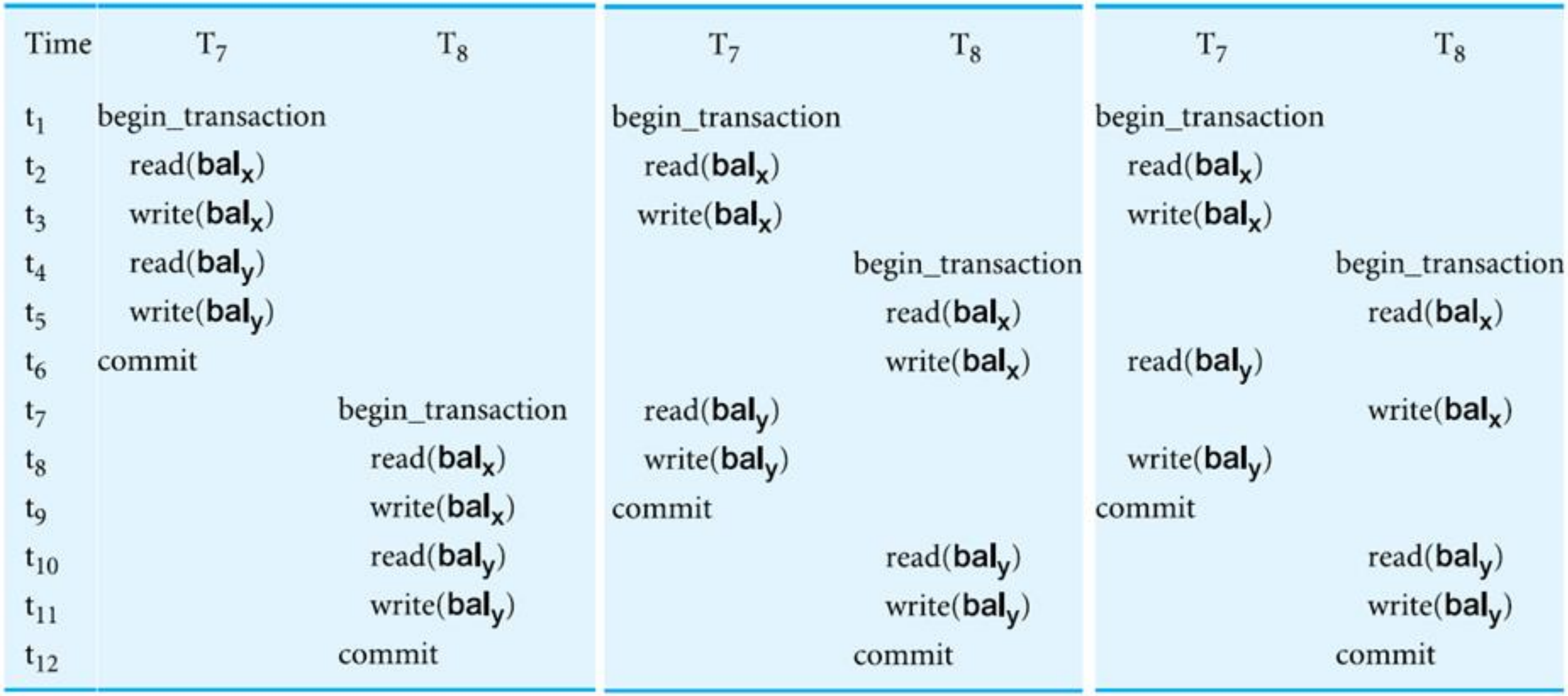


Time	T_1	T_2	bal_x
t_1		begin_transaction	100
t_2	begin_transaction	read(bal_x)	100
t_3	read(bal_x)	$bal_x = bal_x + 100$	100
t_4	$bal_x = bal_x - 10$	write(bal_x)	200
t_5	write(bal_x)	commit	90
t_6	commit		90

Schedule

- Serial schedule
  - A schedule where the operations of each transaction are executed consecutively without any interleaved operations from other transactions.
- Nonserial schedule
  - A schedule where the operations from a set of concurrent transactions are interleaved.
- Serializability
  - The schedule for the concurrent execution of the transactions yields consistent results.

Serializable Schedule



Time	T_7	T_8	T_7	T_8	T_7	T_8
t_1	begin_transaction		begin_transaction		begin_transaction	
t_2	read(bal_x)		read(bal_x)		read(bal_x)	
t_3	write(bal_x)		write(bal_x)		write(bal_x)	
t_4	read(bal_y)			begin_transaction		begin_transaction
t_5	write(bal_y)			read(bal_x)		read(bal_x)
t_6	commit			write(bal_x)	read(bal_y)	
t_7		begin_transaction	read(bal_y)			write(bal_x)
t_8		read(bal_x)	write(bal_y)		write(bal_y)	
t_9		write(bal_x)	commit		commit	
t_{10}		read(bal_y)		read(bal_y)		read(bal_y)
t_{11}		write(bal_y)		write(bal_y)		write(bal_y)
t_{12}		commit		commit		commit

Problems in Concurrency Control

- Lost update
  - Occurs in two concurrent transactions when:
    - Same data element is updated
    - One of the updates is lost
- Uncommitted data
  - Occurs when:
    - Two transactions are executed concurrently
    - First transaction is rolled back after the second transaction has already accessed uncommitted data
- Inconsistent retrievals
  - Occurs when:
    - A transaction accesses data before and after one or more other transactions finish working with such data

The Scheduler

- Establishes the order in which the operations are executed within concurrent transactions
  - Interleaves the execution of database operations to ensure serializability and isolation of transactions
- Bases actions on concurrent control algorithms
  - Determines appropriate order
- Creates serialization schedule
  - Serializable schedule: interleaved execution of transactions yields the same results as the serial execution of the transactions

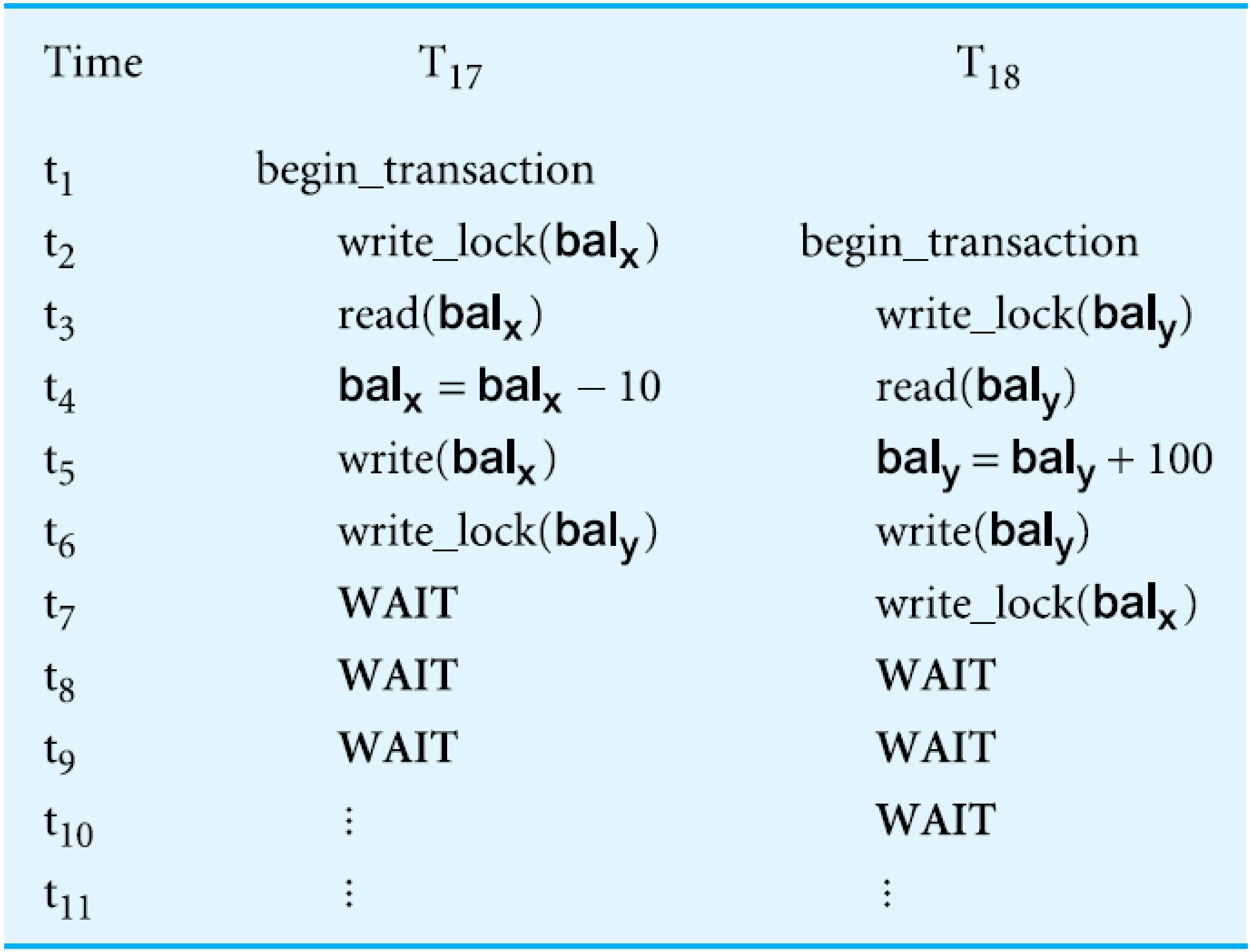
Locking Methods

- Locking methods facilitate isolation of data items used in concurrently executing transactions
  - Lock: guarantees exclusive use of a data item to a current transaction
  - Pessimistic locking: use of locks based on the assumption that conflict between transactions is likely
  - Lock manager: responsible for assigning and policing the locks used by the transactions

Locking Method

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

Deadlock



Time	T_{17}	T_{18}
t_1	begin_transaction	
t_2	write_lock($\mathbf{bal}_x$)	begin_transaction
t_3	read($\mathbf{bal}_x$)	write_lock($\mathbf{bal}_y$)
t_4	$\mathbf{bal}_x = \mathbf{bal}_x - 10$	read($\mathbf{bal}_y$)
t_5	write($\mathbf{bal}_x$)	$\mathbf{bal}_y = \mathbf{bal}_y + 100$
t_6	write_lock($\mathbf{bal}_y$)	write($\mathbf{bal}_y$)
t_7	WAIT	write_lock($\mathbf{bal}_x$)
t_8	WAIT	WAIT
t_9	WAIT	WAIT
t_{10}	⋮	WAIT
t_{11}	⋮	⋮

Lock Granularity

- Database level
- Table level
- Page level
- Row level
- Field level

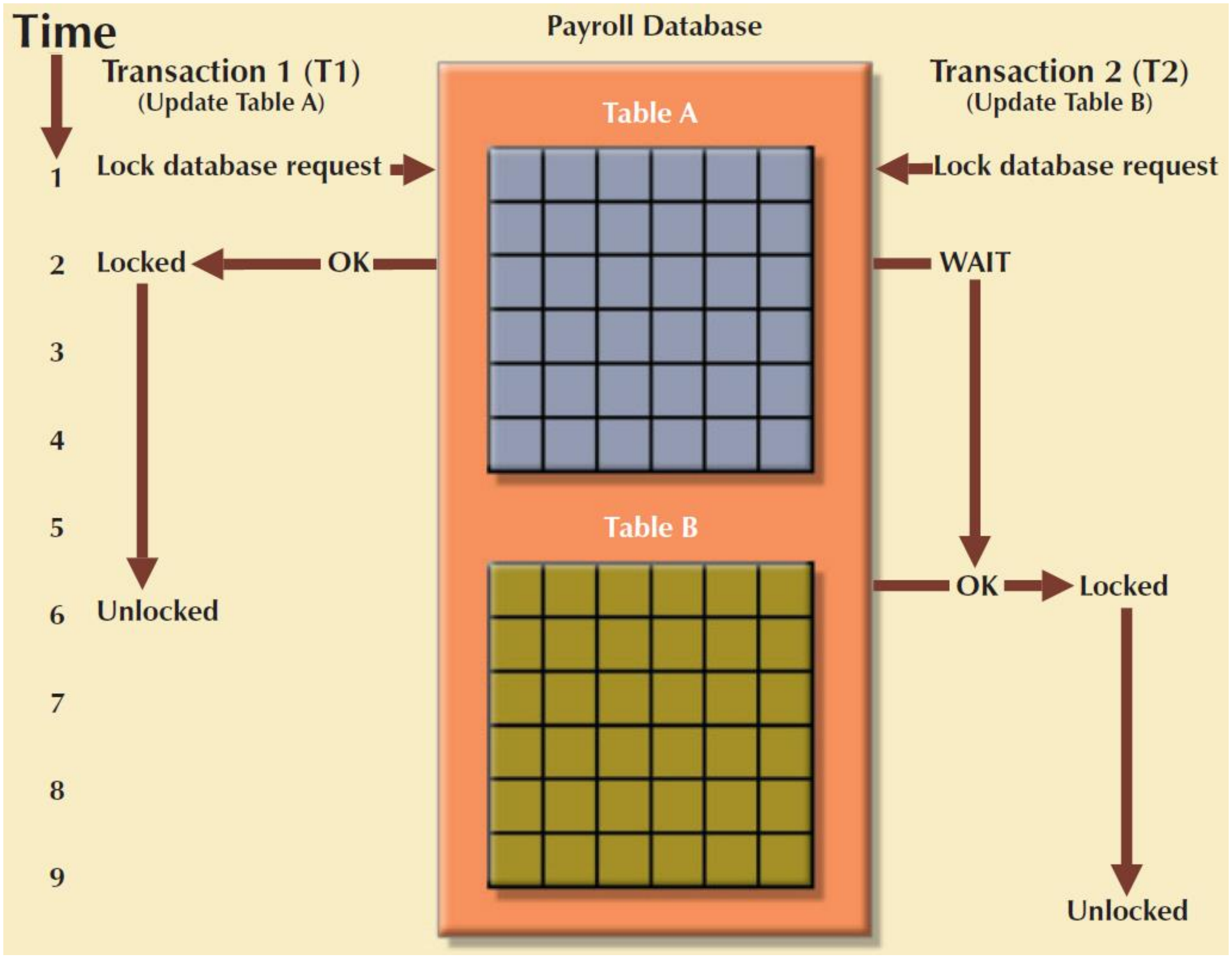


FIGURE 10.4 AN EXAMPLE OF A TABLE-LEVEL LOCK

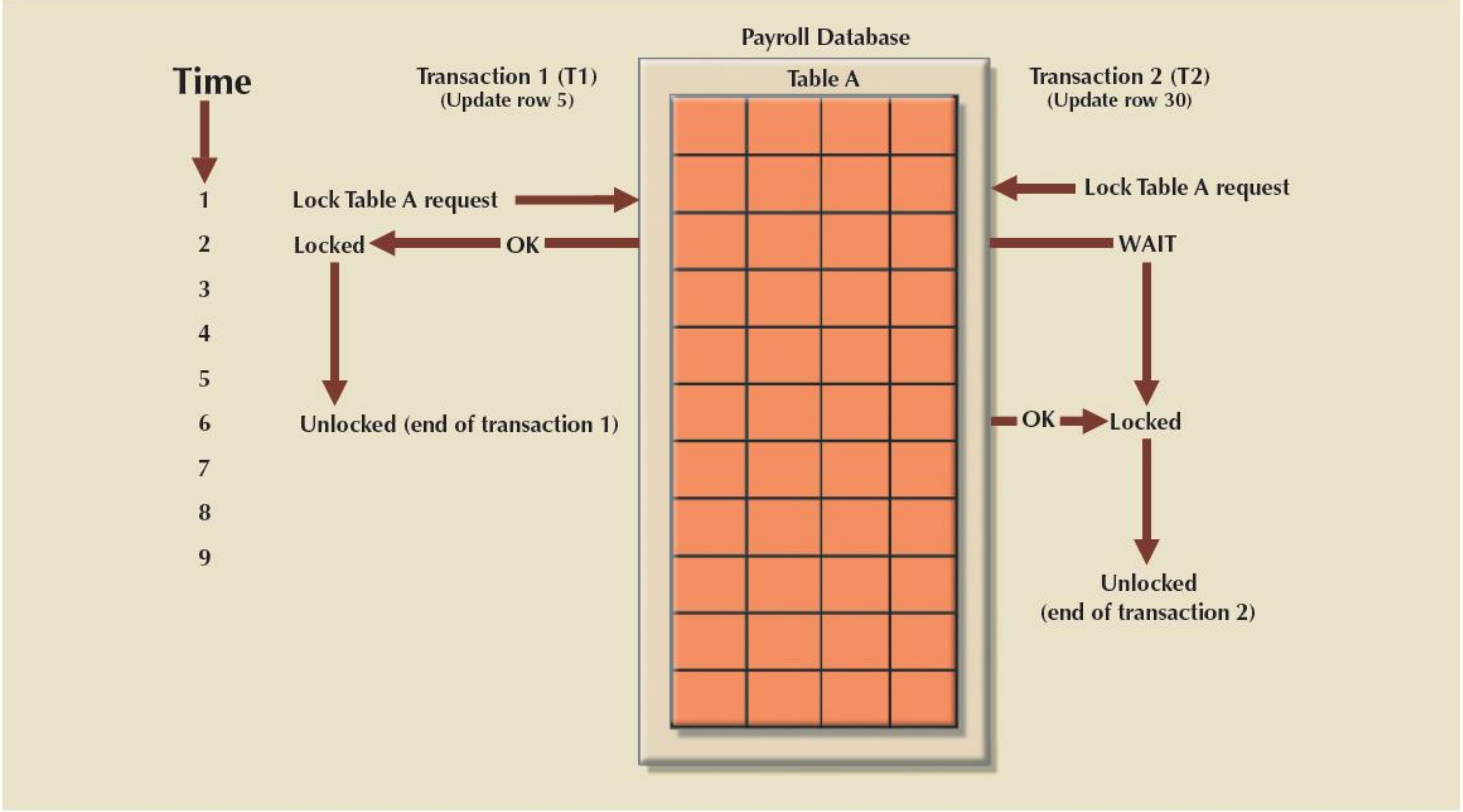


FIGURE 10.5 AN EXAMPLE OF A PAGE-LEVEL LOCK

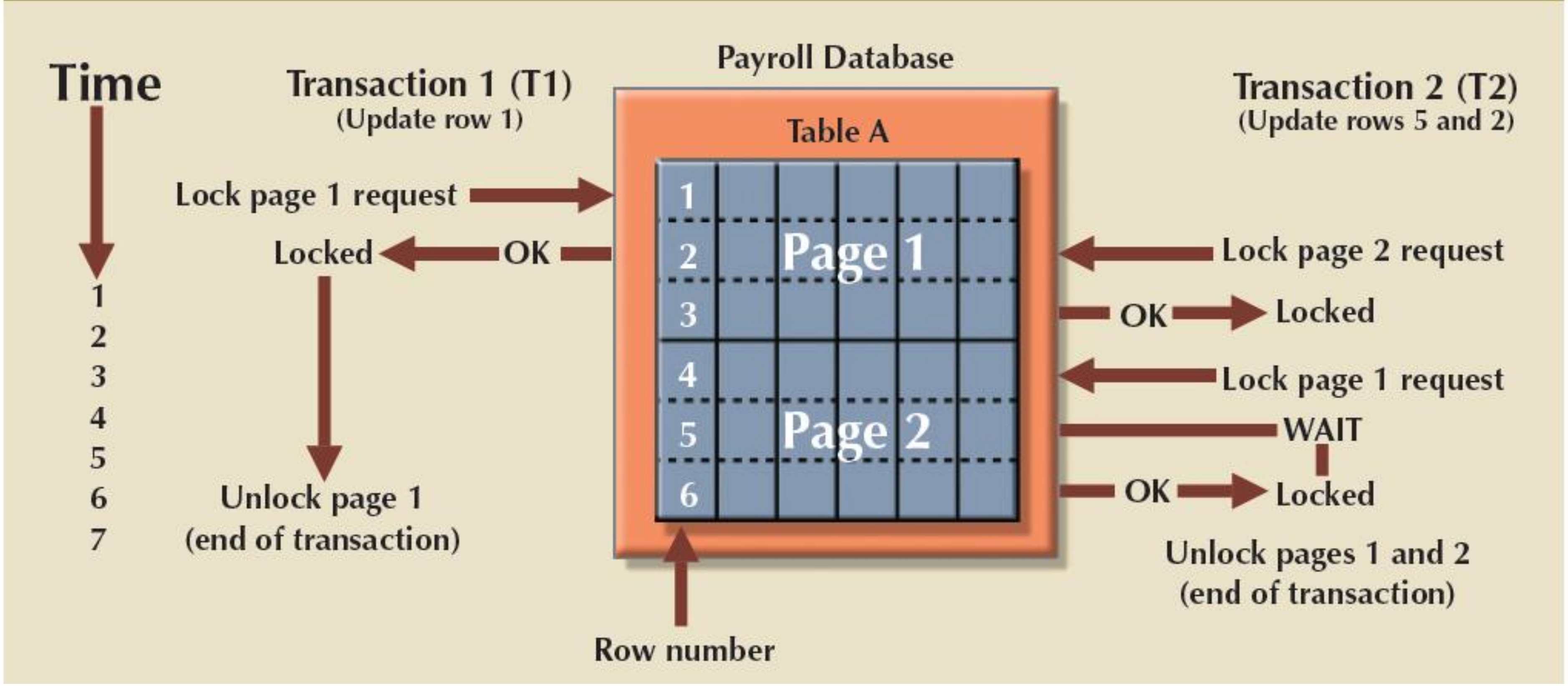
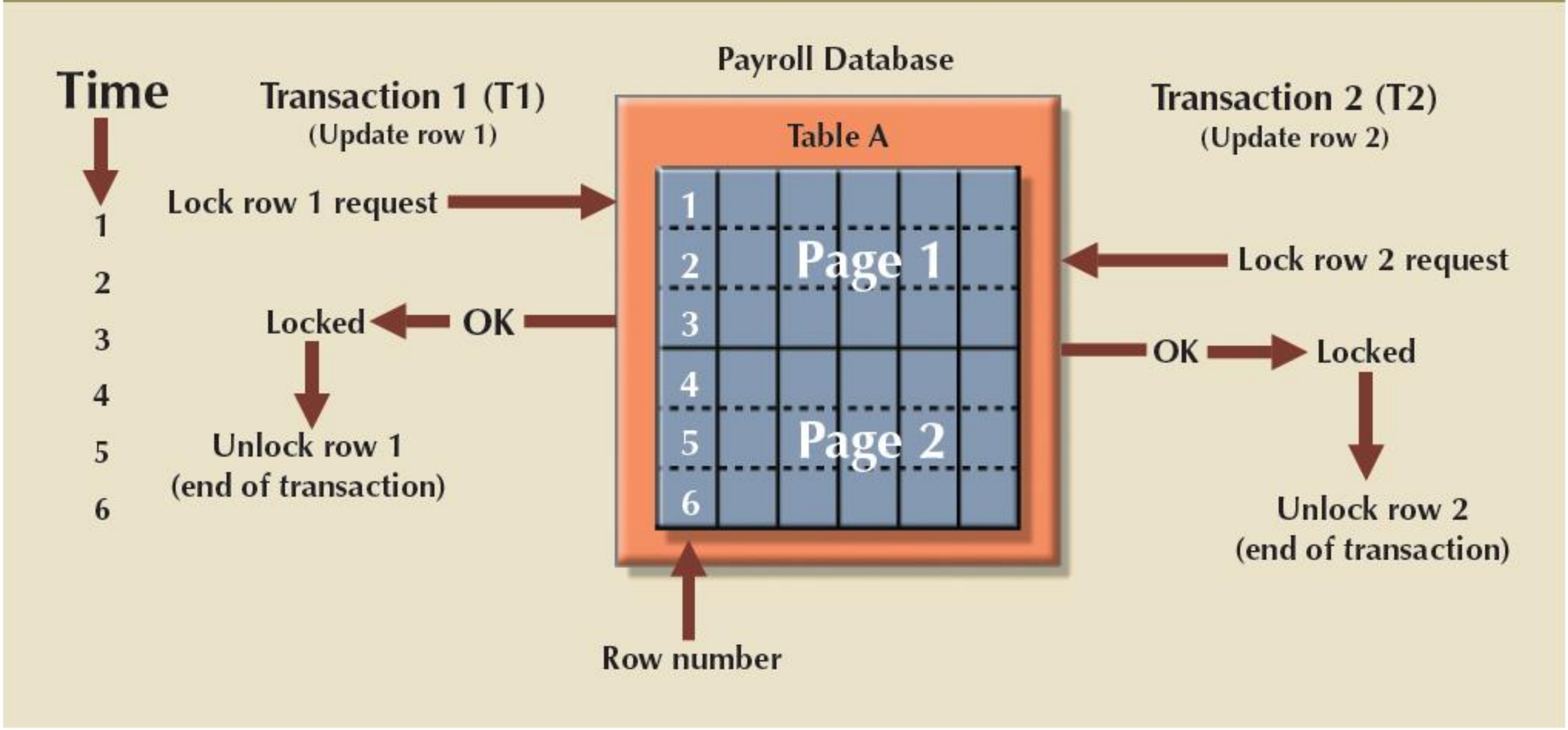


FIGURE 10.6 AN EXAMPLE OF A ROW-LEVEL LOCK



Lock Types

deadlock in PG看老师上课的ppt

- Binary lock
  - Two states: locked (1) and unlocked (0)
    - If an object is locked by a transaction, no other transaction can use that object
    - If an object is unlocked, any transaction can lock the object for its use
- Exclusive lock
  - Access is reserved for the transaction that locked the object
- Shared lock
  - Concurrent transactions are granted read access on the basis of a common lock
- Problems using locks
  - Resulting transaction schedule might not be serializable
  - Schedule might create deadlocks

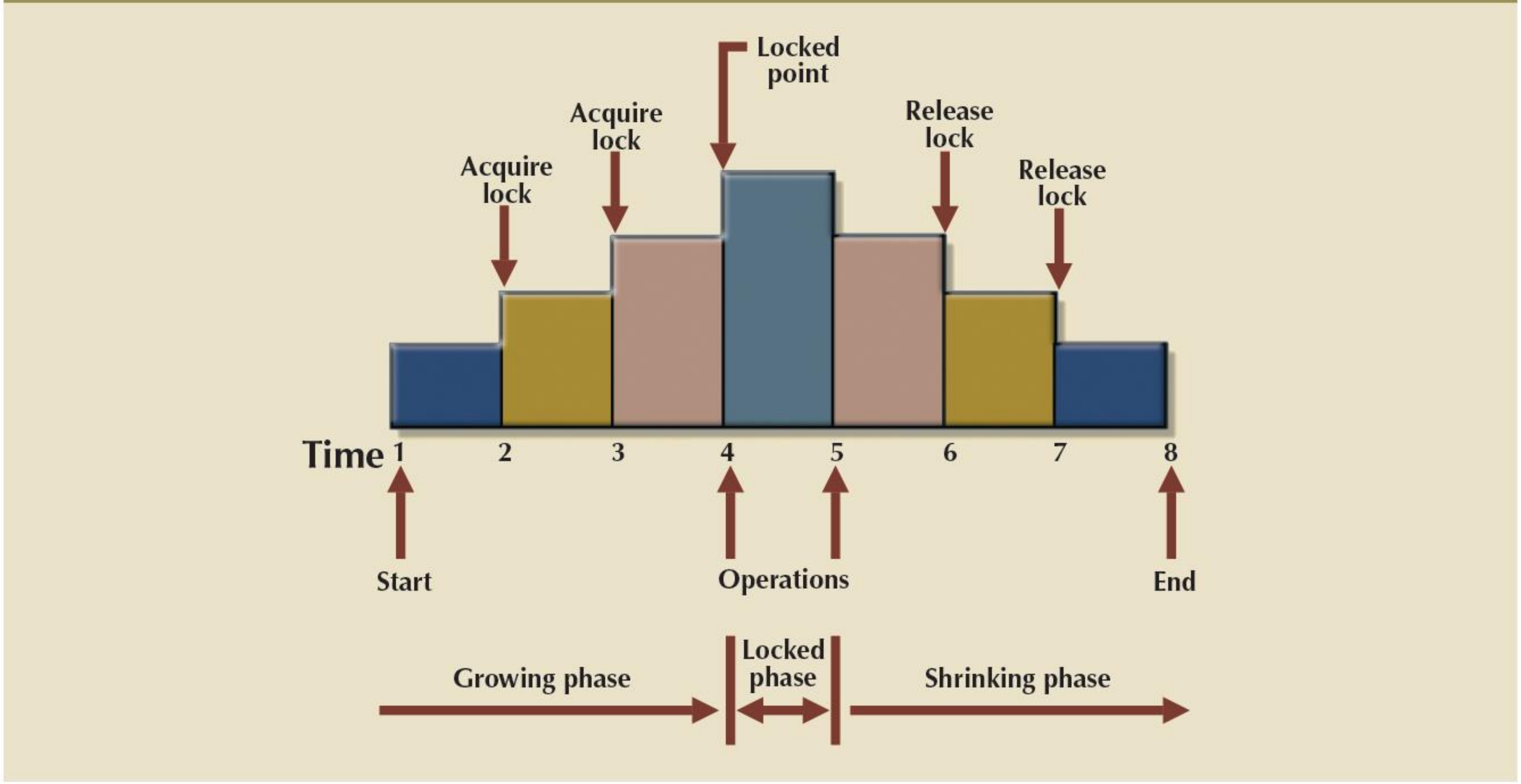
Two-Phase Locking to Ensure Serializability

- Defines how transactions acquire and relinquish locks
  - Guarantees serializability but does not prevent deadlocks
- Phases
  - Growing phase: transaction acquires all required locks without unlocking any data
  - Shrinking phase: transaction releases all locks and cannot obtain any new lock
- Governing rules

两个事务不能有冲突的锁

  - Two transactions cannot have conflicting locks
  - No unlock operation can precede a lock operation in the same transaction
  - No data are affected until all locks are obtained

FIGURE 10.7 TWO-PHASE LOCKING PROTOCOL



Deadlocks 死锁的具体case看雨课堂ppt

- Occur when two transactions wait indefinitely for each other to unlock data
  – Also known as deadly embrace
- Control techniques
  – Deadlock prevention 预防
  – Deadlock detection 检测
  – Deadlock avoidance 避免
- Choice of deadlock control method depends on database environment

Concurrency Control with Time Stamping Methods

时间戳方法

全局统一，唯一的

- Time stamping assigns global, unique time stamp to each transaction
  - Produces explicit order in which transactions are submitted to DBMS
- Properties
  - Uniqueness: ensures no equal time stamp values exist 唯一性
  - Monotonicity: ensures time stamp values always increases 一定是单调增的

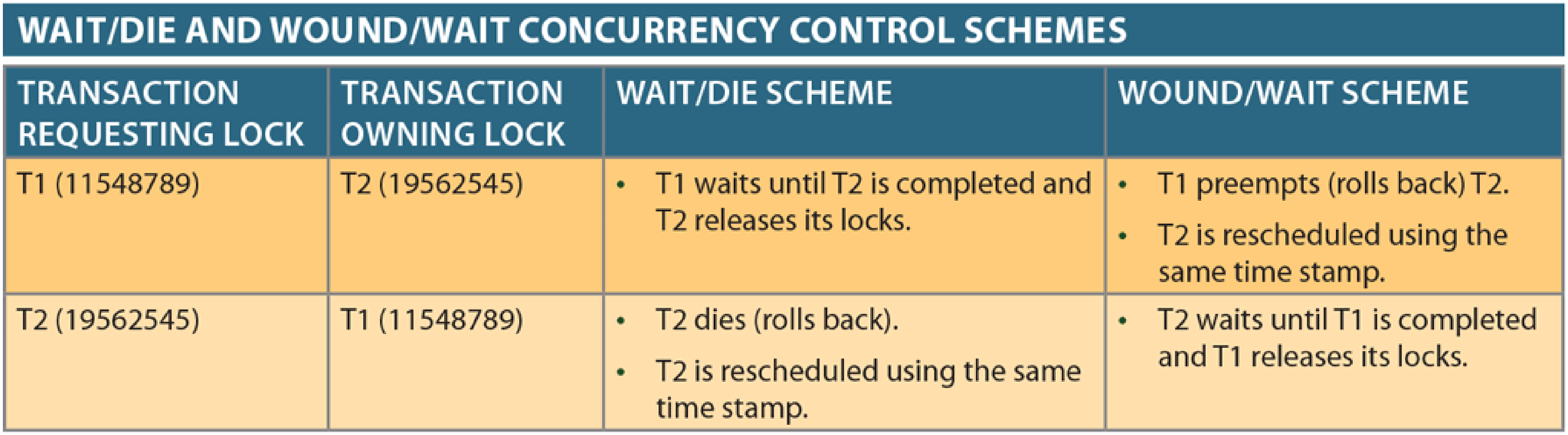
Concurrency Control with Time Stamping Methods

- Disadvantages
  - Each value stored in the database requires two additional stamp fields
  - Increases memory needs
  - Increases the database’s processing overhead
  - Demands a lot of system resources

Wait/Die and Wound/Wait Schemes

- Wait/die 时间戳 数字小的老，数字大的年轻
  - A concurrency control scheme in which an older transaction must wait for the younger transaction to complete and release the locks before requesting the locks itself
    - Otherwise, the newer transaction dies and is rescheduled
- Wound/wait
  - A concurrency control scheme in which an older transaction can request the lock, preempt the younger transaction, and reschedule it 相当于资源冲突时，新事务会rollback
    - Otherwise, the newer transaction waits until the older transaction finishes

Wait/Die and Wound/Wait Schemes



WAIT/DIE AND WOUND/WAIT CONCURRENCY CONTROL SCHEMES

TRANSACTION REQUESTING LOCK	TRANSACTION OWNING LOCK	WAIT/DIE SCHEME	WOUND/WAIT SCHEME
T1 (11548789)	T2 (19562545)	• T1 waits until T2 is completed and T2 releases its locks.	• T1 preempts (rolls back) T2. • T2 is rescheduled using the same time stamp.
T2 (19562545)	T1 (11548789)	• T2 dies (rolls back). • T2 is rescheduled using the same time stamp.	• T2 waits until T1 is completed and T1 releases its locks.

Concurrency Control with Optimistic Methods

- Optimistic approach: Based on the assumption that the majority of database operations do not conflict
  - Does not require locking or time stamping techniques
  - Transaction is executed without restrictions until it is committed
- Phases of optimistic approach
  - Read
  - Validation
  - Write

Concurrency Control with Optimistic Methods

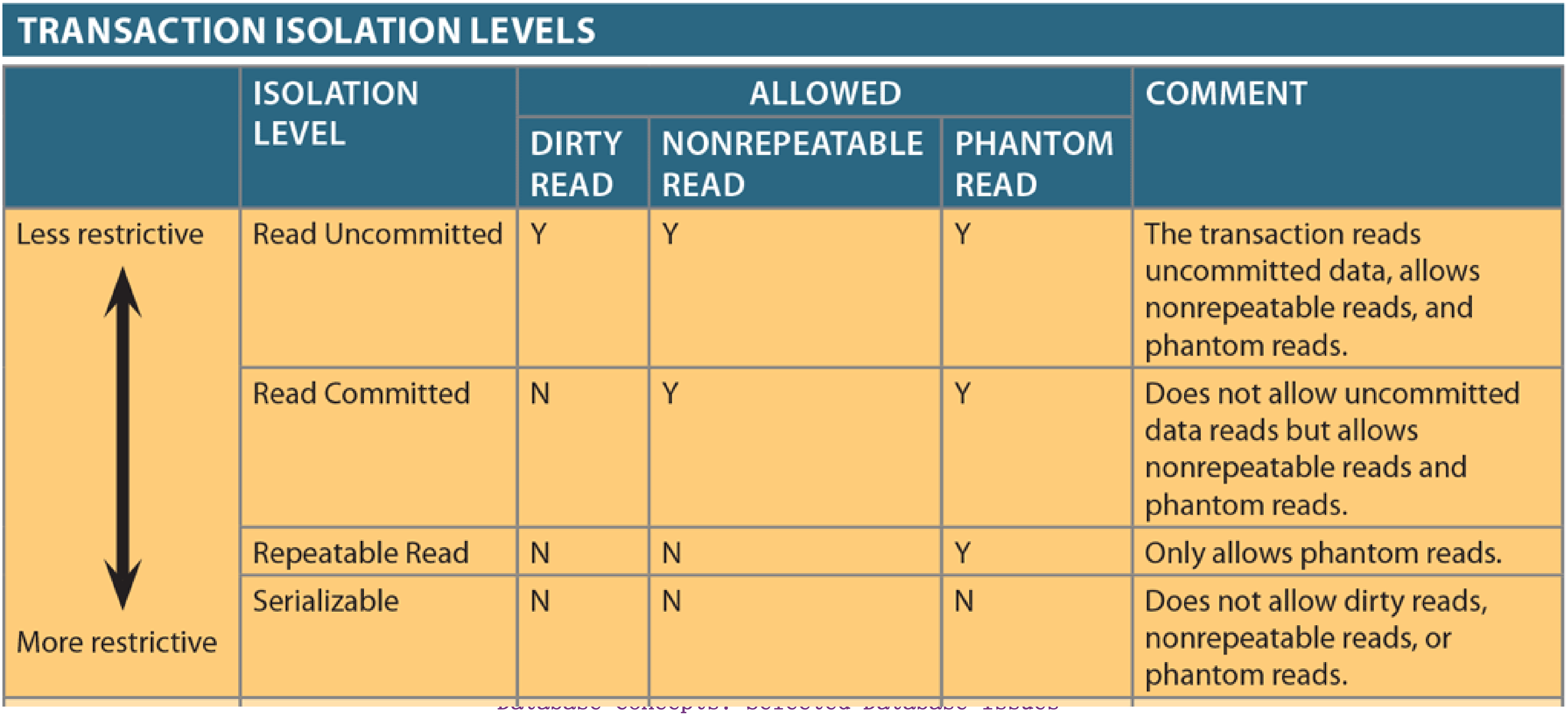
- Read phase
  - Transaction:
    - Reads the database
    - Executes the needed computations
    - Makes the updates to a private copy of the database values
- Validation phase
  - Transaction is validated to ensure that the changes made will not affect the integrity and consistency of the database
- Write phase
  - Changes are permanently applied to the database

ANSI Levels of Transaction Isolation

- The ANSI SQL standard (1992) defines transaction management based on transaction isolation levels
  - Transaction isolation levels refer to the degree to which transaction data is “protected or isolated” from other concurrent transactions
- Transaction isolation levels are described by the type of “reads” that a transaction allows or not
  - Dirty read: transaction can read data that is not yet committed
  - Nonrepeatable read: transaction reads a given row at time t1, and then it reads the same row at time t2, yielding different results
    - The original row may have been updated or deleted

ANSI Levels of Transaction Isolation

– Phantom read: transaction executes a query at time t1, and then it runs the same query at time t2, yielding additional rows that satisfy the query



TRANSACTION ISOLATION LEVELS

	ISOLATION LEVEL	ALLOWED			COMMENT
		DIRTY READ	NONREPEATABLE READ	PHANTOM READ	
Less restrictive	Read Uncommitted	Y	Y	Y	The transaction reads uncommitted data, allows nonrepeatable reads, and phantom reads.
	Read Committed	N	Y	Y	Does not allow uncommitted data reads but allows nonrepeatable reads and phantom reads.
	Repeatable Read	N	N	Y	Only allows phantom reads.
More restrictive	Serializable	N	N	N	Does not allow dirty reads, nonrepeatable reads, or phantom reads.

ANSI Levels of Transaction Isolation

- Read Uncommitted will read uncommitted data from other transactions
  - Increases transaction performance but at the cost of data consistency
- Read Committed forces transactions to read only committed data
  - Default mode of operation for most databases
- Repeatable Read isolation level ensures that queries return consistent results
  - Uses shared locks to ensure other transactions do not update a row after the original query reads it
- Serializable isolation level is the most restrictive level defined by the ANSI SQL standard
  - Deadlocks are still always possible

Database Recovery Management

- Database recovery: restores database from a given state to a previously consistent state
- Recovery transactions are based on the atomic transaction property
  - All portions of a transaction must be treated as a single logical unit of work
    - If transaction operation cannot be completed:
      - Transaction must be aborted
      - Changes to database must be rolled back

Database Recovery Management

- Concepts that affect the recovery process
  - Write-ahead log protocol
    - Ensures that transaction logs are always written before the data are updated
  - Redundant transaction logs
    - Ensure that a physical disk failure will not impair the DBMS’s ability to recover data
  - Buffers
    - Temporary storage areas in a primary memory used to speed up disk operations
  - Checkpoints 相当于加入一些checkpoint去检查，可以看到【之前】【之后】的操作
    - Allows DBMS to write all its updated buffers in memory to disk

Database Recovery Management

- Techniques used in transaction recovery procedures
  - Deferred-write technique or <u>deferred update</u>
    - Transaction operations do not immediately update the physical database
    - Only transaction log is updated
  - Write-through technique or <u>immediate update</u>
    - Database is immediately updated by transaction operations during transaction's execution

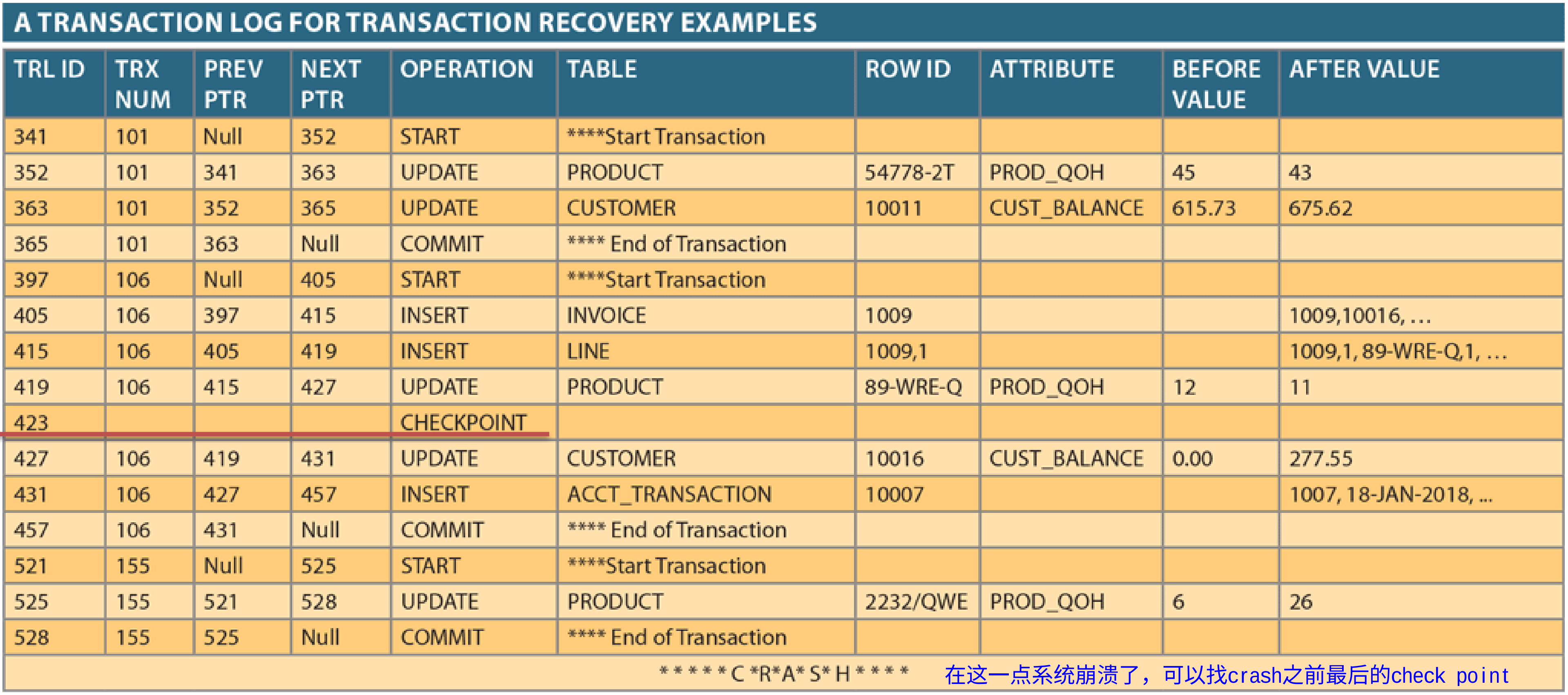
Database Recovery Management

- Recovery process steps
  - Identify <u>the last check point</u> in the transaction log
    - If transaction was committed before the last check point nothing needs to be done
    - If transaction was committed after the last check point the transaction log is used to redo the transaction
    - If transaction had a ROLLBACK operation after the last check point the DBMS uses the transaction log records to ROLLBACK or undo the operations, using the “before” values in the transaction log

Database Recovery Management

更新前的数字

更新后的数字



A TRANSACTION LOG FOR TRANSACTION RECOVERY EXAMPLES

TRL ID	TRX NUM	PREV PTR	NEXT PTR	OPERATION	TABLE	ROW ID	ATTRIBUTE	BEFORE VALUE	AFTER VALUE
341	101	Null	352	START	****Start Transaction				
352	101	341	363	UPDATE	PRODUCT	54778-2T	PROD_QOH	45	43
363	101	352	365	UPDATE	CUSTOMER	10011	CUST_BALANCE	615.73	675.62
365	101	363	Null	COMMIT	**** End of Transaction				
397	106	Null	405	START	****Start Transaction				
405	106	397	415	INSERT	INVOICE	1009			1009,10016, . . .
415	106	405	419	INSERT	LINE	1009,1			1009,1, 89-WRE-Q,1, . . .
419	106	415	427	UPDATE	PRODUCT	89-WRE-Q	PROD_QOH	12	11
423				CHECKPOINT					
427	106	419	431	UPDATE	CUSTOMER	10016	CUST_BALANCE	0.00	277.55
431	106	427	457	INSERT	ACCT_TRANSACTION	10007			1007, 18-JAN-2018, ...
457	106	431	Null	COMMIT	**** End of Transaction				
521	155	Null	525	START	****Start Transaction				
525	155	521	528	UPDATE	PRODUCT	2232/QWE	PROD_QOH	6	26
528	155	525	Null	COMMIT	**** End of Transaction				
					* * * * * C *R*A* S* H * * * *				

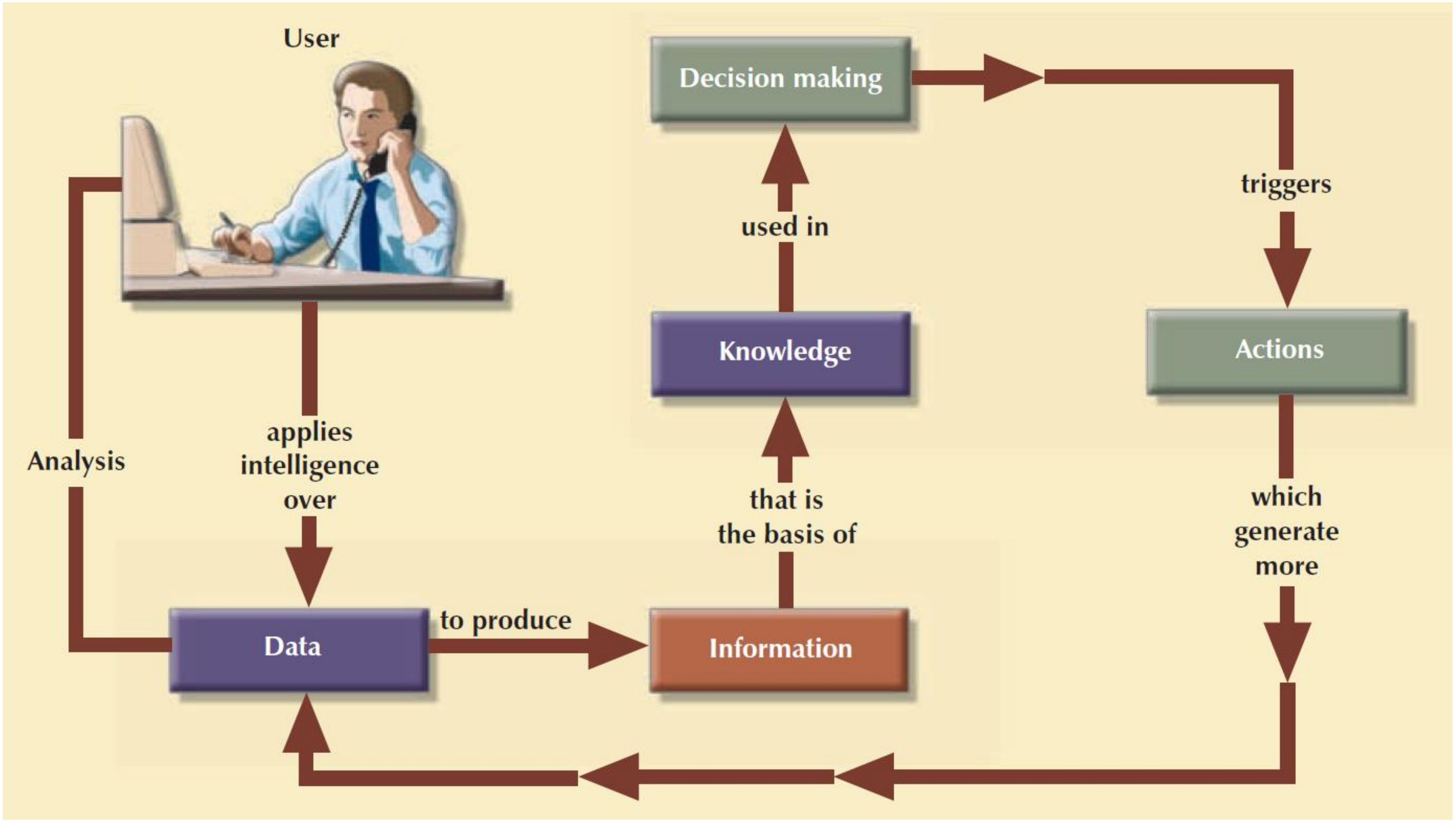
在这一点系统崩溃了，可以找crash之前最后的check point

相当于说问题是出在最后一个checkpoint和crash之间

Outline

- Transaction Management*
- ✈ Database Security
- Database Administration

Data as a Corporate Asset



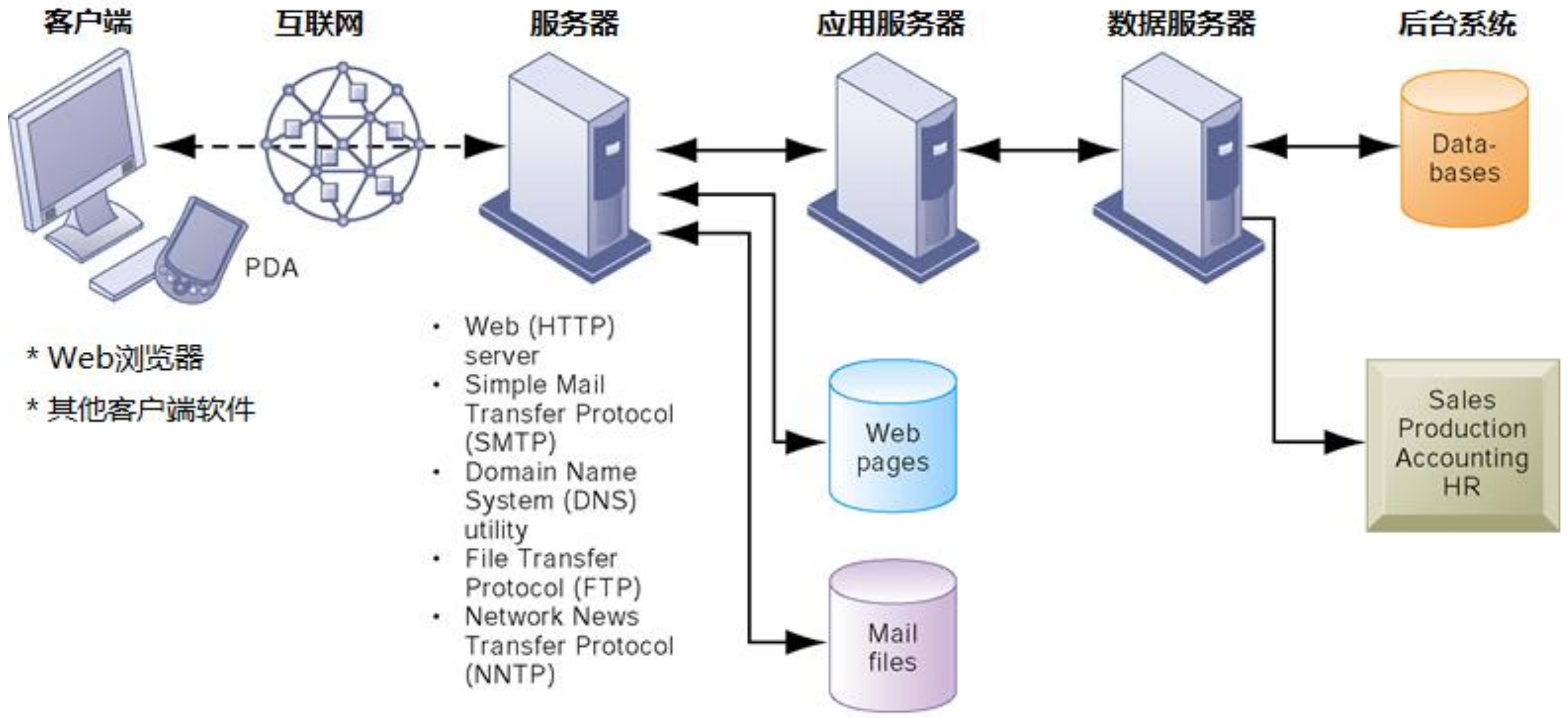
Database Security

- Mechanisms that protect the database against intentional or accidental threats.
  - Data is a valuable resource that must be strictly controlled and managed.
  - Data may have strategic importance.
- Security considerations
  - Data held in a database
  - Other parts of the system
    - which may in turn affect the database

Database Security

- Involves measures to avoid:
  - Theft and fraud
  - Loss of confidentiality (secrecy)
  - Loss of privacy
  - Loss of integrity
  - Loss of availability

Typical Multi-user Computer Environment

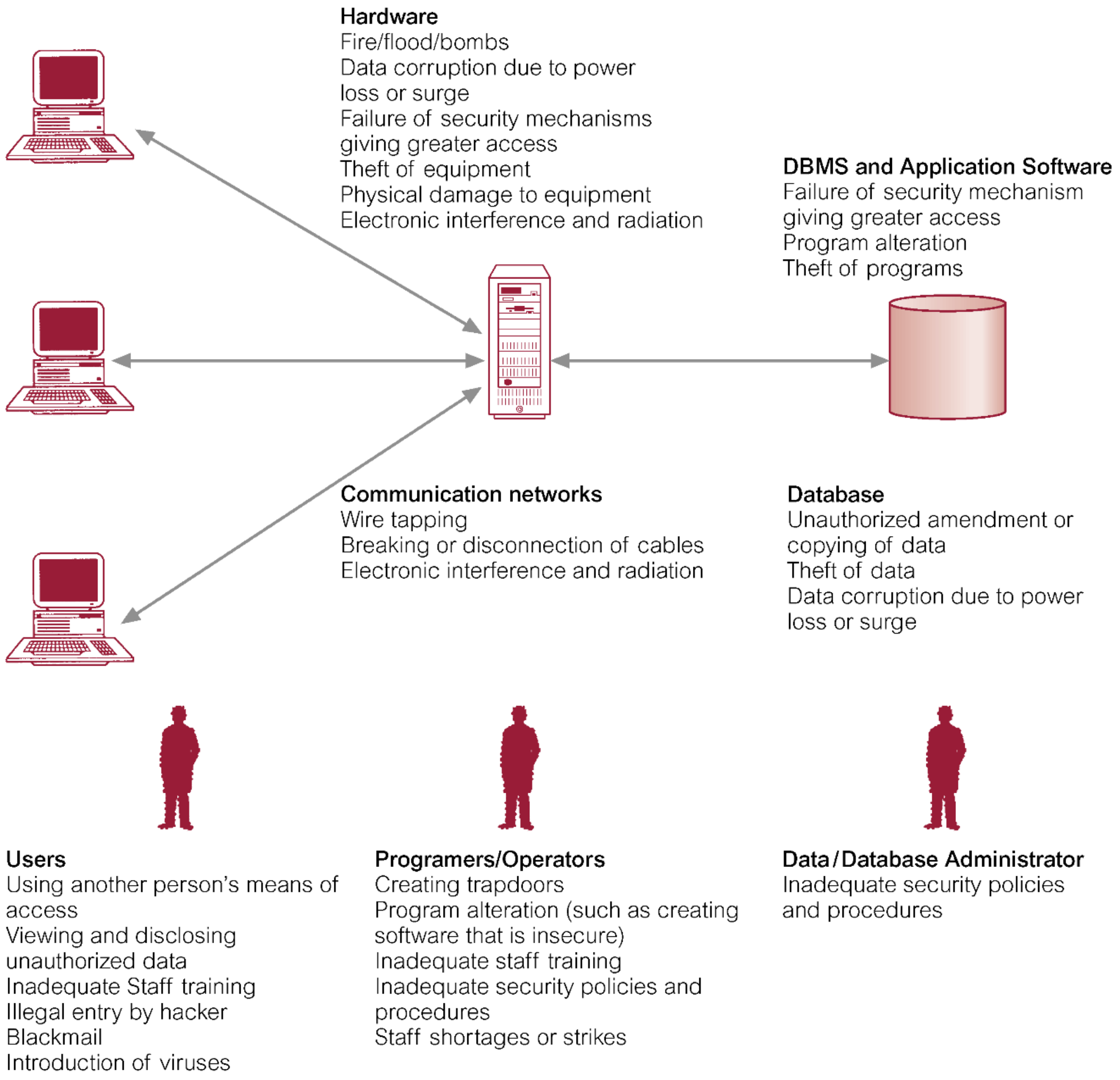


* Web浏览器

* 其他客户端软件

- Web (HTTP) server
- Simple Mail Transfer Protocol (SMTP)
- Domain Name System (DNS) utility
- File Transfer Protocol (FTP)
- Network News Transfer Protocol (NNTP)

Summary of Threats to Computer Systems



Hardware
Fire/flood/bombs
Data corruption due to power loss or surge
Failure of security mechanisms giving greater access
Theft of equipment
Physical damage to equipment
Electronic interference and radiation

DBMS and Application Software
Failure of security mechanism giving greater access
Program alteration
Theft of programs

Communication networks
Wire tapping
Breaking or disconnection of cables
Electronic interference and radiation

Database
Unauthorized amendment or copying of data
Theft of data
Data corruption due to power loss or surge

Users
Using another person's means of access
Viewing and disclosing unauthorized data
Inadequate Staff training
Illegal entry by hacker
Blackmail
Introduction of viruses

Programers/Operators
Creating trapdoors
Program alteration (such as creating software that is insecure)
Inadequate staff training
Inadequate security policies and procedures
Staff shortages or strikes

Data/Database Administrator
Inadequate security policies and procedures

Countermeasures – Computer-Based Controls

- Concerned with physical controls to administrative procedures
  - Authorization
  - Access controls
  - Views
  - Backup and recovery
  - Integrity
  - Encryption
  - RAID technology

Countermeasures – Computer-Based Controls

- Access control
  - Based on the granting and revoking of privileges.
  - A privilege allows a user to create or access (that is read, write, or modify) some database object (such as a relation, view, and index) or to run certain DBMS utilities.
  - Privileges are granted to users to accomplish the tasks required for their jobs.

CREATE USER alice PASSWORD 'a123';

因为这个表格是postgre创建的，所以alice一开始是没有权限的
GRANT SELECT ON table_name TO alice
还可以GRANT(id) ON table_naem就是给某个字段的权限

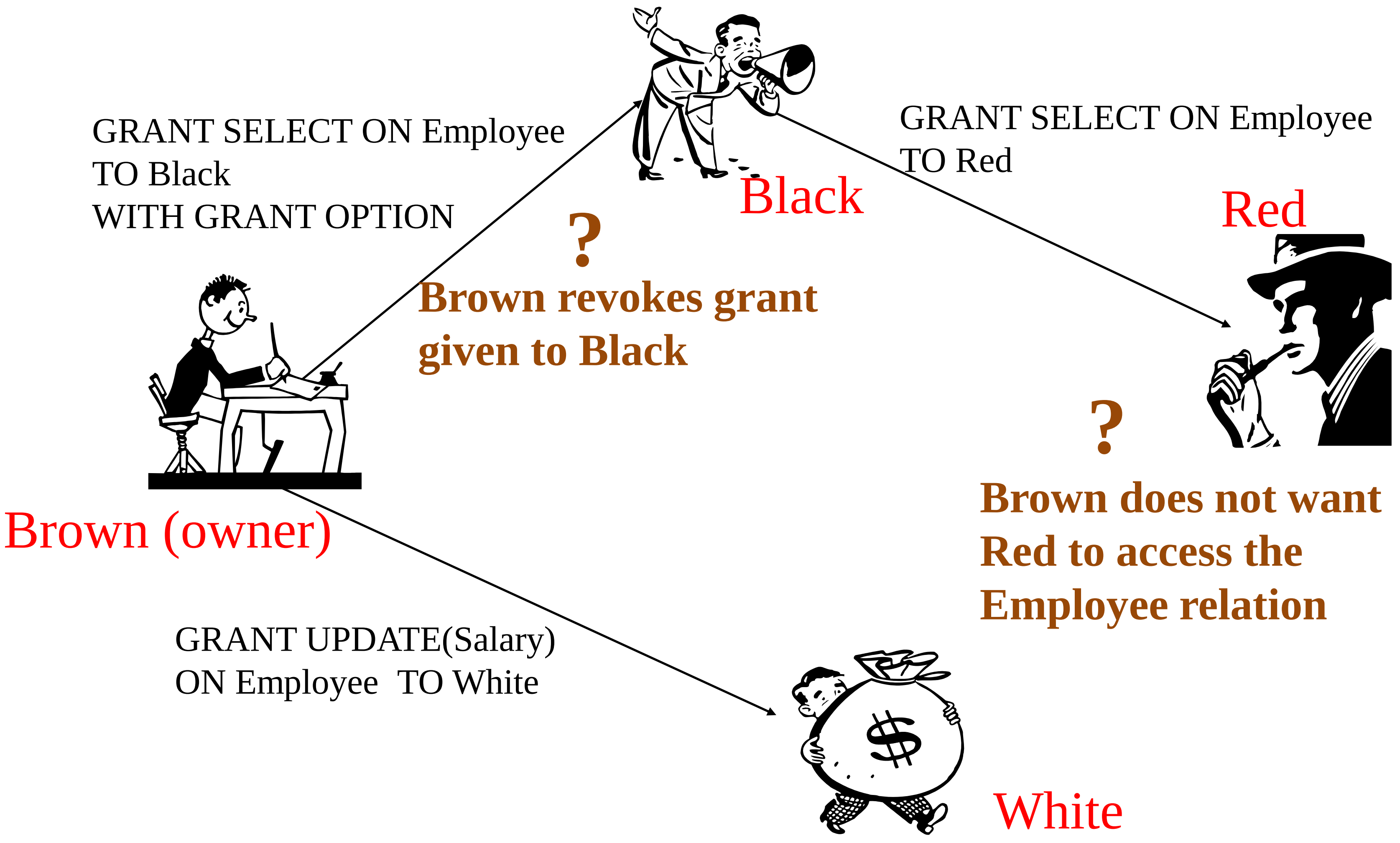
如果你在GRANT TO后面加一个WITH GRANT OPTION;
相当于允许alice把select权限赋予他人

REVOKE SELECT ON table_name FROM alice CASCADE;
收回赋予给alice的权限，CASCADE是限制，相当于把“alice授予其他人的权限”也收回了（不然会被报错）

Countermeasures – Computer-Based Controls

- Most DBMS provide an approach called Discretionary (自主) Access Control (DAC).
- SQL standard supports DAC through the GRANT and REVOKE commands.
- The GRANT command gives privileges to users, and the REVOKE command takes away privileges.

Countermeasures – Computer-Based Controls



```
GRANT SELECT ON Employee
TO Black
WITH GRANT OPTION
```

```
GRANT SELECT ON Employee
TO Red
```

```
GRANT UPDATE(Salary)
ON Employee  TO White
```

Brown revokes grant given to Black

Brown does not want Red to access the Employee relation

Countermeasures – Computer-Based Controls

- DAC while effective has certain weaknesses.
  - An unauthorized user can trick an authorized user into disclosing sensitive data.

- Mandatory (强制) Access Control (MAC)
  - Based on system-wide policies that cannot be changed by individual users

Countermeasures – Computer-Based Controls

- View
  - Is the dynamic result of one or more relational operations operating on the base relations to produce another relation.
  - A view is a virtual relation that does not actually exist in the database, but is produced upon request by a particular user, at the time of request.

Countermeasures – Computer-Based Controls

- Backup
  - Process of periodically taking a copy of the database and log file (and possibly programs) to offline storage media.

- Journaling
  - Process of keeping and maintaining a log file (or journal) of all changes made to database to enable effective recovery in event of failure.

Countermeasures – Computer-Based Controls

- Integrity
  - Prevents data from becoming invalid, and hence giving misleading or incorrect results.
- Encryption
  - The encoding of the data by a special algorithm that renders the data unreadable by any program without the decryption key.

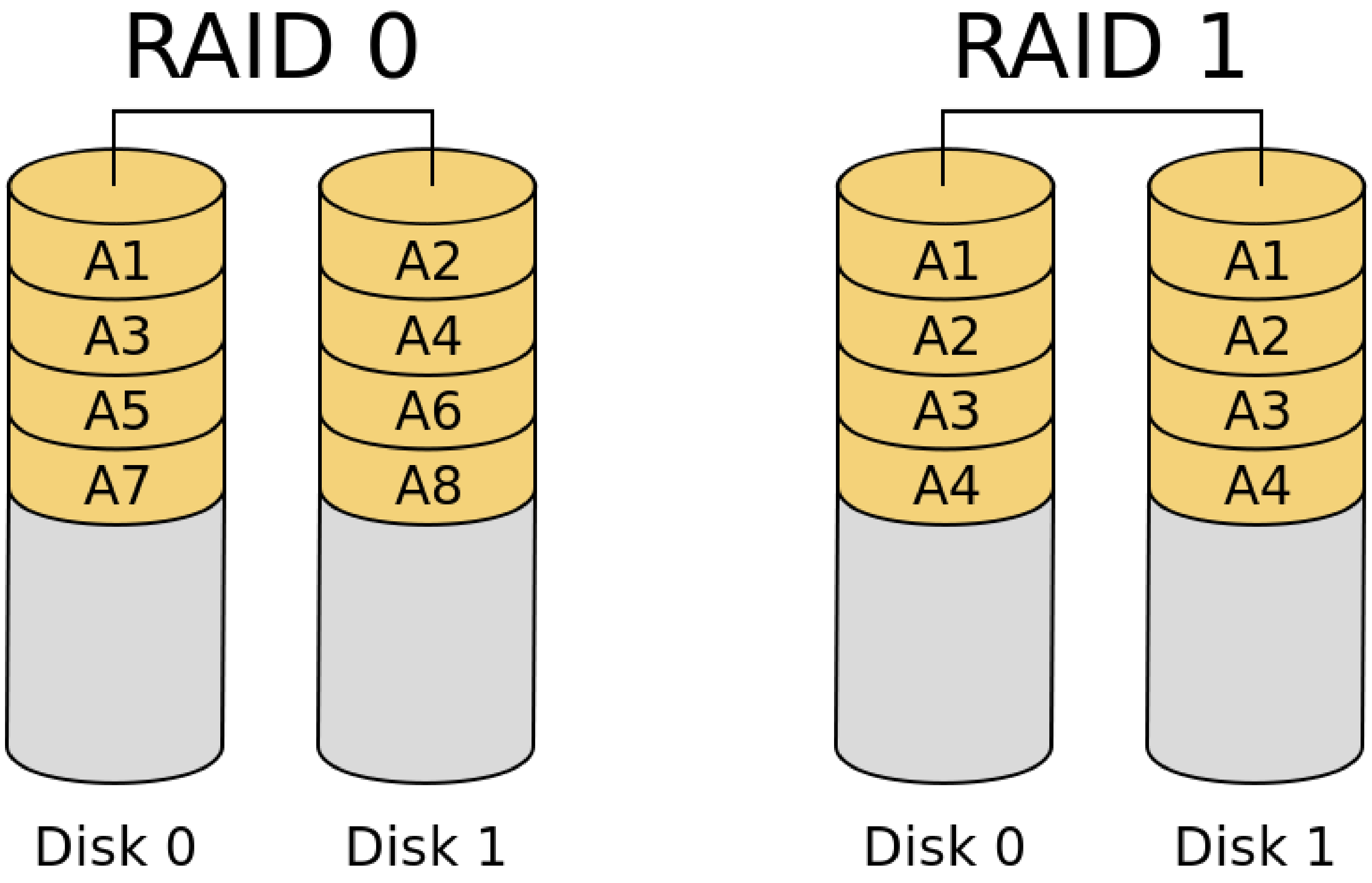
Countermeasures – RAID Technology

- Hardware that the DBMS is running on must be fault-tolerant, meaning that the DBMS should continue to operate even if one of the hardware components fails.
- Disk drives are the most vulnerable components with the shortest times between failure of any of the hardware components.

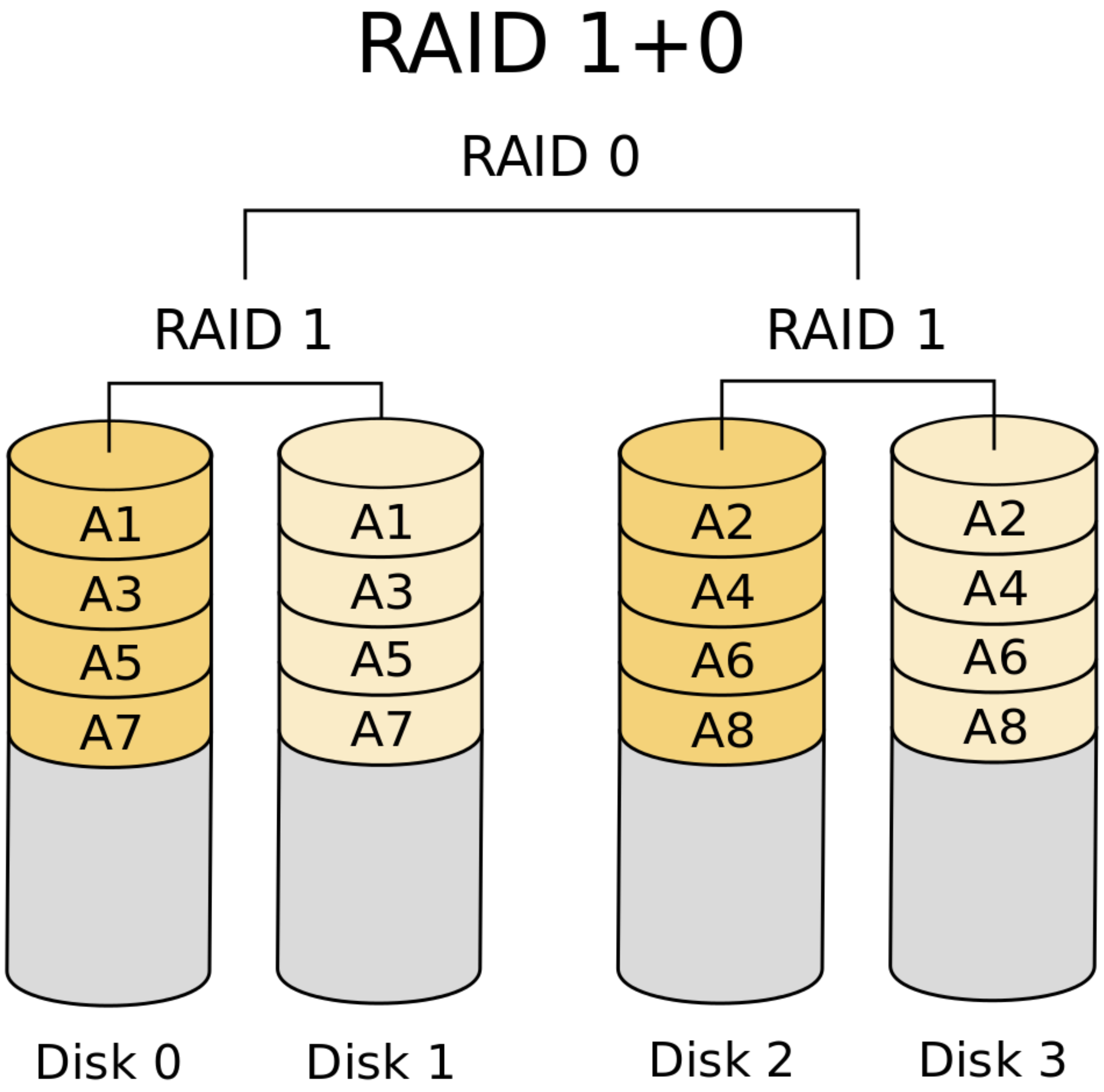
Countermeasures – RAID Technology

- RAID (独立磁盘冗余阵列, Redundant Array of Independent Disks)
- Provide a large disk array comprising an arrangement of several independent disks
  - Increase performance
    - data striping: the data is segmented into equal-size partitions (the striping unit), which are transparently distributed across multiple disks.
  - Improve reliability
    - Storing redundant information across the disks using a parity scheme or an error-correcting scheme.

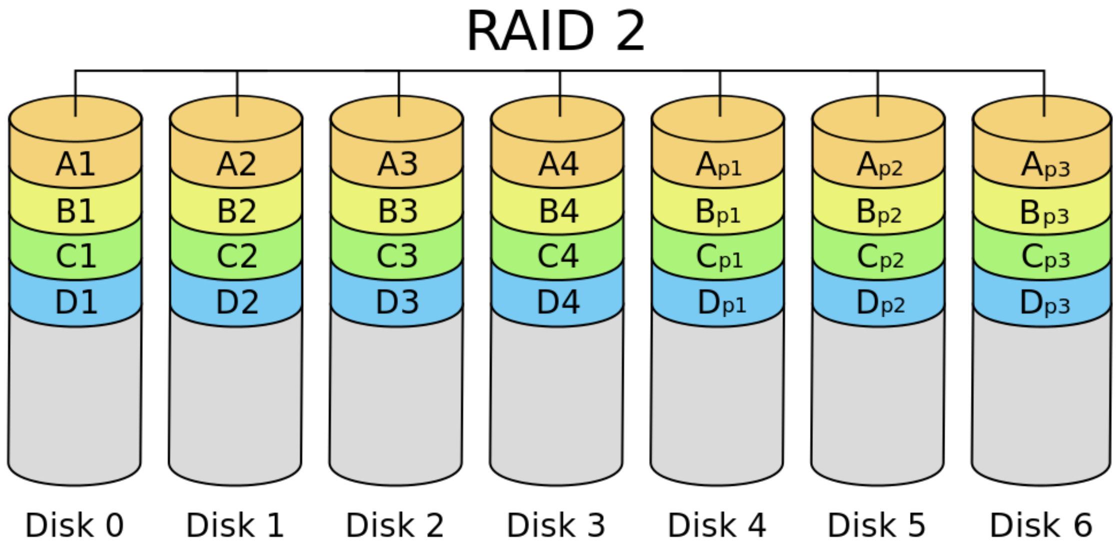
Countermeasures – RAID Technology

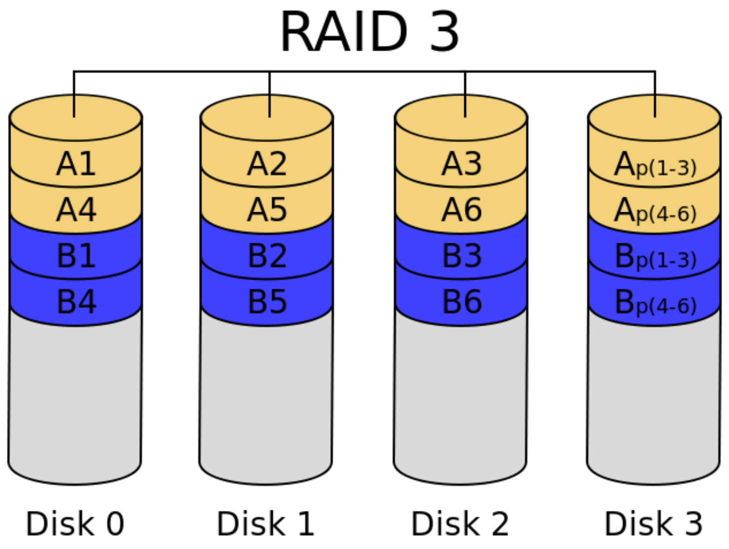


Countermeasures – RAID Technology

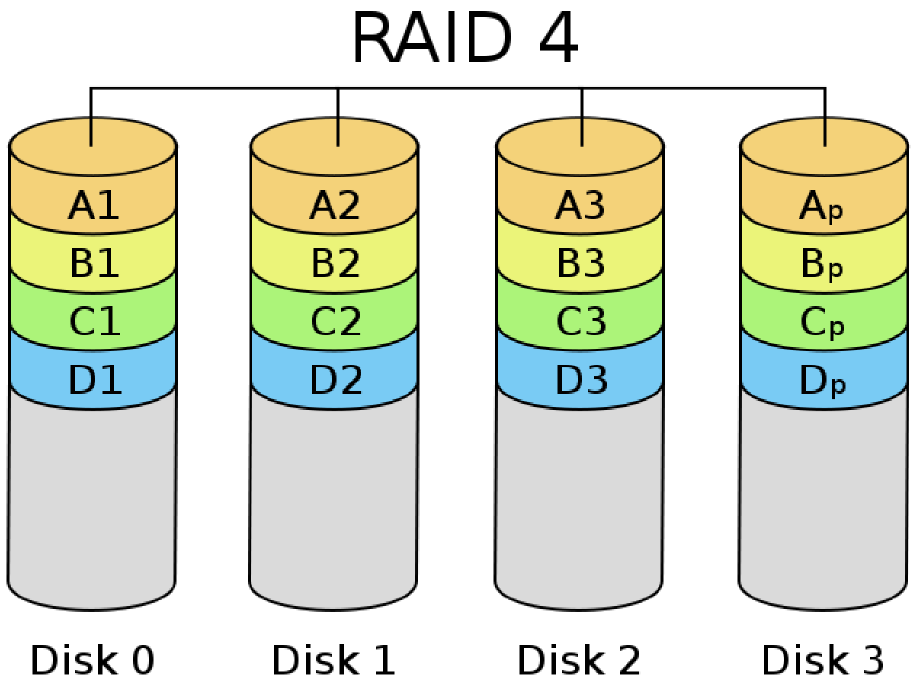


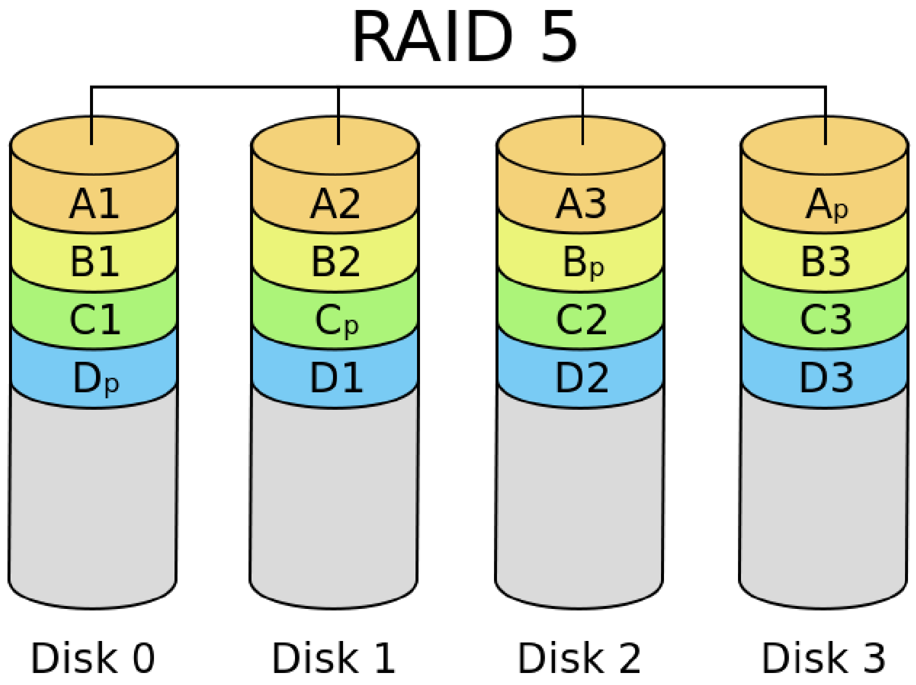
Countermeasures – RAID Technology





Countermeasures – RAID Technology





DBMSs and Web Security

- Internet communication relies on TCP/IP as the underlying protocol.
- However, TCP/IP and HTTP were not designed with security in mind.
- Without special software, all Internet traffic travels ‘in the clear’ and anyone who monitors traffic can read it.

DBMSs and Web Security

- Must ensure while transmitting information over the Internet that:
  - Inaccessible to anyone but sender and receiver (privacy);
  - Not changed during transmission (integrity);
  - Receiver can be sure it came from sender (authenticity);
  - Sender can be sure receiver is genuine (non-fabrication);
  - Sender cannot deny he or she sent it (non-repudiation).

DBMSs and Web Security

- Measures include:
  - Proxy servers
  - Firewalls
  - Message digest algorithms and digital signatures
  - Digital certificates
  - Secure sockets layer (SSL) and Secure HTTP (S-HTTP)
  - Secure Electronic Transactions (SET) and Secure Transaction Technology (SST)
  - Java security
  - ActiveX security

Outline

- Transaction Management*
- Database Security
- ✈ Database Administration

Typical Activities for Database

- Top management level
  - Provide the information necessary for strategic decision making, strategic planning, policy formulation, and goals definition.
  - Provide access to external and internal data to identify growth opportunities and to chart the direction of such growth.
  - Improve the likelihood of a positive return on investment for the company by searching for new ways to reduce costs and/or by boosting productivity.
  - Provide feedback to monitor whether the company is achieving its goals.

Typical Activities for Database

- Middle management level
  - Deliver the data necessary for tactical decisions and planning.
  - Monitor and control the allocation and use of company resources and evaluate the performance of the various departments.
  - Provide a framework for enforcing and ensuring the security and privacy of the data in the database.

Typical Activities for Database

- Operational management level
  - Represent and support the company operations as closely as possible.
    - The data model must be flexible enough to incorporate all required present and expected data.
  - Produce query results within specified performance levels.
    - Must support fast responses to a greater number of transactions at the operational management level.
  - Enhance the company’s short-term operational ability by providing timely information for customer support and for application development and computer operations.