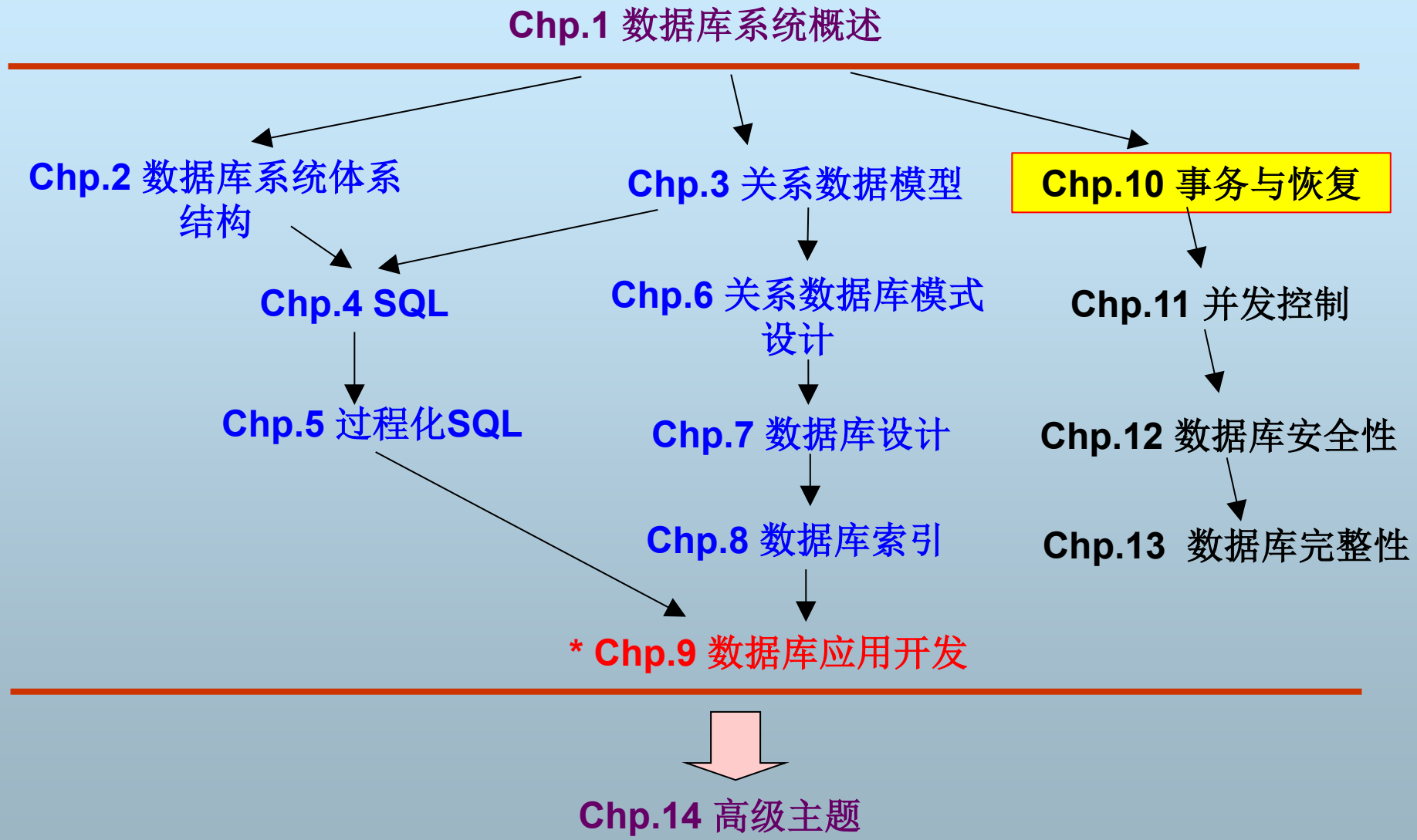
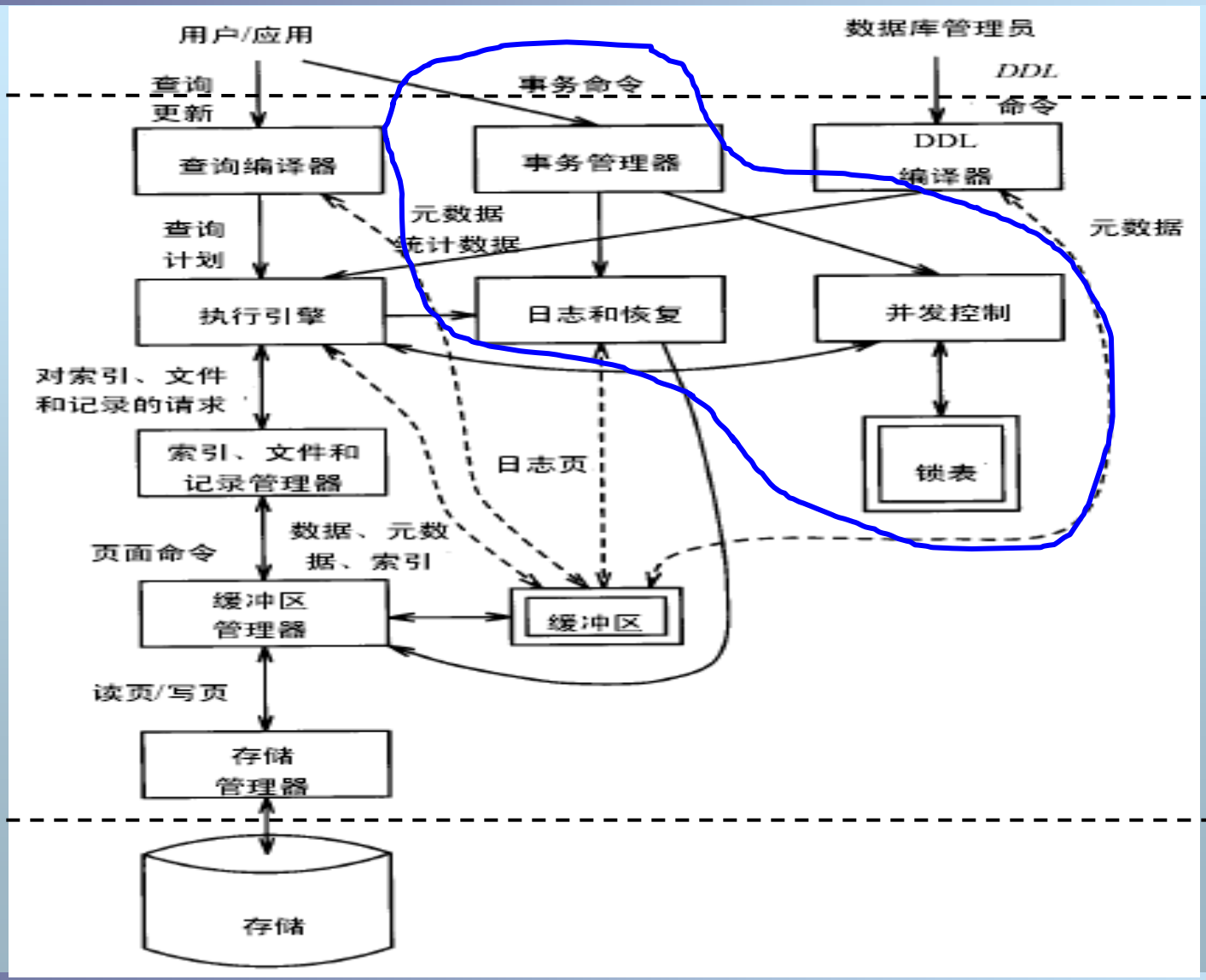
第10章 事务与恢复

金培权（jpq@ustc.edu.cn）

课程知识结构


Chp.1 数据库系统概述
Chp.2 数据库系统体系结构
Chp.3 关系数据模型
Chp.10 事务与恢复
Chp.4 SQL
Chp.6 关系数据库模式设计
Chp.11 并发控制
Chp.5 过程化SQL
Chp.7 数据库设计
Chp.12 数据库安全性
Chp.8 数据库索引
Chp.13 数据库完整性
* Chp.9 数据库应用开发
Chp.14 高级主题

DBMS架构



用户

DBMS

数据库

主要内容

- 事务
- 恢复

Part 1: 数据库事务

金培权（jpq@ustc.edu.cn）

一、事务的概念

- 事务(transaction)
  - 一个不可分割的操作序列，其中的操作要么都做，要么都不做

一、事务的概念

- 事务的例子
  - 银行转帐：A帐户转帐到B帐户100元。该处理包括了两个更新步骤
    - A=A-100
    - B=B+100
  - 这两个操作是不可分的：要么都做，要么都不作

二、事务的性质

- 事务的ACID性质
  - 原子性 Atomicity
    - 事务是不可分的原子，其中的操作要么都做，要么都不做
  - 一致性 Consistency
    - 事务的执行保证数据库从一个一致状态转到另一个一致状态
  - 隔离性 Isolation
    - 多个事务一起执行时相互独立
  - 持久性 Durability
    - 事务一旦成功提交，就在数据库永久保存

三、事务的状态

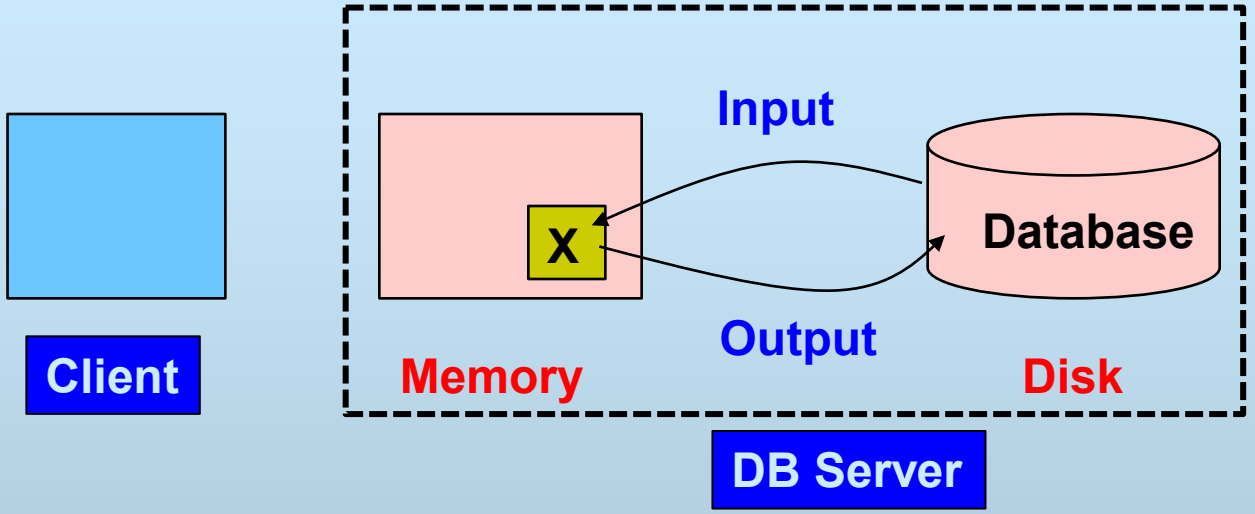
- <Start T>
  - Transaction T has started
- <Commit T>
  - T has finished successfully, and all modifications are reflected to disks
- <Abort T>
  - T has been terminated and all modifications have been canceled

四、事务的原语操作

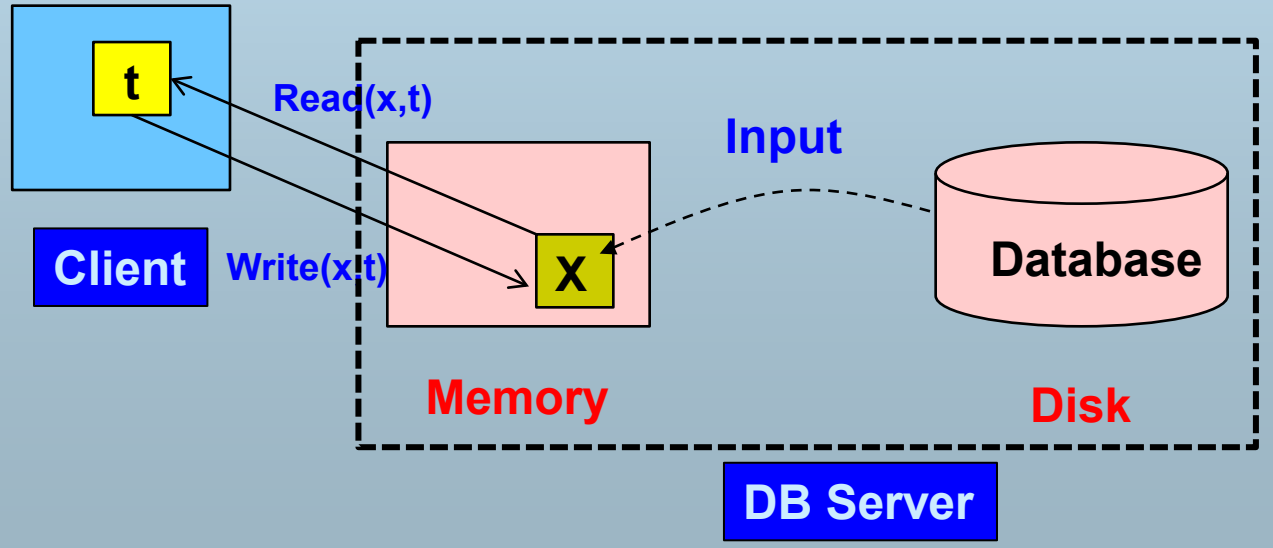
- **Input (x):** disk block with x → memory
- **Output (x):** buffer block with x → disk
- **Read (x,t):** do input(x) if necessary t ← value of x in buffer
- **Write (x,t):** do input(x) if necessary value of x in buffer ← t

四、事务的原语操作

- Input (x)
- Output (x)



- Read (x,t)
- Write (x,t)



1、事务例子

A bank transfer

```
T1:   Read (A,t);
      t ← t −100;
      Write (A,t);
      Read (B,t);
      t ← t + 100;
      Write (B,t);
      Output (A);
      Output (B);
```

- 这两个操作是Client端的本地操作
- DBMS的事务管理器实际上是不知道这些操作的。
- 此处只为了便于大家理解事务的操作过程

2、SQL对事务的支持

- SQL标准提供了三个语句，允许应用程序声明事务和控制事务
  - Begin Transaction
  - Commit Transaction
  - Rollback Transaction
- MySQL
  - Start transaction/commit/rollback
- Oracle
  - Commit或Commit Work
  - Rollback或Rollback Work

3、存储过程中使用事务

```
1
2   delimiter //
3   CREATE PROCEDURE transfer(IN id_from INT, IN id_to INT, IN amount INT, OUT state INT)
4   BEGIN
5     DECLARE s INT DEFAULT 0;
6     DECLARE a INT;
7     DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET s = 1;
8     START TRANSACTION;
9     SELECT count(*) FROM account WHERE id = id_from or id=id_to INTO a;
10    IF  a < 2 THEN -- 至少有一个账户不存在
11        SET s = 2;
12    END IF;
13
14    SELECT balance FROM account WHERE id = id_from INTO a;
15    IF  a < amount THEN  -- 余额不足
16        SET s = 3;
17    END IF;
18    UPDATE account SET balance = balance - amount WHERE id = id_from;
19    UPDATE account SET balance = balance + amount WHERE id = id_to;
20    IF s = 0 THEN
21        SET state = 0;
22        COMMIT;
23    ELSE
24        SET state = -1000;
25        ROLLBACK;
26    END IF;
27  END //
28  delimiter;
```

4、ADO中使用事务编程

- ADODB.Connection对象主要的方法
  - Open，Close
  - Execute ’可执行SQL语句
  - BeginTrans，CommitTrans，RollbackTrans ’用于事务编程
- 示例

```
Dim cnn as New ADODB.Connection

cnn.Connectstring="DSN=Mysql; SERVER=192.168.1.11; UID=root;
PWD=root; PORT=3306; DATABASE=mydb"

Cnn.CursorLocation=adUseClient

Cnn.Open
```

5、ADO中使用事务编程 (VB)

```
cnn.Open
On Error Goto RollbackAll '错误陷阱
cnn.BeginTrans '此连接下的所有操作现在开始都属于一个事务
Dim rst1, rst2 as New ADODB.Recordset '执行记录的增删改
rst1.Open "account", cnn, adUseClient, adOpenKeyset, adLockOptimistic, adCmdTable
rst1.AddNew '增加新记录
.....
rst2.Open "summary", cnn, adUseClient, adOpenKeyset, adLockOptimistic, adCmdTable
.....'更新关联的summary表
'当发生任何预期错误时，RollbackTrans
If rst2.EOF and rst2.BOF Then
   Goto RollbackAll
End If
.....
cnn.CommitTrans '成功到达事务尾部时，提交事务
cnn.Close
RollbackAll: 'Rollback事务的操作统一进行处理
   cnn.RollbackTrans
   cnn.Close
```

五、数据库一致性

- 一致性 (Consistency)
- 正确性 (Correctness)

1、Consistency

Integrity or consistency constraints

- **Predicates data must satisfy**
- **Examples:**
  - **x is key of relation R**
  - **x → y holds in R**
  - **Domain(x) = {Red, Blue, Green}**

1、Consistency

- **Consistent state:** satisfies all integrity constraints
- **Consistent DB:** DB in consistent state

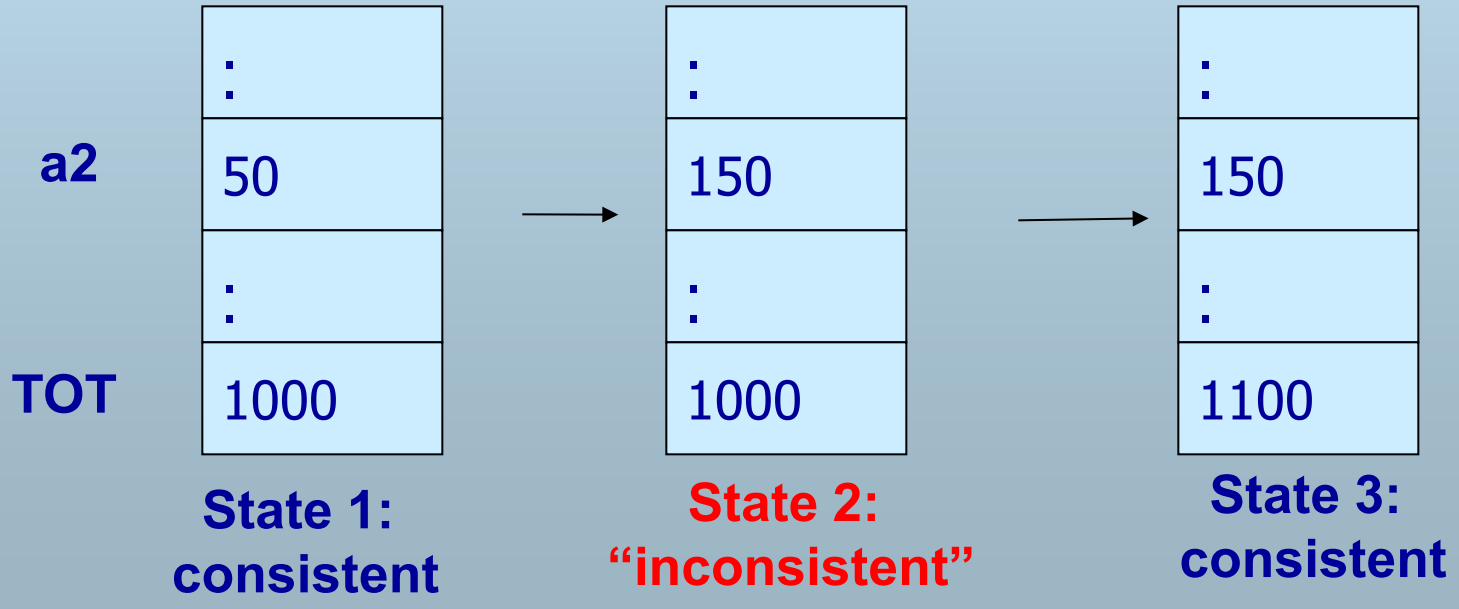
1、Consistency

- **DB will not always satisfy constraints**

Example: a1 + a2 +…. an = TOT (constraint)

Transaction: Deposit $100 in a2: a2 ← a2 + 100

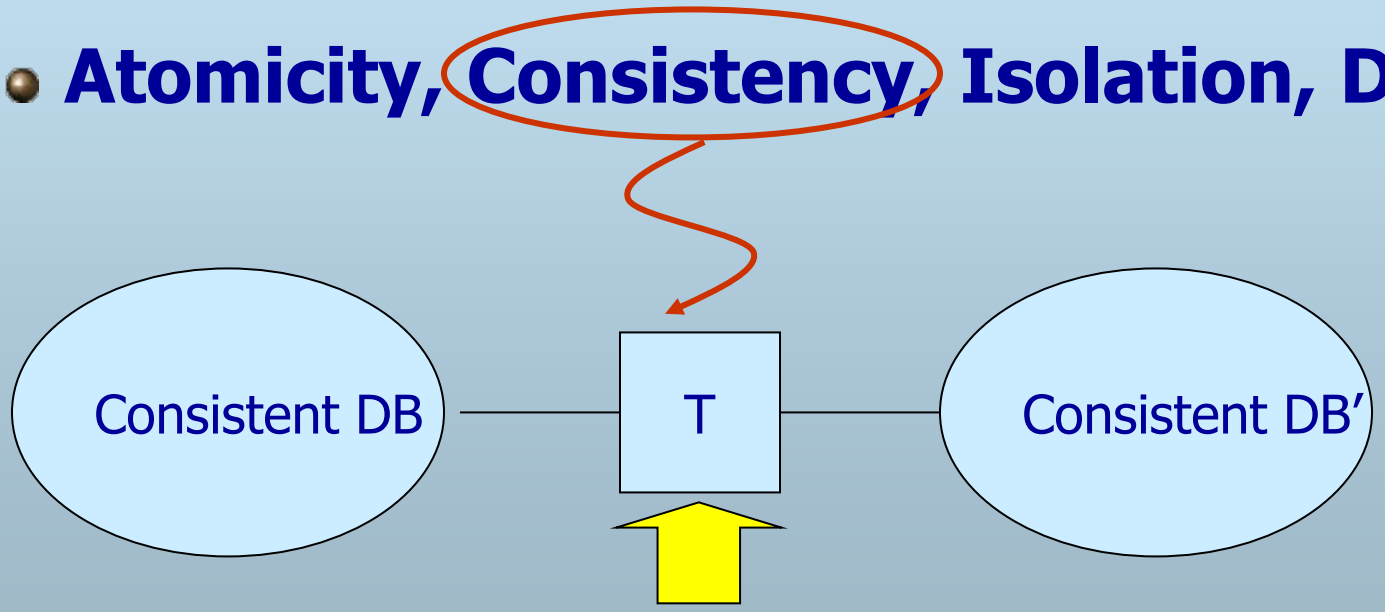
TOT ← TOT + 100



1、Consistency

Consistency of Transaction

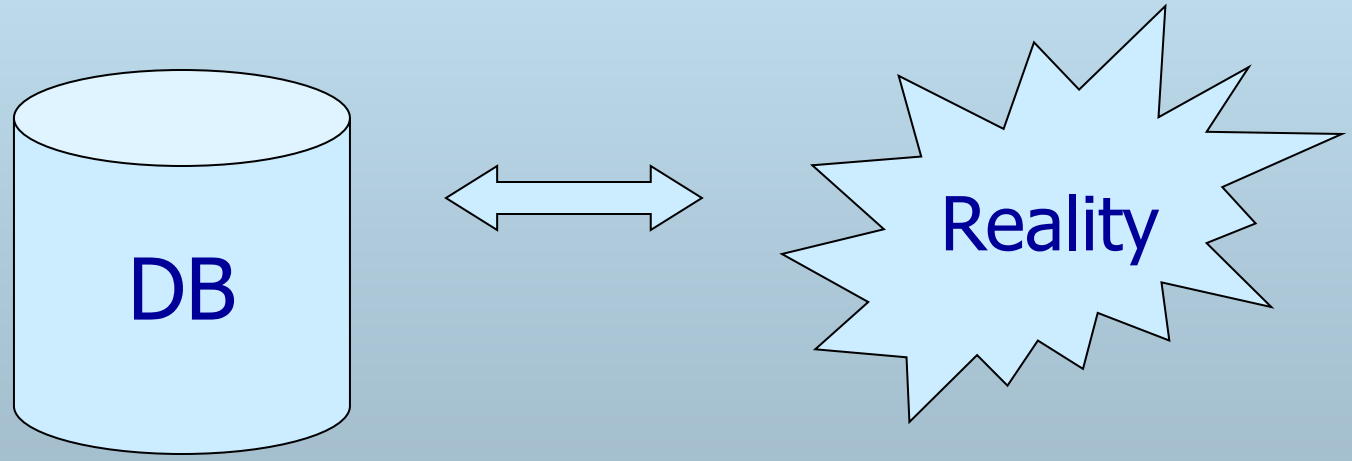
- 事务的ACID性质
  - Atomicity, Consistency, Isolation, Durability



但事务内部允许不保证DB的一致性

2、Correctness

DB should reflect real world



2、Correctness

DB should reflect real world

Example：A telephone number

63600110 —— Correct

abcdefgh —— Not correct

Can be preserved by explicit constraints！

CHECK (tel LIKE '[1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

2、Correctness

DB should reflect real world

Example： A telephone number

63600110 —— Correct

90000000 —— Is it correct?

Not correct in reality, but can DB know this?

Answer: NO!

2、Correctness

Correctness of DB ≠ Correctness of reality

Correctness of DB

如果数据库在事务开始执行时是一致的，并且事务执行结束后数据库仍处于一致状态，则数据库满足正确性.

Consistency of DB + ACID of transaction

Correctness of DB

通常数据库中使用“一致性”来表示数据库的正确性。很少使用“正确性”

事务小结

- 事务的概念
- 事务的性质
- 事务的状态
- 事务的原语操作
- 数据库一致性