



数据库原理与应用

西南交通大学电气工程学院

西南交通大学

第七章 数据库设计

为什么需要设计数据库?



修建茅屋需要设计吗?



修建大厦需要设计吗?

结论：当数据库比较复杂时需要设计数据库

为什么需要设计数据库?

□良好的数据库设计

- 节省数据的存储空间
- 能够保证数据的完整性
- 方便进行数据库应用系统的开发

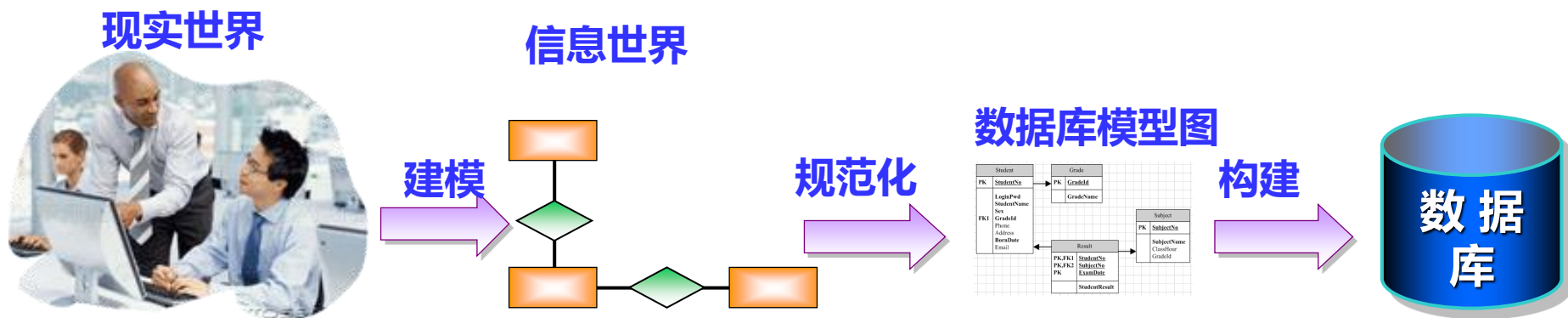
□糟糕的数据库设计

- 数据冗余、存储空间浪费
- 内存空间浪费
- 数据更新和插入的异常

□ 数据库设计

- 对给定的应用环境，设计优化的**数据库逻辑模式**和**物理结构**，并据此建立**数据库及其应用系统**，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括**信息管理要求**和**数据操作要求**
 - 结构设计
 - 行为设计

数据库设计步骤



□ 需求分析阶段：分析客户的业务和数据处理需求

□ 概念结构设计阶段：设计数据库的**E-R模型图**，
确认需求信息的正确和完整

独立于
DBMS

□ 逻辑结构设计阶段：应用**范式**审核数据库结构

□ 物理结构设计阶段：物理实现数据库

□ 数据库实施

□ 数据库运行和维护

与DBMS
有关

一、需求分析

- 收集信息，综合各个用户的应用需求
 - 与有关人员进行交流、座谈，充分了解用户需求，理解数据库需要完成的任务

示例——酒店管理系统的基本功能：

- 旅客办理入住手续：后台数据库需要存放入住客人的信息和入住客房信息
- 客房信息：后台数据库需要存放客房的相关信息，如房间号、床位数、价格等
- 客房管理：后台数据库需要保存客房类型信息和客房当前状态信息

二、概念结构设计

□ 标识实体 (Entity)

- 标识数据库要管理的关键对象或实体，实体一般是名词

酒店管理系统中的实体：

- 客人：入住酒店的旅客。办理入住手续时，需要填写客人的信息
- 客房：酒店为客人提供休息的房间

二、概念结构设计

□ 标识每个实体的属性 (Attribute)

客人信息

客人姓名

身份证号

入住日期

结帐日期

押金

总金额

客房信息

客房号

床位数

入住人数

客房状态

客房类型

价格

□ 标识实体之间的关系 (Relationship)

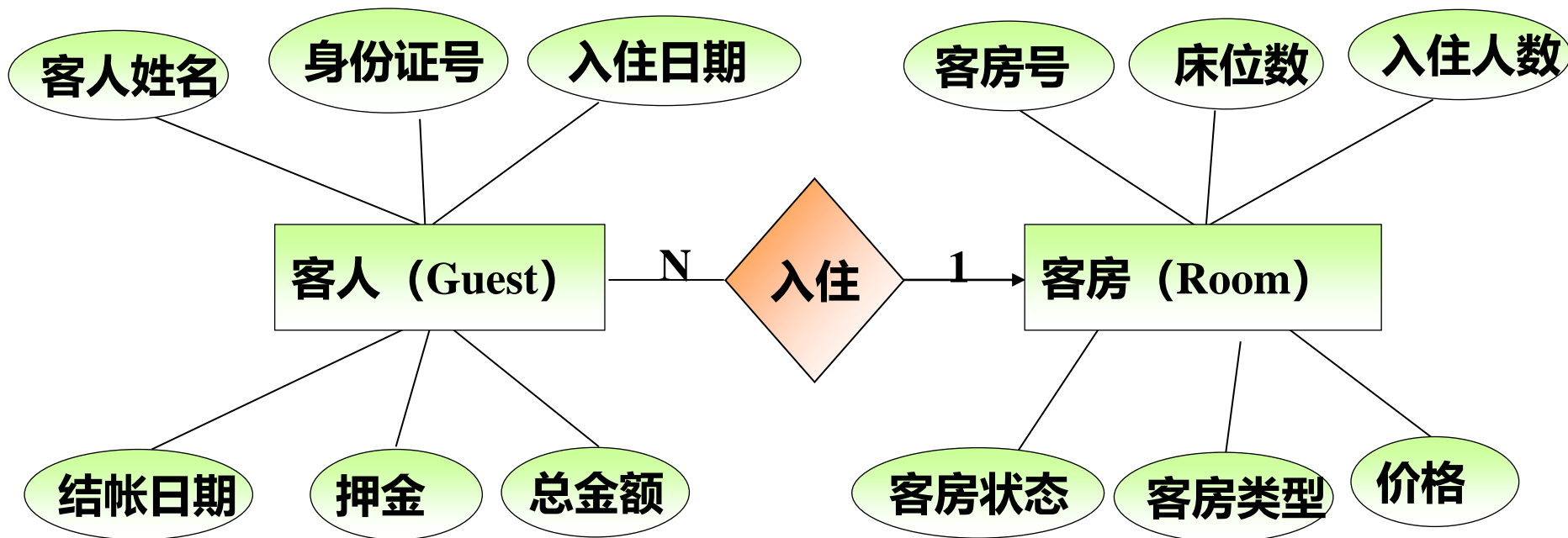
酒店管理系统中实体之间关系：

客房和客人有主从关系：表明客人入住的房间

二、概念结构设计

□ 通过对用户需求进行综合、归纳与抽象，形成一个独立于具体DBMS的**概念模型**

■ 酒店管理系统的ER图



三、逻辑结构设计

□ 将概念结构转换为某个DBMS所支持的数据模型，形成**模式**和**外模式**。

■ 酒店管理系统数据库的关系模式：

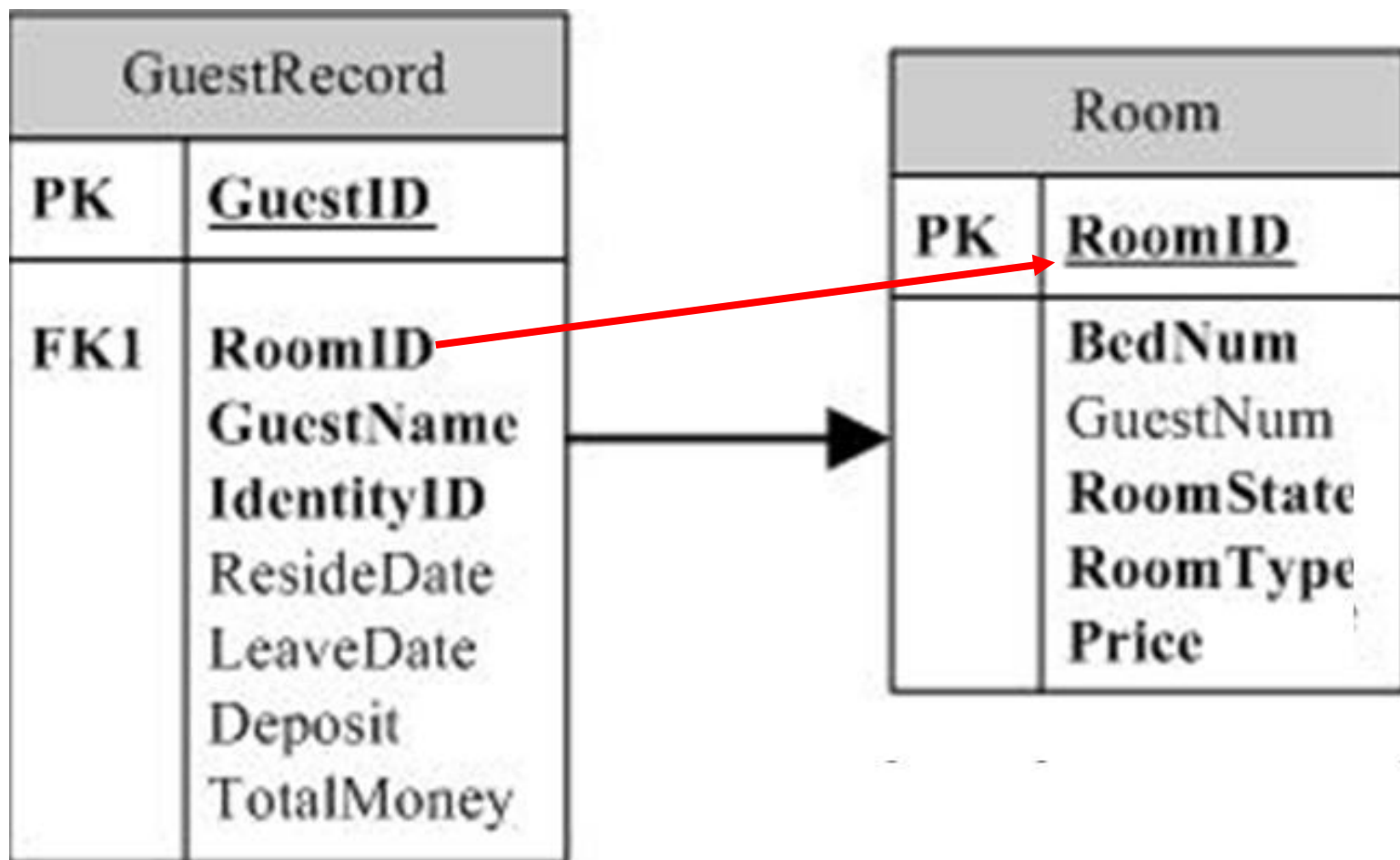
- 客人（客人姓名、身份证号、入住日期、结帐日期、押金、总金额、**客房号**）
- 客房（**客房号**、床位数、入住人数、客房状态、客房类型、价格）

■ 标识每个表的主码，需要注意的是：没有主码的表添加ID编号列，它没有实际含义，用做主码或外码

- 如：客人信息表中增加“客人ID”列

■ 在表之间建立**外码**，体现实体之间的关系

三、逻辑结构设计



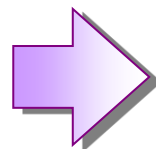
三、逻辑结构设计

□ 数据规范化 第三范式 (3NF)

Room

字段	例子
客房号	1001
床位数	1
入住人数	0
客房状态	空闲
<u>客房类型</u>	标准单间
价 格	\$108.00

Room ∈ 3NF ?



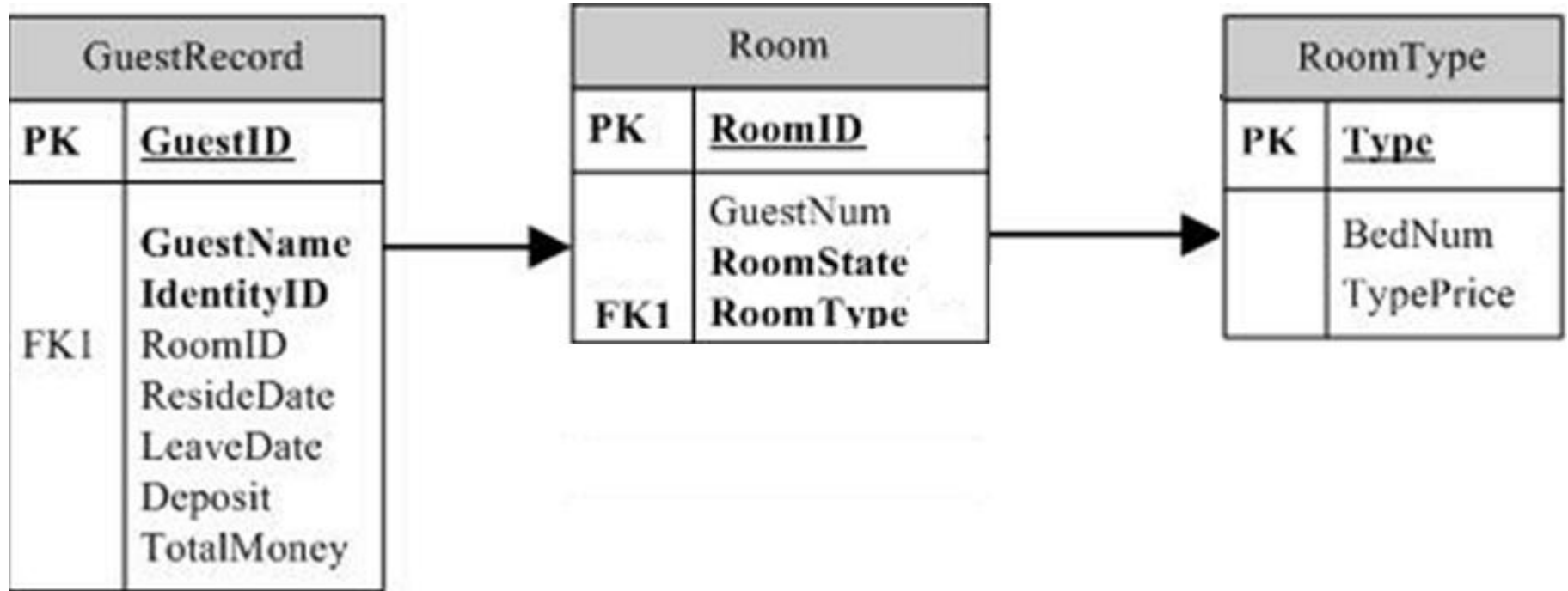
Room

字段	例子
客房号	1001
入住人数	0
客房状态	空闲
客房类型	标准单间

RoomType

字段	例子
客房类型	标准单间
床位数	1
价 格	\$108.00

规范化的酒店管理系统数据库模型图



四、物理结构设计

□ 选取一个最适合应用环境的物理结构（包括存储结构和存取方法），形成**内模式**。

数据库设计的基本步骤

五、数据库实施阶段

- 运用DBMS提供的数据库语言（如SQL）及宿主语言，根据逻辑设计和物理设计的结果
 - 建立数据库
 - 编制与调试应用程序（与数据设计并行进行）
 - 组织数据入库
 - 进行试运行

六、数据库运行和维护阶段

数据库的试运行

- 在有部分数据输入数据库后，就可以开始对数据库系统进行联合调试，称为数据库的**试运行**
- 数据库试运行主要工作包括：
 - 1) **功能测试**
 - 实际运行数据库应用程序，执行对数据库的各种操作，测试应用程序功能是否满足设计要求
 - 2) **性能测试**
 - 测量系统的性能指标，分析是否达到设计目标
- 数据库试运行合格后，数据库即可投入**正式运行**

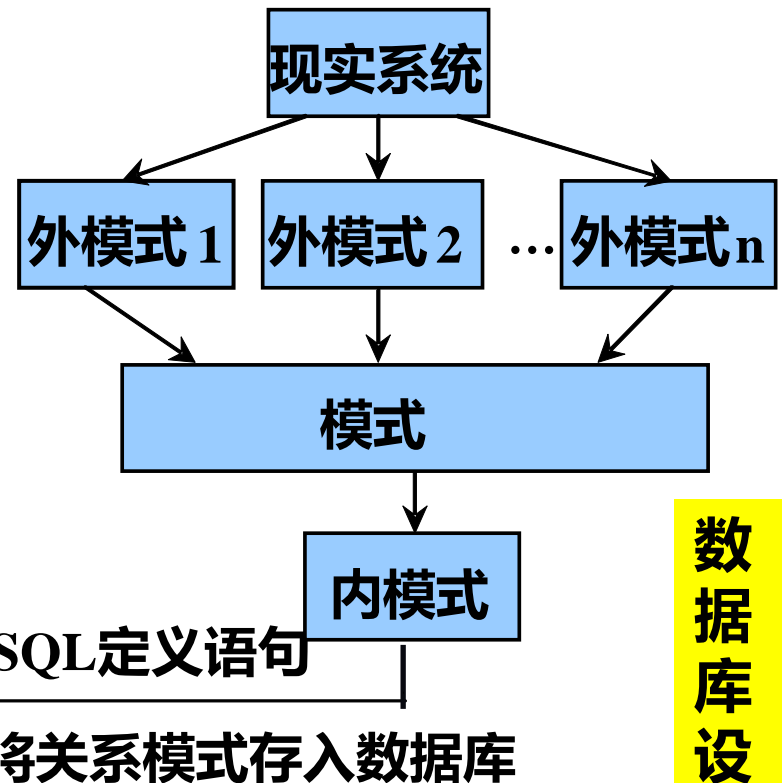
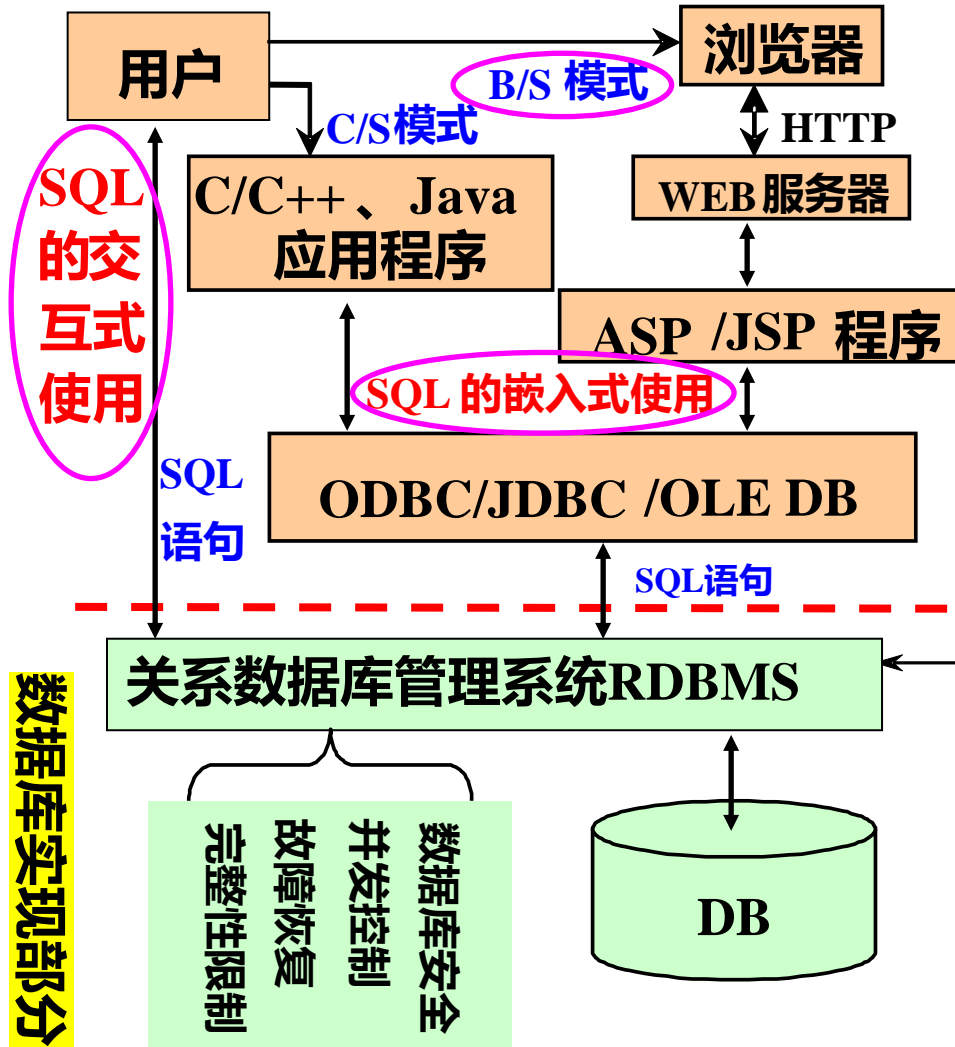
示例：酒店管理系统

单号：	<input type="text"/>	房号：	<input type="text"/>	姓名：	<input type="text"/>	<input type="button" value="查询"/>
-----	----------------------	-----	----------------------	-----	----------------------	-----------------------------------

单号	房号	房型	姓名	入住时间	预离时间	房价
201412260001	202	标准双人间	陈明	2014-12-26 15:20:18	2015-01-13 13:00:00	138.00
201412260003	404	标准单人间	孙群	2014-12-26 15:20:59	2015-01-13 13:00:00	108.00
201412260004	208	标准双人间	张鸣	2014-12-26 15:21:16	2015-01-13 13:00:00	138.00
201501060001	105	标准双人间	彭奇	2015-01-06 13:43:18	2015-01-13 13:00:00	138.00
201501120003	101	标准双人间	华为团队	2015-01-12 13:52:46	2015-01-13 13:00:00	138.00
201501120004	104	标准双人间	sdas	2015-01-12 13:54:26	2015-01-13 13:00:00	138.00
201501120005	204	标准双人间	dwd	2015-01-12 16:03:22	2015-01-13 13:00:00	138.00

数据库设计、实现与应用的关系示意

数据库应用部分



数据库的运行与维护

- 在数据库系统运行过程中必须不断地对其进行评价、调整与修改

设计一个完善的数据库应用系统往往是上述六个阶段的不断反复。

- 数据库投入运行标志着开发任务的基本完成和维护工作的开始

- 应用环境在不断变化
- 数据库运行过程中物理存储会不断变化

课堂练习

1、数据库设计可划分为六个阶段，每个阶段都有自己的设计内容。“为哪些关系，在哪些属性上建什么样的索引”这一设计内容应该属于（ C ）阶段。

A. 概念设计

B. 逻辑设计

C. 物理设计

D. 全局设计

2、在数据库设计中，将ER图转换成关系数据模型的过程属于（ B ）阶段。

A.概念设计

B.逻辑设计

C.物理设计

D.需求分析

课堂练习

- 3、一个好的关系模式应当不发生数据冗余、插入异常、删除异常和更新异常。 **错误**
- 4、关系模式的分解不唯一。 **正确**
- 5、在数据库逻辑设计过程中，规范化程度越高的关系，其所对应数据库应用系统的性能就优越。 **错误**
- 6、在E-R图向关系模型转换的过程中，无论是一个1:1的联系，一个1: n的联系，还是一个m: n的联系，都可以转换为一个独立的关系模式。 **正确**

第十章 数据库恢复技术

10.1 事务的基本概念

10.2 数据库恢复概述

10.3 故障的种类

10.4 恢复的实现技术

事务是系统或用户定义的一组操作序列。这些操作要么全做，要么全不做，是一个不可分割的工作单位。

□ 事务和程序是两个概念

- 在关系数据库中，一个事务可以是一条SQL语句，一组SQL语句或整个程序
- 一个应用程序通常包含多个事务

□ 事务是恢复和并发控制的基本单位

10.1 事务的基本概念

□ 显示定义事务的方式:

BEGIN TRANSACTION

SQL语句1

SQL语句2

.....

COMMIT

BEGIN TRANSACTION

SQL语句1

SQL语句2

.....

ROLLBACK

■ 事务正常结束

- 提交事务的所有操作（读写）
- 事务中所有对数据库的更新写回到磁盘上的物理数据库中

■ 事务异常终止

- 将事务中对数据库的所有已完成的操作全部撤销
- 事务滚回到开始时的状态

□ 隐式方式：未显式定义事务时，DBMS按缺省规定自动划分事务

10.1 事务的基本概念

□ 事务的四大特性 (ACID) :

(1)原子性(Atomicity): 事务是不可分割的数据库逻辑工作单位。事务中的所有操作要么成功执行, 要么都不执行。

(2)一致性(Consistency): 事务执行的结果应保证DB从一个一致性状态 (**正确状态**) 转到另一个一致性状态。

□ **一致性状态:** 数据库中只包含成功事务提交的结果

□ **不一致状态:**

- 数据库系统运行中发生故障, 有些事务尚未完成就被迫中断;
- 这些未完成事务对数据库所做的修改有一部分已写入物理数据库, 这时数据库就处于一种不正确的状态。

示例——银行转帐：从A帐号取出一万元存入B帐号

□ **定义一个事务，该事务包括两个操作：**

■ $A = A - 10000$

■ $B = B + 10000$

原子性

□ **这两个操作要么全做，要么全不做**

■ **全做或者全不做，数据库都处于一致性状态。**

■ **如果只做一个操作，用户逻辑上就会发生错误，少了一万元，数据库就处于不一致性状态。**

10.1 事务的基本概念

□ 事务的四大特性 (ACID) :

(3)隔离性(Isolation): 一个事务的执行不能被其它事务干扰。

- 一个事务内部的操作及使用的数据对其他并发事务是隔离的
- 并发执行的各个事务之间不能互相干扰

(4)持续性(Durability): 一旦事务成功提交, 对数据库的影响是持久的。

- 保证事务ACID特性是事务管理的任务
- 破坏事务ACID特性的因素
 - 多个事务并行运行时, 不同事务的操作交叉执行
 - 事务在运行过程中被强行停止

10.2 数据库恢复概述

□ 故障不可避免

□ 故障的影响

- 运行事务非正常中断，影响数据库中数据的正确性
- 破坏数据库，全部或部分丢失数据

□ 数据库的恢复

- DBMS提供**恢复子系统**，保证故障发生后，能把数据库从错误状态恢复到某一已知的正确状态(一致状态)

□ **恢复子系统是数据库管理系统的一个重要组成部分**

10.3 故障的种类

1、事务内部故障

事务在运行过程中由于种种原因未运行至正常结束

(1) 可预期的：

应用程序可以发现的故障，如违反了某些完整性限制、输入数据有误等，可由应用程序处理。

(2) 不可预期的：

应用程序无法处理此类故障，如溢出、发生死锁等

■ 以后，事务故障仅指这类**非预期的故障**

如银行转账事务

BEGIN TRANSACTION

读帐户甲的余额BALANCE1;

BALANCE1 = BALANCE1-AMOUNT; (AMOUNT 转帐金额)

IF (BALANCE1<0) THEN

{ 打印'金额不足，不能转帐';

ROLLBACK; (撤消刚才的修改，恢复事务) }

若账户甲余额不足，应用程序可发现并让事务滚回，撤销已作的修改，恢复数据库到正确状态。

ELSE

{ 读帐户乙的余额BALANCE2;

BALANCE2 = BALANCE2+AMOUNT;

写回BALANCE2;

COMMIT;}

保证了两个更新操作要么全部完成要么全部不做。

10.3 故障的种类

2、系统故障

- 造成系统停止运转的任何事件，使系统需重新启动
- 系统故障的原因：操作系统或DBMS 代码错误、特定类型的硬件错误（如CPU 故障）、突然停电等。

3、介质故障

- 指外存故障。介质故障发生的可能性较小，但破坏性大，破坏数据库或部分数据库。
- 原因：磁盘损坏、磁头碰撞、瞬时强磁场干扰等。

4、计算机病毒

- 人为的故障或破坏，破坏、盗窃系统中的数据

10.3 故障的种类

□ 各类故障对数据库的影响：

- 数据库本身被破坏；
- 数据库没有被破坏，但数据可能不正确。

□ 恢复操作的基本原理：冗余

- 利用存储在系统其它地方的冗余数据来重建数据库中已被破坏或不正确的那部分数据

□ 恢复机制涉及的两个关键问题：

- 第一，如何建立冗余数据；(数据转储和登记日志文件)
- 第二，如何利用这些冗余数据实施数据库恢复。

10.4.1 数据转储

- **转储**是指DBA将整个数据库复制到磁带、磁盘或其他存储介质上保存起来的过程。
- 备用的数据称为**后备副本**或**后援副本**
- **如何使用**
 - 数据库遭到破坏后可以将后备副本重新装入
 - 重装后备副本只能将数据库恢复到**转储时的状态**
 - 要想恢复到故障发生时的状态，必须重新运行自转储以后的所有更新事务

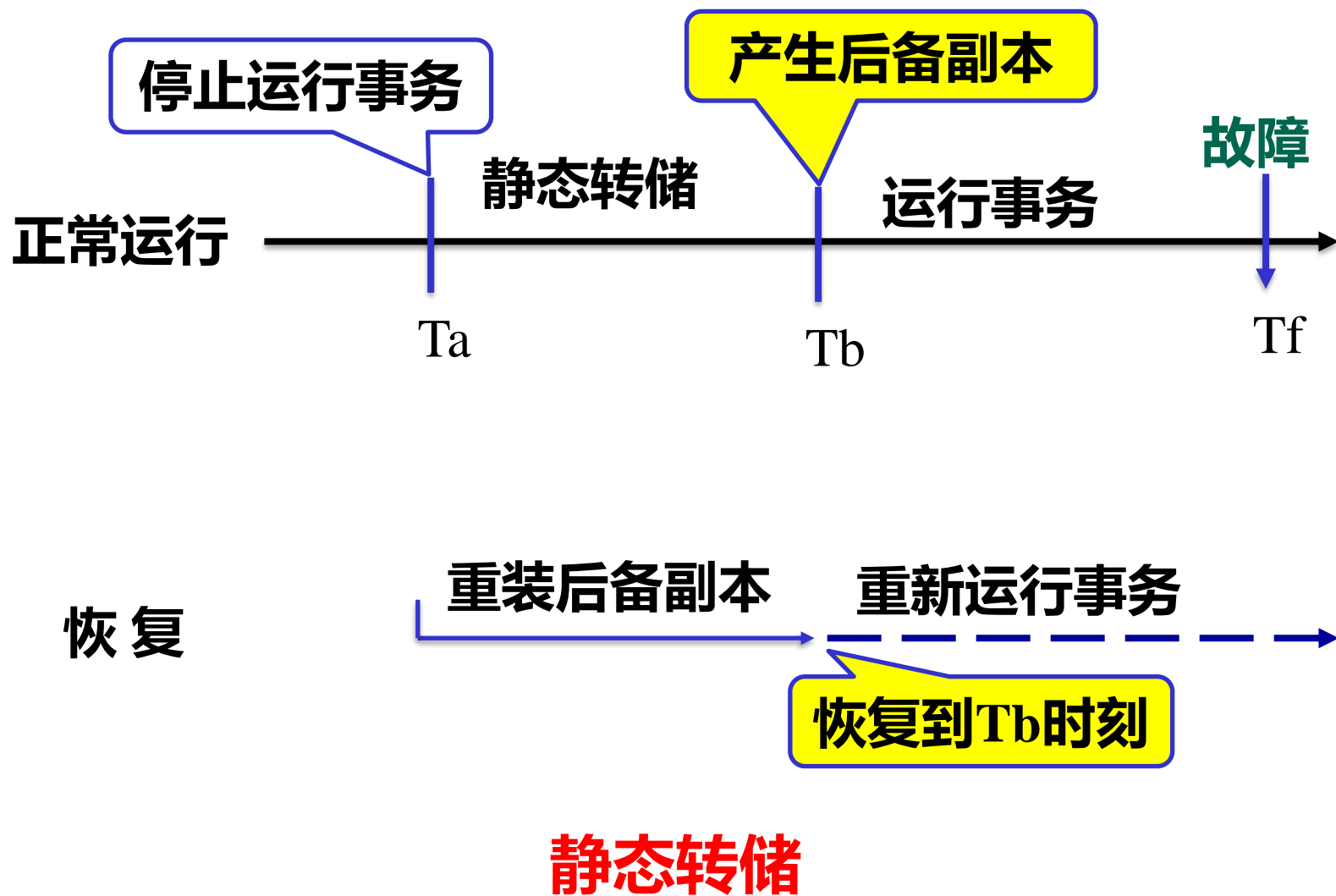
一、静态转储与动态转储

(1) 静态转储

- 系统中无运行事务时执行的转储。
- 转储期间不允许对数据库有任何存取、修改活动。
- 得到的一定是一个数据一致性的副本。
- 优点：简单
- 缺点：降低了数据库的可用性
 - 转储必须等用户事务结束
 - 新的事务必须等转储结束

静态转储得到的后备副本可将数据库恢复到某一时刻的正确状态

一、静态转储与动态转储



一、静态转储与动态转储

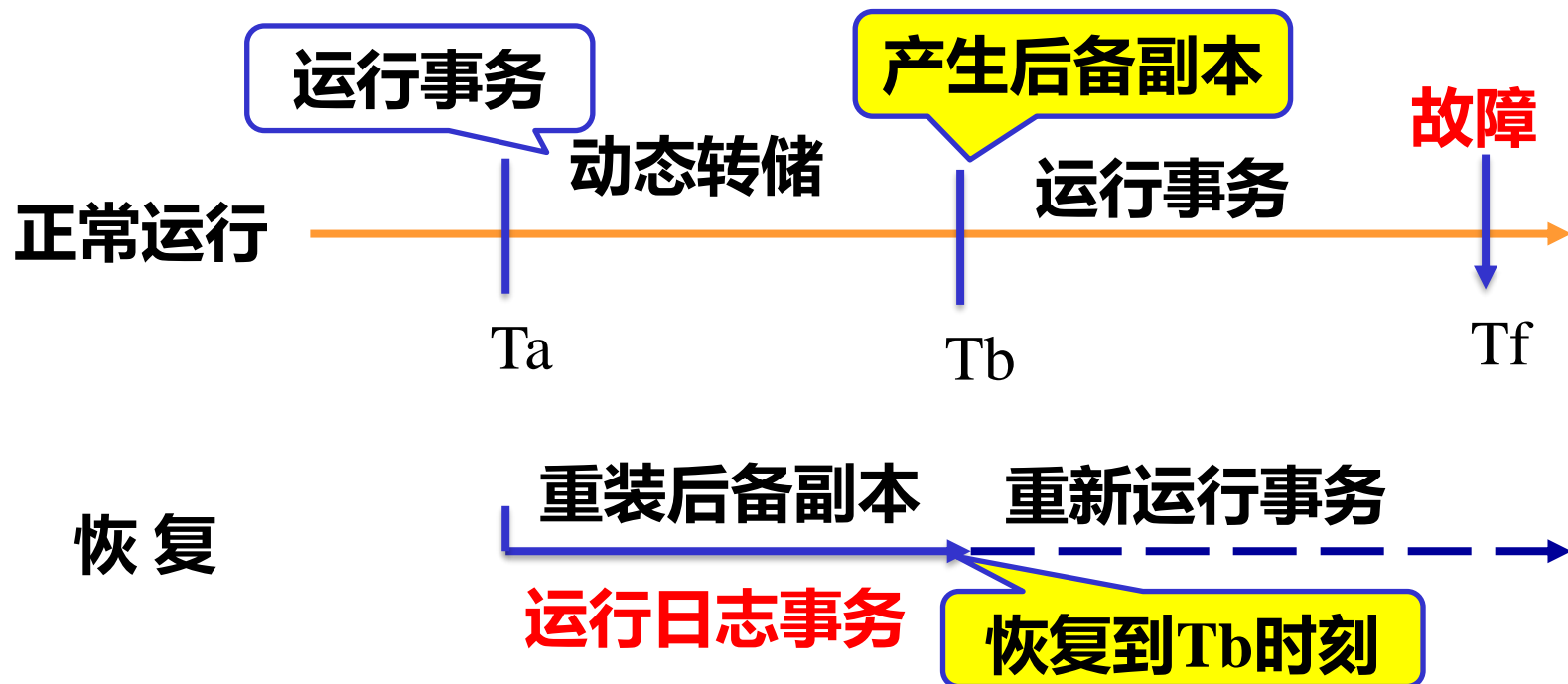
(2) 动态转储

- 转储操作与用户事务并发进行
- 转储期间允许对数据库进行存取或修改
- 优点
 - 不用等待正在运行的用户事务结束
 - 不会影响新事务的运行
- 动态转储的缺点
 - 不能保证副本中的数据正确有效
- 动态转储得到的后备副本与数据库一般不一致。

一、静态转储与动态转储

□ 利用动态转储得到的副本进行故障恢复

- 需要把动态转储期间各事务对数据库的修改活动登记到**日志文件**
- **后备副本加上日志文件**才能把数据库恢复到某一时刻的**正确状态**



动态转储

二、海量转储与增量转储

- **海量转储**：每次转储全部数据库
- **增量转储**：只转储上次转储后更新过的数据
- **海量转储与增量转储比较**
 - 从恢复角度看，使用海量转储得到的后备副本进行恢复往往更方便
 - 如果数据库很大，事务处理又十分频繁，则增量转储方式更实用更有效

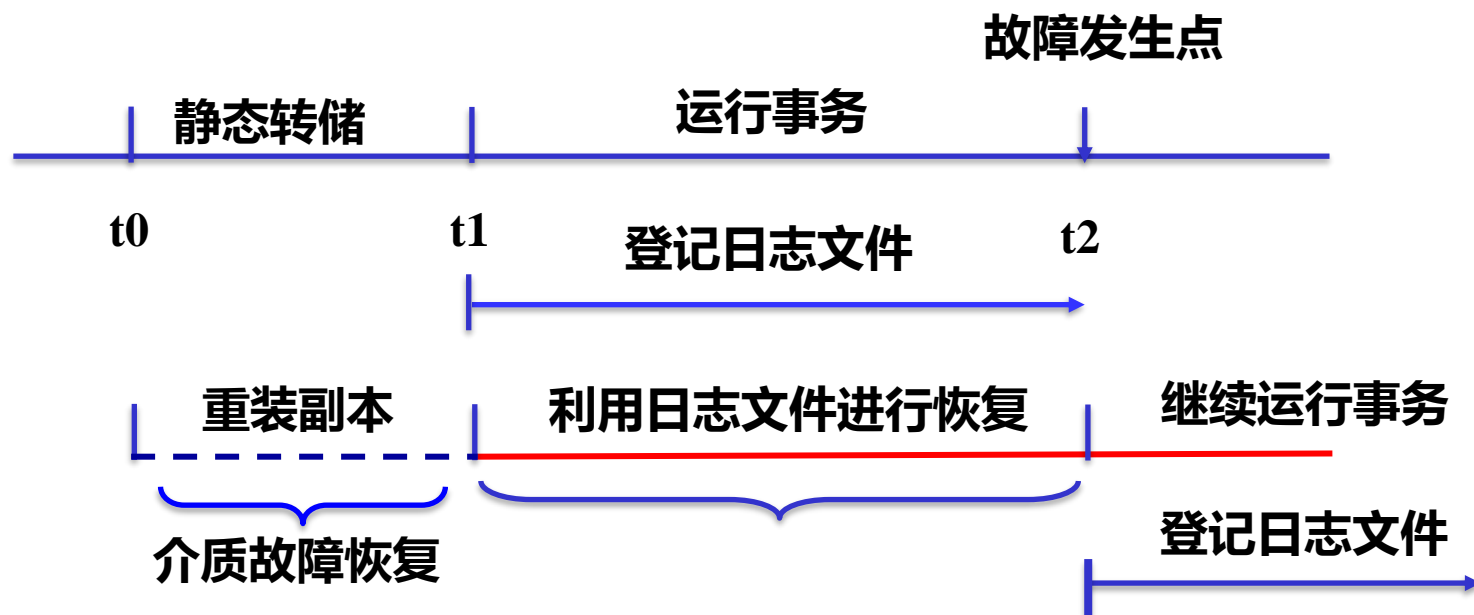
DBA通常根据数据库使用情况，确定一个适当的转储周期，并配合使用这类4类方法。

转储方法分类

转储方式	转储状态	
	动态转储	静态转储
海量转储	动态海量转储	静态海量转储
增量转储	动态增量转储	静态增量转储

10.4.2 登记日志文件

日志文件是用来记录事务对数据库**更新操作**的文件。



利用日志文件恢复

日志文件的作用

- 事务故障恢复和系统故障恢复必须用日志文件。
- 在动态转储方式中必须建立日志文件，后备副本和日志文件结合起来才能有效地恢复数据库。
- 在静态转储方式中，也可以建立日志文件。
 - 故障恢复时重新装入后备副本把数据库恢复到转储时刻的正确状态
 - 利用日志文件，重做已完成事务，撤销未完成的事务
 - 不必重新运行那些已完成的事务程序就可把数据库恢复到故障前某一时刻的正确状态

□ 为保证数据库可恢复，登记日志文件时须遵循两条原则：

■ 登记的次序**严格按照**并行事务执行的时间次序。

■ **必须先写日志文件，后写数据库。**

○ 写日志文件操作：把表示这个修改的日志记录写到日志文件

○ 写数据库操作：把对数据的修改写到数据库中

□为什么要先写日志文件

- 写数据库和写日志文件是两个不同的操作
- 在这两个操作之间可能发生故障
- 如果先写了数据库修改，而在日志文件中没有登记下这个修改，则以后就无法恢复这个修改了
- 如果先写日志但没有修改数据库，按日志文件恢复时只不过是多执行一次不必要的UNDO操作，不会影响数据库的正确性

恢复策略

事务故障：撤消事务 (UNDO)

系统故障：

事务未提交：撤消未完成事务 (UNDO)

事务已提交：重做已完成事务 (REDO)

介质故障：

重装后备副本

重做已完成事务 (REDO)

课堂练习

1、在DBMS中实现事务持久性的子系统是(**D**)

A. 安全管理子系统

B. 完整性管理子系统

C. 并发控制子系统

D. 恢复管理子系统

2、表示事务执行成功的语句是 (**A**)

A.COMMIT

B.RETURN

C.ROLLBACK

D.OK

3、DB的转储属于DBS的 (**D**)

A. 完整性措施

B. 安全性措施

C. 并发控制措施

D. 恢复措施

课堂练习

4、在数据库故障恢复模块中，日志记录了数据库中所有的更新活动。

5、常见的数据库故障分为：事务故障、系统故障、介质故障和计算机病毒。

6、在多个事务并发执行时，系统应保证与这些事务先后单独执行时的结果一样，这是指事务的隔离性。

7、事务是DBMS的基本单位，它是用户定义的一组逻辑一致的程序序列。

8、SQL语言中用rollback语句实现事务的回滚。

4、考虑下图所示的日志记录，初始A、B、C的值都是0

(1)如果系统故障发生在14之后，说明哪些事务需要重做，哪些事务需要回滚。系统恢复后A、B、C的值

系统故障 { 事务已提交：重做
事务未提交：撤消

需要重做的事务有：T1、T3

需要回滚的事务有：T4

A=8, B=7, C=11

序号	日志
1	T ₁ : 开始
2	T ₁ : 写 A=10
3	T ₂ : 开始
4	T ₂ : 写 B, B=9
5	T ₁ : 写 C, C=11
6	T ₁ : 提交
7	T ₂ : 写 C, C=13
8	T ₃ : 开始
9	T ₃ : 写 A, A=8
10	T ₂ : 回滚
11	T ₃ : 写 B, B=7
12	T ₄ : 开始
13	T ₃ : 提交
14	T ₄ : 写 C, C=12

4、考虑下图所示的日志记录，初始A、B、C的值都是0

(2)如果系统故障发生在10之后，说明哪些事务需要重做，哪些事务需要回滚。系统恢复后A、B、C的值

需要重做的事务有：T1

需要回滚的事务有：T3

A=10, B=0, C=11

序号	日志
1	T ₁ : 开始
2	<u>T₁: 写 A=10</u>
3	T ₂ : 开始
4	T ₂ : 写 B, B=9
5	<u>T₁: 写 C, C=11</u>
6	T ₁ : 提交
7	T ₂ : 写 C, C=13
8	T ₃ : 开始
9	T ₃ : 写 A, A=8
10	T ₂ : 回滚

4、考虑下图所示的日志记录，**初始A、B、C的值都是0**

(3)如果**系统故障**发生在**9**之后，说明哪些事务需要**重做**，哪些事务需要**回滚**。系统恢复后A、B、C的值

需要重做的事务有： T1

需要回滚的事务有： T2、 T3

A=10, B=0, C=11

序号	日志
1	T ₁ : 开始
2	T ₁ : 写 A=10
3	T ₂ : 开始
4	T ₂ : 写 B, B=9
5	T ₁ : 写 C, C=11
6	T ₁ : 提交
7	T ₂ : 写 C, C=13
8	T ₃ : 开始
9	T ₃ : 写 A, A=8

4、考虑下图所示的日志记录，**初始A、B、C的值都是0**

(4)如果**系统故障**发生在**7**之后，说明哪些事务需要**重做**，哪些事务需要**回滚**。系统恢复后A、B、C的值

需要重做的事务有：

T1

需要回滚的事务有：

A=10, B=0, C=11

序号	日志
1	T ₁ : 开始
2	T ₁ : 写 A=10
3	T ₂ : 开始
4	T ₂ : 写 B, B=9
5	T ₁ : 写 C, C=11
6	T ₁ : 提交
7	T ₂ : 写 C, C=13