



事务的基本概念

事务的基本概念

总述

事务是数据库的逻辑工作单位，是数据库故障恢复和并发控制的基本单位。本知识点学习事务的基本概念。

事务的基本概念

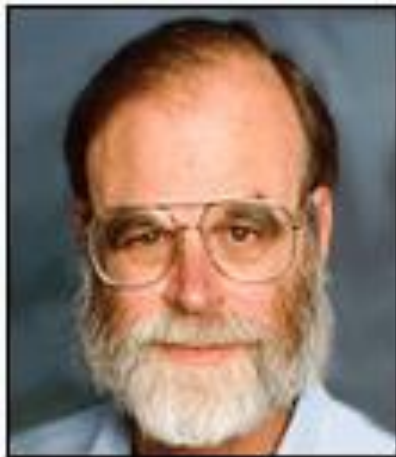
- 一、问题的提出
- 二、事务的概念
- 三、事务的提交
- 四、事务的回滚

一、问题的提出

James Gray

1998年获得图灵奖

在数据库和事务处理研究领域开创性的贡献，以及其在系统实现方面的领导地位。



一、问题的提出

- 1. 运行因故障而终止
- 2. 并发

T1:	T2 :
read(A);	
A=A-100;	
write(A);	
	read(B);
read(B);	
B=B+100;	
	B=B+100;
write(B);	
	write(B);

二、事务的概念

事务 (Transaction) 是用户定义的一个数据库操作序列。这些操作要么都做，要么都不做，是一个不可分割的工作单位。

例如：银行转帐事务 T，从帐户 A 转 1 0 0 元到帐户 B。

```
T:  
  read(A);  
  A=A-100;  
  write(A);  
  read(B);  
  B=B+100;  
  write(B);
```

二、事务的概念

在SQL中，事务通常以**BEGIN TRANSACTION**开始，以COMMIT或ROLLBACK操作结束。

COMMIT即提交，表示**事务成功地结束**。

ROLLBACK即回滚，表示**事务不成功地结束**。

当用户没有显式地定义事务时，DBMS按缺省规定自动划分事务。

```
T:
BEGIN TRANSACTION
read(A);
A=A-100;
write(A);
ROLLBACK;
read(B);
B=B+100;
write(B);
COMMIT
```

三、事务的提交

执行COMMIT操作，即提交事务中所有的操作，完成对数据库的更新，数据库进入一个新的正确状态。

```
T:
BEGIN TRANSCION
read(A);
A=A-100;
write(A);
ROLLBACK;
read(B);
B=B+100;
write(B);
COMMIT
```


三、事务的提交

提交有三种方式：

- (1) **显式**COMMIT --用户发一个SQL指令
COMMIT ;
- (2) **隐式**COMMIT --某些SQL命令执行后自动执行COMMIT命令；
(Create,Alter,Drop,Grant,Revoke)
- (3) **自动**COMMIT --SQL中有一个自动提交开关
AUTO COMMIT , 如果开, 所有数据操作命令,
Insert,Updata,Delete的执行立即提交。

四、事务的回滚

回滚(ROLLBACK)操作告诉系统，已发生错误，数据库可能处在不正确的状态，该事务对数据库已做的所有更新操作必须被撤销，数据库应恢复该事务到初始状态，即回退到事务开始时的状态。

```
T:
  read(A);
  A=A-100;
  write(A);
  ROLLBACK;
  read(B);
  B=B+100;
  write(B);
  COMMIT
```

练习

数据库应用程序的基本逻辑单元是（ ）。

- A. SQL语句
- B. 指令
- C. 进程
- D. 事务

解答

数据库应用程序的基本逻辑单元是（ **D** ）。

- A. SQL语句
- B. 指令
- C. 进程
- D. 事务**

小结

事务是数据库的逻辑工作单位，是数据库故障恢复和并发控制的基本单位。

在关系数据库中，一个事务可以是一条SQL语句、多条SQL语句或整个程序。一个程序通常包含多个事务。



谢谢！



事务的性质

总述

事务是数据库的逻辑工作单位，是数据库故障恢复和并发控制的基本单位。本知识点学习事务的ACID性质。

事务的性质

- 一、原子性
- 二、一致性
- 三、永久性
- 四、隔离性

事务的性质

一、原子性

一个事务中所有对数据库的操作是一个不可分割的操作序列。

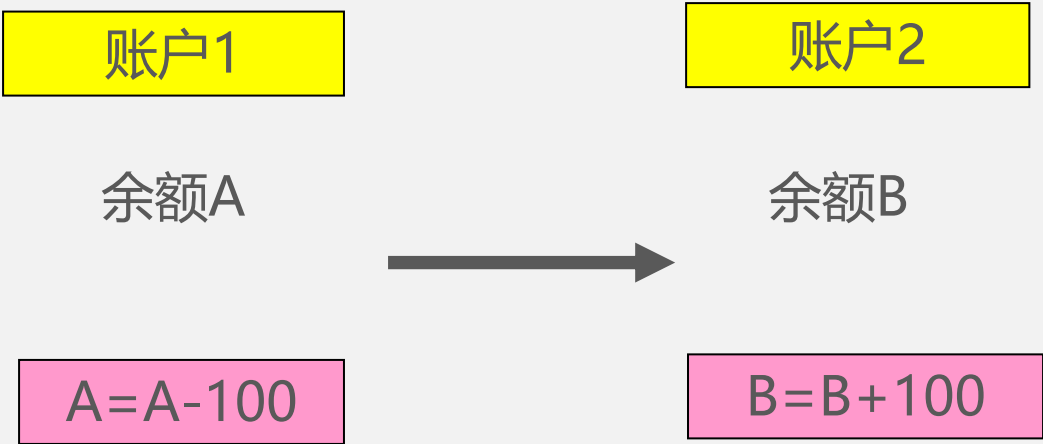
保证原子性是数据库管理系统的职责，原子性是由DBMS的事务管理子系统完成的。

```
T:  
  read(A);  
  A=A-100;  
  write(A);  
  ROLLBACK;  
  read(B);  
  B=B+100;  
  write(B);  
  COMMIT
```

二、一致性

一个事务独立执行的结果将保证数据库的一致性，即数据不会因事务的执行而遭受破坏。

一致性由DBMS的完整性子系统执行测试。



三、永久性

永久性也称为持久性，指对数据库的全部操作提交

(COMMIT) 后，事务对数据库的所有更新应永久地反映在数据库中。

永久性是由DBMS的恢复管理子系统实现的。

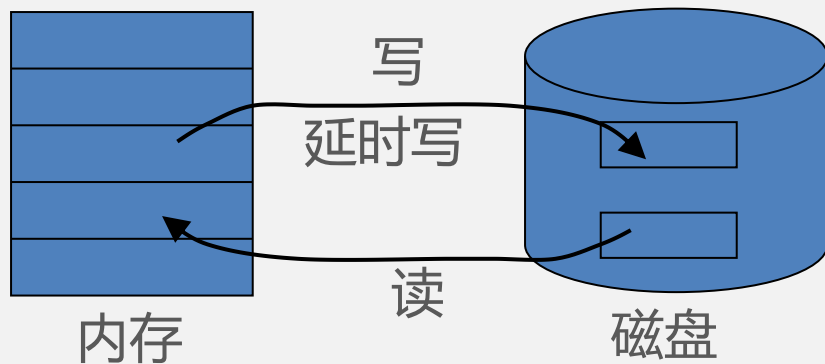
T:

```
read(A);  
A=A-100;  
write(A);  
ROLLBACK;  
read(B);  
B=B+100;  
write(B);  
COMMIT
```

三、永久性

SQL为维护数据的一致性，为每个用户设置了一个临时工作区。所有增删改操作都在这个工作区中进行，而不影响真实数据库。

通过提交，将工作区中的数据写入真实数据库，使数据库操作永久化。



四、隔离性

隔离性要求在并发事务被执行时，系统应保证与这些事务先后单独执行时结果一样，使事务如同在单用户环境下执行一样。

隔离性是由DBMS的并发控制子系统实现的。

四个性质合称为事务的ACID性质。

练习1

下面的概念中不属于事务具有的ACID性质的是（ ）。

- A. 一致性
- B. 隔离性
- C. 原子性
- D. 完整性

事务的ACID性质的是原子性、一致性、隔离性和持久性。

解答

下面的概念中不属于事务具有的ACID性质的是（ D ）。

- A. 一致性
- B. 隔离性
- C. 原子性
- D. 完整性

练习2

下面是银行转账系统的关系模式集，请找出哪些操作应该作为事务处理？

客户 (客户号, 姓名, 年龄, 性别, 身份证号)

银行卡 (银行卡号, 余额, 支付密码, 客户号)

转账记录 (汇款人卡号, 收款人卡号, 金额, 交易时间, 业务摘要)

解答

下面是银行转账系统的关系模式集，请找出哪些操作应该作为事务处理？

客户（客户号，姓名，年龄，性别，身份证号）

银行卡（银行卡号，余额，支付密码，客户号）

转账记录（汇款人卡号，收款人卡号，金额，交易时间，业务摘要）

小结

事务是数据库的逻辑工作单位。如果数据库只包含成功事务提交的结果，就说数据库处于一致性状态。保证数据一致性是对数据库最基本的要求。

DBMS只要能保证系统中一切事务的原子性、一致性、隔离性和持久性（ACID性质），就能保证数据库处于一致性状态。因此保证事务ACID性质是事务管理的重要任务。



谢谢！



故障的种类

故障的种类

总述

数据库管理系统必须具有对故障造成的数据破坏进行恢复的功能。本知识点学习数据库系统中可能发生的故障类型。

故障的种类

- 一、事务故障
- 二、系统故障
- 三、介质故障
- 四、计算机病毒

一、事务故障

事务故障分为两类：可以预期的事务的故障和非预期的事务的故障。

(1) 可以预期的事务的故障

即应用程序可以发现的事务故障，并且可以让事务回退（**ROLLBACK**），撤销错误的事务故障，恢复数据库到正确状态。

例如：下面的银行转帐事务。这个事务是把一笔金额从一个帐户甲转给另一个帐户乙。

一、事务故障

BEGIN TRANSACTION

读帐户甲的余额Balance;

Balance = Balance - Amount; (Amount为转帐金额)

IF (Balance < 0) THEN

{ 打印 '金额不足, 不能转帐' ;

ROLLBACK; } (撤消刚才的修改, 恢复事务)

ELSE { 读帐户乙的余额Balance 1;

Balance 1 = Balance 1 + Amount;

写回Balance 1;

COMMIT; }

一、事务故障

再如，用触发器实现只有S表中存在的学生才能在SC表中添加相应的选课记录。

```
CREATE TRIGGER insert_sc1 on SC FOR INSERT
AS
    IF (NOT EXISTS (SELECT S.sno
                     FROM S,Inserted
                     WHERE S.sno=Inserted.sno))
BEGIN    PRINT( '该生不存在 ')
        ROLLBACK  --回滚事务，撤消插入记录
END
```

一、事务故障

(2) 非预期的事务的故障

即不能由事务程序处理的，如运算溢出、并发事务发生死锁而被选中并被撤销、违反了完整性规则而被终止等。**事务故障一般指这一类故障。**

事务故障意味着事务没有到达预期的终点，因此，数据库可能处于不正确状态。系统就要强行滚回此事务，即撤消该事务已经对数据库所做的任何修改，使得该事务好象根本没有启动一样。

这类恢复操作称为**事务撤消（UNDO）**。

二、系统故障

系统故障常称为**软故障**（Soft Crash）。

系统故障是指造成系统停止运转的任何事件，使得系统要重新启动。例如中央处理器故障、操作系统故障、突然停电等等。这类故障影响正在运行的所有事务，但不破坏数据库。

二、系统故障

恢复子系统必须在系统重新启动时：

- 让所有非正常终止的事务滚回，强行撤消（**UNDO**）所有未完成事务。
- 需要重做（**REDO**）所有已提交的事务，把数据库恢复到正确的状态。

```
T:read(A);  
A=A-100;  
write(A);  
ROLLBACK;  
read(B);  
B=B+100;  
write(B);  
COMMIT
```

三、介质故障

介质故障称为**硬故障** (Hard Crash) 。

硬故障指外存故障，如磁盘的磁头碰撞，瞬时的强磁场干扰。这类故障将破坏数据库或部分数据库，并影响正存取这部分数据的所有事务，这类故障比前两类故障发生的可能性小得多，但破坏性最大。

可以通过DBMS把其他备份磁盘中的内容再复制回来。

四、计算机病毒

计算机病毒是一种人为的故障或破坏，是一种计算机程序。病毒种类很多，特征不同，传播速度很快。一旦入侵，破坏系统。

计算机病毒是计算机系统的主要威胁，自然也是数据库系统的主要威胁。数据库一旦破坏，需要用恢复技术加以恢复。

练习

若系统在运行过程中由于某种硬件故障，使存储在外存上的数据部分损失，这种情况称为（ ）。

- A. 事务故障
- B. 系统故障
- C. 运行故障
- D. 介质故障

解答

若系统在运行过程中由于某种硬件故障，使存储在外存上的数据部分损失，这种情况称为（D）。

- A. 事务故障
- B. 系统故障
- C. 运行故障
- D. 介质故障

小结

计算机系统中硬件的故障、软件的错误、操作员的失误以及故意的破坏等故障可能造成数据库中数据的丢失。

总结各类故障，对数据库的影响有两种可能性：

- (1) 是数据库本身被破坏；
- (2) 是数据库没有破坏，但数据可能不正确。

数据库管理系统必须具有把数据库从错误状态恢复到最近一个正确状态的功能，就是数据库的恢复。不同故障需要不同的恢复策略和恢复方法。



谢谢！



数据转储

总述

数据库管理系统必须具有对故障造成的数据破坏进行恢复的功能。数据库恢复的关键是建立冗余数据，数据转储是最常用的建立冗余数据的技术。本知识点学习数据转储的基本概念和方法。

- 一、恢复机制
- 二、静态转储和动态转储
- 三、海量转储和增量转储

一、恢复机制

各类数据库故障可能破坏数据库本身，或者造成数据的不正确。

数据库的恢复是指当数据库系统遭到破坏时，通过一些技术，使数据库恢复到遭到破坏前的正确状态。

恢复的基本原则很简单，就是“冗余”，即数据的重复存储。数据库中任何一部分的数据可以根据存储在系统别处的冗余数据来重建。

一、恢复机制

恢复机制涉及的两个关键问题：

- 如何建立冗余数据
- 如何利用这些冗余数据实施数据库恢复

建立冗余数据最常用的方法是：

- 数据转储 (backup) 和登记日志文件 (logging)

一、恢复机制

数据转储指数据库管理员定期将数据库复制到磁带、磁盘或其它存储介质上保存起来的过程，这些备用的数据称为**后备副本**或后援副本。

数据库遭到破坏后可以将后备副本重新装入，将数据库恢复到转储时的状态。

数据转储方法有静态转储、动态转储、海量转储和增量转储等。

二、静态转储和动态转储

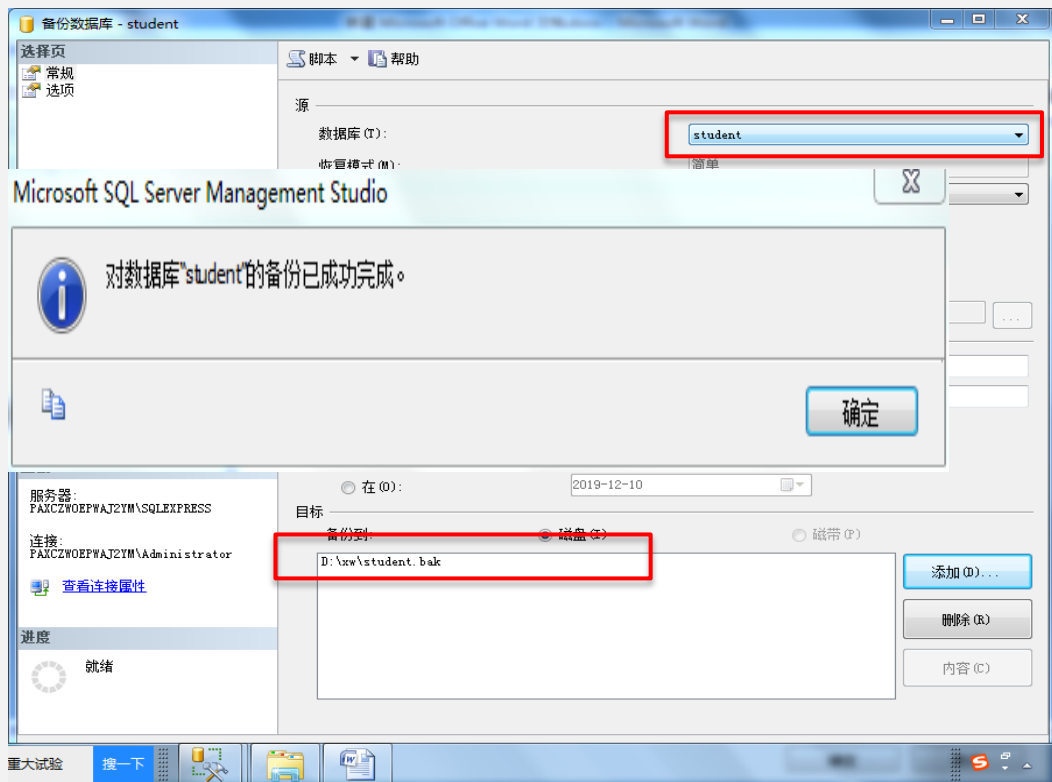
静态转储：系统中无运行事务时转储。

动态转储：用户事务与转储并发执行，允许在转储期间对数据库进行存取和修改。

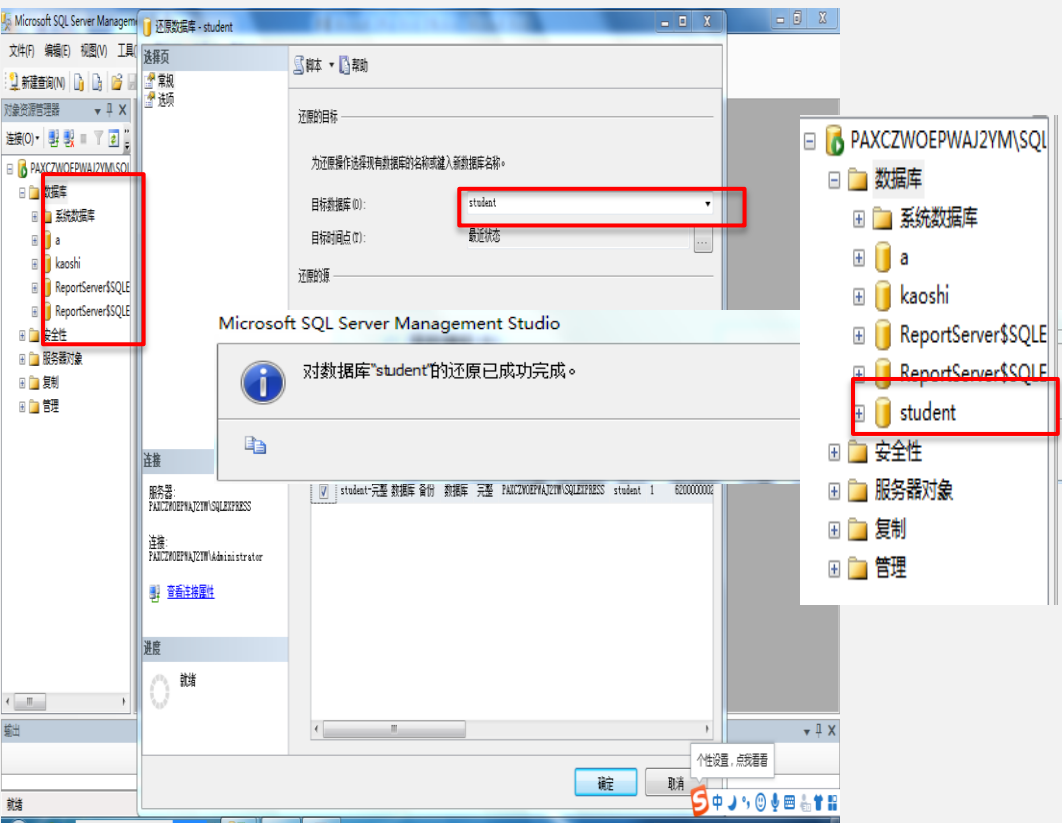
静态转储实现简单，但降低了数据库的可用性。

动态转储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，但不能保证副本中的数据正确有效。

二、静态转储和动态转储



二、静态转储和动态转储

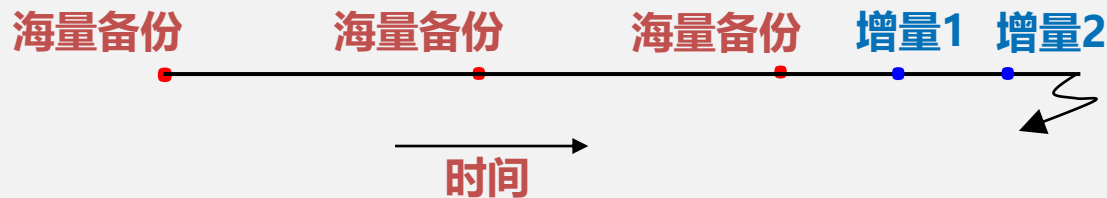


三、海量转储和增量转储

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。

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；但如果数据库很大，事务处理又十分频繁，则增量转储方式更实用更有效。



练习

后援副本的用途是（ ）。

- A. 安全性保障
- B. 一致性控制
- C. 故障后的恢复
- D. 数据的转储

练习

后援副本的用途是（ C ）。

- A. 安全性保障
- B. 一致性控制
- C. 故障后的恢复
- D. 数据的转储

小结

数据库管理系统必须具有把数据库从错误状态恢复到最近一个正确状态的功能，就是数据库的恢复。

恢复的基本原理就是建立冗余数据。建立冗余数据最常用的方法之一是数据转储。



谢谢！



登记日志文件

总述

数据库恢复的关键是建立冗余数据，建立冗余数据的最常用的技术是数据转储和登记日志文件。本知识点学习登记日志文件的基本方法。

- 一、恢复机制
- 二、日志文件的内容
- 三、日志文件的作用
- 四、登记日志文件的原则

一、恢复机制

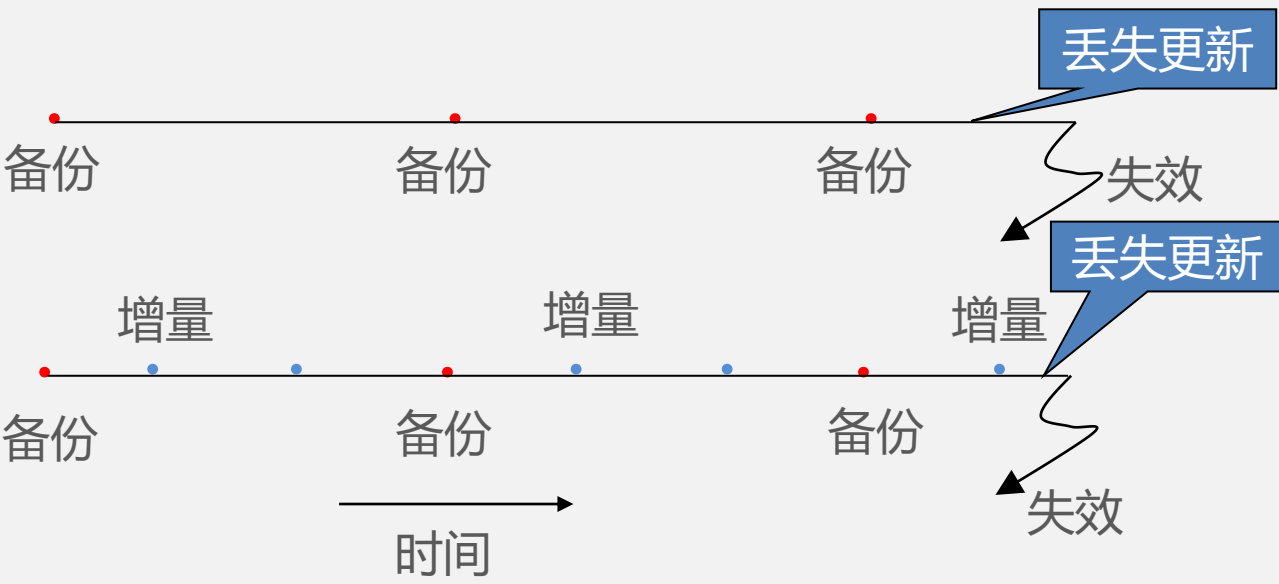
数据库的恢复是指当数据库系统遭到破坏时，通过一些技术，使数据库恢复到遭到破坏前的正确状态。

恢复的基本原则很简单，就是“冗余”，即数据的重复存储。

建立冗余数据最常用的方法是数据转储 (backup) 和登记日志文件 (logging)。

一、恢复机制

必须把转储期间各事务对数据库的修改活动登记下来，建立日志文件，这样，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。



二、日志文件的内容

日志文件是用来记录事务对数据库的更新操作的文件，不同数据库系统采用的日志文件格式并不相同。

日志文件主要有两种格式，即以**记录为单位**的日志文件和以**数据块为单位**的日志文件。

二、日志文件的内容

(1) 以记录为单位

日志记录的内容主要包括:

- 事务标识
- 操作类型（增删改）
- 操作对象
- 数据的旧值
- 数据的新值

有了数据的旧值，可以撤销更新（UNDO操作）；
有了数据的新值，可以重做更新（REDO操作）。

二、日志文件的内容

(2) 以数据块为单位

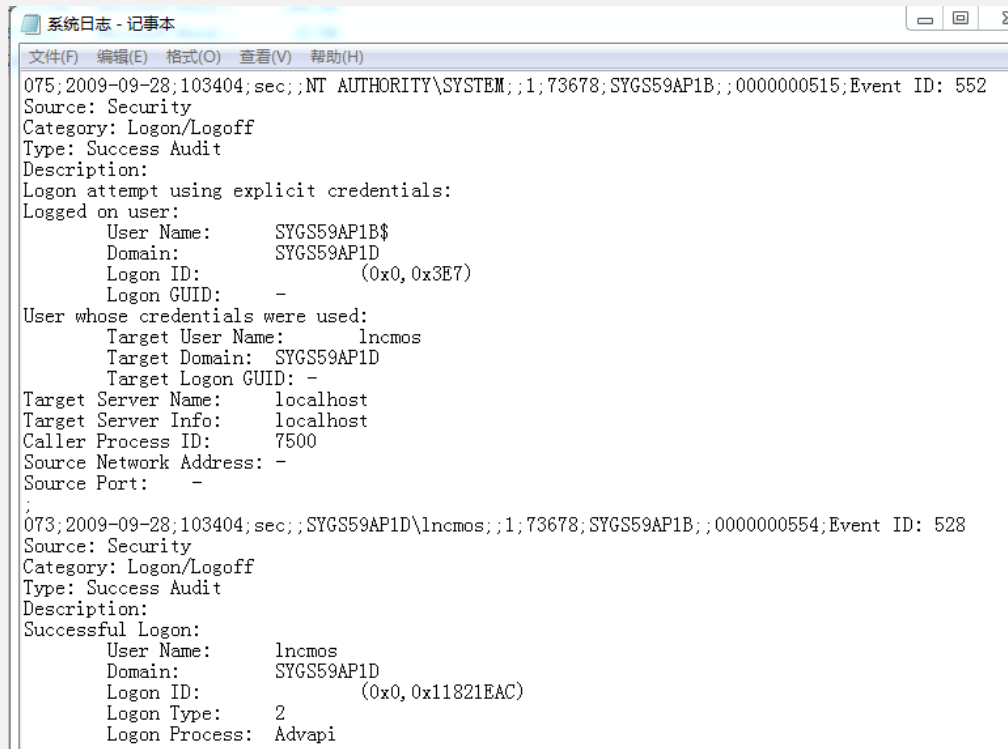
日志记录的内容包括：

- 事务标识
- 事务更新前的物理块映像（前像）
- 事务更新后的物理块映像（后像）

有了前像，可以撤销更新（UNDO操作）；

有了后像，可以重做更新（REDO操作）。

二、日志文件的内容



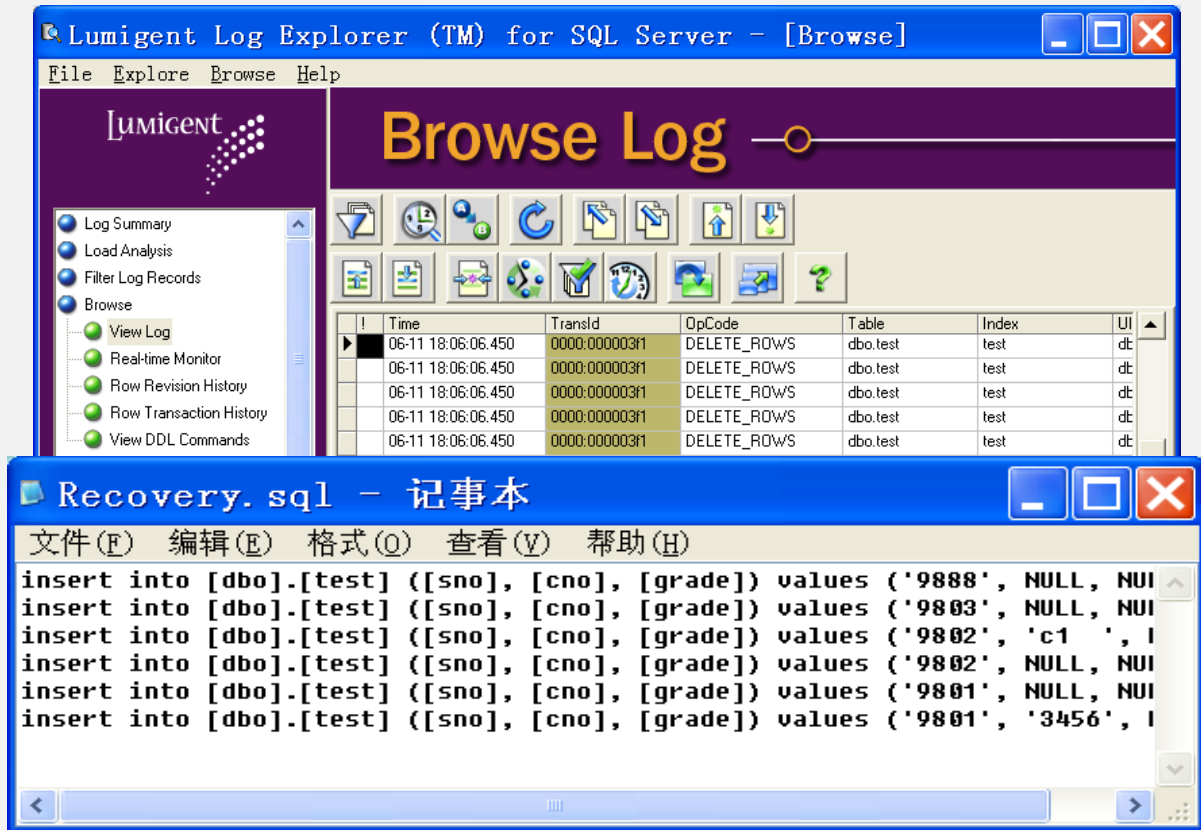
系统日志 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

075;2009-09-28;103404;sec;;NT AUTHORITY\SYSTEM;;1;73678;SYGS59AP1B;;0000000515;Event ID: 552
Source: Security
Category: Logon/Logoff
Type: Success Audit
Description:
Logon attempt using explicit credentials:
Logged on user:
 User Name: SYGS59AP1B\$
 Domain: SYGS59AP1D
 Logon ID: (0x0, 0x3E7)
 Logon GUID: -
User whose credentials were used:
 Target User Name: lncmos
 Target Domain: SYGS59AP1D
 Target Logon GUID: -
Target Server Name: localhost
Target Server Info: localhost
Caller Process ID: 7500
Source Network Address: -
Source Port: -
;

073;2009-09-28;103404;sec;;SYGS59AP1D\lncmos;;1;73678;SYGS59AP1B;;0000000554;Event ID: 528
Source: Security
Category: Logon/Logoff
Type: Success Audit
Description:
Successful Logon:
 User Name: lncmos
 Domain: SYGS59AP1D
 Logon ID: (0x0, 0x11821EAC)
 Logon Type: 2
 Logon Process: Advapi

二、日志文件的内容



The screenshot displays two windows. The top window is 'Lumigent Log Explorer (TM) for SQL Server - [Browse]'. It features a sidebar with navigation options: Log Summary, Load Analysis, Filter Log Records, Browse, View Log, Real-time Monitor, Row Revision History, Row Transaction History, and View DDL Commands. The main area is titled 'Browse Log' and contains a table of log entries.

Time	TransId	OpCode	Table	Index	UI
06-11 18:06:06.450	0000:000003f1	DELETE_ROWS	dbo.test	test	dt
06-11 18:06:06.450	0000:000003f1	DELETE_ROWS	dbo.test	test	dt
06-11 18:06:06.450	0000:000003f1	DELETE_ROWS	dbo.test	test	dt
06-11 18:06:06.450	0000:000003f1	DELETE_ROWS	dbo.test	test	dt
06-11 18:06:06.450	0000:000003f1	DELETE_ROWS	dbo.test	test	dt

The bottom window is a Notepad titled 'Recovery.sql - 记事本'. It contains the following SQL commands:

```
insert into [dbo].[test] ([sno], [cno], [grade]) values ('9888', NULL, NUI
insert into [dbo].[test] ([sno], [cno], [grade]) values ('9803', NULL, NUI
insert into [dbo].[test] ([sno], [cno], [grade]) values ('9802', 'c1 ', I
insert into [dbo].[test] ([sno], [cno], [grade]) values ('9802', NULL, NUI
insert into [dbo].[test] ([sno], [cno], [grade]) values ('9801', NULL, NUI
insert into [dbo].[test] ([sno], [cno], [grade]) values ('9801', '3456', I
```

三、日志文件的作用

日志文件的作用：

- (1) 进行事务故障恢复;
- (2) 进行系统故障恢复;
- (3) 协助后备副本进行介质故障恢复。

四、登记日志文件的原则

登记日志文件时必须遵循如下原则：

- (1) 登记次序按并发事务执行的时间次序；
- (2) 先写日志记录，再写数据库（即运行记录优先原则）。

在数据库系统中，写一个修改到数据库中和写一个表示这个修改的登记记录到日志文件中是两个不同的操作。

四、登记日志文件的原则

登记日志文件时必须遵循如下原则：

- (1) 登记次序按并发事务执行的时间次序；
- (2) 先写日志记录，再写数据库（即运行记录优先原则）。

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，则以后可能无法恢复这个修改。

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，并不会影响数据库的正确性。

练习

数据库恢复时利用的冗余数据指()。

- A. 数据字典、应用程序、审计档案、数据库后备副本
- B. 数据字典、应用程序、日志文件、审计档案
- C. 日志文件、数据库后备副本
- D. 数据字典、应用程序、数据库后备副本

解答

数据库恢复时利用的冗余数据指 (C)。

- A. 数据字典、应用程序、审计档案、数据库后备副本
- B. 数据字典、应用程序、日志文件、审计档案
- C. 日志文件、数据库后备副本
- D. 数据字典、应用程序、数据库后备副本

小结

故障恢复的基本原理是利用存储在后备副本和日志文件中的冗余数据来重建数据库。

为实现利用日志恢复数据，必须遵循先写日志原则。

日志文件可以用来进行事务故障恢复和系统故障恢复，并协助后备副本进行介质故障恢复。

不同故障的恢复策略和方法不同。



谢谢！



故障的恢复策略

总述

利用数据库后备副本和日志文件可以将数据库恢复到故障前的某个一致性状态，不同故障的恢复策略不同。本知识点学习各种故障的恢复方法。

故障的恢复策略

- 一、事务故障的恢复
- 二、系统故障的恢复
- 三、介质故障的恢复

一、事务故障的恢复

事务故障的恢复由系统**自动**完成，利用日志文件**撤消**（UNDO）此事务对数据库进行的修改。

(1) **反向扫描**日志文件，查找该事务的更新操作；

(2) 对该事务的更新操作执行逆操作；（将日志中“更新前的值”写入数据库）

(3) 继续反向扫描日志文件，查找该事务的其他更新操作，做同样的处理；

(4) 直到读到此事务的开始标记，故障恢复完成。

二、系统故障的恢复

由系统在重新启动时**自动**完成。要撤销故障发生时未完成的事务（**UNDO**），重做已完成的事务（**REDO**）。

（1）建立事务的重做队列和撤销队列

从头扫描日志文件，找出故障发生前已提交的事务（既有Begin Transaction记录，也有Commit记录），将事务标识记入**重做队列**；同时找出故障发生时尚未完成的事务（只有Begin Transaction记录，没有Commit记录），将事务标识记入**撤销队列**。

二、系统故障的恢复

(2) 对撤销队列的事务进行UNDO处理

反向扫描日志文件，对每个**UNDO**事务，将日志记录中“更新前的值”写入数据库。

(3) 对重做队列的事务进行REDO处理

正向扫描日志文件，对每个**REDO**事务，将日志记录中“更新后的值”写入数据库。

三、介质故障的恢复

磁盘上的物理数据和日志文件被破坏，数据库已破坏，恢复方法则由DBA进行如下工作：

(1) 装入最近备份的数据库，使数据库恢复到最近一次转储时的一致性状态。

(2) 然后装入相应的日志文件，利用“日志文件”执行REDO（重做）操作，重做已完成的事务。

练习

关于事务故障恢复的描述正确的是（ ）。

- A. 必须利用后备副本和日志文件
- B. DBA必须参与
- C. 只需要后备副本
- D. 只需要日志文件

解答

关于事务故障恢复的描述正确的是（ D ）。

- A. 必须利用后备副本和日志文件
- B. DBA必须参与
- C. 只需要后备副本
- D. 只需要日志文件

小结

当系统运行过程中发生故障，利用数据库后备副本和日志文件可以将数据库恢复到故障前的某个一致性状态。尽管恢复的基本原理很简单，但实现技术的细节却很复杂。

对于事务故障，恢复子系统利用日志文件执行撤销（UNDO）操作。对于系统故障，恢复子系统利用日志文件执行撤销（UNDO）和重做（REDO）操作。对于介质故障，恢复方法是重装备份使数据库恢复到一致性状态，再利用日志文件执行REDO操作。



谢谢！



具有检查点的恢复技术

总述

利用日志文件进行数据库恢复时，恢复子系统必须搜索日志，DBMS会采用一些方法来提高恢复的效率。本知识点学习具有检查点的恢复技术。

具有检查点的恢复技术

- 一、问题的提出
- 二、建立检查点
- 三、使用检查点的恢复

一、问题的提出

利用日志技术进行数据库恢复时，恢复子系统必须搜索日志，确定哪些事务需要REDO，哪些事务需要UNDO。

- 一般需要搜索所有日志记录，耗费大量的时间；
- 需要重做处理的事务有些已经将更新结果写入数据库，恢复系统重新执行这些操作，增加了恢复的时间。

大部分DBMS产品都是用了一种检查点技术恢复数据库。

一、问题的提出

DBMS定时设置检查点，在检查点时才真正做到把对DB的修改写到磁盘，并在日志文件中写入一条检查点记录。恢复时只有那些检查点后面的记录需要恢复。

检查点方法涉及两个问题：

- (1) 建立检查点
- (2) 使用检查点进行恢复

二、建立检查点

建立检查点的方法：

- 在日志文件中增加检查点记录
(checkpoint)
- 增加重新开始文件
- 恢复子系统在登录日志文件期间动态地
维护日志

二、建立检查点

检查点记录的内容

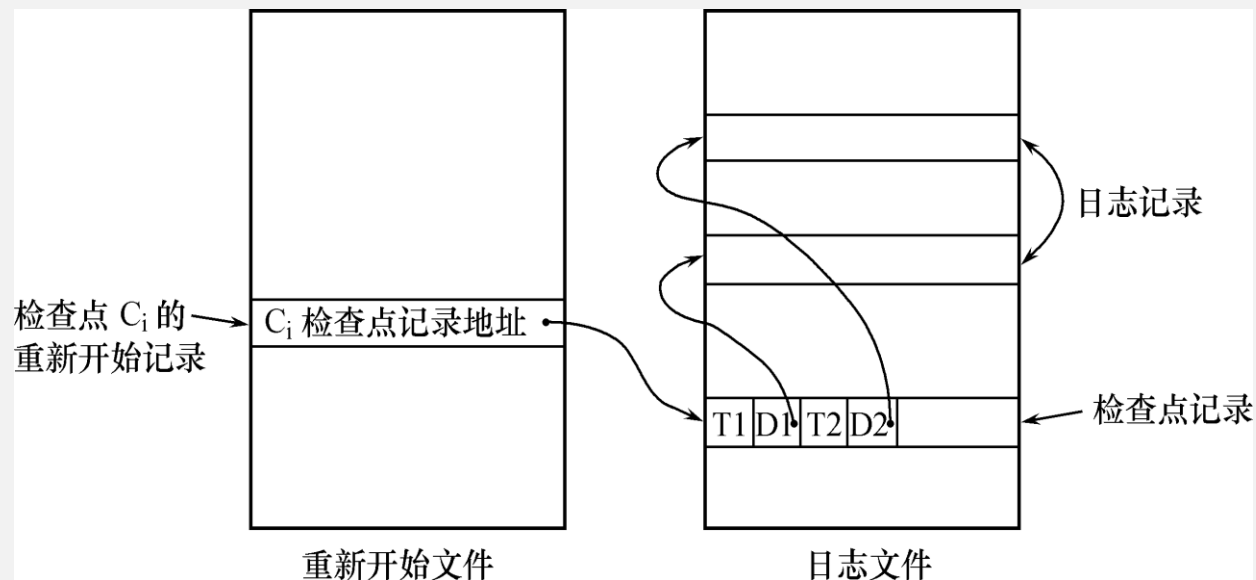
- 建立检查点时刻所有正在执行的事务清单
- 这些事务最近一个日志记录的地址

重新开始文件的内容

- 记录各个检查点记录在日志文件中的地址

二、建立检查点

具有检查点的日志文件和重新开始文件



二、建立检查点

动态维护日志的过程：

- 将当前日志缓冲区中的所有日志记录写入磁盘的日志文件上
- 在日志文件中写入一个检查点记录
- 将当前数据缓冲区的所有数据记录写入磁盘的数据库中
- 把检查点记录在日志文件中的地址写入一个重新开始文件

二、建立检查点

建立检查点的周期：

- 定期

按照预定的一个时间间隔，如每隔一小时建立一个检查点

- 不定期

按照某种规则，如日志文件已写满一半建立一个检查点

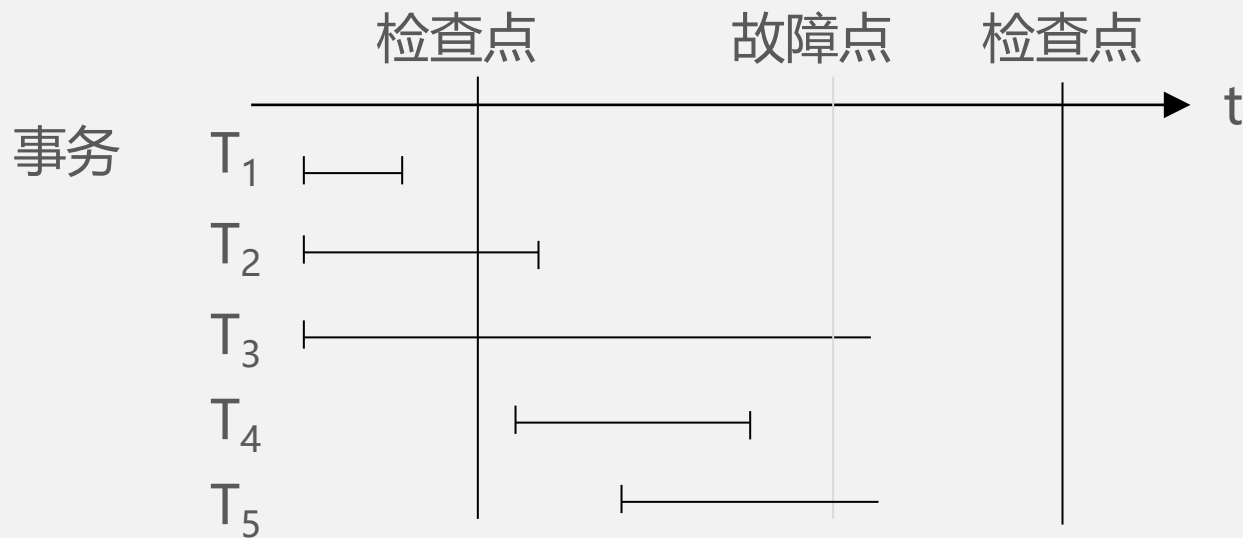
三、使用检查点的恢复

- 从重新开始文件中找到最后一个检查点记录在日志文件中的地址，由该地址在日志文件中**找到最后一个检查点记录**。
- 由该检查点记录得到检查点建立时刻所有正在执行的事务清单ACTIVE-LIST。
 - 建立两个事务队列（UNDO-LIST和REDO-LIST）。
 - 把ACTIVE-LIST暂时放入UNDO-LIST队列，REDO队列暂为空。

三、使用检查点的恢复

- 从检查点开始**正向扫描**日志文件，直到日志文件结束
 - 如有新开始的事务 T_i ，把 T_i 暂时放入UNDO-LIST队列
 - 如有提交的事务 T_j ，把 T_j 从UNDO-LIST队列移到REDO-LIST队列
- 对UNDO-LIST中的每个事务执行**UNDO**操作。
- 对REDO-LIST中的每个事务执行**REDO**操作。

三、使用检查点的恢复



事务 T_1 : 不必恢复

事务 T_2 : 必须REDO

事务 T_3 : 必须UNDO

事务 T_4 : 必须REDO

事务 T_5 : 必须UNDO

练习

关于检查点，下列说法不正确的是（ ）。

- A. 设置检查点记录是为了提高故障时的恢复效率
- B. 可以定期地建立检查点，也可以按照某种规则建立检查点
- C. 检查点最大限度减少数据库恢复时必须执行的日志部分
- D. 无论事务在检查点之前还是之后提交，都需要执行REDO操作

解答

关于检查点，下列说法不正确的是（ D ）。

- A. 设置检查点记录是为了提高故障时的恢复效率
- B. 可以定期地建立检查点，也可以按照某种规则建立检查点
- C. 检查点最大限度减少数据库恢复时必须执行的日志部分
- D. 无论事务在检查点之前还是之后提交，都需要执行REDO操作

小结

DBMS定时设置检查点，在检查点时才真正做到把对DB的修改写到磁盘，并在日志文件中写入一条检查点记录。恢复时只有那些检查点后面的记录需要恢复。

使用检查点方法可以改善恢复效率。



谢谢！



数据库镜像

总述

为了改善介质故障恢复效率，DBMS提供了数据库镜像功能以提高数据库的可用性。本知识点了解数据库镜像方法以及数据容灾的概念。

- 一、镜像磁盘技术
- 二、数据容灾

一、镜像磁盘技术

事务故障、系统故障和介质故障哪个对系统影响最严重？

介质故障是对系统影响最为严重的一种故障，严重影响数据库的可用性。

为预防介质故障，DBA必须周期性地转储数据库，介质故障恢复时间也比较长。

如何提高数据库可用性？

一、镜像磁盘技术

提高数据库可用性的解决方案

- **数据库镜像** (Mirror)

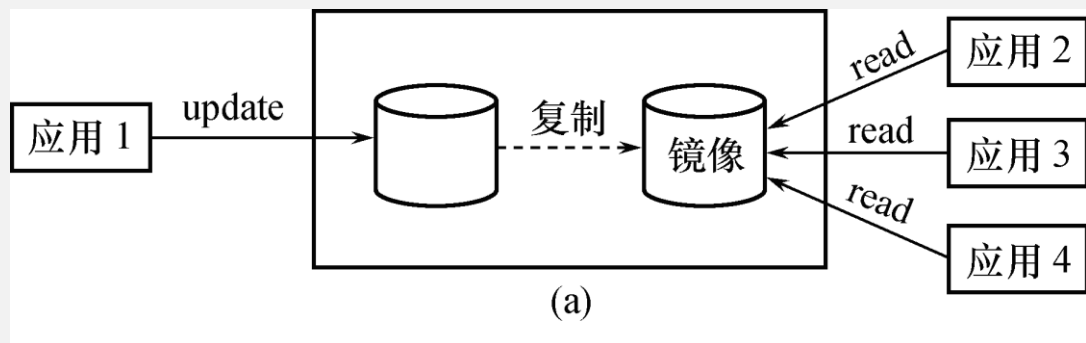
DBMS自动把整个数据库或其中的关键数据复制到另一个磁盘上。

每当主数据库更新时，DBMS自动把更新后的数据复制过去。

DBMS**自动**保证镜像数据与主数据库的一致性。

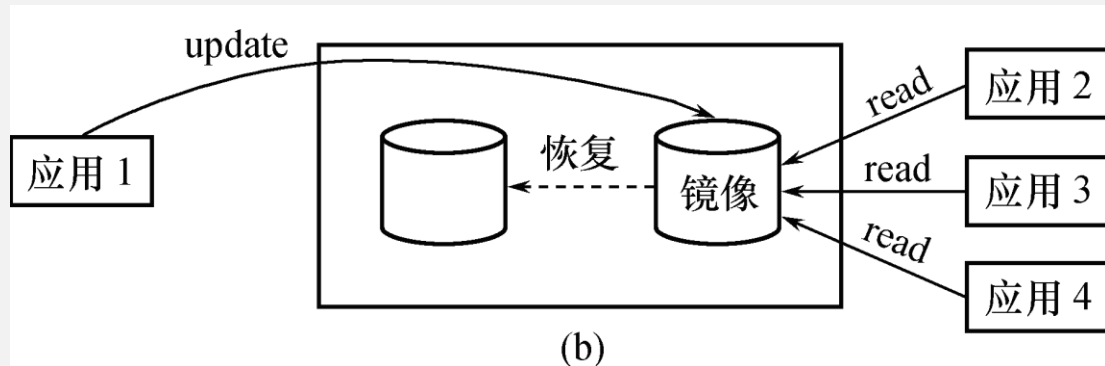
一、镜像磁盘技术

没有出现故障时，可用于并发操作，一个用户对数据加排他锁修改数据，其他用户可以读镜像数据库上的数据，而不必等待该用户释放锁。



一、镜像磁盘技术

出现介质故障时，可由镜像磁盘继续提供使用，同时DBMS自动利用镜像磁盘数据进行数据库的恢复，不需要关闭系统和重装数据库副本。



二、数据容灾

传统的备份主要是采用磁带机进行冷备份，备份磁带同时也在机房中统一管理，一旦整个机房出现了灾难，如火灾和地震等灾难，这些备份磁带也随之销毁。

数据容灾能在灾难发生时恢复整个系统。国际标准SHARE 78 定义的容灾系统有七个层次：从最简单的仅在本地进行磁带备份，到将备份的磁带存储在异地，再到建立应用系统实时切换的异地备份系统，恢复时间也可以从几天到小时级到分钟级、秒级或0数据丢失等。

无论哪种容灾方案，数据备份还是最基础的。

练习

关于数据库镜像，下列描述不正确的是（ ）。

- A. 数据库镜像是DBMS自动把整个数据库或其中的关键数据复制到另一个磁盘上
- B. 磁盘镜像的磁盘利用率是50%
- C. 当主数据库故障时可由镜像磁盘自动提供使用，DBMS自动利用镜像磁盘数据进行恢复
- D. 利用镜像磁盘数据进行恢复时需要关闭系统

解答

关于数据库镜像，下列描述不正确的是（ D ）。

- A. 数据库镜像是DBMS自动把整个数据库或其中的关键数据复制到另一个磁盘上
- B. 磁盘镜像的磁盘利用率是50%
- C. 当主数据库故障时可由镜像磁盘自动提供使用，DBMS自动利用镜像磁盘数据进行恢复
- D. 利用镜像磁盘数据进行恢复时需要关闭系统

小结

随着技术的发展，磁盘容量越来越大，价格越来越便宜。为避免介质故障影响数据库可用性，一些DBMS提供了数据库镜像功能用于数据库恢复。

频繁地复制数据会降低系统运行效率，在实际应用中往往只选择对关键数据和日志文件镜像，而不是对整个数据库进行镜像。



谢谢！