**复习主要内容**

# 题型：选择题30个共30分，判断题10个共10分、简答题2个共10分，程序设计题（SQL及关系代数）30分，综合题（数据库规范化，画出E-R图，找出依赖关系，指出符合第几范式，规范化到第三范式）20分。

**章节主要复习内容：**

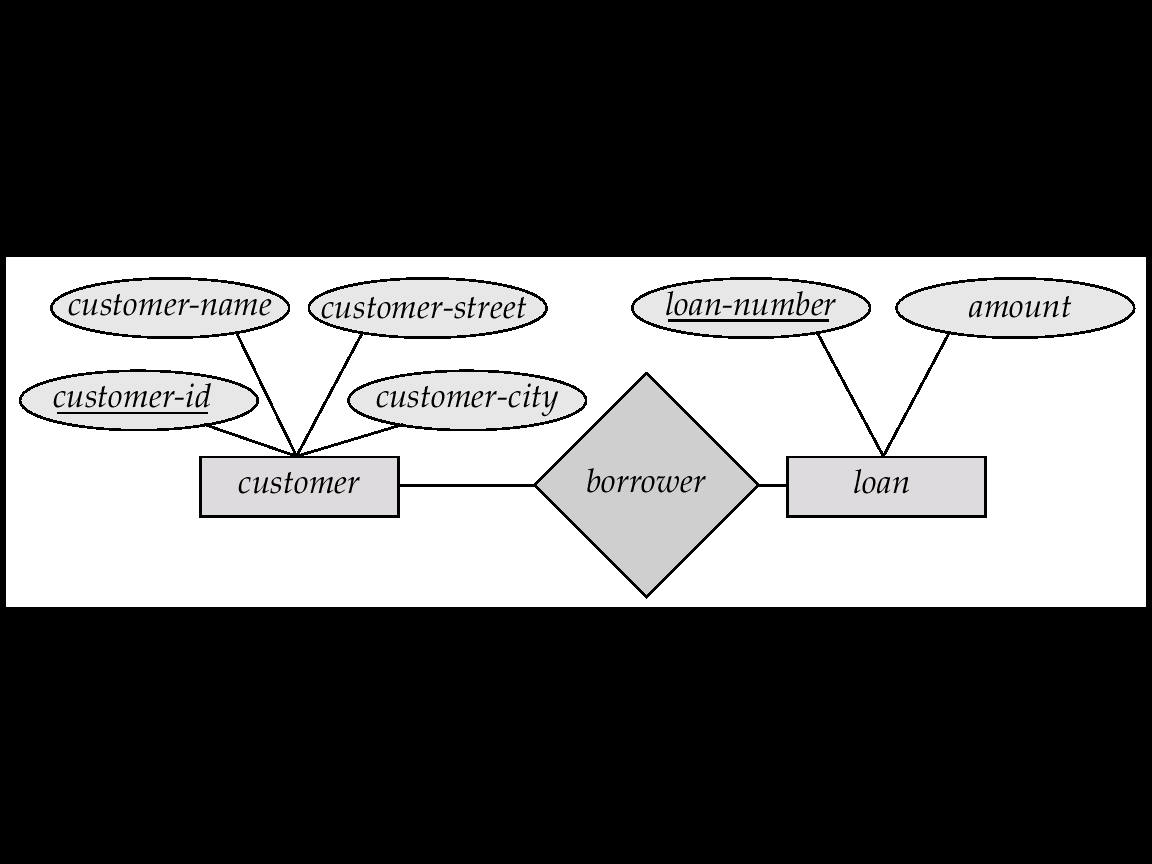
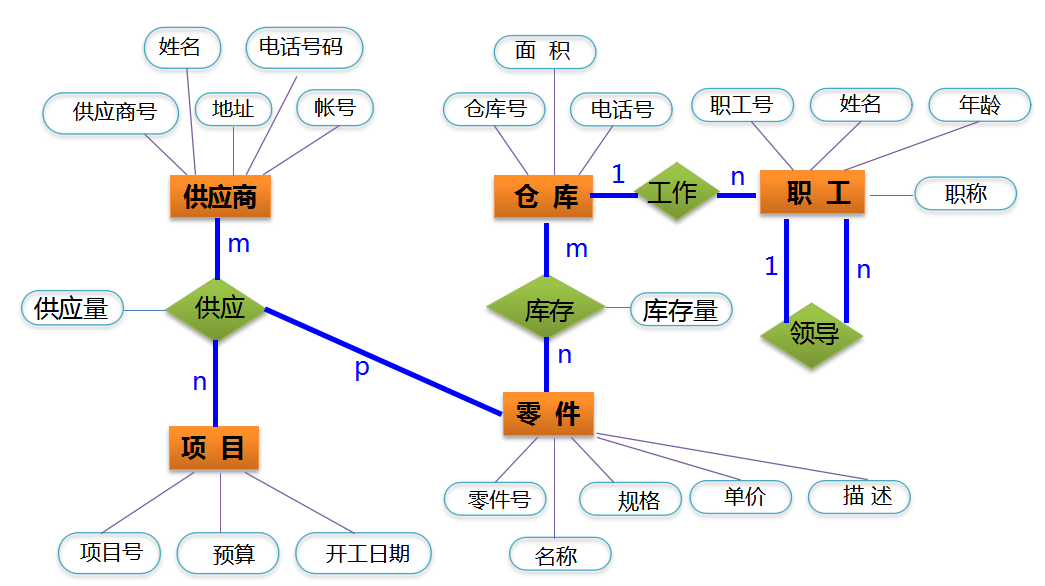
# 概述

1. 理解E-R图图示的表示方法

**用矩形表示实体型，矩形框内写明实体名；**

**用椭圆表示实体的属性，并用无向边将其与相应的实体型连接起来；**

**用菱形表示实体型之间的联系，在菱形框内写明联系名，并用无向边分别与有关实体型连接起来，同时在无向边旁标上联系的类型（1:1,1:n或m:n）**

1. 实体之间的联系形式主要有哪几种？

**实体之间的联系通常是指不同实体集之间的联系。**

**实体之间的联系有一对一、一对多和多对多等多种类型。**

**两个，两个以上，单个**

1. 理解概念模型、逻辑模型、物理模型

**数据模型是数据库系统的核心和基础。各种机器上实现的数据库管理系统软件都是基于某种数据模型或者说是支持某种数据模型的。（两大类，第一类，概念模型。第二类，逻辑模型和物理模型）**

**概念模型也叫信息模型，它是按用户的观点来对数据和信息建模，主要用于数据库设计。**

**逻辑模型主要包括层次模型、网状模型、关系模型、面向对象数据模型和对象关系数据模型、半结构化模型等。它是按计算机系统的观点对数据建模，主要用于数据库管理系统的实现。**

**物理模型是对数据最底层的抽象，它描述数据在系统内部的表示方式和存取方法，或在磁盘或磁带上的存储方式和存取方法，是面向计算机系统的。物理模型的具体实现是数据库管理系统的任务，数据库设计人员要了解和选择物理模型，最终用户则不必考虑物理级的细节。**

1. 什么叫物理独立性？什么叫逻辑独立性？

**物理独立性是指用户的应用程序与数据库中数据的物理存储是相互独立的。**

**逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的。**

1. 数据库管理系统的主要管理功能？数据定义功能（DDL、DML、数据库的事务和运行管理、数据库的建立和维护功能）

**数据定义功能，数据组织、存储和管理，数据操纵功能，数据库的事务管理和运行管理，数据库的建立和维护功能，其它功能（通信功能，数据转换功能，异构数据库的互访互操作功能）**

# 关系代数

1. 熟练掌握选择、投影、并、交、差运算

**课本P49**

1. 投影和选择运算要重点掌握，选择后，属性数量不发生变化。
2. 理解候选键、主键、外键的概念及关系；关系表有多个候选键，但只能选定其中一个做为主键

**某一属性组的值能唯一地标识一个元组，而其子集不能，则称该属性组为候选码**

**若一个关系有多个候选码，则选定其中一个为主码。**

**设F是基本关系R的一个或一组属性，但不是关系R的码，Ks是基本关系S的主码。如果F与Ks相对应，则称F是R的外码，并称基本关系R为参照关系，基本关系S为被参照关系或目标关系，关系R和关系S不一定是不同的关系。**

1. 理解自然连接、外连接（左、右、全外连接）

**课本P55**

1. 熟悉几种完整性约束，

**参照完整性违约处理包括拒绝执行（受限）、级联操作和设置为空值三种**

**关系模型中有三类完整系约束：实体完整性、参照完整性和用户定义的完整性。实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性。**

**实体完整性规则 若属性（一个或一组属性）A是基本关系R的主属性，则A不能取空值。所谓空值就是“不知道”“不存在”或“无意义”的值。**

**参照完整性规则 若属性（或属性组）F是基本关系R的外码，它与基本关系S的主码Ks相对应（基本关系R和S不一定是不同的关系），则对于R中每个元组在F上的值必须：**

**或者取空值（F的每个属性值均为空值）；**

**或者等于S中某个元组的主码值。**

# SQL

SQL这章所占分数最多

1. 要掌握单表查询、多表连接查询、嵌套查询、分组运算、排序、选择、投影等操作
2. 视图的作用？如何定义视图？

**视图能够简化用户的操作**

**视图使用户能以多种角度看待同一数据**

**视图对重构数据库提供了一定程度的逻辑独立性**

**视图能够对机密数据提供安全保护**

**适当利用视图可以更清晰地表达查询**

1. 如何去掉查询结果的重复元组（记录）

**DISTINCT**

1. HAVING和where语句的用途，HAVING作用于分组，对分组进行限制输出，Where作用于整个表，用来选择满足条件的行
2. ALTER TABLE用来修改表的结构（数据库模式）,DROP TABLE删除表结构和数据，DELETE TABLE删除表中数据
3. 判断空值用“IS NULL”
4. 数据系统中，对存取权限的定义称为授权，授权和收回授权如何定义？

**P141 ALL PRIVILEGES**

**GRANT <权限>[,<权限>]**

**ON<对象类型><对象名>[,<对象类型><对象名>]**

**TO<用户>[,<用户>]**

**[WITH GRANT OPTION];**

**如：授予U1查询SC的权限并允许它把该权限赋予给其他用户**

**GRANT SELECT**

**ON TABLE SC**

**TO U1**

**WITH GRANT OPTION**

**REVOKE<权限>[,<权限>]**

**ON<对象类型><对象名>[,<对象类型><对象名>]**

**FROM<用户>[,<用户>][CASCADE|RESTRICT];**

**如：把用户U4修改学生学号的权限收回**

**REVOKE UPDATE(Sno)**

**ON TABLE Student**

**FROM U4;**

1. 单列排序、多列排序

**P97**

1. **大题30分：给定几个关系表，能用关系代数、SQL实现不同的查询**
2. Like查询中“%”与“-”的应用

**P95**

1. 创建索引的语句

**P89**

1. SQL中对两个集合取差集运算，使用except关键字；并集运算用UNION关键字

**P113**

# 查询优化

1. 数据库查询处理的主要步骤及内容。

**查询分析、查询检查、查询优化和查询执行**

**(词法分析，语法分析)（词义分析，符号名转换，安全性检查，完整性初步检查）**

**（代数优化，物理优化）（代码生成00）**

**P276**

1. 代数优化的一般原则

**选择运算应尽可能先做**

**把投影运算和选择运算同时进行**

**把投影同其前或后的双目运算结合起来**

**把某些选择同在它前面要执行的笛卡尔积结合起来成为一个连接运算**

**P284**

1. 哪些情况对属性建立索引比较合适？

**1. 主键自动建立唯一索引。**

**2. 频繁作为查询条件的字段应该创建索引。**

**3.查询中与其他表关联的字段，外键关系建立索引。**

**4. 频繁更新的字段不适合创建索引，因为每次更新不单单是更新了记录还会更新索引文件。**

**5. where条件里用不到的字段不创建索引。**

**6. 单键/组合索引的选择问题，who？（在高并发下倾向创建组合索引）。**

**7. 查询中排序的字段，排序字段若通过索引去访问将大大提高排序速度。（索引干两件事：检索和排序）。**

**8. 查询中统计或者分组字段。**

# 关系规范化

1. 给定一个关系，能判断是否属于第一范式、第二范式和第三范式、BC范式
2. 大题20分：给定一个不规范的表，标注表的函数依赖关系，能判断属于第几范式，如果不规范能将表规范到第三范式、BC范式。**考试内容都在PPT中**
3. 并不是规范化程度越高的关系模式就越好。
4. 任何一个二目关系都属于BCNF。
5. 已知关系R具有属性ABCDEF ,假设该关系有如下函数依赖 AB->C, BC->A,D->E,F->B, 则{A,B}的属性集的闭包是什么？

# 数据库设计

1. 局部E-R图合并成全局E-R图时可能出现的合并冲突有哪几种。

**1.属性冲突**

**属性域冲突（int、string）**

**属性取值单位冲突（米、厘米）**

**解决方法：协商讨论**

**2.命名冲突（可能发生在属性级、实体级、联系级上，属性更为常见）**

**同名异义**

**异名同义**

**解决方法：协商讨论**

**3.结构冲突**

**1）同一对象在不同应用中具有不同的抽象（如课程在一个局部应用中被当成实体，另一个局部应用中被当成属性）**

**解决方法：通常是把属性变换为实体或把实体变换为属性，使同一对象具有相同的抽象。变换时要遵循两个准则。**

**2）同一实体在不同局部视图中所包含的属性不完全相同，或者属性的排列次序不完全相同**

**产生原因：不同的局部应用关心的是该实体的不同侧面。**

**解决方法：使该实体的属性取各分E-R图中属性的并集，再适当设计属性的次序。**

**3）实体之间的联系在不同局部视图中呈现不同的类型**

**（例1， 实体E1与E2在局部应用A中是多对多联系，而在局部应用B中是一对多联系**

**例2， 在局部应用X中E1与E2发生联系，而在局部应用Y中E1、E2、E3三者之间有联系。）**

**解决方法：根据应用语义对实体联系的类型进行综合或调整。**

# 数据库恢复

1. 什么是事务故障、系统故障和介质故障？产生原因，如果进行故障恢复？

**事务故障**：某个事务在运行过程中由于种种原因未运行至正常终止点就夭折了

**事务故障的常见原因**：

输入数据有误

运算溢出

违反了某些完整性限制

某些应用程序出错

并行事务发生死锁

发生事务故障时，夭折的事务可能已把对数据库的部分修改写回磁盘

**事务故障的恢复**：撤消事务（UNDO）

强行回滚（ROLLBACK）该事务

清除该事务对数据库的所有修改，使得这个事务象根本没有启动过一样

**系统故障**：

整个系统的正常运行突然被破坏

所有正在运行的事务都非正常终止

内存中数据库缓冲区的信息全部丢失

外部存储设备上的数据未受影响

**系统故障的常见原因**：

操作系统或DBMS代码错误

操作员操作失误

特定类型的硬件错误（如CPU故障）

突然停电

**系统故障的恢复**

清除尚未完成的事务对数据库的所有修改

系统重新启动时，恢复程序要强行撤消（UNDO）所有未完成事务

将缓冲区中已完成事务提交的结果写入数据库

系统重新启动时，恢复程序需要重做（REDO）所有已提交的事务

**介质故障**

硬件故障使存储在外存中的数据部分丢失或全部丢失

介质故障比前两类故障的可能性小得多，但破坏性大得多

硬件故障

磁盘损坏

磁头碰撞

操作系统的某种潜在错误

瞬时强磁场干扰

**介质故障的恢复**

装入数据库发生介质故障前某个时刻的数据副本

重做自此时始的所有成功事务，将这些事务已提交的结果重新记入数据库

1. 掌握事务的概念及事务的ACID四个特性

**事务(Transaction)：是用户定义的一个数据库操作序列，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**

**一个事务可以是一条SQL语句，一组SQL语句或整个程序构成；**

**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**

**ACID特性**

**原子性（Atomicity）**

原子性：一个事务中的操作要么都做，要么都不做事务是不可拆分的。

事务必须以 Commit/Rollback 结束

**一致性（Consistency）**

一致性：使数据库从一个一致性状态变到另一个一致性状态。单独运行的事务，必须保证保持数据库的一致状态。一致性与原子性密切相关。

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：数据库中包含失败事务的结果

假如数据库的状态满足所有的完整性约束，就说该数据库是一致的

**隔离性（Isolation）**

隔离性：一个事务的执行不被其它事务干扰，并发不影响事务的执行。

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。

由数据库系统的并发控制管理机制负责

**持续性（Durability ）**

持续性：一个事务一旦提交，它对数据库的改变应是永久性的，接下来的其它操作或故障不应该对其执行结果有任何影响。

由数据库系统的恢复管理机制负责；

持久性意味着当系统或介质发生故障时，确保已提交事务的更新不能丢失。即一旦一个事务提交，DBMS保证它对数据库中数据的改变应该是永久性的，耐得住任何系统故障。

持久性通过事务日志、数据库备份和恢复来保证。

1. 常见的数据库保护措施有安全性保护，完整性保护，并发控制及其故障恢复
2. 登记日志文件时可以先写日志文件，后写数据库，也可以先写数据库，后写日志文件。
3. 几种转储（备份）方式

**备份方法**

**静态备份与动态备份**

**静态备份**

在系统中无运行事务时进行备份

备份开始时数据库处于一致性状态

备份期间不允许对数据库的任何存取、修改活动

优点：实现简单

缺点：降低了数据库的可用性

备份必须等用户事务结束

新的事务必须等备份结束

动态备份

备份操作与用户事务并发进行

备份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

动态备份的缺点

不能保证副本中的数据正确有效

**海量备份与增量备份**

海量备份（完全备份）: 每次备份全部数据库

增量备份（差异备份）: 只备份上次备份后更新过的数据

海量备份与增量备份比较

从恢复角度看，使用海量备份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，则增量备份方式更实用更有效

1. 数据库并发操作可能带来的数据不一致的几种形式

**丢失修改（lost update）**

丢失修改是指事务A与事务B从数据库中读入同一数据并修改。事务B的提交结果破 坏了事务A提交的结果，导致事务A的修改被丢失。

**不可重复读（non-repeatable read）**

不可重复读是指事务A读取数据后，事务B执行更新操作，使事务A无法再现前一次 读取结果。

三类不可重复读

事务A读取某一数据后：

事务B对其更新，事务A再次读得到与前一次不同的值。

事务B删除其中部分记录，事务A再次读取，发现某些记录神密地消失。

事务B插入了一些记录，当事务A再次按相同条件读取数据时，发现多了一些记录。

**读“脏”数据（dirty read）**

事务B修改某一数据，并将其写回磁盘事务A读取同一数据后事务B由于某种原因撤 消已修改过的数据恢复原值，事务A读到的数据就与数据库中的数据不一致，是不正 确的数据，又称为“脏”数据。

三种数据不一致产生的主要原因：并发操作破坏了事务的隔离性

并发控制就是要用正确的方式调度并发操作，使一个事务的执行不受其他事务的干扰， 从而避免造成数据的不一致性,保证事务的隔离性。

1. 事务的定义、提交和回滚操作
2. 封锁机制是数据库并发控制的主要方式
3. 锁的主要类型及相容矩阵

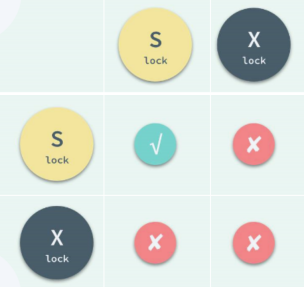
**基本封锁类型**

**排它锁又称为写锁（X锁）**

若事务T对数据对象A加上X锁，则只允许T读取和修改A，其它任何事务都不能再对A加任何类型的锁，直到T释放A上的锁

**共享锁又称为读锁（S锁）**

若事务T对数据对象A加上S锁，则其它事务只能再对A加S锁，而不能加X锁，直到T释放A上的S锁



1. 三级封锁协议及两段锁协议2PL

**1级封锁协议**

事务T在修改数据R之前必须先对其加X锁，直到事务结束才释放

正常结束（COMMIT）

非正常结束（ROLLBACK）

1级封锁协议可防止丢失修改

在1级封锁协议中，如果是读数据，不需要加锁的，所以它不能保证可重复读和不读“脏”数据。

**2级封锁协议**

1级封锁协议+事务T在读取数据R前必须先加S锁，读完后即可释放S锁

2级封锁协议可以防止丢失修改和读“脏”数据。

在2级封锁协议中，由于读完数据后即可释放S锁，所以它不能保证可重复读。

**3级封锁协议**

1级封锁协议 + 事务T在读取数据R之前必须先对其加S锁，直到事务结束才释放

3级封锁协议可防止丢失修改、读脏数据和不可重复读。

**两段锁协议2PL**

两段锁协议的内容

在对任何数据进行读、写操作之前，事务首先要获得对该数据的封锁

在释放一个封锁之后，事务不再获得任何其他封锁。

“两段”锁的含义：事务分为两个阶段

第一阶段是获得封锁，也称为扩展阶段；

第二阶段是释放封锁，也称为收缩阶段。

并行执行的所有事务均遵守两段锁协议，则对这些事务的所有并行调度策略都是可串行 化的。

所有遵守两段锁协议的事务，其并行执行的结果一定是正确的

事务遵守两段锁协议是可串行化调度的充分条件，而不是必要条件

可串行化的调度中，不一定所有事务都必须符合两段锁协议。

1. 满足两段锁协议是数据库并发调度可串行化的充分条件。
2. 一个调度是冲突可串行化，则一定是可串行化的调度 。

**主要的简答题：**

1. 数据库管理系统的主要功能

**课本P5**

1. SQL 语言中提供了哪些数据控制（自主存取控制）的语句？请试举几例说明它们的使用方法。

**SQL 中的自主存取控制是通过GRANT语句和REVOKE语句来实现的。如:**

**GRANT SELECT , INSERT ON Student**

**TO 王平**

**WITH GRANT  OPTION;**

**就将Student 表的SELECT 和INSERT 权限授予了用户王平,后面的“WITH GRANT  OPTION ”子句表示用户王平同时也获得了“授权”的权限,即可以把得到的权限继续授 予其他用户。**

**REVOKE INSERT ON Student FROM 王平CASCADE;**

**就将Student 表的INSERT 权限从用户王平处收回,选项CASCADE 表示,如果用户王平将 Student 的INSERT 权限又转授给了其他用户,那么这些权限也将从其他用户处收回。**

1. 为什么要引进意向锁？意向锁的含义是什么？举例说明。

引进意向锁（intention lock）目的

提高对某个数据对象加锁时系统的检查效率

例：对任一元组 r 加锁，先关系R加意向锁

事务T要对关系R加X锁, 系统只要检查根结点数据库和关系R是否已加了不相容的锁，

不需要搜索和检查R中的每一个元组是否加了X锁

**常用意向锁**

**意向共享锁(Intent Share Lock，简称IS锁)**

**意向排它锁(Intent Exclusive Lock，简称IX锁)**



1. 关系数据库系统中，当操作违反实体完整性、参照完整性和用户定义的完整性约束条件时，一般是如何分别进行处理的？

**对于违反实体完整性和用户定义的完整性的操作一般都采用拒绝的方法进行处理。而对于违反参照完整性的操作，并不都是简单地拒绝执行，有时要根据应用语义执行一些附加的操作，以保证数据库的正确性。**

**实体完整性：**

**（1）检查主码是否唯一，如果不唯一则拒绝插入或修改；**

**（2）检查主码的各个属性是否为空，只要有一个为空就拒绝插入或修改**

**参照完整性：**

**（1）拒绝（NO ACTION）执行**

**（2）级联（CASCADE）执行**

**（3）设置为空值**

**用户定义完整性：**

**（1）不允许空值**

**（2）列值唯一**

**（3）用CHECK短语指定列值应该满足的条件**

1. 什么是数据库镜像？它有什么用途？

**数据库镜像即根据DBA的要求，自动把整个数据库或者其中的部分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，即DBMS自动保证镜像数据与主数据的一致性。  
数据库镜像的用途有：  
 一是用于数据库恢复。当出现介质故障时，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。  
 二是提高数据库的可用性。在没有出现故障时，当一个用户对某个数据加排它锁进行修改时，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**

1. 数据库事务、系统故障的恢复策略是什么？

**事务故障的恢复**

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

3)继续反向扫描

4)直至读到该事务的开始标记，事务故障恢复完成

**系统故障的恢复**

1)正向扫描日志文件，找出在故障发生前已经提交的事务标识记入重做序列，故障发生时尚未完成的事务标识记入撤销队列

2)对撤销队列中的各个事务进行重做处理

3)对重做队列中的各个事务进行重做处理