

一、单选 (1x10=10分)

无

二、填空 (1x10=10分)

1. 数据库系统

数据库系统的构成

- 数据库
- 数据库管理系统 (及其应用开发工具)
- 应用程序
- 数据库管理员

数据库系统的特点

- 数据结构化
- 数据的共享性高，冗余度低且易扩充
- 数据独立性高
- 数据由数据库管理系统统一管理和控制

2. 三级模式

模式 (Schema)

外模式 (External Schema)

内模式 (Internal Schema)

3. 数据模型

==数据模型是由数据结构、数据操作、数据的完整性约束条件组成==

数据模型分为两类（两个不同的层次）

（1） ==概念模型== 也称信息模型，它是按用户的观点来对数据和信息建模，用于数据库设计。

（2） ==逻辑模型和物理模型==

常用的数据模型：

层次模型、网状模型、关系模型

4. 需求分析

5. 数据库备份数据

6. DBMS多种存取方式

数据库管理系统常用存取方法（7.5.2）

1. B+树索引存取方法

2. Hash索引存取方法

3. 聚簇存取方法

7. 数据库安全性

8. 并发控制

事务是并发控制的**基本单位**

基本封锁类型：排它锁、共享锁

一级封锁协议**可防止丢失修改**

二级封锁协议**可以防止丢失修改和读“脏”数据。**

三级封锁协议**可防止丢失修改、读脏数据和不可重复读。**

避免活锁：采用**先来先服务**的策略

死锁的诊断

（1） 超时法

（2） 等待图法

在多粒度封锁中一个数据对象可能以两种方式封锁：**显式封锁和隐式封锁**

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)

共享意向排它锁(Share Intent Exclusive Lock, 简称SIX锁)

9. 三类数据不一致

并发操作带来的数据不一致性：丢失修改、不可重复读、读脏数据

10. 检查点记录

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

三、简答 (5x4=20分)

1. 什么是数据库三级模式？如何保证数据独立性

数据库系统的三级模式结构指：数据库系统是由外模式、模式、内模式三级构成。模式是数据库中全体数据的逻辑结构和特征的描述；内模式是数据物理结构和存储方式的描述；外模式是数据库用户使用的局部数据的逻辑结构和特征的描述

数据库管理系统提供了**两层映像**：外模式/模式、模式/内模式映像

当模式改变时，数据库管理员对外模式 / 模式映象作相应改变，使外模式保持不变，应用程序是依据数据的外模式编写的，应用程序不必修改，保证了数据与程序的逻辑独立性

当数据库的存储结构改变了，数据库管理员修改模式 / 内模式映象，使模式保持不变。应用程序不受影响。保证了数据与程序的物理独立性

2. 怎样进行系统故障恢复

系统故障的恢复步骤:

(1) 正向扫描日志文件

- 在故障发生前已经提交的事务放入重做(REDO) 队列:
- 故障发生时尚未完成的事务放入撤销 (UNDO) 队列

(2) 对撤销(UNDO)队列事务进行撤销(UNDO)处理

反向扫描日志文件, 对每个撤销事务的更新操作执行逆操作
即将日志记录中“更新前的值”写入数据库

(3) 对重做(REDO)队列事务进行重做(REDO)处理

正向扫描日志文件, 对每个重做事务重新执行登记的操作
即将日志记录中“更新后的值”写入数据库

3. 什么是排它锁、共享锁

排它锁

- 排它锁又称为写锁
- 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
- 保证其他事务在T释放A上的锁之前不能再读取和修改A

共享锁

- 共享锁又称为读锁
- 若事务T对数据对象A加上S锁, 则事务T可以读A但不能修改A, 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
- 保证其他事务可以读A, 但在T释放A上的S锁之前不能对A做任何修改

4. 数据库设计的基本步骤有哪些? 视图的结构设计在哪个阶段

需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库运行和维护。视图在逻辑结构设计

5. 什么是活死锁?

活锁：对封锁请求不按照先后顺序批准，如果持续有新的封锁请求产生，可能导致一些事务的封锁请求永远得不到批准，处于永远等待状态

死锁：多个事务已封锁了一些数据对象，然后又都请求已被其它事务封锁的数据加锁，从而出现互相等待的情况

6. 什么是可串行化调度

多个事务并发执行的结果和按某一次序串行调度它们的结果相同，则这种调度策略为可串行化调度

7. 具有检查点的恢复技术有什么优点

有检查点后，恢复子系统不必从头扫描整个日志，只要从最后一个检查点记录开始扫描，减少了扫描日志的时间

在检查点之前提交的事务的更新操作已经写入数据库，不必再重做，节省了重做这些事务的时间。

8. 介绍检测死锁的一种方法；发生死锁如何解除

超时法：如果一个事务的等待时间超过了规定的时限，就认为发生了死锁

等待图法：

用事务等待图动态反映所有事务的等待情况

事务等待图是一个有向图 $G=(T, U)$

T为结点的集合，每个结点表示正运行的事务

U为边的集合，每条边表示事务等待的情况

若T1等待T2，则T1，T2之间划一条有向边，从T1指向T2

并发控制子系统周期性地（比如每隔数秒）生成事务等待图，检测事务。如果发现图中存在回路，则表示系统中出现了死锁。

解除死锁

选择一个处理死锁代价最小的事务，将其撤消

释放此事务持有的所有的锁，使其它事务能继续运行下去

9. 论述两段锁协议概念

指所有事务必须分两个阶段对数据项加锁和解锁

- 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，事务不再申请和获得任何其他封锁

“两段”锁的含义

事务分为两个阶段

第一阶段是获得封锁

事务可以申请获得任何数据项上的任何类型的锁，但是不能释放任何锁

第二阶段是释放封锁

事务可以释放任何数据项上的任何类型的锁，但是不能再申请任何锁

10. 什么是数据库自主存取控制和强制存取控制？

自主存取控制：定义各个用户对不同数据对象的存取权限。当用户对数据库访问时首先检查用户的存取权限。防止不合法用户对数据库的存取。用户可以将其所拥有的存取权限转授其他用户

强制存取控制：每个数据对象被强制标以一定密级，每个用户也被强制授予一个级别的许可证。对于任意一个数据对象，只有具有合法许可证的用户才可以存取

四、关系规范化（10分）

1. 写出关系模式的基本函数依赖和候选码
2. 关系最高到几范式？说明理由
3. 将关系规范到高级范式

五、应用题（20分）

1. 设计ER
2. 将ER转化为关系模型

六、关系运算 (30分)

1. 用关系代数表达查询

选择、投影、连接、差、除

2. 用SQL表达

查、改、删、聚集函数、授权 (双重否定不考)

自背

什么是数据库三级模式？如何保证数据独立性

怎样进行系统故障恢复

什么是排它锁、共享锁

数据库设计的基本步骤有哪些？视图的结构设计在哪个阶段

什么是活死锁？

什么是可串行化调度

具有检查点的恢复技术有什么优点

介绍检测死锁的一种方法；发生死锁如何解除

论述两段锁协议概念

什么是数据库自主存取控制和强制存取控制？