

## 填空题

---

### 文件系统

- 文件系统管理数据特定：
  - i. 数据可以长期保存
  - ii. 由文件系统管理数据，有专门的文件系统进行数据管理
- 共享性差，冗余度大
- 数据独立性差

### 数据库

- 数据：数据库中存储的基本对象
- 数据库：长期存储在计算机内、有组织、可共享 的大量数据的集合。数据库中的数据，按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性和易扩展性，可为各种用户共享

### DBMS内部体系结构

- 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训管理
  - 数据库的建立和维护功能

### 关系型数据库特点

- 实体之间通过关系相关连，存储数据的时采用的就是二维表格形式
- 关系型数据库的值是这些关系模式在某一时刻对应的关系的集合

### 关系数据模型

- 关系模型建立在严格的数学概念的基础上
- 概念单一，数据结构简单
- 数据独立性高、更好的安全性

### 三类完整性

- 实体完整性约束（主键约束primary key）
  - 检查是否为空
  - 检查是否唯一
  - 为空或不唯一，则为违反约束，拒绝执行操作
- 参照完整性约束（外键约束foreign key）
  - 需要在被参照表中存在
  - 参照表元素在被参照表中不存在则违反约束
  - 决绝执行操作
  - 级联更新，被参照表更行，参照表同时更新
  - 值置空
- 用户自定义完整性约束
  - 不满足用户的定义，则拒绝执行

## #### 并发控制正确性

---

### 物理结构设计

- 概念：为一个给定的逻辑数据模型选取一个最适合要求的物理结构的过程
  - 步骤：
    - i. 确定数据库的物理结构，主要指存取方式和存储结构
    - ii. 评估物理结构，时间空间效率

### 存取方法的选择

- 存取方法
  - i. B+树索引存取方法：根据应用要求确定对关系的那些属性创建索引
  - ii. hash索引存取方法：关系属性主要出现在等值连接选择条件中
    - 条件：
      - a. 关系大小可预知且不改变
      - b. 关系大小动态可变，数据库管理系统提供动态hash存取方法
  - iii. 聚簇索引存取方式：将具有相同值的元组集中放在连续的物理块上

- 一个数据库可以建立多个聚簇，一个关系只能属于一个聚簇
- 建立和维护聚簇的开销大。对于关系的元组操纵时，会使聚簇发生改动，开销大

## 存储结构

- 确定物理结构指确定数据存放的位置和存储结构
  - i. 确认数据存放的位置：易变数据与稳定数据、经常访问数据与存取频率低的数据都需要分开存储
  - ii. 确定系统配置：数据库管理系统的相关配置参数的设置

## 评价物理结构

- 空间时间效率、维护代价和用户要求进行权衡

## 恢复原则

- 故障之前提交的事务重做
- 故障时未提交的事务回滚
- 故障之后开始的事务不做任何操作

## 检查点

- 使用检查点可以改善恢复效率
- 动态维护日志：周期设置检查点，保存数据库状态
- 检查点的使用
  - i. 从重做文件中找到最后一个检查点记录在日志中的位置
  - ii. 从检查点得到在检查点建立时刻正在执行的事务清单
  - iii. 从检查点开始正向扫描日志文件并恢复

## 事务

- 概念：用户定义的一个操作序列，操作要么全部执行，要么全部失败
  - commit提交
  - rollback回滚
- 事务ACID特性
  - atomicity：原子性：事务中的操作要么全做，要不全失败
  - consistency：一致性：事务执行结果必须使数据库从一个一致状态到另一个一致状态
  - isolation：隔离性：各事务之间不干扰。并发时相互不影响
  - durability：持久性：事务一旦提交，对数据库的更改时永久的

## 死锁诊断

- 超时法
  - 一个事务等待时间超过限度，就认为发生了死锁
- 等待图法（有向图）
  - 每个事务是一个节点，每个有向边为事务等待情况，出现环路则出现死锁
- 死锁破除
  - 选择代价最小的事务处理掉，将其撤销释放持有的所有锁是其他事务能够运行

## 简答题

### 视图的定义和优点

- 定义
  - 从一个或多个基本表导出的表。只存放视图的定义，不存放数据，是一个虚表
- 优点
  - 简化用户操作，视图定义使数据库结构看起来简单清晰
  - 视图对重构数据库提供了一定程度的逻辑独立性

### 三级封锁协议及解决的问题？

- 协议：通过在一级封锁协议的基础上增加事务在读取数据之前加读锁（S锁），直到事务结束才释放
- 解决的问题：三级封锁协议解决了丢失修改、读脏数据和不可重复读的问题

## 断言的操作

- 通过CREATE ASSERTION来创建断言，实现更具一般性的约束

```
CREATE ASSERTION ASSERTION_NAME
CHECK (CONDITION);
```

## 事务处理过程中发生系统故障会发生什么情况

- 事务过程中发生系统故障会导致内存中的数据会发生丢失，事务中部分数据写入数据库中，导致数据不一致
- 需要撤销所有未完成事务，重做已提交事务

## 什么是数据库镜像，有什么用途

- 含义：数据库镜像是DBMS根据DBA的要求，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会自动把更新后的数据复制到从数据库
- 出现主数据库错误可以及时用从数据库顶替或者从从数据库恢复数据

## 并发操作会导致哪几种数据不一致？怎么避免？

- 数据不一致类型：
  - i. 丢失修改
  - ii. 不可重复读
  - iii. 脏读
- 如何避免：
  - i. 丢失修改：一级封锁（加写锁，事务结束时释放）
  - ii. 脏读：二级封锁（一级封锁基础上读时加读锁，读完就释放）
  - iii. 不可重复读：三级封锁（一级封锁基础上都是加读锁，事务结束时释放）

## 瞬时强磁场造成的数据故障属于哪一种类型故障，如何恢复

- 故障类型：介质故障
- 恢复：装入数据库发生介质故障前某个时刻的数据副本，然后重做所有成功事务，将这些事务已提交的结果重新记入数据库

## 简述关系数据库的完整性规则，违反完整性约束条件时，如何处理

- 简述：定义部分列为外码，指明外码参照那些表的主码
- 违反时处理方式：
  - i. 拒绝执行：不允许执行操作
  - ii. 级联操作：删除或修改被参照表时，参照表也同时删除或修改响应元组
  - iii. 设为空值：置空

## 简述两阶段封锁协议，用途是什么？两阶段协议可以预防死锁发什吗

- 含义：事务必须分两阶段对数据加锁和解锁
  - i. 对数据进行读写是需要申请并获得锁
  - ii. 在释放时不再申请其他锁
- 用途：用于实现并发调度的可串行性，保证调度的正确性
- 两阶段封锁协议不能预防死锁发生

## 为什么要引入意向锁，意向锁的含义是什么

- 意义：提高对某个数据对象加锁时系统的检查效率。数据库管理系统就不用逐个检查下级结点的显式封锁
- 含义：如果对一个结点加意向锁，则说明该结点的下层结点正在被封锁；对任一结点加锁时，必须先对它的上层结点加意向锁

## 关系规范题（10分）

### 范式

- 第一范式：每个分量都必须是不可再分的数据
- 第二范式：满足第一范式，且每一个非主属性完全函数依赖于任何一个候选码
- 第三范式：每一个非主属性既不传递依赖于码，也不部分依赖于码
- BC范式：

### 候选码选择

- 写出关系模式的基本函数依赖和候选码（如何选取）
  - i. 如果属性不在函数依赖集中出现，则必须包含在候选码中
  - ii. 如果属性不在函数依赖中任何函数依赖的右边出现，则必须包含在候选码中
  - iii. 如果只在依赖集左边出现，则必为候选码
  - iv. 如果有属性或者属性组能唯一标识元组，则可为候选码
- R最高达到第几范式，说明理由
- R规范到最高范式

### 应用题（20分）

- 设计E-R图
- 将该E-R图转换为关系模型（要求：1：1和1：n的联系进行合并）

### 关系运算题（30分）

- 关系代数表达式如下查询：
  - 选择、投影、连接、差、除（必考）
- sql语句查询
  - 查询、修改、删除、汇聚函数、授权