



# 简答题总结

- 1.数据模型的概念
- 2.关系模型
- 3.数据库系统的三级模式结构
- 4.E-R模型
- 5.从E-R模型到关系模型
- 6.关系的完整性约束
- 7.数据库的安全性
- 8.自主存取控制（DAC）和强制存取控制（MOC）
- 9.关系模型的范式
- 10.查询优化
- 11.代数优化
- 12.事务
- 13.封锁及封锁协议
- 14.活锁和死锁
- 15.并发调度的可串行化
- 16.数据库的恢复
- 17.数据字典
18. 等值连接和自然连接的区别

## 1.数据模型的概念

数据模型是一种形式机制，用于**数据建模、描述数据、数据之间的联系、数据的语义、数据上的操作和数据的完整性约束条件**。

**数据模型的三要素：数据结构、数据操作、数据的完整性和约束条件**

## 2.关系模型

关系模型的三要素：

- 关系模型只有一种数据结构——关系
- 关系模型的数据操作
- 关系模型的完整性约束



关系模型要求关系必须是规范化的：关系的每个属性只能取原子值（即表中不能包含子表）

在关系模型中，定义数据操作的方法有两种：关系代数和关系演算

### 3.数据库系统的三级模式结构

外模式、模式、内模式

数据独立性是指数据与应用程序相互独立

数据的独立性靠三级模式、两级映像实现

- 外模式-模式映像定义外模式与模式之间的对应关系，可以保证外模式的相对稳定性，为数据的逻辑独立性提供了保证
- 模式-内模式映像定义数据全局逻辑结构与存储结构之间的对应关系，为数据的物理独立性提供了保证

### 4.E-R模型

实体-联系模型是一种概念模型，用于对现实世界的建模

概念简单，并具有很强的语义表达能力、容易转换成实际数据库管理系统支持的数据模型、是最广泛使用的对现实世界进行建模的工具

由实体、联系、属性构成

E-R图是用来描述某一组织的概念模型。

### 5.从E-R模型到关系模型

包括属性处理、实体集处理、联系集处理

- 派生属性的处理：转换时忽略派生属性
- 复合属性的处理：忽略复合属性本身，而直接考虑它的成分属性
- 多值属性的处理：为每个多值属性创建一个关系
- 强实体集处理：实体集名可以作为关系名，实体集的全部属性构成关系属性
- 弱实体集的处理：弱实体集可以作为关系名，弱实体集存在依赖的标识实体集的主码和弱实体集的全部属性构成关系属性

- 联系是一对一的。则可以和任意一端的实体集转换得到关系并合并
- 联系是一对多的，则和多端实体集转换得到的关系模式合并
- 联系是多对多的，则该联系集转换为独立的关系模式

## 6.关系的完整性约束

数据库的完整性是指数据库中的数据的**正确性、一致性、相容性**。

实体完整性约束：关系R上的所有元组在主码上的值必须唯一，并且在主码的任何属性上都不能取空值

参照完整性约束：如果属性集FK是关系R的外码，它参照关系S的主码Ks，则R的任意元组在FK的值等于S的某个元组在主码Ks上的值，或为空值

## 7.数据库的安全性

数据库的安全性是指保护数据库，防止因用户非法使用数据库造成数据泄露、更改或者破坏

数据库的安全分层：

- **物理层**
- **人际层**
- **网络层**
- **操作系统层**
- **数据库系统层**

保护数据库设计的任务：

- 防止数据的未经授权的存取
- 防止未授权人员对数据进行删除和修改
- 监控对数据的访问和更改等使用情况

## 8.自主存取控制（DAC）和强制存取控制（MOC）

自主存取控制是**通过授权实现**的，因此又被称为授权

强制存取控制：在自主存取控制中，系统根据用户对数据库对象的存取权限来进行安全控制，而不考虑数据库本身的安全等级。

- **强制存取控制规则**

- 仅当主体的许可证级别**大于等于**客体的密级时，该主体才能读取相应的客体
- 仅当主体的许可证级别**小于等于**客体的密级时，该主体才能写相应的客体

## 9.关系模型的范式

- **1NF**：属性是原子的
- **2NF**：非主属性不部分依赖于码  
(属性完全依赖于码，消除了非主属性对键的部分函数依赖)
- **3NF**：非主属性不传递依赖于码  
(消除了非主属性对键的传递函数依赖，属性不依赖于其它非主属性)
- **BCNF**：任何函数依赖的决定部分都包含码  
(消除了主属性对键的部分函数依赖和传递函数依赖，一个表中不含其他表中的已包含的非主属性信息)
- **4NF**：多值依赖的决定部分包含码  
(非主属性不应该有多个值，即处理相互独立的多值情况)
- **5NF**:处理相互依赖的多值情况。

## 10.查询优化

选择运算的实现：

- **基本算法**
  - 线性搜索（顺序扫描）
  - 二分法搜索
- **使用索引的选择**
  - 聚簇索引（主索引）
  - 非聚簇索引（辅助索引）
    - **只建立索引结构，而不改变元组次序**

连接运算的实现：

- 嵌套循环
- 块嵌套循环
- 索引嵌套循环

- 归并连接
- 散列连接
- 排序-归并连接

## 11.代数优化

代数优化的启发式方法：

- 选择运算应当尽可能先做
- 投影运算应当尽可能先做（选择一般优于投影）
- 尽量避免笛卡尔积运算

其他启发式优化技术

- 提取公共子表达式
- 流水线技术

## 12.事务

- 什么是事务？
  - 事务是用户定义的一个数据库的操作序列，这些操作要么全做，要么不做，是一个不可分割的工作单元
  - 事务是并发控制与调度的基本单位，也是数据库恢复的基本单位
- 事务的特性
  - 事务具有四个特性（ACID）
  - 原子性：事务是数据库系统运行的原子程序单元
  - 一致性：事务隔离执行时要保持数据库的一致性即正确性，即事务的执行结果必须是使数据库从一个一致的状态变到另一个一致性的状态
  - 隔离性：一个事务的执行不能被其他事务干扰
  - 持久性：一个事务完成后，它对数据库的改变必须是永久的，即使系统出现故障时也是这样
- 事务状态
  - 已提交事务：成功完成的事务

- **终止事务**：没能顺利完成的事务
- **回滚**：撤销终止事务造成的变更

### 13. 封锁及封锁协议

- 锁的类型
  - 共享锁（读锁，S锁）
    - 若事务T获得了数据对象Q上的S锁，则**T可以读但不能写Q**，并且在T释放S锁之前，其他事务**只能获得Q上的S锁，不能获得X锁**
  - 排他锁（写锁，X锁）
    - 若事务T获得了数据对象Q上的X锁，则**T既可以读也可以写Q**，但是在Q释放X锁之前，其他事务**既不能获得S锁，也不能获得X锁**
- 封锁协议
  - 数据对象加锁时需要约定一些规则，约定何时申请封锁何时释放封锁，这些规则被称为封锁协议
  - 一级封锁协议
    - 事务T在**更新数据对象Q之前必须先对其加X锁**，直到事务结束才释放
    - 作用：可防止丢失修改
    - 问题：一级封锁协议对于只读操作，无需加锁，不能保证读脏数据和可重复读
  - 二级封锁协议
    - 在一级封锁协议的基础上，进一步要求**事务T在读取数据对象之前必须先对其加S锁，但是读完后可以立即释放S锁**
    - 作用：可以进一步防止读脏数据
    - 问题：但不能保证可以重复读
  - 三级封锁协议
    - 在一级封锁协议的基础上，进一步要求事务T在读取数据对象前必须先对其加S锁，并且**直到事务结束才能释放S锁**
    - 可以保证重复读

## 14.活锁和死锁

- 活锁
  - 又称饥饿，是某个事务因等待锁而处于无限期等待状态
  - 解决办法：先来先服务，当多个事务请求封锁同一数据对象时，封锁子系统按照请求的先后顺序对事务排队
- 死锁
  - 死锁是两个或两个以上的事务之间的循环等待现象
  - 预防
    - 一次封锁法
      - 对所有使用的数据全部加锁后，再实际进行事务操作
      - 问题：降低系统的并发度，事先确定事务要封锁的数据对象困难
    - 顺序封锁法
      - 事务按照预先规定的顺序对数据实行封锁
      - 问题：维护资源的封锁顺序非常困难
  - 检测
    - 超时法
    - 等待图法
  - 解除
    - 选择一个或多个处于死锁状态的事务，将其撤销并释放这些事务所持有的锁
    - 遵循最小代价原则

## 15.并发调度的可串行化

- 串行调度
  - 串行调度总是正确的
  - $n$ 个事务，串行调度总数= $n!$
- 并发调度

- 即使只有两个事务并发，执行顺序也可能有很多种
- $n$ 个事务，并发调度总数 $>n!$
- 并发调度是可串行化的：一组事务的一个并发调度是正确的，当且仅当调度的执行结果与某一个串行调度的执行结果相同
- 冲突可串行化
  - 冲突可串行化调度一定是可串行化调度，但是其逆命题不真
- 视图可串行化
  - 每个冲突可串行化的调度都是视图可串行化的，但是，存在视图可串行化的调度，它不是冲突可串行化的

## 16.数据库的恢复

- 故障的类型
  - 事务故障
  - 系统故障
  - 戒指故障
- 故障恢复的思想
  - 在系统正常运行时建立冗余数据，保证有足够的信息可用于故障恢复
  - 故障发生后采取措施，将数据库内容恢复到某个一致性状态，保证事务的原子性和持久性
- 冗余数据的建立
  - 数据库系统主要通过登记日志和数据转储来建立冗余数据
- 恢复需要考虑的因素
  - 存储器的性质
  - 更新输出到数据库的时机
  - 缓冲

## 17.数据字典

数据字典包括数据项、数据结构、数据流、数据存储和处理过程五部分



数据项是不可再分的数据单位，说明基本数据信息的数据类型、长度、取值范围等

数据结构反映了数据之间的组合关系

数据流是数据结构在系统内传输的路径

数据存储是数据结构停留或保存的地方，也是数据流的来源和去向之一

处理过程描述该过程的功能及处理要求

## 18. 等值连接和自然连接的区别

- 等值连接中不要求相等属性值的属性名相同，而自然连接要求相等属性值的属性名必须相同，即俩关系只有在同名属性才能进行自然连接
- 等值连接不将重复属性去掉，而自然连接去掉重复属性