

## 《数据库原理与应用》复习提要

题型

填空题：

单项选择题：

判断题：

简答题：

模式设计：

论述题：

### 第一章 绪论

一、知识点分类如下：

1. 需要了解的：数据管理技术的产生和发展过程、数据库系统的优点和好处、层次数据模型及网状数据模型的基本概念、数据库系统的组成、DBA的职责、数据库技术的主要研究领域等。
2. 需要牢固掌握的：概念模型的基本概念及其主要建模方法——E - R方法；关系数据模型的相关概念、数据库系统三级模式和两层映像的体系结构，数据库系统的逻辑独立性和物理独立性等。
3. 需要举一反三的：通过E-R方法描述现实世界的概念模型。
4. 难点：数据模型及数据库系统的体系结构。

二、具体内容

1. 数据管理技术的发展阶段：人工管理阶段、文件系统阶段、数据库阶段，各阶段主要特点。

2. 概念：数据、DB、DBMS、DBS、数据库系统

3. 数据模型

数据模型的概念：数据模型是现实世界数据特征的抽象。

数据模型的组成要素：数据结构、数据操作、数据完整性约束

常用数据模型：层次、网状、关系三种模型。上述三种模型各自的特点。

数据描述的三个领域：现实世界、信息世界和机器世界。

信息世界中的几个概念：实体(即客观存在可以相互区别的事物)、实体集(同类实体的集合)、属性(实体的特性)、码(唯一标识实体的属性(集))、域、联系。

机器世界中的四个概念：字段、记录、文件、键(主码)。

E-R图的设计：

E-R图三要素：实体(型)、属性、联系

联系的种类：1:1、1:n、m:n

如何将E-R图转化成各种数据模型

4. 数据库的体系结构

三级结构模式：外模式、模式、内模式

模式也称为逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图(与数据模型相对应)。

外模式也称子模式(subschema)或用户模式，它是数据库用户(包括应用程序员和最终用户)能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。

内模式也称存储模式(Storage Schema)，一个数据库只有一个内模式。它是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。

二级映象：模式/内模式、外模式/模式，二级映象保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

### 第二章 关系数据库

一、知识点分类如下：

1. 需要了解的：关系数据库理论产生和发展的过程，关系数据库产品的发展及沿革；关系演算的概念。
2. 需要牢固掌握的：关系模型的三个组成部分及各部分所包括的主要内容；牢固关系数据结构及其形式化定义；关系的三类完整性约束的概念。
3. 需要举一反三的：关系代数（包括抽象的语言及具体的语言）；关系代数中的各种运算（包括并、交、差、选择、投影、连接、除、广义笛卡尔积等）、元组关系演算语言 ALPHA及域关系演算语言 QBE等，能够使用这些语言完成各种数据操纵。
4. 难点：关系代数。由于关系代数较为抽象，因此在学习的过程中一定要结合具体的实例进行学习。

### 第三章 关系数据库标准语言 SQL

一、知识点分类如下：

1. 需要了解的：SQL语言发展的过程，从而进一步了解关系数据库技术和 RDBMS产品的发展过程。
2. 需要牢固掌握的：掌握 SQL语言的特点、SQL语言与非关系模型（层次模型、网状模型）数据语言的不同，从而体会 SQL语言之所以能够为用户和业界所接受，并成为国际标准的原因；体会面向过程的语言和 SQL语言的区别和优点；体会关系数据库系统为数据库应用系统的开发提供良好环境、减轻用户负担、提高用户生产率的原因。
3. 需要举一反三的：熟练而正确地使用 SQL语言完成对数据库的查询、插入、删除、更新操作，特别是各种各样的查询（简单查询、联接查询、使用库函数查询），视图，掌握 SQL语言强大的查询功能。在完成具体的 SQL语句时，要有意识地与关系代数、关系演算等语言进行比较，了解它们各自的特点。
4. 难点：用 SQL语言正确完成复杂查询。

### 第五章 关系数据理论

学习本章的目的有两个。一个是理论方面的，本章用更加形式化的关系数据理论来描述和研究关系模型。另一个是实践方面的，关系数据理论是我们进行数据库设计的有力工具。因此，人们也把关系数据理论中的规范化理论称为数据库设计理论，有的书把它放在数据库设计部分介绍以强调它对数据库设计的指导作用。

一、知识点分类如下：

1. 需要了解的：什么是一个“不好”的数据库模式；什么是模式的插入异常和删除异常；规范化理论的重要意义。
2. 需要牢固掌握的：关系的形式化定义；数据依赖的基本概念（函数依赖、平凡函数依赖、非平凡的函数依赖、部分函数依赖、完全函数依赖、传递函数依赖的概念，码、候选码、外码的概念和定义，多值依赖的概念）；范式的概念；从 1NF 到 4NF的定义；规范化的含义和作用。
3. 需要举一反三的：四个范式的理解与应用，各个级别范式存在的问题（插入异常、删除异常、数据冗余）和解决方法；能够根据应用语义，完整地写出关系模式的数据依赖集合，并能根据数据依赖分析某一个关系模式属于第几范式。  
属性集闭包的计算，主码的确定，求最小函数依赖集的方法，将一个模式分解为 3NF.(函数依赖的保持性、无损连接性)，无损分解的判定。无损分解决定一个关系能否分解，而保持依赖决定一个关系分解的好坏。
4. 难点：各个级别范式的关系。

### 第六章 数据库设计

本章讲解数据库设计方法和技术，内容的实践性较强。数据库设计的重要性在于：数据库设计技术是信息系统开发和建设中的核心技术。

一、知识点分类如下：

1. 需要了解的：数据库设计的特点；数据库物理设计的内容和评价；数据库的实施和维护。
2. 需要牢固掌握的：数据库设计的基本步骤；数据库设计过程中数据字典的内容；数据库设计各个阶段的具体设计内容、设计描述、设计方法等。

3. 要举一反三的：E-R 图的设计；E-R图向关系模型的转换。
4. 难点：技术上的难点是 E-R 图的设计，数据模型的优化（模式分解）。真正的难点是理论与实际的结合。

## 第七章 数据库恢复技术

一、知识点分类如下：

1. 需要了解的：什么是数据库的一致性状态。数据库运行中可能产生的故障类型，他们如何影响事务的正常执行，如何破坏数据库数据。数据转储的概念及分类。什么是数据库镜像功能。
2. 需要牢固掌握的：事务的基本概念和事务的 ACID性质。日志文件的内容及作用。数据库恢复的实现技术。登记日志文件所要遵循的原则。具有检查点的恢复技术。
3. 需要举一反三的：恢复的基本原理，针对不同故障的恢复的策略和方法。
4. 难点：日志文件的使用，系统故障恢复策略。

事务管理模块是 DBMS实现中的关键技术。事务恢复的基本原理是数据备份，它貌似简单，实际实现却很复杂。数据库的事务管理策略（不仅有数据库恢复策略，还有并发控制策略）和 DBMS缓冲区管理策略、事务一致性级别密切相关。

作为数据库管理员，必须十分清楚每一个使用中的 DBMS产品提供的恢复技术、恢复方法，并且能够根据这些技术正确制定出实际系统的恢复策略，以保证数据库系统 7 × 24 小时正确运行，保证数据库系统在遇到故障能及时恢复正常运行，提高抗灾难的能力。

## 第八章 并发控制

数据库是一个共享资源，当多个用户并发存取数据库时就会产生多个事务同时存取同一个数据的情况。若对并发操作不加控制就可能会存取和存储不正确的数据，破坏数据库的一致性。所以 DBMS必须提供并发控制机制。并发控制机制的正确性和高效性是衡量一个 DBMS性能的重要标志之一。

一、知识点分类如下：

1. 需要了解的：数据库 并发控制技术的必要性，活锁、死锁的概念。
2. 需要牢固掌握的：并发操作可能产生 数据不一致性的情况（丢失修改、不可重复读、读“脏数据”）及其确切含义；封锁的类型；不同封锁类型的（例如 X 锁，S 锁）的性质和定义，相关的相容控制矩阵；封锁协议的概念（一级封锁协议、二级封锁协议、三级封锁协议、两段锁协议）；封锁粒度的概念；多粒度封锁方法；多粒度封锁协议的相容控制矩阵。
3. 需要举一反三的：封锁协议与数据一致性的关系；并发调度的可串行性概念；两段锁协议与可串行性的关系（两段锁协议是可串行化的充分条件而不是必要条件）；两段锁协议与死锁的关系。
4. 难点：两段锁协议与串行性的关系；与死锁的关系；具有意向锁的多粒度封锁方法的封锁过程。

## 第九章 数据库安全性

一、知识点分类如下：

1. 需要了解的：TDI/TCSEC标准的主要内容；什么是计算机系统安全性问题；什么是数据库的安全性问题；统计数据库的安全性问题。
2. 需要牢固掌握的：安全等级；C2级 DBMS B1级 DBMS的主要特征；实现数据库 安全性控制的常用方法和技术有哪些（用户标识与鉴别、存取控制、视图、加密，审计）；数据库中的自主存取控制方法和强制存取控制方法。
3. 需要举一反三的：使用 SQL语言中的 GRANT语句和 REVOK语句来实现自主存取控制。
4. 难点：MAC机制中确定主体能否存取客体的存取规则，要理解并掌握存取规则为什么要这样规定。

## 第十章 数据库完整性

一、知识点分类如下：

1. 需要了解的：什么是数据库的完整性约束条件；完整性约束条件的分类；数据库的完整性概念与数据库

的安全性概念的区别和联系。

2. 需要牢固掌握的：DBMS完整性控制机制的三个方面，即完整性约束条件的定义、完整性约束条件的检查和违约反应。
3. 需要举一反三的：用 SQL语言定义关系模式的完整性约束条件。包括定义每个模式的主码；定义参照完整性；定义与应用有关的完整性（取值范围等）。
4. 难点：RDBMS如何实现参照完整性的策略，即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，RDBMS应该如何进行处理，以确保数据的正确与有效。其中比较复杂的是参照完整性的实现机制。