考试范围需知

分值分布

1. 判断 10个 10分
2. 选择 10个 20分
3. 填空 10个 20分
4. 分析 （关系代数，sql语句）10分 + 20分
5. 综合 设计关系数据库的模式（画E-R图） 20分

信安方：

1、4、5、6、10、11章考选择，填空，判断；

2、3、7章考大题（关系代数、SQL、概念和逻辑设计）

大数据方：

判断：1、2、4、5、6、7、10、11章

填空：1、4、5、7、10、11

选择：1、2、4、5、6、7、10、11

分析：2.4（关系代数）、3（SQL语句）

综合：7（数据库设计）

计科1-4方：

1、2、3、4、5、6、7、10、11章出题

6.2.7多值依赖、6.2.8 4NF不考、标※不考

计科5-8方：

添加重点：2.4关系代数、6.2.3范式、7数据库设计、11.2封锁

你需要知道

数据库系统的大部分名词、能够进行关系代数运算、能够熟练写出SQL语句、能够设计数据库

约定

重点加粗，必考备注，明文不考不写，未明文不考会添入。

出题大概率是课后习题，推荐刷一遍。**习题答案添加在目录内**

此文档参考意义不大，基本等于方便我自己过一遍知识点。

因为数据库知识点量太大，书上均以加粗标注。

不推荐再看文档，直接去看书刷题，为此暂停更新 SQL

第一章 绪论

（判断+填空+选择 需要了解很多概念）

1.1数据库系统概述

1. 数据库的4个基本概念
    1. 数据（data）
        - **数据**是**数据库**中存储的**基本对象。**
        - 描述事物的**符号记录**称为**数据。**
        - 数据的**含义**称为数据的**语义**，数据与其语义是**不可分的。**
    2. 数据库（Database，DB）
        - 数据库是长期储存在**计算机内**、**有组织的**、**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、描述和储存。
        - 具有**较小的冗余度**、**较高的数据独立性**和**易扩展性**，并可为各种用户共享。
    3. 数据库管理系统（DataBase Management System，DBMS）
        - 主要功能：
            1. 数据定义功能
            2. 数据组织、存储和管理
            3. 数据操纵功能
            4. 数据库的事务管理和运行管理
            5. 数据库的建立和维护功能
            6. 其他功能
    4. 数据库系统（DataBase System，DBS）
        - 数据库系统是由**数据库（DB）、数据库管理系统（DBMS）、应用程序和数据库管理员（DBA）**组成的**存储、管理、处理和维护数据**的系统。
2. 数据库系统的特点
    1. 数据结构化
    2. 数据的共享性高、冗余度低且易扩充
    3. 数据独立性高
        - 包含**物理独立性和逻辑独立性**
        - 数据独立性由数据库管理系统提供的**二级映像**功能来保证的。
    4. 数据由数据库管理系统统一管理和控制

1.2数据模型

数据模型也是一种模型，它是对现实世界数据特征的抽象。

1. 数据模型分类
    1. 概念模型
    2. 逻辑模型和物理模型
2. 概念模型

    基本概念
    1. 实体：客观存在并可相互区别的事物称为实体。
    2. 属性：实体所具有的某一特性称为属性。
    3. 码：唯一标识实体的**属性集**称为码。
    4. 实体型：用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。
    5. 实体集：同一类型实体的集合称为实体集。
    6. 联系
3. **数据模型的组成要素**
    1. **数据结构**

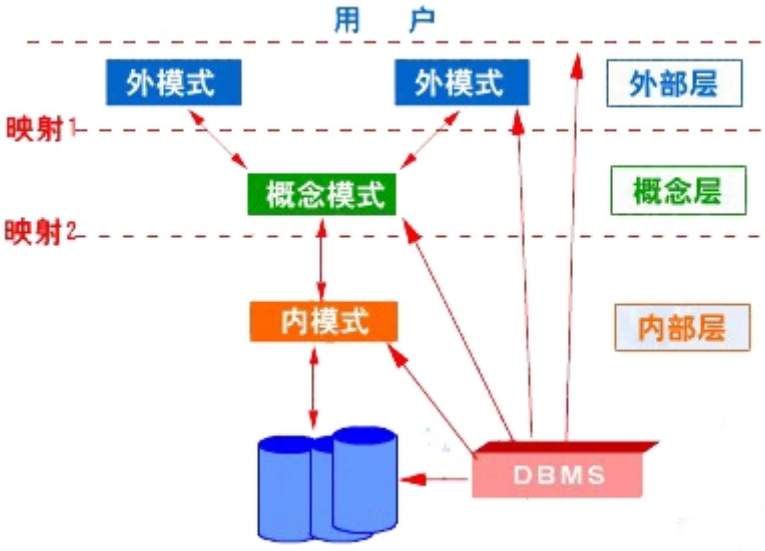
2. **数据操作**
3. **数据的完整性约束条件**

4. 常用的数据模型
    1. 层次模型：数据结构为**有向树**
    2. 网状模型：数据结构为**有向图**
    3. 关系模型：数据结构为**二维表**
    4. 面向对象数据模型、对象关系数据模型、半结构化数据模型

1.3模式

- **模式是相对稳定的，而实例是相对变动的。**

1. **三级模式结构**
    1. **外模式（external schema）**：外模式也称子模式或用户模式，它是数据库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。（用户层，应用搭建在这一模式下）
    2. **模式（schema）**：模式也称逻辑模式，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。（中间层，数据库管理员使用这一层）
    3. **内模式（inner schema）**：内模式也成存储模式，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组织方式。（物理层，决定数据存储方式）
2. **二级映像**
    1. **外模式/模式映像**
    2. **模式/内模式映像**



第二章 关系数据库

（判断+选择+大题 没有填空）概念不需要背很多 大题去做课后习题

2.1关系数据结构

1. 关系模型由**关系数据结构、关系操作集合和关系完整性约束**三部分组成。（区别数据模型）
2. 关系模式：R(U,D,DOM,F) R为关系名、U为组成该关系的属性名集合、D为U中属性所来自的域、DOM为属性向域的映像集合、F为属性间数据的依赖关系集合。
3. 关系数据库管理系统（RDBMS）

2.2关系操作

1. 基本的关系操作
    1. **5种基本关系**操作：并$(\cup)$、差$(-)$、笛卡尔积(X)、选择(σ)、投影(Π)
    2. 其他三种关系操作：交$(\cap)$、除$(\div)$、连接(∞)
2. 关系数据语言的分类
    1. 关系代数
    2. 关系演算
    3. 结构化查询语言（SQL）

    共同特点：具有完备的表达能力（关系完备性）

2.3关系的完整性

1. **实体完整性**：主属性不能取空值。
2. **参照完整性**：外码属性在对应主码表一定存在。
3. **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，被称为关系的**两个不变性。**

第三章 SQL

（只考大题 会写SQL语句就行）

3.1SQL概述

1. SQL的特点
    1. 综合统一
    2. 高度非过程化
    3. 面向集合的操作方式
    4. 以同一种语法结构提供多种使用方式
    5. 语言简洁、易学易用
2. SQL集**数据查询、数据操纵、数据定义和数据控制**功能于一体。

3.3数据定义

1. 模式
    - 定义模式：create schema <模式名> authorization <用户名>
    - 删除模式：drop schema <模式名> <cascade | restrict> (前者级联 一次性把相关的对象全删完，后者则是若存在还有其他对象则停止删除)
2. 表

- 定义基本表：create table <表名> (<列名> <数据类型> [完整性约束],...)
  - 完整性约束关键字有：
    1. primary key // 主键
    2. unique // 唯一
    3. not null // 非空
    4. check(约束条件) // 校验
       - 约束条件举例：
       - a <= b and b <= c // 控制值域范围
       - a in (1,2,4) // 控制值域集合
    5. foreign key // 外键 应该不考
       - references + 表名（列名） // 添加外键

```
-- 如构建学生选课表
create table SC(Cnum int,Snum int,Grade int,
                primary key(Cnum,Snum),
                foreign key(Cnum) references Course(Cnum),
                foreign key(Snum) references Student(Snum));
```

  - 数据类型有：
    1. char(n) // 定长字符串
    2. varchar(n) // 变长最大长度为n字符串
    3. int // 4字节整数
    4. bigint // 8字节整数
    5. real // 单精度浮点数
    6. dec(n,p) // 定点数，n位数 保留p为小数
    7. date // YYYY-MM-DD 格式日期
    8. time // 精确到S的日期 （最多考前5个）
- 修改基础表：alter table <表名> [操作]
  - 操作有：
    1. add column <列名> <数据类型> [完整性约束] // 添加新列
    2. add <表级约束> 如： primary key(a,b) 把a，b定为主键

       其他我感觉不考，考了就开摆。
- 删除基础表：drop table <表名> [cascade | restrict] (参数含义同上)

3. 索引（我觉得不考）

3.4数据查询

（必考，最基础的东西不写了 做题吧）

1. 模糊查询：a **like** b 如： name like '%[映,宇]_'
   - % 多字匹配>=0，_ 单字匹配 = 1
   - 找到叫 什么映，宇 且后面还有一个字的人。
   - 比如 李映飞、黄宇轩。 // 叫黄宇的人就不行，宇文宇轩也可以
2. 去重：select后面接dist（或distinct）关键字
3. 聚集函数：

    1. count(列名 or *) // * 不管null照样记录， 列 只记录非null
    2. sum(列) // 求和
    3. avg(列) // 求平均
    4. max(列) 、min(列) // 最大 最小值
4. 分组查询： group by
    1. where筛选元组
    2. having筛选分组（having基本是使用聚集函数筛选
5. 排序：order by
    1. asc 列名,... 升序 // 默认
    2. desc 列名,... 降序

       多列排序 就是先看第一个参数，如果第一个相等就继续下一个
6. 外连接：[left,right] outer join <表名> on <条件> // 我感觉也不会考这么难
    - 默认不写就是内连接，加上outer就是外连接。可以把左或右悬浮元组
7. in子查询：判断某个属性是否在子查询的集合内。
8. any子查询：判断集合内是否存在一个元素使得属性条件满足。 （如： a >= any(子查询)）
9. all子查询：判断集合内所有元素都使得属性条件满足。（如： a >= all(子查询)）
10. exists子查询（比较难，需要自己做题
    1. 一般不用exists，主要用not exists把一个命题变成双重否定
    2. 例子：

```
// 查询出学过2005010601学过所有课程的学生
// 不存在一门2005010601学过的课程，该学生没学过
select s1.number from grades s1 group by s1.number having not exists(
select * from grades s2 where s2.number = '2005010601' and not
exists(
select * from grades s3 where s3.c_id = s2.c_id and s3.number =
s1.number));
```

11. 集合查询：把多个查询连接起来。（基本不会考）
    1. union（并）
    2. intersect（交）
    3. except（差）

3.5数据更新

1. 插入数据：insert into <表名> [(<列名>,...)] values(<v1>,...) // 不写（列名) 就是默认创建的顺序，若加入列表有列没写就插入null
2. 更新数据：update <表名> set <列名> = <表达式> [where <条件>]
3. 删除数据：delete from <表名> [where <条件>]

3.7视图

视图和基础表的操作基本一致，因为视图也是表。

1. 创建视图：create view <视图名> [(<列名>,...)] as (子查询) [with check option]

- with check option：加入后，要求修视图数据进行增删改的时候，要满足子查询的where条件

第四章 数据库安全性

（判断、填空、选择都考，要背蛮多定义 核心在4.2 4.3）

4.1 数据库安全性概述

1. 不安全的因素
    1. 非授权用户对数据库的恶意存取和破坏。
    2. 数据库中重要或敏感的数据被泄露。
    3. 安全环境的脆弱性。

4.2数据库安全性控制

1. 用户身份鉴定
    1. 静态口令鉴别
    2. 动态口令鉴别
    3. 生物特征鉴别
    4. 智能卡鉴别
2. 存取控制
    1. 定义用户权限
    2. 合法权限检查
3. 自主存取控制方法
    1. 用户权限的两个要素：**数据库对象和操作类型。**
    2. **定义存取权限称为授权。**
4. **授权：授予与收回**
    1. 权限有：

表 4.3 关系数据库系统中的存取权限

对象类型	对象	操作类型
数据库模式	模式	CREATE SCHEMA
	基本表	CREATE TABLE，ALTER TABLE
	视图	CREATE VIEW
	索引	CREATE INDEX
数据	基本表和视图	SELECT， INSERT， UPDATE， DELETE， REFERENCES， ALL PRIVILEGES
	属性列	SELECT，INSERT，UPDATE，REFERENCES，ALL PRIVILEGES

    2. grant关键字：授权
        1. 格式：grant <权限>,[...] on table/schema <表名/模式名> ,[...] to <用户>,[...] [with grant option]
        2. grant授权可以把多个对象授予给多个用户，[...]就是多个的省略
        3. with grant option关键字让拥有权限者可以分配给其他人
        4. 例子：

```
grant all privileges on *.* to username with grant option;
-- *通配符，把所有数据库的所有表的所有权限都交给username
```

3. revoke关键字：收回权限
    1. 格式：revoke <权限>,[...] on table/schema <表名/模式名> ,[...] from <用户>,[...] [cascade | restrict]
    2. revoke一样可以把多个用户的多个对象的多个权限进行收回
    3. cascade递归收回，restrict若权限转交给其他用户则收回失败。
    4. 例子：

```
revoke all privileges on *.* from username cascade;
-- 递归收回所有权限
```

4. 数据库角色
    1. 数据库角色是被命令的一组与数据库操作相关的权限，**角色是权限的集合。**
    2. 角色创建：create role <角色名>
    3. 角色授权：grant <权限> on <表/模式> to <角色名>
    4. 角色授予给其他的角色或用户：grant <角色名> to <角色名/用户名> [with admin option]
    5. 角色权限收回：revoke <权限> on <表/模式> from <角色名>

4.3视图机制

（按照范围的话，这个必考）

为不同的用户定义不同的视图，把数据对象限制在一定的范围内。

通过视图机制把要保密的数据对无权存取的用户隐藏起来，从而自动对数据提供一定程度的安全保护。

视图机制**间接地**实现支持**存取谓词的用户权限定义。**

审计、数据加密看命考。

第五章 数据库完整性

不考填空，判断+选择

前面都有相关内容。（

数据库的完整性是指数据的**正确性和相容性。**

5.6断言

SQL可以使用数据定义语言的create assertion语句，声明性**断言**来指定更具一般性的约束。

任何对断言中所涉及的**关系的操作**都会触发关系数据库管理系统对断言的检查，任何使断言不为真值的操作都会被拒绝。

1. 创建断言：create assertion <断言名> <check 子句>

```sql
-- 例子
-- 限制数据库课程最多60名学生选修
create assertion asse_sc_db_num
    check (60 >= (select count(*) from Course,SC
                  where SC.CNO = Course.CNO and Course.Cname = '数据库'));
```

2. 删除断言：drop assertion <断言名>

第六章 关系数据理论

只有判断+选择题（要了解很多性质，不需要背填写

关键在于6.2 6.3

6.1问题的提出

产生的问题：

1. 数据冗余
2. 更新异常
3. 插入异常
4. 删除异常

6.2规范化

1. 码
    1. 候选码：若K完全函数确定属性集U，则称K为R的候选码。
    2. 主属性：包含在任何一个候选码中的属性称为主属性。
2. 范式：一个低一级范式的关系模式通过**模式分解**可以转换为若干个高一级范式的关系模式的集合，这种过程就叫**规范化。**
3. 1NF：所有字段值都是不可分解的原子值。
4. 2NF：每一个非主属性完全函数依赖于任何一个候选码。
5. 3NF：每一个非主属性既不传递依赖于码，也不部分依赖于码。
6. BCNF：所有非主属性对每一个码都是完全函数依赖、所有主属性对于每一个不包含它的码也是完全函数依赖、所有属性的决定因素都包含码

6.3公理系统

定义：对于关系模式R<U,F>，遵循以下推理规则：

1. 自反律：A->A
2. 增广律：A->B 则 AC -> BC
3. 传递律： A->B and B->C 则 A -> C

推理规则：

1. 合并规则：X->Y,X->Z 有X->YZ
2. 伪传递规则：X->Y,WY->Z 有XW -> Z

3. 分解规则：X->Y 及 Z 部分函数依赖Y ， 则X -> Z

第七章 数据库设计

（所有题型都考，大题部分刷题即可）

7.1数据库设计概述

1. 设计步骤：
    1. **需求分析阶段**
    2. **概念结构设计阶段**
    3. **逻辑结构设计阶段：确定数据模型**
    4. **物理结构设计阶段**
    5. **数据库实施阶段**
    6. **数据库运行和维护阶段**

7.3概念模型设计

1. 主要特点：
    1. 能真实、充分地反映现实世界。
    2. 易于理解。
    3. 易于更改。
    4. 易于向关系、网状、层次等各种数据模型转换。
2. E-R模型是描述概念模型的有力工具。
    1. 实体间的联系（有三种，1对1，1对多，多对多）
    2. 把参与联系的实体型的数目称为联系的度。如：两个实体型之间的联系度为2
3. E-R图
    1. 实体型用矩阵表示
    2. 属性用椭圆形表示
    3. 联系用菱形表示

7.4逻辑结构设计

逻辑结果设计的任务就是把概念结构设计阶段设计好的**基本E-R图转换为**选用数据库系统产品所支持的数据模型相符合的**逻辑结构。**

（主要涉及E-R转关系模型，直接看书做题吧）

7.5物理结构设计

为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程，就是数据库的物理设计。

分为：

1. 确定数据库的物理结构。
2. 对物理结构进行评价

第十章 数据库恢复技术

要考 判断 填空 选择 多背概念

10.1事务

1. 定义

   事务是用户定义的一个**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。

2. 事务的ACID特性

   1. **原子性（Atomicity）**
   2. **一致性（Consistency）**
   3. **隔离性（Isolation）**
   4. **持续性（Durability）**

10.3故障

1. 种类
   1. 事务内部的故障
   2. 系统故障
   3. 介质故障
   4. 计算机病毒
2. 对数据库的影响
   1. **数据库本身被破坏**
   2. **数据库没有被破坏，但数据可能不正确。**
3. 恢复的基本原理：**冗余**

10.4恢复

建立冗余数据最常用的技术是**数据转储**和**登记日志文件。**

1. 数据转储
   - 概念：管理员定期地将整个数据库复制到磁带、磁盘或其他存储介质上保存起来的过程，这些备用的数据称为**后备副本**或后援副本。
   - 局限性：数据库遭到破坏后可以将后备副本重新装入，但**重装后备副本只能将数据库恢复到转储时的状态。**
   - 转储分为**静态转储和动态转储。**
   - 转储还可分为**海量转储和增量转储。**
   - 数据转储方法就可以分为四类：**动态海量转储、动态增量转储、静态海量转储、静态增量转储。**
2. 登记日志文件
   - 概念：日志文件是用来记录事务对数据库的更新操作的文件。
   - 日志文件内容：
     - 各个事务的开始
     - 各个事务的结束
     - 各个事务的所有更新操作
   - 日志记录的内容：

- 事务标识
- 操作的类型
- 操作对象
- 更新前数据的旧值
- 更新后数据的新值

小结：**保证数据一致性是对数据库的最基本要求。**

第十一章 并发控制

要考 判断、填空、选择 要多背概念

11.1并发控制

1. **事务是并发控制的基本单位。**
2. **保证事务的隔离性和一致性。**
3. 并发操作带来的数据不一致性有：
    1. **丢失修改**
    2. **不可重复读**
    3. **读“脏”数据**
4. 并发控制的主要技术有：
    1. **封锁**
    2. **时间戳**
    3. **乐观控制法**
    4. **多版本并发控制**

11.2封锁

1. 锁
    1. **排他锁：又称为写锁，对事务T加上X锁，只允许T读取和修改A，直到锁释放为止。**
    2. **共享锁：又称为读锁，对事务T加上S锁，事务T可以读但不能修改A，其他事务只能再对A加S锁不能加X锁。直到A的所有S锁都释放后，才能加X锁。**
2. 一级封锁协议：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。**
    - 可以防止丢失修改，并保证事务T是可恢复的。
    - 如果仅仅是读数据而不对其进行修改，是不需要加锁的，所以他**不能保证可重复读和不读“脏”数据。**
3. 二级封锁协议：在一级封锁协议的基础上，增加事务T在**读取数据R之前必须对其加S锁，读完后即可释放S锁。**
    - 可以防止丢失修改和读“脏”数据
    - 但不能保证可重复读
4. 三级封锁协议：在一级封锁协议的基础上，增加事务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。**
    - 解决了所有问题

11.3活锁和死锁

1. 活锁：事务永远等待，就是活锁情况。
2. 解决方法：采用**先来先服务**的策略。
3. 死锁：多个事务相互等待，使得事务永远不能结束。
4. 解决方法：**①预防死锁②死锁解除**
5. 死锁诊断一般使用超时法或事务等待图法

小结：数据库管理系统必须提供并发控制机制来协调并发用户的并发操作以保证并发事务的隔离性和一致性，保证数据库的一致性