

数据库系统概论复习资料

第一章

一、名词解释

1. Data: 数据, 是数据库中存储的基本对象, 是描述事物的符号记录。
2. Database: 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。
3. DBMS: 数据库管理系统, 是位于用户与操作系统之间的一层数据管理软件, 用于科学地组织、存储和管理数据、高效地获取和维护数据。
4. DBS: 数据库系统, 指在计算机系统中引入数据库后的系统, 一般由数据库、数据库管理系统、应用系统、数据库管理员 (DBA) 构成。
5. 数据模型: 是用来抽象、表示和处理现实世界中的数据和信息的工具, 是对现实世界的模拟, 是数据库系统的核心和基础; 其组成元素有数据结构、数据操作和完整性约束。
6. 概念模型: 也称信息模型, 是按用户的观点来对数据和信息建模, 主要用于数据库设计。
7. 逻辑模型: 是按计算机系统的观点对数据建模, 用于 DBMS 实现。
8. 物理模型: 是对数据最底层的抽象, 描述数据在系统内部的表示方式和存取方法, 在磁盘或磁带上的存储方式和存取方法, 是面向计算机系统的。
9. 实体和属性: 客观存在并可相互区别的事物称为实体。实体所具有的某一特性称为属性。
10. E-R 图: 即实体-关系图, 用于描述现实世界的事物及其相互关系, 是数据库概念模型设计的主要工具。
11. 关系模式: 从用户观点看, 关系模式是由一组关系组成, 每个关系的数据结构是一张规范化的二维表。
12. 型/值: 型是对某一类数据的结构和属性的说明; 值是型的一个具体赋值, 是型的实例。
13. 数据库模式: 是对数据库中全体数据的逻辑结构 (数据项的名字、类型、取值范围等) 和特征 (数据之间的联系以及数据有关的安全性、完整性要求) 的描述。
14. 数据库的三级系统结构: 外模式、模式和内模式。
15. 数据库内模式: 又称为存储模式, 是对数据库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。一个数据库只有一个内模式。
16. 数据库外模式: 又称为子模式或用户模式, 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。通常是模式的子集。一个数据库可有多个外模式。
17. 数据库的二级映像: 外模式/模式映像、模式/内模式映像。

二、填空题

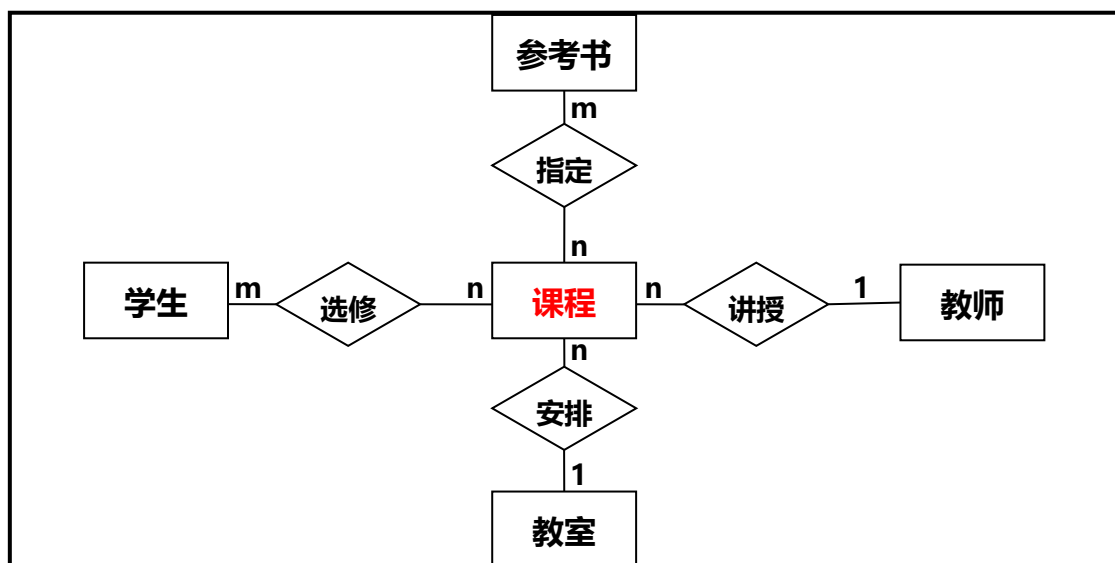
1. 数据库系统由数据库、数据库管理系统、应用系统和数据库管理员构成。
2. 数据管理是指对数据进行分类、组织、编码、存储、检索和维护等活动。
3. 数据管理技术的发展经历了人工管理、文件系统、数据库系统 3 个阶段。
4. 数据模型分为概念模型、逻辑模型和物理模型 3 个方面。
5. 数据模型的组成要素是: 数据结构、数据操作、完整性约束条件。
6. 实体型之间的联系分为一对一、一对多和多对多三种类型。

7. 常见的数据模型包括：关系、层次、网状、面向对象、对象关系映射等几种。
8. 关系模型的完整性约束包括：实体完整性、参照完整性和用户定义完整性。

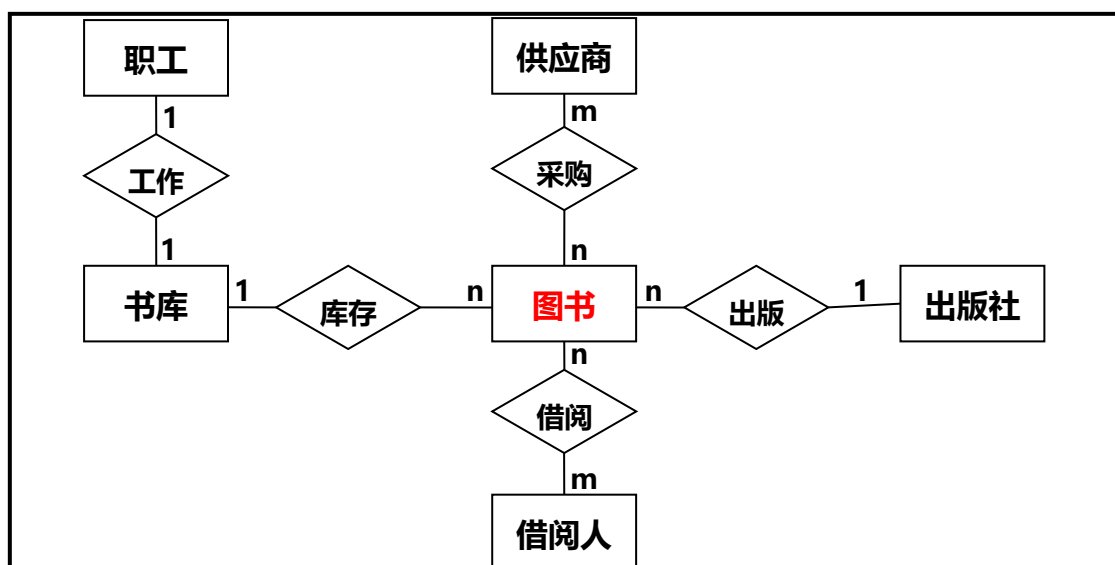
三、问答题

1. 简述 DBMS 的主要功能
 - 数据定义
 - 数据组织、存储和管理
 - 数据操纵(查询、插入、删除和修改)
 - 数据库的事务管理和运行管理
 - 数据库的建立和维护功能
 - 其它功能
2. 简述人工数据管理阶段的特点
 - 由程序员管理，数据不保存
 - 应用程序管理数据
 - 数据不共享、冗余度极大
 - 数据不具独立性，完全依赖于程序
3. 简述文件系统阶段数据管理的特点
 - 数据由文件系统管理，可长期保存
 - 数据面向某一应用程序，共享性差、冗余度大
 - 数据独立性差，数据的逻辑结构改变必须修改应用程序
4. 简述数据库系统阶段数据管理的特点
 - 数据结构化
 - 数据的共享性高，冗余度低，易扩充
 - 数据独立性高
 - 数据由 DBMS 统一管理和控制
5. DBMS 可以对数据提供哪些控制功能？
 - 数据的安全性（Security）保护：保护数据，以防止不合法的使用造成的数据的泄密和破坏。
 - 数据的完整性（Integrity）检查：将数据控制在有效的范围内，或保证数据之间满足一定的关系。
 - 并发（Concurrency）控制：对多用户的并发操作加以控制和协调，防止相互干扰而得到错误的结果。
 - 数据库恢复（Recovery）：将数据库从错误状态恢复到某一已知的正确状态。
6. 何谓数据模型？它应满足哪些要求？
 - 数据模型是用来抽象、表示和处理现实世界中的数据和信息的工具，是对现实世界的模拟。
 - 数据模型应满足三方面要求：
 - 1) 能比较真实地模拟现实世界
 - 2) 容易为人所理解
 - 3) 便于在计算机上实现

7. 何谓数据结构？描述的内容是什么？
- 数据结构是对数据库的组成对象以及对象之间的联系的描述。
 - 描述的内容包括与对象的类型、内容、性质有关的和与数据之间联系有关的对象。
8. 举例说明一对一、一对多和多对多 3 种联系。
- 一个班级只能有一个正班长，一个班长只能在一个班级中任职。班长和班级两个实体集之间就是一对一的联系。
 - 一个班级可以有很多学生，一个学生只能在一个班级中就读。班级和学生两个实体集之间就是一对多的联系。
 - 一门课程可以被多个学生选修，一个学生可以选修多门课程。课程和学生两个实体集之间就是多对多的联系。
9. 分析并画出某教学管理系统的 E-R 图(可不画出属性)。主要实体型包括：学生、教师、课程、参考书、教室。



10. 分析并画出某图书馆管理系统的 E-R 图(可不画出属性)。主要实体型包括：图书、出版社、经销商、借阅人、书库、职工。



11. 简述层次模型的基本特点：

- 结点的双亲是唯一的
- 只能直接处理一对多的实体联系
- 每个记录类型可以定义一个排序字段，也称为码字段
- 任何记录值只有按其路径查看时，才能显出它的全部意义
- 没有一个子女记录值能够脱离双亲记录值而独立存在

12. 简述层次模型的优缺点。

优点：

- 层次模型的数据结构比较简单清晰
- 查询效率高，性能优于关系模型，不低于网状模型
- 层次数据模型提供了良好的完整性支持

缺点：

- 多对多联系表示不自然
- 对插入和删除操作的限制多，应用程序的编写比较复杂
- 查询子女结点必须通过双亲结点
- 由于结构严密，层次命令趋于程序化

13. 阐述数据库三级模式、二级映象的含义及作用。

- 数据库三级模式反映的是数据的三个抽象层次：模式是对数据库中全体数据的逻辑结构和特征的描述。内模式又称为存储模式，是对数据库物理结构和存储方式的描述。外模式又称为子模式或用户模式，是对特定数据库用户相关的局部数据的逻辑结构和特征的描述。
- 数据库三级模式通过二级映象在 DBMS 内部实现这三个抽象层次的联系和转换。外模式面向应用程序，通过外模式/模式映象与逻辑模式建立联系，实现数据的逻辑独立性。模式/内模式映象建立模式与内模式之间的一对一映射，实现数据的物理独立性。

14. 阐述 DBA 的职责。

- 决定数据库中的信息内容和结构
- 决定数据库的存储结构和存取策略
- 决定数据的安全性要求和完整性约束条件
- 监控数据库的使用和运行
- 数据库的改进和重组重构
- 定期对数据库进行重组织，以提高系统的性能

第二章

一、名词解释

1. 主码：能够唯一地标识一个元组的属性或属性组称为关系的码或候选码。若一个关系有多个候选码则可选其一作为主码或主键(Primary key)。
2. 外码：如果一个关系的一个或一组属性引用(参照)了另一个关系的主码，则称这个或这组属性为外码或外键(Foreign key)。
3. 关系数据库：依照关系模型建立的数据库称为关系数据库。它是在某个应用领域的所有关系的集合。
4. 关系模式：简单地说，关系模式就是对关系的型的定义，包括关系的属性构成、各属性的数据类型、属性间的依赖、元组语义及完整性约束等。关系是关系模式在某一时刻的状态或内容，关系模型是型，关系是值，关系模型是静态的、稳定的，而关系是动态的、随时间不断变化的，因为关系操作在不断地更新着数据库中的数据。
5. 实体完整性：用于标识实体的唯一性。它要求基本关系必须要有一个能够标识元组唯一性的主键，主键不能为空，也不可取重复值。
6. 参照完整性：用于维护实体之间的引用关系。它要求一个关系的外键要么为空，要么取与被参照关系对应的主码值，即外码值必须是主码中已存在的值。
7. 用户定义的完整性：就是针对某一具体应用的数据必须满足的语义约束。包括非空、唯一和布尔条件约束三种情况。

二、填空题

1. 在学生(学号, 姓名, 性别, 年龄, 班长学号)、课程(课程号, 课程名, 学时数)、选修(学号, 课程号, 成绩)三个关系中：
 - 选修关系的主码是学号+课程号，外码学号参照学生关系的主码，外码课程号参照课程关系的主码。
 - 学生关系的主码是学号，外码是班长学号，参照学生关系的主码。
2. 关系的三类完整性约束分别是实体完整性、参照完整性和用户定义的完整性。
3. 关系数据库语言分为关系代数、关系演算和结构化查询语言三大类。
4. 关系的 5 种基本操作是选择、投影、并、差、笛卡尔积。

三、问答题

1. 简述基本关系的性质
 - 每个列具有相同的数据类型
 - 不同的列可出自同一个域，但必须是不同的名字
 - 列的顺序可以任意交换
 - 行的顺序也可以任意交换
 - 至少有一个候选码，用于标识元组的唯一性
 - 分量必须取原子值，即属性不能再分

2. 写出关系模式的五元组形式化表示并说明各符号的含义。

关系模式是对关系的描述，形式化表示为： $R(U, D, DOM, F)$

- R —— 关系名
- U —— 组成该关系的属性名集合
- D —— 属性组 U 中属性所来自的域
- DOM —— 属性向域的映象集合
- F —— 属性间的数据依赖关系集合

3. 简述教材“学生-课程”示例数据库三个关系(表)的主外键及参照情况。

学生 (学号, 姓名, 性别, 专业号, 年龄)

课程 (课程号, 课程名称, 学分)

选修 (学号, 课程号, 成绩)

- 学号是学生关系的主键，课程号是课程关系的主键，选修关系的主键是学号+课程号属性组。选修关系有两个外键，其中学号外键参照学生关系的主键(学号)，课程号外键参照课程关系的主键(课程号)。选修关系属于参照关系，学生关系和课程关系以于是被参照关系。

4. 设 R 是包含 k_1 个元组的 n 目关系， S 是包含 k_2 个元组的 m 目关系，写出 R 与 S 的笛卡尔积运算公式并说明其含义。

- 运算公式为： $R \times S = \{ \overset{\frown}{t_r t_s} \mid t_r \in R \wedge t_s \in S \}$

- 运算结果为一个 $(n+m)$ 目关系，其每个元组的前 n 列是关系 R 的一个元组，后 m 列是关系 S 的一个元组。
- 运算结果包含由 $k_1 \times k_2$ 个元组，由来自 R 和 S 的所有元组交叉组合而成

5. 写出关系选择(σ)运算的公式并说明其含义。

- 运算公式为： $\sigma_F(R) = \{ t \mid t \in R \wedge F(t) = \text{‘真’} \}$

- 选择运算的含义为从关系 R 中找出符合条件的所有元组。式中： F 为选择条件， t 表示元组，它是 R 元组的子集，并使逻辑表达式 $F(t)$ 为真。

6. 写出关系投影(π)运算的公式并说明其含义。

- 运算公式为： $\pi_A(R) = \{ t[A] \mid t \in R \}$

- 投影运算的含义为从关系 R 中选择出若干属性列组成新的关系。式中 A 为 R 的部分属性组， t 表示元组， $t[A]$ 表示由属性组 A 上的分量构成的元组。

第三章

一、名词解释

- 1. SQL: 结构化查询语言的简称，是关系数据库的标准语言。SQL 是一种通用的、功能极强的关系数据库语言，是对关系数据存取的标准接口，也是不同数据库系统之间互操作的基础。集数据查询、数据操作、数据定义、和数据控制功能于一体。
- 2. 数据定义: 数据定义功能包括模式定义、表定义、视图和索引的定义。
- 3. 嵌套查询: 指将一个查询块嵌套在另一个查询块的 WHERE 子句或 HAVING 短语的条件中的查询。

二、填空题

- 1. SQL 数据定义语句的操作对象有: 模式、表、视图和索引。
- 2. SQL 数据定义语句的命令动词是: CREATE、DROP 和 ALTER。
- 3. RDBMS 中索引一般采用 B+树或 HASH 来实现。
- 4. 索引可以分为唯一索引、非唯一索引和聚簇索引三种类型。
- 5. 填空完成下列查询条件表:

查询条件	运算符 / 谓词
比 较	=, >, <, >=, <=, !=, <>, !>, !<
确定范围	BETWEEN AND, NOT BETWEEN AND
确定集合	IN, NOT IN
字符匹配	LIKE, NOT LIKE
空 值	IS NULL, IS NOT NULL
多重条件（逻辑运算）	AND, OR, NOT

操作对象	操作方式		
	创建	删除	修改
模式	CREATE SCHEMA	DROP SCHEMA	
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	

三、问答题

- 1. 简述 SQL 的特点。
 - 集数据定义语言 (DDL)、数据操纵 (DML) 和数据控制语言 (DCL) 于一体，可以独立完成数据库生命周期中的全部活动。
 - 属于高度非过程化语言，只要告知“做什么”，而无须了解“如何做”。
 - 采用集合操作方式，操作对象、查询结果均为元组的集合，效率高。
 - 同一语法结构多种使用方式。既可用于联机交互操作，也可嵌入其它高级语言程序中使用。
 - 语言简单易用，完成核心功能只需 9 个动词。

2. 写出 SQL 创建表语句的一般格式并说明其含义。

CREATE TABLE <表名>

(<列名> <数据类型>[<列级完整性约束>]

[, <列名> <数据类型>[<列级完整性约束>]] ...

[, <表级完整性约束>]) ;

- <数据类型>可以是数据库系统支持的各种数据类型，包括长度和精度。
- 列级完整性约束为针对单个列(本列)的完整性约束，包括 PRIMARY KEY、REFERENCES 表名(列名)、UNIQUE、NOT NULL 等。
- 表级完整性约束可以是基于表中多列的约束，包括 PRIMARY KEY (列名列表)、FOREIGN KEY REFERENCES 表名(列名) 等。

3. 写出 SQL 创建索引语句的一般格式并说明其含义。

CREATE [UNIQUE] [CLUSTER] INDEX <索引名>

ON <表名> (<列名列表>) ;

- UNIQUE：表示创建唯一索引，缺省为非唯一索引。
- CLUSTER：表示创建聚簇索引，缺省为非聚簇索引。
- <列名列表>：一个或逗号分隔的多个列名，每个列名后可跟 ASC 或 DESC，表示升/降序，缺省为升序。多列时则按为多级排序。

4. 写出 SQL 查询语句的一般格式并说明其含义。

SELECT [ALL | DISTINCT] <算术表达式列表> FROM <表名或视图名列表>

[WHERE <条件表达式 1>]

[GROUP BY <属性列表 1> [HAVING <条件表达式 2>]]

[ORDER BY <属性列表 2> [ASC | DESC]] ;

说明：

- ALL / DISTINCT：缺省为 ALL，即列出所有查询结果记录，包括重复记录。DISTINCT 则对重复记录只列出一条。
- 算术表达式列表：一个或多个逗号分隔的算术表达式，表达式由常量(包括数字和字符串)、列名、函数和算术运算符构成。每个表达式后还可跟别名。也可用*代表查询表中的所有列。
- <表名或视图名列表>：一个或多个逗号分隔的表或视图名。表或视图名后可跟别名。
- 条件表达式 1：包含关系或逻辑运算符的表达式，代表查询条件。
- 条件表达式 2：包含关系或逻辑运算符的表达式，代表分组条件。
- <属性列表 1>：一个或逗号分隔的多个列名。
- <属性列表 2>：一个或逗号分隔的多个列名，每个列名后可跟 ASC 或 DESC，表示升/降序，缺省为升序。

5. 自己认真调试教材“学生-课程”示例数据库(含完整性约束定义)。**P82 表**

6. 写出插入单个记录的 SQL 语句格式并说明其含义。

- 语句格式：

INSERT INTO <表名> [(<属性列表>)]

VALUES (<常量列表>)

- 说明：

INTO 子句的<属性列表>可为表中的全部属性列，也可为非空部分属性，顺序不限，以逗号分隔。缺省为表中的所有属性。

VALUES 子句的<常量列表>由逗号分隔的常量构成，其个数、类型和顺序必须与属性列表一致。

7. 简述视图的作用。

- 能够简化用户的操作：简化用户的查询操作，用户注意力可集中在自己关心的数据上。
- 使用户能以多种角度看待同一数据：不同的用户可以从不同的角度看待同一数据。
- 对重构数据库提供了一定程度的逻辑独立性：在数据库三级模式结构中，视图属于外模式，因而可保证数据的逻辑独立性。
- 能够对机密数据提供安全保护：视图机制不仅具备受限更新，更可向特定用户屏蔽或开放机密数据。
- 适当的利用视图可以更清晰的表达查询。

第四、五章

一、名词解释

1. 数据库角色：被命名的一组与数据库操作相关的权限，是权限的集合，通过角色授权可简化授权过程。用 **CREATE ROLE** 语句创建角色，然后用 **GRANT** 语句给角色授权。
2. 自主存取控制(DAC)：通过将不同数据库对象的不同操作权限授予不同的用户，来实现数据安全性控制的一种机制。在这种机制下，DBA 和属主用户可以将对象的操作权限自主授予他人，也可随时回收权限。
3. 强制存取控制(MAC)：是建立在 DAC 之上的安全验证机制。这种机制分别对主体和客体施加敏感度标记(许可证和密级)，标记与数据是一个不可分的整体，数据的副本与正本一样拥有密级保护。
4. 数据加密：是根据一定的算法将原始数据(明文)变换为不可直接识别的格式(密文)的一种数据保护手段，加密可以使不知道解密算法的人无法获知数据的内容，从而防止数据在存储和传输过程中失密。
5. 触发器：是用户定义在基本表上的一类由事件驱动的特殊过程。由服务器自动激活，能执行更为复杂的检查和操作，具有更精细和更强大的数据控制能力。使用 **CREATE TRIGGER** 命令建立触发器。

二、填空题

1. 计算机系统存在技术安全、管理安全和政策法律三类安全性问题。
2. TCSEC/TDI 标准由安全策略、责任、保证和文档四个方面内容构成。
3. 常用存取控制方法包括自主存取控制(DAC)和强制存取控制(MAC)两种。
4. 自主存取控制(DAC)的 SQL 语句包括 **GRANT** 和 **REVOKE** 两个。用户权限由数据对象和操作类型两部分构成。
5. 填空，完成以下权限表。

权限	可否执行的操作
----	---------

	CREATE USER	CREATE SCHEMA	CREATE TABLE	CRUD 操作
DBA	可以	可以	可以	可以
RESOURCE	不可以	不可以	可以	可以
CONNECT	不可以	不可以	不可以	可以 但必须被授权

6. 强制存取控制(MAC) 将实体分为主体和客体两大类。
7. MAC 的敏感度标记分为绝密、机密、可信和公开四个安全级别。
8. 数据加密的方法大致分为替换法、置换法和混合法三种类型。
9. 参照完整性违约处理的方式包括拒绝执行、级连操作和置空值三种策略。

三、问答题

1. 数据库安全性控制的常用方法有哪些？
 - 用户标识和鉴定：是数据库访问控制的最外层安全保护措施，通过用户标识和口令实现。
 - 存取控制：通过用户权限定义和合法权限检查机制共同构成 DBMS 的安全子系统
 - 视图：可在一定程度上对保密数据提供隐藏保护。
 - 审计：用于记录用户对数据库的所有操作以便事后跟踪和追溯。
 - 加密存储：对高度机密的数据进行数据加密，是防止数据在存储和传输过程中失密的有效手段。
2. 写出下列 SQL 自主权限控制命令。
 - 1) 把对 Student 和 Course 表的全部权限授予所有用户。
GRANT ALL PRIVILIGES ON TABLE Student, Course TO PUBLIC ;
 - 2) 把对 Student 表的查询权和姓名修改权授予用户 U4。
GRANT SELECT, UPDATE(Sname) ON TABLE Student TO U4 ;
 - 3) 把对 SC 表的插入权限授予 U5 用户，并允许他传播该权限。
GRANT INSERT ON TABLE SC TO U5 WITH GRANT OPTION ;
 - 4) 把用户 U5 对 SC 表的 INSERT 权限收回，同时收回被他传播出去的授权。
REVOKE INSERT ON TABLE SC FROM U5 CASCADE ;
 - 5) 创建一个角色 R1，并使其对 Student 表具有数据查询和更新权限。
CREATE ROLE R1;
GRANT SELECT, UPDATE ON TABLE Student TO R1;
 - 6) 对修改 Student 表结构的操作进行审计。
AUDIT ALTER ON Student ;
3. 简述可能破坏参照完整性的情况及违约处理方式。

被参照表（例如 student）	参照表（例如 SC）	违约处理
可能破坏参照完整性	←— 插入元组	拒绝
可能破坏参照完整性	←— 修改主码值	拒绝
删除元组 —→	可能破坏参照完整性	拒绝/级连删除/设置为空值
修改主码值 —→	可能破坏参照完整性	拒绝/级连删除/设置为空值

- 在参照表中插入或删除元组时可能会破坏参照完整性，DBMS 将拒绝执行。
 - 在被参照表中删除元组时也可能破坏参照完整性，DBMS 将选择拒绝删除或级连删除或设置空值几种方案处理。拒绝为默认策略。
 - 在被参照表中修改主码值时也可能破坏参照完整性，DBMS 将选择拒绝修改或级连修改或设置空值几种方案处理。拒绝为默认策略。
4. 写出创建触发器的一般语法并说明其含义。

```
CREATE TRIGGER <触发器名> { BEFORE | AFTER } <触发事件> ON <表名>
FOR EACH { ROW | STATEMENT } [ WHEN <触发条件> ]
AS BEGIN
<触发动作体>
END ;
```

说明：

- <触发事件>可以是 INSERT、DELETE 或 UPDATE [OF <列>，...>] 等数据更新操作
- BEFORE 表示在操作执行前触发动作体，AFTER 则在操作执行后触发动作体
- FOR EACH ROW 为行级触发器，操作涉及多少行就会执行多少次触发动作体
- FOR EACH STATEMENT 为语句级触发器，事件发生时触发动作体只执行一次
- <触发条件>是一个布尔表达式，表示事件发生时是否触发动作执行的附加条件

<触发动作体>为事件发生时执行的动作。可以是单个 SQL 语句或 PL/SQL 过程，也可以是存储过程调用。可以用 NEW 和 OLD 引用新值和原值。

第六、七章

一、名词解释

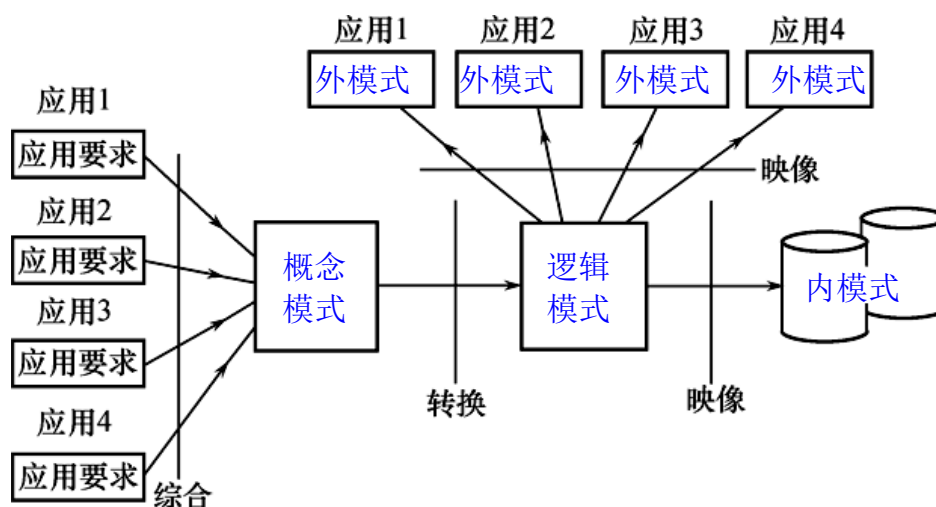
1. 数据依赖：反映一个关系内部属性与属性之间的约束关系，是现实世界属性间相互联系的抽象，属于数据内在的性质和语义的体现。
2. 规范化理论：是用来设计良好的关系模式的基本理论。它通过分解关系模式来消除其中不合适的数据依赖，以解决插入异常、删除异常、更新异常和数据冗余问题。
3. 函数依赖：简单地说，对于关系模式的两个属性子集 X 和 Y ，若 X 的任一取值能唯一确定 Y 的值，则称 Y 函数依赖于 X ，记作 $X \rightarrow Y$ 。
4. 非平凡函数依赖：对于关系模式的两个属性子集 X 和 Y ，如果 $X \rightarrow Y$ ，但 $Y \not\subseteq X$ ，则称 $X \rightarrow Y$ 为非平凡函数依赖；如果 $X \rightarrow Y$ ，但 $Y \subseteq X$ ，则称 $X \rightarrow Y$ 为平凡函数依赖。
5. 完全函数依赖：对于关系模式的两个属性子集 X 和 Y ，如果 $X \rightarrow Y$ ，并且对于 X 的任何一个真子集 X' ，都有 $X' \not\rightarrow Y$ ，则称 Y 对 X 完全函数依赖。
6. 范式：指符合某一种级别的关系模式的集合。在设计关系数据库时，根据满足依赖关系要求的不同定义为不同的范式。
7. 规范化：指将一个低一级范式的关系模式，通过模式分解转换为若干个高一级范式的关系模式的集合的过程。
8. 1NF：若关系模式的所有属性都是不可分的基本数据项，则该关系模式属于 1NF。
9. 2NF：1NF 关系模式如果同时满足每一个非主属性完全函数依赖于码，则该关系模式属

于 2NF。

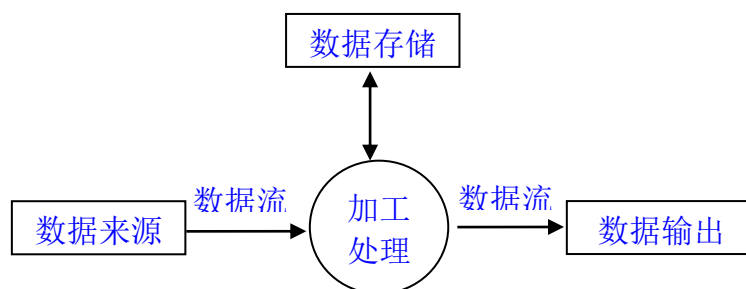
10. 3NF: 若关系模式的每一个非主属性既不部分依赖于码也不传递依赖于码, 则该关系模式属于 3NF。
11. BCNF: 若一个关系模式的每一个决定因素都包含码, 则该关系模式属于 BCNF。
12. 数据库设计: 是指对于一个给定的应用环境, 构造优化的数据库逻辑模式和物理结构, 并据此建立数据库及其应用系统, 使之能够有效地存储和管理数据, 满足各种用户的应用需求, 包括信息管理要求和数据操作要求。
13. 数据库设计的 6 个基本步骤: 需求分析, 概念结构设计, 逻辑结构设计, 物理结构设计, 数据库实施, 数据库运行和维护。
14. 概念结构设计: 指将需求分析得到的用户需求抽象为信息结构即概念模型的过程。也就是通过对用户需求进行综合、归纳与抽象, 形成一个独立于具体 DBMS 的概念模型。
15. 逻辑结构设计: 将概念结构模型 (基本 E-R 图) 转换为某个 DBMS 产品所支持的数据模型相符合的逻辑结构, 并对其进行优化。
16. 物理结构设计: 指为一个给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程。包括设计数据库的存储结构与存取方法。
17. 抽象: 指对实际的人、物、事和概念进行人为处理, 抽取所关心的共同特性, 忽略非本质的细节, 并把这些特性用各种概念精确地加以描述, 这些概念组成了某种模型。

二、填空题

1. 数据库设计必须遵循结构设计和行为设计相结合的原则。



2. 填空, 完成以下数据库各级模式关系示意图。
3. 用户对数据库的要求包括信息要求、处理要求、安全性要求以及完整性要求四个方面。
4. 填空, 完成以下 IPO 抽象图。



5. 数据字典主要包括数据项、数据结构、数据流、数据存储和处理过程五个部分。
6. 三种常用抽象方法是分类、聚集和概括。
7. 局部 E-R 图之间的冲突主要表现在属性冲突、命名冲突和结构冲突三个方面。
8. 数据库常用的存取方法包括索引方法、聚簇方法和 HASH 方法三种。
9. 确定数据存放位置和存储结构需要考虑的因素主要有：存取时间、存储空间利用率和维护代价等。

三、问答题

1. 已知关系模式 Student<U、F>, U = {学号, 所属系, 系主任, 课程号, 成绩}, 分析其属性间的函数依赖 F, 然后将其分解为更高级的范式以解决数据操作异常和冗余问题。
函数依赖: $F = \{ \text{学号} \rightarrow \text{所属系}, \text{所属系} \rightarrow \text{系主任}, (\text{学号}, \text{课程号}) \rightarrow \text{成绩} \}$
模式分解:
 - S (学号, 所属系, 学号 \rightarrow 所属系)
 - SC (学号, 课程号, 成绩, (学号, 课程号) \rightarrow 成绩)
 - DEPT (所属系, 系主任, 所属系 \rightarrow 系主任)
2. 简述数据库设计的内容和目标。
 - 数据库设计的内容就是针对一个给定的应用环境, 构造 (设计) 优化的数据库逻辑模式和物理结构, 并据此建立数据库及其应用系统, 使之能够有效地存储和管理数据, 满足各种用户的应用需求, 包括信息管理要求和数据操作要求。
 - 数据库设计的目标就是要为用户和各种应用系统提供一个信息基础设施和高效率的运行环境。
3. 简述数据库设计的基本过程。
 - 1) 需求分析阶段: 准确了解与分析用户需求 (包括数据与处理), 是最困难、最耗费时间的第一步。
 - 2) 概念结构设计阶段: 通过对用户需求进行综合、归纳与抽象, 形成一个独立于具体 DBMS 的概念模型。是整个数据库设计的关键。
 - 3) 逻辑结构设计阶段: 将概念结构模型转换为某个 DBMS 所支持的数据结构模型, 并对其进行优化。
 - 4) 数据库物理设计阶段: 为逻辑数据模型选取一个最适合应用环境的物理结构 (包括存储结构和存取方法)
 - 5) 数据库实施阶段: 运用 DBMS 提供的数据库语言 (如 SQL) 及宿主语言, 根据逻辑设计和物理设计的结果进行实施。
 - 6) 数据库运行和维护阶段: 试运行通过后的数据库应用系统即可投入正式运行, 并在运行过程中不断地对其进行评价、调整和优化。
4. 简述数据库需求分析的基本任务。
 - 详细调查现实世界要处理的对象 (组织、部门、企业等)
 - 充分了解原系统 (手工系统或计算机系统)
 - 明确用户的各种需求
 - 确定新系统的功能
 - 充分考虑今后可能的扩充和改变

5. 用户需求调查的具体步骤有哪些？
 - 调查组织机构情况
 - 调查各部门的业务活动情况
 - 在熟悉业务活动的基础上，协助用户明确对新系统的各种要求。
 - 确定新系统的边界
6. 数据项描述的内容有哪些？
 - 数据项描述={数据项名, 数据项含义说明, 别名, 数据类型, 长度, 取值范围, 取值含义, 与其他数据项的逻辑关系, 数据项之间的联系}
7. 数据流描述的内容有哪些？
 - 数据流描述={数据流名, 说明, 数据流来源, 数据流去向, 组成: {数据结构}, 平均流量, 高峰期流量}
8. 数据存储描述的内容有哪些？
 - 数据存储描述={数据存储名, 说明, 编号, 输入的数据流, 输出的数据流, 组成: {数据结构}, 数据量, 存取频度, 存取方式}
9. 简述建立索引的一般原则。
 - 如果一个(或一组)属性经常在查询条件中出现, 则考虑在这个(或这组)属性上建立索引(或组合索引);
 - 如果一个属性经常作为最大值和最小值等聚集函数的参数, 则考虑在这个属性建立索引;
 - 如果一个(或一组)属性经常在连接操作的连接条件中出现, 则考虑在这个(或这组)属性上建立索引;
 - 维护和查找索引需要开销, 频繁增删的关系不宜建立太多索引。
10. 数据库投入正式运行后为什么还需要维护? 维护工作由谁负责? 主要工作有哪些?
 - 数据库维护的原因主要有: ①应用环境在不断变化, ②物理存储会不断变化。
 - 数据库的日常维护工作主要由 DBA 负责, 主要工作有有:
 - 1) 数据库的转储和恢复
 - 2) 数据库的安全性、完整性控制
 - 3) 数据库性能的监督、分析和改进
 - 4) 数据库的重组织和重构造

第九章

一、名词解释

1. 代数优化: 指针对关系代数表达式的优化。即通过对关系代数表达式的等价变换(主要为改变查询语句中操作的次序和组合)来提高查询效率。
2. 物理优化: 指针对存取路径和底层操作算法的优化。即选择高效合理的操作算法或存取路径, 求得优化的查询计划以达到查询优化的目的。

二、填空题

1. 查询优化分为关系代数优化和物理优化两大类。
2. 表与表连接操作的实现方法主要有嵌套循环法、排序-合并法、索引连接法和哈希连接

法四种。

3. 查询执行的代价主要包括：I/O 代价(磁盘存取块数)、CPU 代价(处理机时间)以及内存代价等。

三、问答题

1. 简述 RDBMS 查询处理的基本步骤。
 - 查询分析：从查询语句中识别出语言符号(关键字、关系名、属性名等)，进行语法检查和语法分析，判断查询语句是否符合 SQL 语法规则。
 - 查询检查：对合法的查询语句进行语义检查，根据用户权限和完整性约束定义对安全性和完整性进行检查。
 - 查询优化：选择一个高效执行的查询处理策略，包括优化关系代数表达式和优化存取路径和底层操作算法。
 - 查询执行：生成优化的查询代码并执行。
2. 简述等值连接操作的各种实现方法。
 - 嵌套循环法：以 A 表的每一个元组作外层循环，检索 B 表中的每一个元组(内层循环)，检查这两个元组在连接属性(C)上是否相等，是则串接后作为结果输出，直到外层循环表中的所有元组处理完为止。
 - 排序-合并法：首先对连接的 A、B 两表按连接属性 C 排序。从 A 表的第一个元组开始，依次扫描 B 表中具有相同 C 值的元组，并把它们连接起来输出。当扫描到 C 值不同时中断扫描，返回 A 表扫描它的下一个元组，然后再返回 B 表从中断处继续扫描与该元组具有相同 C 值的元组，并把它们连接起来。重复上面的步骤直到 A 表扫描完毕业。
 - 索引连接法：首先必须在 B 表建立连接属性 C 的索引。然后从 A 表的第一个元组开始，由 C 值通过 B 表的索引查找相应的元组，并把这些元组和 A 表的这个元组连接起来输出。循环执行上面的步骤直到 A 表的所有元组处理完为止。
 - 哈希连接法：首先以连接属性 C 作为 hash 码，用同一个 hash 函数把 A 和 B 表的元组散列到同一个 hash 文件中。接着对包含较少元组的表(比如 B)进行一遍处理，把它的元组按 hash 函数分散到 hash 表的桶中。然后对另一个表(A)进行一遍处理，把 A 的元组散列到适当的 hash 桶中，把元组与桶中所有来自 B 并与之相匹配的元组连接起来。
3. 代数优化的典型启发式规则有哪些？
 - 选择运算应尽可能先做，这是优化策略中最重要、最基本的一条。
 - 投影运算和选择运算尽可能同时进行。如果可能，在扫描一个关系的同时完成所有的投影和选择运算以避免重复扫描关系。
 - 把投影同其前后的双目运算结合起来，也可以避免重复扫描关系。
 - 把某些选择同在它前面要执行的笛卡尔积结合起来成为一个连接运算。
 - 找出公共子表达式，先计算一次公共子表达式并把结果写入中间文件中，常常具有更好的效果。

第十章

一、名词解释

1. 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，要么全部做，是一个不可分割的工作单位，是恢复和并发控制的基本单位；通常以 **BEGIN TRANSACTION** 开始，以 **COMMIT** 或 **ROLLBACK** 结束。
2. 数据库恢复：指把数据库从错误状态恢复到某一已知的正确状态(亦称一致状态或完整状态)的过程。
3. 静态转储：指在系统中无事务运行时进行的转储操作。转储期间不允许对数据库的任何存取、修改活动，得到的一定是一个数据一致性的副本。
4. 动态转储：转储操作与用户事务并发进行，转储期间允许对数据库进行存取或修改。

二、填空题

1. 事务的基本特性包括 原子性、一致性、隔离性和持续性。
2. 常用的数据库恢复技术（建立冗余数据技术）主要有数据库转储和登记日志文件。

三、问答题

15. 什么是事务？简述事务的基本特性。

所谓事务是一用户定义的一个数据库操作序列，这些操作要么全做，要么全部做，是一个不可分割的工作单位，是恢复和并发控制的基本单位；通常以 **BEGIN TRANSACTION** 开始，以 **COMMIT** 或 **ROLLBACK** 结束。事务具有以下四个基本特性：

- 1) 原子性：事务是数据库的逻辑工作单位，一个事务的诸操作要么都做，要么都不做。
- 2) 一致性：指事务执行前后必须保持数据库的逻辑一致性。一致性和原子性是密切相关的。
- 3) 隔离性：指并发执行的各个事务之间不能互相干扰。
- 4) 持续性：又称为持久性或永久性，是指一个事务的操作提交后，其对数据库的改变是永久的，属于物理的而非逻辑的。

16. 何谓静态转储？何谓动态转储？它们各有什么优缺点？

- 静态转储指在系统中无事务运行时进行的转储操作。转储期间不允许对数据库的任何存取、修改活动，得到的一定是一个数据一致性的副本。
- 动态转储的转储操作与用户事务并发进行，转储期间允许对数据库进行存取或修改。
- 静态转储实现简单，但必须等待正运行的事务结束后才能进行，新的事务也必须等转储结束后方可开始，降低了数据库的可用性。
- 动态转储不用等待正在运行的用户事务结束即可开始，也不会影响新事务的运行，但不能确保副本中的数据一定正确有效，需要配合日志记录才能完成故障恢复。

17. 日志文件需要登记的内容包括哪些？

- 各个事物的开始（**BEGIN TRANSACTION**）标记；
- 各个事物的结束（**COMMIT** 或 **ROLLBACK**）标记；
- 各个事物的所有更新操作。

18. 日志文件的作用有哪些？

- 事物故障恢复和系统故障恢复必须用日志文件；
- 在动态转储方式中必须建立日志文件，后备副本和日志文件结合起来才能有效地恢

复数据库；

- 在静态转储方式中，也可以建立日志文件。

19. 简述事务故障的恢复步骤。

- 1) 反向扫描日志文件（即从最后向前扫描日志文件），查找该事务的更新操作。
- 2) 对该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) 继续反向扫描日志文件，查找该事务的其他更新操作，并做同样处理。
- 4) 如此处理下去，直至读到此事务的开始标记，事务故障恢复就完成了。

20. 简述系统故障的恢复步骤。

- 1) 正向扫描日志文件（即从头扫描日志文件），找出重做 (REDO) 队列和撤销 (Undo) 队列；
- 2) 对撤销 (Undo) 队列事务进行撤销 (UNDO) 处理：即反向扫描日志文件，对每个 UNDO 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：即正向扫描日志文件，对每个 REDO 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。

21. 检查点记录的内容包括哪些？

- 1) 建立检查点时刻所有正在执行的事物清单；
- 2) 这些事物最近一个日志记录的地址。重新开始文件用来记录各个检查点记录在日志文件中的地址。

22. 简述利用**检查点**的数据库恢复步骤。

- 1) 从重新开始文件中找到最后一个检查点记录在日志文件中的地址，由该地址在日志文件中找到最后一个检查点记录
- 2) 由该检查点记录得到检查点建立时刻所有正在执行的事务清单 ACTIVE-LIST，然后建立两个事务队列 (UNDO-LIST、REDO-LIST)，并把 ACTIVE-LIST 暂时放入 UNDO-LIST 队列，REDO 队列暂为空。
- 3) 从检查点开始正向扫描日志文件，直到日志文件结束。如有新开始的事务 T_i ，把 T_i 暂时放入 UNDO-LIST 队列，如有提交的事务 T_j ，把 T_j 从 UNDO-LIST 队列移到 REDO-LIST 队列，直到日志文件结束。
- 4) 对 UNDO-LIST 中的每个事务执行 UNDO 操作，对 REDO-LIST 中的每个事务执行 REDO 操作。