数据库原理

1.1 试述数据、数据管理、数据库管理系统、数据库的概念。

数据：描述事物的符号记录成为数据，如数值数据、文本数据和多媒体数据（如图形、图像、音频和视频）等。

数据管理：是对数据进行有效的分类、组织、编码、存储、检索、维护和应用，它是数据处理的中心问题。

数据库管理系统： 是由一个相互关联的数据的集合和一组用以访问、 管理和控制这些数据的程序组成。

数据库是长期储存在计算机内、有组织的、可共享的数据集合。

1.4 什么是数据独立性？数据独立性又分为哪两个层次？为什么需要数据独立性？

数据独立性是用来描述数据与应用程序之间的依赖程度， 包括数据的物理独立性和数据的逻辑独立性，依赖程度越低则独立性越高。

数据独立性又分为外模式 / 模式映像、模式 / 内模式映像两个层次。

数据的独立性把数据的定义从应用程序中分离出来，加上存取数据的方法又由数据库管理系统负责提供，从而大大简化了应用程序的编写，并减少了应用程序的维护代价。

1.6 什么是数据模型？数据模型的基本要素有哪些？为什么需要数据模型？

数据模型是一个描述数据语义、数据与数据之间联系（数据结构） 、数据操作，以及一致性（完整性）约束的概念工具的集合。

数据模型的基本要素： 1、数据结构； 2、数据操作； 3、数据的完整性约束条件。

由于计算机不可能直接处理现实世界中的具体事物，所以人们必须事先把具体事物转换成计算机能够处理的数据。也就是把现实世界中具体的人、物、活动、概念等用数据模型这个工具来进行抽象、表示和处理。

1.7 为什么数据模型要分为概念模型、逻辑模型和物理模型 3 类？试分别解释概念模型、逻辑模型和物理模型。

数据模型应满足 3 方面的要求：一是能比较真实地模拟现实世界；二是容易被人所理解；三是便于在计算机上实现。一种数据模型要很好地同时满足这 3 方面的要求是很困难的，因此数据库管理系统针对不同的使用对象和应用目的，分别采用概念模型、逻辑模型和物理模型。

概念模型：概念层次的数据模型称为概念数据模型，它按用户的观点或认识对现实世界的数据和信息进行建模，主要用于数据库设计。

逻辑模型：逻辑层是数据抽象的中间层，用于描述数据库数据的整体逻辑结构。

物理模型：物理层是数据抽象的最底层，用来描述数据的物理存储结构和存取方法。

1.9 关系模型的主要优点有哪些？

关系数据模型具有以下优点：

（1） 关系模型建立在严格的数学概念的基础之上，有关系代数作为语言模型，有关系数据理论作为理论基础。

（2） 关系模型的概念单一。无论实体还是实体之间的联系都是用关系来表示，对数据的操作结果还是关系。所以其数据结构简单、清晰，用户易懂易用。

（3） 关系模型的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作，提高了软件的开发和维护效率。

1.10 为什么数据库管理系统要对数据进行抽象？分为哪几级抽象？

一个商用的数据库管理系统必须支持高效的数据检索。这种高效性的需求促使设计者在数据库管理系统中使用复杂的数据结构来表示和存储数据。由于许多数据库管理系统的用户并未受过计算机专业训练，系统开发人员就通过多个层次上的抽象来实现对用户屏蔽复杂性，以简化用户与系统的交互。分为物理层

抽象、逻辑层抽象和视图层抽象。

1.11 试解释数据库的三级模式结构和两层映像。 为什么数据库管理系统要提供数据库的三级模式结构和两层映像？

数据库的三级模式是指数据库管理系统提供的外模式、模式和内模式 3 个不同抽象级别观察数据库中数据的角度。模式也成为逻辑模式，对应于逻辑层数据抽象，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。外模式也称为子模式或用户模式，对应于视图层数据抽象，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，是与某一具体应用有关的数据的逻辑表示。内模式也称存储模式，对应于物理层数据抽象，它是数据的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。

两层映像是指外模式 / 模式映像和模式 / 内模式映像。模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构。数据库中只有一个模式，也只有一个内模式，所以模式 / 内模式映像是唯一的，它定义了数据全局逻辑结构与存储结构之间的对应关系。

数据库的三级模式是对数据的 3 个级别的抽象， 它将数据的具体组织留给 DBMS 管理， 使用户能够逻辑地、抽象地看待和处理数据，而不必关心数据在计算机中的具体表示方式与存储方式。为了能够在系统内部实现这 3 个抽象层次的联系和转换， DBMS 在这三级模式之间提供了两层映像：外模式 / 模式映像、模式 / 内模式映像。正是这两层映像保证了数据库管理系统中的数据能够具有较高的逻辑独立性和物理独立性。

1.13 数据库管理系统的主要组成部分有哪些？主要功能有哪些？

数据库管理系统主要由数据库以及查询处理器、存储管理器和事物管理器等部分组成。

数据库管理系统的主要功能包括： （1）数据定义，提供了数据定义语言 DLL；（2）数据组织、存储和管理；（3）数据操纵，提供了数据操纵语言 DML；（4）数据库的事物管理和运行管理； （5）数据库的建立和维护等。

1.14 试述数据库系统的组成、 DBA 的主要职责。

数据库系统一般由数据库、数据库管理系统（及其开发工具） 、应用系统、数据库管理员和构成。

负责全面地管理和控制数据库系统。具体职责包括： （1）决定数据库中的信息内容和结构； （2）决定数据库的存储结构和存取策略；（3）定义数据的安全性要求和完整性约束条件； （4）监控数据库的使用和运行； （5）数据库的改进和重组重构。

2.1 简述如下概念，并说明它们之间的联系与区别。

（1）域，笛卡儿积，关系，元组，属性。

域：域是一组具有相同数据类型的值得集合。

笛卡儿积：给定一组域 D1, D2, …, Dn, 这些域中可以有相同的域。这组域的笛卡儿积为

$$D1*D2*\cdots*Dn=\{(d1,d2,\cdots,dn)|di \in Di,\ i=1,2,\cdots,n\}$$

其中，每个元素（ d1, d2…, dn）称为一个 n 元组（ n-tuple ）。元素中的每一个值 di 称为一个分量（component）。

关系：在域 D1, D2, …, Dn 上，笛卡儿积 D1*D2*…*Dn 的子集称为关系，表示为

$$R(D1, D2, \cdots, Dn)$$

元组：关系中的每个元素是关系中的元组。

属性：关系也是一个二维表，表的每行对应于一个元组，表的每列对应于一个域。由于域可以相同，为了加以区分，必须为每列起一个名字，称为属性。

（2）超码，候选码，主码，外码。

超码：对于关系 R的一个或多个属性的集合。如果属性集 A 可以唯一地标识关系 R 中的一个元组，则称属性集 A 为关系 R的一个超码。

候选码：若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码。

主码：若一个关系有多个候选码，则选定其中一个为主码。

外码：设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的码，如果 F 与基本关系 S 的主码 Ks 相对应，则称 F 是基本关系 R 的外码。

基本关系 R 称为参照关系，基本关系 S 称为被参照关系或目标关系。关系 R 和 S 可以是相同的关系。

（3）关系模式，关系，关系数据库。

关系模式：关系的描述称为关系模式，可以形式化地将其表示为 R（U,D,dom,F）其中， R 为关系名， U 为组成该关系的属性名集合， D 为属性组 U 中属性值所来自的域， 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。

关系：在域 D1, D2, …, Dn 上，笛卡儿积 D1*D2…*Dn 的子集称为关系，表示为

R （D1, D2, …, Dn）

关系式关系模式在某一时刻的状态或内容。 关系模式是静态的、 稳定的，而关系式动态的、 随实际而不断变化的，因为关系操作在不断地更新数据库中的数据。

关系数据库：关系数据库也有型和值之分。关系数据库的型也成为关系数据库模式，是对关系数据库的描述，它包括若干域的定义及在这些域上所定义的若干关系模式。关系数据库的值是这些关系模式在某一时刻所对应的关系的集合，通常称为关系数据库。

2.2 为什么需要空值 null?

对于一个关系而言，一个最基本的要求是它的每个属性的域必须是原子的。空值是所有可能的域的一个取值，表明值未知或不存在。

2.6 试述等值连接与自然连接的区别与联系。

（1）自然连接一定是等值连接，但等值连接不一定是自然连接。

（2）等值连接要求相等的分量，不一定是公共属性；而自然连接要求相等的分量必须是公共属性。 （3）等值连接不把重复的属性除去；而自然连接要把重复的属性除去。

2.8 对于图 2-8 所示的成绩管理数据库 ScoreDBoss 的模式导航图， 根据图 2-11 所示的实例数据， 试写出如下查询的关系代数表达式，并给出其查询结果。

（1）查找籍贯为“ 上海 ”的全体学生。

σ nation= ’上海’（ Student ）

（2）查找 1992 年元旦以后出生的全体男同学。

σ year(birthday)>=1992 Λ sex=’男’（ Student ）

（3）查找信息学院非汉族同学的学号、姓名、性别及民族。

∏ studentNo,studentName,sex,nation(σ institute= ’信息学院Λ nation!= ’汉’ (Student ∞Class))

（4）查找 08-09 学年第二学期（ 08092）开出的课程的编号、名称和学分。

∏ courseNo,courseName,creditHour(σ term= ’08092’(Score ∞Course))

（5）查找选修了“ 操作系统 ”的学生学号、成绩及姓名。

∏ studentNo,score,studentName(σ courseName=’操作系统’（ Student ∞Score∞Course）)

2.9 对于图 2-10 所示的学生选课数据库 SCDB 的模式导航图，试写出如下查询的关系代数表达式。

（1）查找 2008 级蒙古族学生信息，包括学号、姓名、性别和所属班级。

∏ studentNo,studentName,sex,className(σ nation= ’蒙古族’ Λ grade= ’2008’(Student ∞Class))

（2）查找“ C 语言程序设计 ”课程的课程号、上课时间以及上课地点

∏ courseNo,time,location(σ courseName=’C语言程序设计’ (Course ∞CourseClass ∞SC))

3.1 查询 1991 年出生的读者姓名、工作单位和身份证号。 （有问题）

SELECT readerName,workUnit,identitycard FROM Book WHERE

3.2 查询在信息管理学院工作的读者编号、姓名和性别。

SELECT readerNo,readerName,sex FROM Reader WHERE workUnit= ’信息管理学院’

3.3 查询图书名中含有“ 数据库 ”的图书的详细信息。

SELECT * FROM Book WHERE bookName LIKE ‘ %数据库 %’

3.4 查询吴文君老师编写的单价不低于 40 元的每种图书的图书编号、入库数量。

SELECT bookNo,shopNum FROM Book WHERE price>=40

3.5 查询在 2005-2008 年之间入库的图书编号、出版时间、入库时间和图书名称，并按入库时间排序输出。

SELECT bookNo,publishingDate,shopDate,bookName FROM Book

WHERE shopDate>=2005 AND shopDate<=2008

ORDER BY shopDate

3.6 查询借阅了 001~000029 图书编号的读者编号、图书编号、借书日期。

SELECT readerNo,bookNo,borrowDate FROM Borrow,Reader

WHERE Book.readerNo=Reader.readerNo AND bookNo BETWEEN 001 AND 000029

3.7 查询没有借阅图书编号以 001 开头的读者编号和姓名。

SELECT Book.readerNo,readerName FROM Borrow,Reader

WHERE bookNo NOT LIKE ‘001%’

3.8 查询读者马永强借阅的图书编号、图书名称、借书日期和应归还日期。

SELECT BookClass.bookNo,bookName,borrowDate,shouldDate FROM BookClass,Reader

WHERE Book.bookNo=Borrow.bookNo AND Reader.readerNo=Borrow.readerNo

AND readerName=’马永强’

3.26 创建一个视图，该视图为所借图书的总价在 150 元以上的读者编号、读者姓名和所借图书的总价。

```
CREATE VIEW TP
AS
  SELECT readerNo,readerName,sum(price) as tprice
    FROM Book,BookClass
    WHERE Reader.readerNo=Book.readerNo AND Book.bookNo=Borrow.bookNo
    AND  tprice>=150
```

3.27 创建一个视图，该视图为年龄在 25~35 之间的读者，属性列包括读者编号、读者姓名、年龄、工作单位、所借图书名称和借书日期。 （有问题）

```
CREATE VIEW AGE
AS
  SELECT
```

3.28 创建一个视图，该视图仅包含“清华大学出版社”在 2008-2009 年出版的“计算机类”的图书基本信息。

```
CREATE VIEW QHCB
AS
  SELECT *
  FROM BookClass,Book
  WHERE BookClass.classNo=Book.clasNo AND publishingName=  ’清华大学出版社’
  AND publishingDate BETWEEN 2008 AND 2009
```

3.29 对由题 3.28 所建立的视图进行插入、删除和更新操作。

插入： INSERT INTO VIEW_NAME VALUES(列值 1，列值 2，…，列值 n)

删除： DELETE FROM 视图名 WHERE 逻辑表达式

更新： UPDATE视图名

SET 列 1=列值 1

列 2 = 列值 2

……

WHERE 逻辑表达式

3.30 将入库数量最多的图书单价下调 5%

UPDATE TABLE Book

set price=price*(1-5%)

WHERE shopNum>ALL

4.3 假定一个销售公司的数据库包括一下信息。

（1）职工信息：职工号、姓名、电话、地址和所在部门；

（2）部门信息：部门号、部门所有职工、经理和销售的产品；

（3）产品信息：产品名、制造商、价格、型号及产品内部编号；

（4）制造商信息：制造商名称、地址、生产的产品号和价格。

试画出该公司的 E-R 图，并转化为关系模式。

【例 5.8 】r(R) 和 F 定义同例 5.7 ，判断 AG是否为 r(R) 的候选码。

例 5.7 已计算出（ AG+）=ABCGHI 则还要进一步分别计算 A+和 G+。经计算得， A+=ABCH G+=G 它们都不包含 R 的所有属性，因此 AG为 r(R) 候选码。

【例 5.13 】r(R)=r(A,B,C) ， F={A B,B C}。r(R) 的候选码为 A， r(R) ?BCNF 因此函数依赖 B C 中的决定属性 B 不是超码。

【例 5.14 】r(R)=r(A,B,C) ， F={AB C,C A}。r(R) 的候选码为 AB或 BC， r(R) ∈BCNF 因为两个函数依赖中的决定属性 AB或是 BC都是 r(R) 的候选码。

【例 5.16 】r(R)=r(A,B,C) ， F={A B, B C}。r(R) 的候选码为 A， r(R) ?3NF且 r(R) ?BCNF

【例 5.17 】r(R)=r(A,B,C) ， F={AB C,C A}。r(R) 的候选码为 AB或 BC,r(R) ∈3NF但 r(R) ?BCNF

【例 5.18 】r(R)=r(A,B,C) ， F={AB C,BC A}。r(R) 的候选码为 AB或 BC,r（R) ∈3NF且 r(R) ∈BCNF

8.2 查询代价如何度量？为什么？

查询处理的代价可以通过该查询对各种资源的使用情况进行度量，主要包括磁盘存取时间和执行一个查询所用 CPU时间以及在并行 / 分布式数据库系统中的通信开销等。由于磁盘存取比内存操作速度慢且大型数据库的数据量打，因此通常忽略 CPU时间，而仅仅用磁盘存取代价来度量查询执行计划的代价。对于磁盘存取代价，可以通过传输磁盘块数以及搜索磁盘次数来度量。例如一个传输 b 块并作 S 次磁盘搜索的操作耗时 b*tT+S*tS ms ，其中 tT 表示传输一块数据的平均耗时， ts 表示搜索一次磁盘的平均定位时间（包括搜索时间加旋转时间） 。

8.8 为什么需要查询优化？什么是查询执行计划？查询优化器的输入和输出分别是什么？

处理一个给定的查询，尤其是复杂的查询，通常会有许多种策略。查询优化就是从这许多策略中找出最有效的查询执行计划的处理过程。不期望用户能够写出一个能高效处理的查询，而是期望 RDBMS能够构造并旋转出一个具有最小查询执行代价的查询执行计划。

查询执行计划是指用于执行一个查询的原语操作序列。

查询优化器的输入和输出分别是……

9.1 列级约束和元组约束的区别在哪里？

如果定义列级的同时定义约束条件，则为列级约束；如果单独定义约束条件，则为元组级的约束。

9.2 由用户定义约束名称有什么好处？

用户命名有两点好处：一是便于理解约束的含义；二是修改约束方便，不必查询数据字典。

9.4 阐述数据库管理系统如何实现完整性约束。

为了实现完整性约束，数据库管理系统必须提供：

（1）定义完整性约束条件的机制。

（2）提供完整性检查方法。

（3）违约处理。若发现用户操作违背了完整性约束条件，应采取一定的措施，如拒绝操作等。

9.5 如果一张表有多种完整性约束，请分析系统按什么顺序来检查这些约束，当其中某个约束违反时，系统如何处理？（前半部分解答有问题）

当插入或对主码列进行更新操作时，关系数据库管理系统按照实体完整性规则自动进行检查。关于参照完整性，对参照表和被参照表进行修改操作有可能会破坏参照完整性，系统首先会检查是否违反了参照完整性，如果违反了，则进行违约处理。

对于违反实体完整性和用户定义的完整性的操作，一般都采用拒绝执行的方式进行处理。而对于违反参照完整性的操作，并非都是简单地拒绝执行，有时要根据应用语义执行一些附加的操作，以保证数据库的正确性。

10.1 什么是事务的 ACID特性？ DBMS分别是如何保证这些特征的？

事务 ACID特性是指：（1）原子性，即事务的所有操作要么全部都被执行，要么都不被执行。（2）一致性，即一个单独执行的事务应保证其执行结果的一致性，即总是将数据库从一个一致性状态转化到另一个一致性状态。（3）隔离性，即当多个事务并发执行时，一个事务的执行不能影响另一个事务，即并发执行的各个事务不能相互干扰。（4）持久性，即一个事务成功提交后，它对数据库的改变必须是永久的，即使随后系统出现故障也不会收到影响。

原子性也称为故障原子性或可靠性，它是由 DBMS通过撤销未完成事务对数据库的影响来实现的。一致性是指单个事务的一致性， 也称为并发原子性或正确性，它是由编写该事务代码的应用程序员负责，但有时也可利用 DBMS提供的数据库完整性约束（如触发器）的自动检查功能来保证。隔离性也称为执行原子性或可串行化，可以看做是多个事务并发执行时的一致性或正确性要求，其正确性由 DBMS的并发控制模块保证。 而持久性则是利用已记录在稳定存储介质中（如磁盘）的恢复信息（如日志、备份等）来实现丢失数据（如因中断而丢失的存放在主存中但未保存到磁盘数据库中去得数据等）的恢复。原子性和持久性是由 DBMS的恢复管理模块保证。

10.2 数据库为什么需要并发控制？

数据库是共享资源，通常有许多事务同时运行。

当多个事务并发存取数据库中的数据时，会产生同时读取和 / 或修改同一数据的情况。若对并发操作不加以控制，可能会导致存取和存储不正确的数据，破坏数据库的一致性。所以，数据库管理系统必须提供并发控制机制。