

# 数据库系统复习

Author: 冰镇南瓜汁

软院人论坛

<http://www.ssduter.com/>

# 1 数据库系统基础概念

## 1.1 关系型数据库系统的基本概念及构成

数据：描述现实世界事物的符号记录。

数据库（Database）：长期存储在计算机内，有组织的，可共享的大量数据的集合。

数据库管理系统（DBMS）：用于科学地组织和存储数据，高效地获取和维护数据的系统软件。

数据库系统（DBS）：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（及其开发工具）、应用系统、数据库管理员和用户构成。

数据库系统的四个特点：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。

## 1.2 数据模型相关概念

模型：

数据模型：

数据模型由数据结构、数据操作和数据的约束条件 3 个要素组成。

数据模型的三种要求：真实模拟；容易理解；便于在计算机上实现。

数据模型的分类：概念模型、逻辑模型及物理模型。

逻辑模型的种类：层次模型，网状模型，关系模型，面向对象模型和对象关系模型。

两步抽象过程：现实世界→概念模型→DBMS 支持的数据模型。

数据结构是数据模型什么特性的描述？

数据操作是数据模型什么特性的描述？

### 1.2.1 概念模型中的基本概念：

实体（Entity）：客观存在并可相互区别的事物称为实体。

属性（Attribute）：实体所具有的某一特性称为属性。

码（Key）：唯一标识实体的属性集称为码。

域（Domain）：属性的取值范围称为该属性的域。

联系（Relationship）：现实世界中事物内部以及事物之间的联系在信息世界中反映为实体内部的联系和实体之间的联系。

### 1.2.2 关系模型中的基本概念

关系（Relationship）：关系是笛卡尔乘积的有限子集。关系中的每行叫做一个元组，每列叫做一个属性。

候选码（Candidate key）：若关系中的某一（最小）属性组的值能唯一地标识一个元组，则该（最小）属性组称为候选码。

主码（Primary key）：若关系中有多个候选码，则选定其中的一个为主码。

主属性（Prime attribute）：候选码的诸属性称为主属性。

非主属性 (Nonprimary attribute)：不被任何候选码包含的属性。

全码 (All-key)：关系模式的所有属性都是这个关系模式的候选码。

外码 (Foreign key)：关系模式  $R$  中的属性或属性组  $X$  不是  $R$  的码，但  $X$  是另一个关系模式的码，则称  $X$  是  $R$  的外码。

关系模式：对关系的描述称为关系模式。可以形式化的表示为： $R(U, D, dom, F)$ ， $R$ ：关系名； $U$ ：组成关系的属性名集合； $D$ ：属性来自的域； $dom$ ：属性到域的映射； $F$ ：属性间数据的依赖关系。

### 1.3 数据库系统结构

数据库系统结构：三层模式和两层映像。

三层模式：内模式、模式和外模式。

两层映像：外模式 / 模式映像，模式 / 内模式映像。

内模式：也称存储模式，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。一个数据库只有一个内模式。

模式：也称逻辑模式或概念模式，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，综合了所有用户的需求，一个数据库只有一个模式。

外模式：也称子模式或用户模式，是数据库用户（包括应用程序员和最终用户）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。一个数据库可以有多个外模式，同一外模式可以为用户的多个应用系统使用，但一个应用只能使用一个外模式。

外模式 / 模式映像：定义外模式与模式之间的对应关系，每一个外模式都对应一个外模式 / 模式映像，保证数据与程序的逻辑独立性。

模式 / 内模式映像：定义了数据全局逻辑结构与存储结构之间的对应关系。数据库中模式 / 内模式映象是唯一的，保证了数据与程序的物理独立性。

### 1.4 关系完整性

关系模型定义了 3 类完整性约束：实体完整性、参照完整性和用户定义的完整性。

实体完整性（约束主属性取值）：主属性不能取空值。

参照完整性（约束外码取值）：基本关系  $R$  的外码  $T$  与基本关系  $S$  的主码  $L$  对应，则  $T$  要么取空值，要么取  $L$  的某个值。

用户定义的完整性：根据应用的具体情况，往往需要一些特殊的约束条件。

### 1.5 数据依赖

函数依赖： $R$  中任意两个元组  $t, s$ ，若  $t[X]=s[X]$ ，则  $t[Y]=s[Y]$ ，则称  $X \rightarrow Y$ ，称作  $Y$  函数依赖于  $X$ 。

在  $R(U)$  中，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  $X$  完全依赖。

若  $X \rightarrow Y$  但  $Y$  不完全依赖于  $X$ ，则称  $Y$  对  $X$  部分函数依赖。

在关系模式  $R(U)$  中，如果  $X \rightarrow Y$ ， $Y \rightarrow Z$ ，且  $Y \subsetneq X$ ， $Y \not\rightarrow X$ ，则称  $Z$  传递函数依赖于  $X$ 。

## 1.6 范式

1NF: 关系模式的每一个属性都是不可再分的数据项。

2NF: 满足 1NF, 且每一个非主属性完全函数依赖于主码。

3NF: 满足 2NF, 且每一个非主属性不传递函数依赖于主码。

BCNF: 满足 3NF, 且每一个主属性不部分函数或传递函数依赖于主码。

BCNF 的特点: 所有非主属性都完全函数依赖于每个候选码, 所有主属性都完全函数依赖于每个不包含它的候选码, 没有任何属性完全函数依赖于非码的任何一组属性。

## 1.7 视图与索引

视图是从一个或几个基本表(或视图)导出的虚表。只存放视图的定义, 不会出现数据冗余。

视图的作用: 视图是关系数据库的外模式, 可以按多种角度组织数据, 对数据有一定的安全保护作用。

索引有两种类型: 唯一值索引和聚簇索引。唯一值索引指每一个索引值只对应唯一的数据记录。聚簇索引指具有相同索引项的记录存放在连续的物理块上。

索引的作用: 加快查询速度。

## 1.8 事务处理与并发控制

事务(Transaction)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。事务是恢复和并发控制的基本单位

**事务的 ACID 特性:** 原子性 (Atomicity)、一致性 (Consistency)、隔离性 (Isolation)、持续性 (Durability)。

原子性: 事务是数据库的逻辑工作单位, 事务中包括的诸操作要么都做, 要么都不做。

一致性: 事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。

隔离性: 事务在执行时互不干扰。

持续性: 一个事务一旦成功提交, 它对数据库中数据的改变就是永久有效的。

数据库故障类型: **事务故障、系统故障、介质故障**

**事务故障和系统故障的回复由 DBMS 自动完成, 介质故障需要 DBA 干预进行恢复。**

三种数据不一致问题: **丢失修改、读脏数据、不可重复读。**

并发控制的主要技术是封锁机制。

封锁的类型: 排他锁 (X 锁, 写锁), 共享锁 (S, 读锁)。

若事务 T 对对象 A 加写锁, 只允许 T 读取和修改 A, 其他事务都不能对 A 加任何类型的锁。

若事务 T 对对象 A 加读锁，则事务 T 可以读取 A，不能修改 A，其他事务只能对 A 加读锁，不能加写锁。

1 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。只能防止丢失修改。

2 级封锁协议：1 级封锁协议 + 事务 T 在读取数据 R 前必须先加 S 锁，读完后即可释放 S 锁。可以防止丢失修改和读脏数据。

3 级封锁协议：1 级封锁协议 + 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。可以防止丢失修改、读脏数据、不可重复读。

死锁预防：一次封锁法、顺序封锁法。

死锁诊断：超时法、事务等待图法。

死锁解除：选择一个处理死锁代价最小的事务，将其撤消，释放此事务持有的所有的锁，使其它事务能继续运行下去。

可串行性是并行事务正确性的唯一准则。

两段锁协议：在对任何数据进行读、写操作之前，事务首先要获得对该数据的封锁；在释放一个封锁之后，事务不再获得任何其他封锁。

对比：（1）三级封锁协议、一次封锁法、两段锁协议的目的不同；（2）一次封锁法遵守两段锁协议，但两段锁协议不一定满足一次封锁法，有可能死锁；（3）满足三级封锁协议一定满足两段锁协议，反之不一定；（4）满足三级封锁协议不一定满足一次封锁法，反之也不一定。

多粒度封锁：显式封锁、隐式封锁。

意向锁：意向共享锁（IS 锁）、意向排他锁（IX 锁）、共享意向排他锁（SIX 锁=S+IX）。

## 2 关系数据理论及应用

### 2.1 应用关系数据理论进行分析

要求掌握函数依赖并根据问题描述写出相应的函数依赖，能根据范式定义判断关系模式的范式级别或进行关系模式的规范化处理。

关系变量  $R$  无损分解至一系列属于某个范式  $XNF$  的投影集  $D$  的算法

1. 初始化  $D$  为只包含  $R$
2. 对  $D$  中的每个非某  $XNF$  关系变量  $T$ ，执行步骤 3 和 4
3. 设  $X \rightarrow Y$  为  $T$  的不满足某  $XNF$  要求的函数依赖。
4. 用  $T$  的两个投影（在  $X$  和  $Y$  上的投影及在除  $Y$  中属性之外的所有属性上的投影）代替  $D$  中的  $T$

例：现有如下关系模式：

(公司编号,分店编号,法人代表,联系电话,分店店长名)

存在以下事实：

不同公司的分店可以具有相同的分店编号

某个公司的某个分店只能有一个唯一的分店店长。

某个公司只能有一个法人代表，但一个法人代表可以代表多个公司。

法人代表只使用一个公开的联系电话。

- 1.请写出此关系模式  $R$  中存在的数据依赖，并确定此关系模式的码。
- 2.判断关系模式  $R$  是否属于第 2 范式，请写明判定依据。
- 3.请将该关系模式分解到满足 BCNF。

### 2.2 关系代数

掌握关系代数的运算方法，要求能够根据查询要求写出使用关系代数表达的查询表达式或运算结果。

专门的关系运算：笛卡尔积、选择、投影、连接、除。

选择：取特定条件的元组（行）。

投影：取指定的属性（列）。

连接：从两个关系的笛卡尔积中选取属性间满足一定条件的元组。

除：一般用于计算满足包含某种条件下的所有元组值的某些属性。

等值连接与自然连接的区别和联系：

连接运算符是“=”的连接运算称为等值连接。它是从关系 R 与 S 的笛卡尔积中选取 A, B 属性值相等的那些元组。自然连接是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并在结果中把重复的属性列去掉。

例题：

设有一个 SPJ 数据库，包括 S、P、J、SPJ 4 个关系模式：

S( SNO, SNAME, STATUS, CITY );

P( PNO, PNAME, COLOR, WEIGHT );

J( JNO, JNAME, CITY );

SPJ( SNO, PNO, JNO, QTY );

供应商表 S 由供应商代码、供应商姓名、供应商状态、所在城市组成；

零件表 P 由零件代码、零件名、颜色、重量组成；

工程项目表 J 由工程项目代码、工程项目名、所在城市组成；

供应情况表 SPJ 由供应商代码、零件代码、工程项目代码、供应数量组成；

**1) 求供应工程 J1 零件的供应商号码 SNO：**

$\pi_{Sno}(\sigma_{Sno='J1'}(SPJ))$

**2) 求供应工程 J1 零件 P1 的供应商号码 SNO：**

$\pi_{Sno}(\sigma_{Sno='J1' \wedge Pno='P1'}(SPJ))$

**3) 求供应工程 J1 零件为红色的供应商号码 SNO：**

$\pi_{Sno}(\sigma_{Pno='P1'}(\sigma_{COLOR='红'}(P) \bowtie SPJ))$

**4) 求没有使用天津供应商生产的红色零件的工程号 JNO：**

$\pi_{Jno}(SPJ) - \pi_{JNO}(\sigma_{city='天津' \wedge Color='红'}(S \bowtie SPJ \bowtie P))$

**5) 求至少用了供应商 S1 所供应的全部零件的工程号 JNO：**

$\pi_{Jno, Pno}(SPJ) \div \pi_{Pno}(\sigma_{Sno='S1'}(SPJ))$

### 3 关系模型数据操作

#### 3.1 DDL 语句

掌握对于基本数据对象（基本表、视图、索引）使用的 DDL 语句，包括 CREATE, DROP, ALTER。

创建表：

```
CREATE TABLE
Course(
    Cno CHAR(9) PRIMARY KEY,
    Cname CHAR(20) NOT NULL UNIQUE,
    Cpno CHAR(2),
    FOREIGN KEY (Cpno) REFERENCES Course(Cno)
);
```

修改表：

```
ALTER TABLE Student
    ADD Scome DATE;
-- ALTER COLUMN Sage SMALLINT;
-- DROP unique(sname);
```

删除表：

```
DROP TABLE Student;
```

创建视图：

```
CREATE VIEW IS_Student
AS
SELECT Sno, Sname, Sage FROM Student
WHERE Sdept= 'IS' WITH CHECK OPTION;
```

删除视图：

```
DROP VIEW IS_S1;
```

创建索引

```
CREATE UNIQUE INDEX SCno ON SC(Sno ASC, Cno DESC);
```

删除索引

```
DROP INDEX Stusname;
```

#### 3.2 DML 语句

掌握 INSERT, UPDATE, DELETE 语句的基本应用。

数据插入：

```
INSERT INTO SC(Sno, Cno)
VALUES ('95020', '1');
```

数据修改：

```
UPDATE Student
```



```
SET Sage= Sage+1
WHERE Sdept='IS';
```

数据删除:

```
DELETE FROM Student WHERE Sno='95019';
```

在这些操作中可以使用子查询，另外这些操作可以对视图进行。例如:

```
INSERT INTO Deptage(Sdept, Average)
SELECT Sdept, AVG(Sage)
FROM Student
GROUP BY Sdept;
```

### 3.3 查询语句

综合掌握和灵活运用所有查询语句。

单表查询基本框架:

```
SELECT [DISTINCT] 目标列表表达式, ...
FROM 表名或视图名
[WHERE 条件表达式]
[GROUP BY 列名, ... [HAVING 条件表达式]]
[ORDER BY 列名[ASC|DESC], ...];
```

确定范围: BETWEEN ... AND ...

确定集合: IN

字符匹配: LIKE

空值: IS NULL

聚集函数: COUNT(\*), COUNT(列名), SUM(列名), AVG(列名), MAX(列名), MIN(列名)

聚集函数中可以使用 DISTINCT 短语进行限定。

GROUP BY 子句与聚集函数配合使用时, SELECT 子句只能出现分组属性和聚集函数。

聚集函数可以出现在 SELECT 子句和 HAVING 子句中, 不能出现在 WHERE 子句中。

多表查询: 常用自然连接、非等值连接和自身连接, 其中自身连接需要给表起别名。

子查询: 使用 IN 或比较运算符或 EXISTS, 子查询中不能使用 ORDER BY。

集合查询: UNION, INTERSECT, EXCEPT。

具体查询练习见附录 A。

## 4 数据库设计应用

掌握 E-R 图的绘制方法，能根据实际问题描述设计正确的概念模型，并用 E-R 图进行表示；掌握 E-R 图转换为逻辑模型的方法，能根据实际情况进行正确的转换；能对所生成的逻辑模型进行分析，能判断所属的范式，并将其规范化到某一级别的范式（最高 BCNF）。

主要注意概念模型向关系模型的转换：

（1）一个实体转换为一个关系模式。关系的属性为实体的属性，关系的码为实体的码。

（2）一个  $m:n$  联系转换为一个关系模式。关系的属性为与该联系相连的各实体的码以及联系本身的属性，关系的码为各实体码的组合。

（3）一个  $1:n$  联系可以转换为一个独立的关系模式，也可以与  $n$  端对应的关系模式合并。如果转换为一个独立的关系模式，关系的属性是与该联系相连的各实体的码以及联系本身的属性，关系的码是  $n$  端实体的码。

如果与  $n$  端对应的关系模式合并，合并后关系的属性是在  $n$  端关系中加入  $1$  端关系的码和联系本身的属性，合并后关系的码不变。可以减少系统中的关系个数，一般情况下更倾向于采用这种方法。

（4）一个  $1:1$  联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。

如果转换为一个独立的关系模式，关系的属性为与该联系相连的各实体的码以及联系本身的属性，每个实体的码均是该关系的候选码。

如果与某一端对应的关系模式合并，合并后关系的属性为加入对应关系的码和联系本身的属性，合并后关系的码不变。

（5）三个或三个以上实体间的一个多元联系转换为一个关系模式。关系的属性为与该多元联系相连的各实体的码以及联系本身的属性，关系的码为各实体码的组合。

（6）同一实体集的实体间的联系，即自联系，也可按上述  $1:1$ 、 $1:n$  和  $m:n$  三种情况分别处理。

（7）具有相同码的关系模式可合并。

## 附录 A SQL 查询练习

现在有一教学管理系统，具体的关系模式如下：

Student (no, name, sex, birthday, class)

Teacher (no, name, sex, birthday, prof, depart)

Course (cno, cname, tno)

Score (no, cno, degree)

**DML 练习 1.** (略) 2 (略)

### 单表查询

3. 以 class 降序输出 student 的所有记录 (student 表全部属性)

```
select * from student order by class desc;
```

4. 列出教师所在的单位 depart (不重复)。

```
select distinct depart from teacher;
```

5. 列出 student 表中所有记录的 name、sex 和 class 列

```
select name,sex,class from student;
```

6. 输出 student 中不姓王的同学的姓名。

```
select name from student where name not like '王%';
```

7. 输出成绩为 85 或 86 或 88 或在 60-80 之间的记录 (no,cno,degree)

```
select * from score where degree in (85,86,88) or degree between 60 and 80;
```

8. 输出班级为或性别为‘女’的同学 (student 表全部属性)

```
select * from student where sex='女' or class='95001';
```

9. 以 cno 升序、degree 降序输出 score 的所有记录。(score 表全部属性)

```
select * from score order by cno,degree desc;
```

10. 输出男生人数及这些男生分布在多少个班级中

```
select count(*), count(distinct class) from student where sex='男';
```

11. 列出存在有 85 分以上成绩的课程编号。

```
select distinct cno from score where degree>85;
```

12. 输出 95001 班级的学生人数

```
select count(*) from student where class='95001';
```

13. 输出‘3-105’号课程的平均分

```
select avg(degree) from score where cno='3-105';
```

14. 输出 student 中最大和最小的 birthday 日期值

```
select max(birthday),min(birthday) from student;
```

15. 显示 95001 和 95004 班全体学生的全部个人信息 (不包括选课)。

```
select * from student where class in ('95001','95004');
```

### 聚合查询

16. 输出至少有个同学选修的并以 3 开头的课程的课程号, 课程平均分, 课程最高分, 课程最低分。

```
select cno,avg(degree), max(degree),min(degree) from score where cno like '3%' group by cno having count(*)>=5;
```

17. 输出所选修课程中最低分大于 70 分且最高分小于 90 分的学生学号及学生姓名。

```
select student.no,student.name from score,student where student.no=score.no group by student.no,student.name having max(degree)<90 and min(degree)>70;
```

18. 显示所教课程选修人数多于 5 人的教师姓名

```
select teacher.name from course,teacher where teacher.no=course.tno and cno in (select cno from score group by cno having count(*)>5);
```

19. 输出'95001'班级所选课程的课程号和平均分

```
select cno,avg(degree) from student,score where student.no=score.no and class='95001' group by cno;
```

20. 输出至少有两男同学的班级编号。

```
select class from student where sex='男' group by class having count(*)>=2;
```

**多表查询**

21. 列出与 108 号同学同年出生的所有学生的学号、姓名和生日

```
select s1.no,s1.name,s1.birthday from student s1,student s2 where year(s1.birthday)=year(s2.birthday) and s2.no=108;
```

22. 列出存在有 85 分以上成绩的课程名称

```
select distinct cname from score,course where score.cno=course.cno and degree>85;
```

23. 列出“计算机系”教师所教课程的成绩表(课程编号, 课程名, 学生名, 成绩)。

```
select course.cno,cname,student.name,degree from student,course,score,teacher where student.no=score.no and course.cno=score.cno and course.tno=teacher.no and depart='计算机系';
```

24. 列出所有可能的“计算机系”与“电子工程系”不同职称的教师配对信息, 要求输出每个老师的姓名和职称

```
select distinct t1.name,t1.prof ,t2.name,t2.prof from teacher t1,teacher t2 where t1.prof!=t2.prof and t1.depart != t2.depart and t1.depart in ('计算机系','电子工程系') and t2.depart in ('计算机系','电子工程系');
```

25. 列出所有处于不同班级中, 但具有相同生日的学生, 要求输出每个学生的学号和姓名。

```
select stu1.no,stu1.name,stu2.no,stu2.name from student stu1,student stu2 where stu1.class<>stu2.class and datediff(day,stu1.birthday,stu2.birthday)=0;
```

26. 显示‘张三’教师任课的学生姓名, 课程名, 成绩

```
select student.name,cname,degree from student,score,course,teacher where student.no=score.no and score.cno=course.cno and teacher.no=course.tno and teacher.name='张三';
```

27. 列出所讲课已被选修的教师的姓名和系别

```
select distinct teacher.name,depart from teacher,course,score where
```

teacher.no=course.tno and score.cno=course.cno;

28. 输出所有学生的 name、no 和 degree。(degree 为空的不输出和为空的输出两种情况)。

```
(1) select name,student.no,degree from student,score where
student.no=score.no;
(2) select name,student.no,degree from student left join score on
student.no=score.no order by student.no;
```

29. 列出所有任课教师的 name 和 depart。(从课程选修和任课两个角度考虑)

--从课程选修角度

```
select distinct teacher.name,depart from teacher,course,score
where teacher.no=course.tno and score.cno=course.cno;
```

--从任课角度

```
select distinct teacher.name,depart from teacher,course
where teacher.no=course.tno;
```

30. 输出男教师所上课程名称。

```
select cname from course,teacher
where course.tno=teacher.no
and teacher.sex='男';
```

31. 出与“李军”同性别的所有同学的名字。

```
select name from student
where sex=(select sex from student where name='李军');
```

或

```
select s1.name from student s1,student s2
where s1.sex=s2.sex and s2.name='李军';
```

32. 输出选修“数据结构”课程的男同学的成绩。

```
select degree from student,score,course
where student.no=score.no and course.cno=score.cno
and course.cname='数据结构' and sex='男';
```

33. 列出选修编号为‘3-105’课程并且该门课程成绩比课程‘3-111’的最高分要高的 cno,no 和

```
select cno,no,degree from score
where cno='3-105' and degree>(select max(degree)
from score where cno='3-111');
```

子查询

34. 输出 score 中成绩最高的学号和课程号

```
select no,cno from score where degree=(select max(degree) from score);
```

35. 输出选修 3-105 课程，其成绩高于 109 号同学在此课程所得成绩的所有同学的学号，姓名

```
select student.no, name from student,score,course where student.no=score.no and
score.cno=course.cno and course.cno='3-105' and degree>(select degree from
score where cno='3-105' and no=109);
```

或

```
select no, name from student where exists(select * from score score1 where
score1.no=student.no and cno='3-105' and exists( select * from score score2 where
score1.degree>score2.degree and score2.cno='3-105' and score2.no=109));
```

36. 列出成绩比该课程平均成绩低的同学的学号，成绩和该门课的平均成绩

```
select no,degree,avg_degree from score s1,(select cno,avg(degree) avg_degree
from score group by cno) avg_score where s1.cno=avg_score.cno and
degree<avg_degree;
```

37. 列出没有实际授课的教师的姓名和系别

```
select teacher.name,depart from teacher where no not in (select tno from
course,score where course.cno=score.cno);
```

38. 列出选修了编号为‘3-105’课程且其成绩高于‘4-109’课程最高成绩的同学的课程编号，学号和成绩

```
select cno,no,degree from score where cno='3-105' and degree> (select max(degree)
from score where cno='4-109');
```

39. \*\*列出符合下述条件的所有可能的同学配对(sno1,sname1,sno2,sname2,difference)。其中要求学号为 sno1 的 sname1 同学的所学课程的平均分大于学号为 sno2 的 sname2 同学的所学课程平均分,两个同学的课程平均分的差值 difference 为(sno1 同学平均分-sno2 同学平均分)

```
select stu1.no,stu1.name,stu2.no,stu2.name,stu1.avgdegree-stu2.avgdegree
difference from (select student.no, student.name,avgdegree from student,
(select no, avg(degree) avgdegree from score group by no) tempstu1 where
student.no=tempstu1.no) stu1, (select student.no, student.name ,avgdegree from
student, (select no, avg(degree) avgdegree from score group by no) tempstu2 where
student.no=tempstu2.no) stu2 where stu1.avgdegree>stu2.avgdegree
```

## 附录 B 上古时期《数据库原理》期末试题

## 大 连 理 工 大 学

课程名称：数据库原理

试卷：A 考试形式：闭卷

授课院（系）：软件学院

考试日期：上古时期 试卷 共\_\_页

答案写在答题纸上。

|    | 一  | 二  | 三  | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 总 |
|----|----|----|----|---|---|---|---|---|---|---|---|
| 标准 | 15 | 20 | 20 | 5 |   |   |   |   |   |   |   |
| 得分 |    |    |    |   |   |   |   |   |   |   |   |

## 一、概念与简答题（共 15 分 每小题 3 分）

1. 简述数据库系统三级模式及二级映射的对应关系
2. 阐明连接操作的重要作用及自然连接与等值连接的区别
3. 阐述关系模型的三个完整性约束
4. 对比分析部分函数依赖、完全函数依赖和传递函数依赖的异同
5. 数据库的故障类型有那几种？在何种情况下不需要实施数据库恢复？

## 二、程序计算题（共计 20 分）

1. 现有关系数据库如下：（总计 8 分）

系别(系别编号，系名称，系主任姓名)

学生(学号，姓名，性别，系编号，班级，年龄)

课程(课程号，名称，学分)

选修(学号，课程号，分数)

奖学金（奖学金编号，奖学金名称，提供单位，奖学金金额）

获奖（学号，奖学金编号，获奖年度）

其中：

学生关系中专业属性使用文字方式记录学生所属专业，

奖学金关系中获奖年度使用整型数值类型存储时间信息

用 SQL 表达式实现：（每题 2 分）

- 1) 显示“0610”班的学生人数
- 2) 查询得过奖学金、同时至少有一门课程成绩在 95 分以上的学生信息，包括学号、姓名和系别名称；
- 3) 显示所有课程中的最高分的学生学号、姓名和课程号、课程名
- 4) 显示选修“数据库原理”课程的成绩高于“06072”号同学成绩的所有同学的记录

2. 基于数据库中的学生表、成绩表、任课表：

学生（学号，姓名，性别，出生日期，系名）

成绩（学号，课程名，成绩）

任课（课程名，任课教师名）

用 SQL 语言创建一个含有学号、姓名、性别、系名、课程名、成绩、任课教师名的学生成绩视图。（2 分）

3. 设有关系 R 和 S 如下，写出  $R \div S$  计算结果。（2 分）

R

| A | B | C |
|---|---|---|
| a | 3 | 8 |
| b | 3 | 4 |
| c | 2 | 4 |
| b | 5 | 8 |
| a | 2 | 4 |
| b | 2 | 4 |

S

| B | C |
|---|---|
| 3 | 4 |
| 2 | 4 |

4. 数据库中存在如下表（总计 10 分）

S (SN, SNAME, CITY)

其中，S 表示供应商，SN 表示供应商代码，SNAME 为供应商名字，CITY 为供应商所在的城市，主码为 SN。SN 取值 (S1, S2, ……)。

P (PN, PNAME, COLOR, WEIGHT)

其中 P 表示零件，PN 为零件代号，PNAME 为零件名字，COLOR 为零件颜色，WEIGHT 为零件重量，主码为 PN。PN 取值 (P1, P2, ……)。

J (JN, JNAME, CITY)



其中，J 表示工程，JN 为工程编号，JNAME 为工程名字，CITY 为工程所在城市，主码为 JN。JN 取值 (J1, J2, ……)。

SPJ (SN, PN, JN, QTY)

其中，SPJ 表示供应关系，SN 是为指定工程提供零件的供应商代号，PN 为所提供的零件代号，JN 为工程编号，QTY 表示提供的零件数量，主码为 SN, PN, JN，外码为 SN, PN, JN。

写出实现以下各题功能的 SQL 语句：(每题 2 分)

- (1)、取出为工程 J1 提供零件 P1 的供应商代号
- (2)、取出供应商代号及他们分别提供的零件总数
- (3)、取出为北京的工程提供红色零件的供应商代号
- (4)、取出上海供应商不提供任何零件的工程的代号
- (5)、取出由 S1 供应商提供蓝色零件的工程名称

### 三、数据库设计题（共 20 分）

1. 根据我们日常学习生活所了解的关于图书出版的知识结合如下的商业描述给出提问的答案，仅限于描述部分的局部应用（即局部的 ER 关系）：

许多教师编写了教材来进行教学工作，教材被录用后由出版社出版。图书出版社出版多种教材，每种教材都有唯一确定的国际唯一标示 ISBN。教材出版后被送到很多的书店里销售。学生所学的课程都有指定的教材和参考用书。这部分的数据可能被用在教材科的信息系统中。

- 1) 画出该图书信息管理数据库的局部 ER 图并注明关系类型（3 分）
- 2) 将 E-R 图转换为等价的关系模型（包括关系名，属性名和主码，外码，主要属性由学生自行构建）。（3 分）

2. 设想你已经为你所喜欢的电视连续剧建立一个通用的数据库，数据库主要存储电视连续剧所属的关联紧密的信息（信息来源是生活中自然的逻辑）。其中主要内容描述如下：

影视数据库设计的实体和实体的固有属性（提示这些不是表中全部的属性,自行扩建）如下：

- ◆ 实体 studios（连续剧工作室），它有 name, address, and website 三个属性。
- ◆ 实体 series（电视连续剧），它有 title, creator, website, startDate, endDate 属性。
- ◆ 实体 actors(演员)，属性有 name, website。
- ◆ 实体 episodes(连续剧中的集)，它所具有的属性 episodeTitle, airdates(放映时间, season, qualityRating(收视率), description, multimediaURL 属性。

- ◆ 实体 characters(电视连续剧中的角色), 它有 name, seriesTitle, seriesCreator, titleOrRole 属性。

主要实体之间的关系如下,未包含的按自然常规逻辑判定:

- ◆ 一个连续剧工作室可以制作出很多个连续剧
  - ◆ 一个连续剧有多个演员来演, 而一个演员也可以出演多个连续剧
  - ◆ 一个连续剧可以有多个角色和集
  - ◆ 每集中可以有很多个角色,而一个角色也可以在很多集中出现。
  - ◆ 一个演员可以演多个角色, 一个角色也可以由多个演员来完成
- 1) 画出该电视连续剧信息管理数据库的 ER 关系图并注明关系类型( 需要自行创建所缺的部分) (6 分)
  - 2) 将 E-R 图转换为等价的关系模型(包括关系名, 属性名和主码, 外码, 需要的属性由学生自行构建)。 (4 分)
  - 3) 分析该关系中属性的依赖关系, 并用规范化的原理来校验所得到的关系, 该最终的关系表达符合哪一级范式? 为什么? (4 分)

#### 四、分析题 (共 5 分)

现有如下关系模式:

(公司编号,分店编号,法人代表,联系电话,分店店长名)

存在以下事实:

不同公司的分店可以具有相同的分店编号

某个公司的某个分店只能有一个唯一的分店店长。

某个公司只能有一个法人代表, 但一个法人代表可以代表多个公司。

法人代表只使用一个公开的联系电话。

- 1.请写出此关系模式 R 中存在的数据库依赖, 并确定此关系模式的码。(1 分)
- 2.判断关系模式 R 是否属于第 2 范式, 请写明判定依据。(2 分)
- 3.请将该关系模式分解到满足 BCNF。(2 分)

## 附录 C 部分参考答案

显示“0610”班的学生人数

```
select count(*) from student where class='0610';
```

查询得过奖学金、同时至少有一门课程成绩在分以上的学生信息，包括学号、姓名和系别名称；

```
select sno, sname, dname from student, depart where student.dno = depart.dno
and sno in ((select sno from scholarship) intersect (select sno from sc where
degree>95));
```

显示所有课程中的最高分的学生学号、姓名和课程号、课程名

```
select student.no, student.name, course.cno, course.cname from student, course,
sc, (select cno, max(degree) maxdegree from sc group by cno) s where sc.cno =
s.cno and sc.degree = s.maxdegree and student.no = sc.no and course.cno = sc.cno;
```

显示选修“数据库原理”课程的成绩高于“06072”号同学成绩的所有同学的记录

```
select * from sc where cno = (select cno from course where cname='数据库原理')
and degree > (select degree from sc where sno='06072' and cno = (select cno from
course where cname='数据库原理'));
```

取出为工程 J1 提供零件 P1 的供应商代号

```
select SN from SPJ where JN='J1' and PN='P1';
```

取出供应商代号及他们分别提供的零件总数

```
select SN, sum(QTY) from SPJ group by SN;
```

取出为北京的工程提供红色零件的供应商代号

```
select SN from SPJ, J, P where SPJ.PN=P.PN and SPJ.JN=J.JN and J.CITY='北京' and
P.COLOR='红';
```

取出上海供应商不提供任何零件的工程代号

```
select JN from SPJ where not exists(select SN from S where S.CITY='上海' and
S.SN=SPJ.SN);
```

取出由 S1 供应商提供蓝色零件的工程名称

```
select JNAME from J where JN in (select JN from SPJ, P where SPJ.PN=P.PN and SN='S1'
and P.COLOR='蓝');
```

## 后 记

获取更多软件学院的学习资料、观看新闻通知、讨论社团爱好、参与跳蚤市场，欢迎来访软院人论坛：<http://www.ssduter.com/>

2012 年数据库考试中 SQL 语句题首次采用了英文描述，此方式或继续沿用，因此复习时可一并参考英文上机题目进行准备。

由于水平所限，文档中难免有不足之处，希望读者给予批评指正。