

# 《数据库系统原理》作业题与参考答案

## 作业题

### PART I 基础篇

#### CHAPTER 1 数据库系统引论

1. {教材 Page 10 习题 3}

#### CHAPTER 2 数据模型

1. {教材 Page 54 习题 7(1) – (4), (6)}

#### CHAPTER 3 SQL 语言

1. {教材 Page 88 习题 2 – 5} (See: [参考答案 1](#))

### Part II 系统篇

#### CHAPTER 4 DBMS 引论

1. 名词解释：(1)事务；(2)数据目录；(3)多线程 DBMS；(4)集中式 DB 与分布式 DB

#### CHAPTER 5 数据库的存储结构

1. 试解释关系 DB 系统中基表数据的三种典型存取路径：(1)索引；(2)簇集；(3)散列。(即基表的四种典型存储机制：(1)表；(2)索引的表；(3)索引簇表；(4)散列簇表)。

#### CHAPTER 6 查询处理与优化

1. 用语法树表示教材 Page 66 例 3-4 查询的代数优化过程。(See: [参考答案 2](#))

#### CHAPTER 7 事务管理

1. {教材 Page 172 习题 6}

2. {教材 Page 172 习题 7}

## CHAPTER 8 数据库的安全与完整性约束

1. {教材 Page 185 习题 1}
2. {教材 Page 185 习题 2}

## CHAPTER 9 触发器与主动数据库系统

1. 职员 (emp) 基表定义见课件。试用 SQL:1999/SQL3 语法定义一个名为 empBandh 的触发器来实现:一旦在 emp 表中删除一个员工的数据,只要此员工的工种不是“bandh”,就在 emp 表中恢复 (插入) 此员工的数据,将其工种 (job) 置为“bandh”,月薪 (sal) 置为 2000.0,佣金 (comm) 置为 NULL,其余属性不变。(See: [参考答案 3](#))

## PART III 应用篇

### CHAPTER 10 数据依赖与关系模式的规范化

1. {教材 Page 225 习题 8} (See: [参考答案 4](#))

### CHAPTER 11 数据库设计

1. 试对四个实体:学生、班级、课程、教师,及它们间的四个联系:组成、选课、任课、班主任进行 E/R 建模,并把 E/R 数据模式转换成关系数据库模式,并写出创建基表的 SQL DDL 语句,同时定义 PK 与 FK、以及你认为必要的其他完整性约束。(See: [参考答案 5](#))

### CHAPTER 12 数据库管理

1. {教材 Page 225 习题 1}

## 参 考 答 案

### 参考答案 1

基表（详细定义见教材）：

student (sno, sname, sex, bdate, height)

course (cno, lhour, credit, semester)

sc (sno, cno, grade)

- (1) 查询身高大于 1.80 米的男生的学号和姓名。

```
SELECT sno, sname
FROM student
WHERE height>1.8 AND sex = '男';
```

- (2) 查询计算机系秋季所开课程的课程号和学分数。

```
SELECT cno, credit
FROM course
WHERE cno LIKE 'cs%' AND semester = '秋';
```

- (3) 查询选修计算机系秋季所开课程的男生姓名、课程号、学分数、成绩。

```
SELECT sc.cno, credit, sname, grade
FROM sc, course, student
WHERE course.cno = sc.cno AND student.sno = sc.sno AND
      sc.cno LIKE 'cs%' AND semester = '秋' AND sex = '男'
ORDER BY sc.cno;
```

- (4) 查询至少选修一门电机系课程的女生的姓名。

```
SELECT sname
FROM student
WHERE sex = '女' AND
      EXISTS ( SELECT *           //相关子查询
               FROM sc
               WHERE sc.sno = student.sno AND
```

sc.cno LIKE 'ee%');

或者：

SELECT sname

FROM student

WHERE sex = '女' AND

sno IN ( SELECT DISTINCT sno //不相关子查询

FROM sc

WHERE cno LIKE 'ee%');

或者用“连接”（不好!）：

SELECT DISTINCT sname

FROM student, sc

WHERE student.sno = sc.sno AND sex = '女' AND cno LIKE 'ee%';

注：此处的 **DISTINCT**，若加，则会丢失其他同姓名而又均选修了电机系课程的学生；若不加，则选多门电机系课程的某学生的姓名重复出现。因此，用“连接”的方法不好！

(5) 查询每位学生已修课程的门数和总平均成绩。

SELECT sc.sno, sname, COUNT(sc.cno), AVG(sc.grade)

FROM sc, student

WHERE student.sno = sc.sno

GROUP BY sc.sno;

注：标准 SQL（见斯坦福书 Page 275）或 Oracle 7 中 SELECT 列表仅可包含下列表达式类型：(1)常数；(2)组函数；(3)与 GROUP BY 子句中相同的列表达式；(4)由 GROUP BY 子句中的列表达式所组成的表达式。因此，上述查询中不能再连接 **student** 表而获得 **sname**。

(6) 查询每门课程选修的学生人数、最高成绩、最低成绩和平均成绩。

SELECT cno, COUNT(sno), MAX(grade), MIN(grade), AVG(grade)

FROM sc

GROUP BY cno;

- (7) 查询所有课程的成绩都在 80 分以上的学生的姓名、学号，并按学号升序排列。

SELECT sname, sno

FROM student

WHERE 80 < ALL ( SELECT grade //相关子查询

FROM sc

WHERE sc.sno = student.sno AND

grade IS NOT NULL)

ORDER BY sno;

注：具体的 RDBMS 对 NULL 值的处理可能不同。

- (8) 查询缺成绩的学生姓名、缺成绩的课程号及其学分数。

SELECT sname, sc.cno, credit

FROM student, sc, course

WHERE grade IS NULL AND

student.sno=sc.sno AND course.cno=sc.cno;

- (9) 查询有（含）一门以上三学分以上课程的成绩低于 70 分的学生的姓名。

SELECT sname

FROM student

WHERE sno IN ( SELECT DISTINCT sno //不相关子查询

FROM sc

WHERE grade < 70 AND

cno IN ( SELECT cno //不相关子查询

FROM course

WHERE credit >= 3 ) );

或者：

SELECT sname

FROM student

```

WHERE EXISTS ( SELECT *    //相关子查询
                FROM sc
                WHERE sc.sno = student.sno AND grade<70 AND
                      cno IN ( SELECT cno    //不相关子查询
                              FROM course
                              WHERE credit >= 3 ) );

```

或者用“连接” (不好!):

```

SELECT DISTINCT sname
FROM student, sc, course
WHERE grade < 70 AND sc.sno = student.sno AND course.cno = sc.cno;

```

注: 此处的 **DISTINCT** 同题(4), 因此, 用“连接”的方法不好!

(10) 查询 1984 – 1986 年出生的姓名、总平均成绩及已修学分数。

```

SELECT sno, AVG(grade), SUM(credit)
FROM sc, course
WHERE course.cno = sc.cno AND
      sno IN ( SELECT sno
                FROM student
                WHERE YEAR(bdate) BETWEEN 1974 AND 1976)
GROUP BY sno;

```

注: 若考虑有同姓名学生, 只能用 ORDER BY sno, 而不能用 ORDER BY name。但此时在 SELECT 子句中就不能出现 name, 见题(5)注。因此, 此题不好!

或者 (也不好!):

```

SELECT sname, student.sno, AVG(grade), SUM(credit)
FROM sc, course, student
WHERE course.cno = sc.cno AND sc.sno = student.sno AND
      YEAR(bdate) BETWEEN 1974 AND 1976
GROUP BY sname, student.sno;

```

## 参考答案 2

基表:

s (sno, sname, sex, bdate, height)

c (cno, lhour, credit, seme)

sc (sno, cno, grade)

查询:

SELECT sname, c.cno, grade FROM s, c, sc

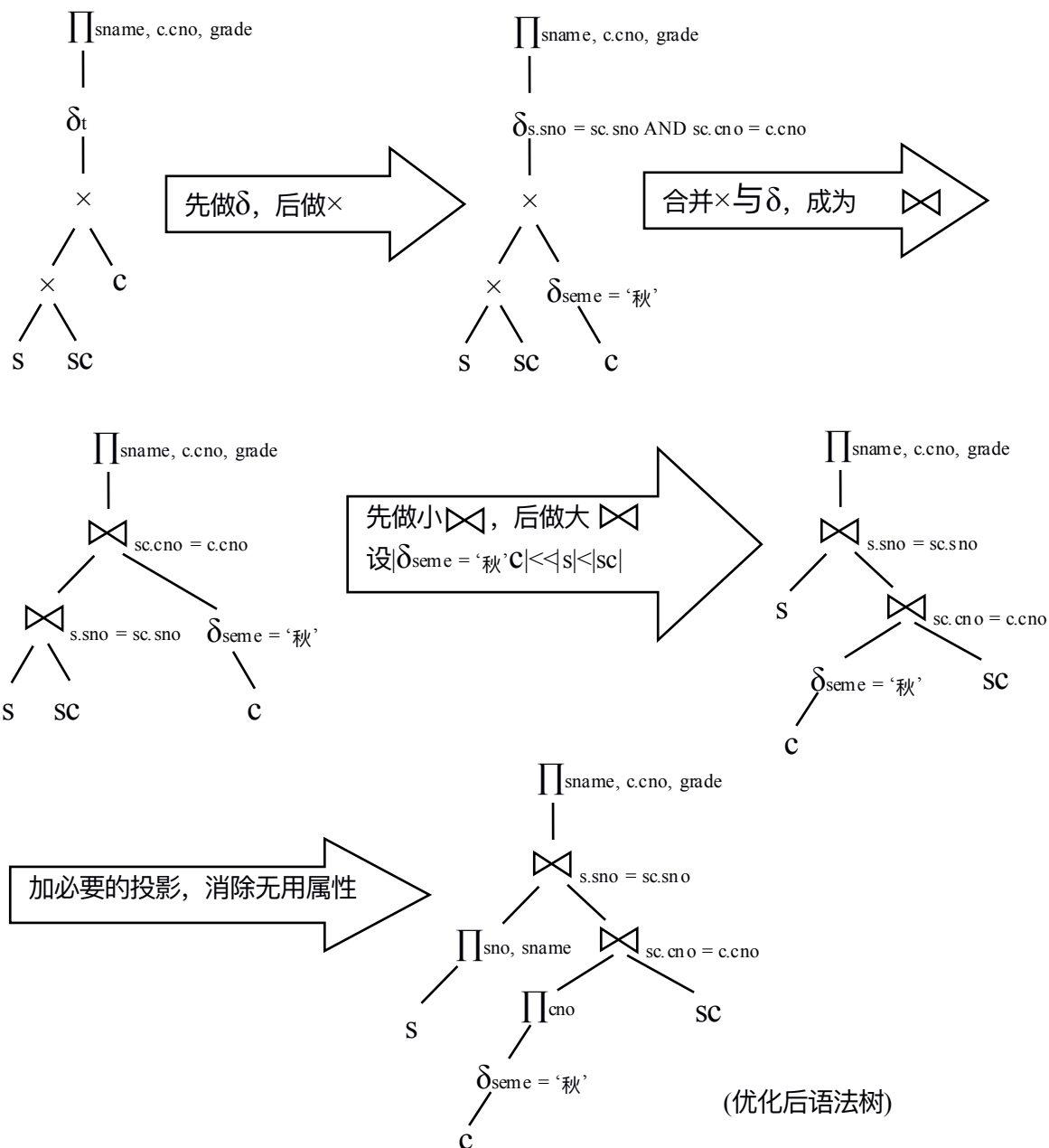
WHERE s.sno = sc.sno AND sc.cno = c.cno AND seme = '秋';

查询可表示成关系代数操作表达式:

$$\Pi_{\text{sname, c.cno, grade}}(\delta_t((s \times sc) \times c))$$

其中:  $t = s.sno = sc.sno \text{ AND } sc.cno = c.cno \text{ AND } seme = \text{'秋'}$

查询可表示成 (原始) 语法树; 其代数优化 (语法树等价变换) 过程为:



优化后的查询可表示成关系代数操作表达式:

$\Pi_{sname, c.cno, grade} (((\Pi_{sno, sname} S) \bowtie_{s.sno = sc.sno} (\Pi_{cno} (\delta_{seme='秋'} C) \bowtie_{c.cno = sc.cno} SC)))$

(Return: [CHAPTER 6 作业题](#))

### 参考答案 3

(职员 (emp) 基表定义: emp (empno, ename, job, mgr, hiredate, sal, comm, deptno) )

CREATE TRIGGER empBandh



AFTER DELETE ON emp

REFERENCING

OLD ROW AS oldrow,

FOR EACH ROW

WHEN oldrow.job <> 'bandh'

BEGIN

INSERT INTO emp

VALUES(oldrow.empno, oldrow.ename, "bandh", oldrow.mgr,  
oldrow.hiredate, 2000.0, NULL, oldrow.deptno)

END;

(Return: [CHAPTER 9 作业题](#))

## 参考答案 4

判别关系模式最高需第几范式？并解释原因。

(1)  $R(ABCD)$ ,  $F=\{B \rightarrow D, AB \rightarrow C\}$ ;

解：KEY={A, B}=AB

非平凡函数依赖  $B \rightarrow D$  中：

决定子 B 不是超键，所以 R 不属于 BCNF；

被决定子 D 不是主属性，所以 R 不属于 3NF；

又，D 部分依赖于 KEY，所以 R 不属于 2NF；

结论：R 属于 1NF。

(2)  $R(ABCDE)$ ,  $F=\{AB \rightarrow CE, E \rightarrow AB, C \rightarrow D\}$ ;

解：KEY=AB 或 E

非平凡函数依赖  $C \rightarrow D$  中：

决定子 C 不是超键，所以 R 不属于 BCNF；

被决定子 D 不是主属性，所以 R 不属于 3NF；

F 中不存在非主属性部分依赖于 KEY；

结论：R 属于 2NF。

(3)  $R(ABCD)$ ,  $F=\{B \rightarrow D, D \rightarrow B, AB \rightarrow C\}$ ;

解：KEY=AB 或 AD

非平凡函数依赖  $B \rightarrow D$  中：

决定子 B 不是超键，所以 R 不属于 BCNF；

但被决定子 D 是主属性；

非平凡函数依赖  $D \rightarrow B$  中被决定子 B 是主属性；

非平凡函数依赖  $AB \rightarrow C$  中决定子 AB 是超键；

结论：R 属于 3NF。

(4)  $R(ABC)$ ,  $F=\{A \rightarrow B, B \rightarrow A, A \rightarrow C\}$ ;

解：KEY=A 或 B

非平凡依赖  $A \rightarrow B$  中决定子 A 是超键；

非平凡依赖  $B \rightarrow A$  中决定子 B 是超键；

非平凡依赖  $A \rightarrow C$  中决定子 A 是超键；

由  $B \rightarrow A, A \rightarrow C$  有  $B \rightarrow C$ ，其中决定子 B 是超键；

即找不到任何冒犯 BCNF 的非平凡依赖；

结论：R 属于 BCNF。

(5)  $R(ABC)$ ,  $F=\{A \rightarrow B, B \rightarrow A, C \rightarrow A\}$ ;

解：KEY=C

非平凡依赖  $A \rightarrow B$  中：

决定子 A 不是超键，所以 R 不属于 BCNF；

被决定子 B 又不是主属性，所以 R 不属于 3NF；

又，F 中不存在非主属性部分依赖于 KEY；

结论：R 属于 2NF。

(6)  $R(ABCD), F=\{A \rightarrow C, D \rightarrow B\};$

解:  $KEY=AD$

非平凡依赖  $A \rightarrow C$  中:

决定子  $A$  不是超键, 所以  $R$  不属于 BCNF;

被决定子  $C$  又不是主属性, 所以  $R$  不属于 3NF;

非主属性  $C$  (及  $B$ ) 部分依赖于  $KEY$ , 所以  $R$  不属于 2NF;

结论:  **$R$  属于 1NF。**

(7)  $R(ABCD), F=\{A \rightarrow C, CD \rightarrow B\};$

解:  $KEY=AD$

非平凡依赖  $A \rightarrow C$  中:

决定子  $A$  不是超键, 所以  $R$  不属于 BCNF;

被决定子  $C$  又不是主属性, 所以  $R$  不属于 3NF;

因  $A \rightarrow C$ , 即非主属性  $C$  部分依赖于  $KEY$ , 所以  $R$  不属于 2NF;

结论:  **$R$  属于 1NF。**

**(补充): 将  $R$  无损分解到 BCNF。**

对  $A \rightarrow C$ , 其冒犯 BCNF 条件, 故将  $R$  无损分解成:  $R_1(AC)$  和  $R_2(ABD)$ 。

因为  $R_1(AC)$  上仅有函数依赖  $A \rightarrow C$ , 此时  $F_1=\{A \rightarrow C\}$ , 其  $KEY=A$ , ( $A$  是超键), 所以  $R_1$  属于 BCNF;

又因为函数依赖  $A \rightarrow C$  和  $CD \rightarrow B$  逻辑蕴含  $AD \rightarrow B$ 。因此,  $R_2(ABD)$  上  $F_2=\{AD \rightarrow B\}$ , 其  $KEY=AD$ , 所以  $R_2$  属于 BCNF;

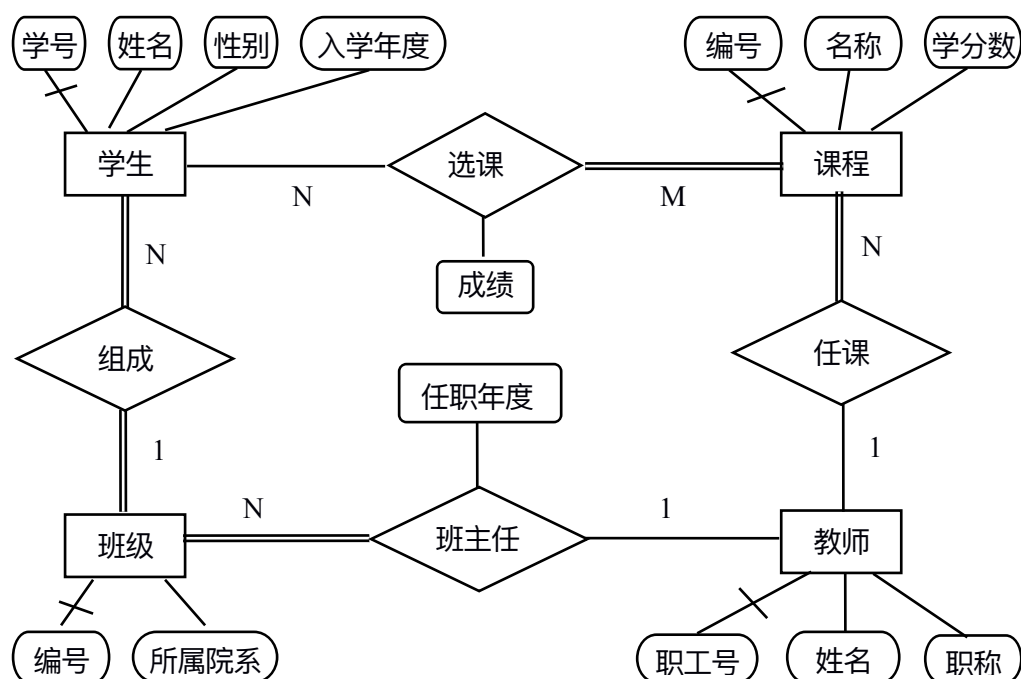
但是  $F_1 \cup F_2 = \{A \rightarrow C\} \cup \{AD \rightarrow B\} \neq F$ , 所以  $\rho=\{R_1(AC), R_2(ABD)\}$  不是对  $R$  的保持依赖分解。

所以说: 一个关系模式可以找到分解到 BCNF 的一个**无损分解**, 但不一定能找到分解到 BCNF 的一个**保持依赖分解**。

(Return: [CHAPTER 10 作业题](#))

## 参考答案 5

E/R 模式：



关系模式：

班级 (编号, 所属院系, 班主任职工号, 任职年度);

学生 (学号, 姓名, 性别, 入学年度, 班级编号);

课程 (编号, 名称, 学分数, 任课教师职工号);

选课 (学号, 课程编号, 成绩);

教师 (职工号, 姓名, 职称);

创建基表的 SQL DDL 语句：略。

(Return: [CHAPTER 11 作业题](#) or [Top of the page](#))