《数据库系统概论》习题课

作为《数据库系统概论》课程的复习内容，我们将在这里例举一些典型的练习题，作为参考。

设有如图所示的关系S、SC和C，试用SQL语言完成下列查询：

1. 检索“程军”老师所授课程的课程号（C#）和课程名（CName）；
2. 查询年龄大于21岁的男学生学号（S#）和姓名（SName）；
3. 查询至少选修了“程军”老师所授全部课程的学生姓名（SName）；
4. 查询“李四”同学不学课程的课程号(C#)；
5. 查询全部学生都选修的课程的课程号（C#）和课程名（CName）；
6. 查询选修课程包含“程军”老师所授课程之一的学生学号（S#）；
7. 查询选修课程号为K1和K2的学生学号（S#）；

S

S#	*SName*	*Age*	*Sex*
001	李四	23	男
002	刘莉	22	女
003	王强	22	男

C

C#	*CName*	*Teacher*
K1	C语言	王旭
K2	数据结构	程军
K3	操作系统	程军

SC

S#	*C#*	*Grade*
001	K1	83
002	K1	85
003	K1	92
002	K2	90
003	K2	84
003	K3	80

关系代数与SQL语言（续一02）

- 检索“程军”老师所授课程的课程号（C#）和课程名（CNa

$$\Pi_{C\#,CName}\left(\sigma_{Teacher='程军'}(C)\right)$$

关系代数表达式：

SQL语言实现：

```
select  C#,  Cname
from  C
where Teacher = ‘程军’
```

关系代数与SQL语言（续一03）

- 查询年龄大于21岁的男学生学号（S#）和姓名（SName）；

关系代数表达式：$\Pi_{S\#,SName}\left(\sigma_{Age>21\wedge Sex='男'}(S)\right)$

SQL语言实现：

```
select  S#,  SName
from  S
where  Age > 21  and  Sex = ‘男’
```

关系代数与SQL语言（续一04）

- 查询至少选修了“程军”老师所授全部课程的学生姓名（SName）；

关系代数表达式：

$$\prod_{SName}\left(S \mathbin{><} \left(\prod_{S\#,C\#}(SC) \div \prod_{C\#}\left(\sigma_{Teacher='程军'}(C)\right)\right)\right)$$

SQL语言实现：(请参考教材P_{113}的例44)

```
select SName from S, SC SC01
where S.S# = SC01.S# and not exists
(select * from C, SC SC02
where C.Teacher = ‘程军’ and not exists
(select * from SC SC03 where SC03.S# =
 SC01.S# and SC03.C# = SC02.C# )
```

关系代数与SQL语言（续一05）

查询“李四”同学不学课程的课程号(C#);

关系代数表达式:

$$\prod_{C\#}(C)-\prod_{C\#}(\sigma_{SName='李四'}(S)\times SC)$$

SQL语言实现:

```
select  C#  from  C
where  not  exists
(select  *  from S,  C
  where  S.Sname = ‘李四’   and
            S.S#  = SC.S#  and  SC.S#  =  C.C#)
```

关系代数与SQL语言（续－06）

查询全部学生都选修的课程的课程号（C#）和课程名（CName）；

关系代数表达式：

$$\prod_{C\#,CName}\left(C \mathbin{><} \left(\prod_{S\#,C\#}(SC) \div \prod_{S\#}(S)\right)\right)$$

SQL语言表达式：（请参考教材P_{113}的例43）

```
select C#, CName from C where not exists
( select * from S where not exists
(select * from SC where S.S# = S#
            and C# = C.C#));
```

关系代数与SQL语言（续－07）

查询选修课程包含“程军”老师所授课程之一的学生学号（S#）；

关系代数表达式：

$$\prod_{S\#}\left(SC \mathbin{><} \prod_{C\#}\left(\sigma_{Teacher='程军'}(C)\right)\right)$$

SQL语言表达式：

```
select S# from SC where C# in
    ( select C# from C
        where Teacher = ‘程军’ );
```

关系代数与SQL语言（续一08）

- 查询选修课程号为K1和K2的学生学号（S#）；

关系代数表达式：

$$\prod_{S\#,C\#}(SC) \div \prod_{C\#}(\sigma_{C\#=K1 \vee C\#=K2}(C))$$

SQL语言表达式：

```
select  S#  from  SC
where  C# = K1 and  C# = K2;
```

规范化理论的应用实例01

如下图所示，请问关系R为第几范式？是否存在操作异常？若存在，则将其分解为高一级范式。分解完的高级范式中是否可以避免分解前关系中存在的操作异常呢？

规范化理论的应用实例01 （续一01 ）

R

工程号	材料号	数量	开工日期	完工日期	价格
P1	M1	4	9805	9902	250
P1	M2	6	9805	9902	300
P1	M3	15	9805	9902	180
P2	M1	6	9811	9912	250
P2	M4	18	9811	9912	350

规范化理论的应用实例01 （续－02）

解：它为1NF。因为该关系的候选关键字为（工程号，材料号），而非主属性开工日期和完工日期部分函数依赖于候选关键字的子集－－工程号，即：

$$\left.\begin{array}{l}(\text{工程号，材料号}) \xrightarrow{P} \text{开工日期;} \\ (\text{工程号，材料号}) \xrightarrow{P} \text{完工日期。}\end{array}\right\} \therefore \text{它不是2NF。}$$

它存在操作异常，如果工程项目确定后，若暂时未用到材料，则该工程的数据因缺少关键字的一部分(材料号)而不能插入到该数据库中，出现插入异常。若某工程完工，则删除该工程的操作也可能丢失材料方面的信息。

规范化理论的应用实例01 （续－03）

将其中的部分函数依赖分解为一个独立的关系，则产生如下图所示的两个2NF关系子模式：

分解后，新工程确定后，尽管还未用到材料，该工程数据可在关系R2中插入。某工程数据删除时，仅对关系R2操作，也不会造成材料方面的信息丢失。

规范化理论的应用实例01 （续一04 ）

R1

工程号	材料号	数量	价格
P1	M1	4	250
P1	M2	6	300
P1	M3	15	180
P2	M1	6	250
P2	M4	18	350

R2

工程号	开工日期	完工日期
P1	9805	9902
P2	9811	9912

规范化理论的应用实例02

请问如下图所示的关系SC为第几范式？是否存在插入、删除异常？若存在，则说明示在什么情况下发生？发生的原因是什么？将它分解为高一级的范式，分解后的关系能否解决操作异常问题呢？

规范化理论的应用实例02（续一01）

关系： ***SC***

SNo	CNo	CTitle	TName	TLoc	Grade
80152	C1	操作系统	王平	D1	70
80153	C2	数据库	高升	D2	85
80154	C1	操作系统	王平	D1	86
80154	C3	算法分析	杨杨	D3	72
80155	C4	数据结构	高升	D2	92

规范化理论的应用实例02（续一02）

解：该关系SC为1NF。

它存在插入、删除异常操作。当增设一门新课程时，因还没有学生选修，则缺少主码的一部分SNo而不能执行插入操作；当所有学生退选某门课程而进行删除操作时，会将不该删除的课程信息删除掉。

规范化理论的应用实例02（续一03）

SC关系中存在插入和删除操作异常的原因在于，该关系的候选关键字为(SNo，CNo)，其中仅有非主码属性Grade完全函数依赖于(SNo，CNo)，其他非主码属性CTitle、TName、TLoc都只是部分函数依赖于(SNo，CNo)。分解后的关系模式如下图所示。

规范化理论的应用实例02（续一04）

关系：***SG***

SNo	CNo	Grade
80152	C1	70
80153	C2	85
80154	C1	86
80154	C3	72
80155	C4	95

关系：***CT***

CNo	CTitle	TName	TLoc
C1	操作系统	王平	D1
C2	数据库	高升	D2
C3	算法分析	杨杨	D3
C4	数据结构	高升	D2

规范化理论的应用实例02（续—05）

分解后的两个关系子模式都为2NF，并且解决了先前存在的插入、删除异常操作。当增设一门新课程时，可将数据插入到CT表中，当所有学生退选某门课程时，只需删除SG表中的有关记录，而该课程的有关信息仍保留在CT表中。

规范化理论的应用实例02（续－06）

分解成2NF后的CT关系中仍存在插入、删除操作异常。若有一个新教师报到，需将其有关数据插入CT表，还是会存在插入异常。当取消某门课程而删除CT表中的一条记录时，会将不该删除的教师的有关信息删除。CT表中出现操作异常的原因是该关系中存在非主属性对候选关键字的传递函数依赖。

CNo→TName，TName↛CNo，TName→TLoc，
所以CNo→Tloc（传递函数依赖）。

规范化理论的应用实例02（续一07）

将CT表进一步分解为如下图所示的两个关系，可以解决上述操作中存在的异常。

关系：*Course*

CNo	CTitle	TName
C1	操作系统	王平
C2	数据库	高升
C3	算法分析	杨杨
C4	数据结构	高升

关系：*Instructor*

TName	TLoc
王平	D1
高升	D2
杨杨	D3

简单的数据库设计实例

设有如下实体：
学生：学号、单位、姓名、性别、年龄、选修课程名；
课程：编号、课程名、开课单位、任课教师号；
教师：教师号、姓名、性别、职称、讲授课程编号；
单位：单位名称、电话、教师号、教师名；
上述实体中存在如下联系：

（1）一个学生可选修多门课程，一门课程可为多个学生选修；

（2）一个教师可讲授多门课程，一门课程可为多个教师讲授；

简单的数据库设计实例（续一01）

（3）一个单位可有多个教师，一个教师只能属于一个单位。

试完成如下工作：

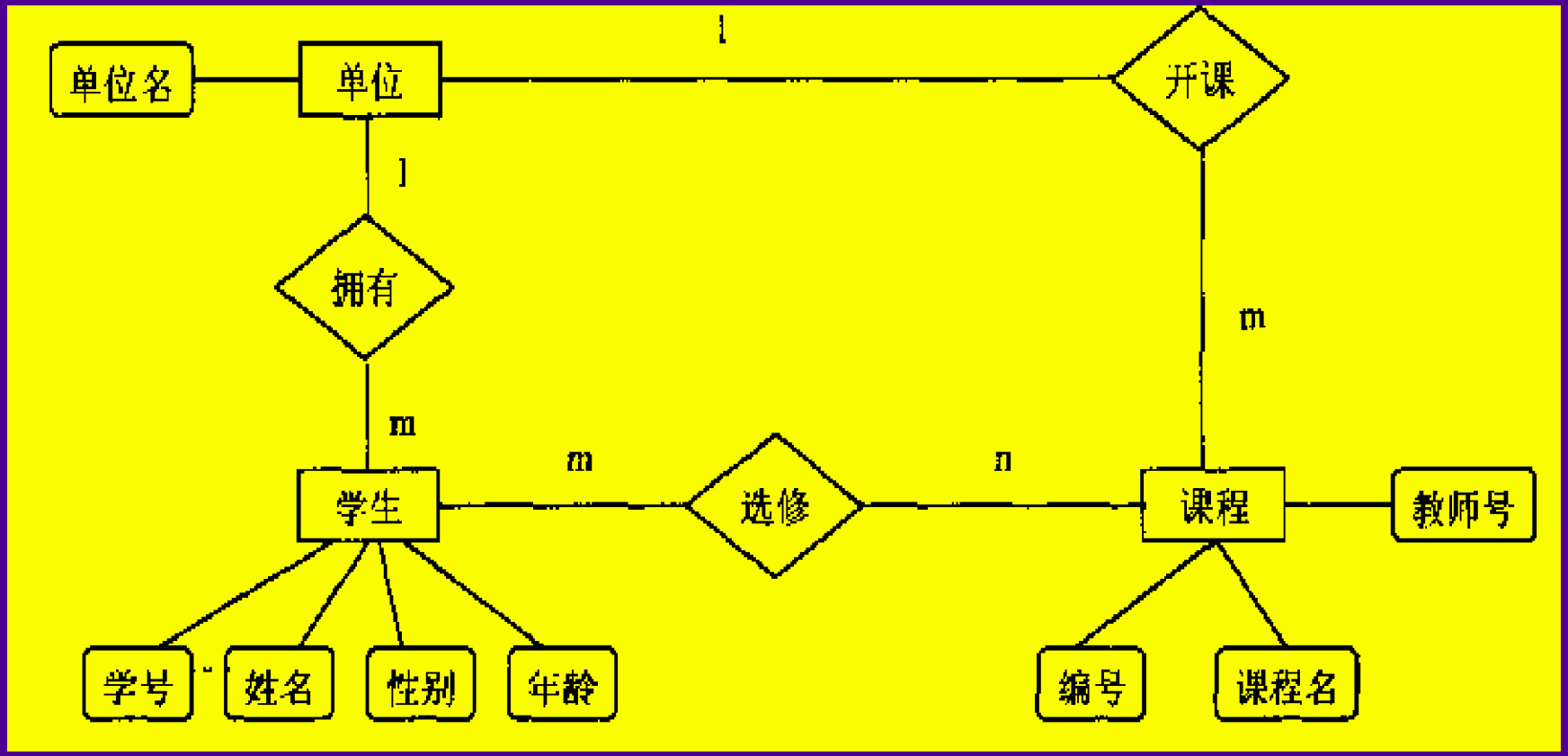
（1）分别设计学生选课和教师任课两个局部信息的结构E－R图。

（2）将上述设计完成的E－R图合并成一个全局E－R图。

（3）将该全局E－R图转换为等价的关系模型表示的数据库逻辑结构。

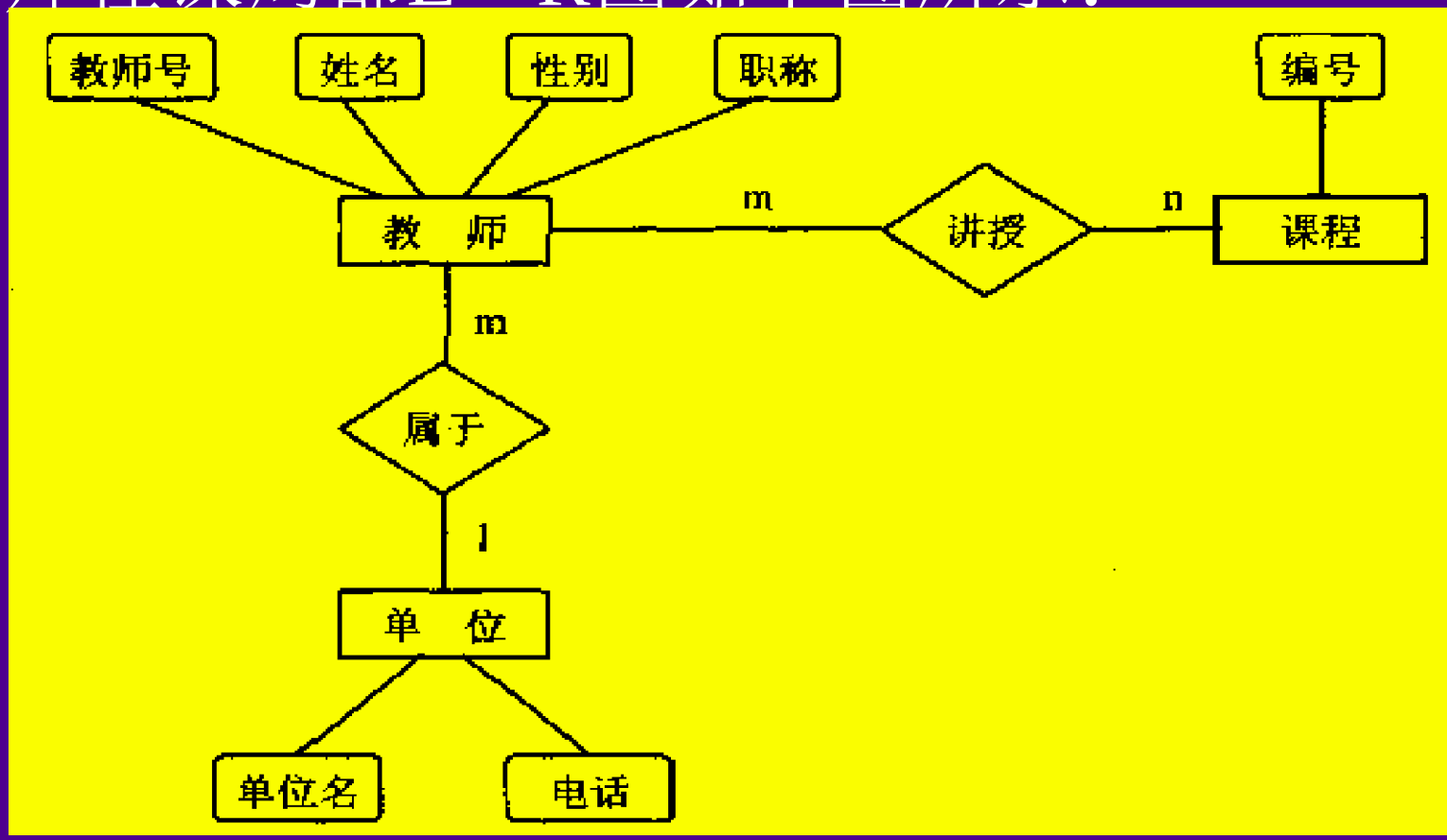
简单的数据库设计实例（续—02）

解：（1）学生选课局部E—R图如下图所示：



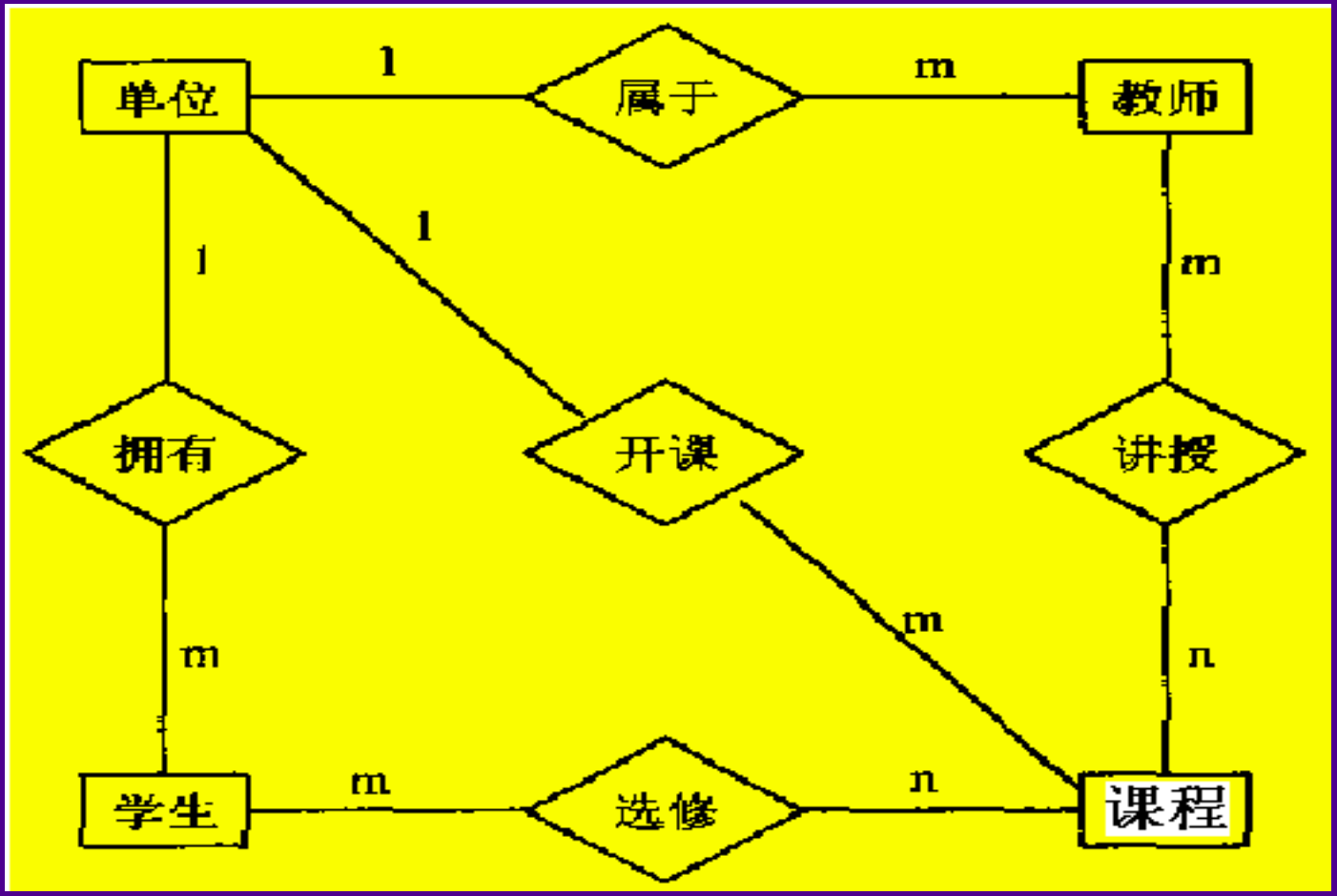
简单的数据库设计实例（续—03）

教师任课局部E—R图如下图所示：



简单的数据库设计实例（续—04）

合并的全局E—R图如下图所示：



简单的数据库设计实例（续—05）

为逐免图形复杂，我们没有在E—R图中给出每个实体的属性，所以在这里详细地给出各实体属性：

单位：单位名、电话；

学生：学号、姓名、性别、年龄；

教师：教师号、姓名、性别、职称；

课程：编号、课程名；

简单的数据库设计实例（续—06）

前面的全局E—R图转换为等价的关系模型表示的数 据库逻辑结构如下：

单位（单位名，电话）

教师（教师号、姓名、性别、职称、单位名）

课程（课程编号，课程名，单位名）

学生（学号，姓名，性别，年龄，单位名）

讲授（教师号，课程编号）

选修（学号、课程编号）

名词——数据独立性

数据库的数据独立性：

数据独立性表示应用程序与数据库中存储的数据之间不存在依赖关系。包括逻辑数据独立性和物理数据独立性。

逻辑数据独立性是指局部逻辑数据结构（外视图）与全局逻辑数据结构（概念视图）之间的独立性。当数据库的全局逻辑数据结构（概念视图）发生变化（数据定义的修改、数据之间联系的变更或增加新的数据类型等）时，它不影响某些局部的逻辑结构的性质，应用程序不必修改。

名词——数据独立性（续）

物理数据独立性是指数据的存储结构与存取方法（内视图）改变时，对数据库的全局逻辑结构（概念视图）和应用程序不必作修改的一种特性，也就是说，数据库数据的存储结构与存取方法独立。

数据独立性的好处是，数据的物理存储设备更新了，物理表示及存取方法改变了，但数据的逻辑模式可以不改变。数据的逻辑模式改变了，但用户的模式可以不改变，因此应用程序也可以不变。这将使程序维护容易。另外，对同一数据库的逻辑模式，可以建立不同的用户模式，提高数据共享性，使数据库系统有较好的可扩充性。给DBA维护、改变数据库的物理存储提供方便。

简答——DBA的职责

DBA的职责是：

- ⇨ 决定DB中的信息内容和结构
- ⇨ 决定DB的存储结构和存取策略
- ⇨ 定义数据的安全性要求和完整性约束条件
- ⇨ 监控数据哼的使用扣运行

简答——数据字典的作用

答：数据字典的任务就是管理有关数据的信息，所以又称为“数据库的数据库”．它的任务主要有：

- 描述数据库系统的所有对象，并确定其属性．如一个模式中包含的记录型与一个记录型包含的数据项：用户的标识、口令；物理文件名称、物理位置及其文件组织方式等。数据字典在描述时赋给每个对象一个惟一的标识。
- 描述数据库系统对象之间的各种交叉联系。如哪个用户使用哪个子模式，哪些模式或记录型分配在哪些区域及对应于哪些物理文件、存储在何种物理设备上。

简答——数据字典的作用（续）

- 登记所有对象的完整性及安全性限制等。
- 对数据字典本身的维护、保护、查询与输出。

数据字典的主要作用是：

- 供数据库管理系统快速查找有关对象的信息。数据库管理系统在处理用户存取时，要经常查阅数据字典中的用户表、子模式表和模式表等。
- 供数据库管理员查询，以掌握整个系统的运行情况。
- 支持数据库设计与系统分析。

简答——自然连接与等值连接

等值连接和自然连接的表示方法相同，都记为：R ⋈ S；自然连接是除去重复属性的等值连接。两者之间的区别和联系如下：

- 自然连接一定是等值连接．但等值连接不一定是自然连接。
- 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。
- 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。

名词——死锁、活锁

死锁：

活锁：

略

简答——数据库转储中日志文件的作用和注意事项

略，请同学们参考教材。

简答——预防和解除死锁的方法

略，请同学们参考教材。

答疑时间及考试安排

答疑时间：12月5号 08:00 ~ 16:00

答疑地点：G802

考试时间：16周的周三 10:00 ~ 12:00

考试地点：A52

Course is over!

Thanks for all!!!

例：查询至少选修了1号课程和3号课程的学生号码。

Sno,Cno(SC) ÷K	Cno
{95001}	1
	3

例：查询选修了全部课程的学生号码和姓名。

Sno,Cno(SC) ÷ Cno(Course) Sno, Sname (Student)

关系除运算

给定关系$R(X,Y)$和$S(Y,Z)$，其中X,Y,Z为属性组。R中的Y与S中的Y可以有不同的属性名，但是必须出自于相同的域集。R与S的除运算得到一个新的关系$P(X)$，P是R中满足下列条件的元组在X属性列上的投影：元组在X上分量值x的象集Y_x包含S在Y上的投影的集合。记作：

$$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_y(S) \subseteq Y_x\}$$

其中Y_x为x在R中的象集，$x = t_r[X]$。

除运算适合于一些求包含关系、或者含有短语“全部”的查询操作。

除运算实例

设关系R和S如下：则R÷S的结果如下：

R		
A	B	C
A_1	B_1	C_2
A_2	B_3	C_7
A_3	B_4	C_6
A_1	B_2	C_3
A_4	B_6	C_6
A_2	B_2	C_3
A_1	B_2	C_1

S		
B	C	D
B_1	C_2	D_1
B_2	C_1	D_1
B_2	C_3	D_2

R÷S
A
A_1

设有如图所示的关系S、SC和C，试用SQL语言完成下列查询：

S

S#	*SName*	*Age*	*Sex*
001	李四	23	男
002	刘莉	22	女
003	王强	22	男

SC

S#	*C#*	*Grade*
001	K1	83
002	K1	85
003	K1	92
002	K2	90
003	K2	84
003	K3	80

C

C#	*CName*	*Teacher*
K1	C语言	王旭
K2	数据结构	程军
K3	操作系统	程军

查询至少选修了“程军”老师所授全部课程的学生姓名（SName）；

```
select SName
from S
where not exists
    (select *
     from C
where C.Teacher = '程军'  and not exists
(select* from SC
where S.S# = SC.S#
and SC.C# = C.C# ))
```