注: 每道题可能会有多种正确答案，本答案未覆盖所有正确答案。供参考。

作业一 练习参考答案

第一章 课后习题

**1.8 列出文件处理系统和DBMS的四个主要区别**

* 两个都是由一个互相关联的数据的集合和一组用以访问这些数据的程序组成的。DBMS：物理和逻辑访问数据，文件：物理访问数据。
* DBMS：被授权的程序可以访问存储的逻辑数据，数据可以被多个程序访问，减少数据的冗余 文件：一个程序所写的数据可能不能被另一个程序访问。
* DBMS 设计为更灵活地访问数据(如，查询)，文件处理系统则被设计为特定的程序访问数据(如，已编译的程序)。
* DBMS 允许多个用户同时访问相同的数据。文件处理系统一般允许一个或多个程序同时访问不同的数据文件。在文件处理系统中，一个文件只有在两个程序都只有只读权限的时候才可以同时被两个程序并行访问。

**1.9 解释物理数据独立性的概念，以及它在数据库系统中的重要性**

物理独立性是指用户的应用程序与存储在磁盘上的数据库中数据是相互独立的。

物理独立性使应用程序与存储在磁盘上的数据相分离，应用程序不依赖于物理模式，因此物理模式改变了它们也无需重写。

**1.12 解释****用图1‐4中的表来设计会导致哪些问题**

数据重复浪费存储空间，同时也带来更新异常、插入异常和删除异常。

• 如果一个系有多于一个教师，则building名和budget就会重复多次。Update 某老师的building名和budget就会造成数据不一致。

• 一个系需要至少有一个教师才能在表中存储building和budget信息。当没有教师时也可以用Null值，但空值带来处理的复杂性。

• 如果一个系中的所有教工都删除了，building和budget信息也丢失了。理想的解决方案，系的信息与instructor分开独立存储成数据库中的一张表， 与instructor中记录的修改无关，也没有空值问题。

**-------------------------------------------------------**

第二章 课后习题

**2.9**

branch: branch\_name

customer: customer\_name

loan: loan\_number

borrower: customer\_name, loan\_number

account: account\_number

depositor: customer\_name, account\_number

loan: branch\_name

borrower: customer\_number, loan\_number

account: branch\_name

depositor: customer\_name, account\_number

**2.10**

**如果一个学生可以有多位指导老师，则s\_id就不能唯一标识元组，可以加上i\_id, 即s\_id,i\_id作主码。**

**--------------------------------------------------**

第三章课后习题

3.22

where(

( select count(title) from course)

=

(select count (distinct title) from course)

)

3.24

有多种方式写这个查询。一种方法是在where子句中用子查询，在其中一个子查询中在from子句中使用二级子查询，如下：

select distinct dept\_name d

from instructor

where

(select sum(salary)

from instructor

where dept\_name = d)

>=

(select avg(s)

from

(select sum(salary) as s

from instructor

group by dept\_name))

**--------------------------------------------------**

第四章 课后习题：

4.1

a.

**select** ID, *name*,**count**(*course id*, *section id*, *year*,*semester*) **as** ’Number of sections’

**from** *instructor* **natural left outer join** *teaches*

select ID, name,count(course\_id, section\_id, year,semester) as ’Number of sections’

from instructor natural left outer join teaches

group by ID, name ;

上面的查询不应该写成count(\*)，因为count(\*)也计算null值。也可以用count(section\_id)或count其他在teaches中(不在instructor中)的属性。

b.

**select** ID, *name*,

(**select count**(\*) **as** ’Number of sections’

**from** *teaches T* **where** *T.id* = *I.id*)

**from** *instructor I*

c.

**select** course\_id, section\_id, ID, **decode**(*name*, **NULL**, ’−’, *name*)

**from** (*section* **natural left outer join** *teaches*) **natural left outer join** *instructor*

**where** *semester*=*’Spring’* and *year*= 2010

d.

**select** *dept\_ name*, **count**(ID)

**from** *department* **natural left outer join** *instructor*

**group by** *dept\_ name*

4.2

a.

**select** \* **from** *student* **natural join** *takes*

**union**

**select** ID, *name*, *dept\_ name*, *tot\_ cred*, **NULL**, **NULL**, **NULL**, **NULL**, **NULL**

**from** *student S1* **where not exists**

(**select** ID **from** *takes T1* **where** *T1.id* = *S1.id*)

b.

(**select** \* **from** *student* **natural join** *takes*)

**union**

(**select** ID, *name*, *dept\_ name*, *tot\_ cred*, **NULL**, **NULL**, **NULL**, **NULL**, **NULL**

**from** *student S1*

**where not exists**

(**select** ID **from** *takes T1* **where** *T1.id* = *S1.id*))

**union**

(**select** ID, **NULL**, **NULL**, **NULL**, *course id*, *section id*, *semester*, *year*, *grade*

**from** *takes T1*

**where not exists**

(**select** ID **from** *student S1* **where** *T1.id* = *S1.id*))

4.13

有一或多个学生没有选任何课程。