第 3 章　关系数据库模式设计

☞本章目标

本章主要介绍关系规范化问题的提出、函数依赖的概念、关系模式的键码、关系的规范化、模式分解的优劣和典型案例分析等内容，学习并掌握好函数依赖的概念、关系模式的键码、关系的规范化和典型案例分析尤为重要，不仅能加深对本章内容的理解，而且有利于在后续的关系数据库逻辑结构设计中对一组关系模式进行优化时提供技术支持。

3.1　关系规范化问题的提出

在关系数据库设计中，如何把现实世界表达成关系数据库模式，并且这种模式设计是合理和有效的，这些是人们一直十分关注的问题。

关系数据库模式是若干关系模式的集合，所谓关系数据库的模式设计实际上就是从多种可能的组合中选取一个合适的或者说性能好的关系模式集合作为关系数据库模式。

为了对关系模式集合的性能好坏有一个直观的认识，我们用一个实例组成不同的关系模式集合，施加不同的影响，以便评价关系数据库模式设计的优劣。

【例 3.1】某高校要建立一个关系数据库来描述学生和系的一些情况，面临的对象有：学生的学号（SNO）、学生的姓名（SNAME）、系的名称（DEPT）、系的负责人（MN）、学生选修的课程号（CNO）、课程名称（CNAME）和学生选修课的成绩（GRADE）。由现实世界的已知事实可以得知上述对象之间有如下联系：

- 一个系有若干个学生，但一个学生只属于一个系；
- 一个系只有一个负责人；
- 一个学生可以选修多门课程，每门课程应有许多学生选修；
- 每个学生选修的每一门课程都有一个成绩。

根据上述情况，我们至少可以考虑以下两种关系数据库模式设计的选择方案：

方案 1：采用一个总的关系模式，即

SA(SNO,SNAME,DEPT,MN,CNO,CNAME,GRADE)

方案 2：采用 4 个关系模式，即

S(SNO,SNAME,DEPT)、D(DEPT,MN)、SC(SNO,CNO,GRADE)和 C(CNO,CNAME)

比较起来，第一个方案可能带来下列问题。

1．数据冗余

如果某个学生选修多门课程，则由于每选修一门课程必须存储一个数据记录，因此他的姓名及其所在系的信息将被重复保存。

2．更新异常或潜在的不一致

由于数据存储冗余，当更新某些数据项（如学生所在的系）时，有可能一部分涉及的元

组被修改而另一部分元组却没有被修改，这就造成存储数据的不一致。

3．插入异常

如果一个系刚刚成立，尚无学生或者虽然有了学生但尚未安排课程，就无法把这个系及其负责人的信息存入数据库，这是因为在关系模式 SA 中，主键码为（SNO,CNO），而关系模型的实体完整性约束不允许主键码属性为空值，因此在学生未选修课程或系里未分配学生之前，相应元组无法插入。

4．删除异常

如果某个系的学生全部毕业了，我们在删除该系全体学生信息的同时，把这个系及其负责人的信息也一同删除了，这显然是人们所不希望的。

由于上述几个问题，方案 1 不是一个好的关系数据库模式设计，而在方案 2 中这些问题都不存在，因而方案 2 在性能上优于方案 1。一个好的模式应具有较少的数据冗余，同时不会发生插入异常、删除异常和更新异常。上述关系模式中的异常统称为存储异常。现在的问题在于：产生这种存储异常的根源何在？

对例 3.1 的两个关系模式设计方案进行对比研究可以发现，存储异常的存在是与每个关系模式内部各属性值之间的内在相关性直接有关的。在关系模式 SA 中，键码为（SNO,CNO），它的值唯一地决定了其他所有属性的值，形成一种函数依赖关系，即属性 SNAME、DEPT、MN、CNAME、GRADE 的值都函数依赖于键码。但另一方面，这些属性对于键码的函数依赖程度又有所不同。GRADE 是真正函数依赖于整个键码的，而 SNAME、DEPT 的值则实际上只受 SNO 的影响而与 CNO 无关，即只是部分函数依赖于键码（SNO,CNO），而 MN 直接函数依赖于 DEPT 而间接传递函数依赖于 SNO，正是这种部分函数依赖和传递函数依赖造成了上述的存储异常，从而导致了人们对这些数据依赖关系的深入研究。

3.2　函数依赖的概念

数据依赖是现实世界事物之间的相互关联性的一种表达，是属性固有语义的体现。人们只有对一个数据库所要表示的现实世界进行认真的分析，才能归纳出与客观事实相符合的数据依赖。数据依赖中最常见、最重要的一种依赖是函数依赖，下面介绍函数依赖的概念。

3.2.1　函数依赖定义

定义 3.1　设 $R(U)$是属性集 U 上的关系模式，X、Y 是 U 的子集。若对于 $R(U)$的任意一个可能的关系 r，r 中不可能存在两个元组在 X 上的属性值相等、在 Y 上的属性值不等，则称“X 函数决定 Y”或“Y 函数依赖于 X”，记作 $X \rightarrow Y$。

一般来讲，只能根据语义来确定一个函数依赖。根据例 3.1 语义可得出 SA 的函数依赖：F={SNO→SNAME,SNO→DEPT,DEPT→MN,CNO→CNAME,(SNO,CNO)→GRADE}。又如，“姓名→年龄”只有在没有同名人的条件下成立，如果允许有相同名字，则年龄就不再函数依赖于姓名了。设计者也可以对现实世界做出不允许同名人出现的强制规定，但在一般情况下这是不合情理的。解决的办法是：用“学号→年龄”来代替“姓名→年龄”。

注意：函数依赖不是指关系模式 R 的某个或某些关系满足的约束条件，而是指 R 的一切关系均要满足的约束条件。例如，学生选课 SC(SNO,CNO,GRADE)关系模式，假定在当前的

记载中，每个学生只选了一门课程，我们也不能就此断言，SNO 的属性值可以唯一地确定 CNO 值，因为当前每个学生只选一门课的事实并不限定他只能选一门课，只有当制度规定每个学生只能选一门课时，上述论断才真正构成一个函数依赖。

下面介绍一些记号和术语。

① $X \rightarrow Y$，但 $Y \subset X$，则称 $X \rightarrow Y$ 是非平凡函数依赖。若不特别声明，我们总是讨论非平凡函数依赖。

② 若 $X \rightarrow Y$，$Y \rightarrow X$，则记作 $X \longleftrightarrow Y$。

③ 若 Y 不函数依赖于 X，则记作 $X \nrightarrow Y$。

3.2.2 完全函数依赖和部分函数依赖

定义 3.2 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 $X' \nrightarrow Y$，则称 Y 对 X 完全函数依赖，记作

$$X \xrightarrow{f} Y$$

在 $R(U)$中，如果 $X \rightarrow Y$，并且存在 X 的一个真子集 X_0，使得 $X_0 \rightarrow Y$，则称 Y 对 X 部分函数依赖，记作

$$X \xrightarrow{p} Y$$

上述完全函数依赖和部分函数依赖定义可以用图 3.1 表示。

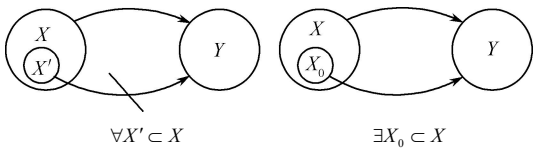


图 3.1 完全函数依赖和部分函数依赖图形表示

在 SC(SNO,CNO,GRADE)中，(SNO,CNO)→GRADE，但 SNO↛GRADE，CNO↛GRADE，∅↛GRADE，故

$$(\text{SNO,CNO}) \xrightarrow{f} \text{GRADE}$$

在 SA(SNO,SNAME,DEPT,MN,CNO,CNAME,GRADE)中，(SNO,CNO)→SNAME，但 SNO→SNAME，故

$$(\text{SNO,CNO}) \xrightarrow{p} \text{SNAME}$$

回顾前面的讨论可以看出，部分函数依赖的存在是关系模式产生存储异常的一个内在原因。另一类值得特别注意的函数依赖是传递函数依赖，它是导致关系模式存储异常的另一个原因。

3.2.3 传递函数依赖

定义 3.3 在 $R(U)$中，如果 $X \rightarrow Y$ $(Y \subset X)$，$Y \nrightarrow X$，$Y \rightarrow Z$，则称 Z 对 X 传递函数依赖，记作 $X \xrightarrow{t} Z$。

加上条件 $Y \nrightarrow X$，是因为如果 $Y \rightarrow X$，则 $X \longleftrightarrow Y$，实际上是 $X \rightarrow Z$，而不是传递函数依赖。

上述传递函数依赖定义可以用图 3.2 表示。

现在考虑关系模式 R（学号，系，系负责人），在该模式中显然存在下列函数依赖：

$$F=\{\text{学号} \rightarrow \text{系}，\text{系} \rightarrow \text{系负责人}\}$$

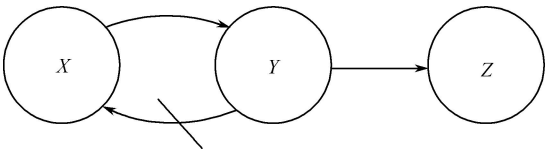


图 3.2　传递函数依赖图形表示

因为每个系有多个学生，所以系和系负责人的信息就要多次重复存储。另外，若该系学生全部毕业，则系和系负责人之间的从属关系数据就无法保留，造成这种数据冗余和更新异常的主要原因是系负责人这一属性对于学号不是直接函数依赖，而是一种传递函数依赖。

3.2.4　函数依赖规则

若已知关系 R（A，B 和 C），它满足函数依赖 $A\rightarrow B$ 和 $B\rightarrow C$，则可断定 R 也满足 $A\rightarrow C$。为了证明 $A\rightarrow C$，需要考察 R 的任意两个在属性 A 上取值一致的元组，证明它们在属性 C 上也取值一致。

设两个在属性 A 上取值一致的元组(a,b_1,c_1)和(a,b_2,c_2)，由于 R 满足 $A\rightarrow B$，并且这两个元组在 A 上一致，故它们在 B 上也必然一致，即 $b_1=b_2$。所以这两个元组实际上就是(a,b,c_1)和(a,b,c_2)，其中 $b=b_1=b_2$。同样，由于 R 满足 $B\rightarrow C$，而且两个元组在 B 上一致，则它们必然在 C 上也一致，即 $c_1=c_2$。至此，我们已经证明了关系模式 R 的任意两个在 A 上一致的元组在 C 上也一致，而这就是函数依赖 $A\rightarrow C$。下面介绍函数依赖的 3 个推理规则。

1．分解／合并规则

① 把一个函数依赖 $A_1A_2\cdots A_n\rightarrow B_1B_2\cdots B_m$ 用一组函数依赖 $A_1A_2\cdots A_n\rightarrow B_i(i=1,2,\cdots,m)$来代替，这种转换称为“分解规则”。

② 把一组函数依赖 $A_1A_2\cdots A_n\rightarrow B_i(i=1,2,\cdots,m)$用一个函数依赖 $A_1A_2\cdots A_n\rightarrow B_1B_2\cdots B_m$ 来代替，这种转换称为“合并规则”。

2．平凡依赖规则

在介绍平凡依赖规则之前，先介绍平凡函数依赖和非平凡函数依赖这两个概念。

设 $A=\{A_1A_2\cdots A_n\}$，$B=\{B_1B_2\cdots B_m\}$，对于函数依赖 $A_1A_2\cdots A_n\rightarrow B_1B_2\cdots B_m$：

① 如果 B 是 A 的子集，则称该依赖为平凡函数依赖。

② 如果 B 中至少有一个属性不在 A 中，则称该依赖为非平凡函数依赖。

③ 如果 B 中没有一个属性在 A 中，则称该依赖为完全非平凡函数依赖。

例如，对于关系 Student(SNO,SNAME,SDEPT,MNAME,CNAME,GRADE)，其属性含义依次为学号、姓名、所在系、系主任姓名、课程名、成绩。我们先列举 3 个函数依赖：

SNO, CNAME, GRADE → CNAME, GRADE，此函数依赖属于平凡函数依赖；

SNO, CNAME→CNAME, GRADE，此函数依赖属于非平凡函数依赖；

SNO, CNAME→SNAME, GRADE，此函数依赖属于完全非平凡函数依赖。

如果函数依赖右边的属性中有一些也出现在左边，那么可以将右边的这些属件删除，即函数依赖 $A_1A_2\cdots A_n\rightarrow B_1B_2\cdots B_m$ 等价于 $A_1A_2\cdots A_n\rightarrow C_1C_2\cdots C_K$，其中 $C=\{C_1C_2\cdots C_K\}$是 B 的子集，但不在 A 中出现。我们称这个规则为“平凡依赖规则”。

3．传递规则

传递规则能将两个函数依赖级联成一个新的函数依赖。

如果 $A_1A_2\cdots A_n \rightarrow B_1B_2\cdots B_m$ 和 $B_1B_2\cdots B_m \rightarrow C_1C_2\cdots C_K$ 在关系 R 中成立，则 $A_1A_2\cdots A_n \rightarrow C_1C_2\cdots C_K$ 在 R 中也成立，这个规则就称为传递规则。

证明过程在前面已有叙述，读者也可自行证明。

例如，若在前面提到的关系 Student 中有两个函数依赖：SNO→SDEPT，SDEPT→MNAME，则根据传递规则，可以得到一个新的函数依赖：SNO→MNAME。

3.3 关系模式的键码

3.3.1 键码的定义

定义 3.4 已知 $R<U,F>$ 是属性集 U 上的关系模式，F 是属性集 U 上的一组函数依赖。设 K 为 $R<U,F>$ 中的属性或属性组合，若 $K \rightarrow U-K$ 且 K 的任何真子集都不能决定 U，则 K 为 R 的键码。

包含键码的属性集称为超键码，它是“键码的超集”的简称。

若键码多于一个，则选定其中之一作为主键码。包含在任何一个键码中的属性，称为主属性，不包含在任何键码中的属性称为非主属性。最简单的情况：单个属性是键码。最极端的情况：整个属性组是键码，称为全码。

【例 3.2】 考虑关系模式：人（身份证号，姓名，性别，住址，出生年月），且有函数依赖集：F={身份证号→（姓名，性别，住址，出生年月），（姓名，住址）→（身份证号，性别，出生年月）}。由定义 3.4 可知该模式有两个键码，一个是身份证号，另一个是（姓名，住址）。对于公安部门，它选身份证号作为主键码，用身份证号作为人的唯一标识；对于邮电部门，它选（姓名，住址）作为主键码，用姓名和住址作为投寄信件的唯一标识。在该模式中，身份证号、姓名、住址这 3 个属性为主属性，性别和出生年月为非主属性。下面再举一个全码的例子。

【例 3.3】 考虑关系模式 R（演奏者，作品，听众）。假设一个演奏者可以演奏多个作品，某一作品可被多个演奏者演奏，听众也可以欣赏不同演奏者的不同作品，这个关系模式的码为（演奏者，作品，听众），即全码。

定义 3.5 关系模式 R 中属性或属性组 X 并非 R 的键码，但 X 是另一个关系模式的键码，则称 X 是 R 的外键码。

例如在关系模式 SC(SNO,CNO,GRADE)中，SNO 不是键码，但 SNO 是关系模式 S(SNO,SNAME,DEPT,AGE)的键码，则 SNO 对关系模式 SC 来说是外键码。

键码与外键码提供了一个表示关系间联系的手段，如在关系模式 S 和 SC 中，通过 SNO 可以自然连接两个模式，从而使 SNAME、DEPT 和 AGE 属性与 CNO 和 GRADE 属性之间建立一定的关联关系，以方便用户访问。

3.3.2 闭包的计算

假设 $A=\{A_1,A_2,\cdots,A_n\}$ 是属性集，F 是函数依赖集。属性集 A 在函数依赖集 F 下的闭包是这样的属性集 X，它使得满足函数依赖集 F 中的所有函数依赖的每个关系也都满足 $A \rightarrow X$。也就是说，$A_1A_2\cdots A_n \rightarrow X$ 是蕴含于 F 中的函数依赖。我们用 $\{A_1,A_2,\cdots,A_n\}^+$ 来表示属性集 $A_1A_2\cdots A_n$ 的闭包。为了简化闭包的计算，允许出现平凡函数依赖，所以 $A_1,A_2,\cdots,A_n$ 总在 $\{A_1,A_2,\cdots,A_n\}^+$ 中。

若要求解属性集$\{A_1,A_2,\ldots,A_n\}$在某函数依赖集下的闭包，则具体的计算过程如下：

① 属性集X最终将成为闭包。首先，将X初始化为$\{A_1,A_2,\ldots,A_n\}$。

② 然后，反复检查某个函数依赖$B_1B_2\cdots B_m\rightarrow C$，使得所有的$B_1B_2\cdots B_m$都在属性集$X$中，但$C$不在其中，于是将$C$加到属性集$X$中。

③ 根据需要多次重复步骤②，直到没有属性能加到X中。由于X是只增的，而任何关系的属性数目必然是有限的，因此最终再也没有属性可以加到X中。

④ 最后得到的不能再增加的属性集X就是$\{A_1,A_2,\ldots,A_n\}^+$的正确值。

【例 3.4】让我们来考虑一个关系$R(A,B,C,D,E,F)$，其函数依赖集为$F=\{AB\rightarrow C，BC\rightarrow AD，D\rightarrow E,CF\rightarrow B\}$，试计算$\{A,B\}$的闭包$\{A,B\}^+$。

从$X=\{A,B\}$出发。首先，函数依赖$AB\rightarrow C$左边的所有属性都在X中，而C不在X中，于是可以把该依赖右边的属性C加到X中，因此$X=\{A,B,C\}$。

然后，将$BC\rightarrow AD$分解为$BC\rightarrow A$和$BC\rightarrow D$。对于$BC\rightarrow A$，因它的左边包含在X中，但它的右边也包含在X中，故不符合条件；对于$BC\rightarrow D$，因它的左边包含在X中，但它的右边不包含在X中，故将属性D加到X中，即$X=\{A,B,C,D\}$。再对于$D\rightarrow E$，因它的左边包含在X中，但它的右边不包含在X中，故将属性E加到X中，即$X=\{A,B,C,D,E\}$。再对于$CF\rightarrow B$，因它的左边不包含在X中，故不符合条件。因此，$\{A,B\}^+=\{A,B,C,D,E\}$。

从上述闭包的计算过程可以进一步理解闭包的实际含义：对于给定的函数依赖集F，属性集A函数决定的属性集合就是属性集A在函数依赖集F下的闭包。

如果知道如何计算任意属性集的闭包，就能检验给定的任一函数依赖$A_1A_2\cdots A_n\rightarrow B$是否蕴含于函数依赖集$F$。首先利用函数依赖集$F$计算$\{A_1,A_2,\cdots,A_n\}^+$。如果$B$在$\{A_1,A_2,\cdots,A_n\}^+$中，则$A_1A_2\cdots A_n\rightarrow B$蕴含于$F$；反之，如果$B$不在$\{A_1,A_2,\cdots,A_n\}^+$中，则该依赖并不蕴含于$F$。

学会计算某属性集的闭包，还可以根据给定的函数依赖集推导蕴含于该依赖集的其他函数依赖。下面看一个具体的例子。

【例 3.5】已知关系模式$R(A,B,C,D)$，其函数依赖集$F=\{AB\rightarrow C,C\rightarrow D,D\rightarrow A\}$，求蕴含于给定函数依赖的所有非平凡函数依赖和键码。

首先，考虑各种属性组合的闭包；然后，依次分析各属性集的闭包，从中找出该属性集所具有的新的函数依赖；最后，根据各属性集的闭包求出键码等。

单属性：$A^+=A$，$B^+=B$，$C^+=ACD$，$D^+=AD$

新依赖：$C\rightarrow A$ （1）

双属性：$AD^+=AD$，$\underline{AB}^+=ABCD$，$AC^+=ACD$，$\underline{BC}^+=ABCD$，$\underline{BD}^+=ABCD$，$CD^+=ACD$

新依赖：$AB\rightarrow D$，$AC\rightarrow D$，$BC\rightarrow A$，$BC\rightarrow D$，$BD\rightarrow A$，$BD\rightarrow C$，$CD\rightarrow A$ （2）

三属性：$\underline{ABC}^+=ABCD$，$\underline{ABD}^+=ABCD$，$ACD^+=ACD$，$\underline{BCD}^+=ABCD$

新依赖：$ABC\rightarrow D$，$ABD\rightarrow C$，$BCD\rightarrow A$ （3）

四属性：$\underline{ABCD}^+=ABCD$

新依赖：无

从上面的分析可以得出，蕴含于给定函数依赖的非平凡函数依赖总共有 11 个。从这个例子还可以看出，只要计算出各种属性组合的闭包，关系模式的键码自然而然就找到了。因为键码函数决定所有其他属性，所以键码属性的闭包必然是属性全集。于是，反过来看，若某属性集的闭包为属性全集，则该属性集即为键码。当然，判断时，应从最小属性集开始，以区别键码和超键码。

在本例中，有 3 个键码 *AB*、*BC* 和 *BD*，分别用下画实线表示；有 4 个超键码 *ABC*，*ABD*，*BCD* 和 *ABCD*，分别用下画虚线表示。

3.4 关系的规范化

对于同一个应用问题，选用不同的关系模式集合作为关系数据库模式，其性能的优劣是大不相同的。为了区分关系数据库模式的优劣，人们常常把关系数据库模式分为各种不同等级的范式。范式可以理解成符合某种级别的关系模式的集合，人们常称某一关系模式 *R* 属于第几范式，就是表示该关系的某种级别。一般地，若 *R* 属于 BC 范式，则可写成 $R\in$BCNF。

对于各种范式之间的联系，有 5NF⊂4NF⊂BCNF⊂3NF⊂2NF⊂1NF 成立。一个低一级范式的关系模式，通过模式分解可以转换为若干个高一级范式的关系模式的集合，这种过程就称为规范化。本节主要讨论第一范式（1NF）、第二范式（2NF）、第三范式（3NF）和 BC 范式（BCNF），并且介绍如何将具有不合适性质的关系模式转换为更合适的形式。

3.4.1 第一范式（1NF）

定义 3.6 设 *R* 是一个关系模式，如果 *R* 中每个属性值域中的每个值都是不可分解的，则称 *R* 是属于第一范式的，记作 $R\in$1NF。

【例 3.6】考察如表 3.1 所示的学生课程记录表 1。

表 3.1 学生课程记录表 1

	学号	课程
t1→	3721	{程序设计，人工智能，数据结构}
t2→	3843	{编译原理，操作系统}

在上述关系模式中，由于课程属性的值是可分解的，故学生课程记录表 1$\notin$1NF。

存在问题：上述学生课程记录表 1 不属于 1NF，会产生以下问题。

① 如果 3721 学生想把选修课改为{编译原理，操作系统}，则系统在处理时将面临一种二义性：是修改元组 t1 中的课程属性值呢？还是把 t2 元组中的学号属性值扩充为{3721,3843}？

② 若要在学生课程记录表 1 中加入一个属性“成绩”，那么随之而来的约束条件（学号，课程）→成绩在这种非 1NF 中也难以表示。

解决办法：将课程属性的属性值拆开，变成一个属性值只有一个值的形式，如表 3.2 所示的学生课程记录表 2。

表 3.2 学生课程记录表 2

学号	课程
3721	程序设计
3721	人工智能
3721	数据结构
3843	编译原理
3843	操作系统

显然，学生课程记录表 2$\in$1NF。

3.4.2 第二范式（2NF）

定义 3.7 若 $R\in$1NF，且每个非主属性完全函数依赖于键码，则 $R\in$2NF。

【例3.7】考察学生、系、选课情况的关系模式：SA(SNO,SNAME,DEPT,MN,CNO,CNAME,GRADE)，其函数依赖 F_{SA}={SNO→SNAME,SNO→DEPT,DEPT→MN,CNO→CNAME,(SNO,CNO)→GRADE }。

通过分析可知：SA 的键码为(SNO,CNO)，因此 SNAME、DEPT、MN、CNAME 和 GRADE 均是非主属性。这些非主属性中只有 GRADE 是完全函数依赖于键码(SNO,CNO)，其他属性 SNAME、DEPT、MN 和 CNAME 只依赖于 SNO 或 CNO，故全是部分函数依赖于键码，由定义 3.7 可知，SA$\notin$2NF。

存在问题：一个关系模式 R 不属于 2NF，就会产生以下问题。

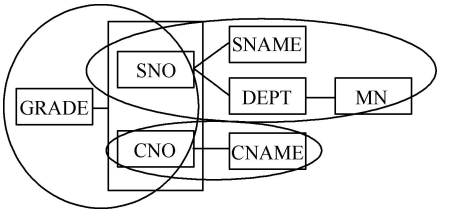
① 插入异常：假如要插入一个还未选课的学生，由于该学生无 CNO 信息，其相应的键码值一部分为空，所以该学生的固有信息无法插入。

② 删除异常：假定某个学生只选一门课，如 S4 选了一门课 C3，现在 C3 这门课他也不选了，那么 C3 这个数据项就要删除。因为 C3 是主属性，故删除了 C3 后整个元组就不存在了，也就是删除了 S4 的其他信息，从而造成删除异常。

③ 修改复杂：如某个学生从网络工程系转到计算机科学与技术系，这本来只需修改此学生元组中的系分量，但因为关系模式 SA 中还含有系负责人属性，学生转系将同时改变系负责人，因而还必须修改元组中的系负责人分量。另外，如果这个学生选修了 K 门课，则系和系负责人要重复存储 K 次。这不仅造成存储冗余度大，而且必须无遗漏地修改 K 个元组中全部系和系负责人的信息，从而造成了修改的复杂化。

解决办法：按“一事一地”的原则对关系模式 SA 进行投影分解，具体步骤如下。

① 根据关系模式 SA 的函数依赖集 F_{SA} ={ SNO→SNAME,SNO→DEPT,DEPT→MN,CNO→CNAME,(SNO,CNO)→GRADE }，画出函数依赖图。



② 按“一事一地”的原则，在函数依赖图中画 3 个圈，并将它拉开形成如下 3 个关系模式：SD(SNO,SNAME,DEPT,MN)，F_{SD}={ SNO→SNAME,SNO→DEPT,DEPT→MN}，SD$\in$2NF；

C(CNO,CNAME)，F_C={ CNO→CNAME }，C$\in$2NF；

SC(SNO,CNO,GRADE)，F_{SC}={ (SNO,CNO)→GRADE }，SC$\in$2NF。

按照定义 3.7，上述 3 个关系模式均属于第二范式。

3.4.3 第三范式（3NF）

定义 3.8 若 $R\in$2NF，且每个非主属性均不传递函数依赖于键码，则 $R\in$3NF。

由定义 3.7 可以证明，若 $R\in$3NF，则每个非主属性既不部分函数依赖于键码也不传递函数依赖于键码，即若 $R\in$3NF，则必然 $R\in$2NF。

【例 3.8】考察上一节的 SD(SNO,SNAME,DEPT,MN)、C(CNO,CNAME)和 SC(SNO,CNO,

GRADE)关系模式。

按照定义 3.8，关系模式 C 和 SC 中没有非主属性对键码的传递函数依赖，故 C∈3NF，SC∈3NF。但 SD 中存在非主属性系负责人 MN 对键码 SNO 的传递函数依赖，因此 SD∉3NF。

存在问题：一个关系模式 *R* 不属于 3NF，就会产生以下问题。

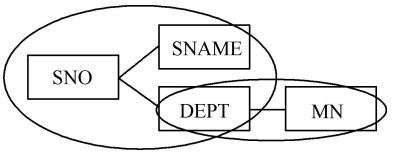
① 插入异常：当新成立一个系，该系还没有招收任何学生时，即无键码 SNO 的值，那么该系的有关信息就无法插入 SD 表中。

② 删除异常：若某个系的全部学生都已毕业，则在删除相应学生信息时，系和系负责人的信息也跟着被删除了。

③ 修改复杂：若一个系有 400 名学生，则系和系负责人的信息就要重复存储 399 遍，造成了数据冗余，并会引起修改困难。比如，某系更换了系负责人，则必须无遗漏地修改 400 个元组中的系负责人信息，从而造成修改的复杂化。

解决办法：按“一事一地”的原则对关系模式 SD 进行投影分解，具体步骤如下。

① 根据关系模式 SD 的函数依赖集 F_{SD}={ SNO→SNAME,SNO→DEPT,DEPT→MN }画出函数依赖图。



② 按“一事一地”的原则，在函数依赖图中画两个圈，并将它拉开形成如下两个关系模式：S(SNO,SNAME,DEPT)，F_S={ SNO→SNAME,SNO→DEPT }，S∈3NF；

D(DEPT,MN)，F_D={ DEPT→MN }，D∈3NF；

按照定义 3.8，上述两个关系模式均属于第三范式。

3.4.4 BC 范式（BCNF）

BCNF 是由 Boyce 与 Codd 提出的，它比上述 3NF 又进了一步，通常认为 BCNF 是修正的第三范式。

定义 3.9 关系模式 *R*<*U*,*F*>∈1NF，若对于任一函数依赖 *X*→*Y*（*Y*⊄*X*），*X*（决定因素）必含有键码，则 *R*∈BCNF。

也就是说，在关系模式 *R*<*U*,*F*>中，若每个决定因素都包含键码，则 *R*∈BCNF。

由于 3NF 不能很好地处理含有多个键码和键码是组合项的情况，因此人们定义了一个更强的范式 BCNF。一个满足 BCNF 的关系模式有以下特点：所有非主属性对每个键码都是完全函数依赖；所有的主属性对每个不包含它的键码也是完全函数依赖；没有任何属性完全函数依赖于非键码的任何一组属性。

可以证明，若 *R*∈BCNF，则 *R*∈3NF；但是若 *R*∈3NF，则 *R* 未必属于 BCNF。

【例 3.9】正例：考察关系模式 SJP（学生，课程，名次）。

该模式中的元组含义是：每个学生、每门课程有一定的名次；每门课程中每一名次只有一个学生。由语义可得到下面的函数依赖集：

$$F=\{（学生，课程）→名次，（课程，名次）→学生\}$$

所以（学生，课程）与（课程，名次）都可以作为键码。这个关系模式中显然没有非主属性对键码的传递函数依赖或部分函数依赖，所以 SJP∈3NF。另外，从函数依赖集 *F* 可以看出，

每个决定因素都包含键码，故 SJP∈BCNF。

【例 3.10】反例：考察关系模式 SJT（学生，课程，教师）。

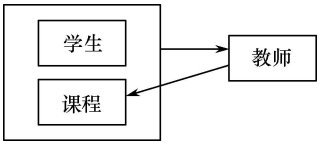
该模式中的元组含义是：某个学生学习某个教师开设的某门课程。现假定存在以下附加条件：

- 对于每门课、每个学生的讲课教师只有一位；
- 每位教师只讲授一门课；
- 每门课可由不同教师讲授。

因此，由语义可得到如下的函数依赖集：

$$F=\{（学生，课程）\rightarrow 教师，教师\rightarrow 课程\}$$

用函数依赖图可表示如下：



在 SJT 关系模式中，可求出两个键码（学生，课程）和（学生，教师），由于没有任何非主属性对键码的部分函数依赖或传递函数依赖，所以 SJT∈3NF；但 SJT∉BCNF，因为教师→课程，而教师不包含键码。

存在问题：不满足 BCNF 范式的关系模式同样存在着更新异常。

例如，在 SJT（学生，课程，教师）中，如果存在元组(S4,J3,T1)，当我们删除信息“学生 S4 学习 J3 课”时，将同时失去“T1 教师主讲 J3 课”这一信息。产生更新异常的原因是，属性教师是决定因子，却不是键码。

解决办法：设关系模式为 $R(A,B,C)$，其中 A，B，C 均为属性集，若存在违背 BCNF 的函数依赖 $A\rightarrow B$（违例），则可以 BCNF 的违例为基础把关系模式分解为：①$\{A,B\}$ ②$\{A,C\}$或$\{R-B\}$。按此规则可将 SJT 分解为两个关系模式：

TJ（教师，课程）和 ST（学生，教师）

按定义 3.9 进行判别，显然 TJ∈BCNF，ST∈BCNF。

3NF 和 BCNF 是在函数依赖条件下对模式分解所能达到的分离程度的测度。若一个关系模式经过分解达到了 BCNF，则在函数依赖范畴内已实现了彻底的分离，并已消除了插入异常、删除异常和修改复杂的问题。

3.5 模式分解的优劣

3.5.1 模式分解的等价性

当某些关系模式存在存储异常现象时，可以通过模式分解的方法使范式等级提高，从而得到一个性能较好的关系数据库的模式设计。但是，我们还须考虑另一个重要因素，就是所产生的分解与原来的关系模式是否等价？人们从不同的角度去观察问题，对“等价”的概念形成了以下 3 种不同的含义。

① 分解具有无损连接：当对关系模式 R 进行分解时，R 的元组将分别在相应属性集上进行投影而产生若干新的关系。若对若干新的关系进行自然连接得到的元组的集合与原关系完

全一致，则称分解具有无损连接。

② 分解保持函数依赖：当对关系模式 R 进行分解时，R 的函数依赖集也将按相应的模式进行分解。若分解后各关系模式的函数依赖集的并集与原函数依赖集保持一致，则称分解保持函数依赖。

③ 分解既要保持函数依赖，又要具有无损连接。

这 3 种含义是实行分解的 3 个不同的准则。按照不同的分解准则，关系模式所能达到的分离程度各不相同，各种范式就是对分离程度的测度。下面先给出分解的定义。

定义 3.10 设 $R(W)$是一个关系模式，$\rho=\{R_1(W_1),R_2(W_2),\cdots,R_k(W_k)\}$是一个关系模式的集合，如果 $W_1\cup W_2\cup\cdots\cup W_k=W$，则称$\rho$是 $R(W)$的一个分解。

从下面的例子可以知道：只要求 $R(W)$分解后的各关系模式所含属性的“并”等于 W，这个限定是很不够的，能否消除存储异常，不仅依赖于分解后各模式的范式，而且依赖于分解的方式。

【例 3.11】已知事实是：一个学生（SNO）只在一个系（DEPT）学习，一个系只有一名系主任（MN），关系模式 R(SNO,DEPT,MN)上的函数依赖集 F_R={SNO→DEPT,DEPT→MN}，R 的关系如下：

SNO	DEPT	MN
S1	D1	张五
S2	D1	张五
S3	D2	李四
S4	D3	王一

由于 R 中存在 MN 对 SNO 的传递函数依赖，故它会发生更新异常。例如，如果 S4 毕业，在该表中需删除该学生的信息，则 D3 系的系主任是王一的信息也就丢掉了；反过来，如果一个系 D5 尚无在校学生，那么这个系的系主任信息也无法存入。于是我们进行 4 种形式的分解。

（1）将 R(SNO,DEPT,MN)分解为 R1(SNO)、R2(DEPT)和 R3(MN)

分解后的 R1、R2 和 R3 列 3 个表如下：

R1

SNO
S1
S2
S3
S4

R2

DEPT
D1
D2
D3

R3

MN
张五
李四
王一

这 3 个表的关系模式显然全是 BCNF 范式，但是要根据这 3 个表回答“S1 在哪个系学习”或者“D1 系的系主任是谁”就不可能了。因此，这样的分解毫无意义。我们所希望的分解至少应不丢失原有信息，这就产生了无损连接的概念。

（2）将 R(SNO,DEPT,MN)分解为 R1(SNO,MN)和 R2(DEPT,MN)

可以证明这个分解是可恢复的，它保持了无损连接性，且分解后的 R1 和 R2 均是 BCNF 范式，但是对于前面提到的插入异常和删除异常仍然没有解决。原因就在于分解后，R1 上存在函数依赖(SNO,MN)→(SNO,MN)，R2 上存在函数依赖 DEPT→MN，但它们都丢失了原来在 R 中存在的函数依赖 SNO→DEPT。

（3）将 R(SNO,DEPT,MN)分解为 R1(SNO,DEPT)和 R2(SNO,MN)

可以证明这个分解是可恢复的，它保持了无损连接性，且分解后的 R1 和 R2 均是 BCNF

范式，但是仍然没有解决前面提到的插入异常和删除异常。原因就在于分解后，R1 上存在函数依赖 SNO→DEPT，R2 上存在函数依赖(SNO,MN)→(SNO,MN)，但它们也都丢失了原来在 R 中存在的函数依赖 DEPT→MN。

（4）将 R(SNO,DEPT,MN)分解为 R1(SNO,DEPT)和 R2(DEPT,MN)

可以证明该分解既具有无损连接性，又保持了函数依赖。此分解既解决了更新异常，又没有丢失原关系模式的信息，这是人们所希望的分解。

3.5.2 模式分解的规则和方法

由于关系模式分解的基础是键码和函数依赖，所以当对关系模式中属性之间的内在联系进行分析并确定了键码和函数依赖之后，模式分解应有一定的规则和方法。

1．模式分解的两个规则

（1）共享公共属性

要把分解后的模式连接起来，公共属性是基础。若分解时模式之间未保留公共属性，则只能通过笛卡儿积相连，导致元组数量膨胀，真实信息丢失，结果失去价值。分解后的两个模式 R_1 和 R_2 能实现无损连接的充分必要条件是：

$$(R_1 \cap R_2) \rightarrow (R_1 - R_2) \text{或} (R_1 \cap R_2) \rightarrow (R_2 - R_1)$$

上式表明：若分解后的两个关系模式的交集属性（公共属性）能决定两个关系模式的差集属性之一，则必能实现无损连接。

- 若原关系模式中存在对键码的部分函数依赖，则作为决定因素的键码的真子集就应作为公共属性，用来把分解后的新关系模式自然地连接在一起。
- 若原关系模式中存在对键码的传递函数依赖，则传递链的中间属性就应作为公共属性，用来把构成传递链的两个基本链的新关系模式自然地连接在一起。

（2）合并相关属性

把以函数依赖的形式联系在一起的相关属性放在一个模式中，从而使原有的函数依赖得以保持。这是分解后的模式实现保持依赖的充分条件。然而，对于存在部分函数依赖或传递函数依赖的相关属性，则不应放在一个模式中，因为这正是导致数据冗余和更新异常的根源，从而也正是模式分解所要解决的问题。

如果关系模式中属性之间的联系错综复杂、交织在一起，难解难分，难免会出现分解后函数依赖丢失的现象，这时也只能权衡主次、决定取舍。

2．模式分解的 3 种方法

（1）部分依赖归子集，完全依赖随键码

要使不属于第二范式的关系模式“升级”，就要消除非主属性对键码的部分函数依赖。解决的办法就是对原有模式进行分解：找出对键码部分依赖的非主属性所依赖的键码的真子集，然后把这个真子集与所有相应的非主属性组合成一个新的模式；对键码完全依赖的所有非主属性则与键码组合成另一个新模式。

【例 3.12】再次考察学生、系、选课情况的关系模式：SA(SNO,SNAME,DEPT, MN,CNO, CNAME,GRADE)，其函数依赖 F_{SA}={SNO→SNAME,SNO→DEPT,DEPT→MN,CNO→CNAME,(SNO,CNO)→GRADE }，键码为：(SNO,CNO)。

按照完全函数依赖和部分函数依赖的概念，可以看出 SNAME、DEPT 和 MN 均完全函数依赖于 SNO；CNAME 完全函数依赖于 CNO；GRADE 完全函数依赖于键码（SNO,CNO），

而 SNAME、DEPT、MN 和 CNAME 却部分函数依赖于键码（SNO,CNO）。

对键码而言，有两个部分函数依赖、一个完全函数依赖，故原来关系模式可分解为：

SD(SNO,SNAME,DEPT,MN)∈ BCNF，C(CNO,CNAME) ∈ BCNF，SC(SNO,CNO,GRADE) ∈ BCNF

本例中的两个部分函数依赖分别对应键码的两个真子集 SNO 和 CNO，真子集作为公共属性，可使 3 个模式实现自然连接。

（2）基本依赖为基础，中间属性作桥梁

要使不属于第三范式的关系模式“升级”，就要消除非主属性对键码的传递依赖。解决的办法就是对原有模式进行分解：以构成传递链的两个基本依赖为基础形成两个新模式，这样既切断了传递链，又保持了两个基本依赖，同时又有中间属性作为桥梁，可以实现无损的自然连接。

【例 3.13】再次考察学生、系情况的关系模式 SD(SNO,SNAME,DEPT,MN)，其函数依赖 F_{SD}={SNO→SNAME,SNO→DEPT,DEPT→MN}，键码为：SNO。

按照上述解决方案，将 DEPT 作为中间属性，可得到分解后的两个关系模式：

S(SNO,SNAME,DEPT)∈BCNF，D(DEPT,MN)∈BCNF

（3）找违例自成一体，舍其右全集归一；若发现仍有违例，再回首如法炮制

要使不属于 BCNF 的关系模式“升级”，就既要消除非主属性对键码的部分依赖和传递依赖，又要消除主属性对键码的部分依赖和传递依赖。解决的办法就是对原有模式进行分解：设关系模式为 $R(A,B,C)$，其中 A，B，C 均为属性集，若存在违背 BCNF 的函数依赖 $A→B$，则可以以 BCNF 的违例为基础把关系模式分解为：R1(A,B)和 R2(A,C)。

【例 3.14】设关系模式 STC(Sname,Tname,Cname,Grade)，其属性含义分别为：学生姓名、教师姓名、课程名和成绩，它的函数依赖集如下：

F={(Sname,Cname)→(Tname,Grade),(Sname,Tname)→(Cname,Grade),Tname→Cname}

键码为：(Sname,Cname)和(Sname,Tname)。

由于决定因素 Tname 没有包含键码，即 BCNF 的违例为 Tname→Cname，所以 STC∉BCNF。

按照上述解决方案，可得到分解后的两个关系模式：R1(Tname,Cname)∈BCNF，R2(Sname, Tname,Grade)∈BCNF。

对于函数依赖关系较复杂的关系模式，分解一次后可能仍有 BCNF 的违例，只要按上述方法继续分解，模式中的属性总是越分越少，最终少到只有两个属性时，必然属于 BCNF。

【例3.15】已知 SA(SNO,SNAME,DEPT,MN,CNO,CNAME,GRADE)，其函数依赖 F_{SA}= {SNO→SNAME,SNO→DEPT,DEPT→MN,CNO→CNAME,(SNO,CNO)→GRADE}，现将 SA 分解为 3 个关系模式 SD(SNO,SNAME,DEPT,MN)、SC(SNO,CNO,GRADE)和 C(CNO, CNAME)，试问这样的分解是否具有无损连接。

因 SD∩SC={ SNO },SD−SC＝{SNAME,DEPT,MN }，并且 SNO→(SNAME,DEPT,MN)，故分解后的两个关系模式 SD 和 SC 能实现无损连接；又因 C∩SC={CNO}，C−SC={CNAME}，并且 CNO→CNAME，故分解后的两个关系模式 C 和 SC 也能实现无损连接。综上所述，将 SA 分解为 SD、SC 和 C 后能实现无损连接。

需要说明的是，无损连接的判断标准是对一个关系模式分解为两个关系模式的判断法则；若一个关系模式分解为三个关系模式，则可以按两两相连的方法进行检验。

归纳总结：叙述对关系模式进行优化的步骤和方法。

在一个数据库应用系统中，假设有一个关系模式 $R(A_1,A_2,\cdots,A_n)$，要求对此关系模式进行优化以去除数据冗余和更新异常问题，对关系模式进行优化的步骤如下：

① 针对每个关系模式，分析属性间的函数依赖关系，写出函数依赖集，并求出键码。根据要求判断每个关系模式的范式等级，若某个关系模式已符合 3NF 或 BCNF 的要求，则优化结束；否则进到步骤②继续。

② 若某个关系模式没有达到 3NF，则可根据函数依赖集画出函数依赖图，按“一事一地”的原则将相关联的属性圈在一起；若某个关系模式没有达到 BCNF，则可根据“找违例自成一体，舍其右全集归一”的原则，将相关联的属性圈在一起。

③ 对圈在一起相关联的属性进行投影分解，分解后形成若干个关系模式，再转步骤①继续。

通常，我们必须根据实际需要多次应用分解规则，直到所有的关系都属于 3NF 或 BCNF 为止。

3.6 典型案例分析

3.6.1 典型案例 7——产品订货系统关系数据库模式的设计

1．案例描述

在一个产品订货系统数据库中，有一个关系模式如下：

订货（订单号，订购单位名，地址，产品型号，产品名，单价，数量）

要求：（1）给出你认为合理的函数依赖集和键码；

（2）根据投影分解求出一组满足 3NF 的关系模式；

（3）要求判断优化后每个关系模式的最高范式等级。

2．案例分析

此案例可以根据关系模式优化的步骤来实现，其关键点是函数依赖集要分析得合理和正确，否则会影响到后续的优化过程。

3．案例实现

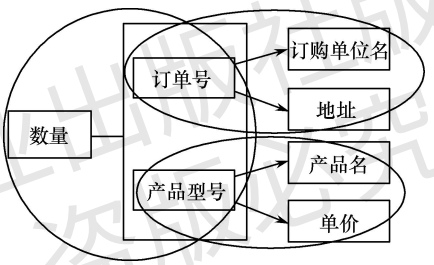
（1）根据产品订货系统数据库的情况，可以给出一个合理的函数依赖集如下：

F={订单号→订购单位名，订单号→地址，产品型号→产品名，产品型号→单价，

（订单号，产品型号）→数量}

键码为：（订单号，产品型号）。

（2）根据函数依赖，可以画出如下函数依赖图：



根据以上函数依赖图，可以进行投影分解如下：

R_1（订单号，订购单位名，地址），F_1={订单号→订购单位名，订单号→地址}

R_2（产品型号，产品名，单价），F_2={产品型号→产品名，产品型号→单价}

R_3（订单号，产品型号，数量），F_3={（订单号，产品型号）→数量}

根据 3NF 判定条件，可以确定 R_1、R_2 和 R_3 均属于 3NF。

（3）根据 BCNF 判定条件，可以确定 R_1、R_2 和 R_3 均可达到 BCNF。

3.6.2 典型案例 8——在线考试系统关系数据库模式的设计

1．案例描述

根据第 2 章 2.5.2 节典型案例 5，可知考生信息属性包括：考生学号、考生姓名、考生密码、考生性别、考生班级和注册日期；试卷信息属性包括：试卷编号、判断题数量、判断题每题分数、选择题数量、选择题每题分数、填空题数量、填空题每题分数和出卷日期；考试信息属性包括：试卷编号、考生学号、考生成绩、考试日期、是否补考、补考成绩和补考日期；判断题信息属性包括：判断题编号、判断题内容、标准答案和添加日期；选择题信息属性包括：选择题编号、选择题内容、标准答案和添加日期；填空题信息属性包括：填空题编号、填空题内容、标准答案和添加日期；管理员信息属性包括：管理员姓名和管理员密码。请设计在线考试系统的关系数据库模式，要求：

（1）按照概念数据模型（E-R 图）向关系模型转换的方法，得到一组关系模式；

（2）分析这一组关系模式，判断每个关系模式能达到的最高范式等级。

2．案例分析

在线考试系统主要包含考生信息和试卷信息两个实体集，此外还应包括考试信息、判断题信息、选择题信息、填空题信息和管理员信息等。其中考生信息和试卷信息之间是多对多联系，在线考试系统的 E-R 图如图 2.12 所示，此处不再重复。

3．案例实现

（1）按照第 4 章 4.4.2 节介绍的 E-R 图向关系模型转换的方法，可以得到如下 7 个关系模式：

管理员信息（管理员姓名，管理员密码）；

考生信息（考生学号，考生姓名，考生密码，考生性别，考生班级，注册日期）；

试卷信息（试卷编号，判断题数量，判断题每题分数，选择题数量，选择题每题分数，填空题数量，填空题每题分数，出卷日期）；

考试信息（试卷编号，考生学号，考生成绩，考试日期，是否补考，补考成绩，补考日期）；

判断题信息（判断题编号，判断题内容，正确答案，添加日期）；

选择题信息（选择题编号，选择题内容，正确答案，添加日期）；

填空题信息（填空题编号，填空题内容，正确答案，添加日期）。

（2）根据 BCNF 判定条件，可以确定管理员信息、考生信息、试卷信息、考试信息、判断题信息、选择题信息和填空题信息 7 个关系模式均已达到 BCNF。

3.6.3 典型案例 9——图书网上销售系统关系数据库模式的设计

1．案例描述

根据第 2 章 2.5.3 节典型案例 6，我们知道图书信息属性包括：图书编号、图书名称、图书规格、图书图片、图书价格、图书说明和是否特价图书；客户信息属性包括：订单编号、客户姓名、送货地址、客户电话、支付方式和订单总金额；订购信息属性包括：订单编号、图书编号、数量；购物袋信息属性包括：临时编号、已选购的图书编号、要选购的数量。请设计图书网上销售系统的关系数据库模式，要求：

（1）按照概念数据模型（E-R 图）向关系模型转换的方法，得到一组关系模式；

（2）分析这一组关系模式，判断每个关系模式能达到的最高范式等级。

2．案例分析

图书网上销售系统主要包含图书信息和客户信息两个实体集，图书信息和客户信息之间是多对多联系。此外，还应包括订购信息和购物袋信息，在订购信息中增加价格属性是为了计算方便起见引入的，它与图书信息中的图书价格含义相同，而购物袋信息是一个临时关系。图书网上销售系统的 E-R 图如图 2.13 所示，此处不再重复。

3．案例实现

（1）按照第 4 章 4.4.2 节介绍的 E-R 图向关系模型转换的方法，可以得到如下 4 个关系模式：

图书信息（图书编号，图书名称，图书规格，图书图片，图书价格，图书说明，是否特价图书）；

客户信息（订单编号，客户姓名，送货地址，客户电话，支付方式，订单总金额）；

订购信息（订单编号，图书编号，数量，价格）；

购物袋信息（临时编号，已选购的图书编号，要选购的数量）。

（2）根据 BCNF 判定条件，可以确定图书信息、客户信息、订购信息和购物袋信息 4 个关系模式均已达到 BCNF。

小　　结

本章主要介绍了函数依赖的概念、关系模式的键码、关系的规范化和典型案例分析等内容，要求熟练掌握函数依赖的概念（包括完全函数依赖、部分函数依赖和传递函数依赖），学会应用定义和闭包算法来求解关系模式的键码，理解 1NF、2NF、3NF 和 BCNF 的定义，掌握关系模式优化为 3NF 或 BCNF 的步骤和方法。

本章最后分析了 3 个典型案例。对于案例 7，要求学生根据题意分析得出产品订货系统合理的函数依赖和键码；通过画出函数依赖图、投影分解可得出初步优化的一组关系模式；根据 3NF 或 BCNF 判定条件，可以判断初步优化后每个关系模式的最高范式等级，最后根据题意确定是否要对关系模式进行优化。对于案例 8 和案例 9，要求学生掌握逻辑结构设计中概念数据模型（E-R 图）向关系模型转换的方法，并要求对于得到的一组关系模式根据 3NF、BCNF 判定条件确定其范式等级，最后根据题意确定是否要对关系模式进行优化。

如果一个关系数据库中的所有关系模式都满足了 BCNF，那么在函数依赖范畴内，它已实现了模式的彻底分解，达到了最高的规范化程度，消除了更新异常和数据冗余。应当强调的

是，规范化理论为数据库设计提供了理论的指南和工具，但仅仅是指南和工具，并不是规范化程度越高，模式就越好，我们必须结合实际应用环境和现实世界的具体情况合理地选择关系数据库的模式。

习　　题

3.1　理解并给出下列术语的定义：函数依赖、完全函数依赖、部分函数依赖、传递函数依赖、键码、主键码、外键码、主属性、非主属性、1NF、2NF、3NF 和 BCNF。

3.2　各举一个属于 1NF、2NF、3NF 和 BCNF 的例子，并加以说明。

3.3　设有关系模式 $R(A,B,C,D,E)$，$F_R=\{AB\rightarrow C,B\rightarrow D,D\rightarrow E,C\rightarrow B\}$，要求：

（1）通过闭包计算求出 R 的所有键码，并说明该模式是哪一类范式。

（2）若 R 分解为 $R_1(A,B,C)$和 $R_2(B,D,E)$，则分解是否保持函数依赖？

（3）指出 R_1 和 R_2 的范式等级，并给出证明。

（4）可否将 R_1 和 R_2 分解成若干个 BCNF 范式？请写出分解结果。

3.4　下面的结论哪些是正确的？哪些是错误的？简单说明理由。

（1）任何一个二目关系是属于 3NF 的；

（2）任何两个二目关系是属于 BCNF 的；

（3）若 K 是 R 的键码，则关系模式 R 是 3NF 的；

（4）只有一个键码的 3NF 关系模式，也必是 BCNF 的；

（5）由全部属性组成键码的关系模式是 3NF 的，也是 BCNF 的；

（6）若 $X\rightarrow Y$ 在 $W(W\subset U)$上成立，则在 U 上也一定成立。

3.5　已知一个关系模式：借阅（借书证号，姓名，所在系，书号，借书日期），要求：

（1）给出你认为合理的函数依赖集和键码；

（2）证明该关系模式是第几范式，要求说明理由；

（3）将该关系模式分解成一组满足 3NF 条件的关系模式。

3.6　假设某公司销售业务中使用的订单格式如下：

订单号：12345　订货日期：03/15/2020　客户名称：NBUT　客户电话：87654321

产品编号	品名	价格	数量	金额
A	电源	100.00	20	2000.00
B	电表	200.00	40	8000.00
C	卡尺	40.00	50	2000.00

总金额：12000.00

公司的业务规定：订单号是唯一的，每个订单对应一个订单号；每个订单号对应唯一的订货日期、客户名称和客户电话；一个订单可以订购多种产品，每种产品可以在多个订单中出现；一个订单有一个客户，且一个客户可以有多个订单；每个产品编号对应一种产品的品名和价格。试根据上述表格和业务规则对下面关系模式 R 进行优化：

R（订单号，订货日期，客户名称，客户电话，产品编号，品名，价格，数量，金额）

要求：（1）写出关系模式 R 的基本函数依赖集和键码；

（2）判断关系模式 R 最高可达到第几范式？为什么？

（3）将关系模式 R 分解成一组满足 BCNF 条件的关系模式。