教材：数据库实用教程（第四版）

《数据库原理》课程

清华大学出版社

2024年12月30日

第五章　规范化设计

本章讨论如何设计关系数据库的模式问题。

关系数据库
规范化设计理论
即“模式设计理论”

函数依赖————研究数据之间的联系

范　　　式 ————关系模式的标准

模式设计方法—— 规范化的方法

本章重要概念：

- 关系模式的一般形式和关系模式的冗余和异常问题。
- 数据依赖的定义、数据依赖与关键码的联系；平凡的FD。
- 关系模式的范式：1NF，2NF，3NF，BCNF。
- 逻辑蕴涵、闭包、数据依赖的公理系统和推理规则；属性集的闭包；FD集的等价；最小依赖集。
- 无损分解的定义、性质、测试；保持依赖的分解定义。
- 分解算法：分解成3NF、BCNF模式集的算法。
- MVD、4NF和5NF的定义。

§1 关系模式的设计问题

一、关系模式的一般形式:

$$\mathbf{R} < \mathbf{U}, \mathbf{D}, \mathbf{Dom}, \mathbf{F} >$$

$R < U, D, Dom, F >$

其中：

U：为组成关系R的全部属性的集合，即$U=\{A_1,A_2,\ldots\ldots A_n\}$。

D：为域的集合，即属性的取值范围的集合。

Dom：为U与D之间的映象，即Dom：U→D| Dom安全约束集。

F：为属性U上的 一组约束，即数据依赖集。

例如：学习关系SC中存在如下数据依赖：

（SNO,CNO）→ GRADE

学生关系S中存在如下数据依赖：

SNO →SNAME

（SNO,SNAME） →AGE

关系模式一般形式可简化为:

R < U, F >　　　　泛关系模式

满足上述制约条件F的关系用符号 **r** 表示

r 表示的关系称为 :　　　泛关系

例如： 学习关系模式 **SC < U, F >**　　其 中:

U= { SNO, CNO, GRADE }　　F={ (SNO, CNO) → GRADE }

设计关系数据库的核心问题是关系模式的设计。

对于一个数据库设计，可以有多个关系模式的选择。

按照什么原则来选择关系模式?

标准是什么?

如何实现?

二. 关系模式的存储异常问题

在数据管理中,数据冗余一直是影响系统性能的大问题。

数据冗余是指同一个数据在系统中多次重复出现。

例： SPJ_A（ SNO，PNO，JNO，JNAME，JCITY，PRICE，QTY ）

属性分别表示:供应商号、零件号、工程项目号、工程项目名称、

工程项目所在城市、零件单价、供应数量。

SPJ_A的一个实例：

关系模式 SPJ_A 的一个实例：SPJ_A1关系

SNO	PNO	JNO	JNAME	JCITY	PRICE	QTY
S1	P1	J1	东方明珠	上海	22.60	80
S1	P1	J4	明珠线	上海	22.60	60
S1	P3	J1	东方明珠	上海	22.80	100
S1	P3	J4	明珠线	上海	22.80	60
S1	P3	J6	南浦大桥	上海	22.80	6
S3	P3	J5	炼钢工地	天津	22.10	100
S3	P4	J1	东方明珠	上海	11.90	30
S3	P4	J4	明珠线	上海	11.90	60
S3	P4	J6	南浦大桥	上海	11.90	6
S4	P2	J4	明珠线	上海	33.80	60
S4	P2	J6	南浦大桥	上海	33.80	8
S5	P5	J1	东方明珠	上海	22.80	20
S5	P5	J4	明珠线	上海	22.80	60
S5	P5	J6	南浦大桥	上海	22.80	8
S8	P3	J1	东方明珠	上海	13.00	20

存在的问题：

① 数据冗余

② 修改异常

③ 插入异常

④ 删除异常

§2　函数依赖

关系可定义为笛卡儿积的一个子集，并不是每个子集都是有意义的。

对关系的值要作各种限制，这种限制称为数据完整性约束条件。

数据完整性约束主要有两种：

1. 依赖于值域的限制------------由**DBMS**完整性子系统实现；

2 . 只依赖于属性间取值的相等与否的 限制 --------------仅取决于

两个元组的某些分量是否相等--------统称为数据依赖

数据依赖描述的是属性与属性之间的对应关系。

一、 函数依赖定义

定义：设有关系模式R（$A_1, A_2, \ldots\ldots A_n$），X和Y是属性集（$A_1, A_2, \ldots\ldots A_n$）的子集，如果对于R的任何一个关系r中的任两个元组u和v，对应于X的属性分量的值相等，而对应于Y的属性分量的值也相等，即：只要有u[X]=v[X]，就有u[Y]=v[Y]，称

"X函数确定Y"或称"Y函数依赖于X", 其符号表示为：$X \rightarrow Y$

更明确地说： 对于r中属性或属性组X的每一个值，r中Y只有一个值与之对应。

FD的确切语义：表示了关系模式集X值与Y值的多对一联系。

(SNO,PNO,JNO)→(JNAME,JCITY) ……….

JNO →(JNAME,JCITY)

关系模式 SPJ_A 的一个实例：SPJ_A1关系

SNO	PNO	JNO	JNAME	JCITY	PRICE	QTY
S1	P1	J1	东方明珠	上海	22.60	80
S1	P1	J4	明珠线	上海	22.60	60
S1	P3	J1	东方明珠	上海	22.80	100
S1	P3	J4	明珠线	上海	22.80	60
S1	P3	J6	南浦大桥	上海	22.80	6
S3	P3	J5	炼钢工地	天津	22.10	100
S3	P4	J1	东方明珠	上海	11.90	30
S3	P4	J4	明珠线	上海	11.90	60
S3	P4	J6	南浦大桥	上海	11.90	6
S4	P2	J4	明珠线	上海	33.80	60
S4	P2	J6	南浦大桥	上海	33.80	8
S5	P5	J1	东方明珠	上海	22.80	20
S5	P5	J4	明珠线	上海	22.80	60
S5	P5	J6	南浦大桥	上海	22.80	8
S8	P3	J1	东方明珠	上海	13.00	20

存在的问题：

① 数据冗余

② 修改异常

③ 插入异常

④ 删除异常

二、 完全函数依赖

定义： 在关系模式R（$A_1,A_2,\ldots\ldots A_n$）中，已知 $X \to Y$,对于X中任何真子集X′(X′⊂X)都不能函数确定Y，称 $X \to Y$是完全函数依赖,用符号 $X \xrightarrow{f} Y$表示。完全依赖也称为“左部不可约依赖”。

三、部分函数依赖

定义：在关系模式R（$A_1,A_2,\ldots\ldots A_n$）中，已知 $X \to Y$,如果存在一个真子集X′(X′⊂X)，满足$X' \to Y$，则称Y对X是部分函数依赖,用符号 $X \xrightarrow{p} Y$ 表示。

四、传递函数依赖

定义：在关系模式R（$A_1,A_2,\ldots\ldots A_n$）中，已知 $X \to Y$，$Y \to Z$ ($Y \not\subset X$， $Z \not\subset Y$不存在$Y \to X$)，则称 $X \xrightarrow{t} Z$ 是传递函数依赖。

五、函数依赖和关键码的联系

定义：设有关系模式$R(A_1,A_2,\ldots\ldots A_n)$,K是属性集$(A_1,A_2,\ldots\ldots A_n)$的一个子集。如果$K \xrightarrow{f} U$（U完全函数依赖于K）在R上成立，那么称K是R的一个<u>候选键</u>。

如果$X \xrightarrow{p} U$（$K \subset X$）在R上成立, 那么称X是R的一个<u>超键</u>。

定义: 如果A是关系模式R的候选键K中属性，那么称A是R的<u>主属性</u>；否则称A是R的非主属性。

§3 关系模式的范式

用什么标准来衡量关系模式设计的好与坏？

---------关系模式的范式(**Normal Forms**,简记为**NF**)。

一、 第一范式（1NF）

定义： 如果关系模式R的每个关系r的属性值都是不可分的原子值, 那么称R是第一范式(first normal form)的模式。

满足1NF的关系称为规范化的关系，否则称为非规范化的关系。

关系数据库研究的关系都是规范化的关系。

既使关系模式是1NF，但很可能具有不受欢迎的冗余和异常现象。

因此需把关系模式作进一步的规范化。

二、 第二范式（2NF）

定义： **如果关系模式 R∈1NF,并且R中每个非主属性都完全函数依赖于R的候选键，称R是第二范式（2NF）的模式。**

如果数据库模式中每个关系模式都是2NF，则称数据库模式为2NF的数据库模式。…

三、第三范式（3NF）

定义：如果关系模式 **R∈1NF, 并且R中**每个非主属性都不传递依赖于**R**的候选键**,** 那么称**R**是第三范式（**3NF**）的模式。

如果数据库模式中每个关系模式都是**3NF**，则称其为**3NF**的数据库模式。

在**3NF**模式中，并未排除主属性对候选键的传递依赖，仍有可能有冗余和异常现象。

3NF分解

四、BCNF模式

定义： **如果关系模式 R∈1NF，并且R中每个属性都不传递依赖于R的候选键,那么称R是BCNF的模式。**

如果数据库模式中每个关系模式都是BCNF，称其为BCNF的数据库模式。

由BCNF的定义得出如下结论：

1. 非主属性对码完全函数依赖；

2. 主属性对不包含它的码也是完全函数依赖；

3. 没有属性完全依赖非码的任何属性组。

例如：关系模式 R(C, S, Z) 其中

C：城市名称 S：街道名称 Z： 邮政编码

假定：F={ CS → Z, Z → C}

CS → Z: 表示地址(城市和街道)决定邮政编码;

Z → C:表示邮政编码决定城市名称。

R的候选码为： CS 和 SZ。

R(C, S, Z)的候选码为：CS　和　SZ。

由于F中存在　Z →C:

主属性C对不包含它的码SZ不是完全函数依赖，

所以R(C, S, Z)不是BCNF模式，但R(C, S, Z)是3NF，

因为没有任何非主属性对码传递函数依赖或部分函数依赖。

如果把R(C, S, Z)分解为：R1（S，Z）

R2（Z，C）

能解决冗余问题，而且R1，R2都是BCNF模式集。

但丢失了CS→Z，数据语义将会引起新的矛盾。

§4 数据依赖的公理系统

一、函数依赖 FD 的逻辑蕴涵

定义一：设 F 是关系模式 R(A1,A2,……An)上成立的函数依赖集，X和Y是属性集（A1,A2,……An）的子集，X→Y是一个其他的函数依赖。如果对于R的每个满足F的关系r也满足X→Y（或从F推导出X→Y也在R（U）上成立）， 那么称

F 逻辑蕴涵 X→Y，记为： $F \models X \rightarrow Y$ 。

定义二：设F是关系模式R（$A_1,A_2,\ldots\ldots A_n$）上成立的函数依赖集，X和Y是属性集（$A_1,A_2,\ldots\ldots A_n$）的子集，

F的所有逻辑蕴涵组成的集合称为函数依赖集F的闭包，

记为F^+。 $$F^+ = \{ X\rightarrow Y \mid F\vDash X\rightarrow Y \}$$

即：从给定的函数依赖集合F推出的所有函数依赖组成的集合，称为F的闭包。

二、 FD推理规则（Armstrong公理）

设U是关系模式 R 的属性集， F是R上成立的一组函数依赖集

X、Y分别是U上的子集，存在如下规则：

1、FD公理（函数依赖公理）：

① 自反性： 若$Y \subseteq X \subseteq U$，则 $X \rightarrow Y$ 在R上成立。

② 增广性： 若 $X \rightarrow Y$ 在R上成立，且 $Z \subseteq U$，则

$XZ \rightarrow YZ$ 在R上成立。

③ 传递性： 若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 在R上成立，则

$X \rightarrow Z$ 在R上成立。

2、FD推理规则

① 并规则：　若 X→Y，X→Z，则 X→YZ。

② 伪传递规则：若 X→Y，YW→Z，则 XW→Z。

③ 分解规则：　若 X→Y，Z⊆Y，则 X→Z。

从并规则和分解规则可得到下面的定理。

定理：如果$A_1 \ldots A_n$是关系模式R的属性集，**那么 $X \to A_1 \ldots A_n$ 成立的充分必要条件是$X \to A_i$（i=1，…，n）成立。**

④ 复合规则：　若X→Y，W→Z，则XW→YZ。

⑤ 通用一致性定理：若X→Y，W→Z，则X∪（W－Y）→YZ。

例: 已知关系模式R(ABC), F=｛ $A \to B$， $B \to C$ ｝,求F^+。

答：根据函数依赖公理系统，可推出F的F^+有43个FD。其中：

①据自反性可得出27个FD：

$\emptyset \to \emptyset$

$A \to \emptyset$	$B \to \emptyset$	$C \to \emptyset$	$AB \to \emptyset$	$BC \to \emptyset$	$CA \to \emptyset$	$ABC \to \emptyset$
$A \to A$	$B \to B$	$C \to C$	$AB \to A$	$BC \to B$	$CA \to A$	$ABC \to A$
			$AB \to B$	$BC \to C$	$CA \to C$	$ABC \to B$
			$AB \to AB$	$BC \to BC$	$CA \to CA$	$ABC \to C$
						$ABC \to AB$
						$ABC \to BC$
						$ABC \to CA$
						$ABC \to ABC$

②据增广性可得出7个FD：

A→B　B→ C　AB→ C　(∵ A→B，AB→B, B→ C)

A→AB　B→ BC　AB→ABC　CA→BC (∵ CA→A, A→B,B→BC)

③据传递性可得出9个FD：

A→ C (∵ A→B，B→ C)　CA→B (∵ CA→A, A→B)

A→ CA

A→ BC (∵ A→CA, CA→BC)

AB→ BC (∵ AB→A,A→B C）

CA→AB (∵ CA→A,A→AB)

CA→AB (∵ CA→A,A→AB)

AB→CA (∵ AB→A,A→CA)

A→ABC (∵ A→CA,CA→ABC)

定义三： 对于函数依赖 $X \to Y$，

如果$Y \subseteq X$，那么称$X \to Y$是一个“平凡的FD”；

否则称为“非平凡的FD”。

平凡的FD并没有实际意义，根据自反性就可推出。我们感兴趣的是非平凡的FD。只有**非平凡的FD与完整性约束条件相关。**

在实际使用中，经常要判断能否从已知的函数依赖集F推导出函数依赖$X \to Y$，那么可先求出F的闭包F^+，然后再看$X \to Y$是否在F^+中。但是从F求F^+是一个复杂且困难的问题（NP完全问题，指数级问题）。下面引入属性集闭包概念，将**使判断问题化为多项式问题。**

三、属性集的闭包

定义：设F是属性集U上的FD集,X是U的子集，那么（相对于F）属性集X的闭包用X_F^+表示，它是一个从F集使用FD推理规则推出的所有满足X→A的属性A的集合：

$$X_F^+ = \{ A \mid A \in U,\ X \to A \in F^+ \}$$

从属性集闭包的定义可得出下面的定理。

定理：X→Y能用FD推理规则推出的充分必要条件是：

$$Y \subseteq X_F^+$$

四、属性集X的闭包的计算方法

输入：一个有限的属性集合U；

U上满足的函数依赖集合F；

U的一个子集X。

输出： X 关于 F 的闭包 X_F^+。

方法：根据下列规则计算属性集序列X（0），X（1），….。

①、置初值X（0）:= X，i=0；

②、$X(i+1):=X(i)\cup\{A|Y\rightarrow Z\in F\wedge A\subseteq Z\wedge Y\subseteq X(i)\}$

③、判断X（i）=X（i+1）否？

若不等，置X（i）=X（i+1）, i:= i +1 转②

若相等，计算终止，此时X（i+1）就是所要求的属性集X

关于F的闭包 X_F^+。

四、属性集X的闭包的计算方法

输入：一个有限的属性集合U；

U上满足的函数依赖集合F；

U的一个子集X。

输出： X 关于 F 的闭包 X_F^+ 。

方法：根据下列规则计算属性集序列X（0），X（1），…。

①、置初值X（0）:= X，i=0；

②、$X(i+1):=X(i)\cup\{A \mid Y\rightarrow Z\in F \wedge A\subseteq Z \wedge Y\subseteq X(i)\}$

③、判断X（i）=X（i+1）否？

若不等，置X（i）=X（i+1），i:= i +1 转②

若相等，计算终止，此时 X（i+1）就是所要求的属性集X关于F的闭包 X_F^+。

例：设U＝{A,B,C,D,E,,G}，F={ AB→C， C→A， BC→D, ACD→B, D→EG, BE→C, CG→BD, CE→AG } 求$(BD)_F^+$

解：设X=BD

① 令： X(0) =BD

② 计算 X(1) :=BD∪EG=BDEG ; (D→EG)

③ 计算 X(2):= BDEG ∪C=BCDEG ; (BE→C)

④ 计算 X(2):= BCDEG ∪A=ABCDEG;

(C→A, BC→D, CG→BD, CE→AG)

X(3)已包括所有属性，算法终止。$(BD)_F^+$ =ABCDEG

五、函数依赖集的等价和最小依赖集

定义一：关系模式 R(U)上的两个函数依赖集F和G，

如果 $F^+ = G^+$，称F和G是等价的；

如果 F和G等价，称F覆盖G，或G覆盖F。

定理： $F^+ = G^+$ 的充分必要条件是：

$$F \subseteq G^+ \quad 且 \quad G \subseteq F^+$$

定义二：如果函数依赖集 F 满足下列三个条件，

称 F 为最小依赖集：

① F中每一个函数依赖的右边都是单属性;---------右端无冗余属性

② 对于F中任一函数依赖$X \rightarrow A$，其$F-\{X \rightarrow A\}$与F不等价；

---------- F中没有冗余的函数依赖

③ 对于F中任一函数依赖$X \rightarrow A$，其$F-\{X \rightarrow A\} \cup \{Z \rightarrow A\}$与F不等价，其中$Z \subset X$。 -------- 左端无冗余属性。

定理：任何一个函数依赖集F至少存在一个最小函数依赖集F_{min}。

例： 分析下列函数依赖集F1，F2， F3是否为最小函数依赖集。

$$F1=\left\{\begin{array}{l} AD\rightarrow BC \\ BE\rightarrow C \\ C\rightarrow H \end{array}\right. \qquad F2=\left\{\begin{array}{l} A\rightarrow D \\ B\rightarrow A \\ A\rightarrow C \\ B\rightarrow D \\ D\rightarrow C \end{array}\right. \qquad F3=\left\{\begin{array}{l} A\rightarrow C \\ AD\rightarrow B \\ C\rightarrow D \\ D\rightarrow A \end{array}\right.$$

解析：F1不是最小集，因为AD→BC 不满足条件①。

F2不是最小集, 因为F2－{A→C}等价于F2，不满足条件②。

F3不是最小集，因为F3－{AD→B }∪{ D→B}等价于F3，

不满足条件③ 。

(F3中： D→A， AD→B 由伪传递得D→B)

函数依赖集最小化处理的一般步骤：

1、用分解规则消去函数依赖右端的冗余属性。

(使F中每一个函数依赖的右边都是单属性).

2、应用传递性消去函数依赖集F中冗余的函数依赖。

(对于F中每一个函数依赖X→A,分别检验F与F－{X→A}不等价)

3、应用伪传递规则消去函数依赖左端的冗余属性。

（对于F中任一函数依赖X→A，其F－{ X→A}∪{ Z→A}与F

不等价，Z⊂X ）。

例: 设函数依赖集$F=\{A\rightarrow C, C\rightarrow A, A\rightarrow B, B\rightarrow A, B\rightarrow C\}$，试求$F_{min}$。

解：第一步：F已满足最小函数依赖集的第①个条件中。

第二步：对于F中每一个函数依赖$X\rightarrow A$,分别检验F与

$F-\{X\rightarrow A\}$是否等价;

① F中的$A\rightarrow C$（∵$A\rightarrow B$，$B\rightarrow C$，据传递性）是多余的，所以

$A\rightarrow C$可删去。得$F=\{A\rightarrow B, B\rightarrow A, B\rightarrow C, C\rightarrow A\}$

②同样F中$B\rightarrow A$也是多余的（∵$B\rightarrow C, C\rightarrow A$）,$B\rightarrow A$可删去。

得$F=\{A\rightarrow B, B\rightarrow C, C\rightarrow A\}$

第三步：消去函数依赖左端的多余属性：在本例中，F中每一个函数依赖左端的都是单属性，不可再分。

$\therefore \quad F_{min}=\{A\rightarrow B, B\rightarrow C, C\rightarrow A\}$

例：已知F={ A→D，B→D，BD→CA，CD→B }，求 F_{min} 。

解：第一步：应用分解规则得：

F1={ A→D，B→D，BD→C，BD→A ，CD→B }

第二步：消去函数依赖左端的冗余属性(应用伪传递)

由B→D, BD→C, 可推出B→C, 所以BD→C的左端的D是多余的；

由B→D，BD→A, 可推出B→A, 所以BD→A的左端的D是多余的；

∴ F2={ A→D，B→D，B→C，B→A ，CD→B }

第三步：去除F 中冗余的FD (应用传递性)；

由 B→A，A→D，，可推出 B→D，所以 B→D 是多余的。

∴ F_{min}={ A→D，B→C，B→A ，CD→B }

§5　关系模式的分解

关系模式存在多余的函数依赖或者说模式不够规范，就有可能出现存储异常现象。

解决的办法是：　对关系模式进行分解

即：　将一个模式分解为多个关系模式。

但在分解后又会出现**新的问题**，原模式所满足的特征（如属性之间的约束）在新模式中是否被保持；

即 模式的等价性问题。

等价性问题包括：数据等价和依赖等价两个方面：

数据等价是指两个数据库实例应表示同样的信息内容，用“无损联接”来衡量。如果是无损联接分解，那么反复投影和自然联接都不会丢失信息。

依赖等价是指两个数据库模式应有相同的依赖集闭包。在依赖集闭包相同的情况下，数据的语义是不会出差错的。一个违反数据等价或依赖等价的分解很难说是一个好的模式设计。

一、模式分解问题

定义：设有关系模式R（U），R_1、…、R_k 都是R的子集，

$U=R_1\cup\ldots\cup R_k$，关系模式R_1、…、R_k的集合用ρ表示，

ρ=｛R_1，…，R_k｝，用ρ代替R的过程称为关系模式R的分解。

ρ称为R的一个分解,ρ也称为数据库模式。

泛关系模式		数据库模式
R	⟹	$\rho=\{R_1, \ldots, R_k\}$
r	⟹	$\sigma=\langle r_1, \ldots, r_k\rangle$
泛关系 (元组的集合）		数据库实例（数据库）

把R分解成ρ的目的：

是为了消除数据冗余和操作异常现象。

问题是： **σ和 r 是否表示同一个数据库。**

如果两者表示不同的内容，那么这个分解就没有什么意义了。

可以从两个角度来考虑分解：

① **σ和 r 是否等价：** 即是否表示同样的数据？

用 **“无损分解”** 特性表示。

② 在模式R上有一个函数依赖集F，在ρ的每一个模式R_i上

也有一个FD集F_i，那么

{ F_1，…，F_n } 与F是否等价？

用 **“ 保持依赖”** 特性表示。

二、 无损联接分解

定义：如果R是一个关系模式，数据库模式 $\rho=\{R_1, \ldots, R_k\}$ 是R的一个分解，F是R上成立的一个函数依赖集。如果在R中满足F的每一个关系r，都有下式成立：

$$r = \pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie \ldots \bowtie \pi_{Rk}(r)$$

称分解 ρ 相对于F是“无损联接分解”，否则称为“损失分解”。

其中：符号 $\pi_{Ri}(r)$ 表示：关系r在模式R_i属性上的投影。

r的投影联接表达式$\pi_{R1}(r) \ldots \bowtie \pi_{Rk}(r)$用符号$m_\rho(r)$表示：

$$m_\rho(r) = \mathop{\bowtie}_{i=1}^{K} \pi_{Ri}(r)$$

如果R是一个关系模式，$\rho=\{R_1,R_2,\ldots,R_k\}$是关系模式R上的一个分解，r是R上的任一关系，$r_i=\pi_{R_i}(r)$（$1\le i\le k$），则有下列性质：

① $r \subseteq m_\rho(r)$；

② 若$s = m_\rho(r)$，则$\pi_{R_i}(s)=r_i$；

③ $m_\rho(m_\rho(r))= m_\rho(r)$； 幂等性。

三、无损联接的测试　　算法1：

输入： 关系模式R（$A_1,A_2,\ldots A_n$）,R上成立的函数依赖集F，

R的一个分解ρ={R_1，R_2，…，R_k}。

输出： 判断ρ相对于F是否具有无损联接特性。

具体步骤:

①构造一张 **k行n列** 的表格,每列对应一个属性$A_j(1\le j\le n)$，每行对应一个模式$R_i(1\le i\le k)$：

如果A_j在R_i中，那么在表格的第i行第 j列处填上符号a_j，否则填上符号 b_{ij}。

②反复检查F中的每一个函数依赖,并修改表格中的元素,方法是：

取F中的一个函数依赖X→Y，如果表格中有两行在X分量上相等

,在Y分量上不相等，那么修改Y，使这两行在Y分量上也相等。

原则：a .如果Y的分量中有一个是a_j，那么另一个也修改成a_j；

b .如果没有a_j，那么用其中一个b_{ij}替换另一个符号（尽

量把下标ij改成较小的数）。一直到表格不能修改为止

（这个过程称为Chase过程）。

③、若修改到最后,一张表格中有一行是全a，即$a_1a_2\ldots a_n$,

那么ρ相对于F是无损联接分解。

例：设有R（ABCDE），F={A→C，B→C，C→D，CE→A，DE→C}，

R 的一个分解 ρ={AD，AB，BE，CDE，AE}，

求 ρ 的分解无损性。

答：①根据已知条件构造一张 5 行 5 列的表格如下：

j / i	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{23}	b_{24}	b_{25}
BE	b_{31}	a2	b_{33}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	b_{53}	b_{54}	a_5

② 根据A→C，对表一进行处理，将b_{23}，b_{53}改为b_{13}，

再考虑B→C，将b33改为b13;

j \ i	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{23} b_{13}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{33}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	b_{53} b_{13}	b_{54}	a_5

② 根据A→C，对表一进行处理，将b_{23}，b_{53}改为b_{13}，

再考虑B→C，将b_{33}改为b_{13}；

j / i	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{23} b_{13}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{33} b_{13}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	b_{53} b_{13}	b_{54}	a_5

② 根据A→C，对表一进行处理，将b_{23}，b_{53}改为b_{13}，

再考虑B→C，将b_{33}改为b_{13};

然后考虑C→D，将b_{24}，b_{34}， b_{54}改为a_4。修改后的表格如下：

i \ j	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{23} b_{13}	b_{24} a_4	b_{25}
BE	b_{31}	a_2	b_{33} b_{13}	b_{34} a_4	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	b_{53} b_{13}	b_{54} a_4	a_5

在上表的基础上，考虑$DE \rightarrow C$，将b_{13}改为 a_3；

考虑$CE \rightarrow A$，将b_{31}，b_{41}改为 a_1。

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b13	a_4	b_{25}
BE	b_{31} a_1	a_2	b_{13} a_3	a_4	a_5
CDE	b_{41} a_1	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	b_{13} a_3	a_4	a_5

③ 由于 第三行为全a，所以 ρ 是无损联接。

2、无损联接的测试 算法2：

如果**R**的一个分解为ρ={ R1, R2}，**F**为**R**所满足的函数依赖集合，分解ρ相对于F是无损分解的充要条件是：

$$R1 \cap R2 \rightarrow R1 - R2$$

或 $$R1 \cap R2 \rightarrow R2 - R1$$

例：试分析下列分解是否具有无损联接性：

①设R（ABC），F={A→B}在R上成立，ρ1={AB，AC}。

答：ρ1相对于F具有无损联接性。

∵R1∩R2=AB∩AC=A；

R1- R2=AB-AC=B；

满足A→B；

∴模式ρ1={AB，AC}具有无损联接性。

52页

②设R(ABC)，F={A→B}在R上成立，ρ2={AB，BC}。

答：ρ 2相对于F3是有损分解：

∵ R1∩R2=AB∩BC=B； R1- R2=AB- BC =A；

不满足A→B；

∴模式ρ2={AB，BC}是有损分解。

四、保持函数依赖的分解

如果关系模式在分解后不能保持函数依赖，那么在数据库中就会出现异常现象。

所以，关系模式R到ρ={R1，R2，…，Rk}的分解，应使函数依赖集 F 被 F 在这些Ri上的投影所逻辑蕴涵。

这就是保持函数依赖问题。

定义一: 设F是属性集U上的函数依赖集, Z是U上的一个子集, F在Z上的一个投影用π_Z（F）表示:

$$\pi_Z(F)=\{X\to Y \mid X\to Y\in F^+ \land X, Y\subseteq Z\}$$

定义二: 设$\rho=\{R_1, R_2, \ldots, R_k\}$是关系模式R的一个分解，F是 R上的函数依赖集，如果

$$\bigcup_{i=1}^{k} \pi_{R_i}(F) \models F$$

称分解ρ保持函数集F。

从定义一可知 $F \models \bigcup_{i=1}^{k} \pi_{R_i}(F)$，从定义二可知 $\bigcup_{i=1}^{k} \pi_{R_i}(F) \models F$，

因此，在分解ρ保持函数依赖情况下有 $\left(\bigcup_{i=1}^{k} \pi_{R_i}(F)\right)^{+} = F^{+}$。

根据定义二，测试一个分解是否保持FD，比较可行的方法是

逐步验证F中每个FD是否被 $\bigcup_{i=1}^{k} \pi_{R_i}(F)$ 逻辑蕴涵。

如果F的投影不蕴涵F，而我们又用 $\rho=\{R_1, \ldots, R_k\}$ 表达R，很可能会找到一个数据库实例σ满足投影后的依赖，但不满足F。对σ的更新也有可能使 r 违反FD。下面的例子说明了这种情况。

例：试分析下列分解是否具有无损联接性和保持函数依赖性：

设R（ABC），$F=\{A\rightarrow C, B\rightarrow C\}$在 R上成立， $\rho=\{AB, AC\}$。

解：ρ相对于F是无损联接分解但不保持函数依赖性。

∵ $R1 \cap R2 = AB \cap AC = A$；　$R2-R1 = AC-AB = C$；

满足 $A\rightarrow C$；

∴模式 $\rho=\{AB, AC\}$具有无损联接性。

又 ∵已知 $F=\{A\rightarrow C, B\rightarrow C\}$

∵ $A^+=AC$　$B^+=BC$　$(AB)^+=ABC$

∴ $\pi_{AB}(F)=\{\phi\}$；

$\pi_{AC}(F)=\{A\rightarrow C\}$；

而 $\pi_{AB}(F) \cup \pi_{AC}(F)=\{A\rightarrow C\}$，$B\rightarrow C$在该分解中丢失了。

∴ 模式ρ不具有保持函数依赖性。

§6 模式分解的方法

一、分解成3NF模式集

算法1：结果为<u>3NF</u>并保持函数依赖性的分解。

输入：关系模式R的属性集合U和R上成立的最小函数依赖集F_{min}。

输出：R的一个分解$\rho=\{R_1, R_2, \ldots, R_k\}$，满足每一个$R_i$相对

π_{Ri}（F）是3NF，且ρ保持F。

方 法:

① 如果R中某些属性在F的所有依赖的左端或右端都不出现，那么这些属性可以从R中分解出去，单独构成一个模式。

② 如果F中存在一个依赖 $X \to A$，而且$\{X, A\}=U$，则

$\rho:=\{R\}$，停止。

③ 如果不为① ②，则进行如下分解:

对F中的 每一个依赖 $X \to A$，构成一个关系模式 XA。

如果 F 中有$X \to A_1$, $X \to A_2$ …, A_n, 则可以用模式$X A_1 A_2 \ldots A_n$代替XA_1，XA_2，…，XA_n。

④ 分解结束，输出 ρ。

例：关系模式 W（CTHSRG），

最小函数依赖集 F_{min}=（C →T，HR →C，HT →R，

CS →G，HS →R）

根据算法，生成的分解为：

ρ=（ CT，HRC，HTR，CSG，HSR ）

这样的分解，显然将原有的函数依赖都保持下来，而且每个分解模式均为3NF模式。

算法2：转换为3NF即具有无损连接性又保持函数依赖的分解。

输入: 关系模式R的属性集合U和R上成立的最小函数依赖集Fmin。

输出: 即保持函数依赖又具有无损联接地分解τ，而且τ中所有关系模式都是3NF。

方法：① 设$\rho=\{R_1, R_2, \ldots, R_n\}$是由算法1得到的一个3NF保持函数依赖的分解;

② 构造 $\tau=\rho\cup\{X\}$, 其中 X 是R的一个侯选键。

例：ρ=（ CT，HRC，HTR，CSG，HSR ）

∵ $HS^+ = CTHSRG$

$\tau=\rho\cup\{X\}$=(CT，HRC，HTR，CSG，HSR)

四、无损联接地分解成BCNF模式集

定理：设F是模式R的FD集，$\rho=\{R_1，R_2，…，R_k\}$是相对于F的一个无损联接分解，则有下面的结论成立：

① 对某个i，设 $F_i = \pi_{R_i}(F)$，且$\rho_1=\{S_1，S_2，…，S_m\}$是相对于 F_i 的 R_i一个无损联接分解。那么R分解成：

$\{R_1 …，R_{i-1}，S_1，S_2，…S_m，R_{i+1}，…，R_k\}$

是相对于F的一个无损联接分解。

② 设$\rho_2 = \{R_1…，R_k，R_{k+1}，…，R_n\}$是R的一个分解，那么$\rho_2$相 对于F也是一个无损联接分解，

算法3：无损联接地分解成BCNF模式集

输入： 关系模式R的属性集合U和R上成立的函数依赖集F。

输出： R的一个无损分解$\rho=\{R_1, R_2, \ldots, R_k\}$，满足每个 Ri 相对 π_{Ri}（F）是BCNF模式集。

方法：反复使用上述定理，逐步分解关系模式R，使每次分解具有无损联接特征，并且分解出来的模式满足BCNF。

方 法：反复使用上述定理，逐步分解关系模式R，使每次分解具有无损联接特征，并且分解出来的模式满足BCNF。

① 置初值： ρ={ R }；

② 如果ρ中所有模式都是BCNF，则转④；

③ 如果ρ中有一个关系模式S不是BCNF， 则 S中必能找到一个函数依赖X→A有X不是S的码,且A⊄X，设 S1=XA，S2=S—A， 用分解 {S1，S2} 代替S，转②；

④分解结束，输出ρ。

举 例1：试分析下列分解是否保持函数依赖性：

设 R（ABCD），$F=\{A\to B，B\to C，C\to D，D\to A\}$，$\rho=\{AB, BC, CD\}$

请运用保持函数依赖集的测试算法验证ρ是否保持函数依赖性。

解：算法的第一步是计算F在每一个关系模式Ri上的投影：

∵ $A^+=ABCD$， $B^+=ABCD$， $C^+=ABCD$， $D^+=ABCD$

∴ $\pi_{AB}(F)=\{A\to B，B\to A\}$ $\pi_{BC}(F)=\{B\to C，C\to B\}$

$\pi_{CD}(F)=\{C\to D，D\to C\}$

算法的第二步是逐步验证F中每个FD是否被$\bigcup_{i=1}^{k}\pi_{R_i}(F)$逻辑蕴涵。

∵ $\bigcup_{i=1}^{k}\pi_{R_i}(F)=\{A\to B, B\to A, B\to C, C\to B, C\to D, D\to C\}$ 只要判断其是否逻辑蕴涵 $D\to A$?

由 $D\to C$，$C\to B$，$B\to A$ 可推出 $D\to A$，

∴ $\rho=\{AB，BC，CD\}$ 保持 FD。

举例2：设关系模式R（ABC）分解成ρ=｛ AB，BC ｝，如果R上的FD集F=｛ A→B ｝，那么这个分解是损失分解。试举出R的一个关系r，不满足 m_ρ（r）= r。

解：反例 r 可以用测试时的初始表格: 即满足F的泛关系r如下所示

r	A	B	C
AB	a_1	a_2	b_{13}
BC	b_{21}	a_2	a_3

π_{AB}（r）

A	B
a_1	a_2
b_{21}	a_2

π_{BC}（r）

B	C
a_2	b_{13}
a_2	a_3

π_{AB}（r）$\bowtie\pi_{BC}$（r）有四个元组:

A	B	C
a_1	a_2	b_{13}
a_1	a_2	a_3
b_{21}	a_2	b_{13}
b_{21}	a_2	a_3

∴ m_ρ（r）$\neq$r

解：反例 r 可以用测试时的初始表格: 即满足F的泛关系r如下所示

r	A	B	C
AB	a_1	a_2	b_{13}
BC	b_{21}	a_2	a_3

$\pi_{AB}(r)$	A	B
	a_1	a_2
	b_{21}	a_2

$\pi_{BC}(r)$	B	C
	a_2	b_{13}
	a_2	a_3

π_{AB}（r）$\bowtie\pi_{BC}$（r）有四个元组:

A	B	C
a_1	a_2	b_{13}
a_1	a_2	a_3
b_{21}	a_2	b_{13}
b_{21}	a_2	a_3

$\therefore m_\rho$（r）$\neq r$

举例3：设有关系模式R（职工编号，日期，日营业额，部门名，部门经理），该模式统计商店里每个职工的日营业额，以及职工所在的部门和经理信息。

如果规定：每个职工每天只有一个营业额；每个职工只在一个部门工作；每个部门只有一个经理。

试回答下列问题：

（1）根据上述规定，写出模式R的基本FD和关键码；

（2）说明R不是2NF的理由，并把R分解成2NF模式集；

（3）进而分解成3NF模式集。

解：（1）据： 每个职工每天只有一个营业额，

每个职工只在一个部门工作，每个部门只有一个经理。

得到基本的函数依赖有三个：（职工编号，日期）→ 日营业额

职工编号 → 部门名， 部门名 → 部门经理

R的关键码为（职工编号，日期）

（2） R中有两个这样的FD：

（职工编号，日期）→（部门名，部门经理）

职工编号 → （部门名，部门经理）

由于前一个函数依赖是部分函数依赖，所以**R不是2NF模式**。

R应分解成 R1（ <u>职工编号</u>，部门名，部门经理 ）

R2（ <u>职工编号，日期</u>，日营业额 ）

此时，**R1**和**R2**都是**2NF**模式。

R1（职工编号，部门名，部门经理）

R2（职工编号，日期，日营业额）

（3）R2已是3NF模式。

但在R1中存在传递依赖：职工编号 → 部门经理

∵ 职工编号 → 部门名， 部门名 → 部门经理

∴ **R1不是3NF模式。**

将 R1 分解为： **R11**（ 职工编号，部门名 ）

R12（ 部门名，部门经理 ）

这样，**ρ= { R11，R12，R2 }是一个3NF模式集。**

五、模式设计方法的原则

关系模式 R < U , F > 分解成数据库模式:

ρ=｛R1，…，Rk｝，一般应具有三个特性:

① ρ是BCNF模式集，或3NF模式集;

② 无损分解，即对于R上任何满足F的泛关系 r 应满足

$$r = \bowtie_{i=1}^{K} \pi_{R_i}(r)$$

③ 保持函数依赖集F，即 $\left(\bigcup_{i=1}^{K} \pi_{R_i}(F)\right) \models F$

关系模式设计方法的原则：

模式设计方法应符合三条原则：表达性, 分离性，最小冗余性。

表达性 涉及两个数据库模式的数据等价和依赖等价问题,分别用无损联接和保持函数依赖来衡量。

分离性 是指属性间的间接联系应该用不同的模式表达，分离的基准是一系列范式。

最小冗余性 要求分解后的模式个数和模式中属性总数应最少。

§7 多值依赖和第四范式

函数依赖有效地表达了属性值之间的多对一联系；

多值依赖和联系依赖能刻划一部分 一 对多联系。

一、多值依赖（multivalued dependency，简记为MVD）

定义一： 设U是关系模式R的属性集，X、Y是U的子集。如果满足R的任一关系r，对于X 的某一个确定值，都存在Y的一组值与之对应,，且Y的这组值又与 Z=U－X－Y 中的属性值不相关，此时称Y多值依赖于X，或称X多值决定Y，记为：$X \rightarrow\rightarrow Y$。

例:模式R(DNAME，TNAME，TSEX ,SNAME，SSEX):

DNAME	TNAME	TSEX	SNAME	SSEX
计算机	马亮	男	吴康	男
计算机	马亮	男	刘红	女
计算机	李燕	女	吴康	男
计算机	李燕	女	刘红	女
……	……	……	……	……
通讯	汪宏伟	男	李志鸣	男
通讯	钱红	女	李志鸣	男

模式R的关键码是（ TNAME ， SNAME ）

其中有函数依赖： TNAME → TSEX，SNAM → SSEX。

还有多值依赖：DNAME →→（TNAME，TSEX）

DNAME →→（SNAME，SSEX）

这个例子也说明了多值依赖具有互补性：X →→ Y 成立

就一定有 X →→ Z 成立（ Z ＝ U－X－Y ）。

例:模式R(DNAME，TNAME，TSEX ,SNAME，SSEX):

DNAME	TNAME	TSEX	SNAME	SSEX
计算机	马亮	男	吴康	男
计算机	马亮	男	刘红	女
计算机	李燕	女	吴康	男
计算机	李燕	女	刘红	女
……	……	……	……	……
通讯	汪宏伟	男	李志鸣	男
通讯	钱红	女	李志鸣	男

模式R的关键码是 （ **TNAME**， **SNAME** ）

其中有函数依赖：TNAME → TSEX，SNAME → SSEX

还有多值依赖：DNAME →→（TNAME，TSEX）

DNAME →→（SNAME，SSEX）

这个例子也说明了：

多值依赖具有互补性：如果 X →→ Y 成立，

就一定有 X →→ Z 成立（ Z ＝ U－X－Y ）。

显然，这里存在着数据冗余和操作异常现象。

产生问题的原因是:教师和学生没有直接的联系。教师与系有直接的联系，学生与系有直接的联系，而教师与学生之间的联系是一种间接的联系。

把有间接联系的属性放在一个模式中就会产生冗余和异常现象。

在模式R中，一个系有很多教师（一对多联系），一个系有很多学生（一对多联系），教师与学生间没有直接联系。这种属性间的一对多联系称为多值依赖。

显然关系模式R中至少有三个属性，才有可能存在多值依赖。

函数依赖可以看成是多值依赖的特殊情况。

即：满足函数依赖的一定是多值依赖，

而多值依赖是函数依赖的概括，

即存在多值依赖的关系，并不一定存在函数依赖关系。

二、关于FD和MVD的推理规则集

设U是关系模式R上的属性集，W、V、X、Y、Z为U的子集，关于FD和MVD的推理规则有以下几条：

A1（FD自反性）：若$Y \subseteq X$，则$X \to Y$。

A2（FD增广性）：若$X \to Y$，且$Z \subseteq U$，则$XZ \to YZ$。FD

A3（FD传递性）：若$X \to Y$，$Y \to Z$则$X \to Z$。

A4（MVD补规则）：若$X \to\to Y$，则$X \to\to U-XY$。

A5（MVD的增广性）：若$X \to\to Y$，且$V \subseteq W$，则$WX \to\to VY$。

A6（MVD的传递性）：若$X \to\to Y$，$Y \to\to Z$，则$X \to\to Z-Y$。

A7（复制性，replication）：若X→Y，则X→→Y。

A8（接合性，coalescence rule）：若X→→Y，W→Z，

并且Z⊆Y，W∩Y=φ，那么X→Z。

根据A1~A8规则，还可以推出另外的推理规则：

A9（MVD的并规则):若X→→Y，X→→Z,则X→→YZ。

A10（MVD的交规则):若X→→Y，X→→Z,则X→→Y∩Z。

A11（MVD的差规则):若X→→Y，X→→Z,则X→→Y－Z，

X→→Z－Y。

A12（MVD的伪传递):若X→→Y,WY→→Z,则WX→→Z－WY。

A13（混合伪传递):若X→→Y，XY→Z，则X→Z－Y。

在有FD和MVD情况下，也可以用表格法（chase过程）来测试关系模式R相对于已知的FD和MVD集分解成ρ是否为无损分解。

另外，我们也可以用无损分解概念来定义MVD。

定义二：若U是关系模式R的属性集, X、Y、Z是U的一个分割。

若对R的每一个关系r,都有 $r = \pi_{XY}(r) \bowtie \pi_{XZ}(r)$，

则称 MVD X→→Y 在R（U）上成立。

这个定义说明：**如果一个模式可以无损分解成两个模式, 那么蕴涵着一个多值依赖。**

三、第四范式（4NF）

定义三：设D是关系模式R上成立的FD和MVD集合。如果D中每个非平凡的MVD:X→→Y的左部X都是R的超键，称R是4NF的模式。

例：在模式R1（DNAME，TNAME，SNAME）中，

键是：(TNAME,SNAME)

存在多值依赖：DNAME→→TNAME， DNAME→→SNAME

左部未包含键，因此R1不是4NF。

把R1分解成 R11（DNAME，TNAME）和

R12（DNAME，SNAME），

则 R11 和 R12 都是4NF。

从4NF的定义可知，是4NF的模式肯定是BCNF模式。

函数依赖和多值依赖是两个重要的数据依赖。

如果只考虑函数依赖,则BCNF是最高的关系模式范式;

如果考虑多值依赖，则4NF是最高的关系范式。

数据依赖中除函数依赖和多值依赖外,还存在着联接依赖。

联接依赖是与关系分解和联接运算有关的函数依赖，

联接依赖是研究5NF的理论基础。

四、联接依赖和第五范式（5NF）

在定义二中，MVD定义为一个模式无损分解为两个模式。类似地，对于一个模式无损分解成n个模式的数据依赖，称为联接依赖。

1、联接依赖的形式定义如下

定义四： 设U是关系模式R的属性集，R_1、…、R_n是U的子集，并满足$U=R_1 \cup \ldots \cup R_n$，$\rho=\{R_1, \ldots, R_n\}$是R的一个分解。如果对于R的每个关系r都有 $m_\rho(r) = r$，那么称联接依赖（join dependency，简记为JD）在模式R上成立,记为 $*(R_1, \ldots, R_n)$。

定义五： 如果 *（R_1，…，R_n）中某个 R_i 就是R，

那么称这个JD是平凡的JD。

如果模式R存在联接依赖，其关系r必将存在冗余和异常现象。譬如在元组插入或删除就会出现各种异常。

2、第五范式（5NF）

定义六：　如果关系模式R的每个联接依赖JD均由R的候选键蕴涵，那么称R是5NF的模式。

5NF也称为投影联接范式（project-join NF，简记为PJNF）。

这里联接依赖JD可由R的键蕴涵，是指JD可由键推导得到。如果JD *（R_1，…，R_n）中某个R_i就是R，那么这个JD是平凡的JD；如果JD中某个R_i包含R的键，那么这个JD可用chase方法验证。

联接依赖也是现实世界属性间联系的一种抽象，是语义的体现。

但是它不像FD和MVD的语义那么直观，要判断一个模式是否5NF也比较困难。

对于JD，已经找到一些推理规则，但尚未找到完备的推理规则集。可以证明，5NF的模式也一定是4NF的模式。根据5NF的定义，可以得出一个模式总是可以无损分解成5NF模式集。

关系规范化小结范式：

在关系数据库中,对关系的基本要求是满足第一范式。在此基础上,为了消除关系模式存在插入异常、删除异常、修改异常和数据冗余等问题,要对关系模式进一步规范化,使之逐步达到2NF、3NF、BCNF、4NF和5NF。

对于一个已满足1NF的关系模式:

当消除了非主属性对码的部分函数依赖后,它就属于2NF;

当消除了非主属性对码的部分和传递函数依赖，它就属于3NF;

当消除了主属性对码的部分和传递函数依赖，它就属于BCNF;

而当消除了非平凡且非函数依赖的多值依赖，它就属于4NF了;

最后，当消除了不是由候选关键字的连接依赖,它就属于5NF了。

其规范化过程如下图所示:

精读和习题要求

精读教材：P. 107~P. 125

习 题5： P. 126 2～5、7、8、

14～ 18