

Microsoft

Visual FoxPro 6.0

实用数据库编程



实用数据库编程系列丛书



王海兵 王 虎 等编著
中国科学技术大学出版社

TP311.132.3
W4B/1

Visual FoxPro 6.0 实用数据库编程

王海兵 王 虎 等编著



中国科学技术大学出版社

0051504

图书在版编目(CIP)数据

Visual FoxPro 6.0 实用数据库编程/王海兵等编著. —合肥:中国科学技术大学出版社,
1999.7

ISBN 7-312-01114-4

I. V… II. 王… III. 关系数据库-数据库管理系统, FoxPro 6.0 IV. TP311.13

中国版本图书馆 CIP 数据核字(1999)第 26930 号

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号 邮编:230026)

中国科学技术大学印刷厂印刷

全国新华书店经销

开本:787×1092/16 印张:20.5 字数:510 千

1999 年 7 月第 1 版 1999 年 7 月第 1 次印刷

印数:1—4000 册

ISBN 7-312-01114-4/TP·236 定价:32.00 元

内 容 提 要

Visual FoxPro 6.0 是由美国 Microsoft 最近推出的程序开发工具软件 Visual Studio 6.0 重要成员之一，主要用于小型数据库系统的开发。由于 dBASE、FoxBASE⁺、FoxPro 2.5、2.6、3.0 等数据库语言在国内应用非常广泛，特别是在很多高校内都作为非计算机专业在校学生的数据库课程，所以 Visual FoxPro 6.0 有着非常强大的读者群。

本书主要介绍 Visual FoxPro 6.0 的新特征和简单的编程知识，并根据培训教程的特点进行编写，所以非常适合作为 Visual FoxPro 6.0 方面的入门参考书，同时也可作为各类 Visual FoxPro 6.0 培训班和在校本专科生的教科书。



JS343/08

目 录

第一章 数据库概述	(1)
1.1 数据库概念简介	(1)
1.2 数据库类型	(1)
1.2.1 关系数据库	(1)
1.2.2 层次数据库	(4)
1.2.3 网状数据库	(4)
1.3 数据库技术的发展趋势	(5)
1.3.1 分布式数据库	(5)
1.3.2 多媒体数据库	(5)
1.3.3 面向对象的数据库	(5)
1.3.4 智能数据库	(5)
1.4 数据库设计	(5)
1.4.1 组织数据	(5)
1.4.2 确定主码	(6)
1.4.3 规范化	(6)
1.5 小 结	(9)
第二章 安装 Visual FoxPro 6.0	(10)
2.1 安装要求	(10)
2.2 安装过程	(10)
第三章 Visual Foxpro 6.0 的新特点	(16)
第四章 项目管理器	(18)
4.1 项目文件的建立	(18)
4.2 项目管理器的使用	(23)
4.2.1 项目选项卡	(23)
4.2.2 树状结构的操作	(28)
4.2.3 项目管理器窗口的位置	(30)
4.2.4 项目管理器的按钮及菜单	(31)
4.2.5 设 计 器	(34)
4.2.6 向 导	(35)
4.3 小 结	(35)

第五章 数据库的创建	(36)
5.1 建立工作目录	(36)
5.2 建立数据库	(37)
5.2.1 利用“Database Designer (数据库设计器)”创建数据库	(37)
5.2.2 利用“Database Wizard (数据库向导)”来创建数据库	(40)
5.3 小 结	(43)
第六章 数据表的创建与操作	(44)
6.1 数据库表的创建	(44)
6.1.1 利用“Table Wizard (表向导)”创建数据库表	(45)
6.1.2 利用“Table Designer (表设计器)”创建数据库表	(50)
6.2 自由表的创建	(54)
6.2.1 创建自由表	(54)
6.2.2 将自由表添加到数据库中	(56)
6.3 维护数据表结构	(57)
6.4 输入数据表记录	(59)
6.5 维护数据表记录	(64)
6.5.1 编辑字段数据内容	(64)
6.5.2 复制、剪切和粘贴数据	(65)
6.5.3 删除或复原表中的记录	(65)
6.6 数据表字段特性	(70)
6.6.1 字段的显示属性	(71)
6.6.2 字段注解	(72)
6.6.3 数据验证	(73)
6.7 索引数据表	(76)
6.7.1 建立索引	(76)
6.7.2 查看索引	(78)
6.7.3 改变索引	(79)
6.7.4 使用索引	(79)
6.7.5 索引的作用	(81)
6.8 浏 览	(82)
6.8.1 定制浏览窗口	(83)
6.8.2 调整浏览窗口的行和列	(83)
6.8.3 去除网格线	(86)
6.8.4 分割浏览窗口	(86)
6.9 查找与替换	(88)
6.9.1 查 找	(88)
6.9.2 替 换	(93)
6.10 创建和编辑关系	(95)

6.10.1 创建永久关系	(96)
6.10.2 编辑关系	(97)
6.10.3 建立表的参照完整性	(98)
6.11 小 结	(100)
第七章 查 询	(101)
7.1 查询的含义	(101)
7.2 创建查询	(102)
7.2.1 利用“Query Wizard (查询向导)”创建查询	(102)
7.2.1 利用“Query Designer (查询设计器)”创建查询	(107)
7.3 修改查询	(129)
7.4 复杂条件查询	(131)
7.4.1 AND 关系	(131)
7.4.2 OR 关系	(132)
7.4.3 组合 AND 和 OR 关系	(133)
7.5 交叉表查询	(134)
7.6 小 结	(138)
第八章 视 图	(139)
8.1 使用向导创建本地视图	(140)
8.2 用“视图设计器”创建本地视图	(146)
8.3 远程视图的创建	(159)
8.3.1 用“Remote View Wizard (远程视图向导)”创建远程视图	(160)
8.3.2 使用“View Designer (视图设计器)”创建远程视图	(161)
8.4 创建参数视图	(163)
8.5 小 结	(165)
第九章 报表和标签	(166)
9.1 使用“报表向导”创建报表	(166)
9.2 使用“一对多报表向导”创建报表	(172)
9.3 使用“快速报表”创建报表	(176)
9.4 使用“报表设计器”创建报表	(181)
9.4.1 启动 Report Designer	(181)
9.4.2 设置报表布局	(182)
9.4.3 添加报表控件	(184)
9.4.4 美化报表设计	(193)
9.4.5 分组/总计报表	(199)
9.5 创建标签	(202)
9.5.1 使用“Label Wizard (检签向导)”创建标签	(202)

9.5.2 使用“Label Designer (标签设计器)”创建标签	(204)
9.6 打印报表或标签	(206)
9.7 小 结	(206)
第十章 表 单	(207)
10.1 什么是表单	(207)
10.2 表单的创建	(207)
10.2.1 利用“Form Wizard (表单向导)”创建表单	(207)
10.2.2 利用“Form Designer (表单设计器)”创建表单	(219)
10.2.3 利用“Form Builder (表单生成器)”快速创建表单	(225)
10.3 向表单中增添和删除控件	(227)
10.3.1 标签控件	(228)
10.3.2 文本框控件	(232)
10.3.3 编辑框控件	(234)
10.3.4 列表框控件	(236)
10.3.5 组合框控件	(239)
10.3.6 页框控件	(239)
10.3.7 表格控件	(241)
10.3.8 项目组控件	(244)
10.3.9 命令组控件	(245)
10.3.10 OLE 控件	(247)
10.3.11 添加 Active X 控件	(248)
10.3.12 添加表单向导中的控件	(250)
10.4 美化表单	(252)
10.4.1 改变控件大小	(252)
10.4.2 改变文本的字体和大小	(252)
10.4.3 移动和对齐控件	(253)
10.4.4 表单颜色	(253)
10.5 小 结	(253)
第十一章 设计菜单与工具栏	(254)
11.1 菜单系统概述	(254)
11.2 设计菜单	(257)
11.3 用菜单设计器设计菜单	(257)
11.4 设计快捷菜单	(264)
11.5 设计工具栏	(266)
11.5.1 新建工具栏	(266)
11.5.2 定制工具栏	(268)
11.6 小 结	(270)

第十二章 函数、命令与程序设计	(271)
12.1 程序设计基础	(271)
12.2 变 量	(272)
12.2.1 变量的数据类型	(272)
12.2.2 变量的定义	(273)
12.2.3 变量的赋值	(276)
12.2.4 数组的定义与使用	(276)
12.3 操 作 符	(277)
12.4 常用命令	(278)
12.4.1 命令的格式	(278)
12.4.2 操作数据库命令	(279)
12.4.3 操作数据表命令	(281)
12.4.4 操作记录命令	(285)
12.4.5 选项操作指令	(291)
12.5 常用函数	(291)
12.6 程序设计	(294)
12.7 过程和自定义函数	(296)
12.8 程序流程控制语句	(297)
12.9 结构化程序设计与面向对象程序设计	(300)
12.10 小 结	(300)
第十三章 实 例	(301)
13.1 建立项目文件	(301)
13.2 建立工作目录	(301)
13.3 创建数据库	(302)
13.4 创建数据表	(303)
13.4.1 “学生基本情况”表的创建(利用表设计器)	(303)
13.4.2 “系别”表的创建(利用表向导)	(304)
13.4.3 创建“获奖情况”表	(306)
13.4.4 创建和编辑表间的关系	(308)
13.5 创建查询	(309)
13.6 创建视图	(311)
13.6.1 本地视图	(311)
13.6.2 参数视图	(312)
13.7 创建标签	(313)
13.7.1 创建报表	(313)
13.7.2 创建标签	(314)
13.8 创建表单	(314)
附录 文件格式列表	(317)

第一章 数据库概述

数据库系统就是能管理大量的、持久的、可靠的、共享的数据的工具。所谓大量是指数据量大，需要大容量的外存支持；所谓持久是指数据必须长久地保留，不是某一特定应用之后就不再需要了；所谓可靠是指发生软硬件故障后，仍可以恢复数据库；所谓共享，是指若干用户能按一定有序的方式存取数据，能避免同步存取可能会造成的错误。

1.1 数据库概念简介

数据库是一个用特殊方法组织的有用数据的集合。一个数据库是由一个应用系统所采用的一个或多个数据库文件的全部数据构成。这些数据可以是数值、文字、日期、货币等类型。数据库的大小与数据量的多少有关，建立数据库时，数据项的多少、取舍，视用户的需要而定。

1.2 数据库类型

根据对数据的不同组织方法，可将数据库分为三种类型：网状数据库、层次数据库、关系数据库，其中以关系型数据库最为流行、最为实用。

1.2.1 关系数据库

关系数据库由数据库结构和一组数据记录组成。数据库的结构，定义字段名、字段类型、字段长度等；数据库记录的集合，它包含全部实在的数据。因为关系数据库将信息按表格的形式组织，所以一个数据表就是数据库的基本单元。每个数据信息安排在相应的数据记录的字段中。

数据字段根据其存放的内容定义类型。存放字符型数据的字段定义为字符型字段，存放数值型数据的字段定义为数值型字段，等等。为节省容量和便于处理，字段存放数据的长度（字节数），必须在设计数据库定义字段时确定，此长度就是表格中各行的这个数据项最多的字符数。

因此，数据库结构中的字段名用于标识某记录的某个字段信息；字段类型确定如何操作使用信息，数值型字段可用于计算和查询，字符型字段绝不能直接出现在数学公式中用于计算；字段长度，可供预计存储量。

关系数据库能有效地存储和处理大量的数据信息。可归纳为如下的主要功能和作用：

- 查找满足需要的数据。
- 按要求排序或分类数据。

- 灵活的关系操作。例如，连接不同数据库的字段，即可按需要自动产生一张含多张表格的指定数据项的表格。
- 存储和维护数据库中的数据。可以避免为表格中部分修改而重选表格。
- 可按需要选用命令或设计程序，完成各种复杂的数据处理。

鉴于关系型数据库结构以上的优点，一般 PC 的数据库软件几乎都使用关系型数据库结构，因此 Visual Foxpro 也不例外。关系型数据库可将多个数据表，利用某一相同字段之间的关系而连结在一起，因此应用范围非常广泛。

根据两个表格间的连结方式的不同，关系型数据表可分作三种关系：一对一，一对多和多对多。下面分别介绍一下这三种关系。

1. 一对一

这是最简单的一种关系型数据表，即是两个数据表之间有一相同的字段数据，此字段记录是一个对应一个记录的关系。例如，下面是一个学生的生源数据表。

姓 名	学 号	毕 业 学 校
王 军	11741	成都十二中
吴 明	14321	北师大附中
张小红	25763	重庆三十八中
杨志宜	54312	山西大学附中

而另一数据表为学生的基本数据。

姓 名	性 别	年 龄	籍 贯
王 军	男	21	四川省成都市
吴 明	男	22	北京
张小红	女	20	重庆市
杨志宜	女	21	山西省太原市

两张数据表之间根据相同的“姓名”，可以相互关联起来，也就是可以同时找出学生的“籍贯”与“毕业学校”的关系。虽然这两类数据不在同一张数据表中，但因为具有相同的“姓名”关系而连结。

在这种关系中，仅有一个学生的生源数据对应于一个学生的基本数据，反之亦然，故称为一对一关系。

2. 一对多

表示数据表的一条记录可以对应到另一数据表的多条记录，反之不成立。

例如，第一个数据表是学生的基本数据。

姓 名	性 别	年 龄	学 号
王 军	男	21	11741
吴 明	男	22	14321
张小红	女	20	25763
杨志宜	女	21	54312

另一数据表是书本数据。

书 名	姓 名	书 名	姓 名
英语精读	王 军	数据库原理	张小红
计算机网络原理	王 军	VisualFoxpro5.0 入门图集	张小红
子夜	王 军	傲慢与偏见	张小红
平凡的世界	吴 明	实用 Unix 用户手册	杨志宜
Windows98 使用速成	吴 明	京华烟云	杨志宜
实用 Windows 编程	吴 明	多媒体技术	杨志宜

由于一个学生可以有多本书,但同一本书仅属于一个学生,所以两数据表之间是一对多的关系。

3. 多对多

表示一个数据表的某一条记录可以对应到另一数据表的多条记录,反之,第二个数据表的某一条记录也可以对应到第一数据表的多条记录。

例如,第一个数据表是学生表。

姓 名	学生—课程注册号	课 程
王 军	007	计算机图形学
王 军	142	数据库原理
吴 明	137	多媒体技术
张小红	349	计算机图形学
杨志宜	265	微机原理

第二个数据表是课程名表

课 程 名	学生—课程注册号
计算机图形学	349
数据库原理	142
微机原理	265
多媒体技术	137
计算机图形学	007

由于一个学生可以选多门课程,而同一门课程可以被多个学生选修。彼此的对应关系

即是多对多的关系。

1.2.2 层次数据库

层次数据库将数据信息按层次结构组织，呈现为树状结构，这种结构既标识了数据库中的数据，还定义了数据间的关系。最简单的结构是数据库中全部的数据元素按一对一的关系组织。一般的层次数据库，组织它的数据元素按一对一或一对多的关系结构。

在图 1-1 所示的数据库结构中，数据元素包括部门、固定资产类别和固定资产。这些元素之间用一条线连接，这样的线称为基本层次连系，常称为系。这些元素的关系可分别从上下往下看和从下往上看：

从上往下看的话，某些关系是一对多的关系。例如部门 1 关系到两个类别，每个类别同样都关系到几种固定资产。

从下往上看的话，一项固定资产和一种类别之间是一对一的关系，也就是说一项固定资产只属于一种类别。

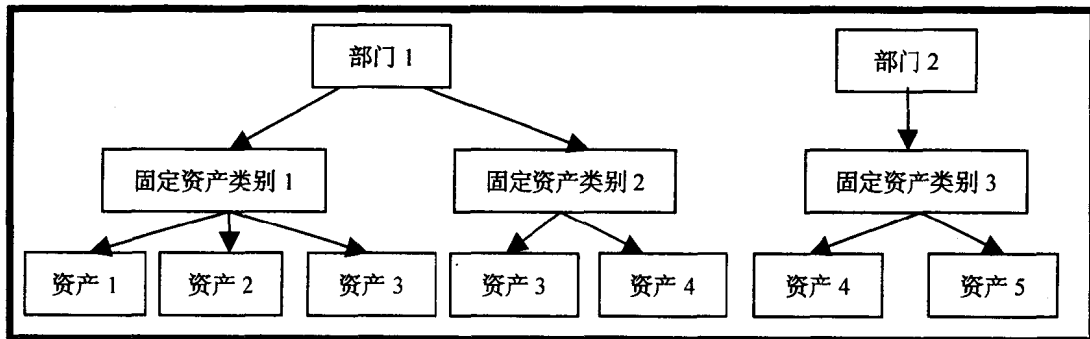


图 1-1 层次数据库结构

1.2.3 网状数据库

在网状数据库中，数据元素之间可以有多种联系。图 1-2 表示网状结构的数据库。这个结构中的关系是多对多的，每个数据元素都是唯一的。

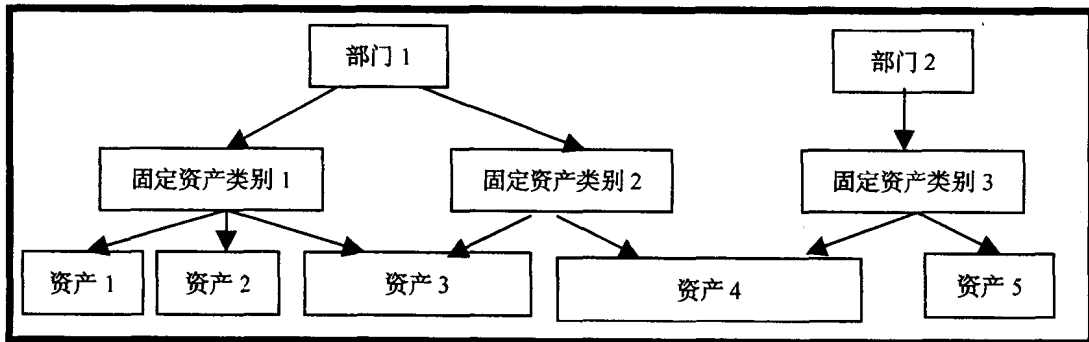


图 1-2 网状数据库结构

1.3 数据库技术的发展趋势

随着微机硬、软件性能的不断提高，在微机环境下运行的数据库也在不断完善。

1.3.1 分布式数据库

Client / Server（客户/服务器）系统有利于用户建立分布式数据库，分布式处理是当今数据库应用的一个发展方向。

从体系结构上看，Client / Server（客户/服务器）是由网络连接的多台硬件组成的协同工作环境，其服务器存储需要共享的数据，运行复杂的事务处理过程。客户机存储、处理本地数据。客户通过网络提出请求，服务器接受请求，并将处理结果返回客户机。

1.3.2 多媒体数据库

多媒体数据库提供在数据库中存储、处理多媒体数据（如：图像、声音、影像）的能力，提供能显示、编辑、压缩、还原多媒体数据信息的多媒体开发工具，逐步满足现实世界对各种媒体数据资料的处理要求。

1.3.3 面向对象数据库

随着数据库应用的发展，数据库将支持新的数据类型及其操作；支持抽象数据类型和用户定义类型的可扩展能力；具有存储和处理复杂对象的能力；提供面向对象的开发语言和程序设计方法及工具。

1.3.4 智能数据库

智能数据库技术将人工智能、数理逻辑与数据库技术相结合，构造专家数据库、知识数据库、演绎数据库等，并集成综合的智能数据库系统。

1.4 数据库设计

当收集有关需要做些什么的信息时，显然，也要收集关于需要输入、存储什么数据，需要发送什么消息以及在报表上都需要打印些什么的信息。

1.4.1 组织数据

收集数据时，将它们组织成表，并且开始整理出它们之间的联系。可以使用自己的图

表系统在纸上做出这项工作，或者也可以从一些诸如实体关系图或者约束图等标准化的作图方法体现这些联系。首要目标是明确需要哪些表和在每个表中放置哪些列（字段）。

1.4.2 确定主码

接下来要做的就是为每个表确定一个主码。主码是指能够唯一确定一行的一列或者一组列。两个不同行的主码列上重复的信息被认为是数据库中重复的行，而不论其余列的内容如何。因此，选择主码一定要小心。

你也许会发现，表中有多于一组的列都能够唯一地确定每一行，这些被称作候选码。从这些候选码中，应选择出一个最好的作为主码。怎样决定哪一个码是最好的呢？有以下几个标准以参考，对于哪个码回答“yes”的越多，那个就越好。

- 码将会在将来保持唯一吗？如果它现在是唯一的，并不意味着它在将来是唯一的，因为系统在将来可能会增强。
- 码中的任何列有可能改变吗？经常改变的数据不应成为码的一部分。
- 码中所包含的列是否不超过两列到三列？因为如果想要连接其它表，对于这些表中的每一个，构成码的五列或六列都要被再一次复制，这就可能成为维护方面的一大障碍。

哇！对每个表要施加这么多标准。如果表中没有一个码符合条件怎么办？如果不能找出作为唯一标志的一组列怎么办呢？在这种情况下，通常有必要创建一个任意码或者替代码。

什么是任意码呢？例如某公司的固定资产从来没进行编号，但数据库中还是创建了一个“编号”列。这一列不保存任何数据，只为每项固定资产创建一个唯一标志符。

1.4.3 规范化

规范化只是一系列能够应用到数据库结构上的测试。建立这些测试是为了看数据结构是否处于一种优化形式。这些测试帮助你识别冗余数据和潜在的完整性问题，以及其它浪费空间和时间的罪魁祸首。

目前，有五个规范化例试，但前三个是最重要的，我们就讨论这三个测试。当数据库结构通过了第一个测试，就说它符合第一范式；当它通过了第一个和第二个测试，就符合第二范式；以此类推。

1. 规范化测试一

测试：对于每一行和列的交叉点，是否仅有一个值？

一个表由行和列组成。对于任何给定行和列的交叉点，可以有一个，并且只能有一个数据项。例如表 1-1 具有固定资产编号、固定资产名称及使用部门。

表 1-1 固定资产和使用部门表

固定资产编号	固定资产名称	使用部门
1610101	丰田可罗纳卧车	办公室、餐饮部
1610102	卡星车	财务部、洗涤部
1611147	复印机	经理部、公关部

对于每一行，使用部门具有多于一个数据项，这是一个不该出现的问题。怎样才能达到第一范式呢？把它分成两个表，如表 1-2 和 1-3 所示。

表 1-2 固定资产表

固定资产编号	固定资产名称
1610101	丰田可罗纳卧车
1610102	卡星车
1611147	复印机

表 1-3 使用部门表

固定资产编号	使用部门
1610101	办公室
1610101	餐饮部
1610102	财务部
1610102	洗涤部
1611147	经理部
1611147	公关部

现在该表通过了测试，这就是规范化的第一范式的全部。

2. 规范化测试二

测试：是否所有的非主码列都要依赖于全部主码？

想象一个固定资产变更系统，表 1-4 保存了变更信息，头两列是主码。

表 1-4 固定资产变更表

日期	固定资产编号	数量	资产描述
1998/09/29	1610104	2	奔腾 II 计算机
1998/09/30	1610312	1	Epson 打印机
1998/10/01	1611201	1	长虹 29 寸彩电

日期、固定资产编号列充当主码，数量和资产描述是非主码列。数量提供在日期所指定的这一天发生的变更情况，固定资产编号为所指定的资产的数目，因此，数量列依赖于码中的所有列。但是，资产描述是对指定固定资产的描述，它只依赖于那一列，与日期毫无关系。这就是这个表为什么不能通过测试的原因。如果通过这种组织方法，浪费了数据库中的大量空间。

其解决方法是再一次分裂这个表，产生另一个表，如表 1-5 和表 1-6 表示。

表 1-5 新的固定资产变更表

日 期	固定资产编号	数 量
1998/09/29	1610104	2
1998/09/30	1610312	1
1998/10/01	1611201	1

表 1-6 固定资产表

固定资产编号	资产描述
1610104	奔腾 II 计算机
1610312	Epson 打印机
1611201	长虹 29 寸彩电

尽管有了一个新表的额外开销，通过移去重复的资产描述冗余性节省了空间，拥有了一个更合理的数据库结构。

3. 规范化测试三

测试：表中有任何依赖其他非主码列的非主码列吗？

举例可能是解释用意的最好方式了。假设你负责为各个部门分配固定资产。表 1-7 是这个表的第一个版本。固定资产编号是主码。

表 1-7 资产—部门安排表

固定资产编号	固定资产名称	部门号	部门名
1610101	丰田可罗纳卧车	0004	办公室
1610203	桑塔纳 2000 型	0002	公关部
1611147	复印机	0045	经理部

我们来仔细分析一下这个表，对于每一行和列的交叉点，只有一个数据项——符合第一范式。它没有任何一个依赖主码部分的非主码列——符合第二范式。但是仍有点问题。

部门名依赖于部门号。如果具有任何依赖于表中的其它非主码列的列，就不能说这个表符合第三范式。同样需要把它们分开，这一次要分成三个表，如表 1-8、表 1-9 和表 1-10 所示。

表 1-8 固定资产表

固定资产编号	固定资产名称
1610101	丰田可罗纳卧车
1610203	桑塔纳 2000 型
1611147	复印机