

国内资深Excel应用专家与培训讲师倾力打造，全情奉献！

Excel +SQL Server 数据库管理技术详解

韩小良 等编著



CD-ROM

- 独一无二：本书是第一本将Excel与SQL Server数据库整合应用的经典图书
- 讲解全面：囊括了整合Excel与SQL Server数据库的各种基本知识、方法和案例分析
- 实操性强：提供了大量的实际案例，详细介绍了利用Excel VBA整合SQL Server数据库的方法和技巧
- 易学易用：提供了每个案例以及各种方法和技巧的详细源代码，这些源代码均被设计成了通用的程序，根据自己电脑的实际配置情况即可迅速套用



机械工业出版社
China Machine Press

TP391.13/136D

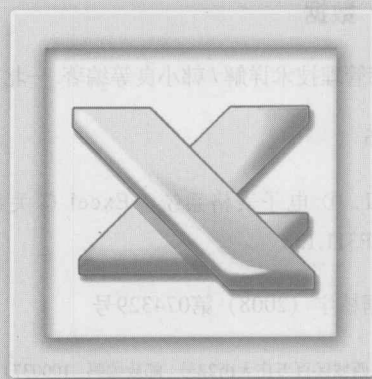
2008

Excel

+SQL Server

数据库管理技术详解

韩小良 等编著



机械工业出版社
China Machine Press

本书共分15章,结合大量有针对性的案例,详细介绍了Excel VBA、ADO、SQL等方面的基础知识以及这些知识在利用Excel VBA整合SQL Server数据库中的各种实际应用技能。通过这些实例,读者可以快速掌握利用Excel VBA来操作SQL Server数据库的方法和技巧,为利用Excel VBA开发各类管理信息应用系统提供坚实的基础。本书提供的这些案例大部分可直接应用到实际工作中,稍作修改也可应用于Excel与其他数据库的整合应用之中。

本书不仅适用于将Excel与SQL Server数据库整合应用的各类管理人员和管理系统开发人员,也适用于将Excel与其他数据库(如Access数据库、Foxpro数据库和Oracle数据库等)整合应用的各类管理人员和管理系统开发人员以及大专院校相关专业本科生、研究生和MBA学员等阅读使用。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

Excel+SQL Server数据库管理技术详解/韩小良等编著. —北京:机械工业出版社,2008.6

ISBN 978-7-111-24055-6

I. E… II. 韩… III. ① 电子表格系统, Excel ② 关系数据库—数据库管理系统, SQL Server IV. TP391.13 TP311.138

中国版本图书馆CIP数据核字(2008)第074329号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:李东震

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2008年6月第1版第1次印刷

190mm×260mm·19.75印张

标准书号:ISBN 978-7-111-24055-6

ISBN 978-7-89482-693-0(光盘)

定价:39.80元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010) 68326294

前 言

Microsoft Excel由于其强大的数据处理功能和计算分析功能,已经在社会经济生活的各个方面得到了广泛的应用。但是,Excel也有其局限性和缺点,例如存储的数据量是有限制的,对于Excel 2003及以前的版本,每张工作表仅能保存 $65\,536 \times 256$ 个数据,如果要将工作簿的所有工作表都存满数据,一方面将占用大量的硬盘空间,另一方面也使工作表的计算速度变得极为缓慢。因此,当有大量的数据要保存时,应当使用数据库,如Access数据库、SQL Server数据库和Oracle数据库等,而不是Excel工作表。Excel的用途应该是对数据库查询出的数据进行计算和分析,也就是发挥Excel的数据计算、统计分析、图表和制作报表等卓越功能。将Excel与数据库结合起来,可以充分利用各自的优势,使工作更加得心应手。

在目前使用的各种数据库软件中,作为网络数据库的SQL Server,已经得到越来越多的应用。SQL Server提供了强大的数据管理与处理分析能力,用户可以利用SQL Server来开发各种管理系统。不过,对于企业管理信息系统而言,不可能在每个部门、每个员工的终端电脑上都安装一套SQL Server软件,而是企业自己有一个SQL Server服务器,每个终端电脑可以通过自己电脑里已有的Office办公软件来实现与SQL Server服务器的沟通和联系。这样,由于Office办公软件在每个电脑上都是必备的操作软件,而且其中的Excel具有强大的数据处理与分析能力,并且利用Excel VBA可以实现对SQL Server服务器的自动链接,从而使得每个人都可以任意地在SQL Server服务器上创建自己的数据库,并与他人共享。或者各个部门将数据汇总到SQL Server服务器的数据库中,使得办公效率大大提高。而当需要某些数据时,又可以利用Excel的现有数据导入工具或利用Excel VBA将SQL Server服务器数据库数据导入到Excel工作表,并进行处理和分析。

本书共15章,结合大量的有针对性的案例,详细介绍了Excel VBA、ADO、SQL等方面的知识以及这些知识在利用Excel VBA整合SQL Server数据库中的各种实际应用。通过这些案例,读者可以快速掌握利用Excel VBA来操作SQL Server数据库的方法和技巧,为利用Excel VBA开发各类管理信息应用系统提供坚实的基础。

第1章主要介绍了Excel的向导工具和SQL Server数据库的向导工具,实现Excel与数据库之间进行数据交换的基本方法,包括导入导出数据、利用SQL Server数据库数据创建数据透视表等。

第2章主要介绍了Excel VBA、ADO和SQL的基础知识,这是为以后利用Excel VBA操作SQL Server数据库提供基础。

第3章至第15章,结合大量精选的有针对性的案例,详细介绍如何利用Excel VBA操作SQL Server数据库和查询数据等。

本书的编写得到了很多学者和专家的帮助,并参考了一些文献和资料,由于无法一一列出,在此真诚地向这些学者、专家以及文献和资料的提供者表示感谢。

参与本书编写的人还有:高会东、王建超、邓薇、黄丽莉、吝晓宁、汪洋、白广元、蔡念光、陈辉、冯彬、刘长江、刘明、沙金、张士强、张洪福、多召英、贾旭、李宽、江宽、陈科、方成林、班晓娟、方中纯、刘兰军、郑雪峰。

编 者

2008年4月

目 录

前言

第1章 Excel与SQL Server的交互	1
1.1 将SQL Server数据库数据导入到Excel工作表	1
1.2 将Excel工作表数据导入到SQL Server数据库	3
1.3 以SQL Server数据库数据创建数据透视表	5
1.4 以SQL Server数据库数据创建多维数据透视表	10
第2章 Excel VBA/ADO/SQL概述	18
2.1 为什么要使用Excel管理SQL Server数据库	18
2.2 SQL Server数据库概述	18
2.2.1 数据库命名规则	18
2.2.2 创建数据库	19
2.2.3 查看、编辑数据库数据	21
2.3 Excel VBA概述	21
2.3.1 了解VBA编辑器	22
2.3.2 VBA的对象、属性、方法和事件	26
2.3.3 Excel中的对象简介	27
2.3.4 VBA编程基础	33
2.3.5 工作簿对象和工作表对象事件编程	46
2.3.6 窗体和控件	51
2.3.7 创建自定义工具栏和自定义菜单	58
2.4 ADO数据库访问技术概述	61
2.4.1 引用ADO	61
2.4.2 ADO的对象	62
2.4.3 记录集对象	63
2.4.4 关于ADOX	63
2.5 SQL语言概述	65
2.5.1 SQL运算符	65
2.5.2 SQL合计函数	66
2.5.3 SQL常用语句	66
第3章 获取SQL Server数据库信息	69
3.1 建立与SQL Server服务器的连接	69

3.2 建立与SQL Server数据库的连接	70
3.3 判断数据库是否存在	71
3.3.1 利用SQL语句判断数据库是否存在	71
3.3.2 利用存储过程判断数据库是否存在	73
3.4 判断数据表是否存在	74
3.4.1 利用SQL语句判断数据表是否存在	74
3.4.2 利用ADO的OpenSchema方法判断数据表是否存在	76
3.4.3 利用ADOX判断数据表是否存在	78
3.5 判断字段是否存在	80
3.5.1 利用SQL语句判断字段是否存在	80
3.5.2 利用ADO的OpenSchema方法判断字段是否存在	82
3.5.3 利用ADOX判断字段是否存在	83
3.5.4 利用查询记录集判断字段是否存在	84
3.6 判断索引是否存在	85
3.6.1 利用SQL语句判断索引是否存在	85
3.6.2 利用OpenSchema方法判断索引是否存在	86
3.6.3 利用ADOX判断索引是否存在	87
3.7 获取SQL Server服务器上所有的数据库清单	88
3.7.1 利用SQL语句获取数据库清单	88
3.7.2 利用存储过程获取数据库清单	90
3.8 获取指定数据库中所有的数据表信息清单	91
3.8.1 利用SQL语句获取数据表信息清单	91
3.8.2 利用ADO的OpenSchema方法获取数据表信息清单	92
3.8.3 利用ADOX获取数据表信息清单	95
3.9 获取指定数据表中所有的字段信息清单	97
3.9.1 利用SQL语句获取指定数据表的全部字段信息清单	97
3.9.2 利用OpenSchema方法获取指定数据表的全部字段信息清单	99
3.9.3 利用数据表查询记录集对象获取指定数据表的全部字段信息清单	102
3.9.4 利用ADOX获取指定数据表中所有字段信息清单	103
3.10 获取指定数据表中所有的索引清单	104
3.10.1 利用SQL语句获取指定数据表的全部索引信息清单	104
3.10.2 利用OpenSchema方法获取指定数据表的全部索引信息清单	106
3.10.3 利用ADOX获取指定数据表中所有索引信息清单	108
第4章 自动创建和删除SQL Server数据库和数据表	110
4.1 创建数据库和数据表的基本方法	110
4.2 字段的数据类型	110
4.3 创建一个新的数据库和数据表	112

4.3.1	创建默认属性的新数据库和数据表	112
4.3.2	创建自定义属性的新数据库和数据表	115
4.4	重新设置数据库文件的属性	118
4.5	在已有的数据库中创建数据表	119
4.6	从SQL Server服务器上删除数据库	121
4.7	从指定的数据库中删除数据表	122
第5章	编辑SQL Server数据库和数据表	123
5.1	重命名数据库	123
5.1.1	利用标准SQL语句重命名数据库	123
5.1.2	利用SQL语句重命名数据库	124
5.2	重命名数据表	125
5.2.1	利用标准SQL语句生成表查询的方法重命名数据表	125
5.2.2	利用SQL语句重命名数据表	126
5.3	编辑字段	127
5.3.1	利用SQL语句添加新字段	127
5.3.2	使用存储过程修改字段名称	128
5.3.3	使用SQL语句修改字段的数据类型和字段长度等	129
5.3.4	利用SQL语句删除字段	130
5.3.5	利用ADOX删除字段	132
5.4	编辑索引	132
5.4.1	利用SQL语句创建索引	133
5.4.2	利用SQL语句删除索引	134
5.4.3	利用ADOX删除索引	135
5.5	编辑外键	136
5.5.1	利用SQL语句创建外键	136
5.5.2	利用SQL语句删除外键	137
5.5.3	利用ADOX删除外键	138
5.6	设置默认值	139
5.6.1	在创建数据表时设置默认值	139
5.6.2	为现有的数据表字段设置默认值	140
5.6.3	删除数据表字段的默认值	141
第6章	查询SQL Server数据库概述	143
6.1	SQL查询语言概述	143
6.1.1	SELECT查询语句的语法结构	143
6.1.2	SQL查询语言的有关函数	144
6.2	将查询结果复制到Excel工作表	144
6.2.1	利用Range对象的CopyFromRecordset方法复制查询结果	144

6.2.2	利用循环的方法复制查询结果	146
6.2.3	对复制到工作表的查询结果进行规范处理	150
6.3	将查询结果显示在窗体	151
6.3.1	将查询结果显示在窗体——利用ListView控件	151
6.3.2	将查询结果显示在窗体——利用浏览按钮	156
第7章	无条件查询	161
7.1	范例数据库设计及使用	161
7.1.1	范例数据库设计	161
7.1.2	附加范例数据库到您的SQL Server服务器上	163
7.2	查询表中所有记录	164
7.3	查询表中所有记录的指定字段	166
7.4	查询结果不显示重复记录	167
7.5	查询前 n 条记录	168
7.6	查询前百分数 n 的记录	170
7.7	将表达式作为查询字段进行查询	171
7.8	将查询结果以别名输出	172
7.9	将字段合并并以别名查询输出	173
7.10	合并查询	175
第8章	条件查询	177
8.1	WHERE条件子句的语法结构	177
8.2	WHERE条件子句中的运算符	177
8.2.1	关系运算符	177
8.2.2	逻辑运算符	178
8.2.3	特殊运算符	178
8.3	单条件查询	179
8.3.1	等于或不等于查询	179
8.3.2	列表(in或not in)查询	180
8.3.3	介于(between)查询	182
8.3.4	模糊(like)查询	183
8.3.5	是否为空(is null或is not null)查询	184
8.4	复杂条件查询	185
8.5	计算条件下的查询	187
第9章	排序查询	189
9.1	ORDER BY排序查询的语法结构	189
9.2	将多个字段按照不同的排序方式进行排序	189
9.3	根据表达式对查询结果进行排序	191
9.4	根据字段别名对查询结果进行排序	192

9.5 根据字段位置对查询结果进行排序	193
第10章 汇总和分组查询	194
10.1 聚合函数	194
10.2 使用聚合函数进行汇总查询	194
10.3 使用GROUP BY子句进行分组查询	197
10.4 使用HAVING子句进行分组查询	198
第11章 生成表查询	202
11.1 INTO查询的语法结构	202
11.2 将查询结果生成一个永久新表	202
11.3 将查询结果追加到已有的表中	204
第12章 多表连接查询	206
12.1 SELECT多表连接查询的语法结构	206
12.2 等值或非等值连接查询	206
12.3 自身连接查询	212
12.4 内连接查询	213
12.5 左外连接查询	214
12.6 右外连接查询	215
12.7 全外连接查询	216
第13章 嵌套查询	218
13.1 嵌套查询的语法结构	218
13.2 返回单值的嵌套查询	218
13.3 返回多值的嵌套查询	222
13.3.1 IN (NOT IN) 嵌套查询	222
13.3.2 SOME (ANY) 嵌套查询	225
13.3.3 ALL嵌套查询	225
13.3.4 EXISTS (NOT EXISTS) 嵌套查询	226
13.4 多层嵌套查询	227
第14章 编辑SQL Server数据库数据	229
14.1 添加新记录	229
14.1.1 一次添加一条记录	229
14.1.2 一次添加多条记录	230
14.1.3 将工作表的数据添加到数据库	231
14.1.4 如何为某字段插入空值 (NULL)	232
14.2 修改更新记录	233
14.2.1 一次修改一条记录	234
14.2.2 一次修改多条记录	234
14.3 删除记录	236

14.3.1 一次删除一条记录	236
14.3.2 一次删除多条记录	237
14.3.3 删除数据表的全部记录	238
14.4 编辑数据综合应用案例	238
14.4.1 设计窗体结构	238
14.4.2 设计程序代码	239
14.4.3 应用说明	243
第15章 数据维护	246
15.1 备份数据库	246
15.1.1 利用企业管理器备份数据库	246
15.1.2 利用程序自动备份数据库	248
15.2 恢复数据库	248
15.2.1 利用企业管理器恢复数据库	248
15.2.2 利用程序自动恢复数据库	249
附录 VBA参考	251
案例目录	298
参考文献	302

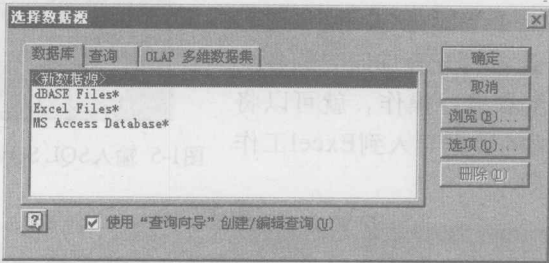
第1章 Excel与SQL Server的交互

本章主要介绍利用Excel的数据导入/导出向导工具查询获取SQL Server数据库数据，以及使用SQL Server应用程序的导入/导出向导工具将工作表数据保存到SQL Server数据库。关于如何利用Excel VBA来操作和控制SQL Server数据库，将从第2章开始进行介绍。

1.1 将SQL Server数据库数据导入到Excel工作表

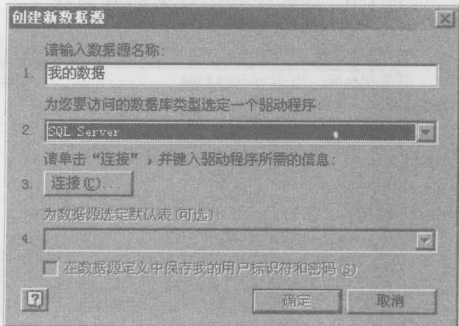
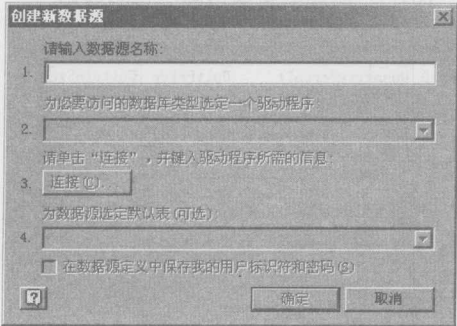
可以通过新建数据库查询法将SQL Server数据导入到Excel工作表，也就是通过建立一个数据库查询，并通过执行这个查询，从而得到需要的数据。下面介绍具体步骤。

(1) 在Excel工作簿中，单击【数据】→【导入外部数据】→【新建数据库查询】，打开【选择数据源】对话框，如图1-1所示。



(2) 在【选择数据源】对话框中，选择“数据库”选项卡列表中的“<新数据源>”，单击【确定】按钮，打开【创建新数据源】对话框，如图1-2所示。

(3) 在【创建新数据源】对话框中，在第1项“请输入数据源名称”文本框中输入要创建的数据源名称（例如输入“我的数据”），然后在出现的第2项“为您要访问的数据库类型选定一个驱动程序”下拉列表框中选择“SQL Server”，如图1-3所示。



(4) 单击【连接】按钮，打开【SQL Server登录】对话框，如图1-4所示。

(5) 首先在“服务器”列表框中选择或输入SQL Server服务器名称。如果该服务器有登录ID和密码，则取消“使用信任连接”复选框，然后在“登录ID”文本框和“密码”文本框中输入用户名和密码。单击【选项】按钮，展开对话框的“选项”部分，在“数据库”下拉列表框中选择要导出数据的数据库（例如选择SQL Server默认安装的示例数据库“Northwind”），在“语言”下拉列表框中选择“Simplified Chinese”，其他保持不变，如图1-5所示。

(6) 单击【确定】按钮，返回到【创建新数据源】对话框，在“为数据源选定默认表（可选）”复合框中选择要导出数据的数据表，如图1-6所示。

(7) 单击【确定】按钮，返回到【选择数据源】对话框，可以看到已经创建了一个名字为“我的数据”的数据源，如图1-7所示。

(8) 双击【选择数据源】对话框中刚刚创建的新数据源“我的数据”，就会打开查询向导的各个对话框，再按照向导的步骤进行操作，就可以将SQL Server数据库中需要的数据导入到Excel工作表。如图1-8所示为一个示例。

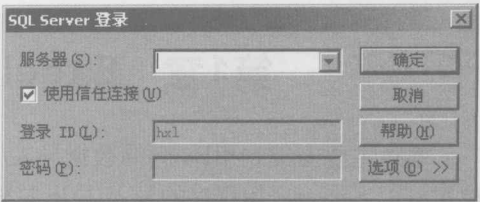


图1-4 【SQL Server登录】对话框

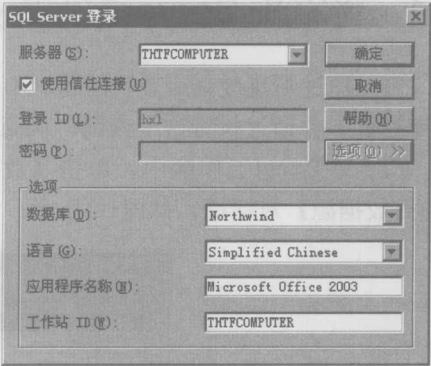


图1-5 输入SQL Server服务器名称，并设置选项

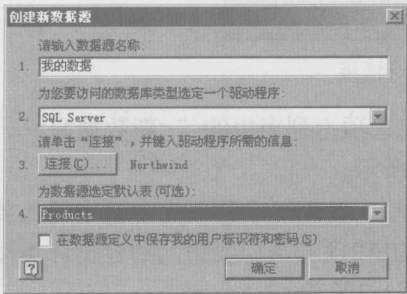


图1-6 选择要导出数据的数据表

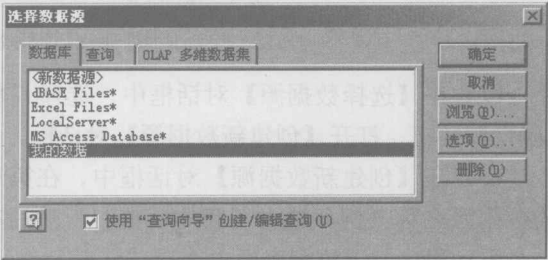


图1-7 创建的新数据源“我的数据”

	A	B	C	D	E	F	G
1	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
2	1	Chai	1	1	10 boxes x 20 bags	18	
3	2	Chang	1	1	24 - 12 oz bottles	19	
4	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	
5	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22	
6	5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35	
7	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25	
8	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30	
9	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40	
10	9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	
11	10	Ikura	4	8	12 - 200 ml jars	31	
12	11	Queso Cabrales	5	4	1 kg pkgs.	21	
13	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38	
14	13	Konbu	6	8	2 kg box	6	
15	14	Tofu	6	7	40 - 100 g pkgs.	23.25	
16	15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	
17	16	Paulownia	7	3	32 - 500 g boxes	17.45	

图1-8 将SQL Server数据库数据导入到Excel工作表的示例

1.2 将Excel工作表数据导入到SQL Server数据库

SQL Server数据库的应用越来越广泛。实现Excel工作表与SQL Server数据库的数据交互，将Excel工作表数据导入SQL Server数据库，可以充分利用SQL Server数据库在数据查询等方面的优势，提高数据管理与分析处理效率。

将Excel工作表数据导入到SQL Server数据库，可以使用SQL Server的导入/导出向导。具体步骤如下：

(1) 单击Windows的【开始】→【程序】→【Microsoft SQL Server】→【导入和导出数据】命令，打开【DTS导入/导出向导】对话框，如图1-9所示。

(2) 单击【下一步】按钮，打开【选择数据源】对话框，如图1-10所示。

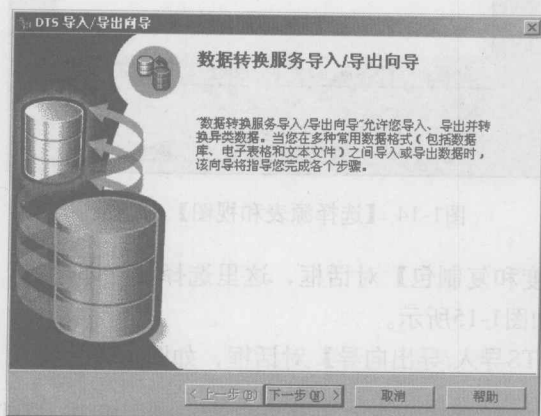


图1-9 【DTS导入/导出向导】对话框

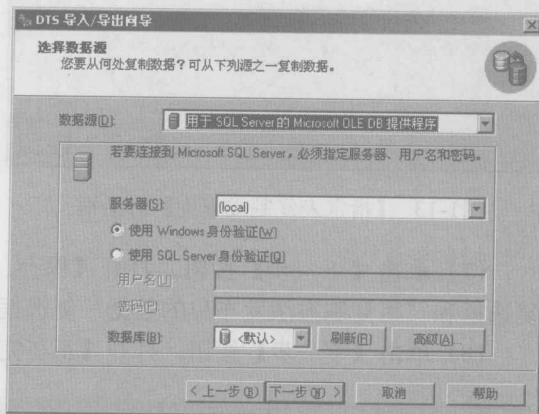


图1-10 选择数据源

(3) 由于要将Excel工作表数据导入到SQL Server数据库，因此在“数据源”下拉列表中选择“Microsoft Excel 97-2000”，然后在“文件名”文本框中输入Excel文件名（也可以通过右边的按钮  查找并选择Excel文件），如图1-11所示。

(4) 单击【下一步】按钮，打开【选择目的】对话框，在“目的”下拉列表中选择“用于SQL Server的Microsoft OLE DB提供程序”，在“服务器”下拉列表中选择SQL Server服务器名称，在“数据库”下拉列表中选择要保存数据的数据库名，如图1-12所示。

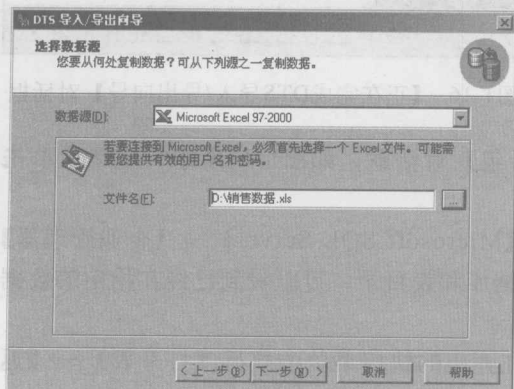


图1-11 选择数据源

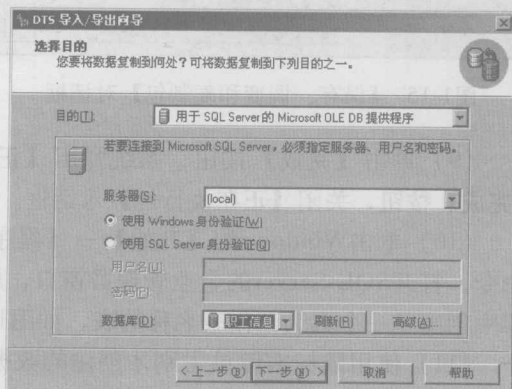


图1-12 选择目的

(5) 单击【下一步】按钮，打开【指定表复制或查询】对话框，这里选择“从源数据库复制表和视图”单选按钮，如图1-13所示。

(6) 单击【下一步】按钮，打开【选择源表和视图】对话框，选择要复制数据的工作表，如图1-14所示。

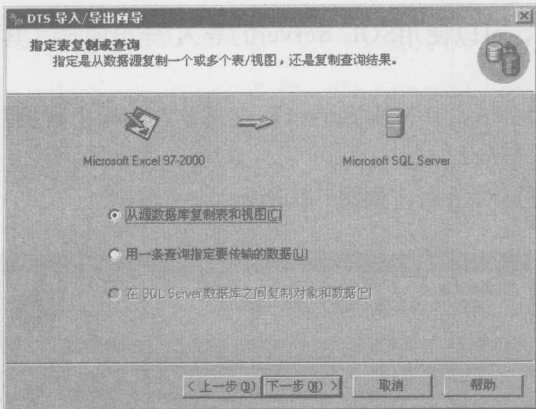


图1-13 【指定表复制或查询】对话框

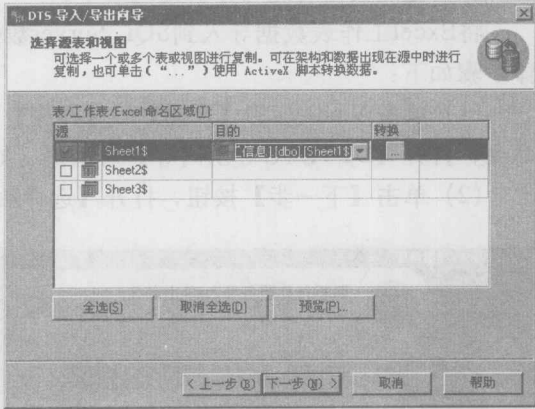


图1-14 【选择源表和视图】对话框

(7) 单击【下一步】按钮，打开【保存、调度和复制包】对话框，这里选择“立即运行”复选框和“用复制方法发布目的数据”复选框，如图1-15所示。

(8) 单击【下一步】按钮，打开【正在完成DTS导入/导出向导】对话框，如图1-16所示。

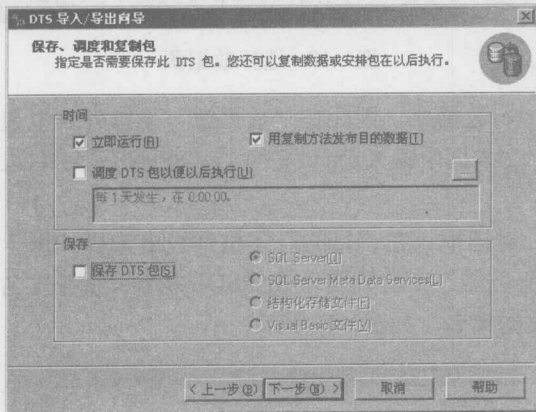


图1-15 【保存、调度和复制包】对话框



图1-16 【正在完成DTS导入/导出向导】对话框

(9) 单击【完成】按钮，系统弹出【正在执行包】对话框，如图1-17所示，完毕后单击【完成】按钮，关闭【正在执行包】对话框。

(10) 单击Windows的【开始】→【程序】→【Microsoft SQL Server】→【企业管理器】命令，打开SQL Server的企业管理器窗口，展开数据库和表目录，可以看到已经在指定的数据库中创建了一个名为工作表名称的表，如图1-18所示。

(11) 使用鼠标右键单击刚才创建的数据表，在弹出的快捷菜单中执行【打开表】→【返回所有行】命令，打开表数据对话框，可以查看导入的Excel工作表数据，如图1-19所示。

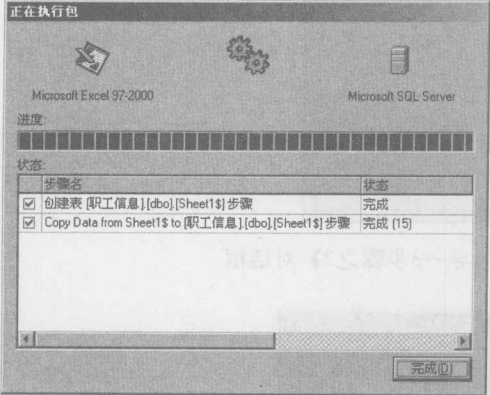


图1-17 【正在执行包】对话框

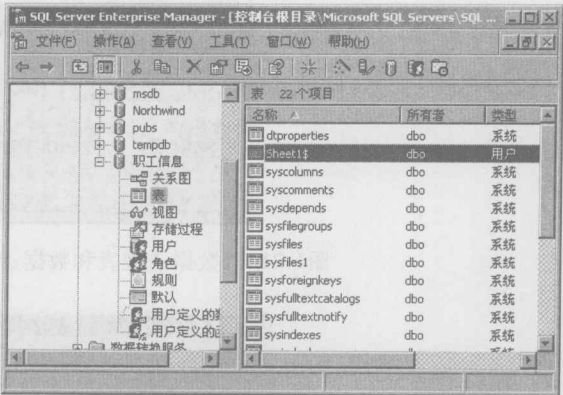


图1-18 查看创建的数据表

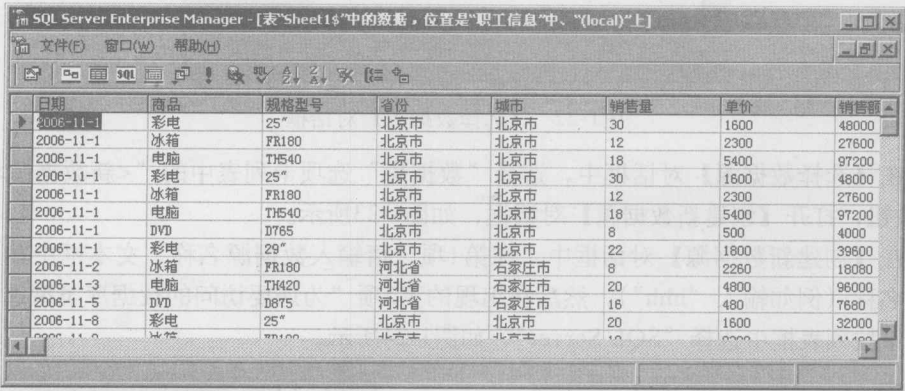


图1-19 导入到SQL Server数据库的Excel工作表数据

1.3 以SQL Server数据库数据创建数据透视表

您对以Excel工作表数据为基础创建数据透视表想必已经不陌生了。这里介绍如何以SQL Server数据库数据为基础创建数据透视表。具体步骤介绍如下。

(1) 单击【数据】→【数据透视表和数据透视图】命令，打开【数据透视表和数据透视图向导—3步骤之1】对话框，如图1-20所示。

(2) 在“请指定待分析数据的数据源类型”项目中选择“外部数据源”单选按钮，单击【下一步】按钮，打开【数据透视表和数据透视图向导—3步骤之2】对话框，如图1-21所示。

(3) 单击【获取数据】按钮，打开【选择数据源】对话框，如图1-22所示。

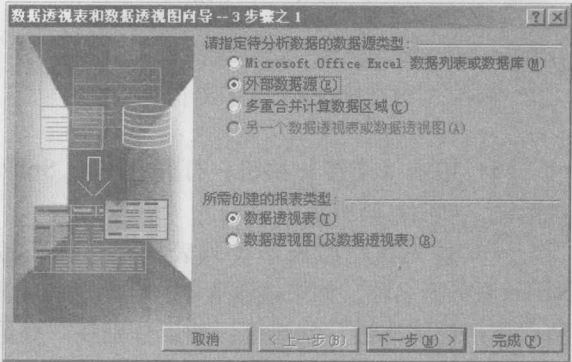


图1-20 【数据透视表和数据透视图向导—3步骤之1】对话框

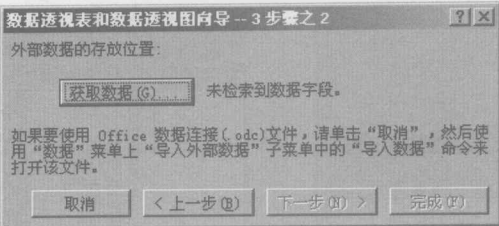


图1-21 【数据透视表和数据透视图向导—3步骤之2】对话框

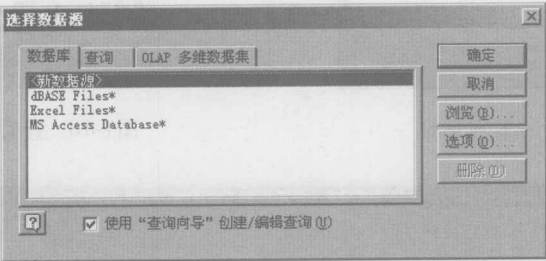


图1-22 【选择数据源】对话框

(4) 在【选择数据源】对话框中，选择“数据库”选项卡列表中的“<新数据源>”，单击【确定】按钮，打开【创建新数据源】对话框，如图1-23所示。

(5) 在【创建新数据源】对话框中，在第1项“请输入数据源名称”文本框中输入要创建的数据源名称（例如输入“hhh”），然后在出现的第2项“为您要访问的数据库类型选定一个驱动程序”下拉列表框中选择“SQL Server”，如图1-24所示。

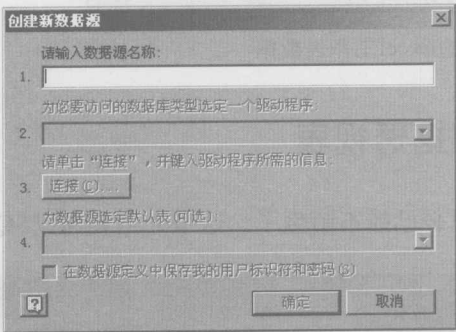


图1-23 【创建新数据源】对话框

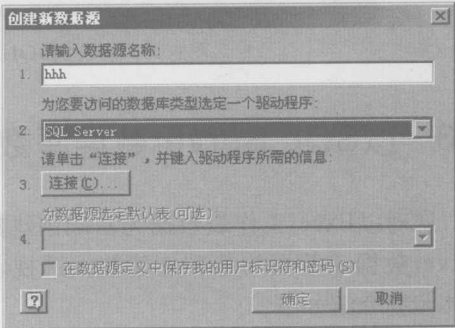


图1-24 输入数据源名称，并选择驱动程序

(6) 单击【连接】按钮，打开【SQL Server登录】对话框，如图1-25所示。

(7) 首先在“服务器”列表框中选择或输入SQL Server服务器名称。如果该服务器有登录ID和密码，则取消“使用信任连接”复选框，然后分别在“登录ID”文本框和“密码”文本框中输入用户名和密码。单击【选项】按钮，展开窗体的“选项”部分，在“数据库”下拉列表框中选择要导出数据的数据源（例如选择SQL Server默认安装的示例数据库“Northwind”），

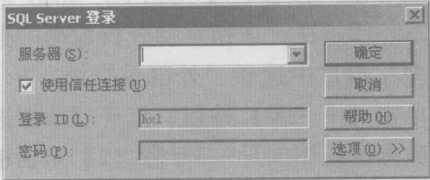


图1-25 【SQL Server登录】对话框

在“语言”下拉列表框中选择“Simplified Chinese”，其他保持不变，如图1-26所示。

(8) 单击【确定】按钮，返回到【创建新数据源】对话框，如图1-27所示。

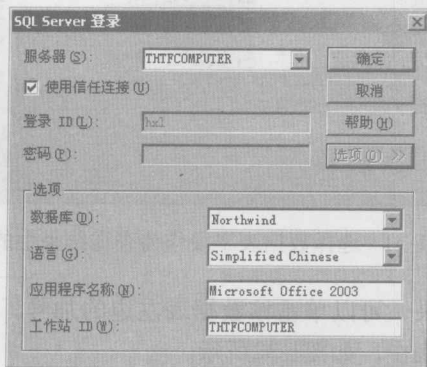


图1-26 输入SQL Server服务器名称，并设置选项

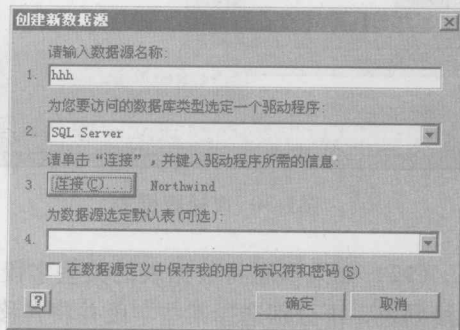


图1-27 与SQL Server数据库连接成功

(9) 单击【确定】按钮，返回到【选择数据源】对话框，可以看到已经创建了一个名字为“hhh”的数据源，如图1-28所示。

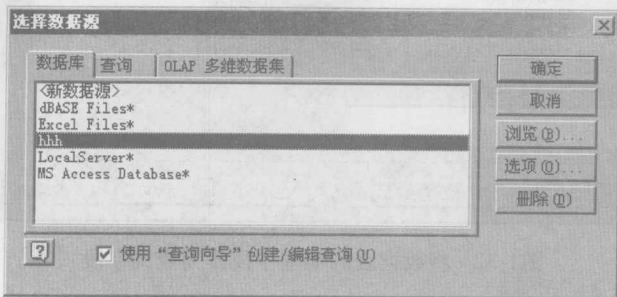


图1-28 创建的新数据源“hhh”

(10) 单击【确定】按钮，打开【查询向导-选择列】对话框，如图1-29所示。

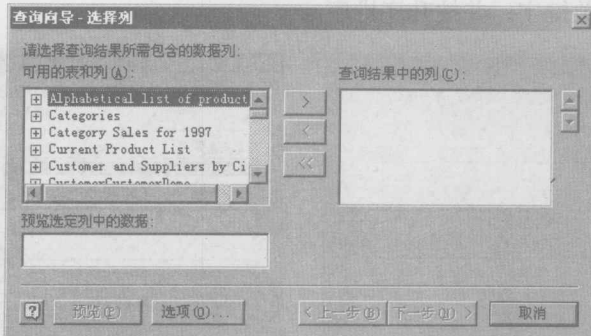


图1-29 【查询向导-选择列】对话框

(11) 在“可用的表和列”列表中选择要创建数据透视表的SQL Server数据表（例如“Invoices”），如图1-30所示，然后单击 > 按钮，将该数据表的所有字段作为查询字段移到右边的列表中，如图1-31所示。