



数
据
库
开
发

COOL-
SQL Server 7.0 +
PowerBuilder 7.0

计算机图书公司出版



数据库 开发

COOL-

SQL Server 7.0 +
PowerBuilder 7.0



北京希望电子出版社





希望电脑 COOL 系列

数据库开发 COOL

SQL Server 7.0+PowerBuilder 7.0

北京希望电脑公司 总策划
蒋长浩 温志华 史树明 主编

北京希望电子出版社

www.bhp.com.cn

内 容 简 介

本书是“希望电脑 COOL 系列”图书之一。全书由两篇构成，分别介绍了两个相关软件的应用。第一篇以 6 章的篇幅讲解了典型的客户/服务器结构的数据库管理系统 SQL Server 最新版本 7.0 的应用，分别介绍了进入 SQL Sever 所进行的操作、设置、SQL Sever 管理、数据库操作和 SQL Sever 数据操作，数据的复制和转换等内容。第二篇分 9 章，讲解了 PowerBuilder 7.0 的强大功能和应用。内容包括：数据库管理与操作，创建和设置应用程序对象、窗口对象、数据窗口的对象，数据控制和 PowerScript 编程，创建和使用用户对象和事件，生成应用程序等。

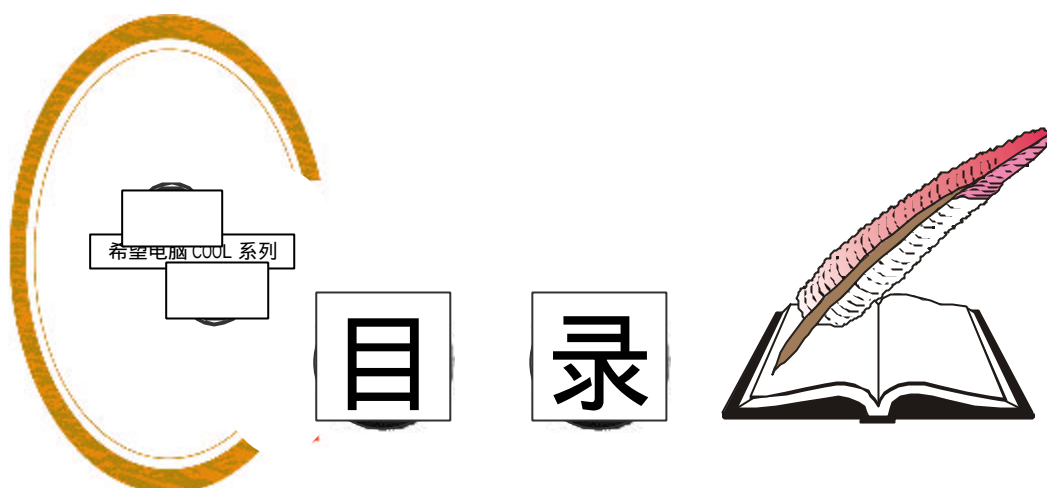
书中通过大量实例讲述了 SQL Server 7.0 和 PowerBuilder 7.0 的使用技巧。全书把功能和实现的知识点与具体的实例紧密地联系在一起，由浅入深地指导读者学习并掌握两种软件的应用。

本书可供编程爱好者及初步了解 SQL Sever 和 PowerBuilder 准备深入开发编程的读者参考、使用。

本书所附光盘含与本书配套的电子图书，并赠送“FoxPro 通用程序设计实战训练”多媒体学习软件。

系 列 书	希望电脑 COOL 系列
书 名	数据库开发 COOL (SQL Server 7.0+PowerBuilder 7.0)
总 策 划	北京希望电脑公司
文 本 著 作 者	蒋长浩 温志华 史树明 主编
审校/责任编辑	周 艳 刘晓融
C D 制 作 者	希望多媒体创作中心
C D 测 试 者	希望多媒体测试中心
出版/发 行 者	北京希望电脑公司 北京希望电子出版社
地 址	北京海淀路 82 号 (100080) 网址：www.bhp.com.cn E-mail：lwm@hope.com.cn 电话： 010-62562329，62541992，62637101，62637102 (图书发行，技术支持) 010-62633308，62633309 (多媒体发行，技术支持) 010-62613322-215 (门市) 010-62531267 (编辑部)
经 销	各地新华书店、软件连锁店
排 版	门槛创作室 WORD 照排部
C D 生 产 者	文录激光科技有限公司
文 本 印 刷 者	北京广益印刷厂
规 格 / 开 本	787 毫米×1092 毫米 16 开本 31.5 印张 700 千字
版 次 / 印 次	1999 年 7 月第 1 版 1999 年 7 月第 1 次印刷
印 数	0001—6000 册
本 版 号	ISBN 7-900024-23-9/TP·23
定 价	39.00 元 (1CD，含配套书)

说明：凡我社光盘配套图书若有自然破损、缺页、倒页、脱页者，本社发行部负责调换。



SQL Server 7.0 篇

- 第一章 进入 SQL Server
- 第二章 管理 SQL Server
- 第三章 数据库操作
- 第四章 SQL Server 数据操作
- 第五章 数据复制
- 第六章 数据转换

PowerBuilder 7.0 篇

- 第一章 数据库管理与操作
- 第二章 应用程序对象
- 第三章 窗口对象
- 第四章 数据窗口对象
- 第五章 使用数据窗口对象
- 第六章 PowerScript 编程
- 第七章 用户对象和用户事件
- 第八章 PBL 库
- 第九章 生成应用程序



[返回总目录](#)

PowerBuilder 7.0篇

第一章

数据库管理与操作

本章导读

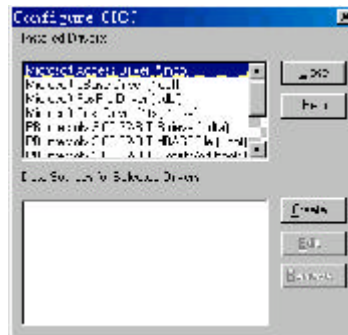
本章的内容分两个部分，第一部分讲述如何通过 ODBC 连接 Power Builder 与数据库；第二部分讲述如何创建、访问和维护后端数据库。

有两个画板(Configure ODBC 和 DB Profile 画板) 与数据库连接有关，本章通过实例讲述了在这两个画板中连接数据库和建立数据库描述文件的操作步骤。Database 画板是后端数据库管理的有力工具。本章详细讲述了利用 Database 所进行的如下操作：打开、创建和删除表、为表设置主码、外码和索引、建立视图、表的数据操作（包括数据的添加、删除、排序、过滤、导入、导出）、列的显示格式与编辑格式、列的有效性规则。

本章的最后是两个具体的例子。

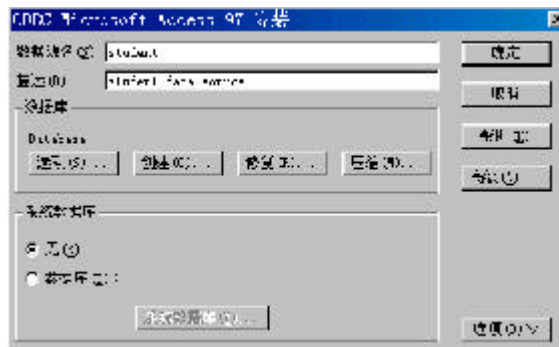
通过 ODBC 接口创建数据源 Access 和 SQL Anywhere 的例子

1



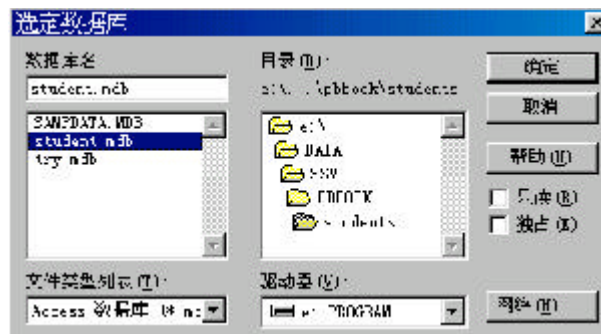
2

在 Installed Drivers 列表框中选择 Microsoft Access Drive(*.mdb), 按 Create 按钮, 出现如下对话框。在“数据源名”文本框内输入数据源的名字(如 student), 并在“描述”文本框输入数据源的描述(不是必需的)。



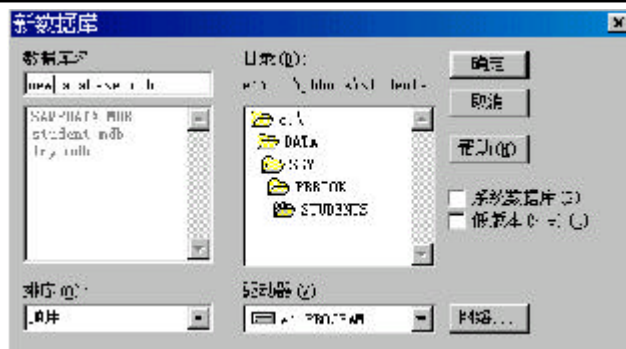
3

按“选取”按钮, 弹出如下对话框。从对话框选取一个用 Microsoft Access 预先建好的数据库文件 student.mdb (如果不方便创建 student.mdb, 跳到步骤 5)。按“确定”或“Close”, 回到主窗口, 完成数据源的创建。



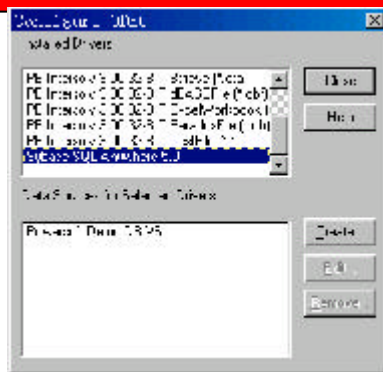
4

知识



5

还可以创建其它接口的数据源（比如 Power Builder 自带的 Sybase SQL Anywhere），观察下图的 Configure ODBC 对话框和步骤 1 有什么不同。



6

上图是在 Configure ODBC 对话框的 Installed Drives 列表框中选择了 Sybase SQL Anywhere 5.0 的结果。注意到 Data Sources 列表框中有: Powersoft Demo DB V6，这是 Power Builder 自带的数据源。单击 Create，得到如下对话框。

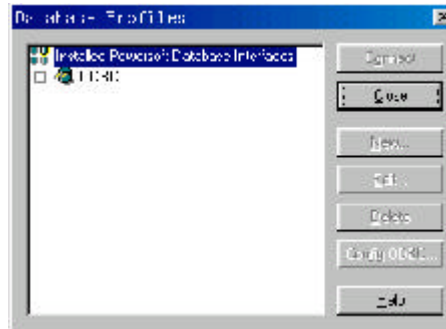


7

利用此对话框可以新建 Sybase SQL Anywhere 数据源（具体操作在本章的其它节会讲到）。按 Cancel 按钮返回步骤 5 对话框，再按 "Close" 返回主窗口，此次操作完成。

ODBC 接口数据库描述文件
Microsoft Access 数据源的例子

1



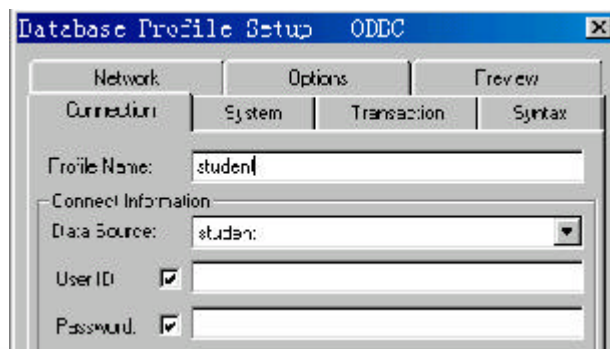
2

点击 ODBC 前的加号或者双击 ODBC，显示已经创建的数据源描述文件列表。如果某数据源已经建立，但列表中没有对应的描述文件，则选中 ODBC，单击 New 按钮，弹出 如下对话框（否则跳到第 5 步）。



3

在上图对话框中选中 Connection 标签，在 Profile Name 文本框中输入描述文件的名字（与数据源同名即可），在 Data Source 组合框中选取数据源的名字（User ID 和 Password 可不填）。完成后的对话框的上半部分如下图。

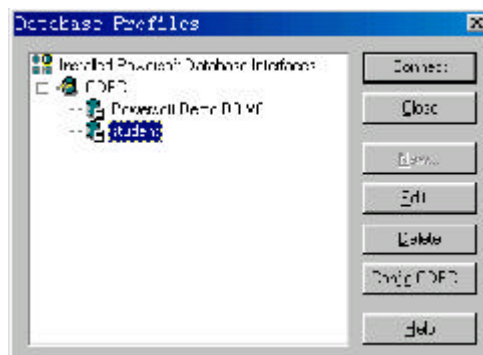


4



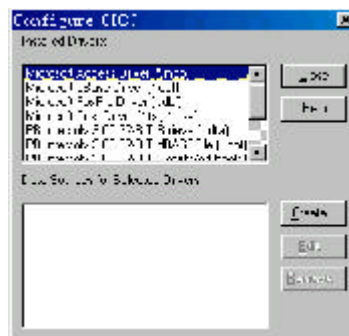
5

单击 OK，关闭上图 Database Profile Setup 对话框，回到 Database Profiles 对话框，见下图。选中一个数据描述（而不是选中 ODBC），按 Edit 按钮得到和步骤 2、3、4 一样的对话框，可以编辑数据描述文件。



6

单击 Configure ODBC 按钮，弹出创建数据源的对话框（和上一节步骤 1 中的对话框一样），见下图。这样就可以重复上一讲的步骤建立新的数据源。

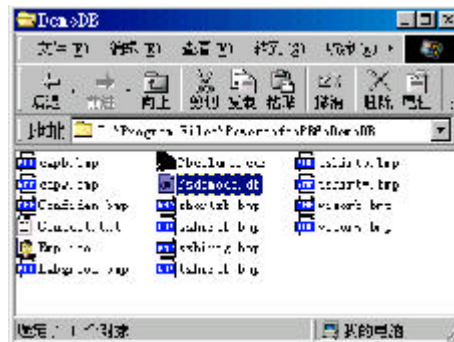


7

在步骤 5 中，双击 ODBC 下的某一个数据描述或者选中数据描述后按 Connect 按钮，就实现了 Power Builder 和数据库的连接。所选择的数据库成为 Power Builder 的当前数据库。

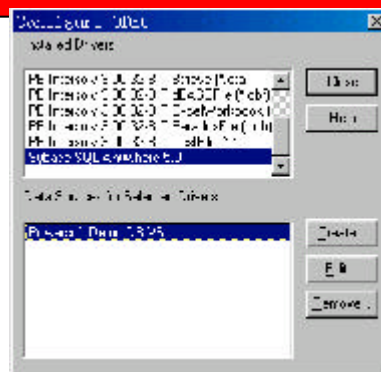
“创建”本地数据库 创建 MyDemoDB

1



2

单击 PowerBar 工具条上的 Configure ODBC 图标按钮，显示下图对话框。
在 Installed Drivers 列表框中选择 Sybase SQL Anywhere 5.0。

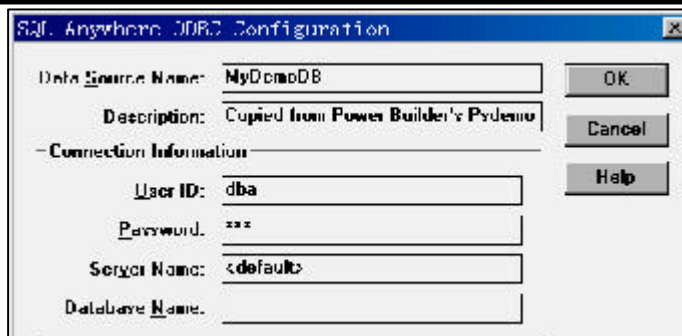


3

单击 Create 按钮，得到如下 Sybase SQL Anywhere 数据库 ODBC 接口的设置对话框。可以按 Help 按钮查看 SQL Anywhere 的设置帮助。

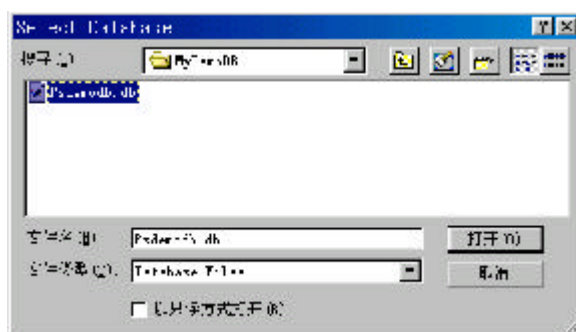


4



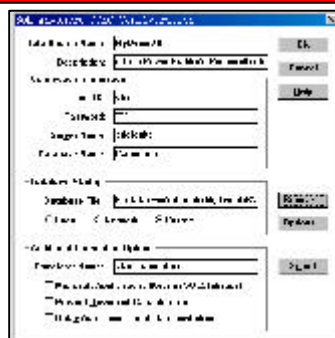
5

单击 Browse 按钮，弹出下图对话框。选择步骤 1 中新建的目录（拷贝过来的 Psdemedb.db 就在此目录下），选中 Psdemedb.db。



6

按“打开”按钮，完成的 SQL Anywhere ODBC Configuration 对话框如下图所示。注意到 Database File 文本框中是刚才所选择的路径。Database Name 文本框自动被填充。

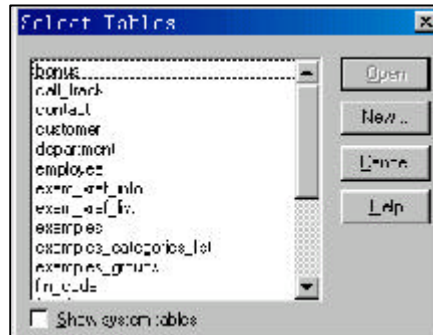


7

按 OK 按钮，回到 Configure ODBC 对话框，再按 Close 按钮关闭之。按照第 2 节所讲的操作步骤连接 MyDemoDB 为当前数据库。至此操作完成。

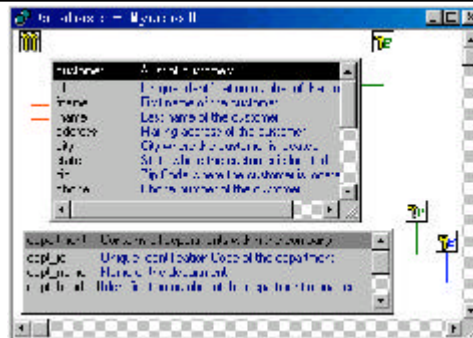
打开、创建和删除表 MyDemoDB 的表操作

1



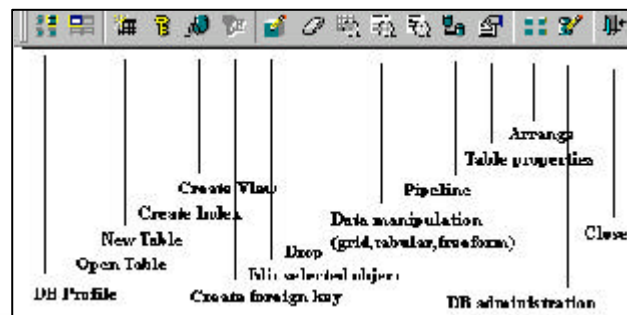
2

在上图对话框的列表框中选择一个或多个表，按 Open 按钮，则在 Database 窗口中显示出了所打开表的名称、各个字段名和索引、主键等（注意图中 maintable 引向钥匙的红、蓝、绿等各种颜色的线。

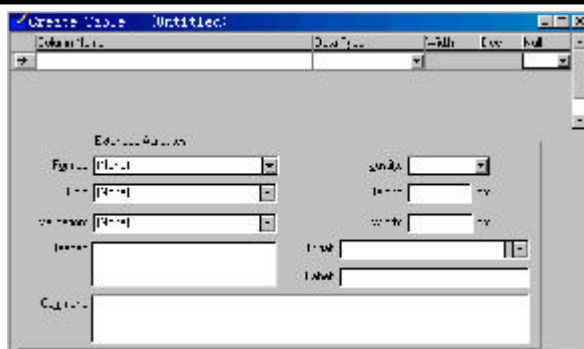


3

这时，PowerBar 的下方新出现了一个工具条，见下图。它叫 Database Painter Bar（数据库画板工具条）。单击其按钮可以完成连接数据库（DB Profile）、打开表（Open Table）、建立索引（Create Index）等操作。

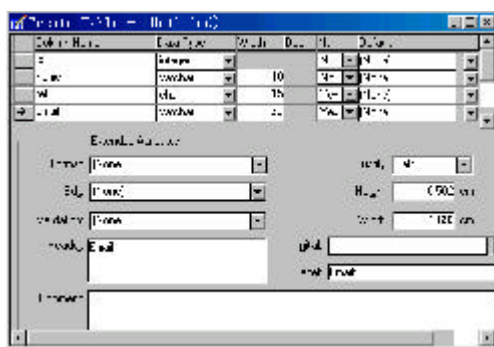


4



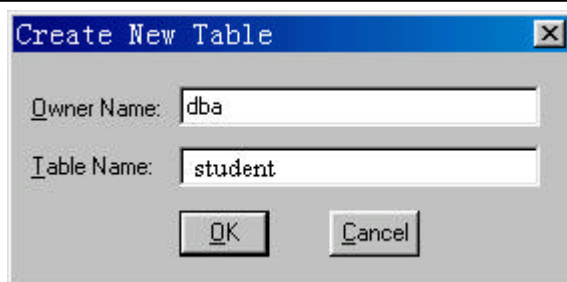
5

可以在 NULL 中指定此字段是否可以为空。按几次键盘上的 Tab 键或单击表画板（见知识点解释）上的 Insert Column 按钮可以添加新的字段。随意添加几个字段，得下图。



6

单击表画板上的 Save 按钮（一个软盘的形状），弹出下图 Create New Table 对话框。输入表的 Owner Name 和 Table Name，按“确定”将表保存在当前数据库中，这样就新建了一个表。



7

退出表画板（单击其上的 Close 按钮），回到数据库画板。选中一个表，单击数据库画板上的 Drop 按钮，可以删除它（不要删 student，以后要用）。在显示表信息的小窗口的 Frame 上按鼠标右键选 Close 可以关闭表。

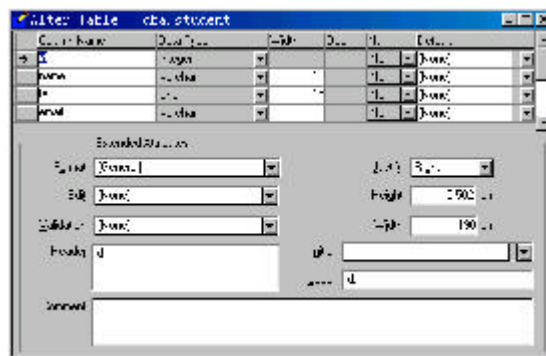
表的字段属性
修改表 student

1



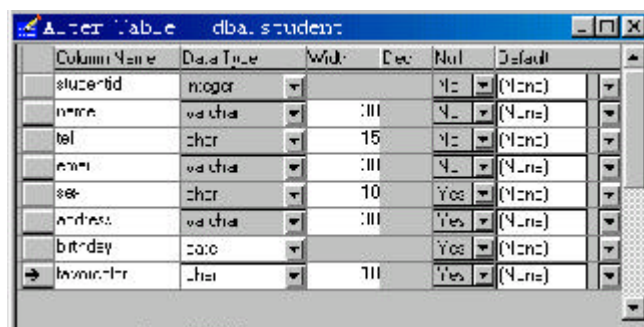
2

双击 Database 画板主窗口中的 student 表窗口，或选中 student 表窗口再按数据库画板上的 Edit Selected Object 图表按钮，进入表画板，如下图。画板主窗口分为上下两个部分，分别为列属性和扩展属性。

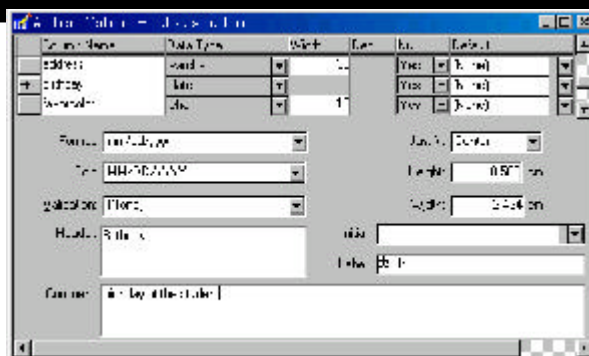


3

注意到列的 Data Type 和 NULL 两个属性变为灰色，这是因为 Watcom/Sybase SQL Anywhere 对于修改数据库有限制（见旁边解释）。将列名 id 改为 studentid，name 列的长度增至 30，再增加几个字段，修改结果见下图。

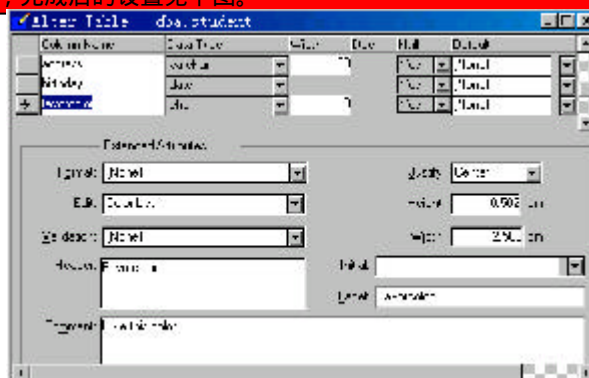


4



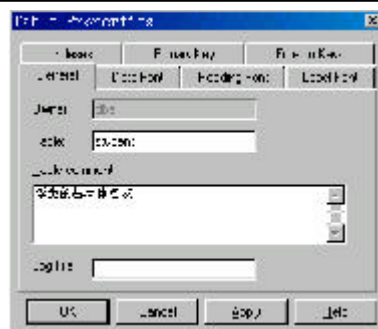
5

选中 favorcolor 列（选中后其左侧有一个箭头），在列扩展属性中选择 Edit 为 Color List, Justify 选择 Center, Width 设为 2.5, 在 Comment 中加入列注释，完成后的设置见下图。



6

单击工具条（注意现在在表画板状态）上的 Properties 图表按钮，弹出表的属性对话框，在其 General 标签页下的 Table comment 编辑框中输入表的注释，如图。然后按 OK 确定。

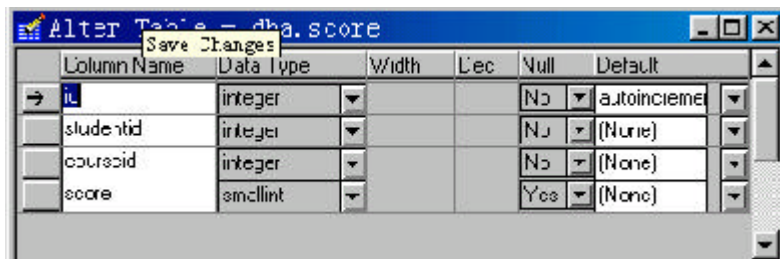


7

单击表画板工具条上的 Save 按钮，保存表；按 Close 图表按钮退出表画板，在数据库主窗口中看到改变了的 student 表。关闭表画板，完成本次操作。

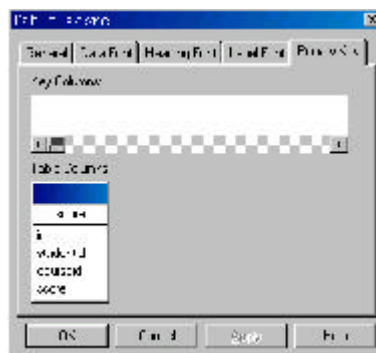
表的主码与外码 设置 score 的主码和外码

1



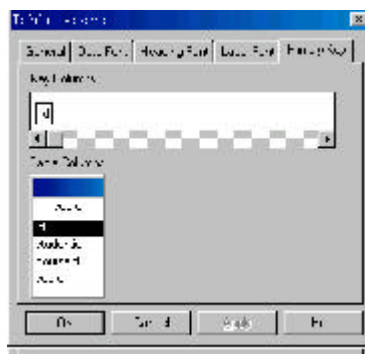
2

在表 score 的设计窗口中单击工具条上的 Properties 按钮或在数据库画板中打开表 score，然后单击 Table Properties 按钮，得 Table score 对话框。单击其 Primary Key 标签页，见下图。

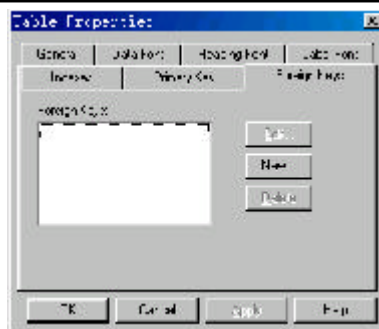


3

在 Table Columns 中选择合适的字段名（本例是“id”），单击之，在 Key Columns 中出现了所选的列名，表示将此列作为主码。再次单击已经选中的主码则取消之，可以选择多于一个主码，按 OK 完成设置。

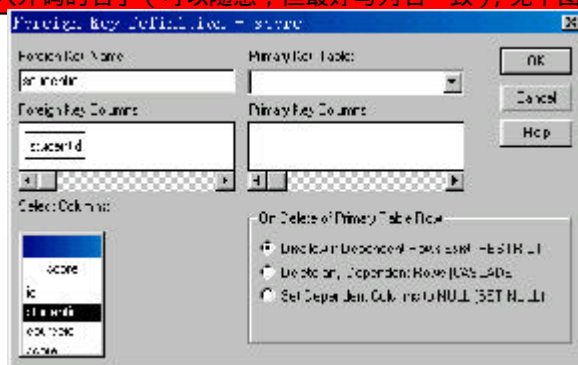


4



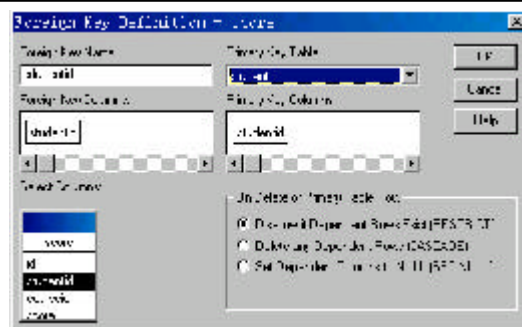
5

单击 **New** 按钮，弹出外码定义对话框，见下图。在对话框左下的 **Select Columns** 中选择要定义外码的字段名（本例为 **studentid**），再在左上的编辑框中输入外码的名字（可以随意，但最好与列名一致），见下图。



6

在右上的 **Primary Key Table** 组合框中选择外码相联系的表（本例为 **student**），在其下的 **Primary Key Columns** 中自动出现表 **student** 的主码。右下的单选框保持默认设置。完成的对话框见下图。

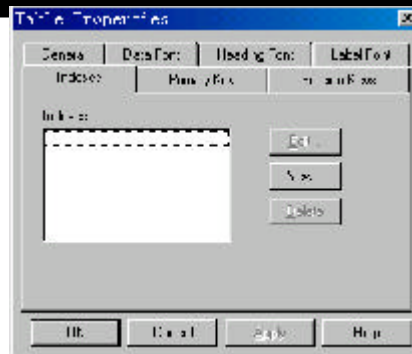


7

按 **OK** 退到步骤 4 对话框（注意到 **Edit** 按钮变为可用，单击之可以用同样的步骤编辑已经存在的外码；选中一个外码然后按 **Delete** 可以删除外码）。按 **OK** 按钮完成操作。

表的索引
为 score 建立索引

1



2

单击 New 按钮，弹出 Create Index 对话框，见图。注意到对话框左上角的“table: dba.score”表示为用户 dba 的表 score 建立索引。

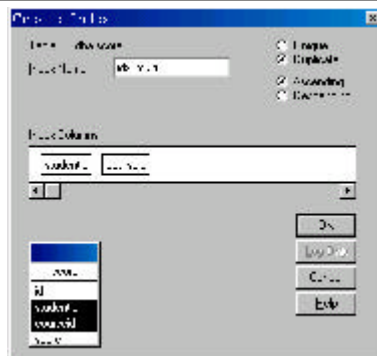


3

在“Index Name”文本框中输入索引的名字：idx_studentid；在对话框左下角选择为其建立索引的列：studentid；右上的两组单选框分别选择 Duplicate 和 Ascending。完成的对话框如下图所示。



4



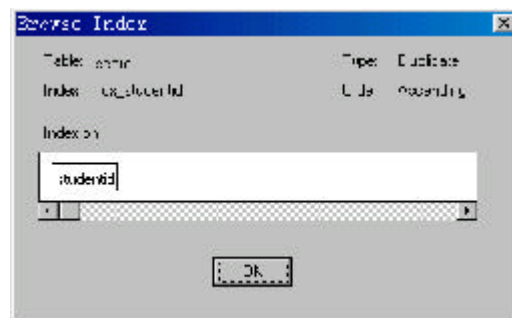
5

一路按 OK 退出各个对话框；保存并退出表的设计窗口；在数据库画板中打开（且只打开）表 student 和 score，见下图。从图中可以很清楚地看到表的索引、主码及表 score 对表 student 的外码。



6

在图中最左上的双钥匙上单击右键，在弹出的菜单上选择“Browse...”（注意不要选下面的“Drop Index”，除非你想删除此索引），弹出如下对话框。它用于浏览索引。

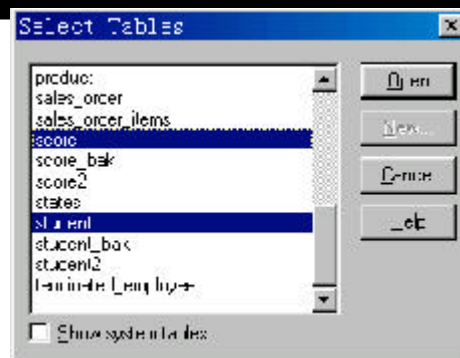


7

在步骤 5 的图中红线通向索引，绿线通向主码，蓝线通向外码。可以右键单击并选择“Drop”来删除索引、主码或外码。右键单击并选择别的菜单项或直接双击它们，看还可以完成什么功能。

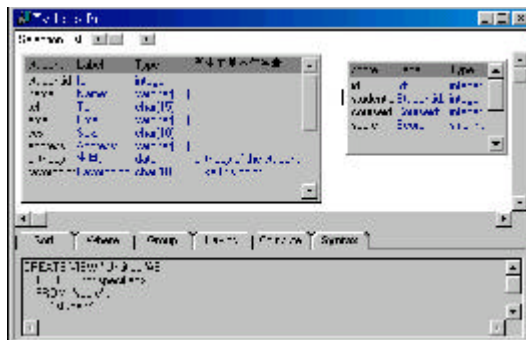
表的视图 创建某学生的成绩表视图

1



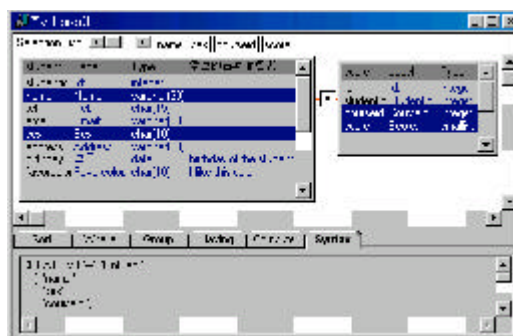
2

选择表 `score` 和 `student`，单击 `Open` 按钮进入视图创建画板，如图。注意两个表之间有一个等号，表示 `score` 中的字段 `studentid` 与 `student` 表的主码 `studentid` 的相等关系。

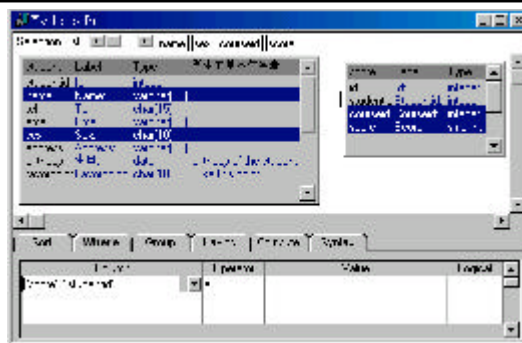


3

在表 `student` 和 `score` 上选择视图中要出现的列（即字段），此学生的成绩表中应该出现姓名、性别、课程号、成绩，故从两个表中选择了 4 项，如图。

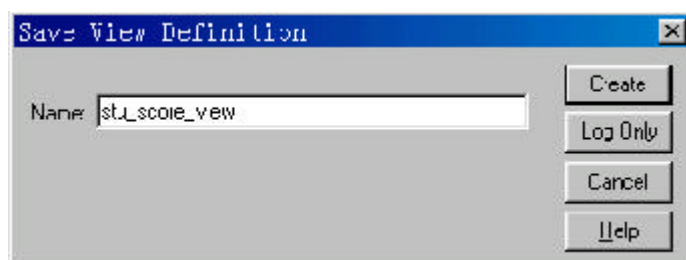


4



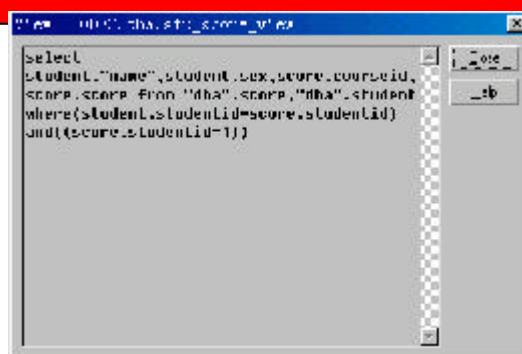
5

按视图画板的关闭按钮（注意不是 Power Builder 本身的关闭按钮），弹出下图对话框，提示保存。



6

输入要保存的视图名字，按 Create 按钮，回到数据库画板。双击刚建立的视图窗口，弹出如下对话框。这就是生成视图的 Select 语句。

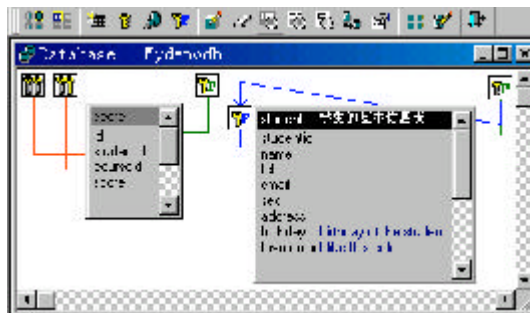


7

按 Close 按钮，关闭对话框；关闭数据库画板，完成本次操作。

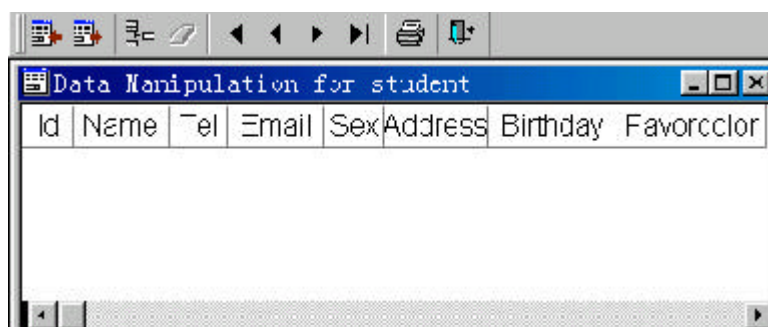
表的数据添加 往 score 中添加记录

1



2

选中 student 表窗口，然后单击 Painter Bar 上的“Data manipulation(Grid)”图标按钮，进入数据操作画板，其主窗口和工具条（Painter Bar）如下图。由于现在尚未在表中添加任何数据，故主窗口中只显示了表的字段名。



3

单击 Painter Bar 上的 Insert Row（添加行）按钮，主窗口中多了一个空行，在空行的表格中添加数据。添加了 3 行数据后的结果见图。照本例进行练习时，id 列照图填写，其它可随意。



4

知识

图标	图标名字	说明
	Refresh	刷新数据，将数据库中的数据更新到表中。
	Save Changes	保存数据，将数据库中的数据保存到表中。
	Insert Row	插入一行。
	Delete	删除一行。
	First, Previous, Next, Last	移动到第一行、上一行、下一行、最后一行。
	Print Data Window	打印数据窗口。
	Close	关闭 Data Manipulation 窗口。

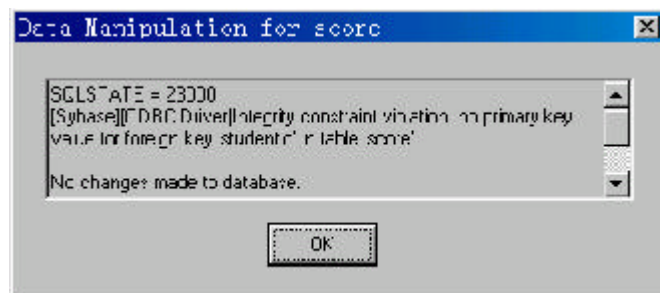
5

按照同样的方法给表 score 添加数据（练习时请严格照下图添加数据），如下图所示。注意 id 字段在前面已经被设置为自动添加数据，只需要在另外三列添加。

Id	Studentid	Courseid	Score
5	1	70	
2	2	86	
2	1	88	
1	2	90	
1	1	99	

6

按 Save 按钮保存修改，出现如下对话框。它提示 score 表的 studentid 外码在 student 表中没有对应的主码值，原因是第一行数据 studentid 为“5”，它在 student 表的主码 id 中不存在这个值。

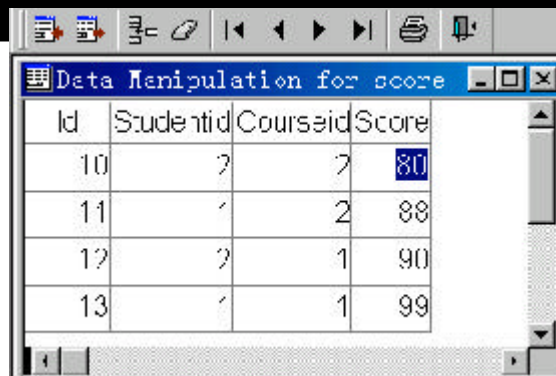


7

按 Clear 按钮删除 score 表的第一行数据，再按 Save 按钮保存即可。

数据的排序
score 中数据的排序

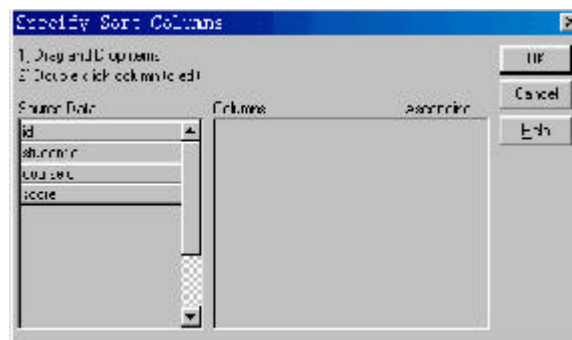
1



Id	Studentid	Courseid	Score
10	2	2	80
11	1	2	88
12	2	1	90
13	1	1	99

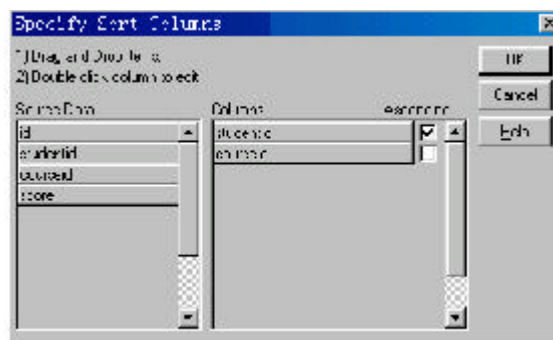
2

选择 Rows|sort 菜单命令，弹出如下对话框。这个对话框用来选择用来排序的列和排序方式（升序还是降序）。

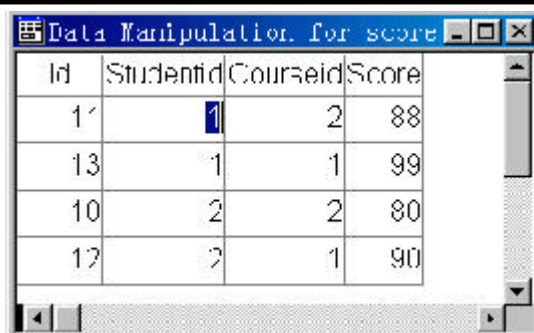


3

按照对话框左上角所描述的操作步骤，从左侧的“Source Data”中选择要排序的列名，将其拖动到右面的“Columns”列表框中，并选择是否按照升序进行排序。完成后的对话框见下图。



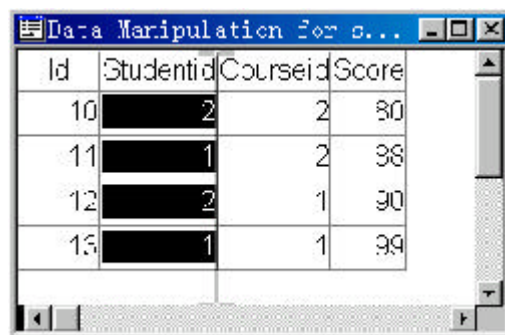
4



Id	Studentid	Courseid	Score
1	1	2	88
13	1	1	99
10	2	2	80
12	2	1	90

5

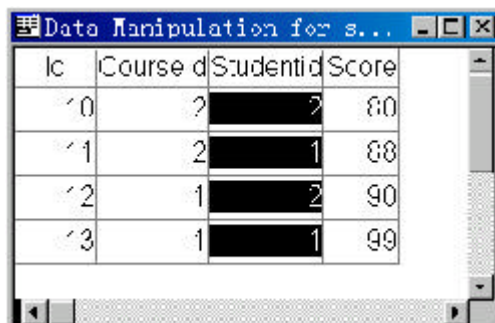
各列的顺序也是可以改变的，比如想让 Studentid 字段在 Courseid 字段之后，可以这样做：在 Studentid 列名上按下（而不是单击）鼠标，见下图。发现本列变黑，并且出现一个纵穿整个窗口的“I”状的粗线。



Id	Studentid	Courseid	Score
10	2	2	80
11	1	2	98
12	2	1	90
13	1	1	99

6

保持鼠标按下的状态向右拖动鼠标直到那个“I”状的粗线挪到 Courseid 列的后面，放开鼠标，见下图。这时可以发现 studentid 列已经到了 Courseid 列的后面。



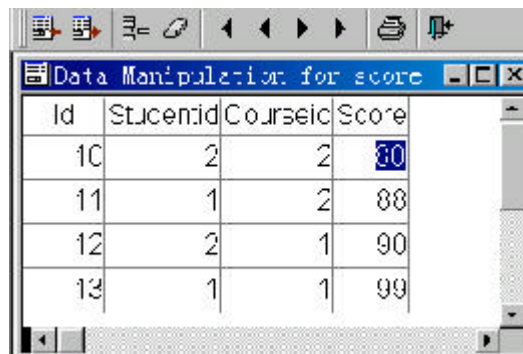
Id	Courseid	Studentid	Score
10	2	2	80
11	2	1	98
12	1	2	90
13	1	1	99

7

关闭画板，完成操作。

数据汇总和过滤
score 数据的汇总和过滤

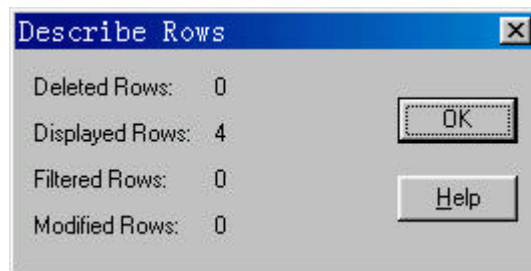
1



Id	Studentid	Courseid	Score
10	2	2	80
11	1	2	88
12	2	1	90
13	1	1	99

2

选择 Rows|Describe...菜单命令，弹出如下“Describe Rows”对话框。对话框显示了表中行的信息，如显示的行、删除的行、改变的行、过滤的行。

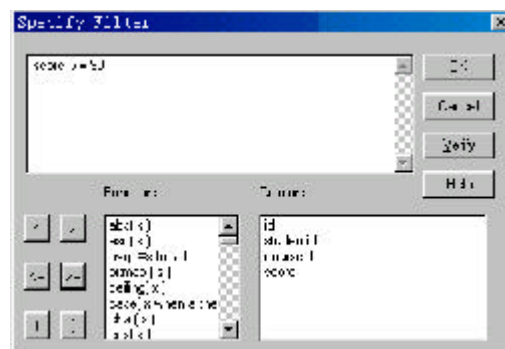


Deleted Rows:	0
Displayed Rows:	4
Filtered Rows:	0
Modified Rows:	0

OK Help

3

关闭上图对话框，选择 Rows|Filter...菜单命令，弹出下图对话框。可以在左上的文本框中输入过滤条件，使表只显示满足过滤条件的记录（数据）。本例输入条件“score >= 90”。



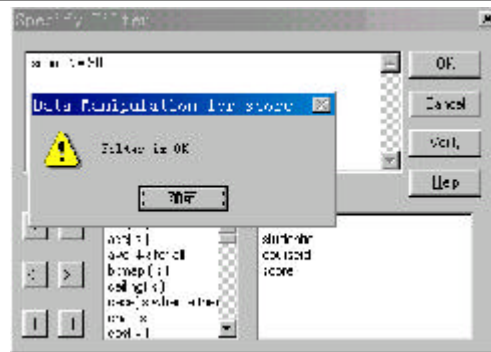
score >= 90

Functions: $abs()$, $ceil()$, $exp()$, $floor()$, $log()$, $log10()$, $round()$, $sign()$, $sqrt()$, $trunc()$, $var()$, $year()$

Columns: id, studentid, courseid, score

OK Cancel Apply Help

4



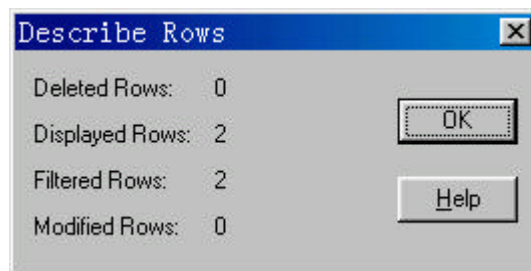
5

一直按“OK”关闭所有对话框，回到 Data Manipulation 主窗口。可以看到只显示了满足条件“score>=90”即分数在 90 分以上的记录。

Data Manipulation for score			
Id	Studentid	Courseid	Score
12	2	1	90
13	1	1	90

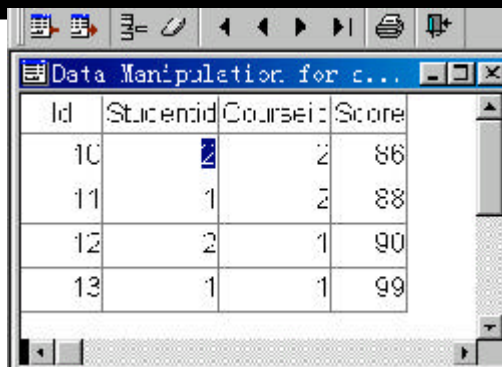
6

再用与步骤 2 相同的操作打开“Describe Rows”对话框，如下图所示。注意到“Filtered Rows”由 0 变为 2，“Displayed Rows”由 4 变为 2。



7

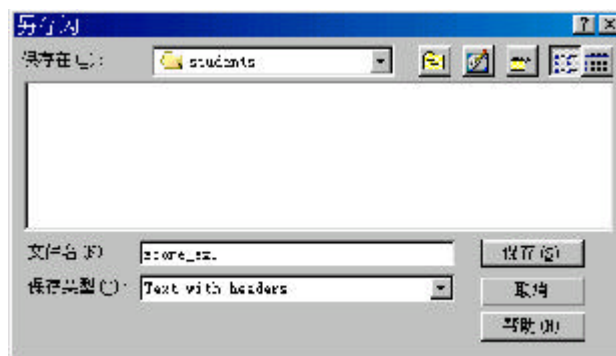
关闭对话框，退出数据操作画板，完成操作。读者也可以用 Data Manipulation 的另外两种方式（Freeform 和 Tabular）来重复以上的操作。

导入/导出数据
表 Score 数据的导入/导出**1**

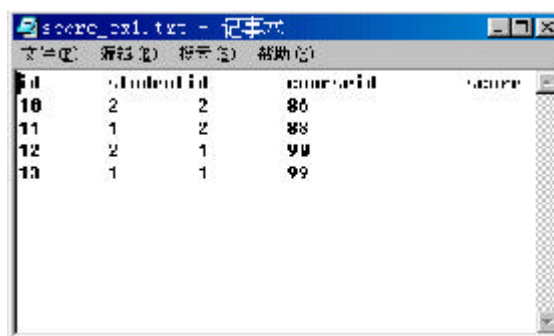
Id	Studentid	Courseid	Score
10	2	2	86
11	1	2	88
12	2	1	90
13	1	1	99

2

选择“File|Save Rows As...”菜单命令，弹出如下“另存为”对话框。在“文件名”编辑框中输入要导出的文件名 score_ex1，在“保存类型”下拉式列表框中选择“Text with headers”，如下图。

**3**

按“保存”按钮，则将表 score 的数据信息导出到文件 score_ex1.txt 中。用记事本打开此文件，如下图。导出的格式是带有头（即字段名）的文本文件，每条记录占一行，记录的各个字段间用 Tab 分隔。



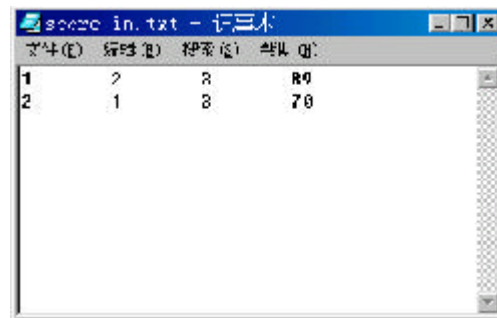
Id	Studentid	Courseid	Score
10	2	2	86
11	1	2	88
12	2	1	90
13	1	1	99

4

```
score_ext.sql - 记事本
CREATE TABLE score_ext (
  id float,
  studentid float,
  courseid float,
  score float);
INSERT INTO score_ext VALUES (
  10,
  2,
  2,
  80);
INSERT INTO score_ext VALUES (
  11,
  1,
  2,
  70);
INSERT INTO score_ext VALUES (
  12,
```

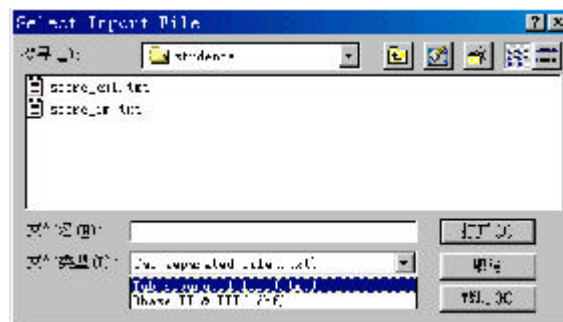
5

新建一个 txt 文件 score_in.txt，用记事本编辑之。文件中每行数据表示要导入表中的一条记录，同行的两个数据之间用 Tab 分隔，如图。这是要导入表的文本文件的标准编辑格式。



6

在 score 的数据操作窗口（见步骤 1）中选择“Rows|Import...”菜单命令。弹出下图对话框。在“文件类型”中选择“Tab separated file”，再选择在步骤 5 新建的文件。

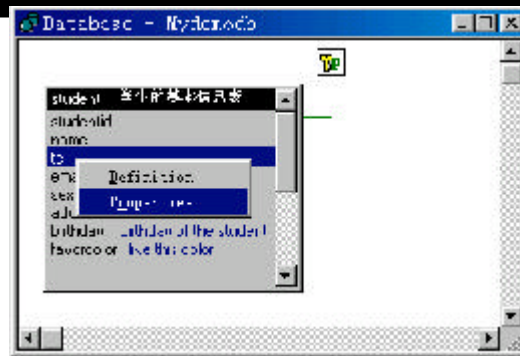


7

按“打开”按钮或双击文件名，则将文件中的数据导入数据库。

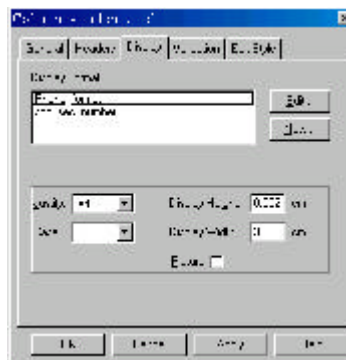
列的显示格式 定制 student 的列显示格式

1



2

选择菜单上的“ Properties ”，在弹出的对话框中选择 Display 标签页，如图。Display 标签页用来设置本列数据的显示格式、对齐方式、大小写、所占面积大小等。

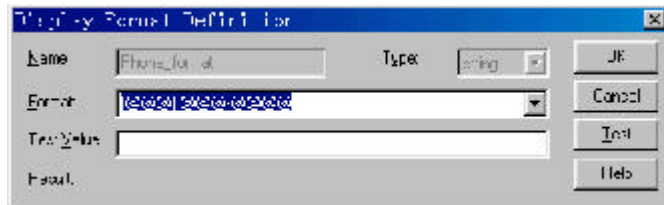


3

单击 Display Format 列表框中的 Phone format 项，按 OK 按钮确定选择。然后单击 Painter Bar 上的 Data Manipulation(Grid)按钮浏览数据，如图。看到 Tel 一系列的数据显示成了电话的格式。

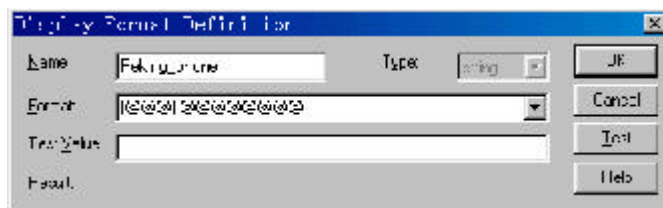
Data Manipulation for student				
d	Name	Tel	Email	
1	flate	(010) 527-3659	fla@163.net	me
2	skn	(010) 527-3660	skn@263.net	han
3	tim	(010) 527-3665	tim@263.net	me

4



5

原来显示格式是在 Format 编辑框中设置的。按 Cancel 按钮回到步骤 2 对话框，再单击对话框上的 New 按钮来新建（或叫定制）显示格式。如下图所示，在 Name 中写入新建显示格式的名字，在 Format 中输入希望的显示方



6

对话框的 Test 按钮用来测试所编辑的格式，比如在 Test Value 中输入 01062777089，按 Test 按钮，则“Result”显示为(010)62777089。



7

一路按 OK 确定。这时再用步骤 3 的方法浏览电话格式。这正是我们所希望的格式。此次操作完毕。

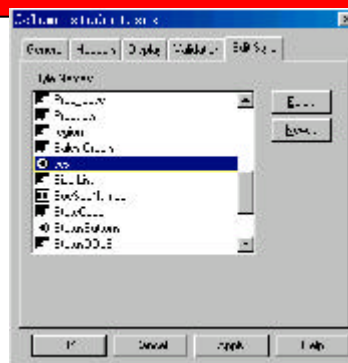
列的编辑格式 定制 student 的列编辑格式

1



2

选择菜单上的“Properties”，在弹出的对话框中选择 Edit Style 标签页，如图。Edit Style 标签页用来设置本列数据的编辑格式。

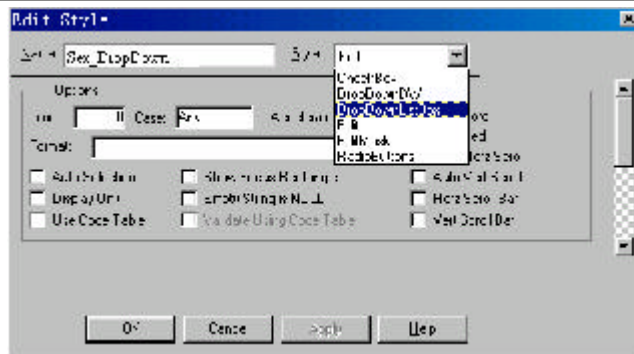


3

选中列表框中的“sex”项，按 OK 按钮确定选择。然后在 Display 页中将 Display Height 设为 1.5。利用数据操作画板浏览数据。添加一个新列，发现 Sex 列不需要输入字符，只需要选择，如图。

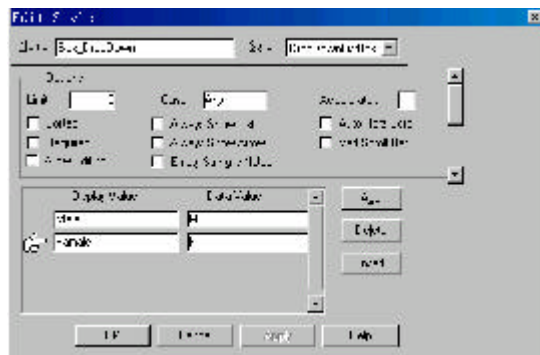
Id	Name	Tel	Email	Sex	Address
1	张三	13800138000	zhangsan@163.com	Male	Beijing
2	李四	13900139000	lisi@163.com	Female	Shanghai
3	王五	13700137000	wangwu@163.com	Male	Guangzhou

4



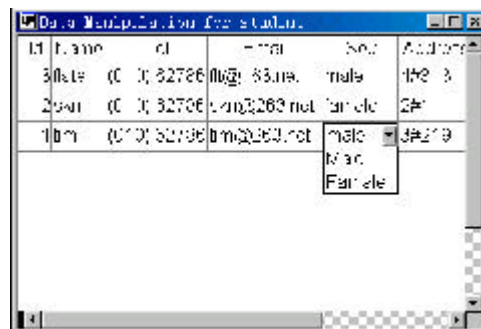
5

这时对话框变为下图的样子。在左下的 Display Value 和 Data Value 编辑框中分别输入 Male 和 M；单击右下的 Add 或 Insert 按钮，在新产生的两个编辑框中输入 Female 和 F。完成后的对话框见下图。



6

一路按确定关闭对话框，然后利用数据操作画板浏览数据，如下图。要编辑 Sex 字段的数据，只需要单击此处，便会出现组合框，从中选择性别而不需要输入字符。

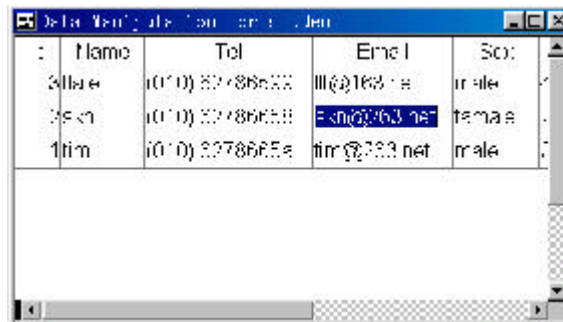


7

读者可以试着新建其它编辑格式，比如对某些字段使用 CheckBox 和 EditMask 样式。

创建有效性规则 student 的有效性规则

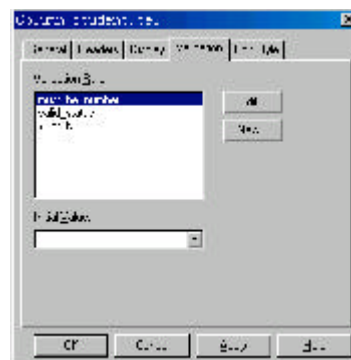
1



Name	Tel	Email	Sex
zhang	(010) 82786655	zhang163.net	male
zhang	(010) 82786655	zhang263.net	female
tim	(010) 82786655	tim@263.net	male

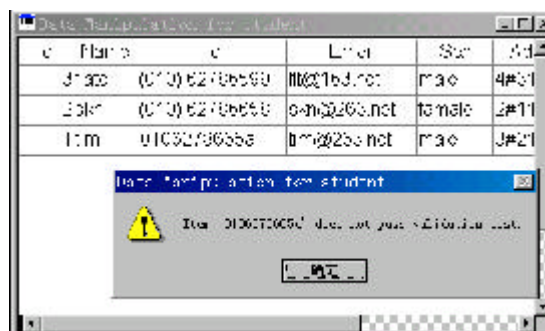
2

应该提醒用户电话号码不应该包含字母。为此退回数据库画板，在 student 窗口中的字段 Tel 处点击鼠标右键，选择所弹出菜单上的“Properties”，在弹出的对话框中选择 Validation 标签页，如图。



3

选中 Validation Rule 列表框中的 must_be_number 项，一路按 OK 按钮退出对话框。再进行步骤 1 的操作，发现将“5”改为“a”时，弹出了一个对话框，提示刚刚修改的这一项不能通过有效性检查。

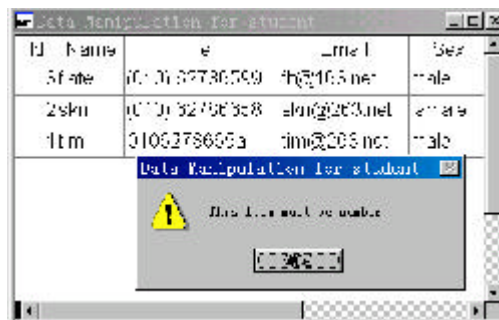


4



5

注意上图中输入的字符串必须在单引号中。一路按 OK 按钮退出对话框。进行步骤 1 的操作，发现将“5”该为“a”时，弹出了一个对话框，有效性检查的错误信息变成了“ This item must be number ”。



6

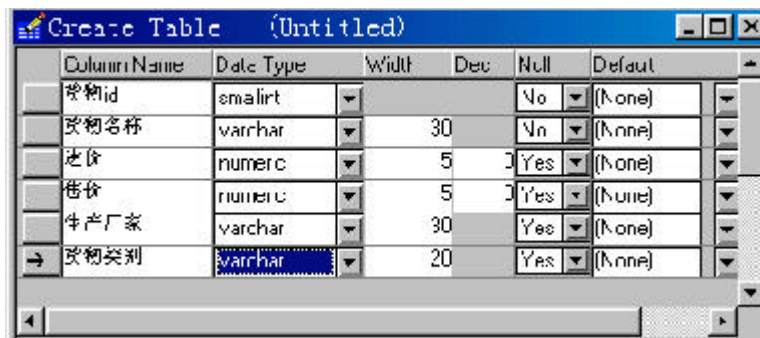
更好的提示方式应该是具体说明哪个数据项错了，为此在步骤 4 的对话框中将错误信息修改为（如图）：'Item ~' + '@tel + '~' must be number'。



7

读者可以新建有效性规则在步骤 2 的对话框中单击 New，其它与以上步骤类似。

1



2

下面设置表的属性。打开表的属性对话框，在 Primary Key 标签页中设置字段“货物 Id”为表的主码，见下图。

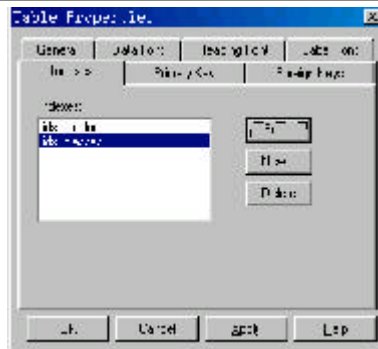


3

在属性对话框的 Index 标签页中为字段“生产厂家”建立名为 idx_producer 的索引，各项设置如下图。



4



5

在属性对话框的 General 标签页的 Table Comment 编辑框中输入对表的注释，如下图。



6

退出表的设计窗口，回到数据库画板，在 sale_goods 表窗口中的“货物类别”上单击左键，在弹出的菜单上选择 Properties，弹出如下对话框。



7

在对话框 General 标签页为本列加注释（见上图）。通过同样的操作为其他的字段加入注释说明。

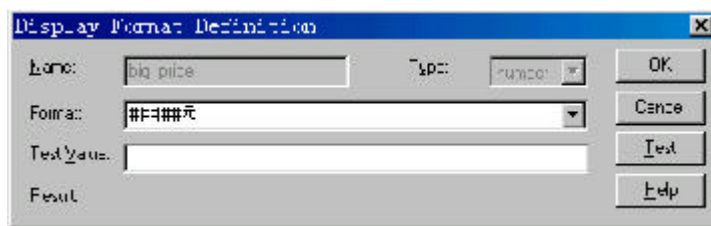
知识

1



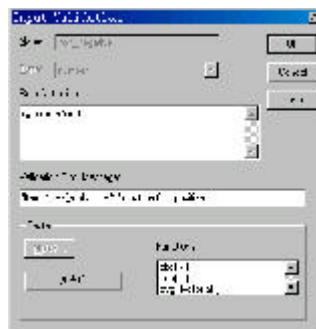
2

新增加一种数据显示格式（在“进价”字段的属性对话框的 Display 页单击 New），用于显示用人民币表示的价格，并将其命名为 big_price，如下图所示。为“进价”和“售价”字段设置这种显示格式。



3

在属性对话框的 Validation 页为“进价”和“售价”字段新增加一种合法性检查：non_negative，它的设置见下图。

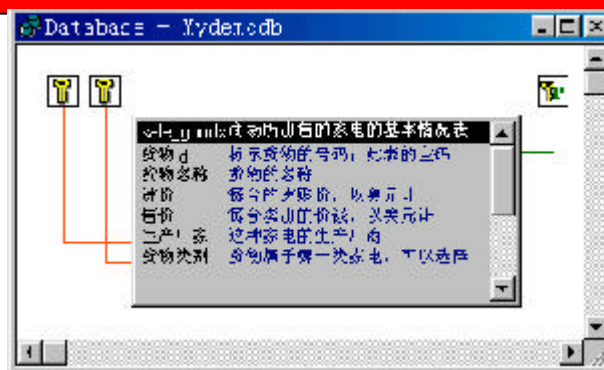


4



5

完成的表 sale_goods 见下图。



6

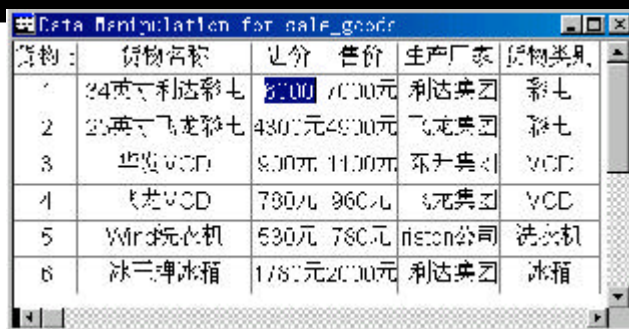
新建表 sale_goods_num, 其各个字段的基本属性见下图。这个表的结构很简单, 它包含每种家电每个季度销售量的信息。

Column Name	Data Type	Width	Dec	Null	Default
货物id	smallint			No	(None)
第一季度销量	integer			Yes	(None)
第二季度销量	integer			Yes	(None)
第三季度销量	integer			Yes	(None)
第四季度销量	integer			Yes	(None)

7

象对待表 sale_goods 一样, 请将表 sale_goods_num 各个字段的 Justify 扩展属性设为 Center。

4



货物	货物名称	进价	售价	生产厂家	货物类别
1	24英寸利达彩电	6000元	7000元	利达集团	彩电
2	25英寸飞龙彩电	4800元	4900元	飞龙集团	彩电
3	华发VCD	900元	1100元	华发集团	VCD
4	飞龙VCD	790元	960元	飞龙集团	VCD
5	Wind洗衣机	530元	780元	riston公司	洗衣机
6	冰王牌冰箱	1780元	2000元	利达集团	冰箱

5

选择 File|Save Row As...菜单项，将表导出为文本。用记事本打开所导出的文本，如下图。请读者思考，为什么文件中“货物类别”一列显示的不是“彩电”、“VCD”等，而是“T”“V”等字母呢？



货物	货物名称	进价	售价	生产厂家	货物类别
1	24英寸利达彩电	6000	7000	利达集团	T
2	25英寸飞龙彩电	4800	4900	飞龙集团	T
3	华发VCD	900	1100	华发集团	V
4	飞龙VCD	790	960	飞龙集团	V
5	Wind洗衣机	530	780	riston公司	U
6	冰王牌冰箱	1780	2000	利达集团	R

6

用记事本编辑一个文件，格式如下图。这是准备导入到表 sale_goods_num 中的数据。切记同一行的数据之间用 Tab 分隔。



1	156	84	77	101
2	110	55	44	120
3	288	166	175	233
4	130	87	99	144
5	110	218	232	66
6	76	188	210	89

7

将步骤 6 所编辑的文件导入到表 sale_goods_num 中。至此，表 sale_goods 和表 sale_goods_num 建立完毕。

知识



[返回总目录](#)

第二章

应用程序对象

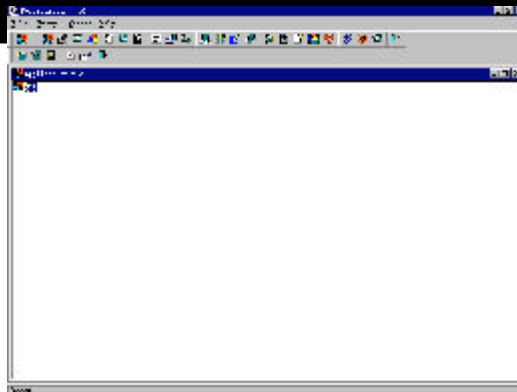
本章导读

应用程序对象是用户应用程序的入口点，我们在 PowerBuilder 中的一切应用都是围绕着应用程序对象来进行的。

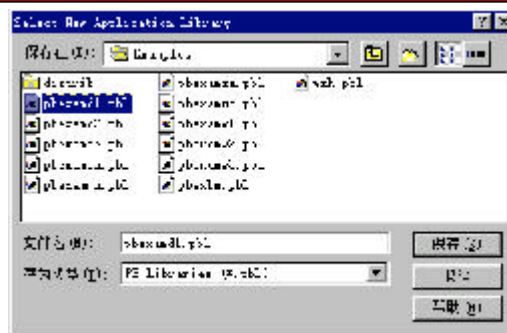
本章将向读者讲述什么是应用对象以及如何利用 Application 画板控制来完成应用程序对象的各种操作，其中包括：建立一个应用程序对象、设置应用程序对象、改变应用程序对象的属性、利用应用程序对象的事件编程等等。

本章还将详细介绍应用程序对象的属性、事件、方法和使用的各种细节。

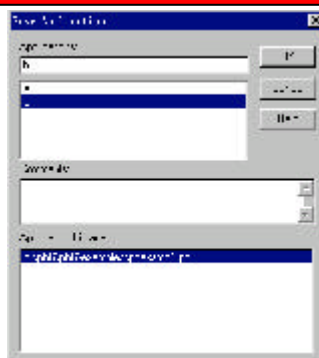
应用程序对象 建立一个新的应用对象

1**2**

从 File 菜单中选择 New 命令，在弹出的 Select New Application Library 对话框中选择一个 pbl 文件，以作为新建应用程序对象的库文件。

**3**

在弹出的 Save Application 对话框中，选择一个新的应用对象名，以创建一个新的应用程序对象。





The screenshot shows a Windows Explorer window titled 'Application - w1'. The left pane displays a tree view of the application's directory structure. The root is 'w1', which contains a subdirectory 'w_genapp_frame'. Inside 'w_genapp_frame' is 'm_genapp_frame', which contains 'w_genapp_toolbar' and 'w_genapp_dialog'. 'w_genapp_toolbar' contains 'w_genapp_toolbar' and 'w_genapp_dialog'.

6

A screenshot of the Microsoft Word 6.0 application window. The title bar at the top reads "Microsoft Word - [document name]". Below it is a menu bar with options: File, Edit, Format, Tools, Window, Help. The "Tools" menu is currently open, displaying several icons and their corresponding names: Spelling & Grammar, Thesaurus, Word Count, Indexing, Mail Merge, and Macro Recorder. The main workspace below the menu is blank white. At the bottom of the window, there is a status bar showing "Page 1 of 1" and "Word 6.0".

7

通过 PowerBuilder 的模板功能，可以很方便创建一个具有菜单和子窗口的应用程序对象。尽管现在还不知道如何使用菜单和窗口，但将在以下几章中分别介绍。

应用程序对象的组成
设置应用程序对象

1



2

在应用程序的属性窗口中选择 TextFont 一栏，用户便可以指定应用程序的缺省字体的颜色和大小，并可以指定字体是否加粗或加下划线。



3

在应用程序的属性窗口中选择 Icon 一栏，用户可以选择自定义的图标。用户选择的图标不但是应用程序的图标，而且也是应用程序中各窗口的缺省图标。



4



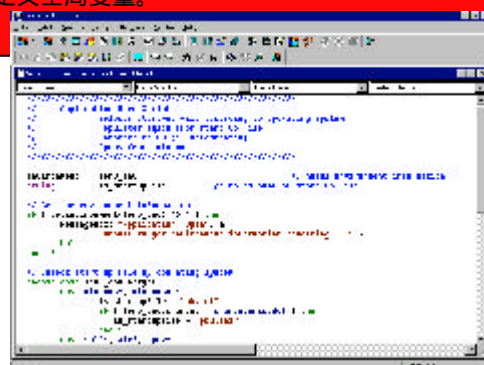
5

在应用程序的属性窗口中选择 Library 一栏，可以对当前应用程序可访问的库进行进行修改和添加。



6

可以通过点击 File 菜单中的 Script 命令来进入应用程序对象的 Open 事件在其中可以定义全局变量。

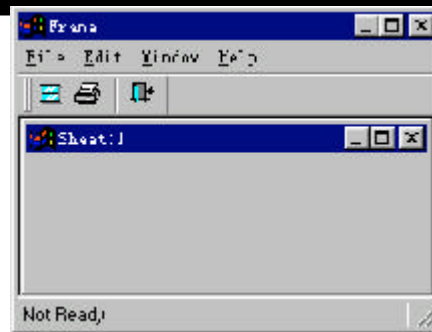


7

用户通过改变应用程序对象的组成部分尤其是全局变量和库搜索路径的设置，可以很方便的对应用程序进行设置。

应用程序对象的属性
改变应用程序对象的属性

1



2

在应用程序对象的 Open 事件中加入代码 `this.ToolbarFrameTitle = "BARBARBAR"`，将工具栏浮动时的标题栏设为 BARBARBAR。

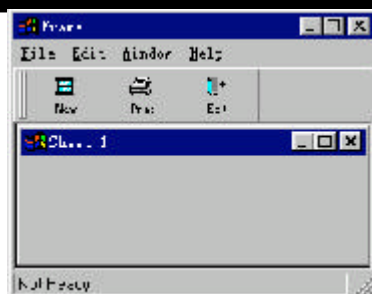


3

在应用程序对象的 Open 事件中加入代码 `this.ToolBarPopupMenuText = "You jump I jump"`，则可改变工具栏弹出时显示的文本。

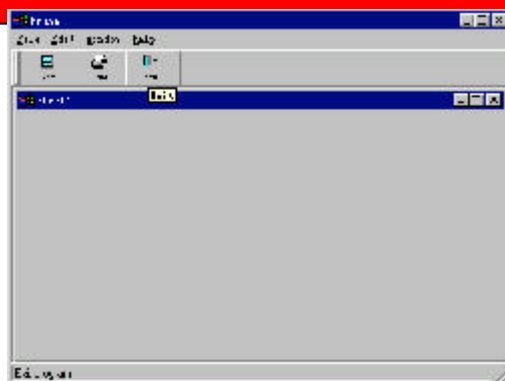


4



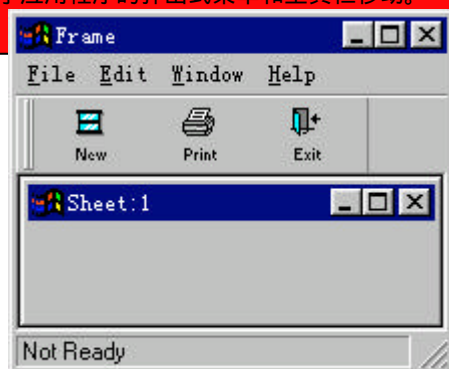
5

在应用程序对象的 `Open` 事件中加入代码 `this.ToolBarTip=True`，使工具栏在鼠标移动到上面时能显示提示。



6

用户可以通过将 `ToolBarUserControl` 设为 `False` 来禁止用户拥有更多的灵活性，这样就禁止了应用程序的弹出式菜单和工具栏移动。

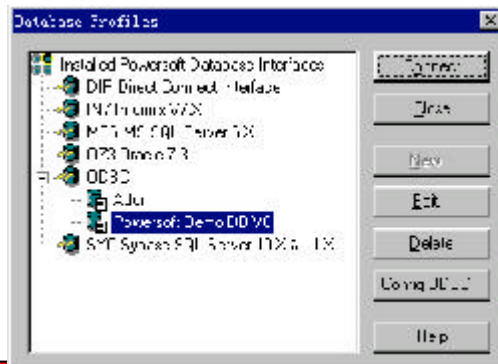


7

在本例中介绍的都是可见的属性，而实际上应用程序还有很多很重要的属性是不可见的，读者在编制自己的应用程序时会更多的接触它们。

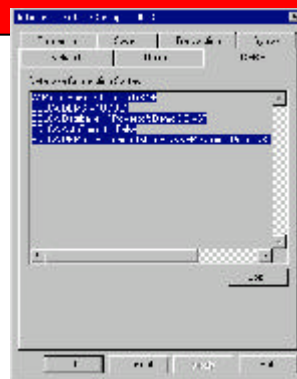
应用程序对象事件
利用应用程序对象事件编程

1



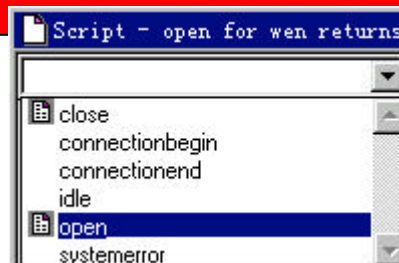
2

在打开的窗口中选择 Preview 一栏，将显示所选择数据库的连接信息，点击 Copy 按钮将联结信息拷贝下来。

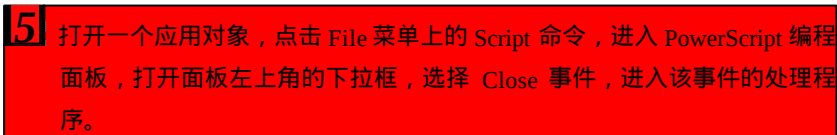


3

打开一个应用对象，点击 File 菜单上的 Script 命令，进入 PowerScript 编程面板，打开面板左上角的下拉框，选择 Open 事件，进入该事件的处理程序。



知识



A screenshot of the 'Connection Properties' dialog box, specifically the 'General' tab. The 'close' button is highlighted in blue. The dialog box has a title bar 'Connection Properties' and a standard Windows-style border. The 'close' button is located at the bottom left of the dialog box, next to the 'open' button. The 'open' button is disabled (grayed out). The 'close' button is enabled (blue). The 'close' button is labeled 'close'.

Script - close for var. returns (None)

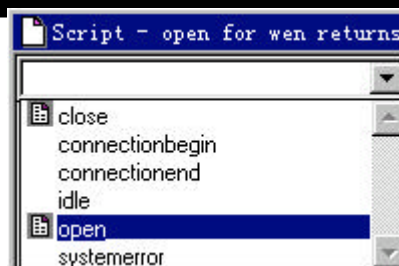
Script: Event Paste Window Paste Global Paste Local

```
disconnect;
```

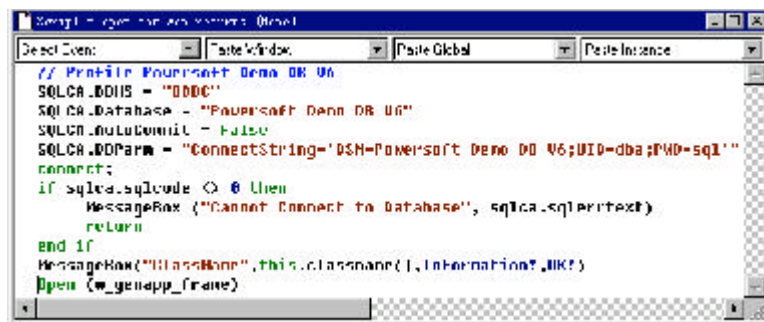
Line 1, Column 1

7 本例中利用了程序的打开和关闭的两个事件实现了程序初始化时打开数据库和主窗口的功能，以及程序关闭时断开与数据库的联结的功能，事件编程是 Windows 下编程的核心，希望读者牢加掌握。

应用程序对象方法 显示应用程序的名称和类型

1**2**

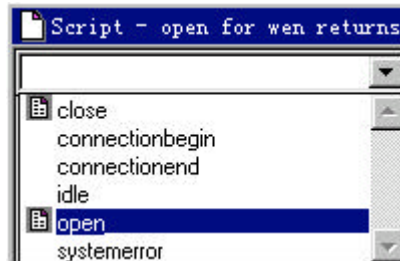
在 Open 事件处理程序中，加入代码 `MessageBox("ClassName",this.classname(),Information!,OK!)`，以在程序初始化时显示程序名称。

**3**

执行程序会出现如下的对话框，显示了当前应用程序的名称，点击确定后程序会继续执行，显示主窗口。

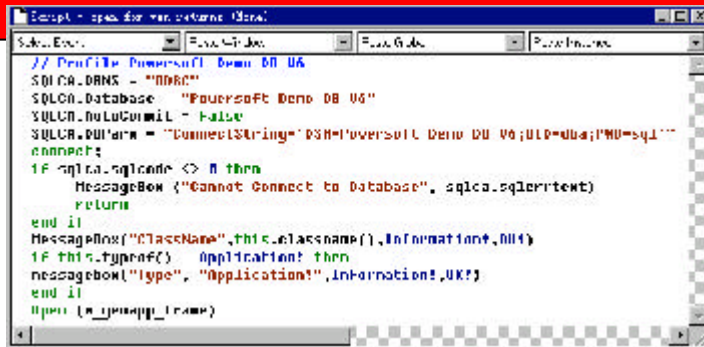


4



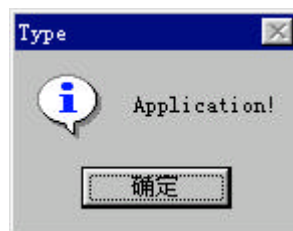
5

在事件中调用方法 `typeof()`，以返回应用程序的类型，然后判断类型是否为 `Application`！如果是则调用 `MessageBox` 显示信息。



6

执行程序，如下图所示，在弹出的信息窗口中显示当前程序的类型为 `Application`！



7

本例中所介绍的两种方法实际上并不经常调用，我们引用这两个例子是为了显示的方便，实际上更经常使用的是 `PostEvent` 和 `TriggerEvent` 方法。



[返回总目录](#)

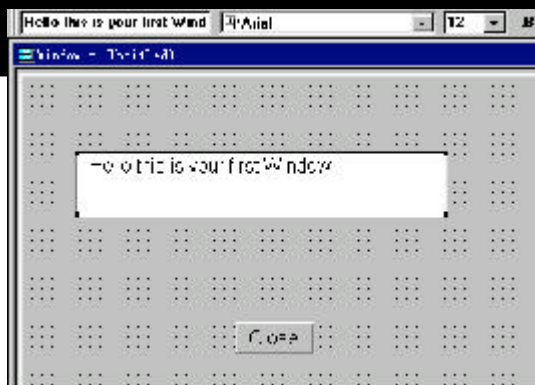
第三章

窗口对象

本章导读

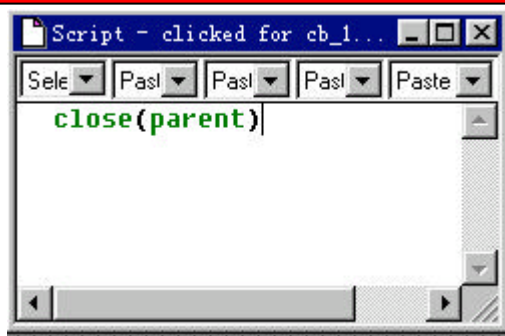
窗口对象是应用程序的主要操作界面，绝大部分 Windows 程序下的人机交互功能都由窗口完成。窗口的特点是具有一定的属性，响应一定的事件，并且能够盛装大量的控件。因此本章的内容将从如何创建一个窗口对象讲起，接下来我们将详细探讨窗口对象的各种属性，并利用它的事件进行编程。本章的重点在于对窗口中各种控件分别介绍，同时也将把菜单作为窗体的一部分加入到本章来。在本章的结尾，将介绍现在比较流行的 MDI 窗口的编程方法，并在最后附有一个综合性的实例。

4



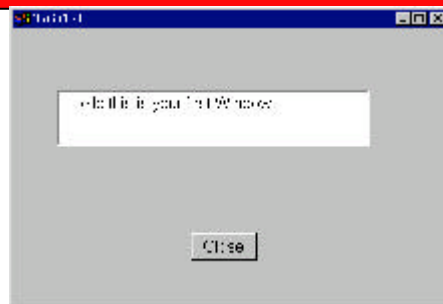
5

选择已经添加到窗口上的按钮控件，点击窗口工具栏上的 Script 命令，进入 PowerScript 环境。为该按钮的 Click 事件添加代码 `close(parent)`，以便在点击按钮时窗口能自动关闭。



6

在 PowerPanel 中选择 RunWindow 命令，将出现对话框询问是否将窗口保存，将窗口保存为 a，然后再执行该窗口，如下图所示，点击按钮时窗口会自动关闭。



7

窗口操作是 Windows 编程的基础和舞台，本例中我们简单介绍了创建一个窗口和添加控件的大致方法。在以下的例子中将分别对其进行详细的描述。

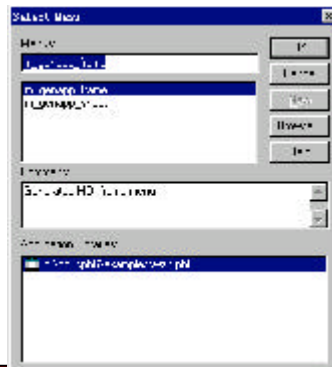
窗口对象的属性
设置窗口对象的属性

1



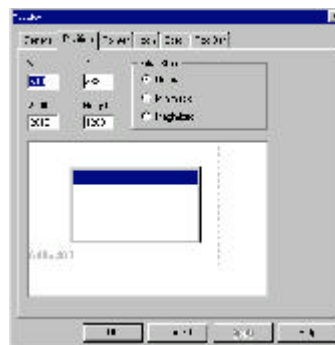
2

在属性窗口中的 General 一栏可以设置窗口的标识，菜单名称，窗口类型等属性。现在首先为窗体添加一个菜单，在 MenuName 旁点击 Browse 按钮，并选择一个已经存在的菜单。



3

在属性窗口的 Position 一栏中，用户可以指定一个窗口的位置和出现时的状态，例如用户可以使窗体在开始时便呈现为最大化。

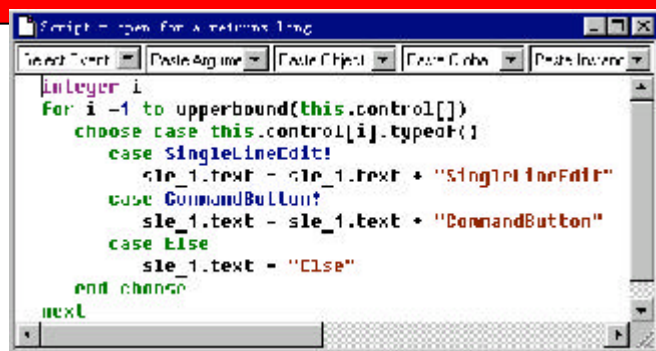


4



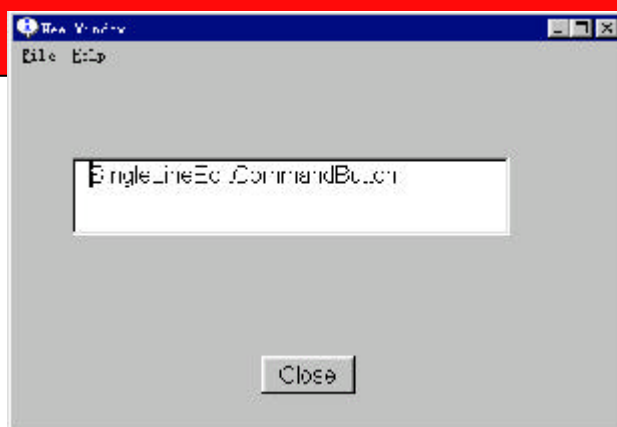
5

除了在属性窗口中设定属性以外，用户还可以在程序中为窗口设置其它属性，如在程序中加入如下代码以访问窗口的 Control 属性。



6

执行程序，可以很明显的看到窗口属性的改变。



7

在本例中，介绍了窗口的各种属性，也介绍了窗口属性设置的两种方法，即设计阶段通过属性窗口和运行阶段通过执行代码进行属性设置。

4

```
Script: CloseQuery for windows long
Select Case
Integer li_RC
Long tt
cpend = cpu()
tt = cpend - cstart
li_RC = MsgBox("ByeBye", "你运行本程序共" &
String (tt / 1000, "0,000.000") &
+ "秒" + "你确定退出吗?", Information, YesNo)

Choose Case li_RC
Case 1
Return 0
Case 2
Return 1
End Choose
```

5

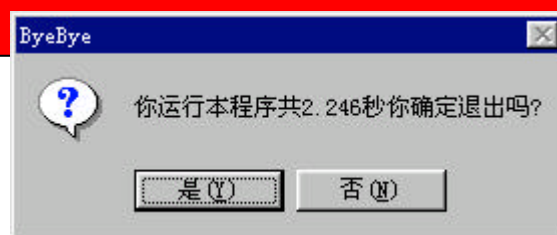
在 Close 事件中加入如下代码，在程序关闭时显示离开信息。

```
Script: Close for windows long
Select Case
Integer li_RC
Long tt
cpend = cpu()
tt = cpend - cstart
li_RC = MsgBox("ByeBye", "You are forever welcome!", &
Information, OK)

Choose Case li_RC
Case 1
Return 0
Case 2
Return 1
End Choose
```

6

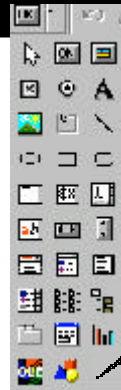
运行程序如下，在关闭程序时弹出对话框，询问退出的有关信息。



7

本例中所用到的 Open 事件主要用于初始化程序，Close 事件用于释放资源，CloseQuery 事件主要用于提示用户保存在运行程序中所做的修改。

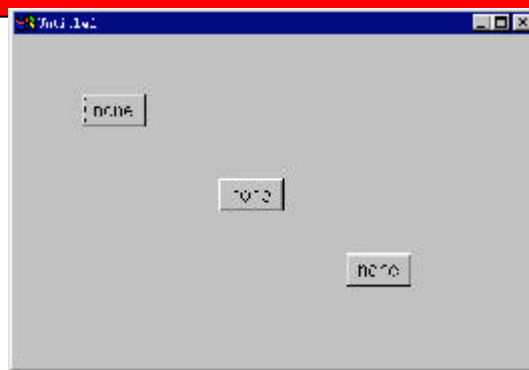
1



选择控件的下
拉式按钮

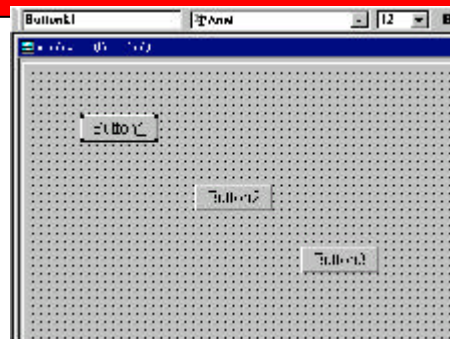
2

向窗口添加控件只需先从下拉式按钮中选择相应的控件，然后再用鼠标单击窗口上某一位置，该控件即出现在窗口中相应的位置上。

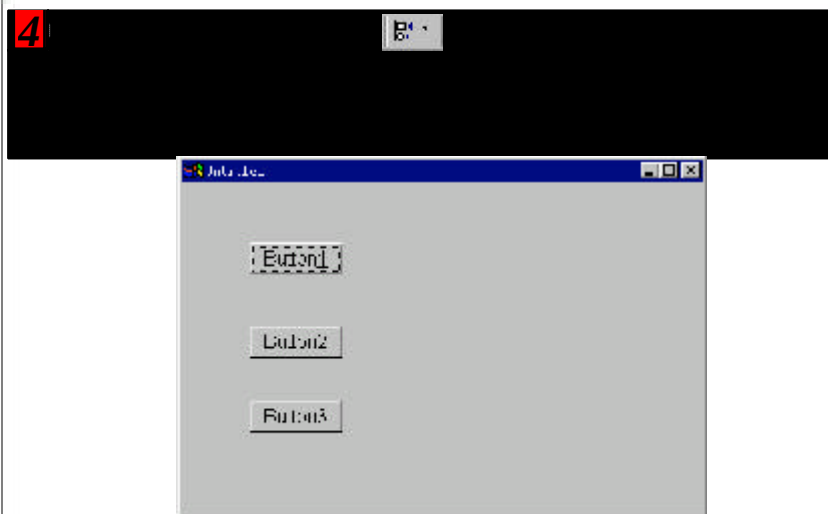


3

用户可以通过窗口工具栏下的文本工具栏来改变控件的标识，并且可以用 & 符号来定义快捷键。

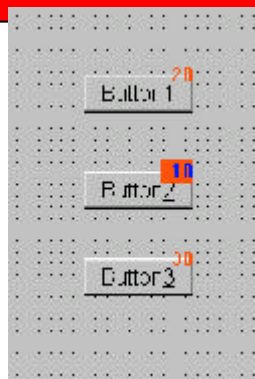


4



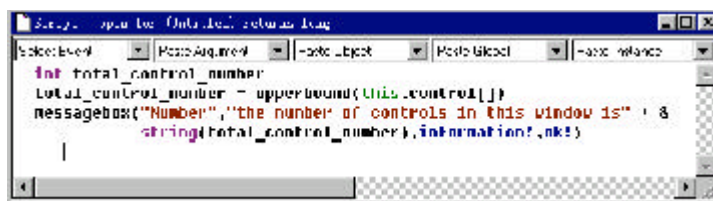
5

用户可以使用窗口工具栏中的 Tab Order 来改变控件的访问顺序，控件的访问顺序通过其 Tab 值来确定，通过 Tab Order 按钮可以显示并修改窗口中控件的 Tab 值。



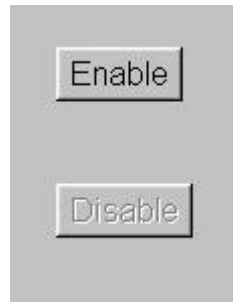
6

所有的控件都是所在窗口的属性 Control[]数组的成员，可以通过该数组来对窗口中控件进行访问。如下图代码计算并输出当前窗口中控件的个数。



7

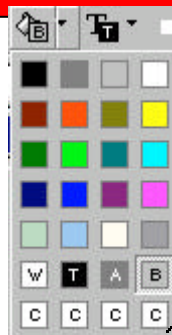
本例中简单介绍了控件的基本操作，大部分的控件需要通过事件响应来实现其功能，我们将在以后的小节中介绍。

1**2**

Visible 属性规定了控件是否可见。和 Enable 属性一样，它可以在控件的属性窗口中设置。同时也可以使用 `Control.Visible = False/True` 来在程序中显示或隐藏控件。

**3**

Foreground Color 属性和 Background Color 属性规定了控件的前景色和背景色。用户可以在窗口工具栏的颜色下拉按钮中设置它们。



选择颜色的
下拉式按钮

4



A screenshot of a dialog box with the following fields and controls:

- Name:
- Text:
- Icon:
- ☒ Style
- ☒ Enabled
- ☐ Default
- ☐ Cancel

5

Border 属性是一个布尔类型的值，它决定了控件是否有边界。只有当 Border 属性被设为 True 时，下面的 BorderStyle 属性才会有效。

Have Border

No Border

6

BorderStyle 属性是一个枚举类型，它有四个值 StyleLowered!，StyleRaised!，StyleBox!，StyleShadowBox! 四个值。用户可以在窗口工具栏上的边界下拉按钮中设置它们。

Box

Shadow Box

Raised

Lowered

7

在本例中，我们介绍了控件的一般属性。从下一节开始，将分别介绍各种控件的使用方法。

知识

控件的使用（一）
利用命令按钮执行命令

1

First Name	Phone Number	State	Last Name
...			

Relieve Clear Close

2

在窗口的 Open 事件里添加联接数据库和为数据窗口设置协议的代码。

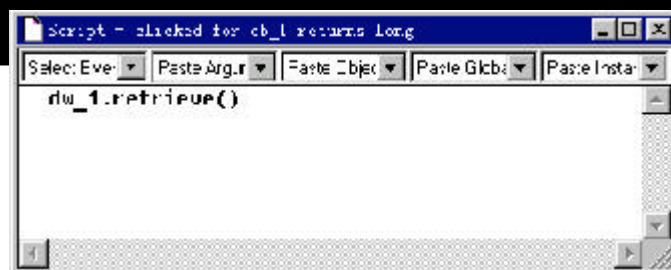
```
Script - Open for eb returns long
Select Event   Post Argument   Post Object   Post Global   Post Instance
SQLCA.DBMS   "ODBC"
SQLCA.Database = "Powersoft Demo DB U6"
SQLCA.AutoCommit = False
SQLCA.DDFName = "ConnectString='DSN=Powersoft Demo DB U6;UID=dta;PWD=sql'"
connect;
do_1.settransobject(sqlca)
```

3

在窗口的 Close 事件中添加代码 Disconnect，以在窗口关闭时断开数据库。

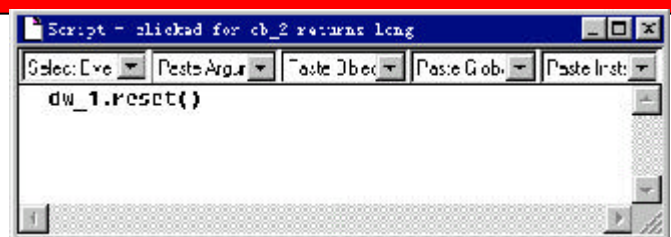
```
Script - Close for eb returns long
Select Event   Post Argument   Post Object   Post Global   Post Instance
Disconnect;
```

4



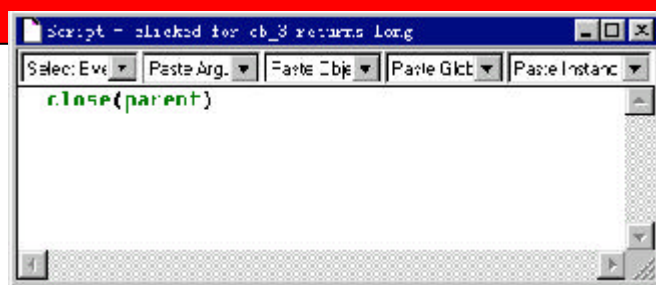
5

在按钮 Clear 的 Click 事件中加入代码 `dw_1.reset()`，执行重置数据窗口内容的功能。



6

在 Close 按钮的 Click 事件中添加代码 `Close (Parent)`，使得点击按钮时能够关闭窗口。

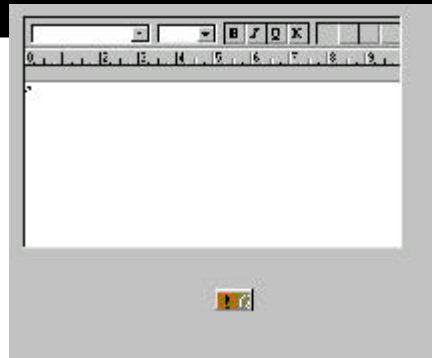


7

本例中充分利用了 CommandButton（命令按钮）的 Click 事件功能，以执行各种命令。另外，从 PowerBuilder 6.0 开始，数据窗口对象添加了命令按钮，所以对数据窗口的各种操作都可以在创建数据窗口时定义。

知识

1



2

将 RichTextEdit 控件设置如下图所示。

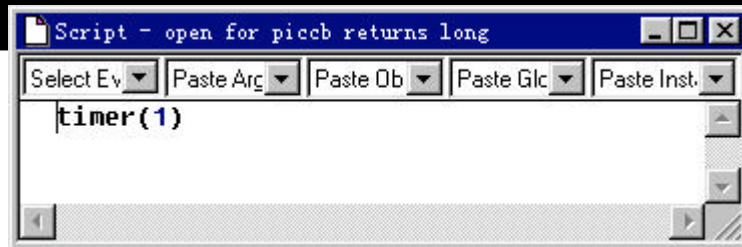


3

将 PictureBox 的 Text 属性设置为空,为 PictureBox 控件设置两幅图片为 Enabled 属性和 Disabled 属性各设置一幅图片。

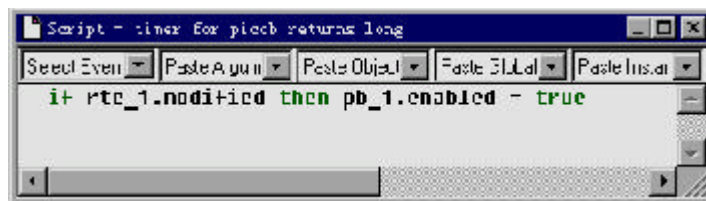


4



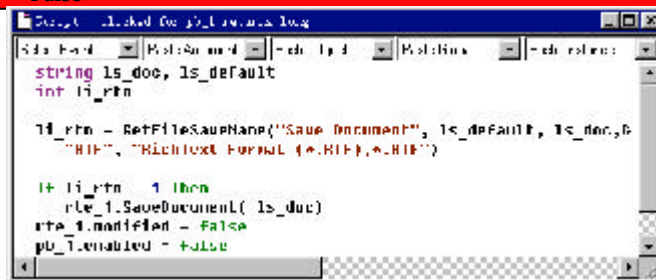
5

在窗口的 Timer 事件中加入代码 `if rte_1.modified then pb_1.enabled = true`，当 RichTextEdit 文本编辑器的内容改变时，将 PictureBox 按钮的 Enabled 属性设为 True。



6

在图形按钮的 Click 事件中，添加如下代码。当点击按钮时完成存储过程并在存储成功后将文本编辑器的 Modified 属性和图形按钮的 Enabled 属性重置为 False。



7

本例执行到最后即可，用户可以看到当编辑器内容改变后，以及当存储完毕后 PictureBox 图形的改变。

控件的使用(三) 静态文本和单行文本编辑器

1

2

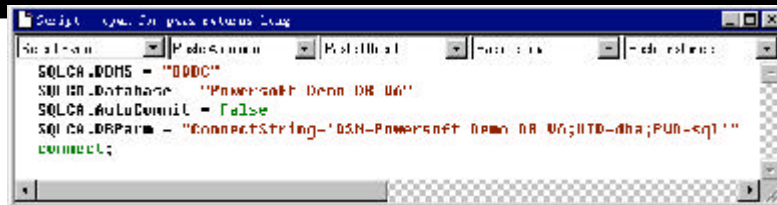
将用于输入密码的单行文本编辑器控件的 `PassWord` 属性设置为 `True`，使得在此控件上输入字符时，只能在屏幕上显示星号。

3

利用 PowerBuilder 的制表功能，制作一个如下的表格，它有 `Id` 和 `Password` 两个字段。

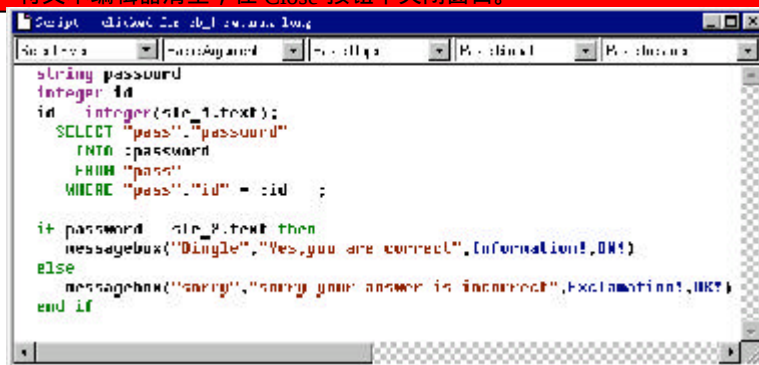
Id	Password
1111	abcd
2222	love
3333	over
4444	luck

4



5

在 Submit 按钮的 Click 事件中, 添加 Sql 查询语句查找与当前输入 Id 匹配的密码, 并与输入密码比较以给出正确或错误信息。同时在 Clear 按钮中将文本编辑器清空, 在 Close 按钮中关闭窗口。



6

执行程序, 如下图所示。可以看到当密码输入正确时的确认信息。



7

本例中利用了 SingleLineEdit 控件的 PassWord 属性来实现密码输入功能, 并且利用了表查询的 Sql 语句来查询密码, 这种方法在密码输入中经常用到。

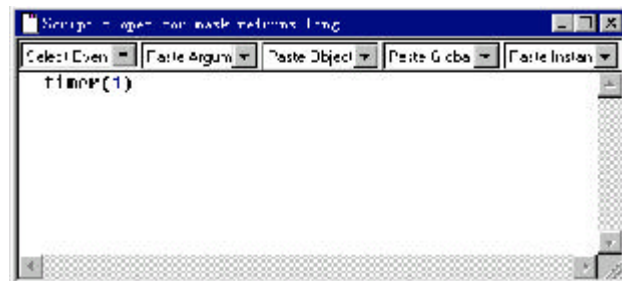
控件的使用(四) 利用 EditMask 实现限制输入

1**2**

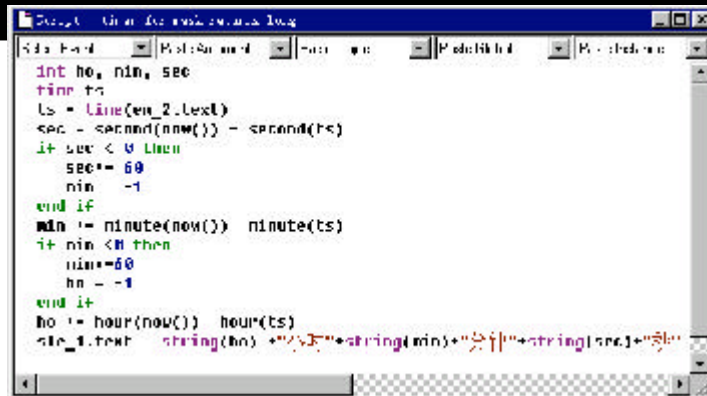
在 EditMask 控件的属性窗口中调整其属性，将 Type 设为 Time，将 Mask 设为 hh:mm:ss。

**3**

在窗口的 Open 事件中加入代码 Timer (1)，打开并设置计时器。



4



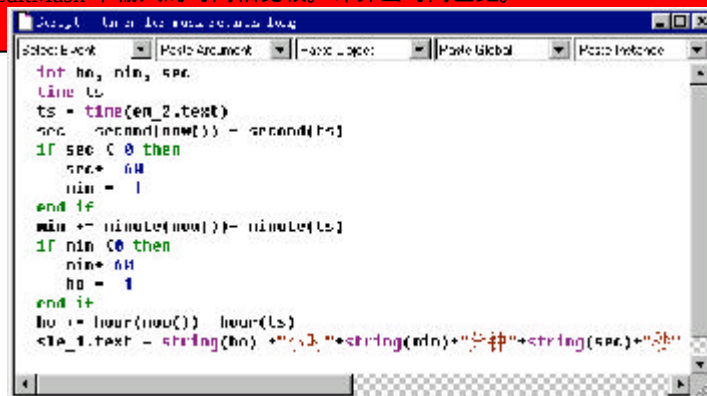
```

int h0, m0, s0;
time ts;
ts = time(en_2.text);
sec = second(now()) - second(ts);
if sec < 0 then
    sec += 60
    min -= 1
end if
min += minute(now()) - minute(ts);
if min < 0 then
    min += 60
    h0 -= 1
end if
h0 += hour(now()) - hour(ts);
str_1.text = string(h0) + ":" + string(min) + ":" + string(sec) + "秒"

```

5

利用 hour , minute , second 函数取得当前时间的小时，分钟和秒数。并和 EditMask 中输入的时间相比较。计算出时间差距。



```

int h0, m0, s0;
time ts;
ts = time(en_2.text);
sec = second(now()) - second(ts);
if sec < 0 then
    sec += 60
    min -= 1
end if
min += minute(now()) - minute(ts);
if min < 0 then
    min += 60
    h0 -= 1
end if
h0 += hour(now()) - hour(ts);
str_1.text = string(h0) + ":" + string(min) + ":" + string(sec) + "秒"

```

6

执行程序，如下图所示。程序能够自动计算出从起床到当前时间的差距。

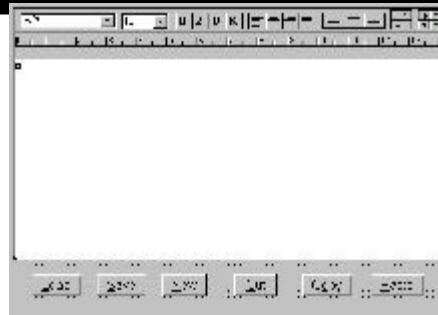


7

本例中我们利用了 EditMask 的限制输入功能，相对于 SingleLineEdit 控件它能够更有效的提示用户按格式输入。

控件的使用(五) 利用 RichTextEdit 实现文本编

1



2

在 Load 按钮的 Script 中加入如下代码，实现打开文件的功能。

```
Script: clicked for ch1 returns long
select event: | Paste document: | Paste Up: | Paste Global: | Paste Inplace:
String    ls_File, ls_Path
Integer    li_HG
If GetFileOpenName("Select File", ls_Path, ls_File, "rtf", &
  "Rich Text (*.rtf), *.rtf") = 1 Then
  ls_FileName = ls_Path
  rte_1.InsertDocument(ls_FileName, True, FileTypeRichText)
  parent.Title = ls_File + " - Richpad"
  rte_1.modified = False
End If
```

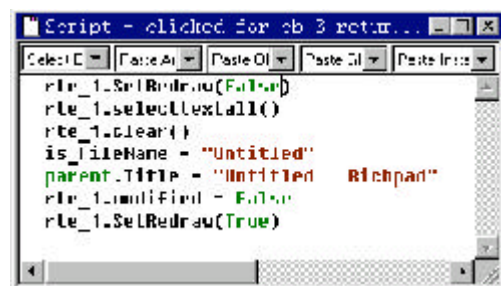
3

在 Save 按钮中加入如下代码，完成存储过程，并且在存储文件已经存在的情况下给与提示信息。

```
Script: clicked for ch1 returns long
Integer    li_Yet
Boolean    lb_Exist
String     ls_File, ls_Path

ls_Path = ls_FileName
If GetFileSaveName("Select File", ls_Path, ls_File, "rtf", &
  "Rich Text (*.rtf), *.rtf") = 1 Then
  ls_FileName = ls_Path
  parent.Title = ls_File + " - Richpad"
  lb_Exist = FileExists(ls_FileName)
  If lb_Exist THEN li_Yet = MsgBox("Save", &
    "OK to write over" + ls_FileName, &
    Question?, YesNo?)
  If li_Yet = 1 Then rte_1.Savedocument(ls_FileName, FileTypeRichText)
End If
```

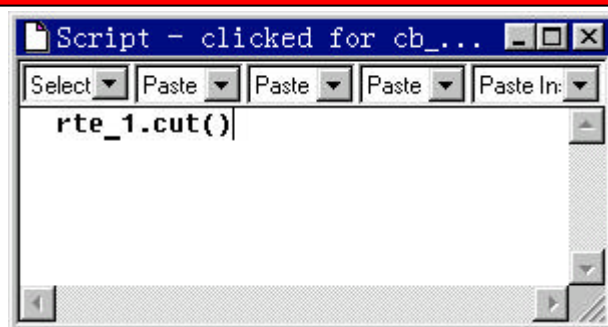
4



```
Script - clicked for cb 3 retur...
Select:
Paste As:
Paste Of:
Paste Gil:
Paste Into:
rte_1.SelRedraw(False)
rte_1.selectAll()
rte_1.clear()
is_fileName = "Untitled"
parent.Title = "Untitled Richpad"
rte_1.modified = False
rte_1.SelRedraw(True)
```

5

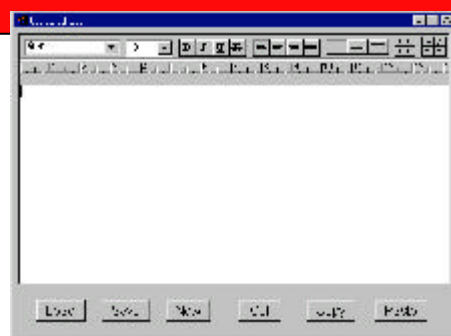
在 Cut , Copy , Paste 三个按钮中，分别加入代码 `rte_1.cut()`、`rte_1.copy()`、`rte_1.paste()`，以实现剪贴、复制、粘贴等功能。



```
Script - clicked for cb_...
Select:
Paste:
Paste:
Paste:
Paste In:
rte_1.cut()
```

6

执行程序，其画面如下，该程序可实现一些最基本的字处理功能。



7

本实例介绍了利用 RichTextEdit 控件进行文本处理的基本功能，它的一些其它功能例如插入字段等，用户可以查询帮助或阅读有关实例。

知识

控件的使用（六） 多行文本编辑器的属性

1**2**

将第二个多行文本编辑器的 Auto Vscroll 属性设为 True，将第三个多行文本编辑器的 Auto Hscroll 属性设置为 True。

**3**

在窗口的 Open 事件中加入如下代码，为三个多行文本编辑器设置同样的文本。

```
Script: Open for vscroll returns long  
// Open script for M_Multitest  
  
//use this to reset the MLE after attributes have been changed  
is_Multitest "This is MultilineEdit Box. You can change the attributes of this " + L  
"box by checking and unchecking the boxes at right. Notice the different effects " + L  
"the attributes have on the MLE. Notice the effects of the scroll bars, and the auto " + L  
"scrolling."  
  
MLE_scroll.text is_Multitest  
MLE_vertical.text is_Multitest  
MLE_horizontal.text is_Multitest
```

4

```
//Add or Remove a vertical scroll bar to the MLE
If this.checked Then
    mle_scroll.vscrollbar = True
Else
    mle_scroll.vscrollbar = False
End If
```

5

在 Horizontal 复选框中加入如下代码，以实现第一个多行文本编辑器的 Auto HScroll 属性的设置。

```
//Add or Remove a horizontal scroll bar to the MLE
If this.checked Then
    mle_scroll.hscrollbar = True
Else
    mle_scroll.hscrollbar = False
End If
```

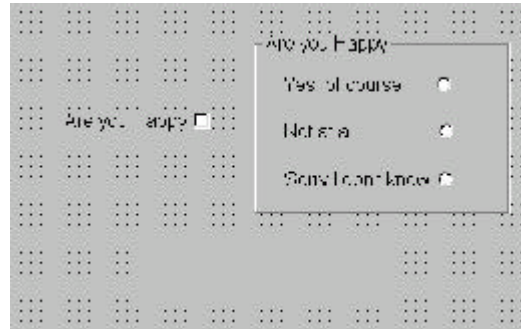
6

在 Close 按钮中加入代码 close (parent) 以关闭窗口，如下为程序执行外观。



7

本例中利用多行文本编辑器的特有属性 Auto VScroll 和 Auto HScroll，这是两个对显示文本很有帮助的属性，希望读者能认真掌握。

1**2**

在第一个单选框 (Radio Button) 的 Click 事件中, 添加如下代码, 使得静态文本控件显示高兴信息。

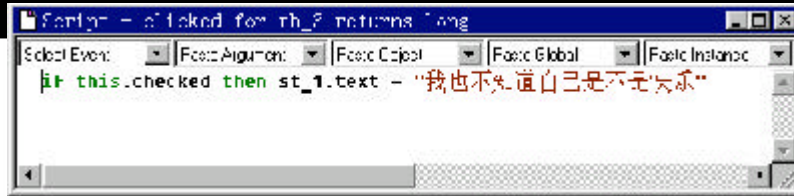
```
if This.checked Then st_1.Text = "嘻嘻哈哈嘻嘻哈哈"
```

3

在第二个单选框的 Click 事件中加入如下代码, 显示悲伤信息。

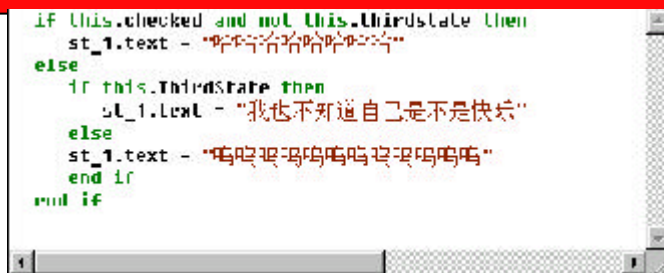
```
if This.checked Then st_1.Text = "呜呜呜 呜呜呜呜 呜呜呜"
```

4



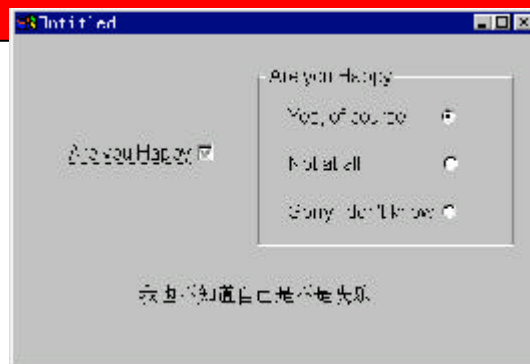
5

在 CheckBox 的 Click 事件中，加入如下代码。根据 Check Box 的三种状态决定显示在静态文本控件中的内容。



6

执行程序，如下图所示。当单选框和复选框的选中情况发生改变时，Static Text 控件的显示内容也随之变化。



7

在本例中，介绍了单选框和复选框的使用方法，用户需要记住的是：同一组单选框按钮必须放在一个 Group Box 或其它容器控件当中，而且在任何情况下都只能有一个被选中。复选框则可同时选中多个。

4



```
Script - clicked for cb 2 returns long
Select Event: Paste Argument Paste Object Paste Global Paste Instance
do while lb_1.totalselected() > 0
  lb_2.additem(lb_1.selecteditem())
  lb_1.deleteitem(lb_1.selectedindex())
loop
```

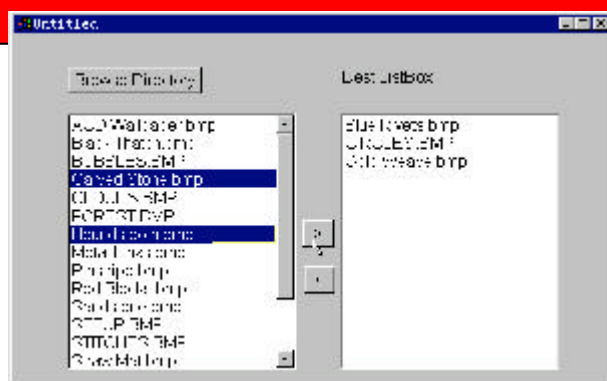
5

在按钮的  Clicked 事件中加入如下代码，使得列表 2 中被选择的元素移至列表 1 中。

```
Script - clicked for cb 3 returns Long
Select Event: Paste Argument Paste Object Paste Global Paste Instance
do while lb_2.totalselected() > 0
  lb_1.additem(lb_2.selecteditem())
  lb_2.deleteitem(lb_2.selectedindex())
loop
```

6

执行程序，其效果如下图所示。



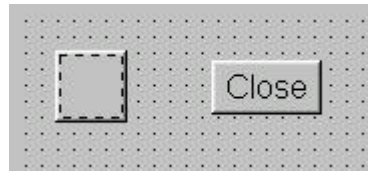
7

本例中主要运用了 ListBox 控件的 DirList、SelectedIndex 和 SelectedItem 函数，这也是在实际编程中经常用到的功能。

知识窗

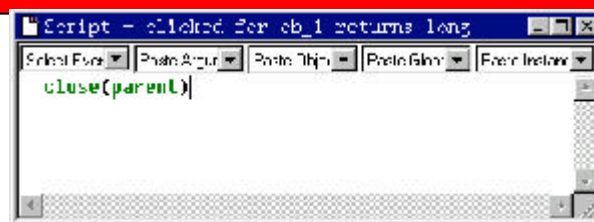
4

知识



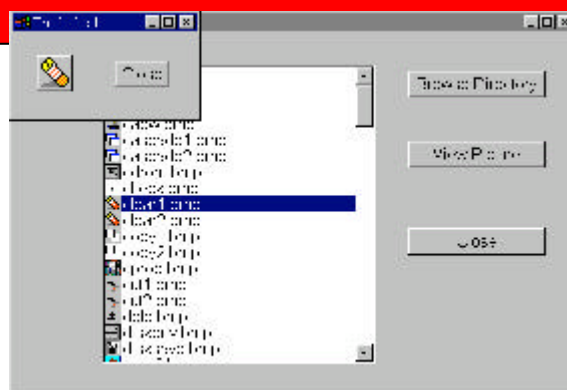
5

在两个窗口的 Close 按钮中，都添加代码 `Close (parent)`，以关闭窗口并返回。



6

执行程序，浏览位图的功能如下图所示。



7

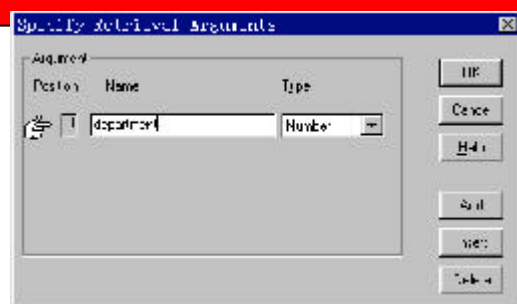
在本例中，首先利用了 `ListBox` 的 `DirList` 函数，然后又用到了 `PictureListBox` 的 `AddItem` 函数，这是一个很重要的技巧。

4

emp/ose	Label	Type	Description
emp_id	Employee ID	number	Id of an employee within the company
manager_id	Manager ID	number	Id of the manager of the employee
emp_name	First Name	varchar2	First name of the employee
emp_name	Last Name	varchar2	Last name of the employee
dept_id	Department ID	number	Id of the department where the employee works
emp_email	Email	varchar2	Employee email address
emp_phone	Phone	number	Employee phone number
emp_fax	Fax	number	Employee fax number
emp_hire_date	Hire Date	date	Date when the employee was hired
emp_salary	Salary	number	Employee salary
emp_commission_pct	Commission Pct	number	Employee commission percentage
emp_manager_id	Manager ID	number	Id of the manager of the employee

5

在菜单上选择 Design|Retrieval Arguments，定义一个 Department 参数。



6

在 where 一栏中选择 WHERE "employee"."dept_id" = :department，至此完成数据窗口设置，将其存储为 emp 即可。

Column	Table	Column	Table
emp_id	employee	dept_id	department

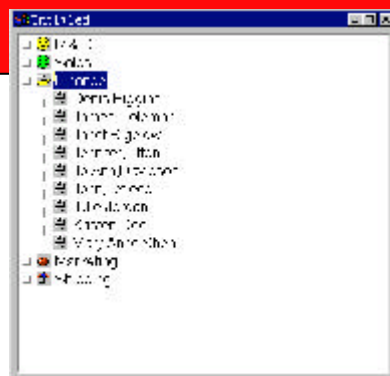
7

以上为带参数的数据窗口控件的一般设置方法，更详细的阐述将在数据窗口一章中介绍。

知识

5 在 TreeView 控件的 ItemPopulate 事件中加入如下代码, 实现检索数据窗口添加子元素的功能。

6 执行程序，如下图所示。



7 本实例重点讲述了将 TreeView 控件和数据库相联接的技巧，这也是 PowerBuilder 中窗口控件编程的目的所在。

控件的使用(十一) ListView 控件的使用

1

Field	Type	Field	Description
id	ProductID	Vendor	Vendor Identifier or Code of the product
name	Product Name	char(15)	Name of the product
description	Product Description	text(40)	Description of the product
unit price	Unit Price	float(8)	Unit price per product
color	Color	char(6)	Color of the product
quantity	Quantity	integer	Amount of the product in stock
unit price	Unit Price	float(8)	Unit price per product
unit price	Unit Price	float(8)	Unit price per product
unit price	Unit Price	float(8)	Unit price per product
unit price	Unit Price	float(8)	Unit price per product

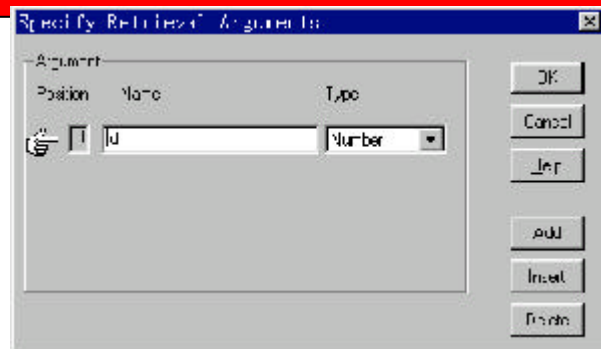
2

另外新建一个 DataWindow，在 Presentation Style 一栏中选定格式为 freeform。

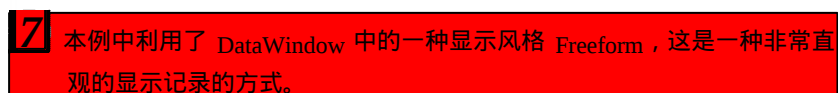
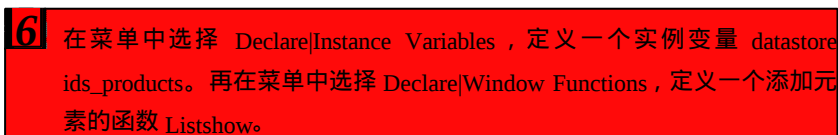
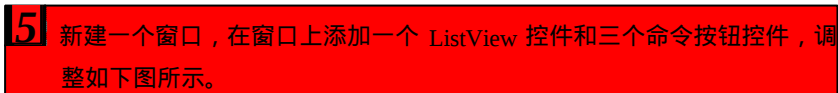


3

选定表格 Product 中和上一个数据窗口相同的字段，并定义一个参数 id。



知识



控件的使用（十一）
ListView 控件的使用

1

```
Script: open for lv returns long
SQLN.DEMS "DB00"
SQLN.Database "PowerSoft Demo DB U6"
SQLN.AutoCommit False
SQLN.DBParam "ConnectString 'DSN PowerSoft Demo DB U6;UID dba;PWD sql'"
connect;
listshow()
```

2

在三个按钮的 Click 事件中，分别加入代码 `lv_1.view = ListViewList!`，`lv_1.view = ListViewLargeIcon!`，`lv_1.view = ListviewsmallIcon!`。以实现改变 ListView 控件的显示方式。

```
Script - clicked for cb 3 returns long
lv_1.view = Listviewsmall!
```

3

在 ListView 控件的 doubleclicked 事件中加入如下代码，首先利用 `GetItem` 获得当前被双击的元素。然后利用参数 `id` 打开新的窗口。

```
Script: doubleclicked for lv_1 returns long
String ls_ID
ListViewItem lviItem

GetItem(index, lviItem)
ls_ID = CStr(lviItem.data)

If Trim(ls_ID) <> "" Then
    OpenWithParamInfo, Integer(ls_ID)
End If
```

4

5

在窗口的 Open 事件中加入代码如下，以实现用传递来的参数来反馈数据窗口的功能，并在 Close 按钮的 Clicked 事件中加入代码 close (parent)，以在点击时关闭窗口。

```
Script - open for info returns long
Select Event: Clicked
Code:
Integer li_ID
Blob blob_BMP

li_ID = Message_DoubleParam
do_1.Set Trans.Object (li_ID)
do_1.Retrieve(li_ID)
```

6

执行程序如下，双击元素会显示它的详细信息窗口。



7

本例中再次将窗口控件和数据库的信息联接起来，这在 PowerBuilder 编程中是非常重要的。

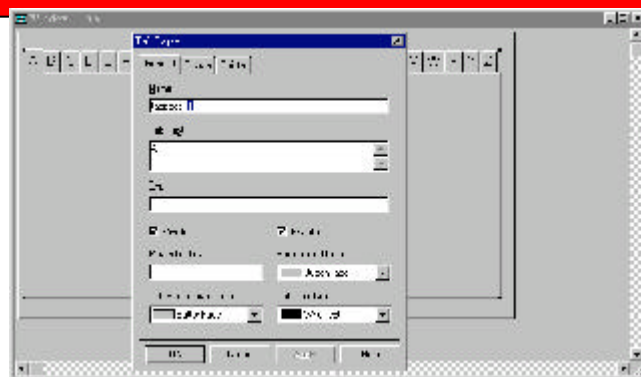
控件的使用（十二） 用 Tab 控件制作电话号码簿

1



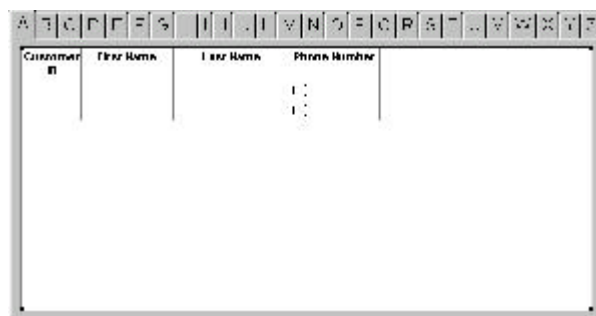
2

在每一个标签页的属性窗口中，将 Tab Text 属性分别设为从 A 到 Z 的 26 个大写字母。



3

在窗口上（Tab 控件之外）建立一个 DataWindow 控件并拖放到 Tab 控件上。将 DataWindow 指向一个在 SQL 语句中有 WHERE "customer"."lname" like :firstletter 的数据窗口。



4

```
SQLCA.DBMS = "ORACLE"
SQLCA.Database = "PowerSoft Demo DB 06"
SQLCA.AutoCommit = False
SQLCA.DBParam = "BannerString 'USH PowerSoft Demo DB 06;UID dba;PWD sql'"
connect;
dw_1.SetTransObject(SQLCA)
dw_1.retrieve("06")
```

5

在标签控件的 SelectionChanged 事件中，加入如下代码。以实现利用标签页字母检索数据窗口的作用。

```
Script - selectionchanged for tab_1 returns long
Select Event | Enter Argument | Paste Object | Paste Code | Enter Instance
dw_1.reset()
dw_1.retrieve([this.Control[newindex].text + "%"])
```

6

执行程序，如下图所示。该程序实现了按姓氏的首字母自动分类的功能。

TestList

Customer ID	First Name	Last Name	Phone Number
1	John	Smith	(919) 555-1000
2	Paul	Green	(919) 555-1001
3	Adam	White	(919) 555-1002
4	John	Black	(919) 555-1003
5	John	Black	(919) 555-1004
6	John	Black	(919) 555-1005
7	John	Black	(919) 555-1006
8	John	Black	(919) 555-1007
9	John	Black	(919) 555-1008
10	John	Black	(919) 555-1009
11	John	Black	(919) 555-1010
12	John	Black	(919) 555-1011
13	John	Black	(919) 555-1012
14	John	Black	(919) 555-1013
15	John	Black	(919) 555-1014
16	John	Black	(919) 555-1015
17	John	Black	(919) 555-1016
18	John	Black	(919) 555-1017
19	John	Black	(919) 555-1018
20	John	Black	(919) 555-1019
21	John	Black	(919) 555-1020
22	John	Black	(919) 555-1021
23	John	Black	(919) 555-1022
24	John	Black	(919) 555-1023
25	John	Black	(919) 555-1024
26	John	Black	(919) 555-1025
27	John	Black	(919) 555-1026
28	John	Black	(919) 555-1027
29	John	Black	(919) 555-1028
30	John	Black	(919) 555-1029
31	John	Black	(919) 555-1030
32	John	Black	(919) 555-1031
33	John	Black	(919) 555-1032
34	John	Black	(919) 555-1033
35	John	Black	(919) 555-1034
36	John	Black	(919) 555-1035
37	John	Black	(919) 555-1036
38	John	Black	(919) 555-1037
39	John	Black	(919) 555-1038
40	John	Black	(919) 555-1039
41	John	Black	(919) 555-1040
42	John	Black	(919) 555-1041
43	John	Black	(919) 555-1042
44	John	Black	(919) 555-1043
45	John	Black	(919) 555-1044
46	John	Black	(919) 555-1045
47	John	Black	(919) 555-1046
48	John	Black	(919) 555-1047
49	John	Black	(919) 555-1048
50	John	Black	(919) 555-1049
51	John	Black	(919) 555-1050
52	John	Black	(919) 555-1051
53	John	Black	(919) 555-1052
54	John	Black	(919) 555-1053
55	John	Black	(919) 555-1054
56	John	Black	(919) 555-1055
57	John	Black	(919) 555-1056
58	John	Black	(919) 555-1057
59	John	Black	(919) 555-1058
60	John	Black	(919) 555-1059
61	John	Black	(919) 555-1060
62	John	Black	(919) 555-1061
63	John	Black	(919) 555-1062
64	John	Black	(919) 555-1063
65	John	Black	(919) 555-1064
66	John	Black	(919) 555-1065
67	John	Black	(919) 555-1066
68	John	Black	(919) 555-1067
69	John	Black	(919) 555-1068
70	John	Black	(919) 555-1069
71	John	Black	(919) 555-1070
72	John	Black	(919) 555-1071
73	John	Black	(919) 555-1072
74	John	Black	(919) 555-1073
75	John	Black	(919) 555-1074
76	John	Black	(919) 555-1075
77	John	Black	(919) 555-1076
78	John	Black	(919) 555-1077
79	John	Black	(919) 555-1078
80	John	Black	(919) 555-1079
81	John	Black	(919) 555-1080
82	John	Black	(919) 555-1081
83	John	Black	(919) 555-1082
84	John	Black	(919) 555-1083
85	John	Black	(919) 555-1084
86	John	Black	(919) 555-1085
87	John	Black	(919) 555-1086
88	John	Black	(919) 555-1087
89	John	Black	(919) 555-1088
90	John	Black	(919) 555-1089
91	John	Black	(919) 555-1090
92	John	Black	(919) 555-1091
93	John	Black	(919) 555-1092
94	John	Black	(919) 555-1093
95	John	Black	(919) 555-1094
96	John	Black	(919) 555-1095
97	John	Black	(919) 555-1096
98	John	Black	(919) 555-1097
99	John	Black	(919) 555-1098
100	John	Black	(919) 555-1099

7

本程序中最关键的是利用了 SQL 语言中的关键字 like，以实现按首字母的检索。用户也可以利用数据窗口的 Filter 函数完成相同的功能。

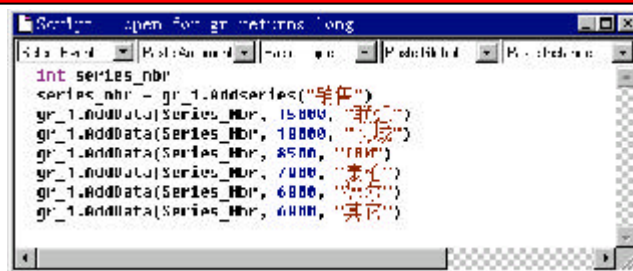
4

知识



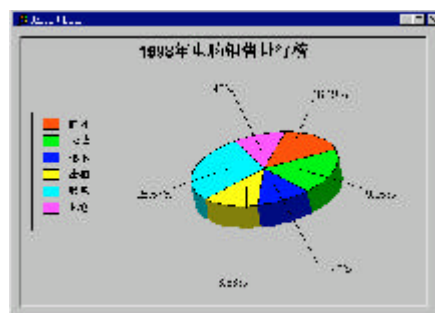
5

在窗口的 Open 事件中加入如下代码，首先利用 Addseries 函数添加一个数据系列。再利用返回值调用 AddData 函数，将数据项添加到图表中。



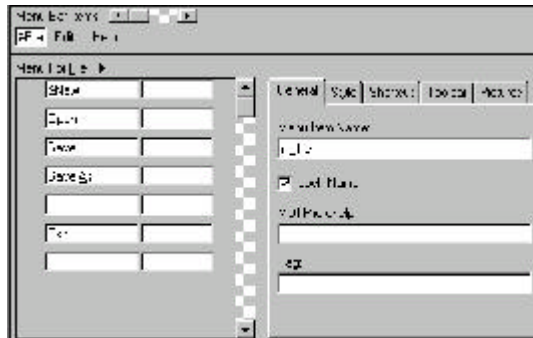
6

执行程序如下图所示。

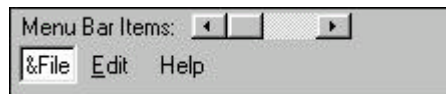


7

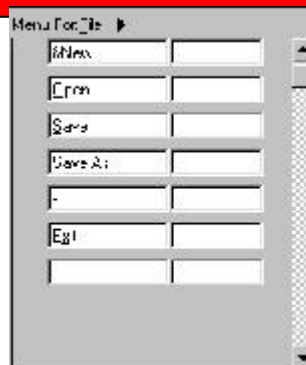
本例中重点是调用了 Addseries 和 AddData 两个函数，另外如何将 Graph 控件与数据窗口相结和将在以后的章节中介绍。

窗口菜单（一）
创建一个菜单**1****2**

下拉式菜单的创建步骤如下，首先在 Menu Bar Items 中输入顶级菜单区的名称。

**3**

在 Menu For File 中输入各下拉式菜单的名称。



4



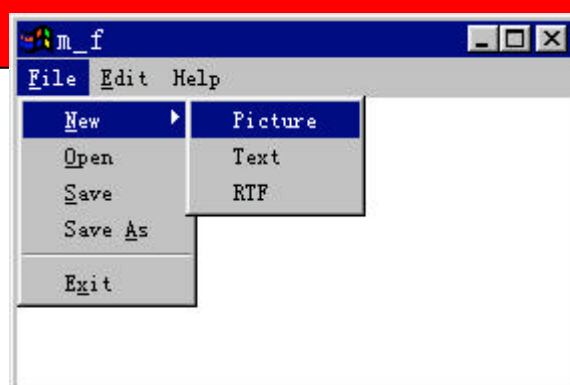
5

通过菜单描绘器的菜单栏，用户可以插入、删除、移动菜单项，还可以实现其它很多功能。



6

单击菜单工具栏上的 Preview 按钮，可以实现对当前菜单的预览。



7

本例中介绍了建立一个普通菜单的基本方法。

窗口菜单（二）
菜单的属性

1

General | Style | Shortcut | Toolbar | Pictures

Menu Item Name:
fr_file

☒ Lock Name

Mnemonic:

Tag:

2

Style 标签页上可以设置选中对勾标志的菜单项，菜单项的有效性，菜单对象的类型可以是 Normal、About 或 Exit。

General | Style | Shortcut | Toolbar | Pictures

☐ Checked ☒ Enabled

☒ Visible ☐ Shift Over/Down

Menu Merge Option: Type Option
Exclude Normal

3

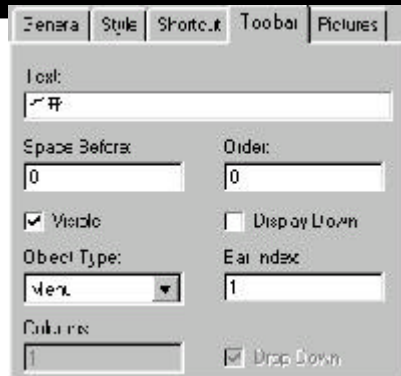
在 Shortcut 标签页中，用户可以为当前标签页设置快捷键，注意菜单项不能有子项。

General | Style | Shortcut | Toolbar | Pictures

Key: 0

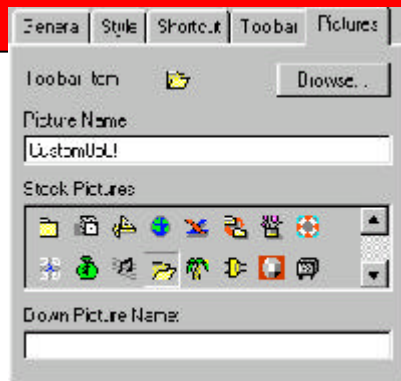
☐ Alt ☒ Ctrl ☐ Shift

4



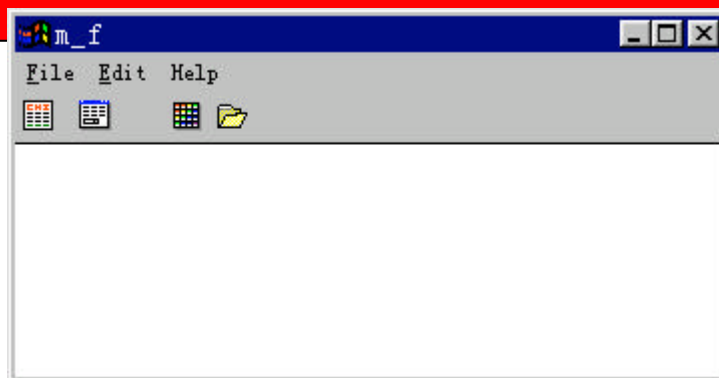
5

Picture 工具栏的作用是为工具栏上的菜单项显示图标。



6

包括工具栏的菜单预览后如下图所示。



7

菜单项的属性能够帮助我们制作出各种不同用途和风格的菜单项，还能够开发出形象生动的工具栏。

4

```
Script: selected for m_save returns 0 xml?
int li_ret
boolean lb_exist
String ls_File, ls_Path

ls_Path = ls_FileName
IF GetFileSaveName("Select File", ls_Path, ls_File, "rtf", &
"Rich Text (*.rtf), *.rtf") = 1 Then
ls_FileName = ls_Path
multiline.Title = ls_File + " - Richpad"
lb_exist = FileExists(ls_FileName)
IF lb_exist THEN li_ret = MessageBox("Save", &
"OK to write over" + ls_FileName, &
Question!, YesNo!)
IF li_ret = 1 THEN multiline.rtf_1.Savedocument(ls_FileName, FileTypeRichText!)
End IF
```

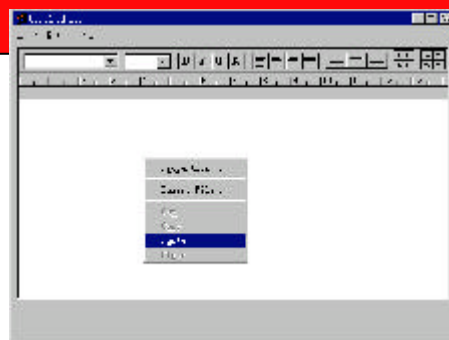
5

在窗口的 RTF 控件的文本编辑器的 RButtonDown 事件中，加入代码如下。实现点击鼠标右键弹出菜单的功能。

```
Script - rbuttondown for mco_1 returns long
$Select: Event | Paste Argument | Focus: Control | Paste Global | Paste Instance
h_+ .m_edit.papmenu(pointerx(), pointery())
```

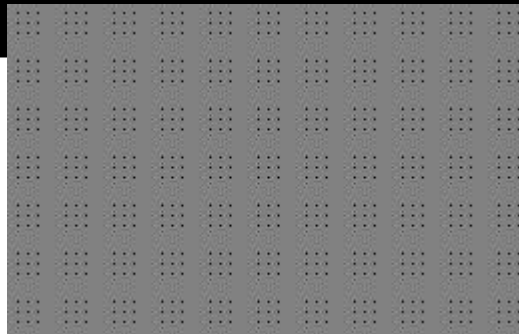
6

完成后的程序执行画面如下。

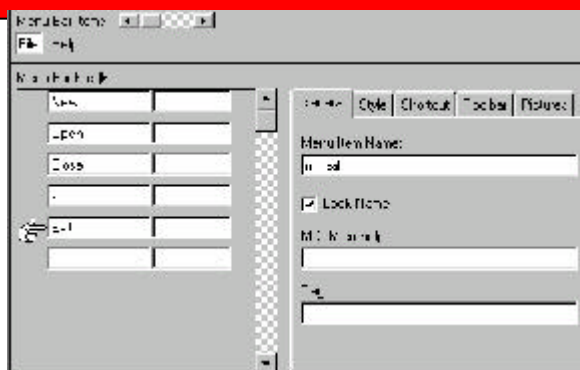


7

本例中介绍了在程序中添加和使用下拉式菜单和弹出式菜单的方法，用户可以充分看出菜单对于命令按钮的优点。

MDI 界面的应用 (一)
创建 MDI 框架**1****2**

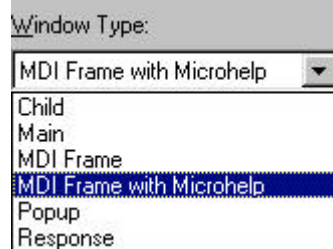
建立一个新的菜单，菜单中应至少有 New、Open 和 Close 等选项，并将菜单保存。

**3**

在窗口的属性窗口中，将刚刚建立的菜单指定给新的窗口。



4



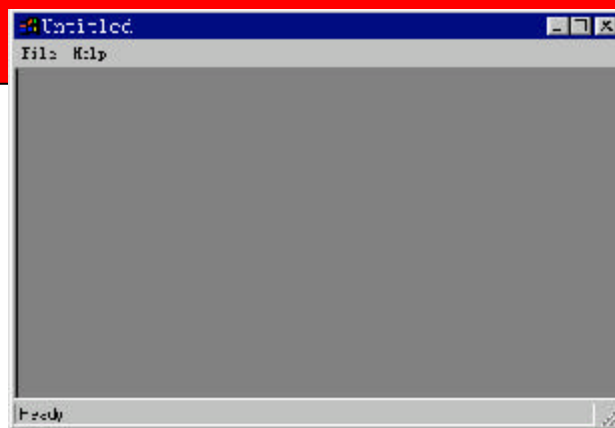
5

将窗口保存为 Frame，以便以后使用。



6

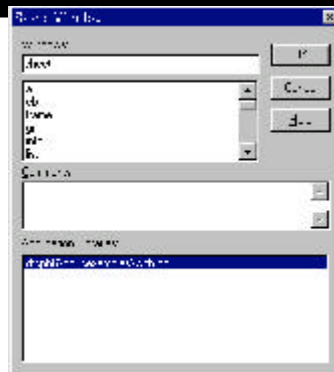
运行窗口，可以看到带状态栏的 MDI 框架。



7

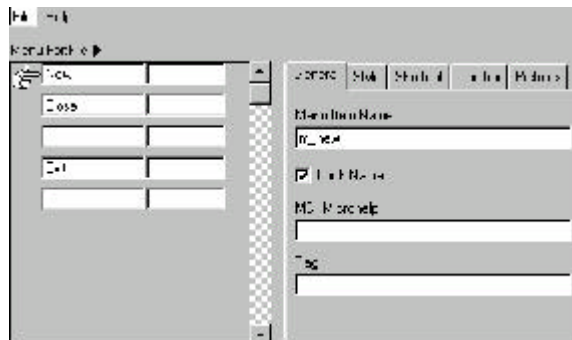
在本例中，介绍了创建一个 MDI 框架的方法，但是还没有介绍如何使之与工作表相结合，我们将在下一小节中介绍。

1



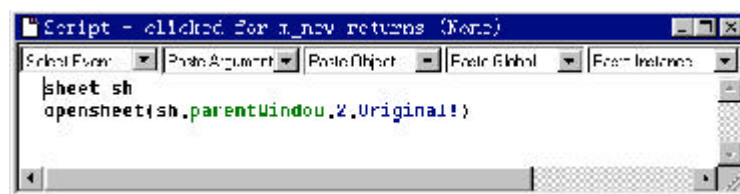
2

打开上例中保存的菜单，菜单项如下。

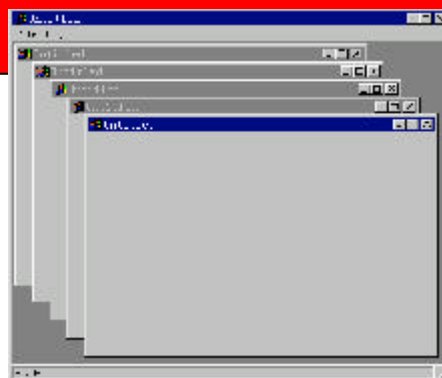


3

在 New 菜单项中加入如下代码，首先定义一个 Sheet 类型的窗口，然后作为工作表打开它。

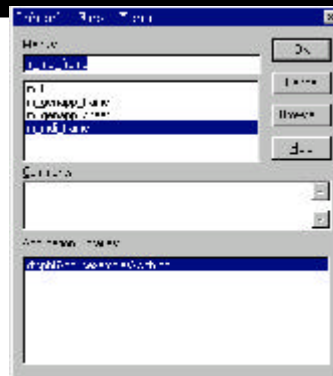


知识 



在本例中，介绍了建立和关闭工作表的方法，用户可以在工作表中加入各种控件，实现所需要的功能。

1



2

在继承的菜单中，原有的菜单项不能改变，且名称都被用灰色矩形框填充。



3

在原有的菜单基础上添加两个新的菜单栏 Edit 和 Window，新添加的菜单项如下图所示。



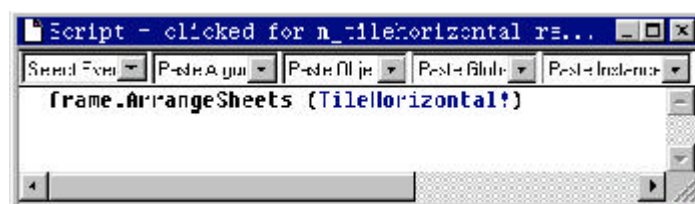
4

知识



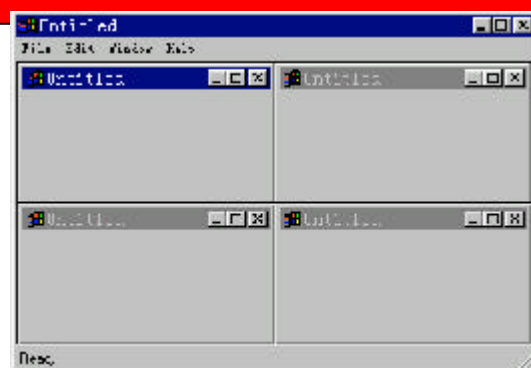
5

在 Window 菜单栏的各个菜单项的 Script 中，利用 ArrangeSheets 函数完成各种工作表的排列方式。



6

将菜单保存为 m_mdi_sheet，并指定给工作表，下图为打开工作表后的 MDI 界面。



7

本例中介绍了一个继承菜单并添加新选项的方法。在 MDI 应用程序中，不但可以在框架菜单的基础上添加新的菜单栏，还可以在原有的菜单栏中添加菜单项。



[返回总目录](#)

第四章

数据窗口对象

本章导读

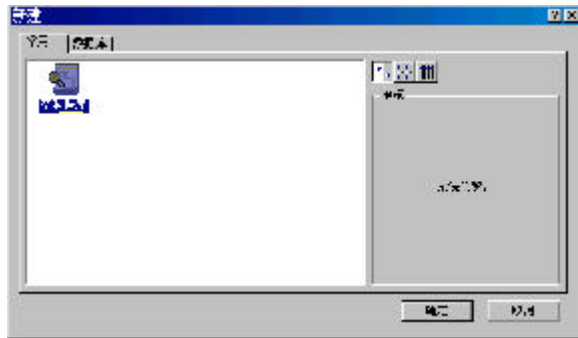
数据窗口 (Data Window) 是 Power Builder 程序向用户展示数据库数据并允许用户修改数据的窗口。数据窗口是在程序中浏览和修改数据最常用的方式，它使程序与用户的交互变得更加容易。由于数据窗口几乎是 Power Builder 中最重要的对象，本书将从各个方面讲述之。本章内容包括：数据窗口的表现方式 (Grid、Tabular、Freeform 等)、数据窗口的数据获取方式 (Quick Select、SQL Select 等)、数据窗口的定制方法 (如何为数据窗口增加文本对象、计算对象、图表对象、命令按钮等)。本章的重点在数据窗口的定制，即如何产生一个界面友好功能强大的数据窗口。在本章的后面几节是两个创建总体性数据窗口的例子。

例

准备工作 创建数据库 student.mdb

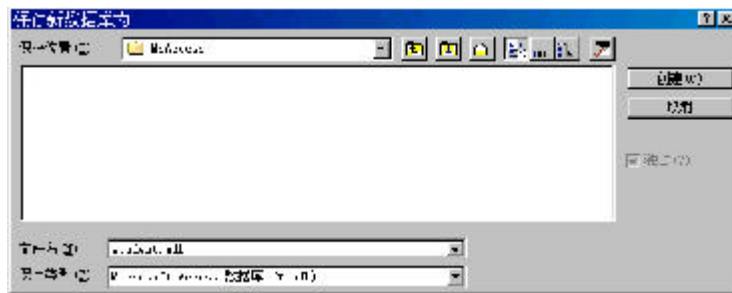
解

1



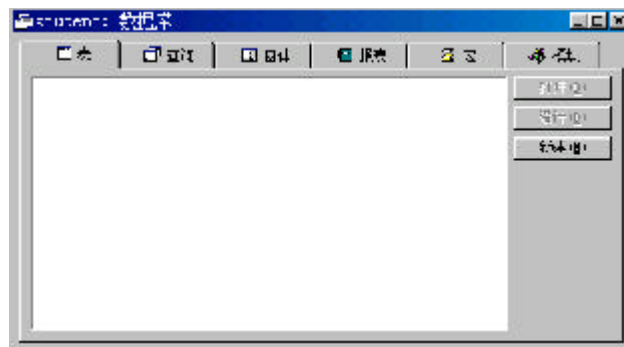
2

选“常用”标签，选中“空数据库”项，单击“确定”，弹出“保存新数据库为”对话框。

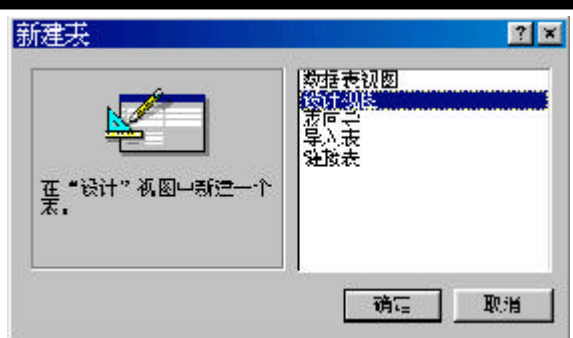


3

为新建的数据库选择路径，然后在“文件名”文本框中输入 student.mdb，单击“创建”按钮，新建的数据库如下图。



4



5

选择“设计视图”，然后单击“确定”，进入表的设计窗口。照下图填写“字段名称”一栏和在“数据类型”一栏中进行选择。将“说明”一栏中的内容写在窗口下面“常规”标签里相应的文本框内。

Table_Primary: 表			
字段名称	数据类型	说明	
studentID	数字	格式长整型，小数位自动，默认值0，必填	
name	文本	字段大小30，必填	
sex	文本	字段大小10	
department	文本	字段大小50	
birthday	日期/时间	格式为短日期	
address	文本	字段大小50	
telephone	文本	字段大小20	
email	文本	字段大小50	

字段属性

6

将表的名字保存为 Table_Primary，这样就成功地创建了一个表。照同样的办法再创建一个名为 Table_Courses 的表，它的字段见下图。选中某一行单击工具条上钥匙状的“主关键字”按钮，可以设置主关键字。

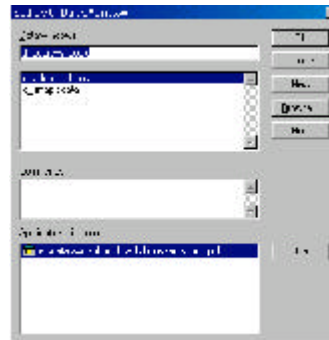
Table_Courses: 表			
字段名称	数据类型	说明	
ID	自动编号	长整型，新值递增	
courseID	数字	长整型，小数位自动	
studentID	数字	长整型，小数位自动	
score	数字	整型，小数位自动	

字段属性

7

在“表”标签下双击表的名字，为已经建好的表添加数据。

1



2

选择一个数据窗口，按 OK 按钮即可打开之。若要新建一个数据窗口，单击对话框的 New 按钮，出现如下 New DataWindow 对话框。



3

数据源中的 Data Source 和 Presentation Style 分别表示数据窗口对象获取数据的方式和数据窗口对数据的表现方式，这在以下几节将讲述。选中 Quick Select 和 Freeform，按 OK 按钮，得如下对话框。

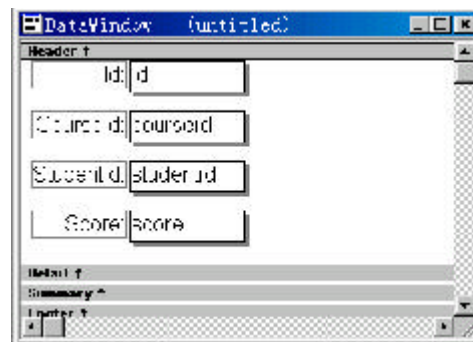


4



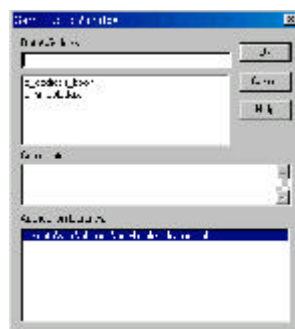
5

按 OK 按钮确认，出现 DataWindow 窗口（见下图）和 DataWindow 画板可以看到被选中的字段出现在 DataWindow 的显示中。



6

单击数据窗口画板上的 Save 按钮（一个软盘形状）或数据窗口的关闭按钮再在弹出的对话框中选“是(Y)”弹出如下 Save DataWindow 对话框。

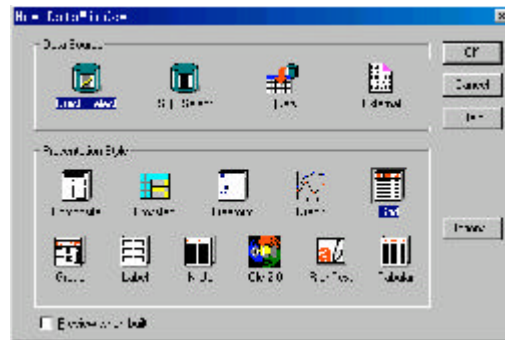


7

在 Data Windows 文本框中输入要保存的数据窗口的名字（按照惯例，数据窗口的名字以“d_”打头），点击 OK 按钮，则数据窗口保存在当前的 Application Library 中。

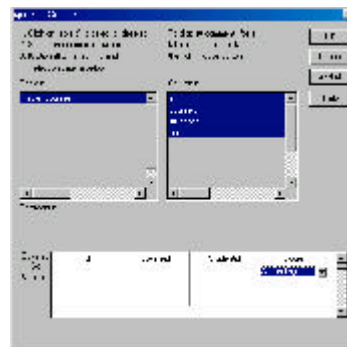
数据窗口的表现方式 表现方式实例

1



2

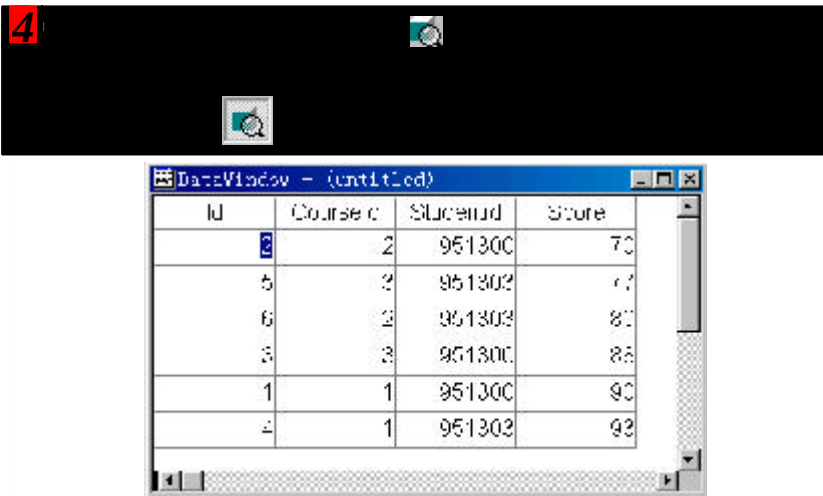
按 OK 按钮，得如下 Quick Select 对话框。仍然选 table_courses 表并选择它的所有列（即字段），在 Comments 中将 Score 的排序（Sort）设为 Ascending（如图）。



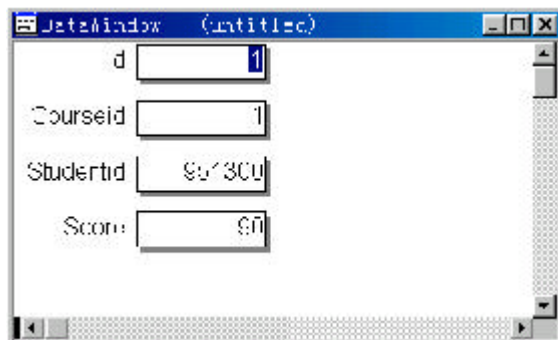
3

按 OK 确定，进入 Data Window 的设计窗口，见下图。以后我们会提到如何利用这个窗口进行 Data Window 的定制。

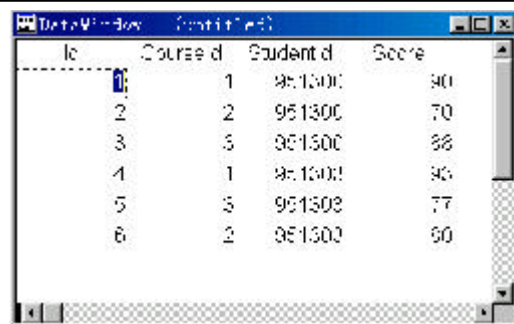
DataWindow - (entitled)			
d	Courseid	Studentid	Score
Header ↑			
d	courseid	studentid	score
Detail ↑			
Summary -			
Footer ↓			



5 打开在上一节所建立的数据窗口（方法见本节知识点的解释），用与步骤 4 相同的方法进行预览，见下图。这就是数据在 Data Window 中的 Freeform 显示方式。

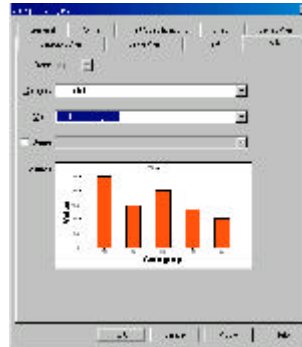


6 在 New DataWindow 对话框（步骤 1）中，将 Presentation Style 设为 Tabular 按照步骤 2、3、4 对数据窗口进行预览，结果如下图。



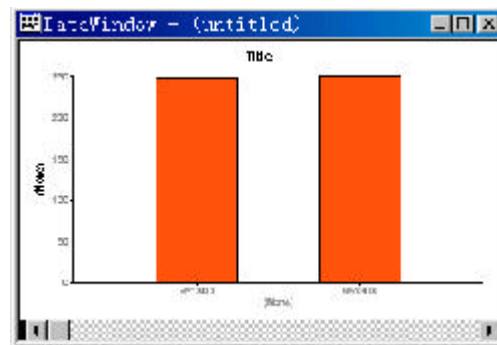
7 注意 Tabular 方式与 Grid 方式类似，只是 Grid 方式有网格。下一节将介绍数据窗口的 Crosstab、Graph 和 Group 表现方式。

1



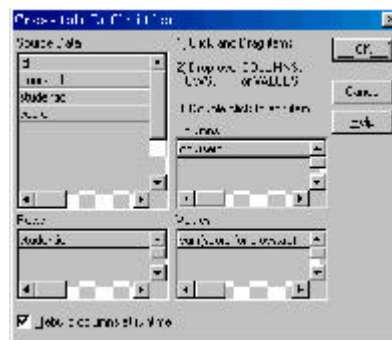
2

按 OK 按钮，进入 Data Window 的设计窗口，按上一节的方法预览数据窗口，得下图。图中柱状图显示了 951300 和 951303 两个学生三门课程的总成绩。想一想，如果要显示学生功课的平均成绩该如何设置？

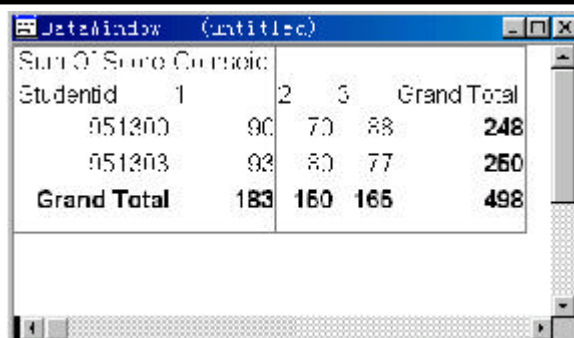


3

在 New DataWindow 对话框(步骤 1)中,将 Presentation Style 设为 Crosstab. 按 OK 按钮,弹出如下 Crosstab definition 对话框。将 Source Data 中的 studentid 拖入 Rows, courseid 拖入 Columns, score 拖入 Values。



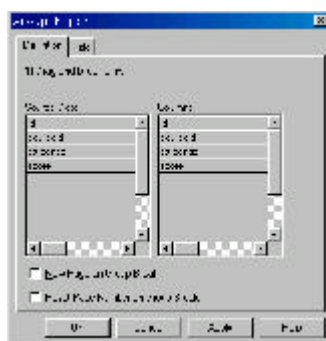
4



Studentid	1	2	3	Grand Total
951300	90	70	88	248
951303	93	80	77	250
Grand Total	183	160	165	498

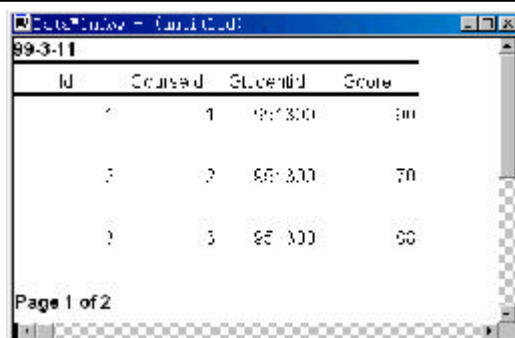
5

在 New DataWindow 对话框（步骤 1）中，将 Presentation Style 设为 Group。按 OK 按钮，弹出如下 Group Report 对话框。将 Source Data 中的各项拖入 Columns。



6

按 OK 按钮，进入 Data Window 的设计窗口，预览数据窗口结果见下图。图中显示是第 1 页，按滚动条浏览其它页。



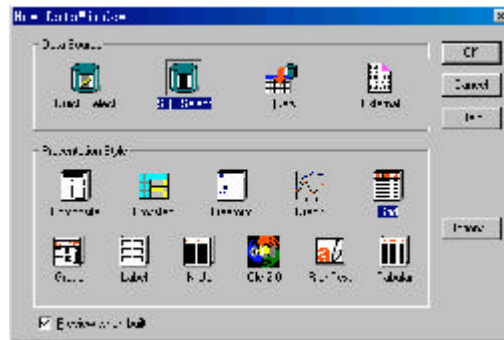
id	Courseid	Studentid	Score
1	1	951300	90
2	2	951300	70
3	3	951300	88

Page 1 of 2

7

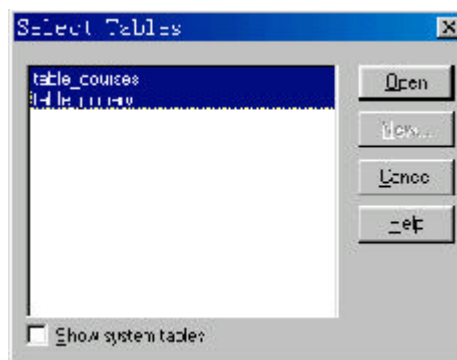
单击 Data Window 的设计或预览窗口的关闭按钮，退回到 Power Builder 主窗口。数据窗口的其它数据显示方式（Label、N-Up、RichText、Ole2.0、Composite）请读者用几乎同样的步骤进行试验。

1



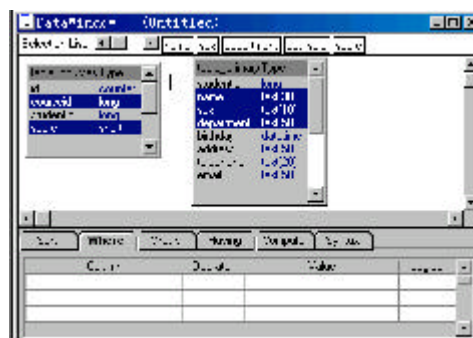
2

在上图对话框中，将 Data Source 设为 SQL Select；Presentation Style 设为 Grid。按 OK 按钮，弹出如下 Select Tables 对话框。选中 table_courses 和 table_primary 两个表。

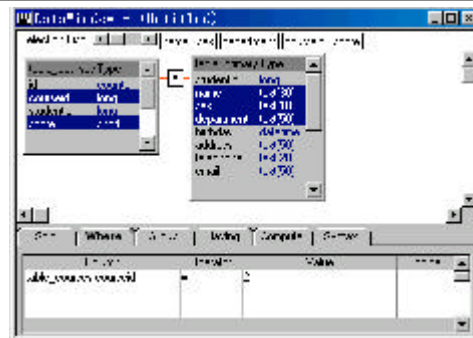


3

按 Open 按钮打开表，进入 Visual SQL Select 画板，如下图。在两个表窗口中单击将来要在数据窗口中出现的列名，这些列名显示于主窗口上部的 Selection list 中，如图。本例要显示所有学生某一门功课的成绩。

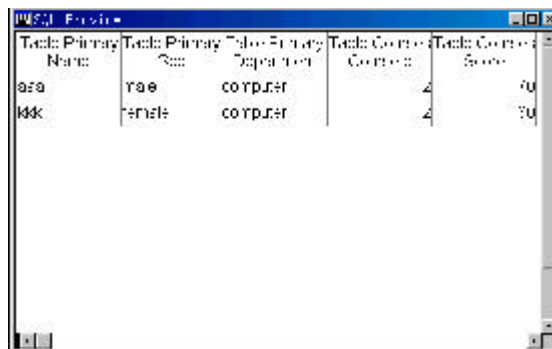


4



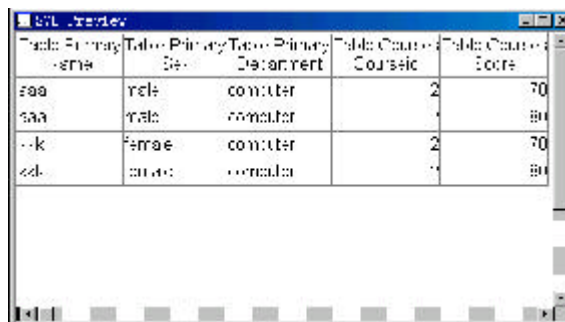
5

单击 Painter Bar 上的“Preview”按钮，预览数据窗口的数据如下图。注意到工具条上的“Insert Rows”按钮变灰，表示不能添加数据。按工具条上的 Close 按钮关闭预览。



6

我们来看步骤 3 两个表窗口之间的等号“=”有何作用。单击等号，在弹出的对话框中按 Delete 按钮，然后按步骤 5 的方法预览数据窗口，见下图。

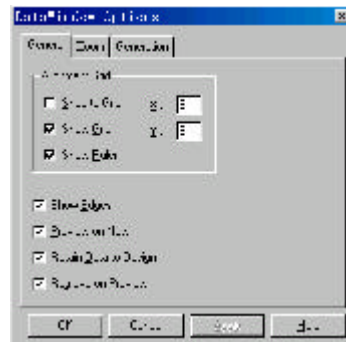


7

上图预览显示了 4 条记录，并且前两条记录和后两条分别矛盾（一个学生的同一门课不可能有两个分数）。这就是从两个表中“Select”数据时应该注意表的连接问题。

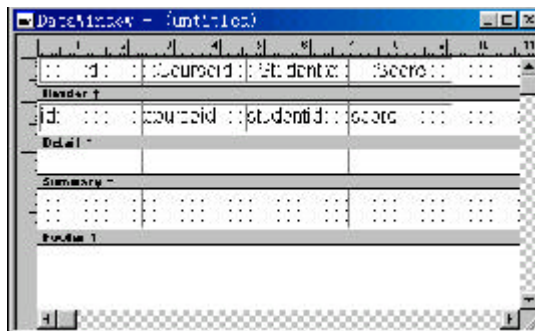
设计选项和属性
对话框标签页介绍

1



2

选中“Show Grid”和“Show Ruler”复选框，按 OK 或 Apply 按钮发现 Data Window 的编辑窗口增加了网格和标尺，如下图。改变上图标签页中 X 和 Y 的值，按 Apply 按钮，看网格有何变化。

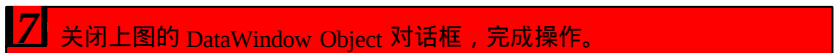
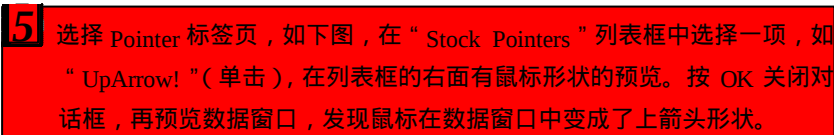


3

在步骤 1 的对话框中选 Zoom 标签页如下图。它用来设置数据窗口的视图大小。选择最下面的 Custom 可以定制缩放的倍率。但编辑数据窗口只能在 100% 的状态下。本对话框的 Generation 标签页见本节的解释。

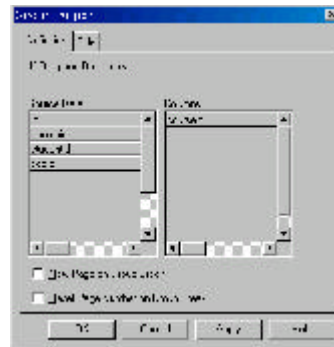


知识



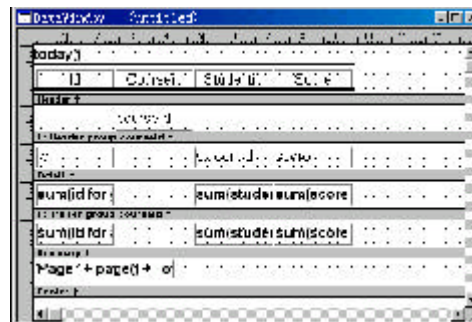
设计窗口的区域划分 区域划分概述

1



2

在上图对话框中通过拖动的方法指定用来分组的列，如上图已经把“courses”拖进 Columns 列表框，表示以此列分组。按 OK 进入下图的数据窗口设计主窗口（若在预览模式，按照前几节的操作退回设计模式）。

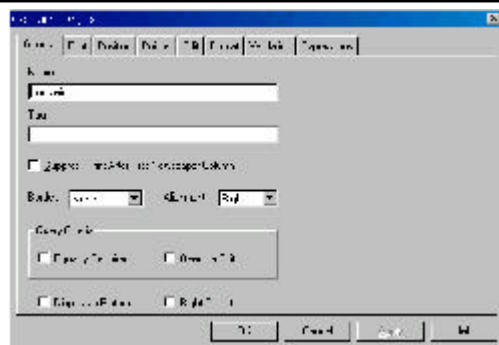


3

注意到设计窗口被分成了 header、footer 等区。选中 header（页眉区）中的列名（如 id），单击鼠标右键弹出如下对话框，由对话框标题可以看出这是一个文本对象。甚至可以在 text 编辑框中修改文本的显示。

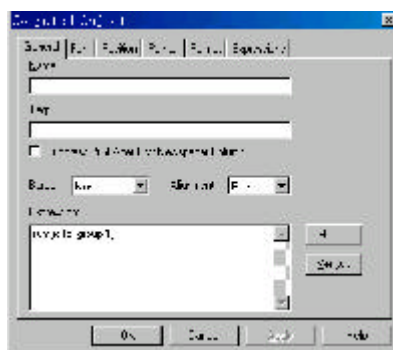


4



5

右键单击 Trailer group (组尾区)、Summary (汇总区) 和 Footer (页脚区) 中任意一个对象, 弹出下图对话框。由对话框标题可以看出这些对象是计算对象 (Computed Object)。



6

预览此数据窗口 (通过单击工具条的 Preview 按钮), 如图。读者可以对照设计窗口和预览体会以下设计窗口中的对象与预览中数字 (或字符串) 的对应关系。

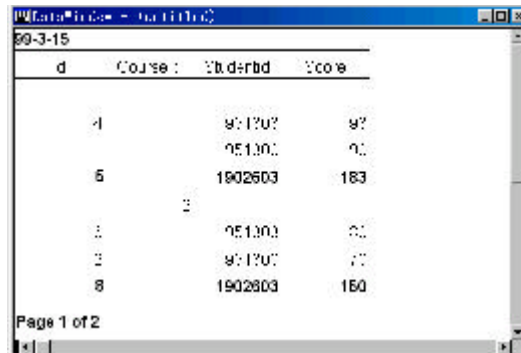
ID	Name	Age	Salary
2	Wang	25	1500
1	Li	28	1200
5	Zhang	30	1800
3	Chen	22	1000
4	Wu	26	1300
6	Yang	29	1600
7	Xu	24	1100
8	Guo	31	1900

7

在以后的几节中会讲到如何利用设计图中各种类型的对象来定制数据窗口。在本节, 读者只需要了解一下 DataWindow 设计窗口的区域构成即可。最后保存此数据窗口, 以备下一节用。

定制数据窗口
增加和编辑文本对象

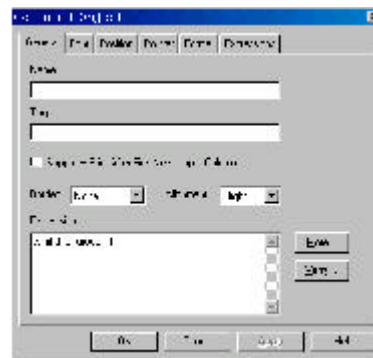
1



d	Course	StudentId	Score
4		901207	97
		951300	90
5		1902603	183
3		951300	90
3		901207	77
8		1902603	150

2

回到设计窗口，选定组尾区最左边的对象，双击得到它的属性对话框，其下部 Expression 编辑框中有一个表达式，如下图。此式表明对本组的所有“Id”列的值求和。删除组尾区左边的两个对象，再预览窗口。

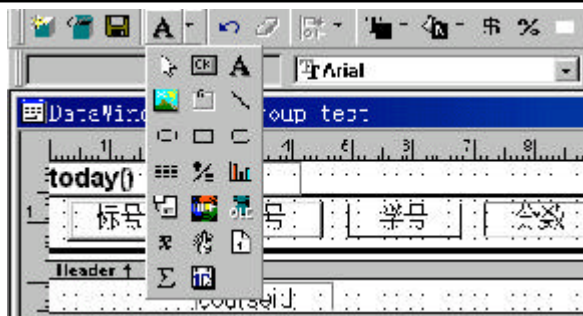


3

选定页眉区的 courseid 文本对象，双击或按回车键进入其属性对话框，选 General 标签页。将 text 编辑框中的字符串改为“课号”；再按照下图在对话框中设置文本对象的 Border 和 Alignment。

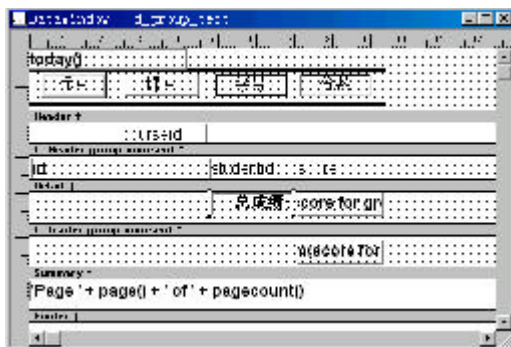


4



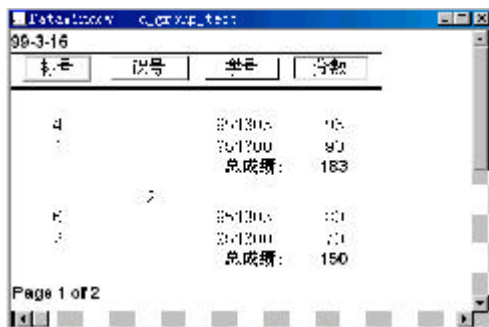
5

单击上图工具栏上的 text 按钮（字母 A 形状的图标），在组尾区对象左侧的空白处单击，则将新的文本对象添加到此处。按照步骤 3 的操作将它所显示的文本改为“总成绩”，完成后的设计窗口如下图。



6

若“总成绩”三个字不能完全显示，调整对象所占空间的大小（方法见解释）。在属性对话框中将所有对象（包括文本对象、计算对象、列对象等）的 Alignment 都设为 center。最后预览数据窗口，如下图。



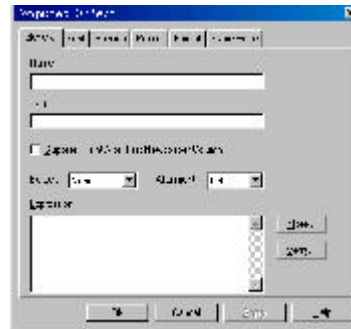
7

仔细观察步骤 1 和步骤 6 的预览中，所显示的数字位置有何不同，这是所有对象的 Alignment 属性由 left 变为 center 的结果。

知识窗

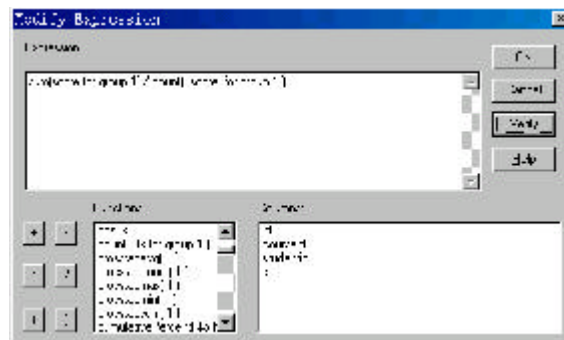
定制数据窗口
增加和编辑文本对象(续)

1



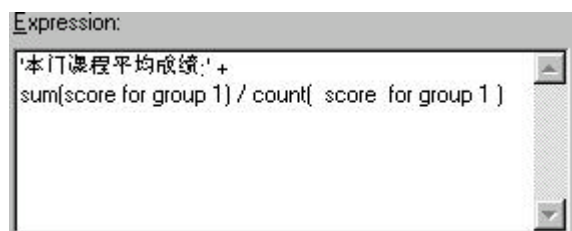
2

单击 more...按钮,弹出如下“表达式编辑”对话框。利用对话框下部的按钮、列表框在 Expression 编辑框中编辑和创建表达式,如下图。其中函数 sum 和 count 是从 Functions 列表框选择出来的。



3

按 OK 按钮,步骤 1 对话框中的 Expression 编辑框中出现了上步所创建的表达式。为了清楚地表达含义,将表达式写成下图的样子。修改后的表达式告诉用户所显示的是课程的平均成绩,体现了界面友好性。



4

知识窗



5

其实 Select Control 已经提供了几个缺省的计算对象，其中包括计算组内某列几个字段值的平均数。下图是 Power Builder 缺省的数据窗口计算对象。



6

在编辑计算对象的表达式时，下图所列出的函数（或叫表达式）是比较有用的。

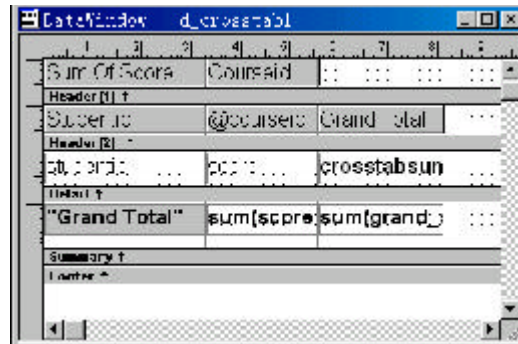
表达式	功能
today()	当天日期
page()	当前页码
pagecount()	总页数
count(#x for group 1)	本组#x列的数目（其中“#x”表示列名）
sum(#x for group 1)	本组#x列值的和
avg(#x for group 1)	本组#x列值的平均数
max(#x for group 1)	本组#x列的最大值（同理 min 表示最小值）
rowcount()	（各组）记录的总行数

7

由上图可见，步骤 2 的表达式可用 avg 函数来代替。读者可以自己添加一些计算字段，体会一下上图所列函数的功能。

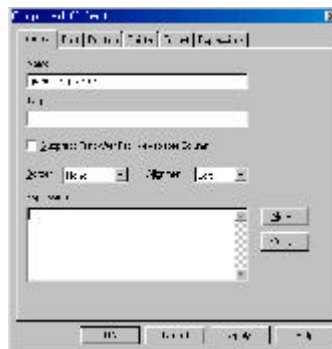
定制数据窗口 增加和编辑计算字段

1



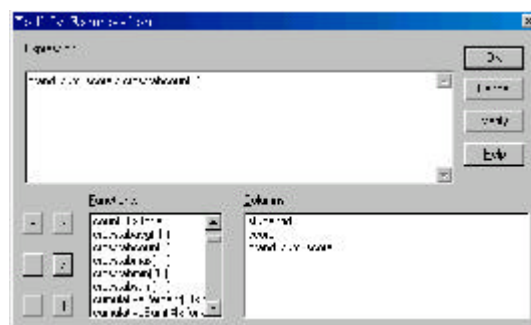
2

在细节区各个对象的最右边新建一个计算对象（计算每一个学生的平均分）作为计算字段，弹出如下对话框。在 Name 编辑框输入此计算字段的名字：grand_avg_score。

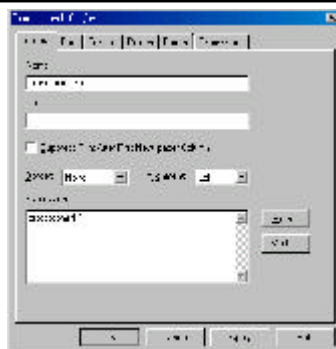


3

单击 More... 按钮，进行表达式的编辑，见下图。表达式中的“grand_sum_score”是计算学生总分的计算对象的名字；函数 crosstabcount(1) 可以从下部的函数列表中找到，表示每一行的课程数。

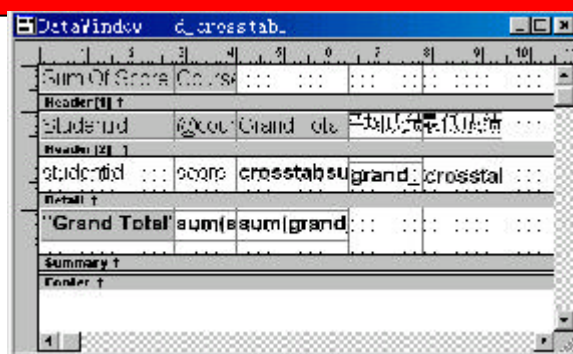


4



5

在 Head2 区加入两个文本对象，以便提示用户所显示的数据分别是各门课程的平均分数和最低分数。完成的设计窗口如下图。



6

预览此数据窗口的结果如下图。可以调整各列的宽度以取得最佳的预览效果。方法是移动鼠标到列边界上，等光标变为双箭头时左右拖动。

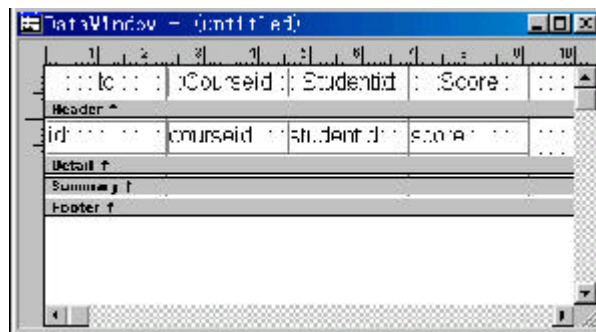
Sum Of Score	Course	Grand Total
90	1	245
82	2	239
183	3	498

7

退回设计状态，保存此数据窗口以备用。此次操作完成。

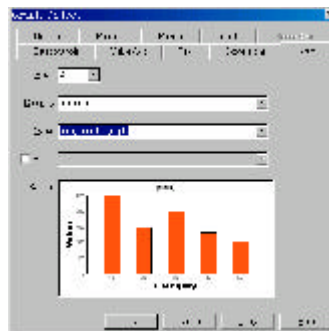
定制数据窗口 增加和编辑图表对象

1



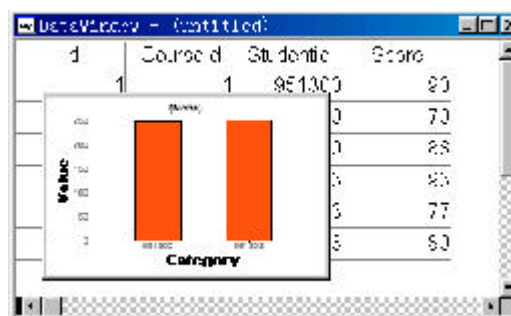
2

通过 Select Control 工具板添加一个 Graph 对象，弹出其属性对话框，如下图所示。在对话框 Data 标签页进行如图设置：在 Category 中选 studentid，Value 中选 sum(score for graph)。

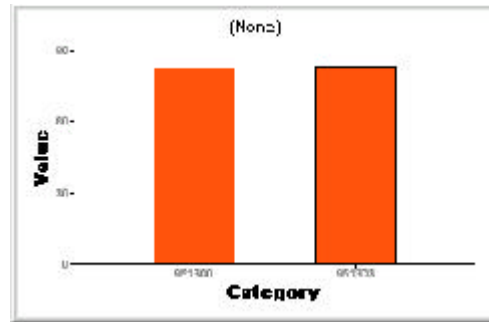


3

按 OK 关闭对话框，回到设计状态。拖动表的边缘使其放大，然后预览数据窗口如下图所示。在预览状态下可以拖动图表，看起来图表好像是浮在数据窗口的表面上的。

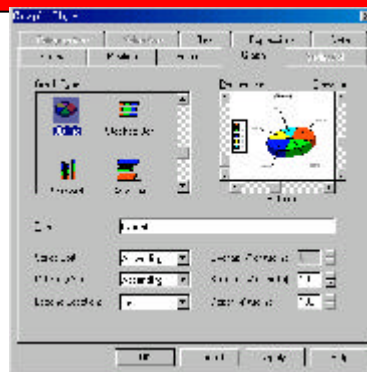


4



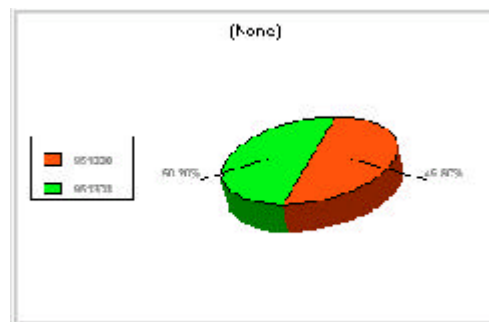
5

选择图表属性对话框的 Graph 标签页，如图，Graph Type 中列出了可选的图表类型。选择 3D Pie（三维饼）。



6

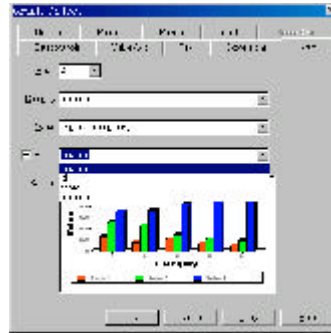
再次预览数据窗口，用三维饼表示的两个学生的平均成绩对比见下图。



7

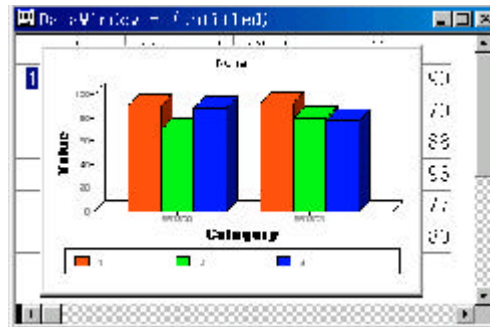
读者可以实验其它图表类型的效果，如 Stacked Bar、3D Area、Scatter 等。最后请考虑一下图表属性对话框 Data 标签页的 Series 复选框有何功能，下一节我们会提到它。

1



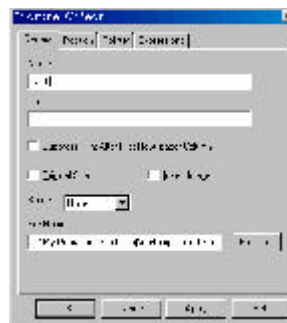
2

完成上面的设置后预览对话框如下图。图中用不同的颜色列出了每个学生各门课程的成绩。退出预览，在设计状态下删除此图表对象。



3

在 Select Control 中选择 Picture，将一个图象对象添加在细节区最右边。图象对象的属性对话框如下图。在 Name 中输入对象的名字（随意），单击“File Name”编辑框右面的 Browse...按钮，选择图象文件。

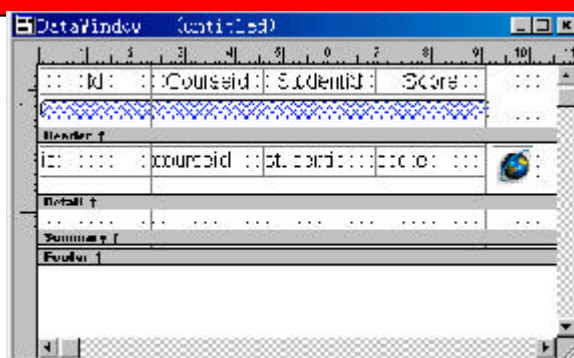


4



5

关闭上图对话框，调整 Picture 和 Round Rectangle 对象的位置，见下图。



6

预览所完成的数据窗口，发现每一列数据的右面都有一幅图，这是因为步骤 3 将新建的 Picture 对象放在了细节区。

ID	Courseid	Studentid	Score
1	1	051300	90
2	2	051300	70
3	3	051300	88
4	1	051303	96
5	3	051302	77

7

关闭和保存此数据窗口，完成操作。

1



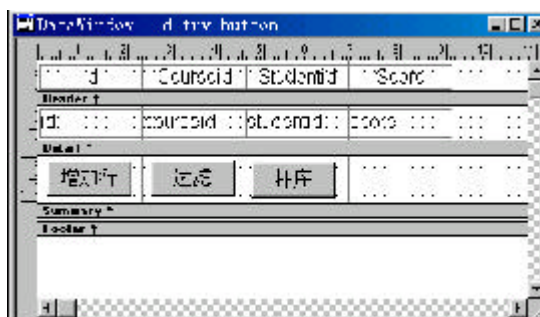
2

通过 Select Control 工具板在汇总区添加一个 Button 对象。在其属性对话框 General 页进行如下设置：在 Text 编辑框输入“增加行”；在 Action 下拉列表框选择“Append Row”；在 Name 编辑框输入名字（随意），如图。



3

按同样操作过程在汇总区再加两个按钮对象，Text 分别为“过滤”和“排序”；Action 分别为 Filter 和 Sort。完成后的设计窗口如下图。



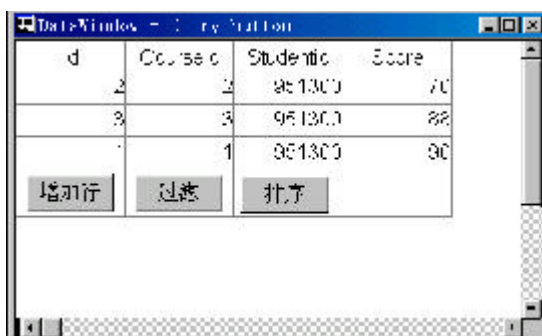
4



d	Course	Student	Score
2	2	951300	70
3	3	951302	77
6	2	951302	80
3	3	951300	88
1	1	951300	90
4	1	951302	95

5

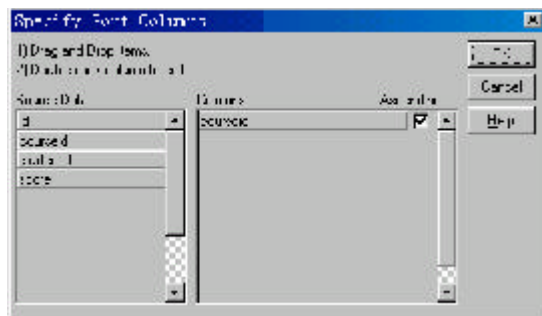
单击新加的“过滤”按钮，弹出编辑过滤表达式的对话框，在其中输入一个表达式“studentid = 951300”(有关数据的过滤和排序见本书第一章)，按 OK 按钮回到预览窗口如下图。



d	Course	Student	Score
3	3	951300	88
1	1	951300	90

6

单击新加的“排序”按钮，弹出下图对话框，将 Source Data 中的 courseid 拖到 Columns 列表框，并复选 Ascending，如图。我们希望数据按课号排序。



7

按 OK 按钮回到预览窗口，发现数据已经按照课号从小到大排序了。关闭和保存数据窗口，操作完毕。

定制数据窗口 设置对象的通用属性

1



2

选取上面属性对话框的 Font 标签页，在这儿设置字体。在对话框中设置 14 号的斜体的“@文鼎中书简”字体，如图。关闭对话框，并按照步骤 1、2 设置页眉区所有文本对象的属性。

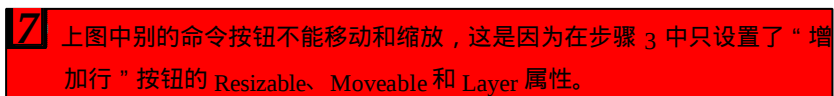
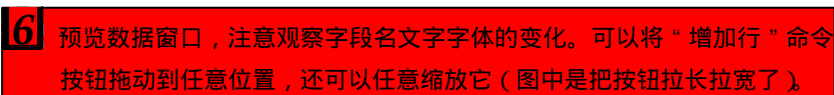
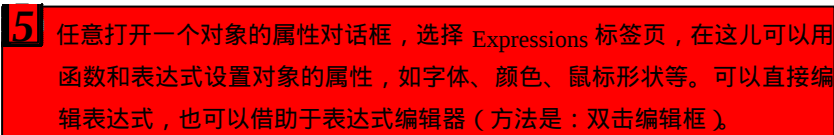


3

从汇总区选择一个命令按钮对象（如“增加行”按钮），按照下图对其属性对话框的 Position 标签页进行设置：复选 Resizable 和 Moveable；Layer 选择 Foreground。关闭对话框。

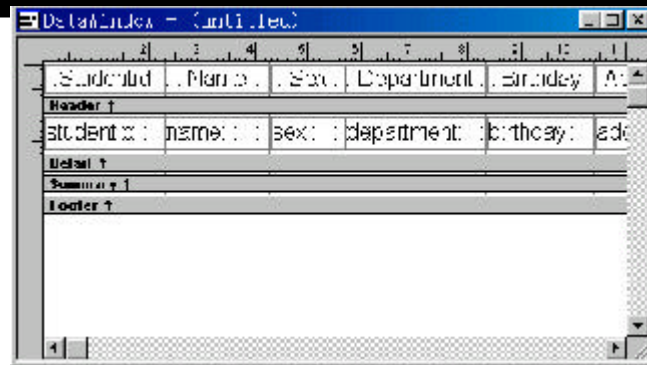


知识



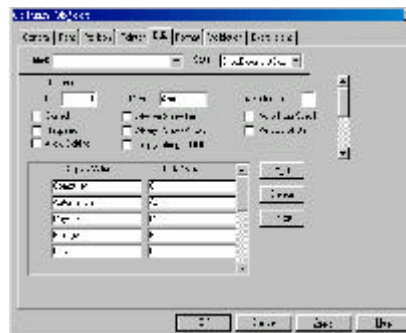
定制数据窗口
设置列的特殊属性

1



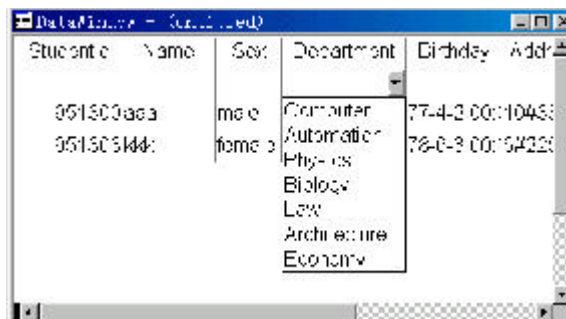
2

要想改变列 department 的编辑格式, 选择其属性对话框的 Edit 标签页, Style 选择 DropDownListBox, 并在左下的列表框中增加系名的显示值 (Display Value) 和存储值 (Data Value)。

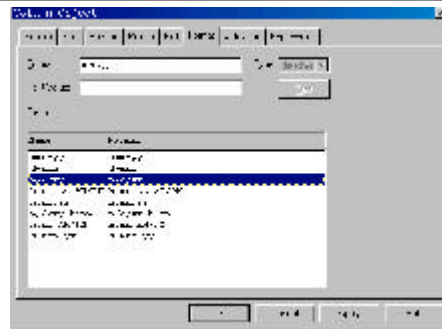


3

按 OK 按钮关闭属性对话框。预览改变后的数据窗口, 试着增加一行, 发现 Department 字段的输入方式变成了下拉式列表框, 只需要再选择一个系的名字即可。

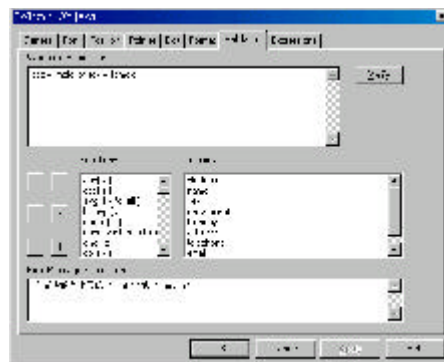


4



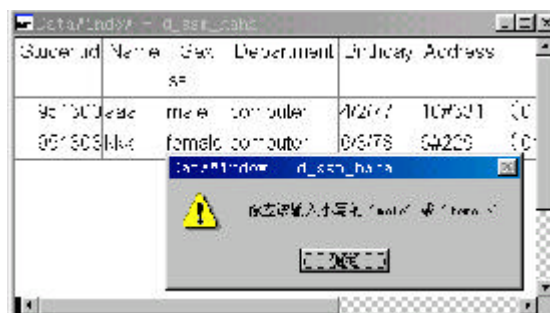
5

打开 Sex 列属性对话框，选择其 Validation 标签页，如图。在 Validation Expression 编辑框中输入 “sex = 'male' or sex = 'female'”；在 Error Message Expression 输入 “你应该输入小写的 ~male~ 或 ~female~”。



6

关闭上图对话框。预览数据窗口，试着增加一行，并在 Sex 列中输入 “ss” 当试图将鼠标挪到别的格子中时，弹出一个提示对话框，这说明输入的数据不符合有效性检查的条件，应该重新输入。

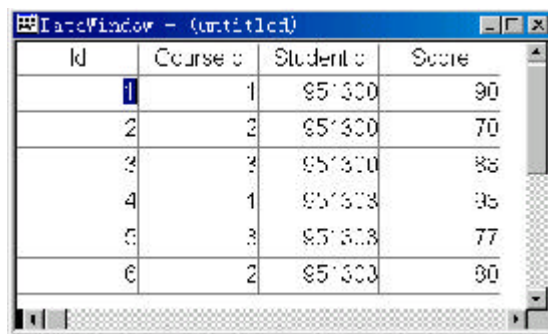


7

注意到上图预览中，Birthday 列的数据变为 m/d/yy 格式，那个讨厌的 “00:00:00” 消失了。

数据窗口预览中的操作 数据的过滤、排序与保存

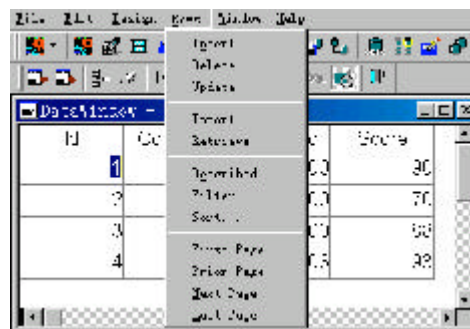
1



Id	Course	Student	Score
1	1	951300	90
2	2	951300	70
3	3	951300	88
4	1	951308	95
5	3	951308	77
6	2	951300	90

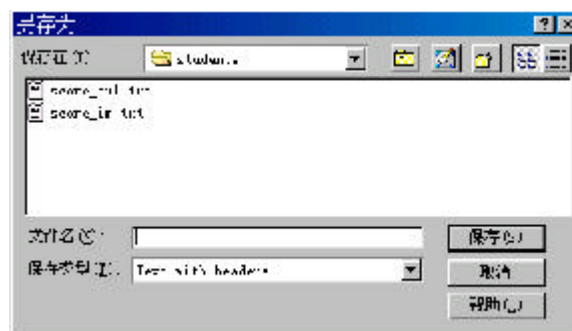
2

单击 Rows 菜单，如下图。可以看出，在预览窗口中可以进行四种类型的操作。

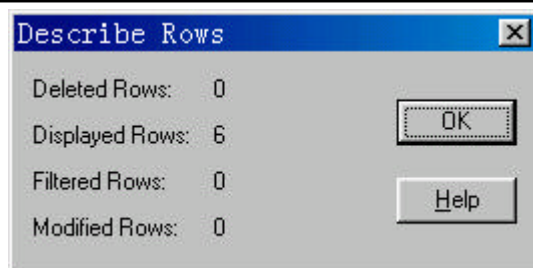


3

选择 File|Save Rows As...菜单项，弹出如下对话框。在“保存类型”中选择保存格式；在“文件名”中输入目标文件的名字；按 OK 即完成保存。

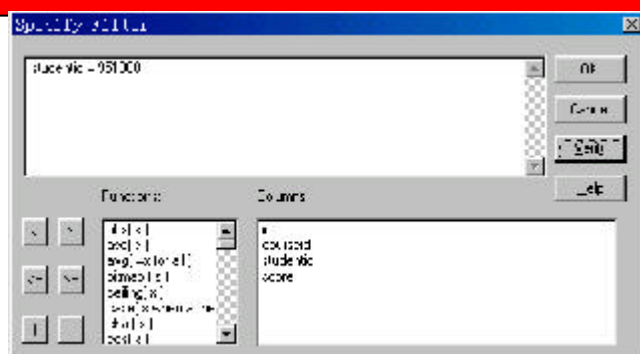


4



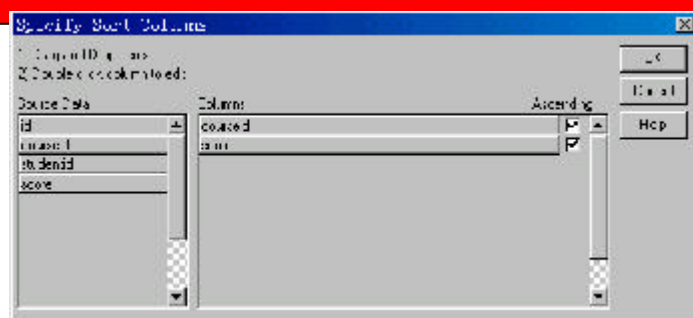
5

选择 Rows|Filter 菜单项弹出如下对话框，在这儿编辑数据过滤表达式，如输入 " studentid = 951300 " 则只显示 studentid 为 951300 的行。



6

选择 Rows|Sort...菜单项弹出如下对话框，如下图。图中设置数据窗口中的数据按照 courseid 排序，若 courseid 相同则按照 score 排序。

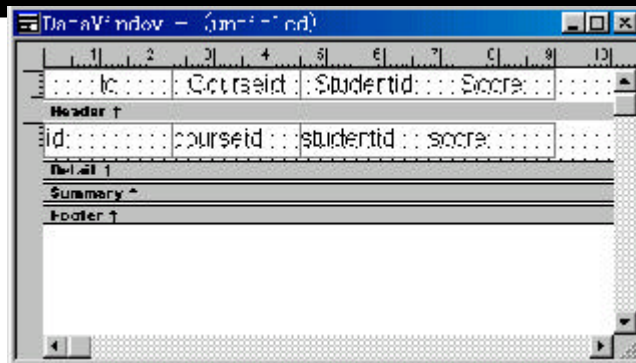


7

关闭预览，完成本次操作。

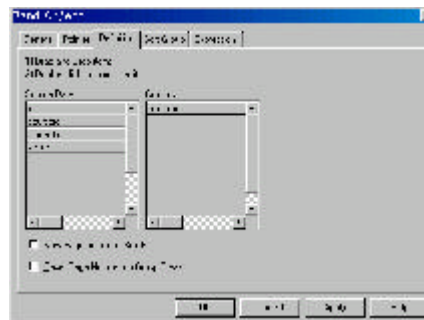
加入分组功能 分组的操作

1



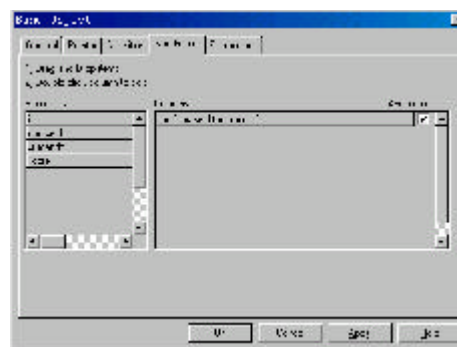
2

我们要为这个 Tabular 表现方式的数据窗口加入分组功能，在设计（而不是预览）窗口中选择 Rows|Create Group 菜单项。在其 Definition 标签页中设定按列 courseid 分组，如下图。

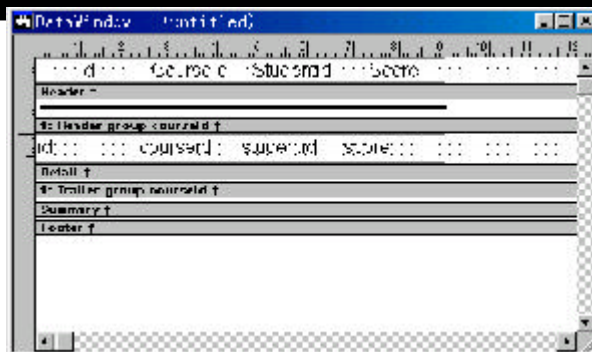


3

在上图对话框的 Sort Group 标签页中将 courseid 从 Source Data 列表框拖进 Columns 列表框并核选 Ascending，表示按列 courseid 的值升序排列。



4



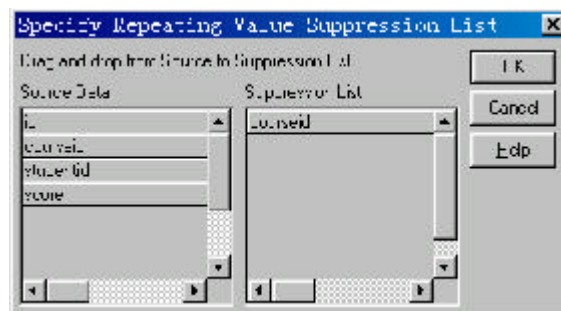
5

预览数据窗口，见下图。数据按照 Courseid 的不同值分成了几组，组和组之间用粗黑线分割。但一个缺憾是，每组的内部 Courseid 值相同，但这个相同的值仍然显示了多次，要是能在一组只显示一个课号值就好了。

id	Courseid	Studentid	Score
1	1	951900	80
4	1	951903	83
2	2	951900	70
6	2	951903	80
3	3	951900	85
5	3	951903	77

6

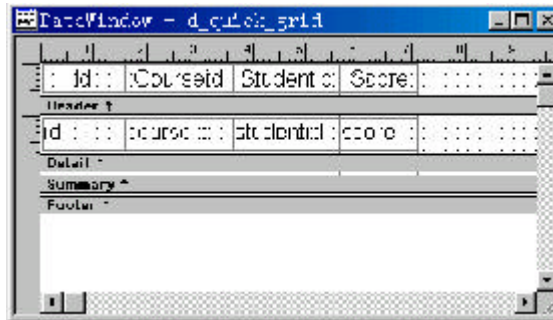
退回设计窗口，选择 Rows|Suppress Repeating Values...菜单项，弹出如下对话框。将 Courseid 拖进右侧的 Suppression List 列表框，表示屏蔽掉 Courseid 的重复值。



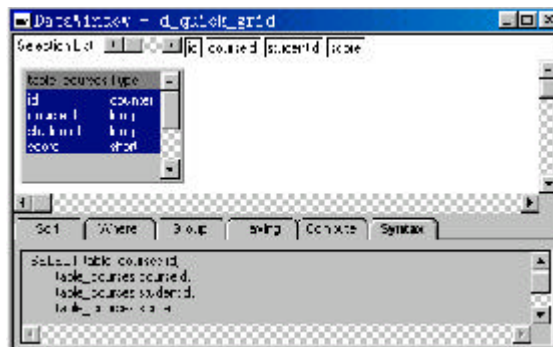
7

接着预览数据窗口，看是否达到了预期的效果。

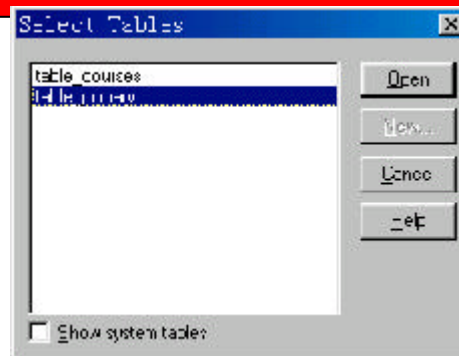
选取方式的统一性 Quick Select 与 SQL Select

1**2**

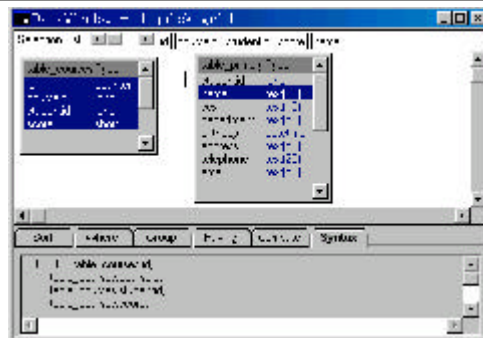
单击工具条上的 Data Source 标签按钮  或者选择 Design|Data Source 菜单项，出现下图窗口，这就是 SQL Select 数据源方式中的 Visual SQL Select 画板。

**3**

单击工具条上的 Tables 标签按钮 ，出现下图 Select Tables 对话框，选中 Table_Primary 单击 Open 按钮，把此表加入 Visual SQL Select 画板中。



4



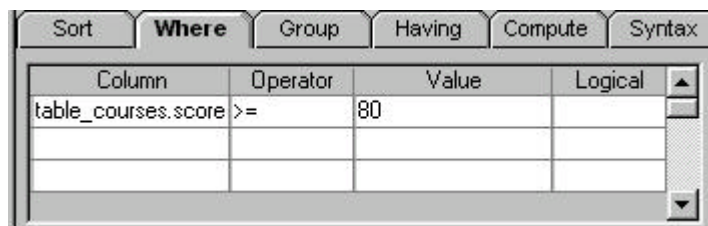
5

点击窗口下面的 Sort 页，在此设置排序条件如下图。图中设置先按课程号升序排列，再按学号降序排列。



6

在 Where 标签页设置查询条件（分数不小于 80 分）见下图。



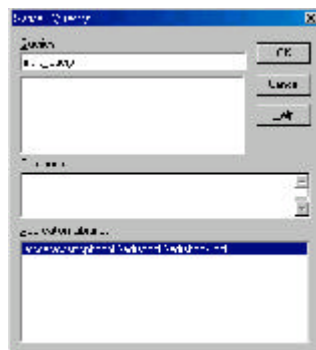
7

这样，我们把 Quick Select 选取方式改成了 SQL Select 方式。现在可以按工具条上的 Preview 按钮预览查询（而不是预览数据窗口）了。

知识窗

选取方式的统一性 (续) Quick Select 与 SQL Select

1



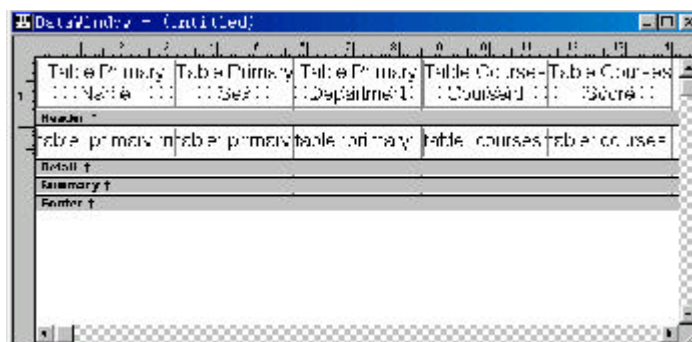
2

按照本章第 5 节的步骤创建一个 SQL Select/Grid 方式的数据窗口，其在 Visual SQL Select 画板中状态如图。

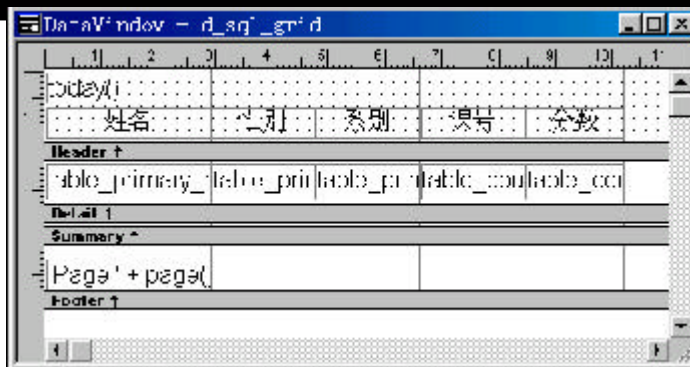


3

单击工具条上的 Data Source 按钮 (或选择 Design|Data Source 菜单项), 进入数据窗口的编辑画板, 如下图。在这儿可以定制数据窗口 (象在 Quick Select 数据窗口编辑画板中那样)。

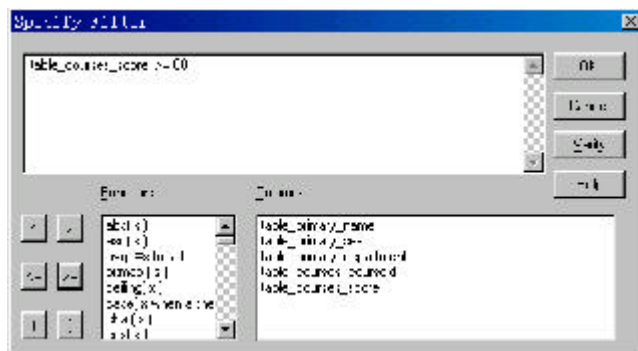


4



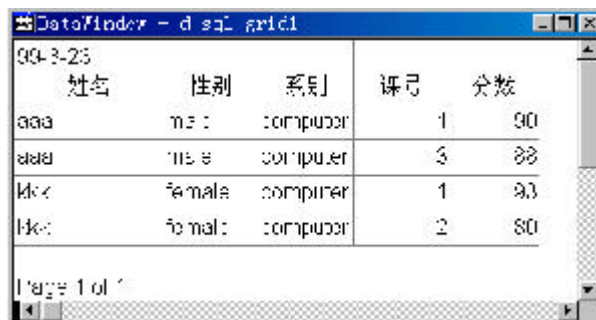
5

选择 Rows|Filter...为数据窗口设置过滤条件，如下图。与 Quick Select 操作步骤基本一样，唯一不同的是由于选择了多个表，所以在引用字段时要指明是哪个表的字段。



6

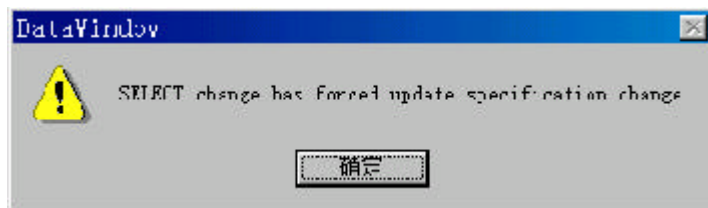
预览数据窗口，见下图。将数据窗口保存为 d_sql_grid1，完成本次操作。



7

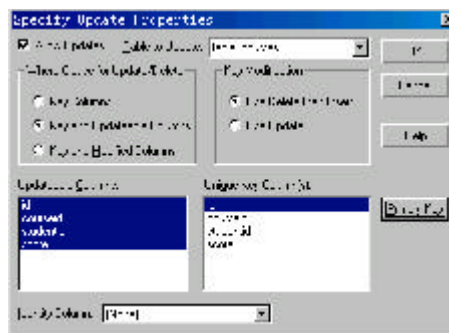
由步骤 4、6 可以看出，SQL Select 数据窗口和 Quick Select 数据窗口在定制方法、数据过滤、数据排序等各方面都是相同的。

1



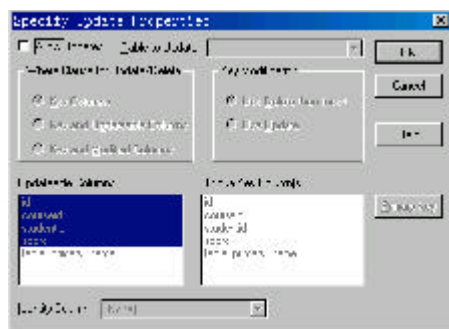
2

打开数据窗口 d_quick_grid，在它的编辑画板中选择 Rows|Update Properties 菜单项，弹出如下对话框。在这儿可以浏览和设置数据窗口的更新属性。注意到左上角的 Allow Updates 复选框是选中的，表示允许更新。

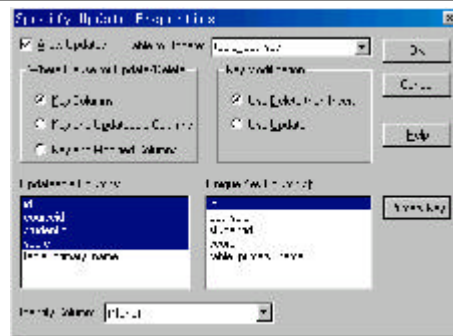


3

打开数据窗口 d_quick_grid2，其更新属性对话框如下图。它的 Allow Updates 复选框没有被选中，表示数据窗口不能更新表。



4



5

按 OK 关闭更新属性对话框，预览数据窗口如下图。发现工具条上的 Insert Row 按钮变成了可用的（而在步骤 4 设置更新属性对话框之前它是不可用的）。单击 Insert Row 按钮，添加一个新的空行如下图。

ID	Courseid	Studentid	Score
1	1	951303	88.333
2	1	951300	80.333
3	2	951303	80.333
4	3	951300	88.333

6

在空行的各列中输入数据，见下图（ID 列可以不输入，因为在设计数据库时把这个字段设成了自动填充值，如果要输入值也可以，但一定要注意不要和已有的 ID 值冲突，因为这一列是主码）。

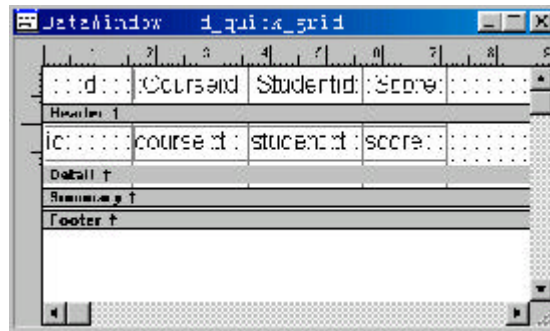
ID	Courseid	Studentid	Score
5	4	951303	100
1	1	951303	88.333
2	1	951300	80.333
3	2	951303	80.333
4	3	951300	88.333

7

退出数据窗口，再重新进入并预览它，发现果然增加了新的一列，如果怀疑这一列是否真的加入了数据库，可以用 Access 打开表 Table_courses，看步骤 6 添加的数据是否在其中。

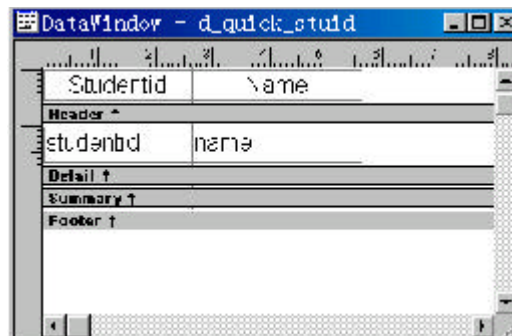
下拉式数据窗口编辑方式 利用下拉式数据窗口的例子

1



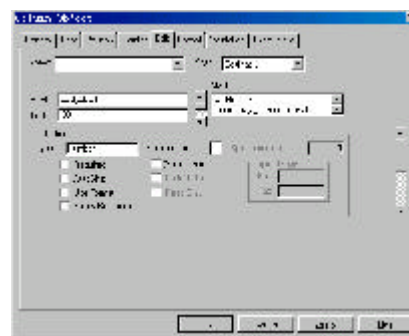
2

列的 Edit Style 能够解决这个问题。为此，先创建一个名为 d_quick_stuid 的数据窗口，它以表 table_primary 的 studentid 和 name 列为数据源，用 Quick Select 数据选取方式，Grid 风格。创建好的数据窗口见下图。

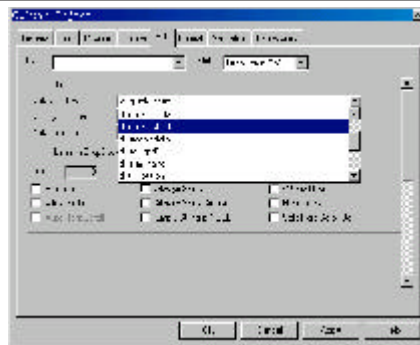


3

再次打开 d_quick_grid，双击 studentid 列对象，在弹出的对话框中选择 Edit 标签页，如下图。在这个标签页可以设置字段值的格式化输入，但这次我们用一种崭新的方式。

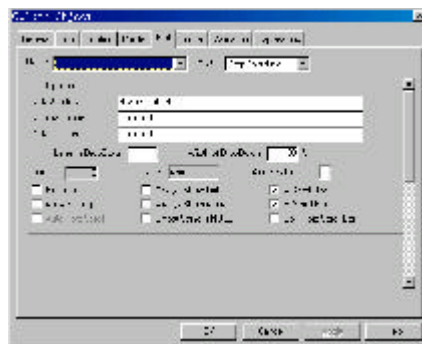


4



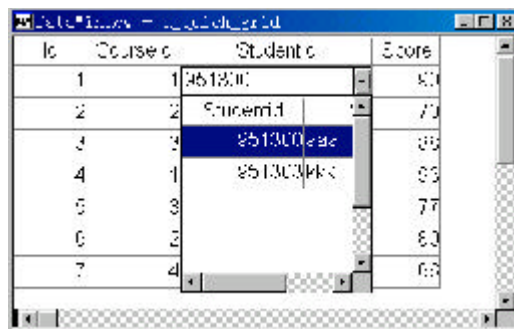
5

在 Display Column 和 Data Column 列表框中都选取 studentid 项，表示 d_quick_grid 的 studentid 列的显示值和存储值都从 d_quick_stuid 的 studentid 列中选取。核选 V Scroll Bar 和 H Scroll Bar 复选框。



6

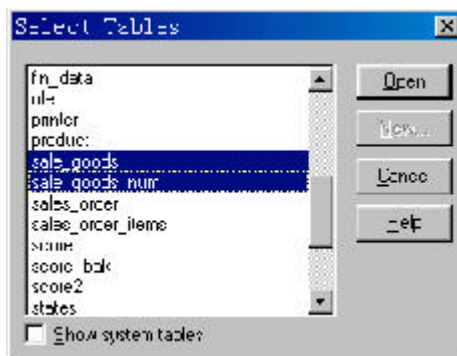
按 OK 保存修改；预览数据窗口。尝试新添加一列或修改一列的 studentid 字段值，看到出现一个带滚动条的下拉式数据窗口，用户必须从中选择一个学号值。这样就保证了用户所输入的学号值在表 table_primary 中。



7

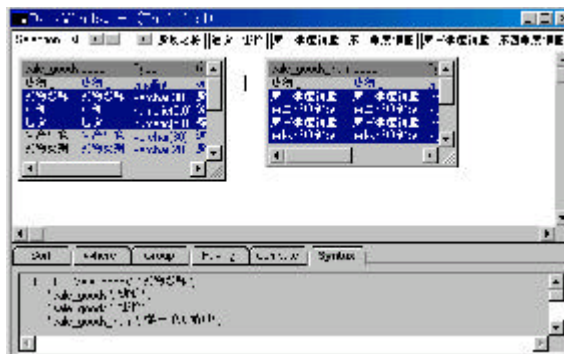
想一想，如果在允许通过下拉式数据窗口选择数据的同时又允许用户直接通过键盘敲入数据值（这样更灵活），该怎么办？（见本节的解释）

1



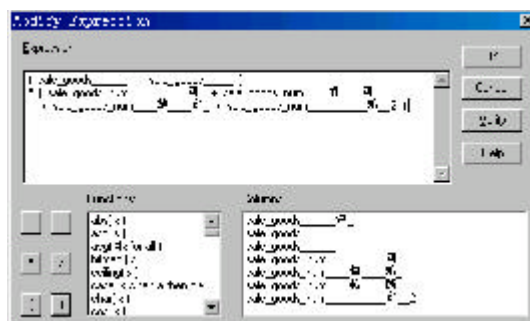
2

要求每一种家电的年度销售额，在两个表中选择家电名称、进价、售价及四个季度的销售量字段。在下面的 Sort、Where、Having、Group 标签页中不添加任何附加信息。



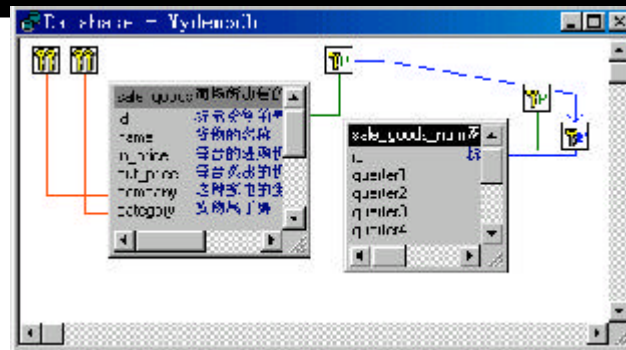
3

按工具条上的 Data Source 按钮进入数据窗口编辑画板，在细节区添加一个计算字段，以统计每种家电的销售额。此计算字段的表达式如下图。表达式的含义是：售价减去进价再乘四个季度的总销售量。



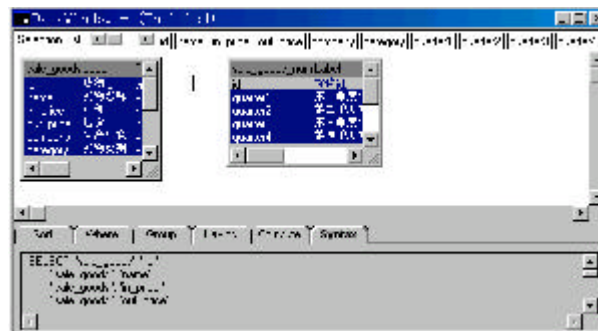
总体例子（二） 分组显示每一类产品销售情况

1



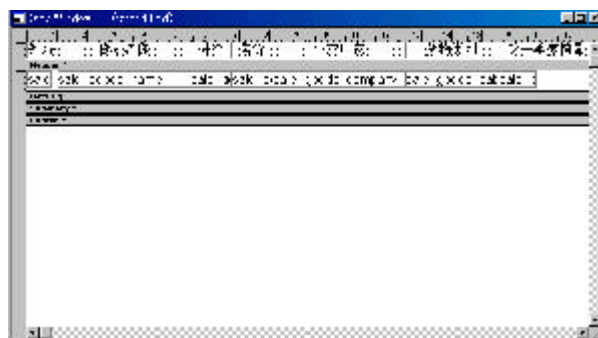
2

新创建一个 SQL Select 数据选取方式、Tabular 表现风格的数据窗口，在 Select Tables 对话框中选择表 Sale_goods 和 sale_goods_num；在 Visual SQL Select 窗口中选择列见下图。



3

单击工具条上的 Data Source 图标按钮，进入数据窗口编辑画板，如下图。读者可以单击 Preview 预览此数据窗口，并设置排序和过滤。

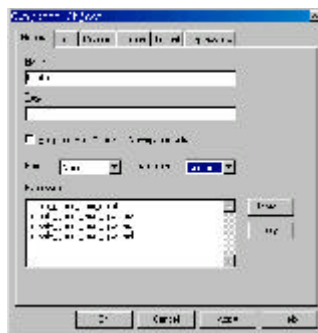


4



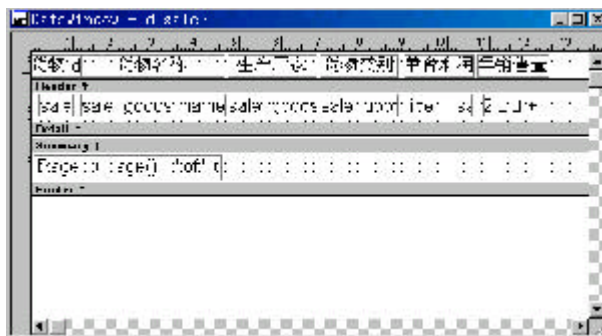
5

删除与四个季度销售量有关的 4 个文本对象和列对象；在细节区最右面新建一个计算对象，其名字为 total_num，表达式为 4 个季度销售量的和，见下图。



6

在页眉区增加“单台利润”和“年销售量”两个文本对象，作为对步骤 4、5 所添加的计算对象的一个说明；在页脚区增加页码计算对象。完成的编辑画板如下图。



7

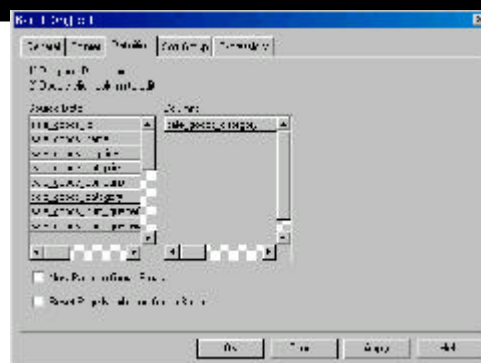
因为在这次操作中，我们所关心的是每一种产品的利润率和年销售量，所以把每一季度的销售量等列对象删除了，而新建了两个计算对象。读者现在可以预览我们的阶段性成果了。

知识窗

总体例子 (二) (续)

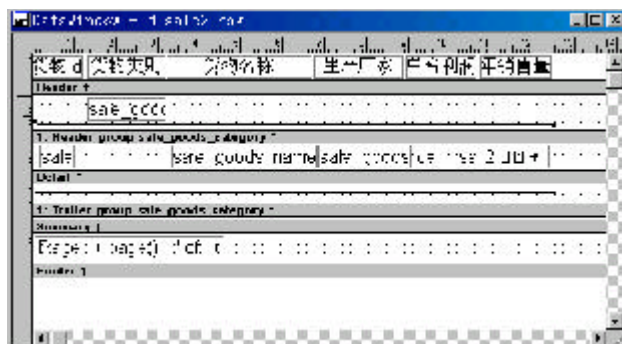
分组显示每一类产品销售情况

1



2

改变数据窗口的布局以适应分组的要求（最起码产品类别字段不适合再在细节区，而应该在组标题区）如下图。在组标题区和组尾区各添加一个 Line 对象，作为组和组之间的分割。



3

在组尾区添加一个计算对象统计本组产品本年度的总利润，如下图。计算对象的名字为 profit，表达式为：对本组各产品 total_num 和 profit_margin 对象值的乘积求和。

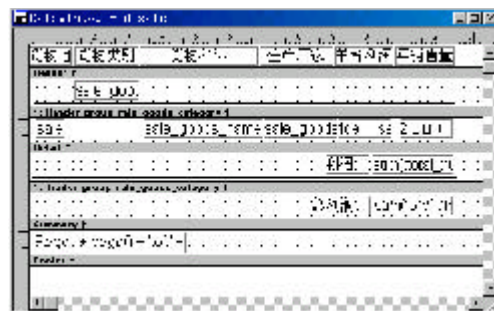


4



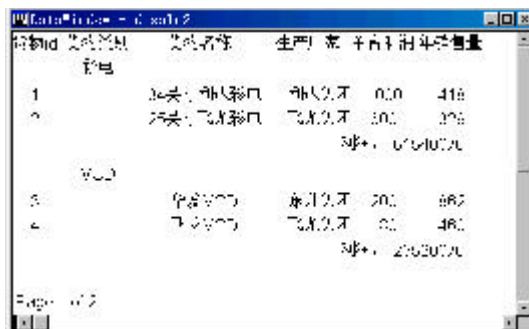
5

再为步骤 4 和步骤 5 的计算对象添加文本说明；完成后的数据窗口布局见下图。读者可以为数据窗口增加一个总标题（建议放在页眉区最上面）和当前日期对象（可以放在页眉区或页脚区）。



6

预览数据窗口，见下图。实际上，创建数据窗口是一个不停地改造布局和预览的过程，当在预览中发现那个地方不太舒服时，马上退出预览，改进布局（如添加分割线、挪动对象位置、设置对象大小等）。



7

保存数据窗口为 d_group_profit。在这儿提醒读者：不要象我这样，等到数据窗口完全设计完毕，大功告成时才想到要保存它；突然的掉电或其它不可预知的意外情况都可能使你的成果毁于一旦。

