

# *PL/SQL Developer 11.0*

## 用户指南

2015 年 1 月





# 目录

1. 介绍 .....	9
2. 安装 .....	13
2.1 系统需求 .....	13
2.2 工作站安装 .....	13
2.3 基于服务器的安装 .....	13
2.4 脚本安装 .....	13
2.5 卸载 PL/SQL DEVELOPER .....	14
3. 登录到一个 ORACLE 数据库 .....	15
4. 编写程序 .....	16
4.1 创建程序 .....	16
4.2 保存程序 .....	17
4.3 修改程序 .....	18
4.4 编译程序 .....	18
4.5 对比程序 .....	19
4.6 编译器首选项 .....	20
5. 测试程序 .....	21
5.1 创建测试脚本 .....	21
5.2 运行测试脚本 .....	22
5.3 变量类型 .....	23
5.4 保存测试脚本 .....	23
5.5 跟踪运行时错误 .....	25
5.6 包状态和 JAVA 会话状态 .....	25
5.7 查看结果集 .....	25
5.8 查看 DBMS_OUTPUT .....	25
5.9 查看 HTP 输出 .....	26
5.10 调试 .....	26
5.11 跟踪运行 .....	30
5.12 回归测试 .....	31
6. 优化 .....	32
6.1 使用解释计划窗口 .....	32
6.2 自动统计 .....	33
6.3 PL/SQL 概览图 .....	34
6.4 SQL 跟踪 .....	35
7. 专用 SQL .....	37
7.1 使用 SQL 窗口 .....	37
7.2 结果表格处理 .....	39
7.3 实例模式查询 .....	43
7.4 链接查询 .....	44
7.5 置换变量 .....	45
7.6 更新数据库 .....	45
7.7 查看和编辑 XMLTYPE 列 .....	46
7.8 比较记录 .....	47

7.9	直接查询导出 .....	47
7.10	注释指令 .....	48
7.11	保存 SQL 脚本 .....	48
7.12	创建标准查询 .....	49
<b>8.</b>	<b>命令窗口 .....</b>	<b>50</b>
8.1	输入 SQL 语句和命令 .....	50
8.2	开发命令文件 .....	51
8.3	支持命令 .....	51
<b>9.</b>	<b>创建和修改非 PL/SQL 对象 .....</b>	<b>55</b>
9.1	表定义编辑器 .....	55
9.2	序列定义编辑器 .....	69
9.3	同义词定义编辑器 .....	69
9.4	库定义编辑器 .....	70
9.5	目录定义编辑器 .....	70
9.6	作业定义编辑器 .....	71
9.7	队列定义编辑器 .....	71
9.8	队列表定义编辑器 .....	72
9.9	用户定义编辑器 .....	73
9.10	角色定义编辑器 .....	76
9.11	概要文件定义编辑器 .....	77
9.12	数据库链接定义编辑器 .....	77
<b>10.</b>	<b>DBMS 调度程序 .....</b>	<b>78</b>
10.1	创建对象 .....	84
10.2	编辑对象 .....	84
10.3	删除对象 .....	84
10.4	查看作业运行 .....	84
10.5	查看正在运行的作业 .....	86
<b>11.</b>	<b>图表 .....</b>	<b>87</b>
11.1	创建图表 .....	87
11.2	缩放 .....	91
11.3	保存和打开图表文件 .....	91
11.4	更新图表 .....	91
11.5	打印图表 .....	92
11.6	导出图表 .....	92
<b>12.</b>	<b>报告 .....</b>	<b>93</b>
12.1	标准报告 .....	93
12.2	定制报告 .....	94
12.3	变量 .....	95
12.4	精制版面 .....	100
12.5	样式库 .....	106
12.6	选项 .....	107
12.7	报告菜单 .....	109

<b>13. 图形 .....</b>	<b>110</b>
<b>14. 工程 .....</b>	<b>112</b>
14.1    创建一个新工程 .....	112
14.2    保存工程 .....	113
14.3    添加文件到工程 .....	113
14.4    添加数据库对象到工程 .....	113
14.5    用工程项目工作 .....	114
14.6    编译工程 .....	115
<b>15. 任务项目 .....</b>	<b>116</b>
15.1    创建任务项目 .....	117
15.2    编辑任务项目 .....	117
15.3    关闭任务项目 .....	118
15.4    删除任务项目 .....	118
<b>16. 列表链接 .....</b>	<b>119</b>
16.1    列表链接使用说明 .....	119
16.2    定义文件夹和链接 .....	122
<b>17. 链接、窗口、数据库会话和事务 .....</b>	<b>125</b>
17.1    主链接 .....	125
17.2    链接窗口 .....	125
17.3    会话模式 .....	126
17.4    运行于多路会话或双路会话模式 .....	126
<b>18. 浏览对象 .....</b>	<b>127</b>
18.1    使用对象浏览器 .....	127
18.2    对象浏览器过滤器 .....	133
18.3    对象浏览器文件夹 .....	134
<b>19. 浏览文件 .....</b>	<b>136</b>
19.1    设置文件浏览器 .....	137
19.2    使用文件浏览器 .....	138
<b>20. 首选项 .....</b>	<b>140</b>
20.1    ORACLE - 链接 .....	141
20.2    ORACLE - 选项 .....	142
20.3    ORACLE - 编译器 .....	144
20.4    ORACLE - 调试器 .....	146
20.5    ORACLE - 输出 .....	147
20.6    ORACLE - 跟踪 .....	148
20.7    ORACLE - 概览图 .....	148
20.8    ORACLE - 登录历史 .....	149
20.9    ORACLE - 提示 .....	151
20.10    用户界面 - 选项 .....	152
20.11    用户界面 - 对象浏览器 .....	154
20.12    用户界面 - 编辑器 .....	155
20.13    用户界面 - 字体 .....	159
20.14    用户界面 - 代码助手 .....	160
20.15    用户界面 - 键配置 .....	161

20.16	用户界面 - 外观 .....	162
20.17	用户界面 - NLS 选项.....	164
20.18	窗口类型 - 程序窗口.....	165
20.19	窗口类型 - SQL 窗口.....	167
20.20	窗口类型 - 测试窗口.....	170
20.21	窗口类型 - 计划窗口.....	170
20.22	工具 - 差异 .....	170
20.23	工具 - 数据生成器 .....	171
20.24	工具 - 任务列表 .....	172
20.25	工具 - 重新调用声明.....	172
20.26	文件 - 目录 .....	173
20.27	文件 - 扩展名 .....	175
20.28	文件 - 格式 .....	176
20.29	文件 - 备份 .....	177
20.30	文件 - HTML/XML .....	178
20.31	其他 - 打印 .....	179
20.32	其他 - 更新与消息 .....	180
20.33	首选项集.....	181
<b>21.</b>	<b>工具.....</b>	<b>183</b>
21.1	对象浏览器 .....	183
21.2	查找数据库对象 .....	183
21.3	编译无效对象 .....	185
21.4	导出表 .....	186
21.5	导入表 .....	189
21.6	导出用户对象 .....	191
21.7	文本导入器 .....	192
21.8	ODBC 导入器 .....	196
21.9	数据生成器 .....	198
21.10	比较用户对象 .....	202
21.11	比较表数据 .....	204
21.12	事件监视器 .....	207
21.13	会话 .....	208
21.14	自定义工具 .....	210
21.15	测试管理器 .....	215
<b>22.</b>	<b>编辑器.....</b>	<b>219</b>
22.1	选择功能 .....	219
22.2	列选择 .....	219
22.3	代码助手 .....	220
22.4	重新调用语句 .....	221
22.5	专用复制 .....	222
22.6	从主机语言粘贴 .....	223
22.7	上下文敏感帮助 .....	224
22.8	数据库对象弹出式菜单 .....	224
22.9	解释计划 .....	224
22.10	宏 .....	224
22.11	书签 .....	225

22.12	颜色标记	226
22.13	代码目录	227
22.14	代码层次	228
22.15	代码折叠	228
22.16	拆分编辑	229
22.17	超链接导航	229
22.18	导航按钮	230
22.19	重构	230
22.20	搜索栏	231
22.21	WEB 搜索	232
<b>23.</b>	<b>大数据编辑器</b>	<b>234</b>
23.1	编辑纯文本	235
23.2	编辑 RTF	236
23.3	编辑 XML	237
23.4	编辑 HTML	239
23.5	编辑图象	239
23.6	编辑十六进制数据	240
23.7	调用外部查看器或编辑器	241
<b>24.</b>	<b>查询设计器</b>	<b>243</b>
24.1	创建新的 SELECT 语句	243
24.2	修改现有的 SELECT 语句	247
24.3	处理查询定义	247
24.4	查询设计器首选项	248
24.5	查询设计器插件	249
<b>25.</b>	<b>PL/SQL 美化器</b>	<b>250</b>
25.1	定义选项	250
25.2	定义规则	251
25.3	使用美化器	252
25.4	取消格式化	252
<b>26.</b>	<b>模板</b>	<b>253</b>
26.1	模板窗口	253
26.2	使用模板	254
26.3	创建和修改模板	255
<b>27.</b>	<b>窗口列表</b>	<b>258</b>
<b>28.</b>	<b>可停放和浮动的工具</b>	<b>259</b>
<b>29.</b>	<b>工具栏</b>	<b>261</b>
<b>30.</b>	<b>授权</b>	<b>264</b>
30.1	允许授权	264
30.2	定义授权	265
30.3	禁用授权	266
<b>31.</b>	<b>ORACLE 文件系统 (OFS)</b>	<b>267</b>
31.1	OFS 管理器	267
31.2	OFS 用法	269

---

<b>32. 帮助系统 .....</b>	<b>271</b>
32.1    MS 帮助文件 .....	271
32.2    HTML 手册 .....	271
<b>33. 定制 .....</b>	<b>275</b>
33.1    首选项 .....	275
33.2    窗口版面 .....	275
33.3    在线文档 .....	275
33.4    命令行参数 .....	277
33.5    SQL、PL/SQL、和命令关键字 .....	280
33.6    插件 .....	280

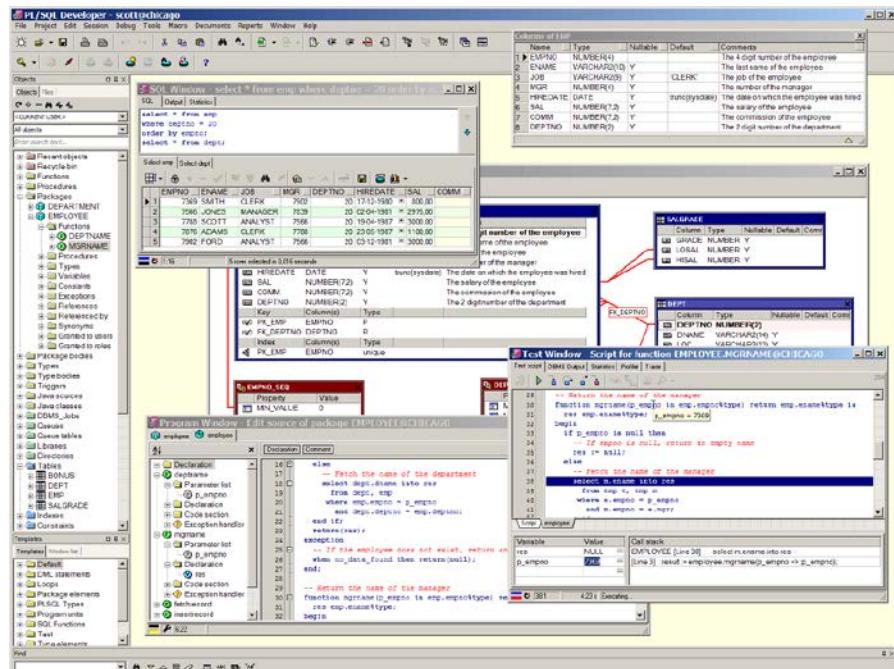
# 1. 介绍

PL/SQL Developer 是一个为 Oracle 数据库开发存储程序单元的集成开发环境(IDE)，使用 PL/SQL Developer，你能方便地创建你的客户端/服务器应用程序的服务器部分。

到目前为止，从最差处想，你还可以完成如下工作：

- 使用文本编辑器编写程序单元（过程、触发器等）。
- 使用 Oracle SQL\*Plus 编译源文件。
- 如果有编译错误，你必须找出它位于源文件中何处，纠正它，并转回到 SQL\*Plus 重新编译它，然后再找下一处错误。
- 使用 SQL\*Plus 或你的客户端应用程序测试程序单元。
- 如果发生运行时错误，你又不得不辛苦地定位问题的原因并纠正错误。
- 使用解释计划工具或 tkprof 工具优化你的 SQL 语句。
- 使用 SQL\*Plus 或其他工具在你的数据库中查看或修改其他对象和数据。

这些任务——编辑、编译、纠正、测试、调试、优化和查询——都可以在不离开 PL/SQL Developer IDE 的情况下被完成。此外，PL/SQL Developer 还提供了其他几个能在日常 PL/SQL 开发中提供帮助的工具。



## 编辑

与其他严肃的开发环境一样，PL/SQL Developer 也可以在磁盘上存储你的源文件，其他工具只让你在数据库中编辑源文件，而这不支持执行任何版本控制或部署方案。源文件能通过 SQL\*Plus 运行，所以你不使用 PL/SQL Developer 也能在任何平台上部署它们。你可以通过标准的多文档界面一次编辑很多文件。

编辑器为程序员提供了广泛的协助。对于 SQL 语句和 PL/SQL 语句，它提供了上下文敏感的帮助。所有信息都可以即查即得：如果你开始键入 *substr*，但已经忘了参数的准确含义，那么现在你可以简单地单击 *F1* 就能获得 *SQL 参考手册* 的相应主题。同样地，在编辑器中卷起的窗口中，还能够为您提供表、视图和程序单元的描述。代码助手被集成在编辑器中，当你键入数据库对象的名称时，它会自动显示出数据库对象的信息，允许你从这些信息中浏览和挑选选项。对于大的包或类型体，程序编辑器提供了带有代码内容的树视图以方便导航，高亮显示了代码结构和变量参照，支持折叠/展开代码部分，还提供了超链接导航。查询设计器允许你图形化地创建 select 语句。PL/SQL Developer 的可扩展模板简化了向程序中插入标准 SQL 和 PL/SQL 代码的过程。所有编辑器都使用恰当的 SQL、PL/SQL 和 SQL\*Plus 语法高亮显示功能，使你的代码更具可读性。

## 编译与纠错

在编辑器中，你可以编译源文件。如果发生编译错误，将自动跳转到相应的源代码行，所有编译错误都被记录在编辑器底部的一列表中。这个列表可以包含代码提示，可以指出常见程序设计错误或不规范的自定义命名约定。

由于你可以编译源文件而不用保存它，你能安全地探索解决问题的很多可选方法。编辑器会记住你已经修改了文件而没有保存或编译它。

## 测试与调试

要测试你的程序单元，可以写一个程序脚本，这个测试脚本的主体包含一个 PL/SQL 块，你可以在其中规划测试代码，可以对要在 PL/SQL 块中使用的任何变量进行声明、分配输入值，以及在运行后进行查看。

当你运行一个测试脚本时，也许会出现运行时错误。在这种情况下，PL/SQL Developer 允许你查看错误堆栈来源，每个涉及运行时错误的错误堆栈源行都会被高亮显示，所以你可以轻松地返回到有问题的地方。

如果你正在使用 7.3.4 或更高版本的 Oracle，可以使用 PL/SQL Developer 的集成调试器。你可以单步调试代码、设置断点、查看/设置变量、查看调用堆栈等。

在 Oracle8i 以及更高版本上，你还可以使用 *dbms\_trace* 特性把程序运行的已选事件记入日志，这能帮助你分析程序流和异常。

调用 *dbms\_output* 和 *PL/SQL Web* 工具箱得到的输出会自动在相应的测试窗口标签页中显示。对于回归测试，你可以使用测试管理器来快速运行和校验一组测试脚本。

## 优化

要在你的程序单元中优化 SQL 语句，Oracle 的解释计划工具能够提供很大帮助，所以它集成在了 PL/SQL Developer 的 IDE 中。只要在源文件中简单地选择 SQL 语句并按 F5，查询计划就会在一个单独的解释计划窗口中生动地显示出来，然后你就能在源文件之外通过修改语句来优化查询计划，并把它复制回去。

对于已运行的 SQL 语句和 PL/SQL 程序单元，你还能查看统计表，这些统计表能包括耗时、CPU 时间、逻辑读、物理读、物理写等内容。

Oracle8i 引入了 PL/SQL 概览图，它允许你概览你的 PL/SQL 代码。对于每个已运行的 PL/SQL 代码行，你能测定运行时间以及已经运行了多少次。

## 查询

要查询数据库中的数据，你可以使用 SQL 窗口运行任何 SQL 语句。所有已运行的语句都被保留在历史缓冲区中，因此你能够轻松地重新运行它们。任何查询结果都会被方便地显示在一个单独的表格中，随后你可以在这个表格中插入、更新或删除记录。结果表格还可用于查询实例模式，以便你能容易地查找你需要的信息。

要查询数据库对象，你可以使用对象浏览器。数据库对象的所有相关属性，例如表、视图、序列、函数、过程、包、类型、触发器等，以及对象之间的依赖关系，都可供查看。对象浏览器使用了一种树视图，该视图类似于 Windows 资源管理器，只需指向并单击即可进行浏览。

## 运行 SQL 脚本

PL/SQL Developer 包括一个可用于运行 SQL 脚本或运行 SQL 语句的命令窗口，你还可以执行与你可能熟悉的 SQL\*Plus 命令很相似的命令。

## 创建和修改表定义

你可以容易地创建和修改带有使用任何 SQL 语句的表定义，只需在对话框窗口中填写定义，修改列、约束、索引、权限、存储信息、注释等。你可以在数据库中应用这些定义，并查看、修改和保存生成的 SQL。

## 图表

要显现应用程序或工程的数据库对象以及它们之间的关系，可以创建图表。图表可用于存档，也可用作处理相关对象的工作空间。

## 报告

PL/SQL Developer 附带了许多基于 HTML 格式的标准报告。你可以在 PL/SQL Developer 中查看这些报告，可以打印它们或将它们另存为 HTML 文件，还可以创建你自己的定制报告。可以从报告菜单轻松访问报告。

## 图形

图形窗口可以运行于 SQL 窗口或报告窗口内部，以获得查询数据的图示。

## 工程

要组织你的工作，你可以使用 PL/SQL Developer 的工程概念，工程由很多文件和数据库对象组成，可以通过工程项目窗口轻松访问这些对象，并且通过单击鼠标来编译它们。

## 工具

PL/SQL Developer 提供了几个对开发有帮助的工具，这些工具包括允许你在数据库对象源中搜索文本的查找数据库对象工具、支持快速编译在开发期间已经变得无效的对象的编译无效对象工具、导出和导入工具、创建测试数据的数据生成器工具、用于导出用户对象 DDL 语句的导出用户对象工具、用于比较两个用户对象定义的比较用户对象工具、会话信息工具、事件监视器

和一个用于比较并使表内容相同的比较表数据工具，以及一个用于查看和管理您的日程安排事务 DBMS 日程管理工具。

除了这些标准工具之外，你还可以定义你自己的工具，并把他们包括在 PL/SQL Developer 的工具菜单中。

## 2. 安装

主要有两种安装 PL/SQL Developer 的方法：

- 工作站安装：在这种情况下，你要在每个被使用的工作站本地安装这个软件。
- 服务器安装：在这种情况下，你要在服务器上从每个被使用的工作站能访问到的地方安装这个软件。

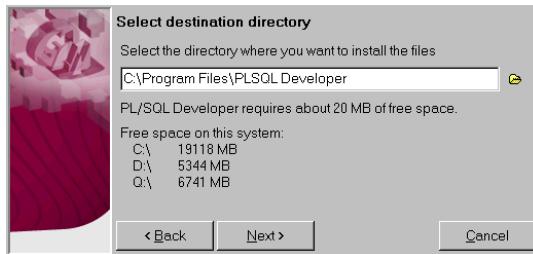
下面将解释这两种安装类型。

### 2.1 系统需求

PL/SQL Developer 可在 Windows 2000 或更新版本上运行，兼容 Oracle Server 7.0 及更新版本。要链接到 Oracle 数据库，PL/SQL Developer 需要 32 位版本的 Oracle Client 7.0 或更新版本。如果你的工作站上使用的是 64 位版本的 Oracle Client 和/或 Server，则需另行安装 32 位的 Oracle Client。32 位 Oracle Client 可与 64 位 Server 通信，在必要情况下，甚至可与位于相同计算机上的 Server 通信。安装在不同 Oracle Home 目录中的 32 位客户端可与 64 位 Oracle 安装共存。

### 2.2 工作站安装

要在本地工作站上安装 PL/SQL Developer，运行安装媒体中的安装程序，然后将显示下面的对话框：



你可以为程序文件选择一个目标目录、一个开始菜单文件夹、一个 PL/SQL Developer 桌面快捷方式以及安装选项。按下最后一页上的完成按钮之后，将会复制程序文件并创建快捷方式。

### 2.3 基于服务器的安装

对于多用户许可，你可以在文件服务器上创建一个单一的 PL/SQL Developer 安装，并对所有许可用户都提供访问权限。为此，您可以使用前一节描述的方法运行服务器上的安装程序。你必须选择使用这个软件的用户的共享目标目录。当 PL/SQL Developer 第一次在工作站上运行时，这种安装方式会被自动检测到，并且本地安装将在无需用户干预的情况下完成。

PL/SQL Developer 能安装到用户只具有读访问权限的目录中。

### 2.4 脚本安装

除了上面描述的安装过程之外，你还可以创建无人值守安装的脚本，请阅读 *install.txt*，了解详细信息。

## 2.5 卸载 PL/SQL Developer

若因为某些原因你希望卸载 PL/SQL Developer，你可以使用控制面板中的添加/删除程序来删除它。

### 3. 登录到一个Oracle数据库

一打开 PL/SQL Developer，就可以快速登录进一个 Oracle 数据库。



如果你取消了这个登录的页面，你就不能连接到 Oracle。你的 PL/SQL Developer 里的所有要求链接的功能都不能使用了，比如编译功能和查询功能。不过你稍后也可以从 PL/SQL Developer IDE 的页面中获取链接，通过此种方式链接，所有需要取得链接的功能同样可以使用。

登入您的用户名，密码和数据库之后，点击确定或者登录，你就可以和数据库连接上，PL/SQL Developer 的 IDE 页面就会显示出来。现在你就可以编写 PL/SQL 程序了。编写 PL/SQL 程序，查询数据库等等功能我们将会在下一章为您做详细介绍。

如果你拥有多个数据库或多个数据库账户，你可以从会话菜单中选择登录选项来添加链接，或者你也可以在主工具栏点击登入按钮。每一个添加的链接都会被添加进链接目录里（详细信息见16章），链接目录随后可以被用来查看和管理你的链接。你以后再打开 PL/SQL Developer 的时候，就可以快速的从链接目录里选择你需要的链接。

#### 登录失败

下述原因可能造成登录失败：

1. 你没有安装 Oracle Client 软件。这是使用 PL/SQL Developer 的必须条件。安装 Oracle Client 后再重试。
2. 你安装了多个 Oracle Client 软件。在这种情况下，PL/SQL Developer 可能会选择一个不正确的 Oracle Client。点击登录页面的取消，点击工具>首选项>Oracle 里的链接，从目录中选择正确的 Oracle Home，重启 PL/SQL Developer 后再重试。
3. 你使用了一个 Oracle Instant Client，但是你没有设定需求的环境变量。

PATH-需要包含 Instant Client 的根目录 (oci.dll 所在位置)

TNS ADMIN-需要指出 tnsnames.ora 所在位置的根目录

NLS LANG-为客户设定语言，区域和特征。

注意：也可以在命令行或 params.ini 文件中设置这些参数，如第[33.4](#)节所述。

## 4. 编写程序

在 Oracle 数据库中，你能辨别出五种不同类型的存储程序单元：函数、过程、包、类型和触发器，程序编辑器允许你用同一种方法创建和修改这五种类型的程序单元。

### 4.1 创建程序

要创建新程序，请按下工具栏中的新建按钮，选择程序窗口项目，然后可以选择函数子项目。你现在会被提示选择与这个程序类型相关的不同变量。在本例中，我们在其中创建函数、函数名、参数列表以及它的返回值类型：



由这个对话框的标题可以看出，此信息来自一个模板。PL/SQL Developer 提供了几个能根据需要修改的标准模板，你也可以定义新模板，关于定义模板的信息将在第 26 章提供。

输入变量并按下确定按钮以后，将出现程序编辑器窗口，其中包含一个模板函数。在程序编辑器中创建的每个程序单元都必须以熟悉的 SQL 语法 'create or replace' 开始。在本例中，我们创建了一个 'even' 函数，源文件如下图所见：

```

create or replace function Even(Value in integer) return boolean is
  Result boolean;
begin
  -- Check if there is a remainder after dividing by 2
  Result := (Value mod 2 = 0);
  return(Result);
end Even;

```

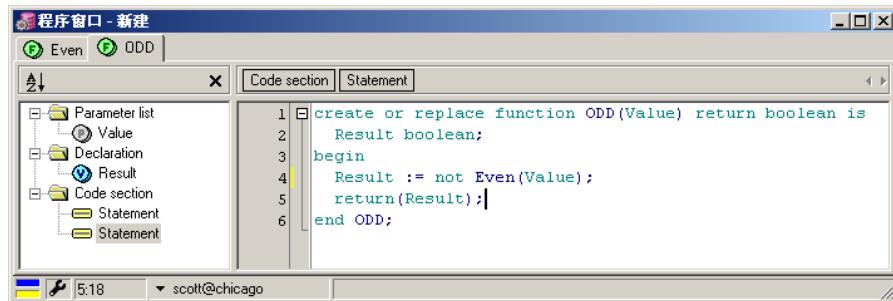
在编辑器左边，可以看到显示程序单元结构的代码目录。当导航包体和类型体等大的程序单元时，此特性很有用。这个特性将在第 22.13 节中详细描述。

在编辑器的上方，可以看到显示光标位置的层次结构的代码层次。在本例中，它显示你位于函数中语句序列中的语句位置（返回语句）。对于复杂程序，准确地看到你的位置很有用，这能帮助你选择 PL/SQL 代码的某一部分。详细描述在第 22.14 节提供。

程序文件可以包含不止一个程序单元。在程序编辑器中右键单击鼠标，将显示一个允许你添加或删除程序单元的弹出式菜单。你可以通过在窗口上方选择适当的标签在程序单元之间互相切

换，这个方法能使你能够方便地将相关程序单元保留在一个源文件中，包规范与体就是这个特性的一个很好的例子。

程序单元应该放在程序文件中它可能参照的其他任何程序单元的后面。如果你创建了一个参照了前面创建的函数 'even' 的函数 'odd'，则程序编辑器应该类似于下图：



## 4.2 保存程序

你可以按下工具栏中的 **保存** 按钮来保存程序文件，将弹出一个保存对话框，其中显示了一些标准扩展名：

程序类型	扩展名
函数	.fnc
过程	.prc
包规范与体	.pck
包规范	.spc
包体	.bdy
类型规范与体	.typ
类型规范	.tps
类型体	.tpb
触发器	.trg
Java 源文件	.jsp

你可以通过如第 20.27 节章所描述的首选项来更改这些扩展名。如果程序文件中只有一个单一的函数、过程、包、类型、触发器或 java 源文件，则文件名会自动根据这个程序单元的名称和类型而生成。文件保存以后，文件名将在窗口标题中显示出来。

已保存的程序文件具有一种与 Oracle SQL\*Plus 兼容的格式。例如，包含 'odd' 和 'even' 函数的源文件格式如下：

```
create or replace function Even(Value in integer) return boolean is
  Result boolean;
begin
  --check if there is a remainder after dividing by 2
  Result := (Value mod 2 = 0);
  return(Result);
end Even;
/
create or replace function Odd(Value in integer) return boolean is
  Result boolean;
begin
  Result := not Even(Value);
  return(Result);
end Odd;
/
```

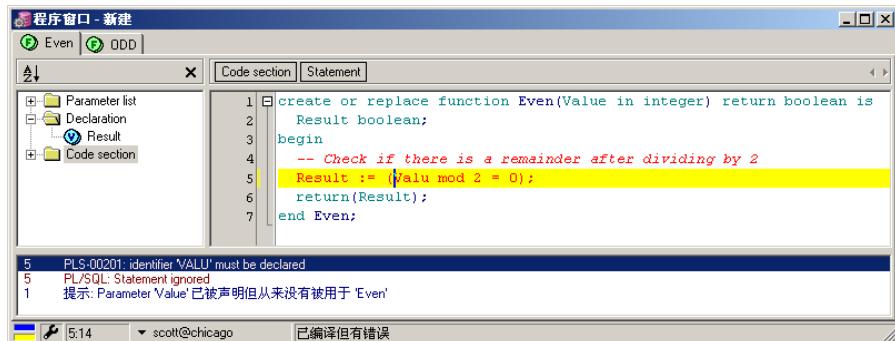
当通过 SQL\*Plus 执行此文件时，将创建这两个函数。

## 4.3 修改程序

按下工具栏上的打开按钮并选择程序文件项目，可以打开以前保存的程序文件。如果文件在最近用过，通过选择文件菜单上的重新打开或单击工具栏中打开按钮旁边的箭头，也可以打开它。当你开始编辑时，窗口底部有一个蓝色的指示器将点亮，这意味着文件已经被修改，但还没有保存。还有一个黄色指示器，它意味着文件已经被修改，但还没有编译。

## 4.4 编译程序

按下工具栏上的执行按钮，可以编译程序文件。程序文件中的所有程序单元都会被编译，不管你当前选择了哪个程序单元，都会从第一个单元开始编译。当出现错误时，编译将被终止，编辑器将定位到引起错误的源代码行：



```
程序窗口 - 新建
Even | ODD |
Code section | Statement
Parameter list
Declaration
Result
Code section
create or replace function Even(Value in integer) return boolean is
  Result boolean;
begin
  -- Check if there is a remainder after dividing by 2
  Result := (Value mod 2 = 0);
  return(Result);
end Even;
/
create or replace function Odd(Value in integer) return boolean is
  Result boolean;
begin
  Result := not Even(Value);
  return(Result);
end Odd;
/
5  PLSQL: Statement ignored
5  提示: Parameter 'Value' 已被声明但从来没有被用于 'Even'
1  PL/SQL: Statement ignored
1  提示: Parameter 'Value' 已被声明但从来没有被用于 'Even'
```

如果有不只一个错误并且你希望纠正它们，可以单击程序编辑器底部的下一个错误信息，转到错误位置。如果你已经为 PL/SQL Developer 配置了如第 32.2 节中所描述的 HTML 手册，在编译错误上双击就可以显示 Oracle 错误信息手册中的相关段落。

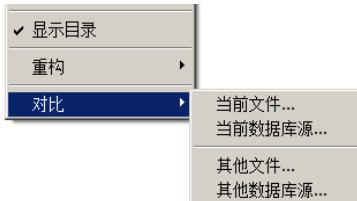
编译错误列表也可以包含提示。提示不是引起编译失败的原因，但它们可能显示能引起运行时错误的问题（例如与 null 比较、函数没有返回值、未使用的参数等）。关于提示的更多信息，请阅读第 20.18 节。

如果在编译后消息框中显示有错误消息，说明创建语句失败，没有实际编译源文件，错误消息应该解释错误的原因。例如，如果你得到的消息是 ‘create or replace’ 语法错误，信息框中会显示 “ORA-00900: 无效的 SQL 语句”。

注：如果你使用的 Oracle 服务器是 7.2 或更早版本，触发器编译错误总是在消息框中报告。从 Oracle 服务器的 7.3 版本开始，触发器编译错误采用与过程、函数、包、类型一样的方式进行报告。

## 4.5 对比程序

要将程序源与数据库或文件系统内的另一版本对比，可右键单击编辑器，再单击**对比**子菜单项：



这四个功能将以可视化的方式显示两个程序源之间的逐行对比。配置外部差异工具按钮允许配置要使用的差异工具。默认使用 *ExamDiff* 实用工具，该工具提供 Pro 专业版本（请参见 *ExamDiff* 信息菜单的关于项目）。另请参见第 20.22 节。

### 当前文件

如果已经打开了文件系统中的某个文件并作出了一些修改，此功能将显示磁盘文件与程序窗口中文件的差异。保存程序窗口后，这两个版本将始终相同。如果无与程序窗口相关的文件，此功能将禁用。

### 当前数据库源

此功能将显示数据库中的对象源于当前编辑器之间的差异。编译之后，这两个版本将始终相同。

### 其他文件

此功能允许将当前程序窗口与任意文件对比。选择文件后，即可显示差异。

### 其他数据库对象

此功能允许将当前编辑器与数据库内的任意对象源对比。选择数据库对象链接后即可显示差异。

## 4.6 编译器首选项

要设置不同的编译器首选项，可以在程序窗口的状态栏中按下**编译器首选项**按钮。此操作会显示相应的首选项页面。有关更多信息，请参见第 20.3 节。

## 5. 测试程序

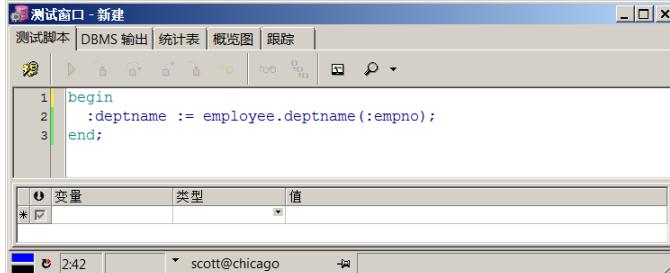
成功编译了程序以后，你还需要测试它。为此，可以使用 PL/SQL Developer 的测试脚本。测试脚本允许你运行一个或多个程序单元，定义输入、输出和输入/输出变量，以及查看变量和给变量赋值。如果在执行测试脚本期间出现运行时错误，可以查看涉及错误的程序单元的源代码。

如果你使用的是 Oracle 7.3.4 或更新版本，可以使用 PL/SQL Developer 的集成调试器。你可以单步调试代码，设置断点，查看和设置变量，查看调用堆栈等。

### 5.1 创建测试脚本

有两种创建测试脚本的方法。可以简单地右键单击对象浏览器中的一个程序单元（一个独立的函数或过程、一个打包的函数或过程、或者一个对象类型方法），选择 **测试项目**。这将创建一个新的测试脚本，这个测试脚本带有参数变量和结果变量。对于本手册，我们将从头开始手工创建一个测试脚本。

要创建空的测试脚本，请按下工具栏中的 **新建** 按钮，选择 **测试窗口** 项目。你现在可以用类似于 `declare...begin...end` 这样的语法在测试脚本的正文内键入一个 PL/SQL 块。作为一个例子，我们将测试包 `employee`，该包实现雇员/部门演示表的雇员函数。函数 `deptname` 返回雇员部门的名称，并且能使用下列 PL/SQL 块来测试：



The screenshot shows the PL/SQL Developer interface with a window titled "测试窗口 - 新建". The window has tabs at the top: "测试脚本" (selected), "DBMS 输出", "统计表", "概览图", and "跟踪". Below the tabs is a toolbar with various icons. The main area contains the following PL/SQL code:

```
1 begin
2   :deptname := employee.deptname(:empno);
3 end;
```

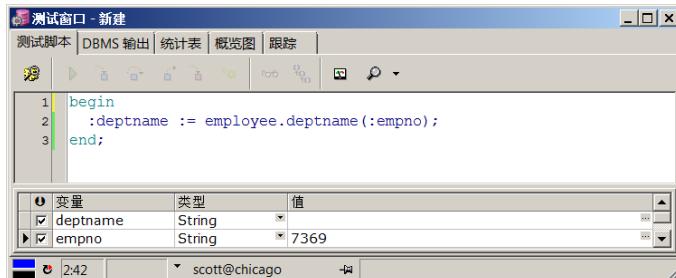
Below the code, there is a table labeled "变量" (Variables) with columns "变量" (Variable), "类型" (Type), and "值" (Value). It shows one entry:

变量	类型	值
*		

At the bottom of the window, there is a status bar showing "2:42" and "scott@chicago".

这个 PL/SQL 块包含对我们要测试的函数的一个简单调用，它使用变量 `deptname` 和 `empno`。我们必须给 `empno` 赋一个值，并在执行之后检查 `deptname` 的值，以确定函数是否返回了正确的值。

用冒号作为这些变量的前缀，可以在测试脚本窗口的底部声明这些变量：



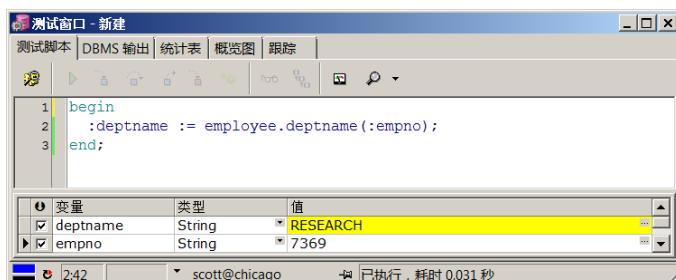
变量列表左上方的**扫描变量源**按钮（）可用于快速复制源代码中的变量。在此之后，整型数据类型和值 7369（来自于调查部的 Smith 先生）就被赋给了 `empno`。现在，我们已经可以执行脚本。

你可以启用首选项（参见第 20.20 节），让测试窗口在执行脚本之前自动搜索绑定变量。

你也可以在 PL/SQL 块的声明部分中本地声明变量，但你将不能修改或查看这些变量的值。这些本地变量的优势是你能使用记录类型、PL/SQL 表、类型等。

## 5.2 运行测试脚本

要运行测试脚本，请按下工具栏中的**执行**按钮。在运行完后，变量的值将被更新，任何更改了值的变量都会用黄色的背景显示出来：



函数运行的结果是 'RESEARCH'（调查部的调查。——译者姜华东注），显然它的运行是正确的。在窗口底部，显示了运行时间，此信息可用于优化你的代码性能。为了进行优化，你也可以选择统计表标签来查看 PL/SQL 块的运行统计表。通过在运行脚本前按下**创建概览图报告**按钮，还可以为已执行的 PL/SQL 代码行创建概览报告。在运行完后，可以转到**概览图**页查看报告。统计表和概览图分别在第 6.2 节和第 6.3 节中详细解释。

你可以按下**中断**按钮来终止一个正在运行的测试脚本，这在程序陷入死循环或者运行时间长于预期时特别有用。要注意的是按下**中断**按钮并不总是成功的。例如，如果程序正在等待一个锁，它并不会响应中断信号。

运行了一个脚本后，被执行的程序单元可能已经启动了一个事务。若是这样，工具栏中的**提交**和**回滚**按钮将被激活。要了解关于事务的更多信息，请参见第 17 章。

## 5.3 变量类型

测试窗口的变量部分的类型列表包含可供使用的所有类型，本节将解释这些类型的特性。

类型	描述
Integer	可用于数字值，范围为 $-2^{32}$ 到 $2^{32} - 1$ 。
Float	可用于其他数字值。
String	相当于表中的 varchar2 数据类型，能容纳 2000 (Oracle7) 或 4000 (Oracle8) 个字符。此数据类型的值单元格中有一个能调用文本编辑器的单元格按钮，所以你能够容易地查看和修改多行字符串。
Date	日期和时间数据类型。
Long	Long 数据类型不在表格中显示它的值，只显示 <Long>，按下单元格按钮可以调出文本编辑器。
Long Raw	按下 Long Raw 值的单元格按钮将调出一个导入/导出文件对话框，允许你载入或保存这个变量的内容。
Cursor	此数据类型用于 PL/SQL 中你能使用游标变量的地方。运行了测试脚本后，你可以按值的单元格按钮来在 SQL 窗口中显示结果集。
CLOB, BLOB 与 BFile	在你能查看 LOB 数据以前，这些 LOB 定位变量必须在服务器上被初始化。CLOB 将调用文本编辑器， BLOB 与 BFile 将调用导入/导出文件对话框。
PL/SQL String	为 PL/SQL 使用此数据类型， varchar2 值可以达到 32000 个字符。
Char	定长字符串（填补空格）。
Substitution	置换变量的使用没有绑定变量的限制。在它被发送到服务器以前，PL/SQL 块中的变量名会被替换为它在文本中的值。
临时 CLOB 与 BLOB	在通过测试脚本被传送到服务器以前，这些 LOB 定位器已经可以持有数据了。

### Boolean 变量

当你选择变量类型列表框时，你会注意到缺少 Boolean 数据类型，这是因为 SQL\*Net 不支持这种数据类型。要使用 Boolean 变量，你可以将它声明为整型，并使用 Oracle 提供的 `sys.diutil.bool_to_int` 和 `sys.diutil.int_to_bool` 函数。这些函数能在 null/true/false 与 null/0/1 之间进行转换。如果你在对象浏览器中使用测试函数，这个转换会为你自动产生。

## 5.4 保存测试脚本

要保存测试脚本，请按下保存工具栏中的保存按钮。保存对话框使用 `.tst` 作为测试脚本的默认扩展名，但你可以使用第 20.27 节中描述的首选项来更改默认扩展名。PL/SQL 块、所有变量（名称、类型与值）以及调试器监视点（请参见第 5.10 节）都被保存下来。由于保存了测试脚本，你以后若要修改它，就能容易地重新测试程序单元。

通过按下工具栏中的 **打开** 按钮并选择 **测试脚本**，可以打开以前保存的测试脚本，这将创建一个新的测试窗口。也可以右键单击前面创建的测试窗口，选择 **载入项目**，这将在现有的测试窗口中打开测试脚本。

### 保存为 SQL\*Plus 脚本

你还可以用 SQL\*Plus 兼容格式保存测试脚本，这允许你在不能用 PL/SQL Developer 的环境中（例如在 Unix 服务器上）运行脚本。为此，请在文件选择器对话框中选择 **SQL\*Plus 脚本 (\*.sql)** 作为文件类型。前一章的示例脚本可以按如下方式保存：

```
rem PL/SQL Developer Test Script

set feedback off
set autotrace off

rem Declare variables
variable result varchar2(2000)
variable p_empno number

rem Set variables
begin
    :result := null;
    :p_empno := 7369;
end;
/

rem Execute PL/SQL Block
begin
    :result := employee.deptname(:p_empno);
end;
/

rem Print variables
print result
print p_empno
```

**注：**你不能再次把 SQL\*Plus 脚本作为测试脚本打开。如果你要重新使用测试窗口，你最好把它保存为测试脚本！

## 5.5 跟踪运行时错误

无论何时你的程序单元引起了运行时错误，PL/SQL Developer 都将询问你是否希望查看错误堆栈的来源。这个方法使你能够快速查找错误的原因。如果在我们的 *deptname* 函数中有一个错误，则错误堆栈窗口将显示如下：

```

function deptname(p_empno in emp.empno%type) return dept.dname%type
result dept.dname%type;
begin
  if p_empno is null
  then
    -- If empno is null, return an empty name
    result := 'null';
  else
    -- Fetch the name of the department
    select dept.dname into result
      from dept, emp

```

在顶部，可以看到每个与运行时错误相关的程序单元的标签，程序单元按从右往左的顺序被调用。你可以通过标签来浏览，这样很容易地找到导致错误的程序流。在本例中，它显示测试脚本已经调用了函数 *employee.deptname*。这显然能获得更复杂的、甚至通过触发器的信息。

**注：**如果你使用的是 Oracle 服务器 7.2 或者更早版本，触发器源将不会显示在错误堆栈窗口中。

错误堆栈以红色高亮显示与错误相关的每一行。对于堆栈中的最后一个程序单元，这是引起错误的行。对于所有其他程序单元，这是调用下一个程序单元的行。

## 5.6 包状态和 Java 会话状态

当你编辑、编译、测试包时，包状态在每次编译后将被 Oracle 抛弃，所有包中的全局变量都将被复位，并且初始化块将被再次运行。这可能导致意外的结果。无论 PL/SQL Developer 何时检测到这种情况，在测试窗口的状态行都会显示一个警告：“警告：ORA-04068：现有的包状态已经被抛弃”。

同样地，如果你编辑和编译了 Java 源文件，任何现有的 Java 会话状态都会被清除。PL/SQL Developer 也将处理这种情况，将在状态行上显示：“警告：ORA-29549：类<类名>已经被更改，Java 会话声明已被清除”。

## 5.7 查看结果集

在测试脚本中，并不仅限于执行 PL/SQL 块。你还可以运行单个 SQL 语句，包括 select 语句。运行 select 语句将显示一个附加的结果标签页，这中显示了所有的已选行。

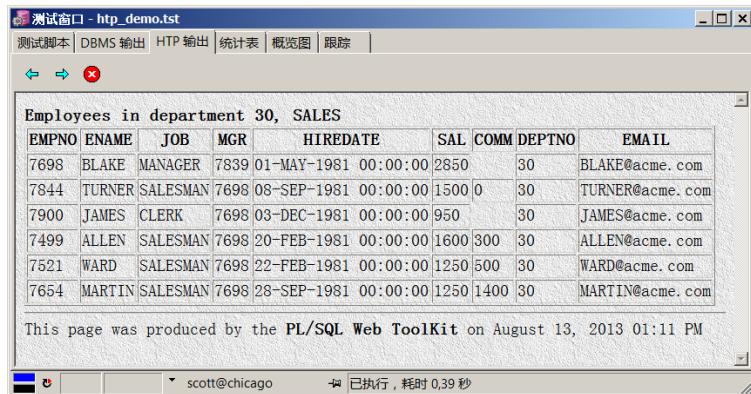
## 5.8 查看 dbms\_output

出于调试目的，将某些信息从程序单元内部“打印”到屏幕上可能很有必要。为此，Oracle 创建了 *dbms\_output* 包。通过调用 *dbms\_output.put\_line*，信息将被放进输出缓冲区。在运行测试脚本后，可以选择测试窗口顶部的输出标签来查看输出缓冲区的内容。在此页上，你还可

以设置输出缓冲区的大小(无限的自动设定为 0)或允许/禁用缓冲。输出页上的默认设置受第 20.5 节中所述的输出首选项控制。

## 5.9 查看 HTP 输出

如果想要利用 PL/SQL Web 工具箱来测试程序单元，将显示 HTP Output 标签页（如果有 HTP 输出的话）：



如果没有 HTP 输出，这个标签页将不可见。

## 5.10 调试

对于确实难以跟踪的编程错误，测试窗口提供了一个集成调试器。在窗口的顶部，可以看到一个具有与调试器相关的功能的工具栏。要开始调试会话，只需按下调试工具栏左边的开始按钮即可，无需按下主工具栏中的执行按钮。现在，其他按钮都会被激活，你可以准备调试了。

### 控制运行

启动调试器以后，运行将暂停于测试脚本的第一条语句之前，你可以使用调试工具栏中的按钮来控制运行：

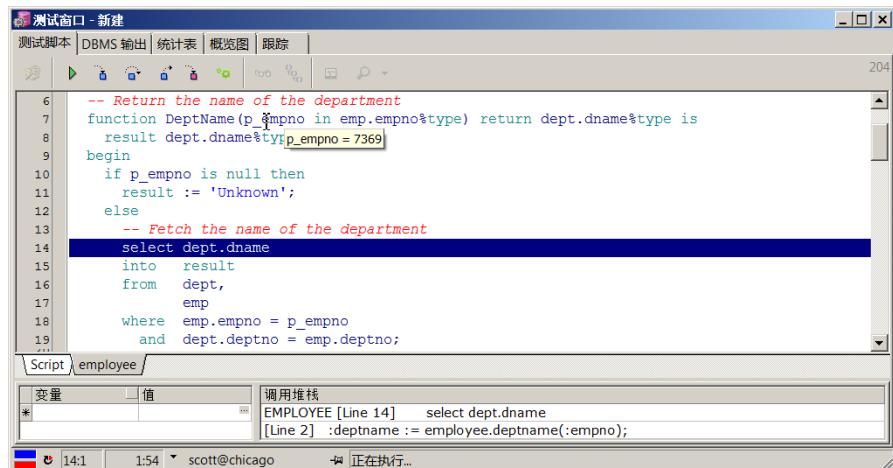
- 运行脚本直到完成。
- 单步进入下一行上的过程、函数或方法调用。如果下一行包含将触发触发器的更新、插入、或删除语句，你将单步进入这个触发器。
- 单步跳过下一行。该行将被运行，但你不会单步进入源代码。
- 单步退出当前的程序单元。

运行到光标行。

- 运行脚本直到出现异常。运行将被暂停在引起异常的行。在执行下一步之后，将实际出现异常。

无论你何时单步进入程序单元，它的源代码都将自动地载入测试窗口中。现在，编辑器面板的底部将显示每个程序单元的标签，因此你能容易地在它们之间互相切换，以查看源、设置/删除

断点等。通过右键单击编辑器面板，你能从测试窗口删除程序单元（如果你不再对它感兴趣的话）：



### 查看和设置变量值

要在调试会话期间查看变量值，可以将鼠标指针悬停在源代码中的变量上。半秒以后，它的值将在一个弹出窗口中显示出来。测试脚本的 PL/SQL 块中的变量可能从不显示。复杂的变量（例如记录与对象类型）也不能被显示，但是可以查看单独的字段。未来版本的 Oracle 也许会解除这些限制。

要设置变量值，在程序源代码中右键单击它的名称，在弹出式菜单中选择 **设置变量** 项目，这将显示一个带有变量当前值的输入区，可以输入一个新值并按回车键来应用它，或者按 **ESC** 取消操作：

**新值 RESEAR**

在同一个弹出式菜单中，你还可以选择将变量添加到监视列表，这意味着在每个调试步骤之后，变量值将在测试窗口左面底部的监视列表中自动显示和更新。

如果变量是一个标量数据类型的集合（PL/SQL 表、数组或嵌套表），你可以通过右键单击它并从弹出式菜单选择 **查看集合变量** 来查看整个集合。

注意，如果被编译的程序单元带有调试信息，择仅能查看和设置变量值。首选项的存在会使每一次编译都包含调试信息。你可以采用手工方式添加调试信息：右键单击对象浏览器中的程序单元，从弹出式菜单上选择 **添加调试信息** 项目。

### 使用断点

断点可用于在你的 PL/SQL 代码的某一行上暂停程序执行。当运行暂停时，可以查看和设置变量、单步调试代码等。你可以定义断点的条件，当这个条件被满足时，运行将被暂停。对于每个断点，你可以定义一个在断点到达时将放在输出页中的消息。

## 设置断点

有两种设置断点的方法：在程序窗口中或在测试窗口中设置。对于这两种方法，你只需简单地单击编辑器左边适当的行的空白处就可以。断点标记将显示出来，这标志着已在该行上设置了断点。当你在调试模式中运行测试窗口时，如果遇到了一个断点，运行将停止。

如果你在程序窗口中设置断点，则可能这个特定的程序单元还没有被编译到数据库中。因此这种断点现在还不能被应用到数据库中。在这种情况下，断点标志将具有不同的外观。当你在后来编译程序单元时，将应用设置的断点，相应的标志将更改以反映此情况。只要程序单元不被编译，任何前面在这个程序单元中设置的断点都是有效的。

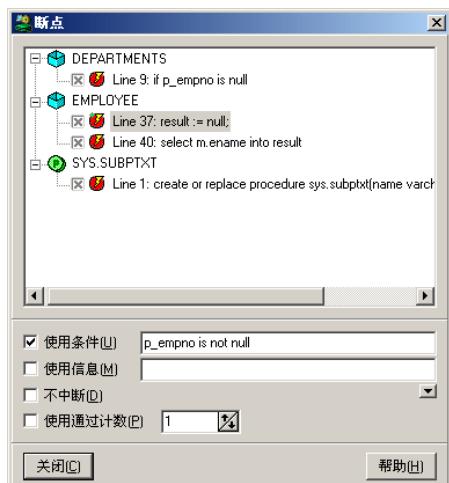
你不能在测试脚本的 PL/SQL 块中设置断点。

要删除断点，再次单击断点标志即可。

## 断点条件

有时候即使你仅对某一环境下的程序状态有兴趣，你也需要在经常执行的行上定义断点。在这种情况下，你可以为断点定义一个条件，运行仅仅在条件满足时才能暂停。

要为断点定义条件，请右键单击断点标志，从弹出式菜单选择 *修改断点*项目，将显示下面的对话框：



在这个对话框中，可以看到所有具有断点的程序单元，断点在它们下面列出。每个断点都有一个可用于启用或禁用它的复选框。如果在断点上双击，将会进入程序窗口中的源代码位置。底部区域显示如下字段：

- 使用条件** - 这个复选框启用或禁用条件，条件本身应该是布尔表达式。当到达断点行时，只有当条件值为 True 时，运行才被暂停。在条件中，你可以使用任何 SQL 表达式，还可以使用任何在断点位置已知的变量。在交互式调试期间，你可以查看或设置这些变量。变量前面应该加上冒号。例如，如果 *ename* 在断点位置是个有效的变量，则 *upper(:ename) = 'SMITH'* 是一个有效的条件。
- 使用消息** - 这个复选框启用或禁用消息。当到达断点行时，如果条件满足，将在输出页上显示消息。

- 不中断** - 这个复选框只在你也定义了一条消息时有用。当选中它时，程序执行不会在该断点行上暂停。通过此方法，你可以定义仅仅在输出页上生成消息的断点。
- 使用通过计数** - 这个复选框启用或禁用通过计数，通过计数定义断点行在运行被暂停前必须被通过多少次。例如，如果你定义通过计数为 10，每到达断点行 10 次就暂停执行一次。

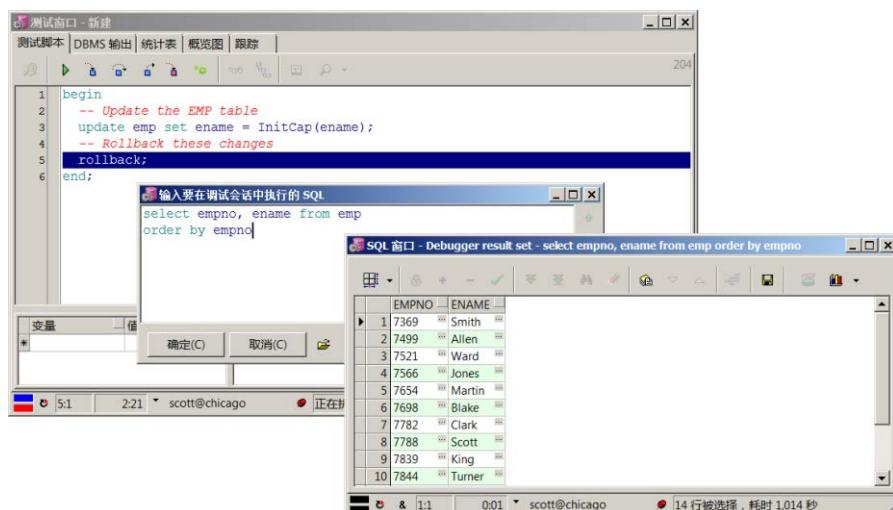
### 查看调用堆栈

可以在测试窗口的右下部查看调用堆栈，它将在每个调试步骤执行后自动更新。

### 在调试模式中执行 SQL

调试会话执行在断点处停止，或者逐步通过代码执行行时，你可以在程序执行点处执行 SQL 语句。例如，可以更改会话状态、插入/更新/删除记录、提交或回滚事务等。还可以执行 select 语句来查看调试会话内的结果集。

为此，请按调试工具栏中的在调试会话中执行 SQL。此时将显示 SQL 编辑器，允许输入 SQL 语句或 PL/SQL 块。如果 SQL 语句为 select 语句结果随后将显示在 SQL 窗口中：



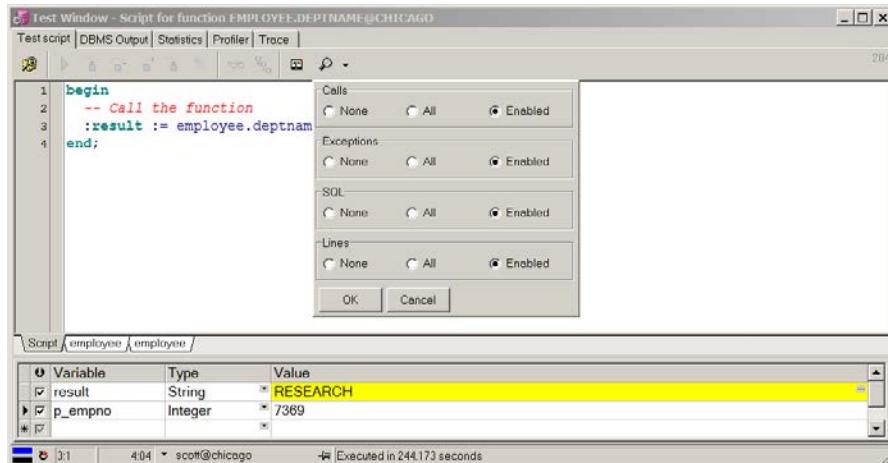
请注意，所有字段都将显示为限制为 1000 个字符的字符串。数字和日期值将使用调试会话的 NLS 格式设置转换为字符串。BLOB 或 LONG RAW 数据类型的字段不显示。

### 调试器首选项

在工具菜单的首选项项目中，可以看到与调试器相关的首选项部分。这些首选项将在第 20.4 节中详细描述。

## 5.11 跟踪运行

你可以使用测试窗口的跟踪工具跟踪你的 PL/SQL 代码的运行。你可以设定要跟踪的事件：调用、异常、SQL 或者甚至每个 PL/SQL 代码的已执行行。按下选择跟踪级别按钮，接下来按下创建跟踪报告按钮，以调出下面的配置屏幕：



你可以控制跟踪每个程序单元的特定事件（全部）、仅跟踪正在编译的具有调试信息的程序单元中的特定事件（允许），还是从不跟踪（无）。

要创建跟踪报告，只需按下测试窗口工具栏中的创建跟踪报告按钮并执行你的测试脚本。在运行完后，可以转到跟踪标签页查看报告，以及查看前面运行的报告：

Comment	Event time	Unit name	Unit line	Unit line text	Except
PL/SQL Virtual Machine started	22-08-2012 15:56:38				
PL/SQL Virtual Machine started	22-08-2012 15:56:38				
Procedure Call	22-08-2012 15:56:38		3		
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	10	function deptname(p_empno in emp.empno%type	
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	12	begin	
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	13	if p_empno is null	
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	19	select deptname into result	
SELECT DEPT.DNAME FROM DEP	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	19	select dept.dname into result	
Exception raised	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	19	select dept.dname into result	1403
Exception handled	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	31	when no_data_found then return(null);	1403
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	29	exception	
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	31	when no_data_found then return(null);	
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	31	when no_data_found then return(null);	
New line executed	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	32	end;	
Return from procedure call	22-08-2012 15:56:38	EMPLOYEE.DEPTNAME	32	end;	
PL/SQL Virtual Machine started	22-08-2012 15:56:38				

在这个（简单的）报告中，你可以看到函数 `employee.deptname` 的第 19 行的 select 语句出现了异常 1403（没有找到数据），这个异常在同一程序单元的第 31 行被处理了。

对于每个被跟踪的事件，你可以在报告中包含一些信息，比如事件描述、时间、程序单元、行数、异常等。要配置此信息，请按下跟踪工具栏中左上方的首选项按钮，这将调来如将在第 20.6 节中描述的跟踪首选项页。

要查看旧的跟踪报告，从运行选择列表选择前面的跟踪运行。按下删除运行按钮可以删除当前已选的运行。

要获得关于 Oracle 跟踪工具的更多信息，请参见 *Oracle 供应包参考手册* 中的 *DBMS\_TRACE*。

## 5.12 回归测试

要使用测试脚本进行回归测试，你可以使用测试管理器工具（请参阅第 21.15 节）来定义和运行基于测试脚本的测试集。

## 6. 优化

要优化程序单元中的 SQL 语句，Oracle 的解释计划工具对显示语句的运行路径有很大帮助，因此它已被集成在 PL/SQL Developer 的 IDE 中。

要查看 SQL 语句或 PL/SQL 程序单元的实际资源使用情况，PL/SQL Developer 可以显示关于它的运行情况的统计表。你可以配置你希望显示的统计表，并且可以包括耗时、CPU 时间、逻辑读、物理读、物理写等。

要确定每个单独的 PL/SQL 代码行的运行时间，你可以使用 PL/SQL 概览图。这个特性对 Oracle 8.0 或更早版本不可用。

最后，通过启用 SQL 跟踪，你可以使用 Oracle 的 *tkprof* 工具来获得程序单元中关于所有已运行的 SQL 语句的资源使用信息。

### 6.1 使用解释计划窗口

要使用 Oracle 的解释计划工具，请按下工具栏中的**新建**按钮，选择**解释计划窗口**。将显示一个空的解释计划窗口。在窗口的上半部，可以键入你希望分析的 SQL 语句。在按下工具栏中的**执行**按钮后，运行计划将窗口的下半部中显示出来：



The screenshot shows the 'Explain Plan' window in PL/SQL Developer. The top pane contains the SQL query:

```
select d.dname
  from dept d, emp e
 where e.empno = 7369
   and e.deptno = d.deptno
```

The bottom pane displays the execution plan with the following details:

Description	对象所有者	对象名称	耗费	基数	字节
SELECT STATEMENT, GOAL = FIRST_ROWS			2	1	48
NESTED LOOPS			2	1	48
TABLE ACCESS BY INDEX ROWID	SCOTT	EMP	1	1	26
INDEX RANGE SCAN	SCOTT	PK_EMP	1	1	
TABLE ACCESS BY INDEX ROWID	SCOTT	DEPT	1	1	22
INDEX UNIQUE SCAN	SCOTT	PK_DEPT		1	

A message at the bottom says: "Select a range of values from an index in ascending order".

你现在可以更改 SQL 语句，再次按下**执行**按钮查看这次更改的影响。要了解关于解释计划的更多信息，可以阅读 *Oracle 服务器调优手册*。

要查看查询计划中各种优化器目标的效果，请在**优化器目标**列表中选择相应条目，计划将立即更新，以反映这些变化。

使用**第一个**、**下一个**、**上一个**和**最近操作**按钮，按操作顺序导航查询计划。计划被确定后，第一个操作将被高亮显示。

解释计划工具使用所谓的计划表来储存运行计划。如果这个表对当前用户不可用，PL/SQL Developer 将询问你是否在当前用户的计划中创建这个表。要定义想要查看计划表中的哪些列，按什么顺序查看，请按下**首选项**按钮，这将调出相应的首选项页（请参见第 20.21 章）。

如果右键单击**对象名称**列，将为已选对象调出弹出式菜单。在 Oracle11g 服务器上，后续可以采用 HTML、文本或 XML 格式另行查看查询计划。

## 分析程序单元中的 SQL 语句

在大多数情况下，你希望分析的 SQL 语句都被包含在程序单元的源代码中。为此，你可以在程序编辑器中选择 SQL 语句并从帮助菜单选择解释计划（或按下 F5），将为已选的 SQL 语句创建一个新的解释计划窗口，而且所有的 PL/SQL 变量都会被绑定变量替换掉。

如果你从程序单元明确地复制和粘贴一个包含 PL/SQL 变量的 SQL 语句并执行解释计划窗口，你会收到一个“ORA-00904：无效的列名称”的错误。解释计划工具错误地把这些变量都当成了列，因此每个变量都必须加上一个冒号前缀。如果我们希望分析 `employee.deptname` 函数中的 SQL 语句，则 `result` 与 `p.empno` 变量肯定会按如下方式发生更改：



```

解释计划窗口
select d.dname
  from dept d, emp e
 where e.empno = :p_deptno
   and e.deptno = d.deptno

优化器目标 First rows
Description | 对象所有者 | 对象名称 | 耗费 | 基数 | 字节
---|---|---|---|---|---
SELECT STATEMENT, GOAL = FIRST_ROWS |  |  | 2 | 1 | 48
  NESTED LOOPS |  |  | 2 | 1 | 48
    TABLE ACCESS BY INDEX ROWID | SCOTT | EMP | 1 | 1 | 26
      INDEX RANGE SCAN | SCOTT | PK_EMP | 1 | 1 |
    TABLE ACCESS BY INDEX ROWID | SCOTT | DEPT | 1 | 1 | 22
      INDEX UNIQUE SCAN | SCOTT | PK_DEPT | 1 | 1 |

Select a range of values from an index in ascending order

```

现在解释计划工具知道哪一个标识符是列、哪个标识符是 PL/SQL 变量。

## 6.2 自动统计

当你在 SQL 窗口或测试窗口执行语句时，PL/SQL Developer 将自动生成此执行的一个统计报告。一个前提是必须在动态性能表 `v$session`、`v$statname` 和 `v$sesstat` 中有 select 权限（通过标准 `plustrace` 角色提供）。

你可以转到 SQL 窗口或测试窗口顶部的统计表标签来查看统计报告。`employee.deptname` 函数运行的统计表可能如下所示：



名称	最近	合计
physical reads	0	0
physical writes	0	0
table scans (short tables)	1	3
table scans (long tables)	0	0
table scan rows gotten	14	16
table scan blocks gotten	1	3
table fetch by rowid	0	3
sorts (memory)	0	5
sorts (disk)	0	0
sorts (rows)	0	12
session logical reads	3	23
CPU used by this session	0	0

已执行，耗时 0.015 秒 (14 行被选择)

对于每个统计值，你可以看到最近运行的值和当前会话的总数。你可以通过设置首选项来配置是否显示统计表或显示哪个值，如第 20.2 节中所描述的那样。统计表的默认设置与你的 SQL 和 PL/SQL 有关，其描述如下：

统计值	含义
CPU used by this session	CPU 使用情况 (以微秒为单位)
Physical reads	从磁盘读出的块数
Physical writes	写入磁盘的块数
session logical reads	从块缓冲区或磁盘中读出的块数
sorts (disk)	在临时分段中运行的排序次数
sorts (memory)	在内存中执行的排序次数
sorts (rows)	排序的行数
table fetch by rowid	按 rowid 获取的行数, 通常作为索引访问的结果
table scan blocks gotten	为全表扫描读取的块数
table scan rows gotten	为全表扫描读取的行数
table scans (long tables)	在长表上执行全表扫描的次数
table scans (short tables)	在短表上执行全表扫描的次数

你能包含的其他统计表都依赖于 Oracle 服务器的版本, 不在本手册中被描述。如果你希望获得关于它们的信息, 可以查看许多介绍此主题的 Oracle 调优书籍, *Oracle 服务器参考手册*也简要地描述了这些统计表。

统计表能被导出为 CSV 文件 (逗号分隔值), 以便以后能用电子表格软件打开它。只要右键单击统计表、选择导出项目、选定 CSV 文件项目即可。你也可以选择 TSV、XML 或 HTML 格式, 或者选择复制它到剪贴板。

## 6.3 PL/SQL 概览图

PL/SQL 概览图是一个帮助你优化 PL/SQL 代码的强大工具。对于每个已运行的代码行, 合计时间、最长时间、最短时间、平均时间和运行次数都将被确定。

可以在测试窗口轻松访问概览图。在运行测试脚本前, 只需按下测试窗口工具栏中的创建概览图报告按钮即可。如果你随后运行脚本, 可以转到概览图页查看报告。

下面的示例报告显示了在 `employee.deptname` 函数中, `select` 语句用时 95 毫秒, 3 个 `dbms_output` 调用用时 44 毫秒:

单元	行	合计时间	具体值	文本
ANONYMOUS BLOCK	1	12	2	
ANONYMOUS BLOCK	2	28	2	
EMPLOYEE	9	1	1	1 if p_empno is null
EMPLOYEE	15	95	95	1 select dept.dname into result
EMPLOYEE	20	44	44	3 dbms_output.put_line(to_char(sysdate, 'dd-mm-yyyy hh24:mi:ss'))
EMPLOYEE	24	1	1	1 return(result);
EMPLOYEE	28	1	1	1 end;

已执行, 耗时 0 秒

在默认情况下, 概览图页将显示最近运行的报告。你也可以从运行列表中选择以前的运行。单元列表允许你放大一个运行的特定的程序单元。

合计时间列显示了该行的相对时间的图形化描述，这个图形与最长时间作为参照，这允许你快速确定最耗时的行。可以按下列标题中的排序按钮来排序报告。

如果源行显示为红色，这意味着自概览图被创建以后程序单元已经被更改，因此现在显示的代码行与概览图被创建时不同。

通过按下首选项按钮，你可以更改概览图报告的各种布局元素，这将调出相应的首选项页，如第 20.7 节中所述。

关于 PL/SQL 概览图的更多信息，请参见“*Oracle 8i 应用包参考*”手册中的 *dbms\_profiler*。

注：不是所有平台都能提供精确的时间信息。

## 6.4 SQL 跟踪

你可以按下工具栏中的 *SQL 跟踪* 按钮来启用 SQL 跟踪，此后由 SQL 窗口或测试窗口引起的所有服务器进程都将记录在数据库服务器上的跟踪文件中。一个前提条件是，你正在使用的数据库实例的定时统计参数必须被设成 True。

你可以再次按下 *SQL 跟踪* 按钮来禁用 SQL 跟踪。

你可以使用数据库服务器上的 Oracle *tkprof* 工具来查看跟踪文件中的信息，它将为每条 SQL 语句产生一个关于耗时、CPU 时间、I/O 等的报告，因此它能使你知道程序单元中哪条语句最耗时。要了解更多关于 Oracle *tkprof* 的知识，可以查阅“*Oracle 7 服务器调整*”手册。

运行 *employee.deptname* 函数的 *tkprof* 报告可能如下所示：

TKPROF: Release 7.2.2.3.1 - Production on Fri Sep 26 14:59:08 1997

Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Trace file: ora07087.trc

Sort options: default

```
*****
count    = number of times OCI procedure was executed
cpu      = cpu time in seconds executing
elapsed  = elapsed time in seconds executing
disk     = number of physical reads of buffers from disk
query    = number of buffers gotten for consistent read
current  = number of buffers gotten in current mode (usually for update)
rows     = number of rows processed by the fetch or execute call
*****
```

```
begin
  :deptname := employee.deptname(:empno);
end;
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.01	0.01	0	0	0	1
Fetch	0	0.00	0.00	0	0	0	0
total	2	0.01	0.01	0	0	0	1

Misses in library cache during parse: 0

Optimizer hint: CHOOSE

Parsing user id: 16

```
*****
```

```
SELECT DEPT.DNAME
FROM
DEPT,EMP WHERE EMP.EMPNO = :b1 AND DEPT.DEPTNO = EMP.DEPTNO
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	5
Fetch	1	0.06	0.06	4	4	0	1
total	3	0.06	0.06	4	4	0	6

Misses in library cache during parse: 0

Optimizer hint: CHOOSE

Parsing user id: 16 (recursive depth: 1)

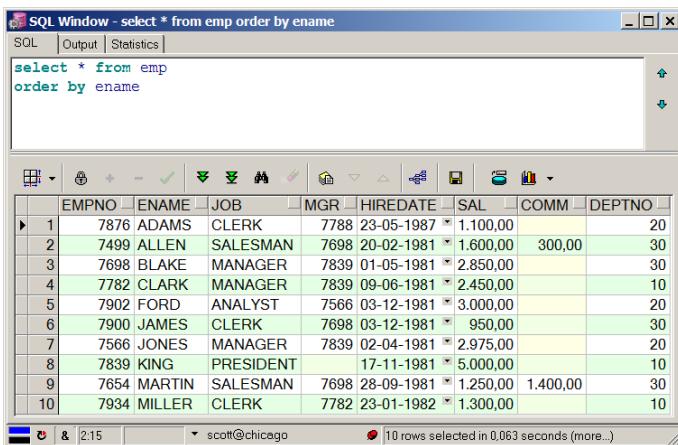
```
*****
```

## 7. 专用 SQL

在程序开发期间，你经常需要运行一些 SQL 语句。既可能是为了测试语句，又可能是为了查看或更新表中的数据。你可以在 PL/SQL Developer 内部使用 SQL 窗口完成此任务。

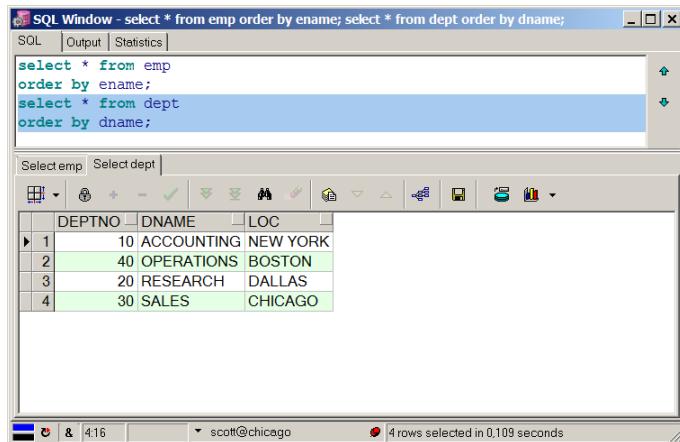
### 7.1 使用 SQL 窗口

要创建一个新 SQL 窗口，请按下工具栏中的新建按钮，再选择 *SQL 窗口*，将显示一个空 SQL 窗口。在这个窗口中，你可以键入 SQL 语句。如果它是 select 语句，窗口会分成两部分，下半部分包含显示结果的表格。如果我们要查看雇员表中的所有雇员，SQL 窗口将显示如下：



SQL 语句被运行，10 行数据被检索。若结果集大于 10 行，可以通过按下结果集工具栏中的获取下一页和获取最后页两个按钮来显示结果集，并且状态行上将显示（更多...）。最初只有 10 行被获取，这是因为能在表格中显示的记录数只有这些。按下获取下一页按钮将获取下 10 行数据等。按下获取最后页按钮将获取所有行。

如果你在 SQL 编辑器中有若干条 SQL 语句并且按下了执行按钮，所有语句都将被执行，结果将在不同的标签页中显示出来：



SQL 语句必须用分号(;)或者斜杠(/)来终止，PL/SQL 块必须用斜杠(/)来终止，这几乎与 SQL\*Plus 一样。如果你选择了一个结果标签，那么 SQL 编辑器中相应的文本就会被高亮显示。

如果你选择了 SQL 编辑器中文本的一部分，那么仅将执行被选择的文本。这样，你可以在编辑器中有多条语句，并仍然逐个执行它们。

如果你执行的 SQL 语句要花很长时间才能完成，你可以按下中断按钮来终止它。注意按下中断按钮并不总是会成功。例如，如果语句正在等待锁，它将不会响应中断信号。如果你在 SQL 窗口正在检索行时按下了中断按钮，它将简单地停止并显示已经检索的行，你现在可以用下一页和最后页来继续检索记录。要刷新（重新执行）当前选定结果标签页的查询，可以按刷新结果集按钮

注意，SQL 窗口首选项允许你确定一个 select 语句最初可以获取多少行，这将在第 20.19 节中描述。

在窗口的右边可以看到两个按钮，它们允许你操纵已经输入 SQL 窗口中的所有 SQL 语句，这样你可以快速地再次运行预先输入的语句。

为了进行优化，你可以选择统计表标签来查看 SQL 语句运行情况的统计表。统计表在第 6.2 节中解释。

当你打印 SQL 窗口时，SQL 语句和结果表格都将被打印。通选择结果表格中的行，你可以限制将要打印的行数。

标签的标题将会根据计算机程序指令自动默认。想要重置，你可以在计算机程序指令前编写 tab=<title>语句。比如：

```
-- tab=Department 10
select * from emp
where deptno = 10;
```

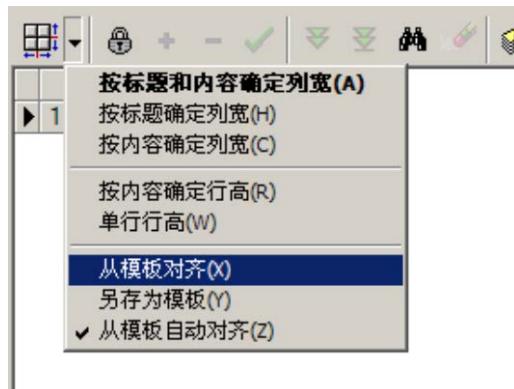
这样，标签标题就由默认的 Select emp.. 变成了 Department 10

## 7.2 结果表格处理

SQL 窗口的结果表格可以用不同的方法来处理。一些单元格类型具有特殊的行为，行、列和单元格排列都能被选择和打印，列可以被移动，行可以被排序，你可以切换到单一记录视图等。

### 对齐单元格

要根据列值快速对齐单元格，可以使用对齐按钮：



若按下此按钮，各单元格的宽度将由列标题和内容确定。如果按下对齐按钮旁边的下拉按钮，将可从下拉菜单中选择不同的对齐选项。可将宽度限制为与标题或内容对齐，也可对齐行高。

还可以为频繁使用的特定查询创建自定义模板对齐。在设置单元格宽度之后，可选择另存为模板菜单项来保存设置。如果此后执行相同的查询，可通过选择与模板对齐选项显式应用此对齐，如果启用了自动与模板对齐选项，也可隐式应用。请注意，除了列宽之外，模板还将保存列顺序。

隐藏编辑器选项可用于隐藏 SQL 编辑器，以最大化结果集的可用空间。调整窗口宽度选项可用于调整 SQL 窗口宽度以匹配结果集的宽度。

### 识别 null 值

Null 值显示在一个具有浅黄色背景的单元格中，所以你可以快速从全是空格的值中区别出它们。这对识别 null 值和在单元格中不直接显示的值也很有用，如 long 值和 LOB 值。你可以通过首选项来更改 null 值单元格的颜色（请参见第 20.19 节）。

### 查看大数据列

Long、Long Raw、CLOB、BLOB 与 BFILE 列的值不直接在结果表格中显示，表格中只简单地显示为 <Long>、<Long Raw>、<CLOB>、<BLOB> 和 <BFILE>：

BCC_ADDRESS
<BLOB>
<BLOB>
<BLOB>
<BLOB>

当你单击某一列的单元格按钮时，将调出大数据编辑器，它允许你查看或编辑不同格式的数据。更多细节请参见第 23 章。

你也可以单击字符列上的单元格按钮来调用大数据编辑器，不过单元格按钮仅适用于大于 20 个字符的列。

### 查看日期列

日期列有一个显示日历的单元格按钮，这个日历的当前日期已被高亮显示。如果日期还有时间部分，你也可以在日历上查看它的值：



### 查看时间戳列

时间戳列显示在指定的格式中，这个格式在你的 Oracle 主目录的 NLS\_TIMESTAMP\_FORMAT 和 NLS\_TIMESTAMP\_TZ\_FORMAT 的注册表设置中指定。

### 查看 XML 数据

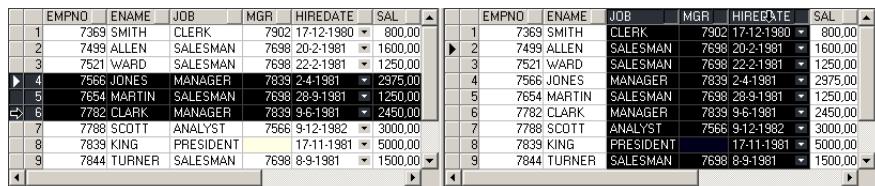
在数据库中储存 XML 数据有多种方式，可以存储为：CLOB (Oracle8i)、XMLTYPE (Oracle9i) 或者甚至 Varchar2、Long 列。只要 XML 格式是以标准的 XML 标题开始，大数据编辑器始终能识别出它并转为 XML 格式。要了解更多详细信息，请参见第 23 章。

### 查看嵌套的游标列

如果你在 select 语句的字段列表中包含了嵌套游标，列值在最初将被显示为 <Cursor>。按下单元格按钮将调出新的 SQL 窗口，它带有一个包含游标结果集的结果表格，这可用于查看简单的嵌套查询。注意，每个嵌套游标都将隐式地打开一个游标，因此对于大的结果集，你很容易陷入 OPEN\_CURSORS 的限制，并且得到 “ORA-01000: 超出最大打开游标数量”的错误。

## 选择行、列和单元格

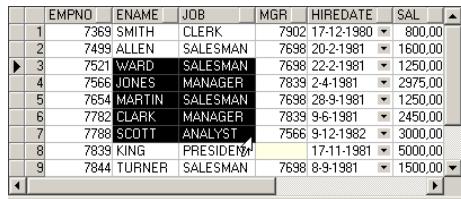
要在结果表格中选择行或列，只需单击行标题或列标题并拖动鼠标指针，以高亮显示要选择的区域：



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800.00
2	7499	ALLEN	SALESMAN	7698	20-2-1981	1600.00
3	7521	WARD	SALESMAN	7698	22-2-1981	1250.00
4	7566	JONES	MANAGER	7839	2-4-1981	2975.00
5	7654	MARTIN	SALESMAN	7698	28-9-1981	1250.00
6	7782	CLARK	MANAGER	7839	9-6-1981	2450.00
7	7788	SCOTT	ANALYST	7566	9-12-1982	3000.00
8	7839	KING	PRESIDENT		17-11-1981	5000.00
9	7844	TURNER	SALESMAN	7698	8-9-1981	1500.00

现在，可以移动选择的列了，只需释放鼠标按钮，再次单击一个已选列标题，并将选择的列拖到新位置。

要选择特定范围内的单元格，可以将鼠标指针移动到一个单元格的左边，直到光标形态更改，按下鼠标按钮，拖动鼠标以高亮显示所选区域：



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800.00
2	7499	ALLEN	SALESMAN	7698	20-2-1981	1600.00
3	7521	WARD	SALESMAN	7698	22-2-1981	1250.00
4	7566	JONES	MANAGER	7839	2-4-1981	2975.00
5	7654	MARTIN	SALESMAN	7698	28-9-1981	1250.00
6	7782	CLARK	MANAGER	7839	9-6-1981	2450.00
7	7788	SCOTT	ANALYST	7566	9-12-1982	3000.00
8	7839	KING	PRESIDENT		17-11-1981	5000.00
9	7844	TURNER	SALESMAN	7698	8-9-1981	1500.00

要选择所有的行和列，请按下左上角的单元格，或者右键单击表格并从弹出式菜单选择选择全部项目。

高亮显示的选择区域通常都能被复制或打印。

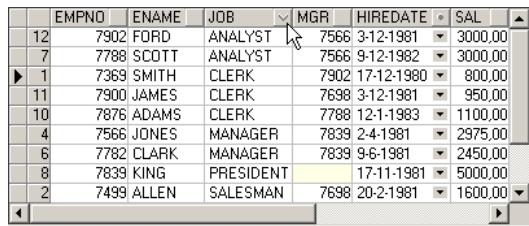
## 滚动锁定

有时候你并不想一些重要的数据在屏幕上滚动。比如当你滚动网格内容到左边页面的时候不想empno纵列的在你的屏幕上显示。你只需右击 empno 纵列，从滚动锁定弹出来的菜单选项中选择锁定纵列。想要锁定多个纵列您可以锁定最右边的纵列。

你也可以通过同样的方法锁定行：右击行，从滚动锁定中弹出来的菜单选项中选择锁定行。

## 排序行

要在结果表格中对行进行排序，请按下你想要根据其对行进行排序的列标题按钮：



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
12	7902	FORD	ANALYST	7566	3-12-1981	3000.00
7	7788	SCOTT	ANALYST	7566	9-12-1982	3000.00
1	7369	SMITH	CLERK	7902	17-12-1980	800.00
11	7900	JAMES	CLERK	7698	3-12-1981	950.00
10	7876	ADAMS	CLERK	7788	12-1-1983	1100.00
4	7566	JONES	MANAGER	7839	2-4-1981	2975.00
6	7782	CLARK	MANAGER	7839	9-6-1981	2450.00
8	7839	KING	PRESIDENT		17-11-1981	5000.00
2	7499	ALLEN	SALESMAN	7698	20-2-1981	1600.00

按下列标题按钮之后，行将按升序排列；如果再次按下列标题按钮，行将按降序排列；第三次按下时，行将撤消排序。按另一列的标题按钮将对这个列所在的行进行排序，但也将使用前面

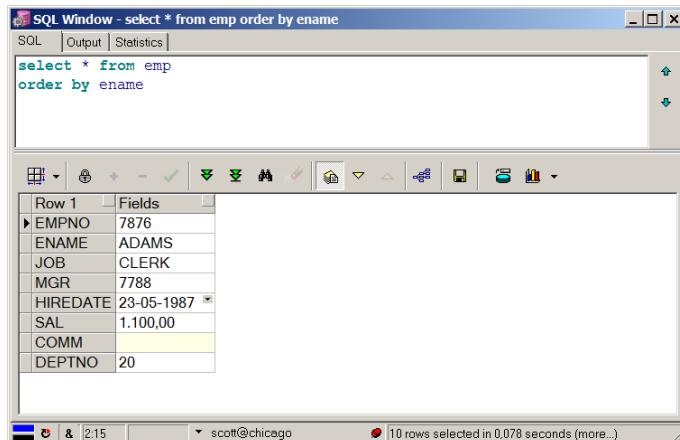
的排序列为次要排序列（标题按钮上显示为一个圆点）。在上面的例子中，列 *job* 是主要排序列，列 *hiredate* 为次要排序列。默认情况下，排序的顺序不区分大小写。右键单击表格并选择区分大小写的排序菜单项即可更改排序行为。

注意，排序是在本地被运行的，并且仅能对已检索到的行进行排序。如果你在对结果排序后又检索到了另外一些行，这些新行将被添加到结果表格的末尾，不会被排序。对于较大的结果集，本地排序可能要花很长时间。在这种情况下，最好能在 select 语句中使用 order by 子句，让 Oracle 服务器来排序。

## 单个记录查看

如果结果集包含很多列，每条记录都显示在一行上也许不太方便，你必须前后滚动相关的列（你也可以移动列），无法立即看到记录的所有列。

通过按下表格工具栏中的单个记录查看按钮，可以一次查看单条记录：



现在每行显示一列名和值，以及列注释（若适用）。表格工具栏中的下一个记录和上一个记录可用于操纵结果集。要转回到多记录查看，就再按一次单个记录查看按钮即可。

## 导出数据

可以采用多种方式来导出结果表格中的数据。运行 select 语句后，你可以采用如上面所述的方法选择一定范围内的单元格，右键单击它，从弹出的菜单上选择导出结果选项。会弹出一个子菜单，在其中你可以选定导出数据为 CSV 格式（逗号分隔值）、TSV 格式（制表符分隔值）、HTML 格式、XML 格式或 SQL 格式。选择了格式后，可以指定导出文件。SQL 格式将生成一个 SQL 文件，文件内容包含针对表或视图的查询语句。弹出菜单还可能包含输出到剪贴板子菜单项，允许直接以给定格式导出到剪贴板。

也可以按下 Ctrl-C，或者右键单击你选择的数据，从弹出式菜单中选择复制或带标题复制选项，把你选择的数据复制到剪贴板上。你还可以把这些数据粘贴到另外的应用程序中，例如电子表格、文字处理软件等。

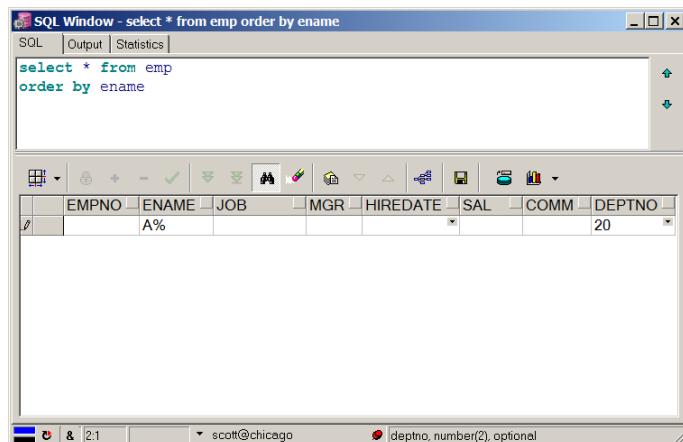
要在 Microsoft Excel 中快速对这些结果集信息进行操作，请选择复制到 Excel 选项，这会将选定的数据导出为临时文件，并在 Excel 中打开它。如果选定的数据超出 65535 条记录，则将拆分为

多个 Excel 页面。如果有多个结果集，也可使用全部复制到 Excel 菜单项，这将为每个结果集创建一个单独的 Excel 页面。无法通过首选项控制将在哪个目录中创建 Excel 文件（请参见第 20.26 节）。

如果希望仅将标题名复制到剪贴板，请右键单击选定列，选择弹出菜单中的 **复制标题**。

### 7.3 实例模式查询

运行 select 语句后，如果你要搜索特定的记录，你可以按下结果集上方的 **实例查询**按钮。按下这个按钮将清除表格，只剩下一个空记录，在其中，你可以输入你的查询值。例如，如果你要查找部门编码为 20、以字母 A 打头的所有雇员，可以在列 *ename* 中输入 A% 并在 *deptno* 列中输入 20：



再次按下 **实例查询** 按钮，或按下 **执行** 按钮，将运行已修改的查询，并显示限定的结果。再次按下 **实例查询** 按钮，前面的查询值仍然会被保留。按下 **清除记录** 按钮可以清除查询值。

查询值不会被限制为带有通配符的单一值，你可以使用下列表达式（也可以选择方括号中的表达式）：

- 值 [= 值]
- 带有通配符的值 [比如带有通配符的值]
- > 值
- != 值 [<> 值]
- in (值 1, 值 2, ...)
- between 值 1 and 值 2
- null [is null]
- not null [is not null]

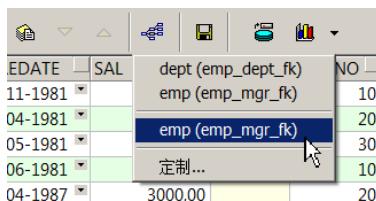
注意，如果你使用了文字值，则可以省略字符值的引号（例如，在 SQL 文本中，SMITH 将自动被转换成 ‘SMITH’）。对于所有其他表达式，你必须提供 Oracle 服务器能够理解的文字值（例如，!=’SMITH’）。对于数字和日期值也是如此。如果你要在表达式中使用它，必须使用 Oracle 服务器能够理解的格式。

几个首选项支持控制实例查询功能的默认行为（例如，区分大小写），第 20.19 节将描述这几个首选项。

## 7.4 链接查询

当查看一个结果集时，你经常需要查询相关的表。例如，当你查看 *dept* 表时，你也许需要查询一个部门中的所有雇员，反之亦然：当你查看 *emp* 表时，你也许需要查询一个雇员所在的部门。

多数情况下，这些父表与子表之间具有外键约束，这时 SQL 窗口能够自动为你生成和运行这些查询。如果你确定了结果集中一条特定记录的位置，按下表格工具栏中的链接查询按钮，这将显示一个弹出式菜单，其中包含当前 select 语句的表的所有父表和子表。例如，如果查询表 *emp*，则下面的弹出式菜单将显示出来：



上面部分显示了父表（和外键名），下面部分显示了子表。项目 *dept* 将为这个雇员的部门（30）生成一个查询，第一个 *emp* 项目（父表中的）将为这个雇员的经理（7839）生成一个查询，第二个 *emp* 项目（子表中的）将为被这个雇员管理的所有雇员生成一个查询。

The screenshot shows the SQL window with the title "SQL 窗口 - 子查询: emp (emp\_mgr\_fk)". It contains a query: "select \* from EMP t where t.mgr = :m\_empno". Below the query is a table with columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO. The table data is as follows:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
▶	1	7844	TURNER	SALESMAN	7698 08-09-1981	1500.00	0,00	30
	2	7900	JAMES	CLERK	7698 03-12-1981	950.00		30
	3	7499	ALLEN	SALESMAN	7698 20-02-1981	1600.00	300.00	30
	4	7521	WARD	SALESMAN	7698 22-02-1981	1250.00	500.00	30
	5	7654	MARTIN	SALESMAN	7698 28-09-1981	1250.00	1400.00	30

At the bottom, it shows "scott@chicago" and "5 行被选择, 耗时 0,343 秒".

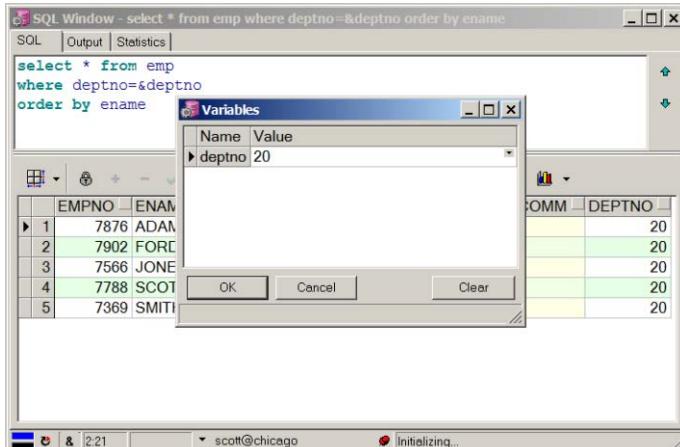
新的详细信息 SQL 窗口通过名称为 *:m\_<master\_field>*（本例中是 *:m\_empno*）的隐式绑定变量动态链接到主 SQL 窗口。如果你转到另外一条主记录，详细信息 SQL 窗口将自动刷新为新主记录。*:m\_empno* 绑定变量也会更新为使用对应的主字段值。

定制项允许编写定制 SQL 语句，使用上述 `:m_<field>` 名称引用主字段。

链接查询将在新 SQL 窗口中执行，除非你禁用了 SQL 窗口首选项在新窗口中执行链接查询（参见第 20.19 节）。当你按住链接查询按钮，并同时按下 `Ctrl` 键时，这个首选项将反向发生。只有在新窗口中打开链接查询时，才能创建动态主/详细信息 SQL 窗口。

## 7.5 置换变量

当查询被运行时，你可以在 SQL 文本中使用置换变量来允许用户输入数据。置换变量的最简单形式类似于你可能熟悉的 SQL\*Plus：



通过在 SQL 文本中指定的 `&deptno`，你将被提示为这个变量输入一个值。在运行前，SQL 文本中的变量参照将被置换成这个输入值。你可以指定不止一个置换变量，而且你可以不止一次地指定一个置换变量。此外，你还可以定义数据类型、默认值、选择列表、列表查询、复选框等其他高级选项。这些特性也被用于报告窗口，详细信息将在第 12 章描述。

要抑制 SQL 窗口中的变量置换，请按状态栏左下角的 `&`-按钮。`&-` 符号将变为红色，表示变量置换已经启用。再次按此按钮可重新启用变量置换。

## 7.6 更新数据库

要更新、插入或删除数据库中的记录，你可以在 SQL 窗口中运行适当的 DML 语句。通过将 `rowid` 包括在选择列表中或者使用 `select ... for update` 语句，使结果表格可以更新，可以为我们提供很大方便：

你应该意识到，一个 `select ... for update` 语句将锁定所有已选的记录，所以，一般来说，包括 `rowid` 可能是一种使结果集可更新的最好方法。如果 `select` 语句是一个内部联接，则第一个表的列将可被更新，所有的其他列都是只读的，有别名的列也不能被更新。

如果结果表格可更新，你可以按下结果表格右边的 `编辑数据` 按钮来编辑记录，你可以按下 `插入记录` 或 `删除记录` 按钮来插入或删除表格中的记录。如果你已经选择了多条记录，所有被选择的记录都会被删除。

在编辑字符或数字字段时，如果输入了超出基础列大小、精度或范围的数据，文本颜色将变为红色。

编辑在虚拟列表达式中出现的列值时，相关的虚拟列值也会同时更新。

实际上修改结果表格不会更改数据库中的任何内容。要将已经更新、插入或删除的记录发布到数据库中，请按下记入更改按钮。任何通过服务器端逻辑电路实现的列值的改变（通过提前插入或更新触发器，或者为插入语句设定的默认值）都会立刻被反映到最后的网格中。传载过后（如果你禁用了自动提交 SQL 窗口选项）则如果事务已经开始，工具栏中的提交和回滚按钮将被启用。要了解更多关于事务的信息，请参见第 17 章。

### 编辑大数据列

上一章已经提到，你可以单击 Long、Long Raw、CLOB、BLOB 和 BFILE 列的单元格按钮来查看各种格式的值。当结果表格可更新时，你可以使用大数据编辑器来更改列的值。

## 7.7 查看和编辑 XMLTYPE 列

SYS.XMLTYPE 类型不受 Oracle Net 9.0 和更早版本支持，所以你将不能直接访问被存储在某一列中的 XML 数据。例如，如果 *xml\_text* 是 SYS.XMLTYPE 列，则下列查询将不会提供预期的结果：

```
select id, xml_text from xml_table
```

仅仅 *id* 列将在结果集中显示。要查看 XML 数据，可以使用 *getclobval()* 成员函数：

```
select id, t.xml_text.getclobval() from xml_table t
```

这样 CLOB 就能在文本编辑器中显示，将自动应用 XML 语法高亮显示功能。要编辑 XML 数据，可以使结果集可更新，这通常通过包括 rowid 来实现：

```
select id, t.xml_text.getclobval(), rowid from xml_table t
```

现在你可以编辑 CLOB 并将已修改的数据发布到数据库。注意，表和视图的查看数据与编辑数据函数将自动应用到这些规则中，因此查看或编辑 XMLTYPE 列的最简单方法是右键单击表或视图，从弹出式菜单选择相应项目。

## 7.8 比较记录

要比较结果集中的记录，可以按结果集工具栏上的**比较记录**按钮。默认情况下，在比较当前记录时，将会高亮显示结果集中的相同字段值：

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7876	ADAMS	CLERK	7788	23-05-1987	1100.00		20
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600.00	300.00	30
3	7698	BLAKE	MANAGER	7839	01-05-1981	2850.00		30
4	7782	CLARK	MANAGER	7839	09-06-1981	2450.00		10
5	7902	FORD	ANALYST	7566	03-12-1981	3000.00		20
6	7900	JAMES	CLERK	7698	03-12-1981	950.00		30
7	7566	JONES	MANAGER	7839	02-04-1981	2975.00		20
8	7839	KING	PRESIDENT		17-11-1981	5000.00		10
9	7654	MARTIN	SALESMAN	7698	28-09-1981	1250.00	1400.00	30
10	7934	MILLER	CLERK	7782	23-01-1982	1300.00		10

如果移至不同的行，高亮显示的字段将会相应更新。再次按**比较记录**按钮可清除高亮显示。

如果单击靠近**比较记录**按钮边上的箭头按钮，可以控制比较功能的以下方面：



- 高亮显示相同** – 高亮显示与当前记录相应字段具有相同值的所有字段。
- 高亮显示不同** – 高亮显示与当前记录相应字段具有不同值的所有字段。
- 比较单列** – 高亮显示当前列中的唯一字段。
- 比较整个记录** – 高亮显示记录的所有字段。
- 区分大小写** – 选中后，比较字段值时将会区分大小写。

## 7.9 直接查询导出

对于具有较大结果集的查询（上万条记录或更多），首先将数据查询到结果表格，随后把数据导出到文件中，这样做也许不太方便。这也许会花很长时间和占用很多内存资源。直接将结果集写入到导出文件中更有效。为此，你可以按下结果集工具栏中的**导出查询结果…**按钮，以取代主工具栏中的**执行**按钮。这将显示一个弹出式菜单，你可以在其中选择导出数据为 CSV 格式（逗号分隔值）、TSV 格式（制表符分隔值）、HTML 格式、XML 格式或 Excel 格式。选择了格式以后，可以指定导出文件，然后将执行查询。在这种情形下，结果集不会显示，只能被写入到导出文件中。

弹出菜单还可包含到剪贴板子菜单项，允许直接以给定格式导出到剪贴板。

## 7.10 注释指令

在每条 SQL 语句前，可以使用特殊指令来控制查询的各个方面。例如：

```
-- Tab=All managers
select * from emp
where job = 'MANAGER';

-- Tab=Department 10
select * from emp
where deptno = 10;
```

其中的 Tab 指令控制结果集标签页的描述部分。下面的段落描述了所有其他指令。其他可用的指令如下所示。

- **Excel=<Filename>**

预先定义导出 Excel 时的文件名。可以使用文件名的环境变量。例如：

```
-- Excel=C:\Excel\test_%date%.xls
```

- **CSV=<Filename>**

预先定义导出 CSV 时的文件名。可以使用文件名的环境变量。例如：

```
-- CSV=C:\Excel\test_%date%.csv
```

- **Records=[All | Page | <n>]**

控制最先获取记录的数量。指定 All 则获取所有记录，指定 Page 则获取一页记录（取决于窗口高度），或者指定确切的记录数。以下例子将会获取最先的 1000 条记录：

```
-- Records=1000
```

- **Totals=[<Mode>:<Column>[,<Column>][,<Mode>:<Column>[,<Column>]]]**

显示指定列的列总计。默认情况下，方式是 Sum，但也可以指定 Count、Avg、Min 和 Max。指定方式后，也可以指定该方式应用到一列或多列。下面的例子将显示 sal 和 comm 列的平均值以及 mgr 列的计数：

```
-- Totals=avg:sal,comm,count:mgr
```

- **AltRowColor=[<RRGGBB> | <Name>]**

控制替代的行颜色。可以指定颜色名称（例如灰色）或十六进制 RGB 值。下面的例子显示的替代颜色为浅紫色：

```
-- AltRowColor=FFEEFF
```

## 7.11 保存 SQL 脚本

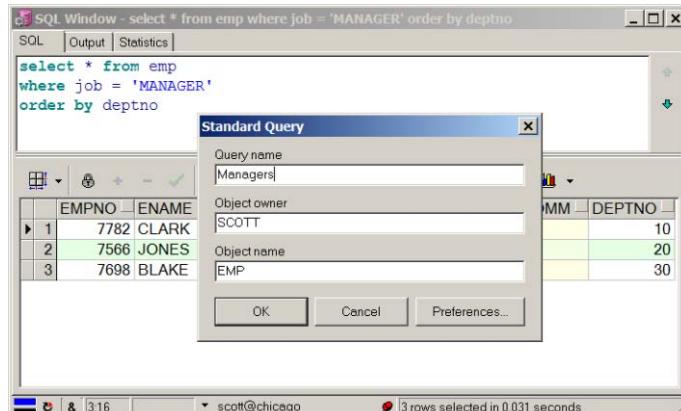
如果你已经创建了 SQL 语句，并希望在以后再次运行它，你可以把它保存为 SQL 脚本。为此，请按下工具栏中的 **保存** 按钮并在保存对话框中输入一个适当的文件名。SQL 脚本的默认扩展名为 **.sql**，但你可以使用第 20.27 节中描述的首选项来更改默认扩展名。注意，只有当前的 SQL 语句可以被保存到文件中，并不是全部的历史语句。要把历史包含到保存文件中，请右键单击 SQL 窗口，选择 **带有历史保存项目**。在其中，所有语句都将用斜杠分开。

你可以通过按下工具栏中的 **打开** 按钮并选择 SQL 脚本来打开以前保存的 SQL 脚本，这将创建一个新的 SQL 窗口。你也可以右键单击以前创建的 SQL 窗口并选择 **载入项目**，这将在现有的 SQL 窗口中打开一个 SQL 脚本。

## 7.12 创建标准查询

对于大多数表和视图，你将有各种需要频繁运行的标准查询，例如：“显示所有是经理的雇员”或者“按部门号码显示所有雇员的工资总额”。你可以将这样的查询保存为“标准查询”，以便以后能容易地从显示相应的表和视图的弹出式菜单中调用它。当你在对象浏览器中、或者在 PL/SQL 或 SQL 源中右键单击表或视图时，这个弹出式菜单就会显示出来。

在 SQL 窗口中创建了标准查询以后，你可以按下结果表格工具栏中的保存为标准查询按钮，将显示下列对话框：



你可以输入在弹出式菜单中显示的查询的名称，你还可以更改将显示标准查询的表或视图的所有者和名称。按下首选项按钮可以更改变标准查询的存储路径。

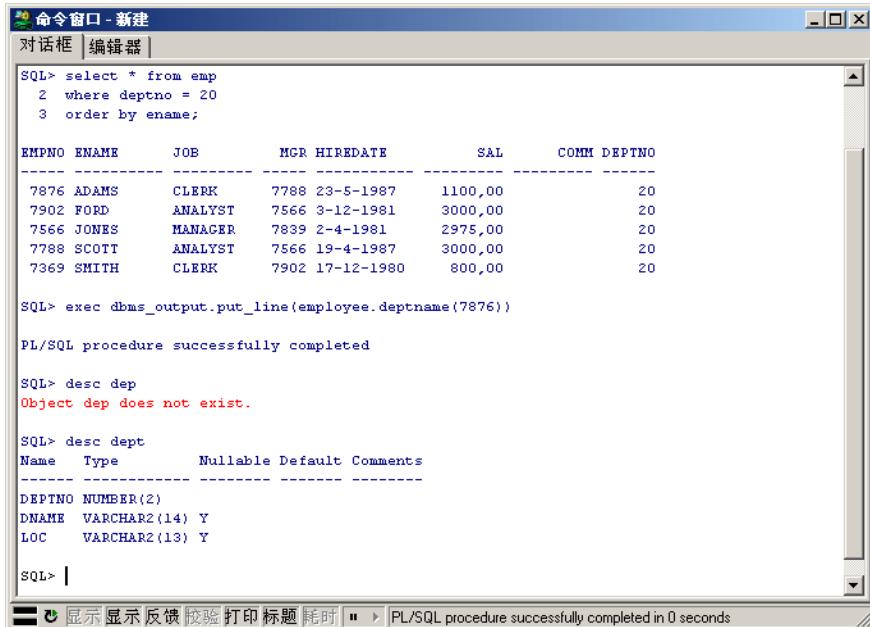
下列信息将与标准查询一起保存：

- SQL 文本
- SQL 窗口的大小
- 表格模式：多记录或单记录查看

注意，你可以使用置换变量（参见第 7.5 节）来使标准查询更加灵活。

## 8. 命令窗口

命令窗口允许你运行 SQL 脚本，这与 Oracle 的 SQL\*Plus 很相似。要创建命令窗口，请按下工具栏中的新建按钮或在文件菜单中选择新建项目。不用离开 PL/SQL Developer 的 IDE，就可以创建一个命令窗口，你可以像过去一样键入 SQL 语句或 SQL\*Plus 命令：



The screenshot shows the PL/SQL Developer interface with a floating command window titled "命令窗口 - 新建". The window contains the following SQL\*Plus-like session:

```

SQL> select * from emp
2 where deptno = 20
3 order by ename;

EMPNO ENAME      JOB         MGR HIREDATE        SAL       COMM DEPTNO
----- -----
7876  ADAMS      CLERK      7788 23-5-1987    1100,00   20
7902  FORD       ANALYST    7566 3-12-1981    3000,00   20
7566  JONES      MANAGER    7839 2-4-1981     2975,00   20
7788  SCOTT      ANALYST    7566 19-4-1987    3000,00   20
7369  SMITH      CLERK      7902 17-12-1980    800,00    20

SQL> exec dbms_output.put_line(employee.deptname(7876))

PL/SQL procedure successfully completed

SQL> desc dep
Object dep does not exist.

SQL> desc dept
Name      Type      Nullable Default Comments
----- -----
DEPTNO   NUMBER(2)
DNAME    VARCHAR2(14) Y
LOC      VARCHAR2(13) Y

SQL>

```

The status bar at the bottom of the window displays: 显示 显示 反馈 校验 打印 标题 耗时 | PL/SQL procedure successfully completed in 0 seconds

### 8.1 输入 SQL 语句和命令

就像在 SQL\*Plus 中一样，你可以键入多行 SQL 语句，用分号或斜杠来结束输入。你可以使用左右箭头来编辑命令行，用上下箭头重新调用前面输入的命令行。

通过输入编辑命令，你可以使用简单的文本编辑器来编辑整个输入缓冲区。编辑了缓冲区以后，可以在命令行上输入一个斜杠来运行它。编辑器有一个包括以前所有已运行命令的历史缓冲区，所以你可以快速运行已修改的命令。

命令窗口的状态行给出了显示命令、显示结果、反馈、校验、自动打印、标题和耗时选项的状态。另外，你还可以双击这些选项来开/关它们。

## 8.2 开发命令文件

要开发带有多个 SQL 语句和命令的命令文件，经常需要编辑这个文件，运行它，再编辑它，再运行它等。要把这些事情变成一个轻松的过程，命令窗口有一个带有 SQL、PL/SQL 和 SQL\*Plus 语法高亮显示功能的内置编辑器：

```
-- This script creates the SQL*Plus demonstration tables in the
-- current schema. It should be STARTed by each user wishing to
-- access the tables. To remove the tables use the demodrop.sql
-- script.

SET TERMOUT ON
PROMPT Building demonstration tables. Please wait.
SET TERMOUT OFF

-- Drop old tables
DROP TABLE EMP;
DROP TABLE DEPT;
DROP TABLE BONUS;
DROP TABLE SALGRADE;

-- Create new tables
CREATE TABLE EMP
(
    EMPNO NUMBER(4) NOT NULL,
    ENAME VARCHAR2(10),
    JOB VARCHAR2(9),
    MGR NUMBER(4),
```

### 打开、执行和保存命令文件

要创建一个命令文件，请转到编辑器页并输入命令。要在编辑器中执行命令，只需按下工具栏中的执行按钮或按下 *F8* 即可，命令窗口将切换回对话框页并将执行所有命令。

要编辑一个现有的命令文件，请按下工具栏中的打开按钮，选择命令文件项目。这将把命令文件载入到一个新的命令窗口的编辑器中。你也可以在编辑器中右键单击鼠标，从弹出式菜单选择载入项目。要保存已修改的命令文件，请按下工具栏中的保存按钮。

### 单步执行命令文件

如果你在执行前或在执行期间按下了状态行上的暂停执行按钮，则可以以可控方式单步运行脚本。下一个命令将在编辑器中高亮显示，当你按下状态行上的执行下一个命令按钮时，它就会被执行。如果你再次按下了暂停执行按钮，执行将在下一步以后正常地继续进行。你还可以使用 SET EXECPAUSE 命令在脚本中强制暂停。

## 8.3 支持命令

除了所有的 SQL 语句之外，在这个版本中，命令窗口还支持下列标准的 SQL\*Plus 命令：

命令	含义
/	运行 SQL 缓冲区
? [关键字]	提供关于关键字的 SQL 帮助
@[@] [文件名] [参数列表]	运行指定的命令文件，传递指定的参数
ACC[EPT] 变量 [DEF[AULT] 值] [PROMPT]	允许用户输入置换变量的值

命令	含义
文本   NOPR[OMPT]]	
CL[EAR] [SCR[EEEN]]	清除屏幕
CL[EAR] SQL	清除 SQL 缓冲区
COL[UMN] [列] [格式] [NEW_VALUE 变量]	定义一列的格式，显示一列的格式，或者显示所有列的格式
CONNECT[用户名/口令@数据库] [EDITION=版本] [WORKSPACE=工作区]	链接到带有指定用户的数据库，同时可选择设置版本和/或工作区
DEF[INE] [变量] [= 文本]	定义置换变量，显示一个变量，或者显示所有置换变量
DESC[RIBE] 对象	描述指定对象
DISC[CONNECT]	从数据库断开
EDIT	显示文本编辑器以编辑 SQL 缓冲区
EXEC[UTE] 过程	运行指定的过程
EXIT [APPLICATION]	退出正在运行的脚本或关闭命令窗口，添加 APPLICATION 参数也将关闭 PL/SQL Developer
GET [文件名]	将命令文件加载到编辑器中
HOST [命令]	执行主机命令
HELP [关键字]	提供关于关键字的 SQL 帮助
PAUSE [信息]	显示消息并暂停，直到用户按下确定或取消按钮
PRI[NT] [变量]	显示绑定变量或所有绑定变量的值
PROMPT [文本]	显示指定文本
QUIT [APPLICATION]	退出正在运行的脚本或关闭命令窗口，添加 APPLICATION 参数也将关闭 PL/SQL Developer
R[UN]	运行 SQL 缓冲区
REM[ARK] [文本]	一个注释行
SET AUTOP[RINT] [ON   OFF]	确定绑定变量在运行了 SQL 语句或 PL/SQL 块以后是否自动显示
SET COLSEP [分隔符   OFF]	确定列分隔符(默认 = “ ”)
SET CON[CAT] [字符   ON   OFF]	确定终止置换变量参照的字符(默认=.)
SET DEF[INE] [字符   ON   OFF]	确定开始置换变量参照的字符(默认 = &)
SET ECHO [ON   OFF]	确定脚本中已执行的命令是否显示
SET ESC[APE] [字符   ON   OFF]	确定中断开始置换变量参照的字符的字符(默认 = \)
SET FEED[BACK] [ON   OFF]	确定 SQL 语句受影响行数是否显示
SET HEA[DING] [ON   OFF]	确定在结果集列上面的标题是否显示
SET LONG [宽度]	确定一个 LONG 列的最大显示宽度
SET NUM[WIDTH] [宽度]	确定不带精度的数字列的最大显示宽度
SET PAGES[IZE] [大小]	确定在标题被重复以前的结果集的显示行数
SET PROMPT [提示]	替换标准的 SQL> 提示，无需使用文字文本，你也能使用变量 [user]、[db] 或 [connection]。此外，你还可以包括一个绑定变量 (set prompt:bind_var_name)
SET SCAN [ON   OFF]	确定置换变量是否应该被扫描
SET SERVEROUT[PUT] [ON   OFF] [SIZE]	确定调用 dbms_output.put_line 的输出是否显

命令	含义
n]	示, 以及输出缓冲区的大小是多少. 无限制的输出大小的 n 值可以被设定为无限制。
SET SPOOLDIR[ECTORY [目录]]	确定如果 SPOOL 命令没有指定一个绝对路径, 假脱机文件应该被储存在哪个目录中
SET TERM[OUT] [ON   OFF]	确定已运行的 SQL 语句的输出是否显示
SET TIMI[NG] [ON   OFF]	确定关于已运行的 SQL 语句的计时信息是否显示
SET VER[IFY] [ON   OFF]	确定当被用于 SQL 语句或 PL/SQL 块时, 置换变量是否显示
SHO[W] ERR[ORS] [类型名]	显示以前的编译或指定对象的错误
SHO[W] REL[EASE]	显示当前链接的 Oracle 版本信息
SHO[W] SQLCODE	显示已运行的 SQL 语句的结果代码
SHO[W] USER	显示当前链接的用户名
SPO[OL] [文件名 [APPEND]   OFF   OUT]	开始或停止假脱机和可选的打印(输出)。如果文件名是压缩文件名, 假脱机文件就会被压缩。
STA[RT] [文件名] [参数列表]	运行指定的命令文件, 传递指定的参数
STORE SET [文件名]	在[文件名]中存储所有选项的值, 你可以在以后运行这个文件以恢复这些选项
UNDEF[INE] 变量	不定义已设定的置换变量
VAR[IABLE] [变量] [数据类型]	定义一个绑定变量, 显示一个绑定变量或显示所有的绑定变量
WHENEVER [OSERROR   SQLERROR] [动作]	为 OS 或 SQL 出现错误时指定一个动作。这个动作既可以是 EXIT 又可以是 CONTINUE 后面也可以添加上 COMMIT 或 ROLLBACK

上面所有这些命令函数都与 SQL\*Plus 相同。下面的命令是 PL/SQL Developer 特有的:

命令	含义
BEAUTIFY[IFY] 文件   对象	使用当前规则或 SET BEAUTIFERRULES 命令指定的规则来美化指定文件或数据库对象
BROWSE 对象	在对象浏览器中选择对象
EDIT 对象	打开带有对象定义的可编辑窗口
EDITD[ATA] 表   视图	为带有可编辑结果集的表或视图打开一个 SQL 窗口
EXPORT[DATA] 表	为指定表打开导出工具
INFO	显示关于链接的信息
PROP[ERTIES] 对象	为指定对象显示一个属性窗口
QUERY[DATA] 表   视图	为带有只读结果集的表或视图打开一个 SQL 窗口
REC[OMPILER] 对象	重新编译对象
RUNTEST[SET] TestSetFile [LogFile]	在测试管理器中运行指定测试集文件, 并将结果记录到日志文件中
SET BEAUTIFY[IFIERRULES] [文件]	临时使用来自指定文件的美化器规则, 这可以连同 BEAUTIFY 命令一起使用
SET COL[WIDTH] [宽度]	确定结果集中的最大列宽。如果 Width = 0 , 则不限制宽度。默认为 80

命令	含义
SET EXEC[PAUSE] [ON   OFF]	在下一个命令处暂停执行 (ON) , 或者在下一个命令处继续执行 (OFF)
SHOW CONN [ECTION]	显示所有的关于 PL/SQL Developer 链接信息。
SQLPLUS	调用带有当前文件的 SQL*Plus
TEST 程序单元	对于指定的程序单元, 打开一个带有标准测试脚本的测试窗口
VIEW 对象	打开一个带有对象定义的只读窗口
ZIP Archive [File   Directory]	将文件和根目录添加进压缩文档中。如果文件和根目录参数被遗漏的话, 压缩文档里的内容就会被显示出来。

## 9. 创建和修改非 PL/SQL 对象

在 PL/SQL 开发期间，你会发现你经常需要创建表，修改约束或索引，复位一个序列等。

PL/SQL Developer 有几个功能允许你创建与修改表（以及相关元素）、序列、同义词、库、目录、用户和角色。这一章并不解释这些对象的功能，只解释你怎样才能创建、修改和查看这些属性。要了解关于每个对象的更多信息，请参见“SQL 参考指南”等 Oracle 文档。

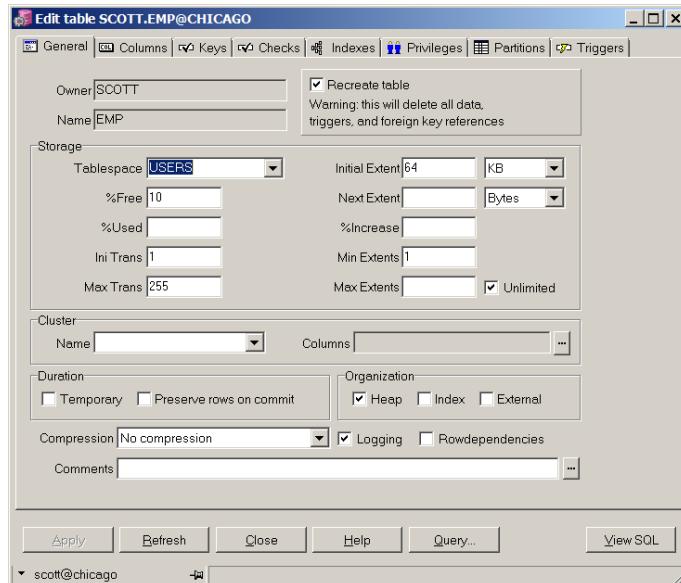
要创建一个对象，既可以按下工具栏中的新建按钮选择相应的对象类型，也可以选择文件菜单中的新建按钮。你还可以右键单击对象浏览器中对象类型的根文件夹，从弹出式菜单选择新建项目。

要修改一个对象，可以在浏览器中选择它，右键单击它，从弹出式菜单选择编辑按钮。你不能通过这种方式更改一个对象的名称，必须明确地使用来自于对象浏览器的重新命名功能来更改名称（如果对象浏览器支持的话）。如果只想查看对象，可以从弹出式菜单选择查看项目。

另外，你还可以通过选择对象浏览器弹出式菜单中的复制项目来复制一个对象。这个对象的编辑器窗口将显示出来，它包含所填充的所有属性，除了所有者（如果适用）和名称。这样，你能快速地创建一个相似的对象，以探索一个替代方法，运行一些危险的或破坏性的测试，或者做其他任何事情。

### 9.1 表定义编辑器

表定义编辑器包含 8 个标签页，分别针对表的不同方面：



在编辑器底部，可以看到下列 6 个按钮：

- **应用** - 将你在编辑器中所做的所有修改应用到数据库中。
- **刷新** - 再次获取来自于数据库的定义，放弃你可能已经执行的任何更改。

- **关闭** - 关闭编辑器窗口。
- **帮助** - 显示在线帮助。
- **查询** - 调用一个带有查询的 SQL 窗口，这个查询允许你查看和编辑表数据。
- **查看 SQL** - 显示带有 SQL 语句的文本编辑器，这些 SQL 语句是表定义编辑器中产生了更改的结果。如果你没有执行任何更改，它将显示完整的 SQL 创建语句。

接下来的章节将对表定义编辑器的各个页进行描述。

## 一般页

一般页在前面的章节中已有显示，它包括表的所有者和名称、存储信息、簇信息和注释。对于一个新表，除了名称之外，你可以将所有的属性保留为空。所有其他属性将设为默认值：

- 所有者 - 当前登录的用户。
- 表空间 - 当前用户默认的表空间
- %空闲 - 10
- %已用 - 40
- 初始事务数 - 1
- 最大事务数 - 255

段属性的默认值（初始大小、下一个大小、%增加、最小数量和最大数量）依赖于相应的表空间默认值。

簇信息、存储信息和持续时间属性互相排斥，这是因为簇会隐式地定义表存储特性，临时表不能成为簇，并且具有隐含的存储特性。

如果你选择外部作为表的组织方式，存储属性将被禁用，并显示新的外部标签页，在其中可定义外部目录和文件信息。

如果你输入或选择一个簇名，存储属性和持续时间属性将变为只读，并且可以输入簇列。清除簇名称具有相反的效果。

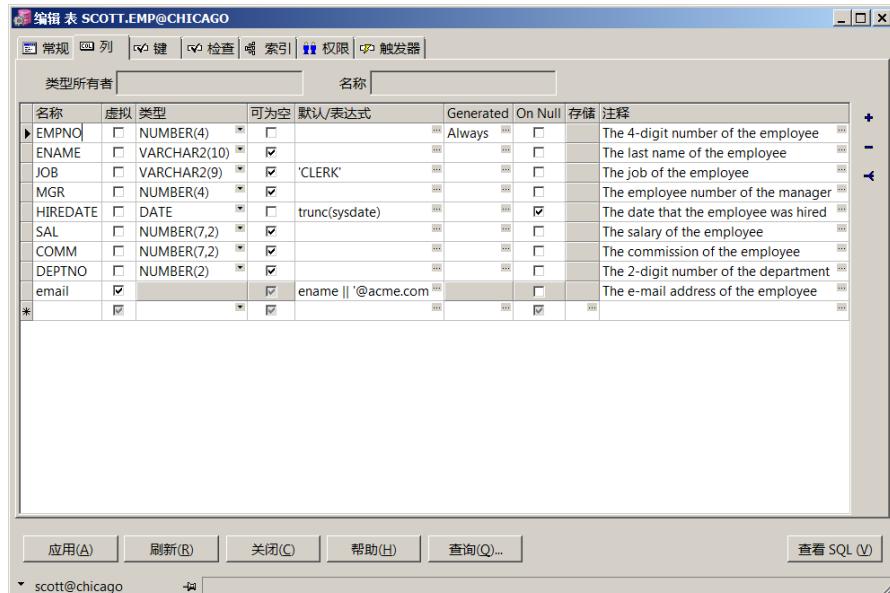
如果你将表定义为临时，簇和存储信息将变为只读，你可以定义行在提交后是否需要被保护。

如果你定义一个索引结构表，必须定义一个主键。对于一个堆积构成的表（默认），则不需要这么做。

如果你正在修改一个现有的表，并不是所有的属性都能更改。对于一个数据库中已经存在的表，表空间、初始大小、最小大小、簇信息、持续时间属性和构成属性都不能更改。如果你想要更改这些属性中的任何一个，你必须选中窗口顶部的重建表选项。结果，首先会删除表，再用新的定义重建。所有数据、触发器、外键参照都会丢失，所以在使用这个选项时，你要特别小心！

## 列页

在列页中，你可以查看、添加、删除、移动和修改表列。如果你创建了一个对象表，你还可以在这个页中选择对象类型：



要添加一列，你可以按下 **插入列** 按钮，或者可以在最后一行键入新列的信息（这将创建一个新的是空的末尾行）。对于大多数通用数据类型，**类型** 列有一个建议列表，但在这中你还可以使用其他数据类型。**默认** 列需要你按适当格式输入一个值，例如字符串值必须用引号引上。

要删除一列，请按下 **删除列** 按钮。这既会删除当前处在焦点中的一列，又会删除所有的已选列。你可以通过单击行标题来选择一个或更多列，拖动鼠标将选择一个列范围。按下 **Ctrl** 键并同时单击行标题将允许你选择多个单独的列。

要移动一个或更多列，可以按前一段的描述选择它们。现在可以单击已选标题中的一个，移动它到你希望的地方。

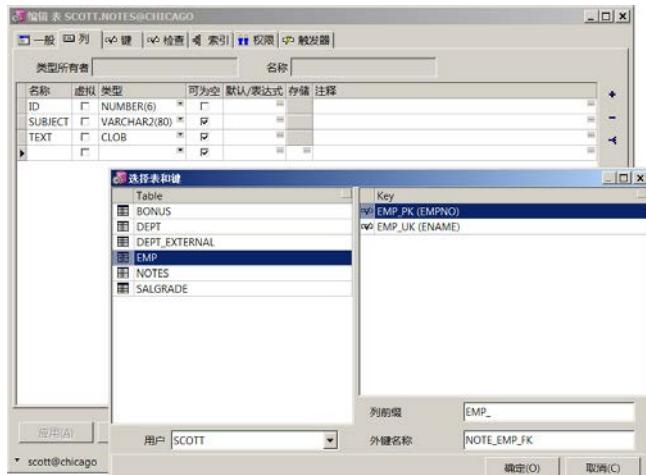
如果是虚拟列的话（如上述例子中的电子邮件列），你可以进入默认/EXPR 字段里的表达式页面。如果是真实的列，这些字段都是默认的。

对于现有的表，你不能更改现有列的名称，除非你用的是 Oracle9.2 或更新版本。你不能删除这样的列，除非你用的是 Oracle8i 或者更新版本。要克服这些限制，你必须选中一般页上的**重建表**选项，这一选项的作用在前面的章节中有描述。

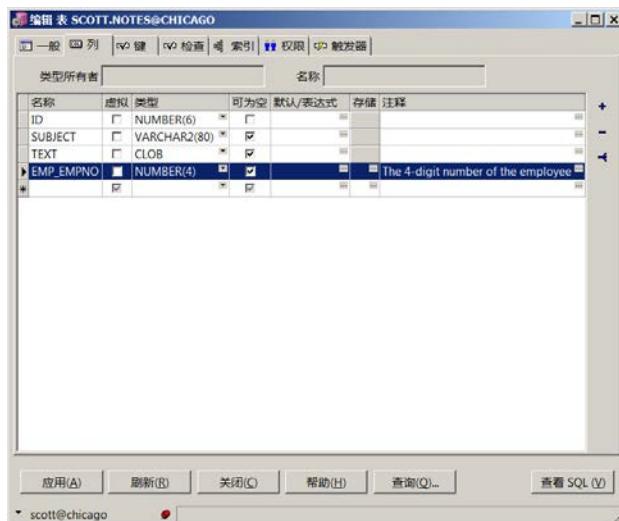
注意，列名的更改将自动传播到约束列、索引列和簇列。

如果你在这个标签页的顶部选择了类型所有者和类型名，你可以创建一个基于这个对象类型的对象表。现在已经定义了列名和类型，并且它们被限制到对象类型的顶级属性。你还可以定义列是否可为 null、默认值、它的存储以及注释。如果你以后要更改对象类型，必须选择一般页上的**重建表**选项。

创建表时，你通常会添加一个外键约束及其对应的列。按添加外键列按钮，即可迅速完成此任务。假设你正在创建表 *NOTES*，该表具有 *EMP* 表的外键约束。在列页面中，可按下添加外键列按钮，选择 *EMP* 表，以及外键约束引用的 *EMP\_PK* 主键约束：

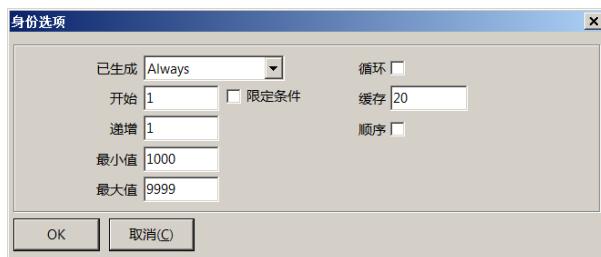


选择表和键后，为外键列输入可选的列前缀，输入新外键约束的名称：



外键列与所引用表的主键列具有相同的名称（包括前缀）、数据类型和备注。

要创建标识列（Oracle12c 及后续版本），可以按生成属性的单元格按钮。这会显示一个表单，在此可以输入生成选项（从不、始终或默认），以及 Oracle 服务器使用的底层序列属性：



## 键页

在键页上，你可以查看、添加、删除和修改表的主键、唯一键和外键约束：

名称	类型	列	启用	参照表	参照列	级联删除	可延迟	已延迟	Validated	上次修改
EMP_PK	Primary	EMPNO	<input checked="" type="checkbox"/>			No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	05-08-2013 12:50:06
EMP_DEPT_FK	Foreign	DEPTNO	<input checked="" type="checkbox"/>	DEPT	DEPTNO	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	05-08-2013 12:50:06
EMP_MGR_FK	Foreign	MGR	<input checked="" type="checkbox"/>	EMP	EMPNO	No action	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	04-12-2014 12:21:51

这个表格可以按与前面章节中的列表格相同的方法进行操作。你可以添加、删除和修改键的约束。移动约束不会有任何作用，因为它们并未实际排序。

注意，被强制作为主键或唯一键约束的基础索引的存储信息不会在这一页中定义。如果启用了约束，在索引页上会自动出现一个具有相同名称的相应索引。对于这个索引，你可以定义存储定义。禁用或删除主键或外键约束将隐式地删除基础索引。

类型列有一个允许你选择主、唯一或外键类型的列表。你可以通过键入首字符（P、U 或 F）快速选择恰当的类型。注意，一个表只能有一个主键约束，这个列（主键或唯一键）的默认值就

依赖此约束。更改类型将影响到参照表、参照列和层叠列，这是因为它们仅仅对外键约束有效。将类型更改外键也将隐含地删除索引。

要定义应用约束的列，可以在适当的单元格（以逗号分隔）中简单键入列名，或者按下单元格按钮。这将调出列选择屏幕，在其中你能容易地添加、删除或移动（重排）键列：

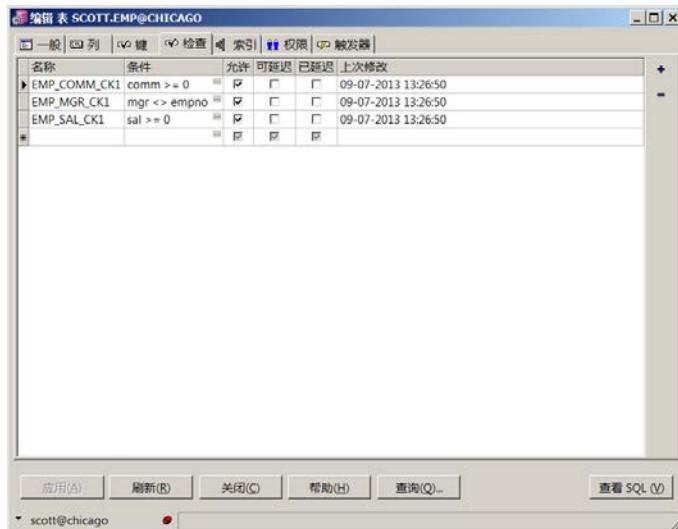


对于外键约束，你可以键入或者选择参照表。这样做将自动选择基于这个表的主键和唯一键约束的默认参照列。这个列的选择列表显示了所有列的设置，它们为参照表整理出主或唯一键约束，因此在这中它们仅仅为候选者。在级联删除列表栏中，可以定义在父记录被删除时，应对子表中的记录执行何种操作。Set null 行为只适用于 Oracle8i 和更新版本。

如果你用的是 Oracle 8i，你还可以定义约束是否可延迟，在最初是否已延迟。这两个选项相互关联：你不能拥有一个最初已延迟而又不可延迟的约束。

## 检查页

在检查页中你可以查看、添加、删除和修改表的检查约束：



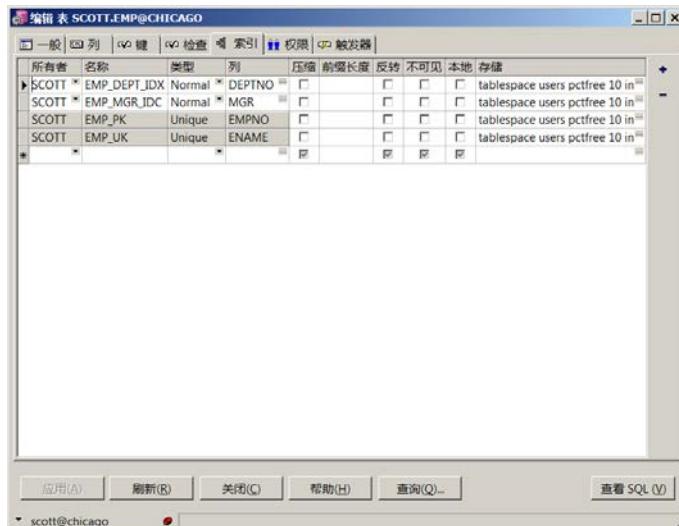
这个表格能用与列表格一样的方法操作，你可以添加、删除和修改检查约束。

条件列应该包括对表中的所有行都必须是真（或空）的布尔表达式。如果一个或多个行违反条件，你将不能应用检查约束，除非你也未选中允许选项。在纠正了违反条件行以后，才可以启用检查约束。

如果你用的是 Oracle8 或更新版本，你还可以定义约束是否可延迟、最初是否被延迟。这两个选项相互关联：你不能拥有一个最初已延迟而又不可延迟的约束。

## 索引页

在索引页中你可以查看、添加、删除和修改表的索引：



这个表格能用与列表格一样的方法操作，你可以添加、删除和修改索引。

索引的所有者可以设为空，默认的所有者是当前登录的用户。如果你要为索引使用不同的用户，可以从这个列表中选择它。

索引的类型可以是 Normal、Unique 或 Bitmap。最后一个索引类型只能用于 Oracle7.3 服务器和更新版本。注意，如果数据库中每一行的索引列都是真正唯一的，你只能创建一个唯一索引。否则，你将不能将这个索引应用到数据库中。

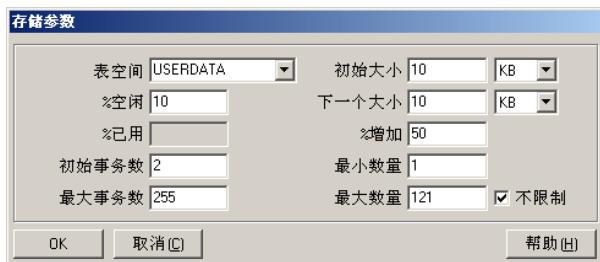
要定义你要创建索引的表的列，可以在适当的单元格（以逗号分隔）中简单键入列名或按下单元格按钮。这将调出列选择屏幕，在其中你能容易地添加、删除或安排索引列。

对于基于索引的函数，你可以以逗号分隔的形式指定函数。对于递减索引，你可以在列名之后添加上 DESC。

你还可以为索引启用键压缩，包括前缀列的数字压缩，你还可以将索引定义为反色索引。

如果索引是主键或唯一键的索引，则所有者、名称、类型和列就不能被修改。这些特性全部得自键约束。更改键约束的名称将自动地更改相应的索引名，更改约束列将更改用于索引的相应的列。索引的所有者总是与表的所有者相同，索引的类型总是唯一的。

存储参数能通过按单元格按钮来定义或查看，这将调用存储编辑器：



对于新索引，你不必输入任何信息，因为每个属性都有一个默认值：

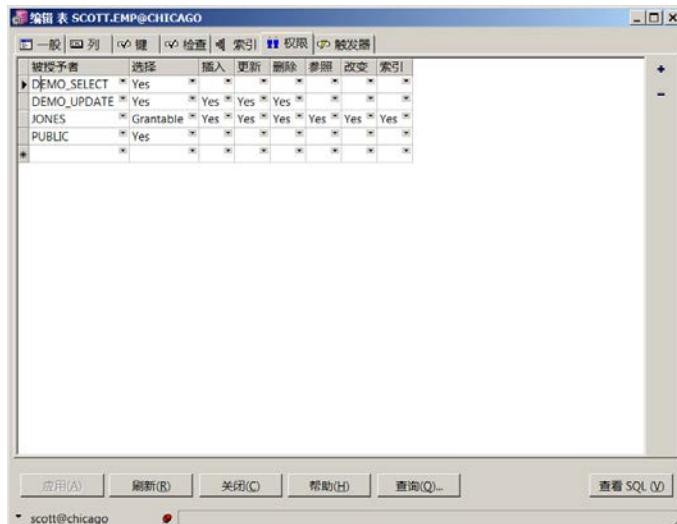
- 表空间 - 针对当前用户默认的表空间。
- %空闲 - 10
- 初始事务数 - 2
- 最大事务数 - 255

段属性的默认值（初始大小、下一个大小、% 增加、最小数量和最大数量）依赖于相应的表空间默认值。% 已用特性不能应用到索引中。

注意，如果你修改了现有索引的 % 空闲、初始大小和最小数量特性，这将暗示索引（和约束）将被删除掉和被重新创建，这种更改也会被应用。对于大的索引，这可能是非常耗费时间的。

## 权限页

在权限页中，你可以在表中授予/撤回给/来自用户和角色的权限：



这个表格能用与列表格一样的方法操作，你可以添加、删除和修改被授予者。删除一个被授予者与撤回这个人所有的权限等效。

**被授予者**是你要授予权限的用户或角色。注意，给角色授予的权限常常不能满足于存储程序单元开发的目的。如果用户在表中通过角色被授予了 select 权限，参照这个在已存储的程序单元中的表，编译将失败，这是因为表是未知的。如果对象是被用于他或她拥有的程序单元中的，用户必须亲自被授予 select 权限。

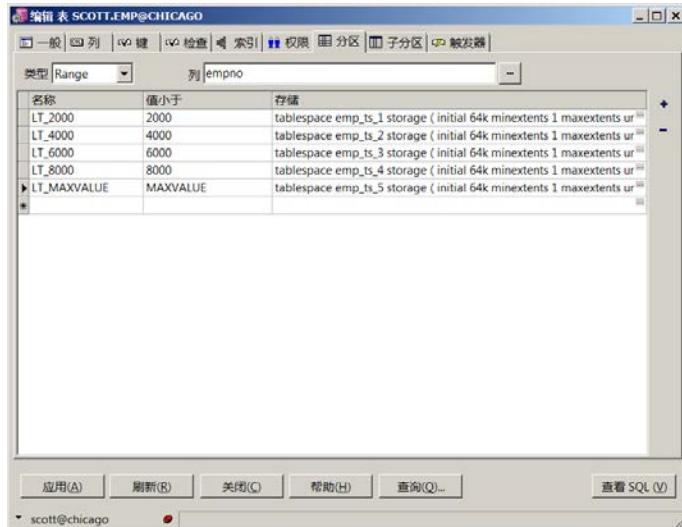
单独的权限列能有 3 个值：

- (空白) - 被授予者没有权限
- Yes - 被授予者有权限
- Grantable - 被授予者有权限，并且能授予它到其他用户和角色

**选择、插入、更新、删除权限**允许被授予者在表记录中运行这些动作，**参照权限**允许被授予者创建外键到这个表，**更改权限**允许被授予者用更改表命令更改表定义，**索引权限**允许被授予者在表中创建索引。

## 分区页

分区页仅仅在你（重新）创建表时或者在你编辑现有的已分区的表时是可见的：

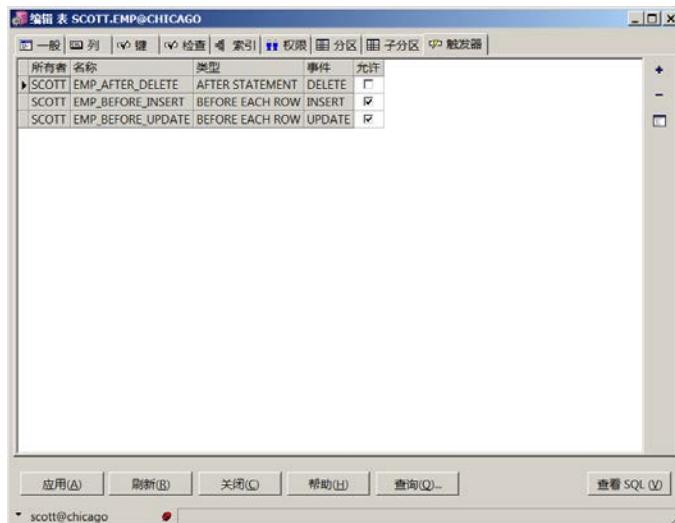


在页的顶部，你可以选择 *Range*、*Hash* 或 *List* 来分区，使用分区列也可以。在表格中，你可以定义名称、列的值（如果可应用的话），可以存储每个分区。如果你定义了不只一个分区列，你必须用逗号来分隔这些值。

对于 *Range* 分区，你还可以在附加的子分区标签页上定义子分区。

## 触发器页面

此页面显示为表定义的所有触发器：



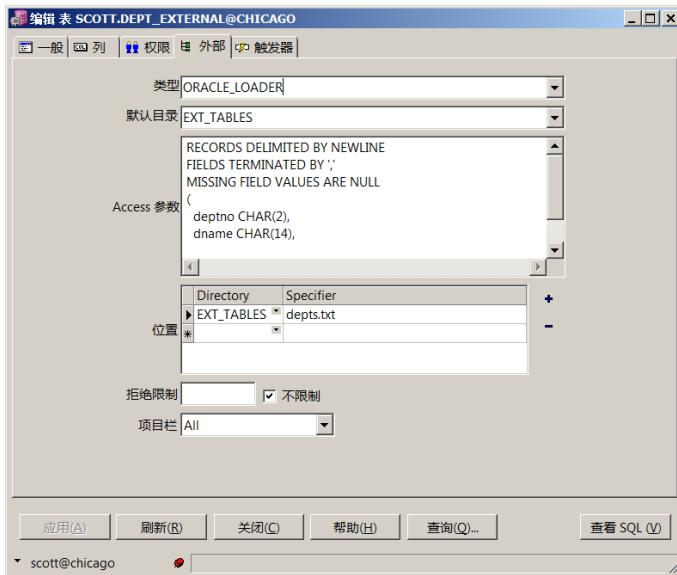
你可以按新建触发器按钮添加触发器。这将调用对应的模板来创建触发器，随后在程序窗口中打开触发器。编译完成后，触发器将显示在触发器页面中。

要删除触发器，请按删除触发器按钮。要启用或禁用触发器，单击已启用复选框。按查看SQL按钮可查看drop/enable/disable trigger语句以及对表做出的所有其他更改。

要查看或编辑触发器源代码，请在列表中双击触发器，或者按编辑触发器按钮。这将打开选定触发器的程序窗口。

## 外部页

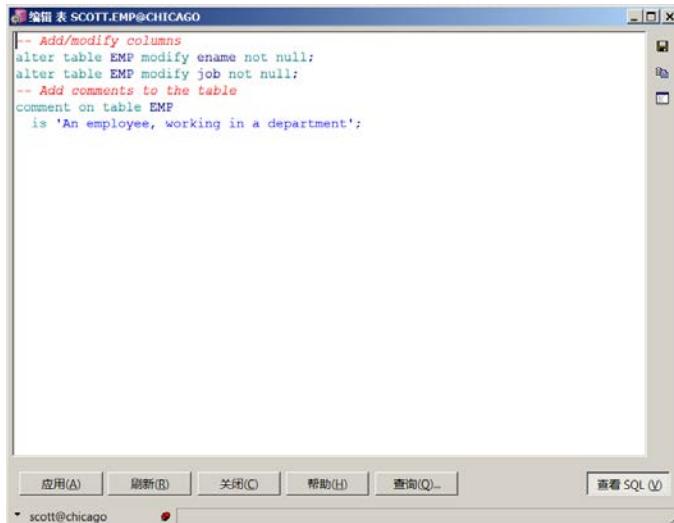
如果在一般页上将表的结构定义为外部，此标签页将可见：



在这里，你可以设置访问驱动器类型和定义外部表的外部数据属性。

## 查看与修改 SQL

在创建了一个表以后，或者修改了一个现有的表以后，你可以通过按查看 SQL 按钮来查看作为结果的 SQL。在添加了注释到表中并将 *ename* 和 *job* 列从可选修改为强制后，生成的 SQL 就像这样：



```
-- Add/modify columns
alter table EMP modify ename not null;
alter table EMP modify job not null;
-- Add comments to the table
comment on table EMP
    is 'An employee, working in a department';
```

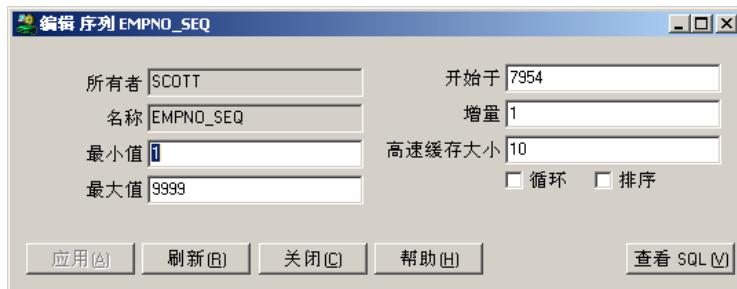
你可以使用窗口右上角的相应按钮将 SQL 保存为一个文件，将其复制到剪贴板或在命令行窗口中打开它。工具栏中的打印按钮将打印 SQL。你还可以产生一个更改到 SQL 并且应用已更改的 SQL。在应用了已更改的 SQL 以后，窗口将转到表格模式并且刷新表定义。已产生到 SQL 的更改将立即在表格中被反映出来。

如果你再次按查看 SQL 按钮，窗口将转到表格模式，抛弃了任何你可能已经生成到 SQL 的未应用的更改。

如果你已经选择了浏览器中的查看选项来查看表定义，查看 SQL 按钮就将显示表创建的 SQL，不带有前面所述的带有删掉命令的它。这样，你可以在文件中保存表定义，或者复制表定义到另一个数据库或用户。

## 9.2 序列定义编辑器

序列编辑器允许你容易地创建或修改一个序列：



大多数属性都有一个默认值，当创建一个序列时，你只需应用一个名称：

- 所有者 - 当前登录的用户
- 最小值 - 1
- 最大值 - 不限制
- 初始值 - 1
- 增量 - 1
- 高速缓冲区大小 - 20
- 循环 - 关
- 排序 - 关

## 9.3 同义词定义编辑器

同义词编辑器允许你容易地创建或修改同义词：



所有者的默认值将是当前登录的用户。设公共选项为已检验将创建公共同义词，并且将产生只读的所有者输入框。对象所有者和对象名称输入框有一个建议列表，以便容易地为同义词选择一个事务对象。如果对象所有者输入框是空的，对象名称列表将显示出所有对象。

## 9.4 库定义编辑器

库定义编辑器允许你容易地创建或修改库：



所有者的默认值将是当前登录的用户，你可以使用建议列表来创建带有不同所有者的库。文件说明是你要使用的 3GL 库完整的文件说明。如果你正在数据库服务器上搞开发，那么你可以使用选择 DLL 文件按钮来选择动态链接库。另外，你必须键入 DLL (Windows) 的名称或共享对象 (UNIX)。文件说明的有效性并不被检验。

## 9.5 目录定义编辑器

目录编辑器允许你容易地创建或修改一个目录，这个目录随后可用于 BFILE：



如果你正在数据库服务器上搞开发，那么你可以使用选择目录按钮来选择目录的路径。目录路径的有效性并不被检验。

## 9.6 作业定义编辑器

作业定义编辑器允许你创建或修改一个作业:



创建一个新的作业与调用 `dbms_job.submit` 相同，修改一个现有的作业与调用 `dbms_job.change` 相同。要了解更多关于作业的信息，请参见 *Oracle 供应包参考手册*中的 **DBMS\_JOB** 章。

*What* 输入区域旁边的打开 *PL/SQL* 编辑器按钮可以调用更加方便的带有语法高亮的 *PL/SQL* 编辑器、代码助手，等。

## 9.7 队列定义编辑器

队列定义编辑器允许你创建或修改一个高级队列:



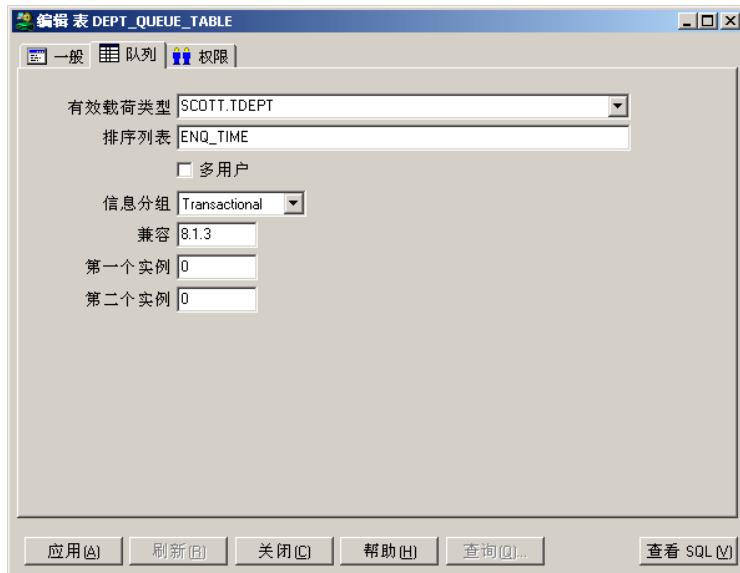
对于队列表的名称，你可以选择一个现有的队列表，或者你可以按**创建**按钮来创建一个新的队列表（请参见第 9.8 章）。

要了解关于 Oracle 高级队列的更多信息，请参见下列 Oracle 文档:

- 应用程序开发指南 - 高级队列
- 补充的 *PL/SQL* 包和类型参考

## 9.8 队列表定义编辑器

队列表定义编辑器允许你创建或修改一个高级队列表：



在一般标签页上，你可以定义表的名称、存储参数和注释。在权限标签页上，你可以在表上授权给用户和角色。请参见第 9.1 章。

在队列页上，你可以提供队列表说明信息。对于有效载荷，你可以选择一个现有的对象类型或 RAW 来创建一个裸队列。

要了解关于 Oracle 高级队列的更多信息，请参见下列 Oracle 文档：

- [应用程序开发指南 - 高级队列](#)
- [补充的 PL/SQL 包和类型参考](#)

## 9.9 用户定义编辑器

用户定义编辑器允许你创建或修改一个用户、它的权限和表空间限额：



注意，你需要指定系统权限（例如由 DBA 角色提供的）来查看和修改用户信息。

### 一般页

在一般页中，你基本上只需输入名称和口令，所有其他属性都有一个默认值：

- 默认表空间 - SYSTEM
- 临时表空间 - SYSTEM
- 概要文件 - DEFAULT
- 口令期限 - 关
- 账户被锁住 - 关

如果你要修改一个现有的用户，当前的口令将不被显示。输入一个新的口令可以更改它，或保留为空白则不更改口令。如果用户要在外表上被鉴别，你就不能输入一个口令。

## 对象权限页



这一页允许你查看和修改被授予到用户的对象权限，这不包括间接地通过角色授予的权限。

在第一列中，你可以看到被授权到用户的对象。所有其他列都描绘出了一个指定的权限，每个权限可以有下列值：

- (空白) - 用户没有权限
- Yes - 用户有权限
- Grantable - 用户有权限，并且能授予权限到其他用户或角色。

## 角色权限页



这一页显示了被授权到用户的角色。每个角色权限都是可授予的，所以用户可以授予权限到其他用户和角色。如果角色权限是默认的，则用户登录时角色将被激活；如果它不是默认的，则对于在用户已经登录后通过执行 `set role` 命令的会话，角色权限必须明确地被激活。

## 系统权限页



这一页显示了被授予到用户的系统权限。每个系统权限都可以是可授予的，所以用户能授予它到其他用户和角色。

## 限额页



这一页显示了用户的表空间限额。你可以输入用户能在表空间中分配的最大数量的字节、千字节或兆字节，或者你可以指定限额为不限制。

## 9.10 角色定义编辑器

角色定义编辑器允许你创建或修改一个角色和它的权限：



在默认的状态下，角色不需要被鉴别。你同样也可以对角色定义口令鉴别或外部鉴别。口令鉴别需要你输入一个口令，如果角色没有默认地被授权到用户，这就是有用的。在用户登录以后，这样的角色需要被明确地设置，并且需要口令。

其他三个标签页（对象权限、角色权限和系统权限）以与用户定义功能相应页一样的方法工作。

## 9.11 概要文件定义编辑器

概要文件定义编辑器允许你编辑资源限制、口令限制和用户的概要配置:



每个限制能被设置为默认, 这样值将从标准概要配置的 DEFAULT 到 *Unlimited* 或一个具体值被继承。

在用户标签页上, 你可以定义到应用这个概要文件的哪一个用户。注意, 如果你从概要文件中删除一个用户, 它将隐式地被更改到 DEFAULT 概要文件中。

## 9.12 数据库链接定义编辑器

数据库链接定义编辑器允许你创建和编辑数据库链接:



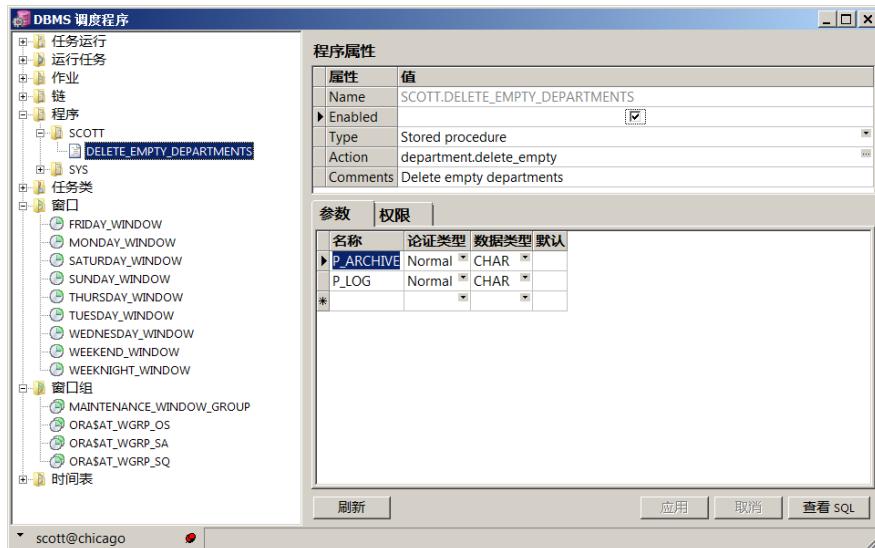
注意, 当你编辑一个现有的数据库链接时, 口令和鉴别属性将不会被获取, 你必须在编辑数据库链接时每次都输入这个信息。

## 10. DBMS 调度程序

可以使用 DBMS 调度程序工具，访问 Oracle 10g 及更新版本中的 Oracle 数据库调度程序（DBMS\_SCHEDULER）。为此，可以使用工具菜单中的 DBMS 调度程序菜单项，或在对象浏览器中右键单击调度程序对象，并从弹出式菜单中选择编辑或查看。对应的对象浏览器文件夹是窗口、窗口组、计划、程序、作业和作业类。

**注意：**关于 DBMS\_SCHEDULER 功能的详细信息，请参见 *Oracle 数据库概念指南*中的调度程序概念，以及 *Oracle 数据库 PL/SQL 程序包和类型参考指南*中的编程接口。

启动 DBMS 调度程序工具之后，你将看到如下屏幕：



在屏幕左侧，可以看到调度程度对象类型的文件夹，以及一个用于正在运行的作业和作业运行的文件夹。窗口组、窗口和作业类都是系统级的，直接显示在对应文件夹的根目录下。计划、程序、链和作业归特定用户所有，因而每个文件夹都包含对应于所有者的子文件夹。

在屏幕右侧，可以看到选定对象的属性，对象的更多细节显示在右下侧。在创建或编辑对象时，可以在这里输入信息。

## 计划

时间表属性	
属性	值
Name	SCOTT.DAILY_SCHEDULE
Type	Calendar
► Start date	
End date	
Frequency	Daily
Interval	
By month	
By week number	
By year day	
By month day	
By week day	
By hour	23
By minute	0
By second	0
By setpos	
Comments	Schedule that runs daily at 23:00 hours

计划指明作业运行的时间和次数。计划可由多个作业共享。举例来说，一个商业季度的结束可能是许多作业的共同时限。你不必在每次定义新作业时定义季末计划，只需指向指定计划即可。

计划的类型可以是**日历**，允许指定重复间隔（如上图所示），也可以是**事件**，允许指定队列和条件：

时间表属性	
属性	值
Name	SCOTT.PROCESS_FINISHES
Type	Event
Start date	
End date	
Queue	SCOTT.PROCESS_QUEUE
AQ Agent	
Event condition	tab.user_data.process_status = 255
► Comments	Schedule for finished processes

## 窗口

**窗口属性**

属性	值
Name	TUESDAY_WINDOW
► Enabled	<input checked="" type="checkbox"/>
Resource plan	DEFAULT_MAINTENANCE_PLAN
<b>Schedule</b>	
Start date	
End date	
Frequency	daily
Interval	
By month	
By week number	
By year day	
By month day	
By week day	TUE
By hour	22
By minute	0
By second	0
By setpos	
Duration (days hh:mm:ss)	0 04:00:00
Priority	Low
Active	<input type="checkbox"/>
Comments	Tuesday window for maintenance tasks

**窗口组成员** | **详情**

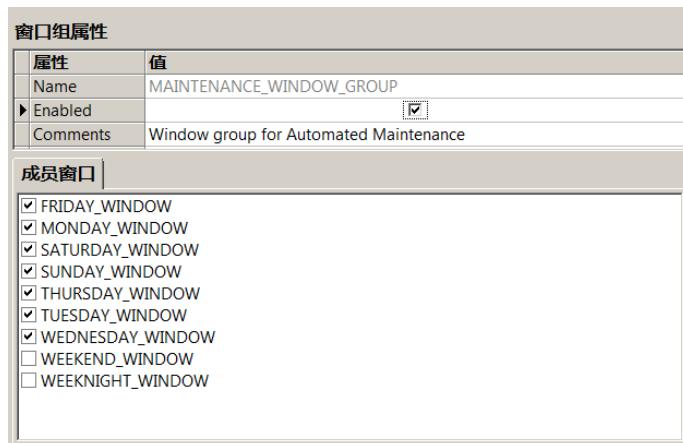
<input checked="" type="checkbox"/> MAINTENANCE_WINDOW_GROUP
<input checked="" type="checkbox"/> ORA\$AT_WGRP_OS
<input checked="" type="checkbox"/> ORA\$AT_WGRP_SA
<input checked="" type="checkbox"/> ORA\$AT_WGRP_SQ

窗口可用于自动启动作业，或者更改一天、一周等各种时间段内作业间的资源分配。窗口表示为具有明确定义的起始和结束时间的时间间隔，例如“从中午 12 点到早 6 点”。

可以为窗口显式输入各计划属性。这称为内置计划（inline schedule）。此外也可为窗口选择现有计划。所有计划属性都将以斜体显示，表示对应选定计划。如果你覆盖了一项或多项计划属性，选定计划将被清除，你将定义一个内置计划。这将允许你使用现有计划作为模板。

在下端，可以通过选中或清除相应选项来添加或删除当前窗口的窗口组成员身份。

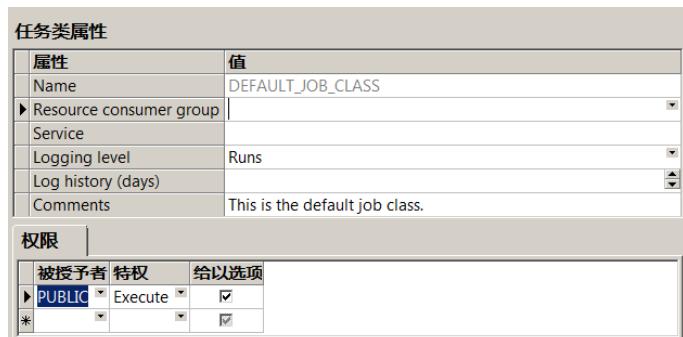
## 窗口组



可以创建窗口组来简化在调度作业时的使用。如果一项作业必须在一天、一周等多个时间段内运行，则可为每个时间段创建一个窗口，随后将窗口添加到窗口组。此后可将作业的计划设置为此窗口组，作业将在窗口组中置顶的所有这些时间段内执行。

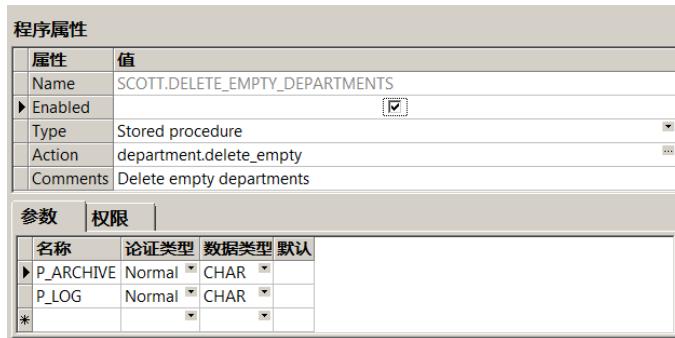
要添加或删除成员窗口，只需选中或清除屏幕下端**成员窗口**部分中的对应选项即可。

## 作业类



作业类提供了为成员作业指派同一组属性值的途径，也可为成员作业设置服务关联。你可以为需要使用此作业类的用户或角色授予执行权限。要撤销权限，右键单击并从弹出式菜单中选择**删除**即可。

## 程序



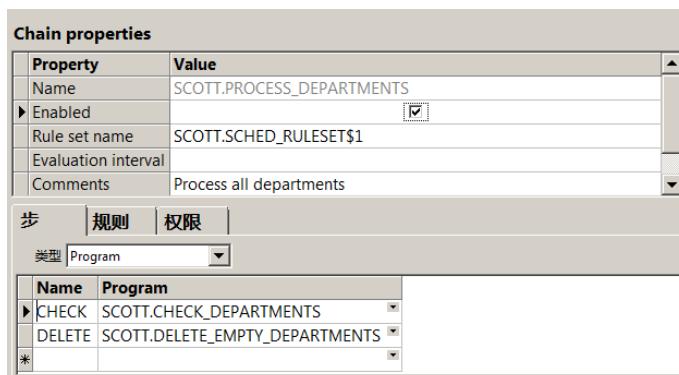
程序描述了调度程序将运行的内容。程序包含一个活动，类似于存储过程名称、OS 文件系统中发现的可执行文件的名称或 PL/SQL 匿名块的文本等。

程序是独立于作业的一个实体。作业在特定时间运行，或根据特定事件的出现而运行，作业将调用特定程序。作业可创建为指向一个程序，也就是说，不同的作业可以使用同一个程序，并在不同的时间使用不同的设置运行程序。给定正确的权限，不同的用户可以使用同一个程序，而无需重新定义。这实现了程序库的创建，用户可从已有程序的列表中进行选择。

如果参照的存储过程或外部可执行文件接受参数，程序可以定义参数。在参数部分中，你可定义这些参数。要删除参数，只需右键单击并在弹出式菜单中选择删除即可。

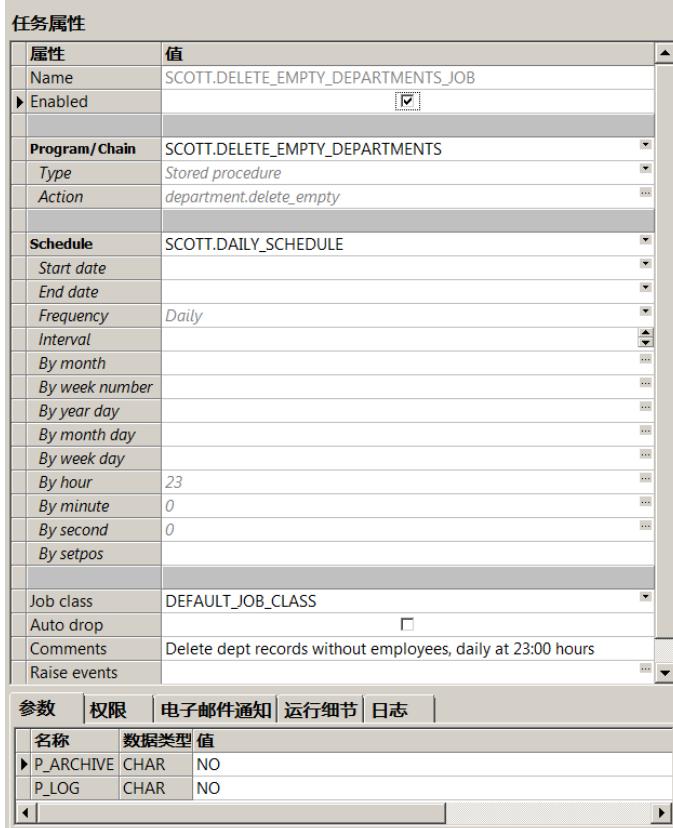
要为其他用户和角色指派程序的执行或更改权限，在权限部分中即可完成。要撤销权限，右键单击该权限并从弹出式菜单中选择删除即可。

## 链



链允许通过一组规则将多个独立步相连接。步的类型可为程序、事件时间表或内联事件。随后可以创建作业来运行这些链。

## 作业



作业将一个程序与一个计划（或窗口、窗口组）相关联，是你在调度程序中使用的主要对象。可以创建多个指向同一个程序的作业，但按照不同的计划运行。作业可覆盖程序参数的默认值，因而多个作业可指向同一个程序，同时提供不同的参数值。

通过指定要运行的程序单元、外部可执行文件或 PL/SQL 块来指定作业活动。这称为指定内置作业活动。此外，也可选择一个现有调度程序或链。

可以为作业显式输入各种计划属性。这称为内置计划。此外也可以为作业选择一个现有计划。所有计划属性都将以斜体显示，表示对应的选定计划。如果你覆盖了一项或多项目计划属性，选定计划将被清除，你将定义一个内置计划。这将允许你使用现有计划作为模板。

要为其他用户和角色指派作业的更改权限，在权限部分中即可完成。要撤销权限，右键单击该权限并从弹出式菜单中选择删除即可。

可以在屏幕下方的相应标签页中查看已完成作业的详情和日志。

## 10.1 创建对象

要创建新对象，在相应文件夹上右键单击，并从弹出式菜单中选择新建即可。现在，你可以在屏幕右侧输入属性和其他具体信息。完成之后，只需按下应用按钮即可在数据库中创建对象。要查看对象创建的 SQL，按下查看 SQL 按钮即可。要取消对象创建，请按下取消按钮。

## 10.2 编辑对象

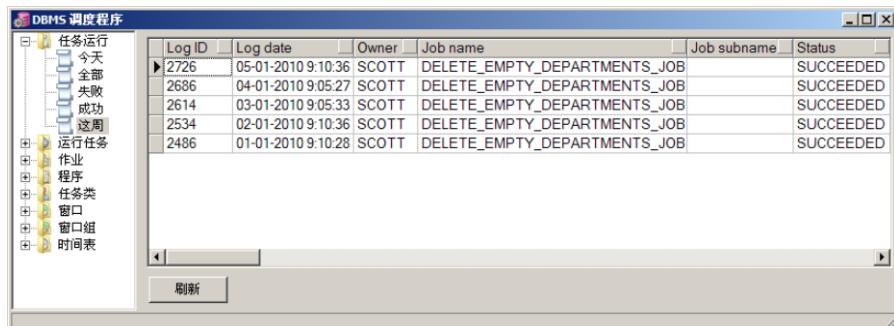
要编辑现有对象，首先在屏幕左侧的文件夹中选中该对象，随后在屏幕右侧编辑属性和其他具体信息。完成之后，只需按下应用按钮即可修改数据库中的对象。要查看对象修改的 SQL，按下查看 SQL 按钮即可。如果你尚未做出任何更改，则会显示对象创建 SQL。要取消更改，请按下取消按钮。

## 10.3 删除对象

要删除对象，首先在屏幕左侧的文件夹中选中该对象，并从弹出式菜单中选择删除。确认之后，对象将永久地从数据库中删除。

## 10.4 查看作业运行

要查看关于已经执行完成的作业运行的信息，可打开作业运行文件夹。在这里，你将看到多种预定义的报告，如全部、失败等。单击报告查看与报告的标准匹配的作业运行细节：



The screenshot shows the DBMS Scheduler application window titled "DBMS 调度程序". The left sidebar contains a tree view with categories like "任务运行", "运行任务", "作业", "程序", "任务类", "窗口", "窗口组", and "时间表". The main pane displays a table of job logs with the following data:

Log ID	Log date	Owner	Job name	Job subname	Status
2726	05-01-2010 9:10:36	SCOTT	DELETE_EMPTY_DEPARTMENTS_JOB		SUCCEEDED
2686	04-01-2010 9:05:27	SCOTT	DELETE_EMPTY_DEPARTMENTS_JOB		SUCCEEDED
2614	03-01-2010 9:05:33	SCOTT	DELETE_EMPTY_DEPARTMENTS_JOB		SUCCEEDED
2534	02-01-2010 9:10:36	SCOTT	DELETE_EMPTY_DEPARTMENTS_JOB		SUCCEEDED
2486	01-01-2010 9:10:28	SCOTT	DELETE_EMPTY_DEPARTMENTS_JOB		SUCCEEDED

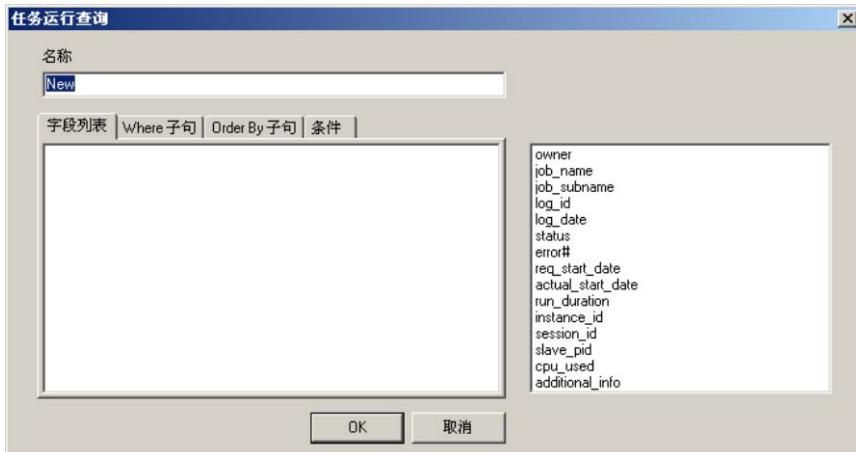
若要排序作业运行，只需单击列标题上的排序按钮即可。通过拖放标题即可移动列。

可以选中一系列行、列或单元格，右键单击并从弹出式菜单中选择以逗号分隔复制或以制表符分隔复制，将作业运行信息复制到剪贴板。如果未选择范围，则将复制整个报告。

要刷新报告内容，按下屏幕底端的刷新按钮即可。

## 创建自定义作业运行报告

要创建自定义报告，只需右键单击作业运行文件夹，从弹出式菜单中选择新建菜单项。此时将出现如下屏幕：



在此屏幕底端可以看到报告的逻辑名称。

在下面的部分中，可以为报告定义查询 `sys.all_scheduler_job_run_details` 视图的方式。关于此视图的更多信息，请参见 *Oracle 数据库参考指南*。右侧的字段选择列表可用于快速选择字段。双击一个字段即可将其应用到左侧的当前选定部分。

**字段列表**允许你选择报告中显示的字段。这是一个以逗号分隔的列表。也可以在字段列表中输入星号，包含所有字段。

**Where 子句**允许你定义报告中显示哪些作业运行。例如，要创建一份报告，显示自今天起所有成功的作业运行，可使用如下 where 子句：

`log_date >= trunc(sysdate) and status = 'SUCCEEDED'`

**Order by 子句**允许你定义一个字段列表，作业运行将按照此列表排序。

在**条件**标签页中，可以定义满足特定条件的作业运行的显示颜色：



条件可以是任何有效的 SQL 表达式。在颜色属性中，可以选择文本或背景。

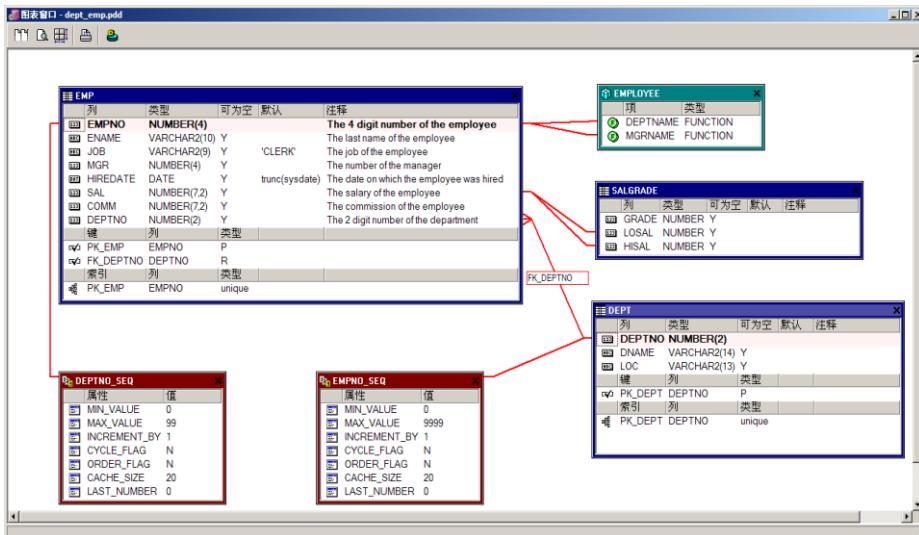
要编辑或删除报告，只需右键单击该报告并从弹出式菜单中选择相应的菜单项即可。

## 10.5 查看正在运行的作业

要查看当前运行的作业的相关信息，请单击 *正在运行的作业*文件夹。屏幕右侧将显示关于当前正在运行的作业的信息。

## 11. 图表

图表窗口允许你创建选定对象的图示，这样你就可以容易地看到（一部分）你的应用程序或工程的数据对象和它们之间的关系：



你可以添加所有相关的对象类型包括它们之间的关系到一个图表，表之间的外键参照将被自动包括在内。对象之间的从属参照也会被自动添加。对于巨大的图表，你可以通过包含对象组添加附加结构。

图表可用于文档目的，还能起到一个工作空间的作用。右键单击一个对象提供了访问所有对象的功能，双击一个对象将激活针对对象类型的默认操作。

### 11.1 创建图表

要创建一个新的图表，请从文件菜单中的新建子菜单选择图表窗口，一个空的图表窗口将显示出来。要添加一个数据库对象到图表，请从对象浏览器中把它拖到图表窗口中。你可以一次拖放多个对象。

要移动一个图表对象，请单击标题栏并且把它拖到新的位置。要调整一个图表对象的大小，你可以拖动一个边或者一个角。要删除一个对象，请按这个对象右上方“X”，这只会影响图表，数据库对象不会被删除掉。

#### 显示和隐藏项目

要显示或隐藏图表对象的一个指定项目，请右键单击它并转到项目子菜单。在这中你将看到一个带有关于所有项目的子菜单，它可以应用到已选对象。对于一个表，你可以选择你要显示的列、键或索引的哪些项目。你还可以隐藏全部的键或索引部分。

你还可以右击群组或图表，来显示或者隐藏群组或图表里的项目。

在默认的情况下，所有的项目都将显示，但是如果你右键单击了一个对象并且把选择了设为默认，那么可见的项目在将来将应用到所有的被添加到图表的对象。对于所有的对象类型，你都可以这样做。

### 项目排序

你可以右击图表项目，从子菜单中选择创建订单或者按字母为订单排序来为所有项目排序。你还可以拖动&拽拉一个项目，将它放到不同的位置来为项目排序。

### 添加关系

要在两个对象之间添加一个关系，请单击源对象的一个单元的图标，并把它拖到目标对象的单元，这样对象的两个单元之间将被画上一条线。通过选择线和上下移动选择点，你可以移动线。要删除一条线，请选择它并按 *Delete* 键。

如果你要添加表到图表，图表上的带有其他表的外键关系将被自动地包括在内。如果必要，这些关系可以被删除。

如果一个对象参照了其他对象，或被其他对象所参照，你可以右键单击对象，并从弹出式菜单中选择参照链接，可看到，它参照了 *DEPT HE EMP* 表：



在默认的情况下，关系将被画成一条直线。通过单击一条线并从弹出式菜单选择插入弯点项目，你可以添加附加的弯点。你还可以在单击要插入弯点的一条线的同时，按下 *Ctrl* 键。结果，一个新的选择点就被添加到了线上。

要更改线的端点，请右键单击线的端点，并从弹出式菜单中选择改变线端点菜单项。在这里，可以为一对多关系、可选关系和箭头选择符号。

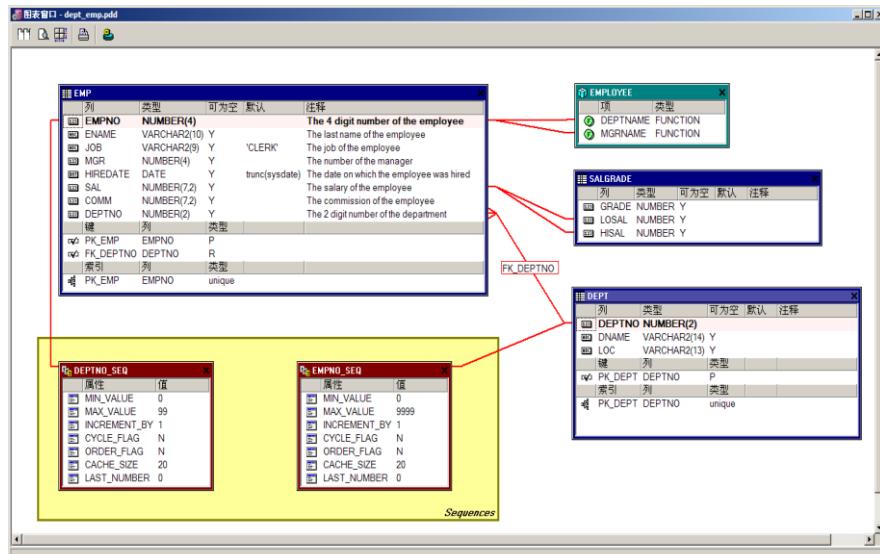
要更改线的颜色、粗细和线的标签，右键单击线并从弹出式菜单中选择线属性即可：



如果对象移动，标签将随线一起移动。

### 添加组

要添加附加结构到一个大的图表，你可以创建对象组。组可以有与众不同的颜色，可以在左（或右）上（或下）方有一个标题：



要添加一个组，请右键单击图表上的一个空白位置，从弹出式菜单选择新建组项目，一个新组将在这个位置显示出来，你可以把它调整到合适的大小。所有的被组完全掩藏的对象将隐含地成为那个组的一部分，移动组也将移动这些被附上的对象。当在移动组的同时，如果你按下了 *Ctrl* 键，被附上的对象将不被移动。

要更改一个组的外观，请右键单击它并从弹出式菜单选择属性项目，你双击组也可以。下列屏幕将显示出来：



你可以更改组的字体、填充颜色、线颜色和线尺寸，还可以使组透明，并使之根据文本自动调整尺寸。你可以随意输入组的标题、指定标题的位置。

如果组看起来彼此重叠，可以右键单击一个组，并从弹出式菜单中选择置于底层或置于顶层。

要创建一个默认组外观，请单击它并且选择设为默认，这在将来会应用到所有的添加到图表的组。

要从图表中删除一个组，请选择它并且按 *Delete* 键。这只会删除组，不会附带对象。

### 添加注释

可以为图表添加文本注释，只需在图表中的所需位置右键单击，并从弹出式菜单中选择新注释即可：



如你所见，注释的属性与组相同，但默认值不同。要创建默认注释外观，右键单击并选择设为默认即可。这将应用于将来添加到图表的所有注释。

### 对齐项目

可通过右键单击项目并从弹出式菜单中选择对齐到表格来快速对齐项目。这会将选定的项目置于一个 8 像素的表格中。要对齐所有项目，右键单击图表，并从弹出式菜单中选择全部对齐到表格，也可以按下图表窗口工具栏中的相应按钮。

## 11.2 缩放

可以右键单击图表并选择弹出式菜单中的缩放菜单项，缩放图表。可以将图表放大到 200%，或缩小到 10%。

## 11.3 保存和打开图表文件

要保存图表到文件，请选择来自于文件菜单的保存项目，图表将被保存到一个以 .pdd 为扩展名的文件中，以后你还可以重新打开它。要为图表文件定义一个标准位置，请转到目录首选项页并且设置相应的选项。

## 11.4 更新图表

图表是数据库的静态表达。随着时间的推移，如果数据库对象发生了变化，你就需要更新图表以反映这些变化。为此，请右键单击图表对象并且从弹出式菜单选择从数据库更新项目。要在图表上更新所有对象，请右键单击它并选择从数据库全部更新项目。也可以按下图表窗口工具栏中的相应按钮。

如果一个对象被从数据库中删除掉了，你也需要从图表中删除它。如果一个对象在数据库中被重新命名了，你也需要删除这个图表对象并且用新的名称添加它。

## 11.5 打印图表

要打印图表，单击主工具栏上的*打印*按钮。要打印当前视图，可以按下图表窗口工具栏中的*打印视图*按钮。在打印当前视图时，页面方向将自动确定。

可以使用图表窗口工具栏上的*页面大小*按钮来控制页面大小。将其设置为*自动*即可自动确定恰当的大小。设置为*手工*即可手工控制页面大小。对于较大的图表，可以缩小，并将水平和垂直页面结束位置移动到恰当的位置，或移动图表对象，使之不会跨多页显示。

## 11.6 导出图表

可以右键单击图表并从弹出式菜单中选择*另存为图片*将图表导出为图像文件。随后可以选择文件格式（*bmp*、*jpg*、*wmf* 或 *emf*）。

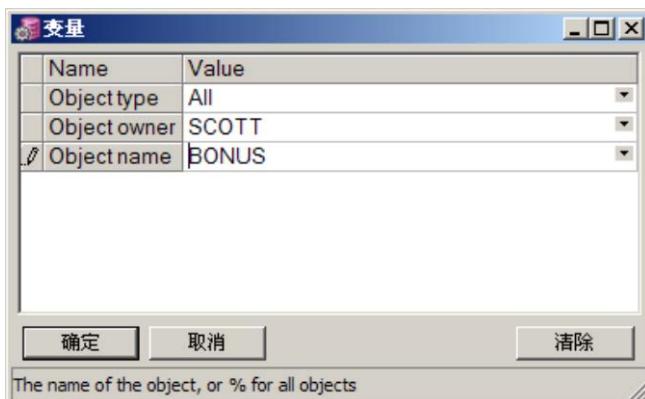
还可以将图表图片复制到剪贴板，只需右键单击并从弹出式菜单中选择*复制为图片*即可。

## 12. 报告

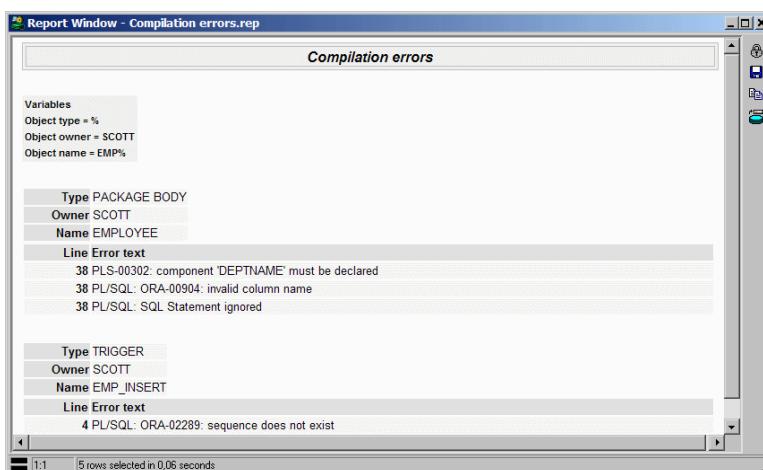
要运行面向你的应用数据或 Oracle 词典的报告，你可以使用 PL/SQL Developer 内置的报告功能。这中有很多标准的报告，并且你还可以创建你自己的定制报告。这些定制报告能被保存在报告文件中，它们能依次被包括在报告菜单中，这对经常运行自己习惯的定制报告是很容易的。

### 12.1 标准报告

标准报告通过报告主菜单是可以访问的。例如，如果你选择了编译错误报告，你将首先被提示输入你要显示当前编译错误的对象类型、所有者和名称：



在输入了适当的值并按下了确定按钮后，报告将显示：



其结果是一个 HTML 文档，可以通过 PL/SQL Developer 内部的 HTML 查看器或者 Internet Explorer ActiveX 控件来显示，这两种显示方法的选择依赖于首选项（请参见第 20.10 节）。

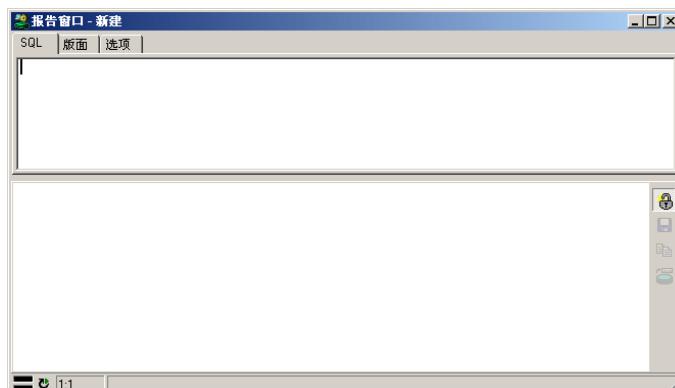
随后你可以通过按下工具栏中的 **打印** 按钮来打印报告，或者你可以通过按报告窗口右边的 **保存结果** 按钮来保存 HTML 格式的报告。**复制为 HTML** 按钮将复制结果到剪贴板。**导出结果** 按钮允许你导出结果为 CSV、TSV 或 XML 格式，或者直接导出结果到 Excel。

如果报告没有被锁定，**编辑报告** 按钮提供了对报告定义的访问。

你可以右键单击报告来选择另外的选项。

## 12.2 定制报告

创建定制报告是很容易的。首先你通过工具栏中的新建按钮创建一个新的空白报告，然后从弹出式菜单选择 **报告窗口** 项目。一个空的报告窗口将显示出来：



在 SQL 编辑器中，你可以为你的报告键入一个简单的 SQL select 语句。对于主题/明细报告，你可以使用内部联接的 select 语句（这将在以后描述）。让我们通过键入 select 语句和按下工具栏中的 **执行** 按钮，简单地以清楚明白的 emp 表来开始：

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7782	CLARK	MANAGER	7839	9-6-1981	2450,00		10
7839	KING	PRESIDENT		17-11-1981	5000,00		10
7934	MILLER	CLERK	7782	23-1-1982	1300,00		10
7876	ADAMS	CLERK	7788	23-5-1987	1100,00		20
7902	FORD	ANALYST	7566	3-12-1981	3000,00		20
7566	JONES	MANAGER	7839	2-4-1981	2975,00		20
7788	SCOTT	ANALYST	7566	19-4-1987	3000,00		20

[223] [14行已被选择 耗时 0.125秒]

这是一个很基本的报告，仅仅带有已应用的默认字体属性。下面章节将描述你怎样才能精制报告功能。

## 12.3 变量

在你可以运行前，你的报告经常需要某些额外的输入。让我们来假定你要限定来自于前面章节的报告到仅仅一个部门的雇员。如果是那样的话，你需要在 SQL 文本中提供一个置换变量：

```
select * from emp
where deptno = &department
order by ename
```

当这个报告被运行时，在 select 语句被运行前，用户会被提示给 department 输入一个值。这当然并不十分安全。要是用户什么都不输入呢？这个语句将会导致 ORA-00936：缺少表达式异常。要是用户输入的值不是数字呢？或者要是输入的数字在 dept 表中不存在呢？要防止这些情况，你可以使用下面章节中描述的参数选项。这些选项被指定在两个括号之间，就像 HTML 或 XML 文档中的属性一样。

### Name 选项

在变量输入表中，变量的名称被作为提示来使用，所以你应该确认它确实描述了变量的含义。如果名称仅仅是一个选项，那么它仅仅是一个不必在两个括号之间指定的选项。如果名称包括空格或其他特殊字符，那么就用双引号引上它。下面的两个例子是等效的：

```
select * from emp
where deptno = &" Department number"
order by ename
```

和

```
select * from emp
where deptno = &<name=" Department number" >
order by ename
```

### Hint 选项

除了名称选项之外，你还可以指定提示文本。当变量有焦点时，这个提示将被显示在变量输入表的状态行上：

```
select * from emp
where deptno = &<name=" Department number"
      hint=" Only employees from this department will be listed" >
order by ename
```

### Type 选项

类型选项限定用户能被输入的信息，也控制什么样的值应该被插入到作为结果的 SQL 文本：

```
select * from emp
where deptno = &<name=" Department number"
      type=" integer" >
order by ename
```

在这种情形下，用户只能输入有效的整数值，输入的值将原封不动地被放到 SQL 文本中。然而对于字符串，你要把值用引号引上。设想一下下列的 select 语句：

```
select * from emp
where ename = &<name=" Employee name"
      type=" string" >
```

如果你在变量输入表中输入 SCOTT（没有引号），作为结果的 SQL 将是：

```
select * from emp
where ename = 'SCOTT'
```

有效的类型是： *none* 、 *string* 、 *integer* 、 *float* 和 *date* 。对于日期值，有效将不出现，所以用户可以另外提供诸如 *sysdate* 、 *trunc(sysdate, 'MM')* 等表达式。

### Required 选项

对于某一变量，如果你要求有一个值，那么就设它的 *required* 选项为 “*yes*” 或 “*true*”：

```
select * from emp
where deptno = &<name=“Department number”
      required= “yes” >
order by ename
```

现在，对于部门号码，如果一个值被指定，那么报告只能被运行。

### Default 选项

要为变量提出一个默认值，就要指定 *default* 选项：

```
select * from emp
where deptno = &<name=“Department number”
      default= “10” >
order by ename
```

默认值也可以是一个 select 语句，这个 select 语句应该仅仅能返回一行一列。例如：

```
select * from emp
where deptno = &<name=“Department number”
      default= “select min(deptno) from dept” >
order by ename
```

在这种情况下，默认值将是最小的部门号码。

### Ifempty 选项

作为一种选择，要生成一个需要的变量或提供一个默认的值，万一用户不指定它，你可以指定 *ifempty* 值：

```
select * from emp
where ename like &<name=“Employee name”
      type= “string”
      ifempty= “%” >
```

如果用户不指定一个值，百分号就会被使用，所有的雇员将会被获取。这个选项不能与 *required* 选项一起被指定。

### Uppercase/Lowercase 选项

如果你有一个字符串变量要隐式地把它的值转换到大写或小写，那么就设置它的 *uppercase* 或 *lowercase* 选项为 “*yes*” 或 “*true*”：

```
select * from emp
where ename = &<name=“Employee name”
      type= “string”
      uppercase= “yes” >
```

值作为形态被显示，但在作为结果的 SQL 文本中被转换为大写或小写。

### Readonly 选项

变量可以通过指定 `readonly=" yes"` 被定义为只读。用户可以看到变量和它的值，但不能更改它。

### Hidden 选项

你可以通过指定 `hidden=" yes"` 来隐藏变量。

### List 选项

*List* 选项为用户提供可能的值的列表，你可以指定项目以逗号分隔的列表：

```
select * from emp
where deptno = &lt;name=" Department number"
      list=" 10, 20, 30, 40" >
order by ename
```

这允许用户选择四个项目中的一个，或键入一个不同的值。在这种情况下，使用 `select` 语句会有更多的意义：

```
select * from emp
where deptno = &lt;name=" Department number"
      list=" select deptno from dept order by deptno" >
order by ename
```

你还可以为每个项目提供一种描述：

```
select * from emp
where deptno = &lt;name=" Department number"
      list=" 10, ACCOUNTING,
            20, RESEARCH,
            30, SALES,
            40, OPERATIONS"
      description=" yes" >
order by ename
```

现在每个项目都被描述跟随着，并且 `description` 选项被添加了。这个列表仅仅显示描述，实际值将被用于作为结果的 SQL 文本。

此外，你还可以用两个字段的 `select` 语句做同样的事情：

```
select * from emp
where deptno = &lt;name=" Department number"
      list=" select deptno, dname from dept order by dname"
      description=" yes" >
order by ename
```

要限定用户到列表中正确的项目，就要指定 `restricted` 选项：

```
select * from emp
where deptno = &lt;name=" Department number"
      list=" select deptno, dname from dept order by dname"
      description=" yes"
      restricted=" yes" >
order by ename
```

现在，用户只能从列表中选择值了，不能手工输入任何其他值。

要创建多选列表，可指定 `multiselect` 选项：

```
select * from emp
where deptno in (&<name="Department number"
    list="select deptno, dname from dept order by dname"
    description="yes"
    multiselect="yes">)
order by ename
```

此时，用户即可按描述选择多个部门。你需要使用 *field in (&variable)* 语法构造 select 语句，以返回与选择相符的记录。

列表中的项目有时能依赖另外一个变量的值。假设你要创建显示指定表所有列的报告，表通过所有者和名称来鉴别，对于所有者你可以使用数据库中所有用户的列表，对于表名你可以查询已选所有者的 *all\_tables* 视图：

```
select * from all_tab_columns
where owner = &<name="Owner"
    type="string"
    list="select username from all_users order by username">
and table_name = &<name="Table"
    type="string"
    list="select table_name from all_tables
        where owner = :owner
        order by table_name">
order by column_id
```

正象你可以看到的，第二个（表）列表的 select 语句通过 *:owner* 绑定变量参照了第一个变量。只要 *owner* 的值被更改，表列表就将被移民。

注意，绑定变量的名称不能长于 30 个字符，不能包含空格或其他特殊字符。变量名中的空格将被转换为下划线。如果变量名是 *Owner of the table*，则绑定变量名将是 *:owner\_of\_the\_table*。

还要注意的是，如果变量的值是空的，则依赖的列表也将是空的。在这种情形下，查询不会被运行。

### Checkbox 选项

如果用户选择两个可能性中的一个，你可以使用 *checkbox* 选项。这个选项需要有已检验和未检验值跟随：

```
select * from emp
order by hiredate &<name="Descending sort order"
    checkbox="desc, asc">
```

即使你总是必须指定两个值，其中之一也还是可以为空：

```
select * from emp
order by hiredate &<name="Descending sort order"
    checkbox="desc, ">
```

因为都默认为上升排序，所以这两个例子是等效的。

## Prefix 和 suffix 选项

如果变量值是空的，这可能意味着 SQL 文本的某一固定部分也必须被省略。设想一下你要提供一个可选择的排序列的例子，如果列没有被指定，*order by* 子句应该被完全忽略。在这种情况下，你可以指定这个文本为变量的 *prefix* 或 *suffix*。如果变量值是空的，前缀和后缀还是被忽略为好：

```
select * from emp
&<name=" Descending sort column"
list=" empno, ename, hiredate"
prefix=" order by "
suffix=" desc">
```

如果这个变量的值没有被指定，那么结果还是为空好：

```
select * from emp
```

如果列被指定，结果应该是：

```
select * from emp
order by empno desc
```

要注意前缀后面的和后缀前面的空格，这些是重要的，因为前缀、值和后缀都象被指定那样紧密地链接在一起！

## Lines 选项

此选项控制多行文本变量输入表单上的行数（默认值为 1）：

```
select sum(sal) as saltotal,
&<name="Notes" type="string" lines="4"> as notes
from emp
```

上例将为 *Notes* 变量显示 4 行。

## Editor 选项

此选项为变量输入字段添加一个按钮，将会打开一个文本编辑器来编辑值。可以将语法变高亮显示选项指定为 Plain（不高亮显示）、SQL 或 XML：

```
select sum(sal) as saltotal,
&<name="Notes" type="string" lines="4" editor="plain"> as notes
from emp
```

上例将为 *Notes* 变量显示 4 行。

## 换码符

如果你希望在 SQL 文本中使用应该不被解释为置换变量的 & 符号，那么就用双 & 符号来代替。下面的例子将获取来自于‘R&D’部门的所有雇员：

```
select * from emp
where emp.deptno in (select dept.deptno from dept
where dname = 'R&&D')
order by empno desc
```

如果代之以使用文本‘R&D’，你就会因 D 变量而被提示。

## 12.4 精制版面

打开了一个新的报告窗口以后，键入一个 select 语句，再运行报告，版面将使用默认的样式属性。它将有一个简单的表格式版面，带有标准的字体属性、表样式、颜色等。要精制标准的版面，请转到版面标签页。对于新的报告，这一页将如下所示：

The screenshot shows the PL/SQL Developer Report window titled "报告窗口 - select \* from emp order by deptno, ename". The window has tabs for SQL, 版面 (Report), and 选项 (Options). The Report tab is selected. A checkbox labeled "自动更新" (Auto Refresh) is checked. Below it is a table with columns: 项目 (Item), 描述 (Description), 风格 (Style), 页眉 (Page Header), 对齐 (Align), 格式 (Format), and 中断 (Break). Items listed include Report Title, Variables, Tabular Tables, Form Tables, and Default Field. The main area displays a table of employee data:

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7782	CLARK	MANAGER	7839	9-6-1981	2450,00		10
7839	KING	PRESIDENT		17-11-1981	5000,00		10
7934	MILLER	CLERK	7782	23-1-1982	1300,00		10
7876	ADAMS	CLERK	7788	23-5-1987	1100,00		20
7902	FORD	ANALYST	7566	3-12-1981	3000,00		20
7566	JONES	MANAGER	7839	2-4-1981	2975,00		20
7788	SCOTT	ANALYST	7566	19-4-1987	3000,00		20
7369	ADAMS	CLERK	7369	17-12-1980	200,00		10

At the bottom left, there are status bars showing "1.1" and "14行 已被选择 耗时 0.125秒".

正如你看到的，你可以为所有版面项目（报告标题、变量、表和字段）定义各种版面属性。如果你要为一个单独的字段定义属性，你首先必须按下工具栏中的刷新字段列表按钮：

This screenshot shows the same Report window after applying style properties to specific fields. The "EMPNO", "ENAME", "JOB", and "MGR" fields now have their own style definitions in the Report Properties table. The main report table remains the same as in the previous screenshot.

现在，select 语句结果集的所有字段都被包括到版面表格中了，你可以设置每个单独字段的版面属性。如果你把字段的样式、标题或对齐保留为空，那么来自于默认字段的相应属性将被使用。

更改了版面属性以后，你可以再次运行报告来查看结果的效果。你也可以激活工具栏中的[自动更新](#)选项，这样每个更改都会马上反映到结果中。

下面的章节详细地描述了各种各样的版面属性。

## 显示

版面表格最左边一列的复选框指出版面项目是否应该被显示，这允许你禁用报告标题或变量，允许你控制指定字段是否应该被显示。

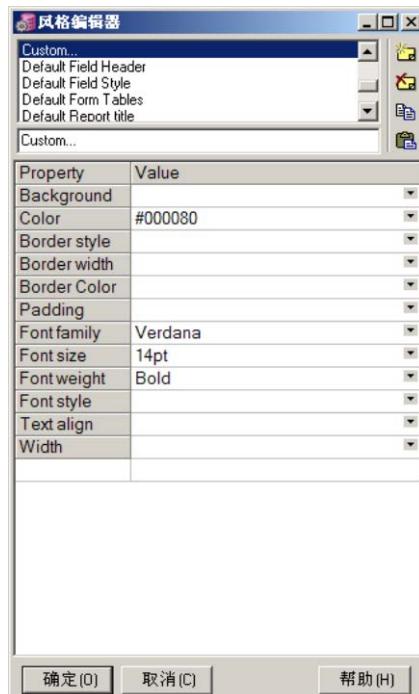
## 描述

报告标题的描述将在报告的顶部和在每个分页符之后显示，[变量](#)的描述将被显示在变量的上面。

对于单独的字段，你可以越过标准的字段名来定义一个描述，这个描述将用于列标题。

## 样式

样式控制了版面项目的外观，按单元格按钮 (...) 就调出了样式编辑器：



在样式编辑器顶部的列表栏中，你可以看到所有的标准样式和[定制](#)样式。你可以从样式库（请参见第 12.5 节）中选择一个标准的样式，或者你可以选择定制的样式以便你能为当前的版面项目更改各种样式属性。

注意，这个样式系统是基于层叠式样式表（CSS）的标准。你在样式编辑器中看到的属性是最常用的，当然你还可以在标准的属性下面添加其他的 CSS 属性和值。例如，你可以添加一个值为 *20pt* 的 *Height* 样式属性，它把单元格高度设置为 20 个点。关于层叠式样式表的更多信息，请参见 <http://www.w3.org/TR/REC-CSS2>。

**复制与粘贴**按钮允许你复制一个样式的属性，所以你可以快速地得到一个来自于现有样式的新的样式。

如果你没有为字段样式定义什么，来自于样式库的 *Default Field* 样式将被使用。你可以通过定义 *Default Field* 版面项目的样式来越过这个，你可以在字段级通过指定相应的样式来越过这个默认样式。注意，每个字段标题继承了它的来自于 *Default Header* 样式的样式属性，所以你只需越过你要更改的属性。例如，如果你定义了 *Default Field* 样式为 Verdana, 14 pt, blue，并且你要显示的 *EMPNO* 列为 Verdana, 14 pt, red，那么你只需在 *EMPNO* 级定义颜色。字体和字号将自动地从 *Default Field* 级被继承。

各种版面项目的默认样式在第 12.5 节中有描述。

## 标题

正如你能为个别字段数据设置样式，你还能为它们的标题设置样式。如果你没做任何定义，来自于样式库的 *Default Field Header* 样式将被使用。你可以通过定义 *Default Header* 版面项目的样式来越过这个，你可以通过指定相应的标题样式在字段级越过这个默认样式。注意，每个字段标题都继承了它的来自于 *Default Header* 样式的样式属性，所以你只需越过你要更改的属性就可以。

## 对齐

你可以按下列方法对齐版面项目：

- 左 - 项目在单元格中左对齐
- 右 - 项目在单元格中右对齐
- 居中 - 项目在单元格中居中对齐
- 默认 - 数字右对齐，其他项目左对齐
- 无 - 不指定对齐

注意，你还可以为版面项目设置文本对齐样式属性，这将把先后次序带到你可以在版面表格中指定的对齐选项。还要注意的是，对齐的效果可能依赖于版面项目的 *Width* 样式属性。

## 格式

在默认的情况下，字段的格式被数据类型、数值范围和精度所控制。你常常要使用被计算的字段或合计的字段，这样数值范围和精度是未知的。在这种情况下，你可以指定一个格式。

格式属性仅仅对日期和数字字段有效果。对于数字字段，你可以使用 9 为禁用零的阿拉伯数字、0 为正常的阿拉伯数字、G 为千分隔符、D 为十进制分隔符、E 为科学符号。另外，你还可以在单引号或双引号中指定文字文本。一些例子：

格式	值 = 1234	值 = 5	值 = 0.1
999G999D00	1,234.00	5.00	0.10
9G990	1,234	5	0

0000	1234	0005	0000
0"%"	1234%	5%	0%
0D000e+00	1. 234e+03	5. 000e+00	1. 000e-01

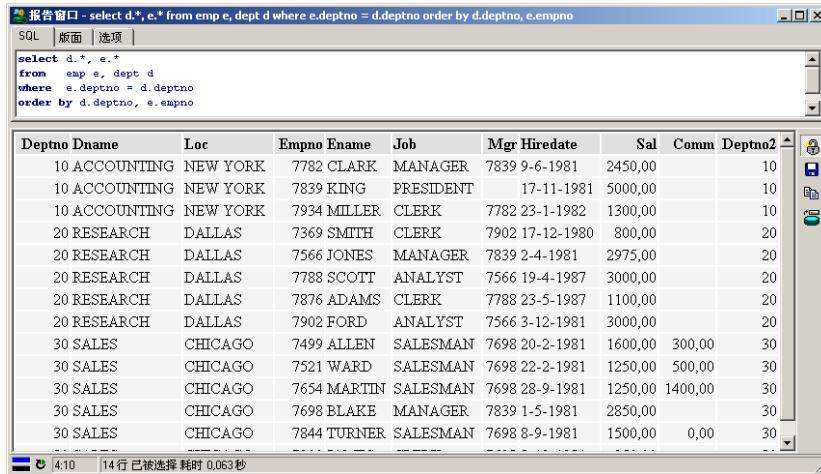
对于日期字段，你可以使用标准的 Windows 日期和时间格式。

## 中断

中断版面属性允许你构成你的报告结果。假定你要显示部门和它们的雇员，你可以为此指定下面的查询：

```
select d.*, e.*
from   emp e, dept d
where  e.deptno = d.deptno
order by d.deptno, e.empno
```

这当然会导致一个简单的表格式报告：



The screenshot shows the PL/SQL Developer interface with a report window open. The window title is "报告窗口 - select d.\* , e.\* from emp e, dept d where e.deptno = d.deptno order by d.deptno, e.empno". The SQL tab contains the following query:

```
select d.* , e.*
from   emp e, dept d
where  e.deptno = d.deptno
order by d.deptno, e.empno
```

The results are displayed in a grid table with the following columns: Deptno, Dname, Loc, Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, and Deptno2. The data is as follows:

Deptno	Dname	Loc	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno2
10	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	9-6-1981	2450,00		10
10	ACCOUNTING	NEW YORK	7839	KING	PRESIDENT		17-11-1981	5000,00		10
10	ACCOUNTING	NEW YORK	7934	MILLER	CLERK	7782	23-1-1982	1300,00		10
20	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17-12-1980	800,00		20
20	RESEARCH	DALLAS	7566	JONES	MANAGER	7839	2-4-1981	2975,00		20
20	RESEARCH	DALLAS	7788	SCOTT	ANALYST	7566	19-4-1987	3000,00		20
20	RESEARCH	DALLAS	7876	ADAMS	CLERK	7788	23-5-1987	1100,00		20
20	RESEARCH	DALLAS	7902	FORD	ANALYST	7566	3-12-1981	3000,00		20
30	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20-2-1981	1600,00	300,00	30
30	SALES	CHICAGO	7521	WARD	SALESMAN	7698	22-2-1981	1250,00	500,00	30
30	SALES	CHICAGO	7654	MARTIN	SALESMAN	7698	28-9-1981	1250,00	1400,00	30
30	SALES	CHICAGO	7698	BLAKE	MANAGER	7839	1-5-1981	2850,00		30
30	SALES	CHICAGO	7844	TURNER	SALESMAN	7698	8-9-1981	1500,00	0,00	30

At the bottom left of the report window, it says "4:10 | 14行 已被选择 耗时 0.063秒".

通过在 *LOC* 列指定一个中断，你会为你的报告获得更好的结构：

The screenshot shows the PL/SQL Developer interface with a report window titled "报告窗口 - select d.\*, e.\* from emp e, dept d where e.deptno = d.deptno order by d.deptno, e.empno". The report is displayed in a grid format. A toolbar at the top includes icons for file operations like Open, Save, and Print. Below the toolbar is a "版面" (Layout) tab selected. In the layout editor, the "LOC" column is highlighted with a red border, indicating it is the current target for a break header. The report data shows employees grouped by department (DEPTNO), with each department having multiple rows of employee information. The "LOC" column contains values like NEW YORK, DALLAS, and CHICAGO.

一直到（包括） *LOC* 的字段的重复值将被禁用显示。简而言之，每个 *dept* 记录只被显示一次。在这种情况下，记录被按部门（*DEPTNO* 列）排序是必要的。如果你要重复每个中断后面的标题，请使用 *Break + Header* 选项。

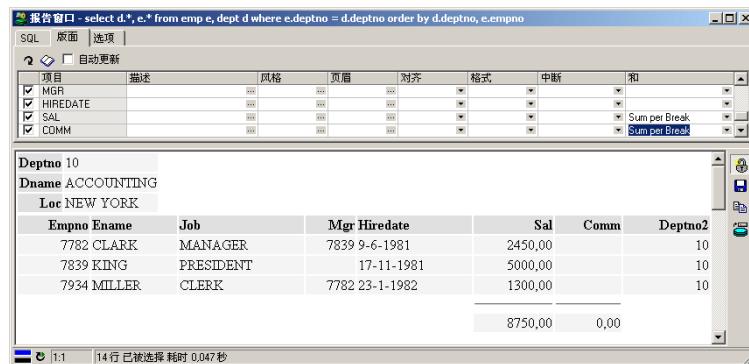
你也可以选择 Master/Detail 版面：

This screenshot shows the same report setup but with the "Master/Detail" option selected in the layout editor for the "LOC" column. The resulting report is presented in two distinct sections for each department. The first section, labeled "Deptno 10", contains the header "Dname ACCOUNTING" and "Loc NEW YORK", followed by a table of employees for department 10. The second section, labeled "Deptno 20", contains the header "Dname RESEARCH" and "Loc DALLAS", followed by a table of employees for department 20. This demonstrates how a single column can serve as both a primary key for grouping and a secondary key for defining detailed sections of the report.

对于一个报告，你可以有多重中断。查询的 *order by* 子句必须保证结果集在所有的中断列上都被排序。

和

和属性允许你包括一个字段的和:



你可以指定和在中断级、报告级或两者都有被计算。

## 页眉

要定义页眉，你首先必须设置页的大小（请参见第 12.6 节）。报告标题将在每一分页符之后被重复，还可以包括下列变量：

- OSUser - 操作系统用户。
- DBUser - 链接到数据库的用户。
- Database - 用户链接到的数据库。
- Date - 当前日期。
- Time - 当前时间。
- Page - 当前页。

要在页眉中包括这样一个变量，在它之前要有一个 & 字符。例如：

```
All employees (&dbuser@&database)
```

如果你要运行用户为 *scott*、数据库为 *chicago* 的报告，页眉将象下面这样：

```
All employees (scott@chicago)
```

既然报告被设定为 HTML 格式，你还能把 HTML 代码包括在中。要包括一个图象，你可以包括 <img> 标签：

```
All employees <img src=logo.gif>
```

要包括左对齐、居中和右对齐信息，你可以使用带有三个单元格的单行表，并且在相应的单元格中包括这些信息：

```
<table width="100%"> <tr> <td>&dbuser</td> <td align="center">All Employees</td> <td align="right">&page</td> </tr> </table>
```

这被显示为：

Scott	All Employees	1
-------	---------------	---

注意，你可以在所有版面项目的描述中使用这些变量和 HTML 代码，但是对于报告标题它们不是最常用的。

## 12.5 样式库

代替一遍又一遍地定义和重新定义同样的样式，你可以按下工具栏中的**样式库**按钮来创建与修改你的标准样式。例如，如果你想把包含 SQL 表达式的所有字段都显示为 Courier New, 12 pt, blue，那么简单地定义一个在库中带有这些属性的 *SQL* 样式，再应用它到相应的地方即可。

中有许多有特定含义的样式名：

样式名	描述
Default Report	将被应用到报告标题
Default Variables	将被应用到变量
Default Tabular	将被应用到所有表格式表（例如明细记录）
Default Form Tables	将被应用到所有表单式表（例如主记录）
Default Field Style	将被应用到没有已定义样式的所有字段
Default Field	将被应用到没有已定义样式的所有字段标题

这些都是默认的样式，所以在个别的报告中你可以通过定义相应的版面项目样式来越过它们。

注意，所有已用的样式（标准的和定制的）都被存储在报告文件中。当一个报告被运行和一个样式需要被应用时，PL/SQL Developer 将尝试在样式库中定位它。当它不可用时，来自于报告文件的样式将被使用。这将确保若非样式仍然被否决你的报告总是能正确地显示出来。

## 12.6 选项

你可以为你的报告指定各种不同的选项：



### 链接参数

链接参数决定了报告查询将怎样运行。例如，如果在 *SYS* 的账号下运行你的报告是必须的，那么你可以在用户名的输入框中输入这个名称。另外，你可以输入口令，或者你可以保留它为空。如果你保留它为空，当报告被运行时，你将被提示输入口令。如果报告应该总是在同样的数据库实例中被运行，你还可以指定数据库。你也可以在数据库输入框中输入 \* 号，它表示报告应该在当前的数据库中被运行。

使用本链接选项允许你禁用或允许报告的链接参数。如果它被禁用，当前的 PL/SQL Developer 链接将被使用。

### 安全

安全选项允许你锁定报告定义（SQL，版面和选项），以至于用户不能查看或更改它。如果链接参数指定了一个有权限的带有口令的账户（例如 *SYS*），安全选项也许是有效的。如果一个用户能够更改 SQL，若报告没有被锁定，他或她就能够运行 *SYS* 帐户下的任何语句。

安全能被设置为下列值之一：

- 不锁定 - 任何人都能更改报告定义
- 锁定其他用户 - 如果你知道报告的口令，或者如果你带有与链接参数中被指定的一样的账号在 PL/SQL Developer 中被链接，只有你才能更改报告的定义。
- 锁定所有用户 - 在你能更改报告的定义前，你总是被提示输入报告的口令，不管当前你是怎样在 PL/SQL Developer 中被链接的。

如果报告被锁定，报告文件 (.rep) 就被加密了，以至于不能用文本编辑器来查看或更改它。

## 版面选项

- 表单版面 - 在默认的情况下，除了主记录以外，所有记录都在表格版面上显示。主记录在表单版面上显示。通过激活这个选项，所有记录都将在表单版面上显示。如果每一条都有很多与/或宽字段，不能显示在一个单一行上，这个选项就可能是有用的。
- 左边距 - 报告左边空白处的像素数量。这个选项仅仅工作于如果你使用 Internet Explorer 作为 HTML 查看器时。
- 上边距 - 报告顶部空白处的像素数量。这个选项仅仅工作于如果你使用 Internet Explorer 作为 HTML 查看器时。
- 分页符 - 行的数量应在分页符后面显示，报告标题将在每个分页符后面重复再现。
- 仅 Nbsp - 当它被允许时，数据中所有的空格都将被转换到 HTML 输出中的 Nbsp（不断开空格）。当它被禁用时，正常的空格将被使用。这能影响到版面，因为 HTML 查看器将显示连续的空格为一个空格。
- 包括 SQL - 当它被允许时，报告的 SQL 文本将被包含在 HTML 输出中。

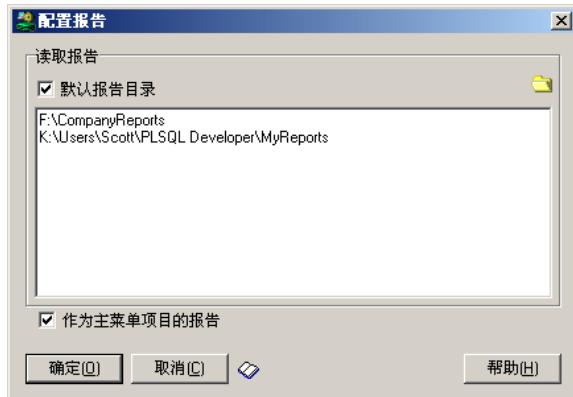
## 其他选项

- 背景图象 - 应该被显示在这个报告的背景中的图象 (.jpg 或 .gif)。你可以使用选择文件按钮来查找你的图象文件。
- 背景颜色 - 应该被显示在这个报告的背景中的颜色。

## 12.7 报告菜单

在默认的情况下，报告菜单项目被包括在主菜单中，只包含了标准的报告。这些标准的报告位于 PL/SQL Developer 安装目录的 Reports 子目录中，菜单是这个目录结构的简单表现。选择来自于报告菜单的一个项目将运行那个报告。

你可以在这个目录中添加或删除报告，你还可以通过工具菜单中的配置报告项目来配置报告菜单。这将调出下列对话框：



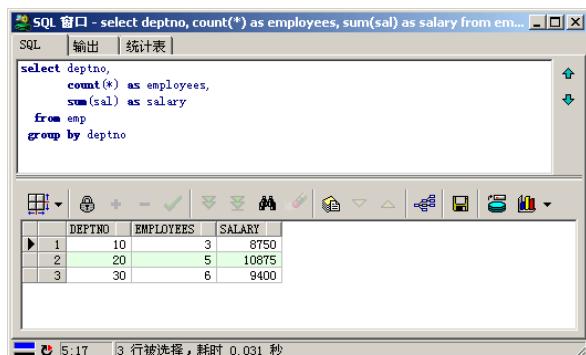
默认报告目录选项指出了报告菜单是否应该包括来自于 PL/SQL Developer 安装目录的 Reports 子目录，这个目录包含了标准的报告。这个选项下面的编辑区包含了应该被包括的目录列表。你可以用手工键入目录，或者你可以通过按右边的添加目录到列表按钮选择一个目录。

注意，报告文件和子目录将被并入到报告菜单中。你应该设法防止在这个目录列表中有重复的子目录和报告文件名，否则，你将在报告菜单中获得不明确的菜单项目。

样式库也可以在这个报告配置对话框中访问到（请参见第 12.5 节）。

## 13. 图形

要快速地显示一个在 SQL 窗口或报告窗口中查询的列数据的图示，你就可以使用图形窗口了。设想一下下列的 SQL 窗口中的查询：

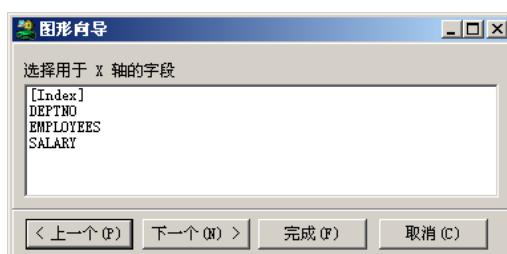


要想看到部门工资的图形，你可以按图形按钮：



同样的按钮还可用于报告窗口。它起始于下面描述的图形向导，或者你可以选择预先定义好的图形样式或模板，并且适当的列将被选为 X 轴和 Y 轴。

图形向导将首先让你选择用于 X 轴的数据：



这中给出了所有的列和一个 [索引] 项目，这个索引项目用于你没有 X 轴的值但要在 Y 轴上显示按照具体值排列的数据之时。在这种情况下，我们要按部门编号来显示工资，所以我们选择 DEPTNO 并按下一个按钮来选择用于 Y 轴的数据：



在选择了用于 Y 轴的 *SALARY* 之后，你可以按下一个按钮来选择不同的图形样式属性：



按下了完成按钮之后，结果图形窗口将显示：



图形窗口的工具栏提供了下列功能：

- 旋转 - 按鼠标左键并移动鼠标围绕着 X 和 Y 轴来转动图形。
- 移动 - 按鼠标左键并移动鼠标越过窗口来移动图形。通过按鼠标右键，这个功能一直是可用的，与被激活的工具栏功能无关。
- 缩放 - 按鼠标左键并移动鼠标来放大或缩小图形。
- 纵深 - 按鼠标左键并移动鼠标来增加或减少图形的纵深。
- 3D - 在 2D 和 3D 视图之间切换图形。
- 编辑 - 编辑图形数据和版面。
- 另存为模板 - 保存图形版面为模板以用于将来的基于相同列名的图形。如果你以后创建了一个图形并且选择了来自于图形按钮弹出式菜单的来自模板功能，来自这个模板的版面信息就会被使用。

另外你还可以打印图形或者通过按主工具栏中的相应按钮把它复制到剪贴板。

## 14. 工程

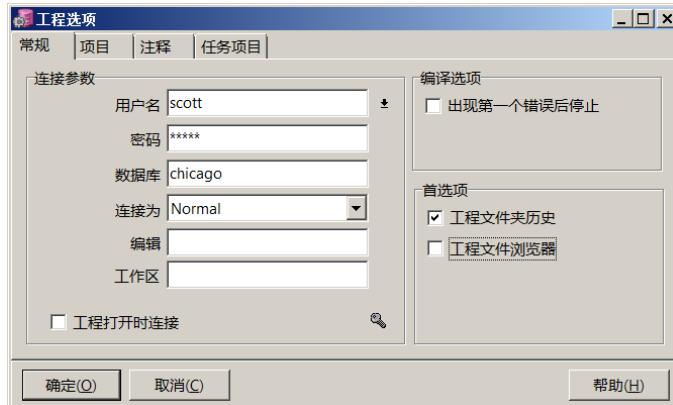
要组织你的工作，你可以使用 PL/SQL Developer 内置的工程概念。一个工程是由源文件、数据库对象、注释和选项的集合构成的，它允许你工作在一组特定项目的范围内，以代替整个数据库或方案。这会使你容易地查找你需要的工程项目，或者容易地从一个地方或数据库移动工程到另外一处。

工程明了你的桌面配置，如果你重新打开一个工程，所有的过去在工程被关闭以前被打开过的项目都将在同样的位置被打开。*自动保存桌面首选项*必须为此激活。这个信息没有被储存在工程定义文件（工程.prj）中，而在工程桌面文件（工程.dsk）中。

**注：**要用工程来工作，*使用工程首选项*必须被激活。如果不是这样，主菜单将不包括工程项目。对于所有的工程菜单项目，也有相应的工具栏按钮，可通过右键单击工具栏并启用*工程组*来添加。

### 14.1 创建一个新工程

要创建一个新工程，请选择来自于*工程*菜单的新建项目，这将调入工程选项对话框：



在一般页上，你可以定义工程被打开时你要链接到何处，你还可以定义编译错误出现后你是否还要继续编译工程。在首选项部分，可以控制在打开和保存文件时工程是否有其自己的文件夹历史或其自己的文件浏览器（第 19 章）。禁用时将使用常用的文件夹历史和文件浏览器。

项目页允许你查看工程项目，这可以是 PL/SQL 程序单元的源文件、测试脚本、SQL 脚本、报告等。工程的项目还可以是被储存在数据库中的对象，这些对象在源文件没有陈述。工程项目也能包含其他文件，例如 MS Word 文档、HTML 文档等。当工程第一次被创建时，项目列表将是空的。

注释页允许你维护工程注释，这些注释能够包含任何东西，例如任务列表、工程决定、设计等。

任务项目标签页显示了工程源文件的代码项目。关于任务项目的更多信息，请参见第 15 章。

## 14.2 保存工程

设置了选项以后，你可以通过来自于工程菜单的保存项目来保存工程，这将创建一个带有扩展名 .prj 的工程文件。工程文件包括项目说明、选项和注释。因此，当你更改选项时或者当你更改注释时，无论你何时添加或删除项目，你都必须保存工程。

保存工程将不会隐含地保存包含在工程中的源文件。

## 14.3 添加文件到工程

有两种方法添加文件到你的工程：

如果文件已经在 IDE 中打开，你可以简单地选择来自于工程菜单的添加到工程项目。对于程序文件（包含 PL/SQL 程序单元的源），项目隐式地为编译而被激活。这意味着当你生成或建立工程时，这个文件将被包括到编译进程中。其他文件，例如测试脚本、SQL 脚本、报告文件等，将不会为编译而被激活。

如果文件没有在 IDE 中打开，你可以按工程项目工具栏中的添加文件按钮，并且选择要添加的文件。

注意，如果可能的话，文件的路径被储存为相对于工程文件的位置。例如，如果文件被定位于与工程文件相同的目录，路径将不被保存。如果文件在别处，整个路径将被保存。这使工程产生容易地从一个目录移动到另外一个目录的“便携式”效果。

## 14.4 添加数据库对象到工程

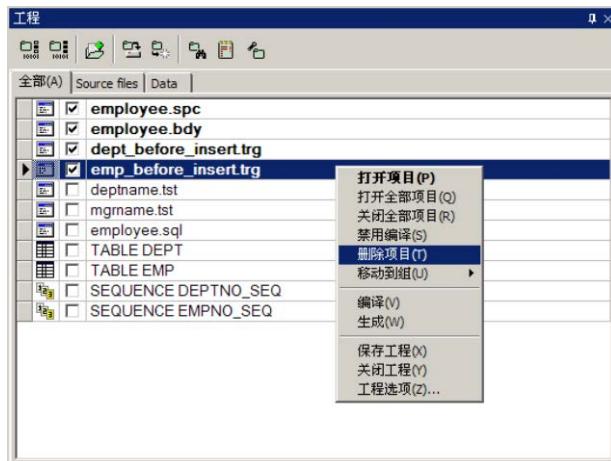
要添加数据库对象到工程，就要在浏览器中右键单击它，从弹出式菜单选择添加到工程项目。正象程序文件一样，PL/SQL 程序单元将隐式地为编译而被激活。其他对象，例如表和序列，则不能被编译。

你也可以从对象浏览器拖放一个数据库对象到工程项目窗口（请参见下面）。

如果当前用户拥有数据库对象，则所有者将不被保存到工程文件中。如果另外一个用户拥有这些对象，则所有者将被保存为对象说明的一部分。这使工程从一个用户移动到另一个用户变得很轻松。

## 14.5 用工程项目工作

在添加了项目到工程以后，你可以选择来自于工程菜单的工程项目来用工程的项目工作。工程项目窗口可以如下所示：



最初的四个程序文件允许被编译，两个测试脚本、SQL 脚本以及表和序列不允许被编译。

你可以保持工程项目窗口一直是打开的，以便快速和容易地访问所有工程项目。这个窗口是工程桌面的一部分，当工程被重新打开时它也将隐式地被重新打开。

要编辑一个项目，你简单地双击它就可以。如果项目是一个源文件，它将在一个相应的窗口（程序窗口、测试窗口等）中被打开。如果项目是一个数据库对象，相应的带有对象定义的编辑器将被打开。如果项目是外部文件，相应的应用程序将被运行（比如用于 PDF 文件的 Acrobat Reader）。

你也可以右键单击一个项目来调入像上面显示的弹出式菜单，在这中你可以禁用/允许编译项目，或者你可以从工程中删除项目。如果项目是数据库对象，你还可以使用作为子菜单的对象弹出式菜单。

要更改工程项目的顺序，你可以单击行标题把它拖到一个新的位置。这个顺序不仅仅在表面上重要，它还决定了编译的顺序。

要在工程项目的内 容中搜索文本，可以按下工程项目工具栏中的在项目中查找按钮，或使用工程菜单中的查找菜单项。

### 项目组

可以使用项目组为大型工程组织工程项目。可以为受限生成和构造操作使用这些组，以限制搜索，打开/关闭所有项目，并获得更好的整体概览。项目仅可属于一个组，此外还属于一个名为全部的特殊组。

要在项目组之间切换，可以选择项目列表上方对应的标签页。最初，仅有一个名为全部的组。这是包含所有项目的特殊组。要添加组，在组标签页上右键单击并从弹出式菜单中选择相应菜单项即可。

要将一个项目移入一个组，右键单击项目，选择 **移动到组**，再从子菜单中选择组即可。如果选择无，项目将仅属于**全部**组。在添加新工程项目时，项目将隐式地移入当前选定的组。

## 14.6 编译工程

要编译一个工程，你可以选择来自于工程菜单的建立或生成项目。

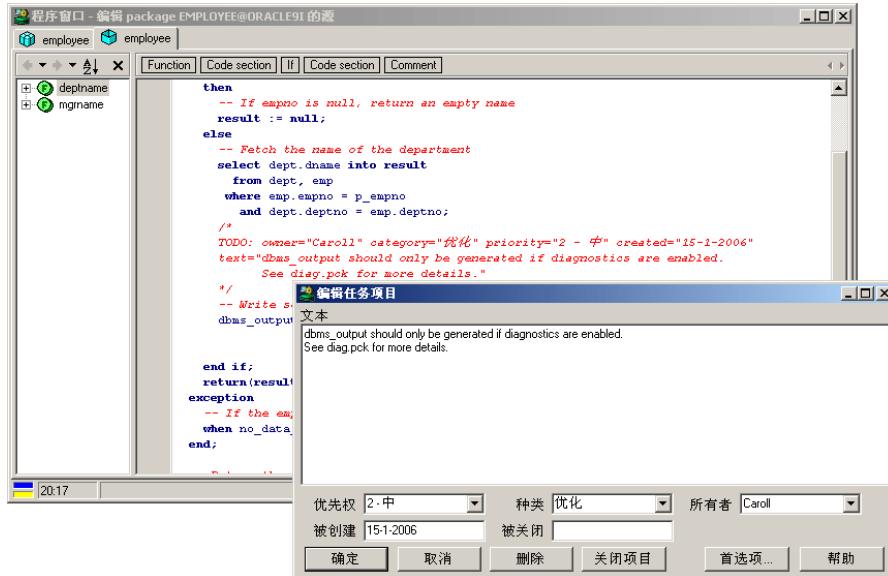
建立功能将简单地编译当前选定组中所有允许编译的工程项目，项目将按显示在工程项目列表中的顺序编译。如果一个项目编译失败，工程选项出现第一个错误后停止决定了工程编译是否应该继续进行：



生成功能只编译那些自上次编译以来已经被更改的项目。如果 PL/SQL 程序单元作为一个数据库对象被包括，并且它在数据库中是无效的，那么它也将被重新编译。编译的时间戳被储存在对象桌面文件中。允许编译但是最新的项目，将在编译进程窗口中被列出为被跳过。

## 15. 任务项目

你可以在任何源文件中使用任务项目以产生一个简要的注释，它记载了在这个源文件中需要做的一些事情，你以后可以从任务列表中访问这个信息。这些任务项目被放到了源文件中，作为一种在适当位置的带有特定格式的注释：



在这个例子中，你可以看到源文件中的注释（开始于文本 `TODO:`）和任务项目编辑器，在这个编辑器中你可以输入文本、优先权、种类、所有者、创建日期和关闭日期。对于优先权、种类和所有者输入框，选择列表的目录能通过首选项（请参见第 20.24 节）来定义。

任务项目能通过任务项目编辑器来访问，在这个编辑器中你可以方便地创建、编辑、查看或删除任务项目。只要不更改格式，你还可以直接在源文件中编辑注释。

因为任务项目在源文件中被作为注释来存储，所以它们将始终与源的实际情况一致，不管源是否来自于数据库、来自于文件系统或者来自于一个版本控制系统的旧版本。

要访问指定源文件的所有任务项目，你可以调用任务列表：



在这里，你可以快速地查找一个任务项目并通过按转到按钮在源文件中定位它。

任务列表可以通过单击列标题来排序。

右键单击列表可以复制、打印或者导出任务项目。

## 15.1 创建任务项目

要创建一个任务项目，你首先必须把光标放到源文件中适当的位置上。这个位置是注释将要被插入的位置，是接近你要工作的地方，所以这件事是重要的。其次你可以右键单击这个位置，选择来自于弹出式菜单任务项目子菜单的添加项目：



在上面提到的任务项目编辑器将显示出来，你可以为新的任务项目输入信息。在按下了确定按钮之后，注释将被准确地插入到光标位置。

## 15.2 编辑任务项目

要编辑现有的任务项目，你既可以右键单击源文件中相应的注释并且选择来自于任务项目子菜单的编辑项目，又可以选择显示列表项目（如果你当前不接近你要编辑的任务项目的话）。那样你可以从列表中选择任务项目并按编辑按钮或双击它。

现在，你可以更改任务项目的各种属性并按确定按钮来更新源文件中的注释了。

## 15.3 关闭任务项目

通过右键单击源文件中的注释并选择来自于任务项目子菜单的关闭项目，任务项目就能被关闭。你也可以调用任务项目编辑器并按关闭按钮。

## 15.4 删除任务项目

通过右键单击源文件中的注释并选择来自于任务项目子菜单的删除项目，任务项目就能被删除。你也可以调用任务项目编辑器并按删除按钮。

你还可以简单地从源文件中删除注释。

## 16. 列表链接

列表链接会显示出你曾经定义过或者最近使用过的链接。你可以在这个列表中设定主链接，为一个链接创建视窗，查看最近的链接的视窗。你还可以创建操作文件夹和子文件夹来对所有的链接进行层次上的归类。

### 16.1 列表链接使用说明

默认情况下，列表链接在左上角区域



在链接列表的顶部，有用来更新列表的按钮，有用来扩展和关闭树状图的按钮，和用来定义操作文件夹和链接的按钮（详见16.2）

链接列表里的第一个选项是用户定义的文件夹。这些文件夹可以保存其内部明确定义的链接，和/或保存那些根据文件夹标准暗中添加的链接。最近的文件夹里可以显示出你最近使用的所有链接。

如果链接是暗中添加的话，它的颜色就是灰色的；如果链接是被明确添加的话，它就可以显示为客户自定义的颜色。你可以通过这些链接，查看相关数据库会话的视窗，并对它进行操作。

每个链接后面的图标表示链接的等级：

链接的口令是已知的，并且这个链接有一个或多个数据库会话。

链接的口令是已知的，并且这个链接没有数据库会话。

链接的口令是未知的，当你尝试链接时会自动弹出口令框。

## 链接功能

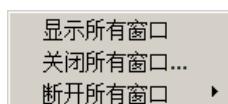
右击链接会弹出以下菜单：



- 前六个选项让你可以为选定的链接打开视窗。比如，你想要使用一个特定的链接来运行一个报告，右击该链接，点击报告菜单选项，从子菜单中选择一个报告。新实例项目将会针对选定连接打开一个新的 PL/SQL Developer 实例。
- 显示所有视窗选项会显示出所有属于选定链接的视窗，而且可以将这些视窗最小化。
- 关闭所有视窗选项可以关闭所有与选定链接相关的视窗。
- 重新链接所有窗口项允许选择一个链接，将当前链接的所有窗口重新指定给该链接。
- 链接选项可以创建一个到数据库的链接。如果口令是未知的，系统很快会提示你。
- 作为主要链接选项会将选定的链接作为主链接。你还可以通过双击该链接将它变成主链接。
- 断开选项会注销所有与选定链接相关的会话。这些会话包括链接的视窗。
- 删除选项只适用于暗中创建的链接。它可以让你清除你不常使用的链接文件夹。
- 添加到文件夹项允许将隐式创建的链接添加到链接列表。新链接将添加到链接列表的根下，随后可以设置其属性将将其移动到恰当的文件夹。

## 文件夹功能

右击一个文件夹会弹出以下菜单：



- 显示所有视窗选项会显示出所有属于选定文件夹内链接的视窗，而且可以将这些视窗最小化。

- 关闭所有视窗选项会关闭所有属于选定文件夹内链接的视窗。
- 重新链接所有窗口项允许选择一个链接，将当前链接的所有窗口重新指定给该链接。

### 视窗功能

右击视窗会弹出以下菜单



通过这个弹出的菜单，你可以激活，关闭，和调整视窗到正常或者最小化状态。

通过从重新连接子菜单里选择链接，你可以改变这个视窗的链接。

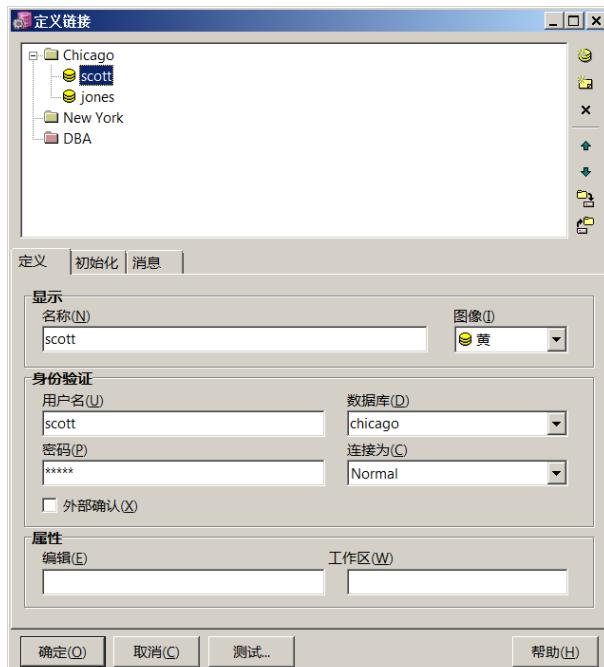
你还可以重命名一个视窗。在你创建一个新的视窗并想要为它命名时，这个选项是很有作用的。

短文本选项会将选项转换为更简洁的描述。比如，Program Window - employee.pck 会缩短为 P - employee.pck。

要执行一个窗口，你可以选择执行菜单里的选项。

## 16.2 定义文件夹和链接

想要定义你的文件夹和链接，你可以选择工具菜单中的定义链接项，或点击位于链接列表上方的定义链接按钮。以下对话窗口就会出现

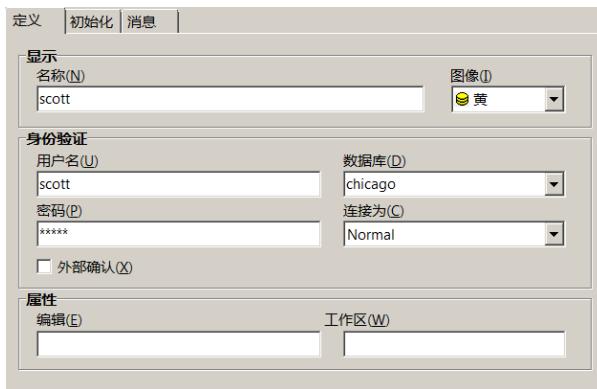


你可以通过右边的按钮创建一个新的链接，或者创建一个新的文件夹，或者删掉选定的链接或文件夹，或者上上下下移动它。你还可以通过拖动或拉拽这些选项来移动它们。

最后两个按钮可用于导出和导入所有链接定义。

## 定义文件夹

定义文件夹，点击新文件夹按钮，并输入如下信息。例如：



显示部分的名称和图像将显示在链接列表中。

用户名，数据库，和链接字段可用于隐式显示文件夹中的链接。在上面的例子中，所有的 chicago 数据库都会隐式显示在 Chicago 文件夹中。你还可以在用户名和数据库字段中使用\*和？通配符。用户名中的 sys\* 可隐式显示以 sys 开头的用户名的所有链接。如果你留下了空字段名，链接就需要被明确地添加到文件夹中。

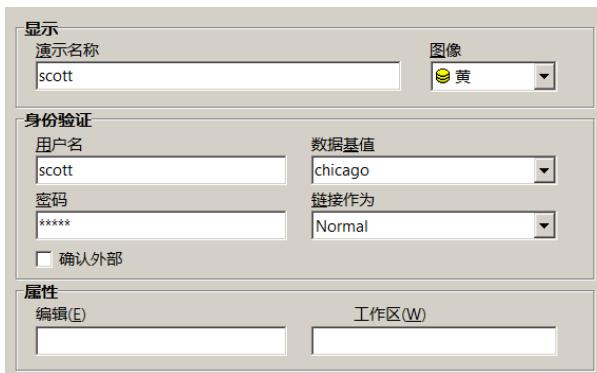
版本和工作区属性可用于显示给定版本和工作区的链接。

新建文件夹将在选定文件夹下创建。要在链接列表的根下创建文件夹，可以按住 Alt 键，并单击新建文件夹按钮。

要复制现有文件夹，可以选定文件夹，按住 Ctrl 键，同时按新建文件夹按钮。

## 定义链接

定义链接，点击新建链接按钮，输入以下信息。例如：



显示部分的名称和图像将显示在链接列表中。

用户名，数据库，口令，链接到和指定到外部字段可以为链接 创建登录信息。如果你没有输入口令，系统会提示你第一次使用 PL/SQL Developer 链接时的密码。

如果你正在使用 **基于版本的重新定义**，**版本**属性可用于设置版本。只要使用此链接，就会为会话设置给定版本。请注意，此属性区分大小写。

如果你正在使用 **工作区管理**，可以使用工作区属性设置工作区。只要使用此链接，就会调用 **DBMS\_WM.GotoWorkspace** 为会话设置给定工作区。请注意，此属性区分大小写。

新建链接将在选定文件夹下创建。要在链接列表的根目录下创建链接，请按住 Alt 键，并按下**新建链接**按钮。

要复制现有链接，可选中链接，按住 Ctrl 键，并按下**新建链接**按钮。

在**初始化**标签页中，可以输入针对此连接创建的每个会话将要执行的 SQL 脚本。用分号分隔多条语句，用斜杠终止 PL/SQL 程序块。

在**消息**标签页中，可以输入登录此连接时将要显示的消息。

定义连接及其初始化脚本后，可以按**测试**按钮来验证所有属性是否正确。

## 17. 链接、窗口、数据库会话和事务

PL/SQL Developer 提供了多会话/多线程环境。

多链接意味着，你可以在同一个 PL/SQL Developer IDE 实例中， 使用不同的账户链接到不同的数据库。所有的窗口都是为不同的链接创建的，而且窗口还会在左边底部的状态栏中显示这个链接。你可以单击窗口中的链接，然后从列表中选择一个新的链接来改变与这个窗口相关的链接。

多会话意味着，即使相同的链接，各个数据库会话也可用于测试窗口、SQL 窗口和命令窗口。此外，程序单元编译也将出现在各个数据库会话中。记住，编译是 DDL 语句并隐式地提交当前的事务，如果与其他窗口共享同样的会话，这有可能产生有害的结果。

多线程意味着，测试窗口、SQL 窗口和命令窗口的运行可以同时运行，不会互相妨碍。它的优势是，当长的连续的 SQL 语句正在运行时你可以做其他的工作，你可以中断或关掉会话，测试你的程序单元的锁定行为，等。

### 17.1 主链接

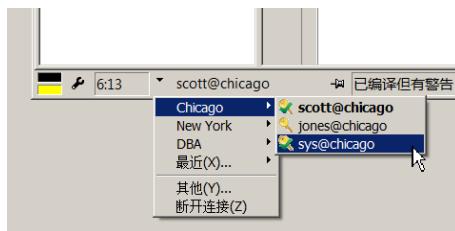
主链接是在你没有明确选择链接的时候，用来实现所有 PL/SQL Developer 功能的链接。当你打开一个新的 SQL 窗口并执行查询的时候，实际上就暗中为数据库会话创建了一个主链接。使用对象浏览器也需要主链接。

你登陆 PL/SQL Developer 就已经暗中设定了一个主链接。你可以点击数据库会话里的登陆或者双击链接列表里的链接 来改变主链接。主链接在链接列表中会以粗体显示出来。

你还可以使用会话菜单中的 **设定主链接** 或 **设置到<当前>** 的主连接项设置主链接。第一项允许从子菜单中选择链接，第二项将使用当前窗口的链接。

### 17.2 链接窗口

所有的窗口都是为不同的链接创建的，而且窗口还会在左边底部的状态栏中显示这个链接。从主菜单打开新窗口后，将使用主链接。你可以点击这个链接，然后在链接列表中选择一个新的链接，这样，这个窗口的链接就改变了。



窗口链接可固定或取消固定。在取消固定时 (F)，窗口链接将在主链接发生更改时与主链接保持一致。固定 (F) 时，窗口链接将在主链接发生更改时保持不变。要查看或更改窗口链接的固定状态，请查看或单击链接状态窗格中的按钮。

新建窗口的默认设置可能是固定的，也可能是取消固定的，具体取决于相应首选项（参见第 20.1 节）。如果显式更改了取消固定的窗口链接，则该链接将转为固定。

如果从弹出菜单中选择 **断开** 项目，数据库会话将会断开，窗口将不再与连接相关联。

## 17.3 会话模式

多线程意味着你需要一个多个会话的环境。记住，对于任何特定的数据库会话，Oracle 服务器每次只能运行一个语句。如果某些事情运行在不同的会话中，那么他们只能同时运行。所有的开发环境都不允许有不限制数量的会话，所以就需要设置首选项（请参见第 20.2 节）以控制 PL/SQL Developer 将要使用的会话的数量。这个首选项被调用到会话模式，可以有下列的值之一：

- 多路会话 - 每个测试窗口、SQL 窗口和命令窗口都将有它自己的会话，另外的一个会话将被用于编译。这是最灵活的设置，明显地会导致最大数量的数据库会话。另外可能的缺点是，在更新被提交之后，它们只在 X 窗口中可以看到而在 Y 窗口中看不到。
- 双路会话 - 属于同一个链接的测试窗口、SQL 窗口和命令窗口将共享一个会话，另一个会话将被用于编译。这个模式的缺点是每次只有一个特定链接的窗口可以运行程序。
- 单路会话 - 共享同一个链接的所有窗口和所有编译都使用同一个会话，这使事务管理变得很困难。在这个模式中调试器被禁用使用。如果你被限制只能使用一个数据库会话，那么你只能使用这个设置了。

## 17.4 运行于多路会话或双路会话模式

当你在多路会话和双路会话模式中运行测试窗口、SQL 窗口和命令窗口时，工具栏中的中断按钮将被激活，这样你可以在任何时间中断运行。中断并不总是成功的，例如在锁定情况下就是这样。这样，如果你有权限，就可以采用会话菜单中的关掉项目。

如果你移动当前运行窗口的鼠标光标，它的状态就会更改，以显示它正在运行并且你不能对那个窗口的内容做任何修改。

当运行完成之后，如果事务已经开始，提交和回滚按钮就会被激活。在多路会话模式中，每个窗口有它自己的会话，所以也有它自己的事务，提交和回滚按钮将反映出最上面窗口事务的状态。

## 18. 浏览对象

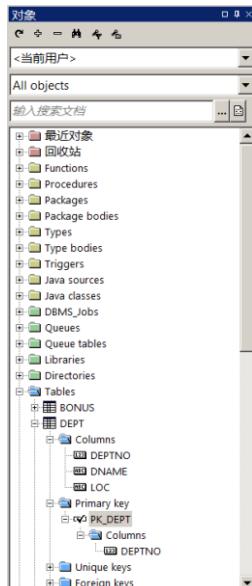
当你在 Oracle 数据库中正在开发一个程序单元时，这个程序单元总是会访问其他的数据库对象。记住，如果这不是实际情况，那么这个程序单元根本就没有在这个数据库中的权利。要写这样一个程序单元，你必须确切地知道这些其他数据库对象是怎样被定义的。因此，下面列出的问题对于 Oracle 程序员来说将是很熟悉的：

- 这个列的数据类型是什么？
- 这个表的主键是什么？
- 那个被打包的函数的准确名称是什么？
- 这个参数的默认值是什么？
- 这个触发器的触发条件是什么？
- 哪个对象调用了我正在用的函数？
- 等等...

通过使用对象浏览器，所有这些信息在 Oracle 词典中都是可利用的，都可以从 PL/SQL Developer 内部访问到。

### 18.1 使用对象浏览器

通过选择工具菜单中的对象浏览器项目，你能获得可用的对象浏览器。对象浏览器显示出了数据库中所有的与开发程序单元有关的基本对象。你可以操纵这个树形视图，这与你使用 Windows 的 Explorer 来操纵文件夹的方法一样。打开表文件夹就显示出了可以通过当前会话访问的所有表，打开一个指定表的文件夹就显示出了与这个表有关的所有属性，打开表下面的 Foreign keys 文件夹就显示出了这个表的所有外键，等。最近对象文件夹包括所有的在 PL/SQL Developer 中最近用过的对象。



无论你何时打开文件夹，从字典中查询的信息都被读进了内存，以使后来的访问尽可能地快。在某些事情被更改后，要强制对象浏览器重新查询这个信息，你必须选择已修改的对象并按在窗口左上边的刷新按钮。你还可以通过从弹出式菜单选择相应项目来刷新对象。要刷新整个浏览器，在按住刷新按钮同时请按 Ctrl 键。

你可以把对象从对象浏览器拖放到工作空间中，这将调出带有对象定义的编辑器。你可以通过设置首选项来准确地控制什么功能应该被执行（请参见第 20.11 节）。

你可以把对象从对象浏览器拖放到编辑器中，这将调出一个弹出式菜单，它的内容依赖于这个对象。你可以选择复制名称（如果必要的话通过所有者加上前缀）、属性、描述和其他有关文本到编辑器中你要放的地方。如果你拖放一个文件夹到一个编辑器中，文件夹中所有的对象名都将被复制，这些对象名都将用逗号分隔。你可以设置一个首选项来控制对象名是放在一行上还是多行上（请参见第 20.12 节）。

通过在对象图标上按右鼠标按钮，一个上下文敏感的弹出式菜单就显示出来了，它允许你执行这个对象上的其他功能。这些来自于弹出式菜单的功能在下面的段落中有所描述。

在选择期间，通过使用 Ctrl 或 Shift 键，你能选择多个对象。如果多个对象被选择，对象的弹出式菜单将只显示那些对所有已选对象

可应用的项目，并且拖放功能将同时工作在所有的对象上。

你可以按 **Ctrl -** 和 **Ctrl +** 按钮依据内容快速地调整对象浏览器的宽度。

对象浏览器的首选项（请参见第 20.11 节）允许你定义哪些文件夹是可见的、以什么顺序显示。它还允许你定义在最近对象文件夹中能看到的对象的最大数量。此外，你还能定义对浏览器对象双击的行为。在默认的情况下，这将扩展相应的节点，你还可以定义这样将调出对象定义编辑器或属性窗口。

### 过滤对象

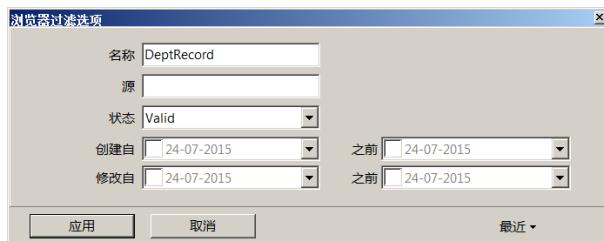
要限制对象浏览器中的对象数量，可以使用对象浏览器顶端的用户选择器和过滤器选择器。还可以通过在过滤器字段中输入表达式来执行特定过滤。

在选择用户时，仅显示该用户拥有的对象。选择 **<当前用户>** 仅查看由当前链接的 Oracle 用户拥有的对象。选择 **<所有用户>** 可查看任意用户拥有的对象。请注意，在打开文件夹时，这可能会降低性能。

选择过滤器，将对象限制为满足特定标准的那部分对象。过滤器也定义了文件夹内的对象排序次序。关于如何创建自己的过滤器，请参见第 18.2 章。

除了这些过滤器定义之外，也可以在过滤器字段输入特定表达式。最基本的过滤器是按对象名称，它可以无需额外语法直接指定。例如，如果在过滤器字段输入 **DEPT**，对象浏览器会将对象名称限制为 **DEPT**。可以通过添加 **%** 通配符进一步改进过滤器。例如，如果在过滤器字段输入 **DEPT%**，那么将显示对象名称以 **DEPT** 打头的所有对象。

要获得更复杂的过滤器表达式，可以按靠近过滤器字段的 [...] 按钮，调用 **浏览器过滤器选项** 表单：



- **名称** – 名称中包含此文本的对象。可以添加 **%** 通配符来改进过滤器。过滤器表示式为 **Name=<text>**。
- **源** – 包含 PL/SQL 源中所出现文本的对象。没有 PL/SQL 源的对象（例如表、视图、序列）也将隐式包含。过滤器表达式为 **Source=<text>**。
- **状态** – 此状态（有效或无效）的对象。过滤器表达式为 **Status=<valid/invalid>**。
- **创建自** – 自此日期之后创建的对象。过滤器表达式为 **CreatedAfter=<date>**。
- **之前创建** – 自此日期之前创建的对象。过滤器表达式为 **CreatedBefore=<date>**。
- **修改自** – 自此日期之后修改的对象。过滤器表达式为 **ModifiedAfter=<date>**。
- **之前修改** – 自此日期之前修改的对象。过滤器表达式为 **ModifiedBefore=<date>**。

对于上述例子，对象浏览器将会包含在其 PL/SQL 源代码中有单词 **DeptRecord** 的有效对象。在表单中输入过滤器选项并单击 **应用** 按钮后，对象浏览器将会更新，对象浏览器上的过滤器字段

将会包含过滤器表达式。在此例中：*Source=DeptRecord Status=Valid*。后续可以直接在过滤器字段中修改表达式，也可以间接通过浏览器选项表单进行修改。

## 创建新对象

要创建一个新的程序单元、表、视图、序列或同义词，请右键单击这样一个对象或这个对象类型的根目录，从弹出式菜单选择新建项目。

使用程序单元的新建功能将创建新的程序窗口，初始内容来自于指定的模板，如第 26 章所述。视图能被创建于 SQL 窗口，初始内容来自于指定的模板，就象程序单元一样。

表、序列和同义词将被显示在指定的编辑器窗口中。表、序列和同义词编辑器在第 9 章中有详细描述。

## 复制对象

通过从弹出式菜单选择 **复制** 项目，你可以复制表、序列或同义词。这将调出一个带有已选对象所有属性的编辑器窗口，除了名称以外。如果你要创建一个相似的表来执行某些测试，这就可能是有用的。

## 复制对象或文件夹名

要复制一个对象名或者一个文件夹中的对象名，请选择逗号分隔 **复制** 项目，这样做将把名称放到剪贴板上。你可以设置首选项来控制对象名是放在一行上还是放到多行上（请参见第 20.12 节）。

## 查看对象属性

要查看指定对象的属性，请从弹出式菜单选择 **属性** 项目。你看到的实际属性依赖于对象类型和你的 Oracle 服务器版本。

EMP_MGR_FK 的属性	
属性	值
OWNER	SCOTT
CONSTRAINT_NAME	EMP_MGR_FK
CONSTRAINT_TYPE	R
TABLE_NAME	EMP
R_OWNER	SCOTT
R_CONSTRAINT_NAME	PK_EMP
DELETE_RULE	NO ACTION
STATUS	ENABLED
DEFERRABLE	NOT DEFERRABLE
DEFERRED	IMMEDIATE
VALIDATED	VALIDATED
GENERATED	USER NAME
LAST_CHANGE	17-10-2005 12:10:57

属性被显示在你能放到屏幕上任何地方的模式窗口中。你可以保持属性窗口是可见的（它将停留在其他窗口的顶部）并且继续在另一个窗口中进行编辑。你还可以通过按窗口右上方的按钮来卷起和放开窗口。这样，你能容易地有空间做其他事情，并保持属性是可用的。

## 查看对象描述

通过从弹出式菜单选择 **描述** 项目，表、视图、函数、过程、包、类型和方法是能被“描述”的，这在某种程度上类似于我们熟悉的 SQL\*Plus。就象前面提到的属性窗口一样，描述也被显示在一个模式窗口中。

对于表和视图，所有的列和它们相关的属性都被显示在属性窗口中。对于函数、过程和方法，参数和它们的属性会被显示。对于包，它们的函数、过程、变量、类型、常数和异常会被显示。对于类型，方法和属性会被显示。

名称	类型	可为空	默认	注释
EMPNO	NUMBER(4)			The 4-digit number of the employee
ENAME	VARCHAR2(10)	Y		The last name of the employee
JOB	VARCHAR2(9)	Y		The job of the employee
MGR	NUMBER(4)	Y		The number of the manager
HIREDATE	DATE	Y		The date on which the employee was hired
SAL	NUMBER(7,2)	Y		The salary of the employee
COMM	NUMBER(7,2)	Y		The commission of the employee
DEPTNO	NUMBER(2)	Y		The 2-digit number of the department

在描述窗口中，你可以选择行、列或一定范围的单元格。要选择行或列，只单击行标题或列标题再拖动鼠标指针使选择区高亮显示就可以。要选择一个特定范围的单元格，移动鼠标指针越过一个单元格的左边直到光标形态更改，再按鼠标按钮，拖动鼠标直到要选择的区域高亮显示。选择的内容能被复制或打印。要产生一个以逗号分界的已选单元格的拷贝，请右键单击窗口从弹出式菜单选择相应项目。要导出描述到 CSV、TSV、HTML 或 XML 文件，请选择来自于导出子菜单的相应项目。要打印出描述窗口的内容，请选择 **打印** 项目。

通过在描述窗口中双击一个单元格，单元格的内容就会被复制到最上面窗口的编辑器中。

要在描述窗口中对行进行排序，请按你要对行进行排序的那一列的标题按钮。

## 查看对象定义

所有的对象类型都有定义，要查看这个定义，你可以从弹出式菜单选择 **查看项目**。对于所有的程序单元、视图、表、序列和同义词，这个项目都是可利用的。对于表的元素如列或索引，如果你选择了 **查看项目**，它将显示出表的定义并导向已选择的元素。

程序单元将显示在程序窗口中，视图将显示在 SQL 窗口中，表、序列和同义词将显示在指定的编辑器窗口中。表、序列和同义词编辑器在第 9 章中有详细描述。

## 编辑对象定义

通过从弹出式菜单选择 **编辑** 项目，你可以编辑一个直接来自于数据库的对象定义。你可以通过第 20.2 节描述的首选项来禁用这个特性。

## 重新命名对象

通过从弹出式菜单选择 **重新命名** 项目，表、视图、序列和同义词能被重新命名。如果你使用的 Oracle 9.2 或更新版本，你还能重新命名列。带有旧名称的对话框将显示出来，在这中你可以修改和应用它。

## 删掉对象

有时候删掉一个对象是必要的，它可能已经没有用了，或者必须从一种类型更改为另一种类型（例如从函数改到过程）。要实现这个，只要从弹出式菜单选择 **删掉** 项目即可。

## 浏览相关对象

大多数对象既有参照又有被参照的对象，通过选择 **浏览** 项目你就能快速地在对象浏览器中定位相关的对象。

## 重新编译对象

对象经常参照数据库中其他对象。无论这些被参照的对象何时更改，参照对象都可能变得无效。在这种情况下，一个红色标志被显示在这个对象的图标中。要编译无效对象，请从弹出式菜单选择 **重新编译项目**。如果编译成功，那个标志就不显示了。如果编译失败，你就可以通过从弹出式菜单选择 **查看项目** 来查看错误。

## 添加对象源到编辑器

通过选择 **添加源到编辑器** 项目，程序单元的源文件能被添加到程序窗口或测试窗口中。

对于程序窗口，源将被添加到当前程序单元后面的一个新页上。这样你可以创建一个包括几个已经存在于数据库中的程序单元的程序文件。

对于测试窗口，在调试会话期间，源将被添加到当前已选择的源的后面。这样你可以在测试窗口中包括源，随后在实际单步进入程序单元以前设置或删除断点。

## 添加调试信息到对象

在调试会话期间，如果你要查看或设置变量的值，包含这些变量的对象需要包含调试信息。通过选择 **添加调试信息** 项目，你可以手工添加调试信息。这个菜单项目有一个检验标志，它能显示出对象是否已经包含了调试信息。调试器的首选项部分包含了一个选项，当程序单元被编译时，能自动地添加调试信息，这样你只需为你没有开发的程序单元使用这个项目。

## 查看和设置 PL/SQL 条件编译标志

使用 **PL/SQL CCFlags** 菜单项可查看或设置一个程序单元的条件编译标志：



在这个屏幕上，可以看到程序单元的当前标志。还可更改标志的值、重命名标志、添加新标志、删除现有标志。按下应用按钮即可将这些更改应用到数据库对象。请注意，这将导致程序单元的隐式重新编译，并可能致使其无效。

## 重新编译参照对象

当你更改一个被其他程序单元或视图参照的对象时，这些参照对象将变为无效。重新编译它们将使它们变为有效或者会发现一些不相容的问题，通过选择 **重新编译参照对象** 项目，这会很容易地被实现。这将调出会立即被执行的编译无效对象窗口。关于这个工具的更多信息在第 21.3 节能够找到。

## 测试程序单元

要测试一个程序单元（孤立的函数或过程、被打包的函数或过程、或者对象类型方法），你可以使用测试脚本，其详细资料在第 5 章中有描述。一个测试脚本一般将包含一个简单的函数或过程调用。在运行测试脚本以后，你需要设置输入参数，查看输出参数和函数结果。你可以从对象浏览器中选择**测试项目**来创建这样一个测试脚本。

## 查询表和视图

要查看表或视图中的数据，请从弹出式菜单选择**查询数据**项目。对于在对象浏览器中当前已选的表或视图，一个带有 select 语句的 SQL 窗口就被创建了。如果你把**自动运行 SQL 窗口**首选项设为允许，select 语句将立即被运行。

## 编辑表数据

如果你要编辑一个表中的数据，当表在对象浏览器中被选择时，你可以选择**编辑数据**项目。一个带有包括 rowid 的 select 语句的窗口将被创建。如果你把**自动运行 SQL 窗口**首选项设为允许，select 语句将立即被运行。在记录被获取之后，SQL 窗口立即进入编辑模式。

## 导出表

要导出表的定义和数据，请选择**导出数据**项目，这将为已选表调出**导出表**工具。详细资料请参见第 21.4 节。

## 允许和禁用触发器和约束

通过从弹出式菜单选择**允许或禁止**项目，触发器和约束能从对象浏览器内部被允许或禁用。被禁用的触发器和约束被显示为灰色。

## 运行作业

对于通过 dbms\_job 包被提交的作业，通过从弹出式菜单选择**运行**项目，你可以明确地运行它。

## 允许和禁止进入队列和脱离队列

对于队列，通过选择来自于弹出式菜单相应的项目，你可以允许和禁止进入队列和脱离队列操作。在菜单项目前面的检验标志指示了操作在当前是否被允许。

## 添加对象到自定义文件夹

要添加对象到自定义文件夹（请参见第 18.3 节），请选择**添加到文件夹**项目，随后选择一个来自于子菜单的文件夹。

## 恢复或清理对象

如果你使用的是 Oracle10g 或更新版本，你可以恢复或清理你前面删掉的表，前提是如果这个特性在数据库中被允许的话。对于来自于**回收站**文件夹中的对象，你可以选择**清理**来永远地删除掉表和它的数据，或者你可以选择**恢复**来在原来的名称下或者在一个新的名称下恢复它。

## 18.2 对象浏览器过滤器

你可以定义过滤器以决定哪些对象在对象浏览器的系统文件夹中是可见的、这些对象按什么顺序显示。位于对象浏览器顶部的选择列表允许你快速地在不同的过滤器之间切换。

通过选择工具菜单中的**对象浏览器过滤器**项目或者通过按下对象浏览器上方的过滤器按钮，对象浏览器过滤器就能被定义。你可以在这个对话框中修改、创建和删除过滤器：



过滤器可以被整理出下列属性：

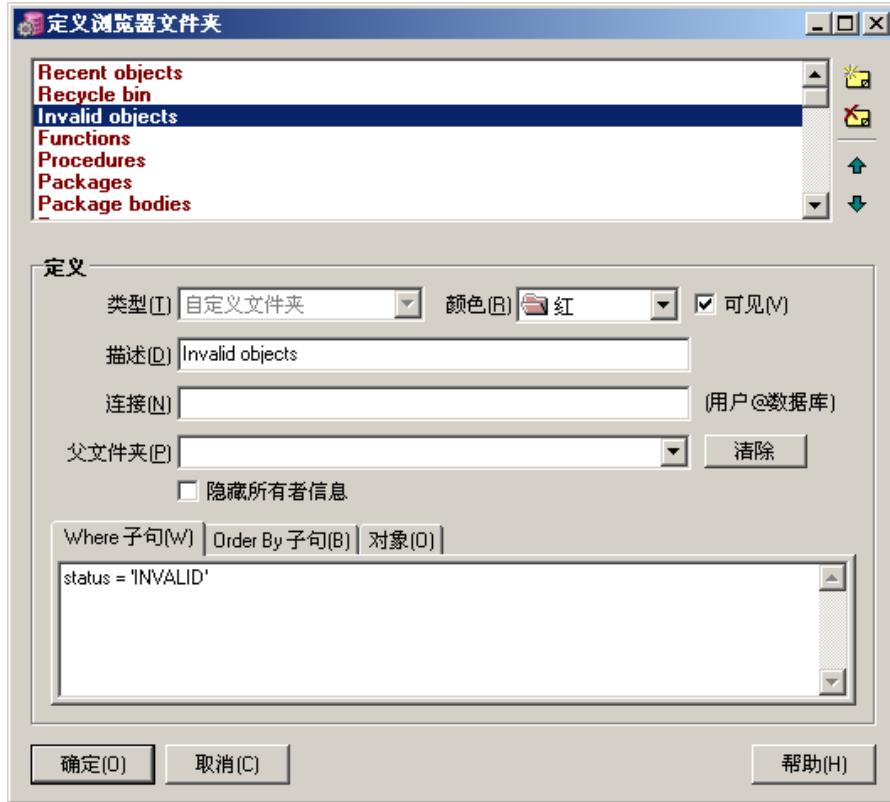
- **描述**  
这个描述在对象浏览器顶部的选择列表中显示。
- **用户**  
定义哪一个 Oracle 用户使用这个过滤器是有效的。如果你留此输入框为空白，那么过滤器对所有的 Oracle 用户都是有效的。如果你指定一个用户和数据库（例如 scott@chicago），那么过滤器只在你登录为这个用户和数据库时是可见的。
- **默认**  
如果你检验了这个选项，当你开始运行 PL/SQL Developer 时，已选的过滤器将被激活。你可以为每个用户都定义一个过滤器，默认的过滤器都以粗体字显示。
- **Where 子句**  
过滤器在 *all\_objects* (或 *dba\_objects*) 视图上定义了一个 select 语句。Where 子句可以用于限制 select 语句的结果集。你可以使用下列的列：
  - owner - 对象的所有者
  - object\_name - 对象名称
  - object\_type - 对象类型 (TABLE、VIEW、PACKAGE 等)
  - status - 对象状态 (VALID 或 INVALID)
  - created - 创建对象的日期/时间
  - last\_ddl\_time - 最近被编译对象的日期/时间
  - Order by 子句。  
使用来自于 *all\_objects* 视图的列来对浏览器中的对象进行排序。

## 18.3 对象浏览器文件夹

对象浏览器的文件夹能用几种方法定制。你可以更改对象浏览器中文件夹的顺序、每个文件夹的颜色、文件夹的层次，并且你可以基于你当前的链接隐藏指定的文件夹。

你还可以定义你自己的文件夹，并带有对象把它们移民。你既可以明确地添加指定的文件夹到一个目录，又可以定义一个查询把它移民，又可以合并二者。当文件夹在浏览器中被打开时，查询将被执行。

通过选择工具菜单中的对象浏览器文件夹项目或者通过按下对象浏览器上面的文件夹按钮，对象浏览器过滤器就能被定义。你可以在这个对话框中修改、创建和删除文件夹：



在这个对话框的上部，你可以看到所有的文件夹和它们的层次。红颜色指示了系统文件夹，黑颜色指示了自定义的文件夹。一些限制应用到了系统文件夹中。

在文件夹列表的右边，有四个可用的按钮，你可以用它们在当前已选文件夹的位置来添加一个新的文件夹、删除一个文件夹、在列表中上移或下移。对象浏览器中的排序与这个列表中的排序是一样的。

在下部，你可以更改当前已选文件夹的定义：

- **类型** - 说明文件夹的类型：自定义或系统。

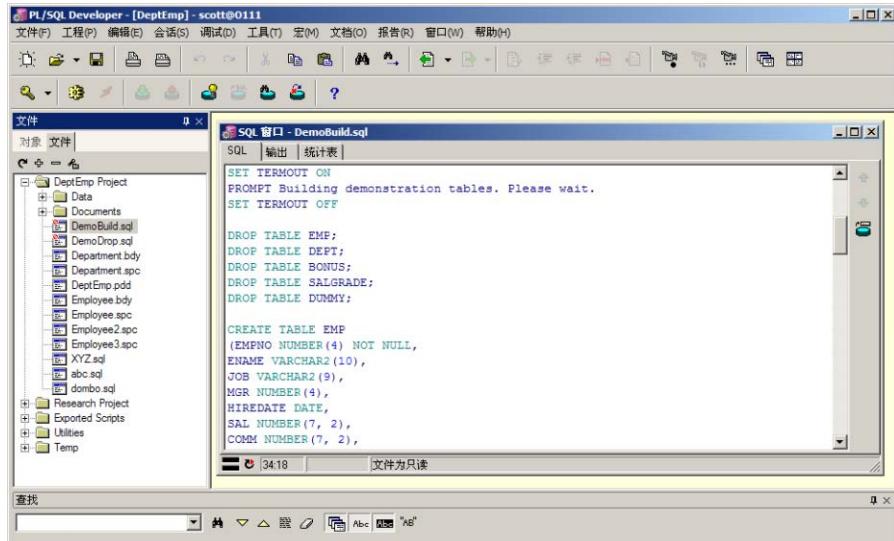
- **颜色** - 选择文件夹的颜色，这个文件夹中的对象子文件夹将继承同样的颜色。
- **可见** - 禁用这个选项使对象浏览器的文件夹为不可见，这可以用于隐藏指定的系统文件夹，并且越过链接选项。
- **描述** - 对象浏览器中文件夹的描述，不能为系统文件夹而被更改。
- **链接** - 定义哪一个 Oracle 用户的文件夹将显示。如果你保留这个选项为空，则所有 Oracle 用户的文件夹都会被显示。如果你指定一个用户和数据库（例如 scott@chicago），文件夹只在你登录为这个用户和数据库时是可见的。你可以使用 ? 和 \* 通配符来匹配多用户。例如，\*@prd 为 prd 数据库中的所有用户。
- **父文件夹** - 要按层次放置一个文件夹到另外一个文件夹内部，你可以从列表中选择它的父文件夹。要清除父文件夹，请按清除按钮。
- 隐藏所有者—启用该功能时，对象浏览器将不会显示文件的项目所有者。这个功能主要被用于文件夹所有者的身份可以从文件内容中得知的时候（比如：石康的桌面视图）
- **Where 子句** - Where 子句可以被用于基于文件夹的查询以限制对象的选择。在上面的例子中，Where 子句被设置为 status = ‘INVALID’，这样就把带有所有无效对象的文件夹的内容移民了。
- **Order By 子句** - Order by 子句可以被用于基于文件夹的查询以定义已选对象的排序。
- **对象** - 对于手工建立的文件夹，你可以定义对象列表。你可以按所有者. 名称格式在每行放置一个对象，例如： SCOTT.EMP。注意，通过右键单击并从弹出式菜单选择添加到文件夹，添加一个来自于对象浏览器的单独的对象，这通常是非常方便的。

Where 子句和 order by 子句可以包括任何 SQL 表达式和来自于 *all\_objects* 视图的任何列，这和浏览器过滤器的工作方法一样，请参见第 18.2 节以获得更多信息。

注意，过滤器不影响自定义文件夹的对象的内容。

## 19. 浏览文件

如果你频繁访问一组有限的位置上的源文件和脚本（例如，工程目录、工具目录等），可以使用文件浏览器来迅速、轻松地进行访问。你不必打开文件选择器来打开或保存文件，从 PL/SQL Developer 的工作区即可直接访问文件：



通过文件浏览器树，可以访问本地计算机或网络上用户定义位置的文件和目录。可以快速打开、保存、重命名和删除文件。

可以使用文件浏览器作为停放或浮动工具。关于可停放工具的更多信息，请参见第 28 章。默认情况下，文件浏览器位于对象浏览器停放区的第二个标签页。

## 19.1 设置文件浏览器

初次显示文件浏览器时，它将是空的。要填充文件浏览器，你必须定义一个或多个文件浏览器位置。为此，从工具菜单中选择文件浏览器位置，或按下文件浏览器顶端的位置按钮。此时将出现以下屏幕：



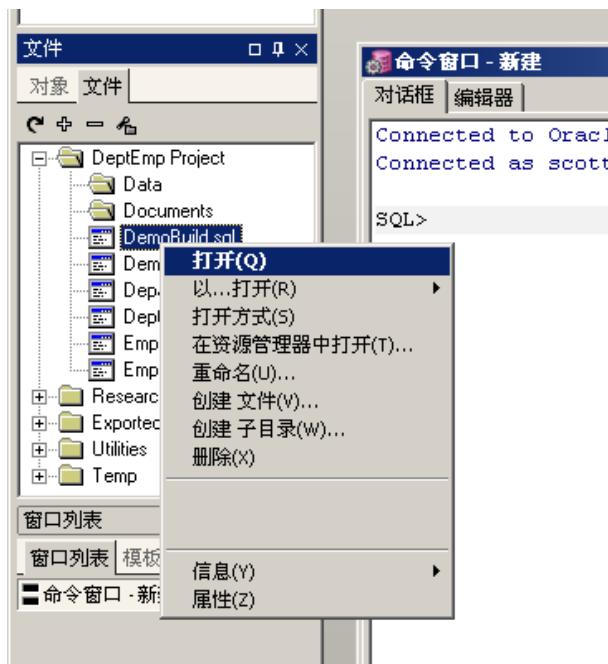
你定义的每个位置都将显示在文件浏览器树的根下。对于每个位置，都可定义如下属性：

- 路径** - 文件和目录所在的本地或网络路径。文件浏览器树将显示来自此起点的文件和目录。
- 包含子目录** - 在启用时，路径中的子目录将显示在文件浏览器树中，并可遍历。
- 包含文件过滤器** - 将显式包含在文件浏览器树中的一个或多个文件过滤器表达式。使用分号分隔多个过滤器表达式（例如，\*.sql; \*.cmd）。如果忽略此属性，则将包含所有文件。
- 排除文件过滤器** - 将显式在文件浏览器树中排除的一个或多个文件过滤器表达式。使用分号分隔多个过滤器表达式（例如，\*.sql; \*.cmd）。如果忽略此属性，则不排除任何文件。默认情况下，此属性将排除 files.info 和 \*.\*。首先排除包含文件状态信息的 PL/SQL Developer 系统文件，然后排除 PL/SQL Developer 备份文件。
- 说明** - 文件浏览器树中所显示位置的说明。
- 颜色** - 文件浏览器树中位置及其子目录的图标颜色。
- 主位置** - 此位置的主位置。在选中时，此位置将不会显示在文件浏览器树中，而是作为主位置的子位置。

使用列表中的相应按钮可以创建、删除和排序位置。

## 19.2 使用文件浏览器

定义了文件浏览器位置之后，文件浏览器将以树形视图显示你的所有文件、子目录和子位置。右键单击文件即可在工作区中以编辑模式打开该文件。右键单击文件将显示如下弹出式菜单：



- **打开** - 如果当前项目是文件，则将以编辑模式打开。如果当前文件是目录或位置，则将展开树形视图的节点。
- **打开为** - 显示一个子菜单，你可将当前文件打开为程序窗口、SQL 窗口或命令行窗口。
- **打开条件** - 打开带有程序的文件。
- 在资源管理器中打开 - 打开 Windows 资源管理器，搜索文件或者根目录。
- **重命名** - 重命名当前项目。
- **创建文件** - 在当前位置或目录创建新文件。
- **创建子目录** - 在当前位置或目录创建新子目录。
- **删除** - 删除当前文件或子目录。
- **窗口另存为“名称”** - 将当前（顶端）窗口的内容保存到当前选定目录或位置中。这将允许你快速在选定位置保存内容，无论其来源如何（新建、来自数据库或来自文件系统的其他位置）。
- **窗口另存为“文件名”** - 将当前（顶端）窗口的内容另存为当前选定文件。注意：这将替换当前文件的内容。

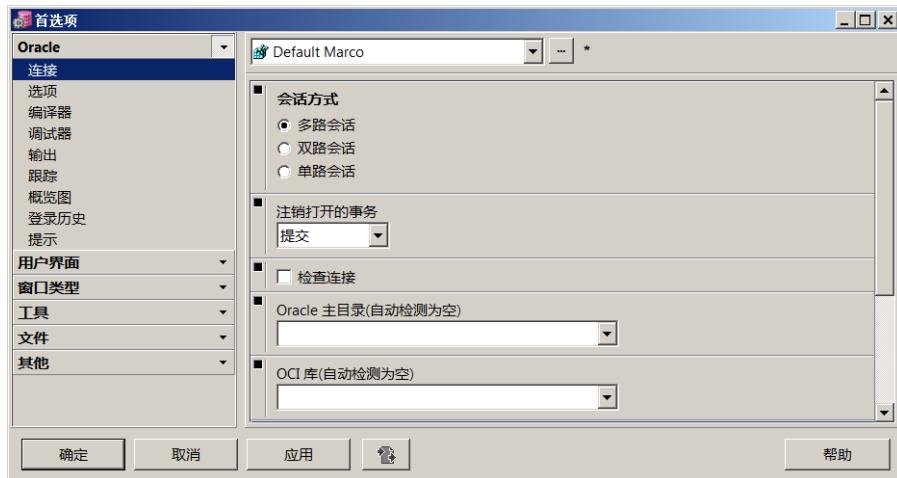
- **信息** – 这个子菜单显示当前文件的完整路径、时间戳和大小。

权限—显示文件或根目录的权限。你可以通过属性对话框设定属性和权限。只读文件会自带一个红色的标记（参见上例的 Department.spc 和 Employees.spc 文件）

如果使用 PL/SQL Developer 以外的其他工具创建、删除或重命名文件，文件浏览器将自动更新内容。

## 20. 首选项

首选项允许你为 PL/SQL Developer 设置不同的首选项。当你选择了这个项目，下列的对话框就显示出来了：



在左边，你可以从六个种类之一中选择一个首选项页（Oracle、用户界面、窗口类型、工具、文件和其他）。

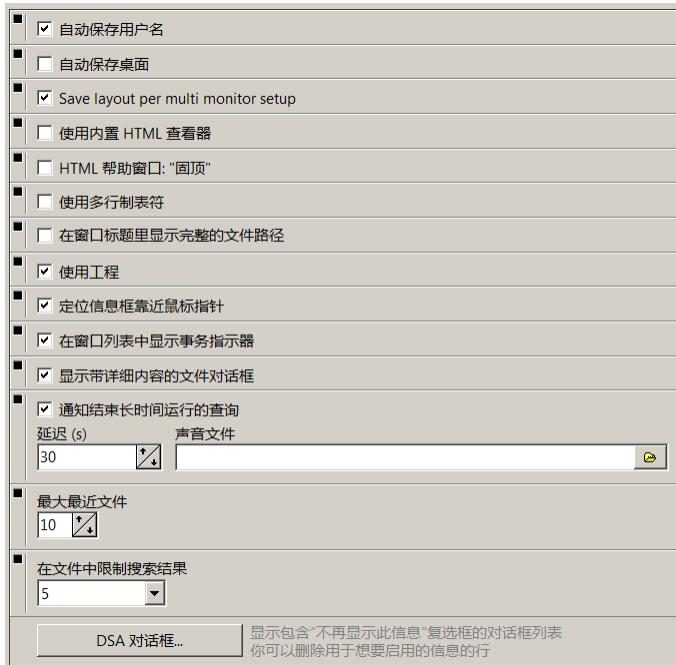
在顶部，你可以选择一个首选项集，它定义了某一层次的一组参数。在默认的情况下，你可以为你的 Windows 用户（如屏幕截图上方的用户 *Mark*）定义个人的首选项，你还可以为指定的 Oracle 链接定义个人的首选项。无论你何时使用这样的链接来登录数据库，相应的首选项集都会被使用。此外，你的系统管理员还可以在全局层级和 Oracle 链接层级定义默认的首选项。要了解关于首选项集的详细信息，请参见第 20.33 节。

在首选项集的下面，你可以看到当前已选的首选项页的首选项（屏幕截图上是 Oracle - 链接页）。对于每个首选项，你将看到当前的值。首选项左边的绿色指示器指示了值在当前首选项集中被设置了。如果指示器是黑色的，则值从不同的集被继承（把鼠标光标放在指示器上面来查看首选项集名）。要恢复首选项到它被继承的值（换句话说，从当前首选项集删除它），你可以单击绿色指示器。

在首选项表单的底部，可以按确定来确认更改，按取消来取消更改，按应用来预览针对可视首选项的更改，或者按导入/导出按钮来将首选项导出至文件或从某个文件导入首选项。

下列章节将详细地描述每个首选项页。

## 20.1 Oracle – 链接



- **会话模式**  
控制 PL/SQL Developer 将使用多少个单独的数据库会话。这个特性在第 17 章中有详细描述。
- **注销(有打开的事务)**  
当会话被注销并且这个会话还有打开的事务，默认的行为是提交那个事务。通过这个首选项，你可以定义这样打开的事务被回滚，或者询问你确认信息。
- **检查链接**  
当这个选项被激活时，如果你的数据库链接仍然是活动的，则 PL/SQL Developer 每 60 秒检查一次。如果你的会话被关掉，或者服务器被关闭，PL/SQL Developer 将自动地被注销。
- **Oracle 主目录**  
在默认情况下，PL/SQL Developer 将使用你的电脑中的主 Oracle 主目录。你可以定义带有 Oracle 主目录选择器（一个标准的 Oracle 实用工具）的主 Oracle 主目录。通过选择来自于这个列表中的一个，你可以强制 PL/SQL Developer 使用指定的 Oracle 主目录。
- **OCI 库**  
PL/SQL Developer 将使用被安装的最新版本的 SQL\*Net 或 Net8。如果这个版本的 SQL\*Net 或 Net8 引起了问题，你可以强制 PL/SQL Developer 使用另外的版本。DLL 的

名称就象 ora72.dll (SQL\*Net 2.2)、ora73.dll (SQL\*Net 2.3)、oci.dll (Net8 8.1.x) 等的样子。这些 DLL 被定位于 Oracle 的 bin 目录中。

- 支持多链接  
可启用时，PL/SQL Developer 允许在同一个实例中添加多个链接。禁用时，改变链接将会隐式断开其他所有的链接，主链接也会相应的改变。
- 默认固定窗口连接  
启用此项时，新打开的窗口的连接将默认固定。禁用此项时，新创建连接将默认取消固定，并在主连接发生更改时与其保持一致。另请参见第 17.2 节。
- 自动关闭未用连接。  
启用此项时，未用连接（未打开此连接的任意窗口）将自动关闭。禁用时，这些连接将保持打开，此时，在稍后重新使用这些连接时速度更快，但占用的服务器资源角度。
- 保存最近使用历史  
在启用的情况下，你退出 PL/SQL Developer 时，*链接列表*>*最近文件夹*里最近使用过的链接会被自动保存，重启 PL/SQL Developer 时，所有的连接会自动复原。在禁用的情况下，你每次打开 PL/SQL Developer，*最近文件夹*都会是空的。
- 密码保存  
在启用的情况下，最近使用链接历史会被输入密码后保存。

## 20.2 Oracle – 选项



- 允许编辑数据库源  
这个选项控制了程序单元或视图的源是否能从对象浏览器中被直接编辑，或者它需要从文件系统中打开它。
- 询问是否保存刚编辑过的数据库源  
如果你要编辑一个来自于被存储在数据库词典的源中的程序单元（通过右键单击程序单元并从弹出式菜单选择 *编辑*），你可以通过禁用这个选项来防止问你是否保存这些

修改到文件中。如果你已经编辑了前面从文件系统中被打开过的程序单元，那么你总是要被问是否保存这些更改。

- 允许编译只读源文件  
如果源文件为只读，当被调入到程序单元窗口中时，它不能被编译。当使用版本控制系统时，这个首选项是常用的，它经常把源文件的只读拷贝保留在工作目录中。这样的拷贝不必被编译，因为这会干扰其他对象成员的工作。
- 确认提交和回滚  
无论你何时通过按相应的按钮来提交或回滚一个事务，你都将被要求确认。禁用这个选项将立即提交或回滚，不需要确认。
- 若可用则使用 DBA 查看  
如果这个首选项被设置，PL/SQL Developer 将试图使用 DBA 视图(db\_objects、db\_triggers 等)以取代 ALL 视图(all\_objects、all\_triggers 等)。如果开发者具有某一系统权限，DBA 视图能比 ALL 视图揭示出更多的信息。
- 自动统计  
对于每个被执行的 SQL 语句和测试脚本，这个选项控制了是否生成统计报告。如果你禁用了这个选项，统计报告将是空的。通过按“选择...”按钮，一个带有你当前链接的数据库中的所有可用的统计项的列表就打开了。你可以选择你希望包括在统计报告中的哪些统计项。如果你没有访问到动态执行表，“选择...”按钮将被禁用。
- 支持 Unicode  
若选中此项，将从 Oracle Server 中按原样获取 Unicode 数据，并将其显示为 Unicode 文本。若禁用此项，则来自服务器的 Unicode 数据将根据 Oracle Home Registry 的 NLS\_LANG 键转换为 Oracle Client 的字符集。
- 检查客户端和服务器字符集的不匹配之处  
此选项将检查客户端字符集和服务器字符集是否兼容。例如，若你的服务器字符集为 UTF8，客户端字符集为 WE8MSWIN1252，则无法在 SQL 或 PL/SQL 代码中使用 UTF8 特有的字符。
- 为视图 DDL 添加列别名列表  
若启用此选项，在解压视图的 DDL 时将添加一个列别名列表。若禁用此选项，PL/SQL Developer 将尝试确定是否需要别名列表。
- 为调度程序对象使用时间戳格式。  
默认情况下，调度程序对象的时间戳属性将被视为常规日期。启用此选项将使用会话的 NLS 设置定义的时间戳格式。此时无法使用从日历中选取日期的功能。

## 20.3 Oracle – 编译器



此首选项页面允许你控制程序窗口的 PL/SQL 编译器的各种选项。关于这些选项的更多信息，请参见 *Oracle 数据库参考指南*。

- **PL/SQL 优化级别**  
指定用于编译 PL/SQL 程序单元的优化级别。此参数设置得越高，编译器在优化 PL/SQL 程序单元时所需的工作就越多。
- **PL/SQL 代码类型**  
若设置为解释，PL/SQL 程序单元将编译为 PL/SQL 字节码格式。此类模块将由 PL/SQL 解释器引擎执行。  
若设置为本地，PL/SQL 程序单元将编译为本地（机器）码。此类模块将本地执行，无需任何解释器负载。

- PL/Scope 标识符

此选项控制编译时 PL/SQL 源代码标识符数据的收集、交叉参照和存储。若设置为无，则将禁用标识符数据的收集。若设置为全部，则将收集所有源代码标识符数据。

- PL/SQL 条件编译标志

在这个部分中，可以为各编译定义默认条件标志。可以使用任何名称，可以使用整型值、布尔 True 或 False 值或 Null。关于条件编译的更多信息，请参见 *PL/SQL 语言参考指南*。如果将此部分保留为空，则将使用数据库实例的默认标志。按下复位到默认标志按钮即可使用数据库实例的默认标志。

- PL/SQL 警告

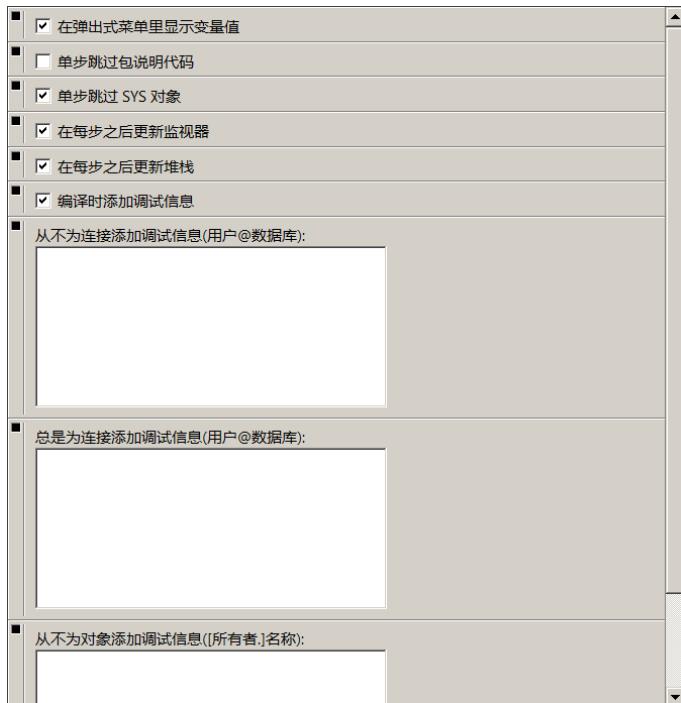
在这个部分中，可以定义编译器处理特定警告异常或特定严重性分类的方式。对于每个异常编号或分类，可以定义禁止（无警告信息）、作为警告处理（带有警告信息的成功编译）还是作为错误处理（带有错误信息的失败编译）。可以指定多个异常编号，使用逗号分隔。

请注意，警告的顺序是有意义的。在列表底部的项目优先于顶部的项目。因而，应首先为分类指定活动，随后为特定异常指定活动。

- PL/SQL 置换变量

如果程序文件包含置换变量，类似于 SQL\*Plus（带有 & 前缀）中所使用的置换变量，你可以在这里定义这些变量的值。请注意，置换变量仅在与程序文件一起使用时才有用，这是由于数据库中的结果源仅包含编译后的置换值。

## 20.4 Oracle – 调试器



- 在弹出式菜单中显示变量值

如果你禁用了这个首选项，当你在测试窗口中移动鼠标光标越过变量名时，变量值将不会自动地在弹出式菜单中显示。这样，在右键单击变量名之后，你必须从弹出式菜单中选择设置变量项目。

- 单步跳过包说明代码。

启用时，调试器不会显示包说明中执行的初始化步骤。

- 单步跳过 SYS 对象

如果你有权查看 SYS 对象的源，如果你禁用了这个选项，调试器就可以单步进入这些程序单元。通常，你不需要这样做，所以这个选项默认为禁用。

- 在每步之后更新监视器

如果你禁用了这个选项，通过按测试窗口的调试工具栏中的更新监视器按钮，你可以手工刷新监视器。这样做可以在比较慢的配置下使调试的执行快起来。

- 在每步之后更新堆栈

如果你禁用了这个选项，通过按测试窗口的调试工具栏中的更新调用堆栈按钮，你可以手工刷新调用堆栈。这样做可以在比较慢的配置下使调试的执行快起来。

- 编译时添加调试信息

当这个选项被允许时，PL/SQL Developer 中的每个编译都将自动地添加调试信息，所以在

调试会话期间，变量值总是能被查看和设置。如果你禁用了这个选项，你可以从浏览器中手工添加调试信息。

- 从不为链接添加调试信息

对于这个链接匹配的列表，调试信息将从不自动被添加，不管“编译时添加调试信息”首选项是否已选。如果你要确保你从不添加调试信息到产品数据库或某一用户，这个功能就可能是有用的。在这中，你可以使用常见的 `用户@数据库` 语法，也可以使用 \* 和 ? 通配符（例如 `sys@*` 即任何数据库的 sys 用户）。

- 总是为链接添加调试信息

对于这个链接匹配的列表，调试信息将总是自动被添加，不管“编译时添加调试信息”首选项是否已选。如果你要确保你总是添加调试信息到开发数据库或某一用户，则这个功能可能是有用的。在这中，你可以使用常见的 `用户@数据库` 语法，可以使用 \* 和 ? 通配符（例如 `*@devdb` 即 devdb 数据库的所有用户）。

- 从不为对象添加调试信息

对于这个对象列表，调试信息将从不被添加。当在特定的情形下添加调试信息时，一些版本的 Oracle 服务器能引起错误。你将在编译期间看到具有代表性的 ORA-00600、ORA-03113 或 PLS-00801 错误。如果这些问题出现了，你可以添加相应的对象到这个列表（所有者. 名称）中，直到 Oracle 提供了修补。

## 20.5 Oracle – 输出



- 允许

在 SQL 窗口和测试窗口中，在运行了脚本之后，调用 `dbms_output.put_line` 将被缓冲和显示。在每个单独的 SQL 窗口和测试窗口的输出页上，这个首选项可以被否决。

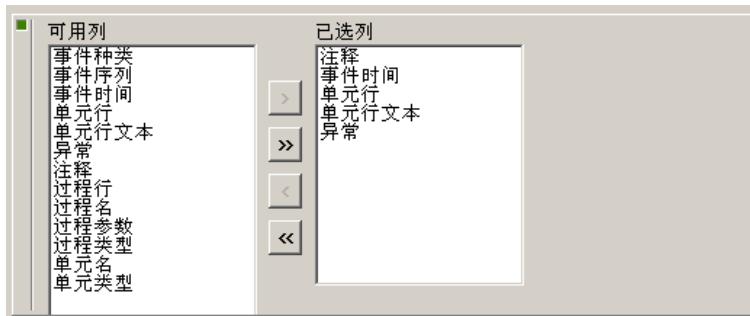
- 执行前清除

在运行 SQL 窗口或测试窗口之前，当前的输出页将被清除。这样，输出页将总是只包含最近运行的输出。

- 缓冲区大小

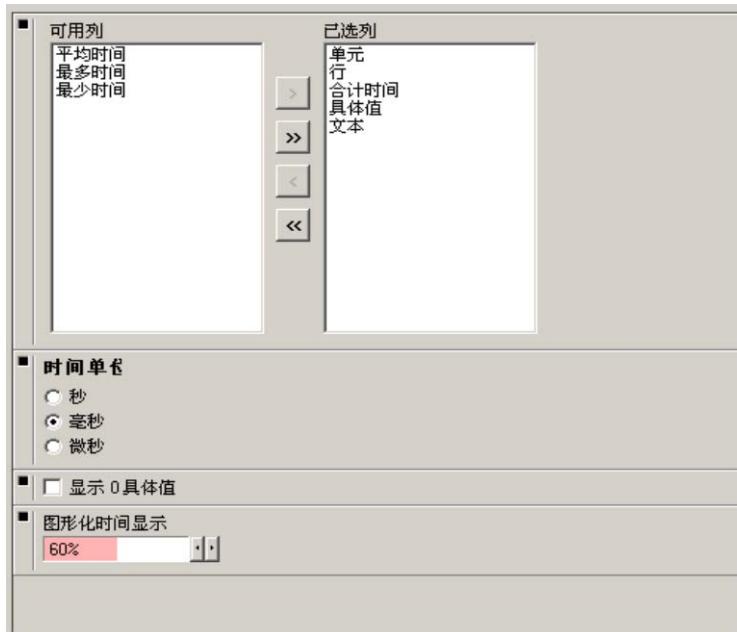
`Dbms_output` 缓冲区字节的大小。在一个运行期间如果多于这个数的字节被输出，异常将会出现。

## 20.6 Oracle – 跟踪



这些首选项控制了测试窗口的跟踪页（请参见第 5.11 节）。你可以控制来自于跟踪表的哪一列应该被显示、应该怎样排序。

## 20.7 Oracle – 概览图



这些首选项控制了测试窗口的概览图页（请参见第 6.3 节）。

- **列**  
控制来自于概览图表的哪些列应该被显示、怎样排序。

- 时间单位  
定义了合计时间、最大时间、最长时间和平均时间列的单位。
- 显示 0 具体值  
确定了被报告的源行是否被显示，这个源行带有执行了 0 次语句获取的具体值。
- 图形化时间显示  
合计时间列将显示出一个相关时间的图形化的表述。你可以控制这个条的颜色的亮度，或者使它不可见。

## 20.8 Oracle – 登录历史



如果你激活了存储历史首选项，PL/SQL Developer 将在登录历史中保存用户名、口令和数据库信息。当你以后重新开始运行 PL/SQL Developer 时，登录对话框允许你快速地选择以前用过的账号：



为了安全起见，你可以不要求在历史中储存口令，即使 PL/SQL Developer 把它们加密也要这样做。这意味着，如果某人要在你的电脑中访问 PL/SQL Developer，他们就不能使用你的账号链接到数据库。在这样的情况下，你应该禁用带口令存储选项。历史大小和显示大小首选项控制了多少个链接将被储存（最早的一个将从历史中被删除）、多少个将被显示在屏幕上。排序选项控制了在选择列表中链接应该怎样排序。

注意，即使你已经在登录历史中选择了储存口令，你仍然可以通过在登录期间按回车键或确定按钮时按 *Ctrl* 键来预防这件事。这样你可以防止有权限账号（例如 SYS 或 SYSTEM）的口令能被重新调用。

你还可以在固定用户部分定义许多固定的账号，可以带有口令也可以不带有口令。这些固定账号将始终被显示并且永远不会自动被删除。你还可以为这些固定用户通过添加 `>menuname` 来定义子菜单项目。在上面的例子中，创建了 *chicago* 和 *detroit* 的子菜单：

```
>chicago
scott/tiger@chicago
sys@chicago as sysdba
>detroit
scott/tiger@detroit
sys@detroit as sysdba
larry@detroit
```

在这种情况下，子菜单为数据库而被创建，但是你可以创建任何种你喜欢的种类。例如，开发、测试、产品。

如果保持被组合的未排序的固定用户选项被激活，固定用户将被显示在来自于历史的一个分开的部分中，并且不会被排序。如果固定用户一经使用就被添加到历史选项被激活，在登录期间选择固定用户也将添加到历史。

在历史部分中，你可以看到在 PL/SQL Developer 中哪些用户作为登录结果被添加到了历史。要删除指定的项目，请选择它并且按列表旁边的删除项目按钮，或删除所有项目按钮。

## 20.9 Oracle – 提示



### 编译器提示

你可以允许和禁用每个单独的提示，并且适当时控制提示的其他属性。每个提示和属性的解释都在提示列表的下面显示。

### 命名约定

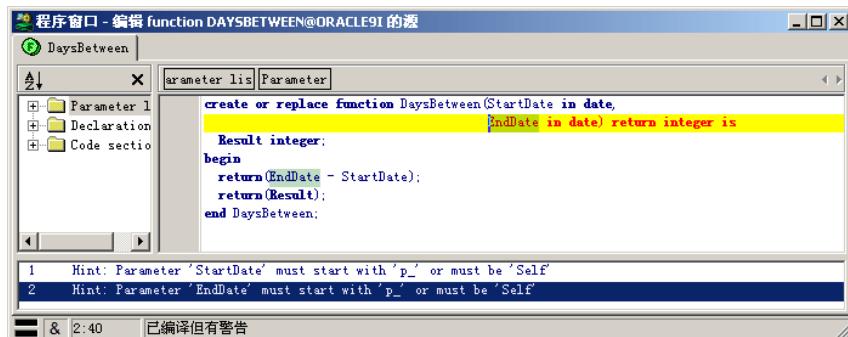
在这个列表中，你可以定义程序单元被编译时或者显示编译器提示功能被激活时将被选取的命名约定。对于不符合这些命名约定的程序单元源文件中的每个元素，都会显示出来提示。

对于每个命名约定，你都可以定义元素类型（参数、变量等）、元素必需的前缀、第一个和后续字符（在前缀后面）的可能值、必需的后缀、命名约定不合适时应该显示的提示的描述。

与上面的例子中显示的一样，描述也应该在表单中，实际的提示信息将为〈元素〉〈名称〉〈描述〉。例如：参数“StartDate”必须起始于“p\_”。

如果一个元素必须符合多个命名约定之一，那么你可以简单地为同样的元素类型添加多个行。所有行的描述都将显示，用“或”来分隔。对于上面的例子，参数提示信息将是：

参数 ‘StartDate’ 必须起始于 ‘p\_’ 或必须是 ‘Self’



## 20.10 用户界面 – 选项



- 自动保存用户名

这个选项可以是已检验，用来自动地保存每个 PL/SQL Developer 会话的用户名。当你以后重新启动 PL/SQL Developer 时，最近使用的用户名将被默认在登录对话框中。

- 自动保存桌面

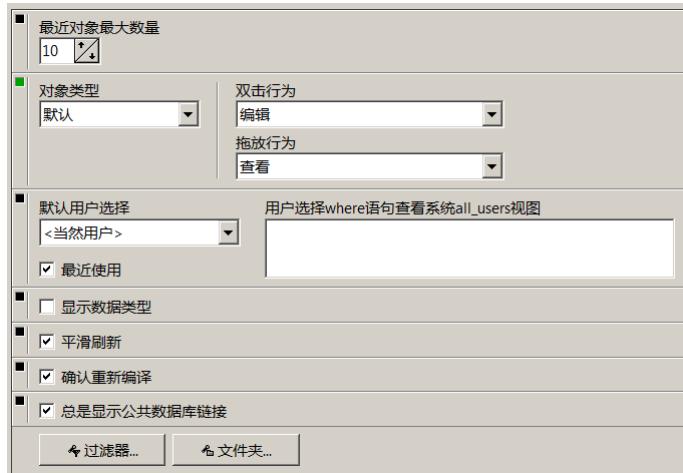
当这个选项被激活时，如果你重新启动 PL/SQL Developer，所有在 PL/SQL Developer 停止运行时被打开过的文件都会自动地被重新打开。当前打开的不与文件关联的命令窗口或解释计划窗口也将被恢复。此外，主窗口和对象浏览器的大小也将被恢复。

- 使用内置 HTML 浏览器

在默认的情况下，PL/SQL Developer 将使用内置的 HTML 查看器来显示 HTML 帮助文档。如果你有 Microsoft Internet Explorer 4 或更新版本，你可以禁用这个选项而使用 Microsoft 的 HTML 组件以代之。这将在弹出式菜单中给你一些额外的特性，并且将继承产生于 Internet Explorer 的首选项。

- HTML 帮助窗口：“固顶”  
如果这个选项被允许，HTML 帮助窗口将停留在顶部，直到你关闭它或者把它最小化。如果它被禁用，HTML 帮助窗口将作为一个单独的任务显示出来，它在 Windows 任务条中有自己的按钮。这样，无论你何时需要它，你都可以把 PL/SQL Developer 主窗口或帮助窗口带到前面。
- 使用多行制表符  
如果一个窗口包含了多个标签页，这个首选项控制了万一出现溢出时这些标签页是否可以被分为多个行。如果这个选项被禁用，万一出现溢出情况两个卷动按钮将显示出来。这当前只应用于程序窗口。
- 在窗口标题中显示完整的文件路径  
在默认的情况下，只有文件名才被显示在窗口标题中。如果这个选项被允许，完整的路径将被显示。
- 使用工程  
如果这个选项被允许，只有工程主菜单项目才是可见的。如果你没有正在使用的工程，就禁用它，这样它不会在主菜单中占用任何空间。要了解关于工程的更多资料，请参见第 14 章。
- 将消息框定位在鼠标指针附近。  
若启用，消息框的显示位置将在鼠标指针附近，使你能够快速使用按钮。若禁用，消息框将始终显示在屏幕中央。
- 显示文件对话框及细节视图。  
若启用，文件打开和保存对话框在打开时将始终显示细节视图。若禁用，则将使用默认视图样式。
- 通知长时间运行的查询的结束。  
如果 SQL 窗口或测试窗口中的查询占用了较长的时间，可以使用此首选项，在查询结束时得到通知。你可以控制发出通知之前的最小延时，也可以定义将播放的声音文件。对于占用的时间低于给定延时的查询，不播放任何声音。
- DSA 对话框  
只要你在信息框中选择了“不再显示这个信息”选项，一个条目就会被添加到你的首选项中的 DSA 部分。如果你随后要重新启用这个信息，请按 DSA 对话框按钮来编辑这部分，这将调出允许你删除相应行的文本编辑器。
- 最大化最近文件  
通过文件菜单和工程菜单中的重新打开子菜单，可以控制最近文件的最大数量。
- 限制文件中的搜索结果  
限制在搜索工具栏中搜索文件时显示的搜索结果的最大数量（参见第 22.20 节）。如果在搜索结果中在文件上单击，将会继续显示所有结果。

## 20.11 用户界面 – 对象浏览器



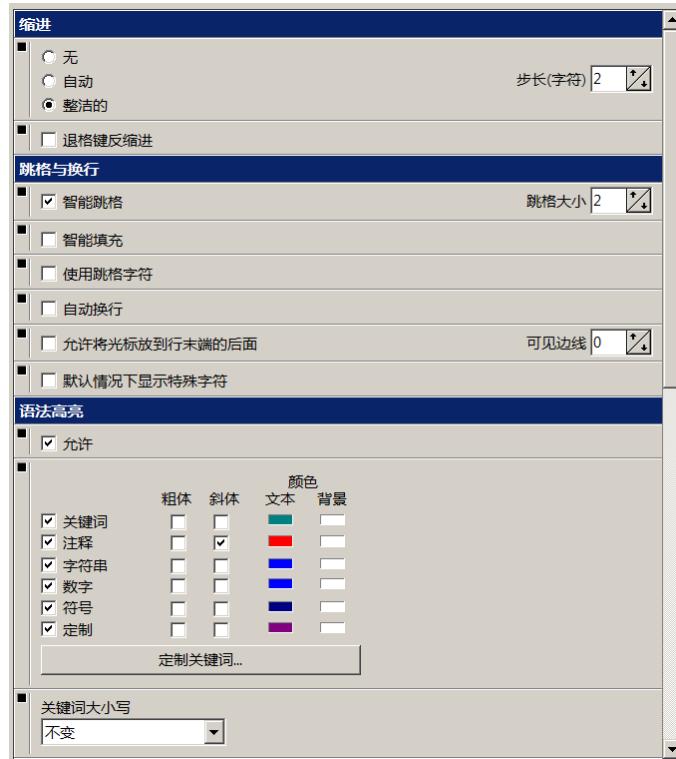
这些首选项控制了对象浏览器的外观和行为（请参见第 18 章）。

- 最近对象最大数量  
控制在最近对象文件夹中多少个对象是可见的。当多于这个数字的对象将要被显示时，最近最少用的对象将被删除。
- 双击和拖放行为  
控制了当你双击一个对象浏览器中的对象或者你把它从对象浏览器拖放到工作空间中时哪种行为将被执行。你可以指定一个默认行为，并且为来自列表的每个对象类型覆盖或者继承这个默认行为。
- 默认的用户选择  
用来控制<当前用户>和<全部用户>是否选择默认的选项。如果选中了最近使用选项，则在主连接发生更改时，将为连接使用最近使用过的用户选择。
- 用户选择 where 子句  
让你可以通过 where 子句来查询。可对 all\_user 视图执行查询，你可以在该视图中访问所有的纵列。
- 显示数据类型  
当这个选项被允许时，列的数据类型（表或视图）、属性（对象）和参数（函数和过程）都将被显示在对象浏览器中。
- 平滑刷新  
在一些系统上，这个选项可以使对象浏览器刷新操作慢下来（例如更改对象浏览器过滤器、登录为不同的用户等）。如果这是你应该禁用的情况，当对象浏览器被刷新时，这可能就会导致一些屏幕闪烁。
- 确认重新编译  
当这个首选项被允许时，在浏览器中重新编译程序单元将以成功或失败信息为结果。当它被禁用时，你将只能在编译失败时看到信息。

- 总是显示公共数据库链接。  
启用时，对象浏览器始终显示用户可访问的所有数据库链接。禁用时，仅显示选定用户拥有的数据库链接。

过滤器（请参见第 18.2 节）按钮和文件夹（请参见第 18.3 节）按钮将调用相应的对话框。

## 20.12 用户界面 – 编辑器



### 缩进

- 不缩进。  
在按下回车键时，编辑器不会缩进，并始终转向下一行的第一个字符。
- 自动  
在按下回车键后，编辑器将自动缩进到与前一行相同的缩进位置。
- 智能。  
与自动缩进相同，但若前一行开始了一个新块（例如，DECLARE、BEGIN、FOR、LOOP 等），此时要插入额外的缩进。这些块字可通过 PL/SQL Developer 安装目录中的 SmartIndent.txt 文件配置。

- 步长 (字符)  
当缩进或撤消缩进一个选择之后编辑器将缩进的字符数。
- 退格键减少缩进量  
若启用此选项，按下退格键会将缩进量减少到之前恰当的缩进位置。若禁用，按下退格键将始终删除一个字符。

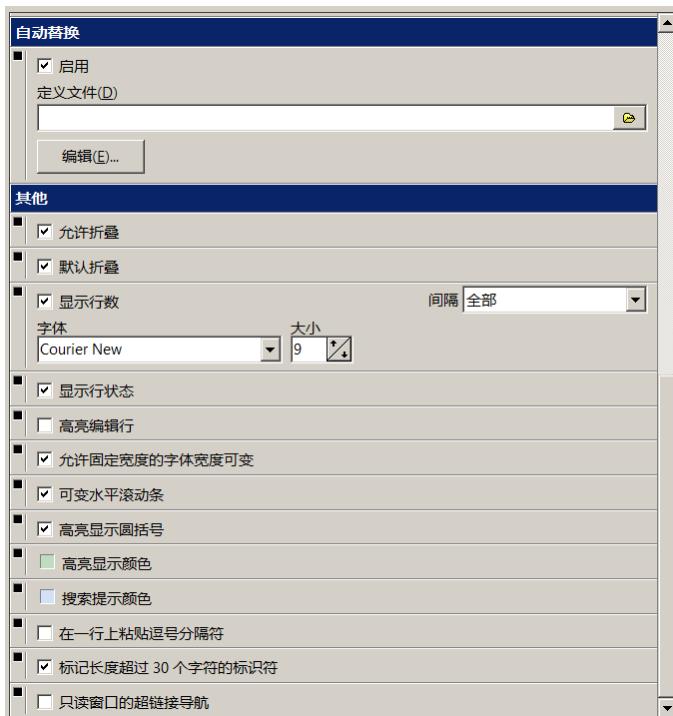
## 跳格与换行

- 智能跳格  
将导致编辑器跳格到前面行相关的位置。
- 跳格大小  
确定一个跳格将使用的字符数。
- 智能填充  
如果可能编辑器将用跳格替换空格。
- 使用跳格字符  
当被禁用时，只有空格才能用于跳格和缩进。
- 自动换行  
比窗口宽度还宽的行将在文本行上被延续。
- 允许将光标放到行末端的后面  
当被允许时，你可以将光标放到行末端的后面。当你在这中插入文本时，行将首先用空格来填补。
- 可见边线  
将在被指示的位置显示垂直线。 0 = 禁用。
- 默认显示特殊字符。  
若启用此选项，将显示空格、制表符和回车等特殊字符。若禁用，则必须通过*编辑*菜单显式启用。

## 语法高亮

这组首选项控制了用于编辑器中的语法高亮。你可以允许或禁用语法高亮，并且定义关键字、注释、字符串、数字和符号 (:= 、 => 、 || 、等) 的字体和颜色。对于关键字，你还可以控制它们是否应该被转换到大写、小写或以大写字母写。你还可以定义定制的语法高亮、定义应该被高亮的词。按*定制关键字*按钮将调出允许你定义这些词的编辑器。

关键字大小写首选项控制键入关键字时的格式：大写、小写或首字母大写。



## 自动替换

若启用此选项，可以定义自动替换为替换文本的词。例如，如果你定义了 *wo* 这个词的替换文本为 “*when others then*”，那么只需键入 *wo* 即可插入这段文本。

按下 **编辑键** 将允许你编辑包含词和替换文本的文件。在此文件的每一行中，都可输入 **词=替换文本** 说明。例如：

```
wo=when others then  
rs=rollback to savepoint
```

还可以参照一个模板作为替换文本。例如：

```
wh=Loops\While loop.tpl
```

在本例中，*While loop* 模板子目录将被调用，允许为替换文本使用变量和光标位置。

**定义文件** 字段允许你指定为自动替换定义使用哪个文件。

## 其他

- 允许折叠。

若启用此选项，程序窗口的 PL/SQL 编辑器将允许你折叠和展开代码部分，如第 22.15 节所述。

- 默认允许折叠。  
指定在打开或创建程序窗口时，代码折叠是否默认打开。
- 显示行数  
选中这个选项会使行数在左边空白处变为可见。*间隔*属性控制所显示的各行号之间的间隔。使用字体和大小可更改行号的外观。
- 显示行状态。  
若启用此选项，编辑器将在有更改的行边缘处显示黄色标记，新行边缘处显示绿色标记。
- 高亮编辑行  
控制了文本光标所在位置的行是否高亮显示。
- 允许为固定宽度的字体使用可变宽度。  
此选项允许你使用固定宽度的字体正确显示双倍宽度的字符（例如，日文）。还可用于在粗体样式下比普通样式宽的固定宽度字体。如果你正在使用这样的字体，而且看到了被截断的字符，则可以使用这个选项。
- 可变水平滚动条  
启用时，水平滚动条将基于编辑器的可见文本。禁用时，水平滚动条将基于整个编辑器的内容，包括目前不可见的文本。
- 圆括号高亮  
当被允许时，如果光标在打开或关闭圆括号处，编辑器将加亮圆括号对。
- 高亮颜色  
决定了诸如括号、变量等所有高亮的颜色。
- 搜索提示颜色  
决定了被搜索栏找到的搜索提示的颜色。
- 粘贴逗号分隔项目于一行上  
控制了在对象浏览器中或在描述窗口中被选择的多个项目被粘贴到编辑器中的一行上还是多行上。
- 标记长度超过 30 个字符的标识符。  
Oracle 不支持长度超过 30 个字符的标识符。若启用了此首选项，在你键入过长的标识符中，将立即显示标记。
- 只读窗口的超链接导航。  
默认情况下，超链接导航（请参见第 22.17 节）将打开选定对象的可更新窗口。启用此首选项即可将此行为更改为只读窗口。

## 20.13 用户界面 – 字体



字体的首选项控制了浏览器字体、编辑器字体和用于各种诸如 SQL 窗口、属性窗口等的表格的字体。通过按**选择**按钮，你可以更改字体的名称、样式和颜色。你可以为每种字体设置背景色（Bkg）。对于表格和编辑器字体，可以指定在将文本复制到 Windows 剪贴板时是否包含字体样式。若禁用，则仅复制文本。你还可以为所有对话框设置主字体。

## 20.14 用户界面 – 代码助手



当你键入数据库对象的名称时，代码助手将自动地显示关于它们的信息（请参见第 22.3 节）。这个首选项页允许你定义这个特性的行为。

- 自动激活**  
在某一个延迟之后，代码助手能自动地被调用（请参见下面）。你还可以通过功能键选择手工激活代码助手。
- 延迟**  
编辑器在显示代码助手列表之前将等待的毫秒数。
- 代码样式**  
控制了当你选择了已选的项目时它们将怎样被插入到编辑器中：
  - Smart - 代码助手将考虑被描述的对象来决定样式。
  - Init Caps - 每个词（用下划线分隔）的首字符大写。
  - Lowercase - 所有字符都转换到小写。
  - Uppercase - 所有字符都转换到大写。
- 如果可能使用原来的大小写**  
这个选项被允许时，如果可能的话，代码助手将确定来自于存储于 Oracle 词典的源中的标识符的大小写。这将应用到所有的程序单元和它们的元素（参数、类型等）以及应用到查看列，并且越过了上面描述的代码样式首选项。如果原始的大小写不能被确定，代码样式将被应用。  
你可以因执行的原因要求禁用这个特性。

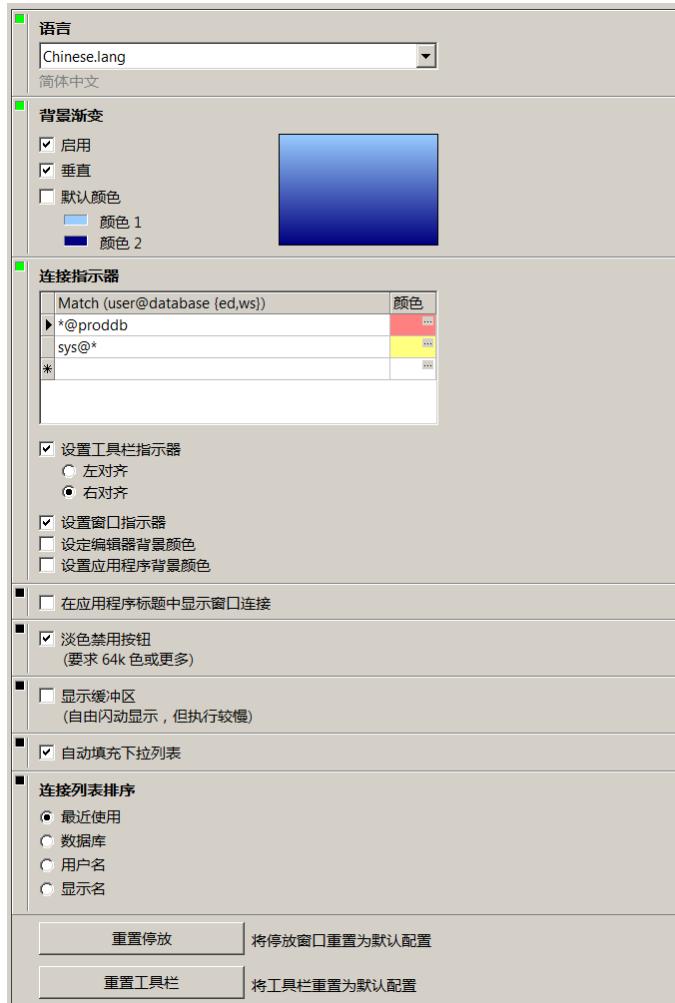
- 描述用户  
确定了当你键入一个后面跟随着句点的用户名时被用户拥有的对象是否被列出来。如果这个选项被允许，你还可以定义哪些对象类型你要包括在中表中。
- 描述上下文  
确定了代码助手是否应该描述当前用户、编辑器和程序单元的上下文。
- 最少字符数  
确定了在上下文描述能自动地被调用之前有多少个字符的词需要被键入。注意，你始终可以手工调用代码助手，即使字符数没有被键入也是这样。
- 描述标准函数  
在默认的情况下，代码助手将描述标准的函数诸如 `to_char`、`add_months` 等。如果你很熟悉这些函数，你可以禁用这个选项。
- 在弹出窗口中显示变量和类型声明。  
若启用此选项，将鼠标指针悬停在程序窗口内的变量或类型上，即可在弹出窗口中显示声明。
- 在选择多个参数时对齐参数值。  
启用时，当参数在不同行时，参数值 (`=>`) 将会对齐。
- 选定多个项目时在一行显示最多项目。  
如果超过选定项目数量时，每个项目将放置在单独一行。否则所有项目将放置在一行。

## 20.15 用户界面 – 键配置



在这一页，你可以为 PL/SQL Developer 中的所有功能定义你自己的键配置。仅仅选择列表中的功能再按你要用于这个功能的组合键就可以。要清除指定功能的热键，请从列表中选择它并按 `ESC` 键。

## 20.16 用户界面 – 外观



- **语言**  
如果你有多国语言版本的 PL/SQL Developer，你可以在这中选择一种语言。随后，所有的说明和信息都将显示在那种语言中。
- **背景梯度**  
允许你定义 PL/SQL Developer 工作空间的颜色和梯度。如果你不要求使用梯度，只用单一的颜色，请把颜色 1 和颜色 2 设为同样的值。

- **链接指示器**

当 PL/SQL Developer 正在使用指定的链接时，这些选项允许你提供一些可见的指示。你可以以用户/数据库格式定义许多链接匹配并与颜色关联起来。随后，这个颜色能与左或右对齐的工具栏指示器、编辑器背景或应用程序背景关联起来（越过背景梯度）。

链接匹配可以包含 \* 和 ? 通配符。在上面的例子中，当以任何用户链接到 PRODDB 数据库时，编辑器背景和右对齐工具栏指示器将是红色的；当以 SYS 用户链接到任何数据库时，它将是黄色的。列表中的第一个项目将具有优先权，所以当以 SYS 用户链接到 PRODDB 数据库时指示器将是红色的。你可以在数据库名称后添加`/版本`或`/工作区`，表示特定版本或工作区。

- **在应用程序标题中显示窗口连接。**

启用时，当前窗口（最顶层）的连接将包含在应用程序标题中。禁用时，则包含主连接，另请参见第 17 章。

- **淡色禁用按钮**

当被允许时，被禁用的按钮将以淡颜色显示。当被禁用时，标准的灰色外观被用于被禁用的按钮。对于 256 色显示，被禁用的按钮总是灰色的。

- **显示缓冲**

当移动窗口或重新确定窗口大小时，屏幕将显示出比较少的闪烁，但显示执行可能稍微慢一些。

- **自动完成下拉列表。**

若启用此选项，带有下拉列表的编辑字段将使用列表中的匹配项目自动完成。

- **连接列表排序。**

控制在连接列表中如何对连接进行排序（参见第 16 章）。

- **禁用 XP 样式。**

此按钮将禁用 Windows 操作系统显示属性中定义的 XP 样式。

- **重置停放窗口。**

将停放窗口重置为默认配置。

- **复位工具栏。**

将工具栏复位为默认配置。

## 20.17 用 户 界 面 – NLS 选 项



在这一页上，你可以定义将被用于 PL/SQL Developer 的日期、时间和数字格式。

### 日期格式:

- 自定义 一种被明确定义的日期格式，遵循与 Windows 地区设置一样的规则。
- Windows 格式 与 Windows 地区设置一样的格式。
- Oracle 格式 与你的 Oracle 环境 (NLS\_DATE\_FORMAT) 定义的格式一样的格式。

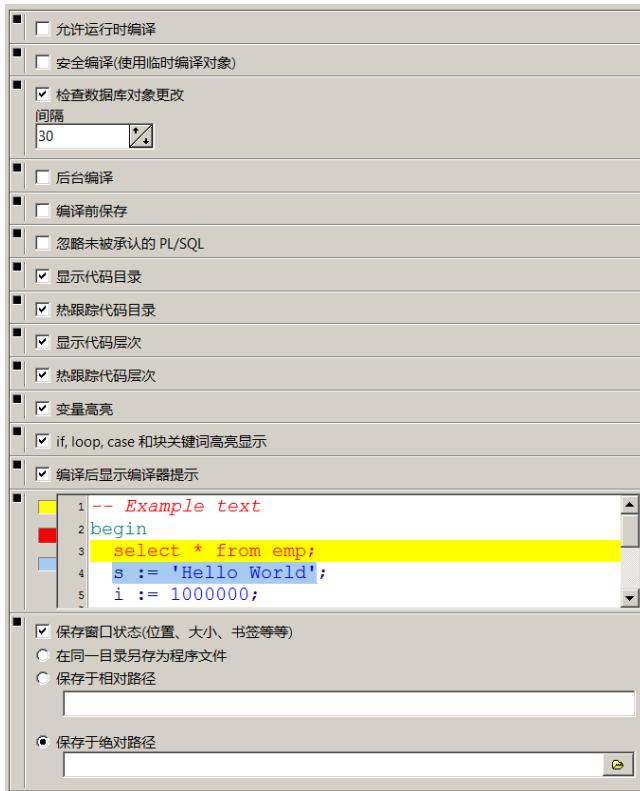
### 时间格式:

- 自定义 一种被明确定义的时间格式，遵循与 Windows 地区设置一样的规则。
- Windows 格式 与 Windows 地区设置一样的格式。

### 数字格式:

- 自定义 在选中此项时，你可以显式选择十进制符号和数字组符号。
- Windows 格式 Windows 区域设置中定义的格式。
- Oracle 格式 你的 Oracle 环境中定义的格式 (NLS\_NUMERIC\_CHARACTERS)

## 20.18 窗口类型 – 程序窗口



这些首选项影响了程序窗口的行为（请参见第 4 章）。

- 允许运行时编译

在多路或双路会话模式中，当你正在运行测试脚本或 SQL 脚本时，你在技术上能够编译程序单元，这可能导致锁定。使用这个首选项，这种情况可以被预防。

- 安全编译(使用临时编译对象)

当被允许时，每个编译都首先通过使用临时编译对象来测试。当这个测试编译成功后，实际的对象就被编译了。在一个无效对象可能引起问题的环境中，如果你正在编译对象，这个选项就可能是有用的。

- 检查数据库对象更改间隔。

启用时，程序窗口将按照给定时间间隔检查是否有另一名用户更改了数据库对象。如果对象已更改，你可以查看数据库源和程序窗口源之间的差异，也可选择是否使用这些更改更新程序窗口。

- 后台编译

当被允许时，编译在后台线程中使用单独的会话被执行。这样，当编译正在发生时，你可

以继续进行其他工作。这将为每个程序窗口导致额外的数据库会话。注意，这个首选项仅有效于会话模式首选项（请参见第 20.1 节）被设为多路会话的情况下。

- 编译前保存  
如果你要保证数据库对象与文件系统是同步的，你可以允许这个选项。
- 忽略未被承认的 PL/SQL  
当被允许时，程序窗口将忽略所有的不以“create or replace <程序单元类型>”开始的 SQL。程序文件中其他的 SQL 或 SQL\*Plus 命令将被忽略。
- 显示代码目录  
这个首选项控制了代码目录窗格是否将被显示，要了解更多详细资料请参见第 22.13 节。
- 热跟踪代码目录  
控制当鼠标在代码目录窗格中的项目上移动时编辑器中的 PL/SQL 码是否将自动地被高亮显示。
- 显示代码层次  
这个首选项控制了代码层次窗格是否将被显示，要了解更多详细资料请参见第 22.14 节。
- 热跟踪代码层次  
控制当鼠标在代码层次窗格中的项目上移动时编辑器中的 PL/SQL 码是否将自动地被高亮显示。
- 高亮变量  
当光标被定位到变量上时，所有这个变量的具体值都会被高亮显示。这使看到变量被用于哪中和更改产生的影响是什么变得容易。如果这个选项被禁用，你需要使用来自于编辑菜单的查找匹配项目。
- If, loop, case 和块关键字高亮显示  
那时光标将被定位于 if / then / else / elsif / end if 关键字、loop / end loop 关键字、for / loop / end loop 关键字、while / loop / end loop 关键字、case / when / then / else / end case 和 declare / begin / end 关键字。如果这个选项被禁用，你需要使用来自于编辑菜单的查找匹配项目。
- 编译后显示编译器提示  
编译器提示提供了关于未用声明、未用分配、潜在错误和警告的信息。当这个首选项被允许时，这些提示将自动地被显示在编译的后面、在编译错误的下面（如果有的话）。如果你禁用了这个首选项，你必须明确地选择来自于工具菜单或工具栏中的显示编译器提示项目。
- 错误背景  
选择编译错误文本的背景的颜色。
- 错误文本  
选择编译错误文本的颜色。
- 热跟踪高亮显示  
为代码目录热跟踪、代码层次热跟踪以及 if 、loop 、case 和块关键字选择高亮文本的背景颜色。
- 保存窗口状态  
当被允许时，程序窗口的位置、大小、书签、颜色标记和置换变量都会被保存在一个单独的 .pfi 文件中，并且当文件或数据库对象在后来再次被查看或编辑时能被恢复。

## 20.19 窗口类型 – SQL 窗口

<input type="checkbox"/> 自动提交 SQL
<input type="checkbox"/> 自动提交已发布的记录
<input checked="" type="checkbox"/> 刷新已发布的记录
<input checked="" type="checkbox"/> 自动执行查询
<input type="checkbox"/> 自动选择语句
<input checked="" type="checkbox"/> 询问保存新窗口
<input checked="" type="checkbox"/> 在新窗口中链接查询
<input type="checkbox"/> 数字字段 to_char
<input type="checkbox"/> 日期字段 to_char
<input checked="" type="checkbox"/> 从数据库中读取 LF 作为 CR/LF
<input checked="" type="checkbox"/> 在结果表格中显示字典信息
<input type="checkbox"/> 显示隔条 (行号)
<input checked="" type="checkbox"/> 导出和复制时包含行号
初始的空行编辑器 <input type="text" value="3"/>
<input type="checkbox"/> 空值单元格颜色
<input type="checkbox"/> 行颜色交替显示 <input checked="" type="checkbox"/> 启用
<input type="checkbox"/> 比较高亮颜色
每页记录数 <input type="radio"/> 自动确定 <input type="radio"/> 固定 <input type="text" value="100"/> <input type="radio"/> 所有记录
最大结果集的大小(0 为不限制) <input type="text" value="100"/> MB
实例查询 <input type="checkbox"/> 不区分大小写 <input type="checkbox"/> 自动包含 <input type="checkbox"/> 忽略时间部分
数字布局 <input type="radio"/> 左对齐 <input type="radio"/> 右对齐 <input checked="" type="radio"/> 格式化 <input type="radio"/> 使用千分符格式

这些首选项影响了 SQL 窗口的行为（请参见第 7 章）。

- 自动提交 SQL  
这个选项控制了在 SQL 窗口中执行的 DML 语句是否被自动提交。
- 自动提交已录入的记录  
这个选项控制了通过结果表格录入的记录是否被自动提交。

- 刷新已录入的记录。  
启用该选项时，向数据库应用更改后，插入或更新的记将显示在结果表格中。这使您在插入触发器、更新触发器之前能立即看到对记录默认值的更改。
- 自动执行查询  
如果这个选项被允许，在为表或视图对象选择*查询数据*或*编辑数据*时被创建的 SQL 窗口将立即运行。如果这个选项被禁用，你可以在执行前修改已产生的 select 语句。
- 自动选择语句  
如果你在 SQL 编辑器中有不只一个 SQL 语句（通过分号分隔），SQL 窗口在运行前将自动地在光标当前被定位的地方选择语句。这样，你不必明确地选择你自己的语句。如果仅用此选项，仍然可以在首选项中定义*执行当前语句* SQL 窗口功能键（参见第 20.15 节），以快速执行当前执行语句。
- 询问保存新窗口  
当你关闭刚被创建的 SQL 窗口时，你通常都会被问是否要保存它到文件中。如果你禁用了这个选项，这个确认将被忽略。如果你已经修改了一个前面打开的 SQL 文件，你将总是被问是否保存这些更改，不管这个首选项是否已选。
- 在新窗口中执行链接查询  
当这个首选项被设置时，执行链接查询（请参见第 7.4 节）将创建一个新的 SQL 窗口。如果没有设置，链接查询将在当前的 SQL 窗口中被执行。
- 数字字段 `to_char`  
当这个首选项被允许时，数字字段在服务器中将被转换为字符值，会话的 NLS 参数将决定值的格式，PL/SQL Developer 将不另外执行数字格式化或确认。
- 日期字段 `to_char`  
当这个首选项被允许时，日期字段在服务器中将被转换为字符值，会话的 NLS 参数将决定值的格式，PL/SQL Developer 将不执行任何另外的日期格式化或确认，并且在编辑日期字段时日期采集者将不可用。
- 从数据库中读取 LF 作为 CR/LF  
Oracle 数据库中的字符数据通常包含 LF 字符作为行终止字符。在 Windows 中常见的此类字符则是 CR/LF 对。启用这个选项时，SQL 窗口会在 LF 和 CR/LF 之间切换。禁用时，则按原样获取 LF 字符。
- 在结果表格中显示字典信息  
当被允许时，下列词典信息将被显示在结果表格中：
  - 列数据类型、可选项和注释都将被显示在状态行上。
  - 对于带有检查指定值的检查约束的列（例如 `col in (值 1, 值 2, ...)`），查找列表也将显示出来。
  - 对于带有外键约束的列（针对少于 1000 行的小表），查找列表也将显示出来。
- 显示边栏（行号）  
若启用此项，SQL 编辑器将在编辑器左侧的边栏中显示行号和行状态。
- 导出和复制时包含行号。  
启用这个选项时，导出或复制结果集时将隐式包含行号。要忽略行号，需要显式选择希望包含的列。禁用该选项时则完全相反，即除非显式选择行号，否则默认忽略行号。

- 初始空白编辑器行

在 SQL 窗口中执行 Select 语句时，结果窗格将显示出来并占用 SQL 编辑器的空间。这个首选项可控制保留在 SQL 编辑器中的最小行数——无论 Select 语句的行数是多少。

- Null 值单元格颜色

这个颜色被用于显示 null 值，这使 null 值与空格值、不直接显示在表格中的值（例如 LOB 和 Long）很容易被辨认出来。

- 可选行颜色

若启用此项，此颜色将用于可视化结果窗格中的可选行。

- 比较高亮颜色

此颜色将用于在使用比较功能时高亮显示不同或相同的值。

- 每页记录数

这个首选项决定了 select 语句最初要获取多少条记录、每次按下一按钮在 SQL 窗口的结果表格中显示多少条记录。

- 自动确定 - 结果表格的大小决定了多少条记录被获取。

- 固定 - 获取指定的记录数。

- 所有记录 - 当查询被执行时，所有记录立即都被获取。

- 最大结果集的带下。

确定结果集的最大大小（单位为 MB）。达到此上限时，SQL 窗口将要求用户确认，之后才能继续操作。

- 数字版面

这个首选项控制了怎样显示数字列：

- 左对齐 - 左对齐显示值，不带任何格式。

- 右对齐 - 右对齐显示值，不带任何格式。

- 指定格式 - 右对齐显示值，带有符合字段的精度和范围的格式。

- 用许多分隔符指定格式 - 类似于前面的选项，但还将包括千位分隔符（例如 1,277.65）。

- 实例查询

这些首选项控制了在 QBE 模式期间 SQL 窗口的默认行为（请参见第 7.3 节）：

- 不区分大小写 - 字符字段的查询值被视为不区分大小写。

- 自动包含 - 字符字段仅仅需要包含查询值。

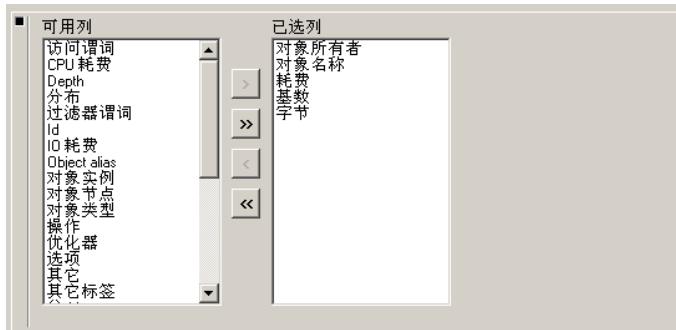
- 忽略日期字段时间部分 - 如果查询值不包含时间部分，日期字段的时间部分将被忽略。

## 20.20 窗口类型 – 测试窗口



- 在执行后自动提交  
这个选项控制了被执行的测试脚本是否自动地被提交。
- 执行前自动扫描绑定变量。  
启用这个选项时，测试窗口将自动在 PL/SQL 块中搜索绑定变量，将缺失的变量添加到变量列表，并从列表中禁用 PL/SQL 块中未出现的变量。所添加的变量类型为字符串。

## 20.21 窗口类型 – 计划窗口



这些首选项控制了计划表的哪一列应该被显示在计划窗口中（请参见第 6.1 节）、它们应该按什么顺序显示出来。

## 20.22 工具 – 差异



在这个首选项页上，对于比较用户对象工具，你可以选择外部的差异工具（请参见第 21.10 节）。你可以用下列指令来设定参数——大小写或空格键忽略。

- <IgnoreCase=on|off>  
通过转换 on 和 off 指令来决定是否忽略大小写
- <IgnoreWhiteSpace=on|off>  
通过转换 on 和 off 指令来决定是否忽略空格

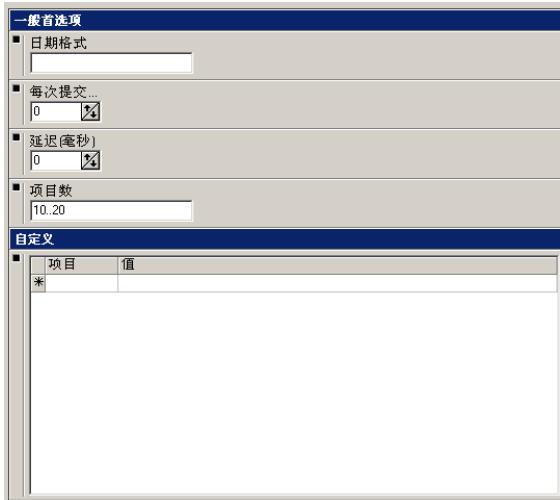
例：在中英双语中，你可以用/i 指令来转换到忽略大小写，用/!i 指令来区分大小写，用/w 指令来忽略空格，用/!w 来强调空格。这些转换指令必须写在特定的文件名后。因此中英双语的转换指令如下：

```
%1 %2 <IgnoreCase=/i|!/i> <IgnoreWhiteSpace=/w|!/w>
```

在需要对比两个文件时，如果需要区分大小写，同时忽略空格差异，所得到的参数列表应如下所示：

```
c:\temp\file1.sql c:\temp\file2.sql /!i /w
```

## 20.23 工具 – 数据生成器



这个首选项页允许你为数据生成器工具定义默认的全局设置。这些默认设置可以支配每个数据生成器的定义。要了解更多关于这些首选项含义的详细资料，请参见第 21.9 节。

## 20.24 工具 – 任务列表



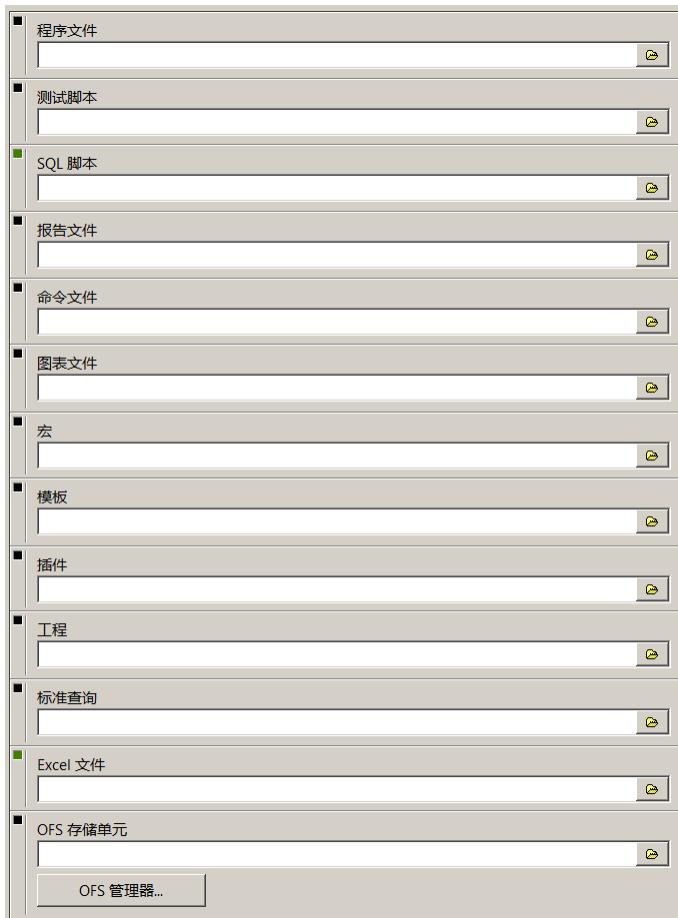
这些首选项控制了在任务项目编辑器中的优先权、种类和所有者选择列表中的值。对于优先权，它可以用数字的值加上前缀，因此这个列表可以用优先权来排序。要了解更多详细资料，请参见第 15 章。

## 20.25 工具 – 重新调用声明



在这个首选项页上，你可以定义声明重新调用缓冲区中声明的最大数和应该被储存的目录。目录名称可以包含环境变量（例如，%APDATA%\PLSQL Developer）。要了解更多详细资料，请参见第 22.4 节。

## 20.26 文件 – 目录



- 程序文件

定义当打开或保存程序文件时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存程序文件的目录。

- 测试脚本

定义当打开或保存测试脚本时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存测试脚本的目录。

- SQL 脚本

定义当打开或保存 SQL 脚本时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存 SQL 脚本的目录。

- 报告文件  
定义当打开或保存报告文件时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存报告文件的目录。
- 命令文件  
当你从菜单中或通过键入 `@file` 打开一个命令文件时，这个目录将被默认使用。如果你保留这个目录为空，正在工作的目录将被使用。
- 宏  
这个目录将被用于保存你的宏库。如果你保留这个目录为空，那么 PL/SQL Developer 目录中的 *Macro* 子目录将被使用。
- 模板  
这个目录将被用于保存程序文件模板。如果你保留这个目录为空，那么 PL/SQL Developer 目录中的 *Template* 子目录将被使用。
- 插件  
这个目录将被用于搜索插件。如果你保留这个目录为空，那么 PL/SQL Developer 目录中的 *PlugIn* 子目录将被使用。
- 工程  
定义当打开或保存工程时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存工程的目录。
- 标准查询  
定义标准查询被储存的目录。如果你保留这个目录为空，标准查询将被储存于你的概要文件的应用数据目录。
- Excel 文件  
定义“复制到 Excel”功能（请参见第 7.2 章）在哪个目录中创建 Excel 文件。
- OFS 存储单元  
定义了 OFS 存储单元目录文件（`ofs.1df`）。OFS 管理器按钮允许你运行 OFS 管理器来查看或编辑这个文件。请参阅第 31.1 节。

## 20.27 文件 – 扩展名



在这一页上，你可以定义所有文件类型的默认扩展名。当保存或载入某一文件时，如果你不在文件对话框中指定扩展名，就假定为默认扩展名。对于每个文件类型，你还可以指定“其他”扩展名，它同样将显示在文件对话框中，多个扩展名可以用逗号来分隔。

通过按**注册文件类型**按钮，你可以把这些默认扩展名与 PL/SQL Developer 联系起来。在 Windows Explorer 中双击带有这样扩展名的文件将自动运行 PL/SQL Developer 。对话框将显示出允许你选择哪一种扩展名与 PL/SQL Developer 联系起来。

## 20.28 文件 – 格式



换行选项控制了行在文本文件中怎样被终止。行可以采用 Windows 样式以 CR/LF 字符对来终止，或者采用 Unix/Linux 样式以单一的 LF 字符来终止。你还可以指定文件采用特定存储单元的 Unix 格式和别的 Windows 格式。你可以指定分开的行的多个目录，并且通过按添加存储单元按钮 (...) 来选择一个目录。注意不但被保存在这个目录中的文件被格式化为 Unix 样式，而且还包括被保存在这个目录下面的子目录中的所有文件。

如果文件总是以 Unicode UTF8 格式、ANSI 格式保存，或 PL/SQL Developer 尝试检测文件是否包含 Unicode 格式特征时，编码选项就会将文件保存为 UTF8 格式。如果文件包含的是 ANSI 特征时，文件就会保存为 ANSI 格式。如果禁用 Save with BOM，Unicode 字节顺序标记头部将从文件中忽略掉，因此打开此文件的工具将需要自动确定格式。

用智能引号导出控制了值是怎样按 CSV 格式被导出的。当被禁用时，所有的值都被加上引号；当被允许时，仅有那些需要的值被加上引号，这是包含了逗号或引号的值。

导出到包含 SQL 语句的 Excel 选项控制在将结果导出为 Excel 格式时 SQL 语句是否包含在单独的 Excel 页面中。

Excel 输出格式控制是否始终以 XLS（Excel 2003 及以下版本）格式或 XLSX（Excel 2007 及更高版本）导出，也可以选择在每次执行导出时询问选择这两种格式中的哪一种。

## 20.29 文件 – 备份



这个首选项页包含了针对你的源文件进行备份和恢复的选项。

### 创建备份文件

在 PL/SQL Developer 保存文件之前，最初的文件在被覆盖以前将被保存到备份文件中。这个备份文件将有同样的名称，扩展名将被加上一个前缀 ~（例如，employee.pck 将被保存为 employee.~pck）。这样，你将始终有一个早先文件的备份。

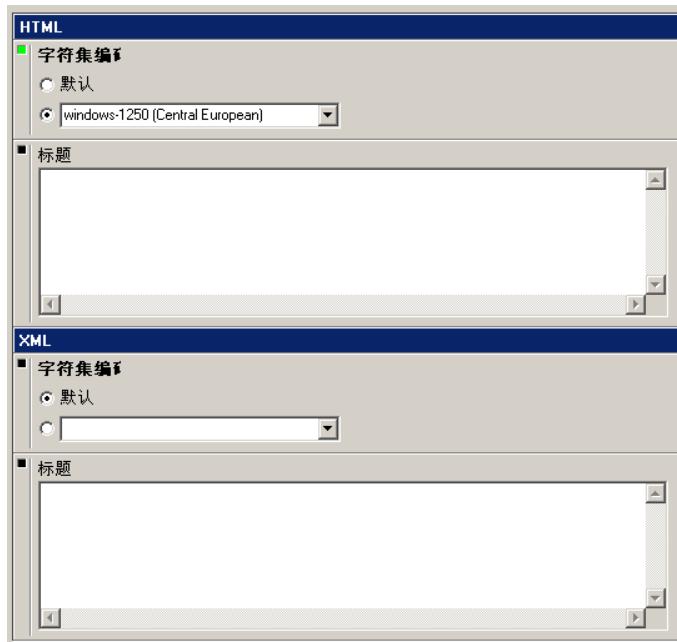
所有的允许你打开文件的文件对话框都将有一个“备份文件 (\*.~\*)”过滤器，它允许你容易地打开备份文件。

### 恢复

在某一事件上，PL/SQL Developer 可以储存所有打开文件到一个临时位置。万一出现了严重的问题（例如，断电、软件崩溃等），PL/SQL Developer 在随后运行时允许你恢复这些文件。你可以指定下列事件：

- 执行前保存  
在你运行程序窗口、测试窗口、命令窗口、SQL 窗口或报告窗口之前，所有打开的文件都将被储存在一个临时的目录中。
- 定时保存  
所有打开的文件都将在指定的时间间隔被储存在一个临时的目录中。

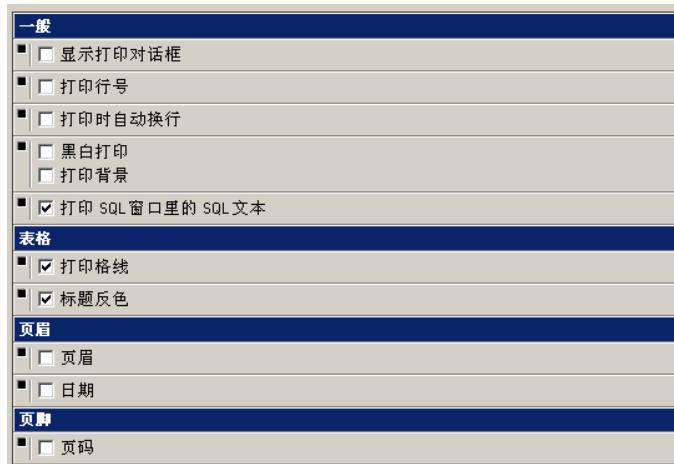
## 20.30 文件 – HTML/XML



这些首选项允许你控制由报告窗口和导出功能生成的 HTML 和 XML 输出：

- **字符集编码**  
允许你定义 HTML 或 XML 文档的字符集编码。这应该与你的 Oracle 环境的字符集相符，所以数据能被正确地解释。
- **标题**  
这个文本将被插入到 HTML 文件的 <HEAD> 部分或者被插入到 XML 文件的标准 XML 标题中。

## 20.31 其他 – 打印



### 一般

- 显示打印对话框  
当按**打印**按钮时，打印对话框将首先显示。
- 打印行号  
当打印 SQL 或 PL/SQL 源代码时，在每一行之前将显示行号。
- 打印时自动换行  
比页宽的行将继续在下一行被打印。
- 黑白打印  
所有被打印的输出都是黑白的，禁用这个选项将创建带有彩色或灰度的输出。
- 打印背景  
当被允许时，背景颜色将被打印。
- 打印 SQL 窗口中的 SQL 文本  
决定了当打印 SQL 窗口的内容时在查询结果之前 SQL 文本是否被打印。

### 表格

- 打印格线  
决定了表格的格线是否被打印。
- 标题反色  
决定了表格的标题是否按相反的颜色打印。

## 页眉

- 标题  
当被允许时，在页眉的左边，每一页都将有文件类型和文件名被打印。
- 日期  
当被允许时，在页眉的右边，每一页都将有当前的日期和时间被打印。

## 页脚

- 页码  
当被允许时，页脚的中间将包括页码。

## 20.32 其他 – 更新与消息



允许你控制 PL/SQL Developer 是否和何时检查 Allround Automations 网站的关于 PL/SQL Developer 的更新和消息。如果你禁用这个选项，通过按相应的选项或者通过帮助菜单中的检查在线更新和检查在线消息项目，你可以明白地检查更新和消息。如果你是通过代理服务器连接到 Internet，请设置代理设置。

## 20.33 首选项集

正如第 20 章所提到的，你可以在多重层级上定义首选项集。要定义首选项，请按首选项集选择列表旁边的配置首选项集按钮。下列对话框将显示出来：



个人首选项是为当前 Windows 用户定义的（插图中的 *john*）。默认首选项可以被系统管理员定义，它为所有的 Windows 用户既在全局层次又在链接层次提供了默认的首选项。

### 个人首选项

在默认的情况下，所有的用户都将有一个 *Default <用户>* 首选项集，这是你将默认访问的首选项集。要为一个或多个 Oracle 链接创建另外的首选项集，请按首选项列表右边的新建按钮。现在，你就可以输入描述和添加一个或多个链接了。对于每个链接，你可以使用通配符来匹配多用户或数据库。在上面的例子中，*Order admin databases* 首选项集已经被 *john* 定义，用于 *dev\_oa*、*tst\_oa* 和 *prd\_oa* 数据库。无论 *John* 何时链接这三个数据库之一，这个首选项集都将被使用。如果首选项没有在这个集中被定义，来自于默认用户的首选项集中的相应的首选项将被使用。

### 默认首选项

系统管理员可以为所有用户或所有链接定义全局的默认首选项，还可以为一个或多个链接定义默认的首选项集。要做到这一步，请转到默认首选项标签页，按首选项列表旁边的新建按钮。如果你不输入任何链接，新的首选项集将为所有用户定义默认的首选项。如果你输入了一个或多个链接，首选项集将为这些链接定义默认的首选项。

## 首选项集优先权

如果首选项是按多层次定义的，则最特殊的层次具有优先权：

1. 个人链接首选项集（最大优先权）
2. 个人 Windows 用户首选项集
3. 默认链接首选项集
4. 默认全局首选项集（最小优先权）

## 首选项集文件和目录

首选项集被存储在 PL/SQL Developer 安装目录中的 *Preferences* 子目录中的一个单独的文件中。在这个首选项目录中，个人首选项被存储在以 Windows 用户命名的子目录中。例如：

```
C:\Program Files\PLSQL Developer\Preferences\John
```

默认首选项被存储在 *Default* 子目录中。例如：

```
C:\Program Files\PLSQL Developer\Preferences\Default
```

如果你在一个文件服务器上安装 PL/SQL Developer，那么你必须提供可读写访问的 *Preferences* 目录、可读访问的 *Preferences\Default* 子目录。你也可以提供 *PREFPATH* 命令行参数来为用户的首选项指定目录：

```
plsqldev.exe prefpath=u:\userdata
```

如果 Windows 用户 *John* 使用 PL/SQL Developer，那么他的个人首选项集将被储存在 *u:\userdata\john* 中。默认首选项始终被储存在 PL/SQL Developer 安装目录中的 *Preferences\Default* 子目录中。

要保证所有用户都使用一个类似的首选项路径，你可以在 *params.ini* 文件中设置这个参数（请参见第 33.4 节）。

## 21. 工具

在工具菜单中有几个工具是可用的，这几个工具将在这一章中被描述。

### 21.1 对象浏览器

对象浏览器工具在第 18 章中有详细描述。在工具菜单中，通过选择对象浏览器项目，你可以查看或隐藏对象浏览器。你还可以定义来自于这个菜单的对象浏览器过滤器，它在第 18 章中也会描述。

### 21.2 查找数据库对象

这个工具允许你查找基于指定条件的数据库对象，这个条件是指象对象名、类型、所有者或状态这样的条件，另外还有针对对象 DDL 源的全文搜索条件：



你可以在这个表单的上面三部分输入搜索条件。

## 全文搜索

这个对话框中的选项或多于或少于文本编辑器中的查找对话框的选项，这些选项允许你按不区分大小写搜索、按全字匹配搜索和按正规表达式搜索。如果你忽略了全文搜索，那么符合其他条件的所有对象都将被获得。

### 对象准则

在这中，你可以输入下列准则：

- **浏览器过滤器** - 根据已选过滤器的准则来限制对象。
- **所有者** - 对象的所有者。你可以使用 SQL 通配符（例如 sys%）。默认值为 <CURRENT USER>，它将为当前链接的用户限制搜索对象。
- **名称** - 对象名称。你可以使用 SQL 通配符（例如 %common）。
- **状态** - 限制搜索有效或无效对象。
- **创建：以后/以前** - 限制搜索在指定日期后和前创建的对象。使用日期输入框左边的复选框来允许或禁用这些条件。
- **修改：以后/以前** - 限制搜索在指定日期后和前被修改的对象。使用日期输入框左边的复选框来允许或禁用这些条件。

### 对象类型

在这中，你可以选择要搜索的对象类型的范围。

### 搜索结果

按下了**搜索**按钮之后，搜索操作就在后台被执行了，搜索结果列表中将显示出搜索结果。

如果你右键单击搜索结果，你会看到对象的弹出式菜单。弹出式菜单中的第一个项目是**定位文本**，它在程序窗口中打开源，并将立即使你获得搜索文本的第一个具体值。你还可以通过双击搜索结果中的项目来调用**定位文本**功能。

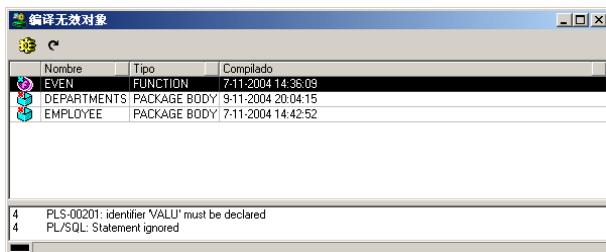
要保存搜索结果，你可以双击它们并从弹出式菜单中的**导出子菜单**项目中选择适当的格式。

### 保存常用的搜索条件

要保存常用的搜索条件，你可以按**保存**按钮并输入目录和文件名。你以后可以按**载入**按钮来重新调用这些搜索条件。

## 21.3 编译无效对象

用这个工具你可以容易地重新编译数据库中所有的无效对象。当你选择它时，显示所有无效对象的对话框将显示出来：



注意，当前活动的浏览器过滤器控制了被包括的无效对象。对话框的底部将显示出当前已选对象以前出现的任何编译错误。

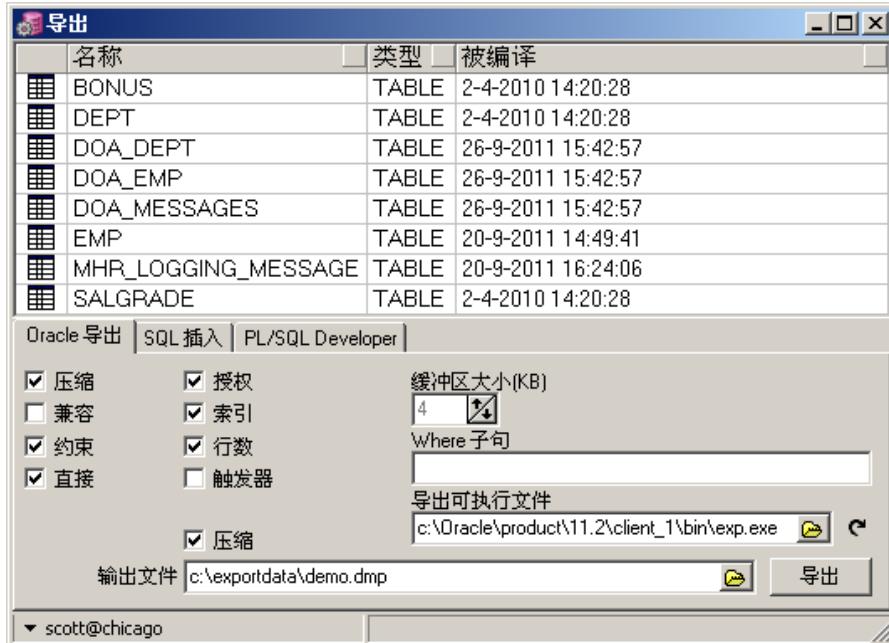
要重新编译对象，请按下工具栏中的**编译无效对象**按钮或按主工具栏中的**执行**按钮。对象的依赖关系将被分析以确定编译应该出现的顺序。对象列表左边的图标指示了哪些对象被成功编译了。编译错误部分将在编译完成之后被更新。

要刷新无效对象的列表，请按下工具栏中的**刷新对象列表**按钮，这将更新带有当前所有无效对象的列表。

在对象列表中，你可以右键单击一个对象来访问它的弹出式菜单。

## 21.4 导出表

导出表工具允许你导出一个或多个表定义和它们的数据到一个文件中，所以你以后还可以导入这些表。在启动了导出表工具之后，你可以选择要导出的用户和表，选定导出方法（Oracle 导出、SQL 插入或 PL/SQL Developer），设置应用到导出方法的各种选项：



导出功能将分析表之间的可能存在的外键依赖，把表按正确的顺序放到导出文件中。

### Oracle 导出方法

前一章中的屏幕截图显示了 Oracle 导出方法的选项。这个方法让你指定你可以应用到 Oracle 导出工具的命令行上的选项（请参见“Oracle 服务器实用工具”指南），随后将运行它来执行导出。导出可执行文件输入框允许你选择一个特定版本的 Oracle 导出工具。在默认的情况下，来自于 Oracle 主目录的最新版本将被使用。

你可以使用压缩选项来自动压缩输出的文件，以节省磁盘空间。例：上面的 C:\exportdata\demo.dmp 文件，添加到压缩文档中会更名为 C:\exportdata\demo.dmp.zip. 原来的那个文件也会随之被删除。

导出完成后，一个新的日志标签页将是可见的，这个页包含了被导出工具创建的日志。

这个方法的优势是导出文件的速度和轻便。记住，无论你在何处都可以用 Oracle 导入工具来导入它。

## SQL 插入方法



这个导出方法创建了一个标准的带有插入语句和（随意的）DDL 语句的 SQL 脚本来重新创建表定义。标签页包含了下列选项：

- **删掉表** - 已产生的 SQL 脚本包含了一些语句在表被重新创建和载入前来删除掉表。**创建表、截短表和删除记录**选项在这种情形下被禁用，它们有固有的值。
- **创建表** - 已产生的 SQL 脚本包含了一些语句在表被载入前来创建表。这包括约束、索引和授权。
- **截短表** - 已产生的 SQL 脚本使用截短表的语句在表被载入前来清空表。这个选项比**删除表**快，但不可用于外键约束存在的情况下。
- **删除记录** - 已产生的 SQL 脚本使用删除语句在表被载入前来清空表。
- **禁用触发器** - 已产生的 SQL 脚本在表被载入前禁用所有的表触发器，之后允许它们。这可以改善执行性能，有时可能在触发器包含不适用于导入过程的情况下是必要的。
- **禁用外键约束** - 已产生的 SQL 脚本在表被载入前禁用了表的所有外键，之后允许它们。这对于自我参照的外键可能是必要的。两个不同表之间的外键将不导致冲突，这是因为表是按正确的顺序被导出的。禁用外键还可能改善执行性能。
- **包括存储** - 已产生的 SQL 脚本为表创建语句包括原始存储信息，例如表空间名和初始大小。这些信息可能与数据库不一致，所以这可能并不总是合适的。
- **包括权限** - 已产生的 SQL 脚本包括对象给其他用户和角色授权的授予权。当对象在不同的数据库中被重新创建时，这些用户和角色显然必须存在。
- **每次提交** - 控制了在提交被执行前有多少条记录将被插入到已产生的 SQL 脚本中。如果这个值是零，在 SQL 脚本的末尾所有的插入将被立刻提交。对于大的导出文件和小的回滚段，在这中给一个适当的值将是必要的。
- **Where 子句** - 仅仅匹配 where 子句的记录将被导出。Where 子句必须适用于所有已选的表！

这个方法的优点是 SQL 脚本的轻便和用文本编辑器来编辑结果的可能性。

使用**压缩**选项来自动压缩输出的文件，以节省磁盘空间。例：上面的 C:\exportdata\demo.dmp 文件，添加到压缩文档中会更名为 C:\exportdata\demo.zip.原来的那个文件也会随之被删除。

## PL/SQL Developer 方法



这个导出方法创建了定制的、压缩的 PL/SQL Developer 导出文件，它带有表定义和数据。这个标签页包含了下列选项：

- Where 子句 - 仅仅匹配 where 子句的记录将被导出。Where 子句必须适用于所有已选的表！
- 压缩文件 - 禁用这个选项将禁用压缩。如果数据不能被很好地压缩（例如，long raw 或 BLOB 数据已经在数据库中被压缩），这可能导致执行性能改善。
- 包括存储 - 已产生的 SQL 为表创建语句包括原始存储信息，例如表空间名和初始大小。这些信息可能与数据库不一致，所以这可能并不总是合适的。
- 包括权限 - 已产生的 SQL 包括对象给其他用户和角色授权的授予权。当对象在不同的数据库中被重新创建时，这些用户和角色显然必须存在。

这个方法的优点是极大地减小了导出文件的大小，所有的选项都重视导入阶段，在导入期间你可以容易地选择表。

## 21.5 导入表

导入表工具允许你从用上一章描述的导出表工具在以前导出的文件中导入表定义和数据。正如导出表工具那样，也有三中方法来导入表，每一种都带有它自己的文件格式。

### Oracle 导入方法



这个方法使用被 Oracle 导出工具创建的 dmp 文件运行 Oracle 的导入工具来导入数据。你可以制定各种选项，每个选项都要符合导入工具的命令行选项（请参见“*Oracle 服务器实用工具*”指南）。导入可执行文件输入框允许你选择一个特定版本的 Oracle 导入工具。在默认的情况下，来自于 Oracle 主目录的最新版本将被使用。

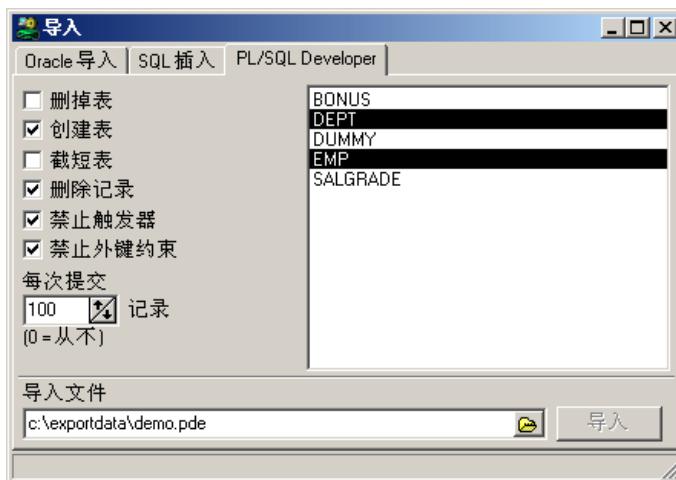
导出完成后，一个新的日志标签页将是可见的，这个页包含了被导入工具创建的日志。

## SQL 插入方法



这个方法是执行用导出表工具生成的 SQL 脚本。你可以选择启动 SQL\*Plus 来运行这个脚本或者使用内置的命令窗口。对于 SQL\*Plus，你也可以选择一个特定的 SQL\*Plus 可执行文件。在默认的情况下，来自于当前 Oracle 主目录的版本将被使用。

## PL/SQL Developer 方法



在选择了以前使用 PL/SQL Developer 方法导出的导入文件之后，你可以从列表中选择导入一个或多个文件。下列选项可以被指定：

- **删掉表** - 在表被重新创建和载入前删除掉表。**创建表**、**截短表**和**删除记录**选项在这种情形下被禁用，它们有固有的值。

- 创建表 - 在表被载入前创建表。这包括约束、索引和授权。
- 截短表 - 在表被载入前截短表。这个选项比删除表快，但不可用于外键约束存在的情况下。
- 删 除记录 - 在表被载入前删除表中的所有记录。
- 禁用触发器 - 在表被载入前禁用所有的表触发器，之后允许它们。这可以改善执行性能，有时可能在触发器包含不适用于导入过程的情况下是必要的。
- 禁用外键约束 - 在表被载入前禁用表的所有外键，之后允许它们。这对于自我参照的外键可能是必要的。两个不同表之间的外键将不导致冲突，这是因为表是按正确的顺序被导出的。禁用外键还可能改善执行性能。
- 每次提交 - 控制了在提交被执行前有多少条记录将被插入到已产生的 SQL 脚本中。如果这个值是零，在 SQL 脚本的末尾所有的插入都将被立刻提交。对于大的导出文件和小的回滚段，在这中给一个适当的值将是必要的。

## 21.6 导出用户对象

要导出一个用户的所有对象的 DDL（数据定义语言）语句，你可以使用导出用户对象工具。这样你可以容易地为另一个用户或者在不同的数据库中重新创建对象。选择工具 菜单的 导出用户对象 并选择用户之后，该用户的所有对象都将显示在这个表格中：



在选择了一个输出文件之后，你可以按 导出按钮来导出对象。如果没有在表格中选择特定的对象，所有的对象都将被导出。你可以通过在它们上面单击来选择对象，使用 *Ctrl* 和 *Shift* 键来选择多个对象和对象范围。

输出文件是一个与 Oracle 的 SQL\*Plus 和 PL/SQL Developer 的命令窗口兼容的 SQL 脚本。在导出操作之后，你可以按 查看按钮在命令窗口中打开已产生的文件。

如果你要包括对象给其他用户和角色授权的权限，**包括权限**选项可以被选取。当对象在不同的数据库中被重新创建时，这些用户和角色显然必须存在。

同样地，你可以允许包括存储选项来包括存储信息，例如表空间名和初始大小。这些可能与数据库不一致，所以这可能并不总是适当的。

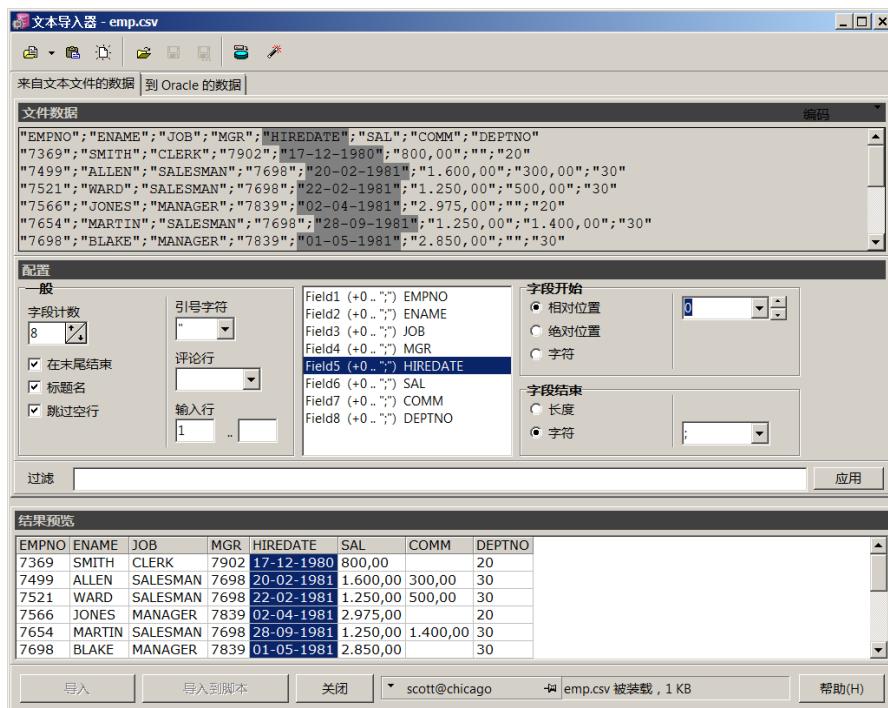
包括所有者选项控制了在结果输出文件中对象是否要加上所有者前缀（例如，用 SCOTT.EMP 代替 EMP）。

单个文件和每个对象的文件以及每个对象的文件/单个文件中的规范和正文选项控制了包含了所有对象的 DDL 的单个 SQL 脚本是否被创建，或者每个对象是否应该被导出到一个特定的文件（对象名为文件名、.sql 为扩展名）。对于包或类型，可以选择在一个文件中获取规范和正文。输出文件将包含一连串的对这些对象特定文件的调用。所有文件都将被写到输出文件的目录中。

## 21.7 文本导入器

使用文本导入器你可以把 ASCII 文件导入到数据库中。象用逗号和制表符分隔的那样的被标定格式的大多数行都是被支持的。导入器将设法自动确定文件的格式，所以大多数时间中你不需要定义任何事情，仅仅选择文件，选择表，如此而已。

当你打开文本导入器并载入文本文件时，你将获得如下窗口：



导入器从文件中载入了前 100 行的数据。原始数据显示在窗口上部的“文件数据”中。中间部分允许你配置文件定义。下部的“结果预览”有一个带有将被导入的数据的表格。

工具栏中有用于选择文本文件或从剪贴板中粘贴文本的按钮。文件可以实际上不限制大小，它们不是被完整地读到内存中。“新建”按钮将清空文本导入器，打开和保存按钮将允许你重新使用定义。

如果你希望将文本导入一个新表，请按下**创建表**按钮。这将调用表定义编辑器，列定义与配置相匹配。

你一载入文件，配制就被自动确定，如果这不正确，你自己可以创建和修改它。把字段计数设置为正确的数字，从字段列表中选择字段来定义字段的定义。在文件数据部分，被指定的字段被高亮显示，这允许你检查字段定义是否是正确的。你有下列的选项来配置文本导入器的定义：

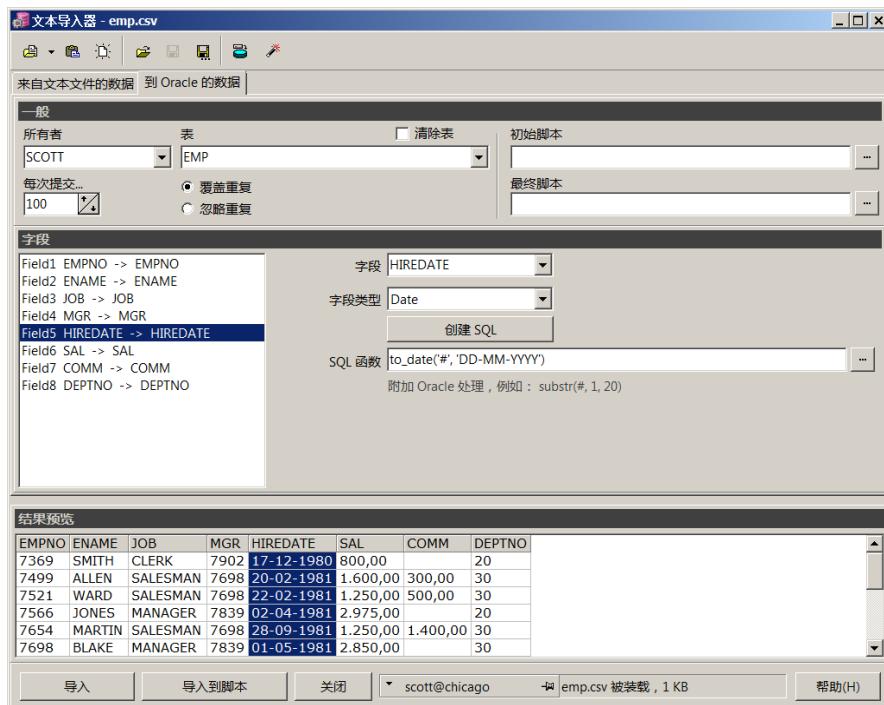
- 一般 - 字段计数  
每行文本的字段数量。
- 一般 - 在行末尾结束  
指出记录数据在行末尾处结束。
- 一般 - 标题名  
指出第一行文本是不是字段名。
- 一般 - 跳过空行  
若启用此项，文本文件中的空行将被跳过。
- 一般 - 引号字符  
指出确定字符串的字符，通常为双引号或单引号。
- 一般 - 注释行  
在这里可以选择指出文本文件中的注释行的一个或多个字符。
- 一般 - 导入行  
导入文本文件中的第一行和最后一行。若将此项保留为空，则导入整个文件。
- 字段开始 - 相对位置  
本字段的首端与前一个字段的末端的相对位置。0 表示前一个字段的末端就是本字段的开始位置。
- 字段开始 - 绝对位置  
这指出字段开始于一个固定位置。
- 字段开始 - 字符  
这指出字段开始于一个特定的字符，这与前一个字段的末端有关。
- 字段结束 - 长度  
指出字段有一个固定的长度。
- 字段结束 - 字符  
指出字段结束于一个特定的字符。

对于标准的逗号分隔的文件（csv），所有的字段的开始相对位置都是 0，结束这个字段的结束字符应该是逗号。行结束符（cr/lf 或 lf 或 cr）也将指出字段的结束，所以对于一行上的最后一个字段，结束字符并不是重要的。

要过滤文件内容，可以在**过滤器**字段中输入有效的 SQL 表达式。可以通过绑定变量及字段名称引用一个字段值。例如，如果想进包含部门 10 中担任“CLERK”职务以外的其他职务员工，可输入以下过滤器表达式：

```
:deptno = 10 and :job <> 'CLERK'
```

当你有一个正确的文件定义时，你需要选择一个 Oracle 表并指出哪个文本字段应该插入到哪个 Oracle 字段中。你可以在第二个标签页上做这件事：



上部有一些一般的导入参数，“字段”部分允许你在文本文件字段与 Oracle 字段之间建立联系，下部是你的信息的结果预览。

你可以选择下列的一般首选项：

- 所有者/表  
你要导入进数据的 Oracle 表（或视图）。在选择了表之后，导入器将设法自动地确定字段和字段类型。这仅仅工作于文本文件有字段标题的情况下。
- 清除表  
若启用此项，表中的所有记录都将在导入前被删除。
- 每次提交...  
指出每次你要提交多少条纪录。如果你设置它为 0，在整个导入的末尾所有的数据都将被提交。
- 覆盖重复  
重复的记录将在数据库中被更新。重复以表的主键值为依据。
- 忽略重复  
重复的记录将被忽略。
- 初始化脚本/终止脚本  
这些脚本将在文本文件导入前/后执行。可以使用任何 SQL 命令或 PL/SQL 块，使用分号或正斜杠分隔。

在字段定义中，你能看到一个来自于文本文件带有字段的列表。对于每个字段，你可以设置如下：

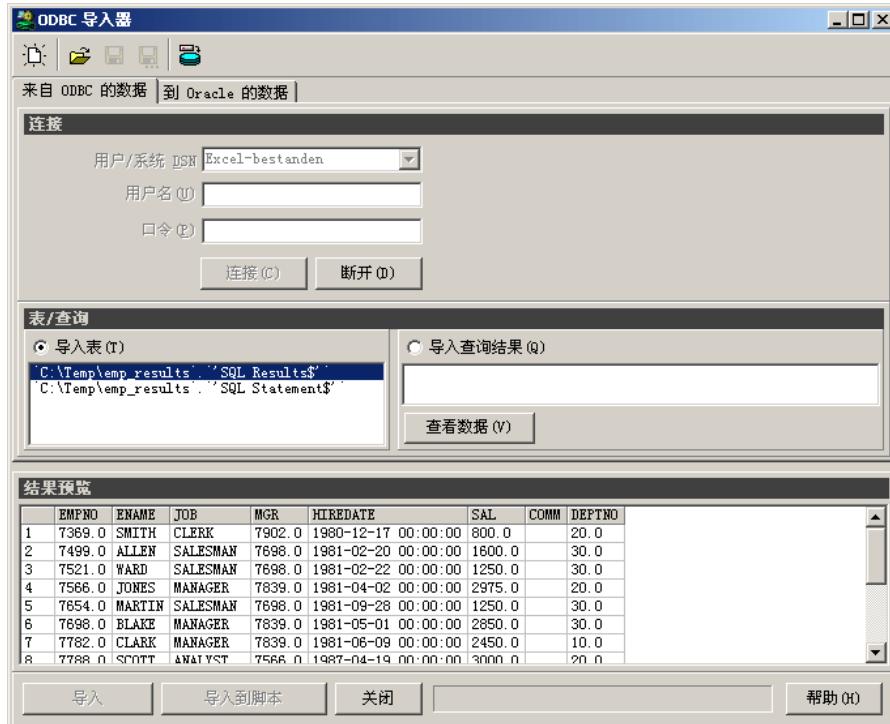
- 字段  
你要与文本字段建立联系的 Oracle 字段。如果你不要求导入这个字段，就可以保留这个字段为空。
- 字段类型  
基本的字段类型：字符串、数字或日期。
- SQL 函数  
这个选项允许你定义附加的 SQL 处理。对于日期字段，`to_date` 函数被自动添加。这是一个很强大的选项，允许你转换导入的数据。基本上，你可以输入能被 Oracle 处理的任何东西。你可以添加一个 # 号来指出数据。“创建 SQL”按钮将为日期字段用 `to_date` 函数来填充这个输入框。

当定义完成时，你可以决定保存它，工具栏中有保存和载入定义文件的按钮。文本导入器将记住哪一个定义文件被用于哪一个文本文件，并且在下一次你打开同样的文本文件时，定义文件将自动被载入。

你有两个导入按钮：“导入”和“导入到脚本”。第一个选项将开始导入数据到已选的表中，第二个选项将创建一个带有插入语句的 SQL 脚本。

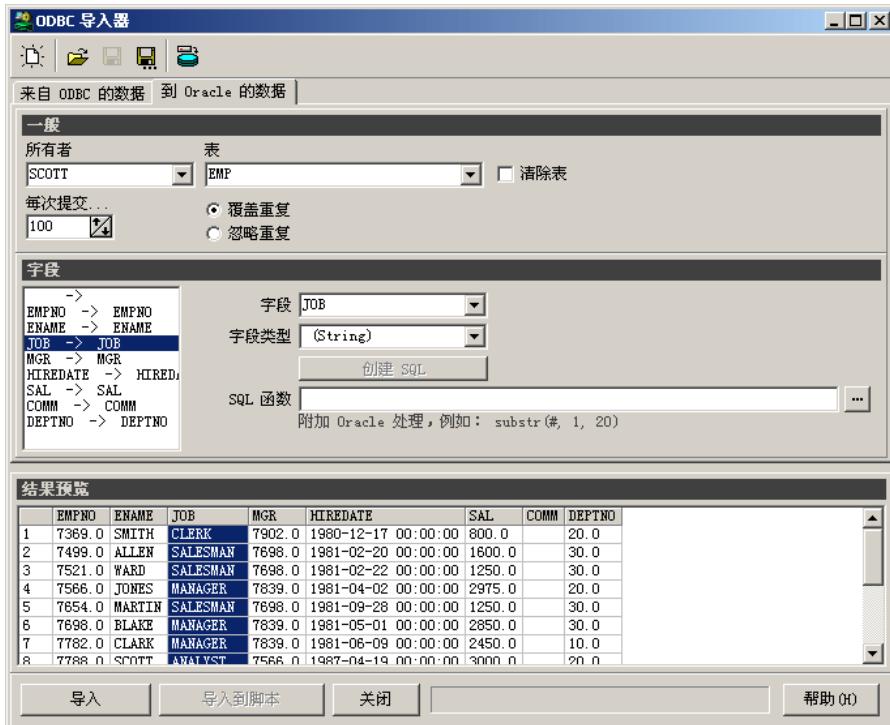
## 21.8 ODBC 导入器

ODBC 导入器工具允许你把来自任何 ODBC 数据源的数据导入到一个 Oracle 表内：



在 ODBC 导入器的第一页上，你需要选择 ODBC 数据源，并且非强制地为这个数据源提供一个用户名和口令。你可以随后链接，之后表列表将被所有的来自数据源的表填充。选择一个表将填充结果预览窗格。代替选择一个表，你还可以通过选择导入查询结果选项并输入一个查询文本来提供一个查询。如果你要限制行数或列数，或者结果来自多个表，这可能是有用的。

在输入了源说明以后，你可以切换到到 Oracle 的数据标签页来选择目标表和指定列映射：



在一般区域中，你可以选择要导入记录的所有者和表，你还可以选择导入前是否要清空表、选择提交时间间隔（0 = 终止提交）和忽略重复行或覆盖它们。

在字段区域中，你可以把 ODBC 数据源（左面）的字段映射到 Oracle 表（右面）的列。对于每一列，你可以另外提供一个 SQL 函数。例如，如果你要把字符串字段转换为大写，就输入 `upper(#)`。散列符号将被用于每条记录的字段数据所替换，并且结果表达式将在插入语句中被提供。

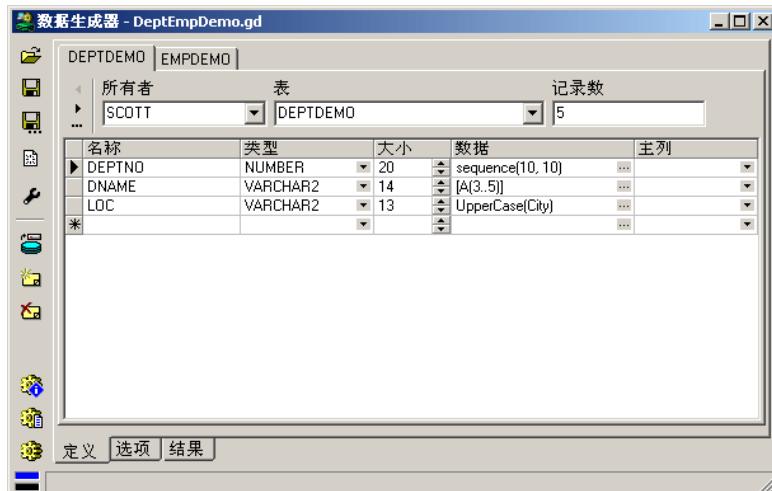
在完成了字段映射之后，你就可以按导入按钮来把数据导入到 Oracle 表中了。如果你在按这个按钮同时又按下了 `Ctrl` 键，一个实例插入语句将显示以便你能校验你指定的字段映射和 SQL 函数。

要在以后再次使用这个导入定义，你可以按下工具栏中的保存定义按钮。以后通过按打开定义按钮，这个定义就能被打开了。

## 21.9 数据生成器

数据生成器允许你创建演示和测试数据。对于测试应用程序和查看它们怎样执行大量的数据，这可能是有帮助的。你可以在工具菜单的下面找到数据生成器。

基本上，定义由一个或多个表、你要生成的记录数量和字段数据定义构成。左上部的按钮允许你打开和保存定义。



在上面，你可以看到被提供的 deptemp 演示，这个演示是为 deptdemo 和 empdemo 表创建数据，这类似于大家都知道的 dept 和 emp 表。在这一页中，你将发现下列项目：

- 表 - 表的名称。
- 记录数 - 你要生成的记录数。这可以是数字，也可以是象 10..100 这样的范围。
- 名称 - 字段名称。
- 类型 - 字段的数据类型。
- 大小 - 字段大小（在适当时）。对于数字字段，这将是数值范围、精度。
- 数据 - 字段数据的定义（见下面）。
- 主列 - 如果这个表是另一个表的明细，你可以设置主列。对于每个已产生的记录，指定数量的明细记录将被生成。

你可以使用“添加现有表”按钮来添加一个现有的表，你还可以从对象浏览器中拖放一个表。

### 数据定义

数据定义决定了被生成的数据。如果你要创建简单的字符，你可以在两个方括号之间输入字符定义：[数据]

数据可以是下列预先确定的集的混合体：

- a: a..z (小写字符)
- A: A..Z (大写字符)
- @: A..Z 和 a..z (所有字符)
- #: A..Z 和 a..z 和 0..9 (所有字符和数字)
- \*: #33..#126 (所有 ASCII 字符)

- 0: 0..9 (*所有数字*)
- 1: 1..9 (*所有除了 0 之外的数字*)
- 9: 0..9 (*所有数字*)

例如：

[Aaa00] 生成字符串如： Gxe21, Liy05 等。

你还可以在两个单引号之间添加文字文本。

例如：

[AA '-' 1000] 生成字符串如： CX-4903 , SY-1044 等。

在定义中空格字符是被忽略的，除非它们在引号中。

如果你要一个字符重复许多次，你可以在字符的后面两个括号之间添加次数(n)。你还可以添加一个随机数(最小..最大)。

例如：

[Aa(5..15) ' ' Aa(8..20)] 的结果就象这样：“Masfae Qwwcdsadif”

定义中的文字文本不必被装入到括号中。换句话说，['hello'] 和 'hello' 是等效的。不带引号被输入的文本被认为是函数。

有几个特定的函数是可用的：

- **Signal(Min, Max, Delta, Noise)**  
返回技术测定数据（象温度这样的）。Min（最小）和 Max（最大）决定了范围，Delta（△）为最大更改量，你还可以添加一些 Noise（噪声）。例如：Signal(-10, 20, 0.1, 0.1)。
- **Random([Min], Max)**  
返回 Min（最小）和 Max（最大）之间的随机数。如果只指定了 Max，那么 Min 就被设为 0。对于日期字段，你可以输入 Min（最小）和 Max（最大）的日期。
- **Sequence(Start, [Inc], [WithinParent])**  
返回序号。Start 为开始值，Inc 为增量（默认为 1）。对于明细表，你可以再指定 WithinParent 关键字来指出序列应该为每个父记录进行重新安排。
- **List('item'(weight), 'item'(weight), ...)**  
随机地返回指定项目(item)之一。加权数(weight)可以被添加到两个括号之间，允许特定项目比其他项目出现的几率更大。  
例如：List('CLERK'(50), 'SALESMAN'(30), 'MANAGER'(10))
- **List(select statement)**  
象前面的 List 函数，但项目是通过 SQL select statement (select 语句) 返回的。你可以使用绑定变量，键入文件名来应用生成的值。  
例：List(select empno from emp where deptno = :deptno)可以为经理创建一个员工部门编号名单。
- **SQL (表达式)**  
返回给定的 SQL 表达式。表达式可以包含标准的 SQL 值、运算符、函数和用户自定义函数。表达式可以每行每列的返回。你还可以使用绑定变量，键入文件名来应用生成的值。  
例：SQL('Department' || :deptno)
- **Text([Style], MaxCharacters, [WordsPerLine, LinesPerParagraph])**  
这个函数返回文本。可选择的 Style (字体) 参数可以是 LorumIpsum (默认) (怀疑是荷

兰语。——译者姜华东注）、英语、德语或日语。它还可以是象 [aA] 这样的字符集，在这种情况下，词汇从指定的字符集中产生。

MaxCharacters 决定了生成的文本的最大大小，WordsPerLine 和 LinesPerParagraph 决定了行和段落的大小。指定的大小可以是指定的数字或范围（最小..最大）。

- **File(path, path, ...)**

从特定的路径 (path) 中随机地选择一个文件，并插入内容。允许你输入二进制数据（象图片这样的）到数据库中。路径可以使用通配符，象 d:\images\\*.bmp 这样的。

你可以通过使用下列函数更改前面函数的文本结果：

- **Uppercase()**
- **LowerCase()**
- **InitCaps()**

例如：InitCaps( List(select ename from emp) )

还有几个预先定义的数据集可用于生成或多或少的真实数据。你可以使用下列定义：

- Firstname - 一般列表中的名
- Lastname - 一般列表中的姓
- Company - 公司名（现有公司的随机列表）
- Address1 - 地址行 1
- Address2 - 地址行 2
- ZIP - 邮政编码
- City - 城市
- State - 省
- Country - 国家
- Email (被关联到 Firstname , Lastname 和 Country)

还有一些可用的实例自定义数据集：

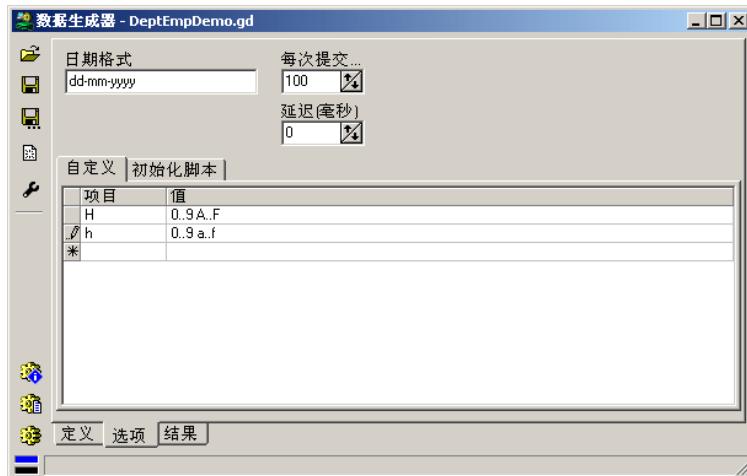
- Components.Code - 一般商品项目：商品代码
- Components.Description - 商品描述（计算机部分）
- Components.Price - 商品价格
- Elements.Name - 化学元素（名称）
- Elements.Symbol - 化学元素（符号）

这些数据集可以在 DataGenerator\UserData 目录中被找到，文件名为 elements.txt 和 components.txt 。如果需要，你可以添加自己的集。添加一个逗号分隔的文件很简单，在文件中第一行两个方括号之间保存着描述。你可以通过指定“文件名. 描述”来使用你的文件中的数据，就象两个例子那样。

上面提及的所有函数和数据都可以添加到一起，例如：Random(10..99) + '-' + [A(4)] 。  
“+”是可选的，但那中至少应该有一个空格作为分隔符。

## 选项

选项标签页允许你设置一些首选项，特别是数据生成器的定义。你可以指定一个日期的格式（与被 Random 函数使用的例子一样）。你还可以指定在多少条记录之后需要提交（设为 0 是一成就提交）。延迟首选项仅仅用于一些实时测试，在这中你需要数据以指定的速度被插入。



自定义部分允许你定义或越过被用于 [数据] 定义的字符集。项目总是单个字符，值是一个或多个空格分隔的字符范围，就象这样：A..Z a..z 0..9 #200..#220。例如，要定义一个十六进制的字符集，你可以指定 H 为项目，0..9 A..F 为值。

“初始化脚本”部分允许你指定在数据被插入到数据库中前应该被执行的 SQL 语句。在典型的情况下，这将是创建表或截短表、选择回滚段等的语句。多个语句需要用分号来分隔。

通过按左边首选项按钮，在默认的情况下，对于所有数据生成器的一般选项可以被定义。要了解更多详细资料，请参见第 20.23 节。如果数据生成器定义中相应的首选项也被设置了，这些首选项将被否决。

## 生成数据

在左下方有三个按钮可用于生成实际的数据：

- 开始测试运行 - 这将在结果标签页上的表格中生成数据和显示结果。通过右键单击表格，你可以以不同的格式导出这些结果。
- 创建数据为 SQL - 生成数据为 SQL 脚本。这仅仅工作于你不用文件功能添加来自文件的数据的情况下。
- 在数据库中创建数据 - 生成数据到 Oracle 数据库中。

## 21.10 比较用户对象

在表定义、视图、程序单元等发生更改之后，把这些更改传递到另外的数据库用户也许是很有用的。这也许是另外一个开发环境或者是一个测试环境。要把你的开发用户对象与另一个用户对象进行比较，你可以使用工具菜单中的**比较用户对象**功能。这将调出下列对话框：



在选择 标签页中选择拥有对象的用户后，可以选择要比较的对象。在做好选择以后，你可以按 目标会话 按钮，选择你要比较的用户和数据库。这将激活 比较 按钮，你可以按它来开始比较操作。

在选项标签页上，你可以设置下列选项：



你可以选择包括存储...选项来包括存储信息，如新对象的表空间名和初始大小。这些可能与数据库有不同的地方，所以这并不总是适用的。

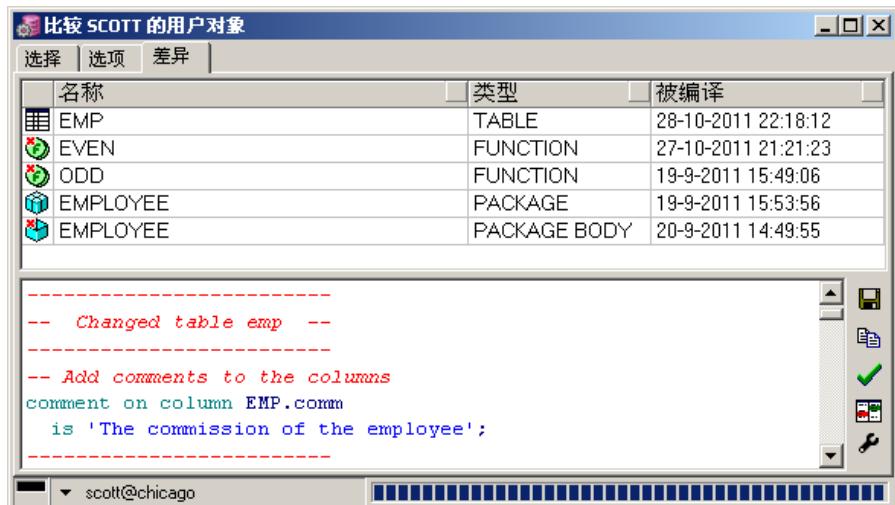
与被链接的目标用户比较，如果目标会话中的对象被别的用户所拥有，那么目标方案就能被设置。

在 PL/SQL 编码中忽略大小写，不会影响到整个编码的执行。但是字符串表达式的区别是会有影响的，这个非常重要。

在 PL/SQL 编码中忽略空格，制表符和换行符不会影响整个编码的执行。但是字符串表达式是会有影响的，这个非常重要。

在 PL/SQL 编码中忽略备注，会忽略所有备注的区别。

当比较操作完成后，对话框将转到差异标签页，它将显示出所有有差异的对象的列表：



这个列表按依赖的顺序被储存。在目标用户不同对象的列表下面，你可以看到需要被执行的 SQL 语句，这些语句能使这些对象与当前用户的相对对象变得一致。如果没有对象被选择，所有对象的 SQL 都被显示。如果一个或多个对象被选择，那么已选对象的 SQL 就被显示。在上面的例子中，用于 EMP 表的 JOB 和 HIREDATE 列的默认值在目标方案中被清除掉了。

显示差异按钮将显示出对象的新旧源文件之间的每个差异。这对于你查看程序单元中产生的更改可能是有用的，或者它可以帮助你来确定对于其他对象类型一个特定的 DDL 语句为什么会被生成。配置外部比较工具允许你配置应该被使用的比较工具。在默认的情况下，ExamDiff 工具将被使用，Pro 版本是可用的（请参见 ExamDiff 的 info 菜单的关于项目）。请参见第 20.22 节。

现在，你可以按应用 SQL 按钮在目标会话中执行这个 SQL 了。你也可以通过按保存 SQL 按钮在一个文件中保存 SQL，或者你可以通过按复制 SQL 按钮把它复制到剪贴板上。

当对象被比较时，下列属性会被忽略：

- 储存 - 对于比较，象表和索引的下一个大小和%空闲这样的属性不被做相应的考虑。

- 带有系统已生成名称的约束 - 对于两个用户，这些约束将有不同的名称，所以它们不能被比较。如果表在目标会话中是新的，这些约束将被生成。
- 表创建属性 - 需要重新创建表的属性被忽略。
- 表数据 - 要导出表数据，请使用导出表功能（请参见第 21.4 节）。
- 序列值 - 序列的当前值是被设想的数据。

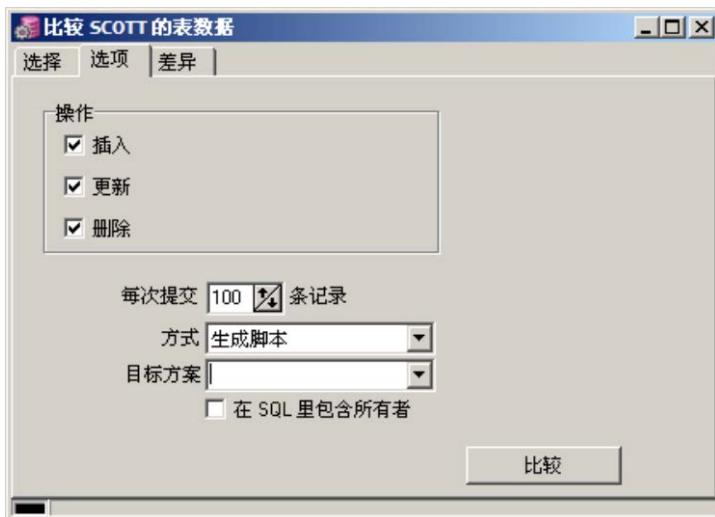
## 21.11 比较表数据

在开发期间，在一个或多个表中插入、更新和删除了一些记录之后，把这些更改传播给被不同的用户拥有的相同的表可能是有用的。这可能是从开发传播到测试数据库，或者产品数据库，或者别的工程成员。

为此，你可以使用来自于工具菜单的比较表数据工具，这将调出下列对话框：



在选择标签页上，你可以选择拥有表的用户以及要比较的表。做了选择之后，你可以按目标会话按钮，选择你要比较的用户和数据库。这将激活比较按钮，你可以按它来开始比较操作，但是你还可以首先转到选项标签页：



在这一页上，你可以为比较过程指定不同的选项。

- **操作** - 你可以选择不同的你要在目标会话中执行的操作。例如，如果你省略了删除操作，那么仅仅插入和更新操作将被执行。
- **每次提交 N 条记录** - 控制了一次提交之后将出现多少条记录。如果你指定为 0，那么仅仅一次提交就出现了这个过程的末尾。
- **方式** - 若选择生成脚本，则使用插入、更新、删除和提交就生成了一个 SQL 脚本。你随后可以保存这个脚本并运行它于命令窗口或 SQL\*Plus。选择更新数据库则在目标会话中立即应用更改。
- **目标方案** - 与目标会话的用户比较，如果表位于不同的方案中，你可以在这中指定这个方案。目标用户将需要有目标方案中的已选表的必要的权限。
- **在 SQL 中包含所有者** - 这个选项仅仅用于脚本模式，并且将在已生成的 SQL 中给表名加上所有者的前缀。这样做在以后你运行脚本时将允许你链接为不同的用户，并且仍然可以在正确的方案中应用更改。

在选择了适当的选项以后，你可以按比较按钮来开始比较过程。在完成之后，差异将会显示：



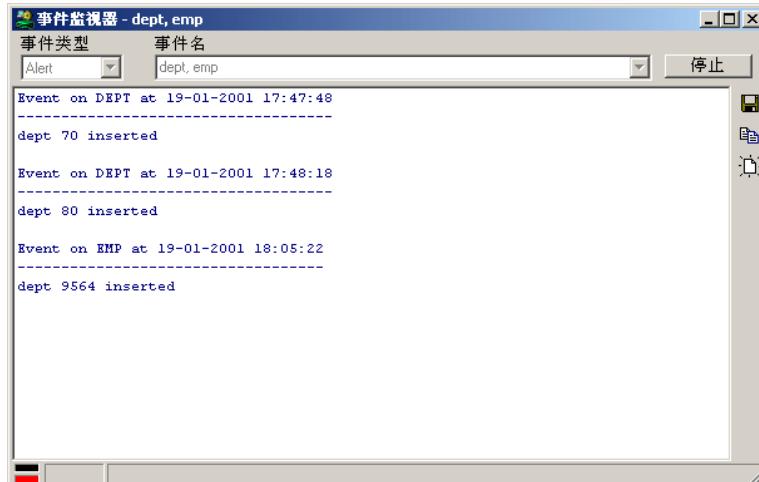
在更新模式下，这个标签页将仅仅显示每个表的被插入、被更新和被删除记录的合计数。

在脚本模式下，你可以检查所有的差异和通过按右边的保存 SQL 到文件按钮来保存作为结果的脚本。注意，通过在上一个窗格中对表做出选择，你可以限制脚本到指定的表。通过按右边的相应的按钮，你还可以复制 SQL 到剪贴板或者在目标会话中应用被选择的更改。

## 21.12 事件监视器

事件监视器允许你捕获和查看被在其他会话中的程序单元生成的 dbms\_pipe 信息或 dbms\_alert 信号。仅仅指定事件类型 (Alert (报警) 或 Pipe (鸣叫)) 、报警和鸣叫的名称，再按开始按钮就可以了。

下列屏幕截图显示了事件监视器在等待名为 ‘DEPT’ 或 ‘EMP’ 的警报：



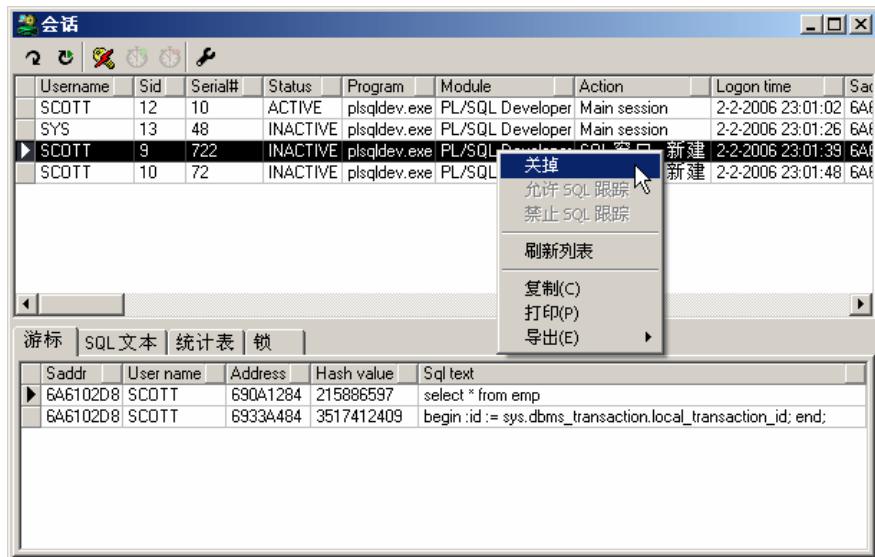
无论这样的事件何时出现，鸣叫或警报的名称和时间戳都被写到输出页中。事件的信息如下：

- 鸣叫：信息的所有项目都显示。
- 报警：dbms\_alert.signal 呼叫的信息参数的值显示。

对于报警事件，你可以指定多个名称，用冒号来分隔。对于鸣叫事件，你可以只指定一个鸣叫名称。

## 21.13 会话

会话工具显示了你当前链接到的数据库实例中所有的会话：



在窗口的上半部分中，你可以看到关于会话的信息，例如用户名、sid 和序列号（用于识别会话）、状态等。你可以通过单击列标题按钮给会话列表排序。例如，要快速查找所有活动的会话，请单击状态列标题按钮。

你还可以选择来自工具栏的会话过滤器来限制会话列表到会话的指定内容，或者限制被显示的列。过滤器还能定义会话和列的顺序。

如果你右键单击一个会话，这将调出一个带有下列项目的弹出式菜单：

- 关掉 - 允许你关掉已选的会话。
- 允许 SQL 跟踪 - 允许 SQL 跟踪已选的会话。如果你的 Oracle 服务器版本不支持这个特性，或者如果你无权使用 dbms\_system 包，这个选项将被禁用。
- 禁用 SQL 跟踪 - 禁用 SQL 跟踪已选的会话。
- 刷新列表 - 刷新会话列表。
- 复制 - 复制会话列表到剪贴板。
- 打印 - 打印会话列表。
- 导出 - 用 CSV、TSV、HTML 或 XML 格式将会话列表导出为文件或导出到剪贴板。

刷新、关掉、允许 SQL 跟踪和禁用 SQL 跟踪功能通过窗口顶部的工具栏也是可访问的。工具栏还包括自动刷新按钮。当它被按时，会话列表和会话信息将定时被更新。你可以通过右键单击按钮来定义刷新周期。

窗口的下半部分包含了关于已选会话信息的几个标签页。通过右键单击列表并从弹出式菜单选择相应项目，这些标签页上的信息将可以被刷新、复制或打印。要了解更多关于显示在列表中的单独列的信息，请参见“Oracle 服务器参考”指南。

对于会话列表和会话详细资料标签页，你可以创建、修改或删除会话查询。为此，请按下工具栏中的`定义会话查询`按钮。这将调出下列对话框：



在顶部，你可以在过滤器和详细资料查询之间切换。过滤器控制了在会话列表中显示什么，详细资料控制了在详细资料标签页上显示什么。

在右边，你可以看到四个按钮：创建新查询、删除查询或在列表中上移、下移查询。当你创建或修改查询时，你必须提供下列信息：

### 过滤器

**标题** 在工具栏中的过滤器选择列表中可以显露出来。

**DB 版本** 是最小的数据库版本，此过滤器适用于该版本。

**查询** 是提取用于主会话列表的会话信息的 select 语句，你可以使用这个查询来：

1. 限制被显示的会话（比如仅有效会话）。
2. 定义会话的显示顺序。
3. 定义哪些列以什么顺序显示出来。

### 详细资料

**标题** 在详细资料标签页的标签上显露出来。列表中的位置决定了会话窗口中的标签位置。

**DB 版本** 是最小的数据库版本，此会话细节适用于该版本。

查询是提取会话信息的 SQL select 语句。这个 select 语句通常应该包含 :sid 绑定变量，当查询被执行时它将包含已选会话的 sid。你可以使用 v\$session 视图的列中的任何一列，例如 :username , :sql\_address 或 :sql\_hash\_value 。如果你的查询导致运行时间错误，它将在执行后相应的标签页上显示出来。

要链接一个会话详细查询的多个行的值，你可以在 SQL 文本中使用 /\* concatenate \*/ 提示。例如：

```
select sql_text from v$sqltext_with_newlines
where address = hextoraw(:sql_address)
and hash_value = :sql_hash_value
order by piece
/* concatenate */
```

所有的被这个查询返回的行的 *sql\_text* 列都会被链接，并且在会话详细资料栅格中被显示为一个值。

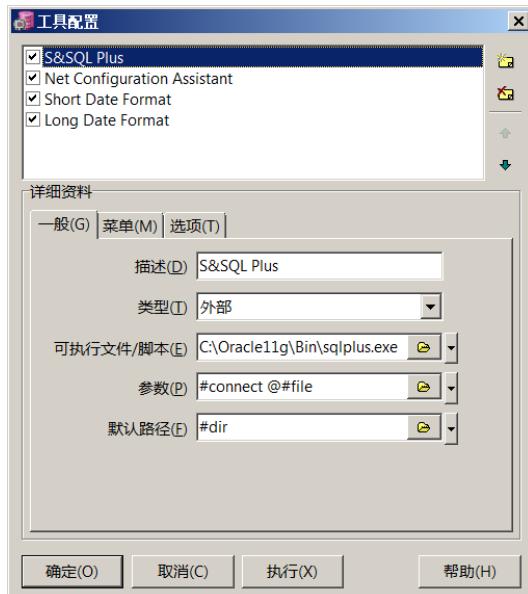
## 21.14 自定义工具

你可以定义你自己的集成在 PL/SQL Developer IDE 中的工具。这些工具可以被添加为菜单项目，用于为当前会话在数据库中启动外部的应用程序或执行 SQL 或 PL/SQL 脚本。你可以配置参数，所以对于应用程序你可以做到忽略文件名和链接字符串。

这样做的一个很好的例子是 SQL\*Plus 。你可以添加一个菜单来启动 SQL\*Plus 并且带有适当的配置，这样你能让 SQL\*Plus 自动登录，甚至让它执行当前打开的文件。

另外一个是为 NLS 设置一套特定的值的工具。你可以创建一个脚本来执行正确的“警报会话”语句，并且添加这个脚本到菜单中，在菜单中它可以被调用用于 SQL 窗口、测试窗口等的会话。

要配置外部工具，你就必须选择工具菜单中的配置工具菜单项目。下列对话框将显示出来：



右边的四个按钮允许你插入和删除项目，还可以在列表中上移或下移项目。如果你在按插入按钮时按下了 Ctrl 键，新的已创建的项目将从当前的选择中被复制。

执行按钮可以被用于执行已选的工具。如果这个按钮被选择时你按下了 Ctrl 键，一条信息将弹出来，它包含了关于应该执行什么的（带有替换参数的）信息。

列表中显示出了所有的已配置的工具，下半部分显示了已选工具的配置。配置被分为四部分：

- 一般 - 定义被运行的可执行文件/脚本和它的参数。
- 菜单 - 定义相应的菜单项目的显示和位置。
- 选项 - 一些附加的工具设置。

要解释外部工具配置，我们将把添加一个 SQL\*Plus 菜单到 PL/SQL Developer 作为一个例子来说明问题。

## 一般标签页

在你用插入按钮创建了一个新项目之后，第一件要做的事情是定义工具类型。如果你选择了外部，工具将被作为一个外部应用程序来运行。这对于 SQL\*Plus 来说是可用的。如果你选择了会话，当工具被调用时，工具将运行当前会话的 SQL 脚本。

其次，你要在一般标签页上输入描述。描述是将被显示在菜单和列表中的名称。你可以在要作为快捷键的字符前面输入一个 & 符号 (S&QL\*Plus 会变成 SQL\*Plus)。如果你输入了 - 符号作为描述，一个分隔行将显示。

第三个至关重要的事情是用于执行的程序或脚本的名称。对于外部工具，你可以输入任何可执行文件甚至是希望的文档，对于文档与其关联的应用程序将被启动。浏览按钮打开了一个文

件对话框，允许你查找文件。大多数 Oracle 工具都位于 Oracle 的 bin 目录中，SQL\*Plus 可以被发现于 <oracle\_home>\bin\sqlplusw.exe 。

要使 SQL\*Plus 能从菜单中工作，这三个步骤已经足够了。另外，你可以定义一些参数传递给应用程序，还可以定义默认路径。你可以在默认路径和完整路径之间选择一个来用于可执行文件或文件参数。对于 SQL\*Plus，你可以添加一个链接字符串 (#connect) 和一个要执行的文件的链接 (#file) 。

右边带有下箭头的小按钮允许你挑选一个变量，这个变量能被插入到输入框的任何地方。这些变量都与当前链接和已打开的文件有关系。当工具运行时，它们将被相应的信息替换。下列变量是被支持的：

变量	含义
#file	当前窗口中不带路径的文件名
#path	带路径的文件名
#dir	当前窗口中文文件的目录
#object	已选的浏览器对象（如 SCOTT.EMP）
#otype	已选的浏览器对象类型（如 TABLE）
#owner	对象所有者（SCOTT）
#oname	对象名（EMP）
#connect	当前 PL/SQL Developer 链接的完整链接字符串（如 scott/tiger@demo）
#username	主链接的用户名
#password	主链接的口令
#database	主链接的数据库
#wconnect	当前窗口链接的完整链接字符串
#wusername	当前窗口链接的用户名
#wpassword	当前窗口链接的口令
#wdatabase	当前窗口链接的数据库

SQL\*Plus 不喜欢可能以空格为文件参数的完整路径，它被关掉了，这就是为什么 #dir 被指定为默认路径的原因，这样仅仅文件名必须被传递到 SQL\*Plus 。

## 会话工具

对于会话工具，你要指定一个能包含多个 SQL 语句或 PL/SQL 块的 SQL 脚本。SQL 语句能被分号或斜杠分隔开，PL/SQL 块必须用斜杠来终止，这与 SQL\*Plus 的语法一样。

例 1 - Germany.sql:

```
alter session set nls_language = 'german';
alter session set nls_territory = 'germany';
```

例 2 - SpecialRole.sql:

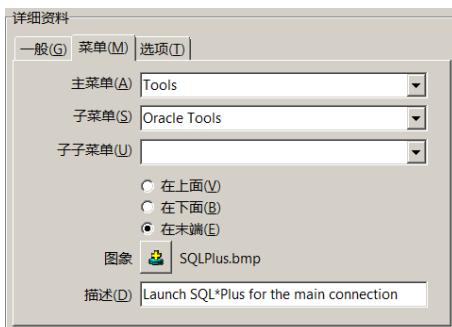
```
begin
  dbms_session.set_role(role_cmd => 'special_role identified by &Password');
end;
/
```

注意，你可以添加置换变量到脚本中以使它变得更为灵活。当工具被调用时，你会得到提示要输入变量的值。当上面的 SpecialRole 脚本被调用时，你将被提示要求输入一个口令。

如果你要使用任何一个脚本中指定的参数，那么你就必须使用 &1、&2 等。数字指出了命令行上参数的顺序。你将不会得到这些命令行置换变量的提示。

## 菜单标签页

在默认的情况下，所有被配置的工具都被加到工具菜单的底部。如果你要在别处甚至它们自己的主菜单中创建它们，你就可以使用菜单标签页：



如果你要求你的工具菜单项目被创建到特定的主菜单中，你可以从主菜单编辑列表中选择一个项目。或者如果你要创建一个新的主菜单，那么你就可以输入一个新的菜单名。

如果你已经输入了一个主菜单，那么你就可以使用子菜单和子子菜单来指出你要创建的新菜单项目的准确位置。

你可以选择三个位置按钮之一（在下面、在上面和在末端），指定新菜单项目在指定菜单中的相对位置。如果你指定了一个新的（子）菜单，你应该始终选择在末端，因为在下面和在上面仅在引用了现有（子）菜单的情况下有意义。

例 1:

主菜单	工具
子菜单	Oracle 工具
子子菜单	
位置	在末端

这将在工具主菜单中创建一个 Oracle 工具子菜单。如果你正在添加多个 Oracle 相关工具，那么设一个特定的子菜单可能是一个好主意。

例 2:

主菜单	Oracle
子菜单	
子子菜单	
位置	在末端

这将创建一个 Oracle 主菜单。如果你正在有规律地使用一些工具并且你要求它们“在附近”，那么你大概应该只好创建一个新的主菜单。

如果你要添加你的工具为工具菜单中的第一个项目，那么你应该输入：

主菜单	工具
子菜单	首选项…
子子菜单	

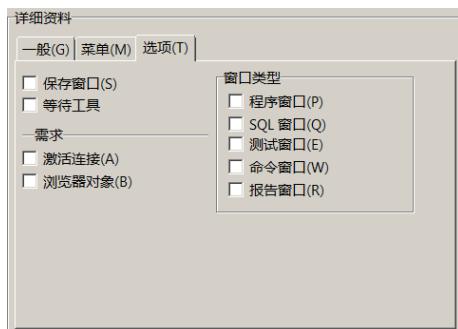
位置	在上面
----	-----

这将在现有的首选项菜单项目上面创建一个菜单。

按图标按钮来为工具选择一个 Windows 位图文件 (\*.bmp)。这个图象的大小应该是 16 x 16 像素比较合适。注意，PL/SQL Developer 将始终从原始位置载入位图文件，因此若相应的外部工具不更改你就不要删除或重新命名这个文件。PL/SQL Developer 带有许多标准的位图文件，你可以从中挑选。这些文件位于 PL/SQL Developer 安装目录中的 Icons 子目录中。这是图片选择器默认的目录。将鼠标光标悬停在工具栏按钮上方时将显示提示。如果你没有指定提示，则显示工具的描述（如菜单中所显示的那样）。

## 选项标签页

选项标签页允许你进行下列设置：



- **保存窗口**

当这个选项被设置时，活动窗口将在工具被执行前被保存。当工具获得作为参数的文件并且你要确信它使用当前数据时，你大概应该做保存窗口这件事。

- **等待工具**

若启用此选项，PL/SQL Developer 将等待工具完成。如果存在修改源文件的外部程序，这项功能可能十分有用。在使用此选项调用工具时，当前窗口的内容将刷新为修改后的源文件。通常与 **保存窗口** 选项一起使用。

- **激活链接**

如果工具正好用于要求 PL/SQL Developer 被链接（如果你传递作为参数的链接字符串），你就应该设置激活链接了。这将在 PL/SQL Developer 没有被链接时禁用使用菜单。

- **浏览器对象**

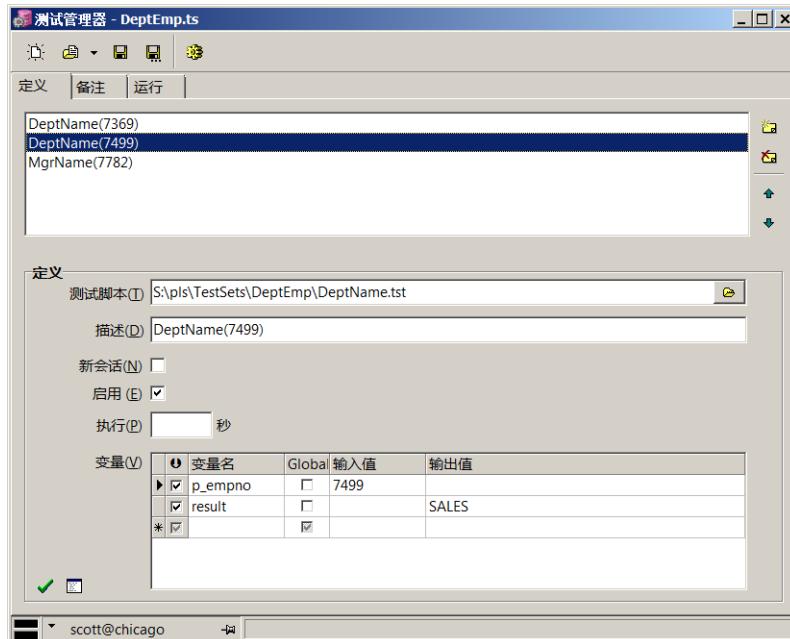
如果工具工作在对象浏览器中已选的对象上，就设置这个选项。

- **窗口类型**

如果工具需要一个特定的窗口，你可以在这部分中指定这些选项。如果一个或多个选项被选择，如果活动窗口类型被指定，菜单就会被激活。

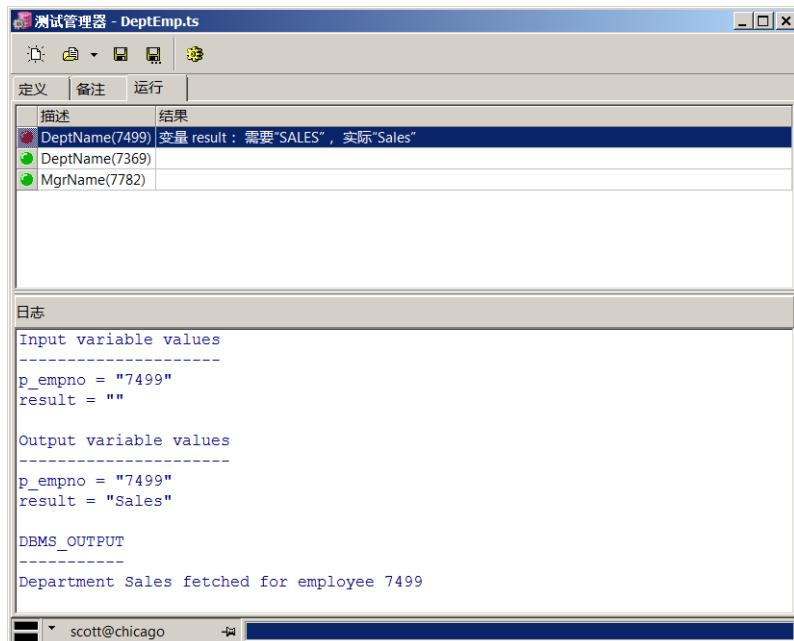
## 21.15 测试管理器

要为你的 Oracle 被储存的程序单元执行回归测试，你就可以使用测试管理器。它允许你定义一个测试集，这个测试集是带有输入变量值和必需的输出的测试脚本的集合。另外，它还允许你指定必需的特性：



对于这个例子，测试集有三个测试脚本组成。对于被选择的 *DeptName.tst* 脚本，*p\_empno* 变量值将获得输入值 7499，执行后 *result* 变量值一定是 *SALES*。

运行测试集将快速地展现出被测试的程序单元是否仍然可以正确地运行并且（或者）带有必需的特性：



所有失败的测试脚本将有一红色的指示并且会被放在顶部。在这种情况下，*DeptName(7499)* 脚本失败了，因为规则要求结果是“SALES”而不是实际的“Sales”值。

结果窗口之下的日志窗口会显示每个变量的输入值和输出值，若选定的测试脚本生成了 dbms\_output，也会显示在日志窗格。

## 创建测试集

要创建一个测试集，请选择来自于工具菜单的**测试管理器**项目。一个空的测试管理器屏幕就显示出来了，你可以在此创建新的测试脚本或者添加现有的测试脚本。在测试管理器的上部，你可以看到测试脚本的列表，在这中你可以添加脚本到集中、从集中删除脚本或者更改脚本的顺序。在下部，你可以看到被选择的脚本的定义。

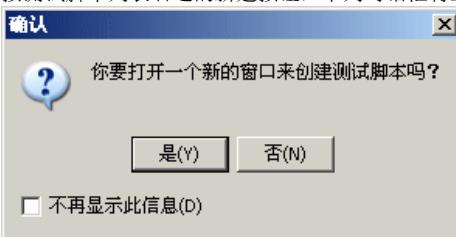
在注释标签页，可以输入描述或其他相关的信息，以便说明测试集。

在创建新的测试脚本之前，通过按下工具栏中的**保存测试集**按钮来保存测试集是一个好主意。所有的新的测试脚本都会被创建在与默认测试集相同的目录中，而对于每个测试集使用分开的目录可能是一个好主意，以便所有相关测试的有用东西能从这个单独的目录中被管理起来。

## 添加测试脚本到测试集

有两种方法可以把一个测试脚本添加到测试集：

1. 把一个过程或函数从对象浏览器拖放到测试管理器中，一个新的测试窗口将被打开，它会调出这个程序单元和用于所有参数和返回值的变量。在一个描述名下保存这些测试脚本，最好与测试集在同样的目录中。
2. 按测试脚本列表右边的新建按钮，下列对话框将显示出来：



单击是创建一个新的、空的测试脚本。你将需要用手工来编测试脚本程序，在它可用于测试集之前把它保存起来。

如果你要添加一个现有的测试脚本到测试集，就单击否。

在测试脚本被保存之后，你可以在测试集中指定定义：

- **测试脚本** - 测试脚本的文件名。如果你已经创建和保存了一个新的测试脚本，这个名就已经被填充在内了。如果你要添加一个现有的测试脚本，请按这个区域右边的浏览按钮。
- **描述** - 脚本的功能描述。如果你要多次运行有不同的输入变量值的单个脚本，你最好就在描述中包含这个，以便它在列表中能被快速地辨认出来。
- **新的会话** - 使用该选项可以强建一个新的会话脚本，这个会话脚本与前一个脚本是相互独立的。使用同样的会话可以默认所有脚本模式。
- **允许** - 使用这个选项能允许或禁用脚本用于测试集。仅仅被允许的脚本会被包括进测试运行中，被禁用的脚本将被跳过。
- **执行** - 指定必需的执行的最大秒数（允许有小数）。如果你允许这个区域为空，执行将不被测试。
- **变量** - 对于测试脚本中的每个变量，输入它们的名称、输入值和必需的输出值。输入值将在执行前被传递到测试脚本，必需的输出值将在执行后被测试。如果你允许输出值为空，它将不被测试。要测试空值，请输入单词“null”。此外，可以使用一个 like 表达式（例如，LIKE ‘%’）或者集合表达式（例如，IN (10, 20, 30)）来测试输出值。

如果一个变量被标记为全局，那么这个全局值就会被传递到其他的测试脚本中。如果一个全局变量没有输入值，那么这个全局值就可以被使用。执行命令后，这个全局值也会发生一些相应的变化。

要测试 dbms\_output，请输入用于变量名的“dbms\_output”，并且要为输出值输入必需的文本。

在左面靠下的地方，你能看到两个按钮，它允许你快速地运行和检查被选择的测试脚本，或者在用于编辑的测试窗口中打开它。

### 复制测试脚本

在很常见的情况下，你需要用不同的输入和输出变量值来运行相同的测试脚本，要适应这种情况，你可以在测试集中选择一个现有的测试脚本，并且在按新建按钮的同时按下 *Ctrl* 键。新创的测试集项目将继承来自被选择项目的所有属性，以致于你只需更改描述和变量值即可。

## 运行测试集

要运行一个测试集，请按下工具栏中的运行按钮，测试管理器将切换到运行标签页，每个被激活的测试脚本将在各自的测试窗口中被执行。失败将被用一个红色的指示器指示出来，并且会被放在列表的顶部。成功的脚本将有一个绿色的指示器，并且将被放在列表的底部。

要调查失败的原因，请右键单击脚本并且从弹出式菜单选择调试项目，或者双击脚本，带有测试脚本的测试窗口将被创建，输入变量值将会因此而被设置。现在你就可以调试测试脚本来确定引起错误的原因了。

右键单击测试运行结果将复制它们到剪贴板、将打印它们或者以不同的格式导出它们。

## 通过命令行运行测试集

还可以通过命令行窗口中的命令行运行测试集。例如：

```
SQL> RunTestSet U:\TestSets\DeptEmp.ts D:\Temp\DeptEmp.log
```

这将运行 *U:\TestSets\DeptEmp.ts* 测试集，并在 *D:\Temp\DeptEmp.log* 处创建一个日志文件。若忽略日志文件名，则将创建与测试集同名的日志文件。

日志文件为每个测试脚本包含三行内容：

```
Test = <Description of the Test Script>
Result = Success | Failed | Error | Slow | Skipped
Message = <Message of the Test Script run>
```

这将允许你创建一个命令文件，执行一个或多个测试集。如下脚本将运行 2 个测试集，随后退出应用程序：

```
RunTestSet U:\TestSets\DeptEmp.ts D:\Temp\DeptEmp.log
RunTestSet U:\TestSets\Sales.ts D:\Temp\Sales.log
Exit Application
```

此脚本随后可通过 PL/SQL Developer 命令行调用。例如：

```
plsqldev.exe UserID=scott/tiger@chicago CommandFile=U:\TestSets\RunTests.sql
```

## 22. 编辑器

允许你编辑一些 SQL 或 PL/SQL 源的每个窗口都使用同一个编辑器。这个编辑器有很多能使程序员工作起来更轻松的特性。

### 22.1 选择功能

编辑器允许你在已选的文本中执行各种功能，这些可用的功能来自于编辑菜单中的选择项目，只适用于你已经选择了的文本。你可以通过键配置首选项给这些功能分配一个功能键。

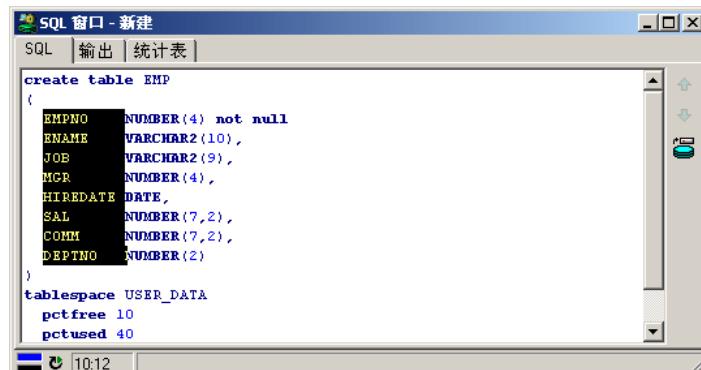
- 缩进 / 撤消缩进  
缩进字符的数量依赖于编辑器首选项。如果文本的选择存在，*Tab* 和 *Shift-Tab* 键总是被指派到这两个功能上。
- 删除尾随空格  
除了属于多行字符串的行之外，删除所有选定航中的尾随空格和标签。
- 小写/大写

将选定文本转换为小写或大写形式，单引号之间的文本除外。

- 注释/撤消注释  
使用 /\* 和 \*/ 可以快速地注释一个选择。
- 应用语法大小写  
如果你已经把编辑器关键字大小写设置为大写、小写或首字母大写，那么在已选文本中这个功能将应用这种样式到所有的关键字。
- 排序  
选择的行将按升序来排序。如果你再次调用排序功能，选择将按降序来排序。
- 颜色标记  
应用当前颜色标记到选择内容（请参阅第 22.12 节）。

### 22.2 列选择

当你做一个选择时，通过按下 *Alt* 键，你就可以选择一个文本列了：

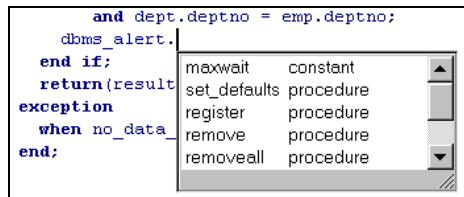


```
SQL 窗口 - 新建
SQL | 输出 | 统计表 |
create table EMP
(
    EMPNO NUMBER(4) not null,
    ENAME VARCHAR2(10),
    JOB VARCHAR2(9),
    MGR NUMBER(4),
    HIREDATE DATE,
    SAL NUMBER(7,2),
    COMM NUMBER(7,2),
    DEPTNO NUMBER(2)
)
tablespace USER_DATA
  pctfree 10
  pctused 40
```

你通常可以对这个选择进行复制、粘贴、剪切、删除、大写、小写、（撤消）缩进和（撤消）注释等操作。如果你使用应用语法大小写功能并且关键字横过列边，它将不被处理。

## 22.3 代码助手

代码助手是一个很强大的工具，在你键入 SQL 或 PL/SQL 代码之时为你提供帮助。例如，当你键入 `dbms_alert` 并且稍微犹豫一小会儿时，编辑器将显示出 `dbms_alert` 包中的元素列表：



当你继续键入 `dbms_alert` 过程的名称时，列表中的内容将被简化。例如，如果你在 `dbms_alert.` 后面键入一个 `r`，那么就只有 `register`、`remove` 和 `removeall` 过程将仍然出现在列表中。当你在代码助手可见时按回车键时，当前选择的项目名称就被插入到源中了。你还可以在代码助手窗口中使用箭头键来选择需要的项目。按代码助手左下方的排序按钮可以对项目进行排序或取消排序。按 `ESC` 键将移除代码助手。

代码助手将为下列对象类型提供帮助：

对象类型	元素
包	过程、函数、类型、变量、常数和异常
函数、过程	参数、带有指定符号的插入 ( <code>param =&gt;</code> )
表、视图	列
序列	<code>nextval</code> 和 <code>currval</code>
用户	被用户拥有的对象
游标变量	游标的 <code>select</code> 语句的字段
记录类型变量	记录类型的字段
表%rowtype 变量	表的列
对象类型	属性 (对于默认构造器)
对象类型变量	属性和方法
对象类型列	属性和方法
集合变量	集合方法 ( <code>first</code> , <code>next</code> , <code>limit</code> , 等)
<code>:new</code> 和 <code>:old</code>	当前触发器的表或视图列
自身	当前类型的属性和方法
<code>@</code>	数据库链接清单

代码助手还可以帮你键入对象的名称（表、包等）、关键字和在当前上下文内部有意义的 PL/SQL 标识符。例如，如果你在 SQL 编辑器中键入 `pro` 并犹豫一下，关键字 `procedure` 和 `profile` 将被代码助手显示出来，连同开头为 `pro` 的任何数据库对象名一起。上下文依赖于当前连接的用户（它决定了可用的对象名）、编辑器类型（它决定了可用的关键字）和你要编辑的程序单元（它决定了可用的 PL/SQL 标识符）。如果你是第一次做一个连接到数据库的链接，数据库对象名称是包含在提取的内容清单里面的。执行的查询在 PL/SQL Developer 安装根

目录或者%APPDATA%\PLSQL Developer 根目录里的 CANames.sql 文件中。你可以根据需求改写这些查询。

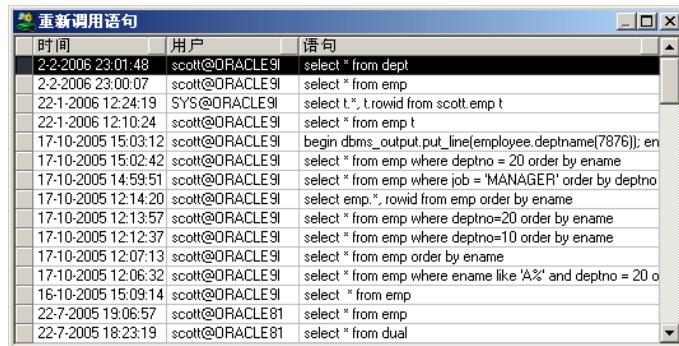
代码助手可以自动/手工地被调用，在键入一个对象名后代码助手被自动激活前的延迟时间可以通过首选项来定义。此外，你可以定义你是否要描述一个用户，如果是，你要求被包括进去哪些对象类型。上下文描述和需要被键入的字符的最小数量在它被激活前应配置好。

对于列列表和参数列表，在列表顶部有一个额外的项目，它允许你一次性添加所有列或参数。你也可以使用 *Ctrl* 键和 *Shift* 键来选择多个项目。首选项控制这些项目何时放在一行，何时放在多行，以及如何对齐参数。

所有这些首选项都已在第 20.14 节中描述。

## 22.4 重新调用语句

无论你何时成功地在 SQL 窗口、命令窗口或报告窗口中执行了一个语句，这个语句都被储存到全局语句重新调用缓冲区中。通过选择来自于编辑菜单的重新调用语句项目，或者通过按 *Ctrl-E* 按钮，你可以在编辑器中重新调用最近执行的语句。这将调出一个象这样的选择列表：



语句按执行顺序显示，最近一个首先显示。在列表中，你可以看到时间、用户和（部分）语句文本。现在，你可以选择一个语句，双击它（或按回车键）把它插入到列表被调用的编辑器中光标所在的位置。可以使用 *Ctrl* 键或 *Shift* 键来选择多条语句。

通过按列表标题中相应的按钮，你可以按时间、用户或语句文本的顺序来给列表排序。

注意，出于安全原因，包含密码的语句不能放置在全局语句回调缓冲区中。

如果右键单击语句列表，将会看到以下弹出菜单：



- **所有用户 / 当前用户** – 显示当前用户或所有用户的语句。
- **查找语句** – 允许你搜索语句。查找功能自当前选定的语句后启动，若找不到匹配则继续从第一条语句开始。也可以通过按 *Ctrl-F* 键调用此功能。
- **复制语句** – 将选定语句的文本复制到剪贴板。.
- **删除语句** – 删除选定的语句。
- **查看语句** – 在文本编辑器中显示选定语句的文本。当查看长语句或多条语句时，这非常有用。
- **展开选定 / 所有** – 展开列表中的语句，使之由多行组成。
- **折叠选定 / 所有** – 在使用展开功能后，将列表折叠为每语句一行。
- **导出** – 允许你将语句列表导出到 CSV 格式的文件。

重新调用缓冲区可以容纳 200 个语句，这些语句都将被储存到你的 Windows 概要文件的应用数据目录中。通过设置首选项，你可以更改这个声明的限制和目录（请参见第 20.25 节）。

## 22.5 专用复制

如果你正在用 PL/SQL Developer 写 SQL 和 PL/SQL 代码，随后你又要在其他工具中使用代码，例如象 3GL 这样的程序设计语言，那么你可能需要把这些代码转换为稍微不同的格式。让我们假设你已经在 PL/SQL Developer 中写了并测试了这样一个 SQL 语句：

```
select deptno, sum(sal) mgr_sal from emp
where job = 'MANAGER'
group by deptno
order by mgr_sal desc
```

例如，如果你要在 Borland Delphi 中使用这个语句，你可能需要象这样的格式：

```
SQL := 'select deptno, sum(sal) mgr_sal from emp' + #13#10 +
      'where job = ''MANAGER''' + #13#10 +
      'group by deptno' + #13#10 +
      'order by mgr_sal desc';
```

为了这个目的，在右键单击了编辑器中的一个选择后，你可以使用来自于编辑菜单或来自于弹出式菜单的专用复制功能。这个功能有一个子菜单，它显示了所有被定义的专用复制格式。在选择了格式之后，被转换的代码就被储存在剪贴板上了，这样你就可以在相应工具的编辑器中粘贴它了。

专用复制格式被定义在 PL/SQL Developer 安装目录下的 SpecialCopy 子目录中。你可以更改预先确定的复制格式或者添加新的复制格式。仅仅简单地添加一个带有 .copy 扩展名的文本文档即可，它包含了一个针对 PL/SQL 代码第一行的变量 (<line\_1>)、一个针对 PL/SQL 代码最后一行的变量 (<line\_N>) 和一个针对所有其他行的变量 (<line\_\*>)。下面是一个针对 Borland Delphi 的例子：

```
;PL/SQL Developer SpecialCopy definition for Borland Delphi
;<line_1> for first line
;<line_*> for all other lines
;<line_N> for last line
;
SQL := '<line_1>' + #13#10 +
      '<line_*>' + #13#10 +
      '<line_n>';
```

第一行需要为指派到 SQL 的变量加上前言，接下来需要有一个 CR/LF 对。最后一行不需要有 CR/LF 对，但需要用分号来终止。所有其他行仅仅需要 CR/LF 接在后面。

如果 <line\_1> 和 <line\_n> 都与 <line\_\*> 一样，你可以忽略它们。

在一些语言中，你需要对特定的字符使用换码序列。例如，在 C++ 中，你要对 tab 字符 (ASCII 码为 9) 使用 \t。要定义这些换码序列，请使用 #define 关键字：

```
#define char(9) = \t
#define \ = \\
String("<line_1>\n") +
String("<line_*>\n") +
String("<line_n>");
```

你还可以使用 “#define compress” 来指出你要从结果中删除所有多余的空字符（空格、制表符和换行）。

注意，.copy 文件的名称将被包括在菜单中，所以你应该使用描述性的文件名。

## 22.6 从主机语言粘贴

编辑菜单中的这个菜单项提供了与专用复制恰好相反的功能。从 C、Visual Basic 或 Pascal 等主机语言中复制一段包含 SQL 语句的源代码之后，可以使用此功能来粘贴实际 SQL 语句。举例来说，如果你将以下 Visual Basic 代码复制到了剪贴板中：

```
SQL = "select empno, sal" & vbCr & -
      " from emp" & vbCr & -
      " where deptno = 10"
```

此时调用从主机语言粘贴功能将粘贴实际 SQL 语句：

```
select empno, sal
  from emp
where deptno = 10
```

## 22.7 上下文敏感帮助

大多数程序员经常访问他们正在程序中使用的功能的参考信息。要尽可能快地找到这些参考信息，请把光标放在源文本中的关键字上面，再按 *F1* 就能搜索 Oracle 在线手册中的这些关键字。

要了解关于 PL/SQL Developer 帮助系统的更多信息，你可以阅读第 32 章。

## 22.8 数据库对象弹出式菜单

当你的程序访问了一个数据库对象时，你经常需要从这个对象中获得一些信息，或者你需要处理它。你可能需要知道它的描述或它的属性。对于表或视图，你可能需要查询或编辑数据；对于函数或过程，你或许需要测试某一设想。

如果你右键单击编辑器中的对象名，你将能看到与你右键单击对象浏览器中的对象名时看到的一样的弹出式菜单。因此，所有针对这个对象的相关功能将在编辑器中立即可用。

## 22.9 解释计划

要为编辑器中的 SQL 语句调用解释计划窗口，请选择语句的文本，右键单击它，再从弹出式菜单选择 **解释计划** 项目。如果没有文本被选择，整个源都会被带到解释计划窗口中。如果 SQL 语句包含了 PL/SQL 变量，你必须在解释计划窗口中更改语句，就象第 6.1 节中描述的那样。

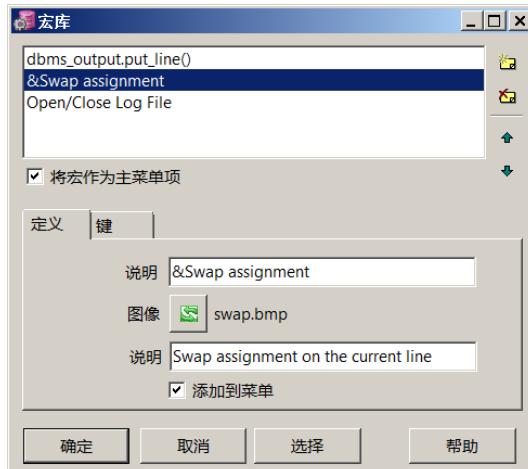
## 22.10 宏

编辑器有一个强大的宏功能，允许进行快速和容易的宏记录和宏回放，这可用于自动执行一个特定的重复的任务。

被记录的宏能被存储于宏库中，以便以后运行。库中的宏通常将是一些反复执行的重复任务（例如交换赋值：`a := b; -> b := a;`），或者可能包含一些代码片断。例如，当你可以一次性地键入游标类型并可以记录下来留着以后使用时，为什么还要不断地查寻一个游标类型应该怎样被声明呢？

要记录一个宏，请按 *F11* 或选择工具菜单中的宏菜单中的 **记录** 项目。现在，工具栏中显示出了一个闪烁的记录器图标，它表示你当前正在记录一个宏。所有的键盘单击都被记录下来留待以后回放，这意味着在宏被记录期间你不应该使用鼠标。再次按 *F11*，宏记录就被终止了。按 *F12* 将回放宏。

要在宏库中储存当前记录的宏，请选择宏菜单中的`/库`项目。这将打开宏库对话框：



按新建按钮可以添加一个带有当前记录定义的新的宏。你可以为宏输入描述并指出它是否应该被包括到宏菜单中。这样你可以容易地生成一些可用的宏，并且你还可以通过键配置首选项给每个宏指派一个功能键。如果作为主菜单项目的宏选项被检验，那么宏菜单就被定位到主菜单中。如果没有被检验，它将位于工具菜单的下面。

描述和图像将显示在菜单中。按图像按钮来为宏按钮选择一个 Windows 位图文件 (\*.bmp)。这个图象的大小应该是 16 x 16 像素比较合适。注意，PL/SQL Developer 将始终从原始位置载入位图文件，因此若相应的宏不更改你就不要删除或重新命名这个文件。PL/SQL Developer 带有许多标准的位图文件，你可以从那中挑选它们。这些文件位于 PL/SQL Developer 安装目录中的 Icons 子目录中。这是图片选择器默认的目录。

将鼠标光标悬停在工具栏按钮上方时将显示提示。如果你没有指定提示，则显示工具的描述（如库和菜单中所显示的那样）。

使用添加到菜单选项可控制是否能通过菜单使用宏。

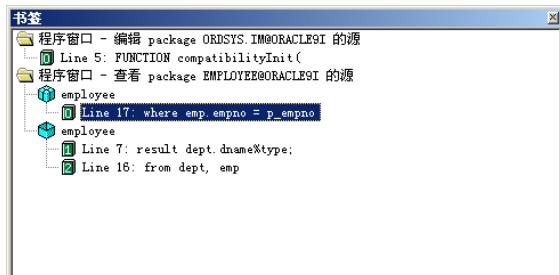
在键标签页上，你可以为宏查看和更改键盘的作用。

要执行一个已存储的宏，请打开宏库并双击它的名称。现在，已选的宏可以通过按 `F12` 被执行了。当你在菜单中选择宏时，它将立即被执行，并且可以通过按 `F12` 重新被执行。

## 22.11 书签

要标记编辑器中的一个位置，你可以添加一个书签，简单地按 `Ctrl-Kn` (`n` 为 0 .. 9 的数字) 就可以在当前位置添加一个书签，一个绿色的书签将在页边空白处显示出来，并带有相应的号码。要在以后定位这个书签，你可以按 `Ctrl-Qn`。你也也可以使用来自于编辑菜单的设置书签和转到书签子菜单。

编辑菜单的 书签列表项目将调出所有编辑器书签的全局分层显示列表：



在这个树形视图中，你可以看到标签被定位的窗口、编辑器和行。单击一个书签将把窗口和编辑器调到前面，将定位那个书签。你可以停放书签使它一直可以访问，不会妨碍任何其他窗口。

## 22.12 颜色标记

你可以在编辑器中添加颜色标记以使指定部分高亮显示。为此，请选择你要标记的文本并且按下工具栏中的 颜色标记 按钮：



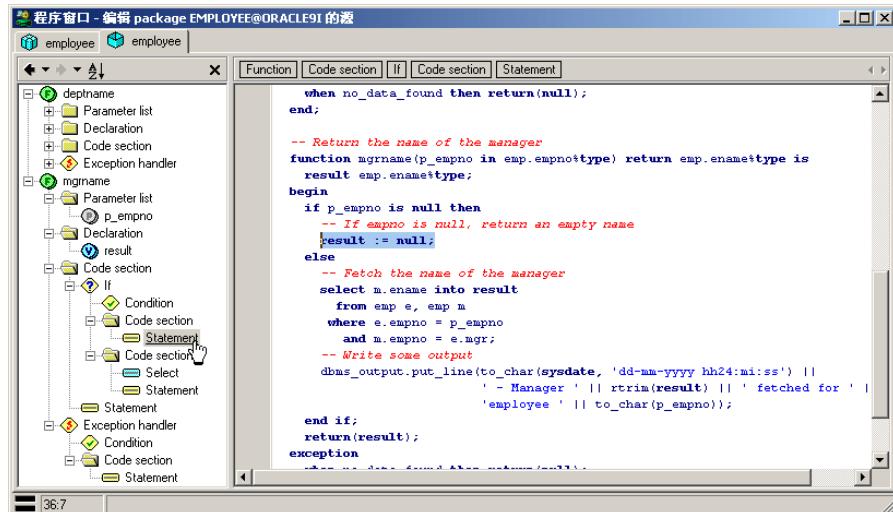
被选择文本的背景颜色将变为当前的标记颜色：

```
-- Fetch the name of the manager
select m.ename into result
from emp e, emp m
where e.empno = xa
and m.empno = e.mgr;
```

要更改标记颜色，请按紧跟在按钮之后的箭头并选择一种新的颜色。要清除颜色标记，请把光标放在被标记文本的内部，不做任何选择，再按 颜色标记 按钮。

## 22.13 代码目录

代码目录特性被限制到程序编辑器中，它显示了程序单元的完整结构。这在大的包体和类型体中是最有用的，它可以跨越数以千计的 PL/SQL 代码行。代码目录被显示在程序窗口左边的目录树中：



代码目录明了你在源代码中的当前位置。当你操作经过源代码时，已选项目将指出你的准确位置。

如果在代码目录中你在一个项目上面移动鼠标，编辑器中相应的源代码将象上面的插图中的一样被高亮显示。

在代码目录中单击一个项目将立即导向它的声明。双击项目将展开或折叠它。右键单击项目将显示出一个弹出式菜单，它允许你在编辑器中选择相应的源代码，允许你剪切或复制它到剪贴板上，允许你从剪贴板中粘贴它，允许你把它转换为一个注释，允许你展开所有的子项目，或者允许你为它创建一个测试脚本（如果它是公共的程序单元的话）。

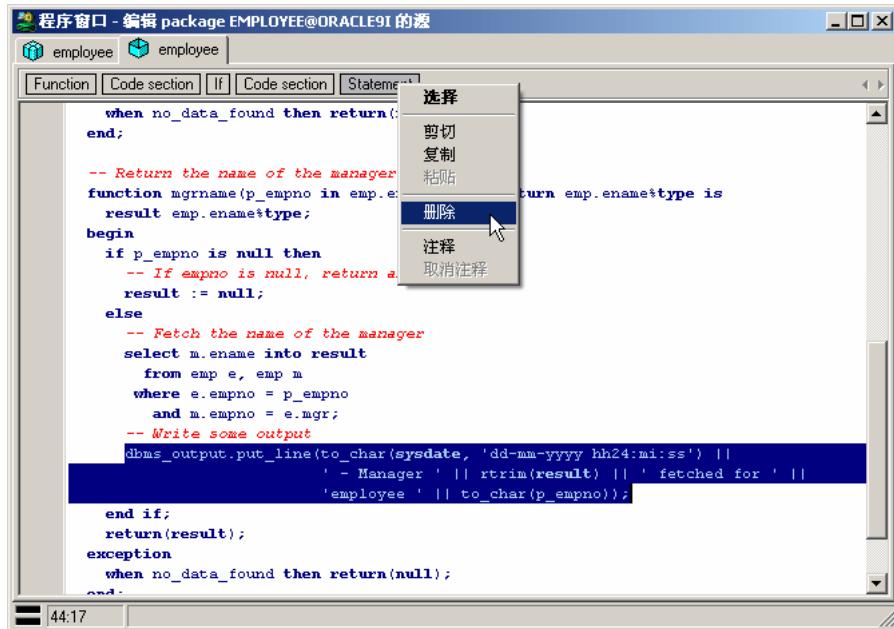
通过按目录窗格顶部的排序目录按钮，目录可以按字母顺序被排序。仅仅主目录将被排序，参数、本地变量等将不被排序。按钮将蹲伏在那中，再按一次按钮将把目录恢复到它们的原始顺序。

要关闭代码目录，你可以单击右上角的关闭按钮。你还可以在编辑器中右键单击并选择目录项目来使它隐藏或显示，或者选择工具菜单中的代码目录项目。

在默认的情况下，代码目录是可见的。要更改这个默认，你可以关闭当前程序窗口中的代码目录并且选择窗口菜单中的保存版面项目。

## 22.14 代码层次

代码层次被限制到程序编辑器中，并在光标位置显示 PL/SQL 代码的层次结构。它位于编辑器的上方：



在上面的例子中，光标被定位于一个 **函数** 内的 **代码部分** 内的 **If** 语句内的 **代码部分** 内的 **Select** 语句上。如果你在代码层次的项目上面移动鼠标，那么相应的源代码将被高亮显示。单击一个项目将选择相应的源代码。

如果你右键单击一个项目（与上面的插图中的一样），一个弹出式菜单将显示，它允许你选择源代码、剪切或复制源代码到剪贴板上、在项目上粘贴剪贴板中的内容、删除源代码、转换源代码到注释或者把注释转换为源代码。

## 22.15 代码折叠

代码折叠允许你显示或隐藏 PL/SQL 代码的特定部分。例如，可以折叠一个包内的所有过程和函数，只展开一个你感兴趣的过程。再如，你可以折叠循环和 if/then/else 语句等结构，获得较长的程序单元内的更好概览。下面的例子折叠了多个函数以及一个 if 语句：

```

-- If you get a PLS-00452 compilation error here, just remove
-- the calls to dbms_output.put_line
function deptname(p_empno in emp.empno%type) return dept.dname%type is ...
begin
    -- Return the name of the manager
    function mgrname(p_empno in emp.empno%type) return emp.ename%type;
    result emp.ename%type;
    begin
        if p_empno is null then ...
        return(result);
    exception
        when no_data_found then return(null);
    end;

    function fetchrecord return boolean is ...
begin

```

要折叠一个代码部分，只需单击编辑器的左端的减号。要展开此部分，单击左端的加号即可。也可以右键单击左边边缘并从弹出式菜单中选择全部折叠或全部展开。如果将鼠标光标移动到编辑器内的一个折叠的行上，折叠的部分将临时显示在单独的窗格中。

## 22.16 拆分编辑

要编辑或查看同一源代码的两个不同部分，可以使用拆分编辑功能。使用鼠标即可将拆分线上下拖动：

```

and dept.deptno = emp.deptno;
end if;
return(result);
exception
-- If the employee does not exist, return an empty name
when no_data_found then return(null);
end;

result emp.ename%type;
begin
if p_empno is null then ...
return(result);
exception
when no_data_found then return(null);
end;

function fetchrecord return boolean is ...
begin

```

现在即可同时浏览、查看和编辑两个部分。

## 22.17 超链接导航

在你的 PL/SQL 代码中，你常需要在别处定义的元素做参考。例如，你可以参考同样的包体内、或同样的包的说明内、或另一个包内的一个 PL/SQL 类型，去找同样的函数或过程调用、变量、视图等。

要从一个程序窗口内容易地找到一个声明，你可以在这样一个参考上面移动鼠标指针，按下 Ctrl 键，再按鼠标左按钮。在按下 Ctrl 键后，在鼠标指针下面的标识符将变为淡蓝和有下划线，类似于 HTML 文档中的我们熟悉的超链接：

```
-- Created : 17-12-1999 17:19:26
-- Purpose : Lock an employee record
procedure lockrecord(p_empno in emp.empno%type) is
    dummy varchar2(1);
begin
    if p_empno != g_lastrecord.empno then
        select null into dummy from emp
        where empno = p_empno
        for update;
    end if;
end lockrecord;
```

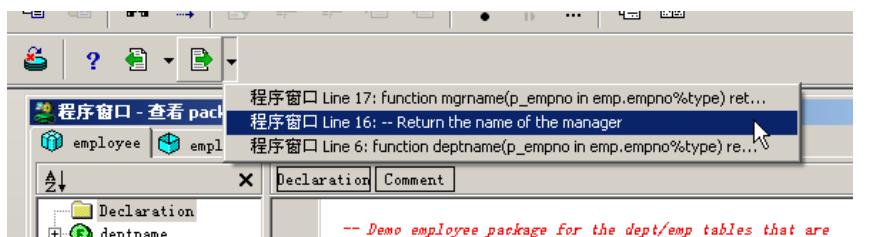
如果声明位于同样的源内，PL/SQL Developer 将导向那中。在那之后，同样的程序窗口内的剩余的程序单元将被搜索。然后，它将搜索所有其他的当前打开的程序窗口。最后，它将为包含声明和将导向那中的对象设法打开一个新的程序窗口或 SQL 窗口。

在默认的情况下，如果适当的话，超链接导航将在包或类型说明中定位一个声明。要打开包或类型体以取代说明，你可以右键单击超链接。

如果超链接导航打开了一个新窗口，则该窗口将是可更新的。要导航到只读窗口，请禁用对应的首选项（请参见第 20.12 节）。

## 22.18 导航按钮

无论你何时跳到窗口中的一个来自编辑器内部的绝对位置，PL/SQL Developer 都将为早先的位置生成一个书签。这些书签可以用主工具栏中的向后导航和向前导航按钮而被访问：



当你在代码目录中航行，使用查找功能，单击一个编译错误，转到编辑器的顶部或底部、使用超链接导航、打开一个新窗口等时，书签将被添加。

## 22.19 重构

重构功能允许你快速地重新组织你的 PL/SQL 代码，它工作于已选择的代码，或者 - 如果没做选择 - 它工作于当前的语句。右键单击一个语句或选择内容在相应的子菜单中提供了下列的重构功能：

- 重命名项目  
重命名当前的变量、参数、常数或程序单元，声明和用法都会被重新命名。
- 提取过程  
如果一个程序单元变得太大或太复杂，你就可以做一个选择并转换它到一个分开的过程。

所有的用于选择内容内部和外部的变量都将被转换为参数。所有的只用于选择内容内部的变量都会从当前的程序单元中被移动到新的程序单元中，选择内容将被一个新的程序单元调用所替换。

- 提取本地常数

如果某一表达式应该被转换为一个本地常数，你就应该选择它并给它提供一个常数名。本地常数将在当前的子程序中被创建，类型由表达式决定。在当前的子程序中，所有表达式的具体取值都将被常数名替换。

- 提取全局常数

如果某一表达式应该被转换为一个全局常数，你就应该选择它并给它提供一个常数名。全局常数将在当前的包内部被创建，类型由表达式决定。在当前的包中，所有表达式的具体取值都将被常数名替换。

- 用初始化代替赋值

如果一个本地变量赋值纯粹用于初始化，你就可以把它移动到变量的声明部分。你可以右键单击一个语句或者选择多个语句。

如果重构操作的结果不令人满意，你可以简单地重做它。

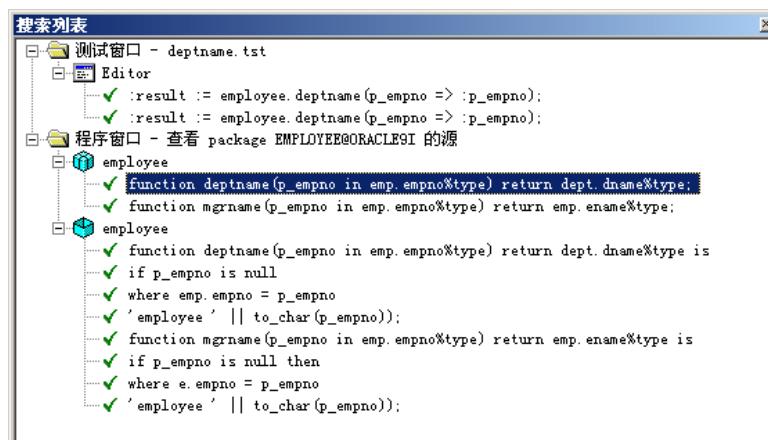
## 22.20 搜索栏

编辑菜单的搜索栏非常有用，可作为一个浮动工具显示，或者被停放在工作区的顶部或底部。在它被停放的位置中，它能被永久地用于执行跨越多个编辑器、多个窗口和多个目录的搜索：



在输入了一个搜索词之后，你可以按 *搜索* 按钮或者按 *Enter* 键。搜索结果将在编辑器中被高亮显示，并且你可以按 *转到下/上一个具体取值* 来定位搜索结果。你可以按 *Esc* 键将焦点从搜索栏移动到编辑器中，按 *隐藏具体取值* 按钮清除编辑器中高亮显示的搜索结果。

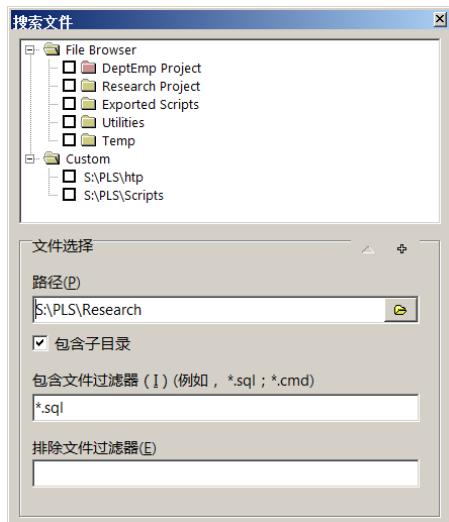
*搜索列表* 按钮将调出搜索结果的分层显示列表：



在这个树形视图中，你可以看到搜索结果中找到的文件、窗口、编辑器和行。单击搜索结果将把窗口和编辑器调到前面、将定位搜索结果和将选择它。你可以停放搜索列表使它一直是可访问的，这不妨碍任何其他窗口。

在搜索栏中，你还可以启用一些选项来搜索所有编辑器或仅当前编辑器、进行目录搜索、进行区分大小写搜索、进行全字匹配搜索和使用正则表达式搜索。

单击靠近文件中搜索按钮的箭头按钮，可以选择在哪里进行搜索。窗口显示如下：

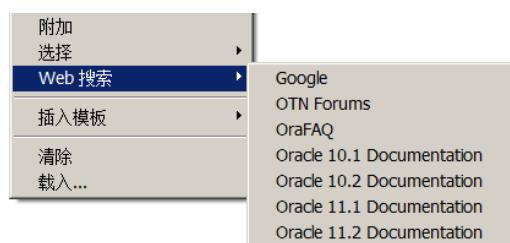


在顶部，可以从文件浏览器中选择一个或多个文件位置（参见第 19 章），或者选择一个或多个以前保存过的自定义搜索。在底部，可以显式输入文件选择标准。关闭此窗口后，选定文件将被包含在下一次搜索中，注意要选中文件中搜索复选框。

通过按 + 按钮，可以将当前文件选择标准保存为自定义选择。按 ^ 按钮以更新当前自定义选择。

## 22.21 Web 搜索

要在 Web 上搜索编辑器内当前选定内容，可以选择编辑菜单的 Web 搜索子菜单，或者右键单击当前选定内容，并从弹出菜单中选择 Web 搜索子菜单。该子菜单将显示已经配置好的多个 Web 搜索：



单击子菜单项即可在 Web 上搜索编辑器内的当前选定文字，并在外部 Web 浏览器或内部 HTML 窗口中显示搜索结果。

## 配置 Web 搜索

你可以配置 Web 搜索，为此，请单击工具菜单的 *配置 Web 搜索* 项。此时将显示如下屏幕：



上半部分显示了已经配置的 Web 搜索。右侧的按钮可用于新建 Web 搜索、删除 Web 搜索以及上下移动 Web 搜索。

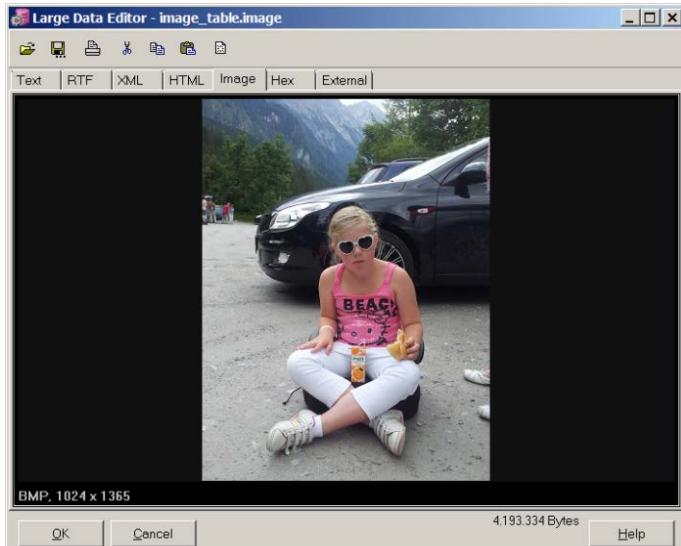
下半部分可以查看和修改选定 Web 搜索：

- **描述** – 描述将显示在菜单中。
- **URL** – 在选择该项 Web 搜索时将打开的 URL。`%s` 表达式将被编辑器中当前选定的内容所取代。
- **显示结果** – 控制结果的显示位置：在外部 Web 浏览器中显示（系统的默认浏览器）还是在内部 HTML 窗口中显示。
- **重用窗口** – 启用此项时，结果将显示在先前为此 Web 搜索打开的 HTML 窗口中。若使用的是外部 Web 浏览器，此选项将不可用。

## 23. 大数据编辑器

只要你需要查看或编辑 long 、 long raw 、 CLOB 、 BLOB 或 BFILE 列或者变量值，大数据编辑器都将被调用。它还可用于 varchar2 数据。大数据编辑器将分析数据并尝试在正确的格式中显示它。它可以显示的值如文本、 RTF 、 XML 、 HTML 、图像 (bmp 、 jpg 、 gif 、 tiff 和其他格式) 和十六进制数据。此外，它还可以运行外部应用程序（例如 MS Word 或 Acrobat Reader ）来查看或编辑数据。

为图像调用大数据编辑器将得出下列屏幕中的结果：



编辑器顶部的工具栏包括下列按钮：

- 打开 - 打开文件内容到编辑器中。
- 保存 - 保存编辑器内容到一个文件中。
- 打印 - 按当前已选格式打印编辑器中的内容。
- 剪切 - 按当前已选格式把编辑器中的内容放到剪贴板上，随后清除 (null) 编辑器中的内容。
- 复制 - 按当前已选格式把编辑器中的内容放到剪贴板上。
- 粘贴 - 按当前已选格式把剪贴板上的内容粘贴到编辑器中。
- 清除 - 清除 (null) 编辑器中的内容。

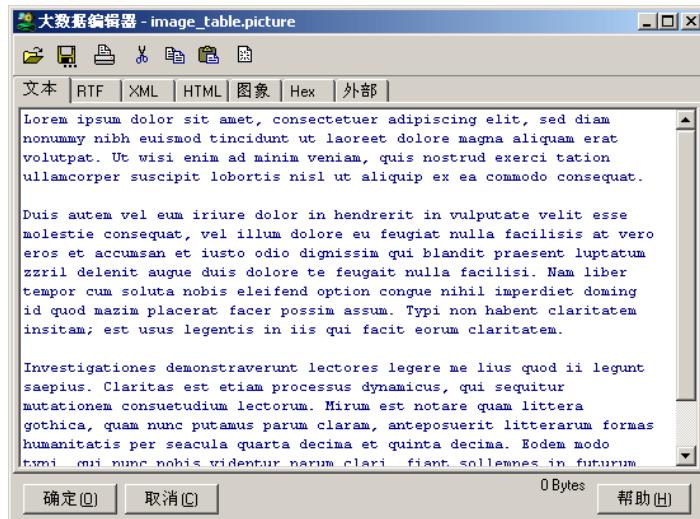
在工具栏的下面，你可以为编辑器中内容的正确格式选择标签页。对于一些数据，有多种格式是可以用的。例如， HTML 文档可以以 HTML 格式或以文本格式被查看。所有工具都可以按十六进制格式 (二进制格式) 被查看。

当特定格式的编辑器内容发生更改时，你可以立即转到一个不同的格式来查看这些更改。例如，当查看一个 HTML 文档时，你可以转到文本格式来产生更改，再转到 HTML 格式来查看这些更改。

在按下了确定按钮之后，你的更改将被传递到来自大数据编辑器被调用的地方的功能中。

## 23.1 编辑纯文本

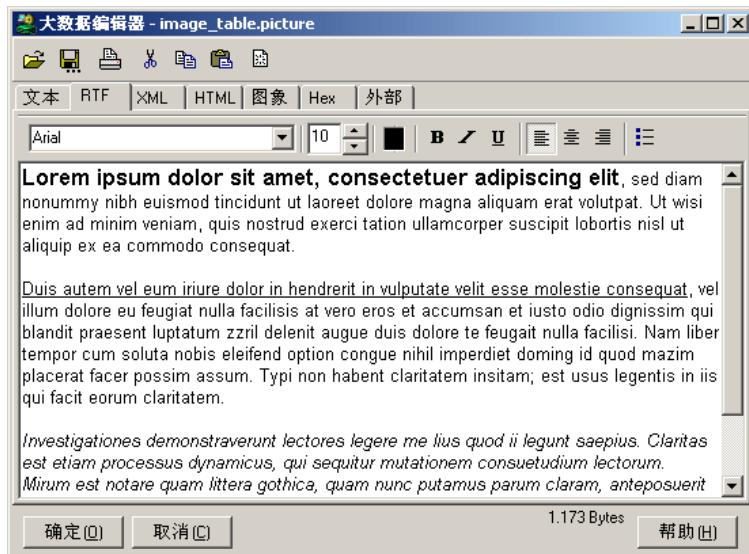
文本编辑器可以被用于查看或编辑任何种类纯文本数据：



它还可以被用于编辑 XML 和 HTML 文本。转到 XML 或 HTML 标签页将立即显示这些更改的结果。

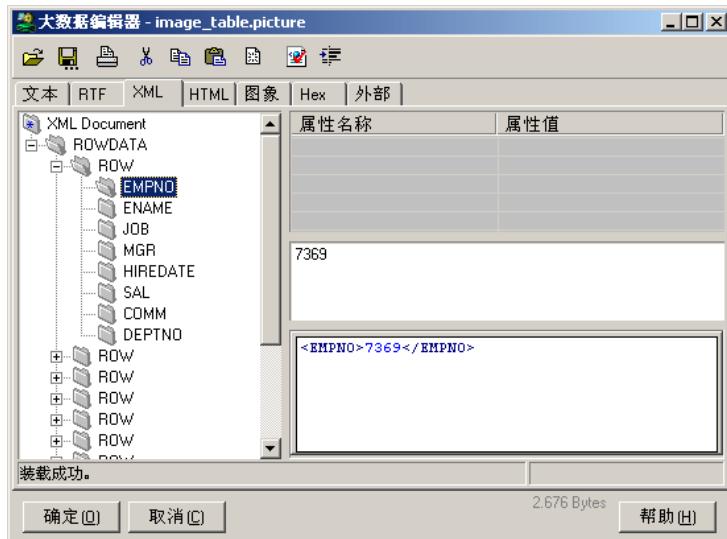
## 23.2 编辑 RTF

RTF 编辑器可以被用于查看和编辑富文本格式的文本。它允许你容易地更改字体类型、大小、颜色、样式和对齐：



## 23.3 编辑 XML

当查看 XML 内容时，编辑器首先转到 XML 标签页：



在这中，你可以看到树形视图中的 XML 文档的结构，可以编辑各个元素和属性。

按解析 XML 按钮将解析 XML 文档，并将报告任何错误。这需要你有 Net8 8.1 或更新版本，并且用户可以访问到 SYS.XMLTYPE 类型或 XMLPARSER 包。

你可以按下工具栏中的格式化 XML 按钮在文档中应用标准的格式标准。

如果你为 XML 文档转到纯文本格式的标签页，那么适当的语法高亮将被应用。



The screenshot shows the PL/SQL Developer Data Editor window titled "大数据编辑器 - image\_table.picture". The window has a toolbar with icons for file operations like Open, Save, Print, and Paste. Below the toolbar is a menu bar with "文本" selected, followed by "RTF", "XML", "HTML", "图象", "Hex", and "外部". The main area contains XML code for the EMPLOYEES table:

```
<?xml version="1.0" ?>
<ROWDATA>

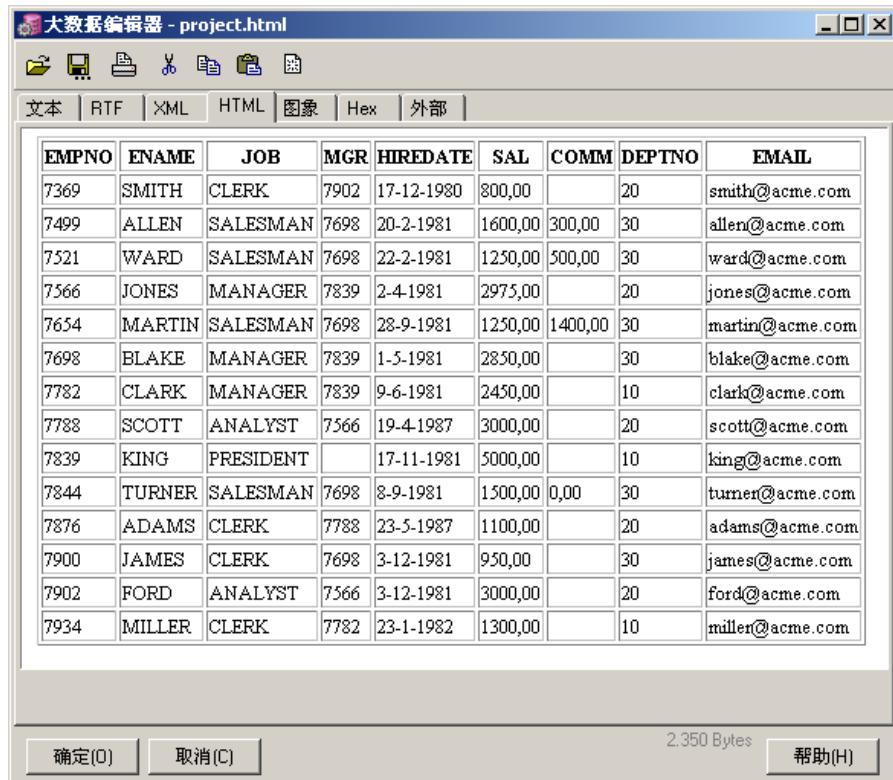
<ROW>
    <EMPNO>7369</EMPNO>
    <ENAME>SMITH</ENAME>
    <JOB>CLERK</JOB>
    <MGR>7902</MGR>
    <HIREDATE>17-12-1980</HIREDATE>
    <SAL>800,00</SAL>
    <COMM></COMM>
    <DEPTNO>20</DEPTNO>
</ROW>

<ROW>
    <EMPNO>7499</EMPNO>
    <ENAME>ALLEN</ENAME>
    <JOB>SALESMAN</JOB>
    <MGR>7698</MGR>
```

At the bottom right of the editor, it says "2.676 Bytes". At the bottom left are buttons for "确定(D)" (Accept), "取消(C)" (Cancel), and "帮助(H)" (Help).

## 23.4 编辑HTML

查看 HTML 内容时，编辑器首先会转换到 HTML 标签页面



整个视图都是只读型的。但是你也可以将它转换到文本标签页面，在纯文本格式中查看、编辑文本数据，完成编辑后，你又可以把它转换回 HTML 标签页面来查看格式化的布局版面。

## 23.5 编辑图象

图象编辑器将显示最流行格式的图象，例如 JPG 、 BMP 、 GIF 等。如果图象比编辑器能显示的图象大，那么这个图象将被调整大小以适应编辑器。这不会影响图象的内容，因为它仅仅是一个视觉效应。在图象的下面，你可以看到图象的格式和图象的像素大小。

要更改一个图象，你既可以按下工具栏中的打开按钮从文件中调用它，又可以从另外一个源复制图象到剪贴板上并按粘贴按钮。注意，当经由剪贴板复制/粘贴图象时，图象格式将更改到 BMP 格式！要保证原始格式被保持住，你应该从一个文件中载入它。

## 23.6 编辑十六进制数据

要查看或编辑编辑器中十六进制格式的内容，你可以转到十六进制标签页：



在左边，你可以看到带有十六进制数据的词（16 位）的 8 个列。在右边，你可以看到十六进制数据的文本内容。你既可以编辑十六进制的内容又可以编辑文本的内容。

如果你右键单击十六进制编辑器，那么你可以选择一个不同的列的版面。

## 23.7 调用外部查看器或编辑器

对于很多被储存在数据库中的大对象，调用外部查看器或编辑器可能是必要的。例如，如果一个 BLOB 列包含了 Acrobat PDF 文件（轻便文档文件）的内容，它将必须调用 Acrobat Reader。当为这样的内容调用了大数据编辑器之后，你将首先被带到文本或十六进制标签页。在那之后，你可以转到外部标签页：



在这中，你可以定义哪一个程序应该被执行。首先，你需要定义文件的扩展名。在执行外部程序之前，带有编辑器内容的临时文件将被带有那个扩展名来保存。其次，你需要定义哪一个程序应该被执行。在那之后，你可以按执行按钮来查看外部程序中的内容。如果你在外部程序中更改了文件并保存了它，大数据编辑器将注意到这些更改并且把这些更改导回到编辑器中。

当查看列数据时，表/列与外部程序二者的关系将被记住，以便你下次查看它。另外，如果你激活下次自动执行选项，那么大数据编辑器将立即调用外部工具。要越过这个自动行为，当调用编辑器时按下 *Ctrl* 键即可。

大多数程序已经注册了它们的文件类型，以取代手工定义扩展名和程序，你可以简单地从列表中选择它。首先你需要按显示已注册文件类型按钮，然后你可以搜索正确的文件类型并双击它。对于 PDF 文件，这将是 *Adobe Acrobat* 文件：



如果你右键单击文件类型列表，你可以按扩展名或描述（默认）的顺序来排序。

## 24. 查询设计器

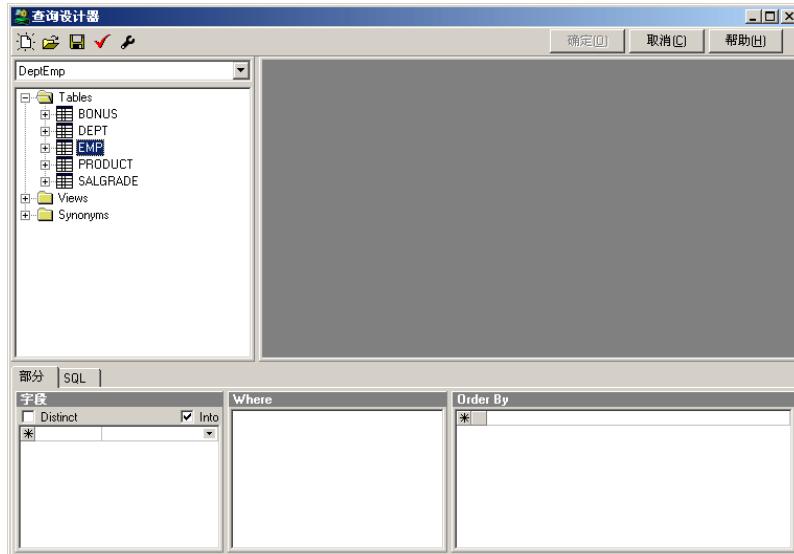
查询设计器允许你创建和修改你的 PL/SQL 和 SQL 源文件中的 select 语句。它提供了简单的指和点界面来建立 select 语句的字段列表、表列表、where 子句和 order by 子句。

### 24.1 创建新的 select 语句

对于这个例子，我们将假定我们要在 dept 和 emp 表之间创建一个链接语句，并显示一些来自于这两个表的一些列：

```
select e.empno, e.ename, d.deptno, d.dname  
from emp e, dept d  
order by e.empno
```

要创建一个新的 select 语句，请把编辑器指针放到你要它被插入的文本位置，再按下工具栏中的查询设计器按钮（或从弹出式菜单选择查询设计器项目）。这将调出一个空的查询设计器表单：



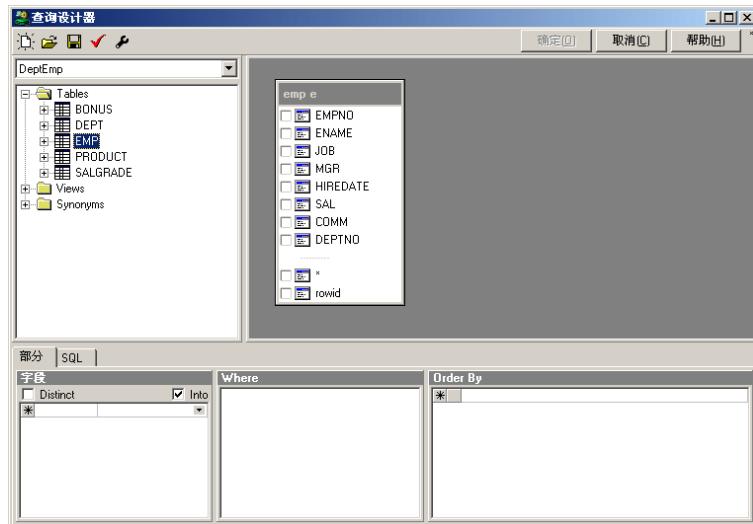
在左边，你可以看到一个只带有表和视图的对象浏览器。在右边，你可以看到工作区，现在正好是空的。在底部，你可以看到带有字段列表（into 项目列表可选）、where 子句和 order by 子句的三个面板。底部包含了一个标签页，它允许你转到将从当前查询定义中被创建的实际的 SQL 文本。

在窗口的顶部，你可以看到一个工具栏，它包括创建新查询的新建按钮、打开来自于文件的以前保存的查询定义的打开按钮、保存当前查询定义到文件中的保存按钮、解析当前查询定义的解析按钮和首选项按钮。确定按钮把你转回到 PL/SQL 或 SQL 编辑器，并且插入你已经建立的 SQL 文本。

要在 select 语句中包括表或视图, 请把它们从对象浏览器中拖出来放到工作区中。你每次添加一个新的表, 表属性对话框都将显示出来:



在这中，你可以输入表的别名。对于我们的例子，我们要使用‘e’作为 *emp* 表的别名：

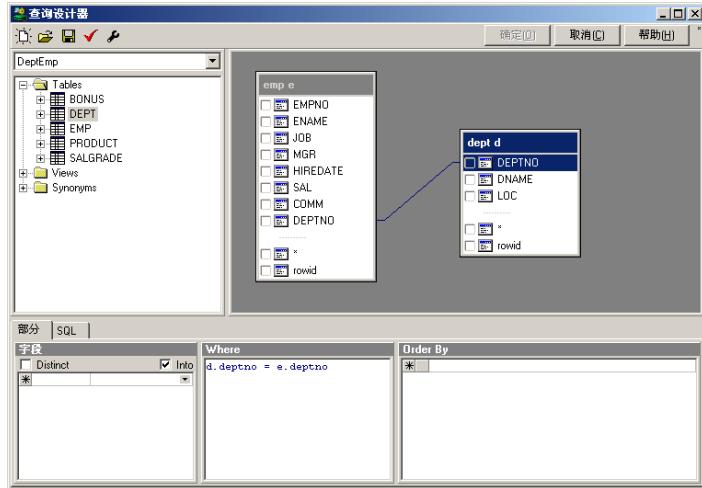


在用同样的方法添加了 *dept* 表之后，你会被问到是否使用名为 *EMP\_FOREIGN\_KEY* 的外键列来链接两个表：

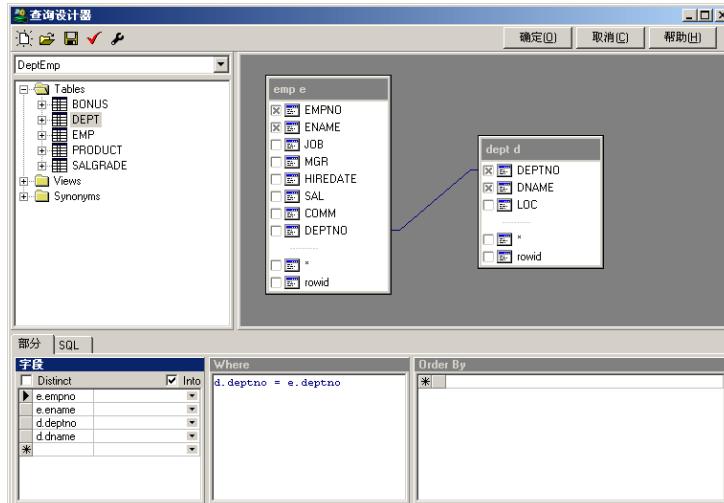


在检验了这个选项之后，你将在表单的下半部看到新的链接条件。

现在, *Dept* 表被添加了, 链接条件在新的工作区中被显现出来了:

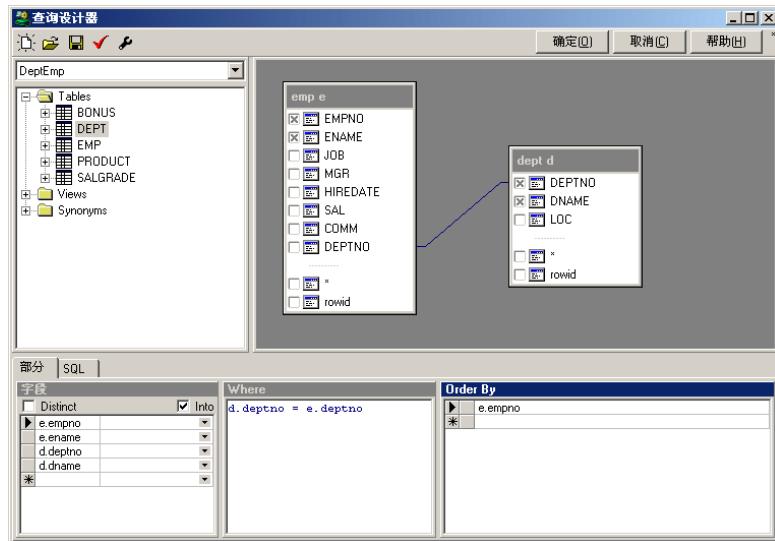


现在, 我们要做的事情是在 select 语句的字段列表中加入列, 只要单击列名旁边的复选框即可。列将被显示在字段面板中:



如果你要创建一个 PL/SQL *select .. into* 语句, 你还必须为邻近字段的 *into* 列表指定项目。这个选择列表将显示出在你插入语句的 PL/SQL 代码中被定义的 PL/SQL 变量。在这样的情况下, 我们不需要创建 *into* 列表, 所以我们可以保留这些项目为空。

要创建 order by 子句，把 *empno* 列拖到 Order By 面板中即可：



现在，你可以按确定按钮来把 SQL 语句插入到编辑器中了。

## 24.2 修改现有的select 语句

要在你的 PL/SQL 或 SQL 源代码中修改现有的 select 语句，简单地右键单击 select 语句并从弹出式菜单选择查询设计器项目即可。整个 select 语句都会在编辑器中被标记，并且查询设计器将显示。现在，你可以象前一章描述的那样来更改查询定义了。

## 24.3 处理查询定义

下列的段落描述了你可以怎样处理查询定义。

### 更改字段列表

要从工作区的表中添加字段到查询，你可以单击列旁边的复选框，或者你可以把列拖到字段列表中。作为一种选择，如果字段列表拥有焦点，你还可以双击列。

你还可以直接把列或列文件夹从对象浏览器拖到字段列表中。列文件夹位于表、它的约束和它的索引的下方。

你可以直接在字段列表中键入列名和其他表达式。

要从字段列表中删除一个字段，请右键单击它并从弹出式菜单选择删除已选项目。如果你选择了多个字段，你可以一下子把它们都删除。

通过选择它们，你可以把字段移动到不同的位置，把它们拖到一个新的位置。

如果你要一个 *distinct* 结果集，在字段列表上方检验相应的选项即可。

## 使用字段别名

要使用字段别名，只要在表达式后面键入别名即可，就象你正在用手键入 select 语句一样。

## 更改 where 子句

Where 子句是标准的文本编辑器，在这中你可以输入条件。要包括一列，你可以从工作区的表中拖它，或者 where 子句若有焦点你可以双击列。

要添加一个基于外键约束定义的链接条件，请右键单击工作区中的表并从弹出式菜单选择外键项目。你不能使用这个功能来删除一个现有的链接条件，必须用在编辑器中删除文本以代之。

## 更改 order by 子句

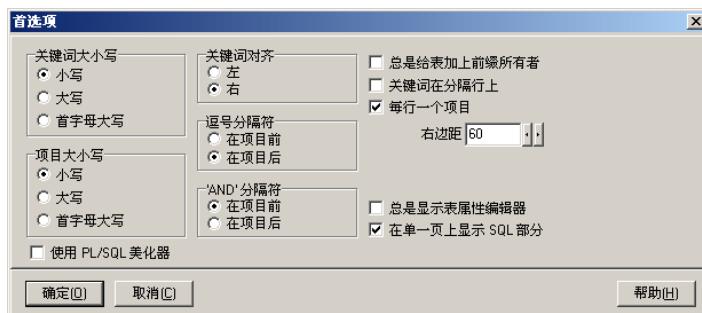
Order by 子句可以用与字段列表同样的方法来处理，除了你不能使用列名旁边的复选框之外。要在升序和降序之间更改一个项目，你可以单击项目旁边的按钮，它将显示出一个相应的箭头。

## 使用表或视图的同义词

要通过同义词使用表或视图，请右键单击工作区并从弹出式菜单选择添加表项目。在名称属性中键入同义词的名称，若必要就提供一个别名，再按确定按钮。

## 24.4 查询设计器首选项

通过按下工具栏中的首选项按钮，你可以更改查询设计器的行为。下列表单将显示出来：



这些首选项中的大多数选项影响了已生成的 select 语句的版面：

### 关键字大小写

决定了 SQL 关键字 (select、from、into、where、order by) 怎样被插入到已生成的 select 语句中。

### 项目大小写

决定了项目 (表名、列名) 怎样被插入到已生成的 select 语句中。

### 关键字对齐

决定了 SQL 关键字是左对齐还是右对齐。

### 逗号分隔符

决定了逗号是放在下一个项目前面还是放在当前项目后面。这主要用于每行一个项目选项被允许时。

### 'AND' 分隔符

决定了在 where 子句中' AND' 分隔符是放在下一个条件前面还是放在当前条件后面。

### 总是给表加上前缀所有者

当被允许时，表名被加上前缀所有者（例如用 scott.emp 取代 emp）。当被禁用时，如果表不被当前用户拥有，仅仅表被加上前缀所有者。

### 关键字在分隔行上

当被允许时，SQL 关键字被放在分隔行上，所有的项目都被放在后来的行上并用两个字符缩进。如果这个选项被禁用，项目将立即在同一行上紧跟关键字。

### 每行一个项目

当被允许时，每个项目都被放到一个新的行上。当被禁用时，项目都被放到一行上，直到到达右边距。

### 右边距

决定了在每行一个项目首选项被禁用时每一行的最大长度。

### 总是显示表属性编辑器

当这个选项被允许时，表属性编辑器将总是在添加新表到查询定义的后面被显示。当这个选项被禁用时，如果最近添加的表已经用于查询并且因此需要有一个别名，那么仅仅是属性将被显示。

### 在单一页上显示 SQL 部分

决定了 SQL 的三部分（字段列表、Where 子句、Order by 子句）被显示在单一的页上。如果这个选项被禁用，每一部分都被显示在一个分隔标签页上，在 SQL 文本标签页的旁边。

## 24.5 查询设计器插件

如果一个查询设计器插件被安装了，那么标准的查询设计器功能将激活这个插件。右键单击查询设计器按钮允许你从所有的已安装的查询设计器列表中选择一个查询设计器。当你随后再次使用查询设计器功能时，最近用过的查询设计器将被默认激活。

## 25. PL/SQL 美化器

如果你与几个开发者一起在工程小组中开发、维护和支持 PL/SQL 代码，那么所有的工程成员都使用相同的代码样式是有用的。这将对产品的质量和生产效率是有益的。要实施这个，PL/SQL Developer 提供了一个 PL/SQL 美化器。

你要做的所有事情就是设置 PL/SQL 美化器选项，并可以通过 PL/SQL 美化器规则随意地定义代码样式（你也可以简单地采用默认规则）。之后，当你工作时，你的 PL/SQL 代码将自动被美化，或者通过激活美化器选项被明确地美化。

### 25.1 定义选项

要定义 PL/SQL 美化器选项，请选择来自于编辑菜单的相应项目，下列首选项对话框将显示出来：



规则文件决定了美化你的代码时将要遵循的规则。你可以保留它为空而采用默认规则，它可以通过在 PL/SQL Developer 安装目录中的 *default.br* 文件来定义。要了解关于这些规则的更多信息，请参见下一章。

此外，你还可以定义你的 PL/SQL 代码将在什么时候被自动美化：

- **在创建后** - 只要一个新的程序文件被通过模板创建，或者一个 DML 语句从对象浏览器中被拖放到程序窗口中。
- **在载入后** - 在程序文件从文件系统中被载入之后。这不影响来自于 Oracle 数据库的被打开的代码。
- **在编译前** - 当你在程序窗口中编译程序时。这保证了 Oracle 数据库中的所有 PL/SQL 代码都遵照美化规则。
- **在保存前** - 在程序文件被保存到文件系统之前。

## 25.2 定义规则

如果你在选项对话框中按编辑按钮，你就可以查看或编辑已选文件的规则：



在更改了这些规则之后，你可以按保存或另存为按钮来保存这些更改。你还可以打开另一个文件用于查看或编辑，并且通过按相应的按钮你可以恢复到默认规则。如果你以新的名称保存了文件或者打开了不同的文件，你需要在选项对话框上选择那个规则文件以使这些规则有效。

在一般标签页上，你可以更改应用到你的 PL/SQL 代码的一般样式的规则。你做的所有更改将立即在对话框下部的预览窗格中可以看到，而且大多数选项都是不言自明。下列选项需要做一些解释：

- 缩进** - 针对嵌套结构（例如 begin/end、if/then/else、loops 等）将要缩进的空格数。
- 右边距** - 只要代码需要转到下一行，右边距就会被当作指引线使用。在那中也许有超越右边距的情况，例如长字符串被使用时就可能是这样。
- 使用跳格字符** - 当被允许时，作为结果的代码将允许用硬跳格符（字符 9）来缩进代码。当被禁用时，空格将被使用。
- 跳格字符大小** - 硬跳格符表现出的字符数。这在使用跳格字符选项被允许时并没有用，但对于正确地对齐包含硬跳格符的注释还是有用的。
- 空行** - 你可以定义空行应该被删除，还可以定义美化器将通过它自己的规则插入空行、或者后来的空行组应该合并为一个空行、或者所有的空行都被保留。
- 项目列表** - 这些规则应用到所有的不分成其他标签页类的项目列表。

- **语法大小写** - 决定了 PL/SQL 关键字和标识符怎样用大写字母来写。使用特殊大小写选项允许你定义指定大写的字数。例如，如果你在这个列表中包含了 DBMS\_Output，这将是那个词在你的 PL/SQL 代码中的准确的大写。你还可以使用 % 或 \* 通配符，例如，DBA\_% 会将所有以 DBA\_ 开头的词的前三个字母转为大写。

其他标签页包含了应用到你的 PL/SQL 代码的特殊样式的规则。这些规则还是不言自明的，被示范在预览窗格中。

## 25.3 使用美化器

如果你在选项对话框上还没有选择任何美化事件，通过选择编辑菜单的 *PL/SQL 美化器项目*、通过按下工具栏中相应的按钮或通过使用被指派给它的功能键，你都可以明确地美化你的代码。在默认的情况下，在当前的 PL/SQL 编辑器中的所有代码都将被美化。如果有严重的错误出现在代码中，它就不会被美化，而且你会收到一个错误消息。

你也还可以选择一块完整的 PL/SQL 代码（例如，一个本地的函数或过程，或者一个 DML 语句）并激活美化器。

## 25.4 取消格式化

如果有一段希望手工格式化的 PL/SQL 代码，需要取消 PL/SQL 美化器的格式化设定，可以使用 *NoFormat Start* 和 *NoFormat End* 指令。例如：

```
...
TimeOut := 1000;
-- NoFormat Start
DBMS_Alert.WaitOne('SHUTDOWN',
    Message, Status,
    TimeOut);
-- NoFormat End
if Status = 0 then
...

```

## 26. 模板

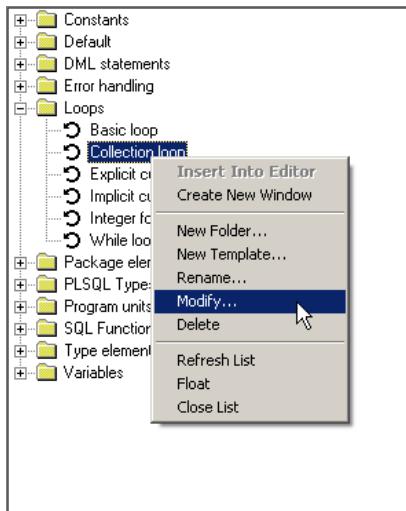
只要你创建一个新的程序单元，它的初始内容都是基于模板的，模板包含了预先定义的文本和变量。当选择了一个模板时，你将被提示输入这些当时放在模板文本中的变量的值。

模板还可以用于把 SQL 或 PL/SQL 代码片断插入到现有的源中。例如，你可以创建一个 cursor-for-loop 的模板，无论你何时需要这样的程序结构你都可以用它。

模板是纯文本文件，它位于 *Template* 目录中，以 *.tpl* 为扩展名。对于所有的程序单元类型，模板都是预先被定义好的，普通的 SQL 和 PL/SQL 结构的模板也是这样。你可以为自己特定的环境修改和创建模板。

### 26.1 模板窗口

模板窗口显示出了模板目录的层次结构，这使你组织模板变得容易了。当你第一次启动 PL/SQL Developer 时，模板窗口将停放在对象浏览器的下方。这样，它总是可用的，从不妨碍其他文档窗口的查看：



这里有很多被预先定义好的模板，它们按下列模板文件夹被组织起来：

- 常数 - 用于 PL/SQL 常数声明的模板。
- 默认 - 带有 SQL 窗口、测试窗口、命令窗口或解释计划窗口的默认内容的模板。它还包含了一个新的视图对象的模板。你大概不会在这中直接使用这些模板，但可以在这中维护它们。
- DML 语句 - 你可以在 PL/SQL 中使用的 DML 语句的模板。
- 循环 - 各种 PL/SQL 循环的模板。
- 包组件 - 可用于包说明或包体的组件的模板。
- PLSQL 类型 - 能在 PL/SQL 中被声明的所有类型的模板。

- 程序单元 - 这是一个特殊的文件夹，它包含了针对新的程序单元的模板。只要你创建一个新的程序单元，来自于这个文件夹带有与程序单元类型一样的名称的模板就会被使用。
- SQL 函数 - 带有所有函数的模板，例如 add\_months 、 substr 等。如果你不知道特定函数的准确名称或参数含义，那么这个模板就可能是有用的。
- 类型组件 - 可用于类型说明或类型体的组件的模板。
- 变量 - PL/SQL 变量声明的模板。

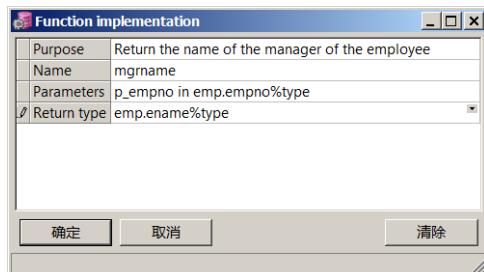
在默认的情况下，模板窗口被停放在对象浏览器的下方，但是通过右键单击它并从弹出式菜单选择浮动项目你可以把它转入到浮动窗口。要再次停放窗口，就按同样的过程行事。

通过从弹出式菜单选择关闭项目，你可以关闭模板窗口。要使模板窗口再现，请选择工具菜单中的模板项目。

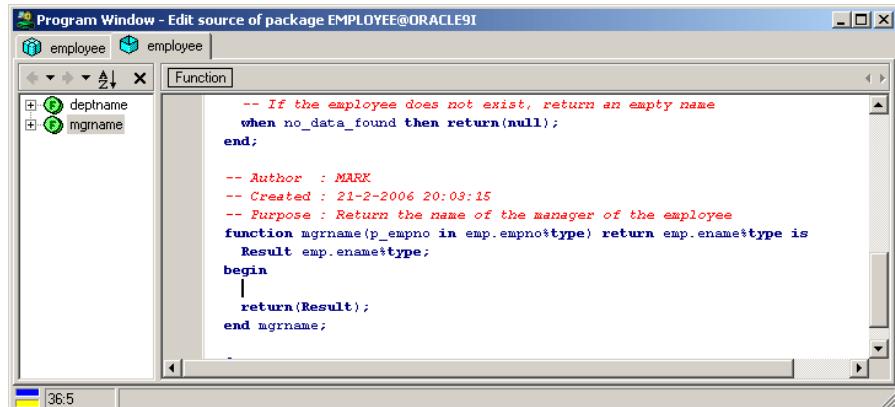
通过在窗口菜单中使用将保存当前状况的保存版面项目，你可以保存这个版面以用于你下次启动 PL/SQL Developer 。这个功能的更多详细资料在第 33.2 节中有描述。

## 26.2 使用模板

要插入模板到现有的源，你可以简单地双击它。如果模板包含了必须被用户指定的变量，一个输入表单将在文本被插入之前显示出来。例如，如果你双击了函数实现模板，下列输入表单将显示出来：



在填了表单并按下了 确定 按钮之后，模板文本将被插入到源中当前光标所在的位置：



你还可以把一个模板从模板窗口拖放到编辑器中指定的位置，这个位置也决定了模板文本的缩进。同样地，你可以右键单击编辑器中的一个地方，并选择一个来自于弹出式菜单的模板子菜单的模板项目。

## 26.3 创建和修改模板

要修改现有的模板，请右键单击它并从弹出式菜单选择修改项目。这将调出带有模板文本的文本编辑器，这个文本包含了模板的完整说明，包括变量、查询等。模板文本的格式在下面的章节中有描述。

要创建一个新模板，请在你要创建它的文件夹中右键单击它，并且从弹出式菜单选择新建模板项目。你将首先被提示输入一个模板的名称，之后模板文本编辑器将显示出来。通过从弹出式菜单选择新建文件夹项目，你还可以创建一个新的文件夹。

### 模板文本

模板说明包含了文字文本和变量。文字文本将简单地被复制到源文件中，不带有任何修改。设想一下一个名叫 *Commit* 的模板，它只包含了文字文本 *commit;*。当这个模板被双击时，文本 *commit;* 就被插入到光标所在位置的源中了。

### 用户变量

模板变量是模板文本中的代码，它将被用置换文本替换掉。这个置换文本可以是一个隐含的值，就象当前日期或用户名，或者它可以被用户指定。当模板被调用时，这些用户变量将在输入表单中显示出来。下面的例子为一个简化的函数模板定义了变量 *Name* 和 *Type*：

```

create or replace function &Name return &Type is
begin
    return(result);
end &Name;

```

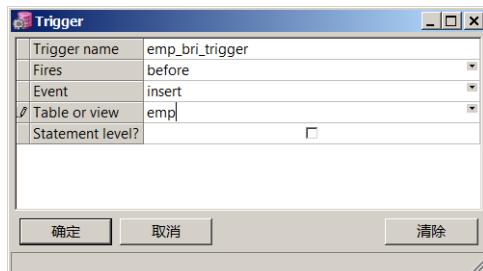
当这个模板被调用时，用户可以指定 *Name* 和 *Type* 变量的值。正如你能看到的，*Name* 变量被使用了两次。提示用户输入值只有一次，两个具体值都将被用同样的值替换。还有更多高级变量属性可以使用。

模板使用与 SQL 窗口和报告窗口相同的变量语法。有关完整的描述，参见第 12.3 节。

下面是一个触发器模板的例子，正好使用了用户变量：

```
create or replace trigger &"Trigger name"
  &<name="Fires"
    list="before,after,instead of"
    restricted="yes"
    default="before"> &<name="Event"
      list="insert,update,delete">
  on &<name="Table or view"
    list="select lower(object_name)
          from user_objects
          where object_type in ('TABLE', 'VIEW')
          order by object_type, object_name">
  &<name="Statement level?" checkbox="for each statement,for each row">
declare
  -- local variables here
begin
  end &"Trigger name";
```

使用此模板时，用户将被提示通过以下方式输入变量值：



注意，你可以在一个模板中创建多个程序单元，用只带有一个斜杠 (/) 字符的线来分隔它们。这样，你就可以在一个单独的程序文件中创建一个包说明和体的模板。

## 隐式变量

除了用户变量之外，你还可以使用隐式变量。这些变量的置换文本不由用户指定，而是由系统定义。

## 系统变量

下面有四个系统变量被定义：

- \$OSUSER - 操作系统用户名。
- \$DBUSER - 当前登录的数据库用户名。
- \$DATE - 当前日期。
- \$TIME - 当前时间。

下面例子模板的前两行把 Windows 用户和当前日期/时间插入到了源文件中：

```
-- Author : $OSUSER
-- Created : $DATE $TIME
-- Purpose : &Purpose
procedure &Name is
begin
;
end &Name;
```

注意，系统变量在模板文本中不使用 & 符号，而是在前面加上 \$ 符号。

如果你要使用系统变量的原名而不是置换值在作为结果的文本中显示出来，你可以通过在名称前放上第二个 \$ 符号来避开这个置换值。例如：

```
-- $$Date$ $$Revision$
```

作为结果的文本将是 \$Date\$ \$Revision\$。

### 光标位置

光标位置变量决定了在模板文本被插入到编辑器中之后文本光标将在编辑器中的什么地方。仅在你希望的地方放上 [#] 即可：

```
loop
[#]
end loop;
```

### 模板图标

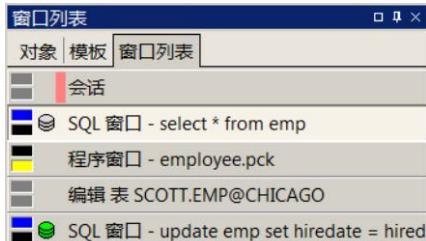
在模板窗口中树形视图为每个模板都显示了一个图标。模板窗口首先寻找与模板名一样的位图文件名。对于模板 Cursor.tpl，它将显示一个来自于同一个目录的名为 Cursor.bmp 的位图，如果这个位图文件存在的话。如果这个位图不存在，它将在当前目录下的目录中寻找 Cursor.bmp 文件。如果这个位图还是没有被找到，它将在当前目录中和下面的目录中寻找 default.bmp 文件。

模板图标的位图的大小应该是 16x16 像素，颜色为 16 色。

## 27. 窗口列表

窗口列表允许你直接和容易地在多个文档窗口之间导航。如果你喜欢用最小化文档窗口来工作，这个特性就特别有用，仅仅是当你有大量的打开的窗口，这样一张列表也总是会使导航更容易。

在默认的情况下，窗口列表被停放在对象浏览器和文件浏览器，模板列表也同样停放在这个位置。它仅仅在一个或多个窗口被打开时才是可见的，并且显示这些窗口的标题、文件状态、事务状态和连接指示器信息。



如果你单击这些项目之一，相应的窗口就会被激活。如果你在窗口列表中右键单击一个项目，下面的弹出式菜单将显示：



从这个弹出式菜单中，你可以激活、关闭和调整窗口大小到正常或最小状态。你还可以通过按 Shift 键并单击列表中的项目来关闭一个窗口。

你可以从链接子菜单中选择一个新的链接来改变窗口的链接。

此外，你可以重新命名一个窗口。如果你正好创建了一个新的窗口并且要为它提供一个有意义的名称，你又不想用比如 *SQL 窗口 - 新建* 这样的名称，这就可能是有用的。

*文本简写* 选项可以让描述更简洁。例如：“*Program Window - employee.pck*”会直接简写为“*P - employee.pck*”。

*新建* 菜单项目允许你创建一个新的程序窗口、SQL 窗口、测试窗口、命令窗口、解释计划窗口、图表窗口、或新的数据库对象。

要运行某一窗口，你可以选择 *执行* 菜单项目。

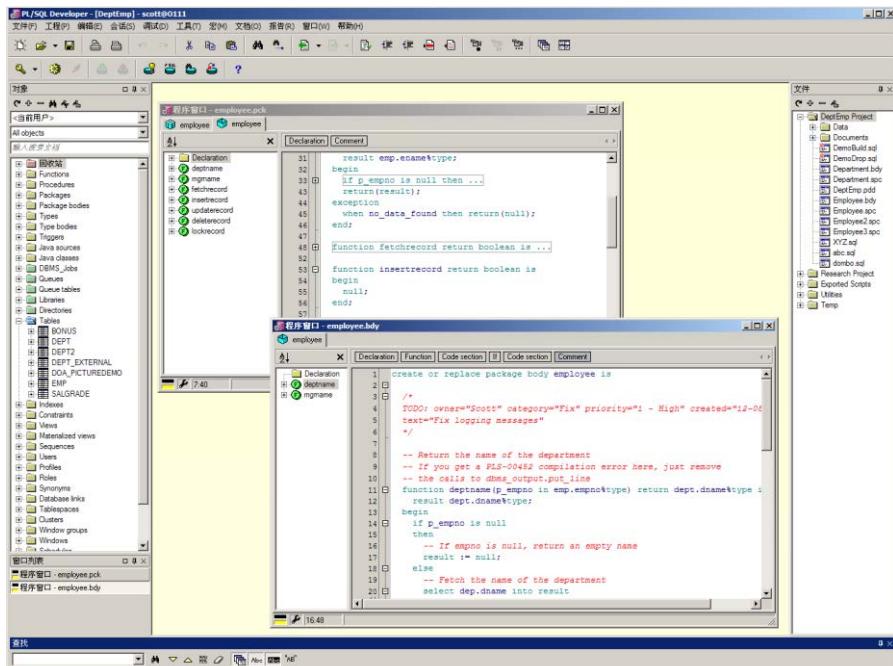
要关闭窗口列表，使用泊入工具里的关闭按钮。要使窗口列表重现，请选择工具菜单中的 *窗口列表* 项目。通过在 *窗口* 菜单中使用将保存当前状况的 *保存版面* 项目，你可以保存这个版面以用于你下次启动 PL/SQL Developer。这个功能的更多详细资料在第 33.2 节中有描述。

## 28. 可停放和浮动的工具

下列工具能被停放或者能被放到浮动窗口中：

- 对象浏览器
- 文件浏览器
- 链接列表
- 窗口列表
- 模板列表
- 书签列表
- 搜索栏
- 搜索结果
- 任务项目
- 工程项目

工具可停放在工作区的左侧、右侧、上端和下端：



要停放一个工具，只需将其拖放到工作区的相应端即可。你一接近到工作区的边，一个图文框就显示了，它指示出了停放位置。如果另一个工具已经被停放到了这个位置，你还可以上、下拖这个新工具到左边、右边或者其他工具内部。如果你拖了一个工具到其他工具内部，它们将

在不同的标签页上的同样位置都变得可用（请参见上面屏幕截图中坐下部的窗口列表和模板列表）。请注意，只有在停放的窗格非垂直分隔时，才能将工具停放到另一个工具的左侧或右侧。要使一个工具浮动，简单地把它从停放位置拖到浮动位置即可。

在每个停放区的右上角都有如下三个自左向右排列的按钮：



此按钮可最大化或恢复停放区的大小。



此按钮将固定或取消固定停放区。在固定时，停放区始终可见，工具始终可直接访问。在取消固定时，若鼠标指针未停留在该区域上，停放区将滑出屏幕，显示为一个垂直标签页。将鼠标悬停在垂直标签页上即可再次显示停放区。



此按钮将关闭停放区的当前工具。

## 29. 工具栏

PL/SQL Developer 的工具栏允许你快速访问常用功能。你可以按照自己的喜好，通过以下方式修改工具栏：

### 移动和取消停放工具栏

单击工具栏左侧边缘处的竖条手柄即可移动各工具栏：



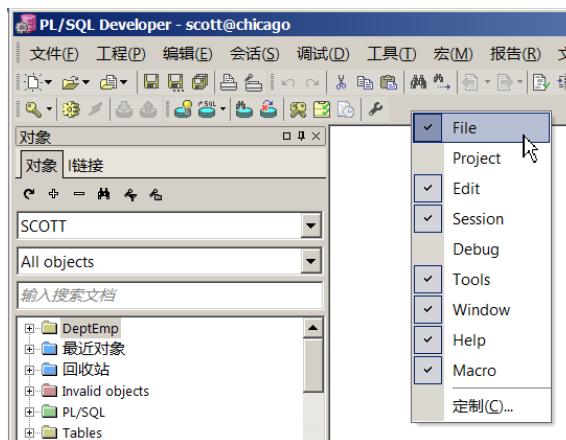
可以将工具栏移动到工具栏区域的其他位置，也可以将其拖出工具栏区域，取消停放工具栏，创建浮动工具栏：



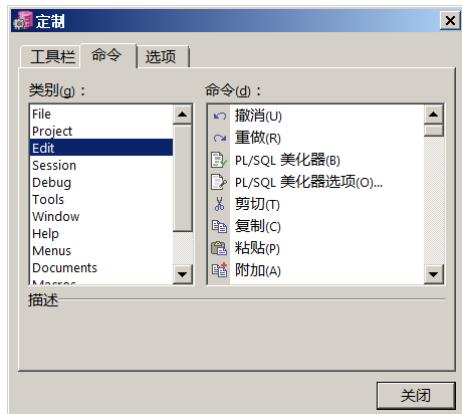
将工具栏移回工具栏区域即可再次停放。

### 添加和删除工具栏与按钮

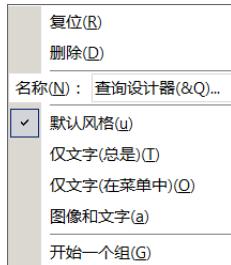
右键单击工具栏区域，即可利用弹出菜单隐藏或显示特定工具栏：



此时选择定制菜单项，即可从定制窗口的命令选项卡页中将命令拖放到工具栏：



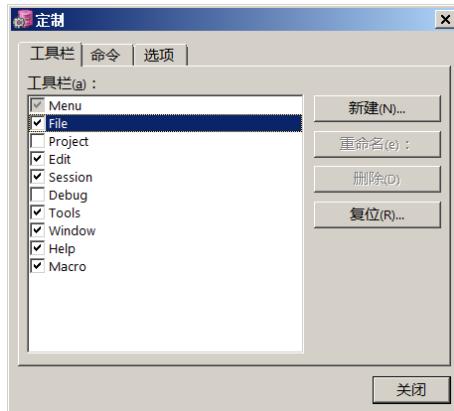
在打开定制窗口的情况下，右键单击工具栏上的按钮即可看到以下操作和属性设置菜单：



- **复位** – 将所有按钮属性复位为默认值。
- **删除** – 从工具栏中删除该按钮。
- **名称** – 如果按钮包含文字选项，则显示按钮名称（见下文）。
- **图像和文字选项** – 控制是否显示图像和/或文字。
- **开始一个组** – 启用此选项时，此按钮前将带有一个组分隔线。

## 创建你自己的工具栏

要创建自己的工具栏，请转到定制窗口的工具栏页面：



单击新建按钮创建新工具栏。这将创建一个新工具栏，随后你可以为该工具栏添加按钮。在相同的选项卡页面上，还可以重命名和删除你自己的工具栏。此页上的标准工具栏不能修改。

## 定制工具栏行为

在工具栏定制窗口的选项选项卡页面中，可以设置如下选项：



## 30. 授权

在安装了 PL/SQL Developer 之后，在系统权限和被授予到链接到数据库的 Oracle 用户的对象权限的范围内，所有的用户都能使用 PL/SQL Developer 的全部功能。例如，如果 Oracle 用户没有 *create user* 的系统权限，PL/SQL Developer 也能在 PL/SQL Developer 中启动新建用户功能，但最终能得到一个来自于 Oracle 的错误信息：“*ORA-01031, 权限不够*”。

你可以明确地把所有相关的 PL/SQL Developer 功能授权给特定的 Oracle 用户和角色。在一个开发数据库中，你将允许所有的开发者使用所有的功能；然而在产品数据库中，你将典型地对大多数用户禁用所有功能，以使他们不能更改数据库或者不能占用太多资源而影响运行。

通过授予 PL/SQL Developer 权限给角色，你可以为特定人群定制授权。你可以利用现有的映射到用户组的角色（例如 DBA 和 RESOURCE）或者你可以专门为 PL/SQL Developer 用户组创建一些角色。

### 30.1 允许授权

授权信息被储存在 sys 方案的 *plsqldev\_authorization* 表中。只要数据库中没有这个表，所有用户就都有权使用 PL/SQL Developer 的所有功能。如果你链接到还没有授权的数据库并且选择来自于文件菜单的授权项目，你将得到下列问题：



如果你选择是，一个空的授权表就被创建了。注意，你必须被链接为 DBA 才能创建这个表。在这个表中，你一创建一个或多个权限，授权就被激活了，而且仅仅被授权的用户可以在这个数据库中使用 PL/SQL Developer。

在默认的情况下，只有表的所有者（sys）或另一个 DBA 才能更改表或更新它的内容。在表中，所有的用户都有 select 权限。

## 30.2 定义授权

要定义授权，你必须被链接为 DBA 并且启动来自于文件菜单的授权功能。下列对话框将显示出来：



在上部，你将看到所有的用户和角色，它们都具有一个或多个已经被授予的 PL/SQL Developer 的权限。在下部，你可以看到当前已选的被授予者的实际权限。

要添加一个被授予者，请按被授予者列表右边的新建被授予者按钮。这将调出一个所有潜在被授予者的列表（用户和角色）。你可以选择一个或多个（Ctrl-单击 或 Shift-单击）被授予者并按确定按钮来添加他们。在这中，这些新的被授予者还没有任何权限。

要删除一个被授予者和它的权限，请按移除被授予者按钮。这不会影响数据库中的实际用户和角色，只不过是从授权表中删除了它的权限而已。

要授予权限，请从列表中选择被授予者并按权限列表右边的授予权限按钮。一个 PL/SQL Developer 所有权限的列表就显示了，在这中你可以选择一个或多个权限。共有三种权限：

- **系统权限** - PL/SQL Developer 系统功能（例如登录）。
- **对象权限** - 数据库对象类型特定权限（例如删掉表）。
- **菜单权限** - 使用 PL/SQL Developer 菜单功能的权限（例如，工具 > 会话）。

**注意：** 如果用户没有 *System.Logon* 或 *All* 权限，他或她就不能在当前数据库实例中使用 PL/SQL Developer 。

你可以选择单独的权限或者选择不同等级的权限。例如：

- **All** - 授予 PL/SQL Developer 的所有权限。
- **Objects.All** - 授予 PL/SQL Developer 的所有权限到数据库对象。

- **Objects.Rename All** - 授予 PL/SQL Developer 的所有的涉及到重新命名数据库对象的权限（重新命名表、视图、序列和同义词）。
- **Objects.Rename Table** - 授予 PL/SQL Developer 的用于重新命名表的权限。

注意，对于数据库对象，Oracle 用户或角色仍然必须有必要的系统权限以用于执行操作。授予 *Objects.Drop User* 权限仍然需要 *删除用户* 的系统权限。

还要注意的是，如果用户有 *删除用户* 的系统权限，他就能从 SQL 或 PL/SQL 脚本中执行 *删除用户* 命令。

### 30.3 禁用授权

要永远禁用授权，你可以删掉 *sys.plsqldev\_authorization* 表或者删除所有记录。要临时禁用授权，你既可以撤回在表中的权限又可以重新命名它（使它在 PL/SQL Developer 用户中不可见）。要在以后再次允许它，你就再次授予 *select* 权限。

## 31. Oracle 文件系统(OFS)

OFS 允许你在 Oracle 数据库中储存你的所有文件（源、脚本、报告等）。这是有好处的，数据库对象与对这些对象产生影响的文件一样，二者能被储存到相同的数据库中，并且都能被已经访问到这个数据库的所有用户访问。此外，数据库对象和文件被保证是一致的，并且能用一致的方法来备份和恢复。

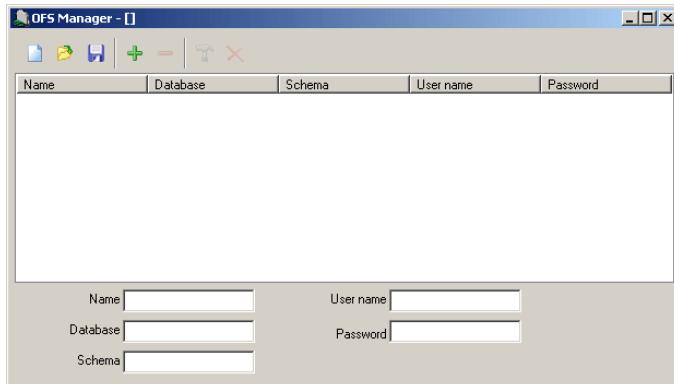
文件能用通常的方法保存和打开，它带有有一些扩展名的标准的文件打开/保存对话框。

### 31.1 OFS 管理器

在任何人能从 PL/SQL Developer 内部使用 OFS 之前，你必须使用 OFS 管理器来定义一个有一个或多个 OFS 存储单元的 OFS 存储单元目录。对于每个存储单元，你需要安装 OFS 数据库对象。

#### 创建 OFS 存储单元目录

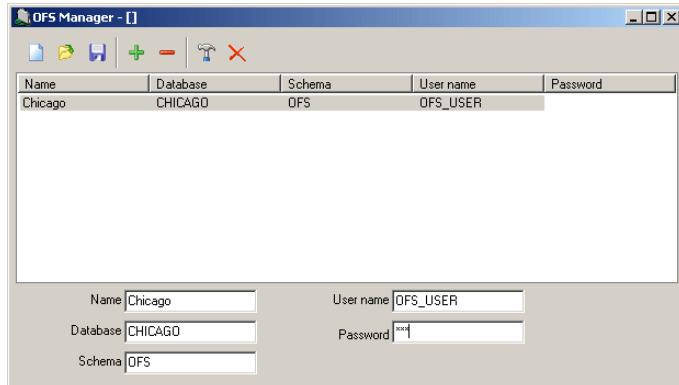
要启动 OFS 管理器，请转到 PL/SQL Developer 首选项并选择目录页。在这一页的底部，你能发现 OFS 管理器按钮。在把它启动起来之后，你将得到下列屏幕：



这是一个空的 OFS 存储单元目录，你可以添加存储单元，你可以保存存储单元到 OFS 存储单元目录文件中 (.ldf)。这个文件被用于 PL/SQL Developer 内部以浏览目录和链接到存储单元来浏览文件。

## 添加 OFS 存储单元

要添加一个 OFS 存储单元, 请按添加存储单元 (+) 按钮。现在, 你可以输入下列信息了:



- **名称** - 将在文件选择器的存储单元列表中显示。
- **数据库** - 储存 OFS 文件的数据库。
- **方案** - 保存 OFS 数据库对象（表、包等）的方案。这个用户需要资源角色权限和查询重写系统权限来创建它的对象。
- **用户名** - 将被用于访问 OFS 的 Oracle 用户的名称。如果你不指定一个 OFS 用户名, 那么访问 OFS 的用户将需要指定一个用户名和口令, 或者被当作当前的 PL/SQL Developer Oracle 用户来访问 OFS 。
- **口令** - 将被用于访问 OFS 的 Oracle 用户的口令。仅仅用于用户名区域被输入时。

创建了一个或多个存储单元之后, 你可以通过按保存按钮来保存 OFS 存储单元目录。如果你在 PL/SQL Developer 目录中以名称 OFS.1df 来保存文件, 那么 PL/SQL Developer 将自动地获得这个目录。另外, PL/SQL Developer 用户将需要指向在首选项中的正确的 OFS 存储单元目录文件。

## 安装 OFS 数据库对象

在添加了一个存储单元到目录以后, 你需要安装它的数据库对象。如果 OFS 方案和（可选的）OFS 用户还没有存在于数据库中, 你要首先创建它们:

```
create user OFS identified by <password>;
grant connect, resource, query rewrite to OFS;
create user OFS_USER identified by <password>;
grant connect to OFS_USER;
```

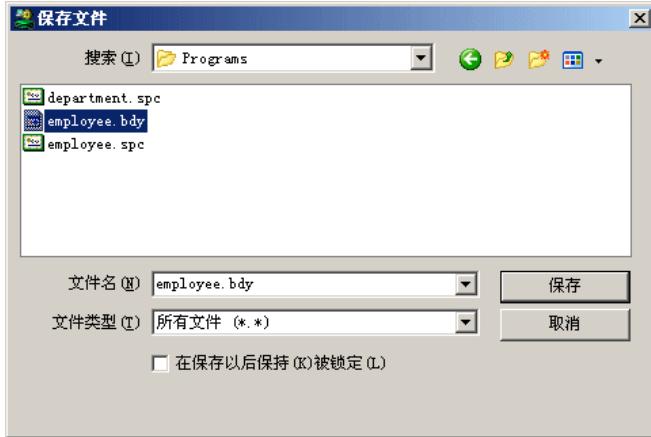
其次, 你可以通过按安装数据库对象按钮来安装 OFS 数据库对象。在提供了方案用户的口令之后, 数据库对象将被创建, 存储单元已经准备好可以使用了。

## 31.2 OFS 用法

在创建了一个 OFS 存储单元目录、安装了存储单元并且使目录文件可以用于 PL/SQL Developer 之后，你就可以使用来自文件菜单的 *OFS 另存为*项目来保存文件到 OFS 中了，下列对话框将显示出来：



你首先需要打开一个存储单元，然后你就可以保存文件或者创建文件：



要从 OFS 中打开一个文件，你可以使用来自文件菜单的 *OFS 打开项目*，或者使用重新打开菜单。OFS 文件有路径名为 OFS:\Location\Path（比如，OFS:\Chicago\Programs\Employee.bdy）。前面被打开的 OFS 文件能通过使用标准的保存功能来保存，你可以使用另存为功能来保存一个 OFS 文件到标准的文件系统，*OFS 另存为*功能可以被用于在 OFS 中以不同的名称或者在不同的目录中来保存它。

当保存或打开一个文件时，你可以选择保持锁定选项来锁定它，其他用户不能锁定或覆盖被你锁定的文件。

在文件选择器中右键单击一个文件会给予你一些选项：剪切、复制、粘贴、删除或重命名文件。你还可以从这个弹出式菜单查看和更改对象属性，你可以更改的属性包括只读状态、被锁定状态和被压缩状态。文件只能在 Oracle 10g 和以后版本中被压缩，因为它使用了只有在 Oracle 10g 以后才可用的 *utl\_compress* 包。

## 32. 帮助系统

你可以配置 PL/SQL Developer 使之与 Oracle 服务器软件的帮助文件和手册结合起来。这一章描述了怎样设置和使用这些帮助系统。

### 32.1 MS 帮助文件

一直到 Oracle 7.2，在线手册都使用 MS 帮助格式。这些帮助文件可以被用于在编辑器中提供上下文有关的帮助，既可以通过右键单击编辑器中的一个词、通过选择帮助菜单中的 *SQL 帮助* 或 *PL/SQL 帮助* 项目来实现，又可以通过按你已经把这两个菜单项目关联起来的功能键来实现。

PL/SQL Developer 将在下列两个目录中搜索帮助文件：

1. 在 PL/SQL Developer 所在的目录中。
2. 在 SQL\*Plus 所在的目录中。

SQL 帮助文件名被设为 *sqlhelp.hlp*，PL/SQL 的帮助文件就是 *plshelp.hlp*。如果这些帮助文件不可用，但你有一些想要使用的针对 SQL 或 PL/SQL 帮助的其他帮助文件，你就可以把这些文件放到 PL/SQL Developer 目录中并按前面提到的方法给它们重新命名。

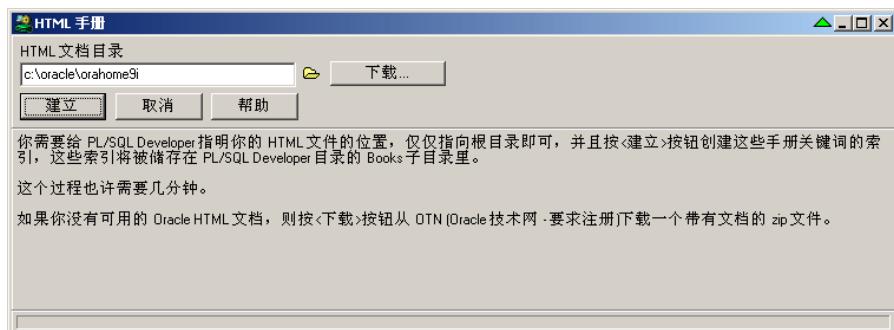
### 32.2 HTML 手册

自 Oracle 7.3 以来，在线手册已经被提供为 Acrobat Reader 和 HTML 格式。你可以通过配置 PL/SQL Developer 来使用 HTML 手册，既可以通过右键单击编辑器中的一个词、通过选择帮助菜单中的 *HTML 手册* 项目来实现，又可以通过按 *F1* 来实现。此外，在 Oracle 出现异常的情况下，你可以在错误消息上按帮助按钮，这将自动地把你带到 Oracle 错误信息手册中的适当的段落。在程序窗口中双击一个编译错误也将显示出这个信息。

**注意：**一些版本的 Oracle 服务器的在线文档包含了超过 1MB 大小的 HTML 文件。载入这些文件能花费很长时间，在这样的情况下建议你使用章节被分成较小文件的一些版本的 Oracle 服务器的在线文档。对于 8.1.5 文档，Oracle 已经再次减小了文件的大小。

#### 配置 HTML 手册

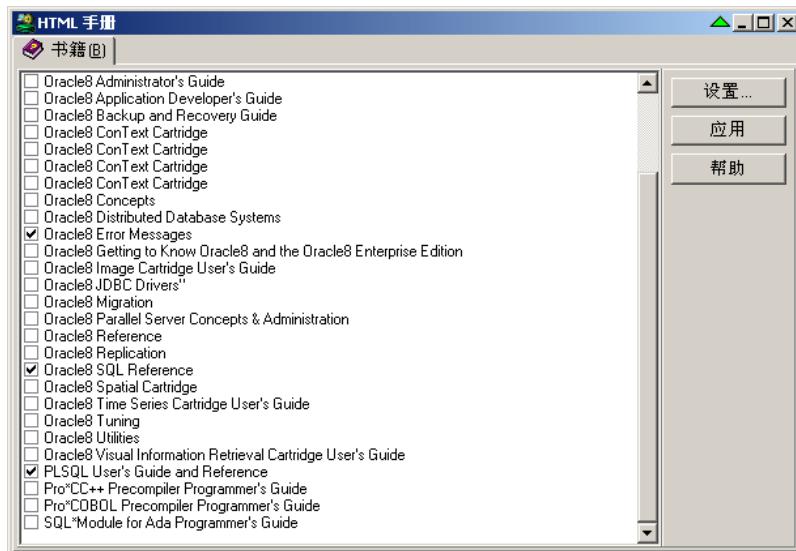
按照上面所描述的方法之一，你第一次调用 HTML 手册时，你需要告诉 PL/SQL Developer 你的 HTML 文件在什么地方：



仅仅指向根目录并且按建立按钮就可以在这些手册中为关键字创建索引了，这些索引被保存在 PL/SQL Developer 目录的 *Books* 子目录中。这个过程可能需要几分钟。

如果你没有可用的 Oracle HTML 文档，那么你就可以按下载按钮。这将把你带到本地的 HTML 页，在这中你可以选择适当版本的 Oracle 服务器并下载一个带有 HTML 文档的 zip 文件。这个下载需要你有一个 Oracle 技术网 (OTN) 的账号。如果你没有 OTN 账号，你可以先创建一个，它是免费的。

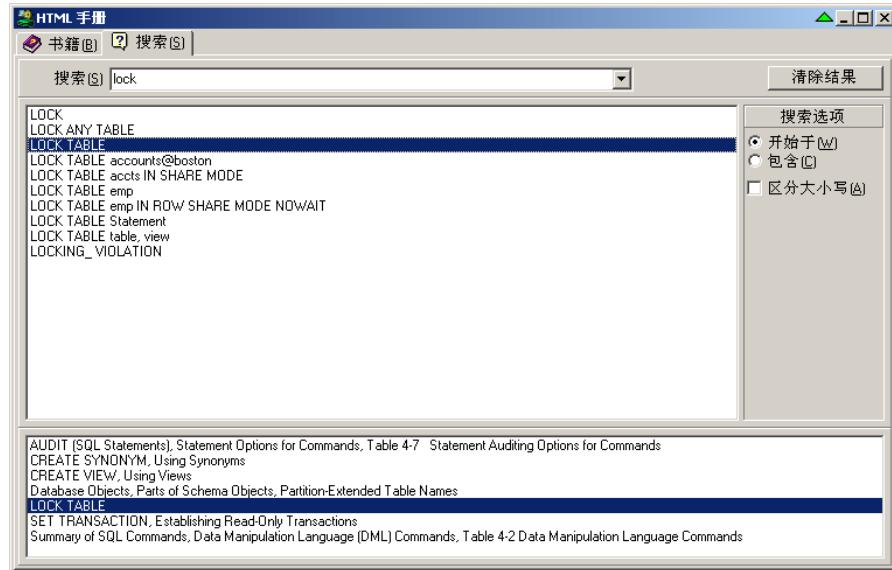
在建立索引完成之后，你会得到已经被找到的手册的列表。现在，你可以选择你要用的上下文关联帮助的书籍了。在默认的情况下，这些选择将包括错误信息、SQL 参考和 PL/SQL 参考手册，它们都包含了在 PL/SQL 开发期间的相关信息。在 Oracle 出现异常或编译错误的情况下，如果你要自动地得到有用的原因和行为，错误信息手册是必需的。当前的 Oracle7 手册不能适当地被检索以允许这个特性，Oracle8 版本就没有这个问题。



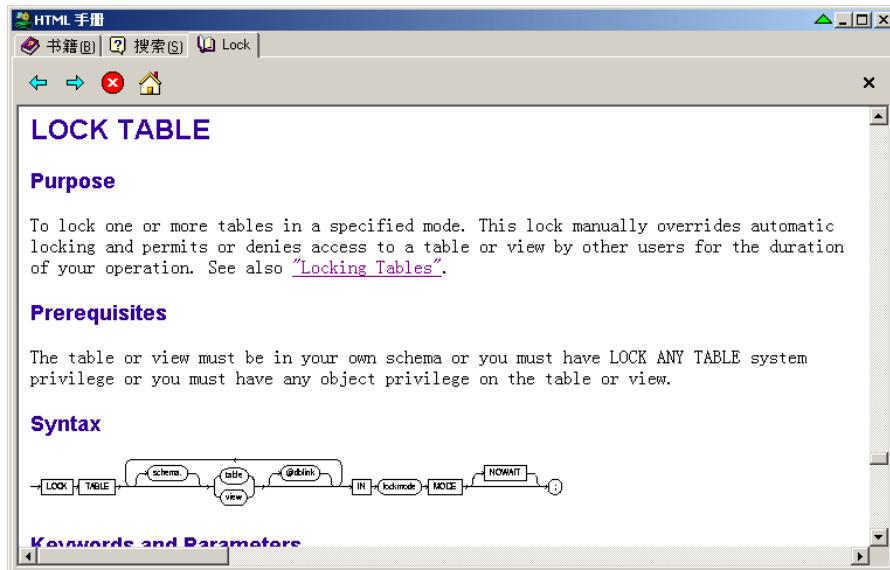
按应用按钮可使选择生效。在此之后，你会被带到搜索页，这将在下一章中讨论。通过按设置按钮，你可以为 HTML 手册扫描不同的目录。

## 使用 HTML 手册

在调用了 HTML 手册之后，在搜索页中，你可以键入你要搜索的关键字。有两个列表来显示匹配这个搜索和手册中的主题的关键字，在这中关键字出现了：



在关键字右边的搜索选项窗格中，你可以精制搜索规则。下一步你可以双击搜索页下部的主题列表中的项目来显示 HTML 手册中相应的段落：



搜索结果页显示了一个 HTML 页，在这个页中你通常可以通过超链接来操作。页顶部的按钮可以被用于操纵你以前要访问的上一个或下一个位置，还可以取消这个页组合，或者在操纵之后转到原始页。

你可以回到搜索页来搜索不同的关键字或者转到书籍页来选择不同的书籍。每个搜索都创建了一个不同的标签页，所以你可以保存不同的有用的搜索结果。要关闭结果页，请按标签页右上方的关闭按钮。要关闭所有页，请按搜索页中的清除结果按钮。

在默认的情况下，HTML 窗口将停留在 PL/SQL Developer IDE 中的其他窗口的上方，所以信息的停留是可见的。要临时隐藏窗口，你可以使用窗口右上方的绿色的卷起按钮。

## 33. 定制

PL/SQL Developer 的一些方面可以被定制以满足你的个人需要。这些方面在下列段落中有描述。

### 33.1 首选项

在工具菜单中，有一个首选项项目允许你为 PL/SQL Developer 设置不同的首选项。这些首选项在第 20 章中有详细描述。

### 33.2 窗口版面

有多种窗口版面可以根据你的需要来制作，在 *窗口*菜单中选择 *保存版面*项目将保存这些版面。版面设置包括：

- 应用程序窗口的大小和位置。
- 对象浏览器的存在和大小。
- 程序窗口中代码目录的存在和大小。
- 窗口列表和模板窗口的存在、状态（浮动或停放）和位置。
- 程序窗口、测试窗口、SQL 窗口、解释计划窗口、命令窗口、报告窗口、编译无效对象窗口、查找数据库对象窗口、导出用户对象窗口和比较用户对象窗口的大小。

当你保存版面时，最后打开的窗口的大小就被保存了。如果某一类型的窗口不存在，则那个窗口类型的大小设置就保持不变。

### 33.3 在线文档

大多数的 Oracle 文档都被提供为在线文档。随着时间的推移，这些文档已经有了几种格式。到目前为止，已经有了 MS 帮助文件、Oracle 书籍文件、Adobe PDF 文件和 HTML 文件。在未来，Oracle 也许还会选择使用另外一种格式。

此外，你可以有一些共同的标准和有时要访问的文字处理软件格式的工程文档。

通过在文档菜单中包括它们，所有这些在线文档都能被集成到 PL/SQL Developer 的 IDE 中。这样，通过一个简单的鼠标单击，它们对你就是可用的了。

要配置在线文档，请选择工具菜单中的**配置文档**项目。下列对话框允许你定义每个文档的描述，以及怎样去查看它：



在右边，你可以看到四个按钮：在菜单中创建新文档、删除文档和上移、下移文档。当你创建或修改一个文档时，你必须提供下列信息：

描述在菜单和工具栏中显示。要为文档的菜单项目创建一个快捷键，就要在描述中的这个字母的前面加上一个 & 前缀。如果你要在描述中使用 & 字符，那么就用两个 & 字符以代之。如果你只输入一个“-”字符到描述中，那么它将变为菜单中的分隔线。这样，你可以把文档分到合理的组中。

可执行文件应该是你要用来查看文档的带有完整路径的程序。如果文档的类型是一个已被注册的文件类型（象 MS 帮助文件或 HTML 文件），你就可以在这中输入带路径的文档。

参数可用于传递信息给可执行文件（如果它是程序的话）。它至少应该包括带路径的文档，但也能用于传递选项给程序。

当**作为主菜单项目的文档**选项被检验时，**文档**菜单就被定位在主菜单中。如果它没有被检验，它将被定位在**工具**菜单的下面。

在上面的例子中，PL/SQL Reference Manual 是一个 PDF 文件 (P:\09DOC\A89856\_01.pdf)，它是一个已注册的文件类型，所以只能在可执行文件输入框输入文件名。字母 'P' 被用于菜单中的快捷方式。文档菜单将像这样：



当你控制鼠标光标越过工具栏按钮时，相应的提示将显示出来。如果你保留这个描述为空，则文档的主描述（与在菜单中显示的一样）将被使用。按图标按钮来为菜单和工具栏按钮选择一个 Windows 位图文件 (\*.bmp)。这个图象的大小应该是 16 x 16 像素比较合适。注意，PL/SQL Developer 将始终从原始位置载入位图文件，因此若相应的文档没有变动你就不要删除或重新命名这个文件。PL/SQL Developer 带有许多标准的位图文件，你可以从那中挑选它们。这些文件位于 PL/SQL Developer 安装目录中的 Icons 子目录中。这是图片选择器默认的目录。

## 33.4 命令行参数

下列参数可以被用于 PL/SQL Developer 命令行。默认的快捷方式不包括任何参数，只不过从 PL/SQL Developer 安装目录中启动 plsqldev.exe 而已。

通过 PL/SQL Developer 安装目录中的 *params.ini* 文件，默认参数可以被定义。你可以在像记事本这样的文本编辑器中编辑这个文件，这个文件包含了每个参数的说明。如果参数在 *params.ini* 中和在命令行上被定义，那么命令行具有优先地位。

### **userid**

每次你启动 PL/SQL Developer，它将提示你至少要输入一个口令。要避免这个，你可以提供一个常见的用户名/口令@数据库格式的 userid 参数：

```
plsqldev.exe userid=scott/tiger@chicago
```

注意，你还可以使用下列的注册表键提供一个默认的登录方法：

```
HKEY_CURRENT_USER\Software\Allround Automations\PL/SQL Developer\Logon
```

在这中，你可以添加 Username、Password 和 Database。

后一个方法的优点是，在双击了 PL/SQL Developer 已注册的文件之后，它能使你自动登录。

### **edition**

设定登录基于版本的重新定义之后的版本。例如：

```
plsqldev.exe userid=scott/tiger@chicago edition=R13
```

### **workspace**

设定登录工作区管理之后的工作区，例如：

```
plsqldev.exe userid=scott/tiger@chicago workspace=WS12
```

### **nologon**

当 PL/SQL Developer 被启动时，Nologon 参数禁用了被显示的登录对话框。这个参数不需要任何自变量：

```
plsqldev.exe nologon
```

### **oraclehome**

你可以在命令行上指定 Oracle 的主目录名，从而越过主 Oracle 主目录（被默认使用）和 Oracle 主目录首选项：

```
plsqldev.exe oraclehome=ora817
```

注意，这是 Oracle 主目录的名称，被指定在注册表中的 ORACLE\_HOME\_NAME 键中。这个名称也被 Oracle 主目录选择器使用，它不是注册表的组名或目录名。

#### **dontadjustpath**

PL/SQL Developer 将不为 Oracle Net 的初始化而临时修改 PATH：

```
plsqldev.exe dontadjustpath
```

#### **nosplash**

当 PL/SQL Developer 被启动时，Nosplash 参数禁用了闪屏。这个参数不需要任何自变量：

```
plsqldev.exe nosplash
```

#### **noplugins**

指定 noplugins 参数将防止任何插件被载入：

```
plsqldev.exe noplugins
```

#### **library**

指定报告样式库的位置：

```
plsqldev.exe library=p:\standard.lib
```

#### **prefpath**

指定个人首选项集的位置。例如：

```
plsqldev.exe prefpath=u:\userdata
```

要了解详细资料，请参见第 20.33 节。

#### **defaultprefpath**

指定默认首选项集的路径。例如：

```
plsqldev.exe defaultprefpath=p:\common\plspref
```

#### **commandfile**

在命令窗口中运行指定的命令文件。例如：

```
plsqldev.exe userid=scott/tiger commandfile="u:\sql scripts\demo_build.sql"
```

这个命令将首先链接到数据库然后运行 *demo\_build.sql*。

#### **viewobject**

查看指定数据库对象。它需要你也指定一个 *userid* 参数。例如：

```
plsqldev.exe userid=scott/tiger@chicago viewobject=emp
```

这个命令将首先链接到数据库，然后显示表 *EMP* 的表定义。注意，对象说明也可能包括所有者（例如 *SCOTT.EMP*）。

#### **editobject**

编辑指定数据库对象。它需要你也指定一个 *userid* 参数。例如：

```
plsqldev.exe userid=scott/tiger@chicago editobject=emp
```

这个命令将首先链接到数据库，然后显示表 *EMP* 的表定义。注意，对象说明也可能包括所有者（例如 *SCOTT.EMP*）。

#### **registry**

要使用不同的 IDE 设置，你可以使用 *registry* 参数。例如：

```
plsqldev.exe registry=home
```

这个命令将导致所有的 IDE 设置（窗口大小和位置、文件历史、登录历史等）从 *home* 名下被载入和被保存于 *home* 名下。

#### **sessionmode**

允许你定义会话模式：单会话、双会话、多会话。有效值包括 SINGLE、DUAL、MULTI 或 1、2、3。例如：

```
plsqldev.exe sessionmode=dual
```

#### **noscwarning**

提供此参数时，在服务合同即将过期时，你将不会收到警告：

```
plsqldev.exe noscwarning
```

#### **hexedit**

此参数将为指定文件调用大数据编辑器的十六进制编辑器。例如：

```
plsqldev.exe hexedit=c:\temp\blob.dat
```

#### **project**

打开指定工程。例如：

```
plsqldev.exe project=u:\userdata\projects\deptemp.prj
```

如果在指定 \* 作为工程名，则在启动过程中将显示工程选择器：

```
plsqldev.exe project=*
```

## 断开链接列表

在启动时禁用连接列表，会弹出旧式登录对话框。例：plsqldev.exe noconnectionlist

## instantclient

使用给定目录的即时客户端，例如：

```
plsqldev.exe instantclient=C:\Oracle\InstantClient11g
```

如果目录中包含 *tnsnames.ora* 文件或 *network* 子目录，或是包含 *tnsnames.ora* 文件的 *network\admin* 子目录，则将隐式使用该目录。

## tns\_admin

使用给定目录作为 Oracle network 目录（即 *tnsnames.ora* 文件所在目录）。例如：

```
plsqldev.exe tns_admin=C:\Oracle\Network\Admin
```

## nls\_lang

使用给定语言、地区和字符集，例如：

```
plsqldev.exe nls_lang=american_america.we8mswin1252
```

## env:[varname]

将环境变量设置为给定值，例如：

```
plsqldev.exe env:nls_date_format=dd-mm-yyyy
```

## NoLoadDesktop

此参数阻止在 PL/SQL Developer 启动时自动重新打开那些以前已经打开的窗口。它仅在自动保存桌面首选项启用时适用（参见第 20.10 节）。

## InstanceName

如果正在使用多显示器安装或分屏安装，可以使用此参数来保存和恢复针对特定实例的窗口位置。例如：

```
plsqldev.exe InstanceName=LEFT
```

## 33.5 SQL、PL/SQL、和命令关键字

在 SQL 、PL/SQL 和命令编辑器中高亮显示的关键字可以被定义在一个关键字文件中。在 PL/SQL Developer 的安装目录中，你可以找到一个名为 default.kwf 的文本文件。这个文件包含了一个『PL/SQL』关键字版块、一个『SQL』关键字版块和一个『命令』文件关键字版块。你可以使用像记事本这样的文本编辑器在适当的地方简单地更改、添加或删除关键字。

## 33.6 插件

你可以规划你自己的插件以扩展 PL/SQL Developer 的功能。插件的功能可以添加到 PL/SQL Developer 的菜单中，它可以执行任何种类的任务：访问数据库、对象浏览器、当前窗口和编辑器等。

插件是 DLL 文件，它有特定的接口与 PL/SQL Developer 链接。要创建一个插件，你可以使用任何能创建 DLL 文件的程序语言。当插件 DLL 文件被放到 PL/SQL Developer 的插件目录中时，它将自动被拾取。这使你分发你的插件变得很容易。

你可以建立你自己的或公用的插件。Allround Automations 网站上的一些标准的插件也是可以利用的 (<http://www.allroundautomations.com/plsqldev.html>)。在这中，第三方插件也是可用的。

在 PL/SQL Developer 根目录下的 *PluginDoc* 子目录中，插件文件和例子也是可用的。