计算机最新技术培训教材

Oracle Developer 2000 学习教程

李瑞轩　杨曼红　王俊　编著

北 京 大 学 出 版 社

北　　京

需要全本请访问 www.ertongbook.com

内 容 提 要

Oracle Developer 2000 是 Oracle 公司新推出的一个集成式的快速应用开发工具集，本书是一本介绍 Oracle Developer 2000 的学习教程。

本书共分 5 个部分，第 1 部分介绍 SQL 语言和 Oracle 基本工具 SQL*PLUS 的使用，第 2 部分介绍 Oracle PL/SQL 的使用，第 3 部分介绍使用 Oracle Forms 开发基于格式的应用，第 4 部分介绍使用 Oracle Reports 产生报表和报告，第 5 部分介绍使用 Oracle Graphics 生成图表，在附录中对 Oracle Developer 2000 的其他组件作了简略介绍。

本书结构清晰、组织合理、讲解细致深入，并通过丰富的范例加深读者对相应技术的理解。本书既可作为 Oracle 应用开发人员的使用参考书，也可作为 Oracle 的培训教材，同时也是大专院校相关专业师生的自学、教学参考用书。

图书在版编目（CIP）数据

Oracle Developer 2000 学习教程/李瑞轩，杨曼红编著．—北京：北京大学出版社，2000.7
（计算机最新技术培训教材）
ISBN 7-301-01681-6

I．O… II．李… III．关系数据库—数据库管理系统，Oracle Developer IV．TP311.13

中国版本图书馆 CIP 数据核字（2000）第 06667 号

书　　名：Oracle Developer 2000 学习教程
著作责任者：李瑞轩　杨曼红　王　俊
责 任 编 辑：黄庆生　汉　明
标 准 书 号：ISBN 7-301-01681-6/TP·105
出　版　者：北京大学出版社
地　　址：北京市海淀区中关村北京大学校内　100871
网　　址：http://cbs.pku.edu.cn
电 子 信 箱：xxjs@pup.pku.edu.cn
排　版　者：南方立德（Leader）信息技术中心
印　刷　者：
发　行　者：北京大学出版社
经　销　者：新华书店
787 毫米×1092 毫米　16 开本　22.375 印张　542 千字
2000 年 7 月第 1 版　2000 年 7 月第 1 次印刷
定　　价：35.00 元

前　　言

Oracle Developer 是 Oracle 公司提供的一个集成式的快速应用开发工具集，它支持企业级的高级应用开发需求。应用开发具有高度的可伸缩性，要维护不断变化的应用需求以及真实环境中的异构性，开发基于数据库的 Form、Reports 和 Graphics 应用，并能够实现 Internet 计算环境的种种复杂应用。Oracle Developer 纳入了一个集成式构造器集合，通过使用快速应用开发（RAD）技术、面向对象技术并通过基于计算机的内部学习，以及统一的客户、应用服务器和数据库服务器体系结构，为客户/服务器和 Web 应用开发建立了易于使用的新标准。

Oracle Developer 的嵌入式特性允许客户在 10 到 1000 个用户、MB 到 GB 级数据，决策支持到复杂 OLTP 应用间自由伸缩。同时 Oracle Developer 灵活的多层体系结构还支持 Oracle 应用服务器或第三方应用服务器，如事务处理监视器，并因此确保了从桌面级到专业级的可伸缩性。在 Oracle Developer 的其他应用和工具间的交互是建立在 OCX/ActiveX 控件、OLE（对象连接于嵌入）以及 DDE（动态数据交换）基础之上的。通过开放 API 对多种多媒体格式的补充支持，Oracle Developer 使开发人员能够灵活地扩充应用，并能将其他部件集成在这些应用之中。通过使用 Oracle Developer Server 可以很方便地把应用实施于多层结构环境下，把复杂的客户/服务器应用快速地实施于安全、易于管理的应用服务器结构之中。

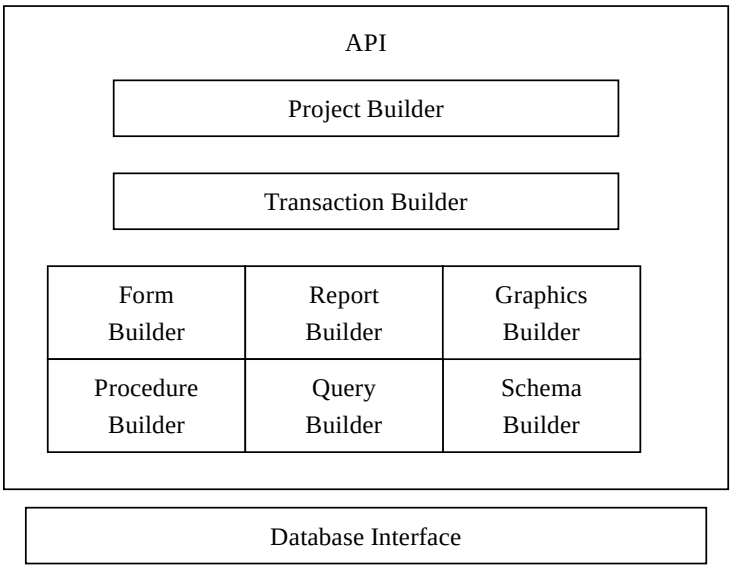


图 1　Oracle Developer 组件的体系结构

Oracle Developer 所包含的组件有：Oracle Forms 5.0、Oracle Reports 3.0、Oracle Graphics 3.0、Procedure Builder 2.0、Project Builder、Query Builder、Schema Builder 和 Transaction Builder。这些组件的体系结构如图 1 所示。Oracle Developer 组件具有如下特性：

- 提供了一个基于向导的易于使用的设计界面，它有对象导航器、布局编辑器、属性选项板及其他所见即所得的 GUI 界面。
- 使用一致的对象导航工具，其直观的结构化浏览忽然编辑界面使得开发人员能够容易地定位、检查和控制所有的应用对象，包括数据库对象。
- 纳入了一个集成式高效易用的构造器集，用于创建表格、报告、图表、查询、数据库对象和过程。
- 紧密集成了 SQL 和 Oracle PL/SQL 语言，方便访问数据库和开发数据库存储过程。
- 开发出来的应用具有较强的可伸缩性和好的可移植性。
- 支持多数据源的处理，适用于 Internet 环境下的网络计算体系结构。

本书共分五部分，第一部分介绍了 SQL 语言和 Oracle 基本工具 SQL*PLUS 的使用，第二部分介绍了 Oracle PL/SQL 的使用，第三部分介绍了使用 Oracle Forms 开发基于格式的应用，第四部分介绍了使用 Oracle Reports 产生报表和报告，第五部分介绍了使用 Oracle Graphics 生成图表，还在附录中对 Oracle Developer 的其他组件作了简略介绍。

本书中所采用的插图是在 Oracle Developer 开发工具软件的中文版上制作的，但为了方便使用英文版的读者，书中涉及到与界面有关的内容均采用中英文一起给出。如对象导航器则用“对象导航器（Object Navigator）”，文本项则用“文本项（Text Item）”给出。其中 Form 除外，由于 Form 有其特殊的含义，中文“格式”或“表格”难于准确表达其意思，故书中一般就使用“Form”。

此外，在本书中使用了以下两个数据库表实例：EMP 表和 DEPT 表，其记录值如下：

表 1　EMP 表

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1001	ZHANG	CLERK	4096	26-DEC-98	1500	0	101
1002	LING	SALESMAN	1105	15-OCT-97	1800	1200	105
1003	LIU	SALESMAN	1105	07-AUG-97	2200	1600	105
1006	JUN	PROGRAMER	2025	17-FEB-99	3900	1500	102
1009	CAO	MANAGER	5032	11-NOV-96	3500	0	101
1010	TANG	SALESMAN	1105	24-JAN-98	1600	1200	105
1021	TU	ANALYST	1009	03-JUL-97	2200	1000	101
1105	JIE	MANAGER	5032	16-MAR-97	3900	2000	105
1059	BING	PROGRAMER	2025	28-SEP-99	3500	1500	102
2010	MAI	SECRETARY	5032	25-AUG-98	3000	0	101
2025	GUI	MANAGER	5032	19-JAN-98	3900	1600	102
2076	ZHEN	SALESMAN	1105	16-JUL-99	1800	1200	105
2095	FANG	PROGRAMER	2025	01-JUL-99	3500	1500	102
4096	XIANG	ANALYST	1009	21-JAN-98	2400	1000	101
5032	DING	PRESIDENT	-	18-SEP-96	8000	0	101

表 2　DEPT 表

DEPTNO	DNAME	LOC
101	ACCOUNTING	WUHAN
102	RESEARCH	BEIJING
105	SALES	SHANGHAI
110	OPERATIONS	SHENZHEN

本书结构清晰，组织合理，讲解细致深入，并通过丰富的范例加深读者对相应技术的理解。本书既可作为 Oracle 应用开发人员的使用参考书，也可作为 Oracle 的培训教材，同时也是大专院校相关专业师生的自学或教学参考用书。

本书由孙景利策划，李瑞轩、杨曼红、王俊主编。作者均为 Oracle 联合大学的教员，长期从事 Oracle 数据库系统管理和应用开发的培训工作，曾参与多项 Oracle 数据库应用的开发，具有丰富的教学和实践经验。另外，肖卫军、吴炜、陆谊、丁雨、黄少棠、瞿小玉、黄瀚华、凌贤伍、孙敬娜、付红梅、康孟霞、张小东、李宁、王强、赵四海、李晓峰、董团结、杨仕润、韩百、涂海滨、张旭、张志明、朱黎、周刚兵、张华开、王登峰、郑忠良、李静、刘天翠等也参加了全书的编写工作，在这里对他们表示诚挚的感谢。

由于时间紧迫，加之笔者水平有限，书中如有疏漏之处，请广大读者批评指正。

编　者

2000 年 7 月

需要全本请访问 www.ertongbook.com

目　　录

第一部分　Oracle SQL*PLUS 1

第 1 章　SQL*PLUS 基础 1

1.1　启动 SQL*PLUS 1
1.2　SQL 和 SQL*PLUS 命令 3
1.3　SQL*PLUS 操作命令 7
1.4　退出 SQL*PLUS 16

第 2 章　结构化查询语言 SQL 17

2.1　数据查询命令 17
2.2　数据插入命令 21
2.3　数据删除命令 22
2.4　数据修改命令 23
2.5　事务控制命令 24
2.6　数据类型 26
2.7　数据定义语言 32

第 3 章　格式化查询结果 40

3.1　环境设置命令 40
3.2　格式化列 42
3.3　使用空行和统计行 46
3.4　定义页标题和大小 50
3.5　存储和打印查询结果 53

第 4 章　SQL*PLUS 的高级特性 55

4.1　使用函数 55
4.2　表的连接 60
4.3　子查询 62
4.4　集合运算 65

第 5 章　数据库安全管理 67

5.1　数据库安全性概念 67

5.2 数据库的系统特权 68
5.3 数据库的用户管理 69
5.4 数据库实体的存取控制 72

第二部分 Oracle PL/SQL 75

第 6 章 PL/SQL 简介 75

6.1 为什么引入 PL/SQL 75
6.2 PL/SQL 的优点 75
6.3 PL/SQL 体系结构 76
6.4 PL/SQL 块 77

第 7 章 PL/SQL 基础 80

7.1 标识符 80
7.2 表达式 81
7.3 变量和常量 81
7.4 数据类型 83
7.5 注释 86

第 8 章 数据操纵语句 87

8.1 数据查询语句（SELECT INTO 语句） 87
8.2 数据插入、更新与删除语句 88
8.3 事务控制语句 90

第 9 章 控制结构 93

9.1 条件控制语句（IF 语句） 93
9.2 循环控制语句（LOOP 语句） 94
9.3 跳转控制语句 95

第 10 章 游标的使用 98

10.1 游标的说明 98
10.2 游标的管理 99
10.3 游标的属性 101
10.4 游标 FOR 循环 104

第 11 章 例外处理 105

11.1 例外的概念 105
11.2 预定义例外 105
11.3 用户自定义例外 107

第 12 章 子程序.....110
12.1 过程.....110
12.2 函数.....112
第 13 章 程序包.....115
13.1 程序包的组成.....115
13.2 程序包的调用.....117
第三部分 Oracle Form.....119
第 14 章 Form 基本概念.....119
14.1 什么是 Form.....119
14.2 Form 模块的结构.....120
14.3 Form 模块的层次结构.....121
第 15 章 Form 的工作方式.....123
15.1 Form 应用模块的生成.....123
15.2 运行 Form 模块.....127
15.3 Form 工作模式.....127
15.4 查询数据.....128
15.5 插入、更新和删除数据.....132
15.6 提交和回滚事务.....134
第 16 章 Form 设计工具.....136
16.1 Form 设计界面.....136
16.2 对象导航器.....137
16.3 布局编辑器.....146
16.4 属性选项板.....150
16.5 其他 Form 设计工具.....155
第 17 章 基本 Form 设计.....156
17.1 块和项.....156
17.2 开发 Form 的一般步骤.....158
17.3 建立基表块.....159
17.4 建立主从块.....163
17.5 建立控制块.....165

第 18 章　常用界面项的设计 167

18.1　建立文本项 167
18.2　建立复选框 174
18.3　建立列表项 177
18.4　建立单选组 182
18.5　建立显示项 184
18.6　建立图像项 186
18.7　建立按钮 189

第 19 章　其他常用功能设计 192

19.1　建立值列表（LOV）和记录组 192
19.2　建立编辑器 197
19.3　建立报警器 199

第 20 章　窗口和画布视图 203

20.1　窗口和画布视图的分类 203
20.2　建立窗口 204
20.3　建立内容画布视图 206
20.4　建立堆叠画布视图 208
20.5　建立工具条画布视图 210
20.6　建立标签画布视图 211

第 21 章　Form 中的触发器 215

21.1　Form 中的触发器的基本概念 215
21.2　Form 中的触发器类型 217
21.3　建立触发器 222

第 22 章　触发器编程 225

22.1　编写触发器 225
22.2　内部子程序 227
22.3　验证 231
22.4　输入项触发器 235
22.5　非输入项触发器 236
22.6　使用变量和参数 238

第 23 章　定制菜单 241

23.1　菜单编辑器 241
23.2　建立菜单模块 242

第四部分 Oracle∗Reports 247

第 24 章 Oracle∗Reports 简介 247

24.1 概述 247
24.2 两种报表创建方法 248
24.3 对象导航器 256

第 25 章 数据模型 259

25.1 功能及构成 259
25.2 数据模型编辑器 259
25.3 查询 260
25.4 组 261
25.5 列 264
25.6 数据连接 267
25.7 参数 269

第 26 章 布局 272

26.1 布局功能 272
26.2 布局模型编辑器 272
26.3 报表区域 273
26.4 基本布局对象 273
26.5 几种缺省的报表布局风格 283

第 27 章 参数表格 288

27.1 建立缺省的参数表格 288
27.2 建立用户定义的参数表格 289

第 28 章 Oracle∗Reports 触发器 291

28.1 Oracle∗Reports 中的 PL/SQL 291
28.2 Oracle∗Reports 中的 PL/SQL 块 292
28.3 Oracle∗Reports 触发器 294
28.4 SRW 包 300

第 29 章 矩阵式报表 302

29.1 建立矩阵式报表的需求 302
29.2 单查询矩阵式报表 302
29.3 多查询矩阵式报表 304
29.4 用零替代无值单元格 307
29.5 具有合计的矩阵式报表 307

29.6 矩阵间断式报表 308

第五部分 Oracle*GRAPHICS 310

第 30 章 Oracle*Graphics 简介 310

30.1 概述 310
30.2 Oracle*Graphics 的对象 311
30.3 Oracle*Graphics 模块 312

第 31 章 图形应用的定义 313

31.1 目的 313
31.2 定义图形应用的步骤 313

第 32 章 Oracle*Graphics 中的触发器 325

32.1 分类 325
32.2 封装例程 325
32.3 布局格式触发器 326
32.4 按钮触发器 326
32.5 过滤触发器 327
32.6 定时触发器 328

第 33 章 应用实例 329

33.1 创建主从关系 329
33.2 创建格式触发器 334
33.3 创建多个显示层 336

第 34 章 Form、Reports、Graphics 模块间的相互调用 338

34.1 从 Form 模块中调用 Reports 模块 338
34.2 Form 与 Graphics 模块间的相互调用 339
34.3 Reports 和 Graphics 间的集成 340

附录 Oracle Developer 其他组件简介 342

需要全本请访问 www.ertongbook.com

第一部分　Oracle SQL*PLUS

第 1 章　SQL*PLUS 基础

本章介绍使用 SQL*PLUS 的基础，包括以下内容：

- 启动和退出 SQL*PLUS。
- 运行 SQL 命令和 SQL*PLUS 命令。
- 理解影响运行命令的变量。
- 自动将所作修改保存到数据库。
- 停止运行的命令。
- 收集运行命令的时间统计。
- 列出表的定义。
- 控制显示。
- 解释出错信息。

1.1　启动 SQL*PLUS

1.1.1　SQL*PLUS 简介

SQL*PLUS 是 ORACLE 数据库的一个基本应用工具。SQL*PLUS 命令可以与 SQL 数据库语言、过程语言的扩充 PL/SQL 块（将在第二部分介绍）一起使用。SQL 数据库语言可存储和检索 ORACLE 中的数据。PL/SQL 可通过过程化逻辑将几个 SQL 命令连接起来。SQL*PLUS 可以操作 SQL 命令和 PL/SQL 块以及执行许多附加的任务。通过使用 SQL*PLUS 可做以下一些工作：

- 输入、编辑、检索和运行 SQL 命令和 PL/SQL 块。
- 格式化查询结果，对查询结果进行计算、存储和以报表形式打印查询结果。
- 列出表的列定义。
- 在 ORACLE 数据库间存取和拷贝数据。
- 运行存储在数据库中的子程序或包。

➢ 从终端用户接收应答或将信息发送给另一终端用户。

SQL*PLUS、SQL 和 PL/SQL 命令语言功能强大且容易使用，既可以满足具有数据库使用经验的用户的需要，也可以满足新用户的要求。

这里所介绍的 SQL*PLUS 适用于所有操作系统。Oracle 和 SQL*PLUS 可运行在许多种计算机上。在使用之前，由 DBA 负责安装 Oracle 和 SQL*PLUS。如果要利用 SQL*PLUS 3.1 及以上版本的高级功能，则后台数据库必须是 Oracle7 或以上版本。在多用户系统中，由 DBA 为用户建立进入 SQL*PLUS 的用户名及口令。要进入操作系统必须要有操作系统用户名和口令，这两者可以相同也可以不同。

1.1.2　启动 SQL*PLUS

在启动 SQL*PLUS 之前，首先要将 Oracle RDBMS 安装到计算机上，启动计算机进行操作系统登录并启动 Oracle 数据库系统，然后才能运行 SQL*PLUS。

执行 SQL*PLUS 有两种情况，一种是在操作系统提示符下，另一种是在网络的客户端运行 Oracle SQL*PLUS 的客户端程序。

在操作系统提示符（假设为＄）下运行 SQL*PLUS：

＄sqlplus【Return】

SQL*PLUS 将显示其版本号、日期和版权信息，并提示你输入用户名：

请输入用户名：用户名【Return】

输入一个合法的用户名之后，接着提示你输入用户口令：

请输入口令：用户口令【Return】

如果用户口令输入正确，SQl*PLUS 将宣布你已经与 Oracle 数据库连接成功，并在屏幕上显示 SQl*PLUS 提示符：SQL＞。

如果用户名或口令输入有误，在向 Oracle 数据库注册失败三次后，屏幕上将出现终止服务信息，并退回到操作系统提示符下。

还可以直接在操作系统提示符下输入用户名和口令：

＄sqlplus 用户名／用户口令【Return】

如果是在网络客户端启动 SQL*PLUS，则其登录界面如图 1-1 所示。在网络客户端启动 SQL*PLUS 时，不仅要给出用户名和用户口令，还要给出主机连接字符串。它也可以以如下的形式给出：

＄sqlplus 用户名／用户口令@主机连接字符串【Return】



图 1-1　SQL*PLUS 网络客户端登录界面

其中主机连接字符串指定了 Oracle 数据库服务器所在的位置和正在运行的数据库实例，它在 SQL*NET 的客户端程序中进行配置，如在 Windows 95/98/NT 中，需要在“SQL*NET EASY CONFIGURATION”中进行设置，如图 1-2 所示。一个典型的主机连接字符串包括连接字符串的名字、所使用的网络通信协议及网络地址以及希望连接的数据库的实例名。

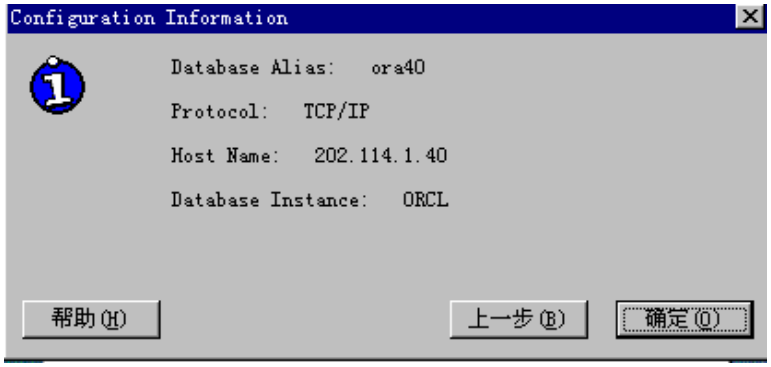


图 1-2　SQL*NET EASY CONFIGURATION 配置界面

1.2　SQL 和 SQL*PLUS 命令

SQL*PLUS 可以处理三种类型的命令：SQL 命令、PL/SQL 块和 SQL*PLUS 命令，功能分别如下所述：

- SQL 命令：它以数据库中的信息为操作对象，主要使用数据定义语言和数据操纵语言来对数据库进行操作。
- PL/SQL 块：也是以数据库中的信息为操作对象，在 SQL 语言中加入一些过程控制以实现对数据库的操作。
- SQL*PLUS 命令：主要用来格式化查询结果、设置选择、编辑及存储 SQL 命令，以设置查询结果的显示格式，并且可以设置一些环境选项。

1.2.1　SQL 命令

SQL 命令主要用于创建、存储、改变、检索和维护 Oracle 数据库中的信息。当输入 SQL 命令后，SQL*PLUS 将其保存在内部缓冲区中，这个内部缓冲区称为 SQL 缓冲区。SQL 缓冲区只保存最近输入的一条 SQL 命令，用户可以编辑、运行保存在 SQL 缓冲区中的 SQL 命令。SQL 命令语言可以操纵数据库中的数据。

例 1-1　显示 EMP 表中工资低于 3000 的职工的职工号、姓名、职务和工资。

在 SQL 命令提示符下输入命令的第一行：

```
SQL>SELECT EMPNO,ENAME,JOB,SAL
```

如果在输入过程中有错误，可用退格键【Backspace】删除它，然后重新输入。完成输入后按【Return】键，将移到下一行。SQL*PLUS 将显示行号“2”，作为第二行的提示符，

此时可输入命令的第二行：

```
2  FROM EMP WHERE SAL<3000;
```

当 SQL 命令输入完成时，可以输入分号“;”并按【Return】键，则结束 SQL 命令的输入，并运行 SQL 命令，并在屏幕上显示结果，结果形式如表 1-1 所示。

表 1-1 EMP 表中工资低于 3000 的职工的职工号、姓名、职务和工资

EMPNO	ENAME	JOB	SAL
1001	ZHANG	CLERK	1500
1002	LING	SALESMAN	1800
1003	LIU	SALESMAN	2200
1010	TANG	SALESMAN	1600
1021	TU	ANALYST	2200
2076	ZHEN	SALESMAN	1800
4096	XIANG	ANALYST	2400

```
7 records selected
SQL>
```

在显示查询结果及记录数之后，SQL*PLUS 再次显示命令提示符。如果在上述输入中有错误，则不能得到上述结果，此时可重新输人命令。

标题在输出中可以重复，它依赖于系统变量 PAGESIZE（页大小）的设置。是否可以看到检索的信息决定于系统变量 FEEDBACK（反馈信息）的设置。

在 SQL*PLUS 中输入 SQL 命令，可将 SQL 命令划分为不同行，可在你所希望的任何点——只要不在单个词中间分开即可，例如，上述例子可分多行输入：

```
SQL>SELECT
2   EMPNO,ENAME,JOB,SAL
3   FROM  EMP
4   WHERE  SAL<3000
5   ;
```

将 SQL 命令划分为多个子句，每个子句为一行，这样划分具有良好的可读性。

当 SQL 命令输入完毕后，可用下列三种方法之一结束 SQL 命令：

- 用一个分号“;”。告诉 SQL*PLUS 要运行命令，在命令的最后一行的末端输入分号，在输入分号后，按下【Return】键，SQL*PLUS 将处理该命令。
- 在独自一行上用斜杠（/）。在独自一行上的斜杠，同样告诉 SQL*PLUS 要运行该命令，在输入斜杠后再按【Return】键，SQL*PLUS 将执行命令。
- 用空行。空行也可以告知 SQL*PLUS 完成输入命令，但不运行该命令。在命令的最后一行的末端按【Return】键，SQL*PLUS 用另一行提示，再按【Return】键，SQL*PLUS 将用 SQL*PLUS 命令提示符提示。SQL*PLUS 并不执行命令，而是将命令存储在 SQL 缓冲区中。

SQL 缓冲区是为 SQL*PLUS 存储最近输入的 SQL 命令区。命令一直保存在缓冲区中，直至输入新的命令。因此，可以对当前 SQL 缓冲区中的命令进行编辑或重新执行，而不需要重新输入。SQL*PLUS 在命令缓冲区中不存储分号和斜杠（/）。SQL 缓冲区是缺省缓冲

区，用户可以定义其他的缓冲区，但一般来说，SQL*PLUS 不需要使用多个缓冲区。在本书中，除特殊注明外，缓冲区和 SQL 缓冲区是同义词。

在命令提示符后键入 RUN 命令或 /（斜杠）命令，可运行或重新运行当前 SQL 命令。RUN 命令在执行命令前列出缓冲区中的 SQL 命令，/（斜杠）命令简单地运行 SQL 命令。

1.2.2　运行 PL/SQL 块

可用 PL/SQL 程序（称为块）操作数据库中的数据。PL / SQL 块从 DECLARE、BEGIN 开始，或者从块名开始。除了分号和空行不能终止或执行 PL/SQL 块外，SQL*PLUS 将按处理 SQL 命令一样的格式处理 PL/SQ 块。终止 PL/SQL 块通过在新行中输入“.”实现。SQL*PLUS 将输入的 PL/SQL 块存储在 SQL 缓冲区，可通过输入 RUN 或 / 命令执行当前块，SQL*PLUS 将完整的 PL / SQL 块一起发送给 Oracle RDBMS，从而减少了和数据库连接的次数，提高了执行效率。

例 1-2　输入并执行 PL / SQL 块。

```
SQL>DECLARE
2  x  NUMBER:=200;
3  BEGIN
4   FOR i IN 1..10 LOOP
5     IF TRUNC(i/2)=i/2 THEN     – –i 是偶数
6       INSERT INTO test VALUES(i,x,'i is even');
7     ELSE
8       INSERT INTO test VALUES(i,x,'i is odd');
9     END IF;
10    x:=x+50;
11   END LOOP;
12 END;
13
SQL> /
PL/SQL procedure successfully completed.
```

有关 PL/SQL 的详细内容将在第二部分讲述。

1.2.3　SQL*PLUS 命令

SQL*PLUS 命令可用于设置查询结果的输出格式，形成复杂的报表，编辑 SQL 命令，设置环境变量，并可提供帮助信息。

SQL*PLUS 的很多命令可以简写成一个或几个字母。SQL*PLUS 命令的输入与 SQL 命令不同，在输入 SQL*PLUS 命令过程中，按【Return】键表示输入完成，而不必输入分号“;”。如果命令输入占满一行，光标会自动移到下一行，这时可以继续输入，直到按【Return】键结束输入。

例 1-3　利用 SQL*PLUS 命令改变显示 EMP 表的 SAL 列的格式。

（1）在命令行中输入 SQL*PLUS 命令：

```
SQL>COLUMN SAL FormAT $99,999 HEADING SALARY;
```

如果在输入过程中有错误，可用退格键【Backspace】删除它，然后重新输入。当输入一行后，按【Return】键，SQL*PLUS 将记录这个系统，并再次显示 SQL*PLUS 提示符，准备接收新命令。

（2）输入 RUN 命令，重新运行最近的查询，SQL*PLUS 将重新处理查询和显示结果。

```
SQL>RUN
1  SELECT EMPNO,ENAME,JOB,SAL
2  FROM EMP WHERE SAL<3000;
```

运行的结果见表 1-2。

表 1-2　格式化 EMP 表中工资低于 3000 的职工的职工号、姓名、职务和工资

EMPNO	ENAME	JOB	SALARY
1001	ZHANG	CLERK	$1500
1002	LING	SALESMAN	$1800
1003	LIU	SALESMAN	$2200
1010	TANG	SALESMAN	$1600
1021	TU	ANALYST	$2200
2076	ZHEN	SALESMAN	$1800
4096	XIANG	ANALYST	$2400

COLUMN 命令用$符号和逗号格式化 SAL 列，并给它换上一个新的标题。SQL*PLUS 不在缓冲区中存储 SQL*PLUS 命令。

SQL*PLUS 命令有不同于 SQL 命令和 PL/SQL 块的语法，一个长的 SQL*PLUS 命令的续行的处理方式如下：

（1）在行的末尾输入连接符，然后按【Return】键。

（2）SQL*PLUS 显示一个右尖括号（>）作为提示符。

（3）在结束一条 SQL*PLUS 命令时，不需要用分号“;”，只要按【Return】键，但如果键入一个分号“;”，SQL*PLUS 也不会报错。

使用 SQL*PLUS 的 SET 命令可设置系统变量，这些变量的设置将影响 SQL*PLUS 命令的运行，比如输出的缺省列的宽度，是否显示由命令选择的记录数和页的大小等。可用 SHOW 命令列出使用 SET 命令设置的当前变量。

前面讲过，利用 SQL 命令的 INSERT、DELETE 和 UPDATE 可以对存储在数据库中的数据进行修改，但 SQL 命令不能使修改永久化，只有输入 COMMIT 命令或者一个数据定义语句，才能将修改永久提交到数据库。利用 SQL*PLUS 的 AUTOCOMMIT 变量控制自动提交的特性，它有如下两种形式：

- SET AUTOCOMMIT ON　打开自动提交。
- SET AUTOCOMMIT OFF　关闭自动提交（缺省方式）。

在自动提交方式下，每一个 SQL 命令或 PL/SQL 块所指定的数据库修改，SQL*PLUS 将自动提交，在自动提交后，SQL*PLUS 显示下列信息：

需要全本请访问 www.ertongbook.com