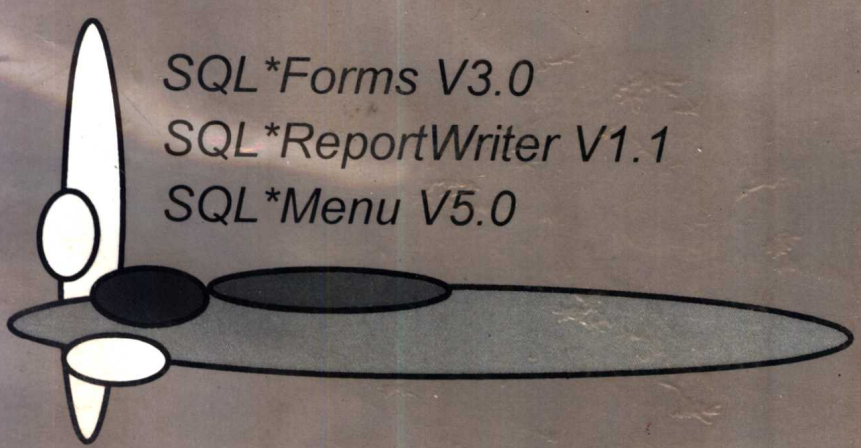
ORACLE 7 应用开发工具系列教程之二

ORACLE SQL
表格工具 报表工具
菜单工具

刘金亭 苏红宇 编



電子工業出版社

ORACLE SQL
表格工具、报表工具、菜单工具

刘金亭　苏红宇　编

電子工業出版社

内容简介

1992 年 Oracle 公司和清华大学联合创办了 Oracle—清华联合大学，旨在培训更多的数据库人员，促进和提高数据库应用及管理领域的水平。ORACLE7 应用开发工具系列教程是为配合 Oracle 大学而编写的。考虑到读者多为 Oracle 用户，在编写过程中既注重实践环节和易读易学，又注重内容的系统性，对每个应用开发工具和产品都作了比较全面的介绍。因此本系列教程不仅是 Oracle 大学的培训教材，也是数据库系统开发人员和最终用户的参考资料。

《表格工具、报表工具、菜单工具》全面地介绍了 ORACLE7 支持的三个 SQL 系列工具：SQL * Forms 3.0、SQL * ReportWriter 1.1 和 SQL * Menu 5.0。全书共分三篇。第一篇表格工具 SQL * Forms 3.0 主要介绍表格工具及表格应用的基本概念，表格的设计和定义，触发器和封装过程，用户出口、文档、调试、集成、运行界面和实用程序等。第二篇报表工具 SQL * ReportWriter 1.1 主要介绍报表工具及表格的设计和定义、运行界面和实用程序等。第三篇菜单工具 SQL * Menu 5.0 主要介绍菜单工具及菜单应用的基本概念、运行界面、菜单的设计和定义、宏命令、封装过程、菜单管理和实用程序等。

ORACLE SQL 表格工具、报表工具、菜单工具
刘金亭　苏红宇　编
责任编辑：宋玉升
*
电子工业出版社出版
北京市海淀区万寿路 173 信箱(100036)
电子工业出版社发行　各地新华书店经销
顺义县天竺颖华印刷厂印刷

开本：787×1092 毫米 1/16　印张：15.5 字数：360 千字
1995 年 11 月第一版　1995 年 11 月北京第一次印刷
印数：3000 册　定价：30 元
ISBN7-5053-3324-0/TP・1261

前　言

作为技术支持和服务的延伸，在1992年Oracle公司和清华大学联合创办了Oracle－清华联合大学，旨在为我国培训更多的数据库人员，促进和提高我国在数据库应用及管理领域的水平。Oracle公司为联合大学提供有最新版本的ORACLE7分布式关系数据库管理系统、应用开发工具(SQL工具系列和CDE协同开发环境等)以及系统的技术资料，清华大学为联合大学提供有经验丰富的师资和良好的上机实习环境。联合大学结合ORACLE系统进行教学，理论与实践相结合，使学员即能学到Oracle公司所提供的最新产品又能在分布式环境下上机操作。目前，Oracle－清华联合大学平均每月招收1－2个班学员。

本系列教程是为配合Oracle大学教学而编写的，具有较强的系统性，内容包括ORACLE7所支持的最新应用开发工具(如SQL＊Plus，SQL＊Forms，SQL＊ReportWriter，SQL＊Menu，CDE等)和相关产品(PL/SQL，PRO＊C，OCI等)。考虑到读者多为Oracle用户，在编写过程中既注重实践环节和易读易学，又注重内容的系统性，对每个应用开发工具和产品都作了比较全面介绍。因此本系列教程不仅是Oracle大学的培训教材，也是数据库系统开发人员和最终用户的参考资料。

《ORACLE7应用开发工具系列教程之二：表格工具、报表工具、菜单工具》全面地介绍了ORACLE7支持的三个SQL系列工具：SQL＊Forms 3.0、SQL＊ReportWriter 1.1和SQL＊Menu 5.0。全书共分三篇。第一篇表格工具SQL＊Forms 3.0，主要介绍表格工具及表格应用的基本概念、表格的设计和定义、触发器、过程和封装过程、用户出口、文档、调试、集成、运行界面和实用程序等。第二篇报表工具SQL＊ReportWriter 1.1，主要介绍报表工具及报表应用的基本概念、表格的设计和定义、运行界面和实用程序等。第三篇菜单工具SQL＊Menu 5.0，主要介绍菜单工具及菜单应用的基本概念、菜单应用的运行界面、菜单的设计和定义、宏命令、封装过程、菜单管理和实用程序等。

本书第一篇、第二篇和附录A、附录B由刘金亭编写，第三篇和附录C由苏红宇编写，全书由刘金亭统稿。

本书编写过程中得到了ORACLE－中国公司和清华大学计算中心的大力支持，同时得到了中国软件行业协会和电子工业出版社的支持和帮助，吴克忠教授、胡昆山高工和宋玉升编审审阅了全稿，并提出许多宝贵意见，在此表示感谢。冯和沁同志为本书的编写提供了许多帮助，并协助完成部分录入工作，在此表示衷心感谢。由于时间仓促，书中错误难免，请读者指正。

编者

1995年8月

目　录

第一篇　SQL * Forms3.0

第一章　SQL * Forms 概述 …… (1)
1.1　SQL * Forms 的基本概念 …… (1)
1.1.1　什么是 SQL * Forms …… (1)
1.1.2　SQL * Forms 的功能 …… (2)
1.1.3　SQL * Forms 的特点 …… (2)
1.2　SQL * Forms 的构成 …… (2)
1.3　FORM 应用的对象 …… (3)
1.4　FORM 应用的开发过程 …… (4)
1.5　功能键 …… (5)
参考资料 …… (6)
第二章　SQL * Forms 设计界面 …… (7)
2.1　交互设计界面 SQLFORMS 的启动、构成与退出 …… (7)
2.2　FORM 应用的定义过程 …… (10)
2.3　FORM 应用存储 …… (11)
2.4　设计界面的功能键 …… (12)
2.5　生成缺省应用 …… (13)
2.5.1　背景条件 …… (13)
2.5.2　应用设计及主从关系 …… (14)
2.5.3　生成缺省块和自动主从关系 …… (15)
2.5.4　试运行缺省应用 …… (18)
2.5.5　修改的必要性 …… (19)
2.6　屏幕画笔 …… (20)
第三章　SQL * Forms 对象的定义 …… (23)
3.1　表定义 …… (23)
3.2　块定义 …… (24)
3.3　域定义 …… (26)
3.4　值表 …… (30)
3.5　页定义 …… (31)
第四章　触发器 …… (35)
4.1　触发器的概念 …… (35)
4.2　触发器的定义 …… (35)
4.3　触发器的设计过程 …… (37)
4.4　触发器的作用域 …… (37)
4.5　事件处理和触发器处理 …… (37)

试读结束：需要全本请在线购买：　www.ertongbook.com

4.6 触发器的类型 …… (38)
4.6.1 键触发器 …… (39)
4.6.2 导航触发器 …… (42)
4.6.3 合法性检查触发器 …… (43)
4.6.4 事务处理触发器 …… (44)
4.6.5 查询触发器 …… (45)
4.6.6 信息处理触发器 …… (46)
第五章 封装函数,封装过程和 Form 级过程 …… (47)
5.1 触发器语言 PL/SQL …… (47)
5.2 例外处理 …… (48)
5.3 封装过程和封装函数 …… (49)
5.3.1 封装过程 …… (49)
5.3.2 封装函数 …… (54)
5.4 Form 级过程 …… (55)
5.5 全局变量和系统变量 …… (57)
5.5.1 全局变量 …… (57)
5.5.2 系统变量 …… (57)
5.6 触发器和 Form 级过程实例 …… (58)
5.6.1 实例背景 …… (59)
5.6.2 实例的 5 个 Form 级过程 …… (59)
5.6.3 实例的 21 个触发器 …… (61)
第六章 用户出口 …… (67)
6.1 用户出口和用户出口类型 …… (67)
6.2 用户出口的编写 …… (67)
6.3 安装和执行用户出口 …… (68)
第七章 文档、调试和集成 …… (69)
7.1 文档 …… (69)
7.2 调试 …… (69)
7.3 SQL * Forms 与其它工具的集成 …… (71)
第八章 SQL * Forms 运行界面 …… (73)
8.1 Form 应用的最终用户 …… (73)
8.2 SQL * Forms 运行界面的启动与退出 …… (73)
8.3 运行功能键 …… (74)
8.4 Form 应用的工作方式和运行模式 …… (76)
8.5 运行界面下的一般操作 …… (76)
8.5.1 移动光标 …… (77)
8.5.2 域编辑 …… (77)
8.6 查询 …… (77)
8.6.1 查询所有记录 …… (77)
8.6.2 查询指定记录 …… (77)
8.6.3 用匹配模式查询 …… (78)
8.6.4 用可变条件查询 …… (78)

8.6.5 复杂查询 …… (79)
8.6.6 统计查询记录 …… (79)
8.7 修改 …… (79)
8.7.1 修改记录 …… (80)
8.7.2 删除记录 …… (80)
8.7.3 生成新记录 …… (80)
第九章 SQL * Forms 构成 …… (82)
9.1 CONVFORM …… (82)
9.2 SQLFORMS …… (83)
9.3 GENERATE …… (83)
9.4 RUNFORM …… (84)

第二篇 表格工具 SQL * ReportWriter 1.1

第一章 SQL * ReportWriter 概述 …… (86)
1.1 SQL * ReportWriter 的基本概念 …… (86)
1.1.1 什么是 SQL * ReportWriter …… (87)
1.1.2 SQL * ReportWriter 的功能 …… (87)
1.1.3 SQL * ReportWriter 的特点 …… (88)
1.2 SQL * ReportWriter 的构成 …… (88)
1.3 SQL * ReportWriter 的对象 …… (89)
1.4 报表的开发过程 …… (89)
1.5 功能键 …… (90)
参考资料 …… (93)
第二章 SQL * ReportWriter 交互式设计界面 …… (94)
2.1 交互式设计界面的启动、构成与退出 …… (94)
2.2 报表的定义过程 …… (96)
2.3 报表的存储和执行过程 …… (97)
2.4 生成缺省报表应用 …… (99)
第三章 SQL * ReportWriter 对象的定义 …… (101)
3.1 报表定义 …… (101)
3.2 查询定义 …… (102)
3.3 域定义 …… (104)
3.3.1 域属性 …… (104)
3.3.2 域维护 …… (107)
3.3.3 域维护实例 …… (108)
3.4 统计定义 …… (109)
3.4.1 统计属性 …… (110)
3.4.2 统计维护 …… (112)
3.4.3 统计维护实例 …… (113)
3.5 组定义 …… (114)
3.5.1 组属性 …… (114)
3.5.2 组维护 …… (118)

3.5.3 组维护实例 …… (119)
3.6 文本定义 …… (123)
3.6.1 文本对象的概念及属性 …… (123)
3.6.2 文本维护 …… (128)
3.6.3 文本实例 …… (128)
3.7 参数定义 …… (131)
3.7.1 参数类型和使用 …… (131)
3.7.2 参数属性 …… (134)
3.7.3 参数管理 …… (135)
3.8 用户出口 …… (135)
3.8.1 用户出口的类型 …… (135)
3.8.2 封装用户出口 …… (136)
第四章 SQL * ReportWriter 实例 …… (138)
4.1 实例的背景 …… (138)
4.2 实例的缺省定义 …… (139)
4.3 维护 …… (141)
4.4 试运行 …… (142)
第五章 SQL * ReportWriter 构成 …… (144)
5.1 SQLREP …… (144)
5.2 GENREP …… (146)
5.3 RUNREP …… (146)
5.4 DUMPREP …… (148)
5.5 LOADREP …… (148)
5.6 PRINTDEF …… (149)
5.7 TERMDEF …… (149)
5.8 MOVEREP …… (149)

第三篇 菜单工具 SQL * Menu 5.0

第一章 SQL * Menu 概述 …… (150)
1.1 SQL * Menu 基本概念 …… (150)
1.1.1 什么是 SQL * Menu …… (150)
1.1.2 SQL * Menu 的优点 …… (150)
1.1.3 SQL * Menu 的特点 …… (151)
1.2 SQL * Menu 的构成 …… (152)
1.3 SQL * Menu 的动态菜单 …… (152)
1.3.1 动态菜单方法 …… (152)
1.3.2 动态菜单的形式 …… (153)
1.4 SQL * Menu 的使用权限 …… (153)
1.5 SQL * Menu 的功能键 …… (154)
参考资料 …… (155)
第二章 SQL * Menu 运行界面操作指南 …… (156)
2.1 登录和退出 SQL * Menu 运行界面 …… (156)

2.1.1 登录 SQL * Menu 运行界面 …… (156)
2.1.2 退出 SQL * Menu 运行界面 …… (157)
2.2 菜单类型和导航 …… (157)
2.2.1 在 SQL * Menu 运行界面中导航 …… (157)
2.2.2 菜单的显示类型 …… (158)
2.3 运行界面功能键 …… (164)
2.4 获得帮助 …… (166)
2.4.1 信息行 …… (166)
2.4.2 功能键帮助 …… (167)
2.4.3 应用帮助系统 …… (167)
2.5 使用背景菜单 …… (167)
2.6 使用操作系统命令 …… (168)
第三章 SQL * Menu 设计基础 …… (170)
3.1 SQL * Menu 的对象 …… (170)
3.2 SQL * Menu 设计的基本步骤 …… (171)
3.3 规划 SQL * Menu 菜单应用 …… (171)
3.3.1 规划菜单树 …… (172)
3.3.2 规划角色 …… (172)
3.4 SQL * Menu 设计界面数据输入元素 …… (173)
3.4.1 警告窗 …… (173)
3.4.2 按钮或选择按钮 …… (174)
3.4.3 确认框 …… (175)
3.4.4 对话盒 …… (175)
3.4.5 域 …… (175)
3.4.6 表和展开表 …… (175)
3.4.7 值表 …… (176)
3.4.8 滚动区 …… (177)
第四章 SQL * Menu 设计指南 …… (180)
4.1 登录和退出 SQL * Menu 设计界面 …… (180)
4.1.1 登录 SQL * Menu 设计界面 …… (180)
4.1.2 退出 SQL * Menu 设计界面 …… (181)
4.2 获得帮助信息 …… (182)
4.3 设计界面功能键 …… (184)
4.4 基本设计操作 …… (185)
4.4.1 定义应用 …… (185)
4.4.2 定义菜单 …… (188)
4.4.3 定义菜单项 …… (189)
4.4.4 定义替换参数 …… (194)
4.4.5 建立 PL/SQL 过程 …… (196)
4.4.6 复制或引用对象 …… (199)
4.5 保存应用 …… (199)
4.6 生成应用 …… (200)
4.7 执行应用 …… (200)

4.8 打印应用信息 …… (201)
4.9 设置选项 …… (203)
4.9.1 选项内容 …… (203)
4.9.2 设置选项的方法 …… (204)
第五章 SQL * Menu 宏命令 …… (205)
5.1 宏语句的命令格式 …… (205)
5.2 宏、封装过程和功能键 …… (205)
5.3 宏命令介绍 …… (206)
5.4 宏命令失败 …… (207)
第六章 SQL * Menu 封装过程 …… (208)
第七章 SQL * Menu 管理 …… (213)
7.1 SQL * Menu 应用维护操作 …… (213)
7.1.1 复制一个应用 …… (213)
7.1.2 重新命名一个应用 …… (214)
7.1.3 删除一个应用 …… (215)
7.1.4 卸载一个应用 …… (215)
7.2 安全操作 …… (216)
7.2.1 授予存取 SQL * Menu 的权限 …… (217)
7.2.2 回收存取 SQL * Menu 的权限 …… (217)
7.2.3 建立角色 …… (217)
7.2.4 给角色分配用户 …… (220)
第八章 SQL * Menu 构成 …… (222)
8.1 SQL * Menu 设计界面 …… (222)
8.2 SQL * Menu 运行界面 …… (222)
8.3 生成运行文件应用程序 …… (223)
8.4 文档打印应用程序 …… (224)
附录 A. SQL * Forms 功能键 …… (225)
附录 B. SQL * ReportWriter 功能键 …… (231)
附录 C. SQL * Menu 功能键 …… (235)

第一篇　表格工具 SQL * Forms 3.0

第一章　SQL * Forms 概述

1.1　SQL * Forms 的基本概念

1.1.1　什么是 SQL * Forms

SQL * Forms 是 ORACLE 关系数据库系统提供的一种用于开发和运行基于表格(Form)应用的第四代语言(4GL)工具软件。用 SQL * Forms 开发表格只须简单地使用菜单、定义表、展开表、屏幕画笔等来说明 Form 应用的需求，而后 SQL * Forms 即把所定义的应用与 ORACLE 数据字典结合生成用户应用。该工具的使用对象是应用程序开发人员和最终用户，要求工具的使用者有 ORACLE 数据库的基础知识和 SQL 语言知识。

由 SQL * Forms 开发的基于表格的交互式应用，通常称之为一个表格(Form)应用，简称为一个 Form。所谓基于表格，是指这种应用类同于人们日常办公业务中用表格对信息管理的办法，通过查表，填表，修改表，插入表，删除表等动作实现信息管理。

比如人事部门对职工的管理，在日常管理中我们可以为每一名职员设一张职员基本信息表。增加一名新职员，填写一张新表；如果要查看某一职员的情况，须在一摞职员表中依职工号码或姓名等查出该职员的基本信息表；一名职员的情况有变动时，比如学历变动，须改动该职员的基本信息表；一名职员调离时，则须撤销相应表格。这些操作分别对应于表格的插入、查询、修改和删除等。

Form 应用就是为了在计算机上类似地实现这种基于表格的信息处理过程，使办公人员以非常接近于日常办公模式的方法在计算机系统中实现对信息的处理。

所谓交互式，是指用户使用 Form 应用对数据库信息处理过程是交互的。综上所述，Form 应用可以定义为：

- 一个由 SQL * Forms 工具开发的基于表格的交互式应用。
- 使用在屏幕上呈现出类似于表格式的填充表，用户可录入，查询，修改和删除数据库中的信息。

关系数据库只是在逻辑上以表格的形式组织和存放数据。在数据库设计过程中，为了减少冗余和确保一致性，要对信息规范化。这样一来，数据库中的基表与最终用户心目中的表格差异很大，而用户对信息的所有操作，最终都是要对基表的操作。对基表的直接操作一般使用 SQL 语言(SQL * Plus)，这不仅要求用户了解 SQL 语言，而且对基表的组织应当了解。对最终用户来说，这种要求是过分的，也就是说，我们不能要求一般的办公人员一定要了解数据库基本结构和 SQL 语言。

Form 应用则是一个模拟日常表格处理的友好界面，最终用户可以通过表格形式来对

试读结束：需要全本请在线购买：　www.ertongbook.com

数据进行处理。Form 应用不仅界面友好，而且可以提高工作效率，减少差错率。SQL * Forms 正是为应用开发人员和最终用户提供的一个开发和运行 Form 应用的工具。

1.1.2 SQL * Forms 的功能

SQL * Forms 具有如下功能：

· 用于定义表 应用；

· 使用 Form 应用可以用类似于日常表格操作的方法实现对数据库的各种基本操作：插入，查询，修改和删除等；

· 进行合法性检查，保证输入的正确性，减少错误；

· 提高开发效率和用户的工作效率。包括为开发人员和用户提供提示信息，提供缺省值，自动完成计算和统计等。

1.1.3 SQL * Forms 的特点

SQL * Forms 具有如下特点

· 具有第四代开发工具的特点。在用 SQL * Forms 开发 Form 应用过程中，是采用交互填表方式，基本上不用编程，且提供有大量缺省操作

· 用 SQL * Forms 可以方便地生成一个 Form 应用的原型，更复杂的应用是通过在型基础上不断修改而得到。

· 为用户提供有编程接口触发器（ Trigger ）和用户出口（ User exit ），利用触发器和用户出口，开发人员可以定义比较复杂的应用。

· 不取代报表工具软件（ 对打印及 I/O 支持不够 ）和 SQL * Plus（ SQL * Plus 是面向 DBA 的，而 SQL * Forms 是面向应用开发人员和最终用户的 ）。

从以上特点可以看出，利用 SQL * Forms 我们可以方便地定义出难度在中等及中等以上的基于表格的应用。但对于更复杂的应用，由于触发器变得更复杂，需要更多的用户出口，使编程工作量大大增加，已难显示出第四代语言（4GL）工具的特点，在这种情况下应考虑选用其它工具。这也正是 4GL 工具专用性的一个特点，即每种工具都有它的适用范围。

1.2 SQL * Forms 的构成

SQL * Forms 作为 ORACLE 公司提供的一个产品，是由一组基表和一组实用程序组成的。基表用以存储用 SQL * Forms 所开发的 Form 应用的有关信息等，由于本书是面向应用程序开发人员和最终用户的，所以不去介绍基表的构成。

SQL * Forms 的实用程序有如下四个：

（1）CONVFORM（SQL * Forms－Convert）：Form 转换实用程序，用以在 Form 应用定义的不同存储格式间进行转换，可完成如下功能：

· 从基表存储的定义生成操作系统的文本文件。

· 把设计界面或文本文件中的应用定义插入到数据库基表中。

· 从低版本 SQL * Forms 的应用定义文本生成 V3.0 的文本文件。

（2）SQLFORMS（SQL * Forms －Design）：Form 应用交互式设计界面，用以定义表 s 对象、保存应用定义，生成运行文件等。

（3）GENERATE（SQL * Forms －Generate）：Form 运行文件生成实用程序，用以从

应用定义文本文件或存储应用定义的基表生成运行文件。

(4) RUNFORM (SQL * Forms —Run Form)：Form 的 Runtime 界面，用以运行已生成运行文件的 Form 应用。

作为一个应用开发人员，经常使用的是设计界面 SQLFORMS 。在设计界面中可以调用其它三个实用程序。

1.3 Form 应用的对象

Form 应用是由 SQL×Forms 对象(有时简称 Form 对象) 构成的。这些对象包含有操作和生成一个 Form 应用的所有信息。

一个 Form 应用的主要对象是 Form (表格)，它包含有 Form 自身及其所拥有的其它对象。Form 对象能与其它 Form 应用、Form 应用组、菜单、报表以及其它应用相连接构成复杂应用。

一个 Form 应用由如下对象构成：

· Form(表格)：表格应用，定义表格级属性。

· Block (块)：是一组域的集合，是 Form 的逻辑构成单位，通常与数据库的一个基表相对应。

· Field (域)：在屏幕上显示一个值的显示区域及相关属性，如显示格、合法性检查、缺省值设置、提示信息、值表等。

· Trigger (触发器)：由 Form 中事件(Event) 触发的处理命令或程序段。

· Form—Level Procedure (Form 级过程)：可带参数的过程。在 Form 内可被 Trigger 或其它 Form 级过程调用，一般在不引起混淆的情况下简称为过程。

· Page (页)：显示信息的集合。一页上可以显示多个块；一个块也可以占用多个页

从 Form 应用运行的角度看，Form 应用还包括如下对象：

· Record (记录)：Form 应用所显示的从基表中提取的一行。一般记录与基表的行相对应。

· Message Line (信息行)：屏幕上的倒数第 2 行，用以显示帮助信息、系统信息和错误信息等。

· Status Line (状态行)：屏幕上的最后一行，用以显示系统状态

SQL * Forms 的对象层次结构如图 1.1 所示。

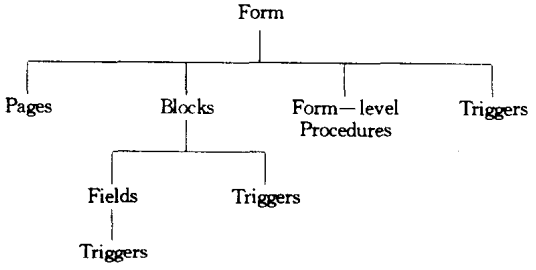


图1.1 SQL * Forms对象的层次结构

图 1.2 是 SQL * Forms 对象示意图

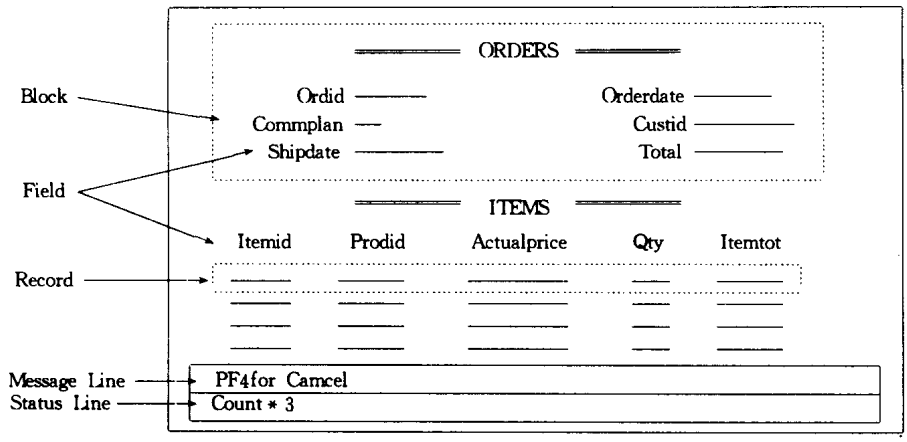


图1.2 SQL * Forms对象示意图

1.4 Form 应用的开发过程

第一步是分析用户需求。这里所说的需求，是假定系统分析员（SA）已完成需求分析，且已完成数据库设计，根据用户要求和对应的数据库用户模式（外模式），详细分析用户需要处理哪些数据、数据的组织格式及关系、处理规则、操作流程等。只有详细了解用户需要，优化用户需要，并得到用户认可，才能开发出满意的应用。这一步需要应用开发人员和用户共同完成。

第二步要把用户的需要转化为 SQL * Forms 的对象。这一步应由开发人员独立完成，也是要求比较高的一步，要求开发人员具有从用户需求中识别出所有 SQL * Forms 对象的能力。比如用户给出的是一张订货单，只要求是对订货单进行插入、查询、修改和删除，看来似乎不复杂，但一张订货单的信息可能因包含有订单基本信息，货品细目，客户基本信息等而对应于数据库的几张不同基表或视图，这样一来，对应成 SQL * Forms 对象就显得很复杂。这样的 Form 应用可能要有多个块，块之间要有主从关系（Master—Detail），协调主从关系要有一组触发器（Trigger）等。通过后面章节的例子可以看出，即使是一个训练有素的开发人员，要想一下子识别出一个用户需求对应的所有 SQL * Forms 对象（比如触发器）不是一件容易的事。好在 SQL * Forms 设计界面有许多缺省功能，开发人员可以利用缺省功能辅助生成许多对象，可以省去识别上的困难，这一点在后面的例题中读者会感觉到。

第三步是应用定义。在识别出一个用户应用（Form）的 SQL * Forms 对象后，接下来是利用 SQL * Forms 设计界面 SQLFORMS 定义该 Form 应用，即对对象的层次构成从顶向下逐层定义各对象。一般可分为四步：

- 利用缺省功能生成 Form 应用的基本框架。
- 修改缺省生成应用的对象，使之满足用户要求。
- 如果需要的话，增加触发器、过程或用户出口，从而进一步完善应用。存储定义到数据库基表并生成运行文件。详细定义过程将在第 2 章中介绍。

第四步是运行已定义好的 Form 应用。利用该应用实现对数据库的操作。

综上所述，Form 应用的开发过程如图 1.3 所示。

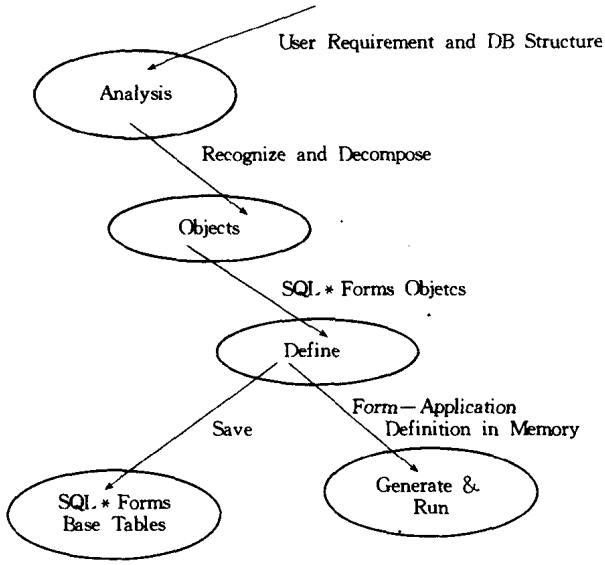


图1.3 Form应用开发过程

1.5 功能键

为了覆盖更多的软件和硬件平台，同其它 ORACLE 工具一样，SQL * Forms 对功能键也采用逻辑键定义与键位映象相结合的方法。对于任何一个功能键，均由两部分构成：

(1) 该功能键的功能。

(2) 该功能键在具体运行平台上的物理键位。前者我们称为逻辑键，后者称为键盘映象 (Map)。比如功能键 [Accept]，逻辑定义如下：

[Accept]：

- 在对话盒状态，关闭对话盒，且接纳你的输入。
- 在定义表(Define Forms) 或展开表(Define Spread Tables)状态，关闭定义表 或展开表，且接纳所有用户的修改。
- 在屏幕画笔状态接纳所有的修改，返回屏幕画笔之前的状态。

ORACLE 的所有资料只介绍逻辑键，这样就回避了具体设备的不同给我们带来的不便，因为功能键的逻辑定义不依于环境。功能键的键位映象在不同的环境中可以不同，且 DBA(数据库管理员) 可以修改键位的定义。比如 [Accept] 键可以定义为物理键 F2，也可定义为其它键。

为了方便用户，ORACLE 工具都配有 [Show Key] 功能键，该功能键用以在联机状态下显示功能键键位映象表 (Keyboard Map)。在不同环境中，[Show Key] 功能键一般对应如下物理键之一：

- CTRL+K
- ESC+K
- F1
- F8

SQL * Forms 的功能键分为两组；一组是用于设计界面的功能键，称之为设计功能键；

另一组是用于运行界面的功能键，称之为运行功能键，我们将分别在后面相应章节中进一步介绍，并在附录 A 中给出全部功能键的逻辑定义。

参考资料

(1) SQL * Forms Designer's Quick Reference Version 3.0, Part No. 3708—V3.0

(2) SQL * Forms Designer's Reference Version 3.0, Part No. 3304—V3.0

(3) SQL * Forms Designer's Tutorial Version 3.0, Part No. 3302—V3.0

(4) SQL * Forms Operator's Guide Version 3.0, Part No. 3301—V3.0

(5) SQL * Fprms Operator's Quick Reference Version 3.0, Part No. 3704—V3.0

(6) SQL Language Reference Version 6.0, Part No. 778—V6.0

(7) PL/SQL User's Guide and Reference Version 1.0, Part No. 800—V1.0

第二章　SQL * Forms 设计界面

2.1　交互式设计界面 SQLFORMS 的启动，构成与退出

交互式设计界面，即 SQLFORMS，是一个用于定义表格式应用的交互式开发工具。该界面不仅提供定义 SQL * Forms 各对象的简捷手段，而且还集成其它 SQL * Forms 实用程序来完成 Form 的生成，运行，转换等工作。也就是说，利用该界面，开发人员几乎可以完成所有与 Form 应用 开发和运行有关的工作。本教程将以 SQLFORMS 为主介绍 Form 应用的设计与开发。

交互式设计界面的启动命令如下：

SQLFORMS [选项参数] [用户名/密码]

其中可选项参数将在第 9 章详细 介绍，[用户名/密码]包括用户标识名，密码 (Password) 及数据库路径名。例如下面是一个启动指定网络地址 Server 上 ORAD7 数据库的 SQL-FORMS 设计界面的例子：

SQLFORMS scott/tiger @t:166.111.8.7:orad7

其中 Scott 为用户名，tiger 为 Password，166.118.7 为地址，orad7 为数据库名。

如果省略 [用户名/密码] 或该选项填写不正确，则 SQLFORM 给出一个注册屏幕如图 1.4 所示。

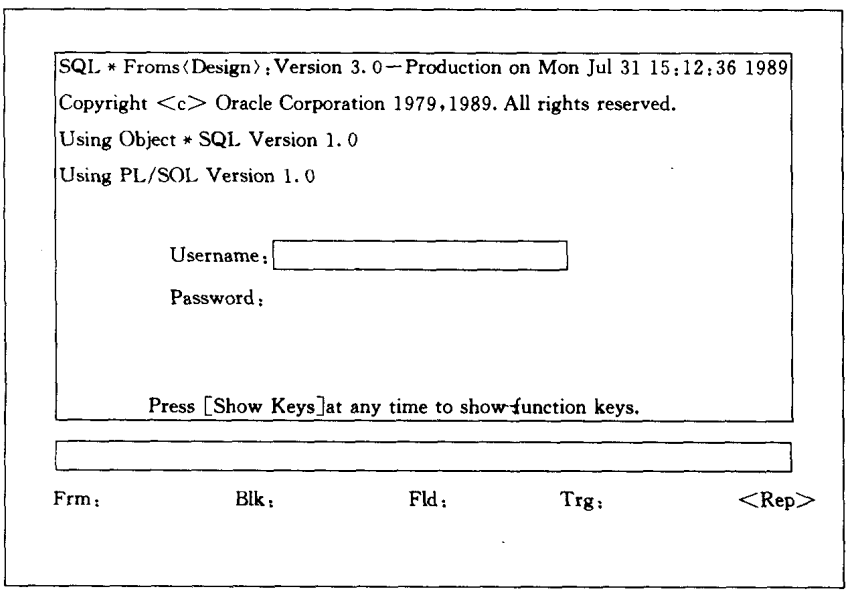


图 1.4　SQLFORMS 注册屏幕

这时在 'Username:' 后应填写正确的用户名和相应的数据库名，而后按 [Next Field] 键（可用 [Show Keys] 键查看键位映象 Map）。在 'Password:' 后填写密码而后按 [Next Field]。如填写正确则注册成功，屏幕上出现如图 1.5 所示主菜单；如果填写不成功，则在信息行显示一条错误信息，用户须重新注册。如果三次注册全失败，则 SQLFORMS 自动退出，返回操作系统。

试读结束：需要全本请在线购买：　www.ertongbook.com