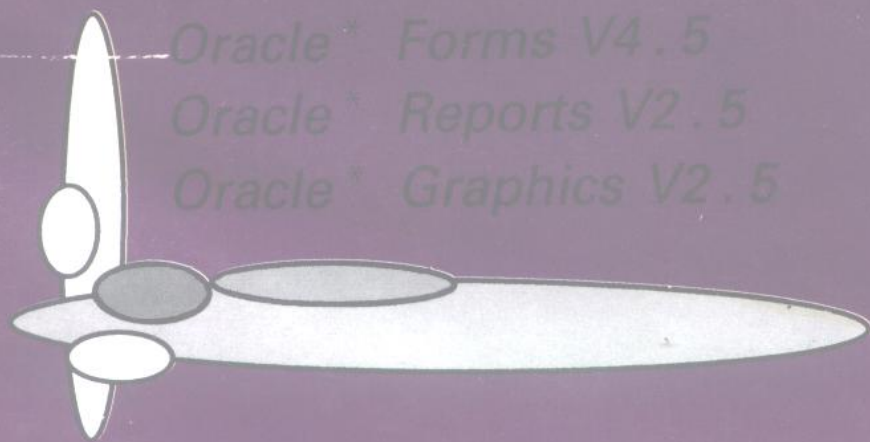


ORACLE7 应用开发工具系列教程之五

CDE 协同开发环境 Oracle Developer/2000

刘金亭 朱 莉 蔡 蔚 编



電子工業出版社

CDE 协同开发环境

Oracle Developer/2000

刘金亭 朱 莉 蔡 蔚 编

電子工業出版社

内 容 提 要

本书是 Oracle-清华联合大学编写的 Oracle 应用开发工具系列教材之一,内容包括 Oracle 7 支持的三个开发工具:Oracle *Forms 4.5,Oracle *Reports 2.5 和 Graphics 2.5。本教程不仅是培训教材,也是数据库系统开发人员和广大用户的参考资料。

CDE 协同开发环境

Oracle Developer/2000

刘金亭 朱莉 蔡蔚 编

责任编辑 宋玉升

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

中国农业出版社印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:18 字数:420 千字

1996 年 8 月第一版 1996 年 8 月第一次印刷

印数:1—4000 册 定价:24.00 元

ISBN 7-5053-3728-9/TP·1581

前 言

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)以及技术资料,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 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。

本教程协同开发环境(CDE2)Oracle Developer/2000 全面地介绍了 ORACLE7 支持的 Window 界面下协同开发环境(CDE2)Oracle Developer/2000 的三个开发工具:Oracle * Forms 4.5、Oracle * Reports2.5 和 Oracle * Graphics2.5。全书共分四篇,第一篇 Oracle * Forms 4.5,主要介绍 FORM(表格)工具及 FORM 应用的基本概念、Oracle * Forms 的设计界面、单一 Form 模块的定义、触发器、多 Form 模块的集成、菜单模块、库模块、运行界面操作等。第二篇 Oracle * Reports 2.5,主要介绍 Report(报表)工具及 Report 应用的基本概念、Oracle * Reports 的设计界面、Report 应用的设计和定义、触发器、矩阵报表的定义等。第三篇 Oracle * Graphics 2.5,主要介绍 Graphic(图形)工具及图形应用的基本概念、Oracle * Graphics 的设计界面、Graphic 应用的设计和定义、触发器、应用实例等。第四篇 Oracle Developer/2000 工具的集成,主要介绍 Oracle Developer/2000 工具集成的概念、简单集成方法、使用记录组集成等。

本书第一篇和附录 A 由刘金亭编写,第二篇由朱莉编写,第三、四篇由蔡蔚编写,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,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。由于时间仓促,书中错误难免,敬请读者指正。

编者

1996 年 6 月于北京清华大学

目 录

第一篇 Oracle * Forms 4.5

第一章 Oracle * Forms 概述	(1)
§ 1.1 Oracle * Forms 的基本概念	(1)
§ 1.2 Form 应用的功能和构成	(1)
§ 1.3 Form 应用的对象	(2)
§ 1.4 Form 应用的开发过程	(3)
§ 1.5 Oracle * Forms 的产品构成和学习过程	(4)
参考资料	(5)
第二章 Oracle * Forms 的设计界面	(6)
§ 2.1 交互式设计界面的启动、构成、数据库连接和退出	(6)
§ 2.2 模块管理和应用存储	(8)
2.2.1 创建和打开模块	(8)
2.2.2 模块的存储格式	(8)
2.2.3 模块的其它操作	(9)
§ 2.3 模块的定义过程	(10)
§ 2.4 模块定义环境及基本操作	(10)
2.4.1 对象导航器	(10)
2.4.2 布局编辑器	(16)
2.4.3 属性窗	(20)
2.4.4 弹出菜单和表/列浏览器	(21)
第三章 定义一个单一 Form 应用	(22)
§ 3.1 Form(表格)对象	(22)
3.1.1 创建 Form 对象	(22)
3.1.2 Form 对象的属性	(22)
3.1.3 Form 对象实例	(23)
§ 3.2 Block(块)对象	(24)
3.2.1 创建 Block 对象	(24)
3.2.2 Block 对象的属性	(27)
3.2.3 创建块实例	(29)
3.2.4 块间联系(Relation)对象	(30)
§ 3.3 Text Item(文本项)对象	(31)
3.3.1 创建 Text Item 对象	(31)
3.3.2 Text Item 对象属性	(32)
3.3.3 Item 对象属性设置实例	(35)
3.3.4 LOV(值表)对象	(35)

3.3.5	Editor(编辑器)对象	(38)
§ 3.4	其它可输入项	(39)
3.4.1	Check Box(检测盒)	(40)
3.4.2	List Item(列表项)	(40)
3.4.3	Radio Group(无线组)	(42)
3.4.4	VBX Control(VBX 控件)	(42)
§ 3.5	非输入项	(42)
3.5.1	Display Item(显示项)	(43)
3.5.2	Image Item(图象项)	(43)
3.5.3	Button(按钮项)	(44)
§ 3.6	Window(窗口)和 Canvas(画布)	(45)
3.6.1	Window 对象	(46)
3.6.2	Canvas 对象	(47)
§ 3.7	其它对象	(49)
3.7.1	Alert	(49)
3.7.2	Object Group	(49)
3.7.3	Parameter	(50)
3.7.4	Property Class	(50)
3.7.5	Record Group	(50)
3.7.6	Visual Attribute	(51)
第四章	触发器	(52)
§ 4.1	触发器概述	(52)
4.1.1	触发器的概念	(52)
4.1.2	触发器特征	(53)
4.1.3	创建触发器及触发器属性	(56)
§ 4.2	触发器编程	(58)
4.2.1	在触发器中引用 Form 变量	(59)
4.2.2	封装过程和封装函数	(60)
4.2.3	用户命名子程序	(64)
§ 4.3	触发器实例	(64)
4.3.1	合法性检查	(64)
4.3.2	给其它输入项增加功能	(65)
4.3.3	给非输入项增加功能	(66)
4.3.4	导航触发器	(68)
4.3.5	事务处理触发器	(69)
4.3.6	查询触发器	(70)
4.3.7	鼠标触发器	(71)
4.3.8	显示信息处理及警示	(71)
第五章	定义有多个 Form 的表格应用	(73)
§ 5.1	Form 集成的概念	(73)
§ 5.2	Form 间的连接	(73)
§ 5.3	启动一个新 Form	(74)
5.3.1	OPEN_FORM	(74)

5.3.2	CALL_FORM	(74)
5.3.3	NEW_FORM	(75)
第六章	其它模块和 Runform 命令	(76)
§ 6.1	定义菜单模块	(76)
6.1.1	菜单的概念和结构	(76)
6.1.2	定义菜单模块操作	(76)
6.1.3	将菜单模块连到 Form 模块	(80)
§ 6.2	定义库模块	(81)
§ 6.3	Runform 命令	(81)
第七章	表格应用的运行	(83)
§ 7.1	FORM 应用的最终用户	(83)
§ 7.2	FORM 应用的启动、界面构成与退出	(83)
7.2.1	FORM 应用的启动	(83)
7.2.2	FORM 应用运行界面的构成	(83)
7.2.3	FORM 应用的退出	(85)
§ 7.3	FORM 运行界面中的导航	(86)
§ 7.4	FORM 应用的工作模式	(86)
§ 7.5	数据库查询	(87)
7.5.1	非受限查询	(87)
7.5.2	受限查询	(87)
§ 7.6	数据库维护	(89)

第二篇 Oracle * Reports 2.5

第一章	Oracle * Reports 概述	(91)
§ 1.1	Oracle * Reports 的基本概念	(91)
1.1.1	什么是 Oracle * Reports	(91)
1.1.2	Oracle * Reports 的功能	(92)
1.1.3	Oracle * Reports 的特点	(93)
§ 1.2	Oracle * Reports 的对象	(93)
1.2.1	模块	(94)
1.2.2	报表级对象	(94)
§ 1.3	报表应用的开发过程	(95)
§ 1.4	Oracle * Reports 的运行环境	(96)
参考资料		(96)
第二章	Oracle * Reports 设计界面	(97)
§ 2.1	交互式设计界面的启动、构成和退出	(97)
2.1.1	启动 Oracle * Reports 设计界面的两种方式	(97)
2.1.2	构成 Oracle * Reports 界面的主窗口	(97)
2.1.3	Oracle * Reports 的退出	(99)
§ 2.2	模块的建立和存储	(99)
2.2.1	模块的建立	(99)
2.2.2	模块的存储	(100)

§ 2.3 模块的安全机制	(102)
§ 2.4 Oracle * Reports 的构成	(103)
§ 2.5 报表应用的定义过程	(106)
§ 2.6 设计环境及其操作	(107)
2.6.1 菜单	(107)
2.6.2 属性表	(109)
2.6.3 工具栏和工具条	(110)
§ 2.7 生成缺省报表应用	(110)
第三章 报表应用的定义	(118)
§ 3.1 定义报表对象	(118)
3.1.1 报表对象	(118)
3.1.2 报表对象的属性	(119)
§ 3.2 定义数据模型对象	(123)
3.2.1 定义查询对象	(124)
3.2.2 定义组对象	(129)
3.2.3 建立连接	(132)
3.2.4 定义列对象	(134)
3.2.5 参数	(142)
§ 3.3 定义显示格式	(145)
3.3.1 显示格式对象的公共属性	(148)
3.3.2 框架	(154)
3.3.3 重复框架	(155)
3.3.4 域(field)	(159)
3.3.5 显示板(boilerplate)	(162)
3.3.6 锚(Anchor)	(165)
3.3.7 按钮(Button)	(167)
3.3.8 OLE2 对象	(168)
§ 3.4 参数表	(169)
3.4.1 域	(169)
3.4.2 显示板	(170)
3.4.3 建立参数表	(171)
第四章 Oracle * Reports 触发器	(173)
§ 4.1 Oracle * Reports 中的 PL/SQL	(173)
4.1.1 PL/SQL 与 Oracle * Reports	(173)
4.1.2 在 PL/SQL 中引用报表对象	(174)
4.1.3 在 Oracle * Reports 中的 PL/SQL 例外	(174)
4.1.4 Oracle * Reports 中的 PL/SQL 结构块	(174)
§ 4.2 外部 PL/SQL 库	(175)
§ 4.3 Oracle * Reports 触发器类型	(176)
4.3.1 报表触发器	(176)
4.3.2 数据模型触发器	(177)
4.3.3 显示格式触发器	(179)
4.3.4 编写公共代码	(183)

§ 4.4 SRW 包	(183)
第五章 定义矩阵式报表	(185)
§ 5.1 定义简单矩阵式报表	(185)
5.1.1 建立单查询矩阵式报表	(185)
5.1.2 建立多查询矩阵式报表	(187)
5.1.3 用零替代无值单元格	(188)
5.1.4 具有合计的矩阵式报表	(189)
§ 5.2 定义复杂矩阵式报表	(191)
5.2.1 建立矩阵间断式报表	(191)
5.2.2 建立嵌套矩阵式报表	(192)

第三篇 Oracle * Graphics 2.5

第一章 Oracle * Graphics 概述	(199)
§ 1.1 Oracle * Graphics 的基本概念	(199)
§ 1.2 Oracle * Graphics 的对象	(200)
§ 1.3 图形应用(display)的开发过程	(201)
§ 1.4 图形应用(display)的运行	(201)
§ 1.5 Oracle * Graphics 的构成	(202)
参考文献	(204)
第二章 Oracle * Graphics 的设计界面	(205)
§ 2.1 交互式设计界面的启动,构成和退出	(205)
2.1.1 交互式设计界面的启动	(205)
2.1.2 交互式设计界面的构成	(206)
2.1.3 交互式设计界面的退出	(209)
§ 2.2 设计环境及其操作	(209)
2.2.1 File 菜单	(209)
2.2.2 Edit 菜单	(210)
2.2.3 Navigator 菜单	(210)
2.2.4 Tools 菜单	(211)
2.2.5 Windows 菜单	(211)
2.2.6 View 菜单	(212)
2.2.7 Chart 菜单	(212)
2.2.8 Format 菜单	(212)
2.2.9 Arrange 菜单	(213)
2.2.10 Help 菜单	(213)
2.2.11 设计环境参数	(213)
§ 2.3 模块管理和应用存储	(215)
2.3.1 模块的种类	(215)
2.3.2 模块的保存	(215)
2.3.3 模块的删除	(216)
2.3.4 模块的改名	(216)
§ 2.4 图形应用的开发过程	(216)

第三章 图形应用的定义	(217)
§ 3.1 创建 display	(217)
§ 3.2 定义查询对象	(217)
§ 3.3 定义图表	(217)
3.3.1 图表的种类	(218)
3.3.2 图表的组成和属性	(219)
3.3.3 图表框架的属性	(222)
3.3.4 域模板的属性	(225)
3.3.5 修改坐标轴的属性	(227)
3.3.6 图表元素的属性	(229)
3.3.7 图表的主从关系	(231)
§ 3.4 定义显示层	(232)
第四章 触发器	(234)
§ 4.1 Oracle * Graphics 封装例程	(234)
§ 4.2 显示格式触发器	(235)
§ 4.3 按钮触发器	(236)
§ 4.4 过滤触发器	(238)
§ 4.5 定时触发器	(240)
第五章 Oracle * Graphics 应用实例	(242)
§ 5.1 实例的背景	(242)
§ 5.2 实例的定义	(242)

第四篇 Developer/2000 工具集成

第一章 Developer/2000 工具集成的基本概念	(249)
§ 1.1 工具的集成及综合应用开发	(249)
§ 1.2 Developer/2000 中的数据相关对象	(249)
1.2.1 Oracle * Forms 中的内部数据相关对象	(249)
1.2.2 Oracle * Reports 中的内部数据相关对象	(250)
1.2.3 Oracle * Graphics 中的内部数据相关对象	(250)
§ 1.3 参数和参数表	(250)
1.3.1 系统参数和用户定义参数	(250)
1.3.2 参数表	(252)
§ 1.4 记录组	(252)
1.4.1 记录组的基本概念	(252)
1.4.2 记录组的使用范围	(253)
参考文献	(253)
第二章 参数的定义和使用	(254)
§ 2.1 Oracle * Forms 中参数的定义和使用	(254)
2.1.1 在 Oracle * Forms 中创建参数	(254)
2.1.2 在 Oracle * Forms 中使用和维护参数	(255)
§ 2.2 Oracle * Reports 中参数的定义和使用	(255)
2.2.1 在 Oracle * Reports 中创建参数	(255)

2.2.2 在 Oracle * Reports 中引用参数	(256)
§ 2.3 Oracle * Graphics 中参数的定义和使用	(257)
第三章 Developer/2000 工具的集成基础	(258)
§ 3.1 开发具有多个 Form 模块的应用	(258)
3.1.1 从 Form 模块调用 Form 模块的方法	(258)
3.1.2 OPEN_FORM/NEW_FORM_CALL_FORM 方法的使用	(259)
3.1.3 在独立 Form 模块间导航	(259)
§ 3.2 从 Form 模块调用 Report 模块的方法	(260)
3.2.1 从 Form 模块调用 Report 模块的方法	(260)
3.2.2 封装例程 RUN_PRODUCT 和 HOST	(261)
§ 3.3 Form 模块与 Graphics 模块的相互调用	(262)
3.3.1 Form 模块与 Graphics 模块的相互调用的机制	(262)
3.3.2 集成 Form 模块和 Graphics 模块的步骤	(262)
§ 3.4 Oracle * Reports 和 Oracle * Graphics 的集成	(264)
3.4.1 以传送数据为目的的应用的设计步骤	(264)
3.4.2 以传送参数为目的的应用的设计步骤	(265)
第四章 参数表和记录组的使用	(267)
§ 4.1 参数表的使用	(267)
§ 4.2 记录组的使用	(267)
§ 4.3 参数表和记录组的应用实例	(269)
4.3.1 实例的背景	(269)
4.3.2 实例的定义	(269)
附录 A 样本数据库	(272)

第一篇 Oracle * Forms 4.5

第一章 Oracle * Forms 概述

§ 1.1 Oracle * Forms 的基本概念

Oracle * Forms 4.5 是 Oracle Developer/2000 中的一个主要产品,是一个开发和运行 Windows 下基于表格应用的工具。从数据库角度看,该工具在原理上与 SQL * Forms 3.0 有许多一致之处,比如集成数据字典,用基表管理应用,把应用分解为对象和属性等。从操作界面看,该工具运行于 Windows 之下,具有非常友好的图形界面,提供有丰富的图形处理功能和对象编辑工具,为应用开发和维护提供了许多方便。所以,如果用户使用过 SQL * Forms 3.0,针对 Windows 下的面向对象程序开发(OOP)比较熟悉(比如 C++),那么学习 Oracle * Forms 4.5 不会有太多困难。另一方面,Oracle * Forms 4.5 与 SQL * Forms 3.0 又有不同,比如,Oracle * Forms 4.5 所支持的对象要比 SQL * Forms 3.0 多,对象的属性也更加丰富,比如,在 Oracle * Forms 4.5 中,Item 对象就有 Text Item, Check Box, List Item, Radio Group, VBX Control, Display Item, Image Item 和 Button 等 8 种,而 SQL * Forms 3.0 仅支持文本型(Text)数据项。在对象创建和属性定义手段方面,Oracle * Forms 4.5 完全使用屏幕图形窗口和工具,操作非常方便直接。总之,由于 Oracle * Forms 4.5 提供更丰富的对象和属性,同时又提供更灵活的应用定义手段,使得应用开发更方便,所支持的应用种类也更广泛,比如带有图象信息的数据库应用等。正因为如此,Oracle * Forms 4.5 更受用户欢迎。

Oracle * Forms 4.5 做为客户(Client)端的开发工具,需要 Windows 环境和 16M 内存作基础,而 SQL * Forms 3.0 仅需 8M 内存。所以如果用户仅处理文本信息,用 SQL * Forms 3.0 就足够了。如果用户需要处理照片、图象等,且希望应用能提供更复杂友好的界面,比如 Button, Check Box, Radio Group 等,则选用 Oracle * Forms 4.5 为宜。

在本系列教程之二 <表格工具、报表工具、菜单工具> 中已详细介绍了 SQL * Forms 3.0。本篇将假定读者已掌握 SQL * Forms 3.0 和 Windows 编程,因此集中介绍 Oracle * Forms 4.5 的定义界面、对象、属性以及触发器等。

§ 1.2 Form 应用的功能和构成

前面已经提到 Oracle * Forms 4.5 只是一个应用开发和运行的支持工具,我们称用该工具开发出的具体应用为一个 FORM。FORM 应用,较用 SQL * Forms 3.0 开发的应用,不仅界面复杂,而且提供更丰富的数据库操作,具体地说,利用 FORM 可完成如下操作:

- 利用各种界面项 (interface item) 插入、更新、删除和查询数据。
- 用文本/ 图象和 VBX (可视化 Basic) 控件显示数据。
- 通过多窗口和数据库事务控制 FORM。
- 直接利用 Oracle Graphics 和 OLE2 (对象连接与嵌入工具) 的应用功能。
- 集成菜单实现综合功能。
- 直接发送数据到 Oracle Reports 等。

FORM 应用是由如下三类模块构成:

- Form (表格) 模块: 表格应用, 支持基于表格的数据库操作。
- Menu (菜单) 模块: 菜单应用, 用以构造层次菜单, 集成应用程序。
- Library (库) 模块: PL/SQL 程序单元的集合。

其中 Form 和 Menu 模块都是可以独立运行的用户应用, 且多把 Menu 直接连入到 Form 中。Library 模块不能独立运行, 而是连到 Form 或 Menu 中, 为这些应用提供共享程序资源。为示区别, 记 FORM (全大写) 表示含有多个 Form 模块, Menu 模块和 Library 模块的应用, 而 Form 仅表示一个单一 Form 模块(应用)。在一个 FORM 应用中, 可以包含多个 Form, Menu 和 Library 模块, 每一个 Form, Menu 和 Library 模块又可以独立开发。另一方面, 在 FORM 中还可直接集成 Developer/2000 的 Graphics 和 Reports 等工具, 所以较 SQL * Forms 3.0 而言, Oracle * Forms 4.5 的应用开发显得更灵活, 功能也更强。

本篇大部分内容将以 Form 模块(应用) 的开发为例, 介绍应用的定义过程, 在讲完 Form 之后, 再分别介绍 Menu 和 Library 的定义, 最后讲 FORM 应用的运行。

§ 1.3 Form 应用的对象

同其它第四代(4GL) 工具一样, Form 应用也是由对象构成的, 定义一个 Form 应用, 同样是理解应用, 分解对象, 定义对象属性等。所以理解 Form 应用的对象及每个对象的属性是学习和开发 Form 应用的关键。

构成 Form 应用的对象如下:

- Form (表格): 表格对象, 定义表格应用的表格级对象及属性。
- Block (块): 是一组项 (Item) 的集合, 是 Form 的逻辑构成单元, 通常分为基表块(数据与数据库基表对应) 和控制块(数据不与数据库基表对应)。
- Item (项): 是 Form 应用处理的数据成员, 是应用所处理的具体信息。
- Trigger (触发器): 由 Form 中事件 (Event) 触发的处理命令或程序段, 用以完成用户的特定处理或构造事物 (Transaction) 处理。
- User-named Program Unit (用户命名程序单元): 用户在 Form 应用内定义的共享 PL/SQL, 过程 (Procedure), 函数 (Function) 或包 (Package) 等。
- Window (窗口): Form 运行时用以显示 Form 画布内容的窗口, 即 Form 应用的所有画布均显示于属于 Form 的所有 Window 对象之内。
- Canvas-View (画布): 是安放 Form 应用可视对象的物理页(画布 Canvas) 的屏幕视图, 在画布上可以粘贴各种背景图形图象(各种项, 如文字/图片/按钮等)。
- 其它对象: 包括 Alert(警示盒), Attached Library(附属库), Editor(编辑器), LOV(值表), Object Group(对象组), Parameter(参数), Property Class(属性类), Record Group

(记录组), Visual Attribute(视觉属性)和 Relation (主从关系)等。

Form 应用的几种主要对象的层次关系如图 1.1 所示:

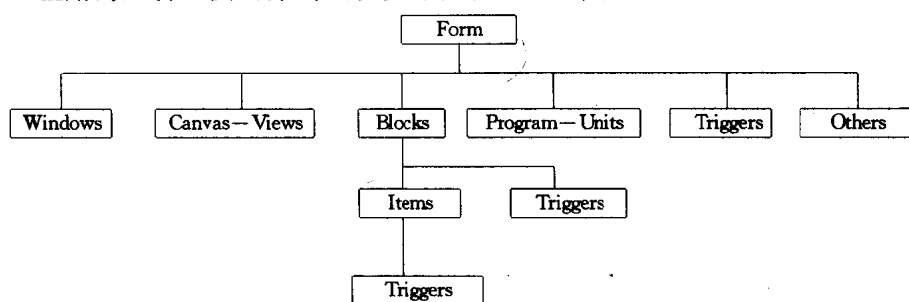


图1.1 Form应用对象的层次关系

其中 Block 和 Item 构成 Form 应用信息的逻辑结构, Window 和 Canvas-View 构成应用信息的物理结构。

§ 1.4 Form 应用的开发过程

使用 4GL (第四代) 工具开发应用, 由于一般 4GL 工具都具有很好的缺省功能, 即都能方便地 提供原型, 使得应用的整个开发在很大程度上是对原型的不断修改和完善。因此, 4GL 工具为用 原型法开发提供了有力的技术支持。

就 Oracle * Forms 应用开发而言, 其过程可以粗略地分为如下几步:

第一步是分析用户的应用需求, 即依据数据库管理员 (DBA) 或系统分析员 (SA) 提供的 应用功能视图 (外模式) 以及用户应用的具体要求, 导出应用的数据及数据组织、格式、 处理规则、操作流程等。只有详细了解用户要求, 优化用户要求, 并得到用户认可, 才能开发 出满意的最终应用。这一步需要应用开发人员和用户一起完成。

第二步是把用户需求识别为 Form 应用对象。这一步应由开发人员完成。要求开发人员 把用 户的具体要求抽象成一个个 Form 对象及其属性。这一步是比较难的一步, 因为即使 是对于一个 训练有素的应用开发人员而言, 要想完全正确地识别出每一个具体应用的所有 对象是一件不容易的事。好在 4GL 工具都有缺省功能, 许多子对象都能缺省生成, 使得开 发人员可以省去好多 麻烦。相信使用过 SQL * Form 3.0 的用户都已体会过这一点。

第三步是应用定义, 即在识别出应用对象的基础上, 利用 Form 定义界面, 对对象自上 而下逐层逐个定义, 定义过程一般先利用缺省功能定义几个 Form 的主要对象, 生成应用的 框架, 从而派生出许多子对象、派生对象和缺省对象, 之后再根据需求修改这些对象, 直到满 足要求为止。

第四步是构造事物处理。对于复杂的应用, 往往事物处理的逻辑操作在应用的对象 属性 中难以体现, 这时需要开发人员设计进一步的触发器 (程序) 来实现这些操作, 这也是 Form 应用 设计的更高一步要求, 也是一个难点。比如要开发一个用以管理转帐的 Form 应 用, 假定应用中仅有帐户 A (Account-A) 和 帐户 B (Account-B) 和钱数 (Amount) 三个可 视项, 表示把 Account-A 的指定钱数 Amount 转入 Account-B。这么一个简单应用就触及到 如下处理:

(a) 检查 Account-A 和 Account-B 是否合法;

- (b) 检查 Account-A 是否透支;
- (c) 封锁 Account-A 和 Account-B (行级封锁);
- (d) 修改 Account-A 和 Account-B 记录,这两个修改要么全提交,要么全回退;
- (e) 释放封锁等。

以上操作从可视对象或对象属性中是无法体现的,需要开发人员去构造该事物处理。

第五步保存应用定义,生成执行文件,试运行。如果应用与用户要求有出入,重复第一至第五步,构成原型法开发的循环圈。

综上所述,应用的开发过程如图 1.2 所示。

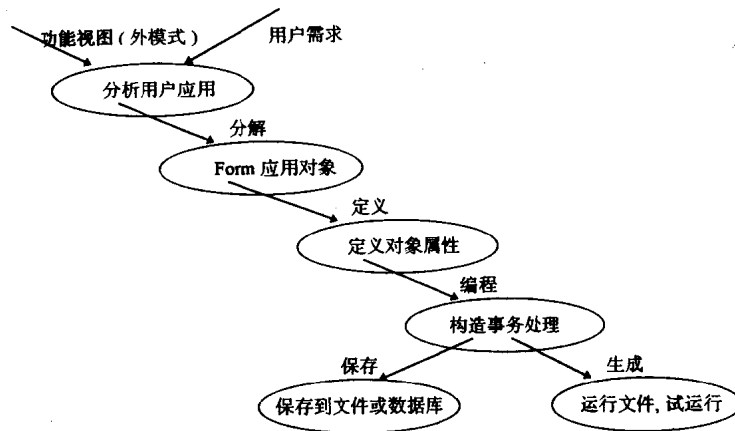


图 1.2 Form 应用的开发过程

§ 1.5 Oracle * Forms 的产品构成和学习过程

从工具角度看,Developer/2000 中与 FORMS 有关的部件有如下三个:

- Oracle * Forms Designer (设计部分):用以设计和存储 Form,Menu 和 Library 模块。该部件包含有图形化设计工具,用来创造对象,设置属性,编写程序等。在设计部件中还可以调用生成部件 (Generte) 和运行部件 (Runform)。
- Oracle * Forms Generate (生成部件):读取模块定义,生成执行文件。
- Oracle * Forms Runtime (运行部件):运行已生成为执行文件的 FORM 应用。

在本篇学习中,将以 Designer 工具为主,介绍 FORM 应用的定义和运行。

由于 Designer 是运行于 Windows 系统之下的一个工具,所以继承和沿袭了许多 Windows 的操作属性,如果读者对 Windows 以及 Windows 下的程序开发比较熟悉,学习 Designer 就会变得轻松得多。本篇将假定读者为 Windows 的经验用户。

了解了 Oracle * Forms 的构成、应用对象及应用开发过程,结合学习其它 4GL 工具的经验,读者不难体会到,Oracle * Forms 的学习过程及方法大致可分为如下步骤:

- 理解 Oracle * Forms 工具的功能及适用范围。
- 掌握设计界面的构成和基本操作。
- 掌握各模块的对象构成。
- 掌握各对象的属性及属性描述。

- 掌握编程接口技术和构造事物处理。
- 保存、生成、运行具体应用。

另外,需要提及一点的是,Developer/2000 的所有产品都有详细的联机帮助资料,并且提供这些资料的打印功能,比如与 Oracle * Forms 有关的联机资料有如下几种:

- Forms Getting Started
- Forms Developer's Guide
- Forms Advanced Techniques
- Forms References Manual
- Forms Messages and Codes

本书只是有关教程,其目的是系统地讲解工具的使用方法,教会读者如何一步步地开发出一个应用,更详细的技术细节读者须查看相关资料。

参考资料

- [1] Oracle Forms (V4.5) I,II
- [2] Forms Getting Started ,Release 4.5 Part No A32504-1
- [3] Forms Developer's Guide ,Release 4.5 Part No A32505-2
- [4] Forms Advanced Techniques ,Release 4.5 Part No A32506-2
- [5] Forms Reference Manual ,Release 4.5 Part No A32507-2
- [6] Forms Messages and Codes ,Release 4.5 Part No A32508-1
- [6] Oracle Forms Reference Manual,Part No A23407
- [7] Oracle 7 Server Application Developer's Guide,PN 6695-70-1292

第二章 Oracle * Forms 设计界面

§ 2.1 交互式设计界面的启动、构成、数据库连接和退出

在 Windows 下,双击 Developer/2000 图标,进入 Developer/2000 组程序管理器,如图 1.3 所示。

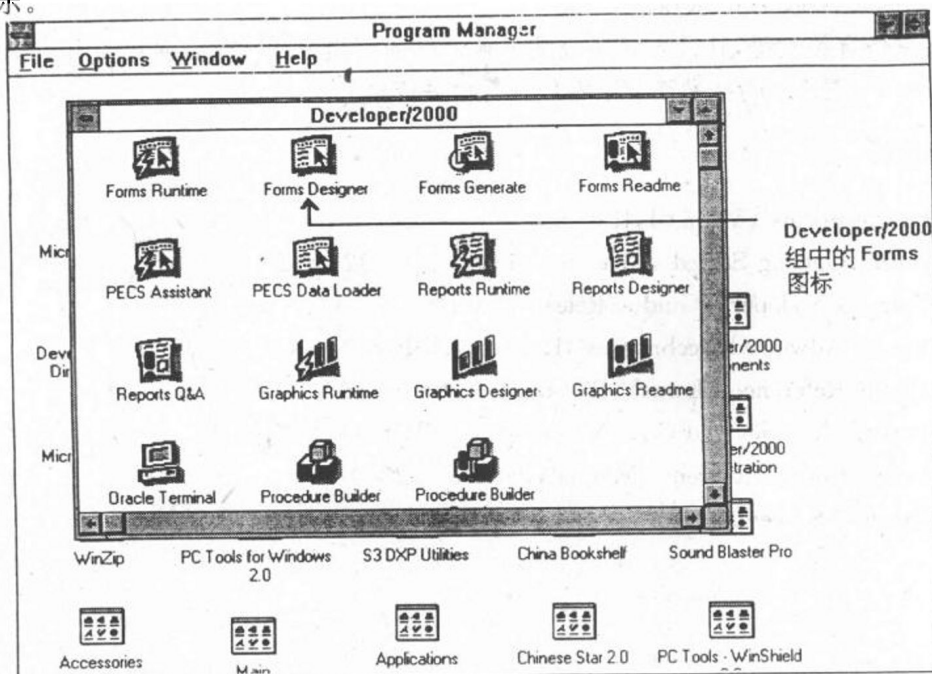


图 1.3 Developer/2000 程序管理器

在 Developer/2000 程序管理器中,双击 Oracle * Forms Designer 即进入交互式设计界面,弹出设计窗口如图 1.4 所示。

该初始窗口由六个下拉式菜单和一个呈现在窗口背景上的对象导航器 (Object Navigator) 窗口构成。六个下拉式菜单分别为:

- File:文件级管理菜单。
- Edit:编辑菜单。
- Tools:工具菜单。
- Navigator: 导航菜单,一般与对象导航器联用。
- Window:窗口菜单。
- Help:帮助信息菜单。

同一般窗口系统类似,File 菜单的功能主要用来建立新应用,打开已有应用,保存应用定义,连接数据库,管理,打印等。File 菜单如图 1.5 所示。

在 FORM 应用定义过程中,不访问数据字典和基表情况下可以不连接数据库,如果需要集成数据字典。比如定义基表块等时,须连接数据库。在 FORM 运行过程中,开发人员可