

第一篇 Oracle Forms

第一章 概 述

1.1 Oracle Forms 的基本概念及结构

我们知道,在 Oracle 数据库系统中提供了数据库语言 (SQL 语言及 PL/SQL 语言),利用这种语言可以对数据库进行各种操作,如查询、检索、修改等等,但是采用这种方法直观性差,操作员的输入工作量大,容易出错。为了解决这些问题,便于用户方便地操作数据库,Oracle 数据库管理系统提供了一种工具软件:Oracle Forms。

Oracle Forms 是以表格的形式呈现在用户面前的,用户通过设计表格、填表来完成对数据库的各种操作。因此,Oracle Forms 的主要功能是自动生成和运行一个可以操作数据库的交互应用。这种交互应用是基于 FORM 来访问数据库中的数据的。FORM 的意思是表格。所谓基于 FORM 的应用是指这些应用完成的工作类似人们在手工数据处理系统对表格的处理:它直接在表格上进行查表、填表、修改表中数据或删除一行、一项等。但是,FORM 所指的表格与手工数据处理系统中印在纸上的“表格”是有很大的区别的,纸上的表格完全是静态的、被动的,而 Oracle 中的 FORM 却可以自动完成许多工作,如可根据用户的要求查表、自动显示有关信息、自动进行计算等等。为了避免误解,在本篇中我们直接用 FORM 一词,而不用中文的“表格”。

我们知道,用数据库语言生成的应用是由大量的程序、函数构成的,那么,由 Oracle Forms 生成的应用(或者说 FORM),其结构如何呢?下面我们详细介绍 Oracle Forms 应用的结构。

1.1.1 Oracle Forms 模块

用 Oracle Forms 生成的应用是基于 Oracle Forms 模块 (Oracle Forms Modules) 建立起来的。Oracle Forms 模块有三种类型:

form 模块 (form module)

菜单模块 (menu module)

库模块 (library module)

其结构如图 1.1 所示。

form 模块:是 Form 对象和程序的集合。这些对象和程序允许用户输入、修改和查询数据库。在 form 模块中可以定义的对象包括文本项 (text item)、校验盒 (checkbox)、按钮 (button)、对话框 (alert)、值清单 (list of values)、图形 (boilerplate graphics) 和被称为触发器 (trigger) 的 PL/SQL 代码块等。

菜单模块:是菜单定义 (一个主菜单和任意多个子菜单) 和菜单项命令的集合。菜单模块可以与单一的或多重的 form 模块相关联。对于多重 form 模块,某些菜单项命

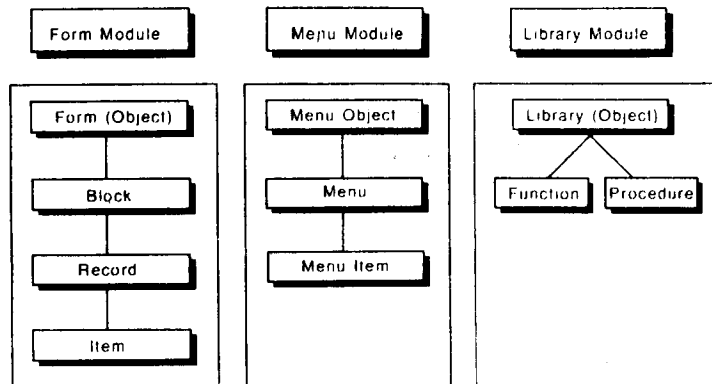


图 1.1 Oracle Forms 模块

令并不是对每个嵌套的 form 都可用。

库模块：是用户自己定义的 PL/SQL 过程、函数和包的集合。这些过程、函数和包可以在整个 form 模块和菜单模块中共享。

由 Oracle Forms 生成的应用，结构可以很简单，也可以很复杂。最简单的应用一般包括一个单一的 form 模块和一个默认菜单。复杂的应用则可包括任意多个 form 模块、菜单模块和库模块，即模块集成——设计者根据需要将各种模块集成在一起，建立一个完整的应用。

如果设计者在一个应用中使用了多个模块，就必须指明各模块间的关系，即定义它们在运行时的调用接口。

- 菜单模块可以连接到 form 模块中。如果设计者在定义一个 form 模块时，指定了该 form 所需的菜单模块，则在运行该 form 时，Oracle Forms 会自动调出与该 form 有关的菜单。在具有多重 form 的应用中，各 form 程序化地调用其它 form。如：用户可以在 form A 中通过选择一个菜单选项调出 form B。然后在 form B 中按一个按钮调出 form C。每个独立的 form 可以与其它 form 共享一个菜单模块，也可以调用它自己的菜单。

- 库模块可以连接到 form 模块或菜单模块中，一旦一个库被连接到 form 模块或菜单模块上，该模块就可以调用存储在该库中的任何函数或过程了。

通过模块集成，设计者可以建立一个复杂的、具有统一界面的应用。由于这样的应用是模块化的，因此易于开发和维护，并可向其它 Oracle Forms 模块及其它 Oracle 应用开发工具（如：Oracle Reports, Oracle Graphics, Oracle Book）的模块扩展。例如：可以在一个 form 中用一个按钮调出定义在 Oracle Reports 中的报告。

在一个单一的 form 模块中可以有很多功能，数量不限。设计一个应用时，是将应用的功能分布到几个模块中去完成，还是在一个大的模块中完成，这需要根据每个应用对其安全性、维护性和可扩展性的要求来决定。

1.1.2 Oracle Forms 对象

Oracle Forms 应用的建立是通过对模块中对象（object）的定义来实现的。这些对象

可以通过 form 模块显示在 workstation 监视器的屏幕或窗口上，以及 PC 屏幕，或字符方式和块方式终端上。Oracle Forms 对象有许多类型，分为以下几类：

- form 模块：
- 处理对象
 - 界面对象
 - 逻辑数据结构

- 菜单模块：
- 对象
 - 菜单项
 - 逻辑数据结构

- 库模块：
- 对象
 - 逻辑数据结构

所有这些对象提供了很强的功能，可以满足用户的需要。在一个应用中，用户可以只使用其中一部分对象。

一、form 模块

form 模块的结构如图 1.2 所示。

1. 处理对象

form 模块包含的处理对象有 form 对象、块 (block) 和项 (item)。

• form 对象

form 是基本的应用对象和结构，它包含对象和指令的所有运行定义，这些对象和指令组合起来，共同完成用户应用的功能。在用户屏幕上，一个 form 被显示成一个或多个窗口，其中包含一个或多个画板 (canvas) 和一组界面对象，如：按钮、校验盒和窗口。form 不像其它对象，它不需要人工定义，每个新定义的 form 模块自动地包含一个默认的 form。用户可以通过修改 form 的属性来改变 form 模块在运行时和设计时的默认特性。

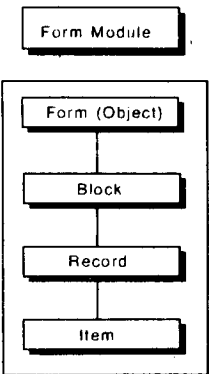


图 1.2 form 模块结构

• 块

块是一些界面项 (如：文本项、校验盒等) 的逻辑组合。一个块可以直接与一个数据库视图或数据库表相关联，这意味着块中的每个项 (item) 可以与表 (又称为基表 base table) 中或视图中的列 (column) 相关联。这种直接的联系使用户能够查询、修改、插入和删除基表或视图中的数据。块是以记录 (record) 方式显示数据的 (记录是 form 模块的一种逻辑数据结构，见本节中 “3. 逻辑数据结构”)。一个块也可以不与数据库有任何关系，如控制块。Oracle Forms 提供了不同的块类型。Oracle Forms 通过块的属性和

块中项的属性，以及块在 form 中完成的功能来识别块的类型。块的类型如下：

基表块 (base table block) —— 直接与数据库表或视图相关联的块，即块中含有一个或多个基表项。注意：基表块中也可以含有非基表项。

控制块 (control block) —— 控制块不与任何数据库表相关联，可用于显示计算值或信息等。

细目块 (detail block) —— 在主-细目块关系 (master-detail block relation) 中。一个细目块与一个主块相联系。细目块显示与主块中主记录相关的那些细目记录。

主块 (master block) —— 在主-细目块关系中，一个主块与一个细目块相联系。主块显示与细目块中细目记录有关的主记录。

多记录块 (multi-record block) —— 一次可以显示多个记录的块。

单记录块 (single-record block) —— 一次只能显示一个记录的块。

不可进入块 (non-enterable block) —— 这种块由不可进入的项组成。光标不能进入这种块的任何项。

事务触发器块 (transactional trigger block) —— 这种块允许用户增加和/或替换标准的 Oracle Forms 数据库交互作用。

• 项

项是一个图形单位，通常可在屏幕上看到。一个项一般对应数据库表中的一列 (column)。在一个应用中，用户可以通过项来执行数据库的维护（如：输入、编辑、查看数据）、激活一个操作等等。Oracle Forms 提供了三种类型的项。

基表项 (base table item) —— 对应于数据库表中的一列。

控制项 (control item) —— 不与数据库表中的任何列相关，通常用于显示计算值或信息，还可用于接收用户输入的数据。用户通过这些数据完成计算、菜单操作或其它操作。

查找项 (look-up item) —— 对应于一个表中的一列。该表不是查找项所属块的基表。用查找项可以显示其所属块中其它非基表的数据。但这些数据是只读的，即用户不能在查找项上对数据执行查询、修改、插入或删除操作。

2. 界面对象

Oracle Forms 为图形用户界面提供了若干界面对象。图 1.3 中显示了 form 模块中的一些界面对象。

- 按钮 (button)：按钮允许用户执行一个预先定义好的操作。
- 图表项 (chart item)：Oracle Graphics 图表可以输入到图表项中，并通过程序连接到用户的应用中。
- 校验盒 (check box)：校验盒允许用户设置 On 或 Off 状态来表示打开或关闭一个校验。
- 显示项 (display item)：显示项与文本项类似，但它显示的内容在运行时不能被编辑。显示项通常用于显示图形或文本。
- 图象项 (image item)：图象可以从数据库或文件中输入到图象项中。图象项可以

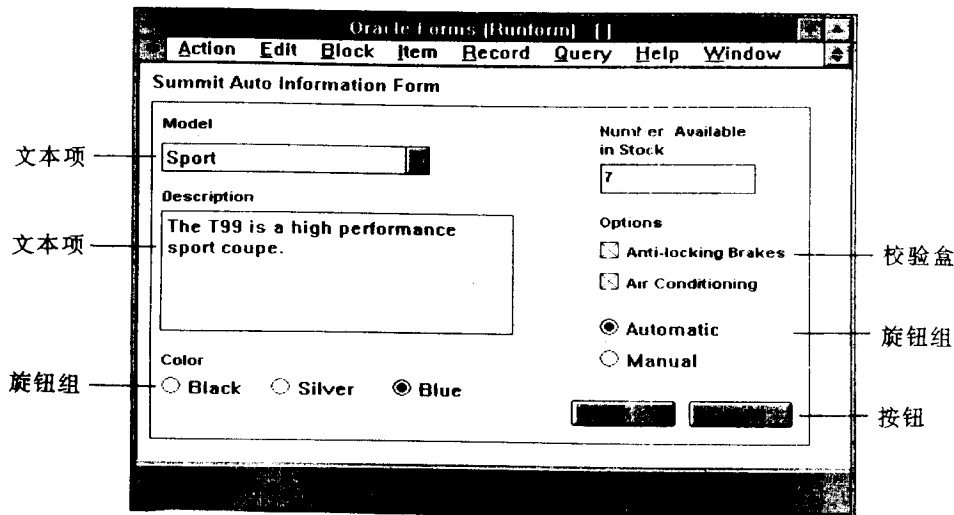


图 1.3 界面对象

用标准的裁剪 (cut)、复制 (copy) 和粘贴 (paste) 功能进行维护。

- 旋钮组 (radio group): 旋钮组包括一组旋钮开关。在一个旋钮组中一次只能打开一个旋钮。
- 文本项 (text item): 文本项在屏幕的一个单行或多行域中显示可编辑的文本, 可能与数据库表的列有关联。
- 用户区 (user area): 用户区是一个带有用户出口的项。用户出口是用户用第三代语言 (3GL) 编写的程序, 连接到 Oracle Forms 可执行文件中, 用来完成具有特殊目的的任务, 如: 连接非 Oracle 数据源。对于 Oracle 工具箱 (Oracle ToolKit), 用户出口代码必须使用 API 来控制用户区项的内容。
- 窗口 (window): 窗口是一种屏幕区域或环境, 其格式可以是固定的, 也可以是不固定的, 它为画板 (canvas) 或其它可显示的界面对象提供一个框架。窗口可以有滚动棒和其它元素, 这依赖于用户所使用的实际的窗口管理机制。窗口有两种显示方式: 文卷方式 (document) 和对话方式 (dialog)。文卷方式的窗口显示在应用的根窗口中, 其显示的位置和大小是可以改变的。对话方式的窗口是固定格式的窗口, 通常显示两类内容: 或者是一条信息和一个可以使窗口从屏幕上消失的按钮; 或者是一系列的域、校验盒和旋钮组。
- 画板 (canvas): 画板是窗口内的一个作图平面。界面对象、图形对象、文本项及其它 GUI 界面对象都可以显示在画板上。
- 对话框 (alert): 对话框是一种固定格式的窗口, 用于显示一条信息, 告诉用户他的操作所产生的情况。一个对话框中最多为用户提供三个可选择的按钮操作。如图 1.4 所示。
- 编辑器 (editor): 编辑器是一个具有文本编辑功能的固定格式窗口, 如图 1.5 所示。
- 值清单 (list of values, 简称 LOV): 值清单是一种固定格式窗口, 其中显示一个可滚动的清单, 用户可以从中选取一个值。LOV 的值对应于数据库表中的列或设

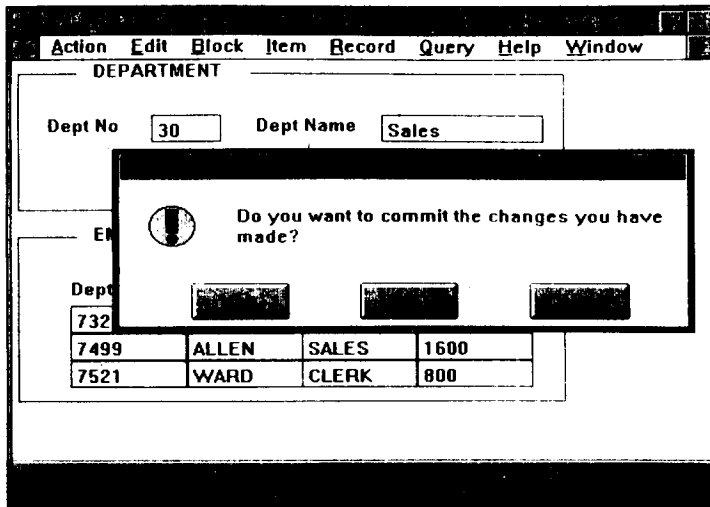


图 1.4 对话框 (alert)

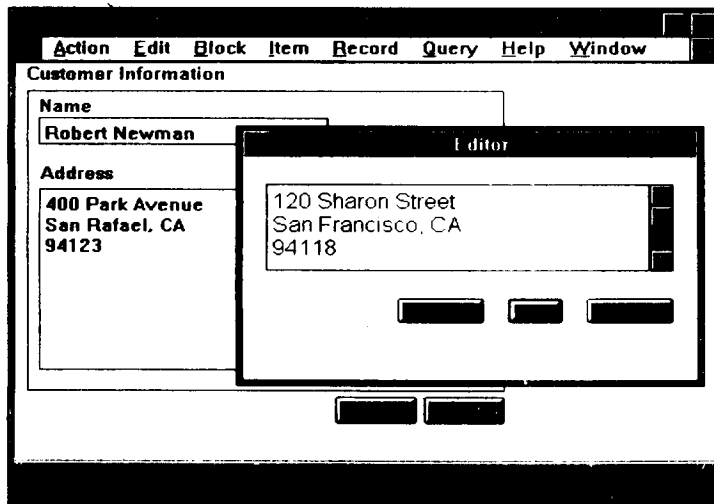


图 1.5 编辑器 (editor)

计者在设计时建立的静态值。LOV 可以通过用户请求调出或从程序中调出。如果 LOV 可以使用，则 Oracle Forms 在屏幕底部的状态行上显示<List>。如图 1.6 所示。

3. 逻辑数据结构

在 form 模块中，Oracle Forms 提供了一组逻辑结构，可以把数据子集逻辑地组织在一起，形成具有一定意义的组。Oracle Forms 包括以下逻辑结构：

- 记录 (record)：记录是包含一组相关项的数据结构。可对应 Oracle 数据库表 (table) 中的一行 (row)。
- 记录组 (record group)：记录组是像表 (table) 一样的包含相同行/列框架的内部数据结构，是设计者在设计时静态地或程序化地定义的。记录组是构成值清单 (LOV) 和参数列表 (ParameterList) 的基本数据结构。

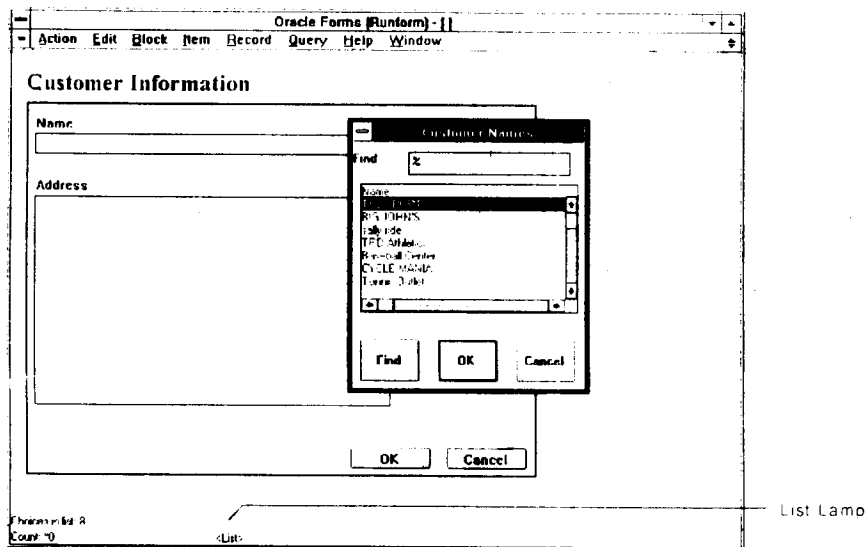


图 1.6 值清单 (list of values)

- 参数列表 (parameter list): 参数列表是一个包含对偶“参数=值”的列表。用于传送信息到任何 Oracle 应用开发工具中, 如: Oracle Forms、Oracle Graphics、Oracle Reports。
- 计时器 (timer): 计时器是一个程序化的设备, 可以在触发器中用内部程序定义、激活和删除。其精度可设置到毫秒。在计时器消亡时可能会引发一个事件。
- 关系 (relation): 在具有多个块的 form 中, 通常需要定义块与块之间的关系, 如: 主—细关系、主—主—细关系。Oracle Forms 的内部机制使用户能很容易地定义这些关系。
- 触发器 (trigger): 触发器是一个代码块, 可以用于增加应用的功能。每个触发器有一个名字, 包含一条或多条 PL/SQL 语句。触发器是作为特殊的对象来维护的。当某一事件发生时, 将会激活与该事件有关的触发器。
- 用户 PL/SQL (user-named PL/SQL): 用户 PL/SQL 是由用户编写和命名的程序 (包括过程和函数) 和包 (package)。form 模块、菜单模块和库模块中都可以包含 PL/SQL 代码。

4. 对象的分层结构

对象的分层结构由四种 form 模块对象组成: form、块、记录、项 (见图 1.2)。

在 form 模块中, form 是顶层对象, 可以包含一个或多个块。若包含多个块, 则必须包含各块间的关系, 如主—细块关系。

一个块最多只能对应一张数据库表, 它是以“记录”这种数据结构方式显示表中的数据的。当用户执行一次查询时, 返回的每条记录对应于表中的每一行。

一个块中含有若干项。通常, 一个项对应于数据库表中的一列。但表中的列与块中的项不一定一一对应。例如: 如果块中包含控制项, 则该项不与表中的任何列对应。

form 与数据库之间的对应关系如表 1.1 所示。

表 1.1 form 与数据库的对应关系

form 对象	数据库元素
块 (block)	表 (table)
记录 (record)	行 (row)
项 (item)	列 (column)

二、菜单模块

1. 对象

菜单模块是由下列对象组成的，如图 1.7 所示。

- 菜单对象 (menu object)

一个菜单模块中包括一个或多个菜单（每个菜单由菜单项构成）。所有菜单构成一个分层结构，或称“树”结构，即其中一个菜单为主菜单（即顶层菜单），其余菜单为子菜单。

- 菜单 (menu)

在一个菜单对象中，主菜单和子菜单在菜单模块中是被命名的对象。

- 菜单项 (menu item)

每个菜单由一个或多个菜单项组成，当用户在菜单上选择一项时，Oracle Forms 执行赋予该项的命令。菜单项命令可以调出子菜单或执行应用功能。菜单项有四种类型，如图 1.8 所示。

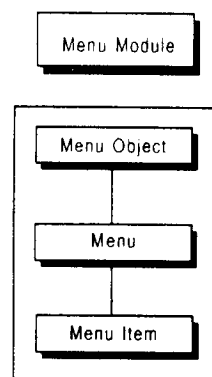


图 1.7 菜单模块

简明菜单项 (plain menu item) —— 只以简单的文本显示在菜单中。这是最常用的菜

FIGURE 2-8:

旋钮菜单项

分类菜单项

校验菜单项

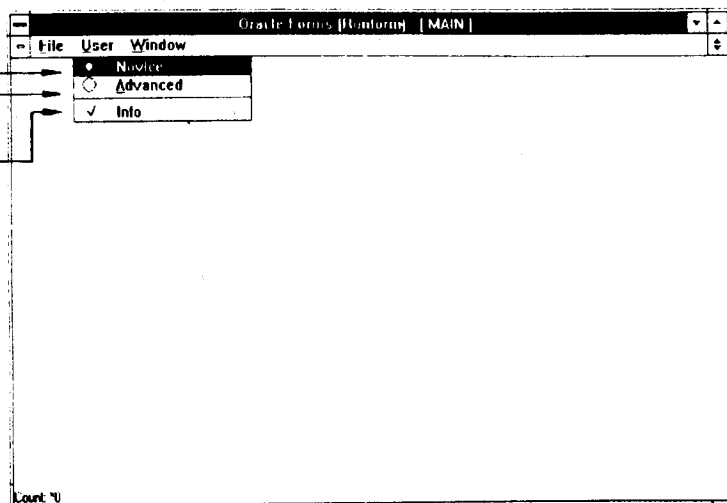


图 1.8 菜单项类型

单项类型。

校验菜单项 (check menu item) —— 类似一个校验盒，有两种状态：On 或 Off (打开或关闭)。这种菜单项带有一个可以显示其当前状态是 On 还是 Off 的可视符，如“√”表示 On，空表示 Off。

旋钮菜单项 (radio menu item) —— 类似一个旋钮开关，也有两个状态。一个旋钮菜单项总是一组两个或多个相关菜单的一部分，它带有一个可以显示当前状态的图符，如空心圆或实心圆。

分类菜单项 (separator menu item) —— 是一个水平线，或是某种其它特定 form 平台的可视元素。用于在同一菜单中从功能上区分相关的菜单项和无关的菜单项。用户不能选择这种菜单项。

2. 逻辑数据结构

在菜单模块中有两种逻辑数据结构：

- 置换参数 (substitution parameter)：是一个交互变量，可以嵌入到菜单项命令语句中。用户可使用内部的置换参数，也可以自己定义置换参数。

- 用户 PL/SQL (user-named PL/SQL)：是由用户编写和命名的程序（过程和函数）和包。这些 PL/SQL 代码可以包含在 form 模块、菜单模块和库模块中。

三、库模块

库模块的结构如图 1.9 所示。

1. 对象

• 库 (library)

库是存储 PL/SQL 程序单元的场所。用户可以将自己的 PL/SQL 程序存入库中，然后将库连接到 form 模块或菜单模块上，这样该模块就可以调用存储在库中的任何程序了。

2. 逻辑数据结构

一个库中有以下几种逻辑数据结构：

- 过程 (procedure)：用户过程 (user-named procedure) 是用户为了增强模块的功能而定义的 PL/SQL 块。

- 函数 (function)：用户函数 (user-named function) 是用户为了增强模块的功能定义的 PL/SQL 块。执行后返回一个值。

- 包 (package)：用户包 (user-named package) 是由用户定义的数据库对象，它从逻辑上将相关的 PL/SQL 的类型、对象和过程分组。

用户 PL/SQL (包括用户过程、用户函数和用户包) 是用户编写和命名的程序和包，可包含在 form 模块、菜单模块和库模块中，加强模块的功能。

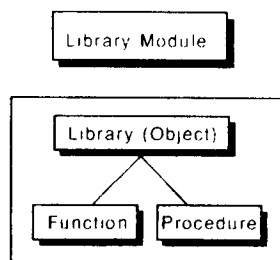


图 1.9 库模块结构

1.2 Oracle Forms 的应用开发过程

一个 Oracle Forms 应用开发过程分为设计、生成和运行三步。对应这三个步骤，Ora-

cle Forms 提供了三个开发工具：

Oracle Forms 设计器——Oracle Forms Designer (DES40)

Oracle Forms 生成器——Oracle Forms Generate (GEN40)

Oracle Forms 运行器——Oracle Forms Runform (RUN40)

应用开发人员首先用 Designer 定义一个应用，然后用 Generate 生成该应用的运行文件，得到运行文件后可以用 Runform 试运行。如发现运行出错，反复用 Designer、Generate、Runform 进行重新定义、生成、试运行，直至没有错误。这样开发人员就构造好了一个应用，可以交给用户使用了。用户运行该应用时也是通过 Runform 实现的。

1.2.1 设计

Oracle Forms 安装好后，开发人员可以用下列命令启动 Designer：

f40des userid=用户名/口令

启动 Designer 后就可以开始设计一个应用了。设计应用实际上是定义应用的模块，即用填表的方式定义模块中的各种对象，并写少量必需的代码程序。为了方便用户设计，Designer 中提供了一些设计工具，如：默认块生成工具、图形编辑器、菜单编辑器、PL/SQL 编辑器等等（见 3.5 节）。完成了必要的定义和编码工作后，一个应用就设计好了，可以存储到文件系统或数据库中。Designer 是以二进制格式存储一个应用的各种模块（form 模块、菜单模块、库模块）的。各种模块的二进制文件扩展名分别为 .FMB、.MMB、和 .PLL（见 1.2.4 节）。为了能使设计人员阅读模块文件中的内容，Designer 提供了文件格式转换功能，可将 form 模块文件和菜单模块文件从二进制格式转换为文本格式。文本格式文件的扩展名分别为 .FMT、.MMT（见 1.2.4 节）。

1.2.2 生成

设计好一个应用后，就可以用 Generate 生成此应用中各模块的运行文件了。启动 Generate 可用如下命令：

f40gen 模块名 userid=用户名/口令 module_type = 模块类型

Generate 所产生的运行文件针对不同的模块有不同的文件扩展名，分别为：.FMX、.MMX 和 .LIB（见 1.2.4 节）。

例如：如果开发人员要设计一个包含三种模块的名为 Order 的应用，经 Designer 设计并存储后，产生三种模块的二进制文件 Order .FMB、Order .MMB 和 Order .PLL。输入命令：f40gen Order userid=用户名/口令 module_type = 模块类型，启动 Generate，将生成 Order .FMX、Order .MMX 和 Order .LIB。这样就构造好了应用 Order。

1.2.3 运行

用 Runform 可以运行一个已构造好的应用。启动 Runform 的命令如下：

f40run 应用名 userid=用户名/口令

例如：运行应用 Order 可用命令 f40run order userid=用户名/口令。

1.2.4 模块文件

构造一个应用所能产生的各种格式的模块文件如表 1.2 所示。

表 1.2 各种格式的模块文件扩展名

模块	二进制文件	文本文件	运行文件
form	• FMB	• FMT	• FMX
菜单	• MMB	• MMT	• MMX
库	• PLL	/	• LIB

在三种模块（form、菜单、库）中，form 模块是应用的基本模块，即一个应用必须有 form 模块。其它两种模块是应用的附加模块，这两种模块的运行文件不能直接用 f40run 命令执行，即 f40run 命令中的应用名实际上指的是 form 模块的运行文件（.FMX 文件）。若设计人员在设计一个应用时定义了菜单模块和库模块，则必须把菜单连接到 form 模块上，把库连接到 form 模块或菜单模块上，这样，用 f40run 运行该应用时，Oracle Forms 会在运行 .FMX 文件的过程中根据需要自动调用 .MMX 和 .LIB 文件，从中调出所需的菜单和库中的程序来执行。

1.3 Oracle Forms 的用户界面

为了适应各种计算机硬件平台，方便用户操作，Oracle Forms 采用窗口工作方式，提供了菜单和大量的功能键。对于有鼠标器及图形终端的用户，利用菜单来操作 Oracle Forms 是极为方便的；对于使用不带鼠标器或只有字符终端的用户，使用功能键也能完成与菜单相同的功能。在大多数平台上的 Oracle Forms 中，菜单和功能键是并存的，用户可根据个人的喜好灵活使用。

1.3.1 菜单

Oracle Forms 的菜单有三种显示方式：下拉式、条形方式和全屏幕方式。下拉式菜单在窗口的菜单行上显示主菜单，当用户选中其中一个菜单项时，若该项有子菜单，则在该项的下方显示其子菜单。在条形菜单中，主菜单及其所有子菜单都以水平菜单条形方式显示在窗口的菜单行上。当用户选中一个有子菜单的菜单项时，新的子菜单将替换掉原来的菜单显示在同一菜单行上。全屏幕菜单方式是以竖式、带序号的列表方式显示菜单的。

Designer 的菜单是以下拉方式显示的，如图 1.10 所示。

Runform 在运行时根据开发人员在设计时对应用的定义来显示菜单，若设计人员没有定义菜单，则 Runform 使用默认菜单。默认菜单是以下拉方式显示的，如图 1.11 所示。若设计者定义了菜单，则 Runform 根据定义来显示菜单，图 1.12 为用户定义的一个条形菜单的例子。

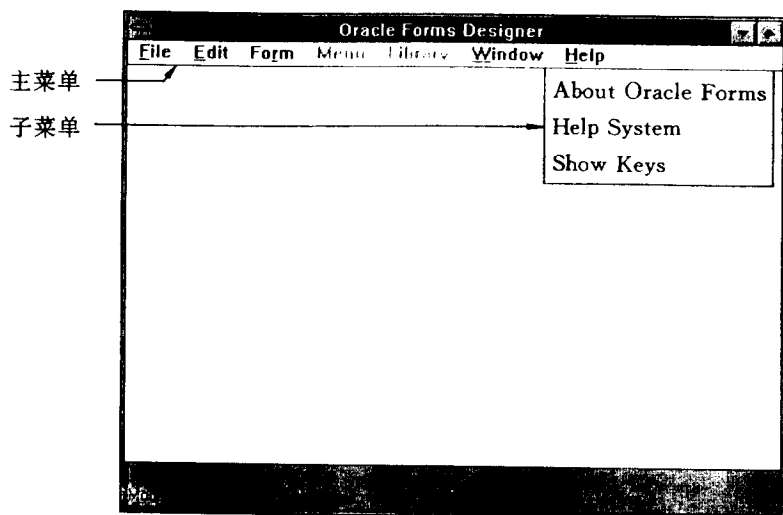


图 1.10 Designer 菜单窗口

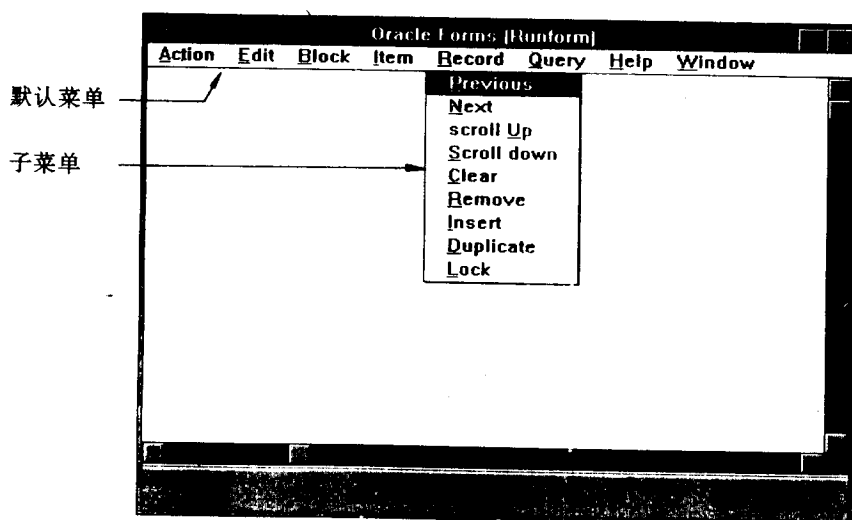


图 1.11 默认菜单

1.3.2 功能键

Oracle Forms 提供了两套功能键，一套在 Designer 中用于定义应用，另一套在 Runform 中用于执行应用。定义功能键包含两部分含义：一是指定各个功能键完成什么功能，二是为触发这一功能在键盘上应操作什么键。我们不妨把前者称为“逻辑键”定义，后者称为“物理键”定义。

在不同的系统和不同的终端上运行 Oracle Forms 时，定义的逻辑键都是相同的，然而逻辑键对应的物理键可能不相同。由于所有的操作都是通过在键盘上打入相应的物理键来完成，所以，应用的设计者和使用者一定要知道自己所使用的系统键盘图，即逻辑键与盘上的物理键的对应图，如图 1.13 所示。了解键盘图有两种方法：

- 运行 Oracle Forms 后，按下功能键 [Show Keys] 即可显示出键盘图。只要用户

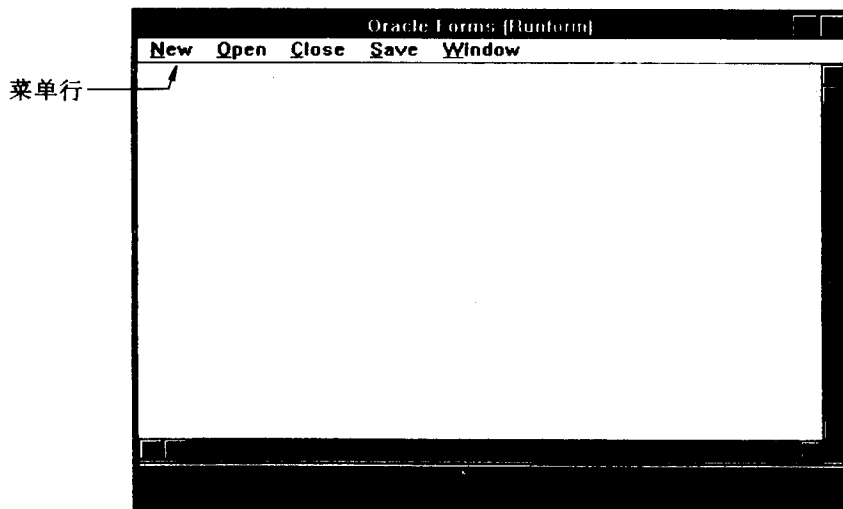


图 1.12 条形用户菜单

熟记 [Show Keys] 键的物理键，就可随时查看所有功能键对应的物理键，对于多数系统，此键对应于键盘上的 ESC 和 K 两键或 CTRL 和 K 两键。

- 运行 Oracle Forms 后，使用联机帮助功能。

1.3.3 联机帮助

为了便于用户操作，Oracle Forms 提供了联机帮助功能，使用 Oracle Forms 的用户几乎可以在任何时候通过菜单选择或按 [Help] 键请求帮助。Oracle Forms 可根据用户的要求提供简要的或详细的帮助信息。显示键盘图即是 Oracle Forms 的联机帮助功能之一，如图 1.13 所示。

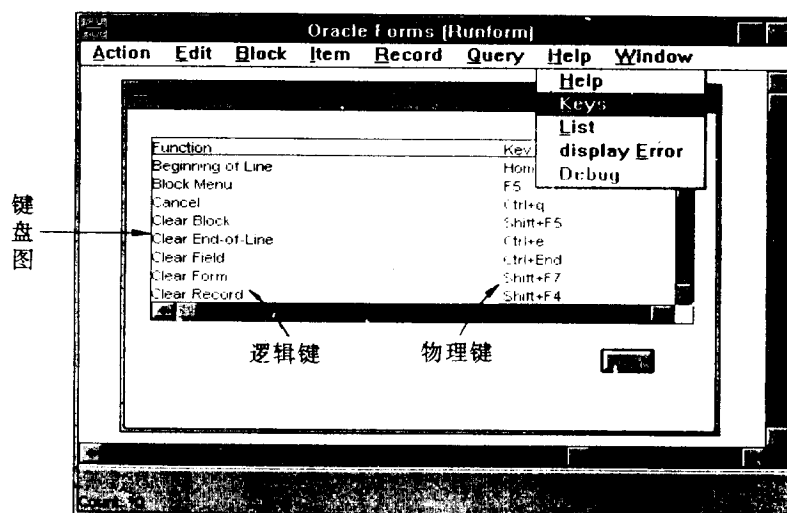


图 1.13 联机帮助及键盘图

第二章 Oracle Forms Runform——运行一个应用

构造好一个应用后就可以运行它了。设计者可以自己运行，也可以授权其它操作员或用户运行。所谓运行一个 Oracle Forms 应用，实际上就是对 Oracle 数据库中的数据执行查询、输入、修改或删除等操作。Oracle 数据库中的数据都是以表的形式组织的，即一些相关的信息放在一张表（table）中。为了保护数据库免受破坏，运行 Oracle Forms 时并不是直接对数据库的表进行操作，而是对其在工作空间中的副本进行操作，当确认操作无误后，才将结果写入数据库中（即“提交”），如果发现操作有误，还可以使其作废（即“回滚”），这样就不会使错误的操作影响到数据库的表本身。Oracle Forms 提供了专门用于运行应用的工具：Oracle Forms Runform。

本章介绍 Oracle Forms Runform 以及如何运行一个应用。主要包括以下内容：

- Runform 的用户界面
- Runform 的基本操作
- 查询和修改数据库
- 从数据库中检索记录
- 提交和回滚事务
- 使用默认菜单和用户菜单

2.1 Runform 的用户界面

每个应用都有一个内部窗口（built-in window）称为根窗口（root window）。当启动 Runform 运行一个应用时，屏幕首先显示该应用的根窗口，典型的根窗口如图 2.1 所示。窗口的顶部有菜单行，显示定义在应用中的菜单；底部用于信息提示和状态说明；中间部分为信息显示（包括文本和图形）和数据接收区。

2.1.1 信息行和状态行

Runform 的信息行和状态行如图 2.2 所示。

信息行：显示 Oracle Forms 的信息和应用的信息。如：提示信息 and 出错信息。

状态行：从左至右包括以下内容：

Count：指明查询所检索出的记录个数。每显示一个检索出的记录时，Count 增 1。当检索到最后一个记录时，在 Count 之前显示一个 * 号。

Λ：表示在本块的当前记录之前有记录。

V：表示在本块的当前记录之后有记录。

ENTER QUERY：表示已进入查询状态，允许输入查询检索条件。

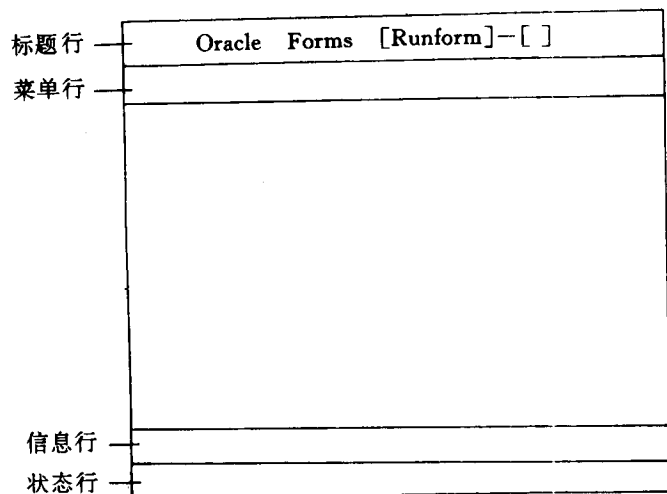


图 2.1 Runform 的根窗口

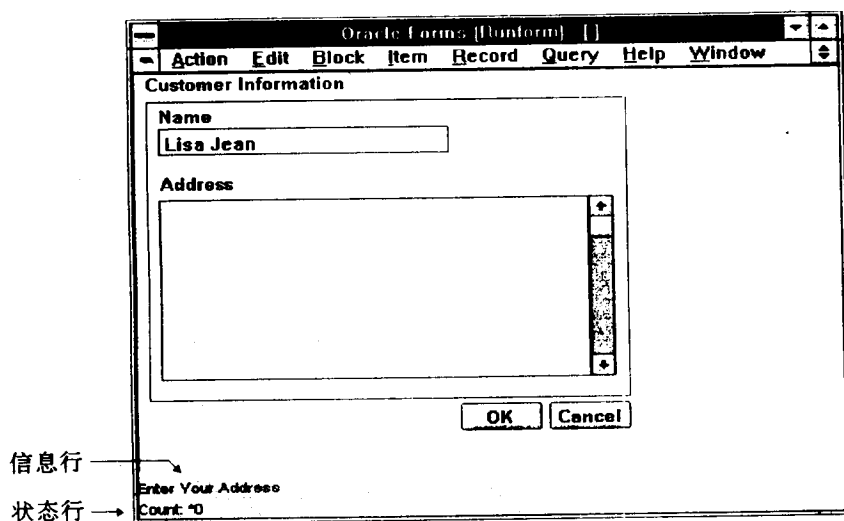


图 2.2 Runform 的信息行和状态行

<BGM>: 表示该应用含有背景菜单 (详见 2.6.4 节)

<List>: 表示当前项有一个值清单, 可从值清单中选择一个值作为该项的输入值。

<OSC>: 表示操作者有权在 Oracle Forms 中使用操作系统命令。

<Insert>或<Replace>: 说明当前字符方式是插入方式 (insert) 还是替换方式 (replace)。

2.1.2 菜单

Runform 菜单包含一系列选项, 它们是一些预先定义好的任务或行动步骤。当用户