

第一部分 建立应用程序

第1章 介绍Developer

Oracle Developer是一个综合的高度有效的开发环境。它可以为因特网或客户机/服务器系统建立可扩充的数据库应用程序。Oracle Developer Server是一个Application Server，专门用来配置和优化因特网的Oracle Developer应用程序。本书为你讲授在开发功能强大的、可重用的、有效的、用于访问数据库中的数据应用程序时所需要了解的一切。

1.1 综述应用程序的建立

应用程序是执行某个任务的计算机程序。数据库应用程序是在数据库管理系统（如Oracle 8）中管理数据的程序。应用程序显示数据，通过添加和修改将数据移入和移出数据库。

建立应用程序系统有许多方法。可以使用各种程序语言，如 COBOL、C、C++或Java。程序语言很多，每个程序语言都有其优缺点。用这些语言编程提供了很大的灵活性、控制权和性能。但是必须努力工作才能充分利用这些特点。甚至要更加努力工作才能确保应用程序在日益广泛的应用中使用。这种编程方法容易发生错误。不精心编制的程序容易失败，降低而不是加速程序执行的速度。

Oracle Developer是一种说明性工具。用它建立应用程序，主要是告诉数据库要做什么，而不是告诉它如何做。只要用几条语句或插入几个参数，就可以生成可扩充的、能够在 Web 上运行业务的应用程序。在 PL/SQL 中另外编制一些代码，就可以做许多想要做的事情。

当前，对因特网开发的应用程序包含有带菜单、工具栏、对话框和窗口的丰富的图形用户界面，不管应用程序的对象是什么，都可以采用窗口显示。我们感兴趣的三种对象是表单、报表和图形。Oracle Developer分别为这三种对象提供了模块：Oracle Form(表单)、Oracle Report(报表)和Oracle Graphic(图形)。

如果不需要高效的配置应用程序，则开发一个应用程序不是很困难。以前，使用一个应用程序就意味着要找出一种方法使用“人工传递网络”(SNEAKERNET)将应用程序安装在商店中所有的用户机器上。现在，采用内部网、外部网和因特网，使用起来非常容易；但扩大则比较困难。只要某个地方有运行应用程序的服务器，则任何人只要有 Web 浏览器，就可以运行现在的应用程序。Application Server(如Developer Server)的任务是执行服务器的应用程序的业务逻辑。这个服务器连接到浏览器上显示表单、报表和图形。

1.1.1 表单应用程序

要了解Oracle Developer，需要了解一些与应用程序对象有关的术语。

表单应用程序(form application)是一种以联机方式的格式提供数据的应用程序。它由一系列或多个窗口上部署的项目组成，这些项目可能是可输入的字段、图形、按钮、复选框或任

何数目的对象。这里给出一个表单应用程序的示例：

如图1-1所示，表单提供了观看数据库信息的一个很好的方式。可以将数据输入到表单的域中，也可以修改域中的数据，这取决于所使用的应用程序。

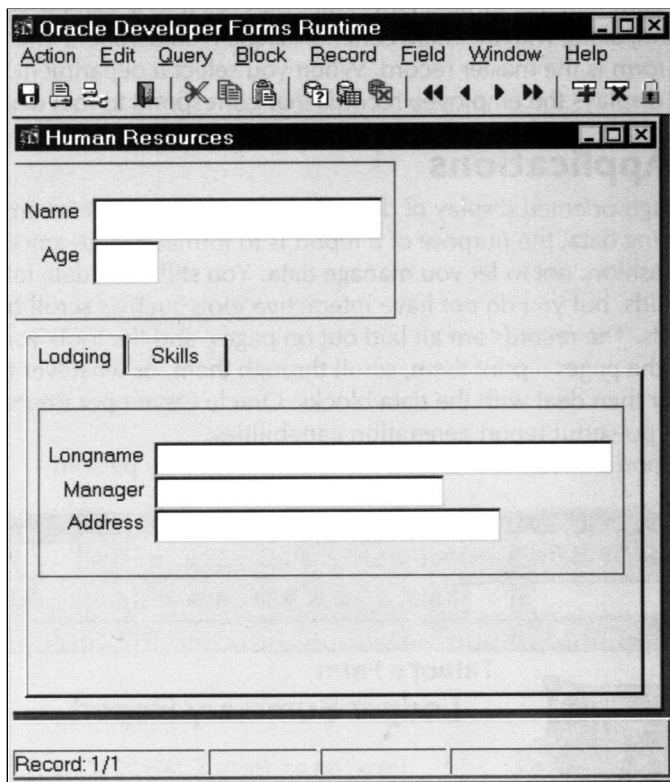


图1-1 表单应用程序示例

依照应用程序所要完成的任务和运行它所要求的体系结构而定，一个应用程序可以只有一个表单或也可以有多个表单。如文件的格式一样，表单的设计也是一种艺术：必须以正确的格式显示信息，否则会难以使用并且容易出错。将一个非常大而且复杂的表单分成几个部分，是减少其复杂性的常用方法。在联机应用程序中，也采用同样的处理方法。简单的表单对应于记录类型，它是一系列数据字段——单行数据。表单可以一次显示单个记录，或者可以同时显示几行。表单中的工具可以管理记录和显示信息的交互操作（查询、插入、更新、删除、滚动等）。

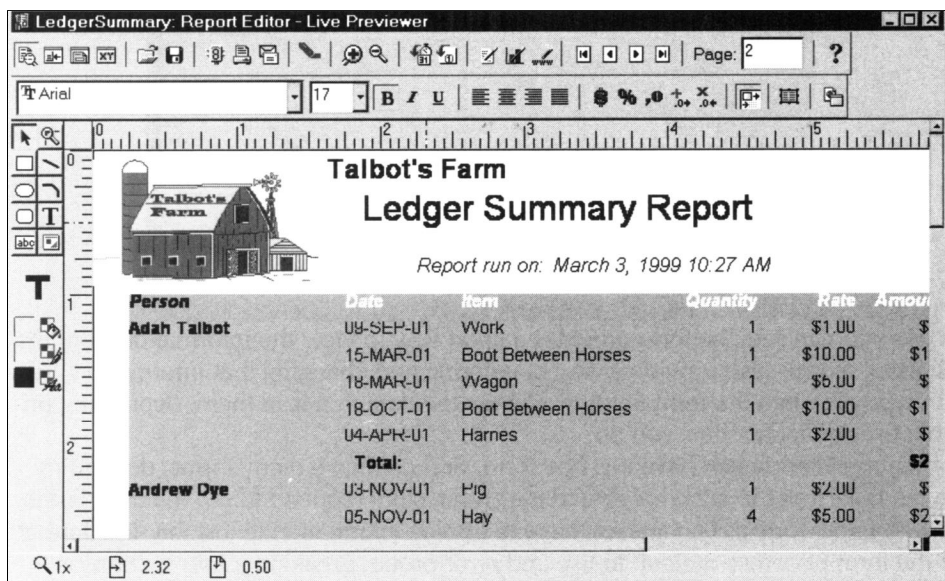
有一种特殊类型的表单叫做主-从(master-detail)表单，它自动地管理父记录与几个子记录的关系。这种情况很普遍，所以 Oracle Developer将它和更加普遍的单个记录一起作为建立表单的专项目选择。表单的各个部门是主记录，当选择一个部门时，表单自动地显示与这个部门对应的雇员情况。

1.1.2 报表应用程序

报表面向页面显示数据。表单为数据管理提供了交互式的工具，而报表的目的是以一种可读的形式来格式化大量的数据，而不是管理数据。报表数据仍然在记录和字段中部署，但

是没有交互工具(例如滚动条)去处理记录。记录在页面上部署,处理页面的工具有页面打印、翻页和一些格式允许的操作,但不处理数据。Oracle Developer Reports Builder有很强的报表生成能力。

图1-2是一个报表,表示按照人员分组的Ledger项目:



Person	Date	Item	Quantity	Rate	Amount
Adah Talbot	08-SEP-01	Work	1	\$1.00	\$
	15-MAR-01	Boot Between Horses	1	\$10.00	\$1
	18-MAR-01	Wagon	1	\$5.00	\$
	18-OCT-01	Boot Between Horses	1	\$10.00	\$1
	04-APR-01	Harness	1	\$2.00	\$
Total:					\$2
Andrew Dye	03-NOV-01	Pig	1	\$2.00	\$
	05-NOV-01	Hay	4	\$5.00	\$2

图1-2 Ledger Summary报表

这个报表和写在纸上的Ledger十分相似。它有页面脚注、标题和所有面向页面报告的装饰。

Oracle Developer提供许多不同种类的报表,而且单个报表可以组合成不同的类型。

tabular(表格):简单表。

mailing label(邮件标签):在每页的某个区域内的一系列有规律的重复的格式化记录。

form letter(格式信笺):一种占据了整个文本空白处的样板文字,它被安排在数据记录的周围,比如接受者的名字和地址。

master-master(主-主):两组显示在一起的没有关系的记录。

master-detail(主-从):一个主记录带有两个或多个有关的从属记录,显示在一处。

matrix or crosstab(矩阵或交叉):一个两列的十字表格,表示某些集合或者其他来自各列值的组合值。

datafile(数据文件):使用一个逗号分隔的或其他的各种分隔的数据文件,使用这种文件将数据传输到其他工具(通常有比较容易的办法)。

graph(图):包含图表或数据图表除此之外或是取代显示数据本身。

注意 Oracle Reports完全能独立使用。事实上,Oracle将Oracle Reports当作独立的产品和Oracle Report Server一起出售。

1.1.3 图形应用程序

图形是数据的形象化表示。术语 chart 指的也是这种表示。在它们所有的变化中,折线图、

饼图和条形图都是使数据形象化，而不仅仅是显示数据。Oracle Developer提供了一套高级工具，用来生成可以独立应用的图形应用程序和用来在表单及报表中包含图形。

图1-3是一个用饼图表示的图形应用程序。

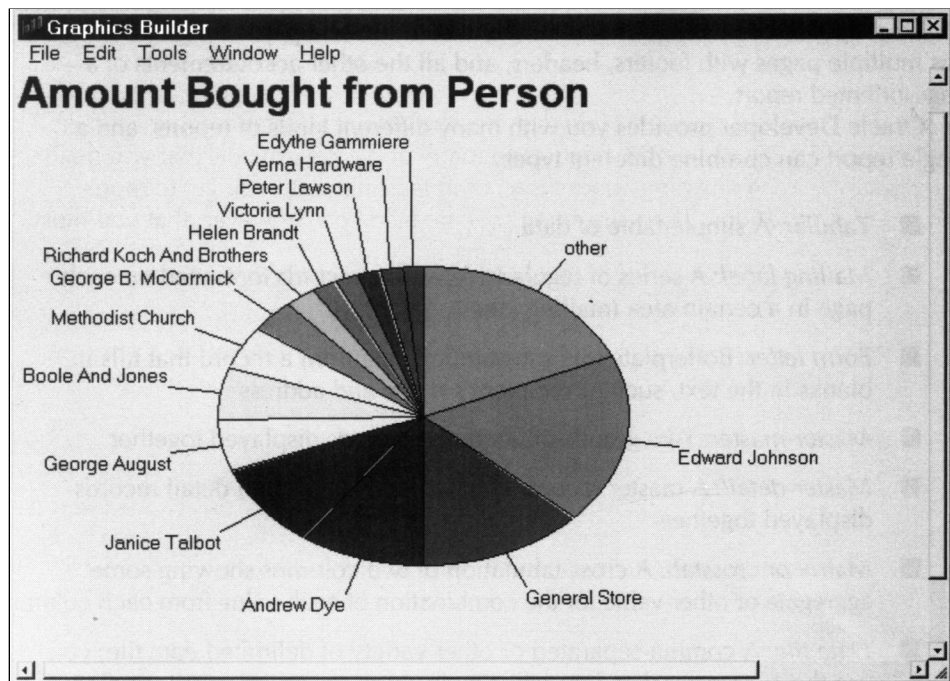


图1-3 饼图图形

甚至可以使图形被激活，这样用户就可以操纵它们。例如，用一个图形表示某种数据的总合计，比如推销员收入的合计。可以为用户产生这样一种能力，下钻 (drill down)查看按城市分解的每个推销员的细节图。当将图形嵌入表单应用程序时，这种能力特别强。有关下钻 (drill-down)图形的详细内容，请参见第8章。

1.2 数据库应用程序有什么不同

是什么使得数据库程序如此特殊？它与通常的应用程序有什么不同？

数据库管理引擎对每况愈下的情况提供数据保护 (恢复)，支持多用户 (并发)、多个应用程序和参照完整性，防止未授权使用数据 (安全性)，而且将用户和在特殊平台上管理数据的细节隔离开 (可移植性)。SQL和关系数据模型或对象关系数据模型提供了一个很强的、可靠的数据访问的范例。数据库管理引擎和所使用的数据特性和有效维护数据的其他工具紧密地连接和结合在一起。

Oracle Developer自动地处理数据管理的细节。事实上，Oracle Developer处理如此多的细节使得你可以不必学习与这些细节有关的任何东西，或者至少要在高级的应用程序中才需要学习。但是，为了有效地使用 Oracle Developer，必须了解其中的一些特殊的组件。

如果想继续研究 Oracle Internet Database Server (Oracle 的因特网数据库服务器) 在现场是如何工作的，请参看 Oracle 的文档资料或有关 Oracle 数据库管理和编程的书籍。比如 George Koch 和 Kevin Loney 著《Oracle 8 完全参考手册》，或 Kevin Loney 著《Oracle 8i 数据库管理员手

册》一书。

事务是工作的逻辑单元。事务从与数据有关的动作开始，用一个成功的终止命令 Commit (提交)或一个不成功的终止命令 rollback(回滚)结束。这些限制本质上建立了数据操作的顺序，如果需要做，可通过回滚数据库的变化来撤消操作。

Oracle Developer应用程序需要操心的是当事务结束时给数据库引擎发信号，通常用 SQL COMMIT WORK 语句或ROLLBACK WORK语句，这对应用程序设计有某些暗示。

必须做的第一个选择是，决定用户是使用提交还是回滚。用菜单、按钮或工具按钮进行选择，这是对最终用户在数据库工作进程上总的控制。他或她可以实现一个小的变化或者许多大的变化——凡是适当的都可以。

接下来要做的选择是决定用什么构成一个逻辑单元。即使是允许最终用户控制提交或回滚，应用程序仍然需要在适当的时刻提供这些选择。例如，可以在删除一行和删除依赖于这一行的各行后允许提交或回滚。如果允许终端用户在删除以后和在相关的删除之前提交，那么就违反了参照完整性(在稍后讨论)。要考虑在磁盘崩溃事件中数据库引擎应该恢复什么。如果要恢复的内容包含了多个操作，那么应将所有有关的操作都组装到一个事务中。

Oracle Developer好的一面是它完全了解所有这些问题，而且自动地处理它们。它自动地在应用程序逻辑上的适当时刻处理终端用户的提交和回滚请求，它在界面上自动产生提交和回滚命令，它提供了一种方法将每个事务支持的操作作为其应用程序结构的一部分包含在内。

参照完整性是数据库确认什么时候一个数据行引用另外一个已存在的数据行的能力。Oracle和其他的数据库管理系统通过表上的完整性约束或通过启动公共事件，比如插入、更新和删除的触发器来管理数据库的大多数参照完整性。在可能的情况下，应该将这种确认移到数据库服务器，以便所有的应用程序都共享它。保留应用程序的有效代码。

每个表(table)有或者应该有一个主关键字。主关键字是表中带有唯一标识的列或多列。关系数据库的本质是通过到另一个表的主关键字的数据引用来表示表之间的逻辑关系。引用的列是外关键字，而且它指的是外表的主关键字。大多数表连接与外关键字和主关键字的匹配有关。

你将会看到维护这些数据指针的完整性是如何重要。在关系数据库中，指针是数据，而不是那种可以被搞坏的物理指针。还有一个问题是保证当出现数据引用时，确实映射到某地址的主键值。对于某些关系还有一个相反的问题是要确认对每个主关键字，存在一个引用它的外关键字。最后必须确认主关键字能够唯一地标识表的行。

可移植性是指在不同的系统中运行的能力。它适用于操作系统，也适用于窗口系统、应用程序框架和任何其他服务平台，其中包括数据库管理平台。大多数应用程序必须以几种不同的方法处理可移植性。对 Oracle Developer应用程序的要求是能够在不同的数据库管理系统或在不同的操作系统运行同一个应用程序。

数据库可移植性是指如何使应用程序与它的基础的数据库引擎无关。通过界面或界面组这种专门的方法访问数据库管理程序可能是个问题。在这里，必须要决定的是，应用程序对于多个数据库平台需要的运行范围，用户要求应用程序在 Oracle 8、Informix和Sybase中运行吗？如果要求，应该立即果断地限制在标准的 SQL上，可以说这已经是最低的标准了。

在这种情况下，ODBC可以有所帮助。但是它的主要作用是使得可以从市场上得到的不同数据库的接口标准化。ODBC(Open Data Base Connectivity standard，开放数据库互联标准)提

供了统一的SQL语言和编程接口，供那些愿意对它写一个驱动程序的数据库使用。对于各个有影响的数据库，都有厂家和第三方的 ODBC驱动程序(有关在Oracle Developer中使用ODBC接口的细节请参看第15章)。通过ODBC，可以在运行时转换数据库，而不需要重新编译代码。

操作系统(OS)的可移植性是工具和数据库管理程序本身的可移植性(原文如此，但这个说法是不对的。——译者注)。如果要在不同的平台上运行应用程序(例如，在Internet、Windows、Macintosh或者UNIX上运行)，则需要将应用程序转向不同的数据库管理程序，或者是数据库管理程序和工具必须能够用同样的接口在不同的操作系统上运行。当 ODBC驱动器在不同的操作系统和各种数据库管理系统中得到利用时，ODBC在逐渐地发挥它的作用。

在使用Oracle Developer时并不一定需要了解SQL。当进入到产品的较高级的使用时，才会发现这些知识有用。如果还不懂得 SQL，那么应该进行训练，或者去读一本有关这个语言的书，比如Koch和Loney著《Oracle 8完全参考手册》，或者Groff和Weinberg著《SQL：The Complete Reference》(Osborne/McGraw-Hill,1999)。

最后，Oracle Developer允诺能够使高质量的可扩充的应用程序在三级环境中工作。两级的应用程序在工作站上运行并访问数据库服务器。三级应用程序主要运行在 Application Server上，由Application Server去访问数据库服务器，通过客户机工作站只管理用户界面的功能。使用Developer Server，可以为大量客户机及巨大的数据库建立健壮的、高性能的应用程序。

1.3 因特网数据库应用程序有什么不同

因特网在应用程序开发方面所取得的进展是令人惊奇的。只在短短几年的时间里，因特网就完全改变了我们关于应用程序的想法。和其他的数据库应用程序一样，因特网计算要求应用程序开发环境比以前所要求得更多。因特网数据库应用程序必须是：

可靠的。

安全的。

高度可用的。

可扩展的。

可靠是很重要的。因为许多公司现在依赖因特网应用程序来运行它们的业务。一个公司要使其产品或服务可以从因特网上得到，必须保证传递这些产品和服务的应用程序工作可靠。在这样的竞争环境中，发生错误会使公司丢失信誉和利益。

许多因特网应用程序处理钱财、私人信息或商业秘密。这种应用程序要求十分安全，以保护具有价值的数据和系统用户秘密。

电子商务的新领域对在因特网上使用的数据库有很严格的要求。这些应用程序必须 24小时可用。你可以睡觉，但是你的客户，不管在世界的什么地方，可能并不睡觉。所以数据库应用程序也不能睡觉。

最后要说的是，最近几年很少向我们显现出的是对扩展性的要求。没有人能够想象得到因特网的急剧增长。当公司将它们的产品和服务上网时，经常发现他们低估了对访问的要求。如果因特网数据库应用程序不能处理用户数量的增长，那么很快就会处于困境之中。

这些要求就像是公司内部使用的企业范围的应用程序要当作企业外部的世界范围的应用程序来使用一样重要。Oracle Developer为这些应用程序提供了很强的解决办法。这些应用程

序对开发工具的要求很高：网络可能对数据流形成瓶颈，服务器提供的性能可能很差，软件的层次和扩展性的要求可能超过了开发者的能力。

Oracle Developer Server是一个专门用于和优化 Oracle Developer应用程序的应用程序服务器。它允许将新的和现行的应用程序应用到因特网上，而不需要修改应用程序的代码。

一个好消息是 Oracle Developer Server可以自动在网络上进行调整和执行，并且自动递送本机服务和内部函数性以确保可靠性、安全性、可用性和可扩展性。

Oracle Developer提供了一个创建可扩展的、执行关键任务的数据库应用程序所必须的集成开发环境。Oracle Developer Server为把Oracle应用程序应用到因特网上服务提供了一个专门的优化的 Application Server。本书将用几部分内容指导你如何使用 Oracle Developer和 Oracle Developer Server，教程的重点将放在产品的基本特性上，还将对产品的细节提供综合参考。将个别指导的内容和参考资料结合起来，就是一个利用 Oracle Developer建立高性能数据库应用程序的完全手册。

开发过程讲述到此为止。第2章讲述在 Oracle Developer应用程序中创建对象的细节。