

《ORACLE关系数据库系统使用说明》之八

ORACLE SQL*FORMS 设计教程

国家统计局计算中心



《ORACLE关系数据库系统使用说明》之八

ORACLE

SQL*FORMS 设计教程

国家统计局计算中心

前 言

ORACLE RDBMS是美国Oracle公司的关系型数据库管理系统产品，它采用SQL(美国国家标准学会——ANSI于1986年10月16日正式颁布SQL语言为美国国家标准)作为数据库查询语言，拥有60多种大中型机、小型机、微机上的运行版本，包含有报表生成、全屏幕表格、高级语言接口等多种实用工具软件产品，是一种较好的高性能关系型数据库管理系统，其1986年8月宣布排出的版本5.1产品又具有了初步的分布式关系数据库管理系统的先进功能。

本套《ORACLE关系数据库系统使用说明》资料系我们对绝大多数ORACLE RDBMS产品说明书进行编译后形成，它包括了ORACLE产品的以下内容：

1. ORACLE RDBMS Kernel

- Basic Utilities
- SQL*Report
- SQL*Loader

2. ORACLE SQL*Plus

3. ORACLE SQL*Forms

4. ORACLE SQL*Calc

5. ORACLE Pro*C

6. ORACLE pro*Fortran

7. MS-DOS的ORACLE安装与用户指南

8. ORACLE Report*Writer

9. ORACLE SQL*Net

本套资料由国家统计局计算中心组织编译，参加工作的有：

吕春莲、王延华、孔祥清、胡杉、王珙、谢坤等同志。

本套资料供国家统计局系统内部使用。

本册为ORACLE SQL*FORMS设计教程。

目 录

第一部分 SQL*FORMS 介 绍

1.0 引言	(4)
2.0 产品表	(5)
2.1 产生一个缺省表: 用少量击键次数开发一个产品管理全功能表.....	(5)
2.2 运行一个表 利用SQL*FORMS在你的产品表上自动建立的功能	(8)
2.3 改变表的功能 方便地按规格改制你的产品表.....	(13)
3.0 顾客表	(19)
3.1 屏幕设置 使顾客表成为你所希望的布局.....	(19)
4.0 订货记录表:	(24)
4.1 掌握SQL*FORMS的功能 建立复杂的订货记录表.....	(24)
4.2 执行总结	(32)

第二部分 启动SQL*Forms

SQL*Forms概论

0.0 本章概述	(35)
0.1 什么是SQL*Forms.....	(35)
0.1.1 快速、简单的数据录入、修改和查询	(35)
0.1.2 保护数据库	(36)
0.1.3 显示计算的值	(37)
0.1.4 生成历史记录	(37)
0.2 理解Forms的基本概念	(37)
0.3 在本教程中你做些什么	(41)
0.3.1 方 案.....	(41)
0.3.2 目 标.....	(45)
0.3.3 步 骤.....	(45)
0.4 在你开始学习之前	(50)
0.4.1 功能键的配置	(50)
0.4.2 样本Forms	(51)
第一章 建立Forms.....	(53)

1.0 本章概述.....	(53)
1.1 启动SQL*Forms.....	(53)
1.1.1 CHOOSE FORM 窗口.....	(54)
1.1.2 建立一个Form.....	(55)
1.2 建立块	(56)
1.2.1 建立一个单记录块	(57)
1.2.2 SELECT COLUMN 窗口	(59)
1.2.3 建立一个多记录块	(60)
1.2.4 建立第三个块	(62)
1.3 存贮, 运行, 和测试Forms	(65)
1.3.1 存贮Forms	(65)
1.3.2 生成和运行Forms	(66)
1.3.3 测试Forms	(67)
1.4 通过Forms窗口观看数据库	(71)
1.5 本章总结: 你学到了什么?	(71)
练习清单	
练习 1-1 启动SQL*Forms.....	(53)
练习 1-2 建立一个模式并报告错误信息	(55)
练习 1-3 建立ORDERS块.....	(57)
练习 1-4 建立LINEITEMS块	(60)
练习 1-5 建立CLIENF_DATA块	(63)
练习 1-6 存贮Form.....	(65)
练习 1-7 生成和运行Form.....	(67)
练习 1-8 测试Form.....	(69)
第二章 改进Forms.....	(74)
2.0 本章概述	(74)
2.1 改进ORDERS块.....	(74)
2.1.1 加入一个新的域	(74)
2.1.2 DEFINE FIELD窗口	(76)
2.1.3 定义一个域的属性	(77)
2.1.4 SPECIFY ATTRIBUTES窗口.....	(79)
2.1.5 完成ORDERS块.....	(80)
2.2 修正LINEITEMS块	(81)
2.2.1 重新设计已存在的域	(81)
2.2.2 向右对齐的域的屏面宽度和域宽	(87)
2.2.3 建立——新的域并对域重新排序	(88)
2.3 增强Forms的直观性	(89)
2.4 测试Forms	(91)

2.4.1 关于非显示域的一点注意	(92)
2.5 通过Forms窗口进行观察	(93)
2.6 本章总结: 你学到了什么	(94)
练习清单	
练习 2-1 在ORDERS窗口中加入一个新的域.....	(75)
练习 2-2 定义一个新的域并声明它的属性	(77)
练习 2-3 完成ORDER块	(80)
练习 2-4 切断、粘接, 取消、重新定义宽度, 重新标注和重新定义已存在 的域.....	(82)
练习 2-5 建立新的域并对序重新排序	(89)
练习 2-6 画方框并加入正文	(90)
练习 2-7 测试模式	(91)
第三章 Forms的再加工.....	(96)
3.0 本章概述	(96)
3.1 修改域的定义	(96)
3.1.1 修改CLIENT_DATA块.....	(96)
3.1.2 SPECIFY VALIDATION窗口	(99)
3.1.3 修改ORDERS块.....	(101)
3.1.4 协调两个块	(104)
3.2 修改一个块	(106)
3.2.1 DEFINE BLOCK窗口.....	(108)
3.2.2 SPECIFY DEFAULT OKDERING 窗口.....	(109)
3.3 测试Forms	(109)
3.4 准备进行第二部分	(113)
3.4.1 本章总结: 你学到了什么	(113)
练习清单:	
练习 3-1 设置一个缺省值和一个值域	(96)
练习 3-2 生成一个缺省日期和一张域值清单	(101)
练习 3-3 把详细记录链接到一个主记录	(104)
练习 3-4 修改块的属性	(106)
练习 3-5 测试模式	(111)

第三部分 触发器学习

第四章 触发器介绍.....	(117)
4.0 触发器介绍	(117)
4.1 触发器入门	(117)
4.2 数据录入的合法性检查	(118)
4.2.1 触发器的目标	(118)
4.2.2 建立一个强功能触发器的简单SQL语句	(120)

4.2.3	域引用的特殊标志	(122)
4.2.4	系列触发器	(122)
4.2.5	SQL语句解析	(123)
4.2.6	CHOOSE TRIGGER窗口	(123)
4.2.7	TRIGGER STEP窗口	(124)
4.3	显示相关数据	(126)
4.3.1	触发器的目标	(126)
4.3.2	关于SELECT ZNTO 语句	(128)
4.3.3	关于临时域	(128)
4.4	测试你前面定义的触发器	(128)
4.5	生成序列号码	(130)
4.5.1	触发器的目标	(130)
4.5.2	写一个两步触发器	(132)
4.5.3	修正ORDERID域的属性	(135)
4.5.4	两步触发器序列	(135)
4.5.5	为什么一个单步触发器不能工作	(136)
4.6	测试块触发器	(136)
4.7	本章总结: 你学到了什么	(137)
	练习清单	
练习 4-1	定义一个域的触发器	(118)
练习 4-2	定义用户ID域的触发器	(126)
练习 4-3	测试用户ID域的触发器	(128)
练习 4-4	测试产品ID域的触发器	(129)
练习 4-5	定义一个块触发器	(130)
练习 4-6	测试块触发器	(136)
第五章	复杂SQL语句的应用	(139)
5.0	本章概述	(139)
5.1	改进产品序号触发器	(139)
5.1.1	触发器目标	(139)
5.1.2	学习双表查询	(142)
5.1.3	显示当前价格	(142)
5.2	校验定价信息	(144)
5.2.1	触发器目标	(144)
5.2.2	DUAL表的使用	(145)
5.3	测试定价触发器	(146)
5.4	为一个域限定常数值	(148)
5.4.1	触发器目标	(148)
5.4.2	大写属性	(149)

5.4.3 对DUAL表的一个补习	(150)
5.4.4 测试回扣方案触发器	(150)
5.4.5 遇到一个SQL句法错误	(150)
5.5 限制访问Form的时间	(152)
5.5.1 触发器目标	(153)
5.5.2 转换触发器的系统日期	(155)
5.6 测试Form触发器	(156)
5.7 本章总结	(158)

练习清单

练习 5-1 通过显示相关数据校验数据录入的合法性	(140)
练习 5-2 定义实际价格触发器	(144)
练习 5-3 测试定价触发器	(146)
练习 5-4 定主委员会计划触发器	(148)
练习 5-5 测试委员会计划触发器	(150)
练习 5-6 定义一个模式触发器	(150)
练习 5-7 测试模式触发器	(153)
练习 5-8 测试触发器	(156)

Oracle公司在为公司和政府部门开发关系数据库中，无论是在市场占有率还是在技术方面都处于领先地位。该公司创立于1977年并于1979年推出其第一代产品：ORACLE关系数据库管理系统。ORACLE的第一代产品是建立在IBM公司的SQL数据库语言基础上的，今天，ORACLE仍有效地保持提供其最有功效的关系数据库。

Oracle公司还给用户提供成套的系列应用开发产品，包括全屏幕表格、菜单、报表，图形以及数据库设计。这些工具能帮助你有效地开发应用和进行维护工作。

SQL*FORMS是Oracle应用开发系列产品中的全屏幕表格产品。

下面我们对该产品作一简单介绍。

一、无须程序语言就可以开发应用

SQL*Forms可使你在输入、查询、更新和删除数据方面迅速地进行基于表格的开发应用。你可使用简单的菜单和高效的屏幕画面——而不必书写程序语言来说明您的应用需要。然后，SQL*Forms就能把你的要求与从ORACLE的数据辞典中得到的信息结合起来，以产生你的应用。

二、有效的原型建立方式

SQL*Forms独特的非过程处理使你能有效地建立一个应用原型。建立原型使你和你用户能够进一步完善你们构造的应用。当你想改善设计时，只要指明这些改变，SQL*FORMS就会自动地接收这些改变。使用SQL*Forms，可使你把精力集中在改善对应用的设计上，而避免冗长的程序语言。只要你的设计工作恰当，就可完成你的应用，这是因为你的原型就是你的应用。

三、无限的功能

SQL*Forms解决了许多其他第4代产品感到棘手的复杂问题。借助于在SQL*Forms中建立一个自动产生逻辑基础的功能，你可容易地开发一个常规的应用问题，这样做比用程序语言设计要快得多。

四、开放型结构

SQL*Forms为一开放型体系结构，以保证你可以将它和以COBOL、C和多数其它程序语言书写的已有程序相连接。因此，你总是能够以你所期望的任何方式来处理信息。

五、改变要求的灵活性

使用SQL*Forms，你可容易地对变化中的事务需求作出反应。调整SQL*Forms的应用只需数分钟而不是数月时间。SQL*Forms的非过程结构和ORACLE关系数据库管理系统的灵活性确保你成功。

本部分通过一个例子向你着重说明SQL*FORMS的功能。该例为建立一个订货采购的应用，含有订货、采购零部件、维持百货、顾客情况及订购信息等内容。关于SQL*FORMS的详细技术细节请查看《SQL*FORMS设计参考手册及操作指南》，或与你的ORACLE代表联系。

1. 引言

本节为SQL*Forms的引言部分，它通过从头至尾完整地构造一个实际应用模型，表明SQL*Forms是如何解决你的应用开发要求的。通过此例，你将看到SQL*Forms能使你有效地工作。

1) SQL*Forms通过自动建立的全功能表，可为你做完大部分的设计开发工作。

2) 使用SQL*Forms的简明屏幕菜单和屏幕画面，可调整该表的功能和图形表达。

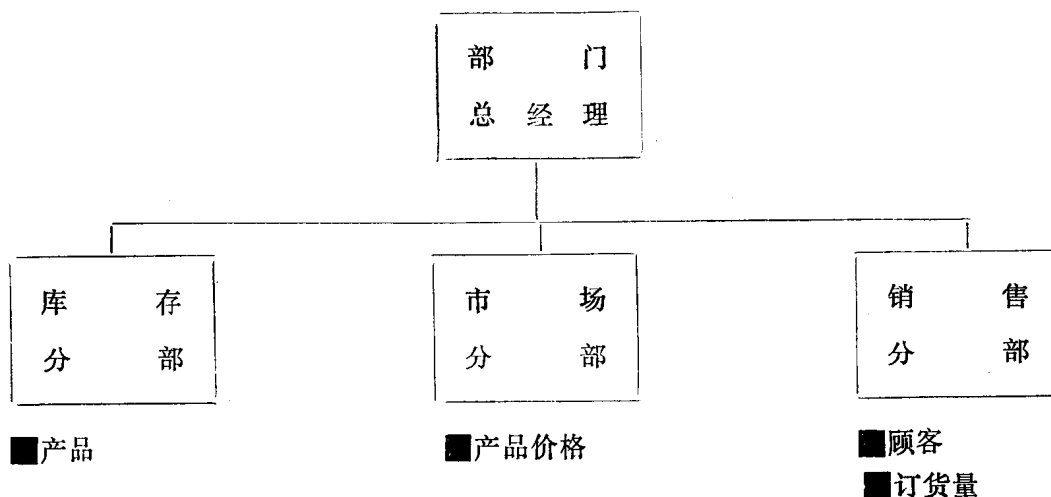
3) 若需要按规格改制某个表，则可利用SQL*Forms的开放型结构，将增加的性能引入到按缺省值建立的基础表中。由于SQL*Forms注意保持这个基础表，故你需确定新增加部分的格式，你的工作量是很小的，甚至对最复杂的应用课题，你也可以迅速完成。

这种先进的开发过程帮助你有效进行工作。开发过程任何步骤上都容易进行更改，使你能迅速地按照用户的建议作出反应。

1.1 应用实例

在本部分中，假设你就是一个应用设计者，为一体育用品制造商工作。他要求你为该体育用品制造公司的某个新部门开发一个订货记录系统。

该部门的几个分部将共同使用这个系统。库存分部将记录和维护有关产品的信息。市场分部要评价和修正价格政策。销售分部要记录顾客信息和实际订货量，如下图所示：



由于各分部要同时使用此订货记录系统，因此，该系统必须同时容纳多个用户，还应该不会引起冲突，也不会造成过分的延迟。

订货记录系统的数据库应包括4个主要的表格。

首先，你应采访销售、库存、市场等分部的经理，以便确定订货记录系统的初步要求，在分析了他们的要求后，就可定义数据库的基本结构。如下表示：

顾 客 (CUSTOMER)

顾 客 编 号	姓 名	住 址	城 市	国 家	邮 政 编 码	区 域	电 话	信 贷 限 制
------------	-----	-----	-----	-----	------------	-----	-----	------------

订 购 (主要信息) (ORDER)

订 购 编 号	顾 客 编 号	订 购 总 数	销 售 号	订 购 日 期	奖 金 计 划	事 态
------------	------------	------------	----------	------------	------------	-----

存 货 项 目 LINEITEM

订 购 编 号	项 目 编 号	产 品 编 号	价 格	数 量	项 目 总 数
------------	------------	------------	-----	-----	------------

产 品 件 PAPT

产 品 编 号	产 品 名	价 格	有 效
		说 明	日 期

你的任务是开发三种表格，这三种表格是分别用于记录、维护、查询此数据库的，如下表所示：

Form 表格	被操作的数据库表	主要用户
产 品	PART	库存和市场分部
顾 客	CUSTOMER	销 售 分 部
订 货 记 录	ORDER.LINEITEM	销 售 分 部

下几节将分别介绍如何建立各种表格的过程。

2. 产品表

2.1 产生一个缺省表

可以用少量击键次数开发一个产品管理全功能表。

构造订货记录系统的第一步是为产品开发一个全功能管理表格。库存分部经理的要求是：

每一产品都应有产品号、产品名、价格以及此价格有效的日期。经理需要使库存部门的人员迅速记录信息。

你可以按照库存部门经理的要求迅速建立一个初步的原始模型。这只需要使用一个简单的SQL语句定义一产品表如下：

```
CREATE TABLE Part
(Part—num      NUMBER,
description     CHAR (25) ,
list—Price      NUMBER,
effective—date  DATE) ;
```

PART

Part Number	Description	List price	Effective Date
9999999	AAAAAAAAAAAAAAAAAAAA	999.99	DD—MON—YY

当产生该表时，ORACLE就将表的定义送入数据辞典。SQL *Forms 就可利用这些信息来帮助你建立你的应用。

定义了PART表后，就可使用SQL*Forms的pop—up菜单，用少量击键次数产生一个全功能全屏幕表格。

SQL*Forms的第一个菜单可供你确定FORMS表格名称，这个表格即是你 要用来工作的表格。若你欲称你的表格为Partform，则只须在键盘上键入 Partform，然后再选择CREATE操作，如下图所示：

步 1：键入

表格名称

CHOOSE FORM

Name

PARTFORM

Actions:

CREATE

MODIFY

LIST

RUN

DEFINE

LOAD

FILE

GENERATE

SQL*Forms可以使你在全屏幕表格上以你所期望的多个表格进行工作。所以，下面的菜单要求你说明你现在所要求的表格名称。选择表的名称：PART，并选择DEFAULT操作，使SQL*FORMS自动定义这个表。

步 二：
键入一个
表格名称

CHOOSE FORM

Name
PARTFORM

CHOOSE BLOCK

Name
→PART

Page Number 1

Actions:

CREATE	MODIFY	DROP
LIST	FIELDS	DEFAULT
PREVIOUS	NEXT	

产生你的SQL*FORMS表格的最后一步是，告知SQL*Forms你在屏幕上欲同时显示几行零部件数据。例如规定为7行，则在下一光标带键入数字“7”，如下图所示：

步 三
选择您欲一次显示的数
据行数。

CHOOSE FORM

Name
PARTFORM

CHOOSE BLOCK

Name
PART

Page Number 1

DEFAULT BLOCK

Table Name;
PART

Rows Displayed: 7
Base Line: 1

Actions:
COLUMNS TABLES

小 结

通过以下三个步骤，你已经给定义了你的产品全屏幕表格：

- 1) 键入全屏幕表格名称，以后我们称之为Form名称，以区别于数据库中的表。
- 2) 键入表的名称，此表为数据库中的表。
- 3) 选择即将显示的数据行数。

到此为止，你仅耗时 1 分钟就产生了一个零部件Form表格。

由于SQL*Forms可从ORACLE数据辞典中得到关于PART表中所有所需的信息，并为你产生了相应的Form表格，因此，你不必逐个详细说明每个域的信息。SQL*Forms自动地设计出具有合适的列和标题的Form表格，并具有你所规定的欲显示的行数。

2.2 运行Form——应用SQL*Forms在你的产品表上自动建起的功能

你刚刚产生的产品Form不仅为某个用户，而且也多个用户同时提供了完整的数据记录、查询、更新和删除手段。下面通过库存分部的经理大威·马歇尔运行新零部件的应用例子，你就可看到SQL*Forms的运行速度和运行效率。

全保密性：你所看到的第一个屏幕是口令屏幕。SQL*Forms提供保密性，即所有表格的用户都必须具有有效的ORACLE用户标识和口令，如下图所示。SQL*Forms只允许用户使用在指定数据上的所有功能，或只允许使用经过选择的记录、查询、更新或删除功能。

SQL*Forms: Version 2.0 Fri Sep 26 15: 04: 14 1986	
Copyright(c)1986, Oracle (orporation, California USA. Att rights reserved	
Username	MARSHALL
Password	

方便的数据记录方式：一旦您已经键入了有效的用户标识符和口令，就可通过将数据键入该表格来记录这些数据，如下所示

PART			
PART—NUM	DESCRIPTION	LIST_PRICE	EFFECTIVE
100860	ACE TENNIS RACKETI	50	10-MAY-86
100861	ACE TENNIS RACKETII		

一次可记入几个记录。您可在行与行之间用箭头或功能键前后移动，也可在行与行之间上下移动。可使用您所熟悉的箭键来迅速更正错误。若正记录的行数多于屏幕容纳的行数，则你可上下卷动屏幕以便观察其余的行数。

SQL*Forms在数据记入时可自动地检查输入工作。它能查阅ORACLE数据字典以便识别哪个字段是字符型、数字型以及日期型，因此，它能自动地对所有输入数据加强适当的限制。

例如，若你试图往数字型字段记入一个字母，则SQL*Forms将不允许你这样做，并立即用某个有益的信息来提示你，如下图示：

PART			
PART_NUM	DESCRIPTION	LIST_PRICE	EFFECTIVE
100860	ACE TENNIS RACKET I	50	10-MAY-86
100861	ACE TENNIS RACKET II	45K	

Numeric Field—Valid Characters are 0-9

上图表明，SQL*Forms确保你只将数字值输入数字型域。

又如，你试图输入日期型数据，SQL*Forms则将确认你记入的日期是否有效。这是由于它置入了一个日历，该日历能检查从公元前4712年至公元4712年各年甚至包括闰年的日期。下图表明，SQL*Forms不会使您输入无效的日期：

PART			
PART_NUM	DESCRIPTION	LIST_PRICE	EFFECTIVE
100860	ACE TENNIS RACKETI	50	10-MAY-86
100861	ACE TENNIS RACKETII	45	31-FEB-86

INVALID DATE

↑

此信息使您不会输入无效日期