

计算机数据库应用技术系列丛书

ORACLE[®] DATA PROCESSING A MANAGER'S HANDBOOK

ORACLE 数据处理管理员手册



- ▶ Oracle CASE development
- ▶ Project management techniques
- ▶ Downsizing strategies

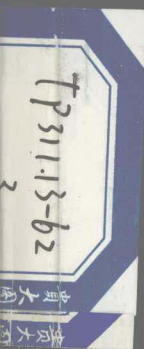
Graham H. Seibert

McGRAW-HILL
学苑出版社



ORACLE DATA PROCESSING
MANAGER'S HANDBOOK

Seibert



学苑出版社

TP 311.13-62

卅 3

计算机数据库应用技术系列丛书

Oracle Data Processing A Manager's Handbook

ORACLE 数据处理管理员手册

学苑出版社

(京)新登字 151 号

内 容 提 要

本书全面地讲述了开发模型、Oracle DP 经理工作、Oracle 环境、Oracle 开发人员运用 CASE 工具、开发数据处理策略、公司层 DP、职员计划、Oracle 专业的人员管理、多面手角色、专家角色、培训 Oracle 专业人员、安全性和完整性、质量控制、文档需求和标准、过程、来源、项目管理、估计、经济分析和成本核算等方面的内容。本书内容新颖,结构严谨,可读性强,可作为广大数据库开发者的参考丛书及各大专院校的教材。

欲购本书的用户,请直接与北京海淀 8721 信箱书刊部联系,邮政编码 100080,电话 2562329。

版 权 声 明

本书英文版名为《Oracle Data Processing A Manager's Handbook》,由 McGraw-Hill 公司出版,版权归 McGraw-Hill 公司所有。本书中文版由 McGraw-Hill 公司授权出版。未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

计算机数据库应用技术系列丛书

ORACLE 数据处理管理员手册

著 者:Graham H. Seibert

译 者:何庆红 姜 寰

审 校:燕卫华 王 真

责任编辑:甄国宪

出版发行:学苑出版社 邮政编码:100036

社 址:北京市海淀区万寿路西街 11 号

印 刷:兰空印刷厂

开 本:787×1092 1/16

印 张:12.375 字 数:277 千字

印 数:1~5000 册

版 次:1994 年 10 月北京第 1 版第 1 次

ISBN7-5077-0874-8/TP·23

本册定价:19.00 元

学苑版图书印、装错误可随时退换

致 谢

在过去的五年中,我曾经与许多优秀的客户打交道,从他们那里我学会了 *Oracle* 并丰富了我的管理知识。我要感谢 *John Harned*, *Sue Manion*, *Rich Weiskopf*, *Greg Druffel*, 以及 *Computer Sciences* 的 *Andrew McWhinnie*; *HFSI* 的 *Cheryl Smith*; *Service Employees International Union* 的 *Bill Mountjoy*; *Central Pension Fund of the Operating Engineers* 的 *Jack Johnson*, *Frank Gould*, *Jenny Brake* 和 *Moirra Dolan*; *CDSI* 的 *Steve Lyman*, *Hillary James*, *Prakash Sanghvi*; *Army Corps of Engineers* 的 *Bob Evans*。

我得到了 *Mid Atlantic Oracle User's Group* 的鼓励和支持以及技术专家的帮助。*Kathy Christian* 鼓励我写了第一本书《*Oracle* 开发工具的使用》。另外, *Warren Capps* 和 *Dale Lowery* (他们推举我为用户协会的主席) 以及 *Gloria Miller*, *John Duncan*, *Mike Venerable*, *Matt Comstock* 也都帮助了这一项目。*Oracle* 的顾问 *Ulka Rogers*, *Charles Peck*, *Dewey Allen* 和 *Helene Thoreson* 进行了认真的校阅并提出了宝贵意见。

在本书写作过程中,我广泛地收集了许多客户的经验,我要感谢那些接受过拜访的人: *ARINC* 的 *Mike Prince*, *Connie Lee Insurance* 的 *Bob Bain*, *GHA* 的 *Mike Armer* 和 *Tim Donohoe*, *John J. McMullen* 的 *Jim Weingart*, *U. S. Navy* 的 *Kent Smith*, *Parklawn Computer Center* 的 *Tom Reddin*, *EDS* 的 *Mary Serafin*, *Blue Cross/Blue Shield of Tennessee* 的 *Frank Schlier*, *World Bank Financial Operations Division* 的 *Ulka Desai* 和 *Mukund Agashe*。

Oracle 公司对本书的写作一直给予了全力支持。我要特别感谢 *CASE* 市场部主任 *Renee Taylor*, *Oracle CASE* 的第一副总经理 *Richard Barker*, *Oracle CASE* 开发部的副总经理 *Alan Gater*, *Oracle* 顾问部的 *Mike Burkett*。

我还要感谢那些软件商,他们的产品被用作示范: *LogicWorks* 的 *Ben Cohen* 和 *Barbara Bogart* 拷贝了 *ERwin*, *Corel Systems* 的 *Paul Bodnoff* 拷贝了极棒的 *Corel DRAW!* 用作画图说明, *Symantec* 拷贝了 *Timeline* 用作说明“项目管理”一章,还有 *Business Objects* 的 *Bill Sautter*。

目 录

第 0 章 简介	1
0.1 本书目的	1
0.2 方法论	1
0.3 来源	2
0.4 成员	2
0.5 读者	2
0.6 本书必要性	3
第一章 开发模型	4
1.1 瀑布开发模型与演示	4
1.2 Oracle 系统剖析	6
1.3 处理成熟性	7
1.4 Oracle 历史回顾	9
1.5 总结	10
第二章 Oracle DP 经理工作	12
2.1 技术领导和策略规划	12
2.1.1 了解公司现状	12
2.1.2 了解企业动产系统历史	13
2.1.3 参与企业管理	13
2.1.4 在策略计划中包括信息管理	14
2.1.5 维护企业的数据模式	16
2.2 建立大范围信息系统计划	17
2.2.1 制定计划和时间表	17
2.2.2 技术引进计划	17
2.2.3 资产取舍	17
2.3 管理项目的管理员	18
2.4 管理软件处理	18
2.5 质量	20
2.6 组织	20
2.7 行政和人事管理	21
2.8 总结	21
第三章 Oracle 环境	23
3.1 Oracle 工作站的操作模型	23
3.1.1 Oracle 工作室	23
3.1.2 Oracle 应用系统工作室	23

3.1.3	服务局环境	23
3.1.4	独立的 PC 用户	24
3.1.5	Oracle 软件屋	24
3.2	不同 Oracle 环境的管理原则	24
3.3	Oracle 克服了各种环境的不利条件	25
3.3.1	适用应用软件的标准	25
3.3.2	秘密获得	26
第四章	Oracle 开发人员运用 CASE 工具	27
4.1	CASE 组成	27
4.2	系统集成的 CASE 含义	30
4.3	用 CASE 改变生命周期法文档过程	32
4.4	CASE 对 Oracle 的影响	33
4.5	CASE 对数据处理的影响	34
4.6	当今开业者计划	35
4.7	总结	37
第五章	开发数据处理策略	38
5.1	企业策略的 MIS 支持	39
5.2	信息系统策略	40
5.3	总结	41
第六章	公司层 DP	43
6.1	部门之间工作共享	43
6.2	MIS 是成本中心还是利润中心	44
6.3	MIS 预算结构化	45
6.4	MIS 功能测量	46
6.5	总结	46
第七章	职员计划	47
7.1	职员来源	47
7.2	核心职员	48
7.3	项目、最大负载和专家需求	49
7.4	就业的长期趋势	50
7.5	制定长期职员计划	51
7.6	总结	51
第八章	Oracle 专业的人员管理	52
8.1	周转	52
8.2	数据处理中的职业路径	53
8.2.1	部门以外	54
8.3	生产力	54
8.4	成功的性能评价的特性	55
8.4.1	了解过程	56

8.4.2	少而可行的目标	56
8.4.3	提供快速反馈	56
8.4.4	两级管理	56
8.4.5	雇员接受评价	57
8.4.6	合理的报酬	57
8.5	评价和评职机制	57
8.5.1	评职机制的正常化	58
8.5.2	评职机制中的评职因素	59
8.5.3	评职技巧	60
8.5.4	不评职系统	60
8.6	工资和职务管理	60
8.7	与不良行为者打交道	62
8.8	下一步计划	63
8.9	雇佣	63
8.10	总结	64
第九章	多面手角色	65
9.1	编程	66
9.2	详细设计	67
9.3	概要设计	68
9.4	数据设计	69
9.5	测试设计	70
9.6	测试数据开发人员	70
9.7	项目实验员	70
9.8	项目经理	71
9.9	一线经理	72
9.10	分析员	72
9.11	店员	73
9.12	配置管理员	74
9.13	用户支持	74
9.14	职业发展	74
9.15	总结	75
第十章	专家角色	76
10.1	系统程序员	76
10.2	系统管理员	78
10.3	专家程序员	79
10.4	数据库管理员	80
10.5	数据管理员	80
10.6	网络协调员	80
10.7	PC 专家	81

10.8	硬件配置	81
10.9	计算机操作	81
10.10	质量控制	82
10.11	配置管理	82
10.12	图书管理员	82
10.13	软件处理协调员	83
10.14	CASE 专家	84
10.15	总结	84
第十一章	培训 Oracle 专业人员	85
11.1	专业培训需求	85
11.2	培训内容	86
11.3	培训计划	88
11.4	深造	88
11.5	培训和回报	89
11.6	总结	89
第十二章	安全性和完整性	90
12.1	存档	91
12.2	备份	92
12.2.1	备份文件数据的频率	92
12.2.2	Oracle 备份	93
12.2.3	备份版本	94
12.2.4	存放地点	94
12.3	安全性	95
12.4	总结	97
第十三章	质量控制	98
13.1	质量控制目标	99
13.2	Oracle 系统中的质量因素	99
13.3	审查质量	99
13.3.1	分析中的质量	101
13.3.2	设计中的质量	101
13.3.3	测试中的质量	101
13.3.4	编程中的质量	101
13.3.5	造型和采购中的质量	102
13.4	总结	102
第十四章	文档需求和标准	103
14.0.1	适用 Oracle 环境的标准	103
14.0.2	执行标准	103
14.1	文档的目的	104
14.2	策略计划文档	105

14.3	分析文档	106
14.3.1	信息收集计划	106
14.3.2	会谈指南	107
14.3.3	会谈注释	107
14.3.4	属性定义	107
14.3.5	实体定义	108
14.3.6	ER 图	109
14.3.7	过程描述	111
14.3.8	数据管理方面的考虑	113
14.3.9	外部界面	113
14.3.10	软件包分析	114
14.4	设计文档	115
14.4.1	数据设计	116
14.4.2	数据执行计划	116
14.4.3	菜单结构和全局参数	117
14.4.4	处理规格说明书	117
14.4.5	测试计划	120
14.4.6	成本核算	121
14.5	开发和维护文档	122
14.5.1	源代码、注释和维护文档	122
14.5.2	用户文档	122
14.6	命名规则	123
14.6.1	命名限定和扩展	123
14.6.2	唯一命名	124
14.6.3	选择易用名	124
14.6.4	Oracle 命名规则	124
14.7	Oracle 安装标准	125
14.7.1	重要用法	125
14.7.2	界面标准	125
14.8	CASE 开发环境文档	125
14.9	总结	126
第十五章	过程	127
15.1	策略计划	128
15.1.1	等待策略计划的基础	130
15.1.2	从会谈开始策略计划	131
15.1.3	更新的业务模型	131
15.1.4	建立流程图	131
15.1.5	建立分解结构	131
15.1.6	MIS 策略计划处理	131

15.2	分析过程	132
15.2.1	定义范围	133
15.2.2	合并分析产品	133
15.2.3	质量控制	134
15.3	设计过程	134
15.3.1	行为顺序	134
15.3.2	应用系统标准	134
15.3.3	质量控制	135
15.4	编程过程	135
15.4.1	开发代码规则	135
15.4.2	修改计划文档	136
15.5	测试过程	136
15.5.1	测试数据	136
15.5.2	测试库	137
15.6	接受和初始化开发工作	138
15.7	策略计划和分析的会谈过程	138
15.8	软件包的挑选	140
15.9	项目管理过程	141
15.10	配置管理(CM)过程	141
15.11	其他过程	142
15.12	总结	142
第十六章	来源	143
16.1	销售渠道	143
16.2	发现产品	143
16.3	Oracle 资料目录	145
16.4	资料服务	146
16.4.1	寻找服务提供方	147
16.4.2	评价服务提供方	147
16.5	选择过程	147
16.5.1	小量购买中的单项评估和挑选	148
16.5.2	中量购买中的非正式评估	148
16.5.3	中型到大型购买中的 RFP 处理	148
16.6	合同	149
16.6.1	软件包和服务合同	149
16.6.2	与独立顾问签定合同	151
16.7	总结	152
第十七章	项目管理	154
17.0	项目管理系统的元素	154
17.1	工作分解结构	155

17.1.1	结构工作分解结构图	155
17.1.2	工作分解结构图用途	156
17.2	使用电子表格进行项目管理	158
17.3	自动化项目管理系统	160
17.4	总结	165
第十八章	评估	166
18.1	评估的类型和使用	166
18.2	评估过程中的统计学	167
18.3	评估技术和方法	168
18.3.1	功能点分析	168
18.3.2	环境因子	170
18.4	不相等比较	171
18.5	Oracle 拇指规则	171
18.6	总结	171
第十九章	经济分析和成本核算	173
19.1	建模	173
19.2	经济分析中的时间因子	174
19.3	风险估算	174
19.4	成本和利润	176
19.5	构造报表金融模型	176
19.5.1	变量使用	176
19.5.2	报表的物理构成	179
19.5.3	单元技巧	179
19.6	总结	179

第 0 章 简 介

当本地的宗教人士比传教士做得好时,传教士将怎么做呢?他需要一些精神寻找。大多数的 Oracle 工作间已经突破了传统的商业处理观念,尤其在那些涉及到使用纸张的领域,并以难以置信的快速开发更好系统。Oracle 正领导着一场系统开发的革命。其先进方法的知识是如此神奇,以至于那些不喜欢系统文档的读者可避免使用纸张来记录新的处理过程。

0.1 本书目的

本书记录了 Oracle 开发者使用的技术。成功的实践结果产生进化论——几种现代思想的综合学说。其中最深奥的概念是质量和生产力受软件处理的限制。例如一个单位的每个人都理解并致力于某些处理过程。事实上,他们应该定义这些处理过程。最显而易见的结果是不用繁复的文档他们就可以工作。而管理方面的问题则是究竟多少是没有也行的?

传统编程方法在解决问题时是很费劲的。Henry Ford 的天才就在于他设计了适合于多种用途汽车。类似地,软件包满足了大多数企业的需要,而且比传统软件可靠性高。服务供应商知道某些应用软件和工具比用户希望的还要好,使用这些工具可减少成本和风险。数据处理的角色已转移给购物者、综合者和合同管理员。本书就为这些角色提供指南。

劳动力密集型产业导致了自动化的出现,在本行业最大的人工成本是程序员。CASE (计算机辅助软件工程)实现了用机器替代人工,但这种替代不是一个突然的过程。CASE 正驱动 ADP(自动数据处理)专家向面向人为工作(例如分析工作)和专家角色方向发展(CASE 专家、算法程序员、数据库管理员和数据管理员)。管理体制还必须使应用软件程序员在公司成为最精锐的部队。本手册还为雇佣、培训职员和职业计划提供要点。

0.2 方法论

作为计算机科学思想的前锋,Carnegie Mellon 的软件工程研究院采纳了全面质量管理学校的远见卓识,即公司中每个人在理解和致力于发展他使用的东西时才会生产出高质量的产品。让职员定义他的工作过程是完成上述理解过程的最佳方法。管理的基本问题就是既要让每个人在组织中有效地工作,又要相互依赖去共同生产产品。

要完成一个原型至少应知道在组织中进行哪些工作。系统基础——包括数据设计、商业规则和系统结构——这些必须在用户、分析员、设计者和开发者之间达成协议。系统不允许对基础的重大修改,因为重大的修改会导致基于该基础上程序的废除。在 Oracle 中组成系统的程序主要是表格和报表,它们是相互独立的。一旦基础确定,开发人员就可以按照达成的协议快速地完成符合用户要求的原型。

对成功原型而言最重要的规则是认识到即使原型已完成了,但工作还远未结束。原型表明当一件事工作时主要过程如何工作。一个生产系统可以处理各种各样的意外情况和用户

错误。原型开发必须与系统化设计、配置管理和测试方法结合起来,才能生产出符合生产系统要求的产品。

CASE 是开发原型的强大工具,本身就具有许多规则。它降低了叙述性文档的地位。现在的功能需求包含 CASE 输出,并附有解释特例的文字,而不是由流程图支持的文档。这样做不是为了减少用纸量。CASE 交互式显示是表现 CASE 设计的理想工具。

我们很容易看出 CASE 解决问题的神奇作用。但是,它仅仅是人们形成答案的工具。为了获得他们的目标,人们需要彻底了解 CASE 工具并用在一串补充性的过程中。CASE 指出建立某一需求和设计的标准,其余的需求仍要在安装标准中说明。

0.3 来 源

虽然 Oracle 是一家亿元公司,但是与用户的联系却是紧密。公司向用户提供大量的杂志,以使用户了解 Oracle 的培训人员、综合人员、咨询部门和世界各地的咨询机构。

Oracle 遵守 DBMS(数据库管理系统)软件的一致性理论。公司几乎每一个功能点都有计划。这些计划的优点在于与其它计划互相协调;缺点是有一些需要立即解决的问题,而解决方法却要在将来才能实现。第三方供应商一直积极地注意这种空缺。第十六章将比较 Oracle 和第三方供应商软件。

在开发应用系统之前需要研究一下软件包。至少看看包装上的说明,以便明确你的需求,并向设计者提供一些值得注意的技术。应用系统供应商却不注意这些,他们的系统比 DBMS 软件更加难以访问。本书还提供购物技巧——发现和衡量软件包的价值。

作者在写 Oracle 一书时不免对命名产生兴趣,就像在"See Ulka Rogers or Dave Hay for CASE support",或者"Charles Peck is the fellow to call for performance tuning help",或者"Dewey Allen is the best I know for configuring client/server installations."一样。因为 Oracle 是一家国际性的公司,并且新的专业人士不断涌现,本书重点强调寻求专家帮助的过程,而不是列出专家的姓名。

0.4 成 员

无论是靠天才还是靠运气,Oracle 比像 IBM 这样机构庞大的公司或是小型自治的 Microsoft 和日本公司提前进行了管理宗旨的重大改革。

组织带来了大量问题,如何配备职员?如何认识组织成功和个人成功?如何付报酬?管理需要哪些管理功能?根据不同的宗旨管理数据处理部的职责是什么?

任何一本书都可明确地回答这些问题。职员计划、人事管理和多面手、专家等几章谈到上述概念并提出一些建议。

0.5 读 者

本书主要针对使用 Oracle 的数据处理管理员,以及负责开发和维护工作的职员。根据本书阐述的管理宗旨,读者包括一线经理和分析人员,技术人员。

0.6 本书必要性

本书在若干方面有独到见解,它强调了 Oracle 的特性。虽然大多数数据处理管理教材认为书写一个数据处理系统能解决数据处理需求,但是本书认为只在功能买不来时才写这样的系统。本书提供的方法包括购货,它认识到并鼓励读者在 Oracle 环境中使用原型和 CASE 工具去定义过程来支持他们。

第一章 开发模型

系统开发过程大约在 20 年前就已定型,它适合于大多数系统的开发需求。但是软件的变化比开发宗旨的变化快得多,Oracle 处于软件进化的前沿。

本章简要介绍瀑布模型——一个系统开发的流行范例。我们将讨论在 Oracle 系统开发中它能做什么,不能做什么。另有一节以图形方式剖析 Oracle 系统,解释软件工程研究院的处理概念在 Oracle 环境中的含义。最后演示为什么尽管 IBM 模型进行多用户数据处理效果好,却还是很快被 Oracle 模型替代了。

本章只做为全书的序言,它介绍小型开发组织的框架,基本文档,并使开发人员能够定义处理过程。

1.1 瀑布开发模型与演示

模型能很好反映事物的真实程度。图 1.1 所示的传统瀑布模型描述了从需求到操作的线性开发过程。它精确说明了 from-scratch 系统开发阶段。回支显示了主要步骤,例如数据设计,被整个开发过程周期性地访问。该模型不涉及存在的元素,例如购买的和以前所写的代码。

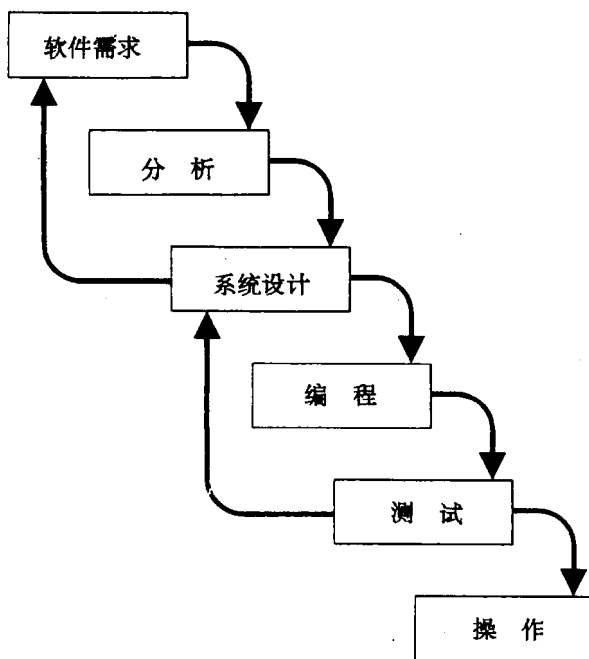


图 1.1 系统开发瀑布模型

瀑布模型虽然精确但是危险。因为瀑布模型的传统方法只看主流却忽视了细节。我们

在做分析之前必须研究需求,而进行分析又是为了设计。细节提供了反馈信息,分析过程教育了提出需求的人,设计过程可以改变分析,编程可以改变设计等等。

危险在于“每一步在下一步开始之前必须彻底完成”的概念。支持瀑布模型的标准指出在设计开始之前分析必须做完并得到认可。就是说不同方面的专家沿着这一大方向处理不同的步骤。从设计到编程会有一大堆的文档供管理员和程序员校正和接受。设计者的工作做完后他极有可能离开该项目。

上面的假设——整个项目必须在一大堆文档的指引下一个阶段一个阶段的进行,会导致一大堆生命周期文档的积累。这个假设的一个推论就是使用桌面文档是发现问题的最佳方法。但一般来说用户在看见代码出现问题之前意识不到这些问题。这种支持用写出标准的方法来建立项目的想法,常常是不可行的,即使是在 3GL 环境下。它们应当适当缩小以适应 Oracle 开发,原因如下:

- 因为从事每一项工作的职员数量较少,所以 Oracle 项目的内部联络较好。像 SQL * Forms 这样的 4GL 提供了更高的生产力。口头联络可以处理组织中大部分的信息传递。
- 用户通常投身于 Oracle 开发,而分析员/开发者更多的是从头到尾地做一个项目。口头联络为合作和连续性提供了更广泛的范围。
- 无须指出写什么或不写什么。CASE 和 Oracle 的 4GL 避免了在其他开发语言中出现的大量的重复代码。文件的打开和关闭,匹配例程、取数例程和记录描述都由关系数据库管理系统(RDBMS)直接处理。
- 分析员和用户合作研究数据设计和开发规则。用户很容易明白图形化的设计说明。图形已替代了文字叙述,特别是有了 CASE 和 ERwin 设计工具之后。
- 如图 1.2 所示数据设计的过程独立于其他部分,它的独立性充分利用了原型开发过程的特点,例如修改和报表子过程。该过程是可预言的,而且影响应用程序其他模块的风险也小。
- Oracle 4GL 工具是非过程化的,而且是自我说明的,根据 Oracle 的结构,它们描述目标而不是怎样做。
- 生命周期文档以今日的痛苦换取明日的收益。痛苦总是存在。Oracle 大部分的收益会得到充分发挥。因为 4GL、SQL * Forms、SQL * Report Writer 代码简单,所以维护起来相对容易。4GL 中的代码被分成若干目的单一的小块。在 CASE 生成代码后,维护工作只是修改设计和重复生成源代码。在原型法中与程序员一起工作的用户能很快发现程序的缺点,而这些缺点在设计说明书中尚未被认识到。

Oracle 像一列负载不多的高速列车。许多公司根据原型法过程的特性和上面谈及的有效性已经放弃了过分庞大的设计文档。它们还考虑其他风险,最大的问题是在解决问题之前没有仔细考虑,并且想当然地认为这套系统已经做完了。

某些文档编制工作还是必要的,它们包括策略和分析文档、数据设计,以及所有功能和界面方面的设计。每家公司都要花时间去查看其他文档以决定 Oracle 是否需要它们。

1.2 Oracle 系统剖析

Oracle 系统面向用户。图 1.2 显示了这种结构。主要结构化元素是维护数据库的表格处理、输出数据的报表处理和控制访问的菜单处理。

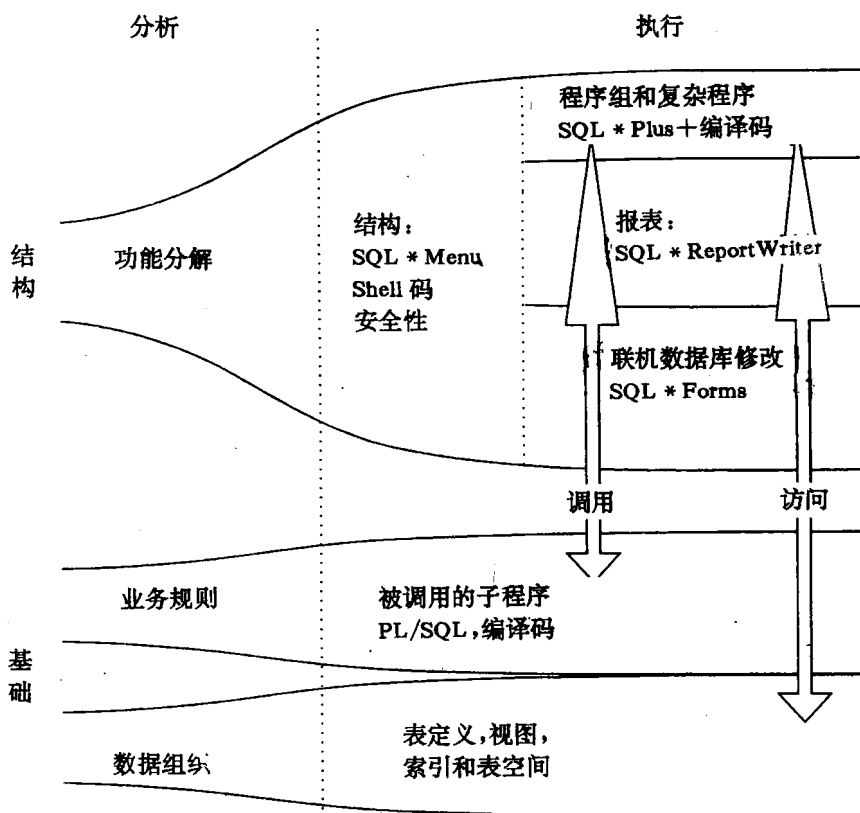


图 1.2 Oracle 应用系统结构

第三代代码和程序组系统中可以使用 Oracle 数据库管理系统。它比其他系统运行得好。但是真正认识到其生产力的巨大收益是在集成系统，像第四代语言之中。

如图 1.2 所示，可被系统中所有程序共享的元素叫基础（Foundation）。包括数据设计和公用子程序。这些模块通常用来完成商业规则或初始化过程。当程序成为原型开发者建立系统结构所用工具的一部分时，系统设计员为这些子程序写规格说明书。图 1.3 说明了子程序库与出现在系统功能分解和菜单上的主模块的关系。

菜单处理把结构提供给用户。它也遵循系统功能分解规则，只不过每个用户只看适合于他的功能子集。菜单包括联机修改，报表和批处理。Oracle 4GL 工具针对这些功能的是 SQL * Menu, SQL * Forms, SQL * Report Writer 和 SQL * Plus。它们中的任何一个或全部都可编译。许多用户在作业控制语言过程（job control language procedure）中写菜单系统，还有不少第三方供应商提供 Oracle 4GL 表格和报表工具。