

之二 INFORMIX-4GL

关系型数据库 管理系统

中国人民建设银行总行

电子计算中心

杨应辉 主编



电子工业出版社

内 容 提 要

本书主要介绍 INFORMIX 关系数据库,在阐述一般第四代语言特点的基础上,着重并详细地论述了 INFORMIX-4GL 的功能和特性,INFORMIX-4GL 程序设计的规则、如何使用 RDSQL-RDS、创建编译屏幕表格及报表的过程、4GL 库函数的调用、4GL 程序员设计环境等。

本书从工程应用设计出发,附有实际银行管理业务的应用设计举例,可供金融业务应用设计、管理信息系统设计及其他应用设计的软件人员使用,也可供大、专院校计算机软件工程学生参考。

关系型数据库管理系统

——INFORMIX-4GL

杨应辉 主编

关振胜 江峡 编写

责任编辑:和德林

电子工业出版社出版(北京市万寿路)
电子工业出版社发行 各地新华书店经营
电子工业出版社计算机排版室排版
北京燕山联营印刷厂印刷

开本:787×1092 毫米 1/16 印张:20.75 字数:525 千字

1992 年 10 月第 1 版 1992 年 10 月第 1 次印刷

印数:8000 册 定价:27.00 元

ISBN7-5053-2746-1/TP·865

前 言

为了加快中国人民建设银行计算机应用工作发展的步伐,有效地提高计算机技术人员的应用水平,推动技术更新,总行电子计算中心组织编写了这套《关系型数据库管理系统》培训教材。

本书的编写受到了总行领导的高度重视和大力支持。

本书分为两册,从实际应用的角度,全面系统地介绍了 INFORMIX 数据库的管理与应用。第一册的主要内容是 INFORMIX-SQL 和 INFORMIX-ESQL/Q 技术的应用。第二册的主要内容是 INFORMIX-4GL 的特点和使用方法。

本书由总行电子计算中心总工程师杨应辉担任主编,解彬彬和关振胜同志担任副主编。编写小组成员有河南洛阳市分行的柴振红和北京市分行信托投资公司的江峡。

本书第一册第一篇中的第一章、第二篇中的第一章和第七章、第二册中的第一章由关振胜编写。第一册第一篇中的第二章至第八章、第二篇中的第二章至第六章由柴振红编写。第二册中的第二章至第十六章由江峡编写。关振胜、解彬彬同志审稿。角彬彬同志负责本书编写的组织协调工作。

总行计算中心杜文和,河南省分行陈金海,北京市分行徐川零,天津市分行杨如生、陈彤彤,吉林省分行杨中清,广东省分行杨裕常,辽宁省分行赵爱平,河北省分行孙富英等同志参加了大纲的审定工作。

在本书的编写过程中得到了总行计算中心原副主任朱裕峰和河南省分行、北京市分行各级领导的大力支持。在此对他们以及所有为本书注过心血的同志们表示感谢。

鉴于编者的水平所限且时间匆忙,本书难免会有缺点、错误,敬请读者批评、指正。

中国人民建设银行总行电子计算中心
一九九五年五月

目 录

第一章 概 述	1
1.1 什么是第四代语言 4GL	1
1.1.1 什么是 4GL	1
1.1.2 4GL 产生的背景	2
1.1.3 4GL 的发展	3
1.2 什么是 INFORMIX-4GL	4
1.2.1 INFORMIX-4GL 快速开发系统	5
1.2.2 INFORMIX-4GL 交互式调试器 Debugger	5
1.2.3 INFORMIX-4GL 编译器	5
1.3 INFORMIX-4GL 4.1 版本规格	6
1.4 INFORMIX-4GL 的安装及使用前的准备工作	7
1.4.1 INFORMIX-4GL 的安装	7
1.4.2 使用前的准备工作	8
1.5 本书中的语法表示约定	8
第二章 面向程序员的工具及其使用方法	9
2.1 本章概要	9
2.2 INFORMIX-4GL 菜单的调用	9
2.2.1 INFORMIX-4GL 菜单图示	9
2.2.2 INFORMIX-4GL 菜单	10
2.3 程序结构	18
2.4 程序编译	20
2.5 程序运行	21
2.6 多模块的运行方式	22
2.6.1 Program 方式	22
2.6.2 操作系统下的命令文件方式	22
2.7 运行调用 C 函数的程序	22
2.7.1 编辑含有 C 函数信息的结构定义文件 Fgiusr.c	23
2.7.2 使用 CFGLGO 编译 Fgiusr.c 建立“用户执行程序”	24
第三章 建立数据库	29
3.1 本章概要	29
3.2 关于数据库的基本概念及数据类型	29
3.2.1 数据库的基本概念	29
3.2.2 数据类型	30
3.3 建立数据库	33
3.3.1 用 INFORMIX-SQL 建立数据库	33
3.3.2 用 INFORMIX-4GL 建立数据库	33

第四章 INFORMIX-4GL 数据库的一般操作	38
4.1 本章概要	38
4.2 选择数据库	38
4.3 定义程序变量	39
4.4 赋值语句	40
4.4.1 数值表达式	40
4.4.2 字符串表达式	41
4.4.3 布尔表达式	42
4.4.4 表达式中的空值	42
4.4.5 数据类型的转换	43
4.5 交互语句	43
4.5.1 接收输入 PROMPT 语句	44
4.5.2 显示输出 DISPLAY 语句	45
4.6 数据库表的基本操作	46
4.6.1 往数据表中插入数据行	46
4.6.2 从数据表中选择数据行	48
4.6.3 更新数据表中的数据行	49
第五章 SELECT 语句	51
5.1 本章概要	51
5.2 记录的定义	51
5.3 检索多个数据行的 SELECT 语句	52
5.3.1 为 SELECT 语句分配游标	53
5.3.2 检索并处理数据行	53
5.4 复杂的 SELECT 语句	59
5.4.1 SELECT 子句的三类用法	59
5.4.2 WHERE 子句的一些复杂用法	60
5.4.3 ORDER BY 子句	62
5.4.4 从若干个表中选择数据	63
第六章 函数	64
6.1 本章概要	64
6.2 FUNCTION 语句及函数调用	64
6.2.1 FUNCTION 语句	64
6.2.2 函数调用	65
6.3 全局与局部变量	68
6.3.1 GLOBALS 语句	68
6.3.2 作用域规则	70
6.3.3 同名变量	71
6.4 参数	72
6.4.1 在函数中使用参数	72
6.4.2 向主程序返回值的 FUNCTION 中参数的使用	74
6.4.3 在表达式中调用函数	74
6.5 INFORMIX-4GL 的函数库	75

6.5.1	ASC II	75
6.5.2	CLIPPED	76
6.5.3	COLUMN	76
6.5.4	DATE	77
6.5.5	DATE()	77
6.5.6	DAY()	77
6.5.7	MDY()	78
6.5.8	MONTH()	78
6.5.9	TIME	79
6.5.10	TODAY	79
6.5.11	USER	79
6.5.12	WEEKDAY()	79
6.5.13	YEAR()	80
6.5.14	SPACES	80
6.5.15	USING	80
6.5.16	ARG_VAL	84
6.5.17	ARR_COUNT	85
6.5.18	ARR_CURR	86
6.5.19	DOWNSHIFT	86
6.5.20	ERR_GET	86
6.5.21	ERR_PRINT	87
6.5.22	ERR_QUIT	87
6.5.23	ERRORLOG	87
6.5.24	INFIELD	88
6.5.25	LENGTH	88
6.5.26	NUM_ARGS	88
6.5.27	SCR_LINE	89
6.5.28	SET_COUNT	89
6.5.29	SHOWHELP	90
6.5.30	STARTLOG	90
6.5.31	UPSHIFT	91
6.6	C 函数在 INFORMIX-4GL 中的调用	91
第七章	设计屏幕表格	93
7.1	本章概要	93
7.2	屏幕表格的应用	93
7.3	屏幕表格说明文件的建立	94
7.4	屏幕表格说明文件的各部分	95
7.4.1	DATABASE 数据库节	95
7.4.2	SCREEN 屏幕节	95
7.4.3	TABLE 表节	97
7.4.4	ATTRIBUTES 属性节	98
7.4.5	INSTRUCTIONS 指令节	106
7.5	屏幕表格的程序员环境	108

第八章 屏幕表格在 INFORMIX-4GL 程序中的使用	111
8.1 本章概要	111
8.2 显示表格	111
8.2.1 显示屏幕表格	111
8.2.2 提示输入	114
8.3 编写使用屏幕表格的交互式程序	114
8.3.1 INPUT 语句	116
8.3.2 DISPLAY 语句	118
8.3.3 CLEAR 语句	119
8.3.4 SQLCA 记录	119
8.3.5 INT_FLAG 变量	121
第九章 使用多数据表屏幕表格和屏幕数组	126
9.1 本章概要	126
9.2 对多数据表屏幕表格的说明	126
9.3 屏幕数组的使用	128
9.3.1 屏幕数组在屏幕表格中的使用	129
9.3.2 定义程序数组	129
9.3.3 对数组操作的 FOR 语句	130
9.3.4 INPUT ARRAY 语句	131
9.3.5 滚动和编辑	132
9.3.6 程序数组和屏幕数组的内部函数	133
9.3.7 使用带有任选子句的 INPUT ARRAY 语句	134
9.3.8 DISPLAY ARRAY 语句	139
9.3.9 关于使用屏幕数组的程序实例	140
第十章 窗口	144
10.1 本章概要	144
10.2 窗口的使用	144
第十一章 用户菜单的建立与使用	149
11.1 本章概要	149
11.2 菜单的形式及使用	149
11.3 菜单设计	150
11.4 改变提示信息 and 提示行	151
11.5 建立帮助提示信息	153
第十二章 借助屏幕表格完成查询	156
12.1 本章概要	156
12.2 搜索标准	156
12.2.1 利用关系操作查询	157
12.2.2 使用通配符	157
12.2.3 使用多选择操作搜索	157
12.2.4 搜索某一范围内的值	158
12.2.5 查询操作符和短域	158

12.2.6 在 SMALLFLOAT 域和 FLOAT 域内输入搜索标准	158
12.3 依据搜索标准构造 SELECT 语句	158
12.3.1 CONSTRUCT 语句	158
12.3.2 构造 SELECT 语句	160
12.3.3 PERPARE 语句	161
12.3.4 执行预处理过的语句	161
12.3.5 用于实例查询的演示程序	162
第十三章 报表设计	165
13.1 本章概要	165
13.2 报表的基本格式设计	165
13.3 MAIN 或 FUNCTION 中报表语句的描述	167
13.4 REPORT 程序	168
13.4.1 DEFINE 节	168
13.4.2 OUTPUT 节	169
13.4.3 ORDER BY 节	171
13.4.4 FORMAT 节	171
13.5 REPORT 程序中用于 FORMAT 语句的几个语句	177
13.5.1 NEED 语句	177
13.5.2 PAUSE 语句	177
13.5.3 PRINT 语句	177
13.5.4 PRINT FILE 语句	178
13.5.5 SKIP 语句	178
13.6 表达式和嵌入的函数	178
13.6.1 聚合函数 GROUP	179
13.6.2 LINENO	179
13.6.3 PAGENO	179
第十四章 出错处理及用户中断	181
14.1 本章概要	181
14.2 预防出错	181
14.2.1 STATUS 变量——判断是否已出错	181
14.2.2 WHENEVER 语句——捕捉错误	182
14.2.3 STARTLOG 函数	183
14.2.4 ERRORLOG 函数	183
14.3 处理用户中断	184
14.3.1 DEFER 语句	184
14.3.2 INFORMIX-4GL 中代替 INTE-RRUPT 键的方法	185
第十五章 INFORMIX-4GL 的深层次应用	187
15.1 本章概要	187
15.2 存取权限	187
15.2.1 授予和取消数据库权限	187
15.2.2 授予和取消数据表权限	188
15.3 修改数据库	189

15.3.1 改变数据库的结构	189
15.3.2 改变数据表结构	191
15.4 事务处理.....	193
15.4.1 创建一个带有事务的数据库	193
15.4.2 指定事务	194
15.4.3 事务处理日志文件的维护	195
15.4.4 恢复数据库	195
15.5 轨迹跟踪.....	196
15.5.1 建立跟踪轨迹	196
15.5.2 恢复表	197
15.6 视图.....	197
15.6.1 建立和删除视图	198
15.6.2 通过视图的查询	199
15.6.3 通过视图的修改	199
15.6.4 视图的权限	200
15.6.5 使用视图的数据限定	200
15.7 输入和显示属性.....	200
第十六章 INFORMIX-4GL 语句一览表	203
16.1 本章概要.....	203
16.2 语句分类.....	203
16.3 标准语法格式.....	204
 附录 A 演示系统“电子信箱”的说明及部分程序清单	 212
附录 B 环境变量	243
附录 C INFORMIX-4GL 保留字	246
附录 D 系统目录.....	248
附录 E ASC II 字符集	251
附录 F INFORMIX-4GL 实用程序	252
附录 G 错误信息表.....	262
一、INFORMIX-SQL 错误信息	262
二、INFORMIX-TURBO 和 INFORMIX-MS-NET 错误信息	286
三、INFORMIX-4GL 错误信息	287

第一章 概 述

1.1 什么是第四代语言 4GL

1.1.1 什么是 4GL

4GL——第四代语言(Four Generation Language)是在数据库技术发展的基础上为解决数据处理领域的软件危机而诞生的新一代计算机语言系统。

4GL 最早出现于 70 年代后期,但作为产品进入我国计算机世界还是近几年的事情。4GL 产生的最直接原因是为了提高劳动生产率。所谓第四代语言,是相对于第三代语言 3GL(There Generation Language)而言,如 FORTRAN、COBOL、C、ADA 等等。但它们要比第三代语言更接近英语自然语言,开发效率更高,功能更强。据统计,采用 4GL 开发应用软件要比用第三代语言提高生产效率近十倍,在第三代语言中需要若干条语句才能实现的功能只需一条简单的 4GL 语句就可替代。因此,使用 4GL 进行程序设计能大大地缩短开发周期,加快建立大型应用系统的过程。另外,4GL 较之第三代语言的进步还在于 4GL 用非过程化语言取代了过程化语言,4GL 只要求用户提出目标,即提出“我要做什么?”,而不必描述实现此目标的途径,4GL 系统会自动给出“怎么做”。4GL 通常具有下列功能特点:

(1) 具有友好的用户界面。4GL 一般具有灵活易操作的用户界面,使得编制赏心悦目的屏幕管理变得简单易行,即使非专业程序员也能在很短的时间内运用自如,编制出自己满意的屏幕效果。

(2) 使用非过程化语言。4GL 与传统 3GL 的最大区别便是其非过程性。它已基本抛弃那种过程化语言的传统模式,而代之以抽象的非过程化。

(3) 具备强大的数据库接口。许多 4GL 提供了与流行的 DBMS 的程序接口,使得面向信息管理系统的应用变得简单易行。

(4) 提供应用程序生成器。提供一些通用程序用以完成诸如文件维护、屏幕管理、报表生成、查询等任务,把程序员从繁琐的重复性劳动中解放出来。

(5) 具有通用缺省功能。许多通用缺省功能已在 4GL 软件包中写好,开发者既可直接使用,也可自己编写特定功能程序来覆盖掉原有的缺省设置。4GL 为此提供了足够的通用性与灵活性。

(6) 提供高效的程序代码。4GL 通常提供的编程语言要比传统的 3GL 强大得多。用它编写的程序具有结构化强、简洁明了的特点,即使对于一个很复杂的系统,真正由程序员进行“编程”的工作量也很小。4GL 通过对重复性的任务提供更高级的程序界面达到减少代码量,从而提高了生产率。

4GL 语言除了具有以上这些比传统程序设计语言强大得多的功能特点外,其开发的应用程序还具有独立于某一特定技术的特点,于是大大缩短了开发周期,降低了开发成本,因而逐渐被广大用户接受和喜爱。

1.1.2 4GL 产生的背景

第四代语言产生的背景，主要是由于近二十年中计算机硬件迅速发展，机器的容量和処理能力大大增强，价格也不断下降。然而与此相反，应用系统的开发却存在着严重超时、超成本的现象。因此，在计算机应用推广中，软件开发形成了“瓶颈”，越来越满足不了日益高涨的应用需求。

形成这种“瓶颈”的原因，大致有两个方面：一是第三代语言功能的局限性，二是软件工程中经典的系统开发生命周期法 SDLC (System Development Life Cycle) 目前所遇到的困境。

(1) 第三代语言的局限性

大家知道，最初的机器语言、汇编语言等程序设计语言，都强烈地依赖于计算机的硬件环境，极大地制约了开发人员的创造性与灵活性。第三代语言如：FORTRAN、COBOL、C、ADA 等的出现，从很大程度上改善了这一状况，使应用系统的开发周期相对缩短，寿命延长，可维护性也相应增强。但仍未从根本上摆脱硬件和操作系统对开发语言的限制，不同机型间的软件移植依旧是一件非常困难的事情。

其次，第三代语言在易用性方面越来越显出它们的局限。衡量一种语言的功能和特性，通常是以灵活性和易用性作为标准。语言的易用性是通过编程工作量的大小和学习掌握的难易程度来体现的。一种语言的开发效率越高，其易用性就越好。语言的灵活性可以定义为开发人员在程序设计中可能控制的程度。一般说来某种语言越具有灵活性，它就越难以使用。第三代语言一般都具有很好的灵活性，能够实现复杂的逻辑控制，但是它也带来了不易学习、不易维护等弊端。另外，第三代语言都为过程化语言，对于开发大型复杂的应用系统来说，编程量大，所需人/年多，且难以维护，易用性差。因此，人们一直在试图研制出一种具有很好的易用性，同时又具有一定灵活性的编程语言，于是诞生了第四代语言 4GL。

(2) 系统开发生命周期法面临的困境

众所周知，系统开发生命周期法 (System Development Life Cycle) 是软件工程中经典的应用系统开发方法，这种开发方法的特点是把整个系统开发划分五个阶段，各阶段目的明确，任务清楚，文档齐全，每个开发阶段的完成都有书面审定记录，开发过程全面，调度有序。

然而，在实际开发实践中，SDLC 方法逐渐暴露出一系列弱点，严重影响了它的实用价值。这些弱点表现为：

软件开发周期长，风险大。有些应用软件，特别是一些大型 MIS 系统，系统开发周期有时持续数年之久。系统尚未开发成功，为之服务的单位管理体制和外部环境早已发生了变化，以致不得不重新修改设计。时间越久，越容易使用户逐步失去对该应用系统的热情和支持。

开发过程中用户介入深度不够，系统需求难以准确确定。而 SDLC 法要求完整而准确地确定系统需求，而后才能进行设计和实施。这样势必给系统设计造成困难。

文档繁多且容量巨大。对 SDLC 繁多的文档用户理解有困难，且各阶段的文档审批工作难以及时实现。这样就会影响开发周期中各阶段任务之间的顺利衔接，贻误开发周期。

大型的计算机应用系统往往是人、机和社会密切交互作用的系统，它的开发虽然应遵循软件工程的基本思想，但也要注意其自身的特点。事实上，要准确而完整地描述一个尚未出现的应用系统是什么样子，往往是不现实的，这就是 SDLC 多年来所面临的困境。

为了提高软件开发效率，人们开始深入研究软件工程学，进一步探索新的系统开发方法。

(1) 原型法 (Prototyping Approach) 的出现。

原型法是当 SDLC 面临困境时出现的一种新的系统开发方法。它不追求开发形式的规范与完善，而刻意追求实效，让用户自始至终参与系统开发，由他们自己来确定什么是真正需要的应用系统。

原型法强调要首先抓住一个应用系统的局部，了解其基本要求，然后以最快的速度构成一个实际可用的模型，让用户观看模型的演示，并实际使用这个模型，在亲口尝一下梨子的滋味之后，再不断提出改进与扩展的要求。根据用户的需要，开发人员不断地快速更新和扩展模型，使用迭代过程，最后将一个原型扩展成为一个信息系统。这就是人们俗称的“滚雪球方法”。

原型法是面向用户的“友好型”开发方法。用户在使用模型中，尤其在对输出报表、屏幕格式等系统输出的认识中逐步确定自己对系统的需求。它符合人们的认识规律，能最大限度地发挥出用户的热情和智慧。用户的深度介入和系统的高度可变性是系统开发成功的基础。

(2) 原型法的技术支持

原型法要求快速生成（数天至数周）一个模型，并能快速地进行不断的模型更新迭代过程。实现这一要求的技术支持是第四代语言和集成化软件开发。因为，只有使用第四代语言才能缩短开发周期，用户才能尽快看到应用系统的雏型，在尚未满足需求的情况下，用户可进一步提出反馈意见，开发者即可利用 4GL 的特点，对系统快速进行修改，使其更进一步接近系统目标。如此反复迭代，系统就会很快达到原设计的最终目标。4GL 是这种原型法有力的技术支持。

当前，世界上无论是大或小的计算机公司都把提高软件开发的生产效率作为头等的战略任务来看待，其目标就是要研制出各种 4GL 及 CASE 工具。如 UNISYS 的 MAPPER IC 便是在微机上实现的，并已在我国获得应用和推广的 4GL，它的强大而又方便直观的数据文件生成功能、报表和屏幕格式生成功能、数据运算处理功能等大大减轻了人们编程的工作量，使劳动生产率有了很大的提高。该公司在 A 系列机上还开发出了第四代语言 LINC，该语言开发效率高，运行速度快，成为金融系统业务处理软件得力的开发工具。最近几年，INFORMIX-4GL 在中国也得到越来越多的应用，其特点将在下面的章节中作详细介绍。

1.1.3 4GL 的发展

4GL 语言是当今计算机软件技术最新发展的结果，是适应复杂的用户环境而诞生的，它为应用软件开发者和最终用户带来了一种革命性的变化。

随着计算机技术的不断发展，4GL 又溶入了许多最新发展的技术。如：

(1) 事件驱动 (Event-driven)。程序的非过程性进一步加强，程序执行的进程由突发性事件来控制，打破了传统的顺序执行方式。

(2) GVI 技术。许多 4GL 产品可支持最新发展的 GVI 环境。

(3) 分布式数据共享。随着 Client /Server 结构的日益时尚，分布式数据共享变得越来越重要，一些完全支持 Client /Server 结构的 4GL 产品已投放市场，从而使 4GL 更添魅力。

(4) 兼容多种数据源。为了适应兼容多种软硬件资源的系统的复杂性，4GL 产品能支持许多不同的数据源 (DBMS/FMS)。

(5) 多媒体技术

由于系统的开放性已成为计算机信息处理技术的首要要求，4GL 本身的开放结构使得设计开放的信息系统成为可能。

1.2 什么是 INFORMIX-4GL

INFORMIX-4GL 诞生于 1986 年,是 INFORMIX 软件公司针对数据库应用而设计的第四代语言。

INFORMIX-4GL 产品家族中包括了 INFORMIX-4GL 快速开发系统 R4GL (Rapid 4GL), INFORMIX-4GL 交互式调试器(Debugger)以及 INFORMIX-4GL 编译器。它是一个综合性的第四代应用程序开发和生成环境,它所提供的强大功能和极好的灵活性完全覆盖了诸如 C 或 COBOL 等第三代语言。

INFORMIX-4GL 是一个自包含的应用程序开发环境,具有以下特点:

- 能提供快捷的开发和交互式调试功能
- 能产生高效代码的生成环境
- 集成了最复杂的应用所需要的功能
- 不再需要使用第三代语言
- 能在若干年之内很轻松地维护应用程序
- 属于基本工业标准 SQL(Standard Query Language)语言
- 具有易移植性

虽然 INFORMIX-4GL 是专门为数据库应用而设计的,是非过程化语言,但它仍保留了很多通用程序设计语言的过程化特征,使其在具有易用性的同时,也具有一定的灵活性。比如,它包含许多与通用程序设计语言相似的语句,有赋值语句 (LET)、循环语句 (WHILE、FOR)、条件语句 (IF, CASE)、分支语句 (FUNCTION) 等等;它提供了丰富的数据结构,如记录、数组;还提供了函数语句,使程序的结构更加清晰明了。

INFORMIX-4GL 的过程和非过程语句的结合特性为使用者提供了特殊的能力,过程语句给开发者提供了全部的灵活性,可以补充实现 4GL 开发者在某些场合未能预见的功能要求,而象 MENU、OPEN WINDOW 等非过程语句则使开发者免去编写数以百计的、精细的代码,较少的代码意味着较少的维护时间和较高的劳动效率。

INFORMIX-4GL 简洁且类似英语,十分有助于浏览、编码及修改。

INFORMIX-4GL 建立在关系数据库 RDSQL (Relation Data Standard Query Language) 基础之上,RDSQL 是 INFORMIX 公司对结构化查询语言 SQL 的扩展,符合 ANSI 规定的数据库语言 SQL 工具标准,它提供对数据库的插入、查询、修改、删除等操作。RDSQL 作为 INFORMIX-4GL 的语言成份,程序员可在 4GL 程序中直接书写 RDSQL 语句,对数据库进行操作。它还可以与用任何一个基于 SQL 工具所建造的应用程序相集成。这些工具中有 INFORMIX-SQL 和 INFORMIX-ESQL/C (在 C 中嵌入了 SQL 语言,使得在用 C 语言所创建的应用程序中使用 SQL 命令)。

INFORMIX-4GL 包括一个屏幕建立实用程序——FORM4GL,开发者可以用它进行定义然后在屏幕上画出一系列弹出式屏幕格式 (Form) 作为应用程序的用户界面,同时还可规定各个 Form 之间的约束关系,并能用 INFORMIX-4GL 的交互式语句 (INPUT) 读入用户在屏幕格式上输入的数据,或将数据通过显示语句 (DISPLAY) 显示在屏幕上。INFORMIX-4GL 能自动处理光标的移动,并在数据录入时提供正文编辑功能。

INFORMIX-4GL 提供了功能强大的菜单语句 MENU,它简化了程序员建立菜单的过程。只要在程序中用菜单语句描述菜单的各选择项及该选项下的相应操作,INFORMIX-4GL 运

行系统将按用户的描述显示菜单，接收光标的移动及菜单的选择，并对每个确认的选择进行相应的操作。

INFORMIX-4GL 还提供菜单选项的联机求助功能 (HELP)。程序员只需写出求助文件，并在菜单语句中说明每个选择项对应的求助信息代码，INFORMIX-4GL 在运行时通过接收到的求助键在屏幕上显示可翻阅的菜单帮助信息。

对数据库应用系统来说，从中提取大量数据形成各类报表是必不可少的，INFORMIX-4GL 提供了建立各种报表的语句。使用 4GL 语句可以查询数据库中的信息，并对它们排序或分组处理，最后按格式输出。4GL 提供了很多内部函数及格式排版功能，使报表的生成变得内容丰富，却又简单易行。

INFORMIX-4GL 可以在屏幕上提供大小不同的长方形窗口，我们在程序中只要给出窗口的大小、位置，4GL 运行系统就能在屏幕上开出相应的窗口，并在窗口上执行一系列的语句。这样可以使设计出来的程序界面层次清晰、灵活友好，也加大了屏幕上的信息容量。

此外，INFORMIX-4GL 有两种新的方法可以方便地获得出错信息。第一种，是由电子设备生成的错误信息，包括错误代码，描述文本及对应的动作，以 ASC II 码和 Postscript 格式输出；第二种是一个称作“FINDER”的工具，它将错误正文及对应的动作都显示出来，这样，我们就不再需要拿着手册去找错误代码的解释串了，调试起来十分方便。

1.2.1 INFORMIX-4GL 快速开发系统

INFORMIX-4GL 快速开发系统 (R4GL) 是 INFORMIX-4GL 的解释版，它通过减少编译时间来提高开发者生产率。

INFORMIX-4GL 快速开发系统采用一步编译过程，开发者所写的代码被极快地编译成伪码 (P 代码^{*}，P-code)，读进存储器并由 P 代码执行器执行，从而将编译时间减至最小，P 代码编译器对于一个源码程序的编译速度比普通的快 60 倍。

1.2.2 INFORMIX-4GL 交互式调试器 Debugger

INFORMIX-4GL 交互式调试器为开发者提供了一个友好的、易于使用的调试环境，使我们能在内部一步一步地运行程序，程序员由此可以：控制运行中程序的执行，观察及改变变量的内容，逐步地分析程序的执行。

Debugger 也是一个维护 INFORMIX-4GL 程序的强有力工具。它可以帮助程序员去理解一个已有的应用程序，可以让程序员在程序中很方便地加入调试点。

本书中不涉及这部分内容，有兴趣的读者需参阅专门的资料。

1.2.3 INFORMIX-4GL 编译器

一旦完成了开发阶段，准备进入生成环境时，开发者需要的是尽可能快的性能，通过将 INFORMIX-4GL 源码编译成标准的 C 语言代码（它在下一步就被编译成机器码），INFORMIX-4GL 编译器就能充分将计算机的速度体现出来了。

* P 代码——移植码，用一种编译程序将源码译成一种叫做 P 代码的中间代码，再由主机上的一种专门的 P 代码解释程序得到可执行的目标码。其优点是可移植性，每一台主机只需要它自己的 P 代码解释程序而不需要不同的编译程序。这种编译程序模块对所有机器是标准化的，其缺点是执行速度比一次全部译成目标码要慢。在第三代语言中，许多 PASCAL 版本使用 P 代码。

1.3 INFORMIX-4GL 4.1 版本规格

本书以 INFORMIX-4GL 4.1 版本为描述蓝本，书中给出的所有示例均在 SCO UNIX 2.3.2 操作系统的 INFORMIX-4GL 4.1 版本环境中运行通过。

INFORMIX-4GL 可以在 85 个不同的制造商所制的超过 450 种硬件机器上运行。它在这些平台上都是兼容的，不需要重新生成代码。

INFORMIX-4GL 4.1 版本 Rapid 开发系统 UNIX 规格

机器需求

标准 RAM 需求

小型多用户 (1~10 用户) 1MB~5MB

大型多用户 5MB+

标准内存占用量

程序员环境 320K+

P 代码执行器 305K+

数据库引擎

INFORMIX-SE 700K

INFORMIX-OnLine 1000K

磁盘占用量 8.6MB

运行量 4.4MB

数据类型

数组 字节 * 字符 日期 时间 十进制

浮点数 整数 间隔 金额 记录 短浮点

短整数 文本 * Varchar *

* 仅用于 INFORMIX-OnLine 中

显示特性

颜色

白、黄、洋红、红、青、绿、兰、黑

属性

正常、下划线、黑体、暗淡、反转、闪烁、不可见

(视频颜色及属性依赖于终端的类型以及所用的 termcap 或 terminfo)

最大窗口数量

与存储器的可寻址空间大小有关

报告规格 (最大规格)

输出行长度 无逻辑上的限制+

报告的列数 无限制

用户变量 与机器有关+

变量总数 与机器有关+

按列序的组合长度 ... 在 INFORMIX-SE 中最多可达 120 字节在 INFORMIX-OnLine 中最多可达 255 字节

中止级数量 可达 8

表格规格 (最大规格)

表格数量 无逻辑上限制+

每一表格的屏幕数 1

每一表格的列数 与机器有关+

INFORMIX-4GL 4.1 版交互式调试器 UNIX 规格

1991 年 7 月

机器需求

标准 RAM 需求

小型多用户 (1~10 用户) 1MB~5MB

大型多用户 5MB+

标准内存占用量

交互式调试器 450K+

数据库引擎

INFORMIX-SE 700K

INFORMIX-OnLine 1000K

磁盘占用量 5.9MB

INFORMIX-4GL4.1 版编译器 UNIX 规格

1991 年 7 月

标准 RAM 需求

小型多用户 (1~10 用户) 1MB~5MB

大型多用户 5MB+

标准存储器占用量

程序员环境 320K+

数据库引擎

INFORMIX-SE 500K

INFORMIX-OnLine 1000K

磁盘占用量 8.5MB

运行量 2.6MB

数据类型

数组	字节 *	字符	日期	时间	十进制
浮点数	整数	间隔	金额	记录	短浮点
短整数	文本 *	Varchar *			

* 仅用于 INFORMIX-OnLine 中

显示特性

颜色

白、黄、洋红、红、青、绿、兰、黑

属性

正常、下划线、黑体、暗淡、反转、闪烁、不可见
(视频颜色及属性依赖于终端的类型以及所用的 termcap 或 terminfo)。

最大窗口数量

与存储器的可寻址空间大小有关

报告规格 (最大规格)

输出行长度 无逻辑上的限制+

报告的列数 无限制

用户变量 与机器有关+

变量总数 与机器有关+

按列序的组合长度 ... 在 INFORMIX-SE 中最多可达 120 字节在 INFORMIX-OnLine 中最多可达 255 字节

中止级数量 可达 8

表格规格 (最大规格)

表格数量 无逻辑上限制+

每一表格的屏幕数 1

每一表格的列数 与机器有关+

注: 所需要的 C 语言编译器不包括在本产品中。+ 依赖于存储器的可寻址空间大小。

这里所列出的所有技术指标是根据 1991 年 7 月 INFORMIX-4GL 编译器、INFORMIX-4GL 快速开发系统以及 INFORMIX-4GL 交互式调试器 4.1 版本在带有 SunOs 3.4 操作系统的 Sun Mod13/280 机器上运行所获得的结果。这些指标在其它机种及不同的操作系统配置下会有所不同, 因此仅作为参考之用。

1.4 INFORMIX-4GL 的安装及使用前的准备工作

1.4.1 INFORMIX-4GL 的安装。

INFORMIX-4GL 各种版本的安装都基本相同,并有专门的安装手册供使用参考。下面简要说明一下 INFORMIX-4GL 的安装过程:

- (1) 在 UNIX 系统下,注册 ROOT(超级用户)。
- (2) 建立 INFORMIX 用户。
- (3) 将 INFORMIX-4GL 软件拷入 usr/informix 目录下。
- (4) 键入安装命令 INSTALL4GL,在输入二次数列号后,INFORMIX 系统自动完成安装过程。

1.4.2 使用前的准备工作

假定 INFORMIX-4GL 软件已从有关介质中根据安装命令完成安装。

在 UNIX/XENIX 操作系统上需设置下述环境变量:

INFORMIXDIR——INFORMIX-4GL 软件安装目录

TERM——终端类型

TERMCAP——终端数据库

PATH——系统命令目录

举例说明:假定 INFORMIX-4GL 安装在/usr/informix 目录下且使用标准(Bourne Shell)和 VT100 终端,则在命令提示符下设置上述环境变量如下:

✎ INFORMIXDIR=/usr/informix <CR>

✎ TERM=VT100 <CR>

✎ TERMCAP=/usr/informix/etc/termcap <CR>

✎ PATH=\$PATH:/usr/informix/bin <CR>

✎ EXPORT INFORMIXDIR TERM TERMCAP PATH <CR>

上述命令应放在 .PROFILE 文件中,使得每次注册后操作系统自动执行以上环境变量的设置。

1.5 本书中的语法表示约定

ABC 黑体大写字母标出的标识符表示关键字。INFORMIX-4GL 在内部处理时不考虑输入的是大写或是小写;

abc 小写字母标出的单词是临时代用名称,实际键入时要用实际值来代替;

() 圆括弧本身是语法的一部分,不是特殊符号;

[] 方括弧是一个特殊符号,其中的内容是选择项;

| 竖线是一个特殊符号,是多个选择项的分隔符;

{ } 花括弧是一个特殊符号,其中的内容表示必须选择其中的一个选择项(用|分隔)。