

系列之二

ORACLE关系数据库管理系统

用 户 手 册

(八)

PRO * FORTRAN

用户指南

《计算机技术》编辑部

编 者 的 话

《ORACLE数据库管理系统使用手册》系列资料是根据美国Oracle公司ORACLE5.1版翻译出版的。目前与读者见面的是该系列资料的第二批共九册书。

ORACLE关系数据库是用高级程序设计语言C编写的。它采用SQL数据语言,具有广泛的兼容性和可移植性,是目前国际上唯一可在世界各主要计算机厂家70余种大小微机(IBM的4300系列、3900系列、PS/2、PC;DEC的VAX各系列,PDP系列;DG系列;DPS系列;HP系列;国产0500、0600系列微机和2000系列小型机、超级小型机等等)系统上运行的RDBMS。

ORACLE关系数据库系统可为不同类型的计算机提供整体化的标准软件环境;与SQL/DS和DB2、DB3等数据库系统兼容;能够直接使用IBM现有的数据库系统的数据和应用程序;可在MVS、VM、VMS、DOS、UNIX、XENIX、UX等十几种著名操作系统下运行,且具有相同的用户界面,使用户在更换或升级系统时都不会影响已开发的软件;还为用户提供一整套包括格式化处理、菜单管理、表格图形、报表生成等第四代语言工具在内的支撑工具环境;另外还有很强的数据词典和决策支持系统等功能。

本资料完整、系统地介绍和分析了ORACLE RDBMS 5.1的全部功能特点和基本原理;全面地讲解了它的操作方法及步骤,以及有关注意事项和维护知识。

因此本资料既可作为广大计算机用户及软件人员使用ORACLE数据库的培训材料和操作指南,又可作为计算机科技工作者、大专院校师生学习掌握大型数据库管理系统的一般教材。

由于时间仓促,水平有限,不当与错误之处在所难免,欢迎广大读者批评指正。

本套资料中的《PRO * FORTRAN》由段云亭、聂培尧、金颖、焦云鹏翻译,宗拔梅、陈骏林译校。

《计算机技术》编辑部

一九八九年三月

目 录

前言.....	(1)	建立表.....	(19)
第一部分: PRO*FORTRAN 预编译程序接口		提示插入行的值.....	(20)
第一章: Pro*FORTRAN 介绍.....	(6)	提示更新时使用的值.....	(25)
pro*FORTRAN 是什么.....	(6)	查询.....	(26)
pro*FORTRAN 的特点.....	(7)	输入主变量.....	(27)
一般概念.....	(7)	输出主变量.....	(27)
pro*FORTRAN 命令.....	(7)	只返回一行的查询.....	(27)
FORTRAN 语句和 SQL 语句混合使用.....	(7)	数据转换.....	(28)
命令前缀 EXEC SQL.....	(8)	数值数据转换.....	(28)
命令前缀 EXEC ORACLE.....	(8)	字符数据转换.....	(28)
SQL 执行语句和说明语句.....	(8)	转换错误.....	(28)
pro*FORTRAN 程序的组成.....	(8)	返回多行的查询所使用的指针.....	(29)
第二章 应用序言.....	(10)	DECLARE CURSOR 语句.....	(29)
头部: DECLARE 部分.....	(10)	CURRENT OF CURSOR 子句.....	(29)
主变量.....	(10)	OPEN CURSOR 语句.....	(30)
主变量准则.....	(11)	取活动集的行.....	(30)
FORTRAN 支持的数据类型.....	(11)	CLOSE CURSOR 语句.....	(31)
指示符变量.....	(12)	指针类型.....	(31)
指示符变量准则.....	(12)	程序实例.....	(32)
VARCHAR 伪类型的说明和使用.....	(13)	带 WHERE 子句的查询.....	(32)
头部: SQL 通讯域的说明.....	(14)	使用提示的比较复杂的查询.....	(33)
ORACA, SQLCA 的扩展.....	(15)	EXEC ORACLE OPTION 语句.....	(39)
ORACA 中的信息.....	(15)	使用数组.....	(39)
头部: 连接 ORACLE.....	(16)	数组 BIND/FETCH 特征.....	(39)
自动注册连接.....	(16)	数组取数.....	(39)
通过 ORACLE*Net 连接.....	(16)	数组取数的限制.....	(40)
进入和退出 ORACLE.....	(17)	数组联编 (BIND).....	(41)
第三章 应用体系.....	(19)	关于数组的一般警告.....	(41)
带 DML 语句的 pro*FORTRAN 程序实例.....	(19)	FOR 语句使用 BIND/FETCH 数组.....	(41)
		使用数组从文件插入的实例程序.....	(42)

使用数组更新.....	(45)	例.....	(67)
使用数组选择.....	(49)	PREPARE和EXECUTE 的	
第四章 工作的提交和滚回	(52)	补充实例.....	(68)
工作的逻辑单位.....	(52)	方法 3 : PREPARE, OPEN和	
工作单位的开始.....	(52)	FETCH	(69)
工作单位的结束.....	(52)	PREPARE, DECLARE, OPEN,	
工作单位需求的资源.....	(53)	FETCH 例子.....	(70)
提交工作.....	(53)	方法 4 : 插述符使用.....	(72)
工作滚回.....	(53)	SQL描述符.....	(72)
释放选择.....	(54)	选择表描述符.....	(74)
第五章 错误检测和恢复	(55)	详细解释选择表描述符的实例...	(75)
指示符变量返回值的使用.....	(55)	联编描述符介绍.....	(78)
指示符变量和 NULLS 的使用.....	(55)	处理运行时查询.....	(82)
NULL值的查找.....	(55)	说明所有描述符变量.....	(83)
插入CNULL 值.....	(55)	预备SQL语句.....	(83)
NULL 值的输出.....	(56)	说明语句的指针.....	(83)
SQLCA 结构.....	(56)	描述联编描述符.....	(83)
何时引用 SQLCA.....	(57)	打开指针.....	(83)
SQLCA 元素的意义.....	(57)	描述选择描述符.....	(84)
WHENEVER语句.....	(59)	从活动集取行.....	(84)
WHENEVER语句的语法.....	(59)	关闭指针.....	(84)
WHENEVER语句的范围.....	(60)	第七章 调用 pro*FORTRAN	
WHENEVER对明显错误的		(PCC命令)	(85)
检查.....	(60)	运行 pro*FORTRAN 所需的条	
第六章 动态定义语句	(62)	件.....	(85)
动态定义语句的定义.....	(62)	需要重新预编译1.0版本的程序...	(85)
动态定义语句的类型.....	(62)	置目录或路径.....	(85)
接收动态SQL语句的输入.....	(63)	命令语法.....	(85)
方法 1 : EXECUTE IMMED-		要求的变元.....	(85)
IATE	(64)	Pro*FORTRAN 选择项.....	(86)
EXECUTE IMMEDIATE的		AREASIZE	(86)
先决条件.....	(64)	ASACC	(86)
EXECUTE IMMEDIATE的		BEGLABEL.....	(86)
阻止.....	(66)	ENDLABEL.....	(86)
EXECUTE IMMEDIATE的	(66)	ERRORS.....	(87)
例子.....		FORMAT	(87)
方法 2 : 使用PREPARE和		HOLD_CURSOR和RELEASE	
EXECUTE.....	(66)	CURSOR 选择.....	(87)
PREPARE和EXECUTE的实		HOST.....	(88)

INCLUDE.....	(88)
IRECLEN.....	(88)
LITDELIM.....	(88)
LNAME.....	(88)
LRECLEN.....	(89)
LTYPE.....	(89)
MAXLTERAL.....	(89)
MAXOPENCURSORS.....	(89)
ONAME.....	(89)
ORACA.....	(89)
ORECLEN.....	(89)
PAGELEN.....	(89)
REBIND.....	(89)
SELECT_ERROR.....	(90)
USER'D.....	(90)
XREF.....	(90)
编译和连接.....	(91)
条件预编译.....	(91)
独立预编译.....	(92)
混合pro*FORTRAN和ORACLE	
调用接口程序的混合使用.....	(92)
当预编译时发生什么?.....	(93)
运行时间一致性检查.....	(93)

第二部分: PRO*FORTRAN

ORACLE调用接口

第八章 介绍怎样编写调用接口程序

.....	(94)
基本程序结构.....	(94)
指针数据区.....	(96)
注册数据区.....	(100)
程序接口数据区.....	(101)
一般编码规则.....	(101)
选择参数.....	(102)
使用替代变量.....	(102)
使用指示符变量.....	(102)
关于使用编译程序优化器的注	
释.....	(103)

第九章 程序调用说明..... (104)

OLON调用.....	(104)
ORLON调用.....	(104)
COPEN调用.....	(105)
OSQL3调用.....	(106)
ODSC调用.....	(107)
ONAME调用.....	(108)
ODEFIN调用.....	(109)
OBNDRV和OBNRN调用.....	(110)
OOPT调用.....	(112)
OEXEC调用.....	(113)
OEXN调用.....	(113)
ORES调用.....	(114)
OFETCH调用.....	(114)
OFEN调用.....	(115)
OBREAK调用.....	(115)
OCAN调用.....	(116)
OCOM调用.....	(116)
OROL调用.....	(116)
OCON调用.....	(117)
OCOF调用.....	(117)
OERMSG调用.....	(117)
OCMN调用.....	(118)
OCLOSE调用.....	(118)
OLOGOF调用.....	(118)

第十章 老的OCI (pro*SQL)调用 (119)

OLOGON调用.....	(120)
OSQL调用.....	(120)
ODSRBN调用.....	(121)
ODFINN调用.....	(122)
OB'ND和OBINDN调用.....	(123)

第十一章 数据类型..... (125)

数据类型说明.....	(125)
数据转换.....	(128)

附录A pro*FORTRAN预编译程序

错误信息..... (129)

附录B 一般程序设计指南..... (132)

附录C 保留字..... (133)

附录D pro*FORTRAN程序实例... (134)

前 言

目的

本手册提供了程序接口资料，这一程序接口可用来开发定义和操纵ORACLE RDBMS上数据的应用程序。

ORACLE为应用程序员提供两种高级接口。第一，也许是对研制应用程序最有效的接口，即预编译程序。由于有预编译程序，便可能使用高级语言（如FORTRAN, pascal, C, PL/1, 或COBOL）来写应用程序。在这种高级语言程序中可包含SQL语言的语句。这样就能在单一应用程序中具有高级语言和SQL语言的两种语言特性。

第二个接口是被称为ORACLE调用接口（如我们所知道的高级接口（HLI）），也允许高级语言在ORACLE关系数据库管理系统内存取数据。使用ORACLE调用接口的程序，能直接调用包含语言专门运行时间库的ORACLE子程序。

这两个接口，提高了被用于在ORACLE数据库上操纵数据的应用程序的开发过程。

读者

这手册是为使用ORACLE所提供的程序设计接口之一的应用程序员设计的。

假定这手册的读者是：

- 习惯于FORTRAN程序设计的程序员；
- 有在ORACLE RDBMS方面处理数据有经验的人；
- 谙熟SQL语言的人。

关于SQL语言语法说明，参见SQL*plus用户指南和ORACLE数据库管理员指南。

内容

这手册分两部分：第一部分包括使用pro*FORTRAN预编译程序接口的信息；第二部分提供pro*FORTRAN ORACLE调用接口的资料。

这手册的读者们也应参考pro*ORACLE的发行注记，这些信息现在还不包括在这手册。例如，包含在ORACLE通讯域（ORACLE）上的信息，这信息将出现在手册的最后版本。

下面给出手册概要：

第一部分：pro*FORTRAN预编译程序接口。手册的这部分叙述pro*FORTRAN预编译程序接口。

1. pro*FORTRAN介绍。这章是对pro*FORTRAN的一般介绍。讨论你将怎样使用它，一般概念和定义，部分pro*FORTRAN程序和程序中出现的语句类型。

2. 应用程序。这一章讨论在pro*FORTRAN程序中使用的变量的说明，讨论在应用程序中所需要的项目。本章也有一个短的代码例子，用以说明如何进入及退出ORACLE系统。

3. 应用体。这章讨论包括在应用体内的不同语句类型，包括数据操作语句和数据定义

语句。这一章介绍指针的概念，指针是在许多查寻结果被返回时而使用的。在介绍了不同指针命令后，我们给出了两个实例程序。在这些例子中也进行了FORTRAN数组的讨论。

4. 提交和重新运行工作。这章定义一个事务或工作的逻辑单位，这些逻辑单位可以由若干SQL语句一起组成的。程序员可以编写程序，程序执行该事务时他可以根据时间来决定事务提交（使“永”在数据库中）或“滚回”（从数据库中删除）。

5. 错误检测和恢复。在这一章中，程序员学习怎样使用SQLCA和指示变量，在不同状况下进行检测及处理，如空值，无条件删除，不返回查询行以及数据溢出或截断等。

6. 定义动态语句。这章涉及高级程序设计技术，这些技术对编写非常灵活的程序是有用的，但也要求程序员比较懂得用FORTRAN和SQL语言编写程序。初学使用 pro*FORTRAN的用户，将跳过或浏览这一章。

7. 调用pro*FORTRAN (PCC命令)。在这一章叙述PCC命令和它的选择项。

第二部分：pro*SQL ORACLE调用接口。手册的这一部分讨论ORACLE调用接口。

8. 如何写ORACLE调用接口程序的介绍。这一章介绍写OCI程序使用的概念，如指针数据域，注册数据域，“返回码”，参数类型和程序结构。

9. 单个程序调用说明。这章介绍每个OCI调用。每个调用部分显示一般语法，并给出FORTRAN的例子。调用是按一般可能出现的顺序来讨论，而不是字母顺序（在附录中建立了一个调用摘要，按字母顺序显示调用）。

10. 老的pro*SQL (OCI) 调用。这一章概括在第2章取代的那些较老的调用。但是为了完整性，仍将包括在内。对每个调用部分，显示一般语法，包含它的使用和参数叙述，并给出FORTRAN的例子。调用一般是按它们可能在程序中出现的顺序叙述，而不是字母顺序（在附录中那些调用的摘要按字母顺序）。

11. 数据类型。这一章讨论pro*SQL和ORACLE RDBMS使用的数据类型，以及所能完成的数据类型的转换。

附录A： pro*FORTRAN预编译程序错误信息。这附录列出pro*FORTRAN 预编译程序错误信息。

附录B： 一般程序设计指南（导向图）。这附录给出写pro*FORTRAN程序一些一般提示。

附录C： 保留字。这附录列出用户通过ORACLE RDBMS或pro*FORTRAN使用的保留字，这些保留字不能作为用户命名的ORACLE目标来使用，如表 (tables)，栏 (columns)，或视图 (views)。

附录D： pro*FORTRAN程序实例。这附录包含实例程序列表，这些实例程序包括在pro*FORTRAN用户分配的目标代码介质中。

附录E： 带动态SQL的pro*FORTRAN程序实例。这附录包括使用动态SQL实例程序的列表，包括分配给pro*FORTRAN用户的目标代码介质。

附录F： 使用ORACLE调用接口的FORTRAN程序实例。这附录包含一个完整的FORTRAN程序，这程序在执行某些检查之后，提示用户向数据库输入（关于雇员的信息）和增加所的雇员记录。

附录G： 程序调用摘要。这附录包含这手册中涉及到所有pro*SQL的字母提要，并且对快速查阅非常有用。

与此有关出版物

在序言这一部分中列出对你的产品所涉及的一些书和出版物。其后是按产品分组列出所有的出版物，你只选择适用于你的书的那些。与这手册一起，你可参考下面oracle公司的出版物。你将自动接收涉及到你购买到的产品的题目，你不必也不用接收所有资料。为反映产品不断变化，发表的注释被频繁更新。

Oracle有限公司出版物：

对ORACLE RDBMS的资料部分是：

- ORACLE RDBMS注释ORACLE部分 No.3001
- 对ORACLE部分的ORACLE综述和导引，No.3801
- ORACLE数据库管理员指南ORACLE部分No.3601
- ORACLE实用用户指南ORACLE部分No.3602
- ORACLE报告输出用户指南：RPF/RPT ORACLE部分 No.3603
- ORACLE错误信息和代码ORACLE部分 No.3605

不同预编译程序产品的资料部分（pro*C，pro*FORTRAN，pro*COBOL，pro*PL/I）是：

- pro*ORACLE注释ORACLE部分 No.3007
- pro*C用户指南ORACLE部分 No.3504
- pro*COBOL用户指南ORACLE 部分No.3503
- pro*FORTRAN用户指南ORACLE部分 No.3502

产品SQL*plus资料部分是

- SQL*plus注释ORACLE部分 No.3003
- SQL*plus用户指南ORACLE部分 No.3201
- SQL*plus快速流览ORACLE部分 No.3803
- SQL*plus快速引用ORACLE部分 No.3703

SQL*Forms产品资料部分：

- SQL*Forms注释ORACLE部分 No.3004
- SQL*Forms用户指南ORACLE部分 No.3301
- SQL*Forms设计指南ORACLE部分 No.3302
- SQL*Forms用户快速调用ORACLE部分 No.3704
- SQL*Forms设计人员快速调用ORACLE部分 No.3708
- SQL*Forms的快速流览ORACLE部分 No.3804

SQL*Menu产品的资料部分是：

- SQL*Menu注释ORACLE部分 No.3009
- SQL*Forms用户指南ORACLE部分 No.3303
- SQL*Menu的快速流览ORACLE部分 No.3808

Easy*SQL产品的资料部分是：

- Easy*SQL注释ORACLE部分 No.3002
- Easy*SQ介绍ORACLE部分 No.3810
- Easy*SQL用户指南ORACLE部分 No.3101

- Easy*SQL的快速流览ORACLE部分 No.3802
- Easy*SQL的调用卡片ORACLE部分 No.3702

SQL*Graph产品的资料部分是:

- SQL*Graph注释ORACLE部分 No.3006
- SQL*Graph用户指南ORACLE部分 No.3402
- SQL*Graph快速流览ORACLE部分 No.3806
- SQL*Graph快速调用ORACLE部分 No.3706

SQL*Calc产品的资料部分是:

- SQL*Calc注释ORACLE部分 No.3005
- SQL*Calc用户指南ORACLE部分No.3401
- SQL*Calc快速流览ORACLE部分 No.3805
- SQL*Calc快速调用ORACLE部分 No.3705

对每个支持ORACLE的操作系统, 也提供安装和用户指南方面的资料, 如:

- 对DEC VAX/VMS的ORACLE安装和用户指南ORACLE部分 No.1001
- 对 BM VM/SP的ORACLE安装和用户指南ORACLE部分 No.1003

根据版本5.1对于ORACLE所进行的某些修改:

请注意, 根据ORACLE版本5.1许多产品已改变了名称。当然涉及到版本5.1的资料反映出这些变化。下面的表显示出产品名称的改变。

注: SQL*Report (以前为ORACLE Report Writer) 的版本序号也从版本5.0变化到1.0。这变化反映了从ORACLE RDBMS产品中去掉SQL*Report。于是SQL*Report出版物将不再包括在ORACLE RDBMS的资料部分中。

老 的	新 的
ORACLE报告输出 (版本5.0)	SQL*Report (版本1.0)
pro*ORACLE	ORACLE包含OCI的接口
对预编译程序和pro*SQL预先保护条件	
pro*SQL和HLI	OCI (ORACLE调用接口)
ORACLE*Net	SQL*Net
SQL*Link	COPY命令
	SQL*Link在SQL*plus 资料中用于提供命令资料

使用这手册的约定

下面是对ORACLE RDBMS和新涉及产品中的资料的某些约定。

文件名 文件名是以大写字母形式出现, 如 NIT.ORA。出现下面情况, 文件名部分可以有变化, 如SGADEFx.ORA。

保留字和关键字 在实例和正文中, 这些字也是以大写字母形式出现的, 它们在ORACLE中有预定意义。

关键字名 关键字名以大写字母形式出现, 并且用方括弧括起, 如(RETURN)。

命令格式（句法）

命令 这字形用于辨认正文，其输入形式必须准确地表示如下：

SELECT*FROM

变量 变量是以斜体形式出现的，用户必须代入适当的值。

替换项 这替换选择总是通过垂直短线分离的。如果一组替换选择是需要项，则用大括号括起来；如果是选择替换项，则用方括号括起来（见下面需要项和选择项的符号约定）。

需要项 需要项是用大括号括起来，用户必须选择替换项之一。

选择项 选择项用方括号括起来。

注：方括号也可括起字的一部分，如

s[elect]...

缺省值 一个缺省值是被加下划线的，它经常表示做对同类型变量的选择。

重复项 一个省略符号表示任意数相同项。

手册中提供了适当的实例。

新的或修订材料 新的或修订材料用在页边上带短线来表示（译者注：我们为印刷方便，在新的或修订材料的相应段落加注释：“增加或修订”）

若下面的符号出现，总能影响到命令格式：

。	句号	；	分号
,	逗号	:	冒号
—	连字符号	=	等号

第一部分：PRO*FORTRAN预编译接口

本章提供PRO*FORTRAN预编译接口的完整描述。第一部分叙述PRO*FORTRAN程序总的结构，并且说明了PRO*FORTRAN语句的语法。其它部分包括错误处理、事务控制、指针、预编译程序的使用和命令选择，以及程序设计的指导。对于每一概念均使用大量例子进行说明。

第一章 PRO*FORTRAN简介

本章是对PRO*FORTRAN的一般介绍。讨论了它的使用，一般概念和定义，PRO*FORTRAN的程序部分，以及出现在PRO*FORTRAN程序中的语句类型。

SQL数据语言是一种非过程语言。就是说，语句的执行是相互独立的，与其前、后语句无关。在这一点上与其它语言如C，FORTRAN，COBOL，或PL/I比较，这些语言是被称为“面向过程的”，它们是基于“循环”、“分支”和“if/then”偶对。SQL语言是一种非常有力的语言，然而，它也存在一些限制。

特殊设计的SQL是一种非过程语言，在对这种语言的限制了解后，SQL的创建者明确把SQL语言结构设计成嵌入在一过程性语言之中，如FORTRAN语言中。程序设计者设计它的应用程序时，则可组合利用其SQL的最好特性及程序设计语言（宿主语言）的最好特性进行设计。这样写出的一些应用程序则比基于其它单独FORTRAN或SQL语言所写出的应用程序要灵活且强有力得多。

ORACLE关系数据库管理系统包括若干工具，程序员使用这些工具，在宿主语言中编写其程序以对ORACLE数据库中的数据进行存取。通常，这些工具是为C，FORTRAN，pascal，COBOL和PL/I等语言提供的。

PRO*FORTRAN是什么

pro*FORTRAN是可以处理SQL语句的ORACLE产品。pro*FORTRAN改造FORTRAN程序可把包含SQL语句的FORTRAN程序变成在ORACLE数据库中可进行存取和修改数据的FORTRAN程序。

通过预编译程序，PRO*FORTRAN把它的输入文件中的EXEC SQL语句变换成输出文件中适当的ORACLE调用。输出文件随后按正常的FORTRAN程序形式被编译、连接及执行。

把pro*FORTRAN（或类似产品，如pro*C和pro*PL/L）与ORACLE调用接口相比较，OCI（以前称HLI或pro*SQL）是对ORACLE数据库的调用接口，它允许用户直接在高级语言中如C，FORTRAN或COBOL中嵌入ORACLE调用。每个事务通过多个调用和

被称为“指针”的特殊数据结构的使用来完成。

PRO*FORTRAN的特点

PRO*FORTRAN编译程序的一些特点是：

- pro*FORTRAN调用比OCI调用更概念化，并且接口也较容易理解；
- 一个pro*FORTRAN调用可以被自动翻译为等效的若干OCI运行时间库调用，以减少程序设计时间；
- 一个程序能使用不同数据库的数据；
- 多个程序能单独预编译并一起执行。

一般概念

使用pro*FORTRAN工具程序员正常的程序设计处理过程，增加了额外的步骤，然而这增加的步骤可使得pro*FORTRAN工具为程序员做了大量的工作。编写和运行一个FORTRAN程序活动的正常顺序在图1中表明。

-
1. 编写一个包含FORTRAN源代码的程序。
 2. 编译此程序（得到一个输出目标文件）。
 3. 连接编辑此目标文件（得到一个可执行文件）。
 4. 为完成新希望的工作，运行该程序。
-

图1. 编写一个FORTRAN程序步骤

如果程序员使原来的FORTRAN源程序代码中包含 pro*FORTRAN 语句，则在开始时，将增加一个步骤，如图2所示。

-
1. 编写一个包括FORTRAN语句和SQL语句的程序。
 2. 使用pro*FORTRAN预编译该程序。得到一个输出FORTRAN源程序文件。
 3. 编译程序，得到目标文件。
 4. 连接—编辑目标文件，PRO*FORTRAN运行时间库（从属于操作系统），得到一个可执行文件。
 5. 运行程序，完成所希望的工作。
-

图2 编写pro*FORTRAN程序的步骤

pro*FORTRAN命令

运行pro*FORTRAN预编译程序的语法和选择，在“调用 pro*FORTRAN (PCC 命令)”中详细讨论。在那一章里你也可以找到，预编译一起运行的若干文件，以及同时使用pro*FORTRAN和OCI的信息。

FORTRAN语句和SQL语句混合使用

任一有效的SQL语句可以由FORTRAN程序来执行。虽然在 PRO*FORTRAN 程序中

存在一些要求的“部分”或语句和语句出现的“次序”，FORTRAN程序设计行可以出现在程序的任意地方（当然要遵循 FORTRAN 程序设计标准）。这些要求的部分在“pro*FORTRAN程序部分”那一节中介绍，并在其后的各节中较详细叙述。也要注意嵌入的 SQL 语句必须遵循一些FORTRAN语句格式规则，也就是这些SQL语句不能写在1到6列；在多行语句中的每一个续行，其续行符号必须写在其行的第6列。这个 pro*FORTRAN 版本不支持VMS关于续行允许<tab><digit><text>的约定，这是因它不是ANS 标准。对于pro*FORTRAN的词法扫描程序已经作了一些修改，使之比以前的pro*FORTRAN的版本能接受与ANSI标准更一致的语句。

命令前缀EXEC SQL

为了尽量减少SQL语句散布到许多不同宿主语言中去引起的语言上的困难，所有 SQL 语句前面都带EXEC SQL前缀。因此预编译程序的其中一个目的，就是把所有以 EXEC SQL开始的语句翻译为源语言代码对数据库的调用（假定是在这手册中叙述的FORTRAN）

命令前缀EXEC ORACLE

虽然大多数pro*FORTRAN语句前置EXEC SQL，但是，还有一些语句前置 EXEC ORACLE。许多SQL语句要求EXEC SQL前缀，也存在固定的要求EXEC ORACLE 命令的一个小的集合；对那些语句的列表和叙述可参见“EXEC ORACLE 选择”。加前缀 EXEC SQL的语句，与 BM嵌入SQL语句是不一致的，但对ORACLE预编译程序来说是唯一的。

SQL执行语句和说明语句

包括在pro*FORTRAN程序中的SQL语句分成两类：可执行的或说明性的。所有语句不管是可执行的还是说明性的，都要求加前缀EXEC SQL。

可执行的SQL语句是生成实际数据库调用的SQL语句。它们包括（但不限于）：数据操作语句（DML）、数据定义语句（DDL）和数据控制语句（DCL）。在可执行SQL语句执行之后，SQLCA（SQL通信域）包括一组返回码。一个作业的逻辑单位是从除CONNECT外的第一个可执行SQL语句开始的。因此在CONNECT，COMMIT或ROLLBACK WORK语句之后，碰到第一个可执行语句便开始新的作业逻辑单位。

说明性的SQL不产生代码并且对作业逻辑单位不产生任何影响，然而它们影响到由执行SQL语句所产生的代码。这时在程序中对最新的判定指针置“flags”位，说明语句不影响SQLCA。说明性SQL语句如图3所示。

SQL说明语句

```
EXEC SQL BEGIN DECLARE SECTION
EXEC SQL END DECLARE SECTION
EXEC SQL WHENEVER...
EXEC SQL DECLARE CURSOR...
EXEC SQL INCLUDE...
```

图3 SQL可说明语句

pro*FORTRAN程序的组成

一个pro*FORTRAN程序是由两个必须部分（应用头部和应用体）和一个可选择部分（错误处理例行程序）组成的。

- 应用头部（在“应用头部”部分叙述）；
- 应用体（在“应用体”部分叙述）；
- 错误处理例行程序。

应用头部定义变量和对pro*FORTRAN程序做一般预备工作。应用体基本上包括由SQL语句组成的FORTRAN程序逻辑单位，这些SQL语句是操作ORACLE数据的语句，比如INSERT或UPDATE语句。

下面一点很重要：在给定的一个源程序文件中只有一个程序（或子程序）可以包含SQL语句。

下一章，“应用头部”叙述应用序，并且包含短的pro*FORTRAN的程序实例：该程序被用来进入或退出ORACLE。

第二章 程序头部

本章叙述pro*FORTRAN程序头部。这一章的最后是短的程序实例，指明怎样定义进入或退出ORACLE的一个程序头部。

在每个pro*FORTRAN程序的开始部分是程序头部。它总是包含三部分：

1. DECLARE部分，命名宿主变量；
2. SQLCA数据结构的说明，通常由EXEC SQL INCLUDE语句来做（SQLCA提供错误处理）；
3. ORACLE数据结构的说明，通常由EXEC SQL INCLUDE语句来做（ORACLE提供错误处理）；
4. CONNECT语句，连接ORACLE RDBMS。

头部：DECLARE部分

在DECLARE部分，SQL语句中使用的所有FORTRAN变量被命名（“说明”）。

DECLARE部分的位置必须遵循FORTRAN说明语句的规则，也就是它必须在任何可执行FORTRAN语句之前。一个模块（子程序、函数）如果包含嵌入的SQL语句，则必须有DECLARE部分，不管它是否使用宿主变量。在后者情况下，DECLARE部分是简单的空。如果DECLARE部分省略了，则程序不能被预编译，并且你将看到下面信息：

Unable to generate descriptor in program unit line %u in file %s

DECLARE部分总是由下面语句开始：

EXEC SQL BEGIN DECLARE SECTION

并且以下面语句结束：

EXEC SQL END DECLARE SECTION

在这两个语句之间，只允许有说明宿主变量或说明指示符变量的说明语句。（两种变量将在后面有关节中定义）。

每个预编译单位（虽然一个程序可以包含若干独立预编译单位）只允许有一个DECLARE部分。

如果一个变量在FORTRAN程序的EXEC SQL语句中被引用，但是变量在DECLARE部分没有被说明，则在程序预编译时便出现下面信息：

Undeclared host Variable a at line b in file c

这里a是变量名，b是被使用变量的行号，c是使用变量的文件名。

主变量

在SQL语句和FORTRAN语句（“宿主”语言）中被引用的每一变量都被称为宿主变量。每个这样的变量必须在一个程序单位的DECLARE部分被显式说明，在这程序单位嵌入的SQL语句中，该变量被使用。变量的数据类型必须在DECLARE部分被说明，且要遵循

宿主语言的规则，变量的数据类型不必与 ORACLE 用于定义表中的数据类型相匹配。ORACLE 将在宿主语言数据类型和 ORACLE 数据类型之间执行一切可能的适当转换。

```
EXEC SQL BEGIN DECLARE SECTION
CHARACTER*20 UID
CHARACTER*20 PWD
INTEGER      DEPTNO
INTEGER      PEMPNO
CHARACTER*10 PNAME
EXEC SQL END DECLARE SECTION
```

图4 DECLARE部分的实例

在图 4 的说明中有三个宿主变量 (PEMPNO, PNAME, DEPTNO)，它们将出现在程序后面的 SQL 语句中 (在应用体中)：

```
EXEC SQL SELECT DEPTNO, ENAME
          INTO: DEPTNO, : PNAME
          FROM EMP
          WHERE EMPNO=: PEMPNO
```

主变量准则

一个主变量：

- 必须在 DECLARE 部分被显式说明；
- 在 SQL 语句中，前面必须加冒号(：)；
- 必须不是 SQL 保留字 (见附录 C)；
- 只能在使用常数的位置被使用；
- 可以有一个关联的指示符变量。

(指示符变量在下面部分讨论)。

注：因为主变量只能作为常数使用，所以主变量不能被用做表名。

FORTRAN 支持的数据类型

pro*FORTRAN 所支持的 FORTRAN 数据类型如图 5 所示。

VARCHAR*n	(参见“VARCHAR伪类型的说明和使用部分”)
BYTE var	单一字符
BYTE var(n)	n字节固定字符串
CHARACTER var(n)	n字节字符数组
CHARACER*n var	n字节固定字符数组
CHARACTER*n var(m)	m*n字节字符数组
CHARACTER var*n	n字节字符数组
COMPLEX var	8字节固定字符数组
COMPLEX*8 var	8字节固定字符数组
COMPLEX*16 var	16字节固定字符数组

DOUBLE COMPLEX var	16字节固定字符数组
DOUBLE PRECISION var	8字节浮点数
INTEGER var	缺省长度整数
INTEGER*2 var	2字节整数
INTEGER*4 var	4字节整数
LOGICAL var	2字节整数
LOGICAL*1 var	单一字符
LOGICAL*1 var(n)	n字节固定字符数组
REAL var	4字节浮点数
REAL*4 var	4字节浮点数
REAL*8 var	8字节浮点数

图5 pro*FORTRAN支持的FORTRAN数据类型

指示符变量

指示符变量是选择变量，与在DECLARE部分说明的宿主变量一一对应。他们主要用于处理Null（空值）。其次被用于存储个别字段返回代码，这些返回码指明了不是“空值返回”就是“字符字段截断”。

指示符变量也能被用于插入Null值。（见“指示符变量中使用返回值”）

指示符变量准则

一个指示符变量：

- 必须在DECLARE部分显式说明；
- 必须按 INTEGER*2形式来说明；
- 当在SQL语句中使用，其前必须置冒号（:）；
- 当在FORTRAN语句中使用，不必在前面加冒号；
- 必须不是SQL保留字（见附录C）；
- 在SQL语句中与它相关联的输入主变量必须在其前。

例如，在前面的查询中，现在我们增加两个指示符变量DPTNOI和NAMEI。

```
EXEC SQL BEGIN DECLARE SECTION
CHARACTER*20 UID
CHARACTER*20 PWD
INTEGER      DEPTNO
INTEGER      PEMPNO
CHARACTER*10 PNAME
EXEC SQL END DECLARE SECTION
EXEC SQL BEGIN DECLARE SECTION
INTEGER*2 DPTNOI, NAMEI
INTEGER DEPTNO, PEMPNO
CHARACTER*12 PNAME
```