

系列之二

ORACLE关系数据库管理系统

用 户 手 册

(七)

SQL * NET用户指南

79

《计算机技术》编辑部

目 录

第一章 SQL*Net介绍	(1)
什么是SQL*Net.....	(2)
什么是分布式处理.....	(2)
SQL*Net结构.....	(2)
利用SQL*Net进行联网操作的优点.....	(3)
SQL*Net功能特性概述.....	(4)
SQL*Net支持的环境.....	(4)
SQL*Net的支持策略.....	(5)
安装SQL*Net注意事项.....	(5)
第二章 SQL*Net结构	(6)
描述.....	(6)
终端——进程通信.....	(7)
任务——任务通信.....	(7)
SQL*Net通信规程要求.....	(7)
使用SQL*Net的数据库说明符.....	(8)
自动注册工具.....	(9)
SQL*Net的限制.....	(9)
第三章 利用SQL*Net进行数据库操作	(10)
利用SQL*Net使用Easy*SQL.....	(10)
LINK (连接) 选项.....	(11)
LINK屏面.....	(11)
选择节点与注册进入ORACLE.....	(12)
选择表.....	(13)
卸出和装入.....	(13)
源表和目标表.....	(13)
选择要传输的列.....	(14)
选选要传输的行.....	(14)
完成传输.....	(14)
传输错误.....	(15)
使用被传输的表.....	(15)
利用SQL*Net使用SQL*Plus.....	(15)
注册至SQL*Plus.....	(15)

CONNECT (联接)	(15)
COPY 命令	(15)
COPY 命令句法	(16)
参数	(16)
选项	(16)
示例	(17)
COPY 命令注记	(17)
列名序列	(18)
出错报告	(18)
宏替换	(18)
对SQL*Plus下开放事务的影响	(18)
内部选项	(18)
利用SQL*Net使用ORACLE编程接口	(19)
利用SQL*Net使用其它ORACLE实用程序	(20)
第四章 ORACLE异步规程	(20)
异步规程与3270/Irma通信规程	(20)
自动注册工具	(20)
建立自动注册交接文件	(21)
数据库标识	(21)
指定本地端口、异步和自动注册参数	(22)
本地端口参数	(22)
异步参数	(23)
自动注册参数	(23)
数据库标识实例	(24)
示例	(24)
交接对话定义	(24)
第五章 DECnet规程	(25)
使用DECnet规程	(25)
数据库标识	(25)
结构	(25)
第六章 TCP/IP规程	(27)
使用TCP/IP规程	(27)
数据库标识	(27)
结构	(28)
附录A 自动注册演练实用程序	(28)
使用自动注册演练实用程序	(29)
指定ALX命令	(29)
ALX命令	(30)
发送异步中止	(30)

调用命令行解释器.....	(30)
建立对话.....	(30)
显示对话.....	(31)
定义替换变量.....	(31)
执行对话.....	(31)
求助.....	(32)
开放端口.....	(32)
重取对话.....	(33)
存贮对话.....	(33)
踪迹开关.....	(33)
ALX对话实例.....	(33)
附录B 交接对话定义	(35)
交接对话示例.....	(37)

引 言

这本SQL*Net用户指南是专为那些希望利用ORACLE产品进行联网操作的用户而编写的，本手册所包括的内容适用于最终用户、应用程序员和系统管理员。

内容提要

本手册的主要内容简述如下。

第一章 SQL*Net介绍 本章给出SQL*Net一般描述，介绍利用ORACLE产品进行联网操作的优点，以及SQL*Net支持的功能特性。本章还包括了支持SQL*Net的运行环境一览。

第二章 SQL*Net结构 本章描述SQL*Net为了给ORACLE提供一种网络环境而采用的体系结构。本章介绍了所有运行环境共有的一些特性，例如数据库规范、自动注册（用于终端规程），以及由SQL*Net所带来的限制。

第三章 利用SQL*Net进行数据库操作 本章介绍如何通过ORACLE的三个接口，即Easy*SQL、SQL*和Pro的每一个接口来存取远程数据库。

第四章 异步规程 本章对如何使用异步规程及其辅助工具，自动注册提供全面的说明。本章还描述了如何指定作为数据库标识串的一部分的适当的异步和自动注册参数，并解释了自动注册交接文件的有关要求。

第五章 DECnet规程 本章全面介绍了如何使用DECnet规程。本章还解释了如何指定DECnet规程，以及如何改变通常所建立的DECnet连接的缺省值。

第六章 TCP/IP规程 本章介绍如何使用TCP/IP规程，还介绍了如何指定作为注册串的一部分的TCP/IP规程，以及如何剪裁这一规程。

附录A 本附录给出了如何使用自动注册演练实用程序(Autologin Exerciser Utility)的有关信息，这一工具允许用户在单一系统上建立、测试以及调试交接对话。

附录B 本附录给出了交接对话的格式。

第一章 SQL*Net介绍

本章作为SQL*Net的引论，介绍下述内容：

- 概述SQL*Net及其应用
- 说明使用SQL*Net的好处

- 概述SQL*Net的功能特性
- 目前支持SQL*Net的环境一览

什么是SQL*Net

SQL*Net (以前叫ORACLE*Net) 是这样的一个软件层, 即它与各种操作系统所支持的通信规程一起来为ORACLE关系数据库管理系统 (RDBMS) 提供分布式处理环境。这样, 网络中各类硬件和操作系统之间的界线实际上就被抹除掉了。

第5版以前的ORACLE体系结构允许应用只能在单个预先定义的网络位置上链接到单个预先定义的数据库。进一步说, 应用程序必须象数据库那样驻留在同一个CPU上。SQL*Net与ORACLE第5版 (RDBMS) 一起推出, 作为一个任选的软件产品, 允许在一台机器上运行的应用程序存取、操纵驻留在另一台机器上的ORACLE数据库中的数据。在这种情况下, 应用程序可以是由ORACLE公司所支持的一个实用程序, 例如SQL*Forms, 也可以使用ORACLE编程接口开发的一个应用程序。此外, SQL*Net1.0版允许应用程序借助多种通信规程动态链接跨越网络节点的多个ORACLE数据库。

那些熟悉ORACLE*Net第5.0版 (支持DECnet通信规程) 的用户应当注意, ORACLE*Net第5.0版与SQL*Net第1.0版的主要差别是: 后者支持ORACLE异步规程、3270PC\Irma规程和TCP/IP规程。ORACLE异步规程为客户与连网到下列主机的IBM-PC机的服务器之间提供基于终端/主机的通信: VAX/VMS、AOS/VS、VAX/UNIX、IBM/VM和IBM/MVS。虽然IBM-PC机是异步通信规程的最常用的主机, 但它也能用于让两个VAX系统或者两个IBM大型之间通信。3270PC/Irma规程允许IBM-PC与IBM/VM或者IBM/MVS之间链接。TCP/IP规程则为多种UNIX环境提供联网支持。

什么是分布式处理

分布式处理是不同网络节点上的多个程序, 通过该网络发送信息来协调它们的操作的过程。分布式处理环境中的用户必须能够辨识远程数据库的位置并必须通过该网络发送信息来启动远程节点上的某进程。可是, 分布式数据库代表用户来协调共享的存取, 数据的位置对用户实际上是透明的。

分布式处理环境通常并不支持跨越网络的接合点 (join), 也不支持在单一语句中访问多个节点的那些语句。

SQL*Net允许用户应用驻留在非数据库所驻留的一台计算机上, 它提供了将数据从网络上的一个节点传送到另一节点的手段。ORACLE应用可以用SQL*Net来提供有关分布式处理的特性 (存取远程节点的数据, 等等)。另外, ORACLE提供的分布式查询能力允许一个单独的SQL语句访问多个结点, 并且支持跨网的接合点。用户还可以使用允许跨越多个网络节点操作的ORACLE编程接口来编写应用。

SQL*Net结构

SQL*Net通过在ORACLE双重任务结构所允许的进程——进程通信中溶入多节点网络通信来进行分布处理。ORACLE RDBMS以双重任务结构为基础, 这一结构意味着用户任务 (或者应用) 作为一个独立于服务员任务 (ORACLE核) 的分立的操作系统进程来运

行。这两个任务借助于操作系统支持的进程间通信工具彼此通信。

SQL*Net被使用时,这种进程间的通信是发生在全网范围内而不是一台机器上的两个进程之间,这种进程间的通信被称为顾客/服务员操作模式。在这种模式中,顾客是一个应用程序或者工具,服务员是ORACLE RDBMS。在传统的单一CPU环境中,顾客和服务员进程在同一台机器上运行,数据库文件也存储在这台机器中。SQL*Net允许顾客程序运行在非核(以及数据库)所驻留的计算机上。顾客程序通过跨网发送信息请求服务员程序的服务。

有关SQL*Net使用顾客/服务员操作模式来执行分布式处理的详细说明请参见第二章。

利用SQL*Net进行联网操作的优点

ORACLE软件最重要的特性之一,是不论运行环境如何它们都具有相同的功能特性。由于这种通用性,ORACLE便非常适合于联网环境。

利用SQL*Net进行分布式处理的基本优点,也许在于充分地利用现有的硬设备。由于SQL*Net允许多种环境中的ORACLE产品彼此跨网通信,数据库管理员具有为一个专门的应用灵活地选择最适宜的硬件的自由。例如,SQL*Net允许运行在PC机上的SQL*Calc存取中央系统(小型机或者大型机)的数据库。这样便利用了这两种计算机的长处,SQL*Calc利用了PC机的快速响应时间;数据库利用了小型机或者大型机的高效的数据共享技术、快速的I/O设备以及廉价的处理能力。

其次,联网非常适合于数据库环境,因为它提供了对于地处在网上遥远处所有地方上的数据的存取方法。例如,利用SQL*Net可以在保持为公共处理所需数据装入或卸出大型中央计算机的能力的同时,把部门性地区性的计算处理请求分布在小的本地计算机中完成。举一个具体例子,比如某一地区销售部门将一台MICROVAX与大的远程VAX机联网,将每天的销售额装入大型机数据库,并将库内的库存和价格数据卸出。这个例子也说明了利用SQL*Net能够实现数据的高效分布,从而实现高效的数据处理。

联网的另一个优点是,它比独立的计算机系统更能适应低成本高性能的功能扩充。例如通过将一些PC机并入一个与大型机连接的网络来提高处理能力,比起购置一台新的大型计算机要经济得多。联网使得廉价的较小的计算机也能分享功能强大的大型机系统的处理能力。

采用SQL*Net的环境的再一优点是,可以跨越网络来存取用ORACLE生成的应用,这样就精简了应用开发的工作量。由于用ORACLE生成的应用非常灵活,一次单独的开发工作就可产生一个对网上每一系统都有相同操作和相同用户接口的单独的应用。这些应用现在可以在网上每一系统中运行,并可存取任何ORACLE数据库。用户还可以使用包括SQL*Net功能的ORACLE编程接口来编写高级语言程序,甚至可以编写跨越多个网络结点的高级语言程序——这是其它环境通常不具备的功能。

更进一步,通常只在本地数据库才具备的数据库的安全性和完整性,已在全网范围内实现了。

总起来说,使用SQL*Net具有如下优点:

- 更有效地利用处理能力
- 数据处理可被分布在远程的地理位置上
- 更有效地分布数据处理的工作量
- 更经济地提高处理能力
- 精简应用开发的工作
- 实现全网范围内的数据的安全性和完整性

以上是SQL*Net所提供的众多优点的一部分, 下面将要逐一叙述SQL*Net的功能特性。

SQL*Net功能特性概述

SQL*Net具有下列功能特性:

- 在网络上任意一处的SQL*Net用户可以通过直接向远程数据库登录注册, 联接到该网上的任何其它ORACLE数据库。

- SQL*Net允许存取多个数据库, 甚至是同时存取。这样, 用户可以在一段时间内使用本地数据库, 其余时间使用远程数据库。从而, 可以从一个远程数据库抽取数据, 在本地数据库内进行处理, 然后再把它送回到远程数据库。可以在影响到另一个数据库的事务处理中使用一个数据库的数据。数据库——数据库拷贝服务即为这类操作的例子, 比如SQL*Plus COPY命令提供的有关服务, 又如在中央数据库进行更新的同时可在本地存贮静态数据, 使数据库有效。

- ORACLE异步规程和3270规程均能提供自动注册工具, 不论何时提出联机请求它都可以自动注册到远程主机上。

- SQL*Net支持同种机联接(在相似环境之间)。例如, 可以连接两台VAX计算机, 一台作为顾客而另一台作为服务员。

- SQL*Net也支持异种机联接(在不同环境之间)。例如, 可将一台IBM-PC机连接到一台VAX, IBM-PC作为顾客VAX作为服务员。

- 同种机联接和异种机联接可以同时建立。

- SQL*Net的工作独立于它所支持的规程, 因此, 无论对于何种规程它的功能都是一样的。

- 用户可以向本地的, 远程的或缺省的数据库注册。

- 在实用程序初始化后SQL*Net动态重定向注册。这样, 可以在对同一个ORACLE应用进行数据库注册时, 从一远程主机改到另一个远程主机。

- 当使用SQL*Net时, ORACLE提供的安全和自动恢复功能将在全网范围内予以扩展和保持。

- 借助ORACLE核提供的分布式查询功能, 可以用单个SQL语句访问多个节点。

SQL*Net支持的环境

表1说明了SQL*Net第1.0版所支持的顾客、服务员以及规程的配置情况。

表1 由SQL*Net第1.0版支持的配置

顾客 (Client)	服务员 (Server)	规 程
PC/MSDOS	VAX/VMS	异 步
PC/MSDOS	DG/AOS/VS	异 步
PC/MSDOS	多种UNIX	异 步
PC/MSDOS	IBM/VM/CMS	异步、3270
PC/MSDOS	IBM MVS/SP	异步、3270
多种UNIX	多种UNIX	TCP/IP
VAX/VMS	VAX/VMS	DECnet
PC/MSDOS	VAX/VMS	异 步

SQL*Net的支持策略

ORACLE公司支持网络环境策略的基本原则是，支持两个运行环境间的“天然的”同机种规程。

目前支持的“天然的”同机种规程举例如下：

VAX/VMS至VAX/VMS (经由DECnet)

支持策略的另一原则是支持不同环境间的异种机通信。ORACLE目前支持RS232 异步TTY通信 (利用ORACLE异步规程)，在这种情况下应用端模拟一个异步TTY终端，数据库端提供一个异步端端口。

这类异种机通信的方式举例如下：

PC/MSDOS与VAX/VMS

PC/MSDOS与VM/CMS

PC/MSDOS与UNIX

ORACLE还支持连接3270通信设备的同轴电缆 (利用3270PC/Irma规程)。在这种情况下应用端模拟同步 (BSC或SNA) 3270终端，数据库端提供一个同步3270端口。

这类异种机通信的方式举例如下：

XT/PCDOS与VM/CMS

XT/PCDOS与MVS/SP

ORACLE还利用TCP/IP支持以太局部网络，在这种情况下同种机连接和异种机连接都能支持，例如：

VAX/UNIX与Pyramid/UNIX

XT/PCDOS与VAX/VMS

ORACLE公司还准备支持那些打算开发支持SQL*Net的专用规程的用户。

安装SQL*Net注意事项

SQL*Net作为ORACLE RDBMS的一部分，没有单独的安装步骤。在操作手册中介绍

了一些特殊操作环境中应考虑的特殊情况，比如指定缺省数据库或缺省规程。

第二章 SQL*Net结构

本章描述SQL*Net的结构，包括下述内容：

- 用于支持分布式处理的双重任务通信技术
- 顾客/服务员通信模式
- 终端——进程通信
- 进程——进程通信
- 支持SQL*Net所需的通信规程原语
- 数据库注册
- 自动注册工具
- SQL*Net的限制

描 述

SQL*Net使用的通信技术，以ORACLE在某些操作环境，如UNIX，中使用的双重任务结构为基础。通常，双重任务是指在同一机器上的两个进程之间进行的通信。这些进程相互之间利用操作系统支持的进程间通信工具进行通信。SQL*Net也使用这一技术，但是这种通信发生在驻留在两个不同机器上的两个进程之间而不是驻留在一个CUP上的两个进程间这种双重任务结构形式被称为顾客/服务员通信模式。

虽然，即使ORACLE在一个单机环境（应用作为前台进程，ORACLE核作为后台进程）中运行这种通信模式也适用，但在网络环境中它显得更加适用。在网络环境中，两个分别运行在不同机器上的任务利用SQL*Net支持的多种通信规程进行通信。其中一个作为顾客，而另一个作为服务员。SQL*Net的顾客是应用程序，它可以是任何ORACLE应用，或是用户使用ORACLE开发的应用。SQL*Net的服务员是RDBMS（ORACLE核）。

在SQL*Net术语中，主机是指驻留着数据库并运行ORACLE核从而支持服务员的机器。重要的是应注意，有些机器比如VAX既可作为顾客也可作为服务员，而另一些机器比如PC则只能作为顾客。只有用作服务员的机器才可作主机。

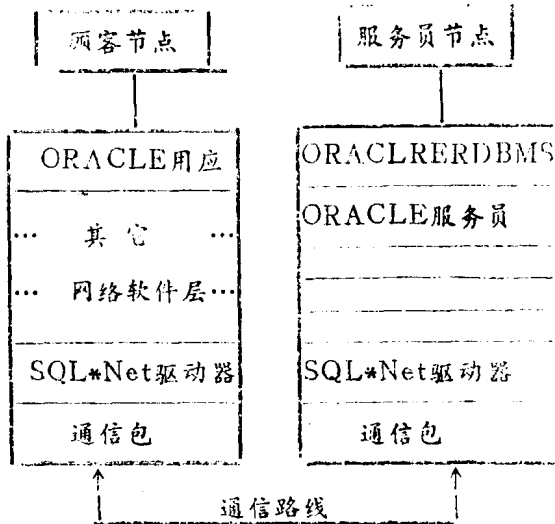
顾客可以是单用户系统（比如PC），而服务员却必须是多用户系统（比如VAX）。下面的框图表明了顾客与服务员间的可能的组合：

机 种	顾 客	服 务 员
PC	永远可以	不行
小型机	可以	可以
中大型机	可能	通常可以

图1描述了顾客/服务员通信模式

SQL*Net支持两种方式的顾客/服务员通信模式：终端——进程通信和任务——任务通信。下述各节解释这两种通信方式。

图1 顾客/服务员通信模式



终端——进程通信

当以终端——进程方式通信时，一个进程（终端）作为一个哑终端工作。在PC机上运行的程序模拟终端的行为以便与远程主机上运行的程序通信。SQL*Net支持两种具备终端模拟能力的通信规程：ORACLE异步规程和3270PC/Irma规程。当采用其中任意一个时，远程主机即把顾客当作一个哑终端看待，首先必须在顾客与服务员之间传递信息，然后就模拟任务——任务通信。

第四章提供了使用这些规程的说明。

任务——任务通信

任务——任务通信与进程——进程通信完全相同，在这种情况下，两个进程相互协作以确保可靠的数据传输。与终端——进程通信不同的是，两个任务均把对方看作是一个智能程序。SQL*Net支持的DECnet规程都能进行进程——进程通信。

DECnet网络服务规程（也叫做NSP）是DECnet体系结构的一部分，它提供了支持该结构的任意两个系统间的任务——任务通信。SQL*Net将此用于VAX/VMS系统间的通信。第五章给出了使用DECnet规程的说明。

TCP/IP通信规程提供了UNIX环境下的任务——任务通信。第六章给出了使用这一规程的说明。

SQL*Net通信规程要求

SQL*Net软件被设计成可以容易地移植到各种主机。事实上，SQL*Net只要求协同工作的通信规程一方必须提供五个低级原语。对于异步规程和3270PC/Irma规程的这些原语是由ORACLE提供的。

为了支持SQL*Net，基础通信规程必须提供执行下述功能的原语：

- 联接远程系统
- 断开与远程系统的连接
- 发送信息
- 接收信息
- 中断联接

使用SQL*Net的数据库说明符

本节介绍了在使用SQL*Net时如何指定一个远程数据库。讨论特定规程的章节则介绍当使用某个特定规程时如何指定一个远程数据库。

应当注意，安装期间应设定远程数据库的缺省值。详细情况请参见特定操作系统的操作和安装手册。您要操作的本地或远程的数据库，不论是在使用用户编写的应用还是ORACLE应用（比如SQL*Plus），为了指定您要操作的本地或远程数据库您都必须在注册时指定数据库名，有关的步骤依赖于应所涉及的程序或接口，而每一个程序或接口的情况文件均有介绍。在大多数情况下，数据库说明符(database specification)可以跟在用户名/通行字符串之后，并以字符“@”作为分隔符。例如，下面是一个合法的注册名：

scott/tiger@780VMS

对那些允许通行字单独输入的程序来说，通行字可从注册名中省略，例如：

scott@D:780VMS

正规的句法形式是：

用户名[/通行字][@数据库-说明符]

数据库说明符含有使SQL*Net定位至交换时使用的数据库所需的信息。这一说明符的多数格式取决于所用的通信规程，它可包括：路径、鉴别信息以及数据库所驻留的节点名。说明符有一部分独立于通信规程，这就是驱动器前缀，它决定使用哪个规程软件(驱动器)。因此，可将数据库的说明符再细分为如下的句法形式：

[驱动器前缀:] [数据库标识]

每个通信规程都规定了一个不相同的驱动器前缀，可以由特定系统上的应用来使用。被安装的通信规程集合（以及由此而允许的前缀）是在安装ORACLE时设定的，下面是SQL*Net支持的前缀：

前缀 驱动器将连接到：

- S: 本地系统，在单个任务中直接使用过程调用（数据库标识是忽略的）。单个程序每次只允许一个实际的直接联接。
- D: 一个远程系统，利用DECnet网络服务规程（远程系统在数据库标识中指定）。允许多个并发联接，甚至对同一主机也可这样做。
- A: 指定ORACLE异步规程。
- T: 指定TCP/IP规程。

如果没有指定驱动器前缀（或是所作指定与已知的驱动器前缀不匹配），则将使用缺省的驱动器。共有两种缺省且都可以在安装时设定。至于选择哪一种则根据是否指定了数据库标识(database-id)而定。在大部分VAX系统中，如果没有指定数据库标识则单个任务（

S:) 驱动器是缺省的, 否则DECNet (D:) 驱动器是缺省的。

安装ORACLE的人员可以选择要安装的驱动器。现举几例予以说明。以下例子中780 VMS是网络节点名。注意在第三例中“780VMS”是被忽略的。

scott/tiger	直接连至本地数据库
scott/tiger@s:	直接连至本地数据库
scott/tiger@s:780VMS	直接连至本地数据库
scott/tiger@d:780VMS	通过DECnet网络服务规程连至780VMS
scott/tiger@780VMS	通过DECnet网络服务规程连至780VMS

注意数据库标识是由通信驱动器为相应的规程解释的。这种解释的更多细节请参见有关特定规程的章节。

自动注册工具

这部分给出了只是在使用终端——主机规程, 比如ORACLE异步规程等时才有用的一些信息。

为了进行顾客与服务员之间的通信, 顾客必须请求与主机的联接。为了方便这一进程, SQL*Net支持的终端——进程规程 (ORACLE异步规程和3270PC/rma) 提供了一个不论顾客系统何时要求联机都可在远程主机自动注册的工具。为了完成这一任务, 自动注册工具自动地在远程操作系统上注册并引导命令行解释器开始服务员进程, 这样就不需要用户通过终端模拟器对远程主机人为地注册。

自动注册支持顾客 (例如PC) 与主机之间的专线联接和拨号线联接, 并可处理通过一智能调制解调器设置电话线联接的所有交接需求, 包括在远程操作系统上注册和启动SQL*Net服务员。

该工具使用包括在一组“交接文件” (handshake files) 中的对话来建立与远程主机的联接。这些交接文件是由用户建立的, 这些文件包含终端用户与远程主机间通常的“对话”的表示。其内容基本上是一系列信息, 其格式为: “发送‘这个’, 并希望送回‘那个’”。参数机制提供了电话号码、用户名、通行字等的替换能力。

一旦建立好了交接文件, 就可通过直接将相应于交接文件的文件名前缀指定为数据库说明符的一部分来启动注册工具。请参阅SQL*Net支持的各种规程的有关章节以获得如何完成这些工作的信息。

SQL*Net的限制

这部分列出了一些影响SQL*Net使用的限制。

- 虽然一个应用可对多个网络节点进行存取, 但单一的事务或工作单元只能对一个节点起作用。例如一个SQL语句可以访问一个且只能是一个数据库。然而利用ORACLE核提供的分布式查询功能, 可以实现一条SQL语句对多个节点的数据存取。

- 实用程序IOR、ODS和AIJ不能通过SQL*Net (注*) 来运行, 只能在装有ORACLE的同一机器上运行。

- 对分布式事务不提供支持。可以进行多场地更新, 但每次更新应当分别提交 (commit) 或者回退 (rollback)。

- 用户必须将SQL语句指向含有数据的节点; 不存在查询路径。

注*, 原文作ORACLE*Net—译注

第三章 利用SQL*Net进行数据库操作

本章解释如何通过各个ORACLE产品接口来存取远程数据库，给出了如何向远程数据库注册以及如何执行相关操作的具体说明。

本章分为下述部分：

- 利用SQL*Net使用Easy*SQL
- 利用SQL*Net使用SQL*Plus
- 利用SQL*Net使用ORACLE编程接口
- 利用SQL*Net使用其它ORACLE实用程序

利用SQL*Net使用Easy*SQL

本节涉及下述内容：

- LINK（连接）选项
- 连接选项的屏面
- 选择节点并注册进入ORACLE
- 选择表
- 卸出和装入
- 源表和目标表
- 选择要传输的列
- 选择要传输的行
- 完成传输
- 传输错误
- 使用被传输的表
- 其它传输选项

本节解释如何利用Easy*SQL的LINK选项来使运行在两台不同机器上的两个ORACLE系统之间相互联接并交换信息。

一般情况下，当用户在个人计算机上运行ORACLE，并希望与中央的主机或小型机系统交换数据时，LINK是有用的。例如，假设公司的全部销售信息保存在中央计算机上的ORACLE数据库中，您可能需要卸出（download）关于本地区的数据，并使用您的PC机进行数据查询、分析、制表和制图。或者您可能需要自己的PC机上输入本地区的销售数据，并将它们装入（upload）中央系统。

无论用户联接何种计算机，都可以将一个系统上的表或视图的一部或全部传输到另一个系统上的表中。Easy*SQL将管理全部的通信过程，用户只需要进行数据库注册，并指明打算传输的数据以及传输的地点。

注意：本节描述了对应于Easy*SQL第2.1版的LINK选项细则，该版的Easy*SQL LINK则作为ORACLE RDBMS第5.0或以上版SQL*Net的一部分，Easy*SQL第2.1版采用的技术与以前的版本不同。如果用户只有Easy*SQL第2.0版，那么有关使用该版本的LINK选项的系统需求以及用户必须完成的其它步骤，请参见《Easy*SQL用户手册》的附录D。

为了使用Easy*SQL第2.1或以上版的LINK选项，用户系统和远程计算机系统都必须作为SQL*Net的一部分。这意味着：

- 两个系统必须经直接的数据线或经电话线相连。
- 两个系统均须安装ORACLE RDBMS第5.0或以上版。
- 两个系统均须安装SQL*Net。
- 两个系统均须安装相互匹配的SQL*Net通信规程。

LINK (连接) 选项

为了在你的ORACLE数据库与另一个数据库之间传输信息，用户需从Easy*SQL ADMINISTRATION菜单中选用LINK选项，参见图2。

图2 选用LINK选项

EASY*SQL	
Q U E R Y	
R E P O R T	
E D I T	
C R E A T E	
G R A P H	
EXTENDED	
ADMINISTRATION <—	BACKUP
C U S T O M	EXPAND
E X I T	REMOVK
	LINK
	TRANSLATE

Establish link to transfer data to another computer. RETURN to choose0-Hint 1-Help 7-List 8-PrevFld 9-Clear RF3-DO -- Reset,=ShowError PF4=Quit

LINK屏面

无论您的Easy*SQL版本如何，LINK屏面都是完全一样的。LINK屏面的方框允许用户进行如下工作：

1. 选择远程计算机（主机）的网络节点。
2. 选择远程计算机的ORACLE数据库并且注册。
3. 选择传输信息的方向和所需使用的表。
4. 选择要传输的列
- 50 选择要传输的行。

大体说来，在LINK屏面上用户要进行ORACLE注册并要构造一个查询以抽取要传输的数据。因此，用户填写LINK屏面与填写Easy*SQL的QUERY（查询）屏面非常相似。

图3表示为卸出远程数据库中最近30天内受雇职员的记录，用户应如何填写LINK屏面。

图3 填写LINK屏面

choose how transfered rows will be merged with existing rows (box appears only if target table exists)

network node on remote system

driver code

ORACLE login on remote system

password on remote system

Easy*LINK

Host:

Host Table:

Local Table:

Column:

Enter conditions each row must meet:

WHERE

(OR BETWEEN LIKE IN NOT UPPER DATE IS ! * (* >)

Example: SAL > 1000 OR JOB = 'CLERK'

0=Hint 1=Help 7=List 8=Prevfid 9=Clear PF3=DO --Reset --ShowError PF4=Quit

From host DBMS

To host DBMS

Table Exists

APPEND REPLACE

COLUMNS

EMPNO
ENAME
JOB
MGR
HIREDATE

version 2.1 and later only

+Hidden entry: scott:central-pers

database file on remote system

选择节点并注册进入ORACLE

LINK屏面的头两个方框允许用户选择远程计算机网络节点并向那里的ORACLE数据库注册。在第一个方框中，填写您在远程系统上的ORACLE用户标识（ID）。如果您使用的是Easy*SQL第2.1或以上版，应按下述步骤填写用户标识：

1. @
2. 一个驱动器码（比如a）
3. :
4. 远程系统的SQL:Net节点名（比如central）

5. —

6. 远程系统用的数据库选择文件 (比如Pers)

例如, 用户可以输入scott@d:central-pers以使用DECnet驱动器联接名为central的远程系统, 并作为scott用户向pers数据库注册。

网络节点名必须在SQL*Net里定义, 如果操作系统需要的话, 还应包括注册信息, 驱动器码也必须在SQL*Net里定义。数据库选择文件必须能由远程系统在用户的当前目录下运行。

如果用户使用Easy*SQL第2.0版, 那么在第一个方框中除ORACLE用户标识外不要输入其它信息。这是第2.0版与第2.1版关于LINK屏面的唯一差异。

不管何种版本, 在第二个方框中应输入用户的远程数据库ORACLE通行字。通行字将不在屏上显示。当用户按“Next Field”功能键后, Easy*SQL验证所输入的ORACLE注册帐号。如果用户标识和通行字不是有效的, 您还有机会重新输入它们。

选择表

利用LINK屏面上随后三个方框, 用户可以选择:

- 是将信息传到远程主机, 还是自远程主机接收信息
- 含有要传输信息的源表
- 接收信息的目标表

图3告诉Easy*SQL从远程ORACLE数据库的EMP表中接受信息, 并在用户本地数据库的EMP样本表中加上这些信息行。

卸出和装入

从远程主机中卸出信息时, 为指明传输方向应选用多项选择方框中的From Host DBMS选项。这也是缺省的选择。

将信息装入远程主机时, 应选择TO Host DBMS选项。表方框的标号将会改变以反映源表和目标表的位置。

源表和目标表

第一个表输入方框定义为源表, 标号为From, 它含有用户目前想要传送的信息。应输入源表名或者从参考方框 (reference box) 中选择一个名字。虽然用户不能在LINK屏面上输入多个源表, 但可利用一个视图作为源表 (参见《Easy*SQL用户指南》第10章)。因此, 如果用户希望从多个表传送信息, 首先应建立一个由它们组成的视图, 然后在表输入方框中输入该视图名。

第二个表输入方框定义为目标表, 标号为TO, 它用于接收信息。注意传送的信息最后总是到达一个表中, 即使源表是一个视图也罢。

如果用户输入一个新的目标表名, Easy*SQL将利用来自源表的列的定义, 自动地建立它。

如果用户输入一个现存表名, 用户将看到一个多项选择方框, 允许您决定是APPEND (添加) 一些新行到旧的表行中还是REPLACE (替换) 它们。APPEND要求两个表的列结构是相容的, 这意味着被传送的全部列均须在目标表中存在, 并且有相容的数据格式。

如果用户输入一个现存的视图名, 用户将看到同样的APPEND/REPLACE方框。然而由于这些信息必须送入一个表中, 用户就只能选择利用同名的新表来REPLACE (替换)