

## 第15章 Oracle 8i概述

本章要点：

在Oracle 8i中的 Java能力

因特网文件系统

Oracle interMedia

Oracle WebDB

Oracle 8i的可用性和可恢复性

安全性

其他特性

Oracle RDBMS的最新版本Oracle 8i是在企业内和因特网上开发、布署和管理应用高移动性和可伸缩性的数据库。它被设计成用来支持很大和高容量的事务处理和数据仓库应用。

Oracle 8i最主要的新特性是在数据库中包括 Java和因特网能力。与数据库服务器集成的是称为Oracle JSever的Java虚拟机 ( JVM ), 因此开发者现在能够用 Java、PL/SQL、C或OCI创建应用。它使开发者能够用Java编写存储过程、数据库函数或触发器。因为Java是平台独立的, 因此用它开发的应用能够很容易地部署在任何支持 Java的服务器中, 而无需重写和再编译。Oracle 8i的另一个主要特性是因特网文件系统 (iFS), 它使数据库能存储和管理关系和非关系数据文件。Oracle 8i还包括能够管理和访问多媒体 ( 视频、音频、图像、文本、空间 ) 的 Oracle interMedia。Oracle 8i还包括Oracle WebDB, 这个Web开发环境使开发者能建立使用标准的Web浏览器和Oracle 8i数据库的动态的、数据驱动的 Web网站。

### 15.1 Oracle 8i 中的Java能力

图15-1说明了Java虚拟机结构。在数据库中有 Java使开发和部署因特网 /intranet应用变得更容易、更快、代价更少和更高效。Java程序语言使开发者能利用容易获得的大量开发工具。与C或C++这样的语言相比, Java代码在数据库服务器中执行是安全的, 因为它不能操作指向内存位置的指针。Oracle 8i Java 虚拟机的内存管理器自动分配和释放内存, 因此它使在数据库中内存越界所引起的数据损坏问题减至最小。

在Oracle 8i之前, 没有企业级的用于开发大规模应用的 Java服务器集成到RDBMS中。许多应用服务器和数据库服务器厂商使用来自 JavaSoft (JDK) 的Java虚拟机来运行服务器端Java应用。JavaSoft中Java 虚拟机被开发作为客户端虚拟机运行在单用户环境, 它不提供在企业范围内应用需要的高可伸缩性。Oracle的Java 虚拟机与数据库有很高的集成度来满足关键任务应用的需要。Oracle 8i Java虚拟机包括集成的Java数据库连接 ( JDBC ) 驱动器SQL翻译器并提供如下特性: 高性能、可伸缩性、高的可用性、管理能力和符合标准以及 Java运行支持。Java虚拟机符合JDK 1.1.6特性并支持所有标准的JDK库。

Java虚拟机有足够的内存管理能力, 对于一般的 Java会话减少用户内存需求到 80~150KB

之间，它与数据库内核运行在同一进程和地址空间，共享大量内存堆并为优化性能进行直接访问数据库的缓冲区。因为 Java 虚拟机从数据库的 SGA 中分配共享内存池，所以为适应 Java 虚拟机，应设置较大 SGA。能够扩展到几万个并行的 Oracle 多线程服务器结构（MTS，从 Oracle 7 版本开始提供）是 Java 被集成到服务器的地方。MTS 基于一种结构体系，即数据库监听器路由用户连接到一组与处理数据库连接的服务器进程交互的调度程序。Oracle 8i 的目的是在因特网应用中支持大量并行用户。Java 虚拟机被设计成更快速、更安全和更可靠地运行 Java 应用。由于 Java 虚拟机被集成到数据库，它可跨所有 Oracle 支持的硬件和操作系统平台。因此实现 Java 的承诺：“一旦编写，任何地方皆可运行”。

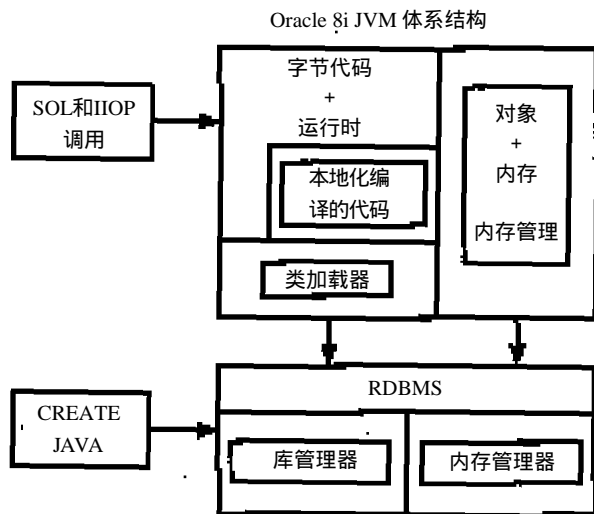


图15-1 Oracle 8i Java虚拟机体系结构

### 15.1.1 Java虚拟机的主要部分

Java 虚拟机的主要部分如下：

库管理器提供方便地装载、存储和管理在数据库中的 Java 程序。用户导出或导入 Java 对象能用三种形式：源、二进制和资源（或归档）。

类装载机定位、装载和初始化本地 DBMS 存储的 Java 类以响应来自 Java 虚拟机的请求。它读取 .class 文件信息，产生 Java 虚拟机运行 Java 代码所需要的所有数据结构。

内存管理器负责自动地管理 Java 虚拟机的内存堆和有效地分配和收集对象内存。内存被分配为称为对象内存的标准大块。每一个对象内存是作为单一单元被分配和收集的一个内存池，在每个单元内所有对象共享同一生命时间和格式。

Java 至 C 编译器（NCOMP）是一个嵌入 Java 虚拟机的本地编译器。它解释标准 Java .class 二进制文件，产生能被编译成 C 可执行文件的特殊 C 程序。

Java 存储过程和触发器能够被 Java 本地编译器编译并存储在数据库中作为与 PL/SQL 存储过程类似的数据对象。用 Java 写的过程能够调用 PL/SQL 写的过程，反之亦然。因为存储过程在数据库中执行，所以它们使在应用和数据库之间的网络业务量最小化。你现在可以在你传统使用 PL/SQL 的所有地方中使用 Java。整个 Java 存储结构在图 15-2 中说明。

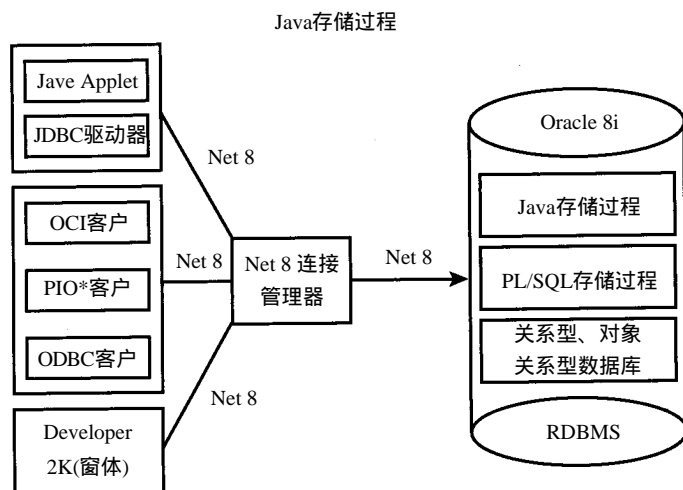


图15-2 在Oracle 8i中Java存储过程

### 15.1.2 编写Java存储过程

编写Java存储过程要求的步骤如下：

- 1) 编写你的Java代码。
- 2) 装载和解决它。
- 3) 发布代码。
- 4) 运行它。

首先，编写和测试你的Java程序，然后把它装入到Oracle 8i JVM中并解决所有引用。装入到数据库的Java程序被存储在数据库概要中作为库单元，它与被存储的PL/SQL程序单元类似。

下列例子是一个称为 student.java 的简单Java程序，它在计算机屏幕上打印一行“ student List ”：

```
public class student {  
    public static void main (string args[]) {  
        system.out.print ("Student List");  
    }  
}
```

一旦你编译并运行这个简单例子，你将知道你的环境已正确地建立了。

你能用两种方法装入Java源代码、二进制或资源文件。一种方法是在SQL\*Plus中用新的DDL命令：

```
"CREATE JAVA {SOURCE | CLASS | RESOURCE}"
```

使用这个语句你能从一个二进制文件或通过使用在数据库中的LOB列导入Java代码。

第二种方法是借助Oracle 8i提供的LOADJAVA工具。这个工具自动装入Java到数据库中。它装入Java文件到数据库中名叫 create\$java\$llob\$stable 表中，然后运行SQL“CREATE JAVA ...”命令。还有用于从数据库单元中删除Java程序的DROPJAVA工具。

当你指定解决选择时，LOADJAVA使用SQL ALTER JAVA CLASS ... RESOLVE语句来解决现在被装载的Java类外部引用；否则ALTER JAVA语句在运行时隐含执行。

下列是一个用于解决一个Java类的语法例子。它装入exam.jar文件到用户SCOTT的概要中，然后解决外部引用：

```
loadjava -user scott/tiger -resolve exam.jar
```

loadjava也能采用列出一项或多项解决项目，每项有名字说明和概要说明组成，它的查找按列表顺序。

以下是一个使用LOADJAVA工具用RESOLVE特性来装载student.class文件到在数据库MYSID中的用户PETER概要中。它在PETER概要中查找以home.java开头的类名，然后在PUBLIC概要中查找：

```
loadjava -user peter/page@tysons1:5521:mysid
         -resolver '("/home/java/*" PETER) (* PUBLIC))' student.class
```

如下是在操作系统级使用二进制文件来装载java代码的例子：

```
/* Create a directory on the server's file system. */
SQL> CREATE DIRECTORY b_dir
      2 as '/home/user/oracle8/student';
Statement processed.

/* Then load the Java class file using the "CREATE JAVA CLASS ..." command. */
SQL> CREATE OR REPLACE JAVA CLASS
      2 USING BFILE (b_dir, 'student.class');
Statement processed.
```

在Java程序被装入到数据库之后，代码需要被发布。这用SQL注册程序来实现，但仅有顶层Java项目点需要被注册。顶层Java项目点是Java类或用SQL初始化调用SQL的方法。Java代码通过使用调用说明被发布。一个PL/SQL子程序被说明为由一个Java方法实现。

下面是一个对类employee.emp的app\_bonus Java方法的调用说明的例子。

```
SQL> CREATE OR REPLACE PROCEDURE A_BONUS
      (C_SAL IN NUMBER, BONUS NUMBER)
      2> AS LANGUAGE JAVA
      3> NAME 'employee.emp.appr_bonus (float,float,float)';
      4> /
      5> Statement processed.
```

最后一步时通过SQL调用运行你的Java代码。有两种方法做这一步：

一种方法时使用CALL语句：

```
SQL> VARIABLE CSAL NUMBER
SQL> VARIABLE BONUS NUMBER
SQL> EXECUTE :CSAL := 20000;
Statement processed.
SQL> EXECUTE :BONUS := 100;
Statement processed.
SQL> CALL A_BONUS (:CSAL, :BONUS);
Statement processed.
```

另一种方法是从SQL查询中调用Java存储函数：

```
CALL-spec for s_grade Java method of class student.stud -

SQL> CREATE OR REPLACE FUNCTION SGRADE (GRAD IN NUMBER)
      RETURN VARCHAR2
      2> AS LANGUAGE JAVA
      3> NAME 'student.stud.s_grade (float) return java.lang.String';
```

```
4> /  
Statement processed.
```

以下例子是使用SQL DML语句来调用Java存储函数SGRADE：

```
SQL> SELECT SNAME,GRAD, SGRADE(GRAD) FROM STUD;
```

PL/SQL能够直接通过简单包含PL/SQL 调用说明来调用Java存储过程。

下面是PL/SQL直接调用Java存储过程的例子：

```
DECLARE  
    sal      number;  
    bonus    number;  
BEGIN  
    ...  
    A_BONUS (sal, bonus);  
    ...  
END;
```

提示 在Java，远端方法请求（RMI）调用简单地是特定的远端过程调用（RPC）。Oracle 8i使用RMI作为在因特网内部ORB协议（IIOP）之上的传输协议。正如你预期的，这个已存在的技术“式样翻新”是一个面向对象和网络编程的扩展。然而网络和分布式编程、面向对象编程和Java这些主题已经超出本书讨论范围，所以请参考适当的书籍以获得更进一步的解释和例子。

### 15.1.3 SQLJ : Oracle 8i Java预处理器

SQLJ是一个新的编程语法，它是用于JDBC的Oracle预编处理器；它可以被认为是PRO\*Java。它使开发者可以在Java程序中嵌入SQL语句，与Oracle的其他预编译器（例如PRO\*C）类似。与既支持动态有支持静态的SQL的Pro\*C不同，SQLJ只支持静态SQL。Oracle与许多其他软件商共同开发SQLJ，这些合作者包括IBM、Tandem（现在的Compaq）和Sun，在1998年12月SQLJ被采用为ANSI标准。它被设计为独立于目标数据库使用，因此如果其他厂家的数据库支持SQLJ，那么在其中使用Oracle SQLJ存储过程将是可能的。

用嵌入式SQL语句写一个Java程序，然后用SQLJ转化器处理它，用SQLJ的运行时调用替代SQL语句。然后任何Java编译器能够被用来编译产生的Java程序。SQLJ由一个转换器和一个运行组件组成。转换器检查嵌入式SQL的语法和标点符号，产生一个.java源文件以及一个或多个SQLJ环境资源文件。SQLJ环境资源文件包含关于你的SQL操作以及被访问数据的类型和模式的详细信息。然后，SQLJ自动地调用Java编译器，从.java文件中产生.class文件。SQLJ还自动调用定制器以适应连接的数据库的环境资源文件。在缺省的情况下，Oracle定制器被用于增加Oracle特定的扩展特性的支持。SQLJ运行库环境由一个SQLJ运行库，这个库是Java并且调用JDBC驱动器来连接正确数据库。

图15-3说明SQLJ开发过程。

开发SQLJ应用有四个基本步骤：

- 1) 使用SQLJ转换器转换SQLJ代码，用产生带有SQLJ的运行时调用的Java源文件。
- 2) 使用Java编译器编译Java源文件。
- 3) 定制Java.class文件。
- 4) 使用SQLJ运行库运行应用。

这里有一个在SQLJ程序中嵌入一个SQL select语句的例子。符号#sql放在SQL语句前，跟

随内容包含在大括号内部：

```
#sql { select deptno into :deptno from emp where empid = :emp };
```

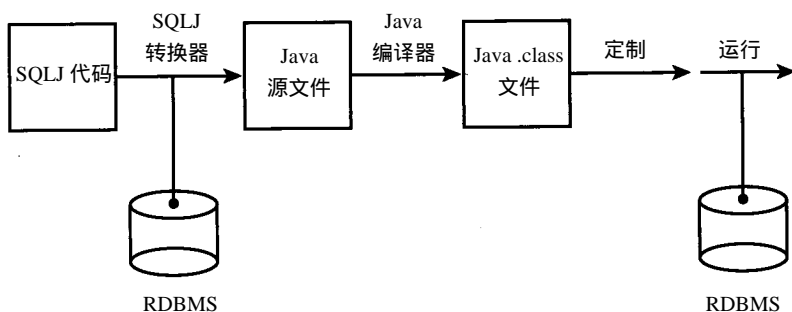


图15-3 SQLJ开发过程

运行SQLJ程序的唯一要求是SQLJ运行库、一个JDBC驱动器和程序将在其上执行的Java虚拟机。SQLJ比JDBC强，因为它在转换时检查SQL的语法和标点符号，而不是在运行时进行。你不必在SQLJ或JSBC之间做出选择，这些API是JDBC的预处理器；它可以被认为是PRO\*Java。它使开发者可以把嵌入SQL语句到Java内部。

SQLJ代码与JDBC代码对比示例

如下是SQLJ和JDBC对相同SELECT语句的程序片段。SQLJ代码比较短，因为它使用主变量来将参数传递到SQL语句，而JDBC使用分开的语句绑定每一个参数并获得每一个结果：

```
###SQLJ
java.sql.DATE a; int b; string c;
...
#sql { SELECT col1 INTO :a FROM EMP WHERE col2 = :b and col3 = :c };

###JDBC
java.sql.DATE a; int b; string c;
...
PreparedStatement x = connection.prepareStatement
("SELECT col1 FROM EMP WHERE col2 = ? and col3 = ?");
x.setInt(1, b);
x.setString(2, c);
ResultSet y = x.executeQuery();
y.next();
a = y.getDate(1);
y.close();
x.close();
```

#### 15.1.4 企业级JavaBeans支持

企业级JavaBeans基本上是实现一系列用于事务处理功能的预定义Java接口的分布的Java组件。它们使开发者能够设计并把应用封装在组件内，这个组件可以和其他开发者编写的部件装配在一起。基于组件的程序已经变得更加流行，因为它可再用应用代码，不同厂家的应用容易装配，布置灵活。更多的组织正在为新的应用使用基于组件的开发，这是由于传统的应用不能再用它们的代码。优点包括更好的设计，代码再用，更容易维护和在多层的环境中更容易跨层布置。JavaBeans已经正在被广泛用于装配到基于GUI的应用，但主要作为客户端应用。不同于客户端组件，服务器端的组件（如EJB）一般是事务性的，它们封装业务逻辑并

需要运行在服务器环境中。

Oracle为基于组件的分布计算选择EJB作为它们的结构。当前对于服务器端部件有三种模型：EJB、CORBA和DCOM。由JavaSoft创建的JavaBean特性将Java扩展到支持跨平台对象技术。分布式组件对象模型（DCOM）是用于建立在桌面上跨网络互操作的组件件的Microsoft组件模型。DCOM是COM的扩展。COM作为Windows操作系统的部分已许多年了；有了它作为基础结构，OLE和ActiveX才成为可能。通用对象请求代理结构（CORBA）3.0是最新特性，在1999年12月由对象管理组发布（OMG）。这个模型指定一个传输机制——因特网Inter-ORB协议（IIOP），它允许在不同硬件上运行的不同操作系统互操作。它也指定用于指定接口到对象的接口定义语言（IDL）。有关CORBA如何与Oracle相关更详细的讨论，见15.1.5节。

EJB在一个容器内执行，容器提供这些部件在其中运行的操作系统进程或线程。容器类型有Web服务器、事务处理（TP）监控器或数据库系统等。

#### 开发和配置企业级JavaBeans

开发和布置企业JavaBeans包括如下步骤：一个开发者需要为客户应用定义4个项目以能够请求包含业务逻辑的块（bean）方法：

定义远端接口——远端接口列出所有方法或客户能调用的公共接口块。

如下是一个用于Stock块的远端接口的例子：

```
public interface Stock extends javax.ejb.EJBObject
{
    float getPrice () throws java.rmi.RemoteException;
    void buy (float amount) throws java.rmi.RemoteException;
}
```

定义主接口——主接口作为使客户能够创建块实例的机制。以下是 Stock块的主接口：

```
public interface StockHome extends javax.ejb.EJBHOME
{
    Stock create () throws java.rmi.RemoteException, javax.ejb.CreateException;
}
```

定义块——块自身是标准的实现业务逻辑的Java代码。

下面是块自身的一个例子：

```
public class StockBean implements javax.ejb.SessionBean
{
    public float getPrice ()
    { ADD ANY BUSINESS LOGIC HERE ...
    }
    public void buy (float amount)
    { ADD ANY BUSINESS LOGIC HERE ...
    }
}
```

定义接口布置描述器——它将所有块组件连结在一起。其语法很像Java类，用Session Bean语句开始。这里是为Stock块创建一个布置描述器的例子：

```
SessionBean StockBean {
    BeanHomeName = "bn=myStock";
    Attribute
        RemoteInterfaceClassName = Stock;
    Attribute
        HomeInterfaceClassName = StockHome;
    ...}
```

这些步骤完成之后，你需要编译和封装你的Java代码。

用如下语句编译EJB、它的远端接口和对Java类的主接口：

```
% javac StockBean.java Stock.java StockHome.java ...
```

然后，用下列语句在EJB包（一个Java归档或JAR文件）中封装类文件：

```
% jar cf Stock.jar *.class
```

接下来，用Oracle提供的CREATEEJB工具增加块的布置描述器：

```
% createejb Stock.jar -desc Stock.dsc
```

最后，在Oracle 8i服务器上使用Oracle提供的DEPLOYEJB工具布置EJB。

如下是用用户SCOTT概要布置EJB和创建一个jar文件的例子：

```
% deployejb -u scott -p passwd -s <servicename> -gen Stockclient.jar  
Stock.jar
```

在成功地布置EJB之后，可以从一个客户端应用激活它。

### 15.1.5 Java和CORBA

CORBA（通用对象请求代理结构）是使用不同语言编写的分布式程序可在网络（包括因特网）上进行通信的面向对象的协议。CORBA由对象管理组（OMG）定为一个基于标准的分布式组件模型。它是建立和布置跨平台管理可互操作的分布式对象应用的标准。

CORBA 2.0（和更高版本）对象使用OMG的Internet Inter\_ORB（IIOP）协议通信。IIOP是在两个或多个运行在因特网、intranet和企业计算环境中分布式对象之间进行通讯的另一个OMG标准。

用不同语言编写和运行在不同平台上的CORBA组件能够透明地通讯和互操作。因此，客户和服务端代码能够用任何语言编写（不仅仅使用Java）并被编译为本地机器代码。Oracle 8i使用Java作为它的CORBA实现语言。Oracle 8i集成基于Java的，CORBA2.0对象请求代理（ORB），它使用户可以利用IIOP调入调出数据库。这个ORB基于来自Inprise VisiBroker for Java的代码。Oracle 8i带有一整套用于开发CORBA应用的工具，使用这些工具，你能编译IDL说明，或装载Java源文件或类到数据库中。

### 15.1.6 Oracle的Java开发工具：JDeveloper

JDeveloper 2.0是一个基于Windows的Java编程工具，它带有集成的项目导航器、一个代码编辑器、一个程序调试器和一个编译器。SQL转换器和调试器也是JDeveloper中的嵌入部分。

JDeveloper被设计供开发者建立多层Java应用。开发者创建、调试并布置Java存储过程、企业级JavaBeans、SQLJ和CORBA对象以及HTML Web应用。项目导航器使开发者能在一个会话的不同层中管理多个项目。有许多代码产生向导来帮助快速开发应用。企业级JavaBeans向导能自动创建新的EJB。在Servlet向导和Java服务器页（JSP）向导帮助下，Servlet和JSP能够很容易产生。数据格式向导指导创建单表或主从格式。开发者能使用PL/SQL到Java向导来从被选择的存储过程产生Java类定义。几个JDBC驱动器随JDeveloper被提供，用于访问其他数据库，尽管它不是最紧密地与Oracle集成。

## 15.2 因特网文件系统

现在用户能使用新的因特网文件系统将 Web页、文献、扩展页、字处理文件、图像和其他传统文件存储在数据库内。它表面上作为另一个文件系统并能够像任何其他网络目录一样被访问。IFS把它的内容组织到文件夹里，它们呈现给客户任何文件夹或目录应有的方法。存储在IFS中文件可得到Oracle数据库提供的所有可靠性和安全性，这是与其他文件夹或目录不同的。系统管理员的工作很容易，因为他们仅须维护、管理、备份和装载单个系统而不是几个分开的系统。

除了在数据库中提供比较容易的文件存储，IFS还提供关系和非关系数据的完整性。用户能用多种协议访问IFS，对它们的数据提供广泛的访问。被授权用户能查找和查看存储在IFS中他们的任何文件，他们可以从任何计算机通过标准Web浏览器、Microsoft Windows Explorer、FTP客户或email客户进行上述操作。IFS使用Java编写，作为特殊的解析器和翻译器。IFS使用XML来解析和“翻译”文件，XML是下一代因特网标识语言。它使数据库能使用文件过滤器将已知的文件类型的内容转换为XML文件。XML能够存储在数据库表中。当被客户请求时，翻译转换存储的XML到文件格式（HTML、XML或初始化文件格式）。用户通过Windows Explorer用拖曳方法能够很容易地移动文件进或出IFS。在IFS内双击它们能够直接编辑并启动适当的应用。当增加或更新一个文件时，IFS使用ConText技术来创建一个基于内容和文件的外部属性（作者、题目、日期等等）的关键字索引。随后，IFS能够使用预构造的索引快速完成查找。

IFS的关键特性包括如下：

高级查找——文件自动按内容和文件属性进行索引以提供关系查询和基于ConText的查询。

检入、检出（CICO）——当检出时文献被锁存，直到返回检入。

访问控制列表（ACL）安全——基于ACL模型的安全和特权。

版本化——使能多版本文献。

改变通知——email通知可以在文件的插入、更新或删除时获得。

自动过期截止——文件在经过一定期限后被清除。

XML支持——它通过文献解析和翻译获得。

广泛访问——它被通过SMB、HTTP、FTP、SMTP、IMAP和POPS协议提供。

Java、CORBA、PL/SQL编程API——用于编写基于IFS的应用。

## 15.3 Oracle interMedia

Oracle interMedia包含5个管理模块来处理数据库内的多媒体。

建立在Oracle Context 技术上的Oracle interMedia文本，使用户可以使用高级查询和自然语言查询搜索文本。Oracle interMedia Image管理二维静态、数字图像并支持基本的操作函数，如缩放和剪修。因为它支持多数桌面发布图像文件格式（包括TIFF、JPG、BMP、TRAGA、PCX、PICT、GIF、CALS Raster、SUN、Flashpix和RPIX），不是图像文件格式专家的用户也能够用interMedia Image存储和获得图像。

Oracle interMedia Video和interMedia Audio能够从不同源（例如，Oracle VideoServer、Oracle 8i和Web站），导入基于浏览器的视频和音频。如下工业标准的视频和音频格式被interMedia支持：AIF、AIFF-C、AUFF、WAV、Quicktime、AVI、MPEG和Real Networks。应用可以像对待传统关系型数据一样查询并获得多媒体数据。Oracle interMedia Locator处理定位查询，例如查找存储、分布点和基于给定空间数据的事件（例如某一地址的位置或距离）。因为interMedia完全集成在Oracle 8i中，开发者通过SQL调用而不是专用接口，具有高级文本查找能力。

Oracle interMedia Text 使用文本索引。文本索引能够在几乎所有数据库的任意列或者对象上创建来完成查找。文本查找利用在数据库内的索引执行并返回一个匹配文档列表。尽管被interMedia查找的文献能够存储在文件系统或任何 URL地址的WEB上，但数据库存储和管理所有文本索引。interMedia能够用单个SQL命令处理混合查询（数据库中的全文或结构化数据）。例如，你能够查找包含“Internet”的文档并结合存储数据中的其他信息（例如，作者、日期或文件类型）。SQL CONTAINS函数使用户能从任何支持SQL或PL/SQL的工具或环境中运行文本查找查询。interMedia提供全文搜索能力，它包括精确单词或短语查找、布尔特性、模糊查询、通配符、大小写敏感和部分查找。它通过使用大约五十万个单词的主题内容识别文档的能力。还有根据主题查找文档，这可通过使用优化组合全文、基于语言和基于主题技术来进行。

下面的例子将对 bdate列进行查找。然后结果与 CONTRAINS函数合并这个函数搜索包含“Internet”字并和浏览器有关的文档的文本索引：

```
SELECT author, isbn, bdate from books
WHERE bdate > '01-MAR-1999' AND
CONTAINS ( btext, 'Internet AND ABOUT (Browser)' ) > 0;
```

Oracle interMedia使用与Java或C++类似的对象数据类型，称之为ORDAudio、ORDImage和ORDVideo来描述声音、图像和视频数据。这些数据类型的一个实例由诸如元数据、多媒体数据和方法等属性组成。元数据是关于数据的信息；多媒体数据是实际的声音、图像和视频；方法是能够在对象上执行的过程（例如，store、deliver和compress）。多媒体数据能够在数据库中作为BLOB类型保存，或者在外部保存并能够很容易地导入到数据库中。多媒体数据能够存储在其他源中，如BFILE（大对象的操作系统文件）、基于HTTP服务器的URL、特定的媒体服务器（例如Oracle VideoServer）或用户定义的源。这些数据类型的元数据和方法总是在数据库中进行存储和管理。interMedia使媒体数据能通过标准查询或关系数据（例如学生姓名）和对象的元数据（例如学生照片）来定位。当定位后，媒体数据能够通过ORDAudio、ORDImage和ORDVideo数据类型，像其他关系数据一样使用SQL对它进行访问。

Oracle interMedia Locator支持用于定位应用和近似查询的联机地理位置编码功能。使用Oracle 8i空间组件创建它。它在Oracle 8i数据库中支持地理编码、存储和地理编码数据查询。地理编码使用点代表地址和位置信息（例如，邮政编码、经度 / 纬度）。和第三方地理编码和绘图工具相结合，interMedia能够完成地理编码数据的距离查询。在数据被地理编码后，interMedia在应用中能计算距离和图形化表述它的位置。例如，你能查询数据库来定位距你工作地点最近的加油站。

## 15.4 Oracle WebDB

Oracle WebDB提供包括HTML服务器的一个完整的Web开发环境，因此它能处理HTTP请求和服务Web页。不需要在每个用户的机器上安装软件，也不需要使用FTP进行远端文件管理，或者进行复杂的软件升级。开发和布置应用的客户软件时，唯一需要的是Web浏览器——Netscape Navigator 3.1或更新的，或者Microsoft Internet Explorer 4.0或更新的。WebDB被设计成方便使用、方便访问和方便管理。它是完全包含在Oracle数据库（版本7.3.3或更新）的PL/SQL过程的集合。它也包括作为Web服务器和PL/SQL到数据库的接口的轻量监听器。WebDB将全套基于浏览器的用于开发Web的HTML工具与HTML接口相结合。通过使用指导用户经过创建组件（也就是表格、报告、图表、菜单、日历、分层）的整个过程的简单向导，可以实现快速开发。向导可以替你写PL/SQL代码，你也可以绕过向导自己编写代码。当开发者创建一个组件时，它被给予一个URL。用WebDB进行布置与增加URL到HTML文本一样容易。图15-4给你一个图表向导的例子。

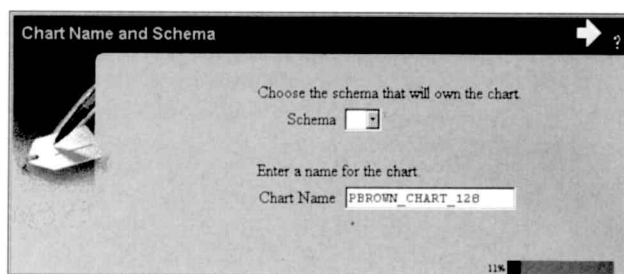


图15-4 WebDB图表向导屏幕

在WebDB中包含所有用于管理和构造数据库的工具。管理员能够从浏览器管理授权和监听器设置或监控终端用户和数据库活动。通过Web浏览器可以创建、查看、编辑和拖动任何数据库对象（表、分区、重做日志等等）。WebDB有高级性能追踪能力，这有助于解决性能问题。你能够监控的用户因素包括：响应时间、按组件名称的请求、按天和小时的请求、按浏览器类型的请求和按IP地址的请求。图15-5 说明按用户的请求数图表。

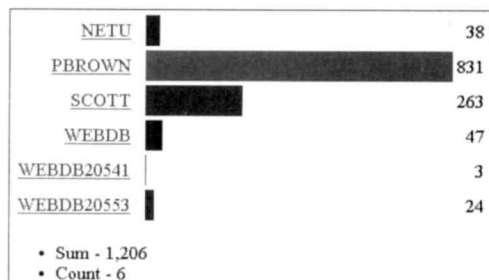


图15-5 一个WebDB图表显示按数据库用户的请求

使用WebDB创建的Web站点被分成叫做角（corner）的几个部分。角包含与特定主题相关的条目。站点管理员为角指定负责管理该区域的拥有者。角的拥有者能将对他们的角的内容进行不同类型访问（例如，增加、编辑和删除）的特权授予其他用户。内容的作者能够更新Web站点的角，只需点击界面而无需任何其他软件。当维护集中控制的Web站点时，它分散内容的管理。

## 15.5 Oracle 8i的可用性和可恢复性

Oracle 8i的高级特性和能力提供随时可获得的增强，包括并行服务器的改进、一个新的快

速启动的失败恢复、联机备份、恢复和透明的 Application Failover。

Oracle并行服务器使多个Oracle实例能运行在一个簇的节点上以并行访问单一数据库。这种共享访问帮助删除任何单个失败点，因此一个系统中一个节点毁坏，其他节点能够继续工作。在Oracle 8i中，并行服务器通过使用缓存融合聚簇结构体系和簇负载平衡来支持成千个用户。缓存融合使用专用高速互连直接在节点间为虚拟无限制规模传递数据。这个技术使在簇中一个节点缓冲区可直接向其他节点的缓存发送数据块，因此可减少磁盘 I/O。簇负载平衡自动将用户连接到利用率最低的系统节点，因此减少了用户访问时间并增加了可被服务的用户数。

自动化和只读独立数据库使Oracle可自动维护一个或多个用于灾难性恢复的成品数据库拷贝。独立数据库最初通过复制成品数据库创建。复制能够驻留在同一位置或不同位置。在独立数据库创建之后，成品数据库的更新和改变被传播到独立数据库，因此数据库保持同步。独立数据库使用加强恢复模式。当在成品数据库产生归档重做日志以后，它们通过 Net 8连接传输到独立数据库。在独立数据库接收归档重做日志后，它们被自动运用以使两个数据库保持同步。这样就不必手动维护独立数据库，但也可以这么做。如果网络发生错误或者发生空间溢出问题，阻止日志文件成功创建，那么，在特定的时间间隔之后，自动重试归档操作。以前，如果没有随后创建的成品数据库产品的另一个拷贝，当前的独立数据库不能被打开或使用。在Oracle 8i中，DBA现在可以只读模式打开独立数据库。这提供了正当对成品数据库卸载一些进程时查询和产生独立数据库报告的一种办法。在这一模式下，独立数据库能够连续自动接收来自成品数据库的归档重做日志；然后，当独立数据库被返回到加强或手动恢复模式，排队的日志被应用。

联机数据库重组和碎片消除是另一特点。Oracle 8i现在在数据库重组与索引创建和重建期间，用户可以完全访问数据库。当表和索引完全联机且没有中断任何正在进行的对基础表的改变时，DBA能够执行维护。

当正在创建索引时，用户能够连续访问和更新基础数据表。在创建期间任何对基础表索引的改变被记录在日志表中，并在操作结束后应用到新索引中。索引也能够被联机重建，提供高的数据库可用性。当索引创建时，重建使用日志表来记录在重建操作期间对基础表所做的改变。除了碎片消除不需要附加的存储空间外，联机索引碎片消除和合并很像联机索引重建。索引碎片消除进程将单块合并到相邻的小于一个数据块的索引块中。

Oracle 8i当用户可用索引组织表（IOT）时，IOT可以被移动和重组。对表所做的改变记录在日志表中，在进程结束时合并表。这些联机重组能够在单个 IOT或在被分区的IOT的单个分区中进行。

快速启动的失败恢复是一个旨在使系统失败恢复尽可能快并减少数据库不可用时间的新特性。它是很迅速的，恢复时间能够由管理员设定。在一个系统崩溃后，数据库一般做向前回滚提交过的改变和回滚未提交的改变。Oracle 8i有一个新的快速启动检查点结构。检查点进程不断地写被修改的块到磁盘中。DBA指定多长时间向前进行回滚取代应发生多少次检查点。回滚处理希望通过使用称之为非块回滚的技术进行消除。现在，新的事务能够在向前回滚完成之后马上开始。如果一个新事务访问被死亡事务锁定的一行，只有阻塞它的进程的改变被回滚。同时，数据库有能力并行回滚死亡事务。

扩展备份和恢复用恢复管理器提高性能。恢复管理器是一个管理创建和还原数据库备份

的进程的工具，它现在有一个能够代表数据库引导备份软件完成复制操作的集成的媒体管理 API。它也能在自己目录上针对媒体管理软件目录和过期标注删除备份进行交叉检查。恢复管理器使 DBA 能在恢复/还原操作发生之前预览即将发生的事。另外，恢复管理器现在能多路备份集到多个 I/O 设备。Oracle 使用 Transparent Application Failover 能增强从系统崩溃中恢复的能力。当数据库连接中断时，客户程序能够自动重新连接。同时，任何在崩溃时正在被处理的查询能够自动重启，返回程序的数据将从失败发生点恢复。

## 15.6 安全性

Oracle 8i 提供一个布置和执行分布式 Java 应用的安全平台。过程、CORBA 服务器和企业级 JavaBeans 组件都能够在 Oracle Net 8 和 CORBA IIOP 上通过因特网标准的安全套接字层 (SSL) 进行访问。Oracle 8i 提供单个符号加密，使用户可在因特网/intranet 环境中认证他们自己，这样可以缓解有太多口令的问题。基于单符号加密的公用密钥结构 (PKI) 使用 X.509 (版本 3) 证书在 SSL 上认证。Oracle Wallet 管理器是一个由 Oracle 提供的用来保护和管理用户证书、密钥和信任点的工具。

Oracle 8 引入了企业角色，这是一个或多个跨越一个或多个数据服务器、被集中管理和维护的全局角色。Oracle 8i 通过在 Internet 目录源中存储和管理用户信息，而不是用专用数据存储来提高这些能力。Internet 目录符合轻量目录访问协议 (LDAP) 版本 3 标准。单站管理是通过 Oracle 企业安全管理工具的增强得到的。现在，管理员能够从单个控制台在多个 Oracle 8i 数据库中创建一个用户，或者跨多个数据库创建企业角色。Oracle 8i 引入单个企业用户，在整个组织为每个用户创建一个帐户，而不是为每个用户创建多个帐户。现在，你要管理的用户帐户更少，安全更容易通过使用 Internet 目录来加强。

Oracle 8i 提供强迫的、灵活的、好的访问控制以及安全的应用环境。通过一个或多个在表或视图上的安全策略使它们可用。用访问控制，在访问数据时有更好的粒度。对不同类型的访问 (也就是 select、update 和 delete) 有不同策略。现在，你能指定谁能访问表中的特定行，区分数据访问到单一级别。访问能够基于用户 (例如，一个用户角色是经理还是职员) 或基于应用 (例如是用户正在访问一个人力资源应用，还是顾客正在创建一个定单)。

下列这个例子中，表 ACCOUNT 有一个策略，银行的顾客仅能看他们自己的帐户，一个顾客通过发出如下命令访问表：

```
SELECT * FROM ACCOUNT;
```

这个 SELECT 语句使数据库调用策略函数，然后透明地和动态地通过附加的 WHERE 语句重写这个查询：

```
SELECT * FROM ACCOUNT WHERE ID = (SELECT ID FROM CUSTTAB WHERE CNAME =  
USERENV('user'));
```

安全应用环境是在 Oracle 8i 中基于访问控制策略提供的。应用环境是由用户作为它们的属性定义；每一个应用能够有不同属性的环境。例如，人力资源应用能有基于 employ number、department、position 属性的安全策略。据此，数据库将通过使在经理位置上的用户只能更新他自己部门里的职员的记录，以执行这个策略。

## 15.7 其他特性

Oracle 企业管理器 2.0 版是 Oracle 8i 提供的最新版本的系统管理工具。提供的一些特性有：

从任何虚拟环境管理的能力、在企业中管理每件事的能力，和最少化手工管理系统的“设置和忘记”管理。它是基于 Java 的应用并能够从浏览器中访问。管理员能够从一个中心控制台管理多个机器。仓库现在是一个共享的公共知识库，它可供多个 OEM 管理员使用。最初有一个缺省的超级用户 SYSMAN，SYSMAN 能创建其他超级用户和管理员并分配给它们适当的职责和特权。每个新的管理员被给予一个完成管理任务需要的访问控制台口令。因为每个管理员有特定的职责，所以他们能为其他管理员定义对他们创建的组、工作和事件的允许级别。允许级别包括无、视图、修改和完全。

LogMiner 是一个让 DBA 使用 SQL 来读、分析和解析重做日志文件的数据库分析工具。LogMiner 要求访问被分析的数据库的数据字典并用来自 Oracle 8.0 的重做日志文件继续工作。正被分析的日志文件被映射到动态性能视图 (V\$logmnr\_contents)。在这个表中的每一行代表在数据库中完成的一次逻辑操作的信息。对这些操作的每个，由给定的 REDO 和 UNDO SQL 语句描述在数据库中所做的改变。SQL REDO 语句显示在数据库中所做的原始操作，而 SQL UNDO 语句显示出能回滚这些改变的语句。

这个例子给出的查询要查找 Benlin 对表 SAVINGS 所做的改变。

```
select sql_redo, sql_undo from v$logmnr_contents where username = 'Benlin' and  
seg_name = 'SAVINGS';
```

查询结果如下，它显示出 Benlin 首先删除他的储蓄记录，然后用更高的量重新创建这个记录。Benlin 所做的回滚改变的语句显示在 SQL\_UNDO 列。

#### SQL\_REDO

```
delete * from savings where id=999  
and rowid='QSDFJKKJKAA';  
insert into savings(name,id,amt)  
values ('benlin', 999, 500);
```

#### SQL\_UNDO

```
insert into savings (name, id, amt)  
values ('binlin', 999, 10);  
delete * from savings where id=999  
and rowid='QSDFJKKJKAA';
```

数据库资源管理器是在 Oracle 8i 中在数据库用户和应用之中分配和管理 CPU 资源的一个新工具。过去，所有用户和应用对数据库进行同等访问。现在你能对更重要的任务分配更多资源。使用这个工具，用户和组被分配到资源消耗组。管理员建立一个资源计划指定如何在不同资源消耗组中进行资源分配。分计划进一步细分资源，它能在资源消耗组中进行定义。对于分配资源最多能定义八个级别。使用数据库资源管理所做的改变能够动态地在数据库上改变，不需要关闭后再重启系统。管理员能够有许多不同的资源规划和资源消耗组，所有这些都被保存在数据字典中。

并行自动化程度 (ADOP) 特性也被集成在数据库资源管理器中来优化系统使用。这个特性限制并行查询操作的并行化程度。

分区使用户能将表和索引分成更多的可管理单元。Oracle 8i 引入了对表的范围分区，在范围分区里数据是基于分区键的值的范围被分开。在 Oracle 8i 中，现在有哈希分区和合成分区。哈希分区使你能均匀地跨设备分离数据，因为行按一个分区键的哈希值分开。合成分区是范围和哈希值的合并。首先，数据被按照值进行范围分区。然后，每一个分区再细分为几个哈希分区。Oracle 8i 也支持面向关系和 LOB 列的类型。分区能够独立地从表或索引的其他分区中被备份和恢复，不妨碍未受影响分区的访问。

具体化视图是一个新的数据库对象类型。它是用于创建、维护和使用总和表的优化特性。

总和表也称为集合表，在其中求总和和连结操作被预计算，结果被存入数据库中。通过访问总和表而不是每次根据数据求总和，性能得到很大提高。管理员能够指定如何和何时刷新这些表（也就是同步、当新数据被装入时、每周等）。同时，总和咨询向导帮助管理员创建和维护总和表。

可传输的表空间使你能在数据库之间移动表空间，而不用做完全的导入 / 导出。DBA能够在相同的系统之间（硬件结构）复制表空间的数据文件，因此转移数据既容易又迅速。要传输的表空间必须被自我包含，也就是没有从表空间内部到表空间之外的引用点。这里有一个称为transport\_set\_check的PL/SQL过程，用于检查一系列表空间是否是自我包含的。

首先，被移动的表空间被置为只读模式来保证被捕获的数据相一致。然后，导出表空间的元数据，这是迅速的，因为数据量很小。接下来对表空间的数据文件进行复制，使用操作系统应用将其复制到新的系统中。在新的系统中，导入表空间的元数据。这也同样迅速，也因为量很小。还有另一种方法来在表空间上执行按时间点恢复。通过移动一个好的表空间的拷贝到成品数据库中，恢复能够被迅速归档。

本地化管理（位映射）表空间是在数据库中提高空间管理和最小化区间碎片的一种新的表空间类型。一个区间是由一个或多个相连的数据块组成的逻辑存储单元。在 Oracle 8i之前，管理自由区间和使用过的区间是通过数据字典表来完成的。现在，有位映射选项能指明表空间中使用过的块和空闲的块，因此最小化了对数据字典的访问。这些表空间使用带有EXTENT MANAGEMENT LOCAL子句的CREATE TABLESPACE命令创建。

这种类型的表空间将自动调节所有新区间的标准大小，既可用 autoallocate选项又可用uniform选项。在autoallocate选项下，系统将自动创建变化区间尺寸。你能指定初始区间的尺寸，然后系统将决定附加区间尺寸。在 uniform选项下，所有区间创建是相同尺寸的。如果你不指定EXTENT MANAGEMENT子句，缺省是EXTENT MANAGEMENT DICTIONARY，它表明这个表空间将由系统通过数据字典进行管理。

注意 如果你使用LOCAL子句，就不能在CREATE TABLESPACE命令中使用缺省的存储语句、MINIMUM EXTENT或TEMPORARY参数。

下面是创建本地化管理表空间 TS的例子。它的每个区间分配为 64K，在位映射中的每位表明32块（假设每块2K）。

```
SQL> create tablespace TS
      2 datafile '/u01/oracle/mysid/ts.dbf' size 10M
      3 extent management local uniform size 64K;
```

现在能够在用户临时表空间中创建临时表。临时表仅在事务或会话期间保持，只对向它们插入数据的会话可用。临时表不能被分区、索引组织、聚簇或者包含嵌套表或变量数组类型的列。这些表不能有外键约束定义。

下列语句创建临时表 stocks：

```
SQL> CREATE GLOBAL TEMPORARY TABLE stocks
      (stockname varchar2
      stockprice float);
```

Drop column命令提供从一个表中删除一列而不必重新创建该表和它的相关索引和约束的能力。

这个SQL语句将从表census中删除列描述符 col\_1和与col\_1相关的每一行数据，在该列上

定义的所有索引或约束也都被删除：

```
SQL>ALTER TABLE census DROPcol_1;
```

## 15.8 小结

有了新特性和增强，Oracle 8i已成为一个因特网计算的理想平台。Oracle 8i将通过服务器端的SQLJ与数据库中的Java虚拟机均衡，从而使上万个Internet用户连接到Oracle 8i数据库的一个单独的实例。因为JVM是集成在Oracle 8i中的，所以它继承了数据库的可靠性、可移植性、可伸缩性、吞吐能力和性能。开发者能用Java过程取代PL/SQL过程，或者使它们共存。提供的工具（例如WebDB）改进使用的简单性和Web应用的布置。iFS选项使多协议跨平台访问相同的信息对所有用户变得更加容易。interMedia产品使标准SQL能访问应用中的文本和多媒体内容。