

基于 Oracle*Forms 的遥感解译知识库查询系统的实现

部委攻关项目：CBERS-2 卫星应用系统图像处理分系统

中国科学院地理科学与资源研究所 黄方红 姚一鸣

本文以一个卫星应用系统图像处理子系统的开发为例阐述了如何利用 Oracle*Forms4.5 做为客户端开发工具, 对建立在大型关系数据库 Oracle 基础上的解译知识库查询系统进行的界面与功能开发及系统开发的有关关键技术。

遥感解译知识库查询系统作为遥感图像应用系统的专家知识支持子系统, 通过查询系统目的是获取解译目标的各种知识, 辅助解译人员在遥感图像上准确地识别特定的目标, 增加解译人员对目标的判读经验和分析处理能力。本文以一个卫星应用系统图像处理子系统的开发为例, 介绍如何利用 Oracle*Forms4.5 做为客户端开发工具, 实现遥感解译知识库查询实体系统的界面与功能开发过程的技术实现。

系统运行环境和开发工具

系统建立在局域网的 Client/Server 结构体系基础上, 以 ORACLE 作为数据库的后台支持, 应用 DEVELOPER2000 的 Oracle*Form4.5 作为前端开发工具进行系统的应用开发。Oracle*Form4.5 作为 Oracle 数据库的应用开发和运行的支持工具, 主要用于设计和生成一个基于表格的应用系统。Oracle*Form4.5 采用基表来管理应用, 把应用分解为对象和属性, 在对象创建和属性定义手段方面, Oracle*Form4.5 提供丰富的对象和属性, 同时提供灵活的应用定义手段, 不仅能够处理文本信息, 还支持带有图像信息的数据库应用。它的这些特点为该系统的开发提供了很好的支持。

系统结构与功能

该系统由目标样片库、目标仿真库、目标特征库、目标光谱库、目标分类库和专用分析模型库六个实体数据库组成, 数据包含目标类的各种描述性文本、文件、专家知识和图像等类型信息。遥感解译知识库查询系统实体结构如图 1。解译知识库查询系统实现功能如下:

- (1) 连接数据库。进入解译知识库查询系统, 必须首先连接数据库。这意味着用户必须拥有一个被授予于一定权限的有效用户名和口令。
- (2) 表数据的浏览、显示。
- (3) 图形、图像的浏览、显示、放大。

- (4) 信息查询与显示。
查询方式:
① 单个或多个属性约束条件组合查询;
② Oracle 的匹配模式查询。
- (5) 数据的维护管理。包括数据的增加、修改、删除、提交。
- (6) 信息打包。所谓信息打包是将用户需求的多层(类)数据或系统为特定任务而需要的多层(类)数据组合成一束数据流整体返回。实际操作是通过空间和非空间数据库查询和浏览, 将查询的图形、图像、文件及数据集等多种类型信息集合, 有选择性地定制成文件包放置到指定的地址上, 以便各级解译人员和用户快速得到打包结果。

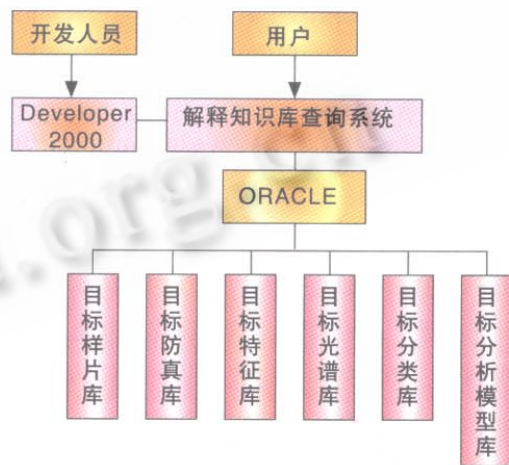


图 1 系统实体结构

Oracle*Forms 的应用开发模式

利用 Oracle*Forms 进行解译知识库查询系统的应用开发范式可大致归纳如下几点:

- (1) 首先必须分析用户的应用需求, 只有详细了解用户要求, 优化用户要求, 得到用户认可, 才能开发出满意的最终应用, 这一步需要应用开发人员和用户一起完成;
- (2) 要求开发人员把用户的具体要求抽象成一个一个 Form 对象及其属性;

(3) 在识别出应用对象的基础上, 利用 Form 定义界面, 对对象自上而下逐层逐个定义, 定义过程一般先利用缺省功能定义几个 Form 的主要对象, 生成应用的框架, 从而派生出许多子对象、派生对象和缺省对象, 之后再根据需求修改这些对象, 直到满足要求为止。

(4) 构造事物处理。在实现数据库查询中, 对于事物处理的逻辑操作在应用对象属性中难以体现的, 将设计进一步采用触发器 Trigger (程序) 来实现这些操作。

解译知识库查询实现若干技术

(1) 视图的使用。视图 (VIEW) 是一种数据库对象, 它允许用户从一个表或一组表中建立一个“虚表”。视图和表的不同之处在于视图中没有数据, 而仅仅有一条 SQL 查询语句。在 Oracle 中创建和使用视图有许多优点, 如: 视图可以提供附加的安全层, 可以隐蔽数据的复杂性, 有助于命名的简洁, 带来更改的灵活性。

在解译知识库查询系统中, 创建和使用视图主要的目的是为了隐蔽数据的复杂性, 避开了相互关联的多表之间的关系, 减少 Oracle*Form4.5 中为建立表与表之间关联的而不得不做的大量编程, 简化了数据库中多表的表达方式。这一方式适用于数据库中表与表是多对一关系的数据库, 对于一对多关系的数据库, 不可使用视图, 否则数据的表达将显得含糊不清。

(2) 主-从关系的数据表达。从解译知识库的 ERD (实体关系图 略) 可知各个实体间存在有一对一、一对多及多对一三种关系。解译知识库中的六个主体库均由多个 Oracle 基表组成, 例如目标样片库是由主表样片表和三个从表目标类型表、传感器表、样片图片表组成。通过 ER 图可得到每一个主体库中主表与从表之间的关系, 在 Oracle*Form4.5 的数据库查询界面设计时, 对于主表与从表之间为一对多关系的数据库, 它的数据库查询界面必须建立表间主-从关系, 在 Oracle*Form4.5 中由 Form 中的 Relation 对象来协调两个相关块 (block) 之间的主从联系。解译知识库中目标特征库和目标光谱库就是两个典型的必须创建主-从关系的库。以目标特征库为例, 作为目标特征库主表的目标特征表与数字特征表之间存在一对多关系, 如果不在查询界面上以不同的块 (block) 分别将目标特征表与数字特征表以一次显示一个记录和多个记录的方式表达数据的一对多关系, 用户将很难理解查询得到的结果。

(3) 使用触发器 (Trigger)。触发器是由 Form 应用运行

时某一事件触发的一段 PL/SQL (过程化的 SQL 语言) 程序代码。在 Oracle*Form 中, 每个事件都有预定义的事件处理, 对于可带有触发器的事件, 开发人员需通过定义触发器来增强或替代原缺省处理过程, 这样可以在多个层次上实现一些复杂的规则。在该系统的界面开发中使用的主要触发器有: 用户自定义多条件组合查询方式的触发器 PRE-QUERY、POST-QUERY, 这是该查询系统中重点使用的触发器, 它对于进行时间数据项的区间段查询特别有用, 例如针对有时间数据项的区间段查询时, 在 PRE-QUERY 触发器中可设定缺省的 where 语句如下:

```
where:='Time between nvl(to_date(:block.time1,
"yyyy/mm/dd"),
to_date("1900/01/01", "yyyy/mm/dd"))
and nvl(to_date(:block.time2, "yyyy/mm/dd"),
to_date("2100/01/01", "yyyy/mm/dd"))';
```

此外, 还有显示系统信息处理及警示的触发器 ON-ERROR、ON-MESSAGE; 设置按钮触发器 WHEN-BUTTON-PRESSED; 设置鼠标的触发器 WHEN-MOUSE-DOUBLECLICK; 设置增强项级上合法性检查的触发器 WHEN-VALIDATE-ITEM; 设定用户界面的触发器 WHEN-NEW-FORM-INSTANCE; 以及主从协调触发器 ON-CLEAR-DETAILS、ON-CHECK-DELETE-MASTER、ON-POPULATE-DELTAILES 等。

(4) 树结构的层状查询。树结构数据存放在表中, 数据之间的层次关系 (即父子关系) 必须通过某种方法加以描述。在 Oracle 系统中, 树结构是通过表中的“列”与“列”的关系来描述的。解译知识库中的目标分类库 (class 表) 就是一个树型结构数据, 在 class 表中, 每个元组都有类型码和父类型码, 父类型码对于它的下级类型来说是父类的类型编码, 而对于该父类本身来讲也是一个类型编码。这个表的父子节点就是通过类型码和父类型码联系起来的, 即父节点的类型码 classtypecode=子节点的父类型码 classparentcode。所以, 对于树型结构的目标分类库的查询, 可选择任何分类树的节点为起始节点, 从此节点开始向下查询目标分类系统中它的下一级目标类别, 并可向下逐级查询直至最后一级。

主要由两个 Trigger (触发器) 来实现目标分类库的查询, 一个是 item 级的 Trigger:

```
WHEN-MOUSE-DOUBLECLICK, 该 Trigger 下的 pl/sql 是:
```

```
BEGIN
```



```

: GLOBAL.CURRENT_V := :
SYSTEM.CURRENT_VALUE;
: QUERY.YES:='Y';
EXECUTE_QUERY;
END;

```

另一个是 block 级的 Trigger: PRE-QUERY, 该 Trigger 的 pl/sql 如下:

```

DECLARE
WHERE1 VARCHAR2(1000);
BEGIN
WHERE1:='classparentcode=(select classtypecode
from ikd.class where
  Classtypename=:GLOBAL.CURRENT_V)';
  SET_BLOCK_PROPERTY
('class',DEFAULT_WHERE,WHERE1);
END;

```

(5)索引的使用。数据库被建立之后,如何根据用户的需求快速得到数据库中的信息,这是数据库查询中重要一个环节。在 Oracle 中实现数据库查询,实际上是使用 SELECT 命令查询记录,如果查询的内容是在最后一个记录,则需要扫描整个表。这样效率太低,一旦文件很大,检索时间就很长。为了提高查询速度,Oracle 提供了索引操作,为表建立索引。

Oracle 提供了两种类型的表索引:唯一性索引和非唯一性索引。唯一性索引实施了关键字和唯一性约束;非唯一性索引改善了查询性能。当你在创建数据库的表时声明主关键字和唯一性约束,Oracle 便自动地建立相应的唯一性索引,故可不建唯一性索引。对于表中除了主关键字(PRIMARY KEY)和唯一性约束(UNIQUE)字段(列)之外的查询项,可建立非唯一性索引,为选择这些字段做查询加快查找速度。例如要为目标样片库(sample)中的时间列(samplecapturedate)建索引:

```

Create index Sample_date on sample
(samplecapturedate);

```

(6)数据打包。解译知识库查询系统,不仅是为了提供本地的当前用户浏览查询 IKD 的数据信息,同时也希望能将查询到的数据集、图像按照客户的要求放置到指定的地址上,即查询结果的数据打包,以便所有其他的用户可以直接从指定的地址得到查询结果。解译知识库(IKD)含有描述性的文本数据、图像数据以及文件存放路径三类信息,所以对查询结果打包时,必须同时考虑对

这三类数据的打包。

①属性数据打包。这里的属性数据是指在 Oracle 数据库的表中数据类型为字符型(varchar2,char)、数值型(number)、日期型(date)的数据项,对于查询结果中的这类数据项,Oracle*Form4.5 开发工具提供了系统记录指针,通过指针可逐一将查询到的多个记录中的属性数据输出成文本格式文件。

②图像数据打包。图像数据指在 Oracle 数据库的表中数据类型为LONG RAW型的数据项,在Oracle*Form4.5 中将 LONG RAW 型的图像数据打包可用:

```

WRITE_IMAGE_FILE('文件名','文件类型','图
象项');

```

的方式,将当前记录中的图象输出成文件,但对于查询结果有多个记录的数据,它无法使用属性数据打包一样的方式,通过指针输出所有的记录。这样,对图像数据的打包,必须在查询界面上逐个显示查询记录的同时,进行图像的直接保存,然后和属性数据一起打包到指定的地址上。

③文件打包。IKD 数据库表中含文件存放路径的数据项,虽然它的数据类型是字符型(varchar2),但由于该数据项指定了文件的存放位置,在数据打包中,用户不仅想知道数据的存放位置,而且希望直接从打包的结果中得到这些文件,Oracle*Form4.5 可以在属性数据打包的同时,通过指针逐一读取记录中存放路径数据项的路径位置,将该位置上的文件重新放置到用户指定的地址上。

结论

本文的解译知识库查询系统是图像处理系统中的综合数据库系统的子系统,它即可以作为一个独立运行的系统,也可以嵌入到综合数据库系统中,形成一体化界面的综合数据库集成系统。该系统目前已利用试点区数据通过了各类功能的测试,有待进一步完善的工作是:如何将关系数据库中多个基表数据,由查询结果更加符合现实和用户需要地表达和体现在系统界面上、查询的优化处理等等。■

参考文献

- 1 Edward Honour, ORACLE 开发人员指南,机械工业出版社,1998
- 2 Scott Urman, Oracle 8 PL/SQL Programming,机械工业出版社,1998
- 3 David M.Kroenke, Database Processing-Fundamentals,Design,and Implementation, Prentice Hall
- 4 George Koch,Kevin Loney, Oracle 数据库使用大全,电子工业出版社,1996