

Informix 关系数据库与 C 语言调用

曹福元 沈 鸣 (南京大学)

摘要:本文对 Informix 关系数据库有关 C 语言的调用作简单的介绍,针对两种不同版本的 Informix,给出了 C 语言调用和程序示例,并对其优缺点作了具体分析。

Informix 关系数据库管理系统是美国关系数据库系统公司为 UN IX 操作系统开发的第一个 OLTP(联机事务处理)系统,现已有微机、小型机、单用户环境、多用户环境等多种版本,提供支持 TCP/IP、NFS 和多种高级程序设计语言接口的专用软件包。Informix 关系数据库管理系统是用 C 语言开发的,并把 C 语言做为其宿主语言。

目前国内使用较为广泛的 Informix 有两种类型,支持 SQL 语言的版本和不支持 SQL 语言的版本,如 Informix 3.30。它们均提供一组完整的数据定义、数据输入、数据查询和报表生成的软件包,见表 1。

表 1

	Dbbuild 数据库描述语言,用户建立或修改数据库结构。
	Perform 屏幕事务处理程序。
	Ace 关系报表书写语言。
Informix 3.x.x版	Enter1 数据录入。
	Enter2 面向屏幕的数据录入。
	Ioformer 用于数据检索的查询语言。
	Dbstatus 数据库监控程序。
	Informix-SQL: 交互式SQL运行环境。
Informix(支持SQL)	Inormix-ESQL / C: C语言程序应用接口。
	Informix-4GL: 过程化的SQL语言。

支持 SQL 语言的 Informix 也有一组应用软件包,如 Perform、Ace、dbload、sqlconv、bcheck、dbupdate 等。

Informix 提供了两种运行环境,既可在数据库管理系统下进行交互操作,也提供了与其它高级程序设计语言的接口,也就是说,可通过在高级程序设计语言中嵌入例行程序(Informix 3.30)或使用 SQL 语句来实现数据库的管理。

本文拟就 Informix 3.30 和 INFORM IC - ESQL

/ C 环境下,在 C 语言程序中,如何操纵数据库作一简单介绍。

C 语言具有描述问题的能力,灵活性好,目标代码质量高,尤其是它丰富的函数调用,为设计高质量的人机界面友好的程序提供了理想的工具,因而被广泛的使用。

Informix 3.30 提供了个 C 语言应用例行程序库 ALL-II;在 C 程序中,用户以函数调用方式调用例行程序库中的相关函数,实现数据库管理。

ALL-II 例行程序库提供了 42 个函数,常用的函数大约有 10 多个,根据函数的功能大致可分为三类:

(1)数据库及其文件的打开、关闭、文件视图定义。如 dbselect(), dbopen(), dbclose(), dbstructview()等函数。

(2)操纵数据库文件,对其增加、删除、修改、查找、加锁、解锁。如 dbadd(), dbupdate(), dbdelete(), dbfind(), dblock(), dbunlock()等函数。

(3)扩充函数。有取日期、日期转换及字符串比较等函数。如 brtoday(), rdatestr(), rstrdate()等函数。

为了清楚地了解在 C 语言中,如何通过调用 ALL-II 库中的函数来管理数据库,现举一简单实例。该程序的功能:由屏幕上接收工号,对指定工号人员的工资级别上调一级,工资增加 10%。程序首部的 dbio.h 是 C 语言程序调用 ALL-II 例行程序库时所必备的前导文件。

```

#include"stdio.h"
#include"dbio.h"
struct dbview wk db []={
{"wk key"},
{"wk code"},
{"wk name"},
{"wk class"},
{"wk money"},
};
/* 文件视图 */
    
```

```

struct{
long wk__key;
char wk__code[8+1];
char wk__name[8+1];
int wk__class;
float wk__money;
}work;_ /* 文件结构 */
main()
{
int len,cc;
char code__str[9];
opendb(); /* 打开数据库及其文件
(省略) */
getstr(22,10,8,code__str); /* 由屏幕上接受"工号" */
cc = dbfind("wk__db",EQUAL,code__str,&len,&work); /*
用工号查找文件 */
if(cc == 0){
work.wk__money += work.wk__money * 0.10;
work.wk__class++;
dbupdate("wk__db",&work); /* 更新文件 */
}
else printf("not found!\n");
closedb(); /* 关闭数据库及其文件
(省略) */
return(0);
}
/* 其中 getstr()函数功能为:从屏幕上定位接受指定长度字符串(省略) */

```

在 Informix 3.30 环境下,对数据库的操作,是通过在 C 语言程序中调用 ALL-II 例行程序库提供的 C 函数实现的,接口简单,操作方便,易于使用。但也存在几个方面的问题:

- (1)没有采用 SQL 语言,兼容性、可移植性较差。
- (2)它是基于 286 环境下开发的数据库管理软件,效率较低。
- (3)虽能实现数据库的管理,但灵活性、功能方面不够理想。

关系数据库系统公司在 IBM 公司开发的 ANSI 标准结构查询语言(SQL)上加以扩充,开发了 RDSQL。它既是交互式的查询语言,又可在 Informix-ESQL/C 环境下,嵌入 C 语言应用程序中使用。

Informix-ESQL/C 是 RDS 公司提供的 C 语言接口,包括预处理程序、C 函数和前导文件。它允许程序员

在 C 语言程序中嵌入 RDSQL 语句来操纵数据库,或在用 RDSQL 语句书写的 PERFORM 屏幕事务处理程序及在 ACE 报表程序中,使用 C 函数。

采用 ALL-II 与 RDSQL 均可实现在 C 语言程序中对数据库管理,但在管理、数据操纵和实现方法等方面差异较大。

以下程序说明 RDSQL 语句在 C 程序中的使用方法:

```

#include <stdio.h>
#include sqlca;
#include sqlda;
main()
{
struct sqlda * demodesc;
char demoquery[80];
char dueryvalue[2];
char fname[20];
char lname[20];
printf(demoquery,"%S %S"
"select fname,lname from customer",
"where lname > ?");
printf(queryvalue,"C");
printf("DEM02 Sample ESQLprogram running.\n\n");
database stores;
prepare qid from $ demoquery;
declare democursor cursor for qid;
open democursor using $ queryvalue;
if(sqlca.sqlcode)
printf("%S %d\n",
"sqlca.code,after the cursor open,is",
sqlca.sqlcode);
for(;;)
{
$ fetch democursor into $ fname, $ lname;
if(sqlca.sqlcode)break;
printf("%S %d\n",
}
if(sqlca.sqlcode!= SQLNOTFOUND)
brentf("%s %d\n",
"sqlca.code,after fetch,is",
sqlca.sqlcode);
$ xlose democursor;
printf("\nProgram Over.\n");
}

```

该程序中由 "\$" 引导的均为 RDSQL 语句和

RDSQL 变量,程序从 stores 数据库的 customer 表中查找 lname 字段值大于“C”的记录,并显示相应的 lname, fname 字段内容。程序中使用了几个最常用的 SQL 语句,如 database、select、prepare、declarefetch 语句。

含有 RDSQL 语句的 C 程序使用扩展名“ec”表示。ec 程序必须调用 esql 预处理程序进行预编译,esql 程序将 ec 源程序中的 SQL 语句转成 C 代码,然后才能调用 C 编译程序,连接 ESQL / C 库函数和用户库函数,产生可执行代码文件。

与 Informix 3.30 相比较,Informix-ESQL / C 在数据库管理方面的优势主要有以下几点:

(1)事务处理功能:一个事务处理是指一个逻辑工作单位,事务处理是一系列你想对数据库完全实施或根本不实施的操作。具体的说,就是将一系列操作,例如在财会系统中,对总帐和分帐库进行更新操作,通过发出 BEGIN WORK 语句启动事务处理,使得其后的一系列操作作为一个逻辑工作单元,即一个整体来对待,如对这一系列操作产生的结果满意的话,可通过 COMMIT WORK 语句来正常结束事务处理,如对操作结果不满意,例如对其中的某个数据库文件更新失败,即可能出现多个数据文件数据不一致,允许用一条 ROLLBACK WORK 语句来结束事务处理,它将使数据库恢复到你发出 BEGIN WORK 语句前的状态。可以想象,若无事务处理功能,一旦更新有误,则用户必须采取显式的行动恢复已被更新的记录。

(2)自动索引:在执行包含两个表连接的 SELECT 语句,且在连接字段上又无索引,则 RDSQL 在执行连接操作前,对记录数目较大的表建立临时索引,检索完成后索引自行消除,这个功能对非索引字段检索的速度,有动态的改善。

(3)聚类索引:UNIX 系统和 DOS 系统均是按块从磁盘中获取信息,因而按索引顺序排列且物理地存放在同一块中的记录越多,使用索引连续提取速度就越快,通常数据文件中数据存放地物理顺序与索引不存在关系,通过聚类的方法,可以使表中的物理存放顺序同其索引顺序相一致,以加快检索速度。

(4)加锁策略:Informix-ESQL / C 提供了记录级加锁和表级加锁。RDSQL 隐含执行记录级加锁。如果其它用户在记录级给一个表的记录上锁,并且试图修改或删除当前记录,RDSQL 返回一个说明该记录已上锁的错误。如果用户试图待已上锁的记录解锁,可通过执行 SET LOCK MODE TO WAIT 语句来实现。

(5)动态 RDSQL 语句:Informix-ESQL / C 提供了动态定义 RDSQL 语句与处理的功能,它允许在程序执行过程中,动态输入 RDSQL 语句,进行随机检索等操作,这就使基于动态 RDSQL 语句上开发的程序更加灵活、实用和方便。

(6)缓冲区处理:可以将插入表中的记录或从表中选取出来的记录送入缓冲区中存放。

支持 SQL 语言的 Informix 版本提供了 sqlconv 和 dbload 实用程序,sqlconv 自动将原有的数据库转换成适合 RDSQL 的数据库,用户可以方便地将已用的 Informix 3.x 数据库卸出,然后使用 dbload 装载到 Informix-SQL 的数据库中,又不破坏原有的数据库。这就保证了系统升级的时候,数据资源不被浪费。

综上所述,Informix-ESQL / C 采用了 SQL 数据语言,因而它具有广泛的兼容性和可移植性,从编程角度出发,用户只需规定想干什么,而无需说明如何去干。即用户只需告诉需要什么数据,而并不具体规定获取数据的具体过程,对数据库中查找所期望的数据的“导航”过程,是由系统自动完成的,因而它为用户提供了一个高效率的工作环境。

参考文献:

[1.]何积功等,《〈汉字 INFORMIX 关系型数据库管理系统〉》,北京科学技术出版社

[2.]李达等,《〈INFORMIX-ESQL / C 程序员手册〉》,北京希望电脑公司

[3.]程瑜荣等,《〈INFORMIX 关系数据库管理系统〉》,清华大学出版社

[4.](美)C.J.戴特著,《〈SQL 语言与关系数据库管理系统 DB2〉》,电子工业出版社

[5.]金茂忠,《〈C 程序设计高级教程〉》,北京航空航天大学出版社