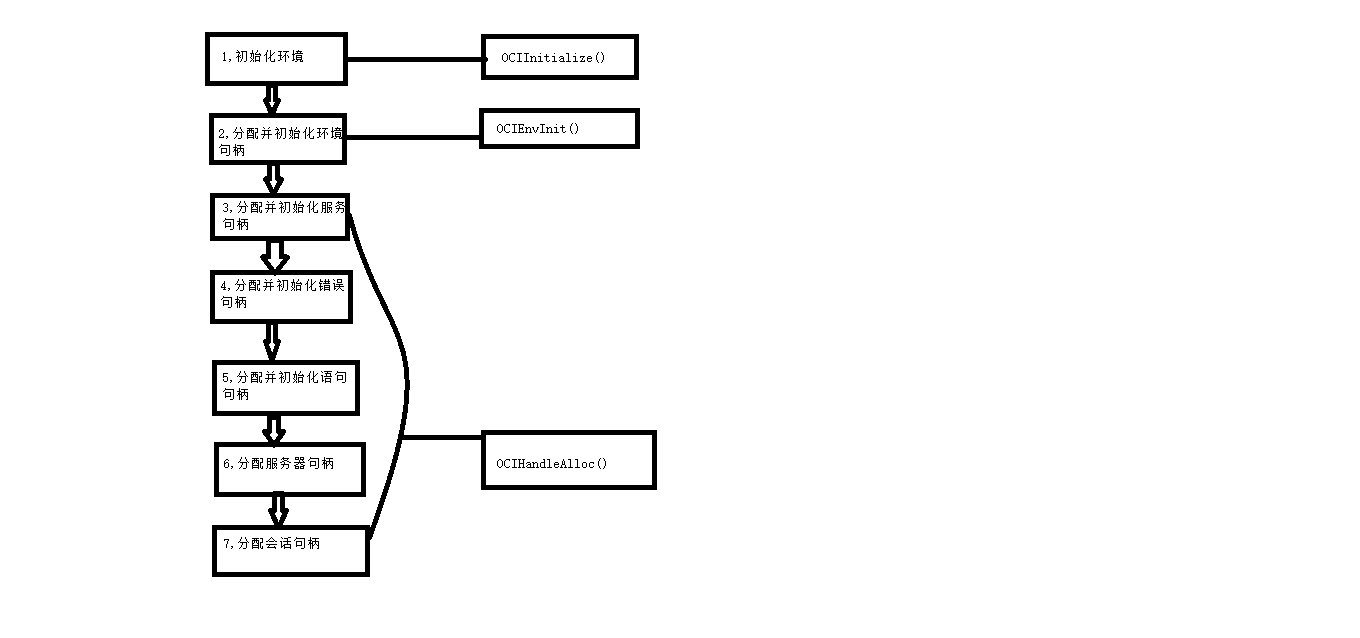
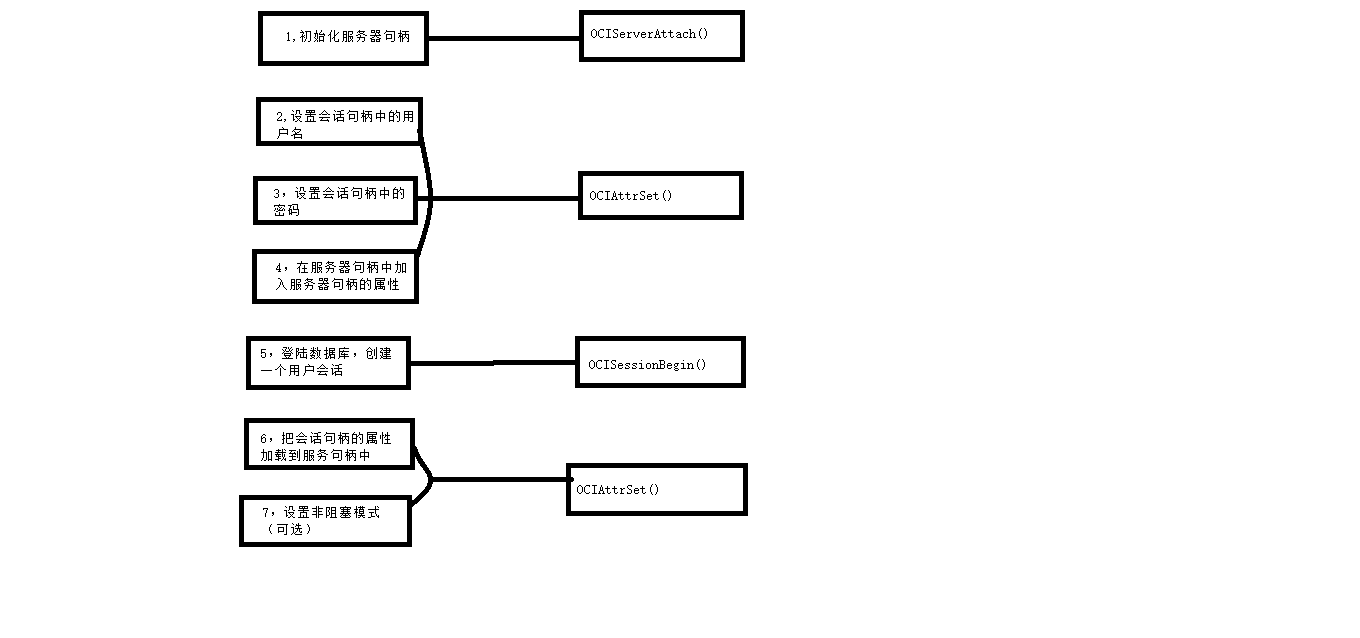
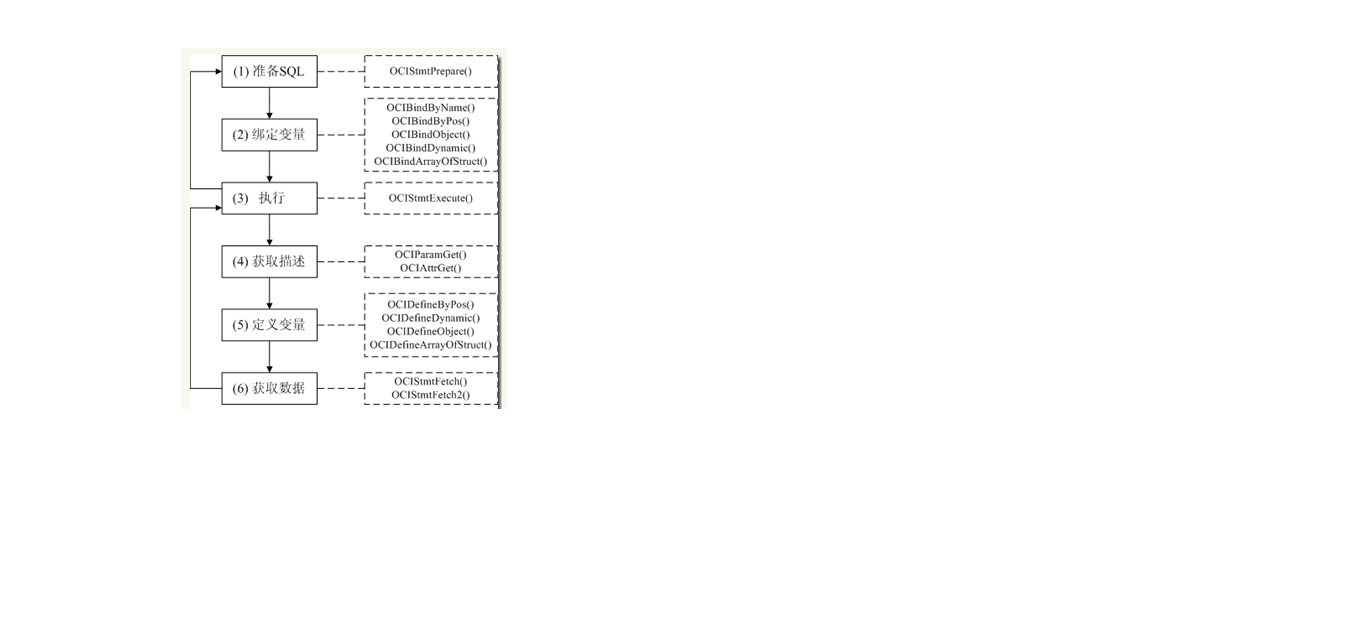
<https://blog.csdn.net/u011573853/article/details/50187139>

一，oci简介：   
OCI(Oracle Call Intedace，即0racle调用层接口)是Oracle公司提供的由头文件和库函数等组成的一个访问Oracle数据库的应用程序编程接口（application programming interface API），它允许开发人员在第三代编程语言（包括C, C++, COBOL 与 FORTRAN）中通过SQL（Structure Query Language）来操纵Oracle数据库，而且OCI在一定程度上支持第三代编程语言（诸如C, C++, COBOL 与 FORTRAN）的数据类型、语法等等。OCI的显著特点是全面支持Oracle的面向对象技术，同时OCI还具有如下的一些特点：   
1)高度控制应用程序的执行；   
2)允许开发人员应用已熟悉的第三代程序设计语言来应用OCI；   
3）可以内嵌到C代码中   
4)支持动态SQL；   
5)几乎所有的Oracle的开发工具都支持OCI；   
6)通过回调技术（callbacks）来实现动态绑定与定义；   
7)通过OCI的描述函数可以获取Oracle数据库的各种参数；   
8)增强了数组在DML(data manipulation language)语言中的应用；   
OCI接口支持Windows NT和Windows 95／98/2000/XP操作系统，它所支持的C语言编译器包括Borland C++和MiroSoft VisualC++等。在使用0CI开发Oralce数据库应用程序之前，应首先安装这些操作系统和C语言编译工具。在选择安装OCI开发工具包后，Oracle安装程序将0CI文件拷贝到oracle主目录内的以下子目录中：   
头文件oci.h在   
/home/oracle\_11/app/oracle/product/11.2.0/db\_1/rdbms/public/oci.h   
二，简单的说一下oci的工作步骤   
oci编程所需要的一些数据   
typedef struct   
{   
OCIEnv\* phOCIEnv ;//OCI环境句柄   
OCIError\* phOCIErr;//OCI错误句柄   
OCISvcCtx\* phOCISvctx;//服务上下文句柄   
OCIServer\* phOCIServe; /服务器上下文句柄/   
OCIStmt\* phOCIstmt; //语句句柄   
OCISession \* phSession; //会话句柄   
char\* DBName;//数据库服务名   
char\* UserName;//数据库用户名   
char\* Pwd;//数据库密码   
}OCIHP;   
2.1，分配和初始化一些句柄(如图)   


2.2，分配初始化好句柄以后进行连接数据库如图   
  
2.3，连接数据库以后就要执行一些sql语句，步骤如下   
数据库连接好后可以执行SQL语句：一条SQL语句在OCI应用程序中的执行步骤一般如下：(1)准备SQL语句。(2)在SQL语句中绑定需要输入到SQL语句中的变量。(3)执行SQL语句。(4)获取SQL中的输出描述。(5)定义输出变量。(6)获取数据。具体过程及过程中调用的函数如下图所示。对于SQL中的定义语句（如CREATE，DROP）和控制语句（如GRANT，REVOKE），由于没有数据的输入输出，只需要图2中第一步和第三步即可。操作语句（如INSERT，DELETE，UPDATE）则需要执行前三步。而查询语句（如SELECT）不仅可能有数据输入，而且也有数据的输出，因此需要执行六个步骤。   


三，常用函数解析可以到这里下载   
<http://download.csdn.net/detail/u011573853/9328969>

四，案例，本人写了一个很浅显的案例，实现了增删改查操作，使用绑定参数和不绑定两种方式完成的，适合我这样的新手看，高手飘过核心代码如下   
全部代码可以到此下载   
<http://download.csdn.net/detail/u011573853/9328997>

//插入数据(不绑定参数的)

void Oci\_insert(OCIHP\* ph,char \*sql)

{

printf("sql =%s\n",sql);

char sErrorMsg[1024];

sb4 sb4ErrorCode;

ub4 ub4RecordNo = 1;

int nRet = 0;

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//绑定参数插入

void Oci\_insert\_bang(OCIHP\* ph)

{

char sErrorMsg[1024];

sb4 sb4ErrorCode;

ub4 ub4RecordNo = 1;

int nRet = 0;

OCIBind\* bhp[10];

char id[5]="11";

char sname[15]="liuyupei";

int age=20;

char sex[]="v";

char sql[]="insert into stu(id,sname,age,sex) values(:Vhid,:Vhname,:Vhage,:Vhsex)";

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

//绑定输入参数变量

/\*

//把id 和:Vhid绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[0], ph->phOCIErr, (text \*) ":Vhid",strlen(":Vhid"), (ub1 \*)id , strlen(id)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","1参数绑定失败");

return ;

}

//把sname 和:Vhname绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[1], ph->phOCIErr, (text \*) ":Vhname",strlen(":Vhname"), (ub1 \*)sname , strlen(sname)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","2参数绑定失败");

return ;

}

//把age 和:Vhage绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[2], ph->phOCIErr, (text \*) ":Vhage",-1, (ub1 \*)&age , (sword)4,

SQLT\_INT, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","3参数绑定失败");

return ;

}

//把sex 和:Vhsex绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[3], ph->phOCIErr, (text \*) ":Vhsex",strlen(":Vhsex"), (ub1 \*)sex , strlen(sex)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","4参数绑定失败");

return ;

}\*/

//第二种绑定的方法

OCIBindByPos(ph->phOCIstmt, &bhp[0], ph->phOCIErr, 1,

(dvoid \*)id, sizeof(id), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0,OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[1], ph->phOCIErr, 2,

(dvoid \*)sname, sizeof(sname), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[2], ph->phOCIErr, 3,

(dvoid \*)&age, sizeof(int), SQLT\_INT,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[3], ph->phOCIErr, 4,

(dvoid \*)sex, sizeof(sex), SQLT\_STR, (dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//更新数据不绑定   
void Oci\_update(OCIHP\* ph,char \*sql)   
{

printf("sql =%s\n",sql);

char sErrorMsg[1024];

sb4 sb4ErrorCode;

ub4 ub4RecordNo = 1;

int nRet = 0;

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//更新数据绑定参数   
void Oci\_update\_bang(OCIHP\* ph)   
{   
char sErrorMsg[1024];   
sb4 sb4ErrorCode;   
ub4 ub4RecordNo = 1;   
int nRet = 0;   
OCIBind\* bhp[10];   
char id[5]=”11”;   
char sname[15]=”liweieieieei”;   
;

char sql[]="update stu set sname=':Vhname' where id=':Vhid'";

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

//绑定输入参数变量

/\*

//把id 和:Vhid绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[0], ph->phOCIErr, (text \*) ":Vhid",strlen(":Vhid"), (ub1 \*)id , strlen(id)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","1参数绑定失败");

return 0;

}

//把sname 和:Vhname绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[1], ph->phOCIErr, (text \*) ":Vhname",strlen(":Vhname"), (ub1 \*)sname , strlen(sname)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","2参数绑定失败");

return 0;

}

\*/

//第二种绑定的方法

OCIBindByPos(ph->phOCIstmt, &bhp[0], ph->phOCIErr, 2,

(dvoid \*)id, sizeof(id), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0,OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[1], ph->phOCIErr, 1,

(dvoid \*)sname, sizeof(sname), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//查询   
void Oci\_select(OCIHP\* ph)   
{   
int nRet = 0;   
ub4 ub4RecordNo = 1;   
OCIDefine \* bhp[10];   
char id[20];   
char sname[30];   
int age;   
char sex[20];

char sErrorMsg[1024];

sb4 sb4ErrorCode;

//char sname[10] ={0};

b2 sb2aIndid[30]; //指示器变量，用于取可能存在空值的字

char sql[]="select id,sname,age,sex from stu ";

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql,

(ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT); //准备SQL语句

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

//获取数据长度

ub2 datalen = 0;

//绑定输出参数

if(OCIDefineByPos(ph->phOCIstmt,&bhp[0],ph->phOCIErr, 1, (dvoid \*)&id, (ub4)sizeof(id),

SQLT\_STR/\*LBI long binary type \*/, &sb2aIndid[0], &datalen, NULL, OCI\_DEFAULT) !=0)

{

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);

printf("%s\n","1参数绑定失败");

return ;

}

if(OCIDefineByPos(ph->phOCIstmt,&bhp[1],ph->phOCIErr, 2,(dvoid \*)&sname, (ub4)sizeof(sname),

SQLT\_STR/\*LBI long binary type \*/, &sb2aIndid[1], &datalen, NULL, OCI\_DEFAULT) !=0)

{

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*)sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);

printf("%s\n","2参数绑定失败");

return ;

}

if(OCIDefineByPos(ph->phOCIstmt,&bhp[2],ph->phOCIErr, 3,(dvoid \*)&age, (ub4)4,   
SQLT\_INT/LBI long binary type /, NULL, &datalen, NULL, OCI\_DEFAULT) !=0)   
{   
//获取错误信息   
if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,   
sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)   
printf(“error msg:%s\n”, sErrorMsg);   
OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);   
printf(“%s\n”,”3参数绑定失败”);   
return ;

}

if(OCIDefineByPos(ph->phOCIstmt,&bhp[3],ph->phOCIErr, 4,(dvoid \*)&sex, (ub4)sizeof(sex),   
SQLT\_STR/LBI long binary type /, &sb2aIndid[3], &datalen, NULL, OCI\_DEFAULT) !=0)   
{   
//获取错误信息   
if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*)sErrorMsg,   
sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)   
printf(“error msg:%s\n”, sErrorMsg);   
OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);   
printf(“%s\n”,”4参数绑定失败”);   
return ;

}

//获取执行语句类型

ub2 stmt\_type;

OCIAttrGet ((dvoid \*)ph->phOCIstmt, (ub4)OCI\_HTYPE\_STMT, (dvoid \*)&stmt\_type, (ub4 \*)0, (ub4)OCI\_ATTR\_STMT\_TYPE, ph->phOCIErr);

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)(stmt\_type==OCI\_STMT\_SELECT?1:0),

(ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg, sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

// 利用游标提取信息

int rows\_fetched;

do   
{   
printf(“id=%s,sname=%s,age=%d,sex =%s\n”,id,(sb2aIndid[1]==-1?”NULL”:sname),age,sex);   
printf(“%d\n”,sb2aIndid[0]);   
printf(“%d\n”,sb2aIndid[1]);

}

while(OCIStmtFetch2(ph->phOCIstmt, ph->phOCIErr, 1, OCI\_FETCH\_NEXT, OCI\_FETCH\_NEXT, OCI\_DEFAULT) != OCI\_NO\_DATA);

// 获得记录条数

OCIAttrGet((CONST void )ph->phOCIstmt, OCI\_HTYPE\_STMT, (void )&rows\_fetched, (ub4 \*)sizeof(rows\_fetched),   
OCI\_ATTR\_ROW\_COUNT, ph->phOCIErr);   
printf(“总共记录数 %d\n”,rows\_fetched);   
}

//删除数据   
void Oci\_delete(OCIHP\* ph,char \*sql)   
{   
printf(“sql =%s\n”,sql);   
char sErrorMsg[1024];   
sb4 sb4ErrorCode;   
ub4 ub4RecordNo = 1;   
int nRet = 0;

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

简单的makefile：//插入数据(不绑定参数的)

void Oci\_insert(OCIHP\* ph,char \*sql)

{

printf("sql =%s\n",sql);

char sErrorMsg[1024];

sb4 sb4ErrorCode;

ub4 ub4RecordNo = 1;

int nRet = 0;

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//绑定参数插入

void Oci\_insert\_bang(OCIHP\* ph)

{

char sErrorMsg[1024];

sb4 sb4ErrorCode;

ub4 ub4RecordNo = 1;

int nRet = 0;

OCIBind\* bhp[10];

char id[5]="11";

char sname[15]="liuyupei";

int age=20;

char sex[]="v";

char sql[]="insert into stu(id,sname,age,sex) values(:Vhid,:Vhname,:Vhage,:Vhsex)";

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

//绑定输入参数变量

/\*

//把id 和:Vhid绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[0], ph->phOCIErr, (text \*) ":Vhid",strlen(":Vhid"), (ub1 \*)id , strlen(id)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","1参数绑定失败");

return ;

}

//把sname 和:Vhname绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[1], ph->phOCIErr, (text \*) ":Vhname",strlen(":Vhname"), (ub1 \*)sname , strlen(sname)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","2参数绑定失败");

return ;

}

//把age 和:Vhage绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[2], ph->phOCIErr, (text \*) ":Vhage",-1, (ub1 \*)&age , (sword)4,

SQLT\_INT, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","3参数绑定失败");

return ;

}

//把sex 和:Vhsex绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[3], ph->phOCIErr, (text \*) ":Vhsex",strlen(":Vhsex"), (ub1 \*)sex , strlen(sex)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","4参数绑定失败");

return ;

}\*/

//第二种绑定的方法

OCIBindByPos(ph->phOCIstmt, &bhp[0], ph->phOCIErr, 1,

(dvoid \*)id, sizeof(id), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0,OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[1], ph->phOCIErr, 2,

(dvoid \*)sname, sizeof(sname), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[2], ph->phOCIErr, 3,

(dvoid \*)&age, sizeof(int), SQLT\_INT,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[3], ph->phOCIErr, 4,

(dvoid \*)sex, sizeof(sex), SQLT\_STR, (dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//更新数据不绑定   
void Oci\_update(OCIHP\* ph,char \*sql)   
{

printf("sql =%s\n",sql);

char sErrorMsg[1024];

sb4 sb4ErrorCode;

ub4 ub4RecordNo = 1;

int nRet = 0;

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//更新数据绑定参数   
void Oci\_update\_bang(OCIHP\* ph)   
{   
char sErrorMsg[1024];   
sb4 sb4ErrorCode;   
ub4 ub4RecordNo = 1;   
int nRet = 0;   
OCIBind\* bhp[10];   
char id[5]=”11”;   
char sname[15]=”liweieieieei”;   
;

char sql[]="update stu set sname=':Vhname' where id=':Vhid'";

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

//绑定输入参数变量

/\*

//把id 和:Vhid绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[0], ph->phOCIErr, (text \*) ":Vhid",strlen(":Vhid"), (ub1 \*)id , strlen(id)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","1参数绑定失败");

return 0;

}

//把sname 和:Vhname绑定在一起

if ((nRet= OCIBindByName(ph->phOCIstmt, &bhp[1], ph->phOCIErr, (text \*) ":Vhname",strlen(":Vhname"), (ub1 \*)sname , strlen(sname)+1,

SQLT\_STR, (void \*) 0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT))!=OCI\_SUCCESS)

{

printf("%s\n","2参数绑定失败");

return 0;

}

\*/

//第二种绑定的方法

OCIBindByPos(ph->phOCIstmt, &bhp[0], ph->phOCIErr, 2,

(dvoid \*)id, sizeof(id), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0,OCI\_DEFAULT);

OCIBindByPos(ph->phOCIstmt, &bhp[1], ph->phOCIErr, 1,

(dvoid \*)sname, sizeof(sname), SQLT\_STR,(dvoid\*)0,(ub2 \*) 0, (ub2) 0, (ub4) 0, (ub4 \*) 0, OCI\_DEFAULT);

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

//查询   
void Oci\_select(OCIHP\* ph)   
{   
int nRet = 0;   
ub4 ub4RecordNo = 1;   
OCIDefine \* bhp[10];   
char id[20];   
char sname[30];   
int age;   
char sex[20];

char sErrorMsg[1024];

sb4 sb4ErrorCode;

//char sname[10] ={0};

b2 sb2aIndid[30]; //指示器变量，用于取可能存在空值的字

char sql[]="select id,sname,age,sex from stu ";

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql,

(ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT); //准备SQL语句

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

return ;

}

//获取数据长度

ub2 datalen = 0;

//绑定输出参数

if(OCIDefineByPos(ph->phOCIstmt,&bhp[0],ph->phOCIErr, 1, (dvoid \*)&id, (ub4)sizeof(id),

SQLT\_STR/\*LBI long binary type \*/, &sb2aIndid[0], &datalen, NULL, OCI\_DEFAULT) !=0)

{

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);

printf("%s\n","1参数绑定失败");

return ;

}

if(OCIDefineByPos(ph->phOCIstmt,&bhp[1],ph->phOCIErr, 2,(dvoid \*)&sname, (ub4)sizeof(sname),

SQLT\_STR/\*LBI long binary type \*/, &sb2aIndid[1], &datalen, NULL, OCI\_DEFAULT) !=0)

{

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*)sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);

printf("%s\n","2参数绑定失败");

return ;

}

if(OCIDefineByPos(ph->phOCIstmt,&bhp[2],ph->phOCIErr, 3,(dvoid \*)&age, (ub4)4,   
SQLT\_INT/LBI long binary type /, NULL, &datalen, NULL, OCI\_DEFAULT) !=0)   
{   
//获取错误信息   
if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,   
sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)   
printf(“error msg:%s\n”, sErrorMsg);   
OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);   
printf(“%s\n”,”3参数绑定失败”);   
return ;

}

if(OCIDefineByPos(ph->phOCIstmt,&bhp[3],ph->phOCIErr, 4,(dvoid \*)&sex, (ub4)sizeof(sex),   
SQLT\_STR/LBI long binary type /, &sb2aIndid[3], &datalen, NULL, OCI\_DEFAULT) !=0)   
{   
//获取错误信息   
if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*)sErrorMsg,   
sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)   
printf(“error msg:%s\n”, sErrorMsg);   
OCIHandleFree(ph->phOCIstmt, OCI\_HTYPE\_STMT);   
printf(“%s\n”,”4参数绑定失败”);   
return ;

}

//获取执行语句类型

ub2 stmt\_type;

OCIAttrGet ((dvoid \*)ph->phOCIstmt, (ub4)OCI\_HTYPE\_STMT, (dvoid \*)&stmt\_type, (ub4 \*)0, (ub4)OCI\_ATTR\_STMT\_TYPE, ph->phOCIErr);

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)(stmt\_type==OCI\_STMT\_SELECT?1:0),

(ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg, sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

// 利用游标提取信息

int rows\_fetched;

do   
{   
printf(“id=%s,sname=%s,age=%d,sex =%s\n”,id,(sb2aIndid[1]==-1?”NULL”:sname),age,sex);   
printf(“%d\n”,sb2aIndid[0]);   
printf(“%d\n”,sb2aIndid[1]);

}

while(OCIStmtFetch2(ph->phOCIstmt, ph->phOCIErr, 1, OCI\_FETCH\_NEXT, OCI\_FETCH\_NEXT, OCI\_DEFAULT) != OCI\_NO\_DATA);

// 获得记录条数

OCIAttrGet((CONST void )ph->phOCIstmt, OCI\_HTYPE\_STMT, (void )&rows\_fetched, (ub4 \*)sizeof(rows\_fetched),   
OCI\_ATTR\_ROW\_COUNT, ph->phOCIErr);   
printf(“总共记录数 %d\n”,rows\_fetched);   
}

//删除数据   
void Oci\_delete(OCIHP\* ph,char \*sql)   
{   
printf(“sql =%s\n”,sql);   
char sErrorMsg[1024];   
sb4 sb4ErrorCode;   
ub4 ub4RecordNo = 1;   
int nRet = 0;

//准备SQL语句

nRet= OCIStmtPrepare(ph->phOCIstmt, ph->phOCIErr, (text\*)sql, (ub4) strlen(sql), (ub4)OCI\_NTV\_SYNTAX, (ub4)OCI\_DEFAULT);

if(nRet)

{

printf("%s\n","准备SQL语句错误");

printf("OCIStmtPrepare() error:%d\n",nRet);

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","kaishi执行SQL语句");

nRet = OCIStmtExecute( ph->phOCISvctx, ph->phOCIstmt, ph->phOCIErr, (ub4)1, (ub4)0, (OCISnapshot \*) NULL,(OCISnapshot \*) NULL, (ub4) OCI\_DEFAULT); //执行SQL语句

if(nRet)

{

printf("%s\n","执行SQL语句错误");

printf("OCIStmtExecute() error:%d\n",nRet);

//获取错误信息

if (OCIErrorGet(ph->phOCIErr, ub4RecordNo++, NULL, &sb4ErrorCode, (OraText\*) sErrorMsg,

sizeof(sErrorMsg), OCI\_HTYPE\_ERROR) == OCI\_SUCCESS)

printf("error msg:%s\n", sErrorMsg);

return ;

}

printf("%s\n","执行SQL语句 OK");

Oci\_commit( ph);

return ;

}

简单的makefile：gcc myocitext.c -o myocitext -IORACLEHOME/rdbms/demo−IORACLEHOME/rdbms/demo−IORACLE\_HOME/rdbms/public -lclntsh

应注意库lclntsh的位置

本人也是新手，下面有不错的资料   
<http://www.cnblogs.com/ychellboy/archive/2010/04/16/1713884.html>   
<http://kulong0105.blog.163.com/blog/static/174406191201162145944574/>

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/u011573853/article/details/50187139