/\* Processed by ecpg (4.8.0) \*/

/\* These include files are added by the preprocessor \*/

#include <ecpglib.h>

#include <ecpgerrno.h>

#include <sqlca.h>

/\* End of automatic include section \*/

#line 1 "ConsoleApp.pgc"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sqlda.h>

//EXEC SQL include sqla.h;

sqlda\_t \*sqlda1;

sqlda\_t \*sqlda2;

/\* exec sql begin declare section \*/

#line 10 "ConsoleApp.pgc"

char search [ 20000 ] , query [ 20000 ] , x [ 20000 ] , username [ 200 ] , name [ 100 ] ;

#line 11 "ConsoleApp.pgc"

char c , \* ans ;

#line 12 "ConsoleApp.pgc"

const char \* password ;

#line 13 "ConsoleApp.pgc"

int i = 0 , intval ;

#line 14 "ConsoleApp.pgc"

long long longlongval ;

/\* exec sql end declare section \*/

#line 15 "ConsoleApp.pgc"

/\* exec sql whenever sqlerror call print\_sqlca ( ) ; \*/

#line 16 "ConsoleApp.pgc"

void print\_sqlca()

{

fprintf(stderr, "==== sqlca ====\n");

fprintf(stderr, "sqlcode: %ld\n", sqlca.sqlcode);

fprintf(stderr, "sqlerrm.sqlerrml: %d\n", sqlca.sqlerrm.sqlerrml);

fprintf(stderr, "sqlerrm.sqlerrmc: %s\n", sqlca.sqlerrm.sqlerrmc);

fprintf(stderr, "sqlerrd: %ld %ld %ld %ld %ld %ld\n", sqlca.sqlerrd[0],sqlca.sqlerrd[1],sqlca.sqlerrd[2],

sqlca.sqlerrd[3],sqlca.sqlerrd[4],sqlca.sqlerrd[5]);

fprintf(stderr, "sqlwarn: %d %d %d %d %d %d %d %d\n", sqlca.sqlwarn[0], sqlca.sqlwarn[1], sqlca.sqlwarn[2],

sqlca.sqlwarn[3], sqlca.sqlwarn[4], sqlca.sqlwarn[5],

sqlca.sqlwarn[6], sqlca.sqlwarn[7]);

fprintf(stderr, "sqlstate: %5s\n", sqlca.sqlstate);

fprintf(stderr, "===============\n");

}

void ExecuteQuerry()

{

printf("Input Your Query :\n");

scanf(" %[^\n]",query);

{ ECPGprepare(\_\_LINE\_\_, NULL, 0, "eq", query);

#line 36 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 36 "ConsoleApp.pgc"

/\* declare cur1 cursor for $1 \*/

#line 37 "ConsoleApp.pgc"

{ ECPGdo(\_\_LINE\_\_, 0, 1, NULL, 0, ECPGst\_normal, "declare cur1 cursor for $1",

ECPGt\_char\_variable,(ECPGprepared\_statement(NULL, "eq", \_\_LINE\_\_)),(long)1,(long)1,(1)\*sizeof(char),

ECPGt\_NO\_INDICATOR, NULL , 0L, 0L, 0L, ECPGt\_EOIT, ECPGt\_EORT);

#line 38 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 38 "ConsoleApp.pgc"

while(sqlca.sqlcode==0)

{

{ ECPGdo(\_\_LINE\_\_, 0, 1, NULL, 0, ECPGst\_normal, "fetch next from cur1", ECPGt\_EOIT,

ECPGt\_sqlda, &sqlda1, 0L, 0L, 0L,

ECPGt\_NO\_INDICATOR, NULL , 0L, 0L, 0L, ECPGt\_EORT);

#line 41 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 41 "ConsoleApp.pgc"

sqlda\_t \*cur\_sqlda;

for (cur\_sqlda = sqlda1;cur\_sqlda != NULL;cur\_sqlda = cur\_sqlda->desc\_next)

{

int i;

char name\_buf[1024];

char var\_buf[1024];

for (i = 0; i < cur\_sqlda->sqld; i++)

{

sqlvar\_t v = cur\_sqlda->sqlvar[i];

char \*sqldata = v.sqldata;

short sqllen = v.sqllen;

strncpy(name\_buf, v.sqlname.data, v.sqlname.length);

name\_buf[v.sqlname.length] = '\0';

switch (v.sqltype)

{

case ECPGt\_char:

memset(&var\_buf, 0, sizeof(var\_buf));

memcpy(&var\_buf, sqldata, (sizeof(var\_buf)<=sqllen ? sizeof(var\_buf)-1 : sqllen) );

break;

case ECPGt\_int: /\* integer \*/

memcpy(&intval, sqldata, sqllen);

snprintf(var\_buf, sizeof(var\_buf), "%d", intval);

break;

case ECPGt\_long\_long: /\* bigint \*/

memcpy(&longlongval, sqldata, sqllen);

snprintf(var\_buf, sizeof(var\_buf), "%lld", longlongval);

break;

default:

{

int i;

memset(var\_buf, 0, sizeof(var\_buf));

for (i = 0; i < sqllen; i++)

{

char tmpbuf[16];

snprintf(tmpbuf, sizeof(tmpbuf), "%02x ", (unsigned char) sqldata[i]);

strncat(var\_buf, tmpbuf, sizeof(var\_buf));

}

}

break;

}

printf("%s = %s\n", name\_buf, var\_buf);

}

printf("\n");

}

}

{ ECPGdo(\_\_LINE\_\_, 0, 1, NULL, 0, ECPGst\_normal, "close cur1", ECPGt\_EOIT, ECPGt\_EORT);

#line 94 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 94 "ConsoleApp.pgc"

{ ECPGtrans(\_\_LINE\_\_, NULL, "commit");

#line 95 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 95 "ConsoleApp.pgc"

}

void UpdateQuerry()

{

printf("Input Your Query : \n");

scanf(" %[^\n]",query);

//printf("%s\n",query);

{ ECPGdo(\_\_LINE\_\_, 0, 1, NULL, 0, ECPGst\_execute, query, ECPGt\_EOIT, ECPGt\_EORT);

#line 103 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 103 "ConsoleApp.pgc"

if(sqlca.sqlcode==0)

{

fprintf(stderr, "Successful\n");

}

}

int main()

{

printf("DBNAME: ");

scanf(" %[^\n]",name);

getchar();

printf("USERNAME: ");

scanf(" %[^\n]",username);

getchar();

strcpy(name,"201501185@10.100.71.21");

{ ECPGconnect(\_\_LINE\_\_, 0, name , username , NULL , NULL, 0);

#line 120 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 120 "ConsoleApp.pgc"

if(sqlca.sqlcode==0)

fprintf(stderr, "Connection Successful\n");

else

exit(1);

printf("Set your search path :\n");

scanf(" %[^\n]",search);

{ ECPGdo(\_\_LINE\_\_, 0, 1, NULL, 0, ECPGst\_normal, "set search\_path to $0",

ECPGt\_char,(search),(long)20000,(long)1,(20000)\*sizeof(char),

ECPGt\_NO\_INDICATOR, NULL , 0L, 0L, 0L, ECPGt\_EOIT, ECPGt\_EORT);

#line 129 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 129 "ConsoleApp.pgc"

if(sqlca.sqlcode==0)

fprintf(stderr,"Search\_path Successfully set\n");

else

{

fprintf(stderr,"No such path available\n");

exit(1);

}

while(1)

{

printf("Press 1 for INSERT,UPDATE and DELETE Querry.\nPress 2 for SELECT querry.\nPress anything else to exit.\n");

scanf(" %c",&c);

if(c=='1')

{

UpdateQuerry();

}

else if(c=='2')

{

ExecuteQuerry();

}

else

break;

}

{ ECPGdisconnect(\_\_LINE\_\_, "ALL");

#line 153 "ConsoleApp.pgc"

if (sqlca.sqlcode < 0) print\_sqlca ( );}

#line 153 "ConsoleApp.pgc"

return 0;

}