Using anonymous SP, CV and SV in ODBC C# Java

Version: 5.4

Document No: 54/DBM54-T03202017-01-ASCS
Author: DBMaster Test Team, Research & Development Division,
SYSCOM Computer Engineering CO.
Print Date: March 20, 2017

凌羣電腦
THE SYSCOM GROUP

Content

1. Concept Introduction

Anonymous stored procedure: The anonymous stored procedure can execute some statements in one block, it is the same with SQL Store procedure compound statements grammar, but it is not store information and execute more times in Database.

- CV：connection variable
- SV：statement variable

➲ Syntax

Create anonymous stored procedure:

```
BEGIN
   -sp_declare_main;
   -sp_statement_main;
END;
```

2. Anonymous Stored Procedure in ODBC

2.1 Using ODBC call anonymous stored procedure:

The following code fragment demonstrates creating tables and returns a result set.

```
strcpy( (char *)str1,
        "begin\n declare sv1 int default 10; \
         declare sv2 varchar(50) default 'DBMaker sv'; \
         drop table if exists testetb;\
         create table testetb(c1 int,c2 varchar(50)); \
         insert into testetb values(1,'DBMaker 1'); \
         insert into testetb values(2,'DBMaker 2'); \
         insert into testetb values(3,'DBMaker 3'); \
         insert into testetb values(sv1,sv2); \
         declare cur cursor with return for select * from testetb; \
         open cur; \
         end;"
      );
rc = SQLExecDirect(cmdp1, str1, SQL_NTS);
```

2.2 Using CV variables

The following code fragment shows how to define two CV (var1 and var2) is inserted into the table and return the desired result set.

```
strcpy((char *)str1, "declare set int @var1 = 100;");
rc = SQLExecDirect(cmdp2, str1, SQL_NTS);

strcpy((char *)str2, "declare set varchar(50) @var2 = 'DBMaker cv';");
rc = SQLExecDirect(cmdp2, str2, SQL_NTS);

strcpy((char *)str3, "insert into testetb values(@var1,@var2);");
rc = SQLExecDirect(cmdp2, str3, SQL_NTS);

strcpy((char *)str4, "select * from testetb where c1=@var1;");
rc = SQLExecDirect(cmdp2, str4, SQL_NTS);
```

2.3 Using SV variables

The following code fragment shows how to define two SV (sv1 and sv2) is inserted into the table and return the desired result set.

```
strcpy( (char *)str1,
"begin\n
declare sv1 int default 10; \
declare sv2 varchar(50) default 'DBMaker sv'; \
create table testetb(c1 int,c2 varchar(50)); \

insert into testetb values(sv1,sv2); \
declare cur cursor with return for select * from testetb; \
open cur; \
end;"
);
rc = SQLExecDirect(cmdp1, str1, SQL_NTS);
```

2.4 ODBC interface program

The following program is tested ok in Visual Studio 2013 environment.

```
#include <stdio.h>
#include <string.h>

#ifdef WIN32
#include <windows.h>
#else
#include "sqlunix.h"
#endif
#include "sqlext.h"
#include "sqlopt.h"

/* define some costant for ex1.c */
#define DSN      "DBSAMPLE5"
#define UID      "SYSADM"
#define PWD      ""

#define ntb1      4    /* number of tuples in testetb */
#define RTN_OK    0    /* return code OK          */

/* MACRO to show error message */
#define getErrMsg(rc, rightrc, xhenv, xhdbc, xcmdp)\
{\
UCHAR   outmsg[100],xstate[10];\
SWORD   size=100;\
SWORD   msglen;\
SDWORD  nativerc=0;\
SDWORD  xrc=0;\
xrc=SQLError(xhenv,xhdbc,xcmdp,xstate,&nativerc,outmsg,size,&msglen);\
if (xrc == SQL_SUCCESS || xrc == SQL_SUCCESS_WITH_INFO)\
   {\
   if (rightrc != nativerc) \
         {\
      printf("------------------ Error ------------------\n"\
```

```
            "Line %d\n"\
            "Unexpected return code: %ld\n"\
            "Error message: %s\n"\
            "-----------------------------------------\n", \
            __LINE__, nativerc, outmsg); \
      rc = nativerc;\
         }\
   }\
}

#define checkrc(rightrc, sheng, xhdbc, xcmdp)\
{\
SDWORD  rrc=0;\
getErrMsg(rrc, rightrc, xhenv, xhdbc, xcmdp)\
if (rrc != rightrc)\
   goto EXIT;\
}


/* MACRO to show error message */

void main()
{
    HDBC    hdbc;
    HSTMT   cmdp1, cmdp2;
    HENV    henv;
    SQLLEN  rc = 0, cbInt = 0, cbChar = SQL_NTS;

    UCHAR   str1[2000], str2[2000], str3[2000], str4[2000];
    SDWORD  bc1, bc3;
    UCHAR   bc2[2000], bc4[2000];
    SQLLEN  szbc1, szbc2, szbc3, szbc4;

   rc = SQLAllocEnv(&henv);
   if (rc) printf("allocate envrionment handle fail !! \n");
   rc = SQLAllocConnect(henv, &hdbc);
   if (rc) printf("allocate connection handle fail !! \n");

   rc = SQLConnect(hdbc, (UCHAR *)DSN, SQL_NTS, (UCHAR *)UID, SQL_NTS,(UCHAR *)PWD, SQL_NTS);
   checkrc(RTN_OK, NULL, hdbc, NULL);

  /*======    set autocommit off   =======*/
   rc = SQLSetConnectOption(hdbc, SQL_AUTOCOMMIT, SQL_AUTOCOMMIT_OFF);
   checkrc(RTN_OK, NULL, hdbc, NULL);

   rc = SQLAllocStmt(hdbc, &cmdp1);
   checkrc(RTN_OK, NULL, hdbc, NULL);
   rc = SQLAllocStmt(hdbc, &cmdp2);
   checkrc(RTN_OK, NULL, hdbc, NULL);

   printf("  1. Anonymous stored procedure\n");
   strcpy((char *)str1, "begin\n declare sv1 int default 10;\
                        declare sv2 varchar(50) default 'DBMaker sv';\
                         drop table if exists testetb;\
                         create table testetb(c1 int,c2 varchar(50));\
                         insert into testetb values(1,'DBMaker 1');\
                         insert into testetb values(2,'DBMaker 2');\
                         insert into testetb values(3,'DBMaker 3');\
```

```
                    insert into testetb values(sv1,sv2);\
                    declare cur cursor with return for select * from testetb;\
                    open cur;\
                end;"
                );

rc = SQLExecDirect(cmdp1, str1, SQL_NTS);
checkrc(RTN_OK, NULL, NULL, cmdp1);

printf("------------------------------------------------------------\n");
rc = SQLBindCol(cmdp1, 1, SQL_C_LONG, &bc1, sizeof(SWORD), &szbc1);
checkrc(RTN_OK, NULL, NULL, cmdp1);
rc = SQLBindCol(cmdp1, 2, SQL_C_CHAR, bc2, 2000, &szbc2);
checkrc(RTN_OK, NULL, NULL, cmdp1);

printf("    c1       c2  \n");
printf("  ======   ============\n");
while (rc == SQL_SUCCESS){
    rc = SQLFetch(cmdp1);
    checkrc(RTN_OK, NULL, NULL, cmdp1);
    if (rc == SQL_NO_DATA) break;
    printf("  %4d     %-10s\n", bc1, bc2);
};

printf("------------------------------------------------------------\n");
printf("  2. CV + multiple SQL statements\n");

memset((char *)str1, 0, strlen((char *)str1));

strcpy((char *)str1, "declare set int @var1 = 100;");
rc = SQLExecDirect(cmdp2, str1, SQL_NTS);
checkrc(RTN_OK, NULL, NULL, cmdp2);

strcpy((char *)str2, "declare set varchar(50) @var2 = 'DBMaker cv';");
rc = SQLExecDirect(cmdp2, str2, SQL_NTS);
checkrc(RTN_OK, NULL, NULL, cmdp2);

strcpy((char *)str3, "insert into testetb values(@var1,@var2);");
rc = SQLExecDirect(cmdp2, str3, SQL_NTS);
checkrc(RTN_OK, NULL, NULL, cmdp2);

strcpy((char *)str4, "select * from testetb where c1=@var1;");
rc = SQLExecDirect(cmdp2, str4, SQL_NTS);
checkrc(RTN_OK, NULL, NULL, cmdp2);

printf("------------------------------------------------------------\n");
rc = SQLBindCol(cmdp2, 1, SQL_C_LONG, &bc3, sizeof(SWORD), &szbc3);
checkrc(RTN_OK, NULL, NULL, cmdp2);
rc = SQLBindCol(cmdp2, 2, SQL_C_CHAR, bc4, 2000, &szbc4);
checkrc(RTN_OK, NULL, NULL, cmdp2);

printf("    c1       c2  \n");
printf("  ======   ============\n");
while (rc == SQL_SUCCESS){
    rc = SQLFetch(cmdp2);
    checkrc(RTN_OK, NULL, NULL, cmdp2);
    if (rc == SQL_NO_DATA) break;
    printf("  %4d     %-10s\n", bc3, bc4);
```

```
    };

    /*======  commit transaction and terminate db  ======*/
    printf("------------------------------------------------------------\n");
    rc = SQLTransact(henv, hdbc, SQL_COMMIT);
    checkrc(RTN_OK, NULL, hdbc, NULL);

EXIT:
    rc = SQLFreeStmt(cmdp1, SQL_DROP);
    if (rc) printf("free statement fail !! \n");
    rc = SQLFreeStmt(cmdp2, SQL_DROP);
    if (rc) printf("free statement fail !! \n");

    rc = SQLDisconnect(hdbc);
    if (rc) printf("disconnect database fail !!\n");

    /* Free the connection handle and environment handle      */
    rc = SQLFreeConnect(hdbc);
    if (rc) printf("free connection fail !! \n");
    rc = SQLFreeEnv(henv);
    if (rc) printf("free envrionment fail !! \n");

    return;

} /* end of main */
```

3. Anonymous Stored Procedure in C#

3.1 Using C# call anonymous stored procedure

The following code fragment demonstrates creating tables and returns a result set.

```
string cmdtxt = "BEGIN " +
" drop table if exists testetb; " +
" create table testetb (c1 int,c2 varchar(50)); " +
" insert into testetb values(1,'DBMaker 1'); " +
" declare cur cursor with return for select * from testetb; " +
" open cur; " +
" END";

OdbcCommand spCmd = conn.CreateCommand();
spCmd.CommandText = cmdtxt;
spCmd.CommandType = CommandType.StoredProcedure;

spCmd.ExecuteNonQuery();
```

3.2 Using CV variables

The following code fragment shows how to define two CV (var1 and var2) is inserted into the table and return the desired result set.

```
string cvVar1 = "declare set int @var1 = 100; ";
string cvVar2 = "declare set varchar(50) @var2 = 'DBMaker cv'";

new OdbcCommand(cvVar1,conn).ExecuteNonQuery();
new OdbcCommand(cvVar2, conn).ExecuteNonQuery();
new OdbcCommand("insert into testetb values(@var1,@var2)", conn).ExecuteNonQuery();

OdbcDataReader rdr = new OdbcCommand("select * from testetb where
c1=@var1",conn).ExecuteReader();
```

3.3 Using SV variables

The following code fragment shows how to define two SV (sv1 and sv2) is inserted into the table and return the desired result set.

```
string cmdtxt = "BEGIN " +
        " declare sv1 int default 10; " +
        " declare sv2 varchar(50) default 'DBMaker sv'; " +
        " drop table if exists testetb; " +
        " create table testetb(c1 int,c2 varchar(50)); " +
        " insert into testetb values(1,'DBMaker 1'); " +
        " insert into testetb values(sv1,sv2); " +
        " declare cur cursor with return for select * from testetb; " +
        " open cur; " +
        " END";
```

3.4 C# interface program

The following program is tested ok in Visual Studio 2013 environment.

```
using System;
using System.Data;
using System.Data.Odbc;

namespace OdbcCallSp
{
    class Program
    {
        public void RunStoredProc()
        {
            OdbcConnection conn = null;
            OdbcDataReader rdr1 = null;
            OdbcDataReader rdr2 = null;

            try
            {
                conn = new OdbcConnection("Driver={DBMaker 5.4
                Driver};Database=dbsample5;Uid=SYSADM;Pwd=;");
                conn.Open();

                Console.WriteLine("1. Anonymous stored procedure:\n");

                string cmdtxt = "BEGIN " +
                 "declare sv1 int default 10; " +
                 "declare sv2 varchar(50) default 'DBMaker sv';" +
                 "drop table if exists testetb; " +
                 "create table testetb(c1 int,c2 varchar(50)); " +
                 "insert into testetb values(1,'DBMaker 1'); " +
                 "insert into testetb values(2,'DBMaker 2'); " +
                 "insert into testetb values(3,'DBMaker 3'); " +
                 "insert into testetb values(sv1,sv2); " +
                 "declare cur cursor with return for select * from testetb; " +
                 "open cur; " +
                 "END";

                OdbcCommand spCmd = conn.CreateCommand();
                spCmd.CommandText = cmdtxt;
                spCmd.CommandType = CommandType.StoredProcedure;
```

```
                rdr1 = spCmd.ExecuteReader();
                while (rdr1.Read())
                {
                    Console.WriteLine("C1: {0,-10} C2: {1,2}", rdr1["c1"],
                rdr1["c2"]);
                }

                Console.WriteLine("\n2. CV + multiple SQL statements:\n");

                string cvVar1 = "declare set int @var1 = 100; ";
                string cvVar2 = "declare set varchar(50) @var2 = 'DBMaker cv'";

                new OdbcCommand(cvVar1, conn).ExecuteNonQuery();
                new OdbcCommand(cvVar2, conn).ExecuteNonQuery();
                new OdbcCommand("insert into testetb values(@var1,@var2)",
                conn).ExecuteNonQuery();
                rdr2 = new OdbcCommand("select * from testetb where c1=@var1",
                conn).ExecuteReader();
                while (rdr2.Read())
                {
                    Console.WriteLine("C1: {0,-10} C2: {1,2}", rdr2["c1"],
                    rdr2["c2"]);
                }

            }
            finally {
                if (conn != null)
                {
                    conn.Close();
                }
                if (rdr1 != null || rdr2 != null)
                {
                    rdr1.Close();
                    rdr2.Close();
                }
            }
        }

        static void Main(string[] args)
        {

            Program p = new Program();
            p.RunStoredProc();

            Console.WriteLine("\nPress Enter key to exit");
            Console.Read();
        }
    }
}
```

4. Anonymous Stored Procedure in JDBC

4.1 Using JDBC call Anonymous stored procedure

The following code fragment demonstrates creating tables and returns a result set.

```
String dmsql = "BEGIN " +
               " drop table if exists testetb; " +
               " create table testetb (c1 int,c2 varchar(50)); " +
               " insert into testetb values(1,'DBMaker 1'); " +
               " declare cur cursor with return for select C1,C2 from testetb; " +
               " open cur; " +
               " END";

Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(dmsql);
```

DBMaker anonymous stored procedures cannot use the following methods.

```
CallableStatement cstmt = conn.prepareCall(dmsql);
cstmt.executeQuery();
```

4.2 Using CV variables

The following code fragment shows how to define two CV(var1 and var2), is inserted into the table and return the desired result set.

```
String cvVar1 = "declare set int @var1 = 100;";
String cvVar2 = "declare set varchar(50) @var2 = 'DBMaker cv'";

Statement stmt = conn.createStatement();
stmt.executeUpdate(cvVar1);
stmt.executeUpdate(cvVar2);
stmt.executeUpdate("insert into testetb values( @var1 ,@var2)");
```

4.3 Using SV variables

The following code fragment shows how to define two SV (sv1 and sv2) is inserted into the table and return the desired result set.

```
String dmsql = "BEGIN " +
     " declare sv1 int default 10; " +
     " declare sv2 varchar(50) default 'DBMaker sv'; " +
     " drop table if exists testetb; " +
     " create table testetb(c1 int,c2 varchar(50)); " +
     " insert into testetb values(1,'DBMaker 1'); " +
     " insert into testetb values(sv1,sv2); " +
     " declare cur cursor with return for select C1,C2 from testetb; " +
     " open cur; " +
     " END";
```

4.4 JDBC interface program

```
import java.sql.*;

public class TestAnonymousSP {

     private static Connection conn = null;

     public static void main(String[] args) throws SQLException {

             String driver = "dbmaker.sql.JdbcOdbcDriver";
             String url = "jdbc:dbmaker:DBSAMPLE5";
             ResultSet rs = null;

             try {
                  Class.forName(driver);
                  conn = DriverManager.getConnection(url, "sysadm", "");

                  System.out.println("1. Anonymous stored procedure:");
                    String dmsql = "BEGIN " +
                           "declare sv1 int default 10; " +
                           "declare sv2 varchar(50) default 'DBMaker sv';" +
                           "drop table if exists testetb; " +
                           "create table testetb(c1 int,c2 varchar(50));" +
                           "insert into testetb values(1,'DBMaker 1');" +
                           "insert into testetb values(2,'DBMaker 2');" +
                           "insert into testetb values(3,'DBMaker 3');" +
                           "insert into testetb values(sv1,sv2);" +
                           "declare cur cursor with return for select C1,C2 from
                            testetb;" +
                           "open cur; " +
                           "END";

                    Statement stmt = conn.createStatement();
                    rs = stmt.executeQuery(dmsql);
                    while(rs.next())
                    {
                        System.out.println( rs.getInt("C1") + " |  " +
                        rs.getString("C2") );             }

                    rs.close();
                    stmt.close();
```

```
                        System.out.println("\n2. CV + multiple SQL statements:");
                        String cvVar1 = "declare set int @var1 = 100; ";
                        String cvVar2 = "declare set varchar(50) @var2 = 'DBMaker
cv'";

                        stmt = conn.createStatement();
                        stmt.executeUpdate(cvVar1);
                        stmt.executeUpdate(cvVar2);
                        stmt.executeUpdate("insert into testetb
                       values( @var1 ,@var2)");

                        String sqlStr="select * from testetb where c2=@var2";
                        rs= stmt.executeQuery(sqlStr);
                          while(rs.next())
                          {
                             System.out.println(rs.getInt(1) + " |  "+
                             rs.getString("C2"));
                          }

                          rs.close();
                          stmt.close();

                       } catch (Exception ex) {
                           ex.printStackTrace();
                       }finally {
                           if (conn != null)
                               conn.close();
                            }
                       }
   }
```