SQL with C

Test Program 1: Select a row with one column

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

EXEC SQL BEGIN DECLARE SECTION;
      VARCHAR userid[20];
      VARCHAR passwd[20];
      int value;
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE SQLCA.H;

main () {

      strcpy ((char *)userid.arr, "mmani");
      userid.len = strlen ((char *)userid.arr);
      strcpy ((char *)passwd.arr, "mmani");
      passwd.len = strlen ((char *)passwd.arr);

      EXEC SQL CONNECT :userid IDENTIFIED BY :passwd;

      EXEC SQL select max (b) into :value from r;

      printf ("connected\n");

      printf ("max (b) = %d\n", value);
}
```

Ensure that your library path is set (refer slides), and make it, and then execute it. Note that we need a table called r with at least one column called b for this.

Test Program 2: Select a row with multiple columns

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

EXEC SQL BEGIN DECLARE SECTION;
      VARCHAR userid[20];
      VARCHAR passwd[20];
      int val1, val2;
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE SQLCA.H;

void sqlerror ();
void sqlwarning ();

main () {

      strcpy ((char *)userid.arr, "mmani");
```

```
      userid.len = strlen ((char *)userid.arr);
      strcpy ((char *)passwd.arr, "mmani");
      passwd.len = strlen ((char *)passwd.arr);

      EXEC SQL WHENEVER SQLERROR DO sqlerror ();
      EXEC SQL WHENEVER SQLWARNING DO sqlwarning ();

      EXEC SQL CONNECT :userid IDENTIFIED BY :passwd;

      EXEC SQL SELECT a, b into :val1, :val2 from r where a = 2 and b =
10;

      printf ("val1 = %d::val2 = %d\n", val1, val2);

}

void sqlerror () {
      printf ("stop error:\t%25i\n", sqlca.sqlcode);
      return;
}

void sqlwarning () {
      printf ("stop warning:\t%25i\n", sqlca.sqlcode);
      return;
}
```

Test Program 3: Introducing Cursors

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

EXEC SQL BEGIN DECLARE SECTION;
      VARCHAR userid[20];
      VARCHAR passwd[20];
      int pnumber;
      char pname [30];
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE SQLCA.H;

void sqlerror ();
void sqlwarning ();

main () {

      strcpy ((char *)userid.arr, "mmani");
      userid.len = strlen ((char *)userid.arr);
      strcpy ((char *)passwd.arr, "mmani");
      passwd.len = strlen ((char *)passwd.arr);

      EXEC SQL WHENEVER SQLERROR DO sqlerror ();
      EXEC SQL WHENEVER SQLWARNING DO sqlwarning ();
      EXEC SQL WHENEVER NOT FOUND DO BREAK;

      EXEC SQL CONNECT :userid IDENTIFIED BY :passwd;
```

```
      EXEC SQL DECLARE myCursor CURSOR FOR select pNumber, pName from
professor;

      EXEC SQL OPEN myCursor;
      while (1) {
            EXEC SQL FETCH myCursor INTO :pnumber, :pname;

            printf ("number = %d::name = %s\n", pnumber, pname);
      }
      EXEC SQL CLOSE myCursor;

}

void sqlerror () {
      printf ("stop error:\t%25i\n", sqlca.sqlcode);
      return;
}

void sqlwarning () {
      printf ("stop warning:\t%25i\n", sqlca.sqlcode);
      return;
}
```

Test Program 4: Updating with a cursor

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

EXEC SQL BEGIN DECLARE SECTION;
      VARCHAR userid[20];
      VARCHAR passwd[20];
      int pnumber;
      char pname [30];
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE SQLCA.H;

void sqlerror ();
void sqlwarning ();

main () {

      int n = 0;

      strcpy ((char *)userid.arr, "mmani");
      userid.len = strlen ((char *)userid.arr);
      strcpy ((char *) passwd.arr, "mmani");
      passwd.len = strlen ((char *)passwd.arr);

      EXEC SQL WHENEVER SQLERROR DO sqlerror ();
      EXEC SQL WHENEVER SQLWARNING DO sqlwarning ();
      EXEC SQL WHENEVER NOT FOUND DO BREAK;

      EXEC SQL CONNECT :userid IDENTIFIED BY :passwd;
```

```
      EXEC SQL DECLARE myCursor CURSOR FOR select pNumber, pName from
professor FOR UPDATE OF pnumber;

      EXEC SQL OPEN myCursor;
      while (1) {

            EXEC SQL FETCH myCursor INTO :pnumber, :pname;

            pnumber = pnumber + 1;
            printf ("number = %d::name = %s\n", pnumber, pname);
            EXEC SQL UPDATE professor SET pnumber = :pnumber WHERE
CURRENT OF myCursor;

      }
      EXEC SQL CLOSE myCursor;
      EXEC SQL commit;

}

void sqlerror () {
      printf ("stop error:\t%25i\n", sqlca.sqlcode);
      exit (0);
}

void sqlwarning () {
      printf ("stop warning:\t%25i\n", sqlca.sqlcode);
      return;
}
```

Test Program 5: DELETING Using cursors

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

EXEC SQL BEGIN DECLARE SECTION;
      VARCHAR userid[20];
      VARCHAR passwd[20];
      int pnumber;
      char pname [30];
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE SQLCA.H;

void sqlerror ();
void sqlwarning ();

main () {

      int n = 0;

      strcpy ((char *) userid.arr, "mmani");
      userid.len = strlen ((char *) userid.arr);
      strcpy ((char *) passwd.arr, "mmani");
      passwd.len = strlen ((char *) passwd.arr);
```

```
      EXEC SQL WHENEVER SQLERROR DO sqlerror ();
      EXEC SQL WHENEVER SQLWARNING DO sqlwarning ();
      EXEC SQL WHENEVER NOT FOUND DO BREAK;

      EXEC SQL CONNECT :userid IDENTIFIED BY :passwd;

      EXEC SQL DECLARE myCursor CURSOR FOR select pNumber, pName from
professor FOR UPDATE;

      EXEC SQL OPEN myCursor;
      while (1) {

            EXEC SQL FETCH myCursor INTO :pnumber, :pname;

            pnumber = pnumber + 1;
            printf ("number = %d::name = %s\n", pnumber, pname);
            if (pnumber >= 7) {
                  EXEC SQL DELETE FROM professor WHERE CURRENT OF
myCursor;
            }

      }
      EXEC SQL CLOSE myCursor;
      EXEC SQL commit;

}

void sqlerror () {
      printf ("stop error:\t%25i\n", sqlca.sqlcode);
      exit (0);
}

void sqlwarning () {
      printf ("stop warning:\t%25i\n", sqlca.sqlcode);
      return;
}
```

Test Program 6: Handling Null values and also introducing scrollable cursors

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

EXEC SQL BEGIN DECLARE SECTION;
      VARCHAR userid[20];
      VARCHAR passwd[20];
      int pnumber;
      char pname [30];
      short isNullNumber;
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE SQLCA.H;

void sqlerror ();
void sqlwarning ();
```

```
main () {

      strcpy ((char *) userid.arr, "mmani");
      userid.len = strlen ((char *)userid.arr);
      strcpy ((char *)passwd.arr, "mmani");
      passwd.len = strlen ((char *)passwd.arr);

      EXEC SQL WHENEVER SQLERROR DO sqlerror ();
      EXEC SQL WHENEVER SQLWARNING DO sqlwarning ();
      EXEC SQL WHENEVER NOT FOUND DO BREAK;

      EXEC SQL CONNECT :userid IDENTIFIED BY :passwd;

      EXEC SQL DECLARE myCursor SCROLL CURSOR FOR select sNumber, sName
from student;

      EXEC SQL OPEN myCursor;
      while (1) {
            EXEC SQL FETCH RELATIVE 2 myCursor INTO
:pnumber:isNullNumber, :pname;

            if (isNullNumber == -1) { printf ("NULL value\n");
continue; }

            printf ("number = %d::name = %s\n", pnumber, pname);
      }
      EXEC SQL CLOSE myCursor;

}

void sqlerror () {
      /* printf ("stop error:\t%25i\n", sqlca.sqlcode); */
      return;
}

void sqlwarning () {
      printf ("stop warning:\t%25i\n", sqlca.sqlcode);
      return;
}
```