

DATABASE MANAGEMENT SYSTEMS

PROGRAM TO CONNECT TO THE DATABASES THROUGH CONNECTIVITY SERVERS

NAME : SIDDHARRTH PRIYADHARSAN

ROLL : 14CS10046

LANGUAGE : C

OPERATING SYSTEM : Ubuntu 12.04

COMPILER : GCC

The following packages are installed for this assignment :

- 1) mySQL (sudo apt-get install mySQL-server)
- 2) libmysqlclient15-dev (sudo apt-get install libmysqlclient15-dev)

The following commands are used to run the program (in order) :

- 1) First compile the program using the below command :

```
gcc -o assignment2 $(mysql_config --cflags) assignment2.c $(mysql_config --libs)
```

“assignment2.c” is the name of the file which consists the code. After compilation, an executable file in the name “assignment2” will be generated in the same directory where the C file is located.

- 2) Run the program using the above generated executable file “assignment2” using the below command :

```
./assignment2
```

Code of “assignment2.c” :

```
#include <mysql.h>
#include <my_global.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

// server and database details
struct connection_details
{
    char *server;
    char *user;
    char *password;
    char *database;
};

// function to perform queries
MYSQL_RES* mysql_perform_query(MYSQL *connection, char *sql_query)
{
    // send the query to the database
    if (mysql_query(connection, sql_query))
    {
        printf("MySQL query error : %s\n", mysql_error(connection));
        exit(1);
    }

    return mysql_use_result(connection);
}

// setting up the connection
MYSQL* mysql_connection_setup(struct connection_details mysql_details)
{
    // first of all create a mysql instance and initialize the
    variables within
    MYSQL *connection = mysql_init(NULL);

    // connect to the database with the details attached.
    if (!mysql_real_connect(connection, mysql_details.server,
mysql_details.user, mysql_details.password, mysql_details.database, 0,
NULL, 0)) {
        printf("Conection error : %s\n", mysql_error(connection));
        exit(1);
    }
    return connection;
}
```

```

}

// main function
int main()
{
    MYSQL_FIELD *field; // attribute fields
    MYSQL *conn;         // the connection
    MYSQL_RES *res;      // the results
    MYSQL_ROW row;       // the results row (line by line)

    struct connection_details mysqlD;

    // memory allocation for the server details
    mysqlD.server = (char*)malloc(20*sizeof(char));
    mysqlD.user = (char*)malloc(20*sizeof(char));
    mysqlD.password = (char*)malloc(20*sizeof(char));
    mysqlD.database = (char*)malloc(20*sizeof(char));

    strcpy(mysqlD.server, "10.5.18.101"); // where the mysql database is
    strcpy(mysqlD.user, "14CS10046");     // the root user of mysql
    strcpy(mysqlD.password, "btech14");   // the password of the root
                                         // user in mysql
    strcpy(mysqlD.database, "14CS10046"); // the database to pick

    // connect to the mysql database
    conn = mysql_connection_setup(mysqlD);

    char* temp = (char*)malloc(3000*sizeof(char));

    // 1st query
    strcpy(temp, "SELECT Course.CourseID, Course.Name, Course.Credits
FROM Teacher, Enrollment, Course WHERE Teacher.TeacherID =
Enrollment.TeacherID AND Course.CourseID = Enrollment.CourseID AND
Teacher.Name='PPC'");

    printf("\nThe result for the 1st SQL Query (given below) is as
follows :\n\n %s \n \n", temp);

    // assign the results return to the MYSQL_RES pointer
    res = mysql_perform_query(conn, temp);

    printf("\n-----\n\n");
    while ((row = mysql_fetch_row(res)) != NULL) {

```

```

int i;

for (i=0; i<mysql_num_fields(res); i++){
    if (i == 0)
    {
        while(field = mysql_fetch_field(res))
        {
            printf("%s\t", field->name);

            printf("\n-----|");
            printf("\n");
        }

        printf("%s\t", row[i] ? row[i] : "NULL");
    }

    //end of while

    printf("\n-----|");
    printf("\n\n");
    // clean up the database result set
    mysql_free_result(res);

    // 2nd query
    strcpy(temp, "SELECT DISTINCT Student.Rollno, Student.Name,
Student.Department FROM Student,Teacher,Enrollment,Gradesheet,Course
WHERE Teacher.TeacherID = Enrollment.TeacherID AND
Gradesheet.CourseID=Enrollment.CourseID AND
Enrollment.CourseID=Course.CourseID AND
Gradesheet.Rollno=Student.Rollno AND Teacher.Name='PPC'");

    printf("\nThe result for the 2nd SQL Query (given below) is as
follows :\n\n %s \n\n",temp);

    // assign the results return to the MYSQL_RES pointer
    res = mysql_perform_query(conn,temp);

    printf("\n-----|");
    printf("\n");
    while ((row = mysql_fetch_row(res)) !=NULL){

        int i;

```

```

    for (i=0; i<mysql_num_fields(res); i++){

        if (i == 0)
        {
            while(field = mysql_fetch_field(res))
            {
                printf("%s      ", field->name);

                printf("\n-----|");
            }
            printf("\n");

            printf("%s      ", row[i] ? row[i] : "NULL");

        }

    } //end of while

    printf("\n-----|");
    // clean up the database result set
    mysql_free_result(res);

    //3rd query
    strcpy(temp, "SELECT Course.Name, ClassTimings.Time, ClassTimings.Day
FROM Classroom,Course,ClassTimings WHERE ClassTimings.CourseID =
Course.CourseID AND ClassTimings.ClassID = Classroom.ClassID AND
ClassRoom.ClassID = 'NC142'");

    printf("\nThe result for the 3rd SQL Query (given below) is as
follows :\n\n %s \n \n",temp);

    // assign the results return to the MYSQL_RES pointer
    res = mysql_perform_query(conn,temp);

    printf("\n-----|");
    while ((row = mysql_fetch_row(res)) !=NULL){

        int i;

        for (i=0; i<mysql_num_fields(res); i++){

            if (i == 0)
            {
                while(field = mysql_fetch_field(res))

```

```

        {
            printf("%s          ", field->name);
        }

        printf("\n-----|");
        printf("\n");
    }

    printf("%s          ", row[i] ? row[i] : "NULL");

}

} //end of while

printf("\n-----|");
// clean up the database result set
mysql_free_result(res);

// 4th query
strcpy(temp, "SELECT Student.Name, Student.Department, T2.maximum
FROM Student INNER JOIN (SELECT DISTINCT Rollno, T1.maximum FROM
Gradesheet INNER JOIN (SELECT Course.CourseID, MAX(Gradesheet.Mark) AS
maximum FROM Course, Teacher, Gradesheet, Enrollment WHERE
Teacher.TeacherID = Enrollment.TeacherID AND Course.CourseID =
Gradesheet.CourseID AND Enrollment.CourseID = Course.CourseID AND
Teacher.Name = 'PPC' GROUP BY Course.CourseID ) T1 ON Gradesheet.Mark =
T1.maximum AND Gradesheet.CourseID = T1.CourseID) T2 ON Student.Rollno
= T2.Rollno ");

printf("\nThe result for the 4th SQL Query (given below) is as
follows :\n\n %s \n \n", temp);

// assign the results return to the MYSQL_RES pointer
res = mysql_perform_query(conn, temp);

printf("\n-----|");
while ((row = mysql_fetch_row(res)) != NULL) {

    int i;

    for (i=0; i<mysql_num_fields(res); i++){

        if (i == 0)
        {
            while(field = mysql_fetch_field(res))
            {

```

```

        printf("%s      ", field->name);
    }

    printf("\n-----|
-----|
");
    printf("\n");
}

    printf("%s      ", row[i] ? row[i] : "NULL");

}

} //end of while

    printf("\n-----|
-----|
");
    // clean up the database result set
    mysql_free_result(res);

//5th query
    strcpy(temp, "SELECT Student.Name, Student.Rollno, COUNT(Grade) AS
No_of_EX FROM Student, Gradesheet WHERE Student.Rollno =
Gradesheet.Rollno AND Gradesheet.Grade ='EX' GROUP BY Student.Rollno
HAVING No_of_EX = (SELECT MAX(EXCount) as maxi FROM (SELECT
Student.Rollno, COUNT(Grade) AS EXCount FROM Student, Gradesheet WHERE
Student.Rollno = Gradesheet.Rollno AND Gradesheet.Grade ='EX' GROUP BY
Student.Rollno)T1) ");

    printf("\nThe result for the 5th SQL Query (given below) is as
follows :
\n\n %s \n\n", temp);

    // assign the results return to the MYSQL_RES pointer
    res = mysql_perform_query(conn, temp);

    printf("\n-----|
-----|
");
    while ((row = mysql_fetch_row(res)) != NULL) {

        int i;

        for (i=0; i<mysql_num_fields(res); i++){

            if (i == 0)
            {
                while(field = mysql_fetch_field(res))
                {
                    printf("%s      ", field->name);

                }

                printf("\n-----|
-----|
");

```

```

        printf("\n");
    }

    printf("%s\t", row[i] ? row[i] : "NULL");

}

} //end of while

printf("\n-----|
-----| \n\n ");
// clean up the database result set
mysql_free_result(res);

// clean up the database link
mysql_close(conn);

return 0;
}
// end of main function

```

FUNCTIONS:

1) mysql_connection_setup(struct connection_details) :

This function takes up the details of the server connection which includes server address, username, password, database name as arguments and connects to the server using **mysql_real_connect()** function and returns a pointer to the corresponding MYSQL object, if the connection is successful. If not, it prints the corresponding error message.

2) mysql_perform_query(MYSQL* connection,char *sql_query) :

This function takes up the connection object and SQL query (in the form of string) as arguments and executes the SQL query by sending

it to the database using `mysql_query()` function and return the result of the query in the form of a pointer to MYSQL_RES object using `mysql_use_result()` function, if the query is valid (in terms of both syntax and logic). If not, it will print the corresponding error message.

3) MYSQL library functions : (`#include<mysql.h>` needed)

This includes :

- a) `mysql_free_result()` : To clean up the database result set.
- b) `mysql_fetch_row()` : To retrieve every tuple of the resultant table.
- c) `mysql_num_fields()` : To return the number of attributes present in the resultant table.
- d) `mysql_fetch_field()` : To retrieve every attribute of the resultant table.
- e) `mysql_close()` : To close the connection.