# Quick-Start Guide on MySQL C-API

EECS 495 – Fall 2016 – Prof. Peter Scheuermann

## Basic Functionality using C-API

### How to start:

```
#include "stdafx.h"     // for windows users
#include <windows.h>    // for windows users
#include "mysql.h"      // you must copy this from MySQL folder to your
                        // project folder
#include <iostream>

using namespace std;

MYSQL *conn;    /* pointer to connection handler */

int main ( int argc, char *argv[] )
{
    conn = mysql_init ( NULL );

    // open connection
    mysql_real_connect (
            conn,           /* pointer to connection handler */
            "localhost",    /* host to connect to */
            "user_name",    /* user name */
            "password",     /* password */
            "test",         /* database to use */
            0,              /* port (default 3306) */
            NULL,           /* socket or /var/lib/mysql.sock */
            CLIENT_MULTI_RESULTS );            /* flags (none) */

    // close connection
    mysql_close ( conn );
    return 0;
}
```

**Note:**

*CLENT_MULTI_RESULTS flag is required in order to obtain results from a call to a stored procedure; without it, it will return an empty set, or null;*

**Note:**

**Ił at compilation it complaints that it cannot find “mysql.h” or other related header files:**

*You must copy the all the header files (.h) from the MySQl directory (mysql\include\*.h) to the project directory of Visual C++ (Windows). Under linux, search for these header files under the filesystem (particularly, look for “mysql.h”)*

## How to submit a plain SQL query and display the query results

```
// Submit query
MYSQL_RES
*res_set;
MYSQL_ROW row;
mysql_query(conn,"SELECT * FROM Students;");
res_set = mysql_store_result(conn);
int numrows = (int)mysql_num_rows(res_set);


// Display results
for (int i = 0; i < numrows; i++)
    {
        row = mysql_fetch_row( res_set );
        if( row != NULL )
        {
            cout << "ID  : " << row[0] <<
            endl; cout << "Name: " << row[1]
            << endl;
        }
    }
// free resources
mysql_free_result( res_set );
```

## How to submit a query using stored procedures

**Note:**

*Stored routines require the* **`proc`** *table in the* `mysql` *database. This table is created during the MySQl 5.0 installation procedure. If you are upgrading to MySQl 5.0 from an earlier version, be sure to update your grant tables to make sure that the* `proc` *table exists*

Step1. Create the store procedure in MySQL. Login to MySQL using the command−line. Type the following (as example)

```
DELIMITER //
CREATE PROCEDURE
listStudents() BEGIN
        SELECT * FROM students;
END //
DELIMITER ;
```

**Note**

*If you are using the mysql command-line utility, pay careful attention to this note.*

*The default MySQL statement delimiter is ; (as you have seen in all of the MySQL statement used thus far). However, the mysql command-line utility also uses ; as a delimiter. If the command-line utility were to interpret the ; characters inside of the stored procedure itself, those would not end up becoming part of the stored procedure, and that would make the SQL in the stored procedure syntactically invalid.*