Building MySQL Applications

Introduction

This guide briefly describes the requirements for building applications that utilize MySQL databases. During this course, you will be provided access to several databases that reside on the Computer Science department's MySQL server, located on *sunapee*. You will be asked to write client applications to access these databases. **You should not code or test your applications on the *sunapee* system; it may adversely affect the performance of the machine for other users.** Instead, you will have access, either directly or through SSH, to other CS systems, including those in Lab 101. A full list of available machines can be found at `http://www.cs.dartmouth.edu/~wbc/suditour/index.html` (only accessible on campus or via VPN).

Using CS Systems

MySQL provides APIs for many languages, including Java, C, and Python. The CS systems' machines have all the necessary libraries to begin writing applications in these languages. The general workflow is the same for each language: connect to a database, execute a query, and then process the results.

Java

To begin writing Java applications, you will need the MySQL Java Connector JAR file, located at `https://dev.mysql.com/downloads/connector/j/`. You will need to include this JAR file on your classpath each time you compile and run your application. There are two methods to do this:

1. Use the **-cp** or **-classpath** command line option.

   - ```
     javac dbexample.java
     java -cp .:/usr/share/java/mysql-connector-java.jar <java-files>
     ```
   - *Note: You must specify the FULL path to the connector jar file. Using ~ or . will not work.*

2. Define the **CLASSPATH** environment variable.

   - ```
     export CLASSPATH=".:/usr/share/java/mysql-connector-java.jar"
     javac dbexample.java
     java dbexample
     ```

If you add the **CLASSPATH** variable to your startup scripts—.bash_profile, .bashrc, or .cshrc —then it will be present for all your Java compilations.

```java
import java.sql.*;

public class dbexample {
    public static final String SERVER   = "jdbc:mysql://sunapee.cs.dartmouth.edu/";
    public static final String USERNAME = "user";
    public static final String PASSWORD = "pass";
    public static final String DATABASE = "db";
    public static final String QUERY    = "SELECT * FROM instructor;";

    public static void main(String[] args) {
    Connection con = null;
    Statement stmt = null;
    ResultSet res  = null;
    int numColumns = 0;

    // attempt to connect to db
    try {
        // load mysql driver
        Class.forName("com.mysql.jdbc.Driver").newInstance();

        // initialize connection
        con = DriverManager.getConnection(SERVER+DATABASE, USERNAME, PASSWORD);

        System.out.println("Connection established.");

        // initialize a query statement
        stmt = con.createStatement();

        // query db and save results
        res = stmt.executeQuery(QUERY);

        System.out.format("Query executed: '%s'\n\nResults:\n", QUERY);

        // the result set contains metadata
        numColumns = res.getMetaData().getColumnCount();

        // print table header
        for(int i = 1; i <= numColumns; i++) {
        System.out.format("%-12s", res.getMetaData().getColumnName(i));
        }
        System.out.println("\n------------------------------------------------");

        // iterate through results
        while(res.next()) {
        for(int i = 1; i <= numColumns; i++) {
            System.out.format("%-12s", res.getObject(i));
        }
        System.out.println("");
        }
    } catch (SQLException e ) {          // catch SQL errors
        System.err.format("SQL Error: %s", e.getMessage());
    } catch (Exception e) {              // anything else
        e.printStackTrace();
    } finally {
        // cleanup
        try {
        con.close();
        stmt.close();
        res.close();
        System.out.print("\nConnection terminated.");
        } catch (Exception e) { /* ignore cleanup errors */ }
    }
    }
}
```

Listing 1: Java connecting to MySQL database.

In any Java class in which you want to access a MySQL database, you will need to import the Java SQL package and load the MySQL driver. This is shown in the Listing 1.

Java uses a Statement for queries and a ResultSet for the query results. The ResultSet can be seen as an iterator through the rows of the result.

C

The header **mysql.h** contains the necessary definitions for connecting to MySQL databases, as seen in Listing 2. When compiling C applications, you will need certain compiler flags and libraries. These can be obtained using the following command line:

```
gcc -o executable `mysql_config --cflags` source.c `mysql_config --libs`
```

```c
#include <stdio.h>                          // for fprintf() and printf()
#include <stdlib.h>                         // for exit()
#include <mysql.h>                          // for mysql functionality

// db connection defines
#define SERVER   "sunapee.cs.dartmouth.edu"  // db server to connect to
#define USERNAME "user"                      // user to connect as
#define PASSWORD "pass"                      // user's password
#define DATABASE "db"                        // db to use
#define QUERY    "SELECT * FROM instructor;" // query statement

int main() {
  // local variables
  MYSQL *con = NULL;                         // db connection pointer
  MYSQL_RES *res = NULL;                     // result set
  MYSQL_ROW row;                             // single row of result
  int numColumns = 0;                        // number of columns

  // attempt to initialize db connection
  if(!(con = mysql_init(NULL))) {
    fprintf(stderr, "ERROR: %s\n", mysql_error(con));
    exit(1);
  }

  // attempt to connect to db
  if(!mysql_real_connect(con, SERVER, USERNAME, PASSWORD, DATABASE, 0, NULL, 0)) {
    fprintf(stderr, "Error: Cannot connect to database server.");
    exit(1);
  }

  printf("Connection established.\n");

  // attempt to query db
  if(mysql_query(con, QUERY)) {
    fprintf(stderr, "Error: Cannot query database.");
    exit(1);
  }

  printf("Query executed: '%s'\n\nResults:\n", QUERY);

  // store query result
  if(!(res = mysql_store_result(con))) {
    fprintf(stderr, "Error: Cannot store query result.");
    exit(1);
  }

  // store number of columns in result set
  numColumns = mysql_num_fields(res);

  // print table header
  int i;
  for(i = 0; i < numColumns; i++) {
    printf("%-12s", mysql_fetch_field(res)->name);
  }
  printf("\n------------------------------------------------\n");

  // iterate through results
  while((row = mysql_fetch_row(res))) {
    for(i = 0; i < numColumns; i++) {
      printf("%-12s", row[i] ? row[i] : "NULL");
    }
    printf("\n");
  }

  // cleanup
  mysql_free_result(res);
  mysql_close(con);

  printf("Connection terminated.");

  return 0;
}
```

Listing 2: C connecting to MySQL database.

Python

Python can be used to access a MySQL database by simply importing the MySQL Python Connector, as shown in Listing 3. You may need to install the mysql package for python using "`pip install mysql`" first. A cursor might be a new concept to you, but it can be seen as an extension of the iterator concept. You will utilize a cursor to execute queries and access result sets.

```
1  from __future__ import print_function        # make print a function
2  import mysql.connector                       # mysql functionality
3  import sys                                   # for misc errors
4
5  SERVER   = "sunapee.cs.dartmouth.edu"        # db server to connect to
6  USERNAME = "username"                           # user to connect as
7  PASSWORD = "password"                           # user's password
8  DATABASE = "db"                              # db to user
9  QUERY    = "SELECT * FROM instructor;"       # query statement
10
11 if __name__ == "__main__":
12    try:
13       # initialize db connection
14       con = mysql.connector.connect(host=SERVER,user=USERNAME,password=PASSWORD,
15                                     database=DATABASE)
16
17       print("Connection established.")
18
19       # initialize a cursor
20       cursor = con.cursor()
21
22       # query db
23       cursor.execute(QUERY)
24
25       print("Query executed: '{0}'\n\nResults:".format(QUERY))
26
27       # print table header
28       print("".join(["{:<12}".format(col) for col in cursor.column_names]))
29       print("------------------------------------------------")
30
31       # iterate through results
32       for row in cursor:
33          print("".join(["{:<12}".format(col) for col in row]))
34
35    except mysql.connector.Error as e:        # catch SQL errors
36       print("SQL Error: {0}".format(e.msg))
37    except:                                   # anything else
38       print("Unexpected error: {0}".format(sys.exc_info()[0]))
39
40    # cleanup
41    con.close()
42    cursor.close()
43
44    print("\nConnection terminated.", end='')
```

Listing 3: Python connecting to MySQL database.

Resources

- `http://dev.mysql.com/doc/refman/5.5/en/index.html`
- `http://dev.mysql.com/doc/refman/5.5/en/connector-j.html`
- `http://dev.mysql.com/doc/refman/5.5/en/connector-python.html`
- `http://dev.mysql.com/doc/refman/5.5/en/connector-c.html`
- `http://dev.mysql.com/doc/workbench/en/wb-intro.html`