# MYSQL in c++

**Nathan Warner**


Northern Illinois
University

Computer Science
Northern Illinois University
United States

## Contents

## Setup and compilation

We begin by including the required library

```
#include <mysql/mysql.h>
```

We can then compile with

```
g++ -o program -I/usr/include/mariadb -lmariadb program.cc
```

## Initialization and deinitialization

We start with the initialization

```
mysql_library_init(0, NULL, NULL);
MYSQL* connection = mysql_init(NULL);

const char* host, *user, *password, *db;
host = "...";
user = "...";
password = "...";
db = "...";

connection = mysql_real_connect(connection, host, user,
↪  password, db, 0, NULL, 0);


// At the end of the program
mysql_close(connection);
mysql_library_end();
```

## Running a query, getting number of affected rows, and getting the number of columns

```
const char* query = "SELECT * FROM S";
int err = mysql_query(connection, query);

// Returns non-zero on failure, zero on success
if (err) {
    const char* e = mysql_error(connection);
    unsigned int en = mysql_errno(connection);
    cout << "Error numeric code: " << en << "\t Message: \t" <<
↪   e << endl;
    exit(1);
}
my_ulonglong affected = mysql_affected_rows(connection);
unsigned int fc = mysql_field_count(connection);

cout << "Affected: " << affected << endl;
cout << "Field count: " << fc << endl << endl;

```

## Errors

Many of the functions return error codes, we also have two functions regarding errors.

```
unsigned int mysql_errno(MYSQL *mysql);
const char *mysql_error(MYSQL *mysql);
```

## Fetching and processing the result set

```
MYSQL_RES* res = mysql_store_result(connection);
ull returned = mysql_num_rows(res);

cout << "Returned: " << returned << endl;

MYSQL_ROW row;
unsigned int num_fields = mysql_num_fields(res);
while ( (row = mysql_fetch_row(res)) ) {
    for (unsigned int i=0; i< num_fields; ++i) {
        cout << (row[i] ? row[i] : "NULL") << '\t';
    }
    cout << endl;
}
mysql_free_result(res);
```