

# 实验五 - 嵌入式 SQL

张海 3130000923

## 1. 实验目的

1. 熟悉通过嵌入式 SQL ( 主语言为 C/C++ ) 访问数据库。

## 2. 实验要求

1. 熟悉数据库的预编译器。
2. 编写一个嵌入式 SQL 程序实现对数据库中数据的插入 / 修改 / 查询 ( 多记录 )。
3. 完成实验报告。

## 3. 实验平台

1. 操作系统平台：Arch Linux
2. 数据库管理系统：MySQL
3. 预编译连接器：MySQL C++ Connector

## 4. 实验过程

1. 使用 pacman 安装 mysql , mysql-connector-c++ , boost 软件包。
2. 找到 MySQL C++ Connector 的官方入门文档，下载为 pdf 文件，其中有相当详尽的解释。
3. 下载 mysql-connector-c++ 包的源文件，找到 examples 目录

并对其中的示例代码进行学习，重点在于其对象使用与函数调用，以及正确的错误处理方法。

#### 4. 运行结果：

```
Terminal - jonathan@Pitaya-VM:~/Programming/C++/MySQL
[jonathan@Pitaya-VM MySQL]$ ./test
# Test table created
# Test table populated
# Running 'SELECT id, label FROM test ORDER BY id ASC'
# Number of rows res->rowCount() = 4
# Fetching row 0 id = 0, label = 'a'
# Fetching row 1 id = 1, label = 'b'
# Fetching row 2 id = 2, label = 'c'
# Fetching row 3 id = 3, label = 'd'
# Fetching 'SELECT id FROM test ORDER BY id DESC' using type conversion
# Fetching row 0# id (int) = 3# id (boolean) = 1# id (long) = 3
# Fetching row 1# id (int) = 2# id (boolean) = 1# id (long) = 2
# Fetching row 2# id (int) = 1# id (boolean) = 1# id (long) = 1
# Fetching row 3# id (int) = 0# id (boolean) = 0# id (long) = 0
# UPDATE indicates 1 affected rows
# Expecting id = 100, label = 'y' and got id = 100, label = 'y'
# Table 'test' dropped!
# done!
[jonathan@Pitaya-VM MySQL]$
```

## 实验心得

1. 环境比较难配置，其中出过大量的问题。包括无法正确地安装软件包，未知的编译选项，运行时错误等。但是环境配置完成后使用上还是十分灵活的，可以通过对 API 的使用，以及对相应函数的调用来做到安全地进行数据库调用。相应地，如果通过 execute 函数来进行操作，虽然会方便不少，但是可能会导致出现注入漏洞。