

实验二：在应用程序中使用 SQL 编程

李建中 刘显敏

1. 实验目的

在熟练掌握 SQL 语言的基本命令的基础上，学会使用高级编程语言操作数据库，进行应用开发。

2. 实验环境

Windows 操作系统、SQLite（或 PostgreSQL）、Java 或者 C 语言、SQLite 的 JDBC 库或者 C 语言接口库

3. 实验任务（以 SQLite 为例）

3.1 建立基本的编程环境

利用实验中给定的开发支持文件：JDBC 对应 sqlite-jdbc-3.7.2.jar 包文件、C 对应的 sqlite3.def, sqlite3.dll, sqlite3.h, libsqlite3.a 文件，调试基本的编程环境。

（1）JDBC

假设已经有 DBforCourse.db3 文件（SQLite 的数据库文件），其中包含课程中讲授的 student、apply、college 表，那么可以编写如下程序

```
import java.sql.*;

public class test{
    public static void main(String[] args) throws Exception
    {
        System.out.println("hello!");
        Class.forName("org.sqlite.JDBC");
        Connection c=DriverManager.getConnection
            ("jdbc:sqlite:C:\\Users\\xianmin\\Desktop\\test\\DBforCourse.db3");
        Statement s=c.createStatement();
        ResultSet r=s.executeQuery("select * from college");
        while(r.next())
        {
            System.out.println(r.getString("cName"));
        }
    }
}
```

编译方法（可选）：

将 sqlite-jdbc-3.7.2.jar 放到程序所在目录下

执行 javac test.java 编译

执行 java -classpath ".;sqlite-jdbc-3.7.2.jar" test 运行

注意：如果是 win7 或者 win8 等系统盘有写入权限保护的情况，需要用管理员权限运行控制台窗口

(2) C 接口

安装 C 编译器，以 cygwin g++为例

建立程序 test.c

```
#include <stdio.h>
```

```
#include "sqlite3.h"
```

```
static int callback(void *data, int argc, char **argv, char **azColName){
    int i;
    fprintf(stderr, "%s: ", (const char*)data);
    for(i=0; i<argc; i++){
        printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");
    }
    printf("\n");
    return 0;
}
```

```
int main(int argc, char* argv[])
{
    sqlite3 *db;
    char *zErrMsg = 0;
    int rc;
    char *sql;
    const char* data = "Callback function called";

    rc = sqlite3_open("DBforCourse.db3", &db);

    if( rc ){
        fprintf(stderr, "Can't open database: %s\n", sqlite3_errmsg(db));
        exit(0);
    }else{
        fprintf(stderr, "Opened database successfully\n");
    }

    /* Create SQL statement */
    sql = "SELECT * from college";

    /* Execute SQL statement */
    rc = sqlite3_exec(db, sql, callback, (void*)data, &zErrMsg);
    if( rc != SQLITE_OK ){
        fprintf(stderr, "SQL error: %s\n", zErrMsg);
    }
}
```

```

        sqlite3_free(zErrMsg);
    }else{
        fprintf(stdout, "Operation done successfully\n");
    }

    sqlite3_close(db);
}

```

将 sqlite3.h、sqlite3.def、sqlite3.dll 放入同一目录下

运行命令 GNU 库的 dlltool 命令生成 lib 文件，
 dlltool --def sqlite3.def --dllname sqlite3.dll --output-lib libsqlite3.a
 或者跳过此步骤直接采用提供的 libsqlite3.a 文件

编译：gcc -o mytest test.c -L. -lsqlite3 生成 mytest.exe

3.2 利用高级编程语言控制数据库

- 1) 如果没有数据库文件 “*.db3”，可以利用高级编程接口中的打开数据库功能新建一个数据库，sqlite 处理不存在的数据库的方法是新建一个名字相同的数据库，因此可以利用这个特性新建数据库，名字任意
- 2) 创建关系表（用高级编程语言调用 create table）

表一
 表名：College（存储大学的信息）
 属性：cName（字符串存储的大学名字），state（字符串格式的大学所在州），enrollment（整数形式的大学入学学费）

表二
 表名：Student（存储学生的信息）
 属性：sID（整数形式的学号），sName（字符串形式的学生名字），GPA（小数形式的成绩），sizeHS（整数形式的所在高中规模）

表三
 表名：Apply（存储学生申请学校的信息）
 属性：sID（整数形式的学号），cName（字符串形式的大学名字），major（字符串形式的专业名字），decision（字符串形式的申请结果）
- 3) 将 dbcollege.txt、dbstudent.txt、dbapply.txt 中的内容导入数据库（利用文件操作读取三个文件，调用 insert 语句逐条插入）
- 4) 调用命令查看导入后的 College、Student、Apply 表中的所有内容（使用 select）
- 5) 找出所有分数大于 3.6 的学生的 ID、名字和分数，并输出到控制台
- 6) 查询所有申请信息中姓名、学校的对，去除重复元组，输出

- 7) 查询所有所处高中规模小于 1000、申请了斯坦福的计算机系的学生、其分数、及申请结果，输出
- 8) 查询学生申请学校的所有信息（输出结果按照学生分数降序排列，对于相同分数的学生，按照入学费用升序排列），输出
- 9) 查询学生的所有信息，并为每一个学生计算一个新的分数 newGPA（按照 GPA 同 sizeHS 的关系换算出来的新分数），输出
$$\text{newGPA} = \text{GPA} \times (\text{sizeHS} \div 1000)$$
- 10) 查询最贵的大学的名字，输出
- 11) 将三个表自然连接，并输出所有属性
- 12) 求查找申请数少于 5 的大学，输出

4. 参考资料

参加实验所给的支持文档