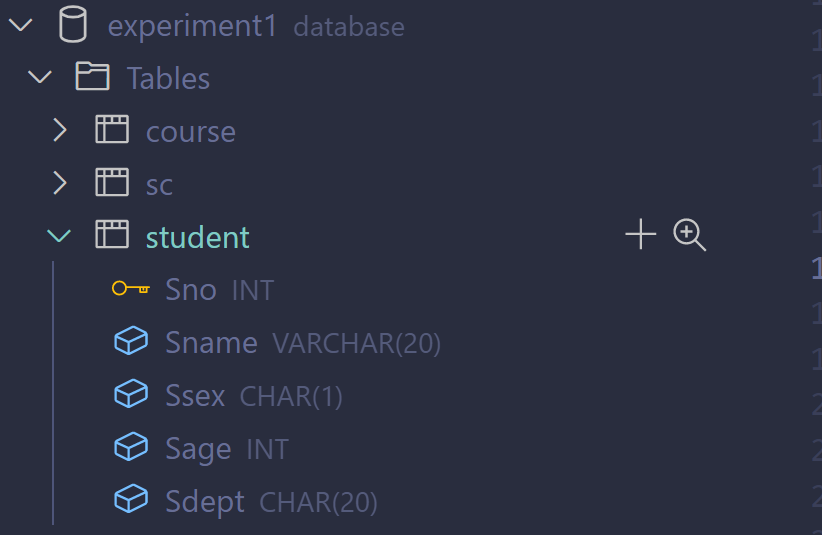
# 实验三

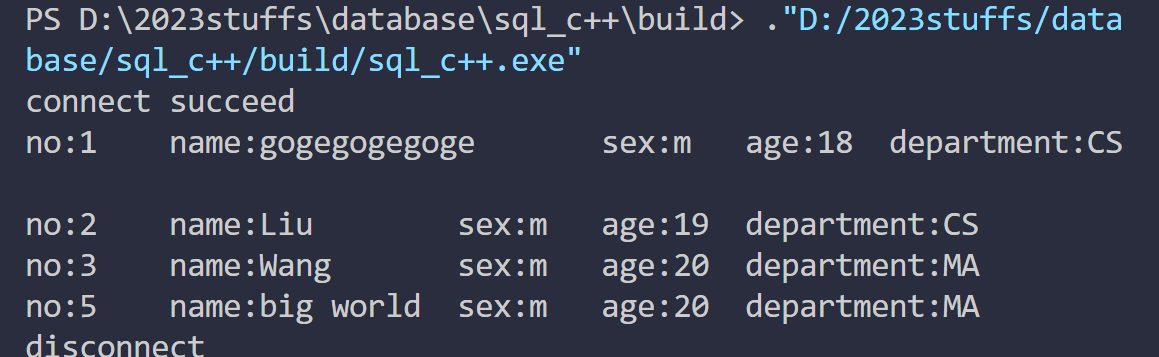
## 数据库

数据库和表如下所示



## 连接数据库及查询

#### 运行结果如下图所示



#### 代码如下所示

###### 连接代码

StudentDAO::StudentDAO() : host("root"), password("520711YY"), database("experiment1")

{

    mysql\_init(&mysql);

*if* (!mysql\_real\_connect(&mysql, "localhost", host, password, database, 3306, nullptr, 0))

    {

        std::cout << "connect failed" << mysql\_errno(&mysql) << std::endl;

    }

*else*

        std::cout << "connect succeed" << std::endl;

}

StudentDAO::~StudentDAO()

{

    mysql\_close(&mysql);

    printf("disconnect\n");

}

###### 查询代码：

bool StudentDAO::loadStudents(std::vector<Student> &*students*)

{

*return* queryStudents(*students*, "select \* from student");

}

bool StudentDAO::queryStudents(std::vector<Student> &*students*, const char \**query*)

{

*if* (exec(*query*))

    {

        res = mysql\_store\_result(&mysql);

*if* (res == nullptr)

        {

            std::cout << "store failed: " << mysql\_errno(&mysql) << std::endl;

*return* false;

        }

*else*

*while* (row = mysql\_fetch\_row(res))

*students*.emplace\_back(Student(row));

    }

    mysql\_free\_result(res);

*return* true;

}

###### 在主函数中：

    StudentDAO dao;

    std::vector<Student> students;

    dao.loadStudents(students);

*for* (auto s : students)

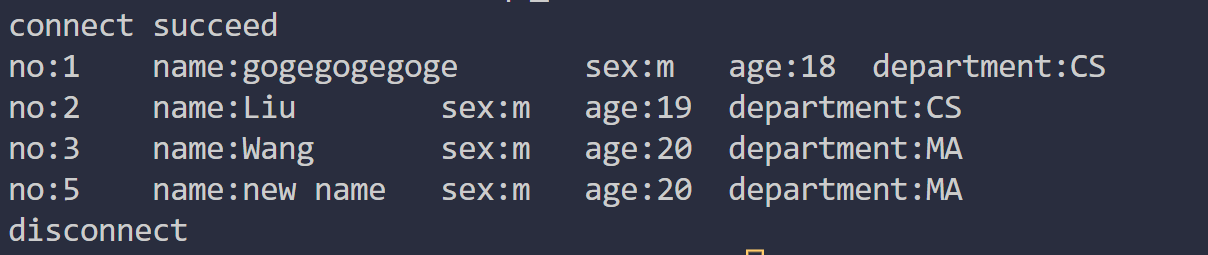
    {

        std::cout << s;

    }

## 更新数据

#### 运行结果如下图所示



#### 代码如下所示

###### 更新代码

bool StudentDAO::updateStudent(const Student &*val*)

{

    std::stringstream builder;

    builder << "update student set Sname = '" << *val*.name << "', Ssex = '" << *val*.sex << "', Sage = " << *val*.age

            << ", Sdept = '" << *val*.dept << "' where Sno = " << *val*.no;

    const string &str = builder.str();

    const char \*query = str.c\_str();

*return* exec(query);

}

bool StudentDAO::exec(const char \**query*)

{

*if* (mysql\_real\_query(&(this->mysql), *query*, (unsigned int)strlen(*query*)))

    {

        std::cout << "exec failed: " << mysql\_errno(&mysql) << '\n'

                  << "sql: " << *query* << std::endl;

*return* false;

    }

*return* true;

}

###### 主函数

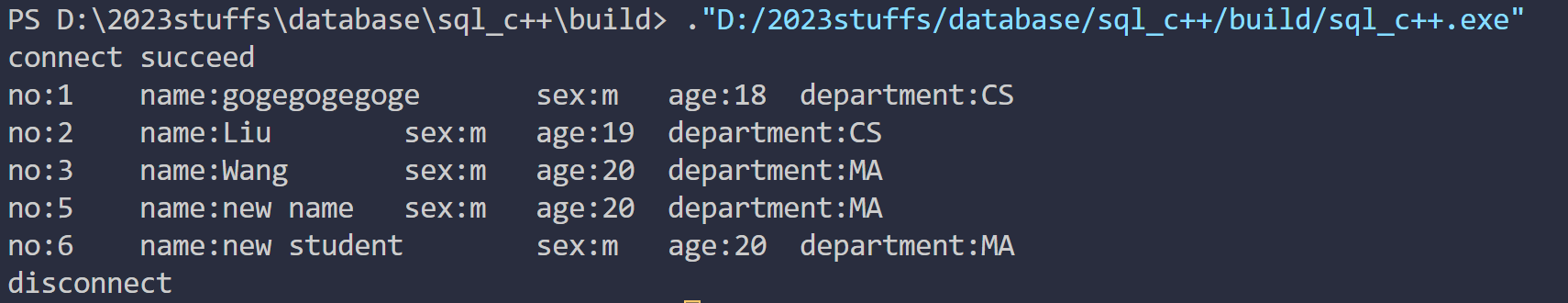
*// update*

    temp.name = "new name";

    dao.updateStudent(temp);

## 插入数据

#### 运行结果如下图所示



#### 代码如下所示

###### 插入代码

bool StudentDAO::updateStudent(const Student &*val*)

{

    std::stringstream builder;

    builder << "update student set Sname = '" << *val*.name << "', Ssex = '" << *val*.sex << "', Sage = " << *val*.age

            << ", Sdept = '" << *val*.dept << "' where Sno = " << *val*.no;

    const string &str = builder.str();

    const char \*query = str.c\_str();

*return* exec(query);

}

###### 主函数

    StudentDAO dao;

    std::vector<Student> students;

    dao.loadStudents(students);

*for* (auto s : students)

    {

        std::cout << s;

    }

    auto &temp = students.back();

*// insert*

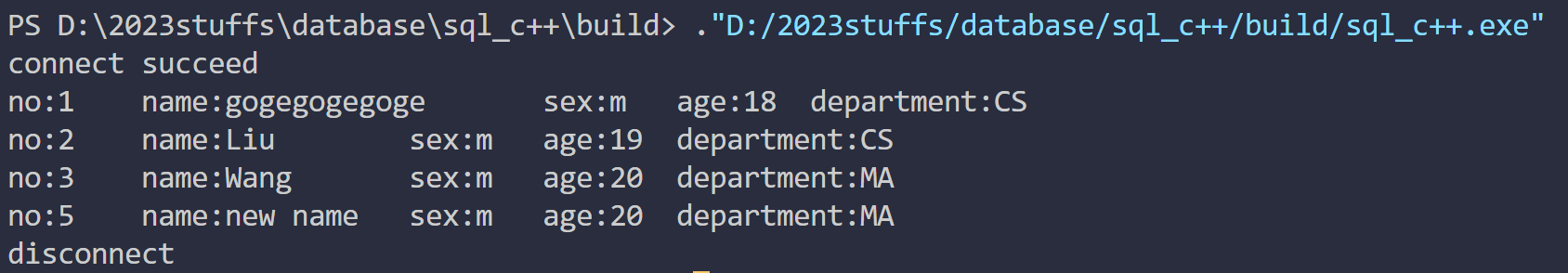
    auto newStudent = temp;

    newStudent.name = "new student";

    dao.insertStudent(newStudent);

## 删除数据

#### 运行结果如下图所示



#### 代码如下所示

###### 删除代码

bool StudentDAO::deleteStudent(const Student &*student*)

{

    char query[] = "delete from student where Sno = ";

    strcat(query, std::to\_string(*student*.no).c\_str());

*return* exec(query);

}

###### 主函数

    StudentDAO dao;

    std::vector<Student> students;

    dao.loadStudents(students);

*for* (auto s : students)

    {

        std::cout << s;

    }

    auto &temp = students.back();

*// delete*

    dao.deleteStudent(temp);