

实验课程名称： 数据库应用系统开发 A

实验项目名称	学生信息管理软件开发（C 语言）			实验成绩	
实 验 者	胡 珊	专业班级	信息 2001	组 别	无
同 组 者	无			实验日期	2022 年 10 月 31 日

**第一部分：实验预习报告**（包括实验目的、意义，实验基本原理与方法，主要仪器设备及耗材，实验方案与技术路线等）

### 一、实验目的

1. 学会配置 ODBC 数据源，熟悉使用 C 语言+ODBC 来进行数据库应用程序的开发
2. 掌握使用 C 语言通过 ODBC 连接数据库的方法
3. 掌握使用 C 语言通过 ODBC 对数据库表进行增删改查操作的方法

### 二、实验平台

DBMS 采用 SQL Server 2019，C 语言及开发平台 CodeBlocks。

### 三、实验要求和内容

使用 C 语言和 ODBC 编写程序，实现对学生基本表数据的增加、删除、查询、修改功能。

## 第二部分：实验过程记录（可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

### 一、实验步骤

#### 1.建立学生数据库、学生信息表

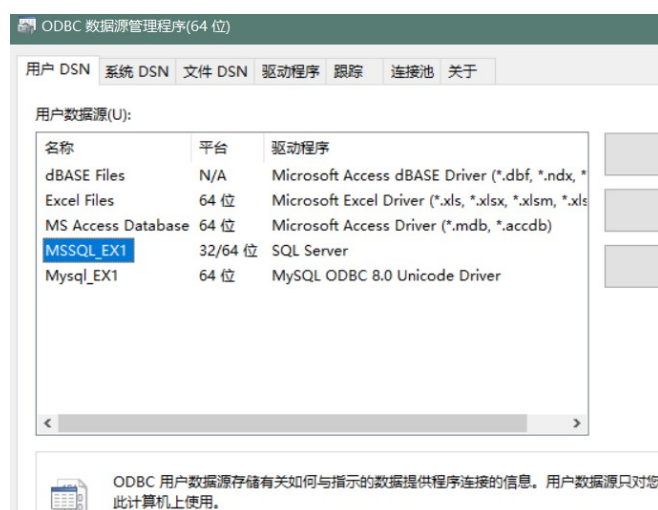
```
CREATE DATABASE Students;

CREATE TABLE Student(
    sno int NOT NULL,
    sname varchar(50) COLLATE Chinese_PRC_CI_AS NULL,
    ssex nchar(10) COLLATE Chinese_PRC_CI_AS DEFAULT '男'
)

INSERT INTO Student (sno, sname, ssex) VALUES ('1', '文')
INSERT INTO Student (sno, sname, ssex) VALUES ('2', '罗')
INSERT INTO Student (sno, sname, ssex) VALUES ('3', '卫')
INSERT INTO Student (sno, sname, ssex) VALUES ('4', '郭')
INSERT INTO Student (sno, sname, ssex) VALUES ('5', '闻')
INSERT INTO Student (sno, sname, ssex) VALUES ('6', '李')
INSERT INTO Student (sno, sname, ssex) VALUES ('7', '张')
```

#### 2.ODBC 数据源配置

##### 1) 打开控制面板管理工具中的 ODBC 数据源



## 2) 配置数据源名称、描述和服务名称

Microsoft ODBC SQL Server DSN 配置

此向导将帮助您建立一个能用于连接 SQL Server 的 ODBC

您想用什么名称来命名数据源?

名称(M): MSSQL\_EX1

您希望如何描述此数据源?

描述(D): DataBase\_Ex

您想连接哪一个 SQL Server?

服务器(S): (local)

## 3) 设置用户登录账号和密码、默认数据库等相关信息

Microsoft ODBC SQL Server DSN 配置

SQL Server 应该如何验证登录 ID 的真伪?

☐ 使用网络登录 ID 的 Windows NT 验证(W)。

☒ 使用用户输入登录 ID 和密码的 SQL Server 验证(C)。

要更改用于与 SQL Server 通讯的网络库, 请单击“客户端配置”

☒ 连接 SQL Server 以获得其它配置选项的默认设置(C)。

登录 ID(L): sa

密码(P): ●●●●●●

Microsoft ODBC SQL Server DSN 配置

☒ 更改默认的数据库为(D): students

☐ 附加数据库文件名(H):

☒ 使用 ANSI 引用的标识符(U)。

☒ 使用 ANSI 的空值、填充及警告(A)。

☐ 若主 SQL Server 不可用, 请使用故障转移 SQL Server

## 4) 测试连接, 连接成功

ODBC Microsoft SQL Server 安装

将按下列配置创建新的 ODBC 数据源:

Microsoft SQL Server ODBC 驱动程序版本 10.00.19041

数据源名称: MSSQL\_EX1

数据源描述: DataBase\_Ex

服务器: (local)

数据库: students

语言: (Default)

翻译字符数据: Yes

日志长运行查询: No

日志驱动程序统计: No

使用区域设置: No

预定义的语句选项: 在断开时删除临时存储过程

使用故障转移服务器: No

使用 ANSI 引用的标识符: Yes

使用 ANSI 的空值、填充和警告: Yes

数据加密: No

SQL Server ODBC 数据源测试

测试结果

Microsoft SQL Server ODBC 驱动程序

正在运行连接测试...

试图连接

连接被建立

验证选项设置

断开与服务器的连接

测试成功!

### 3.编程实现

#### 1) 连接数据库

```
void LinkDatabase(){
    ret=SQLAllocHandle(SQL_HANDLE_ENV,NULL,&henv); //申请环境句柄
    ret=SQLSetEnvAttr(henv,SQL_ATTR_ODBC_VERSION,(SQLPOINTER)SQL_OV
_ODBC3,SQL_IS_INTEGER); //设置环境属性
    ret=SQLAllocHandle(SQL_HANDLE_DBC,henv,&hdbc); //申请数据库连接句柄
    ret=SQLConnect(hdbc,(SQLCHAR*)"MSSQL_EX1",SQL_NTS,(SQLCHAR*)"sa",SQL_NTS,(SQLCHAR*)"123456",SQL_NTS); //连接数据库
}
```

#### 2) 添加学生信息

```
void Insert(){
    if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
        ret=SQLAllocHandle(SQL_HANDLE_STMT,hdbc,&hstmt);
        SQLCHAR sql[] = "INSERT INTO Student VALUES(?,?,?)";
        SQLINTEGER P = SQL_NTS;
        printf("请输入学号:\n");
        scanf("%s",sno);
        printf("请输入姓名:\n");
        scanf("%s",sname);
        printf("请输入性别:\n");
        scanf("%s",ssex);
        ret = SQLPrepare(hstmt,sql,SQL_NTS);
        ret=SQLBindParameter(hstmt,1,SQL_PARAM_INPUT,SQL_C_CHAR,SQL
_VARCHAR,20,0,sno,20,&P);
        ret=SQLBindParameter(hstmt,2,SQL_PARAM_INPUT,SQL_C_CHAR,SQL
_VARCHAR,20,0,sname,20,&P);
        ret=SQLBindParameter(hstmt,3,SQL_PARAM_INPUT,SQL_C_CHAR,SQL
_VARCHAR,10,0,ssex,20,&P);
        ret = SQLExecute(hstmt);
        if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
            printf("添加成功! \n");
        }
        else{
            printf("添加失败! \n");
        }
        SQLDisconnect(hdbc);
    }
    else{
        printf("数据库连接失败! \n");
    }
    SQLFreeHandle(SQL_HANDLE_DBC,hdbc); //释放连接句柄
}
```

```

    SQLFreeHandle(SQL_HANDLE_ENV, henv);
}

```

3) 删除学生信息

```

void Delete(){
    if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
        ret=SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt);
        SQLCHAR sql[]="DELETE FROM Student WHERE sno = ? ";
        SQLINTEGER P = SQL_NTS;
        printf("请输入你要删除学生的学号! \n");
        scanf("%s", myID);
        ret=SQLPrepare(hstmt, sql, SQL_NTS);
        ret=SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR, SQL_
_VARCHAR, 50, 0, myID, 50, &P);
        ret=SQLExecute(hstmt);
        if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
            printf("删除成功! \n");
        }
        else{
            printf("未找到该学生! \n");
        }
        SQLDisconnect(hdbc);
    }
    else{
        printf("数据库连接失败! \n");
    }
    SQLFreeHandle(SQL_HANDLE_DBC, hdbc); //释放连接句柄
    SQLFreeHandle(SQL_HANDLE_ENV, henv); //释放环境句柄
}

```

4) 修改学生信息

```

//修改
void Update(){
    int mark=0;
    printf("请输入你要修改的学生学号! \n");
    scanf("%s", myID);
    ret=SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt); //申请 SQL 语句句柄
    SQLCHAR sql[]="SELECT * FROM Student WHERE sno = ? ";
    SQLINTEGER p = SQL_NTS;
    ret=SQLPrepare(hstmt, sql, SQL_NTS); //准备 SQL 语句
    ret=SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR, SQL_VAR
CHAR, 50, 0, myID, 50, &p); //绑定参数
}

```

```

ret=SQLExecute(hstmt);
if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
    SQLBindCol(hstmt,1,SQL_C_CHAR,sno,20,&sNo);
    SQLBindCol(hstmt,2,SQL_C_CHAR,sname,20,&sName);
    SQLBindCol(hstmt,3,SQL_C_CHAR,ssex,10,&sSex);
    ret=SQLFetch(hstmt); //移动光标
    if(ret == SQL_NO_DATA){
        printf("未找到该学生! \n");
    }
    while(ret != SQL_NO_DATA){
        printf("%s %s %s\n",sno,sname,ssex);
        mark = 1;
        ret=SQLFetch(hstmt);
    }
}
else{
    printf("查询失败! \n");
}
LinkDatabase();
if(mark == 1){
    if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
        ret=SQLAllocHandle(SQL_HANDLE_STMT,hdbc,&hstmt);
        SQLCHAR sql[]="UPDATE Student SET sname=? WHERE sno=?";
        SQLINTEGER P = SQL_NTS;
        printf("请输入新的学生姓名! \n");
        scanf("%s",sname);
        ret=SQLPrepare(hstmt,sql,SQL_NTS); //准备 SQL 语句
        ret=SQLBindParameter(hstmt,1,SQL_PARAM_INPUT,SQL_C_CHAR
,SQL_VARCHAR,20,0,sname,20,&P);
        ret=SQLBindParameter(hstmt,2,SQL_PARAM_INPUT,SQL_C_CHAR
,SQL_VARCHAR,50,0,myID,50,&P); //绑定参数
        ret=SQLExecute(hstmt); //执行 SQL 语句
        if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
            printf("修改成功! \n");
        }
        else{
            printf("修改失败! \n");}
        SQLDisconnect(hdbc); }
    else{
        printf("数据库连接失败! \n");}
    SQLFreeHandle(SQL_HANDLE_DBC,hdbc); //释放连接句柄
    SQLFreeHandle(SQL_HANDLE_ENV,henv); //释放环境句柄
}
}

```

5) 查询学生信息

```
void Select(){
    if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
        ret=SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt);
        SQLCHAR sql[]="SELECT * FROM Student WHERE sno = ? ";
        SQLINTEGER p = SQL_NTS;
        printf("请输入你要查找的学生学号:\n");
        scanf("%s", myID);
        ret=SQLPrepare(hstmt, sql, SQL_NTS); //准备 SQL 语句
        ret=SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR, SQL
_VARCHAR, 50, 0, myID, 50, &p); //绑定参数
        ret=SQLExecute(hstmt);
        if(ret==SQL_SUCCESS || ret==SQL_SUCCESS_WITH_INFO){
            SQLBindCol(hstmt, 1, SQL_C_CHAR, sno, 20, &sNo);
            SQLBindCol(hstmt, 2, SQL_C_CHAR, sname, 20, &sName);
            SQLBindCol(hstmt, 3, SQL_C_CHAR, ssex, 10, &sSex);
            ret=SQLFetch(hstmt); //移动光标
            if(ret == SQL_NO_DATA){
                printf("未找到该学生! \n");
            }
            while(ret != SQL_NO_DATA){
                printf("%s %s %s\n", sno, sname, ssex);
                ret=SQLFetch(hstmt);
            }
        }
        else{
            printf("查询失败! \n");
        }
        SQLDisconnect(hdbc); //断开与数据库的连接
    }
    else{
        printf("数据库连接失败! \n");
    }
    SQLFreeHandle(SQL_HANDLE_DBC, hdbc); //释放连接句柄
    SQLFreeHandle(SQL_HANDLE_ENV, henv); //释放环境句柄
}
```

教师签字\_\_\_\_\_

### 第三部分 结果与讨论（可加页）

#### 一、实验结果分析（包括数据处理、实验现象分析、影响因素讨论、综合分析和结论等）

##### 1. 添加学生信息

添加前：

```
请输入对应数字进行下一步操
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
5
1 文章 男
2 罗杰 男
3 卫佳 男
4 郭昱 男
5 闻锋 男
6 李四 男
7 张洁 女
```

添加学生信息（12，王伟，男）

```
请输入对应数字进行下一步操作：
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
1
请输入学号：
12
请输入姓名：
王伟
```

添加后：

```
请输入对应数字进行下一步操作：
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
5
1 文章 男
2 罗杰 男
3 卫佳 男
4 郭昱 男
5 闻锋 男
6 李四 男
7 张洁 女
8 夏林 男
```



## 2. 删除学生信息

删除刚才添加的学生信息（12，王伟，男）

```
请输入对应数字进行下一步操作
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
4
请输入你要删除学生的学号！
```

删除后查询：

```
请输入对应数字进行下一步操作
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
5
1 文章 男
2 罗杰 男
3 卫佳 男
4 郭昱 男
5 闻锋 男
6 李四 男
7 张洁 女
8 王林 男
```

## 3. 修改学生信息

将学号为 10 的张三同学的姓名修改为张萌：

```
请输入对应数字进行下一步操作
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
3
请输入你要修改的学生学号！
10
10 张三 男
请输入新的学生姓名！
```

修改后:

```
请输入对应数字进行下一步
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
1 文章 男
2 罗杰 男
3 卫佳 男
4 郭昱 男
5 闻锋 男
6 李四 男
7 张洁 女
8 贾林 男
```

4. 查看全部学生信息

```
请输入对应数字进行下一步操
1. 添加学生信息
2. 查找学生信息
3. 修改学生信息
4. 删除学生信息
5. 显示学生信息
6. 退出系统
5
1 文章 男
2 罗杰 男
3 卫佳 男
4 郭昱 男
5 闻锋 男
6 李四 男
7 张洁 女
8 贾林 男
```

## 二、小结、建议及体会

1. 了解数据库编程的思路和方法，学会了使用 C 语言通过 ODBC 连接数据库；
2. 实验过程中遇到许多问题，比如数据库连接失败、插入数据不成功等，通过不断查找资料和修改代码都得到了解决；
3. 复习了 C 语言的基本语法，以及 SQL 语言基础语法。