

Assignment1 Report

71118415 叶宏庭

一、 作业要求

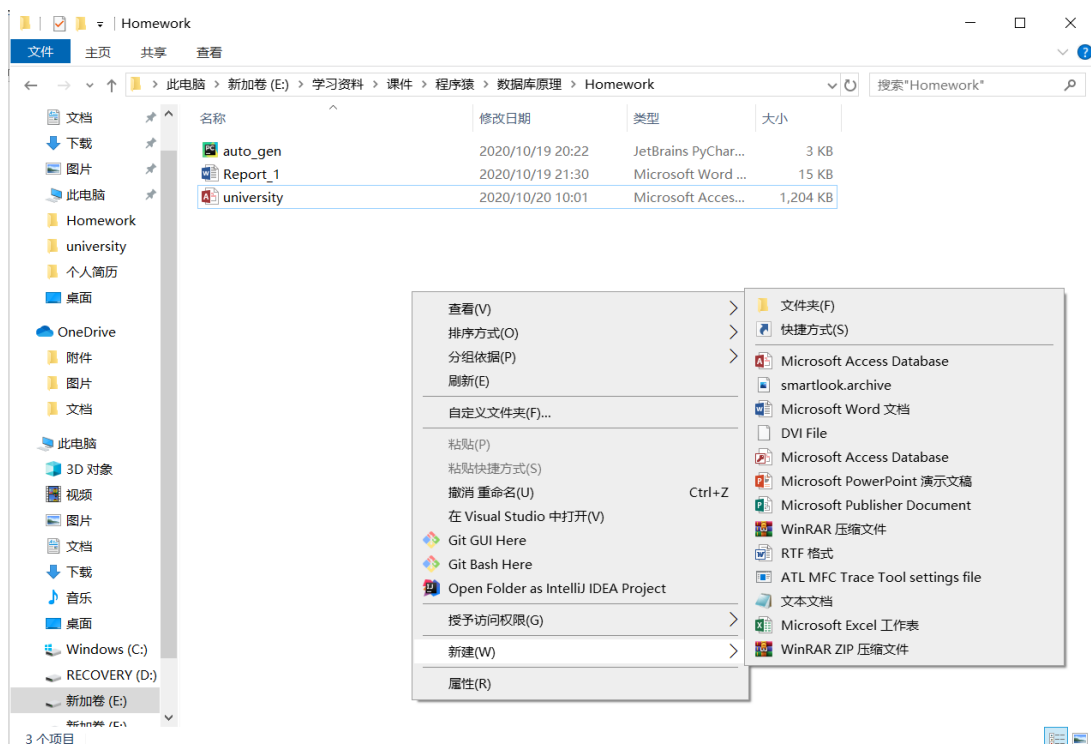
- ✧ 熟悉 MS Access 工具
- ✧ 建立一个 university.accdb 数据库
- ✧ 将文本文档的数据导入到 university.accdb 中去
- ✧ 将 5000 多个随即元组插入到 Students 表中

二、 使用平台和工具

- ✧ Windows 10
- ✧ MS Access 2019
- ✧ Conda 开源软件包管理系统
- ✧ Visual Studio Code 中 Python 编程

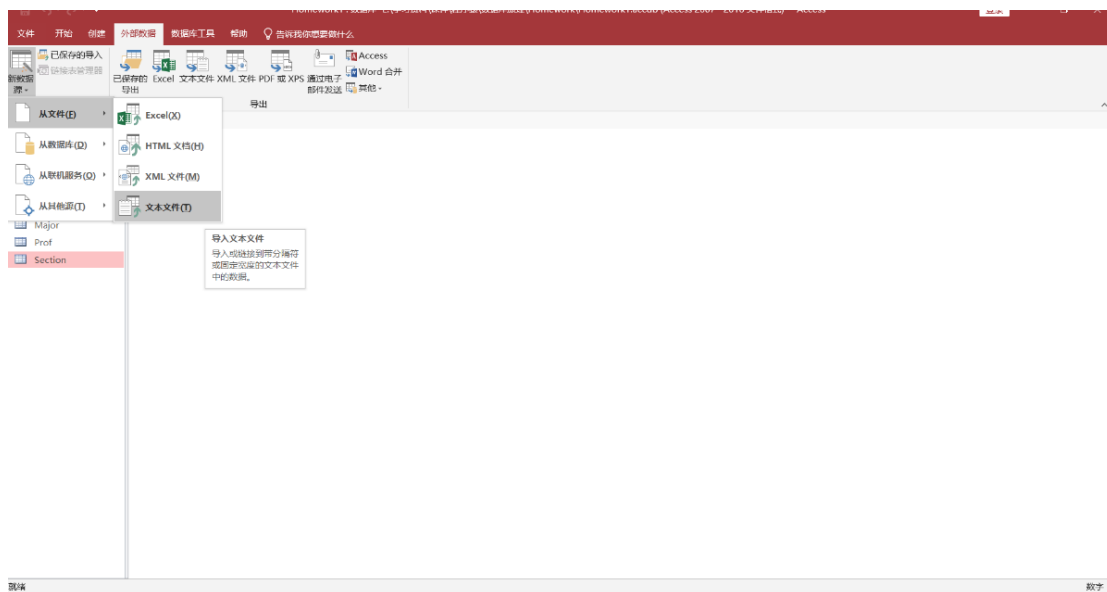
三、 实现方案

- ✧ 建立一个数据库

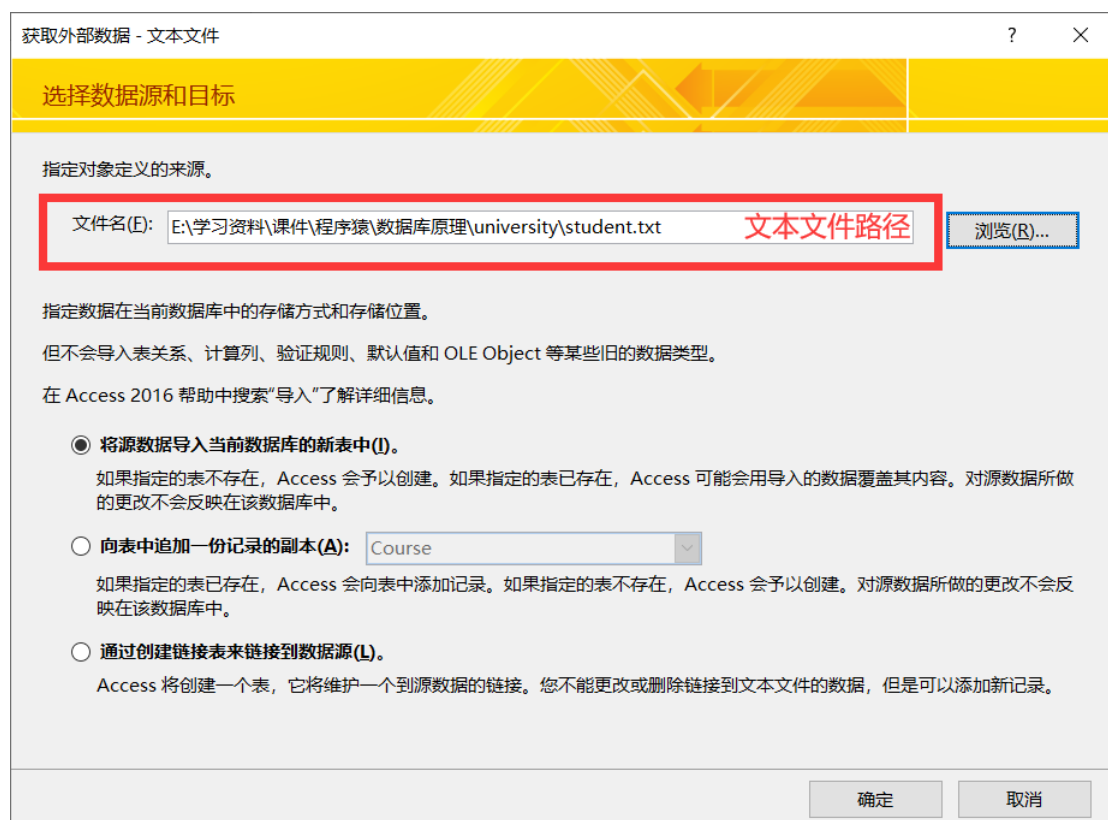


1. 右击文件夹空白处，选择新建 Microsoft Access Database 文件
2. 输入文件名 university.accdb
3. 完成数据库建立，打开数据库

- ✧ 导入文本文件



1. 选择外部数据→新数据源→从文件→文本文件



2. 指定文本文件路径

导入文本向导

数据似乎是“分隔符”格式。如果不是，请选择更能正确说明数据的格式。

☒ 带分隔符 - 用逗号或制表符之类的符号分隔每个字段(D)
 ☐ 固定宽度 - 字段之间使用空格使所有字段在列内对齐(W)

选择此项

来自文件 E:\学习资料\课件\程序猿\数据库原理\UNIVERSITY\STUDENT.TXT 的示例数据。

```

1 "sid" "sname" "sex" "age" "year" "gpa"
2 1"Jacobs, T. " "m"2953.60
3 2"Pierson, E. " "m"3253.50
4 3"Zeene, Ben N. " "m"2153.90
5 4"Sulfate, Barry M. " "m"1922.80
6 5"Form, Clara O. " "f"1813.30
7 6"Scott, Kim J. " "m"2013.80
8 7"Sather, Roberto B. " "m"2242.20
9 8"Stanley, Leotha T. " "m"2133.60
10 9"Smith, Joyce A. " "f"2142.00
11 10"Jones, David S. " "m"1923.50
12 11"Paul, Mary W. " "f"2353.60
13 12"Soong, V. " "f"2453.50
14 13"Kellerman, S. " "f"2132.90
15 14"Cheong, R. " "m"2543.00
16 15"Borchart, Sandra L. " "f"2653.90
17 16"Alsberg, David J. " "m"2553.50
18 17"Thorton, James Q. " "m"2842.70
  
```

高级(V)... 取消 < 上一步(B) 下一步(N) > 完成(F)

3. 指定数据为分隔符格式

导入文本向导

请确定所需的字段分隔符。选择适当的字段分隔符并在下面的预览窗口中查看文本效果。

请选择字段分隔符

☒ 制表符(T)
 ☐ 分号(S)
 ☐ 逗号(C)
 ☐ 空格(P)
 ☐ 其他(O):

☒ 第一行包含字段名称(R)
 文本识别符(Q):

| sid | sname | sex | age | year | gpa |
|-----|---------------------|-----|-----|------|------|
| 1 | Jacobs, T. | m | 29 | 5 | 3.60 |
| 2 | Pierson, E. | m | 32 | 5 | 3.50 |
| 3 | Zeene, Ben N. | m | 21 | 5 | 3.90 |
| 4 | Sulfate, Barry M. | m | 19 | 2 | 2.80 |
| 5 | Form, Clara O. | f | 18 | 1 | 3.30 |
| 6 | Scott, Kim J. | m | 20 | 1 | 3.80 |
| 7 | Sather, Roberto B. | m | 22 | 4 | 2.20 |
| 8 | Stanley, Leotha T. | m | 21 | 3 | 3.60 |
| 9 | Smith, Joyce A. | f | 21 | 4 | 2.00 |
| 10 | Jones, David S. | m | 19 | 2 | 3.50 |
| 11 | Paul, Mary W. | f | 23 | 5 | 3.60 |
| 12 | Soong, V. | f | 24 | 5 | 3.50 |
| 13 | Kellerman, S. | f | 21 | 3 | 2.90 |
| 14 | Cheong, R. | m | 25 | 4 | 3.00 |
| 15 | Borchart, Sandra L. | f | 26 | 5 | 3.90 |
| 16 | Alsberg, David J. | m | 25 | 5 | 3.50 |
| 17 | Thorton, James Q. | m | 28 | 4 | 2.70 |
| 18 | Gooch | m | 26 | 1 | 1.40 |

高级(V)... 取消 < 上一步(B) 下一步(N) > 完成(F)

4. 勾选制表符，第一行包含字段名

导入文本向导

Microsoft Access 建议您为新表定义一个主键。主键用来唯一地标识表中的每个记录。可使数据检索加快。

☐ 让 Access 添加主键(A)
☒ 我自己选择主键(C)
☐ 不要主键(O)

选择sid作为主键

| sid | sname | sex | age | year | gpa |
|-----|---------------------|-----|-----|------|------|
| 1 | Jacobs, T. | m | 29 | 5 | 3.60 |
| 2 | Pierson, E. | m | 32 | 5 | 3.50 |
| 3 | Zeene, Ben N. | m | 21 | 5 | 3.90 |
| 4 | Sulfate, Barry M. | m | 19 | 2 | 2.80 |
| 5 | Form, Clara O. | f | 18 | 1 | 3.30 |
| 6 | Scott, Kim J. | m | 20 | 1 | 3.80 |
| 7 | Sather, Roberto B. | m | 22 | 4 | 2.20 |
| 8 | Stanley, Leatha T. | m | 21 | 3 | 3.60 |
| 9 | Smith, Joyce A. | f | 21 | 4 | 2.00 |
| 10 | Jones, David S. | m | 19 | 2 | 3.50 |
| 11 | Paul, Mary W. | f | 23 | 5 | 3.60 |
| 12 | Soong, V. | f | 24 | 5 | 3.50 |
| 13 | Kellerman, S. | f | 21 | 3 | 2.90 |
| 14 | Cheong, R. | m | 25 | 4 | 3.00 |
| 15 | Borchart, Sandra L. | f | 26 | 5 | 3.90 |
| 16 | Alsberg, David J. | m | 25 | 5 | 3.50 |
| 17 | Thorton, James Q. | m | 28 | 4 | 2.70 |
| 18 | Gooch | m | 26 | 1 | 1.40 |

高级(V)... 取消 < 上一步(B) 下一步(N) > 完成(F)

5. 选择 sid 作为自定义主键，完成创建

Microsoft Access 2016 界面截图，显示数据库 university.accdb 中的 Student 表。表结构如下：

| sid | sname | sex | age | year | gpa |
|-----|---------------------|-----|-----|------|------|
| 1 | Jacobs, T. | m | 29 | 5 | 3.60 |
| 2 | Pierson, E. | m | 32 | 5 | 3.50 |
| 3 | Zeene, Ben N. | m | 21 | 5 | 3.90 |
| 4 | Sulfate, Barry M. | m | 19 | 2 | 2.80 |
| 5 | Form, Clara O. | f | 18 | 1 | 3.30 |
| 6 | Scott, Kim J. | m | 20 | 1 | 3.80 |
| 7 | Sather, Roberto B. | m | 22 | 4 | 2.20 |
| 8 | Stanley, Leatha T. | m | 21 | 3 | 3.60 |
| 9 | Smith, Joyce A. | f | 21 | 4 | 2.00 |
| 10 | Jones, David S. | m | 19 | 2 | 3.50 |
| 11 | Paul, Mary W. | f | 23 | 5 | 3.60 |
| 12 | Soong, V. | f | 24 | 5 | 3.50 |
| 13 | Kellerman, S. | f | 21 | 3 | 2.90 |
| 14 | Cheong, R. | m | 25 | 4 | 3.00 |
| 15 | Borchart, Sandra L. | f | 26 | 5 | 3.90 |
| 16 | Alsberg, David J. | m | 25 | 5 | 3.50 |
| 17 | Thorton, James Q. | m | 28 | 4 | 2.70 |
| 18 | Gooch | m | 26 | 1 | 1.40 |

6. 依次完成所有文本文件导入，得到最终数据库 university.accdb

✧ 编写 Python 程序（附录提供源码 auto_gen.py）

```

auto_gen.py X
auto_gen.py >
1 import pyodbc
2 from random import choice, uniform, randint
3
4 # 链接数据库
5 path = "Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=E:\\学习资料\\课程\\程序猿\\数据库原理\\Homework\\university.accdb;"
6 db = pyodbc.connect(path)
7 # db = pyodbc.win_connect_mdb(path)
8 cur = db.cursor()
9
  
```

导入依赖库

连接数据库

1. 导入依赖库，完成数据库连接

```
17 class student:
18     def __init__(self):
19         super().__init__()
20
21         # 名字list
22         self.firstname = ["Abigail", "Ada", "Adela", "Adelaide", "Afra", "Agatha", "Agens", "Alberta",
23                             "Alexia", "Alice", "Alma", "Althea", "Alva", "Amanda", "Amelia", "Amy", "Anastasia", "Andrea",
24                             "Barbara", "Belinda", "Bella", "Bella", "Belle", "Bernice", "Bertha", "Beryl", "Bess", "Betsy",
25                             "Betty", "Bealuah", "Beverly", "Blanche", "Bonnie", "Candice", "Cara", "Christine", "Claire",
26                             "Clara", "Dana", "Daphne", "Elizabeth", "Emma", "David", "Edward", "Eric", "Fred", "Garfield", "Gavin"]
27         # 名字list
28         self.lastname = ["Smith", "Miller", "Johnson", "Brown", "Jones", "Williams", "Black", "Longfellow", "Turner", "Hall",
29                             "Kent", "Brook", "Hill", "Field", "Green", "Wood", "Brown", "Longman", "Sharp", "Fong",
30                             "Back", "Finger", "Bird", "Bush", "Cotton", "Stock", "Hawk", "Reed", "George"]
31         # 性别list
32         self.sex = choice(['f', 'm'])
33         # 学生姓名
34         self.studentName = self.getName()
35         # 随机生成年龄
36         self.age = randint(18, 25)
37         # 随机生成年级
38         self.year = randint(1, 5)
39         # 随机生成GPA
40         self.gpa = round(uniform(0.00, 4.00), 1)
41
42     # 生成姓名
43     def getName(self):
44         studentName = choice(self.firstname) + ' ' + choice(self.lastname) + '.'
45         return studentName
46
47     # 返回学生信息
48     def toString(self):
49         return "" + self.studentName + "," + self.sex + "," + str(self.age) + "," + str(self.year) + "," + str("%.2f"%self.gpa) + ""
```

2. 创建 student 类，完成类的定义(包括各字段信息的随机生成信息)

```
51 # INSERT Database
52 for i in range(105, 5016):
53     # 生成学生对象
54     s = student()
55     # 插入学生信息
56     insert_str = "INSERT INTO Student VALUES({0},{1})".format(str(i), s.toString())
57     cur.execute(insert_str)
58
59 # 完成提交
60 cur.commit()
61
```

3. 循环随机生成学生对象，插入数据库

✧ 运行 Python 程序，完成 5000 条随机学生信息插入

| Student | | | | | | | 单击以添加 |
|---------|---------------|-----|-----|------|------|--|-------|
| sid | sname | sex | age | year | gpa | | |
| 1 | Jacobs, T. | m | 29 | 5 | 3.60 | | |
| 2 | Pierson, E. | m | 32 | 5 | 3.50 | | |
| 3 | Zeene, Ben N. | m | 21 | 5 | 3.90 | | |
| 4 | Sulfate, Bar | m | 19 | 2 | 2.80 | | |
| 5 | Form, Clara | f | 18 | 1 | 3.30 | | |
| 6 | Scott, Kim J. | m | 20 | 1 | 3.80 | | |
| 7 | Sather, Robe | m | 22 | 4 | 2.20 | | |
| 8 | Stanley, Leo | m | 21 | 3 | 3.60 | | |
| 9 | Smith, Joyce | f | 21 | 4 | 2.00 | | |
| 10 | Jones, David | m | 19 | 2 | 3.50 | | |
| 11 | Paul, Mary W. | f | 23 | 5 | 3.60 | | |
| 12 | Soong, V. | f | 24 | 5 | 3.50 | | |
| 13 | Kellerman, S. | f | 21 | 3 | 2.90 | | |
| 14 | Cheong, R. | m | 25 | 4 | 3.00 | | |
| 15 | Borchart, Sar | f | 26 | 5 | 3.90 | | |
| 16 | Alsberg, Davi | m | 25 | 5 | 3.50 | | |
| 17 | Thorton, Jame | m | 28 | 4 | 2.70 | | |
| 18 | Gooch | m | 26 | 1 | 1.40 | | |
| 19 | Smith, L. | m | 43 | 4 | 0.70 | | |
| 20 | Korpel, E. | f | 19 | 3 | 3.50 | | |
| 21 | Surk, K. | m | 23 | 2 | 2.50 | | |
| 22 | Emile, R. | m | 18 | 1 | 2.00 | | |
| 23 | Bomber, C. | f | 22 | 4 | 3.20 | | |
| 24 | Carter, Jimmy | m | 56 | 5 | 3.50 | | |

附录

```
// 随机学生信息插入代码 auto_gen.py
import pyodbc
from random import choice, uniform, randint

# 链接数据库
path = "Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=E:\\学习资料\\课件\\程序猿\\
数据库原理\\Homework\\university.accdb;"
db = pyodbc.connect(path)
# db = pypyodbc.win_connect_mdb(path)
cur = db.cursor()

''' Test
insert_str = "INSERT INTO Student VALUES(1,'叶宏庭','男',22,3,3.7)"

cur.execute(insert_str)
cur.commit()
'''

class student:
    def __init__(self):
        super().__init__()
        # 姓氏 list
        self.firstname = ["Abigail", "Ada", "Adela", "Adelaide", "Afra", "Agatha", "Agens",
                           "Alberta",
                           "Alexia", "Alice", "Alma", "Althea", "Alva", "Amanda", "Amelia", "Amy",
                           "Anastasia", "Andrea",
                           "Barbara", "Belinda", "Bella", "Bella", "Belle", "Bernice", "Bertha", "Beryl",
                           "Bess", "Besty",
                           "Betty", "Bealuah", "Beverly", "Blanche", "Bonnie", "Candice", "Cara",
                           "Christine", "Claire",
                           "Clara", "Dana", "Daphne", "Elizabeth", "Emma", "David", "Edward", "Eric",
                           "Fred", "Garfield", "Gavin"]
        # 名字 list
        self.lastname = ["Smith", "Miller", "Johnson", "Brown", "Jones", "Williams", "Black",
                           "Longfellow", "Turner", "Hall",
                           "Kent", "Brook", "Hill", "Field", "Green", "Wood", "Brown", "Longman",
                           "Sharp", "Fonng",
                           "Back", "Finger", "Bird", "Bush", "Cotton", "Stock", "Hawk", "Reed",
                           "George"]
        # 性别 list
        self.sex = choice(['f', 'm'])
        # 学生姓名
```

```

        self.studentName = self.getname()
        # 随机生成年龄
        self.age = randint(18, 25)
        # 随机生成年级
        self.year = randint(1, 5)
        # 随机生成 GPA
        self.gpa = round(uniform(0.00, 4.00), 1)

    # 生成姓名
    def getname(self):
        studentName = choice(self.firstname) + ' ' + choice(self.lastname) + '.'

        return studentName

    # 返回学生信息
    def toString(self):
        return "" + self.studentName + "," + self.sex + "," + str(self.age) + "," + str(self.year) +
        "," + str("%.2f"%self.gpa) + ""

# INSERT Database
for i in range(105, 5016):
    # 生成学生对象
    S = student()
    # 插入学生信息
    insert_str = "INSERT INTO Student VALUES({0},{1})".format(str(i), S.toString())
    cur.execute(insert_str)

# 完成提交
cur.commit()

```