# 数据库第一次实验

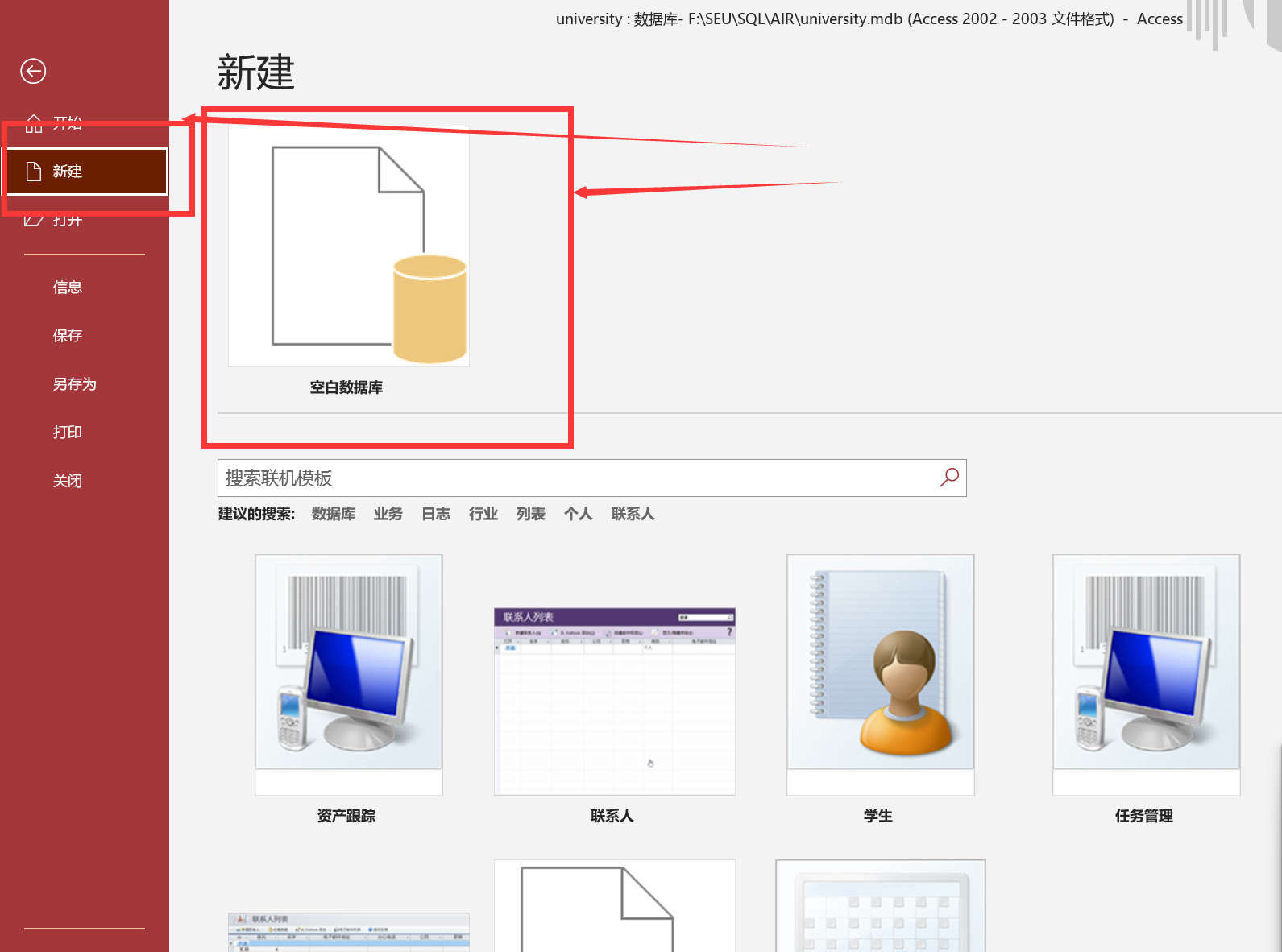
## 1、Familiar with MS Access

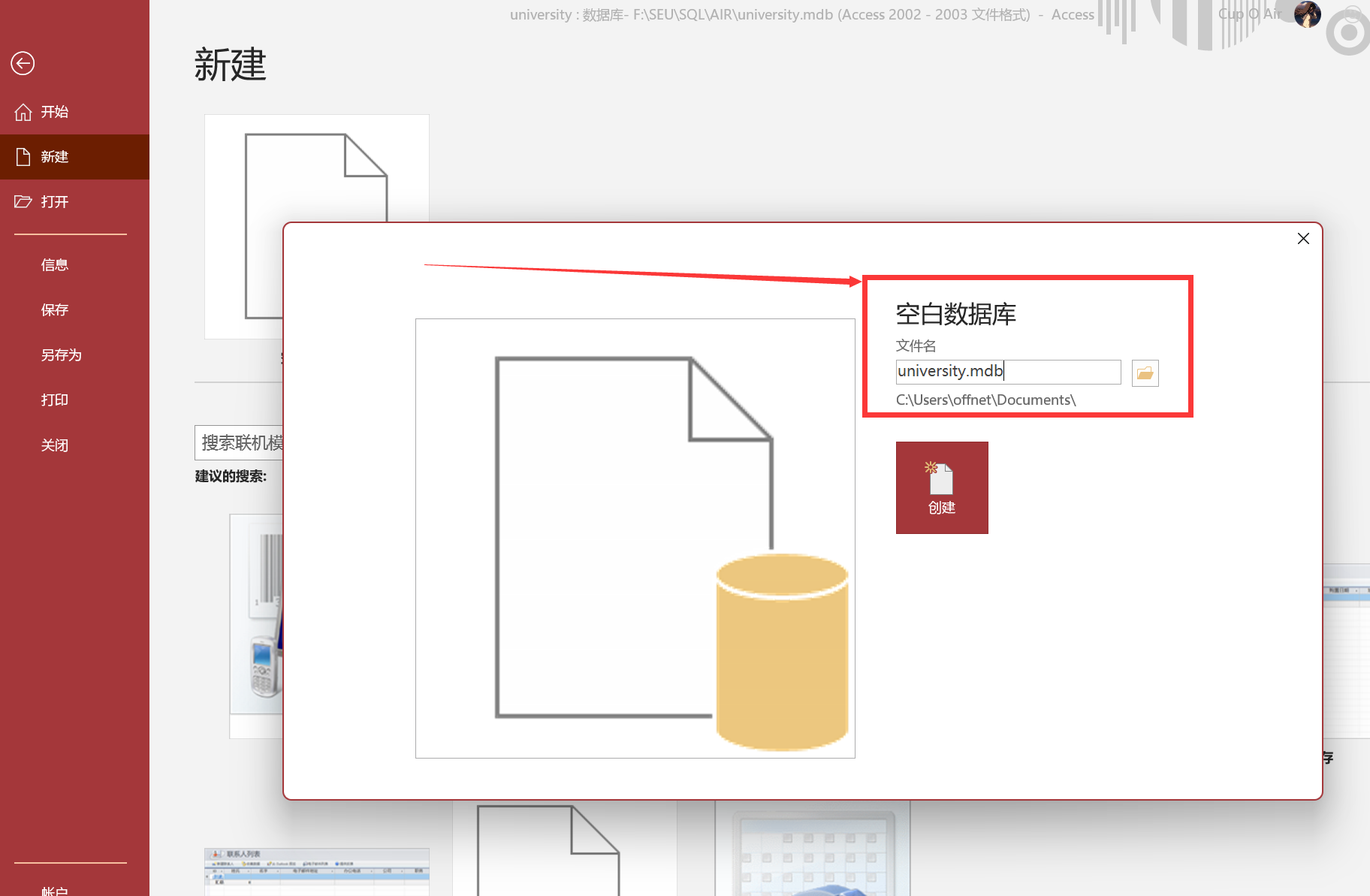
详细的使用介绍可参考[微软官网](https://support.microsoft.com/en-us/office/create-a-database-in-access-f200d95b-e429-4acc-98c1-b883d4e9fc0a?wt.mc_id=otc_access#)，Microsoft Office Access（前名Microsoft Access）是由微软发布的关系数据库管理系统。它结合了Microsoft Jet Database Engine和图形用户界面两项特点，是Microsoft Office的系统程序之一。

Access能够访问Access/Jet、Microsoft SQL Server、Oracle数据库，或者任何ODBC兼容数据库内的资料。熟练的软件设计师和资料分析师利用它来开发应用软件，而一些不熟练的程序员和非程序员的高级用户则能使用它来开发简单的应用软件。虽然它支持部分面向对象技术，但是未能成为一种完整的面向对象开发工具。

其实Access也是微软公司另一个通信程序的名字，想与ProComm以及其他类似程序来竞争。可是事后微软证实这是个失败计划，并且将它中止。数年后他们把名字重命名于此数据库软件。

## 2、Establish a university database(university.mdb)

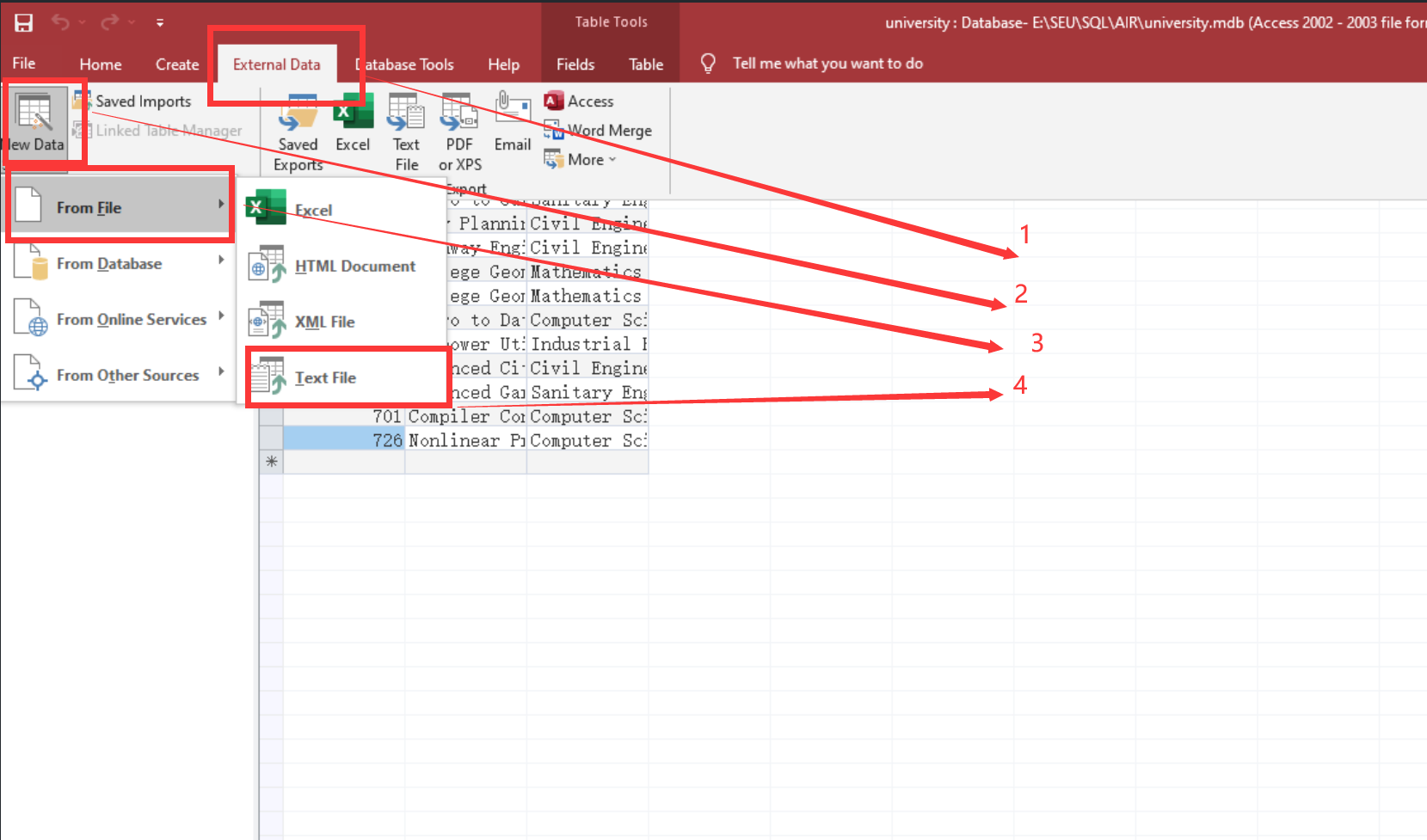
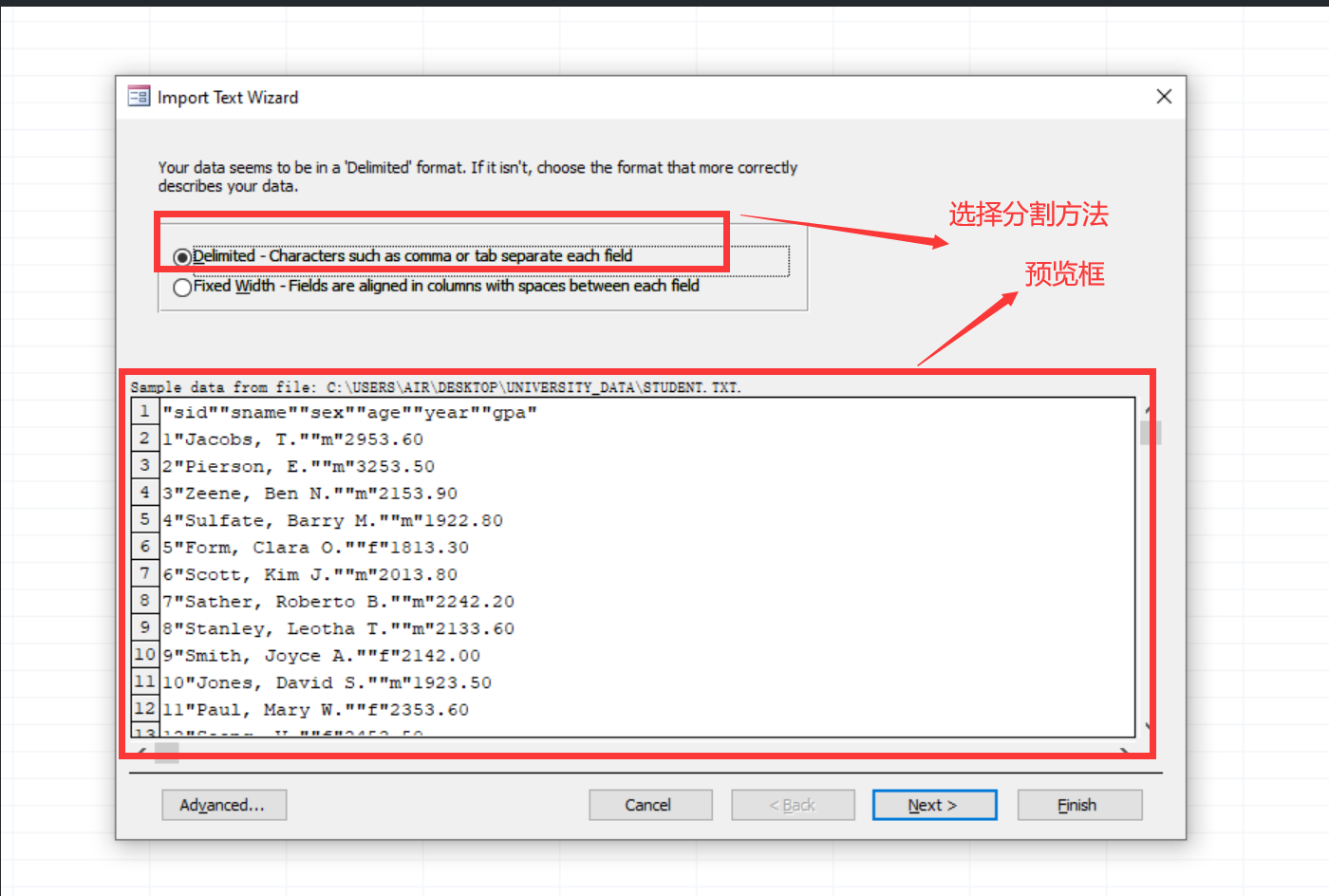
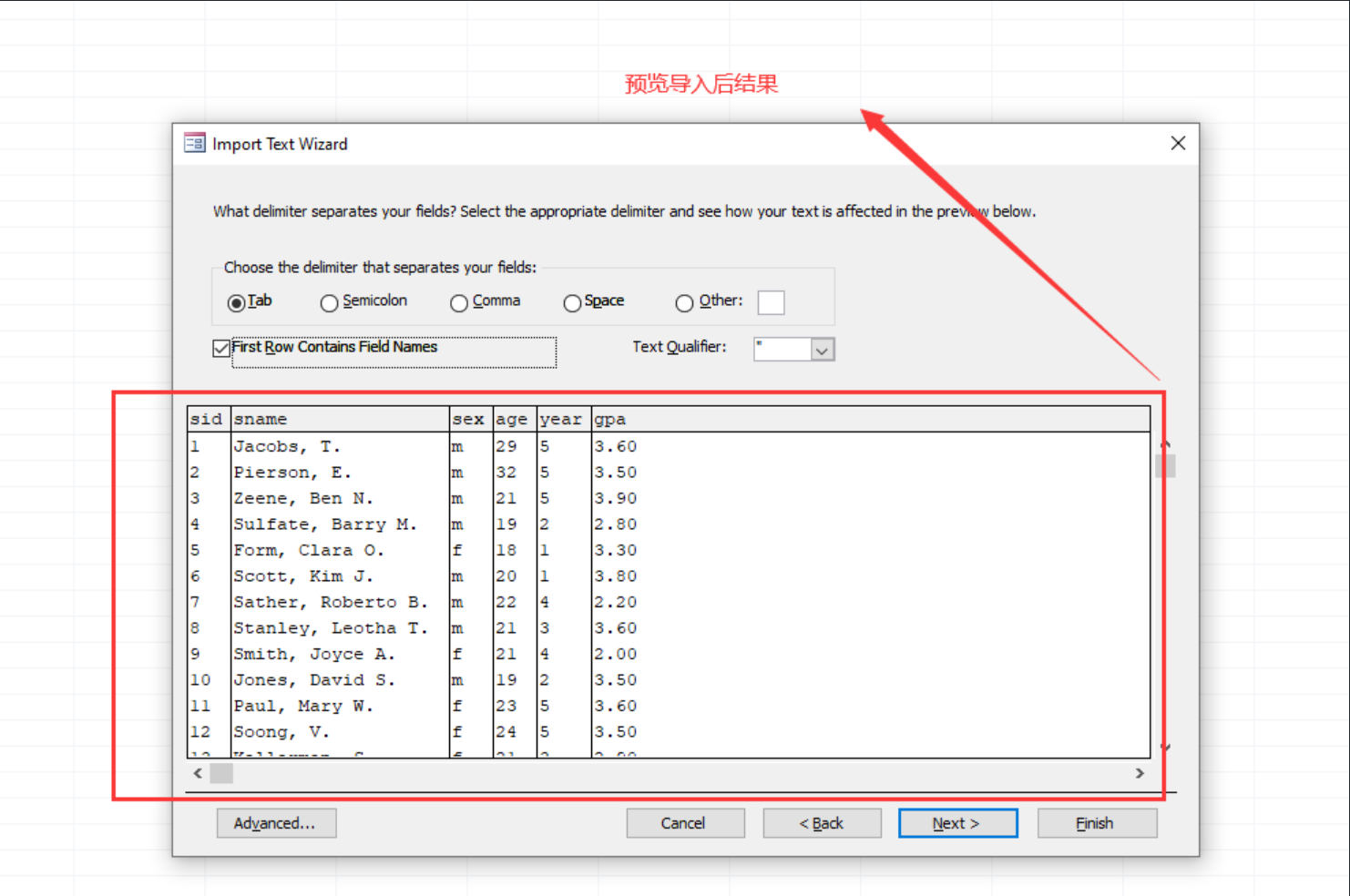
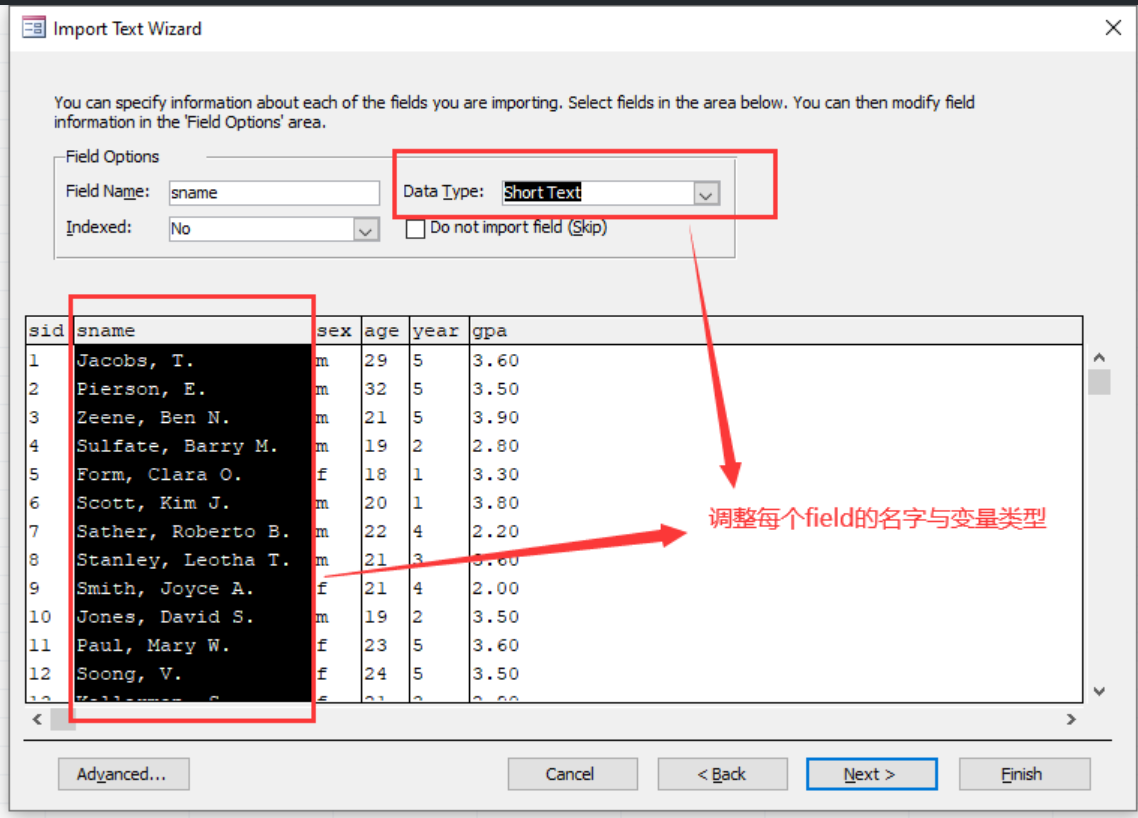
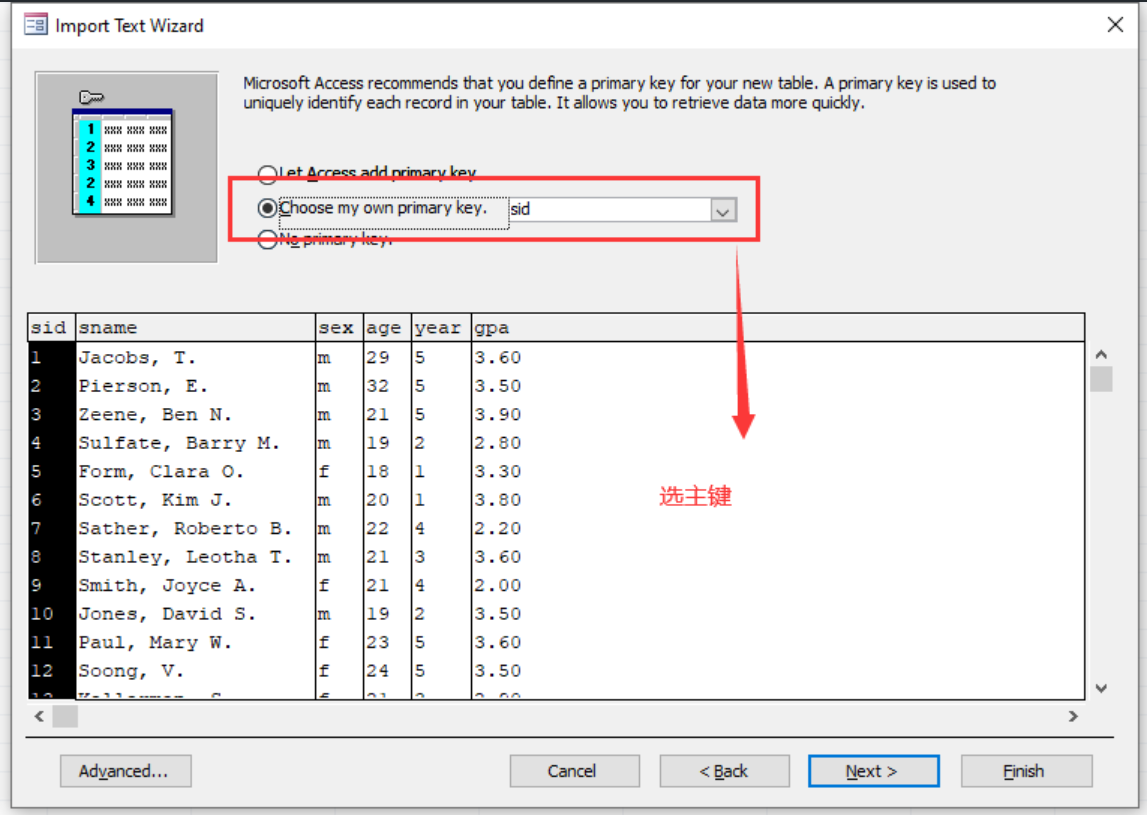
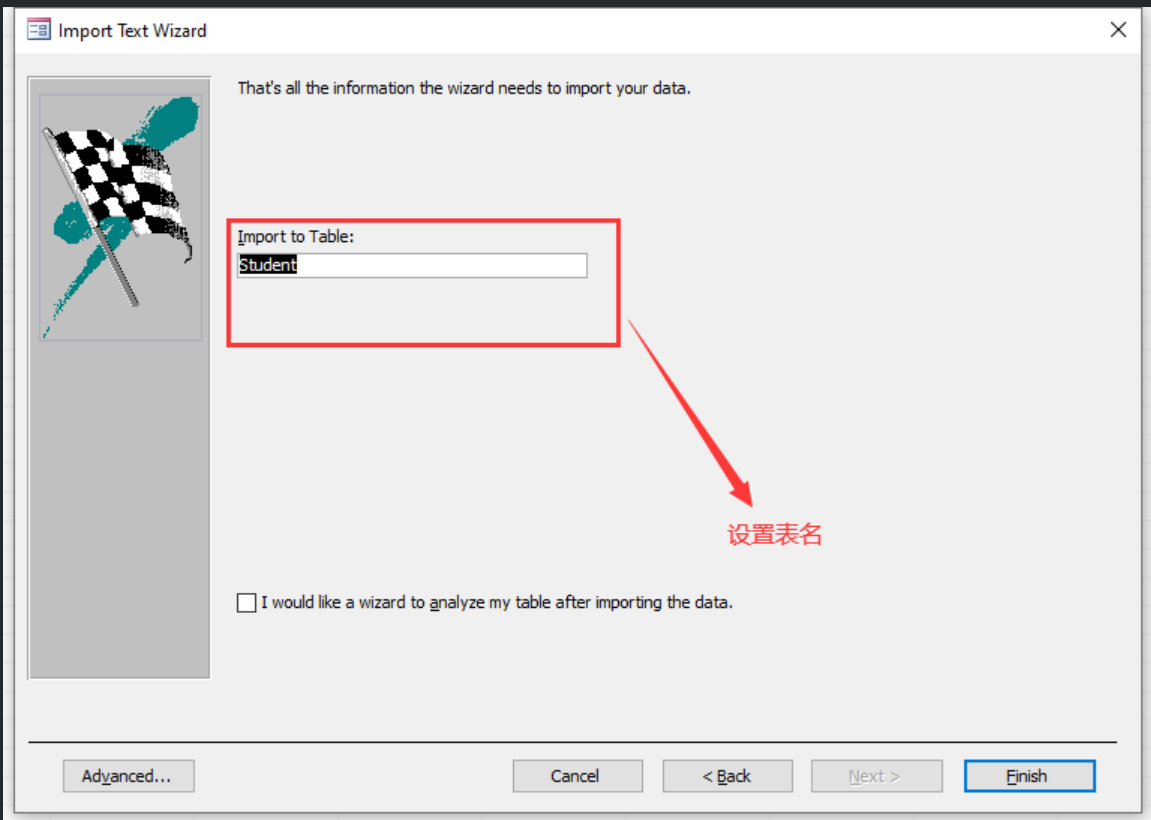
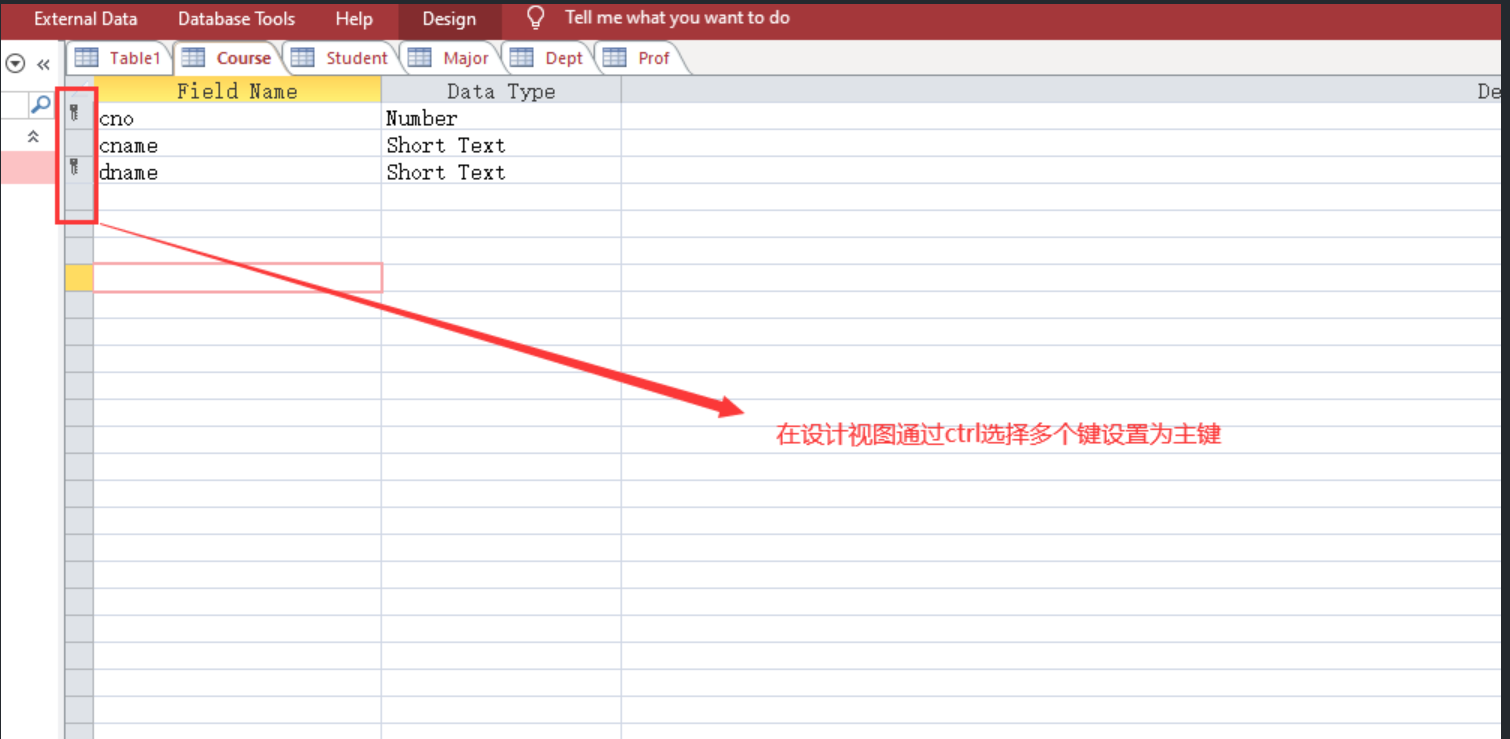
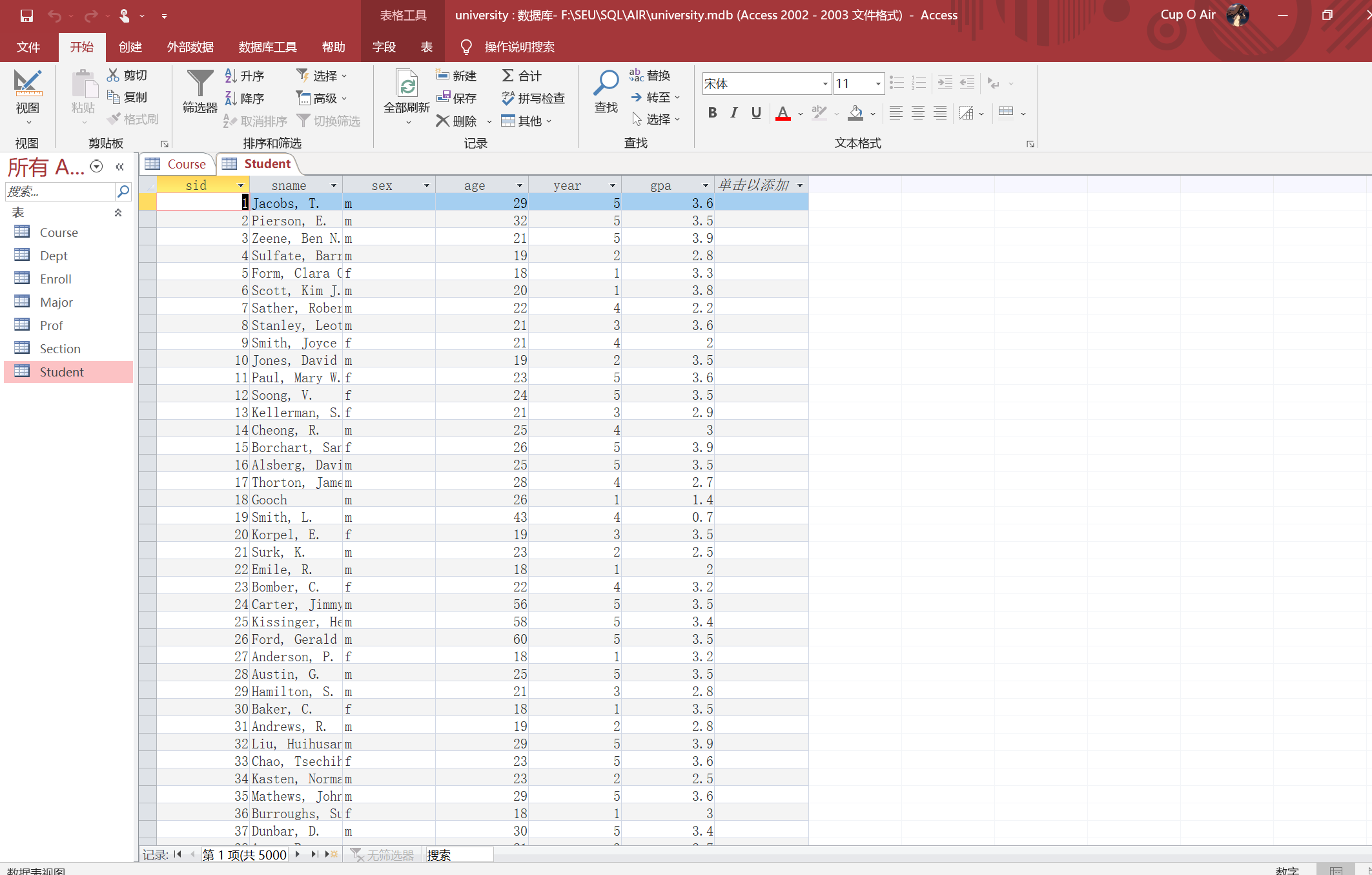
1. 打开MS Access
2. 点击创建空白数据库
3. 设置文件名为 "university"



## 

## 3、Load data into the database

由于每张表的导入方式流程基本相同，以下详细介绍导入student这张表并设置其主键为例：

1. 从文件导入
2. 通过预览文件内容选择合适的分割方法，对于学生这个文本文件可以看出，是通过tab来对每一个元素进行分隔的
3. 勾选将首行作为域名，同时可以在下方框预览表导入后的结果
4. 设置每个域的域名和其变量类型
5. 设置表的主键（只能选择一个，若要设置多个后面会提到）
6. 完成并设置表的名字
7. 设置多个域为主键，打开表的设计视图，多选键后选择设为主键即可
8. 同样的操作完成其余表的导入，最终结果如图

## 

## 4、Make a group of random tuples for student table and let student has more than 5000 tuples

首先生成一个具有5000条记录的文本文件，然后按上述方法导入即可。

1. 循环5000次，每次按行向文件写入一条字符串，字符串格式按照学生表中信息生成

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| * sid | * sname | * sex | * age | * year | * gpa |

1. 以下介绍每个信息生成的方法

* sid: 通过循环的次数决定
* sname: 通过名字生成函数，这个函数将大量随机的名字放入一个集合，再随机的取出两个单词组成名字，详细代码见附件pynamegenerator.py,以下为核心代码

def gen\_two\_words(split=' ', lowercase=False):  
 size = len(ALL\_ENG\_NAMES) - 1  
 idx\_first = random.randint(0, size)  
 idx\_last = random.randint(0, size)  
 first\_words = ALL\_ENG\_NAMES[idx\_first][1:]  
 last\_words = ALL\_ENG\_NAMES[idx\_last][1:]  
 if lowercase:  
 first\_words = first\_words.lower()  
 last\_words = last\_words.lower()  
 return '%s%s%s' % (first\_words, split, last\_words)

* sex: 随机生成0或1， 0为男的，1为女生

def generateSex():  
 if random.randint(0, 1) == 0:  
 return '"m"'  
 else:  
 return '"f"'

* age: 为18到29的随机数
* year: 为1到4的随机数
* gpa: 为0.1到4.0的随机数
* 在每个条目之间插入tab分隔，最终逐行写入

1. 核心代码，详情见附件randomRecord.py

def createRecord(i):  
 def generateSex():  
 if random.randint(0, 1) == 0:  
 return '"m"'  
 else:  
 return '"f"'  
 record = ''  
 record += str(i)  
 record += '\t'  
 record += '"'  
 record += pynamesgenerator.gen\_two\_words()  
 record += '"'  
 record += '\t'  
 record += generateSex()  
 record += '\t'  
 record += str(random.randint(18, 30))  
 record += '\t'  
 record += str(random.randint(1, 5))  
 record += '\t'  
 record += str(random.randint(1, 40) / 10.00)  
 record += '\n'  
 return record

### 最后调用生成记录的函数5000次，并将其输出为文件将文本文件记录导入到数据库中。

## 5、心得体会

【心得】: 本次实验让我更加了解了关系型数据库的创建与导入，通过实操体会到了本课程不仅仅围绕着理论，同时也映射着我们所处的世界。通过对university.mdb的创建，了解到数据库是如何将现实中的信息储存与关联起来的