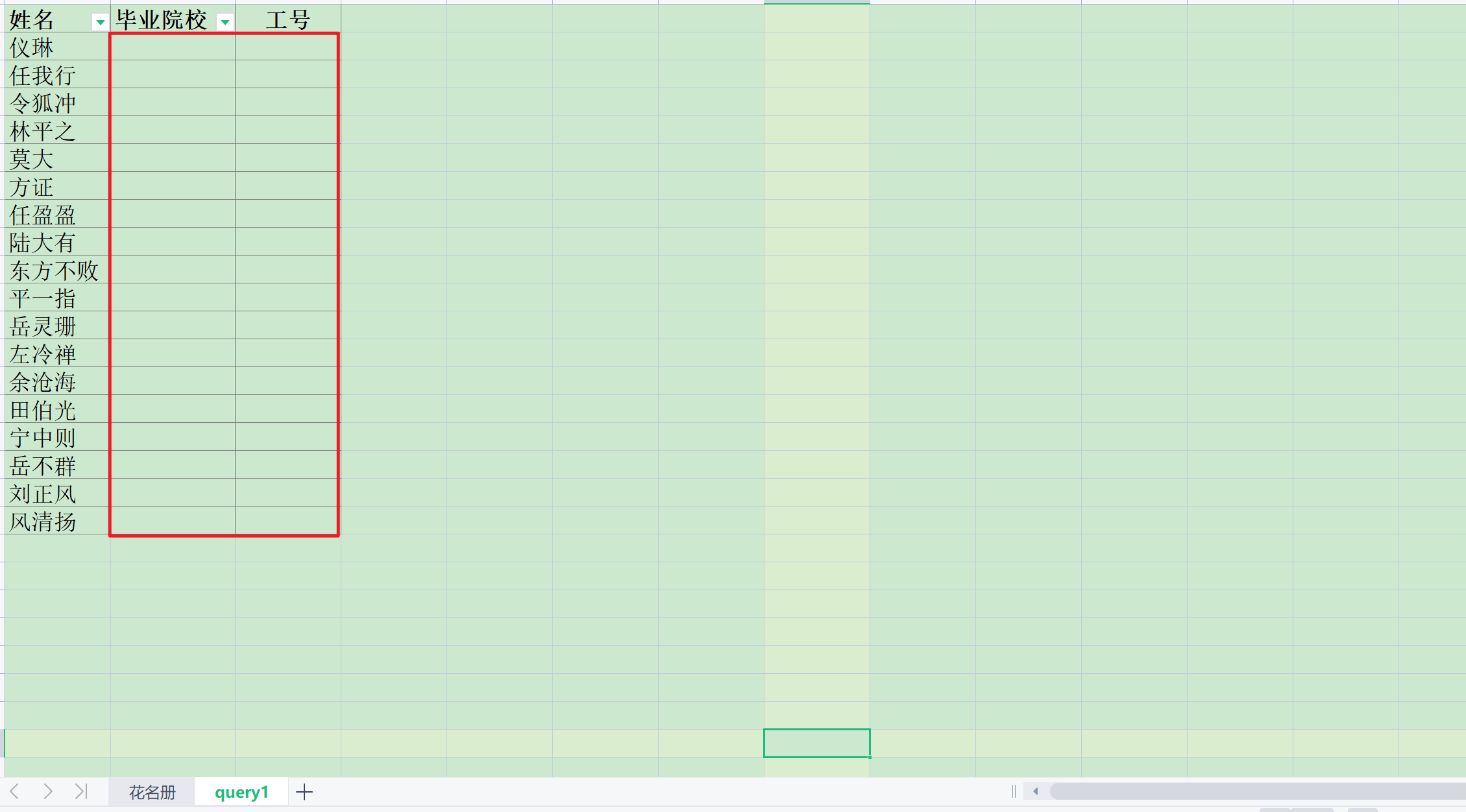
Vlookup查找函数详解：如何用三种工具Excel、Python、SQL解决同一个问题：

摘要：通过举一反三的形式，在EXCEL、Python、SQL三种不同工具下解决同一个实际问题

本例问题：目前工作簿xiao.xlsx里面有两张表格，其中一张表存放基础信息命名为花名册，另一张表我们操作查询用，暂命名为query1(查询1），我们根据“姓名”列信息，从花名册中找到对应人员的毕业院校和工号：

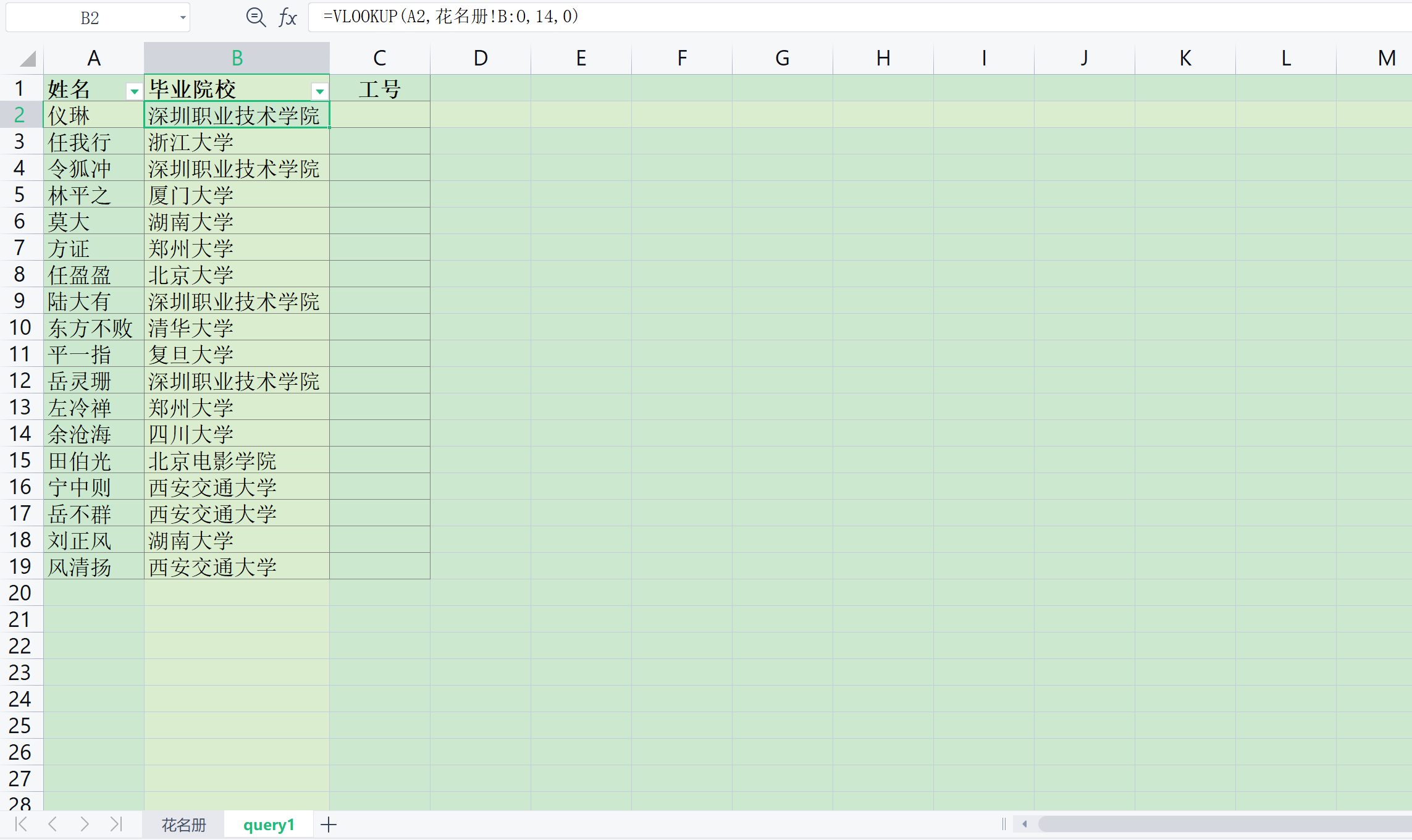




一、EXCEL中实现：

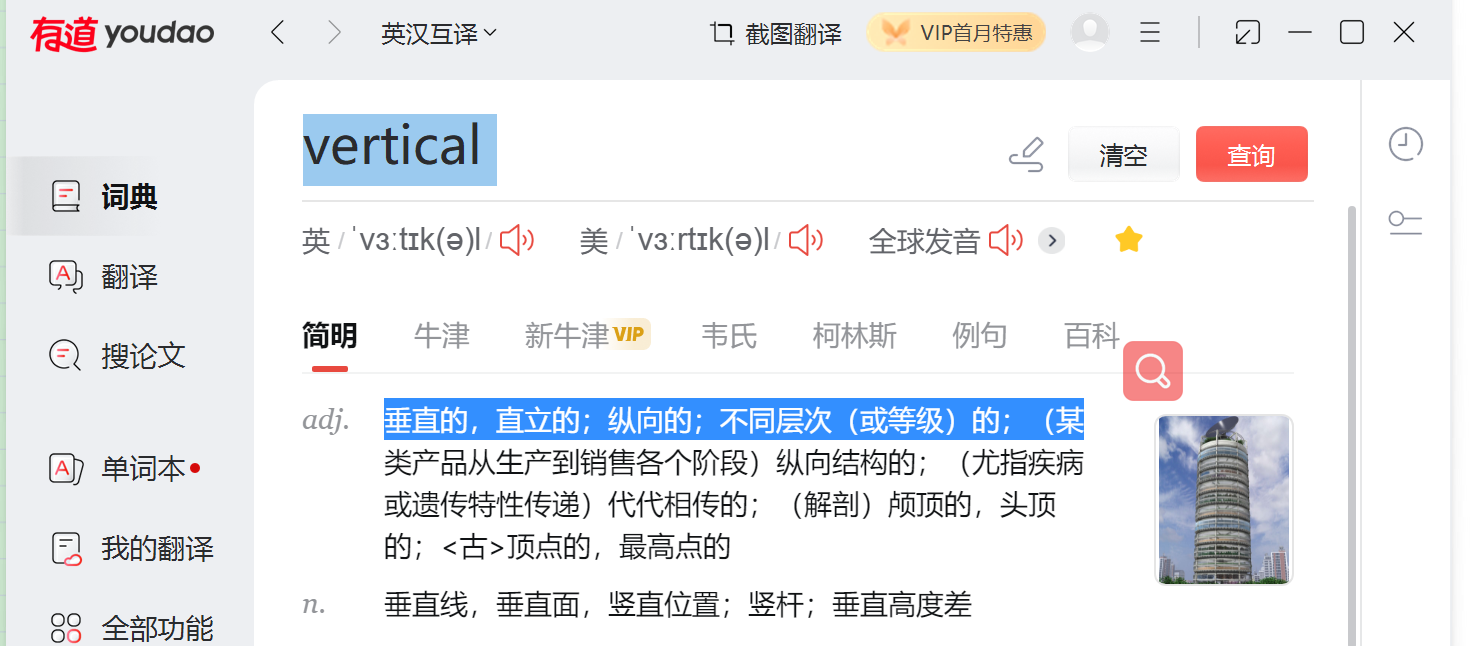
我们在B2单元格输入函数：=VLOOKUP(A2,花名册!B:O,14,0)

然后下拉或者全选后用CTRL+D填充即可完成查询

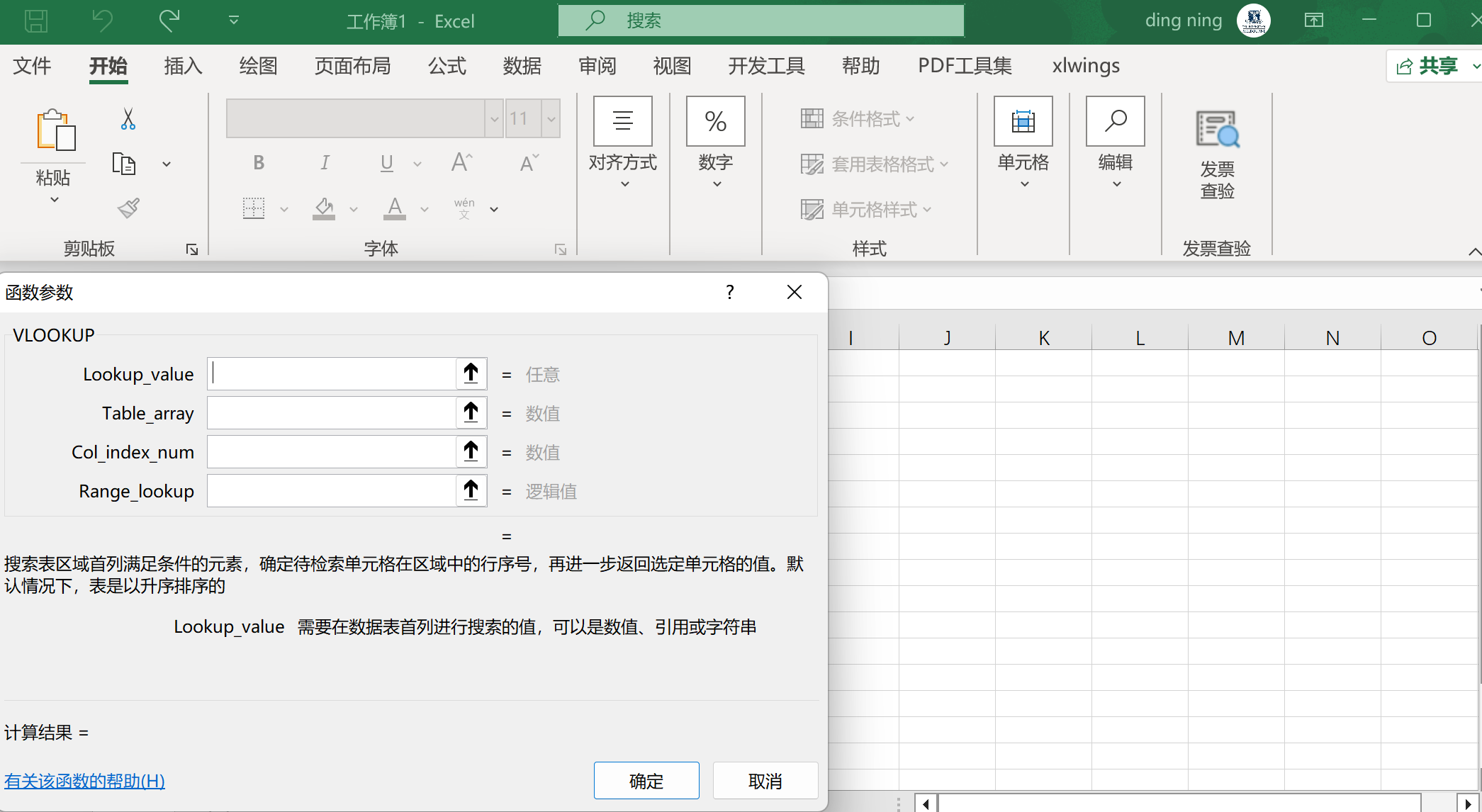


这就是大名鼎鼎的VLOOKUP函数，lookup英文是查找的意思，前面的V是英文单词vertical的意思，表示垂直，以此类推

有一个小众函数HLOOKUP里面的H代表horizontal水平的意思，与VLOOKUP的用法相当于旋转了90度，用处极少

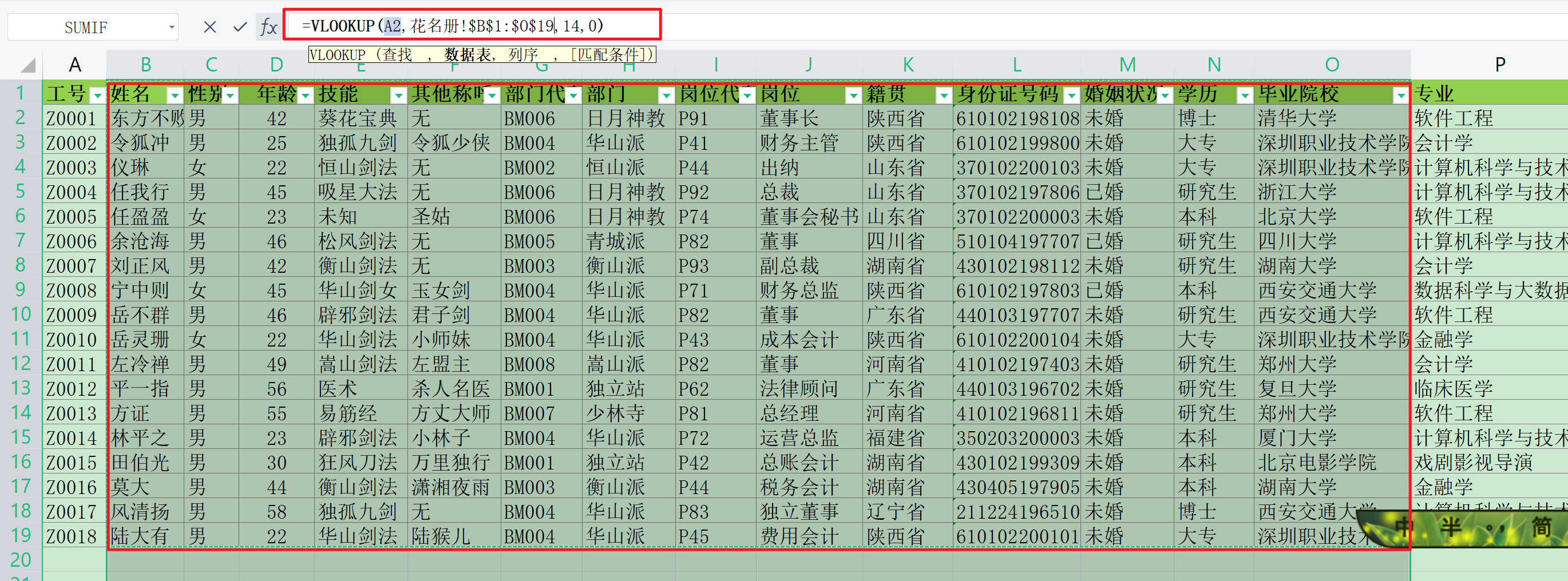


Vlookup用法说明：Vlookup( lookup\_value ,table\_array,col\_index\_num,[range\_lookup] )

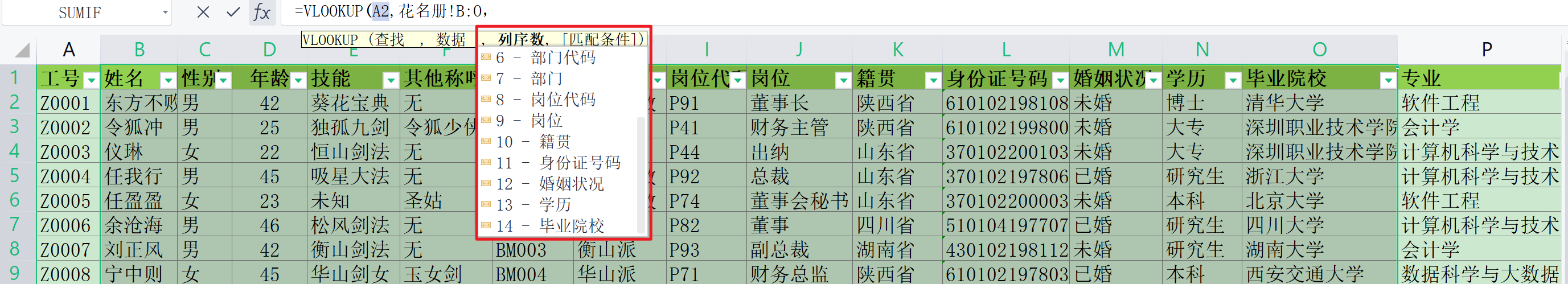


第一个参数 lookup\_value是要查找的值，这里我们查“姓名”，所以第一个参数直接取A2即可，当函数下拉填充的时候，第一个参数就会分别变成A3，A4，A5……直到最后一个

第二个参数table\_array 是要查询的范围，table是表的意思，array是数组的意思，这里表面要查找的范围区域，这里我们选择“花名册!B:O”列，因为我们是跨表进行查询，所以！前面是引用的表名，这个参数也可以用绝对引用，指定一个绝对范围，直接选B:O就是默认为B列到O列所有的行都选择上，如果指定行号也是可以的，比如下面这个绝对引用VLOOKUP(A2,花名册!$B$1:$O$19,14,0)，此时第二个参数就变为“花名册!$B$1:$O$19”，这时从第一列再找下拉，第二个参数不会改变，即指定了B1:O19这个范围，如果没有$符号的话，此时则是相对引用，往下拉第二个就是变成B2:O20,依次类推，所以一定要加绝对引用符号（附：增加或者减除绝对引用符号“$”的快捷键是 F4）



第三个参数Col\_index\_num是要选择的列的序数，即选第几列的值，B列为第1列，按递增顺O列为第14列，我们直接输入参数14即可，这个地方使用WPS会好一些，输入函数的时候，会提示出列名和对应的顺序号，如下图，但是使用微软OFFICE则不会有提示，这时往往就是自己数是第几列，或者叠加其他公式进行取第几列，会变得很麻烦（一个新函数Xlookup完美地解决了这个痛点，也解决了Vlookup倒查不便的困局，后面再说这个函数）



第四个参数range\_lookup是匹配条件，输入逻辑值，也就是0或者1，0代表False是完全匹配，1代表True是模糊匹配，实际应用中一般都是完全匹配，我们这里填0即可

我们再进行下一个查询，查找“工号”，这时会发现这个工号在花名册表里面“姓名”列的左边，而Vlookup函数是从左往右查询的，这怎么办？

如果非要用Vlookup这个函数就需要构建数组，将我们上面说的第二个参数范围进行在函数内部强制互调顺序，就是把工号调整到姓名列的后面，再完成查询，公式如下：

=VLOOKUP(A2,IF({1,0},花名册!$B$1:$B$19,花名册!$A$1:$A$19),2,0)



这个函数的难点在于IF那里，{}是构建一个数组，这段代码相当于两个if语句：if(1,花名册!$B$1:$B$19,花名册!$A$1:$A$19)和if(0,花名册!$B$1:$B$19,花名册!$A$1:$A$19),前面那个if取出来花名册!$B$1:$B$19，后面那个取出来花名册!$A$1:$A$19，然后这两项合在一起作为一个二维数组，有的同学可能会问，那直接把第二个参数写成{花名册!$B$1:$B$19，花名册!$A$1:$A$19}这种数组格式可不可以，答案是不可以！至于为什么，我现在也没有完全弄明白，以后弄明白了再补充给大家。

可以单独复制这个公式：=IF({1,0},花名册!$B$1:$B$19,花名册!$A$1:$A$19)，然后选中F1:G19单元格，在F1单元格输入公式，按CTRL+SHIFT+回车键(数组函数指定组合按键），就可以看到这个公式的值就是把姓名和工号两列调换了顺序组成的一个新的单元格区域，是一个数组区域，当你删除某个或者某行单元格时，会发现竟然删不了，因为这就是个函数的功能，要删除则需要把整个区域全删除，要么则不删

如果嫌这个公式难于理解，可以换一个其他函数，可用的还有3种：xlookup,index+match组合，lookup

先说xlookup函数，这个函数估计以后会大放异彩，将有可能完全取代Vlookup函数，原因有二：一是VLOOKUP有的功能它都能实现而且用法更简单，前面说的用VLOOKUP需要输入列的序号（列多的时候需要自己查数），而Xlookup不用输入序号，二是Xlookup可以轻易实现VLOOKUP倒查功能，再也不用构建那个数组公式

本例中在C2单元格输入公式：=XLOOKUP(A2,花名册!B:B,花名册!A:A)，向拉填充即可完成取数，超级方便

xlookup函数用法说明：（注：此函数WPS可以使用，EXCEL2021版本及以上版本可以使用）

XLOOKUP(lookup\_value, lookup\_array, return\_array, [if\_not\_found], [match\_mode], [search\_mode])



一共有6个参数，3个必填，3个可选

第一个参数：要查找的值，和VLOOKUP第一个参数一样

第二个参数：要查找的区域，相当于VLOOKUP里面第二个参数区域里的首列

第三个参数：要返回的区域，相当于VLOOKUP里面第二个参数区域里的返回值所在的列

第四个参数（非必要）：如果没查到，返回的值，如果不填参数，默认返回  #N/A

第五个参数（非必要）：匹配模式,参数如下

0 - 完全匹配。 如果未找到，则返回 #N/A。 这是默认选项。这个相当于VLOOKUP里面最后一个参数0

-1 - 完全匹配。 如果没有找到，则返回下一个较小的项。

1 - 完全匹配。 如果没有找到，则返回下一个较大的项。

2 - 通配符匹配，其中 \*, ? 和 ~ 有特殊含义。

第六个参数：搜索模式，参数如下

1 - 从第一项开始执行搜索。 这是默认选项。VLOOKUP就是默认从上到下进行匹配出第一个值的

-1 - 从最后一项开始执行反向搜索，这项是从下向上匹配，取出最后一个值，这个功能在数据进行增行更新时比较有用武之地。

2 - 执行依赖于 lookup\_array 按升序排序的二进制搜索。 如果未排序，将返回无效结果。

2 - 执行依赖于 lookup\_array 按降序排序的二进制搜索。 如果未排序，将返回无效结果。

演示一下从下往上匹配以及匹配不到返回特定值：

在原数据增加一行，‘陆大有’岗位由原来“费用会计”变为“总账会计”，后面是否最新为1表示最新，这时我们再匹配就要从下往上匹配

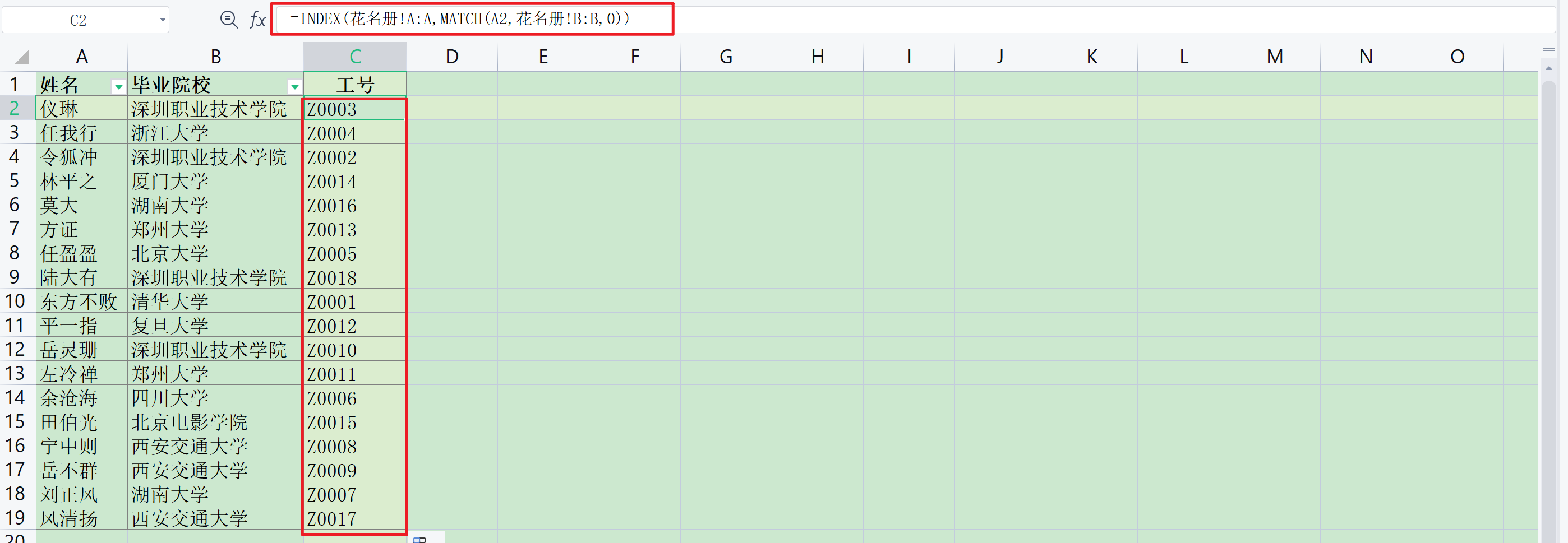




在B2列输入公式，=XLOOKUP(A2,花名册!B:B,花名册!J:J,"查无",0,-1)，如果查不到对应的值就返回“查无”，后面0表示精确匹配，-1表示从下向上匹配，我们可以看到陆大有的岗位查到的就是最下面的“总账会计”,向问天因为花名册里没有他的资料，所以返回”查无“

回到前面的查询，还可以用INDEX+MATCH组合进行查询

在C2单元格输入公式 “=INDEX(花名册!A:A,MATCH(A2,花名册!B:B,0))”向拉填充后即可完成查询



这是一个组合函数，相当于数学里面的复合函数，原理是一样的，其中一个函数的值作为另一个函数的参数，最终得到想要的结果

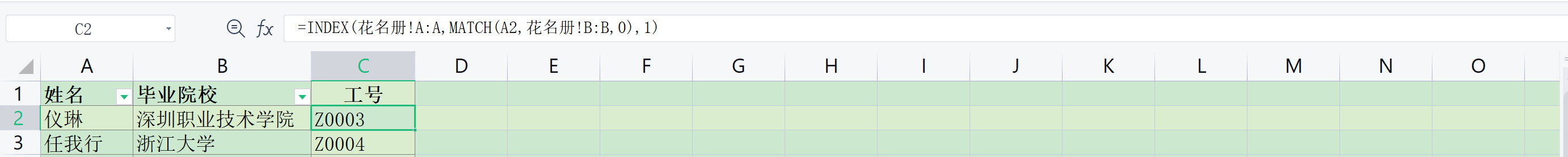
INDEX函数说明：函数如其名，是索引的意思，也叫索引函数，

INDEX(array, row\_num, [column\_num])

第一个参数（必填）：array（数组），也就是表格区域

第二个参数（必填）：row\_num(行号）

第三个参数（可选）：column\_num(列号），如果只选一列，则这个参数就是不必要填了，本例中就是，只选了花名册中的第一列（只有一列，没必要填列的参数了，然后根据MATCH函数传递过来行数，当然填上1也是一样的结果



这个函数大概要做的就是：第一个参数画个区域，后面两个参数进行二维定位，行号多少，列号多少，类似数学上的二维坐标（x,y)

MATCH函数说明：match英文是相匹配的意思

MATCH(lookup\_value, lookup\_array, [match\_type])（前两个必填，第三个可选）

第1个参数：要查找的值，相当于VLOOKUP的第一个参数

第2个参数：要查找的区域

第3个参数：匹配类型：0，1，-1

1： 查找小于或等于 lookup\_value 的最大值。

0： 查找完全等于 lookup\_value 的第一个值。 这个较常用

-1：查找大于或等于 lookup\_value 的最小值。

注意事项：

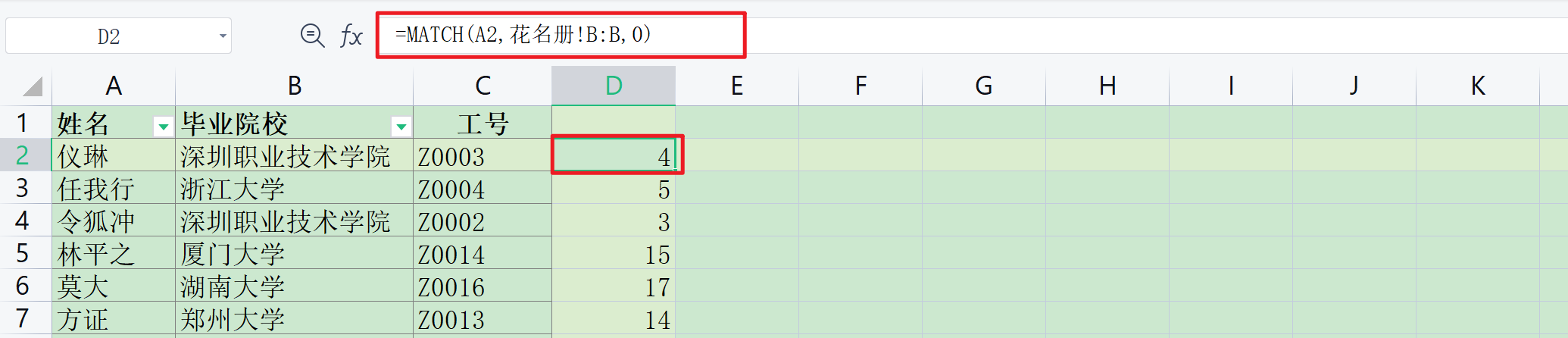
MATCH 返回匹配值在 lookup\_array 中的位置，而非其值本身。

匹配文本值时，MATCH 函数不区分大小写字母。

如果 match\_type 为 0 且 lookup\_value 为文本字符串，可在 lookup\_value 参数中使用通配符问号 (?) 和星号 (\*) ，问号匹配任意单个字符；星号匹配任意一串字符。 如果要查找实际的问号或星号，请在字符前键入波形符 (~)。

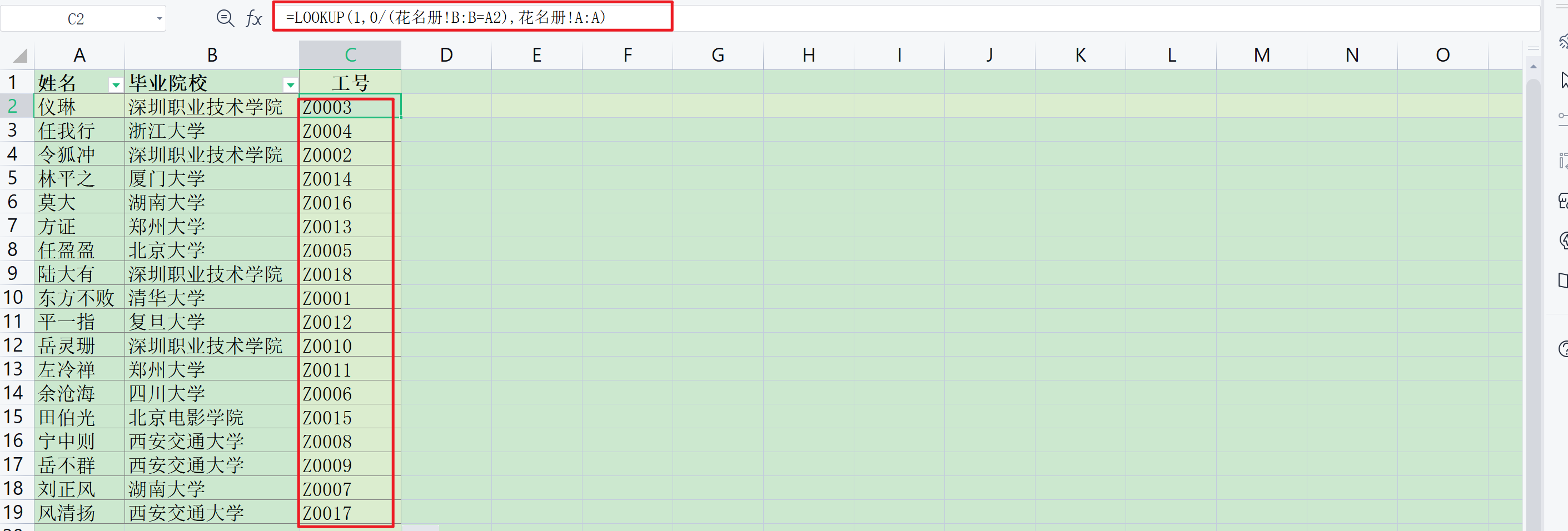
在本例中：=MATCH(A2,花名册!B:B,0)

这个公式相当于找出A2(仪琳) 在花名册“姓名”列里排在第几行，返回的是一个数字（也就是行号），然后传递给前面的INDEX索引函数进行取值



回到前面的查询，还可以用lookup进行查询

在C2单元格输入公式：=LOOKUP(1,0/(花名册!B:B=A2),花名册!A:A) 向下填充即可完成查询



先说一下这个LOOKUP用法，其中LOOKUP(1,0/（(条件1）\*（条件2）\*（条件3）\*……），返回值区域）这种形式最为常用，江湖人称“以大查小”

LOOKUP(lookup\_value, lookup\_vector, [result\_vector])

第一个参数：要查找的值，本例中为数值1

第二个参数：要查找的区域（vector,有单行矩阵，单列矩阵的意思，也可以简单把这个理解为一个一维数组）

第三个参数：要返回的值所有的区域，注意这第三个参数的范围要和第二个区域的范围相同

在本例中，第二个参数“0/(花名册!B:B=A2)”这列实际会返回一个数组{#DIV/0!，#DIV/0!，……0……#DIV/0!}这里面会有很多个错误#DIV/0!，当花名册B列（姓名）中为“仪琳”正好等于A2(仪琳）时，则会返回0，因为0/1=0，而当不等于的时候都是0/0=#DIV/0!，

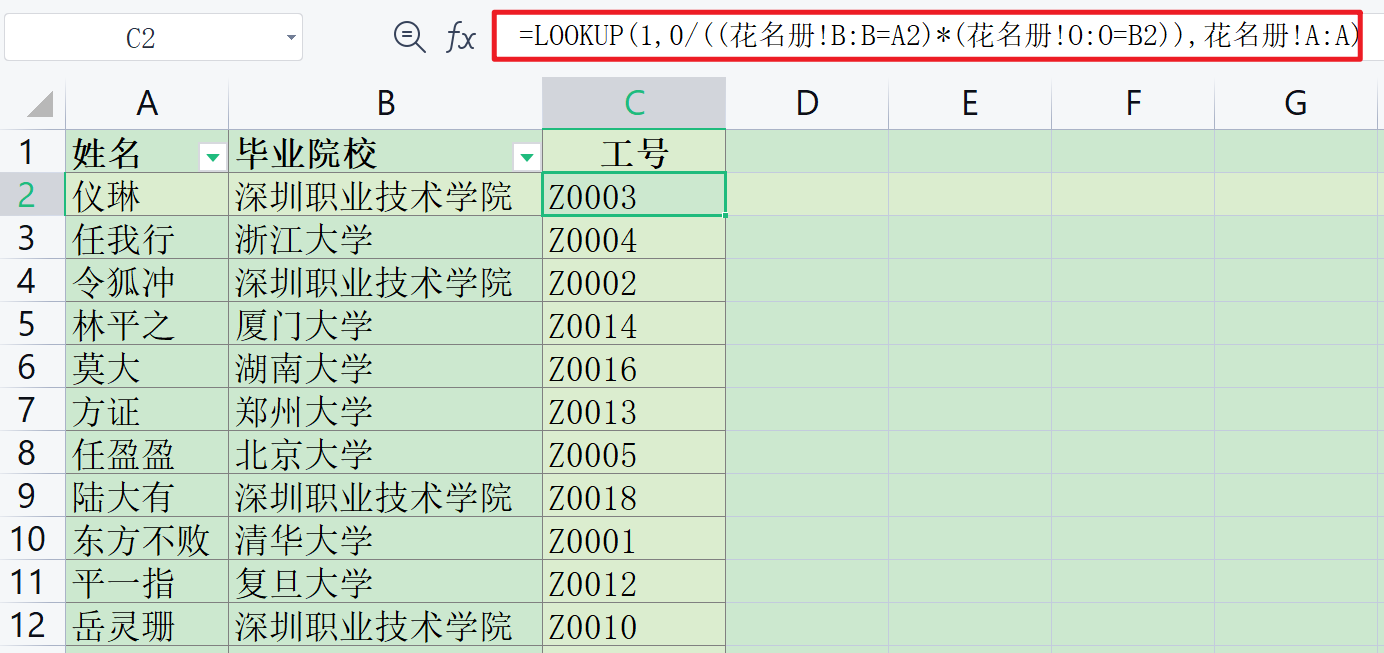
而lookup查找1时因为第二个参数并没有1这个值，所以会查找小于1的数值，则会找到0，而忽略其他的错误值，则当在对应的行找到姓名列“仪琳”时则正好查找这个0值，而第三个参数会跟随第2个参数区域，同样转为数组进行和第二个参数一一对应，

解释起来有点类似这样，每个单元格都是传递这些数据，后面两个构成对应的数组，如果第二个参数里有一个值为0，则进行选取，其他的都PASS掉，这样下来每行都是只有一个相等，都有第二个参数是0的一对值，如果出现多个0值，LOOKUP的机制是从原表从下往上查询，类似于XLOOKUP里的第6个参数为-1时的查询

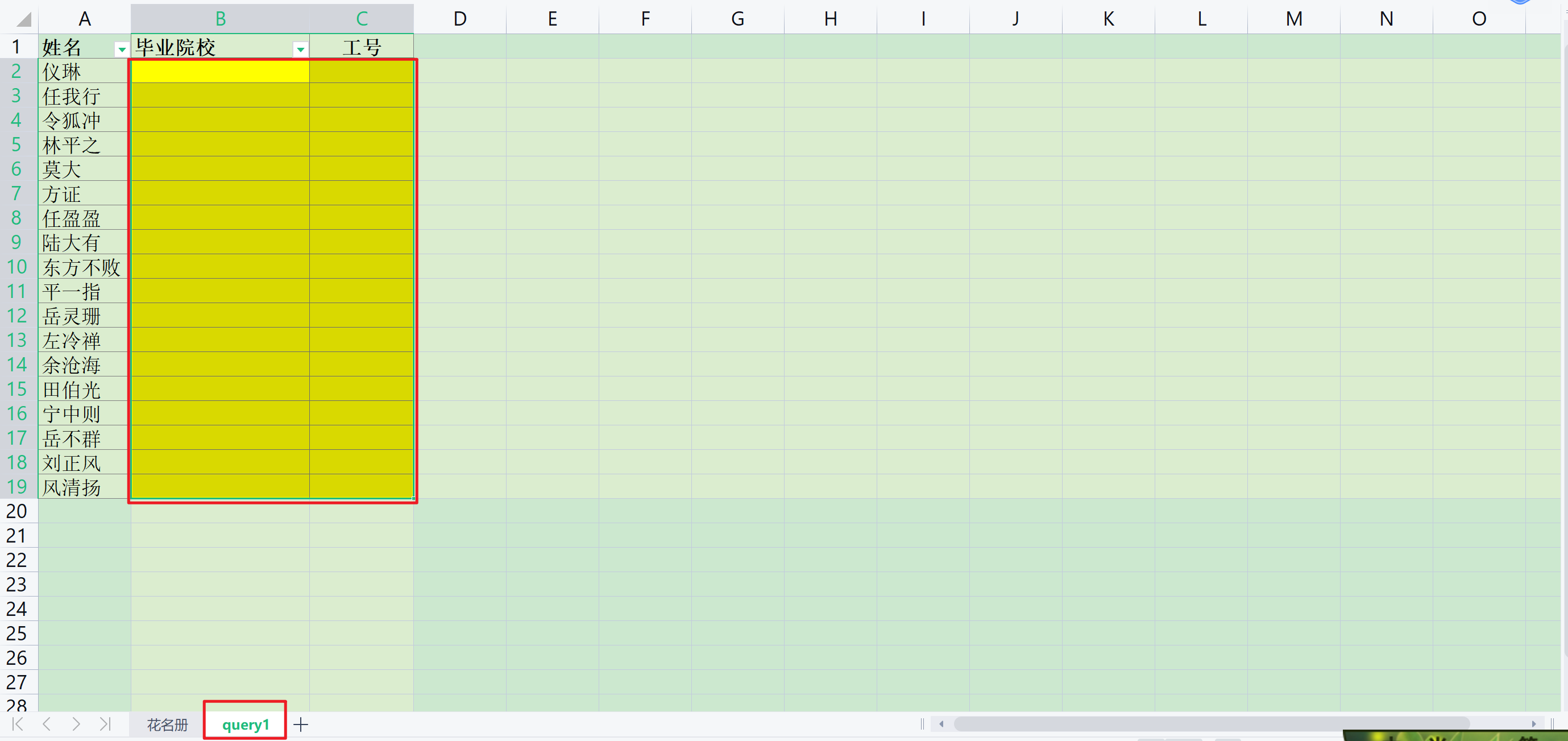
本例中只选了姓名一个条件，假如再加一个条件，把前面第二列的”毕业院校“作为第二个条件，则公式是这样的：

=LOOKUP(1,0/((花名册!B:B=A2)\*(花名册!O:O=B2)),花名册!A:A)

注意第二个参数里面的括号，多条件用乘号\*进行连接，理论上这个条件可以无限套加



二、Python进行实现查询：



下载最新版Anaconda，然后启动Jupyter notebook进行代码编辑和运行：

代码如下：

import pandas as pd

df1 = pd.read\_excel("c:/study\_note/xiao.xlsx",sheet\_name="花名册")

df2 = pd.read\_excel("c:/study\_note/xiao.xlsx",sheet\_name="query1")

df3 = pd.merge(df2['姓名'],df1.loc[:,['姓名','毕业院校','工号']],on='姓名',how='left')

with pd.ExcelWriter("c:/study\_note/xiao.xlsx",mode="a",engine="openpyxl") as writer:

df3.to\_excel(writer,sheet\_name="query2",index=False)

说明：

第一行：导入pandas模块

第二行：读取花名册表到变量df1中，这个路径用的是绝对路径，也就是这个xiao.xlsx工作簿在我的电脑里的位置

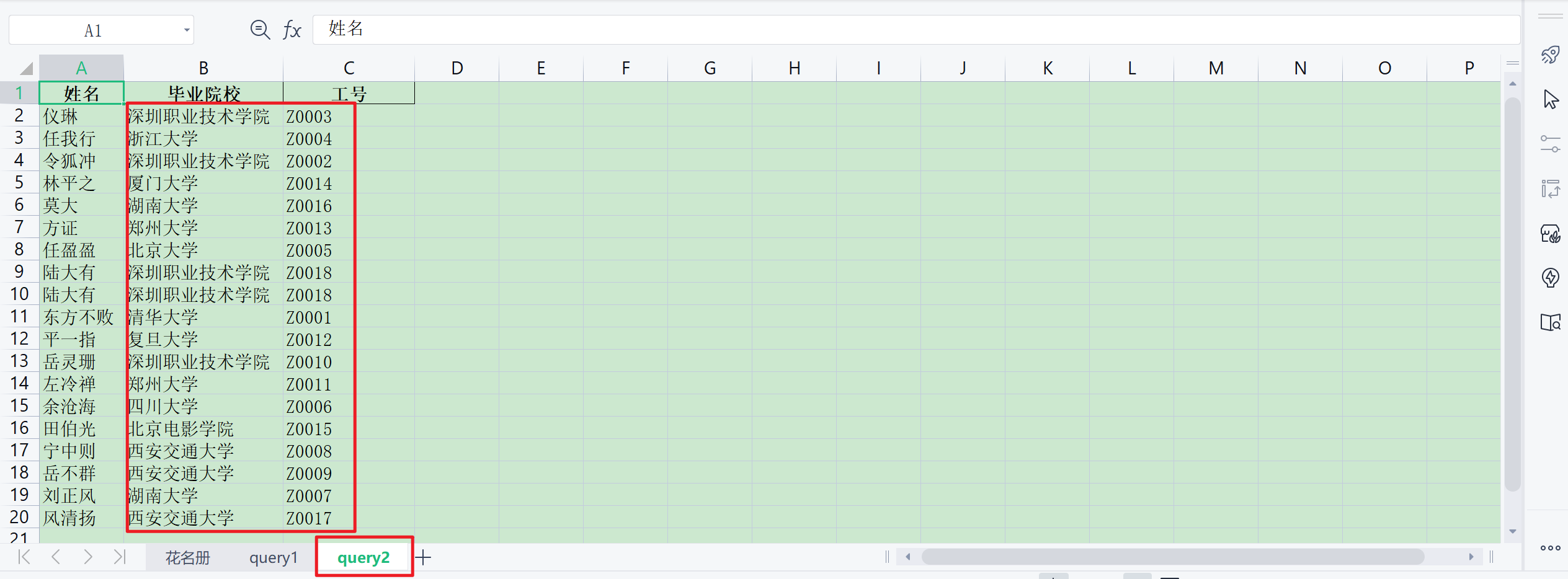
第三行：读取query1表到变量df2中

第四行：这是查询的关键，pandas的merge函数可实现类似EXCEL里面VLOOKUP的功能，我们先选取两个合并区域，从花名册表里只筛选‘姓名，毕业院校，工号’这三列数据’，从query1表只选‘姓名’列数据，然后进行左合并（how='left'），就是以query1表为基准，用‘姓名’列进行查询匹配（on='姓名'），结果保存到变量df3中

1. 六行：把合并的数据写入到工作簿‘xiao.xlsx’里面的‘query2’表中，因为我们要在原文件里面增加一个查询结果的表格，所以不能直接写入，要用ExcelWriter函数，里面要用增加模式mode="a"（a即为add的意思)，引擎要选“openpyxl"，否则会报错然后用写入函数to\_excel把变量df3的内容写到工作簿里面，新表命名为query2，不要索引，否则第一列会变成默认的索引序号



输入代码在jupyter notebook命令行里，然后按SHIFT+回车，即可完成运行，运行后的结果如下：

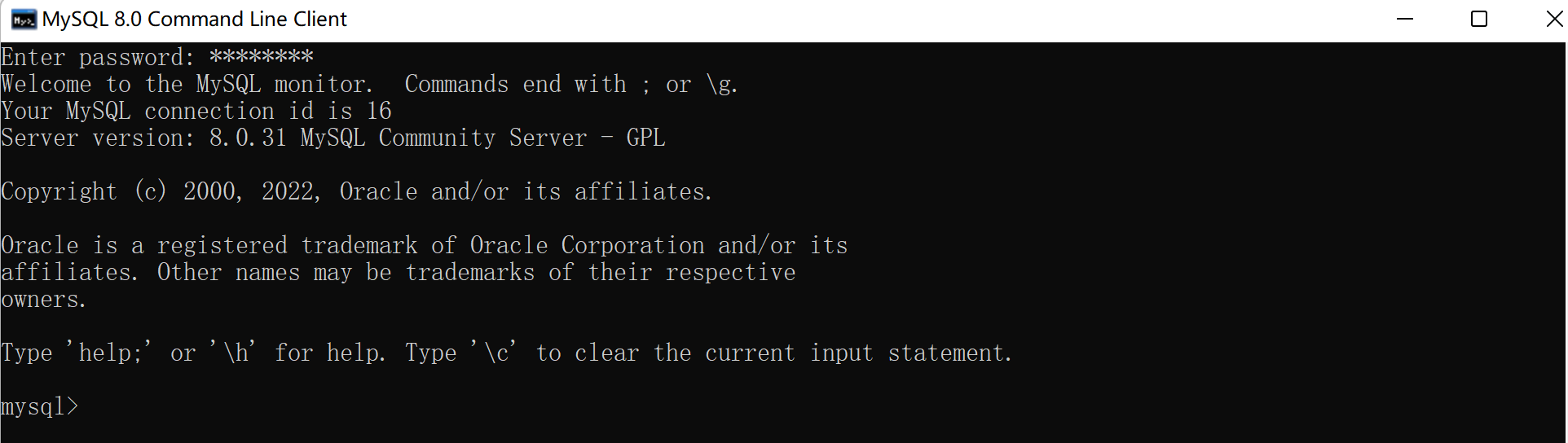


新生成的query2即为我们所要的查询结果，这里我们并没有用Python直接修改query1表的内容，直接修改应该用到类似xlwings这种库，后面我再慢慢研究写教程

到此python已经完成此项任务

三、SQL中实现查询本例：

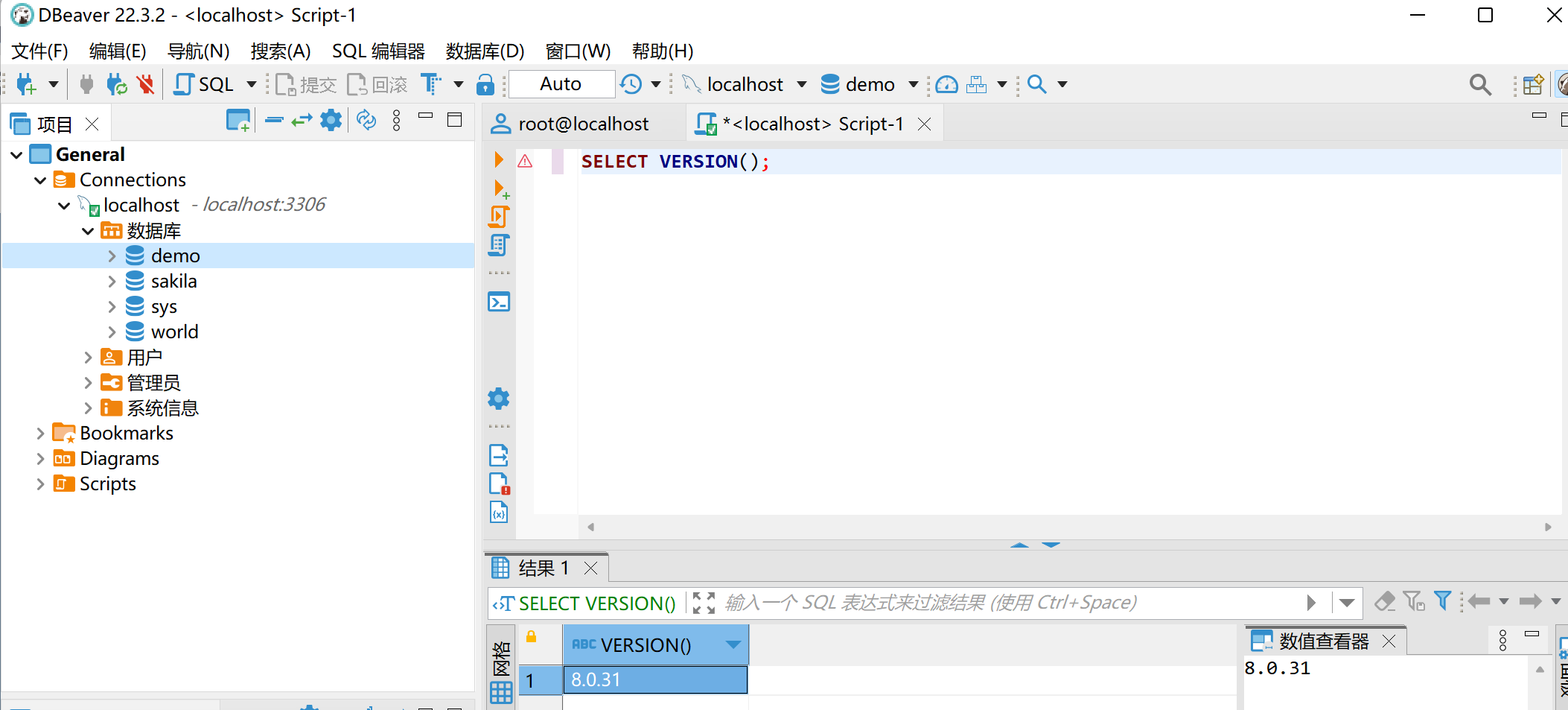
准备工作：本人使用开源数据库软件MySQL进行演示，先在mysql官网下载最新版的mysql软件，windows系统下选installer然后进行图形界面安装即可，安装好后，打开命令行，输入密码，可以看到版本号8.0.31，表示已经安装成功



我们再一个开源的数据库管下载理软件 DBeaver（这个beaver是海狸的意思，LOGO应该就是一只海狸）



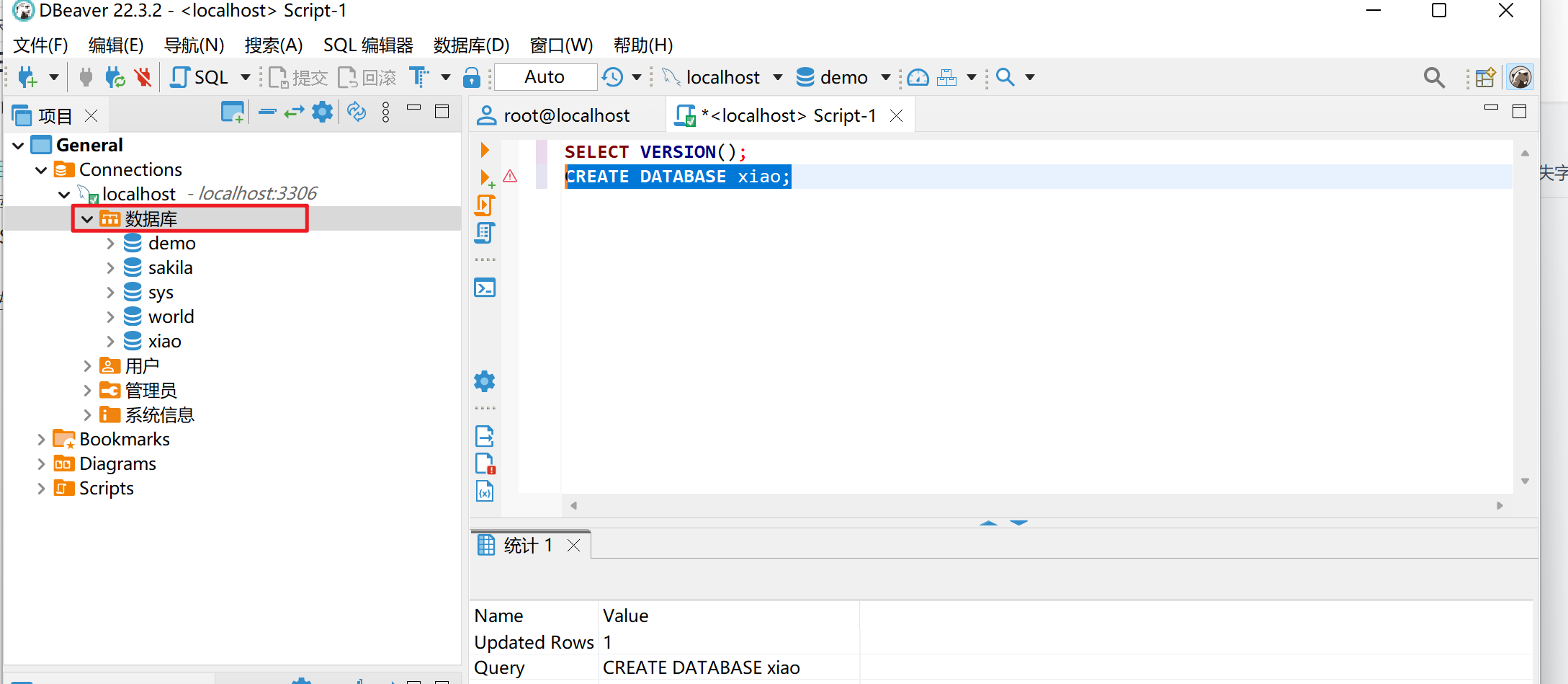
下载普通版就行



连接上我们前面的数据库，可以看到目前版本是 8.0.31

新建数据库xiao，在右侧的SQL编辑器里输入‘CREATE DATABASE xiao;’，按CTRL+回车键即可完成

鼠标选中左边数据库，按F5键刷新，即可看到新建的数据库



选择新建的数据库：USE xiao;



创建一个新表命名为names，对应我们EXCEL里面的花名册表格，然后指定各个字段的数据类型

**代码如下：**

**CREATE** **TABLE** names

(

工号 **CHAR**(5),

姓名 **VARCHAR**(5),

性别 **CHAR**(1),

年龄 **INT**,

技能 **VARCHAR**(10),

其他称呼 **VARCHAR**(8),

部门代码 **CHAR**(5),

部门 **VARCHAR**(4),

岗位代码 **CHAR**(3),

岗位 **VARCHAR**(6),

籍贯 **VARCHAR**(8),

身份证号码 **CHAR**(18),

婚姻状况 **CHAR**(2),

学历 **VARCHAR**(3),

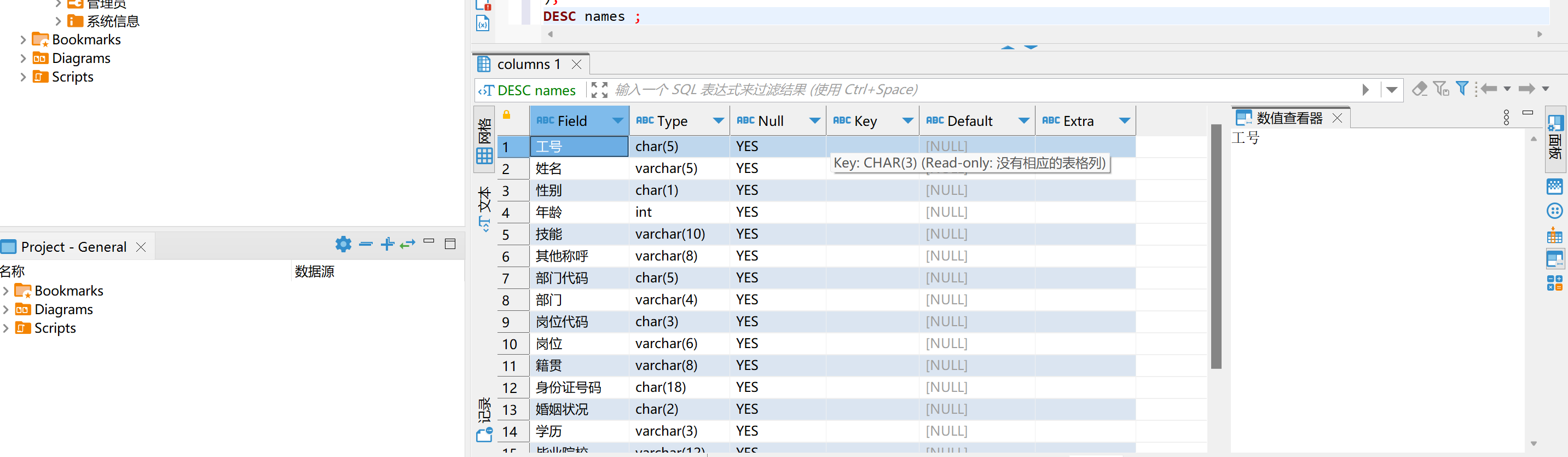
毕业院校 **VARCHAR**(12),

专业 **VARCHAR**(12),

是否最新 **INT**

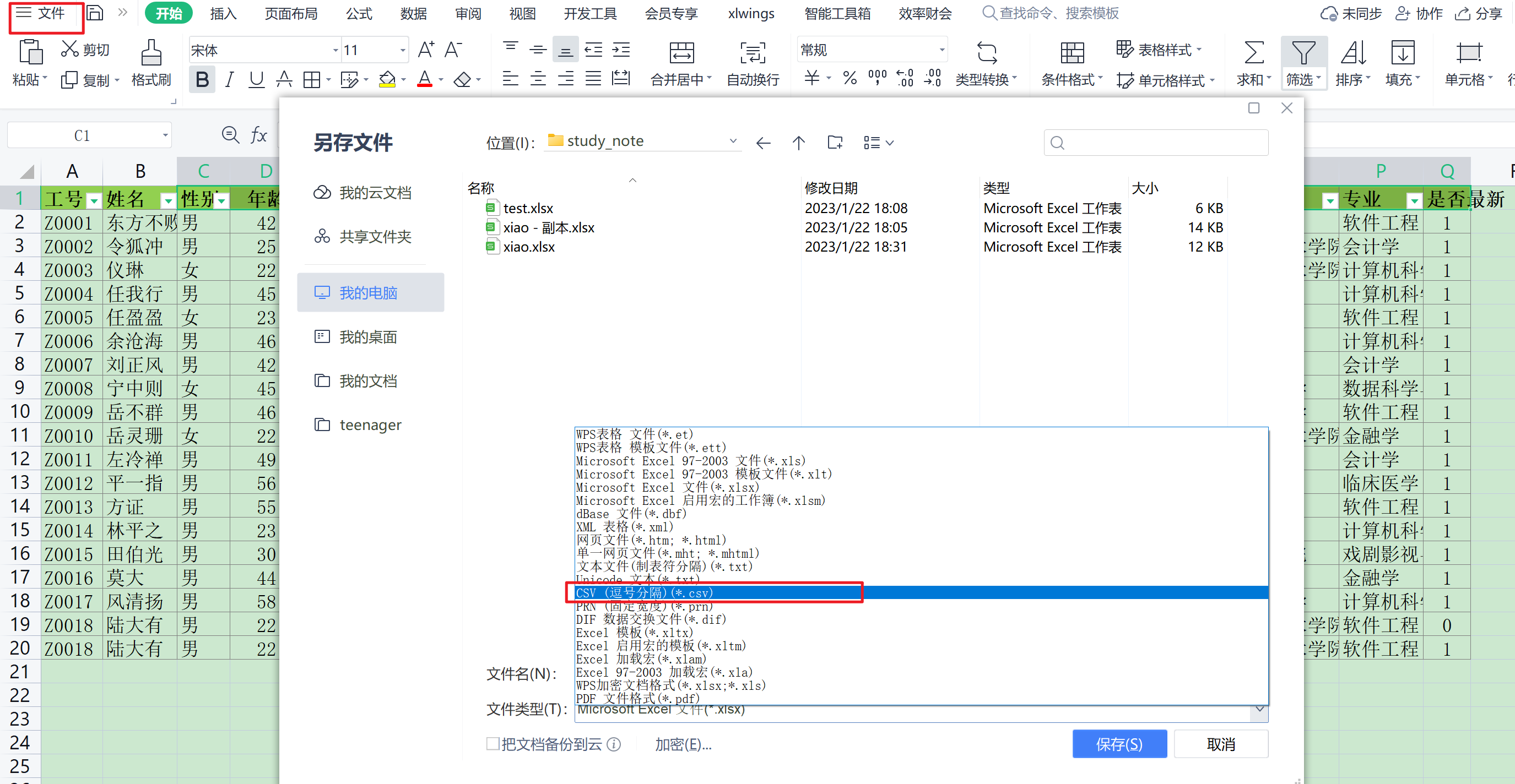
);

然后查看表的字段信息，代码是： DESC names;(也可以写全称 DESCRIBE names;)

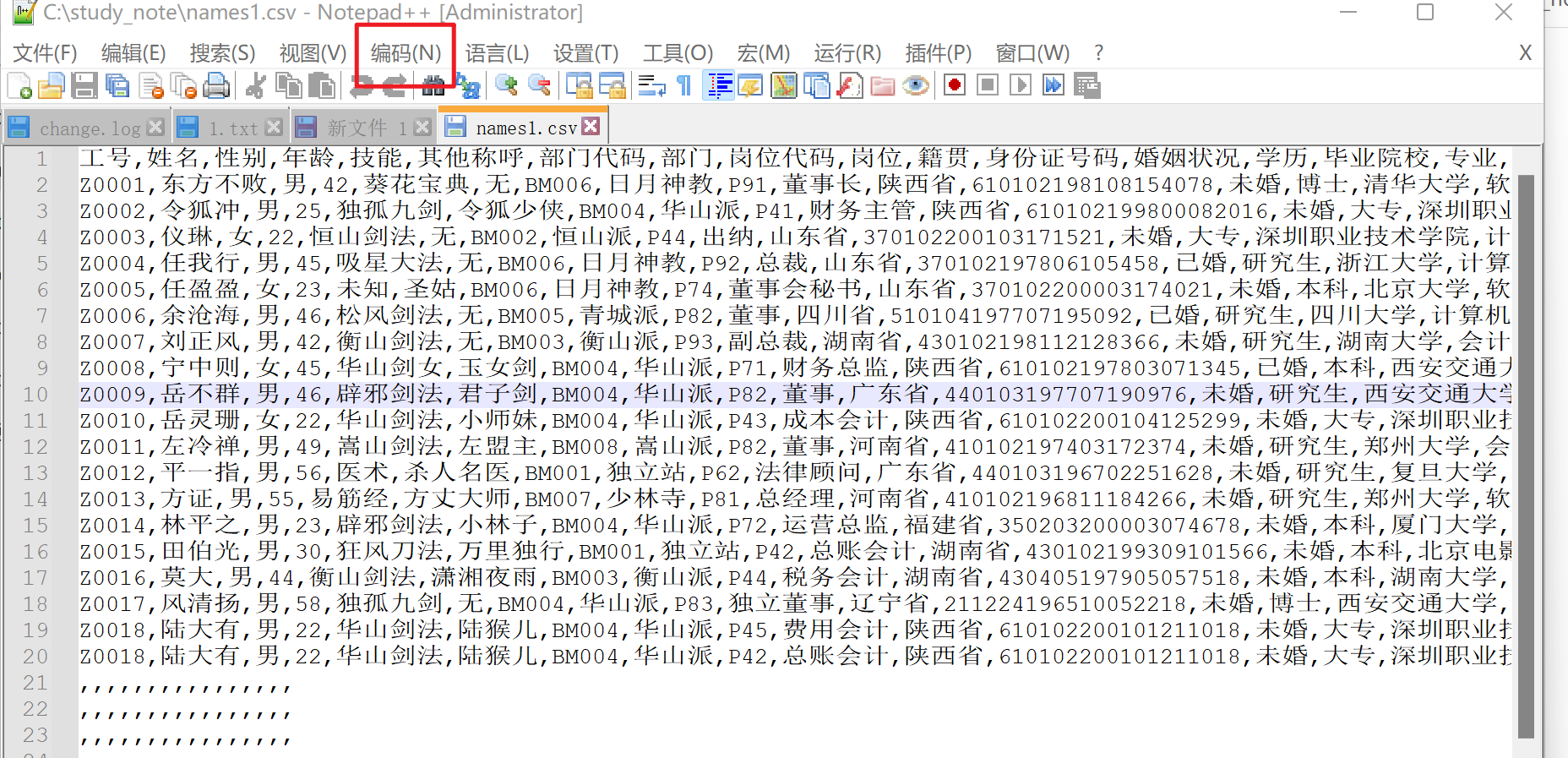


我们先不设置主键，允许为空，把我们的表花名册导入到这个新建的表里面

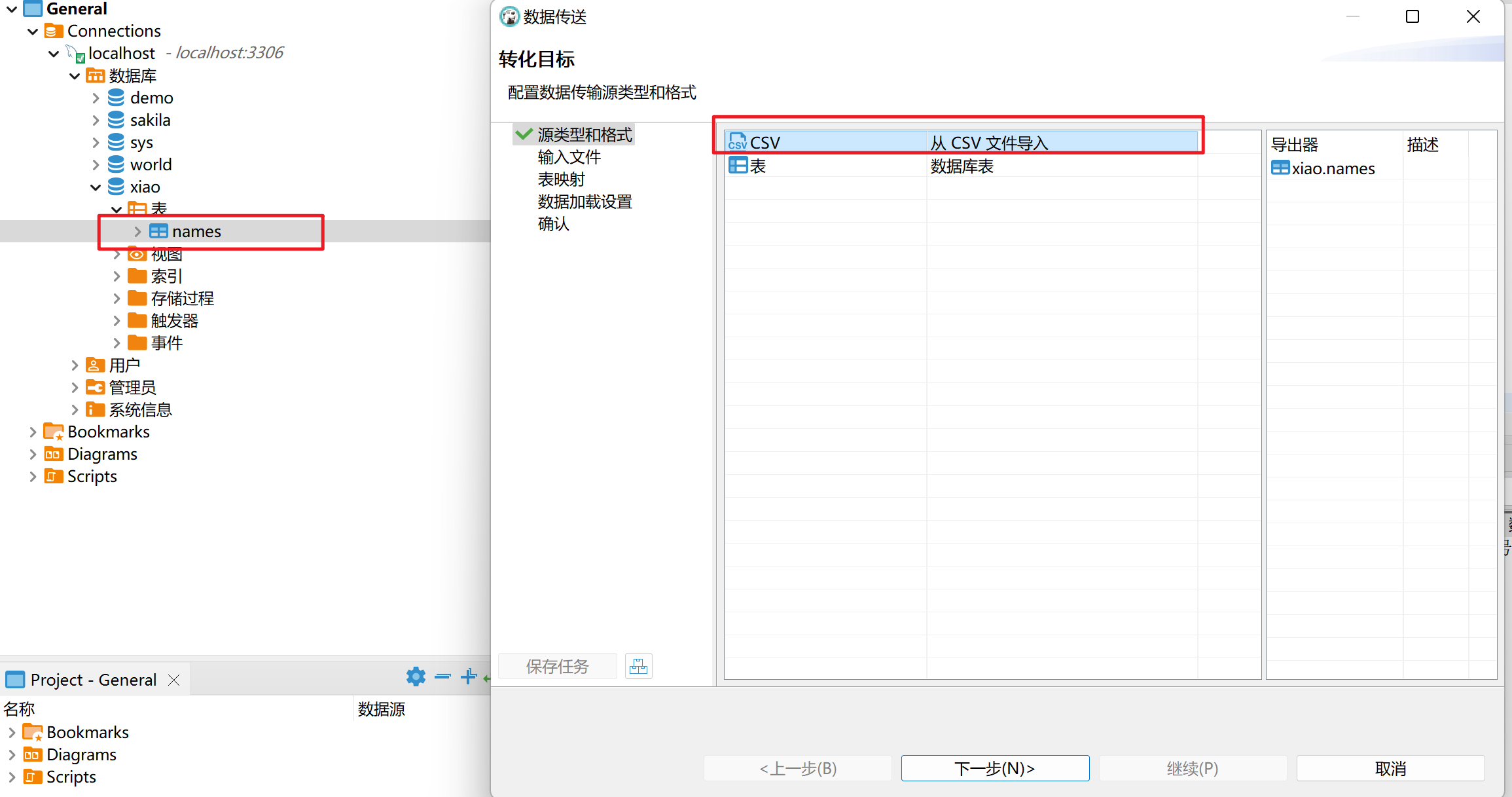
把要导入的花名册工作表先另存为 .csv格式



用文本编辑器把编码改为 UTF-8 格式



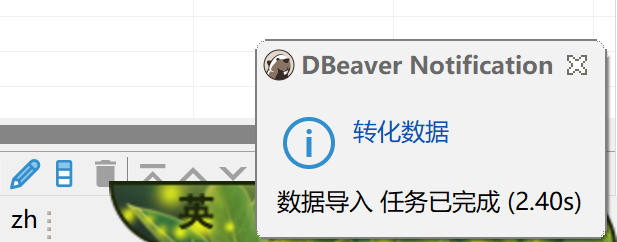
选中names表，鼠标右键点击导入数据：点下一步， 找到我们保存好的CSV文件，一直点下一步就行



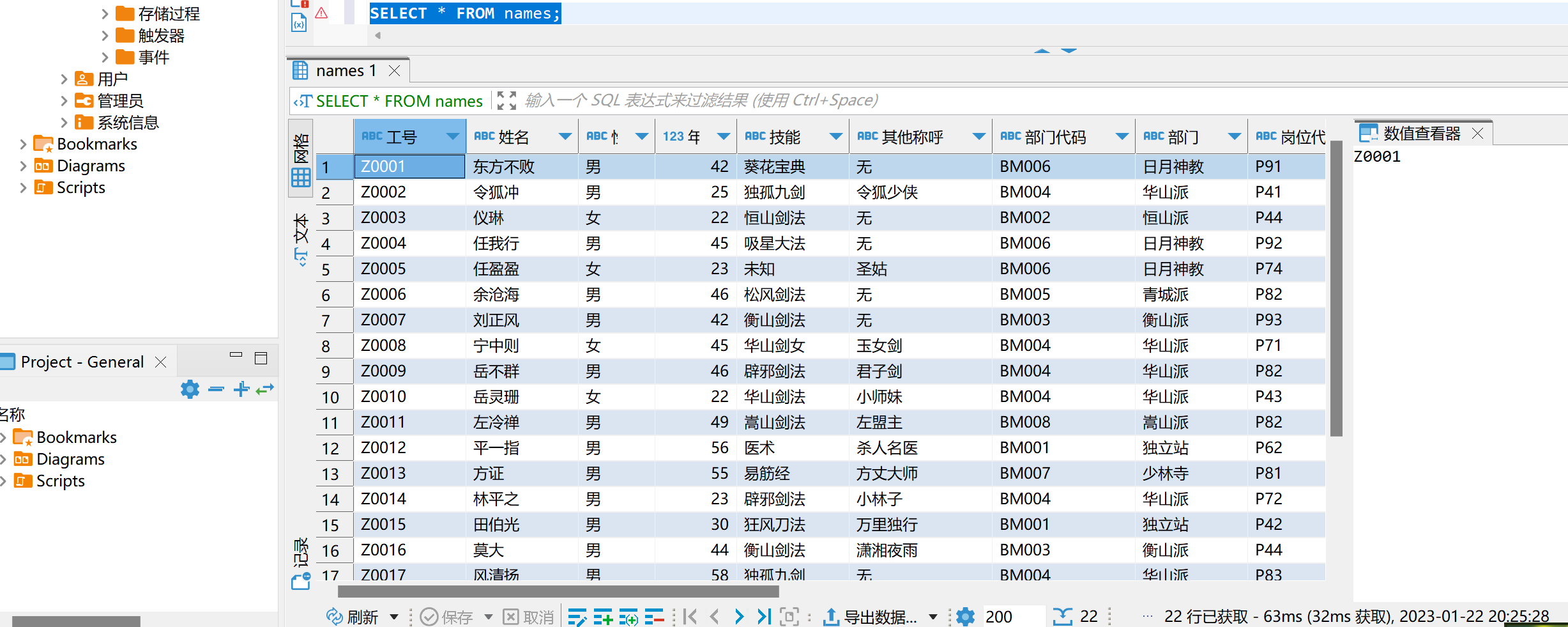
选第一个从CSV文件导入



如果导入成功，在右下角会出现这个提醒界面



然后我们查看一下数据：**SELECT** \* **FROM** names;



我们可以看到数据已经全部导入了数据库，同样的道理，我们也可以用鼠标右键把数据库里面的数据导出来



我们再建另 一张表就是我们要查询的表，命名query1

**CREATE** **TABLE** query1

(

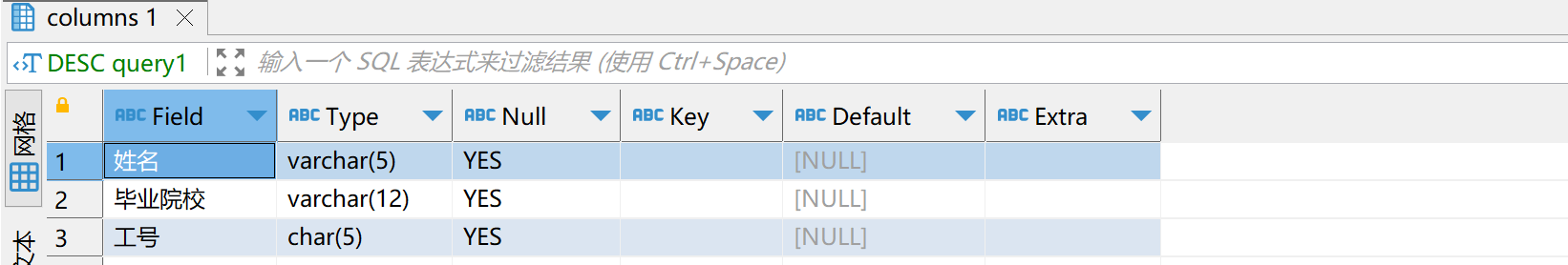
姓名 **VARCHAR**(5),

毕业院校 **VARCHAR**(12),

工号 **CHAR**(5)

);

查看字段信息:

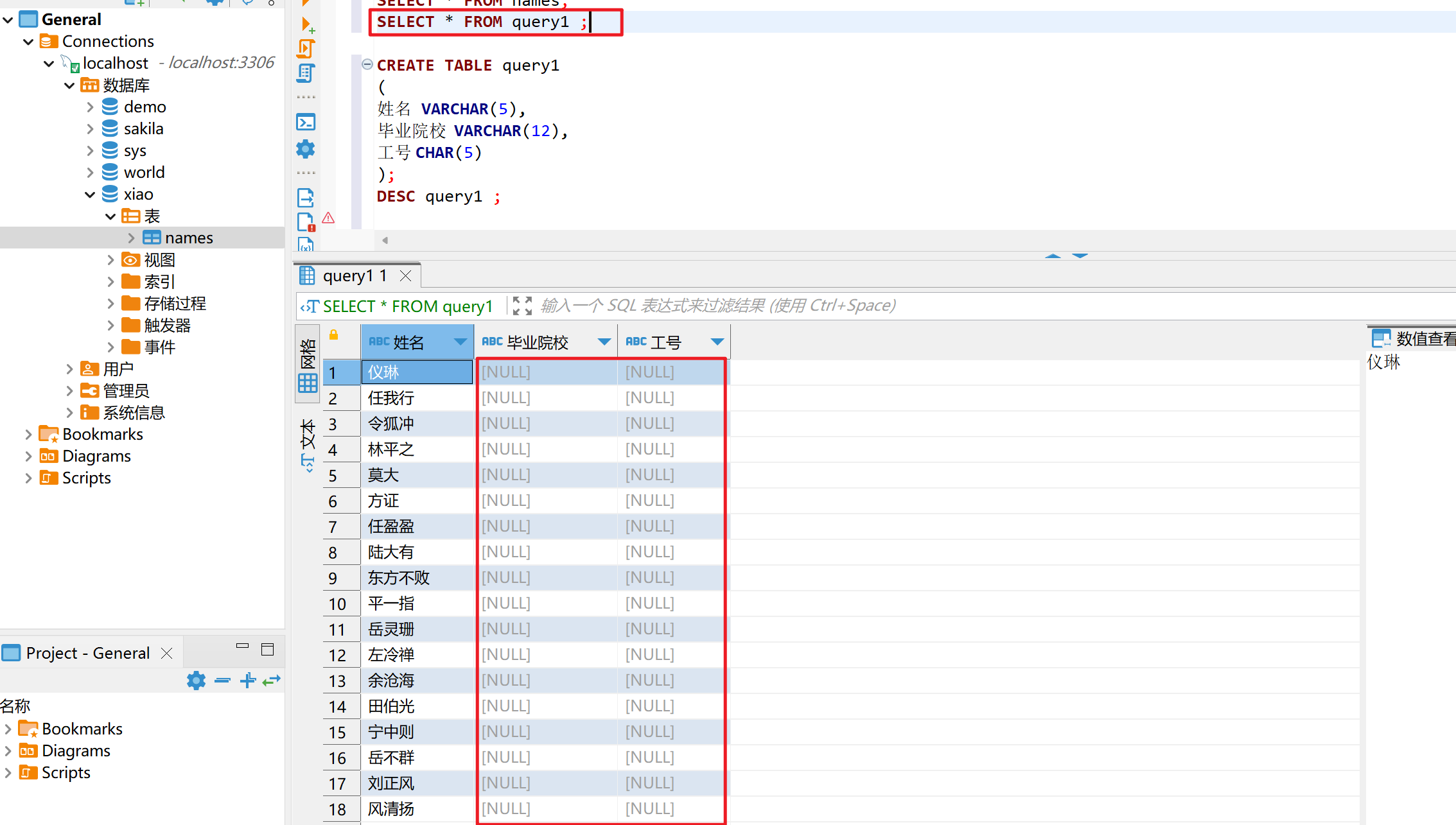


我们把要查找的姓名数据插入到表里面：

**INSERT** **INTO** query1(姓名) **VALUES** ("仪琳"),("任我行"),("令狐冲"),("林平之"),("莫大"),("方证"),("任盈盈"),("陆大有"),("东方不败"),("平一指"),("岳灵珊"),("左冷禅"),("余沧海"),("田伯光"),("宁中则"),("岳不群"),("刘正风"),("风清扬");

这里我们用了代码插入，因为只有一个字段，也是可以按照前面的引入方法，把query1里第一列数据引到表里面来一样

我们查看一下query1表：

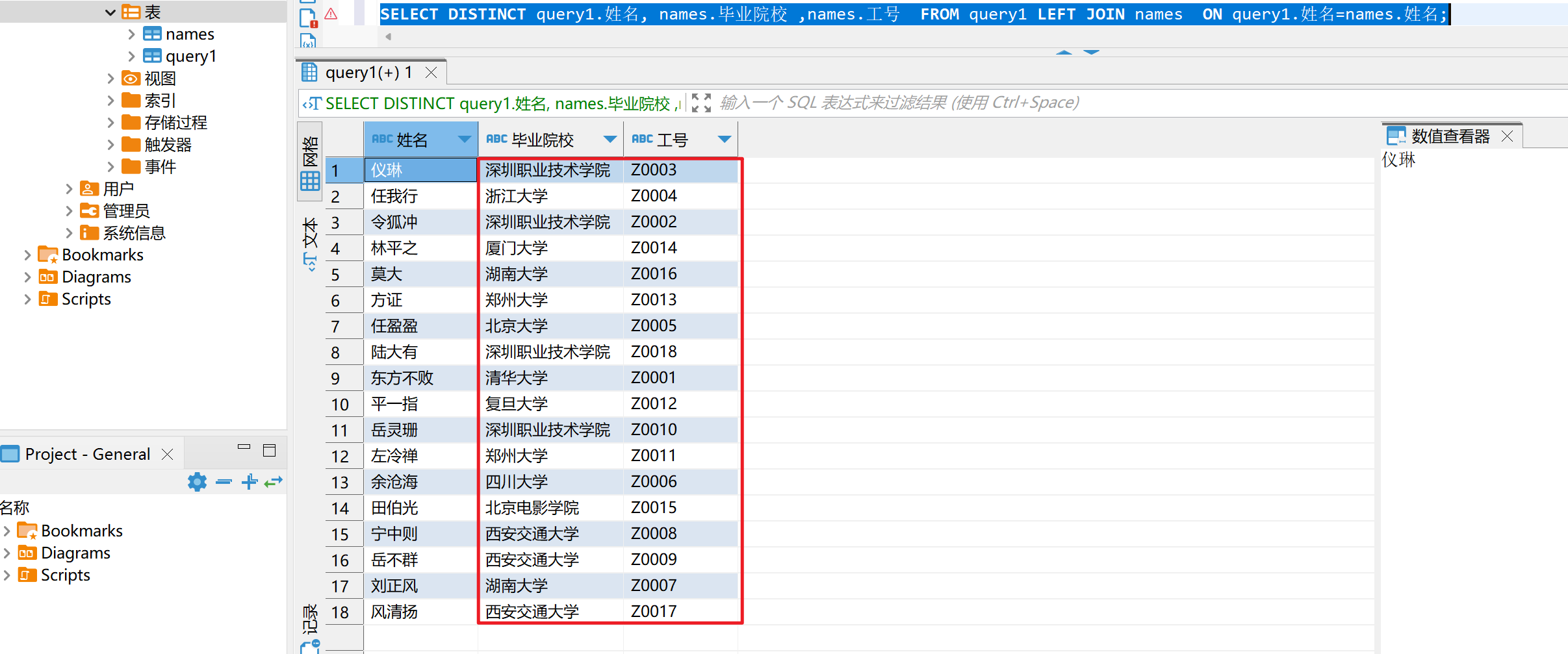


很明显，毕业院校和工号里面的数据是空的，正是本例中我们要查询的，最终就是把查到的结果放到这两列里面

首先建立查询代码：

SELECT DISTINCT query1.姓名, names.毕业院校 ,names.工号 FROM query1 LEFT JOIN names ON query1.姓名=names.姓名;

运行结果如下：红色框里的数据即是我们想要的，只需要把这个数据值插入到表query1中即可



将查询结果保存到表query1中的代码：

**INSERT** **INTO** query1 (姓名,毕业院校,工号) **SELECT** **DISTINCT** query1.姓名, names.毕业院校 ,names.工号 **FROM** query1 **LEFT** **JOIN** names **ON** query1.姓名=names.姓名;

这种方法的结果是在表的下面又插入了相同的行数，如下：

前面INSERT INTO 表名（列名） 这个是标准语句，表示插入新行，在哪个表的哪些列

**SELECT** **DISTINCT** query1.姓名, names.毕业院校 ,names.工号 **FROM** query1 **LEFT** **JOIN** names **ON** query1.姓名=names.姓名;

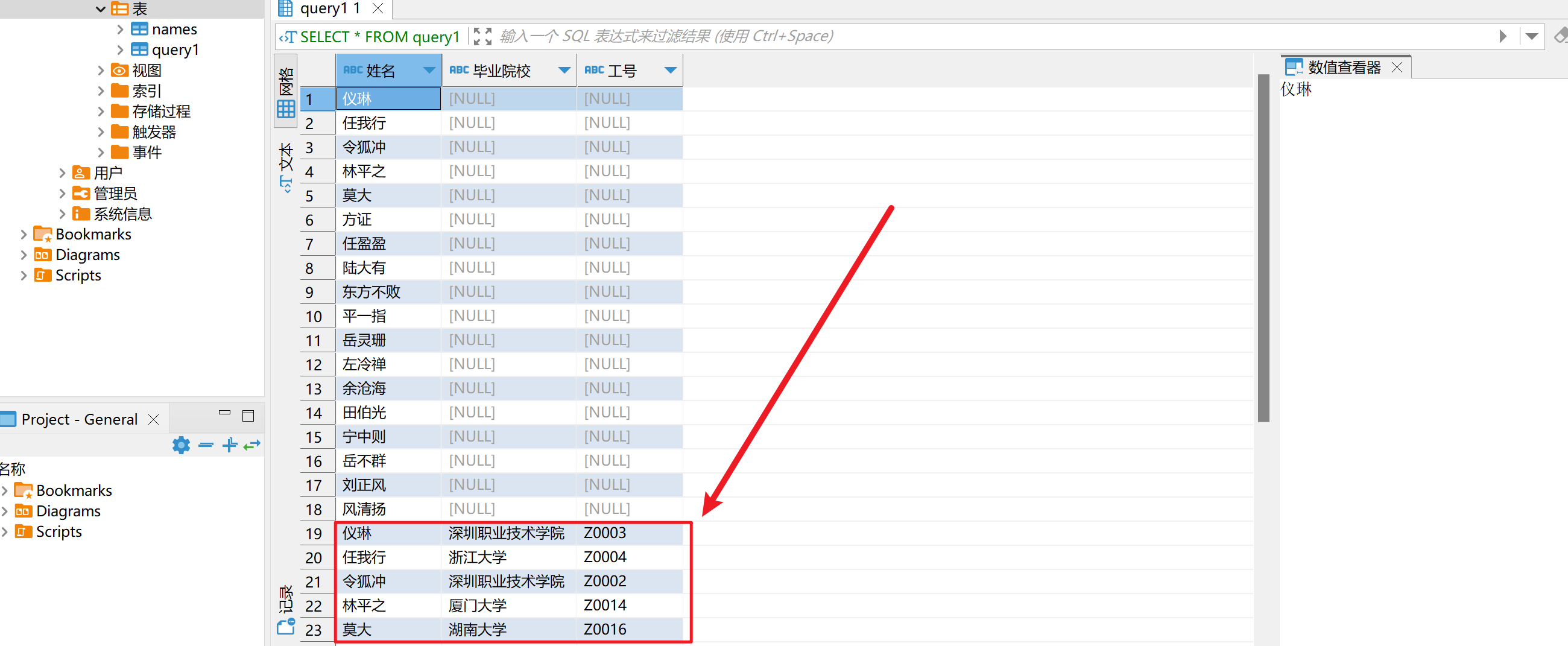
这是一个外连接查询语句，LEFT JOIN 是左连接的意思，有点类似PYTHON里面merge函数里的how=left那个参数

（query1 **LEFT** **JOIN** names **ON** query1.姓名=names.姓名）这是相连的表的区域

前面标准的SELECT 查询语句，查找三个列的值，因为有两张表，所以要加表名.字段的形式

DISTINCT代表去除重复项，因为在names表里面陆大有这个名字有两条数据

然后把查询到的值赋给前面的INSERT INTO语名

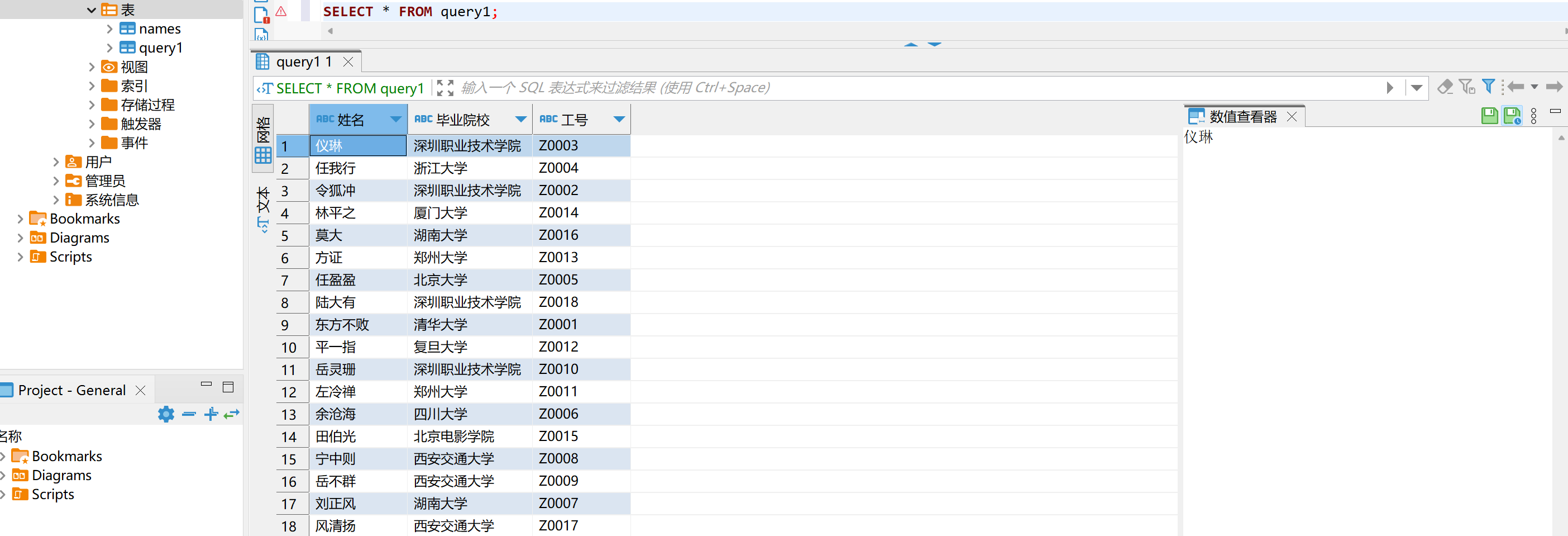


这时我们用一个取巧的办法，把上面的行删掉，即可得到我们想要的结果

**DELETE** **FROM** query1 **LIMIT** 18;

删除query1表前面18行的意思

最终结果如下：



如果直接把值插入到query1里面右边的空值里，则用这个代码：

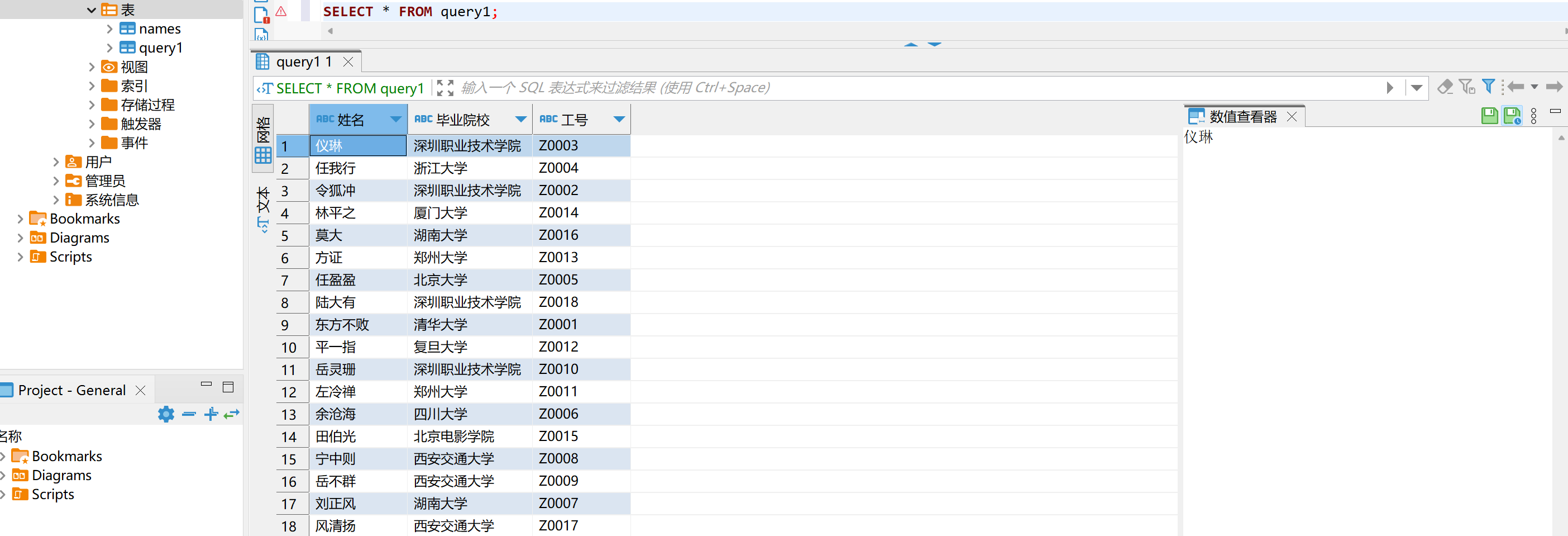
**UPDATE** query1,names **SET** query1.毕业院校 = names.毕业院校 , query1.工号 = names.工号 **WHERE** query1.姓名 =names.姓名 ;

这是一个更新语句，UPDATE +表名 表示更新哪些表

SET后面是更新具体字段的值，我们需要从names表里面找到对应列的值然后赋给query1表里面的字段

WHERE后面是条件，相当于merge里面的on=’’,也相当于SQL左连接里面的ON=‘’

运行后同样得到我们想要的结果



三种方法最终结果：解决同一个实际问题，算是殊途同归吧

个人建议：如果数据量不大则EXCEL完全可以驾驭，如果数据量超过100万行，则考虑用后两种办法实现更快一些

希望朋友多提意见，同时有EXCEL表格领域相关实际问题可以私信发给我，有时间会慢慢研究并和大家一起进步。