西安交通大学实验报告

课程 **Python数据处理** 实验名称 **使用pandas进行数据预处理**  共 **31** 页

系 别 **电信学部** 实 验 日 期 **2023**年**4**月**15**日

专业班级 实 验 报 告 日 期 **2023**年**4**月**15**日

姓 名 学号 报 告 退 发( 订正 、 重做 )

同 组 人\_\_\_无\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 教 师 审 批 签 字

*本次实验合计约使用****3****个小时*

# **实验目的**

1. **熟练掌握pandas库的使用，包括数据清洗、操作、分析和可视化等方面，以便快速处理大量和复杂的数据表。**
2. **了解数据库应用的基本概念、操作和实现方法，能够使用工具如SQLAlchemy连接和查询不同类型的数据库。**
3. **掌握尝试各种方法相融合的能力，通过在处理数据时灵活地使用pandas库和数据库应用，编写出更加快捷、高效和可重用性的程序。**
4. **提高自己的实践能力和思维方式。**

# **实验内容**

## 实验题目一：

**从思源学堂中下载文件“简要教学数据.xlsx”，编写python程序创建数据库jgdb1.db，运用pandas的函数导入简要教学数据.xlsx中的三张表格，第一张表格是学生表STUDENT，该表包括学号Sno，姓名Sname，性别Ssex，年龄Sage和学院Sdept字段；第二张表格是课程表COURSE，该表包括课程号Cno，课程名称Cname和学时Hours字段；第三张表格是选课表SC，该表包括学号Sno，课程号Cno和成绩Grade字段。将这三张表格形成数据库中的表格数据。运用函数pd.read\_sql()完成下列查询：**

**1.查询年龄等于20的男生记录（即学号Sno，姓名Sname，性别Ssex，年龄Sage和学院Sdept）**

**2.查询成绩在70到80之间的学生的学号、课程号和成绩**

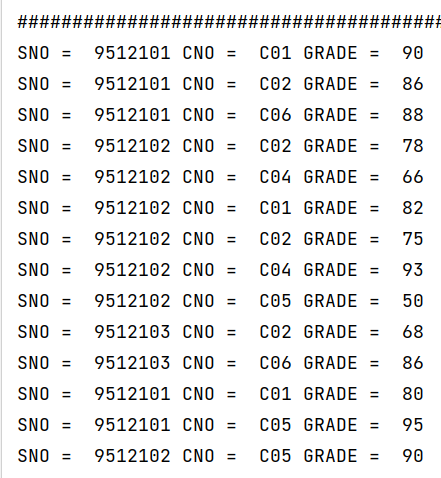
**3.将学生表保存到excel文件中，并提交该excel文件到思源学堂**

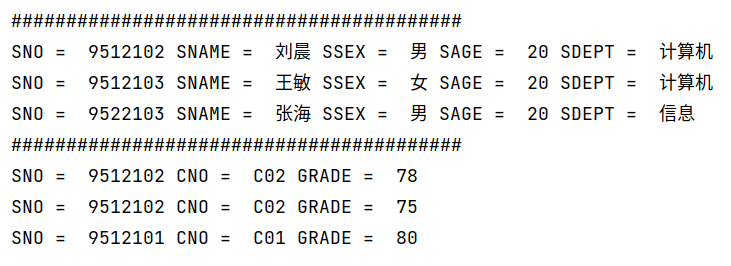
**4.将选课表保存到csv文件中，并提交该文本文件**

#### 程序源代码：

**import sqlite3  
import pandas as pd  
from sqlalchemy import create\_engine  
  
engine = create\_engine('sqlite:///jgdb1.db')  
conn = sqlite3.connect('jgdb1.db')  
c = conn.cursor()  
  
c.execute('''CREATE TABLE STUDENT (SNO INT PRIMARY KEY NOT NULL,SNAME TEXT NOT NULL,SSEX TEXT NOT NULL,SAGE INT NOT NULL,SDEPT CHAR(50)); ''')  
  
c.executemany("INSERT INTO STUDENT (SNO,SNAME,SSEX,SAGE,SDEPT) VALUES (?, ?, ?, ?, ?)", [  
 (9512101,'李勇','男',19,'计算机'),  
 (9512102,'刘晨','男',20,'计算机'),  
 (9512103,'王敏','女',20,'计算机'),  
 (9522101,'张立','男',22,'信息'),  
 (9522102,'吴宾','女',21,'信息'),  
 (9522103,'张海','男',20,'信息'),  
 (9532101,'钱小力','女',18,'数学'),  
 (9532102,'王大力','男',19,'数学')  
])  
conn.commit()  
  
c.execute('''CREATE TABLE COURSE (CNO CHAR(50) PRIMARY KEY NOT NULL,CNAME TEXT NOT NULL,HOURS INT NOT NULL); ''')  
c.executemany("INSERT INTO COURSE (CNO,CNAME,HOURS) VALUES (?, ?, ?)", [  
 ('C01','计算机文化学',70),  
 ('C02','VB',90),  
 ('C03','计算机网络',80),  
 ('C04','数据库基础',108),  
 ('C05','高等数学',180),  
 ('C06','数据结构',72)  
])  
conn.commit()  
  
c.execute('''CREATE TABLE SC (SNO INT NOT NULL,CNO CHAR(50) NOT NULL,GRADE INT NOT NULL); ''')  
c.executemany("INSERT INTO SC (SNO,CNO,GRADE) VALUES (?, ?, ?)", [  
 (9512101,'C01',90),  
 (9512101,'C02',86),  
 (9512101,'C06',88),  
 (9512102,'C02',78),  
 (9512102,'C04',66),  
 (9512102,'C01',82),  
 (9512102,'C02',75),  
 (9512102,'C04',93),  
 (9512102,'C05',50),  
 (9512103,'C02',68),  
 (9512103,'C06',86),  
 (9512101,'C01',80),  
 (9512101,'C05',95),  
 (9512102,'C05',90)  
])  
conn.commit()  
  
result = c.execute("SELECT sno, sname, ssex, sage, sdept FROM STUDENT WHERE sage = 20")  
for row in result:  
 print("SNO = ", row[0], end=' ')  
 print("SNAME = ", row[1], end=' ')  
 print("SSEX = ", row[2], end=' ')  
 print("SAGE = ", row[3], end=' ')  
 print("SDEPT = ", row[4])  
  
result2 = c.execute("SELECT sno, cno, grade FROM SC WHERE grade >= 70 AND grade <= 80")  
for row in result2:  
 print("SNO = ", row[0], end=' ')  
 print("CNO = ", row[1], end=' ')  
 print("GRADE = ", row[2])  
  
detail = pd.read\_sql\_table('STUDENT', con=engine)  
detail.to\_excel(r'student.xlsx', index=False)  
  
detail1 = pd.read\_sql\_table('SC', con=engine)  
detail1.to\_csv(r'sc.csv', index=False, encoding='gbk')**

#### 运行结果：





## 实验题目二：

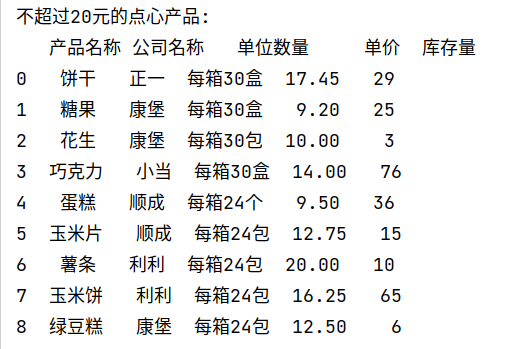
**从思源学堂中下载文件“商品集.xlsx”，编写python程序创建数据库jgdb2.db，运用pandas的函数导入商品集.xlsx，形成数据库中的表格数据。运用函数pd.read\_sql()完成下列查询：(1)查询单价在10元至35元之间的产品；(2)查询单价不超过20元的“点心”类的产品，只要求列出产品名称、供应商、单位数量、单价、库存量。(3)将第1个查询结果保存到excel文件中，提交该excel文件到思源学堂中；(4) 将第2个查询结果保存到csv文件中，提交该文本文件到思源学堂中。**

#### 程序源代码

**import pandas as pd  
from sqlalchemy import create\_engine  
  
# 创建数据库引擎  
engine = create\_engine('sqlite:///jgdb2.db')  
  
# 读取商品信息并写入到数据库中  
asd = pd.read\_excel(r'商品集.xlsx')  
asd.to\_sql('business', con=engine, index=False, if\_exists='replace')  
  
# 查询单价在10~35元之间的商品信息  
detail = pd.read\_sql\_query(  
 'SELECT \* FROM business WHERE 单价 BETWEEN 10 AND 35',  
 engine  
)  
print('单价为10~35元:\n', detail)  
  
# 查询所有单价不超过20元且类别为"点心"的商品信息  
detail1 = pd.read\_sql\_query(  
 'SELECT 产品名称, 公司名称, 单位数量, 单价, 库存量 FROM business WHERE 单价 <= 20 AND 类别名称 = "点心"',  
 engine  
)  
print('不超过20元的点心产品:\n', detail1)  
  
# 将查询结果写入到 Excel 和 CSV 文件中  
detail.to\_excel(r'单价在10元至35元之间的产品.xlsx', index=False)  
detail1.to\_csv(r'单价不超过20元的“点心”类的产品.csv', index=False, encoding='gbk')**

#### 运行结果：





## 实验题目事先说明

**使用Pandas完成以下实验题目三、实验题目四、实验题目五、实验题目六，数据集在思源学堂中，具体csv文件说明如下：**

**Training\_Master.csv：该文件中各字段含义为：用户唯一标识Idx，用户基本信息UserInfo\_i，用户学历信息Education\_Info\_i，用户网页登录信息WeblogInfo\_i，用户社交网络信息SocalNetwork\_i，用户第三方平台信息ThirdParty\_Info\_Period\_i\_j，用户是否逾期target，借款成交时间ListingInfo**

**Training\_Userupdate.csv：该文件中各字段含义为： 用户唯一标识Idx， 借款成交时间ListingInfo，用户更新信息内容UserUpdateInfo\_1，用户更新信息时间UserUpdateInfo\_2**

**Training\_LogInfo.csv：该文件中各字段含义为： 用户唯一标识Idx， 借款成交时间ListingInfo1，用户登录操作代码LogInfo1，用户登录操作类别LogInfo2，用户登录时间LogInfo3**

## 实验题目三：

**1）读取数据集中P2P网络贷款数据主表（Training\_Master.csv）**

**2）使用ndim、shape、memory\_usage属性分别查看该文件的维度、大小、占用内存等信息**

**3）使用describe方法对文件进行描述性统计**

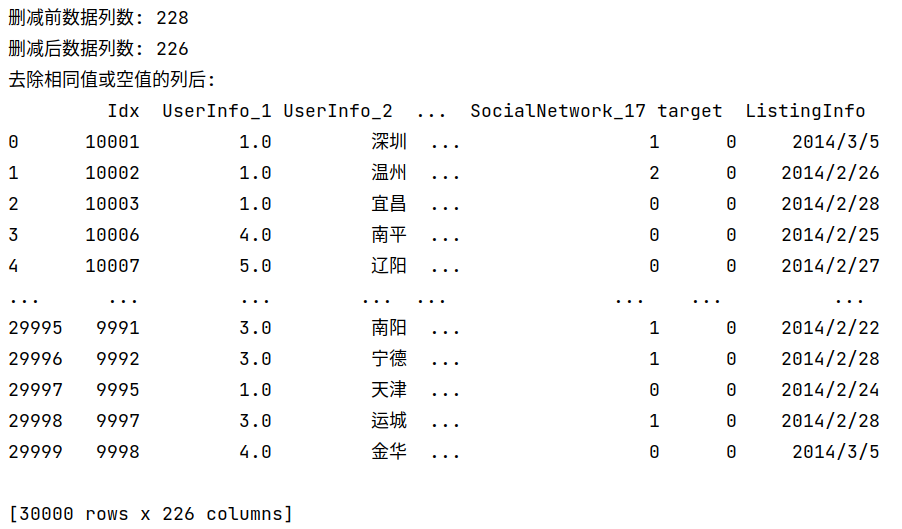
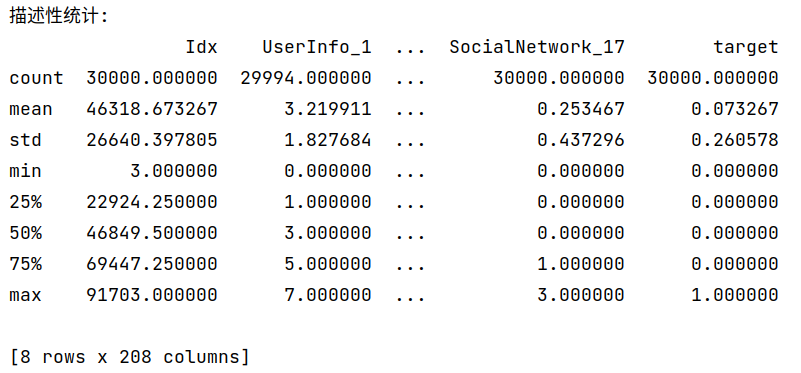
**4）删除值相同或全为空的列**

#### 程序源代码：

**import pandas as pd  
  
# 读取 CSV 文件并打印文件的基本信息  
detail = pd.read\_csv('Training\_Master.csv', encoding='gbk')  
print(f'文件维度: {detail.ndim}')  
print(f'文件大小: {detail.shape}')  
print('文件占用内存:')  
print(detail.memory\_usage())  
  
# 打印文件的描述性统计信息  
print('描述性统计:')  
print(detail.describe())  
  
# 定义函数用于去除空值或标准差为零的列  
def drop\_null\_std(data):  
 before\_len = data.shape[1]  
 col\_is\_null = data.isnull().sum() == data.shape[0]  
 std\_is\_zero = data.describe().loc['std'] == 0  
  
 data.drop(col\_is\_null.index[col\_is\_null], axis=1, inplace=True)  
 data.drop(std\_is\_zero.index[std\_is\_zero], axis=1, inplace=True)  
  
 after\_len = data.shape[1]  
 print(f'删减前数据列数: {before\_len}')  
 print(f'删减后数据列数: {after\_len}')  
  
# 去除空值或标准差为零的列并打印结果  
drop\_null\_std(detail)  
print('去除相同值或空值的列后:')  
print(detail)**

#### 运行结果：





## 实验题目四：

**1) 读取数据集中用户信息更新表（Training\_Userupdate.csv）和用户登录信息表（Training\_LogInfo.csv）**

**2）使用to\_datetime函数将两个表中的时间转换成标准时间**

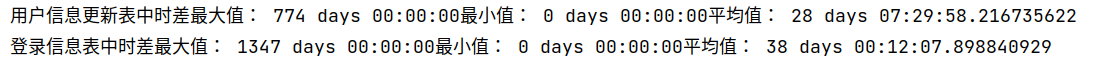
**3）计算用户信息更新表和登录信息表中两个时间的差值**

**4）给出差值的最大值、最小值和平均值**

#### 程序源代码

**import pandas as pd  
  
detail = pd.read\_csv('Training\_Userupdate.csv', encoding='gbk')  
detail1 = pd.read\_csv('Training\_LogInfo.csv', encoding='gbk')  
detail['ListingInfo1'] = pd.to\_datetime(detail['ListingInfo1'])  
detail['UserupdateInfo2'] = pd.to\_datetime(detail['UserupdateInfo2'])  
detail1['Listinginfo1'] = pd.to\_datetime(detail1['Listinginfo1'])  
detail1['LogInfo3'] = pd.to\_datetime(detail1['LogInfo3'])  
timedelta = detail['ListingInfo1'] - detail['UserupdateInfo2']  
timedelta1 = detail1['Listinginfo1'] - detail1['LogInfo3']  
print('用户信息更新表中时差最大值：', timedelta.max(), end='')  
print('最小值：', timedelta.min(), end='')  
print('平均值：', timedelta.mean())  
print('登录信息表中时差最大值：', timedelta1.max(), end='')  
print('最小值：', timedelta1.min(), end='')  
print('平均值：', timedelta1.mean())**

#### 运行结果：



## 实验题目五：

**1）读取数据集中用户信息更新表（Training\_Userupdate.csv）和用户登录信息表（Training\_LogInfo.csv）**

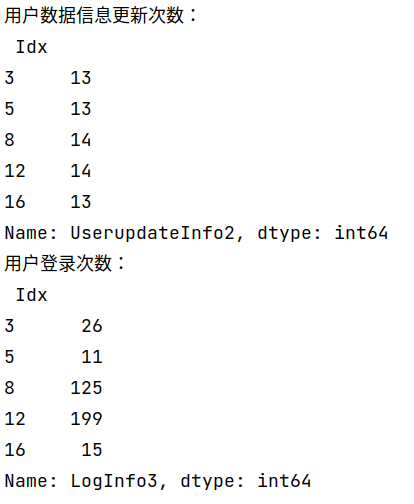
**2）使用groupby方法按照用户标识号Idx对用户信息更新表和登录信息表分组**

**3）计算分组后每个用户的数据信息更新次数和登录次数，只列出头5条记录**

#### 程序源代码

**import pandas as pd  
  
detail = pd.read\_csv('Training\_Userupdate.csv', encoding='gbk')  
detail1 = pd.read\_csv('Training\_LogInfo.csv', encoding='gbk')  
detailg = detail.groupby(by='Idx')  
detailg1 = detail1.groupby(by='Idx')  
print('用户数据信息更新次数：\n', detailg['UserupdateInfo2'].count().head())  
print('用户登录次数：\n', detailg1['LogInfo3'].count().head())**

#### 运行结果：



## 实验题目六：

**1）读取数据集中用户信息更新表（Training\_Userupdate.csv）和用户登录信息表（Training\_LogInfo.csv）**

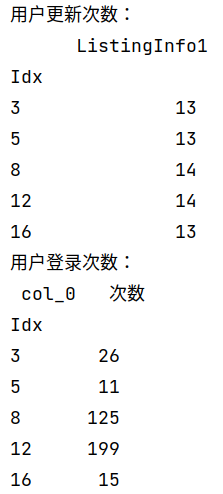
**2）使用pivot\_table函数按照用户标识号Idx统计用户信息更新次数，只列出头5条**

**3）使用crosstab函数按照用户标识号Idx统计用户登录次数，只列出头5条**

#### 程序源代码

**import pandas as pd  
import numpy as np  
  
update = pd.read\_csv(r'Training\_Userupdate.csv', encoding='utf-8')  
login = pd.read\_csv(r'Training\_LogInfo.csv', encoding='utf-8')  
pivot = pd.pivot\_table(update[['Idx', 'ListingInfo1']], index='Idx', aggfunc=len).head()  
crosstab = pd.crosstab(index=login['Idx'], columns='次数', values=login['Idx'], aggfunc=len).head()  
print('用户更新次数：\n', pivot)  
print('用户登录次数：\n', crosstab)**

#### 运行结果：



## 实验题目七：

**编写程序将实验题目一创建数据库jgdb1.db，运用pandas函数完成下列查询或统计：**

**1.输出C01课程成绩最高的分数**

**2.统计学生选修课程的所有课程号**

**3.统计选修C02课程的所有学生的平均成绩、最高成绩和最低成绩**

**4.统计每个院的学生人数**

**5.统计每门课程的选修人数和考试最高分**

**6.统计每个学生的选修门数，并按选课门数的递增顺序显示结果**

**7.统计每个选修课的学生总数和考试的平均成绩**

**8.统计选课门数超过2门的学生的平均成绩和选课门数**

**9.统计总成绩超过200分的学生的学号和总成绩**

**10.统计选修了C02课程的学生的姓名和院**

**11.统计成绩80分以上的学生姓名、课程号和成绩，并按成绩降序排序**

**12.统计计算机学院男生选修了“数据库基础”的学生的姓名、性别和成绩**

**13.统计年龄相同的学生的姓名和年龄**

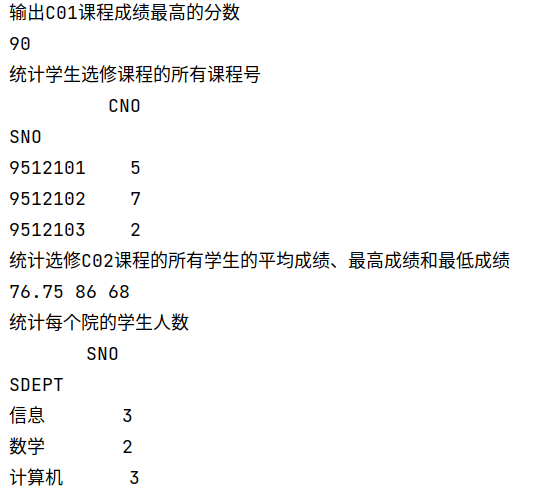
**14.删除成绩小于60分的选课记录**

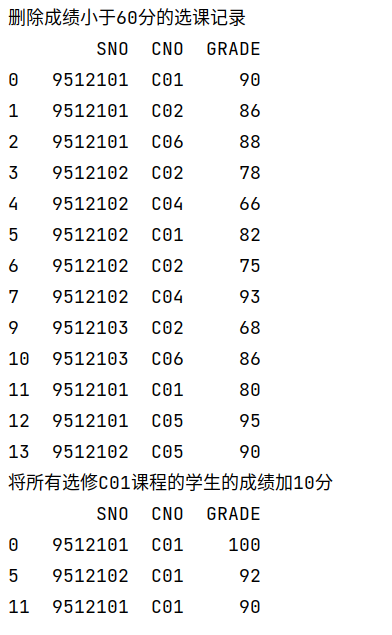
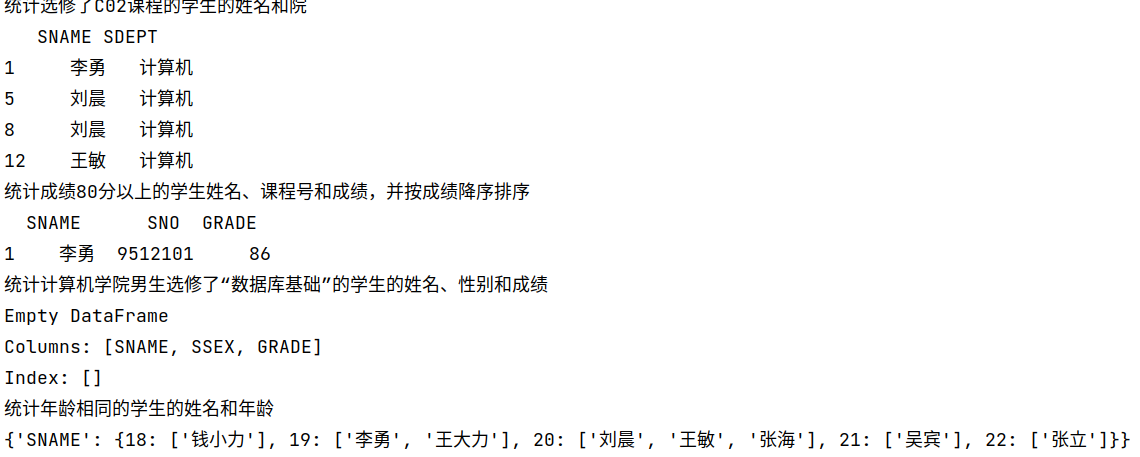
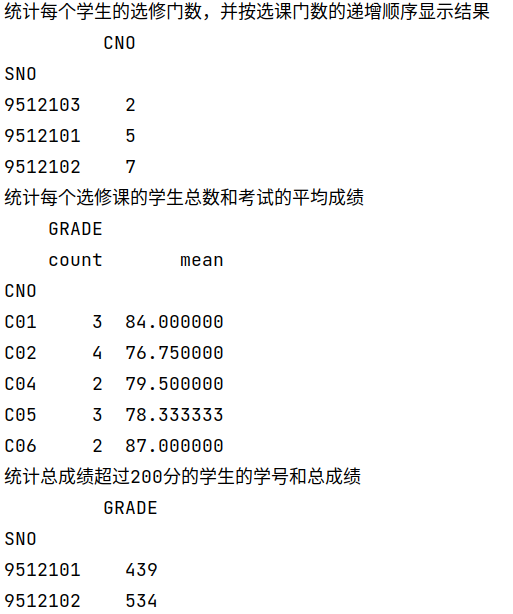
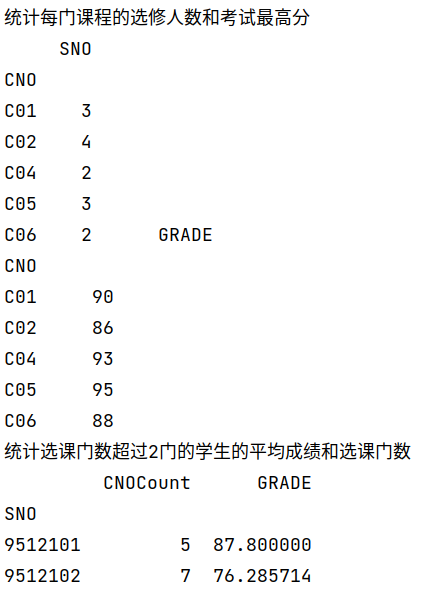
**15.将所有选修C01课程的学生的成绩加10分**

#### 程序源代码：

**import pandas as pd  
from sqlalchemy import create\_engine  
  
  
engine = create\_engine('sqlite:///jgdb1.db')  
detail = pd.read\_sql\_table('STUDENT', con=engine)  
course = pd.read\_sql\_table('COURSE', con=engine)  
sc = pd.read\_sql\_table('SC', con=engine)  
  
print("输出C01课程成绩最高的分数")  
d = sc[sc['CNO'] == 'C01']  
print(d['GRADE'].max())  
  
print("统计学生选修课程的所有课程号")  
d1 = sc[['SNO', 'CNO']].groupby('SNO').count()  
print(d1)  
  
print("统计选修C02课程的所有学生的平均成绩、最高成绩和最低成绩")  
d2 = sc[sc['CNO'] == 'C02']  
print(d2['GRADE'].mean(), d2['GRADE'].max(), d2['GRADE'].min())  
  
print("统计每个院的学生人数")  
d3 = detail[['SDEPT', 'SNO']].groupby('SDEPT').count()  
print(d3)  
  
print("统计每门课程的选修人数和考试最高分")  
d4 = sc[['SNO', 'CNO']].groupby('CNO').count()  
dd4 = sc[['GRADE', 'CNO']].groupby('CNO').max()  
print(d4, dd4)  
  
print("统计选课门数超过2门的学生的平均成绩和选课门数")  
d5 = sc[['SNO', 'CNO']].groupby('SNO').count()  
d5 = d5[d5['CNO'] > 2].sort\_values(by='CNO', na\_position='last')  
d5.rename(columns={'CNO': 'CNOCount'}, inplace=True)  
d5 = pd.merge(d5, sc.groupby('SNO')['GRADE'].mean(), on='SNO')  
print(d5)  
  
################  
  
print("统计每个学生的选修门数，并按选课门数的递增顺序显示结果")  
d6 = sc[['SNO', 'CNO']].groupby('SNO').count()  
d6 = d6.sort\_values(by='CNO', ascending=True)  
print(d6)  
  
  
print("统计每个选修课的学生总数和考试的平均成绩")  
d11 = sc[['CNO', 'GRADE']].groupby('CNO').agg(['count', 'mean'])  
print(d11)  
  
  
  
print("统计总成绩超过200分的学生的学号和总成绩")  
  
d6 = sc[['SNO', 'GRADE']].groupby('SNO').sum()  
d6 = d6[d6['GRADE'] > 200]  
print(d6)  
  
print("统计选修了C02课程的学生的姓名和院")  
  
d7 = pd.merge(detail, sc, on='SNO')  
d7 = d7[d7['CNO'] == 'C02']  
print(d7[['SNAME', 'SDEPT']])  
  
print("统计成绩80分以上的学生姓名、课程号和成绩，并按成绩降序排序")  
  
d8 = d7[d7['GRADE'] > 80]  
print(d8[['SNAME', 'SNO', 'GRADE']].sort\_values(by='GRADE', ascending=False))  
  
print("统计计算机学院男生选修了“数据库基础”的学生的姓名、性别和成绩")  
  
d9 = d7[(d7['CNO'] == 'C04') & (d7['SDEPT'] == '计算机') & (d7['SSEX'] == '男')]  
print(d9[['SNAME', 'SSEX', 'GRADE']])  
  
  
print("统计年龄相同的学生的姓名和年龄")  
result = detail[['SAGE', 'SNAME']].groupby('SAGE')['SNAME'].apply(list).reset\_index(name='SNAME')  
  
# 将年龄列设为索引，并输出结果  
result = result.set\_index('SAGE')  
print(result.to\_dict())  
  
print("删除成绩小于60分的选课记录")  
sc.drop(sc[sc['GRADE'] < 60].index, inplace=True)  
print(sc)  
  
print("将所有选修C01课程的学生的成绩加10分")  
sc.loc[sc['CNO'] == 'C01', 'GRADE'] += 10  
print(sc[sc['CNO'] == 'C01'])**

#### 运行结果：

**?**



## 实验题目八：

**编写python程序将实验题目二创建数据库jgdb2.db，运用pandas函数完成下列查询或统计：**

**1.查询“菊花”、“康”字打头的供应商的产品；**

**2.查询高于平均价格的产品；**

**3.查询10种最贵的产品；**

**4.按商品类别分组统计个数、单价平均值、单价总和，单价最小值、单价最大值；**

**5. 按商品类别分组统计库存总金额；**

**6.按供应商分组统计个数、单价平均值、单价总和，单价最小值、单价最大值；**

**7. 按供应商分组统计库存总金额。**

#### 程序源代码

**import sqlalchemy  
from sqlalchemy import create\_engine**

**import pandas as pd**

**engine = create\_engine(r'sqlite:///jgdb2.db')**

**table = pd.read\_excel(r'商品集.xlsx')**

**table.to\_sql('business', con=engine, index=False, if\_exists='replace')**

**table = pd.read\_sql\_table('business', con=engine)**

**table = table.astype(str)**

**result = table.loc[(table['公司名称'].str.contains('康')) | (table['公司名称'].str.contains('菊花')), :]**

**print('“菊花”、“康”字打头的供应商的产品:\n', result)**

**table[['单价', '库存量']] = table[['单价', '库存量']].astype(float)**

**w = table.loc[table['单价'] > table['单价'].mean(), :]**

**print('高于平均价格的产品:\n', w)**

**result1 = table.sort\_values(by='单价', axis=0, ascending=False, inplace=False, na\_position='last')**

**print('10种最贵的产品:\n', result1.head(10))**

**table1 = table[['类别名称', '单价']].groupby(by='类别名称')**

**print('.按商品类别分组统计个数:\n', table1.size(), '\n平均值:\n', table1.mean(), '\n总和:\n', table1.sum(), '\n最小值:\n',**

**table1.min(), '\n最大值:\n', table1.max())**

**table['库存总金额'] = table['单价'] \* table['库存量']**

**table2 = table[['类别名称', '库存总金额']].groupby(by='类别名称').sum()**

**print('按商品类别分组统计库存总金额:\n', table2)**

**table3 = table[['公司名称', '单价']].groupby(by='公司名称')**

**print('.按供应商分组统计个数:\n', table3.size(), '\n平均值:\n', table3.mean(), '\n总和:\n', table3.sum(), '\n最小值:\n',**

**table3.min(), '\n最大值:\n', table3.max())**

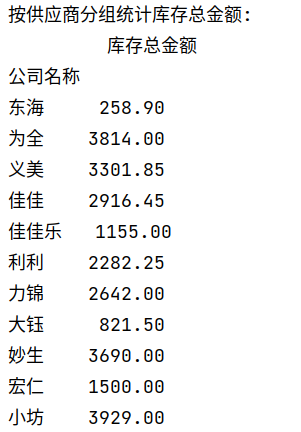
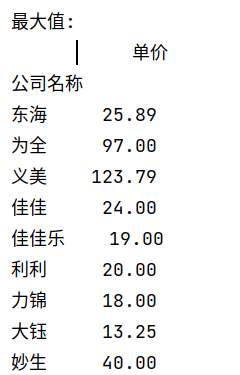
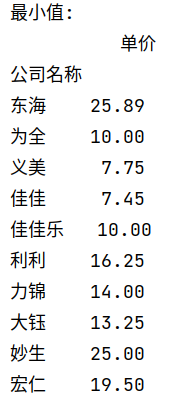
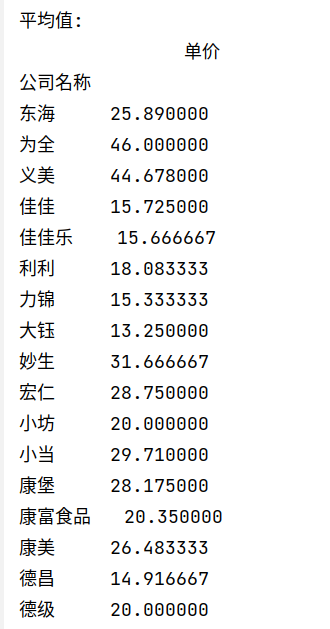
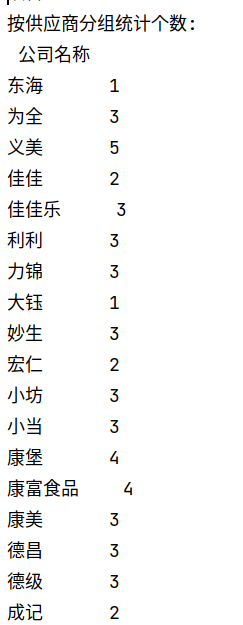
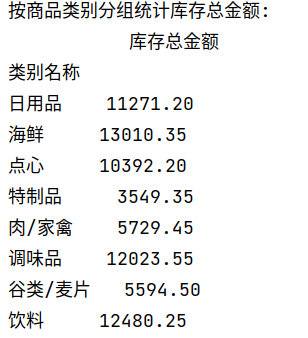
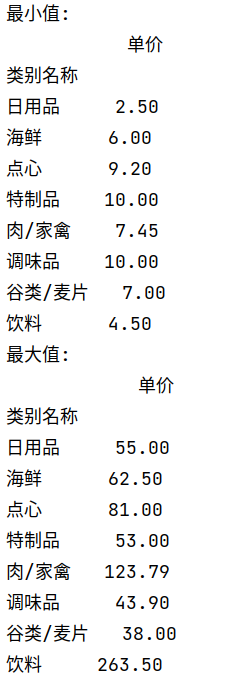
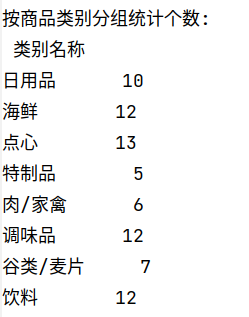
**table4 = table[['公司名称', '库存总金额']].groupby(by='公司名称').sum()**

**print('按供应商分组统计库存总金额:\n', table4)**

#### 运行结果：







## 实验题目九：

**从思源学堂中下载teach.xlsx，编写python程序创建数据库，在Python中通过连接SQLite，运用pandas的函数导入teach.xlsx，形成数据库中的3张表格数据，基本情况表、爱好表和必修成绩表。运用pandas的函数或sql命令语句完成创建下列查询：**

**（1）出生日期在2000年之前的住校学生**

**（2）根据基本情况表，增加年龄字段，形成学生年龄表**

**年龄的计算方法提示: Year(Date())-Year([出生日期])**

**（3）统计每个学生的爱好数量**

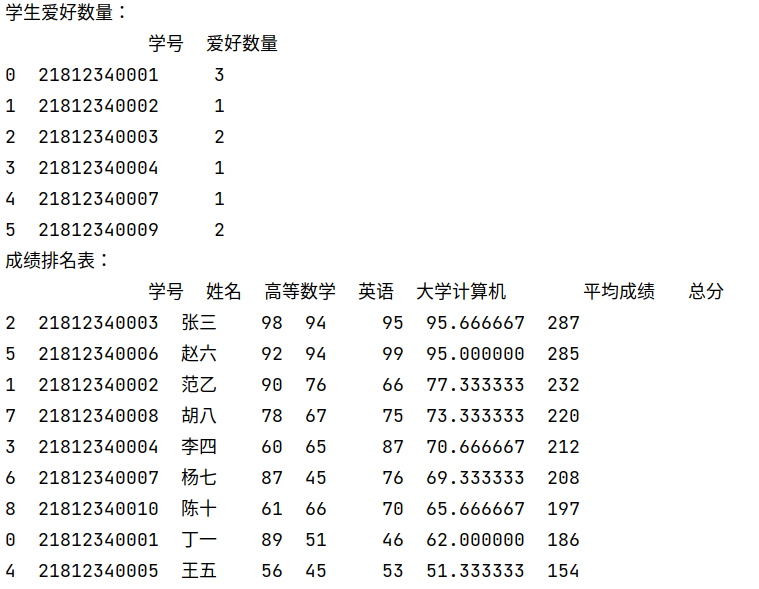
**（4）根据必修成绩表，增加平均分数和总分两个字段，按平均分数排序形成成绩排名表，注意成绩排名表应将基本情况表中姓名字段纳入。**

#### 程序源代码：

**import pandas as pd  
from datetime import datetime  
from sqlalchemy import create\_engine  
  
# Create engine and read data from excel files  
engine = create\_engine('sqlite:///teach.db')  
teach = pd.read\_excel('teach.xlsx', sheet\_name=None)  
  
# Write dataframes into sqlite database tables  
for key, value in teach.items():  
 value.to\_sql(key.lower(), con=engine, index=False, if\_exists='replace')  
  
# Read data from tables in database  
jiben = pd.read\_sql\_table('jiben', con=engine)  
bixiu = pd.read\_sql\_table('bixiu', con=engine)  
aihao = pd.read\_sql\_table('aihao', con=engine)  
  
# Filter students born before 2000 and living in school  
t1 = jiben[(jiben['出生日期'] < '2000-01-01') & (jiben['是否住校'])]  
print('出生于2000年前住校生：\n', t1)  
  
# Calculate age of students  
jiben['年龄'] = (datetime.now().year - jiben['出生日期'].dt.year)  
jiben.to\_sql('nian', con=engine, index=False, if\_exists='replace')  
nian = pd.read\_sql\_table('nian', con=engine)  
print('学生年龄表：\n', nian)  
  
# Count number of hobbies for each student  
ai = aihao.groupby(by='学号', as\_index=False)['爱好'].count()  
ai.rename(columns={'爱好': '爱好数量'}, inplace=True)  
print('学生爱好数量：\n', ai)  
  
# Calculate total score and average score of students and sort by average score  
bixiu['总分'] = bixiu[['高等数学', '英语', '大学计算机']].sum(axis=1)  
bixiu['平均成绩'] = bixiu['总分'] / 3  
bixiu1 = pd.merge(bixiu, jiben, on='学号', how='left')  
bixiu1.sort\_values(by='平均成绩', ascending=False, inplace=True)  
b = bixiu1[['学号', '姓名', '高等数学', '英语', '大学计算机', '平均成绩', '总分']]  
print('成绩排名表：\n', b)**

#### 运行结果：





# **实验小结**

在本次实验过程中，通过学习pandas统计分析和数据库应用，我掌握了处理大量且复杂的数据表的能力，并且学会了如何通过各种方法相融合，以编写出更加快捷高效的可重用性程序。在实验中，我学会了如何使用pandas库进行数据的快速处理、清洗、操作，并通过使用SQLAlchemy等工具连接和查询不同类型的数据库，掌握了数据库应用的基本操作。同时，在实验过程中，我特别注重将理论和实践相结合，通过多次练习和实践，确保自己学到的知识可以在实际的数据分析工作中得到应用。