Project1 分析

1.将原始数据导入表中

1.1将json文件转换为csv文件

jwxt_parser.py

```
import json
import csv

# Run as: python3 jwxt_parser.py <json file>
with open('./course_info.json', 'r', encoding='utf-8') as f:
    data = json.load(f)



f = open('course_info.csv','w',encoding='utf-8')
w = csv.writer(f)
w.writerow(['totalCapacity','courseId','prerequisite','courseHour','courseCredit','courseName','className','courseDept','teacher','weekList1','location1','classTime1','weekday1','weekList2','location2','classTime2','weekday2','weekList3','location3','classTime3','weekday3'])



# iterate through all courses
for course in data:
    courselist = []
    courselist.append(course['totalCapacity'])
    courselist.append(course['courseId'])
    courselist.append(course['prerequisite'])
    courselist.append(course['courseHour'])
    courselist.append(course['courseCredit'])
    courselist.append(course['courseName'])
    courselist.append(course['className'])
```

```
    courselist.append(course['courseDept'])
    courselist.append(course['teacher'])
    for c in course['classList']:
        courselist.append(c['weekList'])
        courselist.append(c['location'])
        courselist.append(c['classTime'])
        courselist.append(c['weekday'])

    w.writerow(courselist)

f.close()
```

1.2将csv文件导入postgreSQL中

csv2postgreSQL

```
import psycopg2
import pandas as pd


def csv2postgreSQL(csv_name, database="postgres", user="postgres", password="20010922nyh", host="localhost",
                   port="5432"):
    # database = input('database:')
    # user = input('user:')
    # password = input('password:')
    # host = input('host:')
    # port = input('port:')

    print('连接数据库中...')
    database = psycopg2.connect(database=database, user=user, password=password, host=host, port=port)
    cursor = database.cursor()
    print('连接成功')

    print('你现在正在准备将csv表导入PostgreSQL里，请注意，目前本方法无法添加外键、主键、修饰等...\n')
    print('目前输入数据类型全部为VARCHAR(400)')

    # schema 名称
```

```
22
    ##################################################################
    ################
23      schema_name = 'project1_dirty'
24
    ##################################################################
    ################
25
26      # 读入csv
27
    ##################################################################
    ################
28      csv = pd.read_csv(csv_name)
29      column_num = len(csv.columns)
30
    ##################################################################
    ################
31
32
33      # 各列名称
34
    ##################################################################
    ################
35      header = []
36      for i in range(0, len(csv.columns)):
37          header.append(csv.columns[i])
38      print('这是一个有 ' + str(column_num) + ' 列的csv文件')
39
    ##################################################################
    ################
40
41      # table 名称
42
    ##################################################################
    ################
43      # table_name = 'course_info'
44      table_name = 'select_course'
45      print('\n请依次输入每一列需要的数据类型(以PostgreSQL的形式正确输
    入)')
46
    ##################################################################
    ################
47
```

```
    # 各列数据类型

 ##################################################################
################
    # datatype = ['INT','VARCHAR(400)','VARCHAR(400)','INT',
    #             'INT','VARCHAR(400)','VARCHAR(400)',
    #             'VARCHAR(400)','VARCHAR(400)','VARCHAR(400)',
    #             'VARCHAR(400)','VARCHAR(400)','INT',
    #
'VARCHAR(400)','VARCHAR(400)','VARCHAR(400)','INT',
    #
'VARCHAR(400)','VARCHAR(400)','VARCHAR(400)','INT']
    datatype =
['VARCHAR(400)','VARCHAR(400)','VARCHAR(400)','INT',
                'VARCHAR(400)','VARCHAR(400)','VARCHAR(400)',
                'VARCHAR(400)','VARCHAR(400)','VARCHAR(400)']

 ##################################################################
################

    # 建表语句

 ##################################################################
################
    create_table_sql = 'CREATE TABLE IF NOT EXISTS ' +
schema_name + '.' + table_name + '('
    for i in range(0, column_num - 1):
        create_table_sql = create_table_sql + header[i] + ' ' +
datatype[i] + ','
    create_table_sql = create_table_sql + header[column_num -
1] + ' ' + datatype[column_num - 1] + ')' + ';'

    create_table_sql1 = 'CREATE TABLE IF NOT EXISTS ' +
schema_name + '.' + table_name + '(\n'
    for i in range(0, column_num - 1):
        create_table_sql1 = create_table_sql1 + header[i] + ' '
+ datatype[i] + ',\n'
    create_table_sql1 = create_table_sql1 + header[column_num -
1] + ' ' + datatype[column_num - 1] + ')' + ';\n'
    print('查看你的建表语句:\n' + create_table_sql1)

 ##################################################################
################
```

```
74
75      # 执行操作
76
    ##################################################################
    ################
77      cursor.execute(create_table_sql)
78      database.commit()
79      print(table_name + '表创建成功\n')
80
    ##################################################################
    ################
81
82      print('开始插入数据...')
83
    ##################################################################
    ################
84      csv = pd.read_csv(csv_name)
85      for i in range(len(csv)):
86          data = []
87          for j in range(column_num):
88              # 空值判断
89              if pd.isnull(csv.at[i, header[j]]):
90                  data.append('NULL')
91              elif (datatype[j] == 'INT'):
92                  data.append(str(csv.at[i, header[j]]))
93              elif (datatype[j][0:7] == 'VARCHAR'):
94                  str_value = str(csv.at[i, header[j]])  # 里面实
    际的值
95                  # 处理单引号问题 替换成
96                  # todo 这里还有其它没有处理的特殊符号
97                  str_value = str_value.replace("'", "")
98                  str_data = '\'' + str_value + '\''
99                  data.append(str_data)
100
101         insert_sql = 'INSERT INTO ' + schema_name + '.' +
    table_name + '('
102         for j in range(len(header)):
103             if (data[j] != 'NULL'):
104                 insert_sql = insert_sql + header[j] + ', '
105         insert_sql = insert_sql[0:-2] + ')' + ' VALUES ('
106         for j in range(len(header)):
107             if (data[j] != 'NULL'):
108                 insert_sql = insert_sql + data[j] + ', '
```

```
109             insert_sql = insert_sql[0:-2]
110             insert_sql = insert_sql + ' )' + ';'
111             print(insert_sql)
112             cursor.execute(insert_sql)
113             database.commit()
114
    ##################################################################
    ################
115         print('插入成功! ')
116         cursor.close()
117         database.commit()
118         database.close()
119         print('csv导入PostgreSQL完成')
120
121
122 csv2postgreSQL('./select_course.csv')
```

建表

student表 ok

STUDENT NUMBER	NAME	GENDER	COLLEGE NUMBER
学号	姓名	性别	书院编号
PRIMARY KEY			FOREIGN KEY→
UNIQUE	NOT NULL	NOT NULL	

college表 ok

COLLEGE NUMBER	CHINESE NAME	ENGLISH NAME
书院编号	中文名	英文名
PRIMARY KEY		
AUTO INCREASEMENT	UNIQUE1	UNIQUE2

department表 ok

ID	NAME
院系编号	院系名称
PRIMARY KEY	
AUTO INCREASEMENT	UNIQUE

teacher表 ok

ID	NAME
教师编号	姓名
PRIMARY KEY	
AUTO INCREASEMENT	NOT NULL UNIQUE

course表 ok

ID	COURSE ID	NAME	COURSE HOUR	CREDIT	DEPARTMENT ID
课程编号	课程代码	课程名字	课时	学分	院系编号
PRIMARY KEY					FOREIGN KEY→
AUTO INCREASEMENT	UNIQUE	NOT NULL			

class表 ok

CLASS ID	COURSE ID	LANGUAGE	CLASS NAME	ABOARDING	REMARK
班级编号	课程编号	授课语言	课程名称	是否为留学生课程	备注
PRIMARY KEY					

CLASS ID	COURSE ID	LANGUAGE	CLASS NAME	ABOARDING	REMARK
AUTO INCREASEMENT	UNIQUE1		UNIQUE2		

sport 表 ok

CLASS ID	COURSE ID	LANGUAGE	CLASS NAME	ABOARDING	REMARK
班级编号	课程编号	授课语言	课程名称	是否为留学生课程	备注
PRIMARY KEY					
AUTO INCREASEMENT	UNIQUE1		UNIQUE2		

distribution表

DISTRIBUTION ID	CLASS ID	LOCATION ID	WEEK ID	CLASS TIME	WEEKDAY
安排编号	班级编号	上课地点编号	week表编号	上课时间	上课的时间是星期几
PRIMARY KEY	FOREIGN KEY→	FOREIGN KEY→	FOREIGN KEY →		
AUTO INCREASEMENT	UNIQUE1	UNIQUE2	UNIQUE3	UNIQUE4	UNIQUE5

location表 ok

LOCATION ID	AREA	BUILDING	FUNCTION	ROOM NUMBR
上课地点编号	区域	楼栋	功能	房间编号
PRIMARY KEY				
AUTO INCREASEMENT	UNIQUE1			UNIQUE2

OR

ID主键　classID(先修课的) groupID(组ID) classID(后修课)

AND

ID主键 classID(后修课) groupID(组ID)

```
1  (材料科学基础 或者 材料科学基础) 并且 (高等数学（下）A 或者 高等数学（下） 或者 数学分析II) 并且 晶体学
```

prerequisite表 待定 还不确定怎么做

P CLASS ID	A CLASS ID
先修课编号	后修课编号
PRIMARY KEY1 FOREIGN KEY→	PRIMARY KEY2 FOREIGN KEY→

P CLASS ID	A CLASS ID

teacher-class表 ok

TEACHER ID	CLASS ID
PRIMARY KEY1 FOREIGN KEY→	PRIMARY KEY2 FOREIGN KEY→

student-class表

STUDENT ID	CLASS ID
学生编号	课程编号
PRIMARY KEY1 FOREIGN KEY→	PRIMARY KEY2 FOREIGN KEY→

week表

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
week ID	第一周	第二周	第三后	第四周	第五周	第六周	第七周	第八周	第九周	第十周	第十一周	第十二周	第十三周	第十四周	第十五周
PRIMARY KEY															
AUTO INCREASEMENT															