

Assignment 5

Jiarong Ye

September 26, 2018

Load the data

```
In [94]: import pandas as pd
         from datetime import datetime
         import pymysql

In [268]: department = pd.read_csv('departments.csv', sep=',')
         department.columns = ['department_name']
         shift = pd.read_csv('shifts.csv', sep=',')
         shift.columns = ['from_time', 'length']
         employees = pd.read_csv('employees.csv', sep=',')
         employees.emptytype = employees.emptytype.fillna('')
         schedule = pd.read_csv('schedule.csv', sep=',')
         schedule.columns = ['date', 'empid', 'dept', 'start_time', 'shift_length']
```

Manipulate the data

```
In [316]: class DataSqlLoader:
         def __init__(self, database):
             # connect to mysql local server
             self.db = pymysql.Connect('localhost',
                                     'root',
                                     '',
                                     database)
             self.c = db.cursor()

             # convert the shift and schedule time to `time` format compatible in MySQL
             def convert_time_format(self, time):
                 return datetime.strptime('{}'.format(time), '%I%p').strftime('%H:%M:%S')

             # convert schedule date to `date` format compatible in MySQL
             def convert_date_format(self, time):
                 return datetime.strptime('{}'.format(time), '%m/%d/%Y').strftime('%Y-%m-%d')

             def creat_tables(self):
                 self.c.execute('''
                     create table if not exists department
                     (
                         department_id int auto_increment
                         primary key,
                         department_name varchar(50) not null
                     );
                 ''')
                 self.c.execute('''
```

```

        create table if not exists shift
        (
            shift_id int auto_increment
                primary key,
            from_time time not null,
            length int not null
        );
'''
self.c.execute('''
create table if not exists schedule
(
    schedule_id int auto_increment
        primary key,
    date date not null,
    empid varchar(10) not null,
    dept varchar(50) not null,
    start_time time not null,
    shift_length int not null
);
''')
self.c.execute('''
create table if not exists employees
(
    empid varchar(10) not null primary key,
    lastname varchar(20) not null,
    firstname varchar(20) not null,
    emptype varchar(3) null,
    cellphone varchar(20) null,
    homephone varchar(20) null,
    ftpt varchar(2) not null,
    constraint employee_empid_uindex
        unique (empid)
);
''')

def insert_into_tables(self, table, table_name):
    for i in range(len(table)):
        attributes = '{}'.format(tuple(table.columns.tolist())).replace("'", "")
        query = "insert into {} {} values {};" .format(
            table_name, attributes, tuple(table.iloc[i,:].values))
        query = query.replace("(none)", "")
        query = query.replace(r",)", ",")
        # print(query)
        try:
            c.execute(query)
            db.commit()
        except Exception as e:
            print(e)

def close(self):
    self.db.close()

```

```

In [317]: dsl = DataSqlLoader('cs431_project')
          dsl.creat_tables()

```

```

schedule['start_time'] = list(map(lambda x: dsl.convert_time_format(x),
                                   schedule['start_time']))
shift['from_time'] = list(map(lambda x: dsl.convert_time_format(x),
                               shift['from_time']))
schedule['date'] = list(map(lambda x: dsl.convert_date_format(x),
                             schedule['date']))

dsl.insert_into_tables(department, 'department')
dsl.insert_into_tables(shift, 'shift')
dsl.insert_into_tables(schedule, 'schedule')
dsl.insert_into_tables(employees, 'employees')

```

SQL Query

```

In [ ]: select e.lastname as LAST, e.firstname as FIRST, e.cellphone as CELL,
          s.date as DATE, d.department_name as DEPT,
          date_format(s.start_time, '%I%p') as START, s.shift_length as SHIFT_LENGTH
from department as d, employees as e, schedule as s
where s.dept=d.department_name
and s.empid=e.empid
order by LAST, DATE, SHIFT_LENGTH asc, DEPT desc limit 50;

```

display results

```

In [315]: query = '''
          select e.lastname as LAST, e.firstname as FIRST, e.cellphone as CELL,
          s.date as DATE, d.department_name as DEPT,
          date_format(s.start_time, '%I%p') as START, s.shift_length as SHIFT_LENGTH
from department as d, employees as e, schedule as s
where s.dept=d.department_name
and s.empid=e.empid
order by LAST, DATE, START, SHIFT_LENGTH asc, DEPT desc limit 50;
          '''

df = pd.read_sql(query, dsl.db)
#print(df.to_latex())

```

	LAST	FIRST	CELL	DATE	DEPT	START	SHIFT_LENGTH
0	Ault	Rosendo	(571)7252-4785	2018-09-24	CARDIOLOGY	03PM	8
1	Ault	Rosendo	(571)7252-4785	2018-09-25	PEDIATRICS	03PM	8
2	Ault	Rosendo	(571)7252-4785	2018-09-25	ONCOLOGY	03PM	8
3	Ault	Rosendo	(571)7252-4785	2018-09-26	PEDIATRICS	11PM	8
4	Ault	Rosendo	(571)7252-4785	2018-09-28	ONCOLOGY	03PM	8
5	Ault	Rosendo	(571)7252-4785	2018-09-29	CARDIOLOGY	07AM	12
6	Ault	Rosendo	(571)7252-4785	2018-10-01	CARDIOLOGY	03PM	8
7	Ault	Rosendo	(571)7252-4785	2018-10-02	ONCOLOGY	03PM	8
8	Ault	Rosendo	(571)7252-4785	2018-10-03	PEDIATRICS	07AM	8
9	Ault	Rosendo	(571)7252-4785	2018-10-04	EMERGENCY	03PM	8
10	Ault	Rosendo	(571)7252-4785	2018-10-04	RADIOLOGY	11PM	8
11	Ault	Rosendo	(571)7252-4785	2018-10-05	EMERGENCY	03PM	8
12	Ault	Rosendo	(571)7252-4785	2018-10-07	CARDIOLOGY	07PM	12
13	Beaver	Viva	(612)7951-8931	2018-09-25	MATERNITY	03PM	8
14	Beaver	Viva	(612)7951-8931	2018-09-25	EMERGENCY	07AM	8
15	Beaver	Viva	(612)7951-8931	2018-09-26	ONCOLOGY	07AM	8
16	Beaver	Viva	(612)7951-8931	2018-09-26	ONCOLOGY	07AM	8
17	Beaver	Viva	(612)7951-8931	2018-09-26	INTENSIVE CARE	07AM	8
18	Beaver	Viva	(612)7951-8931	2018-09-27	PEDIATRICS	11PM	8
19	Beaver	Viva	(612)7951-8931	2018-09-28	CARDIOLOGY	07AM	8
20	Beaver	Viva	(612)7951-8931	2018-10-02	INTENSIVE CARE	07AM	8
21	Beaver	Viva	(612)7951-8931	2018-10-03	PEDIATRICS	11PM	8
22	Beaver	Viva	(612)7951-8931	2018-10-06	RADIOLOGY	07AM	12
23	Beaver	Viva	(612)7951-8931	2018-10-07	PEDIATRICS	07PM	12
24	Beaver	Viva	(612)7951-8931	2018-10-07	MATERNITY	07PM	12
25	Beaver	Viva	(612)7951-8931	2018-10-07	EMERGENCY	07PM	12
26	Berryman	Sabrina	(806)6465-4545	2018-09-24	EMERGENCY	07AM	8
27	Berryman	Sabrina	(806)6465-4545	2018-09-28	CARDIOLOGY	03PM	8
28	Berryman	Sabrina	(806)6465-4545	2018-09-29	CARDIOLOGY	07AM	12
29	Berryman	Sabrina	(806)6465-4545	2018-10-03	SURGERY	03PM	8
30	Berryman	Sabrina	(806)6465-4545	2018-10-04	EMERGENCY	03PM	8
31	Berryman	Sabrina	(806)6465-4545	2018-10-04	ONCOLOGY	11PM	8
32	Berryman	Sabrina	(806)6465-4545	2018-10-04	INTENSIVE CARE	11PM	8
33	Berryman	Sabrina	(806)6465-4545	2018-10-06	PEDIATRICS	07AM	12
34	Berryman	Sabrina	(806)6465-4545	2018-10-06	ONCOLOGY	07AM	12
35	Berryman	Sabrina	(806)6465-4545	2018-10-06	MATERNITY	07AM	12
36	Berryman	Sabrina	(806)6465-4545	2018-10-07	CARDIOLOGY	07AM	12
37	Blagg	Lorenzo	(863)2419-9105	2018-09-26	EMERGENCY	07AM	8
38	Blagg	Lorenzo	(863)2419-9105	2018-09-27	INTENSIVE CARE	03PM	8
39	Blagg	Lorenzo	(863)2419-9105	2018-09-29	SURGERY	07PM	12
40	Blagg	Lorenzo	(863)2419-9105	2018-10-01	CARDIOLOGY	03PM	8
41	Blagg	Lorenzo	(863)2419-9105	2018-10-03	ONCOLOGY	03PM	8
42	Blagg	Lorenzo	(863)2419-9105	2018-10-05	RADIOLOGY	03PM	8
43	Blagg	Lorenzo	(863)2419-9105	2018-10-05	ONCOLOGY	07AM	8
44	Blagg	Lorenzo	(863)2419-9105	2018-10-06	ONCOLOGY	07PM	12
45	Blagg	Lorenzo	(863)2419-9105	2018-10-07	ONCOLOGY	07PM	12
46	Catanzaro	Imelda	(220)3748-1866	2018-09-24	PEDIATRICS	07AM	8
47	Catanzaro	Imelda	(220)3748-1866	2018-09-30	ONCOLOGY	07PM	12
48	Catanzaro	Imelda	(220)3748-1866	2018-09-30	EMERGENCY	07PM	12
49	Catanzaro	Imelda	(220)3748-1866	2018-10-01	MATERNITY	03PM	8