

Assignment 7

Jiarong Ye

November 27, 2018

Load the data

```
In [40]: import pandas as pd
         from datetime import datetime
         import pymysql

In [41]: 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 [48]: class DataSqlLoader:
         def __init__(self, database):
             # connect to mysql local server
             self.db = pymysql.Connect(
                 host='us-cdbr-iron-east-01.cleardb.net',
                 port=3306,
                 user='b033ff6b193dc0',
                 passwd='02f96442',
                 db=database)
             self.c = self.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)
    );
    '''

self.c.execute('''
    create table if not exists department_managers
    (
        dept_name varchar(20),
        manager varchar(30)

```

```

    );
    '''

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:
            self.c.execute(query)
            self.db.commit()
        except Exception as e:
            print(e)

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

```

```

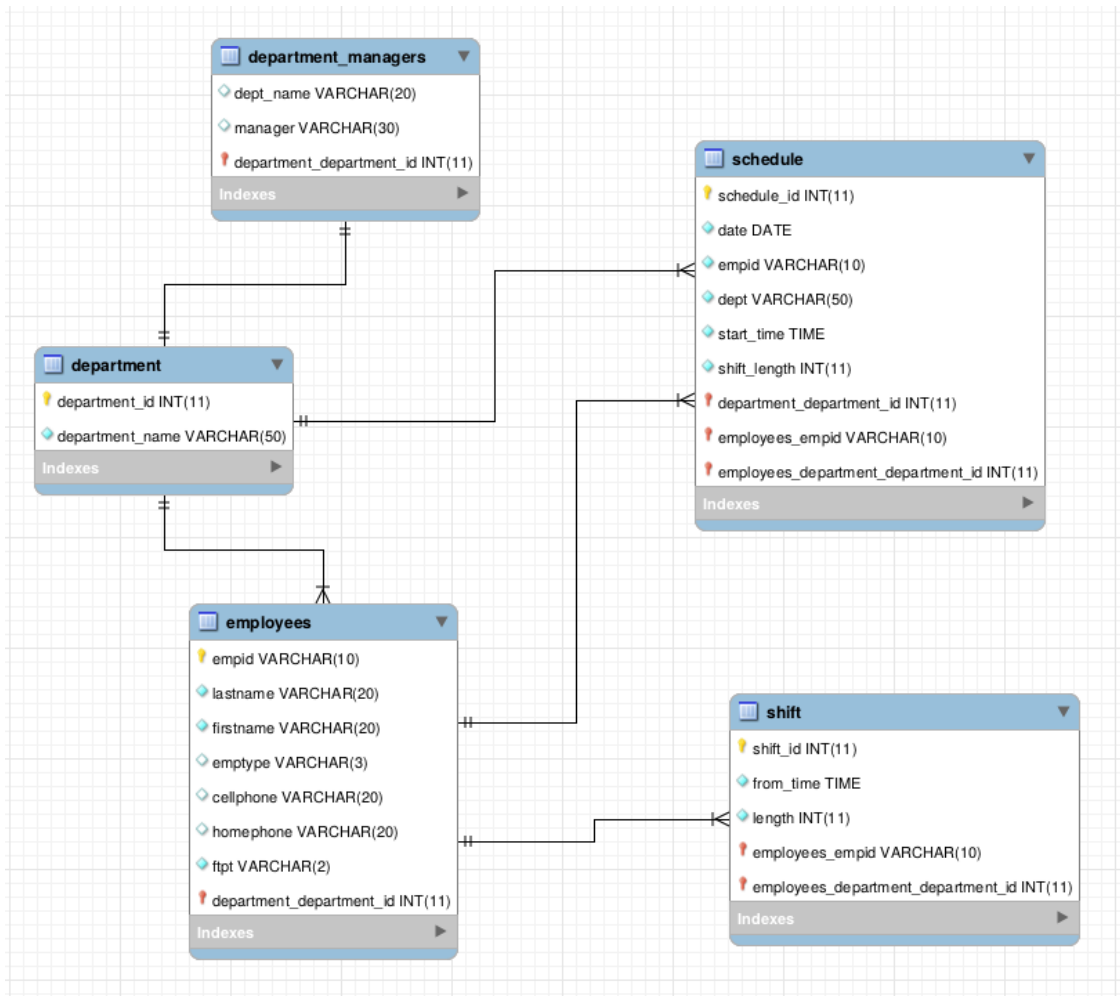
In [110]: dsl = DataSqlLoader('heroku_f2b490785080ebd')
          dsl.creat_tables()

          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')

```

ER Diagram



SQL Query

php demo site: [click here](#)

(source code attached in the submission)

1. Show tables

```
In [62]: query = 'show tables;'
         tables = pd.read_sql(query, dsl.db)
         print(tables.to_latex())
```

	Tables
0	department
1	department_managers
2	employees
3	schedule
4	shift

2. Describe tables

```
In [55]: for table in tables.values:
        query = 'describe {}'.format(table[0])
        df = pd.read_sql(query, dsl.db)
        print(df)
        print('\n')
```

	Field	Type	Null	Key	Default	Extra
0	department_id	int(11)	NO	PRI	None	auto_increment
1	department_name	varchar(50)	NO		None	

	Field	Type	Null	Key	Default	Extra
0	dept_name	varchar(20)	YES		None	
1	manager	varchar(30)	YES		None	

	Field	Type	Null	Key	Default	Extra
0	empid	varchar(10)	NO	PRI	None	
1	lastname	varchar(20)	NO		None	
2	firstname	varchar(20)	NO		None	
3	emptype	varchar(3)	YES		None	
4	cellphone	varchar(20)	YES		None	
5	homephone	varchar(20)	YES		None	
6	ftpt	varchar(2)	NO		None	

	Field	Type	Null	Key	Default	Extra
0	schedule_id	int(11)	NO	PRI	None	auto_increment
1	date	date	NO		None	
2	empid	varchar(10)	NO		None	
3	dept	varchar(50)	NO		None	
4	start_time	time	NO		None	
5	shift_length	int(11)	NO		None	

	Field	Type	Null	Key	Default	Extra
0	shift_id	int(11)	NO	PRI	None	auto_increment
1	from_time	time	NO		None	

2 length int(11) NO None

3. Find out the need of employees of certain type in certain department during certain period (eg: EMERGENCY, RN, from 2018-10-03 to 2018-10-11)

```
In [91]: query = '''
        SELECT s.dept AS Department,
               s.empid AS EmployeeID,
               s.date AS Date,
               date_format(s.start_time, '%I%p') AS ShiftStartTime,
               e.emptype AS EmployeeType
        FROM schedule as s, department as d, employees as e
        WHERE s.dept=d.department_name
              AND s.empid = e.empid
              AND s.date >= '2018-10-03'
              AND s.date <= '2019-10-11'
              AND s.dept = 'EMERGENCY'
              AND e.emptype = 'RN'
        ORDER BY d.department_id, s.date;
        '''

        tables = pd.read_sql(query, dsl.db)
        print(tables.to_latex())
```

	Department	EmployeeID	Date	ShiftStartTime	EmployeeType
0	EMERGENCY	940824	2018-10-03	11PM	RN
1	EMERGENCY	854480	2018-10-04	03PM	RN
2	EMERGENCY	860127	2018-10-05	07AM	RN
3	EMERGENCY	945540	2018-10-07	07AM	RN
4	EMERGENCY	921331	2018-10-07	07PM	RN

4. Find out whether certain employee has been scheduled on a certain date (eg: Remona Locke on 2018-10-03)

```
In [92]: query = '''
        SELECT s.schedule_id AS ScheduleID,
               s.empid AS EmployeeID,
               concat(e.firstname, ' ', e.lastname) AS Fullname,
               s.dept AS Department,
               s.date AS Date,
               date_format(s.start_time, '%I%p') AS ShiftStartTime,
               s.shift_length AS ShiftLength
        FROM employees as e, schedule as s
        WHERE s.empid = e.empid
              AND e.firstname = 'Remona'
              AND e.lastname = 'Locke'
```

```

        AND s.date = '2018-10-03'
    '''
    tables = pd.read_sql(query, dsl.db)
    print(tables.to_latex())

```

ScheduleID	EmployeeID	Fullname	Department	Date	ShiftStartTime	ShiftLength
------------	------------	----------	------------	------	----------------	-------------

5. If not, then this employee is open for schedule

```

In [95]: query = '''
        INSERT INTO schedule (date, empid, dept, start_time, shift_length) VALUES
            ('2018-10-03',
             (SELECT empid FROM employees WHERE firstname='Remona' AND lastname=
              'EMERGENCY',
              '12:00:00',
              8)
            )
    '''
    dsl.c.execute(query)

```

Out[95]: 1

6. Check the schedule

```

In [96]: query = '''
        SELECT s.schedule_id AS ScheduleID,
               s.empid AS EmployeeID,
               concat(e.firstname, ' ', e.lastname) AS Fullname,
               s.dept AS Department,
               s.date AS Date,
               date_format(s.start_time, '%I%p') AS ShiftStartTime,
               s.shift_length AS ShiftLength
        FROM employees as e, schedule as s
        WHERE s.empid = e.empid
              AND e.firstname = 'Remona'
              AND e.lastname = 'Locke'
              AND s.date = '2018-10-03'
    '''
    tables = pd.read_sql(query, dsl.db)
    print(tables.to_latex())

```

	ScheduleID	EmployeeID	Fullname	Department	Date	ShiftStartTime	ShiftLength
0	4371	854480	Remona Locke	EMERGENCY	2018-10-03	12PM	8

7. Unschedule

```

In [107]: query = '''
        SELECT s.schedule_id
        FROM employees as e, schedule as s

```

```

        WHERE s.empid = e.empid
        AND e.firstname = 'Remona'
        AND e.lastname = 'Locke'
        AND s.date = '2018-10-03'
    '''
    dsl.c.execute(query)
    schedule_id = dsl.c.fetchall()[0][0]
    query = '''
        DELETE FROM schedule WHERE schedule_id={}
    '''.format(schedule_id)
    dsl.c.execute(query)

```

Out[107]: 1

8. Check the employee has been unscheduled

```

In [111]: query = '''
        SELECT s.schedule_id AS ScheduleID,
               s.empid AS EmployeeID,
               concat(e.firstname, ' ', e.lastname) AS Fullname,
               s.dept AS Department,
               s.date AS Date,
               date_format(s.start_time, '%I%p') AS ShiftStartTime,
               s.shift_length AS ShiftLength
        FROM employees as e, schedule as s
        WHERE s.empid = e.empid
               AND e.firstname = 'Remona'
               AND e.lastname = 'Locke'
               AND s.date = '2018-10-03'
    '''
    tables = pd.read_sql(query, dsl.db)
    print(tables.to_latex())

```

ScheduleID	EmployeeID	Fullname	Department	Date	ShiftStartTime	ShiftLength
------------	------------	----------	------------	------	----------------	-------------