

Assignment 6

Jiarong Ye

October 8, 2018

Load the data

```
In [1]: import pandas as pd
        from datetime import datetime
        import pymysql

In [2]: 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 [5]: class DataSqlLoader:
        def __init__(self, database):
            # connect to mysql local server
            self.db = pymysql.Connect(
                host='localhost',
                port=3308,
                user='root',
                passwd='',
                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 [11]: 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

show tables

```

In [7]: query = 'show tables;'
        tables = pd.read_sql(query, dsl.db)
        tables

```

```

Out[7]:  Tables_in_cs431_project
0      department
1  department_managers
2      employees
3      schedule
4      shift

```

describe tables

```

In [14]: 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 emptytype	varchar(3)	YES			None
4 cellphone	varchar(20)	YES			None
5 homephone	varchar(20)	YES			None
6 ftype	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

1. GENERATE THE FOLLOWING REPORT LISTING ALL EMPLOYEES WHO ARE “OVERBOOKED” BY BEING SCHEDULED MORE THAN ONCE IN A DAY

```
In [13]: query = '''
SELECT a.empid, a.lastname,a.firstname, a.date AS Overbooked_Date,
       date_format(a.start_time, '%I%p') AS sfrom, a.dept_name, a.manager
FROM
  (SELECT s.empid, e.lastname, e.firstname,
   s.date, s.start_time, d.dept_name, d.manager
   FROM employees AS e, schedule AS s, department_managers AS d
   WHERE e.empid=s.empid
   and d.dept_name=s.dept) AS a
INNER JOIN
  (SELECT empid, date, start_time FROM schedule
   GROUP BY empid, date
   HAVING count(*) >1) AS b
ON (a.empid, a.date)=(b.empid, b.date)
ORDER BY a.empid, a.date, a.start_time;
'''

df = pd.read_sql(query, dsl.db)
```

	empid	lastname	firstname	Overbooked_Date	sfrom	dept_name	manager
0	853537	Zaragoza	Manie	2018-09-27	07AM	ONCOLOGY	Junita Loranger
1	853537	Zaragoza	Manie	2018-09-27	11PM	PEDIATRICS	Remona Locke
2	853537	Zaragoza	Manie	2018-10-04	07AM	INTENSIVE CARE	Colby Witham
3	853537	Zaragoza	Manie	2018-10-04	03PM	PEDIATRICS	Remona Locke
4	853537	Zaragoza	Manie	2018-10-05	07AM	INTENSIVE CARE	Colby Witham
5	853537	Zaragoza	Manie	2018-10-05	03PM	PEDIATRICS	Remona Locke
6	854480	Locke	Remona	2018-09-29	07PM	EMERGENCY	Oralia Swallow
7	854480	Locke	Remona	2018-09-29	07PM	INTENSIVE CARE	Colby Witham
8	854480	Locke	Remona	2018-10-04	03PM	EMERGENCY	Oralia Swallow
9	854480	Locke	Remona	2018-10-04	03PM	SURGERY	Nadene Harwell
10	854480	Locke	Remona	2018-10-05	07AM	CARDIOLOGY	Marylin Mumaw
11	854480	Locke	Remona	2018-10-05	07AM	SURGERY	Nadene Harwell
12	856661	Rabb	Stephan	2018-09-24	11PM	INTENSIVE CARE	Colby Witham
13	856661	Rabb	Stephan	2018-09-24	11PM	ONCOLOGY	Junita Loranger
14	857230	He	Kimiko	2018-10-05	03PM	INTENSIVE CARE	Colby Witham
15	857230	He	Kimiko	2018-10-05	11PM	SURGERY	Nadene Harwell
16	859272	Netter	Almeda	2018-10-03	07AM	ONCOLOGY	Junita Loranger
17	859272	Netter	Almeda	2018-10-03	07AM	INTENSIVE CARE	Colby Witham
18	864272	Catanzaro	Imelda	2018-09-30	07PM	EMERGENCY	Oralia Swallow
19	864272	Catanzaro	Imelda	2018-09-30	07PM	ONCOLOGY	Junita Loranger
20	864272	Catanzaro	Imelda	2018-10-02	07AM	EMERGENCY	Oralia Swallow
21	864272	Catanzaro	Imelda	2018-10-02	03PM	SURGERY	Nadene Harwell
22	864272	Catanzaro	Imelda	2018-10-02	11PM	EMERGENCY	Oralia Swallow
23	864425	Witham	Colby	2018-09-24	03PM	ONCOLOGY	Junita Loranger
24	864425	Witham	Colby	2018-09-24	03PM	SURGERY	Nadene Harwell
25	864425	Witham	Colby	2018-09-25	03PM	CARDIOLOGY	Marylin Mumaw
26	864425	Witham	Colby	2018-09-25	11PM	INTENSIVE CARE	Colby Witham
27	864425	Witham	Colby	2018-10-03	07AM	SURGERY	Nadene Harwell
28	864425	Witham	Colby	2018-10-03	07AM	MATERNITY	Carmel Mersch
29	868513	Mcclelland	Barton	2018-09-29	07AM	ONCOLOGY	Junita Loranger
30	868513	Mcclelland	Barton	2018-09-29	07PM	EMERGENCY	Oralia Swallow
31	868513	Mcclelland	Barton	2018-09-29	07PM	INTENSIVE CARE	Colby Witham
32	868513	Mcclelland	Barton	2018-10-05	03PM	EMERGENCY	Oralia Swallow
33	868513	Mcclelland	Barton	2018-10-05	11PM	SURGERY	Nadene Harwell
34	876713	Concannon	Jeannette	2018-10-01	03PM	MATERNITY	Carmel Mersch
35	876713	Concannon	Jeannette	2018-10-01	11PM	EMERGENCY	Oralia Swallow
36	876713	Concannon	Jeannette	2018-10-05	07AM	PEDIATRICS	Remona Locke
37	876713	Concannon	Jeannette	2018-10-05	11PM	PEDIATRICS	Remona Locke
38	878397	Whitmarsh	Mikel	2018-09-24	11PM	RADIOLOGY	Pamella Deines
39	878397	Whitmarsh	Mikel	2018-09-24	11PM	RADIOLOGY	Pamella Deines
40	881626	Sisk	Ernesto	2018-09-28	07AM	RADIOLOGY	Pamella Deines
41	881626	Sisk	Ernesto	2018-09-28	11PM	EMERGENCY	Oralia Swallow
42	883069	Deines	Pamella	2018-09-26	11PM	EMERGENCY	Oralia Swallow
43	883069	Deines	Pamella	2018-09-26	11PM	SURGERY	Nadene Harwell
44	883069	Deines	Pamella	2018-09-28	07AM	ONCOLOGY	Junita Loranger
45	883069	Deines	Pamella	2018-09-28	07AM	PEDIATRICS	Remona Locke
46	883069	Deines	Pamella	2018-09-30	07AM	SURGERY	Nadene Harwell
47	883069	Deines	Pamella	2018-09-30	07PM	MATERNITY	Carmel Mersch
48	883367	Clyburn	Jeffry	2018-09-27	03PM	RADIOLOGY	Pamella Deines
49	883367	Clyburn	Jeffry	2018-09-27	11PM	SURGERY	Nadene Harwell

	empid	lastname	firstname	Overbooked_Date	sfrom	dept_name	manager
50	883367	Clyburn	Jeffry	2018-10-01	07AM	RADIOLOGY	Pamella Deines
51	883367	Clyburn	Jeffry	2018-10-01	03PM	RADIOLOGY	Pamella Deines
52	883367	Clyburn	Jeffry	2018-10-01	03PM	MATERNITY	Carmel Mersch
53	883367	Clyburn	Jeffry	2018-10-01	11PM	SURGERY	Nadene Harwell
54	886190	Ault	Rosendo	2018-09-25	03PM	PEDIATRICS	Remona Locke
55	886190	Ault	Rosendo	2018-09-25	03PM	ONCOLOGY	Junita Loranger
56	886190	Ault	Rosendo	2018-10-04	03PM	EMERGENCY	Oralia Swallow
57	886190	Ault	Rosendo	2018-10-04	11PM	RADIOLOGY	Pamella Deines
58	891046	Hessler	Latina	2018-10-05	07AM	INTENSIVE CARE	Colby Witham
59	891046	Hessler	Latina	2018-10-05	11PM	PEDIATRICS	Remona Locke
60	898057	Blagg	Lorenzo	2018-10-05	07AM	ONCOLOGY	Junita Loranger
61	898057	Blagg	Lorenzo	2018-10-05	03PM	RADIOLOGY	Pamella Deines
62	900486	Ober	Shalanda	2018-09-30	07PM	CARDIOLOGY	Marylin Mumaw
63	900486	Ober	Shalanda	2018-09-30	07PM	CARDIOLOGY	Marylin Mumaw
64	913891	Warn	Loraine	2018-09-27	07AM	INTENSIVE CARE	Colby Witham
65	913891	Warn	Loraine	2018-09-27	03PM	RADIOLOGY	Pamella Deines
66	919675	Cermak	Vern	2018-09-25	03PM	EMERGENCY	Oralia Swallow
67	919675	Cermak	Vern	2018-09-25	11PM	ONCOLOGY	Junita Loranger
68	919675	Cermak	Vern	2018-09-26	07AM	EMERGENCY	Oralia Swallow
69	919675	Cermak	Vern	2018-09-26	11PM	CARDIOLOGY	Marylin Mumaw
70	919675	Cermak	Vern	2018-09-27	03PM	RADIOLOGY	Pamella Deines
71	919675	Cermak	Vern	2018-09-27	03PM	CARDIOLOGY	Marylin Mumaw
72	919675	Cermak	Vern	2018-09-30	07AM	MATERNITY	Carmel Mersch
73	919675	Cermak	Vern	2018-09-30	07AM	EMERGENCY	Oralia Swallow
74	919675	Cermak	Vern	2018-09-30	07PM	ONCOLOGY	Junita Loranger
75	919675	Cermak	Vern	2018-10-07	07PM	EMERGENCY	Oralia Swallow
76	919675	Cermak	Vern	2018-10-07	07PM	EMERGENCY	Oralia Swallow
77	919847	Gabel	Huong	2018-09-25	03PM	MATERNITY	Carmel Mersch
78	919847	Gabel	Huong	2018-09-25	11PM	ONCOLOGY	Junita Loranger
79	919847	Gabel	Huong	2018-10-01	07AM	EMERGENCY	Oralia Swallow
80	919847	Gabel	Huong	2018-10-01	11PM	ONCOLOGY	Junita Loranger
81	919847	Gabel	Huong	2018-10-06	07AM	SURGERY	Nadene Harwell
82	919847	Gabel	Huong	2018-10-06	07AM	CARDIOLOGY	Marylin Mumaw
83	919847	Gabel	Huong	2018-10-06	07PM	MATERNITY	Carmel Mersch
84	921331	Beaver	Viva	2018-09-25	07AM	EMERGENCY	Oralia Swallow
85	921331	Beaver	Viva	2018-09-25	03PM	MATERNITY	Carmel Mersch
86	921331	Beaver	Viva	2018-09-26	07AM	ONCOLOGY	Junita Loranger
87	921331	Beaver	Viva	2018-09-26	07AM	ONCOLOGY	Junita Loranger
88	921331	Beaver	Viva	2018-09-26	07AM	INTENSIVE CARE	Colby Witham
89	921331	Beaver	Viva	2018-10-07	07PM	EMERGENCY	Oralia Swallow
90	921331	Beaver	Viva	2018-10-07	07PM	PEDIATRICS	Remona Locke
91	921331	Beaver	Viva	2018-10-07	07PM	MATERNITY	Carmel Mersch
92	929086	Noonkester	Asa	2018-09-25	07AM	INTENSIVE CARE	Colby Witham
93	929086	Noonkester	Asa	2018-09-25	07AM	MATERNITY	Carmel Mersch
94	929086	Noonkester	Asa	2018-09-27	07AM	CARDIOLOGY	Marylin Mumaw
95	929086	Noonkester	Asa	2018-09-27	03PM	MATERNITY	Carmel Mersch
96	929086	Noonkester	Asa	2018-09-28	07AM	RADIOLOGY	Pamella Deines
97	929086	Noonkester	Asa	2018-09-28	07AM	RADIOLOGY	Pamella Deines
98	937181	Hauck	Kenton	2018-09-27	07AM	PEDIATRICS	Remona Locke
99	937181	Hauck	Kenton	2018-09-27	03PM	MATERNITY	Carmel Mersch

	empid	lastname	firstname	Overbooked_Date	sfrom	dept_name	manager
100	937181	Hauck	Kenton	2018-10-02	03PM	CARDIOLOGY	Marylin Mumaw
101	937181	Hauck	Kenton	2018-10-02	11PM	PEDIATRICS	Remona Locke
102	939148	Liberto	Delta	2018-09-24	07AM	EMERGENCY	Oralia Swallow
103	939148	Liberto	Delta	2018-09-24	07AM	EMERGENCY	Oralia Swallow
104	939148	Liberto	Delta	2018-09-26	07AM	SURGERY	Nadene Harwell
105	939148	Liberto	Delta	2018-09-26	11PM	EMERGENCY	Oralia Swallow
106	939148	Liberto	Delta	2018-10-02	07AM	MATERNITY	Carmel Mersch
107	939148	Liberto	Delta	2018-10-02	07AM	INTENSIVE CARE	Colby Witham
108	940824	Loranger	Junita	2018-09-25	11PM	MATERNITY	Carmel Mersch
109	940824	Loranger	Junita	2018-09-25	11PM	ONCOLOGY	Junita Loranger
110	940824	Loranger	Junita	2018-10-02	07AM	MATERNITY	Carmel Mersch
111	940824	Loranger	Junita	2018-10-02	03PM	RADIOLOGY	Pamella Deines
112	940824	Loranger	Junita	2018-10-03	11PM	EMERGENCY	Oralia Swallow
113	940824	Loranger	Junita	2018-10-03	11PM	MATERNITY	Carmel Mersch
114	940824	Loranger	Junita	2018-10-06	07PM	INTENSIVE CARE	Colby Witham
115	940824	Loranger	Junita	2018-10-06	07PM	MATERNITY	Carmel Mersch
116	941663	Nalley	Charlott	2018-09-29	07AM	SURGERY	Nadene Harwell
117	941663	Nalley	Charlott	2018-09-29	07AM	ONCOLOGY	Junita Loranger
118	941663	Nalley	Charlott	2018-09-29	07PM	RADIOLOGY	Pamella Deines
119	941663	Nalley	Charlott	2018-10-05	03PM	ONCOLOGY	Junita Loranger
120	941663	Nalley	Charlott	2018-10-05	11PM	PEDIATRICS	Remona Locke
121	943023	Hinze	Mark	2018-09-28	07AM	PEDIATRICS	Remona Locke
122	943023	Hinze	Mark	2018-09-28	03PM	ONCOLOGY	Junita Loranger
123	943023	Hinze	Mark	2018-10-05	07AM	INTENSIVE CARE	Colby Witham
124	943023	Hinze	Mark	2018-10-05	11PM	MATERNITY	Carmel Mersch
125	943023	Hinze	Mark	2018-10-07	07AM	ONCOLOGY	Junita Loranger
126	943023	Hinze	Mark	2018-10-07	07AM	EMERGENCY	Oralia Swallow
127	945520	Crigger	Maudie	2018-10-01	07AM	SURGERY	Nadene Harwell
128	945520	Crigger	Maudie	2018-10-01	11PM	RADIOLOGY	Pamella Deines
129	945520	Crigger	Maudie	2018-10-03	07AM	EMERGENCY	Oralia Swallow
130	945520	Crigger	Maudie	2018-10-03	03PM	INTENSIVE CARE	Colby Witham
131	945540	Stager	Roderick	2018-10-04	07AM	ONCOLOGY	Junita Loranger
132	945540	Stager	Roderick	2018-10-04	11PM	CARDIOLOGY	Marylin Mumaw
133	945540	Stager	Roderick	2018-10-07	07AM	EMERGENCY	Oralia Swallow
134	945540	Stager	Roderick	2018-10-07	07PM	PEDIATRICS	Remona Locke
135	948273	Petties	Rachelle	2018-09-28	07AM	EMERGENCY	Oralia Swallow
136	948273	Petties	Rachelle	2018-09-28	07AM	CARDIOLOGY	Marylin Mumaw
137	953807	Quarles	Timmy	2018-10-02	07AM	MATERNITY	Carmel Mersch
138	953807	Quarles	Timmy	2018-10-02	11PM	INTENSIVE CARE	Colby Witham
139	953817	Harwell	Nadene	2018-09-25	03PM	PEDIATRICS	Remona Locke
140	953817	Harwell	Nadene	2018-09-25	11PM	CARDIOLOGY	Marylin Mumaw
141	957335	Mullally	Vennie	2018-09-27	07AM	PEDIATRICS	Remona Locke
142	957335	Mullally	Vennie	2018-09-27	03PM	ONCOLOGY	Junita Loranger
143	957335	Mullally	Vennie	2018-09-29	07AM	INTENSIVE CARE	Colby Witham
144	957335	Mullally	Vennie	2018-09-29	07PM	CARDIOLOGY	Marylin Mumaw
145	957335	Mullally	Vennie	2018-10-01	03PM	RADIOLOGY	Pamella Deines
146	957335	Mullally	Vennie	2018-10-01	11PM	RADIOLOGY	Pamella Deines
147	964932	Harps	Gigi	2018-10-03	03PM	MATERNITY	Carmel Mersch
148	964932	Harps	Gigi	2018-10-03	11PM	ONCOLOGY	Junita Loranger
149	967577	Swallow	Oralia	2018-09-24	07AM	SURGERY	Nadene Harwell

	empid	lastname	firstname	Overbooked_Date	sfrom	dept_name	manager
150	967577	Swallow	Oralia	2018-09-24	03PM	MATERNITY	Carmel Mersch
151	967577	Swallow	Oralia	2018-10-03	07AM	MATERNITY	Carmel Mersch
152	967577	Swallow	Oralia	2018-10-03	11PM	SURGERY	Nadene Harwell
153	967577	Swallow	Oralia	2018-10-07	07AM	ONCOLOGY	Junita Loranger
154	967577	Swallow	Oralia	2018-10-07	07PM	ONCOLOGY	Junita Loranger
155	971559	Mumaw	Marylin	2018-09-30	07AM	ONCOLOGY	Junita Loranger
156	971559	Mumaw	Marylin	2018-09-30	07PM	ONCOLOGY	Junita Loranger
157	979951	Berryman	Sabrina	2018-10-04	03PM	EMERGENCY	Oralia Swallow
158	979951	Berryman	Sabrina	2018-10-04	11PM	INTENSIVE CARE	Colby Witham
159	979951	Berryman	Sabrina	2018-10-04	11PM	ONCOLOGY	Junita Loranger
160	979951	Berryman	Sabrina	2018-10-06	07AM	PEDIATRICS	Remona Locke
161	979951	Berryman	Sabrina	2018-10-06	07AM	MATERNITY	Carmel Mersch
162	979951	Berryman	Sabrina	2018-10-06	07AM	ONCOLOGY	Junita Loranger
163	992463	Goode	Rex	2018-09-25	11PM	ONCOLOGY	Junita Loranger
164	992463	Goode	Rex	2018-09-25	11PM	SURGERY	Nadene Harwell
165	992463	Goode	Rex	2018-09-26	07AM	MATERNITY	Carmel Mersch
166	992463	Goode	Rex	2018-09-26	03PM	MATERNITY	Carmel Mersch
167	995561	Labarre	Virgina	2018-09-24	11PM	EMERGENCY	Oralia Swallow
168	995561	Labarre	Virgina	2018-09-24	11PM	SURGERY	Nadene Harwell
169	995561	Labarre	Virgina	2018-10-01	07AM	INTENSIVE CARE	Colby Witham
170	995561	Labarre	Virgina	2018-10-01	11PM	PEDIATRICS	Remona Locke
171	995636	Pye	Deanna	2018-09-30	07AM	MATERNITY	Carmel Mersch
172	995636	Pye	Deanna	2018-09-30	07AM	EMERGENCY	Oralia Swallow
173	995636	Pye	Deanna	2018-10-01	07AM	CARDIOLOGY	Marylin Mumaw
174	995636	Pye	Deanna	2018-10-01	11PM	RADIOLOGY	Pamella Deines
175	995636	Pye	Deanna	2018-10-05	07AM	INTENSIVE CARE	Colby Witham
176	995636	Pye	Deanna	2018-10-05	11PM	MATERNITY	Carmel Mersch
177	995636	Pye	Deanna	2018-10-07	07PM	RADIOLOGY	Pamella Deines
178	995636	Pye	Deanna	2018-10-07	07PM	INTENSIVE CARE	Colby Witham

2. GENERATE A REPORT OF ALL OF THE FULL TIME PEOPLE WHO ARE WORKING MORE THAN 80 HOURS, AND HOW MUCH OVERTIME THEY ARE SCHEDULED FOR.

```
In [15]: query = '''
        SELECT CONCAT(e.firstname, ' ', e.lastname) AS fullname,
               CAST(sum(s.shift_length)-80 AS INT) AS scheduled_overtime
        FROM schedule AS s, employees AS e
        WHERE s.empid=e.empid
        AND e.ftpt = 'ft'
        GROUP BY s.empid
        HAVING scheduled_overtime >0
        ORDER BY scheduled_overtime DESC, fullname ASC;
        '''
```

```
df = pd.read_sql(query, dsl.db)
```

	fullname	scheduled_overtime
0	Vern Cermak	60
1	Deanna Pye	48
2	Jeffry Clyburn	44
3	Remona Locke	36
4	Charlott Nalley	32
5	Manie Zaragoza	32
6	Junita Loranger	24
7	Imelda Catanzaro	20
8	Kenton Hauck	20
9	Asa Noonkester	16
10	Mark Hinze	16
11	Shalanda Ober	16
12	Barton Mcclelland	12
13	Oralia Swallow	8
14	Virgina Labarre	8
15	Marylin Mumaw	4