

Лабораторная работа №3. Измерение и анализ Интернет-соединения с хранилищем данных в виде SQL базы данных.

Контрольные вопросы

1. Что такое csvsql, как она работает?
 - **csvsql** функции программы csvkit.
 - Метод **csvsql** позволяет легко импортировать данные в базу данных sql, создавая в этой базе данных соответствующую структуру, которая будет содержать эти данные;

```
Entrée [4]: # Code cell 4
# Removing database if already existing
!test -e InternetSpeed.db && rm InternetSpeed.db
# Importing the LA_wifi_speed_UK.csv file into a sql database file InternetSpeed.db
!csvsql --db sqlite:///InternetSpeed.db --insert ./LA_wifi_speed_UK.csv
# This will take a while
```

2. Можно ли объединить таблицы в базе данных, как это сделать?
 - Да! Можно объединить таблицы данных;
 - Приступить к объединению таблиц из базы данных :
 - ✗ Выберите две таблицы, которые разделяют поле (две таблицы, которые имеют общее поле);
 - ✗ Использовать общий ключ для этих двух таблиц для выполнения соединения;
 - ✗ Сравнение по строкам двух таблиц выполняется SQL. Затем SQL выполняет конкатенацию строк с одинаковым значением ключа;

```
Entrée [38]: # Code cell 27
query = 'SELECT * FROM average_speed JOIN LA_population ON LA_population."LA_code"=average_speed.Area;'
cur.execute(query)
k = 0
for row in cur:
    if k>10:
        break;
    # print ?
# SOLUTION:
print(row)
k+=1

(0, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(1, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(2, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(3, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(4, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(5, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(6, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(7, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(8, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(9, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
(10, 'E06000015', 54.93447752674741, 22.38706665857376, 8.95783448218355, 'Derby', 'E06000015', 248752.0)
```

3. Какой метод Pandas позволяет добавлять информацию в базу данных?
 - Чтобы добавить информацию в базу данных, Pandas помещает содержимое базы данных в DataFrame. См. пример в коде ячейки 19 :
« `df = df.append(pd.Series(tmp_list, index=new_columns), ignore_index=True)` »

```

Entrée [25]: # Code cell 19
# this will take a while...
for i in range(len(columns)-1): # EDL : replace xrange with range
    tmp_list = []
    tmp_list.append(columns[i])
    for s in suffix.keys():
        # query = ??
        # SOLUTION:
        query = 'SELECT AVG("{}{}") FROM LA_wifi_speed_UK'.format(columns[i], s)

        cur.execute(query)

        mean = cur.fetchone()
        tmp_list.append(mean[0])
    # Append the columns to the empty DataFrame
    df = df.append(pd.Series(tmp_list, index=new_columns), ignore_index=True)
# Visualize the head of the dataframe here
# SOLUTION:
df.head()

```

```

Out[25]:

```

	Area	Average_p	Average_d	Average_u
0	E06000015	54.934478	22.387067	8.957834
1	E06000015	54.934478	22.387067	8.957834
2	E06000015	54.934478	22.387067	8.957834
3	E06000015	54.934478	22.387067	8.957834
4	E06000015	54.934478	22.387067	8.957834

- После выполнения необходимой обработки данных, содержащихся в DataFrame, pandas позволяет перенести содержимое этого DataFrame в базу данных. См. пример в коде ячейки 21 :

« `df.to_sql('average_speed', conn)` »

```

Entrée [32]: # Code cell 21
# Make sure to drop the table first if it already exists
try:
    cur.execute('DROP TABLE average_speed')
except:
    pass

# What is the name of the table you want to save? To what connection?
# Save the dataframe table into a SQL table
# df.to_sql(?,?)

# SOLUTION:
df.to_sql('average_speed', conn)

```

4. Что делает метод fetchall() ?
 - Метод « **fetchall()** » используется для получения всех строк результата запроса.
5. Почему для обработки данных используется библиотека Pandas, можно ли выполнить такую же обработку другим способом?
 - Библиотека pandas используется, потому что она позволяет извлекать данные из базы данных для манипулирования ими в dataframe и вводить их обратно в базу данных.
 - Есть еще один способ извлечения данных из базы данных. Но этот метод не предлагает возможность ввода данных непосредственно в DataFrame, как с библиотекой pandas.

<https://github.com/malko1278/labo-analyse-big-data.git>