

Зачет

28 мая 2021 года
ПИ19-3, ПИ19-4 - 3 подгруппа

5 июня 2021 года
ПИ19-2, ПИ19-3, ПИ19-4 - 2 подгруппа

4 июня 2021 года
ПИ19-4, ПИ19-5 - 4 подгруппа

Выполнила Волкова Татьяна ПИ19-4, подгруппа 3

Вариант: ORM - 2, Core - 1

1 - При помощи Core

In [1]:

```
"""
Напишите запрос для получения списка имен, фамилий, названий городов студентов
и названий университетов, расположенных в городе, где живет студент
"""
from sqlalchemy import (select, create_engine, MetaData, Table)

engine = create_engine('sqlite:///Data/Students_2021.sqlite')
metadata = MetaData()
connection = engine.connect()

city = Table('city', metadata, autoload=True, autoload_with=engine)
exam_marks = Table('exam_marks', metadata, autoload=True, autoload_with=engine)
lecturer = Table('lecturer', metadata, autoload=True, autoload_with=engine)
student = Table('student', metadata, autoload=True, autoload_with=engine)
subj_lect = Table('subj_lect', metadata, autoload=True, autoload_with=engine)
subject = Table('subject', metadata, autoload=True, autoload_with=engine)
university = Table('university', metadata, autoload=True, autoload_with=engine)
```

In [2]:

```
columns = [student.c.name, student.c.surname, student.c.city, university.c.univ_name]
s = select(columns)
s = s.select_from(student.join(university))
#s = s.where(student.c.univ_id == university.c.univ_id)
s = s.where(student.c.city == university.c.city)
result = connection.execute(s).fetchall()
for row in result:
    print(row)
```

```
('Вадим', 'Сидоров', 'Москва', 'МГУ')
('Павел', 'Котов', 'Белгород', 'ВГУ')
('Вадим', 'Белкин', 'Воронеж', 'ВГУ')
```

2 - При помощи ORM

In [3]:

```
"""
Напишите запрос для получения списка фамилий студентов
с полученными ими оценками и названиями предметов.
Отсортируйте результат по фамилии студента.
"""

from sqlalchemy.ext.automap import automap_base
from sqlalchemy import create_engine, MetaData, Table
```

```
from sqlalchemy import create_engine
```

```
Base = automap_base()  
engine = create_engine('sqlite:///Data/Students_2021.sqlite')  
Base.prepare(engine, reflect=True)  
Base.classes.keys()
```

Out[3]:

```
['city',  
 'exam_marks',  
 'student',  
 'university',  
 'subject',  
 'lecturer',  
 'subj_lect']
```

In [4]:

```
from sqlalchemy.orm import Session  
session = Session(engine)
```

```
city, exam_marks, student, university, subject, lecturer, subj_lect = Base.classes.values()
```

In [5]:

```
from pprint import pprint  
from sqlalchemy import and_  
import pandas as pd
```

In [6]:

```
#без join  
q = session.query(student.surname, exam_marks.mark, subject.subj_name)  
q = q.filter(and_(student.student_id == exam_marks.student_id, subject.subj_id == exam_marks.subj_id))  
q = q.order_by(student.surname)  
q = q.all()  
pd.DataFrame(q)
```

Out[6]:

	surname	mark	subj_name
0	Белкин	3.0	Web-дизайн
1	Белкин	3.0	ОБЖ
2	Белкин	3.0	Финансовые рынки
3	Белкин	5.0	Информатика
4	Белкин	4.0	Программирование
5	Белкин	NaN	Анализ данных
6	Зайцева	4.0	ОБЖ
7	Зайцева	5.0	Информатика
8	Зайцева	5.0	Программирование
9	Иванов	5.0	Анализ данных
10	Иванов	3.0	Математика
11	Иванов	4.0	Физика
12	Калашников	3.0	Программирование
13	Калашников	2.0	Информатика
14	Котов	4.0	Программирование
15	Котов	5.0	Web-дизайн
16	Котов	3.0	Анализ данных
17	Котов	4.0	Финансовые рынки
18	Котов	3.0	Информатика

19	Мельникова	3.0	Программирование
20	Мельникова	3.0	История
21	Мельникова	3.0	Анализ данных
22	Мельникова	3.0	Информатика
23	Мельникова	5.0	ОБЖ
24	Никитин	4.0	ОБЖ
25	Никитин	4.0	Программирование
26	Никитин	4.0	Информатика
27	Никитин	3.0	Web-дизайн
28	Никитин	3.0	Анализ данных
29	Никитин	5.0	Финансовые рынки
30	Семенова	4.0	Физкультура
31	Семенова	4.0	Английский
32	Семенова	3.0	Финансовые рынки
33	Сергеева	3.0	Анализ данных
34	Сергеева	5.0	Информатика
35	Сергеева	5.0	Программирование
36	Сергеева	5.0	ОБЖ
37	Сергеева	3.0	Web-дизайн
38	Сергеева	2.0	Финансовые риски корпоративного инвестирования
39	Синицин	3.0	Информатика
40	Синицин	4.0	Программирование

In [7]:

```
# c join
j = session.query(student.surname, exam_marks.mark, subject.subj_name)
j = j.join(exam_marks, student.student_id == exam_marks.student_id)
j = j.join(subject)
j = j.order_by(student.surname)
j = j.all()
pd.DataFrame(j)
```

Out [7]:

	surname	mark	subj_name
0	Белкин	3.0	Web-дизайн
1	Белкин	3.0	ОБЖ
2	Белкин	3.0	Финансовые рынки
3	Белкин	5.0	Информатика
4	Белкин	4.0	Программирование
5	Белкин	NaN	Анализ данных
6	Зайцева	4.0	ОБЖ
7	Зайцева	5.0	Информатика
8	Зайцева	5.0	Программирование
9	Иванов	5.0	Анализ данных
10	Иванов	3.0	Математика
11	Иванов	4.0	Физика
12	Калашников	3.0	Программирование
13	Калашников	2.0	Информатика
14	Котов	4.0	Программирование
15	Котов	5.0	Web-дизайн
16	Котов	3.0	Анализ данных

1 /	КОТОВ surname	4.0 mark	Финансовые рынки subj_name
18	Котов	3.0	Информатика
19	Мельникова	3.0	Программирование
20	Мельникова	3.0	История
21	Мельникова	3.0	Анализ данных
22	Мельникова	3.0	Информатика
23	Мельникова	5.0	ОБЖ
24	Никитин	4.0	ОБЖ
25	Никитин	4.0	Программирование
26	Никитин	4.0	Информатика
27	Никитин	3.0	Web-дизайн
28	Никитин	3.0	Анализ данных
29	Никитин	5.0	Финансовые рынки
30	Семенова	4.0	Физкультура
31	Семенова	4.0	Английский
32	Семенова	3.0	Финансовые рынки
33	Сергеева	3.0	Анализ данных
34	Сергеева	5.0	Информатика
35	Сергеева	5.0	Программирование
36	Сергеева	5.0	ОБЖ
37	Сергеева	3.0	Web-дизайн
38	Сергеева	2.0	Финансовые риски корпоративного инвестирования
39	Синицин	3.0	Информатика
40	Синицин	4.0	Программирование

In []: