

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

ПРОГРАММНОЕ ПОДКЛЮЧЕНИЕ К СЕРВЕРУ POSTGRESQL

Отчёт о лабораторной работе № 4
по дисциплине «Базы данных»

Выполнил: студент гр. 431-3

_____ Бекиш Е.П.

«___» _____ 2023 г.

Проверил: ассистент каф. АСУ

_____ Яблонский Я. В.

«___» _____ 2023 г.

Томск 2023

1 Цель лабораторной работы

– Разработать программу, которая соответствует условиям индивидуального задания на любом языке программирования.

2 Описание требований к отчётам

1. Выполняет подключение к серверу СУБД PostgreSQL.
2. Заносит новые данные в одну из таблиц (по выбору студента).
3. Формирует на экране (или в файле) отчет как показано на рисунке

2.1.

УСПЕВАЕМОСТЬ

Год обучения 1

Баранов Алексей Андреевич

<i>Дисциплина</i>	<i>Оценка</i>	<i>Семестр</i>
ПЯВУ	4	1
ВУПЯ	4	1
Теоретические основы куроредения	4	1
Средний балл	4	

Барашкин Алексей Андреевич

<i>Дисциплина</i>	<i>Оценка</i>	<i>Семестр</i>
ПЯВУ	4	1
ВУПЯ	3	2
Средний балл	3,5	

Быков Алексей Андреевич

<i>Дисциплина</i>	<i>Оценка</i>	<i>Семестр</i>
ВУПЯ	5	1
Теоретические основы куроредения	3	1
Меланжевое производство	4	1
Средний балл	4	

Год обучения 2

Воловский Алексей Андреевич

<i>Дисциплина</i>	<i>Оценка</i>	<i>Семестр</i>
ПЯВУ	3	1
Теоретические основы куроредения	5	2
Технология собаководства	3	3
ВУПЯ	4	4
Средний балл	3,75	

Год обучения 3

Гатов Николай Саввич

<i>Дисциплина</i>	<i>Оценка</i>	<i>Семестр</i>
ПЯВУ	5	3
Теоретические основы куроредения	5	3
Меланжевое производство	5	3
Средний балл	5	

Год обучения 4

Батов Иван Константинович

<i>Дисциплина</i>	<i>Оценка</i>	<i>Семестр</i>
ПЯВУ	4	1
ВУПЯ	3	2
Технология собаководства	5	3
Меланжевое производство	3	4
Теоретические основы куроредения	5	5
Технология приготовления яицницы	3	6
Средний балл	3,8	

Рисунок 2.1 – успешный результат индивидуального задания

3 Листинг

```
import psycopg2
originall = {
    'dbname': 'tusun',
    'user': 'postgres',
    'password': '123',
}
connect = psycopg2.connect(**originall)
cursor = connect.cursor()
connect.autocommit = True
query = """
insert into room (rnum, tnum)
values
('000', '000000'),
('001', '000001'),
('002', '000002'),
('003', '000003')
;
insert into student (snum, snam, ann, stel, rnum)
values
('000000', 'Баранов Алексей Андреевич', '1', '9130000000', '000'),
('000001', 'Барашкин Алексей Андреевич', '1', '9130000001', '000'),
('000002', 'Быков Алексей Андреевич', '1', '9130000002', '000'),
('000003', 'Воловский Алексей Андреевич', '2', '9130000003', '001'),
('000004', 'Гатов Николай Саввич', '3', '9130000004', '002'),
('000005', 'Батов Иван Константинович', '4', '9130000005', '003')
;
insert into "pr-dis" (pnam, dnam)
values
('Белоусова София Романовна', 'ПЯВУ'),
```

('Морозова Майя Андреевна', 'ВУПЯ'),
('Медведев Лев Юрьевич', 'Теоритические основы куроедения'),
('Попов Михаил Кириллович', 'Меланжевое производство'),
('Кулаков Роман Родионович', 'Технология собаковыгуливания'),
('Мальцев Матвей Ильич', 'Технология приготовления яичницы')

;

insert into exams (snum, sem, pnam, ball)

values

('000000', '1', 'Белоусова София Романовна', '4'),
('000000', '1', 'Морозова Майя Андреевна', '4'),
('000000', '1', 'Медведев Лев Юрьевич', '4'),
('000001', '1', 'Белоусова София Романовна', '4'),
('000001', '2', 'Морозова Майя Андреевна', '3'),
('000002', '1', 'Морозова Майя Андреевна', '5'),
('000002', '1', 'Медведев Лев Юрьевич', '3'),
('000002', '1', 'Попов Михаил Кириллович', '4'),
('000003', '1', 'Белоусова София Романовна', '3'),
('000003', '2', 'Медведев Лев Юрьевич', '5'),
('000003', '3', 'Кулаков Роман Родионович', '3'),
('000003', '4', 'Морозова Майя Андреевна', '4'),
('000004', '3', 'Белоусова София Романовна', '5'),
('000004', '3', 'Медведев Лев Юрьевич', '5'),
('000004', '3', 'Попов Михаил Кириллович', '5'),
('000005', '1', 'Белоусова София Романовна', '4'),
('000005', '2', 'Морозова Майя Андреевна', '3'),
('000005', '3', 'Кулаков Роман Родионович', '5'),
('000005', '4', 'Попов Михаил Кириллович', '3'),
('000005', '5', 'Медведев Лев Юрьевич', '5'),
('000005', '6', 'Мальцев Матвей Ильич', '3')

;"'"

```

cursor.execute(query)

def foo(snum):

    query = "select * from "pr-dis";"
    cursor.execute(query)
    pr_dis = cursor.fetchall()
    pr_dis_dict = {pr_dis[i][0]: pr_dis[i][1] for i in range(len(pr_dis))}
    query = f"select sem, pnam, ball from exams where snum='{snum}';"
    cursor.execute(query)
    exams = cursor.fetchall()
    query = f"select avg(ball) from exams where snum='{snum}';"
    cursor.execute(query)
    middle = str(cursor.fetchone()[0])
    if sum(list(map(int, middle[2:4]))) == 0:
        middle = int(middle[:middle.index('.')])
    elif middle[3] == '0' or middle[3] != '5':
        middle = middle[:3]
    else:
        middle = middle[:4]
    s = 'Дисциплина\t\t\t\t\tОценка\tСеместр\n'
    for info in exams:
        sem, pnam, ball = info
        s += f'\t\t\t\t\t\t\t {pr_dis_dict[pnam]:35} {ball} {sem:10}\n'
    s += f'\t\t\t\t\t\t\t\t\t\t\t\t\t {"Средний балл":25} {middle}'
    return s

query = "select ann from student;"
cursor.execute(query)
ann = set(cursor.fetchall())
file = open('laba4.txt', 'w')
for year in ann:
    query = f"select * from student where ann={year[0]}"

```

```

cursor.execute(query)
student_info = cursor.fetchall()
print(fГод обучения: {year[0]})
for info in student_info:
    snum, snam, ann, tnum, rnum = info
    print(f{snam}\t\t\t{foo(snum)}\n')
query = """
delete from exams;
delete from student;
delete from room;
delete from "pr-dis";
"""
cursor.execute(query)
cursor.close()
connect.close()

```

4 Описание процесса реализации

В ходе выполнения лабораторной работы будет использоваться язык Python. Создаем файл с расширением .py. Теперь для использования PostgreSQL необходимо импортировать библиотеку psycopg2 как показано на рисунке 4.1.

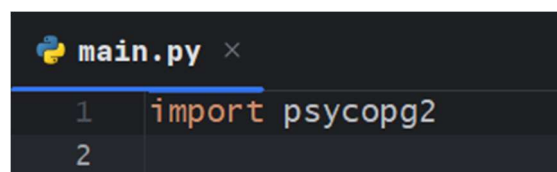


Рисунок 4.1 – импорт библиотеки

Для того, чтобы подключиться к нашей базе данных необходимо использовать метод connect, в котором необходимо указать базу данных, если она существует (по умолчанию подключаемся к postgres), пароль и пользователя (нужно указать postgres). Для удобства создадим словарь, в котором укажем в качестве ключей необходимые название параметров и их

значения. Вообще словарь — неупорядоченная структура данных, которая позволяет хранить пары «ключ — значение».

Пример: `dictionary = {key: value}`

Теперь создаем курсор — область памяти базы, предназначенная для хранения последнего оператора SQL. Иными словами, объект, отвечающий и за отправку запросов, и за получение их результатов. Реальное выполнение команд производится только при выполнении метода `conn.commit()`. Если же надо, чтобы выражения SQL автоматически выполнялись при каждом вызове метода `cursor.execute()`, то можно установить автокоммит с помощью свойства `connection.autocommit`. Где выполнением команды SQL автоматически создается транзакция, в процессе которой можно выполнять различные выражения SQL с помощью методов `execute/execute many` курсора, но для подтверждения их выполнения необходимо вызывать метод `commit()` объекта `connection`.

- `execute(query, vars=None)`: выполняет одну SQL-инструкцию.

Через второй параметр в код SQL можно передать набор параметров в виде списка или словаря

- `executemany(query, vars_list)`: выполняет параметризованное SQL-инструкцию. Через второй параметр принимает наборы значений, которые передаются в выполняемый код SQL.

Результат представлен на рисунке 4.2.

```
3  original1 = {
4      'dbname': 'tuser',
5      'user': 'postgres',
6      'password': '123',
7  }
8
9  connect = psycopg2.connect(**original1)
10 cursor = connect.cursor()
11 connect.autocommit = True
```

Рисунок 4.2 – авторизация

Создадим запрос по добавлению данных в наши созданные таблицы как показано на рисунках 4.3 – 4.4.

```
13 query =
14 insert into room (rnum, tnum)
15 values
16 ('000', '000000'),
17 ('001', '000001'),
18 ('002', '000002'),
19 ('003', '000003')
20 ;
21
22
23 insert into student (snum, snam, ann, stel, rnum)
24 values
25 ('000000', 'Баранов Алексей Андреевич', '1', '9130000000', '000'),
26 ('000001', 'Барашкин Алексей Андреевич', '1', '9130000001', '000'),
27 ('000002', 'Быков Алексей Андреевич', '1', '9130000002', '000'),
28 ('000003', 'Воловский Алексей Андреевич', '2', '9130000003', '001'),
29 ('000004', 'Гатов Николай Саввич', '3', '9130000004', '002'),
30 ('000005', 'Батов Иван Константинович', '4', '9130000005', '003')
31 ;
32
33 insert into "pr-dis" (pnam, dnam)
34 values
35 ('Белоусова София Романовна', 'ПЯВУ'),
36 ('Морозова Майя Андреевна', 'ВУПЯ'),
37 ('Медведев Лев Юрьевич', 'Теоритические основы куроедения'),
38 ('Попов Михаил Кириллович', 'Меланжевое производство'),
39 ('Кулаков Роман Родионович', 'Технология собаководства'),
40 ('Мальцев Матвей Ильич', 'Технология приготовления яичницы')
41 ;
42
43 insert into exams (snum, sem, pnam, ball)
44 values
```

Рисунок 4.3 – добавление данных


```

42
43 insert into exams (snum, sem, pnam, ball)
44 values
45 ('000000', '1', 'Белоусова София Романовна', '4'),
46 ('000000', '1', 'Морозова Майя Андреевна', '4'),
47 ('000000', '1', 'Медведев Лев Юрьевич', '4'),
48
49 ('000001', '1', 'Белоусова София Романовна', '4'),
50 ('000001', '2', 'Морозова Майя Андреевна', '3'),
51
52 ('000002', '1', 'Морозова Майя Андреевна', '5'),
53 ('000002', '1', 'Медведев Лев Юрьевич', '3'),
54 ('000002', '1', 'Попов Михаил Кириллович', '4'),
55
56 ('000003', '1', 'Белоусова София Романовна', '3'),
57 ('000003', '2', 'Медведев Лев Юрьевич', '5'),
58 ('000003', '3', 'Кулаков Роман Родионович', '3'),
59 ('000003', '4', 'Морозова Майя Андреевна', '4'),
60
61 ('000004', '3', 'Белоусова София Романовна', '5'),
62 ('000004', '3', 'Медведев Лев Юрьевич', '5'),
63 ('000004', '3', 'Попов Михаил Кириллович', '5'),
64
65 ('000005', '1', 'Белоусова София Романовна', '4'),
66 ('000005', '2', 'Морозова Майя Андреевна', '3'),
67 ('000005', '3', 'Кулаков Роман Родионович', '5'),
68 ('000005', '4', 'Попов Михаил Кириллович', '3'),
69 ('000005', '5', 'Медведев Лев Юрьевич', '5'),
70 ('000005', '6', 'Мальцев Матвей Ильич', '3')
71 ;
72
73 cursor.execute(query)

```

Рисунок 4.4 – добавление данных

После добавления данных создадим функцию, которая будет выдать необходимую информацию об успеваемости. Так как таблицы STUDENT и EXAMS связаны номером студенческого билета, это будет наша переменная, по которой мы сможем брать информацию из данных таблиц. Но в таблице EXAMS хранятся имена преподавателей, поэтому необходимо сформировать словарь по именам преподавателей, которые ведут этот предмет, чтобы имя преподаватель менялось на название дисциплины. Можно это сделать с помощью кортежа словаря, где ключ будет иметь значение имя преподавателя, а значение имеет дисциплину. Результат представлен на рисунке 4.5.

```

1 usage
def foo(snum):
77     query = '''select * from "pr-dis";'''
78     cursor.execute(query)
79     pr_dis = cursor.fetchall()
80     pr_dis_dict = {pr_dis[i][0]: pr_dis[i][1] for i in range(len(pr_dis))}
81

```

Рисунок 4.5 – словарь дисциплин

Далее нам необходимо сделать запрос в таблицу EXAMS по студенческому билету. Выбрать информацию о экзаменаторе, семестре, в котором была сдана данная дисциплина и полученная за данную дисциплину оценка (рис. 4.6).

```

82     query = f'''select sem, pnam, ball from exams where snum='{snum}';'''
83     cursor.execute(query)
84     exams = cursor.fetchall()
85

```

Рисунок 4.6 – информацию о экзамене

Теперь нам необходимо получить среднее значение за все сданные дисциплины. Это можно сделать с помощью агрегатной функции AVG представленной на рисунке 4.7.

```

86     query = f'''select avg(ball) from exams where snum='{snum}';'''
87     cursor.execute(query)
88     middle = str(cursor.fetchone()[0])

```

Рисунок 4.7 – информацию о экзамене

После полученного нами среднего числа у нас возникает формат decimal, то есть десятичное число, которая содержит в себе погрешности с шестнадцатью нулей, и чтобы от них избавиться конвертируем decimal в строку. Проверим, что если наши два значения после запятой это два нуля, то мы конвертируем в число, если же второй знак после запятой ноль, либо же не пять, то делаем срез до него, в противном случае срезаем до второго знака включительно (рис. 4.8).


```

105 query = '''select ann from student;'''
106 cursor.execute(query)
107 ann = set(cursor.fetchall())
108
109 for year in ann:
110     query = f'''select snum, sname from student where ann={year[0]}'''
111     cursor.execute(query)
112     student_info = cursor.fetchall()
113     print(f'Год обучения:{year[0]}')
114     for info in student_info:
115         snum, sname = info
116         print(f'{sname}\t\t\t{foo(snum)}\n')
117
118 query = """
119 delete from exams;
120 delete from student;
121 delete from room;
122 delete from "pr-dis";
123 """
124
125 cursor.execute(query)
126
127 cursor.close()
128 connect.close()
129

```

Рисунок 4.10 – вывод необходимой информации

Результат работы все программы представлен на рисунке 4.11.

Год обучения:1				
Баранов Алексей Андреевич	Дисциплина	Оценка	Семестр	
	ПЯВУ	4	1	
	ВУПЯ	4	1	
	Теоритические основы куроедения	4	1	
	Средний балл		4	
Барашкин Алексей Андреевич				
	Дисциплина	Оценка	Семестр	
	ПЯВУ	4	1	
	ВУПЯ	3	2	
	Средний балл		3.5	
Быков Алексей Андреевич				
	Дисциплина	Оценка	Семестр	
	ВУПЯ	5	1	
	Теоритические основы куроедения	3	1	
	Меланжевое производство	4	1	
	Средний балл		4	
Год обучения:2				
Воловский Алексей Андреевич	Дисциплина	Оценка	Семестр	
	ПЯВУ	3	1	
	Теоритические основы куроедения	5	2	
	Технология собаковыгуливания	3	3	
	ВУПЯ	4	4	
	Средний балл		3.75	
Год обучения:3				
Гатов Николай Саввич	Дисциплина	Оценка	Семестр	
	ПЯВУ	5	3	
	Теоритические основы куроедения	5	3	
	Меланжевое производство	5	3	
	Средний балл		5	
Год обучения:4				
Батов Иван Константинович	Дисциплина	Оценка	Семестр	
	ПЯВУ	4	1	
	ВУПЯ	3	2	
	Технология собаковыгуливания	5	3	
	Меланжевое производство	3	4	
	Теоритические основы куроедения	5	5	
	Технология приготовления яичницы	3	6	
	Средний балл		3.8	

Рисунок 4.11 – успешное выполнение программы

5 Выводы

В ходе выполнения лабораторной работы я научился работать с языком Python, подключаться через библиотеку к базе данных, получать данные из таблиц, заносить данные в таблицы.