Задание 1

```python
import json
import psycopg2

path = "C:/Users/andre/OneDrive/Рабочий стол/Схемы БД/лаба
3/conf.json"

with open(path,'r') as file:
    conf = json.load(file)

closeable = []

try:
    connection = psycopg2.connect(
        **conf['server'][0]
    )
    closeable.append(connection)

    print("Connection established")
    cursor = connection.cursor()

    closeable.append(cursor)
    print("Cursor established")
    cursor.execute(
        "CREATE OR REPLACE FUNCTION reverse ("
        " msg text"
        ") returns text as $$"
        "return msg[::-1]"
        "$$ language plpython3u;")

    record = cursor.fetchall()
    print(record)
except Exception as e:
    print('Error:', e)
finally:
    if bool(closeable):
        _ = [*map(lambda o:o.close(), closeable)]

print("Connection closed")
```
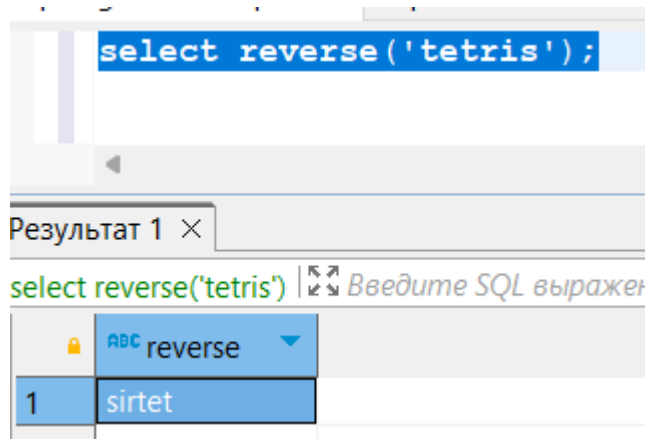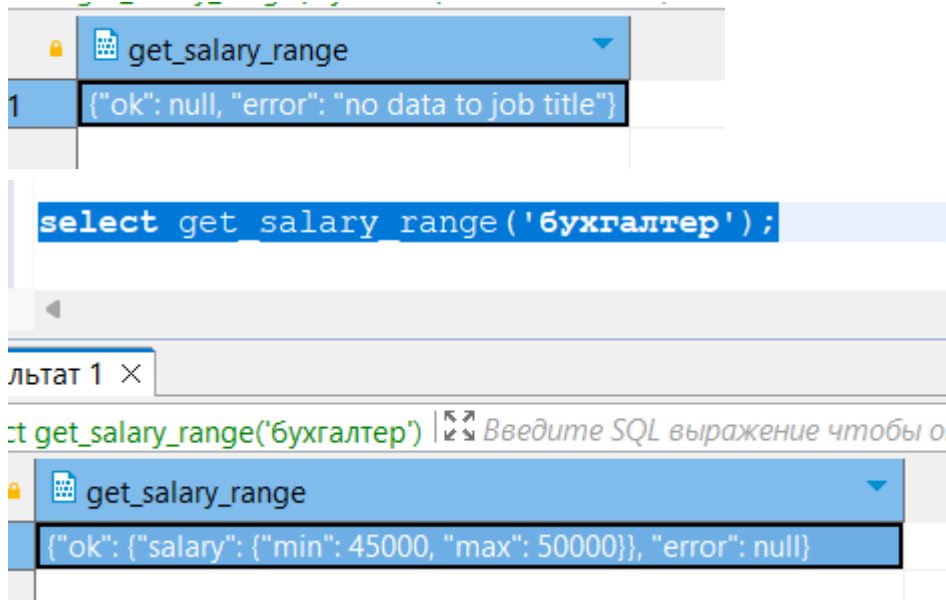
```
select reverse('tetris');
```

Результат 1 ✕

select reverse('tetris') ⤢ *Введите SQL выражен*

🔒	ᴬᴮᶜ reverse ▼
1	sirtet

Задание 2

Т.к. код для 2 задания будет отличаться только тем, что именно мы поместим в cursor.execute(…), далее будет описана только та часть, которая будет помещена вместо … ___

```
CREATE OR REPLACE FUNCTION get_salary_range(job_title TEXT)
RETURNS JSON AS $$
import json
result = {}

try:
    query = plpy.prepare("SELECT pos_id FROM positions WHERE
positions.name = $1", ["text"])
    if len(query.execute([job_title]))!=0:
        pos_id = query.execute([job_title])[0]["pos_id"]
        query = plpy.prepare(f"SELECT MIN(salary), MAX(salary)
FROM employees WHERE pos_id = $1", ["int"])
        row = query.execute([pos_id])[0]
        if row['min'] is not 'null' and row['max'] is not
'null':
                result['ok'] = {'salary': {'min': row['min'],
'max': row['max']}}
                result['error'] = None
        else:
                result['ok'] = None
                result['error'] = 'data not found'
    else:
        result['ok'] = None
        result['error'] = 'no data to job title'
except Exception as e:
    result['ok'] = None
    result['error'] = str(e)

return json.dumps(result)
$$ LANGUAGE plpython3u;
```

get_salary_range

`{"ok": null, "error": "no data to job title"}`

```sql
select get_salary_range('бухгалтер');
```

льтат 1 ✕

t get_salary_range('бухгалтер') ⤢ Введите SQL выражение чтобы о.

get_salary_range

`{"ok": {"salary": {"min": 45000, "max": 50000}}, "error": null}`

Задание 3
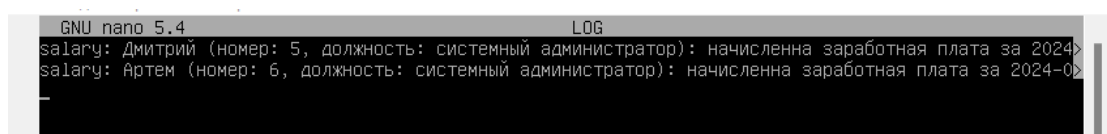
```sql
CREATE Or replace FUNCTION handler()
RETURNS TRIGGER AS $$

from datetime import datetime

query = plpy.prepare("select name from positions where pos_id =
$1",['int'])
pos = plpy.execute(query,[TD['old']['pos_id']])[0]
fn = f"salary: {TD['old']['name']} (номер: {TD['old']
['emp_id']}, должность: {pos['name']})"
fn+= f": начисленна заработная плата за {datetime.now().date()}
в размере {TD['new']['salary']}\n"
with open ("/tmp/LOG","a") as fp:
    fp.write(fn)

$$ language plpython3u;

create trigger salary_add
after UPDATE OF salary ON employees
for each row
EXECUTE FUNCTION handler();
```

```sql
update employees set salary = salary + 10000 where emp_id = 6;
```

```
GNU nano 5.4                              LOG
salary: Дмитрий (номер: 5, должность: системный администратор): начисленна заработная плата за 2024
salary: Артем (номер: 6, должность: системный администратор): начисленна заработная плата за 2024-0
```

Задание 4

```
create or replace function vacancy_message()
returns trigger as $$
if TD['event'] == 'DELETE':
    query = plpy.prepare('select name from positions where
pos_id = $1',['int'])
    pos = plpy.execute(query,[TD['old']['pos_id']])[0]
    query = plpy.prepare('select name from department where
depart_id = $1',['int'])
    dep = plpy.execute(query,[TD['old']['depart_id']])[0]
    fn =f"job: открыта вакансия {pos['name']} в отделе
{dep['name']};\n заработная плата {TD['old']['salary']}\n"
else:
    query = plpy.prepare('select name from positions where
pos_id = $1',['int'])
    pos = plpy.execute(query,[TD['new']['pos_id']])[0]
    query = plpy.prepare('select name from department where
depart_id = $1',['int'])
    dep = plpy.execute(query,[TD['new']['depart_id']])[0]
    fn =f"job: открыта вакансия {pos['name']} в отделе
{dep['name']};\n заработная плата {TD['new']['salary']}\n"
with open ("/tmp/LOG","a") as fp:
    fp.write(fn)
$$ language plpython3u;

create trigger free_vacancy
after insert or delete on employees
for each row
execute function vacancy_message();
```

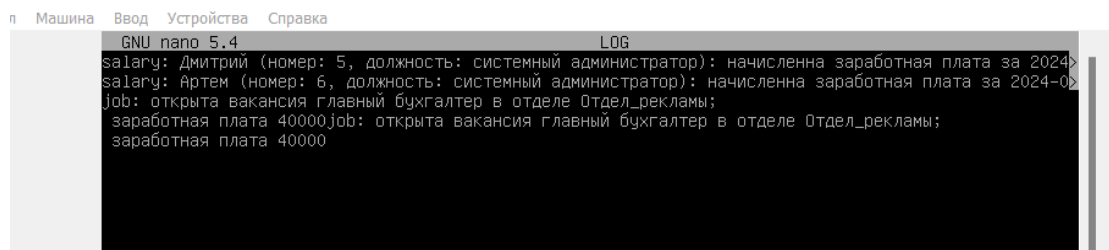Сначала добавил вакансию через insert затем удалил ее

```
insert into employees(salary,depart_id,pos_id) values
(40000,2,4)

update employees set name = 'Александр' where name is null;

delete from employees where name = 'Александр' and emp_id>15;
```

```
n  Машина  Ввод  Устройства  Справка
     GNU nano 5.4                                LOG
  salary: Дмитрий (номер: 5, должность: системный администратор): начисленна заработная плата за 2024
  salary: Артем (номер: 6, должность: системный администратор): начисленна заработная плата за 2024-0
  job: открыта вакансия главный бухгалтер в отделе Отдел_рекламы;
   заработная плата 40000job: открыта вакансия главный бухгалтер в отделе Отдел_рекламы;
   заработная плата 40000
```

- в представленном выше коде перенос на новую строку добавил

In [ ]: