# Отчет по лабораторной работе 4 Панфилов Группа 5

Задание 1

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 ("

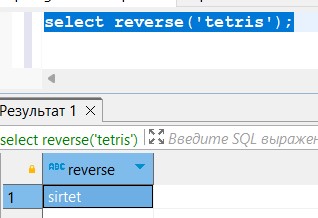
" msg text"

") 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")



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']}}

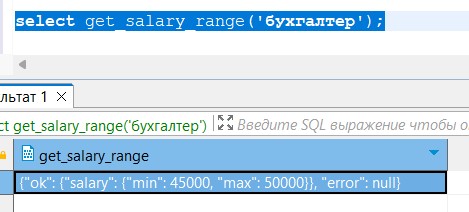
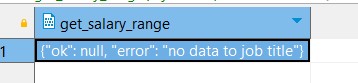
result['error'] = None else:

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;



Задани 3

CREATE Or replace FUNCTION handler() RETURNS TRIGGER AS $$ from datetime import datetime

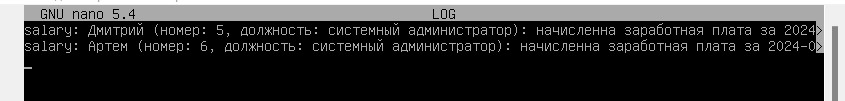
query = plpy.prepare("select name from positions where pos\_id =

$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();



Задание 4

create or replace function vacancy\_message() returns trigger as $$ if TD['event'] == 'DELETE':

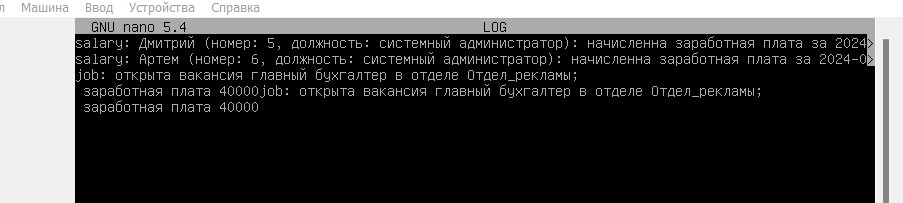
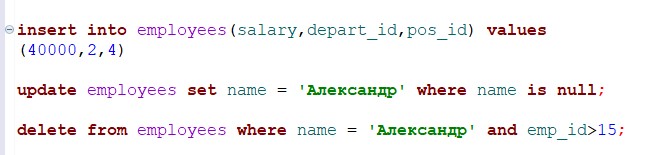
query = plpy.prepare('select name from positions where pos\_id = $1',['int']) pos = plpy.execute(query,[TD['old']['pos\_id']])[0] query = plpy.prepare('select name from department where depart\_id = $1',['int']) dep = plpy.execute(query,[TD['old']['depart\_id']])[0] fn =f"job: открыта вакансия {pos['name']} 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



в представленном выше коде перенос на новую строку добавил

|  |
| --- |
|  |

In [ ]: