Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ №9**

**«База данных с использованием Postgresql»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Яровиков Илья Евгеньевич

Преподаватель:

Сергеева Е.Г.

Киров

2023

1. Цель домашней контрольной работы: Закрепление навыков реализации приложений с графическим интерфейсом пользователя и получение навыков работы с posgtgresql
2. Формулировка задания   
   1) Реализовать приложение «Справочник Call of Duty».

2) Добавить базу данных Postgreqsl.

3) Предусмотреть возможность поиска в базе данных.

1. Код программы

import tkinter as tk

import psycopg2

from tkinter import ttk

from PIL import ImageTk, Image

host = "127.0.0.1"

user = "postgres"

password = "5gcv783nz"

db\_name = "Call\_of\_Duty"

port = "5432"

root = tk.Tk()

root.title("Лучший справочник для фанатов Call of Duty")

root.configure(background='#8C8C8C')

root.resizable(False, False)

connection = psycopg2.connect(

host=host,

user=user,

password=password,

database=db\_name,

port = port

)

connection.autocommit = True

cursor = connection.cursor()

def add\_window():

def add\_entry():

name = name1\_entry.get()

squad = squad1\_entry.get()

nickname = nickname1\_entry.get()

first\_appearance = first\_appearance1\_combobox.get()

if name and squad and nickname and first\_appearance:

insert\_query = "INSERT INTO duty (Имя, Отряд, Позывной, Первое\_появление) VALUES (%s, %s, %s, %s)"

data = (name, squad, nickname, first\_appearance)

cursor.execute(insert\_query, data)

connection.commit()

name1\_entry.delete(0, tk.END)

squad1\_entry.delete(0, tk.END)

nickname1\_entry.delete(0, tk.END)

first\_appearance1\_combobox.delete(0, tk.END)

output\_table()

window\_add.destroy()

window\_add = tk.Tk()

window\_add.title("Добавление записи")

window\_add.geometry("400x300")

window\_add.configure(background='#636363')

window\_add.resizable(False, False)

redac\_label = ttk.Label(window\_add, text="Имя:", background='#636363')

redac\_label.pack()

name1\_entry = ttk.Entry(window\_add)

name1\_entry.pack()

squad1\_label = ttk.Label(window\_add, text="Отряд:", background='#636363')

squad1\_label.pack()

squad1\_entry = ttk.Entry(window\_add)

squad1\_entry.pack()

nickname1\_label = ttk.Label(window\_add, text="Позывной:", background='#636363')

nickname1\_label.pack()

nickname1\_entry = ttk.Entry(window\_add)

nickname1\_entry.pack()

first\_appearance1\_label = ttk.Label(window\_add, text="Первое появление:", background='#636363')

first\_appearance1\_label.pack()

first\_appearance1\_combobox = ttk.Combobox(window\_add, values=["Call of Duty Modern Warfare 2", "Call of Duty Modern Warfare 3", "Call of Duty Modern Warfare 4", "Call of Duty Black Ops ", "Call of Duty Black Ops 2"], width=27)

first\_appearance1\_combobox.pack()

add1\_button = tk.Button(window\_add, text="Добавить", command=add\_entry, bg='#17301A')

add1\_button.pack(pady=5)

def delete\_entry():

selected\_item = treeview.focus()

if selected\_item:

name = treeview.item(selected\_item, "text")

delete\_query = "DELETE FROM duty WHERE Имя = %s"

cursor.execute(delete\_query, (name,))

connection.commit()

treeview.delete(selected\_item)

output\_table()

def output\_table():

treeview.delete(\*treeview.get\_children())

select\_query = "SELECT \* FROM duty"

cursor.execute(select\_query)

records = cursor.fetchall()

for row in records:

name = row[0]

squad = row[1]

nickname = row[2]

first\_appearance = row[3]

treeview.insert("", tk.END, text=name, values=(squad, nickname, first\_appearance))

def search\_entry():

def perform\_search():

search\_value = search\_entry.get()

select\_query = "SELECT \* FROM duty WHERE Имя ILIKE %s OR Отряд ILIKE %s OR Позывной ILIKE %s OR Первое\_появление ILIKE %s"

cursor.execute(select\_query, ('%' + search\_value + '%', '%' + search\_value + '%', '%' + search\_value + '%', '%' + search\_value + '%'))

records = cursor.fetchall()

treeview.delete(\*treeview.get\_children())

for row in records:

name = row[0]

squad = row[1]

nickname = row[2]

first\_appearance = row[3]

treeview.insert("", tk.END, text=name, values=(squad, nickname, first\_appearance))

search\_window = tk.Toplevel(root)

search\_window.title("Поиск")

search\_window.geometry("400x100")

search\_window.configure(background='#636363')

search\_window.resizable(False, False)

search\_label = ttk.Label(search\_window, text="Введите запрос:", background='#636363')

search\_label.pack()

search\_entry = ttk.Entry(search\_window)

search\_entry.pack()

search\_button = tk.Button(search\_window, text="Поиск", command=perform\_search, bg='#17301A', width=15, height=2)

search\_button.pack(pady=5)

def refresh\_table():

output\_table()

def edit\_entry():

selected\_item = treeview.focus()

if selected\_item:

name = treeview.item(selected\_item, "text")

squad = treeview.item(selected\_item, "values")[0]

nickname = treeview.item(selected\_item, "values")[1]

first\_appearance = treeview.item(selected\_item, "values")[2]

# Создайте окно для редактирования записи с предварительно заполненными полями

window\_edit = tk.Tk()

window\_edit.title("Редактирование записи")

window\_edit.geometry("400x300")

window\_edit.configure(background='#636363')

window\_edit.resizable(False, False)

name\_label = ttk.Label(window\_edit, text="Имя:", background='#636363')

name\_label.pack()

name\_entry = ttk.Entry(window\_edit)

name\_entry.pack()

name\_entry.insert(0, name)

squad\_label = ttk.Label(window\_edit, text="Отряд:", background='#636363')

squad\_label.pack()

squad\_entry = ttk.Entry(window\_edit)

squad\_entry.pack()

squad\_entry.insert(0, squad)

nickname\_label = ttk.Label(window\_edit, text="Позывной:", background='#636363')

nickname\_label.pack()

nickname\_entry = ttk.Entry(window\_edit)

nickname\_entry.pack()

nickname\_entry.insert(0, nickname)

first\_appearance\_label = ttk.Label(window\_edit, text="Первое появление:", background='#636363')

first\_appearance\_label.pack()

first\_appearance1\_combobox = ttk.Combobox(window\_edit, values=["Call of Duty Modern Warfare 2", "Call of Duty Modern Warfare 3", "Call of Duty Modern Warfare 4", "Call of Duty Black Ops ", "Call of Duty Black Ops 2"], width=27)

first\_appearance1\_combobox.pack()

first\_appearance1\_combobox.insert(0, first\_appearance)

def save\_changes():

updated\_name = name\_entry.get()

updated\_squad = squad\_entry.get()

updated\_nickname = nickname\_entry.get()

updated\_first\_appearance = first\_appearance1\_combobox.get()

update\_query = "UPDATE duty SET Имя = %s, Отряд = %s, Позывной = %s, Первое\_появление = %s WHERE Имя = %s"

data = (updated\_name, updated\_squad, updated\_nickname, updated\_first\_appearance, name)

cursor.execute(update\_query, data)

connection.commit()

window\_edit.destroy()

output\_table()

save\_button = tk.Button(window\_edit, text="Сохранить", command=save\_changes, bg='light blue')

save\_button.pack(pady=5)

window\_edit.mainloop()

treeview = ttk.Treeview(root)

treeview["columns"] = ("squad", "nickname", "first\_appearance")

treeview.heading("#0", text="Имя")

treeview.heading("squad", text="Отряд")

treeview.heading("nickname", text="Позывной")

treeview.heading("first\_appearance", text="Первое появление")

treeview.pack(pady=10)

# Create a frame to hold the buttons

button\_frame = tk.Frame(root, bg='#8C8C8C')

button\_frame.pack(pady=5)

add\_button = tk.Button(button\_frame, text="Добавить", command=add\_window, bg='#17301A', width=15, height=3)

add\_button.grid(row=0, column=1, padx=5)

delete\_button = tk.Button(button\_frame, text="Удалить", command=delete\_entry, bg='#17301A', width=15, height=3)

delete\_button.grid(row=0, column=3, padx=5)

output\_button = tk.Button(button\_frame, text="Выход", command=root.destroy, bg='#981916', width=15, height=3)

output\_button.grid(row=0, column=4, padx=5)

edit\_button = tk.Button(button\_frame, text="Изменить", command=edit\_entry, bg='#17301A', width=15, height=3)

edit\_button.grid(row=0, column=2, padx=5)

refresh\_button = tk.Button(button\_frame, text="Обновить",command=refresh\_table, bg='#17301A', width=15, height=3)

refresh\_button.grid(row=0, column=6, padx=5)

searh\_button = tk.Button(button\_frame, text="Поиск",command=search\_entry, bg='#17301A', width=15, height=3)

searh\_button.grid(row=0, column=7, padx=5)

def on\_closing():

cursor.close()

connection.close()

root.destroy()

root.protocol("WM\_DELETE\_WINDOW", on\_closing)

output\_table()

root.mainloop()

4.Результат выполнения программы

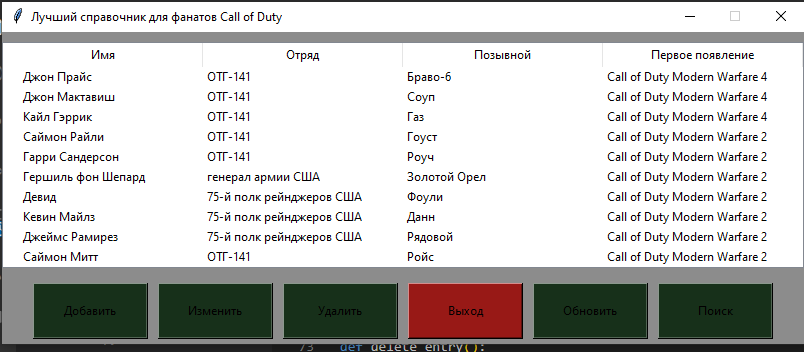


Рисунок1 – Справочник Call of Duty

Данный справочник включает в себя 6 функциональных кнопок: «Добавить», «Изменить», «Удалить», «Выход», «Обновить», «Поиск».

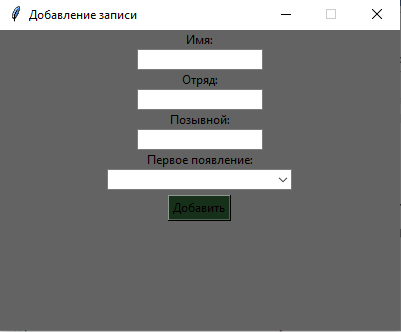
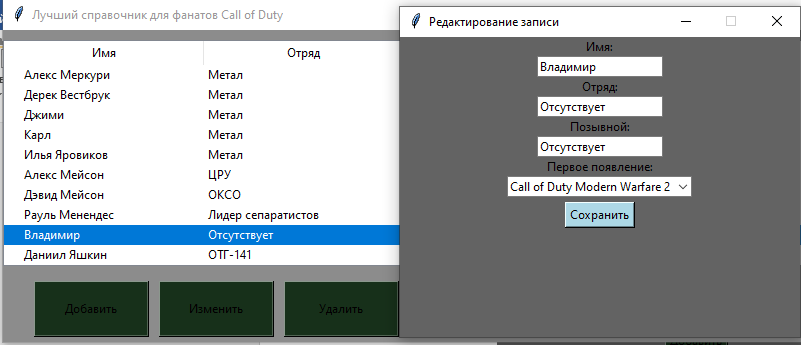


Рисунок 2 – Функция добавления записи

Рисунок 3 – Функция изменения записи

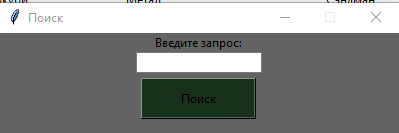


Рисунок 4 – Функция поиска записи

5. Вывод

Исходя из полученных опытным путем знаний и умений мы можем сформулировать емкий, а главное правильный вывод.

Во время выполнений домашняя контрольная номер №9 мы на практике применили знания, полученные на лекционных занятиях по МДК

В ходе работы мы узнали, что такое Postgreql и как с ним работать Научились создавать базу данных и подключать ее к Python. Закрепили навыки работы с графической библиотекой Tkinter для создания таблицы.

Таким образом Домашняя контрольная 8 была выполнена