TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO 2

LẬP TRÌNH PYTHON NÂNG CAO

Chủ đề:

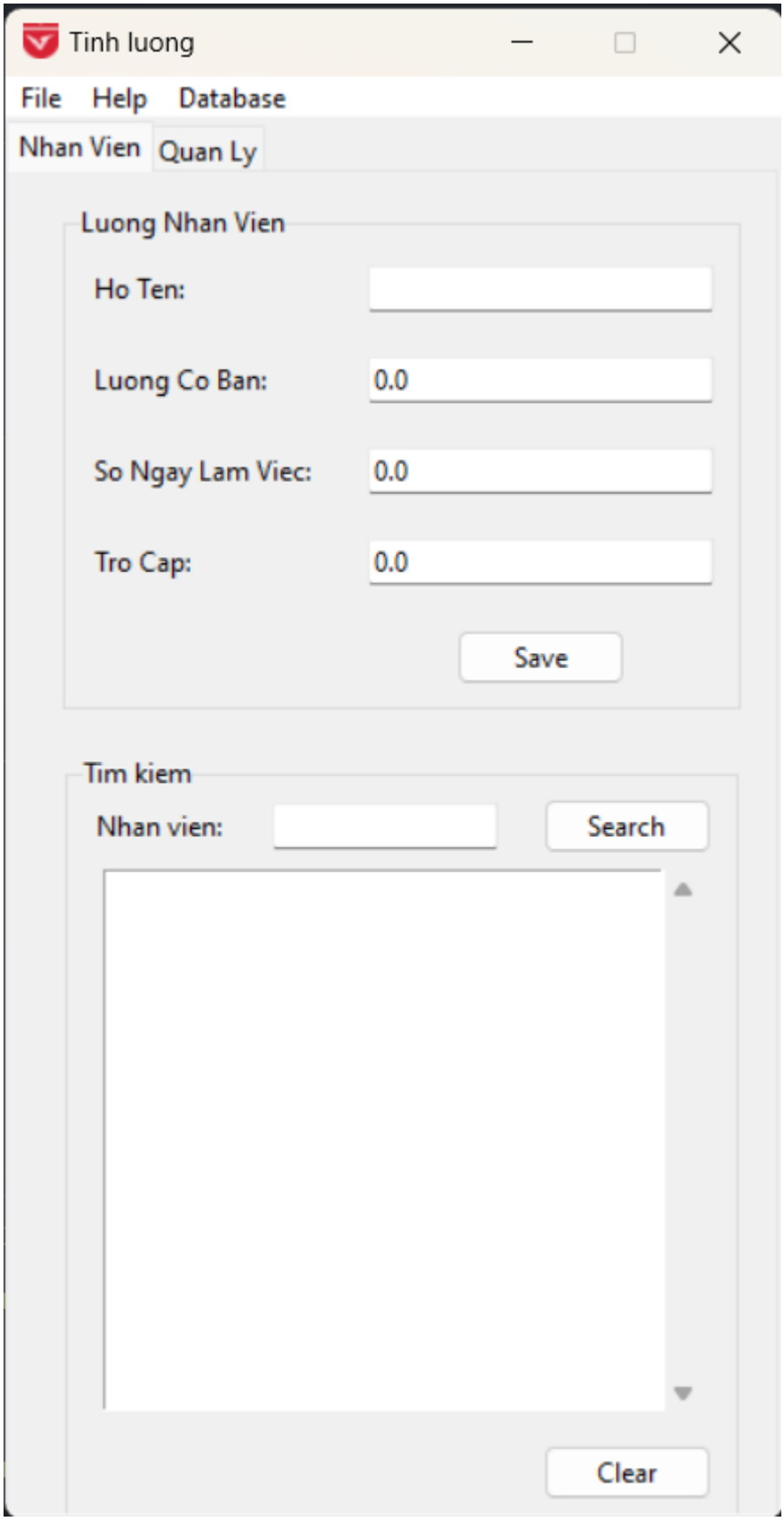
HỌ VÀ TÊN: LƯƠNG NHẬT QUANG

MSSV: 2274802010710

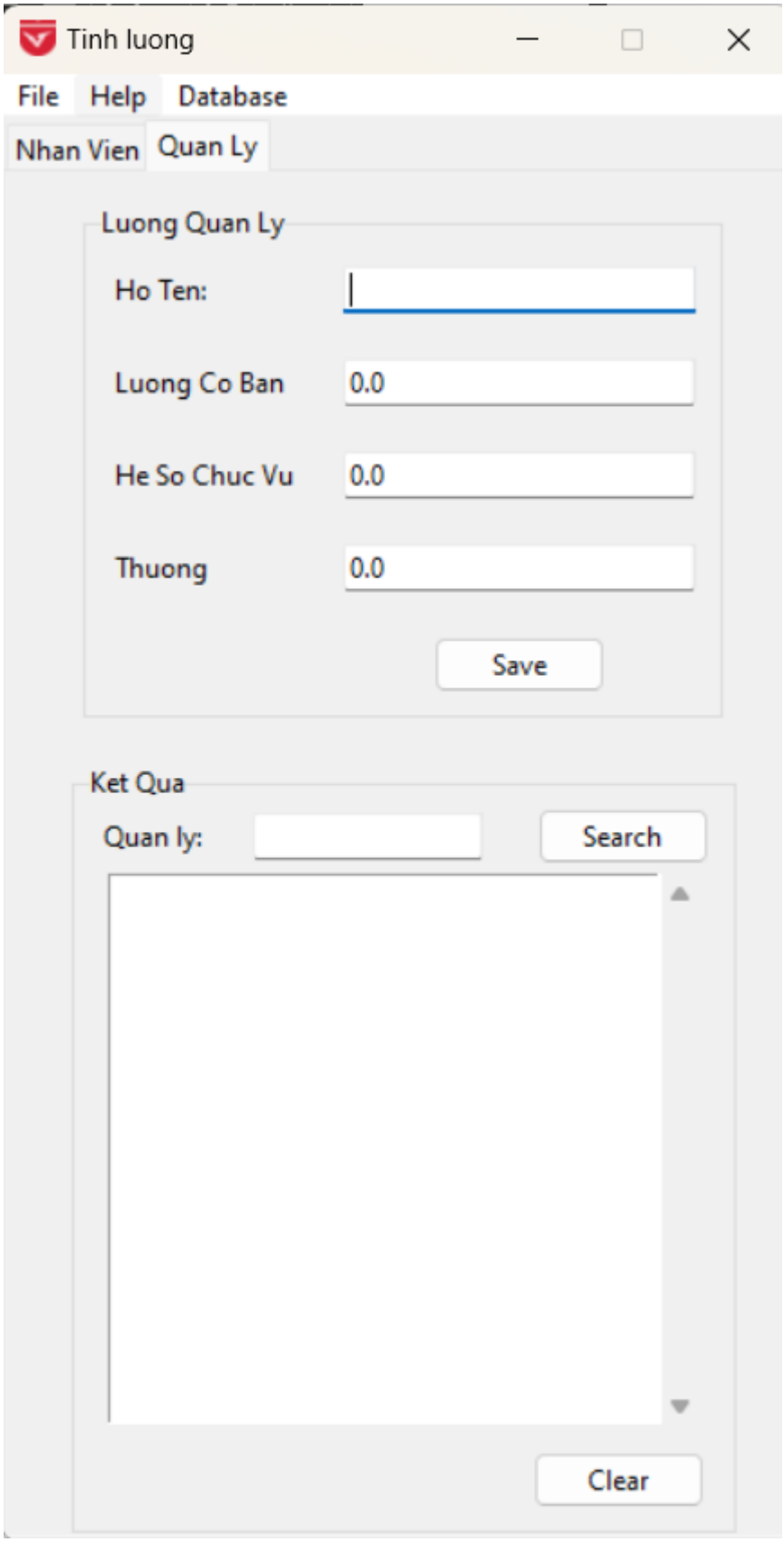
TP. Hồ Chí Minh – 7/2024

1. Giao diện

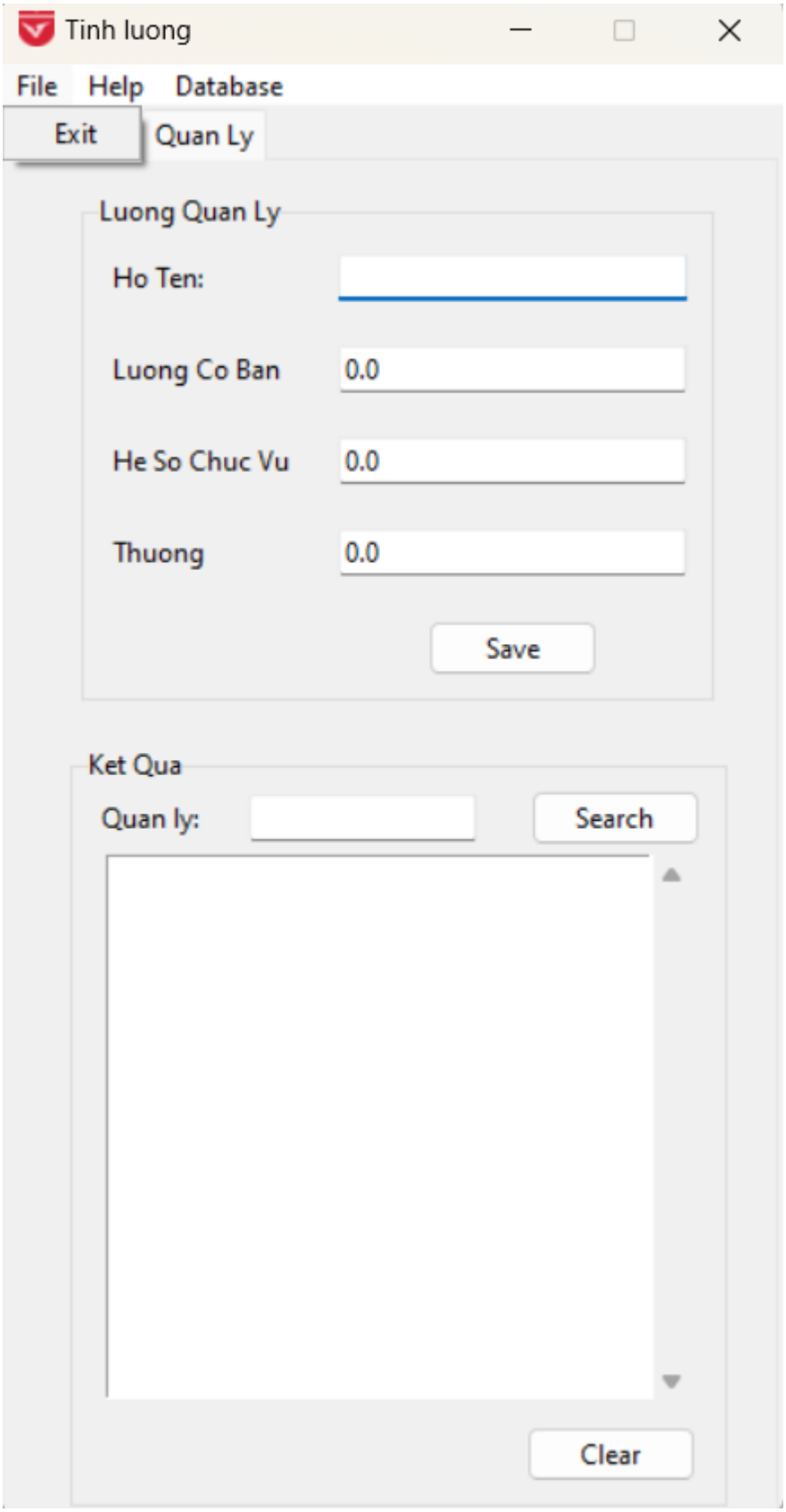
a. Giao diện mặc định (Nhân viên)

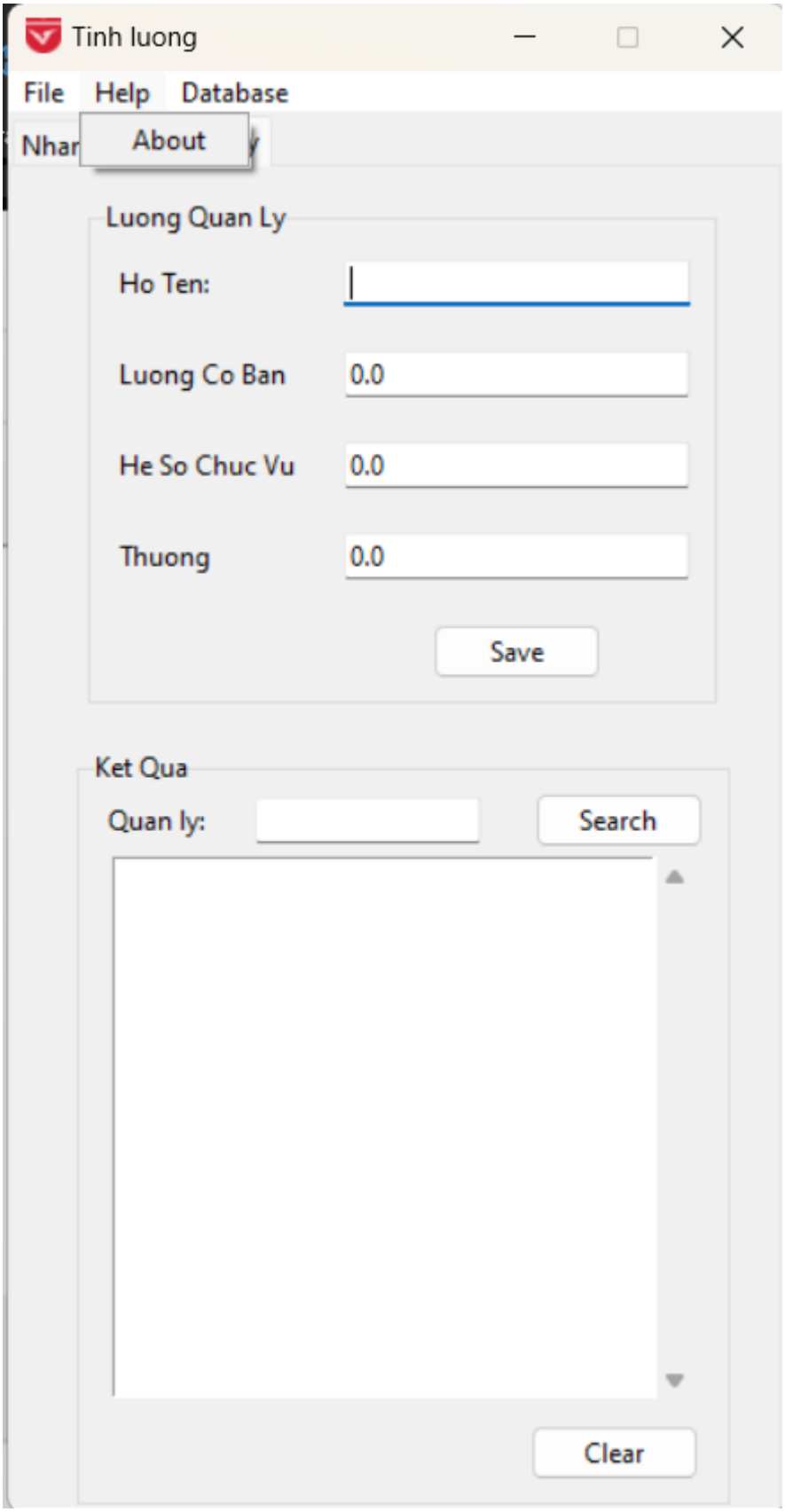


b. Giao diện mặc định (Quản lý)

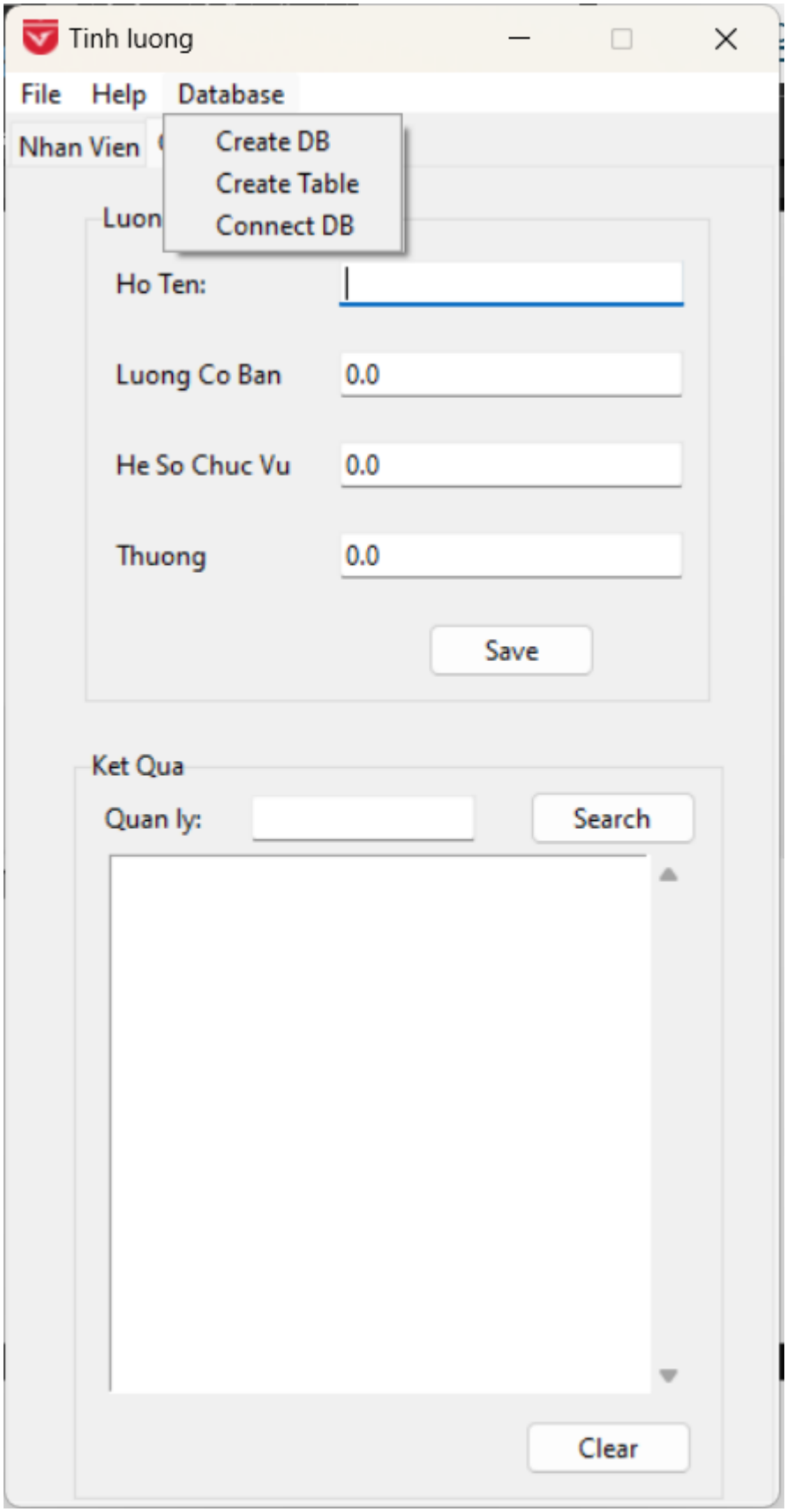


c. Tab File, Tab Help


Tinh luong
File
Help
Database
Exit
Quan Ly
Luong Quan Ly
Ho Ten:
Luong Co Ban
0.0
He So Chuc Vu
0.0
Thuong
0.0
Save
Ket Qua
Quan ly:
Search
Clear

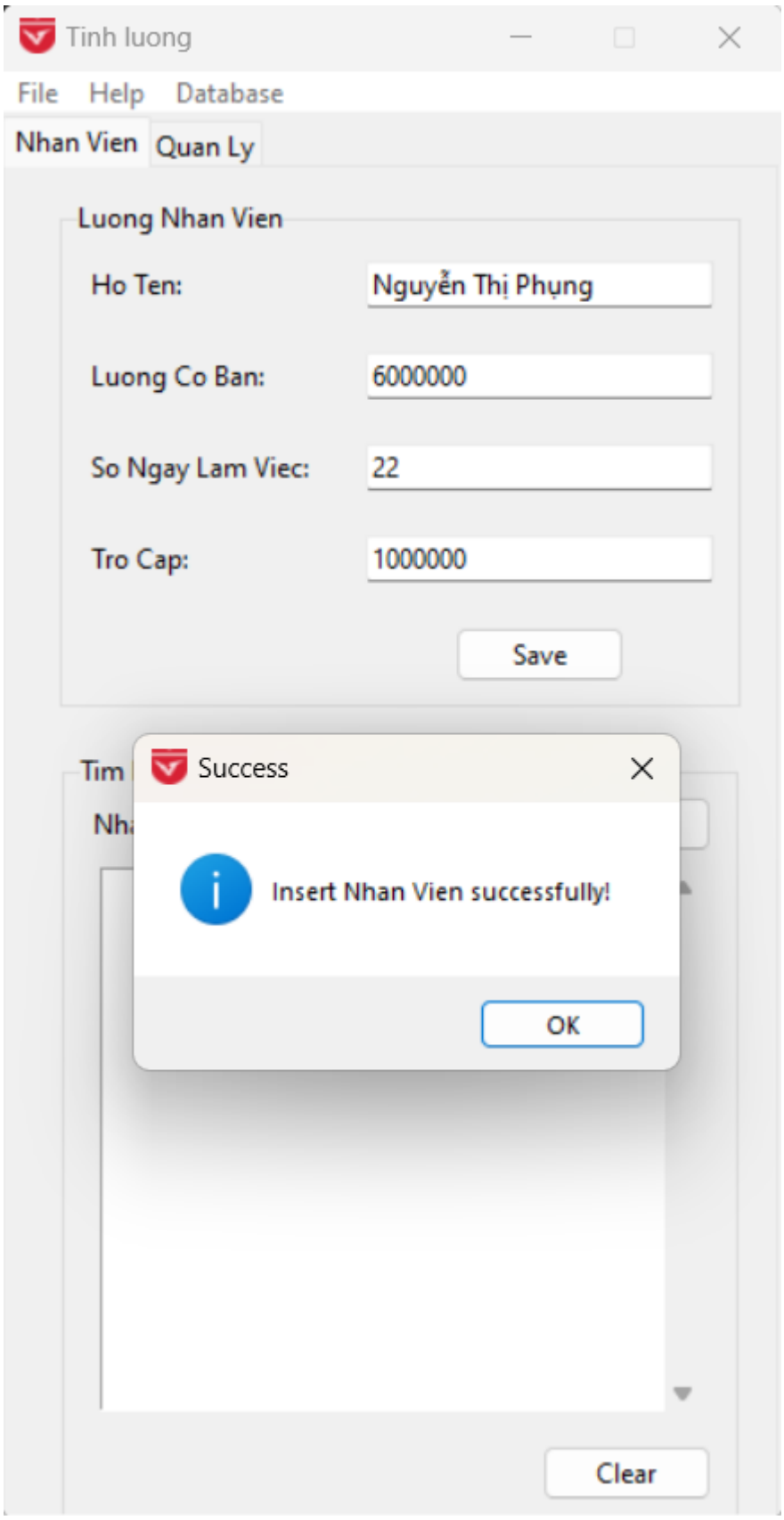


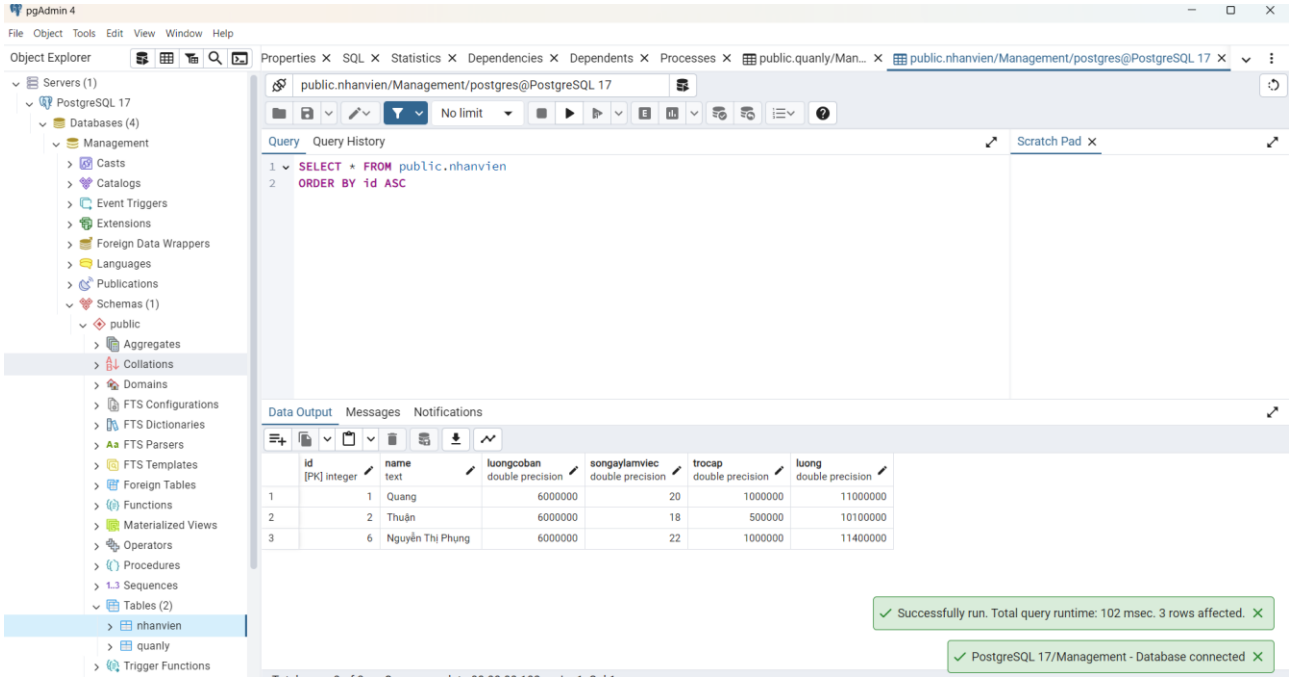
d. Tab Database



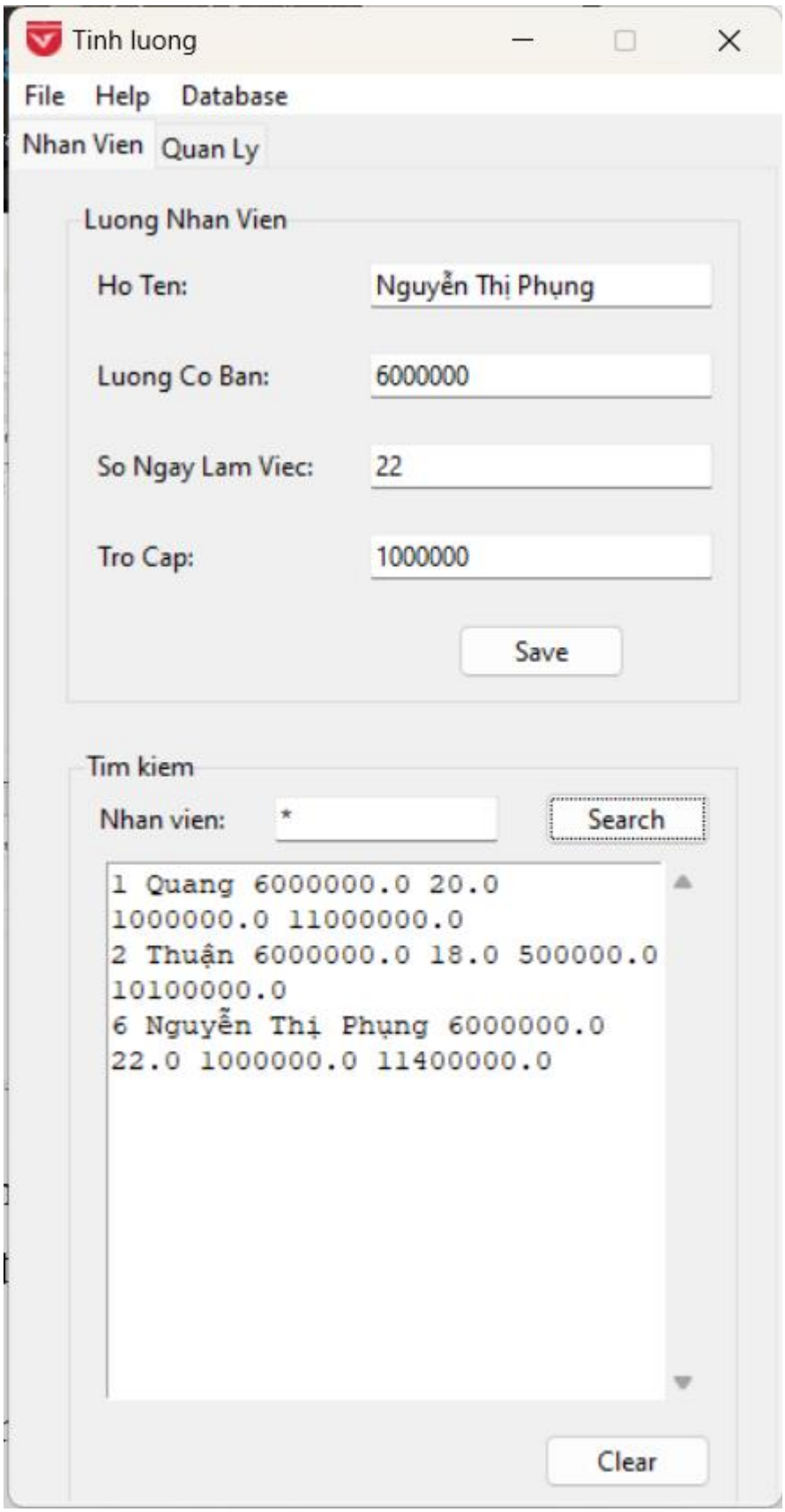
2. Chức năng

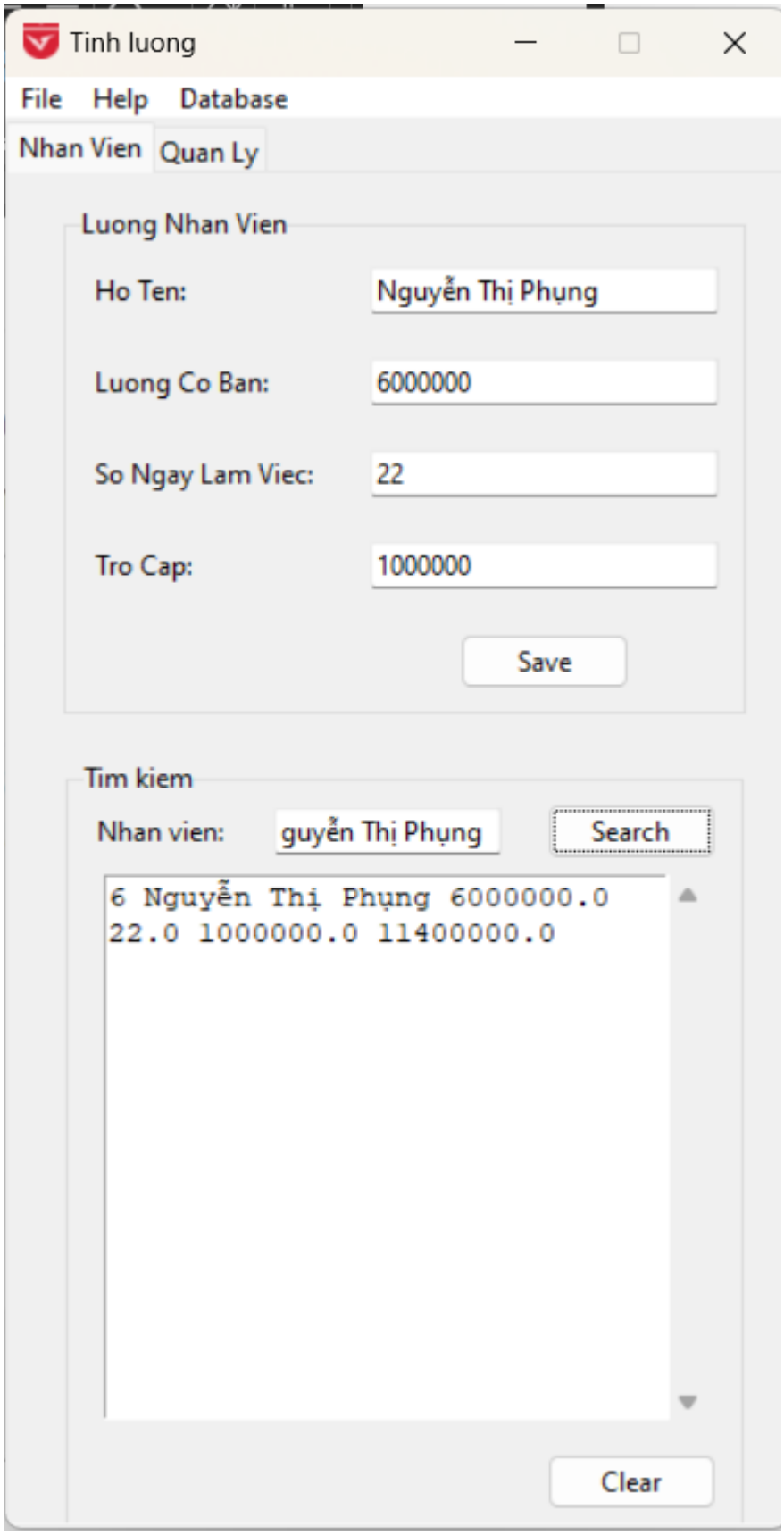
a. Tính lương nhân viên và lưu vào csdl khi nhấn Save


Tinh luong
File
Help
Database
Nhan Vien
Quan Ly
Luong Nhan Vien
Ho Ten:
Nguyễn Thị Phụng
Luong Co Ban:
6000000
So Ngay Lam Viec:
22
Tro Cap:
1000000
Save
Tim
Nh
Success
Insert Nhan Vien successfully!
OK
Clear

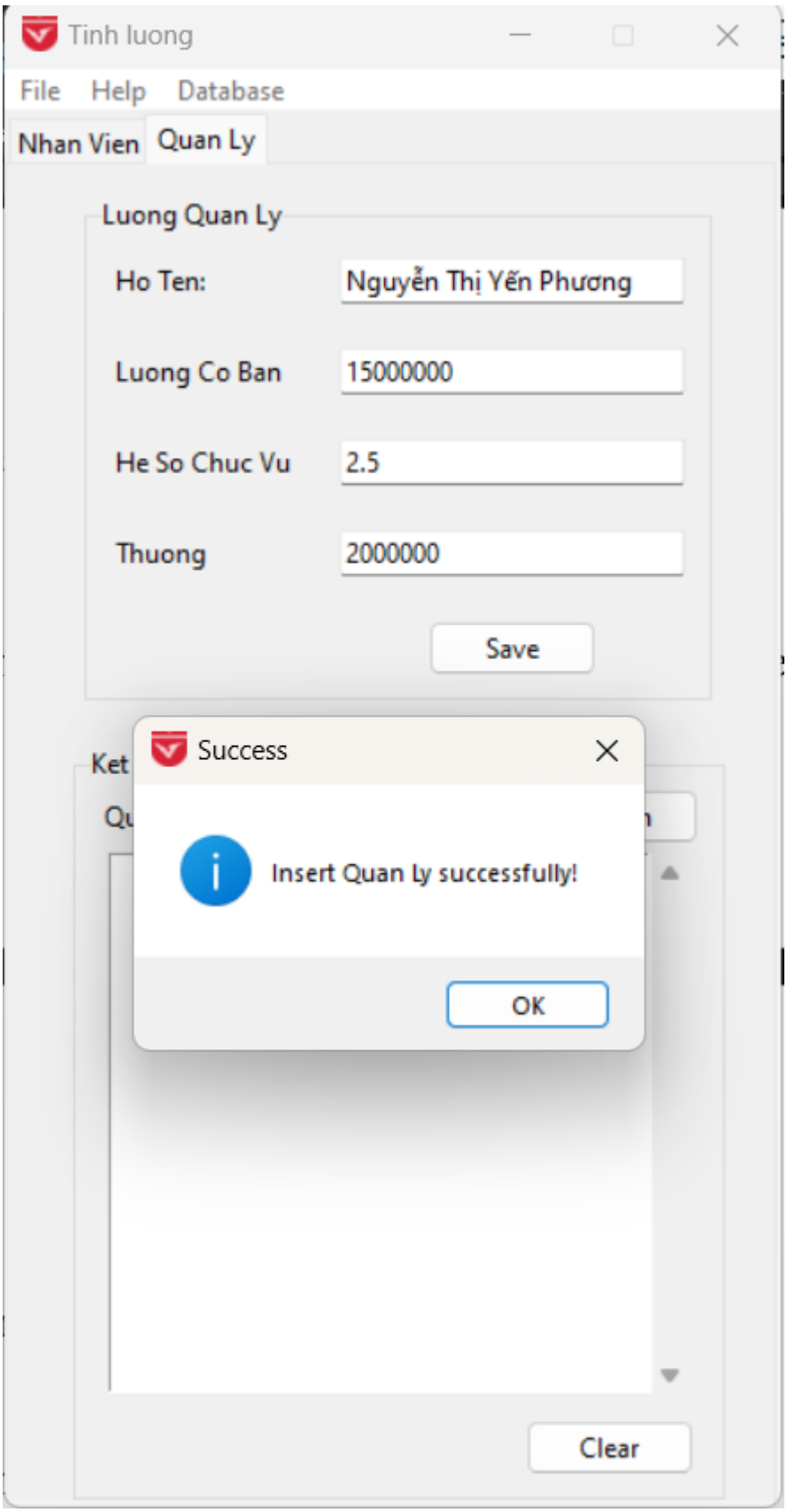


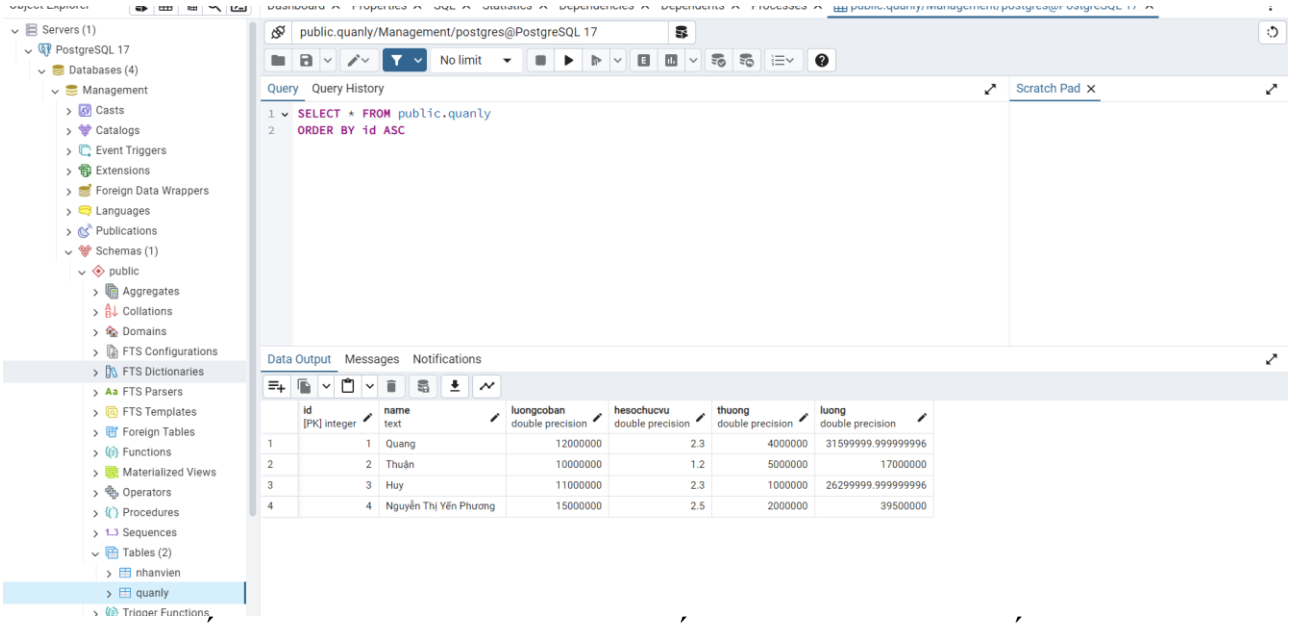
b. Tìm kiếm thông tin của nhân viên khi nhấn Search (“*” là tìm tất cả, “name” để tìm từng cá nhân)


Tinh luong
File
Help
Database
Nhan Vien
Quan Ly
Luong Nhan Vien
Ho Ten:
Nguyễn Thị Phụng
Luong Co Ban:
6000000
So Ngay Lam Viec:
22
Tro Cap:
1000000
Save
Tim kiem
Nhan vien:
*
Search
1 Quang 6000000.0 20.0
1000000.0 11000000.0
2 Thuận 6000000.0 18.0 500000.0
10100000.0
6 Nguyễn Thị Phụng 6000000.0
22.0 1000000.0 11400000.0
Clear

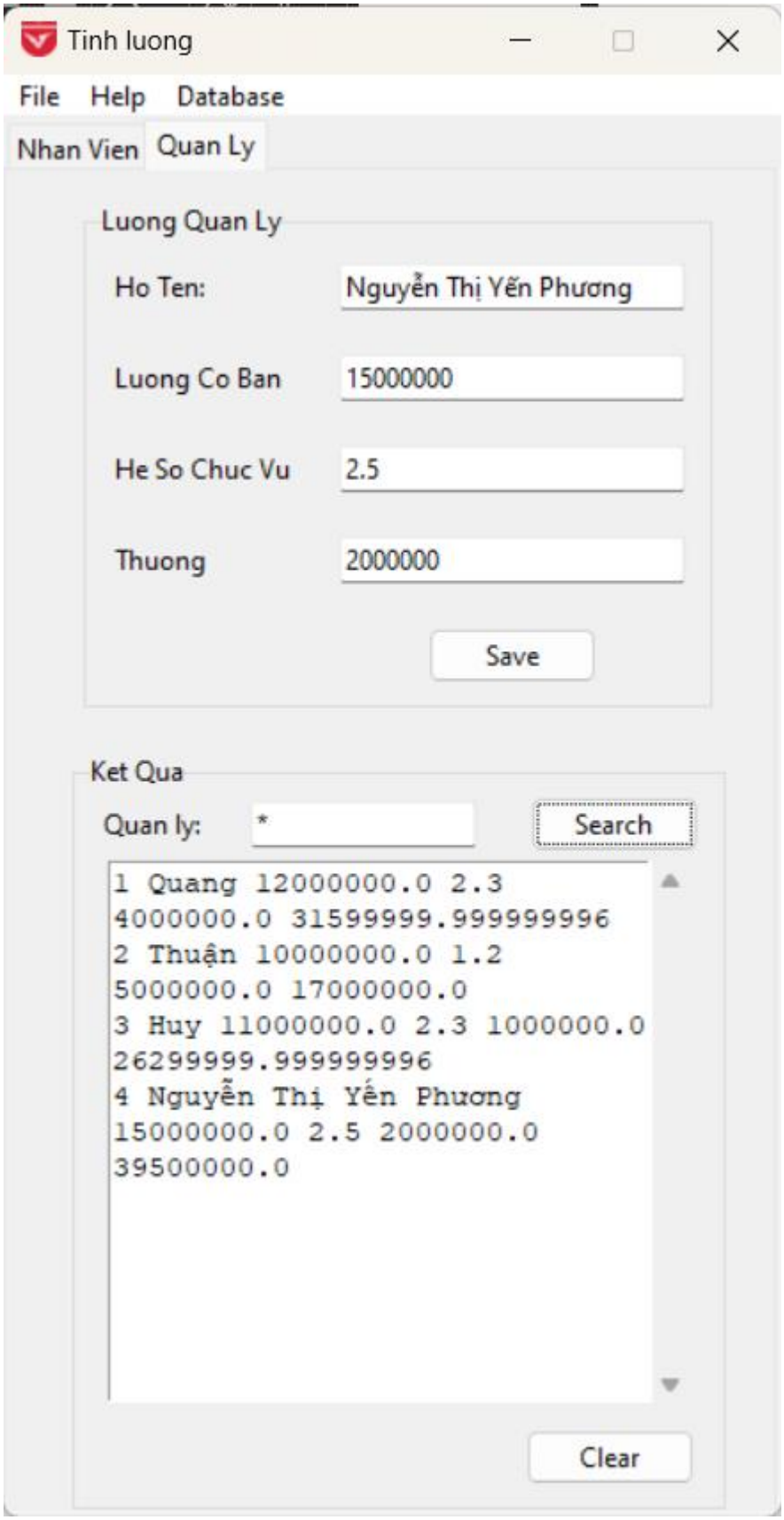


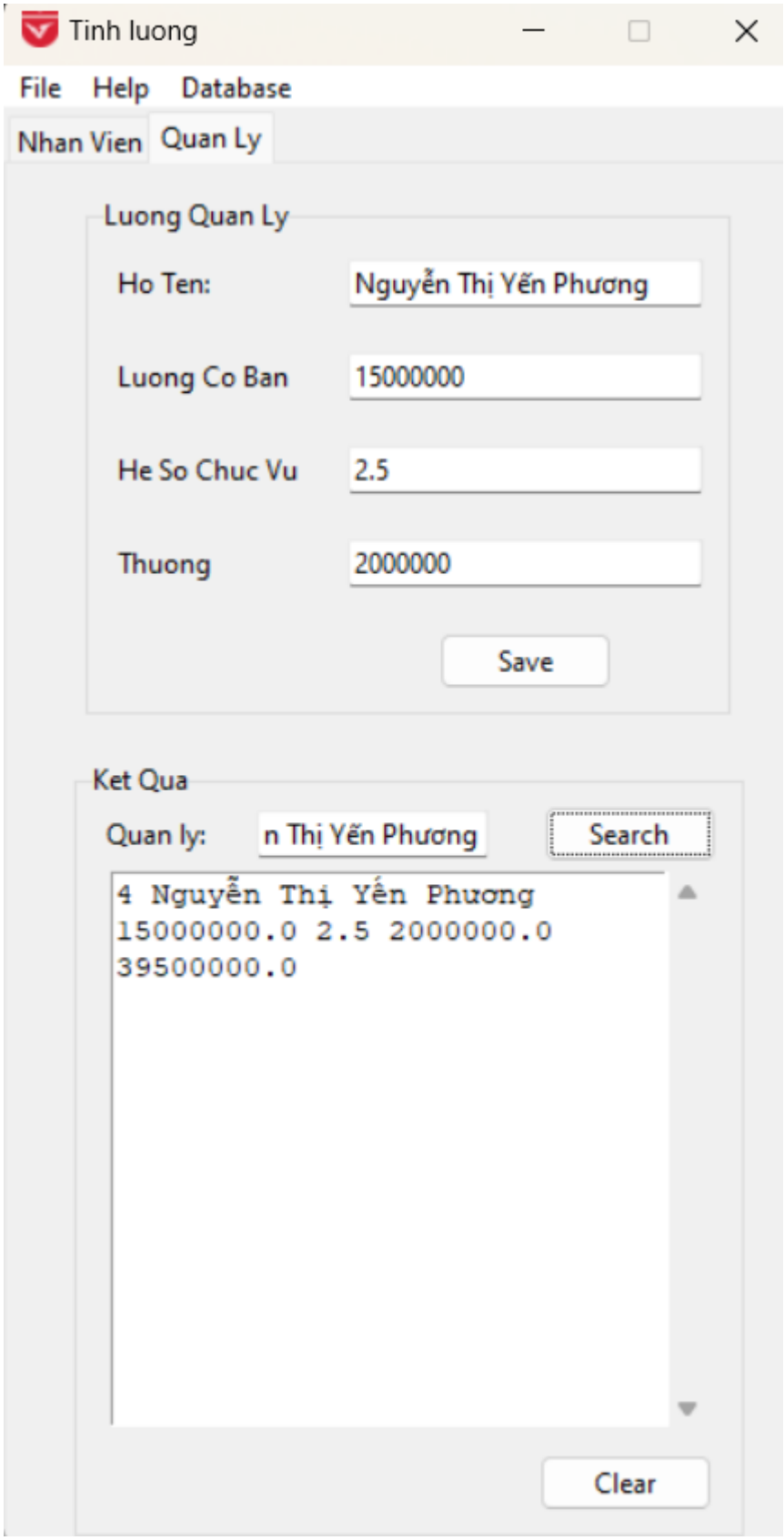
c. Tính lương quản lý và lưu vào csdl khi nhấn Save


Tinh luong
File
Help
Database
Nhan Vien
Quan Ly
Luong Quan Ly
Ho Ten:
Nguyễn Thị Yến Phương
Luong Co Ban
15000000
He So Chuc Vu
2.5
Thuong
2000000
Save
Success
Insert Quan Ly successfully!
OK
Clear

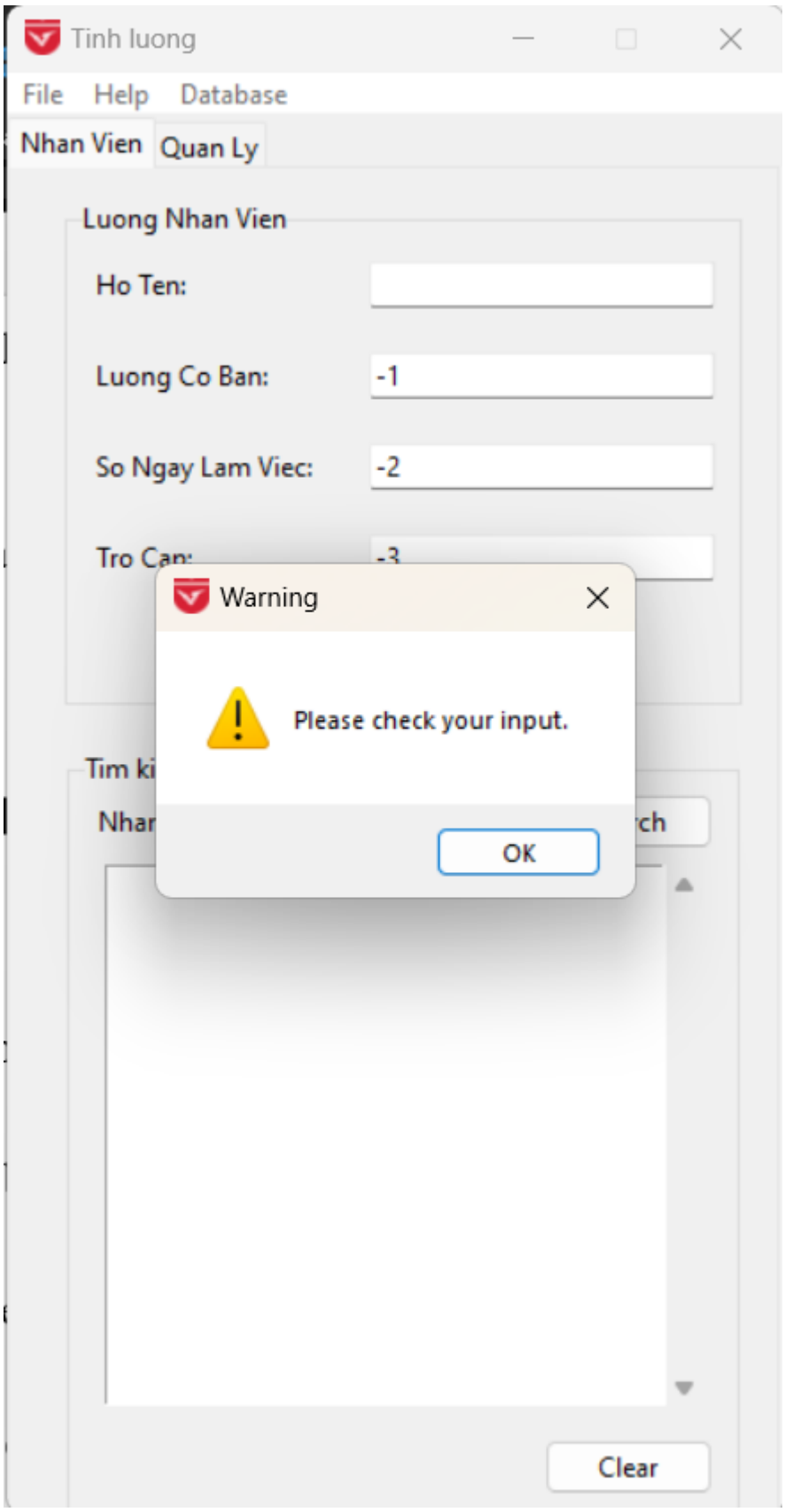


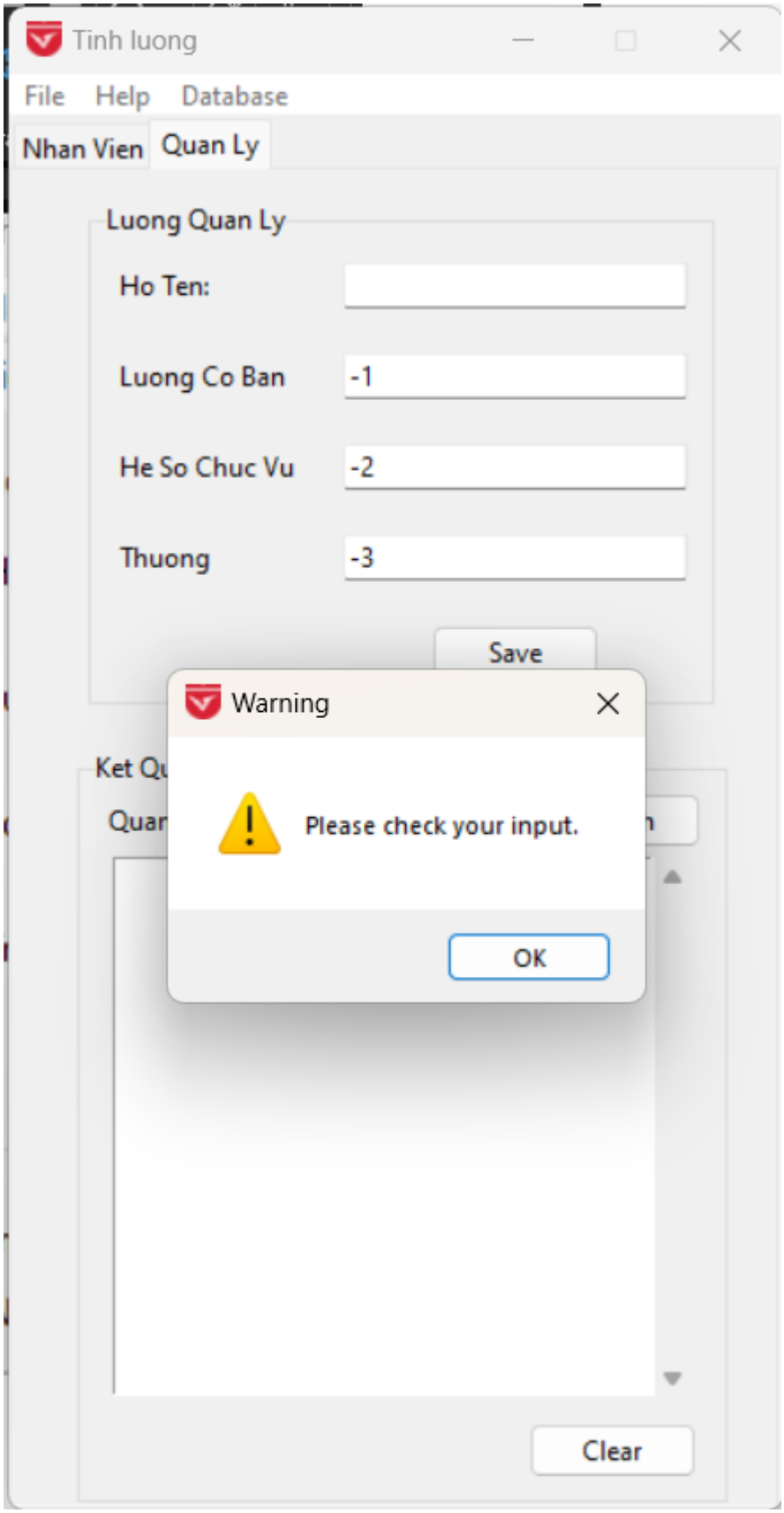
d. Tìm kiếm thông tin của quản lý khi nhấn Search (“*” là tìm tất cả, “name” để tìm từng cá nhân)


Tinh luong
File
Help
Database
Nhan Vien
Quan Ly
Luong Quan Ly
Ho Ten:
Nguyễn Thị Yến Phương
Luong Co Ban
15000000
He So Chuc Vu
2.5
Thuong
2000000
Save
Ket Qua
Quan ly:
*
Search
1 Quang 12000000.0 2.3
4000000.0 31599999.999999996
2 Thuận 10000000.0 1.2
5000000.0 17000000.0
3 Huy 11000000.0 2.3 1000000.0
26299999.999999996
4 Nguyễn Thị Yến Phương
15000000.0 2.5 2000000.0
39500000.0
Clear

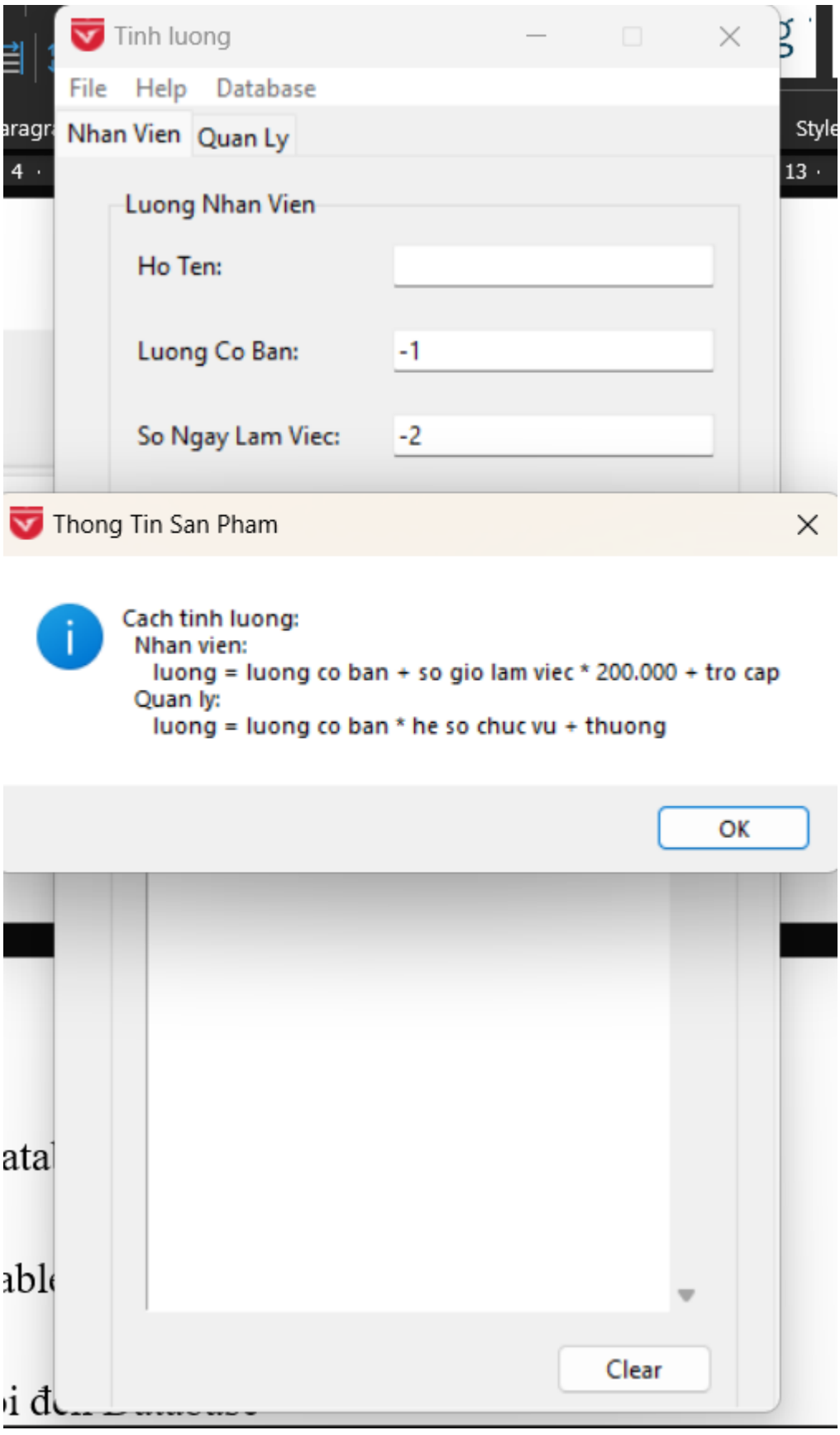


e. Báo lỗi khi dữ liệu đầu vào sai

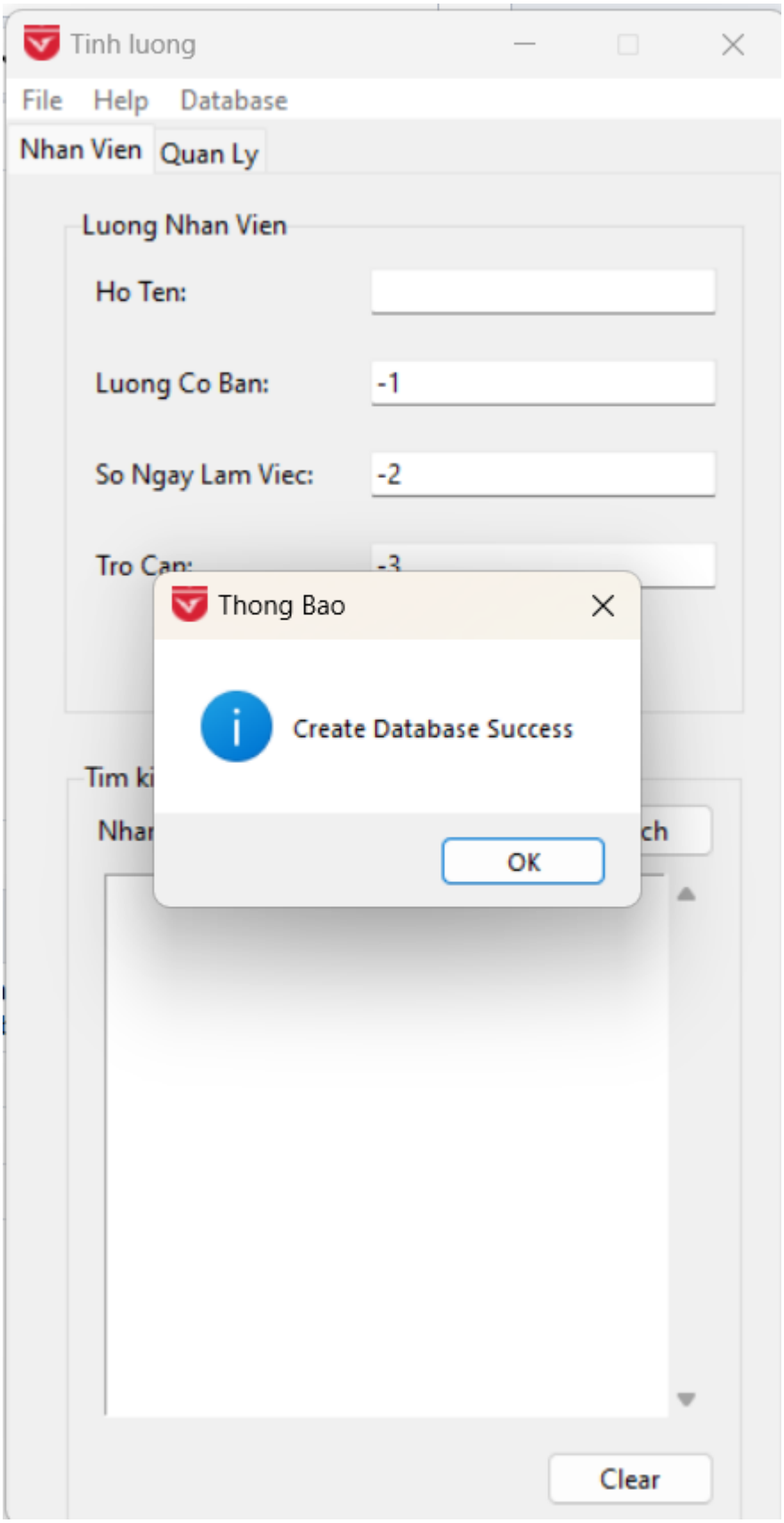

Tinh luong
File
Help
Database
Nhan Vien
Quan Ly
Luong Nhan Vien
Ho Ten:
Luong Co Ban:
-1
So Ngay Lam Viec:
-2
Tro Cap:
-3
Warning
Please check your input.
OK
Tim ki
Nhar
ch
Clear

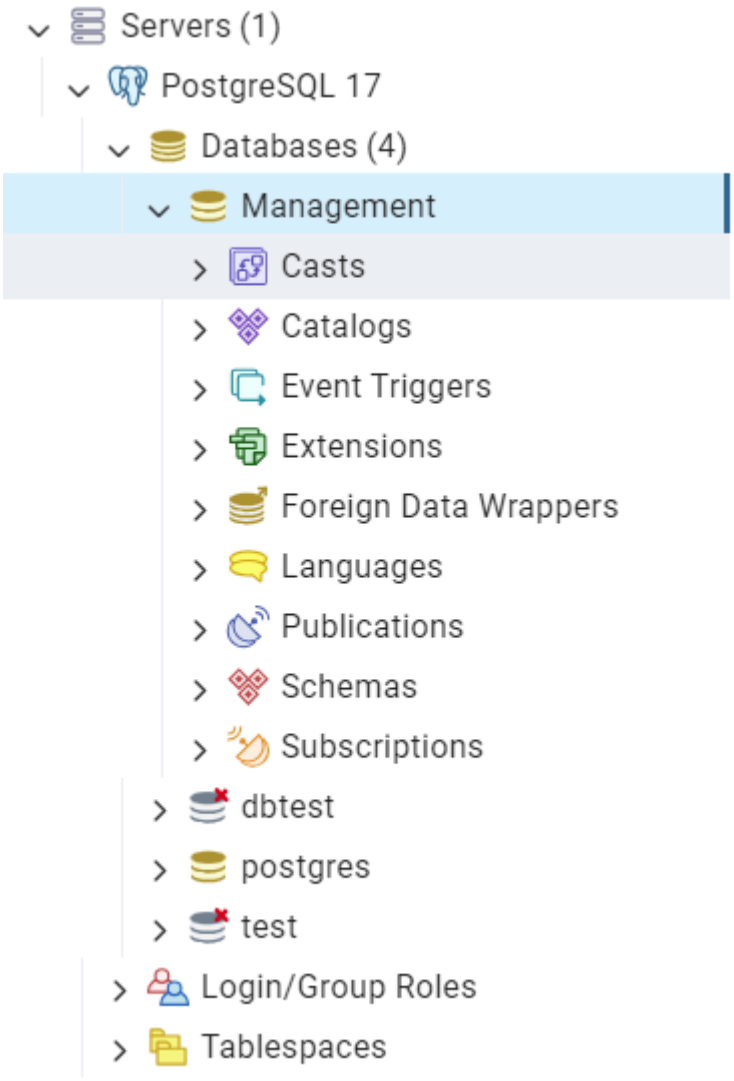


f. Thông báo khi nhấn vào Help -> About sẽ cung cấp cho người dùng cách lương của từng bộ phận được tính như nào

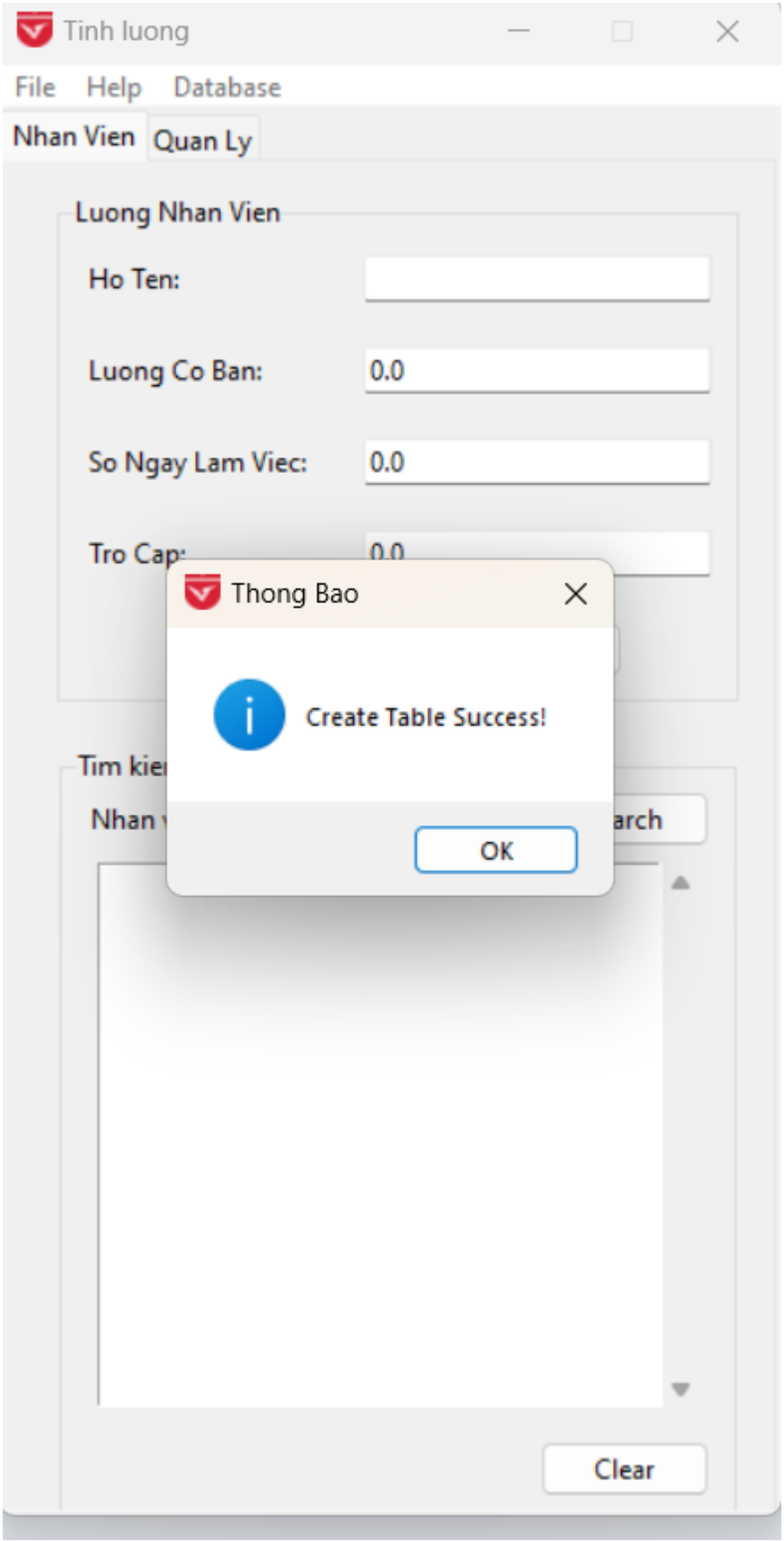


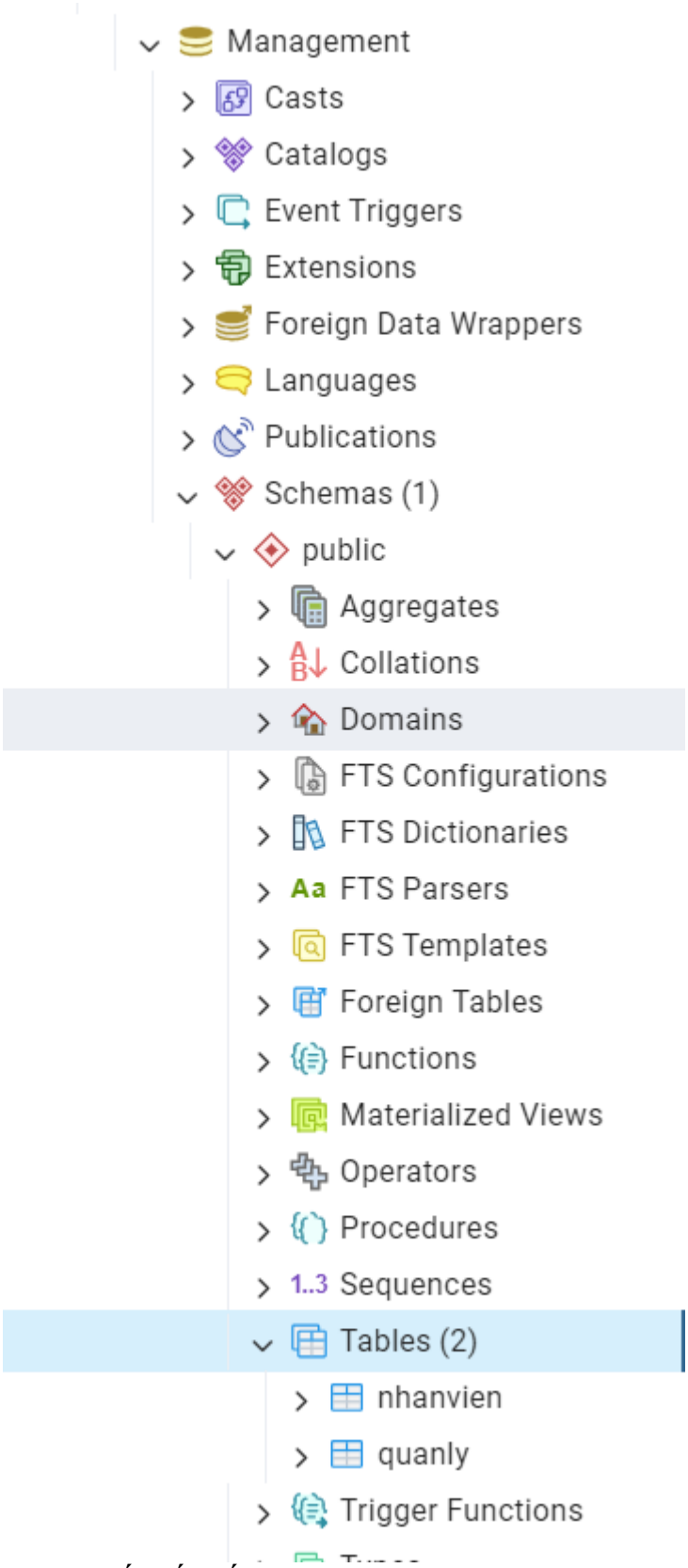
g. Tạo Database


Tinh luong
File
Help
Database
Nhan Vien
Quan Ly
Luong Nhan Vien
Ho Ten:
Luong Co Ban:
-1
So Ngay Lam Viec:
-2
Tro Cap:
-3
Thong Bao
Create Database Success
OK
Tim ki
Nhar
ch
Clear

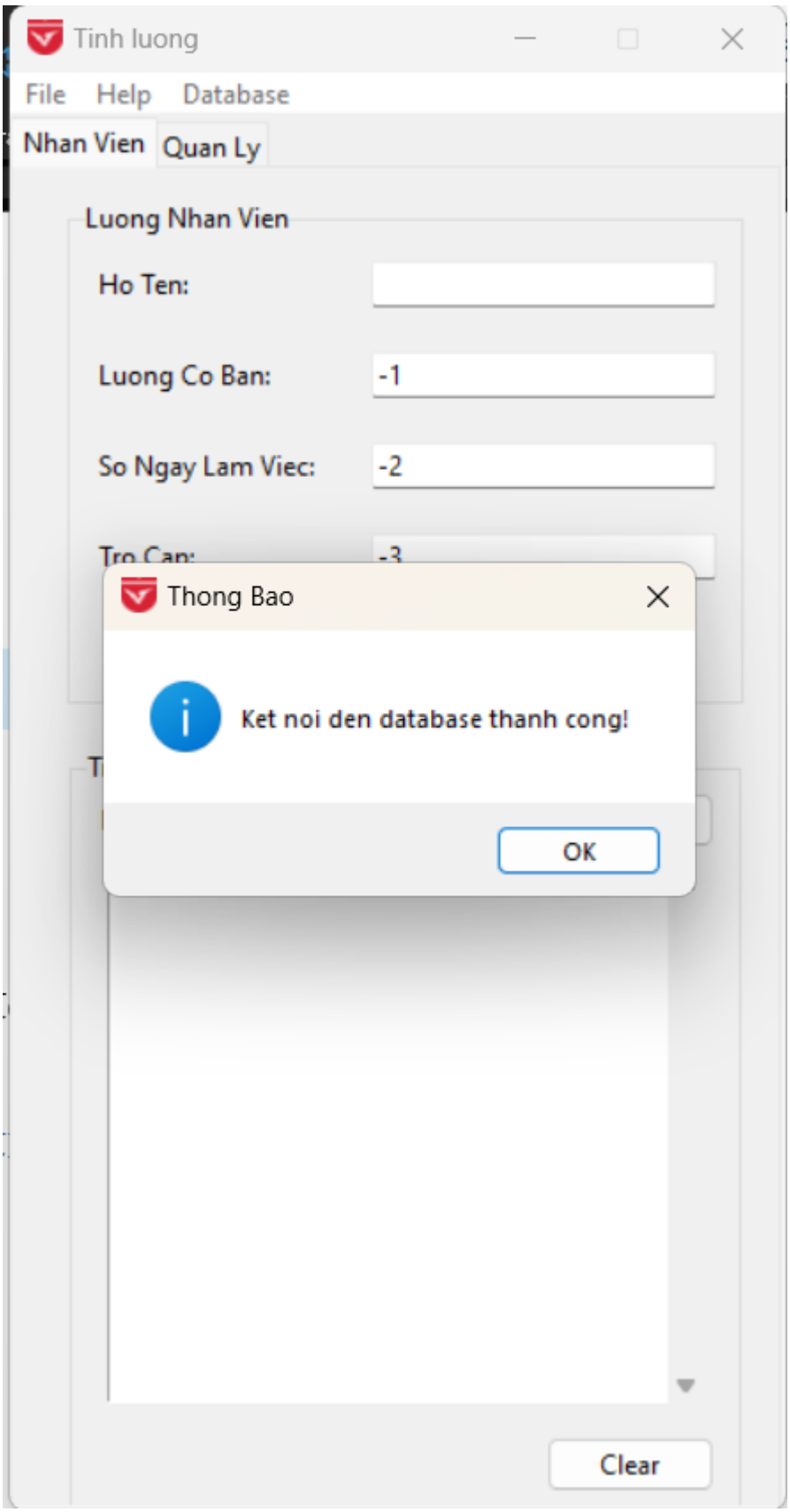

Servers (1)
PostgreSQL 17
Databases (4)
Management
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
dbtest
postgres
test
Login/Group Roles
Tablespaces

h. Tạo Table


Tinh luong
File
Help
Database
Nhan Vien
Quan Ly
Luong Nhan Vien
Ho Ten:
Luong Co Ban:
0.0
So Ngay Lam Viec:
0.0
Tro Cap:
0.0
Thong Bao
Create Table Success!
OK
Tim kie
Nhan
arch
Clear



i. Kết nối đến Database



3. Mã chương trình

a. def__init__

```python
class TinhLuong:
    def __init__(self, root):
        window_height = 650

        # Lấy kích thước màn hình
        screen_width = self.root.winfo_screenwidth()
        screen_height = self.root.winfo_screenheight()

        # Tính toán tọa độ x và y để đặt cửa sổ ở trung tâm
        center_x = int(screen_width/2 - window_width/2)
        center_y = int(screen_height/2 - window_height/2)

        # Đặt kích thước và vị trí cửa sổ
        self.root.geometry(f'{window_width}x{window_height}+{center_x}+{center_y}')

        #Đặt logo co form
        self.root.iconbitmap('2274802010710_LuongNhatQuang_BaoCao2/IconDHVL.ico')

        #menu bar
        menu_bar = Menu(win)
        win.config(menu=menu_bar)
        file_menu = Menu(menu_bar, tearoff=0)
        file_menu.add_command(label="Exit", command=exit)
        menu_bar.add_cascade(label="File", menu=file_menu)

        help_menu = Menu(menu_bar, tearoff=0)
        help_menu.add_command(label="About", command=self.ThongTinSanPham)
        menu_bar.add_cascade(label="Help", menu=help_menu)

        database_menu = Menu(menu_bar, tearoff=0)
        database_menu.add_command(label="Create DB", command=self.create_DB)
        database_menu.add_command(label="Create Table", command=self.create_Table)
        database_menu.add_command(label="Connect DB", command=self.conn_DB)
        menu_bar.add_cascade(label="Database", menu=database_menu)

        tabControl = ttk.Notebook(self.root)
        tabControl.pack(expand=1, fill="both")

        self.NhanVien(tabControl)
        self.QuanLy(tabControl)
```

```python
if __name__ == "__main__":
    win = tk.Tk()
    TinhLuong(win)
    win.mainloop()
```

b. def NhanVien

```python
def NhanVien(self, tabControl):
    #tab Nhan Vien
    tabNhanVien = ttk.Frame(tabControl)
    tabControl.add(tabNhanVien, text="Nhan Vien")

    #frame Nhan Vien
    frame_NhanVien = ttk.LabelFrame(tabNhanVien, text="Luong Nhan Vien")
    frame_NhanVien.grid(column=0, row=0, padx=25, pady=15)

    #label Tinh
    lbl_name = ttk.Label(frame_NhanVien, text="Ho Ten: ")
    lbl_name.grid(column=0, row=0, padx=10, pady=10, sticky="W")

    lbl_lcb = ttk.Label(frame_NhanVien, text="Luong Co Ban: ")
    lbl_lcb.grid(column=0, row=1, padx=10, pady=10, sticky="W")

    lbl_snlv = ttk.Label(frame_NhanVien, text="So Ngay Lam Viec: ")
    lbl_snlv.grid(column=0, row=2, padx=10, pady=10, sticky="W")

    lbl_trocap = ttk.Label(frame_NhanVien, text="Tro Cap: ")
    lbl_trocap.grid(column=0, row=3, padx=10, pady=10, sticky="W")

    #textbox
    self.name_nhanvien = tk.StringVar()
    txt_name = ttk.Entry(frame_NhanVien, width=25, textvariable=self.name_nhanvien)
    txt_name.grid(column=1, row=0, padx=10, pady=10)
    txt_name.focus()

    self.lcb_nhanvien = tk.DoubleVar()
    txt_lcb = ttk.Entry(frame_NhanVien, width=25, textvariable=self.lcb_nhanvien)
    txt_lcb.grid(column=1, row=1, padx=10, pady=10)

    self.snlv = tk.DoubleVar()
    txt_sglv = ttk.Entry(frame_NhanVien, width=25, textvariable=self.snlv)
    txt_sglv.grid(column=1, row=2, padx=10, pady=10)

    self.trocap = tk.DoubleVar()
    txt_trocap = ttk.Entry(frame_NhanVien, width=25, textvariable=self.trocap)
    txt_trocap.grid(column=1, row=3, padx=10, pady=10)

    #btn Tinh
    ttk.Button(frame_NhanVien, text="Save", command=self.insert_DB_NhanVien).grid(column=1, row=4, pady=10)

    #frame Tinh Luong Nhan Vien
    frame_LuongNhanVien = ttk.LabelFrame(tabNhanVien, text="Tim kiem")
    frame_LuongNhanVien.grid(column=0, row=1, padx=20, pady=5)

    #label luong
    lbl_luong = ttk.Label(frame_LuongNhanVien, text="Nhan vien:")
    lbl_luong.grid(column=0, row=0, padx=10, pady=2, sticky="W")

    #input Search
    self.search_nhanvien = tk.StringVar()
    txt_search = ttk.Entry(frame_LuongNhanVien, width=16, textvariable=self.search_nhanvien)
    txt_search.grid(column=1, row=0, padx=10, pady=2)

    #btn Search
    ttk.Button(frame_LuongNhanVien, text="Search", command=self.Search_LuongNhanVien).grid(column=2, row=0, sticky="E", padx=10, pady=2)

    #scrolledtext
    scr_w = 31
    scr_h = 15
    self.scroll_NhanVien = scrolledtext.ScrolledText(frame_LuongNhanVien, width = scr_w, height = scr_h, wrap = tk.WORD)
    self.scroll_NhanVien.grid(column=0, row=1, columnspan=3, padx=10, pady=5)

    #btn Clear
    ttk.Button(frame_LuongNhanVien, text="Clear", command=self.Clear_LuongNhanVien).grid(column=2, row=2, sticky="E", padx=10, pady=10)
```

c. def QuanLy

```python
def QuanLy(self, tabControl):
    #tab Quan Ly
    tabQuanLy = ttk.Frame(tabControl)
    tabControl.add(tabQuanLy, text="Quan Ly")

    #frame Nhan Vien
    frame_QuanLy = ttk.LabelFrame(tabQuanLy, text="Luong Quan Ly")
    frame_QuanLy.grid(column=0, row=0, padx=30, pady=15)

    #label Tinh
    lbl_name = ttk.Label(frame_QuanLy, text="Ho Ten: ")
    lbl_name.grid(column=0, row=0, padx=10, pady=10, sticky="W")

    lbl_lcb = ttk.Label(frame_QuanLy, text="Luong Co Ban")
    lbl_lcb.grid(column=0, row=1, padx=10, pady=10, sticky="W")

    lbl_snlv = ttk.Label(frame_QuanLy, text="He So Chuc Vu")
    lbl_snlv.grid(column=0, row=2, padx=10, pady=10, sticky="W")

    lbl_trocap = ttk.Label(frame_QuanLy, text="Thuong")
    lbl_trocap.grid(column=0, row=3, padx=10, pady=10, sticky="W")

    #textbox
    self.name_quanly = tk.StringVar()
    txt_name = ttk.Entry(frame_QuanLy, width=25, textvariable=self.name_quanly)
    txt_name.grid(column=1, row=0, padx=10, pady=10)
    txt_name.focus()

    self.lcb_quanly = tk.DoubleVar()
    txt_lcb = ttk.Entry(frame_QuanLy, width=25, textvariable=self.lcb_quanly)
    txt_lcb.grid(column=1, row=1, padx=10, pady=10)

    self.hscv = tk.DoubleVar()
    txt_hscv = ttk.Entry(frame_QuanLy, width=25, textvariable=self.hscv)
    txt_hscv.grid(column=1, row=2, padx=10, pady=10)

    self.thuong = tk.DoubleVar()
    txt_thuong = ttk.Entry(frame_QuanLy, width=25, textvariable=self.thuong)
    txt_thuong.grid(column=1, row=3, padx=10, pady=10)

    #btn Tinh
    ttk.Button(frame_QuanLy, text="Save", command=self.insert_DB_QuanLy).grid(column=1, row=4, pady=10)

    #frame Tinh Luong Nhan Vien
    frame_LuongQuanLy = ttk.LabelFrame(tabQuanLy, text="Ket Qua")
    frame_LuongQuanLy.grid(column=0, row=1, padx=30, pady=5)

    #label luong
    lbl_luong = ttk.Label(frame_LuongQuanLy, text="Quan ly:")
    lbl_luong.grid(column=0, row=0, padx=10, pady=2, sticky="W")

    #input search
    self.search_quanly = tk.StringVar()
    txt_search = ttk.Entry(frame_LuongQuanLy, width=16, textvariable=self.search_quanly)
    txt_search.grid(column=1, row=0, padx=10, pady=2)

    #btn Search
    ttk.Button(frame_LuongQuanLy, text="Search", command=self.Search_LuongQuanLy).grid(column=2, row=0, sticky="E", padx=10, pady=2)

    #scrolledtext
    scr_w = 30
    scr_h = 15
    self.scroll_QuanLy = scrolledtext.ScrolledText(frame_LuongQuanLy, width = scr_w, height = scr_h, wrap = tk.WORD)
    self.scroll_QuanLy.grid(column=0, row=1, columnspan=3, padx=10, pady=2)

    #btn Clear
    ttk.Button(frame_LuongQuanLy, text="Clear", command=self.Clear_LuongQuanLy).grid(column=2, row=2, sticky="E", padx=12, pady=10)
```

d. def LuongNhanVien

```
def LuongNhanVien(self):
    try:
        lcb = self.lcb_nhanvien.get()
        snlv = self.snlv.get()
        trocap = self.trocap.get()
        return lcb + snlv * 200000 + trocap
    except Exception as ex:
        messagebox.showerror("Input Error", "Lỗi dữ liệu đầu vào!\nVui lòng xem và nhập lại")
```

e. def LuongQuanLy

```
def LuongQuanLy(self):
    try:
        lcb = self.lcb_quanly.get()
        hscv = self.hscv.get()
        thuong = self.thuong.get()
        return lcb * hscv + thuong
    except Exception as ex:
        messagebox.showerror("Input Error", "Lỗi dữ liệu đầu vào!\nVui lòng xem và nhập lại")
```

f. def ThongTinSanPham

```
def ThongTinSanPham(self):
    messagebox.showinfo("Thong Tin San Pham", "Cach tinh luong:\n  Nhan vien: \n     luong = luong co ban + so gio lam viec * 200.000 + tro cap\n  Quan ly: \n     luong
```

g. def Clear_LuongNhanVien

```
def Clear_LuongNhanVien(self):
    self.scroll_NhanVien.delete(1.0, tk.END)
```

h. def Clear_LuongQuanLy

```
def Clear_LuongQuanLy(self):
    self.scroll_QuanLy.delete(1.0, tk.END)
```

i. def create_DB

```
def create_DB(self):
    try:
        conn = psycopg2.connect(
            database = "postgres",
            user = "postgres",
            password = "130104",
            host = "localhost",
            port = "5432"
        )
        conn.set_isolation_level(psycopg2.extensions.ISOLATION_LEVEL_AUTOCOMMIT)
        cur = conn.cursor()
        database_name = "Management"
        cur.execute(sql.SQL("CREATE DATABASE {}").format(sql.Identifier(database_name)))
        messagebox.showinfo("Thong Bao", "Create Database Success")
    except Exception as e:
        messagebox.showerror("Error", f"Error create the new database: {e}")
    finally:
        if cur is not None:
            cur.close()
        if conn is not None:
            conn.close()
```

j. def conn_DB

```
def conn_DB(self):
    try:
        self.conn = psycopg2.connect(database = self.database, user = self.user, password = self.password, host = self.host, port = self.port)
        messagebox.showinfo("Thong Bao", "Ket noi den database thanh cong!")
        self.cur = self.conn.cursor()
    except Exception as e:
        messagebox.showerror("Error", f"Error connecting to the database: {e}")
```

k. def create_Table

```
def create_Table(self):
    try:
        self.conn_DB()
        self.cur.execute('''CREATE TABLE NhanVien
                            (
                            ID SERIAL PRIMARY KEY NOT NULL,
                            NAME TEXT NOT NULL,
                            LUONGCOBAN DOUBLE PRECISION NOT NULL,
                            SONGAYLAMVIEC DOUBLE PRECISION NOT NULL,
                            TROCAP DOUBLE PRECISION NOT NULL,
                            LUONG DOUBLE PRECISION NOT NULL
                            );''')
        self.cur.execute('''CREATE TABLE QuanLy
                            (
                            ID SERIAL PRIMARY KEY NOT NULL,
                            NAME TEXT NOT NULL,
                            LUONGCOBAN DOUBLE PRECISION NOT NULL,
                            HESOCHUCVU DOUBLE PRECISION NOT NULL,
                            THUONG DOUBLE PRECISION NOT NULL,
                            LUONG DOUBLE PRECISION NOT NULL
                            );''')
        self.conn.commit()
        messagebox.showinfo("Thong Bao", "Create Table Success!")
    except Exception as e:
        messagebox.showerror("Error", f"Error create the new table: {e}")
```

l. def insert_DB_NhanVien

```
def insert_DB_NhanVien(self):
    try:
        if (self.lcb_nhanvien.get() >= 0 and self.snlv.get() >= 0 and self.trocap.get() >= 0) and (self.name_nhanvien.get().strip()):
            insert_query = sql.SQL("INSERT INTO {} (name, luongcoban, songaylamviec, trocap, luong) VALUES (%s, %s, %s, %s, %s)").format(sql.Identifier("nhanvien"))
            data_to_insert = (self.name_nhanvien.get(), self.lcb_nhanvien.get(), self.snlv.get(), self.trocap.get(), self.LuongNhanVien())
            self.cur.execute(insert_query, data_to_insert)
            self.conn.commit()
            messagebox.showinfo("Success", "Insert Nhan Vien successfully!")
        else:
            messagebox.showwarning("Warning", "Please check your input.")
    except Exception as e:
        messagebox.showerror("Error", f"Error inserting data: {e}")
```

m. def insert_DB_QuanLy

```
def insert_DB_NhanVien(self):
    try:
        if (self.lcb_nhanvien.get() >= 0 and self.snlv.get() >= 0 and self.trocap.get() >= 0) and (self.name_nhanvien.get().strip()):
            insert_query = sql.SQL("INSERT INTO {} (name, luongcoban, songaylamviec, trocap, luong) VALUES (%s, %s, %s, %s, %s)").format(sql.Identifier("nhanvien"))
            data_to_insert = (self.name_nhanvien.get(), self.lcb_nhanvien.get(), self.snlv.get(), self.trocap.get(), self.LuongNhanVien())
            self.cur.execute(insert_query, data_to_insert)
            self.conn.commit()
            messagebox.showinfo("Success", "Insert Nhan Vien successfully!")
        else:
            messagebox.showwarning("Warning", "Please check your input.")
    except Exception as e:
        messagebox.showerror("Error", f"Error inserting data: {e}")
```

n. def Search_LuongNhanVien

```
def Search_LuongNhanVien(self):
    try:
        timkiem = self.search_nhanvien.get()
        query_value = sql.SQL("SELECT * FROM {} WHERE name = %s").format(sql.Identifier("nhanvien"))
        self.cur.execute(query_value, (timkiem,))
        row_value = self.cur.fetchall()

        if timkiem == "*":
            query = sql.SQL("SELECT * FROM {}").format(sql.Identifier("nhanvien"))
            self.cur.execute(query)
            rows = self.cur.fetchall()
            self.scroll_NhanVien.delete(1.0, tk.END)
            for values in rows:
                for value in values:
                    self.scroll_NhanVien.insert(tk.END, f"{value} ")
                self.scroll_NhanVien.insert(tk.INSERT, "\n")
        elif not row_value:
            messagebox.showinfo("Thong Bao", "Khong tim thay nhan vien \nhoac do nhap sai")
        else:
            self.scroll_NhanVien.delete(1.0, tk.END)
            for values in row_value:
                for value in values:
                    self.scroll_NhanVien.insert(tk.END, f"{value} ")

    except Exception as e:
        messagebox.showerror("Error", f"Error loading data: {e}")
```

o. def Search_LuongQuanLy

```
def Search_LuongQuanLy(self):
    try:
        timkiem = self.search_quanly.get()
        query_value = sql.SQL("SELECT * FROM {} WHERE name = %s").format(sql.Identifier("quanly"))
        self.cur.execute(query_value, (timkiem,))
        row_value = self.cur.fetchall()

        if timkiem == "*":
            query = sql.SQL("SELECT * FROM {}").format(sql.Identifier("quanly"))
            self.cur.execute(query)
            rows = self.cur.fetchall()
            self.scroll_QuanLy.delete(1.0, tk.END)
            for values in rows:
                for value in values:
                    self.scroll_QuanLy.insert(tk.END, f"{value} ")
                self.scroll_QuanLy.insert(tk.INSERT, "\n")
        elif not row_value:
            messagebox.showinfo("Thong Bao", "Khong tim thay quan ly \nhoac do nhap sai")
        elif row_value:
            self.scroll_QuanLy.delete(1.0, tk.END)
            for values in row_value:
                for value in values:
                    self.scroll_QuanLy.insert(tk.END, f"{value} ")

    except Exception as e:
        messagebox.showerror("Error", f"Error loading data: {e}")
```

4. Github

GitHub - NhatQuangIT1301/pythonnc