



Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут»

## **Розрахунково-графічна робота**

«Створення додатку бази даних,  
орієнтованого на взаємодію з СУБД PostgreSQL»

Виконав студент групи: КВ-11

ПІБ: Горбуль А.О.

Telegram: [@AOSity](https://www.t.me/AOSity)

Перевірив:

## Звіт

Метою роботи є здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

Загальне завдання роботи полягає у наступному:

1. Реалізувати функції перегляду, внесення, редагування та видалення даних у таблицях бази даних, створених у лабораторній роботі №1, засобами консольного інтерфейсу.
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.
3. Забезпечити реалізацію пошуку за декількома атрибутами з двох та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.
4. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

### «Система управління резюме та вакансіями»

Перелік сутностей і опис їх призначення:

**Користувач (User)**, сутність призначена для збереження даних про користувача системи – ім'я, вік, пошту

**Резюме (Resume)**, сутність призначена для збереження даних про резюме завантажене користувачем для подальшого відгуку на вакансію

**Вакансія (Vacancy)**, сутність призначена для збереження даних про вакансію – назву позицій, деталі роботи

### Resume & Vacancy management system

User

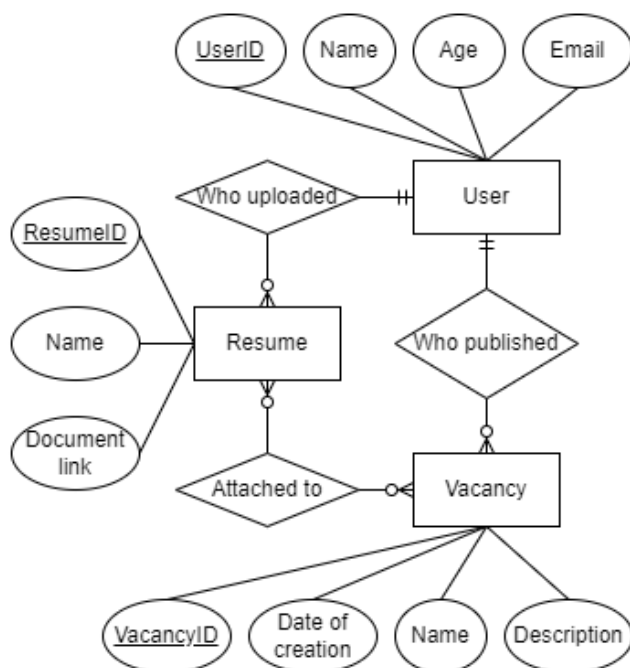
Resume

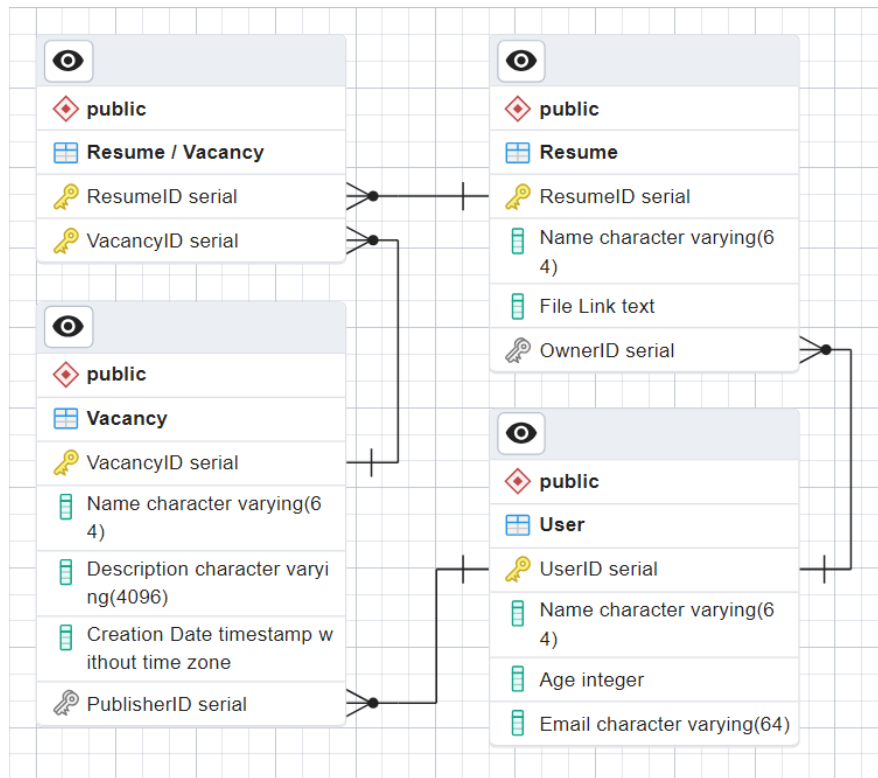
Vacancy

User can upload their resume

User can publish new vacancy

Resume can be attached to vacancy





## Меню користувача

### 1. Add – додати запис у відповідну таблицю

- 1.1. User
- 1.2. Resume
- 1.3. Vacancy
- 1.4. Resume/Vacancy
- 1.5. Cancel

Після вибору таблиці починається введення значень

### 2. View – переглянути відповідну таблицю

- 2.1. User
- 2.2. Resume
- 2.3. Vacancy
- 2.4. Resume/Vacancy
- 2.5. Cancel

Після вибору таблиці вона виводиться в консоль

### 3. Update – оновити запис у відповідній таблиці

- 3.1. User
- 3.2. Resume
- 3.3. Vacancy
- 3.4. Cancel

Після вибору таблиці можна вибрати запис та змінити його значення

### 4. Delete – видалити запис з відповідної таблиці

- 4.1. User
- 4.2. Resume
- 4.3. Vacancy
- 4.4. Cancel

Після вибору таблиці можна вибрати запис та видалити його

5. Generate – згенерувати дані для відповідної таблиці

5.1. User

5.2. Resume

5.3. Vacancy

5.4. Resume/Vacancy

5.5. Cancel

Після вибору таблиці можна вибрати кількість згенерованих записів та згенерувати

6. Find – знайти записи за певними критеріями

7. Quit – вихід з програми

## Пункт №1

Початкові таблиці:

ID	Name	Age	Email
1	Andrii	19	email@email.com

ID	Name	Link	OwnerID
1	Embedded Dev	gdrive.com/file/	1

Операція вилучення запису батьківської таблиці

Menu:

1. Add

2. View

3. Update

4. Delete

5. Generate

6. Find

7. Quit

Enter your choice: 4

1. User

2. Resume

3. Vacancy

4. Cancel

Enter your choice: 1

Enter ID: 1

User deleted successfully!

Таблиці після вилучення стали пустими:

Menu:

1. Add

2. View

3. Update

4. Delete

5. Generate

6. Find

7. Quit

Enter your choice: 2

1. User

2. Resume

3. Vacancy

4. Resume / Vacancy

5. Cancel

Enter your choice: 1

Table "User" is empty

Menu:

1. Add
2. View
3. Update
4. Delete
5. Generate
6. Find
7. Quit

Enter your choice: 2

1. User
2. Resume
3. Vacancy
4. Resume / Vacancy
5. Cancel

Enter your choice: 2

Table "Resume" is empty

Спроба додати запис в дочірню таблицю, без відповідного запису в батьківській:

Menu:

1. Add
2. View
3. Update
4. Delete
5. Generate
6. Find
7. Quit

Enter your choice: 1

1. User
2. Resume
3. Vacancy
4. Resume / Vacancy
5. Cancel

Enter your choice: 3

Enter vacancy name: Dev

Enter vacancy description: dev

Enter vacancy date: now

Enter publisher ID: 123

Foreign key violation!

## Пункт №2

Генерація 100 записів таблиці User

Menu:

1. Add
2. View
3. Update
4. Delete
5. Generate
6. Find
7. Quit

Enter your choice: 5

1. User
2. Resume
3. Vacancy
4. Resume / Vacancy
5. Cancel

Enter your choice: 1

Generated rows number: 100

100 user generated successfully!

Menu:

1. Add
2. View

3. Update  
 4. Delete  
 5. Generate  
 6. Find  
 7. Quit  
 Enter your choice: 2  
 1. User  
 2. Resume  
 3. Vacancy  
 4. Resume / Vacancy  
 5. Cancel  
 Enter your choice: 1

ID	Name	Age	Email
2	13451bd085e6f0d6	33	032a040d8f605a31@email.com
3	71bc95fb47b2c264	54	44a96161fbc8df32@email.com
4	7ad5ef33be7a2307	63	c02302bced5ba073@email.com
5	f1cb84bd78211eb5	8	806490f18a7c4c50@email.com
6	a34c543f60faefdb	10	9fcc377dd1a37b0a@email.com
7	85cfa9a5f1620a8a	63	e88ab46b74b4900c@email.com
8	9178ae25cdb694ab	23	fa5dd09fa33e00e6@email.com
9	821ac5b2dccdc3ce	1	08f72484551f28a7@email.com
...	...	...	...
90	b4178e6e265ceff1	40	5a1b373f14a9e141@email.com
91	86e447c73b150481	10	9fa3760069962b00@email.com
92	7b2b1f4efdbe52de	63	f3bfdbad2175e325@email.com
93	322e9fb6feb52783	40	42a9c7c19984b3ee@email.com
94	009fc47ab958b45c	54	5ebc0bdcd097fa51@email.com
95	435669b42b9a2a94	21	9107704698ce64b3@email.com
96	aa7a288c93124e27	57	6321c7de85f4115f@email.com
97	76c08951369ed137	68	e9da6d63ef4326f2@email.com
98	68f959a1502c346d	70	6f33dcd17e7cd4dc@email.com
99	624fc3530e6024ec	1	2bdc200f212ab450@email.com
100	1bb14d9cc5f3ae70	43	db450467832f6993@email.com
101	ba56b730cbbb136f	24	98f471ba84c646f8@email.com

Копії SQL-запитів для генерації записів таблиць:

```

INSERT INTO public."User" ("Name", "Age", "Email")
SELECT
  substring(md5(random()::text), 1, 16),
  trunc(random() * 70) + 1,
  substring(md5(random()::text), 1, 16) || '@email.com'
FROM generate_series(1, {number});
  
```

```

INSERT INTO public."Resume" ("Name", "File Link", "OwnerID")
SELECT
  substring(md5(random()::text), 1, 16),
  'domain.com/files/' || substring(md5(random()::text), 1, 8) || '/resume.pdf',
  (SELECT "UserID" FROM public."User" ORDER BY random() LIMIT 1
   OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM "User"))
FROM generate_series(1, {number});
  
```

```
INSERT INTO public."Vacancy" ("Name", "Description", "Creation Date", "PublisherID")
SELECT
    substring(md5(random()::text), 1, 16),
    substring(md5(random()::text), 1, 128),
    'now'::timestamp - random() * ('2023-12-31'::timestamp - '2022-01-01'::timestamp),
    (SELECT "UserID" FROM public."User" ORDER BY random() LIMIT 1
    OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM "User"))
FROM generate_series(1, {number});
```

```
INSERT INTO public."Resume / Vacancy" ("ResumeID", "VacancyID")
SELECT
    (SELECT "ResumeID" FROM public."Resume" ORDER BY random() LIMIT 1
    OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM "Resume")),
    (SELECT "VacancyID" FROM public."Vacancy" ORDER BY random() LIMIT 1
    OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM "Vacancy"))
FROM generate_series(1, {number})
ON CONFLICT ("ResumeID", "VacancyID") DO NOTHING;
```

## Пункт №3

Приклади пошуку за параметрами:

```
Enter user name (blank for any): abc
Enter minimum age (blank for any): 18
Enter maximum age (blank for any):
Enter resume name (blank for any):
Enter vacancy name (blank for any):
Enter minimum creation date (YYYY-MM-DD, blank for any):
Enter maximum creation date (YYYY-MM-DD, blank for any):
```

UserName	Age	Email	ResumeName	ResumeLink	VacancyName	VacancyCreationDate
61c8e1bd721abc66	21	99422f843e82a48d@email.com	de7f2107032f17d6	domain.com/files/32f71d46/resume.pdf	4232c541954fabfe	2023-07-28 08:05:43.300297
61c8e1bd721abc66	21	99422f843e82a48d@email.com	314fcf61019e373d	domain.com/files/23dcd3f1/resume.pdf	c51bc29fa844f36f	2022-09-26 14:13:29.905462
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	4b3e0ce5edbf5af4	2022-08-16 23:11:01.489753
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	cde691d95c85c1c0	2022-12-18 11:26:42.634704
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	8cd61adc895dc025	2022-05-18 05:28:03.688624
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	c119a960103dcabd	2022-10-29 02:05:24.237178

Execution time: 4.015922546386719ms

```
Enter user name (blank for any):
Enter minimum age (blank for any): 18
Enter maximum age (blank for any): 25
Enter resume name (blank for any):
Enter vacancy name (blank for any):
Enter minimum creation date (YYYY-MM-DD, blank for any):
Enter maximum creation date (YYYY-MM-DD, blank for any):
```

UserName	Age	Email	ResumeName	ResumeLink	VacancyName	VacancyCreationDate
9178ae25cdb694ab	23	fa5dd09fa33e00e6@email.com	891ea87636eca734	domain.com/files/051185d3/resume.pdf	83540a6102a35c83	2023-10-01 00:15:24.234838
60720e3abba642b3	24	3bbbc434ecf4af00@email.com	e029ee46cb19f3d6	domain.com/files/ed4fc53e/resume.pdf	aa037130a122a128	2023-05-18 12:24:40.606503
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	4b3e0ce5edbf5af4	2022-08-16 23:11:01.489753
ba56b730cbb136f	24	98f471ba84c646f8@email.com	1b881dbcd4d43fcc	domain.com/files/38d7df3e/resume.pdf	4b3e0ce5edbf5af4	2022-08-16 23:11:01.489753
69e017f6a597185e	25	4caabe4d088773bc@email.com	12955eae231ec884	domain.com/files/88df0a0c5/resume.pdf	8b1b886462b64ad5	2022-07-06 21:42:51.457396
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	cde691d95c85c1c0	2022-12-18 11:26:42.634704
9178ae25cdb694ab	23	fa5dd09fa33e00e6@email.com	623552b7ada6d373	domain.com/files/5a07f374/resume.pdf	9c0a9a72df0f1b0e	2023-03-17 20:51:10.765155
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	8cd61adc895dc025	2022-05-18 05:28:03.688624
435669b42b9a2a94	21	9107704698ce64b3@email.com	b84491b3a0d2053b	domain.com/files/75e9f2fb/resume.pdf	40601e2bc5b18b95	2023-01-01 19:45:18.749100
61c8e1bd721abc66	21	99422f843e82a48d@email.com	de7f2107032f17d6	domain.com/files/32f71d46/resume.pdf	4232c541954fabfe	2023-07-28 08:05:43.300297
9178ae25cdb694ab	23	fa5dd09fa33e00e6@email.com	623552b7ada6d373	domain.com/files/5a07f374/resume.pdf	3f65078cd8242324	2023-10-19 19:40:15.955336
69e017f6a597185e	25	4caabe4d088773bc@email.com	12955eae231ec884	domain.com/files/88df0a0c5/resume.pdf	8e7c8be79e47d06d	2022-02-07 06:01:00.925040
60720e3abba642b3	24	3bbbc434ecf4af00@email.com	054089bf174174ef	domain.com/files/d9c40371/resume.pdf	cb4f5338679fd707	2022-06-27 14:54:47.943413
9178ae25cdb694ab	23	fa5dd09fa33e00e6@email.com	623552b7ada6d373	domain.com/files/5a07f374/resume.pdf	39611bc4b58badbe	2023-07-22 23:14:22.094071
ba56b730cbb136f	24	98f471ba84c646f8@email.com	1b881dbcd4d43fcc	domain.com/files/38d7df3e/resume.pdf	3959bfa4360c9df9	2022-10-04 21:01:53.636274
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	c119a960103dcabd	2022-10-29 02:05:24.237178
435669b42b9a2a94	21	9107704698ce64b3@email.com	b84491b3a0d2053b	domain.com/files/75e9f2fb/resume.pdf	a0ba012e3819c0b3	2023-04-03 13:31:16.022645
61c8e1bd721abc66	21	99422f843e82a48d@email.com	314fcf61019e373d	domain.com/files/23dcd3f1/resume.pdf	c51bc29fa844f36f	2022-09-26 14:13:29.905462
69e017f6a597185e	25	4caabe4d088773bc@email.com	12955eae231ec884	domain.com/files/88df0a0c5/resume.pdf	e142bf5c6fe1e9eb	2022-12-03 15:50:38.768495
435669b42b9a2a94	21	9107704698ce64b3@email.com	b84491b3a0d2053b	domain.com/files/75e9f2fb/resume.pdf	c6d52f1601445d47	2022-10-12 22:19:32.901943

Execution time: 0.995635986328125ms

```

Enter user name (blank for any):
Enter minimum age (blank for any):
Enter maximum age (blank for any):
Enter resume name (blank for any):
Enter vacancy name (blank for any): 4b3e0ce5edbf5af4
Enter minimum creation date (YYYY-MM-DD, blank for any):
Enter maximum creation date (YYYY-MM-DD, blank for any):

```

UserName	Age	Email	ResumeName	ResumeLink	VacancyName	VacancyCreationDate
ba56b730cbbb136f	24	98f471ba84c646f8@email.com	1b881dbcd4d43fcc	domain.com/files/38d7df3e/resume.pdf	4b3e0ce5edbf5af4	2022-08-16 23:11:01.489753
a2e84ecbb554dc22	30	462eadefce5b881c@email.com	e6cddf62a1dd31ff	domain.com/files/69acc2b9/resume.pdf	4b3e0ce5edbf5af4	2022-08-16 23:11:01.489753
61c8e1bd721abc66	21	99422f843e82a48d@email.com	155e1394b63f181c	domain.com/files/493dcf42/resume.pdf	4b3e0ce5edbf5af4	2022-08-16 23:11:01.489753
07b1237649e0df42	59	380b87fc421846c5@email.com	3d6b1106022bc2e0	domain.com/files/a8eaa9b9/resume.pdf	4b3e0ce5edbf5af4	2022-08-16 23:11:01.489753

```

Execution time: 0.6761550903320312ms

```

Копії SQL-запитів для пошуку:

```

SELECT "User"."Name" AS "UserName",
       "User"."Age",
       "User"."Email",
       "Resume"."Name" AS "ResumeName",
       "Resume"."File Link" AS "ResumeLink",
       "Vacancy"."Name" AS "VacancyName",
       "Vacancy"."Creation Date" AS "VacancyCreationDate"
FROM "User"
JOIN "Resume" ON "User"."UserID" = "Resume"."OwnerID"
JOIN "Resume / Vacancy" ON "Resume"."ResumeID" = "Resume / Vacancy"."ResumeID"
JOIN "Vacancy" ON "Resume / Vacancy"."VacancyID" = "Vacancy"."VacancyID"
WHERE TRUE

```

В залежності від введених користувачем критеріїв пошуку до запиту додаються:

```

if user_name:
    query += f' AND "User"."Name" LIKE \'{"%" + user_name + "%"}\''
if min_age:
    query += f' AND "User"."Age" >= {min_age}'
if max_age:
    query += f' AND "User"."Age" <= {max_age}'
if resume_name:
    query += f' AND "Resume"."Name" LIKE \'{"%" + resume_name + "%"}\''
if vacancy_name:
    query += f' AND "Vacancy"."Name" LIKE \'{"%" + vacancy_name + "%"}\''
if min_creation_date:
    query += f' AND "Vacancy"."Creation Date" >= {min_creation_date}'
if max_creation_date:
    query += f' AND "Vacancy"."Creation Date" <= {max_creation_date}'

```



## Пункт №4

Программный код модуля model:

```
import psycopg2
import time
from db_setup import db_setup_script

class Model:
    def __init__(self):
        self.conn = psycopg2.connect(
            dbname='postgres',
            user='postgres',
            password='postgres',
            host='localhost',
            port=5432
        )
        self.create_tables()

    def create_tables(self):
        try:
            c = self.conn.cursor()
            c.execute(db_setup_script)
            self.conn.commit()
        except Exception:
            self.conn.rollback()

    def add_user(self, name, age, email):
        try:
            c = self.conn.cursor()
            c.execute('INSERT INTO "User" ("Name", "Age", "Email") VALUES (%s, %s, %s)',
                (name, age, email))
            self.conn.commit()
        except psycopg2.errors.InvalidTextRepresentation:
            self.conn.rollback()
            return "Input error!"
        except psycopg2.errors.ForeignKeyViolation:
            self.conn.rollback()
            return "Foreign key violation!"
        return "User added successfully!"

    def add_resume(self, name, file_link, owner_id):
        try:
            c = self.conn.cursor()
            c.execute('INSERT INTO "Resume" ("Name", "File Link", "OwnerID") VALUES (%s, %s, %s)', (name, file_link, owner_id))
            self.conn.commit()
        except psycopg2.errors.InvalidTextRepresentation:
            self.conn.rollback()
            return "Input error!"
        except psycopg2.errors.ForeignKeyViolation:
            self.conn.rollback()
            return "Foreign key violation!"
        return "Resume added successfully!"
```

```

def add_vacancy(self, name, description, date, publisher_id):
    try:
        c = self.conn.cursor()
        c.execute('INSERT INTO "Vacancy" ("Name", "Description", "Creation Date",
"PublisherID") VALUES (%s, %s, %s, %s)',
                (name, description, date, publisher_id))
        self.conn.commit()
    except psycopg2.errors.InvalidTextRepresentation:
        self.conn.rollback()
        return "Input error!"
    except psycopg2.errors.ForeignKeyViolation:
        self.conn.rollback()
        return "Foreign key violation!"
    return "Vacancy added successfully!"

def add_resvac(self, resume_id, vacancy_id):
    try:
        c = self.conn.cursor()
        c.execute('INSERT INTO "Resume / Vacancy" ("ResumeID", "VacancyID") VALUES
(%s, %s)', (resume_id, vacancy_id))
        self.conn.commit()
    except psycopg2.errors.InvalidTextRepresentation:
        self.conn.rollback()
        return "Input error!"
    except psycopg2.errors.ForeignKeyViolation:
        self.conn.rollback()
        return "Foreign key violation!"
    return "Resume/Vacancy relation added successfully!"

def get_table(self, table_name):
    c = self.conn.cursor()
    c.execute(f'SELECT * FROM "{table_name}"')
    return c.fetchall()

def update_user(self, id, name, age, email):
    c = self.conn.cursor()
    c.execute('UPDATE "User" SET "Name"=%s, "Age"=%s, "Email"=%s WHERE "UserID"=%s',
(name, age, email, id))
    self.conn.commit()

def update_resume(self, id, name, file_link, owner_id):
    c = self.conn.cursor()
    c.execute('UPDATE "Resume" SET "Name"=%s, "File Link"=%s, "OwnerID"=%s WHERE
"ResumeID"=%s', (name, file_link, owner_id, id))
    self.conn.commit()

def update_vacancy(self, id, name, description, date, publisher_id):
    c = self.conn.cursor()
    c.execute('UPDATE "Vacancy" SET "Name"=%s, "Description"=%s, "Creation Date"=%s,
"PublisherID"=%s WHERE "VacancyID"=%s',
                (name, description, date, publisher_id, id))
    self.conn.commit()

```

```

def delete(self, table_name, id):
    c = self.conn.cursor()
    c.execute(f'DELETE FROM "{table_name}" WHERE "{table_name}ID"={id}')
    self.conn.commit()

def generate_user(self, number):
    c = self.conn.cursor()
    c.execute(f'''
        INSERT INTO public."User" ("Name", "Age", "Email")
        SELECT
            substring(md5(random()::text), 1, 16),
            trunc(random() * 70) + 1,
            substring(md5(random()::text), 1, 16) || '@email.com'
        FROM generate_series(1, {number});
    ''')
    self.conn.commit()
    return f"{number} user generated successfully!"

def generate_resume(self, number):
    try:
        c = self.conn.cursor()
        c.execute(f'''
            INSERT INTO public."Resume" ("Name", "File Link", "OwnerID")
            SELECT
                substring(md5(random()::text), 1, 16),
                'domain.com/files/' || substring(md5(random()::text), 1, 8)
                || '/resume.pdf',
                (SELECT "UserID" FROM public."User" ORDER BY random() LIMIT 1
                OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM
                "User"))
            FROM generate_series(1, {number});
        ''')
        self.conn.commit()
    except psycopg2.errors.DivisionByZero:
        self.conn.rollback()
        return "Generation failed, no users to assign OwnerID"
    return f"{number} resume generated successfully!"

def generate_vacancy(self, number):
    try:
        c = self.conn.cursor()
        c.execute(f'''
            INSERT INTO public."Vacancy" ("Name", "Description", "Creation
            Date", "PublisherID")
            SELECT
                substring(md5(random()::text), 1, 16),
                substring(md5(random()::text), 1, 128),
                'now'::timestamp - random() * ('2023-12-31'::timestamp -
                '2022-01-01'::timestamp),
                (SELECT "UserID" FROM public."User" ORDER BY random() LIMIT 1
                OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM
                "User"))
            FROM generate_series(1, {number});
        ''')

```

```

        self.conn.commit()
    except psycopg2.errors.DivisionByZero:
        self.conn.rollback()
        return "Generation failed, no users to assign PublisherID"
    return f"{number} vacancy generated successfully!"

def generate_resvac(self, number):
    try:
        c = self.conn.cursor()
        c.execute(f'''
            INSERT INTO public."Resume / Vacancy" ("ResumeID", "VacancyID")
            SELECT
                (SELECT "ResumeID" FROM public."Resume" ORDER BY random()
LIMIT 1
                OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM
"Resume")),
                (SELECT "VacancyID" FROM public."Vacancy" ORDER BY random()
LIMIT 1
                OFFSET (random()*generate_series)::int%(SELECT COUNT(*) FROM
"Vacancy"))
            FROM generate_series(1, {number})
            ON CONFLICT ("ResumeID", "VacancyID") DO NOTHING;
        ''')
        self.conn.commit()
    except psycopg2.errors.DivisionByZero:
        self.conn.rollback()
        return "Generation failed, no available ResumeID and/or VacancyID"
    return f"{number} Resume/Vacancy relation generated successfully!"

def find(self, user_name, min_age, max_age, resume_name, vacancy_name,
min_creation_date, max_creation_date):
    try:
        query = ('''
            SELECT "User"."Name" AS "UserName",
                "User"."Age",
                "User"."Email",
                "Resume"."Name" AS "ResumeName",
                "Resume"."File Link" AS "ResumeLink",
                "Vacancy"."Name" AS "VacancyName",
                "Vacancy"."Creation Date" AS "VacancyCreationDate"
            FROM "User"
            JOIN "Resume" ON "User"."UserID" = "Resume"."OwnerID"
            JOIN "Resume / Vacancy" ON "Resume"."ResumeID" = "Resume /
Vacancy"."ResumeID"
            JOIN "Vacancy" ON "Resume / Vacancy"."VacancyID" =
"Vacancy"."VacancyID"
            WHERE TRUE
        ''')

        if user_name:
            query += f' AND "User"."Name" LIKE \'{"%" + user_name + "%"}\''
        if min_age:
            query += f' AND "User"."Age" >= {min_age}'
        if max_age:

```

```

        query += f' AND "User"."Age" <= {max_age}'
    if resume_name:
        query += f' AND "Resume"."Name" LIKE \'{"%" + resume_name + "%"}\''
    if vacancy_name:
        query += f' AND "Vacancy"."Name" LIKE \'{"%" + vacancy_name + "%"}\''
    if min_creation_date:
        query += f' AND "Vacancy"."Creation Date" >= {min_creation_date}'
    if max_creation_date:
        query += f' AND "Vacancy"."Creation Date" <= {max_creation_date}'

    c = self.conn.cursor()
    start_time = time.time()
    c.execute(query)
    end_time = time.time()
    return (c.fetchall(), end_time - start_time)
except Exception:
    self.conn.rollback()
    return

```

Опис функцій:

\_\_init\_\_ – функція ініціалізації об'єкту, в ній відбувається підключення до бази даних

create\_tables – функція створення таблиць бази даних, якщо вони не існують

add\_<table\_name> – додати запис до відповідної таблиці

get\_table – отримати всі записи з таблиці, для подальшого виведення на екран

update\_<table\_name> – змінити запис відповідної таблиці

delete – видалити запис з таблиці

generate\_<table\_name> – згенерувати N записів відповідної таблиці

find – функція пошуку записів за певними параметрами