НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ
Кафедра системного програмування та спеціалізованих комп'ютерних систем

Лабораторна робота №2

з дисципліни
«Бази даних і засоби управління»

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

Виконала: студентка III курсу
ФПМ групи КВ-93
Коломієць Анна

Київ – 2021

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

Постановка задачі

1. Реалізувати функції перегляду, внесення, редагування та вилучення 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).

Інформація про програму

Посилання на GitHub: https://github.com/AnnaKolomiietss/DB_lab2
Мова програмування: Python 3.9
Використані бібліотеки: psycopg2 (для встановлення зв'язку з СУБД), time (для виміру часу пошуку), sys (для реалізації інтерфейсу)

Відомості про предметну галузь «Аптека» з лабораторної роботи №1

Опис структури БД «Аптека»

Предметна галузь «Аптека» описує схему замовлень лікарських засобів. Згідно цієї області для побудови бази даних було виділено наступні сутності:

1. Category: вміщує атрибути ID, name та description. Слугує для зберігання інформації про певну категорію ліків;
2. Pill: вміщує атрибути ID, name, price, manufacturer_id. Слугує для зберігання інформації про конкретний лікарський засіб;
3. Category_pill: вміщує ID препарату та ID категорії. Слугує для зв'язування лікарського засобу та її категорії;
4. Manufacturer: вміщує атрибути ID, name, country, email. Слугує для зберігання інформації про виробника.

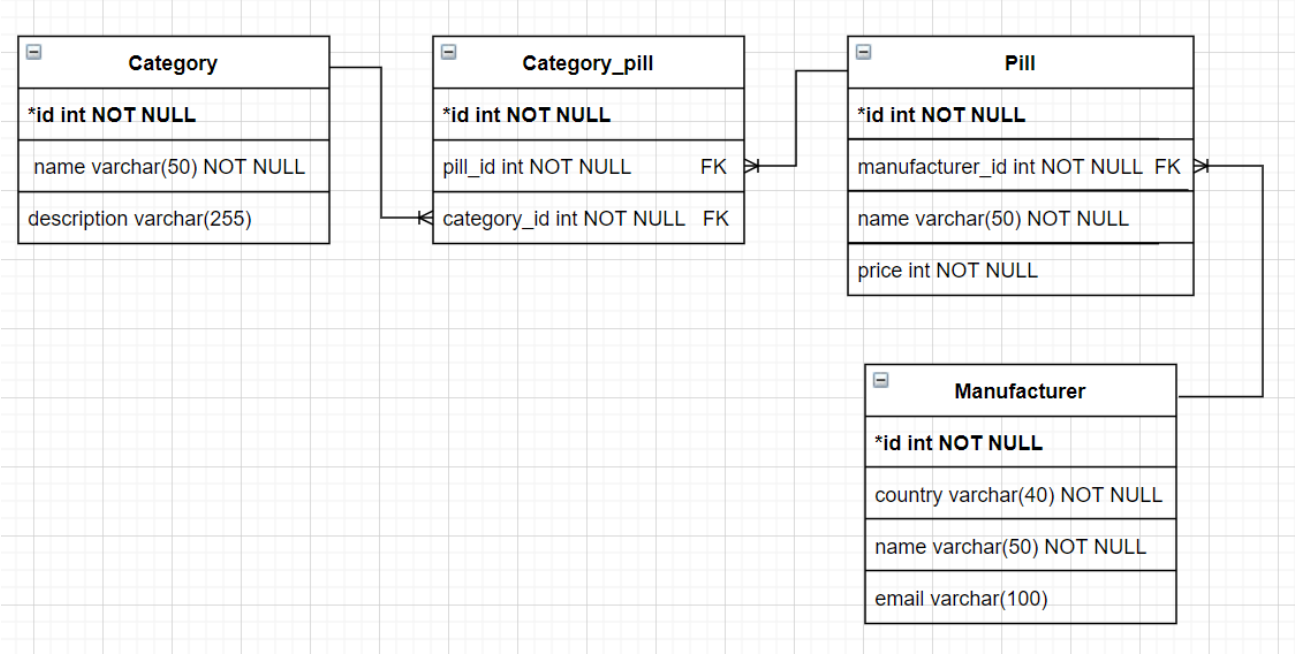

Рисунок 1 - Логічна модель предметної області «Аптека»

Таблиця 1 - Опис структури БД «Аптека»

Відношення	Атрибут	Тип
Сутність "Category" Вміщує інформацію про окрему категорію ліків	**id** – унікальний ID категорії в БД, не допускає NULL **name** – назва категорії, не допускає NULL **description** – опис категорії	Числовий Текстовий(50) Текстовий(255)
Сутність "Pill" Вміщує інформацію про конкретний препарат	**id** – унікальний ID препарату в БД, не допускає NULL **name** – назва препарату, не допускає NULL **price** – ціна, не допускає NULL **manufacturer_id** – ID виробника препаратуб, не допускає NULL	Числовий Текстовий(50) Числовий Числовий
Сутність"Category_pill" Вміщує інформацію про категорію та препарат	**id** – унікальний ID сутності в БД, не допускає NULL **category_id** – ID категорії препарату, не допускає NULL **pill_id** – ID препарату, не допускає NULL	Числовий Числовий Числовий
Сутність"Manufacturer" Вміщує інформацію про виробника	**id** – унікальний ID виробника в БД, не допускає NULL **country** – країна виробника, не допускає NULL **name** –назва виробника, не допускає NULL **email** – email виробника	Числовий Текстовий(40) Текстовий(50) Текстовий(100)

Меню користувача

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py menu
print_table - table output
        argument (table_name) is required
delete_record - deletes the record from table
        arguments (table_name, key_name, key_value) are required
update_record - updates record with specified id in table
        Category args (table_name, id, name, description)
        Category_pill args (table_name, id, pill_id, category_id)
        Pill args (table_name, id, manufacturer_id, name, price)
        Manufacturer args (table_name, id, country, name, email)
insert_record - inserts record into table
        Category args (table_name, id, name, name, description)
        Category_pill args (table_name, id, pill_id, category_id)
        Pill args (table_name, id, manufacturer_id, name, price)
        Manufacturer args (table_name, id, country, name, email)
generate_randomly - generates n random records in table
        arguments (table_name, n) are required
search_records - search for records in two or more tables using one or more keys
        arguments (table1_name, table2_name, table1_key, table2_key) are required,
        if you want to perform search in more tables:
        (table1_name, table2_name, table3_name, table1_key, table2_key, table3_key, table13_key)
        (table1_name, table2_name, table3_name, table4_name, table1_key, table2_key, table3_key, table13_key, table4_key, table24_key)
```

Рисунок 2 – меню користувача додатку БД «Аптека»

На рисунку 2 зображено виконання команди “menu”, яка показує всі доступні для виконання команди та аргументи, які вони приймають.

При виклику команда запускає відповідний метод у файлі “Controller.py”, який в свою чергу передає значення у файл “View”. Після перевірки на правильність внесення даних користувачем, запит передається у файл “Model.py”, де здійснюється запит до бази даних.

Завдання 1

Методи реалізації:

1. Print_table – виводить вміст таблиці у вікно терміналу. За аргумент приймає назву таблиці.
2. Delete_record – видаляє запис з таблиці. За аргументи приймає назву таблиці, первинний ключ та його значення.
3. Update_record – змінює усі поля запису (окрім первинного ключа). Аргументи різні для для кожної таблиці:
    1) для таблиці “Category”: id, name, description;
    2) для таблиці “Category_pill”: id, pill_id, category_id;
    3) для таблиці “Pill”: id, manufacturer_id, name, price;
    4) для таблиці “Manufacturer”: id, country, email.
4. Insert_record – вставляє новий рядок у обрану таблицю. Аргументи різні для для кожної таблиці:
    1) для таблиці “Category”: id, name, description;
    2) для таблиці “Category_pill”: id, pill_id, category_id;

3) для таблиці “Pill”: id, manufacturer_id, name, price;
4) для таблиці “Manufacturer”: id, country, email.

Запит на внесення даних

Внесення запису до таблиці “Pill”

Таблиця до вставки поля:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100
----------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120
----------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40
----------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20
----------------------------------------------------------------
id: 5   manufacturer_id: 2      name: aaa       price: 120
----------------------------------------------------------------
id: 7   manufacturer_id: 4      name: guragin   price: 1000
----------------------------------------------------------------
```

Таблиця після вставки поля з id 8:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py insert_record Pill 8 2 anaferon 200
select count(*) from public."Pill" where id=8
select count(*) from public."Manufacturer" where id=2
insert into public."Pill" (id, manufacturer_id, name, price) VALUES (8, '2', 'anaferon', '200')
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100
----------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120
----------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40
----------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20
----------------------------------------------------------------
id: 5   manufacturer_id: 2      name: aaa       price: 120
----------------------------------------------------------------
id: 7   manufacturer_id: 4      name: guragin   price: 1000
----------------------------------------------------------------
id: 8   manufacturer_id: 2      name: anaferon  price: 200
```

Внесення запису до таблиці “Category”

Таблиця до вставки поля:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category
SELECT * FROM public."Category"
Category table:
id: 1   name: sedative  description: cause soothing or a decrease in emotional stress without a hypnotic effect
-----------------------------------
id: 2   name: antipyretic       description: used to reduce high body temperature in case of fever
-----------------------------------
id: 3   name: analgetic
        description: designed to relieve pain
-----------------------------------
id: 4   name: antiphlogistic    description: a group of drugs that can relieve symptoms of inflammation
-----------------------------------
```

Таблиця після вставки поля з id 5:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py insert_record Category 5 anti-infective against
select count(*) from public."Category" where id=5
insert into public."Category" (id, name, description) VALUES (5, 'anti-infective', 'against');
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category
SELECT * FROM public."Category"
Category table:
id: 1   name: sedative  description: cause soothing or a decrease in emotional stress without a hypnotic effect
-----------------------------------
id: 2   name: antipyretic       description: used to reduce high body temperature in case of fever
-----------------------------------
id: 3   name: analgetic
        description: designed to relieve pain
-----------------------------------
id: 4   name: antiphlogistic    description: a group of drugs that can relieve symptoms of inflammation
-----------------------------------
id: 5   name: anti-infective    description: against
-----------------------------------
```

Внесення запису до таблиці “Category_pill”

Таблиця до вставки поля:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3
---------------------------------------
id: 3   pill_id: 2      category_id: 4
---------------------------------------
id: 4   pill_id: 3      category_id: 1
---------------------------------------
id: 9   pill_id: 2      category_id: 1
---------------------------------------
id: 1   pill_id: 2      category_id: 1
---------------------------------------
id: 10  pill_id: 2      category_id: 1
---------------------------------------
```

Таблиця після вставки поля з id 11:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py insert_record Category_pill 11 8 5
select count(*) from public."Pill" where id=8
select count(*) from public."Category" where id=5
insert into public."Category_pill" (id, pill_id, category_id) VALUES (11, '8', '5');
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3
-----------------------------------
id: 3   pill_id: 2      category_id: 4
-----------------------------------
id: 4   pill_id: 3      category_id: 1
-----------------------------------
id: 9   pill_id: 2      category_id: 1
-----------------------------------
id: 1   pill_id: 2      category_id: 1
-----------------------------------
id: 10  pill_id: 2      category_id: 1
-----------------------------------
id: 11  pill_id: 8      category_id: 5
```

Спроба вставки запису з неіснуючими даними:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py insert_record Category_pill 12 9 6
select count(*) from public."Pill" where id=9
select count(*) from public."Category" where id=6
insert into public."Category_pill" (id, pill_id, category_id) VALUES (12, '0', '0');
ОШИБКА:  INSERT или UPDATE в таблице "Category_pill" нарушает ограничение внешнего ключа "fk_category_pill_category"
DETAIL:  Ключ (category_id)=(0) отсутствует в таблице "Category".
```

Запит на зміну даних

Внесення змін до запису у таблиці “Manufacurer”

Таблиця до внесення змін:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Manufacturer
SELECT * FROM public."Manufacturer"
Manufacturer table:
id: 1   name: reckitt   country: England        email reckitt@gmail.com
----------------------------------------
id: 2   name: Valenta Pharm     country: Russia         email valentapharm@gmail.com
----------------------------------------
id: 3   name: Ternopharm        country: Ukraine        email ternopharm@gmail.com
----------------------------------------
id: 4   name: lufrpbzjrychl     country: xdoxrd         email ihhytlbpicmgyrrb
----------------------------------------
```

Таблиця після зміни запису з id 4:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py update_record Manufacturer 4 UVP Ukraine uvp@gmail.com
select count(*) from public."Manufacturer" where id=4
UPDATE public."Manufacturer" SET name='UVP', country='Ukraine', email='uvp@gmail.com' WHERE id=4;
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Manufacturer
SELECT * FROM public."Manufacturer"
Manufacturer table:
id: 1   name: reckitt   country: England        email reckitt@gmail.com
----------------------------------------
id: 2   name: Valenta Pharm     country: Russia         email valentapharm@gmail.com
----------------------------------------
id: 3   name: Ternopharm        country: Ukraine        email ternopharm@gmail.com
----------------------------------------
id: 4   name: UVP       country: Ukraine        email uvp@gmail.com
----------------------------------------
```

Внесення змін у таблицю “Pill”

Таблиця до внесення змін:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100
------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120
------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40
------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20
------------------------------------------------------------
id: 5   manufacturer_id: 2      name: aaa       price: 120
------------------------------------------------------------
id: 7   manufacturer_id: 4      name: guragin   price: 1000
------------------------------------------------------------
id: 8   manufacturer_id: 2      name: anaferon  price: 200
```

Таблиця після внесення змін у запис з id 8:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py update_record Pill 8 4 anaferon 250
select count(*) from public."Pill" where id=8
select count(*) from public."Manufacturer" where id=4
UPDATE public."Pill" SET manufacturer_id='4', name='anaferon', price='250' WHERE id=8;
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100
------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120
------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40
------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20
------------------------------------------------------------
id: 5   manufacturer_id: 2      name: aaa       price: 120
------------------------------------------------------------
id: 7   manufacturer_id: 4      name: guragin   price: 1000
------------------------------------------------------------
id: 8   manufacturer_id: 4      name: anaferon  price: 250
```

Внесення змін у таблицю “Category_pill”

Таблиця до внесення змін:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3
-------------------------------------
id: 3   pill_id: 2      category_id: 4
-------------------------------------
id: 4   pill_id: 3      category_id: 1
-------------------------------------
id: 9   pill_id: 2      category_id: 1
-------------------------------------
id: 1   pill_id: 2      category_id: 1
-------------------------------------
id: 10  pill_id: 2      category_id: 1
-------------------------------------
id: 11  pill_id: 8      category_id: 5
```

Таблиця після внесення змін до запису з id 11:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py update_record Category_pill 11 8 1
select count(*) from public."Pill" where id=8
select count(*) from public."Category" where id=1
UPDATE public."Category_pill" SET pill_id='8', category_id='1'  WHERE id=11;
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3
-------------------------------------
id: 3   pill_id: 2      category_id: 4
-------------------------------------
id: 4   pill_id: 3      category_id: 1
-------------------------------------
id: 9   pill_id: 2      category_id: 1
-------------------------------------
id: 1   pill_id: 2      category_id: 1
-------------------------------------
id: 10  pill_id: 2      category_id: 1
-------------------------------------
id: 11  pill_id: 8      category_id: 1
```

Спроба внесення змін до таблиці з неіснуючими даними category_id 20:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py update_record Category_pill 11 8 20
select count(*) from public."Pill" where id=8
select count(*) from public."Category" where id=20
UPDATE public."Category_pill" SET pill_id='8', category_id='0'  WHERE id=11;
ОШИБКА:  INSERT или UPDATE в таблице "Category_pill" нарушает ограничение внешнего ключа "fk_category_pill_category"
DETAIL:  Ключ (category_id)=(0) отсутствует в таблице "Category".
```

Запит на видалення:

Видалення запису з таблиці “Category_pill”

Таблиця до видалення запису:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3
------------------------------------
id: 3   pill_id: 2      category_id: 4
------------------------------------
id: 4   pill_id: 3      category_id: 1
------------------------------------
id: 9   pill_id: 2      category_id: 1
------------------------------------
id: 1   pill_id: 2      category_id: 1
------------------------------------
id: 10  pill_id: 2      category_id: 1
------------------------------------
id: 11  pill_id: 8      category_id: 1
```

Таблиця після видалення запису з id 10:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py delete_record Category_pill id 11
select count(*) from public."Category_pill" where id=11
DELETE FROM public."Category_pill" WHERE id=11;
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3
------------------------------------
id: 3   pill_id: 2      category_id: 4
------------------------------------
id: 4   pill_id: 3      category_id: 1
------------------------------------
id: 9   pill_id: 2      category_id: 1
------------------------------------
id: 1   pill_id: 2      category_id: 1
------------------------------------
id: 10  pill_id: 2      category_id: 1
```

Видалення запису з таблиці “Pill”

Таблиця до видалення запису:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100
------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120
------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40
------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20
------------------------------------------------------------
id: 5   manufacturer_id: 2      name: xfroiur   price: 56114
------------------------------------------------------------
id: 6   manufacturer_id: 3      name: poerkbox  price: 32149
------------------------------------------------------------
```

Таблиця після видалення запиту з id 6:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py delete_record Pill id 6
select count(*) from public."Pill" where id=6
select count(*) from public."Category_pill" where pill_id=6
DELETE FROM public."Pill" WHERE id=6;
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100
------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120
------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40
------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20
------------------------------------------------------------
id: 5   manufacturer_id: 2      name: xfroiur   price: 56114
```

Видалення запису з таблиці “Manufacurer”

Таблиця до видалення запису:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Manufacturer
SELECT * FROM public."Manufacturer"
Manufacturer table:
id: 1   name: reckitt   country: England        email reckitt@gmail.com
----------------------------------------
id: 2   name: Valenta Pharm     country: Russia         email valentapharm@gmail.com
----------------------------------------
id: 3   name: Ternopharm        country: Ukraine        email ternopharm@gmail.com
----------------------------------------
id: 4   name: UVP       country: Ukraine        email uvp@gmail.com
----------------------------------------
id: 5   name: dtnlwqyvmpm       country: upjwt  email ipjidfsstlnfrpqov
----------------------------------------
```

Таблиця після видалення запису з id 5:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py delete_record Manufacturer id 5
select count(*) from public."Manufacturer" where id=5
select count(*) from public."Pill" where manufacturer_id=5
DELETE FROM public."Manufacturer" WHERE id=5;
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Manufacturer
SELECT * FROM public."Manufacturer"
Manufacturer table:
id: 1   name: reckitt   country: England        email reckitt@gmail.com

----------------------------------------
id: 2   name: Valenta Pharm     country: Russia         email valentapharm@gmail.com

----------------------------------------
id: 3   name: Ternopharm        country: Ukraine        email ternopharm@gmail.com

----------------------------------------
id: 4   name: UVP       country: Ukraine        email uvp@gmail.com
```

Спроба видалити запис, який використовується у таблиці Pill:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py delete_record Manufacturer id 2
select count(*) from public."Manufacturer" where id=2
select count(*) from public."Pill" where id=2
this record is connected with another table, deleting will throw error
```

Завдання 2

Методи реалізації:

1. Generate_randomly – здійснює додавання до таблиці обраної користувачем кількісті рандомізованих записів. За аргументи приймає назву таблиці та число рандомізованих записів.

Вставка 4 рандомізованих записів до таблиці “Category”
Таблиця до вставки рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category
SELECT * FROM public."Category"
Category table:
id: 1   name: sedative  description: cause soothing or a decrease in emotional stress without a hypnotic effect

-----------------------------------
id: 2   name: antipyretic       description: used to reduce high body temperature in case of fever

-----------------------------------
id: 3   name: analgetic
        description: designed to relieve pain

-----------------------------------
id: 4   name: antiphlogistic    description: a group of drugs that can relieve symptoms of inflammation

-----------------------------------
id: 5   name: anti-infective    description: against

-----------------------------------
```

Запит:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py generate_randomly Category 4
insert into public."Category"select (SELECT MAX(id)+1 FROM public."Category"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)
FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)                FROM generat
e_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), '');
insert into public."Category"select (SELECT MAX(id)+1 FROM public."Category"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)
FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)                FROM generat
e_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), '');
insert into public."Category"select (SELECT MAX(id)+1 FROM public."Category"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)
FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)                FROM generat
e_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), '');
insert into public."Category"select (SELECT MAX(id)+1 FROM public."Category"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)
FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer)                FROM generat
e_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), '');
```

Таблиця після вставки чотирьох рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category
SELECT * FROM public."Category"
Category table:
id: 1   name: sedative  description: cause soothing or a decrease in emotional stress without a hypnotic effect

-------------------------------------
id: 2   name: antipyretic      description: used to reduce high body temperature in case of fever

-------------------------------------
id: 3   name: analgetic
        description: designed to relieve pain

-------------------------------------
id: 4   name: antiphlogistic    description: a group of drugs that can relieve symptoms of inflammation

-------------------------------------
id: 5   name: anti-infective    description: against

-------------------------------------
id: 6   name: npyktq    description: rwxnalrz

-------------------------------------
id: 7   name: kcenpo    description: xbvtfu

-------------------------------------
id: 8   name: cceot     description: cltjtt

-------------------------------------
id: 9   name: rhhx      description: lwafrdppe
```

Вставка 4 рандомізованих записів до таблиці “Pill”

Таблиця до вставки рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100

-----------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120

-----------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40

-----------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20

-----------------------------------------------------------------
```

Запит:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py generate_randomly Pill 4
insert into public."Pill" select (SELECT MAX(id)+1 FROM public."Pill"), (SELECT id FROM public."Manufacturer" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Man
ufacturer")-1)))), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(15-5)+5):: integer)), ''), FLOOR(RANDOM()*(10
0000-1)+1);
insert into public."Pill" select (SELECT MAX(id)+1 FROM public."Pill"), (SELECT id FROM public."Manufacturer" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Man
ufacturer")-1)))), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(15-5)+5):: integer)), ''), FLOOR(RANDOM()*(10
0000-1)+1);
insert into public."Pill" select (SELECT MAX(id)+1 FROM public."Pill"), (SELECT id FROM public."Manufacturer" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Man
ufacturer")-1)))), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(15-5)+5):: integer)), ''), FLOOR(RANDOM()*(10
0000-1)+1);
insert into public."Pill" select (SELECT MAX(id)+1 FROM public."Pill"), (SELECT id FROM public."Manufacturer" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Man
ufacturer")-1)))), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(15-5)+5):: integer)), ''), FLOOR(RANDOM()*(10
0000-1)+1);
```

Таблиця після вставки чотирьох рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Pill
SELECT * FROM public."Pill"
Pill table:
id: 1   manufacturer_id: 1      name: nurofen   price: 100

-------------------------------------------------------------------
id: 2   manufacturer_id: 1      name: strepsils         price: 120

-------------------------------------------------------------------
id: 3   manufacturer_id: 2      name: phenazepam        price: 40

-------------------------------------------------------------------
id: 4   manufacturer_id: 3      name: analgin   price: 20

-------------------------------------------------------------------
id: 5   manufacturer_id: 2      name: xfroiur   price: 56114

-------------------------------------------------------------------
id: 6   manufacturer_id: 3      name: poerkbox  price: 32149

-------------------------------------------------------------------
id: 7   manufacturer_id: 2      name: wgigvovve         price: 61615

-------------------------------------------------------------------
id: 8   manufacturer_id: 1      name: wlfanywdwhqx      price: 1330
```

Вставка 4 рандомізованих записів до таблиці “Category_pill”

Таблиця до вставки рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3

-------------------------------------
id: 3   pill_id: 2      category_id: 4

-------------------------------------
id: 4   pill_id: 3      category_id: 1

-------------------------------------
id: 9   pill_id: 2      category_id: 1

-------------------------------------
id: 1   pill_id: 2      category_id: 1

-------------------------------------
id: 10  pill_id: 2      category_id: 1

-------------------------------------
```

Запит:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py generate_randomly Category_pill 4
insert into public."Category_pill" select (SELECT (MAX(id)+1) FROM public."Category_pill"), (SELECT id FROM public."Pill" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM
 public."Pill")-1)))), (SELECT id FROM public."Category" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Category")-1))));
insert into public."Category_pill" select (SELECT (MAX(id)+1) FROM public."Category_pill"), (SELECT id FROM public."Pill" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM
 public."Pill")-1)))), (SELECT id FROM public."Category" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Category")-1))));
insert into public."Category_pill" select (SELECT (MAX(id)+1) FROM public."Category_pill"), (SELECT id FROM public."Pill" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM
 public."Pill")-1)))), (SELECT id FROM public."Category" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Category")-1))));
insert into public."Category_pill" select (SELECT (MAX(id)+1) FROM public."Category_pill"), (SELECT id FROM public."Pill" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM
 public."Pill")-1)))), (SELECT id FROM public."Category" LIMIT 1 OFFSET (round(random() *((SELECT COUNT(id) FROM public."Category")-1))));
```

Таблиця після вставки чотирьох рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Category_pill
SELECT * FROM public."Category_pill"
Category_Pill table:
id: 2   pill_id: 1      category_id: 3
------------------------------------
id: 3   pill_id: 2      category_id: 4
------------------------------------
id: 4   pill_id: 3      category_id: 1
------------------------------------
id: 9   pill_id: 2      category_id: 1
------------------------------------
id: 1   pill_id: 2      category_id: 1
------------------------------------
id: 10  pill_id: 2      category_id: 1
------------------------------------
id: 11  pill_id: 6      category_id: 8
------------------------------------
id: 12  pill_id: 4      category_id: 9
------------------------------------
id: 13  pill_id: 4      category_id: 4
------------------------------------
id: 14  pill_id: 2      category_id: 5
------------------------------------
```

Вставка 4 рандомізованих записів до таблиці “Manufacturer”

Таблиця до вставки рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Manufacturer
SELECT * FROM public."Manufacturer"
Manufacturer table:
id: 1   name: reckitt   country: England        email reckitt@gmail.com
-----------------------------------------
id: 2   name: Valenta Pharm     country: Russia         email valentapharm@gmail.com
-----------------------------------------
id: 3   name: Ternopharm        country: Ukraine        email ternopharm@gmail.com
-----------------------------------------
id: 4   name: UVP       country: Ukraine        email uvp@gmail.com
```

Запит:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py generate_randomly Manufacturer 4
insert into public."Manufacturer" select (SELECT MAX(id)+1 FROM public."Manufacturer"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_se
ries(1, FLOOR(RANDOM()*(25-10)+10):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4)::
integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(25-10)+10):: integer)), '');
insert into public."Manufacturer" select (SELECT MAX(id)+1 FROM public."Manufacturer"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_se
ries(1, FLOOR(RANDOM()*(25-10)+10):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4)::
integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(25-10)+10):: integer)), '');
insert into public."Manufacturer" select (SELECT MAX(id)+1 FROM public."Manufacturer"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_se
ries(1, FLOOR(RANDOM()*(25-10)+10):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4)::
integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(25-10)+10):: integer)), '');
insert into public."Manufacturer" select (SELECT MAX(id)+1 FROM public."Manufacturer"), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_se
ries(1, FLOOR(RANDOM()*(25-10)+10):: integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4)::
integer)), ''), array_to_string(ARRAY(SELECT chr((97 + round(random() * 25)) :: integer) FROM generate_series(1, FLOOR(RANDOM()*(25-10)+10):: integer)), '');
```

Таблиця після вставки чотирьох рандомізованих записів:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py print_table Manufacturer
SELECT * FROM public."Manufacturer"
Manufacturer table:
id: 1   name: reckitt   country: England        email reckitt@gmail.com
-----------------------------------------
id: 2   name: Valenta Pharm     country: Russia         email valentapharm@gmail.com
-----------------------------------------
id: 3   name: Ternopharm        country: Ukraine        email ternopharm@gmail.com
-----------------------------------------
id: 4   name: UVP       country: Ukraine        email uvp@gmail.com
-----------------------------------------
id: 5   name: dtnlwqyvmpm       country: upjwt  email ipjidfsstlnfrpqov
-----------------------------------------
id: 6   name: pwhojuatfx        country: etbmrilr       email mwyipfqduekieurachlttc
-----------------------------------------
id: 7   name: jbhutwcfyynbjgqrdqq       country: cbsfurx        email vesbbavhlsboqvbo
-----------------------------------------
id: 8   name: ttrcqtqcqxunpbskgm        country: nyrxtsod       email sqlolkplitedwidpuippyvj
```

Завдання 3

Метод реалізації:

1. Search_records – реалізує пошук за атрибутами введиних користувачем таблиць та виводить у вікно терміналу результат пошуку та час, за який він виконався. За аргументи функція приймає:
    1) table1_name, table2_name, table1_key, table2_key;
    2) table1_name, table2_name, table3_name, table1_key, table2_key, table3_key, table13_key;
    3) table1_name table2_name table3_name table4_name table1_key table2_key table3_key table13_key table4_key table24_key (де table13_key, table24_key – це зовнішні ключі, що зв'язують 1 та 3 таблицю, або 2 та 4).

Пошук за двома атрибутами з двох таблиць (Pill, Manufacturer)
Формування запиту:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py search_records Pill Manufacturer id id
specify the number of attributes you`d like to search by: 2
specify the type of data you want to search for (numeric or string): numeric
specify the name of key by which you`d like to perform search in form: table_number.key_name: one.id
specify the left end of search interval: 0
specify the right end of search interval: 8
specify the type of data you want to search for (numeric or string): string
specify the name of key by which you`d like to perform search in form: table_number.key_name: two.country
specify the string you`d like to search for: Ukraine
select * from public."Pill" as one inner join public."Manufacturer" as two on one."id"=two."id" where 0<one.id and one.id<8 and two.country LIKE 'Ukraine'
```

Запит:

```
select * from public."Pill" as one inner join public."Manufacturer" as two on one."id"=two."id" where 0<one.id and one.id<8 and two.country LIKE 'Ukraine'
```

Результат пошуку:

Пошук за трьома атрибутами з трьох таблиць (Pill, Manufacurer, Category_pill) Формування запиту:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py search_records Pill Manufacturer Category_pill id id id id
specify the number of attributes you`d like to search by: 3
specify the type of data you want to search for (numeric or string): numeric
specify the name of key by which you`d like to perform search in form: table_number.key_name: one.price
specify the left end of search interval: 50
specify the right end of search interval: 450
specify the type of data you want to search for (numeric or string): string
specify the name of key by which you`d like to perform search in form: table_number.key_name: two.country
specify the string you`d like to search for: Ukraine
specify the type of data you want to search for (numeric or string): numeric
specify the name of key by which you`d like to perform search in form: table_number.key_name: three.pill_id
specify the left end of search interval: 0
specify the right end of search interval: 10
```

Запит:

```
select * from public."Pill" as one inner join public."Manufacturer" as two on one."id"=two."id" inner join public."Category_pill" as three on three."id"=one."id"where 50<one.price and one.price<450 and two.country LIKE 'Ukraine' and 0<three.pill_id and three.pill_id<10
```

Результат пошуку:

```
--- 0.009563446044921875 seconds ---
search result:
5
2
ehpizol
138
5
Farmak
Ukraine
farmak@gmail.com
5
6
2
-----------------------------------
6
5
agnesti
400
6
biopharma
Ukraine
biopharma@gmail.com
6
6
1
```

Пошук за чотирьма атрибутами з чотирьох таблиць (Pill, Category, Category_pill, Manufacturer)

Формування запиту:

```
PS C:\Users\anna\PycharmProjects\pythonProject> python main.py search_records Pill Manufacturer Category_pill Category id id id id id id
specify the number of attributes you`d like to search by: 4
specify the type of data you want to search for (numeric or string): numeric
specify the name of key by which you`d like to perform search in form: table_number.key_name: one.price
specify the left end of search interval: 47
specify the right end of search interval: 423
specify the type of data you want to search for (numeric or string): string
specify the name of key by which you`d like to perform search in form: table_number.key_name: two.name
specify the string you`d like to search for: reckitt
specify the type of data you want to search for (numeric or string): numeric
specify the name of key by which you`d like to perform search in form: table_number.key_name: three.pill_id
specify the left end of search interval: 0
specify the right end of search interval: 8
specify the type of data you want to search for (numeric or string): numeric
specify the name of key by which you`d like to perform search in form: table_number.key_name: four.id
specify the left end of search interval: 0
specify the right end of search interval: 9
```

Запит:

```
select * from public."Pill" as one inner join public."Manufacturer" as two on one."id"=two."id" inner join public."Category_pill" as three on three."id"=one."id" inner join public."Category" as four on four."id"=two."id"where 47<one.price and one.pr
ice<423 and two.name LIKE 'reckitt' and 0<three.pill_id and three.pill_id<8 and 0<four.id and four.id<9
```

Результат пошуку:

```
--- 0.004987239837646484 seconds ---
search result:
1
1
nurofen
100
1
reckitt
England
reckitt@gmail.com
1
2
1
1
sedative
cause soothing or a decrease in emotional stress without a hypnotic effect
```

Завдання 4

Програмний код модулю “Model”:

```
import psycopg2 as ps

class Model:
    def __init__(self):
        self.conn = None
        try:
            self.conn = ps.connect(
                dbname="pharmacy",
                user='postgres',
                password="markovka",
                host='127.0.0.1',
                port="5432",
            )
        except(Exception, ps.DatabaseError) as error:
            print("[INFO] Error while working with Postgresql", error)

    def request(self, req: str):
        try:
            cursor = self.conn.cursor()
            print(req)
            cursor.execute(req)
            self.conn.commit()
            return True
        except(Exception, ps.DatabaseError, ps.ProgrammingError) as error:
            print(error)
            self.conn.rollback()
            return False

    def get(self, req: str):
        try:
            cursor = self.conn.cursor()
            print(req)
            cursor.execute(req)
            self.conn.commit()
            return cursor.fetchall()
        except(Exception, ps.DatabaseError, ps.ProgrammingError) as error:
            print(error)
            self.conn.rollback()
            return False

    def get_el(self, req: str):
        try:
            cursor = self.conn.cursor()
            print(req)
            cursor.execute(req)
            self.conn.commit()
            return cursor.fetchone()
        except(Exception, ps.DatabaseError, ps.ProgrammingError) as error:
            print(error)
            self.conn.rollback()
            return False

    def count(self, table_name: str):
        return self.get_el(f"select count(*) from public.\"{table_name}\"")

    def find(self, table_name: str, key_name: str, key_value: int):
        return     self.get_el(f"select     count(*)     from     public.\"{table_name}\"     where
{key_name}={key_value}")
```

```
    def max(self, table_name: str, key_name: str):
        return self.get_el(f"select max({key_name}) from public.\"{table_name}\"")

    def min(self, table_name: str, key_name: str):
        return self.get_el(f"select min({key_name}) from public.\"{table_name}\"")

    def print_category(self) -> None:
        return self.get(f"SELECT * FROM public.\"Category\"")

    def print_category_pill(self) -> None:
        return self.get(f"SELECT * FROM public.\"Category_pill\"")

    def print_pill(self) -> None:
        return self.get(f"SELECT * FROM public.\"Pill\"")

    def print_manufacturer(self) -> None:
        return self.get(f"SELECT * FROM public.\"Manufacturer\"")

    def delete_data(self, table_name: str, key_name: str, key_value) -> None:
        self.request(f"DELETE FROM public.\"{table_name}\" WHERE {key_name}={key_value};")

    def update_data_category(self, key_value: int, name: str, description: str) -> None:
        self.request(f"UPDATE          public.\"Category\"          SET          name=\'{name}\',
description=\'{description}\' "
                     f" WHERE id={key_value};")

    def update_data_category_pill(self, key_value: int, pill_id: int, category_id: int) -> None:
        self.request(f"UPDATE      public.\"Category_pill\"      SET      pill_id=\'{pill_id}\',
category_id=\'{category_id}\' "
                     f" WHERE id={key_value};")

    def update_data_pill(self, key_value: int, manufacturer_id: int, name: str, price: int) ->
None:
        self.request(f"UPDATE    public.\"Pill\"    SET    manufacturer_id=\'{manufacturer_id}\',
name=\'{name}\', price=\'{price}\'"
                     f" WHERE id={key_value};")

    def update_data_manufacturer(self, key_value: int, name: str, country: str, email: str) ->
None:
        self.request(f"UPDATE public.\"Manufacturer\" SET name=\'{name}\', country=\'{country}\',
email=\'{email}\'"
                     f" WHERE id={key_value};")

    def insert_data_category(self, key_value: int, name: str, description: str) -> None:
        self.request(f"insert into public.\"Category\" (id, name, description) "
                     f"VALUES ({key_value}, \'{name}\', \'{description}\');")

    def insert_data_category_pill(self, key_value: int, pill_id: int, category_id: int) -> None:
        self.request(f"insert into public.\"Category_pill\" (id, pill_id, category_id) "
                     f"VALUES ({key_value}, \'{pill_id}\', \'{category_id}\');")

    def insert_data_pill(self, key_value: int, manufacturer_id: int, name: str, price: int) ->
None:
        self.request(f"insert into public.\"Pill\" (id, manufacturer_id, name, price) "
                     f"VALUES ({key_value}, \'{manufacturer_id}\', \'{name}\', \'{price}\')")

    def insert_data_manufacturer(self, key_value: int, name: str, country: str, email: str) ->
None:
        self.request(f"insert into public.\"Manufacturer\" (id, name, country, email) "
                     f"VALUES ({key_value}, \'{name}\', \'{country}\', \'{email}\')")

    def category_data_generator(self, times: int) -> None:
        for i in range(times):
```

```
            self.request("insert into public.\"Category\""
                         "select (SELECT MAX(id)+1 FROM public.\"Category\"), "
                         "array_to_string(ARRAY(SELECT  chr((97  +  round(random()  *  25))  ::
integer) \
                         FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), ''), "
                         "array_to_string(ARRAY(SELECT  chr((97  +  round(random()  *  25))  ::
integer) \
                         FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), ''); ")

    def category_pill_data_generator(self, times: int) -> None:
        for i in range(times):
            self.request("insert into public.\"Category_pill\" "
                         "select (SELECT (MAX(id)+1) FROM public.\"Category_pill\"), "
                         "(SELECT id FROM public.\"Pill\" LIMIT 1 OFFSET "
                         "(round(random() *((SELECT COUNT(id) FROM public.\"Pill\")-1)))), "
                         "(SELECT id FROM public.\"Category\" LIMIT 1 OFFSET "
                         "(round(random() *((SELECT COUNT(id) FROM public.\"Category\")-1))));")

    def pill_data_generator(self, times: int) -> None:
        for i in range(times):
            self.request("insert   into   public.\"Pill\"   select   (SELECT   MAX(id)+1   FROM
public.\"Pill\"), "
                         "(SELECT id FROM public.\"Manufacturer\" LIMIT 1 OFFSET "
                         "(round(random()  *((SELECT  COUNT(id)  FROM  public.\"Manufacturer\")-
1))), "
                         "array_to_string(ARRAY(SELECT  chr((97  +  round(random()  *  25))  ::
integer) "
                         "FROM generate_series(1, FLOOR(RANDOM()*(15-5)+5):: integer)), ''), "
                         "FLOOR(RANDOM()*(100000-1)+1);")

    def manufacturer_data_generator(self, times: int) -> None:
        for i in range(times):
            self.request("insert  into  public.\"Manufacturer\"  select  (SELECT  MAX(id)+1  FROM
public.\"Manufacturer\"), "
                         "array_to_string(ARRAY(SELECT  chr((97  +  round(random()  *  25))  ::
integer) "
                         "FROM generate_series(1, FLOOR(RANDOM()*(25-10)+10):: integer)), ''), "
                         "array_to_string(ARRAY(SELECT  chr((97  +  round(random()  *  25))  ::
integer) "
                         "FROM generate_series(1, FLOOR(RANDOM()*(10-4)+4):: integer)), ''), "
                         "array_to_string(ARRAY(SELECT  chr((97  +  round(random()  *  25))  ::
integer) "
                         "FROM generate_series(1, FLOOR(RANDOM()*(25-10)+10):: integer)), '');")

    def search_data_two_tables(self, table1_name: str, table2_name: str, table1_key, table2_key,
                               search: str):
        return  self.get(f"select  *  from  public.\"{table1_name}\"  as  one  inner  join
public.\"{table2_name}\" as two "
                         f"on one.\"{table1_key}\"=two.\"{table2_key}\" "
                         f"where {search}")

    def search_data_three_tables(self, table1_name: str, table2_name: str, table3_name: str,
                                 table1_key, table2_key, table3_key, table13_key,
                                 search: str):
        return  self.get(f"select  *  from  public.\"{table1_name}\"  as  one  inner  join
public.\"{table2_name}\" as two "
                         f"on      one.\"{table1_key}\"=two.\"{table2_key}\"      inner      join
public.\"{table3_name}\" as three "
                         f"on three.\"{table3_key}\"=one.\"{table13_key}\""
                         f"where {search}")

    def  search_data_all_tables(self,  table1_name:  str,  table2_name:  str,  table3_name:  str,
table4_name: str,
```

```
                            table1_key, table2_key, table3_key, table13_key,
                            table4_key, table24_key,
                            search: str):
        return self.get(f"select * from public.\"{table1_name}\" as one inner join
public.\"{table2_name}\" as two "
                        f"on one.\"{table1_key}\"=two.\"{table2_key}\" inner join
public.\"{table3_name}\" as three "
                        f"on three.\"{table3_key}\"=one.\"{table13_key}\" inner join
public.\"{table4_name}\" as four "
                        f"on four.\"{table4_key}\"=two.\"{table24_key}\""
                        f"where {search}")
```

Опис функцій модуля:

Модуль “Model.py” слугує точкую доступу до бази даних. Для реалізації запитів користувача до бази даних використовується бібліотека psycopg2.

У модулі використані такі функції:

1. Request, get, get_el – здійснюють запити до бази даних. При правильному запиті request повертає True, get повертає усі дані що було взято з запитів SELECT (масив кортежів з записами таблиць), get_el повертає тільки перший запис. У разі помилки вони повертають False;
2. Max, min – повертають максимальне і мінімальне значення зазначеного ключа у таблиці;
3. Count – повертає кількість усіх записів у таблиці;
4. Find – повертає кількість записів таблиці, що відповідають заданій користувачем умові
5. Print_category – отримання з бази даних та виведення у консоль користувача таблиці “Category”;
6. Print_category_pill – отримання з бази даних та виведення у консоль користувача таблиці “Category_pill”;
7. Print_pill – отримання з бази даних та виведення у консоль користувача таблиці “Pill”;
8. Print_manufacturer – отримання з бази даних та виведення у консоль користувача таблиці “Manufacturer”;
9. Delete_data – реалізує видалення запису з обраної користувачем таблиці;
10. Update_data_(назва таблиці) – реалізує запит за зміну даних у обраній користувачем таблиці;
11. Insert_data_(назва таблиці) – реалізує запит на вставку запису до обраної користувачем таблиці;
12. (назва таблиці)_data_generator – реалізує запит на вставку рандомізованих записів до обраної користувачем таблиці;
13. Search_data_(кількість таблиць)_tables – реалізую пошук даних у вибраній користувачем кількості таблиць;