

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря
Сікорського ”**

**Факультет прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

Розрахунково-графічна робота

**з дисципліни
“Бази даних та засоби управління”**

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

Група: KB-11

Виконав: Кухта Данило

Київ – 2023

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

Загальне завдання роботи полягає у наступному:

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).

URL репозиторію <https://github.com/joe1i/DB-labs.git>

Модель «сутність-зв'язок» предметної галузі, для проектування бази даних «Платформа для продажу та покупки мистецьких творів»:

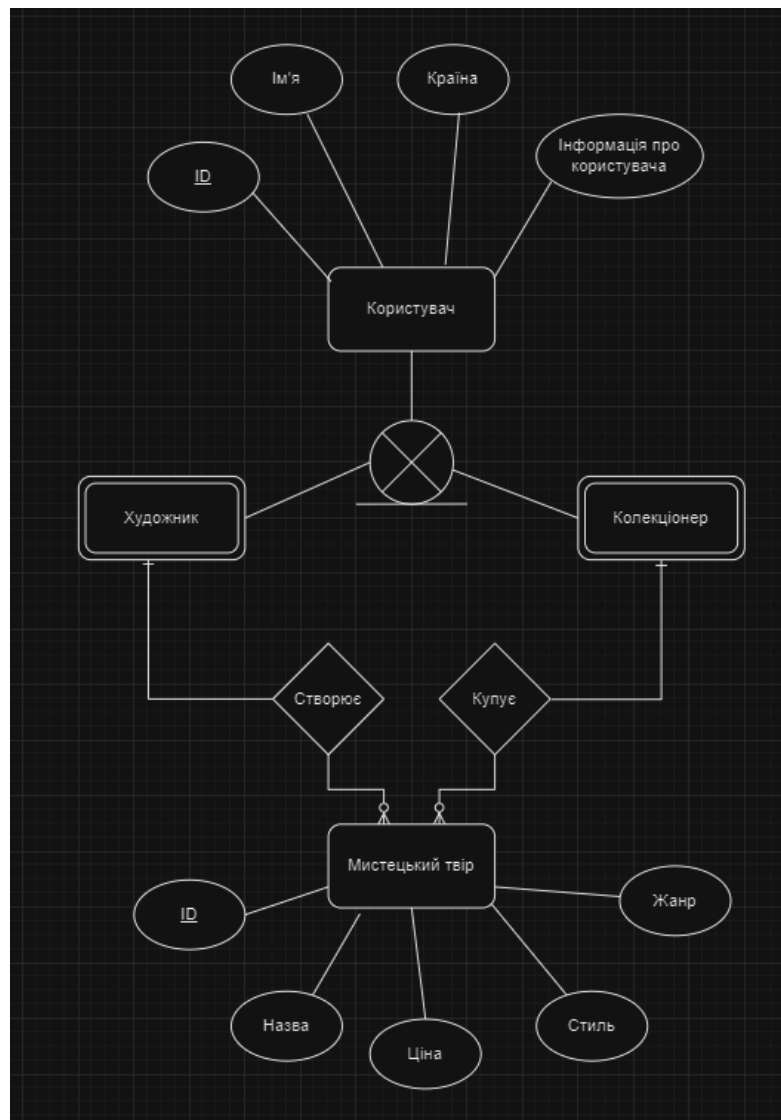


Рис. 1. ER-діаграма побудована за нотацією «Пташина лапка»

Сутності з описом призначення

Користувач (User): Представляє користувачів системи. Ділиться на два підтипи: Художник та Колекціонер. Може включати атрибути, такі як ID, ім'я, країна, особиста інформація про користувача.

- **Художник (Artist):** Представляє користувачів системи, які можуть створювати мистецькі твори.
- **Колекціонер (Collector):** Представляє користувачів системи, які можуть купувати мистецькі твори.

Мистецький твір (Work of art): Представляє продукт, який може бути куплений і створений. Його можуть купити Колекціонери, а розмістити у

платформі можуть Художники. Має атрибути, які включають ID, назву, ціну, жанр, стиль.

Модель “сутність-зв’язок” у схемі бази даних PostgreSQL:

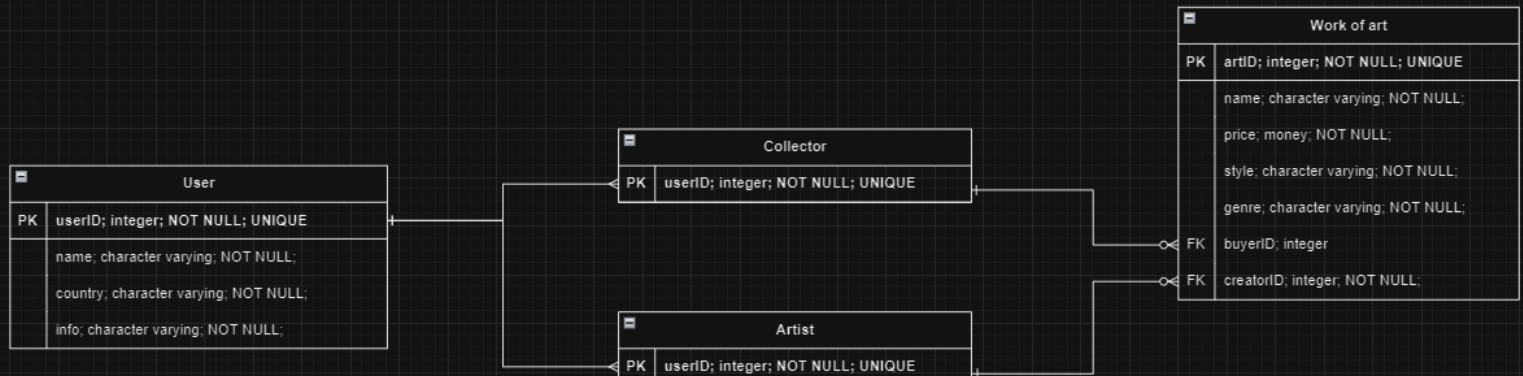


Рис. 2. Схеми бази даних у графічному вигляді

- Середовище для відлагодження SQL-запитів до бази даних – PgAdmin 4;
- Мова програмування – Python 3.10;
- Середовище розробки програмного забезпечення – PyCharm Community Edition;
- Бібліотека взаємодії з PostgreSQL – Psycopg2;
- Бібліотека для вимірювання часу – time;
- Бібліотека для створення таблиць в консолі – PrettyTable.

Схема меню користувача з описом функціональності кожного пункту:

```
Menu:  
1. User  
2. Work of art  
3. Search  
4. Quit  
Enter your choice:
```

Головне меню за допомогою якого можна обрати з якими сутностями взаємодіяти, здійснити пошук та вийти.

Menu/1. Users

```
Enter your choice: 1  
  
Users:  
1. Artist  
2. Collector  
3. Show all users  
4. back  
Enter your choice: |
```

Меню в якому користувач може вибрати з яким саме юзером взаємодіяти (Художник або Колекціонер) або вивести всіх юзерів

Menu/1. Users/1. Artist

```
Enter your choice: 1  
  
Users.Artist:  
1. show  
2. add  
3. edit  
4. remove  
5. random fill  
6. show artist's arts  
7. back  
Enter your choice: |
```

Меню вибору операцій із Художником.

Menu/1. Users/1. Artist/1. show

```
Users.Artist.show
1. show current artist
2. show all artists
3. back
Enter your choice: |
```

Операція виводу всіх або конкретного Художника. Для виведення конкретного Художника користувач спочатку вводить id Художника, і якщо він є в таблиці “Artist”, то програма виводить інформацію про нього. Операція виведення всіх Художників виводить інформації про всіх Художників в базі.

Menu/1. Users/1. Artist/2. add

Операція додавання Художника. Користувач вводить дані Художника(ім’я, країна, інформація) і він додається в таблицю “User”, а його id в таблицю “Artist”.

Menu/1. Users/1. Artist/3. edit

Операція редагування даних про Художника. Користувач вводить id Художника, якого хоче редагувати. Потім, якщо цей Художник є в таблиці “Artist”, користувач вводить нові дані про нього(ім’я, країна, інформація), і він оновлюється в таблиці “User”.

Menu/1. Users/1. Artist/4. remove

Операція видалення Художника. Користувач вводить id Художника, якого хоче видалити. Потім, якщо цей Художник є в таблиці “Artist”, він видаляється з неї та з “User”. Крім того, видаляються ще й всі його Мистецькі твори.

Menu/1. Users/1. Artist/5. random fill

Операція заповнення таблиці “User” псевдовипадковими даними про Художника. Користувач вводить кількість Художників, які мають згенеруватись. Потім вони генеруються та додаються в таблицю “User” а їхні id в таблицю “Artist”.

Menu/1. Users/1. Artist/6. show artist`s arts

Операція виводу всіх робіт Художника. Користувач вводить id Художника і далі, якщо він існує та є Мистецькі твори, які він створив, виводиться список всіх цих Мистецьких творів.

Menu/1. Users/2. Collector

```
Users.Collector:
1. show
2. add
3. edit
4. remove
5. random fill
6. show collector`s collection
7. back
Enter your choice: |
```

Меню вибору операцій із Колекціонером. Всі операції є аналогічні операціям з Художником, окрім **4. remove** та **6. show collectors`s collection**, тому опишу тільки їх.

Menu/1. Users/2. Collector/4. remove

Операція видалення Колекціонера. Користувач вводить id Колекціонера, якого хоче видалити. Потім, якщо цей Колекціонер є в таблиці “Collector”, він видаляється з неї та з “User”. Крім того, у всіх Мистецьких творах, які є придбані цим Колекціонером, атрибут “buyerID” встановлюється в null.

Menu/1. Users/2. Collector/6. show collectors`s collection

Операція виводу всіх Мистецьких творів, що є куплені Колекціонером. Користувач вводить id Колекціонера і далі, якщо він існує та є Мистецькі твори, які він купив, виводиться список всіх цих Мистецьких творів.

Menu/1. Users/3. Show all users

Операція виведення всіх юзерів (Художників та Колекціонерів).

Menu/2. Work of art

```
Work of art:
1. show
2. add
3. edit
4. remove
5. random fill
6. buy
7. back
Enter your choice:
```

Меню вибору операцій із Мистецьким твором.

Menu/2. Work of art/1. show

```
Users.Artist.show
1. show current artist
2. show all artists
3. back
Enter your choice: |
```

Операція виводу всіх або конкретного Мистецького твору. Для виведення конкретного Мистецького твору користувач спочатку вводить id Мистецького твору, і якщо він є в таблиці “Work of art”, то програма виводить інформацію про нього. Операція виведення всіх Мистецьких творів виводить інформації про всіх Мистецьких творів в базі.

Menu/2. Work of art/2. add

Операція додавання Мистецького твору. Користувач вводить дані Мистецького твору(назва, ціна, стиль, жанр, id автора) і він додається в таблицю. При додаванні Мистецького твору, покупець не вказується.

Menu/2. Work of art/3. edit

Операція редагування даних про Мистецький твір. Користувач вводить id Мистецького твору, якого хоче редагувати. Потім, якщо Мистецький твір з цим id є в таблиці “Work of art”, користувач вводить нові дані (назва, ціна, стиль, жанр, id покупця, id автора), і він оновлюється в таблиці.

Menu/2. Work of art/4. remove

Операція видалення Мистецького твору. Користувач вводить id Мистецького твору, якого хоче видалити. Потім, якщо Мистецький твір з цим id є в таблиці “Work of art”, він видаляється з неї.

Menu/2. Work of art/5. random fill

Операція заповнення таблиці “Work of art” псевдовипадковими даними про Мистецький твір. Користувач вводить кількість Мистецьких творів, які мають згенеруватись. Потім вони генеруються та додаються в таблицю.

Menu/2. Work of art/6. buy

Операція купівлі Мистецького твору. Спочатку користувач вводить id Мистецького твору, який буде купуватись. Далі id Колекціонера, і, якщо в базі є Колекціонер та Мистецький твір з відповідними id і, якщо Мистецький твір не є куплений кимось іншим, то в атрибут “buyerID” записується id Колекціонера.

Menu/3. Search

```
Search:
1. search by style, genre, price
2. search by country, availability, price
3. search for each country's most expensive work of art
4. back
Enter your choice: |
```

Меню вибору операцій пошуку.

Menu/3. Search/1. search by style, genre, price

Операція пошуку Мистецьких творів за стилем, жанром, ціною. Користувач вводить стиль, жанр, мінімальну та максимальну ціну. Потім виводиться список всіх Мистецьких творів, які відповідають цим параметрам.

Menu/3. Search/2. search by country, availability, price

Операція пошуку Мистецьких творів за країною, доступністю, ціною. Користувач вводить країну, доступність(true або false), мінімальну та максимальну ціну. Потім виводиться список всіх Мистецьких творів:

- які створені Художником, у якого атрибут “країна” дорівнює введеному значенню;
- у яких “buyerID” не пустий, якщо доступність обрана False, або “buyerID” пустий, якщо доступність обрана True;
- ціна яких знаходиться в межах мін. та макс. введених цін.

Menu/3. Search/3. search for each country's most expensive work of art

Операція виводить список найдорожчих Мистецьких творів в кожній країні.

Menu/4. Quit

Завершення роботи.

Завдання №1

Забезпечити можливість введення/редагування/видалення даних у таблицях бази даних.

Видалення даних

- Видалення Художника

Таблиця “User” до:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
1	2432254	Lucas Jackson	Peru	DTN
2	2432255	Matthew Rodriguez	Malaysia	CKI
3	2432256	Noah Young	India	XHH
4	2432257	Sofia Davis	Thailand	EPC

Таблиця “Artist” до:

	userID [PK] integer
1	2432254
2	2432255

Таблиця “Work of art” до:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14073	Bold Horizon Discovering	66890	Contemporary Art	Miniature	2432256	2432255
2	14074	Tranquil Silhouette Resonating	34438	Abstract	Animal	2432257	2432254
3	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
4	14076	Mesmerizing Silhouette Reflecting	62768	Art Deco	Flower	[null]	2432254

```
Users:
1. Artist
2. Collector
3. Show all users
4. back
Enter your choice: 1

Users.Artist:
1. show
2. add
3. edit
4. remove
5. random fill
6. show artist's arts
7. back
Enter your choice: 4
Enter user ID: 2432254
Artist deleted successfully!
```

Таблиця “User” після:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
1	2432255	Matthew Rodriguez	Malaysia	CKI
2	2432256	Noah Young	India	XHH
3	2432257	Sofia Davis	Thailand	EPC

Таблиця “Artist” після:

	userID [PK] integer
1	2432255

Таблиця “Work of art” після:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14073	Bold Horizon Discovering	66890	Contemporary Art	Miniature	2432256	2432255
2	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255

Лістинг:

controller:

```
def remove_artist(self):
    user_id = self.view.get_user_id()
    if not self.model.get_artist(user_id):
        self.view.show_message("There is no artist with this id!")
        return
    self.model.remove_artist(user_id)
    self.view.show_message("Artist deleted successfully!")
```

view:

```
def get_user_id(self):
    return self.input_digit("Enter user ID: ", True, sys.maxsize)

def input_digit(self, message, notnull, length):
    while True:
        digits = input(message)
        if len(digits) > length:
            print("Input data is too long! Try again.")
            continue
```

```
if digits.isdigit():
    return int(digits)
elif not digits and notnull is False:
    return None
print("Wrong input data! Try again.")
```

model:

```
def remove_artist(self, user_id):
    c = self.conn.cursor()

    c.execute('DELETE FROM "Work of art" WHERE "creatorID"=%s',
              (user_id,))
    c.execute('DELETE FROM "Artist" WHERE "userID"=%s', (user_id,))

    c.execute('DELETE FROM "User" WHERE "userID"=%s', (user_id,))
    self.conn.commit()

def get_artist(self, user_id):
    c = self.conn.cursor()
    c.execute('''SELECT U."userID", U."name", U."country", U."info"
                FROM "User" AS U
                INNER JOIN "Artist" ON U."userID" = "Artist"."userID"
                WHERE "Artist"."userID" = %s;''', (user_id,))
    artist = c.fetchone()
    return artist
```

- Видалення Колекціонера

Таблиця “User” до:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
1	2432255	Matthew Rodriguez	Malaysia	CKI
2	2432256	Noah Young	India	XHH
3	2432257	Sofia Davis	Thailand	EPC
4	2432258	Olivia Thomas	Egypt	UIE

Таблиця “Collector” до:

	userID [PK] integer
1	2432256
2	2432257

Таблиця “Work of art” до:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	2432257	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258

```

Users.Collector:
1. show
2. add
3. edit
4. remove
5. random fill
6. show collector's collection
7. back
Enter your choice: 4
Enter user ID: 2432257
Collector deleted successfully!

```

Таблиця “User” після:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
1	2432255	Matthew Rodriguez	Malaysia	CKI
2	2432256	Noah Young	India	XHH
3	2432258	Olivia Thomas	Egypt	UIE

Таблиця “Collector” після:

	userID [PK] integer
1	2432256

Таблиця “Work of art” після:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	[null]	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258

Лістинг:

controller:

```
def remove_collector(self):
    user_id = self.view.get_user_id()
    if not self.model.get_collector(user_id):
        self.view.show_message("There is no collector with this id!")
        return
    self.model.remove_collector(user_id)
    self.view.show_message("Collector deleted successfully!")
```

view:

```
def get_user_id(self):
    return self.input_digit("Enter user ID: ", True, sys.maxsize)

def input_digit(self, message, notnull, length):
    while True:
        digits = input(message)
```

```

    if len(digits) > length:
        print("Input data is too long! Try again.")
        continue
    if digits.isdigit():
        return int(digits)
    elif not digits and notnull is False:
        return None
    print("Wrong input data! Try again.")

```

model:

```

def remove_collector(self, user_id):
    c = self.conn.cursor()

    c.execute('UPDATE "Work of art" SET "buyerID"=NULL WHERE
"buyerID"=%s', (user_id,))
    c.execute('DELETE FROM "Collector" WHERE "userID"=%s', (user_id,))

    c.execute('DELETE FROM "User" WHERE "userID"=%s', (user_id,))
    self.conn.commit()

def get_collector(self, user_id):
    c = self.conn.cursor()
    c.execute('''SELECT U."userID", U."name", U."country", U."info"
                FROM "User" AS U
                INNER JOIN "Collector" ON U."userID" =
                "Collector"."userID"
                WHERE "Collector"."userID" = %s;''', (user_id,))
    collector = c.fetchone()
    return collector

```


Внесення даних

- Додавання Мистецького твору

Таблиця “User”:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
1	2432255	Matthew Rodriguez	Malaysia	CKI
2	2432256	Noah Young	India	XHH
3	2432258	Olivia Thomas	Egypt	UIE

Таблиця “Artist”:

	userID [PK] integer
1	2432255
2	2432258

Таблиця “Work of art” до:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	[null]	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258

```
Work of art:
1. show
2. add
3. edit
4. remove
5. random fill
6. buy
7. back
Enter your choice: 2
Enter art name: Fantastic Journey Discovering
Enter art price: 4000
Enter art style: Realism
Enter art genre: Landscape
Enter creator ID: 2432255
Art added successfully!
```

Таблиця “Work of art” після:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	[null]	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258
5	14080	Fantastic Journey Discovering	4000	Realism	Landscape	[null]	2432255

Помилка, коли в поле “creatorID” вводимо id, якого немає в таблиці “Artist”:

```
Work of art:
1. show
2. add
3. edit
4. remove
5. random fill
6. buy
7. back
Enter your choice: 2
Enter art name: Colorful Horizon Elevating
Enter art price: 3000
Enter art style: Conceptual
Enter art genre: Landscape
Enter creator ID: 2432256
There is no artist with this id!
```

Лістинг:

controller:

```
def add_art(self):
    name, price, style, genre = self.view.get_art_input()
    creatorID = self.view.get_creator_id()
    if not self.model.get_artist(creatorID):
        self.view.show_message("There is no artist with this id!")
        return
    self.model.add_art(name, price, style, genre, creatorID)
    self.view.show_message("Art added successfully!")
```

view:

```
def get_art_input(self):
    name = self.input_arbitrary("Enter art name: ", True, 50)
    price = self.input_digit("Enter art price: ", True, sys.maxsize)
    style = self.input_text("Enter art style: ", True, 50)
    genre = self.input_text("Enter art genre: ", True, 50)
    return name, price, style, genre

def get_creator_id(self):
    return self.input_digit("Enter creator ID: ", True, sys.maxsize)
```

```

def input_text(self, message, notnull, length):
    while True:
        text = input(message)
        if len(text) > length:
            print("Input data is too long! Try again.")
            continue
        if self.is_alpha_with_space(text) or (not text and notnull is False):
            return text
        print("Wrong input data! Try again.")

def input_digit(self, message, notnull, length):
    while True:
        digits = input(message)
        if len(digits) > length:
            print("Input data is too long! Try again.")
            continue
        if digits.isdigit():
            return int(digits)
        elif not digits and notnull is False:
            return None
        print("Wrong input data! Try again.")

def input_arbitrary(self, message, notnull, length):
    while True:
        text = input(message)
        if len(text) > length:
            print("Input data is too long! Try again.")
            continue
        if not text and notnull is True:
            print("Wrong input data! Try again.")
            continue
        return text

def is_alpha_with_space(self, text):
    return all(char.isalpha() or char.isspace() for char in text)

```

model:

```

def add_art(self, name, price, style, genre, creatorID):
    c = self.conn.cursor()
    c.execute('INSERT INTO "Work of art" ("name", price, "style",
"genre", "buyerID", "creatorID") VALUES (%s, %s, %s, %s, null, %s)',
(name, price, style, genre, creatorID))
    self.conn.commit()

```

- Купівля Мистецького твору

Таблиця “User”:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
1	2432255	Matthew Rodriguez	Malaysia	CKI
2	2432256	Noah Young	India	XHH
3	2432258	Olivia Thomas	Egypt	UIE

Таблиця “Collector”:

	userID [PK] integer
1	2432256

Таблиця “Work of art” до:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	[null]	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258
5	14080	Fantastic Journey Discovering	4000	Realism	Landscape	[null]	2432255

```

Work of art:
1. show
2. add
3. edit
4. remove
5. random fill
6. buy
7. back
Enter your choice: 6
Enter art ID: 14080
Enter buyer ID: 2432256
Art was bought successfully!

```

Таблиця “Work of art” після:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	[null]	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258
5	14080	Fantastic Journey Discovering	4000	Realism	Landscape	2432256	2432255

Помилка, коли в поле “artID” вводимо id Мистецького твору, який вже куплений:

```
WORK ON ART:
1. show
2. add
3. edit
4. remove
5. random fill
6. buy
7. back
Enter your choice: 6
Enter art ID: 14080
This art is already bought!
```

Лістинг:

controller:

```
def buy_art(self):
    art_id = self.view.get_art_id()
    if not self.model.get_art(art_id):
        self.view.show_message("There is no art with this id!")
        return

    if self.model.get_arts_buyer(art_id):
        self.view.show_message("This art is already bought!")
        return

    buyer_id = self.view.get_buyer_id()
    if self.model.get_collector(buyer_id):
        self.model.buy_art(buyer_id, art_id)
        self.view.show_message("Art was bought successfully!")
    else:
        self.view.show_message("There is no collector with this id!")
```

view:

```
def get_art_id(self):
    return self.input_digit("Enter art ID: ", True, sys.maxsize)

def get_buyer_id(self):
    return self.input_digit("Enter buyer ID: ", False, sys.maxsize)
```

model:

```
def get_arts_buyer(self, art_id):
    c = self.conn.cursor()
    c.execute('SELECT * FROM "Work of art" WHERE "artID"=%s',
    (art_id,))
    buyer_id = c.fetchone()[5]
    return buyer_id

def buy_art(self, buyer_id, art_id):
    c = self.conn.cursor()
    c.execute('UPDATE "Work of art" SET "buyerID"=%s WHERE "artID"=%s',
    (buyer_id, art_id))
    self.conn.commit()
```

Редагування даних

Таблиця “Work of art” до:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	[null]	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258
5	14080	Fantastic Journey Discovering	4000	Realism	Landscape	2432256	2432255

```
7. Back
Enter your choice: 3
Enter art ID: 14080
Enter art name: Cinematic Reverie Capturing aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Input data is too long! Try again.
Enter art name: Cinematic Reverie Capturing 2
Enter art price: 5000
Enter art style: Contemporary Art
Enter art genre: Miniature
Enter creator ID:
Wrong input data! Try again.
Enter creator ID: 2432258
Enter buyer ID:
Art edited successfully!
```

Таблиця “Work of art” після:

	artID [PK] integer	name character varying (50)	price integer	style character varying (50)	genre character varying (50)	buyerID integer	creatorID integer
1	14075	Whimsical Mirage Capturing	72713	Cubism	Landscape	2432256	2432255
2	14077	Otherworldly Essence Evoking	23057	Academic Art	Mythological	2432256	2432255
3	14078	Radiant Dream Embracing	38388	Fine Art	Cityscapes	[null]	2432258
4	14079	Radiant Harmony Embarking	23378	Minimalism	Cityscapes	[null]	2432258
5	14080	Cinematic Reverie Capturing 2	5000	Contemporary Art	Miniature	[null]	2432258

Лістинг:

controller:

```
def edit_art(self):
    art_id = self.view.get_art_id()
    if not self.model.get_art(art_id):
        self.view.show_message("There is no art with this id!")
        return
    name, price, style, genre = self.view.get_art_input()
    creatorID = self.view.get_creator_id()
    if not self.model.get_artist(creatorID):
        self.view.show_message("There is no artist with this id!")
        return
    buyerID = self.view.get_buyer_id()
    if buyerID and not self.model.get_collector(creatorID):
        self.view.show_message("There is no collector with this id!")
        return
    self.model.edit_art(art_id, name, price, style, genre, buyerID,
creatorID)
    self.view.show_message("Art edited successfully!")
```

model:

```
def edit_art(self, art_id, name, price, style, genre, buyerID, creatorID):
    c = self.conn.cursor()

    c.execute('UPDATE "Work of art" SET "name"=%s, price=%s, "style"=%s,
"genre"=%s, "creatorID"=%s WHERE "artID"=%s', (name, price, style, genre,
creatorID, art_id))

    if not buyerID:
```

```

        c.execute('UPDATE "Work of art" SET "buyerID"=null WHERE
"artID"=%s', (art_id,))
    else:
        c.execute('UPDATE "Work of art" SET "buyerID"=%s WHERE "artID"=%s',
(buyerID, art_id))

    self.conn.commit()

    c.execute('UPDATE "Work of art" SET "buyerID"=%s WHERE "artID"=%s',
(buyer_id, art_id))
    self.conn.commit()

```

Завдання №2

Забезпечити можливість автоматичної генерації великої кількості даних у таблицях.

Для генерації даних було створено додатковий файл “data_fill.py”, який містить зразки даних, такі як імена, прізвища, країни, стилі, жанри, прикметники, іменники та дієслова. При запуску програми викликається функція, яка додає у базу нові таблиці з цими даними, якщо вони ще не створені. Далі запит, що генерує дані, використовує слова з цих таблиць. Завдяки цьому згенеровані дані є досить реалістичними.

Лістинг:

model:

```

def __init__(self):
    self.conn = psycopg2.connect(
        dbname='Arts Platform',
        user='postgres',
        password='qwerty',
        host='localhost',
        port=5432
    )
    self.data_fill = Data()
    self.create_random_data_table("styles", self.data_fill.styles)
    self.create_random_data_table("genres", self.data_fill.genres)
    self.create_random_data_table("countries", self.data_fill.countries)
    self.create_random_data_table("names", self.data_fill.names)

```



```

self.create_random_data_table("surnames", self.data_fill.surnames)
self.create_random_data_table("adjectives", self.data_fill.adjectives)
self.create_random_data_table("nouns", self.data_fill.nouns)
self.create_random_data_table("verbs", self.data_fill.verbs)

def create_random_data_table(self, name, data):
    c = self.conn.cursor()
    c.execute('''
        SELECT EXISTS (
            SELECT 1
            FROM information_schema.tables
            WHERE table_name = %s
        );
        ''', (name,))
    table_exists = c.fetchone()

    if not table_exists[0]:

        c.execute(f'''
            CREATE TABLE {name} (
                id INTEGER PRIMARY KEY,
                name TEXT
            );
            ''')

        for i, value in enumerate(data):
            c.execute(f'INSERT INTO {name} (id, name) VALUES (%s, %s)', (i,
value,))

        self.conn.commit()
        print(f"Таблицю {name} створено та заповнено.")
    else:
        print(f"Таблиця {name} вже існує.")

```

- Генерація Художників

```
Users.Artist:
1. show
2. add
3. edit
4. remove
5. random fill
6. show artist's arts
7. back
Enter your choice: 5
Enter count: 25000
25000 artists added successfully!
```

Таблиця “Artist”:

	userID [PK] integer
1	2432266
2	2432267
3	2432268
4	2432269
5	2432270
6	2432271
7	2432272
8	2432273
9	2432274
10	2432275

Таблиця “User”:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
1	2432266	Madison Hall	Cambodia	FCS
2	2432267	Amelia Scott	Saudi Arabia	MSB
3	2432268	Jack Wilson	Mexico	FOE
4	2432269	Lily Baker	Australia	CYE
5	2432270	Mia Garcia	Bangladesh	FKH
6	2432271	Oliver Brown	Madagascar	MQS
7	2432272	Olivia Rodriguez	Vietnam	SIO
8	2432273	Jackson Parker	Venezuela	GME
9	2432274	Lily Jones	Poland	TZH
10	2432275	Jackson Hall	China	WNA

Лістинг:

controller:

```
def rand_fill_artist(self):
    count = self.view.get_count()
    self.model.rand_fill_artist(count)
    self.view.show_message(f"{count} artists added successfully!")
```

view:

```
def get_count(self):
    return self.input_digit("Enter count: ", True, sys.maxsize)
```

model:

```
def rand_fill_artist(self, count):
    c = self.conn.cursor()

    c.execute(f'''
DO $$
DECLARE
    i INT := 1;
BEGIN
    WHILE i <= {count} LOOP

        WITH inserted_user AS (
            INSERT INTO "User" ("name", country, "info")
            VALUES (
                (SELECT "name" FROM names ORDER BY random() LIMIT 1)
                || ' ' ||
                (SELECT "name" FROM surnames ORDER BY random() LIMIT 1)
                ,
                (SELECT "name" FROM countries ORDER BY random() LIMIT 1)
                ,
                chr(trunc(65 + random()*26)::int)||chr(trunc(65 +
                random()*26)::int)||chr(trunc(65 + random()*26)::int)
            )
            RETURNING "userID"
        )

        INSERT INTO "Artist" ("userID") VALUES ((SELECT "userID" FROM inserted_user));

        i := i + 1;
    END LOOP;
END $$;
''')

    self.conn.commit()
```

- Генерація Колекціонерів

```

Users.Collector:
1. show
2. add
3. edit
4. remove
5. random fill
6. show collector's collection
7. back
Enter your choice: 5
Enter count: 25000
25000 collectors added successfully!

```

Таблиця “Collector”:

	userID [PK] integer
80	2482180
81	2482185
82	2482184
83	2482183
84	2482182
85	2482181
86	2482180
87	2482179
88	2482178
89	2482177
90	2482176
91	2482175
92	2482174

Таблиця “User”:

	userID [PK] integer	name character varying (25)	country character varying (25)	info character varying (50)
80	2482180	Grace Allen	Solomon	EDV
81	2482185	Eleanor Thompson	United Kingdom	WHQ
82	2482184	Olivia White	Algeria	AEV
83	2482183	Ava Stewart	Canada	HUN
84	2482182	Madelyn Rodriguez	South Africa	JUH
85	2482181	Emily Perez	Nigeria	UTC
86	2482180	Victoria Mitchell	Philippines	HXU
87	2482179	Lucas Garcia	India	WGL
88	2482178	Ella Jackson	Uzbekistan	ZXH
89	2482177	Jackson Collins	Poland	PUL

Лістинг:

controller:

```

def rand_fill_collector(self):
    count = self.view.get_count()
    self.model.rand_fill_collector(count)
    self.view.show_message(f"{count} collectors added successfully!")

```

model:

```

def rand_fill_collector(self, count):
    c = self.conn.cursor()

    c.execute(f'''
DO $$
DECLARE
    i INT := 1;
BEGIN
    WHILE i <= {count} LOOP

```

```

WITH inserted_user AS (
    INSERT INTO "User" ("name", country, "info")
    VALUES (
        (SELECT "name" FROM names ORDER BY random() LIMIT 1)
        || ' ' ||
        (SELECT "name" FROM surnames ORDER BY random() LIMIT 1)
        ,
        (SELECT "name" FROM countries ORDER BY random() LIMIT 1)
        ,
        chr(trunc(65 + random()*26)::int)||chr(trunc(65 +
        random()*26)::int)||chr(trunc(65 + random()*26)::int)
    )
    RETURNING "userID"
)

INSERT INTO "Collector" ("userID") VALUES ((SELECT "userID" FROM inserted_user));

    i := i + 1;
END LOOP;
END $$;
'''

self.conn.commit()

```

● Генерація Мистецьких творів

```

Work of art:
1. show
2. add
3. edit
4. remove
5. random fill
6. buy
7. back
Enter your choice: 5
Enter count: 50000
50000 works of art added successfully!

```

Таблиця “Work of art”:

model:

```
def rand_fill_arts(self, count):
    c = self.conn.cursor()

    c.execute(f'''
DO $$
DECLARE
    i INT := 1;
BEGIN
    WHILE i <= {count} LOOP

        INSERT INTO "Work of art" ("name", price, "style", "genre", "buyerID",
"creatorID")
        VALUES (
            (SELECT "name" FROM adjectives ORDER BY random() LIMIT 1)
            || ' ' ||
            (SELECT "name" FROM nouns ORDER BY random() LIMIT 1)
            || ' ' ||
            (SELECT "name" FROM verbs ORDER BY random() LIMIT 1)
            ,
            trunc(random()*100000)::int
            ,
            (SELECT "name" FROM styles ORDER BY random() LIMIT 1)
            ,
            (SELECT "name" FROM genres ORDER BY random() LIMIT 1)
            ,
            CASE WHEN random() < 0.5
                THEN (SELECT "userID" FROM "Collector" ORDER BY random() LIMIT 1)
                ELSE null
            END,
            (SELECT "userID" FROM "Artist" ORDER BY random() LIMIT 1)
        );

        i := i + 1;
    END LOOP;
END $$;
''')

    self.conn.commit()
```

Завдання №3

Забезпечити реалізацію пошуку за декількома атрибутами.

- Пошук Мистецьких творів за стилем, жанром, ціною

```

Search:
1. search by style, genre, price
2. search by country, availability, price
3. search for each country's most expensive work of art
4. back
Enter your choice: 1
Enter style: Realism
Enter genre: Cityscapes
Enter lowest price: 15000
Enter highest price: 25000

```

```

Works of art:
+-----+-----+-----+-----+-----+
| ID | Name | Customer | Author | Price |
+-----+-----+-----+-----+
| 14338 | Sublime Essence Converging | None | 2443685 | 22630 |
| 14857 | Enigmatic Fusion Discovering | 2480696 | 2453988 | 21462 |
| 14865 | Dynamic Sculpture Awakening | None | 2450272 | 15104 |
| 17346 | Surreal Inception Enchanting | None | 2437627 | 21674 |
| 20360 | Dynamic Metamorphosis Reflecting | None | 2436893 | 22037 |
| 22575 | Captivating Dream Embarking | None | 2438763 | 20774 |
| 23148 | Tranquil Line Evoking | 2477842 | 2441045 | 22111 |
| 35181 | Expressive Canvas Illuminating | None | 2435449 | 16858 |
| 35521 | Provocative Cascade Embarking | None | 2442453 | 16191 |
| 39939 | Experimental Harmony Mesmerizing | None | 2438607 | 19437 |
| 42424 | Serene Reverie Whirling | 2468745 | 2443598 | 24834 |
| 45443 | Provocative Whisper Escaping | 2457285 | 2436178 | 18672 |
| 49116 | Enigmatic Line Embarking | 2477312 | 2456982 | 19145 |
| 50030 | Provocative Eclipse Inspiring | None | 2435828 | 23377 |
| 62665 | Harmonious Aurora Conveying | None | 2441493 | 21020 |
+-----+-----+-----+-----+-----+
Час виконання запиту: 5.9991 мс

```

Лістинг:

controller:

```

def filter_by_price_style_genre(self):
    style = self.view.get_style()
    genre = self.view.get_genre()
    price_from, price_to = self.view.get_price_range()
    start_time = time.time()

```



```

    arts = self.model.filter_by_price_style_genre(style, genre, price_from,
price_to)
    if not arts:
        self.view.show_message("There are no works of art with those
parameters!")
        return
    end_time = time.time()
    self.view.show_arts_filtered_by_price_style_genre(arts)
    execution_time = (end_time - start_time) * 1000
    self.view.show_message(f"Час виконання запиту: {execution_time:.4f} мс")

```

view:

```

def get_price_range(self):
    price_from = self.input_digit("Enter lowest price: ", True, sys.maxsize)
    price_to = self.input_digit("Enter highest price: ", True, sys.maxsize)
    return price_from, price_to

def get_genre(self):
    genre = self.input_text("Enter genre: ", True, 50)
    return genre

def get_style(self):
    style = self.input_text("Enter style: ", True, 50)
    return style

```

model:

```

def filter_by_price_style_genre(self, style, genre, price_from, price_to):
    c = self.conn.cursor()
    c.execute('''SELECT "artID", "name", "buyerID", "creatorID", price
                FROM "Work of art"
                WHERE style = %s AND genre = %s AND price >= %s AND price <= %s
                ''',
                (style, genre, price_from, price_to))
    return c.fetchall()

```

- Пошук Мистецьких творів за країною, доступністю, ціною

```
Search:
1. search by style, genre, price
2. search by country, availability, price
3. search for each country's most expensive work of art
4. back
Enter your choice: 2
Enter country: Ukraine
Enter lowest price: 14000
Enter highest price: 18000
Is available?
1. True
2. False
Enter your choice: 2
```

Works of art in Ukraine:

ID	Name	Style	Genre	Author	Price
15041	Colorful Dream Chasing	Cubism	Landscape	2433724	16549
15520	Vibrant Reverie Discovering	Cubism	Mythological	2443098	16842
15607	Radiant Luminance Enchanting	Post-Impressionism	Genre	2447385	15124
20437	Innovative Whisper Unveiling	Art Deco	Flower	2445733	14144
21057	Harmonious Dream Navigating	Abstract	Flower	2451002	15530
21128	Mysterious Infinity Evoking	Conceptual	Animal	2434529	15271
24194	Whimsical Metamorphosis Transforming	Minimalism	Animal	2452109	14877
24597	Whimsical Paradox Unveiling	Impressionism	Still-Life	2442165	16874
28061	Sublime Cascade Evoking	Fine Art	History	2452202	15280
28641	Glorious Mirage Converging	Photorealistic	Miniature	2437523	15848
30289	Tranquil Fusion Mesmerizing	Art Deco	Religious	2437760	16798
31821	Mysterious Whisper Harmonizing	Post-Impressionism	Miniature	2441824	14122
43782	Tranquil Luminance Elevating	Symbolism	Genre	2453234	14705
46984	Vibrant Silhouette Conveying	Figurative	Still-Life	2451119	16980
49274	Cinematic Resonance Enchanting	Symbolism	History	2456778	16585
50000	Mysterious Cascade Reflecting	Minimalism	Miniature	2434277	15385
50918	Provocative Luminance Transcending	Fine Art	Cityscapes	2437954	17750
62122	Serene Symphony Integrating	Art Deco	Marine Art	2453829	14960

Час виконання запиту: 15.8648 мс

Лістинг:

controller:

```
def filter_by_country_price(self):
    country = self.view.get_country()
    price_from, price_to = self.view.get_price_range()
    availability = self.view.get_availability()
    start_time = time.time()
    arts = self.model.filter_by_country_price(country, price_from, price_to, availability)
    if not arts:
```

```

        self.view.show_message("There are no works of art with those parameters!")
        return

    end_time = time.time()
    self.view.show_arts_filtered_by_country_price(arts, country)

    execution_time = (end_time - start_time) * 1000
    self.view.show_message(f"Час виконання запиту: {execution_time:.4f} мс")

```

view:

```

def get_country(self):
    country = self.input_text("Enter country: ", True, 25)
    return country

def get_availability(self):
    availability = self.get_bool("Is available?")
    return availability

def get_bool(self, message):
    while True:
        print(message)
        print("1. True")
        print("2. False")
        choice = input("Enter your choice: ")
        if choice == '1':
            return True
        elif choice == '2':
            return False
        print("Wrong input data! Try again.")

```

model:

```

def filter_by_country_price(self, country, price_from, price_to, availability):
    c = self.conn.cursor()
    c.execute(f'''SELECT WA."artID", WA."name", WA."style", WA."genre", A."userID", WA.price
                FROM "Work of art" AS WA
                INNER JOIN "Artist" AS A ON WA."creatorID" = A."userID"
                INNER JOIN "User" AS U ON A."userID" = U."userID"
                WHERE U.country = %s AND
                CASE

```

```

        WHEN {availability} = true THEN WA."buyerID" IS NULL
        ELSE WA."buyerID" IS NOT NULL
    END
    AND price >= %s AND price <= %s;
    '',
    (country, price_from, price_to))
return c.fetchall()

```

- Пошук найдорожчих Мистецьких творів в кожній країні

Search:

1. search by style, genre, price
2. search by country, availability, price
3. search for each country's most expensive work of art
4. back

Enter your choice: 3

Country	ID	Name	Style	Genre	Author	Price
Afghanistan	64051	Dynamic Dream Elevating	Art Deco	Flower	2448033	99805
Algeria	38259	Sublime Reflection Enchanting	Fine Art	Cityscapes	2443016	99688
Argentina	44999	Mysterious Cascade Unraveling	Abstract	Battle And War	2456556	99947
Australia	43911	Bold Metamorphosis Conveying	Realism	Marine Art	2439926	99692
Bangladesh	17618	Surreal Reflection Embracing	Art Deco	Marine Art	2454441	99866
Brazil	14956	Ethereal Journey Embarking	Post-Impressionism	Battle And War	2456322	99963
Cambodia	63131	Colorful Sculpture Harmonizing	Street Art	History	2442184	99866
Canada	27422	Innovative Eclipse Discovering	Figurative	Cityscapes	2435735	99907
Chile	46580	Ephemeral Journey Whirling	Academic Art	Cityscapes	2454668	99905
China	41231	Ephemeral Sculpture Resonating	Surrealism	Miniature	2454296	99876
Colombia	20699	Ethereal Sculpture Harmonizing	Photorealistic	History	2447604	99667
DR Congo	18903	Serene Cascade Converging	Impressionism	Animal	2454055	99979
Egypt	38041	Vibrant Harmony Envisioning	Photorealistic	Landscape	2455474	99957
Ethiopia	29394	Bold Paradox Transforming	Street Art	Marine Art	2455795	99926
France	33471	Cinematic Momentum Elevating	Impressionism	Animal	2449903	99927
Germany	16180	Innovative Reverie Unraveling	Art Deco	Cityscapes	2452842	99995
Ghana	37384	Emotional Line Illuminating	Realism	History	2450039	99950
India	35995	Dynamic Harmony Harmonizing	Figurative	History	2445250	99942
Indonesia	57808	Bold Horizon Blossoming	Contemporary Art	Religious	2453130	99888
Iran	20401	Enigmatic Reverie Navigating	Abstract	Mythological	2441541	99813

	Iraq		22878		Enigmatic Dream Embracing		Contemporary Art		Religious		2450096		99719	
	Italy		15206		Fantastic Momentum Evoking		Naïve Art		Portrait		2452348		99972	
	Japan		41458		Experimental Horizon Resonating		Photorealistic		Portrait		2456562		99946	
	Kenya		38778		Radiant Infinity Awakening		Figurative		Portrait		2449435		99952	
	Madagascar		37385		Ephemeral Infinity Awakening		Surrealism		History		2453419		99940	
	Malaysia		44428		Intricate Solitude Resonating		Art Deco		History		2438202		99786	
	Mexico		37693		Captivating Paradox Reflecting		Art Deco		Landscape		2443960		99997	
	Morocco		40782		Dynamic Reverie Envisioning		Cubism		Still-Life		2456535		99999	
	Myanmar		41745		Timeless Panorama Evoking		Cubism		Genre		2441565		99995	
	Netherlands		55182		Enigmatic Line Harmonizing		Naïve Art		Religious		2436400		99979	
	Nigeria		46049		Innovative Momentum Harmonizing		Symbolism		Flower		2437526		99855	
	Pakistan		57056		Captivating Panorama Illuminating		Expressionism		Cityscapes		2443066		99930	
	Peru		31251		Experimental Horizon Converging		Fine Art		Landscape		2456777		99958	
	Philippines		60600		Expressive Infinity Celebrating		Street Art		Genre		2432463		99948	
	Poland		26962		Mysterious Panorama Embarking		Surrealism		Still-Life		2434297		99896	
	Saudi Arabia		58405		Harmonious Journey Celebrating		Symbolism		Battle And War		2433163		99906	
	South Africa		23701		Fantastic Cascade Unveiling		Photorealistic		Mythological		2456752		99974	
	South Korea		48260		Mesmerizing Cascade Discovering		Impressionism		Battle And War		2445958		99975	
	Sudan		27040		Intricate Paradox Elevating		Pop Art		Battle And War		2444723		99846	
	Thailand		61516		Innovative Momentum Blossoming		Expressionism		Battle And War		2444866		99990	
	Turkey		39345		Glorious Panorama Transcending		Figurative		Miniature		2439205		99990	
	Uganda		51127		Mysterious Silhouette Chasing		Art Deco		Miniature		2443071		99938	
	Ukraine		59849		Harmonious Reverie Unveiling		Modern		Religious		2432916		99917	
	United Kingdom		22246		Fantastic Labyrinth Unraveling		Pop Art		Portrait		2441974		99955	
	USA		32943		Fantastic Harmony Whirling		Pop Art		Portrait		2434329		99933	
	Uzbekistan		53974		Whimsical Sculpture Inspiring		Conceptual		Miniature		2452702		99912	
	Venezuela		63439		Harmonious Horizon Exploring		Photorealistic		Cityscapes		2446512		99951	
	Vietnam		34741		Provocative Silhouette Unraveling		Cubism		Interior		2449500		99869	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+														

Час виконання запиту: 134.8405 мс

Лістинг:

controller:

```
def filter_by_most_exp_in_country(self):
    start_time = time.time()
    arts = self.model.filter_by_most_exp_in_country()
    if not arts:
        self.view.show_message("There are no works of art with those parameters!")
        return

    end_time = time.time()
    self.view.show_arts_filtered_by_most_exp_in_country(arts)

    execution_time = (end_time - start_time) * 1000
    self.view.show_message(f"Час виконання запиту: {execution_time:.4f} мс")
```

model:

```
def filter_by_most_exp_in_country(self):
    c = self.conn.cursor()
    c.execute('')
```

```

WITH ranked_artworks AS (
    SELECT
        U.country,
        WA."artID",
        WA."name",
        WA."style",
        WA."genre",
        A."userID",
        WA.price,
        ROW_NUMBER() OVER (PARTITION BY U.country ORDER BY WA.price DESC) AS rank
    FROM
        "Work of art" AS WA
    INNER JOIN
        "Artist" AS A ON WA."creatorID" = A."userID"
    INNER JOIN
        "User" AS U ON A."userID" = U."userID"
    WHERE
        U.country IS NOT NULL
)
SELECT
    country,
    "artID",
    "name",
    "style",
    "genre",
    "userID",
    price
FROM
    ranked_artworks
WHERE
    rank = 1;
'',)
return c.fetchall()

```

Завдання №4

Програмний код виконати згідно шаблону MVC.

- Ілюстрації програмного коду модуля “Model”, згідно із шаблоном MVC

```

import psycopg2
from data_fill import Data

class Model:

```

```

def __init__(self):
    self.conn = psycopg2.connect(
        dbname='Arts Platform',
        user='postgres',
        password='qwerty',
        host='localhost',
        port=5432
    )
    self.data_fill = Data()
    self.create_random_data_table("styles", self.data_fill.styles)
    self.create_random_data_table("genres", self.data_fill.genres)
    self.create_random_data_table("countries", self.data_fill.countries)
    self.create_random_data_table("names", self.data_fill.names)
    self.create_random_data_table("surnames", self.data_fill.surnames)
    self.create_random_data_table("adjectives", self.data_fill.adjectives)
    self.create_random_data_table("nouns", self.data_fill.nouns)
    self.create_random_data_table("verbs", self.data_fill.verbs)

# ----- User
def add_user(self, name, country, info):
    c = self.conn.cursor()
    c.execute('INSERT INTO "User" ("name", country, "info") VALUES (%s, %s, %s) RETURNING
"userID"', (name, country, info))
    user_id = c.fetchone()[0]
    self.conn.commit()
    return user_id

def get_all_users(self):
    c = self.conn.cursor()
    c.execute('SELECT * FROM "User" ORDER BY "userID" ASC ')
    return c.fetchall()

def edit_user(self, user_id, name, country, info):
    c = self.conn.cursor()
    c.execute('UPDATE "User" SET "name"=%s, "country"=%s, "info"=%s WHERE "userID"=%s',
(name, country, info, user_id))
    self.conn.commit()

# ----- Artist
def add_artist(self, name, country, info):
    c = self.conn.cursor()
    userID = self.add_user(name, country, info)
    c.execute('INSERT INTO "Artist" ("userID") VALUES (%s)', (userID,))
    self.conn.commit()

def get_all_artists(self):
    c = self.conn.cursor()
    c.execute(''''SELECT U."userID", U."name", U."country", U."info" FROM "User" AS U
INNER JOIN "Artist" ON U."userID" = "Artist"."userID"
ORDER BY "userID" ASC ;''')
    return c.fetchall()

def get_artist(self, user_id):

```

```

c = self.conn.cursor()
c.execute('''SELECT U."userID", U."name", U."country", U."info" FROM "User" AS U
            INNER JOIN "Artist" ON U."userID" = "Artist"."userID"
            WHERE "Artist"."userID" = %s;''', (user_id,))

artist = c.fetchone()
return artist

def remove_artist(self, user_id):
    c = self.conn.cursor()

    c.execute('DELETE FROM "Work of art" WHERE "creatorID"=%s', (user_id,))
    c.execute('DELETE FROM "Artist" WHERE "userID"=%s', (user_id,))

    c.execute('DELETE FROM "User" WHERE "userID"=%s', (user_id,))
    self.conn.commit()

def rand_fill_artist(self, count):
    c = self.conn.cursor()

    c.execute(f'''
DO $$
DECLARE
    i INT := 1;
BEGIN
    WHILE i <= {count} LOOP

        WITH inserted_user AS (
            INSERT INTO "User" ("name", country, "info")
            VALUES (
                (SELECT "name" FROM names ORDER BY random() LIMIT 1)
                || ' ' ||
                (SELECT "name" FROM surnames ORDER BY random() LIMIT 1)
                ,
                (SELECT "name" FROM countries ORDER BY random() LIMIT 1)
                ,
                chr(trunc(65 + random()*26)::int)||chr(trunc(65 +
random()*26)::int)||chr(trunc(65 + random()*26)::int)
            )
            RETURNING "userID"
        )

        INSERT INTO "Artist" ("userID") VALUES ((SELECT "userID" FROM inserted_user));

        i := i + 1;
    END LOOP;
END $$;
''')

    self.conn.commit()

def show_artists_arts(self, userID):
    c = self.conn.cursor()
    c.execute(''''

```



```

SELECT "artID", "name", price, "style", "genre", "buyerID" FROM "Work of art"
INNER JOIN "Artist" ON "Work of art"."creatorID" = "Artist"."userID"
WHERE "Artist"."userID" = %s;
'', (userID,))
return c.fetchall()

```

```
# ----- Collector
```

```

def add_collector(self, name, country, info):
    c = self.conn.cursor()
    userID = self.add_user(name, country, info)
    c.execute('INSERT INTO "Collector" ("userID") VALUES (%s)', (userID,))
    self.conn.commit()

def get_all_collectors(self):
    c = self.conn.cursor()
    c.execute(''SELECT U."userID", U."name", U."country", U."info" FROM "User" AS U
              INNER JOIN "Collector" ON U."userID" = "Collector"."userID"
              ORDER BY "userID" ASC ;'')
    return c.fetchall()

def get_collector(self, user_id):
    c = self.conn.cursor()
    c.execute(''SELECT U."userID", U."name", U."country", U."info" FROM "User" AS U
              INNER JOIN "Collector" ON U."userID" = "Collector"."userID"
              WHERE "Collector"."userID" = %s;'', (user_id,))

    collector = c.fetchone()
    return collector

def remove_collector(self, user_id):
    c = self.conn.cursor()

    c.execute('UPDATE "Work of art" SET "buyerID"=NULL WHERE "buyerID"=%s', (user_id,))
    c.execute('DELETE FROM "Collector" WHERE "userID"=%s', (user_id,))

    c.execute('DELETE FROM "User" WHERE "userID"=%s', (user_id,))
    self.conn.commit()

def rand_fill_collector(self, count):
    c = self.conn.cursor()

    c.execute(f'''
DO $$
DECLARE
    i INT := 1;
BEGIN
    WHILE i <= {count} LOOP

        WITH inserted_user AS (
            INSERT INTO "User" ("name", country, "info")
            VALUES (
                (SELECT "name" FROM names ORDER BY random() LIMIT 1)
                || ' ' ||
                (SELECT "name" FROM surnames ORDER BY random() LIMIT 1)

```

```

        ,
        (SELECT "name" FROM countries ORDER BY random() LIMIT 1)
        ,
        chr(trunc(65 + random()*26)::int)||chr(trunc(65 +
random()*26)::int)||chr(trunc(65 + random()*26)::int)
        )
        RETURNING "userID"
    )

    INSERT INTO "Collector" ("userID") VALUES ((SELECT "userID" FROM
inserted_user));

    i := i + 1;
END LOOP;
END $$;
'''

self.conn.commit()

def show_collectors_collection(self, userID):
    c = self.conn.cursor()
    c.execute('''
        SELECT "artID", "name", price, "style", "genre", "creatorID" FROM "Work of art"
        INNER JOIN "Collector" ON "Work of art"."buyerID" = "Collector"."userID"
        WHERE "Collector"."userID" = %s;
    ''', (userID,))
    return c.fetchall()

# ----- Work of art
def add_art(self, name, price, style, genre, creatorID):
    c = self.conn.cursor()
    c.execute('INSERT INTO "Work of art" ("name", price, "style", "genre", "buyerID",
"creatorID") VALUES (%s, %s, %s, %s, null, %s)', (name, price, style, genre, creatorID))
    self.conn.commit()

def get_all_arts(self):
    c = self.conn.cursor()
    c.execute('SELECT * FROM "Work of art" ORDER BY "artID" ASC ')
    return c.fetchall()

def get_art(self, art_id):
    c = self.conn.cursor()
    c.execute('SELECT * FROM "Work of art" WHERE "artID"=%s', (art_id,))
    art = c.fetchone()
    return art

def edit_art(self, art_id, name, price, style, genre, buyerID, creatorID):
    c = self.conn.cursor()

    c.execute('UPDATE "Work of art" SET "name"=%s, price=%s, "style"=%s, "genre"=%s,
"creatorID"=%s WHERE "artID"=%s', (name, price, style, genre, creatorID, art_id))

```

```

        if not buyerID:
            c.execute('UPDATE "Work of art" SET "buyerID"=null WHERE "artID"=%s', (art_id,))
        else:
            c.execute('UPDATE "Work of art" SET "buyerID"=%s WHERE "artID"=%s', (buyerID,
art_id))

        self.conn.commit()

def remove_art(self, art_id):
    c = self.conn.cursor()
    c.execute('DELETE FROM "Work of art" WHERE "artID"=%s', (art_id,))
    self.conn.commit()

def buy_art(self, buyer_id, art_id):
    c = self.conn.cursor()
    c.execute('UPDATE "Work of art" SET "buyerID"=%s WHERE "artID"=%s', (buyer_id,
art_id))
    self.conn.commit()

def get_arts_buyer(self, art_id):
    c = self.conn.cursor()
    c.execute('SELECT * FROM "Work of art" WHERE "artID"=%s', (art_id,))
    buyer_id = c.fetchone()[5]
    return buyer_id

def rand_fill_arts(self, count):
    c = self.conn.cursor()

    c.execute(f'''
DO $$
DECLARE
    i INT := 1;
BEGIN
    WHILE i <= {count} LOOP

        INSERT INTO "Work of art" ("name", price, "style", "genre", "buyerID",
"creatorID")
        VALUES (
            (SELECT "name" FROM adjectives ORDER BY random() LIMIT 1)
            || ' ' ||
            (SELECT "name" FROM nouns ORDER BY random() LIMIT 1)
            || ' ' ||
            (SELECT "name" FROM verbs ORDER BY random() LIMIT 1)
            ,
            trunc(random()*100000)::int
            ,
            (SELECT "name" FROM styles ORDER BY random() LIMIT 1)
            ,
            (SELECT "name" FROM genres ORDER BY random() LIMIT 1)
            ,
            CASE WHEN random() < 0.5
                THEN (SELECT "userID" FROM "Collector" ORDER BY random() LIMIT 1)
                ELSE null
    ''')

```

```

        END,
        (SELECT "userID" FROM "Artist" ORDER BY random() LIMIT 1)
    );
    i := i + 1;
END LOOP;
END $$;
'''

self.conn.commit()

# ----- Search
def sort_by_price(self, price_from, price_to):
    c = self.conn.cursor()
    c.execute('SELECT * FROM "Work of art" WHERE price >= %s AND price <= %s ORDER BY
price ASC', (price_from, price_to))
    return c.fetchall()

def filter_by_price_style_genre(self, style, genre, price_from, price_to):
    c = self.conn.cursor()
    c.execute(''SELECT "artID", "name", "buyerID", "creatorID", price FROM "Work of art"
WHERE style = %s AND genre = %s AND price >= %s AND price <= %s
'',
            (style, genre, price_from, price_to))
    return c.fetchall()

def filter_by_country_price(self, country, price_from, price_to, availability):
    c = self.conn.cursor()
    c.execute(f''SELECT WA."artID", WA."name", WA."style", WA."genre", A."userID",
WA.price
                FROM "Work of art" AS WA
                INNER JOIN "Artist" AS A ON WA."creatorID" = A."userID"
                INNER JOIN "User" AS U ON A."userID" = U."userID"
                WHERE U.country = %s AND
                CASE
                    WHEN {availability} = true THEN WA."buyerID" IS NULL
                    ELSE WA."buyerID" IS NOT NULL
                END
                AND price >= %s AND price <= %s;
            '',
            (country, price_from, price_to))
    return c.fetchall()

def filter_by_most_exp_in_country(self):
    c = self.conn.cursor()
    c.execute(''
WITH ranked_artworks AS (
    SELECT
        U.country,
        WA."artID",
        WA."name",
        WA."style",
        WA."genre",
        A."userID",

```

```

        WA.price,
        ROW_NUMBER() OVER (PARTITION BY U.country ORDER BY WA.price DESC) AS rank
    FROM
        "Work of art" AS WA
    INNER JOIN
        "Artist" AS A ON WA."creatorID" = A."userID"
    INNER JOIN
        "User" AS U ON A."userID" = U."userID"
    WHERE
        U.country IS NOT NULL
)
SELECT
    country,
    "artID",
    "name",
    "style",
    "genre",
    "userID",
    price
FROM
    ranked_artworks
WHERE
    rank = 1;
''' ,)
return c.fetchall()

```

----- Fill data

```

def create_random_data_table(self, name, data):
    c = self.conn.cursor()
    c.execute('''
        SELECT EXISTS (
            SELECT 1
            FROM information_schema.tables
            WHERE table_name = %s
        );
    ''', (name,))
    table_exists = c.fetchone()

    if not table_exists[0]:
        c.execute(f'''
            CREATE TABLE {name} (
                id INTEGER PRIMARY KEY,
                name TEXT
            );
        ''')

        for i, value in enumerate(data):
            c.execute(f'INSERT INTO {name} (id, name) VALUES (%s, %s)', (i, value,))

        self.conn.commit()
        print(f"Таблицю {name} створено та заповнено.")
    else:

```

```
print(f"Таблиця {name} вже існує.")
```

● Короткий опис функцій модуля

1. **add_user** – додає користувача в таблицю “User”;
2. **get_all_users** – повертає список всіх користувачів;
3. **edit_user** – редагує дані користувача;
4. **add_artist** – додає Художника в таблицю “User” та його id в таблицю “Artist”;
5. **get_all_artists** – повертає дані всіх Художників;
6. **get_artist** – повертає дані про конкретного Художника;
7. **remove_artist** – видаляє Художника з бази;
8. **rand_fill_artist** – генерує дані про Художників у таблиці “User” та вносить їхні id у таблицю “Artist”;
9. **show_artists_arts** – повертає всі Мистецькі твори, створені Художником;
10. **add_collector** – додає Колекціонера в таблицю “User” та його id в таблицю “Collector”;
11. **get_all_collectors** – повертає дані всіх Колекціонерів;
12. **get_collector** – повертає дані про конкретного Колекціонера;
13. **remove_collector** – видаляє Колекціонера з бази;
14. **rand_fill_collector** – генерує дані про Колекціонерів у таблиці “User” та вносить їхні id у таблицю “Collector”;
15. **show_collectors_collection** – повертає всі Мистецькі твори, куплені Колекціонером;
16. **add_art** – додає Мистецький твір в таблицю “Work of art”
17. **get_all_arts** – повертає дані всіх Мистецьких творів;
18. **get_art** – повертає дані про конкретного Мистецького твору;
19. **edit_art** – редагує дані Мистецького твору;
20. **remove_art** – видаляє Мистецький твір з бази;
21. **buy_art** – виконує покупку Колекціонером Мистецького твору (вносить id Колекціонера в атрибут “buyerID”);
22. **get_arts_buyer** – повертає Колекціонера, що придбав Мистецький твір;
23. **rand_fill_arts** – генерує дані про Мистецькі твори у таблиці “Work of art”;

24. **filter_by_price_style_genre** – виконує пошук Мистецьких творів за стилем, жанром, ціною.

Користувач вводить стиль, жанр, мінімальну та максимальну ціну. Повертає список всіх Мистецьких творів, які відповідають цим параметрам.

25. **filter_by_country_price** – виконує пошук Мистецьких творів за країною, доступністю, ціною. Повертає список всіх Мистецьких творів:

- які створені Художником, у якого атрибут “країна” дорівнює введеному значенню;
- у яких “buyerID” не пустий, якщо доступність обрана False, або “buyerID” пустий, якщо доступність обрана True;
- ціна яких знаходиться в межах мін. та макс. введених цін.

26. **filter_by_most_exp_in_country** – повертає список найдорожчих Мистецьких творів в кожній країні.