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»

Виконав:
студент групи КВ-81
Путієнко Сергій
Перевірив:
Петрашенко А.В.

Київ 2020

Метою роботи є здобуття вмінь програмування прикладних додатків баз даних PostgreSQL

Завдання

Загальне завдання роботи полягає у наступному:

1. Реалізувати функції 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).

Деталізоване завдання:

1. Забезпечити можливість уведення/редагування/вилучення даних у таблицях бази даних з можливістю контролю відповідності типів даних атрибутів таблиць (рядків, чисел, дати/часу). Для контролю пропонується два варіанти: контроль при введенні (валідація даних) та перехоплення помилок (try..except) від сервера PostgreSQL при виконанні відповідної команди SQL. Особливу увагу варто звернути на дані таблиць, що мають зв’язок 1:N. При цьому з боку батьківської таблиці необхідно контролювати **вилучення** рядків за умови наявності даних у підлеглій таблиці. З точки зору підлеглої таблиці варто контролювати наявність відповідного рядка у батьківській таблиці при виконанні **внесення** нових даних. Унеможливити виведення програмою системних помилок на екрані шляхом їх перехоплення і адекватної обробки. Внесення даних виконується користувачем у консольному вікні програми.

2. Забезпечити можливість автоматичної генерації великої кількості даних у таблицях за допомогою вбудованих у PostgreSQL функцій роботи з псевдовипадковими числами. Дані мають бути згенерованими **не мовою програмування, а відповідним SQL-запитом**!

Приклад генерації 100 псевдовипадкових чисел:

```
select trunc(random()*1000)::int
from generate_series(1,100)
```

Data Output | Explain | Messages | Notific

	trunc integer
1	368
2	773
3	29
4	66
5	497
6	956

Приклад генерації 5 псевдовипадкових рядків:

```
select chr(trunc(65+random()*25)::int) || chr(trunc(65+random()*25)::int)
from generate_series(1,5)
```

Data Output | Explain | Messages | Notifications

	?column? text
1	NE
2	MQ
3	RN
4	DW
5	DA

Приклад генерації псевдовипадкової мітки часу з діапазону доступний за посиланням.

Кількість даних для генерування має вводити користувач з клавіатури. Для тесту взяти 100 000 записів для однієї-двох таблиць.

Особливу увагу слід звернути на відповідність даних вимогам зовнішніх ключів з метою уникнення помилок порушення обмежень цілісності (foreign key).

3. Для реалізації пошуку необхідно підготувати 3 запити, що включають дані з декількох таблиць і фільтрують рядки за 3-4 атрибутами цих таблиць. Забезпечити можливість уведення конкретних значень констант для фільтрації з клавіатури користувачем. Крім того, після виведення даних необхідно вивести час виконання запиту у мілісекундах. Перевірити швидкодію роботи запитів на попередньо згенерованих даних.

4. Програмний код організувати згідно шаблону Model-View-Controller(MVC). Приклад організації коду згідно шаблону доступний за даним посиланням. При цьому модель, подання та контролер мають бути реалізовані у окремих файлах. Для доступу до бази даних використовувати **лише мову SQL** (без ORM).

Рекомендована бібліотека взаємодії з PostgreSQL Psycopg2: http://initd.org/psycopg/docs/usage.html)

Завдання 1

Дана програма дозволяє можливість уведення, редагування, вилучення або перегляду даних вибраної таблиці.

```
1. Редагувати дані таблиці
2. Генерація випадкових даних
3. Пошук даних
4. Вихід
Виберіть один з пуктів...
1
1. Films
2. Cinemas
3. Cinema_Session
4. Sessions
Виберіть одну з таблиць...
1
1. Додати запис
2. Оновити дані запису
3. Відобразити дані
4. Видалити дані
5. Вихід
Виберіть один з пуктів...
```

Програма читає дані типів з бази даних і пропонує користувачу ввести відповідні дані. Приклад введення нових даних запису до таблиці Films.

```
Введіть значення поля FilmID
Тип поля число
111
Введіть значення поля f_name
Тип поля текст
Ford v Ferrari
Введіть значення поля release_year
Тип поля число
2019
Введіть значення поля f_genre
Тип поля текст
Sport
Введіть значення поля f_duration
Тип поля число
152
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Успіх! Запис з ідентифікатором 111 було додано до таблиці Films !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Оновлення даних таблиці Films:

```
2
Введіть значення поля FilmID
Тип поля число
111
Введіть значення поля f_name
Тип поля текст
Ford v Ferrari
Введіть значення поля release_year
Тип поля число
2019
Введіть значення поля f_genre
Тип поля текст
Sport
Введіть значення поля f_duration
Тип поля число
153
---   ---   ---   ---   ---   ---   ---   ---   ---   ---   --
Запис 111 було змінено на [111, 'Ford v Ferrari', 2019, 'Sport', 153]
---   ---   ---   ---   ---   ---   ---   ---   ---   ---   --
Для продовження натисніть Enter...
```

Видалення запису з таблиці Films:

```
1. Додати запис
2. Оновити дані запису
3. Відобразити дані
4. Видалити дані
5. Вихід
Виберіть один з пуктів...
4
Введіть ключ для запису який необхідно видалити
111
---------------------------------------------------------------
Елемент 111 було успішно видалено!
---------------------------------------------------------------
Для продовження натисніть Enter...
```

Перегляд записів:

```
1. Додати запис
2. Оновити дані запису
3. Відобразити дані
4. Видалити дані
5. Вихід
Виберіть один з пуктів...
3
['FilmID', 'f_name', 'release_year', 'f_genre', 'f_duration']
--- ТАБЛИЦЯ FILMS ---
1. (23436, 'InterStellar', 2014, 'Fantastic', 168)
2. (53234, 'Joker', 2019, 'Drama', 116)
3. (74574, 'Gentlemen', 2019, 'Criminal', 113)
Для продовження натисніть Enter...
```

Приклад введення запису з ключем, що вже існує у базі даних:

```
Введіть значення поля FilmID
Тип поля число
23436
Введіть значення поля f_name
Тип поля текст
Ford v Ferrari
Введіть значення поля release_year
Тип поля число
2019
Введіть значення поля f_genre
Тип поля текст
Sport
Введіть значення поля f_duration
Тип поля число
155
************************************************************
База даних вже містить елемент 23436 в таблиці Films !
ОШИБКА:  повторяющееся значение ключа нарушает ограничение уникальности "Films_pkey"
DETAIL:  Ключ ("FilmID")=(23436)" уже существует.

************************************************************
Для продовження натисніть Enter...
```

Завдання 2

Дана програма передбачає створення числових даних, рядкових даних або даних, що відповідають датам. Приклад вводу даних для генерації записів у таблиці Films.

```
1. Films
2. Cinemas
3. Cinema_Session
4. Sessions
Виберіть одну з таблиць...
1
Було обрано таблицю Films
Введіть кількість записів, що потрібно згенерувати
100000
Введіть дані для поля FilmID
Введіть максимальне число
2000000
Введіть дані для поля f_name
Введіть довжину рядка
10
Введіть дані для поля release_year
Введіть максимальне число
3000
Введіть дані для поля f_genre
Введіть довжину рядка
10
Введіть дані для поля f_duration
Введіть максимальне число
500
```

Результати:

	FilmID [PK] integer	f_name character varying (20)	release_year integer	f_genre character varying (10)	f_duration integer
1	8	GICPSVXCVW	1878	WIDFPXFRJX	93
2	23	FWXJYMXCWC	1978	MOQAOVHSIC	294
3	40	JJDUDUBJJT	217	MKSUXSUKQW	360
4	59	GKQIXCBSHE	2239	MECACUNOYM	321
5	91	UODDFFVQTT	920	SMDJJWKPPE	222
6	92	UFIBRMHJPV	2467	SPPPDNWVTF	487
7	94	MJVVNBQCXP	20	WJBGDYYESH	411
8	105	GDCFAFLJTH	248	CJFMTETMGT	425
9	140	IMGUOYYIFX	1281	IPBSXXWBFF	148
10	157	BDRGSIHBJN	2253	WGLUPFLLXL	399
11	159	LFROVFQVBQ	499	VSWYNEJRHF	20
12	174	NJYNHMWFCH	1040	RBIOPDAKBE	471
13	178	FMKDBMHOGS	2210	NPVLIXSCTX	405
14	190	KFDRLTXTRJ	2345	XEAOHJWQIS	400
15	219	ICYJDBEUEU	2156	YNRGKSWIIA	215
16	267	ARQAIYIIWU	2279	SOMPQMSJWH	88

Запити:

```
SELECT trunc(random()*2000000)::int from generate_series(1,100000)
SELECT chr(trunc(65 + random()*25)::int) || chr(trunc(65 + random()*25)::int) ||
chr(trunc(65 + random()*25)::int) || chr(trunc(65 + random()*25)::int) || chr(trunc(65
+ random()*25)::int) || chr(trunc(65 + random()*25)::int) || chr(trunc(65 +
random()*25)::int) || chr(trunc(65 + random()*25)::int) || chr(trunc(65 +
random()*25)::int) || chr(trunc(65 + random()*25)::int) from
generate_series(1,100000)
SELECT trunc(random()*3000)::int from generate_series(1,100000)
SELECT chr(trunc(65 + random()*25)::int) || chr(trunc(65 + random()*25)::int) ||
chr(trunc(65 + random()*25)::int) || chr(trunc(65 + random()*25)::int) || chr(trunc(65
+ random()*25)::int) || chr(trunc(65 + random()*25)::int) || chr(trunc(65 +
random()*25)::int) || chr(trunc(65 + random()*25)::int) || chr(trunc(65 +
random()*25)::int) || chr(trunc(65 + random()*25)::int) from
generate_series(1,100000)
SELECT trunc(random()*500)::int from generate_series(1,100000)
```

Завдання 3

У програмі реалізовано три запиту пошуку:

1. Фільми, що будуть показувати після дати у кінотеатрі
2. Фільми, що будуть показувати після дати та тривалістю у рамках заданого
3. Фільми, назви яких містять словосполучення та відповідного жанру

Перші два використовують дані з чотирьох таблиць, а третій запит виконує пошук по таблиці фільми. Також запити виконують фільтрацію за чотирьома атрибутами.

Запит 1.

```
SELECT f_name, f_genre, start_date, hall_name ' \
    'FROM "Films", "Sessions", "Cinema_Session", "Cinemas" ' \
    'WHERE "Films"."FilmID" = "Sessions"."FilmID"' \
    'AND "Sessions"."SessionID" = "Cinema_Session"."SessionID"' \
    'AND "Cinema_Session"."CinemaID" = "Cinemas"."CinemaID"' \
    'AND "Cinemas".c_name = \'{}\' AND "Sessions".start_date > \'{}\' '
```

Запит 2.

```
'SELECT f_name, f_duration, start_date, c_name ' \
    'FROM "Films", "Sessions", "Cinema_Session", "Cinemas" ' \
    'WHERE "Films"."FilmID" = "Sessions"."FilmID" ' \
    'AND "Sessions"."SessionID" = "Cinema_Session"."SessionID" ' \
    'AND "Cinema_Session"."CinemaID" = "Cinemas"."CinemaID" ' \
    'AND "Sessions"."start_date" > \'{}\' ' \
    'AND "Films".f_duration BETWEEN {} AND {}'
```

Запит 3.

```
'SELECT f_name, f_genre, f_duration, release_year ' \
    'FROM "Films"' \
    'WHERE ' \
    '"Films".f_name LIKE \'%{}%\' ' \
    'AND "Films".f_genre LIKE \'%{}%\' '
```

Далі наведено результати роботи програми з використанням відповідних запитів.

Перший запит:

```
1. Фільми, що будуть показувати після дати у кінотеатрі
2. Фільми, що будуть показувати після дати та тривалістю у рамках заданого
3. Фільми, назви яких містять словосполучення та відповідного жанру
Виберіть одну з дій...
1
Введіть назву кінотеатру
Kiev
Введіть дату після якої будуть показувати фільми (дд/мм/рр)
09-09-2020
Finished 'search_query1' in 0.0047 secs
['f_name', 'f_genre', 'start_date', 'hall_name']
--- ТАБЛИЦЯ FILMS ---
1. ('InterStellar', 'Fantastic', datetime.date(2020, 9, 17), 'Almandine')
Для продовження натисніть Enter...
```

Другий запит:

```
1. Фільми, що будуть показувати після дати у кінотеатрі
2. Фільми, що будуть показувати після дати та тривалістю у рамках заданого
3. Фільми, назви яких містять словосполучення та відповідного жанру
Виберіть одну з дій...
2
Введіть дату після якої будуть показувати фільми (дд/мм/рр)
10-09-2020
Введіть мінімальну тривалість фільму
100
Введіть максимальну тривалість фільму
150
Finished 'search_query2' in 0.0016 secs
['f_name', 'f_duration', 'start_date', 'c_name']
--- ТАБЛИЦЯ FILMS ---
1. ('Joker', 116, datetime.date(2020, 9, 17), 'October')
2. ('Gentlemen', 113, datetime.date(2020, 9, 18), 'Torch')
Для продовження натисніть Enter...
```

Третій запит:

```
1. Фільми, що будуть показувати після дати у кінотеатрі
2. Фільми, що будуть показувати після дати та тривалістю у рамках заданого
3. Фільми, назви яких містять словосполучення та відповідного жанру
Виберіть одну з дій...
3
Введіть словосполучення, яке присутнє у назві фільма
GGG
Введіть словосполучення, яке присутнє у назві жанру
M
Finished 'search_query3' in 0.0171 secs
['f_name', 'f_genre', 'f_duration', 'release_year']
--- ТАБЛИЦЯ FILMS ---
1. ('OOCKTGGGXU', 'PQMFMFWRVG', 217, 2745)
2. ('GGGXYLLQCA', 'MCGFXBLNXH', 478, 726)
3. ('GGGBQLXAFK', 'UGGQLFMBGS', 321, 1098)
4. ('QGGGYXBMKG', 'CBHHMWOYVR', 444, 2221)
5. ('GVGGGYTNWU', 'TSHWMENJPE', 457, 1075)
6. ('LLNGGGSPPS', 'AYBMFGIQGT', 442, 1019)
7. ('YGDGGGRXMT', 'XMXBWNRRLH', 340, 2326)
8. ('GGGTELBJOO', 'PDVPCQVGMF', 162, 933)
9. ('GGGTXXQPUR', 'BDSVVTTEUM', 304, 1068)
10. ('YGGGSETWDK', 'QAMXYTAJHB', 125, 2708)
11. ('FJGGGDIRNW', 'MECONYAYVK', 454, 1469)
12. ('QVETLGGGJF', 'CHBVWYPGMY', 124, 1300)
Для продовження натисніть Enter...
```

Завдання 4

Код програми знаходиться у репозиторію github за посиланням:

https://github.com/CrispBS/Databases-and-controls.git