



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: «Проектування бази даних та
ознайомлення з базовими операціями
СУБД PostgreSQL»

Виконав: студент III курсу

ФПМ групи КВ-81

Прокопчук М.О.

Перевірів:

Київ – 2020

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 введення/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 підлеглий 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).

- Для реалізації пошуку необхідно підготувати 3 запити, що включають дані з декількох таблиць і фільтрують рядки за 3-4 атрибутами цих таблиць. Забезпечити можливість введення 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>)

Вимоги до інтерфейсу користувача

Використовувати консольний інтерфейс користувача.

Вимоги до інструментарію

Середовище для відлагодження SQL-запитів до бази даних – PgAdmin4.

Пункт №1

У даній програмі є два види помилок:

Перша коли недостатньо параметрів для запиту.

Друга коли парсер не може конвертувати string в потрібний тип.

```
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
1
TableName Parameters:
Stadiums
Error: Index was outside the bounds of the array.
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
Stadiums name capacity address
Error: Input string was not in a correct format.
```

Insert:

```
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
1
TableName Parameters:
Stadiums Name 30000 Address
+-----+-----+-----+-----+
|StadiumID|s_name|capacity|address|
+-----+-----+-----+-----+
|144|Name|30000|Address|
+-----+-----+-----+-----+
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
1
TableName Parameters:
Matches 12-02-2020 Name
+-----+-----+-----+
|MatchID|start_time|StadiumID|
+-----+-----+-----+
|126|12-Feb-2020|144|
+-----+-----+-----+
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
1
TableName Parameters:
Matches 13-02-2020 144
+-----+-----+-----+
|MatchID|start_time|StadiumID|
+-----+-----+-----+
|127|13-Feb-2020|144|
+-----+-----+-----+
```

Update:

```
TableName Parameters:
Matches 13-02-2020 144
+-----+-----+-----+
|MatchID   |start_time|StadiumID  |
+-----+-----+-----+
|127       |13-Feb-2020|144        |
+-----+-----+-----+
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
2
TableName New_parameters:
Matches 14-02-2020 144 127
+-----+-----+-----+
|MatchID   |start_time|StadiumID  |
+-----+-----+-----+
|127       |14-Feb-2020|144        |
+-----+-----+-----+
```

Delete & Read:

```
4
TableName:
Teams
+-----+-----+-----+
|TeamID    |t_name    |trainer    |
+-----+-----+-----+
|117       |WB        |RSL        |
+-----+-----+-----+
|118       |SR        |FLN        |
+-----+-----+-----+
|120       |YA        |JMF        |
+-----+-----+-----+
|121       |RW        |INQ        |
+-----+-----+-----+
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
3
TableName ID OR TableName:
Teams
Deleted 4 rows
```

```
TableName:
Teams
+-----+-----+-----+
|TeamID  |t_name  |trainer |
+-----+-----+-----+
|117     |WB      |RSL     |
+-----+-----+-----+
|118     |SR      |FLN     |
+-----+-----+-----+
|119     |SS      |EFQ     |
+-----+-----+-----+
|120     |YA      |JMF     |
+-----+-----+-----+
|121     |RW      |INQ     |
+-----+-----+-----+
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
3
TableName ID OR TableName:
Teams 119
Deleted 1 rows
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
```

Пункт №2

Приклад роботи генерації псевдовипадкових рядків в таблицях Stadium і Team:

```
6
TableName Counter:
Stadiums 100000
Inserted 100000 rows
Insert: 1
Update: 2
Delete: 3
Read: 4
Select: 5
RandomInsert: 6
6
TableName Counter:
Teams 100000
Inserted 100000 rows
```

	StadiumID [PK] integer	s_name character varying (30)	capacity integer	address character varying (60)
99983	100132	JW	20599	MDG
99984	100133	CG	33305	XGY
99985	100134	VH	24971	JOJ
99986	100135	HT	6145	GFW
99987	100136	WR	35897	IQL
99988	100137	MK	87130	SYW
99989	100138	NP	18422	QJP
99990	100139	UM	89360	ERD
99991	100140	GO	90773	LWK
99992	100141	DM	40358	JVN
99993	100142	XC	82915	XNJ
99994	100143	PK	31420	FDI
99995	100144	WA	30489	QVB
99996	100145	SE	68894	RUC
99997	100146	VL	44199	FNG
99998	100147	GN	14460	UHL
99999	100148	GT	46525	RGU
100000	100149	HW	50103	GRH

	TeamID [PK] integer	t_name character varying (30)	trainer character varying (30)
99983	100104	US	NRQ
99984	100105	HN	GNE
99985	100106	JJ	KKS
99986	100107	OG	NER
99987	100108	MF	VJG
99988	100109	GH	BTA
99989	100110	ES	WDK
99990	100111	YV	NCJ
99991	100112	IF	BLM
99992	100113	EP	HLI
99993	100114	VH	DRK
99994	100115	VY	MNM
99995	100116	DJ	TGA
99996	100117	VW	GBM
99997	100118	EG	XKW
99998	100119	EE	FTV
99999	100120	HX	KUA
100000	100121	QH	YKF

Пункт №3

У програмі реалізовано декілька запитів пошуку:

Перший запит повертає таблицю де початок імені команди і початок назви стадіону, на якій грала команда співпадають з тими що ввів користувач.

(Було використано запит LIKE)

Другий запит повертає таблицю команд, котрі за відведений час у хвилинах забили відповідну кількість голів.

Третій запит повертає таблицю матчів, які мають найбільшу кількість голів за матч.

```

Select t_name, trainer, s_name WHERE first char in t_name and s_name what you LIKE: 1
Select MatchID WHERE goals >= count up to minute: 2
Select TeamID WHERE max goals in match: 3
1
S S
+-----+-----+-----+
|s_name   |t_name   |trainer  |
+-----+-----+-----+
|SA       |SK       |RQW     |
+-----+-----+-----+
|SA       |SL       |QTO     |
+-----+-----+-----+
|SA       |SN       |DGT     |
+-----+-----+-----+
|SA       |SN       |UIU     |
+-----+-----+-----+
|SA       |SP       |BKG     |
+-----+-----+-----+
|SA       |SQ       |AFW     |
+-----+-----+-----+
|SA       |SQ       |KWF     |
+-----+-----+-----+
|SY       |SB       |HAJ     |
+-----+-----+-----+
|SY       |SC       |VNF     |
+-----+-----+-----+
|SY       |SL       |CSL     |
+-----+-----+-----+
|SY       |SQ       |NGK     |
+-----+-----+-----+
|SY       |SV       |BDR     |
+-----+-----+-----+
|SY       |SV       |FVO     |
+-----+-----+-----+
Time taken: (02.208) s.

```

```

Select t_name, trainer, s_name WHERE first char in t_name and s_name what you LIKE: 1
Select MatchID WHERE goals >= count up to minute: 2
Select TeamID WHERE max goals in match: 3
2
20 1
+-----+
|MatchID |
+-----+
|12334   |
+-----+
|99835   |
+-----+
|104516  |
+-----+
|106037  |
+-----+
|107657  |
+-----+
|108584  |
+-----+
Time taken: (13.512) s.

```

```

Select t_name, trainer, s_name WHERE first char in t_name and s_name what you LIKE: 1
Select MatchID WHERE goals >= count up to minute: 2
Select TeamID WHERE max goals in match: 3
3
+-----+
|TeamID  |
+-----+
|64640   |
+-----+
Time taken: (11.822) s.

```

Вибірка велась з таблиць які мають 100000 записів кожна. Але для того щоб виводимі таблиці були не громісткі, був поставлений ліміт в запитах.

Запити:

```

SELECT DISTINCT public."Stadiums"."s_name", public."Teams"."t_name",
public."Teams"."trainer" FROM public."Teams" INNER JOIN
public."Games" ON public."Teams"."TeamID" = public."Games"."TeamID"
INNER JOIN public."Matches" ON public."Matches"."MatchID" =
public."Games"."MatchID" INNER JOIN public."Stadiums" ON
public."Stadiums"."StadiumID"=public."Matches"."StadiumID"
WHERE "s_name" LIKE 's_ch%' AND "t_name" LIKE 't_ch%';

```

```

SELECT DISTINCT "MatchID" FROM public."Goals" AS "g"
WHERE (SELECT Count(*) FROM public."Goals" WHERE minute < minute AND
"MatchID" = "g"."MatchID") > count_goals;

```


```



SELECT DISTINCT "TeamID" from public."Goals" AS "m" WHERE (SELECT
Count(*) FROM public."Goals" WHERE public."Goals"."MatchID" =
"m"."MatchID" AND public."Goals"."TeamID" = "m"."TeamID") = SELECT
max(y.m) from (select count(*) AS m
FROM public."Goals" group by public."Goals"."MatchID",
public."Goals"."TeamID") AS y)









```

Пункт №4

Код програми можна знайти за посиланням [тут](#) (репозиторій github).

 master ▾ [DataBaseLabs / Lab2 /](#) Go to file

 FaceandControl Add Lab2, Change Dict Lab1 7288b6e 4 minutes ago  History

..		
 Controller.cs	Add Lab2, Change Dict Lab1	4 minutes ago
 DataBaseModel.png	Add Lab2, Change Dict Lab1	4 minutes ago
 Lab2_ProkopchukMaksym.docx	Add Lab2, Change Dict Lab1	4 minutes ago
 Lab2_ProkopchukMaksym.pdf	Add Lab2, Change Dict Lab1	4 minutes ago
 Model.cs	Add Lab2, Change Dict Lab1	4 minutes ago
 Program.cs	Add Lab2, Change Dict Lab1	4 minutes ago
 README.md	Add Lab2, Change Dict Lab1	4 minutes ago
 View.cs	Add Lab2, Change Dict Lab1	4 minutes ago

README.md

DataBaseLabs

Лабораторна робота № 2. Створення додатку бази даних, орієнтованого на взаємодію з СУБД PostgreSQL. Загальне завдання роботи полягає у наступному: Реалізувати функції внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі №1, засобами консольного інтерфейсу. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі. 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. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).