



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 III курсу

ФПМ групи КВ-84

Голуб Володимир Володимирович

Перевірів:

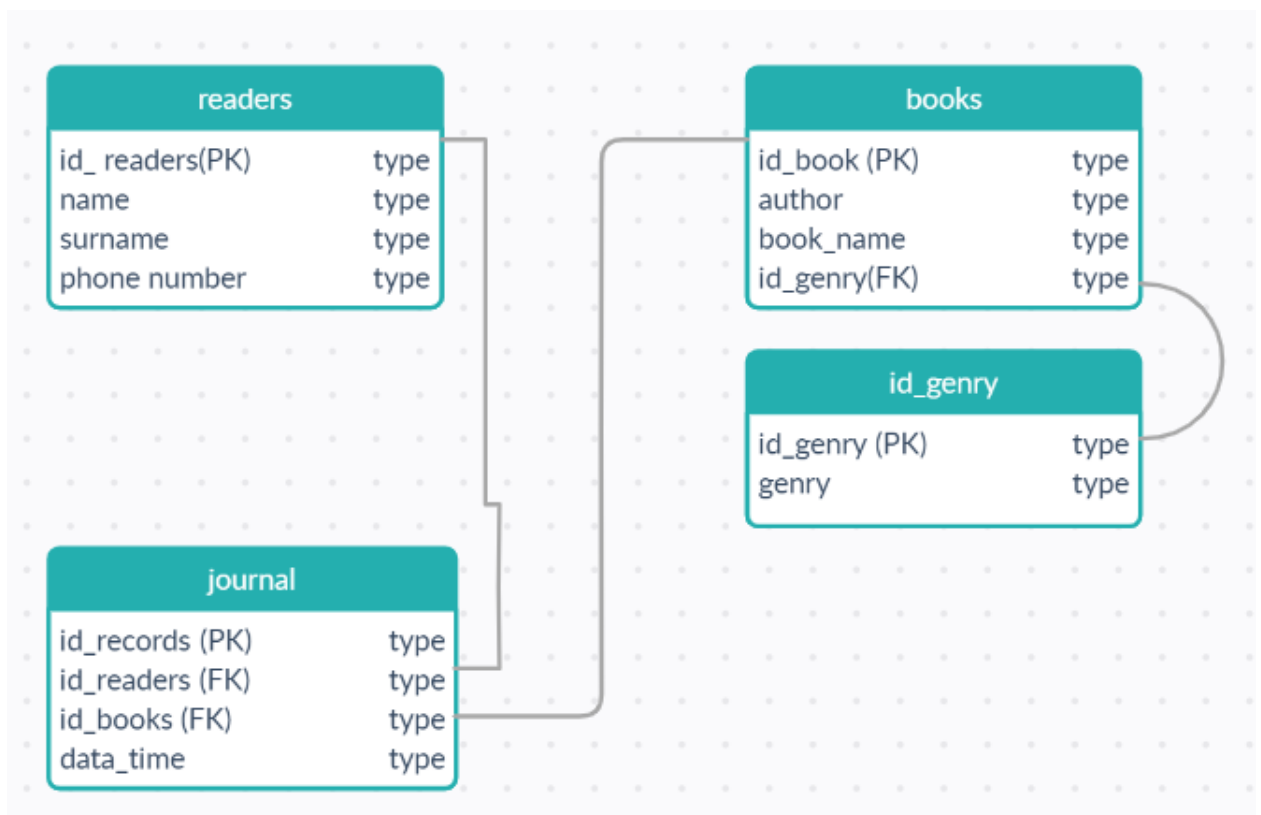
Київ – 2020

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

Загальне завдання роботи полягає у наступному:

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

Структура БД:



Реалізація меню :

```
"Hello,  
select a menu item and enter the number:"  
  
        1 - Task_1  
        2 - Tast_2  
        3 - Task_3  
        4 - exit  
  
Input:  
1  
Select number  
        1 - Insert  
        2 - DELL  
        3 - Update  
        4 - back
```

1 завдання

функції внесення, редагування та вилучення даних у таблицях бази даних

Insert:

```
'name_table' : [coloms]  
  
"books" : [ 'author','name_book','genry'],  
"journal" :['id_readers','id_books','data_time'],  
"id_genry" :[ 'genry'],  
"readers" : ['name','surname','phone_number']  
  
please, input for example:  
[name_table new_value new_value new_value]  
books      Mark_tven 15_capitan 12  
  
journal 4 4 2012-12-12 09:00+2
```

Результат	План выполнения		Сообщения		Notifications	
	id_records [PK] integer	id_readers integer	id_books integer	data_time timestamp with time zone		
1	1	3	3	2021-12-14 17:29:32.048951+02		
2	2	2	2	2021-06-04 21:37:13.831819+03		
3	3	2	1	2020-04-09 04:23:11.874735+03		
4	4	2	4	2023-03-27 02:45:35.841565+03		
5	5	3	1	2021-04-16 06:54:10.273103+03		
6	6	3	4	2023-12-12 15:24:37.657673+02		
7	7	2	4	2020-04-12 20:51:55.476962+03		
8	9	4	4	2022-06-04 01:49:23.439143+03		
9	10	5	4	2021-07-21 12:59:45.505638+03		
10	12	2	2	2020-10-05 00:10:00+03		
11	14	2	1	2020-10-04 23:10:00+03		
12	15	3	4	2020-04-05 00:00:00+03		
13	16	3	4	2020-04-05 00:00:00+03		
14	18	3	4	2018-04-05 00:00:00+03		
15	22	3	2	2016-07-03 14:22:00+03		
16	25	4	4	2012-12-12 09:00:00+02		
17	26	4	4	2012-12-12 09:00:00+02		

Таблица books

```
'name_table' : [coloms]

"books" : [ 'author','name_book','genry'],
"journal" : ['id_readers','id_books','data_time'],
"id_genry" :[ 'genry'],
"readers" : ['name','surname','phone_number']

please, input for example:
[name_table new_value new_value new_value]
books    Mark_tven 15_capitan 12

books FRANKO 5 21
"Hello,
select a menu item and enter the number:"
```

Результат	План выполнения		Сообщения		Notifications	
	 id_boooks [PK] integer	 author character varying	 name_book character varying	 genry integer		
1	1	FT	GJ		1	
2	2	NT	UX		3	
3	3	KL	VC		5	
4	4	Mark_tven	15_capitan		12	
5	5	FRANKO	5		21	

Вставка в таблицу readers

```
1

'name_table' : [coloms]

"books" : [ 'author','name_book','genry'],
"journal" :['id_readers','id_books','data_time'],
"id_genry" :[ 'genry'],
"readers" : ['name','surname','phone_number']

please, input for example:
[name_table new_value new_value new_value]
books    Mark_tven 15_capitan 12

readers test testing 123321
"Hello,
```

Результат	План выполнения		Сообщения	Notifications	
	<div><div>id_readers</div><div>[PK] integer</div></div>	<div><div>name</div><div>character varying</div></div>	<div><div>surname</div><div>character varying</div></div>	<div><div>phone_number</div><div>integer</div></div>	
1	10007	test	testing	123321	
2	10006	CHH	EL^	56825900	
3	10005	DIT	CIP	43174917	

```
"readers" : ['name','surname','phone_number']
```

please, input for example:

```
[name_table new_value new_value new_value]
```

```
books      Mark_tven 15_capitan 12
```

```
id_genry testtingtest
```

```
"Hello,
```

```
1 SELECT * FROM public.id_genry
2 ORDER BY id_genry ASC
```

	id_genry [PK] integer	genry character varying
10	10	D
11	11	T
12	12	V
13	13	T
14	14	S
15	15	E
16	16	V
17	17	O
18	18	T
19	19	G
20	20	X
21	21	B
22	22	S
23	23	t
24	24	t
25	25	test
26	26	testtingtest

DELL

```
1 SELECT * FROM public.books
2 ORDER BY id_boooks ASC
```

	id_boooks [PK] integer	author character varying	name_book character varying	genry integer
1	1	FT	GJ	1
2	2	NT	UX	3
3	3	KL	VC	5
4	4	Mark_tven	15_capitan	12
5	5	FRANKO	5	21

Після

Query Editor	История запросов	Результат	План выполнения	Сообщения	Notifications
1	SELECT * FROM public.books	id_boooks [PK] integer	author character varying	name_book character varying	genry integer
2	ORDER BY id_boooks ASC	1	1 FT	GJ	1
		2	2 NT	UX	3
		3	4 Mark_tven	15_capitan	12
		4	5 FRANKO	5	21

Select number

- 1 - Insert
- 2 - DELL
- 3 - Update
- 4 - back

2

```
"books" : 'id_books',
        "journal" : 'id_records' ,
        "id_genry" : 'id_genry',
        "readers" : 'id_readers'
```

please, input for example:

books 99

id_genry 16

"Hello,

select a menu item and enter the number:"

Результат	План выполнения	Сообщен
	id_genry [PK] integer	genry character varying
1	1	Y
2	2	H
3	3	J
4	4	L
5	5	K
6	6	K
7	7	F
8	8	V
9	9	J
10	10	D
11	11	T
12	12	V
13	13	T
14	14	S
15	15	E
16	17	O
17	18	T
18	19	G

Update

1	SELECT * FROM public.readers	id_readers [PK] integer	name "char" (1)	surname "char" (1)	phone_number integer
2	ORDER BY id_readers ASC	1	I	T	46
		2	W	Y	0
		3	A	V	9999999
		4	C	W	50099050
		5	v	t	111111
		6	t	t	321321

Select number

- 1 - Insert
- 2 - DELL
- 3 - Update
- 4 - back

3

```
'name_table' : [coloms]
```

```
"books" : [ 'author','name_book','genry'],
```

```
"journal" :['id_readers','id_books','data_time'],
```





```
"id_genry" :[ 'genry'],
```

```
"readers" : ['name','surname','phone_number']
```

please, input for example:

name_table	set_columns	data	id
id_genry	genry	літопис	87

readers name kolya surname port phone_number 333333 2

				
	id_readers [PK] integer	name character varyin	surname character varyin	phone_number integer
1	1	I	T	46
2	2	kolya	port	333333
3	3	test	debag	11111
4	4	C	W	50099050
5	5	v	t	111111
6	6	t	t	321321

Як реагує програма на невірні данні

Якщо введено не вірні данні виводить слово 'error' та повертає в головне меню або 'input correct data or 4-exit ' і очікує на введення даних або також повертає на головне меню

В меню Task_1 та його підпунктах

```
'name_table' : [coloms]

"books" : [ 'author','name_book','genry'],
"journal" :['id_readers','id_books','data_time'],
"id_genry" :[ 'genry'],
"readers" : ['name','surname','phone_number']

please, input for example:
[name_table new_value new_value new_value]
books      Mark_tven 15_capitan 12

1
input correct data
error
"Hello,
select a menu item and enter the number:"
```

```
"Hello,
select a menu item and enter the number:"

          1 - Task_1
          2 - Tast_2
          3 - Task_3
          4 - exit

Input:
test
input correct number
654
input correct number
456
input correct number
sdf dsfsd s sd
input correct number
,
```

```
Select number
    1 - Insert
    2 - DELL
    3 - Update
    4 - back
```

```
5
input correct number
654
input correct number
dfbs4
input correct number
|
```

```
2
books" : 'id_books',
        "journal" : 'id_records' ,
        "id_genry" : 'id_genry',
        "readers" : 'id_readers'
please, input for example:
books 99

test test
input correct data or 4-exit
test
input correct data or 4-exit
journal test
input correct data or 4-exit
journal 100
"Hello,
select a menu item and enter the number:"
```

3

```
'name_table' : [coloms]
```

```
"books" : [ 'author','name_book','genry'],
```

```
"journal" :['id_readers','id_books','data_time'],
```

```
"id_genry" :[ 'genry'],
```

```
"readers" : ['name','surname','phone_number']
```

please, input for example:

name_table	set_columns	data	id
------------	-------------	------	----

id_genry	genry	літопис	87
----------	-------	---------	----

reders name test t

error

```
'name_table' : [coloms]
```

```
"books" : [ 'author','name_book','genry'],
```

```
"journal" :['id_readers','id_books','data_time'],
```

```
"id_genry" :[ 'genry'],
```

```
"readers" : ['name','surname','phone_number']
```

please, input for example:

name_table	set_columns	data	id
------------	-------------	------	----

id_genry	genry	літопис	87
----------	-------	---------	----

1

input correct data or 4-exit

4

"Hello,

select a menu item and enter the number:"

Task_2 при введенні не вірних даних

```
about: data generation
      1 - generation of n rows for a specific table
      2 - generation of 10,000 rows for two tables to choose from
1
books
journal
id_genry
readers

      choose table and enter the number of new lines
      example : N table_name
                7 id_genry

test test
input correct data
"Hello,
select a menu item and enter the number:"

      1 - Task_1
      2 - Tast_2
      3 - Task_3
      4 - exit

Input:
```

```
2
about: data generation
      1 - generation of n rows for a specific table
      2 - generation of 10,000 rows for two tables to choose from
}
books
journal
id_genry
readers

      choose table and enter the number of new lines
      example : N table_name
                7 id_genry

+5 test
input correct data
"Hello,
select a menu item and enter the number:"

      1 - Task_1
      2 - Tast_2
```

Task_3 якщо в 3 запиті ввести не існуючий жанр результат буде пустий рядок.

```
input correct data
"Hello,
select a menu item and enter the number:"

1 - Task_1
2 - Tast_2
3 - Task_3
4 - exit

Input:
3
about request:"
1.displays a table with the reader's name, which book he took and its
2.displays the name of the reader who is reading the book that was tak
3.displays the names of readers who have taken books of a certain genre

1 - First request
2 - Second request
3 - Third request
4 - exit

3
displays the names of readers who have taken books of a certain genre
Example: Y

test
time 13
result :
```

```

Connecting to the PostgreSQL database...
"Hello,
select a menu item and enter the number:"

        1 - Task_1
        2 - Tast_2
        3 - Task_3
        4 - exit

Input:
2
about: data generation
        1 - generation of n rows for a specific table
        2 - generation of 10,000 rows for two tables to choose from
2
choose table
        1 - id_genry
        2 - readers
1
"Hello,
select a menu item and enter the number:"

        1 - Task_1
        2 - Tast_2
        3 - Task_3
        4 - exit

```

Виведено останні 100 рядків таблиці

1	SELECT * FROM public.id_genry		id_genry [PK] integer	genry character varying
2	ORDER BY id_genry DESC LIMIT 1	84	19943	YVRRM
3		85	19942	RRMCP
		86	19941	RQPPP
		87	19940	IOECB
		88	19939	GMMUB
		89	19938	TIFOE
		90	19937	SHNDK
		91	19936	GIKMN
		92	19935	UNCYX
		93	19934	VQFEB
		94	19933	PPMEN
		95	19932	IWOYF
		96	19931	WJUWA
		97	19930	HPSWW
		98	19929	KXLVW
		99	19928	PEAIW
		100	19927	DOPJL

Очистив таблицю id_genry

Query Editor	История запросов	Результат	План выполнения	Сообщения	Notifications
1 SELECT * FROM public.id_genry		id_genry	genry		
2 ORDER BY id_genry ASC		[PK] integer	character varying		

2 завдання

Запит для генерації 10000 даних

```
sql = 'INSERT INTO id_genry (genry) 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) limit 1 '
```

```
sql = "INSERT INTO readers(name,surname,phone_number) 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()*15)::int) || chr(trunc(65 + random()*30)::int)," \
      "trunc(random()*100000000)::int limit 1"
```

Для генерації конкретної кількості

```
"books" : ""INSERT INTO books (author,name_book,genry) select
          chr(trunc(65 + random()*26)::int)||chr(trunc(65 + random()*26)::int),
          chr(trunc(65 + random()*26)::int)||chr(trunc(65 + random()*26)::int),
          id_genry FROM id_genry TABLESAMPLE bernoulli(45) limit 1""
```

```
" INSERT INTO journal (data_time , id_readers, id_books)"
"select timestamp '2020-01-10 20:00:00' + random() * "
"(timestamp '2014-01-20 20:00:00' -timestamp '2010-01-10 10:00:00'),"
" id_readers, id_boooks from readers TABLESAMPLE bernoulli(33), books TABLESAMPLE bernoulli(33) limit 1 "
```

```
"INSERT INTO id_genry (genry) select chr(trunc(65 + random()*25)::int)|| chr(trunc(65 + random()*25)::int) "
"|| chr(trunc(65 + random()*25)::int),"
```

```
"INSERT INTO readers(name,surname,phone_number) 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()*15)::int) || chr(trunc(65 + random()*30)::int),"
"trunc(random()*100000000)::int",
```

а самі запити виконуються в циклі

```
while i < count:
    cur.execute(sql)
    i +=1
```

Заповнена таблиця за допомогою 2 підпункту 2 завдання в данної роботи

Перші 100 рядків та останні 100 рядків таблиці

Query Editor	История запросов	Результат	План выполнения	Сообщения
1	SELECT * FROM public.id_genry		id_genry [PK] integer	genry character varying
2	ORDER BY id_genry ASC LIMIT 10	1	20027	SEOLD
3		2	20028	VKCNL
		3	20029	SVEBP
		4	20030	LGIGV
		5	20031	EBGGB
		6	20032	HTGBS
		7	20033	PGRJA
		8	20034	MXFFF
		9	20035	MFMMB
		10	20036	WJGQY
		11	20037	ETXSX

Query Editor	История запросов	Результат	План выполнения	Сообщения
1	SELECT * FROM public.id_genry		id_genry [PK] integer	genry character varying
2	ORDER BY id_genry DESC LIMIT 100	83	29944	LOGLP
3		84	29943	RILJA
		85	29942	YHFWU
		86	29941	JFGYS
		87	29940	VXWVX
		88	29939	LMIXW
		89	29938	CCMBX
		90	29937	GUTAH
		91	29936	KRGJV
		92	29935	GNQSH
		93	29934	QIWUL
		94	29933	KVEHC
		95	29932	DDYLK
		96	29931	JIYOF
		97	29930	ROLXR
		98	29929	DNMKD
		99	29928	GSNOM
		100	29927	BHKPI

Заповнена таблиця за допомогою 2 підпункту 2 завдання в данної роботи

Перші 100 рядків та останні 100 рядків таблиці readers

Query Editor	История запросов	Результат	План выполнения	Сообщения	Notifications
1	<code>SELECT * FROM public.readers</code>	id_readers [PK] integer	name character varying	surname character varying	phone_number integer
2	<code>ORDER BY id_readers ASC</code>	1	1	T	46
		2	2 kolya	port	333333
		3	3 test	debag	11111
		4	4 C	W	50099050
		5	5 v	t	111111
		6	6 t	t	321321

Connecting to the PostgreSQL database...

"Hello,
select a menu item and enter the number:"

- 1 - Task_1
- 2 - Tast_2
- 3 - Task_3
- 4 - exit

Input:

2

about: data generation

- 1 - generation of n rows for a specific table
- 2 - generation of 10,000 rows for two tables to choose from

2

choose table

- 1 - id_genry
- 2 - readers

2

"Hello,
select a menu item and enter the number:"

- 1 - Task_1

1	<code>SELECT * FROM public.readers</code>	id_readers [PK] integer	name character varying	surname character varying	phone_number integer
2	<code>ORDER BY id_readers ASC</code>	9990	9990 VFF	QMW	28654848
		9991	9991 CWV	LE^	91073328
		9992	9992 BWN	IAK	14672901
		9993	9993 FAY	NC^	54683555
		9994	9994 THL	BOO	37551843
		9995	9995 XHM	VDB	30080087
		9996	9996 RQB	EOY	62289816
		9997	9997 EIW	WKA	4563733
		9998	9998 IPO	BMW	80974386
		9999	9999 OWK	NDJ	6547158
		10000	10000 YIT	UEQ	25876078
		10001	10001 HSF	MAY	3246611
		10002	10002 DOP	GFS	16992328
		10003	10003 OMD	IOJ	62764549
		10004	10004 AIU	MOP	72014473
		10005	10005 DIT	CIP	43174917
		10006	10006 CHH	EL^	56825900

Заповнення за допомогою генерації даних інших таблиць

```
about: data generation
    1 - generation of n rows for a specific table
    2 - generation of 10,000 rows for two tables to choose from

1
books
journal
id_genry
readers

choose table and enter the number of new lines
example : N table_name
          7 id_genry

25 journal
"Hello,
select a menu item and enter the number:"
```

Query Editor		История запросов		Результат	План выполнения	Сообщения	Notifications
1	SELECT	*	FROM public.books				
2	ORDER BY	id_boooks	ASC				
				id_boooks [PK] integer	author character varying	name_book character varying	genry integer
				9	324	VF	FL
				10	325	MO	OV
				11	326	IM	ER
				12	327	DU	GW
				13	328	ON	NX
				14	329	LE	UX
				15	330	SV	JG
				16	331	NE	YO
				17	332	AR	JM
				18	333	QR	II
				19	334	WB	OP
				20	335	TT	MH
				21	336	VQ	VJ
				22	337	CD	QS
				23	338	IL	QE
				24	339	KQ	VO
				25	340	AZ	CU

```
4 - exit
Input:
2
about: data generation
      1 - generation of n rows for a specific table
      2 - generation of 10,000 rows for two tables to choose from
1
books
journal
id_genry
readers

      choose table and enter the number of new lines
      example : N table_name
                7 id_genry

30 journal
"Hello,
select a menu item and enter the number:"
```

Query Editor	История запросов	Результат	План выполнения	Сообщения	Notifications
1	SELECT * FROM public.journal	<div><div></div><div>id_records</div><div>[PK] integer</div></div>	<div><div></div><div>id_readers</div><div>integer</div></div>	<div><div></div><div>id_books</div><div>integer</div></div>	<div><div></div><div>data_time</div><div>timestamp with time zone</div></div>
2	ORDER BY id_records ASC				
		14	54	1	321
		15	55	6	321
		16	56	4	318
		17	57	2	316
		18	58	6	317
		19	59	10	321
		20	60	1	317
		21	61	3	316
		22	62	4	317
		23	63	5	320
		24	64	4	316
		25	65	11	320
		26	66	5	317
		27	67	1	316
		28	68	1	317
		29	69	1	317
		30	70	4	317

3 завдання

1 – запит Виводить на екран таблицю ім'я , книга яку читають та її жанр.

```
sql = f'''select j.id_records , r.name , b.name_book ,g.genry
from public.journal j inner join public.readers r on j.id_readers = r.id_readers
inner join public.books b on j.id_books = b.id_boooks
inner join public.id_genry g on b.genry = g.id_genry'''
```

```
select a menu item and enter the number:"
```

- 1 - Task_1
- 2 - Tast_2
- 3 - Task_3
- 4 - exit

```
Input:
```

```
3
```

```
about request:"
```

- 1.displays a table with the reader's name, which book he took and its genre
- 2.displays the genre of the book that was taken the oldest
- 3.displays the names of readers who have taken books of a certain genre

- 1 - First request
- 2 - Second request
- 3 - Third request
- 4 - exit

```
1
```

```
time 20
```

```
result :
```

```
(41, 'v', 'JM', 'VKCNL')  
(42, 'I', 'MS', 'SEOLD')  
(43, 't', 'MS', 'SEOLD')  
(44, 'I', 'OH', 'VKCNL')  
(45, 't', 'MS', 'SEOLD')  
(46, 't', 'MS', 'SEOLD')  
(47, 't', 'EH', 'PGRJA')
```

```
(45, 't', 'MS', 'SEOLD')
(46, 't', 'MS', 'SEOLD')
(47, 't', 'EH', 'PGRJA')
(48, 'I', 'JM', 'VKCNL')
(49, 't', 'OH', 'VKCNL')
(50, 't', 'NH', 'SEOLD')
(51, 'test', 'MS', 'SEOLD')
(52, 'kolya', 'MS', 'SEOLD')
(53, 'C', 'UT', 'SEOLD')
(54, 'I', 'UT', 'SEOLD')
(55, 't', 'UT', 'SEOLD')
(56, 'C', 'JM', 'VKCNL')
(57, 'kolya', 'OH', 'VKCNL')
(58, 't', 'MS', 'SEOLD')
(59, 'MQS', 'UT', 'SEOLD')
(60, 'I', 'MS', 'SEOLD')
(61, 'test', 'OH', 'VKCNL')
(62, 'C', 'MS', 'SEOLD')
(63, 'v', 'NH', 'SEOLD')
(64, 'C', 'OH', 'VKCNL')
(65, 'TJL', 'NH', 'SEOLD')
(66, 'v', 'MS', 'SEOLD')
(67, 'I', 'OH', 'VKCNL')
(68, 'I', 'MS', 'SEOLD')
(69, 'I', 'MS', 'SEOLD')
(70, 'C', 'MS', 'SEOLD')
```

2 запит виводить ім'я того хто брав найдавнішу книгу по записам

```
Second_request(conn, cur):
sql = f'''select public.readers.name from public.readers where public.readers.id_readers =(
        (select public.journal.id_readers from public.journal where public.journal.data_time in
        (select MIN (public.journal.data_time) from public.journal )))'''
```

```

Hello,
select a menu item and enter the number:"

        1 - Task_1
        2 - Tast_2
        3 - Task_3
        4 - exit

Input:
input correct number
3
about request:"
    1.displays a table with the reader's name, which book he took and its genre
    2.displays the name of the reader who is reading the book that was taken the oldest
    3.displays the names of readers who have taken books of a certain genre

        1 - First request
        2 - Second request
        3 - Third request
        4 - exit
3
displays the name of the reader who is reading the book that was taken the oldest
time 2
result : [('test',)]

```

3-й запит виводить ім'я читачі які читають книгу з одним жанром

```

sql = f'''select public.readers.name from public.readers where public.readers.id_readers in (
    (select public.journal.id_readers from public.journal where public.journal.id_books in
    (select public.books.id_bbooks from public.books where public.books.genry in
    (select public.id_genry.id_genry from public.id_genry where public.id_genry.genry LIKE '{genry}'))))'''

about request:"
    1.displays a table with the reader's name, which book he took and its genre
    2.displays the name of the reader who is reading the book that was taken the oldest
    3.displays the names of readers who have taken books of a certain genre

        1 - First request
        2 - Second request
        3 - Third request
        4 - exit
3
displays the names of readers who have taken books of a certain genre
Example: Y
SEOLD
time 3
result :
('TJL',)
('MQS',)
('I',)
('v',)
('C',)
('kolya',)
('t',)
('test',)

```

Вихід із програми

```
"Hello,  
select a menu item and enter the number:"
```

- 1 - Task_1
- 2 - Tast_2
- 3 - Task_3
- 4 - exit

```
Input:
```

```
4
```

```
Database connection closed.
```

```
Process finished with exit code 0
```


4. Програмни код в репозиторії

Посилання на Github : https://github.com/holubvova/DATA_BASE/tree/master/LAB2

master	DATA_BASE / LAB2 /	Go to file	Add file
holubvova	Delete cxeмa.jpg	43c2887	in 2 hours History
..			
DATA.py	Add files via upload		now
controller.py	Add files via upload		now
main.py	Add files via upload		9 days ago
model.py	Add files via upload		now
photo_2020-11-18_10-14-50.jpg	Add files via upload		now
view.py	Add files via upload		now


master DATA_BASE / LAB2 / model.py / <> Jump to

holubvova Add files via upload

1 contributor

240 lines (213 sloc) | 7.51 KB

```
1 import DATA
2 import time
3 from psycpg2 import errors
4
5
6 def DELL_REQUEST(name_table):
7     colum = ''
8     for x in DATA.data_DELL:
9         if x == name_table:
10             colum = DATA.data_DELL[x]
11             break
```

 master ▼

DATA_BASE / LAB2 / controller.py / <> Jump to ▾



[holubvova](#) Add files via upload


1 contributor

74 lines (70 sloc) | 2.25 KB

```

1  import os
2  import view
3  import model
4  import DATA
5
6  def menu(conn, cur):
7
8      while 1:
9          number = view.Menu()
10         if number == '1':
11             mod, table_name, columns = view.task_1()
12             if mod == 1:
13                 model.Insert(conn, cur, table_name, columns)
14                 continue
15             elif mod == 2:
16                 model.DELL(conn, cur, table_name, columns)
17                 continue
18             elif mod == 3:


```

 master ▾

[DATA_BASE](#) / [LAB2](#) / [view.py](#) / <> Jump to ▾




holubvova Add files via upload

 1 contributor

172 lines (161 sloc) | 5.92 KB


```
1  import DATA
2
3
4  menus = '''Hello,\nselect a menu item and enter the number:"
5
6          1 - Task_1
7          2 - Tast_2
8          3 - Task_3
9          4 - exit
10 Input:''
11
12 TASK1 = '''Select number
13         1 - Insert
14         2 - DELL
```

 master ▾

[DATA_BASE](#) / [LAB2](#) / [model.py](#) / <> Jump to ▾




holubvova Add files via upload

 1 contributor

240 lines (213 sloc) | 7.51 KB


```
1  import DATA
2  import time
3  from psycpg2 import errors
4
5
6  def DELL_REQUEST(name_table):
7      colum = ''
8      for x in DATA.data_DELL:
9          if x == name_table:
10             colum = DATA.data_DELL[x]
11             break
12     if colum == '':
13         return 'error'
```

 master ▾

[DATA_BASE](#) / [LAB2](#) / [main.py](#) / <> Jump to ▾




holubvova Add files via upload

 1 contributor

27 lines (20 sloc) | 698 Bytes

```
1  import psycopg2
2  import controller
3
4
5
6  def main():
7      try:
8          # connect to the PostgreSQL server
9          print('Connecting to the PostgreSQL database...')
10         conn = psycopg2.connect(database='library',
11                                 user="postgres",
12                                 password='1705',
13                                 host="localhost",
14                                 port="5432")
15
16         cur = conn.cursor()
17         controller.menu(conn, cur)
18
```

 master ▾

[DATA_BASE](#) / [LAB2](#) / [DATA.py](#) / <> Jump to ▾



holubvova Add files via upload

 1 contributor

34 lines (25 sloc) | 1.51 KB

```
1  data = {
2      "books" : [ 'author','name_book','genry'],
3      "journal" :['id_readers','id_books','data_time'],
4      "id_genry" :['genry'],
5      "readers" : ['name','surname','phone_number']
6  }
7
8  data_DELL = {
9      "books" : 'id_boooks',
10     "journal" : 'id_records' ,
11     "id_genry" : 'id_genry',
12     "readers" : 'id_readers'
13 }
14
```