

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-92

Рукавіцин К. В.

Перевірив: доц. Петрашенко А. В.

Київ – 2021

Мета роботи: здобуття практичних навичок проектування та побудови реляційних баз даних та створення прикладних програм з базами даних.

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 підлеглій таблиці. З точки зору підлеглої таблиці варто контролювати наявність відповідності 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 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!**
3. Для реалізації пошуку необхідно підготувати 3 запити, 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.

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

Інформація про модель та структуру бази даних

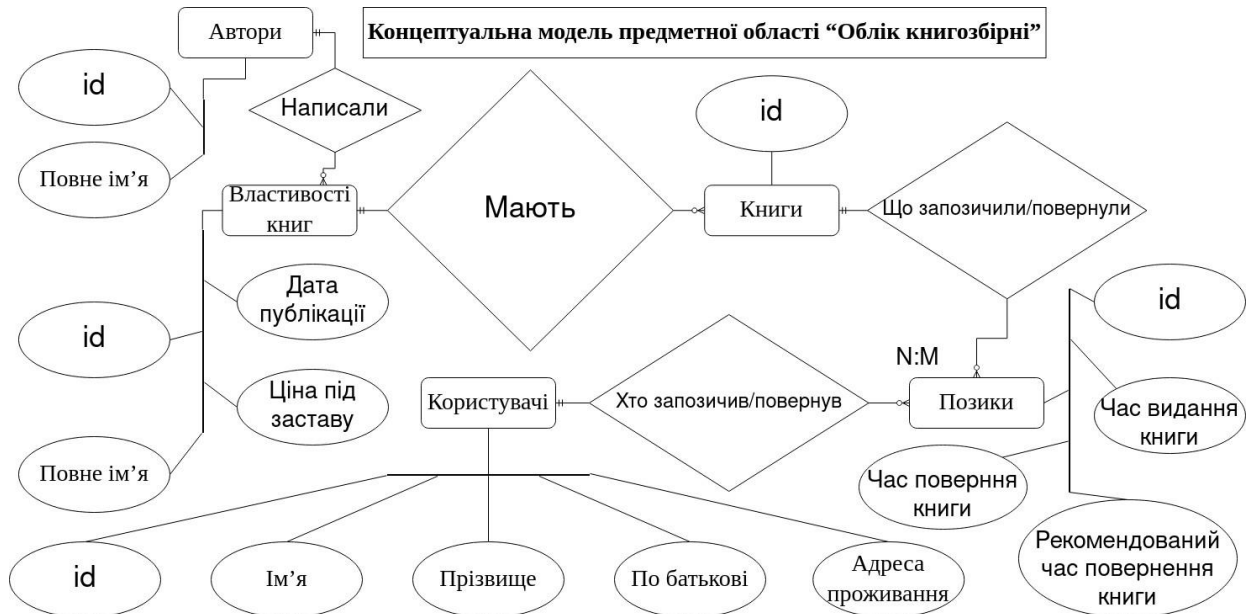


Рис. 1 - Концептуальна модель предметної області “Облік книгозбірні”

Нижче (Рис. 2) наведено логічну модель бази даних:

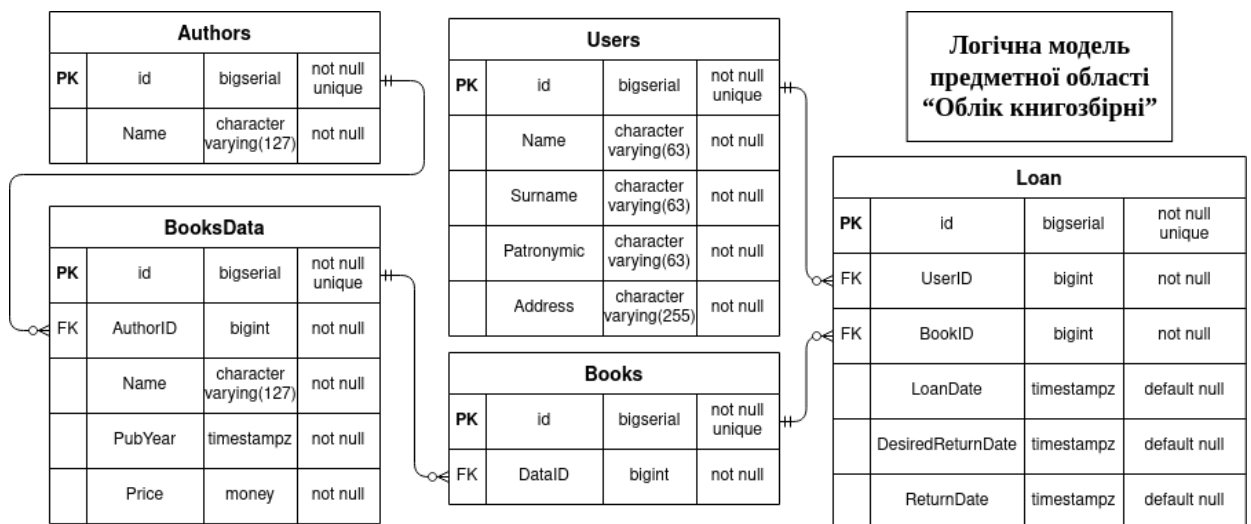


Рис. 2 – Логічна модель бази даних

Зміни у порівнянні з першою лабораторною роботою відсутні.

Середовище розробки та налаштування підключення до бази даних

Для виконання лабораторної роботи використовувалась мова програмування Python та текстовий редактор Sublime Text 3.

Для підключення до серверу бази даних PostgreSQL використано модуль «psycopg2».

Опис структури програми

Програма містить 5 основних модулів: **Lab**, **model**, **view**, **controller**, **utils**. Файл для запуску - «lab2.py3».

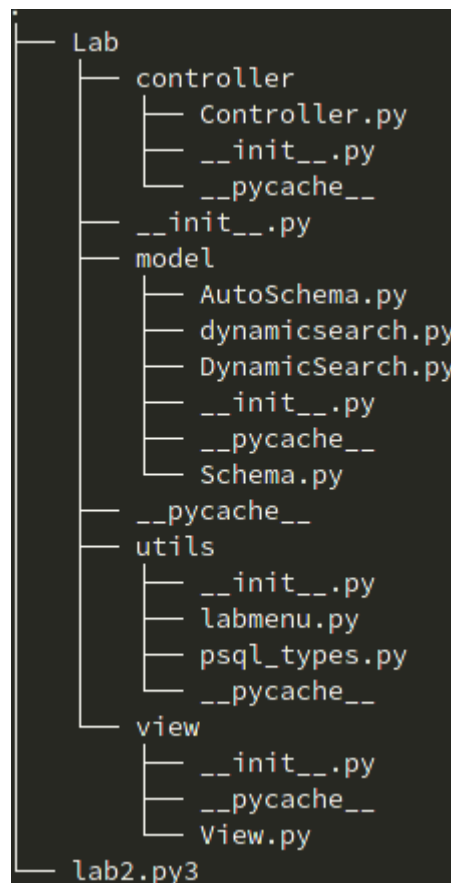


Рис. 3 – Структура програмного коду

Структура меню програми

Головне меню

```
MVC schema "Library" interface
> "Authors" table
  "BooksData" table
  "Books" table
  "Loan" table
  "Users" table
  Schema "Library" utils
  Dynamic search
  exit
```

Меню для таблиці

```
"Library"."Authors" table interface:
> describe
  show data
  add data
  edit data
  remove data
  random fill
  return
```

Меню для вибору динамічних запитів

```
Schema "Library" dynamic search interface
> Books
  UsersLoan
  return
```

```
Books dynamic search interface
"Author" ignored
"Name" ignored
"PubYear" ignored
"Price" ignored
  Author
  Name
  PubYear
  Price
  execute
  sql
> return
```

Меню вибору кількості рядків для генерації

```
instances [100]:
```

Пункт 1

Реалізувати функції внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі №1, засобами консольного інтерфейсу.

Внесення даних

Створення нового автора:

	id [PK] bigint	Name character varying (127)

```
"Library"."Authors" table interfaces:  
describe  
show data  
> add data  
edit data  
remove data  
random fill  
return
```

Name: AUTHOR01

	id [PK] bigint	Name character varying (127)
1	1	AUTHOR01

Видалення даних

	id [PK] bigint	Name character varying (127)
1	1	AUTHOR01
2	2	LOLOL

id: 2

	id [PK] bigint	Name character varying (127)
1	1	AUTHOR01

Якщо уведено неіснуючий id рядку:

```
id: 100
0 rows deleted
```

Неправильно введене число:

```
id: 500a
Error: 500a is not a valid integer
id: 
```

При видаленні рядку програма завжди використовує ключове слово “CASCADE” мови SQL. Ключове слово “CASCADE” дає дозвіл СУБД автоматично видаляти залежні рядки в дочірній таблиці, коли відповідні рядки видаляються в батьківській таблиці.

Редагування даних

```
id: 1
Name: renamed author
1 rows changed
```

	id [PK] bigint	Name character varying (127)
1	1	renamed author

Пункт 2

Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.

У програмі передбачено рандомізоване заповнення кожної таблиці окремо(з вказаннями кількості рандомізованих рядків для генерації), та пакетне рандомізоване заповнення таблиць схеми(без можливості зміни кількості рандомізованих рядків користувачем).

Рандомізоване заповнення таблиці “Users”:

	id [PK] bigint	Name character varying (63)	Surname character varying (63)	Patronymic character varying (63)	Address character varying (255)

```
"Library"."Users" table interface:  
describe  
show data  
add data  
edit data  
remove data  
> random fill  
return
```

```
instances [100]: 10  
"Library"."Users" 10 rows added, execution time: 0:00:00.064108
```

	id [PK] bigint	Name character varying (63)	Surname character varying (63)	Patronymic character varying (63)	Address character varying (255)
1	1	bnmQWERTYU	rtyuiopasd	wertyuiopa	yuiopasdfg
2	2	LZXCVBNM	wertyuiopa	GHJKLZXCVB	NM
3	3	TYUIOPASDF	bnmQWERTYU	jklzxcvbnm	RTYUIOPASD
4	4	cvbnmQWERT	XCVBNM	wertyuiopa	iopasdfghj
5	5	PASDFGHJKL	vbnmQWERTY	ertyuiopas	opasdfghjk
6	6	tyuiopasdf	dfghjklzxc	FGHJKLZXC	XCVBNM
7	7	VBNM	yuiopasdfg	XCVBNM	SDFGHJKLZX
8	8	RTYUIOPASD	bnmQWERTYU	JKLZXCVBNM	klzxcvbnmQ
9	9	ertyuiopas	nmQWERTYUI	ZXCVBNM	vbnmQWERTY
10	10	mQWERTYUIO	FGHJKLZXC	ghjklzxcvb	cvbnmQWERT

Пакетне заповнення таблиць схеми:

Кількість заданих рядків пакетного заповнення таблиць схеми:

Authors	5_000
Users	10_000
BooksData	1_000

Books	50_000
Loan	10_000

```
"Library"."Authors" 5000 rows added, execution time: 0:00:00.027960
"Library"."Users" 10000 rows added, execution time: 0:00:00.070914
"Library"."BooksData" 1000 rows added, execution time: 0:00:01.307833
"Library"."Books" 50000 rows added, execution time: 0:00:18.458428
"Library"."Loan" 10000 rows added, execution time: 0:02:12.333388
```

Витяги деяких рандомізованих рядків з таблиць:

Authors

	id [PK] bigint	Name character varying (127)
1	1	M
2	2	dfghjklzxc
3	3	NM
4	4	ASDFGHJKLZ
5	5	YUIOPASDFG
6	6	cvbnmQWERT
7	7	pasdfghjkl
8	8	HJKLZXCVBN
9	9	klzxcvbnmQ
10	10	ertyuiopas
11	11	sdfghjklzx
12	12	HJKLZXCVBN
13	13	DFGHJKLZXC
14	14	bnmQWERTYU
15	15	LZXCVBNM
16	16	uiopasdfgh
17	17	rtyuiopasd
18	18	yuiopasdfg
19	19	LZXCVBNM
20	20	FGHJKLZXCV
21	21	JKLZXCVBNM
22	22	ERTYUIOPAS
23	23	rtyuiopasd
24	24	ERTYUIOPAS

Users

Data Output Explain Messages Notifications						
	id [PK] bigint	Name character varying (63)	Surname character varying (63)	Patronymic character varying (63)	Address character varying (255)	
1	1	uiopasdfgh	ertyuiopas	ERTYUIOPAS	rtyuiopasd	
2	2	DFGHJKLZXC	XCVBNM	vbnmQWERTY	asdfghjklz	
3	3	NM	YUIOPASDFG	RTYUIOPASD	klzxcvbnmQ	
4	4	bnmQWERTYU	SDFGHJKLZX	mQWERTYUIO	GHJKLZXCVB	
5	5	FGHJKLZXCV	ERTYUIOPAS	FGHJKLZXCV	NM	
6	6	OPASDFGHJK	vbnmQWERTY	OPASDFGHJK	uiopasdfgh	
7	7		lzxcvbnmQW	ertyuiopas	zxcvbnmQWE	
8	8	OPASDFGHJK	XCVBNM	LZXCVCBNM	cvbnmQWERT	
9	9	QWERTYUIOP	tyuiopasdf	asdfghjklz	wertyuiopa	
10	10	bnmQWERTYU	TYUIOPASDF	CVBNM	cvbnmQWERT	
11	11	JKLZXCVBNM	tyuiopasdf	asdfghjklz	RTYUIOPASD	
12	12	wertyuiopa	HJKLZXCVBN	DFGHJKLZXC	NM	
13	13	TYUIOPASDF	nmQWERTYUI	uiopasdfgh	ERTYUIOPAS	
14	14	lzxcvbnmQW	asdfghjklz	lzxcvbnmQW	WERTYUIOPA	
15	15	CVBNM	ERTYUIOPAS	mQWERTYUIO	opasdfghjk	
16	16	uiopasdfgh	jklzxcvbnm	rtyuiopasd	jklzxcvbnm	
17	17	QWERTYUIOP	cvbnmQWERT	KLZXCVBNM	JKLZXCVBNM	
18	18	jklzxcvbnm	nmQWERTYUI	lzxcvbnmQW	RTYUIOPASD	
19	19	dfghjklzxc	wertyuiopa	M	SDFGHJKLZX	
20	20	jklzxcvbnm	ERTYUIOPAS	wertyuiopa	KLZXCVBNM	

BooksData

	id [PK] bigint	AuthorID bigint	Name character varying (127)	PubYear timestamp with time zone	Price money
1	1	4094	asdfghjklz	2021-09-12 10:08:12.522468+03	\$22.00
2	2	3291	rtyuiopasd	2021-03-22 13:22:01.187038+02	\$60.00
3	3	2030	wertyuiopa	2021-01-22 02:47:27.275092+02	\$80.00
4	4	2215	YUIOPASDFG	2021-04-05 05:10:11.836257+03	\$80.00
5	5	792	OPASDFGHJK	2021-01-22 15:51:56.813041+02	\$49.00
6	6	1625	vbnmQWERTY	2021-04-18 07:32:56.945309+03	\$59.00
7	7	4924	xcvbnmQWER	2021-10-21 00:59:08.744526+03	\$41.00
8	8	214	QWERTYUIOP	2021-07-15 11:28:18.613615+03	\$47.00
9	9	4561	XCVBNM	2021-02-10 22:55:26.354821+02	\$56.00
10	10	2899	CVBNM	2021-01-15 23:40:23.911476+02	\$59.00
11	11	2385	uiopasdfgh	2021-07-26 13:15:11.356294+03	\$54.00

Books

	Id [PK] bigint	DataID bigint
1	1	919
2	2	755
3	3	493
4	4	348
5	5	768
6	6	196
7	7	108
8	8	607
9	9	617
10	10	12
11	11	172
12	12	251

Loan

	Id [PK] bigint	UserID bigint	BookID bigint	LoanDate timestamp with time zone	ReturnDate timestamp with time zone	DesiredReturnDate timestamp with time zone
1	1	8546	19354	2020-04-23 01:30:09.035485+03	2021-09-01 08:03:49.168253+03	2021-05-30 05:31:10.194019+03
2	2	1391	23721	2020-08-29 08:02:50.483206+03	2021-06-20 12:52:27.872069+03	2021-07-06 22:43:44.433392+03
3	3	6200	22866	2020-08-23 03:01:04.783773+03	2021-10-18 20:03:49.188135+03	2021-03-09 01:08:11.72062+02
4	4	119	33536	2020-10-08 03:15:09.813634+03	2021-08-26 16:43:44.245709+03	2021-05-22 23:40:48.827276+03
5	5	1551	9122	2020-08-28 17:06:42.836119+03	2021-02-02 22:36:29.296523+02	2021-11-09 17:48:54.814784+02
6	6	3114	32675	2020-06-25 07:15:50.136338+03	2021-04-19 17:50:44.290886+03	2021-05-28 12:52:12.562018+03
7	7	8493	17768	2020-05-30 04:26:34.41568+03	2021-03-29 15:30:07.783105+03	2021-07-12 04:08:12.293844+03
8	8	4100	38014	2020-07-22 16:15:53.471933+03	2021-01-15 04:13:14.803647+02	2021-05-24 06:28:06.429087+03
9	9	232	32	2020-06-08 15:58:17.538011+03	2021-03-13 04:26:54.406226+02	2021-10-27 06:13:09.886635+03
10	10	7537	45641	2020-10-10 16:10:23.922472+03	2021-10-31 06:31:15.897339+02	2021-02-06 14:12:32.87437+02
11	11	289	30063	2020-11-10 00:08:17.650138+02	2021-07-28 09:58:33.592087+03	2021-09-02 05:23:06.359168+03
12	12	403	20527	2020-08-30 01:46:39.031926+03	2021-02-17 10:36:34.46964+02	2021-08-20 05:57:20.962819+03
13	13	5820	44681	2020-05-13 08:56:30.997105+03	2021-01-20 05:11:42.431602+02	2021-03-24 09:39:03.154497+02
14	14	2379	28135	2020-03-24 22:30:32.786255+02	2021-04-17 01:22:25.648806+03	2021-03-03 10:52:42.03529+02
15	15	8686	37547	2020-04-29 23:37:05.901249+03	2021-06-16 10:54:11.585381+03	2021-07-30 04:46:10.778941+03
16	16	6018	35215	2020-10-27 02:16:19.89207+02	2021-03-19 03:29:07.605699+02	2021-07-25 21:54:59.874391+03
17	17	8060	23144	2020-08-24 07:00:43.376384+03	2021-06-29 19:37:07.791765+03	2021-10-23 10:44:31.399298+03
18	18	7831	18230	2020-01-28 01:22:51.082767+02	2021-03-23 00:17:30.587162+02	2021-08-10 20:20:37.16681+03
19	19	9403	13174	2020-03-06 20:30:50.511725+02	2021-07-13 05:56:04.617692+03	2021-09-13 19:21:00.926722+03
20	20	700	42651	2020-06-09 00:14:54.015608+03	2021-04-24 20:24:50.658116+03	2021-09-18 16:08:34.494382+03
21	21	728	48972	2020-03-15 09:57:06.897202+02	2021-11-08 09:42:31.975278+02	2021-07-25 23:41:19.866478+03

SQL запити рандомізованого заповнення:

```
INSERT INTO "Library"."Authors"("Name")
SELECT
    substr(characters, (random() * length(characters) + 1)::integer, 10)
FROM
    (VALUES('qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM')) as
symbols(characters),
    generate_series(1, 5000) as q;
```

```
INSERT INTO "Library"."Users"("Name", "Surname", "Patronymic", "Address")
SELECT
    substr(characters, (random() * length(characters) + 1)::integer, 10),
    substr(characters, (random() * length(characters) + 1)::integer, 10),
    substr(characters, (random() * length(characters) + 1)::integer, 10),
    substr(characters, (random() * length(characters) + 1)::integer, 10)
FROM
    (VALUES('qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM')) as
symbols(characters),
    generate_series(1, 10000) as q;
```

```
INSERT INTO "Library"."BooksData"("AuthorID", "Name", "PubYear", "Price")
SELECT
    (SELECT "id" FROM "Library"."Authors" ORDER BY random()*q LIMIT 1),
    substr(characters, (random() * length(characters) + 1)::integer, 10),
    timestamp '2021-01-01' + random() * (timestamp '2021-11-11' - timestamp '2021-01-01'),
    trunc(random() * 100)::int
FROM
    (VALUES('qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM')) as
symbols(characters),
    generate_series(1, 1000) as q;
```

```
INSERT INTO "Library"."Books"("DataID")
SELECT
    (SELECT "id" FROM "Library"."BooksData" ORDER BY random()*q LIMIT 1)
FROM
    (VALUES('qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM')) as
symbols(characters),
    generate_series(1, 50000) as q;
```

```
INSERT INTO "Library"."Loan"("UserID", "BookID", "LoanDate", "ReturnDate", "DesiredReturnDate")
SELECT
  (SELECT "id" FROM "Library"."Users" ORDER BY random()*q LIMIT 1),
  (SELECT "id" FROM "Library"."Books" ORDER BY random()*q LIMIT 1),

  timestamp '2020-01-01' + random() * (timestamp '2020-11-11' - timestamp '2020-01-01'),
  timestamp '2021-01-01' + random() * (timestamp '2021-11-11' - timestamp '2021-01-01'),
  timestamp '2021-01-01' + random() * (timestamp '2021-11-11' - timestamp '2021-01-01')
FROM
  (VALUES('qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM')) as
symbols(characters),
generate_series(1, 10000) as q;
```

Пункт 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.

Було підготовлено два SQL запити:

- Пошук екземплярів книги за атрибутами:
 - Ім'я автора
 - Назва книги
 - Рік видання
 - Ціна
- Пошук користувачів та книг, що вони запозичили за наступними атрибутами:
 - ім'я користувача
 - прізвище
 - по батькові
 - адреса проживання
 - дата видачі книги
 - дата коли книгу було повернено
 - дата до якої потрібно повернути книгу
 - повна назва книги

Пошук екземплярів книги:
SQL запит без фільтрації рядків:

```
SELECT
  "c"."Name" as "Author",

  "b"."Name" as "Name",
  "b"."PubYear" as "PubYear",
  "b"."Price" as "Price"

FROM
  "Library"."Books" as "a"
  INNER JOIN "Library"."BooksData" as "b"
    ON "a"."DataID" = "b"."id"
  INNER JOIN "Library"."Authors" as "c"
    ON "b"."AuthorID" = "c"."id"
;
```

Результат:

```
tyuiopasdf | opasdfghjk | 2021-02-09 09:53:08.318427+02:00 | $81.00
cvbnmQWERT | xcvbnmQWER | 2021-09-10 22:43:23.631922+03:00 | $55.00
DFGHJKLZXC | fghjklzxcv | 2021-01-05 00:38:35.330403+02:00 | $95.00
UIOPASDFGH | SDFGHJKLZX | 2021-03-23 19:12:54.623567+02:00 | $56.00
rtyuiopasd | OPASDFGHJK | 2021-03-24 16:11:19.111693+02:00 | $7.00
ASDFGHJKLZ | nmQWERTYUI | 2021-05-31 09:57:43.267463+03:00 | $97.00
M           | JKLZXCVBNM | 2021-05-02 05:17:18.821782+03:00 | $33.00
HJKLZXCVPN | fghjklzxcv | 2021-07-19 22:23:29.477769+03:00 | $89.00
IOPASDFGHJ | lzxcvbnmQW | 2021-07-23 08:38:00.716763+03:00 | $13.00
M           | hjklzxcvbn | 2021-07-11 04:49:14.812950+03:00 | $12.00
ASDFGHJKLZ | wertyuiopa | 2021-01-18 13:55:32.226855+02:00 | $71.00
ASDFGHJKLZ | ghjklzxcvb | 2021-10-30 19:05:26.723982+03:00 | $92.00
XCVBNM     | DFGHJKLZXC | 2021-02-04 02:04:01.353438+02:00 | $49.00
HJKLZXCVPN | GHJKLZXCVB | 2021-01-09 05:13:43.429943+02:00 | $52.00
ERTYUIOPAS | GHJKLZXCVB | 2021-11-02 03:31:46.663438+02:00 | $19.00
tyuiopasdf | UIOPASDFGH | 2021-08-02 19:19:20.895582+03:00 | $23.00
ERTYUIOPAS | NM          | 2021-03-10 13:23:22.502111+02:00 | $50.00
RTYUIOPASD | opasdfghjk | 2021-08-29 18:57:19.624220+03:00 | $68.00
xcvbnmQWER | pasdfghjkl | 2021-04-09 04:03:00.883626+03:00 | $40.00
cvbnmQWERT | xcvbnmQWER | 2021-10-15 11:49:52.531632+03:00 | $25.00
xcvbnmQWER | opasdfghjk | 2021-06-13 05:31:02.314335+03:00 | $66.00
DFGHJKLZXC | FGHJKLZXC | 2021-04-03 15:12:59.029687+03:00 | $26.00
rtyuiopasd | OPASDFGHJK | 2021-03-24 16:11:19.111693+02:00 | $7.00
50000 rows, execution time: 0:00:00.070343
Books dynamic search interface
"Author" ignored
"Name" ignored
"PubYear" ignored
"Price" ignored
> Author
  Name
  PubYear
  Price
execute
sql
return
```


Налаштування фільтрування рядків:

- ім'я автора = 'BNM'
- ціна книги > 10.00
- ціна книги != 25.00
- ціна книги < 50.00
- дата видання книги >= 2021-06-10
- дата видання книги <= 2021-09-11

Введені критерії фільтрування:

```
Books dynamic search interface
"Author" = 'BNM'::varchar
"Name" ignored
"PubYear" >= '2021-06-10 00:00:00'::timestamp AND <= '2021-09-11 00:00:00'::timestamp
"Price" > 10.0::money AND != 25.0::money AND < 50.0::money
> Author
  Name
  PubYear
  Price
  execute
  sql
  return
```

SQL запит з заданими налаштуваннями фільтрування рядків:

SELECT

"c"."Name" **as** "Author",

"b"."Name" **as** "Name",

"b"."PubYear" **as** "PubYear",

"b"."Price" **as** "Price"

FROM

"Library"."Books" **as** "a"

INNER JOIN "Library"."BooksData" **as** "b"

ON "a"."DataID" = "b"."id"

INNER JOIN "Library"."Authors" **as** "c"

ON "b"."AuthorID" = "c"."id"

WHERE

("c"."Name" = 'BNM'::varchar) **AND**

("b"."PubYear" >= '2021-06-10 00:00:00'::timestamp **AND** "b"."PubYear" <= '2021-09-11

00:00:00'::timestamp) **AND**

("b"."Price" > 10.0::money **AND** "b"."Price" != 25.0::money **AND** "b"."Price" < 50.0::money);

Результат:

Author	Name	PubYear	Price
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00

...

BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	sdfghjklzx	2021-07-21 18:06:33.697131+03:00	\$48.00
BNM	pasdfghjkl	2021-08-01 12:42:06.839305+03:00	\$28.00

106 rows, execution time: 0:00:00.042448

Books dynamic search interface

"Author" = 'BNM':varchar

"Name" ignored

"PubYear" >= '2021-06-10 00:00:00':timestamp AND <= '2021-09-11 00:00:00':timestamp

"Price" > 10.0::money AND != 25.0::money AND < 50.0::money

> Author

Name

PubYear

Price

execute

sql

return

Пошук користувачів та книг, що вони запозичили:
SQL запит без фільтрації рядків:

```
SELECT
  "a"."Name" as "Name",
  "a"."Surname" as "Surname",
  "a"."Patronymic" as "Patronymic",
  "a"."Address" as "Address",

  "b"."LoanDate" as "LoanDate",
  "b"."ReturnDate" as "ReturnDate",
  "b"."DesiredReturnDate" as "DesiredReturnDate",

  "d"."Name" as "BookName"
```

```
FROM
  "Library"."Users" as "a"
  INNER JOIN "Library"."Loan" as "b"
    ON "a"."id" = "b"."UserID"
  INNER JOIN "Library"."Books" as "c"
    ON "b"."BookID" = "c"."id"
  INNER JOIN "Library"."BooksData" as "d"
    ON "c"."DataID" = "d"."id"
;
```

Результат:

cvbnnQWERT	zxcvbnmQWE	fghjklzxcv	QWERTYUIOP	2020-04-26 21:46:12.760038+03:00	2021-08-14 20:45:50.568974+03:00	2021-01-05 07:16:39.543230+02:00	cvbnnQWERT
klzxcvbnmQ	vbnmQWERTY	fghjklzxcv	M	2020-03-16 13:43:12.575927+02:00	2021-05-16 01:41:13.066842+03:00	2021-02-09 08:06:37.262128+02:00	FGHJKLZXCV
rtYuioPasd	VBNM	klzxcvbnmQ	klzxcvbnmQ	2020-08-01 05:32:45.120009+03:00	2021-02-27 09:41:02.963943+02:00	2021-09-12 01:37:45.017211+03:00	uiopasdFgh
PASDFGHJKL	KLZXCVBNN	tyuiopasdf	ZXCVBNN	2020-02-15 04:47:00.043735+02:00	2021-10-14 03:35:17.948040+03:00	2021-06-04 15:19:17.322407+03:00	dfghjklzxc
SDFGHJKLZX	pasdfghjkl	wertyuiopa	YUIOPASDFG	2020-01-04 11:40:42.023780+02:00	2021-08-23 16:42:30.897492+03:00	2021-06-16 09:22:55.366970+03:00	ASDFGHJKLZ
klzxcvbnmQ	TYUIOPASDF	VBNN	CVBNN	2020-07-23 07:22:05.443279+03:00	2021-10-16 14:11:01.808327+03:00	2021-09-25 15:22:20.546304+03:00	vbnmQWERTY
HJKLZXCVBN	SDFGHJKLZX	sdFghjklzx	klzxcvbnmQ	2020-06-14 03:25:20.130134+03:00	2021-08-22 05:52:43.679187+03:00	2021-08-03 06:50:31.167583+03:00	uiopasdFgh
pasdfghjkl	PASDFGHJKL	TYUIOPASDF	CVBNN	2020-07-01 23:35:30.983833+03:00	2021-05-24 00:53:45.060837+03:00	2021-10-15 02:55:53.106688+03:00	uiopasdFgh
qwertyuiop	zxcvbnmQWE	NM	bnnmQWERTYU	2020-02-25 06:46:28.900396+02:00	2021-02-18 19:49:44.198950+02:00	2021-07-16 04:12:04.050184+03:00	bnnmQWERTYU
	ertyuiopas	klzxcvbnmQ	LZXCVBNM	2020-09-17 01:57:53.711236+03:00	2021-01-26 08:44:24.521542+02:00	2021-09-07 00:28:03.949381+03:00	
ioPasdfghj	dfghjklzxc		sdFghjklzx	2020-01-01 21:23:14.413746+02:00	2021-03-08 08:44:15.002048+02:00	2021-10-03 01:14:02.888720+03:00	hjkLzxcvbn
cvbnnQWERT	KLZXCVBNN	HJKLZXCVBN	SDFGHJKLZX	2020-06-25 18:42:53.442079+03:00	2021-05-14 19:05:39.998133+03:00	2021-05-23 16:48:59.060474+03:00	cvbnnQWERT
ZXCVBNN	M	pasdfghjkl	NM	2020-10-11 18:14:13.902761+03:00	2021-02-15 00:22:42.446648+02:00	2021-06-28 01:21:41.201361+03:00	cvbnnQWERT
asdfghjklz	opasdFghjk	SDFGHJKLZX	vbnmQWERTY	2020-09-28 07:55:02.454393+03:00	2021-10-09 15:04:45.756129+03:00	2021-03-05 10:11:43.239599+02:00	BNM
BNM	lzxcvbnmQW	YUIOPASDFG	WERTYUIOPA	2020-10-01 13:42:22.373639+03:00	2021-08-27 22:52:11.970900+03:00	2021-08-07 07:28:34.408563+03:00	yuioPasdfg
CVBNN	rtYuioPasd	YUIOPASDFG	wertyuiopa	2020-05-01 18:02:54.842919+03:00	2021-06-20 21:36:56.319827+03:00	2021-02-20 15:18:45.976457+02:00	GHJKLZXCVB
xcvbnmQWER	klzxcvbnm	bnnmQWERTYU	rtYuioPasd	2020-01-12 18:10:15.650942+02:00	2021-01-22 20:58:34.168129+02:00	2021-07-06 18:44:01.722266+03:00	xcvbnmQWER
yuioPasdfg	M	M	XCVBNN	2020-07-27 02:12:34.660538+03:00	2021-02-15 20:19:56.567760+02:00	2021-06-20 17:15:04.940589+03:00	uiopasdFgh
hjkLzxcvbn	TYUIOPASDF	XCVBNN	zxcvbnmQWE	2020-09-19 14:28:32.507971+03:00	2021-07-27 13:59:47.851556+03:00	2021-06-08 21:28:09.660437+03:00	uiopasdFgh
NM	DFGHJKLZX	ioPasdfghj	yuioPasdfg	2020-05-29 14:06:20.430819+03:00	2021-10-14 11:44:56.102325+03:00	2021-10-23 00:28:36.759559+03:00	ghjklzxcvbn
wertyuiopa	fghjklzxcv	FGHJKLZXCV	cvbnnQWERT	2020-07-09 20:49:35.684514+03:00	2021-03-08 13:38:40.503955+02:00	2021-11-08 04:17:57.006959+02:00	ASDFGHJKLZ
FGHJKLZXCV	UIOPASDFGH	nmQWERTYUI	wertyuiopa	2020-01-30 04:32:32.343001+02:00	2021-09-04 01:19:38.325913+03:00	2021-07-11 20:07:25.611190+03:00	asdfghjklz
GHJKLZXCVB	KLZXCVBNN	bnnmQWERTYU	vbnmQWERTY	2020-10-03 14:09:15.212819+03:00	2021-07-18 18:29:51.533246+03:00	2021-08-16 16:14:50.005632+03:00	IOPASDFGHJ
tyuiopasdf	YUIOPASDFG	cvbnnQWERT	dfghjklzxc	2020-05-11 20:44:18.419817+03:00	2021-06-23 20:05:09.105565+03:00	2021-09-25 22:22:03.446596+03:00	BNM
wertyuiopa	CVBNN	DFGHJKLZX	yuioPasdfg	2020-11-05 02:16:25.430151+02:00	2021-04-14 18:56:13.655641+03:00	2021-10-04 06:38:25.336596+03:00	yuioPasdfg
XCVBNN	JKLZXCVBNM	DFGHJKLZX	sdFghjklzx	2020-04-28 08:54:07.353428+03:00	2021-07-27 22:44:44.999641+03:00	2021-09-23 22:37:05.570394+03:00	ertyuiopas
DFGHJKLZX	cvbnnQWERT	pasdfghjkl	lzxcvbnmQW	2020-05-11 01:39:35.510221+03:00	2021-01-10 14:42:25.666030+02:00	2021-08-01 20:21:18.620936+03:00	mQWERTYUIO
DFGHJKLZX	zxcvbnmQWER	PASDFGHJKL	fghjklzxcv	2020-08-03 07:22:39.249137+03:00	2021-05-05 11:53:20.141822+03:00	2021-01-08 09:11:54.649587+02:00	bnnmQWERTYU
GHJKLZXCVB	zxcvbnmQWE	cvbnnQWERT	uiopasdFgh	2020-04-08 05:23:01.015990+03:00	2021-08-11 10:03:10.806107+03:00	2021-01-01 14:11:24.153159+02:00	wertyuiopa
vbnmQWERTY	CVBNN	asdfghjklz	CVBNN	2020-01-27 23:40:18.180610+02:00	2021-03-06 03:31:52.078909+02:00	2021-10-29 03:26:27.468102+03:00	NM
mQWERTYUIO	xcvbnmQWER	asdfghjklz	uiopasdFgh	2020-11-03 23:10:31.832949+02:00	2021-07-09 13:46:22.324068+03:00	2021-10-16 15:54:08.142096+03:00	xcvbnmQWER
rtYuioPasd	KLZXCVBNN	FGHJKLZXCV	yuioPasdfg	2020-04-18 20:36:50.013496+03:00	2021-05-22 02:56:36.425900+03:00	2021-07-28 23:43:46.850944+03:00	

```
10000 rows, execution time: 0:00:00.046332
UserLoan dynamic search interface
"Name" ignored
"Surname" ignored
"Patronymic" ignored
"Address" ignored
"LoanDate" ignored
"ReturnDate" ignored
"DesiredReturnDate" ignored
"BookName" ignored
> Name
  Surname
Patronymic
Address
LoanDate
ReturnDate
DesiredReturnDate
BookName
execute
sql
return
```

Налаштування фільтрування рядків:

- прізвище користувача == 'NM'
- по батькові користувача LIKE 'P%'

SQL запит з заданими налаштуваннями фільтрування рядків:

SELECT

```
"a"."Name" as "Name",  
"a"."Surname" as "Surname",  
"a"."Patronymic" as "Patronymic",  
"a"."Address" as "Address",  
  
"b"."LoanDate" as "LoanDate",  
"b"."ReturnDate" as "ReturnDate",  
"b"."DesiredReturnDate" as "DesiredReturnDate",  
  
"d"."Name" as "BookName"
```

FROM

```
"Library"."Users" as "a"
```

INNER JOIN "Library"."Loan" as "b"

```
ON "a"."id" = "b"."UserID"
```

INNER JOIN "Library"."Books" as "c"

```
ON "b"."BookID" = "c"."id"
```

INNER JOIN "Library"."BooksData" as "d"

```
ON "c"."DataID" = "d"."id"
```

WHERE

```
("a"."Surname" = 'NM'::varchar) AND  
("a"."Patronymic" LIKE 'P%'::varchar);
```

Результат:

Name	Surname	Patronymic	Address	LoanDate	ReturnDate	DesiredReturnDate	BookName
FGHJKLZXCV	NM	PASDFGHJKL	LZXCVBNM	2020-02-19 06:55:35.181083+02:00	2021-07-16 09:02:36.329395+03:00	2021-07-24 23:34:05.488062+03:00	lzxvbnmQw

1 rows, execution time: 0:00:00.003926
UserLoan dynamic search interface
"Name" ignored
"Surname" = 'NM'::varchar
"Patronymic" LIKE 'P%'::varchar
"Address" ignored
"LoanDate" ignored
"ReturnDate" ignored
"DesiredReturnDate" ignored
"BookName" ignored
> Name
Surname
Patronymic
Address
LoanDate
ReturnDate
DesiredReturnDate
BookName
execute
sql
return

