

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Лабораторна робота №6
З дисципліни «Організація баз даних і знань»
На тему: «Виконання реляційних операцій реляційної алгебри засобами SQL»

Виконала:
Студентка групи КН-208
Гусаревич Я.В.
Викладач:
Мельникова Н.І.

Львів

2019

МЕТА

Розробити SQL запити для виконання операцій реляційної алгебри: проекції, селекції, натурального з'єднання, умовного з'єднання.

ХІД РОБОТИ

1. Для того, щоб перевірити справність виконання запиту на проекцію, додамо до таблиці department_of_migration_service новий кортеж, який повторюватиме значення town_or_village, street з іншого кортежу. Створюємо запит на проекцію, вибираючи лише поля town_or_village і street з таблиці department_of_migration_service, як показано на скрипті нижче.

```
USE
passport1
;

INSERT INTO department_of_migration_service VALUES(NULL,
'Lviv', 'Lviv', 'Slovatskoho', '10');
SELECT * FROM department_of_migration_service;
CREATE VIEW department AS SELECT DISTINCT town_or_village,
street FROM department_of_migration_service;
SELECT *FROM department;
```

На рисунках нижче результати виконання запитів.

	id	region	town_or_village	street	house_number
►	1	Lviv	Lviv	Chaykovskoho	18
	2	Lviv	Lviv	Slovatskoho	15
	3	Lviv	Truskavets	Mazepy	3
	4	Kyiv	Kyiv	Shevchenka	14
	5	Lviv	Lviv	Slovatskoho	10
*	NULL	NULL	NULL	NULL	NULL

Рис. 1. Таблиця department_of_migration_service після вставки нових даних

	town_or_village	street
►	Lviv	Chaykovskoho
	Lviv	Slovatskoho
	Truskavets	Mazepy
	Kyiv	Shevchenka

Рис. 2. Таблиця department – результат виконання запиту на проекцію

Як можемо бачити з рис. 2. кортеж, який містив повторювані дані, відобразився у таблиці лише один раз. Отже, запит виконано успішно.

2. Виконаємо запит на селекцію осіб, які проживають у Львові, як показано на скрипті нижче.

```
USE
passport1
;

SELECT * FROM person;
SELECT * FROM person WHERE region_registration='''Lviv''' AND
town_or_village_registration='''Lviv''';
```

	id	person_name	person_surname	person_patronymic	sex	date_of_birth	region_registration	town_or_village_registration
▶	1	'Yanina'	'Husarevych'	'Vadymivna'		1999-04-03	'Lviv'	'Lviv'
	2	'Anastasiia'	'Antonyk'	'Yuriivna'		1999-04-03	'Lviv'	'Lviv'
	3	'Yurii'	'Lisnyskyi'	'Dmytrovych'		1999-04-03	'Lviv'	'Truskavets'
	4	'Oleksandra'	'Nevyniak'	'Andriivna'		1999-04-03	'Kyiv'	'Kyiv'
	21	'Ihor'	'Darmoriz'	'Volodymyrovych'		2000-11-09	'Lviv'	'Lviv'
	22	'Olena'	'Pona'	'Andriivna'		2000-11-09	'Lviv'	'Truskavets'
	23	'Oleksandr'	'Nevyniak'	'Andriovych'		2000-11-09	'Kyiv'	'Kyiv'
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 3. Таблиця person

	id	person_name	person_surname	person_patronymic	sex	date_of_birth	region_registration	town_or_village_registration	street_registration
▶	1	'Yanina'	'Husarevych'	'Vadymivna'		1999-04-03	'Lviv'	'Lviv'	'Chaykovskoho'
	2	'Anastasiia'	'Antonyk'	'Yuriivna'		1999-04-03	'Lviv'	'Lviv'	'Stefanyka'
	21	'Ihor'	'Darmoriz'	'Volodymyrovych'		2000-11-09	'Lviv'	'Lviv'	'Bandery'
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 4. Результат виконання запиту на селекцію осіб, які проживають у Львові

Як бачимо з рис.3 і рис. 4., запит виконано успішно.

3. Створимо запит на виконання натурального об'єднання stuff та passport, як показано на скрипті нижче. Результат виконання цього запиту має показати голів міграційних служб та за якою адресою їх шукати.

```
USE
passport1
;

SELECT * FROM stuff;
SELECT * FROM department_of_migration_service;
SELECT stuff.id, stuff.head_name, stuff.head_surname,
stuff.head_patronymic,
department_of_migration_service.town_or_village,
department_of_migration_service.street
```

```
FROM stuff, department_of_migration_service
WHERE stuff.department_id =
department_of_migration_service.id;
```

	id	head_name	head_surname	head_patronymic	former_name	former_surname	former_patronymic	former_position	department_id
▶	1	'Petro'	'Petrov'	'Petrovych'	'Ivan'	'Ivanov'	'Ivanovych'	'leading specialist of Galician RV'	1
	2	'Dmytro'	'Kozak'	'Vadimovich'	'Mykola'	'Malets'	'Yuriiovych'	'leading specialist of Galician RV'	2
	3	'Maksym'	'Zaycev'	'Nazarovych'	'Konstantyn'	'Hryb'	'Viktorovych'	'leading specialist of Pryvokzalnyi RV'	3
	4	'Liubomyr'	'Otaman'	'Oleksiiiovych'	'Oleh'	'Komar'	'Matviiovych'	'leading specialist of Darnytskyi RV'	4
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 5. Таблиця stuff

	id	region	town_or_village	street	house_number
▶	1	Lviv	Lviv	Chaykovskoho	18
	2	Lviv	Lviv	Slovatskoho	15
	3	Lviv	Truskavets	Mazepy	3
	4	Kyiv	Kyiv	Shevchenka	14
	5	Lviv	Lviv	Slovatskoho	10
•	NULL	NULL	NULL	NULL	NULL

Рис. 6. Таблиця department_of_migration_service

	id	head_name	head_surname	head_patronymic	town_or_village	street
▶	1	'Petro'	'Petrov'	'Petrovych'	Lviv	Chaykovskoho
	2	'Dmytro'	'Kozak'	'Vadimovich'	Lviv	Slovatskoho
	3	'Maksym'	'Zaycev'	'Nazarovych'	Truskavets	Mazepy
	4	'Liubomyr'	'Otaman'	'Oleksiiiovych'	Kyiv	Shevchenka

Рис. 7. Результат виконання запиту на натуральне з'єднання

Як бачимо з рис. 7, запит виконано успішно.

4. Створимо запит на виконання умовного з'єднання person і passport, як показано на скрипті нижче. Додатковою умовою поставимо порівняння дати видачі паспорту з датою народження особи, яка його отримала, для виявлення суперечностей у базі даних.

```
USE
passport1
;

SELECT * FROM person;
SELECT * FROM passport;
SELECT person.id, person.person_name, person.person_surname,
person.person_patronymic,
passport.id, passport.record_number, passport.passport_number
FROM person, passport
WHERE passport.person_id = person.id AND
```

```
passport.date_of_issue < person.date_of_birth;
```

	id	person_name	person_surname	person_patronymic	sex	date_of_birth
▶	1	'Yanina'	'Husarevych'	'Vadymivna'		1999-04-03
	2	'Anastasiia'	'Antonyk'	'Yuriivna'		1999-04-03
	3	'Yurii'	'Lisnytskyi'	'Dmytrovych'		1999-04-03
	4	'Oleksandra'	'Nevyniak'	'Andriivna'		1999-04-03
	21	'Ihor'	'Darmoriz'	'Volodymyrovych'		2000-11-09
	22	'Olena'	'Pona'	'Andriivna'		2000-11-09
	23	'Oleksandr'	'Nevyniak'	'Andriovych'		2000-11-09
★	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 8. Таблиця person

	id	record_number	date_of_issue	date_of_expiry	country_code	price	type_of_passport	stuff_id	person_id	passport_number
▶	3	'19991119-02664'	2018-10-03	2028-10-03	'UK	0		2	3	'FP13628
	4	'20121215-02345'	2018-10-03	2028-10-03	'UK	700		1	4	'FP23433
	5	'20223524-64751'	2018-10-03	2028-10-03	'UK	0		3	2	'BA28321
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 9. Таблиця passport

	id	person_name	person_surname	person_patronymic	id	record_number	passport_number
--	----	-------------	----------------	-------------------	----	---------------	-----------------

Рис. 10. Результат виконання запиту на умовне з'єднання passport і person

Як бачимо на рис. 10, результатом виконання запиту на умовне з'єднання є порожня множина, а отже суперечностей у базі нема, і запит виконано успішно.

ВИСНОВОК

У ході виконання лабораторної роботи я розробила SQL запити для виконання операцій реляційної алгебри: проекції, селекції, натурального з'єднання, умовного з'єднання.