

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту



ЗВІТ № 5  
з курсу “ОБДЗ”  
на тему:  
**«Виконання теоретико-множинних операцій реляційної алгебри  
засобами SQL»**

**Виконала:**

студентка групи КН-211

Лаврик Юліана

**Викладач:**

Якимишин Х.М.

## Лабораторна робота №5

**Мета роботи:** Розробити SQL запити для виконання операцій реляційної алгебри: об'єднання, перетину, різниці, декартового добутку.

### Короткі теоретичні відомості

Реляційна алгебра – це множина операцій, що виконуються над відношеннями і мають за мету утворення нових відношень або їх станів. Реляційна алгебра визначає операції, які однаковим чином реалізуються в усіх базах даних реляційного типу, незалежно від їх змісту і технологій, за допомогою яких вони реалізовані. Тобто реляційна алгебра представляє собою процедурну мову обробки реляційних таблиць.

Об'єднанням двох відношень  $R$  та  $S$  з відповідними множинами атрибутів  $(A_1, A_2, \dots, A_n)$  називається відношення  $T$ , що має ту саму множину атрибутів  $(A_1, A_2, \dots, A_n)$ , а його інформаційне наповнення утворюється кортежами першого та другого відношень за вилученням повторень.

Перетином двох відношень  $R$  та  $S$  з відповідними множинами атрибутів  $(A_1, A_2, \dots, A_n)$  називається відношення  $T$ , що має ту саму множину атрибутів  $(A_1, A_2, \dots, A_n)$ , а його інформаційне наповнення утворюється кортежами, які є спільними для цих двох відношень:

Різницею двох відношень  $R$  та  $S$  з відповідними множинами атрибутів  $(A_1, A_2, \dots, A_n)$  називається відношення  $T$ , що має ту саму множину атрибутів  $(A_1, A_2, \dots, A_n)$ , а його інформаційне наповнення утворюється кортежами першого відношення за вилученням кортежів, які є спільними з другим відношенням:

Декартовим добутком двох відношень  $R$  та  $S$  з відповідними множинами атрибутів  $(A_1, A_2, \dots, A_n)$  та  $(B_1, B_2, \dots, B_m)$  називається нове відношення  $T$ , множина атрибутів якого складається з об'єднання множини атрибутів двох відношень, а кожен кортеж інформаційного наповнення утворюється шляхом конкатенації (сполучення) кожного кортежу першого відношення з кожним

кортежем другого відношення.

Для реалізації теоретико-множинних операцій на мові SQL використовують директиву SELECT, спрощений опис якої наведено далі, а також функції роботи з множинами значень IN(), NOT IN().

SELECT

[ALL | DISTINCT | DISTINCTROW ]

елемент\_вибірки [, елемент\_вибірки]

[FROM перелік\_таблиць]

[WHERE умова\_відбору]

***елемент\_вибірки***

Вираз, або назва поля, значення якого потрібно вибрати. Символ «\*» позначає всі поля.

***перелік\_таблиць***

Назва таблиці, з якої здійснюється вибір значень.

***умова\_відбору***

Вказує умови відбору потрібних записів.

***DISTINCT / DISTINCTROW***

Видалення з результату рядків-дублікатів. За замовчуванням вибираються всі рядки.

Для того, щоб виконати операцію об'єднання таблиць, потрібно за допомогою команди **UNION** об'єднати результати вибору рядків з двох, або більше, таблиць.

Наведемо синтаксис команди.

SELECT ...

UNION [ALL | DISTINCT] SELECT ...

[UNION [ALL | DISTINCT] SELECT ...]

## Хід роботи

Перед виконанням завдання, потрібно сформулювати дві таблиці з однаковими множинами атрибутів. Візьмемо за основу таблицю замовлень order і виконаємо вибір двох множин записів, які перетинаються. Результат збережемо в таблицях order1 і order2.

Таблиця order :

	id	employee_id	customer_id	date	time
▶	1	2	1	2020-03-12	18:50:32
	2	3	2	2020-03-17	13:24:12
	3	4	2	2020-03-17	20:41:56
	4	1	4	2020-03-18	10:09:34
	5	2	3	2020-03-10	14:05:21
*	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE confectionary.order1
```

```
AS SELECT id, employee_id, customer_id, date, time FROM confectionary.order
```

```
WHERE (date < '2020-03-18' AND date > '2020-03-07');
```

Таблиця order1 :

```
SELECT * FROM confectionary.order1;
```

	id	employee_id	customer_id	date	time
▶	1	2	1	2020-03-12	18:50:32
	2	3	2	2020-03-17	13:24:12
	3	4	2	2020-03-17	20:41:56
	5	2	3	2020-03-10	14:05:21

```
CREATE TABLE confectionary.order2
```

```
AS SELECT id, employee_id, customer_id, date, time FROM confectionary.order
```

```
WHERE (date < '2020-03-20' AND date > '2020-03-12');
```

Таблиця order2 :

```
SELECT * FROM confectionary.order2;
```

8 • SELECT \* FROM confectionary.order2;

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	id	employee_id	customer_id	date	time
▶	2	3	2	2020-03-17	13:24:12
	3	4	2	2020-03-17	20:41:56
	4	1	4	2020-03-18	10:09:34

1. Запит на виконання об'єднання order1 і order2 :

```
SELECT * FROM confectionary.order1
```

```
UNION SELECT * FROM confectionary.order2 ;
```

	id	employee_id	customer_id	date	time
►	1	2	1	2020-03-12	18:50:32
	2	3	2	2020-03-17	13:24:12
	3	4	2	2020-03-17	20:41:56
	5	2	3	2020-03-10	14:05:21
	4	1	4	2020-03-18	10:09:34

2. Запит на виконання перетину :

```
SELECT * FROM confectionary.order1
WHERE id IN (SELECT id FROM confectionary.order2);
```

	id	employee_id	customer_id	date	time
►	2	3	2	2020-03-17	13:24:12
	3	4	2	2020-03-17	20:41:56

3. Запит на виконання різниці order2 і order1 :

```
SELECT * FROM confectionary.order2
WHERE id NOT IN (SELECT id FROM order1);
```

	id	employee_id	customer_id	date	time
►	4	1	4	2020-03-18	10:09:34

4. Запит на виконання декартового добутку двох таблиць :

```
SELECT * FROM confectionary.order1, confectionary.order2;
```

	id	employee_id	customer_id	date	time	id	employee_id	customer_id	date	time
►	1	2	1	2020-03-12	18:50:32	2	3	2	2020-03-17	13:24:12
	1	2	1	2020-03-12	18:50:32	3	4	2	2020-03-17	20:41:56
	1	2	1	2020-03-12	18:50:32	4	1	4	2020-03-18	10:09:34
	2	3	2	2020-03-17	13:24:12	2	3	2	2020-03-17	13:24:12
	2	3	2	2020-03-17	13:24:12	3	4	2	2020-03-17	20:41:56
	2	3	2	2020-03-17	13:24:12	4	1	4	2020-03-18	10:09:34
	3	4	2	2020-03-17	20:41:56	2	3	2	2020-03-17	13:24:12
	3	4	2	2020-03-17	20:41:56	3	4	2	2020-03-17	20:41:56
	3	4	2	2020-03-17	20:41:56	4	1	4	2020-03-18	10:09:34
	5	2	3	2020-03-10	14:05:21	2	3	2	2020-03-17	13:24:12
	5	2	3	2020-03-10	14:05:21	3	4	2	2020-03-17	20:41:56
	5	2	3	2020-03-10	14:05:21	4	1	4	2020-03-18	10:09:34

**Висновок :** під час виконання даної лабораторної роботи я навчилась розробляти SQL запити для виконання операцій реляційної алгебри: об'єднання, перетину, різниці, декартового добутку.