

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



**Звіт до лабораторної роботи №5**  
**З ПРЕДМЕТУ “ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ”**

**Виконав:**

ст. гр. КН-211

Шебеко Андрій

**Викладач:**

Якимишин Х.М.

Львів – 2020

## Лабораторна робота №5

### Виконання теоретико-множинних операцій реляційної алгебри засобами SQL

**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.

Реляційна алгебра складається з таких операцій: об'єднання, перетин, різниця, декартовий добуток, проекція, селекція, натуральне з'єднання, умовне з'єднання, а також операції включення/вилучення кортежу з відношень, включення/вилучення атрибуту з відношення, зміни параметрів атрибуту.

Перші чотири операції взяті з математичної теорії множин і практично співпадають з операціями над множинами. Це зручно, оскільки реляційні таблиці є множинами, і цілком природно застосовувати до них операції над множинами.

*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:

$$R \cup S = T(A_1, A_2, \dots, A_n) = \{r\} \cup \{s\}$$

Об'єднання дозволяє нам комбінувати дані з двох таблиць з однаковими множинами атрибутів. Однакові множини атрибутів потрібні для того, щоб результатом виконання операції об'єднання була реляційна таблиця.

*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:

$$R \cap S = T(A_1, A_2, \dots, A_n) = \{r\} \cap \{s\}$$

Операція перетину дозволяє нам ідентифікувати рядки, спільні для двох таблиць.

*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:

$$R \setminus S = T(A_1, A_2, \dots, A_n) = \{r\} \setminus \{s\}$$

Операція різниці дозволяє ідентифікувати ті рядки, які є в одній таблиці, але відсутні в іншій.

*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 нову таблицю користувачів test і виконаємо вибір двох множин записів, які перетинаються. Результат збережемо в таблицях test\_1 і test\_2.

1. Створимо 2 таблиці test\_1 і test\_2.

```
create table `test_1` (  
  `id` int PRIMARY KEY AUTO_INCREMENT,  
  `name` int,  
  `value` varchar(50)  
);  
  
create table `test_2` (  
  `id` int PRIMARY KEY AUTO_INCREMENT,  
  `name` int,  
  `value` varchar(50)  
);
```

## 2. Заповнимо таблиці значеннями.

```
insert into `test_1`(`name`, `value`) values
  ('Andrii', 1234),
  ('Taras', 2345),
  ('Vasya', 3456),
  ('Leon', 4567),
  ('Double', 2)
;
```

```
insert into `test_2`(`name`, `value`) values
  ('Katya', 4321),
  ('Lera', 5432),
  ('Tanya', 6543),
  ('Zoya', 7654),
  ('Double', 2)
;
```

## 3. Виконаємо запит на об'єднання test\_1 і test\_2:

```
select * from `test_1`
union
select * from `test_2`;
```

	id	name	value
	1	Andrii	1234
	2	Taras	2345
	3	Vasya	3456
	4	Leon	4567
▶	5	Double	2
	6	Zoya	7654
	7	Tanya	6543
	8	Lera	5432
	9	Katya	4321

## 4. Запит на виконання перетину test\_1 і test\_2:

```
select * from `test_1`
where `name` in (select `name` from `test_2`);
```

	id	name	value
▶	5	Double	2

## 5. Запит на виконання різниці test\_1 і test\_2:

```
select * from `test_1`
where `name` not in (select `name` from `test_2`);
```

	id	name	value
▶	1	Andrii	1234
	2	Taras	2345
	3	Vasya	3456
	4	Leon	4567

## 6. Запит на виконання декартового добутку.

```
select * from `test_1`, `test_2`;
```

	id	name	value	id	name	value
	1	Andrii	1234	5	Double	2
	2	Taras	2345	5	Double	2
	3	Vasya	3456	5	Double	2
	4	Leon	4567	5	Double	2
►	5	Double	2	5	Double	2
	1	Andrii	1234	6	Zoya	7654
	2	Taras	2345	6	Zoya	7654
	3	Vasya	3456	6	Zoya	7654
	4	Leon	4567	6	Zoya	7654
	5	Double	2	6	Zoya	7654
	1	Andrii	1234	7	Tanya	6543
	2	Taras	2345	7	Tanya	6543
	3	Vasya	3456	7	Tanya	6543
	4	Leon	4567	7	Tanya	6543
	5	Double	2	7	Tanya	6543
	1	Andrii	1234	8	Lera	5432
	2	Taras	2345	8	Lera	5432
	3	Vasya	3456	8	Lera	5432
	4	Leon	4567	8	Lera	5432
	5	Double	2	8	Lera	5432
	1	Andrii	1234	9	Katya	4321
	2	Taras	2345	9	Katya	4321
	3	Vasya	3456	9	Katya	4321
	4	Leon	4567	9	Katya	4321
	5	Double	2	9	Katya	4321

**Висновок:** на цій лабораторній роботі було розглянуто операції реляційної алгебри та їх реалізація на мові SQL. Здійснено об'єднання, перетин, різницю та декартовий добуток двох таблиць.