

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №5
із дисципліни
Бази даних

Виконала:
Ст. групи КН-207
Ювченко Ю.
Прийняла:
Мельникова Н.І.

Львів – 2019 р.

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.

Виконаємо такі запити для занесення даних у таблиці і їх подальшої модифікації.

1. Перед виконанням завдання, потрібно сформулювати дві таблиці з однаковими множинами атрибутів. Візьмемо за основу таблицю користувачів movie і виконаємо вибір двох множин записів, які перетинаються. Результат збережемо в таблицях movie1 і movie2.
2. Запит на виконання об'єднання movie1 і movie2.
3. Запит на виконання перетину.
4. Запит на виконання різниці movie1 і movie2.
5. Запит на виконання декартового добутку двох таблиць.

1. Перед виконанням завдання, потрібно сформувати дві таблиці з однаковими множинами атрибутів. Візьмемо за основу таблицю користувачів movie і виконаємо вибір двох множин записів, які перетинаються. Результат збережемо в таблицях movie1 і movie2.

```
mysql> use new_data;
Database changed
mysql> describe movie;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | NO | PRI | NULL | auto_increment |
| name  | varchar(50) | NO | | NULL | |
| year  | int(5) | NO | | NULL | |
| time  | varchar(10) | NO | | NULL | |
| id_country | int(11) | NO | MUL | NULL | |
| id_person | int(11) | NO | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.36 sec)

mysql> select * from movie;
+-----+-----+-----+-----+-----+-----+
| id | name | year | time | id_country | id_person |
+-----+-----+-----+-----+-----+-----+
| 1 | John Wick Chspter 3 | 2019 | 2:11 | 21 | 40 |
| 2 | 'How to train dragon` | 2019 | 2:00 | 21 | 31 |
| 3 | 'Avengers:endgame' | 2019 | 3:00 | 21 | 32 |
| 4 | 'Uglydolls` | 2019 | 1:27 | 21 | 33 |
| 5 | 'Long shot' | 2019 | 2:00 | 21 | 34 |
| 6 | 'Alladin' | 2019 | 2:09 | 21 | 35 |
| 7 | 'Captain Marvel' | 2019 | 2:04 | 21 | 36 |
| 8 | 'Shazam' | 2019 | 2:12 | 21 | 37 |
| 9 | 'Little' | 2019 | 1:49 | 21 | 38 |
| 10 | 'Breakthrough' | 2019 | 1:56 | 21 | 39 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.10 sec)

mysql> create table movie1
-> as select id, name, year, time from movie
-> where id > 2;
Query OK, 8 rows affected (0.32 sec)
Records: 8 Duplicates: 0 Warnings: 0

mysql> select * from movie1;
+-----+-----+-----+-----+
| id | name | year | time |
+-----+-----+-----+-----+
| 3 | 'Avengers:endgame' | 2019 | 3:00 |
| 4 | 'Uglydolls` | 2019 | 1:27 |
| 5 | 'Long shot' | 2019 | 2:00 |
| 6 | 'Alladin' | 2019 | 2:09 |
| 7 | 'Captain Marvel' | 2019 | 2:04 |
| 8 | 'Shazam' | 2019 | 2:12 |
| 9 | 'Little' | 2019 | 1:49 |
| 10 | 'Breakthrough' | 2019 | 1:56 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> create table movie2
-> as select id, name, year, time from movie
-> where time < 2:50;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
mysql> create table movie2
-> as select id, name, year, time from movie
-> where id < 7;
Query OK, 6 rows affected (0.10 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> select * from movie2;
+-----+-----+-----+-----+
| id | name | year | time |
+-----+-----+-----+-----+
| 1 | John Wick Chspter 3 | 2019 | 2:11 |
| 2 | 'How to train dragon` | 2019 | 2:00 |
| 3 | 'Avengers:endgame' | 2019 | 3:00 |
| 4 | 'Uglydolls` | 2019 | 1:27 |
| 5 | 'Long shot' | 2019 | 2:00 |
| 6 | 'Alladin' | 2019 | 2:09 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

2. Запит на виконання об'єднання movie1 і movie2.

```
mysql> select * from movie1
-> union select * from movie2;
```

id	name	year	time
3	'Avengers:endgame'	2019	3:00
4	'Uglydolls`	2019	1:27
5	'Long shot'	2019	2:00
6	'Alladin'	2019	2:09
7	'Captain Marvel'	2019	2:04
8	'Shazam'	2019	2:12
9	'Little'	2019	1:49
10	'Breakthrough'	2019	1:56
1	John Wick Chspter 3	2019	2:11
2	'How to train dragon`	2019	2:00

10 rows in set (0.05 sec)

3. Запит на виконання перетину.

```
mysql> select * from movie1
-> where id in (select id from movie2);
```

id	name	year	time
3	'Avengers:endgame'	2019	3:00
4	'Uglydolls`	2019	1:27
5	'Long shot'	2019	2:00
6	'Alladin'	2019	2:09

4 rows in set (0.06 sec)

4. Запит на виконання різниці movie1 і movie2.

```
mysql> select * from movie2
-> where id not in (select id from movie1);
```

id	name	year	time
1	John Wick Chspter 3	2019	2:11
2	'How to train dragon`	2019	2:00

2 rows in set (0.00 sec)

5. Запит на виконання декартового добутку двох таблиць.

```
mysql> select * from movie1, movie2;
```

id	name	year	time	id	name	year	time
3	'Avengers:endgame'	2019	3:00	1	John Wick Chspter 3	2019	2:11
3	'Avengers:endgame'	2019	3:00	2	'How to train dragon'	2019	2:00
3	'Avengers:endgame'	2019	3:00	3	'Avengers:endgame'	2019	3:00
3	'Avengers:endgame'	2019	3:00	4	'Uglydolls'	2019	1:27
3	'Avengers:endgame'	2019	3:00	5	'Long shot'	2019	2:00
3	'Avengers:endgame'	2019	3:00	6	'Alladin'	2019	2:09
4	'Uglydolls'	2019	1:27	1	John Wick Chspter 3	2019	2:11
4	'Uglydolls'	2019	1:27	2	'How to train dragon'	2019	2:00
4	'Uglydolls'	2019	1:27	3	'Avengers:endgame'	2019	3:00
4	'Uglydolls'	2019	1:27	4	'Uglydolls'	2019	1:27
4	'Uglydolls'	2019	1:27	5	'Long shot'	2019	2:00
4	'Uglydolls'	2019	1:27	6	'Alladin'	2019	2:09
5	'Long shot'	2019	2:00	1	John Wick Chspter 3	2019	2:11
5	'Long shot'	2019	2:00	2	'How to train dragon'	2019	2:00
5	'Long shot'	2019	2:00	3	'Avengers:endgame'	2019	3:00
5	'Long shot'	2019	2:00	4	'Uglydolls'	2019	1:27
5	'Long shot'	2019	2:00	5	'Long shot'	2019	2:00
5	'Long shot'	2019	2:00	6	'Alladin'	2019	2:09
6	'Alladin'	2019	2:09	1	John Wick Chspter 3	2019	2:11
6	'Alladin'	2019	2:09	2	'How to train dragon'	2019	2:00
6	'Alladin'	2019	2:09	3	'Avengers:endgame'	2019	3:00
6	'Alladin'	2019	2:09	4	'Uglydolls'	2019	1:27
6	'Alladin'	2019	2:09	5	'Long shot'	2019	2:00
6	'Alladin'	2019	2:09	6	'Alladin'	2019	2:09
7	'Captain Marvel'	2019	2:04	1	John Wick Chspter 3	2019	2:11
7	'Captain Marvel'	2019	2:04	2	'How to train dragon'	2019	2:00
7	'Captain Marvel'	2019	2:04	3	'Avengers:endgame'	2019	3:00
7	'Captain Marvel'	2019	2:04	4	'Uglydolls'	2019	1:27
7	'Captain Marvel'	2019	2:04	5	'Long shot'	2019	2:00
7	'Captain Marvel'	2019	2:04	6	'Alladin'	2019	2:09
8	'Shazam'	2019	2:12	1	John Wick Chspter 3	2019	2:11
8	'Shazam'	2019	2:12	2	'How to train dragon'	2019	2:00
8	'Shazam'	2019	2:12	3	'Avengers:endgame'	2019	3:00
8	'Shazam'	2019	2:12	4	'Uglydolls'	2019	1:27
8	'Shazam'	2019	2:12	5	'Long shot'	2019	2:00
8	'Shazam'	2019	2:12	6	'Alladin'	2019	2:09
9	'Little'	2019	1:49	1	John Wick Chspter 3	2019	2:11
9	'Little'	2019	1:49	2	'How to train dragon'	2019	2:00
9	'Little'	2019	1:49	3	'Avengers:endgame'	2019	3:00
9	'Little'	2019	1:49	4	'Uglydolls'	2019	1:27
9	'Little'	2019	1:49	5	'Long shot'	2019	2:00
9	'Little'	2019	1:49	6	'Alladin'	2019	2:09
10	'Breakthrough'	2019	1:56	1	John Wick Chspter 3	2019	2:11
10	'Breakthrough'	2019	1:56	2	'How to train dragon'	2019	2:00
10	'Breakthrough'	2019	1:56	3	'Avengers:endgame'	2019	3:00
10	'Breakthrough'	2019	1:56	4	'Uglydolls'	2019	1:27
10	'Breakthrough'	2019	1:56	5	'Long shot'	2019	2:00
10	'Breakthrough'	2019	1:56	6	'Alladin'	2019	2:09

48 rows in set (0.00 sec)

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