

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



Звіт до лабораторної роботи №5
з дисципліни:
“ОБДЗ”
на тему:
**“ Виконання теоретико-множинних операцій реляційної алгебри
засобами SQL ”**

Підготувала:
студентка групи КН-209

Дипко Олександра

Викладач:
Мельникова Н.І.

Львів 2020

Мета роботи:

виконати теоретико-множинних операцій реляційної алгебри засобами SQL.

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 однаковою 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, проєкція, 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:

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:

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:

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)

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 користувачів reader і виконаємо вибір двох множин записів, які перетинаються. Результат збережемо в таблицях reader1 і reader2.

1. Виконати запит на виконання об'єднання reader1 і reader2.
2. Виконати запит на виконання перетину reader1 і reader2.
3. Запит на виконання різниці reader1 і reader2.
4. Запит на виконання декартового добутку двох таблиць.

```
CREATE TABLE reader1
AS SELECT id_reader, name, surname, age FROM reader
WHERE age > "18";
```

```
mysql> select * from reader1;
+-----+-----+-----+-----+
| id_reader | name | surname | age |
+-----+-----+-----+-----+
| 2 | Roman | Pyk | 22 |
| 3 | Kris | Kosyk | 23 |
| 4 | Iryna | Pushkina | 19 |
+-----+-----+-----+-----+
```

```
CREATE TABLE reader2
AS SELECT id_reader, name , age FROM library.reader
WHERE age < "22";
```

```
mysql> select * from reader2;
+-----+-----+-----+-----+
| id_reader | name | surname | age |
+-----+-----+-----+-----+
| 1 | Nastya | Shevchenko | 18 |
| 4 | Iryna | Pushkina | 19 |
+-----+-----+-----+-----+
```

1. Виконати запит на виконання об'єднання reader1 і reader2.

```
SELECT * FROM reader1
UNION SELECT * FROM reader2
```

```
+-----+-----+-----+-----+
| id_reader | name | surname | age |
+-----+-----+-----+-----+
| 2 | Roman | Pyk | 22 |
| 3 | Kris | Kosyk | 23 |
| 4 | Iryna | Pushkina | 19 |
| 1 | Nastya | Shevchenko | 18 |
+-----+-----+-----+-----+
```

2. Виконати запит на виконання перетину reader1 і reader2.

```
SELECT * FROM reader2
WHERE id_reader IN (SELECT id_reader FROM reader1);
```

```
mysql> SELECT * FROM reader2
-> WHERE id_reader IN (SELECT id_reader FROM reader1);
+-----+-----+-----+-----+
| id_reader | name | surname | age |
+-----+-----+-----+-----+
| 4 | Iryna | Pushkina | 19 |
+-----+-----+-----+-----+
```

3. Запит на виконання різниці reader1 і reader2.

```
SELECT * FROM reader1
WHERE id_reader NOT IN (SELECT id_reader FROM reader2);
```

```
mysql> SELECT * FROM reader1
-> WHERE id_reader NOT IN (SELECT id_reader FROM reader2);
```

id_reader	name	surname	age
2	Roman	Pyk	22
3	Kris	Kosyk	23

4. Запит на виконання декартового добутку двох таблиць.

```
SELECT * FROM reader2, reader1;
```

```
mysql> SELECT * FROM reader2, reader1;
```

id_reader	name	surname	age	id_reader	name	surname	age
1	Nastya	Shevchenko	18	2	Roman	Pyk	22
4	Iryna	Pushkina	19	2	Roman	Pyk	22
1	Nastya	Shevchenko	18	3	Kris	Kosyk	23
4	Iryna	Pushkina	19	3	Kris	Kosyk	23
1	Nastya	Shevchenko	18	4	Iryna	Pushkina	19
4	Iryna	Pushkina	19	4	Iryna	Pushkina	19

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.