Лабораторна робота №5
з курсу “ОБДЗ”
на тему:
“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 $(A_1, A_2,..., A_n)$ називається відношення T, що має ту саму множину атрибутів $(A_1, A_2,..., A_n)$, а його інформаційне наповнення утворюється кортежами першого та другого відношень за вилученням повторень:

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

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 $(A_1, A_2,..., A_n)$ називається відношення T, що має ту саму множину атрибутів $(A_1, A_2,..., A_n)$, а його інформаційне наповнення утворюється кортежами, які є спільними для цих двох відношень:

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

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

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

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

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 $(A_1, A_2,..., A_n)$ та $(B_1, B_2,..., B_m)$ 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 сформувати дві таблиці з однаковими множинами атрибутів. Візьмемо за основу таблицю користувачів Author і виконаємо вибір двох множин записів, які перетинаються. Результат збережемо в таблицях Author1 і Author2.

```
CREATE TABLE mycms.author1
 AS SELECT authorID, login, created FROM mycms.author
      WHERE created < '2009-01-01';
```

Таблиця Author1:

```
mysql> select * from author1;
+----------+--------+---------------------+
| authorID | login  | created             |
+----------+--------+---------------------+
|        2 | admin  | 2008-01-01 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |
+----------+--------+---------------------+
2 rows in set (0.00 sec)
```

```
CREATE TABLE mycms.author2
 SELECT authorID, login, created FROM mycms.author
 WHERE created < 20090201;
```

Таблиця Author2:

```
mysql> select * from author2;
+----------+------------+---------------------+
| authorID | login      | created             |
+----------+------------+---------------------+
|        2 | admin      | 2008-01-01 00:00:00 |
|        3 | user1      | 2009-01-02 00:00:00 |
|        4 | user2      | 2009-01-03 00:00:00 |
|        6 | guest1     | 2008-02-16 00:00:00 |
|        8 | superuser1 | 2009-01-01 00:00:00 |
|        9 | superuser2 | 2009-01-01 00:00:00 |
+----------+------------+---------------------+
6 rows in set (0.00 sec)
```

1. Запит на виконання об'єднання Author1 і Author2:

```
SELECT * FROM Author1
 UNION SELECT * FROM Author2
```

```
mysql> select * from author1 union select * from author2;
+----------+------------+---------------------+
| authorID | login      | created             |
+----------+------------+---------------------+
|        2 | admin      | 2008-01-01 00:00:00 |
|        6 | guest1     | 2008-02-16 00:00:00 |
|        3 | user1      | 2009-01-02 00:00:00 |
|        4 | user2      | 2009-01-03 00:00:00 |
|        8 | superuser1 | 2009-01-01 00:00:00 |
|        9 | superuser2 | 2009-01-01 00:00:00 |
+----------+------------+---------------------+
6 rows in set (0.00 sec)
```

2. Запит на виконання перетину:

```
SELECT * FROM author1
 WHERE authorID IN (SELECT authorID FROM author2);
```

```
mysql> SELECT * FROM author1
    -> WHERE authorID IN (SELECT authorID FROM author2);
+----------+--------+---------------------+
| authorID | login  | created             |
+----------+--------+---------------------+
|        2 | admin  | 2008-01-01 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |
+----------+--------+---------------------+
2 rows in set (0.02 sec)
```

3. Запит на виконання різниці Author2 і Author1:

```
SELECT * FROM author2
 WHERE authorID NOT IN (SELECT authorID FROM author1);
```

```
mysql> SELECT * FROM author2
    -> WHERE authorID NOT IN (SELECT authorID FROM author1);
+----------+------------+---------------------+
| authorID | login      | created             |
+----------+------------+---------------------+
|        3 | user1      | 2009-01-02 00:00:00 |
|        4 | user2      | 2009-01-03 00:00:00 |
|        8 | superuser1 | 2009-01-01 00:00:00 |
|        9 | superuser2 | 2009-01-01 00:00:00 |
+----------+------------+---------------------+
4 rows in set (0.00 sec)
```

4. Запит на виконання декартового добутку двох таблиць:

```
SELECT * FROM author1, author2;
```

```
mysql> SELECT * FROM author1, author2;
+----------+--------+---------------------+----------+------------+---------------------+
| authorID | login  | created             | authorID | login      | created             |
+----------+--------+---------------------+----------+------------+---------------------+
|        2 | admin  | 2008-01-01 00:00:00 |        2 | admin      | 2008-01-01 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |        2 | admin      | 2008-01-01 00:00:00 |
|        2 | admin  | 2008-01-01 00:00:00 |        3 | user1      | 2009-01-02 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |        3 | user1      | 2009-01-02 00:00:00 |
|        2 | admin  | 2008-01-01 00:00:00 |        4 | user2      | 2009-01-03 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |        4 | user2      | 2009-01-03 00:00:00 |
|        2 | admin  | 2008-01-01 00:00:00 |        6 | guest1     | 2008-02-16 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |        6 | guest1     | 2008-02-16 00:00:00 |
|        2 | admin  | 2008-01-01 00:00:00 |        8 | superuser1 | 2009-01-01 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |        8 | superuser1 | 2009-01-01 00:00:00 |
|        2 | admin  | 2008-01-01 00:00:00 |        9 | superuser2 | 2009-01-01 00:00:00 |
|        6 | guest1 | 2008-02-16 00:00:00 |        9 | superuser2 | 2009-01-01 00:00:00 |
+----------+--------+---------------------+----------+------------+---------------------+
12 rows in set (0.00 sec)
```

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.