

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

Кафедра систем штучного інтелекту

Лабораторна робота №13
із дисципліни
Бази даних

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

Львів – 2019 р.

Мета роботи: Навчитися аналізувати роботу СУБД та оптимізовувати виконання складних запитів на вибірку даних. Виконати аналіз складних запитів за допомогою директиви EXPLAIN, модифікувати найповільніші запити з метою їх пришвидчення.

Хід роботи.

Потрібно розробити наступні тригери, які виконуватимуть наступні дії.

1. Визначимо індекси таблиці Employee та Passport.
2. Створимо додаткові індекси для таблиць.
3. Дослідимо процес виконання запитів за допомогою EXPLAIN.

1. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць Employee та Passport.

```
mysql> show index from account;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| account | 0 | PRIMARY | 1 | id | A | 13 | NULL | NULL | NULL | BTREE | | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.08 sec)

mysql> show index from movie;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| movie | 0 | PRIMARY | 1 | id | A | 10 | NULL | NULL | NULL | BTREE | | | YES | NULL |
| movie | 1 | countryId_fk | 1 | id_country | A | 1 | NULL | NULL | NULL | BTREE | | | YES | NULL |
| movie | 1 | Movie_fk3 | 1 | id_person | A | 10 | NULL | NULL | NULL | BTREE | | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)
```

2. Створимо новий індекс для таблиць Employee та Passport.

```
mysql> create index movieidx on movie (name, id);
Query OK, 0 rows affected (0.22 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create index accountidx on account (login, id);
Query OK, 0 rows affected (0.18 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

3. Виконаємо аналіз виконання складного запиту з однією з попередніх робіт використовуючи EXPLAIN та опцію STRAIGHT JOIN.

```
mysql> explain select genre.name, count(movie.id) as amount
-> from ((movie inner join genre) inner join hall_movie) inner join hall
-> on movie.id_genre = genre.id
-> and movie.id = hall_movie.id_movie
-> and hall.id = hall_movie.id_hall
-> where hall_movie.datatime between '2018-01-01' and '2019-06-01'
-> group by name;

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | hall_movie | NULL | ALL | Hall_Movie_fk0,Hall_Movie_fk1 | NULL | NULL | NULL | 11 | 11.11 | Using where; Using temporary |
| 1 | SIMPLE | hall | NULL | eq_ref | PRIMARY | PRIMARY | 4 | new_data.hall_movie.id_hall | 1 | 100.00 | Using index |
| 1 | SIMPLE | movie | NULL | eq_ref | PRIMARY | PRIMARY | 4 | new_data.hall_movie.id_movie | 1 | 100.00 | NULL |
| 1 | SIMPLE | genre | NULL | ALL | NULL | NULL | NULL | NULL | 9 | 11.11 | Using where; Using join buffer (Block Nested Loop) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.17 sec)

mysql>

mysql> explain select straight_join genre.name, count(movie.id) as amount
-> from ((movie inner join genre) inner join hall_movie) inner join hall
-> on movie.id_genre = genre.id
-> and movie.id = hall_movie.id_movie
-> and hall.id = hall_movie.id_hall
-> where hall_movie.datatime between '2018-01-01' and '2019-06-01'
-> group by name;

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | movie | NULL | ALL | PRIMARY | NULL | NULL | NULL | 10 | 100.00 | Using temporary |
| 1 | SIMPLE | genre | NULL | ALL | NULL | NULL | NULL | NULL | 9 | 11.11 | Using where; Using join buffer (Block Nested Loop) |
| 1 | SIMPLE | hall_movie | NULL | ALL | Hall_Movie_fk0,Hall_Movie_fk1 | NULL | NULL | NULL | 11 | 9.09 | Using where; Using join buffer (Block Nested Loop) |
| 1 | SIMPLE | hall | NULL | eq_ref | PRIMARY | PRIMARY | 4 | new_data.hall_movie.id_hall | 1 | 100.00 | Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)
```

Висновок: на цій лабораторній роботі навчилася аналізувати і оптимізувати виконання запитів. Для аналізу запитів було використано директиву EXPLAIN, а для оптимізації – модифікація порядку з'єднання таблиць і створення додаткових індексів.