МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту

Лабораторна робота №13

з дисципліни «Організація баз даних та знань»

Виконав: студент групи КН-209

> Качмар Олексій Викладач: Мельникова Н.І.

Лабораторна робота №13

Тема: "Аналіз та оптимізація запитів".

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.

Короткі теоретичні відомості.

Для аналізу виконання запитів в MySQL існує декілька спеціальних директив. Основна з них — EXPLAIN. Директива EXPLAIN дозволяє визначити поля таблиці, для яких варто створити додаткові індекси, щоб пришвидшити вибірку даних. Індекс — це механізм, який підвищує швидкість пошуку та доступу до записів за індексованими полями. Загалом, варто створювати індекси для тих полів, за якими відбувається з'єднання таблиць, перевірка умови чи пошук. За допомогою директиви EXPLAIN також можна визначити послідовність, в якій відбувається з'єднання таблиць при вибірці даних. Якщо оптимізатор вибирає не найкращу послідовність з'єднання таблиць, потрібно використати опцію STRAIGHT_JOIN директиви SELECT. Тоді з'єднання таблиць буде відбуватись в тому порядку, в якому перераховані таблиці у запиті. Також, за допомогою опцій FORCE INDEX, USE INDEX та IGNORE INDEX можна керувати використанням індексів у випадку їх неправильного вибору оптимізатором, тобто, якщо вони не підвищують ефективність вибірки рядків.

Завдання на лабораторну роботу.

1. Визначити індекси таблиці.

За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць Meal i Regular_customer та Meal_has_regular_customer.

2. Створити додаткові індекси для таблиці.

```
mysql> Create index dish_gram_idx On meal(dish,gram);
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show index from meal;
                            | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment
 Table | Non_unique | Key_name
| Visible | Expression |
             0 | PRIMARY | 1 | dish
meal
                                                                        0 | NULL | NULL |
                                                                                             BTREE
 YES
       NULL
 meal |
             1 | fk_meal_restaurant1_idx | 1 | restaurant_Id | A
                                                                              NULL | NULL |
                                                                                             BTREE
 YES
         1 | dish_gram_idx
                                         1 | dish
                                                                        0
                                                                              NULL | NULL |
                                                                                             BTREE
 meal
      NULL
 YES
meal |
| YES
         1 | dish_gram_idx
                                                                        0 |
                                                                              NULL | NULL |
                                                                                             BTREE
                                          2 gram
        NULL
```

```
mysql> create index full_name_idx on regular_customer(`full_name`);
Query OK, 0 rows affected (0.31 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
nysql> show index from regular customer;
Table | Non_unique | Key_name
                                            | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type
Comment | Index_comment | Visible | Expression |
                                                     1 | Id
regular customer
                0 | PRIMARY
                                                                                  5 |
                                                                                       NULL | NULL |
               regular_customer |
                                                                                        NULL | NULL | BTREE
                      NULL
                1 | full_name_idx
regular_customer
                                                      1 | full_name | A
                                                                                        NULL | NULL | BTREE
               YES NULL |
```

3. Дослідити процес виконання запитів за допомогою EXPLAIN.

1)Explain select

```
Explain select IFNULL(meal.dish, 'Total'), count(meal.dish) as count, Sum(meal.price) AS price from meal_has_regular_customer

Inner Join regular_customer

On meal_has_regular_customer.regular_customer_id=regular_customer.id

And meal_has_regular_customer.dish_name = meal.dish

Where regular_customer.full_name = "Alex"

Group By meal.dish

With ROLLUP

Order by gram ;
```

2)Explain select straight_join

```
Explain select straight_join meal.dish,count(meal.dish) as count,Sum(meal.price) AS price from meal_has_regular_customer
Inner Join meal
Inner Join regular_customer
On meal_has_regular_customer.regular_customer_id=regular_customer.id
And meal_has_regular_customer.dish_name = meal.dish
Where regular_customer.full_name = "Alex"
Group By meal.dish
With ROLLUP
Order by gram ;
```

Висновок: я навчився аналізувати роботу СУБД та оптимізовувати виконання складних запитів на вибірку даних. Виконав аналіз складних запитів за

допомогою директиви EXPLAIN, модифікував найповільніші запити з метою їх пришвидчення.