МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту

Звіт до лабораторної роботи №13

з дисципліни
“Організація Баз Даних та знань”

Виконала:
ст. гр. КН-210
Заремба Вікторія

Викладач:
Мельникова Н.І.

Львів – 2020

Лабораторна робота №12
з курсу “ОБДЗ”
на тему:
“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.

Опис директив.

SELECT BENCHMARK(кількість_циклів, вираз) Виконує вираз вказану кількість разів, і повертає загальний час виконання.

EXPLAIN SELECT … Використовується разом із запитом SELECT. Виводить інформацію про план обробки і виконання запиту, включно з інформацією про те, як і в якому порядку з’єднувались таблиці. EXPLAIN EXTENDED виводить розширену інформацію.

Результати директиви виводяться у вигляді рядків з такими полями:

id – порядковий номер директиви SELECT у запиті; select_type – тип вибірки (simple, primary, union, subquery, derived, uncachable subquery тощо); table – назва таблиці, для якої виводиться інформація; type – тип з’єднання (system, const, eq_ref, ref, fulltext, range тощо); possible_keys – індекси, які наявні у таблиці, і можуть бути використані; key – назва індексу, який було обрано для виконання запиту; key_len – довжина індекса, який був використаний при виконанні запиту; ref – вказує, які рядки чи константи порівнюються зі значенням індекса при відборі; rows – (прогнозована) кількість рядків, потрібних для виконання запиту; Extra – додаткові дані про хід виконання запиту.

ANALYZE TABLE Оновлює статистичну інформацію про таблицю (наприклад, поточний розмір ключових полів). Ця інформація впливає на роботу оптимізатора запитів, і може вплинути на вибір індексів при виконанні запитів.

SHOW INDEX FROM ім’я_таблиці Виводить інформацію про індекси таблиці.

CREATE [UNIQUE | FULLTEXT] INDEX назва ON ім’я_таблиці (перелік_полів) Створює індекс для одного або декількох полів таблиці. Одне поле може входити до кількох індексів. Якщо індекс оголошено як UNIQUE, то значення відповідних полів таблиці повинні бути унікальними. Таблиці MyISAM підтримують створення повнотекстових індексів (FULLTEXT) для полів типу TEXT, CHAR, VARCHAR.

Завдання на лабораторну роботу.

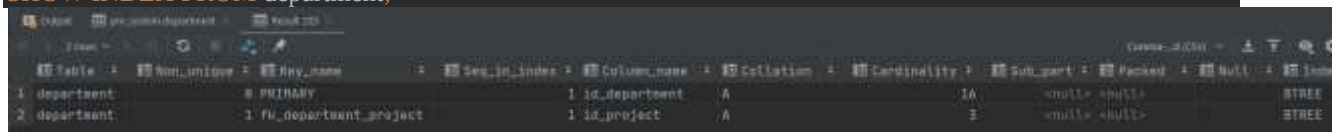
1. Визначити індекси таблиці.
2. Створити додаткові індекси для таблиці.
3. Дослідити процес виконання запитів за допомогою EXPLAIN.

Хід Роботи

1. **Визначити індекси таблиці.**
   За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць project і department.

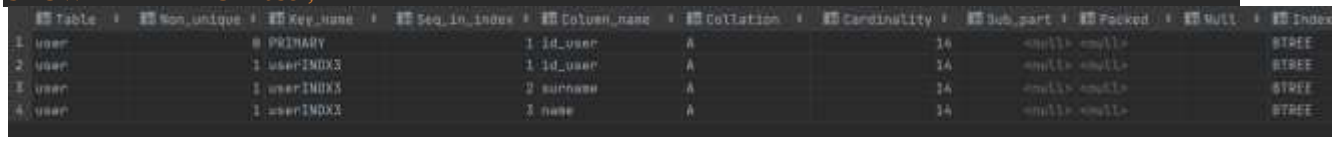
```
SHOW INDEX FROM project;
```

```
SHOW INDEX FROM department;
```



2. **Створити додаткові індекси для таблиці.**
   Створимо додаткові індекси для обраних таблиць. Маємо запити, які здійснюють вибірку даних по прізвищу юзера, по назві проекту, по датах тасків, тощо.

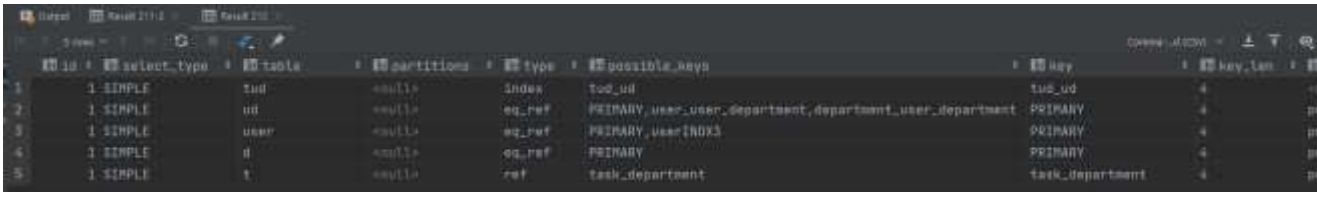
```
CREATE INDEX userINDX3 ON user (id_user, surname, name);
SHOW INDEX FROM user;
```



```
CREATE UNIQUE INDEX projIND ON project (project_name);
SHOW INDEX FROM project;
```

3. **Дослідити процес виконання запитів за допомогою EXPLAIN.**
   Виконаємо аналіз складного запиту з 7 лабораторної використовуючи EXPLAIN та опцію STRAIGHT_JOIN.

```
EXPLAIN SELECT surname, name, task_name FROM user
INNER JOIN user_department ud on user.id_user = ud.id_user
INNER JOIN task_user_department tud on ud.id_user_department = tud.id_user_department
INNER JOIN department d on ud.id_department = d.id_department
INNER JOIN task t on d.id_department = t.id_department;
```



	id	select_type	table	partitions	type	possible_keys	key	key_len
1	1	SIMPLE	tud	<null>	index	tud_ud	tud_ud	4
2	1	SIMPLE	ud	<null>	eq_ref	PRIMARY,user_user_department,department_user_department	PRIMARY	4
3	1	SIMPLE	user	<null>	eq_ref	PRIMARY,userINDX3	PRIMARY	4
4	1	SIMPLE	d	<null>	eq_ref	PRIMARY	PRIMARY	4
5	1	SIMPLE	t	<null>	ref	task_department	task_department	4

```
EXPLAIN SELECT STRAIGHT_JOIN surname, name, task_name FROM user
INNER JOIN user_department ud on user.id_user = ud.id_user
INNER JOIN task_user_department tud on ud.id_user_department = tud.id_user_department
INNER JOIN department d on ud.id_department = d.id_department
INNER JOIN task t on d.id_department = t.id_department;
```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref
1	1	SIMPLE	user	<null>	index	PRIMARY,userINDX3	userINDX3	210	<null>
2	1	SIMPLE	ud	<null>	ref	PRIMARY,user_user_departmen…	user_user_department	4	pe_system.user.id_user
3	1	SIMPLE	tud	<null>	ref	tud_ud	tud_ud	4	pe_system.ud.id_user_depa
4	1	SIMPLE	d	<null>	eq_ref	PRIMARY	PRIMARY	4	pe_system.ud.id_department
5	1	SIMPLE	t	<null>	ref	task_department	task_department	4	pe_system.ud.id_department

Висновок*:* На даній лабораторній роботі я навчився аналізувати і оптимізувати виконання запитів. Для аналізу запитів було використано директиву EXPLAIN, а для оптимізації – модифікація порядку з’єднання таблиць і створення додаткових індексів.