

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



**Звіт до лабораторної роботи №13**  
**З ПРЕДМЕТУ “ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ”**

**Виконав:**

ст. гр. КН-211

Шебеко Андрій

**Викладач:**

Якимишин Х.М.

Львів – 2020

Лабораторна робота №13  
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.

## Хід роботи

1. За допомогою директиви show index визначимо наявні індекси для таблиць client і comment.

```
show index from client;
show index from comment;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
client	0	PRIMARY	1	client_id	A	10	NULL	NULL		BTREE

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
comment	0	PRIMARY	1	comment_id	A	6	NULL	NULL		BTREE

2. Я взяв складний запит з 9 лабораторної роботи та виконав його з директивою explain, яка виводить інформацію про план обробки і виконання запиту.

```
explain select `first name`, count(distinct comment) as comments,
ifnull(sum(distinct amount), 0) as products,
(count(distinct comment) + ifnull(sum(distinct amount), 0)*2) as rating
from client left join comment
on client.`first name` = comment.username
left join basket
on client.client_id = basket.client_id
group by `first name`
order by rating DESC limit 3;
```

select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
SIMPLE	client	NULL	ALL	NULL	NULL	NULL	NULL	10	100.00
SIMPLE	comment	NULL	ALL	NULL	NULL	NULL	NULL	6	100.00
SIMPLE	basket	NULL	ref	b_client_idx	b_client_idx	4	shop.client.client_id	1	100.00

3. Виконав попередній запит з аргументом `straight_join`, який об'єднує таблиці у порядку їх оголошення в запиті.

```
explain select straight_join `first name`, count(distinct comment) as comments,
ifnull(sum(distinct amount), 0) as products,
(count(distinct comment) + ifnull(sum(distinct amount), 0)*2) as rating
from client left join comment
on client.`first name` = comment.username
left join basket
on client.client_id = basket.client_id
group by `first name`
order by rating DESC limit 3;
```

select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
SIMPLE	client	<small>NULL</small>	ALL	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	10	100.00
SIMPLE	comment	<small>NULL</small>	ALL	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	6	100.00
SIMPLE	basket	<small>NULL</small>	ref	b_client_idx	b_client_idx	4	shop.client.client_id	1	100.00

Ми бачимо, що у стовбці `type` містяться записи `All`, що значить що перебираються усі рядки таблиці, що є не оптимально. Щоб це виправити створимо індекси, які повинні пришвидшити пошук.

4. Створимо індекси для таблиць `client` і `comment`.

```
create index clientIND on client (`first name`);
create index commentIND on comment (username);
```

5. Знову перевіримо наявні індекси у даних таблицях

```
show index from client;
show index from comment;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
client	0	PRIMARY	1	client_id	A	10	<small>NULL</small>	<small>NULL</small>		BTREE
client	1	clientIND	1	first name	A	10	<small>NULL</small>	<small>NULL</small>	YES	BTREE

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
comment	0	PRIMARY	1	comment_id	A	6	<small>NULL</small>	<small>NULL</small>		BTREE
comment	1	commentIND	1	username	A	5	<small>NULL</small>	<small>NULL</small>	YES	BTREE

6. Знову використаємо директиву `explain`.

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
1	SIMPLE	client	<small>NULL</small>	index	clientIND	clientIND	163	<small>NULL</small>	10	100.00
1	SIMPLE	comment	<small>NULL</small>	ref	commentIND	commentIND	203	shop.client.first name	1	100.00
1	SIMPLE	basket	<small>NULL</small>	ref	b_client_idx	b_client_idx	4	shop.client.client_id	1	100.00

Ми бачимо що у стовбці type уже нема записів All, що є прогресом, проте наявний запис index, що є краще але не досконало, проте головна задача – оптимізація запиту, виконана успішно.

**Висновок:** я навчився аналізувати роботу СУБД та оптимізувати виконання складних запитів на вибірку даних. А також виконав аналіз складних запитів за допомогою директиви EXPLAIN, модифікувати найповільніші запити з метою їх пришвидшення.