

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”

Виконав:
ст. гр. КН-211
Петров Дмитро

Викладач:
Якимишин Х.М.

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.

Хід роботи

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

1. Визначити індекси таблиці.

SHOW INDEX FROM user;

name	index_type	comment	index_comment	visible	expression
PRIMARY	PRIMARY			YES	

2. Проаналізувати ефективність запитів.

a)

```
EXPLAIN SELECT group_name AS category, COUNT(db_sht.order.id) AS amount
FROM (((db_sht.group INNER JOIN article_group)
INNER JOIN goods)
INNER JOIN db_sht.order)
INNER JOIN db_sht.user ON db_sht.user.email = '1233@hots.com'
AND user.id = db_sht.order.user_id
AND db_sht.order.id = goods.order_id
AND goods.goods_id = article_group.goods_id
AND article_group.group_id = db_sht.group.id
WHERE TO_DAYS(CURRENT_DATE) - TO_DAYS(db_sht.order.ordered) <= 31
GROUP BY category;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	user	NULL	ALL	PRIMARY	NULL	NULL	NULL	10	10.00	Using where; Using temporary
1	SIMPLE	goods	NULL	ALL	goods_id,order_id	NULL	NULL	NULL	7	100.00	Using join buffer (Block Nested Loop)
1	SIMPLE	order	NULL	eq_ref	PRIMARY,user_id	PRIMARY	4	db_sht.goods.order_id	1	25.00	Using where
1	SIMPLE	article_group	NULL	ref	group_id,goods_id	goods_id	4	db_sht.goods.goods_id	1	100.00	NULL
1	SIMPLE	group	NULL	eq_ref	PRIMARY	PRIMARY	4	db_sht.article_group.group_id	1	100.00	NULL

b)

```
EXPLAIN SELECT group_name AS category, COUNT(db_sht.order.id) AS amount
FROM (((db_sht.group INNER JOIN article_group)
INNER JOIN goods)
INNER JOIN db_sht.order)
INNER JOIN db_sht.user ON db_sht.user.email = '1233@hots.com'
AND user.id = db_sht.order.user_id
AND db_sht.order.id = goods.order_id
```

```

AND goods.goods_id = article_group.goods_id
AND article_group.group_id = db_sht.group.id
WHERE db_sht.order.ordered >= DATE_SUB(CURRENT_DATE, INTERVAL 31
DAY)
GROUP BY category;

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	order	NULL	ALL	PRIMARY,user_id	NULL	NULL	NULL	8	33.33	Using where; Using temporary
1	SIMPLE	user	NULL	eq_ref	PRIMARY	PRIMARY	4	db_sht.order.user_id	1	10.00	Using where
1	SIMPLE	goods	NULL	ref	goods_id,order_id	order_id	4	db_sht.order.id	1	100.00	NULL
1	SIMPLE	article_group	NULL	ref	group_id,goods_id	goods_id	4	db_sht.goods.goods_id	1	100.00	NULL
1	SIMPLE	group	NULL	eq_ref	PRIMARY	PRIMARY	4	db_sht.article_group.group_id	1	100.00	NULL

Як бачимо в залежності від формування функції вибору запиту, сама вибірка може проводитися. Про це свідчить поле type, що відображає по яких полях відбувається вибірка. Втаблиці ордер вона має проводитися по одному полю. Також це видно в полі Extras, де вказані дії, що відбувається з кожною таблицею при вибірці – використовується тимчасове з'єднання, що сповільнює операцію вибірки.

3. Створити додаткові індекси для таблиці.

a)

```

SHOW INDEX FROM user;
ALTER TABLE user MODIFY email varchar(50);
CREATE UNIQUE INDEX user_coupled_index_1 ON user (id, email);

-- ALTER TABLE order MODIFY email varchar(50);
CREATE UNIQUE INDEX order_index_1 ON db_sht.order (id);
alter table goods drop index goods_index_1;
CREATE UNIQUE INDEX goods_index_1 ON goods (id, order_id);

```

SHOW INDEX FROM goods;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
user	0	PRIMARY	1	id	A	10	NULL	NULL		BTREE			YES	NULL
user	0	user_coupled_index_1	1	id	A	10	NULL	NULL		BTREE			YES	NULL
user	0	user_coupled_index_1	2	email	A	10	NULL	NULL	YES	BTREE			YES	NULL

4. Дослідити процес виконання запитів за допомогою EXPLAIN.

a)

```

EXPLAIN SELECT group_name AS category, COUNT(db_sht.order.id) AS
amount

```

```

FROM (((db_sht.group INNER JOIN article_group)
INNER JOIN goods)
INNER JOIN db_sht.order)
INNER JOIN db_sht.user ON db_sht.user.email =
'1233@hots.com'
AND user.id = db_sht.order.user_id

```

```

AND db_sht.order.id = goods.order_id
AND goods.goods_id = article_group.goods_id
AND article_group.group_id = db_sht.group_id
WHERE TO_DAYS(CURRENT_DATE) -
TO_DAYS(db_sht.order.ordered) <= 31
GROUP BY category;

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	order	NULL	ALL	PRIMARY,user_id	NULL	NULL	NULL	8	33.33	Using where; Using temporary
1	SIMPLE	user	NULL	eq_ref	PRIMARY,user_coupled_index_1	PRIMARY	4	db_sht.order.user_id	1	10.00	Using where
1	SIMPLE	goods	NULL	ref	goods_id,order_id	order_id	4	db_sht.order.id	1	100.00	NULL
1	SIMPLE	article_group	NULL	ref	group_id,goods_id	goods_id	4	db_sht.goods.goods_id	1	100.00	NULL
1	SIMPLE	group	NULL	eq_ref	PRIMARY	PRIMARY	4	db_sht.article_group.group_id	1	100.00	NULL

b)

```

EXPLAIN SELECT group_name AS category, COUNT(db_sht.order.id) AS amount
FROM (((db_sht.group INNER JOIN article_group)
INNER JOIN goods)
INNER JOIN db_sht.order)
INNER JOIN db_sht.user ON db_sht.user.email = '1233@hots.com'
AND user.id = db_sht.order.user_id
AND db_sht.order.id = goods.order_id
AND goods.goods_id = article_group.goods_id
AND article_group.group_id = db_sht.group_id
WHERE db_sht.order.ordered >= DATE_SUB(CURRENT_DATE, INTERVAL 31
DAY)
GROUP BY category;

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	order	NULL	ALL	PRIMARY,user_id	NULL	NULL	NULL	8	100.00	Using where; Using temporary
1	SIMPLE	user	NULL	eq_ref	PRIMARY,user_coupled_index_1	PRIMARY	4	db_sht.order.user_id	1	10.00	Using where
1	SIMPLE	goods	NULL	ref	goods_id,order_id	order_id	4	db_sht.order.id	1	100.00	NULL
1	SIMPLE	article_group	NULL	ref	group_id,goods_id	goods_id	4	db_sht.goods.goods_id	1	100.00	NULL
1	SIMPLE	group	NULL	eq_ref	PRIMARY	PRIMARY	4	db_sht.article_group.group_id	1	100.00	NULL

В результаті в обох варіантах вибірки ми отримали оптимальний пошук.

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