

Лабораторна робота №13

з дисципліни
«Організація баз даних та знань»

Виконав:
студент групи КН-207
Шиманський П.С.
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.

1. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць producer і category.

```
mysql> SHOW INDEX FROM producer;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
producer	0	PRIMARY	1	idProducer	A	9	NULL	NULL		BTREE			YES	NULL

1 row in set (0.10 sec)

```
mysql>
```

```
mysql> SHOW INDEX FROM category;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
category	0	PRIMARY	1	idCategory	A	2	NULL	NULL		BTREE			YES	NULL

1 row in set (0.13 sec)

Створимо новий індекс для таблиці producer і category. У БД є декілька запитів, які здійснюють вибірку даних за логіном автора (), за датою написання повідомлення тощо. Створення індексів для цих полів повинно оптимізувати виконання запитів.

```
mysql> SHOW INDEX FROM producer;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
producer	0	PRIMARY	1	idProducer	A	9	NULL	NULL		BTREE			YES	NULL
producer	1	producerINDX3	1	idProducer	A	9	NULL	NULL		BTREE			YES	NULL
producer	1	producerINDX3	2	Producer_Name	A	9	NULL	NULL		BTREE			YES	NULL

3 rows in set (0.06 sec)

```
mysql> SHOW INDEX FROM category;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
category	0	PRIMARY	1	idCategory	A	2	NULL	NULL		BTREE			YES	NULL
category	1	categoryINDX4	1	idCategory	A	2	NULL	NULL		BTREE			YES	NULL
category	1	categoryINDX4	2	Category_Name	A	2	NULL	NULL		BTREE			YES	NULL

3 rows in set (0.07 sec)

3. Виконаємо аналіз виконання складного запиту з однієї з попередніх робіт використовуючи EXPLAIN та опцію STRAIGHT_JOIN.

```
mysql> explain select Producer_Country tag, count(product.idProduct) as amount
-> from ((producer inner join product)
-> inner join comment) inner join category
-> on producer.Producer_Name=Product_Name
-> and producer.idProducer=product.idProduct
-> group by tag;
```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref		rows	filtered	Extra
1	1	SIMPLE	category	NULL	index	NULL	PRIMARY	4	NULL		2	100.00	Using index; Using temporary
2	1	SIMPLE	comment	NULL	index	NULL	Comment_Product_Fkey	4	NULL		3	100.00	Using index; Using join buffer (Block Nested Loop)
3	1	SIMPLE	product	NULL	ALL	PRIMARY	NULL	NULL	NULL		6	100.00	Using join buffer (Block Nested Loop)
4	1	SIMPLE	producer	NULL	eq_ref	PRIMARY, producerINDX3	PRIMARY	4	internet_shop.product.idProduct		1	11.11	Using where

4 rows in set, 1 warning (0.24 sec)

```
mysql> explain select straight_join Producer_Country as tag,
-> count(product.idProduct) as amount
-> from ((producer inner join product)
-> inner join comment) inner join category
-> on producer.Producer_Name=Product_Name and producer.idProducer=product.idProducer
-> group by tag;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	producer	NULL	ALL	PRIMARY,producerINDEX3	NULL	NULL	NULL	9	100.00	Using temporary
1	SIMPLE	product	NULL	ALL	NULL	NULL	NULL	NULL	6	15.67	Using where; Using join buffer (Block Nested Loop)
1	SIMPLE	comment	NULL	index	NULL	Comment_Product_Fkey	4	NULL	3	100.00	Using index; Using join buffer (Block Nested Loop)
1	SIMPLE	category	NULL	index	NULL	PRIMARY	4	NULL	2	100.00	Using index; Using join buffer (Block Nested Loop)

4 rows in set, 1 warning (0.00 sec)

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.