

**Тема:** аналіз та оптимізація запитів.

**Мета:** 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. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиці Thing

thing	0	PRIMARY	1	idThing	A	7	NULL	NULL	BTREE
thing	1	Thing_Producer	1	idProducer	A	3	NULL	NULL	BTREE
thing	1	Thing_Brand	1	idBrand	A	5	NULL	NULL	BTREE
thing	1	Type_Thing_Thing	1	idTypeThing	A	5	NULL	NULL	BTREE

2. Створимо новий індекс для таблиці Order і Ordered\_Thing. У БД є декілька запитів, які здійснюють вибірку даних за датою замовлення (поле order\_date), за кількістю замовлених товарів (поле ordered\_num) тощо. Створення індексів для цих полів повинно оптимізувати виконання запитів.

```
CREATE INDEX order_dateINDX ON `Order` (idOrder, order_date);
```

order	0	PRIMARY	1	idOrder	A	15
order	1	Order_Shop	1	idShop	A	3
order	1	order_dateINDX	1	idOrder	A	15
order	1	order_dateINDX	2	order_date	A	15

```
CREATE INDEX ordered_numINDX ON ordered_thing (idOrderedThing, ordered_num);
```

ordered_thing	0	PRIMARY	1	idOrderedThing	A
ordered_thing	1	Ordered_Thing_Order	1	idOrder	A
ordered_thing	1	Ordered_Thing_Thing	1	idThing	A
ordered_thing	1	ordered_numINDX	1	idOrderedThing	A
ordered_thing	1	ordered_numINDX	2	ordered_num	A

3. Виконаємо аналіз виконання складного запиту з однієї з попередніх робіт використовуючи EXPLAIN та опцію STRAIGHT\_JOIN.

```
EXPLAIN SELECT Shop.idShop, Category.name, SUM(DISTINCT
Thing.price*Ordered_Thing.ordered_num) AS cat_price
FROM ((`Order` INNER JOIN Shop) INNER JOIN (Ordered_Thing INNER JOIN Thing))
INNER JOIN (Type_Thing INNER JOIN Category)
ON `Order`.idOrder = Ordered_Thing.idOrder
AND Thing.idThing = Ordered_Thing.idThing
AND Thing.idTypeThing = Type_Thing.idTypeThing
AND Type_Thing.idCategory = Category.idCategory
AND `Order`.idShop = Shop.idShop
AND `Order`.order_date BETWEEN '2019-05-25 00:00:00' AND '2019-06-26
00:00:00'
WHERE Shop.idShop = 3
GROUP BY Category.name;
```

1	SIMPLE	Shop	NULL	const	PRIMARY	PRIMARY	3	const	1
1	SIMPLE	Ordered_Thing	NULL	ALL	Ordered_Thing__Order,Ordered_Thing__Thing	NULL	NULL	NULL	7
1	SIMPLE	Order	NULL	eq_ref	PRIMARY,Order_Shop,order_dateINDX	PRIMARY	3	shop.Ordered_Thing.idOrder	1
1	SIMPLE	Thing	NULL	eq_ref	PRIMARY,Type_Thing__Thing	PRIMARY	3	shop.Ordered_Thing.idThing	1
1	SIMPLE	Type_Thing	NULL	eq_ref	PRIMARY,Type_Thing__Category	PRIMARY	3	shop.Thing.idTypeThing	1
1	SIMPLE	Category	NULL	eq_ref	PRIMARY,name_UNIQUE	PRIMARY	3	shop.Type_Thing.idCategory	1

```

EXPLAIN SELECT STRAIGHT_JOIN Shop.idShop, Category.name, SUM(DISTINCT
Thing.price*Ordered_Thing.ordered_num) AS cat_price
FROM ((`Order` INNER JOIN Shop) INNER JOIN (Ordered_Thing INNER JOIN Thing))
INNER JOIN (Type_Thing INNER JOIN Category)
ON `Order`.idOrder = Ordered_Thing.idOrder
AND Thing.idThing = Ordered_Thing.idThing
AND Thing.idTypeThing = Type_Thing.idTypeThing
AND Type_Thing.idCategory = Category.idCategory
AND `Order`.idShop = Shop.idShop
AND `Order`.order_date BETWEEN '2019-05-25 00:00:00' AND '2019-06-26
00:00:00'
WHERE Shop.idShop = 3
GROUP BY Category.name;

```

1	SIMPLE	Order	NULL	ref	PRIMARY,Order_Shop,order_dateINDX	Order_Shop	3	const	7
1	SIMPLE	Shop	NULL	const	PRIMARY	PRIMARY	3	const	1
1	SIMPLE	Ordered_Thing	NULL	ref	Ordered_Thing__Order,Ordered_Thing__Thing	Ordered_Thing__Order	3	shop.Order.idOrder	2
1	SIMPLE	Thing	NULL	eq_ref	PRIMARY,Type_Thing__Thing	PRIMARY	3	shop.Ordered_Thing.idThing	1
1	SIMPLE	Type_Thing	NULL	eq_ref	PRIMARY,Type_Thing__Category	PRIMARY	3	shop.Thing.idTypeThing	1
1	SIMPLE	Category	NULL	eq_ref	PRIMARY,name_UNIQUE	PRIMARY	3	shop.Type_Thing.idCategory	1

## Висновок

У даній лабораторній роботі я навчився аналізувати роботу СУБД та оптимізовувати виконання складних запитів на вибірку даних. Виконав аналіз складних запитів за допомогою директиви EXPLAIN, модифікувати найповільніші запити з метою їх пришвидчення.