

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 `Staff` і `Offer`.

```
mysql> SHOW INDEX FROM staff//
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Exp
staff	0	PRIMARY	1	id_staff	A	8	NULL	NULL		BTREE			YES	NULL

1 row in set (0.67 sec)

```
mysql> SHOW INDEX FROM offer//
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
offer	0	PRIMARY	1	id_offer	A	12	NULL	NULL		BTREE			YES
offer	0	PRIMARY	2	Staff_id_staff	A	12	NULL	NULL		BTREE			YES
offer	1	fk_Offer_Staff1	1	Staff_id_staff	A	8	NULL	NULL		BTREE			YES

3 rows in set (0.13 sec)

```
mysql>
```

Створимо новий індекс для таблиці staff I offer. 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> CREATE INDEX staffIDX3 ON staff (id_staff, age);
-> //
Query OK, 0 rows affected (1.29 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show index from staff
-> //
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
staff	0	PRIMARY	1	id_staff	A	8	NULL	NULL		BTREE			YES
staff	1	staffIDX3	1	id_staff	A	8	NULL	NULL		BTREE			YES
staff	1	staffIDX3	2	AGE	A	8	NULL	NULL	YES	BTREE			YES

```
3 rows in set (0.10 sec)
```

```
mysql> CREATE INDEX offerINDX3 ON offer (staff_id_staff , time);
-> //
Query OK, 0 rows affected (0.84 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show index from offer//
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Vis |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| offer | 0 | PRIMARY | 1 | id_offer | A | 12 | NULL | NULL | | BTREE | | | | YES |
| offer | 0 | PRIMARY | 2 | Staff_id_staff | A | 12 | NULL | NULL | | BTREE | | | | YES |
| offer | 1 | offerINDX3 | 1 | Staff_id_staff | A | 8 | NULL | NULL | | BTREE | | | | YES |
| offer | 1 | offerINDX3 | 2 | TIME | A | 12 | NULL | NULL | | BTREE | | | | YES |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.07 sec)

mysql> 
```

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

```
mysql> EXPLAIN SELECT staff.name, (sum(drinks_and_dishes.price)*offer_dishes.count) as salary
-> from ((staff INNER JOIN offer)INNER JOIN offer_dishes)INNER JOIN drinks_and_dishes
-> ON staff.id_staff = offer.staff_id_staff
-> AND offer.id_offer = offer_dishes.Offer_id_offer
-> AND offer_dishes.Drinks_and_dishes_id =Drinks_and_dishes.id
-> GROUP BY staff.name//
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | re |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | staff | NULL | ALL | PRIMARY,staffINDX3 | NULL | NULL | NULL |
| 1 | SIMPLE | offer | NULL | ref | PRIMARY,offerINDX3 | offerINDX3 | 4 | co |
| 1 | SIMPLE | offer_dishes | NULL | ref | fk_offer_dishes_Drinks_and_dishes1,fk_offer_dishes_Offer1 | fk_offer_dishes_Offer1 | 4 | co |
| 1 | SIMPLE | drinks_and_dishes | NULL | eq_ref | PRIMARY | PRIMARY | 4 | co |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.07 sec)
```

```
mysql> EXPLAIN SELECT STRAIGHT_JOIN staff.name, (sum(drinks_and_dishes.price)*offer_dishes.count) as salary
-> from ((staff INNER JOIN offer)INNER JOIN offer_dishes)INNER JOIN drinks_and_dishes
-> ON staff.id_staff = offer.staff_id_staff
-> AND offer.id_offer = offer_dishes.Offer_id_offer
-> AND offer_dishes.Drinks_and_dishes_id =Drinks_and_dishes.id
-> GROUP BY staff.name//
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | staff | NULL | ALL | PRIMARY,staffINDX3 | NULL |
| 1 | SIMPLE | offer | NULL | ref | PRIMARY,offerINDX3 | offerINDX3 |
| 1 | SIMPLE | offer_dishes | NULL | ref | fk_offer_dishes_Drinks_and_dishes1,fk_offer_dishes_Offer1 | fk_offer_dishes_Offer1 |
| 1 | SIMPLE | drinks_and_dishes | NULL | eq_ref | PRIMARY | PRIMARY |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql> 
```

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