

**Міністерство освіти і науки України
Національний університет «Львівська політехніка»**

Кафедра систем штучного інтелекту



**Лабораторна робота №13
на тему:
«Аналіз та оптимізація запитів»
з курсу:
«ОБДЗ»**

Виконала:
ст. гр. КН-210
Ямнюк Аліна
Прийняла:
Мельникова Н.І.

Львів – 2020 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 пришвидшення.

Хід роботи

1. Визначимо усі наявні індекси для таблиць staff та orderoftheclient.

Запити:

```
SHOW INDEX FROM beauty.staff;  
  
SHOW INDEX FROM beauty.orderoftheclient;
```

Результати:

Таблиця orderoftheclient

	Table	Non_unique	Key_name	Seq_in_index	Column_name
1	orderoftheclient	0	PRIMARY	1	id_order
2	orderoftheclient	0	id_order	1	id_order
3	orderoftheclient	1	fk_orderOfTheClient_to_client	1	id_client

Таблиця staff

	Table	Non_unique	Key_name	Seq_in_index	Column_name
1	staff	0	PRIMARY	1	id_staff
2	staff	0	id_staff	1	id_staff

2. Індекс для таблиці orderoftheclient. Створимо індекс за полями ID_order, для того щоб оптимізувати виконання запитів отримання інформації про замовлення за його ID.

```
CREATE INDEX client_order_index ON orderoftheclient(id_order);  
SHOW INDEX FROM beauty.orderoftheclient;
```

Результат:

	Table	Non_unique	Key_name	Seq_in_index	Column_name
1	orderoftheclient	0	PRIMARY	1	id_order
2	orderoftheclient	0	id_order	1	id_order
3	orderoftheclient	1	fk_orderOfTheClient_to_client	1	id_client
4	orderoftheclient	1	client_order_index	1	id_order

Індекс для таблиці staff. Створимо індекс за полями ID_staff, для швидкого отримання інформації про майстра.

Результат:

	Table	Non_unique	Key_name	Seq_in_index	Column_name
1	staff	0	PRIMARY	1	id_staff
2	staff	0	id_staff	1	id_staff
3	staff	0	staff_index	1	id_staff

3. Тепер виконаємо аналіз виконання складного запиту з однієї з попередніх робіт за допомогою директиви EXPLAIN.

Запит:

```
EXPLAIN SELECT STRAIGHT_JOIN concat_ws(' ', c.first_name, c.last_name) AS Client, c.telephone,
oc.dateOfTheOrder AS date , oc.timeOfTheOrder AS time, sr.name_service AS service,
concat_ws(' ', s.first_name, s.last_name) AS master
FROM beauty.order AS o
INNER JOIN beauty.orderoftheclient AS oc ON o.id_order = oc.id_order
INNER JOIN beauty.isClient AS c ON oc.id_client = c.id_client
INNER JOIN beauty.staff AS s ON o.id_staff = s.id_staff
INNER JOIN beauty.service AS sr ON o.id_service = sr.id_service;
```

Результат:

id	select_type	table	partitions	type	possible_keys
1	1 SIMPLE	o	<null>	ALL	fk_order_to_service, fk_order_to_staff, f...
2	1 SIMPLE	oc	<null>	ALL	PRIMARY, id_order, fk_orderOfTheClient_to...
3	1 SIMPLE	c	<null>	eq_ref	PRIMARY, id_client
4	1 SIMPLE	s	<null>	eq_ref	PRIMARY, id_staff, staff_index
5	1 SIMPLE	sr	<null>	eq_ref	PRIMARY, id_service
key	key_len	ref	rows	filtered	Extra
<null>	<null>	<null>	2	100	<null>
<null>	<null>	<null>	3	33.33	Using where; Using join buffer (Block
PRIMARY	4	beauty.oc.id_client	1	100	<null>
PRIMARY	4	beauty.o.id_staff	1	100	<null>
PRIMARY	4	beauty.o.id_service	1	100	<null>

Висновок: на цій лабораторній роботі я навчилась аналізувати роботу СУБД та оптимізовувати виконання складних запитів на вибірку даних.