

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота
з дисципліни
«Організація баз даних та знань»

Виконав:

студент групи КН-209

Гавришків Олексій

Викладач:

Мельникова Н. І.

Львів – 2020 р.

Лабораторна робота № 13

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. Переглянемо індекси таблиці user:

```
SHOW INDEX FROM user;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
user	0	PRIMARY	1	id	A	20	NULL	NULL		BTREE		
user	0	email_UNIQUE	1	email	A	20	NULL	NULL	YES	BTREE		
user	0	username_UNIQUE	1	username	A	20	NULL	NULL	YES	BTREE		
user	0	api_key_UNIQUE	1	api_key	A	2	NULL	NULL	YES	BTREE		

2. Створимо індекс для id, username і date_registration, що має оптимізувати виконання запитів.

```
CREATE INDEX userINDX ON user (id,username,date_registration);  
SHOW INDEX FROM user;
```

#	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
1	user	0	PRIMARY	1	id	A	20	NULL	NULL		BTREE
2	user	0	email_UNIQUE	1	email	A	20	NULL	NULL	YES	BTREE
3	user	0	username_UNIQUE	1	username	A	20	NULL	NULL	YES	BTREE
4	user	0	api_key_UNIQUE	1	api_key	A	2	NULL	NULL	YES	BTREE
5	user	1	userINDX	1	id	A	20	NULL	NULL		BTREE
6	user	1	userINDX	2	username	A	20	NULL	NULL	YES	BTREE
7	user	1	userINDX	3	date_registration	A	20	NULL	NULL	YES	BTREE

3. Виконаємо аналіз виконання складного запиту, використовуючи EXPLAIN з опцією STRAIGHT_JOIN.

```
EXPLAIN SELECT STRAIGHT_JOIN count(user.id) AS participants_count,  
project.name AS project  
FROM ((project_participant INNER JOIN user)  
INNER JOIN project  
ON project.id = project_participant.project_id  
AND user.id = project_participant.user_id)  
GROUP BY project.name  
ORDER BY participants_count DESC;
```

#	id	select_type	table	type	possible_keys	key	key_len	ref	rows
1	1	SIMPLE	project_participant	index	unique_proj_part,fk_user_i...	unique_proj_part	10	NULL	19
2	1	SIMPLE	user	eq_ref	PRIMARY,userINDX	PRIMARY	4	docker_example.project_participant...	1
3	1	SIMPLE	project	eq_ref	PRIMARY	PRIMARY	4	docker_example.project_participant...	1

Висновок: виконуючи цю лабораторну роботу я навчився аналізувати роботу запитів в MySQL і оптимізував запити за допомогою індексів.