

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

Кафедра Системи штучного інтелекту

Лабораторна робота №13
з дисципліни
«Організація баз даних та знань»

Виконала:
студентка групи КН-208
Пона Олена
Викладач:
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.

Хід роботи

1. Визначити індекси таблиці.
2. Створити додаткові індекси для таблиці.
3. Дослідити процес виконання запитів за допомогою EXPLAIN.

Виконання

1. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць Employee і Vacancie.

```
mysql> show index from employee;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
employee	0	PRIMARY	1	employee_id	A	11	NULL	NULL		BTREE			YES	NULL

1 row in set (0.09 sec)

```
mysql> show index from vacancie;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
vacancie	0	PRIMARY	1	vacancies_id	A	13	NULL	NULL		BTREE			YES	NULL
vacancie	1	employer_roles	1	employer_id	A	9	NULL	NULL		BTREE			YES	NULL

2 rows in set (0.01 sec)

2. Створимо новий індекс для таблиці Employee і Secretary, що повинно оптимізувати виконання запитів.

```
use employmentcenter;
create index employeeINDX on employee(employee_id,
person_name);
show index from employee;
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
▶	employee	0	PRIMARY	1	employee_id	A	11	NULL	NULL		BTREE			YES	NULL
	employee	1	employeeINDX	1	employee_id	A	11	NULL	NULL		BTREE			YES	NULL
	employee	1	employeeINDX	2	person_name	A	11	NULL	NULL		BTREE			YES	NULL

```
use employmentcenter;
create index secretaryINDX on secretary(employee_id,
start_time);
show index from secretary;
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
►	secretary	0	PRIMARY	1	secretary_id	A	12	NULL	NULL		BTREE			YES	NULL
	secretary	1	vacancies_information	1	vacancies_id	A	12	NULL	NULL		BTREE			YES	NULL
	secretary	1	secretaryINDEX	1	employee_id	A	7	NULL	NULL		BTREE			YES	NULL
	secretary	1	secretaryINDEX	2	start_time	A	10	NULL	NULL		BTREE			YES	NULL

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

```
use employmentcenter;
explain select result as res ,
count(secretary.secretary_id) as amount from employee
inner join secretary
on employee.employee_id = secretary.employee_id
where secretary.start_time > '07:00:00'
group by res;
```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	secretary	NULL	ALL	secretaryINDEX	NULL	NULL	NULL	12	33.33	Using where; Using temporary
	1	SIMPLE	employee	NULL	eq_ref	PRIMARY,employeeINDEX	PRIMARY	4	employmentcenter.secretary.employee_id	1	100.00	Using index

```
use employmentcenter;
explain select straight_join result as res ,
count(secretary.secretary_id) as amount from employee
inner join secretary
on employee.employee_id = secretary.employee_id
where secretary.start_time > '07:00:00'
group by res;
```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	employee	NULL	index	PRIMARY,employeeINDEX	employeeINDEX	400	NULL	11	100.00	Using index; Using temporary
	1	SIMPLE	secretary	NULL	ALL	secretaryINDEX	NULL	NULL	NULL	12	8.33	Using where; Using join buffer (Block Nested Lo...

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