

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

**Виконав:**  
Студент групи КН-209  
Коваль Назар

**Викладач:**  
Мельникова Н.І.

## 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. Додамо індекс унікальності до таблиці location та перевіримо час виконання запитів вставки до і після цього. Також за допомогою директиви show index переглянемо наявні у цій таблиці індекси.

```
SHOW INDEX FROM location;

INSERT INTO location(name)
VALUES ('Aquapark'),
       ('YogaMaster'),
       ('Struyskiy Park');

TRUNCATE location;
CREATE UNIQUE INDEX uq_l_name ON location(name);
SHOW INDEX FROM location;

INSERT INTO location(name)
VALUES ('Aquapark'),
       ('YogaMaster'),
       ('Struyskiy Park');
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
location	0	PRIMARY	1	id	A	0	NULL	NULL		BTREE

  

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
location	0	PRIMARY	1	id	A	0	NULL	NULL		BTREE
location	0	uq_l_name	1	name	A	0	NULL	NULL		BTREE

### Час виконання запитів:

✓	817	18:02:57	INSERT INTO location(name) VALUES ('Aquapark'), ('YogaMaster'), ('Struyskiy Park')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.156 sec
✓	818	18:02:57	TRUNCATE location	0 row(s) affected	0.938 sec
✓	819	18:02:58	CREATE UNIQUE INDEX uq_l_name ON location(name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.515 sec
✓	820	18:02:59	SHOW INDEX FROM location	2 row(s) returned	0.047 sec / 0.015 sec
✓	821	18:02:59	INSERT INTO location(name) VALUES ('Aquapark'), ('YogaMaster'), ('Struyskiy Park')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.188 sec

Як можна побачити, час виконання не покращився, а навіть погіршився. Це пояснюється дуже маленькою кількістю даних, адже при малій кількості робота індексів не помітна.

2. Додамо індекс до таблиці sport\_type та перевіримо час виконання запитів вставки до і після цього. На цей раз будемо здійснювати вставку більшої кількості даних. Також за допомогою директиви show index переглянемо наявні у цій таблиці індекси.

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
sport_type	0	PRIMARY	1	id	A	0	NULL	NULL		BTREE

  

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
sport_type	0	PRIMARY	1	id	A	0	NULL	NULL		BTREE
sport_type	1	sp_name	1	name	A	0	NULL	NULL		BTREE

```

INSERT INTO sport_type(name)
VALUES ('Running'),
      ('Swimming'),
      ('Football'),
      ('Yoga'),
      ('Basketball'),
      ('Volleyball'),
      ('Tennis'),
      ('Racing'),
      ('Baseball'),
      ('1'),
      ('2'),
      ('3'),
      ('4'),
      ('5'),
      ('6'),
      ('7'),
      ('8'),

```

Час виконання запитів:

✓	824	18:09:57	INSERT INTO sport_type(name) VALUES ('Running'), ('Swimming'), ('Football'), ('Yoga'), ('Basketball'), ('Volleyball'), ('Tennis'), ('Racing'), ('Baseball'), ('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'), ('8');	22 row(s) affected Records: 22 Duplicates: 0 Warnings: 0	0.140 sec
✓	825	18:09:57	CREATE INDEX sp_name ON sport_type(name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.875 sec
✓	826	18:09:58	TRUNCATE sport_type	0 row(s) affected	2.282 sec
✓	827	18:10:00	SHOW INDEX FROM sport_type	2 row(s) returned	0.047 sec / 0.000 sec
✓	828	18:10:00	INSERT INTO sport_type(name) VALUES ('Running'), ('Swimming'), ('Football'), ('Yoga'), ('Basketball'), ('Volleyball'), ('Tennis'), ('Racing'), ('Baseball'), ('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'), ('8');	22 row(s) affected Records: 22 Duplicates: 0 Warnings: 0	0.109 sec

Тепер можемо помітити, що трішки збільшивши кількість даних для вставки, ми можемо побачити роботу доданого індекса.

- Використавши директиву Explain проаналізуємо поля таблиці sport\_type на наявність індексів.

```

EXPLAIN SELECT sport_type.name as spn, COUNT(*) FROM sport_type
LEFT JOIN location_sport_type ON location_sport_type.sport_type_id = sport_type.id
GROUP BY spn;

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	sport_type	NULL	index	sp_name	sp_name	130	NULL	22	100.00	Using index
1	SIMPLE	location_sport_type	NULL	ref	fk_location_sport_type_sport_type	fk_location_sport_type_sport_type	4	nulp_db.sport_type.id	1	100.00	Using index

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