

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



**Звіт до лабораторної роботи №13
з дисципліни
“ОБДЗ”**

Виконав:
ст. гр. КН-211
Ткачук Орест

Викладач:
Якимишин Х.М.

Львів – 2020

Лабораторна робота №13

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. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць.

```
mysql> show index from theater;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
theater	0	PRIMARY	1	id_theater	A	6	NULL	NULL	NULL	BTREE			YES

1 row in set (0.01 sec)

```
mysql> show index from hall;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
hall	0	PRIMARY	1	id_hall	A	10	NULL	NULL	NULL	BTREE			YES
hall	1	Hall_fk0	1	theaterID	A	6	NULL	NULL	NULL	BTREE			YES

2 rows in set (0.01 sec)

```
mysql> show index from seance;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
seance	0	PRIMARY	1	id_seance	A	1	NULL	NULL	NULL	BTREE			YES
seance	1	Seance_fk0	1	hallID	A	1	NULL	NULL	NULL	BTREE			YES
seance	1	Seance_fk1	1	repertoireID	A	1	NULL	NULL	NULL	BTREE			YES

3 rows in set (0.01 sec)

```
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	3	NULL	NULL	NULL	BTREE			YES
staff	1	Staff_fk0	1	theaterID	A	1	NULL	NULL	NULL	BTREE			YES

2 rows in set (0.01 sec)

```
mysql> show index from role;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
role	0	PRIMARY	1	id_role	A	15	NULL	NULL	NULL	BTREE			YES
role	1	Role_fk0	1	repertoireID	A	5	NULL	NULL	NULL	BTREE			YES

2 rows in set (0.01 sec)

2. Порівняємо ефективність вибірки з директивою STRAIGHT_JOIN та без неї. Використовуючи директиву STRAIGHT_JOIN

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	theater	NULL	ALL	PRIMARY	NULL	NULL	NULL	6	100.00	Using temporary; Using filesort
1	SIMPLE	staff	NULL	ref	PRIMARY,Staff_fk0	Staff_fk0	4	lab13.theater.id_theater	1	100.00	NULL
1	SIMPLE	hall	NULL	ref	PRIMARY,Hall_fk0	Hall_fk0	4	lab13.theater.id_theater	1	100.00	NULL
1	SIMPLE	seance	NULL	ref	PRIMARY,Seance_fk0	Seance_fk0	4	lab13.hall.id_hall	1	33.33	Using where
1	SIMPLE	actorrole	NULL	ref	ActorRole_fk0,ActorRole_fk1,ActorRole_fk2	ActorRole_fk0	4	lab13.seance.id_seance	1	16.67	Using where
1	SIMPLE	role	NULL	eq_ref	PRIMARY	PRIMARY	4	lab13.actorrole.roleID	1	100.00	NULL

6 rows in set, 1 warning (0.00 sec)

Без директиви STRAIGHT JOIN

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	actorrole	NULL	ALL	ActorRole_fk0,ActorRole_fk1,ActorRole_fk2	NULL	NULL	NULL	6	100.00	Using temporary; Using filesort
1	SIMPLE	seance	NULL	eq_ref	PRIMARY,Seance_fk0	PRIMARY	4	lab13.actorrole.seanceID	1	33.33	Using where
1	SIMPLE	hall	NULL	eq_ref	PRIMARY,Hall_fk0	PRIMARY	4	lab13.seance.hallID	1	100.00	NULL
1	SIMPLE	theater	NULL	eq_ref	PRIMARY	PRIMARY	4	lab13.hall.theaterID	1	100.00	NULL
1	SIMPLE	staff	NULL	eq_ref	PRIMARY,Staff_fk0	PRIMARY	4	lab13.actorrole.staffID	1	7.14	Using where
1	SIMPLE	role	NULL	eq_ref	PRIMARY	PRIMARY	4	lab13.actorrole.roleID	1	100.00	NULL

6 rows in set, 1 warning (0.00 sec)

Де rows відображає кількість проходжень по рядкам таблиці, а filter ефективність селекту.

3. Додамо кілька індексів для полів, що беруть участь у вибірці. Створення індексів для цих полів повинно оптимізувати виконання запитів. Порівняємо ефективність вибірки з директивою `STRAIGHT_JOIN` та без неї.

```
mysql> show index from theater;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
theater	0	PRIMARY	1	id_theater	A	6	NULL	NULL		BTREE			YES
theater	0	theaterINDX2	1	id_theater	A	6	NULL	NULL		BTREE			YES
theater	0	theaterINDX2	2	theater_name	A	6	NULL	NULL		BTREE			YES
theater	1	theaterINDX	1	id_theater	A	6	NULL	NULL		BTREE			YES
theater	1	theaterINDX	2	address	A	6	NULL	NULL		BTREE			YES

5 rows in set (0.02 sec)

```
mysql> show index from hall;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
hall	0	PRIMARY	1	id_hall	A	10	NULL	NULL		BTREE			YES
hall	0	hallINDX	1	id_hall	A	10	NULL	NULL		BTREE			YES
hall	0	hallINDX	2	hall_name	A	10	NULL	NULL		BTREE			YES
hall	0	hallINDX2	1	id_hall	A	10	NULL	NULL		BTREE			YES
hall	0	hallINDX2	2	capacity	A	10	NULL	NULL		BTREE			YES
hall	1	Hall_fk0	1	theaterID	A	6	NULL	NULL		BTREE			YES

6 rows in set (0.01 sec)

```
mysql> show index from seance;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
seance	0	PRIMARY	1	id_seance	A	1	NULL	NULL		BTREE			YES
seance	1	Seance_fk0	1	hallID	A	1	NULL	NULL		BTREE			YES
seance	1	Seance_fk1	1	repertoireID	A	1	NULL	NULL		BTREE			YES
seance	1	seanceINDX	1	id_seance	A	0	NULL	NULL		BTREE			YES
seance	1	seanceINDX	2	date	A	0	NULL	NULL		BTREE			YES

5 rows in set (0.01 sec)

```
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	3	NULL	NULL		BTREE			YES
staff	0	staffINDX	1	id_staff	A	4	NULL	NULL		BTREE			YES
staff	0	staffINDX	2	staff_name	A	4	NULL	NULL		BTREE			YES
staff	1	Staff_fk0	1	theaterID	A	1	NULL	NULL		BTREE			YES

4 rows in set (0.01 sec)

Використовуючи директиву `STRAIGHT_JOIN`

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	theater	NULL	ALL	PRIMARY, theaterINDX2, theaterINDX	NULL	NULL	NULL	6	100.00	Using temporary; Using filesort
1	SIMPLE	staff	NULL	ALL	PRIMARY, staffINDX, Staff_fk0	NULL	NULL	NULL	4	100.00	Using where; Using join buffer (Block Nested Loop)
1	SIMPLE	hall	NULL	ref	PRIMARY, hallINDX, hallINDX2, Hall_fk0	Hall_fk0	4	lab13.theater.id_theater	1	100.00	NULL
1	SIMPLE	seance	NULL	ALL	PRIMARY, Seance_fk0, seanceINDX	NULL	NULL	NULL	1	100.00	Using where; Using join buffer (Block Nested Loop)
1	SIMPLE	actorrole	NULL	ALL	ActorRole_fk0, ActorRole_fk1, ActorRole_fk2	NULL	NULL	NULL	1	100.00	Using where; Using join buffer (Block Nested Loop)
1	SIMPLE	role	NULL	eq_ref	PRIMARY	PRIMARY	4	lab13.actorrole.roleID	1	100.00	NULL

6 rows in set, 1 warning (0.00 sec)

Без директиви `STRAIGHT_JOIN`

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	seance	NULL	ALL	PRIMARY, Seance_fk0, seanceINDX	NULL	NULL	NULL	1	100.00	Using where; Using filesort
1	SIMPLE	actorrole	NULL	ref	ActorRole_fk0, ActorRole_fk1, ActorRole_fk2	ActorRole_fk0	4	lab13.seance.id_seance	1	100.00	NULL
1	SIMPLE	staff	NULL	eq_ref	PRIMARY, staffINDX, Staff_fk0	PRIMARY	4	lab13.actorrole.staffID	1	100.00	NULL
1	SIMPLE	theater	NULL	eq_ref	PRIMARY, theaterINDX2, theaterINDX	PRIMARY	4	lab13.staff.theaterID	1	100.00	NULL
1	SIMPLE	hall	NULL	eq_ref	PRIMARY, hallINDX, hallINDX2, Hall_fk0	PRIMARY	4	lab13.seance.hallID	1	16.67	Using where
1	SIMPLE	role	NULL	eq_ref	PRIMARY	PRIMARY	4	lab13.actorrole.roleID	1	100.00	NULL

6 rows in set, 1 warning (0.00 sec)

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