

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра «Системи штучного інтелекту»



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

Виконав:

студент

групи КН-211

Пучак Віталія

Перевірила:

Якимишин Х.М.

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 пришвидчення.

Завдання на лабораторну роботу.

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

Хід роботи.

1. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць booking, customer, employee.

```
show index from booking;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
booking	0	PRIMARY	1	bookingID	A	6	NULL	NULL		BTREE			YES	NULL
booking	1	tour_booking	1	tour_ID	A	1	NULL	NULL		BTREE			YES	NULL
booking	1	customer_booking	1	customerID	A	1	NULL	NULL		BTREE			YES	NULL
booking	1	employee_booking	1	employeeID	A	1	NULL	NULL		BTREE			YES	NULL

```
show index from customer;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
customer	0	PRIMARY	1	customerID	A	4	NULL	NULL		BTREE			YES	NULL

```
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	employeeID	A	2	NULL	NULL		BTREE			YES	NULL

2. Створимо індекс для поля booking_time

```
create index booking_time_idx on booking (booking_time);
```

Створимо індекс для поля name в таблиці customer

```
create index customer_name_idx on customer (name);
```

Створимо індекс для поля name в таблиці employee

```
create index employee_name_idx on employee (name);
```

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

```
explain select employee.name, count(employee.employeeID) as 'serves customer(times)' from (customer join employee)
join booking on customer.name = 'Su'
and customer.customerID = booking.customerID
and employee.employeeID = booking.employeeID
where booking.booking_time between '2020-01-01 10:10:11' and '2020-01-01 10:10:20'
group by employee.employeeID;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	customer	NULL	ref	PRIMARY, customer_name_idx	customer_name_idx	202	const	1	100.00	Using index; Using temporary
1	SIMPLE	booking	NULL	ref	customer_booking, employee_booking, booking_t...	customer_booking	4	lab2.customer.customerID	6	50.00	Using where
1	SIMPLE	employee	NULL	eq_ref	PRIMARY, employee_name_idx	PRIMARY	4	lab2.booking.employeeID	1	100.00	NULL

```
explain select straight_join employee.name, count(employee.employeeID) as 'serves customer(times)' from (customer join employee)
join booking on customer.name = 'Su'
and customer.customerID = booking.customerID
and employee.employeeID = booking.employeeID
where booking.booking_time between '2020-01-01 10:10:11' and '2020-01-01 10:10:20'
group by employee.employeeID;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	customer	NULL	ref	PRIMARY, customer_name_idx	customer_name_idx	202	const	1	100.00	Using index; Using temporary
1	SIMPLE	employee	NULL	index	PRIMARY, employee_name_idx	employee_name_idx	202	NULL	2	100.00	Using index; Using join buffer (Block Nested Loop)
1	SIMPLE	booking	NULL	ref	customer_booking, employee_booking, booking_t...	customer_booking	4	lab2.customer.customerID	6	50.00	Using where

```
explain select straight_join employee.name, count(employee.employeeID) as 'serves customer(times)' from (customer join employee)
join booking on customer.name = 'Su'

where booking.booking_time between '2020-01-01 10:10:11' and '2020-01-01 10:10:20'
group by employee.employeeID;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	customer	NULL	ref	customer_name_idx	customer_name_idx	202	const	1	100.00	Using index; Using temporary
1	SIMPLE	employee	NULL	index	PRIMARY, employee_name_idx	employee_name_idx	202	NULL	2	100.00	Using index; Using join buffer (Block Nested Loop)
1	SIMPLE	booking	NULL	range	booking_time_idx	booking_time_idx	5	NULL	3	100.00	Using where; Using index; Using join buffer (Blo...

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.