

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота № 13

з дисципліни
«Організація баз даних та знань»

Виконав:
студент групи КН-208
Мокрик Ярослав
Викладач:
Мельникова Н. І.

Львів - 2020 р.

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 пришвидшення.

Хід роботи

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

1. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць Order і Car.

```
SHOW INDEX FROM `order`;
```

```
SHOW INDEX FROM `car`;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
order	0	PRIMARY	1	id	A	1				BTREE			YES	
order	1	Order_fk0	1	destination_id	A	1				BTREE			YES	
order	1	Order_fk1	1	product_id	A	1				BTREE			YES	
order	1	Order_fk2	1	client_id	A	1				BTREE			YES	
order	1	Order_fk3	1	car_id	A	1				BTREE			YES	

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
car	0	PRIMARY	1	id	A	0				BTREE			YES	
car	0	type	1	type	A	0				BTREE			YES	

2. Створимо новий індекс для таблиці Order і Car. У БД є декілька запитів, які здійснюють вибірку даних за замовленням товаром (Order.product_id) і номером вагону (Car.id). Створення індексів для цих полів повинно оптимізувати виконання запитів.

```
CREATE INDEX order_product_idx ON `order` (id, product_id);
```

```
SHOW INDEX FROM `order`;
```

```
CREATE INDEX car_type_idx ON `car` (id, type);
```

```
SHOW INDEX FROM `car`;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
order	0	PRIMARY	1	id	A	1				BTREE			YES	
order	1	Order_fk0	1	destination_id	A	1				BTREE			YES	
order	1	Order_fk1	1	product_id	A	1				BTREE			YES	
order	1	Order_fk2	1	client_id	A	1				BTREE			YES	
order	1	Order_fk3	1	car_id	A	1				BTREE			YES	
order	1	order_product_idx	1	id	A	1				BTREE			YES	
order	1	order_product_idx	2	product_id	A	1				BTREE			YES	

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
car	0	PRIMARY	1	id	A	0				BTREE			YES	
car	0	type	1	type	A	0				BTREE			YES	
car	1	car_type_idx	1	id	A	0				BTREE			YES	
car	1	car_type_idx	2	type	A	0				BTREE			YES	

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

```
EXPLAIN SELECT `order`.id AS `Order`, product.`type` AS `Product`, `order`.product_volume AS `Volume`,
car.`type` AS `Car Type`, train.`name` AS `Train`
FROM `order` INNER JOIN `product` INNER JOIN `car` INNER JOIN `train_car_table` INNER JOIN `train`
ON `order`.product_id=product.id AND `order`.car_id=car.id AND `train_car_table`.car_id=car.id AND `train`.id=`train_car_table`.train_id;
```

```
EXPLAIN SELECT STRAIGHT_JOIN `order`.id AS `Order`, product.`type` AS `Product`, `order`.product_volume AS `Volume`,
car.`type` AS `Car Type`, train.`name` AS `Train`
FROM `order` INNER JOIN `product` INNER JOIN `car` INNER JOIN `train_car_table` INNER JOIN `train`
ON `order`.product_id=product.id AND `order`.car_id=car.id AND `train_car_table`.car_id=car.id AND `train`.id=`train_car_table`.train_id;
```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	order	<small>NULL</small>	ALL	Order_fk1,Order_fk3	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	1	100.00	<small>NULL</small>
	1	SIMPLE	car	<small>NULL</small>	eq_ref	PRIMARY,car_type_indx	PRIMARY	4	trains.order.car_id	1	100.00	<small>NULL</small>
	1	SIMPLE	train_car_table	<small>NULL</small>	ref	Train_Car_Table_fk0,Train_Car_Table_fk1	Train_Car_Table_fk1	4	trains.order.car_id	1	100.00	<small>NULL</small>
	1	SIMPLE	train	<small>NULL</small>	eq_ref	PRIMARY	PRIMARY	4	trains.train_car_table.train_id	1	100.00	<small>NULL</small>
	1	SIMPLE	product	<small>NULL</small>	eq_ref	PRIMARY	PRIMARY	4	trains.order.product_id	1	100.00	<small>NULL</small>

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	order	<small>NULL</small>	ALL	Order_fk1,Order_fk3	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	1	100.00	<small>NULL</small>
	1	SIMPLE	product	<small>NULL</small>	eq_ref	PRIMARY	PRIMARY	4	trains.order.product_id	1	100.00	<small>NULL</small>
	1	SIMPLE	car	<small>NULL</small>	eq_ref	PRIMARY,car_type_indx	PRIMARY	4	trains.order.car_id	1	100.00	<small>NULL</small>
	1	SIMPLE	train_car_table	<small>NULL</small>	ref	Train_Car_Table_fk0,Train_Car_Table_fk1	Train_Car_Table_fk1	4	trains.order.car_id	1	100.00	<small>NULL</small>
	1	SIMPLE	train	<small>NULL</small>	eq_ref	PRIMARY	PRIMARY	4	trains.train_car_table.train_id	1	100.00	<small>NULL</small>

Висновок

На цій лабораторній роботі я навчився аналізувати роботу СУБД та оптимізовувати виконання складних запитів на вибірку даних, виконав аналіз складних запитів за допомогою директиви EXPLAIN.