

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту



ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №13
З дисципліни: “Організація баз даних та знань”

Виконав:

студент групи КН-211

Лазурко Олександр

Викладач:

Якимишин Х.М.

Львів 2020

“Аналіз та оптимізація запитів”

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 пришвидшення.

Директива *EXPLAIN* дозволяє визначити поля таблиці, для яких варто створити додаткові індекси, щоб пришвидшити вибірку даних.

У цій лабораторній ми порівнюємо значення викликавши перший раз *EXPLAIN* без опції *STRAIGHT_JOIN* і другий раз з. Робимо ми це, оскільки у першому варіанті програма самостійно об'єднює таблицю, а у другому так, який ми задали.

Хід роботи

Виконаємо аналіз виконання запиту за допомогою *EXPLAIN* та *EXPLAIN* з опцією *STRAIGHT_JOIN*.

explain select books.book_name, publishing.name

from (books inner join publication on books.book_id = publication.book_id)

inner join publishing on publication.publishing_id = publishing.publishing_id where book_name like "%a%" order by book_name;

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	publication	<small>NULL</small>	index	oo,ol,oj	oj	8	<small>NULL</small>	5	100.00	Using index; Using temporary; Using filesort
	1	SIMPLE	publishing	<small>NULL</small>	eq_ref	PRIMARY	PRIMARY	4	db.publication.publishing_id	1	100.00	<small>NULL</small>
	1	SIMPLE	books	<small>NULL</small>	eq_ref	PRIMARY	PRIMARY	4	db.publication.book_id	1	83.33	Using where

explain select straight_join books.book_name, publishing.name

from (books inner join publication on books.book_id)

inner join publishing on publication.publishing_id = publishing.publishing_id where book_name like "%a%" order by book_name;

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	books	<small>NULL</small>	ALL	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	6	69.44	Using where; Using temporary; Using filesort
	1	SIMPLE	publication	<small>NULL</small>	index	ol,oj	ol	4	<small>NULL</small>	5	100.00	Using index; Using join buffer (Block Nested Loop)
	1	SIMPLE	publishing	<small>NULL</small>	eq_ref	PRIMARY	PRIMARY	4	db.publication.publishing_id	1	100.00	<small>NULL</small>

Оптимізувати цей запит неможливо, оскільки для умови *like* “%a%” нам потрібно переглянути усі записи в таблиці. У стовпці *type* ми можемо побачити як MySQL пов’язує наші таблиці.

Найкращим вважається *eq_ref* - використовує індекси типу *PRIMARY KEY* чи *UNIQUE INDEX*.

ref - використовує звичайні індекси, та *all* - використовується сканування всієї таблиці для знаходження рядків(вважається найгіршим типом зв’язку)

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.