Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп’ютерних наук та інформаційних технологій

Кафедра «Системи штучного інтелекту»

ЛАБОРАТОРНА РОБОТА №13

З предмету: «Організація баз даних та знань**»**

Виконав студент
групи КН-208
Левицький Б.Р
Прийняла:
Мельникова Н. І.

Львів-2020

Аналіз та оптимізація запитів

Мета роботи: Розробити SQL запити, які моделюють роботу тригерів: каскадне знищення, зміна та доповнення записів у зв’язаних таблицях.

Хід роботи

1. Визначити індекси таблиці.

Спочатку потрібно за допомогою дерективи **EXPLAIN** переглянути, які таблиці використовують індекси а які ні. Після цього проаналізувати для якої таблиці та якого поля доречно створити індекс.

EXPLAIN SELECT unemployed.id, unemployed.name, unemployed.surname, Age(unemployed.birth_date), job.job

FROM unemployed

INNER JOIN job **ON** job.id = unemployed.job_id

INNER JOIN worker **ON** unemployed.worker_id = worker.id

INNER JOIN affiliate **ON** worker.affiliate_id = affiliate.id

INNER JOIN market_place **ON** affiliate.market_place_id = market_place.id

WHERE market_place.location **LIKE** concat("%",@city,"%") **AND** job.job LIKE concat("%",@job,"%");

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	affiliate	NULL	index	PRIMARY,market_place_id	market_place_id	4	NULL	8	100.00	Using index
1	SIMPLE	market_place	NULL	eq_ref	PRIMARY	PRIMARY	4	employment_center.affiliate.market_place_id	1	100.00	Using where
1	SIMPLE	worker	NULL	ref	PRIMARY,affiliate_id	affiliate_id	4	employment_center.affiliate.id	1	100.00	Using index
1	SIMPLE	unemployed	NULL	ref	worker_id,job_id	worker_id	5	employment_center.worker.id	1	100.00	NULL
1	SIMPLE	JOB	NULL	eq_ref	PRIMARY	PRIMARY	4	employment_center.unemployed.job_id	1	100.00	Using where

Оскільки умовою WHERE ми виконуємо пошук схожиш записів, тому доречно буде створити індекси для такий полів: **market_place.location** і **job.job** для оптимізації пошуку.

Переглянемо уже існуючі індекси для 2 таблиць: market_place і job

SHOW INDEX FROM job;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
job	0	PRIMARY	1	id	A	11	NULL	NULL		BTREE			YES	NULL

SHOW INDEX FROM market_place**;**

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
market_place	0	PRIMARY	1	id	A	9	NULL	NULL		BTREE			YES	NULL

2. Створити додаткові індекси для таблиці.

Як було згадано вище, створимо 2 індекси для такий полів: **market_place.location** і **job.job** для оптимізації пошуку.

CREATE INDEX look_for_place **ON** market_place(location);

Переглянемо чи індекс добавився

SHOW INDEX FROM market_place;

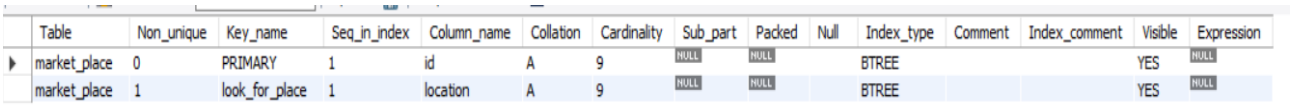


Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
market_place	0	PRIMARY	1	id	A	9	NULL	NULL		BTREE			YES	NULL
market_place	1	look_for_place	1	location	A	9	NULL	NULL		BTREE			YES	NULL

CREATE INDEX look_for_job **ON** job(job);

Переглянемо чи індекс добавився

SHOW INDEX FROM job;

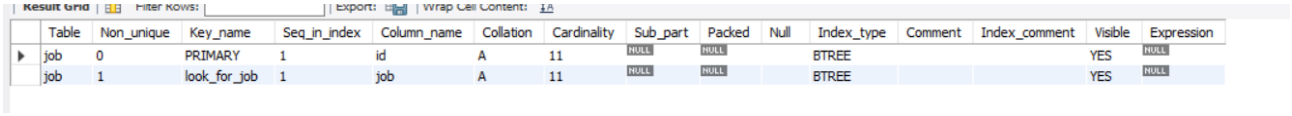


Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
job	0	PRIMARY	1	id	A	11	NULL	NULL		BTREE			YES	NULL
job	1	look_for_job	1	job	A	11	NULL	NULL		BTREE			YES	NULL

3.Дослідити процес виконання запитів за допомогою EXPLAIN.

Виконуємо запит за допомогою директиви **EXPLAIN** та **STRAIGHT_JOIN** (якщо оптимізатор вибирає не найкращу послідовність з’єднання таблиць)

для таблиць **interdiction, staff, client,** який буде виводити опис замовлення, ім’я клієнта та ім’я працівника.

Виконуємо спочатку запит без застосування додаткових індексів

EXPLAIN SELECT unemployed.id, unemployed.name, unemployed.surname, Age(unemployed.birth_date), job.job

FROM unemployed

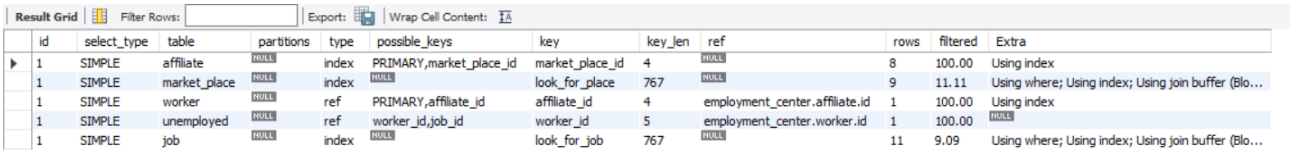
INNER JOIN job **USE INDEX**(look_for_job) **ON** job.id = unemployed.job_id

INNER JOIN worker **ON** unemployed.worker_id = worker.id

INNER JOIN affiliate **ON** worker.affiliate_id = affiliate.id

INNER JOIN market_place **USE INDEX**(look_for_place) **ON** affiliate.market_place_id = market_place.id

WHERE market_place.location **LIKE** concat("%",@city,"%") **AND** job.job **LIKE** concat("%",@job,"%");



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	affiliate	NULL	index	PRIMARY,market_place_id	market_place_id	4	NULL	8	100.00	Using index
1	SIMPLE	market_place	NULL	index	NULL	look_for_place	767	NULL	9	11.11	Using where; Using index; Using join buffer (Blo...
1	SIMPLE	worker	NULL	ref	PRIMARY,affiliate_id	affiliate_id	4	employment_center.affiliate.id	1	100.00	Using index
1	SIMPLE	unemployed	NULL	ref	worker_id,job_id	worker_id	5	employment_center.worker.id	1	100.00	NULL
1	SIMPLE	job	NULL	index	NULL	look_for_job	767	NULL	11	9.09	Using where; Using index; Using join buffer (Blo...

Порівнюючи початковий запит **EXPLAIN** з останнім нашим запитом, можна побачити, що використовуються наші новостворені індекси, які мають оптимізувати пошук в таблиці **job** по полю **job** та в таблиці **market_place** по полю **location**.

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.