

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-209
Дрібнюк Ярослав
Прийняла :
Мельникова Н. І.

Львів-2020

Тема: “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 пришвидчення.

Для початку зроблю запит у якому з’єднаю кілька таблиць по ключах (чим більше тим краще, щоб була достатньо велика кількість полів). Потім перед ключовим словом select пропишу оператор explain який показує як буде проводитися обробка select і повертає рядок про кожну таблицю.

```
explain select types.Name, event.Name, user.Name
```

```
from (user inner join event)
```

```
inner join types on
```

```
types.id_type=event.id_type and
```

```
event.id_user=user.id_user
```

```
where event.date between "2020-05-19" AND '2020-05-29' AND Is_Admin=1;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	user	HULL	ALL	PRIMARY	HULL	HULL	HULL	5	20.00	Using where
1	SIMPLE	event	HULL	ref	id_type,id_user	id_user	4	calendar.user.id_User	1	33.33	Using where
1	SIMPLE	types	HULL	eq_ref	PRIMARY	PRIMARY	4	calendar.event.id_type	1	100.00	HULL

Result 41 x Read Only

Output

Action Output

#	Time	Action	Message
1	22:42:36	explain select straight_join types.Name, event.Name, user.Name from (user inner join eve...	3 row(s) returned

Запустивши команду, бачимо, що у стовпці типах зв’язування (type) є значення ALL для таблиці user що означає про те, що для кожної комбінації рядків буде проводитися повний перегляд таблиці, що є поганим варіантом.

Для усунення цього значення необхідно у таблиці user створити індекси які прискорюватимуть пошук.

create index mail_index on user(`E-mail`);

11	•	create index mail_index on user(`E-mail`);
12	•	drop index mail_index on user;

Output			
Action Output			
#	Time	Action	Message
✓ 1	22:42:36	explain select straight_join types.Name, event.Name, user.Name from (user inner join eve...	3 row(s) returned
✓ 2	22:51:35	create index mail_index on user(`E-mail`)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Спробуємо знову здійснити пошук, отримаємо такий результат:

```
4 types.id_type=event.id_type and
5 event.id_user=user.id_user
6 where event.date between "2020-05-19" AND '2020-05-29' AND Is_Admin=1;
7
8
9 • show index from user;
10 • desc user;
11 • create index mail_index on user(`E-mail`);
12 • drop index mail_index on user;
13
```

Result Grid

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	user	NULL	ALL	PRIMARY	NULL	NULL	NULL	5	20.00	Using where
	1	SIMPLE	event	NULL	ref	id_type,id_user	id_user	4	calendar.user.id_User	1	33.33	Using where
	1	SIMPLE	types	NULL	eq_ref	PRIMARY	PRIMARY	4	calendar.event.id_type	1	100.00	NULL

Result 42

Read Only

Output

Action Output

#	Time	Action	Message
✓ 1	22:42:36	explain select straight_join types.Name, event.Name, user.Name from (user inner join eve...	3 row(s) returned
✓ 2	22:51:35	create index mail_index on user(`E-mail`)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 3	22:54:37	explain select types.Name, event.Name, user.Name from (user inner join event) inner join ...	3 row(s) returned

Тепер пробуємо створити індекс використавши поле Is_Admin
create index admin_index on user(Is_Admin);

Limit to 1000 rows

```

1 • explain select types.Name, event.Name, user.Name
2   from (user inner join event)
3   inner join types on
4   types.id_type=event.id_type and
5   event.id_user=user.id_user
6   where event.date between "2020-05-19" AND '2020-05-29' AND Is_Admin=1;
7
8
9 • show index from user;
10 • desc user;

```

Result Grid

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	event	NULL	ALL	id_type,id_user	NULL	NULL	NULL	3	33.33	Using where
	1	SIMPLE	user	NULL	eq_ref	PRIMARY,admin_index	PRIMARY	4	calendar.event.id_user	1	40.00	Using where
	1	SIMPLE	types	NULL	eq_ref	PRIMARY	PRIMARY	4	calendar.event.id_type	1	100.00	Using where

Result 46 x

Output

Action Output

#	Time	Action	Message
✓ 6	22:57:15	explain select types.Name, event.Name, user.Name from (user inner join event) inner joi...	3 row(s) returned
✓ 7	22:57:30	explain select straight_join types.Name, event.Name, user.Name from (user inner join ev...	3 row(s) returned
✓ 8	22:57:39	explain select types.Name, event.Name, user.Name from (user inner join event) inner joi...	3 row(s) returned

Бачимо що для 2 інших таблиць значення type буде рівне eq_ref, що означає що для кожної комбінації рядків буде зчитуватися лише 1 рядок з таблиці що є оптимальним варіантом.

Спробуємо використати straight_join для з'єднання таблиць у порядку їх прописання в коді, запустивши код отримаємо такий результат:

Limit to 1000 rows

```

1 • explain select straight_join types.Name, event.Name, user.Name
2   from (user inner join event)
3   inner join types on
4   types.id_type=event.id_type and
5   event.id_user=user.id_user
6   where event.date between "2020-05-19" AND '2020-05-29' AND Is_Admin=1;
7
8 • show index from user;
9 • desc user;

```

Result Grid

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	user	NULL	ref	PRIMARY,admin_index	admin_index	1	const	2	100.00	Using index
	1	SIMPLE	event	NULL	range	id_type,id_user	id_type	5	NULL	1	33.33	Using index
	1	SIMPLE	types	NULL	eq_ref	PRIMARY	PRIMARY	4	calendar.event.id_type	1	100.00	Using index

Result 54 x

Даний результат показує, що пошук буде більш оптимальним.

Висновок: на даній лабораторній роботі я ознайомився з індексами, їх призначенням, операторами explain та straight_join та навчився з їх використанням оптимізовувати пошук записів.