Laboratorium VII

Wiktor Zmiendak

1. Szkolenie. Utwórz bazę danych **productdb** i tabelę produktów **product** w następujący sposób a następnie uzupełnij tabelę **product** przykładowymi danymi:

```
create schema productdb;
use productdb;

create table product(
id int auto_increment primary key,
product_name varchar(30) not null,
manufacturer varchar(30) not null,
product_count int default 0,
price decimal not null
);

insert into product(product_name, manufacturer, product_count, price)
values
('iphone X', 'Apple', 2, 7600),
('iphone 8', 'Apple', 2, 5600),
('iphone 7', 'Apple', 4, 4600),
('iphone 6', 'Apple', 2, 3600),
('Nokia 3444', 'Nokia', 6, 2600),
('Nokia 3333', 'Nokia', 6, 1600),
('Nokia 3310', 'Nokia', 4, 1100);
```

Załóżmy, że musisz wybrać z tabeli dane dotyczące dostępnych produktów producenta Nokia kosztuje niecałe 2000. Zapytanie o próbkowanie danych z tabeli będzie wyglądać tak:

```
22 •  select * from product where manufacturer='Nokia' and price < 2000;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

id	product_name	manufacturer	product_count	price
6	Nokia 3333	Nokia	6	1600
7	Nokia 3310	Nokia	4	1100
NULL	NULL	NULL	NULL	NULL

Użyj polecenia, aby przeanalizować możliwą optymalizację tego zapytania EXPLAINE. Dodaj to polecenie do utworzonego zapytania:

```
23 •  explain select * from product where manufacturer='Nokia' and price < 2000;
```

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	product	NULL	ALL	NULL	NULL	NULL	NULL	7	14.29	Using where

Dodaj do tabeli product kompilowany indeks manufact_price na podstawie kolumn producenta i ceny:

```
23 •  alter table product add index manufact_price(manufacturer, price);
24 •  explain select * from product where manufacturer='Nokia' and price < 2000;
```

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	product	NULL	range	manufact_price	manufact_price	127	NULL	2	100.00	Using index condition

Wniosek:

Po dodaniu indeksu w wyniku zapytania widzimy, że są one dostępne i można z nich korzystać w celu zwiększenia wydajności wyszukiwania naszych rekordów.

2. Zadaj i przeanalizuj co najmniej 2 złożone zapytania do utworzonej wcześniej bazy danych dla wybranego obszaru tematycznego. Zaproponuj możliwe podejścia do ich optymalizacji (**tworzenie indeksów lub zmiennych użytkownika, modyfikacja zapytania itp.):**

Przygotowana baza danych:

```
5 •  create table if not exists runway(
6      runway_id int(15) primary key,
7      runway_number int(15),
8      length int(25),
9      runway_condition char(15));
10
11 •  create table if not exists plane(
12     plane_id int(15) primary key,
13     company varchar(25),
14     weight int(15),
15     size int(15),
16     pasengers_slots int(15),
17     speed int(15),
18     id_runway int(15));
```

Przygotowane wprowadzone dane:

```
insert into airport4.runway(runway_id, runway_number, length, runway_condition)
values
('187', '1', '3000', 'A'),
('2998', '2', '3000', 'B'),
('3000', '3', '3000', 'A'),
('4989', '4', '3000', 'C'),
('54232', '5', '3000', 'A'),
('6289', '6', '10000', 'A'),
('762', '7', '3000', 'F'),
('853', '8', '3000', 'F'),
('9115', '9', '3000', 'A'),
('1650', '10', '200', 'A');
```

```
insert into airport4.plane(plane_id, company, weight, size, pasengers_slots, speed, id_runway)
values
('111', 'EasyJet', '1000', '50', '10', '500', '187'),
('224', 'Ryaner', '1000', '50', '103', '500','2998'),
('5253', 'Ryaner', '2500', '50', '200', '500','3000'),
('4564', 'LOT', '1000', '50', '200', '500','4989'),
('1', 'LOT', '1000', '50', '10', '600','54232'),
('667', 'EasyJet', '3900', '50', '104', '550','6289'),
('787', 'EasyJet', '1000', '50', '102', '550','762'),
('832', 'EasyJet', '9000', '50', '10', '550','853'),
('911', 'EasyJet', '8000', '50', '102', '600','9115'),
('1110', 'Swishair', '200', '50', '10', '750','1650');
```

Pierwsze zapytanie:

```
select runway_id, runway_condition from runway
where length > 3000;
```

Result Grid | Filter Rows: | Edit:

runway_id	runway_condition
853	F
4989	C
6289	A
9115	A
NULL	NULL

Zaproponowane podejście:

```
alter table runway add index runway_index(runway_condition, length);

explain select runway_id, runway_condition from runway
where length > 3000;
```

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	runway	NULL	index	runway_index	runway_index	66	NULL	10	33.33	Using where; Using index

Wniosek:

Po dodaniu indeksowania zapytanie informuje nas, że szybsze wyszukiwanie będzie przebiegać poprzez odwoływanie się do runway_index co potencjalnie usprawni proces działania zapytania.

Drugie zapytanie:

```
select plane_id, size, speed from plane
where company in ('EasyJet', 'LOT') and speed > (select avg(speed) from plane);

explain select plane_id, size, speed from plane
where company in ('EasyJet', 'LOT') and speed > (select avg(speed) from plane);
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	plane	NULL	ALL	NULL	NULL	NULL	NULL	10	10.00	Using where
2	SUBQUERY	plane	NULL	ALL	NULL	NULL	NULL	NULL	10	100.00	NULL

Zaproponowane podejście:

```
select @avgSpeed := avg(speed) from plane;
select plane_id, size, speed from plane
where company in ('EasyJet', 'LOT') and speed > @avgSpeed;

explain select * from plane
where company in ('EasyJet', 'LOT') and speed > @avgSpeed;
```

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	plane	NULL	ALL	NULL	NULL	NULL	NULL	10	10.00	Using where

Wniosek:

Dzięki zastosowaniu zmiennej sesyjnej avgSpeed zapytanie nie musi dokonywać wielokrotnych obliczeń średniej prędkości samolotu co usprawnia pracę zapytania.