

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего
образования Санкт-Петербургский национальный
исследовательский университет информационных технологий механики и оптики

Мега факультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа № 3
По дисциплине «Проектирование баз данных»
Создание представления и индексов в PostgreSQL

Выполнил студент группы М34091

Джахан Исрат

Санкт-Петербург

2024

Задача:

Создать представление и индексы к нескольким таблицам (минимум к двум).

Требования к представлению:

1. Должны быть задействованы минимум 2 таблицы.
2. Должно быть задействовано минимум 4 поля.

Требования к индексу:

1. Должны использоваться минимум 2 поля для создания индекса.
2. Должно быть создано минимум 2 индекса к разным таблицам.

Порядок выполнения работы:

1. Выбрать таблицы и поля для представления и создать само представление.
2. Выбрать поля, по которым будет создан индекс и создать индексы (минимум к двум таблицам).
3. Сделать выборку из таблиц, используя созданные индексы.
4. Предоставить отчет.

1. Представления:

1) Представление для таблиц CAR, BRAND, CARMODEL:

Выбраны 6 полей: car_id, model_name, registration_number, brand_name, price_per_day, availability_status.

SQL запрос:

No limit

Query Query History

```
1  CREATE VIEW CarDetails AS
2  SELECT
3      Car.car_id,
4      CarModel.model_name,
5      Car.registration_number,
6      Brand.brand_name,
7      Car.price_per_day,
8      Car.availability_status
9  FROM
10     Car
11  JOIN
12     CarModel ON Car.car_model_id = CarModel.car_model_id
13  JOIN
14     Brand ON Car.brand_id = Brand.brand_id;
```

Результат:

	car_id integer	model_name character varying (100)	registration_number character varying (50)	brand_name character varying (100)	price_per_day numeric (10,2)	availability_status character varying (20)
1	1	Lada Granta	A123BCD7	Lada	1500.00	available
2	2	Gazelle Next	B234CD7	Gaz	2000.00	rented
3	3	Kamaz 6520	C345DE77	Kamaz	3000.00	available
4	4	UAZ Patriot	E567FG77	UAZ	1000.00	rented
5	5	Moskvitch 412	F678GH77	Moskvitch	1700.00	available
6	6	Renault Duster	G789IJ77	Renault	4500.00	available
7	7	Volkswagen Polo	H901JK77	Volkswagen	14000.00	rented
8	8	BMW 3 Series	I901JK77	BMW	3000.00	available
9	9	Toyota Camry	J012KL77	Toyota	2300.00	rented
10	10	Hyundai Solaris	K123MN77	Hyundai	2700.00	available
11	11	Chevrolet Niva	L234OP77	Chevrolet	1600.00	rented
12	12	Ford Focus	M345QR77	Ford	3500.00	available
13	13	Nissan Qashqai	N456ST77	Nissan	4000.00	available
14	14	Kia Rio	P678WX77	Kia	3700.00	available
15	15	Mazda 3	R890YZ77	Mazda	3200.00	available
16	16	Mercedes-Benz A-Class	S901AB77	Mercedes-Benz	1500.00	rented
17	17	Skoda Octavia	T012BC77	Skoda	2500.00	rented
18	18	Audi A4	U123CD78	Audi	1800.00	available
19	19	Honda Civic	V234DE79	Honda	2200.00	rented
20	20	Peugeot 308	W345FG80	Peugeot	1900.00	available

2) Представление для таблиц CLIENT, REVIEW:

Выбраны 4 поля: client_name, email, rating, client_comment

SQL запрос:

No limit

Query

Query History

```
1 CREATE VIEW ClientReview AS
2 SELECT
3     Client.client_name,
4     Client.email,
5     Review.rating,
6     Review.client_comment
7 FROM
8     Client
9 JOIN
10    Review ON Client.client_id = Review.client_id;
```

Результат:

	client_name character varying (100)	email character varying (100)	rating integer	client_comment text
1	Ivan Petrov	ivan.petrov@example.ru	5	Отличная машина и сервис!
2	Olga Sidorova	olga.sidorova@example.ru	4	Хороший опыт в целом
3	Dmitry Ivanov	dmitry.ivanov@example.ru	3	Приемлемо, но есть место для улучшения
4	Anna Kuznetsova	anna.kuznetsova@example.ru	5	Мне понравился автомобиль!
5	Sergey Popov	sergey.popov@example.ru	2	Были проблемы с машиной
6	Elena Smirnova	elena.smirnova@example.ru	4	Хорошая машина, плавная езда
7	Roman Kozlov	roman.kozlov@example.ru	5	Идеальный опыт!
8	Irina Fedorova	irina.fedorova@example.ru	3	Неплохо, но не лучший
9	Mikhail Orlov	mikhail.orlov@example.ru	4	Хорошая цена для своих денег
10	Nina Voronova	nina.voronova@example.ru	1	Очень разочарован машиной
11	Yuri Zaitsev	yuri.zaitsev@example.ru	5	Удивительная машина, очень чистая
12	Natalya Lebedeva	natalya.lebedeva@example.ru	2	Проблемы с кондиционером
13	Andrey Grigoriev	andrey.grigoriev@example.ru	4	Наслаждался поездкой!
14	Ekaterina Sorokina	ekaterina.sorokina@example.ru	5	Настоятельно рекомендую!
15	Vladimir Sokolov	vladimir.sokolov@example.ru	3	Просто нормально, не отлично
16	Marina Pavlova	marina.pavlova@example.ru	2	Не стоит своей цены
17	Alexey Gusev	alexey.gusev@example.ru	5	Исключительное обслуживание!
18	Oksana Krylova	oksana.krylova@example.ru	3	Средний опыт
19	Viktor Nikitin	viktor.nikitin@example.ru	4	Довольно хорошее в целом
20	Anastasia Belova	anastasia.belova@example.ru	5	Мне все понравилось!

2. Индексы:

1) Создание индекса для таблицы CLIENT:

A screenshot of a SQL editor interface. The top toolbar contains icons for file operations (folder, save, dropdown), editing (pencil, dropdown), filtering (funnel, dropdown), and execution (play, dropdown, E icon, bar chart icon, dropdown, and a checkmark icon). Below the toolbar, there are two tabs: 'Query' (selected) and 'Query History'. The 'Query' tab shows a SQL statement: `CREATE INDEX idx_client_email_phone ON Client (email, phone);` on line 1.

2) Создание индекса для таблицы REVIEW:

The image shows the top portion of a SQL editor interface. At the top is a toolbar with various icons for file operations (folder, save, undo, redo), editing (pencil), filtering (funnel), and execution (play, stop, refresh, etc.). Below the toolbar, there are two tabs: 'Query' and 'Query History'. The 'Query' tab is active, and below it, a text input area contains the SQL command: `CREATE INDEX idx_review_rating_date ON Review (rating, review_date);`

3. Выборка из таблиц используя индексы:

1) Для индекса idx_client_email_phone:

Query Query History

```
EXPLAIN ANALYZE SELECT client_id, client_name, email FROM Client WHERE email = 'olga.sidorova@example.ru' AND phone = '89098765432';
```

Data Output Messages Notifications

	QUERY PLAN text
1	Seq Scan on client (cost=0.00..1.30 rows=1 width=440) (actual time=0.012..0.013 rows=1 loops...
2	Filter: (((email)::text = 'olga.sidorova@example.ru':::text) AND ((phone)::text = '89098765432':te...
3	Rows Removed by Filter: 19
4	Planning Time: 0.055 ms
5	Execution Time: 0.026 ms

Этот запрос не использовал индекс, вместо него было использовано последовательное сканирование. Это происходит из-за того, что таблица имеет слишком маленький объем данных, поэтому программа посчитала, что использование индекса займет больше времени. В таком случае принудительно выполняю через индекс:



No limit



E







Query










Query History

```
1 SET enable_seqscan TO off;
2 EXPLAIN ANALYZE SELECT review_id, rating, review_date FROM Review
3 WHERE rating = 5 AND review_date > '2024-01-01';
```

Data Output

Messages

Notifications



SQL

	<div>QUERY PLAN</div> <div>text</div> <div></div>
1	Index Scan using idx_review_rating_date on review (cost=0.14..8.16 rows=1 width=12) (actual time=0.077..0.080 rows=7 loop...
2	Index Cond: ((rating = 5) AND (review_date > '2024-01-01'::date))
3	Planning Time: 0.059 ms
4	Execution Time: 0.096 ms