

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

“Санкт-Петербургский национальный исследовательский университет
информационных технологий механики и оптики”

Мегафакультет: трансляционных информационных технологий

Факультет: информационных технологий и программирования

Лабораторная работа №3

По дисциплине: “Проектирование баз данных”

Тема: “Создание представления и индексов в PostgreSQL”

Выполнила студент группы №М3216:

Шевцов Роман Сергеевич

САНКТ-ПЕТЕРБУРГ

2025

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, включить в него следующие данные:
 - перечень таблиц и полей, задействованных в представлении;
 - представлении и его назначение для ваших данных;
 - данные из представления;
 - индексы и по каким полям они строились;
 - query plan для разбора запросов select, по которому будет видно, что ваши созданные индексы используются.

На защите лабораторной необходимо будет продемонстрировать созданное представление, индексы, select и query plan.

Решение:

Для таблиц User, Subscription и Tariff создадим представление, которое будет объединять их данные.

Query

Query History

1

select * from PBD_Database.user

Data Output

Messages

Notifications

SQL

Show in

	userid [PK] integer	firstname character varying (50)	lastname character varying (50)	phonenumber character varying (30)	email character varying (50)	address character varying (100)
1	1	Коля	Петров	+79161234567	kolya.petrov@example.com	Москва, ул. Ленина, д. 5
2	2	Анна	Смирнова	+79269876543	anna.smirnova@example.com	Санкт-Петербург, Невский проспект, д. 10
3	3	Дмитрий	Иванов	+79991112233	dmitry.ivanov@example.com	Екатеринбург, ул. Мира, д. 20
4	4	Лена	Павлова	+79453786526	lena.pavlova@example.com	Санкт-Петербург, ул. Ленсовета, д. 54
5	5	Костя	Сokolov	+79213354583	kostya.sokolov@example.com	Ростов, ул. Советская, д. 6

Таблица User

Query

Query History

1

select * **from** PBD_Database.Subscription

Data Output

Messages

Notifications

SQL

	subscriptionid [PK] integer	userid integer	tariffid integer	startdate date	enddate date	status character varying (20)
1	1	1	1	2024-02-01	2024-03-01	Active
2	2	2	2	2024-01-15	2024-03-15	Active
3	3	3	3	2024-02-10	2024-05-10	Active
4	4	4	4	2025-01-21	2025-03-21	Active
5	5	5	5	2024-07-13	2024-09-13	Active

Query

Query History

1 `select * from PBD_Database.Tariff`

Data Output

Messages

Notifications

≡

+

▼

▼

SQL

	tariffid [PK] integer	tariffname character varying (100)	price numeric (10,2)	description text	validityperiod integer
1	1	Стандарт	500.00	Базовый тариф на 30 дней	30
2	2	Стандарт+	800.00	Базовый тариф с увеличенным количеством дней на 20	50
3	3	Премиум	1200.00	Расширенный тариф с доп. услугами на 60 дней	60
4	4	Премиум+	1600.00	Расширенный тариф с доп. услугами увеличенный на 30 дней	90
5	5	Ультра	2500.00	Максимальный тариф с VIP-доступом на 100 дней	100

Создание представления:

```

1  CREATE VIEW PBD_Database.UserSubscriptionDetails AS
2  SELECT
3      u.UserID,
4      u.FirstName,
5      u.LastName,
6      s.SubscriptionID,
7      s.StartDate,
8      s.EndDate,
9      t.TariffName,
10     t.Price
11 FROM PBD_Database.User u
12 JOIN PBD_Database.Subscription s ON u.UserID = s.UserID
13 JOIN PBD_Database.Tariff t ON s.TariffID = t.TariffID;

```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 146 msec.

Мы создадим представление, которое объединяет данные из таблиц User, Subscription и Tariff. Это представление объединяет информацию о пользователях, их подписках и тарифах, что упрощает анализ данных.

Query

Query History

1
SELECT * FROM PBD_Database.UserSubscriptionDetails;

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗑️

📥

⬇️

📈

SQL

	userid integer	firstname character varying (50)	lastname character varying (50)	subscriptionid integer	startdate date	enddate date	tariffname character varying (100)	price numeric (10,2)
1	1	Коля	Петров	1	2024-02-01	2024-03-01	Стандарт	500.00
2	2	Анна	Смирнова	2	2024-01-15	2024-03-15	Стандарт+	800.00
3	3	Дмитрий	Иванов	3	2024-02-10	2024-05-10	Премиум	1200.00
4	4	Лена	Павлова	4	2025-01-21	2025-03-21	Премиум+	1600.00
5	5	Костя	Соколов	5	2024-07-13	2024-09-13	Ультра	2500.00

```
Query  Query History
1  CREATE INDEX idx_user_name_email ON PBD_Database.User (FirstName, LastName, Email);

Data Output  Messages  Notifications

CREATE INDEX

Query returned successfully in 154 msec.
```

```
Query  Query History
1  CREATE INDEX idx_subscription_user_dates ON PBD_Database.Subscription (UserID, StartDate, EndDate);
Data Output  Messages  Notifications
CREATE INDEX
Query returned successfully in 115 msec.
```

```
Query Query History
1 SET enable_seqscan TO off; --отключаем последовательное сканирование
Data Output Messages Notifications
SET
Query returned successfully in 80 msec.
```

Далее пишем запросы для проверки:

QueryQuery History

1EXPLAIN ANALYZE

2SELECT * FROM PBD_Database.User

3WHERE FirstName = 'Коля' AND LastName = 'Петров' AND Email = 'kolya.petrov@example.com';

Data OutputMessagesNotifications

SQL

QUERY PLAN

text

1Index Scan using idx_user_name_email on "user" (cost=0.13..8.15 rows=1 width=654) (actual time=0.095..0.098 rows=1 loops=1)

2Index Cond: (((firstname)::text = 'Коля'::text) AND ((lastname)::text = 'Петров'::text) AND ((email)::text = 'kolya.petrov@example.com'::te...

3Planning Time: 0.235 ms

4Execution Time: 0.153 ms

QueryQuery History

1EXPLAIN ANALYZE

2SELECT * FROM PBD_Database.Subscription

3WHERE UserID = 1 AND StartDate = '2024-02-01' AND EndDate = '2024-03-01';

Data OutputMessagesNotifications

SQL

QUERY PLAN

text

1Index Scan using idx_subscription_user_dates on subscription (cost=0.13..8.15 rows=1 width=78) (actual time=0.074..0.075 rows=1 loops=...

2Index Cond: ((userid = 1) AND (startdate = '2024-02-01'::date) AND (enddate = '2024-03-01'::date))

3Planning Time: 5.336 ms

4Execution Time: 0.096 ms

После проверки включаем последовательное сканирование:

QueryQuery History

1SET enable_seqscan TO on;

Data OutputMessagesNotifications

SET

Query returned successfully in 84 msec.