

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 №2**

**По дисциплине: “Проектирование баз данных”**

**Тема: “Создание таблиц в PostgreSQL”**

Выполнила студент группы №М3216:

*Шевцов Роман Сергеевич*

**САНКТ-ПЕТЕРБУРГ**

**2025**

## Задачи:

1. На основе спроектированной PDM из лабораторной 1 создать таблицы и наложить необходимые ограничения (на уникальность, на диапазон значений).
2. Заполнить таблицы данными, по 5–10 записей в основных таблицах.

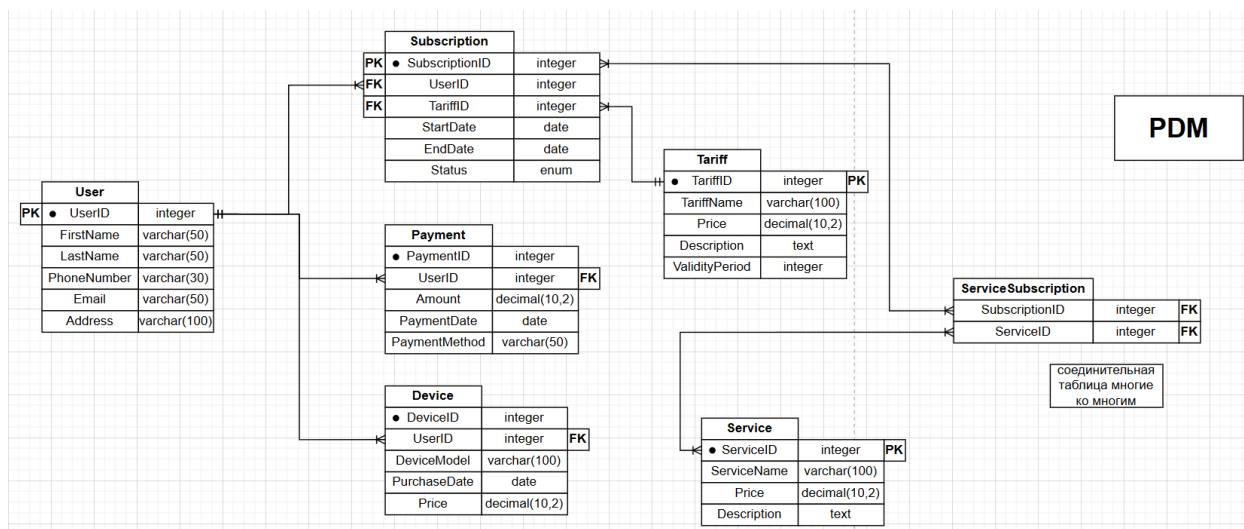
## Порядок выполнения работы:

1. Скачайте и установите [PostgreSQL](#) (также установите pgAdmin, pgBouncer)
2. Выполнение:
  - 1) Создать схему и таблицы, принадлежащие этой схеме. [Документация](#)
  - 2) Заполнить созданные таблицы данными (минимум по 5–10 записей в основных таблицах). Часть таблиц должна быть заполнена SQL-запросами, другая часть таблиц должна быть заполнена из .csv файлов.
3. Предоставить отчёт, включив в него код для создания и заполнения таблиц.

На защите лабораторной необходимо будет продемонстрировать как были созданы схема, таблицы, а также сами SQL-запросы, .csv файлы и заполненные таблицы.

## Решение:

На основе имеющейся схемы PDM базы данных, создаем схему и таблицы.



Была создана схема PBD\_Database, в которой размещены все таблицы. Таблицы были созданы с использованием SQL-запросов, а также наложены необходимые ограничения:

- Первичные ключи (PRIMARY KEY) — для уникальной идентификации записей.
- Внешние ключи (FOREIGN KEY) — для обеспечения связей между таблицами.
- Ограничения (CHECK, UNIQUE, NOT NULL) — для контроля целостности данных.

Query	Query History
1	CREATE SCHEMA PBD_Database;
2	
3	CREATE TABLE PBD_Database.User (
4	UserID SERIAL PRIMARY KEY,
5	FirstName VARCHAR(50) NOT NULL,
6	LastName VARCHAR(50) NOT NULL,
7	PhoneNumber VARCHAR(30) NOT NULL UNIQUE,
8	Email VARCHAR(50) NOT NULL UNIQUE,
9	Address VARCHAR(100) NOT NULL
10	);
11	
12	CREATE TABLE PBD_Database.Tariff (
13	TariffID SERIAL PRIMARY KEY,
14	TariffName VARCHAR(100) NOT NULL UNIQUE,
15	Price DECIMAL(10,2) NOT NULL CHECK (Price > 0),
16	Description TEXT,
17	ValidityPeriod INTEGER NOT NULL CHECK (ValidityPeriod > 0)
18	);
19	
20	CREATE TABLE PBD_Database.Subscription (
21	SubscriptionID SERIAL PRIMARY KEY,
22	UserID INTEGER NOT NULL,
23	TariffID INTEGER NOT NULL,
24	StartDate DATE NOT NULL,
25	EndDate DATE,
26	Status VARCHAR(20) CHECK (Status IN ('Active', 'Inactive', 'Canceled')),
27	FOREIGN KEY (UserID) REFERENCES PBD_Database.User(UserID) ON DELETE CASCADE,
28	FOREIGN KEY (TariffID) REFERENCES PBD_Database.Tariff(TariffID) ON DELETE CASCADE
29	);
30	
31	CREATE TABLE PBD_Database.Payment (
32	PaymentID SERIAL PRIMARY KEY,
33	UserID INTEGER NOT NULL,
34	Amount DECIMAL(10,2) NOT NULL CHECK (Amount > 0),
35	PaymentDate DATE NOT NULL DEFAULT CURRENT_DATE,
36	PaymentMethod VARCHAR(50) NOT NULL,
37	FOREIGN KEY (UserID) REFERENCES PBD_Database.User(UserID) ON DELETE CASCADE
38	);
39	
40	CREATE TABLE PBD_Database.Device (
41	DeviceID SERIAL PRIMARY KEY,
42	UserID INTEGER NOT NULL,
43	DeviceModel VARCHAR(100) NOT NULL,
44	PurchaseDate DATE NOT NULL,
45	Price DECIMAL(10,2) NOT NULL CHECK (Price > 0),
46	FOREIGN KEY (UserID) REFERENCES PBD_Database.User(UserID) ON DELETE CASCADE
47	);
48	
49	CREATE TABLE PBD_Database.Service (
50	ServiceID SERIAL PRIMARY KEY,
51	ServiceName VARCHAR(100) NOT NULL UNIQUE,
52	Price DECIMAL(10,2) NOT NULL CHECK (Price >= 0),
53	Description TEXT
54	);
55	
56	CREATE TABLE PBD_Database.ServiceSubscription (
57	SubscriptionID INTEGER NOT NULL,
58	ServiceID INTEGER NOT NULL,
59	PRIMARY KEY (SubscriptionID, ServiceID),
60	FOREIGN KEY (SubscriptionID) REFERENCES PBD_Database.Subscription(SubscriptionID) ON DELETE CASCADE,
61	FOREIGN KEY (ServiceID) REFERENCES PBD_Database.Service(ServiceID) ON DELETE CASCADE
62	);
63	
64	INSERT INTO PBD_Database.User (FirstName, LastName, PhoneNumber, Email, Address)
65	VALUES
66	('Коля', 'Петров', '+79161234567', 'kolya.petrov@example.com', 'Москва, ул. Ленина, д. 5'),
67	('Анна', 'Смирнова', '+79269876543', 'anna.smirnova@example.com', 'Санкт-Петербург, Невский проспект, д. 10'),
68	('Дмитрий', 'Иванов', '+79991112233', 'dmitry.ivanov@example.com', 'Екатеринбург, ул. Мира, д. 20'),
69	('Лена', 'Павлова', '+79453786526', 'lena.pavlova@example.com', 'Санкт-Петербург, ул. Ленсовета, д. 54'),
70	('Костя', 'Соколов', '+79213354583', 'kostya.sokolov@example.com', 'Ростов, ул. Советская, д. 6');
71	
72	INSERT INTO PBD_Database.Tariff (TariffName, Price, Description, ValidityPeriod)
73	VALUES
74	('Стандарт', 500.00, 'Базовый тариф на 30 дней', 30),
75	('Стандарт+', 800.00, 'Базовый тариф с увеличенным количеством дней на 20', 50),
76	('Премиум', 1200.00, 'Расширенный тариф с доп. услугами на 60 дней', 60),
77	('Премиум+', 1600.00, 'Расширенный тариф с доп. услугами увеличенный на 30 дней', 90),
78	('Ультра', 2500.00, 'Максимальный тариф с VIP-доступом на 100 дней', 100);
79	
80	INSERT INTO PBD_Database.Subscription (UserID, TariffID, StartDate, EndDate, Status)
81	VALUES
82	(1, 1, '2024-02-01', '2024-03-01', 'Active'),
83	(2, 2, '2024-01-15', '2024-03-15', 'Active'),
84	(3, 3, '2024-02-10', '2024-05-10', 'Active'),
85	(4, 4, '2025-01-21', '2025-03-21', 'Active'),
86	(5, 5, '2024-07-13', '2024-09-13', 'Active');

```
87
88 INSERT INTO PBD_Database.Service (ServiceName, Price, Description)
89 VALUES
90 ('Онлайн-кинотеатр', 300.00, 'Доступ к фильмам и сериалам'),
91 ('Облачное хранилище', 200.00, '100 ГБ облачного пространства'),
92 ('Игровой сервис', 500.00, 'Подписка на игровые сервисы'),
93 ('Гороскоп', 100.00, 'Персональный гороскоп'),
94 ('Кто звонит', 400.00, 'Подписка на автоопределение номера');
95
96 INSERT INTO PBD_Database.ServiceSubscription (SubscriptionID, ServiceID)
97 VALUES
98 (1, 1),
99 (1, 2),
100 (2, 1),
101 (3, 3),
102 (3, 5),
103 (4, 1),
104 (5, 4);
```

Проверим добавление через sql:

Query Query History

1 SELECT \* FROM PBD\_Database.User

Data Output Messages Notifications

Showing rows: 1 to 5

	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	Соколов	+79213354583	kostya.sokolov@example.com	Ростов, ул. Советская, д. 6

Query Query History

1 SELECT \* FROM PBD\_Database.Tariff

Data Output Messages Notifications

Showing

	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

Query Query History

1 SELECT \* FROM PBD\_Database.Subscription

Data Output Messages Notifications

	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.Service

Data Output Messages Notifications

	serviceid [PK] integer	servicename character varying (100)	price numeric (10,2)	description text
1	1	Онлайн-кинотеатр	300.00	Доступ к фильмам и сериалам
2	2	Облачное хранилище	200.00	100 ГБ облачного пространства
3	3	Игровой сервис	500.00	Подписка на игровые сервисы
4	4	Гороскоп	100.00	Персональный гороскоп
5	5	Кто звонит	400.00	Подписка на автоопределение номера

Query

Query History

1
SELECT \* FROM PBD\_Database.ServiceSubscription

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	subscriptionid [PK] integer	serviceid [PK] integer
1	1	1
2	1	2
3	2	1
4	3	3
5	3	5
6	4	1
7	5	4

### Заполнение оставшихся таблиц через CSV файлы:

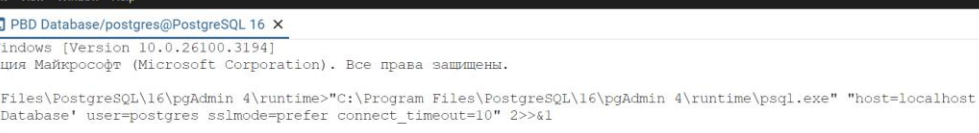
Device.csv

```
DeviceID, UserID, DeviceModel, PurchaseDate, Price
1,1,iPhone 13,2023-01-15,79999.00
2,2,Samsung Galaxy S21,2023-03-20,69999.00
3,3,Xiaomi Mi 11,2023-05-10,49999.00
4,4,Google Pixel 6,2023-07-25,59999.00
5,5,OnePlus 9,2023-09-30,54999.00
```

Payment.csv

```
PaymentID,UserID,Amount,PaymentDate,PaymentMethod
1,1,500.00,2024-02-01,Карта
2,2,800.00,2024-01-15,Карта
3,3,1200.00,2024-02-10,Карта
4,4,1600.00,2025-01-21,Карта
5,5,2500.00,2024-07-13,Карта
```

## Подключение csv файлов и передача данных в соответствующие таблицы через psql:



pgAdmin 4

File Object Tools Edit View Window Help

Welcome PBD Database/postgres@PostgreSQL 16 X

Microsoft Windows [Version 10.0.26100.3194]  
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Program Files\PostgreSQL\16\pgAdmin 4\runtime>"C:\Program Files\PostgreSQL\16\pgAdmin 4\runtime\psql.exe" "host=localhost port=5432 dbname='PBD Database' user=postgres sslmode=prefer connect\_timeout=10" 2>&1  
psql (16.8)  
WARNING: Console code page (866) differs from Windows code page (1251)  
8-bit characters might not work correctly. See psql reference  
page "Notes for Windows users" for details.  
Type "help" for help.

PBD Database=# psql -U postgres -d PBD Database  
PBD Database=# \copy PBD\_Database\_Shm.Device (UserID, DeviceModel, PurchaseDate, Price) FROM 'C:\Users\shero\Downloads\device.csv' WITH  
CSV HEADER;  
COPY 5  
PBD Database=# \copy PBD\_Database\_Shm.Payment (UserID, Amount, PaymentDate, PaymentMethod) FROM 'C:\Users\shero\Downloads\payment.csv' W  
ITH CSV HEADER;  
COPY 5  
PBD Database=#

Проверим добавление через sql запросы:

Для Device.csv

QueryQuery History

1SELECT \* FROM PBD\_Database.Device

Data OutputMessagesNotifications

SQL

	deviceid [PK] integer	userid integer	devicemodel character varying (100)	purchasedate date	price numeric (10,2)
1	1	1	iPhone 13	2023-01-15	79999.00
2	2	2	Samsung Galaxy S21	2023-03-20	69999.00
3	3	3	Xiaomi Mi 11	2023-05-10	49999.00
4	4	4	Google Pixel 6	2023-07-25	59999.00
5	5	5	OnePlus 9	2023-09-30	54999.00

Для Payment.csv

QueryQuery History

1SELECT \* FROM PBD\_Database.Payment

Data OutputMessagesNotifications

SQL

	paymentid [PK] integer	userid integer	amount numeric (10,2)	paymentdate date	paymentmethod character varying (50)
1	1	1	500.00	2024-02-01	Карта
2	2	2	800.00	2024-01-15	Карта
3	3	3	1200.00	2024-02-10	Карта
4	4	4	1600.00	2025-01-21	Карта
5	5	5	2500.00	2024-07-13	Карта

Проверим связи таблиц и ограничения на уникальность:

QueryQuery History

1SELECT u.UserID, u.FirstName, u.LastName, s.SubscriptionID, s.StartDate, s.EndDate  
2FROM PBD\_Database.User u  
3JOIN PBD\_Database.Subscription s ON u.UserID = s.UserID;

Data OutputMessagesNotifications

SQL

	userid integer	firstname character varying (50)	lastname character varying (50)	subscriptionid integer	startdate date	enddate date
1	1	Коля	Петров	1	2024-02-01	2024-03-01
2	2	Анна	Смирнова	2	2024-01-15	2024-03-15
3	3	Дмитрий	Иванов	3	2024-02-10	2024-05-10
4	4	Лена	Павлова	4	2025-01-21	2025-03-21
5	5	Костя	Соколов	5	2024-07-13	2024-09-13

QueryQuery History

1SELECT Email, COUNT(\*) FROM PBD\_Database.User GROUP BY Email HAVING COUNT(\*) > 1;

Data OutputMessagesNotifications

SQL

	email character varying (50)	count bigint
--	---------------------------------	-----------------