МИНИСТЕРСТВО НАУКИ И ВЫСШЕГООБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Учебный проект**

на тему:

**«Создание базы данных нотариальной конторы»**

**Выполнил(а):** студент(ка) группы \_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Коробейников А.П./

Подпись

**Проверил:** преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Колеганов Е.А./

Подпись

Нижний Новгород  
2024

**Содержание**

[1 Описание предметной области 3](#_Toc167093126)

[1.1 Сущности 3](#_Toc167093127)

[2 Оформление структуры в виде скрипта 4](#_Toc167093128)

[2.1 Создание операторов языка SQL для вывода агрегатных данных 6](#_Toc167093129)

[3 Создание тригер функции 7](#_Toc167093130)

# Описание предметной области

Вы работаете в нотариальной конторе Вашей задачей является отслеживание финансовой стороны работы компании.

Деятельность Вашей нотариальной конторы организована следующим образом Ваша фирма готова предоставить клиенту определенный комплекс услуг Для наведения порядка Вы формализовали эти услуги составив их список с описанием каждой услуги При обращении к Вам клиента его стандартные данные: название, вид деятельности, адрес, телефон фиксируются в базе данных По каждому факту оказания услуги клиенту составляется документ В документе указываются услуга сумма сделки, коммисионные (доход конторы), описание сделки.

## Сущности

**Клиенты** (Код клиента, Названиеа, Вид деятельности, Адрес, Телефон).

**Сделки** (Код сделки, Код клиента, Код услуги, Сумма, Коммисионные, Описание сделки).

**Услуги** (Код услуги, Название, Описание).

# Оформление структуры в виде скрипта

CREATE TABLE Clients (

Client\_Code INT PRIMARY KEY,

Company\_Name VARCHAR(255),

Type\_of\_activity VARCHAR(255),

Adress VARCHAR(255),

Number\_phone VARCHAR(20)

);

CREATE TABLE Deal (

Deal\_code INT PRIMARY KEY,

Client\_Code INT,

Service\_Code INT,

Amount DECIMAL(10, 2),

Commission DECIMAL(10, 2),

Description TEXT,

FOREIGN KEY (Client\_Code) REFERENCES Clients(Client\_Code),

FOREIGN KEY (Service\_Code) REFERENCES Service(Service\_Code)

);

CREATE TABLE Service (

Service\_Code INT PRIMARY KEY,

Service\_name VARCHAR(255),

Description TEXT

);

Заполним таблицу **Clients** данными:

INSERT INTO Clients (Client\_Code, Company\_name, Type\_of\_activity, Address, Number\_Phone) VALUES

(1, 'Иван', 'Strogov', 'Restaurant', 'ул. Центральная, 1', '89123456789'),

(2, 'Петр', 'Pegas', 'Turistick', 'пр. Солнечный, 10', '89234567890'),

(3, 'Мария', 'Apple', 'Products', 'пер. Лесной, 5', '89345678901'),

(4, 'Анна', 'Windows', 'OS', 'ул. Парковая, 7', '89456789012'),

(5, 'Алексей', 'Magnit', 'Magazine', 'пл. Центральная,3', '89567890123'),

(6, 'Елена', 'Yandex', 'Browser', 'бул. Городской, 12','89678901234');

Заполним таблицу Routes данными:

INSERT INTO Deal (Deal\_code, Client\_Code, Service\_Code, Amount, Commission, Description) VALUES

(1, 1, 1, 25000.00, 5000.00, 'Best restaurant'),

(2, 2, 2, 20000.00, 4000.00, 'Best lines'),

(3, 3, 3, 30000.00, 6000.00, 'Best apples'),

(4, 4, 4, 28500.00, 5650.00, 'Best Os'),

(5, 5, 5, 35000.00, 7000.00, 'Best magazine'),

(6, 6, 6, 29000.00, 5800.00, 'Best Browser');

Заполним таблицу Packages данными

INSERT INTO Service (Service\_Code, Service\_name, Description) VALUES

(1, 'Purchase', 'Best restaurant'),

(2, 'Sale', 'Best lines'),

(3, 'Rent', 'Best apples'),

(4, 'Purchase' 'Best Os'),

(5, 'Sale', 'Best magazine'),

(6, 'Rent', 'Best Browser');

## Создание операторов языка SQL для вывода агрегатных данных

Сначала выведем всю информацию по нашим таблицам.

SELECT \* from Clients



1. Таблица клиентов

SELECT \* from Deal



1. Таблица сделок

SELECT \* from Service



1. Таблица услуг

Вывод суммы сделок по всем клиентам:

SELECT SUM(Amount) FROM Deal



Вывод средней суммы сделки по всем клиентам:

SELECT AVG(Amount) FROM Deal



Рис. 5.

# Создание тригер функции

create Table Dml\_Log\_Order (

username text,

dml\_command text,

date\_time timestamp);

create or replace function Log\_Dml\_Ord()

returns trigger as $$

begin

Insert into dml\_log\_Order

(usernem, dml\_command, date\_time)

values

(current\_user, current\_query(), current\_timestamp);

return NULL;

end $$ language plpgsql;

create trigger Tr\_Log\_Ord\_Dml

AFTER INSERT OR UPDATE OR DELETE ON lectors

EXECUTE FUNCTION Log\_Dml\_Ord();