Курс:

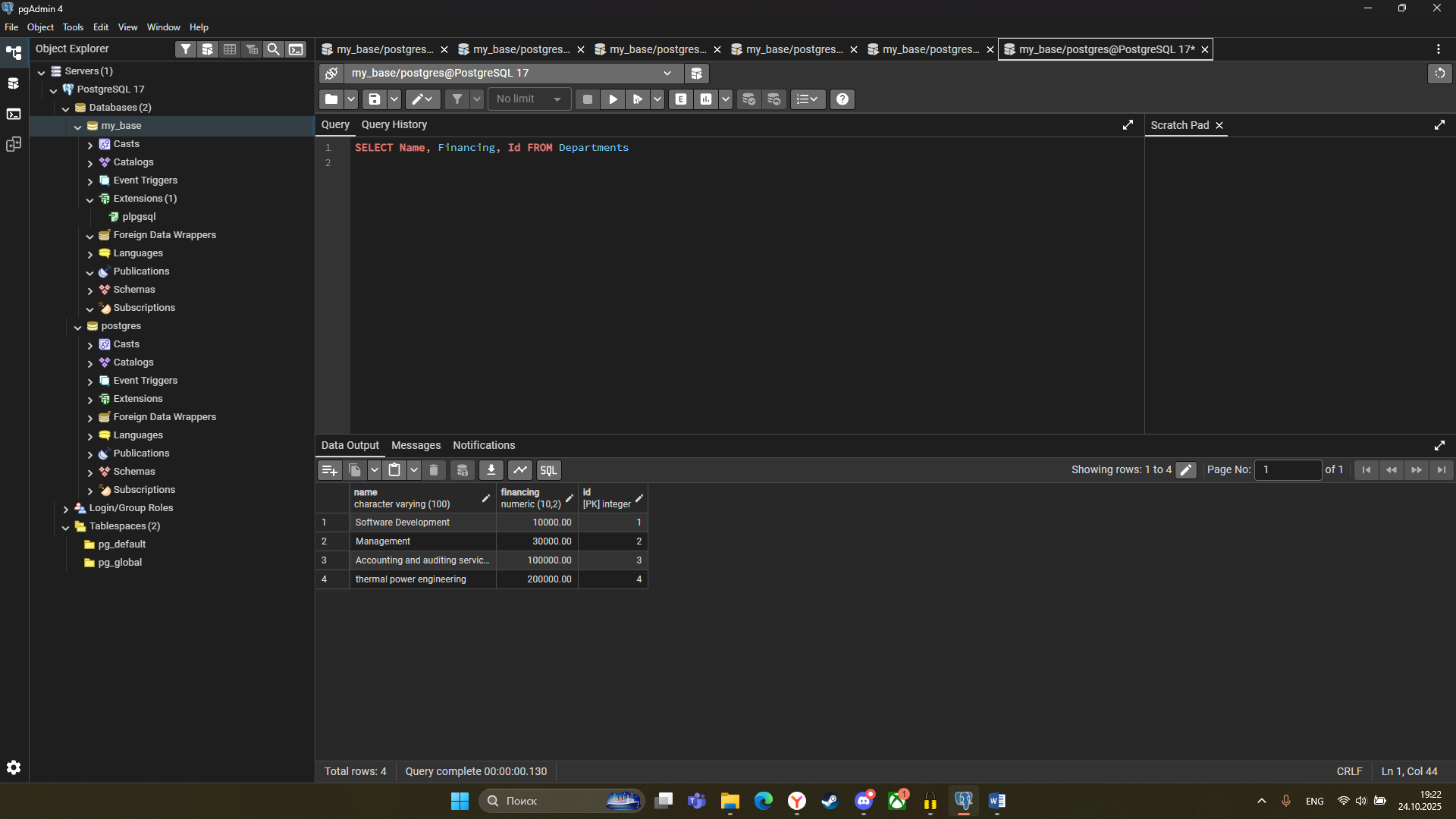
Теория баз данных

Тема: Запросы SELECT, INSERT, UPDATE, DELETE.

Запросы

1. Вывести таблицу кафедр, но расположить ее поля в обратном порядке.

SELECT Name, Financing, Id FROM Departments



1. Вывести названия групп и их рейтинги с уточнением имен полей именем таблицы.

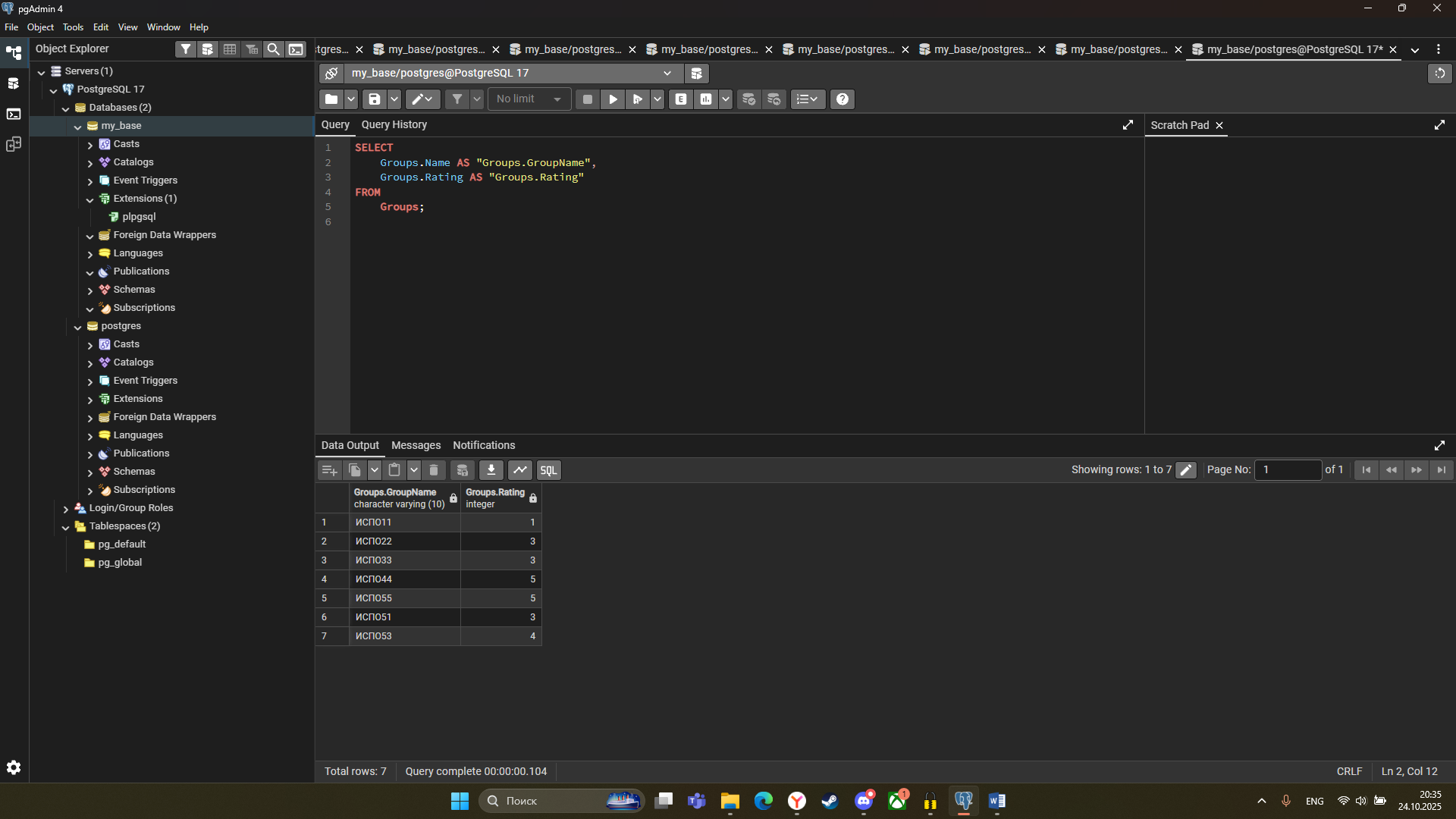
SELECT

Groups.Name AS "Groups.GroupName",

Groups.Rating AS "Groups.Rating"

FROM

Groups;



3. Вывести для преподавателей их фамилию, процент ставки по отношению к надбавке и процент ставки по отношению к зарплате (сумма ставки и надбавки).

SELECT

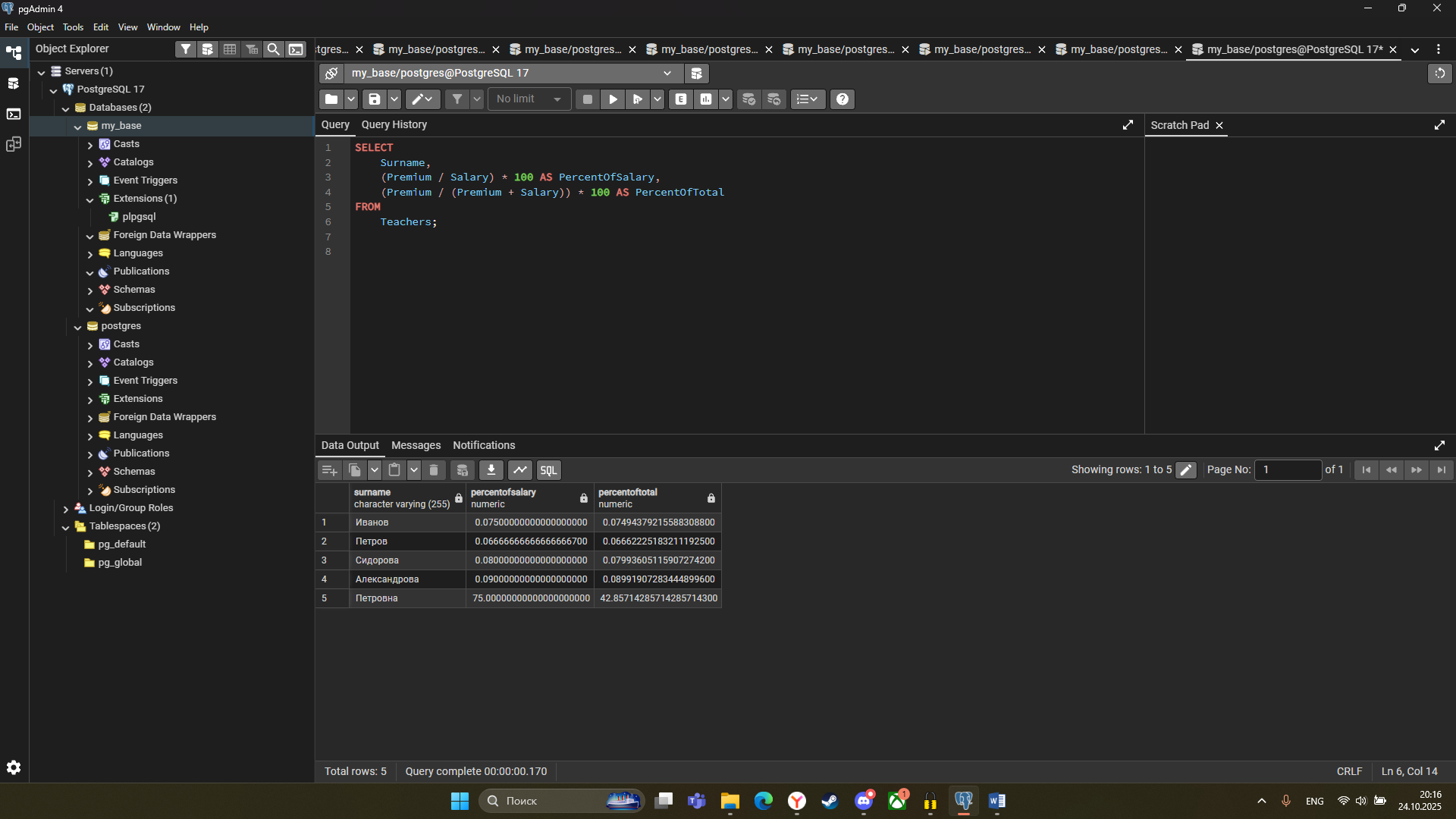
Surname,

(Premium / Salary) \* 100 AS PercentOfSalary,

(Premium / (Premium + Salary)) \* 100 AS PercentOfTotal

FROM

Teachers;



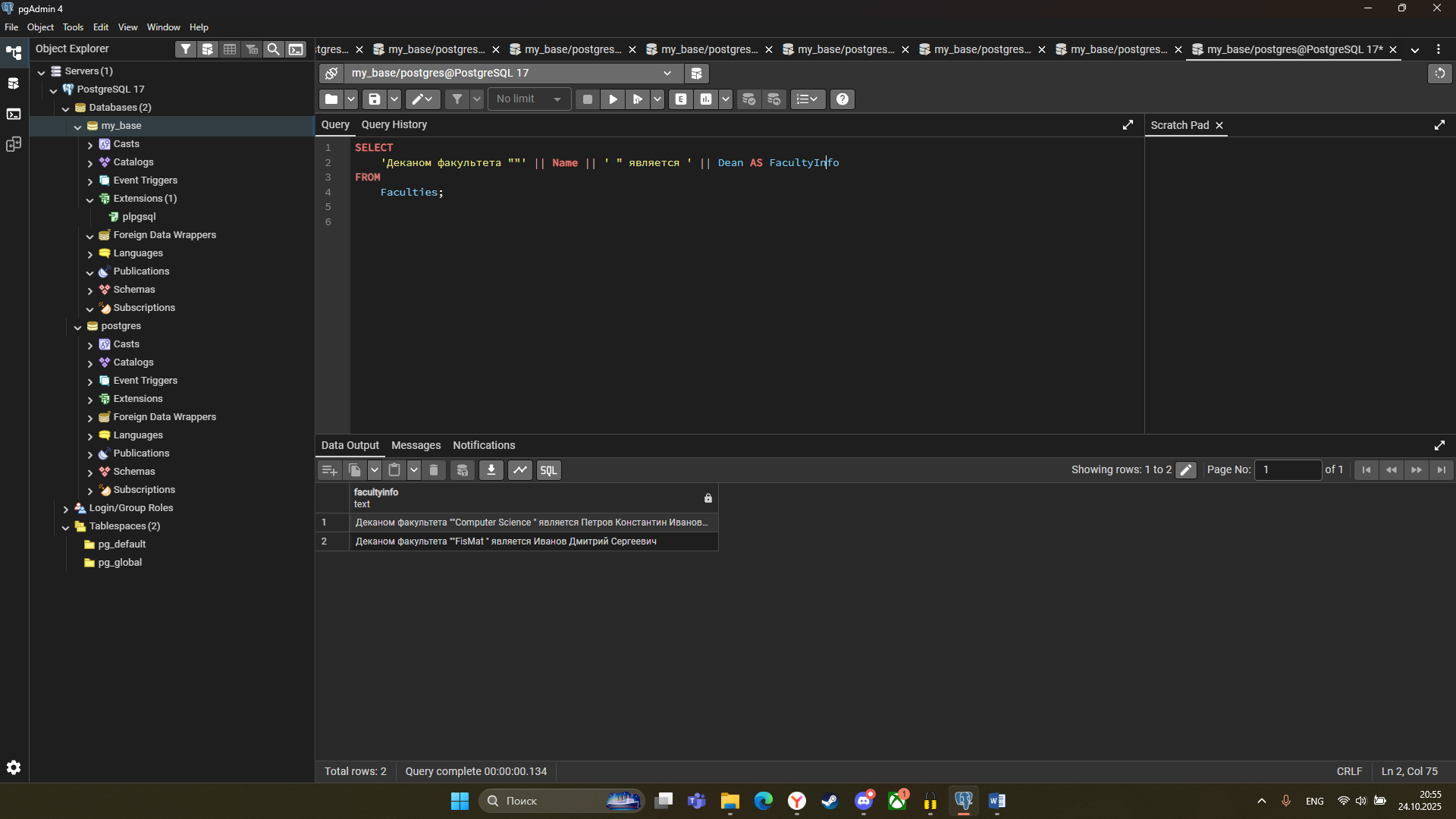
1. Вывести таблицу факультетов в виде одного поля в следующем формате: “The dean of faculty [faculty] is [dean].”.

SELECT

'Деканом факультета ""' || Name || ' " является ' || Dean AS FacultyInfo

FROM

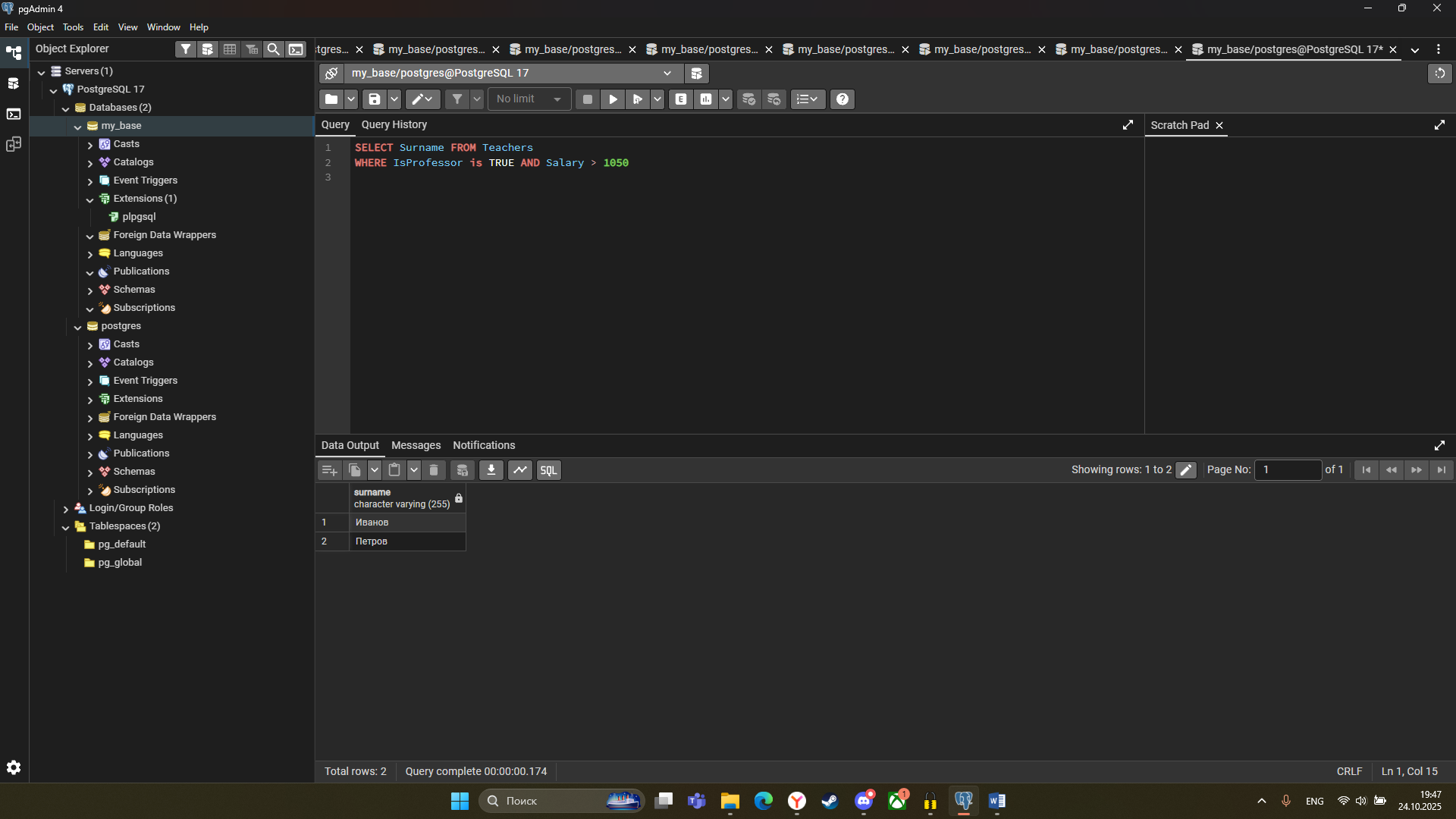
Faculties;



5. Вывести фамилии преподавателей, которые являются профессорами и ставка которых превышает 1050.

SELECT Surname FROM Teachers

WHERE IsProfessor is TRUE AND Salary > 1050

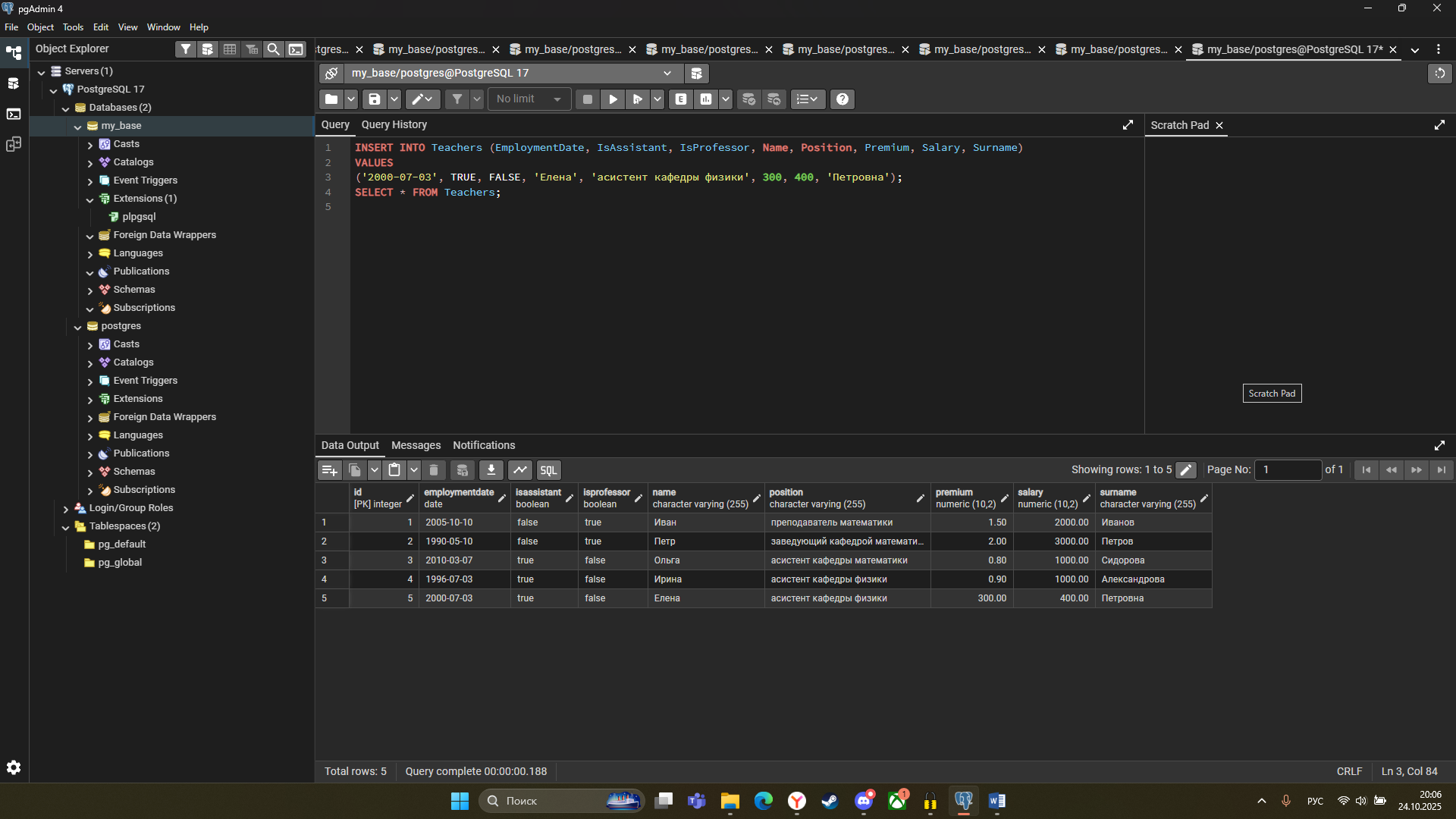


INSERT INTO Teachers (EmploymentDate, IsAssistant, IsProfessor, Name, Position, Premium, Salary, Surname)

VALUES

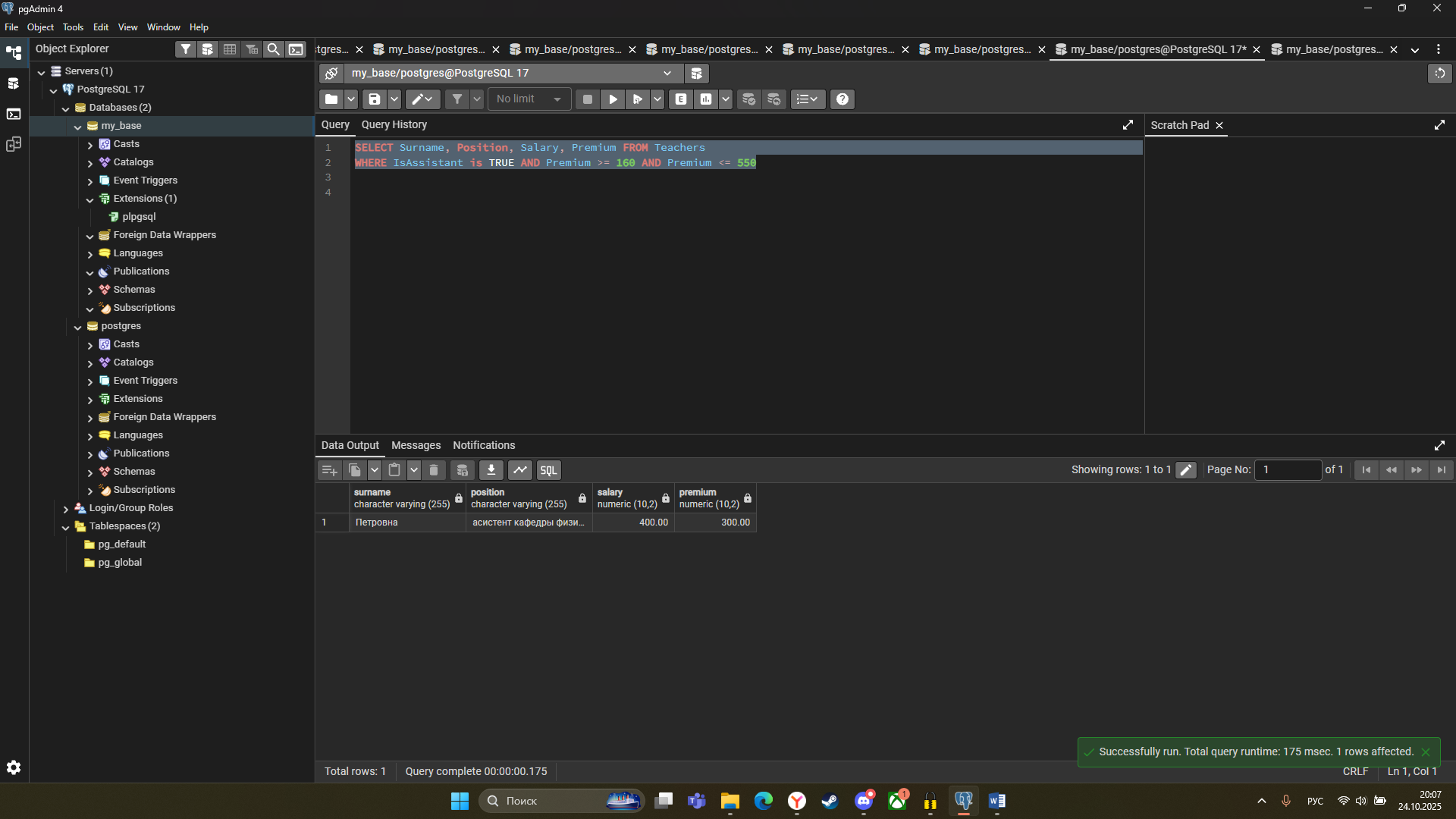
('2000-07-03', TRUE, FALSE, 'Елена', 'асистент кафедры физики', 300, 400, 'Петровна');

SELECT \* FROM Teachers;



SELECT Surname, Position, Salary, Premium FROM Teachers

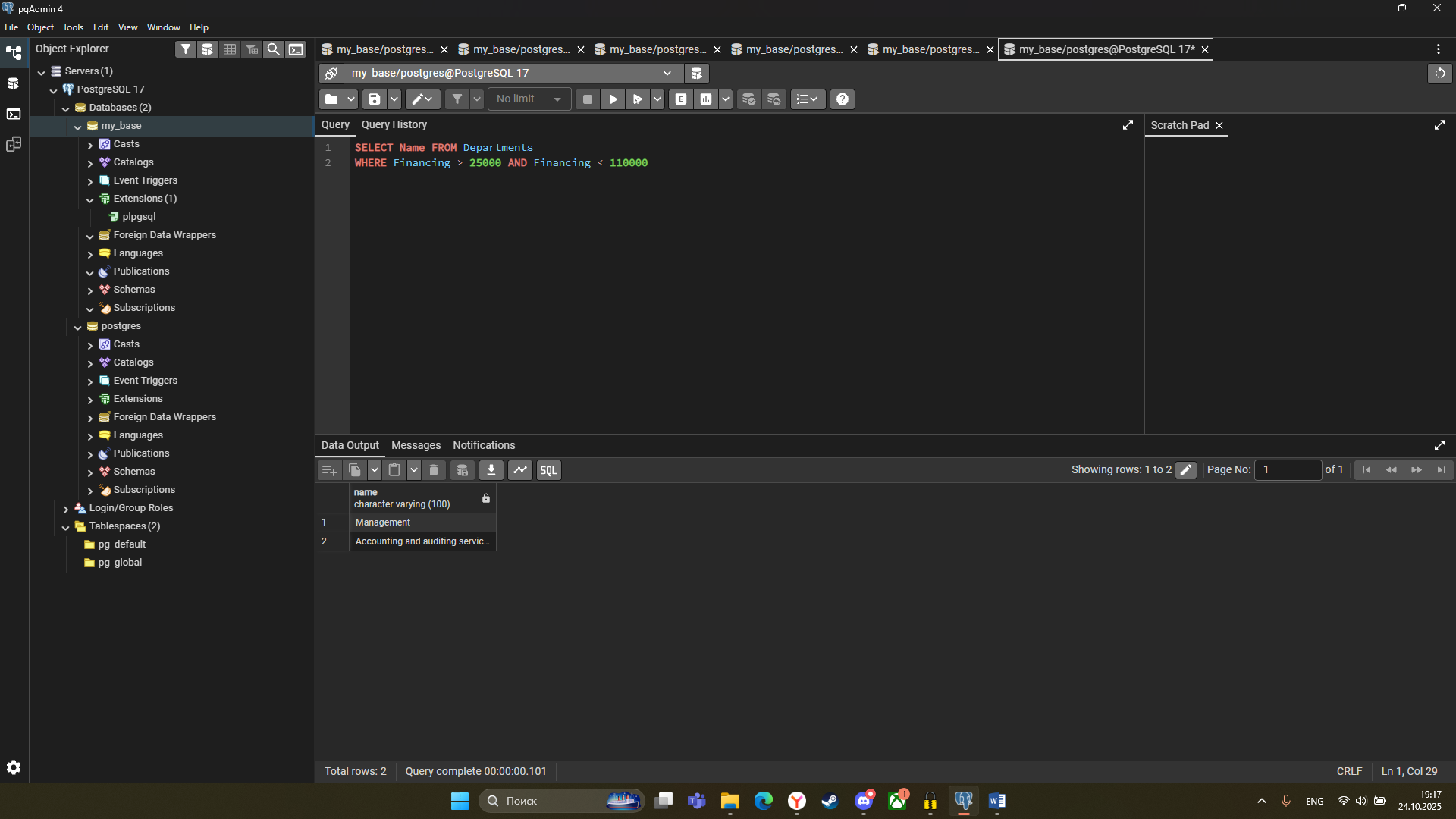
WHERE IsAssistant is TRUE AND Premium >= 160 AND Premium <= 550



6. Вывести названия кафедр, фонд финансирования которых меньше 11000 или больше 25000.

SELECT Name FROM Departments

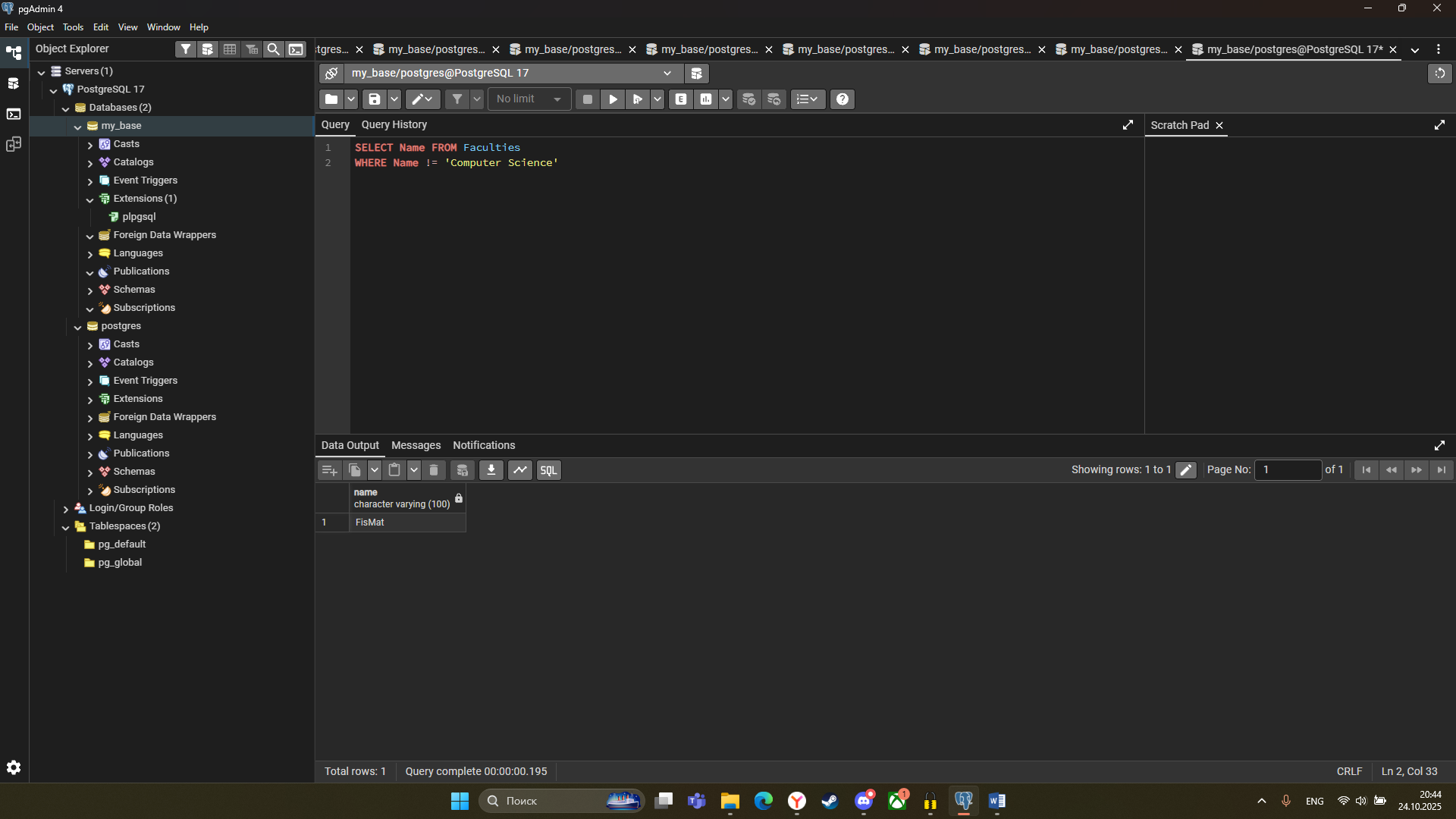
WHERE Financing > 25000 AND Financing < 110000



7. Вывести названия факультетов кроме факультета “Computer Science”.

SELECT Name FROM Faculties

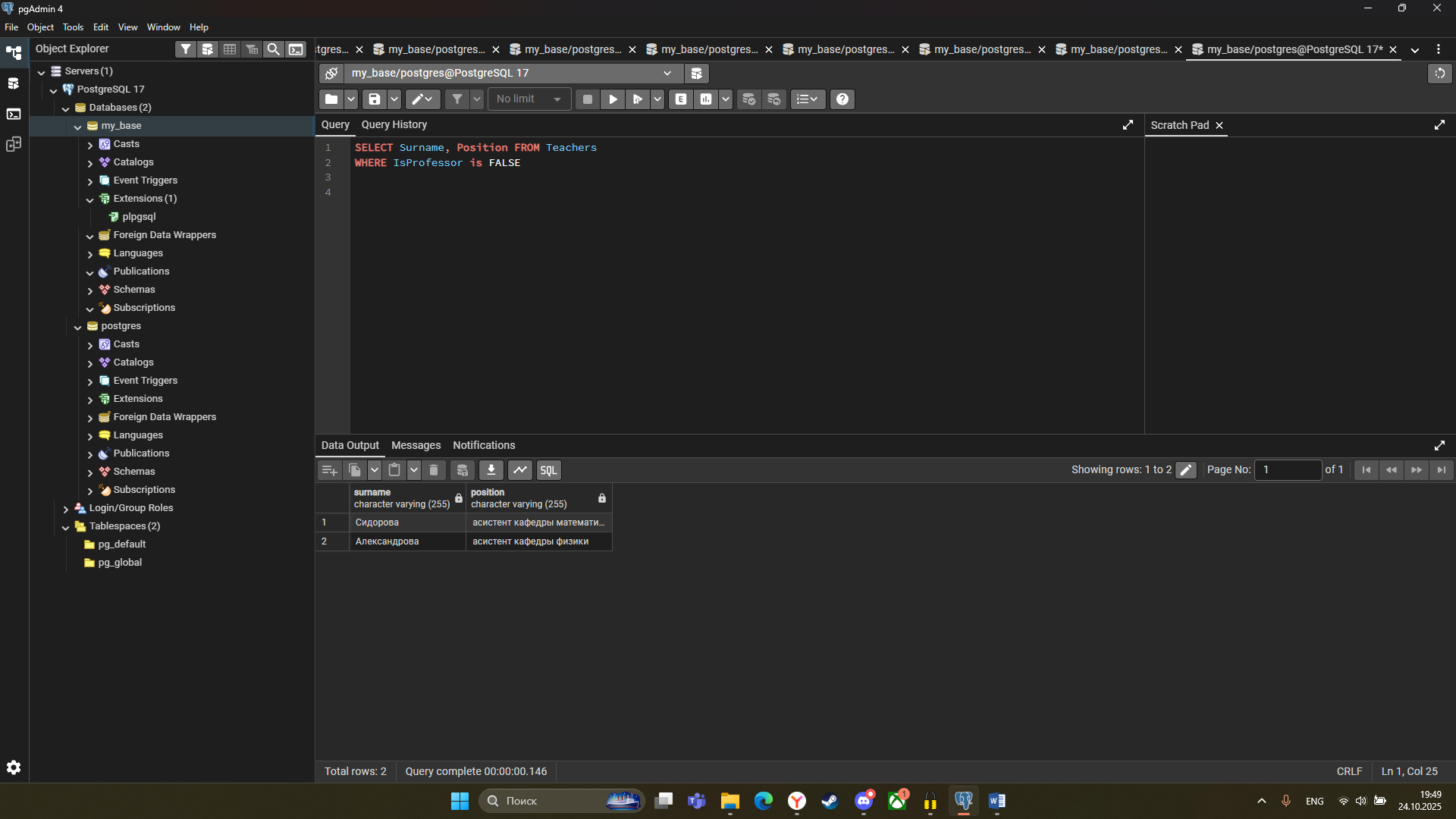
WHERE Name != 'Computer Science'



8. Вывести фамилии и должности преподавателей, которые не являются профессорами.

SELECT Surname, Position FROM Teachers

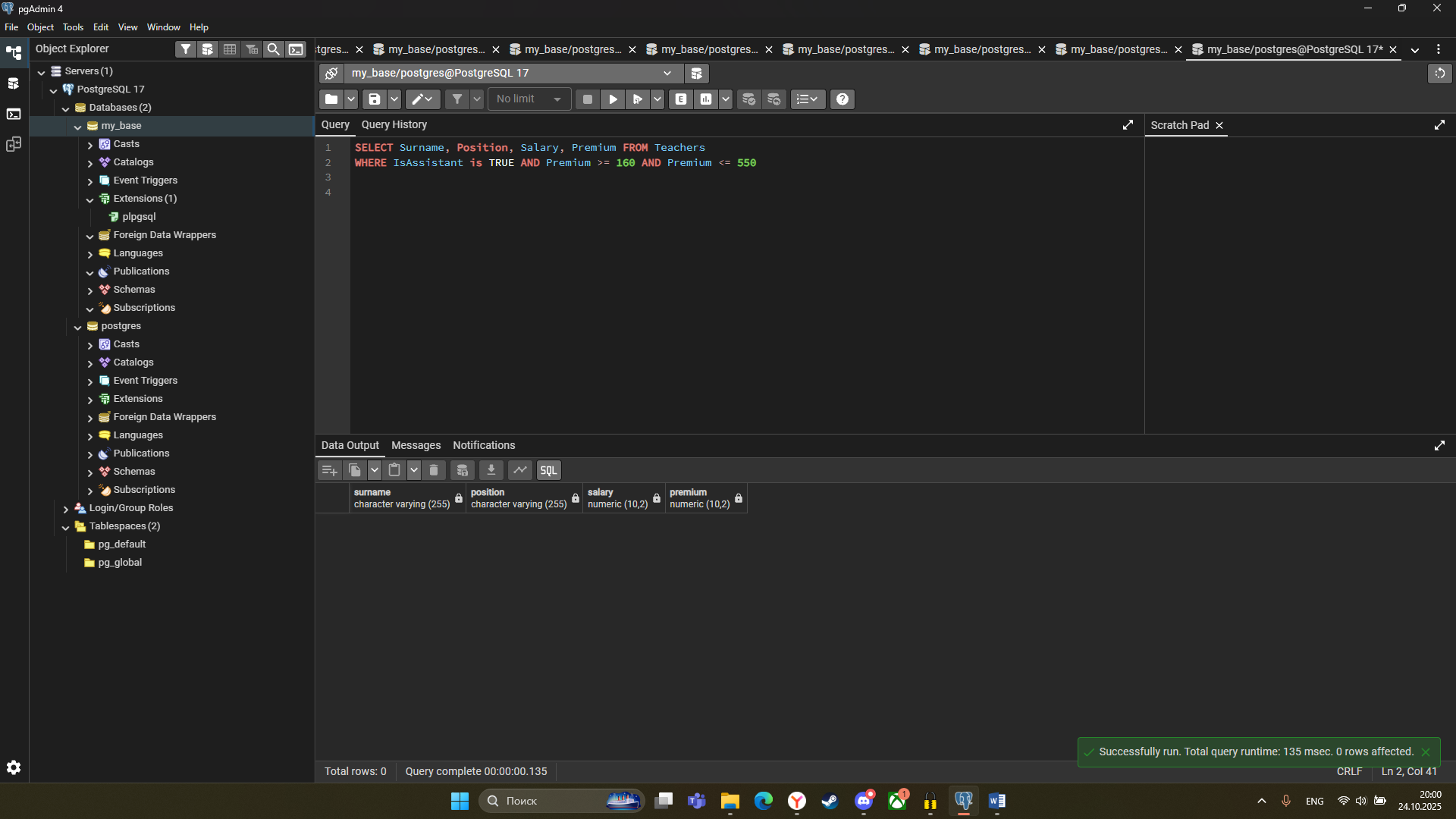
WHERE IsProfessor is FALSE



9. Вывести фамилии, должности, ставки и надбавки ассистентов, у которых надбавка в диапазоне от 160 до 550.

SELECT Surname, Position, Salary, Premium FROM Teachers

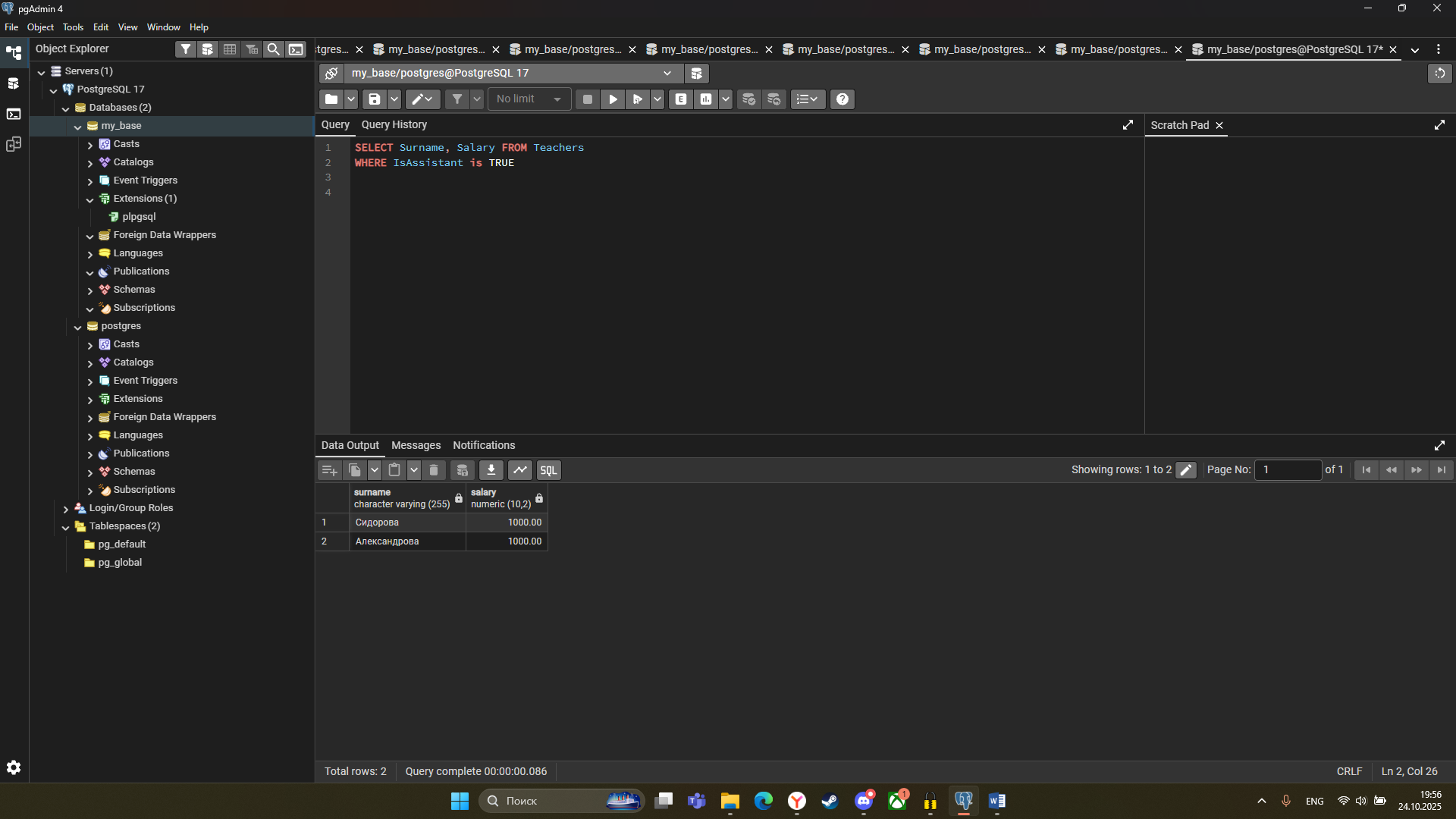
WHERE IsAssistant is TRUE AND Premium >= 160 AND Premium <= 550



10. Вывести фамилии и ставки ассистентов.

SELECT Surname, Salary FROM Teachers

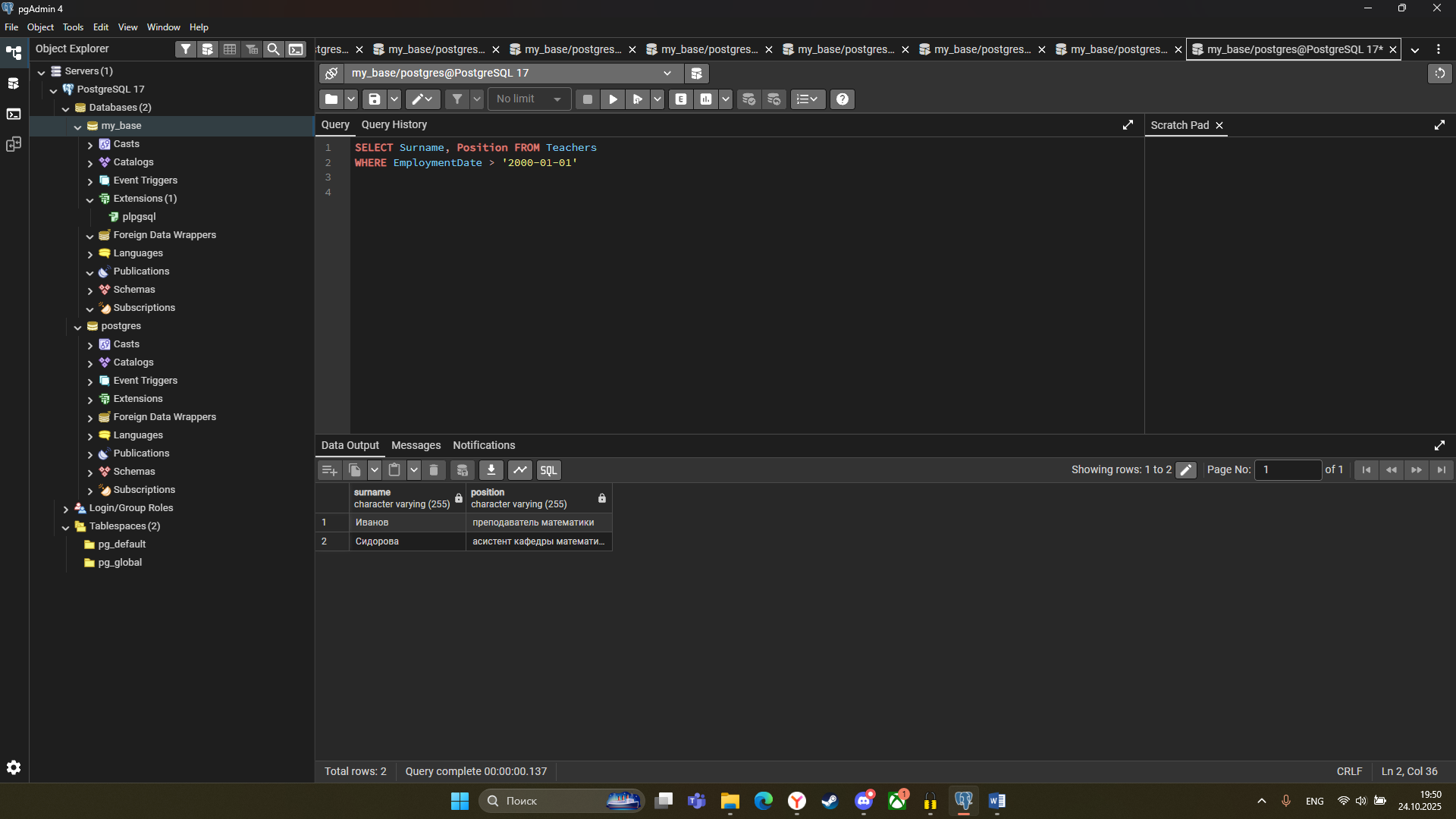
WHERE IsAssistant is TRUE



11. Вывести фамилии и должности преподавателей, которые были приняты на работу до 01.01.2000.

SELECT Surname, Position FROM Teachers

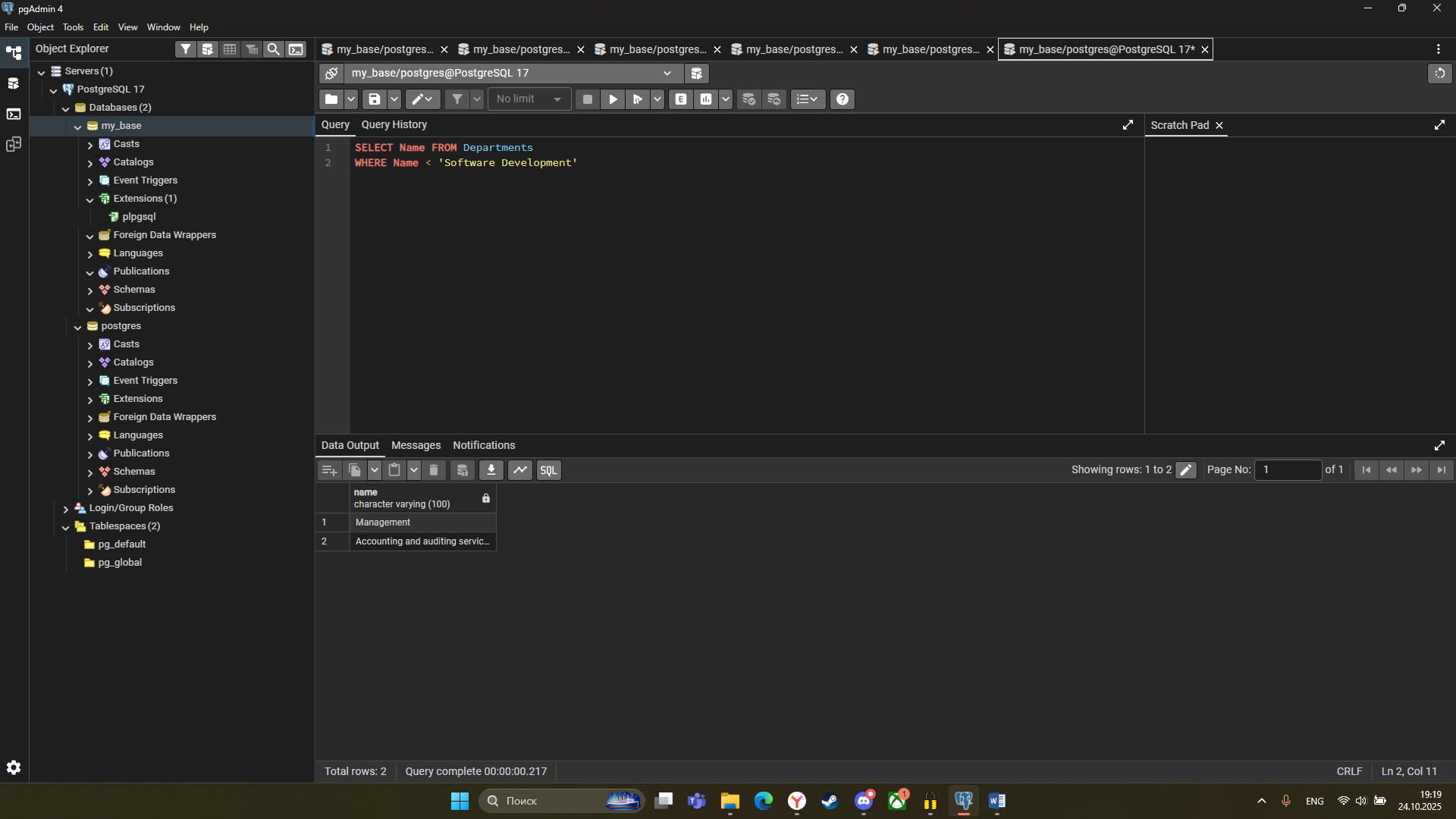
WHERE EmploymentDate > '2000-01-01'



12. Вывести названия кафедр, которые в алфавитном порядке располагаются до кафедры “Software Development”. Выводимое поле должно иметь название “Name of De part ment”.

SELECT Name FROM Departments

WHERE Name < 'Software Development'



13. Вывести фамилии ассистентов, имеющих зарплату (сумма ставки и надбавки) не более 1200.

SELECT Surname FROM Teachers

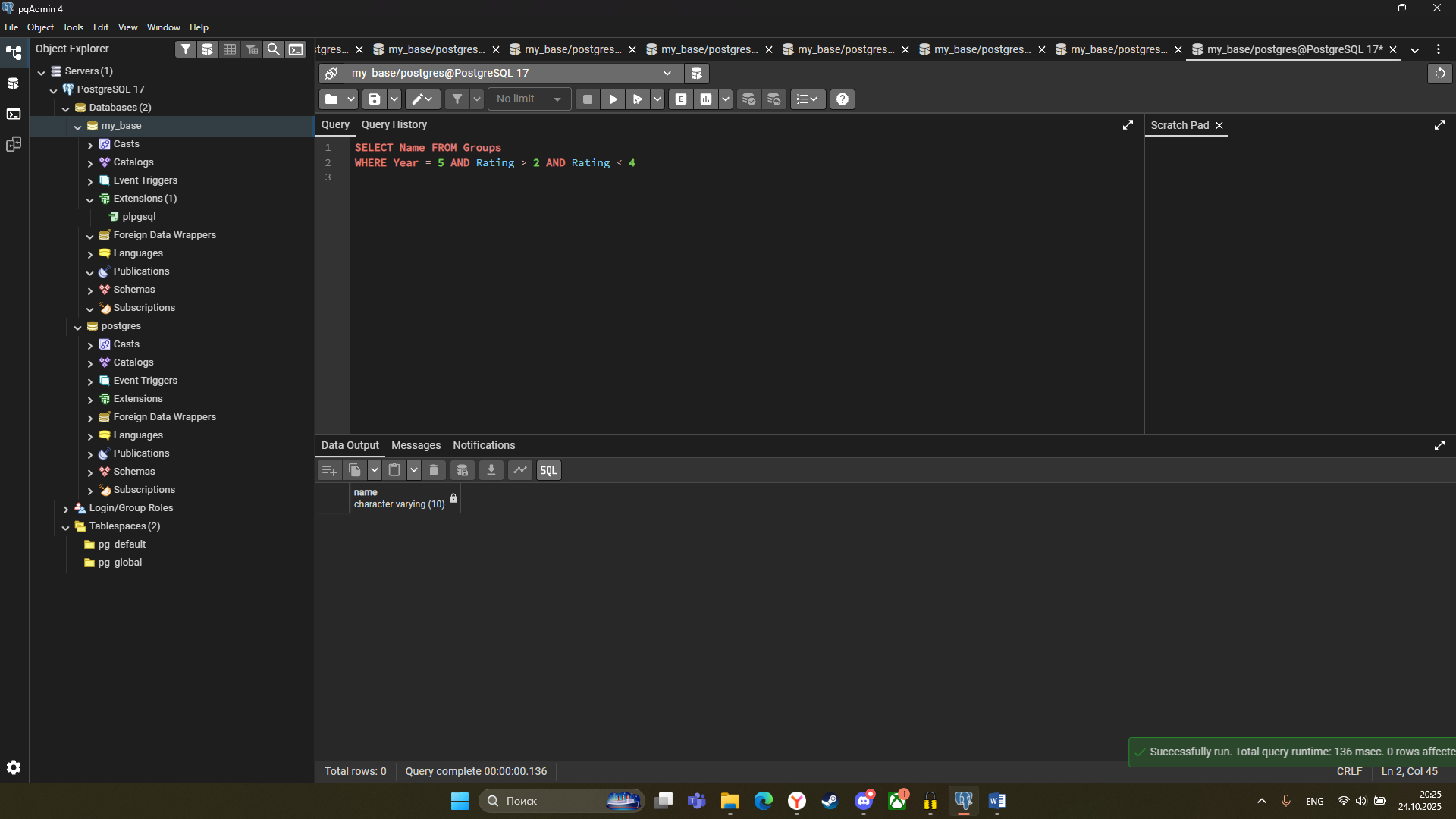
WHERE IsAssistant is TRUE AND Premium + Salary <= 1200



14. Вывести названия групп 5-го курса, имеющих рейтинг в диапазоне от 2 до 4.

SELECT Name FROM Groups

WHERE Year = 5 AND Rating > 2 AND Rating < 4



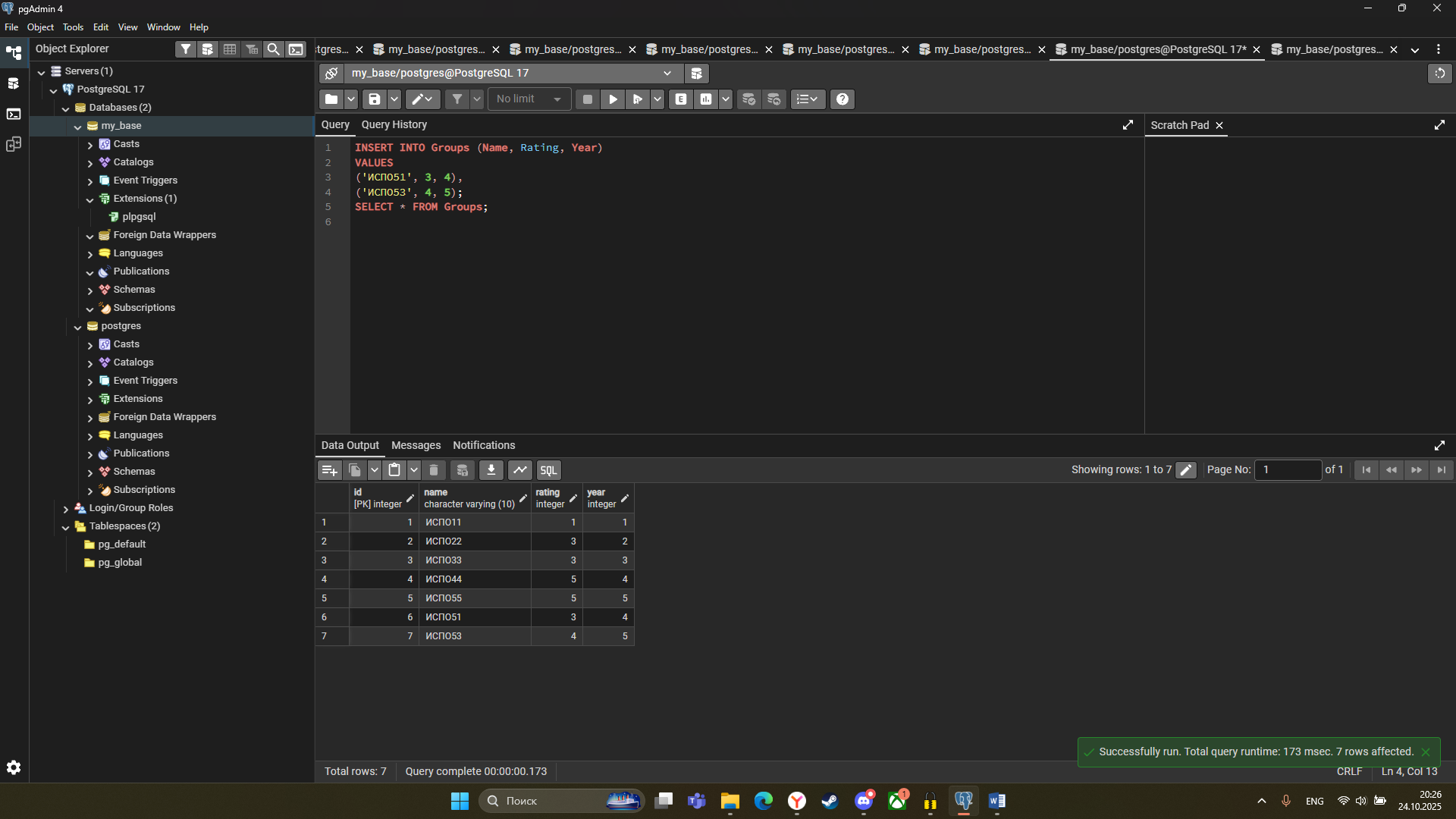
INSERT INTO Groups (Name, Rating, Year)

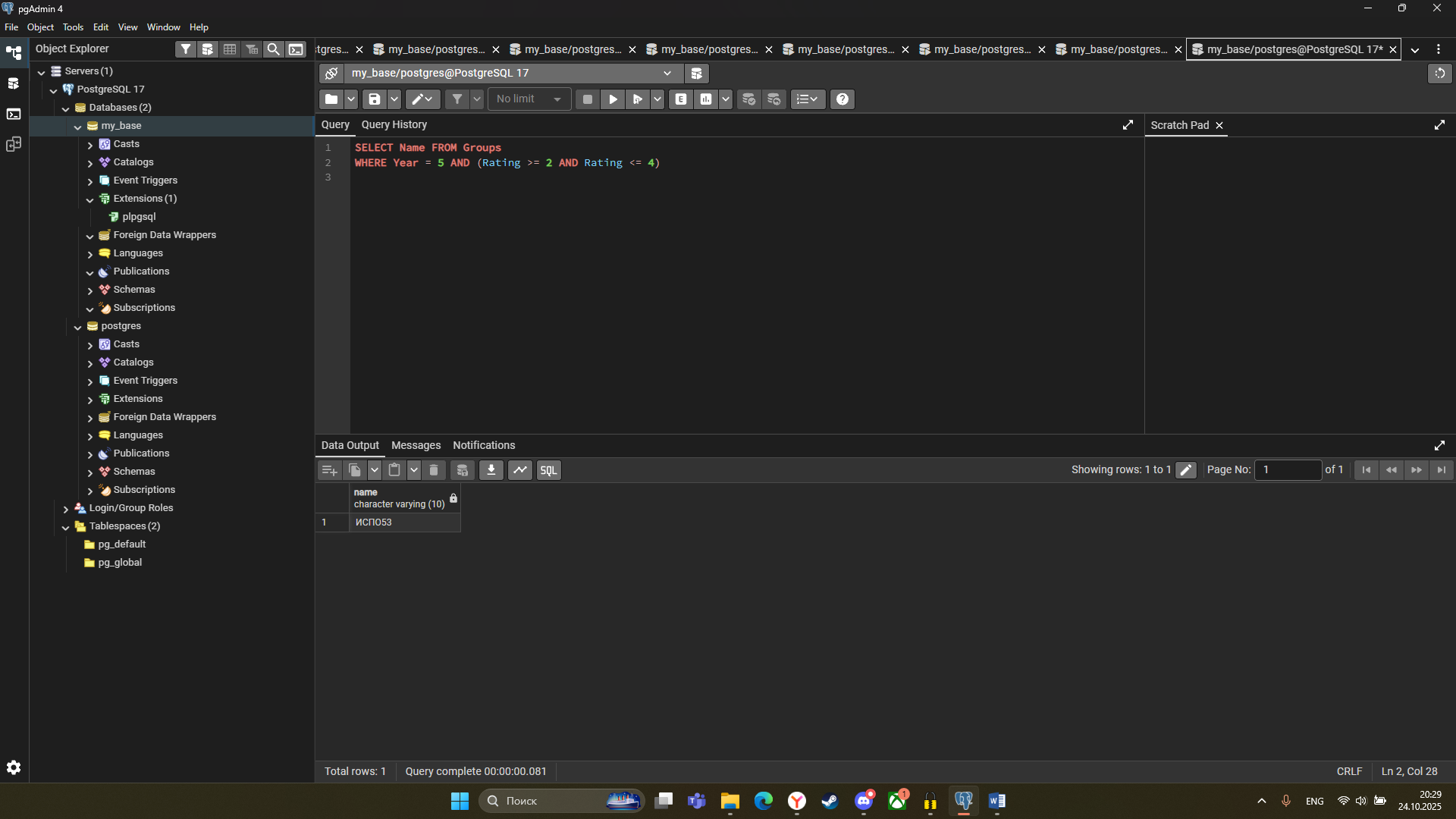
VALUES

('ИСПО51', 3, 4),

('ИСПО53', 4, 5);

SELECT \* FROM Groups;





15. Вывести фамилии ассистентов со ставкой меньше 550 или надбавкой меньше 200.

SELECT Surname FROM Teachers

WHERE IsAssistant is TRUE AND (Salary < 550 OR Premium < 200)

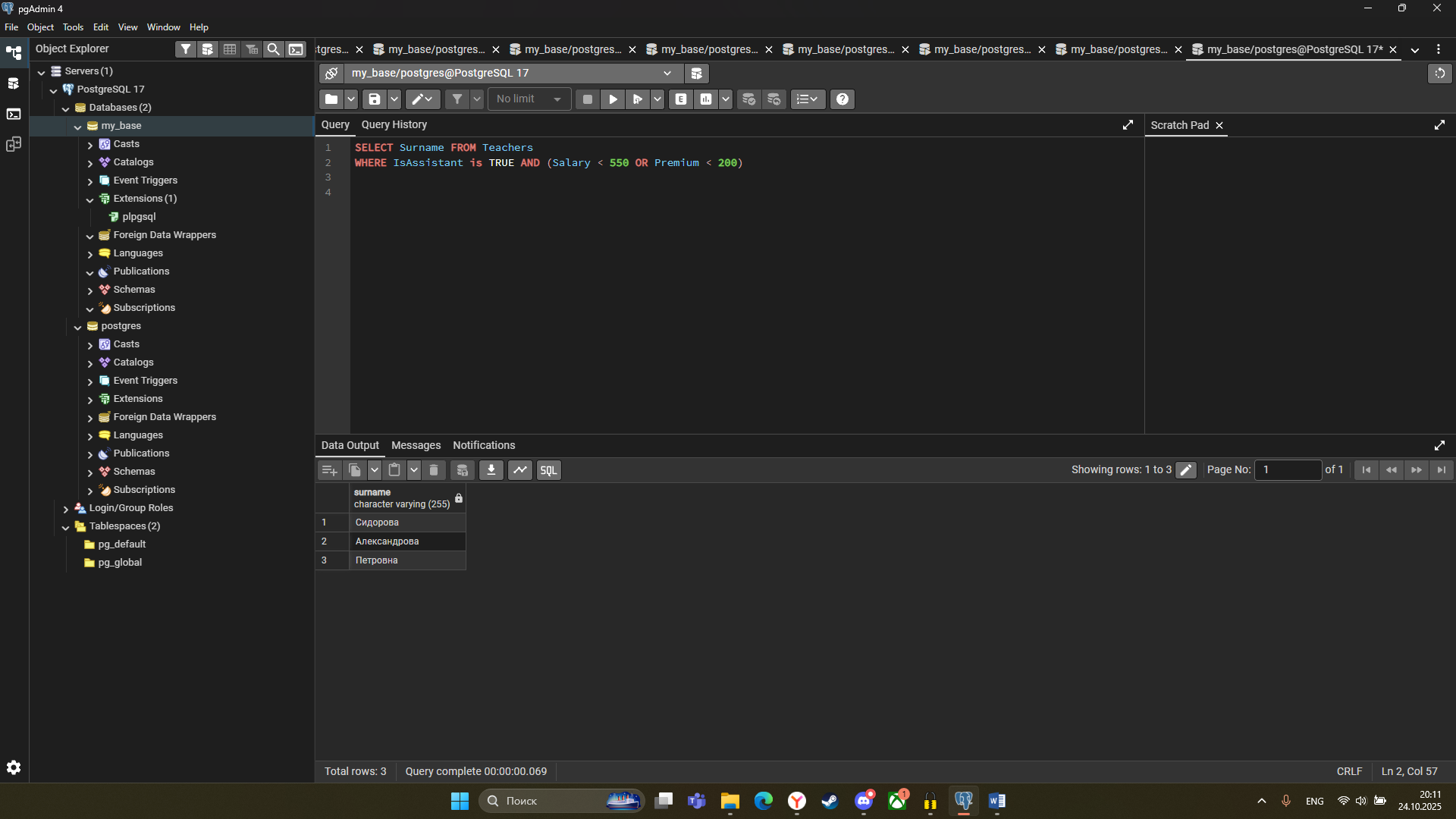


Схема базы данных

К этому заданию мы добавили скрипт для создания структуры базы данных для работы в рамках текущей темы.

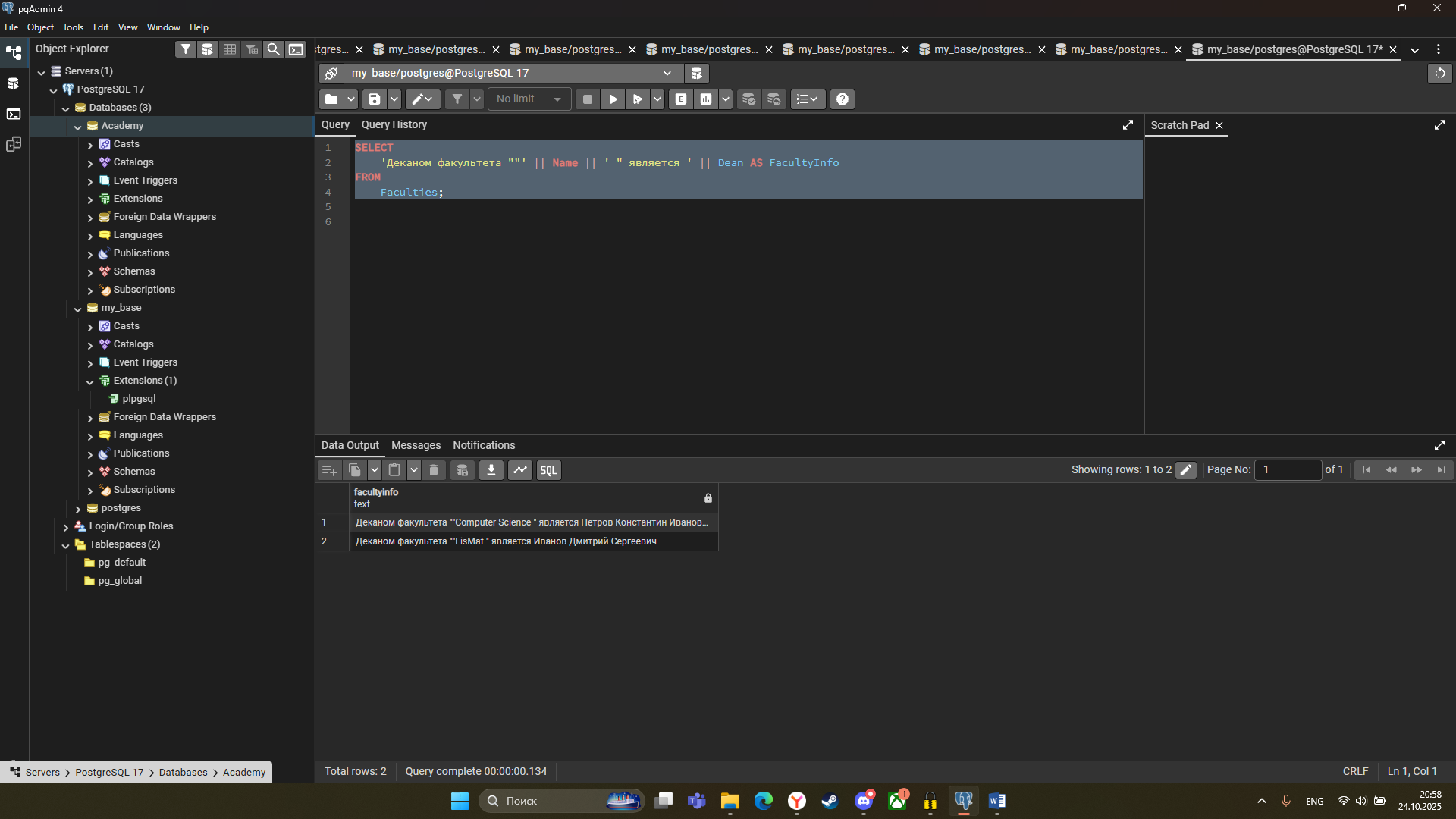
Мы категорически рекомендуем вам создать базу данных самостоятельно, без этого скрипта. Но если у вас возникнет крайняя необходимость вы можете его использовать.

SQL файл с базой данных прикреплен к данному PDF-файлу.

Для доступа к материалу, задание необходимо открыть в программе Adobe Acrobat Reader.

Описание

База данных Академия (Academy) содержит информацию о сотрудниках и внутреннем устройстве академии.



Преподаватели, читающие лекции в академии представлены в виде таблицы Преподаватели (Teachers), в которой собрана основная информация, такая как: имя, фамилия, данные о зарплате, а также дата приема на работу.

Также в базе данных присутствует информация о группах, хранимая в таблице Группы (Groups). Данные об факультетах и кафедрах содержатся в таблицах Факультеты (Faculties) и Кафедры (Departments) соответственно.

Таблицы

Ниже представлено детальное описание структуры каждой таблицы.

1. Кафедры (Departments)

CREATE TABLE Departments

(

Id SERIAL PRIMARY KEY NOT NULL,

Financing DECIMAL(10, 2) NOT NULL CHECK(Financing >= 0) DEFAULT 0,

Name VARCHAR(100) NOT NULL UNIQUE CHECK(Name <> '')

);

SELECT \* FROM Departments;

■ Идентификатор (Id). Уникальный идентификатор кафедры.

▷ Тип данных — int.

▷ Авто приращение.

▷ Не может содержать null-значения.

▷ Первичный ключ.

■ Финансирование (Financing). Фонд финансирования кафедры.

▷ Тип данных — money.

▷ Не может содержать null-значения.

▷ Не может быть меньше 0.

▷ Значение по умолчанию — 0.

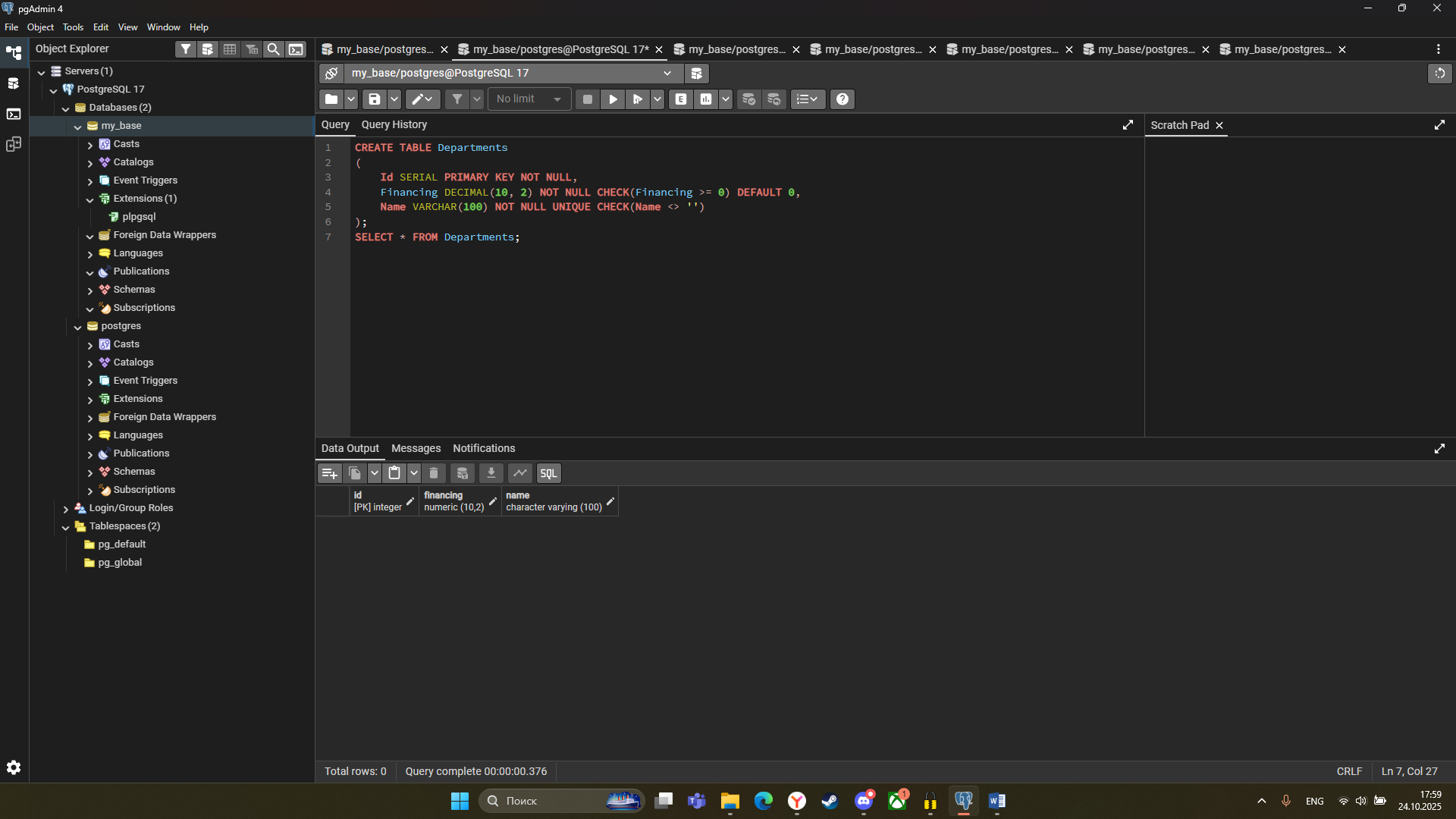
■ Название (Name). Название кафедры.

▷ Тип данных — nvarchar(100).

▷ Не может содержать null-значения.

▷ Не может быть пустым.

▷ Должно быть уникальным.



INSERT INTO Departments (Financing, Name)

VALUES

('10000', 'Software Development'),

('30000', 'Management'),

('100000', 'Accounting and auditing services'),

('200000', 'thermal power engineering');

SELECT \* FROM Departments;



1. Факультеты (Faculties)

CREATE TABLE Faculties

(

Id SERIAL PRIMARY KEY NOT NULL,

Dean VARCHAR(255) NOT NULL CHECK(Dean <> ''),

Name VARCHAR(100) NOT NULL UNIQUE CHECK(Name <> '')

);

SELECT \* FROM Faculties;

■ Идентификатор (Id). Уникальный идентификатор фа

культета.

▷ Тип данных — int.

▷ Авто приращение.

▷ Не может содержать null-значения.

▷ Первичный ключ.

■ Декан (Dean). Декан факультета.

▷ Тип данных — nvarchar(max).

▷ Не может содержать null-значения.

▷ Не может быть пустым.

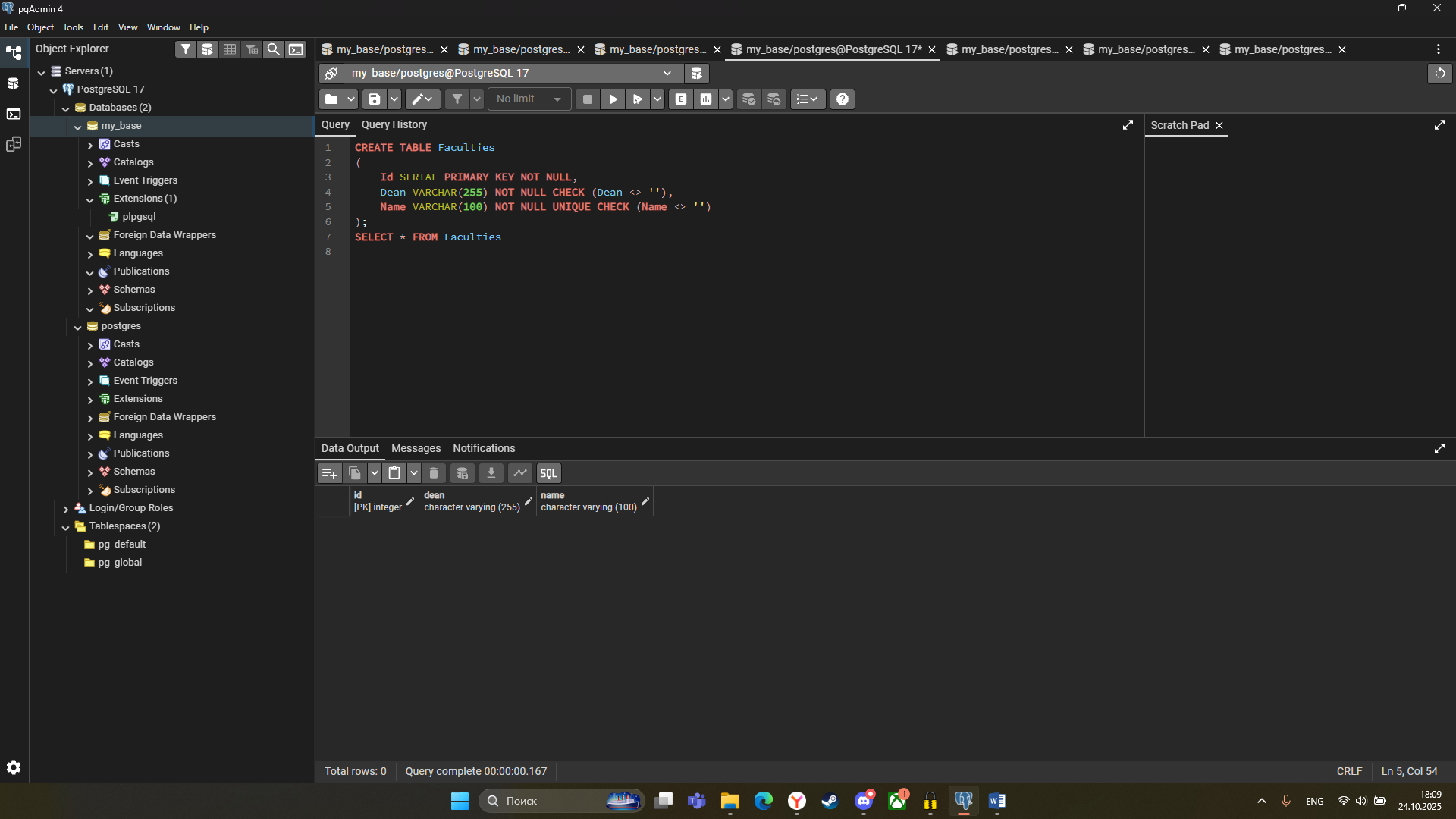
■ Название (Name). Название факультета.

▷ Тип данных — nvarchar(100).

▷ Не может содержать null-значения.

▷ Не может быть пустым.

▷ Должно быть уникальным.

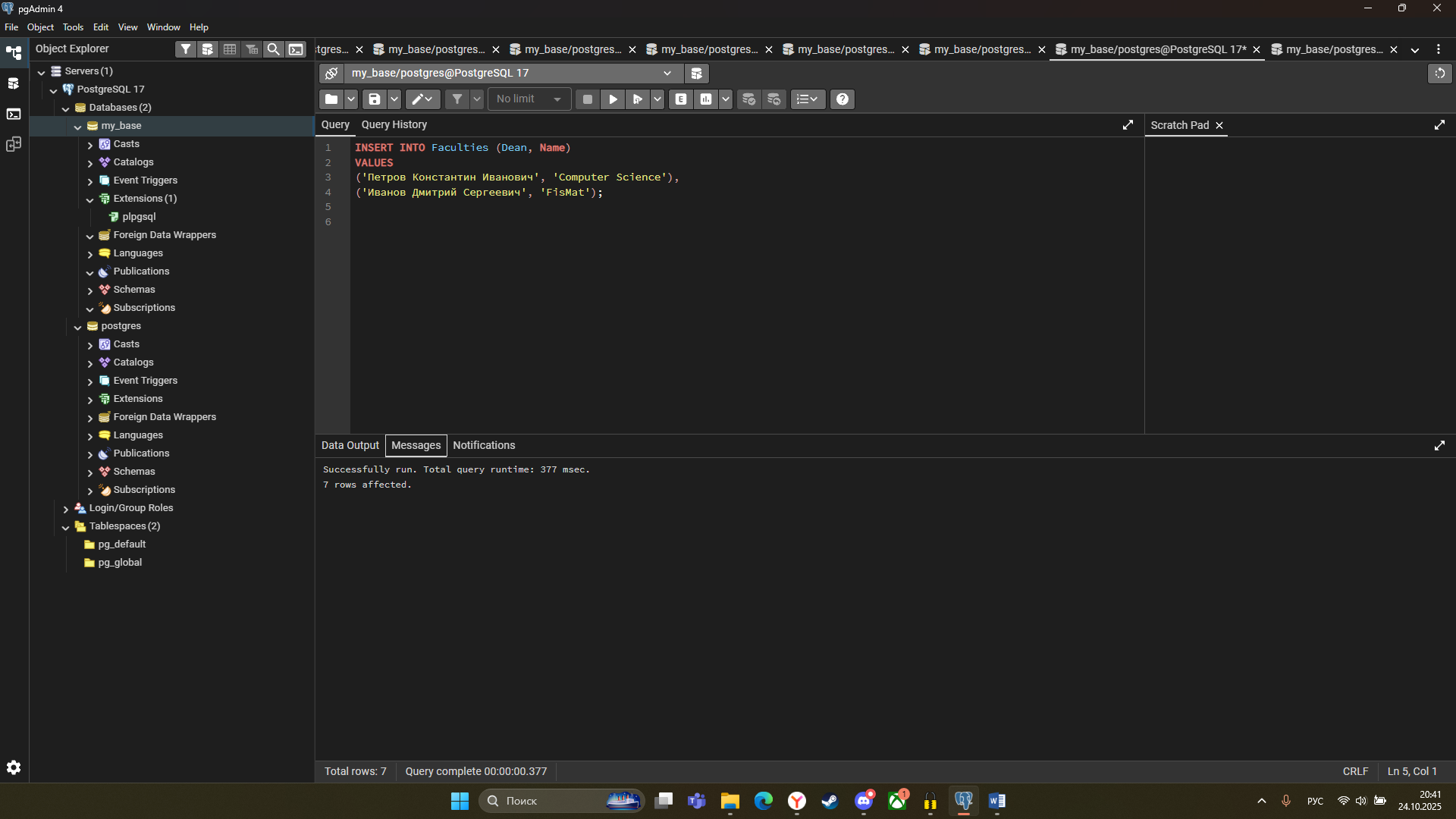


INSERT INTO Faculties (Dean, Name)

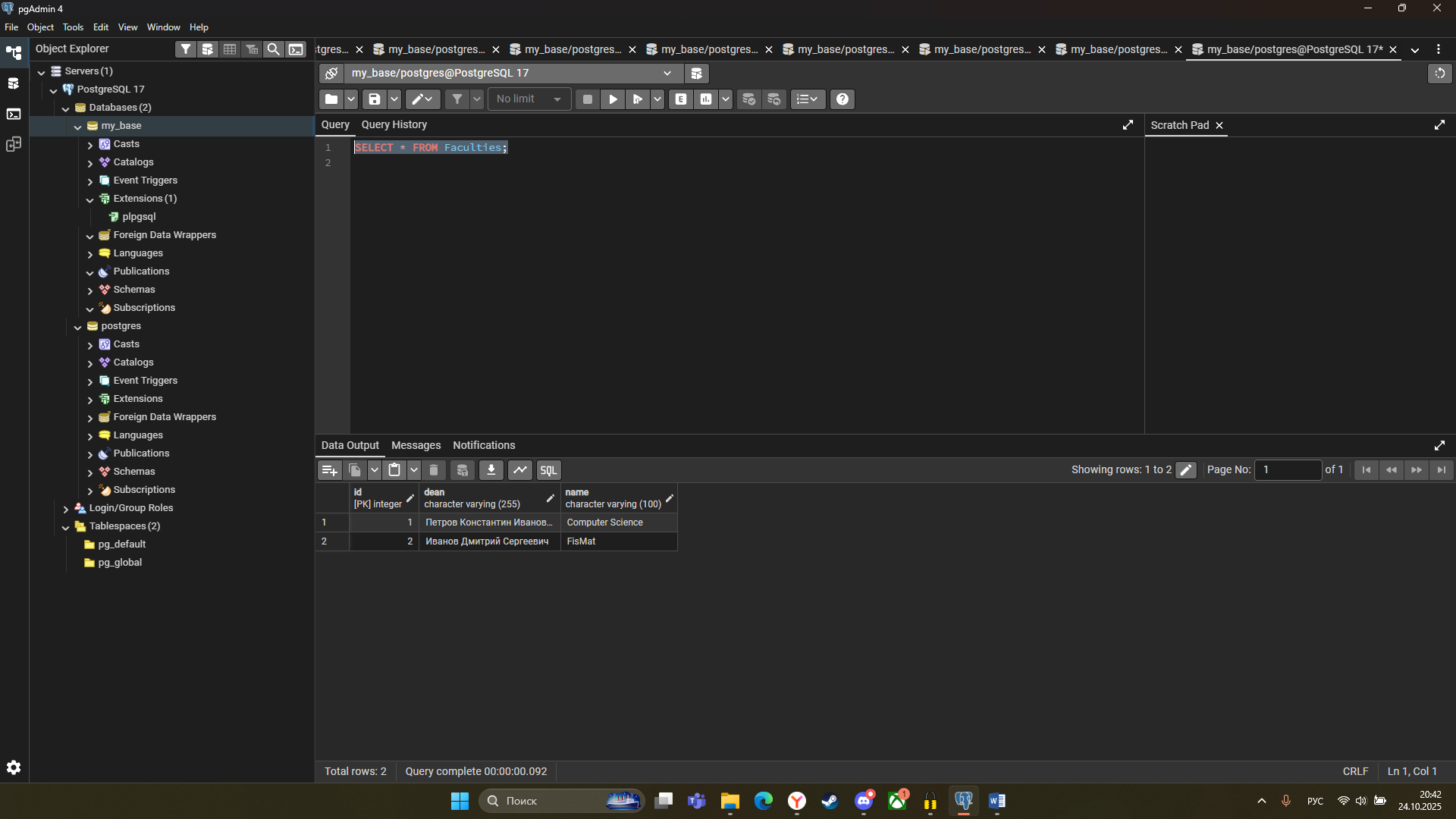
VALUES

('Петров Константин Иванович', 'Computer Science'),

('Иванов Дмитрий Сергеевич', 'FisMat');



SELECT \* FROM Faculties;



1. Группы (Groups)

CREATE TABLE Groups

(

Id SERIAL PRIMARY KEY NOT NULL,

Name VARCHAR(10) NOT NULL UNIQUE CHECK(Name <> ''),

Rating INT NOT NULL CHECK (Rating >= 0 AND Rating <= 5),

Year INT NOT NULL CHECK (Year >= 1 AND Year <= 5)

);

SELECT \* FROM Groups

■ Идентификатор (Id). Уникальный идентификатор

группы.

▷ Тип данных — int.

▷ Авто приращение.

▷ Не может содержать null-значения.

▷ Первичный ключ.

■ Название (Name). Название группы.

▷ Тип данных — nvarchar(10).

▷ Не может содержать null-значения.

▷ Не может быть пустым.

▷ Должно быть уникальным.

■ Рейтинг (Rating). Рейтинг группы.

▷ Тип данных — int.

▷ Не может содержать null-значения.

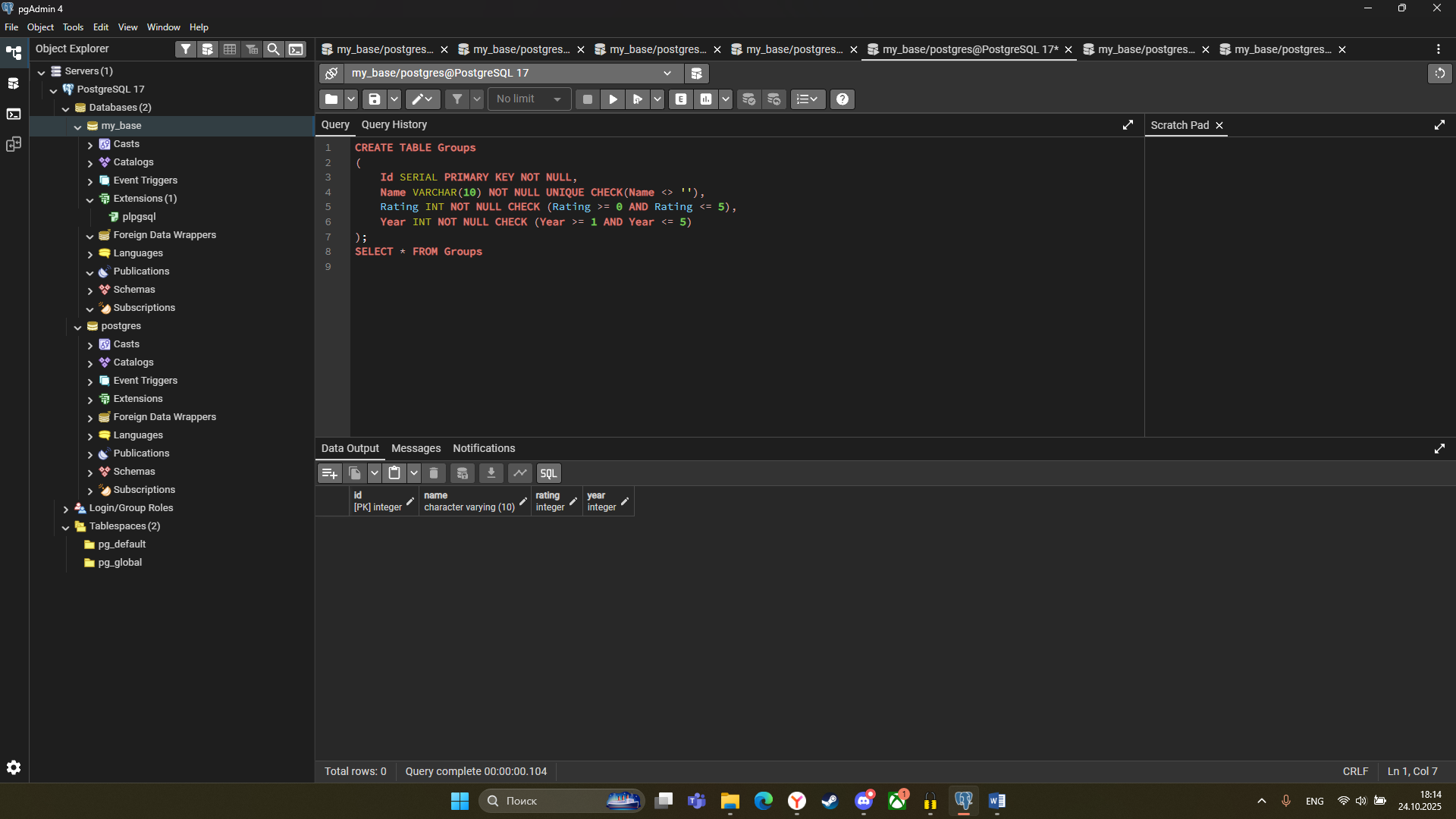
▷ Должно быть в диапазоне от 0 до 5.

■ Курс (Year). Курс (год) на котором обучается группа.

▷ Тип данных — int.

▷ Не может содержать null-значения.

▷ Должно быть в диапазоне от 1 до 5.



INSERT INTO Groups (Name, Rating, Year)

VALUES

('ИСПО11', 1, 1),

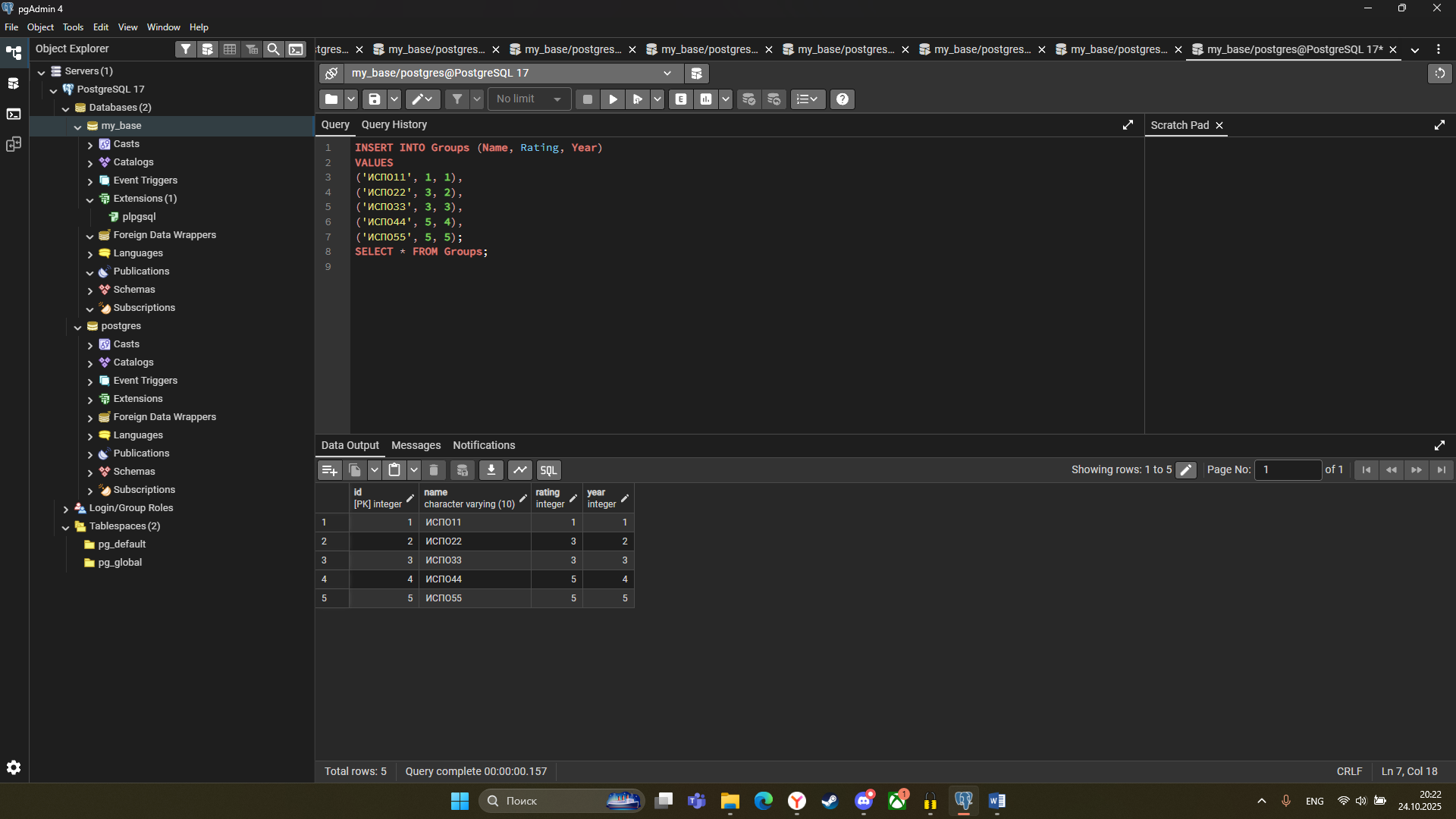
('ИСПО22', 3, 2),

('ИСПО33', 3, 3),

('ИСПО44', 5, 4),

('ИСПО55', 5, 5);

SELECT \* FROM Groups;



1. Преподаватели (Teachers)

CREATE TABLE Teachers

(

Id SERIAL PRIMARY KEY NOT NULL,

EmploymentDate DATE NOT NULL CHECK(EmploymentDate >= '1990-01-01'),

IsAssistant BOOLEAN NOT NULL DEFAULT FALSE,

IsProfessor BOOLEAN NOT NULL DEFAULT FALSE,

Name VARCHAR(255) NOT NULL CHECK(Name <> ''),

Position VARCHAR(255) NOT NULL CHECK(Name <> ''),

Premium DECIMAL(10, 2) NOT NULL CHECK(Premium >= 0) DEFAULT 0,

Salary DECIMAL(10, 2) NOT NULL CHECK(Premium > 0),

Surname VARCHAR(255) NOT NULL CHECK(Name <> '')

);

SELECT \* FROM Teachers

■ Идентификатор (Id). Уникальный идентификатор преподавателя.

▷ Тип данных — int.

▷ Авто приращение.

▷ Не может содержать null-значения.

▷ Первичный ключ.

■ Дата трудоустройства (EmploymentDate). Дата приема преподавателя на работу.

▷ Тип данных — date.

▷ Не может содержать null-значения.

▷ Не может быть меньше 01.01.1990.

■ Ассистент (IsAssistant). Является ли преподаватель ассистентом.

▷ Тип данных — bit.

▷ Не может содержать null-значения.

▷ Значение по умолчанию — 0.

■ Профессор (IsProfessor). Является ли преподаватель профессором.

▷ Тип данных — bit.

▷ Не может содержать null-значения.

▷ Значение по умолчанию — 0.

■ Имя (Name). Имя преподавателя.

▷ Тип данных — nvarchar(max).

▷ Не может содержать null-значения.

▷ Не может быть пустым.

■ Должность (Position). Должность преподавателя.

▷ Тип данных — nvarchar(max).

▷ Не может содержать null-значения.

▷ Не может быть пустым.

■ Надбавка (Premium). Надбавка преподавателя.

▷ Тип данных — money.

▷ Не может содержать null-значения.

▷ Не может быть меньше 0.

▷ Значение по умолчанию — 0.

■ Ставка (Salary). Ставка преподавателя.

▷ Тип данных — money.

▷ Не может содержать null-значения.

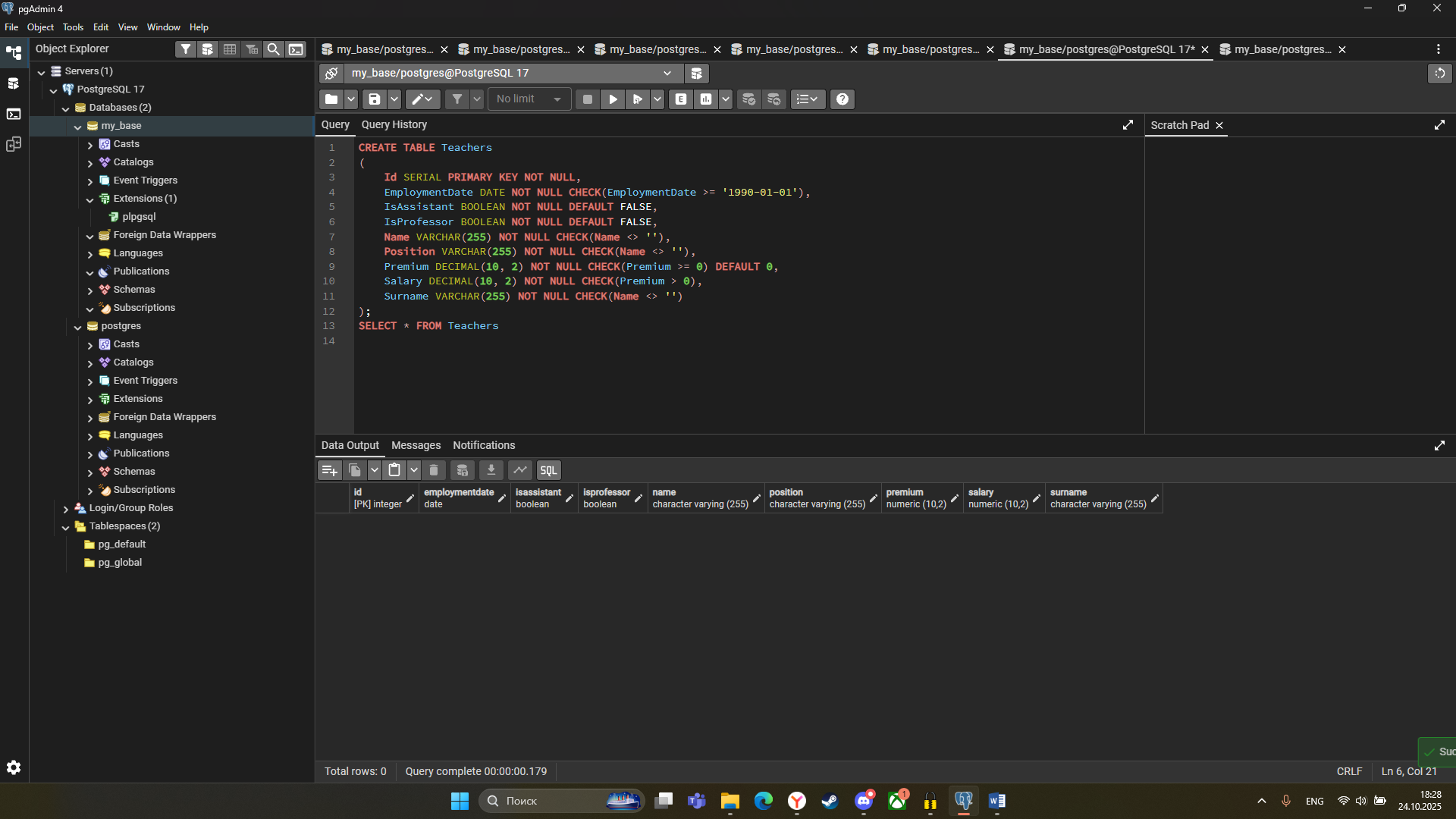
▷ Не может быть меньше либо равно 0.

■ Фамилия (Surname). Фамилия преподавателя.

▷ Тип данных — nvarchar(max).

▷ Не может содержать null-значения.

▷ Не может быть пустым



INSERT INTO Teachers (EmploymentDate, IsAssistant, IsProfessor, Name, Position, Premium, Salary, Surname)

VALUES

('2005-10-10', FALSE, TRUE, 'Иван', 'преподаватель математики', 1.5, 2000, 'Иванов'),

('1990-05-10', FALSE, TRUE, 'Петр', 'заведующий кафедрой математики', 2, 3000, 'Петров'),

('2010-03-07', TRUE, FALSE, 'Ольга', 'асистент кафедры математики', 0.8, 1000, 'Сидорова'),

('1996-07-03', TRUE, FALSE, 'Ирина', 'асистент кафедры физики', 0.9, 1000, 'Александрова');

SELECT \* FROM Teachers;

