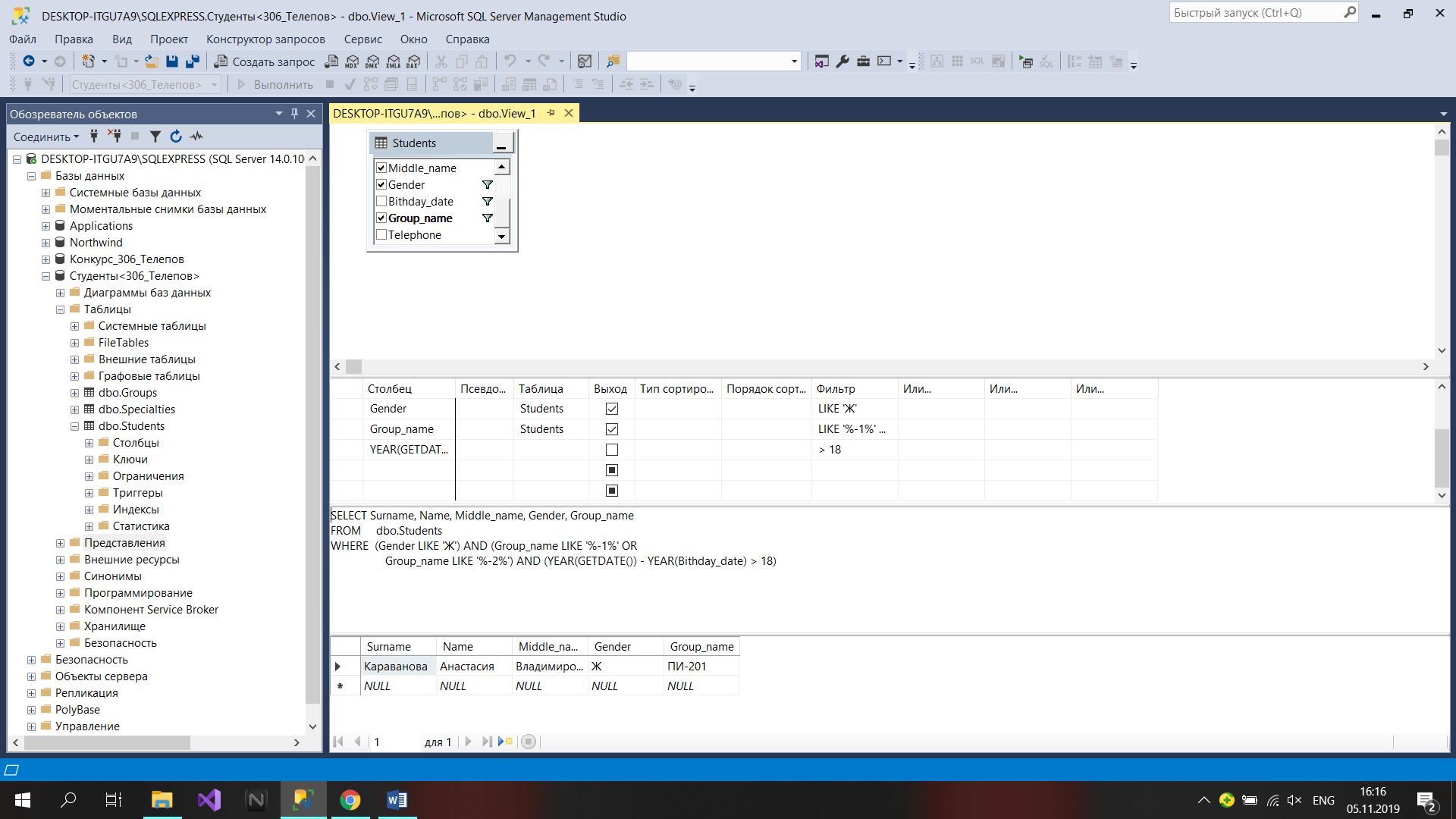
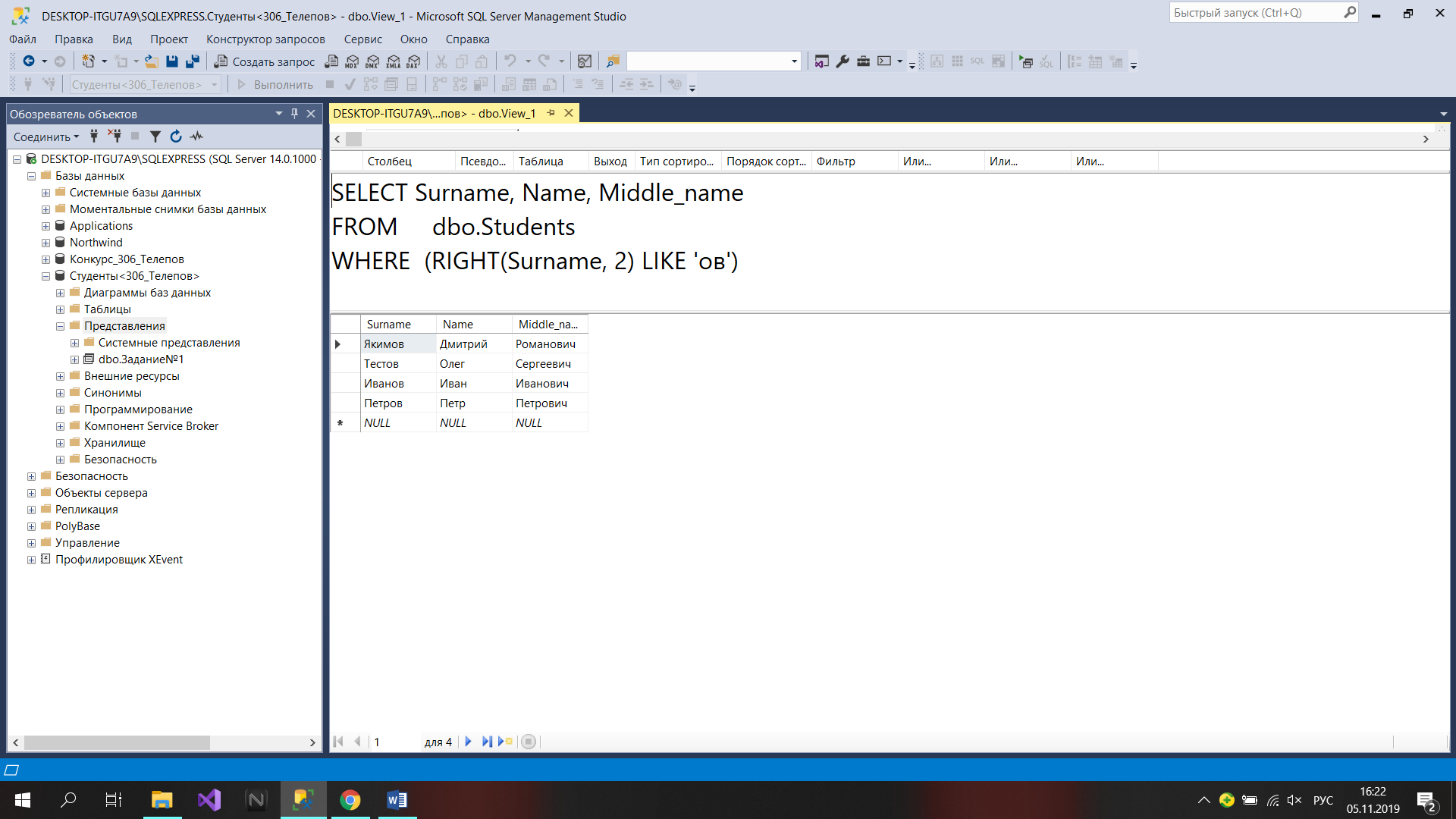
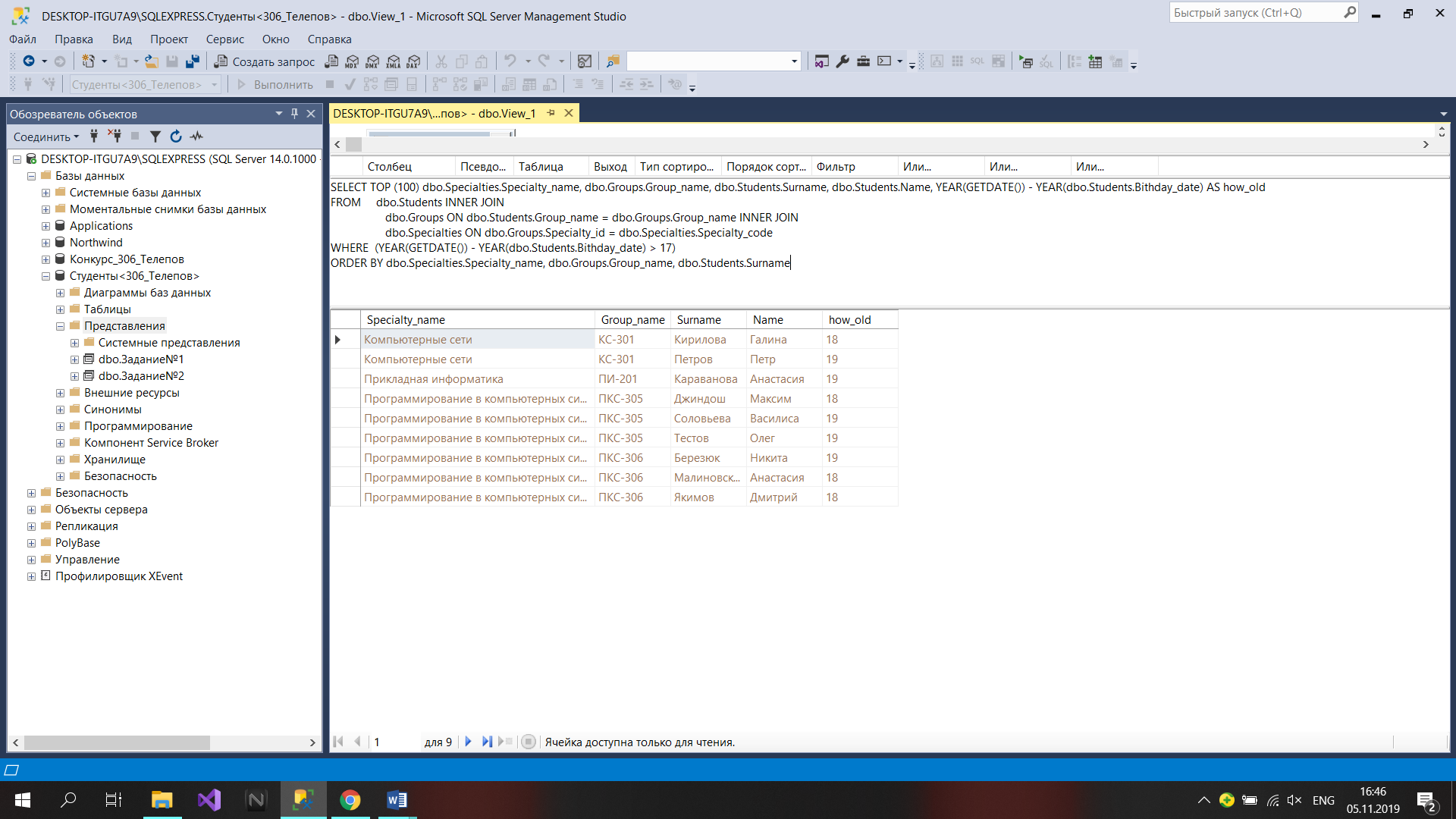
**Практическая работа №2**

**Тема:** «Виды запросов»

**Цель:** « Научиться создавать различные виды запросов».

**Задание**

**Вариант №2**

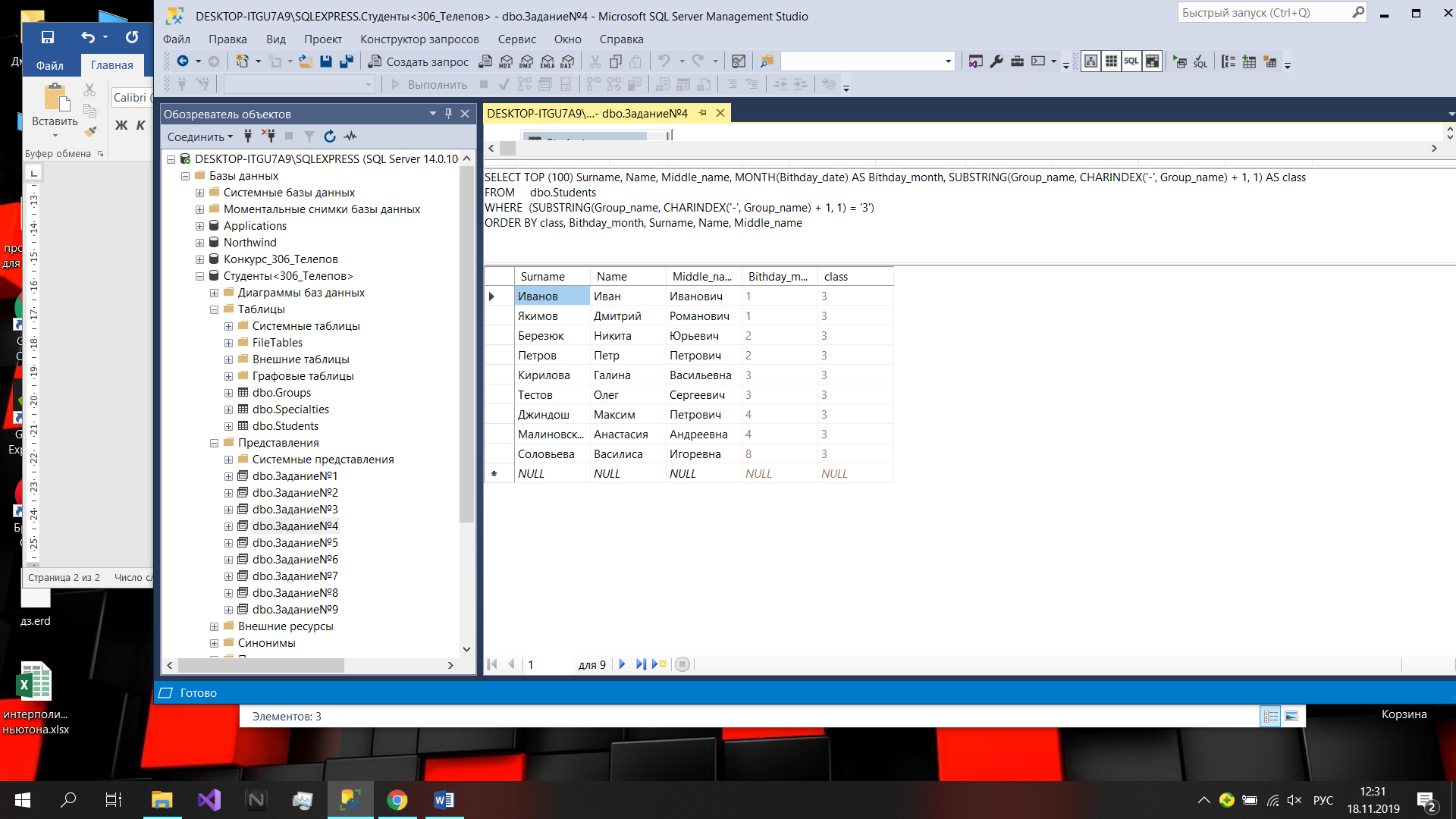
1. Вывести студентов женского пола 1-2 курсов моложе 18 лет.  
   
2. Вывести фамилии студентов, в которых окончание(2 последние буквы фамилии) совпадает с окончанием Вашей фамилии.  
   
3. вывести список студентов в возрасте старше 17 лет в виде: название специальности, группа, фамилия, имя, возраст, отсортированный по названию специальности, номеру группы, внутри группы по фамилии.  
     
   SELECT TOP (100) dbo.Specialties.Specialty\_name, dbo.Groups.Group\_name, dbo.Students.Surname, dbo.Students.Name, YEAR(GETDATE()) - YEAR(dbo.Students.Bithday\_date) AS how\_old  
   FROM dbo.Students INNER JOIN  
    dbo.Groups ON dbo.Students.Group\_name = dbo.Groups.Group\_name INNER JOIN  
    dbo.Specialties ON dbo.Groups.Specialty\_id = dbo.Specialties.Specialty\_code  
   WHERE (YEAR(GETDATE()) - YEAR(dbo.Students.Bithday\_date) > 17)  
   ORDER BY dbo.Specialties.Specialty\_name, dbo.Groups.Group\_name, dbo.Students.Surname  
   
4. Вывести список студентов Вашего курса с указанием их месяца рождения, отсортированный по курсу, месяцу рождения и фамилии, имени и отчеству.

SELECT TOP (100) Surname, Name, Middle\_name, MONTH(Bithday\_date) AS Bithday\_month, SUBSTRING(Group\_name, CHARINDEX('-', Group\_name) + 1, 1) AS class

FROM dbo.Students

WHERE (SUBSTRING(Group\_name, CHARINDEX('-', Group\_name) + 1, 1) = '3')

ORDER BY class, Bithday\_month, Surname, Name, Middle\_name

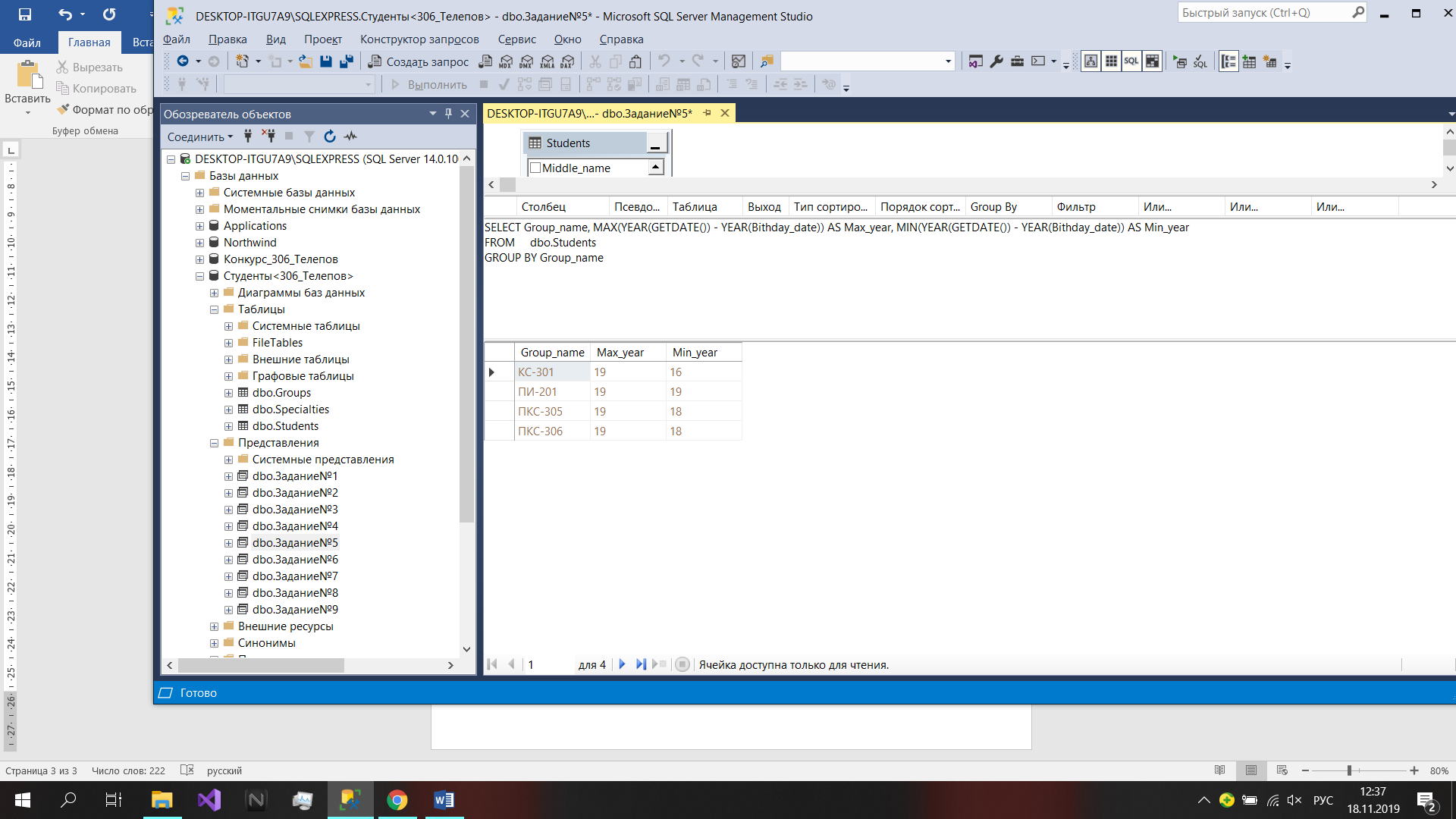


1. Посчитать минимальный и максимальный возраст студентов каждой группы.

SELECT Group\_name, MAX(YEAR(GETDATE()) - YEAR(Bithday\_date)) AS Max\_year, MIN(YEAR(GETDATE()) - YEAR(Bithday\_date)) AS Min\_year

FROM dbo.Students

GROUP BY Group\_name



1. Посчитать число студентов на каждой специальности, отсортировав по числу студентов в порядке убывания.

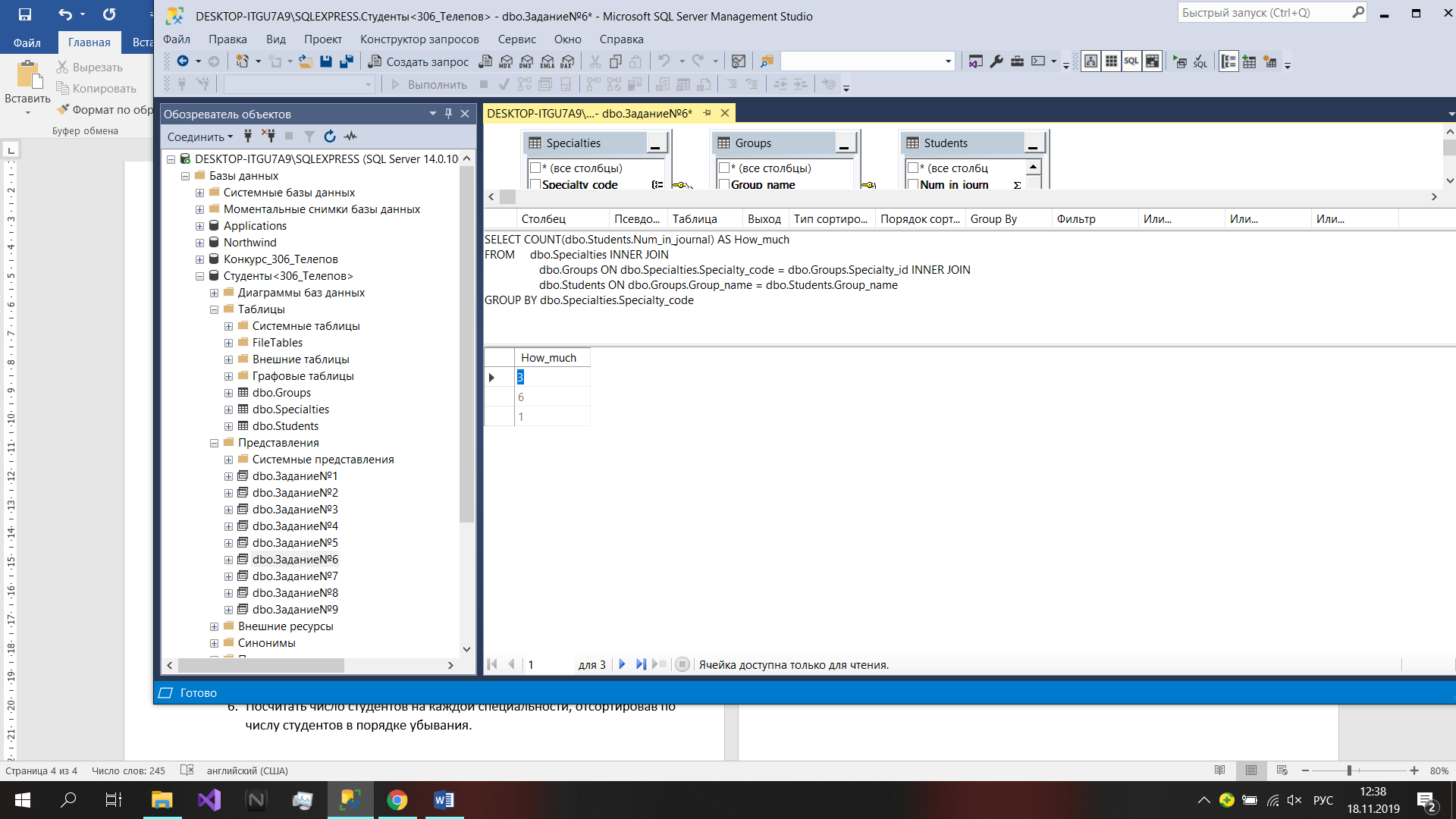
SELECT COUNT(dbo.Students.Num\_in\_journal) AS How\_much

FROM dbo.Specialties INNER JOIN

dbo.Groups ON dbo.Specialties.Specialty\_code = dbo.Groups.Specialty\_id INNER JOIN

dbo.Students ON dbo.Groups.Group\_name = dbo.Students.Group\_name

GROUP BY dbo.Specialties.Specialty\_code



1. Собственное уникальное представление с использованием DATEPART (Вывести дни недели рождения каждого студента)

SELECT CASE DATEPART(weekday, Bithday\_date)

WHEN 7 THEN 'Воскресенье'

WHEN 1 THEN 'Понедельник'

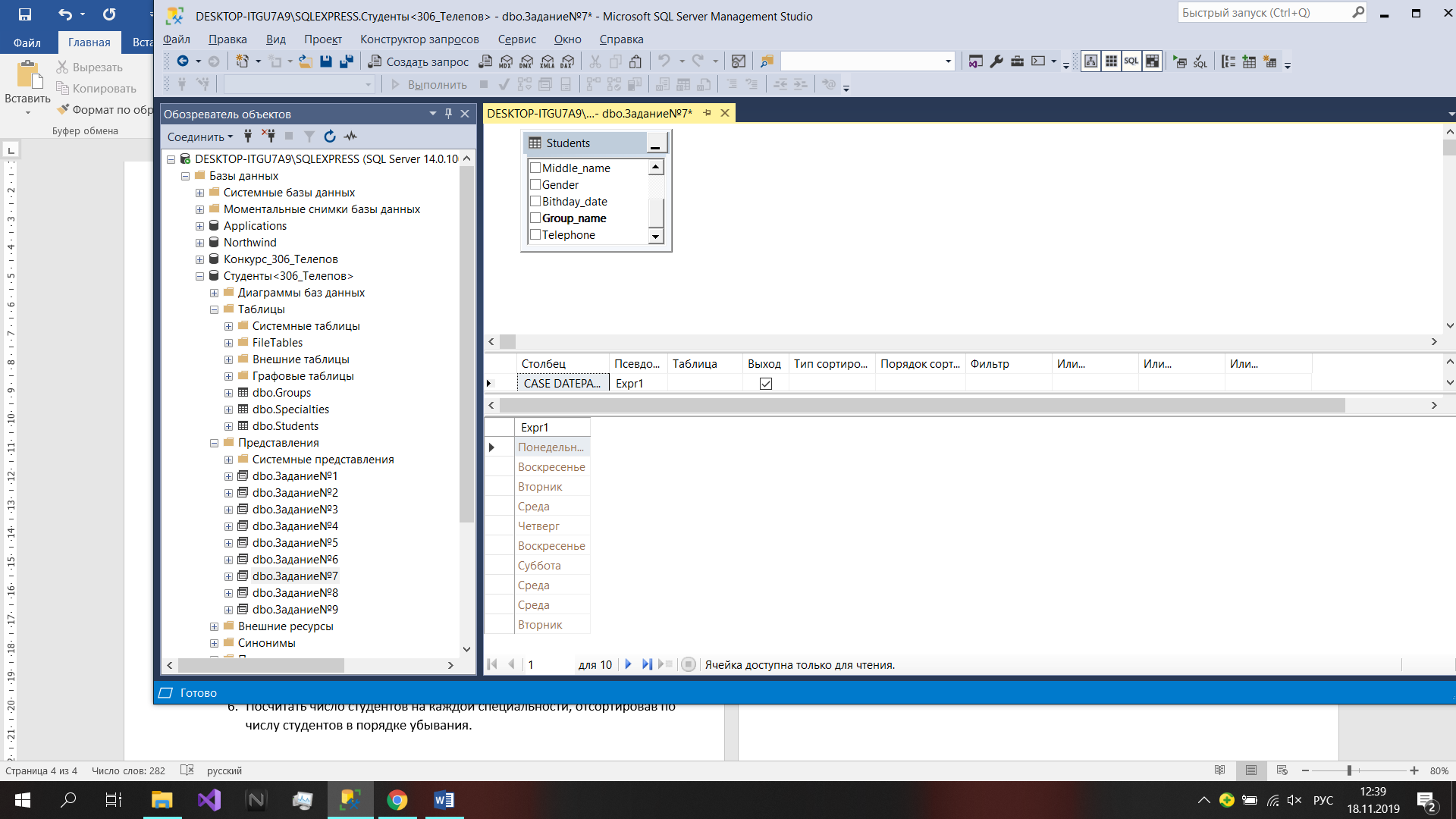
WHEN 2 THEN 'Вторник'

WHEN 3 THEN 'Среда'

WHEN 4 THEN 'Четверг'

WHEN 5 THEN 'Пятница' WHEN 6 THEN 'Суббота' END AS Expr1

FROM dbo.Students

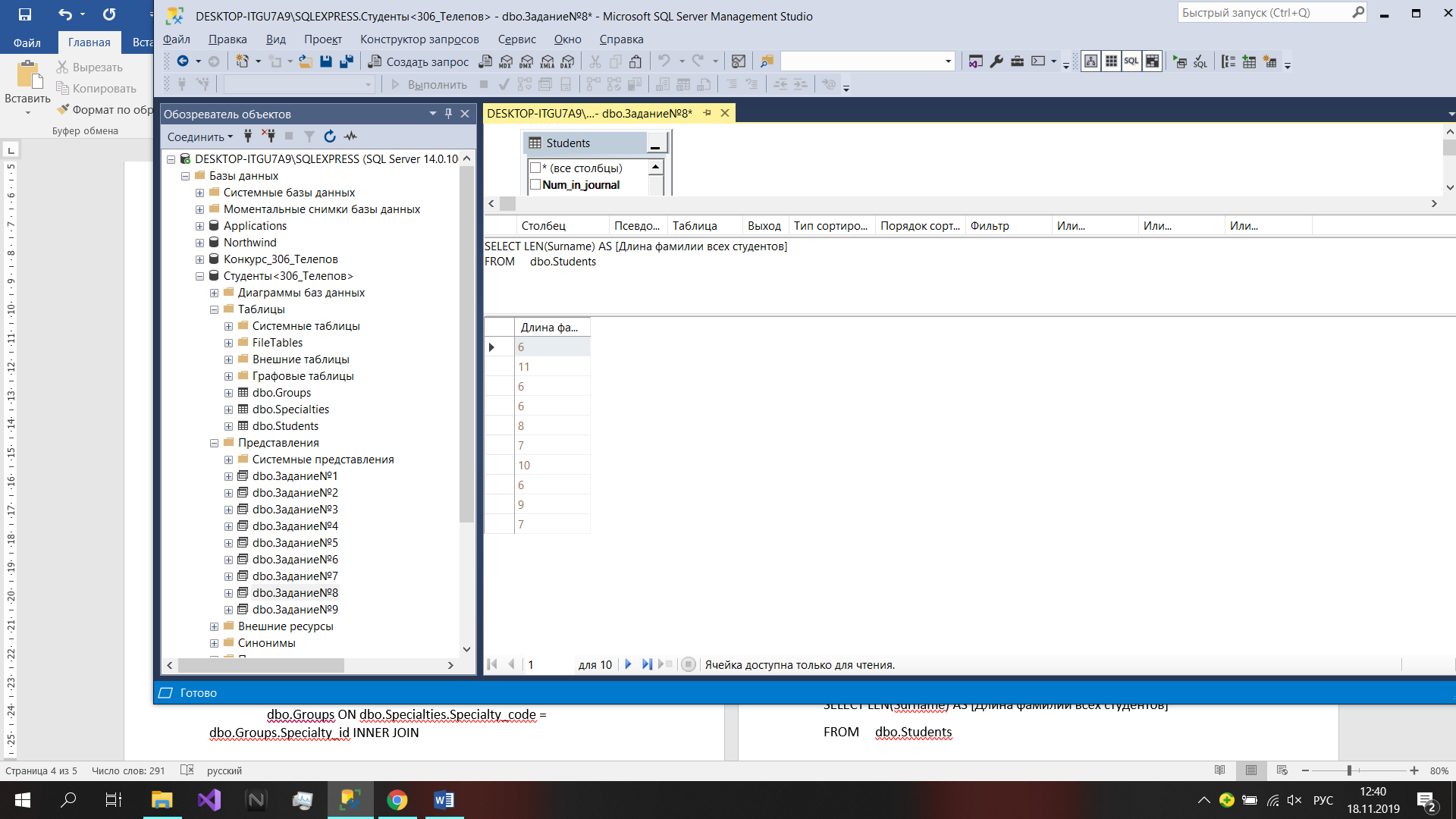


1. Собственное представление с использованием LEN (Вывести длину фамилии каждого студента)

SELECT LEN(Surname) AS [Длина фамилии всех студентов], Surname

FROM dbo.Students

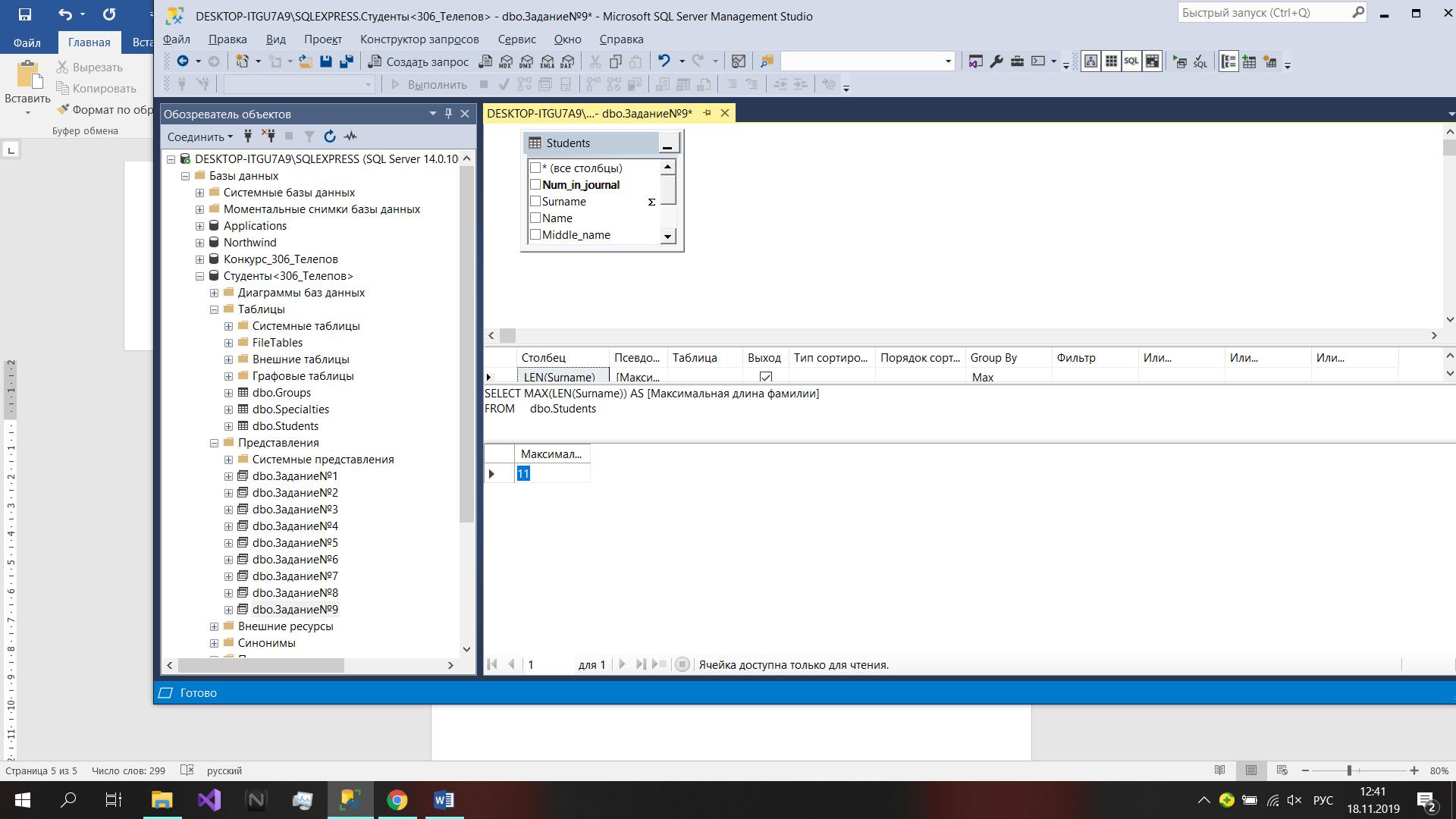
GROUP BY Surname



1. Собственное представление с использованием MAX (Вывести максимальную длину фамилии студента)

SELECT MAX(LEN(Surname)) AS [Максимальная длина фамилии]

FROM dbo.Students



**Ответы на контрольные вопросы**

1. Представление (VIEW) — объект базы данных, являющийся результатом выполнения запроса к базе данных, определённого с помощью оператора SELECT, в момент обращения к представлению.
2. Агрегатные, аналитические, ранжирующие, функции наборов строк, скалярные функции. Пример: MAX( LEN(фамилия) ) – MAX выведет максимальное значение, а LEN – длину строки, следовательно выведется максимальная длина строки.
3. Применяются к полям c GROUP BY
4. Таблицы, запросы, представления, формы, отчёты, макросы и модули. Работали с таблицами, запросами, представлениями.