|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ | | | | | | | | |
|  | | |  | | |  | | |
| ФГБОУ ВО «Пермский государственный национальный исследовательский университет» | | | | | | | | |
|  | | |  | | |  | | |
|  | | ОТЧЕТ  по лабораторной работе №1 «Создание БД» по дисциплине «Базы данных и СУБД» | | | | |  | |
|  | | |  | | |  | | |
|  | Работу выполнил  студент гр. ПМИ-6  Шукшина М.И.  «20» января 2018 | | |  | Проверил ассистент кафедры МОВС  Костарева Т.А.  «23»января 2018 | | |  |
|  |  | | |  |  | | |  |
| Пермь 2018 | | | | | | | | |

ПОСТАНОВКА ЗАДАЧИ

Создать и наполнить базу данных, состоящую их трёх взаимно вложенных таблиц. Пример: университеты - факультеты - студенты. Определить первичные и внешние ключи, установить ограничение уникальности, указать значения по умолчанию. Построить диаграмму.

Выполнить следующие запросы:

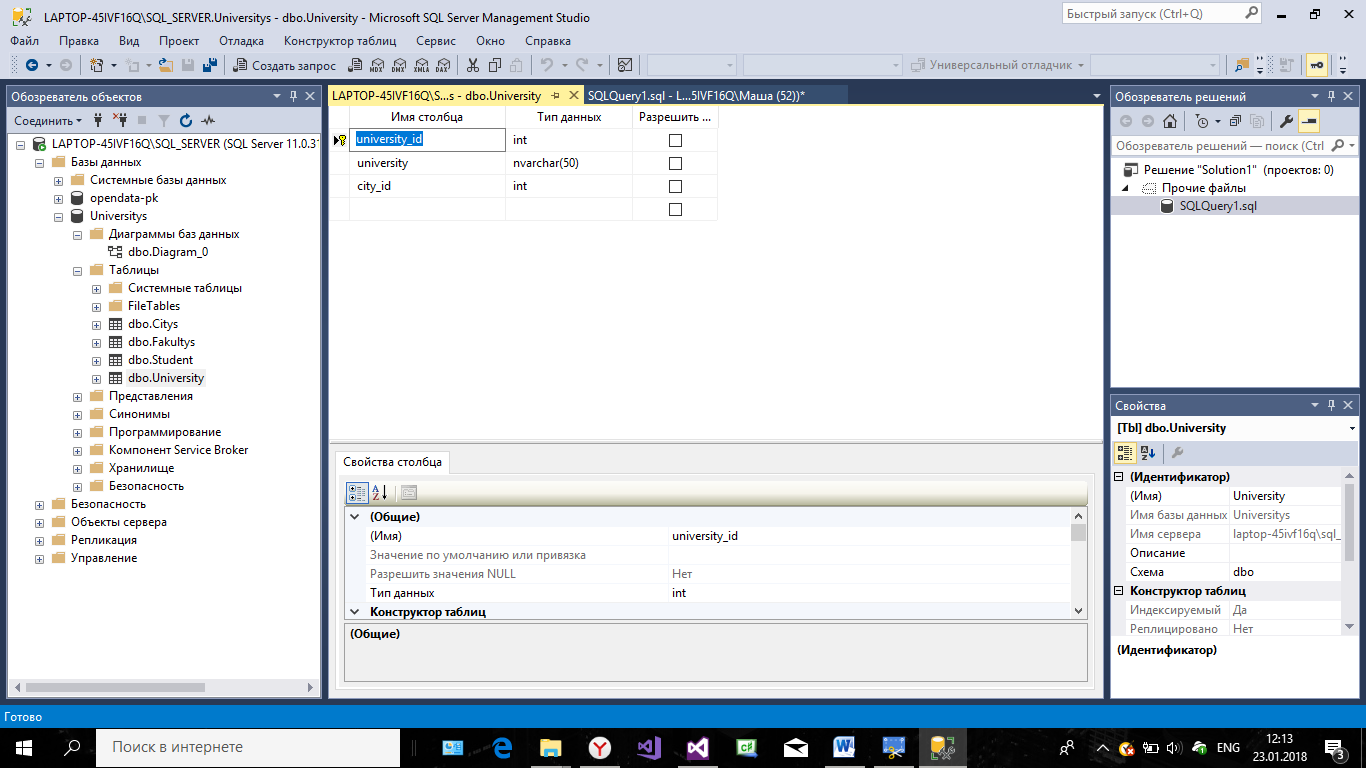
1. Insert-запрос для проверки значения по умолчанию;
2. Insert-запрос для проверки уникальности значения;
3. Insert-запрос для проверки внешнего ключа;
4. Select-запрос всех данных БД (where);
5. Select-запрос всех данных БД (join).

В качестве СУБД была выбрана MS SQL. Для администрирования – MS SQL Management Studio.

ОПИСАНИЕ СТРУКТУРЫ БД

База данных будет состоит из четырех таблиц:

1. *Uuniversity* – список университетов

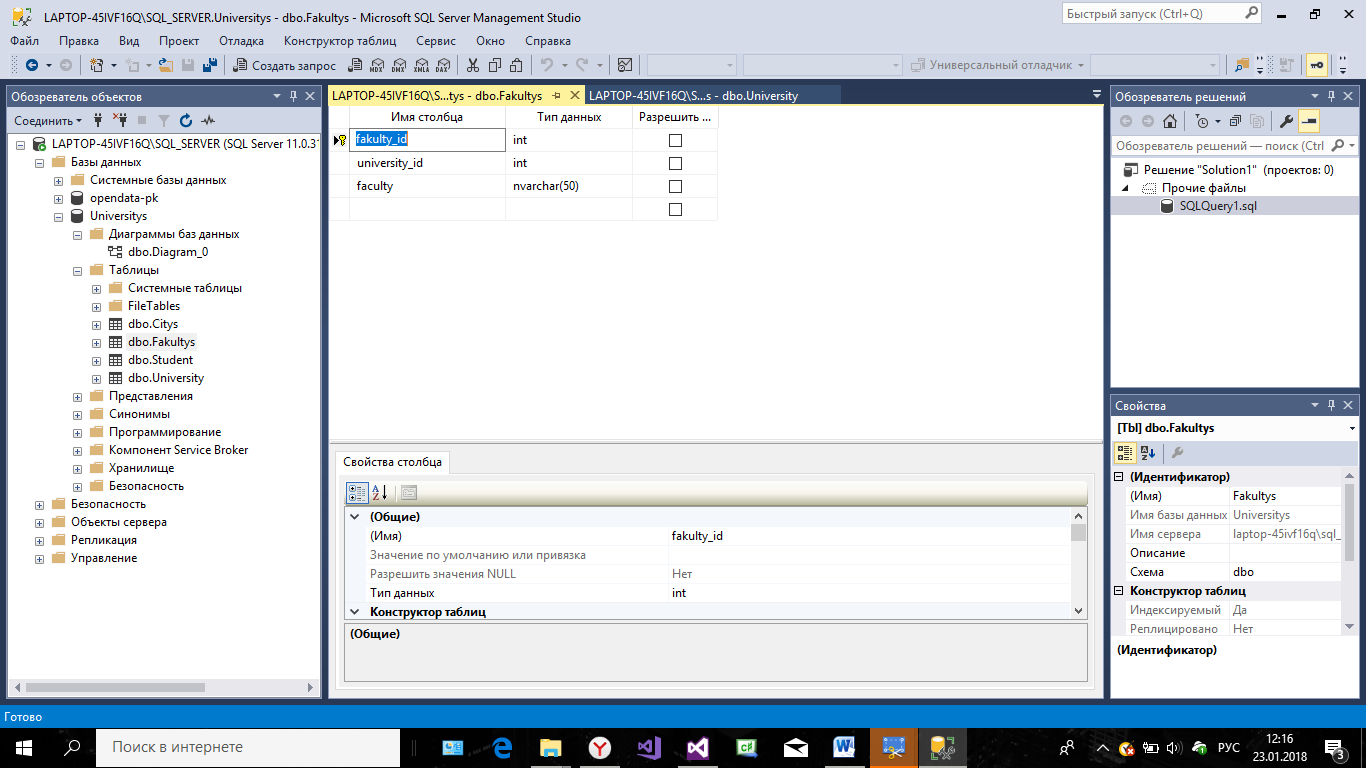


*university\_id* – первичный ключ, авто-инкремент. Используем данное поле для связки с таблицей *facultys*.

*university* – уникальный ключ, т. к. считаем, что в одном городе не может быть университетов с одинаковыми названиями.

*city\_id* – внешний ключ к таблице *Citys.*

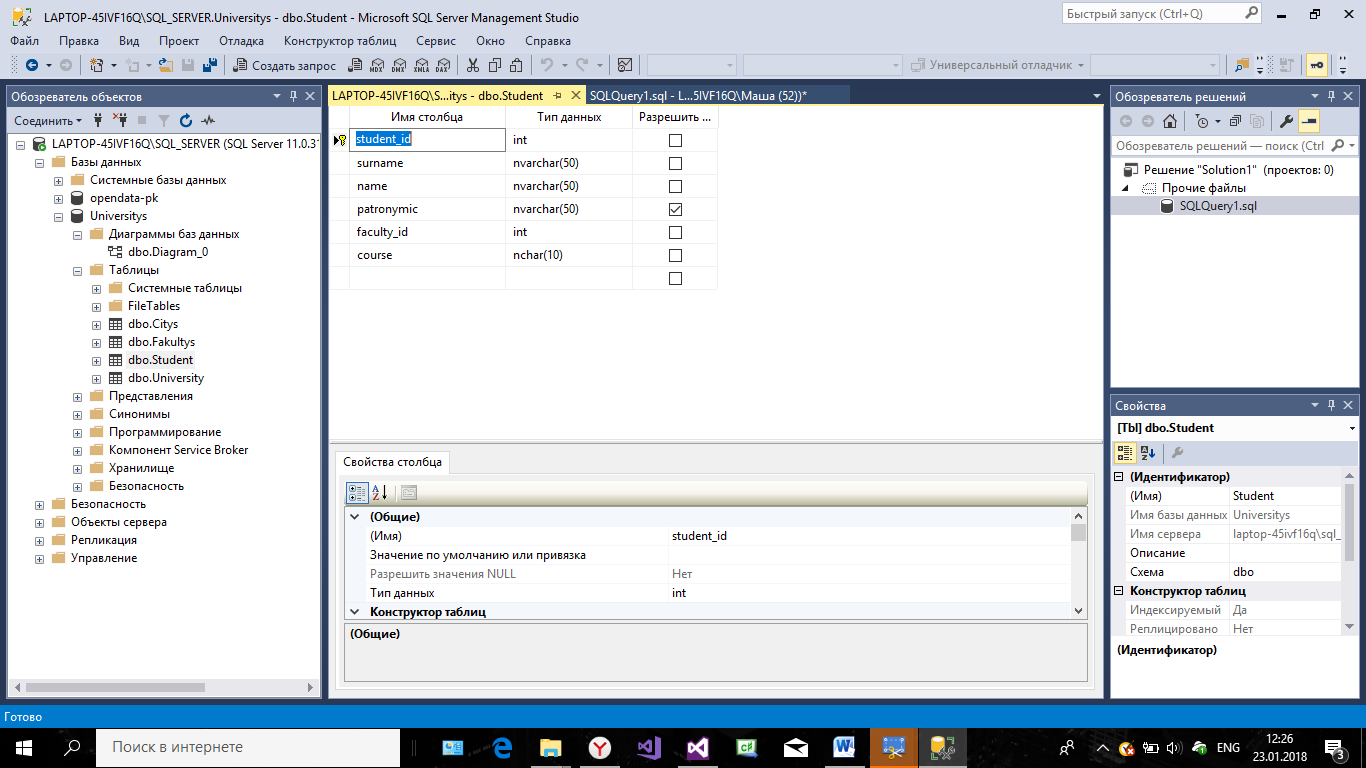
1. *Facultys* – список факультетов



*faculty\_id* – первичный ключ, авто-инкремент. Используем данное поле для связки с другой таблицей.

*university\_id* – внешний ключ к таблице *Universitiy.* Получаем отношение один ко многим, т.к. подразумеваем, что в одном университете может быть несколько факультетов.

1. *Student* – список студентов

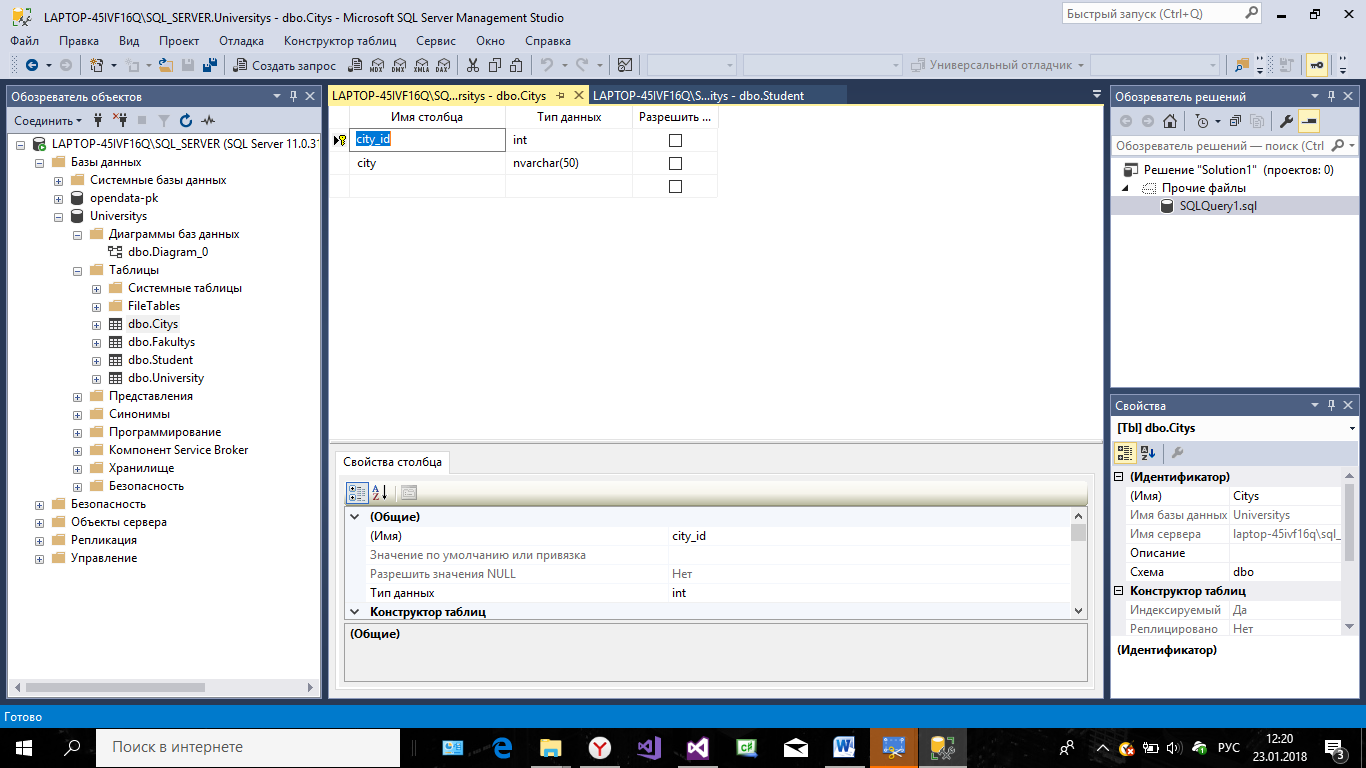


*student\_id* – первичный ключ, авто-инкремент

*faculty\_id* – внешний ключ к таблице *Facultiys.* Получаем отношение один ко многим, т.к. подразумеваем, что на одном факультете может обучаться несколько студентов.

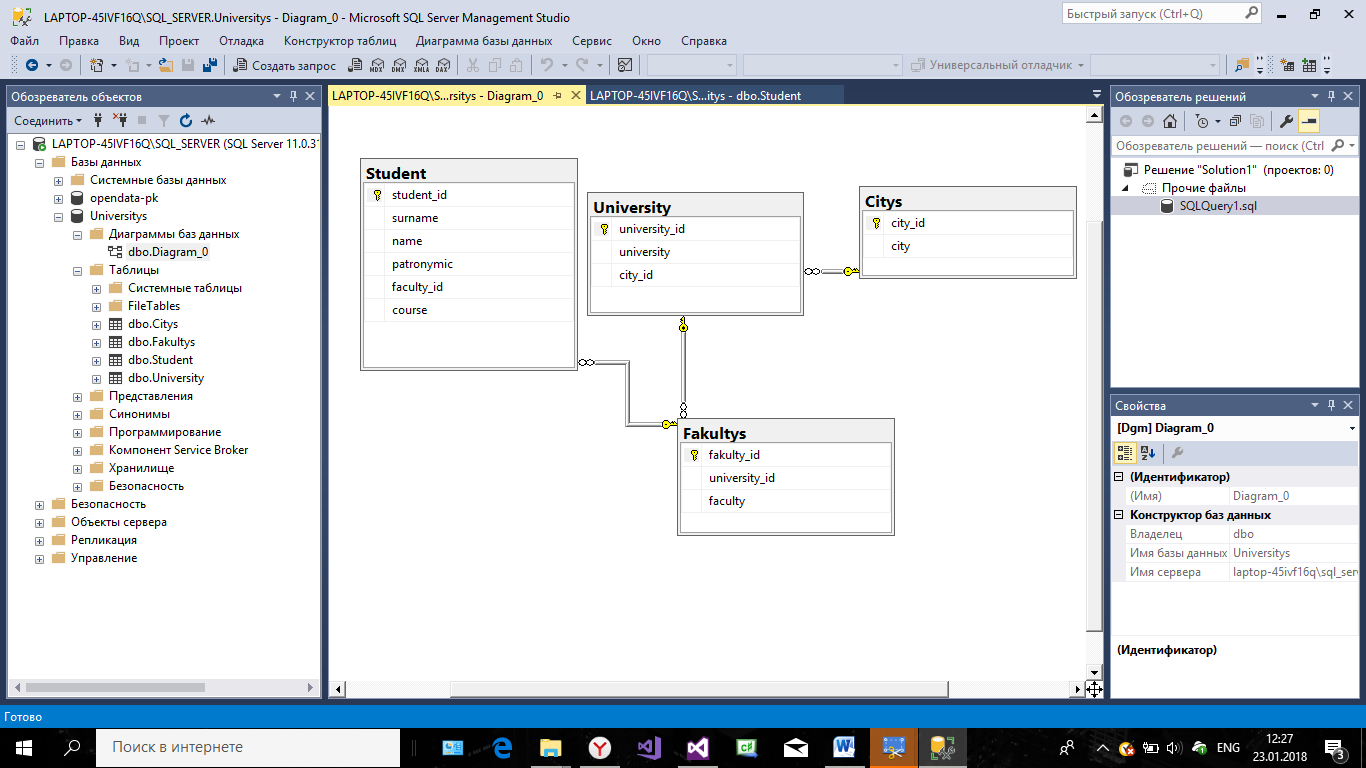
*course –* атозаполняемое поле (‘ 1 ‘)

1. *Citys* – список городов



*city\_id* – первичный ключ, авто-инкремент. Используем данное поле для связки с таблицей *Universitiy.* Получаем отношение один ко многим, т.к. подразумеваем, что в одном городе может быть несколько университетов.

ДИАГРАММА БД



ЗАПРОСЫ

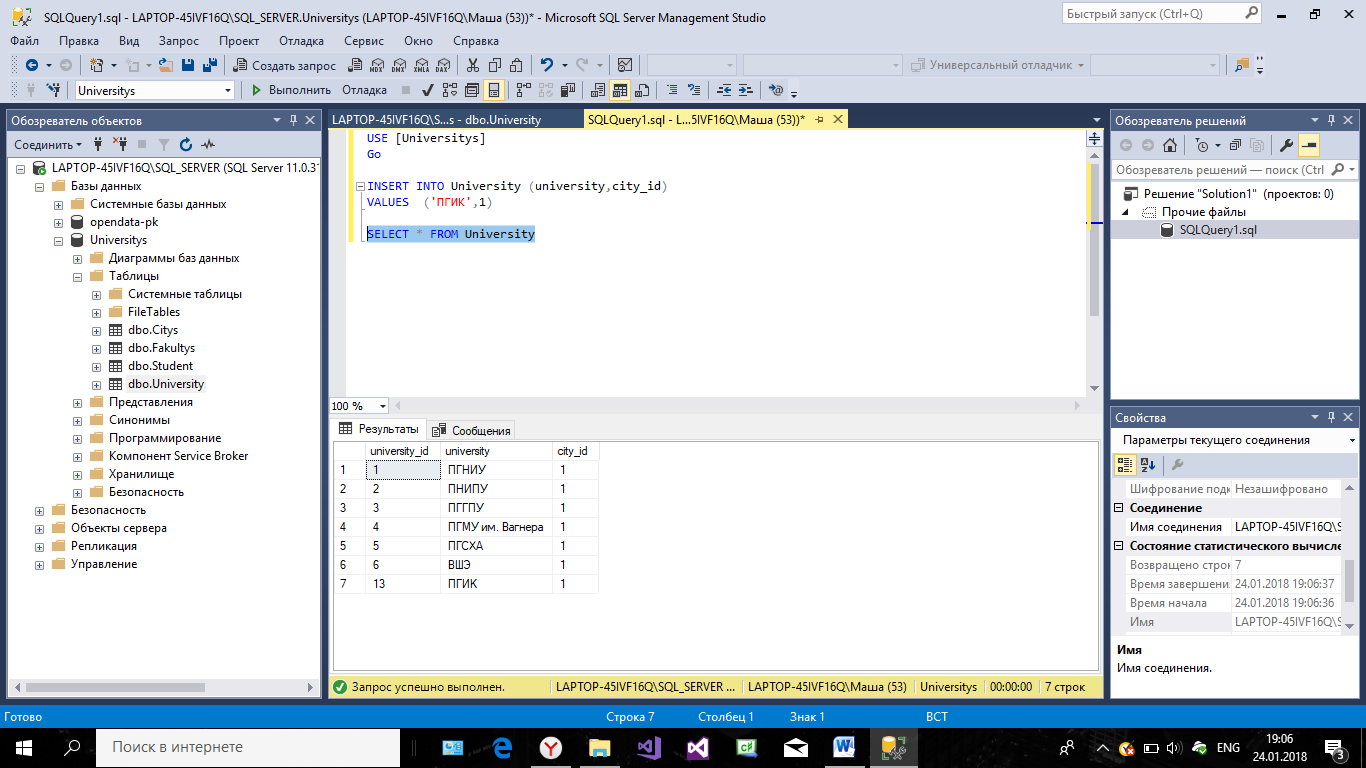
1. Insert-запрос для проверки значения по умолчанию

Попробуем добавить университет, поскольку значение по умолчанию поля – *City* – ‘Пермь”, то нам необязательно указывать значение поля *City*.

INSERT INTO University (university,city\_id)

VALUES ('ПГИК',1)

Результат:



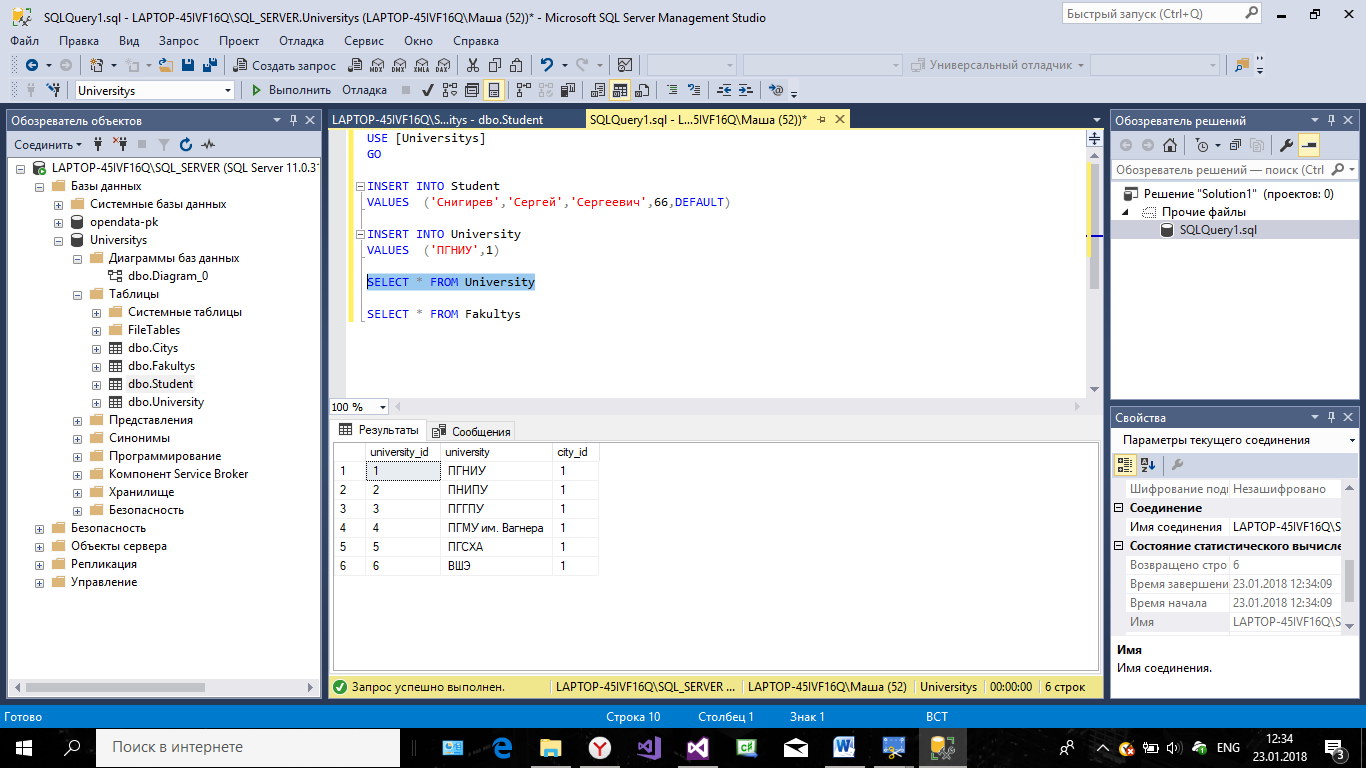
1. Insert-запрос для проверки уникальности значения

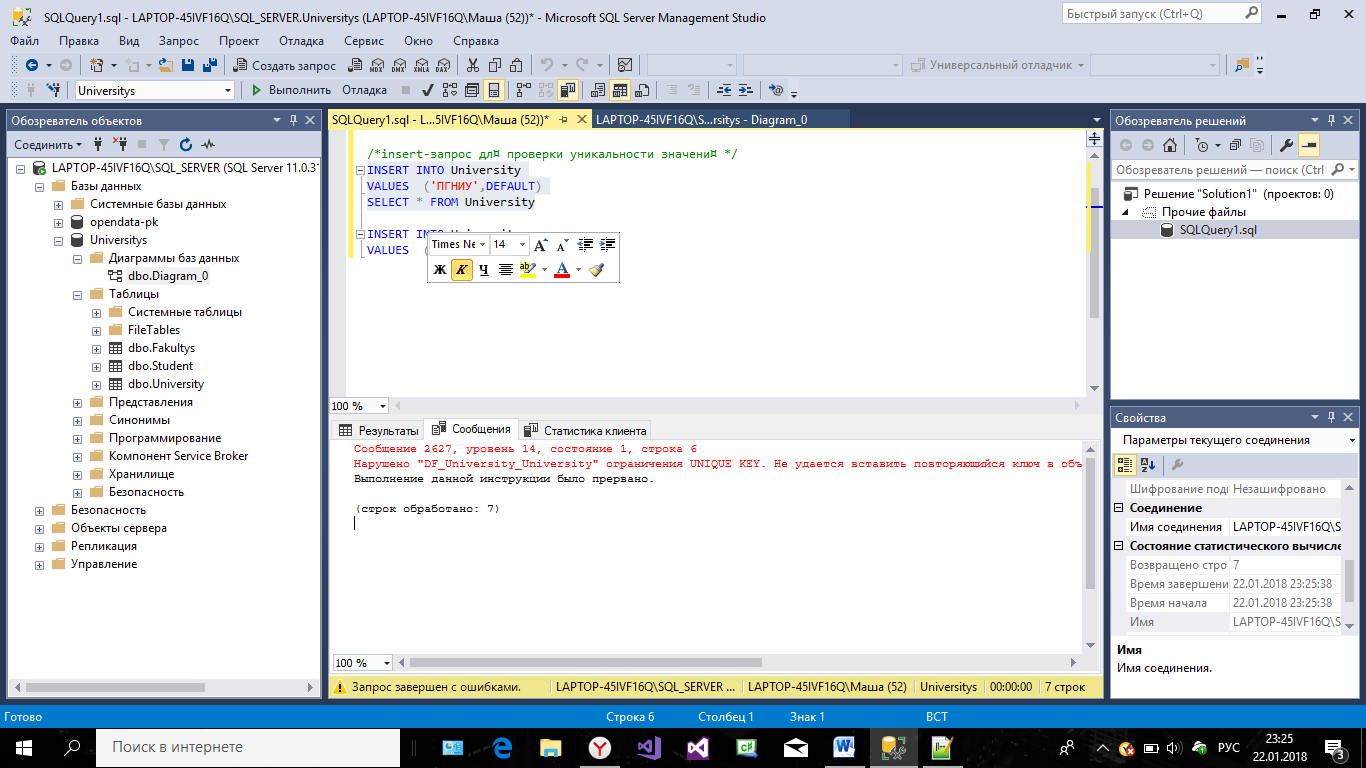
Попробуем добавить университет с названием «ПГНИУ», который уже есть.

INSERT INTO University

VALUES ('ПГНИУ',1)

Результат:





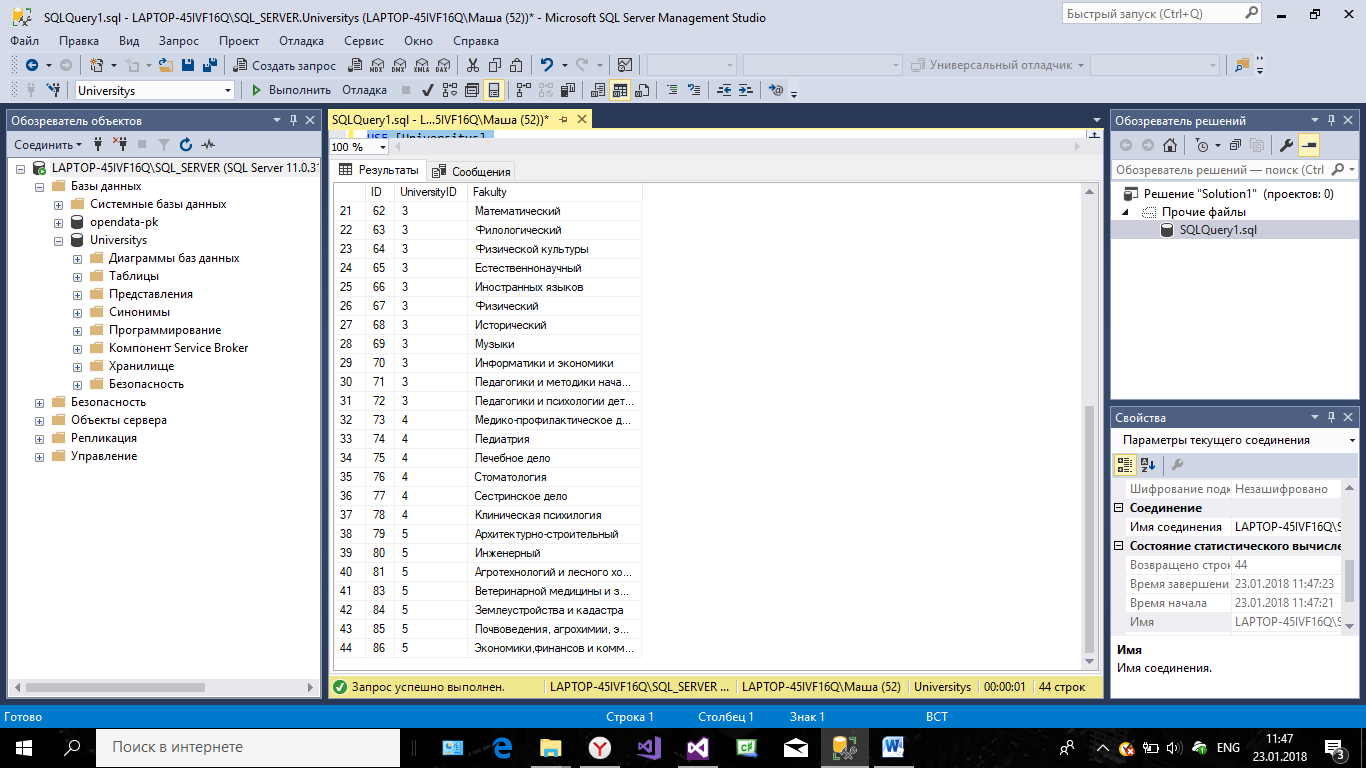
1. Insert-запрос для проверки внешнего ключа

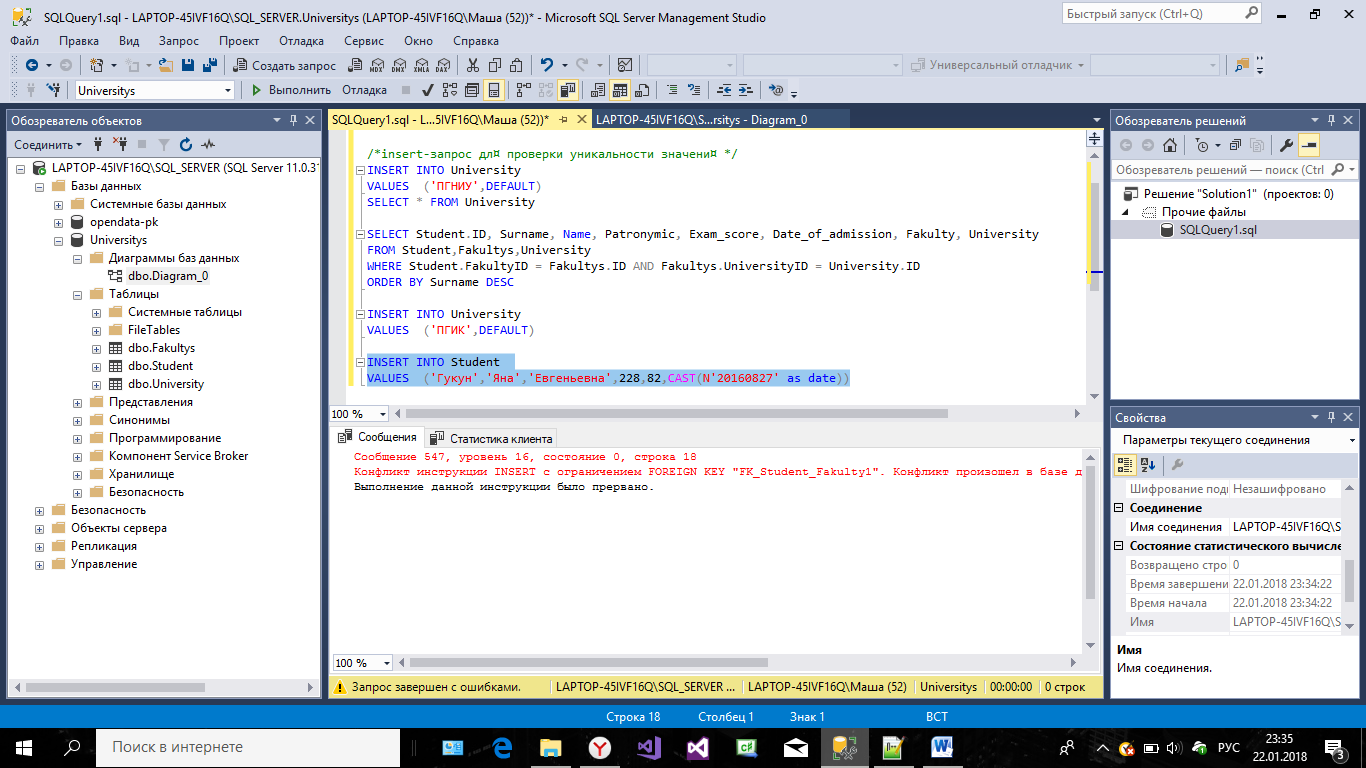
Попробуем добавить студента, который учится на факультете, которого нет в нашей базе данных

INSERT INTO Student (surname, [name], fakulty\_id)

VALUES ('Гукун','Яна',82)

Результат:





1. Select-запрос всех данных БД (where)

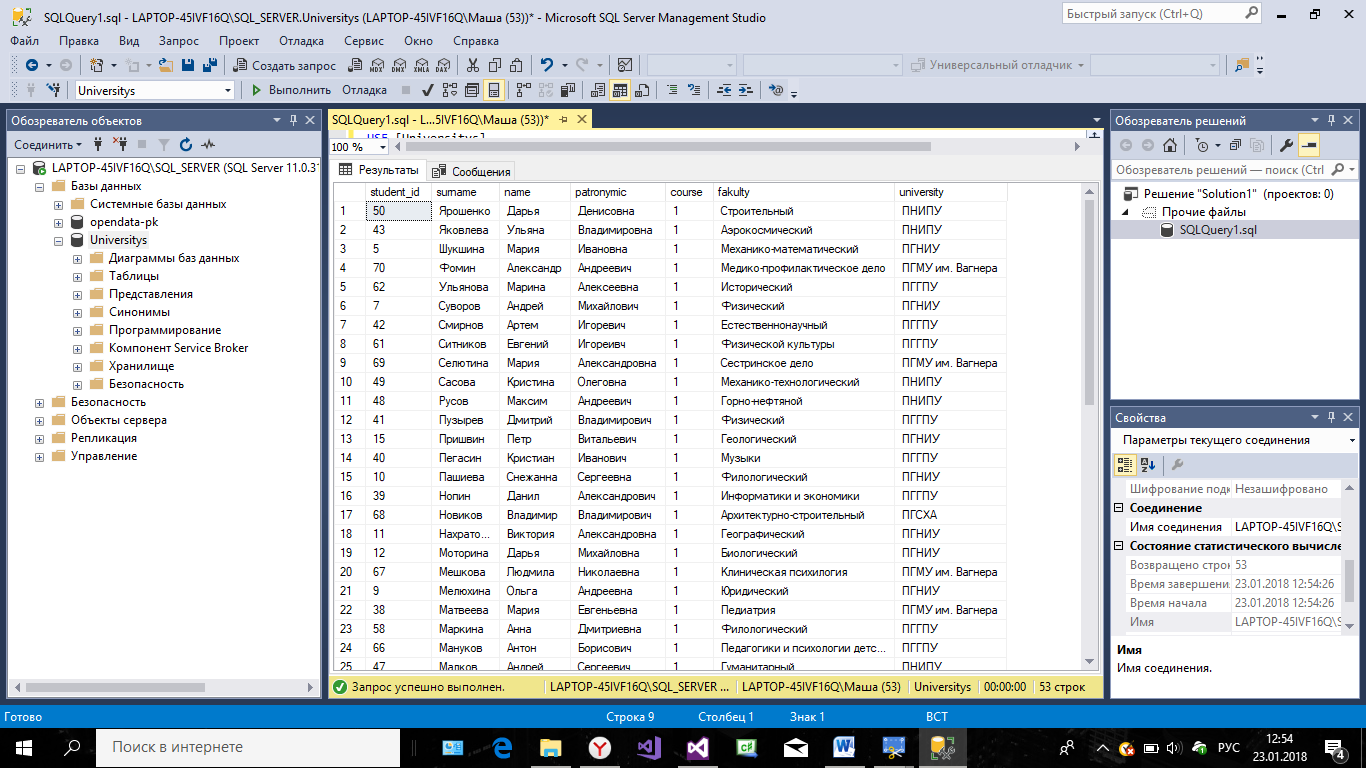
SELECT Student.student\_id, surname, name, patronymic, course, fakulty, university

FROM Student, Fakultys, University

WHERE Student.fakulty\_id = Fakultys.fakulty\_id AND Fakultys.university\_id = University.university\_id

ORDER BY surname DESC

Результат:



1. Select-запрос всех данных БД (join)

SELECT Student.student\_id, surname, name, patronymic, course, fakulty, university

FROM (Student FULL OUTER JOIN Fakultys

ON Fakultys.fakulty\_id = Student.fakulty\_id)

FULL OUTER JOIN University

ON University. university\_id = Fakultys.university\_id

Результат:

