Белорусский государственный университет информатики и радиоэлектроники

Факультет компьютерных технологий

ОТЧЕТ

по лабораторной работе №4

на тему: **Создание базы данных и ее объектов с помощью команд языка Transact-SQL**

|  |  |
| --- | --- |
| Выполнил  студент  группы №980161  Алейчик И.Д. | Проверил  доцент  Бич Н.А. |

Минск 2020

**Лаборатоорная работа №4**

**Вариант 1**

**Тема:**

* Создание базы данных и ее объектов с помощью команд языка Transact-SQL

**Цель работы:**

* Ознакомиться с процессом создания базы данных и ее объектов
* Освоить способы управления пользователями и правами доступа к данным.

**Ход выполнения работы:**

1. **Создание базы данных и ее объектов**

Создается база данных с стандртными натсройками.

create database Whearehouse

use Whearehouse

Создаем обьекты базы данных с связями и необходимыми значениями по умолчанию.

create table Regions(

Key\_Region int identity, -- key region

Country nvarchar(20) default 'Беларусь' not null ,

Area nvarchar(20) not null,

City nvarchar(20) not null,

Adress nvarchar(50) not null,

Phone char(15),

Fax char(15) not null,

constraint PK\_Regions primary key(Key\_Region),

)

create table Providers(

Key\_Provider int identity,

Name\_Provider nvarchar(40) not null,

Payment\_Term varchar(30) default 'Предоплата',

Key\_Region int,

Note nvarchar(MAX)

constraint PK\_Providers primary key(Key\_Provider),

constraint FK\_Providers\_Regions foreign key (Key\_Region) references Regions(Key\_Region),

)

create table Clients(

Key\_Client int identity,

Name\_Client nvarchar(40) not null,

Name\_Director nvarchar(60),

Surname\_Director nvarchar(60),

Patronymic\_Director nvarchar(60),

Key\_Region int,

constraint PK\_Clients primary key(Key\_Client),

constraint FK\_Clients\_Region foreign key (Key\_Region) references Regions(Key\_Region),

)

create table Currencies(

Key\_Currency int identity,

Name\_Currency varchar(30) not null,

Step\_Round numeric(10,4) default 0.01,

Exchange\_Rate smallmoney not null

constraint PK\_Currencies primary key(Key\_Currency),

)

create table Products(

Key\_Product int identity,

Name\_Product nvarchar(50) not null,

Unit\_Product char(10) default 'штука',

Price money,

Key\_Currency int default 'BYR',

Packaged char(3) default 'Нет' not null

constraint PK\_Products primary key(Key\_Product),

constraint FK\_Products\_Currencies foreign key(Key\_Currency) references Currencies(Key\_Currency),

)

create table Orders(

Key\_Order int identity not null,

Key\_Client int not null,

Key\_Product int not null,

Count\_Order\_Product numeric(12,3),

Date\_Order datetime default getdate(),

Delivety\_Time datetime default getdate()+14,

Key\_Provider int,

constraint PK\_Orders primary key(Key\_Order),

constraint FK\_Orders\_Products foreign key(Key\_Product) references Products(Key\_Product),

constraint FK\_Orders\_Clients foreign key(Key\_Client) references Clients(Key\_Client),

constraint FK\_Orders\_Providers foreign key(Key\_Provider) references Providers(Key\_Provider),

)

Go

--Указывает индексы таблиц

CREATE UNIQUE INDEX UIX\_Provider ON Providers (Name\_Provider)

CREATE UNIQUE INDEX UIX\_Client ON Clients (Name\_Client)

CREATE UNIQUE INDEX UIX\_Currency ON Currencies (Name\_Currency)

CREATE UNIQUE INDEX UIX\_Product ON Products (Name\_Product)

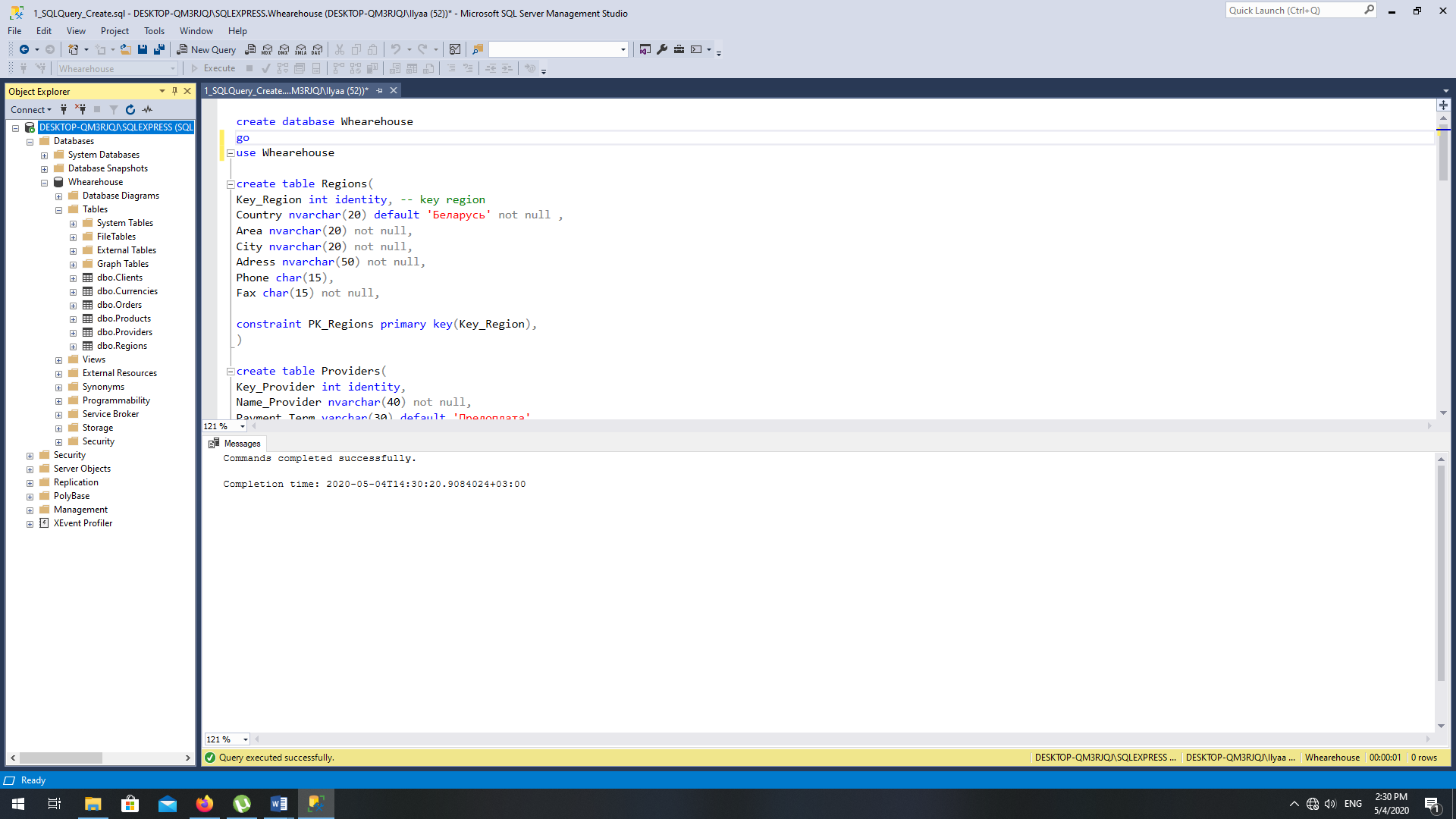
CREATE INDEX IX\_Region ON Regions (Country, City)

CREATE INDEX IX\_Product ON Products (Unit\_Product, Name\_Product)

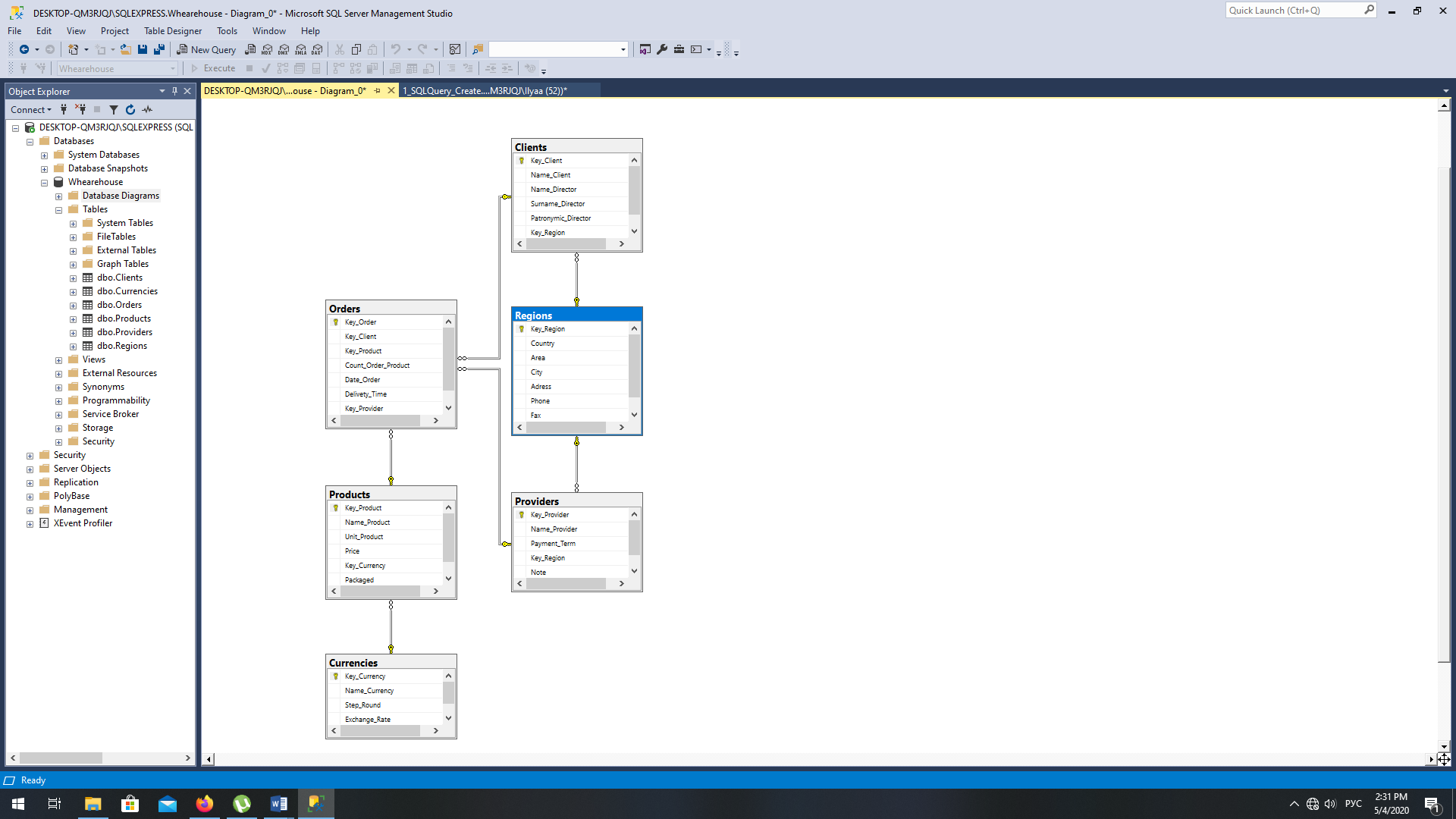
CREATE INDEX IX\_Order ON Orders (Date\_Order)

go

Результат выполнения:



Полученная диаграмма базы данных:



1. **Ввод данных в таблицы базы данных**

INSERT INTO Regions VALUES ('Россия', 'Московская', 'Королев', 'ул.Мира, 15','387-23-04', '387-23-05')

INSERT INTO Regions (Area, City, Adress, Fax) VALUES ('', 'Минск', 'ул.Гикало, 9', '278-83-88')

INSERT INTO Regions (Area, City, Adress, Fax) VALUES ('Минская', 'Воложин', 'ул.Серова, 11', '48-37-92')

INSERT INTO Regions (Area, City, Adress, Phone, Fax) VALUES ('', 'Минск', 'ул.Кирова, 24', '269-13-76','269-13-77')

INSERT INTO Regions (Area, City, Adress, Fax) VALUES ('Витебская', 'Полоцк', 'ул.Лесная, 6', '48-24-12')

INSERT INTO Regions VALUES ('Украина', 'Крымская', 'Алушта', 'ул.Франко, 24', NULL, '46-49-16')

GO

INSERT INTO Currencies(Name\_Currency, Exchange\_Rate) VALUES ( 'BYR', '1')

INSERT INTO Currencies (Name\_Currency, Exchange\_Rate) VALUES ('USD','2.45')

INSERT INTO Currencies (Name\_Currency, Step\_Round, Exchange\_Rate) VALUES ('EUR', '0.45','2.30')

GO

INSERT INTO Products(Name\_Product, Unit\_Product, Price,Packaged) VALUES ( 'Epson Stylus Photo R270', '5','200','да')

INSERT INTO Products(Name\_Product, Unit\_Product, Price) VALUES ( 'SSD Kingston UV400', '2','345')

INSERT INTO Products(Name\_Product, Unit\_Product, Price, Key\_Currency,Packaged) VALUES ( 'Dell MT40T3', '1','120','2','нет')

GO

INSERT INTO Clients(Name\_Client, Name\_Director, Surname\_Director,Patronymic\_Director,Key\_Region) VALUES ('ОАО Амарилис', 'Андрей','Ватмов','Олегович','1')

GO

INSERT INTO Providers(Name\_Provider, Key\_Region, Note) VALUES ('ОАО ЕмплойментТек','6','заметок нет')

GO

INSERT INTO Orders(Key\_Client, Key\_Product, Count\_Order\_Product,Key\_Provider) VALUES (1,1,'3',1)

GO

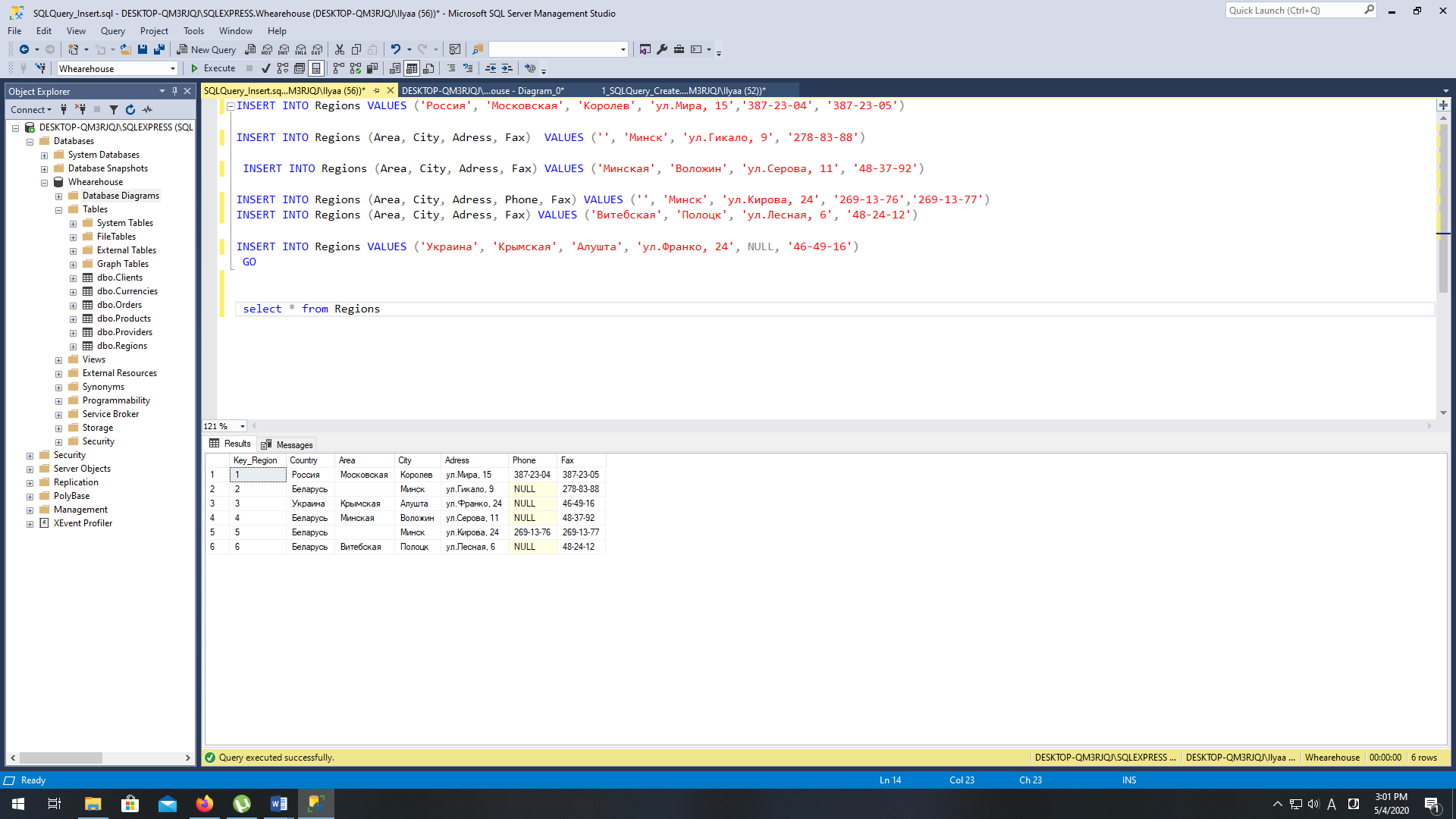
select \* from Regions

select \* from Currencies

select \* from Products

select \* from Orders

Результат выполнения



Создание представления

CREATE VIEW ProductOrderView AS

SELECT Products.Name\_Product, Orders.Count\_Order\_Product,

Products.Unit\_Product, Providers.Name\_Provider

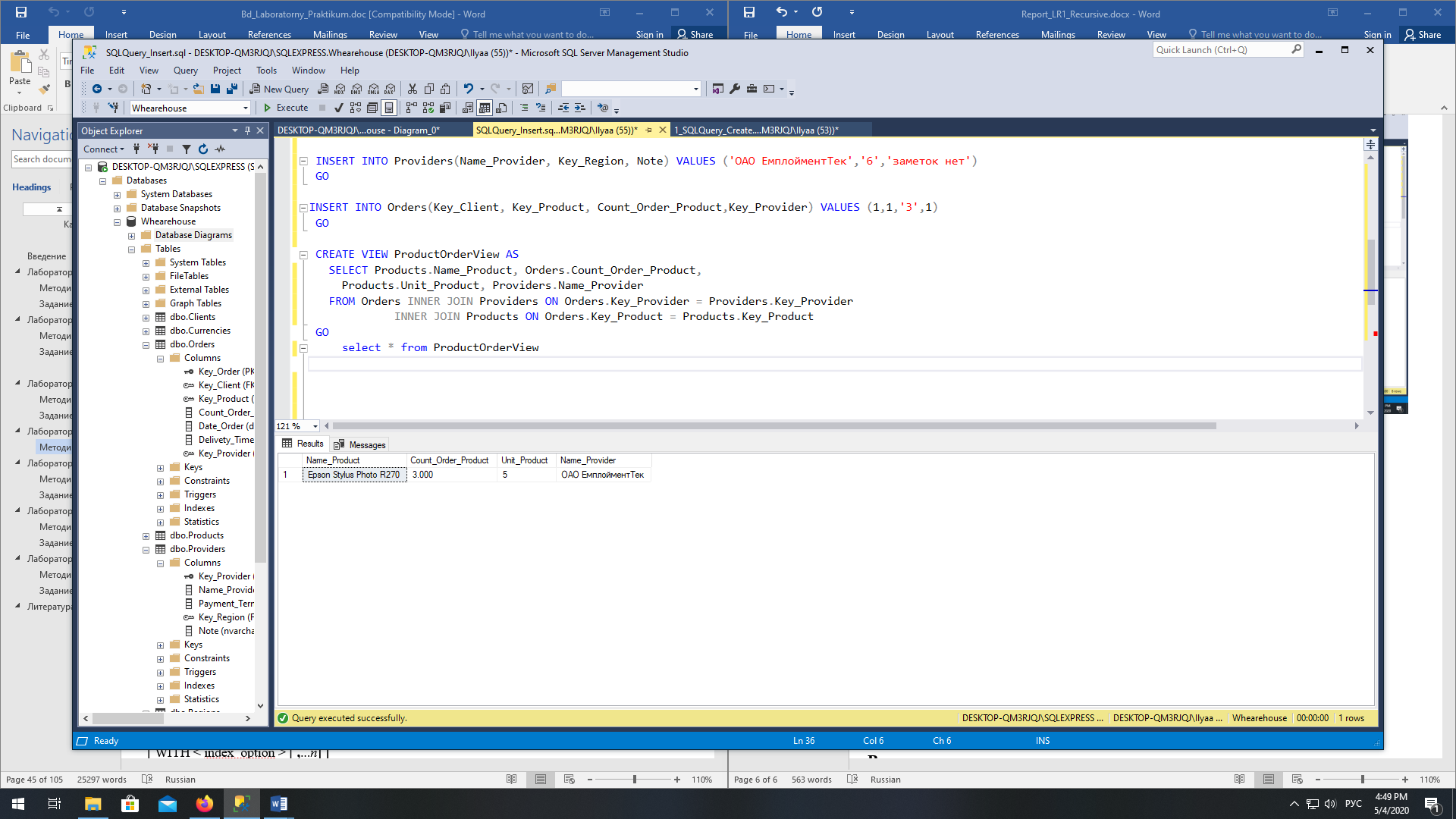
FROM Orders INNER JOIN Providers ON Orders.Key\_Provider = Providers.Key\_Provider

INNER JOIN Products ON Orders.Key\_Product = Products.Key\_Product

GO

select \* from ProductOrderView

Результат выполнения



1. **Управление пользователями и правами доступа к данным**.

EXEC sp\_grantlogin 'DESKTOP-QM3RJQJ\Ilyaa'

GO

-- Отмена доступа к серверу учетной записи

EXEC sp\_droplogin 'DESKTOP-QM3RJQJ\Ilyaa'

GO

-- Добавление учетной записи в фиксированную роль сервера

EXEC sp\_addsrvrolemember 'DESKTOP-QM3RJQJ\Ilyaa', 'dbcreator'

GO

--Удаление учетной записи из фиксированной роли сервера выполняется с помощью храним

EXEC sp\_dropsrvrolemember 'DESKTOP-QM3RJQJ\Ilyaa', 'dbcreator'

GO

--Создание нового пользователя и связывание его с учетной записью

EXEC sp\_grantdbaccess 'DESKTOP-QM3RJQJ\Ilyaa', 'Ilyaa'

GO

--Удаление пользователя

EXEC sp\_revokedbaccess 'DESKTOP-QM3RJQJ\Ilyaa'

GO

--Создание пользовательской роли

EXEC sp\_addrole 'Гл.бухгалтер', 'Ilyaa'

GO

--Удаление пользовательской роли

EXEC sp\_droprole 'Гл.бухгалтер'

GO

-- Добавление нового члена в роль (как фиксированную, так и пользовательскую) базы данныхю

EXEC sp\_addrolemember 'db\_accessadmin', 'Ilyaa'

EXEC sp\_addrolemember 'Гл.бухгалтер', 'Ilyaa'

GO

--Исключение члена из роли выполняется

EXEC sp\_droprolemember 'db\_accessadmin', 'Ilyaa'

EXEC sp\_droprolemember 'Гл.бухгалтер', 'Ilyaa'

GO

--Предоставление привилегий доступа к объектам базы данны

GRANT SELECT, INSERT, UPDATE, DELETE

ON Currencies TO [Гл.бухгалтер] WITH GRANT OPTION

GRANT UPDATE

ON Orders TO [Гл.бухгалтер] WITH GRANT OPTION

GRANT UPDATE, DELETE

ON Clients TO [Гл.бухгалтер] WITH GRANT OPTION

GRANT UPDATE, DELETE

ON Providers TO [Гл.бухгалтер] WITH GRANT OPTION

GRANT UPDATE, DELETE

ON Products TO [Гл.бухгалтер] WITH GRANT OPTION

GO

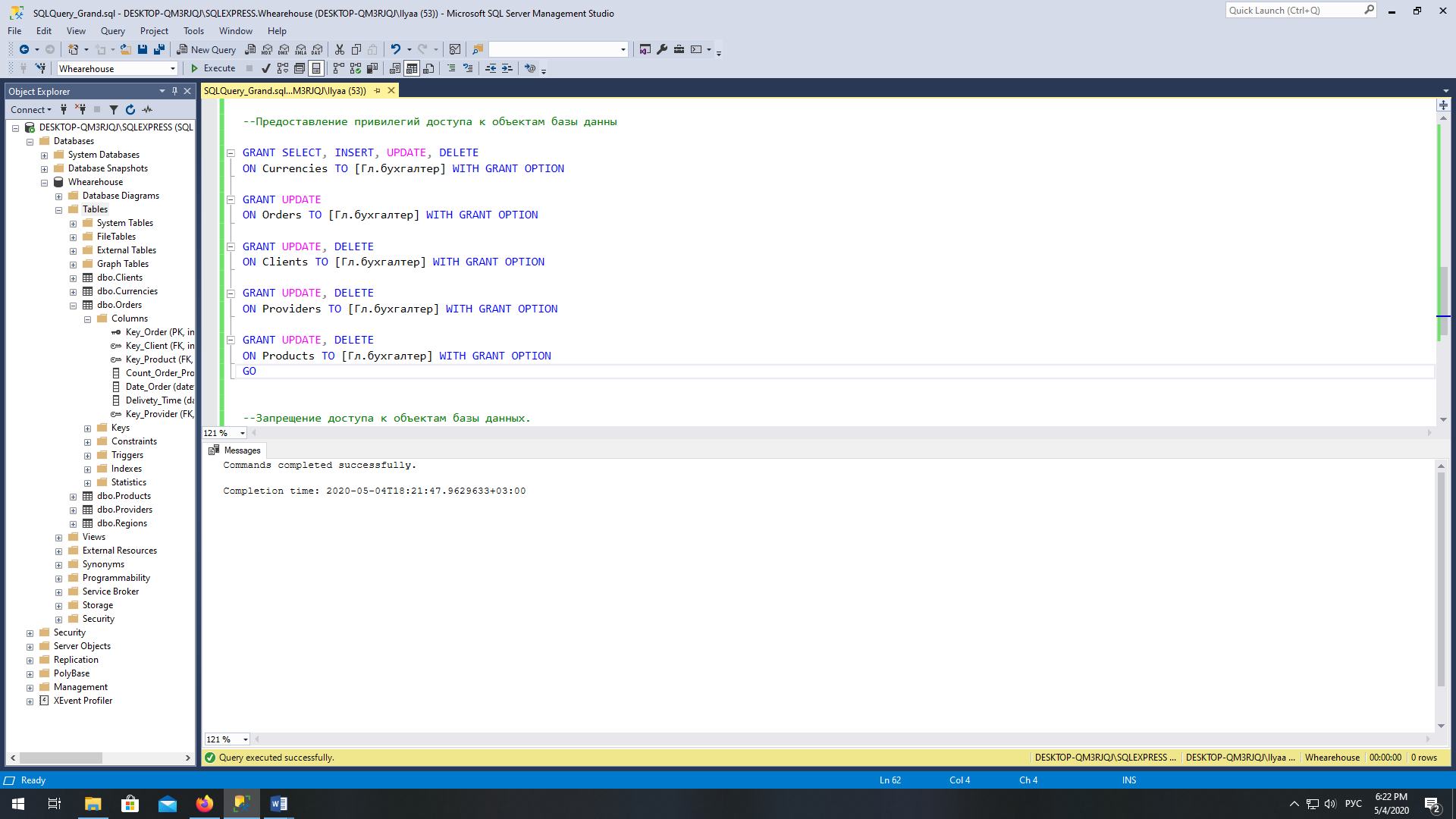
--Запрещение доступа к объектам базы данных.

DENY UPDATE

ON Orders (Date\_Order, Delivery\_Time) TO [Гл.бухгалтер] CASCADE

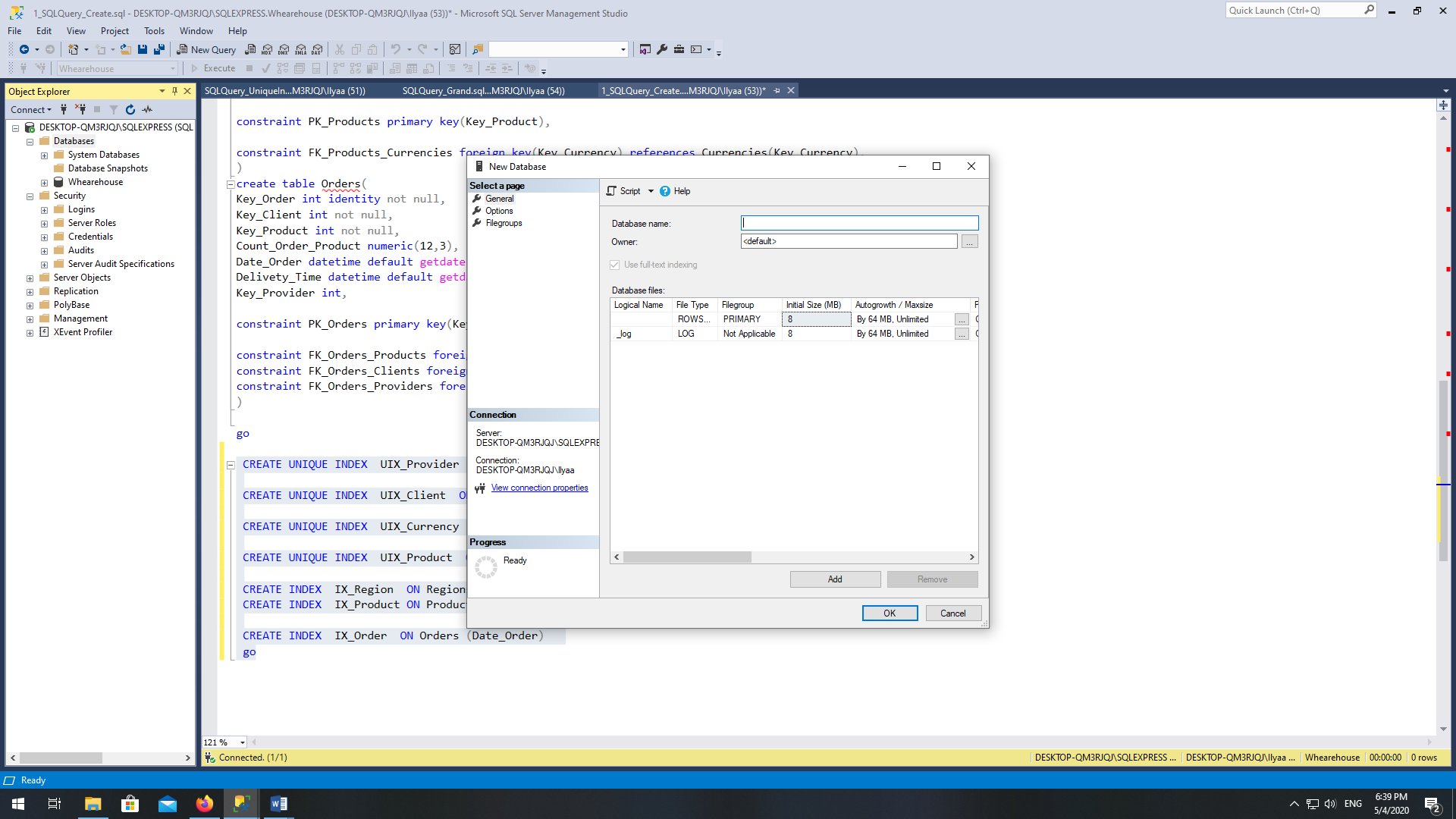
GO

Результат выполнения:

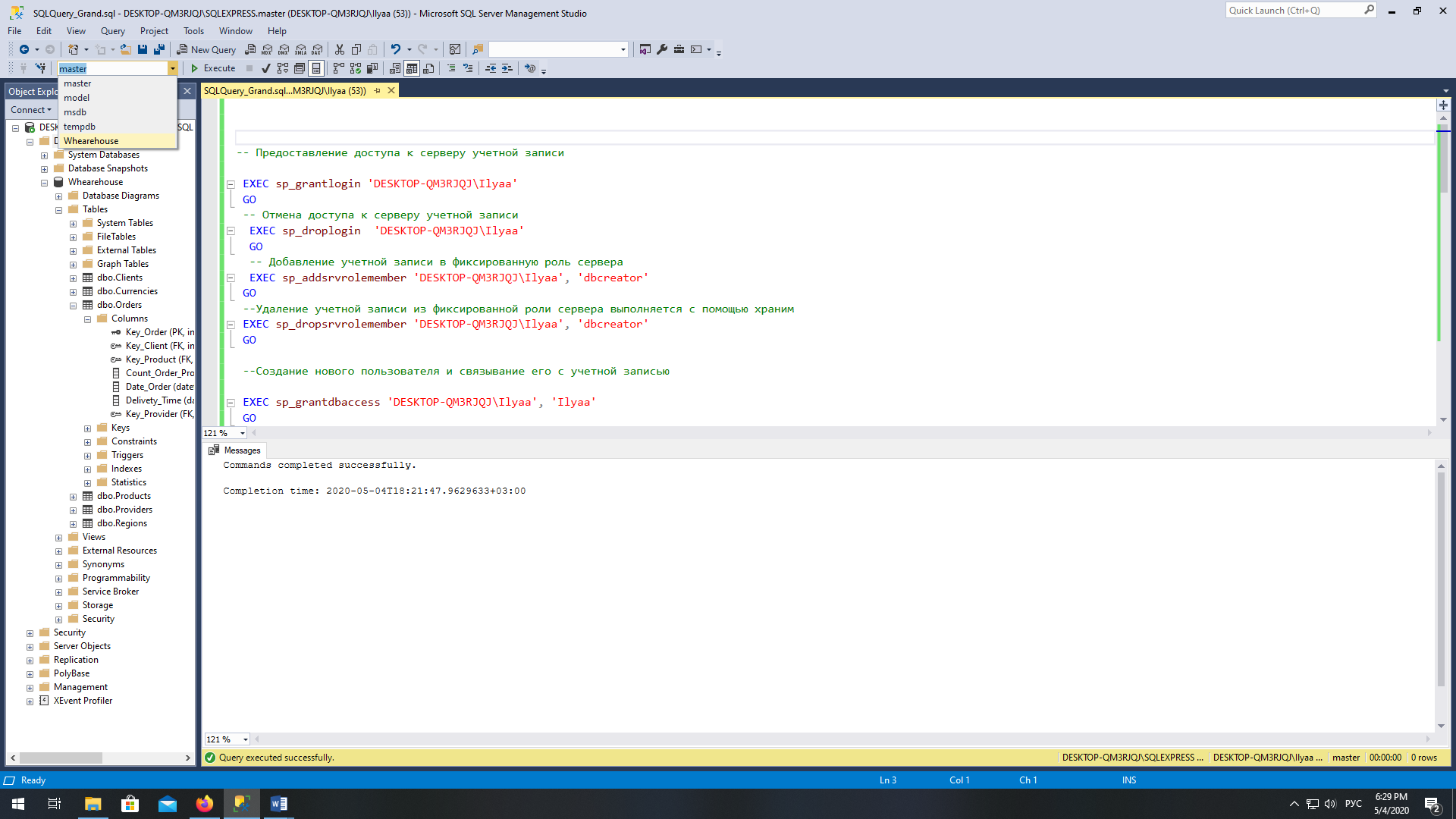


**4. Анализ полученных результатов**

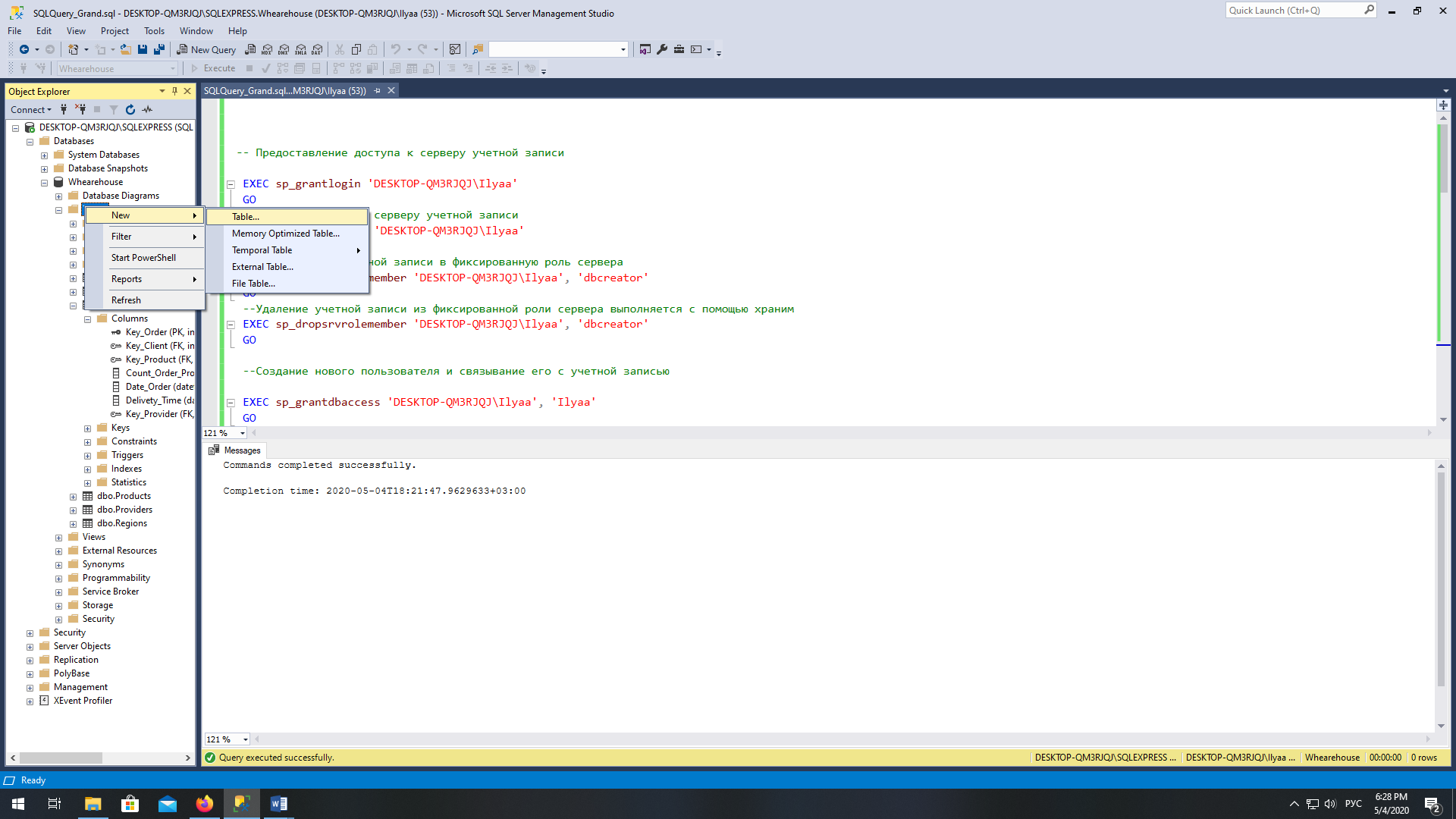
create database @name



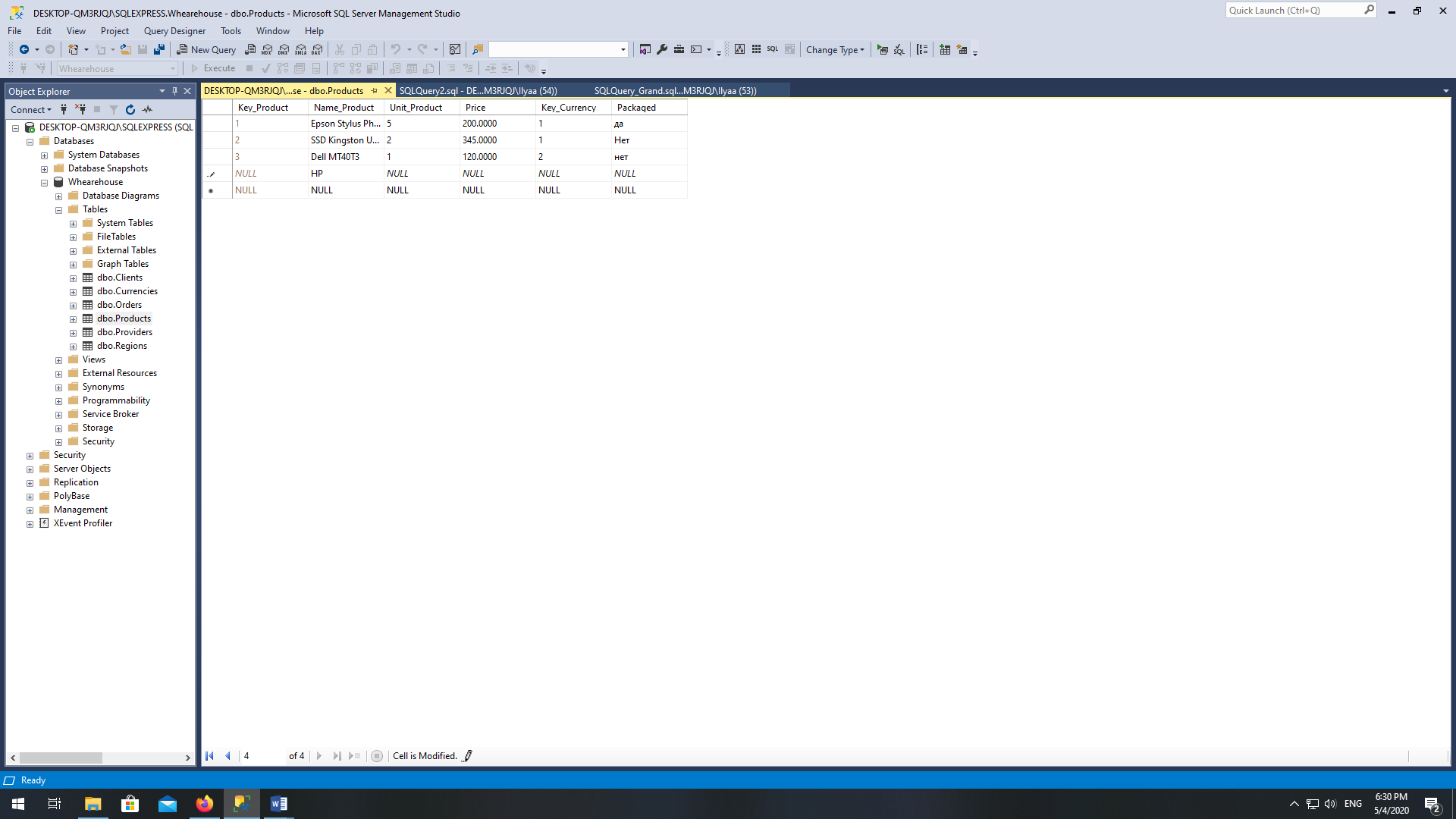
use @name



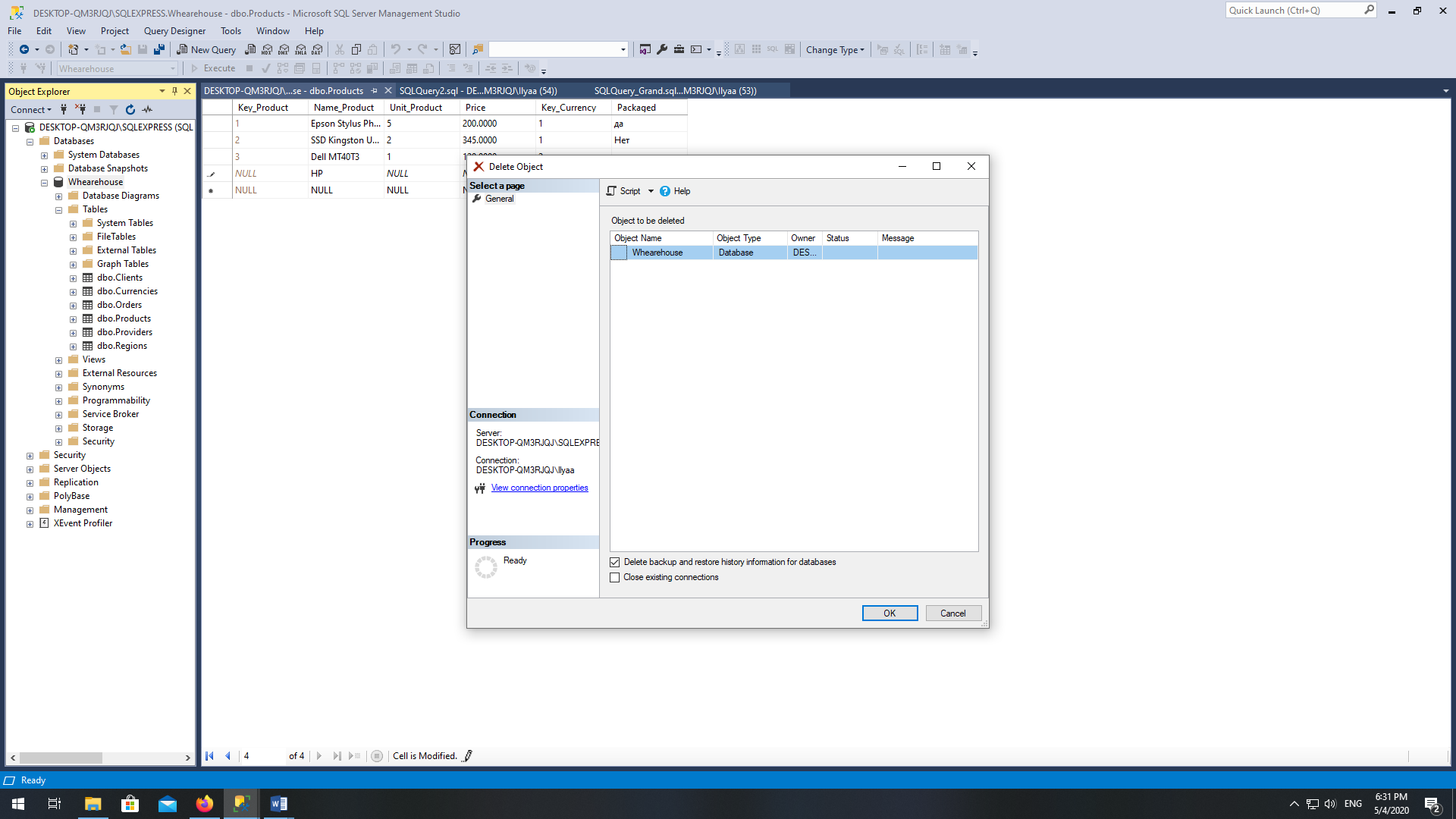
create table @name (id int …)



Insert into (@name) value (‘@n’)



drop database @name



**Вывод:** Ознакомился с процессом создания базы данных, объектов и у управлением правами доступа пользователей для доступа к данным. В процессе с помощью комманд T-SQL, создана база данных индентичная базе из 3 лабораторной работы где использовался графический интерфейс.