

Отчет по практической работе.
Практическая работа №10. «Хранимые процедуры»

Выполнила: Яковлева Евгения

Группа: КИСП-23(1)

Дата: 13.02.2025

1. Аннотация

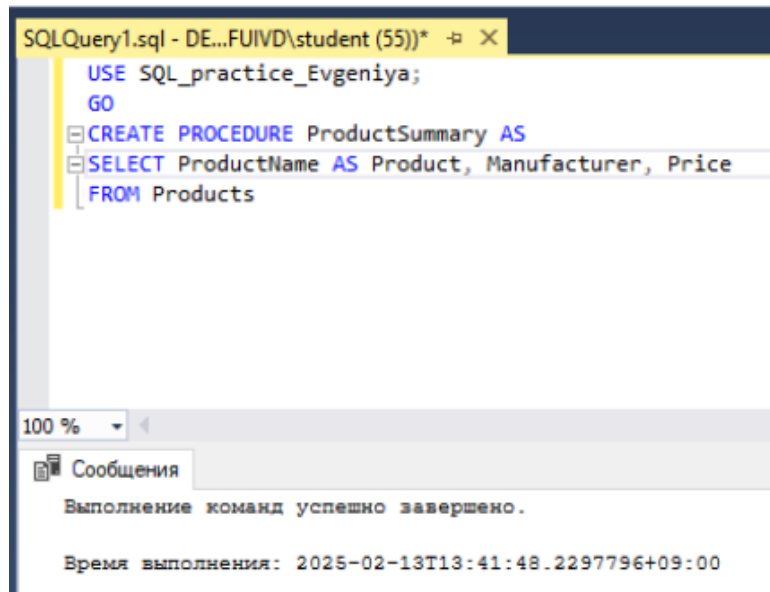
Краткое описание: изучение хранимых процедур.

В ходе выполнения практической работы мы получаем опыт работы в управлении БД используя T-SQL, а также изучим хранимые процедуры.

2. Этапы работы

1. Задание «Создание и выполнение процедур»

Создадим хранимую процедуру для извлечения данных из этой таблицы:



```
SQLQuery1.sql - DE...FUIVD\student (55))* -> X
USE SQL_practice_Evgeniya;
GO
CREATE PROCEDURE ProductSummary AS
SELECT ProductName AS Product, Manufacturer, Price
FROM Products
```

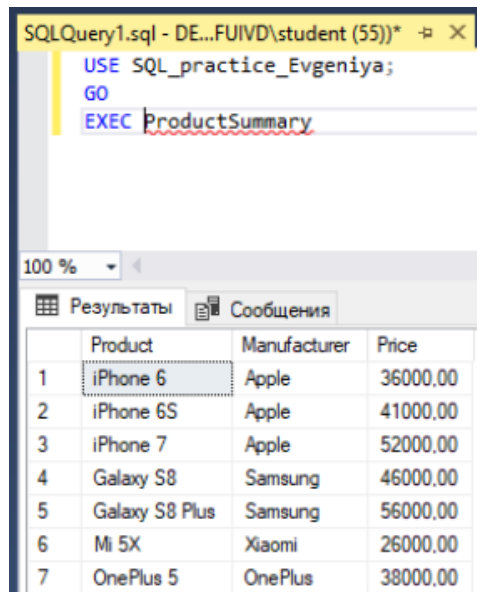
100 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2025-02-13T13:41:48.2297796+09:00

Для выполнения хранимой процедуры вызывается команда EXEC или EXECUTE:



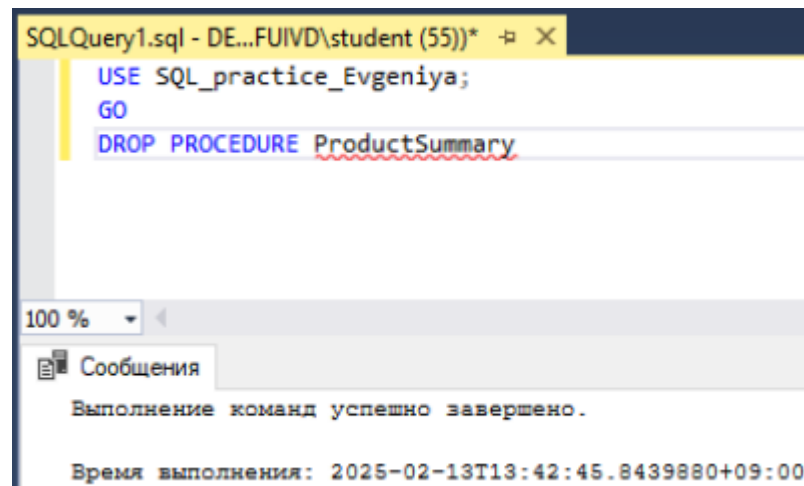
```
SQLQuery1.sql - DE...FUIVD\student (55))* -> X
USE SQL_practice_Evgeniya;
GO
EXEC ProductSummary
```

100 %

Результаты

	Product	Manufacturer	Price
1	iPhone 6	Apple	36000,00
2	iPhone 6S	Apple	41000,00
3	iPhone 7	Apple	52000,00
4	Galaxy S8	Samsung	46000,00
5	Galaxy S8 Plus	Samsung	56000,00
6	Mi 5X	Xiaomi	26000,00
7	OnePlus 5	OnePlus	38000,00

Для удаления процедуры применяется команда DROP PROCEDURE:



The screenshot shows a SQL query window titled "SQLQuery1.sql - DE...FUIVD\student (55))*". The query text is:

```
USE SQL_practice_Evgeniya;  
GO  
DROP PROCEDURE ProductSummary
```

Below the query window, the "Сообщения" (Messages) pane shows the execution status:

Выполнение команд успешно завершено.

Время выполнения: 2025-02-13T13:42:45.8439880+09:00

2. Задание «Параметры в процедурах»

Определим процедуру, которая будет добавлять данные в эту таблицу:

```
SQLQuery1.sql - DE...FUIVD\student (55))* -> X
USE SQL_practice_Evgeniya;
GO
CREATE PROCEDURE AddProduct
    @name NVARCHAR(20),
    @manufacturer NVARCHAR(20),
    @count INT,
    @price MONEY
AS
INSERT INTO Products(ProductName, Manufacturer, ProductCount, Price)
VALUES (@name, @manufacturer, @count, @price)
```

100 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2025-02-13T13:45:07.0157955+09:00

Используем эту процедуру:

```
SQLQuery1.sql - DE...FUIVD\student (55))* -> X
USE SQL_practice_Evgeniya;
GO
DECLARE @prodName NVARCHAR(20), @company NVARCHAR(20);
DECLARE @prodCount INT, @price MONEY
SET @prodName = 'Galaxy C7'
SET @company = 'Samsung'
SET @price = 22000
SET @prodCount = 5

EXEC AddProduct @prodName, @company, @prodCount, @price

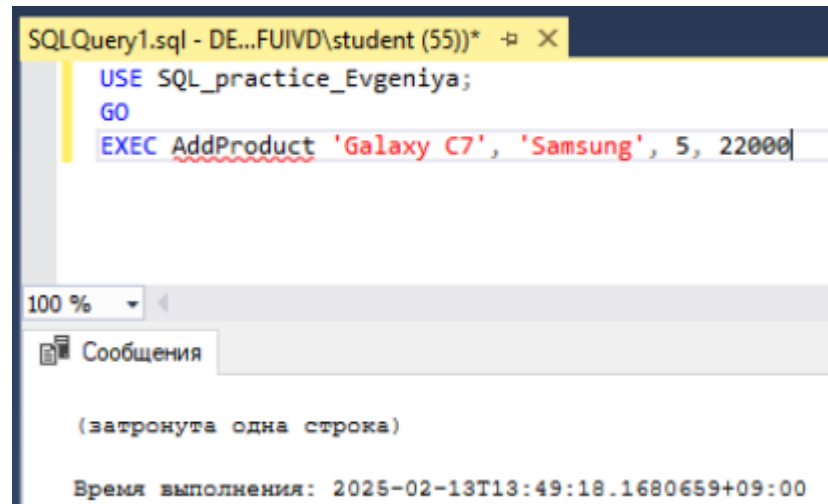
SELECT * FROM Products
```

100 %

Результаты

	Id	ProductName	Manufacturer	ProductCount	Price
1	1	iPhone 6	Apple	2	36000.00
2	2	iPhone 6S	Apple	2	41000.00
3	3	iPhone 7	Apple	5	52000.00
4	4	Galaxy S8	Samsung	2	46000.00
5	5	Galaxy S8 Plus	Samsung	1	56000.00
6	6	Mi 5X	Xiaomi	2	26000.00
7	7	OnePlus 5	OnePlus	6	38000.00
8	1002	Galaxy C7	Samsung	5	22000.00

Передаем значения:



```
SQLQuery1.sql - DE...FUIVD\student (55))* -> X
USE SQL_practice_Evgeniya;
GO
EXEC AddProduct 'Galaxy C7', 'Samsung', 5, 22000
```

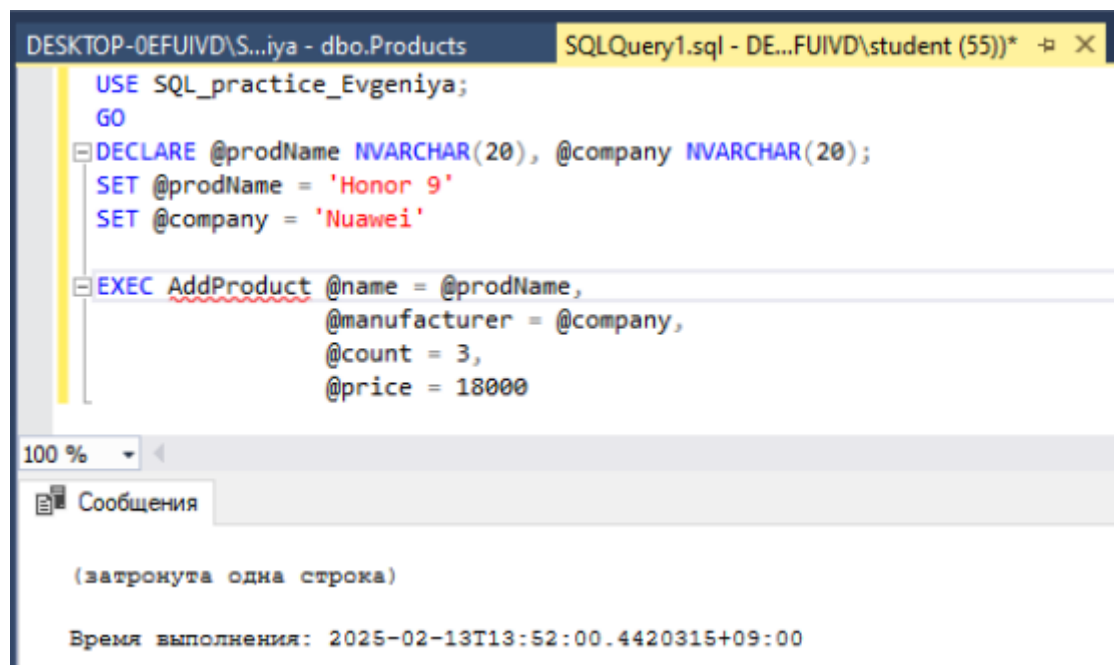
100 %

Сообщения

(затронута одна строка)

Время выполнения: 2025-02-13T13:49:18.1680659+09:00

Передаем по имени:



```
DESKTOP-0EFUIVD\S...iya - dbo.Products SQLQuery1.sql - DE...FUIVD\student (55))* -> X
USE SQL_practice_Evgeniya;
GO
DECLARE @prodName NVARCHAR(20), @company NVARCHAR(20);
SET @prodName = 'Honor 9';
SET @company = 'Nuawei';
EXEC AddProduct @name = @prodName,
                @manufacturer = @company,
                @count = 3,
                @price = 18000
```

100 %

Сообщения

(затронута одна строка)

Время выполнения: 2025-02-13T13:52:00.4420315+09:00

Параметры можно отмечать как необязательные, присваивая им некоторое значение по умолчанию:

```
DESKTOP-0EFUIVD\S...iya - dbo.Products | SQLQuery1.sql - DE...FUIVD\student (55))*
USE SQL_practice_Evgeniya;
GO
CREATE PROCEDURE AddProductWithOptionalCount
    @name NVARCHAR(20),
    @manufacturer NVARCHAR(20),
    @price MONEY,
    @count INT = 1
AS
INSERT INTO Products(ProductName, Manufacturer, ProductCount, Price)
VALUES(@name, @manufacturer, @count, @price)
```

100 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2025-02-13T13:54:23.6303274+09:00

Помещаем в конце списка необязательные параметры:

```
DESKTOP-0EFUIVD\S...iya - dbo.Products | SQLQuery1.sql - DE...FUIVD\student (55))*
USE SQL_practice_Evgeniya;
DECLARE @prodName NVARCHAR(20), @company NVARCHAR(20), @price MONEY
SET @prodName = 'Redmi Note 5A'
SET @company = 'Xiaomi'
SET @price = 22000

EXEC AddProductWithOptionalCount @prodName, @company, @price

SELECT * FROM Products
```

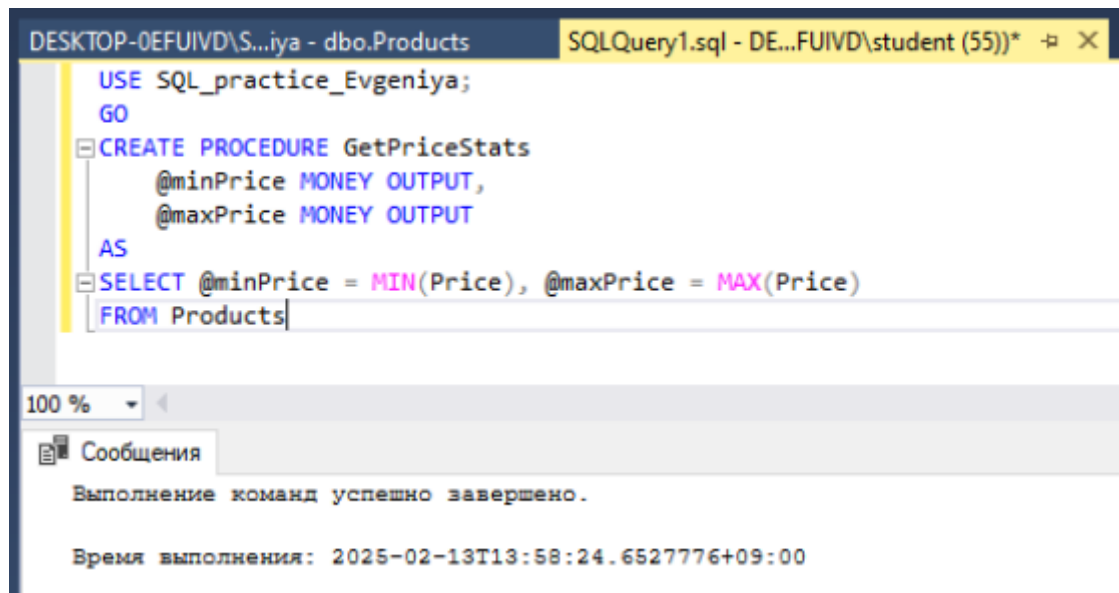
100 %

Результаты | Сообщения

	Id	ProductName	Manufacturer	ProductCount	Price
1	1	iPhone 6	Apple	2	36000,00
2	2	iPhone 6S	Apple	2	41000,00
3	3	iPhone 7	Apple	5	52000,00
4	4	Galaxy S8	Samsung	2	46000,00
5	5	Galaxy S8 Plus	Samsung	1	56000,00
6	6	Mi 5X	Xiaomi	2	26000,00
7	7	OnePlus 5	OnePlus	6	38000,00
8	1002	Galaxy C7	Samsung	5	22000,00
9	1003	Galaxy C7	Samsung	5	22000,00
10	1004	Honor 9	Nuawei	3	18000,00
11	1005	Redmi Note 5A	Xiaomi	1	22000,00

3. Задание «Параметры в процедурах»

Определим процедуру:



```
DESKTOP-0EFUIVD\S...iya - dbo.Products  SQLQuery1.sql - DE...FUIVD\student (55))*  X
USE SQL_practice_Evgeniya;
GO
CREATE PROCEDURE GetPriceStats
    @minPrice MONEY OUTPUT,
    @maxPrice MONEY OUTPUT
AS
SELECT @minPrice = MIN(Price), @maxPrice = MAX(Price)
FROM Products
```

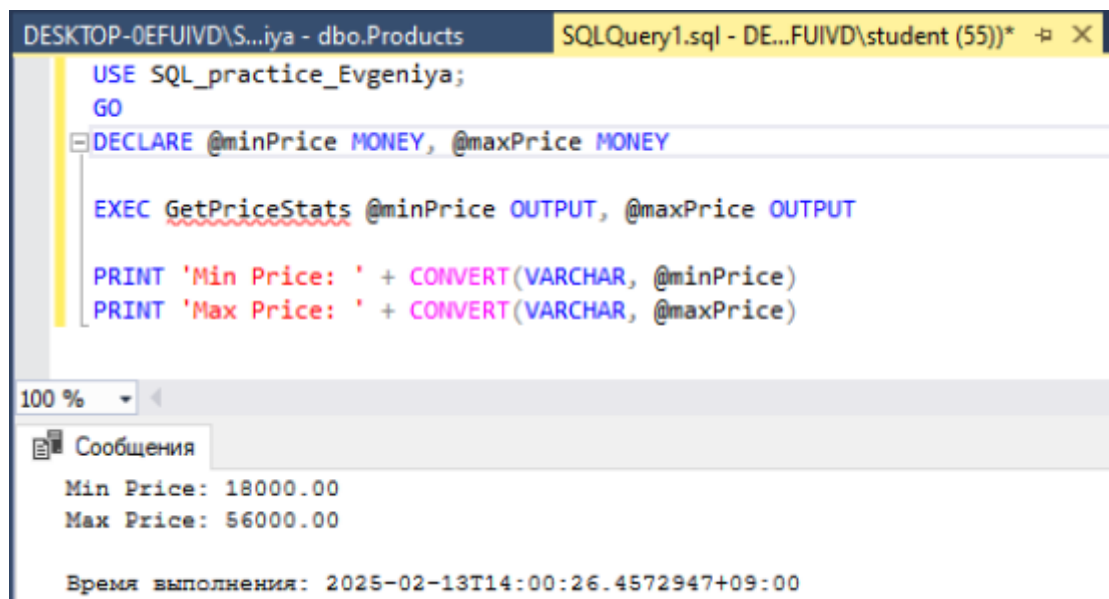
100 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2025-02-13T13:58:24.6527776+09:00

При вызове процедуры для выходных параметров передаются переменные с ключевым словом OUTPUT:



```
DESKTOP-0EFUIVD\S...iya - dbo.Products  SQLQuery1.sql - DE...FUIVD\student (55))*  X
USE SQL_practice_Evgeniya;
GO
DECLARE @minPrice MONEY, @maxPrice MONEY

EXEC GetPriceStats @minPrice OUTPUT, @maxPrice OUTPUT

PRINT 'Min Price: ' + CONVERT(VARCHAR, @minPrice)
PRINT 'Max Price: ' + CONVERT(VARCHAR, @maxPrice)
```

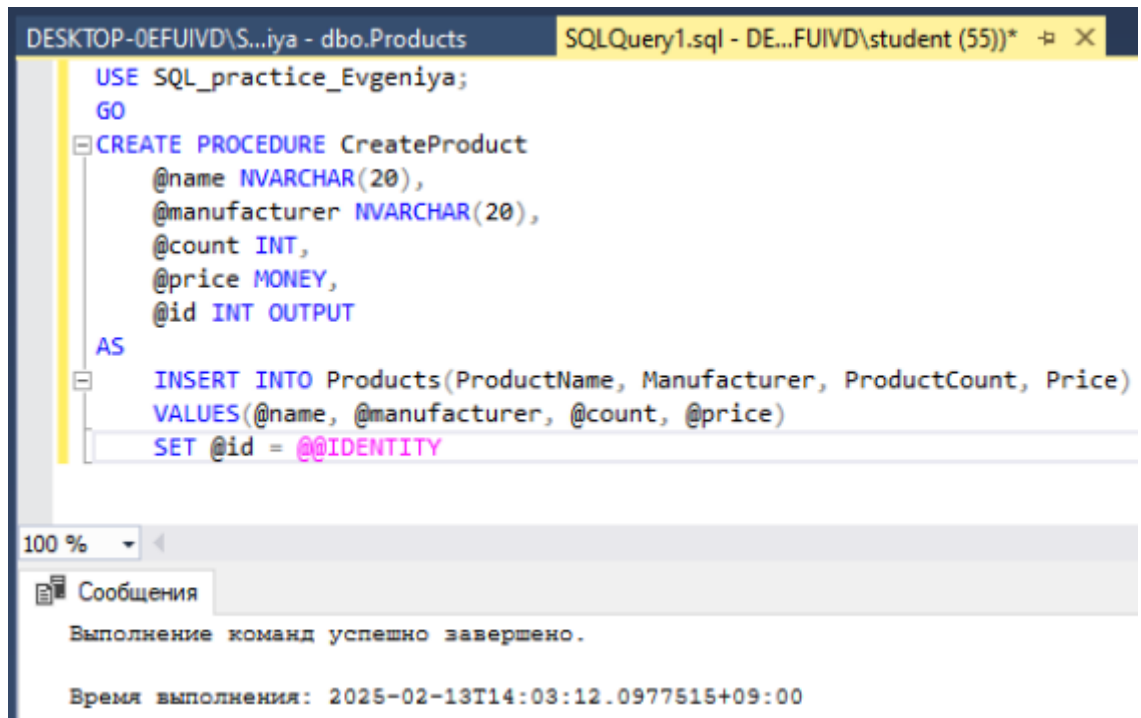
100 %

Сообщения

Min Price: 18000.00
Max Price: 56000.00

Время выполнения: 2025-02-13T14:00:26.4572947+09:00

Определим процедуру, которая добавляет новую строку в таблицу и возвращает ее id:



```
USE SQL_practice_Evgeniya;
GO
CREATE PROCEDURE CreateProduct
    @name NVARCHAR(20),
    @manufacturer NVARCHAR(20),
    @count INT,
    @price MONEY,
    @id INT OUTPUT
AS
INSERT INTO Products(ProductName, Manufacturer, ProductCount, Price)
VALUES(@name, @manufacturer, @count, @price)
SET @id = @@IDENTITY
```

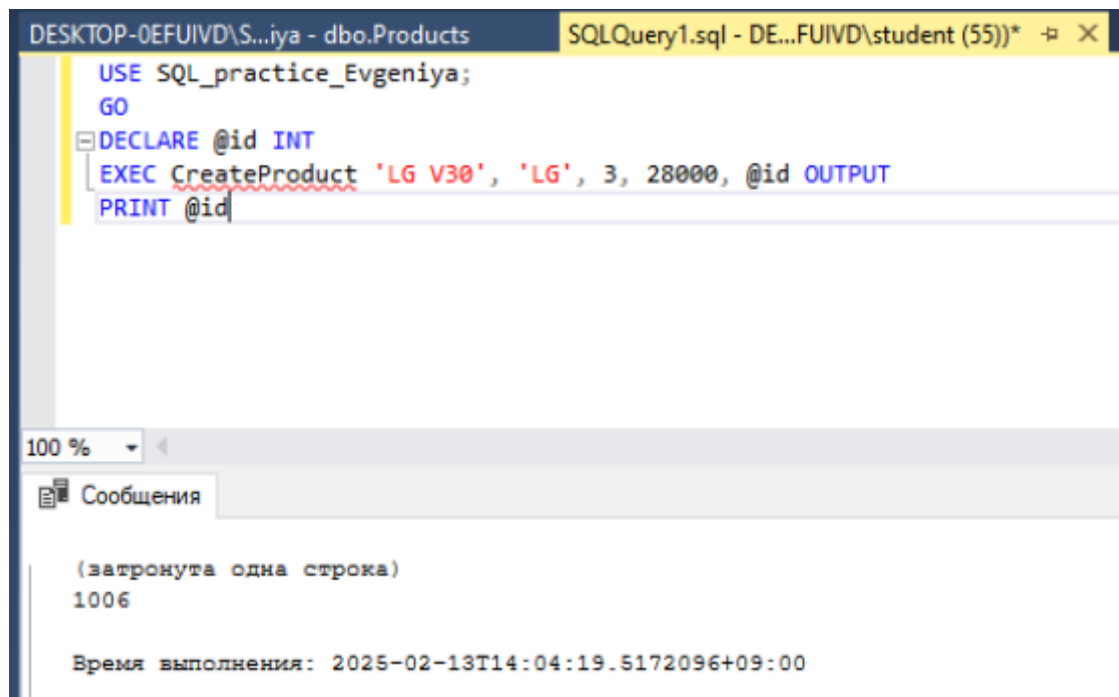
Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2025-02-13T14:03:12.0977515+09:00

С помощью глобальной переменной @@IDENTITY можно получить идентификатор добавленной записи.

При вызове этой процедуры ей также по позиции передаются все входные и выходные параметры:



```
USE SQL_practice_Evgeniya;
GO
DECLARE @id INT
EXEC CreateProduct 'LG V30', 'LG', 3, 28000, @id OUTPUT
PRINT @id
```

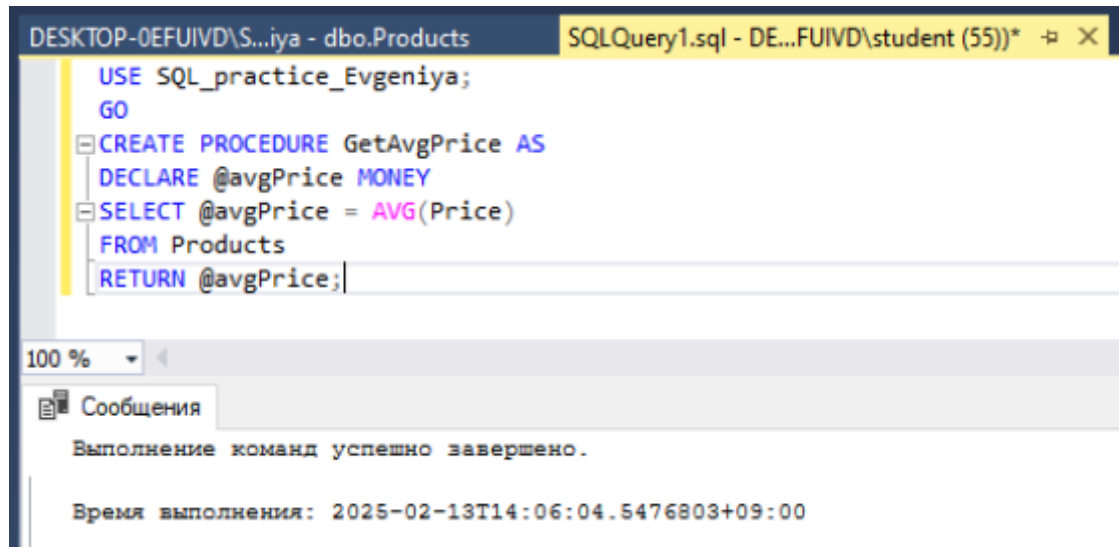
Сообщения

(затронута одна строка)

1006

Время выполнения: 2025-02-13T14:04:19.5172096+09:00

Возвратим среднюю цену на товары:



DESKTOP-0EFUIVD\S...iya - dbo.Products SQLQuery1.sql - DE...FUIVD\student (55))*

```
USE SQL_practice_Evgeniya;  
GO  
CREATE PROCEDURE GetAvgPrice AS  
DECLARE @avgPrice MONEY  
SELECT @avgPrice = AVG(Price)  
FROM Products  
RETURN @avgPrice;
```

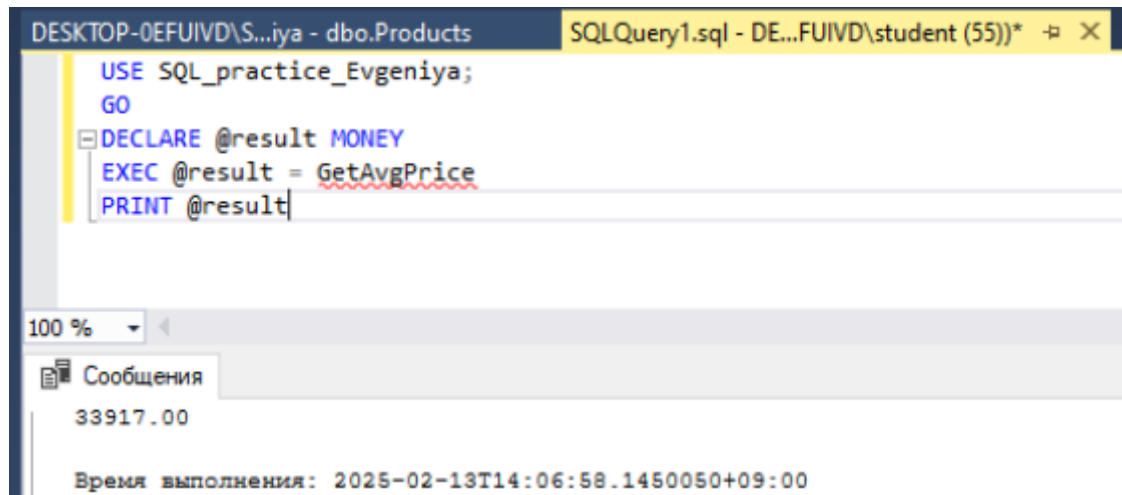
100 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2025-02-13T14:06:04.5476803+09:00

Вызовем процедуру:



DESKTOP-0EFUIVD\S...iya - dbo.Products SQLQuery1.sql - DE...FUIVD\student (55))*

```
USE SQL_practice_Evgeniya;  
GO  
DECLARE @result MONEY  
EXEC @result = GetAvgPrice  
PRINT @result
```

100 %

Сообщения

33917.00

Время выполнения: 2025-02-13T14:06:58.1450050+09:00