

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Систем автоматизированного проектирования

ОТЧЕТ
по лабораторной работе №8
по дисциплине «Базы данных»
ТЕМА: СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ХРАНИМЫХ ПРОЦЕДУР

Студенты гр. 2308

Попов Н.А.

Бебия Р.А.

Чиков А.А.

Преподаватель

Горяинов С.В.

Санкт-Петербург

2024

Цель работы

Целью работы является научиться писать и применять хранимые процедуры. В работе будет использоваться база данных AdventureWorks.

Выполнение работы

Упражнение 1 – создание хранимой процедуры без параметров

Запрос 1: Создание хранимой процедуры GetDiscounts в схеме Sales, которая извлекает следующие столбцы из таблицы Sales.Specialoffer: Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty и MaxQty, с сортировкой по параметрам MinQty и MaxQty.

```
USE AdventureWorks
```

```
GO
```

```
CREATE PROCEDURE Sales.GetDiscounts
```

```
AS
```

```
BEGIN
```

```
SET NOCOUNT ON
```

```
SELECT Description, DiscountPct, Type, Category, StartDate,  
EndDate, MinQty, MaxQty
```

```
FROM Sales.SpecialOffer
```

```
ORDER BY MinQty, MaxQty ASC
```

```
END
```

```
GO
```

Результат выполнения запроса представлен на Рисунке 1.

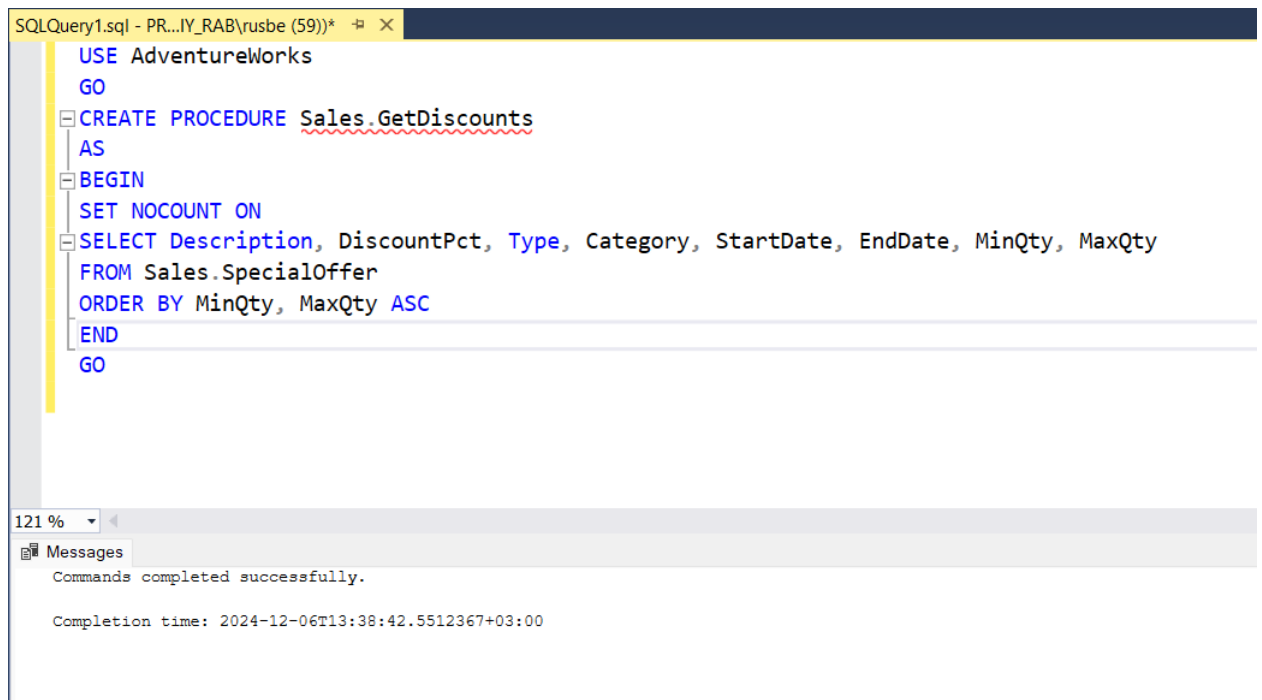


Рисунок 1 – Результат выполнения запроса

Запрос 2: Проверка хранимой процедуры.

EXEC Sales.GetDiscounts

Результат выполнения запроса представлен на Рисунке 2.

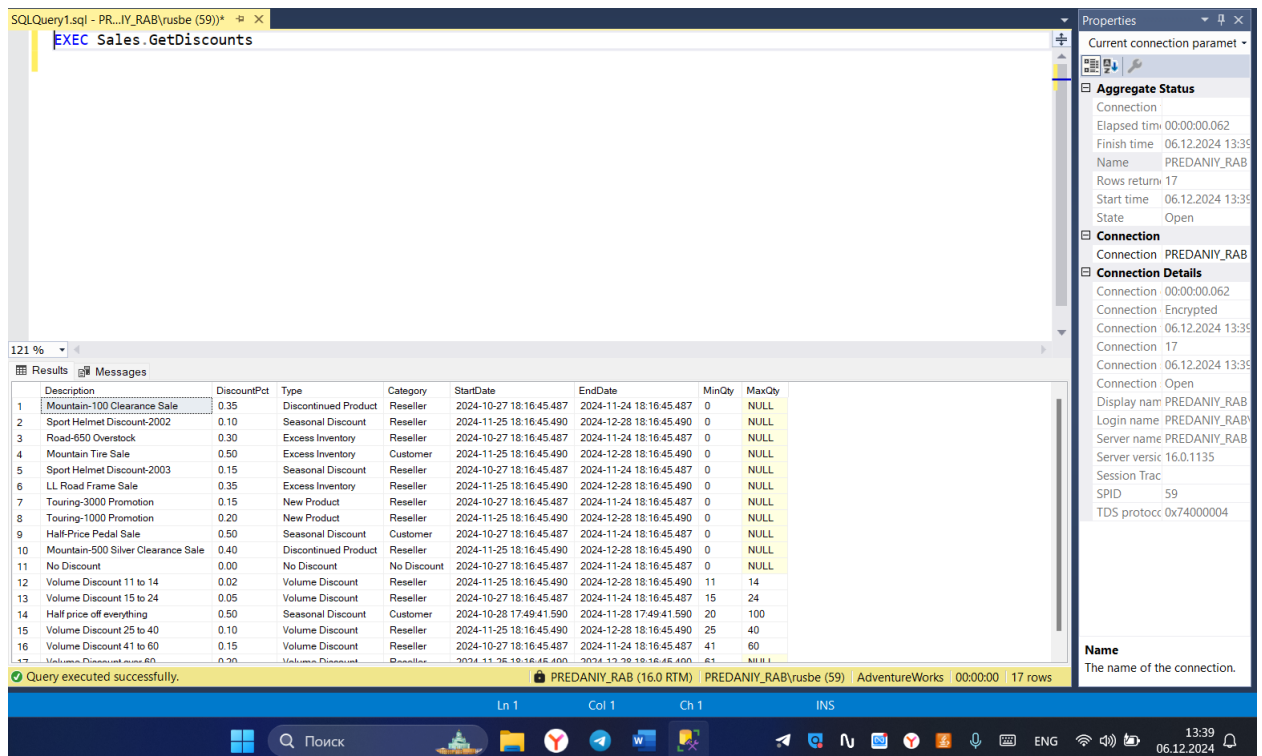


Рисунок 2 – Результат выполнения программы

Упражнение 2 – создание хранимой процедуры с параметром

Запрос 1: Создание хранимой процедуры GetDiscountsForCategory в схеме Sales с входным параметром @Category, который имеет тип данных nvarchar и принимает до 50 символов, с фильтрацией строк на основе @Category.

```
CREATE PROC Sales.GetDiscountsForCategory @Category nvarchar(50)
AS
SELECT Description, DiscountPct, Type, Category, StartDate,
EndDate, MinQty, MaxQty
FROM Sales.SpecialOffer
ORDER BY Category DESC
GO
```

Результат выполнения запроса представлен на Рисунке 3.

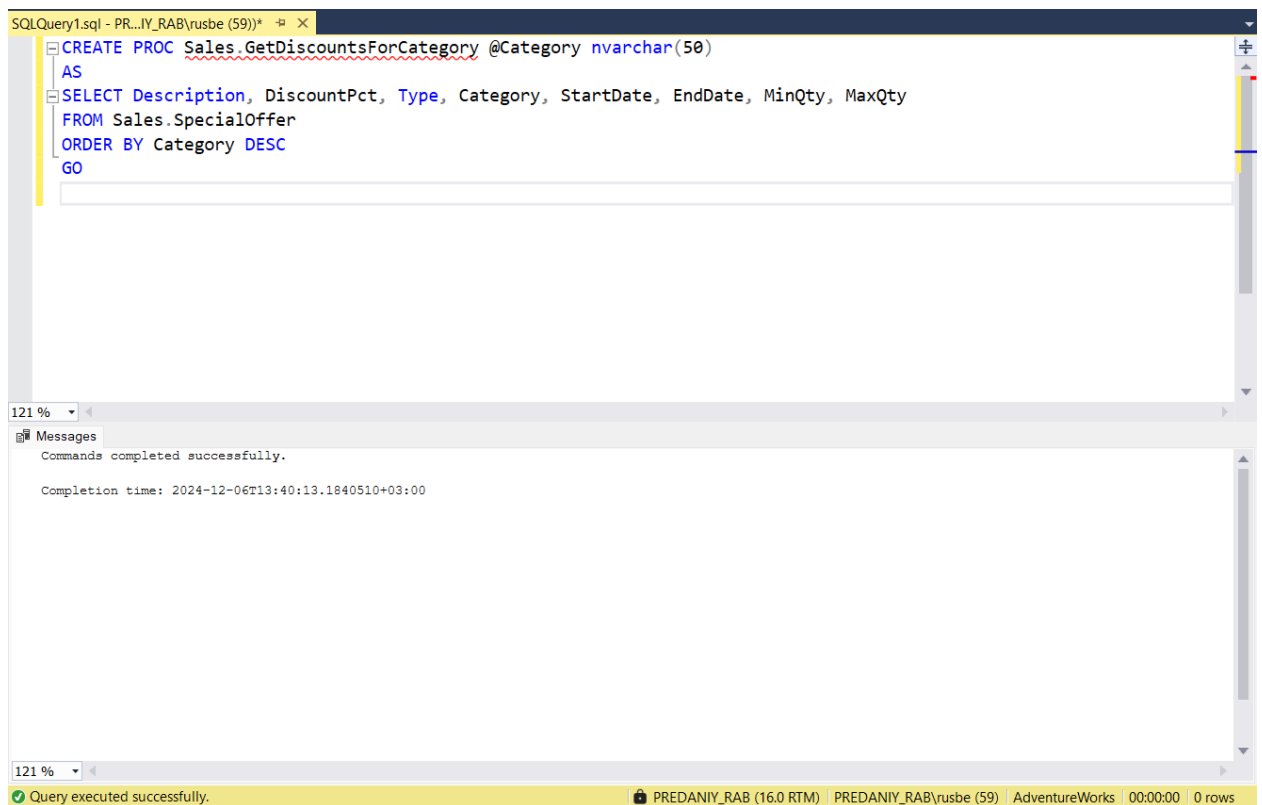
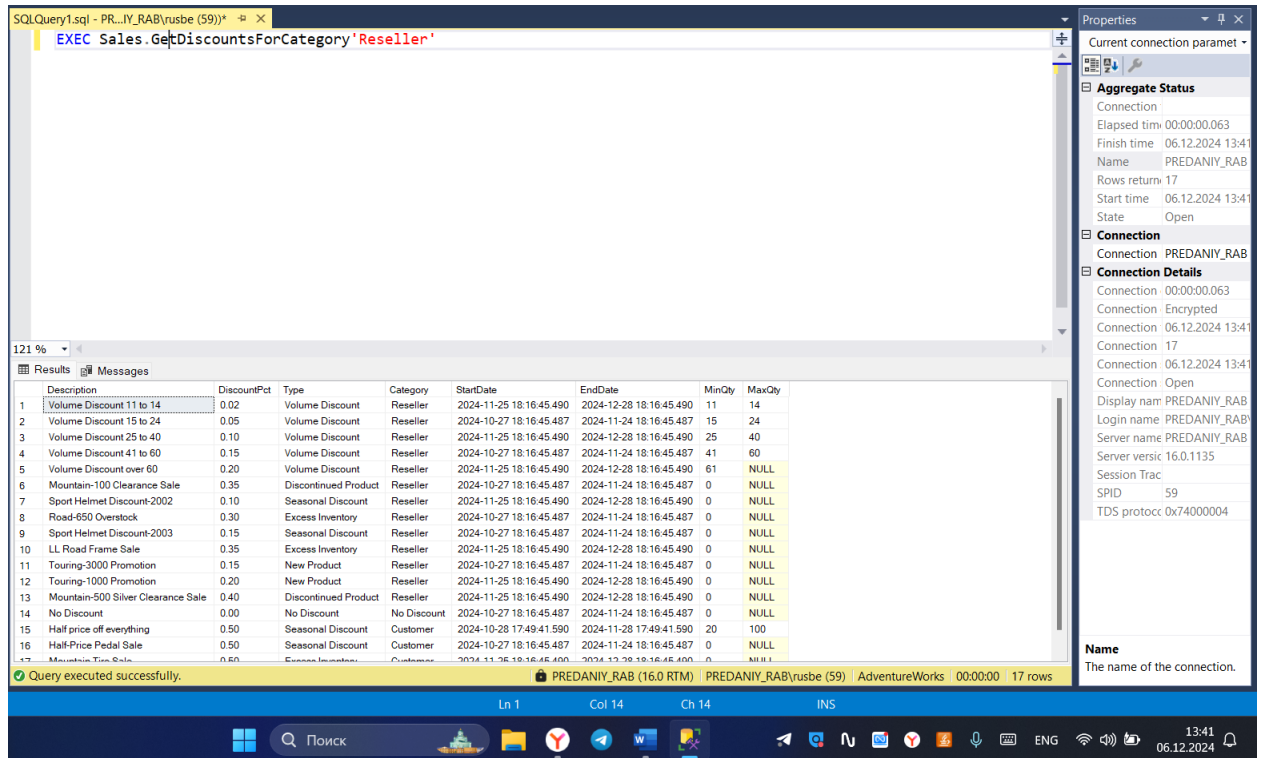


Рисунок 3 – Результат выполнения запроса

Запрос 2: Проверка хранимой процедуры.

```
EXEC Sales.GetDiscounts 'Reseller'
```

Результат выполнения запроса представлен на Рисунке 4.



	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	Volume Discount 11 to 14	0.02	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	11	14
2	Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	15	24
3	Volume Discount 25 to 40	0.10	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	25	40
4	Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	41	60
5	Volume Discount over 60	0.20	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	61	NULL
6	Mountain-100 Clearance Sale	0.35	Discontinued Product	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
7	Sport Helmet Discount-2002	0.10	Seasonal Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
8	Road-650 Overstock	0.30	Excess Inventory	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
9	Sport Helmet Discount-2003	0.15	Seasonal Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
10	LL Road Frame Sale	0.35	Excess Inventory	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
11	Touring-3000 Promotion	0.15	New Product	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
12	Touring-1000 Promotion	0.20	New Product	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
13	Mountain-500 Silver Clearance Sale	0.40	Discontinued Product	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
14	No Discount	0.00	No Discount	Customer	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
15	Half price off everything	0.50	Seasonal Discount	Customer	2024-10-28 17:49:41.590	2024-11-28 17:49:41.590	20	100
16	Half-Price Pedal Sale	0.50	Seasonal Discount	Customer	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
17	Mountain-Tire Sale	0.50	Excess Inventory	Customer	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL

Рисунок 4 – Результат выполнения запроса

Упражнение 3 – создание хранимой процедуры с параметрами и значениями по умолчанию

Запрос 1: Создание хранимой процедуры GetDiscountsForCategoryAndDate в схеме Sales, имеющей кроме параметра @Category параметр @DateToCheck datetime, который может принимать стандартное значение NULL. Если значение NULL указано для параметра @DateToCheck, то для этого параметра задается значение текущих даты и времени с использованием функции GETDATE().

```
CREATE PROC Sales.GetDiscountsForCategoryAndDate
    @Category AS nvarchar(50),
    @DateToCheck AS datetime = NULL
AS
IF (@DateToCheck IS NULL)
BEGIN
    SET @DateToCheck = GETDATE();
```

END

```
SELECT Description, DiscountPct, Type, Category, StartDate,
EndDate, MinQty, MaxQty
FROM Sales.SpecialOffer
WHERE Category = @Category AND StartDate <= @DateToCheck
GO
```

Результат выполнения запроса представлен на Рисунке 5.

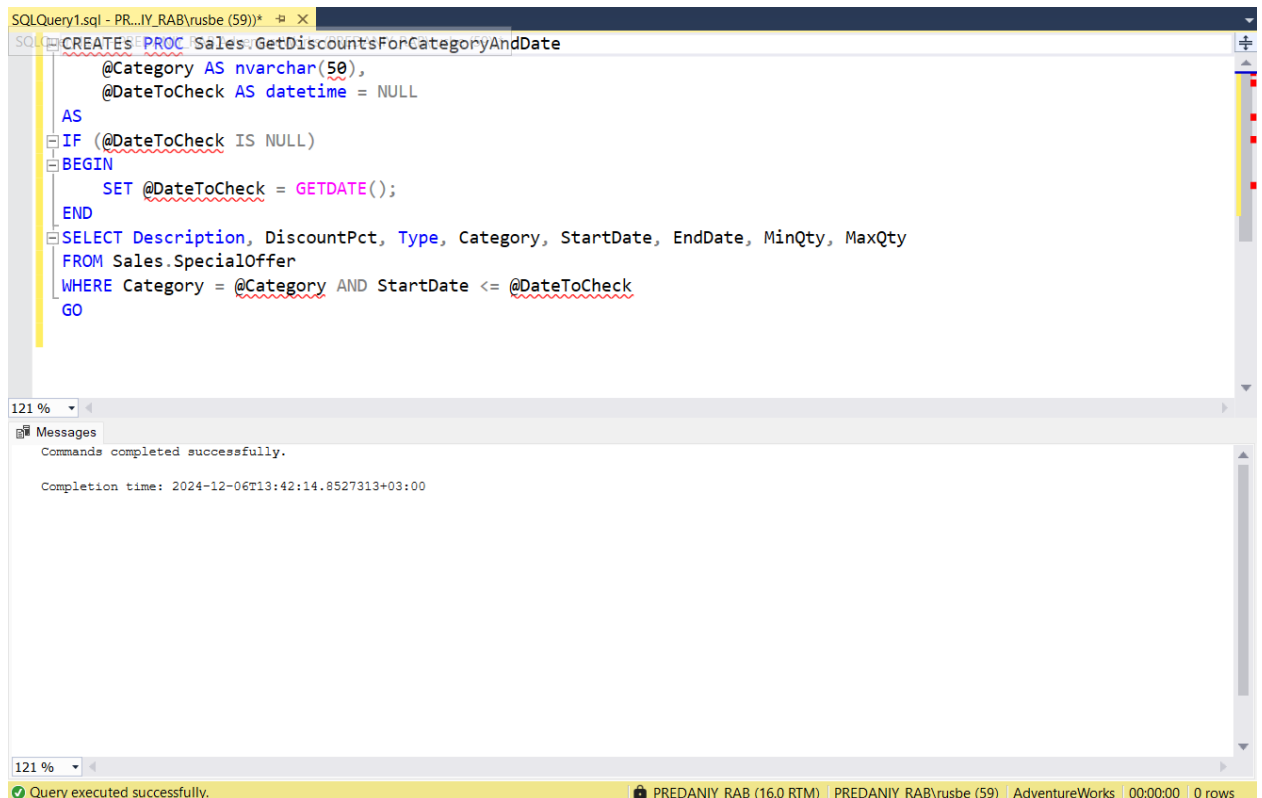


Рисунок 5 – Результат выполнения запроса

Запрос 2: Проверка хранимой процедуры.

```
EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller'
```

Результат выполнения запроса представлен на Рисунке 6.

SQLQuery1.sql - PR...IV_RAB(rusbe (59))

EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller'

Results

Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
Volume Discount 11 to 14	0.02	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	11	14
Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	15	24
Volume Discount 25 to 40	0.10	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	25	40
Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	41	60
Volume Discount over 60	0.20	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	61	NULL
Mountain-100 Clearance Sale	0.35	Discontinued Product	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
Sport Helmet Discount-2002	0.10	Seasonal Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
Road-650 Overstock	0.30	Excess Inventory	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
Sport Helmet Discount-2003	0.15	Seasonal Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
LL Road Frame Sale	0.35	Excess Inventory	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
Touring-3000 Promotion	0.15	New Product	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
Touring-1000 Promotion	0.20	New Product	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
Mountain-500 Silver Clearance Sale	0.40	Discontinued Product	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL

Query executed successfully.

PREDANIY_RAB (16.0 RTM) PREDANIY_RAB(rusbe (59)) AdventureWorks 00:00:00 13 rows

Ln 1 Col 5 Ch 5 INS

Поиск

ENG 13:42 06.12.2024

Properties

Current connection paramet...

Aggregate Status

Connection

Elapsed time 00:00:00.053

Finish time 06.12.2024 13:42

Name PREDANIY_RAB

Rows return 13

Start time 06.12.2024 13:42

State Open

Connection

Connection PREDANIY_RAB

Connection Details

Connection 00:00:00.053

Connection Encrypted

Connection 06.12.2024 13:42

Connection 13

Connection 06.12.2024 13:42

Connection Open

Display nam PREDANIY_RAB

Login name PREDANIY_RAB

Server name PREDANIY_RAB

Server versic 16.0.1135

Session Trac

SPID 59

TDS protoc 0x74000004

Name

The name of the connection.

Рисунок 6 – Результат выполнения запроса

Запрос 3: Проверка варианта, когда указываются оба параметра.

```
DECLARE @DateToCheck datetime
```

```
SET @DateToCheck = DATEADD(MONTH, 1, GETDATE())
```

```
EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller', @DateToCheck
```

Результат выполнения запроса представлен на Рисунке 7.

SQLQuery1.sql - PR...IV_RAB(rusbe (59))

DECLARE @DateToCheck datetime

SET @DateToCheck = DATEADD(MONTH, 1, GETDATE())

EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller', @DateToCheck

Results

Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
Volume Discount 11 to 14	0.02	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	11	14
Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	15	24
Volume Discount 25 to 40	0.10	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	25	40
Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	41	60
Volume Discount over 60	0.20	Volume Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	61	NULL
Mountain-100 Clearance Sale	0.35	Discontinued Product	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
Sport Helmet Discount-2002	0.10	Seasonal Discount	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
Road-650 Overstock	0.30	Excess Inventory	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
Sport Helmet Discount-2003	0.15	Seasonal Discount	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
LL Road Frame Sale	0.35	Excess Inventory	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
Touring-3000 Promotion	0.15	New Product	Reseller	2024-10-27 18:16:45.487	2024-11-24 18:16:45.487	0	NULL
Touring-1000 Promotion	0.20	New Product	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL
Mountain-500 Silver Clearance Sale	0.40	Discontinued Product	Reseller	2024-11-25 18:16:45.490	2024-12-28 18:16:45.490	0	NULL

Query executed successfully.

PREDANIY_RAB (16.0 RTM) PREDANIY_RAB(rusbe (59)) AdventureWorks 00:00:00 13 rows

Ln 4 Col 1 Ch 1 INS

Поиск

ENG 13:42 06.12.2024

Properties

Current connection paramet...

Aggregate Status

Connection

Elapsed time 00:00:00.051

Finish time 06.12.2024 13:42

Name PREDANIY_RAB

Rows return 13

Start time 06.12.2024 13:42

State Open

Connection

Connection PREDANIY_RAB

Connection Details

Connection 00:00:00.051

Connection Encrypted

Connection 06.12.2024 13:42

Connection 13

Connection 06.12.2024 13:42

Connection Open

Display nam PREDANIY_RAB

Login name PREDANIY_RAB

Server name PREDANIY_RAB

Server versic 16.0.1135

Session Trac

SPID 59

TDS protoc 0x74000004

Name

The name of the connection.

Рисунок 7 – Результат выполнения запроса

Упражнение 4 – создание хранимой процедуры с входными и выходными параметрами

Запрос 1: Создание хранимой процедуры AddDiscount в схеме Sales, которая вставляет новые записи в таблицу Sales.SpecialOffer. Команда INSERT защищена обработкой ошибок с использованием конструкции Try...Catch, и любые ошибки записываются в таблицу dbo.ErrorLog. Если новая вставка завершится успешно, параметр @NewProductID обновляется на значение функции SCOPE_IDENTITY.

```
CREATE PROC Sales.AddDiscount
    @Description AS nvarchar(255),
    @DiscountPct AS smallmoney,
    @Type AS nvarchar(50),
    @Category AS nvarchar(50),
    @StartDate AS Datetime,
    @EndDate AS Datetime,
    @MinQty AS int,
    @MaxQty AS int,
    @NewProductID AS int OUTPUT
AS
BEGIN
    BEGIN TRY
        INSERT INTO Sales.SpecialOffer (Description,
DiscountPct, Type, Category, StartDate, EndDate, MinQty, MaxQty)
        VALUES (@Description, @DiscountPct, @Type, @Category,
@StartDate, @EndDate, @MinQty, @MaxQty);
        SET @NewProductID = SCOPE_IDENTITY()
        RETURN 0
    END TRY

    BEGIN CATCH
```



```

INSERT INTO dbo.ErrorLog (ErrorMessage, ErrorLine,
ErrorNumber, ErrorProcedure)

VALUES (ERROR_MESSAGE(), ERROR_LINE(), ERROR_NUMBER(),
ERROR_PROCEDURE())

RETURN 1

END CATCH;

END

```

Результат выполнения запроса представлен на Рисунке 8.

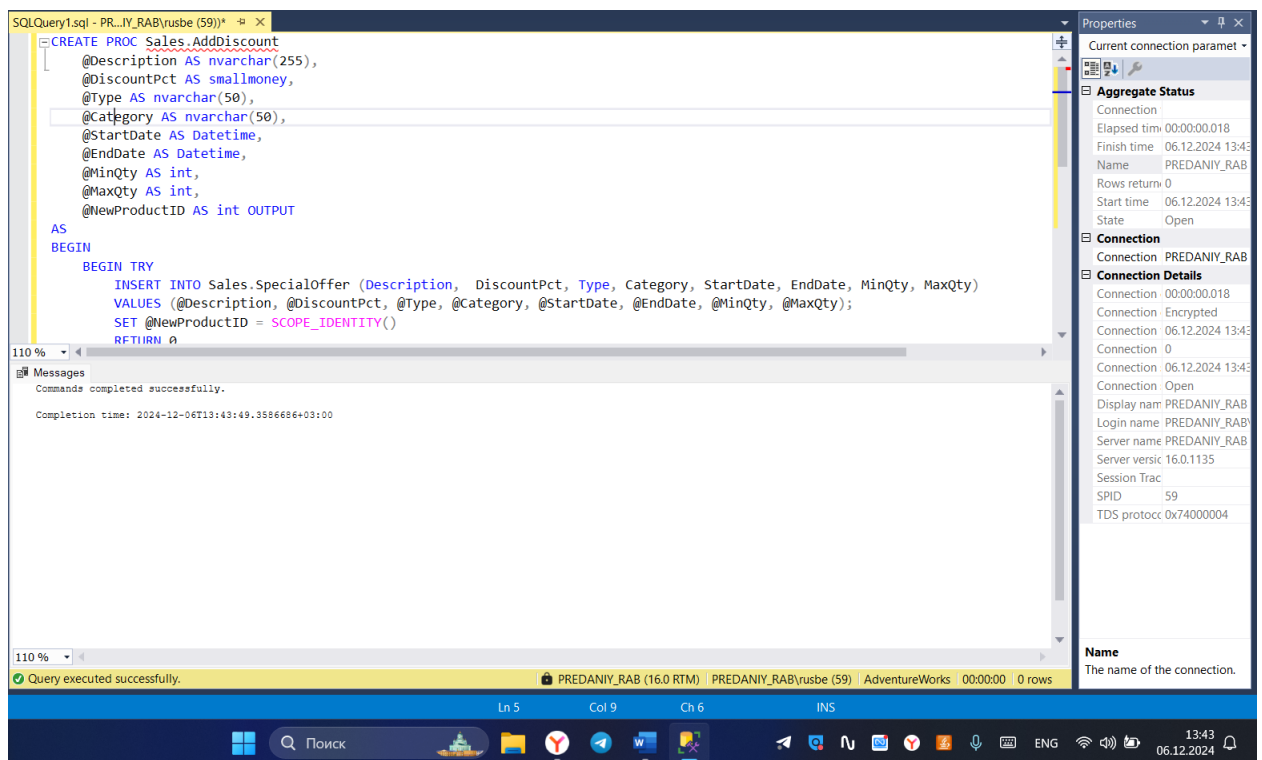


Рисунок 8 – Результат выполнения запроса

Запрос 2: Проверка хранимой процедуры.

```

DECLARE @StartDate datetime, @EndDate datetime
SET @StartDate = GETDATE()
SET @EndDate = DATEADD(month, 1, @StartDate)
DECLARE @NewId int

EXEC Sales.AddDiscount
    'Half price off everything',

```

```

0.5,
'Seasonal Discount',
'Customer',
@StartDate,
@EndDate,
20,
100,
@NewId OUTPUT

```

```
SELECT @NewId
```

Результат выполнения запроса представлен на Рисунке 9.

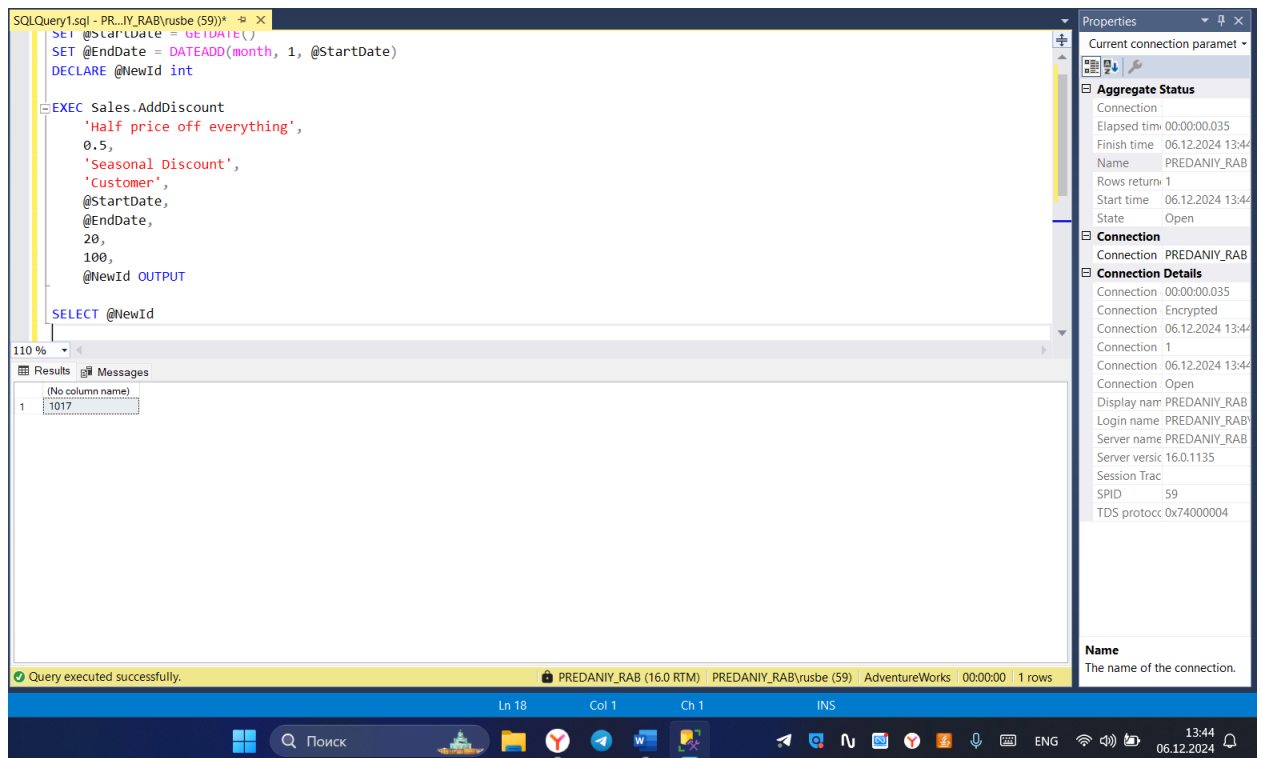


Рисунок 9 – Результат выполнения запроса

Запрос 3: Повторная проверка с использованием некорректных значений.

```

DECLARE @StartDate datetime, @EndDate datetime
SET @StartDate = GETDATE()

```

```

SET @EndDate = DATEADD(month, 1, @StartDate)
DECLARE @NewId int, @ReturntValue int
EXEC @ReturntValue = Sales.AddDiscount
    'Half price off everything',
    -0.5,
    'Seasonal Discount',
    'Customer',
    @StartDate,
    @EndDate,
    0,
    20,
    @NewId OUTPUT
IF (@ReturntValue = 0)
    SELECT @NewId
ELSE
    SELECT TOP 1 * FROM dbo.ErrorLog ORDER BY ErrorTime DESC

```

Результат выполнения запроса представлен на Рисунке 10.

ErrorLogID	ErrorTime	UserName	ErrorNumber	ErrorSeverity	ErrorState	ErrorProcedure	ErrorLine	ErrorMessage
1	2024-10-24 14:04:59.630	RYMARMARYuser	547	16	NULL	Sales.AddDiscount	5	Конфликт инструкции INSERT с ограничением CHECK "...

(0 rows affected)

(1 rows affected)

(1 rows affected)

Completion time: 2024-12-06T13:46:35.2409890+03:00

Рисунок 10 – Результат выполнения запроса

Вывод

В ходе выполнения лабораторной работы мы познакомились с созданием и использованием хранимых процедур на примере базы данных AdventureWorks. Были рассмотрены различные типы хранимых процедур:

1. **Процедуры без параметров** — для выполнения стандартных запросов без передачи дополнительных данных.
2. **Процедуры с параметрами** — позволяющие использовать входные параметры для фильтрации и отбора данных.
3. **Процедуры с параметрами по умолчанию** — дающие возможность задавать значения параметров по умолчанию, что упрощает использование процедуры.
4. **Процедуры с входными и выходными параметрами** — для обработки данных и возврата идентификаторов новых записей при вставке данных.

Кроме того, мы изучили механизмы обработки ошибок в хранимых процедурах с использованием конструкции TRY...CATCH. Это позволило эффективно управлять ошибками и, в случае их возникновения, сохранять информацию об ошибках в отдельной таблице.

Таким образом, данная лабораторная работа продемонстрировала, как с помощью хранимых процедур можно упростить выполнение сложных операций с данными, повысить безопасность и удобство работы с базой данных, а также улучшить производительность за счёт сокращения сетевых запросов и оптимизации обработки данных на стороне сервера.