

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»**

ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

197376, Санкт-Петербург, ул. проф. Попова, 5.

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

«З А Ч Т Е Н О»

_____ О.А. Жирнова

“ ” _____ 2021 г.

ОТЧЁТ
по дисциплине «Базы данных»
Лабораторная работа № 8
«Создание и использование хранимых процедур»

Студент группы 9308

Н.В. Яловега

Санкт Петербург 2021

Цель работы: научиться писать и применять хранимые процедуры

Используемая база данных (БД): AdventureWorks

Порядок выполнения

Упражнение 1 – создание хранимой процедуры без параметров.

Запрос 1. Создание процедуры, извлекающей столбцы Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty и MaxQty. Процедура возвращает все строки, отсортированные по StartDate и EndDate. Результат на рисунке 1.

```
CREATE PROC [Sales].[GetDiscounts]
AS BEGIN
SELECT Description, DiscountPct, Type, Category, StartDate, EndDate, Min-
Qty,
MaxQty
FROM AdventureWorks.Sales.SpecialOffer
ORDER BY StartDate, EndDate ASC
END
```

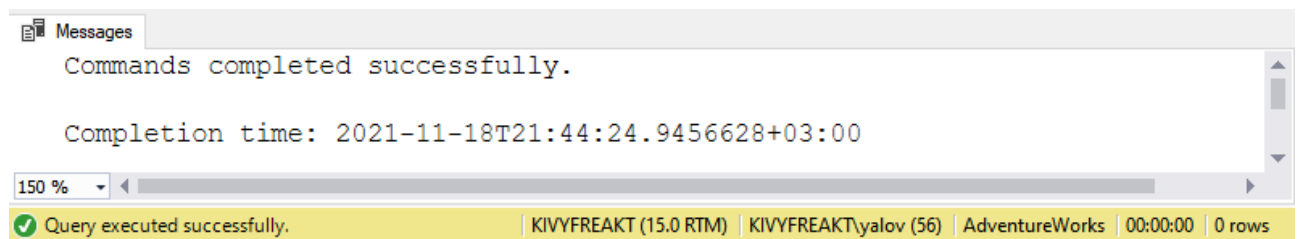


Рисунок 1

Запрос 2. Проверка процедуры. Результат на рисунке 2.

EXEC Sales.GetDiscounts

Results		Messages						
	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	No Discount	0.00	No Discount	No Discount	2001-06-01 00:00:00.000	2004-12-31 00:00:00.000	0	NULL
2	Volume Discount 11 to 14	0.02	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	11	14
3	Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	15	24
4	Volume Discount 25 to 40	0.10	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	25	40
5	Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
6	Volume Discount over 60	0.20	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	NULL
7	Mountain-100 Clearanc...	0.35	Discontinued P...	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
8	Sport Helmet Discount-2...	0.10	Seasonal Disc...	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
9	Road-650 Overstock	0.30	Excess Inventory	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
10	Mountain Tire Sale	0.50	Excess Inventory	Customer	2003-06-15 00:00:00.000	2003-08-30 00:00:00.000	0	NULL
11	Sport Helmet Discount-2...	0.15	Seasonal Disc...	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
12	LL Road Frame Sale	0.35	Excess Inventory	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
13	Touring-3000 Promotion	0.15	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
14	Touring-1000 Promotion	0.20	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
15	Half-Price Pedal Sale	0.50	Seasonal Disc...	Customer	2003-08-15 00:00:00.000	2003-09-15 00:00:00.000	0	NULL
16	Mountain-500 Silver Cle...	0.40	Discontinued P...	Reseller	2004-05-01 00:00:00.000	2004-06-30 00:00:00.000	0	NULL
Query executed successfully.								
KIVYFREKT (15.0 RTM) KIVYFREKT\yalov (56) AdventureWorks 00:00:00 16 rows								

Рисунок 2

Упражнение 2 – создание хранимой процедуры с параметрами.

Запрос 1. Создание процедуры с параметром @Category, имеющий тип данных nvarchar и принимающий до 50 символов. Она извлекает те же столбцы, что и запрос в процедуре GetDiscounts, но фильтрует строки на основе параметра @Category. Результат на рисунке 3.

```
CREATE PROC [Sales].[GetDiscountsForCategory] @Category AS nvarchar(50)
AS BEGIN
SELECT Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty,
MaxQty
FROM AdventureWorks.Sales.SpecialOffer
WHERE Category = @Category
ORDER BY StartDate, EndDate ASC
END
```

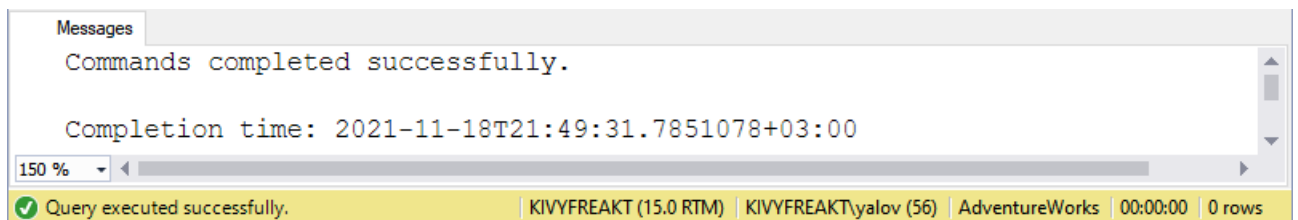


Рисунок 3

Запрос 2. Проверка процедуры на слове «reseller». Результат на рисунке 4.

```
EXEC Sales.GetDiscountsForCategory 'Reseller'
```

The screenshot shows the 'Results' window in SQL Server Enterprise Manager. It displays a table with 13 rows and 9 columns. The columns are: Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty, and MaxQty. The data is filtered by the category 'Reseller'. The status bar at the bottom indicates 'Query executed successfully.' and provides details: 'KIVYFREKT (15.0 RTM) | KIVYFREKT\yakov (56) | AdventureWorks | 00:00:00 | 13 rows'.

	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	Volume Discount 11 to 14	0.02	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	11	14
2	Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	15	24
3	Volume Discount 25 to 40	0.10	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	25	40
4	Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
5	Volume Discount over 60	0.20	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	NULL
6	Mountain-100 Clearanc...	0.35	Discontinued P...	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
7	Sport Helmet Discount-2...	0.10	Seasonal Disc...	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
8	Road-650 Overstock	0.30	Excess Inventory	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
9	Sport Helmet Discount-2...	0.15	Seasonal Disc...	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
10	LL Road Frame Sale	0.35	Excess Inventory	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
11	Touring-3000 Promotion	0.15	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
12	Touring-1000 Promotion	0.20	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
13	Mountain-500 Silver Cle...	0.40	Discontinued P...	Reseller	2004-05-01 00:00:00.000	2004-06-30 00:00:00.000	0	NULL

Рисунок 4

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

Запрос 1. Создание процедуры с параметрами @Category и @DateToCheck, который имеет стандартное значение NULL. Результат на рисунке 5.

```
CREATE PROC [Sales].[GetDiscountsForCategoryAndDate] @Category AS  
nvarchar(50), @Date AS datetime = NULL  
AS  
BEGIN  
IF (@Date IS NULL)  
SET @Date = GETDATE();  
SELECT Description, DiscountPct, Type, Category, StartDate, EndDate, Min-  
Qty,  
MaxQty  
FROM AdventureWorks.Sales.SpecialOffer  
WHERE Category = @Category and StartDate <= @Date and EndDate >=  
@Date  
ORDER BY StartDate, EndDate ASC  
END
```

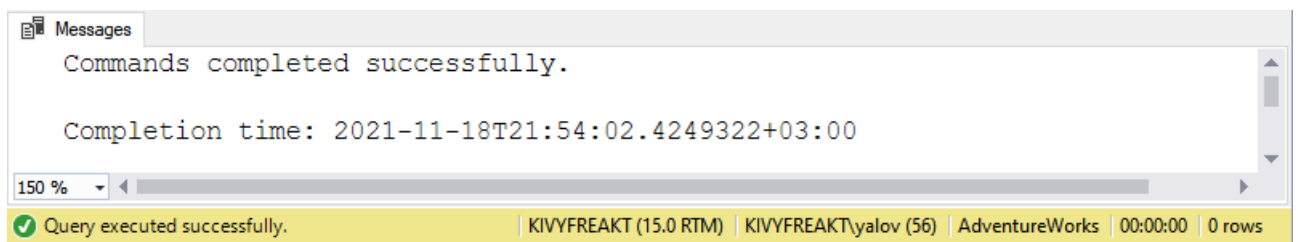


Рисунок 5

Запрос 2. Проверка процедуры с одним параметром. Результат на рисунке 6.

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

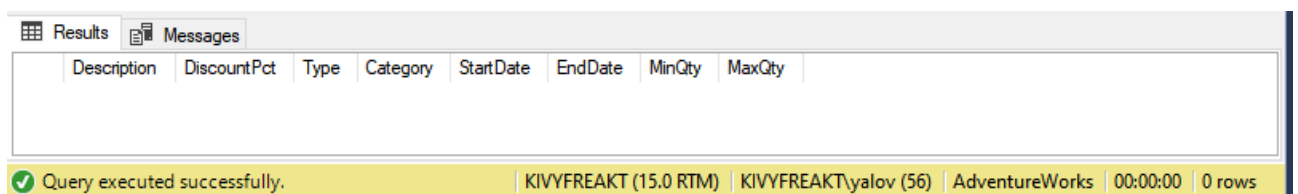


Рисунок 6

Запрос 3. Проверка процедуры с двумя параметрами. Результат на рисунке 7.

```
DECLARE @DateToCheck datetime
SET @DateToCheck = DATEADD(YY, -17, GETDATE())
EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller', @DateToCheck
```

Results		Messages						
	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	Volume Discount 11 to 14	0,02	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	11	14
2	Volume Discount 15 to 24	0,05	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	15	24
3	Volume Discount 25 to 40	0,10	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	25	40
4	Volume Discount 41 to 60	0,15	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
5	Volume Discount over 60	0,20	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	NULL

Query executed successfully.

KIVYFREKT (15.0 RTM)

KIVYFREKT\ygalov (56)

AdventureWorks

00:00:00

5 rows

Рисунок 7

Упражнение 4 – создание хранимой процедуры с входными и выходными параметрами.

Название параметра	Тип данных
@Description	nvarchar(255)
@DiscountPct	smallmoney
@Type	nvarchar(50)
@Category	nvarchar(50)
@StartDate	Datetime
@EndDate	Datetime
@MinQty	int
@MaxQty	int
@NewProductID	int (output)

Рисунок 8

Запрос 1. Создание процедуры. Результат на рисунке 9.

```

CREATE PROC [Sales].[AddDiscount]
@Description as nvarchar(255),
@DiscountPt 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
BEGIN TRAN
INSERT Sales.SpecialOffer (Description, DiscountPct, Type,
Category, StartDate, EndDate, MinQty, MaxQty)
VALUES (@Description, @DiscountPt, @Type, @Category,
@StartDate, @EndDate, @MinQty, @MaxQty);
SET @NewProductID = SCOPE_IDENTITY()
COMMIT TRAN
RETURN 0
END TRY
BEGIN CATCH
IF @@TRANCOUNT > 0
ROLLBACK TRAN
INSERT dbo.ErrorLog (ErrorMessage, ErrorLine, ErrorNumber,
ErrorSeverity, ErrorTime, Username, ErrorProcedure)
VALUES (ERROR_MESSAGE(), ERROR_LINE(), ERROR_NUMBER(),
ERROR_SEVERITY(), SYSDATETIME(), SYSTEM_USER, ERROR_PROCEDURE())

```

```

RETURN 1
END CATCH;
END

```

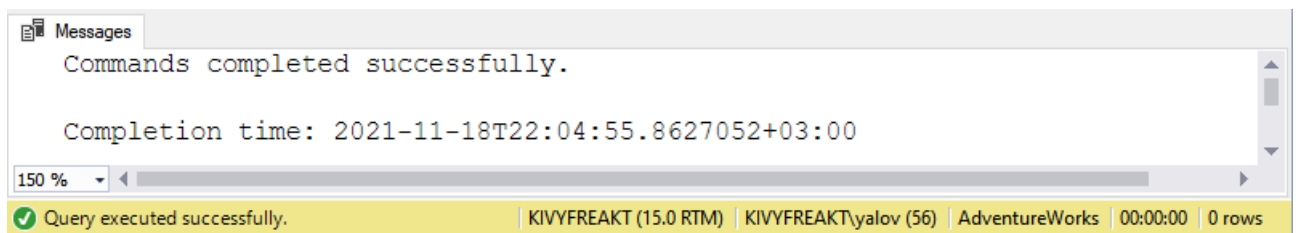


Рисунок 9

Запрос 2. Проверка процедуры. Результат на рисунке 10.

```

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,
0,
20,
@NewId OUTPUT;
SELECT @NewId

```



Рисунок 10

Запрос 3. Проверка процедуры с ошибкой. Результат на рисунке 11.

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

Results		Messages								
	ErrorLogID	ErrorTime	UserName	ErrorNumber	ErrorSeverity	ErrorState	ErrorProcedure	ErrorLine	ErrorMessage	
1	1	2021-11-18 22:07:27.967	KIVYFREKT\yalov	547	16	NULL	Sales.AddDiscount	15	The INSERT	

< >

✓ Query executed successfully. | KIVYFREKT (15.0 RTM) | KIVYFREKT\yalov (56) | AdventureWorks | 00:00:00 | 1 rows

Рисунок 11

Выводы

Были написаны и применены хранимые процедуры, как без параметров, так и с ними (по умолчанию, входные и выходные). Создание хранимой процедуры происходит с помощью команды CREATE PROC. Создание хранимой процедуры с параметром – CREATE PROC @...., с параметром по умолчанию – CREATE PROC @.... = NULL.

Список использованных источников

1. Горячев А. В., Новакова Н. Е. Распределенные базы данных. Мет. указания к лаб. работам., СПб. Изд-во СПбГЭТУ «ЛЭТИ», 2008
2. Горячев А.В, Новакова Н.Е. Особенности разработки и администрирования приложений баз данных: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2016. 68 с.
3. Дейт К. Введение в системы баз данных. : Пер. с англ. – 6-е изд. - К.:Диалектика, 1998.