

Министерство науки и высшего образования Российской Федерации

**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

_____ С. А.Дубенков

Санкт Петербург 2021

Цель работы: научиться писать и применять хранимые процедуры [лит 1].

Используемая база данных (БД): AdventureWorks.

Порядок выполнения

Упражнение 1 – создание хранимой процедуры без параметров

Создана хранимая процедура GetDiscounts в схеме Sales, которая извлекает следующие столбцы из таблицы Sales.SpecialOffer: Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty и MaxQty. Процедура возвращает все строки, отсортированные по параметрам StartDate и EndDate.

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

Результат выполнения запроса показан на рисунке 1

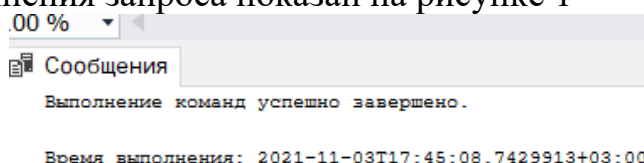


Рисунок 1

Проверка хранимой процедуры: `exec Sales.GetDiscounts`

Результат выполнения запроса показан на рисунке 2

Результаты		Сообщения						
	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 Clearance Sale	0,35	Discontinued Product	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
8	Sport Helmet Discount-2002	0,10	Seasonal Discount	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-2003	0,15	Seasonal Discount	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 Discount	Customer	2003-08-15 00:00:00.000	2003-09-15 00:00:00.000	0	NULL
16	Mountain-500 Silver Clearance Sale	0,40	Discontinued Product	Reseller	2004-05-01 00:00:00.000	2004-06-30 00:00:00.000	0	NULL

Рисунок 2

Упражнение 2 – создание хранимой процедуры с параметром

Создана хранимая процедура GetDiscountForCategory в схеме Sales. Эта процедура имеет входной параметр @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.

```
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
```

Результат выполнения запроса показан на рисунке 3

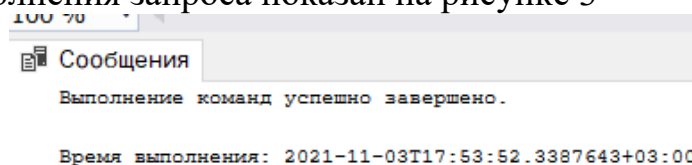


Рисунок 3

Проверка хранимой процедуры: EXEC

Sales.GetDiscountsForCategory 'Reseller'

Результат выполнения запроса показан на рисунке 4

Результаты		Сообщения							
	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 Clearance Sale	0.35	Discontinued Product	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL	
7	Sport Helmet Discount-2002	0.10	Seasonal Discount	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-2003	0.15	Seasonal Discount	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 Clearance Sale	0.40	Discontinued Product	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

В схеме Sales создана хранимая процедура GetDiscountsForCategoryAndDate. Эта процедура имеет входной параметр @Category, как и процедура GetDiscountsForCategory, но включает допол-

нительный входной параметр @DateToCheck datetime. Параметр @DateToCheck имеет возможность принимать стандартное значение NULL. Если значение NULL указано для параметра @DateToCheck, задан для этого параметра значение текущих даты и времени, используя функцию GETDATE()

```
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, MinQty, MaxQty
    FROM AdventureWorks.Sales.SpecialOffer
    WHERE Category = @Category
    and StartDate <= @Date
    and EndDate >= @Date
    ORDER BY StartDate, EndDate ASC
END
```

Результат выполнения запроса показан на рисунке 5

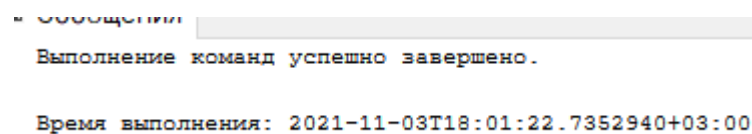


Рисунок 3

Проверка хранимой процедуры:

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

Результат выполнения запроса показан на рисунке 6

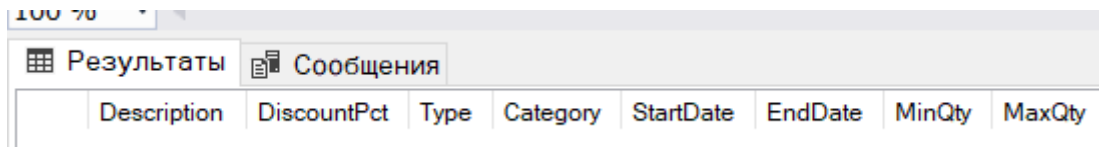
The screenshot shows a table with the following columns: Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty, MaxQty. The table is currently empty.

Рисунок 4

Проверка хранимой процедуры с обоими параметрами:

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

Результат выполнения запроса показан на рисунке 7



Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
-------------	-------------	------	----------	-----------	---------	--------	--------

Рисунок 5

Упражнение 4 – создание хранимой процедуры с входными и выходными параметрами

Создана процедура AddDiscount в схеме Sales, которая вставляет новые записи в таблицу Sales.SpecialOffer. В таблице 1 указаны требуемые параметры для вставки

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)

Таблица 1

Команда INSERT защищена соответствующей обработкой ошибок, и любые ошибки будут записаны в таблицу dbo.ErrorLog. Если новая вставка завершится успешно, параметр @NewProductID будет обновлен на значение функции SCOPE_IDENTITY. Возвращаемое значение указывает успех или неудачу вставки. Для обработки ошибок использована конструкция Try-Catch

```
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, Error-
Number,
ErrorSeverity, ErrorTime, UserName, ErrorProcedure)
            VALUES (ERROR_MESSAGE(), ERROR_LINE(), ERROR_NUMBER(),
ERROR_SEVERITY(), SYSDATETIME(), SYSTEM_USER, ER-
ROR_PROCEDURE())

            RETURN 1
        END CATCH;
END

```

Результат выполнения запроса показан на рисунке 8

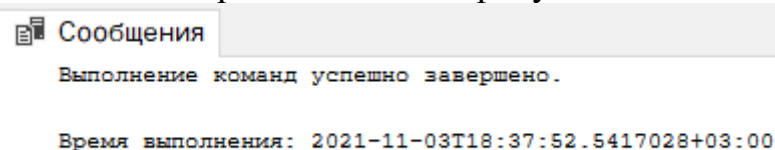


Рисунок 8

Проверка хранимой процедуры.

С помощью следующего запроса убедимся, что в результатах запроса возвращается новое значение параметра SpecialOfferID:

```

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

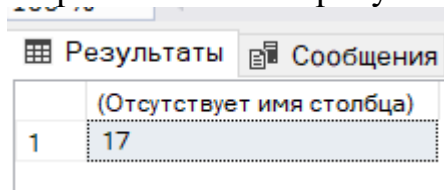
```

```

DECLARE @NewId int;
EXEC Sales.AddDiscount
    'Half price off everything',
    0.5,
    'Seasonal Discount',
    'Customer',
    @StartDate,
    @EndDate,
    0,
    20,
    @NewId OUTPUT;
SELECT @NewId

```

Результат выполнения запроса показан на рисунке 9



	(Отсутствует имя столбца)
1	17

Рисунок 9

Повторная проверка хранимой процедуры.

С помощью следующего запроса убедимся, что в результате отображается запись об ошибке, исключая значение параметра

ErrorProcedure, равное AddDiscount:

```

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

Результат выполнения запроса показан на рисунке 10

Результаты		Сообщения							
	ErrorLogID	ErrorTime	UserName	ErrorNumber	ErrorSeverity	ErrorState	ErrorProcedure	ErrorLine	ErrorMessage
1	1	2021-11-03 18:49:30.230	LAPTOP-G41BAT0M\Семен	547	16	NULL	Sales.AddDiscount	15	Конфликт инструкции INSERT с ограничением CHECK "...

Рисунок 10

Выводы: в ходе выполнения данной лабораторной работы была достигнута ее цель – 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

Методические указания к лабораторным работам / Сост.: А. В. Горячев, Н.Е. Новакова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 32 с.