

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

ОТЧЕТ
по лабораторной работе №9
по дисциплине «Базы данных»
Тема: Создание UDF

Студент гр. 1302

Новиков Г.В.

Студентка гр. 1302

Романова О.В.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2023

Цель работы.

Научиться писать и применять функции, определяемые пользователем (UDF). В лабораторной работе используются база данных AdventureWorks.

Порядок выполнения.

Упражнение 0 – открыть файл InitializeData.sql и выполнить его

Use Master

```
-- Setup user for execution context
IF NOT EXISTS (SELECT * FROM sys.syslogins WHERE name = 'TULENCHI\olya-')
    CREATE LOGIN [TULENCHI\olya-]
    FROM WINDOWS
    WITH DEFAULT_DATABASE = adventure
GO
```

Use adventure

```
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = 'olya-' AND type = 'U')
    CREATE USER Adam FOR LOGIN [TULENCHI\olya-]
GO

-- Setup data
UPDATE Sales.SpecialOffer
SET      StartDate = DateAdd(day, -1, GetDate()),
        EndDate = DateAdd(day, 27, GetDate())
WHERE SpecialOfferId IN (1,3,5,7,9,11,13,15)

UPDATE Sales.SpecialOffer
SET      StartDate = DateAdd(day, 28, GetDate()),
        EndDate = DateAdd(month, 2, GetDate())
WHERE SpecialOfferId IN (2,4,6,8,10,12,14,16)
GO
```

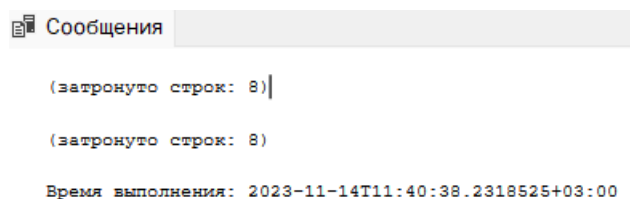


Рис. 1 – Результат выполнения упражнения №0

Упражнение 1 – Создание скалярной функции

Создать определяемую пользователем скалярную функцию Sales.GetMaximumDiscountForCategory, которая находит максимальный процент скидки (поле DiscountPct), доступный на данный момент для конкретной категории. Создан параметр @Category nvarchar(50) для

ограничения результатов на основе категории и использована функция GETDATE() для ограничения строк на основе доступности скидки на данный момент в диапазоне StartDate и EndDate.

```
IF OBJECT_ID ('Sales.GetMaximumDiscountForCategory', 'FN') IS NOT NULL
DROP FUNCTION Sales.GetMaximumDiscountForCategory
GO

CREATE FUNCTION Sales.GetMaximumDiscountForCategory (@Category nvarchar(50))
RETURNS smallmoney

BEGIN
    DECLARE @Max smallmoney
    SELECT @Max = MAX(DiscountPct)
    FROM Sales.SpecialOffer
    WHERE Category = @Category AND GETDATE() >= StartDate AND GETDATE() <= EndDate
    RETURN @Max
END
```

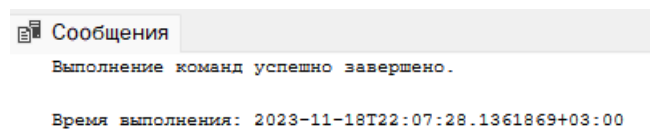


Рис. 2 – Результат выполнения упражнения №1

Для проверки было введено:

```
SELECT Sales.GetMaximumDiscountForCategory('Reseller')
```

Результаты		Сообщения
(Отсутствует имя столбца)		
1	0,35	

Рис. 3 – Результат выполнения упражнения №1

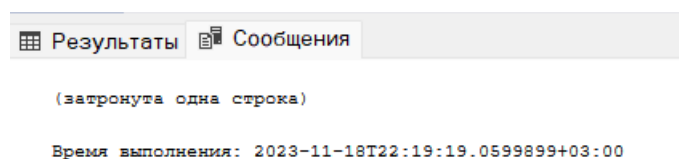


Рис. 4 – Результат выполнения упражнения №1

Упражнение 2 – Создание функции, возвращающей табличное значение (In-Line Table-valued UDF)

Создать функцию Sales.GetDiscountsForDate, которая находит те же столбцы, что и хранимая процедура GetDiscounts. У функции есть входной параметр @DateToCheck datetime. Он используется для фильтрации скидок на

основе введенной даты. Это позволяет компании Adventure Works проверить, какие скидки будут доступны на указанную дату.

```
IF OBJECT_ID ('Sales.GetDiscountForDate', 'FN') IS NOT NULL
DROP FUNCTION Sales.GetDiscountForDate
GO

CREATE FUNCTION Sales.GetDiscountForDate (@DateToCheck datetime)
RETURNS TABLE
RETURN (
    SELECT
        Description,
        DiscountPct,
        Type,
        Category,
        StartDate,
        EndDate,
        MinQty,
        MaxQty
    FROM Sales.SpecialOffer
    WHERE StartDate <= @DateToCheck AND EndDate >= @DateToCheck
    ORDER BY StartDate, EndDate OFFSET 0 ROW
)
```

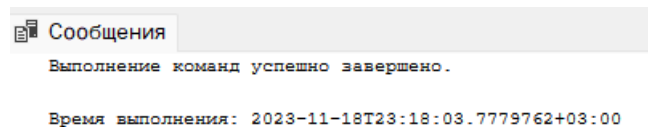


Рис. 5 – Результат выполнения упражнения №2

Для проверки было введено:

```
SELECT *
FROM Sales.GetDiscountForDate(GetDate())
ORDER BY DiscountPct DESC
```

Результаты		Сообщения						
	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	Half-Price Pedal Sale	0,50	Seasonal Discount	Customer	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	0	NULL
2	Mountain-100 Clearance Sale	0,35	Discontinued Product	Reseller	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	0	NULL
3	Road-650 Overstock	0,30	Excess Inventory	Reseller	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	0	NULL
4	Sport Helmet Discount-2003	0,15	Seasonal Discount	Reseller	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	0	NULL
5	Touring-3000 Promotion	0,15	New Product	Reseller	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	0	NULL
6	Volume Discount 41 to 60	0,15	Volume Discount	Reseller	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	41	60
7	Volume Discount 15 to 24	0,05	Volume Discount	Reseller	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	15	24
8	No Discount	0,00	No Discount	No Discount	2023-11-13 11:40:38.220	2023-12-11 11:40:38.220	0	NULL

Запрос успешно выполнен. TULENCHI (15.0 RTM) TULENCHI\olya- (63) adventure 00:00:00 8 строки

Рис. 6 – Результат выполнения упражнения №2

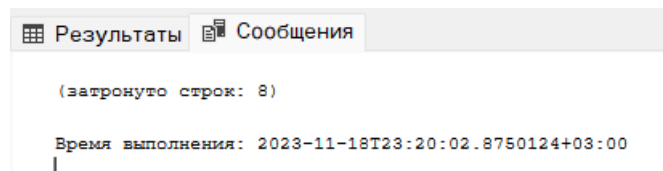


Рис. 7 – Результат выполнения упражнения №2

Упражнение 3 – Создание функции, возвращающей табличное значение (Multi-Statement Table-valued UDF)

Создать функцию GetDiscountedProducts в схеме Sales. В этой функции написан запрос для поиска продуктов, имеющих скидку. Для формирования этого запроса соединены таблицы: Sales.SpecialOfferProduct, Sales.SpecialOffer и Production.Product. Запрос выводит следующие данные: столбцы ProductID, Name, ListPrice из таблицы Production.Product, столбцы Description и DiscountPct из таблицы Sales.SpecialOffer, а также два вычисляемых столбца. Первый вычисляемый столбец получается в результате произведения значений из поля ListPrice на DiscountPct; второй – в результате вычитания из ListPrice произведения значений поля ListPrice на DiscountPct. У функции есть параметр @IncludeHistory bit, который применяется для фильтрации возвращенной таблицы на основе того, требуются ли сведения об истории скидок или необходимы только текущие сведения. Возвращаемая функцией таблица должна содержать следующее определение.

Название столбца	Тип данных
ProductID	int
Name	nvarchar(50)
ListPrice	money
DiscountDescription	nvarchar(225)
DiscountPercentage	smallmoney
DiscountAmount	money
DiscountedPrice	money

```
IF OBJECT_ID ('Sales.GetDiscountedProducts', 'FN') IS NOT NULL
DROP FUNCTION Sales.GetDiscountedProducts
```

GO

```
CREATE FUNCTION Sales.GetDiscountedProducts (@IncludeHistory bit)
RETURNS TABLE
AS
RETURN
    SELECT
        p.ProductID,
        p.Name,
        p.ListPrice,
        so.Description AS DiscountDescription,
        so.DiscountPct AS DiscountPercentage,
        p.ListPrice * so.DiscountPct AS DiscountAmount,
        p.ListPrice - p.ListPrice * so.DiscountPct AS DiscountedPrice
    FROM Sales.SpecialOffer so, Production.Product p
    WHERE
        (@IncludeHistory = 1)
    OR
        (GETDATE() BETWEEN so.StartDate AND so.EndDate);
```

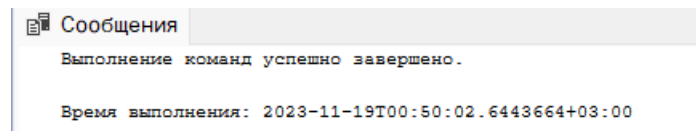


Рис. 8 – Результат выполнения упражнения №3

Для проверки:

```
SELECT * FROM Sales.GetDiscountedProducts(0)
SELECT * FROM Sales.GetDiscountedProducts(1)
```

Результаты		Сообщения					
2	2	Bearing Ball	0,00	No Discount	0,00	0,00	0,00
3	3	BB Ball Bearing	0,00	No Discount	0,00	0,00	0,00
4	4	Headset Ball Bearings	0,00	No Discount	0,00	0,00	0,00
5	316	Blade	0,00	No Discount	0,00	0,00	0,00
6	317	LL Crankarm	0,00	No Discount	0,00	0,00	0,00
7	318	ML Crankarm	0,00	No Discount	0,00	0,00	0,00
8	319	HL Crankarm	0,00	No Discount	0,00	0,00	0,00
9	320	Chainring Bolts	0,00	No Discount	0,00	0,00	0,00
10	321	Chainring Nut	0,00	No Discount	0,00	0,00	0,00
11	322	Chainring	0,00	No Discount	0,00	0,00	0,00
		Name	ListPrice	DiscountDescription	DiscountPercentage	DiscountAmount	DiscountedPrice
1	1	Adjustable Race	0,00	No Discount	0,00	0,00	0,00
2	2	Bearing Ball	0,00	No Discount	0,00	0,00	0,00
3	3	BB Ball Bearing	0,00	No Discount	0,00	0,00	0,00
4	4	Headset Ball Bearings	0,00	No Discount	0,00	0,00	0,00
5	316	Blade	0,00	No Discount	0,00	0,00	0,00
6	317	LL Crankarm	0,00	No Discount	0,00	0,00	0,00
7	318	ML Crankarm	0,00	No Discount	0,00	0,00	0,00
8	319	HL Crankarm	0,00	No Discount	0,00	0,00	0,00
9	320	Chainring Bolts	0,00	No Discount	0,00	0,00	0,00
10	321	Chainring Nut	0,00	No Discount	0,00	0,00	0,00
11	322	Chainring	0,00	No Discount	0,00	0,00	0,00
12	323	Crown Race	0,00	No Discount	0,00	0,00	0,00
13	324	Chain Stays	0,00	No Discount	0,00	0,00	0,00
14	325	Decal 1	0,00	No Discount	0,00	0,00	0,00
15	326	Decal 2	0,00	No Discount	0,00	0,00	0,00

Рис. 9 – Результат выполнения упражнения №3

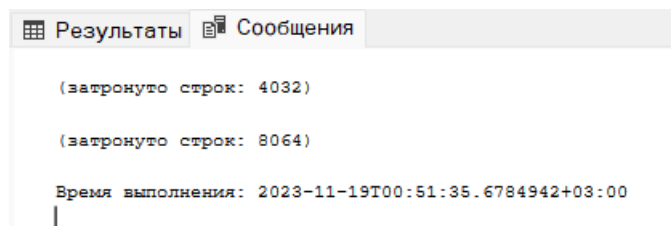


Рис. 10 – Результат выполнения упражнения №3

Выводы.

В лабораторной работе было рассмотрено создание и использование функций, определяемых пользователем. Была рассмотрена скалярная функция и функции, возвращающие табличное значение.

Список используемых материалов.

1. Распределенные базы данных: Методические указания к лабораторным работам / Сост.: А. В. Горячев, Н. Е. Новакова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 32 с
2. Документация по языку Transact-SQL URL:
<https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16>