

**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. 2311

Завьялов Н.С.

Князев И.М.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2024

*Цель работы:* научиться писать и применять функции, определяемые пользователем(UDF)

*Используемая база данных (БД):* AdventureWorks.

### Упражнение 1 – создание скалярной функции.

Запрос 1: создать определяемую пользователем скалярную функцию Sales.GetMaximumDiscountForCategory, которая находит максимальный процент скидки (поле DiscountPct) для конкретной категории. Создать параметр @Category nvarchar(50) для ограничения строк на основе доступности скидки в диапазоне StartDate и EndDate.

```
CREATE FUNCTION Sales.GetMaximumDiscountForCategory (@Category nvarchar(50))
RETURNS smallmoney
AS
BEGIN
    DECLARE @Max smallmoney

    SELECT @Max = MAX(DiscountPct)
    FROM Sales.SpecialOffer
    WHERE Category = @Category
           AND GetDate() BETWEEN StartDate AND EndDate
    GROUP BY Category

    IF (@Max IS NULL)
        SET @Max = 0

    RETURN @Max
END
```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-10-31T14:43:51.5569260+03:00

Запрос 2: Проверка

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

Результат выполнения:

(1 row affected)

Completion time: 2024-10-31T14:45:57.9996424+03:00

Results		Messages	
		(No column name)	
1	0,35		

## Упражнение 2 – создание функции, возвращающей табличное значение.

Запрос 1: создать функцию Sales.GetDiscountsForDate, которая находит те же столбцы, что и хранимая процедура GetDiscounts. У функции должен быть входной параметр @DateToCheck datetime, для фильтрации скидок на основе введенной даты.

```
CREATE FUNCTION Sales.GetDiscountsForDate (@DateToCheck datetime)
RETURNS TABLE
AS
RETURN (
    SELECT Description,
           DiscountPct,
           Type,
           Category,
           StartDate,
           EndDate,
           MinQty,
           MaxQty
    FROM Sales.SpecialOffer
    WHERE @DateToCheck BETWEEN StartDate AND EndDate
)
```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-11-01T17:13:18.9568255+03:00

### Запрос 2: Проверка

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

Результат выполнения:

(1 row affected)

Completion time: 2024-10-31T14:45:57.9996424+03:00

Results		Messages						
	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	Half-Price Pedal Sale	0.50	Seasonal Discount	Customer	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	0	NULL
2	Mountain-100 Clearance Sale	0.35	Discontinued Product	Reseller	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	0	NULL
3	Road-650 Overstock	0.30	Excess Inventory	Reseller	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	0	NULL
4	Sport Helmet Discount-2003	0.15	Seasonal Discount	Reseller	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	0	NULL
5	Touring-3000 Promotion	0.15	New Product	Reseller	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	0	NULL
6	Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	41	60
7	Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	15	24
8	No Discount	0.00	No Discount	No Discount	2024-10-30 14:35:52.880	2024-11-27 14:35:52.880	0	NULL

## Упражнение 3 – создание функции, возвращающей табличное значение.

Запрос 1: создать функцию Sales.GetDiscountedProducts. В этой функции должен быть написан запрос для поиска продуктов, имеющих скидку. Запрос должен выводить следующие данные: столбцы ProductID, Name, ListPrice из таблицы Product.Product, столбцы Description и DiscountPct из таблицы Sales.SpecialOffer, а также два вычисляемых столбца. У функции должен быть параметр @IncludeHistory bit для фильтрации возвращаемой таблицы на основе того, требуются ли сведения об истории скидок или необходимы только текущие сведения.

```
CREATE FUNCTION Sales.GetDiscountedProducts
    (@IncludeHistory bit)
RETURNS @tbl_products TABLE
    (ProductID int,
     Name nvarchar(50),
     ListPrice money,
     DiscountDescription nvarchar(255),
     DiscountPercentage smallmoney,
     DiscountAmount money,
     DiscountedPrice money)
AS
BEGIN
    INSERT @tbl_products
    SELECT
        P.ProductID,
        P.Name,
        P.ListPrice,
        SO.Description,
        SO.DiscountPct,
        P.ListPrice * SO.DiscountPct,
        P.ListPrice - P.ListPrice * SO.DiscountPct
    FROM Sales.SpecialOfferProduct SOP INNER JOIN
        Sales.SpecialOffer SO ON SOP.SpecialOfferID =
        SO.SpecialOfferID INNER JOIN
        Production.Product P ON SOP.ProductID = P.ProductID
    WHERE (SO.DiscountPct > 0) AND GetDate() BETWEEN StartDate AND EndDate
    ORDER BY ProductID

    RETURN
END
```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-11-01T19:16:28.4924630+03:00

Запрос 2: Проверка

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

Результат выполнения:

(86 rows affected)

(86 rows affected)

Completion time: 2024-11-01T19:16:35.5710643+03:00

Results Messages

	ProductID	Name	ListPrice	DiscountDescription	DiscountPercentage	DiscountAmount	DiscountedPrice
1	707	Sport-100 Helmet, Red	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
2	707	Sport-100 Helmet, Red	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
3	708	Sport-100 Helmet, Black	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
4	708	Sport-100 Helmet, Black	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
5	709	Mountain Bike Socks, M	9,50	Volume Discount 15 to 24	0,05	0,475	9,025
6	711	Sport-100 Helmet, Blue	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
7	711	Sport-100 Helmet, Blue	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
8	712	AWC Logo Cap	8,99	Volume Discount 15 to 24	0,05	0,4495	8,5405
9	714	Long-Sleeve Logo Jersey, M	49,99	Volume Discount 15 to 24	0,05	2,4995	47,4905

	ProductID	Name	ListPrice	DiscountDescription	DiscountPercentage	DiscountAmount	DiscountedPrice
1	707	Sport-100 Helmet, Red	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
2	707	Sport-100 Helmet, Red	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
3	708	Sport-100 Helmet, Black	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
4	708	Sport-100 Helmet, Black	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
5	709	Mountain Bike Socks, M	9,50	Volume Discount 15 to 24	0,05	0,475	9,025
6	711	Sport-100 Helmet, Blue	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
7	711	Sport-100 Helmet, Blue	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
8	712	AWC Logo Cap	8,99	Volume Discount 15 to 24	0,05	0,4495	8,5405
9	714	Long-Sleeve Logo Jer...	49,99	Volume Discount 15 to 24	0,05	2,4995	47,4905

Вывод

В данной лабораторной работе были получены навыки написания и применения функций определяемых пользователем(UDF).

Функции определяемые пользователем являются подпрограммами, которые принимают параметры, выполняют действие и возвращают результат этого действия в качестве значения. Возвращаемое значение может быть либо единичным скалярным значением, либо результирующим набором.

- Преимущества использования функций определяемых пользователем:
- Модульное программирование. Можно, однажды создав функцию, сохранить её в базе данных, а затем любое число раз вызывать из своей программы.
  - Быстрое выполнение. Определяемые пользователем функции снижают стоимость компиляции кода путем кэширования планов и повторного их использования для повторяющихся выполнений.

- Уменьшение сетевого трафика. Операция, которая фильтрует данные на основе некоторых сложных ограничений, которые нельзя выразить в одном скалярном выражении, можно выразить как функцию.

В ходе выполнения лабораторной работы была создана скалярная функция, функция возвращающая одно значение типа данных, заданного в предложении RETURN. Также были созданы функции с табличным значением, функции возвращающие тип данных таблицы.