

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. 1335

Максимов Ю.Е.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2024

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

Упражнение 1:

CREATE FUNCTION

Sales.GetMaximumDiscountsForCategory(@Category **NVARCHAR**(50))

RETURNS SMALLINT

AS

BEGIN

DECLARE @DiscountPct **SMALLINT**

SET @DiscountPct = (**SELECT** **MAX**(DiscountPct)

FROM Sales.SpecialOffer

WHERE Category = @Category

AND EndDate < **GETDATE**())

RETURN @DiscountPct

END

SELECT Sales.GetMaximumDiscountsForCategory('Reseller')

Результаты		Сообщения	
	(Отсутствует имя столбца)		
1	35		

Completion time: 2024-11-23T11:00:23.3099535+03:00

Упражнение 2:

CREATE FUNCTION Sales.GetDiscountsForDate (@DateToCheck **DATETIME**)

RETURNS @tableResult **TABLE**

```

(
    Description      NVARCHAR(255),
    DiscountPct      SMALLMONEY,
    Type             NVARCHAR(50),
    Category         NVARCHAR(50),
    StartDate        DATETIME,
    EndDate           DATETIME,
    MinQty            INT,
    MaxQty            INT
)
AS
BEGIN
    IF @DateToCheck IS NULL
        SET @DateToCheck = GETDATE()

    INSERT INTO @tableResult
    SELECT  Description,
            DiscountPct,
            Type,
            Category,
            StartDate,
            EndDate,
            MinQty,
            ISNULL(MaxQty, 0)
    FROM Sales.SpecialOffer
    WHERE EndDate < @DateToCheck
    ORDER BY StartDate,
            EndDate

    RETURN
END

```

```

SELECT *
FROM Sales.GetDiscountsForDate(NULL)
ORDER BY DiscountPct DESC

```

	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	Volume Discount 11 to 15	35,00	Volume Discount	Reseller	2001-07-04 00:00:00.000	2004-06-30 00:00:00.000	11	15
2	Volume Discount 11 to 15	35,00	Volume Discount	Reseller	2001-07-04 00:00:00.000	2004-06-30 00:00:00.000	11	15
3	Volume Discount 11 to 15	35,00	Volume Discount	Reseller	2001-07-04 00:00:00.000	2004-06-30 00:00:00.000	11	15
4	Mountain Tire Sale	0,50	Excess Inventory	Customer	2003-06-15 00:00:00.000	2003-08-30 00:00:00.000	0	0
5	Half-Price Pedal Sale	0,50	Seasonal Discount	Customer	2003-08-15 00:00:00.000	2003-09-15 00:00:00.000	0	0
6	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	0
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	0
8	LL Road Frame Sale	0,35	Excess Inventory	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	0
9	Road-650 Overstock	0,30	Excess Inventory	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	0
10	Touring-1000 Promotion	0,20	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	0
11	Volume Discount over 60	0,20	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	0

Completion time: 2024-11-23T11:21:23.5436235+03:00

Упражнение 3:

```

CREATE FUNCTION Sales.GetDiscountedProducts(@IncludeHistory BIT)
RETURNS @tableResult TABLE
(
    ProductID          INT,
    Name               NVARCHAR(50),
    ListPrice          MONEY,
    DiscountDescription NVARCHAR(255),
    DiscountPercentage SMALLMONEY,
    DiscountAmount     MONEY,
    DiscountedPrice    MONEY
)
AS

```

```

BEGIN
    INSERT INTO @tableResult
    SELECT  p.ProductID
            ProductID,
            p.Name
            Name,
            p.ListPrice
            ListPrice,
            sp.Description
            DiscountDescription,
            sp.DiscountPct
            DiscountPercentage,
            p.ListPrice * sp.DiscountPct
            DiscountAmount,
            p.ListPrice - (p.ListPrice * sp.DiscountPct)
            DiscountedPrice
    FROM Production.Product p
        INNER JOIN Sales.SpecialOfferProduct sop ON sop.ProductID
= p.ProductID
        INNER JOIN Sales.SpecialOffer sp ON sp.SpecialOfferID =
sop.SpecialOfferID
    WHERE (@IncludeHistory = 1 OR sp.EndDate IS NOT NULL)
    GROUP BY p.ProductID,
            p.Name,
            p.ListPrice,
            sp.Description,
            sp.DiscountPct
    RETURN
END

```

SELECT *

FROM Sales.GetDiscountedProducts(0)

Результаты		Сообщения					
	ProductID	Name	ListPrice	DiscountDescription	DiscountPercentage	DiscountAmount	DiscountedPrice
1	680	HL Road Frame - Black, 58	1431,50	No Discount	0,00	0,00	1431,50
2	706	HL Road Frame - Red, 58	1431,50	No Discount	0,00	0,00	1431,50
3	707	Sport-100 Helmet, Red	34,99	No Discount	0,00	0,00	34,99
4	707	Sport-100 Helmet, Red	34,99	Sport Helmet Discount-2002	0,10	3,499	31,491
5	707	Sport-100 Helmet, Red	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
6	707	Sport-100 Helmet, Red	34,99	Volume Discount 11 to 14	0,02	0,6998	34,2902
7	707	Sport-100 Helmet, Red	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
8	708	Sport-100 Helmet, Black	34,99	No Discount	0,00	0,00	34,99
9	708	Sport-100 Helmet, Black	34,99	Sport Helmet Discount-2002	0,10	3,499	31,491
10	708	Sport-100 Helmet, Black	34,99	Sport Helmet Discount-2003	0,15	5,2485	29,7415
11	708	Sport-100 Helmet, Black	34,99	Volume Discount 11 to 14	0,02	0,6998	34,2902
12	708	Sport-100 Helmet, Black	34,99	Volume Discount 15 to 24	0,05	1,7495	33,2405
13	709	Mountain Bike Socks, M	9,50	No Discount	0,00	0,00	9,50
14	709	Mountain Bike Socks, M	9,50	Volume Discount 11 to 14	0,02	0,19	9,31
15	709	Mountain Bike Socks, M	9,50	Volume Discount 15 to 24	0,05	0,475	9,025
16	709	Mountain Bike Socks, M	9,50	Volume Discount 25 to 40	0,10	0,95	8,55
17	710	Mountain Bike Socks, L	9,50	No Discount	0,00	0,00	9,50

Запрос успешно выполнен.

Completion time: 2024-11-23T13:45:23.1239535+03:00

Выводы: в ходе выполнения работы были получены навыки написания и применения функций, определяемых пользователем(UDF).

Вывод:

В ходе выполнения лабораторной работы я научился создавать и применять функции, определяемые пользователем (UDF), для выполнения специфичных операций в базе данных.

В процессе работы я освоил:

- создание пользовательских функций с использованием команды CREATE FUNCTION и указанием возвращаемого значения;

- передачу параметров в функции и их использование в логике вычислений;
- вызов пользовательских функций в SQL-запросах для обработки данных;
- использование UDF для упрощения повторяющихся вычислений и улучшения читаемости запросов.

Полученные навыки позволяют расширять функциональность базы данных, создавать настраиваемые операции для обработки данных и повышать эффективность выполнения сложных задач. Это важный этап в освоении возможностей современных систем управления базами данных.