Лабораторная работа №5

7 вариант

1. Создайте scalar-valued функцию, которая будет принимать в качестве входного параметра код валюты (Sales.Currency.CurrencyCode) и возвращать последний установленный курс по отношению к USD (Sales.CurrencyRate.ToCurrencyCode).

```
∃IF OBJECT_ID (N'Sales.ufnGetLastCurrencyRateToUSD', N'FN') IS NOT NULL
    DROP FUNCTION Sales.ufnGetLastCurrencyRateToUSD;
 GO
∃CREATE FUNCTION Sales.ufnGetLastCurrencyRateToUSD(@CurrencyCode NCHAR(3))
    RETURNS MONEY AS
 BEGIN
    DECLARE @LastSetCurrencyRate DATETIME
    DECLARE @LastCurrencyRate MONEY
    SELECT @LastSetCurrencyRate = MAX(CurrencyRateDate)
    FROM Sales . CurrencyRate
        WHERE FromCurrencyCode = N'USD' AND ToCurrencyCode = @CurrencyCode
    SELECT @LastCurrencyRate = EndOfDayRate
    FROM Sales.CurrencyRate
        WHERE FromCurrencyCode = N'USD' AND ToCurrencyCode = @CurrencyCode AND CurrencyRateDate = @LastSetCurrencyRate
    RETURN @LastCurrencyRate;
 END
```

Результат создания функции:

```
■ DESKTOP-L39DT9G (SQL Server 15.0.2000.5 - DESKTOP-L39DT9G\mar ^
 ■ ■ Databases
                                                          ...
1. Создайте scalar-valued функцию, которая будет принимать в качестве входного
и возвращать последний установленный курс по отношению к USD (Sales.CurrencyRat

    ■ System Databases

   ☐IF OBJECT_ID (N'Sales.ufnGetLastCurrencyRateToUSD', N'FN') IS NOT NULL

    □ AdventureWorks2012

                                                             DROP FUNCTION Sales.ufnGetLastCurrencyRateToUSD;

    ⊞ ■ Database Diagrams

                                                          GO

    ■ Tables

     ⊞ ■ Views
                                                          CREATE FUNCTION Sales.ufnGetLastCurrencyRateToUSD(@CurrencyCode NCHAR(3))

    ■ Synonyms
                                                          BEGIN
                                                             DECLARE @LastSetCurrencyRate DATETIME
DECLARE @LastCurrencyRate MONEY

☐ Programmability

    ■ Stored Procedures

       SELECT @LastSetCurrencyRate = MAX(CurrencyRateDate)
         FROM Sales CurrencyRate
         WHERE FromCurrencyCode = N'USD' AND ToCurrencyCode = @CurrencyCode
          SELECT @LastCurrencyRate = EndOfDayRate
           FROM Sales.CurrencyRate
WHERE FromCurrencyCode = N'USD' AND ToCurrencyCode = @CurrencyCode AND
           RETURN @LastCurrencyRate;
           -- Проверка работоспособности функции
           🖽 🔎 dbo.ufnLeadingZeros
                                                       ™ Messages
         Aggregate Functions
                                                          ompletion time: 2020-11-01714:03:01.8379501+03:00
```

Проверка работоспособности функции.

Значение, которое было возвращено функцией:

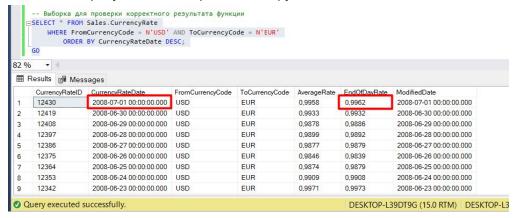
```
-- Проверка работоспособности функции
PRINT(Sales.ufnGetLastCurrencyRateToUSD(N'EUR'));
GO

82 % ▼

■ Messages
1.00

Completion time: 2020-11-01714:04:31.6102005+03:00
```

Сделаем выборку значений из самой таблицы и упорядочим ее по дате для того, чтобы сопоставить результат с возвращенным функцией выше:



Как можно увидеть со скриншотов, значение возвращаемое функцией корректно.

2. Создайте inline table-valued функцию, которая будет принимать в качестве входного параметра id продукта (Production.Product.ProductID), а возвращать детали заказа на покупку данного продукта из Purchasing.PurchaseOrderDetail, где количество заказанных позиций более 1000 (OrderQty).

```
☐ IF OBJECT_ID (N'Purchasing.ufn_GetOrderDetails', N'IF') IS NOT NULL

DROP FUNCTION Purchasing.ufn_GetOrderDetails;

GO

☐ CREATE FUNCTION Purchasing.ufn_GetOrderDetails(@ProductID INT)

RETURNS TABLE

AS

RETURN

(

SELECT * FROM Purchasing.PurchaseOrderDetail

WHERE ProductID = @ProductID AND OrderQty > 1000

]

GO
```

Результат создания функции:

```
    ■ AdventureWorks2012

                                                             -- Выборка для проверки корректного результата функции

SELECT * FROM Sales.CurrencyRate
 Database Diagrams
 III Tables
                                                                WHERE FromCurrencyCode = N'USD' AND ToCurrencyCode = N'EUR'
                                                                    ORDER BY CurrencyRateDate DESC;
 H Wiews
                                                             GO
 ■ Programmability
                                                             2. Создайте inline table-valued функцию, которая будет принимать в каче
                                                             а возвращать детали заказа на покупку данного продукта из Purchasing.Pu

☐ ■ Functions

                                                            ☐IF OBJECT_ID (N'Purchasing.ufn_GetOrderDetails', N'IF') IS NOT NULL

☐ ■ Table-valued Functions

                                                                DROP FUNCTION Purchasing.ufn_GetOrderDetails;

→ 脚 Purchasing.ufn_GetOrderDetails

                                                             CREATE FUNCTION Purchasing.ufn_GetOrderDetails(@ProductID INT)
      RETURNS TABLE

    System Functions

                                                             RETURN
    Database Triggers
                                                                 SELECT * FROM Purchasing.PurchaseOrderDetail
    WHERE ProductID = @ProductID AND OrderQty > 1000
   ⊞ ■ Rules
                                                             GO
   82 %

    ■ Plan Guides

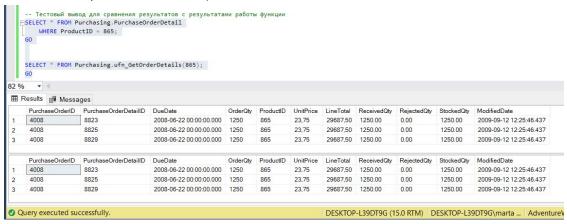
                                                          Messages
   Commands completed successfully.

    Service Broker

                                                            Completion time: 2020-11-01714:28:55.4823029+03:00
```

Проверим работоспособность функции (результат работы функции идентичен с результатом аналогичного запроса, **ProductID** равный 865 был подобран на основе

анализа содержимого таблицы):



3. Вызовите функцию для каждого продукта, применив оператор CROSS APPLY. Вызовите функцию для каждого продукта, применив оператор OUTER APPLY.

```
Вызов функции для каждого продукта с применением оператора CROSS APPLY:
 -- CROSS APPLY
SELECT
```

```
Product . Name ,
   PurchaseOrderID.
   OrderQty
FROM Production.Product Product
    CROSS APPLY Purchasing.ufn_GetOrderDetails(Product.ProductID);
```

Query executed successfully.

Product.ProductID,

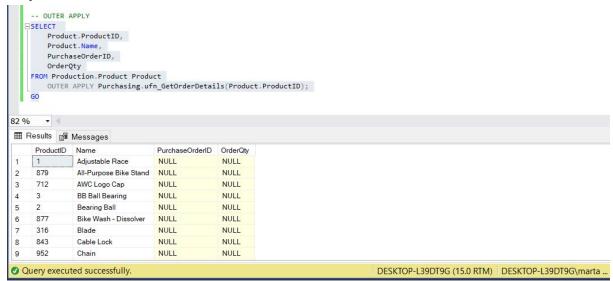
```
Результат вызова:
    -- CROSS APPLY
   SELECT
       Product.ProductID,
       Product.Name
       PurchaseOrderID,
       OrderQty
    FROM Production.Product Product
      CROSS APPLY Purchasing.ufn_GetOrderDetails(Product.ProductID);
    GO
82 %
 Classic Vest, L 4008
 2
            Classic Vest, M 4008
 3
                            1250
          Classic Vest, M 4008
     865
         Classic Vest, M 4008
Classic Vest, S 4008
                            1250
1250
     865
 5
    864
 6
         Decal 1
     325
                      70
                                   1250
 8
     325
            Decal 1
                       149
                                   1250
                  228
         Decal 1
     325
                                   1250
```

DESKTOP-L39DT9G (15.0 RTM) | DESKTOP-L39DT9G\marta

Вызов функции для каждого продукта с применением оператора **OUTER APPLY**:

```
-- OUTER APPLY
⊟SELECT
     Product ProductID.
     Product. Name,
    PurchaseOrderID.
     OrderQty
 FROM Production. Product Product
     OUTER APPLY Purchasing.ufn_GetOrderDetails(Product.ProductID);
 GO
```

Результат вызова:



4. Измените созданную inline table-valued функцию, сделав ее multistatement table-valued (предварительно сохранив для проверки код создания inline table-valued функции).

```
□IF OBJECT_ID (N'Purchasing.ufn_GetMultiOrderDetails') IS NOT NULL
    DROP FUNCTION Purchasing.ufn_GetMultiOrderDetails;
□CREATE FUNCTION Purchasing.ufn_GetMultiOrderDetails(@ProductID INT)
 RETURNS @OrderDetailTable TABLE(
    PurchaseOrderID INT,
     PurchaseOrderDetailID INT,
    DueDate DATETIME,
    OrderQty SMALLINT,
     ProductID INT,
    UnitPrice MONEY,
    LineTotal MONEY,
     RecievedQty DECIMAL(8, 2),
     RejectedQty DECIMAL(8, 2),
    StockedQty DECIMAL(9, 2),
    ModifiedDate DATETIME
 AS
 BEGIN
     INSERT INTO @OrderDetailTable
         SELECT
             PurchaseOrderID,
             PurchaseOrderDetailID,
             DueDate.
             OrderQty,
             ProductID,
             UnitPrice,
             LineTotal,
             ReceivedQty,
             RejectedQty,
             StockedQty,
             ModifiedDate
         FROM Purchasing.PurchaseOrderDetail
             WHERE ProductID = @ProductID AND OrderQty > 1000;
         RETURN
 END
 GO
```

Результат создания функции:

```
DROP FUNCTION Purchasing.ufn_GetMultiOrderDetails;
■ B DESKTOP-L39DT9G (SQL Server 15.0.2000.5 - DESKTOP-L39DT9G\mar ^
  □ ■ Databases
     CREATE FUNCTION Purchasing.ufn_GetMultiOrderDetails(@ProductID INT)

    ■ Database Snapshots

                                                                            RETURNS @OrderDetailTable TABLE(

    ■ AdventureWorks2012

                                                                                PurchaseOrderID INT,
                                                                                PurchaseOrderDetailID INT,

    ⊞ ■ Database Diagrams

                                                                                DueDate DATETIME,
       OrderQty SMALLINT,

    ■ Views
                                                                                ProductID INT.
                                                                                UnitPrice MONEY,

    ■ External Resources

                                                                                LineTotal MONEY

    ■ Synonyms
                                                                                RecievedQty DECIMAL(8, 2),
RejectedQty DECIMAL(8, 2),
StockedQty DECIMAL(9, 2),
       ■ Programmability
          ModifiedDate DATETIME
          □ ■ Functions

    □ ■ Table-valued Functions

               BEGIN

    ⊞ Purchasing.ufn_GetMultiOrderDetails

                                                                                INSERT INTO @OrderDetailTable
                                                                                    SELECT

    ⊞ Purchasing.ufn_GetOrderDetails

                                                                                        PurchaseOrderID,
            PurchaseOrderDetailID,

    ⊞   Aggregate Functions

                                                                                        DueDate.
                                                                                        OrderQty,
            ProductID,
UnitPrice,
          🖽 🖷 Database Triggers
          LineTotal,
         ReceivedQty,
                                                                                        RejectedQty,
         ⊞ ■ Rules
                                                                                        StockedQty,

    ■ Defaults

                                                                                        ModifiedDate
                                                                                    FROM Purchasing.PurchaseOrderDetail
WHERE ProductID = @ProductID AND OrderQty > 1000;
RETURN
          🖽 📁 Plan Guides

    ■ Sequences

    ■ Service Broker

    ■ Storage
                                                                            GO

    ⊞ Security

                                                                        82 %

    ■ MARTA_GULIDA

■ Security

    Messages

    ■ Replication
```

Проверим работоспособность функции:

