

Отчет по 5 лабораторной работе

```
/*
    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).
*/

CREATE FUNCTION Sales.getLatestUSDRate (@currencyCode NCHAR(3))
RETURNS MONEY AS
BEGIN
    DECLARE @latestUSDRate MONEY

    SELECT TOP 1 @latestUSDRate = EndOfDayRate FROM Sales.CurrencyRate
    WHERE ToCurrencyCode = @currencyCode AND FromCurrencyCode = 'USD'
    ORDER BY CurrencyRateDate DESC

    RETURN @latestUSDRate
END
GO
```

Рисунок 1- задание 1 и скрипт

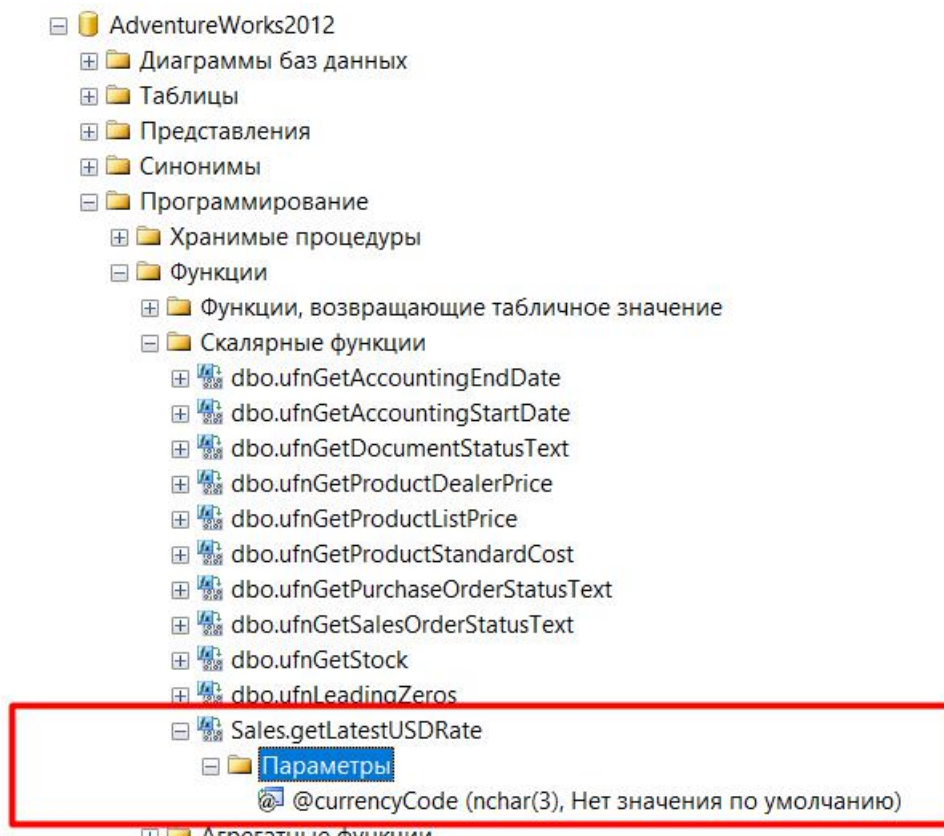


Рисунок 2 - результат выполнения скрипта

```

/*
    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).
*/

CREATE FUNCTION
    Purchasing.getOrderDetails(@productId INT)
RETURNS
    TABLE
AS
RETURN
    (SELECT
        *
    FROM
        Purchasing.PurchaseOrderDetail
    WHERE
        ProductID = @productId
        AND OrderQty > 1000)
GO

```

Рисунок 3- задание 2 и скрипт

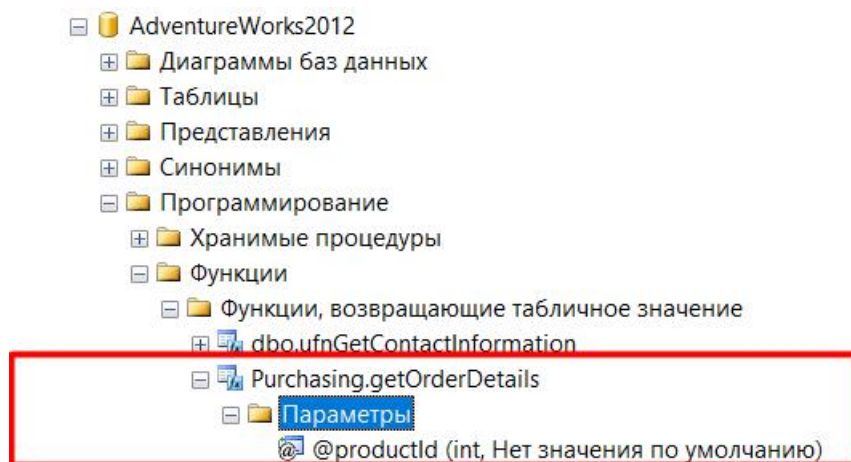


Рисунок 4 - результат выполнения скрипта

```

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

SELECT * FROM Production.Product
CROSS APPLY Purchasing.getOrderDetails(Product.ProductID)
GO

SELECT * FROM Production.Product
OUTER APPLY Purchasing.getOrderDetails(Product.ProductID)
GO

```

Рисунок 5- задание 3 и скрипт

100 %

Результаты Сообщения

	Product...	Name	ProductNum...	MakeFI...	FinishedGoodsF...	Color	SafetyStockLe...	ReorderP...	StandardC
19	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00
20	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00
21	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00
22	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00
23	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00
24	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00
25	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00
26	325	Decal 1	DC-8732	0	0	NULL	1000	750	0,00

	Product...	Name	ProductNum...	MakeFI...	FinishedGoodsF...	Color	SafetyStockLe...	ReorderP...	Si
1	1	Adjustable Race	AR-5381	0	0	NULL	1000	750	0
2	2	Bearing Ball	BA-8327	0	0	NULL	1000	750	0
3	3	BB Ball Bearing	BE-2349	1	0	NULL	800	600	0
4	4	Headset Ball ...	BE-2908	0	0	NULL	800	600	0
5	316	Blade	BL-2036	1	0	NULL	800	600	0
6	317	LL Crankarm	CA-5965	0	0	Black	500	375	0

Запрос ус... | DESKTOP-4H8UT7T\SQLEXPRESS ... | DESKTOP-4H8UT7T\saks2 ... | AdventureWorks2012 | 00:00:00 | 726 строк

Рисунок 6 - результат выполнения скрипта

```

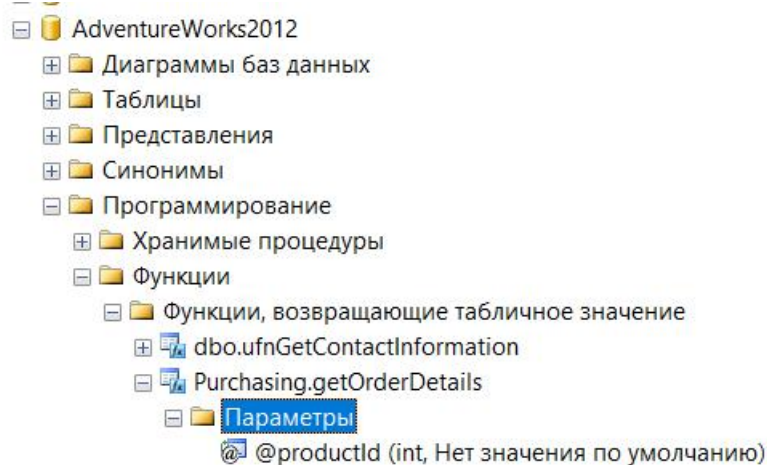
--/*
    Измените созданную inline table-valued функцию, сделав ее multistatement table-valued
    (предварительно сохранив для проверки код создания inline table-valued функции).
*/

--DROP FUNCTION
    Purchasing.getOrderDetails
GO

--CREATE FUNCTION
    Purchasing.getOrderDetails(@ProductId INT)
RETURNS
    @OrderDetails TABLE
(
    PurchaseOrderID INT,
    PurchaseOrderDetailID INT,
    DueDate DATETIME,
    OrderQty SMALLINT,
    ProductID INT,
    UnitPrice MONEY,
    LineTotal MONEY,
    ReceivedQty DECIMAL(8,2),
    RejectedQty DECIMAL(8,2),
    StockedQty DECIMAL(9,2),
    ModifiedDate DATETIME
)
AS
BEGIN
    INSERT INTO
        @OrderDetails
    (
        PurchaseOrderID,
        PurchaseOrderDetailID,
        DueDate,
        OrderQty,
        ProductID,
        UnitPrice,
        LineTotal,
        ReceivedQty,
        RejectedQty,
        StockedQty,
        ModifiedDate
    )
    SELECT
        PurchaseOrderID,
        PurchaseOrderDetailID,
        DueDate,
        OrderQty,
        ProductID,
        UnitPrice,
        LineTotal,
        ReceivedQty,
        RejectedQty,
        StockedQty,
        ModifiedDate
    FROM
        Purchasing.PurchaseOrderDetail
    WHERE
        ProductID = @ProductId
        AND OrderQty > 1000
    RETURN
END
GO

```

Рисунок 7 - задание 4 и скрипт



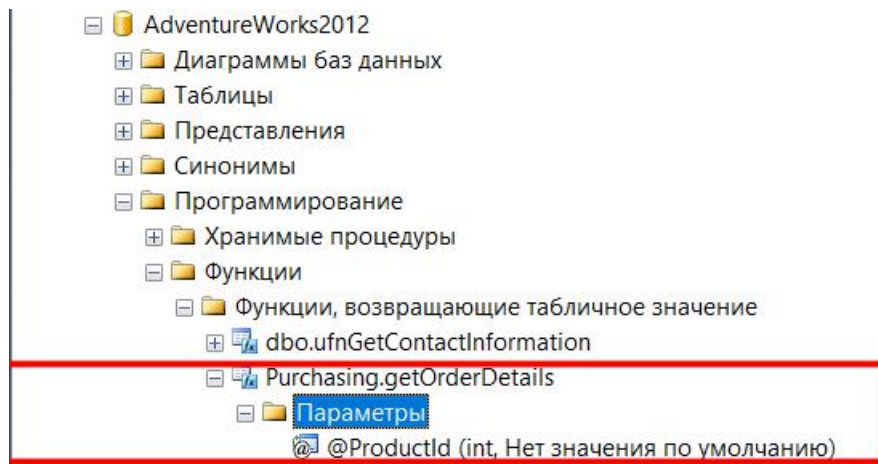


Рисунок 8-9 - результат выполнения скрипта(до и после)