

Лабораторная работа №5

Задание 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).

```
create function Sales.LastCurrencyRateUsd (@currencyCode nchar(3))
returns money as
begin
    declare @lastDate datetime
    declare @lastCurrencyRate money

    select @lastDate = 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 = @lastDate;

    return @lastCurrencyRate;
end
go

print(Sales.LastCurrencyRateUsd(N'CAD'));
go
```

100 %
Messages
1.58

```
select
    CurrencyRateID,
    CurrencyRateDate,
    FromCurrencyCode,
    ToCurrencyCode,
    EndOfDayRate
from Sales.CurrencyRate
where FromCurrencyCode = N'USD'
    and ToCurrencyCode = N'CAD'
order by CurrencyRateDate desc;
go
```

100 %

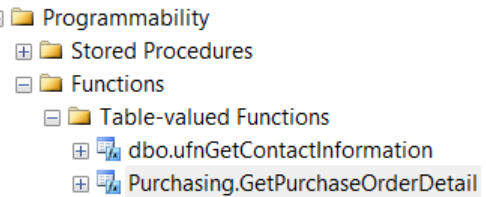
Results

Messages

	CurrencyRate...	CurrencyRateDate	FromCurrencyCo...	ToCurrencyCo...	EndOfDayR...
1	12428	2008-07-01 00:00:00.000	USD	CAD	1.5758
2	12417	2008-06-30 00:00:00.000	USD	CAD	1.5746
3	12406	2008-06-29 00:00:00.000	USD	CAD	1.5784
4	12395	2008-06-28 00:00:00.000	USD	CAD	1.5699
5	12384	2008-06-27 00:00:00.000	USD	CAD	1.5646
6	12373	2008-06-26 00:00:00.000	USD	CAD	1.5645
7	12362	2008-06-25 00:00:00.000	USD	CAD	1.565
8	12351	2008-06-24 00:00:00.000	USD	CAD	1.5567
9	12340	2008-06-23 00:00:00.000	USD	CAD	1.5549
10	12329	2008-06-22 00:00:00.000	USD	CAD	1.5561

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.GetPurchaseOrderDetail(@productID int)
returns table
as
return
select *
from Purchasing.PurchaseOrderDetail
where ProductID = @productID
and OrderQty > 1000;
go
```



```
select
count(PurchaseOrderID)
from Purchasing.GetPurchaseOrderDetail(325);
go
```

Results	Messages
(No column na...)	
1	50

```
select
count(PurchaseOrderID)
from Purchasing.PurchaseOrderDetail
where ProductID = 325 and OrderQty > 1000;
go
```

Results	Messages
(No column na...)	
1	50

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

```
select
Product.ProductID,
Product.Name,
PurchaseOrderID,
PurchaseOrderDetailID,
OrderQty
from Production.Product
cross apply Purchasing.GetPurchaseOrderDetail(ProductID);
go
```

100 %

Results Messages

	Product...	Name	PurchaseOrder...	PurchaseOrderData...	Order...
1	866	Classic Vest, L	4008	8824	1250
2	866	Classic Vest, L	4008	8826	1250
3	865	Classic Vest, M	4008	8823	1250
4	865	Classic Vest, M	4008	8825	1250
5	865	Classic Vest, M	4008	8829	1250
6	864	Classic Vest, S	4008	8822	1250
7	325	Decal 1	70	159	1250
8	325	Decal 1	149	358	1250
9	325	Decal 1	228	543	1250
10	325	Decal 1	307	699	1250

```
select
Product.ProductID,
Product.Name,
PurchaseOrderID,
PurchaseOrderDetailID,
OrderQty
from Production.Product
outer apply Purchasing.GetPurchaseOrderDetail(ProductID);
go
```

00 %

Results Messages

	Product...	Name	PurchaseOrde...	PurchaseOrderDeta...	Order...
1	1	Adjustable Race	NULL	NULL	NULL
2	879	All-Purpose Bike Stand	NULL	NULL	NULL
3	712	AWC Logo Cap	NULL	NULL	NULL
4	3	BB Ball Bearing	NULL	NULL	NULL
5	2	Bearing Ball	NULL	NULL	NULL
6	877	Bike Wash - Dissolver	NULL	NULL	NULL
7	316	Blade	NULL	NULL	NULL
8	843	Cable Lock	NULL	NULL	NULL
9	952	Chain	NULL	NULL	NULL
10	324	Chain Stays	NULL	NULL	NULL

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

```

create function Purchasing.GetPurchaseOrderDetailMulti(@productID int)
returns @resultTable 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 @resultTable
    select
        PurchaseOrderID,
        PurchaseOrderDetailID,
        DueDate,
        OrderQty,
        ProductID,
        UnitPrice,
        LineTotal,
        ReceivedQty,
        RejectedQty,
        StockedQty,
        ModifiedDate
    from Purchasing.PurchaseOrderDetail
    where ProductID = @productID
        and OrderQty > 1000;
return
end
go

```

Programmability

- Stored Procedures
- Functions
 - Table-valued Functions
 - dbo.ufnGetContactInformation
 - Purchasing.GetPurchaseOrderDetail
 - Purchasing.GetPurchaseOrderDetailMulti**

```

select
    PurchaseOrderID,
    PurchaseOrderDetailID,
    ProductID,
    OrderQty
from Purchasing.GetPurchaseOrderDetailMulti(325);
go

```

100 %

	PurchaseOrder...	PurchaseOrderData...	Produc...	Order...
1	70	159	325	1250
2	149	358	325	1250
3	228	543	325	1250
4	307	699	325	1250
5	386	865	325	1250
6	495	1101	325	1250
7	578	1288	325	1250
8	665	1488	325	1250
9	744	1651	325	1250
10	827	1850	325	1250

```

select
    PurchaseOrderID,
    PurchaseOrderDetailID,
    ProductID,
    OrderQty
from Purchasing.GetPurchaseOrderDetail(325);
go

```

100 %

	PurchaseOrder...	PurchaseOrderData...	Produc...	Order...
1	70	159	325	1250
2	149	358	325	1250
3	228	543	325	1250
4	307	699	325	1250
5	386	865	325	1250
6	495	1101	325	1250
7	578	1288	325	1250
8	665	1488	325	1250
9	744	1651	325	1250
10	827	1850	325	1250