**6-7**

**1. Создать скалярную функцию, которая подсчитывает количество заказов, проданных за определенный период. Параметры – покупатель, дата начала периода, дата окончания периода.**

use ПРОДАЖИ

go

create function COUNT\_Zakazy(@p varchar(20) = null, @start date, @end date) returns int

as begin

declare @rc int = 0;

set @rc = (

select count([Номер заказа]) from ЗАКАЗЫ

where ([Дата поставки] between @start and @end)

and Заказчик = @p);

return @rc;

end;

go

declare @n1 int = dbo.COUNT\_Zakazy ('Буслiк', '2014-07-02','2014-08-02');

print 'Количество товаров: ' + cast(@n1 as varchar(4));

**2. Создать процедуру, результатом действия которой будет список заказов с выводом количества товаров. Параметр – наименование покупателя.**

--2

create procedure Zkz\_REPORT @p CHAR(50)

as

declare @rc int = 0;

declare @tv char(20), @t char(300) = ' ';

declare ZkTov CURSOR for

select [Наименование товара] from ЗАКАЗЫ where Заказчик = @p;

if not exists (select [Номер заказа]

from Заказы where Заказчик = @p)

raiserror('ошибка', 11, 1);

else

open ZkTov;

fetch ZkTov into @tv;

print 'Номера заказов';

while @@fetch\_status = 0

begin

set @t = rtrim(@tv) + ', ' + @t;

set @rc = @rc + 1;

fetch ZkTov into @tv;

end;

print @t;

close ZkTov;

return @rc;

declare @rc1 int;

exec @rc1 = Zkz\_REPORT @p = 'Буслiк';

print 'Общее количество товаров = ' + cast(@rc1 as varchar(3));

**3. Создать процедуру, которая увеличивает на 10% выпуск определенного товара. Параметр – наименование товара.**

create procedure Up @n varchar (20)

as begin

declare @rc int = 0;

DECLARE @name4 char(20), @n3 int;

DECLARE Task6a CURSOR LOCAL for

SELECT Наименование, Цена from ТОВАРЫ

where Наименование=@n

OPEN Task6a;

fetch Task6a into @name4, @n3;

UPDATE ТОВАРЫ set Цена=Цена+(Цена\*0.1) where CURRENT OF Task6a;

CLOSE Task6a;

end;

declare @rc1 int;

exec @rc1 = Up @n ='Твистер';

select \* from ТОВАРЫ

**4. Создайте процедуру, которая выводит список заказов в зависимости от параметров: все заказы, заказы по стоимости выше определенного значения, заказы по стоимости ниже определенного значения. Обработайте возможные ошибки.**

**create procedure Zkz\_REPORT @cena int**

**as**

**declare @rc int = 0;**

**begin try**

**declare @tv char(20), @t char(300) = ' ';**

**declare ZkTov CURSOR for**

**select Заказ from Заказы where Стоимость > @cena;**

**if not exists (select Заказ**

**from Заказы where Стоимость < @cena)**

**if not exists (select Заказ**

**from Заказы)**

**raiserror('ошибка', 11, 1);**

**else**

**open ZkTov;**

**fetch ZkTov into @tv;**

**print 'Заказанные товары';**

**while @@fetch\_status = 0**

**begin**

**set @t = rtrim(@tv) + ', ' + @t;**

**set @rc = @rc + 1;**

**fetch ZkTov into @tv;**

**end;**

**print @t;**

**close ZkTov;**

**return @rc;**

**end try**

**begin catch**

**print 'ошибка в параметрах'**

**if error\_procedure() is not null**

**print 'имя процедуры : ' + error\_procedure();**

**return @rc;**

**end catch;**

**declare @rc int;**

**exec @rc = Zkz\_REPORT @сena = 3000;**

**print 'количество заказов = ' + cast(@rc as varchar(3));**

**5. Создайте функцию, которая выводит список и вычисляет количество заказов, выполненных в определенном году для определенного покупателя. Параметры – покупатель, год.**

create function FCTEACHER(@p varchar(20), @d date) returns int

as begin

declare @rc int = (select count(\*) from ЗАКАЗЫ where Заказчик = ISNULL(@p, Заказчик));

return @rc;

end;

go

select Заказчик,dbo.FCTEACHER(Заказчик)

[Количество товаров] from ЗАКАЗЫ

**6. Создайте процедуру, которая выводит список товаров, которые не продавались в определенном году. Параметр – год. Обработайте возможные ошибки.**

**7. Создайте процедуру, которая выводит список заказов и их среднюю стоимость для определенного покупателя. Параметр – наименование покупателя. Обработайте возможные ошибки.**

Пример триггера **TRIG\_Tov**,которыйзаписывает данные в таблицу **TR\_Tov** для каждой изменяемой строки таблицы **Товары**:

**create trigger TRIG\_Tov on Товары after INSERT, DELETE, UPDATE**

**as declare @a1 varchar(20), @a2 real, @a3 int, @in varchar(300);**

**declare @ins int = (select count(\*) from inserted),**

**@del int = (select count(\*) from deleted);**

**if @ins > 0 and @del = 0**

**begin**

**print 'Событие: INSERT';**

**set @a1 = (select [Наименование] from INSERTED);**

**set @a2 = (select [Цена] from INSERTED);**

**set @a3 = (select [Количество] from INSERTED);**

**set @in = @a1+' '+cast(@a2 as varchar(20))+' '+cast(@a3 as varchar(20));**

**insert into TR\_Tov(ST, TRN, C) values('INS', 'TRIG\_Tov', @in);**

**end;**

**else**

**if @ins = 0 and @del > 0**

**begin**

**print 'Событие: DELETE';**

**set @a1 = (select [Наименование] from deleted);**

**set @a2 = (select [Цена] from deleted);**

**set @a3 = (select [Количество] from deleted);**

**set @in = @a1+' '+cast(@a2 as varchar(20))+' '+cast(@a3 as varchar(20));**

**insert into TR\_Tov(ST, TRN, C) values('DEL', 'TRIG\_Tov', @in);**

**end;**

**else**

**if @ins > 0 and @del > 0**

**begin**

**print 'Событие: UPDATE';**

**set @a1 = (select [Наименование] from inserted);**

**set @a2 = (select [Цена] from inserted);**

**set @a3 = (select [Количество] from inserted);**

**set @in = @a1+' '+cast(@a2 as varchar(20))+' '+cast(@a3 as varchar(20));**

**set @a1 = (select [Наименование] from deleted);**

**set @a2 = (select [Цена] from deleted);**

**set @a3 = (select [Количество] from deleted);**

**set @in = @a1+' '+cast(@a2 as varchar(20))+' '+cast(@a3 as varchar(20))+' '+@in;**

**insert into TR\_Tov(ST, TRN, C) values('UPD', 'TRIG\_Tov', @in);**

**end;**

**return;**

DML-операторы изменения таблицы **Товары**:

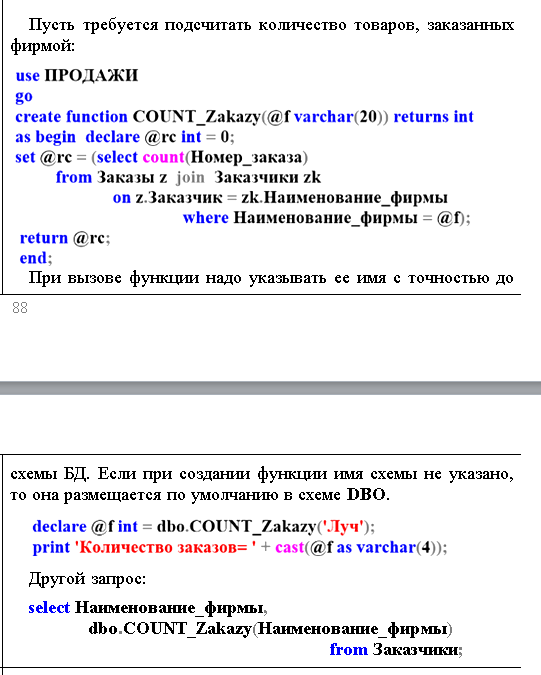
**insert into Товары(Наименование, Цена, Количество)**

**values('Стол', 140, 20);**

**delete from Товары where Наименование = 'Стол';**

**update Товары set Количество = 20 where Наименование = 'Стул';**

Оператор **select \* from TR\_Tov** позволяет проверить содержимое **TR\_Tov**

.

