## Домашнее задание 13

Добавить в БД магазин следующие функции:

- 1. Пользовательская функция возвращает количество уникальных покупателей;
- 2. Пользовательская функция возвращает среднюю цену товара конкретного вида. Вид товара передаётся в качестве параметра. Например, среднюю цену обуви;
- 3. Пользовательская функция возвращает среднюю цену продажи по каждой дате, когда осуществлялись продажи;
- 4. Пользовательская функция возвращает информацию о последнем проданном товаре. Критерий определения последнего проданного товара: дата продажи;
- 5. Пользовательская функция возвращает информацию о первом проданном товаре. Критерий определения первого проданного товара: дата продажи;
- 6. Пользовательская функция возвращает информацию о заданном виде товаров конкретного производителя. Вид товара и название производителя передаются в качестве параметров.

## Функции

1. Пользовательская функция возвращает количество уникальных покупателей

```
CREATE function Uni_CL()
RETURNS int
AS BEGIN
DECLARE @tempble table (Full_Name nvarchar(100))
INSERT INTO @tempble SELECT DISTINCT c.lname + ' ' + c.name + ' '+
c.fname AS 'Full_Name'
FROM Client as c
DECLARE @uniq int
SET @uniq = (SELECT count(Full_Name) from @tempble)
RETURN @uniq
END
```

```
SQLQuery2.sql - PT...g (РТG\Павел (53))* SQLQuery1.sql - PT...g (РТG\Павел (62))* ⇒ Х
       1 ☐ CREATE function Uni_CL()
          RETURNS int
          AS BEGIN
          DECLARE @tempble table (Full_Name nvarchar(100))
INSERT INTO @tempble SELECT DISTINCT c.lname + ' ' + c.name + ' '+ c.fname AS 'Full_Name'
FROM Client as c
          DECLARE @uniq int
      8 SET @uniq = (SELECT count(Full_Name) from @tempble)
      9
          RETURN @uniq
     10
     11
     12 --print (dbo.Uni_CL ())
100 % - 4
 В Сообщения
   9
   Время выполнения: 2022-12-17T13:27:06.3026119+03:00
```

2. Пользовательская функция возвращает среднюю цену товара конкретного вида. Вид товара передаётся в качестве параметра. Например, среднюю цену обуви

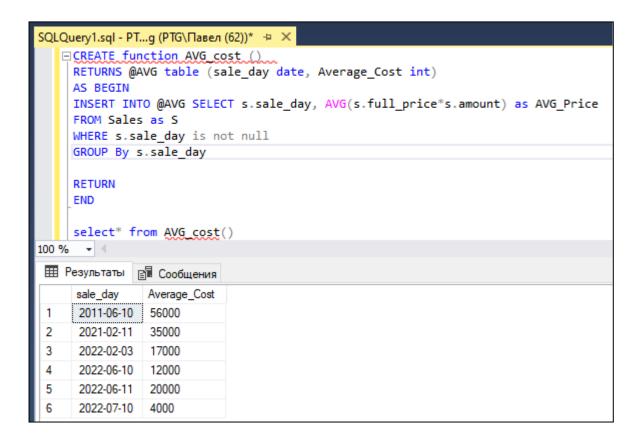
```
CREATE function Sred_price (@kind nvarchar(50))
RETURNS money
AS BEGIN
DECLARE @AVG money
SET @AVG = (select AVG(full_price) FROM Sales as S
JOIN goods as g ON g.id = s.goods_id
WHERE g.kind = @kind)
RETURN @AVG
END
```

```
SQLQuery2.sql - PT...g (РТG\Павел (53))*
                                      SQLQuery1.sql - PT...q (РТG\Павел (62))* → ×
      1 □ CREATE function Sred price (@kind nvarchar(50))
          RETURNS money
      3
          AS BEGIN
         DECLARE @AVG money
          SET @AVG = (select AVG(full_price) FROM Sales as S
          JOIN goods as g ON g.id = s.goods_id
          WHERE g.kind = @kind)
      7
      8
          RETURN @AVG
      9
          END
     10
          print (dbo.Sred_price ('Prazdnik'))
     11
     12
100 % ▼ 4
В Сообщения
   8000.00
   Время выполнения: 2022-12-17T13:22:43.1134656+03:00
```

3. Пользовательская функция возвращает среднюю цену продажи по каждой дате, когда осуществлялись продажи

```
CREATE function AVG_cost ()
RETURNS @AVG table (sale_day date, Average_Cost int)
AS BEGIN
INSERT INTO @AVG SELECT s.sale_day, AVG(s.full_price*s.amount) as
AVG_Price
FROM Sales as S
WHERE s.sale_day is not null
GROUP By s.sale_day

RETURN
END
```



4. Пользовательская функция возвращает информацию о последнем проданном товаре. Критерий определения последнего проданного товара: дата продажи

```
CREATE function Last_sale_from (@date date)
RETURNS @last table (Title nvarchar(50), Sale_Amount int, Price money, Sale_Date date)

AS BEGIN

DECLARE @s_day date
SET @s_day= (select top (1) sale_day from Sales as S
WHERE s.sale_day < @date
ORDER BY sale_day desc);

INSERT INTO @last SELECT g.title, s.amount, s.full_price,
s.sale_day
FROM sales as S
JOIN goods as g ON g.id = s.goods_id
WHERE s.sale_day = @s_day

RETURN
END
```

```
SQLQuery1.sql - PT...g (РТG\Павел (62))* □ ×
   □ CREATE function Last_sale_from (@date date)
     RETURNS @last table (Title nvarchar(50), Sale Amount int, Price money, Sale Date date)
     AS BEGIN
    DECLARE @s_day date
     SET @s_day= (select top (1) sale_day from Sales as S
     WHERE s.sale_day < @date
    ORDER BY sale_day desc);
     INSERT INTO @last SELECT g.title, s.amount, s.full_price, s.sale_day
     FROM sales as S
     JOIN goods as g ON g.id = s.goods_id
     WHERE s.sale day = @s day
     RETURN
     END
     select* from Last sale from('2022-07-10')
100 % - 4
Результаты По Сообщения
                    Sale Amount Price
                                       Sale Date
     Брюшко, гудбай 5
                               4000,00 2022-06-11
```

5. Пользовательская функция возвращает информацию о первом проданном товаре. Критерий определения первого проданного товара: дата продажи

```
CREATE function First_sale_from (@date date)
RETURNS @first table (Title nvarchar(50), Sale_Amount int, Price money, Sale_date date)

AS BEGIN

DECLARE @s_day date
SET @s_day= (select top (1) sale_day from Sales as S
WHERE s.sale_day >= @date
order by sale_day);

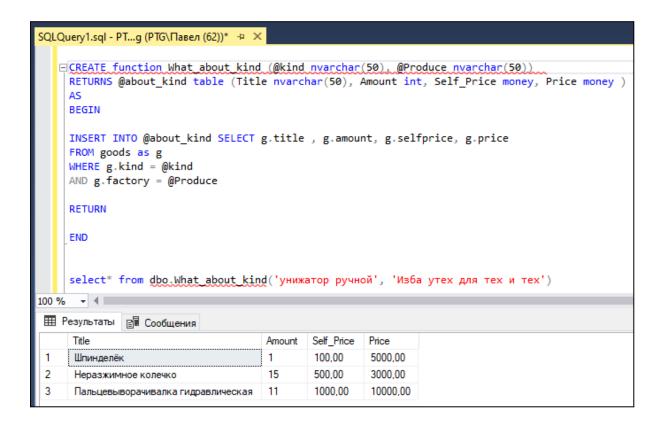
INSERT INTO @first SELECT g.title, s.amount, s.full_price,
s.sale_day
FROM sales as S
JOIN goods as g ON g.id = s.goods_id
WHERE s.sale_day = @s_day

RETURN
END
```

```
SQLQuery1.sql - РТ...g (РТG\Павел (62))* 垣 🗶
   □ CREATE function First sale from (@date date)
     RETURNS @first table (Title nvarchar(50), Sale Amount int, Price money, Sale date date)
     AS BEGIN
     DECLARE @s_day date
     SET @s day= (select top (1) sale day from Sales as S
     WHERE s.sale day >= @date
     order by sale day);
     INSERT INTO @first SELECT g.title, s.amount, s.full price, s.sale day
     FROM sales as S
     JOIN goods as g ON g.id = s.goods_id
     WHERE s.sale day = @s day
     RETURN
     END
     select* from First sale from('2022-02-04')
100 % - 4
 Результаты В Сообщения
                    Sale_Amount Price
                                       Sale_date
     Брюшко, гудбай 3
                               4000,00 2022-06-10
```

6. Пользовательская функция возвращает информацию о заданном виде товаров конкретного производителя. Вид товара и название производителя передаются в качестве параметров

```
CREATE function What_about_kind (@kind nvarchar(50), @Produce
nvarchar(50))
RETURNS @about_kind table (Title nvarchar(50), Amount int,
Self_Price money, Price money )
AS
BEGIN
INSERT INTO @about_kind SELECT g.title , g.amount, g.selfprice,
g.price
FROM goods as g
WHERE g.kind = @kind
AND g.factory = @Produce
RETURN
END
```



## База данных Sportmag

```
CREATE database Sportmag;
use sportmag;
CREATE table Goods
( id int identity(1,1)primary key,
title nvarchar(100) check(title != '')not null,
kind nvarchar(100) check(kind != '') not null,
amount int check(amount>=0) not null default(0),
selfprice money check(selfprice>=0) not null,
factory nvarchar(100) check(factory!='')not null,
price money not null
);
CREATE table Staff (
 id int identity(1,1)primary key,
Name nvarchar(30) check(Name!='') not null,
Lname nvarchar(30) check(Lname!='')not null,
Fname nvarchar(30) check(Fname!='')not null,
 post nvarchar(30) check(post!='')not null,
 first day date not null CHECK(first day < = GETDATE()),</pre>
 sex bit not null default(0),
salary money not null
);
CREATE table OldStaff (
 id int identity(1,1)primary key,
Name nvarchar(30) check(Name!='') not null,
Lname nvarchar(30) check(Lname!='') not null,
 Fname nvarchar(30) check(Fname!='')not null,
 post nvarchar(30) check(post!='')not null,
 first day date not null,
 lASt day date not null,
sex bit not null default(0)
 );
CREATE table Client (
 id int identity(1,1)primary key,
Name nvarchar(30) check(Name!='')not null,
Lname nvarchar(30) check(Lname!='')not null,
 Fname nvarchar(30) check(Fname!='')not null,
 email nvarchar(30) check(email LIKE ('%0%'))not null,
phone INT not null CHECK(phone != ''),
sex bit not null default(0),
percents DECIMAL (4,2) default(0),
```

```
subscribe bit not null default(0)
 );
 CREATE table Sales (
 id int identity(1,1)primary key,
 goods id int not null Foreign key (goods id) REFERENCES
goods (id),
 full price money not null,
 amount int check(amount>=0) not null default(0),
 sale day date not null CHECK(sale day <= GETDATE()),</pre>
 staff id int not null default(0) Foreign key (staff id)
REFERENCES staff(id) ON DELETE CASCADE ,
 client id int not null Foreign key (client id) REFERENCES
client(id)
);
 CREATE table Archive (
id int identity(1,1) primary key,
 goods id INT not null Foreign key (goods id) REFERENCES
goods(id))
Alter table goods add CONSTRAINT CHECK SALE
check(price>selfprice);
Alter table staff add CONSTRAINT CHECK SALARY check(salary>0);
Alter table OldStaff add CONSTRAINT CHECK FIRST CHECK(first day !>
GETDATE());
Alter table OldStaff add CONSTRAINT CHECK LAST CHECK(lASt day >=
first day);
INSERT INTO Goods (title, kind, amount, selfprice, factory, price)
VALUES
('Брюшко, гудбай', 'гантели', 77, 500, 'Кировский завод', 2000),
('Шпинделёк', 'унижатор ручной', 1, 100, 'Изба утех для тех и
Tex', 5000),
('Жух-жух 17', 'скребок ушной', 1000, 199, 'Когда никто не
слышит', 400),
('Klizma', 'Prazdnik', 44, 200, 'StroyKlizmMASh', 1000),
('Неразжимное колечко', 'унижатор ручной', 15, 500, 'Изба утех для
тех и тех', 3000),
('Пальцевыворачивалка гидравлическая', 'унижатор ручной', 11,
1000, 'Изба утех для тех и тех', 10000);
INSERT INTO Client ( Name, Lname, Fname, email, phone, sex,
percents,
subscribe) VALUES
('Кшиштоф', 'Мухин', 'Агафонович', 'fly@mail.com', 656254, 1,
15.00, 1),
```

```
('Гвидон', 'Кулебякин', 'Пироксимович', 'gvido@mail.com', 253545,
1, 11.00, 0),
('Don', 'Juan', 'MAximovich', 'gvieo@mail.com', 2533345, 1, 14.00,
1),
('Donna-Margarita', 'Juan', 'Nikolavna', 'gvo@mail.com', 2577345,
0, 24.00, 1),
('Ivan', 'Hlebov', 'Oristarhovich', 'hlebalo@mail.com', 233377345,
1, 21.00, 1),
('Gleb', 'Hlebov', 'Oristarhovich', 'Glebalo@mail.com', 233377344,
1, 25.00, 1);
INSERT INTO Staff (Name, Lname, Fname, post, first day, sex,
salary) VALUES
('Ганс', 'Поберушкин', 'Охламонович', 'страшный продаван',
'2014-05-05', 1, 2500),
('Клёпа', 'Фон Амурский', 'Сигизмундович', 'продажник 2 кат.',
'2016-02-02', 1, 500),
('Галя', 'Гром', 'Победитовна', 'кассир-продавец', '2000-01-01',
0, 200),
('Стивен', 'Крючковски', 'Джонович', 'консультант-инсультант',
'2018-02-01', 1, 2100),
('Баба', 'Маня', 'Филипповна', 'уборщица-продавец', '1965-01-01',
0, 200),
('Зульфия', 'Наршланбэ', 'Барабековна', 'грузчица-реализатор',
'2021-07-05', 0, 800),
('Barabek', 'Chelovek', 'Poedatel', 'crisis manager',
'2017-07-07', 1, 7300),
('Islambek', 'Chelovek', 'Poedatel', 'manager of judgement day',
'2011-01-01', 1, 17300),
('Bob', 'Hlebov', 'Bobovich', 'bean mASter', '2014-04-01', 1,
1300),
('George', 'Hlebov', 'Shprotovich', 'sandvich mASter',
'2013-04-04', 1, 1700);
INSERT INTO Sales ( goods_id, full_price, amount, sale_day,
staff id, client id) VALUES
(1, 4000, 3, '2022-06-10', 1, 1),
(1, 4000, 1, '2022-07-10', 1, 1),
(1, 4000, 5, '2022-06-11', 2, 1),
(2, 2000, 7, '2022-02-03', 3, 2),
(3, 7000, 5, '2021-02-11', 2, 2),
(1, 4000, 5, '2022-02-03', 4, 1),
(4, 8000, 7, '2011-06-10', 4, 3);
```