

## Домашнее задание 12

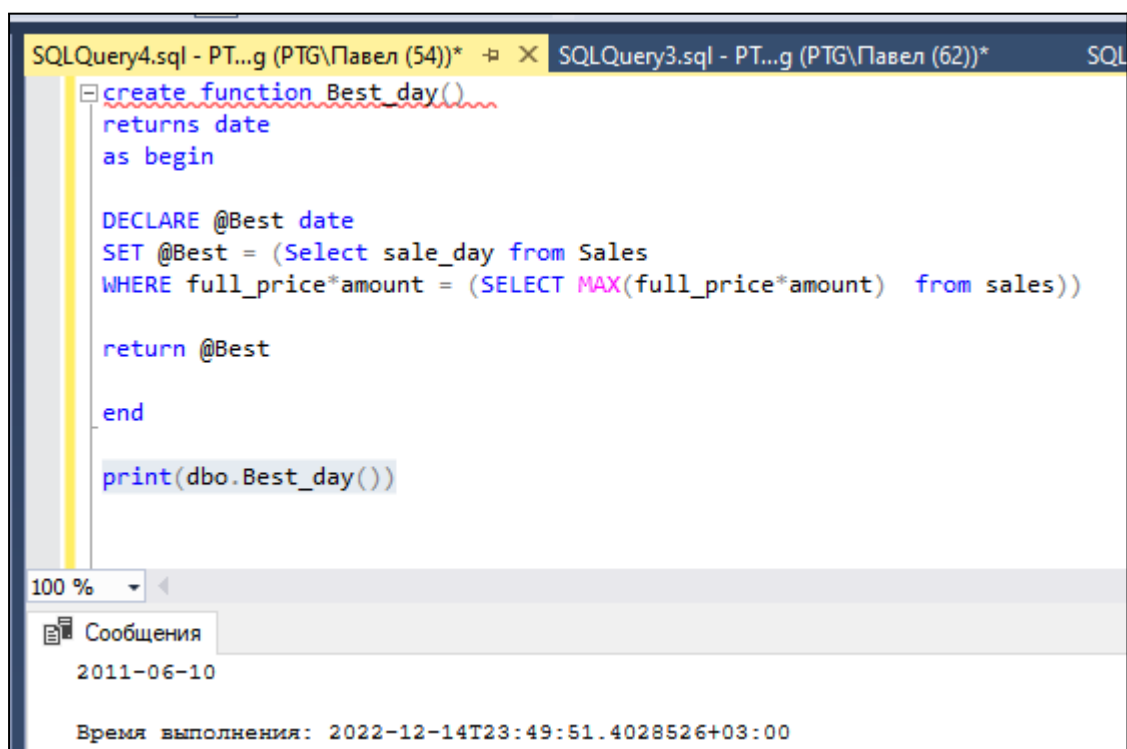
*Добавить в БД магазин следующие функции:*

- 1. Пользовательская функция возвращает дату, когда общая сумма продаж за день была максимальной;*
- 2. Пользовательская функция возвращает информацию о всех продажах заданного товара. Название товара передается в качестве параметра;*
- 3. Пользовательская функция возвращает информацию о всех продавцах однофамильцах;*
- 4. Пользовательская функция возвращает информацию о всех покупателях однофамильцах;*
- 5. Пользовательская функция возвращает информацию о всех покупателях и продавцах однофамильцах.*

## Функции

1. Пользовательская функция возвращает дату, когда общая сумма продаж за день была максимальной

```
CREATE function Best_day()  
RETURNS date  
AS BEGIN  
  
DECLARE @Best date  
SET @Best = (Select sale_day FROM Sales  
WHERE full_price*amount = (SELECT MAX(full_price*amount) FROM  
sales))  
  
RETURN @Best  
  
END
```



2. Пользовательская функция возвращает информацию о всех продажах заданного товара. Название товара передаётся в качестве параметра

```
CREATE function All_of_Saled_Thing(@T nvarchar (50))
RETURNS TABLE
AS

RETURN (Select g.title AS Thing, s.sale_day AS Day_of_sale,
s.full_price AS Cost, s.amount AS Amount,
s.full_price*s.amount AS Full_Money
FROM Sales AS S
JOIN goods AS g ON g.id = s.goods_id
WHERE g.title = @T)
```

The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery4.sql - PT...g (PTG\Павел (54))' and 'SQLQuery3.sql - PT...g (PTG\Павел (62))'. The active tab displays the following SQL code:

```
create function All_of_Saled_Thing(@T nvarchar (50))
returns TABLE
as

return (Select g.title as Thing, s.sale_day as Day_of_sale,
s.full_price as Cost, s.amount as Amount,
s.full_price*s.amount as Full_Money
from Sales as S
JOIN goods as g ON g.id = s.goods_id
WHERE g.title = @T)

SELECT* from All_of_Saled_Thing('Klizma');
```

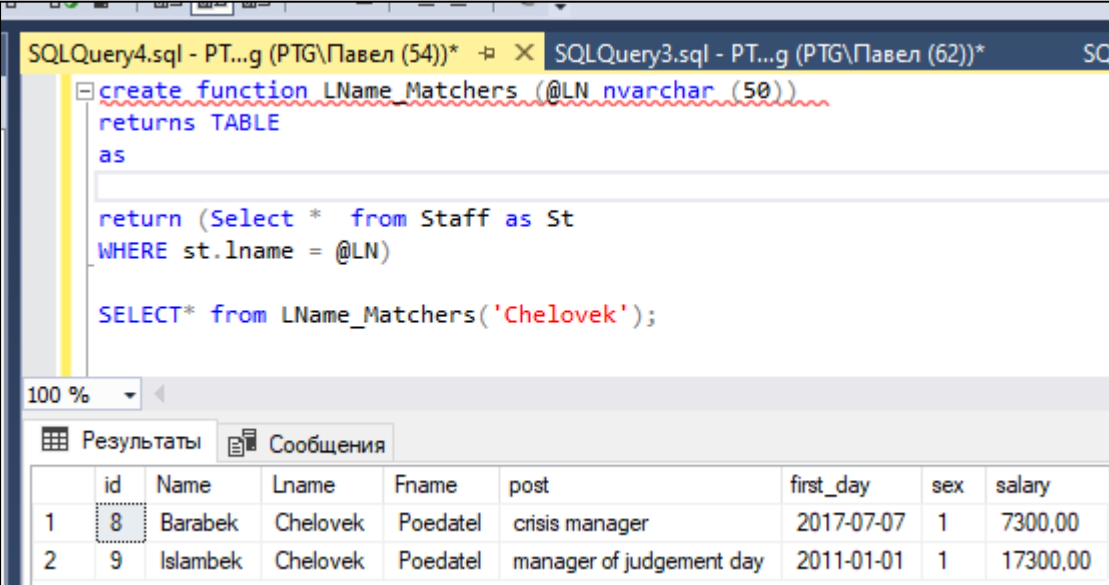
Below the code editor, the 'Results' tab is selected, showing a table with the following data:

	Thing	Day_of_sale	Cost	Amount	Full_Money
1	Klizma	2011-06-10	8000,00	7	56000,00

3. Пользовательская функция возвращает информацию о всех продавцах однофамильцах

```
CREATE function LName_Matchers1 (@LN nvarchar (50))
RETURNS TABLE
AS

RETURN (Select * FROM Staff AS St
WHERE st.lname = @LN);
```



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery4.sql - PT...g (PTG\Павел (54))\*' and 'SQLQuery3.sql - PT...g (PTG\Павел (62))\*'. The active tab displays the following SQL code:

```
create function LName_Matchers (@LN nvarchar (50))
returns TABLE
as

return (Select * from Staff as St
WHERE st.lname = @LN)

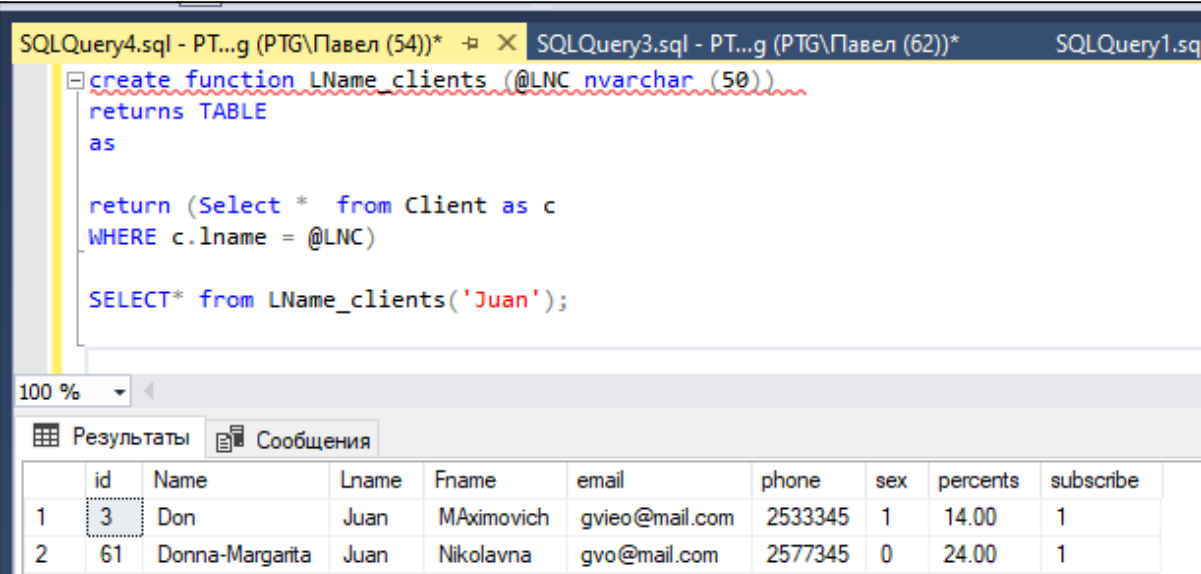
SELECT* from LName_Matchers('Chelovek');
```

Below the code editor, the 'Results' tab is selected, showing a table with 9 columns: id, Name, Lname, Fname, post, first\_day, sex, and salary. The table contains two rows of data.

	id	Name	Lname	Fname	post	first_day	sex	salary
1	8	Barabek	Chelovek	Poedatel	crisis manager	2017-07-07	1	7300,00
2	9	Islambek	Chelovek	Poedatel	manager of judgement day	2011-01-01	1	17300,00

4. Пользовательская функция возвращает информацию о всех покупателях однофамильцах

```
CREATE function LName_clients (@LNC nvarchar (50))  
RETURNS TABLE  
AS  
  
RETURN (Select * FROM Client AS c  
WHERE c.lname = @LNC)
```



The screenshot shows a SQL Server Enterprise Manager window with three tabs: 'SQLQuery4.sql - PT...g (PTG\Павел (54))\*', 'SQLQuery3.sql - PT...g (PTG\Павел (62))\*', and 'SQLQuery1.sql'. The active tab is 'SQLQuery4.sql'. The SQL text in the editor is:

```
create function LName_clients (@LNC nvarchar (50))  
returns TABLE  
as  
  
return (Select * from Client as c  
WHERE c.lname = @LNC)  
  
SELECT* from LName_clients('Juan');
```

Below the editor, there is a zoom level of '100 %' and two tabs: 'Результаты' (Results) and 'Сообщения' (Messages). The 'Результаты' tab is active, displaying a table with 10 columns: 'id', 'Name', 'Lname', 'Fname', 'email', 'phone', 'sex', 'percents', and 'subscribe'. The table contains two rows of data.

	id	Name	Lname	Fname	email	phone	sex	percents	subscribe
1	3	Don	Juan	MAximovich	gvieo@mail.com	2533345	1	14.00	1
2	61	Donna-Margarita	Juan	Nikolavna	gvo@mail.com	2577345	0	24.00	1

5. Пользовательская функция возвращает информацию о всех покупателях и продавцах однофамильцах.

```
CREATE function ALL_LName_matches (@LA nvarchar (50))

RETURNS @Teskas TABLE(First_name nvarchar (50) not null,
Last_Name nvarchar (50) not null,
Fathers_Name nvarchar (50) not null)
AS
BEGIN

DECLARE @tmp_teskas table (Name nvarchar (50) not null,
Lname nvarchar (50) not null, F_Name nvarchar (50) not null)

INSERT @tmp_teskas Select c.Name, c.Lname, c.fname FROM Client AS
c
WHERE c.lname = @LA

INSERT @tmp_teskas Select st.Name, st.Lname, st.Fname FROM Staff
AS st WHERE st.lname = @LA

INSERT @Teskas SELECT* FROM @tmp_teskas
RETURN
END;
```

The screenshot displays a SQL Server Enterprise Manager window. The top pane shows the definition of a user-defined function named `ALL_LName_matches`. The function takes a parameter `@LA nvarchar (50)` and returns a table `@Teskas` with columns `First_name`, `Last_Name`, and `Fathers_Name`. The function body includes two `INSERT` statements to populate a temporary table `@tmp_teskas` with data from the `Client` and `Staff` tables, where the last name matches `@LA`. Finally, the data from `@tmp_teskas` is inserted into `@Teskas`. The bottom pane shows the results of the query `SELECT* from ALL_LName_matches('Hlebov');`. The results are displayed in a table with 4 rows and 3 columns: `First_name`, `Last_Name`, and `Fathers_Name`.

	First_name	Last_Name	Fathers_Name
1	Ivan	Hlebov	Oristarkovich
2	Gleb	Hlebov	Oristarkovich
3	Bob	Hlebov	Bobovich
4	George	Hlebov	Shprotovich

## 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()),  
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 ('%@%'))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()),
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
тех', 5000),
('Жух-жух 17', 'скребок ушной', 1000, 199, 'Когда никто не
слышит', 400),
('Klizma', 'Prazdnik', 44, 200, 'StroyKlizmMASH', 1000);

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', 'hlebal@mail.com', 233377345,
1, 21.00, 1),
('Gleb', 'Hlebov', 'Oristarhovich', 'Glebal@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);

```