

## Домашнее задание 10

**Задание 1.** Доработать БД «Спортивный магазин», доделать недостающие таблицы

**Задание 2.** Реализовать следующие запросы

-Вывести название , себестоимость и цену товара проданного за определенную дату 10.06.2022

-Вывести ФИО, пол, email покупателя купившего товар определенного наименования (например "гантели")

-Вывести сумму продаж каждого сотрудника

-Вывести сумму продаж каждого наименования для каждого сотрудника

-Вывести сколько клиентов обслужил каждый сотрудник в дату 3.02.2022

**Задание 3.** К базе данных «Спортивный магазин» создайте следующие триггеры:

1. При добавлении нового товара триггер проверяет его наличие на складе, если такой товар есть и новые данные о товаре совпадают с уже существующими данными, вместо добавления происходит обновление информации о количестве товара

2. Если после продажи товара не осталось ни одной единицы данного товара, необходимо перенести информацию о полностью проданном товаре в таблицу «Архив»

3. Триггер запрещает добавлять нового продавца, если количество существующих продавцов больше 6.

4. Запретить удаление сотрудников, принятых на работу до 2015 года

**Задание 1. Доработать БД «Спортивный магазин», доделать недостающие таблицы**

```
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);

INSERT INTO Client ( Name, Lname, Fname, email, phone, sex,
percents,
subscribe) VALUES
('Кишштоф', 'Мухин', 'Агафонович', 'fly@mail.com', 656254, 1,
15.00, 1),
('Гвидон', 'Кулебякин', 'Пироксимович', 'gvido@mail.com', 253545,
1, 11.00, 0);

```

```

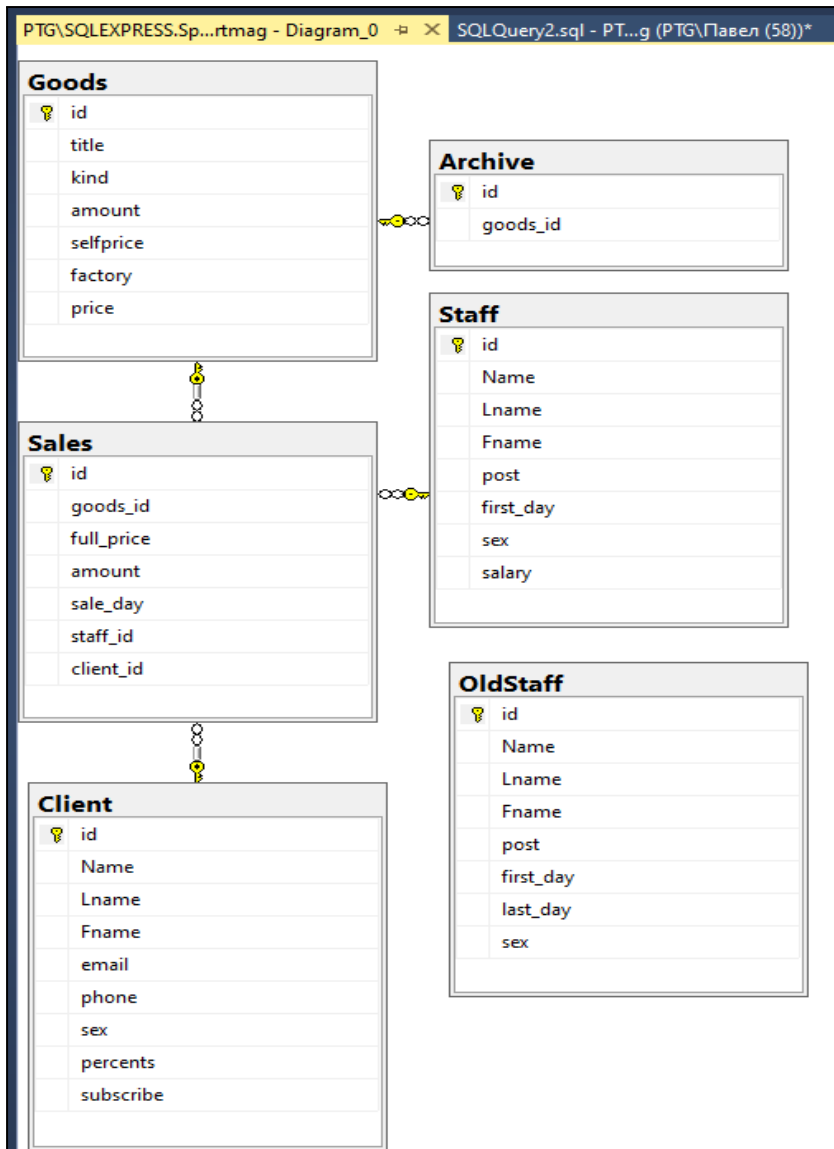
INSERT INTO Staff (Name, Lname, Fname, post, first_day, sex,
salary) VALUES
('Ганс', 'Поберушкин', 'Охламонович', 'страшный продаван',
'2014-05-05', 1, 2500),
('Клёпа', 'Фон Амурский', 'Сигизмундович', 'продажник 2 кат.',
'2016-02-02', 1, 500),
('Галя', 'Гром', 'Победитовна', 'кассир-продавец', '2000-01-01',
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);

```

```

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);

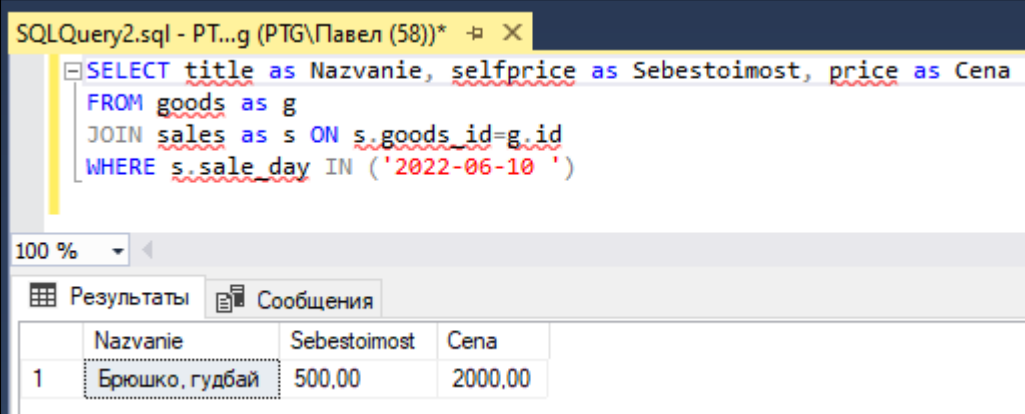
```



## Задание 2. Реализовать следующие запросы

-Вывести название , себестоимость и цену товара проданного за определенную дату 10.06.2022:

```
SELECT title as Nazvanie, self_price as Sebestoimost, price as Cena
FROM goods as g
JOIN sales as s ON s.goods_id=g.id
WHERE s.sale_day IN ('2022-06-10 ')
```

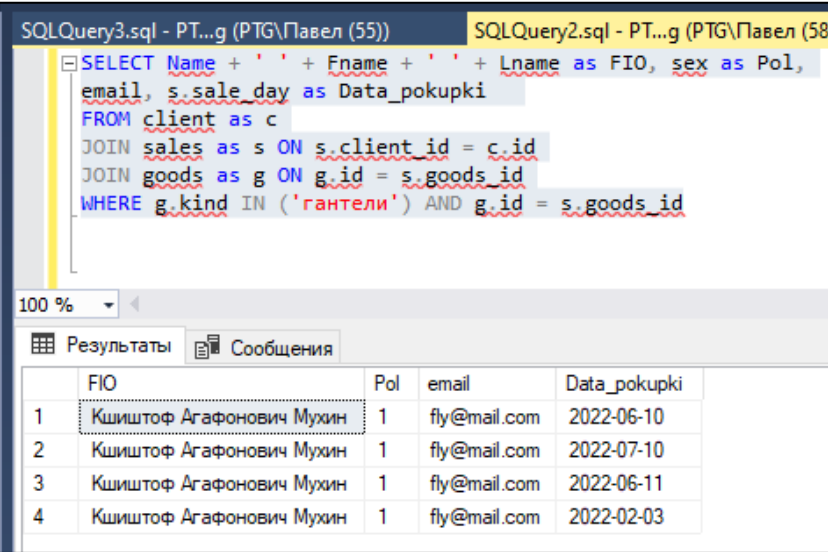


The screenshot shows a SQL query window titled "SQLQuery2.sql - PT...g (PTG\Павел (58))". The query is:   
`SELECT title as Nazvanie, selfprice as Sebestoimost, price as Cena  
FROM goods as g  
JOIN sales as s ON s.goods_id=g.id  
WHERE s.sale_day IN ('2022-06-10 ')`  
Below the query, the "Результаты" (Results) tab is active, displaying a table with the following data:

	Nazvanie	Sebestoimost	Cena
1	Брюшко, гудбай	500,00	2000,00

-Вывести ФИО, пол, email покупателя купившего товар определенного наименование (например "гантели"):

```
SELECT Name + ' ' + Fname + ' ' + Lname as FIO, sex as Pol,
email, s.sale_day as Data_pokupki
FROM client as c
JOIN sales as s ON s.client_id = c.id
JOIN goods as g ON g.id = s.goods_id
WHERE g.kind IN ('гантели') AND g.id = s.goods_id
```

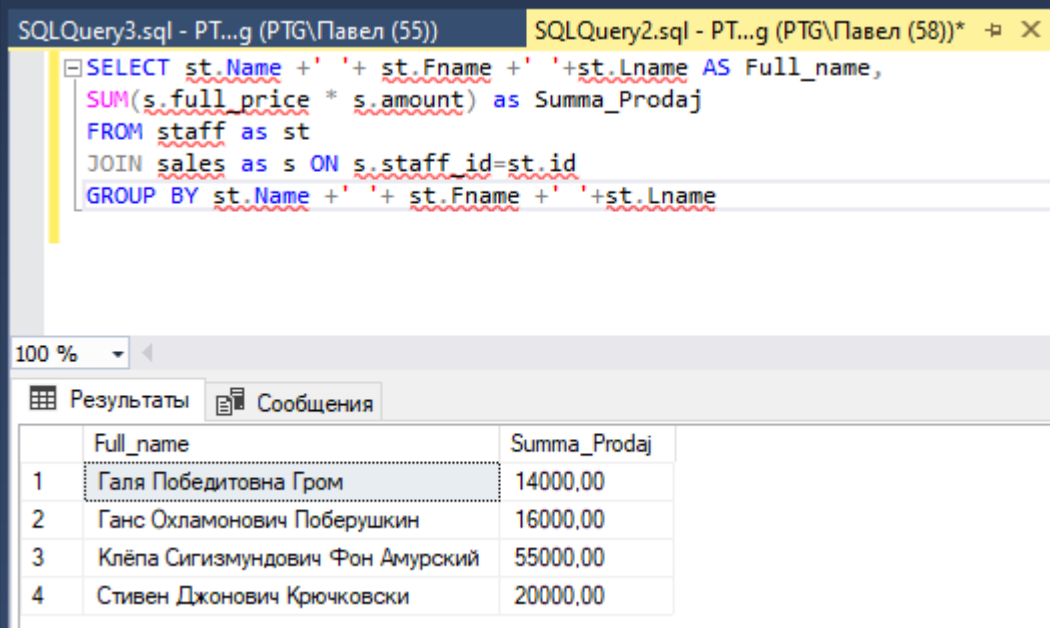


The screenshot shows a SQL query window titled "SQLQuery3.sql - PT...g (PTG\Павел (55))". The query is:   
`SELECT Name + ' ' + Fname + ' ' + Lname as FIO, sex as Pol,  
email, s.sale_day as Data_pokupki  
FROM client as c  
JOIN sales as s ON s.client_id = c.id  
JOIN goods as g ON g.id = s.goods_id  
WHERE g.kind IN ('гантели') AND g.id = s.goods_id`  
Below the query, the "Результаты" (Results) tab is active, displaying a table with the following data:

	FIO	Pol	email	Data_pokupki
1	Кшиштоф Агафонович Мухин	1	fly@mail.com	2022-06-10
2	Кшиштоф Агафонович Мухин	1	fly@mail.com	2022-07-10
3	Кшиштоф Агафонович Мухин	1	fly@mail.com	2022-06-11
4	Кшиштоф Агафонович Мухин	1	fly@mail.com	2022-02-03

-Вывести сумму продаж каждого сотрудника:

```
SELECT st.Name + ' ' + st.Fname + ' ' + st.Lname AS Full_name,  
SUM(s.full_price * s.amount) as Summa_Prodaj  
FROM staff as st  
JOIN sales as s ON s.staff_id=st.id  
GROUP BY st.Name + ' ' + st.Fname + ' ' + st.Lname
```



The screenshot shows a SQL query editor with two tabs. The active tab is 'SQLQuery3.sql - PT...g (PTG\Павел (55))'. The query is as follows:

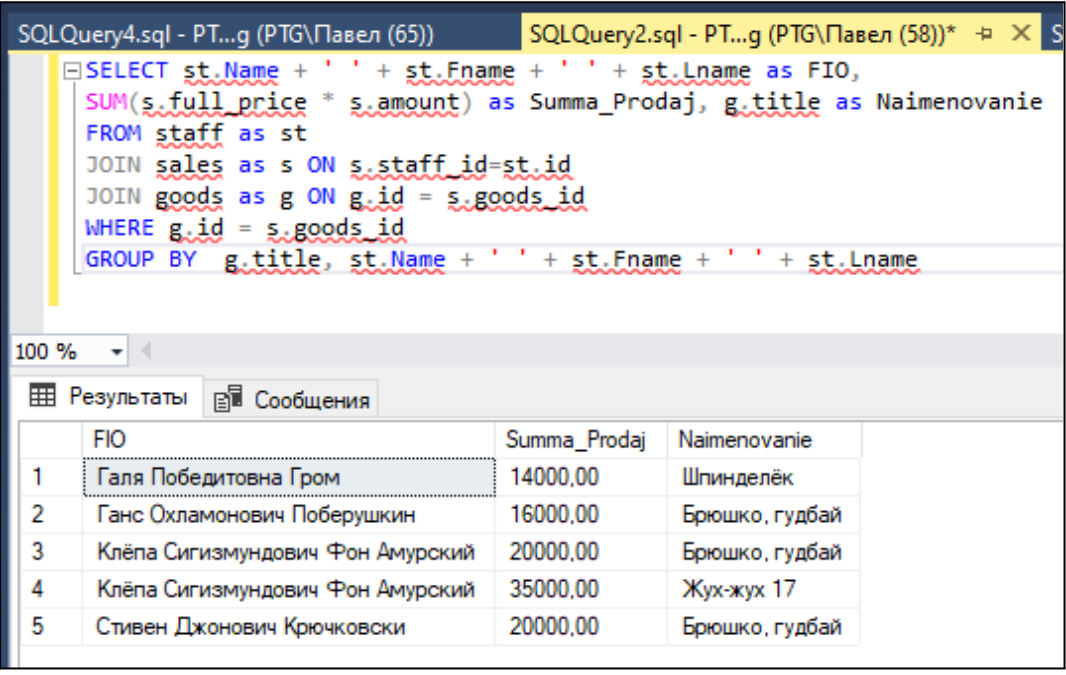
```
SELECT st.Name + ' ' + st.Fname + ' ' + st.Lname AS Full_name,  
SUM(s.full_price * s.amount) as Summa_Prodaj  
FROM staff as st  
JOIN sales as s ON s.staff_id=st.id  
GROUP BY st.Name + ' ' + st.Fname + ' ' + st.Lname
```

Below the query editor, there is a zoom level of 100% and two tabs: 'Результаты' (Results) and 'Сообщения' (Messages). The 'Результаты' tab is active, showing a table with 4 rows and 2 columns: 'Full\_name' and 'Summa\_Prodaj'.

	Full_name	Summa_Prodaj
1	Галя Победитовна Гром	14000,00
2	Ганс Охламонович Поберушкин	16000,00
3	Клёпа Сигизмундович Фон Амурский	55000,00
4	Стивен Джонович Крючковски	20000,00

*-Вывести сумму продаж каждого наименования для каждого сотрудника*

```
SELECT st.Name + ' ' + st.Fname + ' ' + st.Lname as FIO,  
SUM(s.full_price * s.amount) as Summa_Prodaj, g.title as  
Naimenovanie  
FROM staff as st  
JOIN sales as s ON s.staff_id=st.id  
JOIN goods as g ON g.id = s.goods_id  
WHERE g.id = s.goods_id  
GROUP BY g.title, st.Name + ' ' + st.Fname + ' ' + st.Lname
```



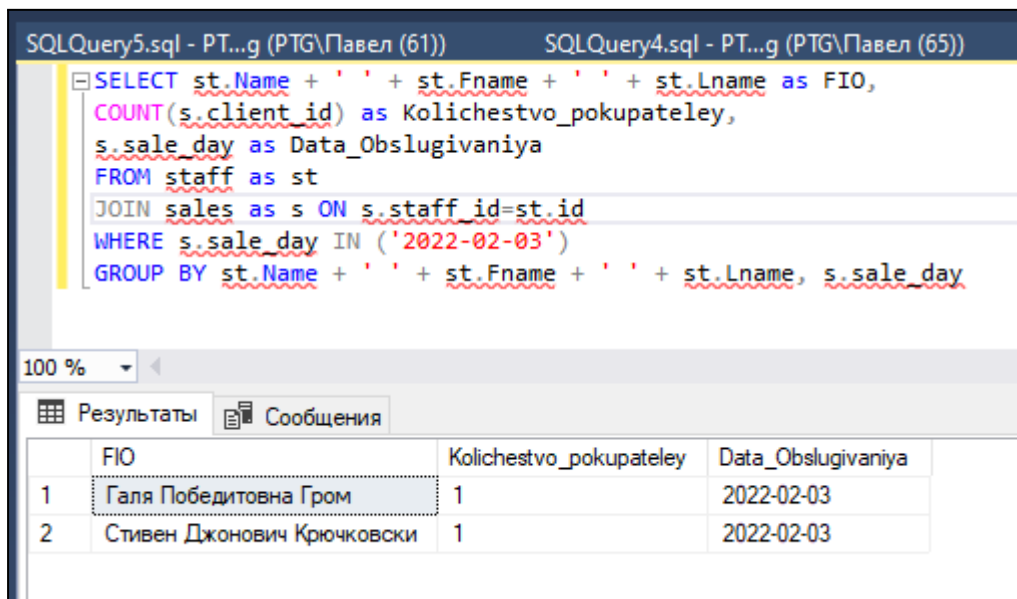
The screenshot shows a SQL query editor with two tabs. The active tab is 'SQLQuery4.sql - PT...g (PTG\Павел (65))'. The query is the same as the one in the previous block. Below the query editor, there is a '100 %' zoom level indicator and two tabs: 'Результаты' (Results) and 'Сообщения' (Messages). The 'Результаты' tab is active, displaying a table with 5 rows and 4 columns: FIO, Summa\_Prodaj, Naimenovanie, and an implicit index column. The data is as follows:

	FIO	Summa_Prodaj	Naimenovanie
1	Галя Победитовна Гром	14000,00	Шпинделёк
2	Ганс Охламонович Поберушкин	16000,00	Брюшко, гудбай
3	Клёпа Сигизмундович Фон Амурский	20000,00	Брюшко, гудбай
4	Клёпа Сигизмундович Фон Амурский	35000,00	Жух-жух 17
5	Стивен Джонович Крючковски	20000,00	Брюшко, гудбай



**-Вывести сколько клиентов обслужил каждый сотрудник в дату 3.02.2022**

```
SELECT st.Name + ' ' + st.Fname + ' ' + st.Lname as FIO,  
COUNT(s.client_id) as Kolichество_pokupateley,  
s.sale_day as Data_Obslugivaniya  
FROM staff as st  
JOIN sales as s ON s.staff_id=st.id  
WHERE s.sale_day IN ('2022-02-03')  
GROUP BY st.Name + ' ' + st.Fname + ' ' + st.Lname, s.sale_day
```



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery5.sql - PT...g (PTG\Павел (61))' and 'SQLQuery4.sql - PT...g (PTG\Павел (65))'. The active tab displays the following SQL query:

```
SELECT st.Name + ' ' + st.Fname + ' ' + st.Lname as FIO,  
COUNT(s.client_id) as Kolichество_pokupateley,  
s.sale_day as Data_Obslugivaniya  
FROM staff as st  
JOIN sales as s ON s.staff_id=st.id  
WHERE s.sale_day IN ('2022-02-03')  
GROUP BY st.Name + ' ' + st.Fname + ' ' + st.Lname, s.sale_day
```

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

	FIO	Kolichество_pokupateley	Data_Obslugivaniya
1	Галя Победитовна Гром	1	2022-02-03
2	Стивен Джонович Крючковски	1	2022-02-03

**Задание 3.** К базе данных «Спортивный магазин» создайте следующие триггеры:

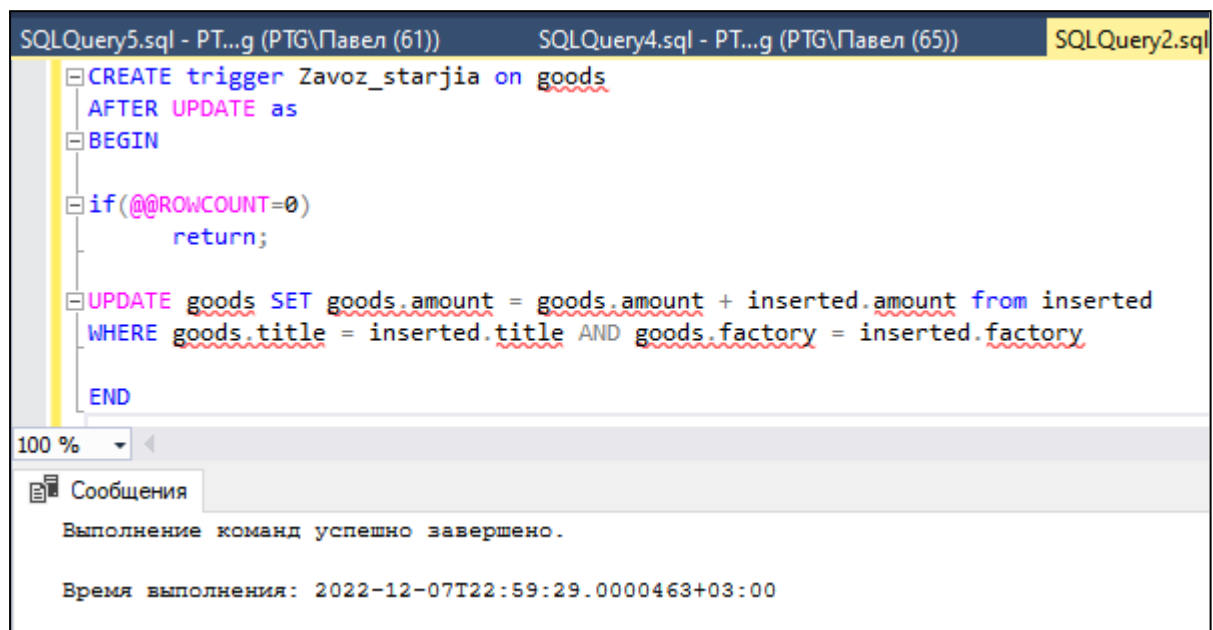
1. При добавлении нового товара триггер проверяет его наличие на складе, если такой товар есть и новые данные о товаре совпадают с уже существующими данными, вместо добавления происходит обновление информации о количестве товара.

```
CREATE trigger Zavoz_starjia on goods
AFTER UPDATE as
BEGIN

if (@@ROWCOUNT=0)
    return;

UPDATE goods SET goods.amount = goods.amount + inserted.amount
from inserted
WHERE goods.title = inserted.title AND goods.factory =
inserted.factory

END
```



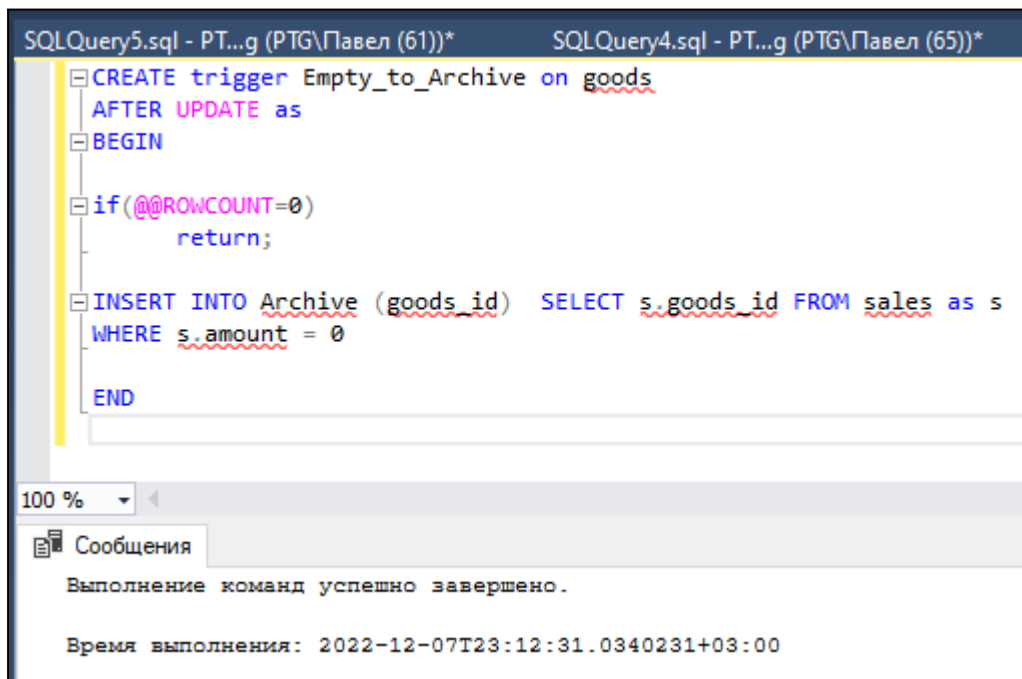
2. Если после продажи товара не осталось ни одной единицы данного товара, необходимо перенести информацию о полностью проданном товаре в таблицу «Архив»

```
CREATE trigger Empty_to_Archive on goods
AFTER UPDATE as
BEGIN

if (@@ROWCOUNT=0)
    return;

INSERT INTO Archive (goods_id)  SELECT s.goods_id
FROM sales as s
WHERE s.amount = 0

END
```



3. Триггер запрещает добавлять нового продавца, если количество существующих продавцов больше 6.

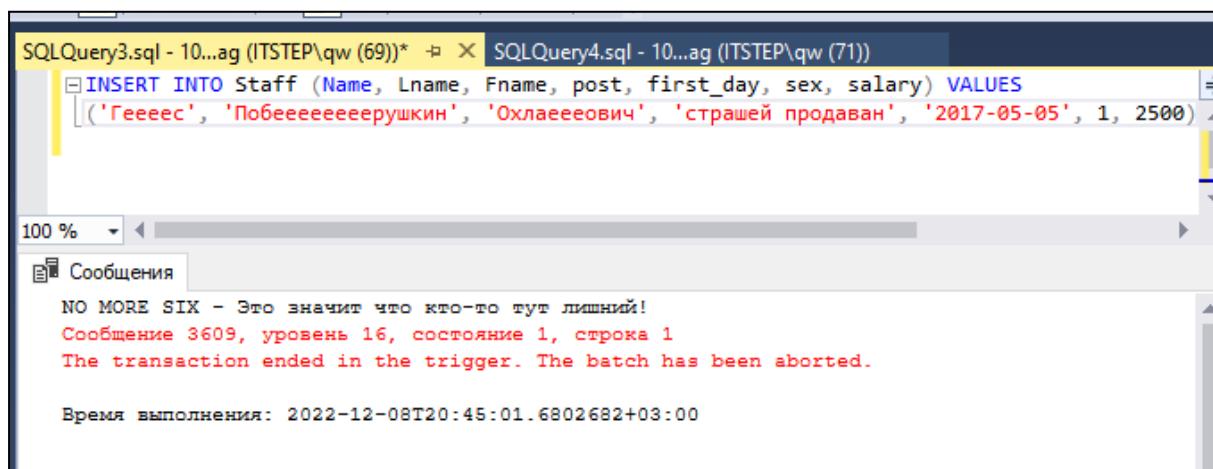
```
CREATE trigger No_More_Six ON Staff
FOR INSERT as

BEGIN
    declare @how_much int
    SET @how_much = (SELECT COUNT(id) FROM Staff)
    declare @New_id int
    SELECT @New_id = id
    FROM inserted
    IF (@how_much > 6 AND @New_id > @how_much )

BEGIN
    raiserror ('NO MORE SIX - Это значит что кто-то тут лишний!', 0,1)
    ROLLBACK TRANSACTION
END

ELSE
    PRINT ('Добавлен новый герой')

END
```



#### 4. Запретить удаление сотрудников, принятых на работу до 2015 года

```
CREATE trigger No_Del_2015 on Staff
FOR DELETE as

BEGIN

if (@@ROWCOUNT=0)
    return;

declare @bad_day date
SET @bad_day = (SELECT first_day FROM deleted)

IF (@bad_day <  '2015-01-01' )

BEGIN
raiserror ('Такого не сотрёшь!', 0,1)
ROLLBACK TRANSACTION
END

ELSE
PRINT ('Гудбай, товарищ!')

END
```

