

Домашнее задание 11

Триггеры

1. При добавлении продавца триггер проверяет есть ли он в таблице покупателей, если запись существует добавление нового продавца отменяется
2. Триггер не позволяет вставлять информацию о продаже таких товаров: яблоки, груши, сливы, кинза.
3. Не позволять регистрировать уже существующего клиента. При вставке проверять наличие клиента по ФИО и email
4. Запретить удаление существующих клиентов

Процедуры

1. Хранимая процедура показывает информацию о всех продавцах
2. Хранимая процедура показывает информацию о всех покупателях
3. Хранимая процедура показывает полную информацию о продажах
4. Хранимая процедура показывает полную информацию о всех продажах в конкретный день. Дата продажи передаётся в качестве параметра
5. Хранимая процедура отображает информацию о продажах конкретного продавца. ФИО продавца передаётся в качестве параметра хранимой процедуры
6. Хранимая процедура возвращает среднеарифметическую цену продажи в конкретный год. Год передаётся в качестве параметра.

Триггеры

1. При добавлении продавца триггер проверяет есть ли он в таблице покупателей, если запись существует добавление нового продавца отменяется

```
CREATE trigger Staff_not_client ON Staff
for INSERT as
BEGIN

IF (EXISTS(SELECT staff.name, staff.lname, staff.fname      FROM
staff
join inserted ON staff.id= inserted.id
join client on client.name = inserted.name
WHERE  client.name = inserted.name
client.lname = inserted.lname
AND client.fname = inserted.fname))

BEGIN
RAISERROR('Не бывать Продавцу покупателем!',0,1)
ROLLBACK TRANSACTION
END

ELSE
PRINT ('Добавлен новый герой')

END
```

```
SQLQuery2.sql - PT...g (PTG\Павел (52)) *
CREATE trigger Staff_not_client ON Staff
for INSERT as
BEGIN
IF (EXISTS(SELECT staff.name, staff.lname, staff.fname FROM staff
join inserted ON staff.id= inserted.id
join client on client.name = inserted.name
WHERE
client.lname = inserted.lname
AND client.fname = inserted.fname))
BEGIN
RAISERROR('Не бывать Продавцу покупателем!',0,1)
ROLLBACK TRANSACTION
END

ELSE
PRINT ('Добавлен новый герой')

END

INSERT INTO Staff (Name, Lname, Fname, post, first_day, sex, salary) VALUES
('Don', 'Juan', 'MAximovich', 'страшный продаван', '2014-05-05', 1, 2500)
```

100 %

Сообщения

Не бывать Продавцу покупателем!
Сообщение 3609, уровень 16, состояние 1, строка 23
Транзакция завершилась в триггере. Выполнение пакета прервано.

Время выполнения: 2022-12-12T22:18:46.8833850+03:00

2. Триггер не позволяет вставлять информацию о продаже таких товаров: яблоки, груши, сливы, кинза.

```
CREATE trigger Destroy_Fruit ON Sales
FOR INSERT as
BEGIN
if(@@ROWCOUNT=0)
    return;

declare @fruit nvarchar(100)

SET @fruit = (SELECT kind FROM Goods
join inserted ON goods.id = inserted.goods_id
WHERE goods.kind in ('яблоки')
OR goods.kind in ('груши')
OR goods.kind in ('сливы')
OR goods.kind in ('кинза'))

IF (@fruit = ('яблоки')
OR @fruit = ('груши')
OR @fruit = ('сливы')
OR @fruit = ('кинза'))

BEGIN
RAISERROR ('Это спортмагазин, а не овощебаза!!',0,1)
ROLLBACK TRANSACTION
END

ELSE print('Если ты не овощ, тогда ты идешь к нам!')
END
```

```
SQLQuery2.sql - PT...g (PTG\Павел (65))    SQLQuery1.sql - PT...g (PTG\Павел (57))
CREATE trigger Destroy_Fruit ON Sales
for INSERT as
BEGIN
if(@@ROWCOUNT=0)
    return;

declare @fruit nvarchar(100)
SET @fruit = (SELECT kind FROM Goods
join inserted ON goods.id = inserted.goods_id
WHERE goods.kind in ('яблоки')
OR goods.kind in ('груши')
OR goods.kind in ('сливы')
OR goods.kind in ('кинза'))

if (@fruit = ('яблоки')
OR @fruit = ('груши')
OR @fruit = ('сливы')
OR @fruit = ('кинза'))

BEGIN
RAISERROR ('Это спортмагазин, а не овощебаза!!',0,1)
ROLLBACK TRANSACTION
END

ELSE print('Если ты не овощь, тогда ты идёшь к нам!')
END
```

100 %

Сообщения

Это спортмагазин, а не овощебаза!!
Сообщение 3609, уровень 16, состояние 1, строка 28
Транзакция завершилась в триггере. Выполнение пакета прервано.

Время выполнения: 2022-12-11T13:28:01.4362267+03:00

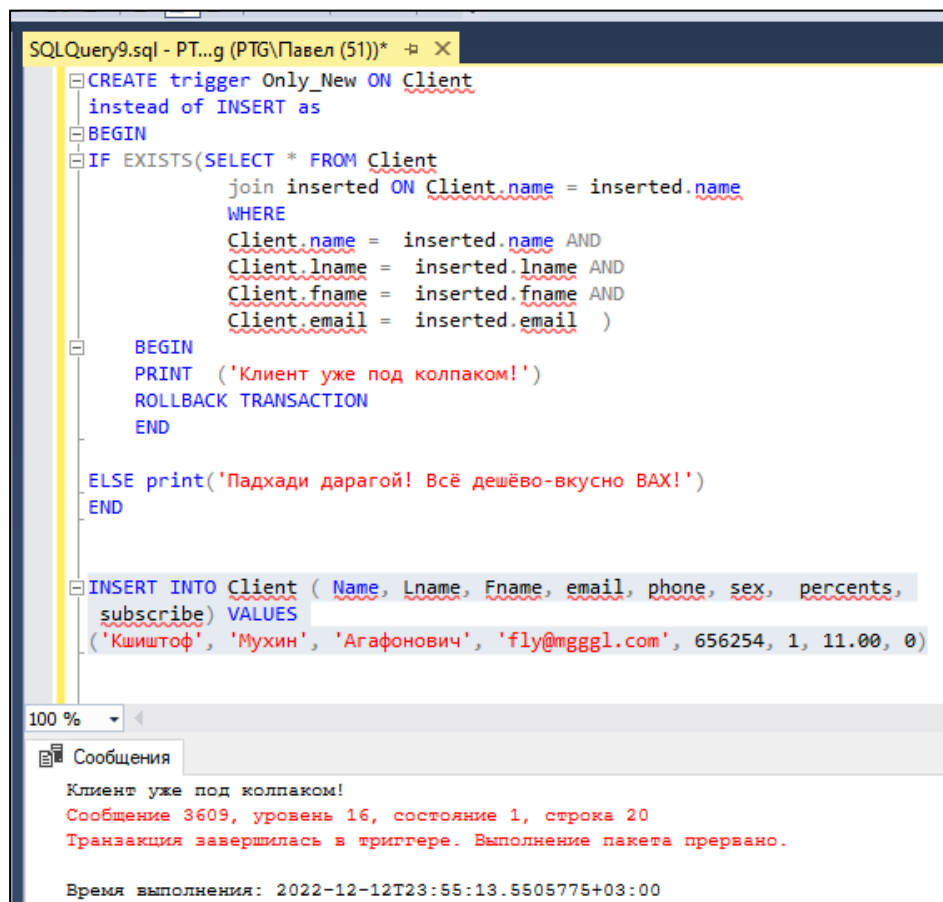
3. Не позволять регистрировать уже существующего клиента. При вставке проверять наличие клиента по ФИО и email

```
CREATE trigger Only_New ON Client
instead of INSERT as

BEGIN
IF EXISTS(SELECT * FROM Client
join inserted ON Client.name = inserted.name
WHERE
Client.name = inserted.name AND
Client.lname = inserted.lname AND
Client.fname = inserted.fname AND
Client.email = inserted.email )

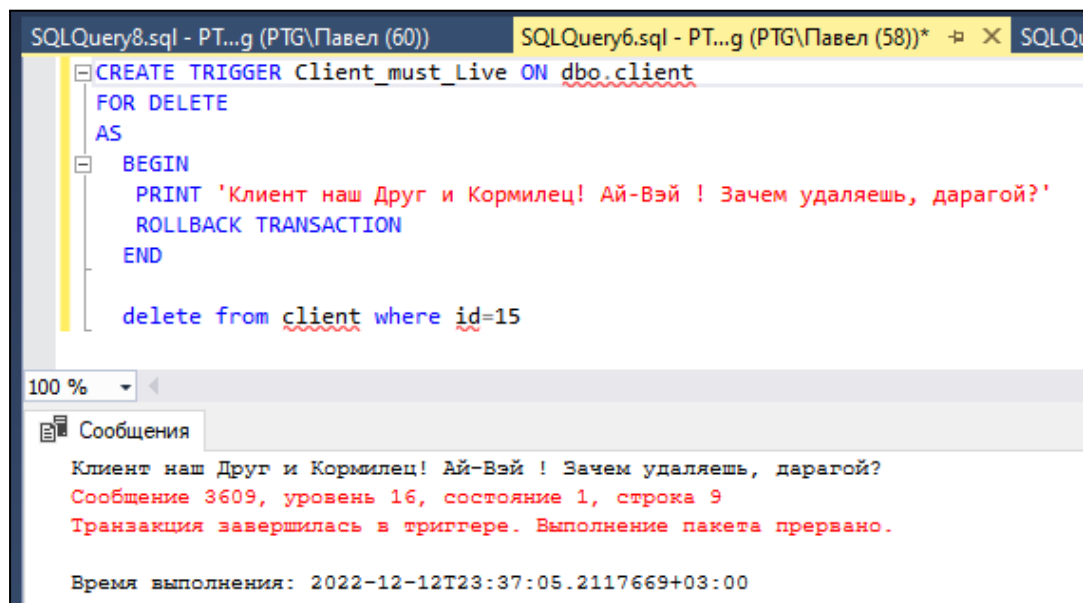
BEGIN
PRINT ('Клиент уже под колпаком!')
ROLLBACK TRANSACTION
END

ELSE print('Падхади дарагой! Всё дешёво-вкусно BAX!')
END
```



4. Запретить удаление существующих клиентов

```
CREATE TRIGGER Client_must_Live ON dbo.client
FOR DELETE
AS
BEGIN
PRINT 'Клиент наш Друг и Кормилец! Ай-Вэй ! Зачем удаляешь,
дарагой?'
ROLLBACK TRANSACTION
END
```

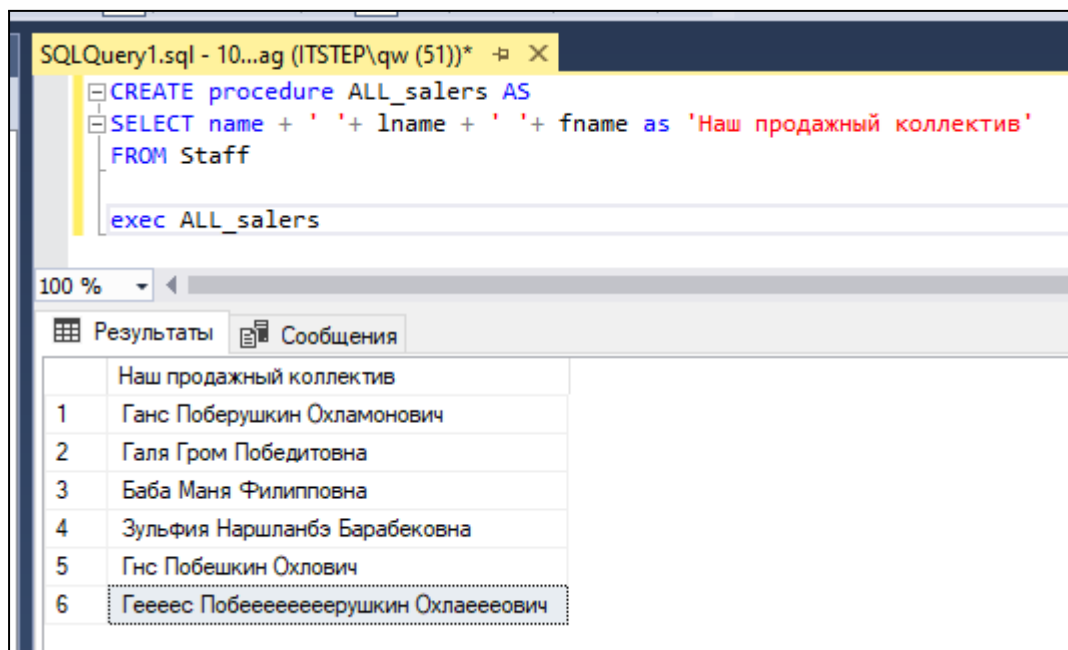


Процедуры

1. Хранимая процедура показывает информацию о всех продавцах

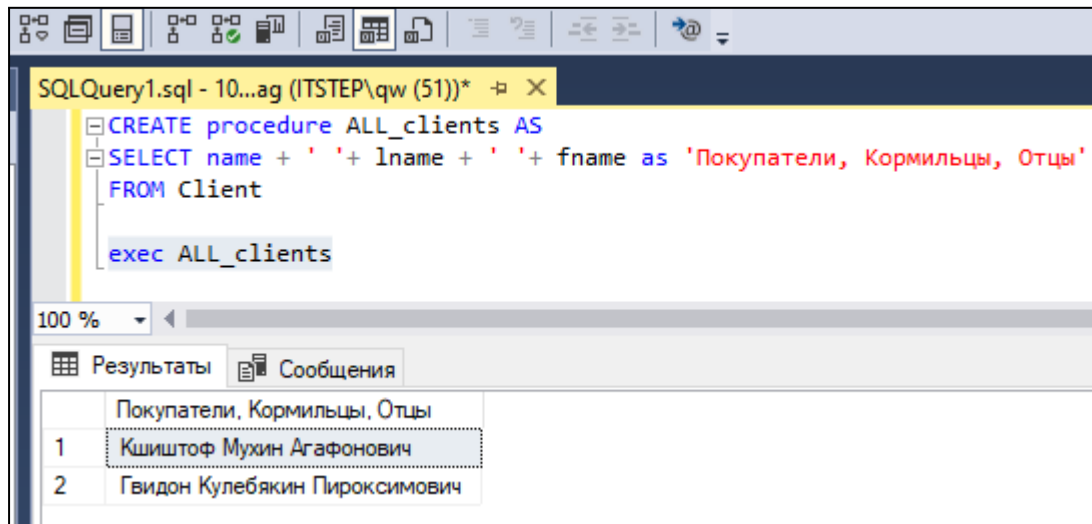
```
CREATE procedure ALL_salers AS  
SELECT name + ' ' + lname + ' ' + fname as 'Наш продажный коллектив'  
FROM Staff
```

```
exec ALL_salers
```



2. Хранимая процедура показывает информацию о всех покупателях

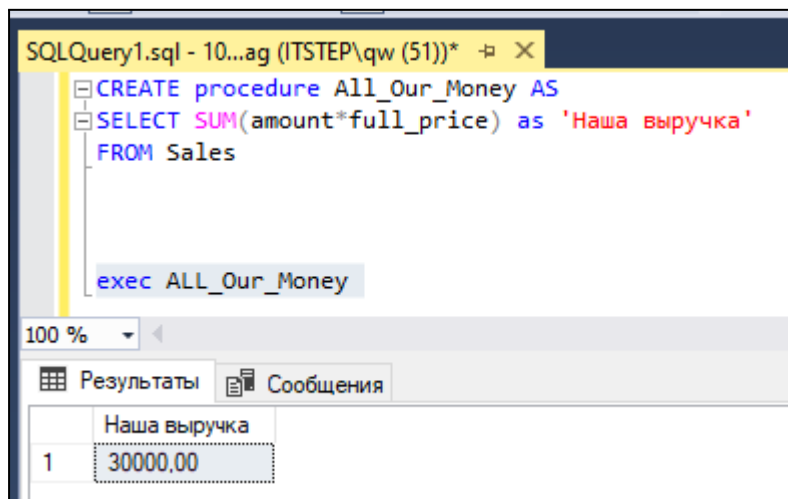
```
CREATE procedure ALL_clients AS  
SELECT name + ' ' + lname + ' ' + fname as 'Покупатели, Кормильцы, Отцы'  
FROM Client  
  
exec ALL_clients
```



3. Хранимая процедура показывает полную информацию о продажах

```
CREATE procedure All_Our_Money AS  
SELECT SUM(amount*full_price) as 'Наша выручка'  
FROM Sales
```

```
exec ALL_Our_Money
```

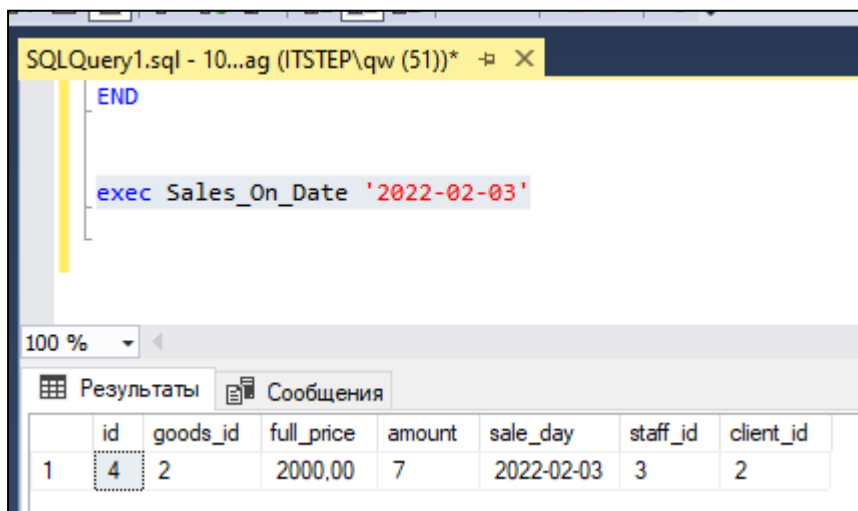


4. Хранимая процедура показывает полную информацию о всех продажах в конкретный день. Дата продажи передается в качестве параметра

```
CREATE procedure Sales_On_Date (@date date) AS  
BEGIN
```

```
SELECT *  
FROM Sales  
WHERE sale_day = @date  
END
```

```
exec Sales_On_Date '2022-02-03'
```



The screenshot shows a SQL Server Enterprise Manager window with the title 'SQLQuery1.sql - 10...ag (ITSTEP\qw (51))*'. The query editor displays the stored procedure definition and its execution: `END` and `exec Sales_On_Date '2022-02-03'`. Below the editor, the 'Results' tab is active, showing a table with 8 columns: `id`, `goods_id`, `full_price`, `amount`, `sale_day`, `staff_id`, and `client_id`. The table contains one row of data.

	id	goods_id	full_price	amount	sale_day	staff_id	client_id
1	4	2	2000,00	7	2022-02-03	3	2

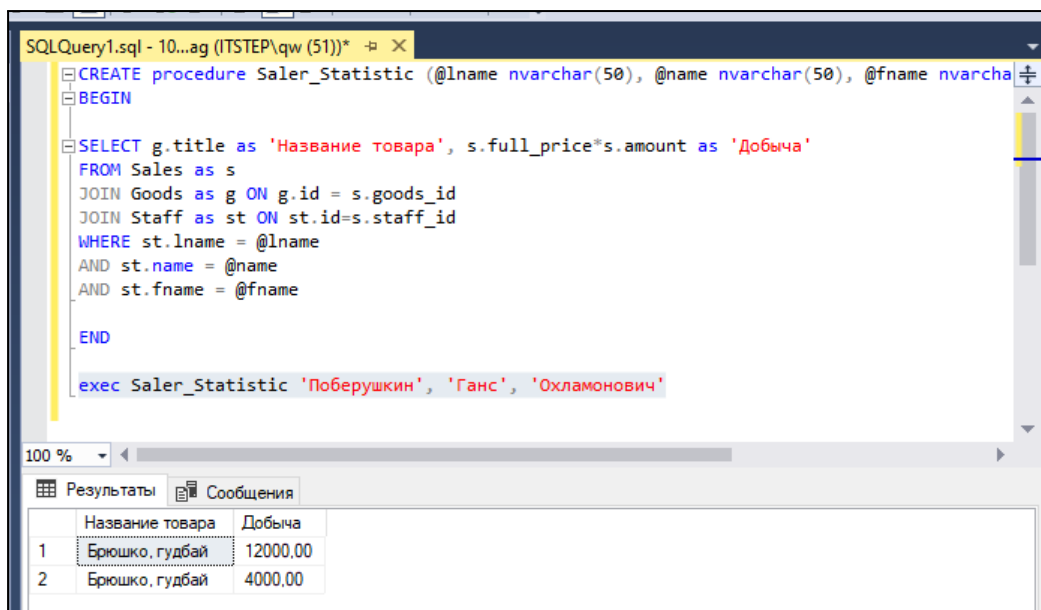
5. Хранимая процедура отображает информацию о продажах конкретного продавца. ФИО продавца передается в качестве параметра хранимой процедуры

```
CREATE procedure Saler_Statistic (@lname nvarchar(50), @name
nvarchar(50), @fname nvarchar(50)) AS
BEGIN

SELECT g.title as 'Название товара', s.full_price*s.amount as
'Добыча'
FROM Sales as s
JOIN Goods as g ON g.id = s.goods_id
JOIN Staff as st ON st.id=s.staff_id
WHERE st.lname = @lname
AND st.name = @name
AND st.fname = @fname

END
```

```
exec Saler_Statistic 'Поберушкин', 'Ганс', 'Охламонович'
```

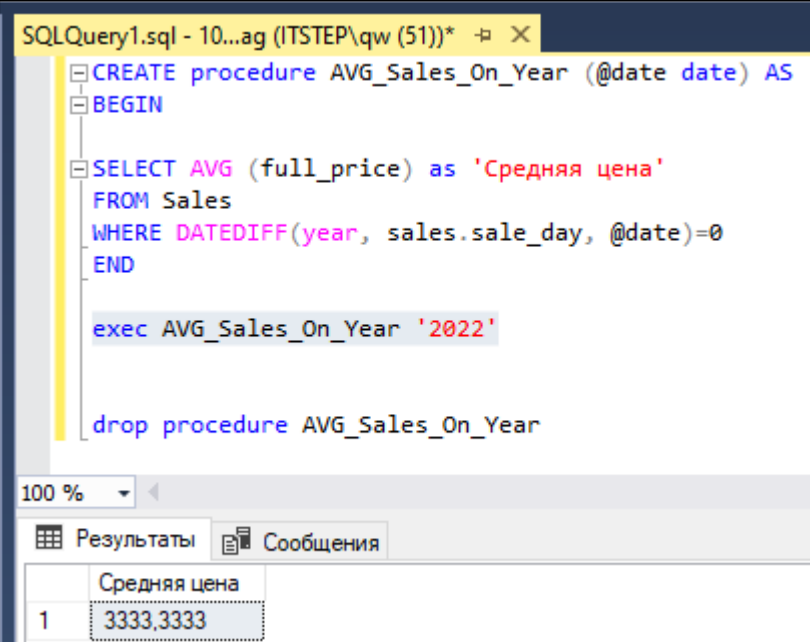


6. Хранимая процедура возвращает среднюю арифметическую цену продажи в конкретный год. Год передается в качестве параметра.

```
CREATE procedure AVG_Sales_On_Year (@date date) AS  
BEGIN
```

```
SELECT AVG (full_price)  
FROM Sales  
WHERE DATEDIFF(year, sales.sale_day, @date)=0  
END
```

```
exec AVG_Sales_On_Year '2022'
```



The screenshot shows a SQL Server query window titled "SQLQuery1.sql - 10...ag (ITSTEP\qw (51))*". The query text is as follows:

```
CREATE procedure AVG_Sales_On_Year (@date date) AS  
BEGIN  
    SELECT AVG (full_price) as 'Средняя цена'  
    FROM Sales  
    WHERE DATEDIFF(year, sales.sale_day, @date)=0  
END  
  
exec AVG_Sales_On_Year '2022'  
  
drop procedure AVG_Sales_On_Year
```

Below the query window, the "Results" tab is active, displaying a single row of data:

	Средняя цена
1	3333,3333

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);

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);
```

```
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);
```