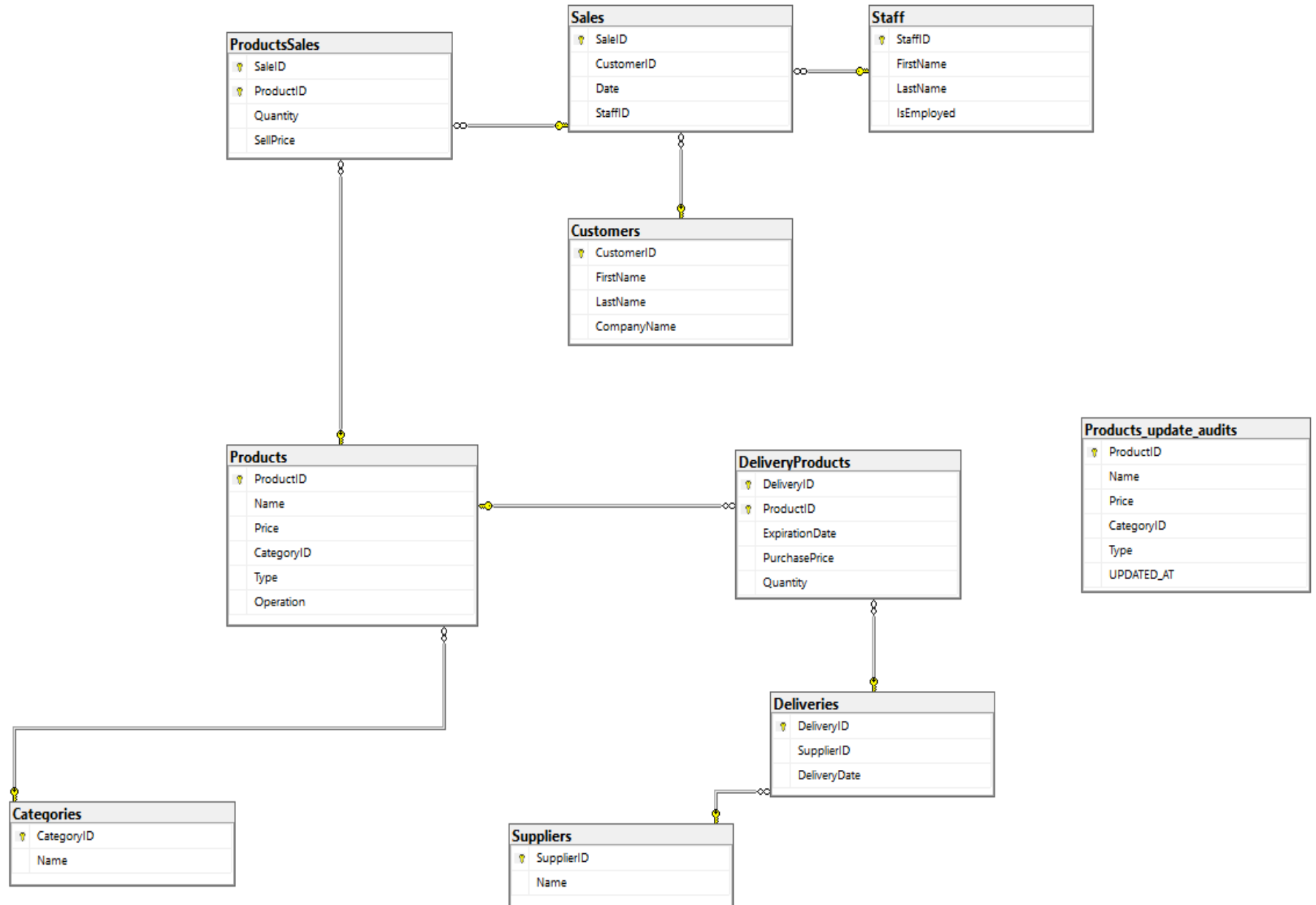


База от данни на магазин за хранителни СТОКИ

Изготвили: Любка Ангелинина 62342,
Мая Бораджиева 62335,
Николета Вълчинова 62322

Диаграма





Таблицы

- Products
- Categories
- Deliveries
- DeliveryProducts
- Suppliers
- Sales
- ProductsSales
- Staff
- Customers
- Products_update_audits

❑ Products

- Таблица, която съдържа номер на Продукт (уникален), неговото име, цена, тип (дали е на брой или на кг) и номер на категория. Това е таблица за основната информация за продуктите в магазина.
- Свързва се с Deliveries, чрез таблицата DeliveryProducts (връзка many-many), със Sales, чрез ProductsSales (връзка many-many) и с Categories (one-many).

```
CREATE TABLE Products(  
    ProductID INT PRIMARY KEY NOT NULL,  
    --играе роля на баркод  
    Name VARCHAR(50) NOT NULL,  
    Price DECIMAL(6,2) NOT NULL,  
    --цена на кг и на брой  
    CategoryID INT NOT NULL,  
    Type VARCHAR(5) NOT NULL DEFAULT 'piece' CHECK(Type IN ('kg', 'piece')),  
    CONSTRAINT FK_Product_Category  
    FOREIGN KEY(CategoryID)  
    REFERENCES Categories(CategoryID),  
);
```

```
CREATE TABLE DeliveryProducts(  
    DeliveryID INT NOT NULL,  
    ProductID INT NOT NULL,  
    PRIMARY KEY(DeliveryID, ProductID),  
    ExpirationDate DATETIME,  
    PurchasePrice DECIMAL(6,2) NOT NULL,  
    Quantity INT NOT NULL,  
    CONSTRAINT FK_DeliveryProduct_Delivery  
    FOREIGN KEY(DeliveryID)  
    REFERENCES Deliveries(DeliveryID),  
    CONSTRAINT FK_DeliveryProduct_Products  
    FOREIGN KEY(ProductID)  
    REFERENCES Products(ProductID)  
);
```

```
CREATE TABLE Categories(  
    CategoryID INT PRIMARY KEY NOT NULL,  
    Name VARCHAR(50) NOT NULL  
);
```

❑ Categories

- Таблица, която съдържа номер на категория и нейното име. Това е за разпределение на продуктите на категории - напитки, плодове и зеленчуци, млечни и т.н.
- Свързва се с таблицата Products, като всяка категория може да съдържа много продукти.

```
CREATE TABLE Deliveries(  
    DeliveryID INT PRIMARY KEY NOT NULL,  
    SupplierID INT NOT NULL,  
    DeliveryDate DATETIME NOT NULL,  
    CONSTRAINT FK_Delivery_Supplier  
    FOREIGN KEY(SupplierID)  
    REFERENCES Suppliers(SupplierID)  
);
```

❑ Deliveries

- Таблица, която съдържа номер на доставка (уникален), номер на Снабдителя (булстат) и дата на доставка. Така лесно проследяваме кога сме получавали от съответна компания доставка.
- Свързва се със Suppliers с връзка one-many, също и с Products чрез таблицата DeliveryProducts (в нея пазим цената, за която е доставен продукта, количеството от дадения продукт, както и неговия срок на годност).

```
CREATE TABLE Suppliers(  
    SupplierID INT PRIMARY KEY NOT NULL,  
    Name VARCHAR(50) NOT NULL  
);
```

❑ Suppliers


- Таблица, която съдържа номер на Снабдителя (булстат) и името на компанията-снабдител на магазина. Така знаем кои компании могат да ни доставят (т.е. с тях има сключен договор). Всеки доставчик може да участва в много доставки.

❑ Sales

- Таблица, която съдържа номер на продажба (уникален), номер на продавача, номер на клиента (optional) и дата на продажба, като така пазим данните за всички продажби.
- Свързва се с Customers и Staff (и двете са връзки one-many), също и с Products чрез таблицата ProductsSales (в нея пазим каква е била продажната цена на всеки продукт в даден момент).

```
CREATE TABLE Sales(  
    SaleID INT PRIMARY KEY NOT NULL,  
    CustomerID INT,  
    -- може да е NULL, ако не е фирма  
    Date DATETIME NOT NULL DEFAULT GETDATE(),  
    StaffID INT NOT NULL,  
    CONSTRAINT FK_Sales_Customers  
    FOREIGN KEY(CustomerID)  
    REFERENCES Customers(CustomerID),  
    CONSTRAINT FK_Sales_Staff  
    FOREIGN KEY(StaffID)  
    REFERENCES Staff(StaffID)  
);
```


```
CREATE TABLE ProductsSales(  
    SaleID INT NOT NULL,  
    ProductID INT NOT NULL,  
    Quantity DECIMAL(6,3) NOT NULL,  
    SellPrice DECIMAL(6,2) NOT NULL  
    PRIMARY KEY(SaleID,ProductID),  
    CONSTRAINT FK_ProductSales_Sale  
    FOREIGN KEY(SaleID)  
    REFERENCES Sales(SaleID),  
    CONSTRAINT FK_ProductSales_Product  
    FOREIGN KEY(ProductID)  
    REFERENCES Products(ProductID)  
);
```

```
CREATE TABLE Staff(  
    StaffID INT PRIMARY KEY NOT NULL,  
    FirstName VARCHAR(50) NOT NULL,  
    LastName VARCHAR(50) NOT NULL  
);
```

❑ Staff

- Таблица, която съдържа номер на продавача (уникален), две имена и дали все още има валиден трудов договор (isEmployed).



```
CREATE TABLE Customers(  
    CustomerID INT PRIMARY KEY NOT NULL,  
    FirstName VARCHAR(50) NOT NULL,  
    LastName VARCHAR(50) NOT NULL,  
    CompanyName VARCHAR(50) NOT NULL  
);
```

❑ Customers

- Таблица, която съдържа номер на купувача (уникален), две имена и име на фирмата. Ако купувача не желае фактура (т.е. не е фирма), то в таблицата Sales CustomerID е NULL.

Прости заявки

```
-- Заявка, която извежда ИМЕ, ЦЕНА и номер на Категория  
-- (името на колоната да се казва 'Category') на продуктите,  
-- които съдържат в името си или 'Water' или 'cafe'.  
-- Да са подредени по цената им.
```

```
SELECT Name, Price, CategoryID AS Category  
FROM Products  
WHERE Name LIKE '%Water%' OR Name LIKE '%cafe%'  
ORDER BY Price
```

RESULT:

	Name	Price	Category
1	Water Devin 500ml	0.80	2
2	Water Devin 750ml	1.10	2
3	Water Devin 1l	1.50	2
4	Nescafe Classic Crema 100g	6.30	10
5	Nescafe Classic 500g	8.50	10

```
-- Заявка, която извежда ПЪРВОИМЕ (името на колоната да се казва 'NAME')  
-- и номер на Персонал (името на колоната да се казва 'PASSWORD') на  
-- всички работници без тези на 'Ivan' и 'Lyubka'
```

```
SELECT FirstName AS NAME, StaffID AS PASSWORD  
FROM Staff  
WHERE FirstName NOT Like 'Ivan' AND FirstName NOT Like 'Lyubka'
```

RESULT:

	NAME	PASSWORD
1	Maya	2530
2	Boris	6240
3	Nikoleta	6569

Заявки на две и повече релации

```
-- Заявка, която извежда номер на прдукти,  
--които не са били включени в никоя продажба  
SELECT ProductID FROM Products  
EXCEPT  
((SELECT ProductID FROM DeliveryProducts)  
INTERSECT  
(SELECT ProductID FROM ProductsSales))
```

RESULT:

	ProductID
1	109
2	110
3	111
4	112
5	113
6	114
7	115
8	116
9	117
10	118

```
-- Заявка, която извежда номер на продукт,  
--количество, продажна цена и дата на продукти,  
--които са продадени от работник с номер 2810  
SELECT ProductID,Quantity, SellPrice,Date  
FROM ProductsSales , Sales  
WHERE ProductsSales.SaleID = Sales.SaleID  
AND StaffID=2810
```

RESULT:

	ProductID	Quantity	SellPrice	Date
1	102	2.000	1.20	2020-01-15 18:15:30.000
2	104	3.000	0.80	2020-01-15 18:15:30.000
3	106	0.800	3.60	2020-01-15 18:15:30.000
4	108	2.000	2.30	2020-01-15 18:15:30.000
5	103	1.000	3.50	2020-01-15 16:36:32.000

```
-- Заявка, която извежда името на снабдители,  
--от които нямаме доставка  
SELECT Name  
FROM Suppliers,  
((SELECT SupplierID FROM Suppliers)  
EXCEPT  
(SELECT SupplierID FROM Deliveries)) AS Table1  
WHERE Suppliers.SupplierID = Table1.SupplierID
```

RESULT:

	Name
1	Coca-cola

Подзаявки

```
--Изведете имената на продуктите с цена по-висока от  
--цената на продукта Rois Almond 80g.
```

```
SELECT Name  
FROM Products  
WHERE Price > ALL (SELECT Price  
                   FROM Products WHERE  
                   NAME = 'Rois Almond 80g');
```

RESULT:

	Name
1	Red Wine Cycle 750ml
2	Cucumbers
3	Nescafe Classic Crema 100g
4	Pink Tomatoes
5	White Wine Cycle 750ml
6	Nescafe Classic 500g

```
--Изведете срока на годност на продуктите Nescafe  
--доставени на 20 март 2020г.  
SELECT ProductID, ExpirationDate  
FROM DeliveryProducts  
WHERE DeliveryID IN (SELECT DeliveryID  
                    FROM Deliveries  
                    WHERE DeliveryDate = '2020-03-20')  
  
AND ProductID IN (SELECT ProductID  
                 FROM Products  
                 WHERE Name LIKE 'Nescafe %')
```

RESULT:

	ProductID	ExpirationDate
1	107	2021-11-10 00:00:00.000
2	114	2021-11-10 00:00:00.000

Съединения

```
--Изведете надценката на продуктите
--от категорията Grains
SELECT DISTINCT P.NAME, (Price - PurchasePrice) as Surcharge
FROM Products P JOIN DeliveryProducts DP ON P.ProductID = DP.ProductID
JOIN Categories C ON C.CategoryID = P.CategoryID
WHERE C.Name = 'Grains';
```

RESULT:

	NAME	Surcharge
1	Dobrudzha 700g	0.50
2	Dobrudzha 800g	0.40
3	Spaghetti	0.30

```
--Изведете какви продукти са доставени на 21 декември 2019,
--чието количество надхвърля 10 кг
SELECT P.Name as ProductName, Quantity, Type
FROM
Products P JOIN DeliveryProducts DP ON P.ProductID = DP.ProductID
JOIN Deliveries D ON DP.DeliveryID = D.DeliveryID
WHERE DeliveryDate = '2019-12-21' AND
Quantity > 10 AND Type = 'kg';
```

RESULT:

	ProductName	Quantity	Type
1	Cucumbers	15	kg
2	Golden Superb Apple	20	kg

Групиране и агрегация

```
--Изведете броя на продуктите във всяка категория,  
--заедно с нейното име.  
SELECT c.Name AS [Category], COUNT(ProductID) AS [ProductCount]  
FROM Products p  
JOIN Categories c ON p.CategoryID = c.CategoryID  
GROUP BY c.Name
```

RESULT:

	Category	ProductCount
1	Beer, Wine and Alcohol drinks	2
2	Coffee, tea, cocoa	2
3	Dairy	1
4	Drinks	3
5	Grains	3
6	Nuts	1
7	Sugar, chocolate, honey products	2
8	Vegetables and Fruits	4

```
--Изведете името на клиентите и сумата, която са заплатили за конкретната покупка.  
SELECT Name, SUM(OverallPrice) AS [OverallPrice] FROM  
(SELECT ps.SaleID, FirstName+' '+ LastName AS [Name], (Quantity*SellPrice) AS [OverallPrice]  
FROM ProductsSales ps  
JOIN Sales s ON ps.SaleID = s.SaleID  
JOIN Customers c ON s.CustomerID = c.CustomerID) t  
GROUP BY t.SaleID, Name
```

RESULT:

	Name	OverallPrice
1	Liliq Dimova	9.30000
2	Kostadin Dimitrov	1.20000
3	Margarita Kireva	5.40000

```
--Изведете общата сума на приходите от всички покупки  
SELECT SUM(Quantity*SellPrice) AS [Income]  
FROM ProductsSales
```

RESULT:

	Income
1	52.98000

Индекси

```
CREATE INDEX idx_productPrice
ON Products(Price)

CREATE INDEX idx_productName
ON Products(Name)

CREATE INDEX idx_customerCompanyID
ON Customers(CustomerID)

CREATE INDEX idx_customerCompanyName
ON Customers(CompanyName)

CREATE INDEX idx_deliveryDate
ON Deliveries(DeliveryDate)
```

- Търси продуктите с определена цена
- Търси се продукта по име
- При въвеждане на булстат в where клаузата
- Търси се и по име, не само по булстат
- Търси на коя дата е доставено нещо

Изгледи

```
CREATE VIEW DailyIncome AS
SELECT CAST(s.Date AS DATE) AS [Date], SUM(ps.Quantity*ps.SellPrice) AS [Income]
FROM ProductsSales AS ps
JOIN Sales AS s ON ps.SaleID = s.SaleID
GROUP BY CAST(s.Date AS DATE)
```

RESULT:

	Date	Income
1	2019-02-01	5.40000
2	2020-01-15	15.78000
3	2020-02-03	9.30000
4	2020-03-10	22.50000

```
CREATE VIEW SoldProductsCounts AS
SELECT ProductID, SUM(Quantity) AS [SoldQuantity]
FROM ProductsSales
GROUP BY ProductID
```

RESULT:

	ProductID	SoldQuantity
1	101	4.000
2	102	3.000
3	103	1.500
4	104	5.000
5	105	2.000
6	106	1.800
7	107	1.000
8	108	2.000

```
CREATE VIEW SalesInfo AS
SELECT t.SaleID, t.OverallPrice, s.Date FROM
(SELECT ps.SaleID, SUM(Quantity*SellPrice) AS [OverallPrice]
FROM ProductsSales ps
GROUP BY ps.SaleID) t
JOIN Sales s ON t.SaleID = s.SaleID
```

RESULT:

	SaleID	OverallPrice	Date
1	1	9.30000	2020-02-03 11:25:09.000
2	2	1.20000	2020-03-10 10:45:02.000
3	3	12.28000	2020-01-15 18:15:30.000
4	4	3.50000	2020-01-15 16:36:32.000
5	5	21.30000	2020-03-10 15:11:37.000
6	6	5.40000	2019-02-01 11:15:47.000

Тригери

```
ALTER TABLE Staff ADD IsEmployed VARCHAR (5) NOT NULL DEFAULT 'True';
```

```
CREATE TRIGGER Staff_deleteStaff ON Staff
INSTEAD OF DELETE
AS
BEGIN
    DECLARE @StaffID INT
    SELECT @StaffID = deleted.StaffID FROM deleted
    UPDATE Staff
    SET IsEmployed = 'False'
    WHERE StaffID = @StaffID
END
```

EXAMPLE:

```
DELETE FROM Staff WHERE StaffID = 8841
SELECT* FROM Staff
```

RESULT:

	StaffID	FirstName	LastName	IsEmployed
1	2530	Maya	Boradzhieva	True
2	2810	Lyubka	Angelinina	True
3	6240	Boris	Todorov	True
4	6569	Nikoleta	Valchinova	True
5	8841	Ivan	Popov	False

- Тригер, който се задейства при опит за изтриване на запис от таблицата Staff. Вместо това полето isEmployed се променя на False.

```

ALTER TABLE Products ADD Operation VARCHAR (30) NOT NULL DEFAULT 'INSERTED';

CREATE TRIGGER Products_delete_audits_trigger ON Products
INSTEAD OF DELETE AS
BEGIN
    DECLARE @Now AS DATETIME = GetDate()
    INSERT INTO Products_update_audits
    SELECT ProductID, Name, Price, CategoryID, Type, @Now
    FROM deleted
    UPDATE Products
    SET Operation = 'Deleted' WHERE ProductID = (SELECT deleted.ProductID FROM deleted)
END

```

EXAMPLE:

```

DELETE FROM Products
WHERE ProductID = 101
SELECT* FROM Products_update_audits
SELECT *FROM Products

```

RESULT:

	ProductID	Name	Price	CategoryID	Type	UPDATED_AT
1	101	Milk Vereya 1L	2.10	5	piece	2020-05-11 20:37:25.793

	ProductID	Name	Price	CategoryID	Type	Operation
1	101	Milk Vereya 1L	2.10	5	piece	Deleted
2	102	Dobrudzha 700g	1.20	1	piece	INSERTED
3	103	Red Tomatoes	3.00	3	kg	INSERTED
4	104	Water Devin 500ml	0.80	2	piece	INSERTED
5	105	Red Wine Cycle 750ml	7.90	11	piece	INSERTED
6	106	Cucumbers	3.60	3	kg	INSERTED
7	107	Nescafe Classic Crema 100g	6.30	10	piece	INSERTED
8	108	Mirazhki	2.30	8	piece	INSERTED
9	109	Dobrudzha 800g	1.30	1	piece	INSERTED
10	110	Pink Tomatoes	4.00	3	kg	INSERTED
11	111	Water Devin 750ml	1.10	2	piece	INSERTED
12	112	Water Devin 1l	1.50	2	piece	INSERTED
13	113	White Wine Cycle 750ml	7.90	11	piece	INSERTED
14	114	Nescafe Classic 500g	8.50	10	piece	INSERTED

- Тригери, които се задействат при промяна на цената на някой от продуктите. В нова таблица се запазва старата информация, както и датата на промяна.

```

ALTER TABLE Staff ADD IsEmployed VARCHAR (5) NOT NULL DEFAULT 'True';

```

```

CREATE TRIGGER Staff_deleteStaff ON Staff
INSTEAD OF DELETE
AS
BEGIN

```

```

    DECLARE @StaffID INT
    SELECT @StaffID = deleted.StaffID FROM deleted
    UPDATE Staff
    SET IsEmployed = 'False'
    WHERE StaffID = @StaffID

```

END

EXAMPLE:

```

DELETE FROM Staff WHERE StaffID = 8841
SELECT* FROM Staff

```

RESULT:

	StaffID	FirstName	LastName	IsEmployed
1	2530	Maya	Boradzhieva	True
2	2810	Lyubka	Angelinina	True
3	6240	Boris	Todorov	True
4	6569	Nikoleta	Valchinova	True
5	8841	Ivan	Popov	False

```
CREATE TABLE Products_update_audits(  
    ProductID INT PRIMARY KEY NOT NULL,  
    Name VARCHAR(50) NOT NULL,  
    Price DECIMAL(6,2) NOT NULL,  
    CategoryID INT NOT NULL,  
    Type VARCHAR(5) NOT NULL DEFAULT 'piece' CHECK(Type IN ('kg', 'piece')),  
    UPDATED_AT DATETIME NOT NULL,  
);
```

❑ Products_update_audits

- При всеки опит за триене на продукт или промяна на някой от детайлите, се добавя нов запис, който включва старата информация за продукта както и времето, в което е направена промяната