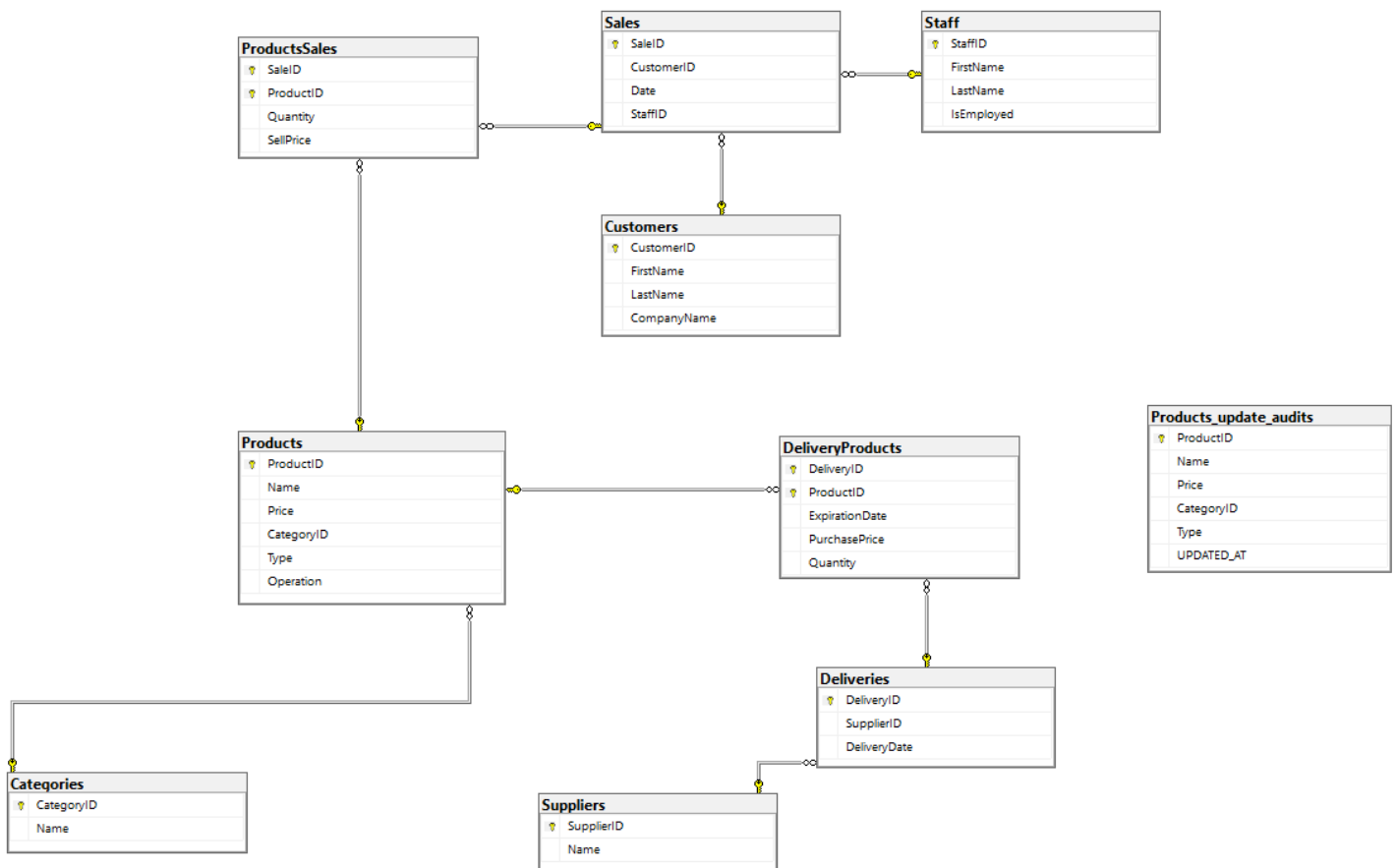


Бази от данни

Курсов проект

Магазин за хранителни стоки

Мая Росенова Бораджиева, ФН 62335, група 5
Николета Стойчева Вълчинова, ФН 62322, група 5
Любка Димитрова Ангелинина, ФН 62342, група 5



1. Създаване на таблицата

```
USE master
GO
if exists (select * from sysdatabases where name='GroceryStore')
    DROP DATABASE GroceryStore
GO
```

```
CREATE DATABASE GroceryStore
GO
USE GroceryStore
GO
```

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

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

```
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)
);
```

```
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 Customers(
```

```

        CustomerID INT PRIMARY KEY NOT NULL,
        FirstName VARCHAR(50) NOT NULL,
        LastName VARCHAR(50) NOT NULL,
        CompanyName VARCHAR(50) NOT NULL
    );

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

    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)
    );

```

2. Добавяне на информация

```

INSERT INTO Categories(CategoryID,Name)
VALUES (1,'Grains'),
(2,'Drinks'),
(3,'Vegetables and Fruits'),
(4,'Fish and fishery products'),
(5,'Dairy'),
(6,'Meat and meat products'),
(7,'Eggs and egg products'),
(8,'Sugar, chocolate, honey products'),
(9,'Nuts'),
(10,'Coffee, tea, cocoa'),

```

(11, 'Beer, Wine and Alcohol drinks');

```
INSERT INTO Suppliers(SupplierID,Name)
VALUES (855965820,'Coca-cola'),
(698523225,'Prestige - 96'),
(555220394,'Krustev'),
(895003551,'Nestle'),
(305263255,'Vereia'),
(200143205,'BioLife'),
(503026002,'Cycle'),
(895202235,'Devin'),
(362555910,'Dobrudzha'),
(556235146,'Rois');
```

```
INSERT INTO Deliveries(DeliveryID, SupplierID, DeliveryDate)
VALUES (200, 305263255, '2019-12-01'),
(201, 362555910, '2019-12-07'),
(202, 200143205, '2019-12-21'),
(203, 895202235, '2020-02-15'),
(204, 503026002, '2020-03-07'),
(205, 305263255, '2020-03-16'),
(206, 362555910, '2020-03-17'),
(207, 556235146, '2020-03-20'),
(208, 895003551, '2020-03-20'),
(209, 555220394, '2020-03-21'),
(210, 503026002, '2020-03-25'),
(211, 698523225, '2020-04-05');
```

```
INSERT INTO Products(ProductID, Name, Price, Type, CategoryID)
VALUES (101,'Milk Vereya 1L', 2.10,'piece', 5 ),
(102,'Dobrudzha 700g', 1.20, 'piece',1 ),
(103,'Red Tomatoes', 3.00,'kg',3),
(104,'Water Devin 500ml', 0.80, 'piece', 2 ),
(105,'Red Wine Cycle 750ml', 7.90, 'piece', 11),
(106,'Cucumbers', 3.60, 'kg', 3),
(107,'Nescafe Classic Crema 100g', 6.30,'piece',10),
(108,'Mirazhki', 2.30,'piece',8),
(109,'Dobrudzha 800g', 1.30, 'piece', 1),
(110,'Pink Tomatoes', 4.00, 'kg', 3),
(111,'Water Devin 750ml', 1.10,'piece',2),
(112,'Water Devin 1l', 1.5, 'piece',2),
(113,'White Wine Cycle 750ml', 7.90,'piece', 11),
(114,'Nescafe Classic 500g', 8.50, 'piece',10),
(115,'Golden Superb Apple', 1.49, 'kg', 3),
(116,'Trayana 350g', 2.10,'piece',8),
(117,'Rois Almond 80g', 3.50, 'piece', 9),
(118,'Spaghetti', 3.50, 'piece', 1);
```

```
INSERT INTO DeliveryProducts(DeliveryID, ProductID, ExpirationDate, PurchasePrice, Quantity)
VALUES (200, 101, '2019-12-20', 1.70, 30),
(201, 102, '2019-12-12', 0.70, 20),
(201, 109, '2019-12-12', 0.90, 15),
(202, 103, '2020-05-01', 2.30, 10),
```

```
(202, 106, '2020-01-06', 2.50, 15),
(202, 110, '2020-01-05', 2.70, 10),
(202, 115, '2020-01-10', 1.20, 20),
(203, 104, '2021-02-15', 0.40, 40),
(203, 111, '2021-02-15', 0.60, 30),
(203, 112, '2021-02-15', 0.70, 30),
(205, 101, '2020-04-02', 1.70, 20),
(206, 102, '2020-03-22', 0.70, 20),
(206, 109, '2020-03-22', 0.90, 10),
(207, 117, '2020-09-20', 2.90, 30),
(208, 107, '2021-11-10', 5.00, 50),
(208, 114, '2021-11-10', 6.90, 50),
(209, 118, '2022-07-05', 3.20, 30),
(211, 108, '2020-06-05', 1.50, 20),
(211, 116, '2020-06-05', 1.50, 20);
```

-- специално за виното - няма срок на годност

```
INSERT INTO DeliveryProducts(DeliveryID, ProductID, PurchasePrice, Quantity)
VALUES (204, 105, 7.50, 20),
(204, 113, 7.50, 20),
(210, 105, 7.50, 30);
```

```
INSERT INTO Staff(StaffID, FirstName, LastName)
VALUES (2530, 'Maya', 'Boradzhieva'),
(6569, 'Nikoleta', 'Valchinova'),
(2810, 'Lyubka', 'Angelinina'),
(6240, 'Boris', 'Todorov'),
(8841, 'Ivan', 'Popov');
```

```
INSERT INTO Customers(CustomerID, FirstName, LastName, CompanyName)
VALUES (965820193, 'Liliq', 'Dimova', 'Cafe VIP'),
(206318002, 'Georgi', 'Ivanov', 'Bulgaria Restaurant'),
(663256985, 'Kostadin', 'Dimitrov', 'Club 35'),
(588203698, 'Nevena', 'Mandeva', 'Mandeva's Company'),
(996325682, 'Mila', 'Iordanova', 'Orpheus Bar & Dinner'),
(320569822, 'Margarita', 'Kireva', 'Magis catering'),
(114451237, 'Elena', 'Krasimirova', 'Monroe Bar and Grill');
```

```
INSERT INTO Sales(SaleID, CustomerID, Date, StaffID)
VALUES (1, 965820193, '2020-02-03 11:25:09 AM', 2530),
(2, 663256985, '2020-03-10 10:45:02 AM', 2530),
(6, 320569822, '2019-02-01 11:15:47 AM', 8841);
```

```
INSERT INTO Sales(SaleID, Date, StaffID)
VALUES (3, '2020-01-15 06:15:30 PM', 2810),
(4, '2020-01-15 04:36:32 PM', 2810),
(5, '2020-03-10 03:11:37 PM', 2530);
```

```
INSERT INTO ProductsSales(SaleID, ProductID, Quantity, SellPrice)
VALUES (1, 101, 2, 2.10),
(1, 103, 0.5, 3.00),
(1, 106, 1, 3.60),
```

```
(2,102, 1, 1.20),  
(3,106, 0.8, 3.60),  
(3,108, 2, 2.30),  
(3,102, 2, 1.20),  
(3,104, 3, 0.80),  
(4,103, 1, 3.50),  
(5,105, 2, 7.50),  
(5,107, 1, 6.30),  
(6,101, 2, 2.10),  
(6,104, 2, 0.60);
```

3. Заявки

Прости заявки

--1. Заявка, която извежда ИМЕ и ЦЕНА на продуктите, които са с цена над 4.00лв и са от тип 'piece' (на брой)

```
SELECT Name, Price  
  
FROM Products  
  
WHERE Price>4.00 AND Type = 'piece'
```

--2. Заявка, която извежда ИМЕ и ЦЕНА на продуктите, които започват с буквата 'N'

```
SELECT Name, Price  
  
FROM Products  
  
WHERE Name LIKE 'N%'
```

--3. 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
```

--4. Заявка, която извежда номер на доставка и номер на снабдител на продуктите, които имат дата на доставка 20.03.2020

```
SELECT DeliveryID, SupplierID  
  
FROM Deliveries  
  
WHERE DeliveryDate='2020-03-20 '
```

--5. Заявка, която извежда номер на ПРОДУКТА и срок на годност на доставените продукти, чието количество е по-малко или равно на 20

```
SELECT ProductID, ExpirationDate
```

```
FROM DeliveryProducts
```

```
WHERE Quantity <=20
```

```
ORDER BY ProductID
```

--6. 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'
```

Заявки върху две и повече релации

--1. Заявка, която извежда : номер на клиент(номер на фирма, която е клиент на магазина), дата, име и фамилия на персонал т.е. съответната информация за продажби, които са били ким фирми и са били в смяната на 'Maya'

```
SELECT CustomerID, Date, FirstName, LastName
```

```
FROM Sales,Staff
```

```
WHERE CustomerID IS NOT NULL AND FirstName LIKE 'Maya'
```

```
AND Sales.StaffID=Staff.StaffID
```

--2. Заявка, която извежда номер на продукти, които не са били включени в никоя продажба

```
SELECT ProductID FROM Products
```

```
EXCEPT
```

```
((SELECT ProductID FROM DeliveryProducts)
```

```
INTERSECT
```

```
(SELECT ProductID FROM ProductsSales))
```

--3. 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
```

--4. Заявка, която извежда името на снадители, от които нямаме доставка

```
SELECT Name
FROM Suppliers,
((SELECT SupplierID FROM Suppliers)
EXCEPT
(SELECT SupplierID FROM Deliveries)) AS Table1
WHERE Suppliers.SupplierID = Table1.SupplierID
```

--5. Заявка, която извежда име и фамилия на собственик на компания, компания(името на колоната да се казва 'Company') на тези компании, които са направили покупки

```
SELECT FirstName, LastName, CompanyName AS Company
FROM Customers, (SELECT * FROM Sales
WHERE Sales.CustomerID IS NOT NULL) AS CS
WHERE CS.CustomerID = Customers.CustomerID
```

SUBQUERIES

----TASK 1

--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');
```

----TASK 2

--Изведете ID-тата на всички продажби, направени от служител

--с първо име Maya

SELECT SaleID

FROM Sales

WHERE StaffID IN (SELECT StaffID

FROM Staff

WHERE FirstName = 'Maya');

----TASK 3

--Изведете имената и цените на всички продукти от

--категорията 'Vegetables and Fruits'

SELECT Name, Price

FROM Products

WHERE CategoryID IN (SELECT CategoryID

FROM Categories

WHERE Name = 'Vegetables and Fruits');

--TASK 4

--Изведете категорията на продуктите,

--които присъстват поне 2 пъти в покупките

--на клиент

SELECT Name

FROM Categories

WHERE CategoryID IN (SELECT CategoryID

FROM Products

WHERE ProductID IN (SELECT ProductID

FROM ProductsSales

WHERE Quantity >=

2));

--TASK 5

--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 %')
```

JOINS

--TASK 1

--Изведете всички продукти, доставени от фирма 'Nestle'

```
SELECT *
FROM
Products P JOIN DeliveryProducts DP ON P.ProductID = DP.ProductID
JOIN Deliveries D ON DP.DeliveryID = D.DeliveryID
JOIN Suppliers S ON S.SupplierID = D.SupplierID
WHERE S.Name = 'Nestle';
```

--TASK 2

--Изведете датата на продажба на всички артикули

--от категорията 'Coffee, tea, cocoa'

```
SELECT Date AS SellDate
FROM
Products P JOIN ProductsSales PS ON P.ProductID = PS.ProductID
JOIN Sales S ON S.SaleID = PS.SaleID
JOIN Categories C ON C.CategoryID = P.CategoryID
WHERE C.Name = 'Coffee, tea, cocoa';
```

--TASK 3

--Изведете името на служителя и времето, в което е извършил своите продажби

--(кога е бил на касата)

```
SELECT S.StaffID, FirstName, LastName, Date
```

FROM

Sales S JOIN Staff ST ON S.StaffID = ST.StaffID;

--TASK 4

--Изведете информация за това кои са

--доставчиците на отделните категории продукти

SELECT DISTINCT S.Name AS SupplierName, C.Name AS CategoryName

FROM

Products P JOIN DeliveryProducts DP ON P.ProductID = DP.ProductID

JOIN Deliveries D ON D.DeliveryID = DP.DeliveryID

JOIN Suppliers S ON D.SupplierID = S.SupplierID

JOIN Categories C ON C.CategoryID = P.CategoryID;

--TASK 5

--Изведете датата на доставка на всеки продукт

SELECT P.Name AS ProductName, DeliveryDate

FROM

Products P JOIN DeliveryProducts DP ON P.ProductID = DP.ProductID

JOIN Deliveries D ON D.DeliveryID = DP.DeliveryID;

--TASK 6

--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';

--TASK 7

--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';

GROUPING AND AGGREGATION

--1. Изведете общата сума на приходите от всички покупки

```
SELECT SUM(Quantity*SellPrice) AS [Income] FROM ProductsSales
```

--2. Изведете печалбата за всеки ден.

```
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)
```

--3 Изведете средната печалба от всички продажби.

```
SELECT AVG(Income) AS [AverageIncome] FROM DailyIncome
```

--4. Изведете кой е най-скъпия продукт.

```
SELECT MAX(Price) FROM Products
```

--5 Изведете общия брой на всички продукти.

```
SELECT COUNT(ProductID) FROM Products
```

--6. Изведете деня с най-ниска печалба.

```
SELECT Date FROM DailyIncome
```

```
WHERE Income = (SELECT MIN(Income) FROM DailyIncome)
```

--7. Изведете най-евтината продажба за всеки ден.

```
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
```

```
SELECT CAST(Date AS DATE) AS [Date], MIN(OverallPrice) AS [MinIncomePerDay] FROM SalesInfo
GROUP BY CAST(Date AS DATE)
```

--8. Изведете на всяка дата по колко различни продукти са доставени.

```
SELECT CAST(DeliveryDate AS DATE) AS[Date], COUNT(p.ProductID) AS[ProductsCount] FROM
Deliveries d
JOIN DeliveryProducts dp ON d.DeliveryID = dp.DeliveryID
JOIN Products p ON dp.ProductID = p.ProductID
GROUP BY CAST(DeliveryDate AS DATE)
```

--9. Изведете името на най-продавания продукт и броя продадените артикули от този продукт.

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

```
SELECT Name, SoldQuantity FROM SoldProductsCounts sp
JOIN Products p ON sp.ProductID = p.ProductID
WHERE SoldQuantity = (SELECT MAX(SoldQuantity) FROM SoldProductsCounts)
```

--10. 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
```

--11. 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

ИЗГЛЕДИ

--Изведете най-евтината продажба за всеки ден.

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

-- Изведете името на най-продавания продукт и броя продадените артикули от този продукт.

CREATE VIEW SoldProductsCounts AS

SELECT ProductID,SUM(Quantity) AS [SoldQuantity] FROM ProductsSales

GROUP BY ProductID

--Изведете печалбата за всеки ден.

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)

ИНДЕКСИ

--non-clustered, because all tables have primary key

--INDEX 1

--Ако искаме да видим продуктите с определена цена

CREATE INDEX idx_productPrice

ON Products(Price)

--INDEX 2

--Търси се продукта по име

```
CREATE INDEX idx_productName
```

```
ON Products(Name)
```

```
--INDEX 3
```

```
--При въвеждане на булстат в where клаузата
```

```
CREATE INDEX idx_customerCompanyID
```

```
ON Customers(CustomerID)
```

```
--INDEX 4
```

```
--Търси се и по име, не само по булстат
```

```
CREATE INDEX idx_customerCompanyName
```

```
ON Customers(CompanyName)
```

```
--INDEX 5
```

```
--На коя дата е доставено нещо се търси често
```

```
CREATE INDEX idx_deliveryDate
```

```
ON Deliveries(DeliveryDate)
```

ТРИГЕРИ

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

```
--1
```

```
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
```

```
--2
```

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

```
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,
    );

```

```

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

```

```

--3

```

```

CREATE TRIGGER Products_update_audits_trigger ON Products
FOR UPDATE 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 = 'Updated' WHERE ProductID = (SELECT deleted.ProductID FROM
deleted)

```

```

END

```


Задача	Фак. номер
1. Създаване на схемите и релациите	62322
2. Добавяне на съдържание	62335
3. Примерни прости заявки и заявки върху две и повече релации	62342
4. Примери с подзаявки	62335
5. Примери със съединения	62335
6. Примери с групиране и агрегация	62322
7. Примери с ограничения	62342
8. Примери с изгледи и индекси	62335
9. Примери с тригери	62322
10. Подготовка на презентацията	62342