База от данни за хранителен магазин

Задача	Фак. №
1. Създаване на схемите и релациите	62250
2. Добавяне на съдържание	62262
3. Примерни прости заявки и заявки върху две и повече релации	62264
4. Примери с подзаявки	62306
5. Примери със съединения	62250
6. Примери с групиране и аграгация	62262
7. Примери с ограничения	62250
8. Примери с изгледи и индекси	62306
9. Примери с тригери	всички
10. Подготовка на презентацията	всички

Членове на екипа:

Теодора Иванова- 62250

Благовест Конецовски- 62262

Иванка Паунова- 62306

Румен Минчев- 62264

Описание на проекта и предметната област

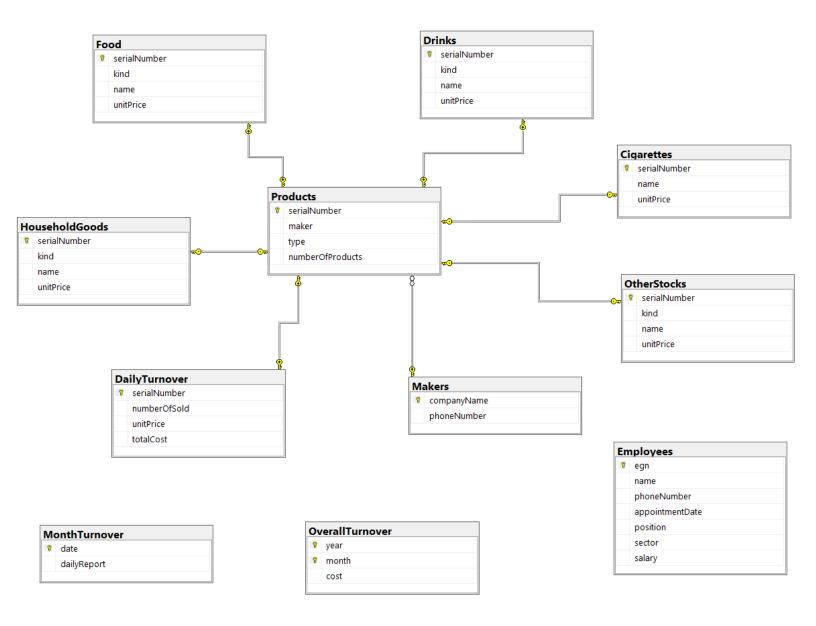
Нашият екип избра да разработи информационна система, обслужваща магазин за

хранителни стоки.

В нея ще се съхранява информация за продукти, които ще се разделят на храни, напитки и други видове стоки, и за тяхната наличност, информация за доставчици, информация за продадени стоки за ден, месец и по години и информация за служителите, работещи в магазина.

В нея ще могат да се извършват следните операции:

- -добавяне на информация за продукти, производители и служители;
- -модифициране на информация за продукти, производители, служители
- търсене на информация за продукти, производители и служители;
- премахване на информация за продукти, производители и служители;
- изчисляване на дневен и месечен оборот;
- история на оборота по години (в допълнение, може да се наблюдава оборотът по месеци за съответната година);



----- Прости заявки

- -- 1.Напишете заявка, която извежда серийния номер, името и цената на храната от тип месо или
- -- мляко, където цената е по-малка от 10.

SELECT serialNumber, name, unitPrice

FROM Food

WHERE (kind='meat' OR kind='milky') AND unitPrice<10

	serialNumber	name	unitPrice
1	151732349	chicken	6
2	654816514	white cheese	9,8
3	654816515	yoghurt 500gr	0,95

-- 2.Напишете заявка, която извежда цената и имената на храните, чийто име започва с кроасан.

SELECT name, unitPrice

FROM Food

WHERE name LIKE ('croissant %')

	name	unitPrice
1	croissant "Chocolate"	0,9
2	croissant cream "Brule"	0,9
3	croissant cream "Cherry"	0,9

-- 3.Напишете заявка, която извежда серийния номер, име и цена на напитките, които не съдържат -- алкохол.

SELECT serialNumber, name, unitPrice

FROM Drinks

WHERE kind <> 'alcoholic'

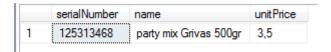
	serialNumber	name	unitPrice
1	151637234	Water Devin 1,5 liter	1,6
2	151637235	Water Devin 0,5 liter	0,7
3	151637236	Water Devin 10 liter	3,2
4	156186573	Monster "Classic"	1,99
5	156186574	Monster "Ultra red"	1,99
6	156186575	Monster "Ultra blue"	1,99
7	156186576	Monster "Zero"	1,99
8	481623946	Cappy juice "Apple"	2
9	481623947	Cappy juice "Orange"	2
10	481623948	Cappy juice "Banana"	2
11	809412896	Coca Cola 0,500 liter	1,2
12	809412897	Coca Cola 2 liter	2,5
13	809412898	Fanta 2 liter	2,4
14	809412899	Sprite 2 liter	2,3

- -- 4.Напишете заявка, която извежда серийния номер, името и цената на ядките, в чийто име се
- -- съдържа 'mix'.

SELECT serialNumber, name, unitPrice

FROM OtherStocks

WHERE kind='nuts' and name like '%mix%'

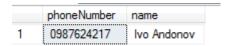


- -- 5. Напишете заявка, която извежда телефонните номера и имената на тези служители, които са
- -- CEO или отговарят за млеката, но не са на позиция test.

SELECT phoneNumber, name

FROM Employees

WHERE sector='CEO' OR (sector='milky' AND position<>'test')



------ Заявки върху две и повече релации -------

- -- 1. Напишете заявка, която извежда серийните номера и цената на алкохолните напитки, които са
- -- по-малко от 10 в наличност.

 ${\color{red} \textbf{SELECT} product. serial Number, unit Price}$

FROM Products, Drinks

WHERE kind='alcoholic' AND Products.serialNumber = Drinks.serialNumber

AND Products.numberOfProducts < 10

	serialNumber	unitPrice
1	507207575	12
2	507207576	9,99
3	715017555	28,99
4	715017556	21,99
5	916641476	39,99
6	916641477	46

-- 2. Напишете заявка, която извежда цената, името и производителят на всички напитки със цена

-- по-висока от цената на 'Monster "Zero"'.

SELECT Drinks.name, Drinks.unitPrice, maker

FROM Products, Drinks, Drinks as ProductOfMonsterZero

WHERE Products.serialNumber = Drinks.serialNumber

AND ProductOfMonsterZero.name = 'Monster "Zero"'

AND Drinks.unitPrice > ProductOfMonsterZero.unitPrice

	name	unitPrice	maker
1	Water Devin 10 liter	3,2	Devin
2	Cappy juice "Apple"	2	Сарру
3	Cappy juice "Orange"	2	Сарру
4	Cappy juice "Banana"	2	Сарру
5	Whiskey Savoy 1 liter	12	Savoy
6	Vodka Savoy 1 liter	9,99	Savoy
7	Whiskey Jameson 1 liter	28,99	Jameson
8	Whiskey Jameson 700 milliliter	21,99	Jameson
9	Coca Cola 2 liter	2,5	Coca Cola Bulgaria
10	Fanta 2 liter	2,4	Coca Cola Bulgaria
11	Sprite 2 liter	2,3	Coca Cola Bulgaria
12	Whiskey Jack Daniels 700 milliliter	39,99	Jack Daniels
13	Whiskey Jack Daniels 1 liter	46	Jack Daniels

- -- 3. Напишете заявка, която извежда името на компанията, телефонния й номер, серийния номер
- -- на храната и цената, която се образува като произведение на броя на наличната стока и цената --
- -- за единична бройка, ако това произведение надхвърля 100.

SELECT companyName, phoneNumber, Products.serialNumber,

numberOfProducts*unitPrice AS totalPrice

FROM Makers, Products, Food

WHERE Makers.companyName = Products.maker

AND Foods.serialNumber = Products.serialNumber

AND numberOfProducts*unitPrice > 100

	companyName	phoneNumber	serialNumber	totalPrice
1	Bozmov	0841594610	151732347	330
2	Bozmov	0841594610	151732348	382,5
3	Bozmov	0841594610	151732349	183
4	Milk Heaven	084636242	654816513	122
5	Milk Heaven	084636242	654816514	147

-- 4. Напишете заявка, която извежда името, номера на производителя и бройките в наличност на

-- млечните продукти.

SELECT companyName, phoneNumber, numberOfProducts

FROM Makers, Products, Food

WHERE Makers.companyName = Products.maker

AND Food.serialNumber = Products.serialNumber

AND kind='milky'

	companyName	phoneNumber	numberOfProducts
1	Milk Heaven	084636242	10
2	Milk Heaven	084636242	15
3	Milk Heaven	084636242	40

-- 5. Напишете заявка, която извежда общата стойност на агнешкото месо с производител 'Bozmov'.

SELECT maker, name, numberOfProducts*unitPrice as totalValue

FROM Products, Food

WHERE maker='Bozmov'

AND name ='lamb'

AND Products.serialNumber = Food.serialNumber

	maker	name	totalValue
1	Bozmov	lamb	382,5

----- Подзаявки -----

- -- 1. Да се изведат имената на производителите, които имат налични продукти от тип
- -- 'householdGoods' в хранителния магазина.

SELECT companyName

FROM Makers

WHERE companyName IN (SELECT maker FROM Products WHERE type LIKE 'HouseholdGoods');

	companyName
1	Emeka
2	Fairy
3	Household Goods Bulgaria
4	Semana

- -- 2. Да се изведат името, типът и единичната цена на напитките,
- -- чиято цена е по-висока от единичните цени на хранителните продукти.

SELECT name, kind, unitPrice

FROM Drinks

WHERE unitPrice > ALL (SELECT unitPrice FROM Food);

	name	kind	unitPrice
1	Whiskey Jameson 11	alcoholic	28,99
2	Whiskey Jameson 700 ml	alcoholic	21,99
3	Whiskey Jack Daniels 700 ml	alcoholic	39,99
4	Whiskey Jack Daniels 11	alcoholic	46

-- 3. Да се изведат името и цената на цигарите с най-висока цена.

SELECT name, unitPrice

FROM Cigarettes

WHERE unitPrice = (SELECT MAX(unitPrice) FROM Cigarettes);



- -- 4. Да се изведе името и телефонният номер на производителите, чиито продукти са
- -- най-продавани за деня.

I вариант:



II вариант:

151732347

20

beaf

16,5

```
SELECT companyName, phoneNumber
FROM Makers
WHERE companyName IN (
                       SELECT maker
                       FROM Products
                       WHERE serialNumber IN (
                                               SELECT serialNumber
                                               FROM DailyTurnover
                                               WHERE numberOfSold = (SELECT MAX(numberOfSold)
                                                                       FROM DailyTurnover)
                                               )
                       );
      companyName
                     phoneNumber
      Davidoff
                     0234262727
-- 5. Да се изведе серийният номер, името, наличността и единичната цена на месото, което е с
-- най-малка наличност в магазина.
SELECT f.serialNumber, f.name, p.numberOfProducts, f.unitPrice
FROM Products p
JOIN Food f ON p.serialNumber = f.serialNumber
WHERE p.serialNumber IN (
                       SELECT serialNumber
                       FROM Food
                       WHERE kind LIKE 'meat')
AND p.numberOfProducts = (
                         SELECT MIN(numberOfProducts)
                          FROM Products
                         WHERE serialNumber IN (SELECT serialNumber FROM Food WHERE kind LIKE 'meat')
                       );
      serialNumber
                        numberOfProducts
                                        unit Price
                 name
```

- -- 6. Да се изведе името и цената на алкохолната напитка, с цена по-ниска от тази на
- -- 'Whiskey Jack Daniels 700 ml' и в разфасовка от 1 литър.

SELECT name, unitPrice

FROM Drinks

WHERE kind LIKE 'alcoholic'

AND name LIKE '%1 I'

AND unitPrice < (SELECT unitPrice FROM Drinks WHERE name LIKE 'Whiskey Jack Daniels 700 ml');

	name	unitPrice
1	Whiskey Savoy 11	12
2	Vodka Savoy 11	9,99
3	Whiskey Jameson 11	28,99

- -- 7. Да се изведе серийният номер, името, типът, цената и наличността на продукта от тип храни
- -- или напитки с най-висока цена.

```
SELECT TOP 1 ^{\ast}
```

FROM (

SELECT name, unitPrice FROM Food

UNION ALL

SELECT name, unitPrice FROM Drinks

) d

ORDER BY d.unitPrice DESC;

	name	unitPrice
1	Whiskey Jack Daniels 11	46

------ Съединения ------

- -- 1. Да се изведат телефонните номера на тези компании (различните), които произвеждат
- -- храни и от чиито продукти са налични поне 15 броя в хранителния магазин.

SELECT DISTINCT phoneNumber AS 'company phone number'

FROM Makers

JOIN Products ON companyName = maker

WHERE type = 'Food' AND numberOfProducts >= 15

	company phone number
1	024234251
2	0841594610
3	084636242

- -- 2. Да се изведат личните данни (ЕГН, име, телефонен номер) и позицията, която заемат в
- -- магазина, на служителите, за които няма налични продукти в сектора, за който отговарят.
- -- Да се сортира резултата по имената на служителите.

SELECT egn, name, phoneNumber, position

FROM Employees

LEFT JOIN Products ON sector = type

WHERE serialNumber IS NULL

ORDER BY name

	egn	name	phoneNumber	position
1	123412892	Ivo Andonov	0987624217	main
2	123452898	Martin Ivanov	0957654217	main
3	137463145	Petq Goshova	0285414619	test
4	515161616	Petur Peshov	0281414619	cashier

- -- 3. Да се изведат серийният номер, името, производителят, броят продадени
- -- продукти и единичната цена на всички продукти от категория 'householdGoods',
- -- за които броят на продадените продукти е не повече от 10% от наличните в магазина
- -- продукти.

SELECT Products.serialNumber, name AS 'name of product', maker, numberOfSold, householdGoods.unitPrice FROM Products

JOIN DailyTurnover ON Products.serialNumber = DailyTurnover.serialNumber JOIN HouseholdGoods ON Products.serialNumber = HouseholdGoods.serialNumber WHERE numberOfSold <= 0.1*numberOfProducts

	serialNumber	name of product	maker	numberOfSold	unitPrice
1	762755342	dishwashing detergent "Mint" Fairy	Fairy	1	5

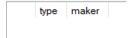
- -- 4. Време е да се поръча стока за магазина. За целта трябва да се проведат
- -- телефонни разговори с всички производители, за да се извършат необходимите поръчки.
- -- Да се напише заявка, която извежда различните типове на продуктите (храни, напитки и т.н.) и
- -- техните производители, за които координатите на производителите липсват в базата от данни на
- -- магазина и следователно е невъзможно те да бъдат поръчани.

SELECT DISTINCT(type), maker

FROM Products p

LEFT JOIN Makers m ON p.maker = m.companyName

WHERE companyName IS NULL



- -- 5. Да се изведат имената на производителите, имената на продуктите и единичната им цена за
- -- тези продукти от тип 'otherStocks', чиято единична цена е най-много средната единична цена за
- -- всички продукти от този тип.

SELECT maker, name AS 'name of product', unitPrice

FROM OtherStocks os

JOIN Products p ON os.serialNumber = p.serialNumber

WHERE unitPrice <= (SELECT AVG(unitPrice)

FROM OtherStocks)

	maker	name of product	unitPrice
1	Grivas	seeds Grivas 500 gr	2,1
2	Grivas	beer peanuts Grivas 500 gr	2,1
3	Orbit	gums "Mint" Orbit 10 gr	1
4	Orbit	gums "Double Mint" Orbit 10 gr	1

------ Групиране

- -- 1.Групиране по типа на продуктите. Извеждане на типа и броя продукти, които магазина
- -- предлага от конкретния тип.

SELECT type, COUNT(type) AS numberOfProducts

FROM Products

GROUP BY type

	type	numberOfProducts
1	Cigarettes	7
2	Drinks	20
3	Food	9
4	HouseholdGoods	6
5	OtherStocks	8

- -- 2.Групиране по сектора на работа на служителите. Извеждане на сектора и преброяване на
- -- хората, които работят в дадения сектор.

SELECT sector, COUNT(sector) AS numberOfPeople

FROM Employees

GROUP BY sector

	sector	numberOfPeople
1	CEO	1
2	checkout	1
3	drinks	2
4	meat	1
5	milky	1

-- 3.Групиране на приходите по година. Сумиране на приходите по година.

SELECT year, SUM(cost) AS money

FROM OverallTurnover

GROUP BY year

	year	money
1	2019	19000
2	2020	31509,3

-- 4.Групиране по стойността на заплатата на служителите.Преброяване на хората, получаващи

-- съответната заплата.

SELECT salary, COUNT(salary) AS number Of People

FROM Employees

GROUP BY salary

	salary	numberOfPeople
1	1000	1
2	1234	5

-- 5.Групириране по телефония номер на производителите, за да се направи справка, колко

-- продукти са поръчани през всеки един от номерата.

SELECT phoneNumber, COUNT (phoneNumber) AS number Of Products FROM Makers JOIN Products ON Makers.companyName=Products.maker GROUP BY phoneNumber

	phoneNumber	numberOfProducts
1	016491643	2
2	0192412643	2
3	0234262727	4
4	024234251	3
5	0246924765	2
6	0249632946	3
7	0254924524	4
8	026429549	2
9	0286412845	4
10	029364926	1
11	0296242426	2
12	031752854	2
13	0435353411	2
14	0492649264	1
15	053965399	4
16	062954252	2
17	0841594610	3
18	084636242	3
19	0856723123	3
20	094769476	1

------ Агрегация -----

- -- 1.Избиране на продукта с най-ниска цена от тип Храни (ако има два продукта с равна и
- -- минмална цена, ще изведе и двата продукта).

SELECT name

FROM Food

WHERE unitPrice = (SELECT MIN(unitPrice) FROM Food)



- -- 2.Избиране на продукта с най-висока цена от тип Храни (ако има два продукта с равна и
- -- минмална цена, ще изведе и двата продукта).

SELECT name

FROM Food

WHERE unitPrice = (SELECT MAX(unitPrice) FROM Food)



-- 3.Изчисляване на средната цена на всички млечни продукти.

SELECT 'milky' AS type, AVG(unitPrice) AS averagePrice

FROM Food

WHERE kind = 'milky'



-- 4. Изчисляване на средната цена на всички цигари на марка Davidoff.

SELECT 'Davidoff' AS maker, AVG(unitPrice) AS averagePrice

FROM Cigarettes

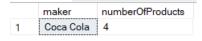
WHERE name LIKE '%Davidoff%'



-- 5.Преброяване на всички продукти, които Coca Cola доставя на магазина.

SELECT 'Coca Cola' AS maker, COUNT(maker) AS numberOfProducts

FROM Products WHERE maker LIKE '%Coca Cola%'



-- 6.Извеждане на броя на различните видове кроасани, които магазина предлага.

SELECT 'Croissant' AS kind, COUNT(name) AS number Of Products

FROM Food

WHERE name LIKE '%Croissant%'



- -- 7.Показване на името на продукта, който има най-малко бройки налични и колко налични
- -- бройки има магазина в момента.

SELECT *

FROM (SELECT name, number Of Products FROM Food JOIN Products ON

Food.serialNumber=Products.serialNumber

UNION ALL

SELECT name, number Of Products FROM Drinks JOIN Products ON

Drinks.serialNumber=Products.serialNumber

UNION ALL

SELECT name, number Of Products FROM Cigarettes JOIN Products ON

Cigarettes.serialNumber=Products.serialNumber

UNION ALL

SELECT name.numberOfProducts FROM OtherStocks JOIN Products ON

OtherStocks.serialNumber=Products.serialNumber

UNION ALL

SELECT name, number Of Products FROM Household Goods JOIN Products ON

HouseholdGoods.serialNumber=Products.serialNumber) innerQuery

WHERE numberOfProducts <= (SELECT MIN(numberOfProducts) FROM Products)

ORDER BY numberOfProducts

	name	numberOfProducts
1	plastic basin 15 l	3

------ Ограничения

-- Primary keys

ALTER TABLE Products ADD CONSTRAINT pk Products PRIMARY KEY (serialNumber);

ALTER TABLE Food ADD CONSTRAINT pk Food PRIMARY KEY (serialNumber);

ALTER TABLE Drinks ADD CONSTRAINT pk_Drinks PRIMARY KEY (serialNumber);

ALTER TABLE Cigarettes ADD CONSTRAINT pk_Cigarettes PRIMARY KEY (serialNumber);

ALTER TABLE OtherStocks ADD CONSTRAINT pk OtherStocs PRIMARY KEY (serialNumber);

ALTER TABLE HouseholdGoods ADD CONSTRAINT pk_HouseholdGoods PRIMARY KEY (serialNumber);

ALTER TABLE DailyTurnover ADD CONSTRAINT pk_DailyTurnover PRIMARY KEY (serialNumber);

ALTER TABLE MonthTurnover ADD CONSTRAINT pk_MonthTurnover PRIMARY KEY (date);

ALTER TABLE OverallTurnover ADD CONSTRAINT pk_OverallTurnover PRIMARY KEY (year,month);

ALTER TABLE Makers ADD CONSTRAINT pk_Makers PRIMARY KEY (companyName);

ALTER TABLE Employees ADD CONSTRAINT pk_Employees PRIMARY KEY (egn);

```
ALTER TABLE Products ADD CONSTRAINT fk Products Makers FOREIGN KEY(maker) REFERENCES
Makers(companyName);
ALTER TABLE Food ADD CONSTRAINT fk Food Products FOREIGN KEY (serialNumber) REFERENCES
Products(serialNumber);
ALTER TABLE Drinks ADD CONSTRAINT fk Drinks Products FOREIGN KEY (serialNumber) REFERENCES
Products(serialNumber);
ALTER TABLE Cigarettes ADD CONSTRAINT fk Cigarettes Products FOREIGN KEY (serialNumber) REFERENCES
Products(serialNumber);
ALTER TABLE OtherStocks ADD CONSTRAINT fk OtherStocks Products FOREIGN KEY (serialNumber) REFERENCES
Products(serialNumber);
ALTER TABLE HouseholdGoods ADD CONSTRAINT fk HouseholdGoods Products FOREIGN KEY (serialNumber)
REFERENCES Products(serialNumber);
ALTER TABLE DailyTurnover ADD CONSTRAINT fk DailyTurnover Products FOREIGN KEY (serialNumber)
REFERENCES Products(serialNumber);
-- Check constraints
-- Products
ALTER TABLE Products ADD CONSTRAINT ck_Products_serialNumber CHECK(LEN(serialNumber) > 0)
ALTER TABLE Products ADD CONSTRAINT ck Products maker CHECK(LEN(maker) > 0)
ALTER TABLE Products ADD CONSTRAINT ck Products type CHECK(LEN(type) > 0)
ALTER TABLE Products ADD CONSTRAINT ck Products numberOfProducts CHECK(numberOfProducts >= 0)
-- Food
ALTER TABLE Food ADD CONSTRAINT ck Food serialNumber CHECK(LEN(serialNumber) > 0)
ALTER TABLE Food ADD CONSTRAINT ck Food kind CHECK(LEN(kind) > 0)
ALTER TABLE Food ADD CONSTRAINT ck_Food_name CHECK(LEN(name) > 0)
ALTER TABLE Food ADD CONSTRAINT ck Food unitPrice CHECK(unitPrice > 0)
-- Drinks
ALTER TABLE Drinks ADD CONSTRAINT ck Drinks serialNumber CHECK(LEN(serialNumber) > 0)
ALTER TABLE Drinks ADD CONSTRAINT ck_Drinks_kind CHECK(LEN(kind) > 0)
ALTER TABLE Drinks ADD CONSTRAINT ck Drinks name CHECK(LEN(name) > 0)
ALTER TABLE Drinks ADD CONSTRAINT ck Drinks unitPrice CHECK(unitPrice > 0)
-- Cigarettes
ALTER TABLE Cigarettes ADD CONSTRAINT ck Cigarettes serialNumber CHECK(LEN(serialNumber) > 0)
ALTER TABLE Cigarettes ADD CONSTRAINT ck_Cigarettes_name CHECK(LEN(name) > 0)
ALTER TABLE Cigarettes ADD CONSTRAINT ck_Cigarettes_unitPrice CHECK(unitPrice > 0)
-- OtherStocks
ALTER TABLE OtherStocks ADD CONSTRAINT ck_OtherStocks_serialNumber CHECK(LEN(serialNumber) > 0)
ALTER TABLE OtherStocks ADD CONSTRAINT ck OtherStocks kind CHECK(LEN(kind) > 0)
ALTER TABLE OtherStocks ADD CONSTRAINT ck OtherStocks name CHECK(LEN(name) > 0)
ALTER TABLE OtherStocks ADD CONSTRAINT ck OtherStocks unitPrice CHECK(unitPrice > 0)
```

-- Foreign keys

```
-- HouseholdGoods
ALTER TABLE HouseholdGoods ADD CONSTRAINT ck_HouseholdGoods_serialNumber
CHECK(LEN(serialNumber) > 0)
ALTER TABLE HouseholdGoods ADD CONSTRAINT ck HouseholdGoods kind CHECK(LEN(kind) > 0)
ALTER TABLE HouseholdGoods ADD CONSTRAINT ck HouseholdGoods name CHECK(LEN(name) > 0)
ALTER TABLE HouseholdGoods ADD CONSTRAINT ck_HouseholdGoods_unitPrice CHECK(unitPrice > 0)
-- MonthTurnover
ALTER TABLE MonthTurnover ADD CONSTRAINT ck MonthTurnover date CHECK(ISDATE(date)=1)
-- Makers
ALTER TABLE Makers ADD CONSTRAINT ck Makers companyName CHECK(LEN(companyName) > 0)
ALTER TABLE Makers ADD CONSTRAINT ck Makers phoneNumber CHECK(LEN(phoneNumber) > 0)
-- Employees
ALTER TABLE Employees ADD CONSTRAINT ck Employees name CHECK(LEN(name) > 0)
ALTER TABLE Employees ADD CONSTRAINT ck_Employees_phoneNumber CHECK(LEN(phoneNumber) > 0)
ALTER TABLE Employees ADD CONSTRAINT ck Employees appointmentDate CHECK(ISDATE(appointmentDate)=1)
ALTER TABLE Employees ADD CONSTRAINT ck Employees position CHECK(LEN(position) > 0)
ALTER TABLE Employees ADD CONSTRAINT ck Employees sector CHECK(LEN(sector) > 0)
ALTER TABLE Employees ADD CONSTRAINT ck Employees salary CHECK(salary > 0)
 ----- Изгледи и индекси
CREATE NONCLUSTERED INDEX ix_Products_Maker
ON Products(maker);
CREATE NONCLUSTERED INDEX ix Products Number
ON Products(numberOfProducts);
-- Сериен номер и цена на всички налични продукти в базата на магазина.
CREATE VIEW ProductsPriceAndNumber (serialNumber, unitPrice) AS
       SELECT serialNumber, unitPrice FROM dbo.Food
       UNION ALL
       SELECT serialNumber, unitPrice FROM dbo.Drinks
       SELECT serialNumber, unitPrice FROM dbo.Cigarettes
       UNION ALL
       SELECT serialNumber, unitPrice FROM dbo.HouseholdGoods
       UNION ALL
       SELECT serialNumber, unitPrice FROM dbo.OtherStocks;
-- Общо оборот за деня.
CREATE VIEW DailyReport AS
       SELECT SUM (numberOfSold * unitPrice) AS totalCostOfTheDay
       FROM DailyTurnover;
```

```
-- Общо оборот за месеца.
CREATE VIEW MonthlyReport AS
       SELECT SUM(dailyReport) AS monthTotalCost
       FROM MonthTurnover;
-- Информация за оборота по години.
CREATE VIEW Annual Turnover AS
       SELECT year, SUM(cost) as annualReport
       FROM OverallTurnover
       GROUP BY year;
-- Датата на последноно добавения запис в MonthTurnover, която ще се използва
-- при създаването на месечния отчет и прехвърлянето му в OverallTurnover.
CREATE VIEW MonthlyReportDate AS
       SELECT TOP 1 date
       FROM MonthTurnover
       ORDER BY date DESC;
------ Тригери ------
-- След продажба на продукт (добавяне на продукт в DailyTurnover) се обновява количеството на
-- съответния продукт (numberOfProducts) в Products.
CREATE TRIGGER tr DecreaseNumberOfProduct
ON dbo.DailyTurnover
AFTER INSERT
AS
BEGIN
  UPDATE Products
  SET numberOfProducts = numberOfProducts - (SELECT numberOfSold FROM inserted)
  WHERE serialNumber = (SELECT serialNumber FROM inserted);
END
GO
-- В таблицата за дневния оборот има данни за продажбите само за текущия ден,
-- като след изтичането му те се прехвърлят в обобщен вид в MonthTurnover и се изтриват от
-- DailyTurnover, за да се изчисти тя за следващия ден. Детайлите около дневния оборот (номер на
-- продукт, брой продадени, единична цена и обща цена) се изтриват, а не се създава история за
-- тях, защото впоследствие те няма да ни бъдат необходими, т.е. няма да се налага да ги
-- възстановяваме.
CREATE TRIGGER tr RemoveDailyInfo
ON dbo.MonthTurnover
AFTER INSERT
AS
BEGIN
  DELETE FROM DailyTurnover
END
GO
```

- -- В таблицата за месечния оборот има данни за само за текущия месец,
- -- като след изтичането му те се прехвърлят в обобщен вид OverallTurnover и се изтриват от
- -- MonthTurnover, за да се изчисти тя за следващия месец. Детайлите около месечния оборот (дата
- -- и оборот за нея) се изтриват, а не се създава история за тях, защото впоследствие
- -- те няма да ни бъдат необходими, т.е. няма да се налага да ги възстановяваме.

CREATE TRIGGER tr_RemoveMonthlyInfo

ON dbo.OverallTurnover

AFTER INSERT

AS

BEGIN

DELETE FROM MonthTurnover

END

GO