



**CSE 3055 – DATABASE SYSTEMS
PROJECT**

Authors:

- 1) 150117005 Kadir ACUN**
- 2) 150117905 Erhan YALNIZ**

CSE3055: Database Systems, Fall 2020

Date Submitted: Jan 01, 2021

NAME OF THE PROJECT

We decided that the name of our Project should be Göçmen Supermarket.

PROJECT DESCRIPTION AND SCOPE

The scope and purpose of our project is to provide all information about supermarket, carry out all works. By this database implementation, we list and hold datas for the employees of supermarket, sections of the supermarket, information about products and brands who supply products (These will be our tables). Furthermore, we design all of these important requirements easier for our customers by this database implementation. Briefly, this program provides appropriate infrastructure data for market owner, it's customers, it's brands and relationship between them.

By using this Project our customer can control statistic and information about each product. By the way, our costumer can decide which way should he choose to improve his brand's financial status. So our customer can determine:

- Which product should be bought from supplier by looking for the consumption ratio by using related table ?
- Which product has stock?
- Which product has passed expiration date?
- Which product sold more than the other product?

By using this Project our customer can reach and update information such as:

- Name, stock and cost of product
- Buy date, sale date and expiration date of product
- Personal information of employees
- Contact information of brands who supply the products

Note: In this iteration, all processed we expect are defined by us and there are some other process as follows:

- Customer's (regular customer) informations
- Payment informations
- Promotion informations that supplied by brands
- Etc..

Our data will come from Göçmen Market which is located in Buca/İzmir and which is a 25-year-old market with 3 separate sales segments.

CUSTOMER INFO

The customer of this Project we decided owner of the Göçmen Market (Buca/İzmir). In this way, the customers of the company that we will create a database for, and the product suppliers will indirectly become our customers.

CSE3055 KODLU DATABASE DERSİ PROJESİ İÇİN DÜKKANIMIZIN İSMİNİN VE VERİLERİNİN KULLANILMASI KONUSUNDA AŞAĞIDA BİLGİLERİ YAZAN ÖĞRENCİLERE İZİN VERİYORUM.

150117905 – ERHAN YALNIZ

150117005 – KADİR ACUN

ENTITIES AND THEIR DEFINITIONS

We have 4 entity as mentioned above before.

1) Product Entity

We created this entity to hold product which will be sold in this market and their required informations. We identify the following additional attributes for products;

- ProductID → This is a unique identifier for products and holds the value of barcode number of the products.
- ProductIDForInfo → This attribute holds the number of the product.
- BuyDate → This attribute holds the date of purchase of the product from the supplier.
- SellDate → This attribute holds the date of sale of product to the consumer.
- ExpirationDate → This attribute holds the date of expiration date of the product.
We created this attribute because a product should be sold to the consumer before the date expires.

2) Product Info Entity

We separate the product entity into the 2 different entity to provide third normal form normalization requirements. This entity is again for product and we identify the following additional attributes for product info(s):

- ProductIDForInfo → This is a unique identifier for products and holds the number of the product.
- ProductName → This attribute holds the name of product.
- Price → This attribute holds the prices of the products.
- BrandName → This attribute holds the name of brands which supplies products for this market.
- SectionNo → We have different sections in this market so that customers can find what they are looking for easily and market owner can follow products states easily. Therefore, this attribute holds the number of the sections.

- Discount → The Market owner may wish to make discounts and campaigns on certain products if he deems it necessary. Furthermore, this attribute holds this state.

3) Brand Entity

We created this entity to hold brands which are the suppliers of the products sold in the market and their necessary informations. We identify the following additional attributes for brands;

- BrandName → This is a unique identifier for brands. This attribute holds the names of companies that produce products.
- PhoneNumber → This attributes holds the phone numbers of suppliers so that the market owner can communicate with them easily when he or she wants.
- EmailAddress → This attributes holds the address of suppliers so that the market owner can communicate with them easily when he or she wants.
- Address → This attributes holds the addresses of the product suppliers. Furthermore, this is a composite attribute which includes city, state and zipcode informations.
- DayForSell → This attributes holds the selling date for the brand. On a certain day of the week, a supplier's employee comes to the market and performs the sale of their products.

4) Section Entity

In this market, there are separate sections for the products so that both employees can easily track the products and customers can easily find the products they are looking for. That's why we created this being. We identify the following additional attributes for sections;

- SectionID → This is a unique identifier for sections. This attribute hold the ID number of sections which includes products separately.
- SectionName → This attribute holds the name of sections.

5) Employee Entity

We created this entity to hold the necessary informations of the employees working in this market and we identify the following additional attributes for employees;

- SSN (TCNo) → This is a unique identifier for employees. This attribute holds the identification number of the Republic of Turkey of employee.
- FirstName → This attribute holds the names of employees.
- LastName → This attribute holds the surnames of employees.
- Birthdate → This attribute holds the birthdates of employees.
- Age → This attribute holds the ages of the employees. Furthermore, this attribute is calculated from the birthDate attribute. (Calculated Column)
- PhoneNumber → This attribute holds the phone numbers of employees.
- Salary → This attribute holds salaries of employees.
- SectionID → This attribute holds the sectionID for the employees. By looking for this information, we can find and get information about workers of the sections.
- Contract → This attribute holds the dates of contract of employees. Furthermore, this attribute is a composite attribute which includes startDate and endDate.
- DaysOff → This attribute holds the days when the employee is on leave.
- WorkingDay → This attribute holds the date informations about working days of employees.

6) RegularCustomer Entity

We created this entity to hold market's regular customers and their required informations. In order for a customer to become a regular customer, they must give their personal information to the grocery store employees and make regular purchases at the market.

- CustomerID → This is a unique identifier for regular customers. This attribute holds the id of the customers.
- FirstName → This attribute holds the first name of the customers.
- LastName → This attribute holds the last name of the customers.
- PhoneNumber → This attribute holds the phone numbers of the customers so that if it is required, market employees or owner can contact with them.

7) Payment Entity

We created this entity to hold payment informations. According to this entity, market's profit and required business rules can identified easily.

- PaymentID → This is a unique identifier for payments. This attribute holds the id of the payments.
- PaymentDescription → This attribute holds the information about the payment type to indicate and record whether payments are made by credit card or cash.
- CustomerID → This attribute holds the customerID who purchase products from the market. If the customer is a regular customer, this attribute holds the id information of him/her. If the customer is not a regular customer, this attribute holds a NULL value.
- ProductID → This attribute holds the value of barcode number of the products.
- PromotionID → If there is a promotion which defined for customers by the market, this attribute holds these promotions identifier numbers(id).
- PaymentDate → This attribute holds the date information of payments.
- Amount → This attribute holds the amount information of payments.

8) Promotion Entity

We created this entity to hold promotions which are defined by the brands (suppliers).

- PromotionID → This attribute holds id of the promotions which is the unique identifier for the promotions.
- PromotionName → This attribute holds the name of the promotions.
- StartDate → This attribute holds the start date of the promotion.
- EndDate → This attribute holds the end date of the promotion.
- DiscountAmount → This attribute holds the discount amount determined when creating promotion.
- BrandName → This attribute holds the names of companies that produce products.

BUSINESS PROCESSES AND THEIR DEFINITIONS

The start of the market's operations are 8:00 am in the morning to 11:30 pm in the afternoon, Monday to Friday. However, on the weekend, since the market is more crowded, the start of the market's operations are 9:00 am in the morning to 1:00 am. The personnels and assigned salespersons must be on time, at least thirty minutes before the opening of the market. The working hours of the workers are form the opening of the market to 6:00 pm in the afternoon, Monday to Saturday. Each worker has 1 day off per week. Employee permits are determined by the day-to-day density of the market and whether there are other employees who can replace it when the employee is on leave. In the market, there are seperate and different sections for the products both employees can easily track the products and customers can easily find the products they are looking for because products are placed in these sections according to their type. Each employee is responsible for his / her department(section) he/she workes. Employee mainly takes care of the customers requirements and have to process on customer request. He/She works on the available products and daily checks the stock availability on time and informs the administrative in charge about the products. Employees of each brand come to the market at regular intervals to ensure that missing or additional products in the market are supplied by their suppliers. If a product of a brand runs out before the day when the supplier's employee arrives at the market, or if more than the number of this product in stock is needed, the supplier company is contacted, which ensures that the product is quickly supplied. Apart from such emergencies, the business process between the market and the supplier companies continues in this way. Furthermore, market where we conduct database work can sometimes

have good deals and opportunities with the supplier companies. For example, a brand can make a campaign for its product, and according to this campaign, if the market owner buys that product from that brand as much as the limit set by the firm, the firm can discount it on the total amount of that purchase. Again for example, a company can set a purchase limit for its customers, and according to this limit, if the amount of purchases made by a customer from this company exceeds the limit, the customer can win gift products from the company. Furthermore, according to the agreements made by the market owner with the supplier company, products that have expired can be returned to the supplier company, so that the market is not damaged. In addition to these, anyone who walks in by the store is considered as a customer and will be treated special and welcomed by the all employees with a smile and respect. Employees at the market welcome incoming customers and guide them to the sections where they can find the products they are looking for. After selecting the products to be purchased, the customer enters the order of purchase. In order to complete customer's shopping, the employee at the checkout reads the barcodes of the products that customers receive to the barcode reader and tells the customers the total shopping amount, and then the customers pay this amount to the employee of the grocery store.

BUSINESS RULES AND CONSTRAINTS

Products are supplied by the brands which are the suppliers.

Brands can produce and supply many products to stores.

A product located in a section so that it can be found easily.

A section can include many product, according the description of its section.

A section has exact one employee.

Employee works on an exact section and he/she responsible from this section.

A product may have a payment but if it is not sold, this payment attribute can not contain anything. Briefly, this information only occurs when the product is sold.

Directly, a payment must have an exact one product.

Payments may be made by a regular customer which makes regular purchases at the market.

A regular customer must make at least one payment so that he or she protect his or her state for this market.

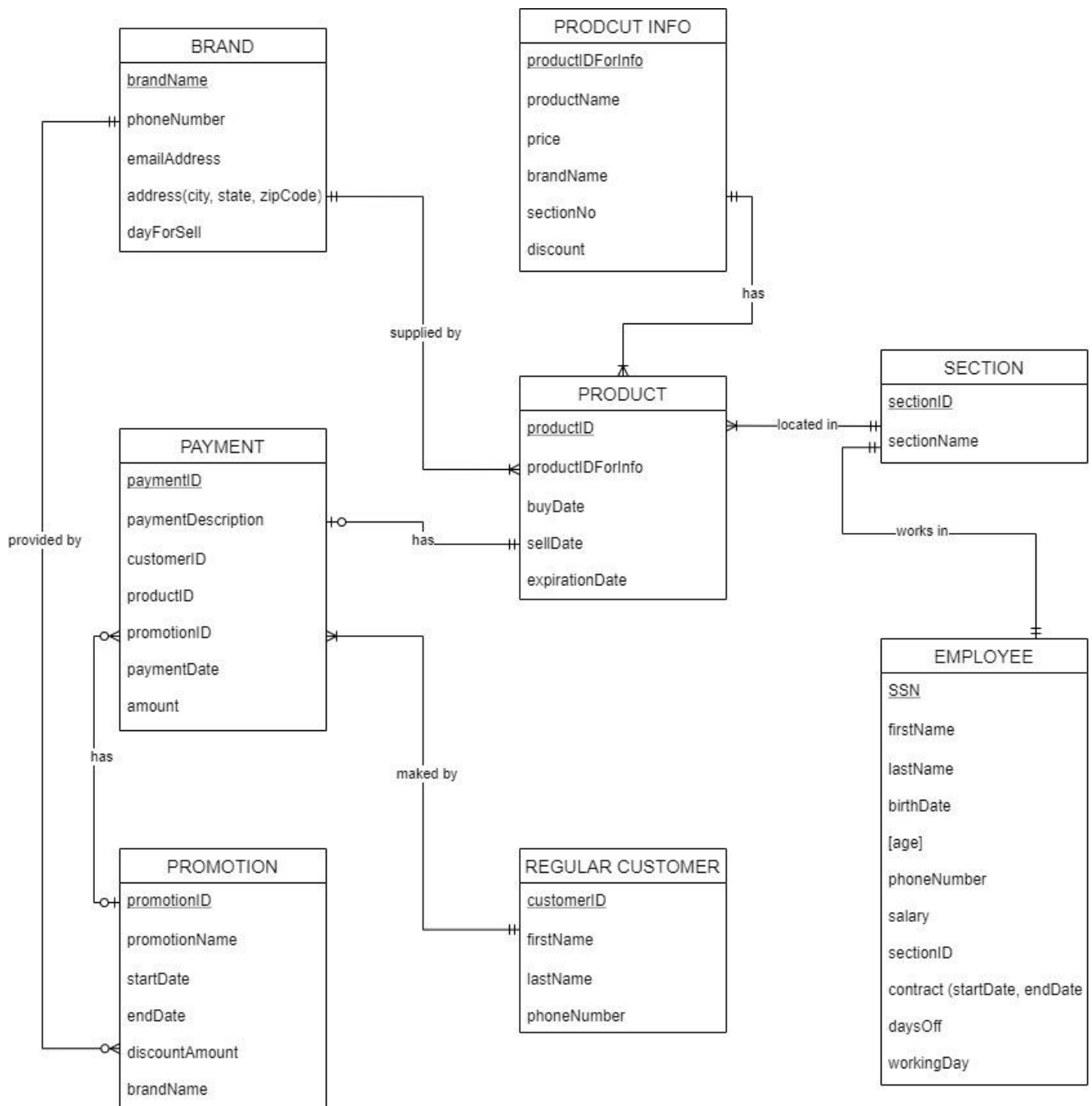
A payment has at most one promotion.

A promotion may have many payments.

OTHER FUNCTIONAL & NON-FUNCTIONAL BUSINESS REQUIREMENTS

At the same time, products imported from Bulgaria are sold in this market, and since there are many Bulgarian immigrants in the location of the market, these products are also very preferred by the customers. But since there is no supplier company for products imported from Bulgaria, the owner or employees of the market go to Bulgaria at regular intervals to be able to allocate these products.

ER DIAGRAM OF THIS DATABASE



TABLES

1) Brand Table

BRAND			
Attributes	Data Type	Primary Key	Foreign Key
brandName	nvarchar(50)	+	
phoneNumber	varchar(25)		
emailAddress	varchar(50)		
city	nvarchar(25)		
state	nvarchar(25)		
zipCode	char(5)		
dayForSell	varchar(10)		

We created this table to hold brands which are the suppliers of the products sold in the market and their necessary informations.

2) Product Table

PRODUCT			
Attributes	Data Type	Primary Key	Foreign Key
productID	int	+	
productIDForInfo	int		+ (Product Info)
buyDate	date		
sellDate	date		
exprationDate	date		

We created this table to hold product which will be sold in this market and their required informations.

3) Product Info Table

PRODUCT_INFO			
Attributes	Data Type	Primary Key	Foreign Key
productIDForInfo	int	+	
productName	nvarchar(50)		
price	Float(24)		
brandName	nvarchar(50)		+ (Brand)
SectionNo	tinyint		+ (Section)
discount	Float(24)		

We separate the product table into the 2 different table to provide third normal form normalization requirements. This table is again for product and it holds the information about products.

4)Section Table

SECTION			
Attributes	Data Type	Primary Key	Foreign Key
sectionID	tinyint	+	
sectionName	Nvarchar(50)		

In this market, there are separate sections for the products so that both employees can easily track the products and customers can easily find the products they are looking for. That's why we created this table.

5)Employee Table

EMPLOYEE			
Attributes	Data Type	Primary Key	Foreign Key
SSN	varchar(11)	+	
firstName	Nvarchar(25)		
lastName	Nvarchar(25)		
birthDate	Date		
Age	smallint		
phoneNumber	varchar(25)		
Salary	smallint		
sectionID	tinyint		+ (Section)
cStartDate	Date		
cEndDate	Date		
daysOff	nvarchar(20)		
workingDay	smallint		

We created this table to hold the necessary informations of the employees working in this market.

6)Payment Table

PAYMENT			
Attributes	Data Type	Primary Key	Foreign Key
paymentID	Bigint	+	
paymentDescription	Char(5)		
customerID	Smallint		+ (Regular Customer)
productid	int		+ (Product)
promotionID	tinyint		+ (Promotion)
paymentDate	Date		
Amount	Float(24)		

We created this table to hold payment informations. According to this table, market's profit and required business rules can identified easily.

In this table, we have 4 trigger which names are afterPromotion, forAmount, forPromotion and forSellDate. Detailed information about them is given in the trigger part of this report.

7)Regular Customer Table

REGULAR_CUSTOMER			
Attributes	Data Type	Primary Key	Foreign Key
customerID	Smallint	+	
firstName	Nvarchar(25)		
lastName	Nvarchar(25)		
phoneNumber	varchar(25)		

We created this table to hold market's regular customers and their required informations. In order for a customer to become a regular customer, they must give their personal information to the grocery store employees and make regular purchases at the market.

8)Promotion Table

PROMOTION			
Attributes	Data Type	Primary Key	Foreign Key
promotionID	tinyint	+	
promotionName	Nvarchar(50)		
startDate	Date		
endDate	Date		
discountAmount	float		
brandName	Nvarchar(50)		+ (Brand)

We created this table to hold promotions which are defined by the brands (suppliers).

INDEXES

We have 2 indexes in 2 table which names are REGULAR_CUSTOMER and EMPLOYEE.

1) For REGULAR_CUSTOMER

Index Name : customer_fullName

It holds the full names of the customers.

```
create index customer_fullName
on REGULAR_CUSTOMER (firstName, lastName);
```

2) For EMPLOYEE

Index Name : employee_fullName

It holds the full names of the employees.

```
create index employee_fullName
on EMPLOYEE (firstName, lastName);
```

UNIQUE CONSTRAINT

We have 2 unique constraints in 2 table which names are BRAND and PAYMENT.

1) Brand Table

- phoneNumber : only have unique value and not have duplicate data so it is UNIQUE.

```
phoneNumber varchar(25) unique
```

2) Payment Table

- ProductID: only have unique value and not have duplicate data so it is UNIQUE.

```
productID int foreign key references PRODUCT(productID) unique
```

IDENTITIES

We have 5 identities in 5 table which names are SECTION, PRODUCT_INFO, PRODUCT, PROMOTION and PAYMENT.

1) Section Table

- sectionID: by using identity, increment the sectionID one by one.

```
sectionID tinyint primary key NOT NULL identity (1,1)
```

2) Product_Info Table

- productIDForInfo: by using identity, increment the productIDForInfo one by one.

```
productIDForInfo int primary key NOT NULL identity (1,1)
```

3) Product Table

- productID: this attribute is defined for barcode number and we want to initialize it from one billion. After this definition, by using identity keyword, increment the product ID one by one.

```
productID int primary key NOT NULL identity (1000000000,1)
```

4) Promotion Table

- promotionID: by using identity, increment the promotionID one by one.

```
promotionID tinyint primary key NOT NULL identity(1,1)
```

5) Payment Table

- paymentID: by using identity, increment the paymentID one by one.

```
paymentID bigint primary key NOT NULL identity(1,1)
```

CHECK CONSTRAINTS

We have 4 check constraints in 3 table which names are PRODUCT, EMPLOYEE and PROMOTION.

1) Product Table

- buyAndSellCheck → Buy date of a product must be smaller than the sell date of this product.

```
constraint buyAndSellCheck check (buyDate <= sellDate)
```

- expirationForProduct → Expiration date of a product must be bigger than or equal to sell date of this product.

```
constraint expirationForProduct check (expirationDate >= sellDate)
```

2) Employee Table

- controlForEmployee → Every employee has an contract about the job and by using this constraint we check the states of employee's contracts. Contract start date (cStartDate) must be smaller than the contract end date (cEndDate).

```
constraint controlForEmployee check (cStartDate < cEndDate)
```

3) Promotion Table

- controlForPromotion → Start date of a promotion must be smaller than the end date of this promotion.

```
constraint controlForPromotion check (startDate < endDate)
```

DEFAULTS

We have 4 defaults in 3 table which names are PRODUCT_INFO, PRODUCT, PAYMENT.

1) Product Info Table

- discount: A product may have a discount value but if the discount is not exist, this attribute is setted to NULL by default.

```
discount float(24) default(NULL)
```

2) Product Table

- sellDate: Before the sell date of a product, it's sellDate attribute is not defined but after it is sold, the value of this attribute is updated.

```
sellDate date default(NULL)
```

3) Payment Table

- promotionID: If a product has not a promotion, the value of this attribute is otomatically setted to NULL value. If a product has a promotion which defined by the suppliers of it, the value of this attribute is updated.

```
promotionID tinyint foreign key references PROMOTION(promotionID) default(NULL)
```

- paymentDate: If date of payment is not entered, this value takes the date of the same day.

```
paymentDate date default (format(GETDATE(), 'yyyy-mm-dd'))
```

COMPUTED COLUMN

We have one computed column in EMPLOOYEE table.

1) Employee Table

- There is a computed column in this Employee table. This column is age column and it computed from the birth date of the employee.

```
Update EMPLOYEE
```

```
Set age = DATEDIFF(year, birthDate, GETDATE())
```


VIEWS

We have 5 views.

1) expirationDateCheck

By using this view, we retrieve productID which is used as barcode number, productIDForInfo, brandName, productName and expirationDate attributes of products which are not sold and have over expiration dates.

Code:

```
--First View For:
--Retrieve productID(barcode number), productIDForInfo, brandName
--productName and expiration dates of Products which are not sold
--and have over expiration date
Create View expirationDateCheck
AS
Select p.productID, p.productIDForInfo, p_i.brandName, p_i.productName, p.expirationDate
From PRODUCT p inner join PRODUCT_INFO p_i
on p.productIDForInfo=p_i.productIDForInfo
Where p.expirationDate<GETDATE() and p.sellDate is NULL
```

Output:

productID	productIDFor...	brandName	productName	expirationDate
1000000603	239	Lezita	Piliç Pirzola	2020-12-28
1000000604	240	Lezita	Bütün Piliç	2020-12-28
1000000605	241	Lezita	Ciğer	2020-12-28
1000000660	296	Pınar	Pınar Süt 1L	2020-04-19
1000000661	297	Pınar	Pınar Süt Deng...	2020-04-19
1000000662	298	İzbakbay	Balkan Süt 1L	2020-04-19
1000000663	299	Pınar	Pınar Süt 180ml	2020-04-19
1000000732	370	Koska	Koska Kağıt Hel...	2020-07-11
1000000779	53	Algida	Boom Boom	2020-08-07
1000000956	234	Lezita	Piliç Bonfile	2020-12-28
1000000957	235	Lezita	Fileto	2020-12-28
1000000958	236	Lezita	Izgara Tava	2020-12-28
1000000959	237	Lezita	Piliç Baget	2020-12-28
1000000960	238	Lezita	Piliç Kanat	2020-12-28
1000000961	239	Lezita	Piliç Pirzola	2020-12-28
1000000962	240	Lezita	Bütün Piliç	2020-12-28
1000000963	241	Lezita	Ciğer	2020-12-28
1000001018	296	Pınar	Pınar Süt 1L	2020-04-19
1000001019	297	Pınar	Pınar Süt Deng...	2020-04-19
1000001020	298	İzbakbay	Balkan Süt 1L	2020-04-19
1000001021	299	Pınar	Pınar Süt 180ml	2020-04-19
1000001090	370	Koska	Koska Kağıt Hel...	2020-07-11

productID	productIDForl...	brandName	productName	expirationDate
1000000046	53	Algida	Boom Boom	2020-08-07
1000000230	234	Lezita	Piliç Bonfile	2020-12-28
1000000231	235	Lezita	Fileto	2020-12-28
1000000232	236	Lezita	Izgara Tava	2020-12-28
1000000233	237	Lezita	Piliç Baget	2020-12-28
1000000234	238	Lezita	Piliç Kanat	2020-12-28
1000000235	239	Lezita	Piliç Pirzola	2020-12-28
1000000236	240	Lezita	Bütün Piliç	2020-12-28
1000000237	241	Lezita	Ciğer	2020-12-28
1000000292	296	Pınar	Pınar Süt 1L	2020-04-19
1000000293	297	Pınar	Pınar Süt Deng...	2020-04-19
1000000294	298	İzbakbay	Balkan Süt 1L	2020-04-19
1000000295	299	Pınar	Pınar Süt 180ml	2020-04-19
1000000370	370	Koska	Koska Kağıt Hel...	2020-07-11
1000000421	53	Algida	Boom Boom	2020-08-07
1000000598	234	Lezita	Piliç Bonfile	2020-12-28
1000000599	235	Lezita	Fileto	2020-12-28
1000000600	236	Lezita	Izgara Tava	2020-12-28
1000000601	237	Lezita	Piliç Baget	2020-12-28
1000000602	238	Lezita	Piliç Kanat	2020-12-28
1000000603	239	Lezita	Piliç Pirzola	2020-12-28
1000000604	240	Lezita	Bütün Piliç	2020-12-28
1000000605	241	Lezita	Ciğer	2020-12-28
1000000660	296	Pınar	Pınar Süt 1L	2020-04-19
1000000661	297	Pınar	Pınar Süt Deng...	2020-04-19
1000000662	298	İzbakbay	Balkan Süt 1L	2020-04-19
1000000663	299	Pınar	Pınar Süt 180ml	2020-04-19
1000000732	370	Koska	Koska Kağıt Hel...	2020-07-11
1000000779	53	Algida	Boom Boom	2020-08-07
1000000956	234	Lezita	Piliç Bonfile	2020-12-28
1000000957	235	Lezita	Fileto	2020-12-28
1000000958	236	Lezita	Izgara Tava	2020-12-28
1000000959	237	Lezita	Piliç Baget	2020-12-28
1000000960	238	Lezita	Piliç Kanat	2020-12-28
1000000961	239	Lezita	Piliç Pirzola	2020-12-28
1000000962	240	Lezita	Bütün Piliç	2020-12-28
1000000963	241	Lezita	Ciğer	2020-12-28
1000001018	296	Pınar	Pınar Süt 1L	2020-04-19

2) stockStateCheck

By using this view, we retrieve productIDForInfo, brandName, productName and the stock states of products which are not sold and ready to sale.

Code:

```
--Second View For:
--Retrieve productIDForInfo, brandName, productName and the stock state of products
--which are not sold and ready to sale.
Create View stockStateCheck
AS
Select p.productIDForInfo, p_i.brandName, p_i.productName, Count(*) stockState
From PRODUCT p inner join PRODUCT_INFO p_i on p.productIDForInfo=p_i.productIDForInfo
Where p.sellDate is NULL
Group By p.productIDForInfo , p_i.brandName, p_i.productName
```

Output:

productIDFor...	brandName	productName	stockState
1	Hal	Yerli Muz	5
2	Hal	Limon	5
3	Hal	Portakal	5
4	Hal	Mandalina	5
5	Hal	Kırmızı Elma	5
6	Hal	Ananas	5
7	Hal	Nar	5
8	Hal	Ayva	5
9	Hal	Sarı Elma	5
10	Hal	Armut	5
11	Hal	Kestane	5
12	Hal	Greyfurt	5
13	Hal	Karpuz	5
14	Hal	Kavun	5
15	Hal	Muşmula	5
16	Hal	Çilek	5
17	Hal	Kivi	5
18	Hal	Domates	4
19	Hal	Salatalık	5
20	Hal	Patates	5
21	Hal	Soğan	5
22	Hal	Marul	5
23	Hal	Lahana	4
24	Hal	Maydanoz	3
25	Hal	Dereotu	5
26	Hal	Nane	5
27	Hal	Roka	5
28	Hal	Taze Soğan	4
29	Hal	Havuç	4
30	Hal	Patlıcan	5
31	Hal	Charleston Biber	5
32	Hal	Sivri Biber	5
33	Hal	Dolmalık Biber	5
34	Hal	Kabak	5
35	Hal	Fasülye	5
36	Hal	Pırasa	5
37	Hal	İspanak	4
44	Algida	Cornetto Classi...	4

productIDForl...	brandName	productName	stockState
45	Algida	Unicornetto	4
46	Algida	Cornetto Milka	4
47	Algida	Cornetto Disc Ç...	4
48	Algida	Cornetto Disc A...	4
49	Algida	Cornetto Disc K...	4
50	Algida	Maraş Usulü Ko...	4
51	Algida	Frootie	4
52	Algida	Magnum Kurab...	4
53	Algida	Boom Boom	4
54	Algida	Magnum Badem	4
55	Algida	Magnum Beyaz	4
56	Algida	Magnum Classic	4
57	Algida	Magnum Doub...	4
58	Algida	Magnum Doub...	4
59	Algida	Magnum Ante...	4
60	Algida	Magnum Ruby	4
61	Algida	Magnum Mini ...	4
62	Algida	Magnum Mini ...	4
63	Algida	Magnum Mini ...	4
64	Algida	Magnum Mini ...	4
65	Algida	Magnum Mini ...	4
66	Algida	Maraş Usulü Cup	3
67	Algida	Maraş Usulü Sa...	4
70	Algida	Maraş Usulü Sa...	4
71	Algida	Maraş Usulü Çif...	4
72	Algida	Maraş Usulü Sa...	4
73	Algida	Maraş Usulü Sa...	4
74	Algida	Maraş Usulü Ke...	4
75	Algida	Maraş Usulü Sü...	4
76	Algida	Maraş Supangle	4
77	Algida	Maraş Dondur...	4
78	Algida	Carte d'Or Kat ...	4
79	Algida	Carte d'Or Sele...	4
80	Algida	Carte d'Or Clas...	4
81	Algida	Carte d'Or Bal B...	4
82	Algida	Carte d'Or Sele...	4
83	Algida	Carte d'Or Clas...	4
84	Algida	Carte d'Or Sele...	4

productIDForl...	brandName	productName	stockState
85	Algida	Big Bold	4
86	Algida	Nogger	4
87	Eti	Alaska Frigo M...	4
88	Eti	Alaska Frigo Ka...	4
89	Eti	Alaska Frigo Bit...	4
90	Eti	Alaska Frigo Sü...	4
91	Algida	Keyif Vanilya	4
92	Algida	Carte d'Or Clas...	4
93	Algida	Magnum Yakut	4
94	Algida	Carte d'Or Sele...	4
95	Algida	Magnum Cherry	4
96	Algida	Classics Kakao	4
97	Algida	Classics Vanilya	4
98	Algida	Classics Antep ...	4
99	Selin Kırtasiye	Copier Bond Fo...	4
100	Selin Kırtasiye	Pritt Keçeli Kale...	4
101	Selin Kırtasiye	Sele Bant	4
102	Selin Kırtasiye	Pritt Yapıştırıcı ...	4
103	Selin Kırtasiye	Elişi Kağıdı 10'lu	4
104	Selin Kırtasiye	Çizim Seti	4
105	Selin Kırtasiye	Büyük Makas	4
106	Selin Kırtasiye	Pritt Yapıştırıcı ...	4
107	Selin Kırtasiye	Çıt Çıt Dosya	4
108	Selin Kırtasiye	Resim Defteri 3...	4
109	Selin Kırtasiye	Yağlı Pastel Boya	4
110	Selin Kırtasiye	Pritt Beyaz Tutkal	4
111	Selin Kırtasiye	Etiket	4
112	Selin Kırtasiye	Cetvel	4
113	Selin Kırtasiye	Pritt Yapıştırıcı ...	4
114	Selin Kırtasiye	Resim Defteri 2...	4
115	Selin Kırtasiye	Pritt Guaj Boya	4
116	Selin Kırtasiye	Poşet Dosya	4
117	Selin Kırtasiye	Pritt Sulu Boya	4
118	Selin Kırtasiye	Fon Kartonu	4
119	Selin Kırtasiye	Renkli Silgi	4
120	Selin Kırtasiye	Tükenmez Kalem	4
121	Selin Kırtasiye	Pilot Kalem	4
122	Selin Kırtasiye	Uhu 7ml Yapıştı...	4

productIDForl...	brandName	productName	stockState
123	Selin Kırtasiye	Kalemtraş Büyük	4
124	Selin Kırtasiye	Kalemtraş Küçük	4
125	Selin Kırtasiye	Fırça Set 6'lı	4
126	Selin Kırtasiye	Boyama Kitabı	4
127	Selin Kırtasiye	5'li Kaplık	4
128	Selin Kırtasiye	Pritt Silgi	4
129	Selin Kırtasiye	Selafon Bant	4
130	Selin Kırtasiye	Şeffaf Kaplık	4
131	Selin Kırtasiye	Fosforlu Kalem	4
132	Selin Kırtasiye	Sunum Dosyası...	4
133	Selin Kırtasiye	Pritt Kuru Boya ...	4
134	Selin Kırtasiye	Kurşun Kalem	4
135	Selin Kırtasiye	Kırmızı Kalem	4
136	Selin Kırtasiye	Ataç	4
137	Selin Kırtasiye	Kalemlik	4
138	Selin Kırtasiye	Post-it Set	4
139	Selin Kırtasiye	Okul Çantası	4
140	Selin Kırtasiye	Krapon Kağıdı 1...	4
141	Selin Kırtasiye	Pritt Boyama Seti	4
142	Selin Kırtasiye	Mukavva	4
143	Selin Kırtasiye	Küçük Makas	4
144	Selin Kırtasiye	Klasör	4
145	Selin Kırtasiye	Sayı Çubuğu	4
146	Selin Kırtasiye	Daksil	4
147	Selin Kırtasiye	Tahta Kalemi	4
148	Faber-Castell	Faber-Castell K...	4
149	Faber-Castell	Faber-Castell Sı...	4
150	Faber-Castell	Faber-Castell P...	4
151	Faber-Castell	Faber-Castell K...	4
152	Faber-Castell	Faber-Castell K...	4
153	Faber-Castell	Faber-Castell Gl...	4
154	Faber-Castell	Faber-Castell T...	4
155	Faber-Castell	Faber-Castell S...	4
156	Faber-Castell	Faber-Castell P...	4
157	Faber-Castell	Faber-Castell O....	4
158	Faber-Castell	Faber-Castell O....	4
159	Faber-Castell	Faber-Castell O...	4
160	Faber-Castell	Faber-Castell Tr...	4

productIDForl...	brandName	productName	stockState
161	Faber-Castell	Faber-Castell F...	4
162	Faber-Castell	Faber-Castell P...	4
163	Faber-Castell	Faber-Castell G...	4
164	Faber-Castell	Faber-Castell M...	4
165	Faber-Castell	Faber-Castell Y...	4
166	Faber-Castell	Faber-Castell A...	4
167	Faber-Castell	Faber-Castell G...	4
168	Faber-Castell	Faber-Castell M...	4
169	Faber-Castell	Faber-Castell S...	4
170	Faber-Castell	Faber-Castell K...	4
171	Selin Kırtasiye	Rotring Uçlu Ka...	4
172	Selin Kırtasiye	Mikro 72yp Def...	4
173	Selin Kırtasiye	Mikro 96yp Def...	4
174	Selin Kırtasiye	Mikro 210yp De...	4
175	Selin Kırtasiye	Mikro 120yp De...	4
176	Selin Kırtasiye	Müzik Defteri	4
177	Selin Kırtasiye	Mikro 40yp Def...	4
178	Selin Kırtasiye	Play-Doh Oyun...	4
179	Unilever	Domestos 750ml	4
180	Unilever	Domestos 3240...	4
181	Pril	Pril Sıvı Deterjan	4
182	Unilever	Omo Toz Çama...	4
183	Unilever	Omo Sıvı Çama...	4
184	Unilever	Omo Toz Çama...	4
185	Unilever	Omo Toz Çama...	4
186	Unilever	Omo Sıvı Çama...	4
187	Unilever	Omo Active Sıvı	4
188	Unilever	Omo Konsantre...	4
189	Unilever	Cif Powe&Shin...	4
190	Unilever	Cif Tüm Yüzeyl...	4
191	Unilever	Cif Jel Zemin U...	4
192	Unilever	Cif Power Perfect	4
193	Unilever	Yumoş Uzman ...	4
194	Unilever	Yumoş Extra 14...	4
195	Unilever	Dove Deodoran...	4
196	Unilever	Dove Şampuan ...	4
197	Unilever	Dove Duj Jeli 50...	4
198	Unilever	Dove Sabun 10...	4

productIDForl...	brandName	productName	stockState
199	Unilever	Dove Saç Baki...	4
200	Unilever	Axe Cep Parfü...	4
201	Unilever	Axe Deodorant	4
202	Unilever	Signal Diş Fırçası	4
203	Unilever	Signal White N...	4
204	Bizim Toptan	Parex Temizlik ...	4
205	Bizim Toptan	Vileda Temizlik...	4
206	Bizim Toptan	Finish Konsantr...	4
207	Bizim Toptan	Finish Jel 1300ml	4
208	Bizim Toptan	Finish Quantu...	4
209	Bizim Toptan	Finish Quantu...	4
210	Bizim Toptan	Fairy Platinum ...	4
211	Bizim Toptan	Fairy Platinum ...	4
212	Bizim Toptan	Fairy Hepsi Bir ...	4
213	Bizim Toptan	Fairy Bulaşık De...	4
214	Bizim Toptan	Paex 8'li Bulaşık...	4
215	Bizim Toptan	Parex 3'lü Ovm...	4
216	Bizim Toptan	Selpak Tuvalet ...	4
217	Bizim Toptan	Selpak Tuvalet ...	4
218	Bizim Toptan	Papia Tuvalet K...	4
219	Bizim Toptan	Papia Havlu 6'lı	4
220	Bizim Toptan	Solo Havlu 6'lı	4
221	Bizim Toptan	Kloroplast Çöp ...	4
222	Bizim Toptan	Kloroplast Çöp ...	4
223	Bizim Toptan	Banat Faraşlı Sü...	4
224	Bizim Toptan	Sensodyne Diş ...	4
225	Bizim Toptan	Sensodyne Diş ...	4
226	Bizim Toptan	Listerine 500ml	4
227	Bizim Toptan	Listerine 250ml	4
229	Bizim Toptan	Head&Shoulde...	4
230	Bizim Toptan	Clear Şampuan ...	4
231	Bizim Toptan	Pantene Şampu...	4
232	Bizim Toptan	Pantene Saç Kr...	4
234	Lezita	Piliç Bonfile	4
235	Lezita	Fileto	4
236	Lezita	Izgara Tava	4
237	Lezita	Piliç Baget	4
238	Lezita	Piliç Kanat	4

productIDForl...	brandName	productName	stockState
239	Lezita	Piliç Pirzola	4
240	Lezita	Bütün Piliç	4
241	Lezita	Ciğer	4
242	Tadsüt	Polonez Dana K...	4
243	Tadsüt	Polonez Macar ...	4
244	Tadsüt	Polonez Kangal...	4
245	Tadsüt	Polonez Fıstıklı ...	4
246	Tadsüt	Polonez Kasap ...	4
247	Tadsüt	Polonez Ferma...	4
248	Tadsüt	Polonez Fit Yaş...	4
249	Tadsüt	Polonez Pastır...	4
250	Tadsüt	Polonez Fûme ...	4
251	Pınar	Aç Bitir Hindi S...	4
252	Pınar	Aç Bitir Macar S...	4
253	Pınar	Aç Bitir Kokteyl ...	4
254	Pınar	Aç Bitir Sucuk 9...	4
255	Bizim Toptan	Banvit Piliç Nu...	4
256	Bizim Toptan	Banvit Piliç Sch...	4
257	Coca-Cola	Coca-Cola 2.5L	3
258	Coca-Cola	Coca-Cola 1L	4
259	Coca-Cola	Coca-Cola 250ml	4
260	Coca-Cola	Coca-Cola 330ml	4
261	Coca-Cola	Coca-Cola Ener...	4
262	Coca-Cola	Fanta 2.5L	3
263	Coca-Cola	Fanta 1L	4
264	Coca-Cola	Fanta 250ml	4
265	Coca-Cola	Fanta 330ml	4
266	Coca-Cola	Sprite 2.5L	4
267	Coca-Cola	Sprite 1L	4
268	Coca-Cola	Schweppes 1L	4
269	Coca-Cola	Damla Su 19L	4
270	Coca-Cola	Damla Su 1L	4
271	Coca-Cola	Damla Su 500ml	4
272	Erikli	Erikli 19L Su	4
273	Erikli	Erikli 1.5L Su	4
274	Erikli	Erikli 500ml Su	4
275	Pepsi	Pepsi Cola 2.5L	4
276	Pepsi	Pepsi Cola 1L	4

productIDForl...	brandName	productName	stockState
277	Pepsi	Lipton İce Tea 2L	3
278	Pepsi	Lipton İce Tea 1...	4
279	Pepsi	Lipton İce Tea 1L	4
280	İzbakbay	Dimes Nektar 1L	3
281	İzbakbay	Dimes Nektar 2...	4
282	İzbakbay	Tropicana Nekt...	4
283	İzbakbay	Cappy Nektar 1L	4
284	Uludağ	Uludağ Gazoz 2...	4
285	Uludağ	Uludağ Gazoz 1...	4
286	Uludağ	Uludağ Limona...	4
287	Uludağ	Uludağ Frutti M...	4
288	Coca-Cola	Damla Meyveli ...	4
289	Uludağ	Uludağ Sade So...	3
290	İzbakbay	Sarıköz Sade So...	4
291	İzbakbay	Doğanay Şalga...	4
292	İzbakbay	Red Bull 250ml	4
293	İzbakbay	Red Bull 473ml	4
294	Nestle	Nescafe Expres...	3
295	Sütaş	Sütaş Ayran 1L	3
296	Pınar	Pınar Süt 1L	4
297	Pınar	Pınar Süt Deng...	4
298	İzbakbay	Balkan Süt 1L	4
299	Pınar	Pınar Süt 180ml	4
300	Işık Yumurta	Yumurta	4
301	Sütaş	Sütaş Geleneks...	4
302	Pınar	Pınar Tereyağı F...	4
303	Öncü	Öncü Zeytin 1kg	4
304	Balparmak	Balparmak Çiçe...	4
305	Balparmak	Balparmak Ça...	4
306	Tukaş	Tukaş Vişne Reç...	4
307	Tukaş	Tukaş Kayısı Re...	4
308	Tukaş	Tukaş Vişne Bö...	4
309	Tukaş	Tukaş Çilek Reç...	4
310	Koska	Koska Sade Hel...	4
311	Koska	Koska Kakaolu ...	4
312	Koska	Koska Light Hel...	4
313	Koska	Koska Fıstıklı H...	4
314	Koska	Koska Tahin 550...	4

productIDFor...	brandName	productName	stockState
315	Koska	Koska Pekmez ...	4
316	Bizim Toptan	Nutella 750gr	4
317	Bizim Toptan	Nutella 400gr	4
318	Ülker	Ülker Çokokre...	3
319	Ülker	Ülker Çokokre...	3
320	Ülker	Ülker Çokokre...	3
321	Ülker	Ülker Golden Fı...	4
322	Nestle	Nesquik 450gr	4
323	Nestle	Nestle Cheerios...	4
324	Nestle	Nestle Corn Fla...	4
325	Eti	Eti Form Tahıl G...	4
326	Nestle	Nestle Nesfit Sa...	4
327	Nestle	Nestle Nesfit M...	4
328	Nestle	Nestle Nesfit Çi...	4
329	Sütaş	Sütaş Tam Yağlı...	4
330	Sütaş	Sütaş Süzme Pe...	4
331	Sütaş	Sütaş Süzme Pe...	4
332	Pınar	Pınar Süzme Pe...	4
333	Tahsildaroğlu	Tahsildaroğlu K...	4
334	Tahsildaroğlu	Tahsildaroğlu E...	4
335	Tahsildaroğlu	Tahsildaroğlu E...	4
336	Tahsildaroğlu	Tahsildaroğlu Ö...	4
337	Tahsildaroğlu	Tahsildaroğlu D...	4
340	Tahsildaroğlu	Tahsildaroğlu E...	4
341	Tahsildaroğlu	Tahsildaroğlu L...	4
342	Tahsildaroğlu	Tahsildaroğlu İz...	4
343	Sütaş	Sütaş Kaşar Pey...	4
344	MuratBey	Muratbey Burg...	4
345	MuratBey	MuratBey Kaşk...	4
346	MuratBey	MuratBey Tel P...	4
347	MuratBey	MuratBey Taze ...	4
348	MuratBey	MuratBey Dilim...	4
349	Pınar	Pınar Beyaz 200gr	4
350	Pınar	Pınar Labne 200...	4
351	Pınar	Pınar Labne 400...	4
352	Pınar	Pınar Krem Pey...	4
353	İzbakbay	Tıkveçli Salep 1L	4
354	Ülker	Bizim Paket Ma...	4
355	Ülker	Teremyağ Paket...	4
356	Ülker	Teremyağ Kase ...	4
357	Ülker	Teremyağ Kase ...	4
358	Pınar	Pınar Krema 20...	4
359	Fritolay	Doritos Taco	3
360	Fritolay	Doritos Hotcorn	3
361	Fritolay	Lay's Klasik Pat...	4
362	Fritolay	Doritos Extreme	3
363	Fritolay	Lay's Baharat P...	3
364	Fritolay	Doritos Naco	4
365	Fritolay	Ruffles Originals	3
366	Fritolay	Ruffles Ketçaplı	4
367	Tukaş	Tukaş Domates ...	4
368	Tukaş	Tukaş Domates ...	4
369	Tukaş	Tukaş Biber Sal...	4
370	Koska	Koska Kağıt Hel...	4
NULL	NULL	NULL	NULL

3) totalCustomerPayment

By using this view, we retrieve customerID, totalAmount which is calculated by sum syntax and collection of amount attributes and paymentDescription. Then, it shows customer's total payment with their ids and payment ways to us.

Code:

```
--Third View For:  
--Retrieve customerID, totalAmount, paymentDescription.  
Create View totalCustomerPayment  
AS  
Select p.customerID, Sum(p.amount) totalAmount, p.paymentDescription  
From PAYMENT p inner join REGULAR_CUSTOMER rc on p.customerID = rc.customerID  
Group By p.customerID, p.paymentDescription
```

Output:

customerID	totalAmount	paymentDescription
1	7,949999809265...	card
3	4,5	card
7	13,5	card
11	11,75	card
13	44,75	card
14	14,5	card
18	2	card
1	6	cash
8	6	cash
9	13,875	cash
10	8,0625	cash
13	4,6875	cash
14	2	cash
15	7,5	cash
17	11,25	cash
NULL	NULL	NULL

4) soldProductAmount

By using this view, we retrieve productIDForInfo, productName and saleAmount which is calculated by count syntax. Then, it shows amount of the sale products with their names and ids to us.

Code:

```
--Forth View For:  
--Retrieve productIDForInfo, productName, saleAmount.  
Create View soldProductAmount  
AS  
Select pi.productIDForInfo, pi.productName , Count(*) saleAmount  
From Payment p inner join Product pro on p.productID = pro.productID  
        inner join PRODUCT_INFO pi on pi.productIDForInfo = pro.productIDForInfo  
Group by pi.productIDForInfo, pi.productName
```

Output:

productIDFor...	productName	saleAmount
18	Domates	1
23	Lahana	1
24	Maydanoz	2
28	Taze Soğan	1
29	Havuç	1
37	Ispanak	1
66	Maraş Usulü Cup	1
257	Coca-Cola 2.5L	1
262	Fanta 2.5L	1
277	Lipton İce Tea 2L	1
280	Dimes Nektar 1L	1
289	Uludağ Sade So...	1
294	Nescafe Expres...	1
295	Sütaş Ayran 1L	1
318	Ülker Çokokre...	1
319	Ülker Çokokre...	1
320	Ülker Çokokre...	1
359	Doritos Taco	1
360	Doritos Hotcorn	1
362	Doritos Extreme	1
363	Lay's Baharat P...	1
365	Ruffles Originals	1
NULL	NULL	NULL

5) promotionWithLoss

By using this view, we retrieve promotionID, brandName, startDate, endDate and paymentDifferences. Then, it shows how much money the market loses when the products are purchased at promotional prices to us.

Code:

```
--Fifth View For:  
--Retrieve promotionID, brandName, startDate, endDate, paymentDifferences.  
Create View promotionWithLoss  
AS  
Select prom.promotionID, prom.brandName, prom.startDate, prom.endDate , (Sum(pi.price) - Sum(pay.amount)) paymentDifferences  
From Payment pay inner join PROMOTION prom on pay.promotionID = prom.promotionID  
            inner join PRODUCT pro on pro.productID = pay.productID  
            inner join PRODUCT_INFO pi on pro.productIDForInfo = pi.productIDForInfo  
Group By prom.promotionID, prom.brandName, prom.startDate, prom.endDate
```

Output:

promotionID	brandName	startDate	endDate	paymentDiffer...
1	Fritolay	2020-12-23	2021-01-31	7,375
2	Nestle	2020-12-23	2021-01-31	1,5
3	Pepsi	2020-12-23	2021-01-31	1,25
NULL	NULL	NULL	NULL	NULL

We have 4 triggers and these are in PAYMENT table.

After a payment is inserted, it sets the payment amount according to product price.

```
--Trigger 1 (for amount)

create trigger trg_forAmount
ON PAYMENT
AFTER INSERT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    UPDATE p
    set p.amount=pri.price
    from PAYMENT p inner join inserted i on i.paymentID=p.paymentID
    inner join PRODUCT pr on pr.productID=p.productID
    inner join PRODUCT_INFO pri on pri.productIDForInfo=pri.productIDForInfo
    -- Insert statements for trigger here

END
GO
```

[illegible]

2) Trg_afterPromotion

If a product has a promotion, it calculates the payment amount according to promotion discount.

Code:

```
--Trigger 2 (for promotion) and it updates promotion if it is exist

create trigger trg_afterPromotion
ON PAYMENT
AFTER INSERT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    UPDATE p
    set p.amount=p.amount-(p.amount*pro.discountAmount)
    from PAYMENT p inner join inserted i on i.paymentID=p.paymentID
    inner join PRODUCT pr on pr.productID=p.productID
    inner join PRODUCT_INFO pri on pri.productIDForInfo=pr.productIDForInfo
    inner join BRAND b on b.brandName=pri.brandName
    inner join PROMOTION pro on pro.brandName=b.brandName
    where pro.startDate <= GETDATE() and pro.endDate >= GETDATE()
    -- Insert statements for trigger here
END
GO
```

Output:

[illegible]

3) Trg_forPromotion

If a product has a promotion, it updates the information of productID.

Code:

```
--Trigger 3 (for promotion) and it updates promotionID fo payment
create trigger trg_forPromotion
ON    PAYMENT
    AFTER INSERT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    UPDATE p
    set p.promotionID=pro.promotionID
    from PAYMENT p inner join inserted i on i.paymentID=p.paymentID
    inner join PRODUCT pr on pr.productID=p.productID
    inner join PRODUCT_INFO pri on pri.productIDForInfo=pr.productIDForInfo
    inner join BRAND b on b.brandName=pri.brandName
    inner join PROMOTION pro on pro.brandName=b.brandName
    where pro.startDate <= GETDATE() and pro.endDate >= GETDATE()
    -- Insert statements for trigger here
END
GO
```

Output:

[illegible]

4) Trg_forSellDate

After a product is sold, it updates the sell date of this product.

Code:

```
--Trigger 4 (for updating sell date)

create trigger trg_forSellDate
ON PAYMENT
AFTER INSERT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    update p
    set p.sellDate=GETDATE()
    from PRODUCT p inner join inserted i on i.productID=p.productID
    -- Insert statements for trigger here

END
GO
```

Output:

productID	productIDForl...	buyDate	sellDate	expirationDate
1000000283	287	2020-11-06	NULL	2021-10-23
1000000284	288	2020-11-06	NULL	2021-10-23
1000000285	289	2020-11-06	2020-12-28	2021-10-23
1000000286	290	2020-11-06	NULL	2021-10-23
1000000287	291	2020-11-06	NULL	2021-10-23
1000000288	292	2020-11-13	NULL	2022-11-13
1000000289	293	2020-11-13	NULL	2022-11-13
1000000290	294	2020-11-13	2020-12-28	2022-11-13
1000000291	295	2020-12-25	2020-12-28	2021-01-20
1000000292	296	2020-10-30	NULL	2020-04-19
1000000293	297	2020-10-30	NULL	2020-04-19
1000000294	298	2020-10-30	NULL	2020-04-19
1000000295	299	2020-10-30	NULL	2020-04-19
1000000297	300	2020-12-25	NULL	2021-01-23
1000000301	301	2020-12-24	NULL	2021-02-22
1000000302	302	2020-12-23	NULL	2021-02-02
1000000303	303	2020-04-22	NULL	2022-10-22
1000000304	304	2020-04-22	NULL	2022-10-22
1000000305	305	2020-09-09	NULL	2021-09-09
1000000306	306	2020-09-09	NULL	2021-09-09

STORED PROCEDURES

We have 12 stored procedures. They are listed below.

1) Sp_insertBrand

This stored procedure is created for BRAND table and it inserts new brand.

Code:

```
--Stored Procedure for Brand
--Inserting New Brand
create procedure sp_insertBrand
@brandName nvarchar(50),
@phoneNumber varchar(25),
@emailAddress varchar(50),
@city nvarchar(25),
@state nvarchar(25),
@zipCode char(5),
@dayForSell varchar(10)
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO BRAND(brandName,phoneNumber,emailAddress,city,state,
        zipCode,dayForSell)
    VALUES (@brandName,@phoneNumber,@emailAddress,@city,@state,
        @zipCode,@dayForSell);
    -- Insert statements for procedure here
END

exec sp_insertBrand @brandName = 'Eker', @phoneNumber = '02324799687',
    @emailAddress = 'info@eker.com', @city = 'İzmir',
    @state = 'Bornova', @zipCode = '35060',
    @dayForSell = 'Cuma'
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T04:46:26.0905993+03:00

Output:

.Gocmen_Sup...rket - dbo.BRAND X SQLQueryMarket.s...(ERHAN\ASUS (54))*						
brandName	phoneNumber	emailAddress	city	state	zipCode	dayForSell
Algida	02166339000	NULL	İstanbul	Ümraniye	37768	Cumartesi
Balparmak	02166430202	NULL	İstanbul	Çekmeköy	34782	Çarşamba
Bizim Toptan	02322650101	NULL	İzmir	Gazimir	35410	Çarşamba
Coca-Cola	02328770500	kutsal_adali@c...	İzmir	Kemalpaşa	35737	Cuma
Eker	02324799687	info@eker.com	İzmir	Bornova	35060	Cuma
Erikli	02324796699	NULL	İzmir	Bornova	35050	Cuma
Eti	02322815657	NULL	İzmir	Gazimir	35414	Pazartesi
Faber-Castell	0 850 677 70 00	NULL	Kocaeli	Çayırova	41480	Salı
Fritolay	02324361594	NULL	İzmir	Bornova	35070	Cumartesi
Hal	02324430004	NULL	İzmir	Buca	35395	Pazartesi
Işık Yumurta	02324588383	NULL	İzmir	Buca	35390	Cuma
İzbakbay	02322761100	NULL	İzmir	Buca	35390	Cuma
Koska	02324722097	NULL	İzmir	Bornova	35070	Perşembe
Lezita	05543331033	NULL	İzmir	Menemen	35663	Perşembe
Muratbey	02322326668	NULL	İzmir	Bornova	35070	Salı
Nestle	02323280909	NULL	İzmir	Çiğli	35620	Cuma
Öncü	02325255211	NULL	İzmir	Tire	35900	Çarşamba
Pepsi	08502795050	NULL	İzmir	Bornova	35070	Cuma
Pınar	4447627	info@pinar.co...	İzmir	Bornova	35060	Salı
Pril	02165794000	NULL	İstanbul	Ümraniye	34771	Çarşamba
Selin Kırtasiye	02324498364	info@selinkirta...	İzmir	Konak	35170	Salı
Sütaş	02324353930	NULL	İzmir	Bornova	35050	Salı
TadSüt	02327811001	info@tadsüt.com	İzmir	Menderes	35473	Perşembe
Tahsildaroğlu	02324362595	NULL	İzmir	Bornova	35060	Salı
Tukaş	02328651555	info@tukas.co...	İzmir	Torbalı	35880	Çarşamba
Uludağ	02324358511	NULL	İzmir	Bornova	35070	Cuma
Unilever	44425199	NULL	İstanbul	Ümraniye	37768	Çarşamba
Ülker	02125671567	ulkeras@ulker.c...	İstanbul	Üsküdar	34692	Pazartesi
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2) sp_brandPhone

This stored procedure is created for BRAND table and it updates brand's phone number.

Code:

```
--Stored Procedure for Brand
--Updating Brand's phone number if it changes
create procedure sp_brandPhone
@brandName nvarchar(50),
@phoneNumber varchar(25)
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE BRAND
    SET phoneNumber = @phoneNumber
    FROM BRAND
    WHERE brandName = @brandName
END

exec sp_brandPhone @brandName = 'Eker', @phoneNumber = '01234567890'
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T04:59:28.3625119+03:00

Output:

[illegible]

3) sp_insertEmployee

This stored procedure is created for EMPLOYEE table and it inserts new employee.

Code:

```
--Stored Procedure for Employee
--Inserting New Employee
create procedure sp_insertEmployee
@SSN varchar(11),
@firstName nvarchar(25),
@lastName nvarchar(25),
@birthDate date,
@age smallint,
@phoneNumber varchar(25),
@salary smallint,
@sectionID tinyint,
@cStartDate date,
@cEndDate date,
@daysOff nvarchar(20),
@workingDay smallint
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO EMPLOYEE(SSN,firstName,lastName,birthDate,age,
        phoneNumber,salary,sectionID,cStartDate,
        cEndDate,daysOff,workingDay)
    VALUES (@SSN,@firstName,@lastName,@birthDate,@age,
        @phoneNumber,@salary,@sectionID,@cStartDate,
        @cEndDate,@daysOff,@workingDay);
    -- Insert statements for procedure here
END

exec sp_insertEmployee @SSN = '98765432100' , @firstName = 'Murat',
    @lastName = 'Gündoğdu' , @birthDate = '1995-05-17' ,
    @age = NULL , @phoneNumber = '05398765432' ,
    @salary = 3100 , @sectionID = 2 ,
    @cStartDate = '2017-12-03' , @cEndDate = '2022-12-03' ,
    @daysOff = 'Salı' , @workingDay = NULL
```

Messages

Commands completed successfully.

Completion time: 2020-12-29T05:19:06.0426075+03:00

Output:

SSN	firstName	lastName	birthDate	age	phoneNumber	salary	sectionID	cStartDate	cEndDate	daysOff	workingDay
11111111111	Özcan	Sever	1972-06-17	48	05311111111	4600	2	2015-08-11	2025-08-11	Pazar	NULL
22222222222	Kadir	Can	1994-08-01	26	05322222222	2400	5	2018-09-23	2023-09-23	Pazartesi	NULL
33333333333	Meltem	Aslan	1996-05-02	24	05333333333	2400	6	2019-03-08	2024-03-08	Salı	NULL
44444444444	Gizem	Başar	1991-07-21	29	05344444444	2900	4	2016-04-29	2021-04-29	Cuma	NULL
55555555555	Serkan	Yanık	1983-12-31	37	05355555555	3000	3	2012-06-16	2022-06-16	Çarşamba	NULL
66666666666	Ezgi	Ormancı	1992-01-11	28	05366666666	2950	1	2016-01-03	2021-01-03	Perşembe	NULL
77777777777	İrem	Bayer	1987-11-09	33	05377777777	4000	7	2009-11-07	2024-11-07	Cumartesi	NULL
98765432100	Murat	Gündoğdu	1995-05-17	25	05398765432	3100	2	2017-12-03	2022-12-03	Salı	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4) sp_deleteEmployee

This stored procedure is created for EMPLOYEE table and it deletes an employee.

Code:

```
--Stored Procedure for Employee
--Deleting an Employee
create procedure sp_deleteEmployee
@SSN varchar(11)
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM EMPLOYEE
    WHERE SSN = @SSN
    -- Insert statements for procedure here
END

exec sp_deleteEmployee @SSN = '98765432100'
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T05:26:10.0901270+03:00

Output:

SSN	firstName	lastName	birthDate	age	phoneNumber	salary	sectionID	cStartDate	cEndDate	daysOff	workingDay
11111111111	Özcan	Sever	1972-06-17	48	05311111111	4600	2	2015-08-11	2025-08-11	Pazar	NULL
22222222222	Kadir	Can	1994-08-01	26	05322222222	2400	5	2018-09-23	2023-09-23	Pazartesi	NULL
33333333333	Meltem	Aslan	1996-05-02	24	05333333333	2400	6	2019-03-08	2024-03-08	Salı	NULL
44444444444	Gizem	Başar	1991-07-21	29	05344444444	2900	4	2016-04-29	2021-04-29	Cuma	NULL
55555555555	Serkan	Yanık	1983-12-31	37	05355555555	3000	3	2012-06-16	2022-06-16	Çarşamba	NULL
66666666666	Ezgi	Ormancı	1992-01-11	28	05366666666	2950	1	2016-01-03	2021-01-03	Perşembe	NULL
77777777777	İrem	Bayer	1987-11-09	33	05377777777	4000	7	2009-11-07	2024-11-07	Cumartesi	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5) sp_updateForEmployee

This stored procedure is created for EMPLOYEE table and it updates employee's contract date and salary.

Code:

```
--Stored Procedure for Employee
--Updating Employee's contract date and salary
create procedure sp_updateForEmployee
@SSN varchar(11),
@cEndDate date,
@salary smallint
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE EMPLOYEE
    SET cEndDate = @cEndDate , salary = @salary
    FROM EMPLOYEE
    WHERE SSN = @SSN
END

exec sp_updateForEmployee @SSN = '44444444444' , @cEndDate = '2026-04-29' , @salary = 3500
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T05:48:59.0548589+03:00

Output:

ERHAN.Gocmen_Su...t - dbo.EMPLOYEE - X SQLQueryMarket.s...(ERHAN\ASUS (54))*												
	SSN	firstName	lastName	birthDate	age	phoneNumber	salary	sectionID	cStartDate	cEndDate	daysOff	workingDay
	11111111111	Özcan	Sever	1972-06-17	48	05311111111	4600	2	2015-08-11	2025-08-11	Pazar	NULL
	22222222222	Kadir	Can	1994-08-01	26	05322222222	2400	5	2018-09-23	2023-09-23	Pazartesi	NULL
	33333333333	Meltem	Aslan	1996-05-02	24	05333333333	2400	6	2019-03-08	2024-03-08	Salı	NULL
▶	44444444444	Gizem	Bagar	1991-07-21	29	05344444444	3500	4	2016-04-29	2026-04-29	Cuma	NULL
	55555555555	Serkan	Yanık	1983-12-31	37	05355555555	3000	3	2012-06-16	2022-06-16	Çarşamba	NULL
	66666666666	Ezgi	Ormancı	1992-01-11	28	05366666666	2950	1	2016-01-03	2021-01-03	Perşembe	NULL
	77777777777	İrem	Bayer	1987-11-09	33	05377777777	4000	7	2009-11-07	2024-11-07	Cumartesi	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6) sp_insertPayment

This stored procedure is created for PAYMENT table and it inserts new payment.

Code:

```
--Stored Procedure for Payment
--Inserting Payment
create procedure sp_insertPayment
@description char(5), @customerId smallint, @productId int
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO PAYMENT(paymentDescription,customerID,productId,paymentDate)
    VALUES (@description,@customerId,@productId,GETDATE());
    -- Insert statements for procedure here
END

exec sp_insertPayment @description = 'card' , @customerId = 2 , @productId = 1000000441
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T05:57:34.4438877+03:00

Output:

[illegible]

7) sp_insertProduct

This stored procedure is created for PRODUCT table and it inserts new product.

Code:

```
--Stored Procedure for Product
--Inserting Product
create procedure sp_insertProduct
@productIDForInfo int,
@buyDate date,
@expirationDate date
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO PRODUCT(productIDForInfo,buyDate,expirationDate)
    VALUES (@productIDForInfo,@buyDate,@expirationDate);
    -- Insert statements for procedure here
END

exec sp_insertProduct @productIDForInfo = 324 , @buyDate = '2020-12-06' , @expirationDate = '2022-12-06'
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T06:14:13.2283893+03:00

Output:

ERHAN.Gocmen_Su...et - dbo.PRODUCT					
SQLQueryMarket.s...(ERHAN\ASUS (54))*					
	productID	productIDForl...	buyDate	sellDate	expirationDate
	1000001451	3	2020-12-28	NULL	NULL
	1000001452	4	2020-12-28	NULL	NULL
	1000001453	5	2020-12-28	NULL	NULL
	1000001454	6	2020-12-28	NULL	NULL
	1000001455	7	2020-12-28	NULL	NULL
	1000001456	8	2020-12-28	NULL	NULL
	1000001457	9	2020-12-28	NULL	NULL
	1000001458	10	2020-12-28	NULL	NULL
	1000001459	11	2020-12-28	NULL	NULL
	1000001460	12	2020-12-28	NULL	NULL
	1000001461	13	2020-12-28	NULL	NULL
	1000001462	14	2020-12-28	NULL	NULL
	1000001463	15	2020-12-28	NULL	NULL
	1000001464	16	2020-12-28	NULL	NULL
	1000001465	17	2020-12-28	NULL	NULL
	1000001466	18	2020-12-28	NULL	NULL
	1000001467	19	2020-12-28	NULL	NULL
	1000001468	20	2020-12-28	NULL	NULL
	1000001469	21	2020-12-28	NULL	NULL
	1000001470	22	2020-12-28	NULL	NULL
	1000001471	23	2020-12-28	2020-12-28	NULL
	1000001472	24	2020-12-28	2020-12-28	NULL
	1000001473	25	2020-12-28	NULL	NULL
	1000001474	26	2020-12-28	NULL	NULL
	1000001475	27	2020-12-28	NULL	NULL
	1000001476	28	2020-12-28	NULL	NULL
	1000001477	29	2020-12-28	2020-12-28	NULL
	1000001478	30	2020-12-28	NULL	NULL
	1000001479	31	2020-12-28	NULL	NULL
	1000001480	32	2020-12-28	NULL	NULL
	1000001481	33	2020-12-28	NULL	NULL
	1000001482	34	2020-12-28	NULL	NULL
	1000001483	35	2020-12-28	NULL	NULL
	1000001484	36	2020-12-28	NULL	NULL
	1000001485	37	2020-12-28	2020-12-28	NULL
▶	1000001487	324	2020-12-06	NULL	2022-12-06
*	NULL	NULL	NULL	NULL	NULL

8) sp_insertProductInfo

This stored procedure is created for PRODUCT_INFO table and it inserts new product info.

Code:

```
--Stored Procedure for Product_Info (Insert)
create procedure sp_insertProductInfo
@productName nvarchar(50),
@price float(24),
@brandName nvarchar(50),
@sectionNo tinyint,
@discount float
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO PRODUCT_INFO(productName,price,brandName,sectionNo,discout)
VALUES (@productName,@price,@brandName,@sectionNo,@discount);
    -- Insert statements for procedure here
END

exec sp_insertProductInfo @productName = 'Nescafe Gold 200gr' , @price = 47 ,
@brandName = 'Nestle' , @sectionNo = 2 , @discount = NULL
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T06:24:53.3277278+03:00

Output:

ERHAN.Gocmen_S...bo.PRODUCT_INFO - X SQLQueryMarket.s...(ERHAN\ASUS (54))*						
	productIDForl...	productName	price	brandName	sectionNo	discount
	334	Tahsildaroğlu Eski Kaşar 350gr	32,75	Tahsildaroğlu	4	NULL
	335	Tahsildaroğlu Ezine Keçi Peynir 1kg	62	Tahsildaroğlu	4	NULL
	336	Tahsildaroğlu Örgü Peynir 225gr	17,9	Tahsildaroğlu	4	NULL
	337	Tahsildaroğlu Dil Peyniri 250gr	17,9	Tahsildaroğlu	4	NULL
	340	Tahsildaroğlu Ezine Koyun Peynir 1kg	42,5	Tahsildaroğlu	4	NULL
	341	Tahsildaroğlu Lor Peyniri 350gr	7	Tahsildaroğlu	4	NULL
	342	Tahsildaroğlu İzmir Tulum Peyniri 350gr	28,75	Tahsildaroğlu	4	NULL
	343	Sütaş Kaşar Peyniri 700gr	32,25	Sütaş	4	NULL
	344	MuratBey Burgu Peynir 1kg	48,75	MuratBey	4	NULL
	345	MuratBey Kaşkaval Peyniri 400gr	28	MuratBey	4	NULL
	346	MuratBey Tel Peynir 200gr	10,5	MuratBey	4	NULL
	347	MuratBey Taze Kaşar Peyniri 400gr	23,9	MuratBey	4	NULL
	348	MuratBey Dilimli Kaşar 225gr	16,75	MuratBey	4	NULL
	349	Pınar Beyaz 200gr	8,5	Pınar	4	NULL
	350	Pınar Labne 200gr	8	Pınar	4	NULL
	351	Pınar Labne 400gr	15,5	Pınar	4	NULL
	352	Pınar Krem Peynir 160gr	7,5	Pınar	4	NULL
	353	Tikveşli Salep 1L	10,75	İzbackbay	5	NULL
	354	Bizim Paket Margarin 250gr	3,75	Ülker	2	NULL
	355	Teremyağ Paket Margarin 250gr	4,75	Ülker	2	NULL
	356	Teremyağ Kase 250gr	5,75	Ülker	2	NULL
	357	Teremyağ Kase 500gr	10,5	Ülker	2	NULL
	358	Pınar Krema 200ml	9	Pınar	2	NULL
	359	Doritos Taco	6,25	Fritolay	2	NULL
	360	Doritos Hotcorn	6,25	Fritolay	2	NULL
	361	Lay's Klasik Patates Cipsi	6	Fritolay	2	NULL
	362	Doritos Extreme	4,5	Fritolay	2	NULL
	363	Lay's Baharat Patates Cipsi	6,25	Fritolay	2	NULL
	364	Doritos Naco	6,25	Fritolay	2	NULL
	365	Ruffles Originals	6,25	Fritolay	2	NULL
	366	Ruffles Ketçaplı	6,25	Fritolay	2	NULL
	367	Tukaş Domates Salçası 1650gr	38,75	Tukaş	2	NULL
	368	Tukaş Domates Salçası 700gr	10,75	Tukaş	2	NULL
	369	Tukaş Biber Salçası 700gr	14	Tukaş	2	NULL
	370	Koska Kağıt Helva	2	Koska	2	NULL
▶	371	Nescafe Gold 200gr	47	Nestle	2	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

9) sp_updateProductInfo

This stored procedure is created for PRODUCT_INFO table and it updates price of a product.

Code:

```
--Stored Procedure for Product_Info (Update)
create procedure sp_updateProductInfo
@productIDForInfo int,
@price float(24)
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE PRODUCT_INFO
    SET price = @price
    FROM PRODUCT_INFO
    WHERE productIDForInfo = @productIDForInfo
END

exec sp_updateProductInfo @productIDForInfo = 371 , @price = 52
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T06:30:01.8629970+03:00

Output:

ERHAN.Gocmen_S...bo.PRODUCT_INFO - X SQLQueryMarket.s...(ERHAN\ASUS (54))*						
	productIDForl...	productName	price	brandName	sectionNo	discount
	334	Tahsildaroğlu E...	32,75	Tahsildaroğlu	4	NULL
	335	Tahsildaroğlu E...	62	Tahsildaroğlu	4	NULL
	336	Tahsildaroğlu Ö...	17,9	Tahsildaroğlu	4	NULL
	337	Tahsildaroğlu D...	17,9	Tahsildaroğlu	4	NULL
	340	Tahsildaroğlu E...	42,5	Tahsildaroğlu	4	NULL
	341	Tahsildaroğlu L...	7	Tahsildaroğlu	4	NULL
	342	Tahsildaroğlu İz...	28,75	Tahsildaroğlu	4	NULL
	343	Sütaş Kaşar Pey...	32,25	Sütaş	4	NULL
	344	Muratbey Burg...	48,75	MuratBey	4	NULL
	345	MuratBey Kaşk...	28	MuratBey	4	NULL
	346	MuratBey Tel P...	10,5	MuratBey	4	NULL
	347	MuratBey Taze ...	23,9	MuratBey	4	NULL
	348	MuratBey Dilim...	16,75	MuratBey	4	NULL
	349	Pınar Beyaz 200gr	8,5	Pınar	4	NULL
	350	Pınar Labne 200...	8	Pınar	4	NULL
	351	Pınar Labne 400...	15,5	Pınar	4	NULL
	352	Pınar Krem Pey...	7,5	Pınar	4	NULL
	353	Tikveşli Salep 1L	10,75	İzbakbay	5	NULL
	354	Bizim Paket Ma...	3,75	Ülker	2	NULL
	355	Teremyağ Paket...	4,75	Ülker	2	NULL
	356	Teremyağ Kase ...	5,75	Ülker	2	NULL
	357	Teremyağ Kase ...	10,5	Ülker	2	NULL
	358	Pınar Krema 20...	9	Pınar	2	NULL
	359	Doritos Taco	6,25	Fritolay	2	NULL
	360	Doritos Hotcorn	6,25	Fritolay	2	NULL
	361	Lay's Klasik Pat...	6	Fritolay	2	NULL
	362	Doritos Extreme	4,5	Fritolay	2	NULL
	363	Lay's Baharat P...	6,25	Fritolay	2	NULL
	364	Doritos Naco	6,25	Fritolay	2	NULL
	365	Ruffles Originals	6,25	Fritolay	2	NULL
	366	Ruffles Ketçaplı	6,25	Fritolay	2	NULL
	367	Tukaş Domates ...	38,75	Tukaş	2	NULL
	368	Tukaş Domates ...	10,75	Tukaş	2	NULL
	369	Tukaş Biber Sal...	14	Tukaş	2	NULL
	370	Koska Kağıt Hel...	2	Koska	2	NULL
	371	Nescafe Gold 2...	52	Nestle	2	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

10) sp_insertPromotion

This stored procedure is created for PROMOTION table and it inserts new promotion.

Code:

```
--Stored Procedure for Promotion
--Inserting Promotion
create procedure sp_insertPromotion
@promotionName nvarchar(50),
@startDate date,
@endDate date,
@discountAmount float,
@brandName nvarchar(50)
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO PROMOTION(promotionName,startDate,
                           endDate,discountAmount,brandName)
    VALUES (@promotionName,@startDate,@endDate,
            @discountAmount,@brandName);
    -- Insert statements for procedure here
END

exec sp_insertPromotion @promotionName = 'Okula Dönüş' , @startDate = '2020-09-01' ,
@endDate = '2021-03-01' , @discountAmount = 0.3 , @brandName = 'Faber-Castell'
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T06:37:42.4460997+03:00

Output:

ERHAN.Gocmen_Su...- dbo.PROMOTION SQLQueryMarket.s...(ERHAN\ASUS (54))*						
	promotionID	promotionNa...	startDate	endDate	discountAmo...	brandName
	1	Yeni Yıl	2020-12-23	2021-01-31	0,25	Fritolay
	2	Yeni Yıl2	2020-12-23	2021-01-31	0,2	Nestle
	3	Yeni Yıl3	2020-12-23	2021-01-31	0,2	Pepsi
	4	Coca-Cola ile K...	2021-01-10	2021-04-10	0,15	Coca-Cola
	5	Üker ile Mutlul...	2021-01-01	2021-02-01	0,1	Ülker
▶	6	Okula Dönüş	2020-09-01	2021-03-01	0,3	Faber-Castell
*	NULL	NULL	NULL	NULL	NULL	NULL

11) sp_insertCustomer

This stored procedure is created for REGULAR_CUSTOMER table and it inserts new customer.

Code:

```
--Stored Procedure for Customer
--Inserting Customer
create procedure sp_insertCustomer
@customerID smallint,
@firstName nvarchar(25),
@lastName nvarchar(25),
@phoneNumber varchar(25)
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO REGULAR_CUSTOMER(customerID,firstName,lastName,phoneNumber)
    VALUES (@customerID,@firstName,@lastName,@phoneNumber);
    -- Insert statements for procedure here
END

exec sp_insertCustomer @customerID = 25, @firstName = 'Kaan' , @lastName = 'Yavuz',
@phoneNumber = '05321233210'
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T06:43:09.8934696+03:00

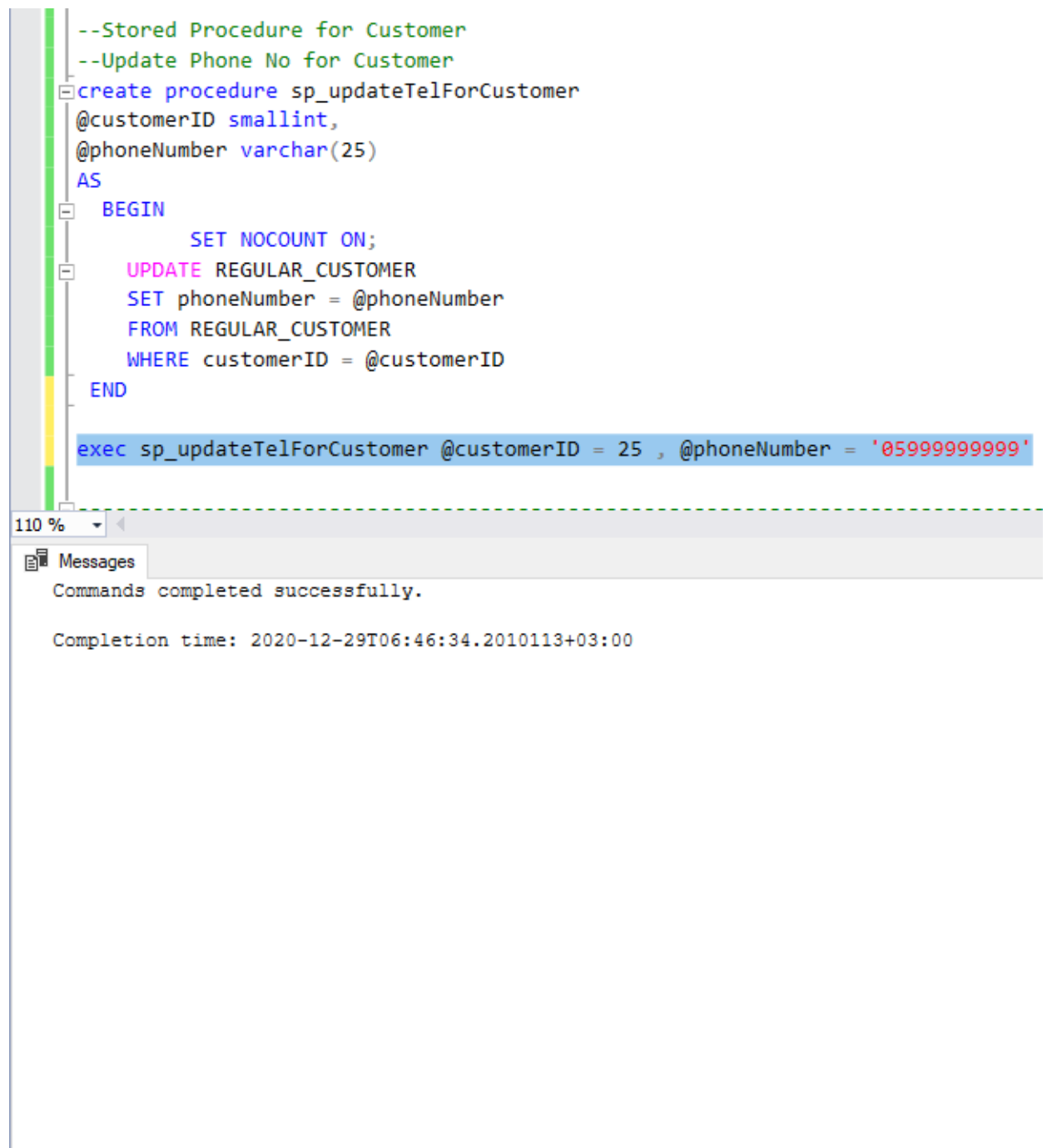
Output:

ERHAN.Gocmen_S...EGULAR_CUSTOMER ↗ ✕ SQLQueryMarket.s...(ERHAN\ASUS (54))*				
	customerID	firstName	lastName	phoneNumber
	1	Deniz	Yıldız	05111111111
	2	Yunus	Solak	05111111112
	3	Sevda	Özdemir	05111111113
	4	Burak	Demir	05111111114
	5	Hazal	Çakan	05111111115
	6	Taner	Güngör	05111111116
	7	Tuna	Solak	05111111117
	8	Savaş	Kurt	05111111118
	9	Sezgin	Bacak	05111111119
	10	Sedat	Gündüz	05111111120
	11	Sibel	Koç	05111111121
	12	Çiğdem	Çam	05111111122
	13	Kubilay	Güzel	05111111123
	14	Filiz	Kaya	05111111124
	15	Özlem	Özdağ	05111111125
	16	Ali	Şahin	05111111126
	17	Buket	Özen	05111111127
	18	Hakan	Doğan	05111111128
	19	Hüseyin	Yıldırım	05111111129
	20	Aysel	Yılmaz	05111111130
	21	Zeynep	Elmas	05111111131
	22	Erol	Çetin	05111111132
	23	Ahmet	Mert	05111111133
	24	Aysun	Gök	05111111134
▶	25	Kaan	Yavuz	05321233210
*	NULL	NULL	NULL	NULL

12) sp_updateTelForCustomer

This stored procedure is created for REGULAR_CUSTOMER table and it updates the phone number of a customer.

Code:



```
--Stored Procedure for Customer
--Update Phone No for Customer
create procedure sp_updateTelForCustomer
@customerID smallint,
@phoneNumber varchar(25)
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE REGULAR_CUSTOMER
    SET phoneNumber = @phoneNumber
    FROM REGULAR_CUSTOMER
    WHERE customerID = @customerID
END

exec sp_updateTelForCustomer @customerID = 25 , @phoneNumber = '05999999999'
```

110 %

Messages

Commands completed successfully.

Completion time: 2020-12-29T06:46:34.2010113+03:00

Output:

ERHAN.Gocmen_S...EGULAR_CUSTOMER SQLQueryMarket.s...(ERHAN\ASUS (54))*				
	customerID	firstName	lastName	phoneNumber
	1	Deniz	Yıldız	05111111111
	2	Yunus	Solak	05111111112
	3	Sevda	Özdemir	05111111113
	4	Burak	Demir	05111111114
	5	Hazal	Çakan	05111111115
	6	Taner	Güngör	05111111116
	7	Tuna	Solak	05111111117
	8	Savaş	Kurt	05111111118
	9	Sezgin	Bacak	05111111119
	10	Sedat	Gündüz	05111111120
	11	Sibel	Koç	05111111121
	12	Çiğdem	Çam	05111111122
	13	Kubilay	Güzel	05111111123
	14	Filiz	Kaya	05111111124
	15	Özlem	Özdağ	05111111125
	16	Ali	Şahin	05111111126
	17	Buket	Özen	05111111127
	18	Hakan	Doğan	05111111128
	19	Hüseyin	Yıldırım	05111111129
	20	Aysel	Yılmaz	05111111130
	21	Zeynep	Elmas	05111111131
	22	Erol	Çetin	05111111132
	23	Ahmet	Mert	05111111133
	24	Aysun	Gök	05111111134
►	25	Kaan	Yavuz	05999999999
*	NULL	NULL	NULL	NULL