

İÇİNDEKİLER

İÇİNDEKİLER	II
ŞEKİL LİSTESİ	IV
TABLO LİSTESİ	VI
1.Giriş.....	1
2.Gereklilik Analizi.....	2
3.Kavramsal Tasarım.....	2
3.1.Kasa Varlık Kümesi.....	3
3.2.Ürün Varlık Kümesi.....	3
3.3.Satış Varlık Kümesi.....	4
3.4.Satılan Ürünler Varlık Kümesi.....	5
3.5.Marka Varlık Kümesi.....	6
3.6.Ödeme Varlık Kümesi.....	6
3.7.Ciro Varlık Kümesi.....	7
3.8.Kar Varlık Kümesi.....	8
3.9.Varlık İlişkileri(ER).....	9
4.Mantıksal Tasarım.....	12
5.Fiziksel Tasarım.....	15
5.1.dbo.KASA Tablosu.....	15
5.2.dbo.URUN Tablosu.....	16
5.3.dbo.SATIS Tablosu.....	16
5.4.dbo.SAT_URUN Tablosu.....	17
5.5.dbo.MARKA Tablosu.....	17
5.6.dbo.ODEME Tablosu.....	18
5.7.dbo.CIRO Tablosu.....	18
5.8.dbo.KAR Tablosu.....	18

5.9. Veri Tabanı Diyagramı.....	19
5.10. Tabloların SQL Kodları.....	19
6. Projenin Akış Şeması.....	24
6.1. Kasa Giriş Formu Akış Şeması.....	24
6.2. Ürün Ekleme Formu Akış Şeması.....	25
6.3. Ürün Güncelleme Formu Akış Şeması.....	26
6.4. Stok Ekleme Formu Akış Şeması.....	27
6.5. Satış Geçmişi Formu Akış Şeması.....	28
6.6. Rapor Formu Akış Şeması.....	29
7. Projenin Uygulama Kısmı.....	30
7.1 Kasa Giriş Ekranı.....	30
7.2. Menü Ekranı.....	31
7.3. Satış Ekranı.....	32
7.4. Ürün Ekleme Ekranı.....	36
7.5. Ürün Bilgilerini Güncelleme Ekranı.....	38
7.6. Toplu Stok Ekranı.....	39
7.7. Satış Geçmişi Ekranı.....	40
7.8. Kâr ve Ciro Ekranı.....	42
7.9. Stok Bilgileri Ekranı.....	42
8. Projenin Kodları.....	44
KAYNAKÇA.....	83

ŞEKİL LİSTESİ

Şekil 1: Kasa Varlık Kümesi.....	3
Şekil 2: Ürün Varlık Kümesi.....	4
Şekil 3: Satış Varlık Kümesi.....	5
Şekil 4: Satılan Ürünler Varlık Kümesi.....	6
Şekil 5: Marka Varlık Kümesi.....	6
Şekil 6: Ödeme Varlık Kümesi.....	7
Şekil 7: Ciro Varlık Kümesi.....	8
Şekil 8: Kar Varlık Kümesi.....	8
Şekil 9: Kasa-Satış İlişki Şeması.....	9
Şekil 10: Ürün-Marka İlişki Şeması.....	9
Şekil 11: Satılan Ürünler-Satış İlişki Şeması.....	10
Şekil 12: Satılan Ürünler-Alış İlişki Şeması.....	10
Şekil 13: Satış-Ödeme İlişki Şeması.....	11
Şekil 14: Genel Gösterim.....	11
Şekil 15: Kasa Giriş Formu Akış Şeması.....	24
Şekil 16: Ürün Ekleme Formu Akış Şeması.....	25
Şekil 17: Ürün Güncelleme Formu Akış Şeması.....	26
Şekil 18: Stok Ekleme Formu Akış Şeması.....	27
Şekil 19: Satış Geçmişi Formu Akış Şeması.....	28
Şekil 20: Rapor Formu Akış Şeması.....	29
Şekil 21: Kasa Giriş Ekranı	30
Şekil 22: Kasa Giriş Ekranı	31
Şekil 23: Menü Ekranı	32
Şekil 24: Satış Ekranı	33

Şekil 25: Satış Ekranı	34
Şekil 26: Satış Ekranı	35
Şekil 27: Yeni Ürün Ekle Ekranı	36
Şekil 28: Yeni Ürün Ekle Ekranı	37
Şekil 29: Yeni Ürün Ekle Ekranı	37
Şekil 30: Ürün Bilgilerini Güncelleme Ekranı	38
Şekil 31: Ürün Bilgilerini Güncelleme Ekranı	39
Şekil 32: Toplu Stok Ekranı	40
Şekil 33: Satış Geçmişi Ekranı	41
Şekil 34: Satış Geçmişi Ekranı	41
Şekil 35: Kâr ve Ciro Rapor Ekranı	42
Şekil 36: Stok Bilgileri Ekranı	43
Şekil 37: Stok Bilgileri Ekranı	43

TABLO LİSTESİ

Tablo 1: Kasa Tablosu.....	12
Tablo 2: Ürün Tablosu.....	12
Tablo 3: Satış Tablosu.....	13
Tablo 4: Satılan Ürünler Tablosu.....	13
Tablo 5: Marka Tablosu.....	14
Tablo 6: Ödeme Tablosu.....	14
Tablo 7: Ciro Tablosu.....	14
Tablo 8: Kar Tablosu.....	15
Tablo 9: dbo.KASA Tablosu.....	15
Tablo 10: dbo.URUN Tablosu.....	16
Tablo 11: dbo.SATIS Tablosu.....	17
Tablo 12: dbo.SAT_URUN Tablosu.....	17
Tablo 13: dbo.MARKA Tablosu.....	17
Tablo 14: dbo.ODEME Tablosu.....	18
Tablo 15: dbo.CIRO Tablosu.....	18
Tablo 16: dbo.KAR Tablosu.....	19
Tablo 17: Veri Tabanı Diagramı	19

1.GİRİŞ

Bu uygulamada; marketlerin ürün, satış, stok, ciro ve kâr bilgilerinin elektronik ortamda tutularak, kullanıcının her daim bilgilendirilmesi ve gerektiği anda kolaylıkla müdahale edebilmesi amaçlanmıştır. Yapılacak bu uygulama mevcut marketlerin ihtiyaçlarını karşılamak amacıyla tasarlanmıştır.

Bugün Türkiye’de 30.000 civarındaki market sayısı bulunmaktadır. Bu sayı 2024 yılında yüzde 53 artışla 44.000 civarına ulaşması beklenmektedir. Her geçen gün artan bu sayı sektördeki market uygulaması ihtiyacını da artırmaktadır. Bu ihtiyaç, marketlerin çeşitliliğine göre değişiklik gösterebilir. Değişiklikler, uygulamaya da uyarlanabilir şekilde tasarlanabilir.

İlk aşamada; uygulama kullanıcısının tedarik ettiği ürünleri en hızlı ve doğru şekilde kayıt altına alınması, bu ürünlerin satışından sonra mevcut ürünlerin sayısını hesaplayarak stok bilgisinin edinilmesi ve satıştan elde edilen cironun gün sonunda kasada ne kadar olması gerektiğinin hesabı yapılarak, kullanıcıya hem zaman hem de kolaylık sağlanması amaçlanmıştır. Sonraki aşamalarda ise kullanıcı ile iletişim halinde kalarak diğer ihtiyaçlara cevap verilmesi planlanmıştır.

2.GEREKLİLİK ANALİZİ

Gerekli arařtırmalar sonucunda elde edilen bilgilerin analizi sonucunda kullanıcı gereksinimleri detaylı ve bir bütün oluřturacak řekilde listelendikten sonra program oluřturulur.

Market uygulamasında kullanıcıya saėlanan kolaylıklar;

- Ürün bilgilerinin kayıt altına alınması,
- Satıř bilgilerinin kayıt altına alınması,
- Ciro hesabının yapılarak kayıt altına alınması,
- Kar tespitinin yapılarak kayıt altına alınması
- Ciro ve kar istatistiklerinin takip edilebilmesi.

Bunları bir program ile kayıt altında tutmak iřletme sahibinin kolaylıkla istediėi veriye ulařmasını saėlayacak ve verimliliėi arttıracaktır.

3.KAVRAMSAL TASARIM

Kavramsal tasarım ařaėıdaki ihtiyaların giderilmesi maksadıyla tasarlanmıřtır.

- Ürün bilgilerinin kolay ve hızlı řekilde girilmesi,
- Ürün bilgilerinin anlařılır bir biimde listelenmesi,
- Satıř iřleminin kolay ve hızlı řekilde yapılması,
- Satıř bilgilerinin kayıt altında tutularak gemiř dönük takip edilebilmesi,
- Stok bilgilerinin sürekli güncellenmesi,
- Stok bilgilerinin takibinin yapılması,
- Günlük ciro ve kar hesabının yapılması,
- Ciro ve karın istatistiklerine daha kolay eriřme,
- Her türlü bilgiye ait ekleme, ıkarma, düzenleme, silme ve güncelleme iřleminin yapılması.

Bu çerevede; gereklilik analizinden de yola ıkılarak yapılan alıřma sonucunda ařaėıdaki varlıkların planlanması gerekli bulunmuř, planlama da buna yönelik olarak yapılmıřtır. Buna göre Varlık Kümelerinin ařaėıdaki maddelerden oluřması planlanmıřtır.

- Kasa Varlık Kümesi,
- Ürün Varlık Kümesi,
- Satıř Varlık Kümesi,

- Satılan Ürünler Varlık Kümesi,
- Marka Varlık Kümesi,
- Ödeme Varlık Kümesi,
- Ciro Varlık Kümesi,
- Kar Varlık Kümesi.

3.1.Kasa Varlık Kümesi

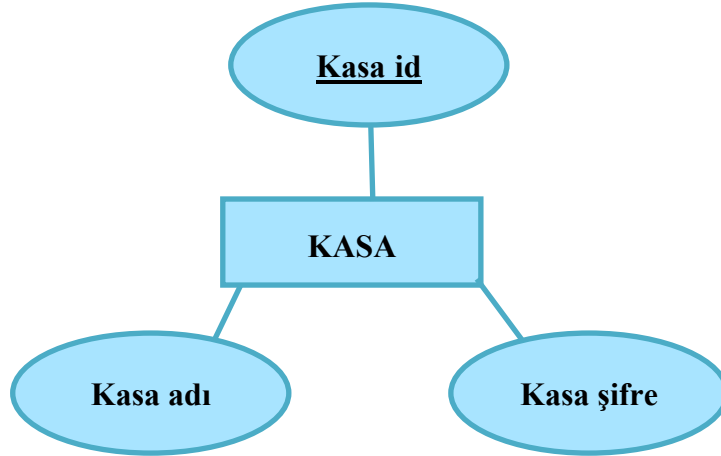
Kasa Varlık Kümesi; herhangi bir satış elemanı ya da kullanıcının hangi kasadan satış yaptığının bilgilerini tutmak amacıyla tasarlanmıştır. Bu tasarımda; satış yapılan her kasa için ayrı bir isim ve şifre tanımlanması düşünülmüştür.

Buna göre Kasa Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Kasa Varlık Kümesi;

- Kasa id,
- Kasa adı,
- Kasa şifre.

Buna göre Kasa Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Kasa id** anahtar nitelik olarak tanımlanmıştır.



Şekil 1: Kasa Varlık Kümesi

3.2.Ürün Varlık Kümesi

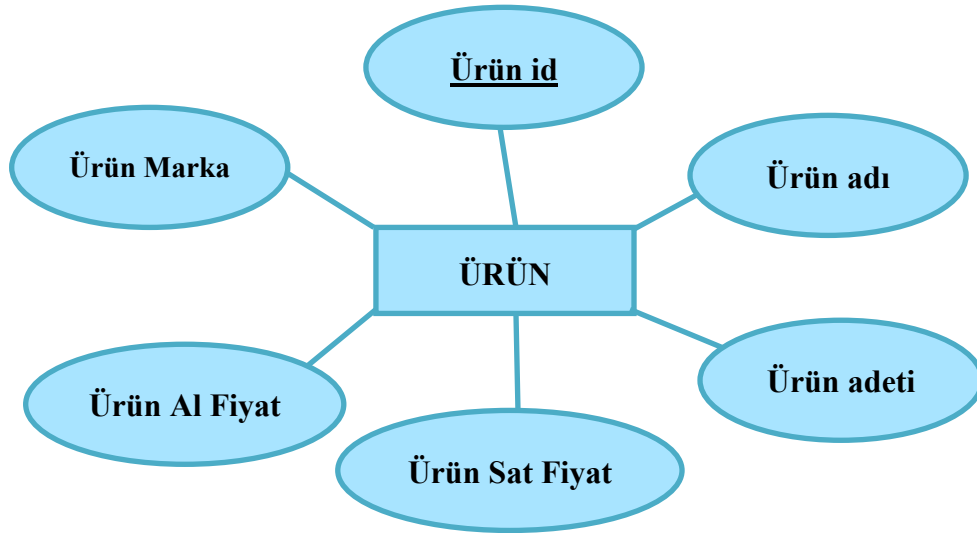
Ürün Varlık Kümesi; temin edilen her bir ürünün kayıtlarının tutulması amacıyla tasarlanmıştır. Bu tasarımda; her bir ürüne ait ürün ismi, temin edilen adet, ürünün markası, ürünün alış ve satış fiyatı bilgileri tutulmaktadır.

Buna göre Ürün Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Ürün Varlık Kümesi;

- Ürün id,
- Ürün adı,
- Ürün adeti,
- Ürün Markası,
- Ürün alış fiyatı.
- Ürün Satış fiyatı.

Buna göre Ürün Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Ürün id** anahtar nitelik olarak tanımlanmıştır.



Şekil 2: Ürün Varlık Kümesi

3.3.Satış Varlık Kümesi

Satış Varlık Kümesi; müşterinin satın alma talebinde bulunduğu ürünlerin kayıtlarının kayıtlarını tutmak amacıyla tasarlanmıştır. Bu tasarımda; satılan ürün veya ürünlerin adının, ne zaman satıldığının, toplam satış tutarının, ödemenin nasıl yapıldığının ve satış işleminin hangi kasadan gerçekleştiğinin bilgileri tutulmaktadır.

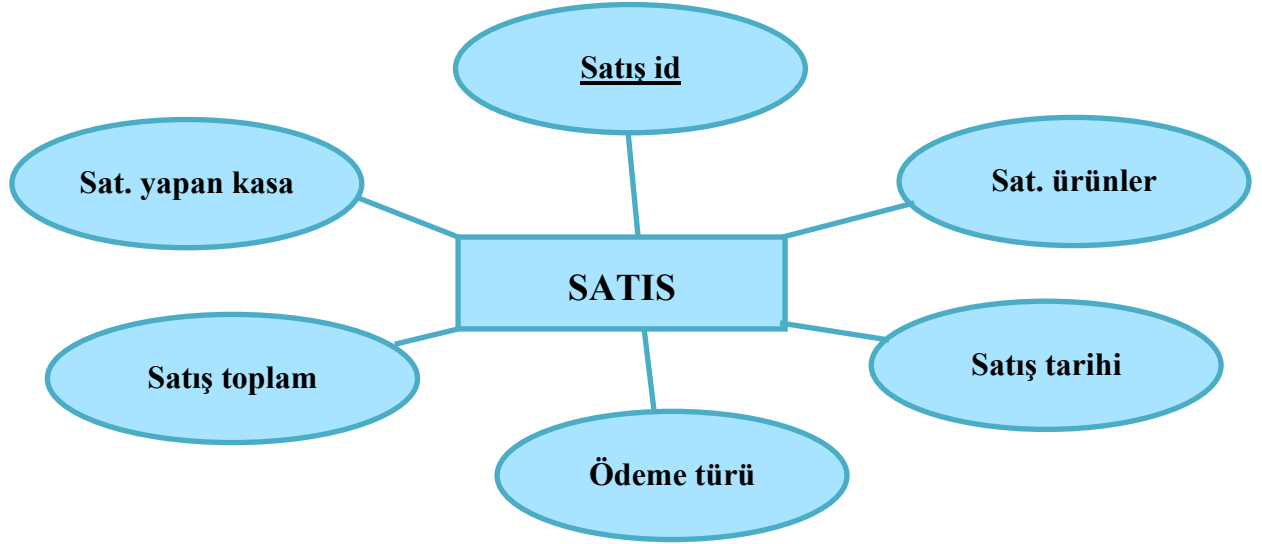
Buna göre Satış Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Satış Varlık Kümesi;

- Satış id,
- Satılan ürün adı,

- Satış tarihi,
- Satış toplam tutarı,
- Ödeme türü
- Satış yapılan kasa.

Buna göre Satış Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Satış id** anahtar nitelik olarak tanımlanmıştır.



Şekil 3: Satış Varlık Kümesi

3.4.Satılan Ürünler Varlık Kümesi

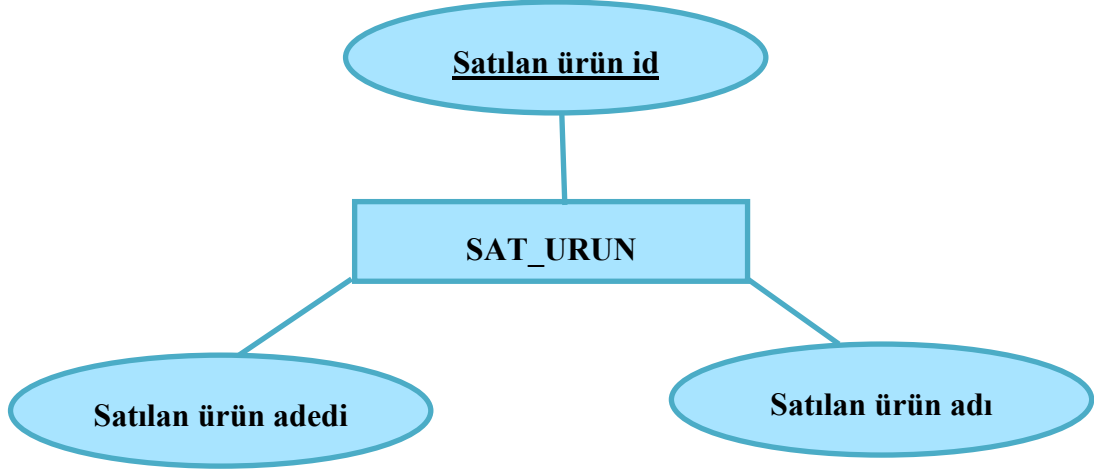
Satılan Ürünler Varlık Kümesi; birçok ürünün aynı anda satılması ile veri tabanındaki oluşacak olan bilgi tekrarını önlemek amacıyla tasarlanmıştır. Bu tasarımda; satılan ürün veya ürünlerin isimleri ve her bir üründen kaç adet satıldığı bilgisi tutulmaktadır

Buna göre Satılan Ürünler Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Satılan Ürünler Varlık Kümesi;

- Satılan ürün id,
- Satılan ürün adı,
- Satılan ürünün adedi,

Buna göre Satılan Ürünler Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Satılan ürün id** anahtar nitelik olarak tanımlanmıştır.



Şekil 4: Satılan Ürünler Varlık Kümesi

3.5.Marka Varlık Kümesi

Marka Varlık Kümesi; her bir ürüne ait marka bilgilerinin tutulması amacıyla tasarlanmıştır. Bu tasarımda; marka ismi bilgisi tutulmaktadır.

Buna göre Marka Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Marka Varlık Kümesi;

- Marka id,
- Marka adı.

Buna göre Marka Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Marka id** anahtar nitelik olarak tanımlanmıştır.



Şekil 5: Marka Varlık Kümesi

3.6.Ödeme Varlık Kümesi

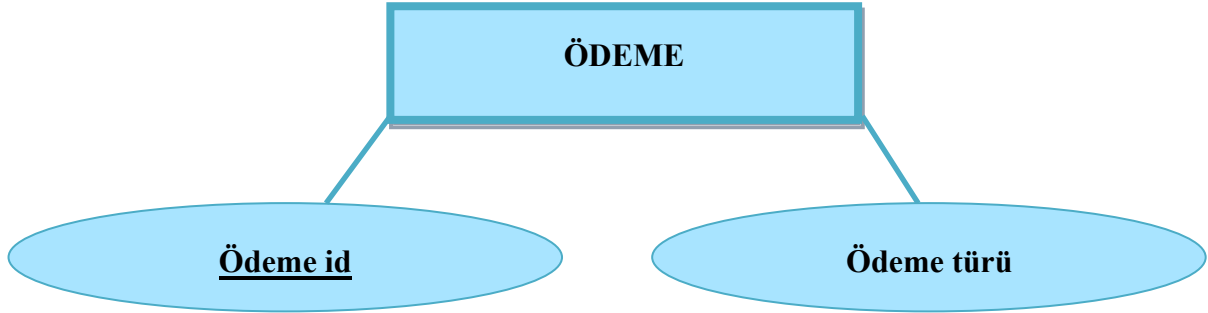
Ödeme Varlık Kümesi; müşterinin kasada yapmış olduğu ödeme türünü bilgisinin tutulması amacıyla tasarlanmıştır. Bu tasarımda; ödemenin nakit ile mi yoksa kart ile mi yapıldığının bilgisi tutulmaktadır.

Buna göre Ödeme Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Ödeme Varlık Kümesi;

- Ödeme id,
- Ödeme türü.

Buna göre Ödeme Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Ödeme id** anahtar nitelik olarak tanımlanmıştır.



Şekil 6: Ödeme Varlık Kümesi

3.7.Ciro Varlık Kümesi

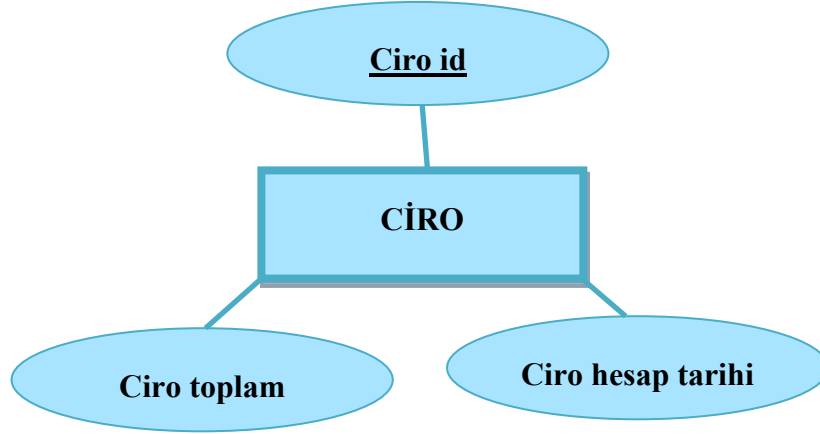
Ciro Varlık Kümesi; marketin yapmış olduğu günlük ciro hesabının bilgileri tutulması amacıyla tasarlanmıştır. Bu tasarımda; bir günde yapılan toplam ciro ve bu cironun ne zaman yapıldığı ile olduğu bilgisi tutulmaktadır.

Buna göre Ciro Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Ciro Varlık Kümesi;

- Ciro id,
- Ciro hesaplanan tarih,
- Ciro toplam.

Buna göre Ciro Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Ciro id** anahtar nitelik olarak tanımlanmıştır.



Şekil 7: Ciro Varlık Kümesi

3.8.Kar Varlık Kümesi

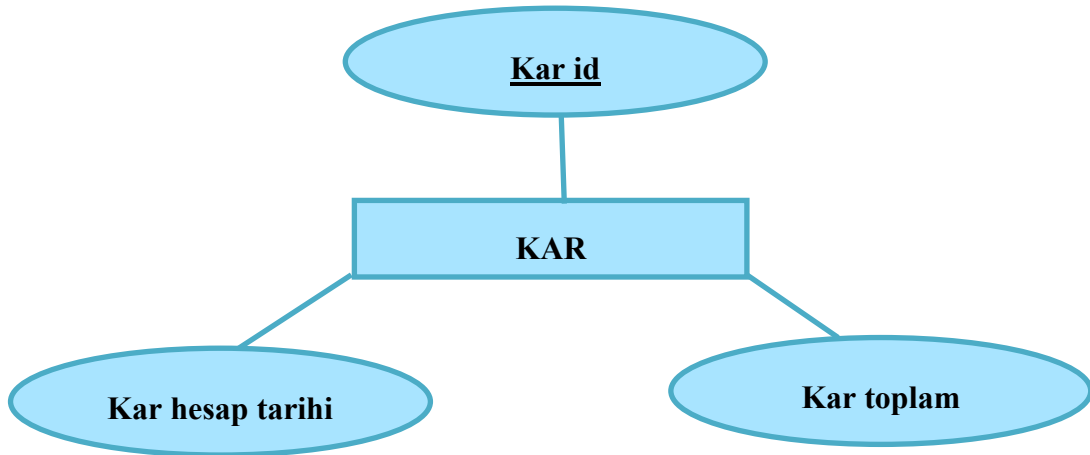
Kar Varlık Kümesi; marketin yapmış olduğu satışlardan elde edilen kâr bilgilerinin tutulması amacıyla tasarlanmıştır. Bu tasarımda; günlük elde edilen kâr bilgisi tutulmaktadır.

Buna göre Kar Varlık Kümesi aşağıdaki şekilde oluşturulmuştur.

Kar Varlık Kümesi;

- Kar id,
- Kar hesaplanan tarih,
- Kar toplam.

Buna göre Kar Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Kar id** anahtar nitelik olarak tanımlanmıştır.

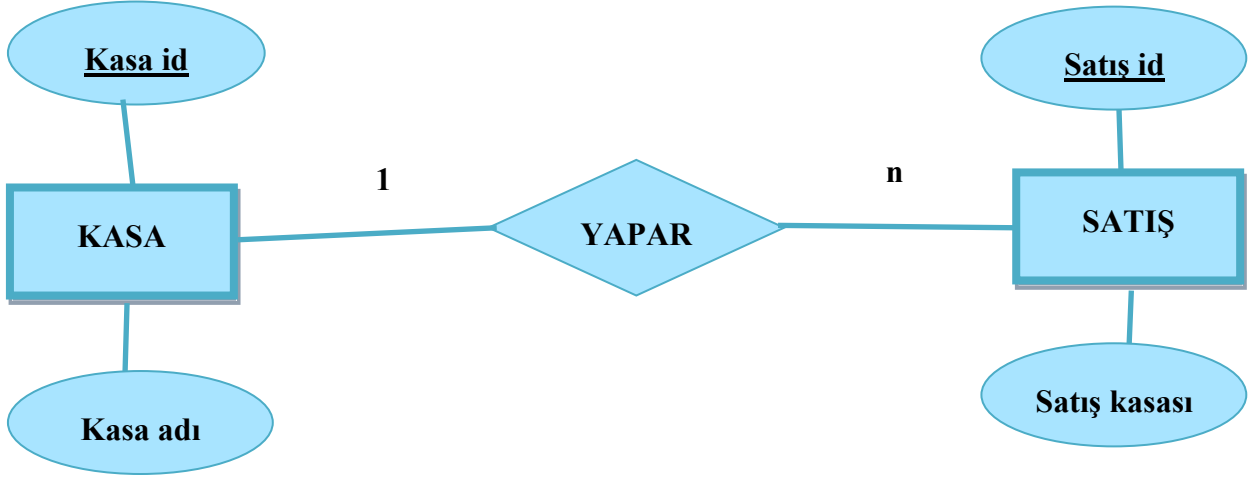


Şekil 8: Kar Varlık Kümesi

3.9.Varlık İlişkileri (ER)

3.9.1.Kasa-Satış İlişkisi

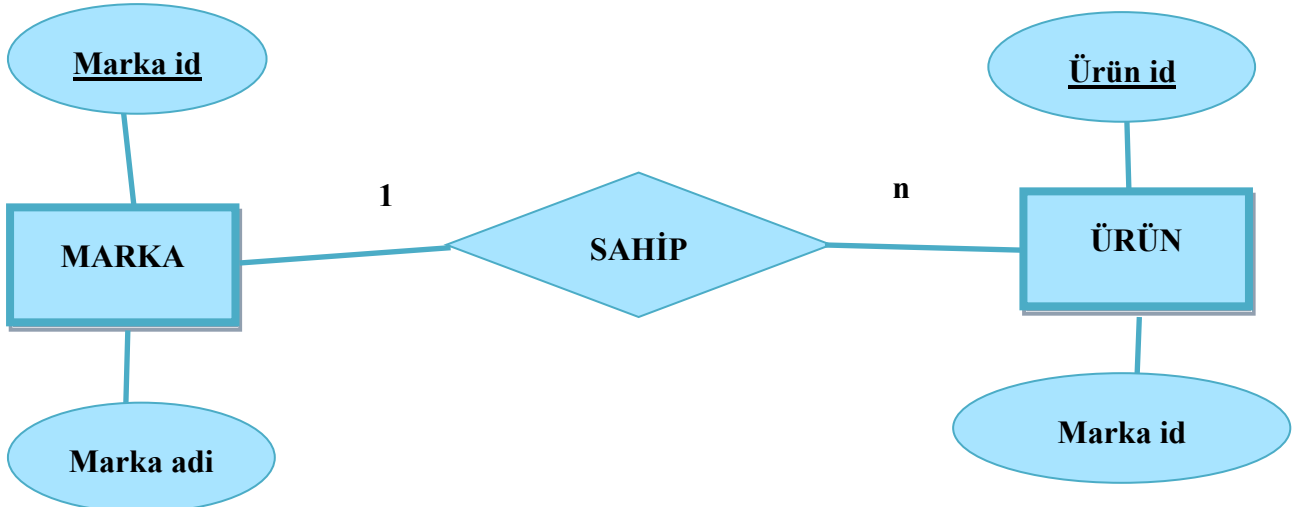
Her bir kasa satış yapmakla yetkilidir. Bir kasa birçok satış yapabilir lakin bir satıştan sadece bir kasa yükümlüdür. Bu nedenle Kasa-Satış varlık kümeleri arasındaki ilişki 1:n (bire-çok) ilişkidir. Kasa-Satış varlık kümeleri arasındaki ilişki ER şeması olarak Şekil 9’da gösterilmiştir.



Şekil 9: Kasa-Satış İlişki Şeması

3.9.2.Ürün-Marka İlişkisi

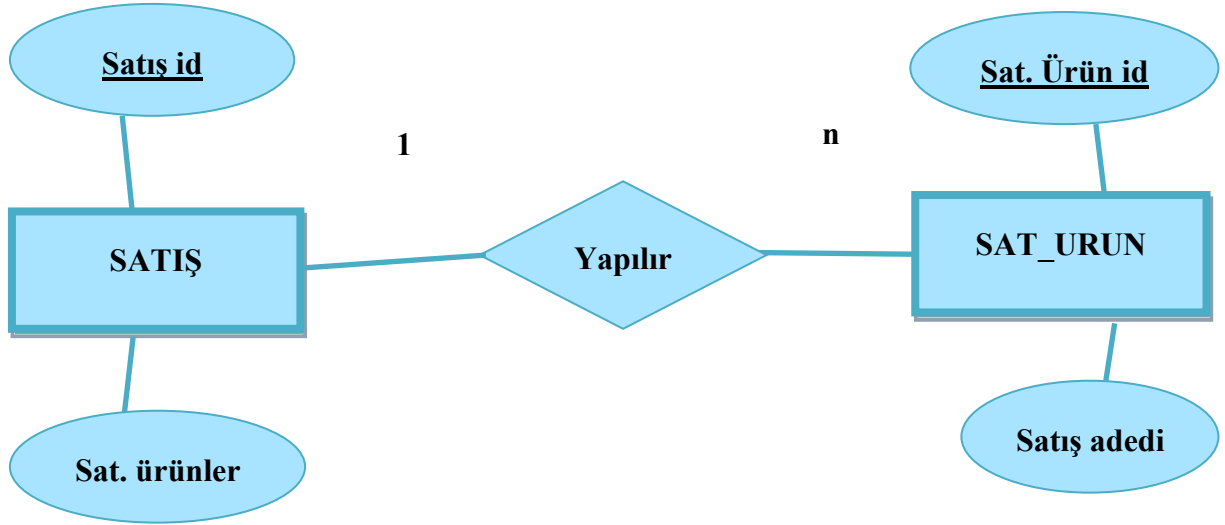
Her bir ürün bir markaya sahiptir. Bir marka birçok ürüne sahip olabilir lakin bir ürünün sadece bir markası vardır. Bu nedenle Ürün-Marka varlık kümeleri arasında 1:n(bire-çok) bir ilişki vardır. Ürün-Marka varlık kümeleri arasındaki ilişki ER şeması olarak şekil 10’da gösterilmiştir.



Şekil 10: ÜRÜN-MARKA İlişki Şeması

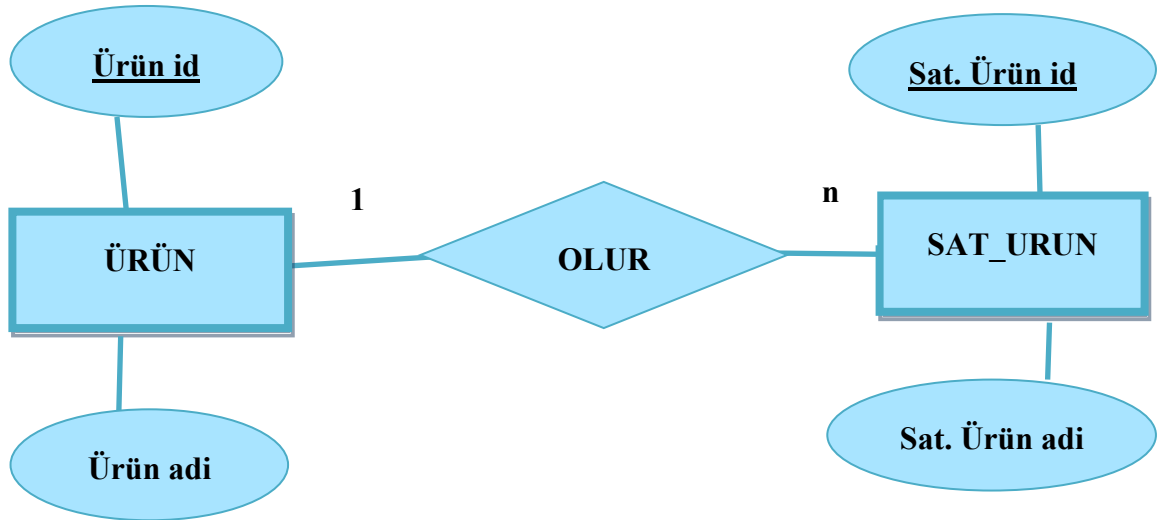
3.9.3.Satılan Ürünler-Satış İlişkisi

Bir satış yapılabilmesi için en az bir ürünün bulunması gerekir. Ürün olmadan satış da olmaz. Birçok ürünün aynı anda satışı yapıldığında, veri tabanının veri tekrarını önlemek amacıyla Satılan Ürünler varlık kümesi oluşturulmuştur. Böylece gerçekleşen bir satışta birçok ürün bulunabilir ama bir ürün sadece bir defa satılabilir. Bu nedenle Satılan ürünler-Satış varlık kümeleri arasındaki ilişki 1:n(bire-çok) ilişkidir. Satılan Ürünler-Satış varlık kümeleri arasındaki ilişki ER şeması olarak şekil 11’de gösterilmiştir.



Şekil 11: Satılan Ürünler-Satış İlişki Şeması

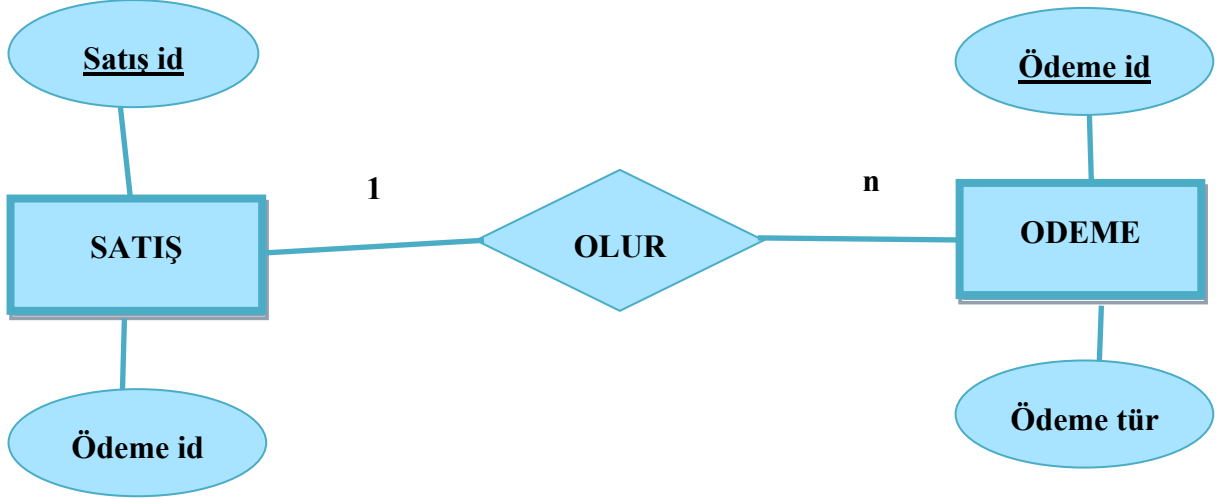
3.9.4.Satılan Ürünler-Ürün İlişkisi



Şekil 12: Satılan Ürünler-Ürün İlişki Şeması

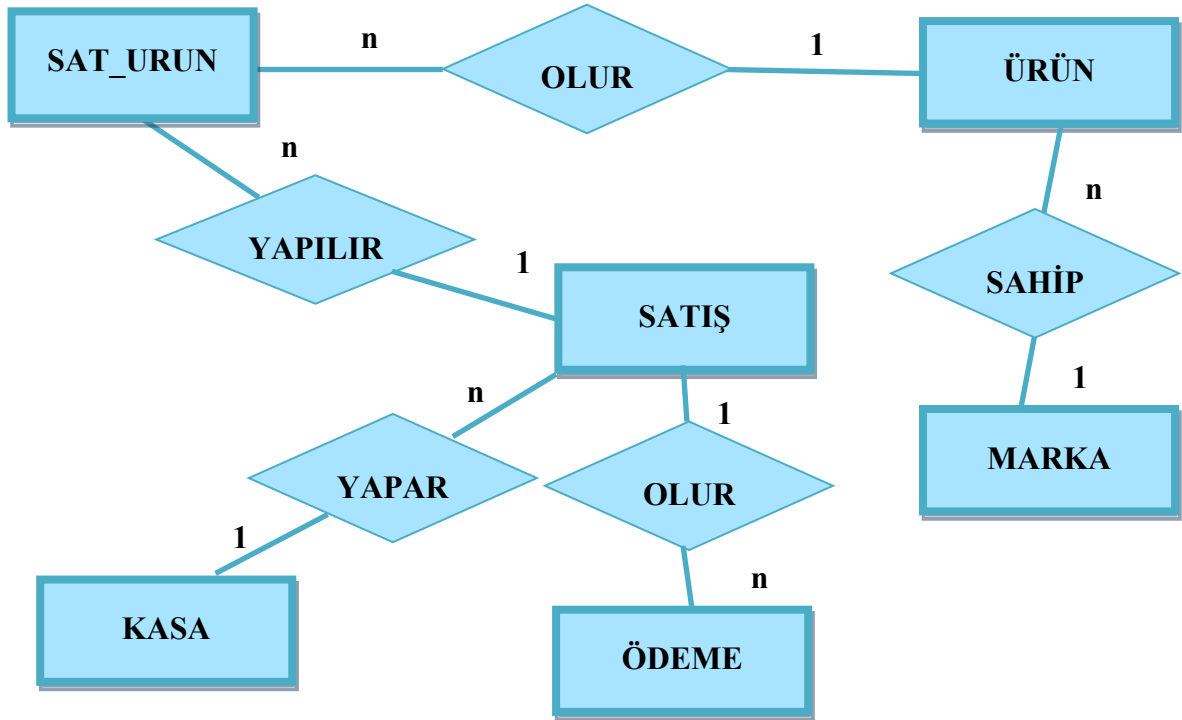
3.9.5.Satış-Ödeme İlişkisi

Yapılan her satışta bir ödeme de vardır. Ödeme yapılmadan satışta yapılmış olmaz. Bir satışta bir ödeme türü bulunur lakin bir ödeme türü ile bir çok satış yapılabilir. Bu nedenle Satış-Ödeme varlık kümeleri arasında 1:n (bire çok) ilişki vardır. Satış-Ödeme varlık kümeleri arasındaki ilişki ER şeması olarak şekil 12’de gösterilmiştir.



Şekil 13: Satış-Ödeme İlişki Şeması

3.9.6.Genel Gösterim



Şekil 14: Genel Gösterim Şeması


4.MANTIKSAL TASARIM

Market uygulaması veri tabanı tasarımında varlık kümeleri arasındaki ilişkilerin tasarlanmasından sonra mantıksal tasarım aşamasına geçilmiştir. Bu aşamada;

Kasa Tablosu;

- KASAid, int veri tipi, birincil anahtar, otomatik artan,
- KASAadi, nvarchar(20) veri tipi,
- KASAsifre, nvarchar(20) veri tipi olarak eklenmiştir.

Bu durumda Kasa Tablosu aşağıdaki şekilde tasarlanmıştır.


KASA	
	KASAid
	KASAadi
	KASAsifre

Tablo 1: Kasa Tablosu

Ürün Tablosu;

- URUNid, int veri tipi, birincil anahtar, otomatik artan,
- URUNadi, nvarchar(20) veri tipi,
- URUNadet, int veri tipi,
- URUNalfiyat, float veri tipi,
- URUNsatfiyat, float veri tipi,
- MARKAid, int veri tipi, ikincil anahtar olarak eklenmiştir.

Bu durumda Ürün Tablosu aşağıdaki şekilde tasarlanmıştır.

URUN	
	URUNid
	URUNadi
	URUNadet
	URUNalfiyat
	URUNsatfiyat
	MARKAid

Tablo 2: Ürün Tablosu

Satış Tablosu;

- SATid, int veri tipi, birincil anahtar, otomatik artan,
- SATtoplaml, float veri tipi,
- FATid, int veri tipi, ikincil anahtar,
- SATtarih, datetime veri tipi,
- Oid, int veri tipi, ikincil anahtar,
- KASAd, id veri tipi olarak eklenmiştir.

Bu durumda Satış Tablosu aşağıdaki şekilde tasarlanmıştır.

SATIS	
🔑	SATid
	SATtoplaml
	FATid
	SATtarih
	Oid
	KASAd

Tablo 3: Satış Tablosu

Satılan Ürünler Tablosu;

- FATid, int veri tipi, birincil anahtar, otomatik artan,
- URUNid, int veri tipi, ikincil anahtar,
- FATadet, int veri tipi olarak eklenmiştir.

Bu durumda Satılan Ürünler Tablosu aşağıdaki şekilde tasarlanmıştır.

SAT_URUN	
🔑	FATid
	URUNid
	FATadet


Tablo 4: Satılan Ürünler Tablosu

Marka Tablosu;

- MARKAd, int veri tipi, birincil anahtar, otomatik artan,

- MARKAadi, nvarchar(20) veri tipi olarak eklenmiştir.

Bu durumda Marka Tablosu aşağıdaki şekilde tasarlanmıştır.


MARKA	
 MARKAid	
	MARKAadi

Tablo 5: Marka Tablosu

Ödeme Tablosu;

- Oid, int veri tipi, birincil anahtar, otomatik artan,
- Oadi, nvarchar(20) veri tipi olarak eklenmiştir.

Bu durumda Ödeme Tablosu aşağıdaki şekilde tasarlanmıştır.


ODEME	
 Oid	
	Oadi

Tablo 6: Ödeme Tablosu

Ciro Tablosu;

- CIROid, int veri tipi, birincil anahtar, otomatik artan,
- CIRO, float veri tipi,
- CIROtarih, datetime veri tipi olarak eklenmiştir.

Bu durumda Ciro Tablosu aşağıdaki şekilde tasarlanmıştır.

CIRO	
 CIROid	
	CIRO
	CIROtarih


Tablo 7: Ciro Tablosu

Kar Tablosu;

- KARid, int veri tipi, birincil anahtar, otomatik artan,

- KAR, float veri tipi,
- KARTarih, datetime veri tipi olarak eklenmiştir.

Bu durumda Kar Tablosu aşağıdaki şekilde tasarlanmıştır.

KAR	
	KARid
	KAR
	KARTarih

Tablo 8: Kar Tablosu

5.FİZİKSEL TASARIM



Proje MSSQL ile tasarlanmıştır. Projede toplamda 8 adet tablo bulunmaktadır. Bulunan tabloların isimleri aşağıda listelenmiştir.

- dbo.KASA – Kasa bilgilerinin tutulduğu tablodur.
- dbo.URUN – Ürün bilgilerinin tutulduğu tablodur.
- dbo.SATIS – Satış bilgilerinin tutulduğu tablodur.
- dbo.SAT_URUN – Satılan ürün bilgilerinin tutulduğu tablodur.
- dbo.MARKA – Marka bilgilerinin tutulduğu tablodur.
- dbo.ODEME – Ödeme bilgilerinin tutulduğu tablodur.
- dbo.CIRO – Ciro bilgilerinin tutulduğu tablodur.
- dbo.KAR – Kar bilgilerinin tutulduğu tablodur.

5.1.dbo.KASA Tablosu:

Kasa Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- KASAAid, int veri tipi, birincil anahtar, otomatik artan,
- KASAadi, nvarchar(20) veri tipi,
- KASAsifre, nvarchar(20) veri tipi olarak tasarlanmıştır.



	Column Name	Data Type	Allow Nulls
	KASAAid	int	<input type="checkbox"/>
	KASAadi	nvarchar(20)	<input type="checkbox"/>
	KASAsifre	nvarchar(20)	<input type="checkbox"/>
			<input type="checkbox"/>

Tablo 9: dbo.KASA Tablosu

5.2.dbo.URUN Tablosu:

Ürün Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- URUNid, int veri tipi, birincil anahtar, otomatik artan,
- URUNadi, nvarchar(20) veri tipi,
- URUNadet, int veri tipi,
- URUNalfiyat, float veri tipi,
- URUNsatfiyat, float veri tipi,
- MARKAid, int veri tipi, ikincil anahtar olarak tasarlanmıştır.



	Column Name	Data Type	Allow Nulls
	URUNid	int	<input type="checkbox"/>
	URUNadi	nvarchar(20)	<input checked="" type="checkbox"/>
	URUNadet	int	<input checked="" type="checkbox"/>
	URUNalfiyat	float	<input checked="" type="checkbox"/>
	URUNsatfiyat	float	<input checked="" type="checkbox"/>
	MARKAid	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tablo 10: dbo.URUN Tablosu

5.3.dbo.SATIS Tablosu:

Satış Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- SATid, int veri tipi, birincil anahtar, otomatik artan,
- SATtoplam, float veri tipi,
- FATid, int veri tipi, ikincil anahtar,
- SATtarih, datetime veri tipi,
- Oid, int veri tipi, ikincil anahtar,
- KASAid, id veri tipi olarak tasarlanmıştır.



	Column Name	Data Type	Allow Nulls
	SATId	int	<input type="checkbox"/>
	SATtoplam	float	<input checked="" type="checkbox"/>
	FATId	int	<input type="checkbox"/>
	SATtarih	datetime	<input checked="" type="checkbox"/>
	Oid	int	<input type="checkbox"/>
	KASAd	int	<input type="checkbox"/>
			<input type="checkbox"/>

Tablo 11: dbo.SATIS Tablosu

5.4.dbo.SAT_URUN Tablosu:

Satılan Ürünler Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- FATId, int veri tipi, birincil anahtar, otomatik artan,
- URUNId, int veri tipi, ikincil anahtar,
- FATadet, int veri tipi olarak tasarlanmıştır.



	Column Name	Data Type	Allow Nulls
	FATId	int	<input type="checkbox"/>
	URUNId	int	<input type="checkbox"/>
	FATadet	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tablo 12: dbo.SAT_URUN Tablosu

5.5.dbo.MARKA Tablosu:

Marka Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- MARKAid, int veri tipi, birincil anahtar, otomatik artan,
- MARKAadi, nvarchar(20) veri tipi olarak tasarlanmıştır.



	Column Name	Data Type	Allow Nulls
	MARKAid	int	<input type="checkbox"/>
	MARKAadi	nvarchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tablo 13: dbo.MARKA Tablosu

5.6.dbo.ODEME Tablosu:

Ödeme Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- Oid, int veri tipi, birincil anahtar, otomatik artan,
- Oadi, nvarchar(20) veri tipi olarak tasarlanmıştır.



	Column Name	Data Type	Allow Nulls
	Oid	int	<input type="checkbox"/>
	Oadi	nvarchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tablo 14: dbo.ODEME Tablosu

5.7.dbo.CIRO Tablosu:

Ciro Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- CIROid, int veri tipi, birincil anahtar, otomatik artan,
- CIRO, float veri tipi,
- CIROtarih, datetime veri tipi olarak tasarlanmıştır.

	Column Name	Data Type	Allow Nulls
	CIROid	int	<input type="checkbox"/>
	CIRO	float	<input checked="" type="checkbox"/>
	CIROtarih	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tablo 15: dbo.CIRO Tablosu

5.8.dbo.KAR Tablosu:

Kar Tablosunun alanları ve değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

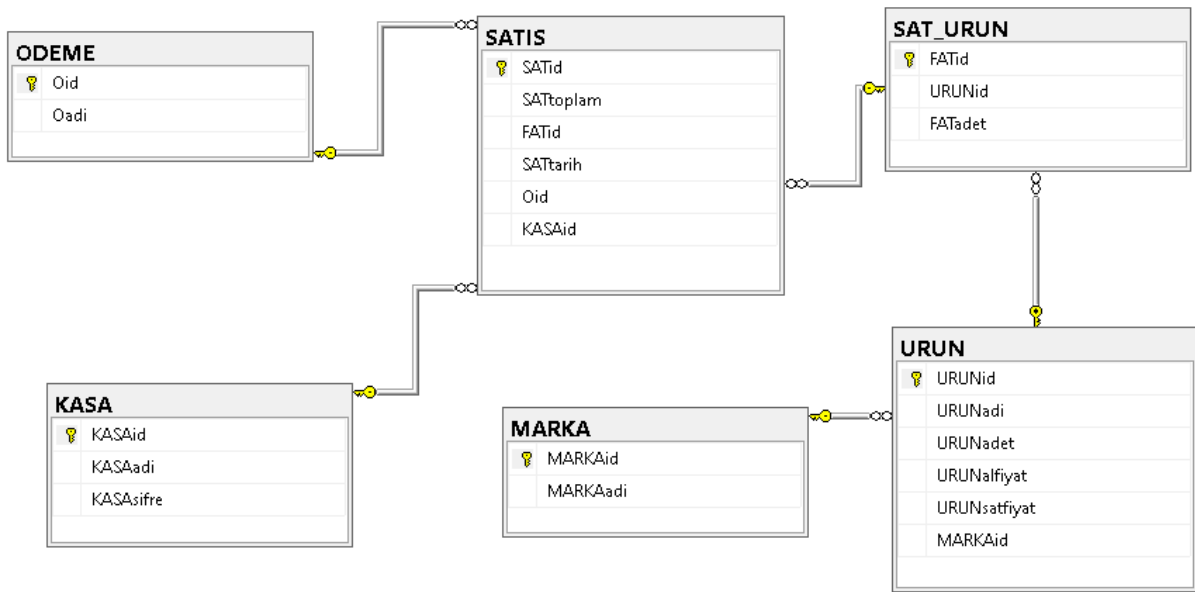
- KARid, int veri tipi, birincil anahtar, otomatik artan,
- KAR, float veri tipi,
- KARTarih, datetime veri tipi olarak tasarlanmıştır.

	Column Name	Data Type	Allow Nulls
	KARid	int	<input type="checkbox"/>
	KAR	float	<input checked="" type="checkbox"/>
	KARtarih	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tablo 16: dbo.KAR Tablosu

5.9. Veri Tabanı Diyagramı

Tablolar yukarıda belirtilen analizler ışığında aşağıda belirtildiği şekilde birbirleri ile ilişkilendirilmiş ve diyagram oluşturulmuştur.



Tablo 17: Veri tabanı Diyagramı

5.10. Tabloların SQL Kodları

Oluşturulan Tablolara ait SQL kodları aşağıda sunulmuştur.

```
USE [db_market]
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[KASA](
```



```

[KASAAid] [int] NOT NULL,
[KASAAadi] [nvarchar](20) NOT NULL,
[KASAsifre] [nvarchar](20) NOT NULL,
CONSTRAINT [PK_KASA] PRIMARY KEY CLUSTERED
(
    [KASAAid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[URUN](
    [URUNid] [int] IDENTITY(1,1) NOT NULL,
    [URUNadi] [nvarchar](20) NULL,
    [URUNadet] [int] NULL,
    [URUNalfiyat] [float] NULL,
    [URUNsatfiyat] [float] NULL,
    [MARKAid] [int] NULL,
CONSTRAINT [PK__URUN__D4628B3365F2F557] PRIMARY KEY CLUSTERED
(
    [URUNid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[URUN] WITH CHECK ADD CONSTRAINT [FK_URUN_MARKA] FOREIGN
KEY([MARKAid])
REFERENCES [dbo].[MARKA] ([MARKAid])
GO

ALTER TABLE [dbo].[URUN] CHECK CONSTRAINT [FK_URUN_MARKA]
GO
GO

SET ANSI_NULLS ON
GO

```

SET QUOTED_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[SATIS](
    [SATId] [int] NOT NULL,
    [SATtoplamlam] [float] NULL,
    [FATId] [int] NOT NULL,
    [SATtarih] [datetime] NULL,
    [Oid] [int] NOT NULL,
    [KASAIId] [int] NOT NULL,
    CONSTRAINT [PK__SATIS__40F346F78C1A7C94] PRIMARY KEY CLUSTERED
(
    [SATId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[SATIS] WITH CHECK ADD CONSTRAINT [FK_SATIS_KASA] FOREIGN
KEY([KASAIId])
REFERENCES [dbo].[KASA] ([KASAIId])
GO
```

```
ALTER TABLE [dbo].[SATIS] CHECK CONSTRAINT [FK_SATIS_KASA]
GO
```

```
ALTER TABLE [dbo].[SATIS] WITH CHECK ADD CONSTRAINT [FK_SATIS_ODEME] FOREIGN
KEY([Oid])
REFERENCES [dbo].[ODEME] ([Oid])
GO
```

```
ALTER TABLE [dbo].[SATIS] CHECK CONSTRAINT [FK_SATIS_ODEME]
GO
```

```
ALTER TABLE [dbo].[SATIS] WITH CHECK ADD CONSTRAINT [FK_SATIS_SAT_URUN] FOREIGN
KEY([FATId])
REFERENCES [dbo].[SAT_URUN] ([FATId])
GO
```

```
ALTER TABLE [dbo].[SATIS] CHECK CONSTRAINT [FK_SATIS_SAT_URUN]
GO
```

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[SAT_URUN](
    [FATId] [int] NOT NULL,
    [URUNId] [int] NOT NULL,
    [FATadet] [int] NULL,
    CONSTRAINT [PK__SAT_URUN__4A4AEA354B86BB95] PRIMARY KEY CLUSTERED
(
    [FATId] ASC
)
```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[SAT_URUN] WITH CHECK ADD CONSTRAINT [FK_SAT_URUN_URUN] FOREIGN
KEY([URUNId])
REFERENCES [dbo].[URUN] ([URUNId])
GO
ALTER TABLE [dbo].[SAT_URUN] CHECK CONSTRAINT [FK_SAT_URUN_URUN]
GO
GO
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MARKA](
    [MARKAid] [int] NOT NULL,
    [MARKAadi] [nvarchar](20) NULL,
    CONSTRAINT [PK__MARKA__FD829AF52E700438] PRIMARY KEY CLUSTERED
(
    [MARKAid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[ODEME](
    [Oid] [int] NOT NULL,
    [Oadi] [nvarchar](20) NULL,
    CONSTRAINT [PK__ODEME__CB3E4F314ABC5387] PRIMARY KEY CLUSTERED
(
    [Oid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[CIRO](
    [CIROid] [int] NOT NULL,
    [CIRO] [float] NULL,
    [Oid] [int] NULL,
    [CIROtarih] [datetime] NULL,
    CONSTRAINT [PK__CIRO__6F835337436D9F83] PRIMARY KEY CLUSTERED
(
    [CIROid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[CIRO] WITH CHECK ADD CONSTRAINT [FK_CIRO_ODEME] FOREIGN KEY([Oid])
REFERENCES [dbo].[ODEME] ([Oid])
GO

ALTER TABLE [dbo].[CIRO] CHECK CONSTRAINT [FK_CIRO_ODEME]
GO

GO

SET ANSI_NULLS ON
GO

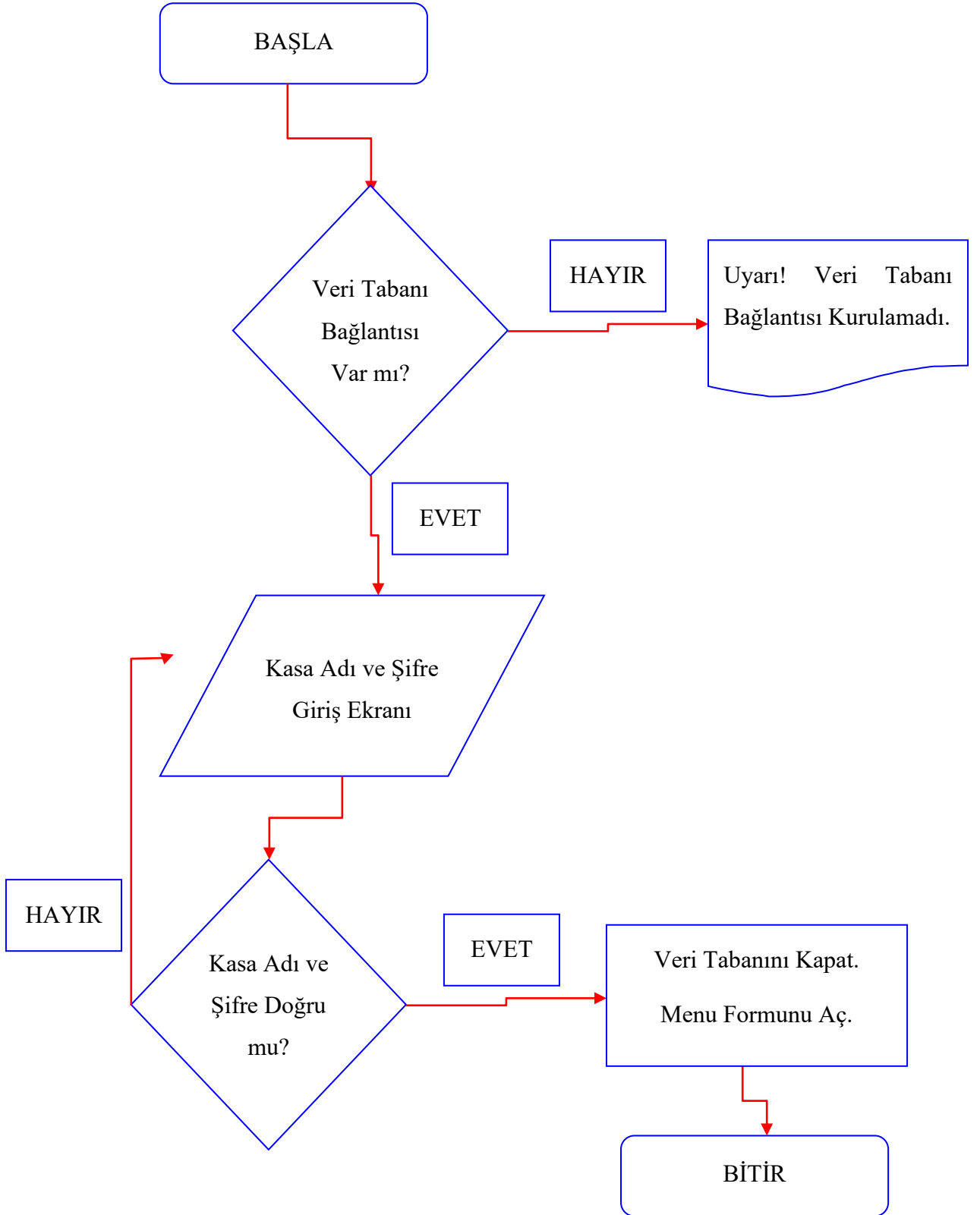
SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[KAR](
    [KARid] [int] NOT NULL,
    [KAR] [float] NULL,
    [KARtarih] [datetime] NULL,
    CONSTRAINT [PK__KAR__D17869E86B873EE4] PRIMARY KEY CLUSTERED
(
    [KARid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

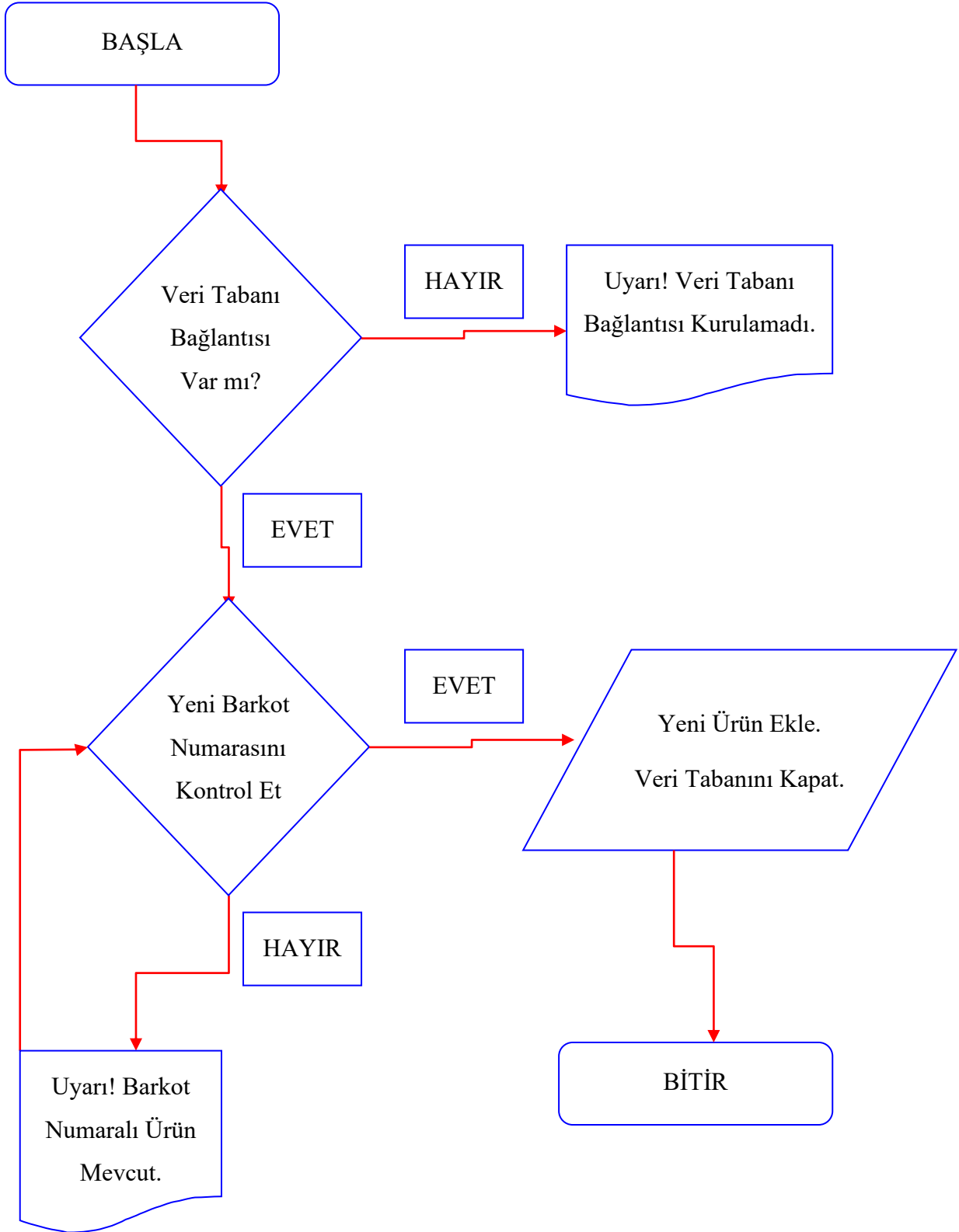
6.PROJENİN AKIŞ ŞEMASI

6.1.Kasa Giriş Formu Akış Şeması



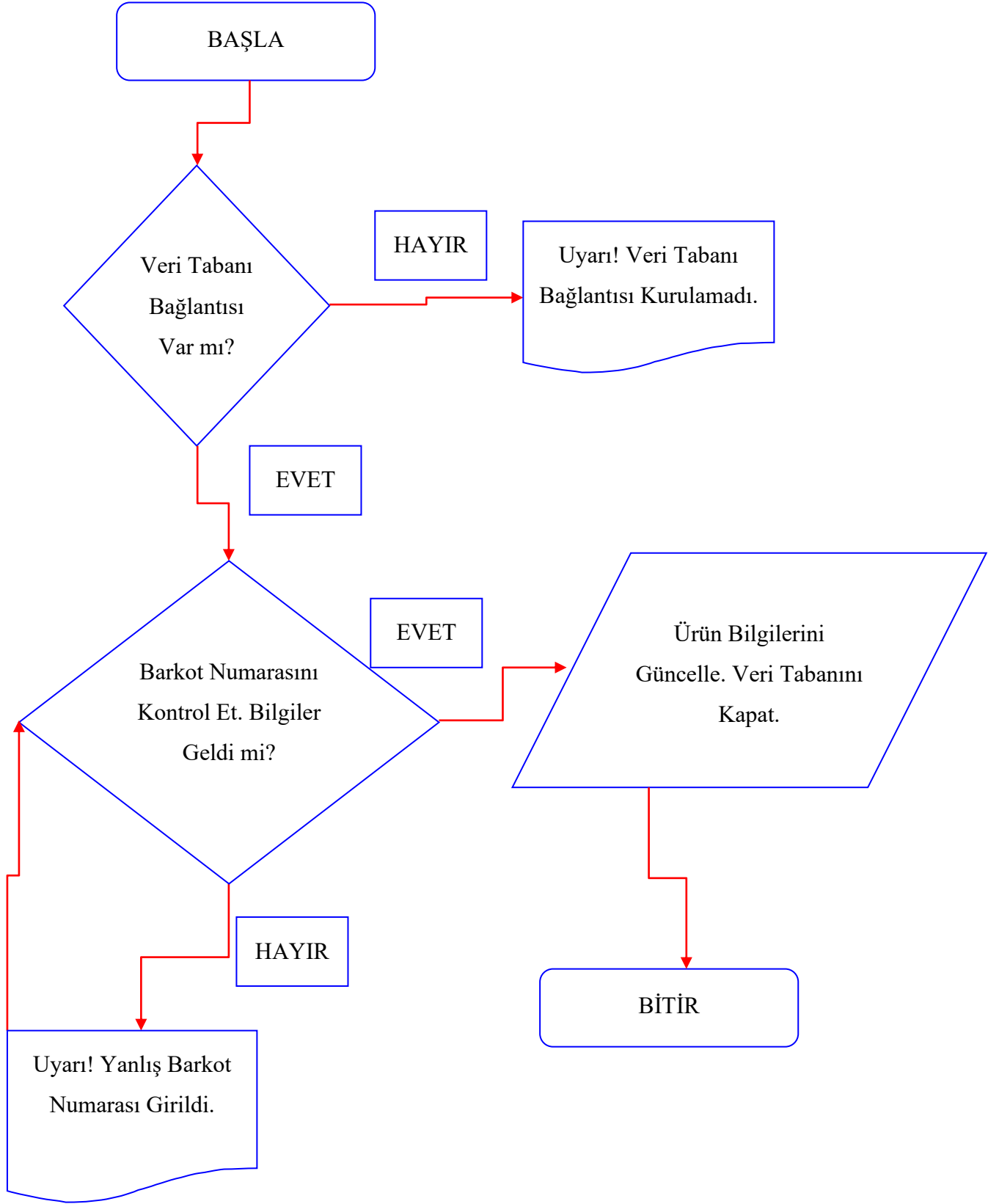
Şekil 15: Kasa Giriş Formu Akış Şeması

6.2.Ürün Ekleme Formu Akış Şeması



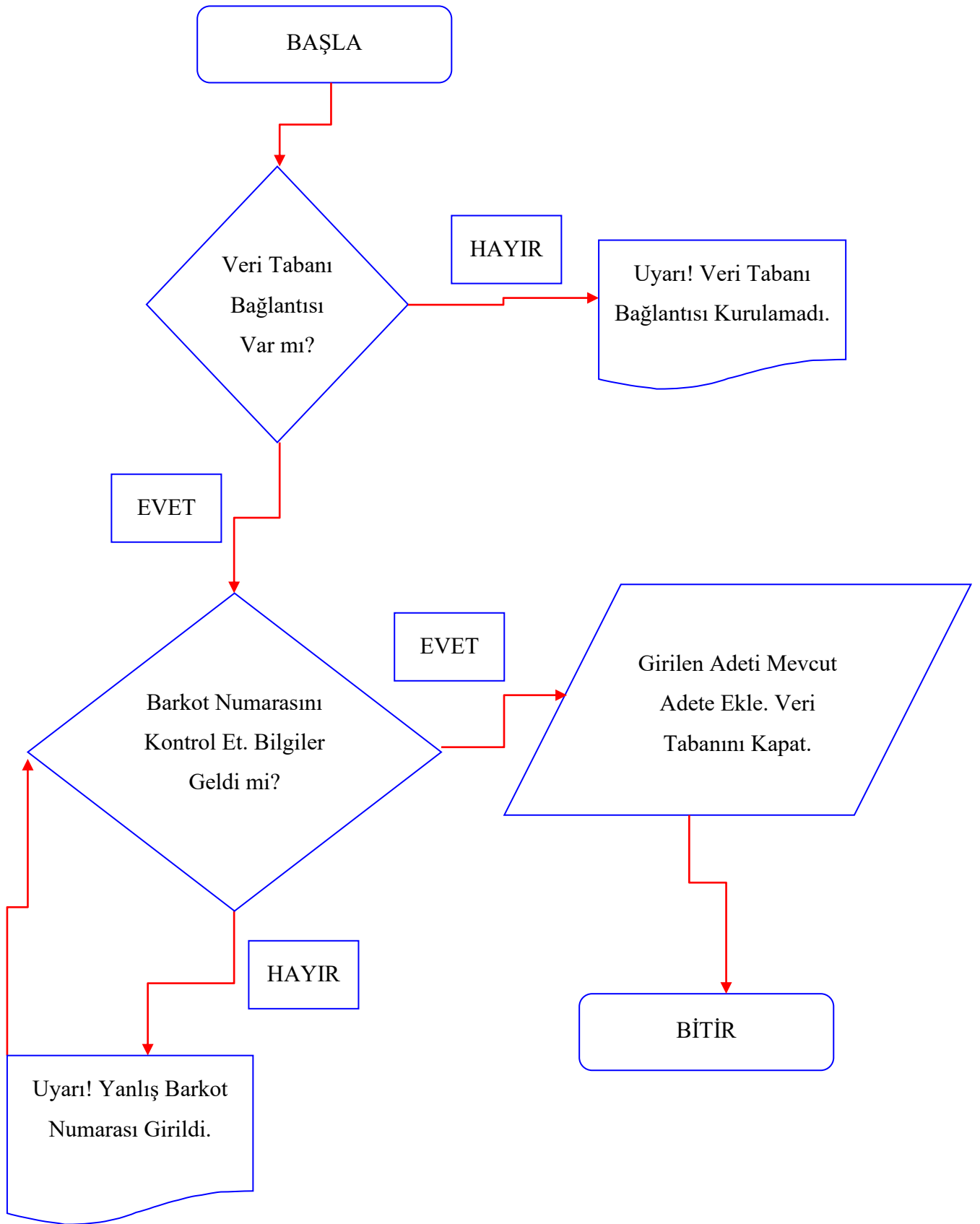
Şekil 16: Ürün Ekleme Formu Akış Şeması

6.3.Ürün Güncelleme Formu Akış Şeması



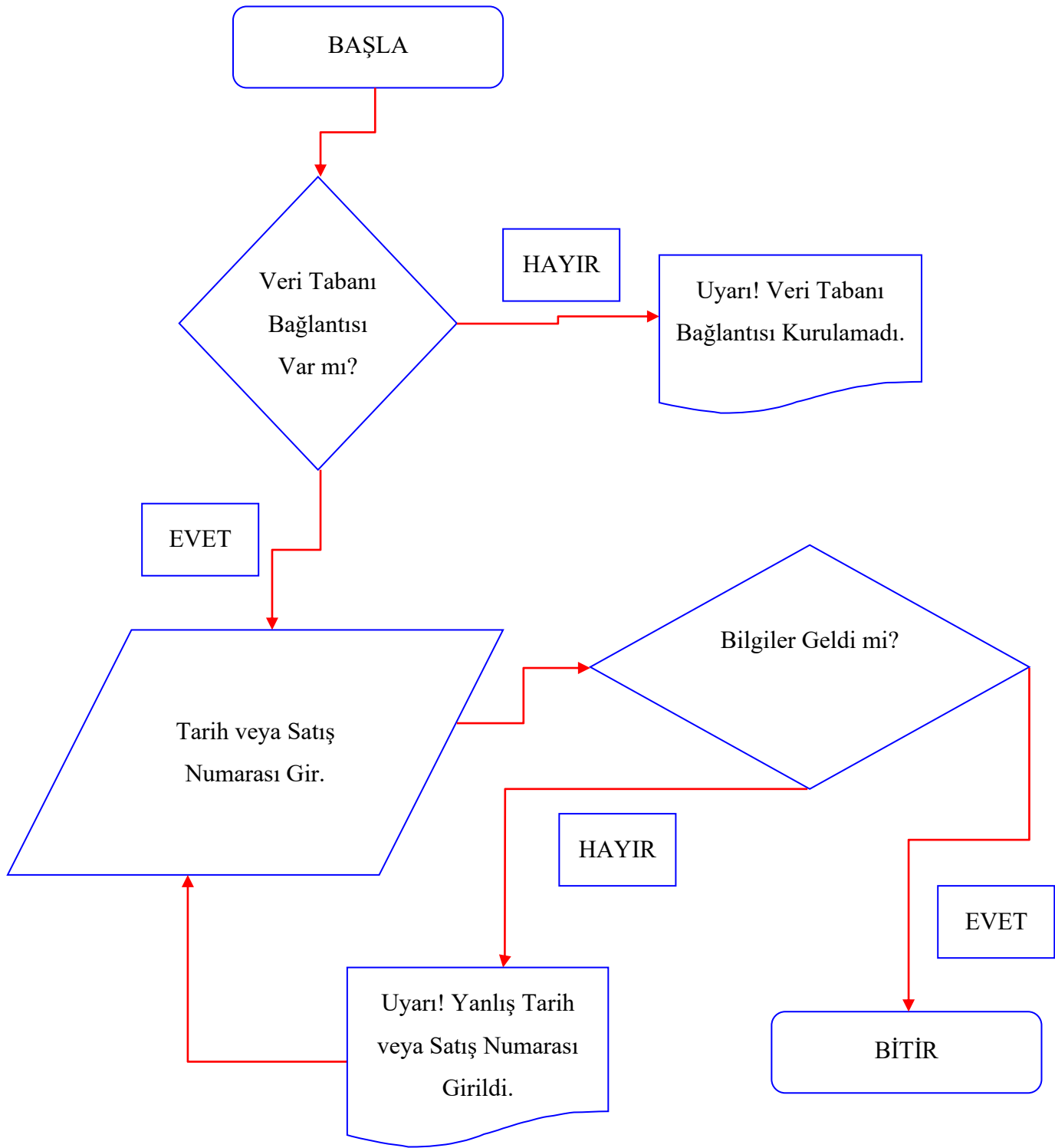
Şekil 17: Ürün Güncelleme Formu Akış Şeması

6.4. Stok Ekleme Formu Akış Şeması



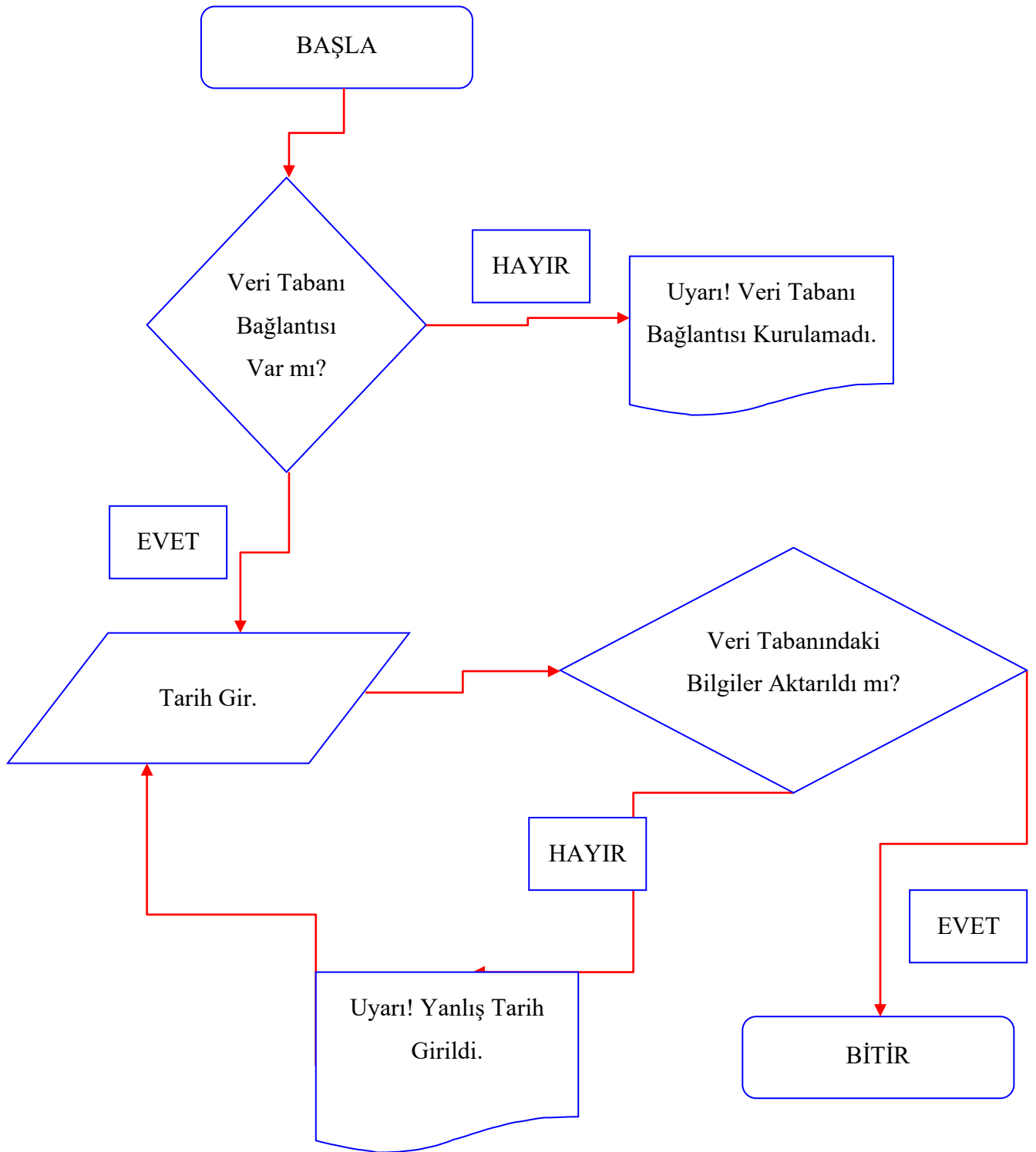
Şekil 18: Stok Ekleme Formu Akış Şeması

6.5. Satış Geçmişi Formu Akış Şeması



Şekil 19: Satış Geçmişi Formu Akış Şeması

6.6. Rapor Formu Akış Şeması



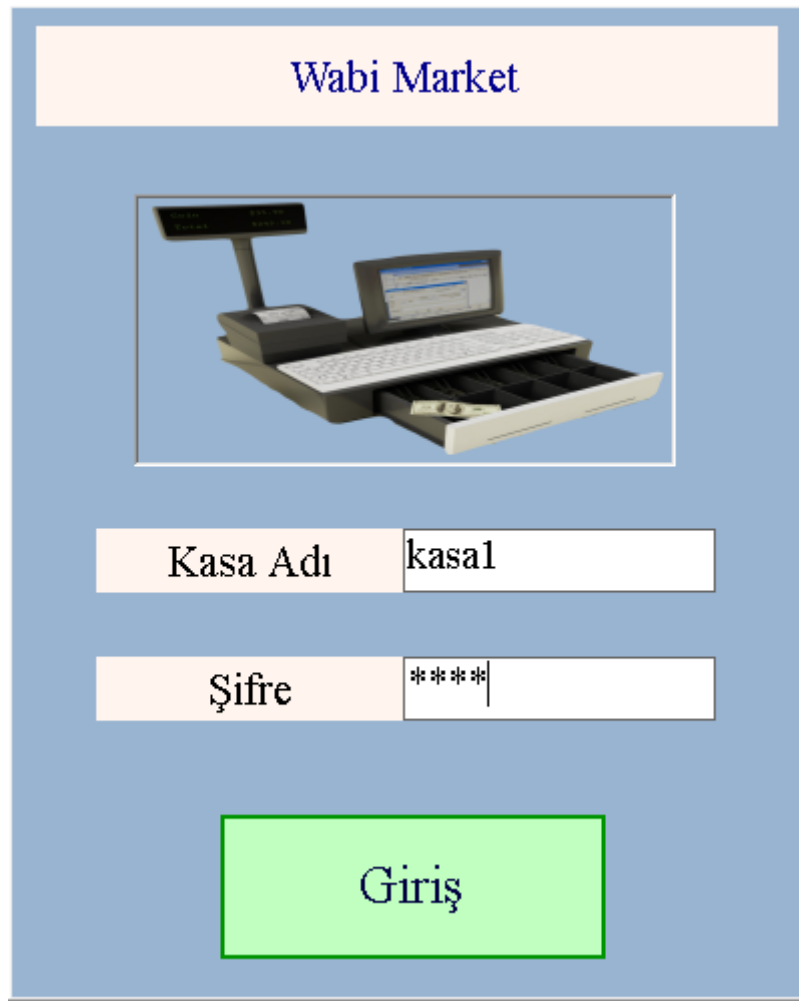
Şekil 20: Rapor Formu Akış Şeması

7.PROJENİN UYGULAMA KISMI

Projenin uygulaması C# dili ile hazırlanmıştır. Veri tabanı olarak ise MSSQL kullanılmıştır. Uygulamada bütün işlemler formlar aracılığı ile yapılmaktadır. Formlar aracılığı ile yapılan işlemler aşağıda bahsedilmiştir.

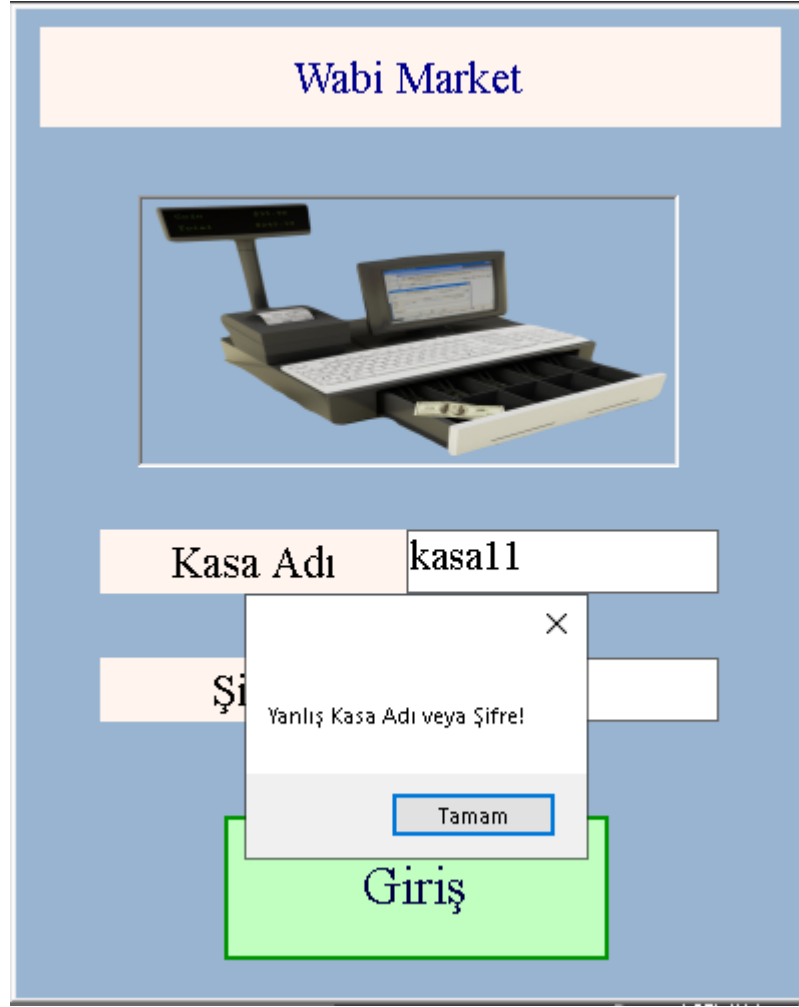
7.1. Kasa Giriş Ekranı

Kasa giriş ekranında; mevcut kasaya ait bilgiler girilerek, kasanın kullanıma hazır olması hedeflenmiştir. MSSQL ile bağlantı kurularak veri tabanında kayıtlı kasa adı ve şifresi eşleştirilerek programa giriş sağlanmıştır.



Şekil 21: Kasa Giriş Ekranı

Eğer girilen veriler, veri tabanındaki veriler ile eşleşiyor ise; program, ana menü açılarak çalışmaya devam eder. Lakin yanlış bir kasa adı veya şifre girilir ise program hata mesajı verir.



Şekil 22: Kasa Giriş Ekranı

7.2. Menü Ekranı

Kasa adı ve şifre doğru girildiğinde ana menü ekranına yönlendirilmektedir. Bu ana menü; kullanıcının menülere kolayca erişebilmesini ve işlemlerini hızlı bir şekilde gerçekleştirebilmesi için tasarlanmıştır. Ana menüde mevcut olan menüler aşağıdaki gibidir.

- Satış Ekranı
- Yeni Ürün Ekle
- Mevcut Ürün Bilgilerini Değiştir
- Toplu Stok İşlemleri
- Satış Geçmişi
- Kâr ve Ciro Geçmişi
- Stok Bilgileri ve
- Çıkış butonları bulunmaktadır.



Şekil 23: Menü Ekranı

7.3. Satış Ekranı

Ana menü ekranından sonra satış ekranı seçilerek bu ekrana ulaşılır. Programın en çok kullanılacağı ekrandır. Barkod okuyucu yardımıyla veya barkod numarası el ile girilerek veri tabanından ürün ile ilgili bilgiler ürün adı ve ürün fiyatı kısmına getirilir.

	BarkodNo	Ürün
Barkod Numarası	1	
Ürün Adı	Ekmek	
Ürün Adedi		
Ürün Fiyatı	1	
Toplam		
 EKLE	 SİL	 SİL


Şekil 24: Satış Ekranı

Kullanıcı bu ekranda; ürün adedi kısmına satılacak olan ürün miktarını girdikten sonra toplam kısmı, adet ve fiyat çarpılarak, program tarafından otomatik olarak doldurulur. Ekranda gözüken butonların yaptığı işlemler aşağıdaki gibidir.

- Ekle butonu ile satılacak ürün tabloya eklenir.
- Sil butonu ile tablodaki herhangi bir ürünü seçerek silinir.
- Satış iptal butonu ile tüm tablodaki veriler temizlenir ve yeni bir satış işlemine başlanır.
- Ödeme türü olarak nakit veya kredi kartı seçilerek satış yapma işlemi sonlandırılır.
- Çıkış butonu ile ana menüye dönülür.

Ayrıca tablodaki ürünlerin satış toplamları ekranın en alt orta kısmında genel toplam olarak gösterilir.

Wabi Market



Barkod Numarası23


Ürün AdıLimonata


Ürün Adedi1


Ürün Fiyatı3


Toplam3


BarkodNo	Ürün	Marka	Adet	SatışFiyatı
1	Ekmek	Yok	3	3
7	Burçak	Eti	2	5
23	Limonata	Torku	1	3


 EKLE

 SİL

 SATIŞ İPTAL

☒ NAKİT 

☐ KREDİ KARTI 

 ÇIKIŞ


GENEL TOPLAM : 11TL

SATIŞ YAP

Şekil 25: Satış Ekranı

Satış yap butonu ile satış işlemi gerçekleştirilir. Bu işlem gerçekleştirilmeden önce kullanıcıdan bir onaylama penceresi ile satışı onaylaması istenilir.

Wabi Market



Barkod Numarası


Ürün Adı


Ürün Adedi


Ürün Fiyatı

Toplam

BarkodNo	Ürün	Marka	Adet	SatışFiyatı
1	Ekmekek	Yok	3	3
4	Karpuz	Yok	1	5
24	Sirke	Torku	1	20


 **EKLE**

 **SİL**

 **SATIŞ İPTAL**

☒ **NAKİT**

☐ **KREDİ KARTI**

 **ÇIKIŞ**

GENEL TOPLAM : 28TL

SATIŞ YAP

Şekil 26: Satış Ekranı

Eğer hayır butonuna basılırsa herhangi bir şey olmaz ve mevcut durumdaki satış ekranında, ürün ekleme veya çıkarma gibi işlemler yapılabilir. Evet tuşuna basıldığında ise aşağıdaki olaylar gerçekleşir.

- Satışı gerçekleşen ürünler ve bilgileri, satış ve satılan ürünler tablosuna eklenir.
- Ürünler tablosundan ürün adeti çekilerek, satılan ürün adeti kadar azaltılıp tekrar veri tabanında güncellenir. Böylelikle ürün stokları güncellenmiş olur.
- Ekrandaki genel toplam değeri alınarak günlük yapılan cironun hesaplanması için ciro tablosuna gönderilir. Eğer günün ilk cirosu eklenecek ise, tabloya yeni bir satır eklenir, gün içerisinde daha önceden eklenmiş ciro değeri var ise, bu ciro ile mevcut satış getirisi toplanarak, o günkü ciro değeri güncellenir.
- Satışı gerçekleştirilen her ürünün alış fiyatı ile satış fiyatı arasındaki fark, veri tabanındaki kâr tablosuna günlük olarak eklenir. Bu işlem yapılırken günlük değerler baz alınır. Ciro değerinde olduğu gibi, gün içerisindeki ilk satışta, hesaplanan kâr eklenir, sonraki satışlarda ise kâr sürekli güncellenir.

7.4. Ürün Ekleme Ekranı

Ana menüden yeni ürün ekle butonuna tıklayarak bu ekrana ulaşılır. Daha önce eklenmemiş olan ürünleri veri tabanına eklemek için kullanılır. Marka kısmında daha önceden girilmiş markalar veri tabanından çağırılarak gösterilir. Eğer daha önce girilmeyen bir marka adı girilirse, bu marka veri tabanında marka tablosuna kaydedilir. Daha sonra bütün ürün bilgileri eksiksiz girilerek ekle butonuna tıklanır ve yeni ürün veri tabanına kaydedilir.

Wabi Market

Barkod Numarası

1

Ürün Adı

Ekmek

Ürün Adedi

12

Ürün Alış Fiyatı

1

Ürün Satış Fiyatı

3

Marka

Yeni Marka

+

EKLE

BarkotNo	Ürün	SatışFiyatı	Marka
	Yok		
	Ülker		
	Eti		
	Torku		

×

ÇIKIŞ

Şekil 27: Yeni Ürün Ekle Ekranı

Eğer daha önce eklenmiş bir ürünün barkod numarası girilirse, program hata verir. Kullanıcı yeni bir barkod girerek ekleme işlemine devam edebilir.

Wabi Market

Barkod Numarası

Ürün Adı

Ürün Adedi

Ürün Alış Fiyatı

Ürün Satış Fiyatı

Marka

EKLE

BarkotNo	Ürün	Adet	SatışFiyatı	Marka

ÇIKIŞ

Şekil 28: Yeni Ürün Ekle Ekranı

Eklenen ürünler tabloda gösterilir. Ekle butonunun altında ise “Ürün Eklendi” yazısı belirir. Böylelikle ürün sorunsuz şekilde eklenmiş olup, yeni ürün eklemeye devam edilebilir. Çıkış butonu ile ana menüye dönülür.

Wabi Market

Barkod Numarası

Ürün Adı

Ürün Adedi

Ürün Alış Fiyatı

Ürün Satış Fiyatı

Marka

EKLE

Ürün Eklendi

BarkotNo	Ürün	Adet	SatışFiyatı	Marka
▶ 30	Çikolata	10	2	Torku

ÇIKIŞ

Şekil 29: Yeni Ürün Ekle Ekranı

7.5. Ürün Bilgilerini Güncelleme Ekranı

Ana menüden mevcut ürün bilgilerini güncelle butonuna tıklayarak bu ekrana ulaşılır. Daha önceden eklenmiş ürünler hakkında bilgileri değiştirmek için bu ekran kullanılır. Barkod numarası girilen ürünlerin bilgileri, veri tabanından çekilerek otomatik olarak ekrandaki kutucuklara doldurulur.

Wabi Market

Barkod Numarası

30

Ürün Adı

Çikolata

Ürün Adedi

25

Ürün Alış Fiyatı


1

Ürün Satış Fiyatı


2

Marka

Torku

 GÜNCELLE

BarkotNo	Ürün	Adet	AlışFiyatı	SatışFiyatı	Marka

 ÇIKIŞ

Şekil 30: Ürün Bilgilerini Güncelleme Ekranı

Bu kutucuklardaki bilgileri değiştirdikten sonra güncelle butonuna basarak bilgiler güncellenir ve ürünün güncel hali tabloda gösterilir. Çıkış butonu ile ana menüye dönülür.

Wabi Market

Barkod Numarası

Ürün Adı

Ürün Adedi

Ürün Alış Fiyatı

Ürün Satış Fiyatı

Marka

GÜNCELLE

	BarkotNo	Ürün	Adet	AlışFiyatı	SatışFiyatı	Marka
►	30	Çikolata	25	1,5	2,5	Torku

ÇIKIŞ

Şekil 31: Ürün Bilgilerini Güncelleme Ekranı

7.6. Toplu Stok Ekranı

Ana menüden toplu stok işlemleri butonuna tıklayarak bu menüye ulaşılır. Veri tabanındaki ürünlere hızlı şekilde stok eklemek için kullanılır. Barkod numarası girilen ürüne, stoka eklenecek miktar girilerek, stok ekle butonuna tıklanır ve mevcut stoğun üstüne girilen miktar eklenerek veri tabanında güncellenir. Çıkış butonu ile ana menüye dönülür.

Wabi Market

Barkod Numarası

24

Ürün Adı

Sirke

Ürün Adedi Gir

3

+

STOK EKLE

	BarkodNo	Ürün	Marka	Adet
►	1	Ekmek	Yok	128
	12	Çokoprens	Ülker	13

✖

ÇIKIŞ

Şekil 32: Toplu Stok Ekranı

7.7. Satış Geçmişi Ekranı


Ana menüden satış geçmişi butonuna tıklayarak bu menüye ulaşılır. Veri tabanında kayıt altına alınmış olan satışları tarihe veya satış koduna göre aratmak için kullanılır. Aratılan satış ile ilgili bilgiler tabloda gösterilir. Çıkış butonu ile ana menüye dönülür.

Wabi Market

Tarih Seçiniz 14 Mayıs 2020 Perşembe ▾

Satış No Giriniz

 **Tarihe Göre Ara**

 **Satış Koduna Göre Ara**

Satış Tarihi	Satış Toplam	Ödeme Tipi	Kasa Adı
14.05.2020 13:47	15	Nakit	kasa1


 **ÇIKIŞ**


Şekil 33: Satış Geçmişi Ekranı

Wabi Market


Tarih Seçiniz 9 Haziran 2020 Salı ▾

Satış No Giriniz

 **Tarihe Göre Ara**

 **Satış Koduna Göre Ara**

Satış Tarihi	Satış Toplam	Ödeme Tipi	Kasa Adı
15.05.2020 16:01	25,5	Kredi Kartı	kasa2

 **ÇIKIŞ**


Şekil 34: Satış Geçmişi Ekranı


7.8. Kâr ve Ciro Rapor Ekranı

Ana menü ekranından kâr ve ciro raporu butonuna tıklanarak bu ekrana ulaşılır. Günlük kayıt altına alınan bu raporlar, veri tabanından tarih seçilerek çekilir ve tabloya yansıtılır. Çıkış butonu ile ana menüye dönülür.


Wabi Market

Tarih Seçiniz 15.05.2020 ▾

Ciro Getir

Kâr Getir

Rapor	Tarih
► 25,5	15.05.2020

ÇIKIŞ

Şekil 35: Kâr ve Ciro Rapor Ekranı

7.9. Stok Bilgileri Ekranı

Ana menü ekranından stok bilgileri butonuna basılarak bu ekrana ulaşılır. Veri tabanına kayıtlı ürünlerin stok bilgileri; az kalan stoklar sıralanarak veya barkod numarası ile aratılarak istenilen ürünün mevcut stoku tabloya yansıtılır. Çıkış butonu ile ana menüye dönülür.

Wabi Market

Barkod Numarası

Barkod

Numarası ile Ara

Az Kalan

Stokları Ara

	ÜrünAdı	Marka	ÜrünAdeti
▶	Limonata	Torku	30

ÇIKIŞ

Şekil 36: Stok Bilgileri Ekranı

Wabi Market

Barkod Numarası

Barkod

Numarası ile Ara

Az Kalan

Stokları Ara

	ÜrünAdı	Marka	ÜrünAdeti
▶	Domates	Yok	10
	Biber	Yok	6
	Karpuz	Yok	3
	Kavun	Yok	4
	Pötibör	Ülker	2
	Sirke	Torku	5

ÇIKIŞ

Şekil 37: Stok Bilgileri Ekranı

8.PROJENİN KODLARI

8.1.Kasa Giriş Ekranı Kodları

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Market

{

    public partial class giris : Form

    {

        public giris()

        {

            InitializeComponent();

            textBox2.PasswordChar = '*';

        }

        public static string bagla = "Data Source = (localdb)\\MSSQLLocalDB;Initial Catalog = db_market; Integrated Security = True";

        SqlConnection con = new SqlConnection(bagla);

        public static string kasaadi;
```

```

private void button1_Click(object sender, EventArgs e)
{
    kasaadi = textBox1.Text;

    if (kasaadi=="")
    {
        MessageBox.Show("Kasa adı boş bırakılamaz!");
    }
    else
    {
        con.Open();

        SqlCommand komut = new SqlCommand("SELECT * FROM KASA where
KASAadi = @n and KASAsifre =@m", con);

        komut.Parameters.AddWithValue("@n", kasaadi);
        komut.Parameters.AddWithValue("@m", textBox2.Text);

        SqlDataReader a = komut.ExecuteReader();

        Boolean b = a.Read();

        con.Close();

        if (b == true)
        {
            this.Hide();

            menu go = new menu();

            go.Show();

        }
        else
        {

```

```

        MessageBox.Show("Yanlış Kasa Adı veya Şifre!");

    }

}

}

}

```

8.2.Menü Ekranı Kodları

```

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Market
{
    public partial class menu : Form
    {
        public menu()
        {
            InitializeComponent();
        }
    }
}

```

```

private void button10_Click(object sender, EventArgs e)
{
    DialogResult Cikis;

    Cikis = MessageBox.Show("Uygulama Kapatılacak Emin misiniz?", "Kapatma
    Uyarısı!", MessageBoxButtons.YesNo);

    if (Cikis == DialogResult.Yes)
    {
        Application.Exit();
    }

    if (Cikis == DialogResult.No)
    {
        }
    }
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    this.Close();

    satis go = new satis();

    go.Show();
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}

```

```
    urunekle go = new urunekle();  
    go.Show();  
}
```

```
private void button3_Click(object sender, EventArgs e)  
{  
    this.Close();  
    urunguncelle go = new urunguncelle();  
    go.Show();  
}
```

```
private void button4_Click(object sender, EventArgs e)  
{  
    this.Close();  
    stokbilgi go = new stokbilgi();  
    go.Show();  
}
```

```
private void button7_Click(object sender, EventArgs e)  
{  
    this.Close();  
    stokekle go = new stokekle();  
    go.Show();  
}
```

```
private void button9_Click(object sender, EventArgs e)
```

```

    {
        this.Close();

        satisgecmis go = new satisgecmis();

        go.Show();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        this.Close();

        rapor go = new rapor();

        go.Show();
    }
}

```

8.3.Satış Ekranı Kodları

```

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Data.Common;

```

```
using System.Net.NetworkInformation;
```

```
namespace Market
```

```
{
```

```
    public partial class satis : Form
```

```
    {
```

```
        public satis()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void button10_Click(object sender, EventArgs e)
```

```
        {
```

```
            DialogResult Cikis;
```

```
            Cikis = MessageBox.Show("Menüye Dönülecek Emin misiniz?", "Kapatma Uyarısı!",  
MessageBoxButtons.YesNo);
```

```
            if (Cikis == DialogResult.Yes)
```

```
            {
```

```
                this.Close();
```

```
                menu go = new menu();
```

```
                go.Show();
```

```
            }
```

```
            if (Cikis == DialogResult.No)
```

```
            {
```

```
            }
```

```

    }

    SqlConnection con = new SqlConnection(giris.bagla);

    DataTable dt = new DataTable();

    public static double gtop = 0;

    private void button8_Click(object sender, EventArgs e)
    {
        con.Open();

        SqlCommand kmt4 = new SqlCommand("Select u.URUNid as BarkodNo ,
u.URUNadi as Ürün, u.URUNsatfiyat as SatışFiyatı, m.MARKAadi as Marka From URUN u,
MARKA m where u.MARKAid = m.MARKAid and u.URUNid = @id ", con);

        kmt4.Parameters.AddWithValue("@id", int.Parse(tbbarkod.Text));

        SqlDataReader a2 = kmt4.ExecuteReader();

        while (a2.Read())
        {
            dt.Rows.Add(a2["BarkodNo"], a2["Ürün"], a2["Marka"], int.Parse(tbadet.Text),
double.Parse(tbtotlam.Text));
        }

        con.Close();

        double b = double.Parse(tbtotlam.Text);

        gtop += b;

        label7.Text = "GENEL TOPLAM : " + gtop + "TL";
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        con.Open();

        tburun.Text = "";

        tbfiyat.Text = "";
    }

```



```

tbadet.Text = "";

tbtoplamlam.Text = "";

SqlCommand kmt0 = new SqlCommand("Select URUNadi, URUNsatfiyat From
URUN where URUNid like '" + tbbarkod.Text + "' ", con);

SqlDataReader a = kmt0.ExecuteReader();

while (a.Read())

{

    tburun.Text = a["URUNadi"].ToString();

    tbfiyat.Text = a["URUNsatfiyat"].ToString();

}

con.Close();

}

static bool Kontrol(string deger)

{

    try

    {

        int.Parse(deger);

        return true;

    }

    catch (Exception)

    {

        return false;

    }

}

private void tbadet_TextChanged(object sender, EventArgs e)

{

```

```

if ( (Kontrol(tbadet.Text)))
{
    double x = double.Parse(tbadet.Text) * double.Parse(tbfiyat.Text);
    tbtoplam.Text = x.ToString();
}

}

private void satis_Load(object sender, EventArgs e)
{
    dt.Columns.Add("BarkodNo", typeof(int));
    dt.Columns.Add("Ürün", typeof(string));
    dt.Columns.Add("Marka", typeof(string));
    dt.Columns.Add("Adet", typeof(int));
    dt.Columns.Add("SatışFiyatı", typeof(double));
    dataGridView2.DataSource = dt;
}

private void button1_Click(object sender, EventArgs e)
{
    if (dataGridView2.SelectedRows.Count > 0)
    {
        int b = int.Parse(dataGridView2.SelectedRows[0].Cells[4].Value.ToString());
        dataGridView2.Rows.RemoveAt(dataGridView2.SelectedRows[0].Index);
        gtop -= b;
        label7.Text = "GENEL TOPLAM : " + gtop + " TL";
    }
}

```

```

    }
    else
    {
        MessageBox.Show("Lütfen silinecek satırı seçin.");
    }
}

```

```

private void button2_Click(object sender, EventArgs e)

```

```

{
    tburun.Text = "";
    tbfiyat.Text = "";
    tbadet.Text = "";
    tbtoplam.Text = "";
    tbbarkod.Text = "";
    dt.Clear();
    label7.Text = "GENEL TOPLAM : 0 TL";
}

```

```

int fid;

```

```

double ciro;

```

```

double kar;

```

```

double alfyt;

```

```

double stfyt;

```

```

double urunkar;

```

```

int stok;

```

```

private void btnsat_Click(object sender, EventArgs e)

```

```

{

```

```

DialogResult sat;

sat = MessageBox.Show("Satışı Onaylıyor musunuz ?", "Satış Uyarısı!",
MessageBoxButtons.YesNo);

if (sat == DialogResult.Yes)
{
    con.Open();

    SqlCommand kmt0 = new SqlCommand("Select MAX(FATid) From SAT_URUN
", con);

    fid = Convert.ToInt32(kmt0.ExecuteScalar());

    con.Close();

    for (int i = 0; i < dt.Rows.Count ; i++)
    {
        int urid = Convert.ToInt32(dt.Rows[i]["BarkodNo"]);

        int fadet = Convert.ToInt32(dt.Rows[i]["Adet"]);

        double satfyt = Convert.ToDouble(dt.Rows[i]["SatışFiyatı"]);

        con.Open();

        SqlCommand kmt1 = new SqlCommand("Insert into
SAT_URUN(FATid,URUNid,FATadet,URUNsatfiyatı)
Values(@fatid,@uid,@adet,@fiyat)", con);

        kmt1.Parameters.AddWithValue("@fatid", (fid + 1));

        kmt1.Parameters.AddWithValue("@uid", urid);

        kmt1.Parameters.AddWithValue("@adet", fadet);

        kmt1.Parameters.AddWithValue("@fiyat", satfyt);

        kmt1.ExecuteNonQuery();

        con.Close();

        con.Open();
    }
}

```

```

SqlCommand kmtalfyt = new SqlCommand("Select URUNalfiyat,URUNsatfiyat
From URUN Where URUNid=@brkd",con);

kmtalfyt.Parameters.AddWithValue("@brkd",urid);

SqlDataReader urd = kmtalfyt.ExecuteReader();

while (urd.Read())

{

    stfyt = Convert.ToInt32(urd["URUNsatfiyat"]);

    alfyt = Convert.ToInt32(urd["URUNalfiyat"]);

    urunkar = (stfyt - alfyt) * fadet;

}


con.Close();

con.Open();

SqlCommand kmtstk = new SqlCommand("Select URUNadet From URUN
Where URUNid=@uid", con);

kmtstk.Parameters.AddWithValue("@uid", urid);

SqlDataReader urd1 = kmtstk.ExecuteReader();

while (urd1.Read())

{

    stok = Convert.ToInt32(urd1["URUNadet"]);

    stok -= fadet;

}


con.Close();

con.Open();

SqlCommand kmtstokdus = new SqlCommand("Update URUN Set URUNadet
=@adet Where URUNid =@urid", con);

```

```

        kmtstokdus.Parameters.AddWithValue("@adet",stok);

        kmtstokdus.Parameters.AddWithValue("@urid",urid);

        con.Close();

    }

    int odemeid;

    if (radioButton1.Checked == true)

    {

        odemeid = 1;

    }

    else

    {

        odemeid = 2;

    }

    int kasaaid =0;

    con.Open();

    SqlCommand kmt3 = new SqlCommand("Select KASAid From KASA Where
KASAadi=@kadi", con);

    kmt3.Parameters.AddWithValue("kadi", giris.kasaadi);

    SqlDataReader ka = kmt3.ExecuteReader();

    while (ka.Read())

    {

        kasaaid = Convert.ToInt32(ka["KASAid"]);

    }

    con.Close();

    con.Open();

    SqlCommand kmt2 = new SqlCommand("Insert into
SATIS(SATtoplaml,FATid,Oid,KASAid) values (@stplm,@fid,@oid,@kid)", con);

```

```

kmt2.Parameters.AddWithValue("stplm", gtop);

kmt2.Parameters.AddWithValue("fid", (fid+1));

kmt2.Parameters.AddWithValue("oid", odemeid);

kmt2.Parameters.AddWithValue("kid", kasaid);

kmt2.ExecuteNonQuery();

con.Close();

con.Open();

SqlCommand kmt4 = new SqlCommand("Select CIROtarih From CIRO Where
CIROid=(Select CIROid From CIRO Where CIROtarih=@cirotar)",con);

kmt4.Parameters.AddWithValue("@cirotar",
DateTime.Today.ToShortDateString());

SqlDataReader cdr = kmt4.ExecuteReader();

if (cdr.Read())
{
    con.Close();

    con.Open();

    SqlCommand kmt5 = new SqlCommand("Select CIRO From CIRO Where
CIROid=(Select CIROid From CIRO Where CIROtarih=@cirotar)",con);

    kmt5.Parameters.AddWithValue("@cirotar",
DateTime.Today.ToShortDateString());

    SqlDataReader cda = kmt5.ExecuteReader();

    while (cda.Read())
    {
        ciro = Convert.ToInt32(cda["CIRO"]);

        ciro += gtop;
    }
}

```

```

        con.Close();

        con.Open();

        SqlCommand kmt6 = new SqlCommand("Update CIRO Set CIRO =@ciro
Where CIROid=(Select CIROid From CIRO Where CIROtarih=@cirotar)", con);

        kmt6.Parameters.AddWithValue("@ciro", ciro);

        kmt6.Parameters.AddWithValue("@cirotar",
DateTime.Today.ToShortDateString());

        kmt6.ExecuteNonQuery();

        con.Close();
    }
    else
    {
        con.Close();

        con.Open();

        SqlCommand kmt7 = new SqlCommand("Insert into CIRO(CIRO)
Values(@ciro)", con);

        kmt7.Parameters.AddWithValue("@ciro", gtop);

        kmt7.ExecuteNonQuery();

        con.Close();
    }

    con.Open();

    SqlCommand kmt8 = new SqlCommand("Select KARTarih From KAR Where
KARid=(Select KARid From KAR Where KARTarih=@kartar)", con);

    kmt8.Parameters.AddWithValue("@kartar",
DateTime.Today.ToShortDateString());

    SqlDataReader kdr = kmt8.ExecuteReader();

```



```

        if (kdr.Read())
        {
            con.Close();

            con.Open();

            SqlCommand kmt9 = new SqlCommand("Select KAR From KAR Where
KARid=(Select KARid From KAR Where KARTarih=@kartar)", con);

            kmt9.Parameters.AddWithValue("@kartar",
DateTime.Today.ToShortDateString());

            SqlDataReader kda = kmt9.ExecuteReader();

            while (kda.Read())
            {
                kar = Convert.ToInt32(kda["KAR"]);
            }

            kar += urunkar;

            con.Close();

            con.Open();

            SqlCommand kmt10 = new SqlCommand("Update KAR Set KAR =@kar Where
KARid=(Select KARid From KAR Where KARTarih=@kartar)", con);

            kmt10.Parameters.AddWithValue("@kar", kar);

            kmt10.Parameters.AddWithValue("@kartar",
DateTime.Today.ToShortDateString());

            kmt10.ExecuteNonQuery();

            con.Close();
        }
        else
        {
            con.Close();

```

```

        con.Open();

        SqlCommand kmt11 = new SqlCommand("Insert into KAR(KAR)
Values(@kar)", con);

        kmt11.Parameters.AddWithValue("@kar", urunkar);

        kmt11.ExecuteNonQuery();

        con.Close();

    }

    tbbarkod.Text = "";

    tburun.Text = "";

    tbfiyat.Text = "";

    tbadet.Text = "";

    tbtoplamlam.Text = "";

    dt.Clear();

    label7.Text = "GENEL TOPLAM : 0 TL";

    gtop = 0;

}

if (sat == DialogResult.No)

{

}

}

}

}

```

8.4.Ürün Ekleme Ekranı Kodları

```

using System;

```

```

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Market
{
    public partial class urunekle : Form
    {
        public urunekle()
        {
            InitializeComponent();

            MarkaGetir();
        }

        SqlConnection con = new SqlConnection(giris.bagla);

        DataTable dt = new DataTable();

        private void button10_Click(object sender, EventArgs e)
        {
            this.Close();

            menu go = new menu();

            go.Show();
        }
    }
}

```

```

    }

    private void Temizle()
    {
        tbbarkot.Text = "";
        tburun.Text = "";
        tbadet.Text = "";
        tbfiyat.Text = "";
        tbsatfiyat.Text = "";
        cbmarka.Text = "";
    }

    private void MarkaGetir()
    {
        con.Open();

        SqlCommand kmt = new SqlCommand("Select * From MARKA Order By
MARKAid", con);

        SqlDataReader dr = kmt.ExecuteReader();

        while (dr.Read())
        {
            cbmarka.Items.Add(dr["MARKAadi"].ToString());
        }

        con.Close();
    }

    public static int y;

    private void button8_Click(object sender, EventArgs e)
    {
        if (tbbarkot.Text!="" && tbadet.Text!="" && tburun.Text!="" && tbfiyat.Text!=""
&& tbsatfiyat.Text!="" && cbmarka.Text!="")

```

```

{
    con.Open();

    SqlCommand kmt3 = new SqlCommand("Select * FROM URUN Where URUNId
= @uid", con);

    kmt3.Parameters.AddWithValue("@uid", int.Parse(tbbarkot.Text));

    SqlDataReader d = kmt3.ExecuteReader();

    Boolean c = d.Read();

    con.Close();

    if (c == true)
    {
        MessageBox.Show("Bu Barkot Numarasına Sahip Bir Ürün Zaten Var!");
    }
    else
    {

        con.Open();

        SqlCommand kmt0 = new SqlCommand("Select * FROM MARKA Where
MARKAadi = @madi", con);

        kmt0.Parameters.AddWithValue("@madi", cbmarka.Text);

        SqlDataReader a = kmt0.ExecuteReader();

        Boolean b = a.Read();

        con.Close();

        if (b == true)
        {

        }
    }
}

```

```

else
{
    con.Open();

    SqlCommand mekle = new SqlCommand("Insert into MARKA(MARKAadi)
values(@m) ", con);

    mekle.Parameters.AddWithValue("@m", cbmarka.Text);

    mekle.ExecuteNonQuery();

    con.Close();
}

con.Open();

SqlCommand kmt2 = new SqlCommand("Select MARKAid FROM MARKA
Where MARKAadi = @mad", con);

kmt2.Parameters.AddWithValue("@mad", cbmarka.Text);

SqlDataReader a1 = kmt2.ExecuteReader();

while (a1.Read())
{
    y = Convert.ToInt32(a1["MARKAid"]);
}

a1.Close();

con.Close();

con.Open();

SqlCommand kmt1 = new SqlCommand("Insert into
URUN(URUNid,URUNadi,URUNadet,URUNalfiyat,URUNsatfiyat,MARKAid)
values(@id,@adi,@adet,@alfyt,@satfyt,@mid)", con);

kmt1.Parameters.AddWithValue("@id", int.Parse(tbbarkot.Text));

kmt1.Parameters.AddWithValue("@adi", tburun.Text);

```

```

kmt1.Parameters.AddWithValue("@adet", int.Parse(tbadet.Text));

kmt1.Parameters.AddWithValue("@alfyt", double.Parse(tbfiyat.Text));

kmt1.Parameters.AddWithValue("@satfyt", double.Parse(tbsatfiyat.Text));

kmt1.Parameters.AddWithValue("@mid", y);

kmt1.ExecuteNonQuery();

con.Close();

con.Open();

SqlCommand kmt4 = new SqlCommand("Select u.URUNid as BarkotNo ,
u.URUNadi as Ürün, u.URUNadet as Adet, u.URUNsatfiyat as SatışFiyatı, m.MARKAadi as
Marka From URUN u, MARKA m where u.MARKAid = m.MARKAid and u.URUNid =
@id ", con);

kmt4.Parameters.AddWithValue("@id", int.Parse(tbbarkot.Text));

SqlDataReader a2 = kmt4.ExecuteReader();

while (a2.Read())
{
    dt.Rows.Add(a2["BarkotNo"], a2["Ürün"], a2["Adet"], a2["SatışFiyatı"],
a2["Marka"]);
}

con.Close();

label8.Text = "Ürün Eklendi";

Temizle();
}
}
else
{
    MessageBox.Show("Alanlar Boş Bırakılamaz!");
}

```

```

    }

    private void urunekle_Load(object sender, EventArgs e)
    {
        dt.Columns.Add("BarkotNo", typeof(int));
        dt.Columns.Add("Ürün", typeof(string));
        dt.Columns.Add("Adet", typeof(int));
        dt.Columns.Add("SatışFiyatı", typeof(double));
        dt.Columns.Add("Marka", typeof(string));
        dataGridView2.DataSource = dt;
    }
}
}

```

8.5.Ürün Bilgilerini Güncelleme Ekranı Kodları

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

```



```

namespace Market
{
    public partial class urunguncelle : Form
    {
        public urunguncelle()
        {
            InitializeComponent();
        }

        private void Temizle()
        {
            tburun.Text = "";
            tbadet.Text = "";
            tbfiyat.Text = "";
            tbsatfiyat.Text = "";
            cbmarka.Text = "";
        }

        private void button10_Click(object sender, EventArgs e)
        {
            this.Close();

            menu go = new menu();
            go.Show();
        }

        SqlConnection con = new SqlConnection(giris.bagla);
        DataTable dt = new DataTable();

        private void tbbarkot_TextChanged(object sender, EventArgs e)
        {

```

```

        Temizle();

        con.Open();

        SqlCommand kmt0 = new SqlCommand("Select u.URUNadi, u.URUNadet,
u.URUNalfiyat, u.URUNsatfiyat, m.MARKAadi From URUN u, Marka m where
u.MARKAid=m.MARKAid and URUNid like '"+tbbarkot.Text +"' ",con);

        SqlDataReader a = kmt0.ExecuteReader();

        while (a.Read())
        {
            tburun.Text = a["URUNadi"].ToString();
            tbadet.Text = a["URUNadet"].ToString();
            tbfiyat.Text = a["URUNalfiyat"].ToString();
            tbsatfiyat.Text = a["URUNsatfiyat"].ToString();
            cbmarka.Text = a["MARKAadi"].ToString();
        }

        con.Close();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        con.Open();

        SqlCommand kmt1 = new SqlCommand("Update URUN Set URUNadi = @adi,
URUNadet = @adet, URUNalfiyat = @alfyt, URUNsatfiyat = @satfyt, MARKAid = (Select
MARKAid From MARKA Where MARKAadi = @mid) Where URUNid = @uid",con);

        kmt1.Parameters.AddWithValue("@uid",int.Parse(tbbarkot.Text));

        kmt1.Parameters.AddWithValue("@adi",tburun.Text);

        kmt1.Parameters.AddWithValue("@adet",int.Parse(tbadet.Text));

        kmt1.Parameters.AddWithValue("@alfyt",double.Parse(tbfiyat.Text));

```

```

kmt1.Parameters.AddWithValue("@satfyt",double.Parse(tbsatfiyat.Text));

kmt1.Parameters.AddWithValue("@mid",cbmarka.Text);

kmt1.ExecuteNonQuery();

con.Close();

con.Open();

SqlCommand kmt4 = new SqlCommand("Select u.URUNid as BarkotNo ,
u.URUNadi as Ürün, u.URUNadet as Adet, u.URUNalfiyat as AlışFiyatı, u.URUNsatfiyat as
SatışFiyatı, m.MARKAadi as Marka From URUN u, MARKA m where u.MARKAid =
m.MARKAid and u.URUNid = @id ", con);

kmt4.Parameters.AddWithValue("@id", int.Parse(tbbarkot.Text));

SqlDataReader a2 = kmt4.ExecuteReader();

while (a2.Read())

{

    dt.Rows.Add(a2["BarkotNo"],    a2["Ürün"],    a2["Adet"],    a2["AlışFiyatı"],
a2["SatışFiyatı"], a2["Marka"]);

}

con.Close();

Temizle();

tbbarkot.Text = "";

}

private void urunguncelle_Load(object sender, EventArgs e)

{

    dt.Columns.Add("BarkotNo", typeof(int));

    dt.Columns.Add("Ürün", typeof(string));

    dt.Columns.Add("Adet", typeof(int));

    dt.Columns.Add("AlışFiyatı", typeof(double));

```

```

        dt.Columns.Add("SatışFiyatı", typeof(double));

        dt.Columns.Add("Marka", typeof(string));

        dataGridView2.DataSource = dt;
    }
}
}

```

8.6.Toplu Stok Ekranı Kodları

```

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Market
{
    public partial class stokekle : Form
    {
        public stokekle()
        {
            InitializeComponent();
        }
    }
}

```

```

private void button10_Click(object sender, EventArgs e)
{
    this.Close();

    menu go = new menu();

    go.Show();
}

SqlConnection con = new SqlConnection(giris.bagla);
DataTable dt = new DataTable();

private void tbbarkot_TextChanged(object sender, EventArgs e)
{
    label3.Text = "";

    con.Open();

    SqlCommand kmt0 = new SqlCommand("Select URUNadi From URUN where
    URUNId like '" + tbbarkod.Text + "' ", con);

    SqlDataReader a = kmt0.ExecuteReader();

    while (a.Read())
    {
        label3.Text = a["URUNadi"].ToString();

    }

    con.Close();
}

public static int tpx;

private void button8_Click(object sender, EventArgs e)
{

```

```

con.Open();

SqlCommand kmt1 = new SqlCommand("Select URUNadet From URUN where
URUNId like '" + tbbarkod.Text + "'", con);

SqlDataReader a1 = kmt1.ExecuteReader();

while (a1.Read())
{
    tpx = Convert.ToInt32(a1["URUNadet"]);
}

con.Close();

tpx += int.Parse(tbadet.Text);

con.Open();

SqlCommand kmt2 = new SqlCommand("Update URUN Set URUNadet = @adet
Where URUNId = @uid", con);

kmt2.Parameters.AddWithValue("@uid", int.Parse(tbbarkod.Text));

kmt2.Parameters.AddWithValue("@adet", tpx);

kmt2.ExecuteNonQuery();

con.Close();

con.Open();

SqlCommand kmt4 = new SqlCommand("Select u.URUNId as BarkodNo ,
u.URUNadi as Ürün, m.MARKAadi as Marka, u.URUNadet as Adet From URUN u,
MARKA m where u.MARKAid = m.MARKAid and u.URUNId = @uid ", con);

kmt4.Parameters.AddWithValue("@uid", int.Parse(tbbarkod.Text));

SqlDataReader a2 = kmt4.ExecuteReader();

while (a2.Read())
{
    dt.Rows.Add(a2["BarkodNo"], a2["Ürün"], a2["Marka"], a2["Adet"]);
}

```

```

    }

    con.Close();

    tbbarkod.Text = "";

    tbadet.Text = "";

    label3.Text = "";

}

private void stokekle_Load(object sender, EventArgs e)
{
    dt.Columns.Add("BarkodNo", typeof(int));

    dt.Columns.Add("Ürün", typeof(string));

    dt.Columns.Add("Marka", typeof(string));

    dt.Columns.Add("Adet", typeof(int));

    dataGridView2.DataSource = dt;

}

}

}

```

8.7.Satış Geçmişi Ekranı Kodları

```

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Security.Cryptography;

namespace Market
{
    public partial class satisgecmis : Form
    {
        public satisgecmis()
        {
            InitializeComponent();
        }

        private void button10_Click(object sender, EventArgs e)
        {
            this.Close();

            menu go = new menu();

            go.Show();

        }

        SqlConnection con = new SqlConnection(giris.bagla);

        DataTable dt = new DataTable();

        private void button8_Click(object sender, EventArgs e)
        {
            if (tbbarkod.Text == "")
            {

```



```

    }

    else

    {

        con.Open();

        SqlCommand kmt4 = new SqlCommand("Select s.SATtarih as SatışTarihi,
s.SATtoplam as SatışToplam, o.Oadi as ÖdemeTipi, k.KASAadi as KasaAdı From SATIS s,
ODEME o, KASA k where o.Oid = s.Oid and s.KASAid=k.KASAid and SATid =@stid",
con);

        kmt4.Parameters.AddWithValue("@stid", int.Parse(tbbarkod.Text));

        SqlDataAdapter a1 = new SqlDataAdapter();

        a1.SelectCommand = kmt4;

        a1.Fill(dt);

        con.Close();

        dataGridView2.DataSource = dt;

    }

}

private void button1_Click(object sender, EventArgs e)

{

    string zmn = dateTimePicker1.Value.ToString("MMM dd yyyy");

    con.Open();

    SqlCommand kmt0 = new SqlCommand("Select s.SATtarih as SatışTarihi,
s.SATtoplam as SatışToplam, o.Oadi as ÖdemeTipi, k.KASAadi as KasaAdı From SATIS s,
ODEME o, KASA k where o.Oid = s.Oid and s.KASAid=k.KASAid and SATtarih like '%" +
zmn + "%'", con);

    SqlDataAdapter a1 =new SqlDataAdapter();

```

```

        a1.SelectCommand = kmt0;

        a1.Fill(dt);

        con.Close();

        dataGridView2.DataSource = dt;
    }

    private void satisgecmis_Load(object sender, EventArgs e)
    {
        dt.Columns.Add("Satış Tarihi", typeof(DateTime));
        dt.Columns.Add("Satış Toplam", typeof(double));
        dt.Columns.Add("Ödeme Tipi", typeof(string));
        dt.Columns.Add("Kasa Adı", typeof(string));
        dataGridView2.DataSource = dt;
    }
}

```

8.8.Kâr ve Ciro Rapor Ekranı Kodları

```

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```
using System.Windows.Forms;
```

```
namespace Market
```

```
{
```

```
    public partial class rapor : Form
```

```
    {
```

```
        public rapor()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        SqlConnection con = new SqlConnection(giris.bagla);
```

```
        DataTable dt = new DataTable();
```

```
        private void button1_Click(object sender, EventArgs e)
```

```
        {
```

```
            dt.Clear();
```

```
            string zmn = dateTimePicker1.Value.ToString("MMM dd yyyy");
```

```
            con.Open();
```

```
            SqlCommand kmt0 = new SqlCommand("Select CIRO as Rapor,CIROtarih as Tarih  
From CIRO where CIROid =(Select CIROid From CIRO where CIROtarih like  
'%" + zmn + "%' )", con);
```

```
            SqlDataAdapter a1 = new SqlDataAdapter();
```

```
            a1.SelectCommand = kmt0;
```

```
            a1.Fill(dt);
```

```
            con.Close();
```

```
            dataGridView2.DataSource = dt;
```

```
        }
```

```

private void button10_Click(object sender, EventArgs e)
{
    this.Close();

    menu go = new menu();

    go.Show();
}

private void button2_Click(object sender, EventArgs e)
{
    dt.Clear();

    string zmn = dateTimePicker1.Value.ToString("MMM dd yyyy");

    con.Open();

    SqlCommand kmt0 = new SqlCommand("Select KAR as Rapor,KARtarih as Tarih
From KAR where KARid =(Select KARid From KAR where KARtarih like '%" + zmn + "%' )",
con);

    SqlDataAdapter a1 = new SqlDataAdapter();

    a1.SelectCommand = kmt0;

    a1.Fill(dt);

    con.Close();

    dataGridView2.DataSource = dt;
}

private void rapor_Load(object sender, EventArgs e)
{
}

}

}

```

8.9.Stok Bilgileri Ekranı Kodları

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Market

{

    public partial class stokbilgi : Form

    {

        public stokbilgi()

        {

            InitializeComponent();

        }

        private void button10_Click(object sender, EventArgs e)

        {

            this.Close();

            menu go = new menu();

            go.Show();

        }

    }

}
```

```

    }

    SqlConnection con = new SqlConnection(giris.bagla);

    DataTable dt = new DataTable();

    private void button3_Click(object sender, EventArgs e)
    {
        if (tbbarkot.Text == "")
        {

        }

        else
        {
            dt.Clear();

            con.Open();

            SqlCommand kmt0 = new SqlCommand("Select u.URUNadi as
            ÜrünAdı,m.MARKAadi as Marka, u.URUNadet as ÜrünAdeti From URUN u, MARKA m
            Where u.MARKAid=m.MARKAid and u.URUNid=@uid ", con);

            kmt0.Parameters.AddWithValue("@uid", int.Parse(tbbarkot.Text));

            SqlDataAdapter a1 = new SqlDataAdapter();

            a1.SelectCommand = kmt0;

            a1.Fill(dt);

            con.Close();

            dataGridView2.DataSource = dt;

        }
    }

    private void button1_Click(object sender, EventArgs e)
    {

```

```

dt.Clear();

con.Open();

SqlCommand kmt1 = new SqlCommand("Select u.URUNadi as
ÜrünAdı,m.MARKAadi as Marka, u.URUNadet as ÜrünAdeti From URUN u, MARKA m
Where u.MARKAid=m.MARKAid and u.URUNadet <= 10 ", con);

SqlDataAdapter a1 = new SqlDataAdapter();

a1.SelectCommand = kmt1;

a1.Fill(dt);

con.Close();

dataGridView2.DataSource = dt;

}

}

}

```

KAYNAKÇA

<https://stackoverflow.com>

<https://github.com>

<https://www.reddit.com/r/programming/>