

T.C.  
KIRIKKALE ÜNİVERSİTESİ  
MESLEK YÜKSEKOKULU  
BİLGİSAYAR PROGRAMCILIĞI

SİSTEM ANALİZİ ve TASARIMI DERSİ  
ARAÇ KİRALAMA OTOMASYONU PROGRAMI PROJESİ

Dersin Öğretim Elemanı  
Doç. Dr. Aşır ÖZBEK

ZUHURİ ALTUN  
190705003

KIRIKKALE-2021

## İçindekiler

<b>1. Giriş</b> .....	1
<b>2. VERİ TABANI TASARIMI</b> .....	2
<b>2.1. Gerekliklik Analizi</b> .....	2
<b>2.2. Kavramsal Tasarım</b> .....	2
2.2.1. İller Varlık Kümesi .....	3
2.2.2. Müşteriler Varlık Kümesi .....	3
2.2.3. Sigortalar Varlık Kümesi .....	4
2.2.4 Marka Varlık Kümesi.....	5
2.2.5 Model Varlık Kümesi.....	5
2.2.6 Yakıttip Varlık Kümesi .....	5
2.2.7 Vites Varlık Kümesi.....	6
2.2.8 Cekis Varlık Kümesi.....	6
2.2.9 Renk Varlık Kümesi .....	6
2.2.10. Araçlar Varlık Kümesi .....	7
<b>2.3. İlişkilendirmeler</b> .....	8
2.3.1. Araçlar, Müşteri ilişkisi .....	8
2.3.2. Araçlar, Modeller İlişkisi .....	8
2.3.3. Araçlar, Yakıttip İlişkisi.....	9
2.3.4. Araçlar, Vites İlişkisi .....	9
2.3.5. Araçlar, Cekis İlişkisi .....	10
2.3.6. Araçlar, Renk İlişkisi .....	10
2.3.7. Markalar, Modeller İlişkisi .....	11
2.3.8. İller, Araçlar İlişkisi.....	11
2.3.9. Genel Gösterim.....	12
<b>2.4. Mantıksal Tasarım</b> .....	13
<b>2.5. Fiziksel Tasarım</b> .....	17
2.5.1. musteriler Tablosu .....	17
2.5.2. markalar Tablosu: .....	18
2.5.3. modeller Tablosu: .....	19
2.5.4. yakıttip Tablosu: .....	19
2.5.5. cekis Tablosu: .....	19
2.5.6. renk Tablosu: .....	20
2.5.7. vites Tablosu:.....	20
2.5.8. araçlar Tablosu .....	20
2.5.9. iller Tablosu.....	21
2.5.10. sigortalarTablosu .....	21

2.5.11. kiralama Tablosu .....	22
2.5.12. Veri tabanı Diyagramı .....	23
2.5.13. Tabloların SQL Kodları .....	24
2.6. Projenin Akış Şeması .....	33
2.6.1. Ana Sayfa Akış Şeması .....	33
2.6.2. Veri Al Metodu Akış Şeması .....	34
2.6.3. Veri Ekle Metodu Akış Şeması .....	35
2.6.4. Veri Sil Metodu Akış Şeması.....	36
2.6.5. Kayıt Güncelle Metodu Akış Şeması .....	37
2.6.6. Sorgu Gönder Metodu Akış Şeması .....	38
2.7. Projenin Uygulama Kısım .....	39
2.7.1. Ana Sayfa Ekranı.....	39
2.7.2. Araç İşlemleri Ekranı .....	43
2.7.3. Müşteri İşlemleri Ekranı .....	46
2.7.4. Sigorta İşlemleri Ekranı .....	48
2.7.5. İstatistik Ekranı .....	49
2.8. Projenin Kodları.....	52
2.8.1. Başlangıç Formu Kodları .....	52
2.8.2. Ana Sayfa UserControl Kodları .....	54
2.8.3. HomePage UserControl Kodları.....	56
2.8.4. AraçKiralama UserControl Kodları.....	57
2.8.5. Araç İşlemleri UserControl Kodları.....	60
2.8.6. Yeni Araç Ekle UserControl Kodları.....	62
2.8.7. Araç Bilgisi Güncelle UserControl Kodları .....	65
2.8.8. Araç Silme UserControl Kodları .....	68
2.8.9. Marka-Model Ekle UserControl Kodları .....	69
2.8.10. Müşteri İşlemleri UserControl Kodları.....	71
2.8.11. Müşteri Kayıt UserControl Kodları.....	73
2.8.12. Müşteri Güncelle UserControl Kodları.....	75
2.8.13. Müşteri Sil UserControl Kodları .....	78
2.8.14. Sigorta İşlemleri UserControl Kodları.....	79
2.8.15. İstatistik UserControl Kodları.....	81
2.8.16. Araç İstatistikleri UserControl Kodları.....	82
2.8.17. Müşteri İstatistikleri UserControl Kodları.....	84
2.8.18. VtClass Kodları.....	85

## **TABLO LİSTESİ**

Tablo 1: Müşteriler Tablosu.....	13
Tablo 2: İller Tablosu.....	13
Tablo 3: Sigortalar Tablosu.....	14
Tablo 4: Markalar Tablosu.....	14
Tablo 5: Modeller Tablosu.....	14
Tablo 6: Araçlar Tablosu.....	15
Tablo 7: Kiralama Tablosu.....	15
Tablo 8: Yakıttip Tablosu.....	16
Tablo 9: Vites Tablosu.....	16
Tablo 10: Cekis Tablosu.....	16
Tablo 11: Renk Tablosu.....	17

## ŞEKİLLER LİSTESİ

Şekil 1: İller varlık kümesi.....	3
Şekil 2: Müşteriler varlık kümesi.....	4
Şekil 3: Sigortalar varlık kümesi.....	4
Şekil 4: Marka varlık kümesi.....	5
Şekil 5: Model varlık kümesi.....	5
Şekil 6: Yakıttip varlık kümesi.....	5
Şekil 7: Vites varlık kümesi.....	6
Şekil 8: Cekis varlık kümesi.....	6
Şekil 9: Renk varlık kümesi.....	6
Şekil 10: Araçlar varlık kümesi.....	7
Şekil 11: Araçlar Müşteri İlişkisi.....	8
Şekil 12: Araçlar, Modeller İlişkisi.....	8
Şekil 13: Araçlar, Yakıt İlişkisi.....	9
Şekil 14: Araçlar, Vites İlişkisi.....	9
Şekil 15: Araçlar, Cekis İlişkisi.....	10
Şekil 16: Araçlar, Renk İlişkisi.....	10
Şekil 17: Markalar, Modeller İlişkisi.....	11
Şekil 18: Araçlar, İl İlişkisi.....	11
Şekil 19: Genel Gösterim.....	12
Şekil 20: müşteriler Tablosu.....	18
Şekil 21: markalar Tablosu.....	18
Şekil 22: modeller Tablosu.....	19
Şekil 23: yakıttip Tablosu.....	19
Şekil 24: cekis Tablosu.....	19
Şekil 25: renk Tablosu.....	20
Şekil 26: vites Tablosu.....	20
Şekil 27: araçlar Tablosu.....	21
Şekil 28: iller Tablosu.....	21
Şekil 29: sigortalar Tablosu.....	22
Şekil 30: kiralama Tablosu.....	22
Şekil 31: Veri tabanı Diyagramı.....	23

## 1. Giriş

Araç Kiralama otomasyonu projesinde firmanın elinde bulunan araçların her daim takip edilebilir olması, anlık olarak kiralanan araçların görüntülenebilmesi, müşterilerin daha kolay ve hızlı bir şekilde araç kiralayabilmesi, kiralanan araçların istatistiklerinin tutulması ve bu istatistiklere göre yeni araç alımlarının daha kolay yapılması ve elbette müşterilere daha iyi bir hizmet verebilmek amaçlanmıştır. Öncelikle araçların ve müşterilerin gerekli olan her türlü bilgilerinin saklanacağı veri tabanı tasarımı ve uygulaması yapılmıştır. Araç Kiralama otomasyonu genel olarak; iller, müşteriler, araçlar, araç markaları, araç modelleri, araç kasa tipleri, araç segmentleri, sigorta bilgileri ve elbette kiralanan araçların kaydı tutulacaktır. Şubeye gelen müşterinin eğer daha önceden oluşturulmuş bir kaydı yoksa müşterinin adı, soyadı, TC kimlik numarası ve telefon numarası gibi bilgileri alınarak müşterinin kaydı oluşturulacaktır. Müşterinin istekleri doğrultusunda istenen marka model vb. bilgilere uygun araçlar müşteriye sunulacak ve müşterinin kiralamak istediği aracı hızlı bir şekilde seçebilmesi sağlanmış olacaktır. Ardından istenen kiralama günü ve sigorta bilgileri girildikten sonra işlemler tamamlanarak araç müşteriye teslim edilecektir. Bu projede giriş kısmından sonra ikinci kısım olarak veri tabanı tasarlanmıştır. İlk önce gereklilik analizi yapılarak otomasyonda ihtiyaç duyulan durumlar belirlenmiştir. Daha sonra üçüncü kısım olarak bu durumlara uygun olarak ihtiyaçların giderilmesi amacıyla kavramsal tasarım yapılmıştır. Kavramsal tasarımda veri tabanında kullanılacak varlık kümeleri belirlenip bu kümelere ait nitelikler ve diğer kümelerle aralarındaki ilişkiler tasarlanmıştır. Dördüncü kısımda mantıksal tasarım yapılmıştır. Burada kavramsal tasarımda belirlenen varlıklar arası ilişkiler açıklanarak varlıklar tablosu tasarlanmıştır. Beşinci kısım fiziksel tasarım yani uygulama alanında projenin veritabanı MSSQL Server programı ile tasarlanmıştır. Bu kısımda projedeki tablolara ait bilgiler listelenmiştir. Son olarak altıncı kısımda projeye ait SQL kodları yazılmıştır. Böylece proje 6 temel kısımda tamamlanmıştır.

## **2. VERİ TABANI TASARIMI**

### **2.1. Gereklilik Analizi**

Projede en önemli şey verilerin doğru ve düzenli bir şekilde kayıt altında tutulmasıdır. Firmanın elinde olan araçların markalarının modellerinin her araca ait diğer bilgilerin ve müşterilere ait bilgilerin tutulması gerekir.

Programda ihtiyaç duyulan işleyiş durumu yapılan araştırma sonucu aşağıdaki belirtildiği şekilde tespit edilmiştir.

- Hizmet Sunulan İllerin kayıt altına alınması,
- Müşteri bilgilerinin kayıt altına alınması,
- Kiralık Araç markalarının kayıt altına alınması,
- Kiralık Araç modellerinin kayıt altına alınması,
- Kiralık Araç bilgilerinin kayıt altına alınması,
- Araç kiralama işlemlerinin kayıt altına alınması.

### **2.2. Kavramsal Tasarım**

Kavramsal tasarım otomasyonun işleyişine uygun olarak aşağıdaki ihtiyaçların giderilmesi amacıyla tasarlanmıştır.

- Kiralanan araçların takibinin yapılması,
- Müşteri bilgilerinin hızlı bir şekilde kayıt altına alınabilmesi,
- Müşterilerin araç seçim işlemlerini hızlı bir şekilde yapabilmesinin sağlanması,
- Araçların kolay bir şekilde sisteme eklenip sistemden çıkartılabilmesi,
- Araçların sistemdeki bilgilerinin kolayca güncellenebilmesi,
- Kiralama işlemlerinin kayıt altına alınması ile istatistiklerinin oluşturulması.

İhtiyaç duyulan otomasyonun işleyişi için gereklilik analizine de bakılarak kavramsal tasarım planlaması ve varlıklar oluşturulmuştur. Bu plana göre varlık kümeleri aşağıda belirtildiği gibi oluşturulmuştur.

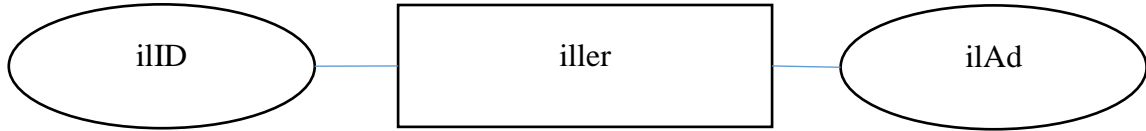
- İller Varlık Kümesi,
- Müşteriler Varlık Kümesi,
- Sigorta Varlık Kümesi,
- Marka Varlık Kümesi,
- Model Varlık Kümesi,
- Yakıttip Varlık Kümesi,
- Vites Varlık Kümesi,
- Cekis Varlık Kümesi,
- Renk Varlık Kümesi,
- Araçlar Varlık Kümesi.

### 2.2.1. İller Varlık Kümesi

İller Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- ilID
- ilAd

Bu özelliklere göre iller varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda ilID anahtar nitelik olarak tanımlanmıştır.



**Şekil 1:** İller varlık kümesi

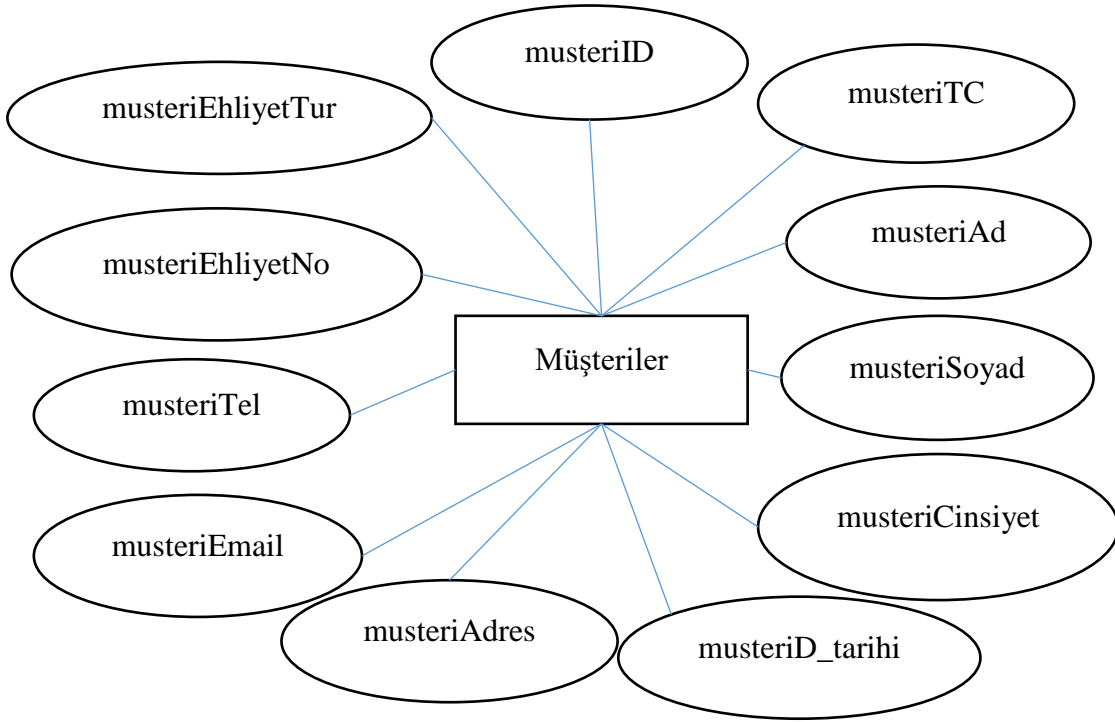
### 2.2.2. Müşteriler Varlık Kümesi

Müşteriler Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- müşteriID
- müşteriTC
- müşteriAd
- müşteriSoyad
- müşteriCinsiyet
- müşteriDogumTarihi
- müşteriAdres
- müşteriEmail
- müşteriTel
- müşteriEhliyetNo
- müşteriEhliyetTur

Bu özelliklere göre müşteriler varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda müşteriID anahtar nitelik olarak tanımlanmıştır.





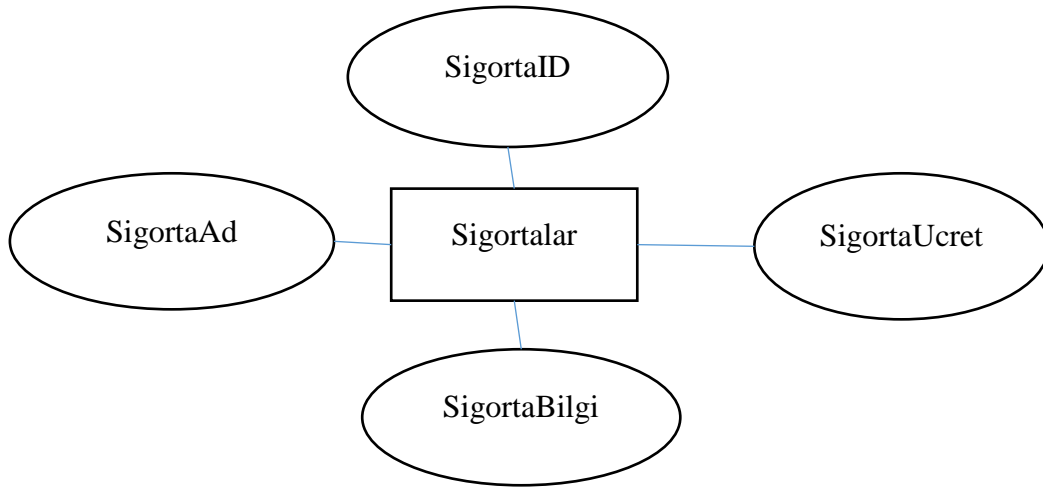
**Şekil 2:** Müşteriler varlık kümesi

### 2.2.3. Sigortalar Varlık Kümesi

Sigortalar Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- SigortaID
- SigortaAd
- SigortaUcret
- SigortaBilgi

Bu özelliklere göre sigortalar varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda SigortaID anahtar nitelik olarak tanımlanmıştır.



**Şekil 3:** Sigortalar varlık kümesi

#### 2.2.4 Marka Varlık Kümesi

Marka Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- markaID
- markaAd

Bu özelliklere göre marka varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda markaID anahtar nitelik olarak tanımlanmıştır.



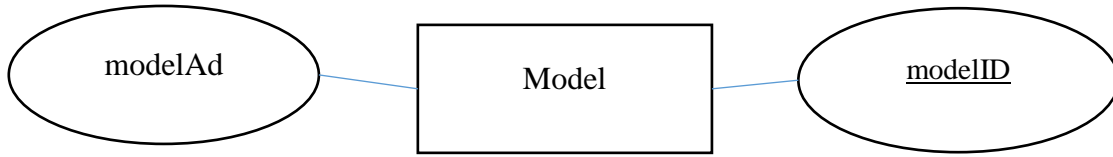
Şekil 4: Marka varlık kümesi

#### 2.2.5 Model Varlık Kümesi

Model Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- modelID
- modelAd

Bu özelliklere göre Model varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda modelID anahtar nitelik olarak tanımlanmıştır.



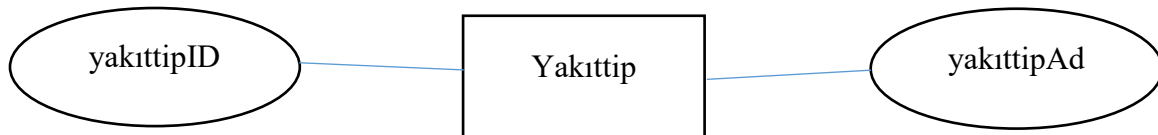
Şekil 5: Model varlık kümesi

#### 2.2.6 Yakıttip Varlık Kümesi

Yakıttip Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- yakıttipID
- yakıttipAd

Bu özelliklere göre Yakıttip varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda yakıttipID anahtar nitelik olarak tanımlanmıştır.



Şekil 6: Yakıttip varlık kümesi

### 2.2.7 Vites Varlık Kümesi

Vites Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- vitesID
- vitesAd

Bu özelliklere göre Vites varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda vitesID anahtar nitelik olarak tanımlanmıştır.



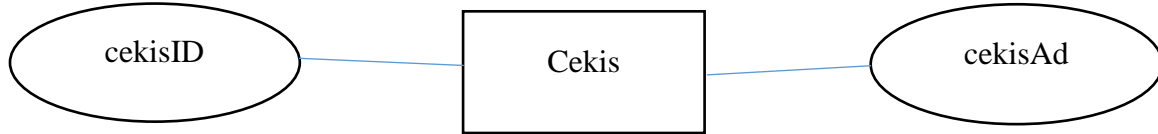
Şekil 7: Vites varlık kümesi

### 2.2.8 Cekis Varlık Kümesi

Cekis Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- cekisID
- cekisAd

Bu özelliklere göre Cekis varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda cekisID anahtar nitelik olarak tanımlanmıştır.



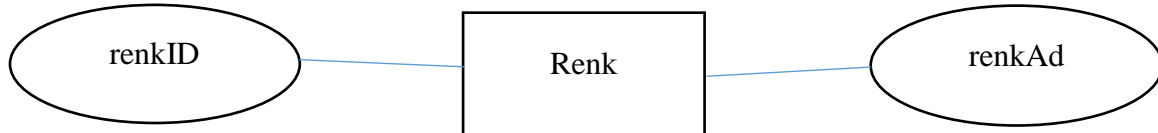
Şekil 8: Cekis varlık kümesi

### 2.2.9 Renk Varlık Kümesi

Renk Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- renkID
- renkAd

Bu özelliklere göre Renk varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda renkID anahtar nitelik olarak tanımlanmıştır.



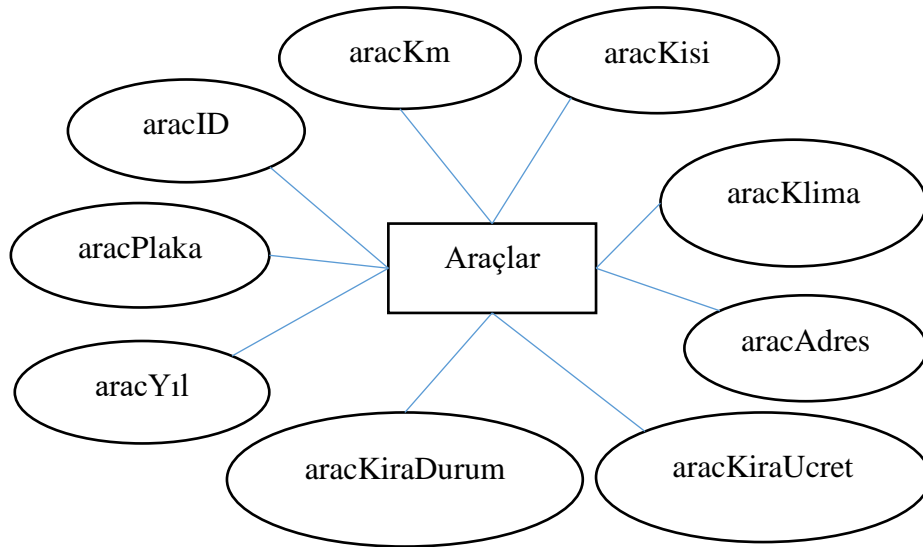
Şekil 9: Renk varlık kümesi

### 2.2.10. Araçlar Varlık Kümesi

Araçlar Varlık Kümesi tasarlanırken bu kümedeki varlıklara ait özellikler aşağıdaki gibi oluşturulmuştur.

- aracID
- aracPlaka
- aracYıl
- aracKm
- aracKisi
- aracKlima
- aracAdres
- aracKiraUcret
- aracKiraDurum

Bu özelliklere göre Arac varlık kümesi aşağıdaki şekle göre tasarlanmıştır. Bu tasarımda aracID anahtar nitelik olarak tanımlanmıştır.

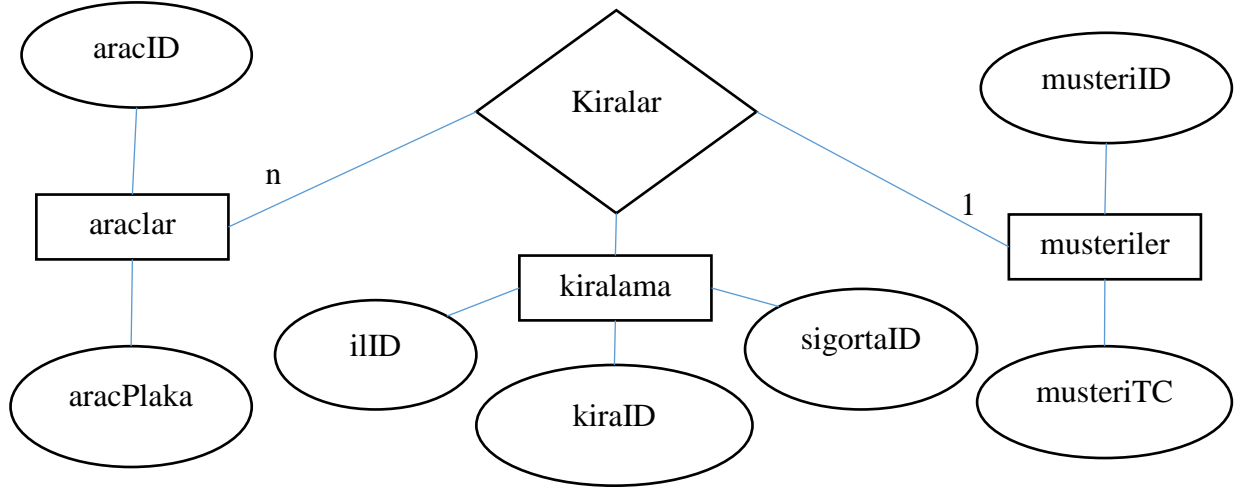


**Şekil 10:** Araçlar varlık kümesi

## 2.3. İlişkilendirmeler

### 2.3.1. Araçlar, Müşteri ilişkisi

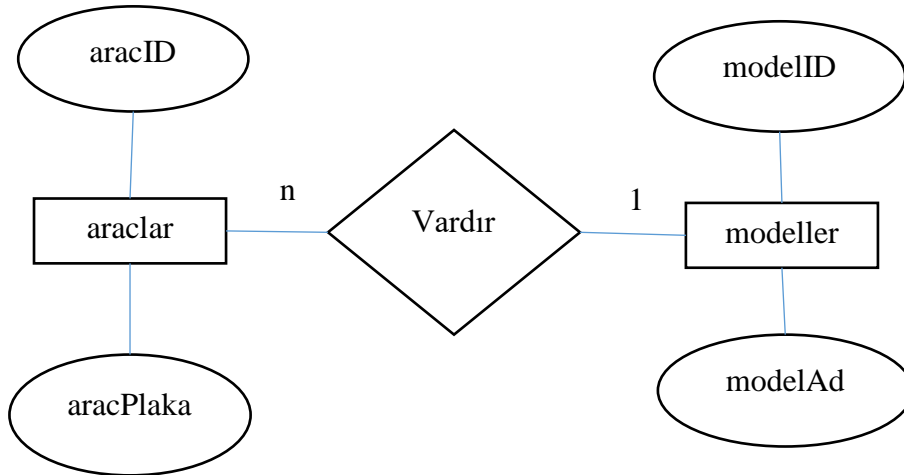
Bir araç aynı anda birden fazla kişiye kiralanamaz ama bir müşteri birden fazla araç kiralayabilir. Bu sebeple müşteriler ve araçlar varlık kümeleri arasında 1:n ilişki vardır. Ve de bu ilişkiden kiralama tablosu çıkmaktadır. Musteriler\_Araclar varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 11’de gösterilmiştir.



Şekil 11: Araçlar Müşteri İlişkisi

### 2.3.2. Araçlar, Modeller İlişkisi

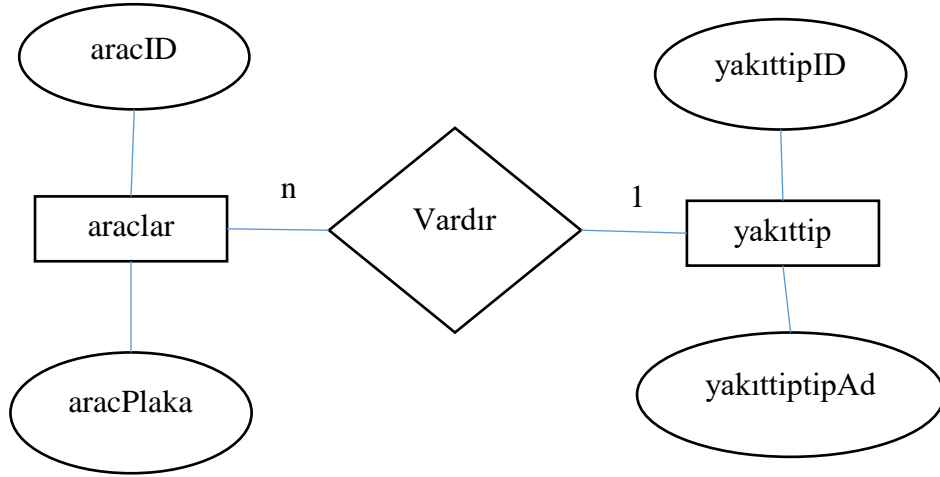
Bir modelde birçok araç olabilir ama bir aracın yalnızca bir modeli olur. Araçlar ve modeller varlık kümeleri arasındaki ilişki 1:n ilişkidir. Araclar\_modeller varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 12’de gösterilmiştir.



Şekil 12: Araçlar, Modeller İlişkisi

### 2.3.3. Araçlar, Yakıttip İlişkisi

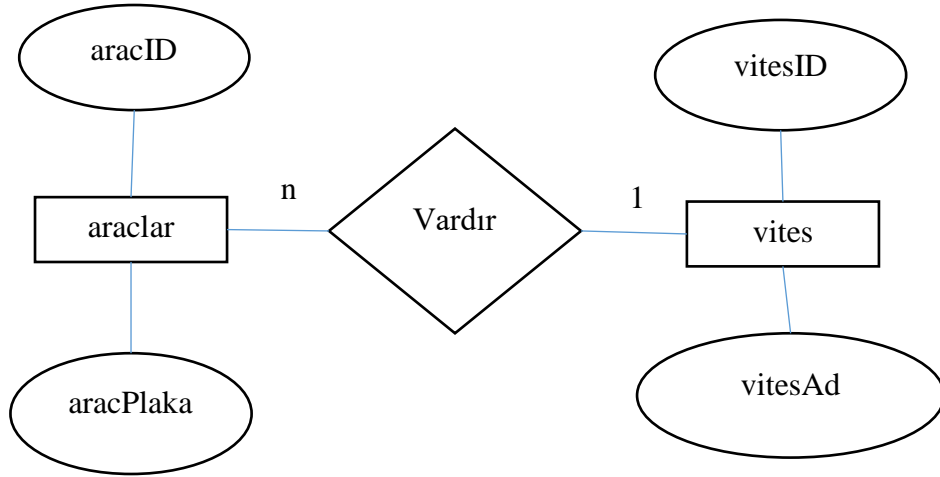
Bir yakıt tipinde birçok araç olabilir ama bir aracın yalnızca bir yakıttipi olur. Araçlar ve yakıttip varlık kümeleri arasındaki ilişki 1:n ilişkidir. Araclar\_yakıttip varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 13’de gösterilmiştir.



Şekil 13: Araçlar, Yakıttip İlişkisi

### 2.3.4. Araçlar, Vites İlişkisi

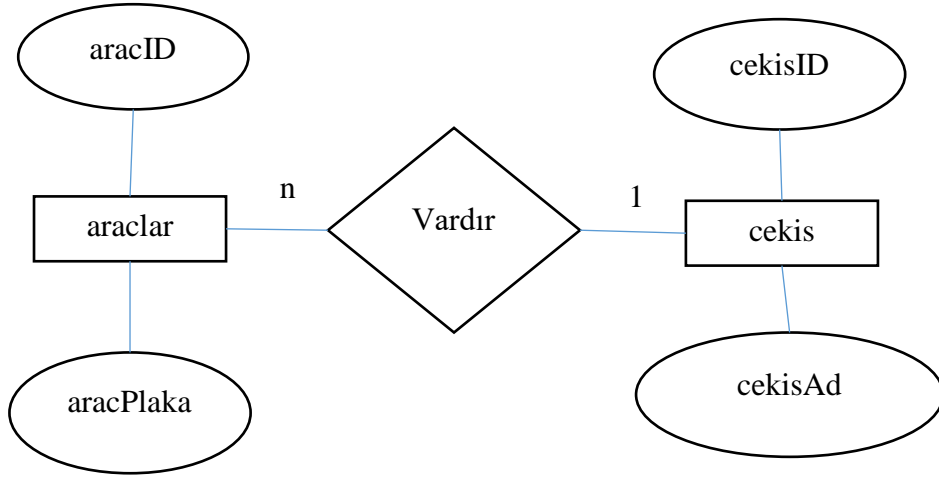
Bir vites tipinde birçok araç olabilir ama bir aracın yalnızca bir vitesi olur. Araçlar ve vites varlık kümeleri arasındaki ilişki 1:n ilişkidir. Araclar\_vites varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 14’de gösterilmiştir.



Şekil 14: Araçlar, Vites İlişkisi

### 2.3.5. Araçlar, Cekis İlişkisi

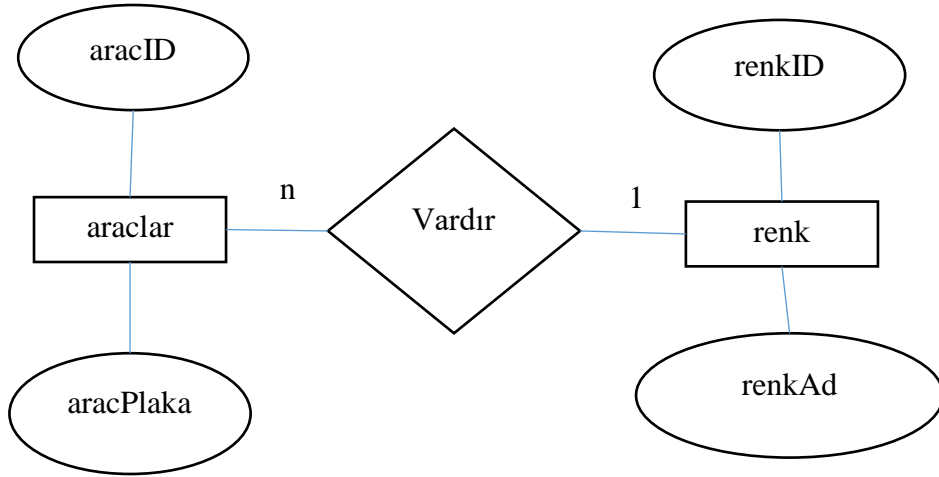
Bir çekiş tipinde birçok araç olabilir ama bir aracın yalnızca bir çekiş tipi olur. Araçlar ve cekis varlık kümeleri arasındaki ilişki 1:n ilişkidir. Araclar\_cekis varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 15’de gösterilmiştir.



Şekil 15: Araçlar, Cekis İlişkisi

### 2.3.6. Araçlar, Renk İlişkisi

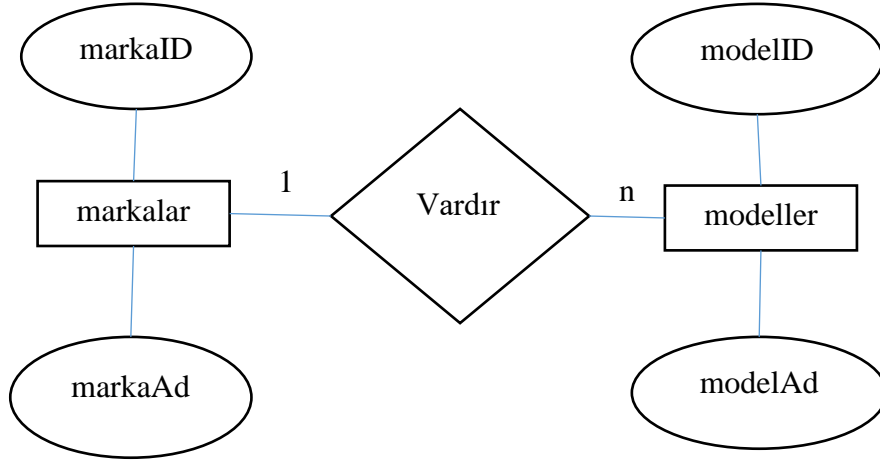
Bir renkte birçok araç olabilir ama bir aracın yalnızca bir rengi olur. Araçlar ve renk varlık kümeleri arasındaki ilişki 1:n ilişkidir. Araclar\_renk varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 16’da gösterilmiştir.



Şekil 16: Araçlar, Renk İlişkisi

### 2.3.7. Markalar, Modeller İlişkisi

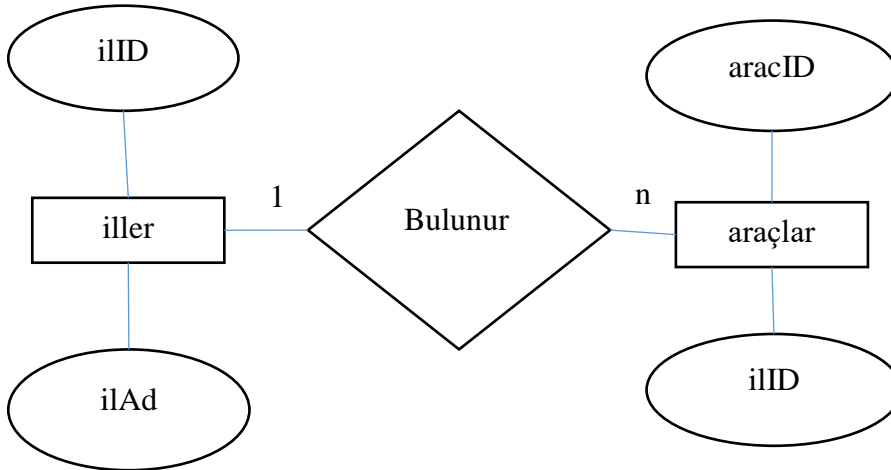
Her markanın birçok modeli olur, her model bir markaya aittir. Aralarındaki ilişki 1:n'dir. Markalar\_Modeller varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 17'de gösterilmiştir.



Şekil 17: Markalar, Modeller İlişkisi

### 2.3.8. İller, Araçlar İlişkisi

Bir ilde birçok araç olabilir fakat bir araç sadece bir ilde bulunabilir. İller\_Araclar varlık kümeleri arasındaki ilişki ER şeması kullanılarak Şekil 18'de gösterilmiştir.

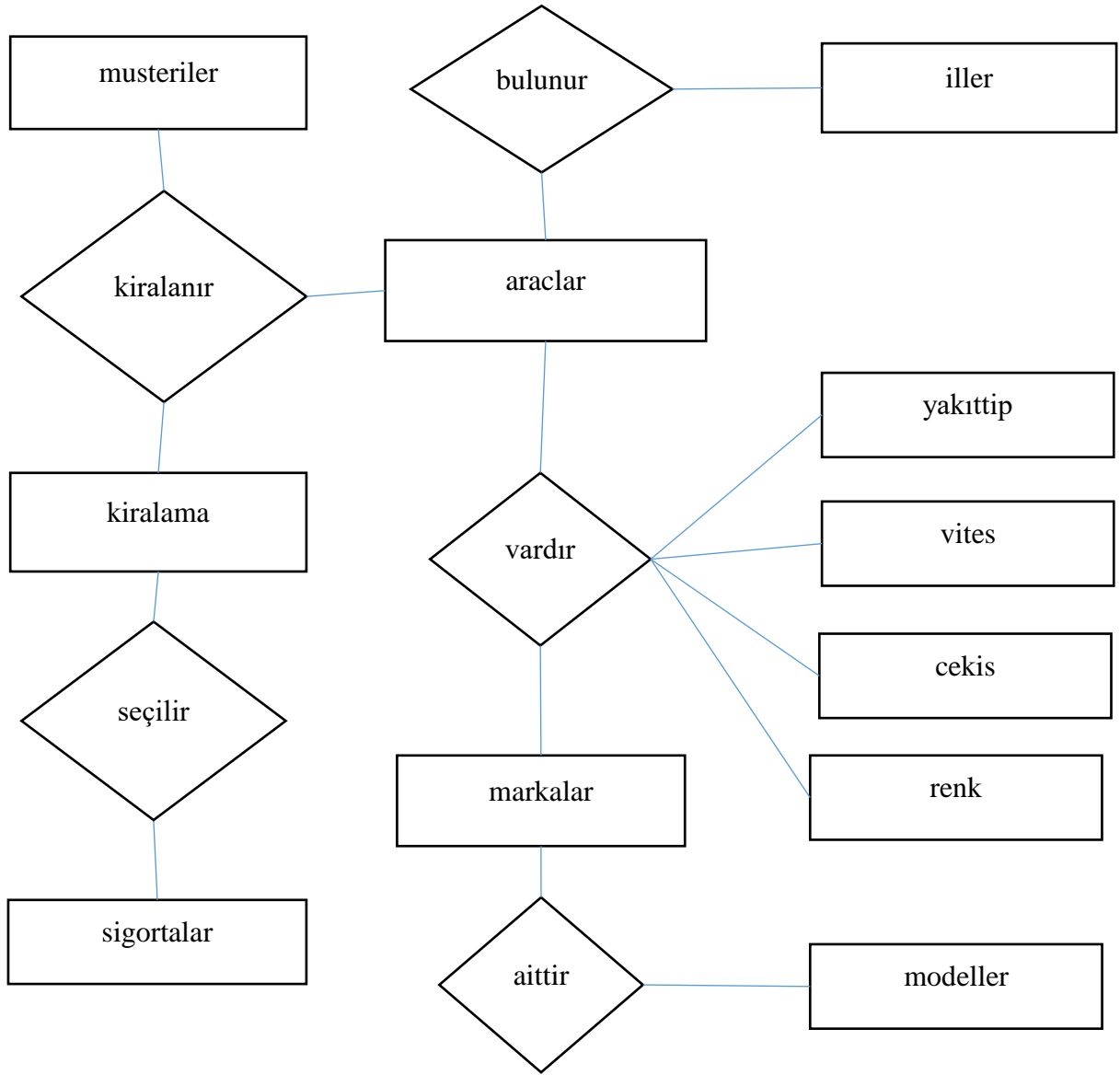


Şekil 18: Araçlar, İl İlişkisi



### 2.3.9. Genel Gösterim

Varlık kümeleri arasındaki ilişkiler toplu olarak Şekil 19’da gösterilmiştir.



**Şekil 19:** Genel Gösterim

## 2.4. Mantıksal Tasarım

Veri tabanı tasarımında varlık kümeleri arasındaki ilişki yapıldıktan sonra mantıksal tasarım kısmına geçilmiştir. Bu aşamada;

### Müşteriler Tablosu;

- musteriID **Birincil Anahtar** olarak kullanılmıştır.

Bu durumda Müşteriler tablosunun gösterimi Tablo 1’de yapılmıştır.

**Tablo 1:** Müşteriler Tablosu

<b>musteriler</b>
<b><u>musteriID</u></b>
musteriTc
musteriAd
musteriSoyad
musteriCinsiyet
musteriDogumTarihi
musteriAdres
musteriMail
musteriPhone
musteriEhliyetNo
musteriEhliyetTur

### İller Tablosu;

- ilID **Birincil Anahtar** olarak kullanılmıştır.

İller tablosunun gösterimi Tablo 2’de yapılmıştır.

**Tablo 2:** İller Tablosu

<b>iller</b>
<b><u>ilID</u></b>
ilAd

### Sigortalar Tablosu;

- sigortaID **Birincil Anahtar** olarak kullanılmıştır.

Sigortalar tablosunun gösterimi Tablo 3’de yapılmıştır.

**Tablo 3:** Sigortalar Tablosu

<b>sigortalar</b>
<b><u>sigortaID</u></b>
sigortaAd
sigortaUcret
sigortaBilgi

### Markalar Tablosu;

- markaID **Birincil Anahtar** olarak kullanılmıştır.

Markalar tablosunun gösterimi Tablo 4’de yapılmıştır.

**Tablo 4:** Markalar Tablosu

<b>markalar</b>
<b><u>markaID</u></b>
markaAd

### Modeller Tablosu;

- modelID **Birincil Anahtar** olarak kullanılmıştır.
- Markalar tablosuna bağlantı için modelMarka yabancı anahtar olarak kullanılmıştır.
- Kasa Tip tablosuna bağlantı için modelKasa yabancı anahtar olarak kullanılmıştır.
- Segmentler tablosuna bağlantı için modelSegment yabancı anahtar olarak kullanılmıştır.

Bu durumda Modeller tablosunun gösterimi Tablo 5’de yapılmıştır.

**Tablo 5:** Modeller Tablosu

<b>modeller</b>
<b><u>modelID</u></b>
modelAd
<u>markaID</u>

### Araçlar Tablosu;

- aracID Birincil Anahtar olarak kullanılmıştır.
- Diğer tablolarla bağlantı kurabilmek için modelID, yakittipID, vitesID, cekisID, renkID ve ilID yabancı anahtar olarak kullanılmıştır.

Bu durumda Araçlar tablosunun gösterimi Tablo 6’da yapılmıştır.

**Tablo 6:** Araçlar Tablosu

<b>araclar</b>
<b><u>aracID</u></b>
aracPlaka
modelID
aracYil
yakittipID
vitesID
aracKm
cekisID
aracKisi
renkID
aracKlima
ilID
aracAdres
aracKiraDurum
aracKiraUcret

### Kiralama Tablosu;

Müşteriler ve araçlar varlık kümelerinin ilişkilendirilmesinden ortaya çıkan tablodur.

- kiraID Birincil Anahtar olarak kullanılmıştır.
- Diğer tablolara bağlantı kurabilmek için musterIID, aracID, ilID, ve sigortaID yabancı anahtar olarak kullanılmıştır.

Bu durumda Kiralama tablosunun gösterimi Tablo 7’de yapılmıştır.

**Tablo 7:** Kiralama Tablosu

<b>kiralama</b>
<b><u>kiraID</u></b>
musterIID
aracID
sigortaID

kiraUcret
kiraGirisTarih
kiraBitisTarih
<b>ilID</b>
kiraAdres

#### **Yakıttip Tablosu;**

- yakıttipID Birincil Anahtar olarak kullanılmıştır.

Bu durumda Yakıttip tablosunun gösterimi Tablo 8’de yapılmıştır.

**Tablo 8:** Yakıttip Tablosu

<b>Yakıttip</b>
<b><u>yakıttipID</u></b>
yakıttipAd

#### **Vites Tablosu;**

- vitesID Birincil Anahtar olarak kullanılmıştır.

Bu durumda Vites tablosunun gösterimi Tablo 9’da yapılmıştır.

**Tablo 9:** Vites Tablosu

<b>Vites</b>
<b><u>vitesID</u></b>
vitesAd

#### **Cekis Tablosu;**

- cekisID Birincil Anahtar olarak kullanılmıştır.

Bu durumda Cekis tablosunun gösterimi Tablo 10’da yapılmıştır.

**Tablo 10:** Cekis Tablosu

<b>Cekis</b>
<b><u>cekisID</u></b>
cekisAd

### Renk Tablosu;

- renkID Birincil Anahtar olarak kullanılmıştır.

Bu durumda renk tablosunun gösterimi Tablo 11’de yapılmıştır.

**Tablo 11:** Renk Tablosu

Renk
<u>renkID</u>
renkAd

## 2.5. Fiziksel Tasarım

Proje MSSQL Server programı ile tasarlanmıştır. Projede toplamda 9 adet tablo bulunmaktadır. Bulunan tablo adları aşağıda listelendiği gibidir.

- **musteriler:** Müşteri bilgilerinin kayıt altında tutulduğu tablodur.
- **markalar:** Marka bilgilerinin olduğu tablodur.
- **modeller:** Araç modellerinin olduğu tablodur.
- **yakıttip:** Araç yakıt tiplerinin olduğu tablodur.
- **cekis:** Araç çekiş tiplerinin olduğu tablodur.
- **renk:** Araç renklerinin olduğu tablodur.
- **vites:** Araç vites tiplerinin olduğu tablodur.
- **araclar:** Araç bilgilerinin kayıt altında tutulduğu tablodur.
- **iller:** İl bilgilerinin olduğu tablodur.
- **sigorta:** Araç sigorta bilgilerinin bulunduğu tablodur.
- **kiralama:** Müşterilerin kiraladıkları araçların bilgilerinin olduğu tablodur.

### 2.5.1. musteriler Tablosu

Müşteriler Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- musterIID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- musterITC, varchar veri tipi
- musterAd, varchar veri tipi
- musterSoyad, varchar veri tipi
- musterCinsiyet, varchar veri tipi
- musterDogumTarihi, date veri tipi
- musterAdres, varchar veri tipi
- musterEmail, varchar veri tipi
- musterTel, varchar veri tipi
- musterEhliyetNo, varchar veri tipi
- musterEhliyetTur, varchar veri tipi

PC\SQLEXPRESS.arac...a - dbo.musteriler* ❏ ✕			
	Column Name	Data Type	Allow Nulls
▶	musteriID	int	<input type="checkbox"/>
	musteriTC	varchar(11)	<input type="checkbox"/>
	musteriAd	varchar(80)	<input type="checkbox"/>
	musteriSoyad	varchar(80)	<input type="checkbox"/>
	musteriCinsiyet	varchar(1)	<input type="checkbox"/>
	musteriDogumTarihi	date	<input type="checkbox"/>
	musteriAdres	text	<input type="checkbox"/>
	musteriMail	varchar(80)	<input type="checkbox"/>
	musteriPhone	varchar(11)	<input type="checkbox"/>
	musteriEhliyetNo	varchar(6)	<input type="checkbox"/>
	musteriEhliyetTur	varchar(2)	<input type="checkbox"/>
			<input type="checkbox"/>

**Şekil 20:** musteriler Tablosu

### 2.5.2. markalar Tablosu:

Markalar Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- markaID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- markaAd, varchar veri tipi

PC\SQLEXPRESS.arac...ma - dbo.markalar* ❏ ✕			
	Column Name	Data Type	Allow Nulls
▶	markaID	int	<input type="checkbox"/>
	markaAd	varchar(80)	<input type="checkbox"/>
			<input type="checkbox"/>

**Şekil 21:** markalar Tablosu

### 2.5.3. modeller Tablosu:

Modeller Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- modelID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- modelAd, varchar veri tipi
- modelMarka int veri tipi. Yabancı Anahtar

Column Name	Data Type	Allow Nulls
modelID	int	<input type="checkbox"/>
modelAd	varchar(100)	<input type="checkbox"/>
modelMarka	int	<input type="checkbox"/>

Şekil 22: modeller Tablosu

### 2.5.4. yakittip Tablosu:

Yakittip Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- yakittipID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- yakittipAd, varchar veri tipi

Column Name	Data Type	Allow Nulls
yakittipID	int	<input type="checkbox"/>
yakittipAd	varchar(30)	<input checked="" type="checkbox"/>

Şekil 23: yakittip Tablosu

### 2.5.5. cekis Tablosu:

Cekis Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- cekisID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- cekisAd, varchar veri tipi

Column Name	Data Type	Allow Nulls
cekisID	int	<input type="checkbox"/>
cekisAd	varchar(30)	<input checked="" type="checkbox"/>

Şekil 24: cekis Tablosu



### 2.5.6. renk Tablosu:

Renk Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- renkID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- renkAd, varchar veri tipi

Column Name	Data Type	Allow Nulls
renkID	int	<input type="checkbox"/>
renkAd	varchar(50)	<input checked="" type="checkbox"/>

Şekil 25: renk Tablosu

### 2.5.7. vites Tablosu:

Vites Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- vitesID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- vitesAd, varchar veri tipi

Column Name	Data Type	Allow Nulls
vitesID	int	<input type="checkbox"/>
vitesAd	varchar(50)	<input checked="" type="checkbox"/>

Şekil 26: vites Tablosu

### 2.5.8. araçlar Tablosu

Araçlar Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- aracID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- aracPlaka, varchar veri tipi
- modelID, int veri tipi, Yabancı Anahtar
- aracYıl, varchar veri tipi
- yakittipID, varchar veri tipi, Yabancı Anahtar
- vitesID, varchar veri tipi, Yabancı Anahtar
- aracKm, int veri tipi
- cekisID, varchar veri tipi, Yabancı Anahtar
- aracKisi, varchar veri tipi
- renkID, varchar veri tipi, Yabancı Anahtar
- aracKlima, varchar veri tipi
- ilID, int veri tipi, Yabancı Anahtar
- aracAdres, varchar veri tipi
- aracKiraUcret, int veri tipi
- aracKiraDurum, varchar veri tipi

PC\SQLEXPRESS.ara...lama - dbo.aracilar -> X			
	Column Name	Data Type	Allow Nulls
PK	aracID	int	<input type="checkbox"/>
	aracPlaka	varchar(8)	<input type="checkbox"/>
	aracModel	int	<input type="checkbox"/>
	aracYil	varchar(4)	<input type="checkbox"/>
	aracYakit	varchar(30)	<input type="checkbox"/>
	aracVites	varchar(30)	<input type="checkbox"/>
	aracKm	int	<input type="checkbox"/>
	aracCekis	varchar(30)	<input type="checkbox"/>
	aracKisi	varchar(3)	<input type="checkbox"/>
	aracRenk	varchar(30)	<input type="checkbox"/>
	aracKlima	varchar(1)	<input type="checkbox"/>
	aracIi	int	<input type="checkbox"/>
	aracAdres	varchar(1000)	<input type="checkbox"/>
	aracKiraUcret	int	<input type="checkbox"/>
	aracKiraDurum	varchar(1)	<input type="checkbox"/>
			<input type="checkbox"/>

Şekil 27: aracilar Tablosu

### 2.5.9. iller Tablosu

İller Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- ilID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- ilAd, varchar veri tipi

PC\SQLEXPRESS.arackiralama - dbo.iller* -> X			
	Column Name	Data Type	Allow Nulls
PK	ilID	int	<input type="checkbox"/>
	ilAd	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Şekil 28: iller Tablosu

### 2.5.10. sigortalarTablosu

Sigortalar Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

- sigortaID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- sigortaAd, varchar veri tipi
- sigortaUcret, int veri tipi
- sigortaBilgi, text veri tipi

PC\SQLEXPRESS.arac...a - dbo.sigortalar* -> X			
	Column Name	Data Type	Allow Nulls
PK	sigortalID	int	<input type="checkbox"/>
	sigortaAd	varchar(80)	<input type="checkbox"/>
	sigortaUcret	int	<input type="checkbox"/>
	sigortaBilgi	text	<input type="checkbox"/>
			<input type="checkbox"/>

**Şekil 29:** sigortalar Tablosu

### 2.5.11. kiralama Tablosu

Kiralama Tablosunun alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

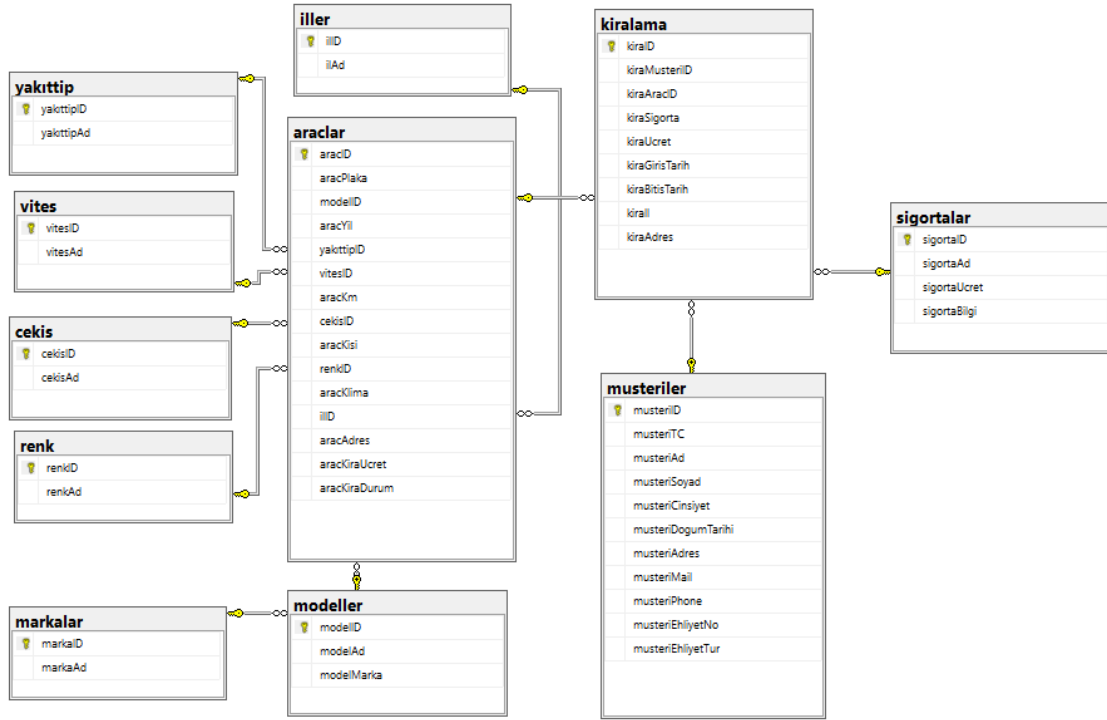
- kiraID, int veri tipi, Birincil Anahtar, Otomatik Artan ve Benzersiz
- kiraMusteriID, int veri tipi, Yabancı Anahtar
- kiraAracID, int veri tipi, Yabancı Anahtar
- kiraSigorta, int veri tipi
- kiraUcret, int veri tipi
- kiraGirisTarih, date veri tipi
- kiraBitisTarih, date veri tipi
- kiraİl, int veri tipi
- kiraAdres, varchar veri tipi

PC\SQLEXPRESS.arac...ma - dbo.kiralama -> X			
	Column Name	Data Type	Allow Nulls
PK	kiraID	int	<input type="checkbox"/>
	kiraMusteriID	int	<input type="checkbox"/>
	kiraAracID	int	<input type="checkbox"/>
	kiraSigorta	int	<input type="checkbox"/>
	kiraUcret	int	<input type="checkbox"/>
	kiraGirisTarih	date	<input type="checkbox"/>
	kiraBitisTarih	date	<input type="checkbox"/>
	kirall	int	<input type="checkbox"/>
	kiraAdres	varchar(1000)	<input type="checkbox"/>
			<input type="checkbox"/>

**Şekil 30:** kiralama Tablosu

## 2.5.12. Veri tabanı Diyagramı

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



Şekil 31: Veri tabanı Diyagramı

### 2.5.13. Tabloların SQL Kodları

Oluşturulan Tablolara ait SQL Kodları aşağıda sunulmuştur.

```
USE [arackiralama]
```

```
GO
```

```
/****** Object: Table [dbo].[araclar]   Script Date: 18.4.2021 14:05:27 *****/
```

```
SET ANSI_NOT_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

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

```
    [aracID] [int] IDENTITY(1,1) NOT NULL,
```

```
    [aracPlaka] [varchar](8) NOT NULL,
```

```
    [modelID] [int] NOT NULL,
```

```
    [aracYil] [varchar](4) NOT NULL,
```

```
    [yakittipID] [int] NOT NULL,
```

```
    [vitesID] [int] NOT NULL,
```

```
    [aracKm] [int] NOT NULL,
```

```
    [cekisID] [int] NOT NULL,
```

```
    [aracKisi] [varchar](3) NOT NULL,
```

```
    [renkID] [int] NOT NULL,
```

```
    [aracKlima] [varchar](1) NOT NULL,
```

```
    [ilID] [int] NOT NULL,
```

```
    [aracAdres] [varchar](1000) NOT NULL,
```

```
    [aracKiraUcret] [int] NOT NULL,
```

```
    [aracKiraDurum] [varchar](1) NOT NULL,
```

```
CONSTRAINT [PK_araclar] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [aracID] 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].[araclar] WITH CHECK ADD CONSTRAINT [FK_araclar_cekis]  
FOREIGN KEY([cekisID])
```

```
REFERENCES [dbo].[cekis] ([cekisID])
```

GO

ALTER TABLE [dbo].[araclar] CHECK CONSTRAINT [FK\_araclar\_cekis]

GO

ALTER TABLE [dbo].[araclar] WITH CHECK ADD CONSTRAINT [FK\_araclar\_iller]  
FOREIGN KEY([ilID])

REFERENCES [dbo].[iller] ([ilID])

GO

ALTER TABLE [dbo].[araclar] CHECK CONSTRAINT [FK\_araclar\_iller]

GO

ALTER TABLE [dbo].[araclar] WITH CHECK ADD CONSTRAINT [FK\_araclar\_modeller]  
FOREIGN KEY([modelID])

REFERENCES [dbo].[modeller] ([modelID])

GO

ALTER TABLE [dbo].[araclar] CHECK CONSTRAINT [FK\_araclar\_modeller]

GO

ALTER TABLE [dbo].[araclar] WITH CHECK ADD CONSTRAINT [FK\_araclar\_renk]  
FOREIGN KEY([renkID])

REFERENCES [dbo].[renk] ([renkID])

GO

ALTER TABLE [dbo].[araclar] CHECK CONSTRAINT [FK\_araclar\_renk]

GO

ALTER TABLE [dbo].[araclar] WITH CHECK ADD CONSTRAINT [FK\_araclar\_vites]  
FOREIGN KEY([vitesID])

REFERENCES [dbo].[vites] ([vitesID])

GO

ALTER TABLE [dbo].[araclar] CHECK CONSTRAINT [FK\_araclar\_vites]

GO

ALTER TABLE [dbo].[araclar] WITH CHECK ADD CONSTRAINT [FK\_araclar\_yakittip]  
FOREIGN KEY([yakittipID])

REFERENCES [dbo].[yakittip] ([yakittipID])

GO

ALTER TABLE [dbo].[araclar] CHECK CONSTRAINT [FK\_araclar\_yakittip]

GO

USE [arackiralama]

GO

/\*\*\*\*\*\* Object: Table [dbo].[cekis] Script Date: 18.4.2021 14:06:26 \*\*\*\*\*/

SET ANSI\_NOT\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[cekis](

[cekisID] [int] IDENTITY(1,1) NOT NULL,

[cekisAd] [varchar](30) NOT NULL,

CONSTRAINT [PK\_cekis] PRIMARY KEY CLUSTERED

(

[cekisID] 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

USE [arackiralama]

GO

/\*\*\*\*\*\* Object: Table [dbo].[iller] Script Date: 18.4.2021 14:06:55 \*\*\*\*\*/

SET ANSI\_NOT\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[iller](

[ilID] [int] IDENTITY(1,1) NOT NULL,

[ilAd] [varchar](50) NOT NULL,

CONSTRAINT [PK\_iller] PRIMARY KEY CLUSTERED

(

[ilID] 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

```

USE [arackiralama]
GO
/***** Object: Table [dbo].[kiralama]  Script Date: 18.4.2021 14:07:12 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[kiralama](
    [kiraID] [int] IDENTITY(1,1) NOT NULL,
    [kiraMusteriID] [int] NOT NULL,
    [kiraAracID] [int] NOT NULL,
    [kiraSigorta] [int] NOT NULL,
    [kiraUcret] [int] NOT NULL,
    [kiraGirisTarih] [date] NOT NULL,
    [kiraBitisTarih] [date] NOT NULL,
    [kiraIl] [int] NOT NULL,
    [kiraAdres] [varchar](1000) NOT NULL,
    CONSTRAINT [PK_kiralama] PRIMARY KEY CLUSTERED
(
    [kiraID] 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].[kiralama] WITH CHECK ADD CONSTRAINT
[FK_kiralama_aracilar] FOREIGN KEY([kiraAracID])
REFERENCES [dbo].[aracilar] ([aracID])
GO
ALTER TABLE [dbo].[kiralama] CHECK CONSTRAINT [FK_kiralama_aracilar]
GO
ALTER TABLE [dbo].[kiralama] WITH CHECK ADD CONSTRAINT
[FK_kiralama_musteriler] FOREIGN KEY([kiraMusteriID])
REFERENCES [dbo].[musteriler] ([musteriID])
GO

```



```

ALTER TABLE [dbo].[kiralama] CHECK CONSTRAINT [FK_kiralama_musteriler]
GO
ALTER TABLE [dbo].[kiralama] WITH CHECK ADD CONSTRAINT
[FK_kiralama_sigortalar] FOREIGN KEY([kiraSigorta])
REFERENCES [dbo].[sigortalar] ([sigortaID])
GO
ALTER TABLE [dbo].[kiralama] CHECK CONSTRAINT [FK_kiralama_sigortalar]
GO

```

```
USE [arackiralama]
```

```
GO
```

```
/****** Object: Table [dbo].[markalar] Script Date: 18.4.2021 14:07:47 *****/
```

```
SET ANSI_NOT_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```

CREATE TABLE [dbo].[markalar](
    [markaID] [int] IDENTITY(1,1) NOT NULL,
    [markaAd] [varchar](80) NOT NULL,
    CONSTRAINT [PK_markalar] 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

```

```
USE [arackiralama]
```

```
GO
```

```
/****** Object: Table [dbo].[modeller] Script Date: 18.4.2021 14:08:05 *****/
```

```
SET ANSI_NOT_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

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

```

[modelID] [int] IDENTITY(1,1) NOT NULL,
[modelAd] [varchar](100) NOT NULL,
[modelMarka] [int] NOT NULL,
CONSTRAINT [PK_modeller] PRIMARY KEY CLUSTERED
(
    [modelID] 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].[modeller] WITH CHECK ADD CONSTRAINT
[FK_modeller_markalar] FOREIGN KEY([modelMarka])
REFERENCES [dbo].[markalar] ([markaID])
GO
ALTER TABLE [dbo].[modeller] CHECK CONSTRAINT [FK_modeller_markalar]
GO

USE [arackiralama]
GO
/***** Object: Table [dbo].[musteriler] Script Date: 18.4.2021 14:08:27 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[musteriler](
    [musteriID] [int] IDENTITY(1,1) NOT NULL,
    [musteriTC] [varchar](11) NOT NULL,
    [musteriAd] [varchar](80) NOT NULL,
    [musteriSoyad] [varchar](80) NOT NULL,
    [musteriCinsiyet] [varchar](1) NOT NULL,
    [musteriDogumTarihi] [date] NOT NULL,
    [musteriAdres] [text] NOT NULL,
    [musteriMail] [varchar](80) NOT NULL,
    [musteriPhone] [varchar](11) NOT NULL,
    [musteriEhliyetNo] [varchar](6) NOT NULL,

```

```

[musteriEhliyetTur] [varchar](2) NOT NULL,
CONSTRAINT [PK_musteriler] PRIMARY KEY CLUSTERED
(
    [musteriID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
USE [arackiralama]
GO
/***** Object: Table [dbo].[renk] Script Date: 18.4.2021 14:08:46 *****/
SET ANSI_NOT_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[renk](
    [renkID] [int] IDENTITY(1,1) NOT NULL,
    [renkAd] [varchar](50) NOT NULL,
    CONSTRAINT [PK_renk] PRIMARY KEY CLUSTERED
(
    [renkID] 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
USE [arackiralama]
GO
/***** Object: Table [dbo].[sigortalar] Script Date: 18.4.2021 14:09:06 *****/
SET ANSI_NOT_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[sigortalar](

```

```

[sigortaID] [int] IDENTITY(1,1) NOT NULL,
[sigortaAd] [varchar](80) NOT NULL,
[sigortaUcret] [int] NOT NULL,
[sigortaBilgi] [text] NOT NULL,
CONSTRAINT [PK_sigortalar] PRIMARY KEY CLUSTERED
(
    [sigortaID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

```

```
USE [arackiralama]
```

```
GO
```

```
/****** Object: Table [dbo].[vites] Script Date: 18.4.2021 14:09:23 *****/
```

```
SET ANSI_NOT_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```

CREATE TABLE [dbo].[vites](
    [vitesID] [int] IDENTITY(1,1) NOT NULL,
    [vitesAd] [varchar](50) NOT NULL,
    CONSTRAINT [PK_vites] PRIMARY KEY CLUSTERED
(
    [vitesID] 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

```

```
USE [arackiralama]
```

```
GO
```

```
/****** Object: Table [dbo].[yakittip] Script Date: 18.4.2021 14:09:39 *****/
```

```
SET ANSI_NOT_NULLS ON
```

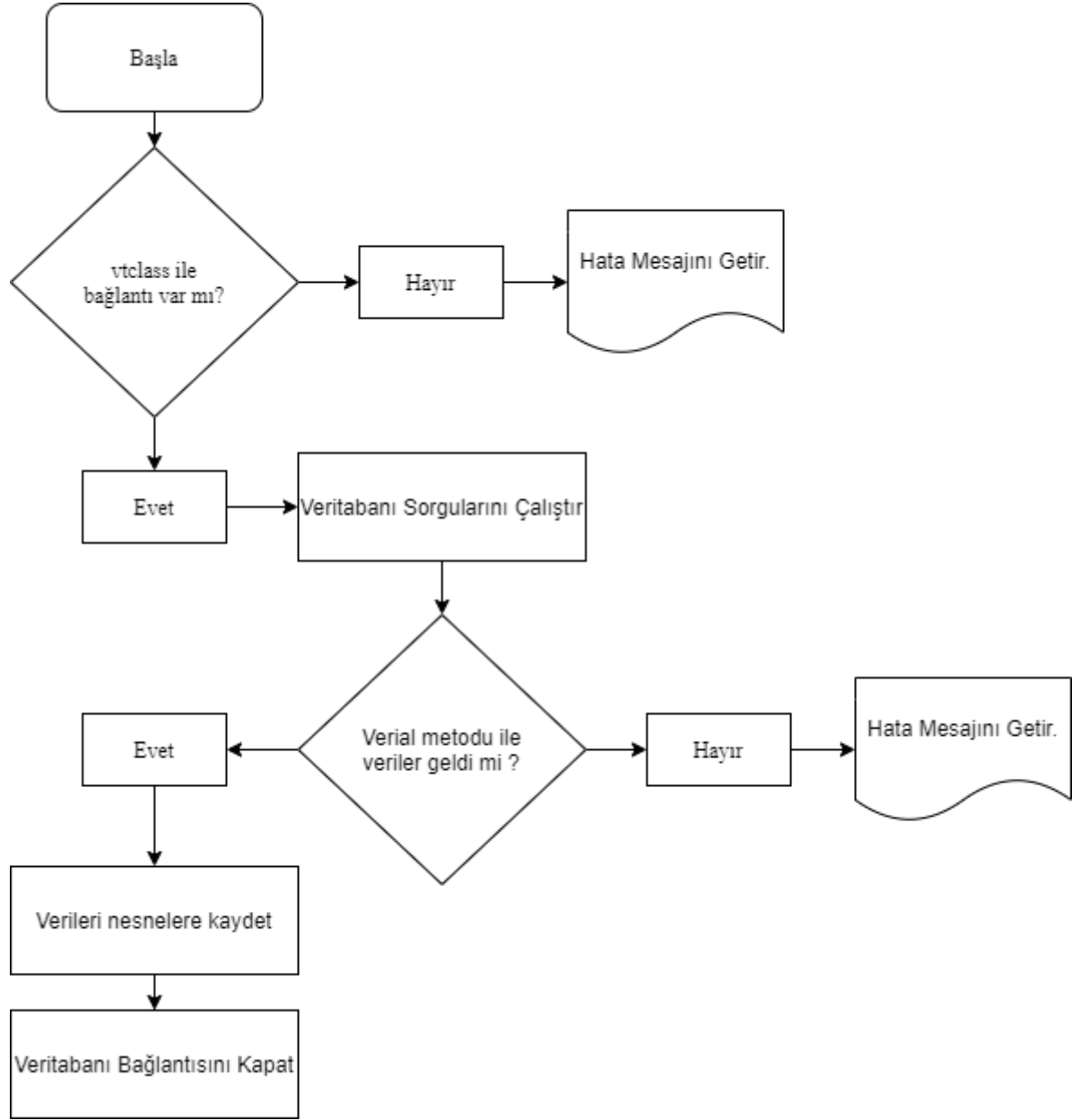
```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[yakittip](
    [yakittipID] [int] IDENTITY(1,1) NOT NULL,
    [yakittipAd] [varchar](30) NOT NULL,
    CONSTRAINT [PK_yakittip] PRIMARY KEY CLUSTERED
(
    [yakittipID] 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

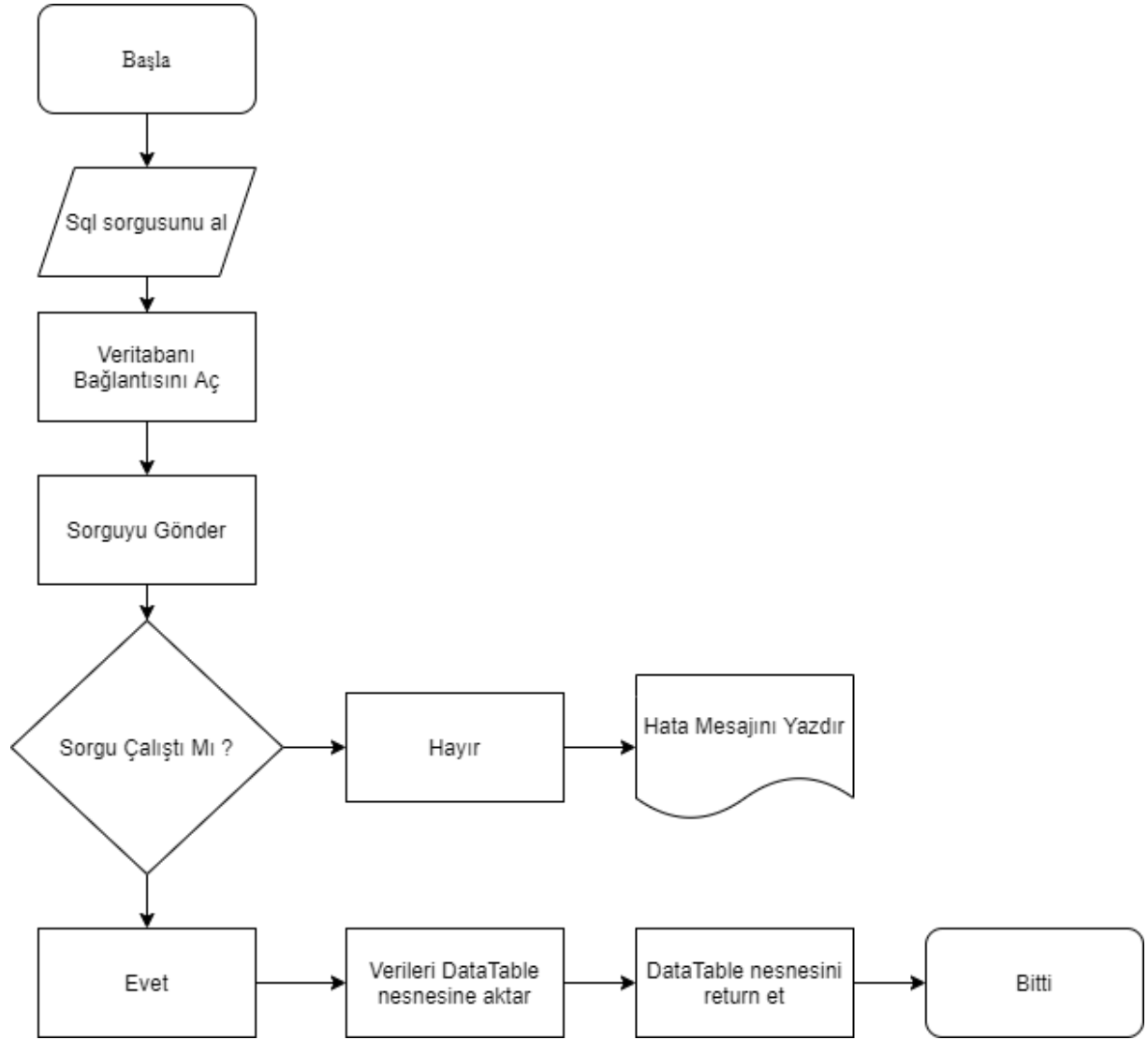
```

## 2.6. Projenin Akış Şeması

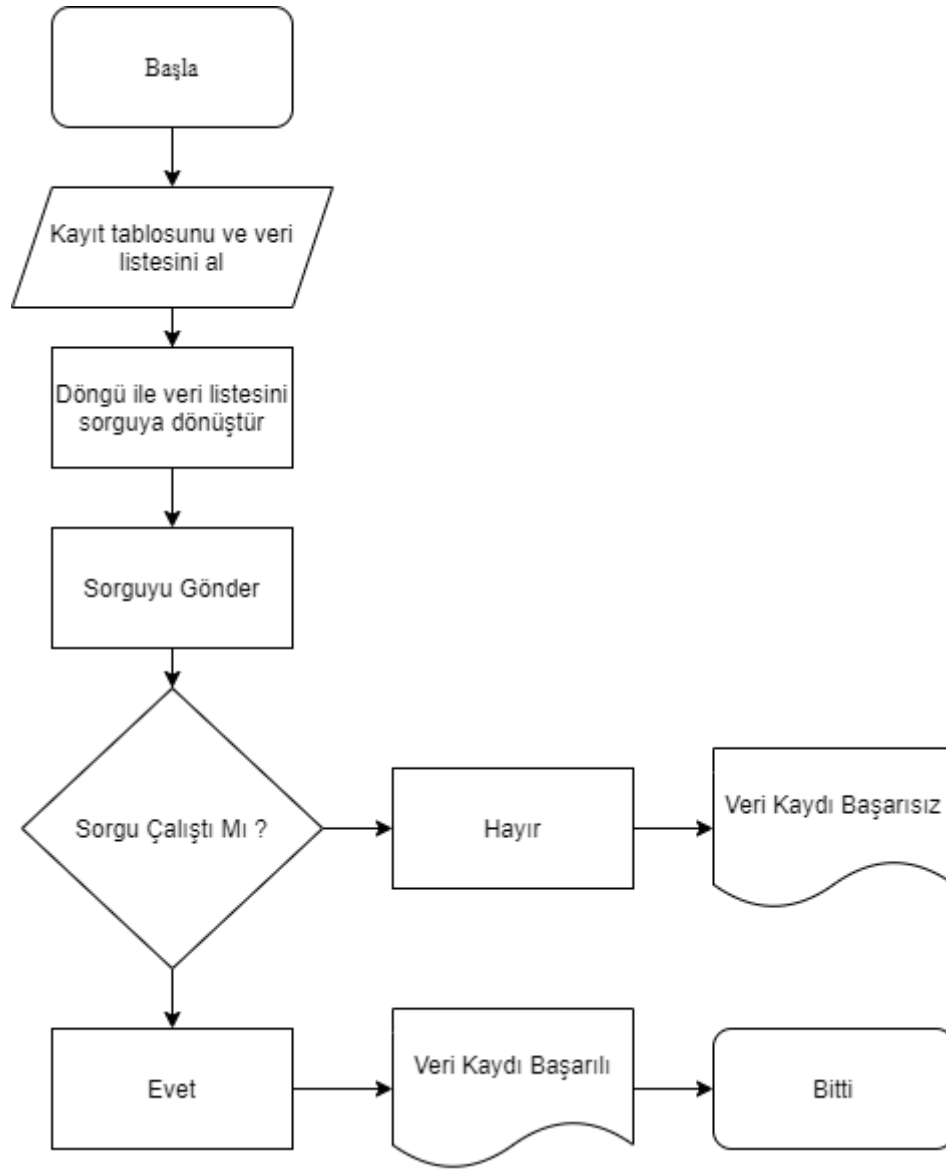
### 2.6.1. Ana Sayfa Akış Şeması



### 2.6.2. Veri Al Metodu Akış Şeması

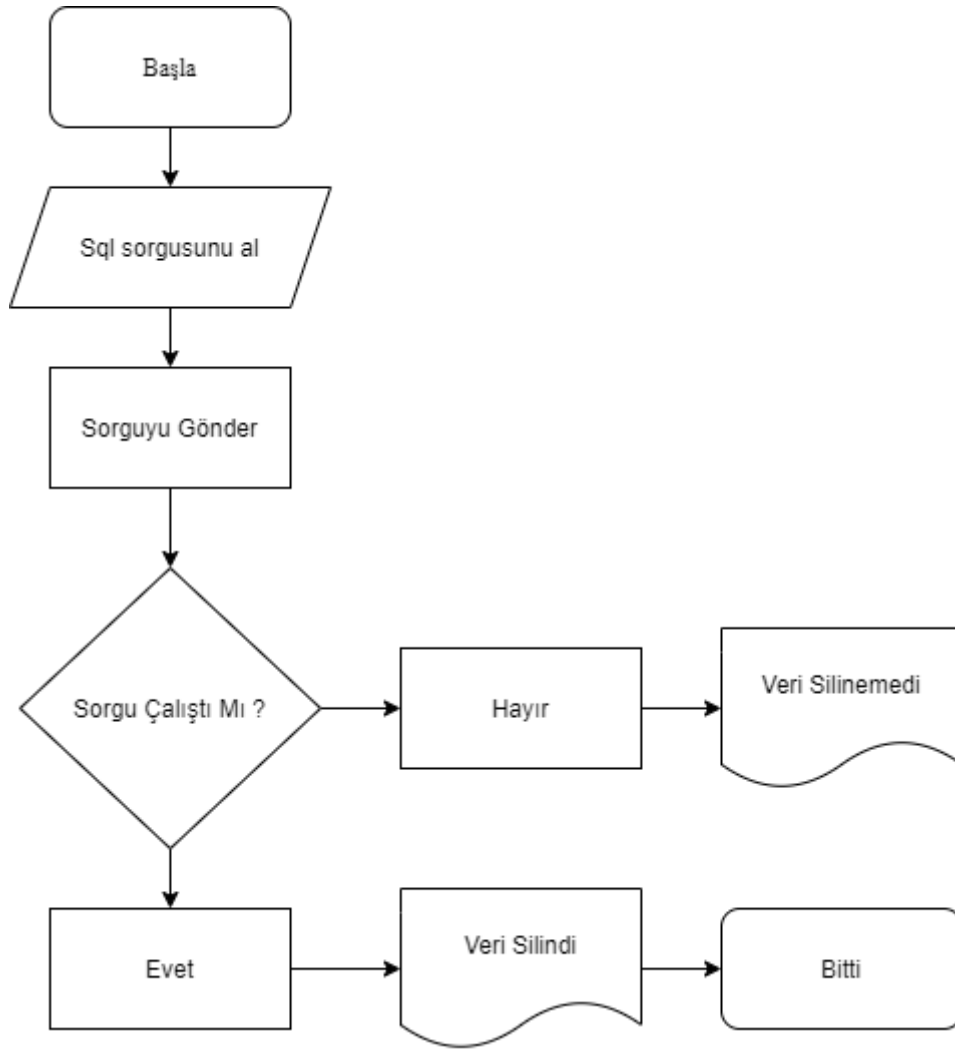


### 2.6.3. Veri Ekle Metodu Akış Şeması

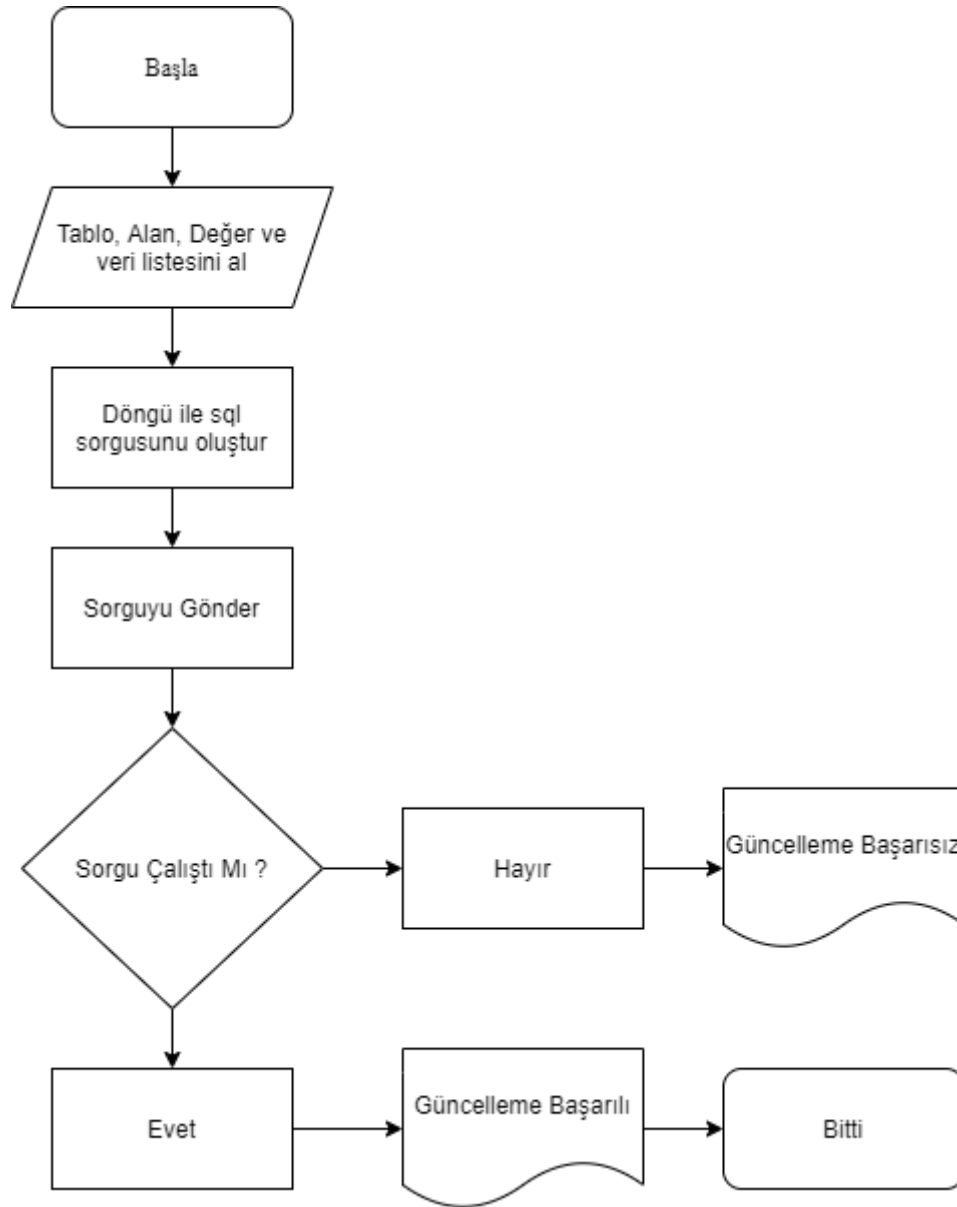




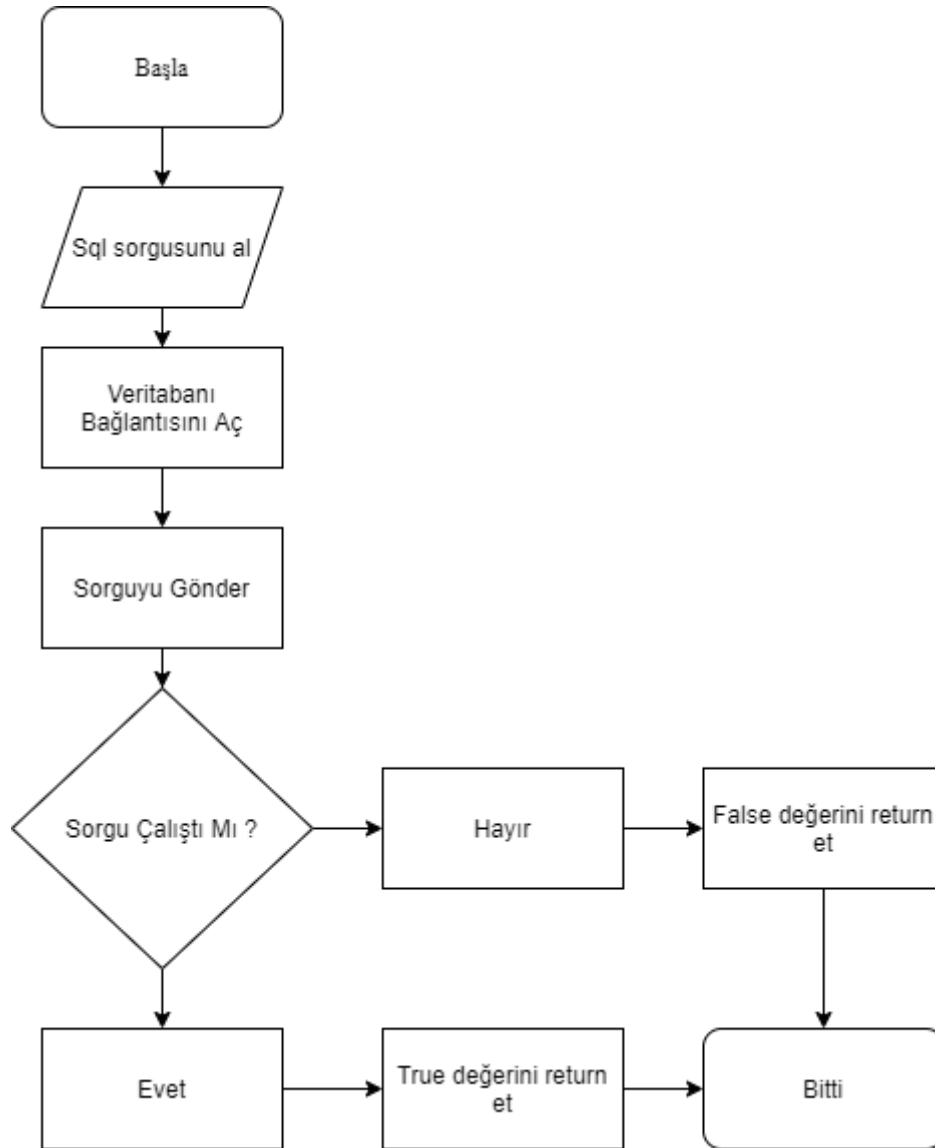
#### 2.6.4. Veri Sil Metodu Akış Şeması



### 2.6.5. Kayıt Güncelle Metodu Akış Şeması



### 2.6.6. Sorgu Gönder Metodu Akış Şeması



## 2.7. Projenin Uygulama Kısımı

### 2.7.1. Ana Sayfa Ekranı

Öncelikli olarak MSSQL veri tabanı ile bağlantı kurulması sağlanmıştır. Ana sayfanın sol tarafındaki menü ana menüdür. Yapılacak olan ana işlem bu menüden seçilmektedir. Bu menüler de yine kendi içlerinde uygulamanın yukarısında görülen menü ile bölümlere ayrılmaktadır.

- Ana sayfanın bölümleri sırayla;
- Ana Sayfa,
- Kiradaki Araçlar,
- Araç Kirala,
- Araçlar Tam Listesi,
- Müşteriler Tam Listesi.



#### Ana Sayfa:

Ana sayfada sistem tarihi, saati ve birkaç temel istatistik görüntülenmektedir.

Araç Kiralama Sistemi Yönetim Paneli

### Kiradaki Araçlar:

Kiradaki araçlar sayfasında halen kiralamada olan araçların bir listesi DataGridView ile gösterilir. Bu listede;

- aracPlaka,
- modelAd,
- aracYil,
- yakittipAd,
- vitesAd,
- aracKm,
- cekisAd,
- aracKisi,
- renkAd,
- aracKlima,
- ilAd,
- aracKiraUcret bilgileri bulunur.

Yönetim Paneli

Ana Sayfa

Kiradaki Araçlar

Araç Kirala

Araçlar Tam Liste

Müşteriler Tam Liste

Ana Sayfa

Araç İşlemleri

Müşteri İşlemleri

Sigorta İşlemleri

İstatistik

Araç Kiralama

Kiralaayan Müşteri :

12345698745

Kiralanacak Araç :

Symbol Sedan 06AZ1234

100 TL

Araç Sigortası :

Temel Sigorta

450 TL

Kiralanacak İl :

Adana

Kiralama Başlangıç Tarihi :

30 Mayıs 2021 Pazar

Kiralama Bitiş Tarihi :

30 Mayıs 2021 Pazar

0 Gün

Toplam Tutar: 450 TL

Araç Kiraya Ver

Araç Adresi :

Kira İade

Kiradan Çıkarılacak Araç :

Focus Sedan 01DC1632

Araç Kiradan Çıkar

## Araç Kirala:

Araç kiralama işlemi için öncelikle gerekli bilgiler Combobox, RichTextBox ve DateTimePicker ile girilir. Seçilen bilgilere göre fiyat bilgileri Label ile görüntülenerek kolaylık sağlar. Gerekli tüm bilgiler girildikten sonra Aracı Kiraya Ver butonu ile kiralama işlemi tamamlanır.

Kirada olan bir aracı kiradan çıkartmak için ise Combobox yardımıyla model ve plakasına bakarak kiradan çıkartılacak olan araç seçilir ve ardından Aracı Kiradan Çıkar botunu ile kiradan çıkartma işlemi tamamlanır.

Yönetim Paneli	Ana Sayfa	Kiradaki Araçlar	Araç Kirala	Araçlar Tam Liste	Müşteriler Tam Liste
Ana Sayfa					
Araç İşlemleri					
Müşteri İşlemleri					
Sigorta İşlemleri					
İstatistik					

aracPlaka	modelAd	aracYil	yakittipAd	vitesAd	aracKm	cekisAd	aracKisi	renkAd	aracKlima	ilAd	aracKiraUcret
06AZ1234	Symbol Sedan	2000	Benzin	Manuel	40000	Önden Çekişli	5	Gümüş Gri	0	Ankara	100
06ZA4321	Elantra Sedan	2016	Benzin & LPG	Manuel	50000	Önden Çekişli	5	Turuncu	1	Ordu	250
01AZ1234	Polo Hatchback	2014	Benzin	Otomatik	135000	Önden Çekişli	5	Gri	0	Adana	200
06CZ2514	Egea Sedan	2002	Dizel	Manuel	150000	Önden Çekişli	5	Beyaz	1	Ankara	180
01DC1632	Focus Sedan	2015	Benzin & LPG	Otomatik	145000	Önden Çekişli	5	Beyaz	1	Adana	150
06BG1485	5.20i Sedan	2017	Benzin	Otomatik	20000	Arkadan Çekişli	5	Beyaz	1	Adıyaman	500
81BG1485	C180 Sedan	2016	Benzin	Otomatik	89000	Arkadan Çekişli	5	Siyah	1	Düzce	450

## Araçlar Tam Listesi:

Araçlar tam listesi sayfasında sistemde kayıtlı olan tüm araçların bir listesi DataGridView ile gösterilir. Bu listede;

- aracPlaka,
- modelAd,
- aracYil,
- yakittipAd,
- vitesAd,
- aracKm,
- cekisAd,
- aracKisi,
- renkAd,
- aracKlima,
- ilAd,
- aracKiraUcret bilgileri bulunur.

Araç Kiralama Sistemi Yönetim Paneli									
Yönetim Paneli	Ana Sayfa	Kiradaki Araçlar	Araç Kirala	Araçlar Tam Liste	Müşteriler Tam Liste				
Ana Sayfa									
Araç İşlemleri									
Müşteri İşlemleri									
Sigorta İşlemleri									
İstatistik									
	musteriAd	musteriSoyad	musteriCinsiyet	musteriDogumTarihi	musteriAdres	musteriMail	musteriPhone	musteriEhliyetNo	musteriEhliyet1
	Osman	Balci	0	27.12.1990	Falanca sokak filanca cadde	deneme@gmail.com	05321456984	415263	B
	Emel	Şahin	1	15.10.1992	Falanca Sokak Filanca Cadde	deneme2@mail.com	05365478526	418596	B
	Muarrem	Ekinci	0	30.12.1984	Muarrem caddesi deneme sokağı	muarrem@gmail.com	05365214569	065214	BE
	Zuhuri	Altun	0	24.5.2021	Ankara merkez mahallesi 11 sokak	zuhuri@gmail.com	05303333333	748520	A
	Kubra	Yön	1	26.3.1998	altındağ	kubrayon@gmail.com	55555555555	413232	B

## Müşteriler Tam Listesi:

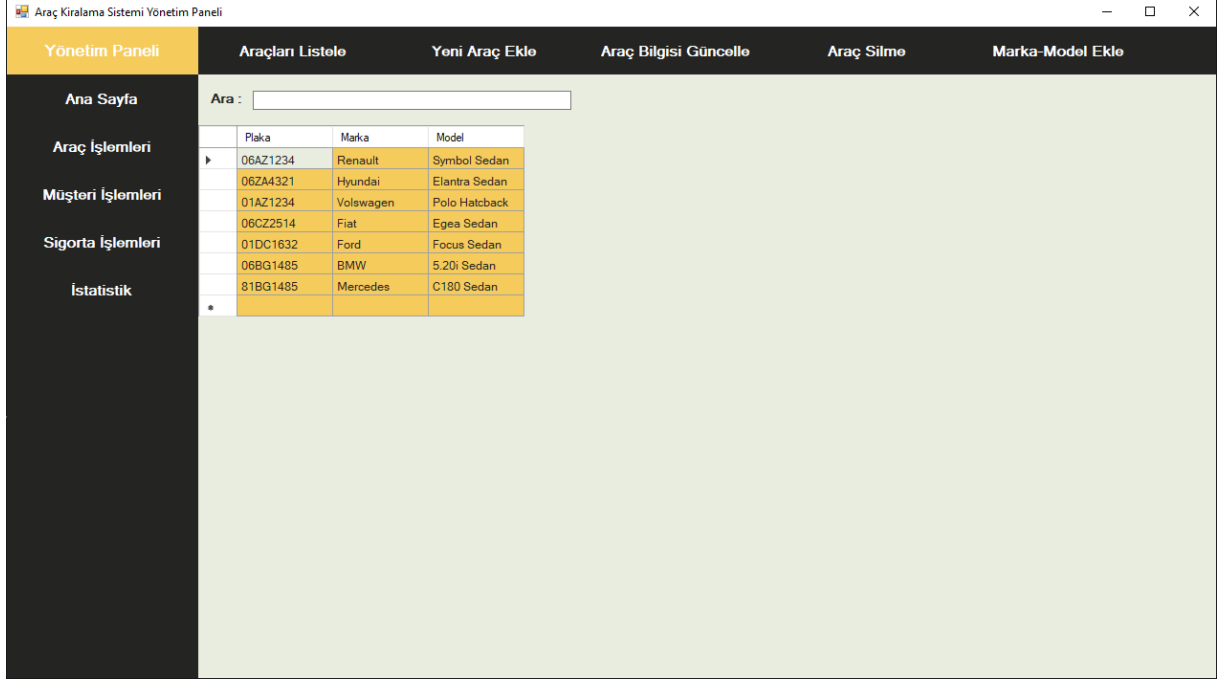
Müşteriler tam listesi sayfasında sistemde kayıtlı olan tüm müşterilerin bir listesi DataGridView ile gösterilir. Bu listede;

- musteriAd,
- musteriSoyad,
- musteriCinsiyet,
- musteriDogumTarihi,
- musteriAdres,
- musteriMail,
- musteriPhone,
- musteriEhliyetNo,
- musteriEhliyetTur bilgileri bulunur.

## 2.7.2. Araç İşlemleri Ekranı

Araç işlemleri ekranında araçlar ile ilgili yapılacak işlemler bulunur. Bu işlemler sırasıyla;

- Araçları Listele,
- Yeni Araç Ekle,
- Araç Bilgisi Güncelle,
- Araç Silme,
- Marka-Model Ekle işlemleridir.



### Araçları Listele:

Araçları listele sayfasında sistemde bulunan araçları birkaç temel bilgileri gelecek şekilde listeleyebiliriz. Bu bilgiler sırasıyla;

- Plaka,
- Marka,
- Model bilgileridir.

Aynı zamanda ara işlemi ile tüm araçları getirmek yerine plaka bilgisine göre belirli bir araç grubunun da gelmesi sağlanabilmektedir. Bunun için TextBox nesnesine istenilen plakanın bir kısmı ya da tamamının yazılması yeterlidir. Listeleme işlemi otomatik olarak yapılacaktır.





girilen metni içeren araçlar listelenir. Listedeki güncelleme yapılacak araç seçildiği zaman Aşağıda bulunan ComboBox, TextBox ve RichTextBox nesneleri otomatik olarak seçilen aracın bilgileriyle dolar. Sonrasında değiştirilmek istenen bilgiler düzenlenir ve Güncelle butonuna basılarak güncelleme işlemi tamamlanır. Bir hata ile karşılaşmadığı durumda işlemin tamamlandığına dair bir mesaj gösterilir.

**Araç Kiralama Sistemi Yönetim Paneli**

**Yönetim Paneli**

**Araçları Listele**   **Yeni Araç Ekle**   **Araç Bilgisi Güncelle**   **Araç Silme**   **Marka-Model Ekle**

**Ana Sayfa**

**Araç İşlemleri**

**Müşteri İşlemleri**

**Sigorta İşlemleri**

**İstatistik**

Ara :

Araç Plaka :

**Araç Sil**

Plaka	Marka	Model
06AZ1234	Renault	Symbol Sedan
06ZA4321	Hyundai	Elantra Sedan
01AZ1234	Volswagen	Polo Hatchback
06CZ2514	Fiat	Egea Sedan
01DC1632	Ford	Focus Sedan
06BG1485	BMW	5.20i Sedan
81BG1485	Mercedes	C180 Sedan

### Araç Silme:

Sistemde kayıtlı olan araçlardan biri silinmek istenir ise Araç İşlemleri ekranındaki araç silme sekmesinden silme işlemi yapılabilir. Bunun için ComboBox nesnesinden silinmek istenen aracın seçilmesi ve sonrasında Aracı Sil butonuna basılması yeterlidir. Silinmek istenen araç aracPlaka alanına bakılarak bulunur ve sistemden silinir. Eğer bir hata ile karşılaşılmaz ise aracın silindiğine dair bir onay mesajı gösterilir.

**Araç Kiralama Sistemi Yönetim Paneli**

**Yönetim Paneli**

**Araçları Listele**   **Yeni Araç Ekle**   **Araç Bilgisi Güncelle**   **Araç Silme**   **Marka-Model Ekle**

**Ana Sayfa**

**Araç İşlemleri**

**Müşteri İşlemleri**

**Sigorta İşlemleri**

**İstatistik**

Ara :

**Araç Silme**

**Araç Markası Ekle**

Marka Adı :

**Markayı Ekle**

Marka
Renault
Nissan
Fiat
Opel
Volswagen
Hyundai
Toyota

**Araç Modeli Ekle**

Marka Adı :

Model Adı :

**Modeli Ekle**

Model	Marka
5.20i Sedan	BMW
C4 Cactus Hat...	Citroen
Duster 4x4 SUV	Dacia
Egea Sedan	Fiat
Focus Sedan	Ford
Accent Blue Se...	Hyundai
Elantra Sedan	Hyundai
C180 Sedan	Mercedes
E180 Sedan	Mercedes
Micra Hatchback	Nissan
Astra Sedan	Opel
Symbol Sedan	Renault
Clio Hatchback	Renault
Megane Sedan	Renault
Plia A Hatchback	Renault

## Marka-Model Ekle:

Marka-Model ekle sekmesinden yeni araç markaları ve modelleri eklenebilir. Yeni bir marka eklemek için TextBox nesnesine eklenecek yeni markanın adının yazılması ve ardından Markayı Ekle butonuna basılması yeterlidir. Yeni bir model eklemek için ise öncelikle ComboBox nesnesinden eklenecek modelin markası seçilir ve ardından TextBox nesnesine modelin adı yazılarak Modeli Ekle butonuna basılır. Böylece marka veya model ekleme işlemi tamamlanmış olur. Diğer sekmelerde olduğu gibi burada da işlem başarıyla tamamlandığı zaman ekrana tamamlandığına dair bir mesaj gösterilir.

### 2.7.3. Müşteri İşlemleri Ekranı

Müşteri işlemleri ekranında müşteriler ile ilgili yapılacak işlemler bulunur. Bu işlemler sırasıyla;

- Müşterileri Listele,
- Müşteri Kayıt,
- Müşteri Güncelle,
- Müşteri Sil işlemleridir.

TC	Ad	Soyad
12345698745	Osman	Balci
12395478569	Emel	Şahin
14725836985	Muarrem	Ekinci
11111111111	Zuhuri	Altun
41125555555	Kübra	Yön

## Müşterileri Listele:

Müşterileri listele sayfasında sistemde bulunan araçları birkaç temel bilgileri gelecek şekilde listeleyebiliriz. Bu bilgiler sırasıyla;

- TC,
- Ad,
- Soyad bilgileridir.

Aynı zamanda ara işlemi ile tüm müşterileri getirmek yerine T.C. kimlik bilgisine göre belirli bir müşteri grubunun da gelmesi sağlanabilmektedir. Bunun için TextBox nesnesine istenilen T.C. kimliğin bir kısmı ya da tamamının yazılması yeterlidir. Listeleme işlemi otomatik olarak yapılacaktır.

Araç Kiralama Sistemi Yönetim Paneli

Yönetim Paneli   Müşterileri Listele   Müşteri Kayıt   Müşteri Güncelle   Müşteri Sil

Ana Sayfa

Araç İşlemleri

Müşteri İşlemleri

Sigorta İşlemleri

İstatistik

Ara :

TC	Ad	Soyad
12345698745	Osman	Balci
12395478569	Emel	Şahin
14725836985	Muarrem	Ekinci
11111111111	Zuhuri	Altun
41125555555	Kübra	Yön

TC No :

Ad :

Soyad :

Cinsiyet : Kadın

Doğum Tarihi : 30 Mayıs 2021 Pazar

Mail :

Tel No :

Ehliyet No :

Ehliyet Tür : M

Adres :

Kayıt Et

### Müşteri Kayıt:

Müşteri Kayıt sayfasında TextBox ComboBox, RichTextBox ve DateTimePicker nesneleri kullanılarak sisteme yeni bir müşteri eklenebilir. Öncelikle gerekli bilgiler doldurulur ve Kayıt Et butonuna basarak kayıt işlemi yapılır. Kayıt veritabanına gönderilmeden önce müşterinin zaten sistemde kayıtlı olup olmadığı T.C. kimlik alanına bakarak kontrol edilir eğer zaten böyle bir müşteri sistemde kayıtlı ise işlem gerçekleştirilmez ve müşterinin zaten sistemde bulunduğu dair bir hata gösterilir. Eğer herhangi bir sorun yok ise müşteri başarıyla sisteme kayıt edilir ve bir mesaj ile kaydın tamamlandığı bildirilir.

Araç Kiralama Sistemi Yönetim Paneli

Yönetim Paneli   Müşterileri Listele   Müşteri Kayıt   Müşteri Güncelle   Müşteri Sil

Ana Sayfa

Araç İşlemleri

Müşteri İşlemleri

Sigorta İşlemleri

İstatistik

Ara :

TC	Ad	Soyad
12345698745	Osman	Balci
12395478569	Emel	Şahin
14725836985	Muarrem	Ekinci
11111111111	Zuhuri	Altun
41125555555	Kübra	Yön

TC No : 11

ID	TC	Ad	Soyad	Cinsiyet	Doğum Tarihi
9	11111111111	Zuhuri	Altun	0	24.5.2021
10	41125555555	Kübra	Yön	1	26.3.1998

Ad :

Soyad :

Cinsiyet : Kadın

Doğum Tarihi : 30 Mayıs 2021 Pazar

Mail :

Tel No :

Ehliyet No :

Ehliyet Tür : M

Adres :

Bilgileri Güncelle

### Müşteri Güncelle:

Müşteri güncelle ekranında sistemde kayıtlı olan müşterilerin çeşitli bilgileri güncellenebilir. T.C. kimlik numarası TextBox nesnesine yazıldığı zaman aşağısında bulunan DataGridView

nesnesine, girilen sayıyı içerisinde barındıran T.C. kimlik numaralı müşterileri listeler. Listedeki güncelleme yapılacak müşteri seçildiği zaman aşağıda bulunan ComboBox, TextBox, RichTextBox ve DateTimePicker nesneleri otomatik olarak seçilen müşterinin bilgileriyle dolar. Sonrasında değiştirilmek istenen bilgiler düzenlenir ve Güncelle butonuna basılarak güncelleme işlemi tamamlanır. Bir hata ile karşılaşılmadığı durumda işlemin tamamlandığına dair bir mesaj gösterilir.

TC	Ad	Soyad
12345698745	Osman	Balci
12395478569	Emel	Şahin
14725836985	Muarrem	Ekinci
11111111111	Zuhuri	Altun
41125555555	Kübra	Yön

### Müşteri Sil:

Sistemde kayıtlı olan müşterilerden biri silinmek istenir ise Müşteri İşlemleri ekranındaki Müşteri Sil sekmesinden silme işlemi yapılabilir. Bunun için ComboBox nesnesinden silinmek istenen müşterinin T.C. kimlik numarasına göre seçilmesi ve sonrasında Sil butonuna basılması yeterlidir. Silinmek istenen müşteri müşteriTc alanına bakılarak bulunur ve sistemden silinir. Eğer bir hata ile karşılaşılmaz ise müşterinin silindiğine dair bir onay mesajı gösterilir.

### 2.7.4. Sigorta İşlemleri Ekranı

Sigorta İşlemleri ekranında sigortalar ile ilgili yapılacak işlemler bulunur. Bu işlemler sırasıyla;

- Sigortayı Kaydet
- Sigorta Bilgilerini Güncelle
- Sigortayı Sil işlemleridir.

Araç Kiralama Sistemi Yönetim Paneli

Yönetim Paneli

- Ana Sayfa
- Araç İşlemleri
- Müşteri İşlemleri
- Sigorta İşlemleri
- İstatistik

ID	SigortaAdı	Ücret	Açıklama
1	Temel Sigorta	450	Temel araç kasko ve sigortalarını içerir. Kullanıcı hatası olmayan durumlar için geçerlidir.
2	Full Paket Sigorta	12000	Araç her türlü soruna karşı korumaya alır. Kullanıcı hatası olan kaza durumları, araca olası bir zarar gelmesi veya aracın çalınması gibi durumlar için geçerlidir.
3	Mini Hasar Sigortası	2200	Temel basit araç hasarları gibi durumlarda tek seferlik geçerlidir. Olası bütün temel hasarları kapsar. Ek bir ücrete gerek olmadan araç sizden

Sigorta İşlemleri

Sigorta Adı :  Sigorta Ücreti :  Sigorta Adı : Temel Sigorta

Sigorta Açıklaması :

Sigortayı Sil

Sigortayı Kaydet Sigorta Bilgilerini Güncelle

### Sigortayı Kaydet:

Sigorta İşlemleri sayfasında TextBox ve RichTextBox nesneleri kullanılarak sisteme yeni bir sigorta eklenebilir. Öncelikle gerekli bilgiler doldurulur ve Sigortayı Kaydet butonuna basarak kayıt işlemi yapılır. Eğer herhangi bir sorun yok ise sigorta başarıyla sisteme kayıt edilir ve bir mesaj ile kaydın tamamlandığı bildirilir.

### Sigorta Bilgilerini Güncelle:

Sigorta İşlemleri ekranında sistemde kayıtlı olan sigortaların çeşitli bilgileri güncellenebilir. Sayfanın üst kısmında bulunan DataGridView nesnesinden güncelleme yapılacak sigorta seçildiği zaman aşağıda bulunan TextBox ve RichTextBox nesneleri otomatik olarak seçilen sigortanın bilgileriyle dolar. Sonrasında değiştirilmek istenen bilgiler düzenlenir ve Sigorta Bilgilerini Güncelle butonuna basılarak güncelleme işlemi tamamlanır. Bir hata ile karşılaşmadığı durumda işlemin tamamlandığına dair bir mesaj gösterilir.

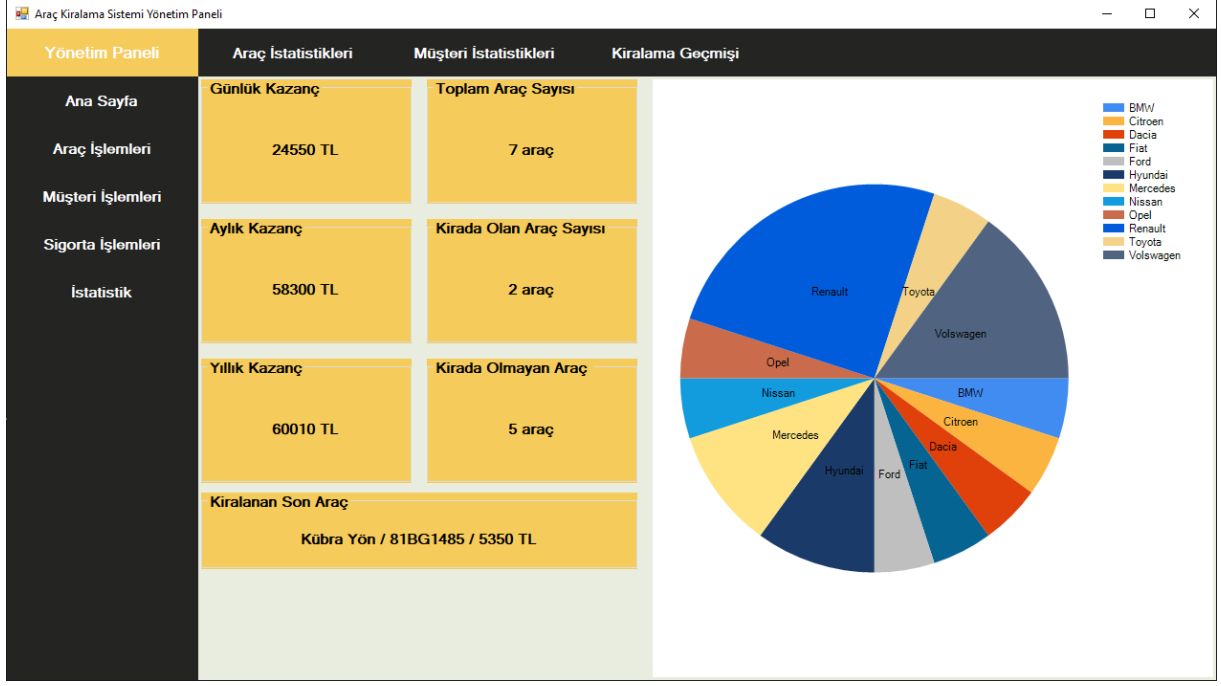
### Sigortayı Sil:

Sigorta İşlemleri ekranında bulunan ComboBox ile herhangi bir sigorta sistemden silinmek istenir ise rahatlıkla silinebilir. Bunun için silinmek istenen sigortanın Combobox nesnesinden seçilmesi ve ardından Sigortayı Sil butonuna basılması yeterlidir.

### 2.7.5. İstatistik Ekranı

İstatistik ekranında işletmenin toplam araç sayısı aylık kazancı gibi çeşitli istatistikler bulunur. İstatistik sayfası kendi içerisinde 3 bölüme ayrılır bunlar;

- Araç İstatistikleri
- Müşteri İstatistikleri
- Kiralama Geçmişi bölümleridir.

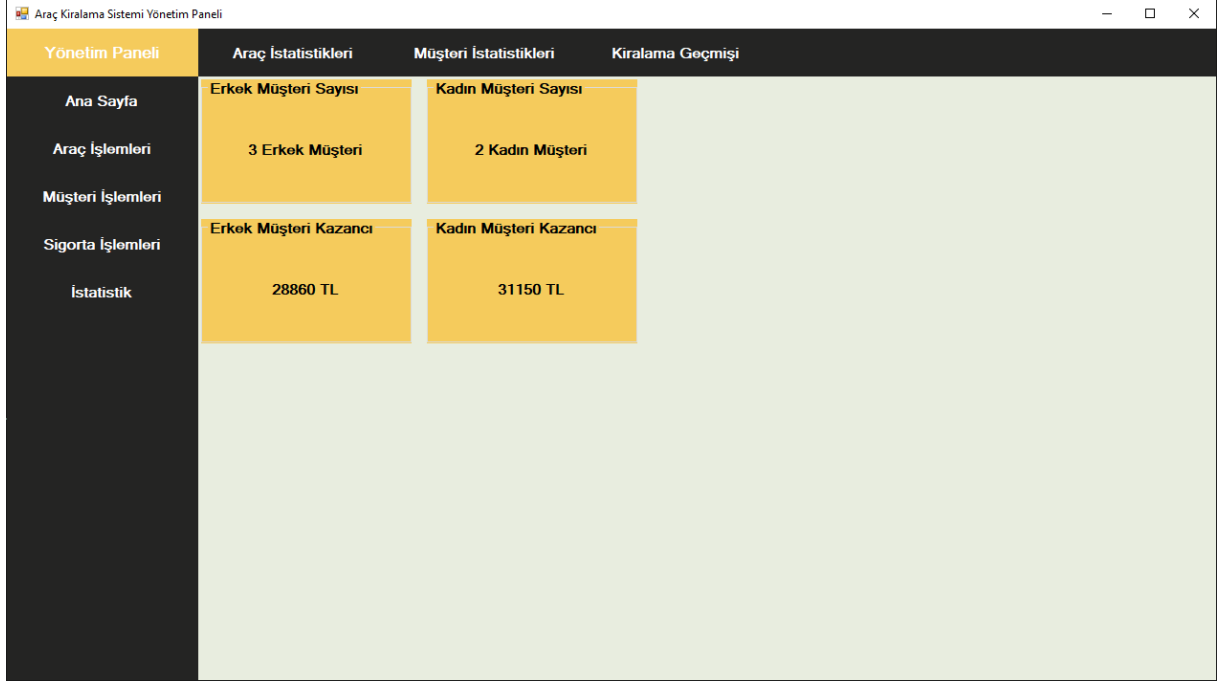


### Araç İstatistikleri:

Araç istatistikleri sayfasında araçlar ile ilgili çeşitli istatistikler bulunur. Bunlar sırasıyla:

- Günlük Kazanç,
- Aylık Kazanç,
- Yıllık Kazanç,
- Toplam Araç Sayısı,
- Kirada Olan Araç Sayısı,
- Kirada Olmayan Araç Sayısı,
- Kiralanan Son Araç
- Araçların Markalarına Göre Oranları olarak sıralanabilir.

Burada bulunan bilgiler sayesinde çeşitli çıkarımlar yapılabilir ve bu çıkarımlara göre çeşitli düzenlemeler yapılabilir.



### Müşteri İstatistikleri:

Müşteri istatistikleri sayfasında müşteriler ile ilgili çeşitli istatistikler bulunur. Bunlar sırasıyla:

- Erkek Müşteri Sayısı
- Erkek Müşteri Kazancı
- Kadın Müşteri Sayısı
- Kadın Müşteri Kazancı olarak sıralanabilir.

Burada bulunan bilgiler sayesinde çeşitli çıkarımlar yapılabilir ve bu çıkarımlara göre çeşitli düzenlemeler yapılabilir.

**Araç Kiralama Sistemi Yönetim Paneli**

Yönetim Paneli	Araç İstatistikleri	Müşteri İstatistikleri	Kiralama Geçmişi
Ana Sayfa			
Araç İşlemleri			
Müşteri İşlemleri			
Sigorta İşlemleri			
İstatistik			

Araç	Müşteri	Sigorta	TOPLAMTUTAR	BaşlangıçTarihi	BitişTarihi	İl
Symbol Sedan 06AZ1234	Osman Balcı	Temel Sigorta 450 TL	1150 TL	23.5.2021	30.5.2021	Adana
Polo Hatchback 01AZ1234	Emel Şahin	Mini Hasar Sigortası 2200 TL	4000 TL	23.5.2021	1.6.2021	İstanbul
Egea Sedan 06CZ2514	Muarrem Ekinci	Temel Sigorta 450 TL	1710 TL	7.4.2021	14.4.2021	Aydın
Symbol Sedan 06AZ1234	Zuhuri Altun	Full Paket Sigorta 12000 TL	12100 TL	30.5.2021	31.5.2021	Ankara
C180 Sedan 81BG1485	Zuhuri Altun	Full Paket Sigorta 12000 TL	12450 TL	30.5.2021	31.5.2021	Ankara
5.20i Sedan 06BG1485	Emel Şahin	Full Paket Sigorta 12000 TL	19000 TL	28.5.2021	11.6.2021	Balıkesir
Symbol Sedan 06AZ1234	Osman Balcı	Temel Sigorta 450 TL	550 TL	27.5.2021	28.5.2021	Adana
Polo Hatchback 01AZ1234	Kübra Yön	Mini Hasar Sigortası 2200 TL	2800 TL	27.5.2021	30.5.2021	Adana
Focus Sedan 01DC1632	Zuhuri Altun	Temel Sigorta 450 TL	900 TL	28.5.2021	31.5.2021	Artvin
C180 Sedan 81BG1485	Kübra Yön	Mini Hasar Sigortası 2200 TL	5350 TL	8.6.2021	15.6.2021	Çorum



## Kiralama Geçmişi:

Kiralama Geçmişi sayfasında şimdiye kadar yapılmış olan bütün kiralama işlemlerinin kaydını görebiliriz. Bu kayıtlar sırasıyla;

- Araç,
- Müşteri,
- Sigorta,
- Toplam Tutar,
- Başlangıç Tarihi,
- Bitiş Tarihi,
- İl gibi birkaç temel bilgiye sahiptir.

## 2.8. Projenin Kodları

### 2.8.1. Başlangıç Formu 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 arackiralama
{
    public partial class Anasayfa : Form
    {
        public Anasayfa()
        {
            InitializeComponent();
        }

        public void showControl(Control control)
        {
            Content.Controls.Clear();
            control.Dock = DockStyle.Fill;
            control.BringToFront();
            control.Focus();
            Content.Controls.Add(control);
        }
    }
}
```

```
}
```

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

```
{
```

```
    home hm = new home();
```

```
    showControl(hm);
```

```
}
```

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

```
{
```

```
    cars cr = new cars();
```

```
    showControl(cr);
```

```
}
```

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

```
{
```

```
    users us = new users();
```

```
    showControl(us);
```

```
}
```

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

```
{
```

```
    istatistik ist = new istatistik();
```

```
    showControl(ist);
```

```
}
```

```
private void Form1_Load(object sender, EventArgs e)
```

```
{
```

```
    home hm = new home();
```

```
    showControl(hm);
```

```
}
```

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

```
{
```

```
    sigorta si = new sigorta();
```

```

        showControl(si);
    }
}

```

## 2.8.2. Ana Sayfa UserControl Kodları

```

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

namespace arackiralama
{
    public partial class home : UserControl
    {
        public home()
        {
            InitializeComponent();
            homePage hp = new homePage();
            showControl(hp);
        }

        public void showControl(Control control)
        {
            home_Content.Controls.Clear();
            control.Dock = DockStyle.Fill;
            control.BringToFront();
            control.Focus();
            home_Content.Controls.Add(control);
        }

        vtClass vt = new vtClass(false);
        private void btn_home_Click(object sender, EventArgs e)

```

```

{
    homePage hp = new homePage();
    showControl(hp);
}

private void btn_carsUsing_Click(object sender, EventArgs e)
{
    fullList fL = new
fullList(vt.verial("araclar.aracPlaka,modeller.modelAd,araclar.aracYil,yakıttip.yakıttipAd,vite
s.vitesAd,araclar.aracKm,cekis.cekisAd,araclar.aracKisi,renk.renkAd,araclar.aracKlima,iller.i
lAd,araclar.aracKiraUcret", "araclar", "INNER JOIN modeller ON
modeller.modelID=araclar.modelID INNER JOIN yakıttip ON
yakıttip.yakıttipID=araclar.yakıttipID INNER JOIN vites ON vites.vitesID=araclar.vitesID
INNER JOIN cekis ON cekis.cekisID=araclar.cekisID INNER JOIN renk ON
renk.renkID=araclar.renkID INNER JOIN iller ON
iller.ilID=araclar.ilID", "aracKiraDurum='1'"));
    showControl(fL);
}

private void btn_carsGive_Click(object sender, EventArgs e)
{
    aracKiral aK = new aracKiral();
    showControl(aK);
}

private void btn_araclarList_Click(object sender, EventArgs e)
{
    fullList fL = new
fullList(vt.verial("araclar.aracPlaka,modeller.modelAd,araclar.aracYil,yakıttip.yakıttipAd,vite
s.vitesAd,araclar.aracKm,cekis.cekisAd,araclar.aracKisi,renk.renkAd,araclar.aracKlima,iller.i
lAd,araclar.aracKiraUcret", "araclar", "INNER JOIN modeller ON
modeller.modelID=araclar.modelID INNER JOIN yakıttip ON
yakıttip.yakıttipID=araclar.yakıttipID INNER JOIN vites ON vites.vitesID=araclar.vitesID
INNER JOIN cekis ON cekis.cekisID=araclar.cekisID INNER JOIN renk ON
renk.renkID=araclar.renkID INNER JOIN iller ON iller.ilID=araclar.ilID", "0"));
    showControl(fL);
}

private void btnn_musteriList_Click(object sender, EventArgs e)
{

```

```

        fullList = new
fullList(vt.verial("musteriAd,musteriSoyad,musteriCinsiyet,musteriDogumTarihi,musteriAdr
es,musteriMail,musteriPhone,musteriEhliyetNo,musteriEhliyetTur", "musteriler", "0", "0"));
        showControl(fL);
    }
}
}

```

### 2.8.3. HomePage UserControl Kodları

```

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

namespace arackiralama
{
    public partial class homePage : UserControl
    {
        public homePage()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        private void timer1_Tick(object sender, EventArgs e)
        {
            lbl_time.Text = "Sistem Zamanı : " + DateTime.Now;
            try
            {
                lbl_carsCount.Text = vt.verial("Count(aracID)", "araclar", "0",
"0").Rows[0][0].ToString() + " araç";
            }
            catch { }
        }
    }
}

```

```

        lbl_usersCount.Text = vt.verial("Count(musteriID)", "musteriler", "0",
"0").Rows[0][0].ToString() + " müşteri";

        lbl_moneyCount.Text = vt.verial("SUM(kiraUcret)", "kiralama", "0",
"kiraGirisTarih >= " + (DateTime.Now.Year.ToString() + "-" +
DateTime.Now.Month.ToString() + "-1") + "").Rows[0][0].ToString() + " TL";
    }
    catch (Exception h)
    {
        if (vt.HataYakala == true)
        {
            MessageBox.Show(h.Message);
        }
    }
}

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

```

#### 2.8.4. AraçKiralama UserControl Kodları

```

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

namespace arackiralama
{

```

```

public partial class aracKiralda : UserControl
{
    public aracKiralda()
    {
        InitializeComponent();
    }

    vtClass vt = new vtClass(false);
    public double total=0;
    public void aracYenile()
    {
        cb_cars.DataSource = vt.verial("araclar.aracID,(modeller.modelAd +' '+
araclar.aracPlaka) as arac", "araclar", "INNER JOIN modeller ON modeller.modelID =
araclar.modelID", "aracKiraDurum='0'");
        cb_cars.ValueMember = "aracID";
        cb_cars.DisplayMember = "arac";

        cb_carsKirada.DataSource = vt.verial("araclar.aracID,(modeller.modelAd +' '+
araclar.aracPlaka) as arac", "araclar", "INNER JOIN modeller ON modeller.modelID =
araclar.modelID", "aracKiraDurum='1'");
        cb_carsKirada.ValueMember = "aracID";
        cb_carsKirada.DisplayMember = "arac";
    }
    private void aracKiralda_Load(object sender, EventArgs e)
    {
        cb_musteri.DataSource = vt.verial("?", "musteriler", "0", "0");
        cb_musteri.ValueMember = "musteriID";
        cb_musteri.DisplayMember = "musteriTc";

        aracYenile();

        cb_sigorta.DataSource = vt.verial("?", "sigortalar", "0", "0");
        cb_sigorta.ValueMember = "sigortaID";
        cb_sigorta.DisplayMember = "sigortaAd";

        cb_iller.DataSource = vt.verial("?", "iller", "0", "0");
        cb_iller.ValueMember = "ilID";
    }
}

```

```

        cb_iller.DisplayMember = "ilAd";
    }

    private void btn_kiraAdd_Click(object sender, EventArgs e)
    {
        TimeSpan Sonuc = Convert.ToDateTime(dt_end.Text) -
        Convert.ToDateTime(dt_start.Text);
        double kiragun = Sonuc.TotalDays;
        if (kiragun <= 30 && kiragun > 0)
        {
            if (vt.veriekle("kiralama", cb_musteri.SelectedValue, cb_cars.SelectedValue,
            cb_sigorta.SelectedValue, int.Parse(total.ToString()), dt_start.Value.Date.ToString("yyyy-
            MM-dd"), dt_end.Value.Date.ToString("yyyy-MM-dd"), cb_iller.SelectedValue,
            rtb_aracAdres.Text))
            {
                vt.sorguGonder("UPDATE aracler SET aracKiraDurum = '1' Where aracID =
                '"+cb_cars.SelectedValue+"'");
                aracYenile();
                MessageBox.Show("Kiralama Kaydı Başarılı.");
            }
            else
            {
                MessageBox.Show("Kiralama Kaydı Başarısız!");
            }
        }
        else
        {
            MessageBox.Show("30 günden fazla veya 0 günden az sürede kiralama yapılamaz!");
        }
    }

    private void toplamTutar(object sender, EventArgs e)
    {
        try
        {
            if (cb_cars.Items.Count > 0)
            {

```



```

        double aracUcret = Convert.ToDouble((vt.verial("aracKiraUcret", "araclar", "0",
("aracID LIKE " + cb_cars.SelectedValue)).Rows[0][0]).ToString());

        double sigortaUcret = Convert.ToDouble((vt.verial("sigortaUcret", "sigortalar",
"0", ("sigortaID LIKE " + cb_sigorta.SelectedValue)).Rows[0][0]).ToString());

        lbl_aracUcret.Text= aracUcret + " TL";
        lbl_sigortaUcret.Text = sigortaUcret + " TL";

        TimeSpan Sonuc = Convert.ToDateTime(dt_end.Text) -
Convert.ToDateTime(dt_start.Text);

        double kiragun = Sonuc.TotalDays;
        lbl_kiraGun.Text = kiragun + " Gün";
        total = ((kiragun * aracUcret) + sigortaUcret);
    }
    lbl_toplamTutar.Text = "Toplam Tutar: " + total + " TL";
}
catch (Exception h)
{
    if (vt.HataYakala == true)
    {
        MessageBox.Show(h.Message);
    }
}
}

private void btn_kiraOut_Click(object sender, EventArgs e)
{
    vt.sorguGonder("UPDATE araclar SET aracKiraDurum = '0' Where aracID = '" +
cb_carsKirada.SelectedValue + "'");
    aracYenile();
}
}
}

```

### 2.8.5. Araç İşlemleri UserControl Kodları

```

using System;
using System.Collections.Generic;
using System.ComponentModel;

```

```

using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace arackiralama
{
    public partial class cars : UserControl
    {
        public cars()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        public void showControl(Control control)
        {
            cars_Content.Controls.Clear();
            control.Dock = DockStyle.Fill;
            control.BringToFront();
            control.Focus();
            cars_Content.Controls.Add(control);
        }

        private void btn_carsList_Click(object sender, EventArgs e)
        {
            data_cars.DataSource= vt.verial("araclar.aracPlaka as Plaka,markalar.markaAd as
Marka, modeller.modelAd as Model", "araclar", "INNER JOIN modeller ON araclar.modelID
=          modeller.modelID          INNER          JOIN          markalar          ON
modeller.modelMarka=markalar.markaID", "0");
        }

        private void btn_carsAdd_Click(object sender, EventArgs e)
        {
            carsAdd caA = new carsAdd();

```

```

        showControl(caA);
    }

    private void btn_carsUpdate_Click(object sender, EventArgs e)
    {
        carsUpdate caU = new carsUpdate();
        showControl(caU);
    }

    private void btn_carsDelete_Click(object sender, EventArgs e)
    {
        carsDelete caD = new carsDelete();
        showControl(caD);
    }

    private void btn_carsExtra_Click(object sender, EventArgs e)
    {
        carsExtra caE = new carsExtra();
        showControl(caE);
    }

    private void tb_carSearch_TextChanged(object sender, EventArgs e)
    {
        data_cars.DataSource = vt.verial("araclar.aracPlaka as Plaka,markalar.markaAd as
Marka, modeller.modelAd as Model","araclar", "INNER JOIN modeller ON araclar.modelID
=          modeller.modelID          INNER          JOIN          markalar          ON
modeller.modelMarka=markalar.markaID",("araclar.aracPlaka          LIKE
'%" +tb_carSearch.Text+"%'"));
    }
}

```

### 2.8.6. Yeni Araç Ekle UserControl Kodları

```

using System;
using System.Collections.Generic;
using System.ComponentModel;

```

```

using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace arackiralama
{
    public partial class carsAdd : UserControl
    {
        public carsAdd()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        private void carsAdd_Load(object sender, EventArgs e)
        {
            cb_marka.DataSource = vt.verial("?", "markalar", "", "markaID IN (Select modelMarka
from modeller) ORDER BY markaAd");
            cb_marka.ValueMember = "markaID";
            cb_marka.DisplayMember = "markaAd";

            cb_model.DataSource = vt.verial("?", "modeller", "0", ("modelMarka LIKE " +
cb_marka.SelectedValue.ToString() + "" + " ORDER BY modelAd"));
            cb_model.ValueMember = "modelID";
            cb_model.DisplayMember = "modelAd";

            cb_aracYakitTip.DataSource=vt.verial("?", "yakittip", "0", "0");
            cb_aracYakitTip.ValueMember = "yakittipID";
            cb_aracYakitTip.DisplayMember = "yakittipAd";

            cb_AracVitesTip.DataSource = vt.verial("?", "vites", "0", "0");
            cb_AracVitesTip.ValueMember = "vitesID";
            cb_AracVitesTip.DisplayMember = "vitesAd";
        }
    }
}

```

```

cb_aracCekisTip.DataSource = vt.verial("*", "cekis", "0", "0");
cb_aracCekisTip.ValueMember = "cekisID";
cb_aracCekisTip.DisplayMember = "cekisAd";

cb_aracRenk.DataSource = vt.verial("*", "renk", "0", "0");
cb_aracRenk.ValueMember = "renkID";
cb_aracRenk.DisplayMember = "renkAd";

cb_AracKlima.DisplayMember = "Text";
cb_AracKlima.ValueMember = "Value";
var items = new[] {
    new { Text = "var", Value = "1" },
    new { Text = "yok", Value = "0" }
};
cb_AracKlima.DataSource = items;

cb_il.DataSource = vt.verial("*", "iller", "0", "0");
cb_il.ValueMember = "ilID";
cb_il.DisplayMember = "ilAd";
}

private void btn_userAdd_Click(object sender, EventArgs e)
{
    if (tb_aracPlaka.TextLength<8)
    {
        MessageBox.Show("Araç Plakası en az 8 hane olmalıdır!");
    }
    else
    {
        if (int.Parse(vt.verial("Count(aracPlaka)", "araclar", "0", "aracPlaka =" +
tb_aracPlaka.Text + "").Rows[0][0].ToString()) > 0)
        {
            MessageBox.Show("Bu araç zaten sisteme kayıtlı!");
        }
        else

```

```

        {
            if (vt.veriekle("araclar", tb_aracPlaka.Text.ToString(), cb_model.SelectedValue,
tb_aracYil.Text.ToString(), cb_aracYakitTip.SelectedValue, cb_AracVitesTip.SelectedValue,
tb_aracKm.Text.ToString(), cb_aracCekisTip.SelectedValue, tb_aracKisi.Text.ToString(),
cb_aracRenk.SelectedValue, cb_AracKlima.SelectedValue.ToString(), cb_il.SelectedValue,
rtb_aracAdres.Text.ToString(), tb_aracUcret.Text.ToString(), 0))
            {
                MessageBox.Show("Kayıt Başarılı");
            }
            else
            {
                MessageBox.Show("Kayıt Yapılamadı");
            }
        }
    }
}

private void cb_marka_SelectedIndexChanged(object sender, EventArgs e)
{
    cb_model.DataSource = vt.verial("?", "modeller", "0", ("modelMarka LIKE " +
cb_marka.SelectedValue.ToString() + ""));
    cb_model.ValueMember = "modelID";
    cb_model.DisplayMember = "modelAd";
}
}
}

```

### 2.8.7. Araç Bilgisi Güncelle UserControl Kodları

```

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

```

```

namespace arackiralama
{
    public partial class carsUpdate : UserControl
    {
        public carsUpdate()
        {
            InitializeComponent();
        }

        vtClass vt = new vtClass(false);

        private void btn_userUpdate_Click(object sender, EventArgs e)
        {
            vt.kayitGuncelle("araclar", "aracPlaka", tb_aracPlaka.Text.ToString(), tb_aracPlaka.Text.ToString(),
            cb_model.SelectedValue.ToString(), tb_aracYil.Text.ToString(), cb_aracYakitTip.SelectedValue.ToString(),
            cb_AracVitesTip.SelectedValue.ToString(), tb_aracKm.Text.ToString(), cb_aracCekisTip.SelectedValue.ToString(),
            tb_aracKisi.Text.ToString(), cb_aracRenk.SelectedValue.ToString(), cb_AracKlima.SelectedValue.ToString(),
            cb_il.SelectedValue.ToString(), rtb_aracAdres.Text.ToString(), tb_aracKira.Text.ToString(), 0);
        }

        private void carsUpdate_Load(object sender, EventArgs e)
        {
            cb_marka.DataSource = vt.verial("?", "markalar", "0", "0");
            cb_marka.ValueMember = "markaID";
            cb_marka.DisplayMember = "markaAd";

            cb_model.DataSource = vt.verial("?", "modeller", "0", ("modelMarka LIKE " +
            cb_marka.SelectedValue.ToString() + ""));
            cb_model.ValueMember = "modelID";
            cb_model.DisplayMember = "modelAd";

            cb_aracYakitTip.DataSource = vt.verial("?", "yakittip", "0", "0");
            cb_aracYakitTip.ValueMember = "yakittipID";
            cb_aracYakitTip.DisplayMember = "yakittipAd";
        }
    }
}

```

```

cb_AracVitesTip.DataSource = vt.verial("*", "vites", "0", "0");
cb_AracVitesTip.ValueMember = "vitesID";
cb_AracVitesTip.DisplayMember = "vitesAd";

cb_aracCekisTip.DataSource = vt.verial("*", "cekis", "0", "0");
cb_aracCekisTip.ValueMember = "cekisID";
cb_aracCekisTip.DisplayMember = "cekisAd";

cb_aracRenk.DataSource = vt.verial("*", "renk", "0", "0");
cb_aracRenk.ValueMember = "renkID";
cb_aracRenk.DisplayMember = "renkAd";

cb_AracKlima.DisplayMember = "Text";
cb_AracKlima.ValueMember = "Value";
var items = new[] {
    new { Text = "var", Value = "1" },
    new { Text = "yok", Value = "0" }
};
cb_AracKlima.DataSource = items;

cb_il.DataSource = vt.verial("*", "iller", "0", "0");
cb_il.ValueMember = "ilID";
cb_il.DisplayMember = "ilAd";
}

private void cb_marka_SelectedIndexChanged(object sender, EventArgs e)
{
    cb_model.DataSource = vt.verial("*", "modeller", "0", ("modelMarka LIKE " +
cb_marka.SelectedValue.ToString() + ""));
    cb_model.ValueMember = "modelID";
    cb_model.DisplayMember = "modelAd";
}

private void tb_aracPlaka_TextChanged(object sender, EventArgs e)
{

```



```

        dataG_car.DataSource =
vt.verial("araclar.aracPlaka,markalar.markaAd,modeller.modelAd,araclar.aracYil,yakittip.yak
ittipAd,vites.vitesAd,araclar.aracKm,cekis.cekisAd,araclar.aracKisi,renk.renkAd,araclar.arac
Klima,iller.ilAd,araclar.aracKiraUcret,araclar.aracAdres", "araclar", "INNER JOIN modeller
ON modeller.modelID=araclar.modelID INNER JOIN markalar ON markalar.markaID =
modeller.modelMarka INNER JOIN yakittip ON yakittip.yakittipID=araclar.yakittipID
INNER JOIN vites ON vites.vitesID=araclar.vitesID INNER JOIN cekis ON
cekis.cekisID=araclar.cekisID INNER JOIN renk ON renk.renkID=araclar.renkID INNER
JOIN iller ON iller.ilID=araclar.ilID", ("aracPlaka LIKE '%" + tb_aracPlaka.Text.ToString() +
"%"));
    }

```

```

private void dataG_car_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    tb_aracPlaka.Text = dataG_car.CurrentRow.Cells[0].Value.ToString();
    cb_marka.Text = dataG_car.CurrentRow.Cells[1].Value.ToString();
    cb_model.Text = dataG_car.CurrentRow.Cells[2].Value.ToString();
    tb_aracYil.Text = dataG_car.CurrentRow.Cells[3].Value.ToString();
    cb_aracYakitTip.Text = dataG_car.CurrentRow.Cells[4].Value.ToString();
    cb_AracVitesTip.Text = dataG_car.CurrentRow.Cells[5].Value.ToString();
    tb_aracKm.Text = dataG_car.CurrentRow.Cells[6].Value.ToString();
    cb_aracCekisTip.Text = dataG_car.CurrentRow.Cells[7].Value.ToString();
    tb_aracKisi.Text = dataG_car.CurrentRow.Cells[8].Value.ToString();
    cb_aracRenk.Text = dataG_car.CurrentRow.Cells[9].Value.ToString();

    if (int.Parse(dataG_car.CurrentRow.Cells[10].Value.ToString())==1)
    { cb_AracKlima.SelectedIndex = 1;}
    else {cb_AracKlima.SelectedIndex = 1;}

    cb_il.Text = dataG_car.CurrentRow.Cells[11].Value.ToString();
    tb_aracKira.Text = dataG_car.CurrentRow.Cells[12].Value.ToString();
    rtb_aracAdres.Text = dataG_car.CurrentRow.Cells[13].Value.ToString();
}
}
}

```

## 2.8.8. Araç Silme UserControl Kodları

```
using System;
```

```

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

namespace arackiralama
{
    public partial class carsDelete : UserControl
    {
        public carsDelete()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        private void carsDelete_Load(object sender, EventArgs e)
        {
            cb_aracPlaka.DataSource = vt.verial("*", "araclar", "0", "0");
            cb_aracPlaka.ValueMember = "aracID";
            cb_aracPlaka.DisplayMember = "aracPlaka";
        }

        private void btn_carFind_Click(object sender, EventArgs e)
        {
            vt.verisil("araclar", "aracPlaka", cb_aracPlaka.SelectedText.ToString());
        }
    }
}

```

### 2.8.9. Marka-Model Ekle UserControl Kodları

```

using System;
using System.Collections.Generic;

```

```

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

namespace arackiralama
{
    public partial class carsExtra : UserControl
    {
        public carsExtra()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        private void carsExtra_Load(object sender, EventArgs e)
        {
            dataG_marka.DataSource = vt.verial("markaAd as Marka","markalar", "0", "0");
            dataG_model.DataSource = vt.verial("modeller.modelAd as Model,markalar.markaAd
as Marka","modeller", "Inner Join markalar On markalar.markaID=modeller.modelMarka
ORDER BY markalar.markaAd", "0");

            cb_marka.DataSource = vt.verial("","markalar", "0", "0");
            cb_marka.ValueMember = "markaID";
            cb_marka.DisplayMember = "markaAd";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (vt.veriekle("markalar",tb_marka.Text.ToString()))
            {
                MessageBox.Show("Kayıt Başarılı");
            }
            else

```

```

        {
            MessageBox.Show("Kayıt Yapılamadı");
        }
    }

private void button2_Click(object sender, EventArgs e)
{
    if (vt.veriekle("modeller", tb_model.Text.ToString(), cb_marka.SelectedValue))
    {
        MessageBox.Show("Kayıt Başarılı");
    }
    else
    {
        MessageBox.Show("Kayıt Yapılamadı");
    }
}
}
}

```

#### 2.8.10. Müşteri İşlemleri UserControl Kodları

```

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

namespace arackiralama
{

```

```

public partial class users : UserControl
{
    public users()
    {
        InitializeComponent();
    }
    vtClass vt = new vtClass(false);
    public void showControl(Control control)
    {
        users_Content.Controls.Clear();
        control.Dock = DockStyle.Fill;
        control.BringToFront();
        control.Focus();
        users_Content.Controls.Add(control);
    }

    private void btn_usersList_Click(object sender, EventArgs e)
    {
        data_users.DataSource = vt.verial("musteriTc as TC,musteriAd as Ad,musteriSoyad as
Soyad", "musteriler", "0", "0");
    }

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

    private void btn_usersAdd_Click(object sender, EventArgs e)
    {
        userAdd uA = new userAdd();
        showControl(uA);
    }

    private void btn_usersUpdate_Click(object sender, EventArgs e)
    {

```

```

        userUpdate uU = new userUpdate();
        showControl(uU);
    }

    private void btn_usersDelete_Click(object sender, EventArgs e)
    {
        userDelete uD = new userDelete();
        showControl(uD);
    }

    private void tb_userSearch_TextChanged(object sender, EventArgs e)
    {
        data_users.DataSource = vt.verial("musteriTc as TC,musteriAd as Ad,musteriSoyad as
Soyad", "musteriler", "0", "musteriTc LIKE '%" + tb_userSearch.Text + "%'");
    }
}

```

#### 2.8.11. Müşteri Kayıt UserControl Kodları

```

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

namespace arackiralama
{
    public partial class userAdd : UserControl
    {
        public userAdd()
        {

```

```

        InitializeComponent();
    }
    vtClass vt = new vtClass(false);
    private void userAdd_Load(object sender, EventArgs e)
    {
        cb_cinsiyet.DisplayMember = "Text";
        cb_cinsiyet.ValueMember = "Value";
        var items = new[] {
            new { Text = "Kadın", Value = "1" },
            new { Text = "Erkek", Value = "0" }
        };
        cb_cinsiyet.DataSource = items;

        cb_ehliyettur.DisplayMember = "Text";
        cb_ehliyettur.ValueMember = "Value";
        var items2 = new[] {
            new { Text = "M", Value = "0" },
            new { Text = "A1", Value = "1" },
            new { Text = "A2", Value = "2" },
            new { Text = "A", Value = "3" },
            new { Text = "B1", Value = "4" },
            new { Text = "B", Value = "5" },
            new { Text = "BE", Value = "6" },
            new { Text = "C1", Value = "7" },
            new { Text = "C", Value = "8" },
            new { Text = "CE", Value = "9" },
            new { Text = "D1", Value = "10" },
            new { Text = "D", Value = "11" },
            new { Text = "DE", Value = "12" },
            new { Text = "F", Value = "13" },
            new { Text = "G", Value = "14" }
        };
        cb_ehliyettur.DataSource = items2;
    }

```

```

private void btn_userAdd_Click(object sender, EventArgs e)
{
    if (tb_tc.TextLength<11)
    {
        MessageBox.Show("TC kimlik numarası 11 haneden az olamaz");
    }
    else
    {
        if (int.Parse(vt.verial("Count(musteriTC)", "musteriler", "0", "musteriTC =" +
tb_tc.Text + "").Rows[0][0].ToString()) > 0)
        {
            MessageBox.Show("Bu müşteri sisteme kayıtlı!");
        }
        else
        {
            if (vt.veriekle("musteriler", tb_tc.Text, tb_ad.Text, tb_soyad.Text,
cb_cinsiyet.SelectedValue, dateTimePicker1.Value.Date.ToString("yyyy-MM-
dd").ToString(), richTextBox1.Text, tb_mail.Text, tb_telno.Text, tb_ehliyetno.Text,
cb_ehliyettur.Text))
            {
                MessageBox.Show("Müşteri Kaydı Başarılı.");
            }
            else
            {
                MessageBox.Show("Kayıt Yapılamadı!");
            }
        }
    }
}
}

```

## 2.8.12. Müşteri Güncelle UserControl Kodları

```

using System;
using System.Collections.Generic;
using System.ComponentModel;

```



```

using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace arackiralama
{
    public partial class userUpdate : UserControl
    {
        public userUpdate()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        private void tb_tc_TextChanged(object sender, EventArgs e)
        {
            dataG_user.DataSource = vt.verial("musteriID as ID, musteriTC as TC, musteriAd as
Ad, musteriSoyad as Soyad, musteriCinsiyet as Cinsiyet, musteriDogumTarihi as
DoğumTarihi, musteriAdres as Adres, musteriMail as Mail, musteriPhone as Tel,
musteriEhliyetNo as EhliyetNo, musteriEhliyetTur as EhliyetTürü","musteriler","0",
("musteriTC LIKE '%" + tb_tc.Text.ToString() + '%""));
        }

        private void userUpdate_Load(object sender, EventArgs e)
        {
            cb_cinsiyet.DisplayMember = "Text";
            cb_cinsiyet.ValueMember = "Value";
            var items = new[] {
                new { Text = "Kadın", Value = "1" },
                new { Text = "Erkek", Value = "0" }
            };
            cb_cinsiyet.DataSource = items;

            cb_ehliyettur.DisplayMember = "Text";

```

```

cb_ehliyettur.ValueMember = "Value";
var items2 = new[] {
    new { Text = "M", Value = "0" },
    new { Text = "A1", Value = "1" },
    new { Text = "A2", Value = "2" },
    new { Text = "A", Value = "3" },
    new { Text = "B1", Value = "4" },
    new { Text = "B", Value = "5" },
    new { Text = "BE", Value = "6" },
    new { Text = "C1", Value = "7" },
    new { Text = "C", Value = "8" },
    new { Text = "CE", Value = "9" },
    new { Text = "D1", Value = "10" },
    new { Text = "D", Value = "11" },
    new { Text = "DE", Value = "12" },
    new { Text = "F", Value = "13" },
    new { Text = "G", Value = "14" }
};
cb_ehliyettur.DataSource = items2;
}

private void dataG_user_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    tb_tc.Text = dataG_user.CurrentRow.Cells[1].Value.ToString();
    tb_ad.Text = dataG_user.CurrentRow.Cells[2].Value.ToString();
    tb_soyad.Text = dataG_user.CurrentRow.Cells[3].Value.ToString();
    if (dataG_user.CurrentRow.Cells[4].Value.ToString()=="1")
    { cb_cinsiyet.SelectedIndex = 0;}
    else
    {cb_cinsiyet.SelectedIndex = 1;}
    dt_dogumT.Text = dataG_user.CurrentRow.Cells[5].Value.ToString();
    rtb_adres.Text = dataG_user.CurrentRow.Cells[6].Value.ToString();
    tb_mail.Text = dataG_user.CurrentRow.Cells[7].Value.ToString();
    tb_telno.Text = dataG_user.CurrentRow.Cells[8].Value.ToString();
    tb_ehliyetno.Text = dataG_user.CurrentRow.Cells[9].Value.ToString();
}

```

```

        cb_ehliyetkur.Text = dataG_user.CurrentRow.Cells[10].Value.ToString();
    }

    private void btn_userUpdate_Click(object sender, EventArgs e)
    {
        vt.kayitGuncelle("musteriler", "musteriTC", tb_tc.Text, tb_tc.Text, tb_ad.Text, tb_soyad.Text, cb_
        _cinsiyet.SelectedValue, dt_dogumT.Value.Date.ToString("yyyy-MM-
        dd"), rtb_adres.Text, tb_mail.Text, tb_telno.Text, tb_ehliyetno.Text, cb_ehliyetkur.Text);
    }
}

```

### 2.8.13. Müşteri Sil UserControl Kodları

```

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

namespace arackiralama
{
    public partial class userDelete : UserControl
    {
        public userDelete()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        private void btn_userDelete_Click(object sender, EventArgs e)
        {
            vt.verisil("musteriler", "musteriTC", cb_musteriTc.Text);
            cb_musteriTc.DataSource = vt.verial("?", "musteriler", "0", "0");
        }
    }
}

```

```

        cb_musteriTc.ValueMember = "musteriID";
        cb_musteriTc.DisplayMember = "musteriTC";

    }

    private void userDelete_Load(object sender, EventArgs e)
    {
        cb_musteriTc.DataSource = vt.verial("?", "musteriler", "0", "0");
        cb_musteriTc.ValueMember = "musteriID";
        cb_musteriTc.DisplayMember = "musteriTC";
    }
}

```

#### 2.8.14. Sigorta İşlemleri UserControl Kodları

```

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

namespace arackiralama
{
    public partial class sigorta : UserControl
    {
        public sigorta()
        {
            InitializeComponent();
        }

        vtClass vt = new vtClass(false);
        public void rSigorta()
    }
}

```

```

{
    data_sigorta.DataSource = vt.verial("sigortaID as ID, sigortaAd as SigortaAdı,
sigortaUcret as Ücret, sigortaBilgi as Açıklama", "sigortalar", "0", "0");
    cb_sigorta.DataSource = vt.verial("?", "sigortalar", "0", "0");
    cb_sigorta.ValueMember = "sigortaID";
    cb_sigorta.DisplayMember = "sigortaAd";
}
private void sigorta_Load(object sender, EventArgs e)
{
    rSigorta();
}

private void btn_sigortaAdd_Click(object sender, EventArgs e)
{
    vt.veriekle("sigortalar",tb_sigortaAd.Text,tb_sigortaUcret.Text,rtb_sigortaBilgi.Text);
    rSigorta();
}

private void btn_sigortaUpdate_Click(object sender, EventArgs e)
{
    vt.kayitGuncelle("sigortalar","sigortaAd",tb_sigortaAd.Text,
tb_sigortaAd.Text,tb_sigortaUcret.Text,rtb_sigortaBilgi.Text);
    rSigorta();
}

private void btn_sigortaDelete_Click(object sender, EventArgs e)
{
    vt.verisil("sigortalar","sigortaID",cb_sigorta.SelectedValue);
    rSigorta();
}

private void data_sigorta_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    tb_sigortaAd.Text = data_sigorta.CurrentRow.Cells[1].Value.ToString();
    tb_sigortaUcret.Text = data_sigorta.CurrentRow.Cells[2].Value.ToString();
}

```

```

        rtb_sigortaBilgi.Text = data_sigorta.CurrentRow.Cells[3].Value.ToString();
    }
}

```

### 2.8.15. İstatistik UserControl Kodları

```

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

namespace arackiralama
{
    public partial class ıstatistik : UserControl
    {
        public ıstatistik()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        public void showControl(Control control)
        {
            ist_Content.Controls.Clear();
            control.Dock = DockStyle.Fill;
            control.BringToFront();
            control.Focus();
            ist_Content.Controls.Add(control);
        }
        private void btn_carsInfo_Click(object sender, EventArgs e)
        {

```

```

carsİ hp = new carsİ();
showControl(hp);
}

private void btn_usersInfo_Click(object sender, EventArgs e)
{
    usersİ hp = new usersİ();
    showControl(hp);
}

private void btn_kiraAll_Click(object sender, EventArgs e)
{
    fullList fL = new fullList(vt.verial("(modeller.modelAd+' '+araclar.aracPlaka) as Araç,
(musteriler.musteriAd + ' '+ musteriler.musteriSoyad) as Müşteri, (sigortalar.sigortaAd + ' '+
CONVERT(varchar(8),sigortalar.sigortaUcret)      +'      TL')      as      Sigorta,
CONVERT(varchar(8),kiralama.kiraUcret)+'      TL'      as
TOPLAMTUTAR,kiralama.kiraGirisTarih as BaşlangıçTarihi,kiralama.kiraBitisTarih as
BitişTarihi, iller.ilAd as İl", "kiralama", "INNER JOIN musteriler ON
musteriler.musteriID=kiralama.kiraMusteriID INNER JOIN sigortalar On
sigortalar.sigortaID=kiralama.kiraSigorta INNER JOIN araclar ON
araclar.aracID=kiralama.kiraAracID INNER JOIN modeller ON
modeller.modelID=araclar.modelID INNER JOIN iller ON iller.ilID=kiralama.kiraİl", "0"));

    showControl(fL);
}
}
}

```

## 2.8.16. Araç İstatistikleri UserControl Kodları

```

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

```

```

namespace arackiralama
{
    public partial class carsİ : UserControl
    {
        public carsİ()
        {
            InitializeComponent();
        }

        vtClass vt = new vtClass(false);

        private void timer1_Tick(object sender, EventArgs e)
        {
            lbl_carsCount.Text = vt.verial("Count(*)", "araclar", "0", "0").Rows[0][0].ToString() +
" araç";

            lbl_carsKiraT.Text = vt.verial("Count(*)", "araclar", "0",
"aracKiraDurum=1").Rows[0][0].ToString() + " araç";

            lbl_carsKiraF.Text = vt.verial("Count(*)", "araclar", "0",
"aracKiraDurum=0").Rows[0][0].ToString() + " araç";

            lbl_carsDay.Text = vt.verial("SUM(kiraUcret)", "kiralama", "0", "kiraGirisTarih = '"+
DateTime.Today.ToString("yyyy-MM-dd") + "']").Rows[0][0].ToString() + " TL";

            lbl_carsMonth.Text = vt.verial("SUM(kiraUcret)", "kiralama", "0", "kiraGirisTarih >=
'" + (DateTime.Now.Year.ToString() + "-" + DateTime.Now.Month.ToString() + "-
1") + "']").Rows[0][0].ToString() + " TL";

            lbl_carsYear.Text = vt.verial("SUM(kiraUcret)", "kiralama", "0", "kiraGirisTarih >
'" + (DateTime.Now.Year.ToString() + "-1-1") + "']").Rows[0][0].ToString() + " TL";

            lbl_carsLast.Text = vt.verial("(musteriler.musteriAd + ' ' + musteriler.musteriSoyad + ' /
' + araclar.aracPlaka + ' / ' + CONVERT(varchar(8), kiralama.kiraUcret) + ' TL') as Adsoyad",
"kiralama", "INNER JOIN musteriler ON musteriler.musteriID=kiralama.kiraMusteriID
INNER JOIN araclar ON araclar.aracID=kiralama.kiraAracID ORDER BY kiralama.kiraID
DESC", "0").Rows[0][0].ToString();

            cars_grafik.DataSource = vt.verial("distinct markalar.markaAd,(((Select Count(*)
FROM modeller WHERE modeller.modelMarka=markalar.markaID)*100)/(select Count(*)

```



```
FROM modeller)) as yuzde", "markalar", "INNER JOIN modeller ON modeller.modelMarka
= markalar.markaID", "0");
```

```
cars_grafik.Series[0].XValueMember = "markaAd";
cars_grafik.Series[0].YValueMembers = "yuzde";
cars_grafik.Series[0].Name = "Grafik";
cars_grafik.Series[0].ChartType = SeriesChartType.Pie;//grafik şeklini değiştirir.
cars_grafik.DataBind();
    }
}
}
```

## 2.8.17. Müşteri İstatistikleri UserControl Kodları

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

```
namespace arackiralama
```

```
{
    public partial class usersİ : UserControl
    {
        public usersİ()
        {
            InitializeComponent();
        }
        vtClass vt = new vtClass(false);
        private void timer1_Tick(object sender, EventArgs e)
        {
            lbl_erkek.Text = vt.verial("Count(*)", "musteriler", "0",
"musteriCinsiyet=0").Rows[0][0].ToString() + " Erkek Müşteri";
```

```

        lbl_kadin.Text = vt.verial("Count(*)", "musteriler", "0",
"musteriCinsiyet='1'").Rows[0][0].ToString() + " Kadın Müşteri";

        lbl_erkekMoney.Text = vt.verial("SUM(kiralama.kiraUcret)", "kiralama", "INNER
JOIN        musteriler        ON        musteriler.musteriID=kiralama.kiraMusteriID",
"musteriler.musteriCinsiyet='0'").Rows[0][0].ToString() + " TL";

        lbl_kadinMoney.Text = vt.verial("SUM(kiralama.kiraUcret)", "kiralama", "INNER
JOIN        musteriler        ON        musteriler.musteriID=kiralama.kiraMusteriID",
"musteriler.musteriCinsiyet='1'").Rows[0][0].ToString() + " TL";

    }
}
}

```

### 2.8.18. VtClass Kodları

```

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

namespace arackiralama
{
    class vtClass
    {
        public bool HataYakala = false;

        public vtClass(bool hata)
        {
            HataYakala = hata;
        }

        SqlConnection bag = new SqlConnection("server=PC\\SQLEXPRESS;initial
catalog=arackiralama;integrated security = true");

```

```

public DataTable verial(string alan, string tablo,string join, string sorgu)//kayıt çekme
metodu
{
    DataTable dt = new DataTable();
    try
    {
        bag.Open();
        string veriAlSorgu = "";
        //SqlDataAdapter da = new SqlDataAdapter("Select " + alan + " From " + tablo, bag);
        if (join=="0" && sorgu=="0")
        {
            veriAlSorgu = ("Select " + alan + " From " + tablo);
        }
        else if (join != "0" && sorgu == "0")
        {
            veriAlSorgu = ("Select " + alan + " From " + tablo + " " + join);
            //da = new SqlDataAdapter("Select " + alan + " From " + tablo + " " + join, bag);
        }
        else if (sorgu != "0" && join == "0")
        {
            veriAlSorgu = ("Select " + alan + " From " + tablo + " Where " + sorgu);
            //da = new SqlDataAdapter("Select " + alan + " From " + tablo + " Where " + sorgu,
bag);
        }
        else if (join != "0" && sorgu != "0")
        {
            veriAlSorgu = ("Select " + alan + " From " + tablo + " " + join + " Where " +
sorgu);
            //da = new SqlDataAdapter("Select " + alan + " From " + tablo + " " + join + "
Where " + sorgu, bag);
        }
        if (HataYakala==true)
        {
            MessageBox.Show("Sql Sorgu = "+veriAlSorgu);
        }
        SqlCommand komut = new SqlCommand(veriAlSorgu,bag);
    }
}

```

```

        SqlDataAdapter da = new SqlDataAdapter(komut);
        da.Fill(dt);
        bag.Close();
        veriAlSorgu = "";
    }
    catch (Exception e)
    {
        if (HataYakala == true)
        {
            MessageBox.Show(e.Message);
        }
    }
    bag.Close();
    return dt;
}

public bool veriekle(string tablo, params object[] liste)//kayıt ekleme metodu
{
    SqlCommand kmt = new SqlCommand();
    string ekleSorgu = "INSERT INTO " + tablo + " (";
    ArrayList KolonAdlariList;
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM " + tablo, bag);
    DataTable dt2 = new DataTable();
    da.Fill(dt2);
    KolonAdlariList = new ArrayList();
    for (int i = 1; i < dt2.Columns.Count; i++)
    {
        DataColumn dr = dt2.Columns[i];
        KolonAdlariList.Add(dr.ToString());
    }
    for (int k = 0; k < KolonAdlariList.Count; k++)
    {
        if (k != KolonAdlariList.Count - 1)
        {
            ekleSorgu = ekleSorgu + " " + KolonAdlariList[k] + ",";
        }
    }
}

```

```

else
{
    ekleSorgu = ekleSorgu + " " + KolonAdlariList[k];
}
}
ekleSorgu = ekleSorgu + ") VALUES (";
for (int i = 0; i < KolonAdlariList.Count; i++)
{

    if (i != KolonAdlariList.Count - 1)
    {
        if (liste[i] is string)
        {
            kmt.Parameters.AddWithValue("@ " + KolonAdlariList[i].ToString() + " ",
liste[i].ToString());
            ekleSorgu = ekleSorgu + "@ " + KolonAdlariList[i] + ",";
        }
        else if (liste[i] is int)
        {
            kmt.Parameters.AddWithValue("@ " + KolonAdlariList[i].ToString() + " ",
liste[i].ToString());
            ekleSorgu = ekleSorgu + "@ " + KolonAdlariList[i] + ",";
        }
        else
        {
            kmt.Parameters.AddWithValue("@ " + KolonAdlariList[i].ToString() + " ",
null);
            ekleSorgu = ekleSorgu + "@ " + KolonAdlariList[i] + ",";
        }
    }
    else
    {
        if (liste[i] is string)
        {
            kmt.Parameters.AddWithValue("@ " + KolonAdlariList[i].ToString() + " ",
liste[i]);

```

```

        ekleSorgu = ekleSorgu + "@" + KolonAdlariList[i];
    }
    else if (liste[i] is int)
    {
        kmt.Parameters.AddWithValue("@ " + KolonAdlariList[i].ToString() + "",
liste[i]);
        ekleSorgu = ekleSorgu + "@" + KolonAdlariList[i];
    }
    else
    {
        kmt.Parameters.AddWithValue("@ " + KolonAdlariList[i].ToString() + "",
null);
        ekleSorgu = ekleSorgu + "@" + KolonAdlariList[i] + ",";
    }
}

}
ekleSorgu = ekleSorgu + ")";
//////////Sorgu Oluşturuldu
if (HataYakala==true)
{
    MessageBox.Show("Sql Sorgusu = "+ekleSorgu);
}
kmt.Connection = bag;
kmt.CommandText = ekleSorgu;
bag.Open();
if (kmt.ExecuteNonQuery() ==1)
{
    MessageBox.Show("Veri Kaydı Başarılı");
}
else
{
    MessageBox.Show("Veri Kaydı Başarısız");
}
kmt.Dispose();
bag.Close();

```

```

        return true;
    }
    public void verisil(string tablo, string alan, object deger)//kayıt silme metodu
    {
        SqlCommand kmt = new SqlCommand();
        bag.Open();
        kmt.Connection = bag;
        kmt.CommandText = "";
        if (deger is int)
        {
            kmt.CommandText = "DELETE FROM " + tablo + " WHERE " + alan + " = " + (int)deger + "";
        }
        else
        {
            kmt.CommandText = "DELETE FROM " + tablo + " WHERE " + alan + " = " + (string)deger + "";
        }
        if (kmt.ExecuteNonQuery() == 1)
        {
            bag.Close();
            MessageBox.Show("Veri Silindi");
        }
        else
        {
            bag.Close();
            MessageBox.Show("Veri Silinemedi");
            if (HataYakala == true)
            {
                MessageBox.Show(kmt.CommandText);
            }
        }
        kmt.Dispose();
    }

    public void kayitGuncelle(string tabloAdi, string alanAdi, object deger, params object[]
liste)//kayıt güncelleme metodu

```

```

{
    /// parametre listeli sorgu oluřturuyor
    string ekleSorgu = "UPDATE " + tabloAdi + " SET ";
    ArrayList KolonAdlariListGelen;
    SqlCommand kmt = new SqlCommand();
    bag.Open();

    ///alan adları alınıyor
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM " + tabloAdi, bag);
    DataTable dt2 = new DataTable();
    da.Fill(dt2);
    KolonAdlariListGelen = new ArrayList();
    ///array list ieirisine alan adlarını aktarıyoruz

    for (int i = 1; i < dt2.Columns.Count; i++)
    {
        DataColumn dr = dt2.Columns[i];
        KolonAdlariListGelen.Add(dr.ToString());
    }
    ///alanadlarını array liste e ekleme iřlemi bitti
    for (int i = 0; i < KolonAdlariListGelen.Count; i++)
    {
        if (i != KolonAdlariListGelen.Count - 1)
        {
            if (liste[i] is string)
            {
                if (liste[i] == null)
                    liste[i] = "";
                kmt.Parameters.Add("@ " + KolonAdlariListGelen[i].ToString(),
                liste[i].ToString());
                ekleSorgu = ekleSorgu + KolonAdlariListGelen[i].ToString() + "=" + "@" +
                KolonAdlariListGelen[i] + ",";
            }
            else if (liste[i] is int)
            {
                if (liste[i] == null)

```



```

        liste[i] = 0;
        kmt.Parameters.Add("@ " + KolonAdlariListGelen[i].ToString(),
liste[i].ToString());
        ekleSorgu = ekleSorgu + KolonAdlariListGelen[i].ToString() + "=" + "@" +
KolonAdlariListGelen[i] + ",";
    }
}
else
{
    if (liste[i] is string)
    {
        if (liste[i] == null)
            liste[i] = "";
        kmt.Parameters.Add("@ " + KolonAdlariListGelen[i].ToString(),
liste[i].ToString());
        ekleSorgu = ekleSorgu + KolonAdlariListGelen[i].ToString() + "=" + "@" +
KolonAdlariListGelen[i];
    }
    else if (liste[i] is int)
    {
        if (liste[i] == null)
            liste[i] = 0;
        kmt.Parameters.Add("@ " + KolonAdlariListGelen[i].ToString(),
liste[i].ToString());
        ekleSorgu = ekleSorgu + KolonAdlariListGelen[i].ToString() + "=" + "@" +
KolonAdlariListGelen[i];
    }
}

}
ekleSorgu = ekleSorgu + " WHERE " + alanAdi + "= " + deger + """;
//////////sorgu oluřturuldu
if (HataYakala==true)
{
    MessageBox.Show(ekleSorgu);
}

```

```

kmt.Connection = bag;
kmt.CommandText = ekleSorgu;
if (HataYakala==true)
{
    MessageBox.Show("Sql Sorgusu = "+ekleSorgu);
}
if (kmt.ExecuteNonQuery() == 1)
{
    bag.Close();
    MessageBox.Show("Güncelleme Başarılı");
}
else
{
    bag.Close();
    MessageBox.Show("Güncelleme Başarısız");
    if (HataYakala == true)
    {
        MessageBox.Show(kmt.CommandText);
    }
}
kmt.Dispose();
bag.Close();
}

public bool sorguGonder(string sorgu)
{
    try
    {
        bag.Open();
        SqlCommand kmt = new SqlCommand();
        kmt.Connection = bag;
        kmt.CommandText = sorgu;
        //MessageBox.Show(sorgu.ToString());
        if (kmt.ExecuteNonQuery() == 1)
        {
            kmt.Dispose();

```

```

        return false;
    }
    else
    {
        kmt.Dispose();
        return true;
    }
}
catch (Exception e)
{
    return false;
    if (HataYakala == true)
    {
        MessageBox.Show(e.Message);
    }
}
bag.Close();
}
}
}

```