**T.C**

**KIRIKKALE ÜNİVERSİTESİ**

**MESLEK YÜKSEKOKULU**

**BİLGİSAYAR PROGRAMCILIĞI**

**SİSTEM ANALİZİ ve TASARIMI DERSİ**

**TELEFON SATIŞ PROJESİ**

**Doç. Dr. Aşır ÖZBEK**

**İlayda TEKELİ**

**210720022**

**KIRIKKALE – 2023**

İÇİNDEKİLER

[Şekiller Listesi](#_Toc122822585)

[Tablolar Listesi](#_Toc122822586)

[GİRİŞAGEREF1](#_Toc122822587)

[1. GEREKLİLİK ANALİZİ](#_Toc122822588)

[2. KAVRAMSAL TASARIM](#_Toc122822589)

[2.1. Varlık Kümeleri](#_Toc122822590)

[2.1.1. Telefonlar Varlık Kümesi2](#_Toc122822591)

[2.1.2. Ram Bellek Varlık Kümesi](#_Toc122822592)

[2.1.3. Dahili Depolama Varlık Kümesi](#_Toc122822593)

[2.1.4. Ekran Boyutu Varlık Kümesi](#_Toc122822594)

[2.1.5. Markalar Varlık Kümesi](#_Toc122822595)

[2.1.6.Batarya Kapasitesi Varlık Kümesi Varlık Kümesi](#_Toc122822596)

2.1.7. Kamera Çözünürlüğü Varlık Kümesi

2.1.8. Ön Kamera Çözünürlüğü Varlık Kümesi

2.1.9. Ekran Yenileme Hızı Varlık Kümesi

2.1.10.Desteklenen Frekans Varlık Kümesi

2.1.11.Hat Sayısı Varlık Kümesi

2.1.12.Hızlı Şarj Desteği Varlık Kümesi

2.1.13 Parmak İzi Okuyucu Varlık Kümesi

2.1.14 Renkler Varlık Kümesi

2.1.15.Kablosuz Şarj Varlık Kümesi

2.1.16.Hafıza Kartı Desteği Varlık Kümesi

2.1.17. Çıkış Yılı Varlık Kümesi

2.1.18. Şubeler Varlık Kümesi

2.1.19. Satışlar Varlık Kümesi

2.1.20 Personeller Varlık Kümesi

2.1.21 Müşteriler Varlık Kümesi

[2.2. Varlık Kümeleri Arasındaki İlişki](#_Toc122822597)

[2.2.1. Telefonlar –Nitelik İlişkisi](#_Toc122822598)

[2.2.2. Telefonlar Satışlar İlişkisi](#_Toc122822599)

[2.2.3. Satışlar –Şubeler İlişkisi](#_Toc122822600)

[2.2.4. Satışlar-Personeller İlişkisi](#_Toc122822601)

[2.2.5. Satışlar –Müşteriler İlişkisi](#_Toc122822602)

[2.2.6. Genel Gösterim](#_Toc122822603)

[3. MANTIKSAL TASARIM](#_Toc122822604)

[3.1. Telefonlar Tablosu](#_Toc122822605)

[3.2. Ram Bellek Tablosu](#_Toc122822606)

[3.3. Dahili Depolama Tablosu](#_Toc122822607)

[3.4. Ekran Boyutu Tablosu](#_Toc122822608)

[3.5. Markalar Tablosu](#_Toc122822609)

[3.6. Batarya Kapasitesi Tablosu](#_Toc122822610)

[3.7. Kamera Çözünürlüğü Tablosu](#_Toc122822611)

3.8 Ön Kamera Çözünürlüğü Tablosu

3.9 Ekran Yenileme Hızı Tablosu

3.10 Desteklenen Frekans Tablosu

3.11 Hat Sayısı Tablosu

3.12 Hızlı Şarj Desteği Tablosu

3.13Parmak izi Okuyucu

3.14 Renkler Tablosu

3.15Kablosuz Şarj Tablosu

3.16 Hafıza Kartı Desteği Tablosu

3.17 İşletim sistemi Tablosu

3.18 Çıkış Yılı Tablosu

3.19 Satışlar Tablosu

3.20 Şubeler Tablosu

3.21 Personeller Tablosu

3.22 Müşteriler Tablosu

3.23 Admin Tablosu

[4. ŞEMATİK FİLTRELEME](#_Toc122822612)

[4.1. Telefonlar Veri Tablosu](#_Toc122822613)

[4.2. Ram Bellek Tablosu](#_Toc122822614)

[4.3. Dahili Depolama Veri Tablosu](#_Toc122822615)

[4.4. Ekran Boyutu Veri Tablosu](#_Toc122822616)

[4.5. Markalar Veri Tablosu](#_Toc122822617)

[4.6. Batarya Kapasitesi Veri Tablosu](#_Toc122822618)

[4.7. Kamera Çözünürlüğü Veri Tablosu](#_Toc122822619)

4.8 Ön Kamera Çözünürlüğü Veri Tablosu

4.9 Ekran Yenileme Hızı Veri Tablosu

4.10. Desteklenen Frekans Tablosu

4.11. Hat Sayısı Veri Tablosu

4.12 Hızlı Şarj Desteği Veri Tablosu

4.13 Parmak izi Okuyucu Veri Tablosu

4.14Renkler Veri Tablosu

4.15 Kablosuz Şarj Veri Tablosu

4.16 Hafıza Desteği Veri Tablosu

4.17 İşletim sistemi Veri Tablosu

4.18Çıkış Yılı Veri Tablosu

4.19 Satışlar Veri Tablosu

4.20 Şubeler Veri Tablosu

4.21 Personeller Veri Tablosu

4.22 Müşteriler Veri Tablosu

[5. FIZIKSEL TASARIM](#_Toc122822620)(Ek2- Ek3)

5.1.Telefonlar Tablosu

5.2.Ram Bellek Tablosu

5.3. Dahili Depolama Tablosu

5.4 Ekran Boyutu Tablosu

5.5 Markalar Tablosu

5.6. Batarya Kapasitesi Tablosu

5.7. Kamera Çözünürlüğü Tablosu

5.8. Ön Kamera Çözünürlüğü Tablosu

5.9. Ekran Yenileme HızıTablosu

5.10 Desteklenen Frekans Tablosu

5.11 Hat Sayısı Tablosu

5.12 Hızlı Şarj Desteği Tablosu

5.13 Parmak İzi Okuyucu Tablosu

5.14 Renkler Tablosu

5.15 Kablosuz Şarj Tablosu

5.16 Hafıza Kartı Desteği Tablosu

5.17 Çıkış Yılı Tablosu

5.18 İşletim Sistemi Tablosu

5.19 Satışlar Tablosu

5.20 Şubeler Tablosu

5.21 Personeller Tablosu

5.22 Müşteriler Tablosu

[6. UYGULAMA](#_Toc122822628) ARAYÜZÜ ve Özellikleri

6.1 Anasayfa Arayüzü

6.2 Giriş Arayüzü

6.3 Admin Liste Arayüz

6.4Admin Sifre Güncelleme Arayüzü

6.5 Admin Silme Arayüz

6.6 Admin Ekleme Arayüz

6.7 AdminGüncelleme Arayüz

6.8 GecnşsSatışlar Arayüz

6.9 Musteri Islemerli Arayüzü

6.10 Personel Islemleri Arayuzu

6.11 Personel Liste Arayüzü

6.12 Personel Arayüzü

6.13 Personel Performans Arayüzü

6.14 PersonelSifreGuncelle Arayüzü

6.15 SatisYap Arayüzü

6.16 Stok Arayüzü

6.17 StokArttır Arayüzü

6.18 Sube Liste Arayüzü

6.19 Sube Islemleri Arayüzü

6.20 TabloEkle Arayüzü

6.21Telefon Bilgileri Arayüzü

**7-Veri Tabanı Degisiklikleri**

**8-Proje Kodları Acıklamaları**

8.1 Veri Tabanı Trigger Acıklamaları

8.2 Kodlar ve Acıklamaları

**9 Zaman İş Planı**

9.0 Zaman İş Planı Çizelgesi

10- Akıs Seması

# GİRİŞ

Telefon satış programının amacı ; Telefonun özelliklerini bir veri tabanında düzenli bir şekilde saklayıp bu verilere hızlı bir şekilde erişebilmek, düzenleyebilmek, güncelleyebilmek, ekleme ve çıkarma yapabilmeyi sağlamaktır.

# GEREKLİLİK ANALİZİ

Farklı Şubelere sahip telefon satış firması için şubeler arasında arasında oluşabilecek yanlış anlaşılmaları en aza indirtilmek için tasarlanacak olan veri tabanı aşağıda belirtilen verileri saklamalıdır ve bu bilgiler üzerinde yapılabilecek sorgu (görüntüleme, ekleme, silme ve güncelleme) ve değişiklere cevap vermelidir. Elde edilen bilgiler doğrultusunda kullanıcı detayları aşağıdaki gibi belirlenmiştir.

* Telefonların kayıt altında tutulması
* Müşteri bilgilerinin kayıt altında tutulması
* Personel bilgilerinin kayıt altında tutulması
* Satışların kayıt altında tutulması ve bir iade durumda kontrol edilmesi
* Stok bilgilerinin sürekli kontrol edilmesi ve kayıt aldında tutulması

olarak belirlenmiştir.

# KAVRAMSAL TASARIM

Gereklilik analizinde ihtiyaç duyulanlar doğrultusunda aşağıda belirtilen varlık kümeleri oluşturulmuştur. Varlık kümeleri bu ihtiyaçlar doğrultusunda oluşturulur. Veri tabanında kullanılacak olan varlık türleri ve bunlar arasındaki ilişkiler detaysız bir şekilde ele alınmamıştır:) Böylelikle oluşturulan kavramsal şemanın kullanıcı gereksinimlerini karşılayıp karşılamadığı belirlenmiştir. Belirlenen varlık kümeleri aşağıda belirtilmiştir.

* Telefonlar Varlık Kümesi
* Ram Bellek Varlık Kümesi
* Dahili Depolama Varlık Kümesi
* Ekran Boyutu Varlık Kümesi
* Markalar Varlık Kümesi
* Batarya Kapasitesi Varlık Kümesi
* Kamera Çözünürlüğü Varlık Kümesi
* Ön Kamera Çözünürlüğü Varlık Kümesi
* Ekran Yenileme Hızı Varlık Kümesi
* Desteklenen Frekans Varlık Kümesi
* Hat Sayısı Varlık Kümesi
* Hızlı Şarj Desteği Varlık Kümesi
* Parmak İzi Okuyucu Varlık Kümesi
* Renkler Varlık Kümesi
* Kablosuz Şarj Varlık Kümesi
* Hafıza Kartı Desteği Varlık Kümesi
* Çıkış Yılı Varlık Kümesi
* İşletim Sistemi Varlık Kümesi
* Satışlar Varlık Kümesi
* Şubeler Varlık Kümesi
* Personeller Varlık Kümesi
* Müşteriler Varlık Kümesi

## VVarlık Kümeleri

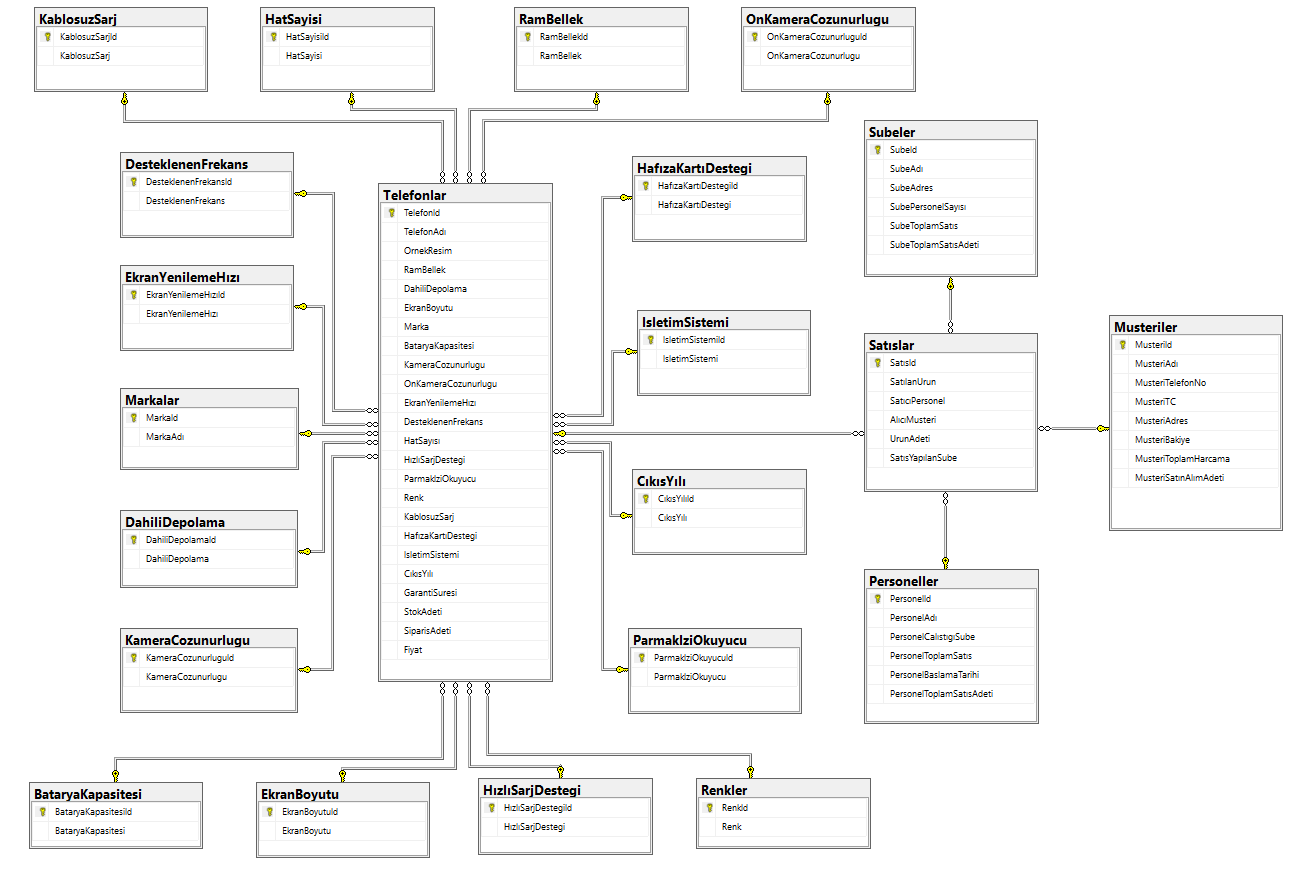
Oluşturulan varlık kümeleri tek tek incelenmek zorunda kalınılmıştır ve her bir varlık kümesi onu oluşturan nitelikler ile ER şeması kullanılarak görselleştirilmiştir.

### Telefonlar Varlık Kümesi

Telefon Varlık Kümesi; satın alım yapan müşterilerin istediği gibi filtreleme ve sıralama yapabilmeleri amacıyla oluşturulmuştur. Buna göre Telefon Id anahtar nitelik olarak belirlenmiştir. Aşağı sayfadaki **diagram\_1** de gösterilmiştir. Bazı özellikler için ayrı tablo yapmak bizi veri tekrarına düşmekten kurtaracağı için 17 adet nitelik için ayrı tablo yapılmıştır. İade istemlerinde kolay olması amacıyla satış bilgileri, yüksek kâr sağlayan şubelere yatırım yapılması amacıyla şube bilgileri, sürekli alışveriş yapan müşterileri belirleyip reklam göstermek amacıyla müşteri bilgileri, yüksek satış oranına sahip kalifiye elemanların belirlenmesi amacıyla ise personel bilgileri kayıt altına alınmıştır. **TelefonId** anahtar nitelik olacak şekilde;

* Telefon ID
* Telefon Adı
* Örnek Resim
* Ram Bellek ID
* Dahili Depolama ID
* Ekran Boyutu ID
* Marka ID
* Batarya Kapasitesi ID
* Kamera Çözünürlüğü ID
* Ön Kamera Çözünürlüğü ID
* Ekran Yenileme Hızı ID
* Desteklenen Frekans ID
* Hat Sayısı ID
* Hızlı Şarj Desteği ID
* Parmak İzi Okuyucu ID
* Renk ID
* Kablosuz Şarj ID
* Hafıza Kartı Desteği ID
* İşletim Sistemi ID
* Çıkış Yılı ID
* Garanti Süresi
* Stok Adeti
* Sipariş Adeti
* Fiyat

olarak belirlenmiş olup çok fazla varlık kümesi olmasından dolayı ER şemasıyla değil diagram kullanılarak Diagram-1’ de gösterilmiştir.

**Diagram-1** Genel diagram

### Ram Bellek Varlık Kümesi

Ram Bellek varlık kümesi, Ram bellek miktarlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan  **Ram Bellek ID** ve kümeyi tanımlayan diğer nitelik Ram Belleğin miktarını belirtir.

* Ram Bellek ID
* Ram Bellek

Belirtilen nitelikler doğrultusunda Ram Bellek varlık kümesi Şekil-1’de ER şeması yardımıyla gösterilmiştir.

Ram Bellek

**Şekil-1** Ram Bellek ER şeması

### DDahili Depolama Varlık Kümesi

Dahili Depolama varlık kümesi, Dahili Depolama miktarlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Dahili Depolama ID** ve kümeyi tanımlayan diğer nitelik Dahili Depolama miktarını belirtir.

* Dahili Depolama ID
* Dahili Depolama

Belirtilen nitelikler doğrultusunda Dahili Depolama varlık kümesi Şekil-3’te ER şeması yardımıyla gösterilmiştir.

Dahili Depolama

**Şekil-2** Dahili depolama ER şeması

### EEkran Boyutu Varlık Kümesi

Ekran Boyutu varlık kümesi, Ekran Boyutu miktarlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Ekran Boyutu ID** ve kümeyi tanımlayan diğer nitelik Dahili Depolama miktarını belirtir.

* Ekran Boyutu ID
* Ekran Boyutu

Belirtilen nitelikler doğrultusunda Ekran Boyutu varlık kümesi Şekil-3’te ER şeması yardımıyla gösterilmiştir.

Ekran Boyutu

**Şekil-3** Ekran Boyutu ER şeması

### Markalar Varlık Kümesi

Markalar varlık kümesi, Marka adlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan  **Marka ID** ve kümeyi tanımlayan diğer nitelik Markanın adını belirtir.

* Marka ID
* Marka

Belirtilen nitelikler doğrultusunda Markalar varlık kümesi Şekil-4’te ER şeması yardımıyla gösterilmiştir.

Markalar

**Şekil-4** Markalar ER şeması

### Batarya Kapasitesi Varlık Kümesi

Batarya Kapasitesi varlık kümesi, Batarya Kapasitesinin sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan  **Batarya Kapasitesi ID** ve kümeyi tanımlayan diğer nitelik Ram Belleğin miktarını belirtir.

* Batarya Kapasitesi ID
* Batarya Kapasitesi

Belirtilen nitelikler doğrultusunda Batarya Kapasitesi varlık kümesi Şekil-5’te ER şeması yardımıyla gösterilmiştir.

Batarya Kapasitesi

**Şekil-5** Batarya Kapasitesi ER şeması

### Kamera Çözünürlüğü Varlık Kümesi

Kamera Çözünürlüğü varlık kümesi, Kamera Çözünürlüğü miktarlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Kamera Çözünürlüğü ID** ve kümeyi tanımlayan diğer nitelik Kamera çözünürlüğünü belirtir.

* Kamera Çözünürlüğü ID
* Kamera Çözünürlüğü

Belirtilen nitelikler doğrultusunda Kamera Çözünürlüğü varlık kümesi Şekil-6’da ER şeması yardımıyla gösterilmiştir.

Kamera Çözünürlüğü

**Şekil-6** Kamera Çözünürlüğü ER şeması

### Ön Kamera Çözünürlüğü Varlık Kümesi

Ön Kamera Çözünürlüğü varlık kümesi, Ön Kamera Çözünürlüğü miktarlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Ön** **Kamera Çözünürlüğü ID** ve kümeyi tanımlayan diğer nitelik Kamera çözünürlüğünü belirtir.

* Ön Kamera Çözünürlüğü ID
* Ön Kamera Çözünürlüğü

Belirtilen nitelikler doğrultusunda Ön Kamera Çözünürlüğü varlık kümesi Şekil-7’de ER şeması yardımıyla gösterilmiştir.

Ön Kamera Çözünürlüğü

**Şekil-7** Ön Kamera çözünürlüğü ER şeması

### Ekran Yenileme Hızı Varlık Kümesi

Ekran Yenileme Hızı varlık kümesi, Ekran Yenileme Hızı miktarlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Ekran Yenileme Hızı ID** ve kümeyi tanımlayan diğer nitelik Ekran Yenileme Hızını belirtir.

* Ekran Yenileme Hızı ID
* Ekran Yenileme Hızı

Belirtilen nitelikler doğrultusunda Ekran Yenileme Hızı varlık kümesi Şekil-8’de ER şeması yardımıyla gösterilmiştir.

Ekran Yenileme Hızı

**Şekil-8** Ekran Yenileme Hızı ER şeması

### Desteklenen Frekans Varlık Kümesi

Desteklenen Frekans varlık kümesi, Ekran Yenileme Hızı miktarlarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Desteklenen Frekans ID** ve kümeyi tanımlayan diğer nitelik Ekran Yenileme Hızını belirtir.

* Desteklenen Frekans ID
* Desteklenen Frekans Hızı

Belirtilen nitelikler doğrultusunda Desteklenen Frekans varlık kümesi Şekil-9’da ER şeması yardımıyla gösterilmiştir.

Desteklenen Frekans

**Şekil-9** Desteklenen Frekans ER şeması

### Hat Sayısı Varlık Kümesi

Hat Sayısı varlık kümesi, Hat Sayısı miktarının sürekli veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Hat Sayısı ID** ve kümeyi tanımlayan diğer nitelik Hat Sayısını belirtir.

* Hat Sayısı ID
* Hat Sayısı Hızı

Belirtilen nitelikler doğrultusunda Hat Sayısı varlık kümesi Şekil-10’da ER şeması yardımıyla gösterilmiştir.

Hat Sayısı

**Şekil-10** Hat Sayısı ER şeması

### Hızlı Şarj Desteği Varlık Kümesi

Hızlı Şarj Desteği varlık kümesi, Hızlı Şarj Desteğinin var olup olmamasının yazılması veri tekrarına sebep olacağı için için ayrıca oluşturulmuştur Anahtar nitelik olan **Hızlı Şarj Desteği ID** ve kümeyi tanımlayan diğer nitelik Hızlı Şarj Desteğini belirtir.

* Hızlı Şarj Desteği ID
* Hızlı Şarj Desteği

Belirtilen nitelikler doğrultusunda Hızlı Şarj Desteği varlık kümesi Şekil-11’de ER şeması yardımıyla gösterilmiştir.

Hızlı Şarj Desteği

**Şekil-11** Hızlı Şarj Desteği ER şeması

### Parmak İzi Okuyucu Varlık Kümesi

Parmak İzi Okuyucu varlık kümesi, Parmak İzi Okuyucu özelliğinin olup olmamasının yazılması veri tekrarına sebep olacağı için için ayrıca oluşturulmuştur Anahtar nitelik olan **Parmak İzi Okuyucu ID** ve kümeyi tanımlayan diğer nitelik Parmak İzi Okuyucu özelliğinin olup olmamasını belirtir.

* Parmak İzi Okuyucu ID
* Parmak İzi Okuyucu

Belirtilen nitelikler doğrultusunda Parmak İzi Okuyucu varlık kümesi Şekil-12’de ER şeması yardımıyla gösterilmiştir.

Parmak İzi Okuyucu

**Şekil-12** Parmak İzi Okuyucu ER şeması

### Renkler Varlık Kümesi

Renkler varlık kümesi, Renklerin sürekli tanımlanmasının veri tekrarına sebep olacağı için için ayrıca oluşturulmuştur Anahtar nitelik olan **Renk ID** ve kümeyi tanımlayan diğer nitelik rengini belirtir.

* Renk ID
* Renk

Belirtilen nitelikler doğrultusunda Renkler varlık kümesi Şekil-13’te ER şeması yardımıyla gösterilmiştir.

Renkler

**Şekil-13** Renkler ER şeması

### KKablosuz Şarj Varlık Kümesi

Kablosuz Şarj varlık kümesi, Kablosuz Şarj özelliğinin sürekli tanımlanması veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **Kablosuz Şarj ID** ve kümeyi tanımlayan diğer nitelik Dahili Depolama miktarını belirtir.

* Kablosuz Şarj ID
* Kablosuz Şarj

Belirtilen nitelikler doğrultusunda Kablosuz Şarj varlık kümesi Şekil-14’te ER şeması yardımıyla gösterilmiştir.

Kablosuz Şarj

**Şekil-14** Kablosuz Şarj ER şeması

### Hafıza Kartı Desteği Varlık Kümesi

Hafıza Kartı Desteği varlık kümesi, Hafıza Kartı Desteğinin olup olmamasının sürekli tanımlanması veri tekrarına sebep olacağı için ayrıca oluşturulmuştur Anahtar nitelik olan  **Hafıza Kartı Desteği ID** ve kümeyi tanımlayan diğer nitelik Hafıza Kartı Desteğinin miktarını belirtir.

* Hafıza Kartı Desteği ID
* Hafıza Kartı Desteği

Belirtilen nitelikler doğrultusunda Hafıza Kartı Desteği varlık kümesi Şekil-15’te ER şeması yardımıyla gösterilmiştir.

Hafıza Kartı Desteği

**Şekil-15** Hafıza Kartı Desteği ER şeması

### İşletim Sistemi Varlık Kümesi

İşletim Sistemi varlık kümesi, İşletim Sisteminin sürekli tanımlanması veri tekrarına sebep olmaması için ayrıca oluşturulmuştur Anahtar nitelik olan **İşletim Sistemi ID** ve kümeyi tanımlayan diğer nitelik İşletim Sistemini belirtir.

* İşletim Sistemi ID
* İşletim Sistemi

Belirtilen nitelikler doğrultusunda İşletim Sistemi varlık kümesi Şekil-16’da ER şeması yardımıyla gösterilmiştir.

İşletim Sistemi

**Şekil-16** İşletim Sistemi ER şeması

### Çıkış Yılı Varlık Kümesi

Çıkış Yılı varlık kümesi, Çıkış Yılının sürekli tanımlanması veri tekrarına sebep olacağı için ayrıca oluşturulmuştur. Anahtar nitelik olan  **Çıkış Yılı ID** ve kümeyi tanımlayan diğer nitelik Hafıza Kartı Desteğinin miktarını belirtir.

* Çıkış Yılı ID
* Çıkış Yılı

Belirtilen nitelikler doğrultusunda Çıkış Yılı varlık kümesi Şekil-17’de ER şeması yardımıyla gösterilmiştir.

Çıkış Yılı

**Şekil-17** Çıkış Yılı ER şeması

### Şubeler Varlık Kümesi

Şubeler varlık kümesi, Şube bilgilerini kayıt altına almak için oluşturulmuştur. Buna göre nitelikleri aşağıdaki gibi belirtilmiş olup **Şube ID** anahtar nitelik olarak kullanılmıştır.

* Şube ID
* Şube Adı
* Şube Adresi
* Şube Personel Sayısı
* Şube Toplam Satış Adeti
* Şube Toplam Satış

Tasarım Şekil-18’de gösterilmiştir.

Şube Adı

Şube Adres

Şubeler

**Şekil-18** Şubeler ER şeması

### Satışlar Varlık Kümesi

Satışlar varlık kümesi, satışlara ait bilgileri kayıt altına almak için oluşturulmuştur. Buna göre nitelikleri aşağıdaki gibi belirtilmiş olup **Satış ID** anahtar nitelik olarak kullanılmıştır.

* Satış ID
* Telefon ID
* Personel ID
* Müşteri ID
* Ürün Adeti
* Şube ID

Tasarım Şekil-19’da gösterilmiştir.

Telefon ID

Personel ID

Satışlar

**Şekil-19** Satışlar ER şeması

### Personeller Varlık Kümesi

Personeller varlık kümesi, Personel bilgilerini kayıt altına almak için oluşturulmuştur. Buna göre nitelikleri aşağıdaki gibi belirtilmiş olup **Personel ID** anahtar nitelik olarak kullanılmıştır.

* Personel ID
* Personel Adı
* Personel Çalıştığı Şube
* Personel Topma Satış
* Personel Başlama Tarihi
* Personel Toplam Satış Adeti

Tasarım Şekil-20’de gösterilmiştir.

Personel Adı

Personel Çalıştığı Şube

Personeller

**Şekil-20** Personeller ER şeması

### Müşteriler Varlık Kümesi

Müşteriler varlık kümesi, Müşteri bilgilerini kayıt altına almak için oluşturulmuştur. Buna göre nitelikleri aşağıdaki gibi belirtilmiş olup **Müşteri ID** anahtar nitelik olarak kullanılmıştır.

* Müşteri ID
* Müşteri Adı
* Müşteri Telefon No.
* Müşteri Tc
* Müşteri Adres
* Müşteri Bakiye
* Müşteri Toplam Harcama
* Müşteri Satın Alım Adeti

Tasarım Şekil-21’de gösterilmiştir.

Müşteri Satın Alım Adeti

Müşteri Adı

Müşteri Tc

Müşteri Bakiye

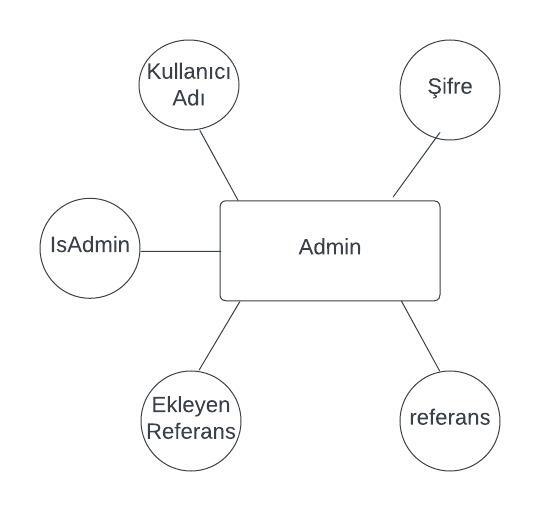
Müşteriler

**Şekil-21** Müşteriler ER şeması

### Admin Varlık Kümesi

Müşteriler varlık kümesi, Müşteri bilgilerini kayıt altına almak için oluşturulmuştur. Buna göre nitelikleri aşağıdaki gibi belirtilmiş olup **Müşteri ID** anahtar nitelik olarak kullanılmıştır.

* AdminId
* Kullanici\_Adı
* Sifre
* Referans
* Ekleyen\_referans
* IsAdmin



**Şekil-22** Admin ER şeması

## Varlık Kümeleri Arasındaki İlişki

Oluşturulan varlık kümeleri arasındaki ilişkiler bu bölümde kurulmuştur.

### Telefonlar- Nitelikler İlişkisi

Telefonların belli özellikleri yabancı anahtarlar kullanılarak başka tablolardan alınacaktır. Her telefonun tek bir markası olabilir ama aynı markadan birden fazla telefon da oladilir. Bu sebeple Telefon varlık kümeleri ile RamBellek, Ekran boyutu gibi nitelik belirten varlık kümeleri arasında bire-çok (1 : N) ilişkisi vardır. Bu ilişki gereği Nitelik belirten tabloların birincil anahtarı olan Ram Beller ID, Ekran Boyutu ID gibi anahtarlar ; telefonlar tablosuna yabancı anahtar olarak eklenmiştir. Oluşturulan ilişki Şekil-22’de gösterilmiştir.

Ram Bellek

Telefonlar

**1**

BELİRTİR

**N**

**Şekil-22** Üretici Firma İlaç İlişkisi

### Telefonlar-Satışlar İlişkisi

Bir telefon birden fazla siparişte bulunabilir ama bir siparişte de birden çok telefon olabilir. Bu nedenle Telefonlar varlık kümesi ile Satışlar varlık kümesi arasında çoktan-çoğa (N : N) ilişkisi vardır. Bu ilişki (Şekil-23) neticesinde Satış tablosunda, Telefon ID yabancı anahtar olarak yer almıştır.

Telefon ID

Telefonlar

VARDIR

**N**

Satışlar

**N**

**Şekil-23** Telefonlar Satışlar İlişkisi

### Satışlar-Şubeler İlişkisi

Her satış sadece bir şubede olabilir, şubelerde ise birden fazla satış gerçekleştirilebilir. Bu nedenle Satışlar Varlık Kümesi ile Şubeler Varlık Kümesi arasında çoktan-çoğa (N: N) ilişki vardır. Bu ilişki (Şekil-24) neticesinde Satış tablosunda, Şube ID yabancı anahtar olarak yer almıştır.

Telefon ID

Şubeler

SATILIR

**1**

Satışlar

**N**

**Şekil-24** Satışlar-Şubeler İlişkisi

### Satışlar-Personeller İlişkisi

Bir personel birden çok satış yapabilir ama bir satışı ancak bir personel pazarlayabilir. Bu nedenle Satışlar varlık kümesi ile Personeller varlık kümesi arasında bire-çok (1 : N) ilişki vardır. Bu ilişki (Şekil-25) neticesinde Satış tablosunda, Personel ID yabancı anahtar olarak yer almıştır.

Telefon ID

Personeller

SATAR

**1**

Satışlar

**N**

**Şekil-25** Satışlar-Personel İlişkisi

### Satışlar-Müşteriler İlişkisi

Bir müşteri birden çok alışveriş yapabilir ama satışı ancak bir müşteri alabilir. Bu nedenle Satışlar varlık kümesi ile Müşteriler varlık kümesi arasında bire-çok (1 : N) ilişki vardır. Bu ilişki (Şekil-26) neticesinde Satış tablosunda, Müşteri ID yabancı anahtar olarak yer almıştır.

Telefon ID

Müşteriler

ALIR

**1**

Satışlar

**N**

**Şekil-26** Satışlar-Müşteriler İlişkisi

# MANTIKSAL TASARIM

Daha önceki bölüm olan kavramsal tasarımda varlık kümeleri tanımlanmış ve aralarındaki ilişkiler açıklanmıştır. Bu bölümde, oluşturulan kavramsal tasarıma uygun olarak mantıksal tasarım gerçekleştirilmiştir.

## Telefonlar Tablosu

* Telefon ID **birincil anahtardır.**
* Ram Bellek ID, Telefonlar varlık kümesi ile Ram Bellek varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Dahili Depolama ID, Telefonlar varlık kümesi ile Dahili Depolama varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Ekran Boyutu ID, Telefonlar varlık kümesi ile Ekran Boyutu varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Marka ID, Telefonlar varlık kümesi ile Markalar varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Batarya Kapasitesi ID, Telefonlar varlık kümesi ile Batarya Kapasitesi varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Kamera Çözünürlüğü ID, Telefonlar varlık kümesi ile Kamera Çözünürlüğü varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Ön Kamera Çözünürlüğü ID, Telefonlar varlık kümesi ile Ön Kamera Çözünürlüğü varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Ekran Yenileme Hızı ID, Telefonlar varlık kümesi ile Ekran Yenileme Hızı varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Desteklenen Frekans ID, Telefonlar varlık kümesi ile Desteklenen Frekans varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Hat Sayısı ID, Telefonlar varlık kümesi ile Hat Sayısı varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Hızlı Şarj Desteği ID, Telefonlar varlık kümesi ile Hızlı Şarj Desteği varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Parmak İzi Okuyucu ID, Telefonlar varlık kümesi ile Parmak İzi Okuyucu varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Renk ID, Telefonlar varlık kümesi ile Renkler varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Kablosuz Şarj ID, Telefonlar varlık kümesi ile Kablosuz Şarj varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Hafıza Kartı Desteği ID, Telefonlar varlık kümesi ile Hafıza Kartı Desteği varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* İşletim Sistemi ID, Telefonlar varlık kümesi ile İşletim Sistemi varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**
* Çıkış Yılı ID, Telefonlar varlık kümesi ile Çıkış Yılı varlık kümesi arasındaki ilişkiden dolayı **yabancı anahtardır.**

Bu koşullar göz önünde bulundurularak oluşturulan tablo, Tablo-1’de gösterilmiştir.

**Tablo-1** Telefonlar Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Telefonlar** | | |
| TelefonId (Birincil Anahtar) | int | Not Null |
| TelefonAdi | nvarchar(50) | Not Null |
| OrnekResim | image | Null |
| RamBellekID (Yabancı Anahtar) | int | Not Null |
| DahiliDepolamaID (Yabancı Anahtar) | int | Not Null |
| EkranBoyutuID (Yabancı Anahtar) | int | Not Null |
| MarkaID (Yabancı Anahtar) | int | Not Null |
| BataryaKapasitesiID (Yabancı Anahtar) | int | Not Null |
| KameraCozunurluguID (Yabancı Anahtar) | int | Not Null |
| OnKameraCozunurluguID (Yabancı Anahtar) | int | Not Null |
| EkranYenilemeHiziID (Yabancı Anahtar) | int | Not Null |
| DesteklenenFrekansID (Yabancı Anahtar) | int | Not Null |
| HatSayisiID (Yabancı Anahtar) | int | Not Null |
| HizliSarjDestegiID (Yabancı Anahtar) | int | Not Null |
| ParmakIziOkuyucuID (Yabancı Anahtar) | int | Not Null |
| RenkID (Yabancı Anahtar) | int | Not Null |
| KablosuzSarjID (Yabancı Anahtar) | int | Not Null |
| HafizaKartiDestegiID (Yabancı Anahtar) | int | Not Null |
| IsletimSistemiID (Yabancı Anahtar) | int | Not Null |
| CikisYiliID (Yabancı Anahtar) | int | Not Null |
| GarantiSuresi | nvarchar(50) | Not Null |
| StokAdeti | int | Not Null |
| Fiyat | int | Not Null |

## Ram Bellek Tablosu

* Ram Bellek ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-2’de, oluşturulan tablo gösterilmiştir.

**Tablo-2** Ram Bellek Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Ram\_Bellek** | | |
| RamBellekId (Birincil Anahtar) | int | Not Null |
| RamBellek | nvarchar(10) | Not Null |

## Dahili Depolama Tablosu

* Dahili Depolama ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-3’de, oluşturulan tablo gösterilmiştir.

**Tablo-3** Dahili Depolama Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Dahili\_Depolama** | | |
| DahiliDepolamaId (Birincil Anahtar) | int | Not Null |
| DahiliDepolama | nvarchar(10) | Not Null |

## Ekran Boyutu Tablosu

* Ekran Boyutu ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-4’de, oluşturulan tablo gösterilmiştir.

**Tablo-4** Ekran Boyutu Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Ekran Boyutu** | | |
| EkranBoyutuId (Birincil Anahtar) | int | Not Null |
| EkranBoyutu | nvarchar(10) | Not Null |

## Markalar Tablosu

* Marka ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-5’de, oluşturulan tablo gösterilmiştir.

**Tablo-5** Ram Bellek Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Markalar** | | |
| MarkaId (Birincil Anahtar) | int | Not Null |
| Marka | nvarchar(20) | Not Null |

## Batarya Kapasitesi Tablosu

* Batarya Kapasitesi ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-6’da, oluşturulan tablo gösterilmiştir.

**Tablo-6** Batarya Kapasitesi Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Batarya\_Kapasitesi** | | |
| BataryaKapasitesiId (Birincil Anahtar) | int | Not Null |
| Batarya Kapasitesi | nvarchar(10) | Not Null |

## Kamera Çözünürlüğü Tablosu

* Kamera Çözünürlüğü ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-7’de, oluşturulan tablo gösterilmiştir.

**Tablo-7** Kamera Çözünürlüğü Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Kamera\_Çözünürlüğü** | | |
| KameraCozunurluguId (Birincil Anahtar) | int | Not Null |
| KameraCozunurlugu | nvarchar(10) | Not Null |

## Ön Kamera Çözünürlüğü Tablosu

* Ön Kamera Çözünürlüğü ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-8’de, oluşturulan tablo gösterilmiştir.

**Tablo-8** Ön Kamera Çözünürlüğü Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Ön\_Kamera\_Çözünürlüğü** | | |
| OnKameraCozunurluguId (Birincil Anahtar) | int | Not Null |
| OnKameraCozunurlugu | nvarchar(10) | Not Null |

## Ekran Yenileme Hızı Tablosu

* Ekran Yenileme Hızı ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-9’da, oluşturulan tablo gösterilmiştir.

**Tablo-9** Ekran Yenileme Hızı Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Ekran\_Yenileme\_Hizi** | | |
| EkranYenilemeHızıId (Birincil Anahtar) | int | Not Null |
| Ekran Yenileme Hızı | nvarchar(10) | Not Null |

## 3.10. Desteklenen Frekans Tablosu

* Desteklenen Frekans ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-10’da, oluşturulan tablo gösterilmiştir.

**Tablo-10** Desteklenen Frekans Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Desteklenen\_Frekans** | | |
| DesteklenenFrekansId (Birincil Anahtar) | int | Not Null |
| DesteklenenFrekans | nvarchar(10) | Not Null |

## 3.11. Hat Sayısı Tablosu

* Hat Sayısı ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-11’de, oluşturulan tablo gösterilmiştir.

**Tablo-11** Hat Sayısı Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Hat\_Sayısı** | | |
| HatSayisiId (Birincil Anahtar) | int | Not Null |
| HatSayisi | nvarchar(10) | Not Null |

## 3.12. Hızlı Şarj Desteği Tablosu

* Hızlı Şarj Desteği ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-12’da, oluşturulan tablo gösterilmiştir.

**Tablo-12** Hızlı Şarj Desteği Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Hızlı\_Şarj\_Desteği** | | |
| HizliSarjDestegiId (Birincil Anahtar) | int | Not Null |
| HizliSarjDestegi | nvarchar(10) | Not Null |

**3.13. Parmak İzi Okuyucu Tablosu**

* Parmak İzi Okuyucu ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-13’te, oluşturulan tablo gösterilmiştir.

**Tablo-13** Parmak İzi Okuyucu Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Parmak\_İzi\_Okuyucu** | | |
| ParmakIziOkuyucuId (Birincil Anahtar) | int | Not Null |
| ParmakIziOkuyucu | nvarchar(50) | Not Null |

**3.14. Renkler Tablosu**

* Renk ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-14’te, oluşturulan tablo gösterilmiştir.

**Tablo-14** Renk Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Renk** | | |
| RenkId (Birincil Anahtar) | int | Not Null |
| Renk | nvarchar(50) | Not Null |

**3.15. Kablosuz Şarj Tablosu**

* Kablosuz Şarj Desteği ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-15’de, oluşturulan tablo gösterilmiştir.

**Tablo-15** Kablosuz Şarj Desteği

|  |  |  |
| --- | --- | --- |
| **Table\_Kablosuz\_Şarj\_Desteği** | | |
| KablosuzSarjId (Birincil Anahtar) | int | Not Null |
| KablosuzSarj | nvarchar(10) | Not Null |

**3.16. Hafıza Kartı Desteği Tablosu**

* Hafıza Kartı Desteği ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-16’da, oluşturulan tablo gösterilmiştir.

**Tablo-16** Hafıza Kartı Desteği Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Hafıza\_Kartı\_Desteği** | | |
| HafizaKartiDestegiId (Birincil Anahtar) | int | Not Null |
| HafizaKartiDestegi | nvarchar(10) | Not Null |

**3.17. İşletim Sistemi Tablosu**

* İşletim Sistemi ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-17’de, oluşturulan tablo gösterilmiştir.

**Tablo-17** İşletim Sistemi Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_İşletim\_Sistemi** | | |
| IsletimSistemiId (Birincil Anahtar) | int | Not Null |
| IsletimSistemi | nvarchar(50) | Not Null |

**3.18. Çıkış Yılı Tablosu**

* Çıkış Yılı ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Tablo-18’de, oluşturulan tablo gösterilmiştir.

**Tablo-18** Çıkış Yılı Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Çıkış\_Yılı** | | |
| CikisYiliId (Birincil Anahtar) | int | Not Null |
| CikisYili | int | Not Null |

## 3.19. Satışlar Tablosu

* Satış ID **birincil anahtardır.**
* Telefon ID, tabloda **yabancı anahtar** olarak bulunur.
* Personel ID, tabloda **yabancı anahtar** olarak bulunur.
* Müşteri ID, tabloda **yabancı anahtar** olarak bulunur.
* Şube ID, tabloda **yabancı anahtar** olarak bulunur.

Oluşturulan Satışlar Tablosu, aşağıda (Tablo-19) verilmiştir.

**Tablo-19** Satışlar Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Satışlar** | | |
| SatışId (Birincil Anahtar) | int | Not Null |
| TelefonId (Yabancı Anahtar) | int | Not Null |
| PersonelId (Yabancı Anahtar) | int | Not Null |
| MusterilId (Yabancı Anahtar) | int | Not Null |
| SubeId (Yabancı Anahtar) | int | Not Null |
| UrunAdeti | int | Not Null |

## 3.20. Şubeler Tablosu

* Şube ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Oluşturulan Satışlar Tablosu, aşağıda (Tablo-20) verilmiştir.

**Tablo-20** Satışlar Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Şubeler** | | |
| SubeId (Birincil Anahtar) | int | Not Null |
| SubeAdi | nvarchar(50) | Not Null |
| SubeAdres | nvarchar(50) | Not Null |
| SubePersonelSayisi | int | Not Null |
| SubeToplamSatis | Nvarchar(20) | Not Null |
| SubeToplamSatısAdeti | int | Not Null |
| SubeAktif | Bit | Default(1) |

## 3.21. Personeller Tablosu

* Personel ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Oluşturulan Satışlar Tablosu, aşağıda (Tablo-21) verilmiştir.

**Tablo-21** Satışlar Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Personeller** | | |
| PersonelId (Birincil Anahtar) | int | Not Null |
| PersonelAdi | nvarchar(50) | Not Null |
| PersonelCalistigiSubeId | int | Not Null |
| PersonelToplamSatis | int | Default(0) |
| PersonelBaslamaTarihi | date | **Default(getDate)** |
| PersonelToplamSatisAdeti | int | Default(0) |

## 3.22. Müşteriler Tablosu

* Müşteri ID **birincil anahtardır.**
* Yabancı anahtar yoktur.

Oluşturulan Satışlar Tablosu, aşağıda (Tablo-22) verilmiştir.

**Tablo-22** Satışlar Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Müşteriler** | | |
| MusteriId (Birincil Anahtar) | int | Not Null |
| MusteriAdi | nvarchar(50) | Not Null |
| MusteriTelefonNo | nchar(11) | Not Null |
| MusteriTc | nchar(11) | Not Null |
| MusteriAdres | nvarchar(50) | Not Null |
| MusteriBakiye | int | Default(0) |
| MusteriToplamHarcama | int | Default(0) |
| MusteriSatinAlimAdeti | int | Not Null |

3.23 Admin Tablosu

* AdminID **birincil anahtardır.**
* Yabancı anahtar yoktur.

**Tablo-23** Admin Tablosu

|  |  |  |
| --- | --- | --- |
| **Table\_Admin** | | |
| AdminId (Birincil Anahtar) | int | Not Null |
| Kullanici\_adi | nvarchar(20) | Not Null |
| sifre | nchar(20) | Not Null |
| referans | nchar(10) | Not Null |
| Ekleyen\_referans | nvarchar(10) | Not Null |
| Isadmin | Bit | Default(0) |

# ŞEMATİK FİLTRELEME

Bu bölümde, hazırlanan yapı çeşitli ölçülere göre filtrelenmiştir ve tasarlanan yapının istenilen sonuçlara ulaşılıp ulaşılamayacağı belirlenmiştir.Bu bölümde, hazırlanan yapı çeşitli ölçülere göre filtrelenmiştir ve tasarlanan yapının istenilen sonuçlara ulaşılıp ulaşılmayacağı belirlenmiştir.

## Telefonlar Veri Tablosu

Telefonlar Firma tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılmıştır. Çok fazla veri olduğu için Telefonlar veri tablosu 2 tabloyla gösterilmiştir (Tablo-23 ve Tablo-24).

**Tablo-23** Telefonlar Veri Tablosu

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Telefon\_ID** | **Telefon\_Adı** | **Ornek\_Resim** | **Ram\_Bellek\_ID** | **Dahili\_Depolama\_ID** | **Ekran\_Boyutu\_ID** | **Marka\_ID** | **Batarya\_Kapasitesi\_ID** | **Kamera\_Çözünürlüğü\_ID** | **Ön\_Kamera\_Çözünürlüğü\_ID** | **Ekran\_Yenileme\_Hızı\_ID** | **Desteklenen\_Frekans\_ID** |
| 1 | Iphone12 Pro Max | Null | 8 | 17 | 1 | 1 | 19 | 17 | 12 | 12 | 5 |
| 2 | Galaxy s21 | Null | 6 | 14 | 2 | 2 | 13 | 16 | 10 | 12 | 5 |
| 3 | Iphone 12 Pro | Null | 8 | 15 | 3 | 1 | 1 | 16 | 12 | 12 | 5 |
| 4 | Galaxy A5 | Null | 6 | 14 | 4 | 2 | 9 | 15 | 10 | 12 | 5 |
| 5 | Reeder P13 | Null | 4 | 13 | 5 | 3 | 12 | 8 | 6 | 6 | 5 |
| 6 | Nokia 3310 | Null | 1 | 4 | 6 | 4 | 16 | 4 | 4 | 6 | 5 |
| 7 | Xiaomi Mi 11 | Null | 8 | 13 | 7 | 5 | 3 | 16 | 12 | 12 | 5 |
| 8 | Huawei 10 Pro | Null | 6 | 15 | 8 | 6 | 4 | 17 | 15 | 12 | 5 |
| 9 | Galaxy 04A | Null | 6 | 15 | 9 | 2 | 5 | 16 | 4 | 6 | 5 |
| 10 | HTC 10S | Null | 6 | 13 | 10 | 7 | 6 | 8 | 6 | 6 | 5 |
| 11 | İphone 11 | Null | 6 | 14 | 11 | 1 | 2 | 15 | 12 | 6 | 5 |
| 12 | Samsung s20 | Null | 6 | 13 | 12 | 2 | 7 | 17 | 12 | 6 | 5 |
| 13 | Oppo A5 | Null | 4 | 13 | 13 | 8 | 14 | 12 | 6 | 6 | 5 |
| 14 | Reeder S19 | Null | 2 | 15 | 14 | 3 | 2 | 15 | 8 | 6 | 5 |
| 15 | İphone 13Pro Max | Null | 10 | 16 | 15 | 2 | 8 | 16 | 10 | 12 | 5 |
| 16 | Galaxy A23 | Null | 8 | 14 | 16 | 2 | 3 | 15 | 10 | 12 | 5 |
| 17 | General  M13 | Null | 6 | 14 | 17 | 9 | 20 | 17 | 12 | 6 | 5 |
| 18 | Z Flip | Null | 4 | 12 | 18 | 2 | 2 | 16 | 10 | 12 | 5 |
| 19 | Galaxy S22 | Null | 6 | 14 | 19 | 2 | 15 | 15 | 12 | 12 | 5 |
| 20 | İphone X | Null | 6 | 13 | 20 | 1 | 2 | 16 | 8 | 6 | 5 |

**Tablo-24** Telefonlar Veri Tablosu devamı

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hat Sayısı** | **Hızlı Sarj Destegi** | **ParmakIziOkuyucu** | **Renk** | **KablosuzSarj** | **Hafıza Kartı Destegi** | **Isletim sistemi** | **Cıkıs Yılı** | **Garanti Suresi** | **Stok Adeti** | **Siparis Adeti** | **Fiyat** |
| 2 | 1 | 1 | 1 | 1 | 1 | 12 | 18 | 1 yıl | 54 | 5 | 63290 |
| 2 | 2 | 2 | 2 | 2 | 2 | 18 | 17 | 2 yıl | 48 | 41 | 18400 |
| 2 | 3 | 3 | 3 | 3 | 3 | 11 | 16 | 1 yıl | 27 | 3 | 10739 |
| 2 | 4 | 4 | 4 | 4 | 4 | 17 | 17 | 2 yıl | 0 | 0 | 18200 |
| 2 | 5 | 5 | 5 | 5 | 5 | 17 | 18 | 2 yıl | 10 | 5 | 9500 |
| 1 | 6 | 6 | 6 | 6 | 6 | 20 | 1 | 8 yıl | 15 | 1 | 1500 |
| 2 | 7 | 7 | 7 | 7 | 7 | 16 | 16 | 2 yıl | 20 | 6 | 6320 |
| 2 | 8 | 8 | 8 | 8 | 8 | 18 | 15 | 2 yıl | 0 | 8 | 4205 |
| 2 | 9 | 9 | 9 | 9 | 9 | 18 | 18 | 2 yıl | 13 | 9 | 3652 |
| 2 | 10 | 10 | 10 | 10 | 10 | 17 | 17 | 2 yıl | 5 | 0 | 8889 |
| 2 | 11 | 11 | 11 | 11 | 11 | 10 | 18 | 2 yıl | 6 | 3 | 3265 |
| 2 | 12 | 12 | 12 | 12 | 12 | 18 | 16 | 2 yıl | 19 | 1 | 1546 |
| 2 | 13 | 13 | 13 | 13 | 13 | 16 | 14 | 2 yıl | 26 | 0 | 3652 |
| 2 | 14 | 14 | 14 | 14 | 14 | 18 | 18 | 2 yıl | 32 | 3 | 2487 |
| 2 | 15 | 15 | 15 | 15 | 15 | 9 | 15 | 2 yıl | 52 | 2 | 2365 |
| 2 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 2 yıl | 40 | 1 | 9874 |
| 2 | 17 | 17 | 17 | 17 | 17 | 17 | 18 | 2 yıl | 13 | 5 | 5586 |
| 2 | 18 | 18 | 18 | 18 | 18 | 18 | 17 | 2 yıl | 20 | 3 | 5412 |
| 2 | 19 | 19 | 19 | 19 | 19 | 17 | 18 | 2 yıl | 9 | 8 | 5525 |
| 2 | 20 | 20 | 20 | 20 | 20 | 8 | 15 | 1 yıl | 1 | 0 | 2025 |

## Ram Bellek Veri Tablosu

Ram Bellek tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-25’de gösterilmiştir.

**Tablo-25** Ram Bellek Veri Tablosu

|  |  |
| --- | --- |
| **Ram\_Bellek\_ID** | **Ram\_Bellek** |
| 1 | 1 GB |
| 2 | 2 GB |
| 3 | 3 GB |
| 4 | 4 GB |
| 5 | 5 GB |
| 6 | 6 GB |
| 7 | 7 GB |
| 8 | 8 GB |
| 9 | 9 GB |
| 10 | 10 GB |
| 11 | 11 GB |
| 12 | 12 GB |
| 13 | 13 GB |
| 14 | 14 GB |
| 15 | 15 GB |
| 16 | 16 GB |
| 17 | 17 GB |
| 18 | 18 GB |
| 19 | 19 GB |
| 20 | 20 GB |

## Dahili Depolama Veri Tablosu

Dahili Depolama tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-26’de gösterilmiştir.

**Tablo-26** Dahili Depolama Veri Tablosu

|  |  |
| --- | --- |
| **Dahili\_Depolama\_ID** | **Dahili\_Depolama** |
| 1 | 1 GB |
| 2 | 2 GB |
| 3 | 3 GB |
| 4 | 4 GB |
| 5 | 5 GB |
| 6 | 6 GB |
| 7 | 7 GB |
| 8 | 8 GB |
| 9 | 9 GB |
| 10 | 10 GB |
| 11 | 16 GB |
| 12 | 32 GB |
| 13 | 64 GB |
| 14 | 128 GB |
| 15 | 256 GB |
| 16 | 512 GB |
| 17 | 1 TB |
| 18 | 2 TB |
| 19 | 4 TB |
| 20 | 8 TB |

## Ekran Boyutu Veri Tablosu

Ekran Boyutu tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-27’de gösterilmiştir.

**Tablo-27** Ekran Boyutu Veri Tablosu

|  |  |
| --- | --- |
| **Ekran\_Boyutu\_ID** | **Ekran\_Boyutu** |
| 1 | 5 inc |
| 2 | 5.1 inc |
| 3 | 5.2 inc |
| 4 | 5.3 inc |
| 5 | 5.4 inc |
| 6 | 5.5 inc |
| 7 | 5.6 inc |
| 8 | 5.7 inc |
| 9 | 5.8 inc |
| 10 | 5.9 inc |
| 11 | 6 inc |
| 12 | 6.1 inc |
| 13 | 6.2 inc |
| 14 | 6.3 inc |
| 15 | 6.4 inc |
| 16 | 6.5 inc |
| 17 | 6.6 inc |
| 18 | 6.7 inc |
| 19 | 6.8 inc |
| 20 | 6.9 inc |

## Markalar Veri Tablosu

Markalar tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-28’de gösterilmiştir.

**Tablo-28** Markalar Veri Tablosu

|  |  |
| --- | --- |
| **Marka\_ID** | **Marka** |
| 1 | Apple |
| 2 | Samsung |
| 3 | Reeder |
| 4 | Nokia |
| 5 | Xiaomi |
| 6 | Huawei |
| 7 | Htc |
| 8 | Oppo |
| 9 | General Mobile |
| 10 | Honor |
| 11 | Realme |
| 12 | Turkcell |
| 13 | TurkTelekom |
| 14 | Sony |
| 15 | Philips |
| 16 | OnePlus |
| 17 | LG |
| 18 | Meizu |
| 19 | Lenovo |
| 20 | Asus |

## Batarya Kapasitesi Veri Tablosu

Batarya Kapasitesi tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-29’da gösterilmiştir.

**Tablo-29** Batarya Kapasitesi Veri Tablosu

|  |  |
| --- | --- |
| **Batarya\_Kapasitesi\_ID** | **Batarya\_Kapasitesi** |
| 1 | 3000 mAh |
| 2 | 3100 mAh |
| 3 | 3200 mAh |
| 4 | 3300 mAh |
| 5 | 3400 mAh |
| 6 | 3500 mAh |
| 7 | 3600 mAh |
| 8 | 3700 mAh |
| 9 | 3800 mAh |
| 10 | 3900 mAh |
| 11 | 4000 mAh |
| 12 | 4100 mAh |
| 13 | 4200 mAh |
| 14 | 4300 mAh |
| 15 | 4400 mAh |
| 16 | 4500 mAh |
| 17 | 4600 mAh |
| 18 | 4700 mAh |
| 19 | 4800 mAh |
| 20 | 4900 mAh |

## Kamera Çözünürlüğü Veri Tablosu

Kamera Çözünürlüğü tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-30’da gösterilmiştir.

**Tablo-30** Kamera Çözünürlüğü Veri Tablosu

|  |  |
| --- | --- |
| **Kamera\_Çözünürlüğü\_ID** | **Kamera\_Çözünürlüğü** |
| 1 | 1 MP |
| 2 | 2 MP |
| 3 | 3 MP |
| 4 | 4 MP |
| 5 | 5 MP |
| 6 | 6 MP |
| 7 | 7 MP |
| 8 | 8 MP |
| 9 | 9 MP |
| 10 | 10 MP |
| 11 | 11 MP |
| 12 | 12 MP |
| 13 | 16 MP |
| 14 | 24 MP |
| 15 | 32 MP |
| 16 | 48 MP |
| 17 | 64 MP |
| 18 | 96 MP |
| 19 | 128 MP |
| 20 | 192 MP |

## Ön Kamera Çözünürlüğü Veri Tablosu

Ön Kamera Çözünürlüğü tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-31’de gösterilmiştir.

**Tablo-31** Ön Kamera Çözünürlüğü Veri Tablosu

|  |  |
| --- | --- |
| **Ön\_Kamera\_Çözünürlüğü\_ID** | **Ön\_Kamera\_Çözünürlüğü** |
| 1 | 1 MP |
| 2 | 2 MP |
| 3 | 3 MP |
| 4 | 4 MP |
| 5 | 5 MP |
| 6 | 6 MP |
| 7 | 7 MP |
| 8 | 8 MP |
| 9 | 9 MP |
| 10 | 10 MP |
| 11 | 11 MP |
| 12 | 12 MP |
| 13 | 16 MP |
| 14 | 24 MP |
| 15 | 32 MP |
| 16 | 48 MP |
| 17 | 64 MP |
| 18 | 96 MP |
| 19 | 128 MP |
| 20 | 192 MP |

## Ekran Yenileme Hızı Veri Tablosu

Ekran Yenileme Hızı tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-32’de gösterilmiştir.

**Tablo-32** Ekran Yenileme Hızı Veri Tablosu

|  |  |
| --- | --- |
| **Ekran\_Yenileme\_ID** | **Ekran\_Yenileme** |
| 1 | 10 Hz |
| 2 | 20 Hz |
| 3 | 30 Hz |
| 4 | 40 Hz |
| 5 | 50 Hz |
| 6 | 60 Hz |
| 7 | 70 Hz |
| 8 | 80 Hz |
| 9 | 90 Hz |
| 10 | 100 Hz |
| 11 | 110 Hz |
| 12 | 120 Hz |
| 13 | 130 Hz |
| 14 | 140 Hz |
| 15 | 150 Hz |
| 16 | 160 Hz |
| 17 | 170 Hz |
| 18 | 180 Hz |
| 19 | 190 Hz |
| 20 | 200 Hz |

## 4.10. Desteklenen Frekans Veri Tablosu

Desteklenen Frekans tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-33’de gösterilmiştir.

**Tablo-33** Desteklenen Frekans Veri Tablosu

|  |  |
| --- | --- |
| **Desteklenen\_Frekans\_ID** | **Desteklenen\_Frekans** |
| 1 | 1 G |
| 2 | 2 G |
| 3 | 3 G |
| 4 | 4 G |
| 5 | 5 G |
| 6 | 6 G |
| 7 | 7 G |
| 8 | 8 G |
| 9 | 9 G |
| 10 | 10 G |
| 11 | 11 G |
| 12 | 12 G |
| 13 | 13 G |
| 14 | 14 G |
| 15 | 15 G |
| 16 | 16 G |
| 17 | 17 G |
| 18 | 18 G |
| 19 | 19 G |
| 20 | 20 G |

## 4.11. Hat Sayısı Veri Tablosu

Hat Sayısı tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-34’te gösterilmiştir.

**Tablo-34** Hat Sayısı Veri Tablosu

|  |  |
| --- | --- |
| **Hat\_Sayısı\_ID** | **Hat\_Sayısı** |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | 10 |
| 11 | 11 |
| 12 | 12 |
| 13 | 13 |
| 14 | 14 |
| 15 | 15 |
| 16 | 16 |
| 17 | 17 |
| 18 | 18 |
| 19 | 19 |
| 20 | 20 |

## 4.12. Hızlı Şarj Desteği Veri Tablosu

Hızlı Şarj Desteği tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-35’de gösterilmiştir.

**Tablo-35** Hızlı Şarj Desteği Veri Tablosu

|  |  |
| --- | --- |
| **Hızlı\_Şarj\_Desteği\_ID** | **Hızlı\_Şarj\_Desteği** |
| 1 | 10 Wolt |
| 2 | 11 Wolt |
| 3 | 12 Wolt |
| 4 | 13 Wolt |
| 5 | 14 Wolt |
| 6 | 15 Wolt |
| 7 | 16 Wolt |
| 8 | 17 Wolt |
| 9 | 18 Wolt |
| 10 | 19 Wolt |
| 11 | 20 Wolt |
| 12 | 21 Wolt |
| 13 | 22 Wolt |
| 14 | 23 Wolt |
| 15 | 24 Wolt |
| 16 | 25 Wolt |
| 17 | 26 Wolt |
| 18 | 27 Wolt |
| 19 | 28 Wolt |
| 20 | 29 Wolt |

## 4.13. Parmak İzi Okuyucu Veri Tablosu

Parmak İzi Okuyucu tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-36’da gösterilmiştir.

**Tablo-36** Parmak İzi Okuyucu Veri Tablosu

|  |  |
| --- | --- |
| **Parmak\_İzi\_Okuyucu\_ID** | **Parmak\_İzi\_Okuyucu** |
| 1 | 0 Okuyucu |
| 2 | 1 Okuyucu |
| 3 | 2 Okuyucu |
| 4 | 3 Okuyucu |
| 5 | 4 Okuyucu |
| 6 | 5 Okuyucu |
| 7 | 6 Okuyucu |
| 8 | 7 Okuyucu |
| 9 | 8 Okuyucu |
| 10 | 9 Okuyucu |
| 11 | 10 Okuyucu |
| 12 | 11 Okuyucu |
| 13 | 12 Okuyucu |
| 14 | 13 Okuyucu |
| 15 | 14 Okuyucu |
| 16 | 15 Okuyucu |
| 17 | 16 Okuyucu |
| 18 | 17 Okuyucu |
| 19 | 18 Okuyucu |
| 20 | 19 Okuyucu |

## 4.14. Renkler Veri Tablosu

Renkler tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-37’de gösterilmiştir.

**Tablo-37** Renkler Veri Tablosu

|  |  |
| --- | --- |
| **Renk\_ID** | **Renk** |
| 1 | Kırmızı |
| 2 | Turuncu |
| 3 | Sarı |
| 4 | Yeşil |
| 5 | Mavi |
| 6 | Lacivert |
| 7 | Mor |
| 8 | Pembe |
| 9 | Altın Sarısı |
| 10 | Siyah |
| 11 | Gri |
| 12 | Beyaz |
| 13 | Kahverengi |
| 14 | Magenta |
| 15 | Cam göbeği |
| 16 | Bej |
| 17 | Bordo |
| 18 | Bakır |
| 19 | Alaca Mavi |
| 20 | Koyu Mavi |

## 4.15. Kablosuz Şarj Veri Tablosu

Kablosuz Şarj tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-38’de gösterilmiştir.

**Tablo-38** Kablosuz Şarj Veri Tablosu

|  |  |
| --- | --- |
| **Kablosuz\_Şarj\_ID** | **Kablosuz\_Şarj** |
| 1 | Yok |
| 2 | 0.1 Amper |
| 3 | 0.2 Amper |
| 4 | 0.3 Amper |
| 5 | 0.4 Amper |
| 6 | 0.5 Amper |
| 7 | 0.6 Amper |
| 8 | 0.7 Amper |
| 9 | 0.8 Amper |
| 10 | 0.9 Amper |
| 11 | 1 Amper |
| 12 | 1.1 Amper |
| 13 | 1.2 Amper |
| 14 | 1.3 Amper |
| 15 | 1.4 Amper |
| 16 | 1.5 Amper |
| 17 | 1.6 Amper |
| 18 | 1.7 Amper |
| 19 | 1.8 Amper |
| 20 | 1.9 Amper |

## 4.16. Hafıza Kart Desteği Veri Tablosu

Hafıza Kartı Desteği tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-39’de gösterilmiştir.

**Tablo-39** Hafıza Kartı Desteği Veri Tablosu

|  |  |
| --- | --- |
| **Hafıza\_Kartı\_Desteği\_ID** | **Hafıza\_Kartı\_Desteği** |
| 1 | Yok |
| 2 | 1 GB |
| 3 | 2 GB |
| 4 | 3 GB |
| 5 | 4 GB |
| 6 | 5 GB |
| 7 | 6 GB |
| 8 | 7 GB |
| 9 | 8 GB |
| 10 | 9 GB |
| 11 | 10 GB |
| 12 | 16 GB |
| 13 | 32 GB |
| 14 | 64 GB |
| 15 | 128 GB |
| 16 | 256 GB |
| 17 | 512 GB |
| 18 | 1 TB |
| 19 | 2 TB |
| 20 | 4 TB |

## 4.17. İşletim Sistemi Veri Tablosu

İşletim Sistemi tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-40’da gösterilmiştir.

**Tablo-40** İşletim Sistemi Veri Tablosu

|  |  |
| --- | --- |
| **İşletim\_Sistemi\_ID** | **İşletim\_Sistemi** |
| 1 | iOS 3 |
| 2 | iOS 4 |
| 3 | iOS 5 |
| 4 | iOS 6 |
| 5 | iOS 7 |
| 6 | iOS 8 |
| 7 | iOS 9 |
| 8 | iOS 10 |
| 9 | iOS 11 |
| 10 | iOS 12 |
| 11 | iOS 13 |
| 12 | iOS 14 |
| 13 | Android 4 |
| 14 | Android 5 |
| 15 | Android 6 |
| 16 | Android 7 |
| 17 | Android 8 |
| 18 | Android 9 |
| 19 | Windows Phone |
| 20 | BlackBerry OS |

## 4.18. Çıkış Yılı Veri Tablosu

Çıkış Yılı tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-41’de gösterilmiştir.

**Tablo-41** Çıkış Yılı Veri Tablosu

|  |  |
| --- | --- |
| **İşletim\_Sistemi\_ID** | **İşletim\_Sistemi** |
| 1 | 2005 |
| 2 | 2006 |
| 3 | 2007 |
| 4 | 2008 |
| 5 | 2009 |
| 6 | 2010 |
| 7 | 2011 |
| 8 | 2012 |
| 9 | 2013 |
| 10 | 2014 |
| 11 | 2015 |
| 12 | 2016 |
| 13 | 2017 |
| 14 | 2018 |
| 15 | 2019 |
| 16 | 2020 |
| 17 | 2021 |
| 18 | 2022 |
| 19 | 2023 |
| 20 | 2024 |

## 4.19. Satışlar Veri Tablosu

Satışlar tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-42’de gösterilmiştir.

**Tablo-42** Satışlar Veri Tablosu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Satış\_ID** | **Telefon\_ID** | **Personel\_ID** | **Müşteri\_ID** | **Urun\_Adeti** | **Şube\_Id** |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 1 | 2 |
| 3 | 3 | 3 | 3 | 1 | 3 |
| 4 | 4 | 4 | 4 | 1 | 4 |
| 5 | 5 | 5 | 5 | 1 | 5 |
| 6 | 6 | 6 | 6 | 1 | 6 |
| 7 | 7 | 7 | 7 | 1 | 7 |
| 8 | 8 | 8 | 8 | 1 | 8 |
| 9 | 9 | 9 | 9 | 1 | 9 |
| 10 | 10 | 10 | 10 | 1 | 10 |
| 11 | 11 | 11 | 11 | 1 | 11 |
| 12 | 12 | 12 | 12 | 1 | 12 |
| 13 | 13 | 13 | 13 | 1 | 13 |
| 14 | 14 | 14 | 14 | 1 | 14 |
| 15 | 15 | 15 | 15 | 1 | 15 |
| 16 | 16 | 16 | 16 | 1 | 16 |
| 17 | 17 | 17 | 17 | 1 | 17 |
| 18 | 18 | 18 | 18 | 1 | 18 |
| 19 | 19 | 19 | 19 | 1 | 19 |
| 20 | 20 | 20 | 20 | 1 | 20 |

## 4.20. Şubeler Veri Tablosu

Şubeler tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-43’te gösterilmiştir.

**Tablo-43** Şubeler Veri Tablosu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Şube\_ID** | **Şube\_Adı** | **Şube\_Adres** | **Şube\_Personel\_Sayısı** | **Şube\_Toplam\_Satış** | **Şube\_Toplam\_Satış\_Adeti** |
| 1 | Bekirbey | Çankaya/Ankara | 14 | 1.3 milyon | 260 |
| 2 | Hamidiye | Fatih/İstanbul | 23 | 3.5 milyon | 523 |
| 3 | Karsu | Osmangazi/Cankaya | 7 | 456 bin | 108 |
| 4 | Batıknet | Yenimahalle/Ankara | 2 | 356 bin | 230 |
| 5 | Dikimevi | Çankat | 15 | 125 bin | 125 |
| 6 | Söğütözü | Söğütözü/ankara | 12 | 1.3milyon | 365 |
| 7 | Eryaman | Yenimahalle/Ankara | 25 | 1.2milyon | 245 |
| 8 | Emek | Çankaya/Ankara | 12 | 3.4milyon | 456 |
| 9 | Kızılay | Çankaya/Ankara | 14 | 325bin | 253 |
| 10 | Ataç | Çankaya/Ankara | 16 | 126bin | 585 |
| 11 | Balgat | Çankaya/Ankara | 2 | 85bin | 216 |
| 12 | Cebeci | Çankaya/Ankara | 3 | 123bin | 235 |
| 13 | Çankaya | Çankaya/Ankara | 5 | 15bin | 245 |
| 14 | Çayyolu | Çankaya/Ankara | 56 | 123bin | 365 |
| 15 | Odtü | Çankaya/Ankara | 25 | 1.5milyon | 75 |
| 16 | Demet | Yenimahalle /Ankara | 26 | 36bin | 365 |
| 17 | Dikmen | Çankaya/Ankara | 23 | 26bin | 123 |
| 18 | Elmadağ | Elmadağ /Ankara | 12 | 15bin | 564 |
| 19 | Esat | Çankaya/Ankara | 24 | 255bin | 845 |
| 20 | Etlik | Keçiören/Ankara | 29 | 124bin | 626 |

## 4.21. Personeller Veri Tablosu

Personeller tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-44’te gösterilmiştir.

**Tablo-44** Personeller Veri Tablosu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Personel\_ID** | **Personel\_Adı** | **Personel\_Çalıştığı\_ŞubeId** | **Personel\_Toplam\_Satış** | **Personel\_Başlama\_Tarihi** | **Personel\_Toplam\_Satış\_Adeti** |
| 1 | Rıfat Yılmaz | 1 | 1 | 2003-12-21 | 1 |
| 2 | Hakkı Deveci | 2 | 2 | 2005-12-31 | 2 |
| 3 | Gokberk Pala | 3 | 3 | 2004-01-06 | 3 |
| 4 | Hayri tekin | 4 | 4 | 2006-12-15 | 4 |
| 5 | Hakkı çelik | 5 | 5 | 2003-12-21 | 5 |
| 6 | Haydar güllü | 6 | 6 | 2005-05-21 | 6 |
| 7 | Burak mutlu | 7 | 7 | 2004-03-21 | 7 |
| 8 | Baran koç | 8 | 8 | 2002-11-21 | 8 |
| 9 | Pelinsu ünal | 9 | 9 | 2004-01-01 | 9 |
| 10 | Berru özelik | 10 | 10 | 2002-03-14 | 10 |
| 11 | Derya su | 11 | 11 | 2003-05-30 | 11 |
| 12 | Irmak aygör | 12 | 12 | 2006-01-29 | 12 |
| 13 | Aslan güler | 13 | 13 | 2006-09-17 | 13 |
| 14 | Şeyma su | 14 | 14 | 2003-07-24 | 14 |
| 15 | Ezgi tekin | 15 | 15 | 2003-05-26 | 15 |
| 16 | Hanım ırmak | 16 | 16 | 2006-03-19 | 16 |
| 17 | Zeynep barun | 17 | 17 | 2003-02-01 | 17 |
| 18 | Yunus kaplan | 18 | 18 | 2003-10-05 | 18 |
| 19 | Emir ışık | 19 | 19 | 2006-01-27 | 19 |
| 20 | Emre şanlı | 20 | 20 | 2003-06-11 | 20 |

## 4.22. Müşteriler Veri Tablosu

Müşteriler tablosu mantıksal tasarıma uygun oluşturulmuştur. 20 adet veri girişi yapılarak Tablo-45’de gösterilmiştir.

**Tablo-45** Müşterler Veri Tablosu

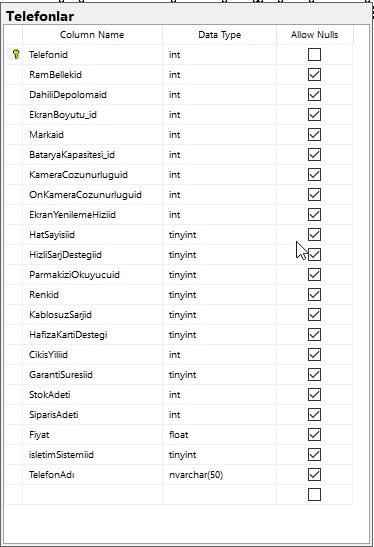
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Müşteri\_ID** | **Müşteri\_Adı** | **Müşteri\_Tel** | **Müşteri\_Tc** | **Müşteri\_Adres** | **Müşteri\_Bakiye** | **Müşteri\_Toplam\_Harcama** | **Müşteri\_Satın\_Alım\_Adeti** |
| 1 | Mehmet Soylu | 05764831598 | 2148965782 | Adana | 1254 | 24658 | 3 |
| 2 | Fatma Taşlı | 05483165729 | 6584765165 | Karaman | 9805 | 561 | 1 |
| 3 | Hidayet Gedik | 05458255672 | 6516584855 | İzmir | 24658 | 76582 | 17 |
| 4 | Metin Türker | 05987642596 | 8491584165 | Ankara | 15657 | 5475 | 1 |
| 5 | Bülent Türkel | 05468254578 | 8956146956 | Ankara | 561 | 34682 | 8 |
| 6 | Arif Can | 05147141903 | 1891561614 | Kırıklareli | 7658219 | 8437 | 2 |
| 7 | Haydar Yılmaz | 05348765198 | 1056168452 | Bayburt | 54 | 4352 | 2 |
| 8 | Hamza Koç | 05248736514 | 5462891354 | İstanbul | 34682 | 10697 | 1 |
| 9 | Özkan Şen | 05481564823 | 2154865967 | Van | 843 | 6720 | 2 |
| 10 | Alperen Yıldırım | 05615845486 | 3465875618 | Gaziantep | 4352 | 5403 | 1 |
| 11 | Halil Güler | 05178465855 | 6489645198 | Bilecik | 1069 | 7568 | 2 |
| 12 | Cemal Gürsel | 05134681907 | 3215468495 | İstanbul | 67205 | 1867 | 1 |
| 13 | Ali Ersoy | 05684827846 | 3215648496 | Adıyaman | 5403 | 9468 | 4 |
| 14 | İlteber Kalkan | 05794561905 | 1647328954 | Gümüşhane | 75691 | 36401 | 7 |
| 15 | Kemal Şahin | 05483157982 | 2054684351 | Artvin | 18672 | 4976 | 2 |
| 16 | İsmet Tümay | 05478958765 | 4598215632 | Ankara | 2468911 | 7168 | 3 |
| 17 | Tansu Özdağ | 05132589511 | 2154365854 | İstanbul | 3640 | 6481 | 2 |
| 18 | Mesut Ecevit | 05136253223 | 4256783915 | Erzincan | 4976 | 9815 | 3 |
| 19 | Alper Dinç | 05234681652 | 2316487562 | Kırıkkale | 7168 | 7548 | 2 |
| 20 | Hüseyin Çevik | 05362451524 | 2108709501 | İstanbul | 648 | 3564 | 1 |

# FIZIKSEL TASARIM

Fiziksel tasarım, verit abani içindeki dosya organizasyonunu ve işlemlerin hızlı ve etkili bir şekilde gerçekleştirilmesi için gerekli yapıları içermektedir. Kavramsal ve mantıksal tasarımda belirlenen gereksinimlere uygunlukları denetlemek için oluşturulmuştur.

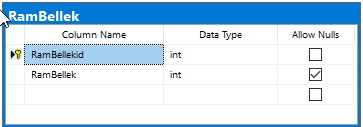
## Telefonlar Tablosu (EK-2 - Ek-3)

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Telefonlar tablosunu aşağıdaki gibidir. Veri Tabanı Kodları Veri ve Triggerları ekteki dosyada verilmiştir.



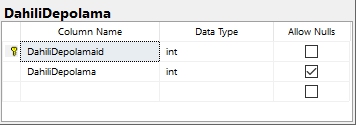
## Ram Bellek Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Ram Bellek tablosu aşağıdaki gibidir.



## Dahili Depolama Tablosu

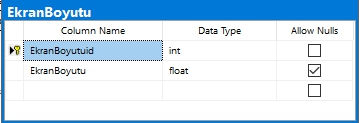
Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Dahili Depolama tablosunu oluşturmak için gereken SQL script kodları aşağıdaki gibidir.



## Ekran Boyutu Tablosu

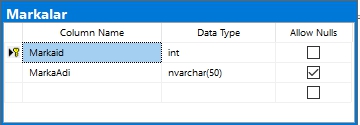
Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Ekran Boyutu

aşağıdaki gibidir.



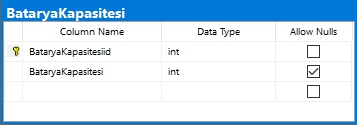
## Markalar Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Markalar Tablosu aşağıdaki gibidir.



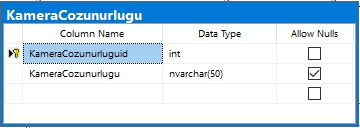
## Batarya Kapasitesi Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Batarya Kapasitesi tablosunu oluşturmak için gereken SQL script kodları aşağıdaki gibidir.



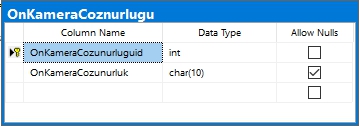
## Kamera Çözünürlüğü Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Kamera Çözünürlüğü aşağıdaki gibidir.



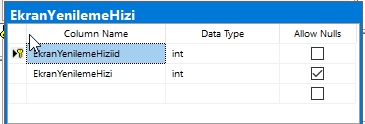
## Ön Kamera Çözünürlüğü Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Ön Kamera Çözünürlüğü tablosu aşağıdaki gibidir.



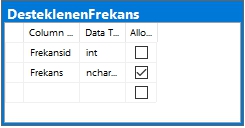
## Ekran Yenileme HızıTablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Ekran Yenileme Hızı Tablosu aşağıdaki gibidir.



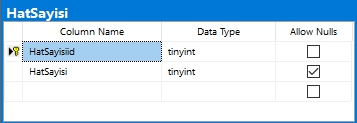
## 5.10. Desteklenen Frekans Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Desteklenen Frekans tablosu aşağıdaki gibidir.



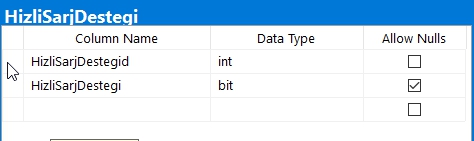
## 5.11. Hat Sayısı Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Hat Sayısı tablosu aşağıdaki gibidir.



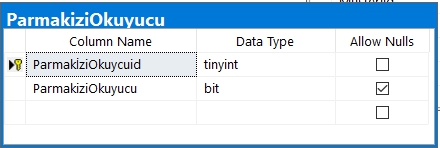
## 5.12. Hızlı Şarj Desteği Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Hızlı Şarj Desteği aşağıdaki gibidir.



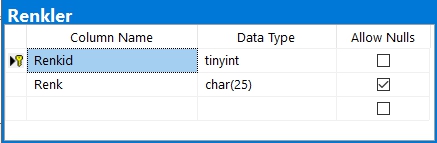
## 5.13.Parmak İzi Okuyucu Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Parmak İzi Okuyucu aşağıdaki gibidir.



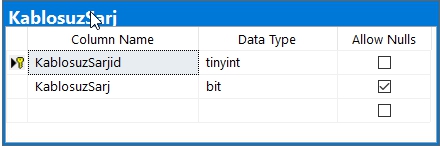
## 5.14. Renkler Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Renkler Tablosu aşağıdaki gibidir.



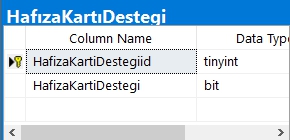
## 5.15. Kablosuz Şarj Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Kablosuz Şarj tablosu kodları aşağıdaki gibidir.



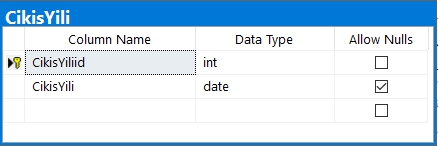
## 5.16. Hafıza Kartı Desteği Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Hafıza Kartı Desteği tablosu aşağıdaki gibidir.



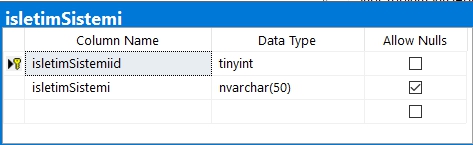
## 5.17. Çıkış Yılı Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Çıkış Yılı tablosu aşağıdaki gibidir.



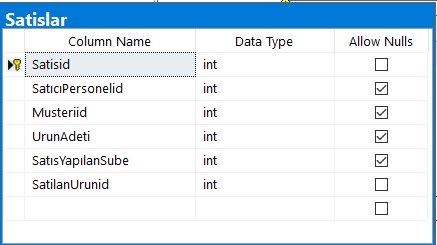
## 5.18. İşletim Sistemi Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan İşletim Sistemi tablosu aşağıdaki gibidir.



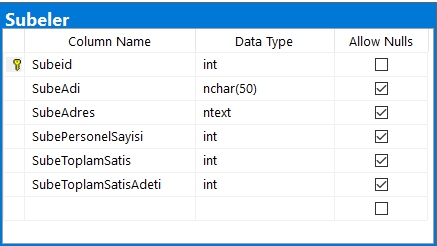
## 5.19. Satışlar Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Satışlar tablosu aşağıdaki gibidir.



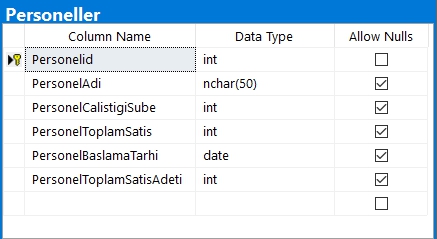
## 5.20. Şubeler Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Şubeler tablosu aşağıdaki gibidir.



## 5.21. Personeller Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Personeller tablosu aşağıdaki gibidir.



## 5.22. Müşteriler Tablosu

Kavramsal ve mantıksal tasarıma uygun olarak tasarlanan Müşteriler tablosu aşağıdaki gibidir.



# UYGULAMA

**6.0 Projede Kullanılan Kütüphanler**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

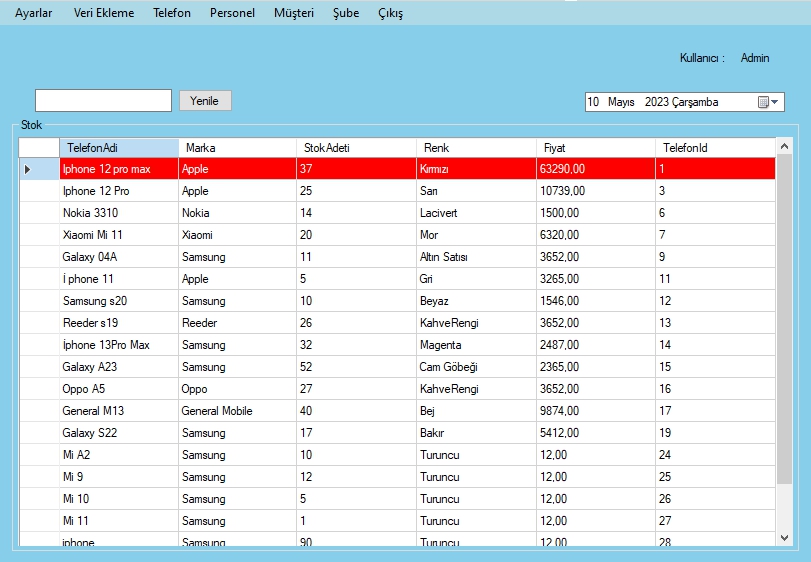
using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

**6.1 Anasayfa Arayüzü**



**Sayfa Özellikleri Temel Tanıtım:**

* **Admin Sayfasıdır.**
* Textbox ile isime göre filtreleme yapabilme
* Datagridview e DoubleClick özelliği ile Telefonların Özelliklerini Görünütleyebilme
* Ayarlar Menusu ile Temel Admin Ekleme, Cıkarma islemlerini yapablir
* Sifre Güncelleme , Bilgi güncelleme islemlerini yapabilir.
* Yedekleme ile veri tabanı yedeğini C: diznine kayıt yapabilir .

**Menu : Veri Ekleme**

* Temel Telefon özelliklerinin bulunduğu tablolalara veri ekleme işlemi yapabilir .

Veriler listelenirken sorgu aracılığı ile dönen değerlere göre Tablo isimleri menu item olarak aktarılır .

**Menu : Telefon**

* Stok Arttırma
* Telefon Ekleme

Seçenekleri ile yeni kayıt eklenebilir , resimli kayıtlar veritabanına eklenebilir .

Telefon ekleme seçeneğinin ekstra özelliği resimleri byte[] olarak dönüşüm yapmasıdır.

**Menu : Personel**

* Temel Personel İslemleri Yapılmaktadır
* Personel Performans ile belirttiğiniz tarihler arasında personel performanslarını izleyebilirsiniz.

**Menu:Musteri**

**-**Temel Musteri İslemleri yapılmaktadır.

-Bakiye güncellemesi ve diger islemler sadece adminler tarafından yapılabilmektedir.

**Menu:Sube**

**-**Temel Sube islemleri gerçekleştirilmektedir.

-Sube silinme sonrasında da silinen subeler görüntülenebilmektedir.

**Menu:Cıkıs**

-Uygulamayı kapatmak veya giriş ekranına dönmek için arayüz saglar.

**6.2 Giriş Arayüzü**

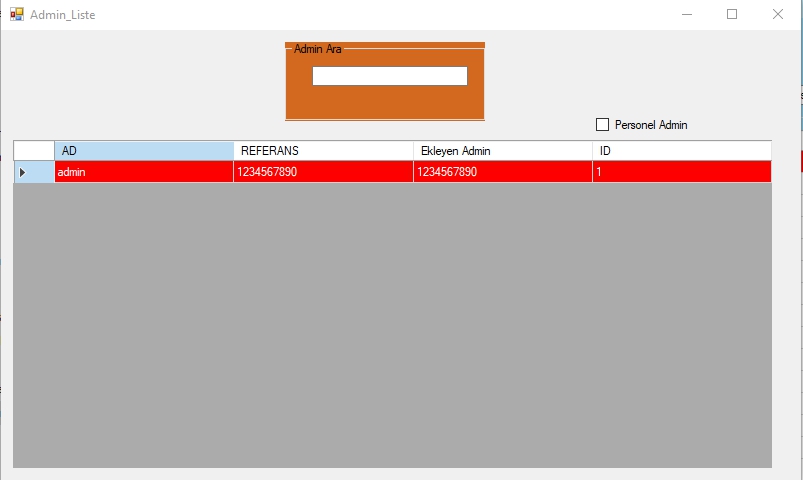


**Temel Özellikler :**

Basit Sql sorgusu ile 1 veya 0 dönderip , Gelen Veriye göre Sonraki acılacak sayfalar için

Admin için : 1 , Personel için : 0 bilgilerini göndererek , açılacak olan sayfaların niteliklerini belirler.

**6.3 Admin Liste Arayüz**

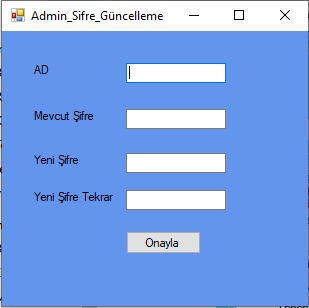


**Temel Özellikler :**

Textboxa girilen bilgiye göre basit sql sorgularından dönen DataTable nesnesine göre gridView üzerinde listeleme yapar.

Personl/Admin seçeneği ile listeleme yapılacak olan sistemde söz sahibi olan çalışanları veya adminleri görüntüleyebilirsiniz.

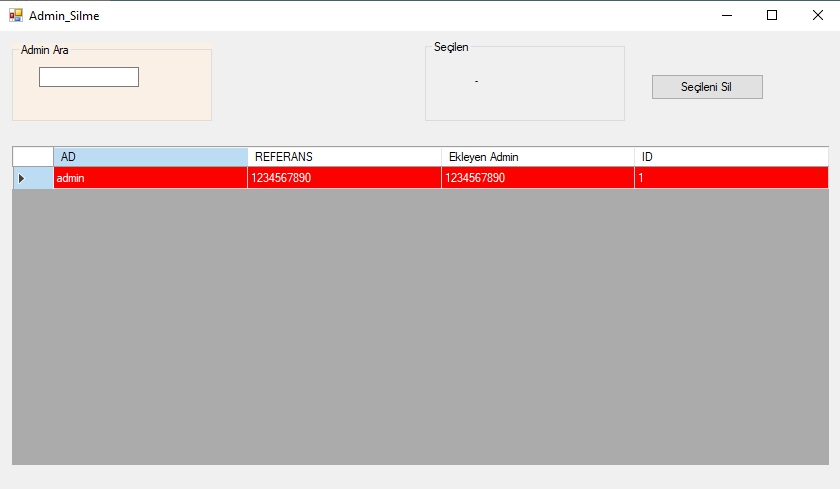
**6.4Admin Sifre Güncelleme Arayüzü**



**Temel Özellikler :**

Veri tabanı ile sürekli olarak iletşime geçerek girilen bilgilerin doğruluğunu sorgulayan sorgular göndererek geri değerler döndürür ve sifre güncelleme islemini yapar

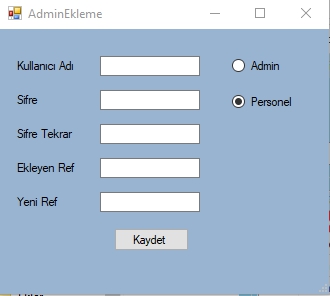
**6.5 Admin Silme Arayüz**



**Temel Özellikler :**

Textbox aracılığ ile filtrelenen adminleri seçip slime işlemini yerine getirir . Seçilen adminin bilgilerini Seçilen kısmında görebilirsiniz.

**6.6 Admin Ekleme Arayüz**



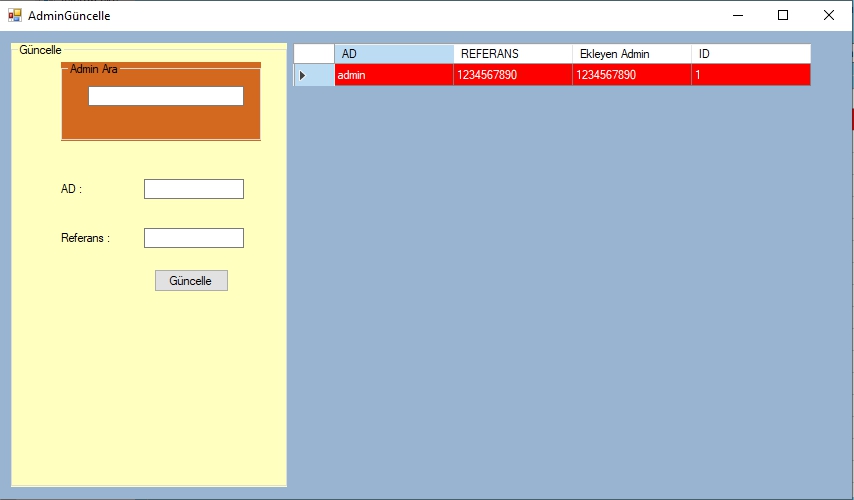
**Temel Özellikler :**

Duyulan ihtiyaca göre kullanıcı adına göre filtrelem yaparak Personel / Admin (her ikisde yönetici) olarak ekleme yapabilirsiniz.

Yeni Admin eklemek için mutlaka baska bir admini referansına ihtiyacınz vardır .

Default Ref (1234567890)

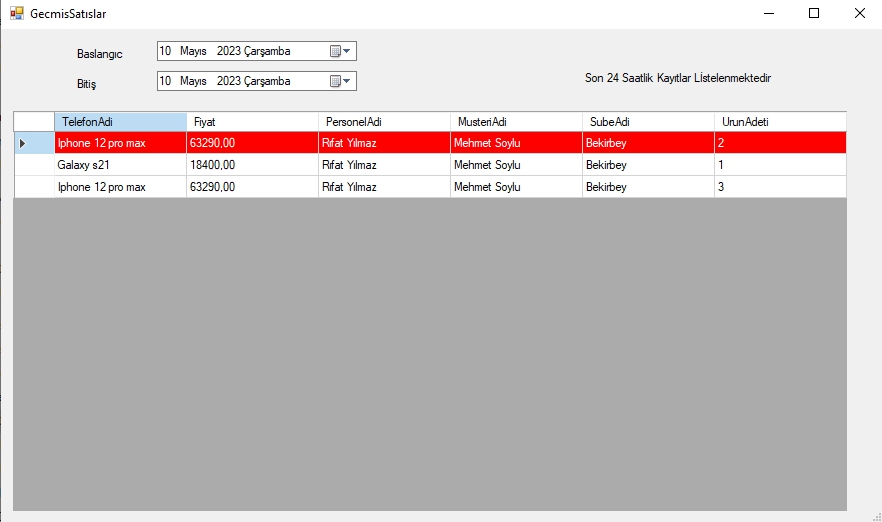
**6.7 AdminGüncelleme Arayüz**

****

**Temel Özellikler :**

Filtreleme Yaparak belirtilen özellikleri güncelleyebilirsiniz

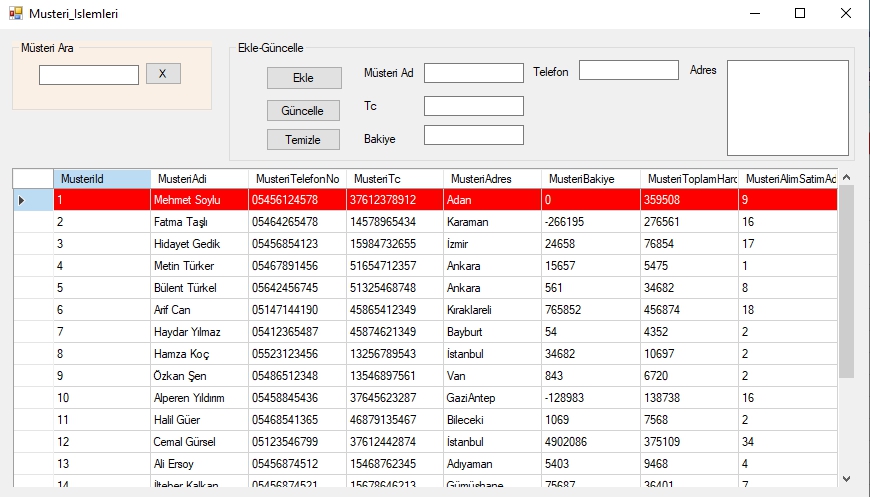
**6.8 GemisSatışlar Arayüz**



**Temel Özellikler :**

Yapıla Satışlar istediğiniz tarihler arasında filtreleyrek görüntüleyebilirsiniz

**6.9 Musteri Islemleri Arayüzü**

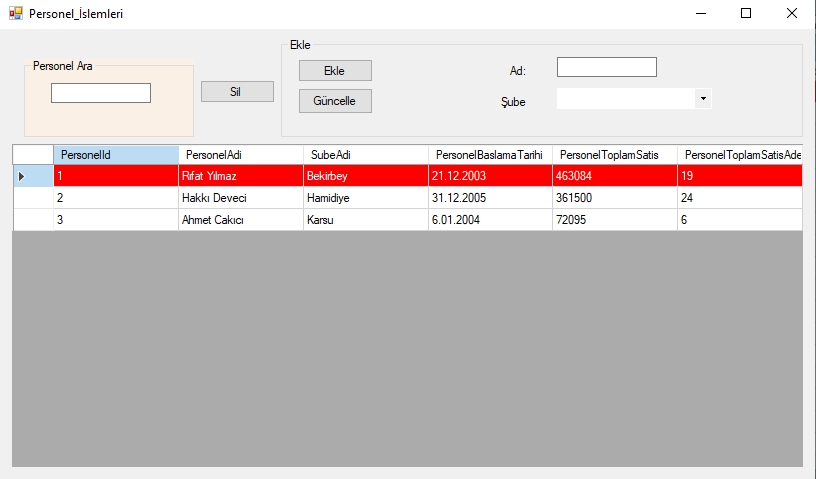


**Temel Özellikler:**

Admin tarafından müsterinin tüm bilgileri güncellenebilir .

Personell tarfından ise bakiye haricindeki bilgiler güncellenebilirdir.

**6.10 Personel Islemleri Arayuzu**



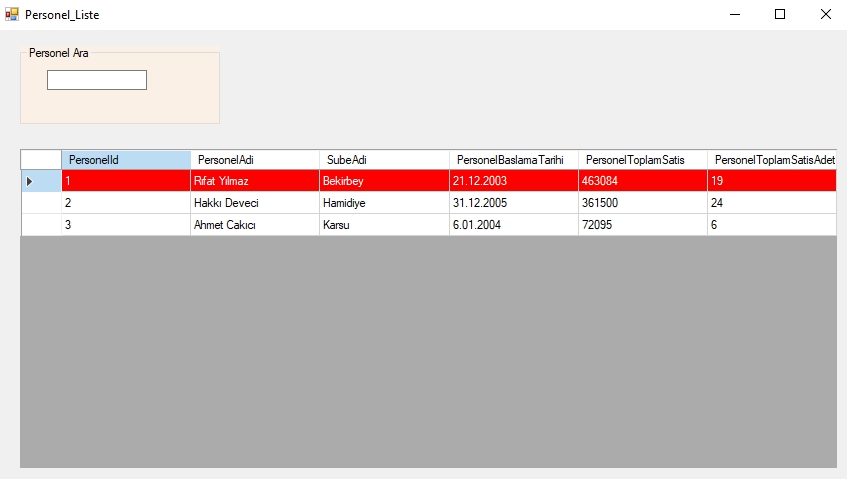
**Temel Özellikler :**

Çalışan eklemek için kullanılacak sayfadır.

Aynı zamanda çalışan bilgilerini güncellemek içinde kullanılacak sayfadır.

Silme islemi icin Personel ile yaptığı işlemler arasında cascade ilişkisi vardır.

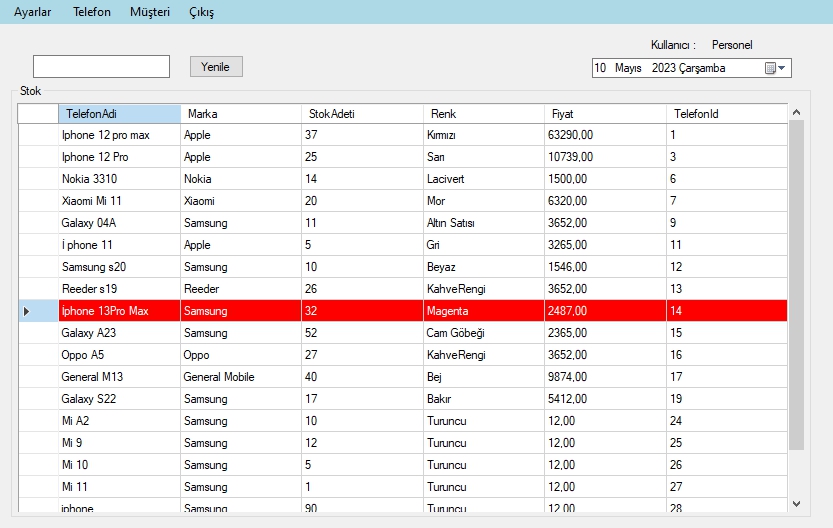
**6.11 Personel Liste Arayüzü**



**Temel Özellikler :**

Personel listelemk için kullanılır.

**6.12 Personel Arayüzü**

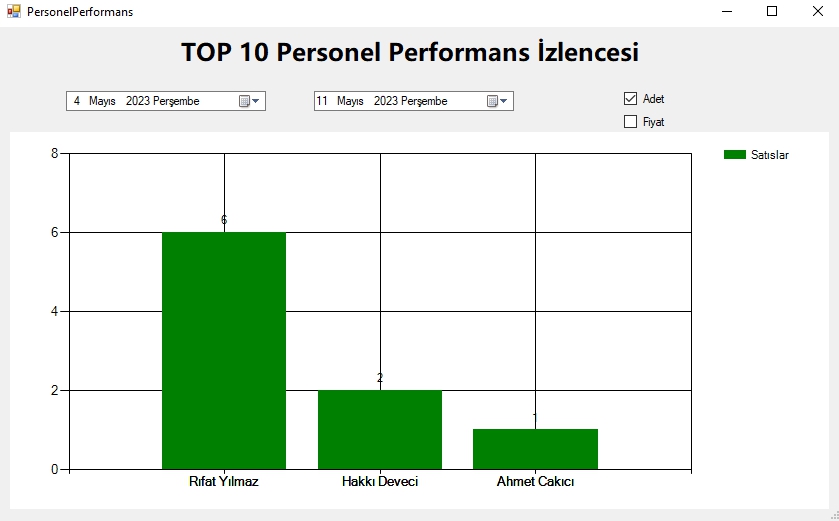


**Temel Özellikler :**

Sistemde yetkili olarak çalışan personellerin erişebileceği sayfadır.

Müşteri güncelleme , ekleme , ve satış işlemleri için kullanılmaktadır.

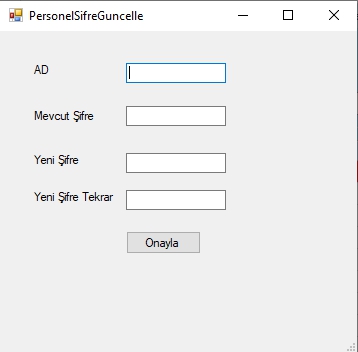
**6.13 Personel Performans Arayüzü**



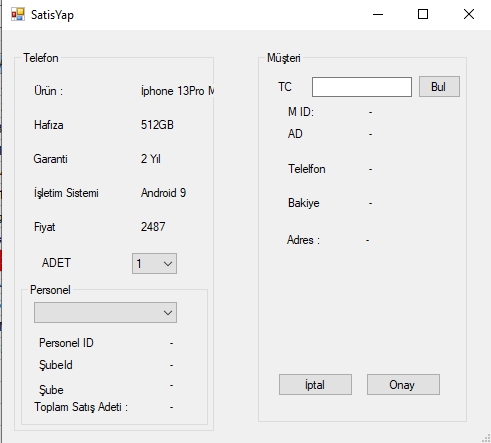
**Temel Özellikler :**

Yanlızca adminlerin erişebileceği Personellerin performans izlencesidir. Yapılan seçime göre en çok satış yapan miktar, veya adet olarak personellerin performanslarını listeler. En fazla 10 personele kadar listeleme yapar .

**6.14 PersonelSifreGuncelle Arayüzü**



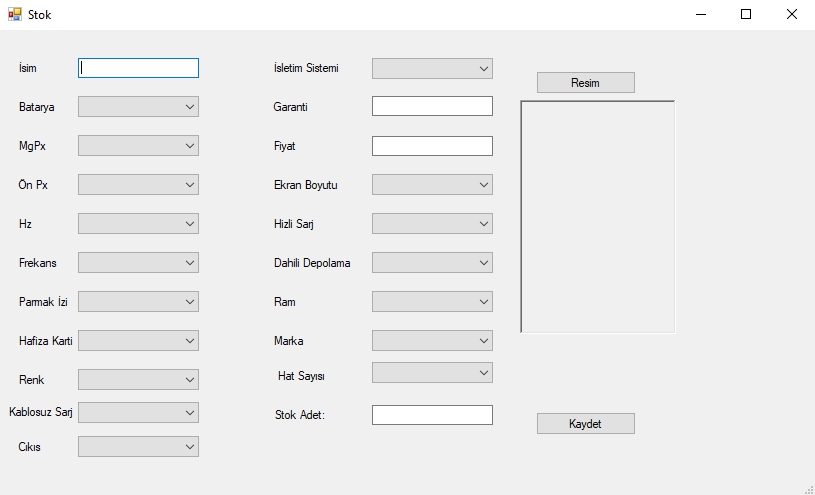
**6.15 SatisYap Arayüzü**



**Temel Özellikler :**

Sadece sistemdeki yetkili personeller tarafından seçilen telefon üzerine satış yapmak amacı ile kullanılır . Yapılan satıs adeti ile stok miktarı arasında ilişki vardır . Aynı zamanda satışı yapan eleman burada belirlenir.

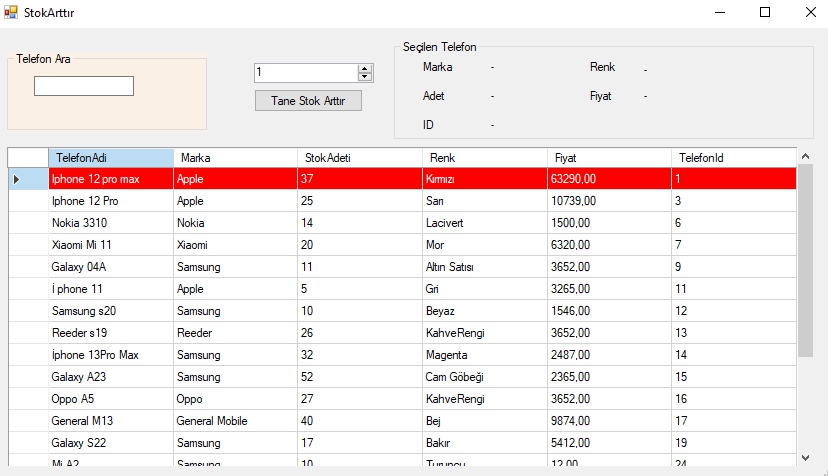
**6.16 Stok Arayüzü**



**Temel Özellikler :**

Telefon eklemek için kullanılan arayüzdür. Tüm özellikler doldurulmak zorundadır ve veriler Veri tabanından çekilerek listlenir

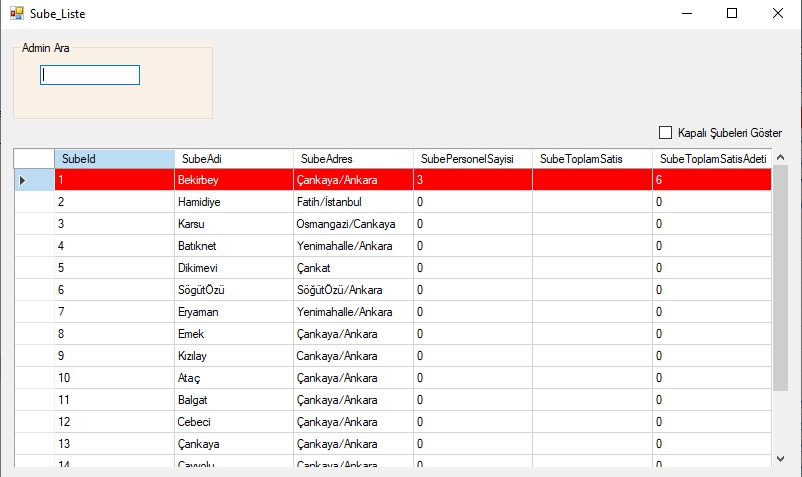
**6.17 StokArttır Arayüzü**



**Temel Özellikler :**

Mevcut stok üzerine arttırım yapmak için kullanılır .

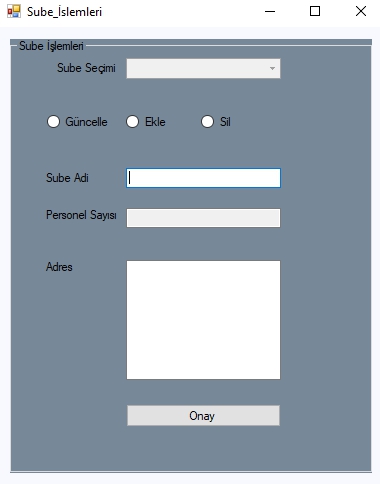
**6.18 Sube Liste Arayüzü**



**Temel Özellikleri :**

Sayfa tamamen listeleme için kullanılır .Seçilen seçeneğe göre kapatılan , silinen şubeleride de listeleyebilir

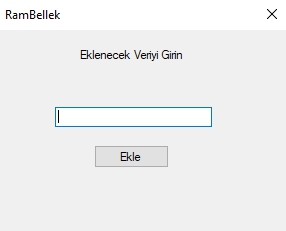
**6.19 Sube Islemleri Arayüzü**



**Temel Özellikler :**

Temel ekleme , cıkarma işlemlerini yapmak için kullanılır.

**6.20 TabloEkle Arayüzü**



**Temel Özellikler :**

Daha sonar kullanılmak üzere nitelik bilgisi kaydetmek için kullanılır .

Bu sayfa dinamik olarak oluşturulur ve seçilen seçeneğe göre ismi ve gönderdiği parametreler değişerek seçilen tablolara ekleme işlemi yapar .

**6.21Telefon Bilgileri Arayüzü**



**Temel Özellikler :**

Satış öncesi telefon bilgilerini göstermek için kullanılır .

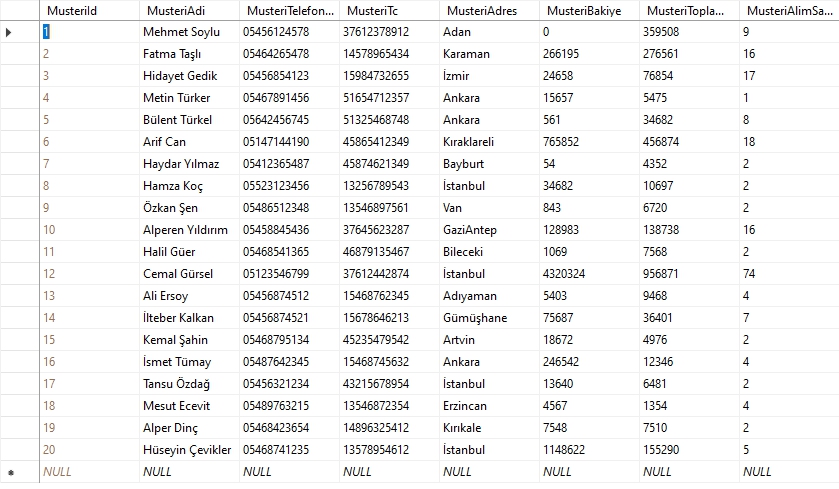
**7.Veri Tabanı Değişiklikleri**

**:** Veri Tabanında 20+ tablo bulunmaktadır. İslemlerden sonra değişiklik gösteren tablolar burada gösterilecektir

**İslem Oncesi :**

**Müşteriler Tablosu :**

Müşteri bilgilerinin tutulduğu ve müşteri id ile diğer tablolara foreign key ile bağlı olan tablo



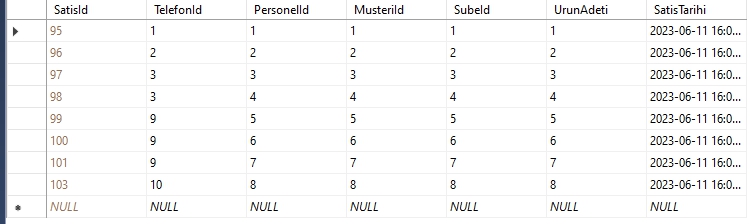
**Personeller Tablosu :**

Personel bilgilerini içeren ve Personel ıd , subeid ile diğer tablolara bağlı olan tablo



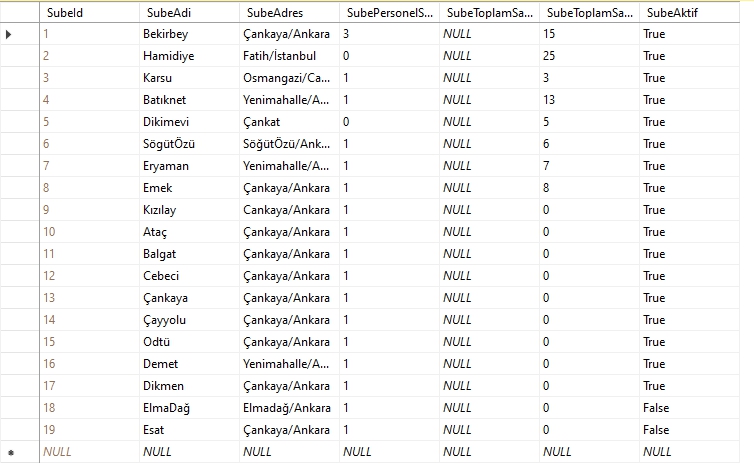
**Satıslar Tablosu :**

Satıs bilgilerinin tutulduğu sat personelıd , subeıdi telefonıd müşteri id ile diğer tablolara bağlı tablo.



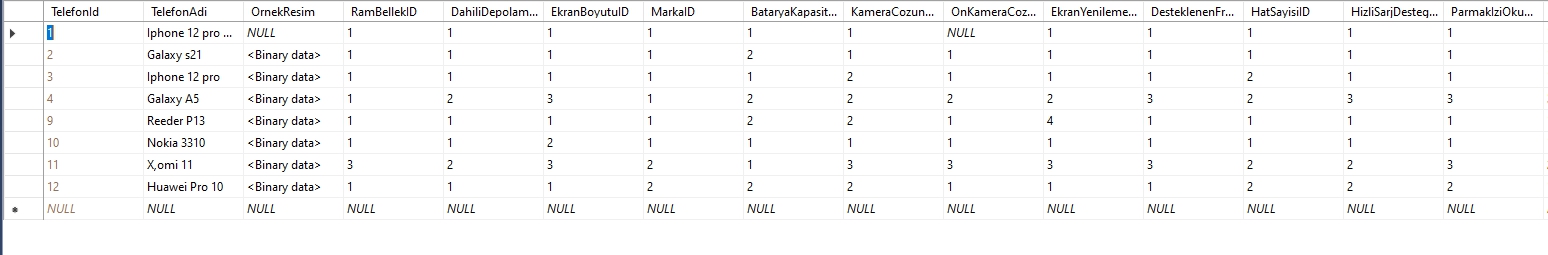
**Subeler Tablosu :**

Sube id ile diğer tablolara bağlı olan sube bilgilerini içeren tablo



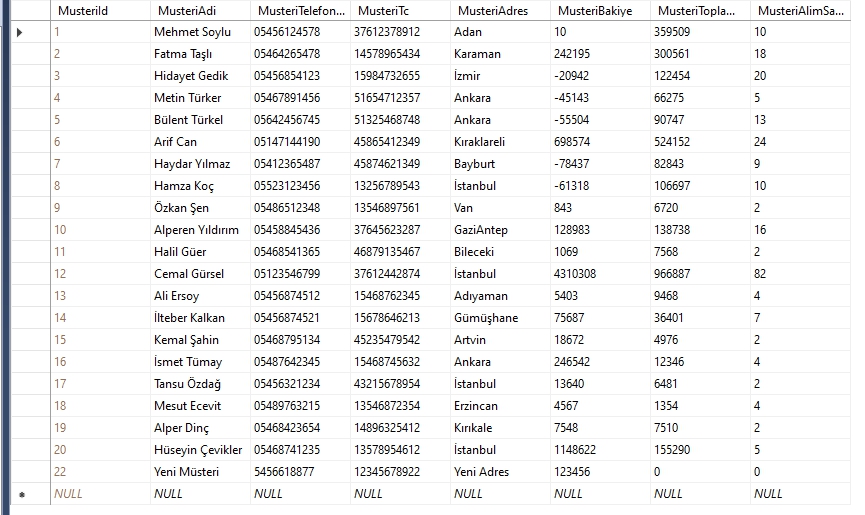
**Telefonlar Tablosu :**

Telefon bilgilerinin tutulduğu telefonid ile diğer tablolara bağlı tablo.



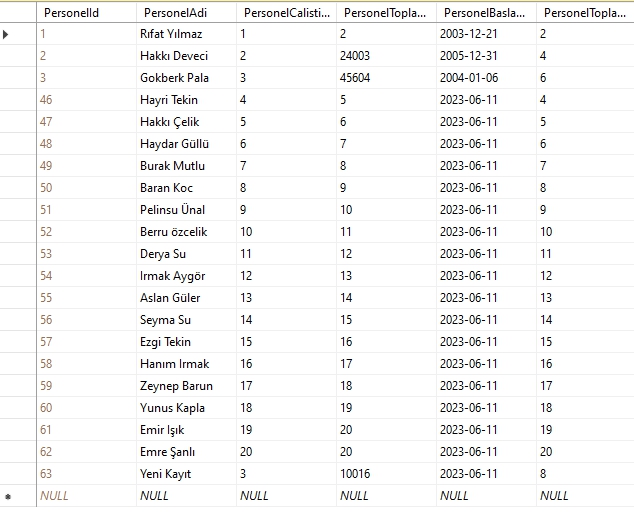
**Temel İslemlerden Sonra Tablolar :**

**Müşteriler Tablosu :**



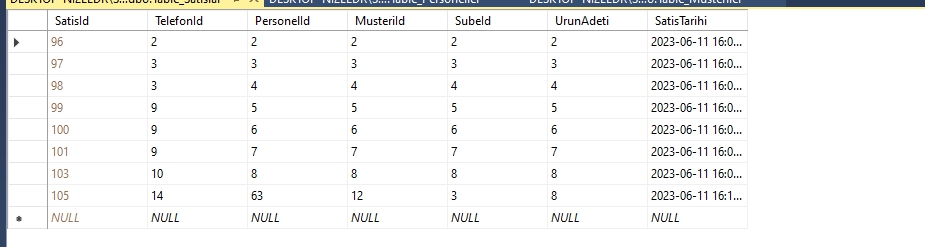
1 id li müşterinin bakiye bilgisi güncellenip , 22 id li yeni müşteri eklenmiştir . Null alabilen değerler default value ler ile eklenmiştir.

**Personeller Tablosu :**



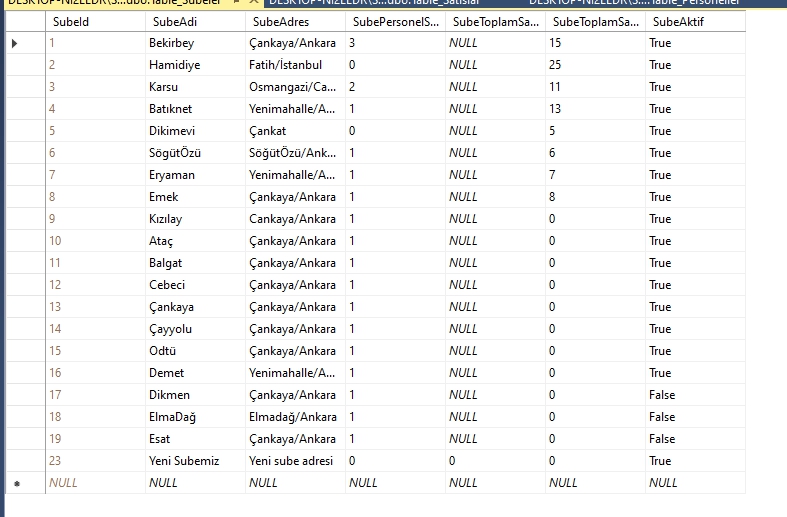
68 id li yeni kayıt eklenmiş ve satış işlemi yaptırılmıştır.

**Satışlar Tablosu :**



Yeni eklenen 63 id li kayıt ile satış yaptırılmıştır .

**Subeler Tablosu :**



23 idli yeni sube eklenmiştir. Yeni eklenen personel kaydı bu subeye ait olsydi personel sayısı otomatik güncellenecekti . Bu subeye ait personel kaydı olmadığı için personel sayisi 0 görünmektedir.

**Telefonlar Tablosu :**



1 id li telefon kaydı silinip 14 idli yeni telefon kayıt seklinde yeni kayıt eklenmiştir.

**8. Kodlar Acıklamaları**

**8.1 Veri Tabanı Triggerlar(EK-3)**

**create TRIGGER stokGuncellemesi**

Satis sonrasi stok güncellemeleri

ON Table\_Satislar

AFTER INSERT

AS

BEGIN

-- SatisAdeti'ni ve TelefonId'yi al

DECLARE @SatisAdeti INT, @TelefonId INT, @personelId INT;

SELECT @SatisAdeti = UrunAdeti, @TelefonId = TelefonId, @personelId =PersonelId

FROM inserted;

-- Table\_Telefonlar tablosundaki StokAdeti'ni azalt

UPDATE Table\_Telefonlar

SET StokAdeti = CASE

WHEN StokAdeti > @SatisAdeti THEN StokAdeti - @SatisAdeti

ELSE 0 -- Stok adedi 0'dan küçük olmasın

END

WHERE TelefonId = @TelefonId;

END;

**--Trigger ->personel\_sube\_satis\_guncellemesi--**

Satis sonrasi personel tablosunun güncellenmesi

create TRIGGER personel\_sube\_guncellemesi

ON [dbo].[Table\_Satislar]

AFTER INSERT

AS

BEGIN

-- Eklenen verileri geçici bir tabloya aktar

SELECT \* INTO #InsertedData FROM inserted;

-- PersonelToplamSatis ve PersonelToplamSatisAdeti güncelleme işlemleri

UPDATE p

SET p.PersonelToplamSatis = p.PersonelToplamSatis + (SELECT (Convert(INT, t.Fiyat) \* #InsertedData.UrunAdeti) FROM Table\_Telefonlar t WHERE t.TelefonId = #InsertedData.TelefonId),

p.PersonelToplamSatisAdeti = p.PersonelToplamSatisAdeti + #InsertedData.UrunAdeti

FROM Table\_Personeller p

INNER JOIN #InsertedData ON p.PersonelId = #InsertedData.PersonelId;

-- SubeToplamSatis ve SubeToplamSatisAdeti güncelleme işlemleri

UPDATE s

SET s.SubeToplamSatis = s.SubeToplamSatis + (#InsertedData.UrunAdeti \* CONVERT(INT, t.Fiyat)),

s.SubeToplamSatisAdeti = s.SubeToplamSatisAdeti + #InsertedData.UrunAdeti

FROM Table\_Subeler s

INNER JOIN #InsertedData ON s.SubeId = #InsertedData.SubeId

INNER JOIN Table\_Telefonlar t ON t.TelefonId = #InsertedData.TelefonId;

-- Geçici tabloyu temizle

DROP TABLE #InsertedData;

END;

**--Trigger ->musteri bilgiler—**

Satis Sonrasi müşteri tablolarının güncellenmesi

create TRIGGER MusteriBilgileriGuncelle

ON [dbo].[Table\_Satislar]

AFTER INSERT

AS

BEGIN

-- Tablo değişikliklerini içeren inserted tablosundan gerekli verileri al

DECLARE @TelefonId INT, @UrunAdeti INT, @MusteriId INT;

SELECT @TelefonId = TelefonId, @UrunAdeti = UrunAdeti, @MusteriId = MusteriId

FROM inserted;

-- MusteriBakiye'yi güncelle

UPDATE Table\_Musteriler

SET MusteriBakiye = case

when MusteriBakiye>0 Then MusteriBakiye - (SELECT Fiyat FROM Table\_Telefonlar WHERE TelefonId = @TelefonId) \* @UrunAdeti

else 0 end

WHERE MusteriId = @MusteriId;

-- MusteriToplamHarcama'yi güncelle

UPDATE Table\_Musteriler

SET MusteriToplamHarcama = MusteriToplamHarcama + (SELECT Fiyat FROM Table\_Telefonlar WHERE TelefonId = @TelefonId) \* @UrunAdeti

WHERE MusteriId = @MusteriId;

-- MusteriAlimSatimAdeti'ni güncelle

UPDATE Table\_Musteriler

SET MusteriAlimSatimAdeti = MusteriAlimSatimAdeti + @UrunAdeti

WHERE MusteriId = @MusteriId;

END;

**--trigger -> personel sayisi azaltma**—

Personel silme – ekleme işlemleri sonrası personel ve sube tablolarının güncellenmesi

CREATE TRIGGER sube\_personel\_sayisi\_azaltma

ON Table\_Personeller

AFTER DELETE

AS

BEGIN

UPDATE Table\_Subeler

SET SubePersonelSayisi = CASE

WHEN SubePersonelSayisi > 0 THEN SubePersonelSayisi - 1

ELSE 0

END

WHERE SubeId IN (SELECT PersonelCalistigiSubeId FROM DELETED);

END;

--trigger -> personel sayisi arttirma --

create TRIGGER sube\_personel\_sayisi\_arttırma

ON Table\_Personeller

AFTER insert

AS

BEGIN

UPDATE Table\_Subeler

SET SubePersonelSayisi =

SubePersonelSayisi + 1

WHERE SubeId IN (SELECT PersonelCalistigiSubeId FROM inserted);

END;

**8.2 Kod Acıklamaları**

Acıklama : Sürekli Benzer İslemler yapıldığı için AnaKodlar aktarılacaktır.Kalan tüm kodlara proje teslimindeki kaynak kodlardan erişebilirsiniz.

**Class Acıklamaları :**

Program Cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Telefon

{

static class Program

{

/// <summary>

/// Uygulamanın ana girdi noktası.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Giris());

}

}

}

**Acıklama :**

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Satırları metin düzenlemelerini sağlar

Application.Run(new Giris());

Baslangic formunu tetikler. Projeyi ayağa kaldırır.

using System;

Kütüphaneleri içe aktarır .

**VeriTabaniStok.cs**

public SqlConnection baglanti = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=telefon;Integrated Security=True");

private SqlCommand command;

**:: Baglantı cümlesi . Veri tabanı ile iletişimi sağlar .**

public void comboBoxDoldur(ComboBox combobox, string tablo, string kolon)

{

try

{

baglanti.Open();

string query = $"SELECT {kolon} FROM {tablo} ";

// Bağlantıyı açın ve sorguyu çalıştırın

SqlCommand command = new SqlCommand(query, baglanti);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

// Her satırdaki değerini ComboBox'a ekleyin

combobox.Items.Add(reader[kolon].ToString().Trim());

}

reader.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

**::Aldıgı combobox ve kolon parametreleri ile comboboxı seçilen tablo değerleri ile doldurur**

public void telefonEkle(Dictionary<string, object> comboData)

{

try

{

string tableName = "Table\_Telefonlar";

string columnNames = string.Join(", ", comboData.Keys);

string columnValues = string.Join(", ", comboData.Values);

string query = $"INSERT INTO {tableName} ({columnNames}) VALUES ({string.Join(", ", comboData.Keys.Select(k => "@" + k))})";

command = new SqlCommand(query, baglanti);

foreach (var kvp in comboData)

{

command.Parameters.AddWithValue("@" + kvp.Key, kvp.Value);

}

baglanti.Open();

command.ExecuteNonQuery();

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Sözlükteki verileri sözlükten ayırdığı tablo isimleri ile veri tabanına kaydeder .**

public string idAl(string idStunu, string tablo, string stunadi, string stundegeri)

{

try

{

string iddegeri = "0";

baglanti.Open();

// Bataryalar tablosundan bataryaid'ye göre batarya değerini almak için sorgu

string idQuery = $"SELECT {idStunu} FROM {tablo} WHERE {stunadi} = '{stundegeri}'";

command = new SqlCommand(idQuery, baglanti);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

iddegeri = (reader[idStunu]).ToString();

}

reader.Close();

return iddegeri;

}

catch (Exception)

{

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Gönderilen verinin id sini gönderir.**

public void resimGoster(int id,PictureBox pictureBox)

{

try

{

string query = "SELECT OrnekResim FROM Table\_Telefonlar WHERE TelefonID = @id";

command = new SqlCommand(query, baglanti);

command.Parameters.AddWithValue("@id", id);

baglanti.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

if (reader["OrnekResim"] != DBNull.Value)

{

byte[] resimBytes = (byte[])reader["OrnekResim"];

using (MemoryStream ms = new MemoryStream(resimBytes))

{

Image resim = Image.FromStream(ms);

pictureBox.Image = resim;

}

}

else

{

// Eğer resim null ise, PictureBox'i temizleyebilirsiniz.

pictureBox.Image = null;

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Veri tabanındaki kayıtlı resimleri Picture box a aktarmak için kullanılır.**

public void stokArttir(int id ,int adet)

{

try

{

baglanti.Open();

string sql = $"update Table\_Telefonlar set StokAdeti=StokAdeti+{adet} where TelefonId = {id} ";

command = new SqlCommand(sql, baglanti);

command.ExecuteNonQuery();

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Telefon stok miktarını arttırmak için update işlemi yapar**

public void tabloyaEkle(string tabloadi , string deger)

{

try

{

baglanti.Open();

string query ="INSERT INTO " + tabloadi + " VALUES (@deger)";

command = new SqlCommand(query, baglanti);

command.Parameters.AddWithValue("@deger", deger);

command.ExecuteNonQuery();

string mesaj = tabloadi + " tablosuna veri Eklenmiştir";

MessageBox.Show(mesaj, "Ekleme Başarılı");

baglanti.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State ==System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Gönderilen parametre ve verileri tabloya ekleme işlemi yapar .**

**VeriTabanıAdmin.cs**

public bool adminGiris(string admin\_kul\_adi, string admin\_sifre)

{

try

{

bool giris = false;

baglantı.Open();

SqlCommand command = new SqlCommand("SELECT COUNT(\*) FROM Table\_Admin WHERE kullanici\_adi=@kullaniciAdi AND sifre=@sifre and IsAdmin=1", baglantı);

command.Parameters.AddWithValue("@kullaniciAdi", admin\_kul\_adi);

command.Parameters.AddWithValue("@sifre", admin\_sifre);

int count = (int)command.ExecuteScalar();

if (count > 0)

{

giris = true;

}

return giris;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

::**Gelen verilerden sorgu oluşturarak adminlerin girişin sağlayarak admin paneline yönlendirir. Ayırıcı değer 1 ve 0 olarak gelir 1 ler adminleri 0 lar personelleri temsil eder .**

public bool adminGiris(string admin\_kul\_adi, string admin\_sifre)

{

try

{

bool giris = false;

baglantı.Open();

SqlCommand command = new SqlCommand("SELECT COUNT(\*) FROM Table\_Admin WHERE kullanici\_adi=@kullaniciAdi AND sifre=@sifre and IsAdmin=1", baglantı);

command.Parameters.AddWithValue("@kullaniciAdi", admin\_kul\_adi);

command.Parameters.AddWithValue("@sifre", admin\_sifre);

int count = (int)command.ExecuteScalar();

if (count > 0)

{

giris = true;

}

return giris;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

::**Girilen değerleri karşılaştırma yaparak sifrelerin uyuşup uyuşmadığı hakkında bool değer döndürür**

public bool adminReferansKontrol(string referans)

{

try

{

bool onay = false;

baglantı.Open();

SqlCommand command = new SqlCommand("SELECT referans FROM Table\_Admin", baglantı);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

if (reader["referans"].ToString() == referans)

{

onay = true;

}

}

return onay;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

return false;

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

::**Admin eklerken sadece diğer adminler tarafından eklenebilir olmasu acısından referans numaralarını kontrol eder. Her adminin kendine ait referansı vardır .**

public bool adminKullaniciAdiKontrol(string kul\_adi)

{

try

{

baglantı.Open();

string query = "SELECT COUNT(\*) FROM Table\_Admin WHERE kullanici\_adi = @Username";

command = new SqlCommand(query, baglantı);

// Parametreleri ayarlama

command.Parameters.AddWithValue("@Username", kul\_adi);

// Sorguyu çalıştırma ve sonucu alıp döndürme

int count = (int)command.ExecuteScalar();

//eğer var ise true , yok ise false dönecek

return (count > 0);

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

return false;

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

**::Aynı kullancı adlarının önüne geçmek amacı ile sorgu ile kayıt olup olmadığını kontrol eder**

public DataTable personeladminListeleTable()

{

try

{

// SqlCommand kullanarak verileri seçin

string query = "SELECT kullanici\_adi as 'AD',referans as 'REFERANS', ekleyen\_referans as 'Ekleyen Admin', AdminId as 'ID' FROM Table\_Admin where Isadmin=0";

command = new SqlCommand(query, baglantı);

// SqlDataAdapter ile verileri bir DataTable'e yükleyin

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dataTable = new DataTable();

da.Fill(dataTable);

return dataTable;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

::**Admin listelemek için kullanılan metot**

public void adminSil(string ad)

{

try

{

// SqlConnection ile veritabanına bağlanın

baglantı.Open();

// SqlCommand ile sorguyu belirleyin

string query = "DELETE FROM Table\_Admin WHERE kullanici\_adi ='" + ad + "' ";

SqlCommand cmd = new SqlCommand(query, baglantı);

// Sorguyu veritabanına gönderin

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

::**Admin silmek için kullanılan metot**

public void adminGuncelle(string ad, string referans, int id)

{

try

{

baglantı.Open();

command = new SqlCommand($"UPDATE Table\_Admin SET kullanici\_adi = @Ad, referans = @Referans WHERE AdminId = @Id", baglantı);

command.Parameters.AddWithValue("@Ad", ad);

command.Parameters.AddWithValue("@Referans", referans);

command.Parameters.AddWithValue("@Id", id);

command.ExecuteNonQuery();

// Veritabanı bağlantısını açma

// Sorguyu çalıştırma ve etkilenen satır sayısını alınması

// Veritabanı bağlantısını kapatma

baglantı.Close();

System.Windows.Forms.MessageBox.Show("Güncelleme İşlemi Başarılı ", "Admin Update");

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

// metodu overload yapıyoruz ada göre update atıyoruz

public void adminGuncelle(string ad, string yeni\_sifre)

{

try

{

baglantı.Open();

command = new SqlCommand($"UPDATE Table\_Admin SET sifre = @sifre WHERE kullanici\_adi= @ad", baglantı);

command.Parameters.AddWithValue("@ad", ad);

command.Parameters.AddWithValue("@sifre", yeni\_sifre);

command.ExecuteNonQuery();

// Veritabanı bağlantısını açma

// Sorguyu çalıştırma ve etkilenen satır sayısını alınması

// Veritabanı bağlantısını kapatma

baglantı.Close();

System.Windows.Forms.MessageBox.Show(" Güncelleme İşlemi Başarılı ", "Admin Update");

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglantı.State == System.Data.ConnectionState.Open)

{

baglantı.Close();

}

}

}

::**Admin güncellemek için kullanılan overload 2 metot .**

public void VeritabaniYedekle()

{ // klasör yolu oluşturuluyor

string klasor = "C://" + DateTime.Now.ToShortDateString();

Directory.CreateDirectory(klasor);

try

{

// Veritabanı bağlantısını açın

baglantı.Open();

command = new SqlCommand();

// sorgu parametreler ile backup işlemi yapıypr

command = new SqlCommand(@"backup database telefon to disk = @directory with init,stats=10", baglantı);

command.Parameters.AddWithValue("@directory", string.Concat("C://", DateTime.Now.ToShortDateString(), "/telefon.bak"));

command.ExecuteNonQuery();

MessageBox.Show($"Veritabanı yedeği \n C: Dizinine -- {DateTime.Now.ToShortDateString()}-- adı ile \nKlasörüne başarıyla oluşturuldu.");

}

catch (Exception ex)

{

MessageBox.Show("Hata: " + ex.Message);

}

finally

{

// Veritabanı bağlantısını kapatın

baglantı.Close();

}

}

::**Veri Tabanı yedeği almak için kullanılan metot.**

**VeriTabanıMüsteri.cs**

public void musteriEkle(string[] musteri\_bilgiler)

{

try

{

baglanti.Open();

// INSERT sorgusu hazırlama

string query = "INSERT INTO Table\_Musteriler (MusteriAdi, MusteriTelefonNo,MusteriTc,MusteriAdres,MusteriBakiye) VALUES (@adi, @telefon, @tc,@Adres, @bakiye)";

command = new SqlCommand(query, baglanti);

// Parametreleri ayarlama

command.Parameters.AddWithValue("@adi", musteri\_bilgiler[0]);

command.Parameters.AddWithValue("@telefon", musteri\_bilgiler[1]);

command.Parameters.AddWithValue("@tc", musteri\_bilgiler[2]);

command.Parameters.AddWithValue("@Adres", musteri\_bilgiler[3]);

command.Parameters.AddWithValue("@bakiye", float.Parse( musteri\_bilgiler[4]));

// Sorguyu çalıştırma

command.ExecuteNonQuery();

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**dizi ile gelen değerleri veri tabanına ekler .**

public DataTable müsteriFiltrele(string filtre)

{

try

{

baglanti.Open();

// SqlCommand ile sorguyu belirleyin

string query = "SELECT \* FROM Table\_Musteriler WHERE MusteriAdi LIKE @name";

SqlCommand cmd = new SqlCommand(query, baglanti);

// SqlParameter ile kullanıcının girdiği isim verisini belirleyin

string inputName = filtre;

SqlParameter param = new SqlParameter("@name", "%" + inputName + "%");

cmd.Parameters.Add(param);

// SqlDataReader ile sorgudan verileri okuyun

SqlDataReader reader = cmd.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(reader);

return dt;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Filtreleme işlemi için kullanılan metot**

public void musteriHarcama(int musteriid, string harcama)

{

try

{

baglanti.Open();

string query = "update Table\_Musteriler set MusteriBakiye= MusteriBakiye - @harcama , MusteriToplamHarcama = @harcama where MusteriId=@id ";

command = new SqlCommand(query, baglanti);

command.Parameters.AddWithValue("@harcama",decimal.Parse( harcama));

command.Parameters.AddWithValue("@id",musteriid);

command.ExecuteNonQuery();

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Musteri bakiyesini düzenleyen metot**

**VeriTabaniPersonel.cs**

public int subeIdAl(string subeadi)

{

try

{

int subeId = 0;

string query = "SELECT SubeId FROM Table\_Subeler WHERE SubeAdi = @subeAdi";

command = new SqlCommand(query, baglanti);

command.Parameters.AddWithValue("@subeAdi", subeadi);

object result = command.ExecuteScalar();

if (result != null && result != DBNull.Value)

{

subeId = Convert.ToInt32(result);

}

return subeId;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

}

::**birleştirme işlemleri için Personellerin id sini alan metot**

public bool personelGiris(string admin\_kul\_adi, string admin\_sifre)

{

try

{

bool giris = false;

baglanti.Open();

SqlCommand command = new SqlCommand("SELECT COUNT(\*) FROM Table\_Admin WHERE kullanici\_adi=@kullaniciAdi AND sifre=@sifre and IsAdmin=0", baglanti);

command.Parameters.AddWithValue("@kullaniciAdi", admin\_kul\_adi);

command.Parameters.AddWithValue("@sifre", admin\_sifre);

int count = (int)command.ExecuteScalar();

if (count > 0)

{

giris = true;

}

return giris;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

**::Personel girişi sağlayan metot**

public DataTable personelPerformansFiltrele(string filtre)

{

try

{

baglanti.Open();

// SqlCommand ile sorguyu belirleyin

string query = "SELECT p.PersonelAdi ,p.PersonelToplamSatis as 'Miktar' ,p.PersonelToplamSatisAdeti as 'Adet' ,s.SubeAdi ,p.PersonelId from Table\_Personeller p inner join Table\_Subeler s on p.PersonelCalistigiSubeId =s.SubeId where PersonelAdi LIKE @name";

SqlCommand cmd = new SqlCommand(query, baglanti);

// SqlParameter ile kullanıcının girdiği isim verisini belirleyin

string inputName = filtre;

SqlParameter param = new SqlParameter("@name", "%" + inputName + "%");

cmd.Parameters.Add(param);

// SqlDataReader ile sorgudan verileri okuyun

SqlDataReader reader = cmd.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(reader);

return dt;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

public DataTable performansListeleTable()

{

try

{

// SqlCommand kullanarak verileri seçin

string query = "SELECT p.PersonelAdi ,p.PersonelToplamSatis as 'Miktar' ,p.PersonelToplamSatisAdeti as 'Adet' ,s.SubeAdi ,p.PersonelId from Table\_Personeller p inner join Table\_Subeler s on p.PersonelCalistigiSubeId =s.SubeId ";

command = new SqlCommand(query, baglanti);

// SqlDataAdapter ile verileri bir DataTable'e yükleyin

SqlDataAdapter da = new SqlDataAdapter(command);

DataTable dataTable = new DataTable();

da.Fill(dataTable);

return dataTable;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Personel Performans değerlendirmesi için gereken değerleri ceken overlaod 2 metot**

public bool personelAdKontrol(string adi,string sube)

{

try

{

baglanti.Open();

string query = "SELECT COUNT(\*) FROM Table\_Personeller p " +

"INNER JOIN Table\_Subeler s ON p.PersonelCalistigiSubeId = s.SubeId " +

"WHERE p.PersonelAdi = @personelAdi AND s.SubeAdi = @subeAdi";

using (SqlCommand command = new SqlCommand(query, baglanti))

{

command.Parameters.AddWithValue("@personelAdi", adi);

command.Parameters.AddWithValue("@subeAdi", sube);

int count = (int)command.ExecuteScalar();

return count == 0;

}

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

return false;

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

***::*Personel Ad kontrol için metot**

**VeriTabaniSatisİslemleri.Cs**

public DataTable tarihFiltre(DateTime baslangicTarihi, DateTime bitisTarihi)

{

try

{

baglanti.Open();

// SqlCommand ile sorguyu belirleyin

string formattedParam1 = baslangicTarihi.ToString("yyyy-MM-dd HH:mm:ss.fff");

string formattedParam2 = bitisTarihi.ToString("yyyy-MM-dd HH:mm:ss.fff");

string query = "SELECT TelefonAdi , tel.Fiyat,PersonelAdi , MusteriAdi,SubeAdi,UrunAdeti FROM Table\_Satislar t " +

"inner join Table\_Personeller p on p.PersonelId =t.PersonelId " +

"inner join Table\_Musteriler m on m.MusteriId = t.MusteriId" +

" inner join Table\_Subeler s on s.SubeId = t.SubeId " +

"inner join Table\_Telefonlar tel on tel.TelefonId = t.TelefonId "+

"WHERE (SatisTarihi >= @param1 AND SatisTarihi <= @param2) OR (SatisTarihi = @param1 OR SatisTarihi = @param2)";

SqlCommand cmd = new SqlCommand(query, baglanti);

cmd.Parameters.AddWithValue("@param1", formattedParam1);

cmd.Parameters.AddWithValue("@param2", formattedParam2);

// SqlParameter ile kullanıcının girdiği isim verisini belirleyin

// SqlDataReader ile sorgudan verileri okuyun

SqlDataReader reader = cmd.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(reader);

return dt;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

**::Satıs filtrelemesi için belirlenen tarihler arasındaki kayıtları getiren metot**

public void satisTablosubaEkle(int telefonid, int personelid, int musteriid, int subeid,int urunadet)

{

try

{

baglanti.Open();

// SqlCommand ile sorguyu belirleyin

string query = "insert into Table\_Satislar (TelefonId, PersonelId, MusteriId, SubeId, UrunAdeti) values (@telefonid, @personelid, @musteriid, @subid, @urunadeti)";

command = new SqlCommand(query, baglanti); // SqlCommand nesnesini önce oluşturun

command.Parameters.AddWithValue("@telefonid", telefonid);

command.Parameters.AddWithValue("@personelid", personelid);

command.Parameters.AddWithValue("@musteriid", musteriid);

command.Parameters.AddWithValue("@subid", subeid);

command.Parameters.AddWithValue("@urunadeti", urunadet);

command.ExecuteNonQuery();

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Satis yapıldığında kayıtları satışlar tablosuna ekleyen metot**

public string[] musteriGetir(string tc)

{

try

{

baglanti.Open();

command = new SqlCommand($"select MusteriAdi, MusteriAdres,MusteriBakiye,MusteriTelefonNo,MusteriId from Table\_Musteriler where MusteriTc={tc}", baglanti);

List<string> bilgiler = new List<string>();

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

for (int i = 0; i < reader.FieldCount; i++)

{

string bilgi = reader[i].ToString();

bilgiler.Add(bilgi);

}

}

else

{

bilgiler.Add("false");

}

return bilgiler.ToArray();

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

**::Müsteriye satis yapılırken tc numarasını kullanarak müşteri bilgilerini ceken metot**

public List<string> bilgileriDoldur(int id)

{

try

{

baglanti.Open();

List<string> gelenBilgiler = new List<string>();

// SqlCommand kullanarak verileri seçin

string query = "SELECT t.TelefonAdi, r.RamBellek, d.DahiliDepolama, ekran.EkranBoyutu, m.Marka, b.BataryaKapasitesi, hs.HizliSarjDestegi, ks.KablosuzSarjDestegi, k.KameraCozunurlugu, onk.OnKameraCozunurlugu, ey.EkranYenilemeHizi, fr.DesteklenenFrekans, p.ParmakIziOkuyucu, rk.Renk, hk.HafizaKartiDestegi, i.IsletimSistemi, c.CikisYili, GarantiSuresi, StokAdeti ,hat.HatSayisi ,t.Fiyat " +

"FROM Table\_Telefonlar t " +

"INNER JOIN Table\_Ram\_Bellek r ON t.RamBellekID = r.RamBellekId " +

"INNER JOIN Table\_Dahili\_Depolama d ON t.DahiliDepolamaID = d.DahiliDepolamaId " +

"INNER JOIN Table\_Ekran\_Boyutu ekran ON t.EkranBoyutuID = ekran.EkranBoyutuId " +

"INNER JOIN Table\_Marka m ON t.MarkaID = m.MarkaId " +

"INNER JOIN Table\_Batarya\_Kapasitesi b ON t.BataryaKapasitesiID = b.BataryaKapasitesiId " +

"INNER JOIN Table\_Kamera\_Cozunurlugu k ON t.KameraCozunurluguID = k.KameraCozunurluguId " +

"INNER JOIN Table\_On\_Kamera\_Cozunurlugu onk ON t.OnKameraCozunurluguID = onk.OnKameraCozunurluguId " +

"INNER JOIN Table\_Ekran\_Yenileme\_Hizi ey ON t.EkranYenilemeHiziID = ey.EkranYenilemeHiziId " +

"INNER JOIN Table\_Desteklenen\_Frekans fr ON t.DesteklenenFrekansID = fr.DesteklenenFrekansId " +

"INNER JOIN Table\_Parmak\_Izi\_Okuyucu p ON t.ParmakIziOkuyucuID = p.ParmakIziOkuyucuId " +

"INNER JOIN Table\_Renk rk ON t.RenkID = rk.RenkId " +

"INNER JOIN Table\_Hafiza\_Karti\_Destegi hk ON t.HafizaKartiDestegiID = hk.HafizaKartiDestegiId " +

"INNER JOIN Table\_Hizli\_Sarj\_Destegi hs ON t.HizliSarjDestegiID = hs.HizliSarjDestegiId " +

"INNER JOIN Table\_Kablosuz\_Sarj\_Destegi ks ON t.KablosuzSarjDestegiID = ks.KablosuzSarjDestegiId " +

"INNER JOIN Table\_Isletim\_Sistemi i ON t.IsletimSistemiID = i.IsletimSistemiId " +

" INNER JOIN Table\_Hat\_Sayisi hat on t.HatSayisiID=hat.HatSayisiId " +

"INNER JOIN Table\_Cikis\_Yili c ON t.CikisYiliID = c.CikisYiliId " +

"WHERE t.TelefonID='" + id + "'";

command = new SqlCommand(query, baglanti);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

// gelen verilerin tamamı oknur

for (int i = 0; i < reader.FieldCount; i++)

{// gelen veri null mu kontrolu yapılır , değilse gelen veri object tipinde alınarak stringe dönüştürlür ve listeye eklenir

string veri = reader.IsDBNull(i) ? string.Empty : reader.GetValue(i).ToString();

gelenBilgiler.Add(veri);

};

}

return gelenBilgiler;

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

throw;

}

finally

{

if (baglanti.State == System.Data.ConnectionState.Open)

{

baglanti.Close();

}

}

}

::**Gelen diziden bilgileri tabloya ekleyen metot**

**Kalan kodlar birbirini tekrar ettiği için eklenmemiştir.**

**9.0 Zaman İş Planı Çizelgesi(EK-1)**

****

**10 – Akıs Seması**

Akıs seması Ek-4 ile beraber proje klasöründe verilmiştir. Görüntü bozulması nedeni ile buraya eklenememktedir.

****