# **İÇİNDEKİLER**

[İÇİNDEKİLER 1](#_Toc514365999)

[ŞEKİLLER 3](#_Toc514366000)

[1.VERİ TABANI TASARIM METODOLOJİSİ 4](#_Toc514366001)

[**1.1. Kavramsal Veri Tabanı Tasarımı** 4](#_Toc514366002)

[**Adım 1. Kavramsal veri modelini oluşturmak** 4](#_Toc514366003)

[**1.1.1.Varlık tiplerini belirlemek** 4](#_Toc514366004)

[**1.1.2 İlişki tiplerini belirlemek** 4](#_Toc514366005)

[**1.1.5 Fazlalık için modeli kontrol etmek** 6](#_Toc514366006)

[**1.1.6 Kullanıcı hareketlerine karşı kavramsal modeli doğrulamak** 6](#_Toc514366007)

[**1.2.3 Kullanıcı hareketlerine karşı ilişkileri doğrulamak** 7](#_Toc514366008)

[**1.2.4 Bütünlük kısıtlarını tanımlamak** 7](#_Toc514366009)

[**1.3. Fiziksel Veri Tabanı Tasarımı** 7](#_Toc514366010)

[**Adım 3. Hedef VTYS için mantıksal veri modelini dönüştürmek** 7](#_Toc514366011)

[**1.3.1 Temel ilişkileri tasarlamak** 7](#_Toc514366012)

[**1.3.2 Türetilen verilerin gösterimini tasarlamak** 7](#_Toc514366013)

[**1.3.3 Genel kısıtları tasarlamak** 8](#_Toc514366014)

[**Adım 4. Dosya organizasyonlarını ve indeksleri tasarlamak** 8](#_Toc514366015)

[**1.4.1 Hareketleri analiz etmek** 8](#_Toc514366016)

[**1.4.2 Dosya organizasyonunu seçmek** 9](#_Toc514366017)

[**1.4.3 İndeksleri seçmek** 9](#_Toc514366018)

[**1.4.4 Disk alanı gereksinimlerini tahmin etmek** 9](#_Toc514366019)

[**Adım 5. Kullanıcı görünümlerini tasarlamak** 9](#_Toc514366020)

[**Adım 6. Güvenlik mekanizmalarını tasarlamak** 9](#_Toc514366021)

[2.VERİ SÖZLÜĞÜ 10](#_Toc514366022)

[3.NORMALİZASYON 11](#_Toc514366023)

[**3.1.Normalleştirilmemiş Form (UNF)** 11](#_Toc514366024)

[**3.2.Birincil Normal Form (1NF)** 11](#_Toc514366025)

[**3.3.İkincil Normal Form (2NF)** 12](#_Toc514366026)

[**3.4. Üçüncü Normal Form (3NF)** 12](#_Toc514366027)

[4.ER DİAGRAMI 14](#_Toc514366028)

[5.PROJE EKRAN GÖRÜNTÜLERİ 15](#_Toc514366029)

[**5.1.Kayıt Silme Ekran Görüntüsü** 15](#_Toc514366030)

[**5.2.Kayıt Güncelleme Ekran Görüntüsü** 15](#_Toc514366031)

[**5.3.Kayıt Güncelleme Ekran Görüntüsü** 16](#_Toc514366032)

[**5.4.Kayıt Listeleme Ekran Görüntüsü** 16](#_Toc514366033)

[**5.5.Kayıt Arama Ekran Görüntüsü** 17](#_Toc514366034)

[6.PROJE KODLARI 18](#_Toc514366035)

[**6.1.Kütüphaneler Eklendi** 18](#_Toc514366036)

[**6.2.Bağlantılar Tanımlandı** 18](#_Toc514366037)

[**6.3.Güncelleme Kodu** 18](#_Toc514366038)

[**6.4.Silme Kodu** 18](#_Toc514366039)

[**6.5.Kayıt Ekleme Kodu** 19](#_Toc514366040)

[**6.6.Listeleme Kodu** 19](#_Toc514366041)

[**6.7.Arama Kodu** 20](#_Toc514366042)

# **ŞEKİLLER**

Şekil 1.1 Kavramsal model ilişki tipleri 3

Şekil 1.2 Anahtarlar 5

Şekil 1.3 Mantıksal ve global model birleşimi 6

Şekil 1.4 INTEL\_URUNLER tablosu Şekil 1.5 ARDUNİO\_URUNLER tablosu 7

Şekil 1.6 ARDUNİO\_TABLE tablosu 7

Şekil 1.7 INTEL\_TABLE tablosu 7

Şekil 1.8 ALAN tablosu 7

Şekil 2.1 Veri sözlüğü 9

# **1.VERİ TABANI TASARIM METODOLOJİSİ**

## **1.1. Kavramsal Veri Tabanı Tasarımı**

### **Adım 1. Kavramsal veri modelini oluşturmak**

#### **1.1.1.Varlık tiplerini belirlemek**

Beş varlığımız olacak. Bunlar:

INTEL\_TABLE : Intel ürünlerinin bulunacağı varlık.

ARDUNIO\_TABLE : Ardunio ürünlerinin bulunacağı varlık.

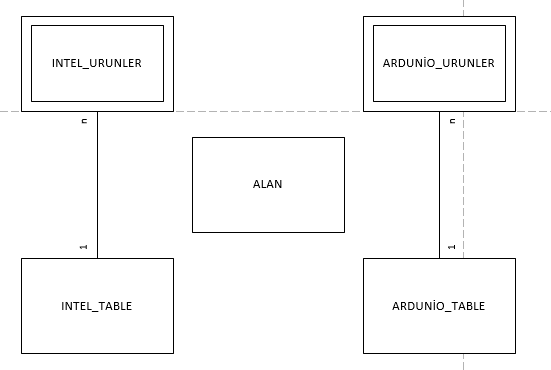
INTEL\_URUNLER : Seçilecek intel ürünlerinin bulunacağı varlık.

ARDUNİO\_URUNLER : Seçilecek ardunio ürünlerinin bulunacağı varlık.

ALAN : Yönetici bilgilerinin bulunacağı varlık.

#### **1.1.2 İlişki tiplerini belirlemek**

INTEL\_TABLE ve INTEL\_URUNLER tabloları arasında bire çok (1-n) ilişkisi vardır.

ARDUNIO\_TABLE ve ARDUNIO\_URUNLER tabloları arasında bire çok (1-n) ilişkisi vardır.

Şekil 1.1 Kavramsal model ilişki tipleri

**1.1.3 Varlık veya ilişki tipleriyle nitelikleri belirlemek ve birleştirmek**

INTEL\_TABLE tablosu önbellek, işlemci, hız, nesil, resim ve id sütunlarından oluşacak.

ARDUNİO\_TABLE tablosu spam, resim, eeprom, microdenetleyici, pwm, digital\_giris\_cikis, analog\_giris ve id sütunlarından oluşacak.

INTEL\_URUNLER tablosu intel\_id ve id sütunlarından oluşacak.

ARDUNİO\_URUNLER tablosu ardunio\_id ve id sütunlarından oluşacak.

ALAN tablosu ad ve parola sütunlarından oluşacak.

Çizelge 1.1 Varlık nitelikleri ve özellikleri

|  |  |  |  |
| --- | --- | --- | --- |
| Varlık | Nitelikler | Veri tipi & uzunluk | Null |
| INTEL\_TABLE | önbellek işlemci  hız  nesil  resim  id | Varchar2(20)  Varchar2(20)  Varchar2(20)  Varchar2(250)  Varchar2(20) | -  -  -  -  - |
| ARDUNİO\_TABLE | Spam  resim  eeprom microdenetleyici pwm digital\_giris\_cikis analog\_giris  id | Varchar2(20)  Varchar2(250)  Varchar2(20)  Varchar2(20)  Varchar2(20)  Varchar2(20)  Varchar2(20)  Varchar2(20) | -  -  -  -  -  -  -  - |
| INTEL\_URUNLER | intel\_id  id | Varchar2(20)  Varchar2(20) | -  - |
| ARDUNİO\_URUNLER | ardunio\_id  id | Varchar2(20)  Varchar2(20) | -  - |
| ALAN | ad  parola | Varchar2(20)  Varchar2(20) | -  - |

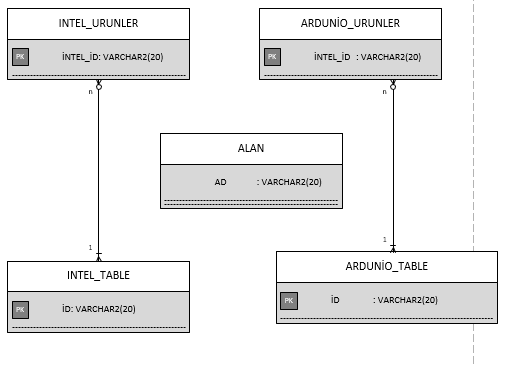
**1.1.4 Aday, birincil ve alternatif anahtar niteliklerini belirlemek**

INTEL\_TABLE tablosunda id primary key olarak belirlendi.

ARDUNİO\_TABLE tablosunda id primary key olarak belirlendi.

INTEL\_URUNLER tablosunda intel\_id primary key, id foreign key olarak belirlendi.

ARDUNİO\_URUNLER tablosunda ardunio\_id primary key, id foreign key olarak belirlendi.

****

Şekil 1.2 Anahtarlar

#### **1.1.5 Fazlalık için modeli kontrol etmek**

Modelde herhangi bir fazlalık bulunmuyor.

#### **1.1.6 Kullanıcı hareketlerine karşı kavramsal modeli doğrulamak**

Kullanıcı 0 ya da daha çok intel ürününü seçebilecek. INTEL\_TABLE ve INTEL\_URUNLER tablosu arasında ilişki bulunuyor ve bulunan bu 1-n ilişkisi kullanıcının bu işlemi yapmasına olanak sağlar.

Kullanıcı 0 ya da daha çok ardunio ürününü seçebilecek. ARDUNİO\_TABLE ve ARDUNİO\_URUNLER tablosu arasında ilişki bulunuyor ve bulunan bu 1-n ilişkisi kullanıcının bu işlemi yapmasına olanak sağlar.

**1.1.7 Kullanıcı ile kavramsal veri modelini gözden geçirmek**

Oluşturulan modelde bulunan tablolar yapacağımız tablolarda bulunan nitelikler ve bu niteliklerin özellikleri kullanıcı işlemlerini gerçekleştirmemiz için yeterli olmaktadır. İlişki tipleri istenilen gibi oluşturuldu.

**1.2. Mantıksal Veri Tabanı Tasarımı**

**Adım 2. Mantıksal veri modelini oluşturmak ve doğrulamak**

**1.2.1 Mantıksal veri modeli için ilişkileri türetmek**

INTEL\_TABLE ve INTEL\_URUNLER tabloları arasında bire çok (1-n) ilişkisi vardır.

ARDUNIO\_TABLE ve ARDUNIO\_URUNLER tabloları arasında bire çok (1-n) ilişkisi vardır.

**1.2.2 Normalleştirme kullanarak ilişkileri doğrulamak**

Normalleştirmeler sonucunda elde ettiğimiz tablolar ilişkilerimizi kullanmamız için uygundur.

#### **1.2.3 Kullanıcı hareketlerine karşı ilişkileri doğrulamak**

Kullanıcı ürünü(intel ya da ardunio) seçtiğinde, seçilen ürün ilgili tabloya(INTEL\_URUNLER ya da ARDUNİO\_URUNLER) tablosuna aktarılacak. Oluşturduğumuz tablalar bunun için uygundur.

Yönetici sisteme alan tablosundaki verileri kullanarak giriş yapabilecek. Ürün eklemek istediğinde ilgili ürünün(intel ya da ardunio) bilgilerini girerek veri tabanına kayıt yapabilecek, Ürün silmek istediğinde ürünün id bilgisini girerek ürünü silebilecek. Bilgilerini güncellemek istediği ürünün değiştirmek istediği bilgilerini girerek güncelleme yapabilecek. Tablolarımız bu işlemler için yeterlidir.

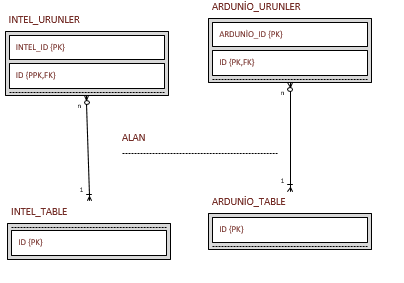
#### **1.2.4 Bütünlük kısıtlarını tanımlamak**

Bütünlük kısıdımız bulunmuyor.

**1.2.5 Kullanıcı ile mantıksal veri modelini gözden geçirmek**

Mantıksal model gözden geçirildi ve geliştirilmesi planlanan sistem için uygun olunduğu görüldü

**1.2.6 Mantıksal veri modellerini global modele birleştirmek**

****

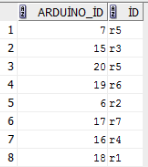
Şekil 1.3 Mantıksal ve global model birleşimi

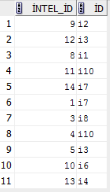
## **1.3. Fiziksel Veri Tabanı Tasarımı**

### **Adım 3. Hedef VTYS için mantıksal veri modelini dönüştürmek**

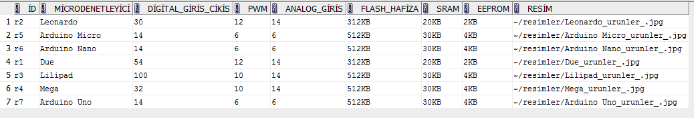
#### **1.3.1 Temel ilişkileri tasarlamak**

#### **1.3.2 Türetilen verilerin gösterimini tasarlamak**

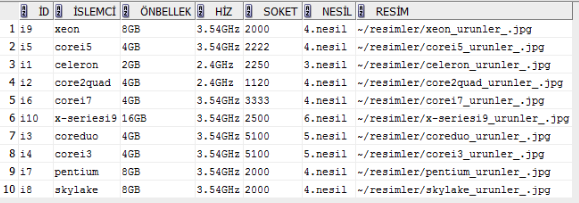




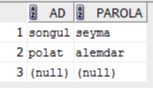
Şekil 1.4 INTEL\_URUNLER tablosu Şekil 1.5 ARDUNİO\_URUNLER tablosu



Şekil 1.6 ARDUNİO\_TABLE tablosu



Şekil 1.7 INTEL\_TABLE tablosu



Şekil 1.8 ALAN tablosu

#### **1.3.3 Genel kısıtları tasarlamak**

Kısıdımız bulunmamakta.

### **Adım 4. Dosya organizasyonlarını ve indeksleri tasarlamak**

#### **1.4.1 Hareketleri analiz etmek**

Yönetici sisteme ürün ekleyebilecek, silebilecek, ürünleri listeleyebilecek ve ürün araması yapabilecek.

Normal kullanıcı sitemden ürün araması yapabilecek ve ürün seçimi yapabilecek.

#### **1.4.2 Dosya organizasyonunu seçmek**

#### **1.4.3 İndeksleri seçmek**

#### **1.4.4 Disk alanı gereksinimlerini tahmin etmek**

### **Adım 5. Kullanıcı görünümlerini tasarlamak**

Yönetici sistemde ekleme, silme, güncelleme, listeleme ve arama yapabiliyorken normal bir kullanıcı arama işlemini ve ürün seçme işlemini yapabiliyor.

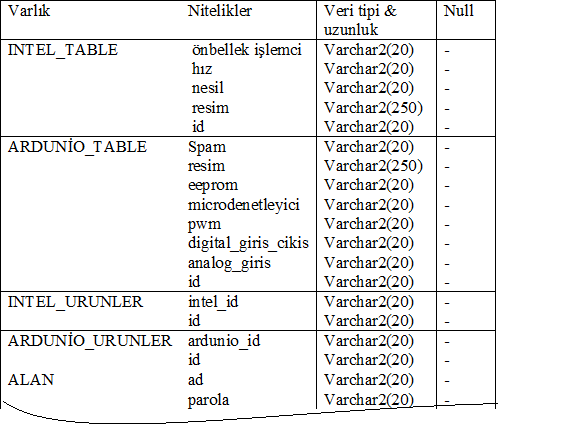
### **Adım 6. Güvenlik mekanizmalarını tasarlamak**

Yönetici giriş bölümü oluşturuldu ve bu şekilde yetkisi olmayan kişiler sisteme ekleme, silme, güncelleme ve listeleme gibi tablolar için önemli işlemleri herkesin yapması önlenmiş oldu.

Session ayarıyla da sistemde geri denerek önceki bulunulan sayfalara giriş önlendi.

**Adım 7. Operasyonel sistemi izlemek ve ayarlamak**

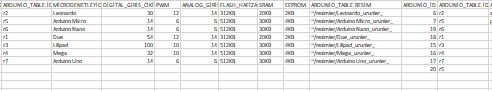
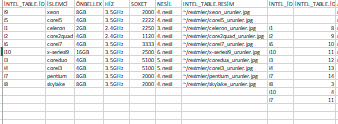
# **2.VERİ SÖZLÜĞÜ**



Şekil 2.1 Veri sözlüğü

# **3.NORMALİZASYON**

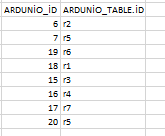
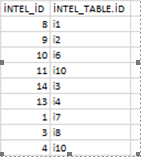
## **3.1.Normalleştirilmemiş Form (UNF)**



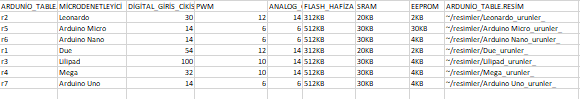
C:\Users\ALEMDAR\Desktop\unnf3.PNG

## **3.2.Birincil Normal Form (1NF)**

Her sütunda bir özellik bulunmakta. Aynı tipte faklı özellikler aynı sütunda bulunmuyor. Her sütundan bir tane bulunmakta. Veri girişinde sıranın önemi yok. Boş olan veri alanları bulunmamakta.

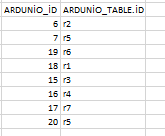
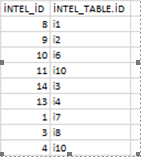




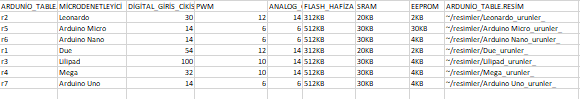
nf2

## **3.3.İkincil Normal Form (2NF)**

1 NF ‘a uyumlu. Sütunların kısmi bağımlılığı bulunmamakta.

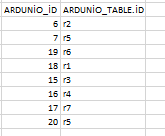
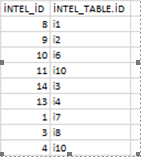




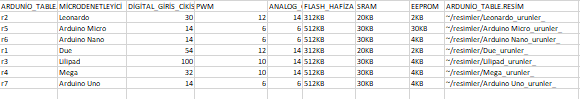
**C:\Users\ALEMDAR\AppData\Local\Microsoft\Windows\INetCache\Content.Word\nf2.5.png**

## **3.4. Üçüncü Normal Form (3NF)**

2 NF ‘a uyumlu. Geçişli bağımlılık yoktur.

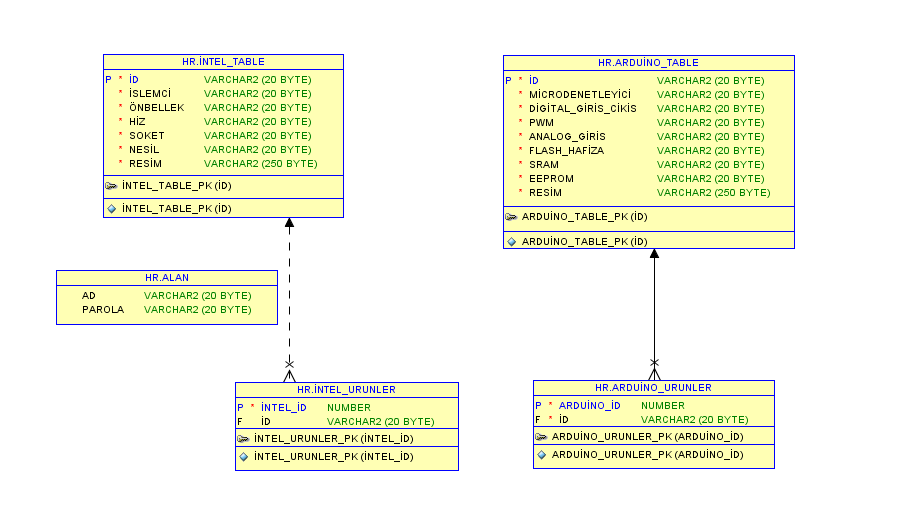
 





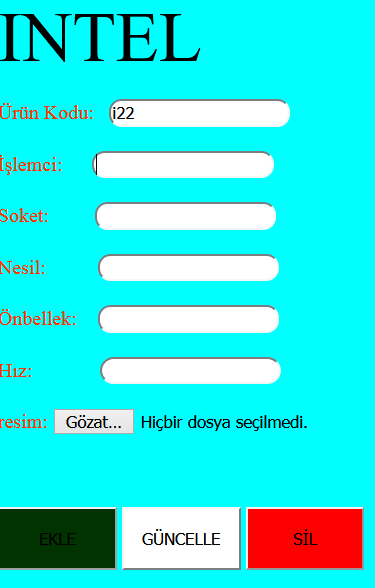
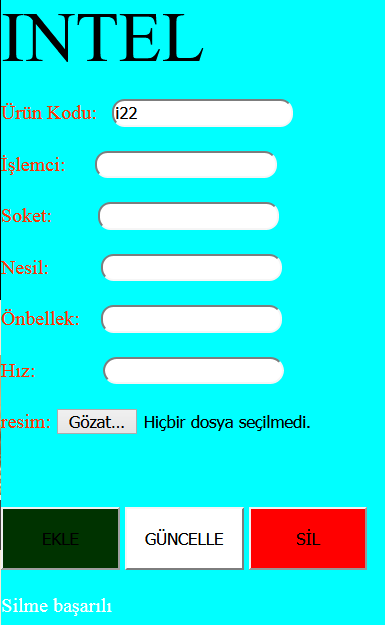
**C:\Users\ALEMDAR\AppData\Local\Microsoft\Windows\INetCache\Content.Word\nf2.5.png**

# **4.ER DİAGRAMI**



# **5.PROJE EKRAN GÖRÜNTÜLERİ**

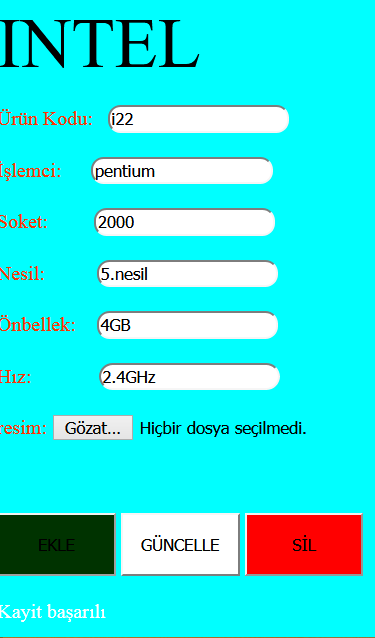
## **5.1.Kayıt Silme Ekran Görüntüsü**

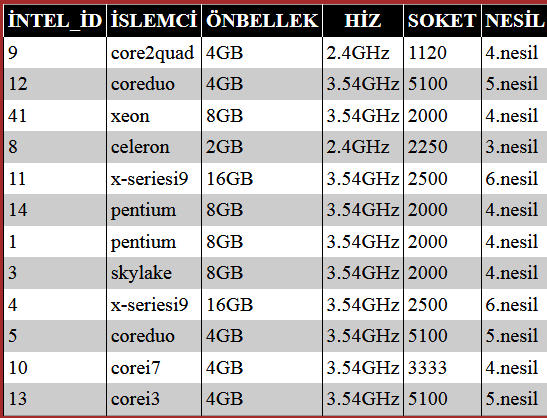
## **5.2.Kayıt Güncelleme Ekran Görüntüsü**

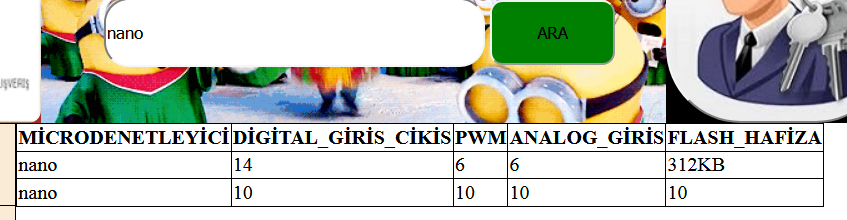
## **5.3.Kayıt Güncelleme Ekran Görüntüsü**

## **5.4.Kayıt Listeleme Ekran Görüntüsü**



## **5.5.Kayıt Arama Ekran Görüntüsü**



# **6.PROJE KODLARI**

## **6.1.Kütüphaneler Eklendi**

using Oracle.ManagedDataAccess.Client;

using System.Data;

using System.IO;

namespace ChipCountry

{

public partial class arduino : System.Web.UI.Page

{

## **6.2.Bağlantılar Tanımlandı**

OracleConnection baglanti = new OracleConnection();

OracleCommand sorgu = new OracleCommand();

OracleDataAdapter adtr = new OracleDataAdapter();

DataSet ds = new DataSet();

String uzanti, resimAdi;

protected void Page\_Load(object sender, EventArgs e)

{

}

## **6.3.Güncelleme Kodu**

protected void btnguncelle\_Click(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

uzanti = Path.GetExtension(FileUpload1.PostedFile.FileName);

resimAdi = (txtmikro.Text) + "\_urunler\_" + uzanti;

FileUpload1.SaveAs(Server.MapPath("resimler/" + uzanti));

FileUpload1.SaveAs(Server.MapPath("resimler/" + resimAdi));

FileInfo figecici = new FileInfo("~/resimler/" + resimAdi);

try

{

if (baglanti.State == ConnectionState.Open)

baglanti.Close();

baglanti.ConnectionString = "DATA SOURCE=WIN-4OIQSKFVRQ9:1521/XE;PERSIST SECURITY INFO=True;USER ID=HR; PASSWORD=HR";

baglanti.Open();

OracleCommand sorgu = new OracleCommand();

sorgu.CommandText = "update ARDUİNO\_TABLE set MİCRODENETLEYİCİ='" + txtmikro.Text + "',DİGİTAL\_GİRİS\_CİKİS='" + txtdigital.Text + "',PWM='" + txtpwm.Text + "',ANALOG\_GİRİS='" + txtanalog.Text + "' ,FLASH\_HAFİZA='" + txtflash.Text + "',SRAM='" + txtsram.Text + "',EEPROM='" + txteeprom.Text + "',RESİM='" + figecici + "' where İD='" + txtid.Text + "'";

sorgu.Connection = baglanti;

sorgu.ExecuteNonQuery();

baglanti.Close();

kayit.Text = "Güncelleme başarılı";

}

catch (Exception hata)

{

kayit.Text = "hata" + hata;

}

}

}

## **6.4.Silme Kodu**

protected void btnsil\_Click1(object sender, EventArgs e)

{

try

{

if (baglanti.State == ConnectionState.Open)

baglanti.Close();

baglanti.ConnectionString = "DATA SOURCE=WIN-4OIQSKFVRQ9:1521/XE;PERSIST SECURITY INFO=True;USER ID=HR; PASSWORD=HR";

baglanti.Open();

OracleCommand sorgu = new OracleCommand();

sorgu.Connection = baglanti;

sorgu.CommandText = "delete from ARDUİNO\_TABLE where ARDUİNO\_TABLE.İD IN(select İD from ARDUİNO\_URUNLER where ARDUİNO\_URUNLER.İD LIKE ' + txtid.Text + ')";

sorgu.ExecuteNonQuery();

baglanti.Close();

kayit.Text = "Silme başarılı";

}

catch (Exception hata)

{

kayit.Text = "hata" + hata;

}

}

## **6.5.Kayıt Ekleme Kodu**

protected void btnEkle\_Click(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

uzanti = Path.GetExtension(FileUpload1.PostedFile.FileName);

resimAdi = (txtmikro.Text) + "\_urunler\_" + uzanti;

FileUpload1.SaveAs(Server.MapPath("resimler/" + uzanti));

FileUpload1.SaveAs(Server.MapPath("resimler/" + resimAdi));

FileInfo figecici = new FileInfo("~/resimler/" + resimAdi);

try

{

if (baglanti.State == ConnectionState.Open)

baglanti.Close();

baglanti.ConnectionString = "DATA SOURCE=WIN-4OIQSKFVRQ9:1521/XE;PERSIST SECURITY INFO=True;USER ID=HR; PASSWORD=HR";

baglanti.Open();

OracleCommand sorgu = new OracleCommand();

sorgu.CommandText = "insert into ARDUİNO\_TABLE(İD,MİCRODENETLEYİCİ,DİGİTAL\_GİRİS\_CİKİS,PWM,ANALOG\_GİRİS,FLASH\_HAFİZA,SRAM,EEPROM,RESİM) values('" + txtid.Text + "','" + txtmikro.Text + "','" + txtdigital.Text + "','" + txtpwm.Text + "','" + txtanalog.Text + "','" + txtflash.Text + "','" + txtsram.Text + "','" + txteeprom.Text + "','" + figecici + "')";

sorgu.Connection = baglanti;

sorgu.ExecuteNonQuery();

baglanti.Close();

kayit.Text = "Kayit başarılı";

}

catch (Exception hata)

{

kayit.Text = "hata" + hata;

}

}

}

}

}

## **6.6.Listeleme Kodu**

cnn.ConnectionString = "DATA SOURCE=WIN-4OIQSKFVRQ9:1521/XE;PERSIST SECURITY INFO=True;USER ID=HR; PASSWORD=HR";

            string sorgu = "select u.ARDUİNO\_İD, a.MİCRODENETLEYİCİ,a.DİGİTAL\_GİRİS\_CİKİS,a.PWM,ANALOG\_GİRİS,a.FLASH\_HAFİZA,a.SRAM,a.EEPROM   from ARDUİNO\_TABLE a inner  join ARDUİNO\_URUNLER u ON a.İD=u.İD";

            OracleDataAdapter sadp = new OracleDataAdapter(sorgu, cnn);

            DataSet ds = new DataSet();

            sadp.Fill(ds);

            this.GridView1.DataSource = ds.Tables[0];

            this.GridView1.DataBind();

## **6.7.Arama Kodu**

 protected void btnAra\_Click(object sender, EventArgs e)

        {

            baglanti.ConnectionString = "DATA SOURCE=WIN-4OIQSKFVRQ9:1521/XE;PERSIST SECURITY INFO=True;USER ID=HR; PASSWORD=HR";

            try

            {

                String sorgu = "Select İSLEMCİ,ÖNBELLEK,HİZ,SOKET,NESİL from İNTEL\_TABLE where İSLEMCİ LIKE '%" + txtArama.Text + "%'";

                OracleCommand cmd = new OracleCommand(sorgu, baglanti);

                baglanti.Open();

                OracleDataAdapter sadp = new OracleDataAdapter(sorgu, baglanti);

                DataSet ds = new DataSet();

                sadp.Fill(ds);

                Grid.DataSource = ds.Tables[0];

                Grid.DataBind();

                String sorgu2 = "Select MİCRODENETLEYİCİ,DİGİTAL\_GİRİS\_CİKİS,PWM,ANALOG\_GİRİS,FLASH\_HAFİZA from ARDUİNO\_TABLE where MİCRODENETLEYİCİ LIKE '%" + txtArama.Text + "%'";

                OracleCommand cmd2 = new OracleCommand(sorgu2, baglanti);

                OracleDataAdapter sadp2 = new OracleDataAdapter(sorgu2, baglanti);

                DataSet ds2 = new DataSet();

                sadp2.Fill(ds2);

                Grid2.DataSource = ds2.Tables[0];

                Grid2.DataBind();

            }

            catch (Exception hata)

            {

                kayit.Text = "hata" + hata;

            }

        }