**İSTANBUL GELİŞİM ÜNİVERSİTESİ İSTANBUL GELİŞİM MESLEK YÜKSEKOKULU BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ BİLGİSAYAR PROGRAMCILIĞI PROGRAMI**

**LİSE VERİ TABANI OLUŞTURMA VE SORGULARI**

**FİNAL PROJE ÖDEVİ**

**210111295 – Eren ÇITAK**

**220111447 Emre KELEŞ**

**210111267 Ceren DENİZ**

**Ödev Danışmanı Öğr.Gör. NAZMİYE İNAM**

**İSTANBUL – 2022**

# ÖDEV TANITIM FORMU

**YAZAR ADI SOYADI :** Eren Çıtak

**ÖDEVİN DİLİ :** Türkçe

**ÖDEVİN ADI :**  LİSE VERİ TABANI OLUŞTURMA VE SORGULARI

**BÖLÜM :** Bilgisayar Teknolojileri

**PROGRAM :** Bilgisayar Programcılığı

**ÖDEVİN TÜRÜ :** Final

**ÖDEVİN TES. TARİHİ :** 18.05.2022

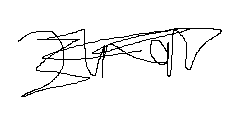
**ÖDEV DANIŞMANI :** Öğr.Gör. NAZMİYE İNAM

# BEYAN

Bu ödevin/projenin hazırlanmasında bilimsel ahlak kurallarına uyulduğu, başkalarının ederlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğu, kullanılan verilerde herhangi tahrifat yapılmadığını, ödevin/projenin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir ödev/proje olarak sunulmadığını beyan eder, aksi durumda karşılaşacağım cezai ve/veya hukuki durumu kabul eder; ayrıca üniversitenin ilgili yasa, yönerge ve metinlerini okuduğumu beyan ederim.

17.05.2022

Adı Soyadı

Eren ÇITAK

# KABUL VE ONAY SAYFASI

**210111295** numaralı **Eren ÇITAK**’ın **LİSE VERİ TABANI OLUŞTURMA VE SORGULARI** adlı çalışması, benim tarafımdan Vize/Ders içi/Final ödevi olarak kabul edilmiştir.

Öğretim Görevlisi

Nazmiye İNAM

# ÖZET

Yapmış olduğumuz projede bir lisenin veri tabanını yeterli seviyede yapmaya çalıştık.

# Proje Nasıl Tasarlandı

Projemizin amacı bir lise veri tabanı oluşturmak. Bunu için okul ile alakalı varlıkları düşündük ve aklımıza Öğretmen, Öğrenci, Personel, Okul, Ders Programı ve Dersler geldi bunları varlık olarak varlık bağıntı çizeneğimize ekledik. Ardından bunlara kendi içinde nitelik eklemeye başladık sadece Personel, Okul, Öğretmen ve Dersler tablosuna primary key olacaktı diğer tablolara primary key ekleme gereği duymadık. Bu koymuş olduğumuz primary keyler ise şu şekilde idi;

Dersler tablosundaki ‘Ders\_ID’ , Öğretmen tablosundaki ‘Hoca \_No’ , Personel tablosundaki ‘Gorevli\_ID’ ve son olarak Okul tablosundaki ‘Sinif\_No’.

Bu işlemler yapıldıktan sonra aralındaki eksik olan bağıntıları yerleştirdik ve varlık bağıntı çizeneğimizi bu şekilde tamamladık.

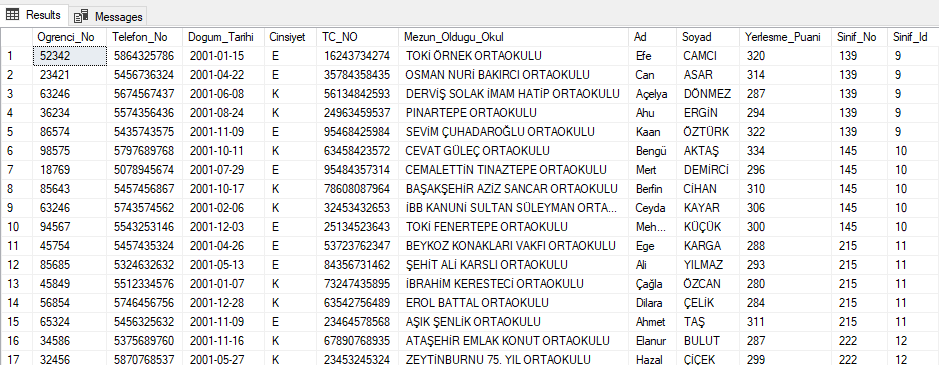
Şimdi ise SQL dosyamızı oluşturarak bu planlamış olduğumuz varlıkları ve nitelikleri hepsi not null olucak şekilde yeni bir query dosyasına yazdık . Gerekli öğrenci, öğretmen, personel, okul, ders programı ve dersler verilerini girip sorgu yazamaya hazır bir hale getirdik. Sorgu kısmına geçmeden önce projenin bu kısmının nasıl tasarlandığı hakkında biraz bilgi vermek istiyorum. Öğrenci tablosunda normal bir öğrencini bilgileri bulunuyor onun haricinde ‘Sinif\_No’ ve ‘Sinif\_Id’ diye 2 tane varlığımız var. Bunlardan ‘Sinif\_Id’ dediğimiz nitelik o sınıfın kaçıncı sınıf olduğunu belirtiyor (9.-10.-11. veya 12. Sınıf olduklarını). ‘Sinif\_No‘ diye isimlendirilen nitelik ise o sınıfın hangi derslikte gireceğini gösteren bir nitelik. Okul varlığını ise her dersliğe sadece 1 tane personel atayacak şekilde tasarladık ve o şekilde birleştirdik. Personel varlığına gelince her bir personele kendine ait bir numara verdik bu sayede ‘Okul’ varlığı ile aralarındaki ilişkiyi kolayca kurabildik. Aynı şekilde öğretmen tablomuzu da o mantıkla oluşturduk ve her öğretmene kendine ait bir ‘Hoca\_No’ verdik bu sayede ‘Ogretmen’ ve ‘Ders\_Programi’ varlıklarına birleştirebildik.

Son olarak ‘Dersler’ tablomuzu da her derse ait bir numara vererek (1-Matematik , 2-Fizik vb..) ‘Ders\_Programi’ varlığı ile arasındaki bağıntıyı halletmiş olduk.

Yapmış olduğumuz projeyi bu şekilde tasarladık.

# SQL SORGULARI

select \* from Ogrenci



select Ad,Soyad,Hoca\_No from Ogretmen



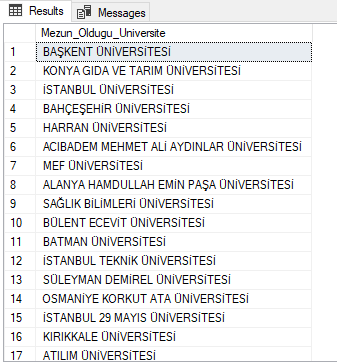
select Ad,Soyad from Ogretmen where Cinsiyet='E'



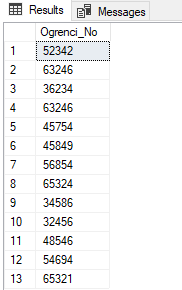
select \* from Ogretmen where Ad like 'a%'



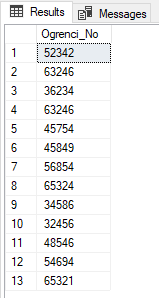
select Mezun\_Oldugu\_Universite from Ogretmen where Soyad like '%e%'



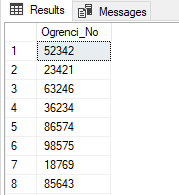
select Ogrenci\_No from Ogrenci where Ogrenci\_No>=25000 and Ogrenci\_No<=75000



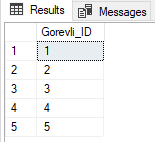
select Ogrenci\_No from Ogrenci where Ogrenci\_No between 25000 and 75000



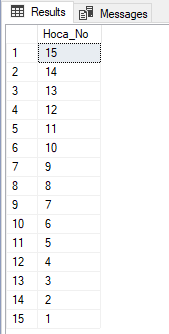
select Ogrenci\_No from Ogrenci where Ogrenci\_No>=30000 or Ogrenci\_No<=80000



select Gorevli\_ID from Personel where Gorevli\_ID<=10 order by Gorevli\_ID asc



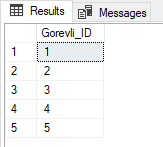
select Hoca\_No from Ogretmen where Hoca\_No<=15 order by Hoca\_No desc



select Gorevli\_ID from Personel

union

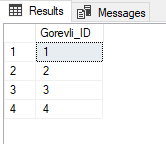
select Gorevli\_ID from Okul



select Gorevli\_ID from Personel

intersect

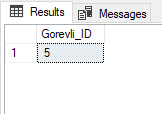
select Gorevli\_ID from Okul



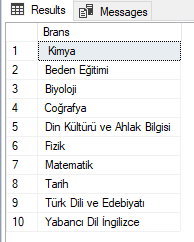
select Gorevli\_ID from Personel

except

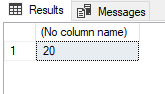
select Gorevli\_ID from Okul



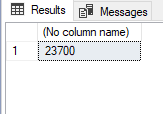
select distinct (Brans) from Ogretmen



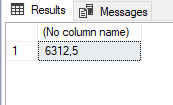
select count(Ogrenci\_No) from Ogrenci



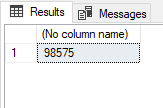
select sum(Maas) from Personel



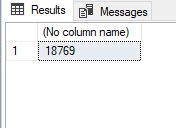
select avg(Maas) from Ogretmen



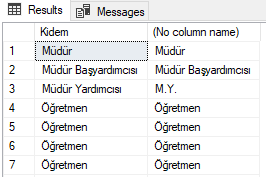
select max(Ogrenci\_No) from Ogrenci



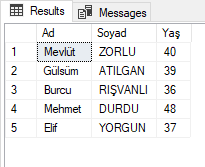
select min(Ogrenci\_No) from Ogrenci



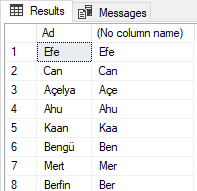
select Kidem, replace (Kidem,'Müdür Yardımcısı','M.Y.') from Ogretmen



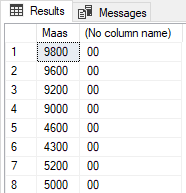
select Ad,Soyad, datediff(year, Dogum\_Tarihi,getdate()) as 'Yaş' from Personel



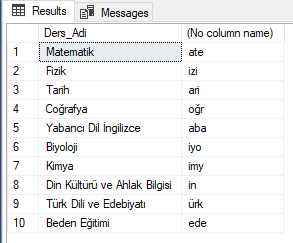
select Ad, left (Ad,3) from Ogrenci



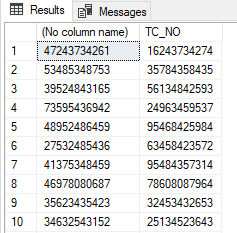
select Maas, right (Maas,2) from Ogretmen



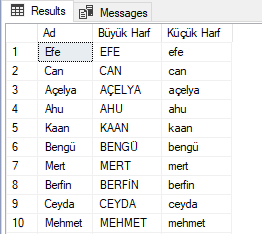
select Ders\_Adi, substring(Ders\_Adi,2,3) from Dersler



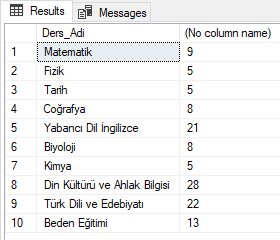
select reverse (TC\_NO) , TC\_NO from Ogrenci



select Ad, upper(Ad) as 'Büyük Harf', lower (Ad) as 'Küçük Harf' from Ogrenci



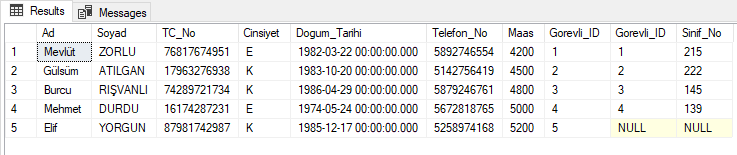
select Ders\_Adi, len(Ders\_Adi) from Dersler



select \* from Ogretmen INNER JOIN Ders\_Programi on Ogretmen.Hoca\_No = Ders\_Programi.Hoca\_No



select \* from Personel left outer join Okul on Personel.Gorevli\_ID = Okul.Gorevli\_ID



select \* from Dersler right outer join Ders\_Programi on Dersler.Ders\_ID = Ders\_Programi.Ders\_ID

