



ONDOKUZ MAYIS ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# VERİTABANI LABORATUVARI

## FÖY 2 RAPOR

ZEYNEP SILA KAYMAK

20060358

DERSİN ÖĞRETMENİ

Dr.Öğr.Üyesi İsmail İŞERİ

MART, 2024

---

## TEŞEKKÜR

---

Dersimizin öğretmeni sayın Dr.Öğr.Üyesi İsmail İŞERİ'ye ve dersimizin asistanı sayın Arş.Gör. Kübra SEYHAN'a teşekkürlerimi sunarım.

---

## Ö Z E T

---

Proje, bir veritabanının SQL Server kullanılarak oluşturulması, varlık ilişki diyagramının çizilmesi, ilişkilerin tanımlanması, normalizasyon sürecinin uygulanması ve veritabanı yönetimi konularının kapsandığı bir süreçte gerçekleştirilmiştir. Ayrıca, veritabanı güvenliğinin sağlanması amacıyla kullanıcı tanımlamaları yapılmıştır.

İlk aşamada, SQL Server kurulumu tamamlanmış ve SQL Server Management Studio (SSMS) aracılığıyla yeni bir veritabanı oluşturulmuştur. Bu aşamada, proje gereksinimlerine uygun olarak beş farklı tablo oluşturulmuş ve her bir tablo için çeşitli veri tipleri belirlenmiştir.

Daha sonra, varlık ilişki diyagramı çizilerek oluşturulan tablolar arasındaki ilişkiler görselleştirilmiştir. Bu diyagram, veritabanının yapısını ve ilişkileri anlamak için önem taşımaktadır. Varlık ilişki diyagramında belirlenen ilişkiler, SQL dilindeki FOREIGN KEY kısıtlamaları kullanılarak veritabanı içinde tanımlanmıştır.

Ardından, normalizasyon süreci gerçekleştirilerek oluşturulan tabloların normalizasyonu sağlanmıştır. Bu süreç, veritabanının daha etkin, verimli ve tekrarlanan verilerden arındırılmış bir yapıya sahip olmasını amaçlamaktadır. İlk olarak, veritabanındaki tabloların normal formlara uygun olup olmadığı incelenmiş ve gerekli düzeltmeler yapılmıştır.

---

## İÇİNDEKİLER

---

### I GİRİŞ

1 Amaç	2
--------	---

### II Yöntem

2 Soru 1	4
3 Soru 2	6
4 Soru 3	14
5 Soru 4	16

### III SONUÇ

6 Sonuç	20
---------	----

---

## ŞEKİL LİSTESİ

---

---

## TABLO LİSTESİ

---

## BÖLÜM: İ

### GİRİŞ

## A M A Ç

---

Projenin amacı, varlık-ilişki diyagramı çizme, ilişkilerin tanımlanması, normalizasyon süreci gibi temel kavramları anlamak ve bu kavramları uygulamak suretiyle veritabanı performansını artırmak ve veri bütünlüğünü sağlamaktır. Ayrıca, kullanıcı tanımlamaları ve yetkilendirme gibi güvenlik önlemleri de ele alınarak, veri güvenliği ve erişim kontrolü sağlanır. Bu proje aynı zamanda proje yönetimi ve iletişim becerilerini geliştirirken gerçek dünya uygulamalarıyla veri yönetimi alanında deneyim kazandırır.



BÖLÜM: İİ

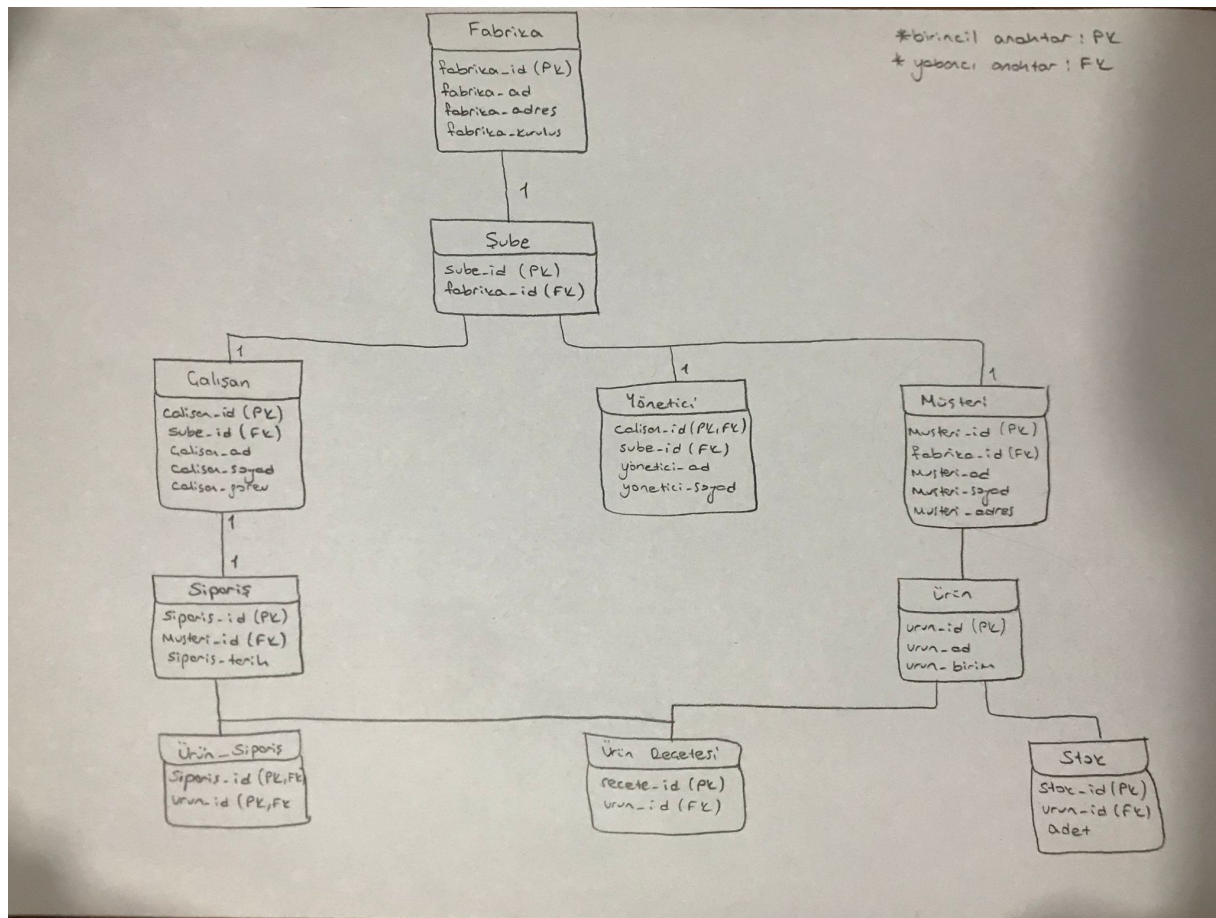
YÖNTEM

---

## SORU 1

---

Veri tabanı tasarımı sürecinde, bir fabrikanın şubeleri, çalışanları, müşterileri ve siparişlerini kapsayan detaylı bir Varlık-İlişki diyagramı oluşturuldu. Bu diyagram, her varlığın özelliklerini ve ilişkilerini net bir şekilde ortaya koymak için tasarlandı. Ana varlıkların yanı sıra, ilişkilerin doğru şekilde tanımlanması ve çoklukların belirlenmesi büyük önem taşıdı. Her varlık için birincil anahtarlar ve ilişkiler arasındaki bağlantılar açık bir şekilde belirtildi. Bu sayede, veri tabanının veri bütünlüğü ve tutarlılığı sağlanarak, fabrikanın işleyişine dair verilerin etkin bir şekilde yönetilmesi hedeflendi.



---

## SORU 2

---

Talep edilen tabloların tasarımı, Microsoft SQL Server Management Studio kullanılarak gerçekleştirildi. Bu süreçte her bir tablonun sütunları ve veri tipleri belirlendi, birincil anahtarlar ve yabancı anahtarlar ile ilişkiler düzgün bir şekilde tanımlandı. Ardından, tablolar arasındaki ilişkilerin kurulması için gerekli olan yabancı anahtar kısıtlamaları eklenerek veri bütünlüğü sağlandı. Son olarak, oluşturulan tablolar veritabanına eklenerek gerekli indeksler oluşturuldu. Bu adımların tamamlanmasıyla, Microsoft SQL Server’da çalışacak bir veritabanı tasarımı oluşturulmuş oldu.

SQL dilinde veritabanı tablolarının oluşturulması için CREATE TABLE ifadeleri kullanıldı. Her bir tablo, belirli bir varlık veya ilişkiyi temsil etmektedir.

Fabrika, fabrikanın temel bilgilerini tutar. fabrika-id birincil anahtar olarak kullanılır ve her fabrikanın eşsiz bir kimliği vardır. fabrika-ad ve fabrika-adres, fabrikanın adını ve adresini tutar. fabrika-kurulus, fabrikanın kuruluş tarihini tutar.

Sube, her bir fabrikanın şubelerini temsil eder. sube-id birincil anahtar olarak kullanılır ve her bir şubenin eşsiz bir kimliği vardır. fabrika-id, şubenin hangi fabrikaya ait olduğunu belirtir.

Calisan, şubelerde çalışanların bilgilerini tutar. calisan-id birincil anahtar olarak kullanılır ve her bir çalışanın eşsiz bir kimliği vardır. sube-id, çalışanın hangi şubede çalıştığını belirtir. Diğer alanlar çalışanın adı, soyadı ve görevini tutar.

Siparis, müşterilerin verdiği siparişleri temsil eder. siparis-id birincil anahtar olarak kullanılır ve her bir siparişin eşsiz bir kimliği vardır. musteri-id, siparişi veren müşteriyi belirtir. siparis-tarih, siparişin alındığı tarihi tutar.

Urun-Siparis, siparişler ile ürünler arasındaki ilişkiyi temsil eder. Her bir siparişin hangi ürünleri içerdiğini belirtir. siparis-id ve urun-id, ilişkinin anahtarlarını oluşturur.

Urun-Recetesi, ürünlerin üretiminde kullanılan malzemelerin listesini tutar. recete-id birincil anahtar olarak kullanılır ve her bir ürün reçetesinin eşsiz bir kimliği vardır. urun-id, reçetenin hangi ürüne ait olduğunu belirtir.

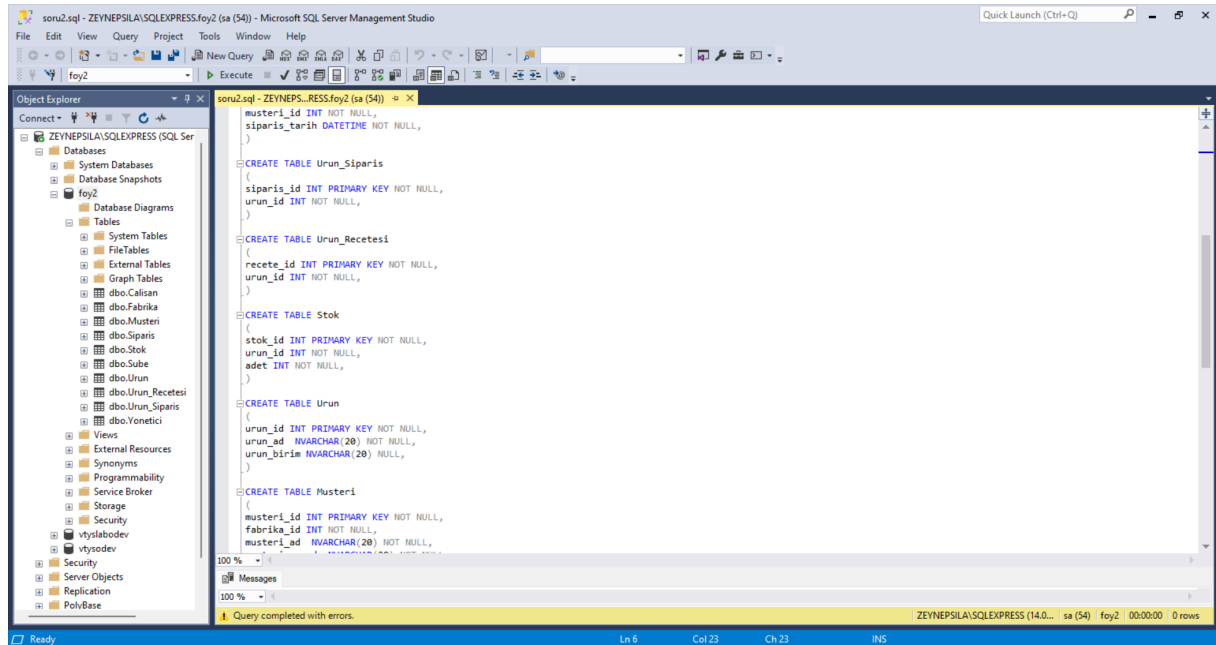
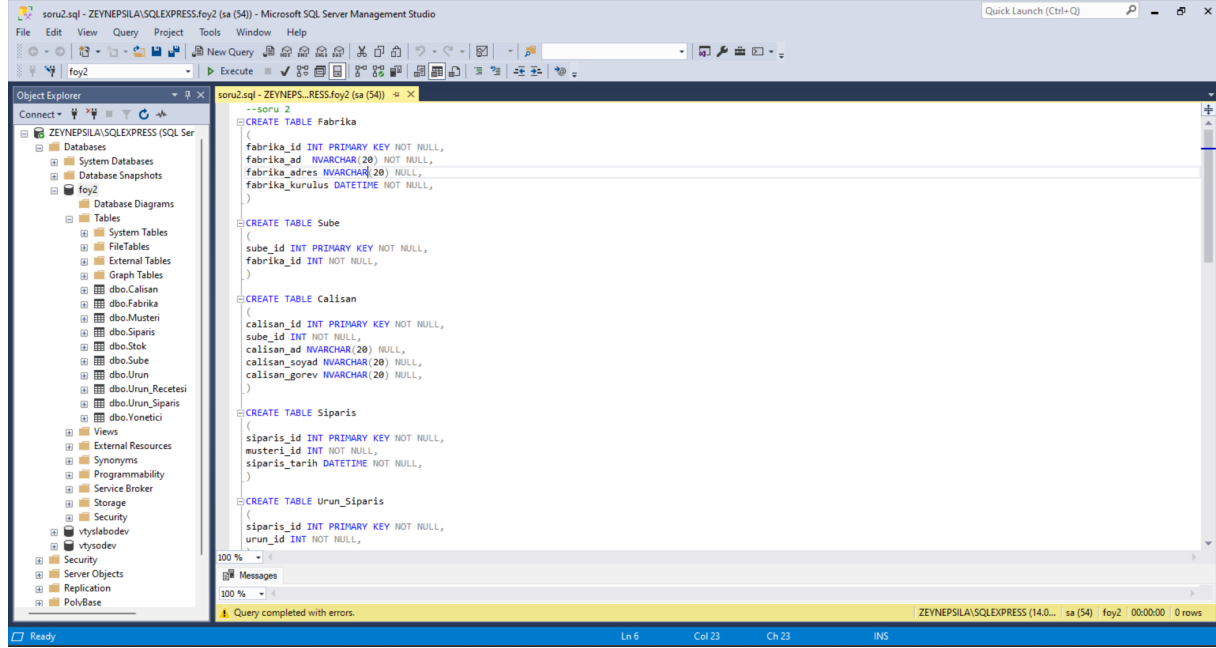
Stok, fabrikanın stok bilgilerini tutar. stok-id birincil anahtar olarak kullanılır ve her bir stok kaleminin eşsiz bir kimliği vardır. urun-id, stoktaki ürünü belirtir. adet, stoktaki ürün miktarını tutar.

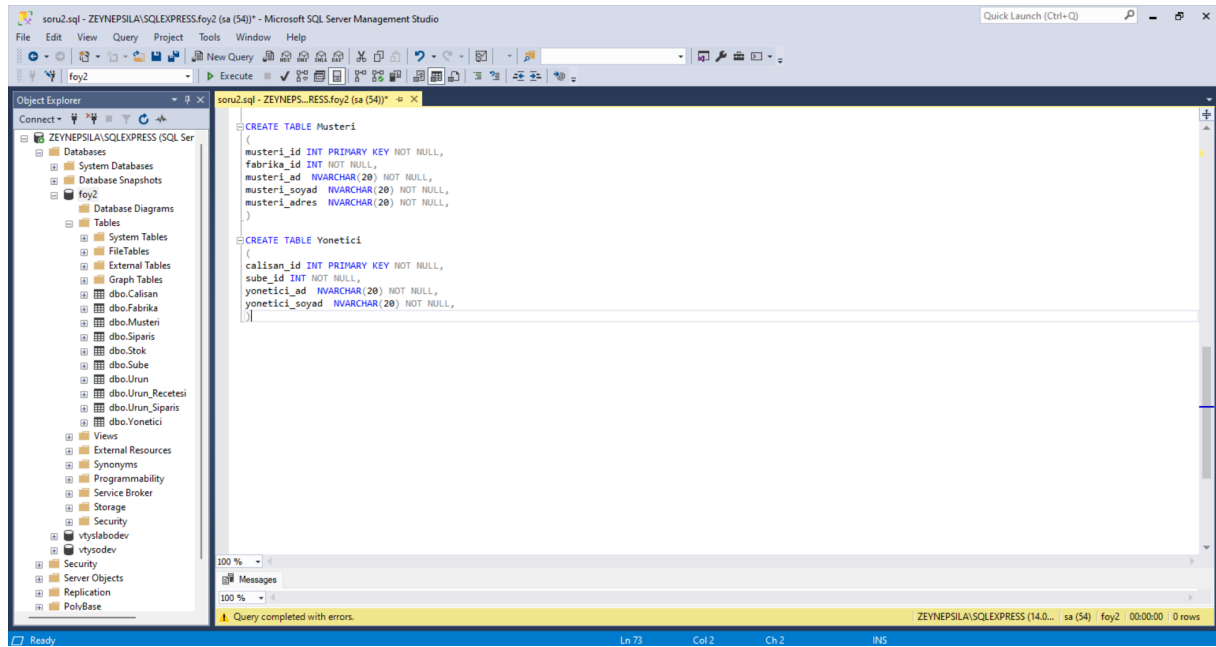
Urun, fabrikada üretilen ürünlerin bilgilerini tutar. urun-id birincil anahtar olarak kullanılır ve her bir ürünün eşsiz bir kimliği vardır. urun-ad, ürünün adını tutar. urun-birim, ürünün birimini (adet, gram vb.) tutar.

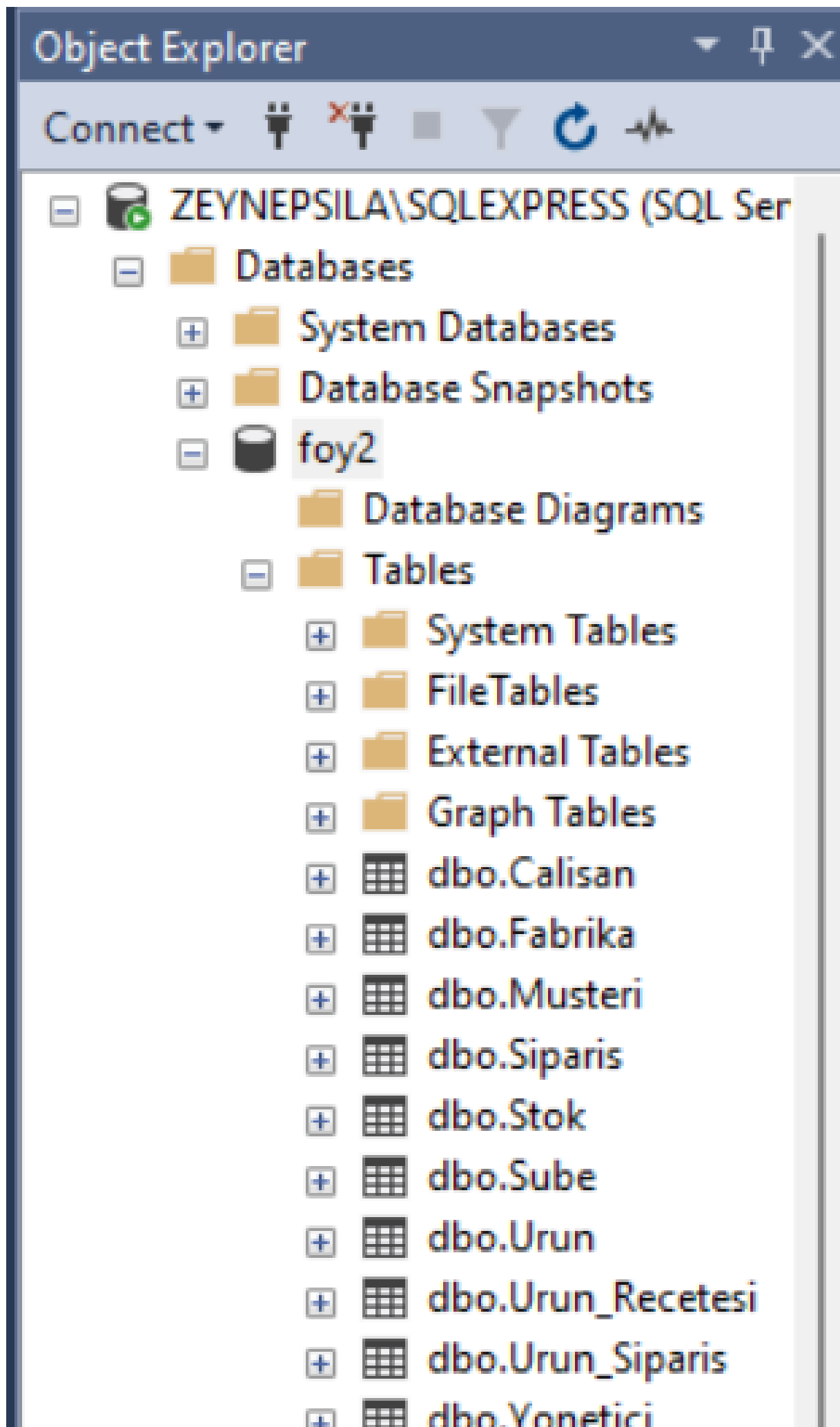
Musteri, fabrikadan ürün siparişi veren müşterilerin bilgilerini tutar. musteri-id birincil anahtar olarak kullanılır ve her bir müşterinin eşsiz bir kimliği vardır. fabrika-id, müşterinin hangi fabrikaya ait olduğunu belirtir. Diğer alanlar müşterinin adı, soyadı ve adresini tutar.

Yonetici, şubelerdeki yöneticilerin bilgilerini tutar. calisan-id birincil anahtar olarak kullanılır ve her bir yöneticinin eşsiz bir kimliği vardır. sube-id, yöneticinin hangi şubede olduğunu belirtir. Diğer alanlar yöneticinin adı ve soyadını tutar.

Bu tablolar, fabrikanın şubelerini, çalışanlarını, müşterilerini, siparişlerini, ürünlerini, stoklarını ve üretim reçetelerini temsil eden veritabanı şemasını oluşturur. Bu sayede fabrikanın faaliyetlerine dair kapsamlı bir veri tabanı sağlanır.









[-]	[-]	dbo.Calisan
	[-]	Columns
		calisan_id (PK, int, not null)
		sube_id (int, not null)
		calisan_ad (nvarchar(20), null)
		calisan_soyad (nvarchar(20), null)
		calisan_gorev (nvarchar(20), null)
	[-]	Keys
		PK_Calisan_927226D3CD2D9E04

+	[-]	dbo.Calisan
[-]	[-]	dbo.Fabrika
	[-]	Columns
		fabrika_id (PK, int, not null)
		fabrika_ad (nvarchar(20), not null)
		fabrika_adres (nvarchar(20), null)
		fabrika_kurulus (datetime, not null)
	[-]	Keys
		PK_Fabrika_EF5AF565B2C32035

[-]	[-]	dbo.Fabrika
[-]	[-]	dbo.Musteri
	[-]	Columns
		musteri_id (PK, int, not null)
		fabrika_id (int, not null)
		musteri_ad (nvarchar(20), not null)
		musteri_soyad (nvarchar(20), not null)
		musteri_adres (nvarchar(20), not null)
	[-]	Keys
		PK_Musteri_966D46E28CBB7952

+	[-]	dbo.Musteri
[-]	[-]	dbo.Siparis
	[-]	Columns
		siparis_id (PK, int, not null)
		musteri_id (int, not null)
		siparis_tarih (datetime, not null)
	[-]	Keys
		PK_Siparis_BB5E5880F66A4C65

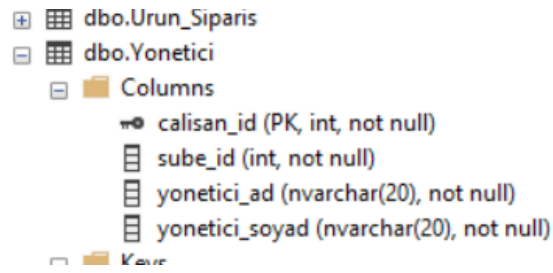
+ dbo.Siparis  
 - dbo.Stok  
   - Columns  
     - stok\_id (PK, int, not null)  
     - urun\_id (int, not null)  
     - adet (int, not null)  
   - Keys  
     - PK\_Stok\_08308A095B7985F0  
   - Constraints

+ dbo.Stok  
 - dbo.Sube  
   - Columns  
     - sube\_id (PK, int, not null)  
     - fabrika\_id (int, not null)  
   - Keys  
     - PK\_Sube\_64674BC395C61705

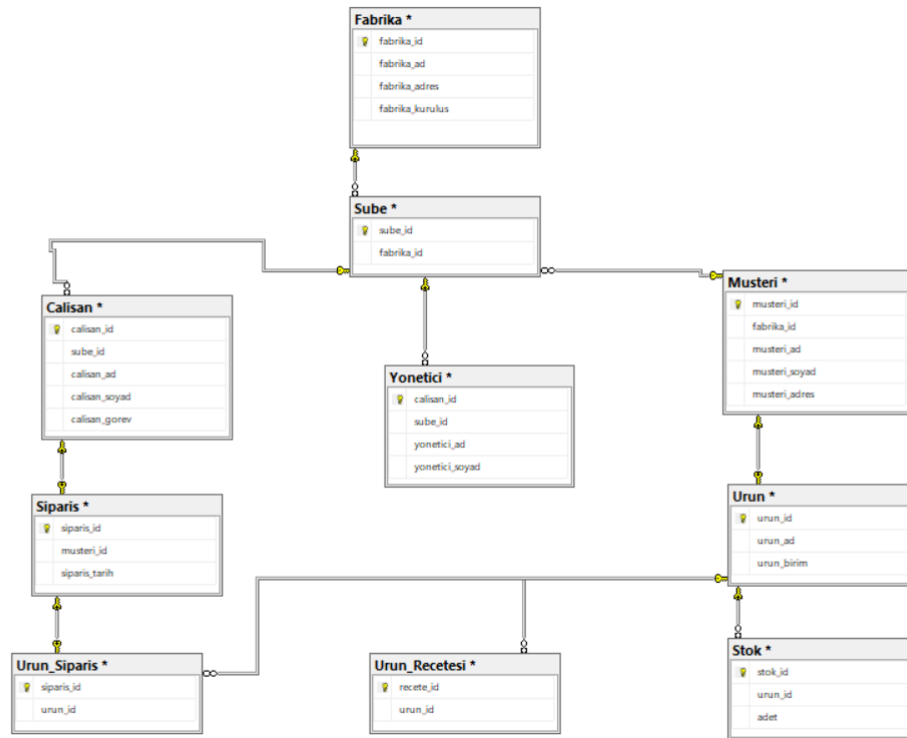
+ dbo.Sube  
 - dbo.Urun  
   - Columns  
     - urun\_id (PK, int, not null)  
     - urun\_ad (nvarchar(20), not null)  
     - urun\_birim (nvarchar(20), null)  
   - Keys  
     - PK\_Urun\_933C200ADFB3B6B7  
   - Constraints

- dbo.Urun\_Recetesi  
   - Columns  
     - recete\_id (PK, int, not null)  
     - urun\_id (int, not null)  
   - Keys  
     - PK\_Urun\_Rec\_6D069EA0B3DC6C22  
   - Constraints

- dbo.Urun\_Siparis  
   - Columns  
     - siparis\_id (PK, int, not null)  
     - urun\_id (int, not null)  
   - Keys  
     - PK\_Urun\_Sip\_BB5E588067F5B787  
   - Constraints



ZEYNEPSILA\SQLEXP....foy2 - Diagram\_0\* -> X soru2.sql - ZEYNEPS...RESS.foy2 (sa (54))\*



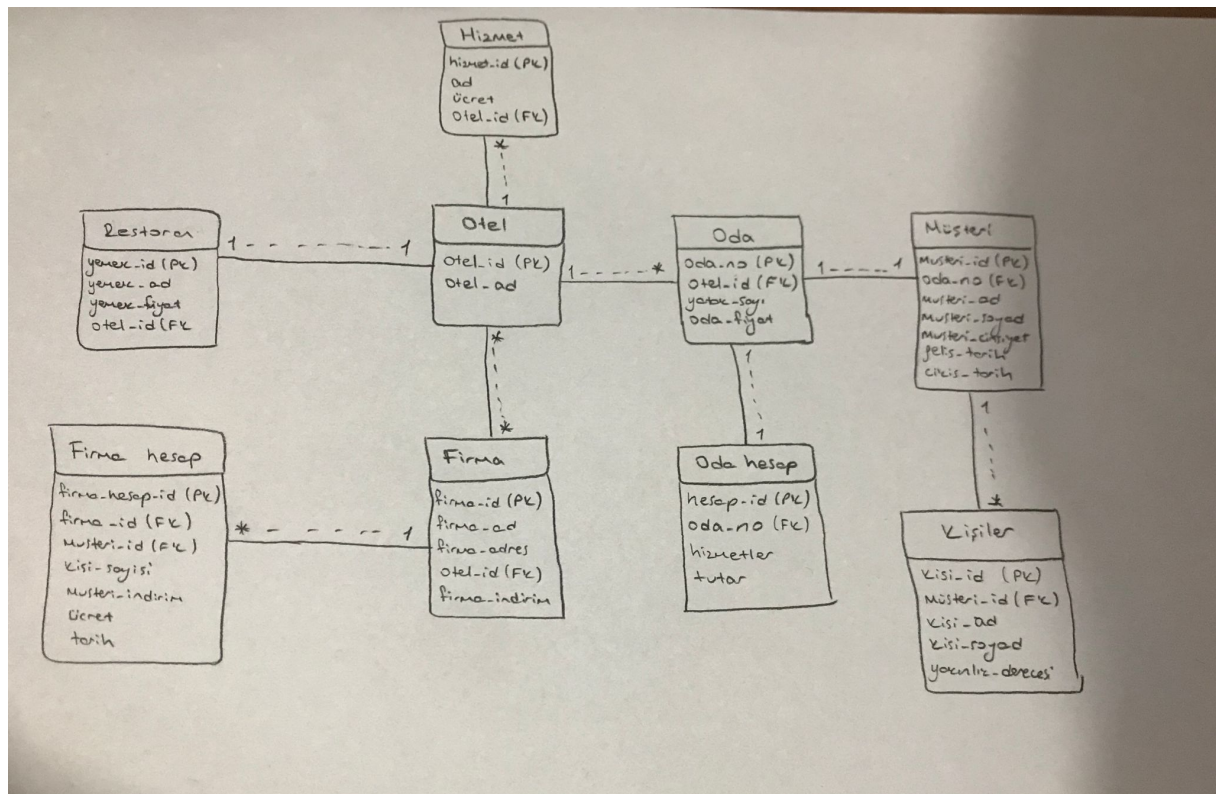
---

### SORU 3

---

Gereksinimlere uygun olarak otel veri tabanı tasarımı oluşturuldu. Bu tasarım, otelde bulunan odaları, hizmetleri, yemek ve içecekleri, müşterileri ve ilişkili firmaları kapsamaktadır. Her bir tablo belirli bir varlık veya ilişkiyi temsil etmektedir.

Bu tasarım, otelin işleyişine dair kapsamlı bir veri tabanı sağlar. Her bir tablo, otel yönetiminin odaları, hizmetleri, müşterileri ve ilişkili firmaları etkin bir şekilde yönetmesini sağlar. Ayrıca, müşteri hesapları ve firmalarla yapılan anlaşmalara ilişkin bilgilerin düzenli bir şekilde tutulmasını sağlar.



---

#### SORU 4

---

Tabloyu 3NF'ye dönüştürmek için normalleştirme adımları takip edilecek ve şu şekilde düzenlenecektir.

Öncelikle, verilen veriler tabloya dönüştürülecek.

Öğrenci No	Ad	Şehir	Bölge
100	ahmet	İstanbul	Marmara
101	deniz	Ankara	İç Anadolu
102	ali	Kars	Doğu Anadolu
103	yasemin	İstanbul	Marmara

Ders	Öğrenci No	Kayıt Tarihi
VTYS	100	01.01.2017
Mobil	100	20.05.2017
VTYS	101	21.05.2017
Web	101	21.05.2017
Mobil	102	22.05.2017
VTYS	103	03.06.2017

Daha sonra, normalleştirme adımları izlenecek.

Adım 1: Verilerdeki gereksiz tekrarları kaldırarak, ders ve kayıt tarihi alanlarına bağımlı olduğu için yeni bir tablo oluşturulacaktır.

Öğrenci No	Ad	Şehir	Bölge
100	ahmet	İstanbul	Marmara
101	deniz	Ankara	İç Anadolu
102	ali	Kars	Doğu Anadolu
103	yasemin	İstanbul	Marmara

Ders	Öğrenci No	Kayıt Tarihi
VTYS	100	01.01.2017
Mobil	100	20.05.2017
VTYS	101	21.05.2017
Web	101	21.05.2017
Mobil	102	22.05.2017
VTYS	103	03.06.2017

Adım 2: Kısmi bağımlılıkları kaldırmak için, anahtar olmayan veri alt kümeleri kaldırılacak ve tablo iki tabloya ayrılacaktır.

Öğrenci No	Ad	Şehir	Bölge
100	ahmet	İstanbul	Marmara
101	deniz	Ankara	İç Anadolu
102	ali	Kars	Doğu Anadolu
103	yasemin	İstanbul	Marmara

Bu adımları izleyerek veriler 3NF formuna getirilmiştir. Bu normalleştirme işlemi, veritabanı tasarımında verilerin daha düzenli ve etkili bir şekilde depolanmasını sağlar.

Ders	Öğrenci No	Kayıt Tarihi
VTYS	100	01.01.2017
Mobil	100	20.05.2017
VTYS	101	21.05.2017
Web	101	21.05.2017
Mobil	102	22.05.2017
VTYS	103	03.06.2017



## BÖLÜM: İİİ

### SONUÇ

---

## SONUÇ

---

Bu ödevde varlık-ilişki diyagramı çizilmiş, ilişkiler tanımlanmış, normalizasyon süreci uygulanmış ve veritabanı yönetimi konusunda gerekli adımlar atılmıştır. Sonuç olarak elde edilen veritabanı, organizasyonun veri yönetim ihtiyaçlarını karşılamak için etkili bir araç oluşturulmuştur. Veritabanı, organizasyonun şubeleri, çalışanları ve müşterileri gibi önemli varlıklarının verilerini düzenli bir şekilde saklamak, güncellemek ve erişmek için bir platform sağlamaktadır. Ayrıca, veritabanının normalizasyon sürecinden geçirilmesi, veri bütünlüğünün sağlanmasına ve veritabanı performansının artırılmasına katkıda bulunmuştur.