

**T.C.**  
**SELÇUK ÜNİVERSİTESİ**  
**TEKNOLOJİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**VERİ TABANI YÖNETİM SİSTEMLERİ DERSİ FİNAL PROJESİ**

**ÖZEL SAĞLIK KLİNİĞİ OTOMASYONU UYGULAMASI**

Projeyi Hazırlayan Öğrenci

**Kaan PULAT**

**Mayıs, 2025**

# İÇİNDEKİLER

1. Sistem Gereksinim Analizi ve Kavramsal Modelin Oluşturulması.....	4
2. Tabloların Normalizasyon İşlemleri.....	6
2.1. “Hasta” Tablosunun Normalizasyon İşlemleri.....	7
2.2. “Doktor” Tablosunun Normalizasyon İşlemleri.....	10
2.3. “Randevular” Tablosunun Normalizasyon İşlemleri.....	12
2.4. “MuayeneKayitlari” Tablosunun Normalizasyon İşlemleri .....	14
2.6. “LaboratuvarTestleri” Tablosunun Normalizasyon İşlemleri .....	16
2.7. “Faturalar” Tablosunun Normalizasyon İşlemleri.....	18
2.8. Tablo İlişkilerinin Tanımlanması .....	20
2.8.1. 1:N ilişkisi Buluan Tablolar.....	20
2.8.2. 1:1 ilişkisi Buluan Tablolar .....	22
2.8.3. M:N ilişkisi Bulunan Tablolar .....	22
3. Varlık-İlişki (E-R) Modeli.....	23
4. İlişkisel Cebir Örnekleri .....	24
5. Tabloların Veri Tabanında Oluşturulması.....	25
5.1. “Hasta” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	30
5.2. “AcilDurumKisileri” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	32
5.3. “KronikHastaliklar” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	33
5.4. “HastaninKronikHastaliklar” Tablosunun Veri Tabanı Üzerinde Oluşturulması.....	33
5.5. “Sigorta” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	34
5.6. “Doktor” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	35
5.7. “DoktorCalismaProgrami” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	36
5.8. “Randevular” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	37
5.9. “MuayeneKayitlari” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	38
5.10. “Receteler” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	39
5.11. “Faturalar” Tablosunun Veri Tabanı Üzerinde Oluşturulması.....	41
5.12. “Odemeler” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	42
5.13. “LaboratuvarTestleri” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	43
5.14. “TestTuru” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	44
5.15. “TestDetay” Tablosunun Veri Tabanı Üzerinde Oluşturulması .....	46

6. Tablolara Veri Eklenmesi.....	48
7. Veri Tabanı Sorgu Örnekleri .....	57
7.1. Basit Sorgu Örnekleri.....	57
7.2. Saklı Yordamlar (Stored Procedures) Örnekleri .....	60
7.3. Trigger (Tetikleyici) Örneği.....	63

## 1. Sistem Gereksinim Analizi ve Kavramsal Modelin Oluřturulması

Bu sistem özel bir saėlık kliniėinde hasta takibi, doktor planlaması, randevu yönetimi, tıbbi kayıtların saklanması, faturalandırma ve envanter kontrolü gibi tüm süreçlerin veri tabanında güvenli ve verimli şekilde yönetilmesi amacıyla tasarlanmıştır. Bu otomasyon sistemi hastaların kayıt işlemlerinden muayene sonrası takiplerine kadar geçen tüm işlemleri bulundurur ve hem saėlık çalışanlarının iş yükünü azaltmayı hem de hasta memnuniyetini artırmayı hedefler.

Sistem içerisinde geliştirilecek projeye uygun olan 6 tablo bulunmaktadır. Bu tabloların içerisinde bulunan temel varlıklar ve varlıkların içeriėi ařaėıdaki gibidir:

- Kliniėe kayıtlı tüm hastalara ait “Hasta” tablosunda hastaların kişisel bilgileri, iletişim bilgileri, tıbbi geçmişleri, randevuları, muayene kayıtları, reçeteleri ve ödeme bilgileri gibi detayları bulunur.
- Kliniėe kayıtlı tüm doktorlar için “Doktor” tablosunda doktorların kimlik bilgileri, uzmanlık alanları, çalışma saatleri ve hasta ilişkili tıbbi kayıtları (muayene ve reçeteler) yer alır.
- Kliniėe ait her randevu için “Randevular” tablosu, hastaların doktorlarla gerçekleştirecekleri görüşmelerin tarih, saat, tür ve durum gibi bilgilerini içerir.
- Kliniėe ait geçmişteki ve yeni tüm muayene kayıtlarını tutan “MuayeneKayıtlari” tablosunda her muayene sonucunda oluşturulan tanı, tedavi planı ve ilgili reçetelerin bilgilerinin tutulduėu kayıtlardır.
- Özel bir klinik olmasından dolayı muayene sonucu bir ücretlendirme olur. Bu ücretlendirmeler “Faturalar” tablosunda saėlanan saėlık hizmetlerinin ücretlendirilmesi, ödeme yöntemi ve ödeme durumu gibi finansal işlemler saklanacaktır.
- Doktor tarafından istenilen testleri yaptırmış tüm hastalara ait laboratuvar testlerinin ve sonuçlarının bulunduėu “LaboratuvarTestleri” tablosu da Kan testleri, röntgenler veya MR gibi tetkiklerin türü, sonuçları ve teslim tarihlerini kapsar.

Özet olarak kullanılacak tablolar ve varlıkları normalizasyon öncesi aşağıdaki gibidir.

**Hasta**(HastaID, Ad, Soyad, TCKimlikNo, DogumTarihi, Cinsiyet, Telefon, Eposta, Adres, AcilAd, AcilSoyad, AcilEposta, AcilTel, SigortaSirketAdi, PoliçeNo, SigortaBaslangic, SigortaBitis, SigortaTuru, HastalikGecmis )

**Doktor**(DoktorID, Ad, Soyad, TCKimlikNo, UzmanlikAlani, Telefon, Eposta, CalismaGun, CalismaBaslangicSaat, CalismaBitisSaat)

**Randevular**(RandevuID, HastaID, HastaAd, HastaSoyad, DoktorID, DoktorAd, DoktorSoyad, RandevuTarihi, RandevuSaati, RandevuTuru, RandevuDurumu)

**MuayeneKayitlari**(MuayeneID, HastaID, DoktorID, HastaAd, HastaSoyad, DoktorAd, DoktorSoyad, MuayeneTarihi, ReceteTarihi, Tani, Tedavi, IlacAd, Dozaj)

**LaboratuvarTestleri**(TestID, HastaID, HastaAd, HastaSoyad, DoktorID, DoktorAd, DoktordSoyad, TestTuru, TestAd, NumuneAlimTarihi, TestSonuc, SonucTeslimTarihi)

**Faturalar**(FaturaID, HastaID, DoktorID, OdemeYontem, OdemeDurum, Ucret, HizmetTuru, FaturaTarihi)

Bu normalizasyon öncesi başlangıç durumundaki tablolar ile birlikte hem hastaların hem de kliniğin yönetiminde kritik rol oynayacak güvenilir, hızlı ve stabil bir sistem geliştirilecektir.

## 2. Tabloların Normalizasyon İşlemleri

Bu bölümde sistem gereksinim analizi bölümünde kararlaştırılmış olan projeye uygun tabloların **1NF (Birinci Normal Form)**, **2NF (İkinci Normal Form)** ve **3NF (Üçüncü Normal Form)** normalizasyon işlem basamaklarına tabi tutulacaktır. Bilgilendirme amaçlı bu normalizasyon işlemlerinin tablolara ne gibi etkisinin olacağı ve ne zaman yapılması gerektiği açıklanacaktır.

### 1NF (Birinci Normal Form):

Bir tablo, 1NF’de olabilmek için her hücreinde yalnızca tek bir değer bulunmalıdır yani tekrarlayan ya da çoklu değerler olmamalıdır. Bu yöntemin tablolara etkisi:

- Çok değerli alanlar ayrı satırlara veya alt tablolara bölünür.
- Veri tekrarları azaltılır
- Sorgulama kolaylaşır.

Bu yöntem verilerde bir hücrede birden fazla bilgi varsa, 1NF ihlali vardır. Bu durumun olduğu yereler düzeltilmelidir.

### 2NF (İkinci Normal Form):

Bir tablo 1NF’teyse ve herhangi bir bileşik anahtarın yalnızca bir kısmına bağımlı olan sütunlar yoksa bu tablo 2NF’tedir. Yani tüm bağımlılıklar tam anahtara bağlı olmalıdır. Bu yöntemin tablolara etkisi :

- Bileşik anahtarlardan kısmi bağımlı sütunlar ayrılır.
- Örneğin “Doktor” tablosunda sadece “DoktorID” yerine hem HastaID hem DoktorID varsa ayrılır ve ayrı bir tabloya konur.

Bu yöntem bir tabloda bileşik anahtar varsa ve bazı sütunlar bu anahtarın sadece bir kısmına bağlıysa, 2NF uygulanmalıdır.

### 3NF (Üçüncü Normal Form)

2NF sağlandıktan sonra tabloda anahtar olmayan sütunlar birbirine bağımlı olmamalıdır. Daha basit bir şekilde açıklanacak olursa her sütun sadece doğrudan anahtara bağlı olmalıdır. Bu yöntemin tablolara etkisi :

- Örneğin: “HastaID → SigortaID → SigortaAdi” gibi dolaylı bağlar varsa, "Sigorta" bilgisi ayrı bir tabloya alınır.
- Gereksiz veri tekrarları ortadan kaldırılır, veri tutarlılığı artar.

Tablolarda anahtar olmayan sütunların birbirine bağımlı olduğu fark edilirse 3NF’e getirilip tablo düzeltilmelidir.

#### 2.1. “Hasta” Tablosunun Normalizasyon İşlemleri

"Hasta" tablosu sistem tasarımında bahsedilen halinde birçok bağımlı ve tekrarlı bilgiyi aynı yerde bulunduruyor. Oluşturulacak veri tabanının sorunlara yol açıp ve verimsiz olmaması için “Hasta” tablosu üzerinde birinci (1NF), ikinci (2NF) ve üçüncü normal form (3NF) kuralları adım adım uygulanacaktır.

Hasta	
PK	<u>HastaID</u>
	Ad
	Soyad
	TCKimlikNo
	DogumTarihi
	Cinsiyet
	Telefon
	Eposta
	Adres
	AcilAd
	AcilSoyad
	AcilEposta
	AcilTel
	SigortaSirketAdi
	PoliceNo
	SigortaBaslangic
	SigortaBitis
	SigortaTuru
	HastalikGecmis

**Tablo 1 :** Normalizasyon Öncesi “Hasta” Tablosu

1NF sonucunda tekrarlanabilir gruplar olan AcilAd, AcilSoyad, AcilEposta, AcilTel kaldırılıp atomik olmayan HastalikGecmis düzenlenip sonucunda Hasta, AcilDurumKisileri, KronikHastaliklar, HastaninKronikHastaliklari adında 4 farklı tablo oluşmuştur. Sigorta bilgileri ayrı bir tabloya henüz ayrılmadı ama potansiyel tekrar içeriyor.

Hasta	
PK	<u>HastaID</u>
	Ad
	Soyad
	TCKimlikNo
	DogumTarihi
	Cinsiyet
	Telefon
	Eposta
	Adres

**Tablo 2 :** 1NF Normalizasyonu Sonrası “Hastalar” Tablosu

AcilDurumKisileri	
PK	<u>AcilID</u>
FK	HastaID
	AcilAd
	AcilSoyad
	AcilEposta
	Telefon

**Tablo 3 :** 1NF Sonucunda Oluşan “AcilDurumKisileri” Tablosu

KronikHastaliklar	
PK	<u>HastalikID</u>
	HastalikAdi

**Tablo 4 :** 1NF Sonucunda Oluşan “KronikHastaliklar” Tablosu



HastanınKronikHastaliklari	
PK	<u>HastaID</u>
FK	HastalikID

**Tablo 5 :** 1NF Sonucunda Oluşan “HastanınKronikHastaliklari” Tablosu

Hasta tablosunda bulunan sigorta bilgilerinde (SigortaSirketAdi, PoliceNo) tekrar eden bağımlı gruplar var. Birden çok hasta aynı sigorta şirketiyle çalışabileceği için sigorta bilgileri tekrar etmeye başlar bu da 2NF normalizasyonunu yapılmasını zorunlu kılar. SigortaID referansı “Hasta” tablosunda bırakılıp tekrar edecek olan satırlar “Sigorta” tablosuna atanır.

Hasta	
PK	<u>HastaID</u>
FK	SigortaID
	Ad
	Soyad
	TCKimlikNo
	DogumTarihi
	Cinsiyet
	Telefon
	Eposta
	Adres

AcilDurumKisileri	
PK	<u>AcilID</u>
FK	HastaID
	AcilAd
	AcilSoyad
	AcilEposta
	Telefon

KronikHastaliklar	
PK	<u>HastalikID</u>
	HastalikAdi

Sigorta	
PK	<u>SigortaID</u>
	SirketAdi
	PoliceNo
	SigortaBaslangic
	SigortaBitis
	SigortaTuru

HastanınKronikHastaliklari	
PK	<u>HastaID</u>
FK	HastalikID

**Tablo 6 :** 2NF Sonucunda Oluşan “Sigorta” ve Güncellenmiş “Hasta” Tablosu

Oluşan tablolar 3NF normalizasyonunu gerektirmez. O yüzden bu normalizasyon işlemi atlanır.

## 2.2. “Doktor” Tablosunun Normalizasyon İşlemleri

“Doktor” tablosu ilk halinde doktorların kişisel ve mesleki bilgilerini tek bir yapıda tutmakta ayrıca tekrarlı ve birden fazla değeri içeren bazı alanları da aynı tabloda barındırmaktadır. Özellikle çalışma programı gibi tekrarlı veri içeren yapılar tabloyu hem yönetilmesi zor hale getiriyor hem de veri tekrarına ve tutarsızlığına yol açıyor. Bu nedenle “Doktor” tablosu üzerinde birinci (1NF), ikinci (2NF) ve üçüncü normal form (3NF) kuralları uygulanarak yapı daha verimli hale getirilecektir.

Doktor	
PK	<u>DoktorID</u>
	Ad
	Soyad
	TCKimlikNo
	UzmanlikAlani
	Telefon
	Eposta
	CalismaProgrami

**Tablo 7 :** Normalize Öncesi “Doktor” Tablosu

Bu yapıda CalismaProgrami bilgileri çoğul olarak tutulduğu için 1NF'e aykırıdır. Çalışma günleri arttıkça tablo genişleyecek ve veri tekrarı oluşacaktır. Bu bozukluklara 1NF uygulandıktan sonra çoklu değer içeren sütunlar ayrıştırılmıştır. Tekrar eden sütunlar bağımsız bir tabloya dönüştürülmüştür. Sonuç olarak “Doktor” ve “DoktorCalismaProgrami” adlı iki tablo oluşturulmuştur.

Doktor	
PK	<u>DoktorID</u>
	Ad
	Soyad
	TCKimlikNo
	UzmanlikAlani
	Telefon
	Eposta

**Tablo 8 :**1NF Normalizasyonu Sonrası “Doktor” Tablosu

DoktorCalismaProgrami	
PK	<u>ProgramID</u>
FK	DoktorID
	Gun
	BaslangicSaati
	BitisSaati

**Tablo 9 :** “DoktorCalismaprogrami” Tablosu

Her sütun bu anahtara tam bağımlı olduğu ve herhangi bir sütunun başka bir sütun aracılığıyla anahtara bağımlı olması durumu bulunmadığı için 3NF ve 2NF normalizasyonlarının uygulanmasına gerek görülmemiştir. “Doktor” tablosunun yeni hali yukarıda gösterildiği gibidir.

### 2.3. “Randevular” Tablosunun Normalizasyon İşlemleri

Bu tablo hasta ve doktor bilgileriyle alakalı birçok bilgiyi bir arada bulundurmaktadır. Bu durum veri tekrarına yol açabilme şansı ve ileride randevulara ek bilgiler eklendiğinde tablo karmaşıklıklaştırma şansı yüksektir. Bu nedenle 1NF, 2NF ve 3NF kuralları uygulanarak “Randevular” tablosu normalleştirilecektir.

Randevular	
PK	<u>RandevuID</u>
FK	HastaID
	HastaAd
	HastaSoyad
FK	DoktorID
	DoktorAd
	RandevuTarihi
	RandevuSaati
	RandevuTuru
	RandevuDurumu

**Tablo 10 :** Normalizasyon Öncesi “Randevular” Tablosu

Bu tabloda çoklu değer içeren veya tekrar eden sütunlar olmadığından yapısal olarak 1NF’ye uygundur. RandevuID birincil anahtar olduğundan tüm alanlar ona tam bağımlıdır, kısmi bağımlılık yoktur yani 2NF’e de uygundur. Hasta ve doktor isimleri doğrudan RandevuID yerine sırasıyla HastaID ve DoktorID alanlarına bağımlı olmalıdır. Bu alanlar “Randevu” tablosunda tutulduğunda transitif bağımlılık oluşur. Bu durum 3NF’i ihlal eder. Bu nedenle bu alanlar kaldırılarak yalnızca HastaID ve DoktorID gibi yabancı anahtarlar korunur.

Randevular	
PK	<u>RandevuID</u>
FK	HastaID
FK	DoktorID
	RandevuTarihi
	RandevuSaati
	RandevuTuru
	RandevuDurumu

**Tablo 11 :** Normalizasyon Öncesi “Randevular” Tablosu

Normalizasyon öncesi “Randevular” tablosunda HastaID ve DoktorID, ait oldukları “Hasta” ve “Doktor” tabloları ile ilişkilendirilmemişti; bu nedenle bu alanlar yabancı anahtar (FK) olarak tanımlı değildi. Normalizasyonla bu ilişkiler kurulmuş ve veri bütünlüğü sağlanmış oldu.

## 2.4. “MuayeneKayitlari” Tablosunun Normalizasyon İşlemleri

Bu tablo hasta ve doktor bilgileriyle birlikte tanı, tedavi ve reçete bilgilerini aynı yerde tutmaktadır. Bu durum veri tekrarına, bağımlılıklara ve tutarsızlıklara yol açar. Bu nedenle aşağıda 1NF, 2NF ve 3NF işlemleri adım adım uygulanmış ve oluşan tablolar belirtilmiştir.

MuayeneKayitlari	
PK	<u>MuayeneID</u>
FK	HastaID
FK	DoktorID
	HastaAd
	HastaSoyad
	DoktorAd
	DoktorSoyad
	MuayeneTarih
	ReceteTarih
	Tani
	Tedavi
	IlacAd
	Dozaj

**Tablo 12 :** Normalizasyon Öncesi “MuayeneKayitlari” Tablosu

Bu tablonun her hücrelerinde yalnızca tek bir bilgi barındırır ve çoklu ya da tekrarlı değer içermez dolayısıyla 1NF’e aykırılık yoktur.

MuayeneID birincil anahtar olduğundan diğer alanlar bu anahtara tam bağımlıdır. Yani tabloda herhangi bir kısmi bağımlılık bulunması için 2NF’e de uygundur ve doğrudan 3NF’e geçilebilir.

3NF aşamasında, HastaAd, HastaSoyad, DoktorAd ve DoktorSoyad gibi alanlar aslında HastaID ve DoktorID'ye bağımlı olduklarından dolayı transitif bağımlılık vardır. Bu durum 3NF'e aykırıdır. Bu nedenle bu bilgiler tablodan kaldırılır HastaID ve DoktorID ile kladırılan bilgiler tutulur. Ayrıca reçete bilgileri (ReceteTarihi, IlacAd, Dozaj) de kendi tablosuna sahip olması gerektiğinden ayrı bir "Receteler" tablosu oluşturulur. Sonuç olarak veri tekrarı ve anlamsal bağımlılıklar ortadan kaldırılmış olur. "MuayeneKayıtları" tablosunun son hali aşağıdaki tabloda görselleştirilmiştir.

Receteler	
PK	<u>ReceteID</u>
FK	MuayeneID
FK	HastaID
FK	DoktorID
	ReceteTarihi
	IlacAd
	Dozaj

MuayeneKayıtları	
PK	<u>MuayeneID</u>
FK	HastaID
FK	DoktorID
	MuayeneTarihi
	Tani
	Tedavi

**Tablo 13 :** Normalizasyon Sonrası Oluşan "Receteler" ve Düzenlenmiş "MuayeneKayıtları" Tablosu

## 2.6. “LaboratuvarTestleri” Tablosunun Normalizasyon İşlemleri

Başlangıçta çoklu veri içeren TestTuru, TestSonuçları ve tekrarlı nitelik taşıyan alanlar, hasta ve doktor adlarının doğrudan yazılması gibi yapısal sorunlar bulunmaktadır. Bu tablo birinci normal formdan başlanarak üçüncü normal forma kadar adım adım normalize edilerek düzgün hale getirilmiştir.

LaboratuvarTestleri	
PK	<u>TestID</u>
FK	HastaID
FK	DoktorID
	HastaAd
	HastaSoyad
	DoktorAd
	DoktorSoyad
	TestTuru
	TestAd
	NumuneAlimTarihi
	TestSonuc
	SonucTeslimTarihi

**Tablo 14 :** Normalizasyonu Öncesi” LaboratuvarTestleri” Tablosu

Başlangıçta "LaboratuvarTestleri" tablosunda yer alan TestTuru ve TestSonuc elemanları zaman zaman “MR, Kan” gibi birden fazla değeri aynı hücrede barındırıyor. Bu durum her hücrede yalnızca tek bir bilgi bulunmadığı için 1NF’e aykırıdır. Dolayısıyla her “TestTuru” ve “TestSonuc” için ayrı satırlar oluşturularak çoklu değerler ayrılacaktır ayrıca hasta ve doktor bilgileri doğrudan isim olarak tutuluyordu bu da yine veri tekrarı oluşturuyordu. Bu alanlar yalnızca HastaID ve DoktorID şeklinde tutulup ilgili varlık tabloları ile ilişkilendirilerek gereksiz tekrarlar ve transitif bağımlılıklar giderilmiştir.



LaboratuvarTestleri	
PK	<u>TestID</u>
FK	HastaID
FK	DoktorID
	TestTuru
	NumuneAlimTarihi
	TestSonuc
	SonucTeslimTarihi

TestTuru	
PK	<u>TestTuruID</u>
	TestAd

**Tablo 15 :** 1NF Normalizasyonu Sonrası Oluşan “TestTuru” ve Düzenlenmiş “LaboratuvarTestleri” Tablosu

Tablo yapısında TestID birincil anahtar olarak tüm diğer tanımladığı için kısmi bağımlılık yoktur. Dolayısıyla 2NF ile ilgili bir sorun yoktur ama TestTuru ve TestSonucu sütunları arasında doğrudan bir bağımlılık var bu yüzden transitif bağımlılık oluşuyor. Bu sorunu ortadan kaldırmak için test türü ve sonucunu içeren bilgiler “TestDetay” adlı ayrı bir tabloya taşınmalıdır. Böylece bir testin birden fazla test türü ve sonuç bilgisiyle ilişkilendirilebilmesi mümkün hale gelir. Ayrıca TestTuru alanı içerdiği tekrar eden veriler nedeniyle ayrı bir “TestTuru” tablosuna ayrılması gerekmiştir. Bu işlemler sonucunda tekrar eden veriler ortadan kaldırılmış ve sistem tasarımı 3NF’e uygun hale gelmiştir.

LaboratuvarTestleri	
PK	<u>TestID</u>
FK	HastaID
FK	DoktorID
	NumuneAlimTarihi
	SonucTeslimTarihi

TestTuru	
PK	<u>TestTuruID</u>
	TestAd

TestDetay	
PK	<u>TestDetayID</u>
FK	TestTuruID
	TestSonucu

**Tablo 16 :** Normalizasyonu Sonrası Oluşan “TestDetay” ve Düzenlenmiş “LaboratuvarTestleri” Tablosu

## 2.7. “Faturalar” Tablosunun Normalizasyon İşlemleri

Başlangıçta “Faturalar” tablosunda bazı tekrar eden bilgiler doğrudan tutuluyor. Dolayısıyla veri tekrarı tutarsızlıkları oluşuyor. Bu sorunları düzeltmek için tablo 1NF, 2NF ve 3NF kurallarına uygun olarak normalizasyon uygulanacak ve sade tablolar elde edilecektir.

Faturalar	
PK	<u>FaturalID</u>
FK	Hasta ID
FK	DoktorID
	OdemeYontem
	OdemeDurumu
	Ucret
	HizmetTuru
	FaturaTarihi

**Tablo 17 :** Normalizasyonu Öncesi “Faturalar” Tablosu

“OdemeYontem” ve “OdemeDurum” alanları ileride ödeme yöntemleri veya durumlarıyla ilgili yapılacak değişikliklerde zorluklarına neden olması ve dolayısıyla transitif bağımlılık bulundurması bu tabloyu 3NF’e aykırı yapıyordu. Bu sorunu ortadan kaldırmak için “OdemeYontem” ve “OdemeDurum” “Odemeler” tablosuna taşınmıştır. Böylece fatura kayıtlarında yalnızca OdemeID ile tuutlmaya başlnmıştır. Bu normalizasyon ile veri tekrarı önlenmiş, tutarsızlıkların önüne geçilmiş aynı zamanda veri bütünlüğü sağlanmıştır. Sonuç olarak sistem tasarımı 3NF’e uygun hale getirilmiştir.

Faturalar		Odemeler	
PK	<u>FaturalID</u>	PK	<u>OdemeID</u>
FK	Hasta ID		OdemeYontem
FK	DoktorID		OdemeDurum
FK	OdemeID		
	Ucret		
	HizmetTuru		
	FaturaTarihi		

**Tablo 18 :** Normalizasyonu Sonrası Oluşan “Odemeler” ve Düzenlenmiş “Faturalar” Tablosu

Normalizasyon işlemleri uygulandıktan sonra oluşan tabloların son durumu aşağıdaki gibidir. Toplamda 15 tablo oluşmuştur.

**Hasta**(HastaID, SigortaID, AcilID, Ad, Soyad, TcKimlikNo, DogumTarihi, Cinsiyet, Telefon, Eposta, Adres)

**AcilDurumKisileri**(AcilID, HastaID, Telefon, AcilAd, AcilSoyad, AcilEposta)

**KronikHastaliklar**(HastalikID, HastalikAdi)

**HastaninKronikHastaliklar**(HastaID, HastalikID)

**Sigorta**(SigortaID, SirketAdi, PoliceNo, SigortaBaslangic, SigortaBitis, SigortaTuru)

**Doktor**(DoktorID, Ad, Soyad, TcKimlikNo, UzmanlikAlani, Telefon, Eposta)

**DoktorCalismaProgrami**(ProgramID, DoktorID, Gun, BaslangicSaati, BitisSaati)

**Randevular**(RandevuID, HastaID, DoktorID, RandevuTarihi, RandevuSaati, RandevuTuru, RandevuDurumu)

**MuayeneKayitlari**(MuayeneID, HastaID, DoktorID, MuayeneTarihi, Tani, Tedavi)

**Receteler**(ReceteID, MuayeneID, HastaID, DoktorID, ReceteTarihi, IlacAd, Dozaj)

**Faturalar**(FaturaID, HastaID, DoktorID, OdemeID, Ucret, HizmetTuru, FaturaTarihi)

**Odemeler**(OdemeID, OdemeYontem, OdemeDurum)

**LaboratuvarTestleri**(TestID, HastaID, DoktorID, NumuneAlimTarihi, SonucTeslimTarihi)

**TestTuru**(TestTuruID, TestAd)

**TestDetay**(TestDetayID, TestTuruID, TestSonucu)

## 2.8. Tablo İlişkilerinin Tanımlanması

Bu tablolara bir de ek olarak göstermemiz gereken, veri tabanı tasarımını daha anlaşılabilir kılan tablolar arası yapısına ve içerdiği bilgilere göre değişen ilişkiler vardır. Bu ilişkiler şöyledir:

- Bire-Bir(1-1): Tablolar arası ilişki kurulan alanların her iki tabloda da tek olması anlamına gelir.
- Bire-Çok(1-n): Tablodaki 1 değer diğer tabloda birden fazla alana (n) karşılık gelmektedir.
- Çoğa-Çok(n-n): Tabloların birindeki birçok kaydın değeri diğer tablolarda birden fazla kayda karşılık gelir.

### 2.8.1. 1:N İlişkisi Buluan Tablolar

**Hasta - AcilDurumKisileri :**

Bir hastanın birden fazla acil kişisi olabilir.

**Hasta – Randevular :**

Bir hasta birden fazla randevu alabilir ancak her randevu yalnızca bir hastaya aittir.

**Hasta – Receteler**

Bir hasta birçok reçeteye sahip olabilir. Her reçete yalnızca bir hastaya aittir.

**LaboratuvarTestleri – Hasta**

Bir hasta birçok test yaptırabilir.

**Hasta – HastanınKronikHastalıklar :**

Bir hasta birden fazla kronik hastalığa sahip olabilir. Her hasta hastalığı ilişkisi yalnızca bir hastaya aittir.

**HastanınKronikHastalıklar – KronikHastalıklar:**

Bir hastalık birçok hastada bulunabilir.

**Doktor – DoktorÇalışmaProgramı :**

Bir doktor birden fazla çalışma günü ve saatine sahip olabilir.

**Doktor – Randevular :**

Bir doktor birçok hastayla randevu yazabilir ama her randevu tek bir doktora aittir.

**Doktor – Receteler**

Bir doktor birçok reçete yazabilir. Her reçete yalnızca bir doktora aittir.

**LaboratuvarTestleri – Doktor**

Bir doktor birçok laboratuvar testini istemiş olabilir.

**TestTuru – Detay**

Her test türünün (Kan Testi, MR) birçok sonucu olabilir.

### **MuayeneKayitlari – Receteler**

Bir muayeneye birden fazla reçete yazılabilir. Her reçete yalnızca bir muayeneye aittir.

### **MuayeneKayitlari – Hasta :**

Bir hasta birçok muayene kaydına sahip olabilir. Her muayene yalnızca bir hastaya aittir.

### **MuayeneKayitlari – Doktor :**

Bir doktor birçok muayene gerçekleştirmiş olabilir. Her muayene yalnızca bir doktora aittir.

### **Faturalar – Hasta :**

Bir hasta birçok faturaya sahip olabilir. Her fatura yalnızca bir hastaya aittir.

### **Faturalar – Odemeler :**

Bir ödeme yöntemi birden fazla faturada kullanılabilir. Her fatura yalnızca bir ödeme yöntemiyle ilişkilidir.

## **2.8.2. 1:1 İlişkisi Bulunan Tablolar**

### **Hasta - Sigorta**

Bir hasta bir sigortaya bağlıdır. (1:1)

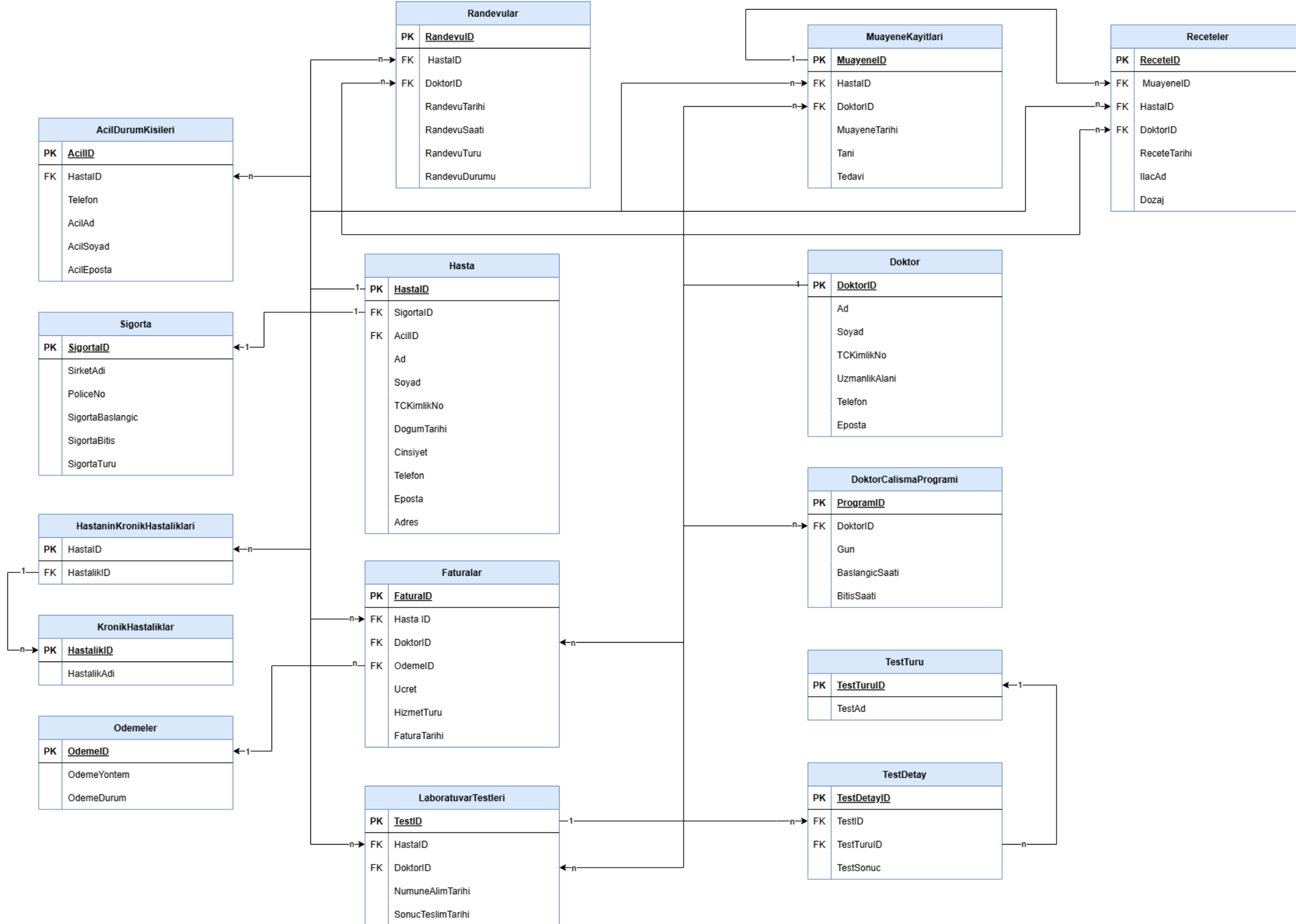
## **2.8.3. M:N İlişkisi Bulunan Tablolar**

### **Hasta-KronikHastaliklar :**

Bir hastanın birden fazla kronik hastalığı olabilir. Bir kronik hastalık birden fazla hastada bulunabilir. (M:N)

### 3. Varlık-İlişki (E-R) Modeli

Normalize edilmiş ve gerekli bağımlılıklar ile ilişkilendirilmiş veri tabanı artık E-R çizimi için uygun hale getirilmiştir. Uygun hale getirilen veri tabanı tablolarının ilişkilerinin bağlantıları aşağıdaki tabloda görselleştirilmiştir.



Tablo 19 : E-R Şeması

#### 4. İlişkisel Cebir Örnekleri

1) HastaID'si 1 olan hastanın sistemde bulunan tüm randevularını listeleme

$\pi_{\text{RandevuID}, \text{RandevuTarihi}, \text{RandevuSaati}, \text{RandevuTuru}, \text{RandevuDurumu}}(\sigma_{\text{HastaID}=1}(\text{Randevular}))$

2) 2025 yılının 5. ayının 1. günü sigorta süresi dolmuş hastaların ad ve soyad bilgilerini listeleme

$\pi_{\text{Hasta.Ad}, \text{Hasta.Soyad}}((\text{Hasta} \bowtie \text{Sigorta}) \sigma_{\text{SigortaBitis}='2025-05-01'})$

3) Randevu alıp gelmeyen hastaların e-posta ve telefon bilgileri

$\pi_{\text{Hasta.Eposta}, \text{Hasta.Telefon}}((\sigma_{\text{RandevuDurumu}='Gelmedi'}(\text{Randevular})) \bowtie \text{Hasta})$

4) Randevu alıp gelmeyen hastaların e-posta ve telefon bilgileri

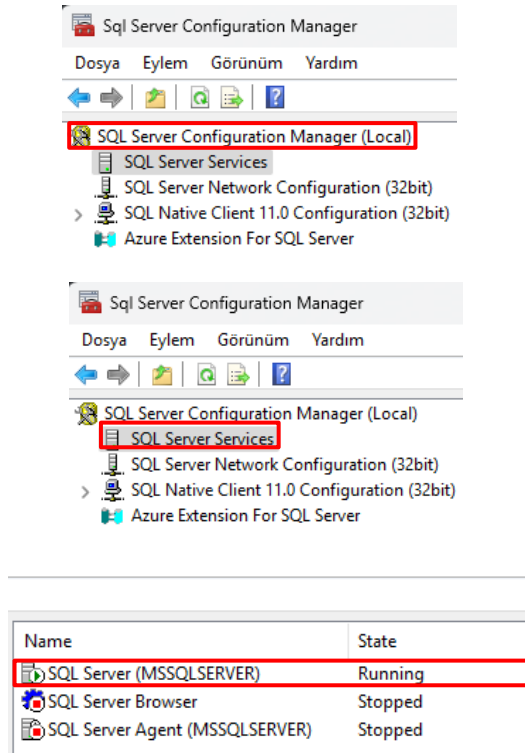
$\pi_{\text{Hasta.Ad}, \text{Hasta.Soyad}, \text{MuayeneKayitlari.MuayeneTarihi}}((\text{Hasta} \bowtie \text{MuayeneKayitlari}) \bowtie (\sigma_{\text{IlacAd}='Parol'}(\text{Receteler})))$



## 5. Tabloların Veri Tabanında Oluşturulması

Bu bölümde yapılacak işlemlere öncesinde Microsoft SQL Server veritabanının kurulu olması gerekmektedir. “SQL Server Management Studio (SSMS)” uygulamasının yüklü ve çalışır durumda olması önemlidir. Kurulum sonrası “SQL Server 2022 Configuration Manager” üzerinden “SQL Server Services” altında yer alan “SQL Server (MSSQLSERVER)” hizmetinin "Running" durumda olması kontrol edilmelidir.

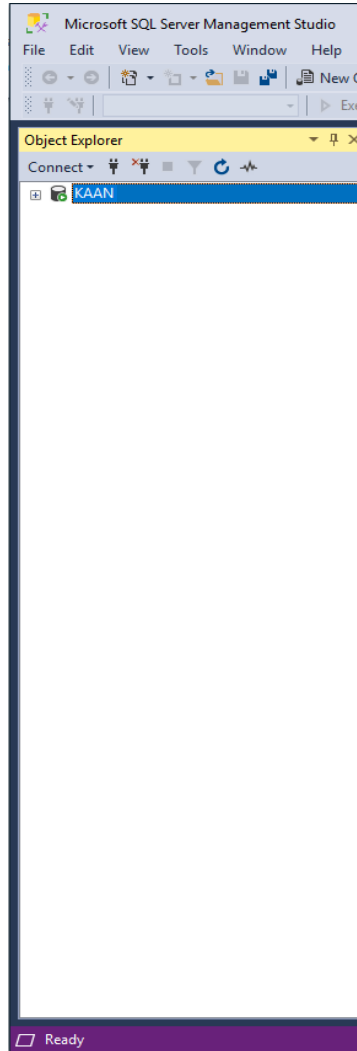
SQL Server’ın kurulumu veya sorunla karşılaşılması durumunda YouTube gibi video platformlarından veya ilgili teknik internet sitelerinden yardım alabilirsiniz. Bu projede veri tabanı tasarımı için SQL Server 2022 sürümü kullanılacaktır. Ancak bu sürümle uyumlu diğer MSSQL versiyonları da tercih edilebilir.



**Görsel 1 :** “SQL Server 2022 Configuration Manager” İçerisinde İşlem Öncesi Kontrol Edilmesi Gereken Kısım

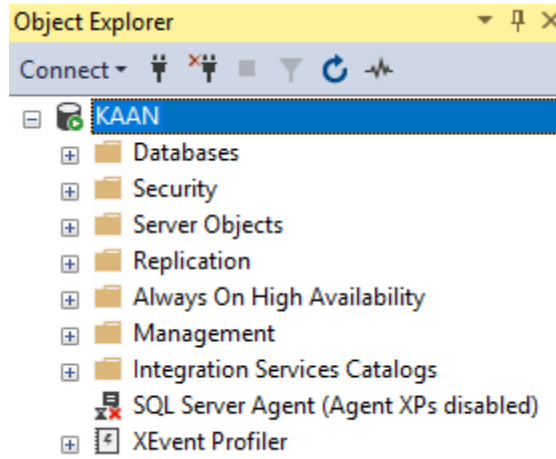
Kurulum tamamlandıktan ve varsa karşılaşılan sorunlar giderildikten sonra kurulum sırasında oluşturulan dosya içerisinde "SQL Server Management Studio 20 (SSMS)" çalıştırılır. Açılan giriş ekranında kurulum sırasında belirlediğimiz sunucu için kullanıcı adı ve şifre bilgileri girilip "Connect" butonuna tıklanarak SQL Server 2022 ana ekranına erişilir.

Ekranın sol üst kısmında yer alan "Object Explorer" yan paneli üzerinden oluşturulan sunucu (server) bulunur. Projeye ilgili tüm veri tabanı işlemleri bu sunucu altında gerçekleştirilecektir.



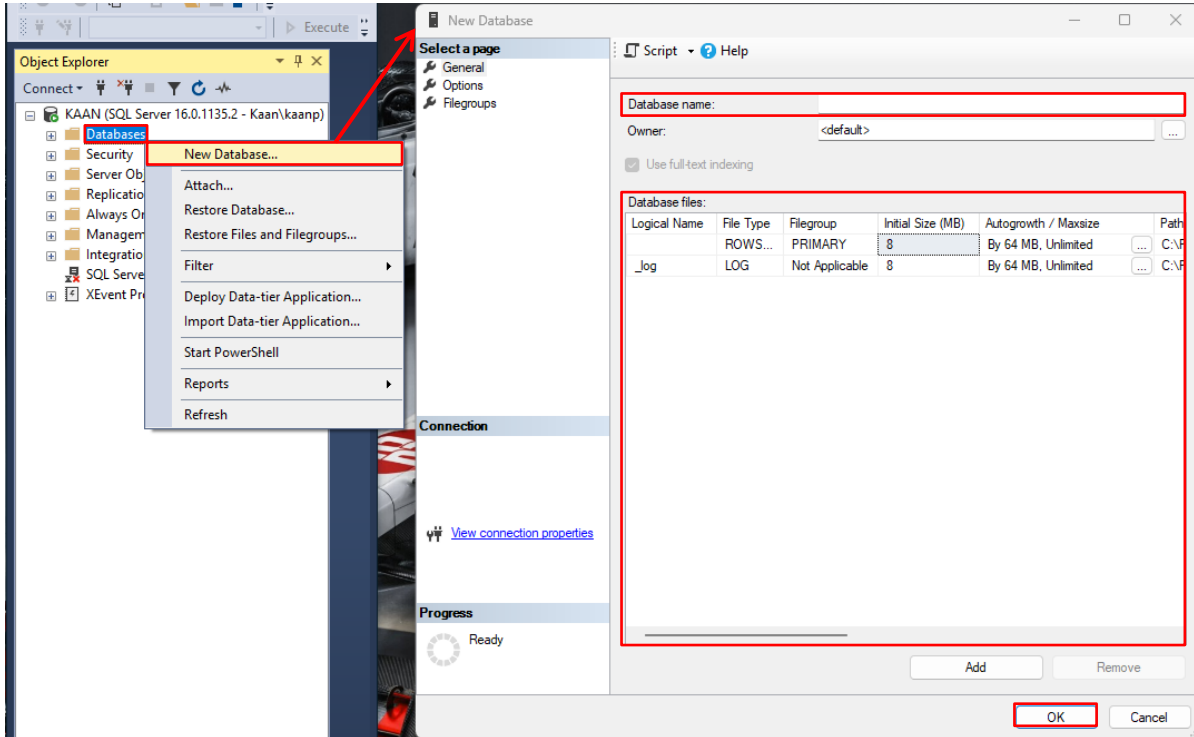
**Görsel 2 :** "SQL Server Management Studio 20"de Bulunan "Object Explorer"

Bu ağacın solunda bulunan "+" simgesine tıklayarak bütün düzenlenebilecek alanlar görülebilir.



Görsel 3 : Sunucu İçinde Bulunan Düzenlenebilecek Alanlar

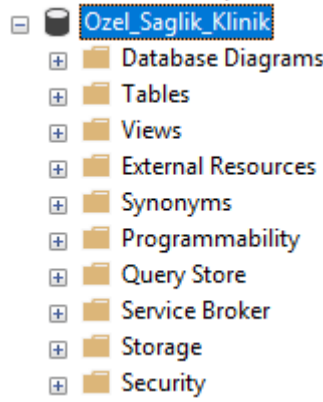
Bir veri tabanı oluşturacağımız için “Object Explorer” panelinde yer alan "Databases" alanına sağ tıklayıp açılan menüden "New Database..." seçeneği seçilir. Bu işlem, yeni bir veri tabanı oluşturma ekranını açar.



Görsel 4 : Yeni Veri Tabanı Oluşturma

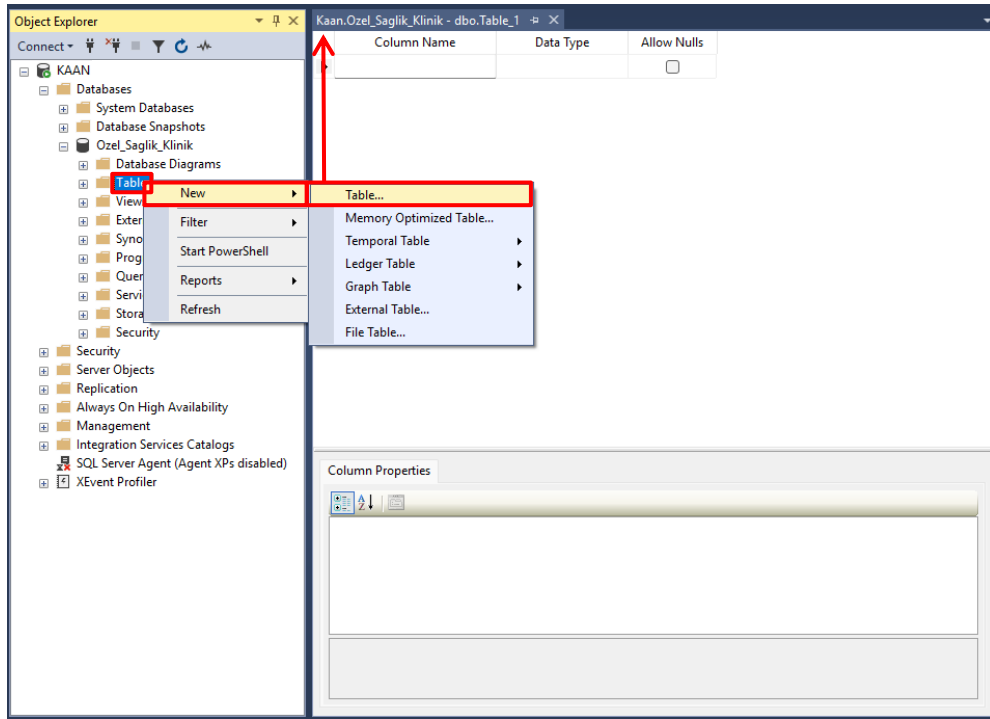
Karşımıza çıkan bu ekranda "Database Name" alanına oluşturmak istediğimiz veri tabanının adı girilir. Alt kısımda yer alan "Database Files" bölümünde ise veri tabanına ait "ROWS Data" ve "LOG" dosyalarının nerede saklanacağını seçebileceğimiz "PATH" alanları bulunur. Bu alanları konumuz olan “Özel Sağlık Kliniği”ne uygun olarak oluşturduktan sonra pencerenin aşağısında bulunan “OK” butonu ile veri tabanımızı oluşturmuş oluruz.

Yine “Object Explorer” penceresinin içinde oluşturulan veri tabanımız bulunur. Ayrıca Microsoft tarafından oluşturulan veri tabanımız içerisinde ekleme, silme, düzenleme yapabileceğimiz birçok alan da alt başlık olarak görülebilir.

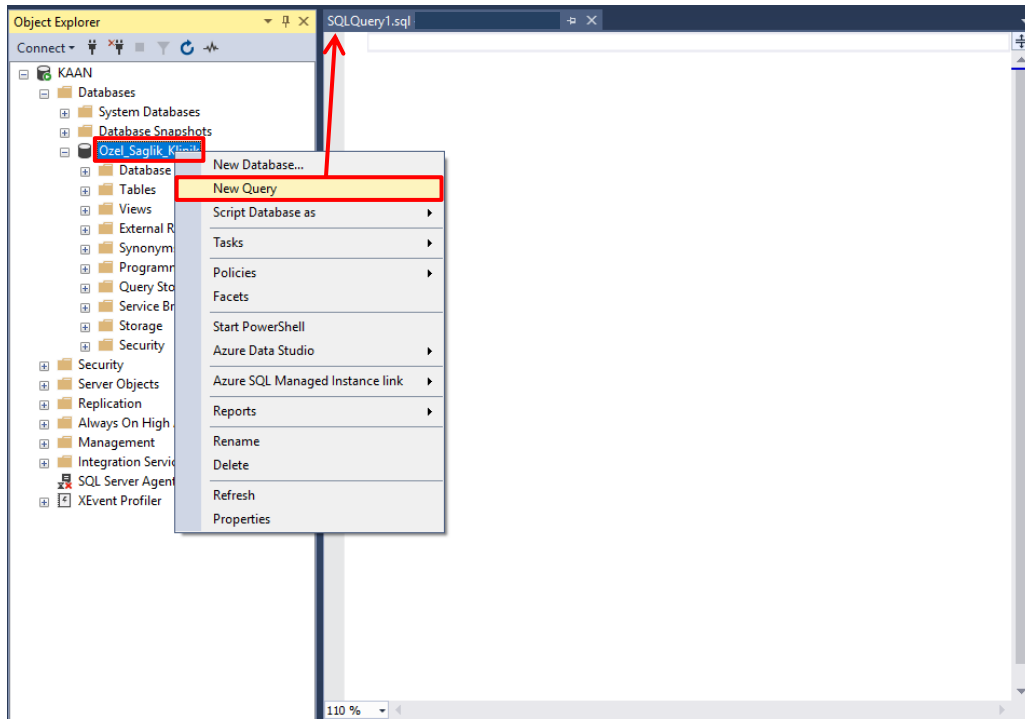


**Görsel 5 :** “Ozel\_Saglik\_Klinik” İsimli Veri Tabanı ve Alt Başlıkları

Tüm bu işlemlerin ardından, normalizasyon adımları uygulanarak oluşturulan tabloların son halleri sisteme eklenebilir. Veri tabanında tablo oluşturmak için iki farklı yöntem kullanılabilir. Birinci yöntemde "Object Explorer" panelinde yer alan "Tables" bölümüne sağ tıklanarak "New → Table" yolu izlenir ve MsSQL arayüzü üzerinden sütun adları, veri türleri gibi özellikler manuel olarak girilir. Arka planda bu bilgilere göre SQL komutları MsSQL otomatik olarak oluşturup çalıştırır. İkinci yöntem ise kendi veri tabanımızın adına sağ tıklayarak "New Query" seçeneği seçilerek açılan sorgu penceresinde tablolar elle SQL komutlarıyla oluşturulur. Bu yöntem ilk yöntemle göre kullanıcıya daha geniş bir kontrol alanı sağlar. Bu yüzden bu raporda ikinci yöntem kullanılarak tablolar oluşturulacaktır.



**Görsel 6 : Birinci Yöntem ile Tablo Oluşturma**



**Görsel 7 : İkinci Yöntem İçin Tablo Oluşturma Arayüzünü Açma**

Artık sorgu ile tablo oluşturma arayüzünü de açtığımıza göre artık SQL sorgularıyla birlikte tablolarımızı oluşturabilir ve içerisine istenilen değerleri atayabiliriz.

## 5.1. “Hasta” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Bu bölümde, yapısında HastaID, SigortaID, AcilD, Ad, Soyad, TcKimlikNo, DogumTarihi, Cinsiyet, Telefon, Eposta ve Adres alanlarını içeren tablo oluşturulacak ve veri eklenecektir. Bu tabloda diğer bazı tablolara ait yabancı anahtarlar (foreign key) bulunduğu için öncelikle bu anahtarların referans verdiği tabloların öncelik sırasına göre veya aynı anda oluşturulması gerekmektedir. Örneğin SigortaID alanı için önce “Sigorta” tablosu, AcilD alanı için ise önce “AcilDurumKisileri” tablosu oluşturulmalıdır.

- **HastaID:** Birincil anahtardır. Her hasta için benzersiz bir kimlik numarasıdır. CHAR(10) türünde veri içerir.
- **SigortaID :** 'Sigorta' tablosundaki 'SigortaID' sütununa yabancı anahtar olarak referans verir. Hasta ile sigorta bilgisi arasında bağlantı kurulmuş olur. CHAR(10) tipindedir.
- **AcilD :** Acil durum kişilerine ait referans ID'yi belirtir. Sayısal (INT) veri içerir ve boş geçilemez.
- **Ad :** Hastanın ad bilgisini içerir. NVARCHAR(50) türünde tanımlanmıştır. Maksimum 50 karakter içerir.
- **Soyad :** Hastanın soyad bilgisini içerir. NVARCHAR(50) türünde olup maksimum 50 karakter uzunluğundadır.
- **TcKimlikNo :** Hastaya ait Türkiye Cumhuriyeti kimlik numarasını içerir. CHAR(11) veri türündedir ve her hasta için benzersizdir. Dolayısıyla UNIQUE kısıtlaması uygulanmıştır.
- **DogumTarihi :** Hastanın doğum tarihini içerir. DATE veri türündedir. 'YYYY-MM-DD' formatında veri girişi beklenir.
- **Cinsiyet :** Yalnızca 'Erkek' ve 'Kadın' değerlerini kabul eder. Varsayılan değeri '-' olup CHECK kısıtlaması ile kontrol edilir. NVARCHAR(5) veri tipindedir.

- **Telefon** : Hastanın iletişim numarasını içerir. NVARCHAR(10) tipindedir. Başında 0 olmadan bitişik şekilde ve tam 10 karakter uzunluğunda olmalıdır.
- **Eposta** : Hastanın e-posta adresini içerir. NVARCHAR(50) türündedir.
- **Adres** : Hastanın açık adresini belirtir. NVARCHAR(255) veri türündedir ve maksimum 255 karakter barındırabilir.

```
CREATE TABLE Hasta (  
    HastaID INT PRIMARY KEY,  
    SigortaID INT NOT NULL FOREIGN KEY (SigortaID) REFERENCES Sigorta(SigortaID),  
    AcilD INT NOT NULL,  
    Ad NVARCHAR(50) NOT NULL,  
    Soyad NVARCHAR(50) NOT NULL,  
    TcKimlikNo CHAR(11) UNIQUE NOT NULL,  
    DogumTarihi DATE NOT NULL,  
    Cinsiyet NVARCHAR(5) NOT NULL DEFAULT '-', CHECK (Cinsiyet IN ('Erkek' , 'Kadın')),  
    Telefon NVARCHAR(10) NOT NULL,  
    Eposta NVARCHAR(50) NOT NULL,  
    Adres NVARCHAR(255) NOT NULL,  
);
```

**Görsel 8 :** “Hasta” Tablosunu Oluşturan SQL Sorgusu

## 5.2. “AcilDurumKisileri” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Her hastanın acil durumda ulaşılabilecek kişilerine ait bilgileri saklamak için kullanılır. Her hasta birden fazla acil durum kişisi tanımlayabilir. Bu yüzden **HastaID** sütunu **Hasta** tablosuna yabancı anahtar olarak tanımlanır.

- **AcilID:** Birincil anahtar (PRIMARY KEY) olarak tanımlanır. Her acil durum kişisi için benzersiz bir kimliktir. CHAR türündedir.
- **HastaID:** Hangi hastaya ait olduğunu belirtir. FOREIGN KEY olarak Hasta(HastaID) sütununa referans verir. CHAR türündedir.
- **Telefon:** Acil durum kişinin telefon numarası. NVARCHAR türünde ve tam 10 karakter uzunluğunda olmalıdır.
- **AcilAd:** Acil durumda ulaşılabilecek kişinin adı. NVARCHAR türündedir.
- **AcilSoyad:** Acil durumda ulaşılabilecek kişinin soyadı. NVARCHAR türündedir.
- **AcilEposta:** Acil durumda ulaşılabilecek kişinin e-posta adresi. NVARCHAR türündedir.

```
CREATE TABLE AcilDurumKisileri (  
    AcilID CHAR(10) PRIMARY KEY,  
    HastaID CHAR(10) NOT NULL FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),  
    Telefon NVARCHAR(10) NOT NULL,  
    AcilAd NVARCHAR(50) NOT NULL,  
    AcilSoyad NVARCHAR(50) NOT NULL,  
    AcilEposta NVARCHAR(50) NOT NULL,  
);
```

Görsel 9 : “AcilDurumKisileri” Tablosunu Oluşturan SQL Sorgusu



### 5.3. “KronikHastaliklar” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Sistemde tanımlı olan kronik hastalıkların listesini içerir. Her hastalık için benzersiz bir ID bulunur.

- **HastalikID:** PRIMARY KEY olarak tanımlanır. Her kronik hastalık için benzersiz bir kimlik numarasıdır. CHAR türündedir.
- **HastalikAdi:** Diyabet, Hipertansiyon gibi kronik olan hastalığın adını belirtir. NVARCHAR türündedir.

```
CREATE TABLE KronikHastaliklar (  
    HastalikID CHAR(10) PRIMARY KEY,  
    HastalikAdi NVARCHAR(30) NOT NULL  
);
```

Görsel 10 : “KronikHastaliklar” Tablosunu Oluşturan SQL Sorgusu

### 5.4. “HastaninKronikHastaliklar” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Hastalarla birlikte sahip oldukları kronik hastalıklar arasındaki çoktan çoğa (m:n) ilişkiyi kurmak için ara tablo görevi görür. Bu nedenle iki yabancı anahtar içerir.

- **HastaID:** FOREIGN KEY olarak Hasta(HastaID) sütununa referans verir. CHAR türündedir.
- **HastalikID:** FOREIGN KEY olarak KronikHastaliklar(HastalikID) sütununa referans verir. CHAR türündedir.

Birleştirilmiş birincil anahtarlar (Composite Primary Key) ile aynı hasta için aynı hastalık birden fazla kez eklenemez.

```
CREATE TABLE HastaninKronikHastaliklar (  
    HastaID CHAR(10) NOT NULL FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),  
    HastalikID CHAR(10) NOT NULL FOREIGN KEY (HastalikID) REFERENCES KronikHastaliklar(HastalikID),  
    PRIMARY KEY (HastaID, HastalikID)  
);
```

Görsel 11 : “HastaninKronikHastaliklar” Tablosunu Oluşturan SQL Sorgusu

## 5.5. “Sigorta” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Hastaların bağlı olduğu sigorta bilgilerini saklamak için kullanılır. Her kayıt sigorta şirketine ait poliçe bilgisini temsil eder.

- **SigortaID:** Her sigorta kaydı için benzersiz bir kimliktir. PRIMARY KEY olarak tanımlanır. CHAR türündedir.
- **SirketAdi:** Sigorta şirketinin adını belirtir. NVARCHAR türündedir.
- **PoliceNo:** Poliçe numarası. NVARCHAR türündedir. Her poliçe numarası birbirinde farklı olduğu için benzersiz (UNIQUE) olarak tanımlanır.
- **SigortaBaslangic:** Sigortanın başlangıç tarihidir. DATE türündedir.
- **SigortaBitis:** Sigortanın bitiş tarihidir. DATE türündedir.
- **SigortaTuru:** Özel, Devlet gibi sigortanın türünü belirtir. NVARCHAR(50) türündedir.

```
CREATE TABLE Sigorta (  
    SigortaID CHAR(10) PRIMARY KEY,  
    SirketAdi NVARCHAR(100) NOT NULL,  
    PoliceNo NVARCHAR(50) UNIQUE NOT NULL,  
    SigortaBaslangic DATE NOT NULL,  
    SigortaBitis DATE NOT NULL,  
    SigortaTuru NVARCHAR(50) NOT NULL  
);
```

Görsel 12 : “Sigorts” Tablosunu Oluşturan SQL Sorgusu

## 5.6. “Doktor” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Bu bölümde “Doktor” tablosunda bulunan DoktorID, Ad, Soyad, TcKimlikNo, UzmanlikAlani, Telefon ve Eposta alanları oluşturulacaktır. Bu tabloda sağlık bilgi sisteminde yer alan doktorların kimlik, iletişim ve mesleki bilgilerini tutulur.

- **DoktorID:** Birincil anahtardır. Her doktor için benzersiz bir kimlik numarasıdır. CHAR türünde veri içerir.
- **Ad:** Doktorun ad bilgisini içerir. NVARCHAR veri türündedir ve maksimum 50 karakter uzunluğunda olabilir.
- **Soyad:** Doktorun soyad bilgisini içerir. NVARCHAR veri türündedir ve maksimum 50 karakter içerir.
- **TcKimlikNo:** Doktora ait Türkiye Cumhuriyeti kimlik numarasını içerir. CHAR veri türündedir. Her doktor için benzersiz olması gerektiği için UNIQUE kısıtlaması uygulanmalıdır.
- **UzmanlikAlani:** Doktorun uzmanlık alanını belirtir. Örneğin, "Kardiyoloji", "Ortopedi" gibi değerler içerir. NVARCHAR veri türündedir.
- **Telefon:** Doktorun iletişim numarasını belirtir. NVARCHAR veri türünde olup tam 10 karakter içermelidir. Başında '0' olmadan, bitişik şekilde girilmelidir.
- **Eposta:** Doktorun e-posta adresini içerir. NVARCHAR veri türündedir ve maksimum 50 karakter uzunluğundadır.

```
CREATE TABLE Doktor (  
    DoktorID CHAR(10) PRIMARY KEY,  
    Ad NVARCHAR(50) NOT NULL,  
    Soyad NVARCHAR(50) NOT NULL,  
    TcKimlikNo CHAR(11) UNIQUE NOT NULL,  
    UzmanlikAlani NVARCHAR(100) NOT NULL,  
    Telefon NVARCHAR(10) NOT NULL,  
    Eposta NVARCHAR(50) NOT NULL,  
    CHECK(UzmanlikAlani IN('Kardiyoloji','Ortopedi','Nöroloji','Pediatri','Onkoloji','Genel Cerrahi','KB8', 'Dermatoloji'))  
);
```

Görsel 13 : “Doktor” Tablosunu Oluşturan SQL Sorgusu

## 5.7. “DoktorCalismaProgrami” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Bu bölümde, doktorların haftalık çalışma programlarını takip etmek amacıyla bir tablo oluşturulacaktır. Tabloda her kayıta bir doktora ait belirli bir gün ve saat aralığı bilgisi tutulur. DoktorID alanı başka bir tabloya (Doktor) ait yabancı anahtardır. Bu nedenle “DoktorCalismaProgrami” tablosu oluşturulmadan önce “Doktor” tablosunun var olması gerekir.

- **ProgramID:** Birincil anahtardır. Her çalışma programı kaydı için benzersiz bir kimlik numarasıdır. Otomatik artar.
- **DoktorID:** Doktor tablosundaki DoktorID alanına **yabancı anahtar** olarak referans verir. Çalışma saatleri ile doktorlar arasında ilişki kurulmuş olur.
- **Gun:** Haftanın gününü belirtir. NVARCHAR veri türündedir. Girdi kısıtlaması için CHECK komutu ile yalnızca 7 gün adı girilmesi sağlanır.
- **BaslangicSaati:** Doktorun o günkü mesaisinin başlangıç saatini belirtir. TIME veri türünde tanımlanır. Örnek olarak 08:30.
- **BitisSaati:** Doktorun o günkü mesaisinin bitiş saatini belirtir. TIME veri türünde tanımlanır. Örneğin 16:30.

```
CREATE TABLE DoktorCalismaProgrami (  
    ProgramID INT IDENTITY(1,1) PRIMARY KEY,  
    DoktorID CHAR(10) NOT NULL FOREIGN KEY (DoktorID) REFERENCES Doktor(DoktorID),  
    Gun NVARCHAR(10) NOT NULL,  
    BaslangicSaati TIME NOT NULL,  
    BitisSaati TIME NOT NULL,  
    CHECK(Gun IN('Pazartesi','Salı','Çarşamba','Perşembe','Cuma','Cumartesi','Pazar'))  
);
```

Görsel 14: “DoktorCalismaProgrami” Tablosunu Oluşturan SQL Sorgusu

## 5.8. “Randevular” Tablosunun Veri Tabanı Üzerinde Oluşturulması

İçeriğinde RandevuID, HastaID, DoktorID, RandevuTarihi, RandevuSaati, RandevuTuru ve RandevuDurumu alanları bulunan tablo oluşturulacaktır. Bu tablonun amacı hastaların doktorlarla yaptığı randevu kayıtlarını tutmaktır. HastaID ve DoktorID alanları sırasıyla Hasta ve Doktor tablolarına yabancı anahtar (foreign key) olarak tanımlanmalıdır.

- **RandevuID:** Her randevu için benzersiz bir kimlik numarasıdır. PRIMARY KEY olarak tanımlanır. INT veri türündedir. Otomatik olarak artar.
- **HastaID:** Randevunun hangi hastaya ait olduğunu belirtir. FOREIGN KEY olarak Hasta tablosundaki HastaID sütununa referans verir. CHAR veri türündedir.
- **DoktorID:** Randevunun hangi doktor ile yapılacağını belirtir. FOREIGN KEY olarak Doktor tablosundaki DoktorID sütununa referans verir. CHAR veri türündedir.
- **RandevuTarihi:** Randevunun yapılacağı tarihi belirtir. DATE veri türündedir. 'YYYY-MM-DD' formatında girilmelidir.
- **RandevuSaati:** Randevu saatini belirtir. TIME veri türündedir ve 'HH:MM ' formatında girilir.
- **RandevuTuru:** Randevunun türünü belirtir (Muayene, Kontrol, Tahlil vb.). NVARCHAR veri türündedir.
- **RandevuDurumu:** Randevunun geçerli durumunu belirtir (Planlandı, Gerçekleşti, ). NVARCHAR veri türündedir. Varsayılan olarak "Planlandı" değeri atanmıştır.

```
CREATE TABLE Randevular (  
    RandevuID INT IDENTITY(1,1) PRIMARY KEY,  
    HastaID CHAR(10) NOT NULL FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),  
    DoktorID CHAR(10) NOT NULL FOREIGN KEY (DoktorID) REFERENCES Doktor(DoktorID),  
    RandevuTarihi DATE NOT NULL,  
    RandevuSaati TIME NOT NULL,  
    RandevuTuru NVARCHAR(50) NOT NULL,  
    RandevuDurumu NVARCHAR(50) NOT NULL DEFAULT 'Planlandı'  
    CHECK ( RandevuDurumu IN ('Planlandı','Gerçekleşti','İptal Edildi'))  
);
```

Görsel 15 : “Randevular” Tablosunu Oluşturan SQL Sorgusu

## 5.9. “MuayeneKayitlari” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Hastaların doktorlar tarafından gerçekleştirilen muayene bilgilerini tutmak amacıyla oluşturulmuştur. Tablodaki varlıkların açıklaması şu şekildedir:

- **MuayeneID:** Her muayene kaydı için benzersiz bir kimlik numarasıdır ve birincil anahtar olarak kullanılır. CHAR türündedir.
- **HastaID:** Muayene edilen hastanın kimliğini belirtir ve "Hasta" tablosundaki **HastaID** alanına yabancı anahtar olarak bağlanır.
- **DoktorID:** Muayeneyi gerçekleştiren doktoru ifade eder ve "Doktor" tablosundaki **DoktorID** alanına referans verir.
- **MuayeneTarihi:** Muayenenin yapıldığı tarihi belirtir. DATE veri türündedir ve ‘YYYY-MM-DD’ formatında tutulur.
- **Tani:** Doktorun hastaya koyduğu tanıyı içeren açıklama bilgisidir. NVARCHAR türündedir.
- **Tedavi:** Hastaya uygulanan tedavi bilgilerini içerir ve yine NVARCHAR olarak boyutu 255 olan bir değişkende saklanır.

```
CREATE TABLE MuayeneKayitlari (  
    MuayeneID CHAR(10) PRIMARY KEY,  
    HastaID CHAR(10) NOT NULL FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),  
    DoktorID CHAR(10) NOT NULL FOREIGN KEY (DoktorID) REFERENCES Doktor(DoktorID),  
    MuayeneTarihi DATE NOT NULL,  
    Tani NVARCHAR(255) NOT NULL,  
    Tedavi NVARCHAR(255) NOT NULL  
);
```

Görsel 16 : “MuayeneKayitlari” Tablosunu Oluşturan SQL Sorgusu

## 5.10. “Receteler” Tablosunun Veri Tabanı Üzerinde Oluşturulması

**Receteler** tablosu, hastalara yazılan ilaç reçetelerinin detaylarını saklamak için kullanılır. Tablonun içerisindeki değişkenlerin açıklaması şu şekildedir:

- **ReceteID:** Her reçete için benzersiz bir kimlik numarasıdır ve birincil anahtar olarak görev yapar. INT türündedir otomatik olarak artar.
- **MuayeneID:** Reçetenin hangi muayene sonucunda verildiğini belirtir. "MuayeneKayitlari" tablosundaki **MuayeneID** ile ilişkilidir.
- **HastaID:** Reçetenin hangi hastaya ait olduğunu gösterir. "Hasta" tablosundaki **HastaID** ile ilişkilidir.
- **DoktorID:** Reçeteyi yazan doktoru belirtir. "Doktor" tablosundaki **DoktorID** alanına bağlanır.
- **ReceteTarihi:** Reçetenin verildiği tarihi belirtir. DATE veri türünde tanımlanır.
- **IlacAd:** Reçeteye yazılan ilacın adını belirtir ve boyutu 54 olan NVARCHAR türü değişkende saklanır. Örnek olarak Lipitor, Calcimax, Pantil, Sebamed gibi değerler verilebilir.
- **Dozaj:** İlacın kullanım dozu ya da sıklığı hakkında bilgi verir.

```

CREATE TABLE Receteler (
    ReceteID INT IDENTITY(1,1) PRIMARY KEY,
    MuayeneID CHAR(10) NOT NULL FOREIGN KEY (MuayeneID) REFERENCES MuayeneKayitlari(MuayeneID),
    HastaID CHAR(10) NOT NULL FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),
    DoktorID CHAR(10) NOT NULL FOREIGN KEY (DoktorID) REFERENCES Doktor(DoktorID),
    ReceteTarihi DATE NOT NULL,
    IlacAd NVARCHAR(45) NOT NULL,
    Dozaj NVARCHAR(50) NOT NULL,
    CHECK (IlacAd IN (
        -- Kardiyooloji
        'Beloc',
        'Plavix',
        'Norvasc',
        'Lipitor',
        -- Ortopedi
        'Voltaren',
        'Naproxen',
        'Calcimax',
        'Dvit3',
        -- Nöroloji
        'Tegretol',
        'Lustral',
        'Aricept',
        'Neurontin',
        -- Pediatri
        'Parol Şurup',
        'Augmentin',
        'Ventolin Şurup',
        'Zinco',
        -- Onkoloji
        'Zoladex',
        'Herceptin',
        'Xeloda',
        'Neupogen',
        -- Genel Cerrahi
        'Clexane',
        'Flagyl',
        'Pantil',
        'Parol',
        -- KBB
        'Aerius',
        'Rhinocort',
        'Bactroban',
        'Otrivine',
        -- Dermatoloji
        'Madecassol',
        'Fucidin',
        'Acnelyse',
        'Sebamed'
    ))
);

```

**Görsel 17 :** “Receteler” Tablosunu Oluşturan SQL Sorgusu



## 5.11. “Faturalar” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Faturalar tablosu, hastaların almış oldukları hizmetler karşılığında oluşturulan fatura bilgilerini saklamak amacıyla oluşturulmuştur. Tablodaki varlıkların açıklaması şu şekildedir:

- **FaturaID:** Her fatura kaydı için benzersiz bir kimlik numarasıdır ve birincil anahtar olarak kullanılır. INT türündedir. IDENTITY(1,1) ile otomatik olarak artacak şekilde ayarlanmıştır.
- **HastaID:** Fatura edilen hastanın kimliğini belirtir ve "Hasta" tablosundaki HastaID alanına yabancı anahtar olarak bağlanır.
- **DoktorID:** Faturada yer alan hizmeti sağlayan doktoru ifade eder ve "Doktor" tablosundaki DoktorID alanına referans verir.
- **OdemeID:** Fatura ile ilişkilendirilen ödeme bilgisinin kimliğini belirtir ve "Ödemeler" tablosundaki OdemeID alanına yabancı anahtar olarak bağlanır.
- **Ucret:** Hastaya verilen hizmet için talep edilen toplam ücreti içerir. DECIMAL türündedir ve virgülden sonra 2 basamak alır.
- **HizmetTuru:** Sağlanan hizmetin türünü belirtir (örneğin: muayene, tetkik, vb.). NVARCHAR türündedir.
- **FaturaTarihi:** Faturanın düzenlendiği tarihi belirtir. DATE veri türündedir ve ‘YYYY-MM-DD’ formatında tutulur.

```
CREATE TABLE Faturalar (  
    FaturaID INT IDENTITY(1,1) PRIMARY KEY,  
    HastaID CHAR(10) NOT NULL FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),  
    DoktorID CHAR(10) NOT NULL FOREIGN KEY (DoktorID) REFERENCES Doktor(DoktorID),  
    OdemeID INT NOT NULL FOREIGN KEY (OdemeID) REFERENCES Odemeler(OdemeID),  
    Ucret DECIMAL(10, 2) NOT NULL,  
    HizmetTuru NVARCHAR(100) NOT NULL,  
    FaturaTarihi DATE NOT NULL  
);
```

Görsel 18 : “Faturalar” Tablosunu Oluşturan SQL Sorgusu

## 5.12. “Ödemeler” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Bu tablo hastaların muayene ve sağlık hizmetleri için sağladıkları ödeme bilgilerini tutmak amacıyla oluşturulmuştur. Tablodaki varlıkların açıklaması şu şekildedir:

- **ÖdemeID:** Her ödeme kaydına benzersiz bir kimlik numarası atanır ve birincil anahtar olarak kullanılır. INT türündedir. Otomatik olarak artar.
- **ÖdemeYöntem:** Nakit, Sigorta, Kart gibi ödemenin yapıldığı yöntemi belirtir. NVARCHAR türündedir.
- **ÖdemeDurum:** Ödemenin durumunu ifade eder (tamamlandı, beklemede, iptal edildi). NVARCHAR türündedir.

```
CREATE TABLE Ödemeler (  
    ÖdemeID INT IDENTITY(1,1) PRIMARY KEY,  
    ÖdemeYöntem NVARCHAR(50) NOT NULL ,  
    ÖdemeDurum NVARCHAR(50) NOT NULL CHECK (ÖdemeDurum IN('Tamamlandı','İptal Edildi','Beklemede'))  
);
```

Görsel 18 : “Ödemeler” Tablosunu Oluşturan SQL Sorgusu

### 5.13. “LaboratuvarTestleri” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Hastalara yapılan testlerle ilgili bilgileri saklamak amacıyla oluşturulmuştur. Testin kim tarafından yapıldığını, kime ait olduğunu ve testin zamanlamasını belirtir. Tablodaki varlıkların açıklaması şu şekildedir:

- **TestID:** Her laboratuvar testi için benzersiz bir kimlik numarasıdır ve birincil anahtar olarak kullanılır. CHAR türündedir.
- **HastaID:** Testin uygulandığı hastayı belirtir ve "Hasta" tablosundaki HastaID alanına yabancı anahtar olarak bağlanır.
- **DoktorID:** Testi isteyen ya da sonuçlarını değerlendiren doktoru ifade eder ve "Doktor" tablosundaki DoktorID alanına referans verir.
- **NumuneAlimTarihi:** Laboratuvar testine ait numunenin alındığı tarihi belirtir. DATE veri türündedir ve ‘YYYY-MM-DD’ formatında tutulur.
- **SonucTeslimTarihi:** Test sonuçlarının teslim edildiği tarihi gösterir. DATE veri türündedir.

```
CREATE TABLE LaboratuvarTestleri (  
    TestID CHAR(10) PRIMARY KEY,  
    HastaID CHAR(10) NOT NULL FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),  
    DoktorID CHAR(10) NOT NULL FOREIGN KEY (DoktorID) REFERENCES Doktor(DoktorID),  
    NumuneAlimTarihi DATE NOT NULL,  
    SonucTeslimTarihi DATE NOT NULL,  
);
```

Görsel 19 : “LaboratuvarTestleri” Tablosunu Oluşturan SQL Sorgusu

## 5.14. “TestTuru” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Gerçekleştirilen testlerin türlerini sınıflandırmak amacıyla oluşturulmuştur. Her test türü benzersiz bir kimlikle tanımlanır ve test adları tutarsızlığın önüne geçmek için “Doktor” tablosundaki branşlara özgü olacak şekilde sınırlanmıştır. Tablodaki varlıkların açıklaması şu şekildedir:

- **TestTuruID:** Her test türüne ait benzersiz kimlik numarasıdır ve birincil anahtar olarak kullanılır. CHAR(10) veri türündedir.
- **TestAd:** Testin adını belirtir ve NVARCHAR türündedir. Bu alan boş bırakılamaz (NOT NULL) ve sadece önceden tanımlanmış test adlarından biri olabilir. Test sınırlamasına göre Branş-Test ilişkisi aşağıdaki gibidir
  - **Kardiyoloji:** EKG, Efor
  - **Ortopedi:** Kemik Dansitometri, Romatoid Faktör
  - **Nöroloji:** EEG, B12 Vitamin Seviyesi, Beyin MR
  - **Pediyatri:** Hemogram, D Vitamini Seviyesi
  - **Onkoloji:** Tümör Belirteçleri, PET-CT Görüntüleme
  - **Genel Cerrahi:** Kanama Zamanı ve Pıhtılaşma, Ameliyat Öncesi Panel, Karaciğer Fonksiyon
  - **Kulak Burun Boğaz (KBB):** Boğaz Sürüntüsü, Odyometri
  - **Dermatoloji:** Cilt Prick, Biyopsi, Mantar Kültürü

```

CREATE TABLE TestTuru (
    TestTuruID CHAR(10) PRIMARY KEY,
    TestAd NVARCHAR(70) NOT NULL,
    CHECK (TestAd IN (
        -- Kardiyoloji
        'EKG',
        'Efor',
        --Ortopedi
        'Kemik Dansitometri',
        'Romatoid Faktör',
        --Nöroloji
        'EEG',
        'B12 Vitamin Seviyesi',
        'Beyin MR',
        --Pediatri
        'Hemogram',
        'D Vitamini Seviyesi',
        --Onkoloji
        'Tümör Belirteçleri',
        'PET-CT Görüntüleme',
        --Genel Cerrahi
        'Kanama Zamanı ve Pıhtılaşma',
        'Ameliyat Öncesi Panel',
        'Karaciğer Fonksiyon',
        --KBB
        'Boğaz Sürüntüsü',
        'Odyometri',
        --Dermatoloji
        'Cilt Prick',
        'Biyopsi',
        'Mantar Kültürü'
    ))
);

```

**Görsel 20 :** “TestTuru” Tablosunu Oluşturan SQL Sorgusu

## 5.15. “TestDetay” Tablosunun Veri Tabanı Üzerinde Oluşturulması

Yapılan testlerinin sonuçlarını tutmak için oluşturulmuştur. Her detay kaydı bir test türüne ait sonucu ifade eder. Tablodaki varlıkların açıklaması şu şekildedir:

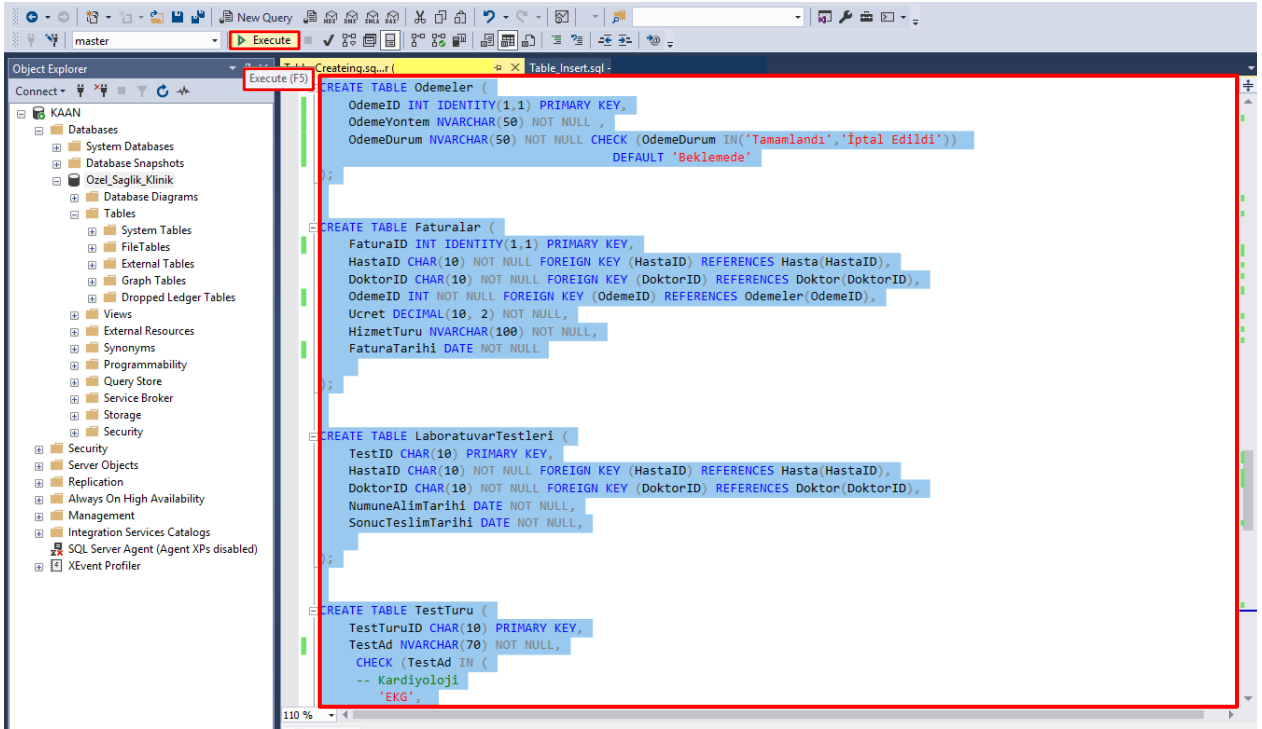
- **TestDetayID:** Her test sonucu kaydı için benzersiz bir kimlik numarasıdır ve birincil anahtar olarak kullanılır. CHAR türündedir.
- **TestTuruID:** Bu test sonucunun hangi test türüne ait olduğunu belirtir ve "TestTuru" tablosundaki TestTuruID alanına yabancı anahtar olarak bağlanır.
- **TestSonucu:** Negatif, Pozitif gibi yapılan testin sonucunu ifade eder. NVARCHAR türündedir ve testin sonucuna göre yorum içerir.

```
CREATE TABLE TestDetay (  
    TestDetayID CHAR(10) PRIMARY KEY,  
    TestTuruID CHAR(10) NOT NULL FOREIGN KEY (TestTuruID) REFERENCES TestTuru(TestTuruID),  
    TestSonucu NVARCHAR(100) NOT NULL CHECK(TestSonucu IN('Negatif','Pozitif')) DEFAULT 'Beklemede'  
);
```

Görsel 21 : “TestDetay” Tablosunu Oluşturan SQL Sorgusu

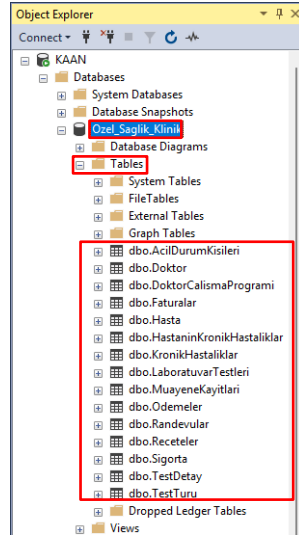
Tablo oluşturmak için sorgu yazma işlemi tamamlandıktan sonra bu SQL sorgularını çalıştırarak tabloları veri tabanına eklememiz gerekir. Bunun için ilk olarak yazdığımız tüm SQL komutlarını seçmeliyiz. Bu işlemi hızlıca yapmak için klavyeden CTRL + A tuş kombinasyonunu kullanabiliriz. Ardından üst menüde yer alan “Execute” butonuna tıklayarak ya da F5 tuşuna basarak sorguları çalıştırabiliriz.

Sorguları çalıştırmadan önce seçtiğimiz veri tabanı altında oluşturmak istediğimiz tablolarla aynı isme sahip tabloların olup olmadığıdır. Kontrol edilmezse sistem aynı isimdeki tabloları yeniden oluşturamayacağı için hata verecektir. Kontrolünü yapmak için sol tarafta yer alan “Object Explorer” penceresinden çalışacağımız veri tabanını bulmalıyız. Sonra altındaki “Tables” klasörünü genişletmeliyiz. Burda oluşturulacak tabloların daha önce eklenip eklenmediğini görebiliriz.



Görsel 22 : SQL Sorgularını Çalıştırılması

Kontroller yapıp tüm sorgular çalıştırdıktan sonra tabloların oluşup oluşmadığını kontrol etmek için tekrar “Object Explorer” penceresinde ilgili veri tabanının “Tables” klasörünü sağ tıklayıp “Refresh” seçeneğini kullanabiliriz. Bu işlemden sonra klasörü genişleterek oluşturulan tabloların listelendiğini görebiliriz.



Görsel 22 : Tabloların Görüntülenmesi

## 6. Tablolara Veri Eklenmesi

Artık tablolar oluşturulduğuna göre her bir tabloya veri ekleme işlemine geçebiliriz. Bu işlemi iki şekilde gerçekleştirebiliriz. İlk seçeneğimiz aynı sorgu dosyası içerisinde devam etmek ya da modüler çalışma tarzına uygun olarak sadece veri ekleme işlemlerine özel yeni bir sorgu dosyası oluşturmak.

Modüler çalışmayı tercih ettiğimiz için yeni bir sorgu dosyası açarak veri ekleme işlemlerini bu dosyada gerçekleştireceğiz.

Veri ekleme sorgularına geçmeden önce, mevcut tabloların boş olup olmadığını kontrol etmek faydalı olacaktır. Bunun için her tabloya özel olarak aşağıdaki sorgu yazıldıktan sonra sadece bu satır seçilip F5 veya “Execute” ile çalıştırılıp kontrol yapılabilir.

### 1) **SELECT \* FROM TABLO\_ADİ;**

Bu sorgu ilgili tablonun içeriği görüntüler ve tablonun boş olup olmadığı anlaşılır ve daha sonra eklenecek verilerin düzgün girilip girilmediği de kontrol edilebilir.

Bu kontrol işlemleri sonrası her tablo için veri ekleme sorguları “INSERT INTO TABLO\_ADİ VALUES()” sorgusuyla yazılıp ardından tabloların son durumlarını gösteren görseller eklenecektir.

```
INSERT INTO Sigorta (SigortaID, SirketAdi, PoliceNo, SigortaBaslangic, SigortaBitis, SigortaTuru) VALUES
('S0001', 'Anadolu Sigorta', 'P123456', '2024-01-01', '2025-01-01', 'Tam Kapsamlı'),
('S0002', 'Allianz', 'P654321', '2024-02-01', '2025-02-01', 'Kapsamlı'),
('S0003', 'Axa Sigorta', 'P789012', '2024-03-01', '2025-03-01', 'Sınırlı'),
('S0004', 'Mapfre', 'P345678', '2024-04-01', '2025-04-01', 'Tam Kapsamlı'),
('S0005', 'Sompo Sigorta', 'P456789', '2024-05-01', '2025-05-01', 'Kapsamlı'),
('S0006', 'Allianz', 'P234567', '2024-06-01', '2025-06-01', 'Sınırlı'),
('S0007', 'Anadolu Sigorta', 'P567890', '2024-07-01', '2025-07-01', 'Tam Kapsamlı'),
('S0008', 'Axa Sigorta', 'P890123', '2024-08-01', '2025-08-01', 'Kapsamlı'),
('S0009', 'Mapfre', 'P012345', '2024-09-01', '2025-09-01', 'Sınırlı'),
('S0010', 'Sompo Sigorta', 'P678901', '2024-10-01', '2025-10-01', 'Tam Kapsamlı'),
('S0011', 'Allianz', 'P123123', '2024-11-01', '2025-11-01', 'Kapsamlı'),
('S0012', 'Anadolu Sigorta', 'P321321', '2024-12-01', '2025-12-01', 'Sınırlı'),
('S0013', 'Allianz', 'P111222', '2024-11-15', '2025-11-15', 'Tam Kapsamlı'),
('S0014', 'Mapfre', 'P333444', '2024-12-10', '2025-12-10', 'Kapsamlı'),
('S0015', 'Sompo Sigorta', 'P555666', '2024-12-20', '2025-12-20', 'Tam Kapsamlı');
```

Görsel 23 : “Sigorta” Tablosuna Veri Eklenmesi



SELECT * FROM Sigorta						
110 %						
Results	Messages					
	SigortaID	SirketAdi	PoliceNo	SigortaBaslangic	SigortaBitis	SigortaTuru
1	S0001	Anadolu Sigorta	P123456	2024-01-01	2025-01-01	Tam Kapsamlı
2	S0002	Allianz	P654321	2024-02-01	2025-02-01	Kapsamlı
3	S0003	Axa Sigorta	P789012	2024-03-01	2025-03-01	Sınırlı
4	S0004	Mapfre	P345678	2024-04-01	2025-04-01	Tam Kapsamlı
5	S0005	Sompo Sigorta	P456789	2024-05-01	2025-05-01	Kapsamlı
6	S0006	Allianz	P234567	2024-06-01	2025-06-01	Sınırlı
7	S0007	Anadolu Sigorta	P567890	2024-07-01	2025-07-01	Tam Kapsamlı
8	S0008	Axa Sigorta	P890123	2024-08-01	2025-08-01	Kapsamlı
9	S0009	Mapfre	P012345	2024-09-01	2025-09-01	Sınırlı
10	S0010	Sompo Sigorta	P678901	2024-10-01	2025-10-01	Tam Kapsamlı
11	S0011	Allianz	P123123	2024-11-01	2025-11-01	Kapsamlı
12	S0012	Anadolu Sigorta	P321321	2024-12-01	2025-12-01	Sınırlı
13	S0013	Allianz	P111222	2024-11-15	2025-11-15	Tam Kapsamlı
14	S0014	Mapfre	P333444	2024-12-10	2025-12-10	Kapsamlı
15	S0015	Sompo Sigorta	P555666	2024-12-20	2025-12-20	Tam Kapsamlı

Görsel 24 : “Sigorta” Tablosuna Eklenen Verilerin Görüntülenmesi

```
INSERT INTO Hasta (HastaID, SigortaID, AcilID, Ad, Soyad, TcKimlikNo, DogumTarihi, Cinsiyet, Telefon, Eposta, Adres) VALUES
('H0001', 'S0001', 'A001', 'Ali', 'Yılmaz', '12345678901', '1990-05-12', 'Erkek', '5051234567', 'ali@mail.com', 'Konya, Türkiye'),
('H0002', 'S0002', 'A002', 'Ayşe', 'Demir', '10987654321', '1985-08-23', 'Kadın', '5329876543', 'ayse@mail.com', 'İstanbul, Türkiye'),
('H0003', 'S0003', 'A003', 'Mehmet', 'Kara', '23456789012', '1975-03-15', 'Erkek', '5413332211', 'mehmet.kara@mail.com', 'Ankara, Türkiye'),
('H0004', 'S0004', 'A004', 'Fatma', 'Çelik', '34567890123', '1992-09-08', 'Kadın', '5438887744', 'fatma.celik@mail.com', 'İzmir, Türkiye'),
('H0005', 'S0005', 'A005', 'Ahmet', 'Öztürk', '45678901234', '1980-12-22', 'Erkek', '5324445566', 'ahmet.ozturk@mail.com', 'Bursa, Türkiye'),
('H0006', 'S0006', 'A006', 'Zeynep', 'Arslan', '56789012345', '1995-06-30', 'Kadın', '5345556677', 'zeynep.arslan@mail.com', 'Antalya, Türkiye'),
('H0007', 'S0007', 'A007', 'Emre', 'Yıldız', '67890123456', '1988-10-05', 'Erkek', '5301112233', 'emre.yildiz@mail.com', 'Eskişehir, Türkiye'),
('H0008', 'S0008', 'A008', 'Elif', 'Koc', '78901234567', '2000-01-20', 'Kadın', '5367778899', 'elif.koc@mail.com', 'Gaziantep, Türkiye'),
('H0009', 'S0009', 'A009', 'Burak', 'Şahin', '89012345678', '1978-11-11', 'Erkek', '5379990001', 'burak.sahin@mail.com', 'Kayseri, Türkiye'),
('H0010', 'S0010', 'A010', 'Merve', 'Aydın', '90123456789', '1993-04-17', 'Kadın', '5381112233', 'merve.aydin@mail.com', 'Trabzon, Türkiye'),
('H0011', 'S0011', 'A011', 'Kerem', 'Balcı', '11223344556', '1982-07-25', 'Erkek', '5394445566', 'kerem.balcı@mail.com', 'Malatya, Türkiye'),
('H0012', 'S0012', 'A012', 'Seda', 'Kılıç', '66778899001', '1996-02-14', 'Kadın', '5318889990', 'seda.kilic@mail.com', 'Adana, Türkiye'),
('H0013', 'S0013', 'A013', 'Deniz', 'Kaya', '22334455667', '1987-09-19', 'Kadın', '5351234567', 'deniz.kaya@mail.com', 'Konya, Türkiye'),
('H0014', 'S0014', 'A014', 'Can', 'Çetin', '33445566778', '1991-12-03', 'Erkek', '5362345678', 'can.cetin@mail.com', 'İstanbul, Türkiye'),
('H0015', 'S0015', 'A015', 'Esra', 'Güneş', '44556677889', '1984-07-27', 'Kadın', '5373456789', 'esra.gunes@mail.com', 'Ankara, Türkiye');
```

Görsel 25 : “Hasta” Tablosuna Veri Eklenmesi

SELECT * FROM Hasta											
110 %											
Results Messages											
	HastaID	SigortaID	AcilID	Ad	Soyad	TcKimlikNo	DogumTarihi	Cinsiyet	Telefon	Eposta	Adres
1	H0001	S0001	A001	Ali	Yılmaz	12345678901	1990-05-12	Erkek	5051234567	ali@mail.com	Konya, Türkiye
2	H0002	S0002	A002	Ayşe	Demir	10987654321	1985-08-23	Kadın	5329876543	ayse@mail.com	İstanbul, Türkiye
3	H0003	S0003	A003	Mehmet	Kara	23456789012	1975-03-15	Erkek	5413332211	mehmet.kara@mail.com	Ankara, Türkiye
4	H0004	S0004	A004	Fatma	Çelik	34567890123	1992-09-08	Kadın	5438887744	fatma.celik@mail.com	İzmir, Türkiye
5	H0005	S0005	A005	Ahmet	Öztürk	45678901234	1980-12-22	Erkek	5324445566	ahmet.ozturk@mail.com	Bursa, Türkiye
6	H0006	S0006	A006	Zeynep	Arslan	56789012345	1995-06-30	Kadın	5345556677	zeynep.arslan@mail.com	Antalya, Türkiye
7	H0007	S0007	A007	Emre	Yıldız	67890123456	1988-10-05	Erkek	5301112233	emre.yildiz@mail.com	Eskişehir, Türkiye
8	H0008	S0008	A008	Elif	Koç	78901234567	2000-01-20	Kadın	5367778899	elif.koc@mail.com	Gaziantep, Türkiye
9	H0009	S0009	A009	Burak	Şahin	89012345678	1978-11-11	Erkek	5379990001	burak.sahin@mail.com	Kayseri, Türkiye
10	H0010	S0010	A010	Merve	Aydın	90123456789	1993-04-17	Kadın	5381112233	merve.aydin@mail.com	Trabzon, Türkiye
11	H0011	S0011	A011	Kerem	Balcı	11223344556	1982-07-25	Erkek	5394445566	kerem.balci@mail.com	Malatya, Türkiye
12	H0012	S0012	A012	Seda	Kılıç	66778899001	1996-02-14	Kadın	5318889990	seda.kilic@mail.com	Adana, Türkiye
13	H0013	S0013	A013	Deniz	Kaya	22334455667	1987-09-19	Kadın	5351234567	deniz.kaya@mail.com	Konya, Türkiye
14	H0014	S0014	A014	Can	Çetin	33445566778	1991-12-03	Erkek	5362345678	can.cetin@mail.com	İstanbul, Türkiye
15	H0015	S0015	A015	Esra	Güneş	44556677889	1984-07-27	Kadın	5373456789	esra.gunes@mail.com	Ankara, Türkiye

Görsel 26 : “Hasta” Tablosuna Eklenen Verilerin Görüntülenmesi

```

INSERT INTO AcilDurumKisileri (AcilID, HastaID, Telefon, AcilAd, AcilSoyad, AcilEposta) VALUES
('A001', 'H0001', '5320000001', 'Mehmet', 'Yılmaz', 'mehmet@mail.com'),
('A002', 'H0002', '5301111111', 'Zeynep', 'Demir', 'zeynep@mail.com'),
('A003', 'H0003', '5332223333', 'Ahmet', 'Kara', 'ahmet.kara@mail.com'),
('A004', 'H0004', '5304445555', 'Fatma', 'Yıldız', 'fatma.yildiz@mail.com'),
('A005', 'H0005', '5336667777', 'İsmail', 'Polat', 'ismail.polat@mail.com'),
('A006', 'H0006', '5308889999', 'Ayşe', 'Öztürk', 'ayse.ozturk@mail.com'),
('A007', 'H0007', '5330001111', 'Deniz', 'Kılıç', 'deniz.kilic@mail.com'),
('A008', 'H0008', '5302223333', 'Emine', 'Toprak', 'emine.toprak@mail.com'),
('A009', 'H0009', '5334445555', 'Selim', 'Demirtaş', 'selim.demirtas@mail.com'),
('A010', 'H0010', '5306667777', 'Gökhan', 'Çelik', 'gokhan.celik@mail.com'),
('A011', 'H0011', '5338889999', 'Zeynep', 'Balkan', 'zeynep.balkan@mail.com'),
('A012', 'H0012', '5300001111', 'Hasan', 'Savaş', 'hasan.savas@mail.com'),
('A013', 'H0013', '5321234567', 'Ayşe', 'Kaya', 'ayse.kaya@mail.com'),
('A014', 'H0014', '5332345678', 'Ali', 'Çetin', 'ali.cetin@mail.com'),
('A015', 'H0015', '5343456789', 'Fatma', 'Güneş', 'fatma.gunes@mail.com');

```

Görsel 27 : “AcilDurumKisileri” Tablosuna Veri Eklenmesi

SELECT * FROM AcilDurumKisileri						
110 %						
Results Messages						
	AcilID	HastaID	Telefon	AcilAd	AcilSoyad	AcilEposta
1	A001	H0001	5320000001	Mehmet	Yılmaz	mehmet@mail.com
2	A002	H0002	5301111111	Zeynep	Demir	zeynep@mail.com
3	A003	H0003	5332223333	Ahmet	Kara	ahmet.kara@mail.com
4	A004	H0004	5304445555	Fatma	Yıldız	fatma.yildiz@mail.com
5	A005	H0005	5336667777	İsmail	Polat	ismail.polat@mail.com
6	A006	H0006	5308889999	Ayşe	Öztürk	ayse.ozturk@mail.com
7	A007	H0007	5330001111	Deniz	Kılıç	deniz.kilic@mail.com
8	A008	H0008	5302223333	Emine	Toprak	emine.toprak@mail.com
9	A009	H0009	5334445555	Selim	Demirtaş	selim.demirtas@mail.com
10	A010	H0010	5306667777	Gökhan	Çelik	gokhan.celik@mail.com
11	A011	H0011	5338889999	Zeynep	Balkan	zeynep.balkan@mail.com
12	A012	H0012	5300001111	Hasan	Savaş	hasan.savas@mail.com
13	A013	H0013	5321234567	Ayşe	Kaya	ayse.kaya@mail.com
14	A014	H0014	5332345678	Ali	Çetin	ali.cetin@mail.com
15	A015	H0015	5343456789	Fatma	Güneş	fatma.gunes@mail.com

Görsel 28 : “AcilDurumKisileri” Tablosuna Eklenen Verilerin Görüntülenmesi

<b>INSERT INTO</b> KronikHastaliklar (HastalikID, HastalikAdi) <b>VALUES</b>	<b>SELECT * FROM</b> KronikHastaliklar
('KH001', 'Hipertansiyon'),	
('KH002', 'Diyabet'),	
('KH003', 'Astım'),	
('KH004', 'Kolesterol Yüksekliği'),	
('KH005', 'Depresyon'),	
('KH006', 'Obezite'),	
('KH007', 'Kanser'),	
('KH008', 'Epilepsi'),	
('KH009', 'Romatizma'),	
('KH010', 'Böbrek Yetmezliği'),	
('KH011', 'Alzheimer'),	
('KH012', 'Karaciğer Hastalıkları'),	
('KH013', 'Migren'),	
('KH014', 'Tansiyon Düşüklüğü'),	
('KH015', 'Multipl Skleroz');	

**Görsel 29 :** “KronikHastaliklar” Tablosuna Veri Eklenmesi ve Eklenen Verilerin Görüntülenmesi

<b>INSERT INTO</b> HastaninKronikHastaliklar (HastaID, HastalikID) <b>VALUES</b>	<b>SELECT * FROM</b> HastaninKronikHastaliklar
('H0001', 'KH001'),	
('H0002', 'KH002'),	
('H0003', 'KH003'),	
('H0004', 'KH004'),	
('H0005', 'KH005'),	
('H0006', 'KH006'),	
('H0007', 'KH007'),	
('H0008', 'KH008'),	
('H0009', 'KH009'),	
('H0010', 'KH010'),	
('H0011', 'KH011'),	
('H0012', 'KH012'),	
('H0001', 'KH003'),	
('H0004', 'KH006'),	
('H0002', 'KH007'),	
('H0008', 'KH005'),	
('H0013', 'KH001'),	
('H0013', 'KH015'),	
('H0014', 'KH002'),	
('H0014', 'KH013'),	
('H0015', 'KH005'),	
('H0015', 'KH009'),	
('H0015', 'KH014'),	
('H0013', 'KH004'),	
('H0014', 'KH010'),	
('H0015', 'KH011'),	

**Görsel 30 :** “HastaninKronikHastaliklar” Tablosuna Veri Eklenmesi ve Eklenen Verilerin Görüntülenmesi

```
INSERT INTO Doktor (DoktorID, Ad, Soyad, TcKimlikNo, UzmanlikAlani, Telefon, Eposta) VALUES
('D0001', 'Murat', 'Koç', '1111111111', 'Kardiyoloji', '5051112222', 'murat.koc@mail.com'),
('D0002', 'Elif', 'Aydın', '2222222222', 'Nöroloji', '5062223333', 'elif.aydin@mail.com'),
('D0003', 'Ahmet', 'Yıldız', '3333333333', 'Ortopedi', '5073334444', 'ahmet.yildiz@mail.com'),
('D0004', 'Zeynep', 'Kara', '4444444444', 'Pediyatri', '5084445555', 'zeynep.kara@mail.com'),
('D0005', 'Mehmet', 'Öztürk', '5555555555', 'Onkoloji', '5095556666', 'mehmet.ozturk@mail.com'),
('D0006', 'Ayşe', 'Demir', '6666666666', 'Genel Cerrahi', '5106667777', 'ayse.demir@mail.com'),
('D0007', 'Cem', 'Aksoy', '7777777777', 'KBB', '5117778888', 'cem.aksoy@mail.com'),
('D0008', 'Selin', 'Erdoğan', '8888888888', 'Dermatoloji', '5128889999', 'selin.erdogan@mail.com'),
('D0009', 'Hakan', 'Şen', '9999999999', 'Nöroloji', '5139990001', 'hakan.sen@mail.com'),
('D0010', 'Burcu', 'Arslan', '1010101010', 'KBB', '5140001111', 'burcu.arslan@mail.com'),
('D0011', 'Tuncay', 'Işık', '1212121212', 'Kardiyoloji', '5151112222', 'tuncay.isik@mail.com'),
('D0012', 'Melis', 'Çelik', '1313131313', 'Ortopedi', '5162223333', 'melis.celik@mail.com'),
('D0013', 'Serdar', 'Gül', '1414141414', 'Kardiyoloji', '5173334444', 'serdar.gul@mail.com'),
('D0014', 'Nazan', 'Yılmaz', '1515151515', 'Ortopedi', '5184445555', 'nazan.yilmaz@mail.com'),
('D0015', 'Özgür', 'Taş', '1616161616', 'Nöroloji', '5195556666', 'ozgur.tas@mail.com');
```

**Görsel 31 :** “Doktor” Tablosuna Veri Eklenmesi

SELECT \* FROM Doktor

	DoktorID	Ad	Soyad	TcKimlikNo	UzmanlikAlani	Telefon	Eposta
1	D0001	Murat	Koç	11111111111	Kardiyoloji	5051112222	murat.koc@mail.com
2	D0002	Elif	Aydın	22222222222	Nöroloji	5062223333	elif.aydin@mail.com
3	D0003	Ahmet	Yıldız	33333333333	Ortopedi	5073334444	ahmet.yildiz@mail.com
4	D0004	Zeynep	Kara	44444444444	Pediyatri	5084445555	zeynep.kara@mail.com
5	D0005	Meh...	Öztürk	55555555555	Onkoloji	5095556666	mehmet.ozturk@mail....
6	D0006	Ayşe	Demir	66666666666	Genel Cerrahi	5106667777	ayse.demir@mail.com
7	D0007	Cem	Aksoy	77777777777	KBB	5117778888	cem.aksoy@mail.com
8	D0008	Selin	Erdo...	88888888888	Dermatoloji	5128889999	selin.erdogan@mail.c...
9	D0009	Hakan	Şen	99999999999	Nöroloji	5139990001	hakan.sen@mail.com
10	D0010	Burcu	Arslan	10101010101	KBB	5140001111	burcu.arslan@mail.com
11	D0011	Tuncay	Işık	12121212121	Kardiyoloji	5151112222	tuncay.isik@mail.com
12	D0012	Melis	Çelik	13131313131	Ortopedi	5162223333	melis.celik@mail.com
13	D0013	Serdar	Gül	14141414141	Kardiyoloji	5173334444	serdar.gul@mail.com
14	D0014	Nazan	Yılmaz	15151515151	Ortopedi	5184445555	nazan.yilmaz@mail.c...
15	D0015	Özgür	Taş	16161616161	Nöroloji	5195556666	ozgur.tas@mail.com

Görsel 32 : “Doktor” Tablosuna Eklenen Verilerin Görüntülenmesi

```
INSERT INTO DoktorCalismaProgrami (DoktorID, Gun, BaslangicSaati, BitisSaati) VALUES
('D0001', 'Pazartesi', '09:00', '17:00'),
('D0002', 'Salı', '10:00', '18:00'),
('D0003', 'Çarşamba', '08:30', '16:30'),
('D0004', 'Perşembe', '09:30', '17:30'),
('D0005', 'Cuma', '08:00', '14:00'),
('D0006', 'Cumartesi', '10:00', '16:00'),
('D0007', 'Pazar', '11:00', '15:00'),
('D0008', 'Pazartesi', '12:00', '18:00'),
('D0009', 'Salı', '09:00', '17:00'),
('D0010', 'Çarşamba', '08:00', '16:00'),
('D0011', 'Perşembe', '10:30', '18:30'),
('D0012', 'Cuma', '09:00', '15:00'),
('D0013', 'Pazartesi', '08:00', '16:00'),
('D0014', 'Çarşamba', '10:00', '18:00'),
('D0015', 'Cuma', '09:30', '17:30');
```

Görsel 33 : “DoktorCalismaProgrami” Tablosuna Veri Eklenmesi

SELECT \* FROM DoktorCalismaProgrami

	ProgramID	DoktorID	Gun	BaslangicSaati	BitisSaati
1	1	D0001	Pazartesi	09:00:00.0000000	17:00:00.0000000
2	2	D0002	Salı	10:00:00.0000000	18:00:00.0000000
3	3	D0003	Çarşamba	08:30:00.0000000	16:30:00.0000000
4	4	D0004	Perşembe	09:30:00.0000000	17:30:00.0000000
5	5	D0005	Cuma	08:00:00.0000000	14:00:00.0000000
6	6	D0006	Cumartesi	10:00:00.0000000	16:00:00.0000000
7	7	D0007	Pazar	11:00:00.0000000	15:00:00.0000000
8	8	D0008	Pazartesi	12:00:00.0000000	18:00:00.0000000
9	9	D0009	Salı	09:00:00.0000000	17:00:00.0000000
10	10	D0010	Çarşamba	08:00:00.0000000	16:00:00.0000000
11	11	D0011	Perşembe	10:30:00.0000000	18:30:00.0000000
12	12	D0012	Cuma	09:00:00.0000000	15:00:00.0000000
13	13	D0013	Pazartesi	08:00:00.0000000	16:00:00.0000000
14	14	D0014	Çarşamba	10:00:00.0000000	18:00:00.0000000
15	15	D0015	Cuma	09:30:00.0000000	17:30:00.0000000

Görsel 34 : “DoktorCalismaProgrami” Tablosuna Eklenen Verilerin Görüntülenmesi

```

INSERT INTO Randevular (HastaID, DoktorID, RandevuTarihi, RandevuSaati, RandevuTuru, RandevuDurumu) VALUES
('H0001', 'D0001', '2025-05-12', '10:00', 'Kontrol', 'Gerçekleşti'),
('H0002', 'D0002', '2025-05-13', '11:30', 'Muayene', 'Planlandı'),
('H0003', 'D0003', '2025-05-14', '14:00', 'Kontrol', 'Planlandı'),
('H0004', 'D0004', '2025-05-15', '09:30', 'Muayene', 'Gerçekleşti'),
('H0005', 'D0005', '2025-05-16', '08:00', 'Kontrol', 'İptal Edildi'),
('H0006', 'D0006', '2025-05-17', '13:30', 'Muayene', 'Planlandı'),
('H0007', 'D0007', '2025-05-18', '16:00', 'Kontrol', 'Gerçekleşti'),
('H0008', 'D0008', '2025-05-19', '10:30', 'Muayene', 'İptal Edildi'),
('H0009', 'D0009', '2025-05-20', '11:00', 'Kontrol', 'Planlandı'),
('H0010', 'D0010', '2025-05-21', '09:00', 'Muayene', 'Gerçekleşti'),
('H0011', 'D0011', '2025-05-22', '15:00', 'Kontrol', 'Planlandı'),
('H0012', 'D0012', '2025-05-23', '12:30', 'Muayene', 'Gerçekleşti'),
('H0013', 'D0013', '2025-05-24', '10:00', 'Muayene', 'Planlandı'),
('H0014', 'D0014', '2025-05-25', '11:30', 'Kontrol', 'Planlandı'),
('H0015', 'D0015', '2025-05-26', '14:00', 'Muayene', 'Planlandı');

```

Görsel 35 : “Randevular” Tablosuna Veri Eklenmesi

SELECT \* FROM Randevular

	RandevuID	HastaID	DoktorID	RandevuTarihi	RandevuSaati	RandevuTuru	RandevuDurumu
1	1	H0001	D0001	2025-05-12	10:00:00.0000000	Kontrol	Gerçekleşti
2	2	H0002	D0002	2025-05-13	11:30:00.0000000	Muayene	Planlandı
3	3	H0003	D0003	2025-05-14	14:00:00.0000000	Kontrol	Planlandı
4	4	H0004	D0004	2025-05-15	09:30:00.0000000	Muayene	Gerçekleşti
5	5	H0005	D0005	2025-05-16	08:00:00.0000000	Kontrol	İptal Edildi
6	6	H0006	D0006	2025-05-17	13:30:00.0000000	Muayene	Planlandı
7	7	H0007	D0007	2025-05-18	16:00:00.0000000	Kontrol	Gerçekleşti
8	8	H0008	D0008	2025-05-19	10:30:00.0000000	Muayene	İptal Edildi
9	9	H0009	D0009	2025-05-20	11:00:00.0000000	Kontrol	Planlandı
10	10	H0010	D0010	2025-05-21	09:00:00.0000000	Muayene	Gerçekleşti
11	11	H0011	D0011	2025-05-22	15:00:00.0000000	Kontrol	Planlandı
12	12	H0012	D0012	2025-05-23	12:30:00.0000000	Muayene	Gerçekleşti
13	13	H0013	D0013	2025-05-24	10:00:00.0000000	Muayene	Planlandı
14	14	H0014	D0014	2025-05-25	11:30:00.0000000	Kontrol	Planlandı
15	15	H0015	D0015	2025-05-26	14:00:00.0000000	Muayene	Planlandı

Görsel 36 : “Randevular” Tablosuna Eklenen Verilerin Görüntülenme

```

INSERT INTO MuayeneKayitlari(MuayeneID, HastaID, DoktorID, MuayeneTarihi, Tani, Tedavi)VALUES
('M0001', 'H0001', 'D0001', '2025-05-12', 'Hipertansiyon kontrolü', 'İlaç düzenlemesi'),
('M0002', 'H0002', 'D0002', '2025-05-12', 'Baş ağrısı', 'Ağrı kesici reçetesi'),
('M0003', 'H0003', 'D0003', '2025-05-13', 'Efor testi sonucu olumlu', 'Beta blokör tedavisi'),
('M0004', 'H0004', 'D0004', '2025-05-14', 'Omuz ağrısı', 'Fiziksel terapi ve ağrı kesici'),
('M0005', 'H0005', 'D0005', '2025-05-15', 'Depresyon semptomları', 'Antidepresan tedavisi'),
('M0006', 'H0006', 'D0006', '2025-05-16', 'B12 vitamini eksikliği', 'B12 takviyesi'),
('M0007', 'H0007', 'D0007', '2025-05-17', 'Sinüs enfeksiyonu', 'Antibiyotik tedavisi'),
('M0008', 'H0008', 'D0008', '2025-05-18', 'Cilt döküntüleri', 'Topikal tedavi ve krem kullanımı'),
('M0009', 'H0009', 'D0009', '2025-05-19', 'Gastrit semptomları', 'Mide koruyucu ilaç tedavisi'),
('M0010', 'H0010', 'D0010', '2025-05-20', 'Kanser taraması', 'Kemoterapi danışmanlığı'),
('M0011', 'H0011', 'D0011', '2025-05-21', 'İrritabl bağırsak sendromu', 'Diyet düzenlemesi ve ilaç tedavisi'),
('M0012', 'H0012', 'D0012', '2025-05-22', 'Alerjik reaksiyon', 'Antihistamin tedavisi'),
('M0013', 'H0001', 'D0013', '2025-05-24', 'Tiroid fonksiyon bozukluğu', 'Hormonal dengeleme tedavisi'),
('M0014', 'H0002', 'D0014', '2025-05-25', 'Kronik bronşit', 'Bronkodilatör ve solunum tedavisi'),
('M0015', 'H0003', 'D0015', '2025-05-26', 'Gastroözofageal reflü', 'Asit bastırıcı ilaç tedavisi');

```

Görsel 37 : “MuayeneKayitlari” Tablosuna Veri Eklenmesi



SELECT \* FROM MuayeneKayıtları

110 %

Results Messages

	MuayeneID	HastaID	DoktorID	MuayeneTarihi	Tani	Tedavi
1	M0001	H0001	D0001	2025-05-12	Hipertansiyon kontrolü	İlaç düzenlemesi
2	M0002	H0002	D0002	2025-05-12	Baş ağrısı	Ağrı kesici reçetesi
3	M0003	H0003	D0003	2025-05-13	Efor testi sonucu olumlu	Beta blokör tedavisi
4	M0004	H0004	D0004	2025-05-14	Omuz ağrısı	Fiziksel terapi ve ağrı kesici
5	M0005	H0005	D0005	2025-05-15	Depresyon belirtileri	Antidepresan tedavisi
6	M0006	H0006	D0006	2025-05-16	B12 vitamini eksikliği	B12 takviyesi
7	M0007	H0007	D0007	2025-05-17	Sinüs enfeksiyonu	Antibiyotik tedavisi
8	M0008	H0008	D0008	2025-05-18	Cilt döküntüleri	Topikal tedavi ve krem kullanımı
9	M0009	H0009	D0009	2025-05-19	Gastrit belirtileri	Mide koruyucu ilaç tedavisi
10	M0010	H0010	D0010	2025-05-20	Kanser taraması	Kemoterapi danışmanlığı
11	M0011	H0011	D0011	2025-05-21	İntestinal bağırsak sendromu	Diyet düzenlemesi ve ilaç tedavisi
12	M0012	H0012	D0012	2025-05-22	Alerjik reaksiyon	Antihistamin tedavisi
13	M0013	H0001	D0013	2025-05-24	Tiroid fonksiyon bozukluğu	Homonal dengeleme tedavisi
14	M0014	H0002	D0014	2025-05-25	Kronik bronşit	Bronkodilatör ve solunum tedavisi
15	M0015	H0003	D0015	2025-05-26	Gastroözofageal reflü	Asit bastırıcı ilaç tedavisi

Görsel 38: “MuayeneKayıtları” Tablosuna Eklenen Verilerin Görüntülenmesi

```
INSERT INTO Receteler (MuayeneID, HastaID, DoktorID, ReceteTarihi, IlacAd, Dozaj) VALUES
('M0001', 'H0001', 'D0001', '2025-05-12', 'Lipitor', '10mg/gün'),
('M0002', 'H0002', 'D0002', '2025-05-12', 'Neurontin', '300mg/gün'),
('M0003', 'H0003', 'D0003', '2025-05-13', 'Calcimax', '500mg/gün'),
('M0004', 'H0004', 'D0004', '2025-05-14', 'Augmentin', '5ml x 2/gün'),
('M0005', 'H0005', 'D0005', '2025-05-15', 'Herceptin', '150mg/hafta'),
('M0006', 'H0006', 'D0006', '2025-05-16', 'Clexane', '40mg/gün'),
('M0007', 'H0007', 'D0007', '2025-05-17', 'Rhinocort', '1 sprey/gün'),
('M0008', 'H0008', 'D0008', '2025-05-18', 'Fucidin', 'Günde 2 kez'),
('M0009', 'H0009', 'D0009', '2025-05-19', 'Plavix', '75mg/gün'),
('M0010', 'H0010', 'D0010', '2025-05-20', 'Lustral', '50mg/gün'),
('M0011', 'H0011', 'D0011', '2025-05-21', 'Naproxen', '250mg x 2/gün'),
('M0012', 'H0012', 'D0012', '2025-05-22', 'Ventolin Şurup', '2.5ml x 3/gün'),
('M0013', 'H0001', 'D0013', '2025-05-24', 'Norvasc', '5mg/gün'),
('M0014', 'H0002', 'D0014', '2025-05-25', 'Parol Şurup', '5ml x 3/gün'),
('M0015', 'H0003', 'D0015', '2025-05-26', 'Pantil', '40mg/gün');
```

Görsel 39 : “Receteler” Tablosuna Veri Eklenmesi

SELECT \* FROM Receteler

110 %

Results Messages

	ReceteID	MuayeneID	HastaID	DoktorID	ReceteTarihi	IlacAd	Dozaj
1	1	M0001	H0001	D0001	2025-05-12	Lipitor	10mg/gün
2	2	M0002	H0002	D0002	2025-05-12	Neurontin	300mg/gün
3	3	M0003	H0003	D0003	2025-05-13	Calcimax	500mg/gün
4	4	M0004	H0004	D0004	2025-05-14	Augmentin	5ml x 2/gün
5	5	M0005	H0005	D0005	2025-05-15	Herceptin	150mg/hafta
6	6	M0006	H0006	D0006	2025-05-16	Clexane	40mg/gün
7	7	M0007	H0007	D0007	2025-05-17	Rhinocort	1 sprey/gün
8	8	M0008	H0008	D0008	2025-05-18	Fucidin	Günde 2 kez
9	9	M0009	H0009	D0009	2025-05-19	Plavix	75mg/gün
10	10	M0010	H0010	D0010	2025-05-20	Lustral	50mg/gün
11	11	M0011	H0011	D0011	2025-05-21	Naproxen	250mg x 2/gün
12	12	M0012	H0012	D0012	2025-05-22	Ventolin Şurup	2.5ml x 3/gün
13	13	M0013	H0001	D0013	2025-05-24	Norvasc	5mg/gün
14	14	M0014	H0002	D0014	2025-05-25	Parol Şurup	5ml x 3/gün
15	15	M0015	H0003	D0015	2025-05-26	Pantil	40mg/gün

Görsel 40: “Receteler” Tablosuna Eklenen Verilerin Görüntülenmesi

INSERT INTO Odemeler (OdemeYontem, OdemeDurum) VALUES	SELECT * FROM Odemeler
('Kredi Kartı', 'Tamamlandı'),	
('Nakit', 'Tamamlandı'),	
('Sigorta', 'Tamamlandı'),	
('Sigorta', 'Tamamlandı'),	
('Kredi Kartı', 'Beklemede'),	
('Nakit', 'İptal Edildi'),	
('Sigorta', 'İptal Edildi'),	
('Sigorta', 'Beklemede'),	
('Kredi Kartı', 'Tamamlandı'),	
('Nakit', 'Beklemede'),	
('Sigorta', 'Tamamlandı'),	
('Sigorta', 'İptal Edildi'),	
('Kredi Kartı', 'Tamamlandı'),	
('Nakit', 'Tamamlandı'),	
('Sigorta', 'Beklemede');	

OdemeID	OdemeYontem	OdemeDurum
1	Kredi Kartı	Tamamlandı
2	Nakit	Tamamlandı
3	Sigorta	Tamamlandı
4	Sigorta	Tamamlandı
5	Kredi Kartı	Beklemede
6	Nakit	İptal Edildi
7	Sigorta	İptal Edildi
8	Sigorta	Beklemede
9	Kredi Kartı	Tamamlandı
10	Nakit	Beklemede
11	Sigorta	Tamamlandı
12	Sigorta	İptal Edildi
13	Kredi Kartı	Tamamlandı
14	Nakit	Tamamlandı
15	Sigorta	Beklemede

**Görsel 41 :** “Odemeler” Tablosuna Veri Eklenmesi ve Eklenen Verilerin Görüntülenmesi

INSERT INTO Faturalar (HastaID, DoktorID, OdemeID, Ucret, HizmetTuru, FaturaTarihi) VALUES
('H0001', 'D0001', 1, 250.00, 'Muayene', '2025-05-12'),
('H0002', 'D0002', 2, 300.00, 'Tetkik', '2025-05-12'),
('H0003', 'D0003', 3, 180.00, 'Kontrol', '2025-05-13'),
('H0004', 'D0004', 4, 220.00, 'Muayene', '2025-05-13'),
('H0005', 'D0005', 5, 350.00, 'Tetkik', '2025-05-14'),
('H0006', 'D0006', 6, 270.00, 'Muayene', '2025-05-14'),
('H0007', 'D0007', 7, 150.00, 'Kontrol', '2025-05-15'),
('H0008', 'D0008', 8, 200.00, 'Tetkik', '2025-05-15'),
('H0009', 'D0009', 9, 280.00, 'Muayene', '2025-05-16'),
('H0010', 'D0010', 10, 320.00, 'Kontrol', '2025-05-16'),
('H0011', 'D0011', 11, 230.00, 'Muayene', '2025-05-17'),
('H0012', 'D0012', 12, 250.00, 'Tetkik', '2025-05-17'),
('H0001', 'D0013', 13, 300.00, 'Muayene', '2025-05-24'),
('H0002', 'D0014', 14, 280.00, 'Tetkik', '2025-05-25'),
('H0003', 'D0015', 15, 260.00, 'Kontrol', '2025-05-26'),
('H0013', 'D0014', 14, 200.00, 'Tetkik', '2025-05-25'),
('H0014', 'D0014', 14, 380.00, 'Tetkik', '2025-05-25'),
('H0015', 'D0014', 14, 180.00, 'Tetkik', '2025-05-25'),
('H0014', 'D0014', 14, 80.00, 'Tetkik', '2025-05-25');

**Görsel 42 :** “Faturalar” Tablosuna Veri Eklenmesi

SELECT * FROM Faturalar						
110 %						
Results Messages						
FaturaID	HastaID	DoktorID	OdemeID	Ucret	Hizmet Turu	Fatura Tarihi
1	H0001	D0001	1	250.00	Muayene	2025-05-12
2	H0002	D0002	2	300.00	Tetkik	2025-05-12
3	H0003	D0003	3	180.00	Kontrol	2025-05-13
4	H0004	D0004	4	220.00	Muayene	2025-05-13
5	H0005	D0005	5	350.00	Tetkik	2025-05-14
6	H0006	D0006	6	270.00	Muayene	2025-05-14
7	H0007	D0007	7	150.00	Kontrol	2025-05-15
8	H0008	D0008	8	200.00	Tetkik	2025-05-15
9	H0009	D0009	9	280.00	Muayene	2025-05-16
10	H0010	D0010	10	320.00	Kontrol	2025-05-16
11	H0011	D0011	11	230.00	Muayene	2025-05-17
12	H0012	D0012	12	250.00	Tetkik	2025-05-17
13	H0001	D0013	13	300.00	Muayene	2025-05-24
14	H0002	D0014	14	280.00	Tetkik	2025-05-25
15	H0003	D0015	15	260.00	Kontrol	2025-05-26
16	H0013	D0014	14	200.00	Tetkik	2025-05-25
17	H0014	D0014	14	380.00	Tetkik	2025-05-25
18	H0015	D0014	14	180.00	Tetkik	2025-05-25
19	H0014	D0014	14	80.00	Tetkik	2025-05-25

**Görsel 43 :** “Faturalar” Tablosuna Eklenen Verilerin Görüntülenmesi

```

INSERT INTO LaboratuvarTestleri (TestID, HastaID, DoktorID, NumuneAlimTarihi, SonucTeslimTarihi)
VALUES
('T0001', 'H0001', 'D0001', '2025-05-12', '2025-05-15'),
('T0002', 'H0002', 'D0002', '2025-05-13', '2025-05-16'),
('T0003', 'H0003', 'D0003', '2025-05-14', '2025-05-17'),
('T0004', 'H0004', 'D0004', '2025-05-15', '2025-05-18'),
('T0005', 'H0005', 'D0005', '2025-05-16', '2025-05-19'),
('T0006', 'H0006', 'D0006', '2025-05-17', '2025-05-20'),
('T0007', 'H0007', 'D0007', '2025-05-18', '2025-05-21'),
('T0008', 'H0008', 'D0008', '2025-05-19', '2025-05-22'),
('T0009', 'H0009', 'D0009', '2025-05-20', '2025-05-23'),
('T0010', 'H0010', 'D0010', '2025-05-21', '2025-05-24'),
('T0011', 'H0011', 'D0011', '2025-05-22', '2025-05-25'),
('T0012', 'H0012', 'D0012', '2025-05-23', '2025-05-26'),
('T0013', 'H0013', 'D0013', '2025-05-24', '2025-05-27'),
('T0014', 'H0014', 'D0014', '2025-05-25', '2025-05-28'),
('T0015', 'H0015', 'D0015', '2025-05-26', '2025-05-29');

```

Görsel 44 : “LaboratuvarTestleri” Tablosuna Veri Eklmesi

SELECT \* FROM LaboratuvarTestleri

	TestID	HastaID	DoktorID	NumuneAlimTarihi	SonucTeslimTarihi
1	T0001	H0001	D0001	2025-05-12	2025-05-15
2	T0002	H0002	D0002	2025-05-13	2025-05-16
3	T0003	H0003	D0003	2025-05-14	2025-05-17
4	T0004	H0004	D0004	2025-05-15	2025-05-18
5	T0005	H0005	D0005	2025-05-16	2025-05-19
6	T0006	H0006	D0006	2025-05-17	2025-05-20
7	T0007	H0007	D0007	2025-05-18	2025-05-21
8	T0008	H0008	D0008	2025-05-19	2025-05-22
9	T0009	H0009	D0009	2025-05-20	2025-05-23
10	T0010	H0010	D0010	2025-05-21	2025-05-24
11	T0011	H0011	D0011	2025-05-22	2025-05-25
12	T0012	H0012	D0012	2025-05-23	2025-05-26
13	T0013	H0013	D0013	2025-05-24	2025-05-27
14	T0014	H0014	D0014	2025-05-25	2025-05-28
15	T0015	H0015	D0015	2025-05-26	2025-05-29

Görsel 45 : “LaboratuvarTestleri” Tablosuna Eklmesi Verilerin Görüntülenmesi

```

INSERT INTO TestTuru (TestTuruID, TestAd) VALUES
('TT001', 'EKG'),
('TT002', 'EEG'),
('TT003', 'Efor'),
('TT004', 'Kemik Dansitometri'),
('TT005', 'Romatoid Faktör'),
('TT006', 'B12 Vitamin Seviyesi'),
('TT007', 'Beyin MR'),
('TT008', 'Hemogram'),
('TT009', 'D Vitamini Seviyesi'),
('TT010', 'Tümör Belirteçleri'),
('TT011', 'PET-CT Görüntüleme'),
('TT012', 'Kanama Zamanı ve Pıhtılaşma'),
('TT013', 'Ameliyat Öncesi Panel'),
('TT014', 'Karaciğer Fonksiyon'),
('TT015', 'Boğaz Sürüntüsü'),
('TT016', 'Odyometri'),
('TT017', 'Cilt Prick'),
('TT018', 'Biyopsi'),
('TT019', 'Mantar Kültürü');

```

SELECT \* FROM TestTuru

	TestTuruID	TestAd
1	TT001	EKG
2	TT002	EEG
3	TT003	Efor
4	TT004	Kemik Dansitometri
5	TT005	Romatoid Faktör
6	TT006	B12 Vitamin Seviyesi
7	TT007	Beyin MR
8	TT008	Hemogram
9	TT009	D Vitamini Seviyesi
10	TT010	Tümör Belirteçleri
11	TT011	PET-CT Görüntüleme
12	TT012	Kanama Zamanı v...
13	TT013	Ameliyat Öncesi P...
14	TT014	Karaciğer Fonksiyon
15	TT015	Boğaz Sürüntüsü
16	TT016	Odyometri
17	TT017	Cilt Prick
18	TT018	Biyopsi
19	TT019	Mantar Kültürü

Görsel 46 : “TestTuru” Tablosuna Veri Eklmesi ve Eklmesi Verilerin Görüntülenmesi



INSERT INTO TestDetay (TestDetayID, TestTuruID, TestSonucu) VALUES	SELECT * FROM TestDetay
('TD001', 'TT001', 'Pozitif'), -- EKG	
('TD002', 'TT002', 'Negatif'), -- EEG	
('TD003', 'TT003', 'Negatif'), -- Efor	
('TD004', 'TT004', 'Pozitif'), -- Kemik Dansitometri	
('TD005', 'TT005', 'Negatif'), -- Romatoid Faktör	
('TD006', 'TT006', 'Pozitif'), -- B12 Vitamin	
('TD007', 'TT007', 'Pozitif'), -- Beyin MR	
('TD008', 'TT008', 'Negatif'), -- Hemogram	
('TD009', 'TT009', 'Pozitif'), -- D Vitamini	
('TD010', 'TT010', 'Pozitif'), -- Tümör Belirteçleri	
('TD011', 'TT011', 'Pozitif'), -- PET-CT	
('TD012', 'TT012', 'Pozitif'), -- Kanama Zamanı	
('TD013', 'TT013', 'Negatif'), -- Ameliyat Öncesi Panel	
('TD014', 'TT014', 'Negatif'), -- Karaciğer Fonksiyon	
('TD015', 'TT015', 'Pozitif'), -- Boğaz Sürüntüsü	
('TD016', 'TT016', 'Pozitif'), -- Odyometri	
('TD017', 'TT017', 'Negatif'), -- Cilt Prick	
('TD018', 'TT018', 'Negatif'), -- Biyopsi	
('TD019', 'TT019', 'Pozitif'), -- Mantar Kültürü	

Görsel 47 : “TestDetay” Tablosuna Veri Eklmesi ve Eklmesi Verilerin Görüntülenmesi

## 7. Veri Tabanı Sorgu Örnekleri

Bu bölümde artık hazır olan veri tabanımızın üzerinde sorgu örnekleri uygulanacaktır. Bu örneklerin içinde Basit, Saklı Yordam (Stored Procedure) ve Tetikleyici (Trigger) yöntemleri kullanılacaktır.

### 7.1. Basit Sorgu Örnekleri

- 1) Aşağıdaki resimde gösterilen SQL sorgusu “M0001”e ait muayene kaydının reçete bilgilerini getirir. Muayenin reçetesindeki hasta, doktor, reçete tarihi, ilaç adı ve dozaj bilgileri MuayeneKayitlari, Receteler, Hasta ve Doktor tabloları birleştirilerek gösterilir.

```
SELECT
    m.MuayeneID,
    h.Ad AS HastaAdi,
    h.Soyad AS HastaSoyadi,
    d.Ad AS DoktorAdi,
    d.Soyad AS DoktorSoyadi,
    r.ReceteTarihi,
    r.IlacAd,
    r.Dozaj
FROM Receteler r
JOIN MuayeneKayitlari m ON r.MuayeneID = m.MuayeneID
JOIN Hasta h ON r.HastaID = h.HastaID
JOIN Doktor d ON r.DoktorID = d.DoktorID
WHERE r.MuayeneID = 'M0001';
```

110 %

Results

Messages

	MuayeneID	HastaAdi	HastaSoyadi	DoktorAdi	DoktorSoyadi	ReceteTarihi	IlacAd	Dozaj
1	M0001	Ali	Yılmaz	Murat	Koç	2025-05-12	Lipitor	10mg/gün

Görsel 48 : “M001” e Ait Reçete Bilgilerini Getiren Sorgu ve Çıktısı

- 2) Aşağıdaki resimde gösterilen SQL sorgusu kronik hastalığa sahip olan hastaların isimlerini ve soyadlarını getirir. HastalıkAdi ile hastanın adı ve soyadı “Hasta” ve ”KronikHastaliklar” tablosu birbirine join edilerek getirilir. Bir hastanın birden fazla kronik hastalığının olabileceği de unutulmamalıdır.

```
SELECT
    h.Ad AS HastaAdi,
    h.Soyad AS HastaSoyadi,
    kh.HastalikAdi
FROM HastaninKronikHastaliklar kh
JOIN Hasta h ON kh.HastaID = h.HastaID
JOIN KronikHastaliklar kh ON kh.HastalikID = kh.HastalikID;
```

	HastaAdi	HastaSoyadi	HastalikAdi
1	Ali	Yilmaz	Hipertansiyon
2	Ali	Yilmaz	Astım
3	Ayşe	Demir	Diyabet
4	Ayşe	Demir	Kanser
5	Mehmet	Kara	Astım
6	Fatma	Çelik	Kolesterol Yüksekliği
7	Fatma	Çelik	Obezite
8	Ahmet	Öztürk	Depresyon
9	Zeynep	Anılan	Obezite
10	Emre	Yıldız	Kanser
11	Elif	Koç	Depresyon
12	Elif	Koç	Epilepsi
13	Burak	Şahin	Romatizma
14	Merve	Aydın	Böbrek Yetmezliği
15	Kerem	Balci	Alzheimer
16	Seda	Kılıç	Karaciğer Hastalıkları

**Görsel 48 :** Kronik Hastalığı Olan Hastaların İsimlerini, Soyadlarını ve Hastalıklarını Listeleyen Sorgu ve Çıktısı

- 3) Aşağıdaki SQL sorgusu, Faturalar tablosundaki her hastanın ödediği toplam ücreti hesaplar “GROUP BY HastaID” ile her hasta için ayrı bir grup oluşturulur. “HAVING SUM(Ucret) > 200” ile toplam ücreti 200'den fazla olan hastaları filtreler. Bu hastaların HastaID ve ToplamUcret bilgilerini döndürür.

```
select HastaID, SUM(Ucret) AS ToplamUcret From Faturalar
GROUP BY HastaID
HAVING SUM(Ucret) > 200;
```

	HastaID	ToplamUcret
1	H0001	550.00
2	H0002	580.00
3	H0003	440.00
4	H0004	220.00
5	H0005	350.00
6	H0006	270.00
7	H0009	280.00
8	H0010	320.00
9	H0011	230.00
10	H0012	250.00
11	H0014	460.00

**Görsel 49 :** Toplam Ücreti 200'den Az Olan Hastaları Filtreleyen Sorgu ve Çıktısı

- 4) Aşağıdaki resimde bulunan T-SQL sorgusu belirli bir hastanın kimlik, iletişim, sigorta bilgileri ve acil durumda ulaşılacak kişinin bilgilerini listeler. Hasta, Sigorta ve AcilDurumKisileri tabloları hasta ID'si üzerinden birleştirilmiştir. “@istenen\_hasta” değişkeni ile bilgileri istenen hasta getirilir.

```
DECLARE @istenen_hasta CHAR(10);
PRINT 'Hasta ID giriniz (örnek: H0001): ';
SET @istenen_hasta = 'H0009';
SELECT
    h.HastaID,
    h.Ad,
    h.Soyad,
    h.TcKimlikNo,
    h.DogumTarihi,
    h.Cinsiyet,
    h.Telefon,
    h.Eposta,
    s.SirketAdi,
    s.PoliceNo,
    s.SigortaTuru,
    a.AcilAd,
    a.AcilSoyad,
    a.Telefon AS AcilTelefon
FROM Hasta h
JOIN Sigorta s ON h.SigortaID = s.SigortaID
JOIN AcilDurumKisileri a ON h.AcilID = a.AcilID
WHERE h.HastaID = @istenen_hasta;
```

HastaID	Ad	Soyad	TcKimlikNo	DogumTarihi	Cinsiyet	Telefon	Eposta	SirketAdi	PoliceNo	SigortaTuru	AcilAd	AcilSoyad	AcilTelefon
H0009	Burak	Şahin	89012345678	1978-11-11	Erkek	5379990001	burak.sahin@mail.com	Mapfre	P012345	Sırlı	Selim	Demirtaş	5334445555

Görsel 50 : Belirli Bir Hastanın Acil ve Sigorta Bilgileri Dahil Bütün Bilgilerini Getirir

- 5) Bu SQL sorgusu hangi testten kaç tane pozitif ve kaç tane negatif sonuç çıktığını sayar. TestDetay ve TestTuru tabloları birleştirilmiştir. Her test adı için pozitif ve negatif sonuçlar ayrı ayrı sayılır. Sonuçlar test adına göre sıralanır ve gruplanır.

```
SELECT
    tt.TestAd,
    SUM(CASE WHEN td.TestSonucu = 'Pozitif' THEN 1 ELSE 0 END) AS PozitifSayisi,
    SUM(CASE WHEN td.TestSonucu = 'Negatif' THEN 1 ELSE 0 END) AS NegatifSayisi
FROM TestDetay td
JOIN TestTuru tt ON td.TestTuruID = tt.TestTuruID
GROUP BY tt.TestAd
ORDER BY tt.TestAd;
```

TestAd	Pozitif Sayısı	Negatif Sayısı
1 Ameliyat Öncesi Panel	0	1
2 B12 Vitamini Seviyesi	1	0
3 Beyin MR	1	0
4 Biyopsi	0	1
5 Boğaz Sürüntüsü	1	0
6 Cilt Pisk	0	1
7 D Vitamini Seviyesi	1	0
8 EEG	0	1
9 Efor	0	1
10 EKG	1	0
11 Hemogram	0	1
12 Kanama Zamanı ve Pıhtılaşma	1	0
13 Karaciğer Fonksiyon	0	1
14 Kemik Danışmanlığı	1	0
15 Mantar Kültürü	1	0
16 Odyometri	1	0
17 PET-CT Görüntüleme	1	0
18 Romatoid Faktör	0	1
19 Tümör Belirteçleri	1	0

Görsel 51 : Her Test Türü İçin Kaç Adet Pozitif ve Negatif Sonuç Olduğunu Gösteren Sorgu ve Çıktısı

- 6) Bu SQL sorgusu sigorta türü "Sınırlı" olan ve henüz süresi dolmamış sigortaları bulur. Ayrıca bu sigortaların ait olduğu hastaların adını ve soyadını, sigorta şirketi adını ve sigorta türünü listeler.

```
SELECT
    s.SirketAdi AS SigortaSirketi,
    s.SigortaTuru,
    s.SigortaBitis,
    h.Ad + ' ' + h.Soyad AS HastaAdi
FROM Sigorta s
JOIN Hasta h ON s.SigortaID = h.SigortaID
WHERE
    s.SigortaTuru = 'Sınırlı'
    AND s.SigortaBitis > GETDATE();
```

121 %

Results Messages

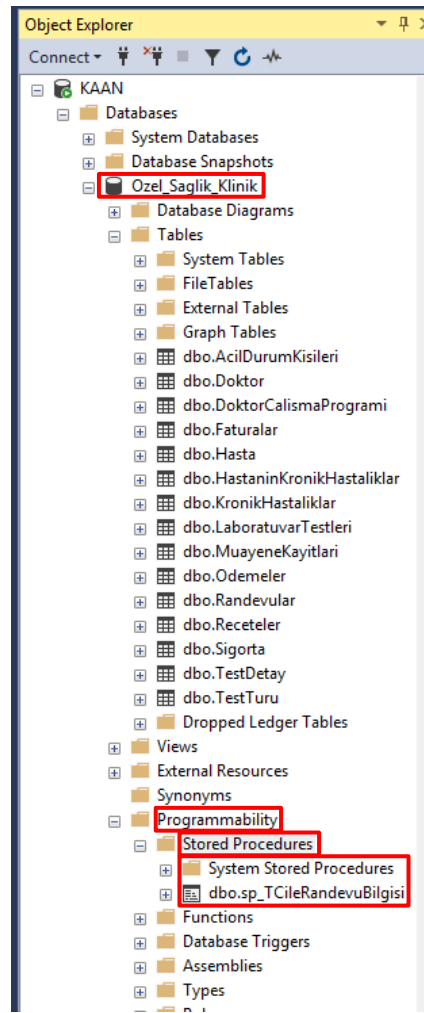
	SigortaSirketi	SigortaTuru	SigortaBitis	HastaAdi
1	Allianz	Sınırlı	2025-06-01	Zeynep Arslan
2	Mapfre	Sınırlı	2025-09-01	Burak Şahin
3	Anadolu Sigorta	Sınırlı	2025-12-01	Seda Kılıç

**Görsel 52 :** “Sınırlı” Türe Sahip ve Tarihi Geçerli Olan Sigortaya Sahip Hasta Adı ve Soyadını Birleştirerek Yazan Sorgu ve Çıktısı

## 7.2. Saklı Yordamlar (Stored Procedures) Örnekleri

Saklı yordam (Stored Procedure) veritabanında depolanan ve belirli bir görevi yerine getiren SQL komutları kümesidir. Kullanıcılar ve uygulamalar bu yordamları veri tabanında istedikleri zaman çağırarak çalıştırabilirler. Özellikle birden fazla kez kullanılması gereken işlemleri tek bir saklı yordamda yazarak tekrarı azaltabilir, işlem güvenliğini artırabilir ve sorgunun okunmasını kolaylaştırabiliriz.

Bu saklı yordamlar, yine "Object Explorer" penceresinde bulunan kendi veri tabanımızda "Programmability → Stored Procedures" yolunu takip ederek bulabileceğimiz bir yerde saklanır. "Stored Procedures" üzerine sağ tıklanıp "New... → Stored Procedure" seçeneği ile yeni bir saklı yordam oluşturabilir veya "New Query" ile yeni bir sorgu dosyası açıp saklı yordamımızı burada yazarak "Execute" butonuyla veri tabanımıza ekleyebiliriz.



**Görsel 53 :** Oluşturulmuş Saklı Yordamların (Stored Procedures) Konumu

Aşağıdaki görselde TC Kimlik Numarası girilen bir hastanın, hangi doktora ve ne zaman muayene olacağını gösteren bir saklı yordam örneği ve çıktısı gösterilmiştir.

```
CREATE PROCEDURE sp_TCileRandevuBilgisi
    @TcKimlikNo CHAR(11)
AS
BEGIN
    SELECT
        h.Ad AS Hasta_Ad,
        h.Soyad AS Hasta_Soyad,
        d.Ad AS Doktor_Ad,
        d.Soyad AS Doktor_Soyad,
        d.UzmanlikAlani,
        m.MuayeneTarihi
    FROM Hasta h
    JOIN MuayeneKayitlari m ON h.HastaID = m.HastaID
    JOIN Doktor d ON m.DoktorID = d.DoktorID
    WHERE h.TcKimlikNo = @TcKimlikNo
    ORDER BY m.MuayeneTarihi DESC;
END;
```

**Görsel 54 :** “sp\_TCileRandevuBilgisi” Adlı Saklı Yordamın Yapısı

EXEC sp\_TCileRandevuBilgisi @TcKimlikNo = '12345678901' ;

121 %

Results Messages

	HastaAd	HastaSoyad	DoktorAd	DoktorSoyad	UzmanlikAlani	MuayeneTarihi
1	Ali	Yılmaz	Murat	Koç	Kardiyoloji	2025-05-12

**Görsel 55 :** “sp\_TCileRandevuBilgisi” Adlı Saklı Yordamın Çağırılması ve Sorgunun Sonucu

Aşağıdaki görselde bulunan saklı yordam hastanın telefon numarasını kontrol ederek ona göre fatura ve ödeme bilgilerini alan ve ücret toplamalarını hesaplayan bir saklı yordam yazılmıştır. Çıktıda ise hastanın ad ve soyadını birleşik bir şekilde yazar ve toplam ücreti virgülden sonra 2 basamak olacak şekilde yazar. hastanın birden fazla faturası olabileceği ve bu faturaların toplam ücretini almak nedeniyle en sonda “GROUP BY” ile gruplama yapılır.

```
CREATE PROCEDURE sp_HastaTelefonIleToplamUcret
@Telefon NVARCHAR(10)
AS
BEGIN
SELECT
h.Ad + ' ' + h.Soyad AS HastaAdi,
h.Telefon AS HastaTelefon,
SUM(f.Ucret) AS ToplamUcret
FROM
Faturalar f
JOIN Hasta h ON f.HastaID = h.HastaID
WHERE
h.Telefon = @Telefon
GROUP BY
h.Ad, h.Soyad, h.Telefon;
END;
```

**Görsel 56 :** “sp\_HastaTelefonIleToplamUcret” Adlı Saklı Yordam Yapısı

EXEC sp\_HastaTelefonIleToplamUcret @Telefon = '5413332211';

121 %

Results Messages

	HastaAdi	HastaTelefon	ToplamUcret
1	Mehmet Kara	5413332211	180.00

**Görsel 57 :** “sp\_HastaTelefonIleToplamUcret” Adlı Saklı Yordamın Çağırılması ve Sorgunun Sonucu

### 7.3. Trigger (Tetikleyici) Örneği

Trigger (tetikleyici) veritabanı yönetim sistemlerinde INSERT, UPDATE, DELETE ile verilerde değişiklik olduğunda otomatik olarak kendine atanmış görevi gerçekleştiren araçlardır. Özellikle denetim, veri bütünlüğü sağlama, tablolar arası otomatik işlem yönetimi gibi durumlarda kolaylık sağlar.

Tetikleyicilerin uygun senaryolarda kullanılması önemlidir. Çünkü özellikle fazla verinin sistemlerde karmaşık ve iç içe geçmiş trigger yapıları sisteme ciddi yük bindirebilir ve hata ayıklamayı zorlaştırabilir. Ayrıca halihazırda bulunan tetikleyicilerin birçok kez kullanılması da gereksiz karmaşıklığa yol açabilir. Bu nedenle tetikleyicileri verimli bir şekilde kullanılması için sistem gereksinimleri ve mimari bütünlüğüne göre tanımlanmalı ve sorgusu yazılmalıdır.

Aşağıdaki resimde oluşturduğumuz veri tabanına uygun bir tetikleyici örneği görülebilir. Bu tetikleyici “Faturalar” tablosunda bulunan “Ucret” alanında fiyat yükselmesi olması durumunda zammın ne zaman yapıldığını, bu zamdan kimin etkilendiğini, eski ücretle birlikte yeni ücreti ve zam yüzdesini tutacak ve bir log tablosuna yazacaktır. Bu tablo ile birlikte gelecekte ücret nedeniyle oluşabilecek bir şikayet durumuna veya denetlemeye karşı bir önlem alınması planlanmaktadır.

Bu tetikleyicinin çalışabilmesi için, önceden belirlenmiş bir log tablosuna ihtiyaç vardır. “FaturaGuncellemeLog” isimli log tablosu oluşturularak bu ihtiyaç giderilir. Bu tabloda aşağıdaki sütunları bulunacaktır:

- HastaID: Güncelleme yapılan fatura kaydına ait hasta kimliğini tutar.
- OdemeID: Güncelleme yapılan faturanın ödeme kimliğini belirtir.
- EskiUcret: Güncelleme öncesinde fatura üzerinde kayıtlı olan ücret bilgisini tutar. “Faturalar” tablosunda “Ucret” sütunun veri tipiyle uyumlu olması için bu alan da DECIMAL(10,2) olarak tanımlanır.
- YeniUcret: Güncelleme sonrasında geçerli olan yeni ücret bilgisini tutar. Bu sütun da “Faturalar” tablosunda “Ucret” sütunun veri tipiyle uyumlu olması için DECIMAL(10,2) veri tipinde tanımlanmalıdır.
- ZamYuzdesi: Ücretteki değişim oranını yüzdesel olarak ifade eder. DECIMAL(4,2) veri tipindedir.
- GuncellemeTarihi: Ücret güncellemesinin yapıldığı tarih ve saati sistem zamanı (GETDATE()) ile kaydeder.

```

CREATE TABLE FaturaGuncellemeLog (
    LogID INT IDENTITY(1,1) PRIMARY KEY,
    HastaID CHAR(10),
    OdemeID INT,
    EskiUcret DECIMAL(10,2),
    YeniUcret DECIMAL(10,2),
    ZamYuzde DECIMAL(4,2),
    GuncellemeTarihi DATETIME DEFAULT GETDATE()
);

```

**Görsel 58 :** “FaturaGuncellemeLog” Log Tablosunu Yapısı ve Oluşturan Sorgu

Tablomuz oluşturduktan sonra bu tabloya tetiklendiğinde verileri yazacak bir Trigger (Tetikleyici) yazılması gerekmektedir. “trg\_FaturaUcretGuncelleme” isimli bu trigger "Faturalar" tablosunda bir güncelleme işlemi yapıldığında otomatik olarak devreye girer. Kayıtta "Ucret" sütunundaki değer değiştiğinde tetiklenir. Güncelleme gerçekleştiğinde tetikleyici değişiklik öncesi ve sonrası ücret değerlerini karşılaştırır. Eğer ücrette bir farklılık varsa bu değişikliğe ait bilgileri "FaturaGuncellemeLog" adlı log tablosuna kaydeder. "Faturalar" tablosundaki ücret değişiklikleri otomatik olarak kayıt altına alınmış olur. Kullanıcının herhangi ekstra bir işlem yapmasına gerek yoktur sadece "Faturalar" tablosunda ücret güncellemesi yapılması trigger'ın çalışması için yeterlidir.

```

CREATE TRIGGER trg_FaturaUcretGuncelleme
ON Faturalar
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.FaturaID = d.FaturaID
        WHERE i.Ucret <> d.Ucret
    )
    BEGIN
        INSERT INTO FaturaGuncellemeLog ( HastaID, OdemeID, EskiUcret, YeniUcret, ZamYuzde, GuncellemeTarihi)
        SELECT
            i.HastaID,
            i.OdemeID,
            d.Ucret AS EskiUcret,
            i.Ucret AS YeniUcret,
            CASE WHEN
                d.Ucret = 0 THEN 0
            ELSE ROUND(((i.Ucret - d.Ucret) / d.Ucret) * 100.0, 2)
            END AS ZamYuzde,
            GETDATE()
        FROM inserted i
        JOIN deleted d ON i.FaturaID = d.FaturaID
        WHERE i.Ucret <> d.Ucret;
    END
END;

```

**Görsel 59 :** “trg\_FaturaGuncelleme” Tetikleyicisinin Yapısı ve Tetikleyiciyi Oluşturan Sorgu



Bir örnekle de bu tetikleyicinin ve tablonun birbirleriyle düzgün çalışıp çalışmadığını görmemiz gerekir. Örnek olarak “Fatura” tablosundaki “FaturaID” si 1 ve 7 olan satırlarda değişikliğe gidelim. 1 ID’sine sahip faturanın ücretini 250.00’den 300.00’a, ID’si 7 olan faturanın ücretini 150.00’dan 280.00’a .ıkarmış olalım.

```
select * from Faturalar
```

	FaturaID	HastaID	DoktorID	OdemeID	Ucret	Hizmet Turu	Fatura Tarihi
1	1	H0001	D0001	1	250.00	Muayene	2025-05-12
2	2	H0002	D0002	2	300.00	Tetkik	2025-05-12
3	3	H0003	D0003	3	180.00	Kontrol	2025-05-13
4	4	H0004	D0004	4	220.00	Muayene	2025-05-13
5	5	H0005	D0005	5	350.00	Tetkik	2025-05-14
6	6	H0006	D0006	6	270.00	Muayene	2025-05-14
7	7	H0007	D0007	7	150.00	Kontrol	2025-05-15
8	8	H0008	D0008	8	200.00	Tetkik	2025-05-15
9	9	H0009	D0009	9	280.00	Muayene	2025-05-16
10	10	H0010	D0010	10	320.00	Kontrol	2025-05-16
11	11	H0011	D0011	11	230.00	Muayene	2025-05-17
12	12	H0012	D0012	12	250.00	Tetkik	2025-05-17

Görsel 60 : Tetiklenme Öncesi 1 ve 7 ID’li Fatura Satırları

```
UPDATE Faturalar SET Ucret = 300.00 WHERE FaturaID = 1;
```

```
UPDATE Faturalar SET Ucret = 280.00 WHERE FaturaID = 7;
```

Görsel 61 : Tetikleme İşleminin Çalışması İçin “UPDATE” Sorguları

Bu işlem sonrası “FaturaGuncellemeLog” tablosunda değişiklik olup olmadığını görebilmek için “select \* from FaturaGuncellemeLog” sorgusu yazılarak görülebilir. Log tablosunda değişiklik var ve düzgün kayıt yapılmış ise “Faturalar” tablosunda ilgili alanların da değiştiğini “select \* from Faturalar” ile kontrol edebiliriz.

select \* from FaturaGuncellemeLog

110 %

Results Messages

	LogID	HastaID	OdemeID	EskiUcret	YeniUcret	ZamYuzde	Guncelleme Tarihi
1	1	H0001	1	250.00	300.00	20.00	2025-05-15 14:32:32.490
2	2	H0007	7	150.00	280.00	86.67	2025-05-15 15:05:58.403

Görsel 62 : Log Tabosu Kontrolü

select \* from Faturalar

110 %

Results Messages

	FaturaID	HastaID	DoktorID	OdemeID	Ucret	HizmetTuru	Fatura Tarihi
1	1	H0001	D0001	1	300.00	Muayene	2025-05-12
2	2	H0002	D0002	2	300.00	Tetkik	2025-05-12
3	3	H0003	D0003	3	180.00	Kontrol	2025-05-13
4	4	H0004	D0004	4	220.00	Muayene	2025-05-13
5	5	H0005	D0005	5	350.00	Tetkik	2025-05-14
6	6	H0006	D0006	6	270.00	Muayene	2025-05-14
7	7	H0007	D0007	7	280.00	Kontrol	2025-05-15
8	8	H0008	D0008	8	200.00	Tetkik	2025-05-15
9	9	H0009	D0009	9	280.00	Muayene	2025-05-16
10	10	H0010	D0010	10	320.00	Kontrol	2025-05-16
11	11	H0011	D0011	11	230.00	Muayene	2025-05-17
12	12	H0012	D0012	12	250.00	Tetkik	2025-05-17

Görsel 63 : “Faturalar” Tabosu Kontrolü

Tetikleyicimizin çalıştırılan sorgular sonrası gerekli işlemleri düzgün bir şekilde gerçekleştirip istenilen yerlerdeki güncellemeleri yaptığı görülebilir.