İÇİNDEKİLER

1. **Sistem/Gereksinim Analizi ve Kavramsal Modelin Oluşturulması**
2. **Varlık – İlişki (E-R) modelinin oluşumu ve tablolaştırılması**
3. **Tabloların Normalizasyonu**
   - **3.1. 'Ogrenci' tablosu için normalizasyon işlemleri**
   - **3.2. 'Akademisyen' tablosu için normalizasyon işlemleri**
   - **3.3. 'Calisan' tablosu için normalizasyon işlemleri**
   - **3.4. 'Rezervasyon' tablosu için normalizasyon işlemleri**
   - **3.5. 'Yemekler' tablosu için normalizasyon işlemleri**
   - **3.6. 'Menuler' tablosu için normalizasyon işlemleri**
   - **3.7. 'YemeklerMenuler' tablosu için normalizasyon işlemleri**
   - **3.8. 'Odemeler' tablosu için normalizasyon işlemleri**
   - **3.9. 'Geribildirimler' tablosu için normalizasyon işlemleri**
4. **İlişkisel Cebir Örnekleri**
5. **Veri tabanı ve tabloların oluşturulması**
   - **5.1. Veri tabanının oluşturulması**
   - **5.2. Tabloların oluşturulması**
     - **5.2.1. 'Ogrenci' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.2. 'Akademisyen' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.3. 'Calisan' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.4. 'Bolum' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.5. 'Rezervasyon' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.6. 'Menuler' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.7. 'Yemekler' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.8. 'YemeklerMenuler' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.9. 'Odemeler' tablosunun oluşturulması ve verilerin eklenmesi**
     - **5.2.10. 'Geribildirimler' tablosunun oluşturulması ve verilerin eklenmesi**
6. **Veri tabanı sorgu örnekleri (SQL/T-SQL)**
   - **6.1. Normal sorgu örnekleri**
   - **6.2. Saklı yordamlar (Stored procedures) örnekleri**
   - **6.3. Tetikleyici (Trigger) örneği**

1) Sistem/Gereksinim Analizi ve Kavramsal Modelin Oluşturulması

Bir üniversiteye ait yemekhanede otomasyon yazılımı kullanılacaktır. Bu otomasyon yazılımına gelen/giden verilerin saklanabilmesi için bir veri tabanı oluşturulacaktır. Veri tabanında; **öğrenciler, akademisyenler, çalışanlar**, **rezervasyonlar, yemekler**, **menüler**, **ödemeler**, **geribildirimler** hakkında veriler tutulacaktır. Aşağıda belirtilen kavramsal model, varlık-ilişki modelindeki ilişkilerin tablolara eklenmesiyle oluşması beklenen şemayı ifade etmektedir.

Her öğrenciye ait; **öğrenci numarası**, **adı** ve **soyadı**, **doğum tarihi**, **cinsiyeti**, **sınıfı**, **bölüm adı**, **bölüm numarası** ve **bina ismi**, **e-posta** ve **telefon numarası** bilgileri yer alacaktır. Her öğrenci, içinde bulunduğu tarih ve gelecek tarihler için sabah veya akşam menüsü satın alabilecektir, yapılan her satın alım için bir rezervasyon bilgisi oluşacaktır ve ardından ödemeler tablosunda da ödeme bilgisine dair kayıtlar tutulacaktır. Her öğrenci, yalnızca bir bölüme aittir.

Her akademisyenlere ait; **akademisyen numarası**, **adı** ve **soyadı**, **doğum tarihi**, **cinsiyeti**, **unvan**, **bölüm adı**, **bölüm numarası** ve **bina ismi**, **e-posta** ve **telefon numarası** bilgileri yer alacaktır. Her akademisyen, içinde bulunduğu tarih ve gelecek tarihler için sabah veya akşam menüsü satın alabilecektir, yapılan her satın alım için bir rezervasyon bilgisi oluşacaktır ve ardından ödemeler tablosunda da ödeme bilgisine dair kayıtlar tutulacaktır.

Her üniversite çalışanlarına ait; **çalışan numarası**, **adı** ve **soyadı**, **doğum tarihi**, **cinsiyeti**, **telefon numarası**, **bölüm adı**, **bölüm numarası**, **bina ismi** ve **maaş bilgisi** yer alacaktır. Her çalışan, içinde bulunduğu tarih ve gelecek tarihler için sabah veya akşam menüsü satın alabilecektir, yapılan her satın alım için bir rezervasyon bilgisi oluşacaktır ve ardından ödemeler tablosunda da ödeme bilgisine dair kayıtlar tutulacaktır.

Rezervasyonlar tablosunda her rezervasyona ait; **rezervasyon numarası**, **rezervasyon yapan kişinin öğrenci/akademisyen/personel numarası**, **menü numarası (ID)** bilgileri yer alacaktır. Her kişi, birden fazla rezervasyona sahip olabilir fâkat bir rezervasyon birden fazla kişiye ait olamaz.

Yemekler tablosunda her yemeğe ait; **yemeğin numarası**, **adı, türü** ve **kalori değeri** bulunacaktır.

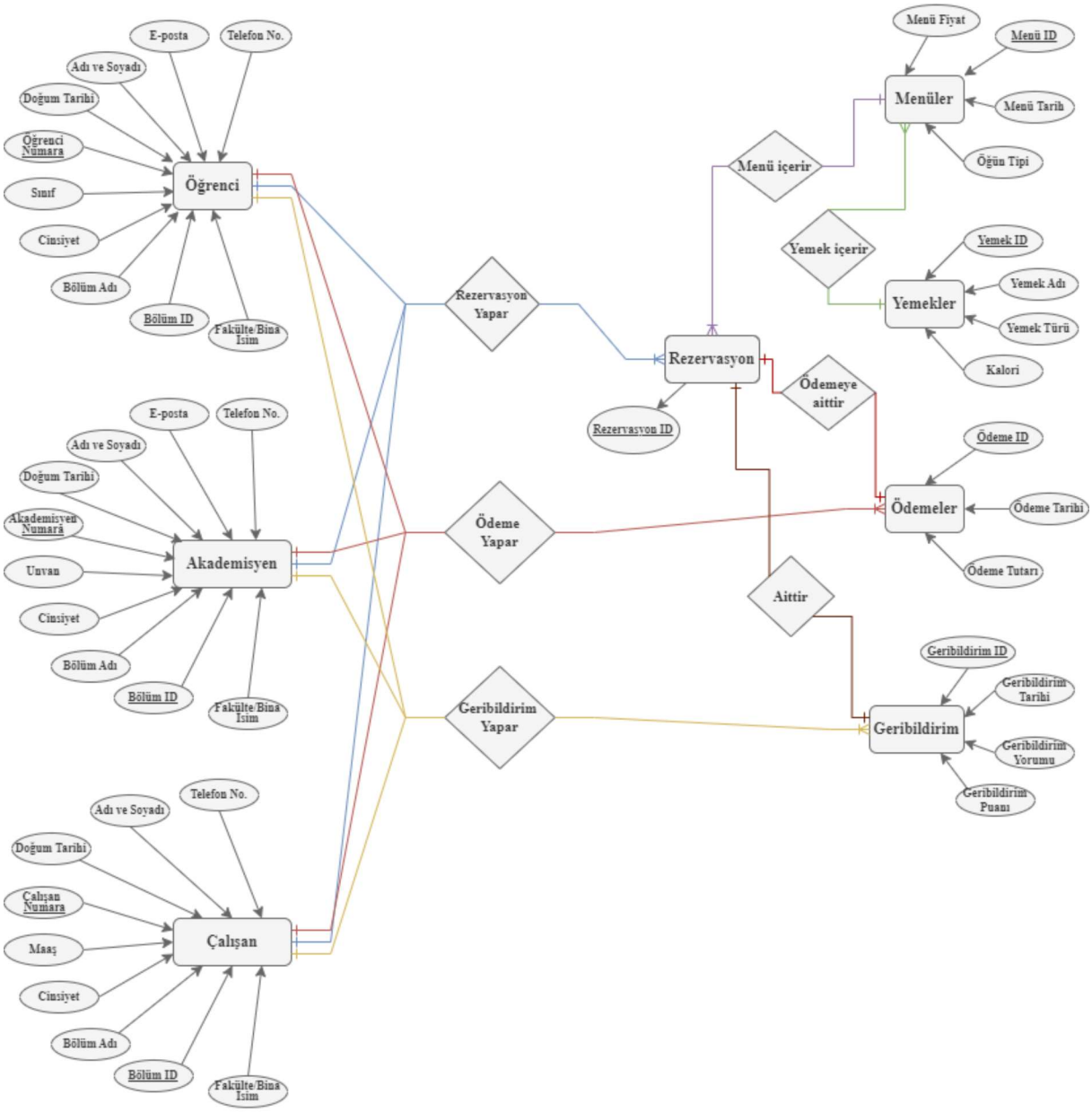
Menüler tablosunda, sabah, akşam ve öğlen menüleri bulunacaktır. Her menü; **menü numarası (ID)**, **menü fiyatı**, **menünün çıktığı tarih**, **öğün tipi (sabah/öğlen/akşam)** içerecektir.

Ödemeler tablosu herhangi bir menü için yapılan ödemeye ait; **ödeme numarası**, **ödemeyi yapan öğrencinin/personelin/çalışanın numarası**, **ödeme tarihi**, **ödeme yapılan rezervasyonun numarası**, **ödeme tutarı** bilgileri yer alacaktır.

Geribildirimler tablosunda; her geri bildirim için bir **geribildirim numarası**, **geribildirim yapan öğrenci/akademisyen/personelin numarası**, **geribildirimin yapılma tarihi**, **geribildirim yorumu**, **geribildirim puanı(0-10)**, **geribildirimin yapıldığı rezervasyonun numarası** bilgileri içermektedir.

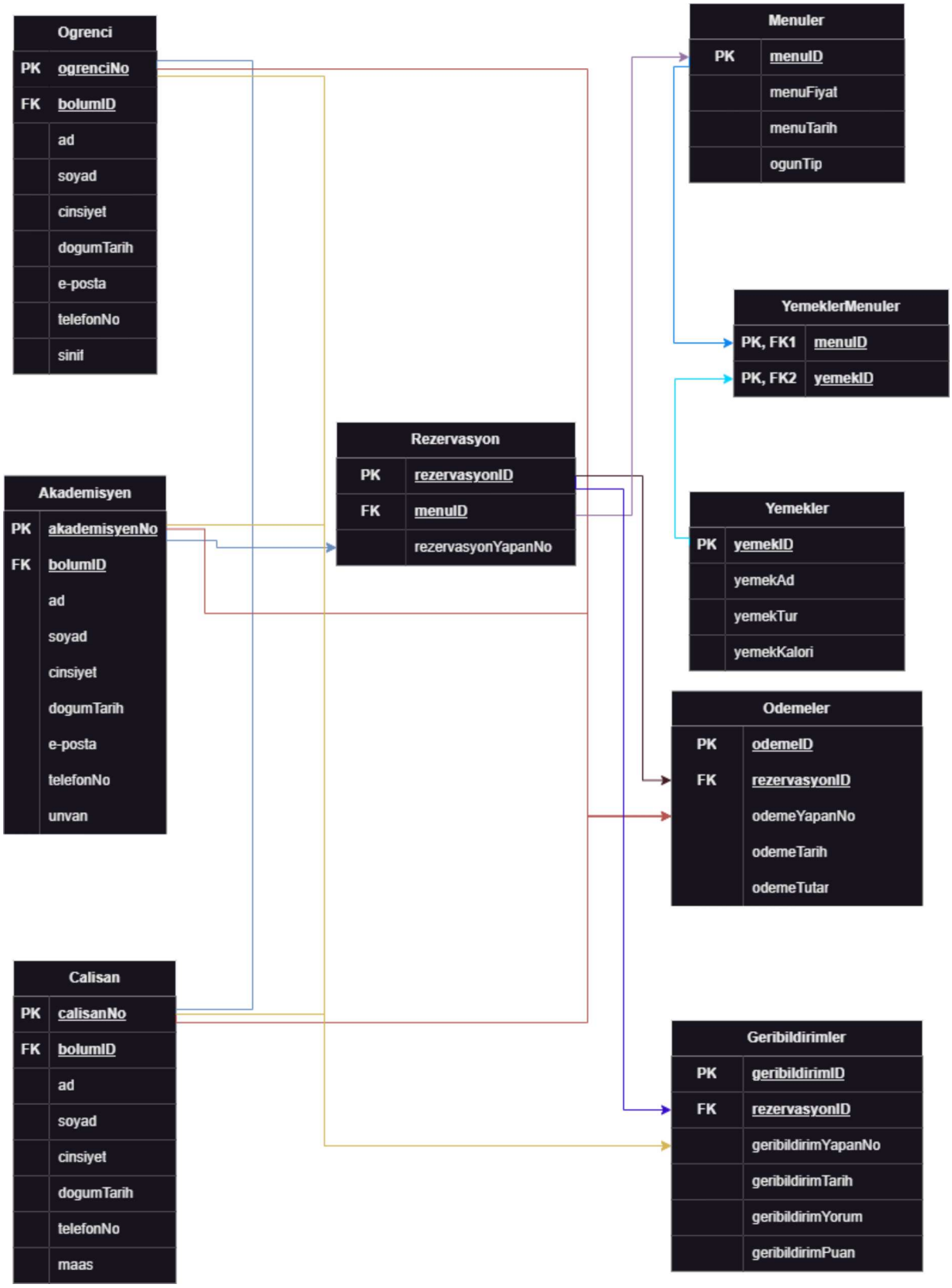
2) Varlık-İlişki (E-R) Modelinin Oluşturulması ve Tablolaştırılması

Kavramsal modelde belirtilen varlıklara, niteliklere ve ilişkilere uygun olarak modele ait varlık-ilişki (E-R) diyagramı aşağıdaki görselde tasarlanmıştır. Varlıklar arasındaki ilişkiler, 'Crow's foot notation' gösterimine uygun olarak tasarlanmıştır.



Şekil 1: Varlık-İlişki (E-R) diyagramı

Şekil 1'de gösterilen varlık-ilişki diyagramı, ilişkilerden doğan nitelikler, tablolar göz önüne alınarak aşağıda gösterildiği gibi tablolaştırılmıştır.



Şekil 2: Varlık-İlişki diyagramının tablolaştırılması

3) Tabloların Normalizasyonu

Tabloların normalizasyonu bu başlık altında anlatılmaktadır. Tablolardaki satır kayıtları temsili olarak kullanılmaktadır.

3.1) 'Ogrenci' tablosu için normalizasyon işlemleri

Öğrenci tablosunu oluşturup nitelikler arasındaki bağımlılıkları inceledikten sonra yapılması gereken normalizasyon işlemlerini inceleyelim.

Ogrenci									
ogrenciNo	ad ve soyad	cinsiyet	dogumTarih	e-posta	telefonNo	sinif	bolumID	bolumAd	binalsim
[redacted]01200	Ahmet [redacted]	E	2002-01-05	ahme[redacted]@gmail.com	53344[redacted]	2	100	Bilgisayar Mühendisliği	Teknoloji Fakültesi
[redacted]01201	Sena [redacted]	K	2003-02-07	sena.[redacted]l@gmail.com	52356[redacted]	2	304	Makina Mühendisliği	Teknoloji Fakültesi

Kısmi Fonksiyonel Bağımlılık

Kısmi Fonksiyonel Bağımlılık

Geçişli Bağımlılık

Şekil 3: Normalizasyona uğramamış öğrenci tablosu

UNF → 1N

'Ogrenci' tablosuna baktığımızda 'ad ve soyad' sütunu için **atomik olmamaya** yönelik yani her sütunda birden fazla kayıt olmasına yönelik bir anomali mevcuttur. Bu anomaliyi ortadan kaldırmak için belirtilen sütunu 'ad' ve 'soyad' olarak iki farklı sütuna ayırarak bu anomaliyi ortadan kaldırabiliriz. Farklı bir anomali olmadığı için tablomuzu 1N normal formuna dönüştürmüş olduk.

Ogrenci										
ogrenciNo	ad	soyad	cinsiyet	dogumTarih	e-posta	telefonNo	sinif	bolumID	bolumAd	binalsim
[redacted]1200	Ahmet	[redacted]	E	2002-01-05	ahme[redacted]@gmail.com	53344[redacted]	2	100	Bilgisayar Mühendisliği	Teknoloji Fakültesi
[redacted]1201	Sena	[redacted]	K	2003-02-07	sena.[redacted]@gmail.com	52356[redacted]	2	304	Makina Mühendisliği	Teknoloji Fakültesi

Şekil 4: 1N formuna dönüştürülmüş öğrenci tablosu

1N → 2N

‘Ogrenci’ tablomuzu 1N’den 2N formuna dönüştürebilmemiz için tablomuzdaki **kısmi bağımlılıkların (partial dependency)** kaldırılarak her sütunun kendisiyle ilgili bir anahtar sütunla tam bağımlı olmasını sağlamalıyız. Şekil 3’te gösterilen kısmi bağımlılıkları kaldırmak için ‘bolumID’, ‘bolumAd’, ‘binaIsim’ sütunlarını ‘Bolum’ adında farklı bir tabloya taşıyarak kısmi bağımlılıklardan kurtulabiliriz. Bu sayede öğrenciye ait bilgiler doğrudan ‘ogrenciNo’ sütununa tam bağımlı, bölüme ait bilgiler ise farklı bir tabloda doğrudan ‘bolumID’ sütununa tam bağımlı olacaktır. İki tablo arasındaki ilişkinin bozulmaması adına ‘Ogrenci’ tablomuzda, öğrencilerin bulunduğu bölüme ait ‘bolumID’si **yabancı anahtar (foreign key)** olarak bulundurulmalıdır.

Ogrenci								
ogrenciNo	ad	soyad	cinsiyet	dogumTarih	e-posta	telefonNo	sinif	bolumID
[redacted]01200	Ahmet	[redacted]	E	2002-01-05	ahme[redacted]@gmail.com	5334[redacted]	2	100
[redacted]01201	Sena	[redacted]	K	2003-02-07	sena.[redacted]@gmail.com	5235[redacted]	2	304

Bolum		
bolumID	bolumAd	binaIsim
100	Bilgisayar Mühendisliği	Teknoloji Fakültesi
304	Makina Mühendisliği	Teknoloji Fakültesi

Şekil 5: 2N ve ardından otomatik olarak 3N formuna dönüştürülmüş öğrenci tablosu

2N → 3N

Şekil 3’teki tabloda gösterilen **geçişli bağımlılıklar (transitive dependency)**, tabloyu 2N formuna dönüştürdüğümüz işlemler sonucunda ortadan kalkmıştır. Şekil 5’te görüldüğü üzere tablomuz 3N normal form hâlini almıştır.

3.2) ‘Akademisyen’ tablosu için normalizasyon işlemleri

Akademisyen tablosunu inceleyelim ve bağımlılıkları dikkate alarak yapılması gereken normalizasyon işlemlerini sırasıyla uygulayalım.

Akademisyen									
akademisyenNo	ad ve soyad	cinsiyet	dogumTarih	e-posta	telefonNo	unvan	bolumID	bolumAd	binalsim
[redacted]11111111	Kaan Ender	E	1978-02-24	ender@selcuk.edu.tr	512444[redacted]	Profesör	100	Bilgisayar Mühendisliği	Teknoloji Fakültesi
[redacted]2222222	Meliha Yazar	K	1984-01-12	yazar@selcuk.edu.tr	513222[redacted]	Doçent	304	Makine Mühendisliği	Teknoloji Fakültesi

Kısmi Fonksiyonel Bağımlılık

Kısmi Fonksiyonel Bağımlılık

Geçişli Bağımlılık

Şekil 6: Normalize olmayan akademisyen tablosu

UNF → 1N

‘Akademisyen’ tablosu incelendiğinde ‘ad’ ve ‘soyad’ olarak ayrılması gereken sütunların tek bir sütunda olduğunu görmekteyiz. Bu durum, **atomik olmamaya** yönelik bir anomalidir. Bu anomaliyi ortadan kaldırmak için şekil 6’daki ‘ad ve soyad’ sütununu ‘ad’ ve ‘soyad’ olarak iki farklı sütuna ayırmamız gerekmektedir. Aşağıda bu tablonun 1N formuna göre normalize edilmiş hâlini görmektesiniz.

Akademisyen										
akademisyenNo	ad	soyad	cinsiyet	dogumTarih	e-posta	telefonNo	unvan	bolumID	bolumAd	binalsim
[redacted]11111111	Kaan	Ender	E	1978-02-24	ender@selcuk.edu.tr	512444[redacted]	Profesör	100	Bilgisayar Mühendisliği	Teknoloji Fakültesi
[redacted]2222222	Meliha	Yazar	K	1984-01-12	yazar@selcuk.edu.tr	513222[redacted]	Doçent	304	Makine Mühendisliği	Teknoloji Fakültesi

Kısmi Fonksiyonel Bağımlılık

Kısmi Fonksiyonel Bağımlılık

Geçişli Bağımlılık

Şekil 7: 1N formuna dönüştürülmüş akademisyen tablosu

1N → 2N

‘Akademisyen’ tablosunu 1N’den 2N normal formuna dönüştürebilmemiz için **kısmi bağımlılıkları (partial dependency)** ortadan kaldırmamız ve her sütunun kendisiyle ilgili olan bir birincil anahtar sütuna tam bağımlı kılmamız gerekiyor. Bu ayırma işleminde ‘bolumAd’ ve ‘binaIsim’ sütunlarını ‘bolumID’ birincil anahtarına bağımlı kılabileceğimiz bir şekilde farklı tabloya aktarmamız gerekmektedir. Şekil 5’te bu işlemi daha önce yaptığımız için farklı bir tabloya ayırmadan sadece akademisyen tablosunda bulunması beklenen sütun değerlerini önceden ayırdığımız ‘Bolum’ tablosuna aktarmak yeterli olacaktır. ‘Bolum’ ve ‘Akademisyen’ tabloları arasındaki bağlantıyı sağlamak adına ‘Akademisyen’ tablosuna, ‘Bolum’ tablosunda bulunan ‘bolumID’ anahtarını **yabancı anahtar (foreign key)** olarak ekleyerek bağlantıyı sağlamış oluruz.

Akademisyen								
akademisyenNo	ad	soyad	cinsiyet	dogumTarih	e-posta	telefonNo	unvan	bolumID
[redacted]1111111	Kaan	Ender	E	1978-02-24	ender@selcuk.edu.tr	51244[redacted]	Profesör	100
[redacted]1111111	Meliha	Yazar	K	1984-01-12	yazar@selcuk.edu.tr	51322[redacted]	Doçent	304

Bolum		
bolumID	bolumAd	binaIsim
100	Bilgisayar Mühendisliği	Teknoloji Fakültesi
304	Makina Mühendisliği	Teknoloji Fakültesi

Şekil 8: 2N ve ardından otomatik olarak 3N formuna dönüştürülmüş akademisyen tablosu

2N → 3N

Şekil 6’daki tabloda gösterilen **geçişli bağımlılıklar (transitive dependency)**, tabloyu 2N formuna getirirken uyguladığımız işlemler sonucunda ortadan kalkmıştır. Şekil 8’de görüldüğü üzere tablomuz 3N normalizasyon formunu almıştır.

3.3) ‘Calisan’ tablosu için normalizasyon işlemleri

‘Calisan’ tablosunu inceleyerek yapılması gereken normalizasyon işlemlerini belirleyip uygulayalım.

Calisan								
calisanNo	ad ve soyad	cinsiyet	dogumTarih	maas	telefonNo	bolumID	bolumAd	binaIsim
881[redacted]12	Onur[redacted]	E	1987-05-14	17000	524[redacted]	240	Guvenlik	Teknoloji Fakültesi
881[redacted]94	[redacted] Kara	K	1980-02-15	24000	528[redacted]	210	Asci	Yemekhane

Kısmi Fonksiyonel Bağımlılık

Kısmi Fonksiyonel Bağımlılık

Geçişli Bağımlılık

Şekil 9: Normalize olmayan çalışan tablosu

UNF → 1N

‘Calisan’ tablosu incelendiğinde ‘ad ve soyad’ sütununda; boşlukla ayrılmış, sıralı birden fazla değere sahip kayıtlar olduğunu fark etmekteyiz. Bu durum, **atomik olmamaya** yönelik bir anomalidir. Bu anomaliyi düzeltmek için belirtilen sütunu, ‘ad’ ve ‘soyad’ olarak iki farklı sütuna bölebiliriz.

Calisan									
calisanNo	ad	soyad	cinsiyet	dogumTarih	maas	telefonNo	bolumID	bolumAd	binaIsim
881[redacted]2	Onur	[redacted]	E	1987-05-14	17000	524[redacted]	240	Guvenlik	Teknoloji Fakültesi
881[redacted]4	[redacted]	Kara	K	1980-02-15	24000	52[redacted]	210	Asci	Yemekhane

Kısmi Fonksiyonel Bağımlılık

Kısmi Fonksiyonel Bağımlılık

Geçişli Bağımlılık

Şekil 10: 1N formuna dönüştürülmüş çalışan tablosu

1N → 2N

1N'den 2N formuna geçiş yapabilmek için tablomuzdaki **kısmi bağımlılıkların (partial dependency)** kaldırılması gerekmektedir. Kısmi fonksiyonel bağımlılıklara sahip her sütunu yalnız tek bir birincil anahtara bağımlı kılmak için gerekli ayırma işlemlerini yapalım. Bu ayırma işleminde 'bolumAd' ve 'binaIsim' sütunlarını 'bolumID' birincil anahtarına bağımlı kılabileceğimiz bir şekilde farklı tabloya aktarmamız gerekmektedir. Yukarıdaki şekil 5'te hâlihazırda bu ayırma işlemini uyguladığımız için ayrılması gereken sütunları 'Bolum' tablosuna ekleyebiliriz. Ekleme işlemi yapıldıktan sonra 'Calisan' ile 'Bolum' tablosu arasındaki ilişkinin bozulmaması adına 'Bolum' tablosundaki 'bolumID' sütununun 'Calisan' tablosunda **yabancı anahtar (foreign key)** olarak bulundurulması gerekmektedir.

Calisan							
calisanNo	ad	soyad	cinsiyet	dogumTarih	maas	telefonNo	bolumID
881[redacted]12	Onur	[redacted]	E	1987-05-14	17000	524[redacted]	240
881[redacted]14	[redacted]	Kara	K	1980-02-15	24000	528[redacted]	210

Bolum		
bolumID	bolumAd	binaIsim
240	Guvenlik	Teknoloji Fakültesi
210	Asci	Yemekhane

Şekil 11: 2N ve ardından otomatik olarak 3N formuna dönüştürülmüş çalışan tablosu

2N → 3N

Şekil 9'daki tabloda gösterilen **geçişli bağımlılıklar (transitive dependency)**, tabloyu 2N formuna dönüştürdüğümüz işlemler sonucunda ortadan kalkmıştır. Şekil 11'de görüldüğü üzere tablomuz 3N normal form hâlini almıştır.

3.4) ‘Rezervasyon’ tablosu için normalizasyon işlemleri

‘Rezervasyon’ tablomuzda ‘rezervasyonID’ sütunu tablo için birincil anahtar olmak üzere ‘menuID’, ‘Menuler’ tablosundaki ‘menuID’ sütunu için bir **yabancı anahtar (foreign key)** oluşturmaktadır. ‘Rezervasyon’ tablosunu normalizasyon için incelediğimizde mevcut tablonun 1N, 2N ve 3N normal formlarına uyduğunu görmekteyiz. Bu sebepten dolayı tabloda herhangi bir normalizasyon işlemi uygulanmamaktadır.

Rezervasyon		
rezervasyonID	menuID	rezervasyonYapanNo
555	100	[redacted]1200
444	101	[redacted]1111111

Tam Fonksiyonel Bağımlılık

Şekil 12: 1N, 2N ve 3N normal formunda bulunan rezervasyon tablosu

3.5) ‘Yemekler’ tablosu için normalizasyon işlemleri

‘Yemekler’ tablomuzda ‘yemekID’ sütunu tablo için birincil anahtar olmak üzere ‘yemekAd’, ‘yemekTur’ ve ‘kalori’ sütunları ise birincil anahtara bağlı niteliklerdir. Normalizasyon uygulaması için tabloyu incelediğimizde hâlihazırda tablonun 1N, 2N ve 3N normal formlarına uyduğunu görmekteyiz. Bu sebepten dolayı tabloda herhangi bir normalizasyon işlemi uygulanmamaktadır.

Yemekler			
yemekID	yemekAd	yemekTur	kalori
444	İmambayıldı	Sebze	164
333	Trileçe	Tatlı	301

Tam Fonksiyonel Bağımlılık

Şekil 13: 1N, 2N ve 3N normal formunda bulunan yemekler tablosu

3.6) ‘Menuler’ tablosu için normalizasyon işlemleri

‘Menuler’ tablomuzda ‘menuID’ sütunu tablomuz için birincil anahtar olmaktadır. Tabloda uygulanacak normalizasyon işlemlerini inceleyecek olursak tablonun hâlihazırda 1N, 2N ve 3N normal formlarına uygun olduğunu görmekteyiz. Bu sebeple tabloda herhangi bir normalizasyon işlemi uygulanmamaktadır.

Menuler			
menuID	menuFiyat	menuTarih	ogunTip
150	40	2024-05-10	Sabah
151	30	2024-05-10	Öğlen
152	25	2024-05-10	Akşam

Tam Fonksiyonel Bağımlılık

Şekil 14: 1N, 2N ve 3N normal formunda bulunan menüler tablosu

3.7) ‘YemeklerMenuler’ tablosu için normalizasyon işlemleri

‘YemeklerMenuler’ tablomuzda ‘menuID’ ve ‘yemekID’ sütunları, tablomuz için **birleşik birincil (composite key) anahtar** olmaktadır. Bu tablo, ‘Yemekler’ tablosu ile ‘Menüler’ tablosu arasındaki **M-N (Çoka – çok) ilişki** sonucu oluşmuştur. Tabloda uygulanacak normalizasyon işlemlerini inceleyecek olursak tablonun hâlihazırda 1N, 2N ve 3N normal formlarına uygun olduğunu görmekteyiz. Bu sebeple tabloda herhangi bir normalizasyon işlemi uygulanmamaktadır.

YemeklerMenuler	
menuID	yemekID
150	40
151	30
152	25

Tam Fonksiyonel Bağımlılık

Şekil 15: 1N, 2N ve 3N normal formunda bulunan yemekler-menüler tablosu

3.8) ‘Odemeler’ tablosu için normalizasyon işlemleri

‘Odemeler’ tablomuzda ‘odemeID’ sütunu tablomuz için birincil anahtar olmak üzere ‘rezervasyonID’ sütunu ise ‘Rezervasyonlar’ tablosunda bulunan ‘rezervasyonID’ için bir **yabancı anahtar (foreign key)** oluşturmaktadır. Tablodaki her sütunun birincil anahtara tam fonksiyonel bağımlı olduğunu görmekteyiz. Tablo, halihazırda 1N, 2N ve 3N normal formlarına uygun olduğu için herhangi bir normalizasyon işlemi uygulanmamaktadır.

Odemeler				
odemeID	odemeYapanNo	odemeTarih	odemeTutar	rezervasyonID
777	[redacted]1200	2024-05-11	40	444
888	[redacted]111111	2024-05-12	30	555
999	881[redacted]	2024-05-13	25	666

Tam Fonksiyonel Bağımlılık

Şekil 16: 1N, 2N ve 3N normal formunda bulunan ödemeler tablosu

3.9) ‘Geribildirimler’ tablosu için normalizasyon işlemleri

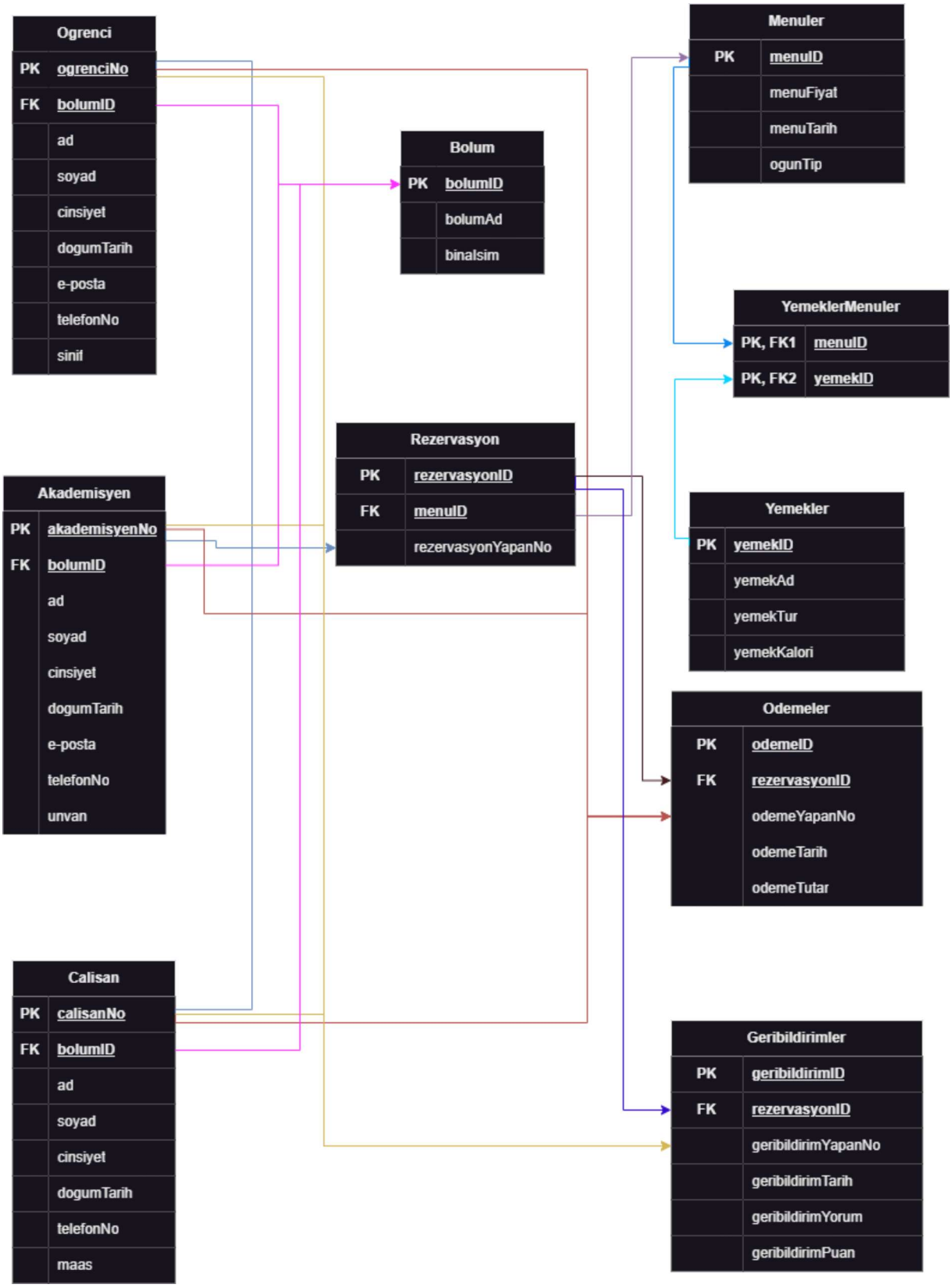
‘Geribildirimler’ tablosunda ‘geribildirimID’ sütunu tablo için birincil anahtar sütun olmakla beraber ‘rezervasyonID’ sütunu, ‘Rezervasyonlar’ tablosunda bulunan ‘rezervasyonID’ sütunu için bir **yabancı anahtar (foreign key)** oluşturmaktadır. Tabloyu normalizasyon için incelediğimiz takdirde tablonun 1N, 2N ve 3N formlarına uygun olduğunu görmekteyiz. Bu nedenle tabloda herhangi bir normalizasyon işlemi uygulanmamaktadır.

Geribildirimler					
geribildirimID	geribildirimYapanNo	geribildirimTarih	geribildirimYorum	geribildirimPuan	rezervasyonID
3333	[redacted]1200	2024-05-11	"Yemekler çok iyiydi!"	5	444
4444	[redacted]1111111	2024-05-12	"Çok beğenmedim."	3	555
5555	881[redacted]	2024-05-13	"Tatlıdan kıl çıktı!"	1	666

Tam Fonksiyonel Bağımlılık

Şekil 17: 1N, 2N ve 3N normal formunda bulunan geribildirimler tablosu

Normalizasyon işlemleri 3N formuna kadar tamamlanmış olan tabloların son tablo şeması gösterimi aşağıdaki şekildeki gibidir.



Şekil 18: 3N normal formuna ulaşan tabloların son şematik gösterimi

4) İlişkisel Cebir Örnekleri

Aşağıdaki ilişkisel cebir sorgusu, geribildirim puanı 5'ten büyük olan ve '22-05-2024' tarihinde gönderilen geribildirimlerin bilgilerini sorgular.

$\prod_{\text{geribildirimTarih, geribildirimPuan}} (\sigma_{\text{geribildirimPuan} > 5 \wedge \text{geribildirimTarih} = \text{'22-05-2024'}} (\text{Geribildirimler}))$

Aşağıdaki ilişkisel cebir sorgusu, '[redacted]001' numaralı öğrencinin yaptığı geribildirime ait yorum, puan ve ilgili menüye ödediği tutar bilgisini sorgular.

$\prod_{\text{geribildirimYorum, geribildirimPuan, odemeTutar}} (\sigma_{\text{rezervasyonYapanNo} = \text{"223301001"}}$

$(\text{Geribildirimler} \bowtie_{\text{geribildirimler.rezervasyonID} = \text{odemeler.rezervasyonID}} \text{Odemeler}))$

Aşağıdaki ilişkisel cebir sorgusu; 'Yemekler' tablosundaki her yemek çeşidinin 500 kalori üstündeki yemeklerinden kaç adet olduğunu hesaplar.

$\prod_{\text{yemekAd, yemekTur, yemekKalori}} (\sigma_{\text{yemekKalori} > 500} (\zeta_{\text{COUNT yemekTur}} (\text{Yemekler})))$

Aşağıdaki ilişkisel cebir sorgusu ise her menüde bulunan yemeklerin toplam kalorilerini hesaplar.

$\prod_{\text{menuID, yemekKalori}} (\sigma (\text{Yemekler} \bowtie \text{Menuler} (\bowtie \text{YemeklerMenuler}))$

$\zeta_{\text{SUM yemekKalori}} (\text{Yemekler}))$

5) Veri tabanı ve tabloların Oluşturulması

5.1) Veri tabanının oluşturulması

Görselde olduğu gibi **CREATE** komutu ile **'dbYemekhane'** isminde bir veri tabanı oluşturuyoruz. Ardından **'USE dbYemekhane'** ve **'GO'** komutları ile veri tabanını seçerek sonraki işlemler için kullanılmaya hazır hâle getiriyoruz.

```
CREATE DATABASE dbYemekhane
USE dbYemekhane
GO
```

Şekil 19: Veri tabanının oluşturulmasıyla ilgili SQL komutları

5.2) Tabloların oluşturulması

5.2.1) 'Ogrenci' tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile 'Ogrenci' tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

ogrenciNo niteliği: Birincil anahtar olarak tanımlanan nitelik 9 adet karakterden oluşuyor.

bolumID niteliği: Yabancıl anahtar (Bolum(bolumID) referansı) olarak tanımlanan nitelik 4 adet karakterden oluşuyor. Bu nitelik, sahip olduğu kısıtlamayı **'Bolum' tablosunun 'bolumID'** sütunundan **miras alıyor**.

ad niteliği: Maksimum 30 tane karakterden oluşuyor.

soyad niteliği: Maksimum 30 tane karakterden oluşuyor.

cinsiyet niteliği: Yalnızca 'E' ve 'K' karakterlerini içerebilir, tek karakterden oluşuyor. Değer girilmediğinde varsayılan olarak '?' değerini alıyor.

dogumTarih niteliği: 'YYYY-MM-DD' tarih formatından veri girişini kabul ediyor. Değer girilmediğinde varsayılan olarak '1970-01-01' UNIX Epoch değerine sahip oluyor.

eposta niteliği: Maksimum 50 tane karakterden oluşuyor. Değer girilmediğinde 'example@mail.com' değerine sahip oluyor.

telefonNo niteliği: Yalnızca 10 karakterden oluşuyor. Değer girilmediğinde '5555555555' varsayılan değerine sahip oluyor.

sinif niteliği: Yalnızca '1', '2', '3' ve '4' karakterlerini içerebilir, tek karakterden oluşuyor.

```sql
CREATE TABLE Ogrenci (
    ogrenciNo       char(9) NOT NULL PRIMARY KEY,
    bolumID         char(4) FOREIGN KEY REFERENCES Bolum(bolumID),
    ad              varchar(30) NOT NULL DEFAULT '',
    soyad           varchar(30) NOT NULL DEFAULT '',
    cinsiyet        char(1) NOT NULL DEFAULT '?', CHECK (cinsiyet IN ('E', 'K')),
    dogumTarih      DATE NOT NULL DEFAULT '1970-01-01',
    eposta          varchar(50) NOT NULL DEFAULT 'example@mail.com',
    telefonNo       char(10) NOT NULL DEFAULT '5555555555',
    sinif           char(1) NOT NULL DEFAULT '0', CHECK (sinif IN ('1', '2', '3', '4'))
)
```

Şekil 20: 'Ogrenci' tablosunun oluşturulmasına dair SQL komutları

```sql
INSERT INTO Ogrenci(ogrenciNo, bolumID, ad, soyad, cinsiyet, dogumTarih, eposta, telefonNo, sinif)
VALUES
([redacted]01001', 'B001', 'Ahmet', [redacted], 'E', '2002-01-04', 'ahmet[redacted]@ogr.selcuk.edu.tr', '542[redacted]', '2'),
([redacted]10001', 'B010', 'Derya', [redacted] 'K', '2002-04-06', 'd[redacted]@ogr.selcuk.edu.tr', '543[redacted]', '3'),
([redacted]14001', 'B014', 'Enes', [redacted] 'E', '2003-06-14', 'e[redacted]@ogr.selcuk.edu.tr', '5343[redacted]', '4'),
([redacted]05001', 'B005', 'Müzeyyen', [redacted], 'K', '2005-08-14', 'm[redacted]@ogr.selcuk.edu.tr', '53[redacted]', '1'),
([redacted]03001', 'B003', 'Meryem', [redacted], 'K', '2001-12-05', 'meryem[redacted]@ogr.selcuk.edu.tr', '528[redacted]', '4'),
([redacted]01002', 'B001', 'Şükrü', [redacted] 'E', '2003-06-14', 'sukru[redacted]@ogr.selcuk.edu.tr', '556[redacted]', '2'),
([redacted]02001', 'B002', 'Kamile', [redacted] 'K', '2002-11-24', 'kamil[redacted]@ogr.selcuk.edu.tr', '554[redacted]', '3'),
([redacted]11001', 'B011', 'Aslı', [redacted] 'K', '2001-07-01', 'asl[redacted]ogr.selcuk.edu.tr', '52415[redacted]', '4'),
([redacted]11002', 'B011', 'Ahmet', [redacted], 'E', '2004-01-05', 'ahmet[redacted]@ogr.selcuk.edu.tr', '574[redacted]', '1'),
([redacted]01003', 'B001', 'Sıla', [redacted], 'K', '2004-08-15', 'sila[redacted]@ogr.selcuk.edu.tr', '512[redacted]', '1'),
([redacted]05002', 'B005', 'Anıl', [redacted], 'E', '2002-02-03', 'anil[redacted]@ogr.selcuk.edu.tr', '51[redacted]', '3'),
([redacted]03002', 'B003', 'Sema', [redacted], 'K', '2004-03-08', 'sema[redacted]ogr.selcuk.edu.tr', '566[redacted]', '1'),
([redacted]14002', 'B014', 'Ayşe', [redacted] K', '2003-02-14', 'ays[redacted]@ogr.selcuk.edu.tr', '5864[redacted]', '3'),
([redacted]08001', 'B008', 'Alparslan', [redacted], 'E', '2004-06-15', 'alp[redacted]ogr.selcuk.edu.tr', '56[redacted]', '2'),
([redacted]08002', 'B008', 'İlayda', [redacted], 'K', '2003-12-24', 'ilayda[redacted]ogr.selcuk.edu.tr', '[redacted]', '3'),
([redacted]15001', 'B015', 'Acun', [redacted], 'E', '2004-01-04', 'acun[redacted]ogr.selcuk.edu.tr', '571[redacted]', '1'),
([redacted]15002', 'B015', 'Müjgan', [redacted], 'K', '2003-09-17', 'mujgan[redacted]@ogr.selcuk.edu.tr', '58[redacted]', '2'),
([redacted]16001', 'B016', 'Işıl', [redacted] 'K', '2001-05-23', 'isil[redacted]ogr.selcuk.edu.tr', '554[redacted]', '4'),
([redacted]17001', 'B017', 'Onur', [redacted], 'E', '2004-10-21', 'onur[redacted]@ogr.selcuk.edu.tr', '512[redacted]', '1'),
([redacted]17002', 'B017', 'Gamze', [redacted] 'K', '2003-09-14', 'gamz[redacted]ogr.selcuk.edu.tr', '58[redacted]', '3'),
([redacted]18001', 'B018', 'Bayhan', [redacted], 'E', '2004-11-15', 'bayhan[redacted]@ogr.selcuk.edu.tr', '5[redacted]', '2'),
([redacted]18002', 'B018', 'Gül', [redacted], 'K', '2003-09-01', 'gul[redacted]@ogr.selcuk.edu.tr', '5[redacted]', '3'),
([redacted]04001', 'B004', 'Mehmet', [redacted], 'E', '2000-07-18', 'mehmet[redacted]ogr.selcuk.edu.tr', '54[redacted]', '4'),
([redacted]04002', 'B004', 'Metin', [redacted], 'E', '2003-02-14', 'meti[redacted]@ogr.selcuk.edu.tr', '521[redacted]', '3'),
([redacted]04003', 'B004', 'Sude', [redacted], 'K', '2004-08-13', 'sud[redacted]ogr.selcuk.edu.tr', '56[redacted]', '2'),
([redacted]19001', 'B019', 'Aysima', [redacted] 'K', '2003-09-11', 'aysima[redacted]ogr.selcuk.edu.tr', '56[redacted]', '3'),
([redacted]19002', 'B019', 'Mine', [redacted] K', '2004-06-12', 'min[redacted]ogr.selcuk.edu.tr', '546[redacted]', '1'),
([redacted]20001', 'B020', 'Jale', [redacted] K', '2005-01-04', 'jale[redacted]@ogr.selcuk.edu.tr', '51[redacted]', '1'),
([redacted]20002', 'B020', 'Mahmut', [redacted] 'E', '2001-04-12', 'mahmut[redacted]ogr.selcuk.edu.tr', '524[redacted]', '4'),
([redacted]20003', 'B020', 'Çiğdem', [redacted], 'K', '2004-11-03', 'cigdem[redacted]@ogr.selcuk.edu.tr', '526[redacted]', '1');
```

Şekil 21: 'Ogrenci' tablosuna örnek verilerin eklenmesi

5.2.2) 'Akademisyen' tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile 'Akademisyen' tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

akademisyenNo niteliği: Birincil anahtar olarak tanımlanan nitelik 9 adet karakterden oluşuyor.

bolumID niteliği: Yabancıl anahtar (Bolum(bolumID) referansı) olarak tanımlanan nitelik 4 adet karakterden oluşuyor. Bu nitelik, sahip olduğu kısıtlamayı **'Bolum' tablosunun 'bolumID'** sütunundan **miras alıyor**.

ad niteliği: Maksimum 30 tane karakterden oluşuyor.

soyad niteliği: Maksimum 30 tane karakterden oluşuyor.

cinsiyet niteliği: Yalnızca ‘E’ ve ‘K’ karakterlerini içerebilir, tek karakterden oluşuyor. Değer girilmediğinde varsayılan olarak ‘?’ değerini alıyor.

dogumTarih niteliği: ‘YYYY-MM-DD’ tarih formatından veri girişini kabul ediyor. Değer girilmediğinde varsayılan olarak ‘1970-01-01’ UNIX Epoch değerine sahip oluyor.

eposta niteliği: Maksimum 50 tane karakterden oluşuyor. Değer girilmediğinde ‘example@mail.com’ değerine sahip oluyor.

telefonNo niteliği: Yalnızca 10 karakterden oluşuyor. Değer girilmediğinde ‘5555555555’ varsayılan değerine sahip oluyor.

unvan niteliği: Yalnızca ‘Profesör’, ‘Doçent Doktor’, ‘Doktor’, ‘Öğretim Görevlisi’ ve ‘Araştırma Görevlisi’ değerlerini içerebilir, maksimum 20 karakterden oluşuyor.

```
CREATE TABLE Akademisyen (
    akademisyenNo   char(9) NOT NULL PRIMARY KEY,
    bolumID         char(4) FOREIGN KEY REFERENCES Bolum(bolumID),
    ad              varchar(30) NOT NULL DEFAULT '',
    soyad           varchar(30) NOT NULL DEFAULT '',
    cinsiyet        char(1) NOT NULL DEFAULT '?', CHECK (cinsiyet IN ('E', 'K')),
    dogumTarih      DATE NOT NULL DEFAULT '1970-01-01',
    eposta          varchar(50) NOT NULL DEFAULT 'example@mail.com',
    telefonNo       char(10) NOT NULL DEFAULT '5555555555',
    unvan           varchar(20) NOT NULL DEFAULT '',
    CHECK (unvan IN ('Profesör', 'Doçent Doktor', 'Doktor', 'Öğretim Görevlisi', 'Araştırma Görevlisi'))
)
```

Şekil 22: ‘Akademisyen’ tablosunun oluşturulmasına dair SQL komutları

```
INSERT INTO Akademisyen(akademisyenNo, bolumID, ad, soyad, cinsiyet, dogumTarih, eposta, telefonNo, unvan)
VALUES
('993[redacted]01', 'B001', [redacted] 'Ünal', 'E', '1985-11-05', 'ilker[redacted]l@selcuk.edu.tr', '51[redacted]', 'Doçent Doktor'),
('993[redacted]01', 'B002', [redacted] 'Arkın', 'K', '1974-01-08', 'deni[redacted]kin@selcuk.edu.tr', '[redacted]45', 'Profesör'),
('993[redacted]01', 'B003', [redacted] 'Kara', 'K', '1985-06-12', 'emine[redacted]a@selcuk.edu.tr', '58[redacted]', 'Öğretim Görevlisi'),
('993[redacted]01', 'B004', [redacted] 'Yavuz', 'K', '1992-01-15', 'senay[redacted]z@selcuk.edu.tr', '51[redacted]', 'Araştırma Görevlisi'),
('993[redacted]01', 'B005', [redacted] 'Kuşku', 'E', '1978-01-02', 'kami[redacted]@selcuk.edu.tr', '[redacted]12', 'Profesör'),
('993[redacted]01', 'B008', [redacted] 'Kıraat', 'E', '1991-09-22', 'suk[redacted]at@selcuk.edu.tr', '[redacted]', 'Araştırma Görevlisi'),
('993[redacted]01', 'B010', [redacted] 'İpekoğlu', 'K', '1988-11-23', 'nil[redacted]@selcuk.edu.tr', '5812[redacted]', 'Doktor'),
('993[redacted]01', 'B011', [redacted] 'Karadayı', 'E', '1993-01-15', 'an[redacted]yi@selcuk.edu.tr', '58[redacted]1', 'Öğretim Görevlisi'),
('993[redacted]01', 'B014', [redacted] 'Bahadır', 'E', '1974-02-12', 'ah[redacted]ir@selcuk.edu.tr', '54[redacted]1', 'Profesör'),
('993[redacted]01', 'B015', [redacted] 'Bilginer', 'K', '1994-07-24', 'n[redacted]iner@selcuk.edu.tr', '[redacted]512', 'Araştırma Görevlisi'),
('993[redacted]01', 'B016', [redacted] 'İnanır', 'E', '1979-04-14', 'kem[redacted]@selcuk.edu.tr', '5746[redacted]', 'Doçent Doktor'),
('993[redacted]01', 'B017', [redacted] 'Karaman', 'K', '1984-02-12', 'sem[redacted]@selcuk.edu.tr', '51246[redacted]', 'Öğretim Görevlisi'),
('993[redacted]01', 'B018', [redacted] 'Baloğlu', 'K', '1992-01-17', '[redacted]aloglu@selcuk.edu.tr', '[redacted]259231', 'Doktor'),
('993[redacted]01', 'B019', [redacted] 'Çolak', 'E', '1971-11-14', 'yusu[redacted]@selcuk.edu.tr', '5262849[redacted]', 'Profesör'),
('993[redacted]01', 'B020', [redacted] 'Aloğlu', 'K', '1994-02-13', 'irem[redacted]@selcuk.edu.tr', '5859712[redacted]', 'Araştırma Görevlisi');
```

Şekil 23: ‘Akademisyen’ tablosuna örnek verilerin eklenmesi

5.2.3) ‘Calisan’ tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile ‘Calisan’ tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

calisanNo niteliği: Birincil anahtar olarak tanımlanan nitelik 9 adet karakterden oluşuyor.

bolumID niteliği: Yabancıl anahtar (Bolum(bolumID) referansı) olarak tanımlanan nitelik 4 adet karakterden oluşuyor. Bu nitelik, sahip olduğu kısıtlamayı **‘Bolum’ tablosunun ‘bolumID’** sütunundan **miras alıyor**.

ad niteliği: Maksimum 30 tane karakterden oluşuyor.

soyad niteliği: Maksimum 30 tane karakterden oluşuyor.

cinsiyet niteliği: Yalnızca ‘E’ ve ‘K’ karakterlerini içerebilir, tek karakterden oluşuyor. Değer girilmediğinde varsayılan olarak ‘?’ değerini alıyor.

dogumTarih niteliği: ‘YYYY-MM-DD’ tarih formatından veri girişini kabul ediyor. Değer girilmediğinde varsayılan olarak ‘1970-01-01’ UNIX Epoch değerine sahip oluyor.

telefonNo niteliği: Yalnızca 10 karakterden oluşuyor. Değer girilmediğinde ‘5555555555’ varsayılan değerine sahip oluyor.

maas niteliği: ‘money’ tipinde 0’dan büyük sayısal değer alıyor. Değer girilmediğinde varsayılan olarak ‘0’ değerini alıyor.

```
CREATE TABLE Calisan (
    calisanNo       char(9) NOT NULL PRIMARY KEY,
    bolumID         char(4) FOREIGN KEY REFERENCES Bolum(bolumID),
    ad              varchar(30) NOT NULL DEFAULT '',
    soyad           varchar(30) NOT NULL DEFAULT '',
    cinsiyet        char(1) NOT NULL DEFAULT '?', CHECK (cinsiyet IN ('E', 'K')),
    dogumTarih      DATE NOT NULL DEFAULT '1970-01-01',
    telefonNo       char(10) NOT NULL DEFAULT '5555555555',
    maas            money DEFAULT '0', CHECK (maas > 0)
)
```

Şekil 24: ‘Calisan’ tablosunun oluşturulmasına dair SQL komutları

```
INSERT INTO Calisan(calisanNo, bolumID, ad, soyad, cinsiyet, dogumTarih, telefonNo, maas)
VALUES
('8[redacted]01', 'B100', [redacted] 'Köksoy', 'E', '1985-01-07', '[redacted]2841', 24000),
('8[redacted]01', 'B101', [redacted] 'Hepkaya', 'K', '1989-01-16', [redacted]123', 24000),
('8[redacted]02', 'B101', [redacted] 'Cümbüş', 'E', '1990-07-23', '5[redacted]31245', 24000),
('8[redacted]01', 'B102', [redacted] 'Karaoğlan', 'K', '1993-02-17', '51[redacted]20', 24000),
('8[redacted]02', 'B102', [redacted] 'Kavuk', 'E', '1986-11-25', '58[redacted]32561', 24000),
('8[redacted]01', 'B108', [redacted] 'Aslanoğlu', 'E', '1970-02-03', '51[redacted]02', 55000),
('8[redacted]01', 'B107', [redacted] 'Eser', 'K', '1982-04-13', '591[redacted]23', 30000),
('8[redacted]02', 'B107', [redacted] 'İnanır', 'E', '1988-04-19', '514[redacted]01', 30000),
('8[redacted]01', 'B105', [redacted] 'Çırakoğlu', 'E', '1986-04-15', '[redacted]96878', 20000),
('8[redacted]02', 'B105', [redacted] 'Çetin', 'K', '1979-02-23', '587[redacted]4', 20000),
('8[redacted]03', 'B105', [redacted] 'Çalışkan', 'K', '1985-01-11', '59[redacted]02', 20000),
('8[redacted]01', 'B106', [redacted] 'Erdem', 'K', '1981-06-17', '5[redacted]02', 28000),
('8[redacted]02', 'B106', [redacted] Bolivar', 'K', '1994-08-24', '5[redacted]39192', 28000),
('8[redacted]03', 'B106', [redacted] 'Sadık', 'E', '1987-11-02', '50[redacted]', 28000),
('8[redacted]04', 'B106', [redacted] 'Kadıoğlan', 'K', '1983-09-12', '562[redacted]', 28000);
```

Şekil 25: 'Calisan' tablosuna örnek verilerin eklenmesi

5.2.4) 'Bolum' tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile 'Bolum' tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

bolumID niteliği: Birincil anahtar olarak tanımlanan nitelik 4 adet karakterden oluşuyor. Baş harfi 'B' ve kalan üç karakter ise 0 ile 9 arasındaki bir sayısal karakter olması için ise uygun bir RegEx ifadesi kullanılıyor.

bolumAd niteliği: Maksimum 50 karakterden oluşuyor.

binaIsim niteliği: Maksimum 50 tane karakterden oluşuyor.

```
CREATE TABLE Bolum (
    bolumID         char(4) NOT NULL PRIMARY KEY, CHECK (bolumID LIKE 'B[0-9][0-9][0-9]'),
    bolumAd         varchar(50) NOT NULL DEFAULT '',
    binaIsim        varchar(50) NOT NULL DEFAULT '',
)
```

Şekil 26: 'Bolum' tablosunun oluşturulmasına dair SQL komutları

```sql
INSERT INTO Bolum(bolumID, bolumAd, binaIsim)
VALUES
('B001', 'Bilgisayar Mühendisliği', 'Teknoloji Fakültesi'),
('B002', 'Biyomedikal Mühendisliği', 'Teknoloji Fakültesi'),
('B003', 'Elektrik-Elektronik Mühendisliği', 'Teknoloji Fakültesi'),
('B004', 'Makine Mühendisliği', 'Teknoloji Fakültesi'),
('B005', 'Mekatronik Mühendisliği', 'Teknoloji Fakültesi'),
('B006', 'Metalurji ve Malzeme Mühendisliği', 'Teknoloji Fakültesi'),
('B007', 'Eczacılık', 'Eczacılık Fakültesi'),
('B008', 'Diş Hekimliği', 'Diş Hekimliği Fakültesi'),
('B009', 'Biyoloji', 'Fen Fakültesi'),
('B010', 'Fizik', 'Fen Fakültesi'),
('B011', 'Kimya', 'Fen Fakültesi'),
('B012', 'Hukuk', 'Hukuk Fakültesi'),
('B013', 'İktisat', 'İktisadi ve İdari Bilimler Fakültesi'),
('B014', 'İşletme', 'İktisadi ve İdari Bilimler Fakültesi'),
('B015', 'Uluslararası İlişkiler', 'İktisadi ve İdari Bilimler Fakültesi'),
('B016', 'Gazetecilik', 'İletişim Fakültesi'),
('B017', 'Reklamcılık', 'İletişim Fakültesi'),
('B018', 'İç Mimarlık', 'Mimarlık ve Tasarım Fakültesi'),
('B019', 'Felsefe', 'Edebiyat Fakültesi'),
('B020', 'Sosyoloji', 'Edebiyat Fakültesi'),
('B100', 'Güvenlik', 'Teknoloji Fakültesi'),
('B101', 'Güvenlik', 'Edebiyat Fakültesi'),
('B102', 'Güvenlik', 'İktisadi ve İdari Bilimler Fakültesi'),
('B103', 'Güvenlik', 'Fen Fakültesi'),
('B104', 'Temizlik', 'Teknoloji Fakültesi'),
('B105', 'Temizlik', 'Yemekhane'),
('B106', 'Mutfak', 'Yemekhane'),
('B107', 'Depo', 'Yemekhane'),
('B108', 'Yönetim', 'Yemekhane'),
('B109', 'Muhasebe', 'Yemekhane');
```

Şekil 27: 'Bolum' tablosuna örnek verilerin eklenmesi

5.2.5) ‘Rezervasyon’ tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile ‘Bolum’ tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

rezervasyonID niteliği: Birincil anahtar olarak ve tam sayı tipinde tanımlanan nitelik, her kayıtta otomatik olarak birer birer artması için IDENTITY(1,1) anahtar kelimesi kullanılıyor.

menuID niteliği: Yabancıl anahtar (Menuler(menuID) referansı) olarak tanımlanan ve tam sayı tipinde bir nitelik oluşturuluyor.

rezervasyonYapanNo niteliği: Yalnızca 9 karakterden oluşuyor.

```
CREATE TABLE Rezervasyon (
    rezervasyonID           INT IDENTITY(1,1) PRIMARY KEY,
    menuID                  INT FOREIGN KEY REFERENCES Menuler(menuID),
    rezervasyonYapanNo      char(9) NOT NULL DEFAULT '0',
)
```

Şekil 28: ‘Rezervasyon’ tablosunun oluşturulmasına dair SQL komutları

```
INSERT INTO Rezervasyon(menuID, rezervasyonYapanNo)
VALUES
(1, '[redacted]301003'), (1, '[redacted]9001'), (1, [redacted]311001'), (1, '[redacted]04001'), (1, [redacted]310201'),
(2, '[redacted]301001'), (2, '[redacted]0501'), (2, [redacted]302001'), (2, '[redacted]03001'), (2, [redacted]308002'),
(3, '[redacted]303001'), (3, '[redacted]4001'), (3, [redacted]310001'), (3, '[redacted]14001'), (3, [redacted]315001'),

(4, '[redacted]310604'), (4, '[redacted]5001'), (4, '[redacted]316001'), (4, '[redacted]317001'), (4, '[redacted]318001'),
(5, '[redacted]315001'), (5, '[redacted]7001'), (5, '[redacted]310701'), (5, '[redacted]319001'), (5, '[redacted]319002'),
(6, '[redacted]10801'), (6, '[redacted]5002'), (6, '[redacted]305001'), (6, '[redacted]304001'), (6, '[redacted]320003'),

(7, '[redacted]20001'), (7, [redacted]7001'), (7, '[redacted]4002'), (7, '[redacted]303001'), (7, [redacted]310601'),
(8, '[redacted]02001'), (8, [redacted]0001'), (8, '[redacted]0601'), (8, '[redacted]319001'), (8, [redacted]314001'),
(9, [redacted]11001'), (9, [redacted]0201'), (9, '[redacted]0601'), (9, '[redacted]315002'), (9, [redacted]318001'),

(10, '[redacted]4001'), (10, '[redacted]0602'), (10, [redacted]310501'), (10, [redacted]308001'), (10, [redacted]318002'),
(11, '[redacted]3002'), (11, '[redacted]0102'), (11, [redacted]310001'), (11, [redacted]308001'), (11, [redacted]303001'),
(12, '[redacted]15001'), (12, '[redacted]1001'), (12, [redacted]310604'), (12, [redacted]304003'), (12, [redacted]303001'),

(13, '[redacted]0801'), (13, '[redacted]0202'), (13, [redacted]317001'), (13, '[redacted]303002'), (13, [redacted]311002'),
(14, '[redacted]02001'), (14, '[redacted]0602'), (14, [redacted]310501'), (14, '[redacted]320001'), (14, [redacted]318001'),
(15, [redacted]3001'), (15, '[redacted]20001'), (15, [redacted]310503'), (15, '[redacted]304003'), (15, [redacted]304001')
```

Şekil 29: ‘Rezervasyon’ tablosuna örnek verilerin eklenmesi

5.2.6) ‘Menuler’ tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile ‘Menuler’ tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

menuID niteliği: Birincil anahtar olarak ve tam sayı tipinde tanımlanarak oluşturuluyor.

yemekID niteliği: Yabancıl anahtar (Yemekler(yemekID) referansı) olarak tanımlanan ve tam sayı tipinde bir nitelik oluşturuluyor.

menuFiyat niteliği: Tam sayı tipinde değer alıyor ve 0’dan büyük değerler alması için kontrol ediliyor.

menuTarih niteliği: ‘YYYY-MM-DD’ tarih formatından veri girişini kabul ediyor.

ogunTip niteliği: Yalnızca 5 karakterden oluşuyor ve yalnızca ‘Sabah’, ‘Öğlen’ ve ‘Akşam’ değerlerinden birini alıyor.

```
CREATE TABLE Menuler (
    menuID          INT IDENTITY(1,1) PRIMARY KEY,
    menuFiyat       SMALLINT, CHECK (menuFiyat > 0),
    menuTarih       DATE,
    ogunTip         char(5), CHECK (ogunTip IN ('Sabah', 'Öğlen', 'Akşam')),
)
```

Şekil 30: ‘Menuler’ tablosunun oluşturulmasına dair SQL komutları

```
INSERT INTO Menuler(menuFiyat, menuTarih, ogunTip)
VALUES
(20, '2024-05-20', 'Sabah'),
(25, '2024-05-20', 'Öğlen'),
(30, '2024-05-20', 'Akşam'),

(20, '2024-05-21', 'Sabah'),
(25, '2024-05-21', 'Öğlen'),
(30, '2024-05-21', 'Akşam'),

(20, '2024-05-22', 'Sabah'),
(25, '2024-05-22', 'Öğlen'),
(30, '2024-05-22', 'Akşam'),

(20, '2024-05-23', 'Sabah'),
(25, '2024-05-23', 'Öğlen'),
(30, '2024-05-23', 'Akşam'),

(20, '2024-05-24', 'Sabah'),
(25, '2024-05-24', 'Öğlen'),
(30, '2024-05-24', 'Akşam');
```

Şekil 31: ‘Menuler’ tablosuna örnek verilerin eklenmesi

5.2.7) ‘Yemekler’ tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile ‘Yemekler’ tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

yemekID niteliği: Birincil anahtar olarak ve tam sayı tipinde tanımlanan nitelik, her kayıtta otomatik olarak birer birer artması için IDENTITY(1,1) anahtar kelimesi kullanılıyor.

yemekAd niteliği: Maksimum 50 karakterden oluşuyor.

yemekKalori niteliği: Tam sayı tipinde değer alıyor ve 0’dan büyük değerler alması için kontrol ediliyor.

yemekTur niteliği: Maksimum 50 karakterden oluşuyor ve yalnızca ‘Ana Yemek’, ‘Çorba’, ‘Salata’, ‘Tatlı’, ‘Atıştırmalık’, ‘Kahvaltılık’ ve ‘İçecek’ değerlerini alabiliyor.

```
CREATE TABLE Yemekler (
    yemekID         INT IDENTITY(1,1) PRIMARY KEY,
    yemekAd         varchar(50),
    yemekTur        varchar(50),
    CHECK (yemekTur IN ('Ana Yemek', 'Çorba', 'Salata', 'Tatlı', 'Atıştırmalık', 'İçecek')),
    yemekKalori     SMALLINT, CHECK (yemekKalori > 0)
)
```

Şekil 32: ‘Yemekler’ tablosunun oluşturulmasına dair SQL komutları

```
INSERT INTO Yemekler(yemekAd, yemekTur, yemekKalori)
VALUES
('Salçalı Makarna', 'Ana Yemek', 230), ('Bulgur Pilavı', 'Ana Yemek', 168), ('Pirinç Pilavı', 'Ana Yemek', 147),
('Tavuk Izgara', 'Ana Yemek', 302), ('Et Köfte', 'Ana Yemek', 235), ('Karnıyarık', 'Ana Yemek', 191),
('Patates Oturtma', 'Ana Yemek', 469), ('Kabak Mücver', 'Ana Yemek', 145), ('Lazanya', 'Ana Yemek', 494),
('Çökertme Kebabı', 'Ana Yemek', 1201), ('Etli Nohut', 'Ana Yemek', 285), ('Kuru Fasulye', 'Ana Yemek', 379),
('Tavuk Sote', 'Ana Yemek', 288), ('Karnabahar', 'Ana Yemek', 112), ('Mantı', 'Ana Yemek', 322),
('Ezogelin Çorbası', 'Çorba', 91), ('Yayla Çorbası', 'Çorba', 99), ('Domates Çorbası', 'Çorba', 67),
('Kelle Paça Çorbası', 'Çorba', 330), ('Mantar Çorbası', 'Çorba', 85), ('Yuvarlak Köfte Çorbası', 'Çorba', 120),
('Mercimek Çorbası', 'Çorba', 138), ('Kremalı Tavuk Çorbası', 'Çorba', 117), ('Havuç Çorbası', 'Çorba', 95),
('Brokoli Çorbası', 'Çorba', 206), ('Tavuklu Salata', 'Salata', 130), ('Yeşil Salata', 'Salata', 77),
('Çoban Salata', 'Salata', 115), ('Rus Salatası', 'Salata', 255), ('Mevsim Salata', 'Salata', 78),
('Şam Tatlısı', 'Tatlı', 330), ('Lokma Tatlısı', 'Tatlı', 957), ('Kabak Tatlısı', 'Tatlı', 139),
('Kemalpaşa Tatlısı', 'Tatlı', 476), ('Kadife Tatlısı', 'Tatlı', 550), ('Salam ve Kaşarlı Sandviç', 'Atıştırmalık', 450),
('Tam Tahıllı Buğday Ekmeği', 'Atıştırmalık', 68), ('Beyaz Buğday Ekmeği', 'Atıştırmalık', 83), ('Yoğurt', 'Atıştırmalık', 122),
('Beyaz Peynir', 'Atıştırmalık', 95), ('Sarı ve Siyah Zeytin', 'Atıştırmalık', 100), ('Domates ve Salatalık', 'Atıştırmalık', 35),
('Peynirli Börek', 'Atıştırmalık', 380), ('Peynirli Poğaça', 'Atıştırmalık', 275), ('Su', 'İçecek', 1),
('Ayran', 'İçecek', 56), ('Meyve Suyu', 'İçecek', 120), ('Çay', 'İçecek', 15), ('Bitki Çayı', 'İçecek', 10);
```

Şekil 33: ‘Yemekler’ tablosuna örnek verilerin eklenmesi

5.2.8) ‘YemeklerMenuler’ tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile ‘YemeklerMenuler’ tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

menuID niteliği: ‘yemekID’ sütunu ile beraber **(menuID, yemekID) birleşik anahtarının (composite key)** bir birincil anahtarını oluşturuyor. Aynı zamanda yabancı anahtar olarak ‘Menuler’ tablosunun ‘menuID’ sütununu referans gösteriyor.

yemekID niteliği: ‘menuID sütunu ile beraber **(menuID, yemekID) birleşik anahtarının (composite key)** bir birincil anahtarını oluşturuyor. Aynı zamanda yabancı anahtar olarak ‘Yemekler’ tablosunun ‘yemekID’ sütununu referans gösteriyor.

```
CREATE TABLE YemeklerMenuler (
    menuID          INT,
    yemekID         INT,
    PRIMARY KEY (menuID, yemekID),
    FOREIGN KEY (menuID) REFERENCES Menuler(menuID),
    FOREIGN KEY (yemekID) REFERENCES Yemekler(yemekID)
)
```

Şekil 34: ‘YemeklerMenuler’ tablosunun oluşturulmasına dair SQL komutları

```
INSERT INTO YemeklerMenuler(menuID, yemekID)
VALUES
(1, 45), (1, 48), (1, 43), (1, 40), (1, 41), (1, 42),
(2, 1), (2, 16), (2, 13), (2, 46), (2, 31), (2, 45),
(3, 17), (3, 5), (3, 2), (3, 30), (3, 46), (3, 45),

(4, 37), (4, 40), (4, 41), (4, 42), (4, 44), (4, 48),
(5, 22), (5, 15), (5, 30), (5, 39), (5, 45),
(6, 3), (6, 24), (6, 35), (6, 45), (6, 29),

(7, 36), (7, 40), (7, 41), (7, 42), (7, 45), (7, 49),
(8, 20), (8, 8), (8, 26), (8, 45), (8, 48),
(9, 7), (9, 23), (9, 29), (9, 45), (9, 46),

(10, 43), (10, 40), (10, 41), (10, 42), (10, 45), (10, 47),
(11, 24), (11, 14), (11, 32), (11, 45), (11, 46),
(12, 19), (12, 11), (12, 28), (12, 45), (12, 46),

(13, 36), (13, 40), (13, 41), (13, 42), (13, 45), (13, 49),
(14, 18), (14, 12), (14, 3), (14, 45), (14, 46),
(15, 17), (15, 3), (15, 11), (15, 30), (15, 33), (15, 45);
```

Şekil 35: ‘YemeklerMenuler’ tablosuna örnek verilerin eklenmesi

5.2.9) ‘Odemeler’ tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile ‘Odemeler’ tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

odemeID niteliği: Birincil anahtar olarak ve tam sayı tipinde tanımlanan nitelik, her kayıtta otomatik olarak birer birer artması için IDENTITY(1,1) anahtar kelimesi kullanılıyor.

rezervasyonID niteliği: Yabancı anahtar (Rezervasyon(rezervasyonID) referansı) olarak tanımlanan ve tam sayı tipinde bir nitelik oluşturuluyor.

odemeYapanNo niteliği: Yalnızca 9 karakterden oluşuyor.

odemeTarih niteliği: ‘YYYY-MM-DD HH:MM:SS’ tarih formatından veri girişini kabul ediyor.

odemeTutar niteliği: Tam sayı tipinde değer alıyor ve 0’dan büyük değerler alması için kontrol ediliyor.

```
CREATE TABLE Odemeler (
    odemeID          INT IDENTITY(1,1) PRIMARY KEY,
    rezervasyonID    INT FOREIGN KEY REFERENCES Rezervasyon(rezervasyonID),
    odemeYapanNo     char(9) NOT NULL DEFAULT '0',
    odemeTarih       DATETIME,
    odemeTutar       SMALLINT, CHECK (odemeTutar > 0)
)
```

Şekil 36: ‘Odemeler’ tablosunun oluşturulmasına dair SQL komutları

```
INSERT INTO Odemeler(rezervasyonID, odemeYapanNo, odemeTarih, odemeTutar)
VALUES

(1, [redacted]1003', '2024-05-13 07:12:08', 20), (2, [redacted]9001', '2024-05-12 04:01:37', 20),
(3, [redacted]1001', '2024-05-11 17:39:45', 20), (4, [redacted]04001', '2024-05-12 03:46:58', 20),
(5, [redacted]0201', '2024-05-15 21:55:31', 20),
(6, [redacted]1001', '2024-05-11 14:34:12', 25), (7, [redacted]10501', '2024-05-12 00:15:05', 25),
(8, [redacted]2001', '2024-05-11 19:47:52', 25), (9, [redacted]03001', '2024-05-13 06:23:40', 25),
(10, [redacted]08002', '2024-05-15 10:57:24', 25),
(11, [redacted]03001', '2024-05-15 14:34:12', 30), (12, [redacted]304001', '2024-05-14 00:15:05', 30),
(13, [redacted]10001', '2024-05-16 19:47:52', 30), (14, [redacted]314001', '2024-05-14 06:23:40', 30),
(15, [redacted]15001', '2024-05-17 10:57:24', 30),
(16, [redacted]10604', '2024-05-12 13:12:08', 20), (17, [redacted]315001', '2024-05-13 12:31:37', 20),
(18, [redacted]16001', '2024-05-13 19:32:35', 20), (19, [redacted]17001', '2024-05-12 16:16:58', 20),
(20, [redacted]18001', '2024-05-13 21:23:31', 20),
(21, [redacted]15001', '2024-05-14 04:35:12', 25), (22, [redacted]317001', '2024-05-11 13:55:05', 25),
(23, [redacted]10701', '2024-05-16 12:36:52', 25), (24, [redacted]19001', '2024-05-16 13:23:40', 25),
(25, [redacted]19002', '2024-05-15 15:56:24', 25),
(26, [redacted]10801', '2024-05-12 15:16:17', 30), (27, [redacted]05002', '2024-05-15 14:35:05', 30),
(28, [redacted]05001', '2024-05-12 12:45:59', 30), (29, [redacted]04001', '2024-05-16 16:53:40', 30),
(30, [redacted]20003', '2024-05-12 13:57:24', 30),
(31, [redacted]20001', '2024-05-13 11:58:08', 20), (32, [redacted]317001', '2024-05-19 14:31:37', 20),
(33, [redacted]04002', '2024-05-15 09:47:38', 20), (34, [redacted]3001', '2024-05-15 23:18:58', 20),
(35, [redacted]10601', '2024-05-13 21:23:31', 20),
(36, [redacted]02001', '2024-05-14 11:27:25', 25), (37, [redacted]310001', '2024-05-16 17:55:05', 25),
(38, [redacted]10601', '2024-05-17 16:37:23', 25), (39, [redacted]319001', '2024-05-17 11:29:40', 25),
(40, [redacted]14001', '2024-05-19 18:21:27', 25),
(41, [redacted]11001', '2024-05-18 22:42:17', 30), (42, [redacted]310201', '2024-05-18 19:18:05', 30),
(43, [redacted]10601', '2024-05-20 16:22:38', 30), (44, [redacted]15002', '2024-05-20 19:12:45', 30),
(45, [redacted]18001', '2024-05-17 11:37:45', 30)
```

Şekil 37: 'Odemeler' tablosuna örnek verilerin eklenmesi

5.2.10) 'Geribildirimler' tablosunun oluşturulması ve verilerin eklenmesi

Aşağıdaki görselde görüldüğü üzere kullandığımız SQL komutları ile 'Geribildirimler' tablomuzu oluşturuyor ve tabloda bulunan nitelikleri (sütunları) ve kullanılacak veri tiplerini belirliyoruz.

geribildirimID niteliği: Birincil anahtar olarak ve tam sayı tipinde tanımlanan nitelik, her kayıtta otomatik olarak birer birer artması için IDENTITY(1,1) anahtar kelimesi kullanılıyor.

rezervasyonID niteliği: Yabancıl anahtar (Rezervasyon(rezervasyonID) referansı) olarak tanımlanan ve tam sayı tipinde bir nitelik oluşturuluyor.

geribildirimYapanNo niteliği: Yalnızca 9 karakterden oluşuyor.

geribildirimTarih niteliği: 'YYYY-MM-DD' tarih formatından veri girişini kabul ediyor.

geribildirimYorum niteliği: Metin tipinde uzun boyutlu veriler tutuyor ve değer girilmediği zaman varsayılan olarak 'Yorum yok.' değerini alıyor.

geribildirimPuan niteliği: Tam sayı tipinde değer alıyor ve yalnızca 0 ile 10 arasındaki değerleri alabiliyor.

```sql
CREATE TABLE Geribildirimler (
    geribildirimID          INT IDENTITY(1,1) PRIMARY KEY,
    rezervasyonID           INT FOREIGN KEY REFERENCES Rezervasyon(rezervasyonID),
    geribildirimYapanNo     char(9) NOT NULL DEFAULT '0',
    geribildirimTarih       DATETIME,
    geribildirimYorum       TEXT NOT NULL DEFAULT 'Yorum yok.',
    geribildirimPuan        TINYINT, CHECK (geribildirimPuan BETWEEN 0 AND 10)
)
```

Şekil 38: 'Geribildirimler' tablosunun oluşturulmasına dair SQL komutları

```sql
INSERT INTO Geribildirimler(rezervasyonID, geribildirimYapanNo, geribildirimTarih, geribildirimYorum, geribildirimPuan)
VALUES

(1, [redacted]003', '2024-05-20 09:14:08', 'Peynir taze değildi!', 6), (2, '993319001', '2024-05-20 10:15:37', 'Çok beğendim!', 10),
(3, [redacted]001', '2024-05-20 10:32:45', 'Çay dışında beğendim.', 8), (4, '993304001', '2024-05-20 11:23:58', 'Salatalık çürüktü!', 5),
(5, [redacted]201', '2024-05-20 09:57:31', 'Domates ezilmiş, beğenmedim.', 5),
(6, [redacted]001', '2024-05-20 13:32:12', 'Çorba çok tuzlu.', 5), (7, '883310501', '2024-05-20 15:42:05', 'Hastayım, lütfen yemekleri tuzlu yap
(8, [redacted]001', '2024-05-20 15:21:52', 'Çorba dışında iyiydi.', 6), (9, '223303001', '2024-05-20 16:53:40', 'Ana yemek muazzamdı.', 8),
(10, [redacted]8002', '2024-05-20 15:58:24', 'Çok tuzlu!', 4),
(11, [redacted]3001', '2024-05-20 19:23:12', 'En iyi akşam yemeği!', 10), (12, '993304001', '2024-05-20 20:11:05', 'Çorbayı beğenmedim.', 7),
(13, [redacted]0001', '2024-05-20 23:15:52', 'Çorba çok sulu.', 6), (14, '223314001', '2024-05-20 20:24:40', 'İdare eder, tuzu orta.', 6),
(15, [redacted]5001', '2024-05-20 20:48:28', 'Çorba dışında fena değildi', 8),
(16, [redacted]0604', '2024-05-21 09:23:08', 'Berbat bir kahvaltı!', 3), (17, '223315001', '2024-05-21 11:45:37', 'Sebzeler çürük!', 4),
(18, [redacted]6001', '2024-05-21 09:32:35', 'Ana yemeği beğenmedim.', 5), (19, '223317001', '2024-05-21 11:23:55', 'Fena değil, klasik.', 6),
(20, [redacted]8001', '2024-05-21 10:42:34', 'Hiç güzel değil kahvaltı.', 3),
(21, [redacted]5001', '2024-05-21 13:42:14', 'Çorba harika!', 8), (22, '993317001', '2024-05-21 14:12:32', 'Yemek güzeldi.', 7),
(23, [redacted]0701', '2024-05-21 13:55:12', 'Ana yemekle içecek uyumlu!', 10), (24, '223319001', '2024-05-21 14:12:32', 'Ben beğenmedim!', 4),
(25, [redacted]9002', '2024-05-21 15:23:25', 'Çorba çok sulu olmuş!', 4),
(26, [redacted]0801', '2024-05-21 17:36:11', 'Çorba tuzlu olmuş!', 4), (27, '223305002', '2024-05-21 18:42:13', 'Yemek idare eder!', 5),
(28, [redacted]5001', '2024-05-21 19:23:19', 'Tatlıya bayıldım!', 8), (29, '223304001', '2024-05-21 21:23:19', 'Çorba fena değildi!', 6),
(30, [redacted]0003', '2024-05-21 19:42:15', 'Çorba fena değil bence!', 7),
(31, [redacted]0001', '2024-05-22 11:32:08', 'İdare eder!', 6), (32, '993317001', '2024-05-22 09:15:27', 'Sebzeler çürüktü!', 4),
(33, [redacted]4002', '2024-05-22 08:45:32', 'Ana yemek biraz kuru!', 6), (34, '223303001', '2024-05-22 12:14:28', 'Çay tam demlenmemiş.', 6),
(35, [redacted]0601', '2024-05-22 11:24:52', 'Daha besleyici menüler gerekiyor, kahvaltı kötüydü.', 5),
(36, [redacted]2001', '2024-05-22 14:12:54', 'Çorba fena değildi!', 6), (37, '993310001', '2024-05-22 14:11:27', 'Kötü bir akşam yemeği.', 3),
(38, [redacted]0601', '2024-05-22 16:17:23', 'Eh, ana yemek idare eder.!', 6), (39, '223319001', '2024-05-22 13:12:44', 'İdare eder!', 5),
(40, [redacted]4001', '2024-05-22 15:42:19', 'Ana yemek muazzamdı, bir daha yapın.', 9),
(41, [redacted]1001', '2024-05-22 19:41:23', 'Aç kaldık!', 3), (42, '883310201', '2024-05-22 21:13:15', 'Ben beğenmedim!', 4),
(43, [redacted]0601', '2024-05-22 20:22:48', 'Hem güzel hem kötü!', 5), (44, '223315002', '2024-05-22 20:15:12', 'Yemek güzeldi.', 8),
(45, [redacted]8001', '2024-05-22 22:14:23', 'İdare eder, daha güzel yapılabilir.', 7)
```

Şekil 39: 'Geribildirimler' tablosuna örnek verilerin eklenmesi

6) Veri tabanı sorgu örnekleri (SQL/T-SQL)

6.1) Normal sorgu örnekleri

Aşağıdaki sorgu ile ‘Menuler’, ‘YemeklerMenuler’ ve ‘Yemekler’ tablosunu birleştirip belirli bir ‘menuID’ değeri üzerinden ilgili menüdeki yemeklerin listesine ve özelliklerine ulaşılıyor.

```sql
-- Belirli bir menüde bulunan yemeklerin listesini sorgulama

SELECT Menuler.menuID, Yemekler.*  FROM Menuler
JOIN YemeklerMenuler ON YemeklerMenuler.menuID = Menuler.menuID
JOIN Yemekler ON Yemekler.yemekID = YemeklerMenuler.yemekID WHERE Menuler.menuID = 5;
```

119 %

Results | Messages

	menuID	yemekID	yemekAd	yemekTur	yemekKalori
1	5	15	Mantı	Ana Yemek	322
2	5	22	Mercimek Çorbası	Çorba	138
3	5	30	Mevsim Salata	Salata	78
4	5	39	Yoğurt	Atıştırmalık	122
5	5	45	Su	İçecek	1

Şekil 40: Bir menüdeki yemeklerin listesini veren SQL komutu

Aşağıdaki sorgu ile belirli bir ‘rezervasyonID’ değişkeninin değeri üzerinden ilgili rezervasyona ait menüdeki yemeklerin listesi sorgulanıyor.

```sql
-- Bir rezervasyonun yapıldığı menüdeki yemeklerin listesi

DECLARE @rezervasyonID INT = 5;
SELECT Menuler.menuTarih, Menuler.ogunTip, Yemekler.* FROM Rezervasyon
JOIN Menuler ON Rezervasyon.menuID = Menuler.menuID
JOIN YemeklerMenuler ON YemeklerMenuler.menuID = Menuler.menuID
JOIN Yemekler ON YemeklerMenuler.yemekID = Yemekler.yemekID
WHERE Rezervasyon.rezervasyonID = @rezervasyonID
```

119 %

Results | Messages

	menuTarih	ogunTip	yemekID	yemekAd	yemekTur	yemekKalori
1	2024-05-20	Sabah	40	Beyaz Peynir	Atıştırmalık	95
2	2024-05-20	Sabah	41	Sarı ve Siyah Zeytin	Atıştırmalık	100
3	2024-05-20	Sabah	42	Domates ve Salatalık	Atıştırmalık	35
4	2024-05-20	Sabah	43	Peynirli Börek	Atıştırmalık	380
5	2024-05-20	Sabah	45	Su	İçecek	1
6	2024-05-20	Sabah	48	Çay	İçecek	15

Şekil 41: Bir rezervasyon bilgisi üzerinden ilgili rezervasyona ait menüdeki yemeklerin listesini veren SQL komutu

Aşağıdaki sorgu ile belirli bir ‘hesaplanacakMenu’ değişkeninin değeri ile eşleşen ‘menuID’ üzerinden ilgili menüye ait yemeklerin kalorileri ortalamasını hesaplanıyor.

```sql
-- Belirli bir menüdeki yemeklerin kalori ortalamalarını hesaplama

DECLARE @hesaplanacakMenu INT = 5;
SELECT Menuler.menuID, AVG(Yemekler.yemekKalori) as OrtalamaKalori FROM Menuler
JOIN YemeklerMenuler ON YemeklerMenuler.menuID = Menuler.menuID
JOIN Yemekler ON YemeklerMenuler.yemekID = Yemekler.yemekID
WHERE Menuler.menuID = @hesaplanacakMenu
GROUP BY Menuler.menuID
```

119 %

Results | Messages

	menuID	OrtalamaKalori
1	5	132

Şekil 42: Bir menü bilgisi üzerinden ilgili menüdeki yemeklerin kalori ortalamalarını hesaplayan SQL komutu

Aşağıdaki sorgu ile belirli bir ‘kalori’ değişkeninin değerinden büyük yemekler, ‘Yemekler’ tablosundan karşılaştırılarak her yemek çeşidinden kaçar adet olduğu bulunuyor.

```sql
-- Belirli bir kalori değerinin üstünde, her yemek çeşidinden kaçar adet olduğunu bulma
DECLARE @kalori INT = 150;
SELECT Yemekler.yemekTur, COUNT(Yemekler.yemekTur) as 'Çeşit Sayısı' FROM Yemekler
WHERE Yemekler.yemekKalori > @kalori
GROUP BY Yemekler.yemekTur
```

119 %

Results | Messages

	yemekTur	Çeşit Sayısı
1	Ana Yemek	12
2	Atıştırmalık	3
3	Çorba	2
4	Salata	1
5	Tatlı	4

Şekil 43: Belirli bir kalori değeri üstünde olan yemek çeşitlerinden kaç adet yemek olduğunu bulan SQL komutu

Aşağıdaki sorgu ile ‘Calisan’ tablosunda bulunan çalışanların yaş ve maaş ortalamaları hesaplanıyor.

```sql
-- Çalışan kişilerin yaş ve maaş ortalamasını bulma

SELECT
SUM(Calisan.maas) / COUNT(Calisan.calisanNo) as OrtalamaMaas,
SUM(DATEDIFF(year, Calisan.dogumTarih, GETDATE())) / COUNT(Calisan.calisanNo) as YasOrtalamasi
FROM Calisan
```

119 %

Results | Messages

	OrtalamaMaas	YasOrtalamasi
1	27133,3333	38

Şekil 44: Çalışanların yaş ve maaş ortalamalarını hesaplayan SQL komutu

Aşağıdaki sorgu ile bir ‘girilenTarih’ değişkeni ile alınan tarih-saat değeri ile anlık tarih-saat değeri arasındaki ödeme kayıtlarının toplamı sorgulanıyor.

```sql
-- Girilen bir tarih/saat ile günümüz tarih/saat arasında yapılan ödemelerin toplamını bulma

DECLARE @girilenTarih DATETIME = '2024-05-16 15:14:25';
SELECT SUM(Odemeler.odemeTutar) as ToplamOdeme
FROM Odemeler
WHERE odemeTarih BETWEEN @girilenTarih AND GETDATE()
```

119 %

Results | Messages

	ToplamOdeme
1	360

Şekil 45: Belirli bir tarih-saat ile anlık tarih-saat değerleri arasında yapılan ödemelerin toplamını bulan SQL komutu

Aşağıdaki sorgu ile ‘geribildirimYapan’ isimli değişkenin değeri üzerinden ilgili kişinin yaptığı geribildirimler kayıtları bulunuyor.

```sql
-- Belirli bir kişinin numarasından yaptığı geribildirimleri bulma

DECLARE @geribildirimYapan INT = 993304001;
SELECT * FROM Geribildirimler
WHERE Geribildirimler.geribildirimYapanNo = @geribildirimYapan
```

119 %

Results | Messages

	geribildirimID	rezervasyonID	geribildirimYapanNo	geribildirimTarih	geribildirimYorum	geribildirimPuan
1	4	4	[redacted]04001	2024-05-20 11:23:58.000	Salatalık çürüktü!	5
2	12	12	[redacted]304001	2024-05-20 20:11:05.000	Çorbayı beğenmedim.	7

Şekil 46: Belirli bir kişinin yaptığı geribildirimleri bulan SQL komutu

Aşağıdaki sorgu ile ‘geribildirimMenu’ değişkeni değeri üzerinden ilgili menüye yapılan geribildirimleri bulunuyor ve geribildirimlere ait puanların ortalaması hesaplanıyor.

```
-- Belirli bir menü için yapılan geribildirim puanlarının ortalaması

DECLARE @geribildirimMenu INT = 5;
SELECT Menuler.menuID, CONVERT(decimal, AVG(Geribildirimler.geribildirimPuan)) as OrtalamaPuan
FROM Geribildirimler
JOIN Rezervasyon ON Rezervasyon.rezervasyonID = Geribildirimler.rezervasyonID
JOIN Menuler ON Rezervasyon.menuID = Menuler.menuID
WHERE Menuler.menuID = @geribildirimMenu
GROUP BY Menuler.menuID
```

119 %

Results | Messages

	menuID	OrtalamaPuan
1	5	6

Şekil 47: Belirli bir menü için yapılan geribildirimlerin puanlarının ortalamasını hesaplayan SQL komutu

Aşağıdaki sorgu ile belirli bir ‘zamOran’ değişkeninin değeri üzerinden ‘Calisan’ tablosundaki çalışanların maaş değerleri güncelleniyor.

```
-- Belirli bir oran üzerinden çalışanların maaşına zam yapma

DECLARE @zamOran INT = 20;
UPDATE Calisan
SET Calisan.maas = Calisan.maas + (Calisan.maas * @zamOran / 100)
```

119 %

Messages

```
(15 rows affected)

Completion time: 2024-05-25T18:46:10.0891721+03:00
```

Şekil 48: Belirli bir zam oranı üzerinden çalışanların maaşına zam yapan SQL komutu

Aşağıdaki görselde ilgili SQL komutu çalıştırılmadan önce ‘Calisan’ tablosundaki değerler görünüyor.

calisanNo	bolumID	ad	soyad	cinsiyet	dogumTarih	telefonNo	maas
[redacted]0001	B100	[redacted]	Köksoy	E	1985-01-07	5[redacted]2841	24000,0000
[redacted]0101	B101	[redacted]	Hepkaya	K	1989-01-16	5[redacted]9123	24000,0000
[redacted]0102	B101	[redacted]	Cümbüş	E	1990-07-23	5[redacted]1245	24000,0000
[redacted]0201	B102	[redacted]	Karaoğlan	K	1993-02-17	5[redacted]1020	24000,0000
[redacted]0202	B102	[redacted]	Kavuk	E	1986-11-25	5[redacted]2561	24000,0000
[redacted]0501	B105	[redacted]	Çırakoğlu	E	1986-04-15	5[redacted]6878	20000,0000
[redacted]0502	B105	[redacted]	Çetin	K	1979-02-23	5[redacted]9294	20000,0000
[redacted]0503	B105	[redacted]	Çalışkan	K	1985-01-11	5[redacted]0902	20000,0000
[redacted]0601	B106	[redacted]	Erdem	K	1981-06-17	5[redacted]0102	28000,0000
[redacted]0602	B106	[redacted]	Bolivar	K	1994-08-24	5[redacted]9192	28000,0000
[redacted]0603	B106	[redacted]	Sadık	E	1987-11-02	5[redacted]2342	28000,0000
[redacted]0604	B106	[redacted]	Kadıoğlan	K	1983-09-12	5[redacted]9621	28000,0000
[redacted]0701	B107	[redacted]	Eser	K	1982-04-13	5[redacted]4123	30000,0000
[redacted]0702	B107	[redacted]	İnanır	E	1988-04-19	5[redacted]9201	30000,0000
[redacted]0801	B108	[redacted]	Aslanoğlu	E	1970-02-03	5[redacted]9102	55000,0000

Şekil 49: Çalışan maaşlarına zam yapan SQL komutu çalıştırılmadan önce

Aşağıdaki görselde ise ilgili SQL komutu çalıştırıldıktan sonra ‘Calisan’ tablosundaki güncel değerler görünüyor.

calisanNo	bolumID	ad	soyad	cinsiyet	dogumTarih	telefonNo	maas
[redacted]0001	B100	[redacted]	Köksoy	E	1985-01-07	5886712841	28800,0000
[redacted]0101	B101	[redacted]	Hepkaya	K	1989-01-16	5154869123	28800,0000
[redacted]0102	B101	[redacted]	Cümbüş	E	1990-07-23	5196981245	28800,0000
[redacted]0201	B102	[redacted]	Karaoğlan	K	1993-02-17	5189601020	28800,0000
[redacted]0202	B102	[redacted]	Kavuk	E	1986-11-25	5827582561	28800,0000
[redacted]0501	B105	[redacted]	Çırakoğlu	E	1986-04-15	5129496878	24000,0000
[redacted]0502	B105	[redacted]	Çetin	K	1979-02-23	5875819294	24000,0000
[redacted]0503	B105	[redacted]	Çalışkan	K	1985-01-11	5981020902	24000,0000
[redacted]0601	B106	[redacted]	Erdem	K	1981-06-17	5998520102	33600,0000
[redacted]0602	B106	[redacted]	Bolivar	K	1994-08-24	5748689192	33600,0000
[redacted]0603	B106	[redacted]	Sadık	E	1987-11-02	5009912342	33600,0000
[redacted]0604	B106	[redacted]	Kadıoğlan	K	1983-09-12	5628699621	33600,0000
[redacted]0701	B107	[redacted]	Eser	K	1982-04-13	5916864123	36000,0000
[redacted]0702	B107	[redacted]	İnanır	E	1988-04-19	5148689201	36000,0000
[redacted]0801	B108	[redacted]	Aslanoğlu	E	1970-02-03	5172869102	66000,0000

Şekil 50: Çalışan maaşlarına zam yapan SQL komutu çalıştırıldıktan sonra

6.2) Saklı yordamlar (Stored procedures) örnekleri

Aşağıdaki saklı yordam, çalıştırıldığı takdirde en çok rezervasyon yapılan tarih bilgisini sorguluyor.

```sql
CREATE PROCEDURE enCokRezervasyon
AS
BEGIN
    SELECT TOP 1 Menuler.menuTarih AS 'En çok rez. yapılan tarih', COUNT(Menuler.menuTarih) AS RezervasyonSayisi
    FROM Rezervasyon
    JOIN Menuler ON Menuler.menuID = Rezervasyon.menuID
    GROUP BY menuTarih ORDER BY COUNT(Menuler.menuTarih) DESC
END

exec enCokRezervasyon
```

119 %

Results | Messages

	En çok rez. yapılan tarih	RezervasyonSayisi
1	2024-05-24	16

Şekil 51: En çok rezervasyon yapılan tarihi bulan SQL saklı yordamı

Aşağıdaki saklı yordam, belirli bir 'rezervasyonYapanNo' değişken değerini 'Ogrenci', 'Calisan' ve 'Akademisyen' tabloları üzerinden sorguluyor ve ilgili değerin ait olduğu kayıt bilgilerini ve kişiye ait rezervasyon bilgilerini sorguluyor.

```sql
-- Belirli bir rezervasyon yapan kişinin numarasından kim olduğunu bulma ve numaranın ait olduğu rezervasyon bilgilerini gösterme
-- (stored procedure)
CREATE PROCEDURE rezervasyonKisiOgrenme
@rezervasyonYapanNo char(9)
AS
BEGIN
DECLARE @tip NVARCHAR(50);
SELECT @tip =
    CASE
        WHEN EXISTS (SELECT 1 FROM Ogrenci WHERE OgrenciNo = @rezervasyonYapanNo) THEN 'Ogrenci'
        WHEN EXISTS (SELECT 1 FROM Akademisyen WHERE AkademisyenNo = @rezervasyonYapanNo) THEN 'Akademisyen'
        WHEN EXISTS (SELECT 1 FROM Calisan WHERE CalisanNo = @rezervasyonYapanNo) THEN 'Calisan'
        ELSE 'Aradığınız numara, belirtilen tablolarda bulunamadı.'
    END;

    IF @tip = 'Ogrenci'
    BEGIN
        SELECT Ogrenci.*, Rezervasyon.menuID, Rezervasyon.rezervasyonID FROM Ogrenci
        INNER JOIN Rezervasyon ON Rezervasyon.rezervasyonYapanNo = @rezervasyonYapanNo
        WHERE ogrenciNo = @rezervasyonYapanNo;
    END

    ELSE IF @tip = 'Akademisyen'
    BEGIN
        SELECT Akademisyen.*, Rezervasyon.menuID, Rezervasyon.rezervasyonID FROM Akademisyen
        INNER JOIN Rezervasyon ON Rezervasyon.rezervasyonYapanNo = @rezervasyonYapanNo
        WHERE akademisyenNo = @rezervasyonYapanNo;
    END
```

Şekil 52: Rezervasyon yapan belirli bir kişinin numarasının kime ait olduğunu bulan SQL saklı yordamı

```
    END
    ELSE IF @tip = 'Calisan'
    BEGIN
        SELECT Calisan.*, Rezervasyon.menuID, Rezervasyon.rezervasyonID FROM Calisan
        INNER JOIN Rezervasyon ON Rezervasyon.rezervasyonYapanNo = @rezervasyonYapanNo
        WHERE calisanNo = @rezervasyonYapanNo
    END
    ELSE
    BEGIN
        PRINT 'Rezervasyon yapan kişi bulunamadı.';
    END
END

exec rezervasyonKisiOgrenme '223317001'
```

119 %

Results | Messages

	ogrenciNo	bolumID	ad	soyad	cinsiyet	dogumTarih	eposta	telefonNo	sinif	menuID	rezervasyonID
1	[redacted]7001	B017	Onur	Altınay	E	2004-10-21	onu[redacted]@ogr.selcuk.edu.tr	5[redacted]512	1	4	19
2	[redacted]7001	B017	Onur	Altınay	E	2004-10-21	onu[redacted]@ogr.selcuk.edu.tr	5[redacted]2512	1	13	63

Şekil 53: Rezervasyon yapan belirli bir kişinin numarasının kime ait olduğunu bulan SQL saklı yordamı (sorgunun devamı)

6.3) Tetikleyici (Trigger) örneği

Aşağıdaki tetikleyici, bir öğrencinin numarası değiştiğinde tetiklenerek ilgili öğrenciye ait rezervasyon, ödeme ve geribildirim kayıtlarındaki eski numaranın yeni numarayla değişmesini sağlıyor.

```
CREATE TRIGGER GuncellemeOgrenci
ON Ogrenci
AFTER UPDATE
AS
BEGIN
    DECLARE @eskiNumara INT;
    DECLARE @yeniNumara INT;

    SELECT @eskiNumara = deleted.ogrenciNo FROM deleted
    SELECT @yeniNumara = inserted.ogrenciNo FROM inserted



    UPDATE Odemeler SET odemeYapanNo = @yeniNumara
    FROM Odemeler WHERE Odemeler.odemeYapanNo = @eskiNumara

    UPDATE Geribildirimler SET geribildirimYapanNo = @yeniNumara
    FROM Geribildirimler WHERE Geribildirimler.geribildirimYapanNo = @eskiNumara

    UPDATE Rezervasyon SET rezervasyonYapanNo = @yeniNumara
    FROM Rezervasyon WHERE Rezervasyon.rezervasyonYapanNo = @eskiNumara
END
```

Şekil 54: Bir öğrencinin numarası değiştiğinde ilgili tablolardaki değerleri güncelleyen SQL tetikleyicisi

Aşağıdaki görselde, ilgili tetikleyici çalışmadan önce bir öğrencinin ‘Ogrenci’ tablosundaki kaydı görülüyor.

ogrenciNo	bolumID	ad	soyad	cinsiyet	dogumTarih	eposta	telefonNo	sinif
[redacted]01001	B001	Ahmet	[redacted]	E	2002-01-04	ahmet[redacted]	54[redacted]12	2
[redacted]01002	B001	Şükrü	[redacted]	E	2003-06-14	sukru[redacted]@...	55[redacted]23	2
[redacted]01003	B001	Sıla	[redacted]	K	2004-08-15	sila[redacted]@...	51[redacted]36	1

Şekil 55: Tetikleyici çalışmadan önce ilgili öğrenciye ait bilgiler

Aşağıdaki görselde ise ilgili tetikleyici çalışmadan önce ilgili öğrencinin ‘Rezervasyon’ tablosundaki kaydı görülüyor.

rezervasyo...	rezervasyo...	menuID
1	[redacted]1003	1

Şekil 56: Tetikleyici çalışmadan önce ilgili öğrenciye ait rezervasyon bilgisi

Aşağıdaki görselde ise ilgili tetikleyici çalışmadan önce ilgili öğrenciye ait ‘Odemeler’ tablosundaki kayıt görülüyor.

odemeID	rezervasyo...	odemeYap...	odemeTarih	odemeTutar
1	1	[redacted]1003	2024-05-13 ...	20

Şekil 57: Tetikleyici çalışmadan önce ilgili öğrenciye ait ödeme bilgisi

Aşağıdaki görselde ise ilgili tetikleyici çalışmadan önce ilgili öğrencinin ‘Geribildirimler’ tablosundaki kaydı görülüyor.

geribildiri...	rezervasyo...	geribildiri...	geribildiri...	geribildiri...	geribildiri...
1	1	[redacted]003	2024-05-20 ...	Peynir taze ...	6

Şekil 58: Tetikleyici çalışmadan önce ilgili öğrenciye ait geribildirim bilgisi

Aşağıdaki görselde ise tetikleyici çalıştıktan sonra ilgili öğrencinin ‘Ogrenci’ tablosundaki kaydı görülüyor. Görselde görüldüğü üzere öğrencinin numarasını ‘223301007’ olarak güncelliyoruz.

ogrenciNo	bolumID	ad	soyad	cinsiyet	dogumTarih	eposta	telefonNo	sinif
[redacted]01001	B001	Ahmet	[redacted]	E	2002-01-04	ahmet[redacted]	54[redacted]2	2
[redacted]002	B001	Şükrü	[redacted]	E	2003-06-14	sukru[redacted]@...	55[redacted]3	2
[redacted]1007	B001	Sıla	[redacted]	K	2004-08-15	sila[redacted]@...	51[redacted]6	1

Şekil 59: Tetikleyici çalıştıktan sonra ilgili öğrenciye ait bilgiler

Aşağıdaki görselde, ilgili tetikleyici çalıştıktan sonra ilgili öğrencinin ‘Rezervasyon’ tablosundaki kaydı görülüyor. Burada tetikleyicinin çalışarak ‘Rezervasyon’ tablosunda ilgili sütundaki değeri güncellediği görülüyor.

	rezervasyo...	rezervasyo...	menuID
▶	1	[redacted]007	1

Şekil 60: Tetikleyici çalıştıktan sonra ilgili öğrenciye ait rezervasyon bilgisi

Aşağıdaki görselde, ilgili tetikleyici çalıştıktan sonra ilgili öğrencinin ‘Odemeler’ tablosundaki kaydı görülüyor. Buradaki görselde tetikleyicinin çalışarak ‘Odemeler' tablosunda ilgili sütundaki değeri güncellediği görülüyor.

	odemeID	rezervasyo...	odemeYap...	odemeTarih	odemeTutar
▶	1	1	[redacted]1007	2024-05-13 ...	20

Şekil 61: Tetikleyici çalıştıktan sonra ilgili öğrenciye ait ödeme bilgisi

Aşağıdaki görselde ise ilgili tetikleyici çalıştıktan sonra ilgili öğrencinin ‘Geribildirimler’ tablosundaki kaydı görülüyor. Buradaki görselde de tetikleyicinin çalışarak ‘Geribildirimler’ tablosunda ilgili sütundaki değeri güncellediği görülüyor.

	geribildiri...	rezervasyo...	geribildiri...	geribildiri...	geribildiri...	geribildiri...
▶	1	1	[redacted]1007	2024-05-20 ...	Peynir taze ...	6

Şekil 62: Tetikleyici çalıştıktan sonra ilgili öğrenciye ait geribildirim bilgisi