Ondokuz Mayıs Üniversitesi

Bilgisayar Mühendisliği

Aleyna KAHRAMAN

20060355

Er Diyagramları ve Normalizasyon Kuralları

Kübra SEYHAN

ÖZET

Bu rapor, veritabanı tasarımı ve veri normalizasyonu konularını kapsayan bir föyden bahsetmektedir. Başlangıçta, bir fabrikanın işleyişini modelleyen bir UML diyagramıyla başlamaktadır. Bu diyagram, fabrikanın bileşenlerini, süreçlerini ve kaynaklarını tanımlamaktadır, böylece işletmenin yapısal yapısını anlamamıza yardımcı olmaktadır.

Ardından, fabrika modeli Microsoft SQL Server'da yeniden oluşturulmaktadır. Bu adım, fabrikanın UML diyagramındaki varlıklarını ve ilişkilerini ilişkisel bir veritabanında nasıl temsil edebileceğimizi göstermektedir. İlişkisel veritabanı tasarımı, verileri organize etmek ve veritabanı işlemlerini yönetmek için temel bir unsurdur.

Sonraki adımda, bir otelle ilgili bir UML diyagramı çizilmektedir. Bu diyagram, otel müşterilerini, odalarını, rezervasyonlarını ve diğer önemli bileşenlerini göstermektedir. Otel işletmesinin yapısını ve işleyişini anlamak için bu diyagram önemlidir.

Föyün sonunda, verilen bir tablo 3NF (Üçüncü Normal Form) formuna dönüştürülmektedir. Bu adım, veritabanının normalizasyon sürecini temsil etmektedir. Normalizasyon, veritabanının veri bütünlüğünü sağlamak ve veri tekrarını azaltmak için kullanılan bir tekniktir. 3NF formuna dönüşüm, gereksiz veri tekrarlarını ortadan kaldırarak veritabanının daha iyi performans göstermesine yardımcı olmaktadır.

Tüm bu adımlar, bir işletmenin veya organizasyonun verilerini etkin bir şekilde yönetmek ve veri bütünlüğünü sağlamak için izlenen önemli adımlardır. Veritabanı tasarımı ve normalizasyon,

veri tabanlı sistemlerin etkin ve güvenilir bir şekilde çalışmasını sağlayan temel prensipleri içermektedir.

BÖLÜM: İ

GİRİŞ

1

AMAÇ

Bu raporun amacı, veritabanı tasarımı ve veri normalizasyonu konularını açıklamak ve uygulamak için bir rehber sağlamaktır. Rapor, öncelikle bir fabrikanın işleyişini modellemek için UML diyagramlarının nasıl kullanılabileceğini ve bu yapıların ilişkisel veritabanlarında nasıl uygulanabileceğini göstermektedir. Ardından, bir otelin yapısal yapısını ve işleyişini anlamak için benzer bir yaklaşım kullanılmaktadır.

Bu raporun bir diğer amacı da, veritabanı tasarımının önemini vurgulamak ve normalizasyonun veri bütünlüğünü sağlama ve veri tekrarını azaltma konusundaki önemini anlatmaktır. 3NF formuna dönüşüm süreci, gereksiz veri tekrarlarını azaltarak veritabanının daha etkin ve performanslı hale gelmesine yardımcı olur.

Son olarak raporun amacı, işletmeler veya organizasyonlar için veritabanı tasarımı ve normalizasyonun önemini anlatarak, bu konuları anlamalarını ve uygulamalarını teşvik etmektedir. Veritabanı tasarımı ve normalizasyon, verilerin etkin yönetimi ve veri bütünlüğünün korunması için kritik bir öneme sahiptir ve bu sunum, bu konuları anlatarak bu önemi vurgulanmaktadır.

BÖLÜM: İİ

MATERYAL

2

UYGULAMALAR

Microsoft SQL Server 2017 Kurulumu ve SQL Server Management Studio Yüklemesi yapılmıştır.

2.1 *Microsoft SQL Server 2017*

Microsoft SQL Server 2017, işletmelerin veri yönetimi, analizi, raporlama ve iş zekası gibi çeşitli gereksinimlerini karşılayan bir ilişkisel veritabanı yönetim sistemidir. Büyük miktarlarda veri depolama, veri güvenliği sağlama, veri analizi yapma, iş zekası çözümleri oluşturma ve uygulama geliştirme için kullanılır. Ayrıca bulut tabanlı platformlarla entegrasyon sağlayarak ölçeklenebilirlik ve esneklik sunar..

2.2 *SQL Server Management Studio (SSMS)*

SQL Server Management Studio (SSMS), Microsoft SQL Server'ı yönetmek, veritabanı nesnelerini yönetmek, sorguları çalıştırmak, veri işlemek ve performansı izlemek için kullanılan

entegre bir geliştirme ortamıdır. Veritabanı yöneticileri ve geliştiriciler için temel bir araç olan SSMS, SQL Server'a erişimi kolaylaştırır ve yönetimi daha verimli hale getirir.

BÖLÜM: İİİ

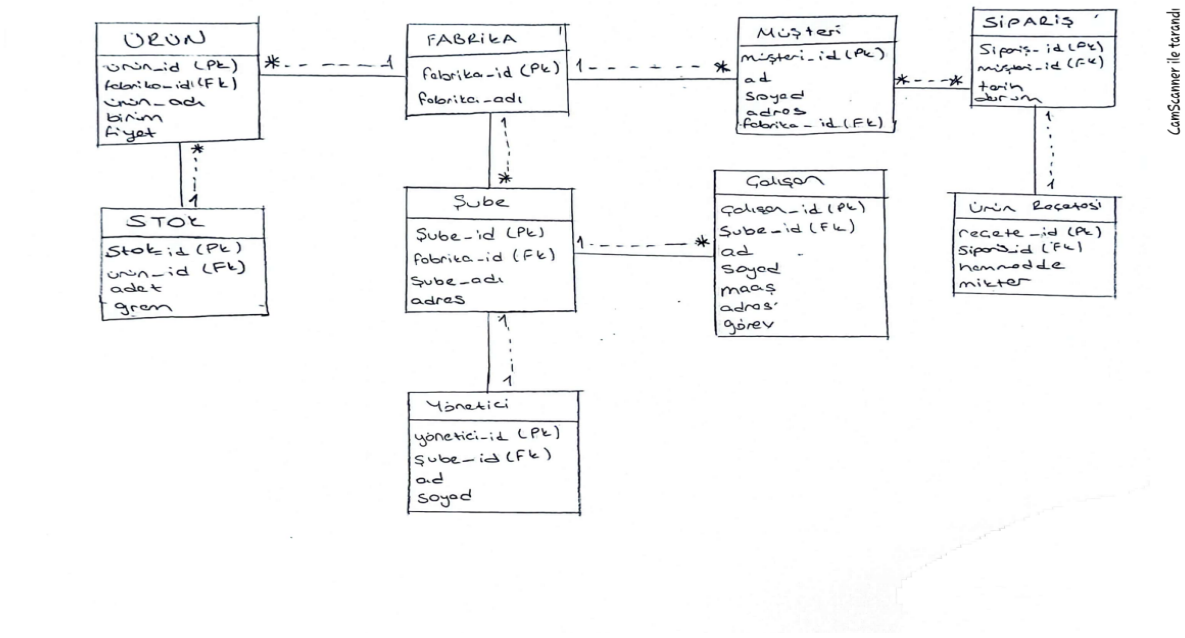
YÖNTEM

3

ER DİAGRAMLARINI VE NORMALİZASYON

Bu rapor, ER diagramlarını ve normalizasyon işlemlerini kapsamaktadır.

3.1 *Fabrika UML Diagramının Elle Çizilmesi*



Şekil 1: Fabrika UML Diagramı

Bu veritabanının UML diagramını çizmek için öncelikle ilişkileri ve tablolar arasındaki bağlantıları anlamak gerekir. İlişkisel veritabanlarında, tablolar arasındaki ilişkiler genellikle

anahtarlar (primary keys) ve dış anahtarlar (foreign keys) aracılığıyla tanımlanır.

Fabrika varlık kümesi oluşturulmuş ve bu varlık kümesine `fabrika_id` ve `fabrika_adı` nitelikleri eklenmiştir. `fabrika_id` primary key olarak belirtilmiştir. Şube varlık kümesine `sube_id`, `sube_adı` ve adres nitelikleri eklenmiştir. `sube_id` primary key ve `fabrika_id` dış anahtarı kullanılarak fabrika ile aralarında ilişki kurulmuştur. Fabrikanın birden çok şubesi olabileceğinden dolayı aralarında birden çoğa ilişki kurulmuştur. Yönetici varlık kümesine `yönetici_id`, ad ve soyad nitelikleri eklenmiştir. `yönetici_id` primary key ve `sube_id` dış anahtarı kullanılarak sube ile aralarında ilişki kurulmuştur. Şubenin bir yöneticisi olabileceğinden dolayı aralarında birden bire ilişki kurulmuştur.

Müşteri varlık kümesi oluşturulmuş ve bu varlık kümesine `müşteri_id`, ad, soyad ve adres nitelikleri eklenmiştir. `müşteri_id` primary key olarak belirtilmiştir. `müşteri_id` primary key ve `fabrika_id` dış anahtarı kullanılarak fabrika ile aralarında ilişki kurulmuştur. Fabrikanın birden çok müşterisi olabileceğinden dolayı aralarında birden çoğa ilişki kurulmuştur. Sipariş varlık kümesine `sipariş_id`, tarih ve durum nitelikleri eklenmiştir. `sipariş_id` primary key ve `müşteri_id` dış anahtarı kullanılarak müşteri ile aralarında ilişki kurulmuştur. Müşterinin birden çok siparişi olabileceğinden dolayı aralarında çoktan çoğa ilişki kurulmuştur. Ürün Reçetesi varlık kümesi oluşturulmuştur ve `reçete_id`, hammadde ve miktar nitelikleri eklenmiştir. `reçete_id` primary key olarak belirtilmiştir. `reçete_id` primary key ve `sipariş_id` dış anahtarı kullanılarak sipariş ile aralarında ilişki kurulmuştur. Siparişin bir reçetesi olabileceğinden dolayı aralarında birden bire ilişki kurulmuştur.

Çalışan varlık kümesi oluşturulmuş ve bu varlık kümesine `çalışan_id`, ad, soyad, maaş, görev ve adres nitelikleri eklenmiştir. `çalışan_id` primary key olarak belirtilmiştir. `çalışan_id` primary key ve `şube_id` dış anahtarı kullanılarak şube ile aralarında ilişki

kurulmuştur. Şubede birden çok çalışan olabileceğinden dolayı aralarında birden çoğa ilişki kurulmuştur.

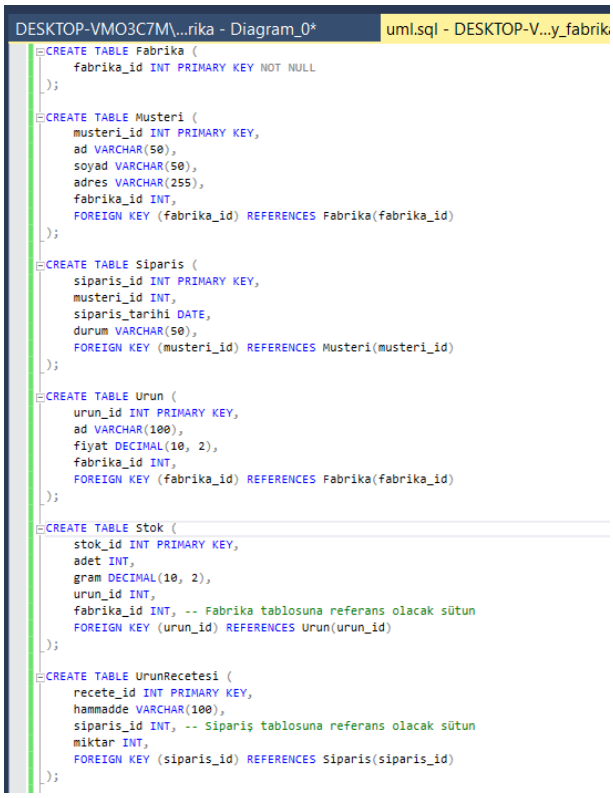
Ürün varlık kümesi oluşturulmuş ve bu varlık kümesine `ürün_id`, `ürün_adı`, birim ve fiyat nitelikleri eklenmiştir. `ürün_id` primary key olarak belirtilmiştir. `ürün_id` primary key ve `fabrika_id` dış anahtarı kullanılarak fabrika ile aralarında ilişki kurulmuştur. Fabrikada birden çok ürün olabileceğinden dolayı aralarında birden çoğa ilişki kurulmuştur. Stok varlık kümesine `stok_id`, adet ve gram nitelikleri eklenmiştir. `stok_id` primary key ve `ürün_id` dış anahtarı kullanılarak ürün varlık kümesi ile arasında ilişki kurulmuştur. Stokta birçok ürün bulunabileceğinden dolayı stok-ürün ilişkisi birden-çoğa olarak kurulmuştur.

3.2 *SQL Server ER Diagramı*

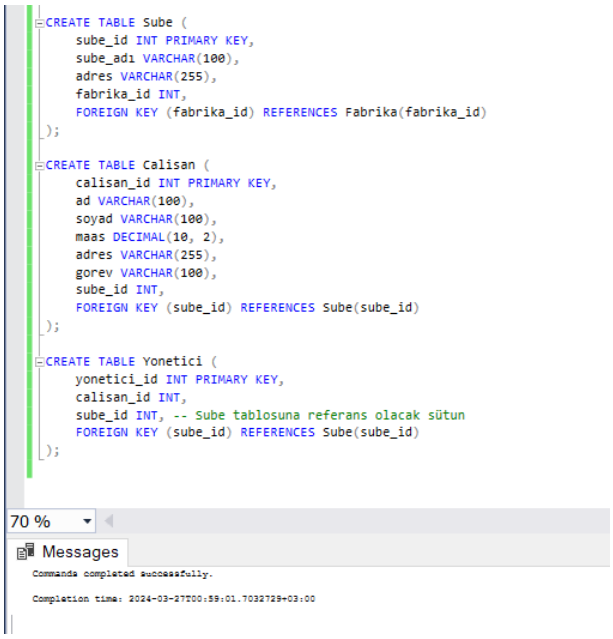
Microsoft SQL Server'da kod ile tablolar ve bağlantılar oluşturulmuştur.

Bu veritabanı şeması, bir işletmenin faaliyetlerini yönetmek için tasarlanmıştır. İşletme içindeki temel bileşenleri ve bu bileşenler arasındaki ilişkileri tanımlar. Tablolar, işletmenin farklı alanlarını temsil ederken, ilişkiler bu alanlar arasındaki bağlantıları gösterir.

Fabrika tablosu, işletmenin üretim merkezlerini temsil eder. Her bir fabrika, benzersiz bir kimlikle `fabrika_id` belirlenir. Müşteri tablosu, işletmenin müşterilerini saklar. Her müşteriye ait ad, soyad ve adres gibi bilgiler yanı sıra hangi fabrikaya bağlı olduklarını belirten bir `fabrika_id` içerir. Sipariş tablosu, müşterilerin verdiği siparişlerin detaylarını içerir. Her sipariş, bir müşteriye `musteri_id` ve siparişin alındığı tarihe sahiptir.



Şekil 2: SQL SERVER KOD-1



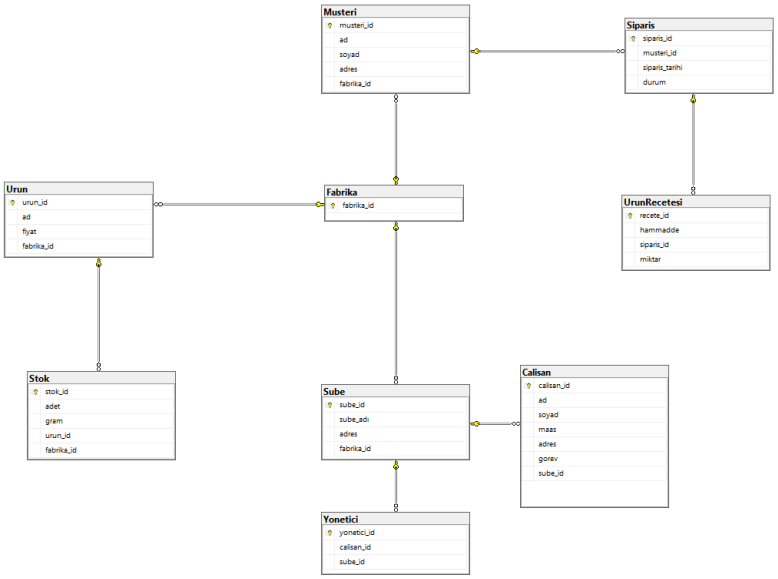
Şekil 3: SQL SERVER KOD-2

Ürün tablosu, işletmenin ürettiği veya sattığı ürünleri tanımlar. Ürünlerin adı, fiyatı ve hangi fabrikada üretildiği `fabrika_id` bilgilerini içerir. Stok tablosu, ürünlerin mevcut

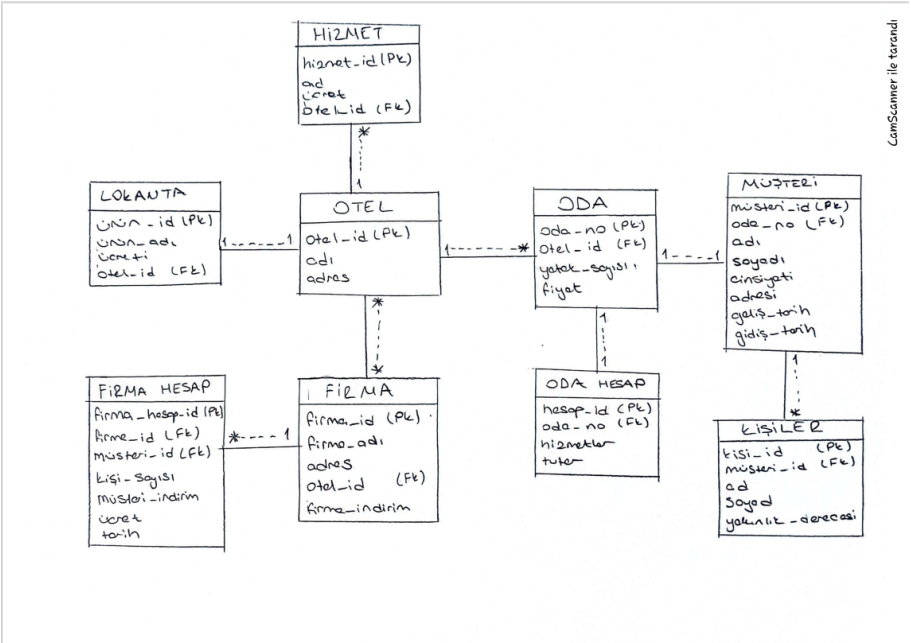
miktarlarını ve ağırlıklarını takip eder. Her stok girişi, bir ürüne `urun_id` ve hangi fabrikada bulunduğuna `fabrika_id` işaret eder.

Ürün Reçetesi tablosu, ürünlerin üretiminde kullanılan hammaddelerin listesini saklar. Her reçete, bir siparişe `siparis_id` ve kullanılan hammaddelerin miktarına sahiptir. Şube tablosu, işletmenin farklı şubelerini temsil eder. Şubelerin adı, adresi ve hangi fabrikaya bağlı oldukları `fabrika_id` bilgisini içerir.

Çalışan tablosu, işletmedeki personeli tanımlar. Her çalışanın adı, soyadı, maaşı, görevi ve hangi şubede çalıştığı `sube_id` bilgilerini içerir. Yönetici tablosu, işletmedeki yöneticileri temsil eder. Her yönetici, bir çalışana `calisan_id` ve hangi şubeyi yönettiğine `sube_id` işaret eder.



Şekil 4: UML



Şekil 5: Otel ER Diagramı

3.3 *Otel ER Diagramı*

Otel varlık kümesi, `otel_id`, ad ve adres niteliklerinden oluşmaktadır. Her bir otel, benzersiz bir kimlikle `otel_id` belirlenir. Firma varlık kümesi, `firma_id`, firma_adı, firma_indirim ve adres niteliklerinden oluşmaktadır. Her bir firma, benzersiz bir kimlikle `firma_id` belirlenir. Otelin birden çok firma ile anlaşması olabileceğinden dolayı firma ile otel arasında çoktan çoğa ilişki kurulmuştur. Firma hesap varlık kümesi, `firma_hesap_id`, kişi sayısı, müşteri_indirim, ücret ve tarih niteliklerinden oluşmaktadır. Her bir firma hesap, benzersiz bir kimlikle `firma_hesap_id` belirlenir. Firma ile firma_hesap arasında birden çoğa ilişki kurulmuştur.

Lokanta varlık kümesi `ürün_id`, ürün_adı ve ücret niteliklerinden oluşmaktadır. Her bir lokanta, benzersiz bir kimlikle `ürün_id` belirlenir. Otelin bir lokantası olduğundan dolayı otel ile lokanta arasında bire bir ilişki kurulmuştur.

Hizmet varlık kümesi `hizmet_id`, ad ve ücret niteliklerinden oluşmaktadır. Her bir hizmet, benzersiz bir kimlikle `hizmet_id` belirlenir. Otelin birçok hizmeti olabileceğinden dolayı otel ile hizmet arasında birden çoğa ilişki kurulmuştur.

Oda varlık kümesi `oda_no`, yatak_sayısı ve fiyat niteliklerinden oluşmaktadır. Her bir oda, benzersiz bir kimlikle `oda_no` belirlenir. Otelin birçok odası bulunabileceğinden dolayı otel ile oda arasında birden çoğa ilişki kurulmuştur. Oda hesap varlık kümesi, `hesap_id`, hizmetler ve tutar niteliklerinden oluşmaktadır. Her bir oda hesabı, benzersiz bir kimlikle `hesap_id` belirlenir. Her odanın bir hesabı olacağından oda ile hesap arasında bire bir ilişki kurulmuştur.

Müşteri varlık kümesi `müşteri_id`, adı, soyadı, cinsiyeti, adresi, geliş ve gidiş tarihi niteliklerinden oluşmaktadır. Her bir müşteri, benzersiz bir kimlikle `müşteri_id` belirlenir. Odanın bir müşterisi olacağından dolayı aralarında bire bir ilişki kurulmuştur. Kişiler varlık kümesi odada bulunan birden çok müşteri için eklenmiştir. `kisi_id`, ad, soyad ve yakınlık derecesi niteliklerinden oluşmuştur. Bir müşterinini yanında birden çok kişi kalabileceğinden dolayı aralarında birden çoğa ilişki kurulmuştur.

3.4 *Normalizasyon*

Üçüncü Normal Form (3NF), veritabanı tasarımında önemli bir normalleştirme adımıdır. Bu aşama, veritabanı tablolarının yapılarını optimize etmek ve veri bütünlüğünü sağlamak için kullanılan bir tekniktir. Bir tabloyu 3NF'ye dönüştürmek, veri tekrarını minimize ederek veritabanının daha etkili ve tutarlı olmasını sağlar. Bu, veri tabanının güncellenmesini, silinmesini veya sorgulanmasını daha kolay hale getirir ve aynı zamanda veri bütünlüğünü artırır.

3NF'ye dönüştürme işlemi, veri tabanının performansını da artırabilir. Çünkü gereksiz veri tekrarı azaltılır ve gereksiz sütunlar kaldırılır. Bu da sorgulama ve veritabanı işlemlerinin daha hızlı gerçekleştirilmesine olanak sağlar. Bu nedenlerle, 3NF'ye dönüştürme, veritabanı tasarımında kritik bir adımdır ve veritabanının etkinliğini, tutarlılığını ve performansını artırır.

3.4.1 *1NF*

1NF (First Normal Form), her bir sütunda atomik (bölünemez) değerler içeren bir tabloyu ifade eder. Bir tablo 1NF'ye dönüştürüldüğünde, her hücre yalnızca bir değeri içermelidir ve bu değer daha fazla parçalara bölünemez.

öğrenci-no	ad	Şehir	Bölge	Ders	kayıt Tarihi
100	Ahmet	İstanbul	Marmara	VTYS	01.01.2017
100	Ahmet	İstanbul	marmara	mobil	20.05 2017
101	Deniz	Ankara	İç Anadolu	VTYS	21.05.2017
101	Deniz	Ankara	İç Anadolu	WEB	21.05 .2017
102	Ali	kars	Doğu Anadolu	Mobil	22.05. 2017
103	Yasemin	İstanbul	Marmara	VTYS	03.06.2017

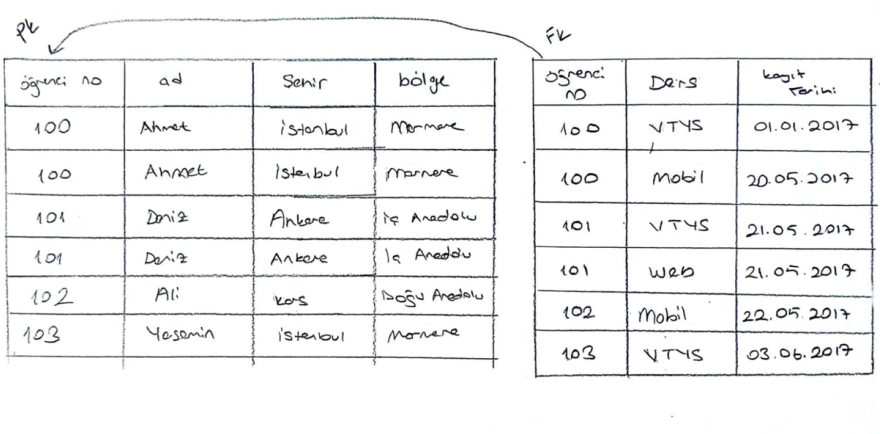
Şekil 6: 1NF

Aynı satırda bulunan 2 ders farklı satırlara yerleştirilmiştir.

3.4.2 *2NF*

İkinci Normal Form (2NF), her sütunun tablonun anahtar alanlarına tam olarak bağımlı olduğu bir düzenlemeyi ifade eder. Bu, kısmi bağımlılıkların ortadan kaldırıldığı ve veri

tablosunun daha tutarlı hale getirildiği anlamına gelir. 2NF'ye dönüştürme işlemi, veri bütünlüğünü sağlamak ve veritabanı yapısını optimize etmek için önemlidir.



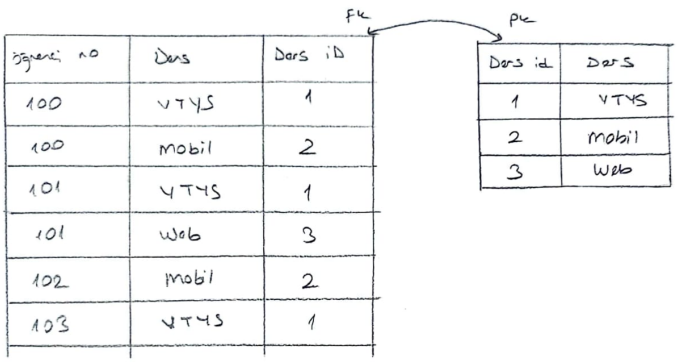
Şekil 7: 2NF

1NF tablosu öğrenci ve kayıt olacak şekilde 2 ayrı tabloya bölündü.

3.4.3 *3NF*

Üçüncü Normal Form (3NF), veritabanı tasarımındaki önemli bir normalizasyon aşamasını ifade eder. Bu aşama, bir tablonun ikinci normal forma (2NF) uygun olmasının yanı sıra, herhangi bir transitive bağımlılığın da ortadan kaldırıldığı bir düzenlemeyi ifade eder. 3NF'ye göre, bir tablo herhangi bir transitive bağımlılığa sahip olmamalıdır, yani her sütun tablonun birincil anahtarlarına doğrudan bağımlı olmalıdır.

Veri tekrarını azaltmak için kayıt tablosundan ders_id kullanılarak yeni tablolar oluşturuldu.


FK
PK
öğrenci no
Ders
Ders iD
100
VTYS
1
100
mobil
2
101
VTYS
1
101
Web
3
102
mobil
2
103
VTYS
1
Ders id
Ders
1
VTYS
2
mobil
3
Web

Şekil 8: 3NF

BÖLÜM: İV

SONUÇ

4

SONUÇ

Bu raporun sonucunda, çizilen UML diyagramları da önemli bir rol oynamıştır. İlk olarak, fabrika UML diyagramı, işletmenin yapısını ve işleyişini görsel olarak temsil etmiştir. Bu diyagram, fabrikanın bileşenlerini, iş süreçlerini ve kaynaklarını tanımlamış ve bu unsurlar arasındaki ilişkileri açıklamıştır. Fabrika UML diyagramı, işletmenin genel yapısını anlamak için bir kılavuz sağlamıştır.

Ardından, Microsoft SQL Server'da yeniden oluşturulan versiyon, fabrika UML diyagramını ilişkisel bir veritabanında nasıl uygulayabileceğimizi göstermiştir. UML diyagramındaki varlıklar, tablolara dönüştürülmüş ve bu tablolar arasındaki ilişkiler, birincil ve dış anahtarlar kullanılarak ifade edilmiştir. Bu adım, işletmenin verilerini etkin bir şekilde yönetmek için veritabanı tablolarının nasıl yapılandırılabileceğini göstermiştir.

Daha sonra, otel UML diyagramı çizilmiştir. Bu diyagram, bir otelin müşterilerini, odalarını, rezervasyonlarını ve diğer önemli bileşenlerini göstermiştir. Otel UML diyagramı, otelin işleyişini ve bileşenler arasındaki ilişkileri anlamak için kullanılmıştır. Bu diyagram, otel işletmesinin yapısını ve işleyişini net bir şekilde görselleştirmiştir.

Son olarak, verilen tablonun 3NF formuna dönüştürülmesi, veri normalizasyonunun bir örneğini sunmuştur. Bu adım, gereksiz veri tekrarlarını azaltarak ve veritabanının yapısal

bütünlüğünü güçlendirerek veri yönetimini iyileştirmiştir. Veri tabanı tasarımı ve normalizasyon, veri tabanı yapılarını ve ilişkilerini anlamak için kullanılan UML diyagramları ile birlikte bir bütün oluşturmuştur.