

BÖLÜM 5

SİSTEM GEREKSİNİMLERİNİN BELİRLENMESİ

ANAHTAR KAVRAMLAR

Kullanım Durum, Kavramsal Veri Modeli,
Veri Modelleme, Mantıksal Veri Modeli,
Aktörler- Durumlar- İlişkiler, Fiziksel Veri Modeli



ÖĞRENME HEDEFLERİ

-  Projenin kullanım durumlarına ilişkin kavramları açıklar
-  Veri modelleme yönetimlerini kavrar
-  Geliştireceği sistemin kullanım durumlarını oluşturabilir

GİRİŞ

Sistem analistleri, sistem geliştirmenin analiz aşamasında verilerin nasıl oluşturulacağı, nasıl depolanacağı, veriler üzerinde hangi işlemlerin gerçekleştirileceği gibi teknik ayrıntıya girmeden, sistemi genel hatlarıyla ifade edebilmek için model oluştururlar. Veri modelinde sistemin görsel bir temsili oluşturulur. Benzer biçimde, sistemin gereksinimlerini modelleyebilmek için kullanım durum kullanılmaktadır. Kullanım durumlar, sistem mimarisini görselleştirmenin önemli bir unsuru olarak karşımıza çıkmaktadır. Veri modelleme ise veri noktaları ve yapıları arasındaki bağlantıları iletmek için sistem parçalarının görsel bir temsili oluşturma sürecidir. Bu ünite, kullanım durumun ve bileşenlerinin neler olduğu, ne amaçla kullanıldığı, veri modelleme kavramı, veri modeli çeşitleri ve veri modelleme türleri konularına değinilmiştir.



1. KULLANIM DURUMLARI (USE CASES)

Sistem tasarımında bir müşterinin bir ürün, hizmet veya sistemle nasıl etkileşeceğini ifade etmek, gereksinimlerin toplanması ve üst düzey paydaş iletişimi için önemlidir. Kullanım durumları, bir sistemin gereksinimlerini yakalamak, modellemek ve belirtmek için kullanılan bir tekniktir (Bittner, 2003). Bir kullanım durumu, sistemin aktörleriyle etkileşim içinde gerçekleştirebileceği ve hedeflerine katkıda bulunan gözlemlenebilir bir sonuç üreten bir dizi davranışa karşılık gelir. Aktörler, insan kullanıcıların veya diğer sistemlerin etkileşimdeki rolünü temsil eder. Gereksinim analizinde, bir kullanım durumu tanımlanırken, birincil aktörü için temsil ettiği belirli kullanıcı hedefine göre adlandırılır. Vaka, özel koşullar, istisnalar veya hata durumları gibi varyantların yanı sıra genel faaliyet ve olay sırasını açıklayan metinsel bir açıklama

veya ek grafik modellerle daha da detaylandırılır. Kullanım durum diyagramı bir ürünü veya hizmeti kullananların o ürün veya hizmetle nasıl etkileşime gireceklerinin ve ürün veya hizmetin ne yaptığının görsel olarak temsilidir. Bir kullanım durumu, başarı senaryolarını, başarısızlık senaryolarını ve kritik varyasyonları veya istisnaları oluşturabilir.

Kullanım Durumuyla ilgili ilk çalışma, İsveçli bilgisayar bilimcisi Ivar Jacobson tarafından 1987 yılında yapılmıştır. Ivar Jacobson senaryoyu sistem ile kullanıcı arasında gerçekleştirilen etkileşimli diyalog olarak tanımlamaktadır. Jacobson daha sonra iki yazılım mühendisi ile birlikte bir sistem tasarımını görselleştirmenin standart yolunu sunan Birleşik Modelleme Dili'ni (UML) oluşturmuştur.

1.1. Kullanım Durumlarının Amaçları

- Kapsam yönetimi
- Gereksinimlerin belirlenmesi
- Kullanıcının sistemle etkileşime gireceği yolların ana hatlarıyla belirtilmesi
- Sistem mimarisini görselleştirme
- Teknik gereksinimleri iş paydaşlarına iletme
- Risk yönetimi

1.2. Kullanım Durum Modeli

Kullanım durum modeli, bir sistem ile bir aktör arasındaki etkileşimlerin görsel olarak sunumudur. Kullanım durum modelleri iş süreçlerini temsil eder. Bu da ön koşulların ve tetikleyicilerin daha fazla ifade edilmesine yardımcı olur.

Bir kullanım durum modeli genellikle UML ile ifade edilir. Bu modelde dört ana bileşen bulunmaktadır: aktörler, sistem, kullanım durumları ve ilişkiler.

UML içerisinde sistem, bir dikdörtgen ile gösterilir. Aktörler dikdörtgenin dışında çubuktan insanlar olarak gösterilirken, kullanım durumları ise dikdörtgenin içerisinde ovalerde metin olarak sunulurlar. Düz ve kesikli çizgiler sistem

ile aktörlerin kullanım durumları arasındaki ilişkiyi temsil etmektedir (Daly, 2022).

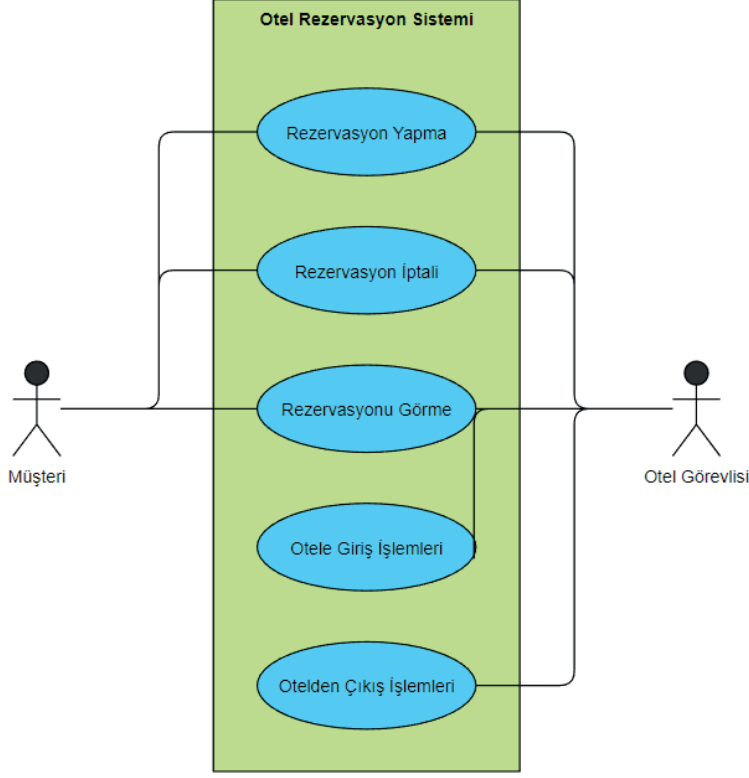
Kullanım durum diyagramı ise, temel olarak bir tür kullanım durum modelidir. Kullanım durum diyagramları aşağıda belirtilen durumlar için kullanılır:

- Sistemin akışını ve davranışını görselleştirmek
- Sistemin işlevselliğini örneklemek
- Temel sistem-kullanıcı etkileşimlerini temsil etmek.

Kullanım durum diyagramının amacı, oluşturulan sistemin hangi işlevinin hangi aktörler

tarafından gerçekleştirileceğini ve kullanım durumlarının kendi arasındaki ilişkilerini göstermektedir.

Kullanım durum diyagramına ait bir örnek Şekil 1’de gösterilmiştir:



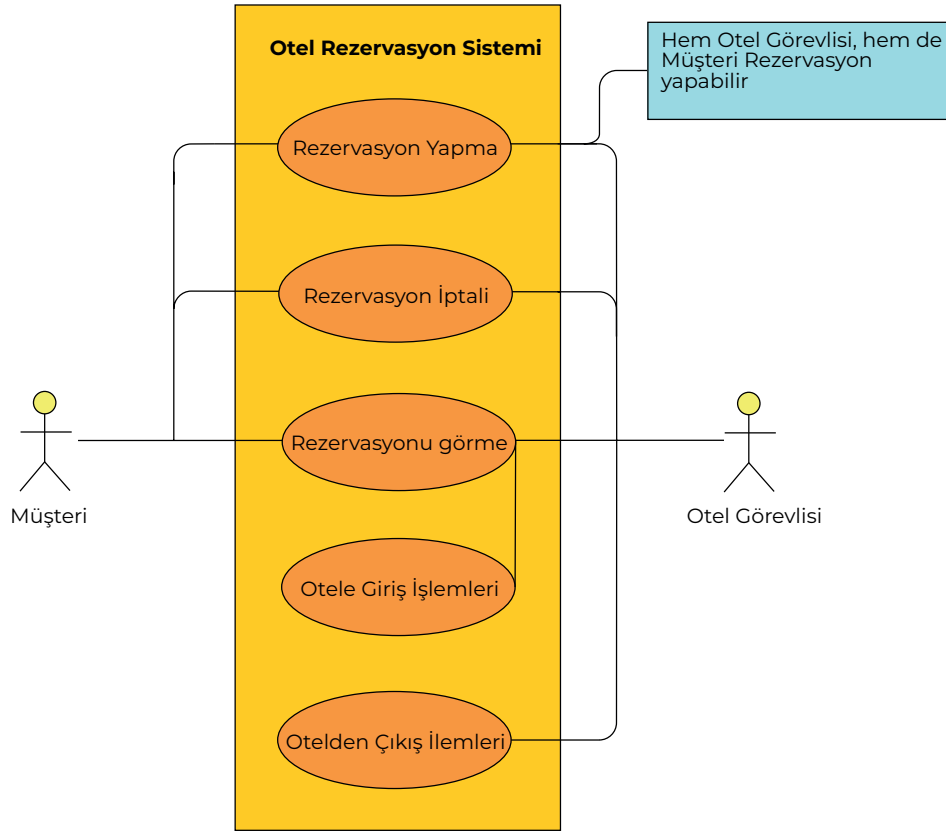
Şekil 1. Otel Rezervasyon Sistemine Ait Kullanım Durum Diyagramı

Aktör, bir sistemin dışında onunla etkileşime giren herhangi bir şeydir. Örneğin, insanların internet üzerinden mal satın almalarını sağlayan bir sistemde, insan kullanıcılar önemli aktörler olacaktır, ancak kullanıcıların alışverişleri için ödeme yapmalarını sağlayan kredi kartı sistemi de aynı şekilde bir aktör olacaktır (Jackson, 1995).

Kullanım durumları, sistemin gerçekleştirdiği bazı görevleri veya tutarlı işlevsellik birimlerini temsil eder. Bir aktör figürü ile bir kullanım durumu ovalini birleştiren çizgi, aktör ile kullanım durumu arasındaki iletişimi temsil eden bir iliş-

kiyi gösterir. Bu, aktörün görevin yerine getirilmesine dahil olabileceği ama dahil olmak zorunda olmadığı anlamına gelir.

Kullanım durum diyagramları yalnızca fonksiyonel gereksinimlerle (sistemin yapması gereken şeyler) ilgilenir, ancak bir kullanım durum diyagramı geliştirirken ve işlevsel olmayan gereksinimlerle karşılaşıldığında, kullanım durum diyagramına açıklayıcı bir not eklenerek bunların kaydedilmesi gerekir. Bu durumla ilgili bir örnek, Şekil 2’de gösterilmiştir.



Şekil 2. Kullanım Durum Diyagramında Açıklayıcı Not Kullanımı

UML’de kullanım durumlarını çevreleyen bir dikdörtgenle gösterilen sistem sınırı, sistemi diğer yapılardan ayıran önemli bir kavramsal çizgidir. Bir kullanım durum diyagramında temsil edilen sistemin etrafına sınır çizilerek çözümün kapsamı belirlenmiş olur. Genel kullanım durum diyagramlarında sınır dikdörtgeni çoğunlukla gösterilmez.

Şekil 1 ve Şekil 2 aynı zamanda yaygın bir problemi de ortaya koymaktadır. Yalnızca diyagramdan, iki aktörden hangisinin (Müşteri veya Otel Görevlisi) rezervasyon yapma kullanım senaryosunu başlattığı net değildir. Kullanım durumu başarıyla tamamlamak için her iki aktöre de ihtiyaç duyuluyor olabilir. UML bu sorunu ortadan kaldırmak için çeşitli yollar sunmaktadır. Bunların en basit olanı, UML içerisinde açıklayıcı not kullanmaktır.

Senaryo, kullanım durumlarını gösteren bir dizi eylemin açıklamasıdır. Bir sistemin kullanıcıları (aktörleri) ile sistemin kendisi arasındaki

etkileşimi ve diyalogu tanımlar. Aşağıda iki örnek senaryo görülmektedir:

1. Yunus isimli müşteri, 30 Mart için X Otel’de bir oda ayırtmak istemektedir. O tarih için bir oda müsaittir ve otel görevlisi Yunus için rezervasyon yapar.

2. Azize isimli müşteri, Haziran ayının ilk haftası için Y Otel’de bir oda ayırtmak istemektedir. Haziran ayında yedi gün için boş oda yoktur, ancak dört gün için bir oda ve ardından üç gün için aynı tipte başka bir oda mevcuttur. Otel görevlisi bu seçeneği Azize’ye sunar fakat Azize bu teklifi reddeder.

Her iki senaryo da rezervasyon yapma kullanım durumunun olası örnekleridir. Etkileşimleri ve sonuçları farklıdır. İlkinde, kullanım durumunun başarılı bir sonuca götüren bir açıklaması vardır. İkincisinde ise ana başarı senaryosunda bir istisna vardır. Bir kullanım durumu için normal davranışın istisnaları, özellikle aktörlerin bir kullanım durumunu tamamlamadan iptal etme-

ye karar verdiği durumlarda yaygındır. Bununla birlikte, tüm senaryolar arasındaki ortak tema, bir aktörün bir kullanım senaryosu tarafından tanımlanan hedefe ulaşma niyetidir. Yukarıdaki başarısız senaryoda Azize, Y Otel’de rezervasyon yaptırmaya çalışıyordu. Belki de kaldığı süre boyunca oda değiştirme fikrinden hoşlanmamıştı. Bu nedenle, bir kullanım durumu olağandışı veya alternatif senaryoları içermelidir.

Bir araştırmaya sadece bir kullanım durumu ve onun ana başarı senaryosu belirlenerek başlanabilir ve daha sonra bu durum geliştirilebilir. Sadece ana başarı senaryosu için değil, ilgili

tüm senaryolar için her bir kullanım durumuna ilişkin bilgilerin nasıl kaydedileceğine karar verilmesi gerekecektir. En basitinden, her bir kullanım durumu için her bir senaryoyu sonucuyla birlikte detaylandıran metinsel bir açıklama kaydedilebilir.

Her gereksinim, geliştirici ile müşteri arasındaki sözleşmenin bir parçasıdır. Her iki tarafın da her bir kullanım senaryosunda neyin ifade edildiği ve ne anlama geldiği konusunda net bir anlayışa sahip olması gerekir. Aşağıdaki tablo, bir kullanım senaryosunu yapılandırmanın ve kaydetmenin bir yolunu göstermektedir.

Tablo 1. Otel Rezervasyon Sisteminin Metinsel Açıklaması

Kullanım Durum Adı	Rezervasyon Yapma İşlemi
Aktörler	Otel Görevlisi
Amaç	Otel odasının rezerve edilmesi
Ön Koşul	Yok (yani, bu kullanım senaryosunu gerçekleştirmeden önce yerine getirilmesi gereken hiçbir koşul yoktur).
Son Koşul	İstenilen tipte bir oda, talep edilen dönem için müşteri için rezerve edilmiş olacak ve oda o dönem için dolu olacaktır.
Varsayımlar	Bir müşteri internet üzerinden de rezervasyon yapabilir.

1.3. Temel Başarı Senaryosu

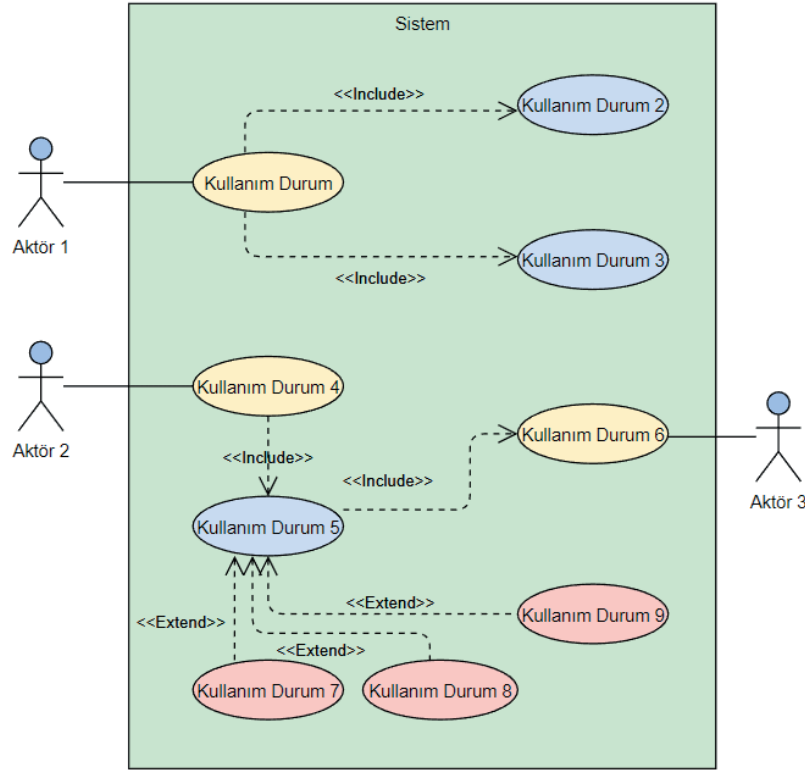
1. Müşteri rezervasyon talebinde bulunur.
2. Müşteri istediği oteli, tarihleri ve oda tipini seçer.
3. Otel Görevlisi, talep için müsaitlik durumunu ve fiyatı sağlar.
4. Müşteri teklife devam etmeyi kabul eder.
5. Müşteri, otelin kayıtları için kimlik ve iletişim bilgilerini verir.
6. Müşteri, ödeme bilgilerini verir.
7. Otel Görevlisi bir rezervasyon oluşturur ve ona bir numara verir.
8. Otel Görevlisi rezervasyon numarasını müşteriye açıklar.
9. Otel Görevlisi rezervasyonun onayını oluşturur ve müşteriye gönderir.

Bazı durumlarda kullanım durumların, işlevini tamamlayabilmesi için başka kullanım durumlarıyla birlikte kullanılması gerekir. Yani mevcut kullanım durum, başka kullanım durumlar olmaksızın tamamlanamaz. Örneğin, bir alışveriş uygulamasından sipariş verebilmesi için öncelikle sipariş vermek istediği firmayı seçip, o firmaya ait ürünleri görüntülemeli ve daha sonra satın almak istediği ürünleri sepete eklemelidir. Yani “Sipariş Ver” isimli kullanım durumunun tamamlanabilmesi için “Firma Seç” ve “Sepete Ekle” kullanım durumların çalışması gerekir. Include (içerme) olarak isimlendirilen bu ilişki türü, Şekil 3’de gösterildiği şekilde kullanılmaktadır.

Kullanım durumları arasındaki başka bir ilişki türü ise Extend (genişletme) olarak adlandırılır ve iki kullanım durumu arasında opsiyonel olarak var olabilecek bir durumu ifade etmek için kul-

lanılır. Bu ilişki türü, bir kullanım durumunun kapsamının, başka bir kullanım durum tarafından genişletilebilmesinin mümkün olduğu durumlar için geçerlidir. Extend (genişletme) ilişki türü, Şekil 3’de gösterildiği şekilde kullanılmaktadır.

Diyagramlar üzerindeki okların yönü, kullanım durumlar arasındaki bağımlılığın yönünü gösterir.



Şekil 3. Kullanım Durumlar için Include ve Extend İlişkisinin Kullanımı

2. KULLANIM DURUM DİYAGRAMINDA DİKKAT EDİLMESİ GEREKEN UNSURLAR

- Kullanım durum diyagramı oluştururken dikkat edilmesi gereken unsurlardan birisi, kullanım durum isimlendirilmesinde tercih edilen kelimelerdir. Kelimeler sıradan değil, sistemin söz konusu davranışını ve işlevini yansıtacak şekilde seçilmelidir. Kullanım durumlar isimlendirilirken genellikle emir kipi kullanılır.
- Aktörler için de ilgili kullanıcı gruplarını yansıtan isimler tercih edilmelidir.

- Include ve Extend ilişkileri sadece gerekli olduğu durumlarda kullanılmalı, diyagram oluşturma temel amacının gereksinimleri belirlemek olduğu unutulmamalı, basitliğin korunmasına mümkün olduğunca dikkat edilmelidir.

- Gerektiğinde bazı hususlara açıklık getirebilmek için diyagrama açıklayıcı notlar eklenmelidir (Alagöz, 2014).

3. VERİ MODELLEME

Veri modelleme, veri noktaları ve yapıları arasındaki bağlantıları iletme için tüm bir bilgi sisteminin veya sistem parçalarının görsel bir temsiliyi oluşturma sürecidir. Amaç, sistem içinde kullanılan ve depolanan veri türlerini, bu veri türleri arasındaki ilişkileri, verilerin gruplandırıl-

ma ve organize edilme yollarını, biçimlerini ve niteliklerini göstermektir.

Veri modelleri ihtiyaçlar etrafında oluşturulur. Kurallar ve gereksinimler, yeni bir sistemin tasarımına dahil edilebilmeleri veya mevcut bir

sistemin yinelenmesi sırasında uyarlanabilmeleri için iş paydaşlarından alınan geri bildirimler yoluyla önceden tanımlanır.

Veriler çeşitli seviyelerde modellenenebilir. Süreç, gereksinimler hakkında bilgi toplanmasıyla başlar. Bu iş kuralları daha sonra somut bir veritabanı tasarımı formüle etmek için veri yapılarına dönüştürülür.

Veri modellemesinde standartlaştırılmış şemalar ve teknikler kullanılır. Bu, veri kaynaklarını tanımlamak ve yönetmek için ortak, tutarlı ve öngörülebilir bir yol sağlar.

Veri Modeli Çeşitleri

Her tasarım sürecinde olduğu gibi, sistem tasarımı da soyut seviyede başlar ve giderek daha somut ve spesifik hale gelir. Veri modelleri genellikle soyutlama derecelerine göre değişen üç kategoriye ayrılabilir. Süreç kavramsal bir modelle başlar, mantıksal bir modele doğru ilerler ve fiziksel bir modelle sonuçlanır.

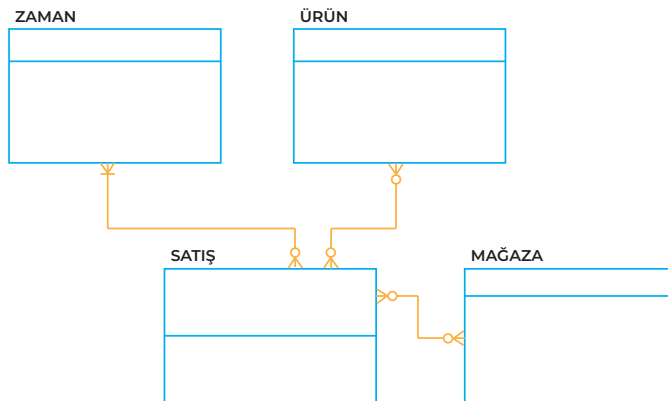
Kavramsal Veri Modeli

Etki alanı modelleri olarak da adlandırılırlar ve sistemin neleri içereceği, nasıl organize edile-

ceği ve hangi iş kurallarının söz konusu olduğuna dair büyük resimli bir görünüm sunarlar. Kavramsal modeller genellikle ilk proje gereksinimlerinin toplanması sürecinin bir parçası olarak oluşturulur. Tipik olarak, varlık sınıflarını, bunların özelliklerini ve kısıtlamalarını, aralarındaki ilişkileri ve ilgili güvenlik ve veri bütünlüğü gereksinimlerini içerirler. Kavramsal veri modelleri, teknik ayrıntılara girmeden modellemekte olduğunuz iş süreçlerini yöneten kavramları ve kuralları görselleştirir (IBM,2020).

Tipik olarak, kavramsal bir model sistemin içeriğinin, organizasyonunun ve ilgili iş kurallarının üst düzey bir görünümünü gösterir. Örneğin, bir e-ticaret işletmesi için veri modeli satıcıları, ürünleri, müşterileri ve satışları içerecektir. Bir iş kuralı, her satıcının en az bir ürün tedarik etmesi gerektiği şeklinde olabilir.

Kavramsal modeller için standart bir format yoktur. Önemli olan, hem teknik hem de teknik olmayan paydaşların veri projelerinin amacı, kapsamı ve tasarımı üzerinde uyum sağlamalarına ve anlaşmalarına yardımcı olmasıdır. Aşağıda örnek bir kavramsal veri modeli sunulmuştur.



Şekil 4. Kavramsal Veri Modeli Örneği

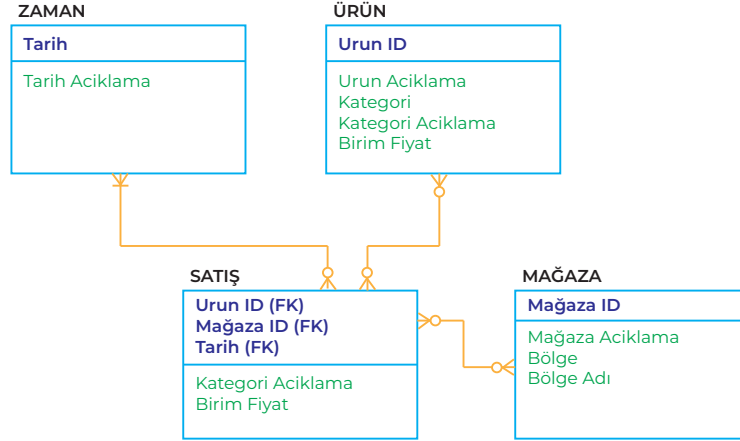
Mantıksal Veri Modeli

Daha az soyutlar ve söz konusu etki alanındaki kavramlar ve ilişkiler hakkında daha fazla

ayrıntı sağlarlar. Mantıksal veri modeli, kavramsal modeli temel alır ve projenin veri öğelerini ve ilişkilerini tanımlar. Mantıksal veri modelleri

herhangi bir teknik sistem gereksinimi belirtmez. Son derece prosedürel uygulama ortamlarında veya veri ambarı tasarımı ya da raporlama sistemi geliştirme gibi doğası gereği veri odaklı olan projeler için faydalı bir modeldir.

Veri mimarları ve iş analistleri, bir veritabanı yönetim sisteminin (bir veritabanındaki verileri depolayan, alan, tanımlayan ve yöneten yazılım) uygulanmasını planlamak için mantıksal veri modelini kullanırlar. Aşağıda mantıksal veri modeline ait bir görsel yer almaktadır.



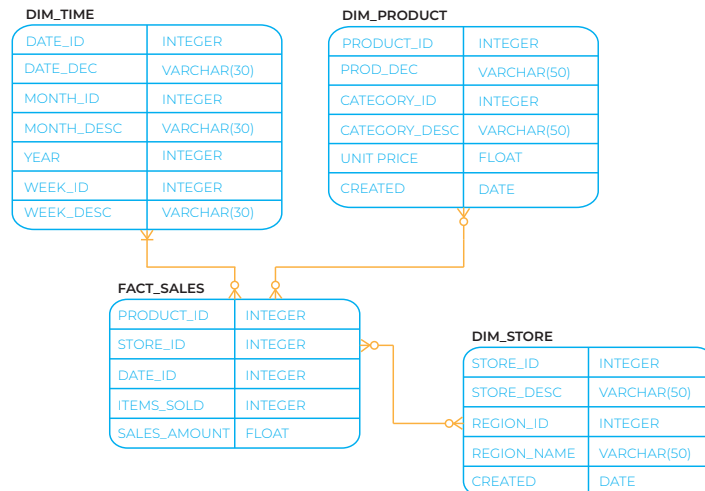
Şekil 5. Mantıksal Veri Modeli Örneği

Fiziksel Veri Modeli

Fiziksel veri modeli tekniktir. Veritabanı analistleri ve geliştiricileri bunu veritabanının ve ilgili veri yapılarının tasarımı için kullanır. Model, teknik veri gereksinimleriyle birlikte depolanacak veri türlerini belirtir.

Veri türü belirtilmelerine örnek olarak, bir verinin tamsayı mı yoksa ondalıklı mı olacağı, veya metinsel bir tipte mi olacağı verilebilir. Teknik gereksinimler arasında depolama ihtiyaçları, erişim hızı ve veri yedekliliği ile ilgili ayrıntılar yer alır.

Verilerin bir veritabanı içinde fiziksel olarak nasıl saklanacağına ilişkin bir şema sağlarlar. Bu nedenle, veri modelleri içerisinde en az soyut olanıdır. Varlıklar arasındaki ilişkileri gösteren ilişkisel tabloların yanı sıra bu ilişkileri sürdürmek için kullanılacak birincil anahtarlar ve yabancı anahtarlar da dahil olmak üzere ilişkisel bir veritabanı olarak uygulanabilecek nihai bir tasarım sunarlar. Aşağıda fiziksel veri modeline ait bir görsel yer almaktadır.



Şekil 6. Fiziksel Veri Modeli Örneği

3.1. Veri Modelleme Türleri

Veri modelleme, veritabanı yönetim sistemleri ile birlikte gelişmiştir ve işletmelerin veri depolama ihtiyaçları arttıkça model türleri de karmaşıklaşmıştır. Veri modelleme, bir veritabanı içindeki verilerin mantıksal yapısının bir diyagramıdır. Veri modelleme, insanların verileri daha iyi anlamalarına ve gelecekteki sonuçları tahmin etmek için verileri kullanmalarına yardımcı olabilir. En sık kullanılan veri modelleme türleri aşağıda özetlenmiştir.

1. Hiyerarşik Veri Modeli

Hiyerarşik veri modeli, verileri ebeveyn-çocuk ilişkisi olarak da bilinen ağaç benzeri bir hiyerarşide düzenlemeye yönelik bir yapıdır. Hiyerarşik bir veri modelinde her kayıt, hiyerarşide aynı seviyedeki her kayıt için aynı değer olan bir anahtarla benzersiz bir şekilde tanımlanır. Bu model, 1966 yılında tanıtılan ve özellikle bankacılıkta hızla yaygın kullanım alanı bulan IBM Bilgi Yönetim Sisteminde (IMS) uygulanmıştır. Bu yaklaşım daha yeni geliştirilen veritabanı modellerine göre daha az verimli olsa da, Genişletilebilir Biçimlendirme Dili (XML) sistemlerinde ve coğrafi bilgi sistemlerinde (GIS) hala kullanılmaktadır.

2. İlişkisel Veri Modeli

İlişkisel veri modelleri ilk olarak 1970 yılında IBM araştırmacısı E.F. Codd tarafından önerilmiştir. Bugün hala kurumsal bilgi işlemde yaygın olarak kullanılan birçok farklı ilişkisel veritabanında uygulanmaktadır. Bu modelde, veri segmentleri tablolar kullanılarak açıkça birleştirilir ve veritabanı karmaşıklığı azaltılır. İlişkisel bir model, ilişkisel veriler içeren bağlantılar aracılığıyla birbirleriyle ilişkili düğümler içerir. Bu modeller genellikle bilgileri hızlı ve kolay bir şekilde depolamak ve almak için veritabanları

oluşturmakta kullanılır. İlişkisel veri modeline bir örnek, insanlar hakkında bilgi depolamak için kullanılan bir tablo olabilir. Tabloda ad, soyad, sosyal güvenlik numarası, doğum tarihi vb. için sütunlar olacaktır.

3. Varlık- İlişki (ER) Veri Modeli

Varlık-ilişki (ER) veri modelleri, bir veritabanındaki varlıklar arasındaki ilişkileri temsil etmek için şekilsel diyagramlar kullanır. Veri mimarları tarafından veritabanı tasarım hedeflerini aktaran görsel haritalar oluşturmak için çeşitli ER modelleme araçları kullanılır.

ER modeli verileri aşağıdaki kategorilere ayırır:

Varlıklar: Birlikte çalıştığınız nesneler, eylemler veya kavramlar. Örneğin, müşteriler, ürünler ve satışlar birer varlıktır.

İlişkiler: Varlıklar arasındaki bağlantılar. Bunlar bire bir veya bire çok ilişkiler olabilir.

Öznitelikler: Bir varlığı veya ilişkiyi tanımlayan veriler. Örneğin, bir ürünün adı o ürünün bir özneliğidir.

4. Nesne Yönelimli Veri Modeli

Nesne yönelimli veri modeli, bilgileri tanımlamak ve açıklamak için nesneleri kullanan kavramsal bir veri modelidir. Bu, bilgiyi ilişkilerle birbirine bağlı varlıklar olarak tanımlayan varlık-ilişki modelinin tersidir.

Nesne yönelimli veri modelleri, nesne yönelimli programlama olarak ilgi görmüş ve 1990'ların ortalarında popüler hale gelmiştir. İlgili “nesneler” gerçek dünya varlıklarının soyutlamalarıdır. Nesneler sınıf hiyerarşileri içinde gruplandırılır ve ilişkili özelliklere sahiptir. Nesne yöne-

limli veritabanları tablolar içerebilir, ancak daha karmaşık veri ilişkilerini de destekleyebilir.

5. Boyutsal Veri Modeli

Boyutsal veri modelleri Ralph Kimball tarafından geliştirilmiştir ve bir veri ambarında analitik amaçlar için veri alma hızlarını optimize etmek üzere tasarlanmıştır. İlişkisel ve ER modelleri verimli depolamayı vurgularken, boyutsal modeller raporlama ve erişim için bilgi bulmayı kolaylaştırmak amacıyla veri fazlalığını artırır.

Boyutsal veri modelleri, iş zekası (BI) ve çevrimiçi analitik işleme (OLAP) sistemlerinin te-

melini oluşturur. Bu modeller genellikle geçmiş işlem verilerini içeren veri ambarları için uygulanır ancak daha küçük veri kümelerine de uygulanabilir.

Boyutsal bir modelin temel amacı, kullanıcıların iş tahminleri, tüketim eğilimleri ve diğer ilgili sorularına hızlı bir şekilde yanıt bulmalarına yardımcı olmaktır. Boyutsal modelleme, iş zekası raporlaması için organize bir yöntem sağlar. Kullanıcıların etkili işbirliği ve karar verme için bir kuruluş içindeki farklı bölümler arasında bilgi paylaşmasına olanak tanır.

3.2. Veri Modellemenin Faydaları

- Veri modelleme, geliştiricilerin, veri mimarlarının, iş analistlerinin ve diğer paydaşların bir veri tabanındaki veya veri ambarındaki veriler arasındaki ilişkileri görmesini ve anlamasını kolaylaştırır.
- Yazılım ve veritabanı geliştirmedeki hataları azaltır.
- Kurum genelinde dokümantasyon ve sistem tasarımında tutarlılığı artırır.
- Uygulama ve veritabanı performansını iyileştirmede önemli bir rol oynar.
- Sistem organizasyonunda veri haritalamayı kolaylaştırır.
- Geliştiriciler ve iş zekası ekipleri arasındaki iletişimi iyileştirir.
- Kavramsal, mantıksal ve fiziksel seviyelerde veritabanı tasarımı sürecini kolaylaştırır ve hızlandırır.

BÖLÜM ÖZETİ

Sistem analistleri, sistem geliştirmenin analiz aşamasında verilerin nasıl oluşturulacağı, nasıl depolanacağı, veriler üzerinde hangi işlemlerin gerçekleştirileceği gibi teknik ayrıntıya girmeden, sistemi genel hatlarıyla ifade edebilmek için model oluştururlar. Veri modelinde sistemin görsel bir temsili oluşturulur. Benzer biçimde, sistemin gereksinimlerini modelleyebilmek için kullanım durum kullanılmaktadır. Kullanım durumları, sistem mimarisini görselleştirmenin önemli bir unsuru olarak karşımıza çıkmaktadır. Veri modelleme ise veri noktaları ve yapıları arasındaki bağlantıları iletmek için sistem parçalarının görsel bir temsilini oluşturma sürecidir.

Sistem tasarımında bir müşterinin bir ürün, hizmet veya sistemle nasıl etkileşeceğini ifade etmek, gereksinimlerin toplanması ve üst düzey paydaş iletişimi için önemlidir. Kullanım durumları, bir sistemin gereksinimlerini yakalamak, modellemek ve belirtmek için kullanılan bir tekniktir. Kullanım Durumlarının Amaçları; Kullanım Durum Modeli, Kapsam yönetimi, Gereksinimlerin belirlenmesi, Kullanıcının sistemle etkileşime gireceği yolların ana hatlarıyla belirtilmesi, Sistem mimarisini görselleştirme, Teknik gereksinimleri iş paydaşlarına iletme ve risk yönetimidir.

Kullanım durum modeli, bir sistem ile bir aktör arasındaki etkileşimlerin görsel olarak sunumudur. Kullanım durum modelleri iş süreçlerini temsil eder. Bu da ön koşulların ve tetikleyicilerin daha fazla ifade edilmesine yardımcı olur. Bir kullanım durum modeli genellikle UML ile ifade edilir. Bu modelde dört ana bileşen bulunmaktadır: aktörler, sistem, kullanım durumları ve ilişkiler. Kullanım durum diyagramı ise, temel olarak bir tür kullanım durum modelidir. Kullanım durum diyagramları sistemin akışını ve davranışını görselleştirmek, sistemin işlevselliğini örneklemek, Temel sistem-kullanıcı etkileşimlerini temsil etme durumlarında kullanılır.

Kullanım durum diyagramı oluştururken dikkat edilmesi gereken unsurlardan birisi, kullanım durum isimlendirilmesinde tercih edilen kelimelerdir. Kelimeler sıradan değil, sistemin söz konusu davranışını ve işlevini yansıtacak şekilde seçilmelidir. Kullanım durumlar isimlendirilirken genellikle emir kipi kullanılır.

Veri modelleme, veri noktaları ve yapıları arasındaki bağlantıları iletmek için tüm bir bilgi sisteminin veya sistem parçalarının görsel bir temsilini oluşturma sürecidir. Amaç, sistem içinde kullanılan ve depolanan veri türlerini, bu veri türleri arasındaki ilişkileri, verilerin gruplandırılma ve organize edilme yollarını, biçimlerini ve niteliklerini göstermektir.

Kaynakça

- Alagöz, Emre (2014), Use Case Analizi #1. Elektronik Adres: <https://tr.linkedin.com/pulse/use-case-analizi-1-emre-alag%C3%B6z>
- Bittner, Kurt (2003). Use case modeling. Spence, Ian. Addison Wesley. ISBN 0-201-70913-9. OCLC 50041546
- Daly, Nicky (2022). What Is a Use Case?. Elektronik adres: <https://www.wrike.com/blog/what-is-a-use-case/>
- IBM Cloud Education (2020). Data Modeling. Elektronik Adres: <https://www.ibm.com/cloud/learn/data-modeling>
- Jackson, Michael (1995), Software Requirements & Specifications, Addison-Wesley. ISBN 0-201-87712-0.

