VERİTABANI SİSTEMLERİ

GİRİŞ

İşletmelerde kullanılan bilgi sistemleri stratejik, organizasyonel gereksinimlerin karşılanması için veritabanı geliştirilmesine ihtiyaç duyar. En yaygın kullanılan veritabanı tasarım modeli Varlık İlişki (ER-Entity Relationship model) modelidir. Günümüzde yaygın olarak kullanılmaya başlanılan diğer bir modelde tümleşik modelleme dili şemasıdır (UML-Unified Modelling Language).

VERİTABANI TASARIM AŞAMALARI

Veritabanı tasarım aşamaları üç aşamadan oluşmaktadır. Bunlar:

* *kavramsal veri modelleme (conceptual data modeling)*
* *mantıksal veri modelleme (logical data modeling)*
* *fiziksel veri modellemedir (physical data modeling*)

Kavramsal Veri Modeli

Kavramsal veri modeli (conceptual data model) veritabanında saklanacak verilerin, kısıtlarının ve ilişkilerinin bir gösterimidir. Bu model ile kullanıcıların gereksinimleri ve bu gereksinimleri karşılayacak tasarım tam olarak ifade edilmiş olur. Kavramsal veri modeli varlık ilişki diyagramının tasarlanması ya da tümleşik modelleme dili şeması ile oluşturulur.

**Kavramsal Veri Modelinin Özellikleri:**

*• Stratejik veri projelerinde en sık kullanılan veri modelidir*

*• Kullanıcı gereksinimlerini karşılamak için oluşturulur*

*• Varlıklar, ilişkiler ve bunların özelliklerini içerir*

Mantıksal Veri Modeli

Mantıksal veri modeli ilişkisel veritabanı teknolojilerinden bağımsız, veri depolama, veri kısıtları ve ilişkiler tarafından tamamıyla bağlanmış veri modelidir. Bir mantıksal veri modeli insanları, yerleri, nesneleri, kuralları ve ilişkiler ile onlar arasındaki olayları standartlaştırır.

**Mantıksal Veri Modelinin Özellikleri:**

*• Tüm varlıkları ve aralarındaki ilişkileri içermektedir.*

*• Her varlığın tüm öznitelikleri belirlenmiştir.*

*• Her varlık için benzersiz anahtar belirtilir.*

*• Farklı varlıklar arasındaki ilişkiyi tanımlayan yabancı anahtarlar belirtir.*

*• Bu aşamada Normalleştirme uygulanır.*

*• Veri modeli kapsamına bağlı olmakla birlikte tipik olarak 50-1000 arasında varlık içerir.*

*• Veritabanı yönetim sisteminden, veri depolama konumlarından ya da teknolojilerinden bağımsız olarak oluşturulur.*

*• Varlıklar ve öznitelikler tanımlara sahip olur.*

*• Veri özniteliklerinin kısıtları ve veri türleri genellikle belirlenmiş olur.*

Fiziksel Veri Modeli

Fiziksel veri modeli (physical data model) veri süreklilik teknolojisinin belirli bir sürümü ile bağımlı, tamamıyla bağlanmış veri modelidir. Burada hedeflenen teknoloji ilişkisel bir veritabanı yönetim sistemi (ACCESS, MS SQL, ORACLE vb.) , XML dokümanı, elektronik tablo veya başka bir veri uygulama seçeneği olabilir. Fiziksel veri modeli veritabanının nasıl inşa edileceğini belirtir.

**Fiziksel Veri Modelinin Özellikleri:**

*• Tüm tablo ve sütunlar belirtilir*

*• Tablolar arasındaki ilişki yabancı anahtar ile sağlanır*

*• Denormalization kullanıcı gereksinimlerine ve performans ölçütlerine dayalı olarak oluşabilir*

*• Fiziksel veri modeli farklı ilişkisel veritabanı yönetim sistemleri için farklılık gösterebilir*

*• Veritabanı yönetim sisteminin belirli bir sürümü, veri depolama konumları ya da teknolojileri için bağımlı olarak tasarlanmış ve geliştirilmiştir*

*• Öznitelikler genellikle duyarlı ve uzunlukları atanmış veri türlerine sahip olacaktır*

VARLIK İLİŞKİ MODELLEME

Varlık ilişki modelleme bir işletme veya iş alanında kullanılan verilerin detaylı ve mantıksal gösterimidir. Varlık ilişki modeli gerçek hayattaki varlıklar, aralarındaki ilişkiler ve varlıklar ile ilişkilerin özelliklerini içerir. Varlık İlişki modeli grafiksel olarak varlık ilişki diyagramı ile ifade edilir.

**Varlık ilişki modelinde veri yapısı grafiksel olarak gösterilirken üç temel öge kullanılır. Bunlar;**

*• Varlıklar*

*• Öznitelikler*

*• İlişkilerdir*

Varlıklar (Entity)

Varlık işletmenin verisini tutmak istediği kullanıcı ortamındaki kişi, yer, nesne, olay veya kavramı temsil eder. Bu varlık çeşitleri için bazı örnekler şunlardır:

Kişi: Çalışan, öğrenci, hasta

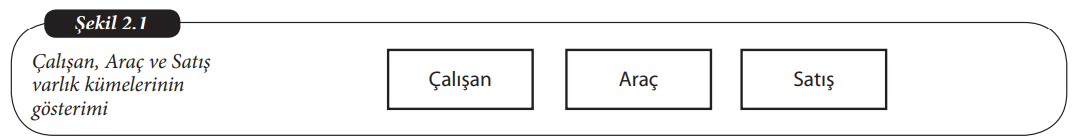
Yer: Depo, ambar, şehir

Nesne: Makine, inşaat, araç

Olay: Satış, kayıt, sözleşme yenileme

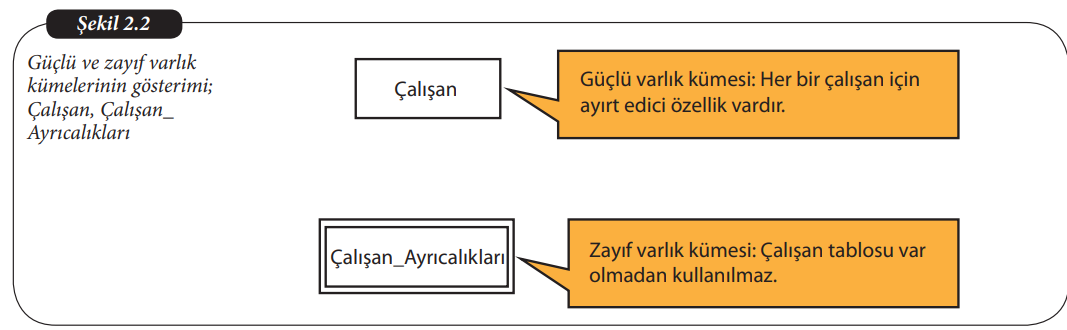
Kavram: Muhasebe hesabı, okulda verilen ders

Model içerisinde varlık kümesi dikdörtgen ile gösterilir ve içine varlığın adı yazılır. Mantıksal veritabanı tasarımında varlık kümesi tablo ile gösterilir. Şekil 2.1’de Çalışan, Araç ve Satış varlık kümelerinin grafiksel gösterimi verilmiştir.



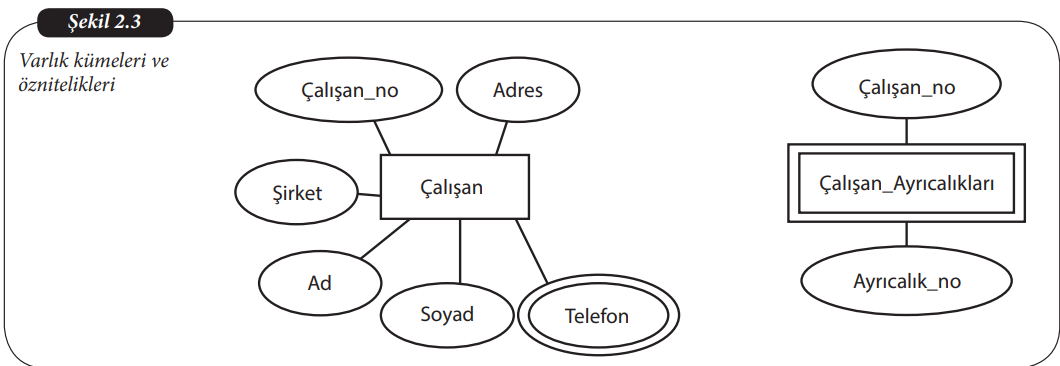
Güçlü ve Zayıf Varlık Kümeleri

Güçlü varlık kümesi diğer varlık kümelerinden bağımsız tanımlanabilen kümelerdir. Örneğin; **çalışan, öğrenci, hasta** güçlü varlık kümeleridir. Bu kümelerde her zaman varlıkları birbirinden benzersiz olarak ayıran bir öznitelik veya öznitelik birleşimi bulunur. Bunun tersine, bir varlık kümesi güçlü varlık kümesine bağlı olarak var olabiliyorsa zayıf varlık kümesi olarak adlandırılır. Şekil 2.2’deki Çalışan\_Ayrıcalıkları zayıf varlık kümesine örnek verilebilir. Çalışan varlık kümesi olmadan Çalışan Ayrıcalıkları kümesi var olamayacaktır. Zayıf varlık kümesi çift çizgili dikdörtgen ile gösterilir.



Öznitelikler (Attributes)

Her bir varlık kümesinin özellikleri olur. Öznitelik, işletmede kullanılan varlık kümesinin özellikleri veya karakteristikleridir. Modelde öznitelikler oval ile gösterilir ve içine adı yazılır. Öznitelik bağlı olduğu varlık kümesine düz çizgi ile bağlanır. Öznitelikleri adlandırırken ilk harfi büyük karakter ile başlayıp ikinci kelime var ise alt çizgi ile birleştirerek yazılır.

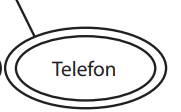


Türetilen Öznitelikler

Bazı öznitelik değerleri veritabanında depolanan diğer ilişkili özniteliklerden hesaplanabilir veya türetilebilir. Bu oluşturulan yeni özniteliğe türetilen öznitelik (derived attribute) denir. Örneğin, bir işletmede çalışanın işe başlama tarihini gösteren bir öznitelik (İş\_başlama\_tarihi) olsun. Bir çalışanın kaç yıl çalıştığı belirlenmek istenirse Çalışan\_yılı özniteliği İş\_başlama\_tarihi özniteliği ve sorgu tarihi kullanılarak hesaplanabilir. Öğrenci varlık kümesinde bulunanların Doğum\_tarihi özniteliği kullanılarak yaş özniteliği elde edilebilir. Türetilen öznitelikler varlık ilişki diyagramında kesikli çizgili oval şekillerle gösterilir.

Tek ve Çok Değerli Öznitelikler

Bir varlığın özniteliği sadece tek değer alıyorsa buna tek değerli öznitelik (single valued attribute) denir. Örneğin Çalışan varlık kümesi için Çalışan\_no özniteliği tek değerli özniteliktir. Bir varlığın özniteliği birden çok değer alıyorsa buna çok değerli öznitelik (multivalued attribute) denir. Örneğin Çalışan tablosunda çalışanların telefon bilgilerini saklamak için kullanılan Telefon özniteliğinde çalışana ait birden fazla telefon bilgisi tutulursa çok değerli öznitelik olur. Varlık ilişki diyagramında çok değerli öznitelik çift oval şekil ile ifade edilir.

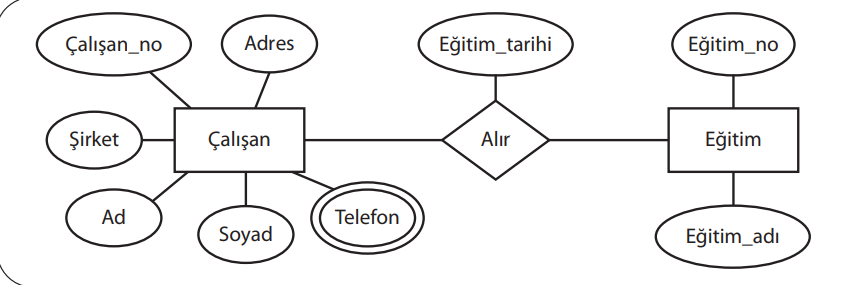


Anahtar Öznitelik

Varlık kümesindeki her bir örnek için farklı değer alan özniteliğe anahtar öznitelik denir. Yani anahtar öznitelikte iki örneğin değeri aynı olamaz. Anahtar öznitelik tanımlandığı varlık kümesinde bulunan tüm örnekler için benzersiz değerler alır. Örneğin Çalışan\_no

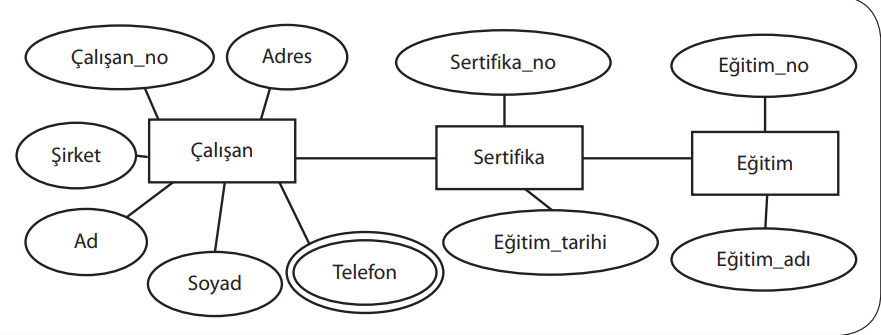
İlişki

İlişkiler veri modelinde farklı ögeleri bir arada tutar. lişkiler varlık ilişki diyagramında eşkenar dörtgen yani baklava dilimi şekliyle gösterilir ve içerisine ilişkinin adı yazılır. Verilen çoka çok ilişkide, bir çalışan birden fazla eğitim alabilir ve aynı şekilde bir eğitim birden fazla çalışan tarafından alınabilir. Çalışanların aldıkları eğitimlerin tarihleri de saklanmak istenirse Alır ilişkisine Eğitim\_tarihi özniteliği eklenir. İlişkilere eklenen öznitelikler de modelde oval içerisine özniteliğin adı yazılarak gösterilir.



İlişkisel Varlıklar

Bir ilişkinin özniteliği birden fazla ise bu ilişki bir varlık olarak gösterilebilir. Bu tip varlıklar ilişkisel varlık olarak adlandırılır. Şekil 2.5’te Çalışan ve Eğitim varlık kümeleri arasındaki bağlantı ilişki ve üzerinde özniteliği ile gösterilirken Şekil 2.6’da ilişkiyi tanımlayan öznitelik sayısı arttığından ilişkisel varlık gösterimi kullanılmıştır.



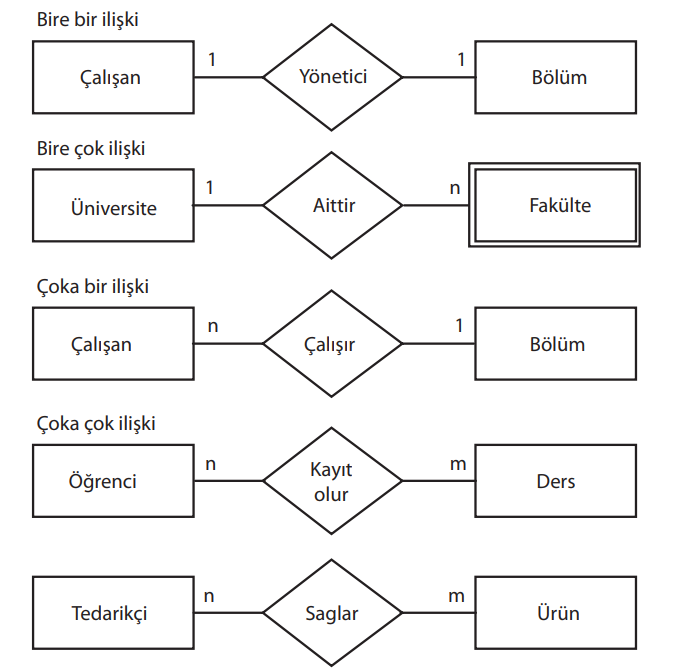
İlişki Türleri

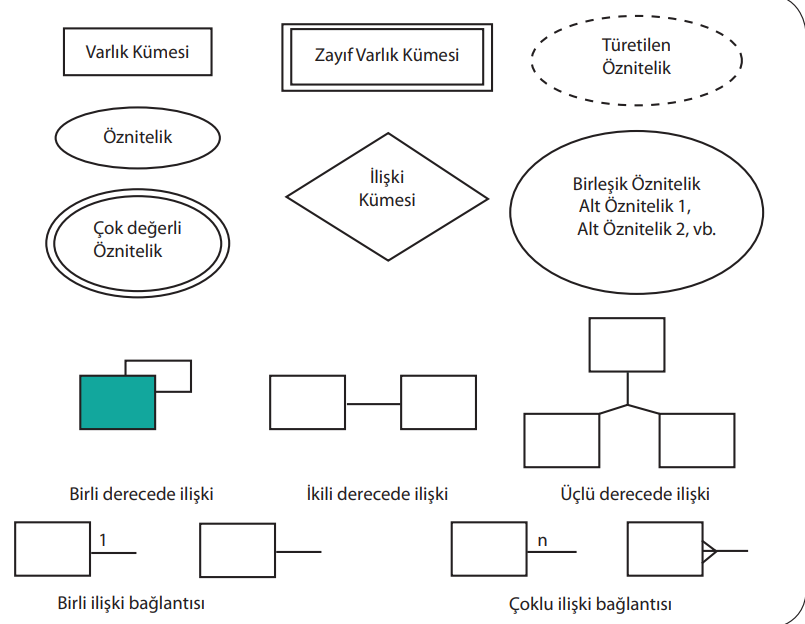
Varlık kümeleri arasında üç tür ilişkiden söz edilebilir. Bunlar,

*• Bire Bir İlişki (One to One 1:1)*

*• Bire Çok İlişki (One to Many 1:N) veya Çoka Bir İlişki (Many to One N:1)*

*•Çoka Çok İlişki (Many to Many N:N)’dir*

**

**