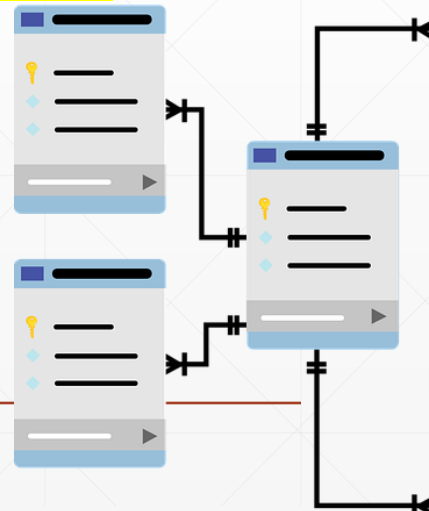


VERİ MODELİ



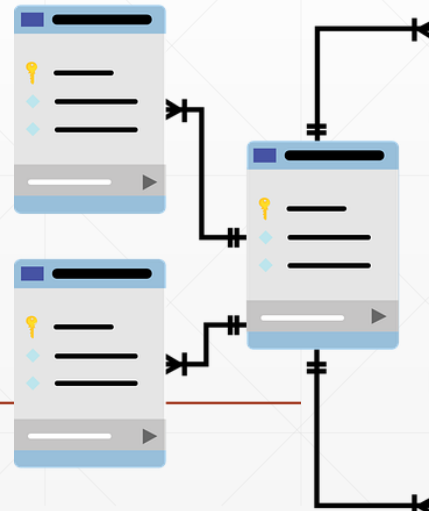
Veri Modeli

Bir veri tabanı yapısının temelini veri modeli kavramı oluşturmaktadır. Veriyi mantıksal düzeyde düzenlemek için kullanılan kavramlar, yapılar ve işlemler topluluğuna **veri modeli** adını veriyoruz. Veri modeli, bir veritabanının yapısını açıklayan kavramlar bütünüdür. Veritabanı yapısı ise; veri türleri, bağıntılar ve veri kısıtlamalarından oluşur. **Verileri mantıksal düzeyde düzenlemek için kullanılan yapılar, kavramlar ve işlemler topluluğuna veri modeli (data model) denir.**



Veri Modeli

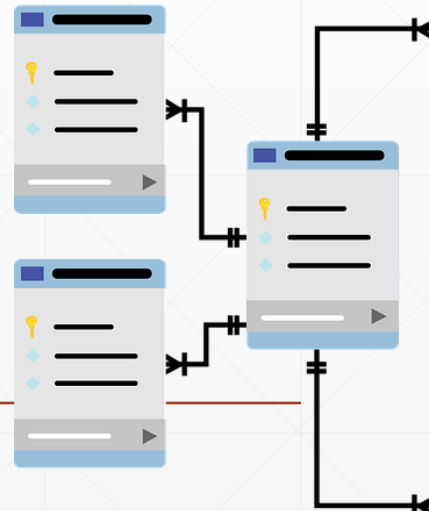
Bir veri tabanı yönetim sistemi, bir veri modeli üzerine inşa edilir. Veri modelleri, veri tabanındaki ilişkileri soyut bir şekilde canlandırdığından kavramsal düzeyde oluşturulur. Veri tabanının oluşturulması amacıyla geçmişten günümüze kadar bir takım farklılıklarla çeşitli veri modelleri geliştirilmiştir. Her biri, bir öncekinin eksikliğini tamamlamak amacına yöneliktir.



Veri Modeli

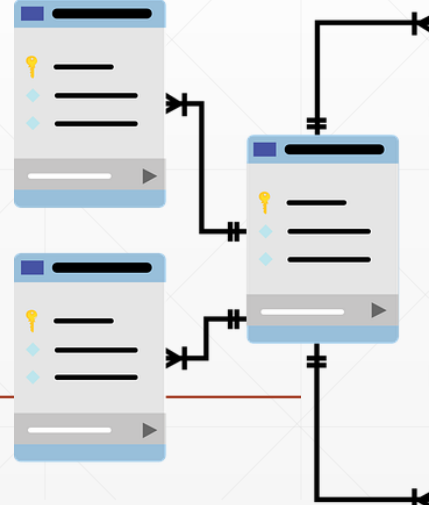
Veri modeli; veri tabanının yapısını tanımlayan kavramların bir kümesidir.

Veri tabanının modeli; veri tipleri, işlemler ve kısıtlamalardan oluşur. Veri modeli veri tabanının nasıl oluşturulacağı, nasıl kullanılacağı ve hangi sınırlar içerisinde tanımlı olduğu ile ilgilidir. Veri modeli içerisinde yer alan işlemler kümesi veri tabanından seçim yapma ve veri tabanını güncelleme ile ilgilidir.

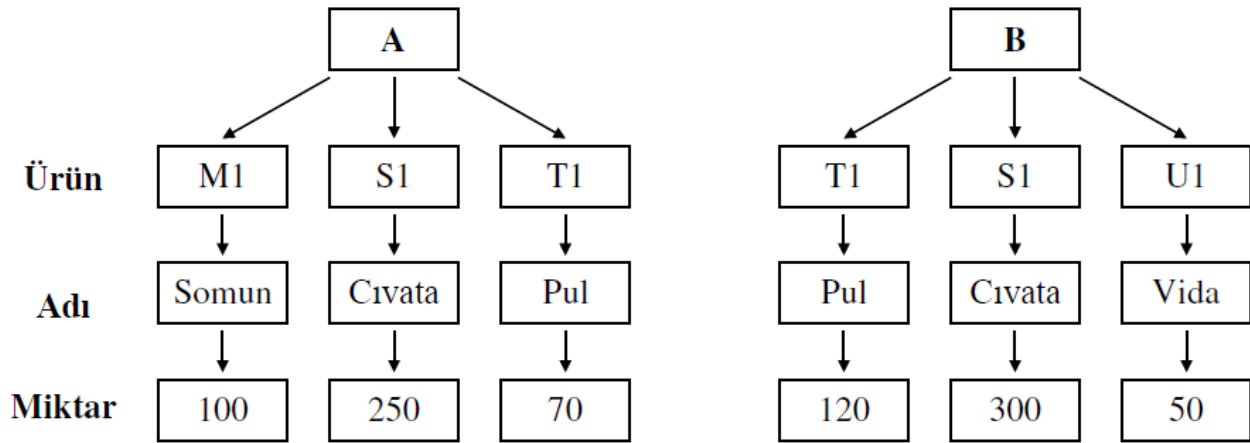


Veri Modeline Göre Veri Tabanı Yapıları

Kullanıcıların, kullanıcı ihtiyaçlarına en iyi hizmet verebilecek uygulama programlarını geliştirebilmeleri için veri tabanındaki veri elemanları arasındaki ilişkinin mantıksal olarak yapılanması gereklidir. **Amaç, verilerin nasıl ilişkilendirileceğinin gösterilmesidir.** Veri tabanlarında depolanan kayıtlar ve onların birbirleriyle ilişkileri değişik mantıksal yapılarla gösterilebilir. Veri Tabanı Yönetim Sistemleri bu yapıların oluşumları ve onların fonksiyonları kullanılarak tasarlanır. Veri tabanları **4 yapı**dan herhangi birisiyle yapılandırılabilir. Burada yapılan sınıflandırma veri modeline göre yapılmaktadır.



Bu yapı, bilinen en eski veri modelleme sistemidir. Kullandığı veri modeli ağaç yapısıdır. Ebeveyn ve çocuk ilişkisine dayalı bir sistemdir. Hiyerarşim modelde veri türleri üstten aşağıya doğru bir usulde düzenlenir. Oluşturulan mantıksal bağlantılarla veri türleri ilişkilendirilir. Buda bir ağaç gibi veya organizasyon haritası şeklinde görülür.



1

Hiyerarşik Veri Tabanları

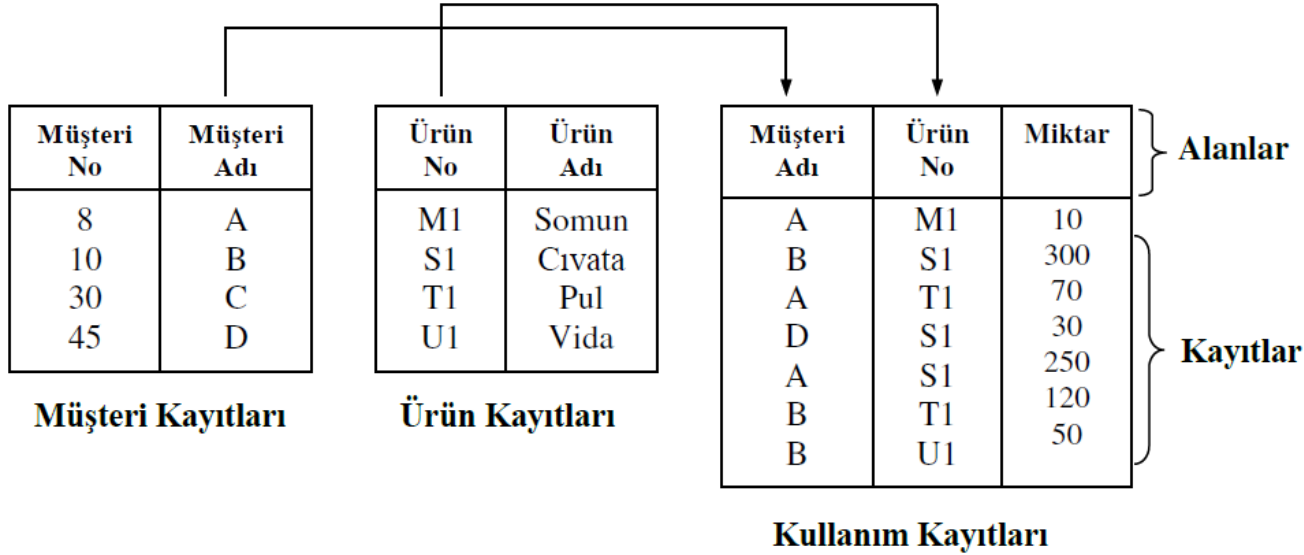
Bu modelde temel alınan şeyler verilerin birbiriyle ilişkileridir. 1970'li yılların başında geliştirilmiştir. Günümüzde en çok kullanılan modeldir. PC'lerden sunuculara kadar yayılmış bir yelpazeye sahiptir ve bu mimari kullanılarak birçok yazılım üretilmiştir.

Bu formda veri tabanı organizasyonu iki boyutlu tablolar şeklinde düşünülerek kullanıcıya sunulur. Veri tabanı kullanıcıları veri raporlarını bu yolla görür. İlişkisel VTYS birçok giriş sorgusu sunar. Böylece bir veri dosyası bir sayfada aşağıya doğru ilerleyen birkaç sütun içerir. Bu sütunlar kişisel alanları teşkil eder. Bir sayfadaki satırlar çeşitli alanların oluşturduğu kişisel kayıtları gösterir.

3

İlişkisel Veri Tabanı Sistemi

İki yada daha fazla veri dosyasında bulunan ortak veri alanları vasıtasıyla bu tür veri dosyalarında ilişki kurulabilir. Bu ortak alanlar tümüyle benzer olmalı ve aynı boyutta ve tipte olmalıdır.



3

İlişkisel Veri Tabanı Sistemi

İlk geliştirilen VTYS'ler kolayca önceden tanımlanmış veri alanlarına ve kayıtlara göre yapılandırılabilen homojen veriler için tasarlanmıştı. Fakat bugünkü ve gelecekteki uygulamalar, sadece önceden tanımlanmış sayıları ve karakterleri değil, çizimleri, resimleri fotoğrafları, sesleri ve videoları da saklayabilen veri tabanlarını gerektirecek.

İlk VTYS'ler bu tür grafik ve multimedya uygulamaları için uygun değildir. Bu tür verileri ilişkisel sistemlerde yönetmek için büyük çaplı programlama gerekir çünkü bu karmaşık veriler tablolara ve satırlara dönüştürülmelidir.

4

Nesneye Yönelik Veri Tabanları

Fakat nesneye yönelik (object-oriented) VTYS'ler verileri ve prosedürleri nesnelere halinde saklar ve bu nesnelere istediğinde otomatik olarak alınabilir ve paylaşılabilir. Nesneye yönelik veritabanı yönetim sistemleri (Object-oriented VTYS, OOVTYS) popüler olmaya başladı çünkü bu sistemler web uygulamalarında kullanılan çeşitli multimedya bileşenlerini ya da Java "apletlerini" yönetmek için kullanılabilir.

4

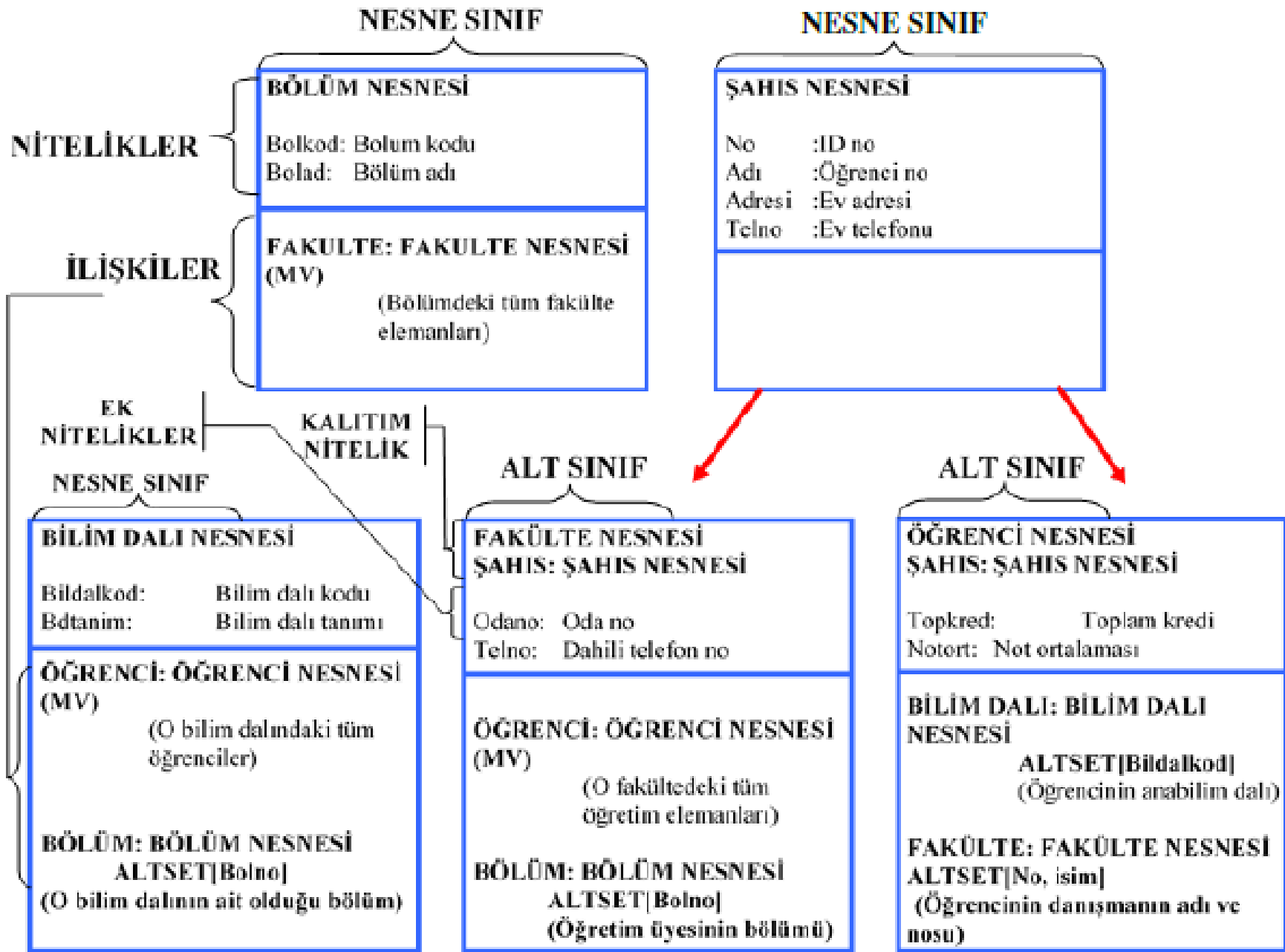
Nesneye Yönelik Veri Tabanları

OOVTYS, tekrarlı veriler gibi veri türlerini saklamak için de oldukça kullanışlıdır.

Örneğin üretim uygulamalarında parçalar içinde parçaların bulunduğu (malzeme listesi) tekrarlı veriler bu sistemlerde kolayca baş edilebilir. Finans ve ticaret uygulamaları da genellikle OOVTYS kullanır çünkü bu uygulamalar yeni ekonomik şartlar altında kolayca değiştirilebilen veri modelleri ister.

4

Nesneye Yönelik Veri Tabanları



Yandaki şekilde Nesne yönelimli veri tabanı yapısı görülmektedir.

Varlık İlişki Diyagramları

Varlık İlişki Diyagramları

İlişkisel veri tabanı tasarımında en fazla kullanılan yapısal tasarım yada modelleme aracı Varlık İlişki Diyagramlarıdır (Entity-Relationship E-R).

Varlık İlişki Diyagramları, veri tabanı tasarımında kullanılan ve varlıklar (entity) arası ilişkileri gösteren diyagramlardır. Bu diyagramlar, bir veri tabanı için referans dokümanlar olarak da kullanılabilirler. İlişkisel veri tabanı tasarımında amaç, veri tekrarını azaltan ve veri tutarlılığını yükselten bir yapının oluşturulmasıdır.

Varlık İlişki Diyagramları

Veri tabanı tasarım ilk olarak, veri tabanı yapılacak kurum ya da kuruluşun ihtiyacı olan bilgilerin analizi yapılır. Bu süreçte, veri tabanı tasarımcısı, veri tabanını kullanıcıları ile görüşerek kullanıcı ihtiyaçlarını tespit eder. Veriler toplanıp sistem analizi yapıldıktan sonra, ilişkisel veri modeline göre veri tabanı şeması oluşturulur. Bu şema, **veri tabanı kullanıcılarının ihtiyaç duydukları verilerin kısa açıklamasıdır.**

Bu şema içerisinde varlık tipleri ve ilişkilerin açıklaması bulunur. Daha sonra bu şemaya göre veri tabanını oluşturacak tablolar çıkarılır.

Varlık İlişki Diyagramları

Varlık-ilişki modeli, VTYS'den bağımsız veri çözümlemede en çok kullanılan modeldir. Bu model kullanılarak önce; VTYS'den bağımsız olarak veriler çözümlenir, veri modellenmesi yapılır, veriler ve veriler arası ilişkilerin anlamları ve özellikleri incelenerek Varlık-İlişki Diyagramları oluşturulur. Ardından kullanılacak VTYS belirlenir. VTYS'de de fiziksel olarak veri tabanı şemaları oluşturulur.

Varlık-İlişki Diyagramlarının Bileşenleri

Varlık-İlişki diyagramları oluşturulmada varlıkların ve onlarla ilgili özelliklerin belirlenmesi gereklidir. Varlıklar ve özellikler belirlendikten sonra yapılacak olan, varlıklar arasındaki ilişkilerin oluşturularak Varlık-İlişki Diyagramının yapılandırılmasıdır. Varlık - İlişki Diyagramlarının aşağıdaki temel semboller kullanılır.



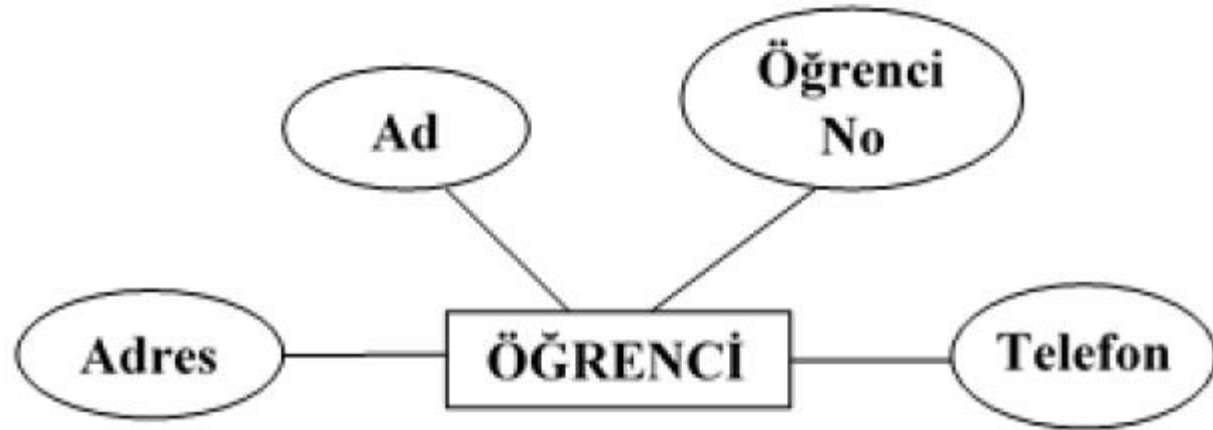
Gerçek hayatta diğerlerinden ayırt edilebilen nesnelere varlık denir. Bir varlık, bir çevresel veri ya da bir kaynak olabilir. Bir varlık, kişi, araba, ev veya çalışan gibi fiziksel nesnelere olabileceği gibi, şirket, iş veya ders gibi fiziksel olmayan nesnelere de olabilir.

Örneğin, kişi, yer, müşteri, depo, parça, ürün, ekipman, sipariş vb. hepsi birer varlıktır. Her varlık kendisini tanımlayan kendisine has özelliklere sahiptir.

1

Varlık ve Özellikleri

Varlık kümesi, Veri tabanında benzer varlıklar ve özellik değerlerinden oluşan kümedir. Varlık-İlişki şemasında varlık kümeleri dikdörtgen içinde belirtilir. Özellikler ise oval bir daire içinde belirtilerek ilgili varlık kümesine çizgi ile bağlanır. Şekilde öğrenci varlık kümesi ve ad, numara, adres ve telefon özelliklerinin şeması görülmektedir.



1

Varlık ve Özellikleri

Özellik, herhangi bir varlığa ait durumlardır.

Örneğin müşteri no, müşteri adı, müşteri adresi vb. durumlar, müşteri varlığının özellikleri olarak ifade edilebilirler. Yada bir çalışan varlığı, çalışan adı, yaşı, adresi, geliri ve görevi özellikleri ile tanımlanır.

Varlıklar Varlık-İlişki Diyagramı da dikdörtgen sembolüyle ifade edilirler. Özellikler o varlığı diğer varlıklardan ayırt etmeye yarar. Özellikler Varlık-İlişki Diyagramlarında elips şeklinde gösterilirler. Özellikler de kendi aralarında farklılık arz ederler.

2

Özellikler
(Attributes)

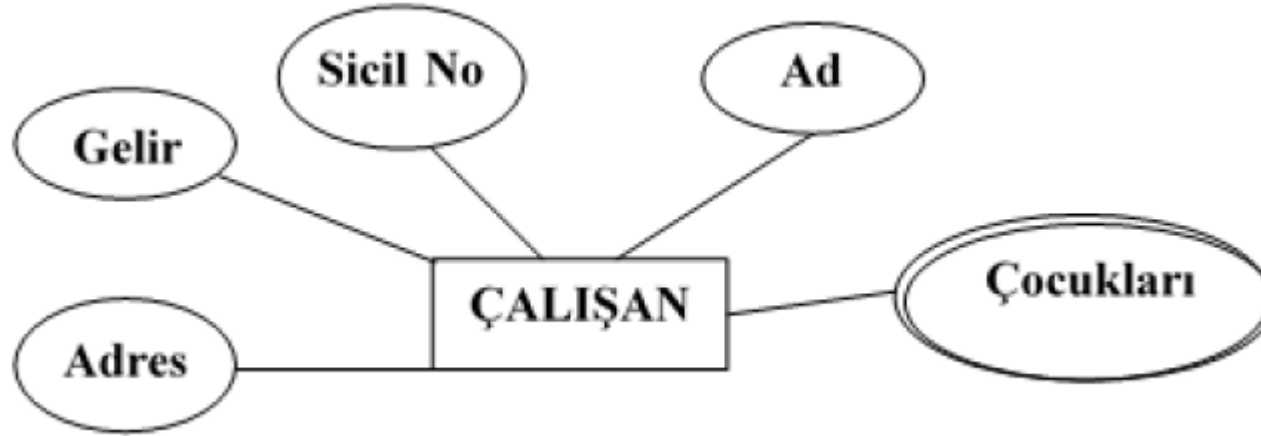
Tek Deęerli zellikler - ok Deęerli zellikler

Bir varlıęın sadece 1 deęer alabilen nitelięine denir. rneęin bir iř yerindeki alıřanların tek sosyal sigorta numaraları vardır. Genellikle bir varlıęın tek bir deęeri vardır. **Bir varlıęın bir zellięinin aldıęı deęer tek ise bu nitelięe tek deęerli zellik denir.** Ancak bazı zelliklerin birden ok deęeri olabilir.

rneęin alıřan varlıęının yabancı dil zellięi, birden ok deęer alabilir. Bu durumda alıřan varlıęının yabancı dil zellięi ok deęerli zellik denir. **Dięer ifadeyle ok deęerli zellik, Bir varlıęın birden fazla deęer alabilen zellikleridir.**

Tek Deęerli zellikler - ok Deęerli zellikler

Örneęin bir alıřanın birden fazla telefon numarası olabilir. Bařka bir örnek olarak, bir alıřanın ocukları özellięi bir kiři de olabileceęi gibi birden ok kiři de olabilir. Bu durumda alıřan varlıęının ocukları özellięi ok deęerli özelliktir. Varlık-İliři řemasında ok deęerli özellikler **ift izgili oval** olarak gösterilir.



Veri Kümesi

Veri Kümesi (Domain): Bir özelliğin alabileceği değerler. Varlık kümesinde özelliklerin alabileceği değerler aralığını belirler. Örneğin öğrenci varlık kümesinde, doğumTarih özelliği 01.01.1980 ile 01.01.1990 arasında olabileceğini belirlemek için domain kullanılır.

Örnekleri şu şekilde çoğaltabiliriz Cinsiyet → Erkek, Kız. Göz Rengi → Mavi, Ela, Kara, vs. Vize → [0-100]. Domain aralığı Varlık-İlişki şemasında gösterilmez.

Veri Türü

Veri Türü: Bir özelliğin içerebileceği verinin hangi türden olacağını belirler.

Tamsayı - INTEGER,

Değişken sayıda karakter - VARCHAR,

Tarih – DateTime ve Para - Money gibi.

Türetilmiş Özellik

Türetilmiş Özellik: Bir varlığa ait özellikler kullanılarak yeni bir özellik türetilebilir.

Örnek; Doğum Tarihi özelliğinden türetilen Yaş özelliği.

Anahtar Özellik

Anahtarlar Özellik: Varlıkları ya da ilişkileri birbirlerinden ayırt etmek için kullanılan Özellik ya da özellik grubuna denir. Her varlığa ait bir özellik anahtar olmalıdır. Varlık kümesindeki bir veya daha fazla özelliğin değeri, her bir varlık için farklı ise bu özellik anahtar özelliktir.

Örneğin öğrenci varlık kümesinde **öğrencino** anahtar özelliktir. Çünkü bir üniversitede, öğrenci varlık kümesinde hiçbir varlığın öğrencino özelliği aynı olamaz. Benzer şekilde Çalışan varlık kümesinde, **sicilno** özelliği anahtar özelliktir. Hiçbir çalışanın sicilno'su aynı olamaz. Anahtar özellik, Varlık-İlişki şemasında özelliğin altı çizilerek gösterilir.

Aday Anahtar

Aday Anahtar (Candidate Key): Bir varlığı kesin olarak tanımlayan ve tek olan özelliklere aday anahtar denir.

Örneğin bir çalışanın **sigorta numarası** ve **TC kimlik numarası** aday anahtar olabilir fakat, çalışanın adı aday anahtar olamaz çünkü aynı ada sahip birden fazla çalışan olabilir.

Birincil Anahtar

Birincil Anahtar (Primary Key): Aday anahtarlar arasından seçilen anahtardır.

Varlık - İlişki Diyagramlarında bir niteliğin birincil anahtar olduğunu belirtmek için o özellik **elips içinde altı çizili** olarak yazılır.

Birleşik Anahtar

Birleşik Anahtar (Composite Key): Eğer bir aday anahtar iki ya da daha fazla özellikten oluşuyor ise bu anahtar birleşik anahtardır.

İlişki (Relationship): Varlıklar arasındaki bağlantılardır.

İki veya daha fazla varlık kümesi arasında kurulan anlamlı bağıntılara **ilişki** denir.

Aynı varlıklar arasında farklı türden ilişkiler olabilir veya bir ilişki bir ya da daha fazla varlık arasında olabilir. Bir ilişki, iki ya da daha fazla veri varlığı arasındaki mantıksal etkileşimdir. Varlık-ilişki, hem varlıkları hem de ilişkileri göstermektedir.

3

İlişkiler

İlişkiler, eşkenar dörtgen sembolüyle ifade edilirler.

Dörtgen içine ilişkinin adı yazılır. Müşteri varlık kümesi ile ürün varlık kümesi arasında satın alır ilişkisi örnekleri verilebilir.



3

İlişkiler

İlişki Derecesi: İlişkide bulunan varlık kümelerinin sayısı ilişkinin derecesini oluşturur. Genelde ilişkiler iki varlık kümesi arasında yapılır. Ancak bazı durumlarda, ilişkide ikiden fazla kümesi yer alabilir.

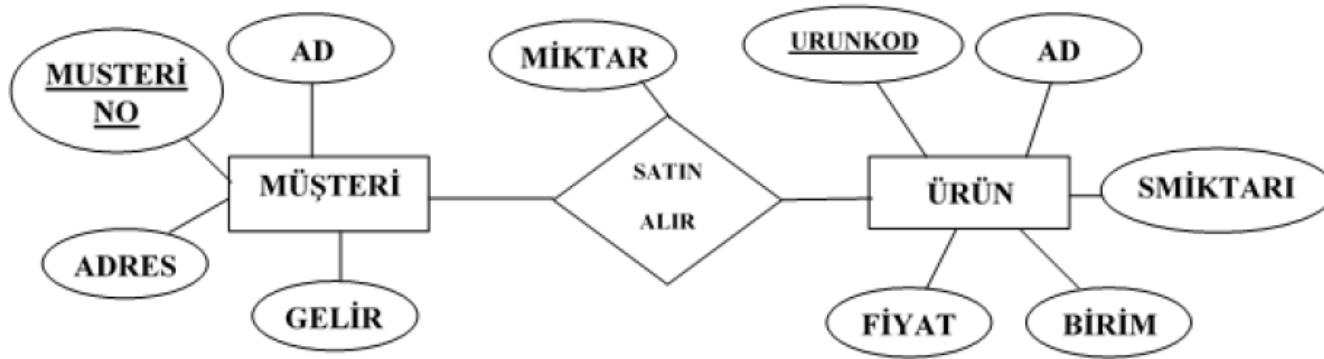
Örneğin bir çalışanın hangi projede hangi görevi yaptığı bilgisi gerekli ise; çalışan, proje ve görev varlık kümeleri arasında çalışır ilişkisi yapılır.

3

İlişkiler

Tanımlayıcı Özellik: Varlık kümeleri arasındaki ilişkiden oluşan özelliklere tanımlayıcı özellik denir.

Tanımlayıcı özellikler ilişkinin olması ile var olabilir. İlişki oluşmaz ise bu özellikler de var olamaz. Müşteri varlık kümesi ile ürün varlık kümesi arasında satın alır ilişkisinde, müşterinin satın aldığı ürünün miktarı özelliği tanımlayıcı özelliktir. Tanımlayıcı özellikler ilişki ile bağlanır.



3

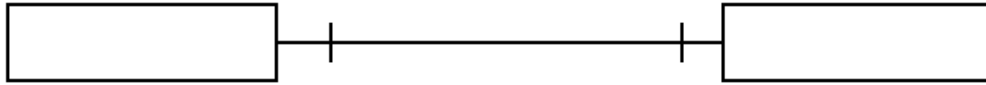
İlişkiler

İki varlık arasında genelde 3 ilişkiden bahsedilebilir.

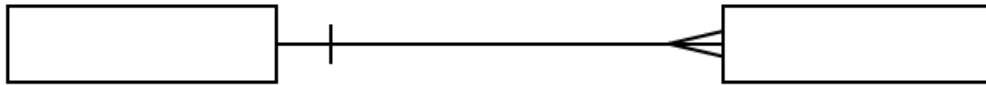
Bunlar;

1. Bire-Bir (one-to-one) ilişki (1:1)
2. Bire-Çok (one-to-many) ilişki (1:M)
3. Çoğa-çok (many-to-many) ilişki (M:M) dir.

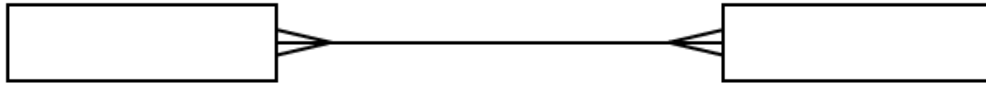
Bire Bir İlişki



Bire-Çok İlişki



Çok-Çok İlişki



3

İlişkiler

Bire-bir (1-1) iliřki

A varlık kümesinin bir elemanı, B varlık kümesinin bir elemanı ile iliřki kurabiliyorsa bu iliřki türüne bire-bir iliřki denir.

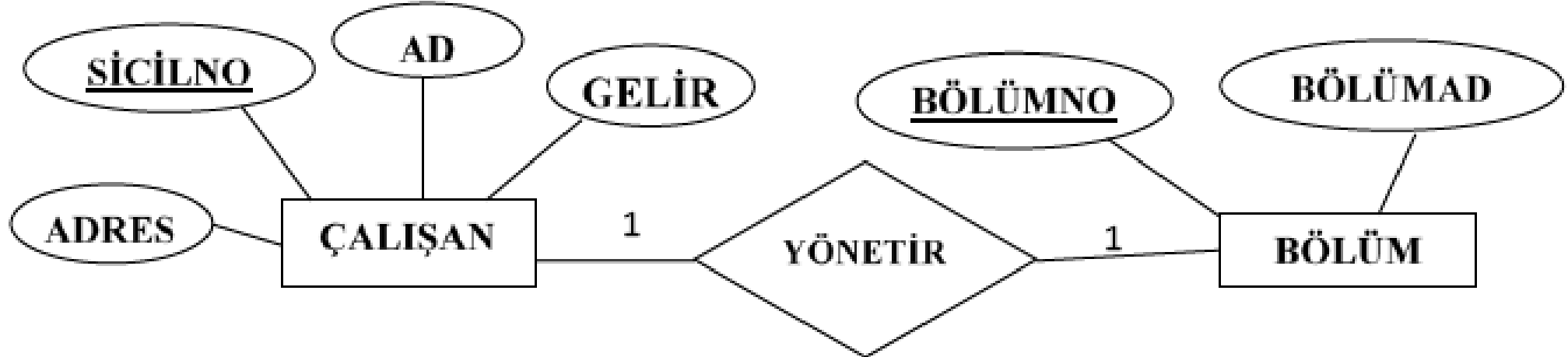
Örneđin Her müşteri'nin bir hesabı olabilir iliřkisi bire-bir iliřkidir(Şekil 8). Kiři ve kimlik, zaman kartı ve bordro fiři, araba ve plaka vb. arasındaki iliřkiler bire-bir iliřkiye örnek olarak verilebilir. Şekil 9. Çalışan - Bölüm yönetir iliřkisini gösterir.

Şekil

Müşteri No	Müşteri Adı	Hesaplar	Bakiye
101	Ali	34013	5.000
203	Ayşe	33790	45.000
405	Mehmet	33567	2.500,
507	Derya	33344	1.000

8: Her

Müşterinin Bir Hesabı Olabilir 1-1 İlişkisi

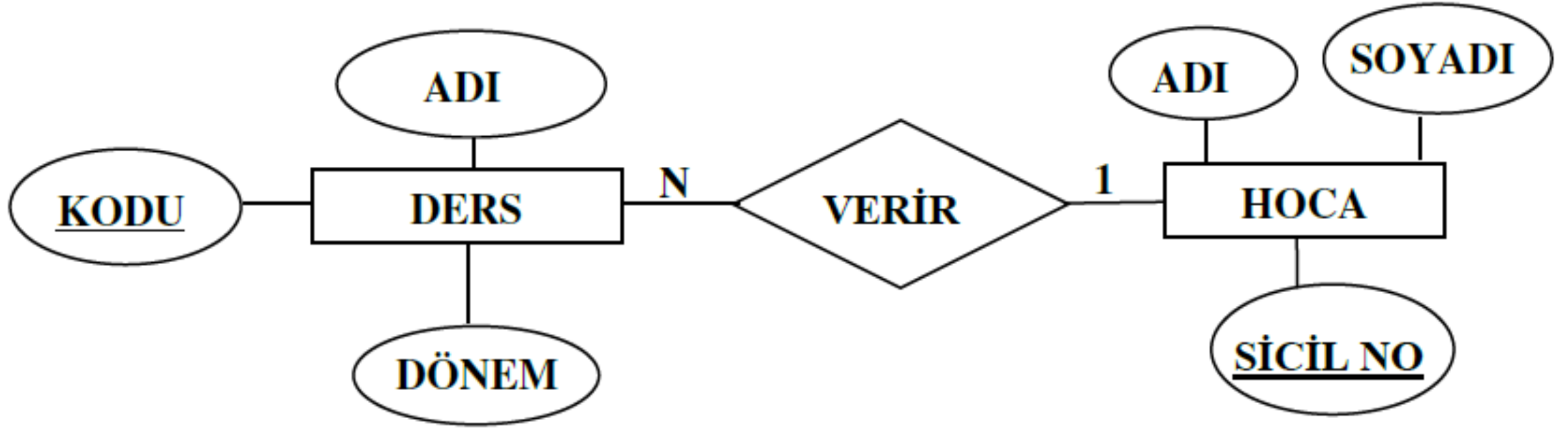


Şekil 9. Çalışan - Bölüm Yönetir 1-1 İlişkisi

Bire-Çok (1-N) İlişki

A varlık kümesinin bir elemanı B varlık kümesinin birden çok elemanı ile ilişki kurabiliyorsa bu ilişki bire-bir ilişkidir.

Hoca ile ders varlıkları arasında bire-çok bir ilişkinin olduğu görülmektedir. Yani bir hoca, birden fazla derse girebildiği, buna karşılık bir dersi sadece bir hoca tarafından verildiği ifade edilmektedir (Şekil 10).



Şekil 10. Hoca Ders Verir Bire-Çok İlişkisi

Çoğa-Çok (n-n ya da n-m) İlişki

A varlık kümesinin bir elemanı, B varlık kümesinin sıfır ya da birçok eleman ile ilişki kurabiliyor ve B varlık kümesinin bir elemanı, A varlık kümesinin sıfır ya da birçok elemanı ile ilişki kurabiliyorsa bu ilişki türü çoğa çok ilişki türüdür.

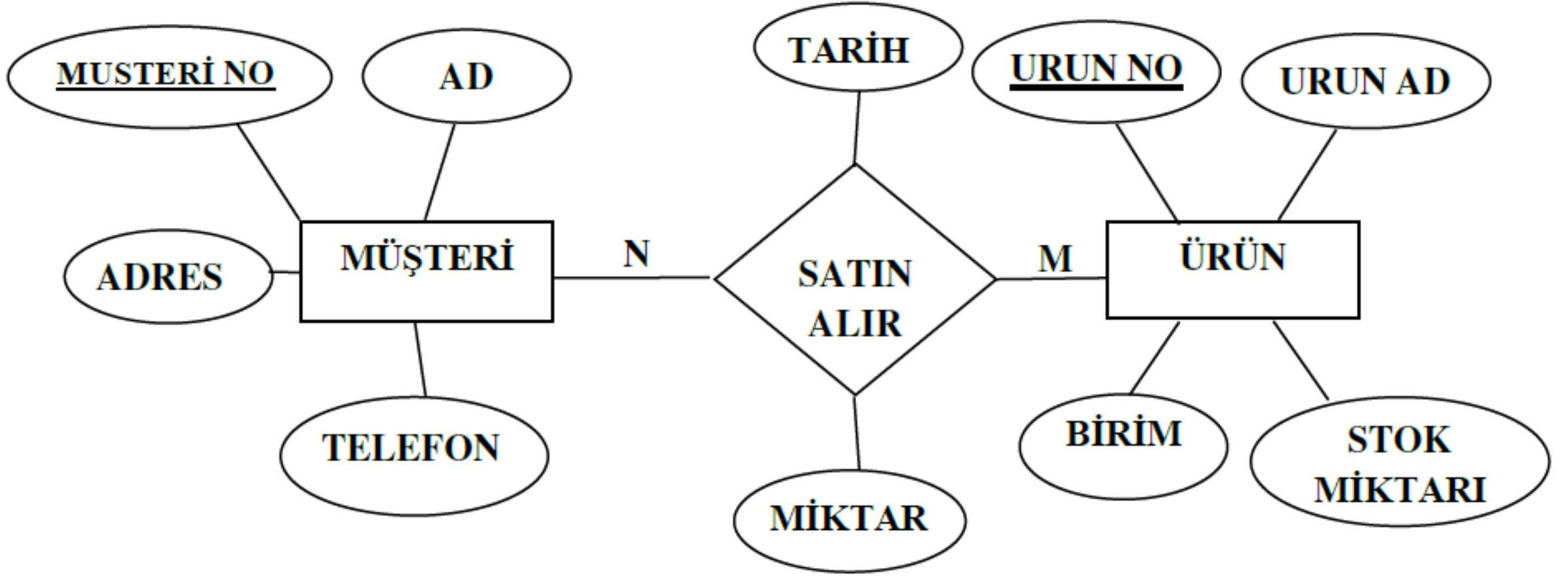
Çoğa çok ilişki kısaca n-m ilişki olarak da adlandırılabilir. Kitap ile Yazar varlıkları arasında çoğa-çok bir ilişkinin olduğu görülmektedir. Yani bir kitabın birden fazla yazar tarafından yazılabildiği, aynı zamanda bir yazarın da birden fazla kitap yazabilmesidir.

Çoğa-Çok (n-n ya da n-m) İlişki

Benzer şekilde, Proje ile Çalışan, Müşteri ile Ürün arasındaki ilişkiler de çoğa-çok ilişkiye örnek olarak verilebilirler.

Örneğin müşteri ile ürün varlık kümeleri arasında satın alır ilişkisi çoğa çok ilişkidir.

Çünkü bir müşteri birden çok ürün satın alabilirken, bir ürünü birden çok müşteri satın alabilir(Şekil 11).



Şekil 11: Müşteri-Ürün Satın Alır N-M İlişkisi

Varlık-İlişki Şemasının Tablolara Dönüştürülmesi

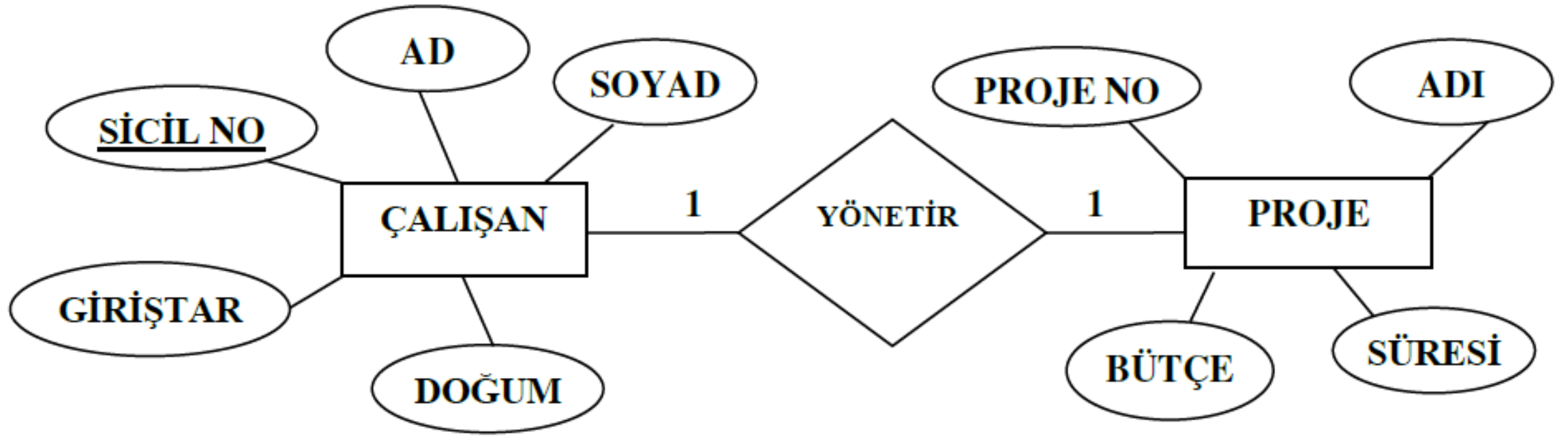
1. Bire-Bir İlişkilerin Dönüştürülmesi

Birebir ilişkiyi oluşturan varlık kümeleri tablolara dönüştürülür. Özellikleri tabloların alanlarına dönüşür. Uygun olan varlık kümesinin anahtar alanı diğer varlık kümesine yabancı anahtar olarak eklenir. Bire-bir ilişkide belirtilen tanımlayıcı özellikler, yabancı anahtar eklenen tabloya alan olarak eklenir.

Varlık-İlişki Şemasının Tablolara Dönüştürülmesi

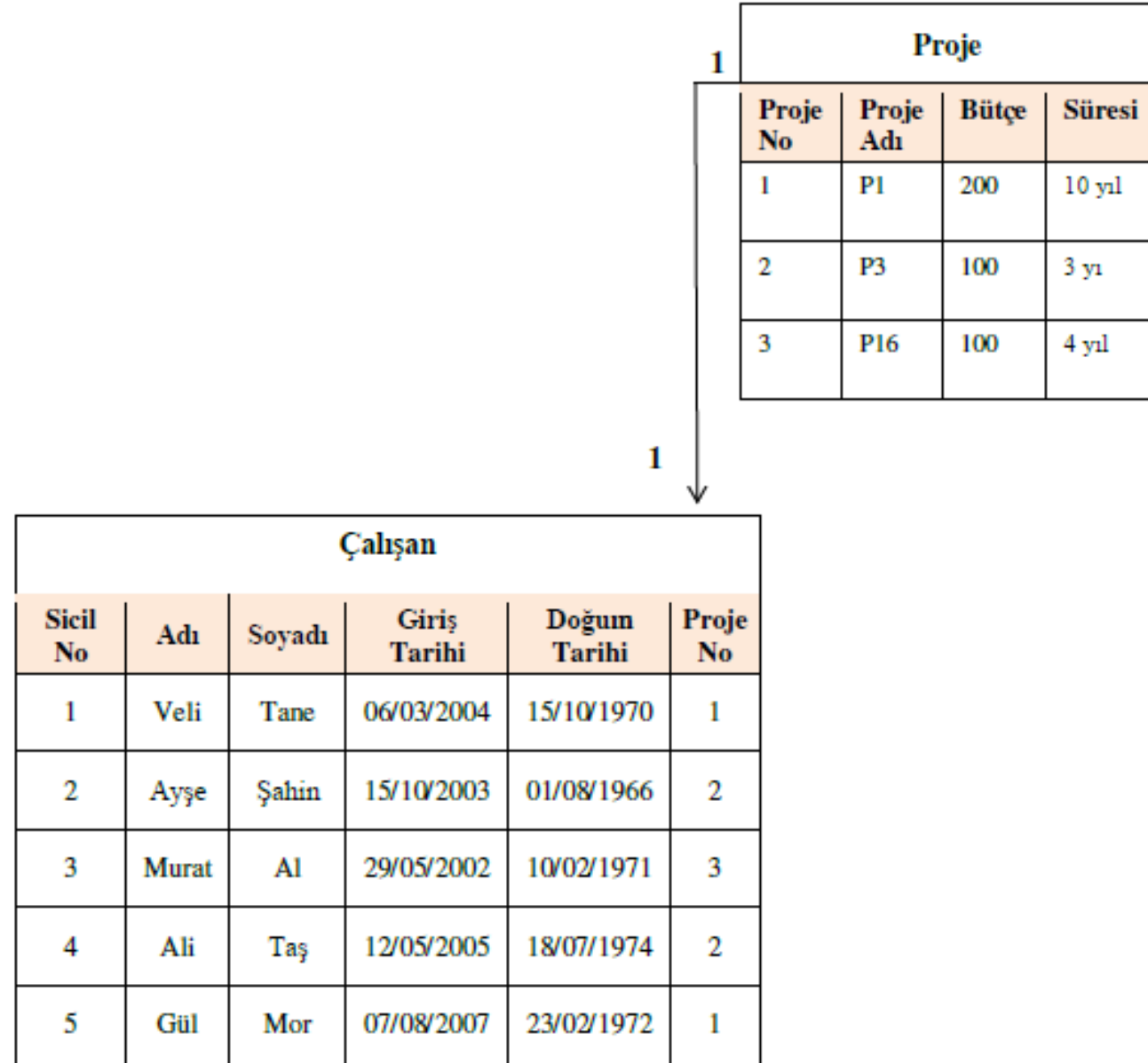
1. Bire-Bir İlişkilerin Dönüştürülmesi

Örneğin proje ve Çalışan varlık kümeleri arasındaki yönetir birebir ilişkisinde iki şekilde dönüştürme yapılır (Şekil 12). Birincisi, her iki varlık kümesi proje ve Çalışan tablolarına dönüştürülür. Proje tablosuna Çalışan tablosunun anahtar alanı olan **sicilno** alanı yabancı anahtar olarak eklenir. İkincisi, benzer şekilde her iki varlık kümesi proje ve öğrenci tablolarına dönüştürülür. Öğrenci tablosuna proje tablosunun anahtar alanı olan **projeno** alanı yabancı anahtar olarak eklenir.



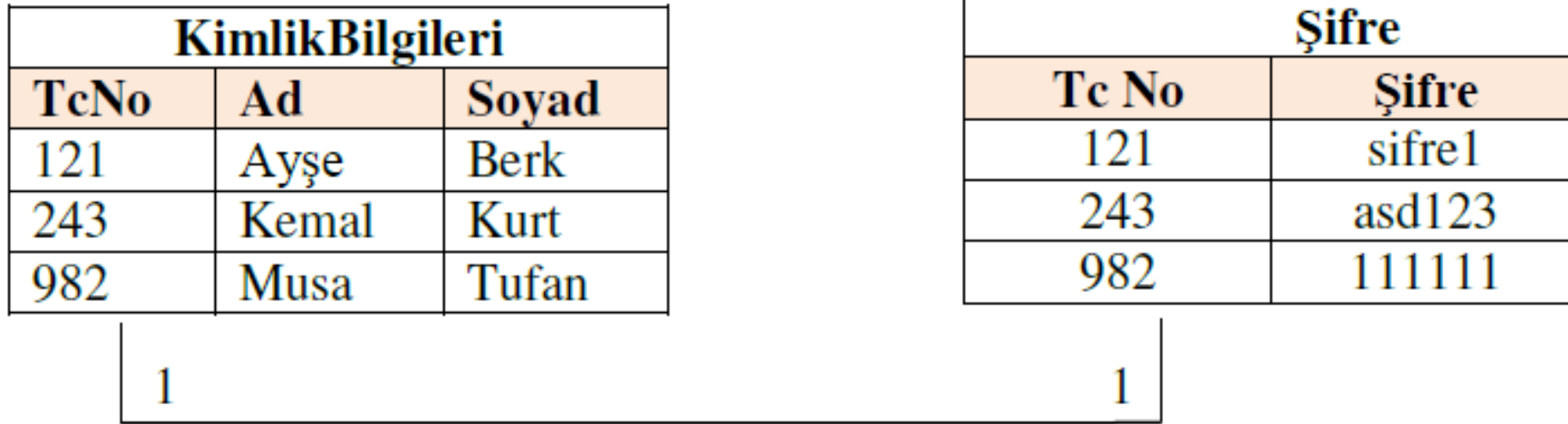
Şekil 12 Çalışan-Proje Yönetir Bire-Bir İlişkisi Varlık-İlişki Şeması

Varlık-İlişki şemasının tablolara dönüştürme ya birinci anahtar alanı diğer tablonun içerisine koyarak ya da ikinci tablonun anahtar alanını birinci tablonun içerisine koyarak yapılır.



Şekil 8.1. Çalışan-Proje Yönetir İlişkisi Tabloları

Bire-Bir (1-1) İlişki; Tablolar arası ilişki kurulan alanların her iki tabloda da tek olması anlamına gelir (Şekil 13).



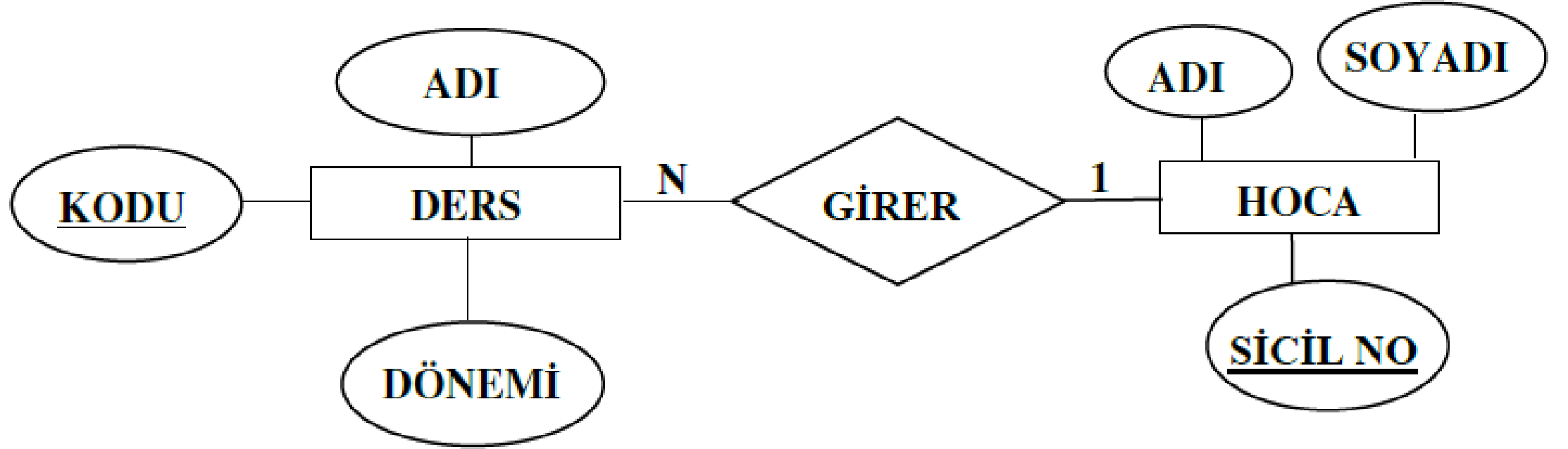
Şekil 13 Bire-Bir (1-1) İlişki: 1 Kişiye Ait Sadece 1 Şifre Olabilir, 1 Şifre 1 Kişiye Ait Olabilir.

Varlık-İlişki Şemasının Tablolara Dönüştürülmesi

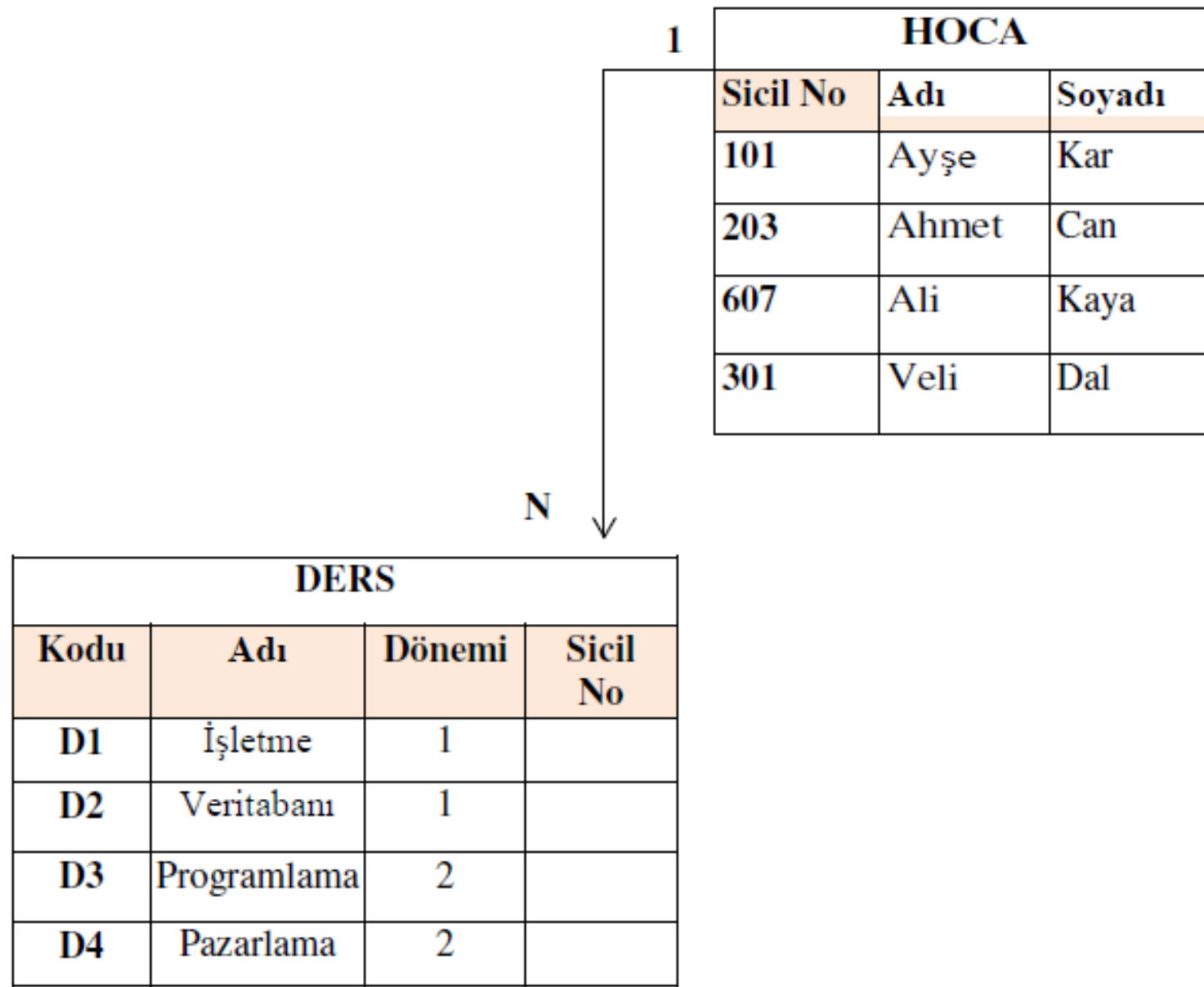
2. Bire-Çok (1-n) İlişkilerin Tablolara Dönüştürülmesi

İlişkiyi oluşturan varlık kümeleri tablolara dönüştürülür. İlişkinin n tarafındaki tabloya 1 tarafındaki tablonun anahtar alanı yabancı anahtar olarak eklenir. İlişkide belirtilen tanımlayıcı özellikler n tarafına alan olarak eklenirler.

Hoca ders varlıkları arasındaki bire bir ilişki ve tablo yapısı şekilde gözükmektedir(Şekil 14)



Şekil 10. Hoca Ders Verir Bire-Çok İlişkisi

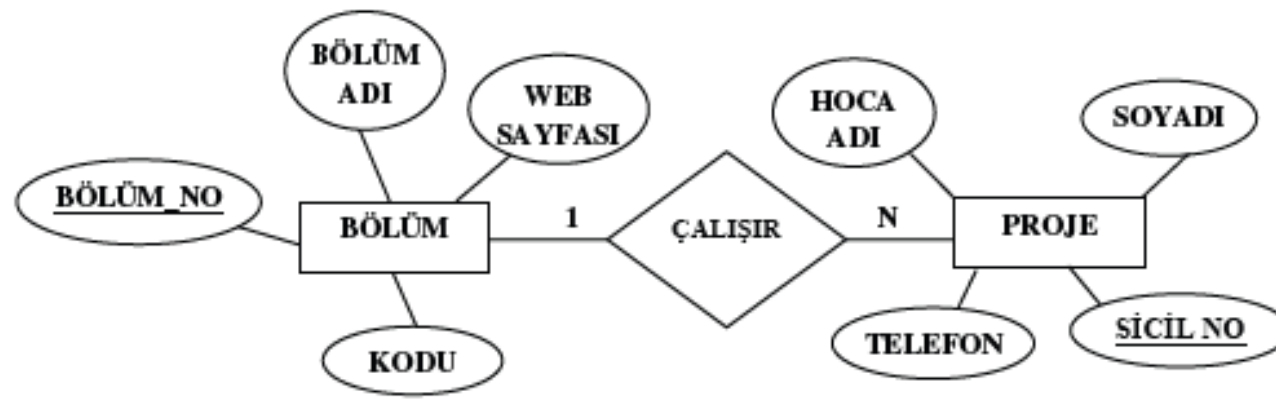


Şekil 14 Hocalarbirden fazla derse girmektedir.

Varlık-İlişki Şemasının Tablolara Dönüştürülmesi

2. Bire-Çok (1-n) İlişkilerin Tablolara Dönüştürülmesi

Örneğin Hoca bölüm varlık kümeleri arasındaki 1-n çalışır ilişkisinde Hoca ve bölüm varlık kümeleri tablolara dönüştürülür (Şekil 15). İlişkinin n tarafındaki tablo olan Hoca tablosuna bölüm tablosunun anahtar alanı olan **bölümno** alanı yabancı anahtar olarak eklenir.



BÖLÜM			
BölümNo	Adı	Kodu	Web Sayfası
1	Matematik	38	www.sak.edu.tr
2	Fizik	47	www.sak.edu.tr
3	Kimya	18	www.ist.edu.tr
4	Biyoloji	293	www.sak.edu.tr

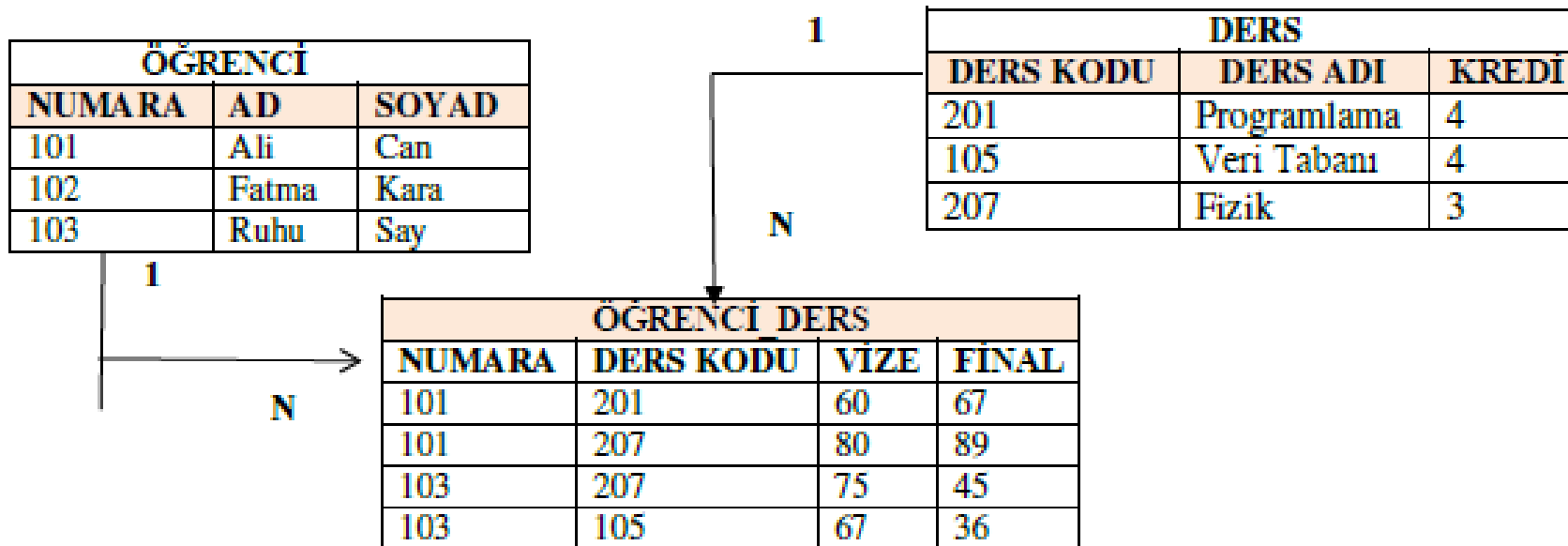
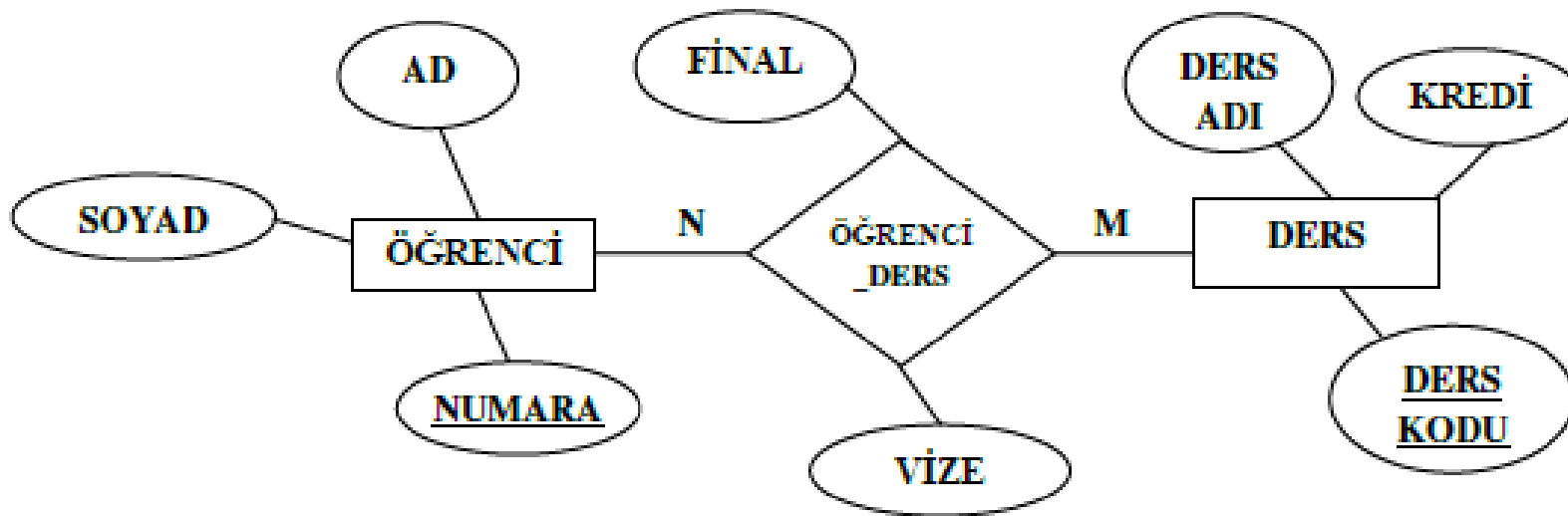
HOCA				
SicilNo	Adı	Soyadı	Telefon	Bölüm
546	Ali	Can	29298	2
342	Fatma	Kara	29245	1
124	Ruhu	Say	29284	3
432	Ali	Demir	29283	4
263	Hüseyin	Cemal	29283	4

Şekil 15 Hoca Bölüm Varlık Kümeleri Arasındaki 1-N Çalışır İlişkisi ve Oluşturulan Tablolar

Varlık-İlişki Şemasının Tablolara Dönüştürülmesi

3. Çoğa-Çok (N-M) İlişkilerin Tablolara Dönüştürülmesi

İlişkiyi oluşturan varlık kümeleri tablolara dönüştürülür. Ancak ilişki isminde yeni bir tablo oluşturulur. İlişkiyi oluşturan tabloların anahtar alanları yeni tabloya yabancı anahtar olarak eklenir. İlişkide belirtilen tanımlayıcı özellikler varsa yeni tabloya alan olarak eklenir. Yeni tablonun anahtar alanı ilişkiyi oluşturan tabloların yabancı anahtarlarından oluşan ikili veya daha fazla alandan oluşur. Diğer ifadeyle N-M ilişkiler için ayrı bir ara-tablo oluşturulur. Bu ara-tabloda anahtar alan iki tablonun anahtar alanlarının birleşimidir(Şekil 16).

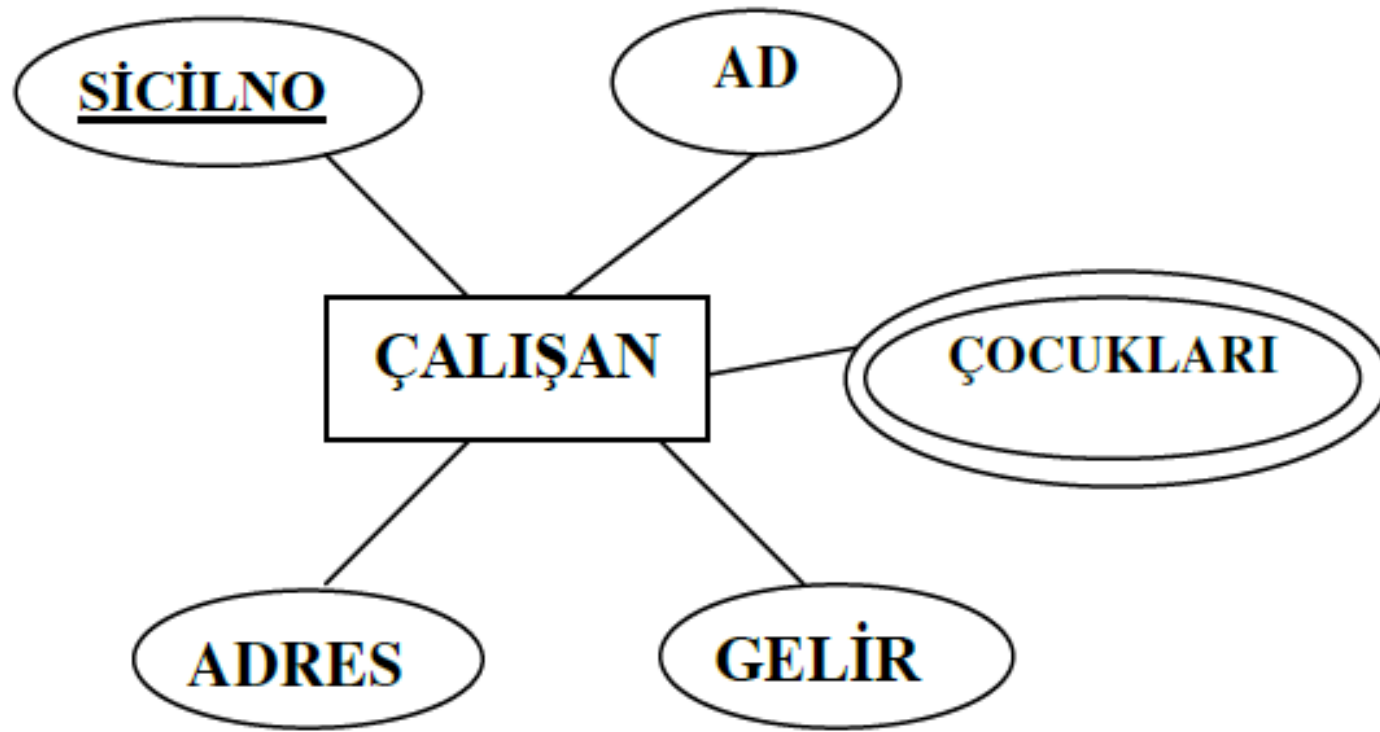


Şekil 16 Öğrenci Ve Dersin-M İlişkisi

Çok Değerli Özelliklerin Dönüştürülmesi

Çok değerli her bir özellik için yeni bir tablo oluşturulur. Yeni tabloya çok değerli alan adı ve çok değerli alanın bulunduğu ilk tablonun anahtar alanı eklenir. Bu yeni tablonun anahtar alanı ise, çok değerli alan ile birlikte eklenen yabancı anahtarın birleşimidir.

Örneğin şekil 17'deki çocukları özelliği çok değerli bir özelliktir. Çünkü her Çalışanın birden çok Çocukları olabilir. Bu durumda çocukları adında yeni bir tablo oluşturulur. Bu tablonun alanları olarak Çocukları ve Çalışan tablosunun anahtar alanı olan **sicilno** yabancı anahtar olarak eklenir. Bu tablonun anahtar alanı çocukları ve sicilno alanlarının birleşimidir.



Şekil 17. Çalışan Varlık Kümesinin Çoklu Değer Özelliği Ve Diğer Özellikleri

Çalışan varlık kümesi tabloya dönüştürülürse;

Çalışan (sicilNo, adres, ad, Gelir) Çocukları (Çocukları, sicilNo)

Çok Değerli Özelliklerin Dönüştürülmesi

Varlık-İlişki Diagramı hazırlayabilmek için hazır toollar vardır. Bunlardan bazıları **Microsoft Visio, Sybase Power Designer ve ERwin**dir. Bu tool sayesinde primary key, foreign key, index, view gibi unsurlar da gösterilebilmektedir. Varlık-İlişki diyagramları veritabanı tasarımının olmazsa olmazlarından. Özellikle büyük projelerde veriler arasındaki ilişkiler ve bu verilerin çözümlenmesi işlemleri çok karışık olmaktadır. Bu tür durumlarda Varlık-İlişki diyagramları tüm bu ilişkilerin açıkça belirtilmesi açısından önemlidir.
