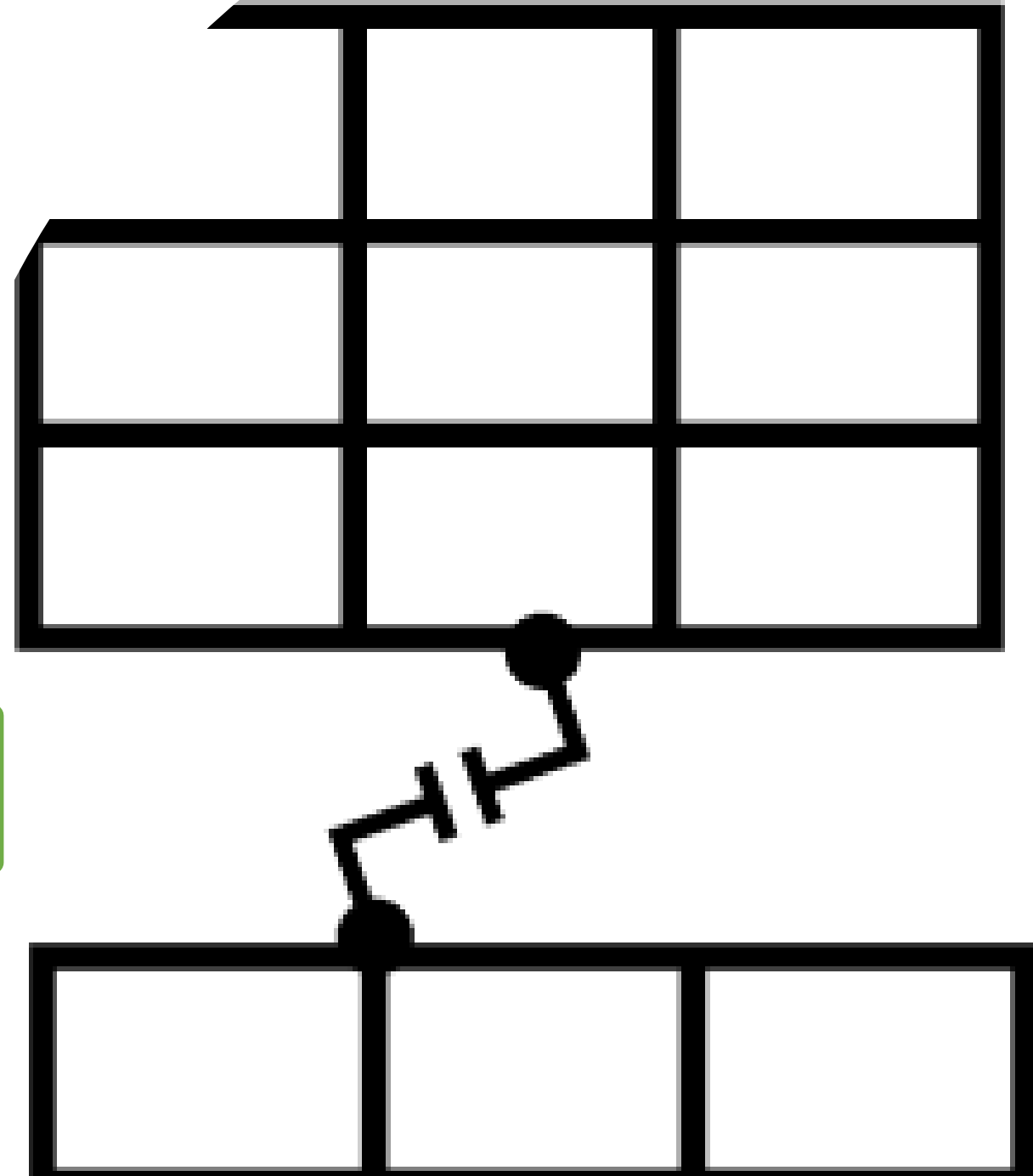
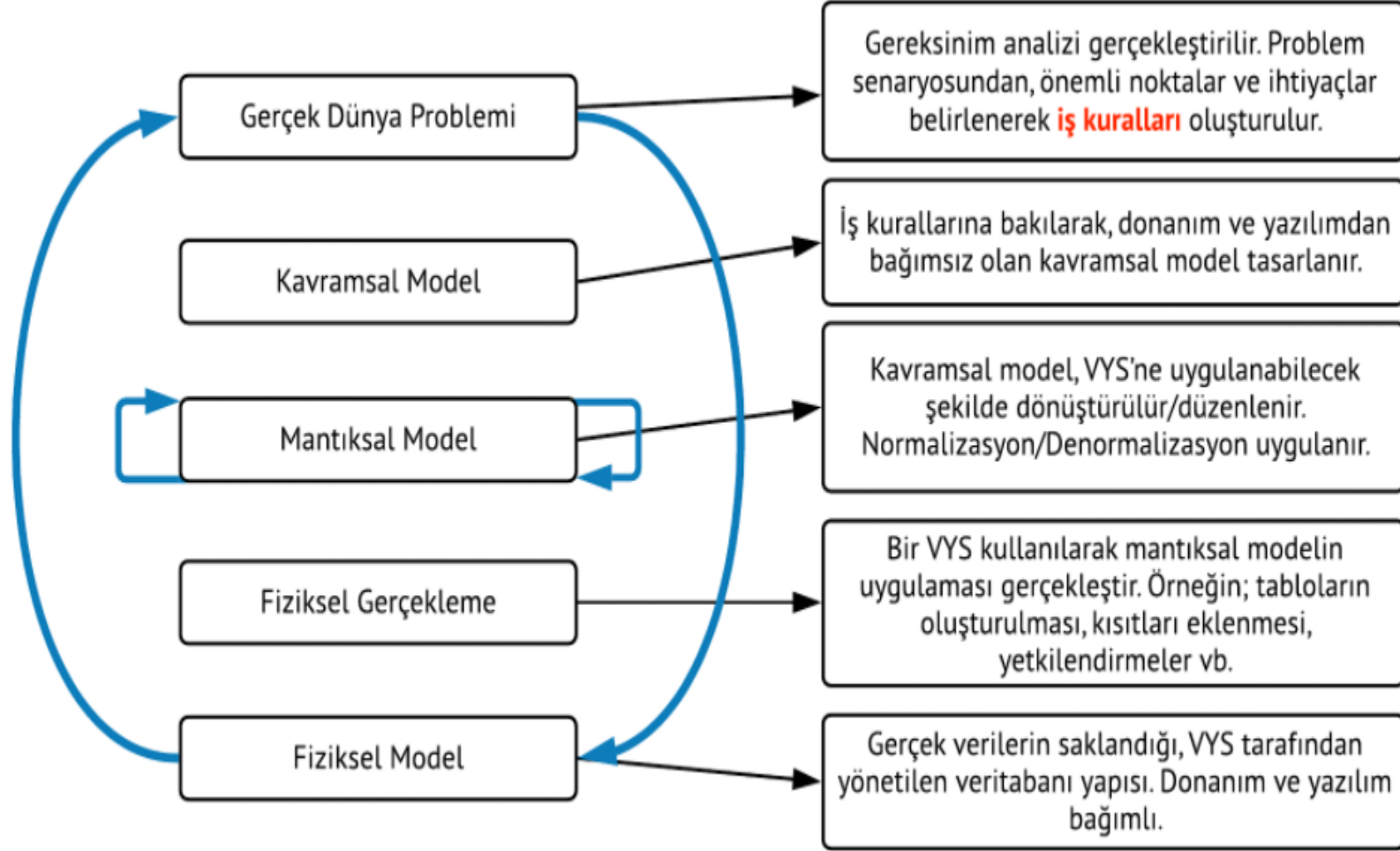


Veri Tabanı
Yönetim
Sistemleri

Hafta 4: İlişkisel Model





Veri Tabanı Geliştirme Yaşam Döngüsü

Tablo

- Tablo (table), satır(row) ve sütunlardan (column) oluşan iki boyutlu bir yapı olarak algılanır.
- İlişkisel veritabanının mantıksal görünümü, **ilişki(relation)** olarak bilinen mantıksal bir yapıya dayalı veri ilişkilerinin oluşturulmasıyla kolaylaştırılır.
- Bir ilişki matematiksel bir yapı olduğu için, son kullanıcılar ilişkiyi tablo olarak düşünmeyi çok daha kolay bulurlar.
- Tablo = ilişki.
- Her tablo satırı (kayıt/tuple), varlık kümesi (entity set) içerisindeki tek bir varlık (entity) oluşumunu temsil eder.
 - Örneğin, ÖĞRENCİ tablosu, her biri bir öğrenciyi temsil eden varlık oluşumlarının bir koleksiyonunu içerir. Bu nedenle, varlık kümesi ve tablo terimleri genellikle birbirinin yerine kullanılır.



Tablo Karakteristikleri

- Bir tablo (table); satır (row) ve sütunlardan (column) oluşan iki boyutlu bir yapı olarak algılanır.
 - Her tablo satırı (kayıt/tuple), varlık kümesi (entity set) içerisindeki tek bir varlık (entity) oluşumunu temsil eder.
 - Her tablo sütunu bir niteliği (attribute) temsil eder ve her bir sütun farklı bir isme sahiptir.
 - Her satır-sütun kesişimi tek bir veri değerini temsil eder.
 - Bir sütun içerisinde yer alan tüm değerler aynı veri tipine sahip olmalıdır.
 - Her sütun, nitelik değer alanı (attribute domain) olarak bilinen kesin bir değerler aralığına sahiptir.
 - Satır ve sütunların sırası Veritabanı Yönetim Sistemi (VTYS/DBMS) tarafından bakıldığında önemsizdir.
 - Her tablo, her bir satırını eşsiz bir şekilde tanımlayan bir niteliğe veya nitelik birleşimine sahip olmak zorundadır. Buna birincil anahtar ismi verilir.
-

```

SELECT TOP (1000) [Student_ID]
, [Std_Name]
, [Std_Surname]
, [Std_BirthDate]
, [Std_Email]
, [Std_Phone]
, [Std_Address]
, [City]
, [Std_Gender]
FROM [Slayt].[dbo].[TB_Students]

```

Örnek Tablo

(TB_Students)

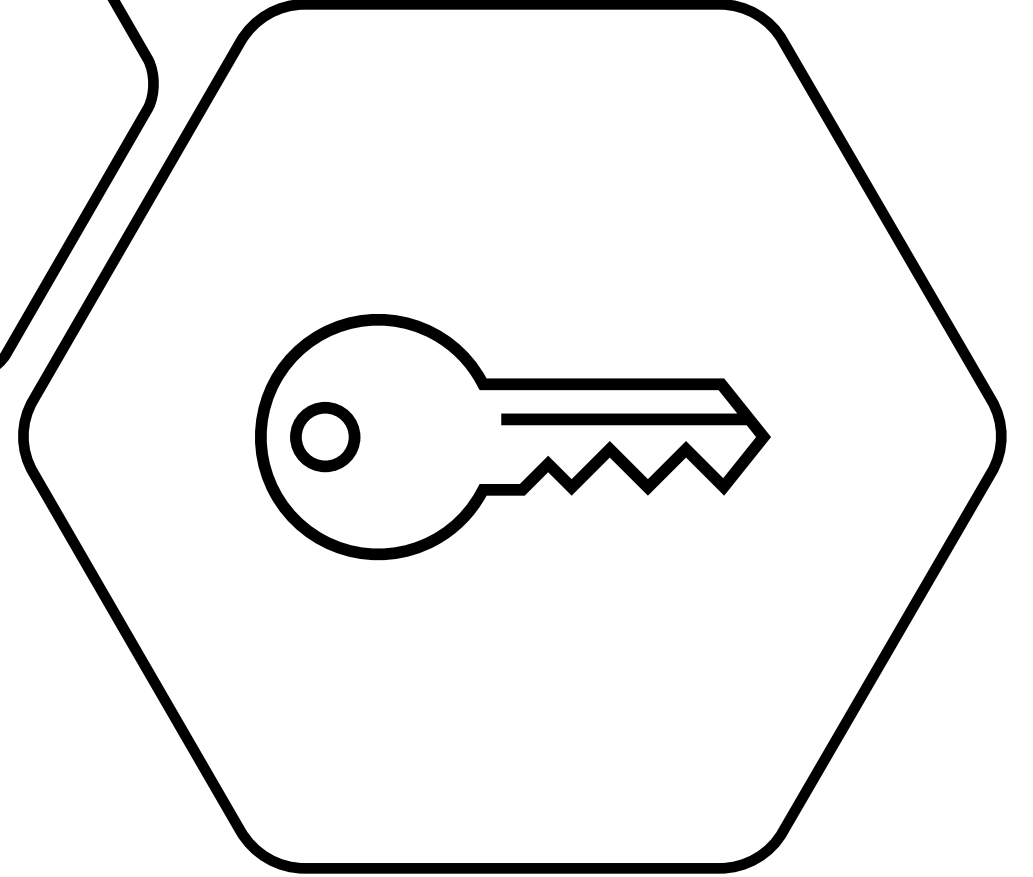
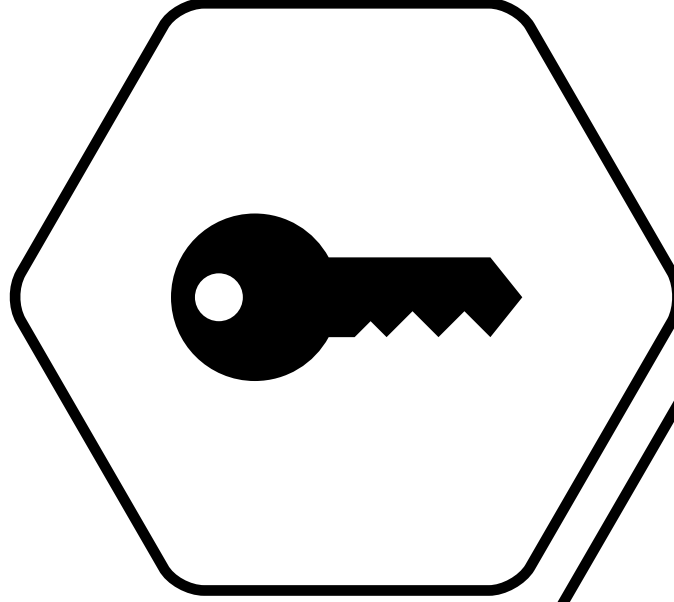
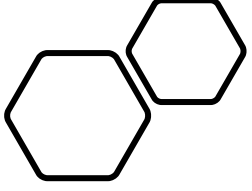
	Student_ID	Std_Name	Std_Surname	Std_BirthDate	Std_Email	Std_Phone	Std_Address	City	Std_Gender
	000000001	Elvan	Çeber	2000-08-12	eceber@gmail....	123456789	Erenler	54	K
	000000002	Hüseyin	Demirci	1993-06-03	hdemirci@gma...	545465211	Yazlık	42	E
	100221002	Ahmet	Erengil	1999-12-07	erengil@sakary...	432345453	Acıpayam	20	E
	123343544	Esra	Akar	2001-07-04	eakar@sakarya....	544355434	İstiklal Mah.	81	K
	123443545	Ömer Faruk	Seymen	1988-03-05	ofseymen@gm...	114477889	Merkez	54	E
	543434434	Burcu	Yavuz	1999-06-04	byavuz@gmail....	435453454	Merkez	54	K
	433222311	Yağmur	Aydın	2000-07-06	yaydin@gmail....	545423453	Keçiören	6	K
	544212134	İrfan	Kısmet	2000-07-03	ikismet@sakary...	332221111	Merkez	49	E

	Student_ID	Std_Name	Std_Surname	Std_BirthDate	Std_Email	Std_Phone	Std_Address	City	Std_Gender
	000000001	Elvan	Çeber	2000-08-12	eceber@gmail....	123456789	Erenler	54	K
	000000002	Hüseyin	Demirci	1993-06-03	hdemirci@gma...	545465211	Yazlık	42	E
	100221002	Ahmet	Erengil	1999-12-07	erengil@sakary...	432345453	Acıpayam	20	E
	123343544	Esra	Akar	2001-07-04	eakar@sakarya....	544355434	İstiklal Mah.	81	K
	123443545	Ömer Faruk	Seymen	1988-03-05	ofseymen@gm...	114477889	Merkez	54	E
	543434434	Burcu	Yavuz	1999-06-04	byavuz@gmail....	435453454	Merkez	54	K
	433222311	Yağmur	Aydın	2000-07-06	yaydin@gmail....	545423453	Keçiören	6	K
	544212134	İrfan	Kısmet	2000-07-03	ikismet@sakary...	332221111	Merkez	49	E

- Yukarıdaki Öğrenciler Tablosu (TB_Students) ele alınırsa:
- Öğrenciler tablosu, 8 satır (kayıt) ve 9 sütundan (nitelik) oluşan bir iki boyutlu yapı olarak görülür.
- Öğrenciler tablosundaki her satır, varlık kümesindeki bir varlığı tanımlar (Varlık kümesi, Öğrenciler tablosu ile temsil edilir).
 - Örneğin 1. satır, Elvan Çeber isimindeki bir öğrenciyi tanımlar.
- Her sütun, bir niteliği temsil eder ve her sütunun ayrı bir ismi mevcuttur.
- Bir sütundaki tüm değerler, niteliğin karakteristikleri ile uyumludur.
 - Örneğin, dogumTarihi sütunu, tüm satırlar için doğum tarihi bilgilerini içerir.

	Student_ID	Std_Name	Std_Surname	Std_BirthDate	Std_Email	Std_Phone	Std_Address	City	Std_Gender
	000000001	Elvan	Çeber	2000-08-12	eceber@gmail....	123456789	Erenler	54	K
	000000002	Hüseyin	Demirci	1993-06-03	hdemirci@gma...	545465211	Yazlık	42	E
	100221002	Ahmet	Erengil	1999-12-07	erengil@sakary...	432345453	Acıpayam	20	E
	123343544	Esra	Akar	2001-07-04	eakar@sakarya....	544355434	İstiklal Mah.	81	K
	123443545	Ömer Faruk	Seymen	1988-03-05	ofseymen@gm...	114477889	Merkez	54	E
	543434434	Burcu	Yavuz	1999-06-04	byavuz@gmail....	435453454	Merkez	54	K
	433222311	Yağmur	Aydın	2000-07-06	yaydin@gmail....	545423453	Keçiören	6	K
	544212134	İrfan	Kismet	2000-07-03	ikismet@sakary...	332221111	Merkez	49	E

- Veriler, biçimine ve fonksiyonuna göre sınıflandırılmalıdır. VTYS'lerin desteklediği temel veri tipleri:
 - Nümerik (Numeric): Üzerinde anlamlı aritmetik işlemler yapılabilen veri. Örneğin not ortalamasının saklandığı bir alan nümerik tiptedir.
 - Karakter (Character): Metin verisi, karakter dizisi. Aritmetik işlemler için tanımlanmamış olan karakter ve semboller. Örneğin adı, soyadı, telefon numarası vb. bilgilerin saklandığı alanlar.
 - Tarih (Date): Tarih verisi. Sezar tarihi (Julian date) formatında takvim tarihlerini ihtiva eden veridir. Örneğin doğum tarihi, bilgisinin saklandığı alan tarih tipindedir.
 - Mantıksal (Logical): Doğru ya da yanlış (evet veya hayır) değerleri.
- Bir sütun için izin verilen değerler aralığına, o sütunun değer alanı (domain) adı verilir.
 - City alanı 1-81 aralığında değer alabilir. [1,81].
 - Örneğin tablodaki öğrenci not bilgileri 0-4 aralığında sınırlandırılmış olduğundan ötürü, bu sütunun değer aralığı [0,4] olarak ifade edilir.
- Student_ID alanı birincil anahtardır.



Anahtar (Keys)

- **Anahtar:** Kayıtların ayırt edilebilmesini (süper, aday, birincil, ikincil ve vekil anahtarlar) ve/veya tablolar arasında ilişki kurulabilmesini (birincil, yabancı anahtarlar) sağlayan yapı.
- İlişkisel modelde anahtarlar önemlidir.
- Bir tablodaki her satırın benzersiz bir şekilde tanımlanabilir olmasını sağlamak.
- Tablolar arasında ilişkiler kurmak ve verilerin bütünlüğünü sağlamak.
- Anahtarlar belirlenirken **fonksiyonel bağımlılık** kavramının bilinmesi gerekir.

Anahtar (Keys) – Fonksiyonel Bağımlılık

- Bir anahtarın rolü, bir özelliğin değerini bilmenin diğer özelliğin değerini belirlemeyi mümkün kıldığı durumlara dayalıdır.
- Örnekte, Student_ID alanı kullanılarak öğrenci adı belirlenebilir. Tersisi doğru değildir.
- Bu durumda:
 - Student_ID alanı, Std_Name alanını belirler.
 - Std_Name alanı, Student_ID alanına **fonksiyonel bağımlıdır**.
 - Student_ID → Std_Name şeklinde gösterilir.

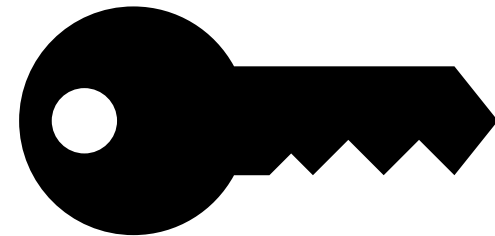
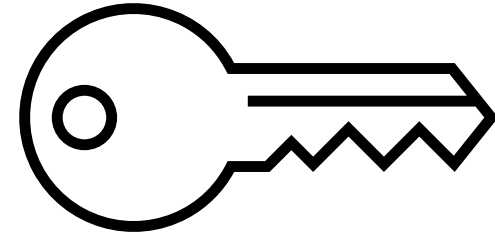
Anahtar (Keys) – Anahtar Çeşitleri

 **Süper Anahtar:** Herbir satırı eşsiz olarak tanımlayabilen nitelik veya nitelik gruplarına süper anahtar denir. Bir ya da birden çok alanı içerebilir.

- Student_ID
- Student_ID, Std_Name
- Student_ID, Std_Name, Std_Surname , ...

 **Aday Anahtar:** Herbir satırı eşsiz olarak tanımlayabilen ve en az sayıda alana sahip olan süper anahtara aday anahtar denir.

- “Student_ID, Std_Name” ikilisi süper anahtardır. Sadece StudentID alanı kayıtları eşsiz olarak tanımlayabildiğinden dolayı “Student_ID, Std_Name” ikilisi aday anahtar değildir.



Anahtar (Keys) – Anahtar Çeşitleri



Birincil Anahtar (Primary Key, PK): Herbir satırı eşsiz olarak tanımlayabilen ve aday anahtarlar içerisinde seçilen anahtar. Birincil anahtar eşsiz değerlere sahip olmalıdır, NULL değerler alamaz.

- öğrenciNo, kimlikNo, pasaportNo alanları aday anahtarlardır. Her biri birincil anahtar olarak seçilebilir (sadece biri seçilmelidir).

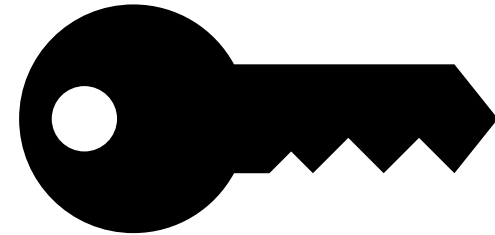
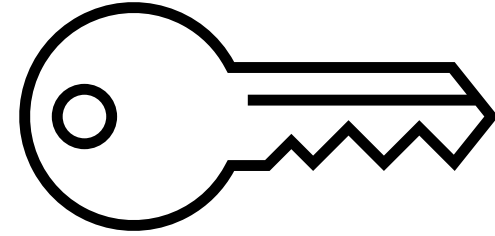


İkincil/Alternatif Anahtar: Birincil anahtar olarak seçilmeyen diğer aday anahtarlara ikincil/alternatif anahtar denir.



Vekil Birincil Anahtar: Birincil alan fonksiyonlarını yerine getirmek için eklenen alandır.

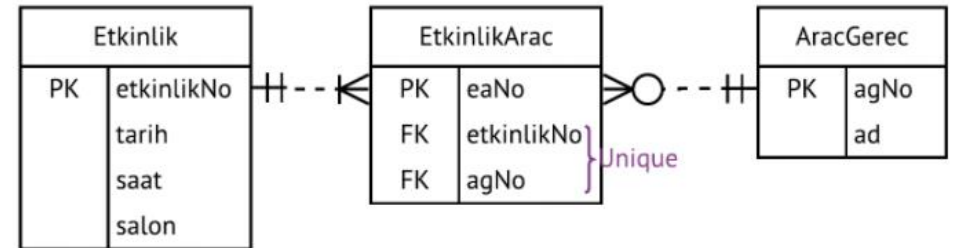
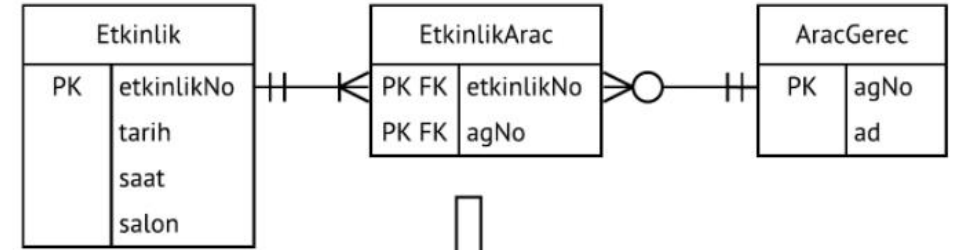
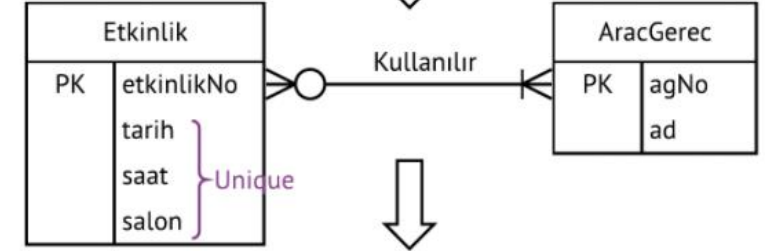
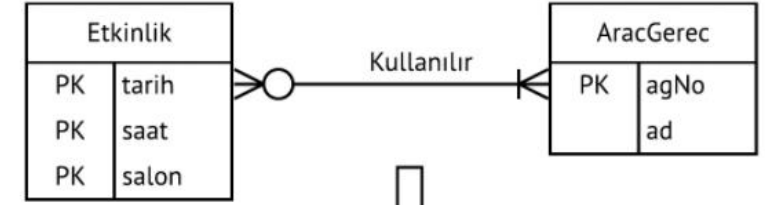
- Uygulamada çoğunlukla vekil birincil anahtarlar kullanılır.



Anahtar (Keys) – Anahtar Çeşitleri

- **Vekil Birincil Anahtar:** Birincil alan fonksiyonlarını yerine getirmek için eklenen alandır.

tarih	saat	salon
1.1.2000	10.00	1
1.1.2000	11.00	1
2.1.2000	10.00	1
2.1.2000	11.00	1
2.1.2000	11.00	2



Anahtar (Keys) – Anahtar Çeşitleri

Yabancı Anahtar (Foreign Key, FK):

- Bağıntı kurulan tablo içerisinde yer alan ve bağıntı kuran tablodaki birincil anahtarın değerlerine benzeyen değerler içeren alan.
- İki tablo arasında bağıntı kurulurken bağıntı kuran tablonun birincil anahtarı ile bağıntı kurulan tablonun yabancı anahtarı kullanılır. Dolayısıyla iki anahtar da aynı tipte olmalıdır.

Database: OğrenciBilgiSistemi Schema: public Table: Oğrenci

	id	ogrenciNo	adi	soyadi	bolum	danisman
3	3	B151212322	Ayşe	Güler	1	4
4	4	B111212134	Ayşen	Şahin	1	3
5	5	B101212320	Ahmet	Mert	1	3
6	6	B101212322	Ayşe	Mert	1	3
7	7	B111212323	Mehmet	Aydın	1	3
8	8	B131212122	Buse	Yılmaz	2	3
9	9	B121212134	Ayşen	Aydın	1	3
10	10	B111212321	Berkay	Aydın	1	3
11	11	B131212123	Berat	Yılmaz	2	3
12	12	B121212135	Aylin	Aydın	1	4
13	13	B151212349	Mehmet	Gen.	<NULL>	<NULL>
14	14	B151212350	Ali	Katı	<NULL>	<NULL>
15	15	B151212351	Alican	Konyalı	<NULL>	<NULL>

Related Table: AkademikPersonel : Ink_Oğrenci_AkademikPersonel (M:1) Mode: show all Set: OR (unic

	id	personelNo	adi	soyadi	bolum	sifre
1	3	A1	Ahmet	Yılmaz	1	c4ca4238a0b923820dcc509a6f75849b
2	4	A2	Umut	Işık	1	c81e728d9d4c2f636f067f89cc14862c

Oğrenci

- adi Character ...ying[50]:I
- bolum Integer:N
- danisman Integer:N
- id Integer:I:U
- ogrenciNo Character[10]:I
- soyadi Characte...ng[50]:I

Indexes (3)

AkademikPersonel

- adi Character ...ying[40]:I
- bolum Integer
- id Integer:I:U
- personelNo Chara...g[10]
- sifre Character[128]:N
- soyadi Characte...ng[40]:I

Indexes (2)

FIGURE 3.2 AN EXAMPLE OF A SIMPLE RELATIONAL DATABASE

Table name: PRODUCT Database name: Ch03_SaleCo
Primary key: PROD_CODE
Foreign key: VEND_CODE

PROD_CODE	PROD_DESCRIPTION	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Housette chain saw, 16-in. bar	189.99	4	235
GER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-557JG	Rat-tail file	2.99	15	232
ZZK0245Q	Steel tape, 12-ft. length	6.79	6	235

link

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREA CODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	603	224-2134
233	Candice Wallace	904	342-5567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Bütünlük Kuralları (Integrity Rules)

- İlişkisel veritabanı bütünlüğü kuralları, iyi bir veritabanı tasarımı için çok önemlidir. RDBMS'ler bütünlük kurallarını otomatik olarak uygular, ancak uygulama tasarımınızın bu bölümde bahsedilen varlık ve referans bütünlüğü kurallarına uygun olduğundan emin olmak çok daha güvenlidir.
- **Varlık Bütünlüğü (Entity Integrity):** Varlık bütünlüğünün sağlanabilmesi için her tabloda birincil anahtar (PK) olmalıdır.
 - Her varlığın "not null" kısıtını sağlayan eşsiz ("unique" kısıtı) bir tanımlayıcısı olması garanti edilir.
 - Varlık bütünlüğü kullanılarak, örneğin; aynı öğrenci numarasına sahip çok sayıda öğrencinin ya da öğrenci numarası olmayan öğrencinin bulunamaması garanti edilir.

Bütünlük Kuralları (Integrity Rules)

- **Referans Bütünlüğü (Referential Integrity):** Referans bütünlüğünün sağlanabilmesi için yabancı anahtar(FK)tanımlanır.
 - Yabancı anahtar alanı ya NULL olabilir ya da bağıntı kuran tablodaki birincil anahtar alanı değerlerinden biri olabilir.
 - Her yabancı anahtar değerinin diğer tablonun birincil anahtarı değeriyle tutarlı olması garanti edilir.
 - Referans bütünlüğü kullanılarak, örneğin; öğrencinin açılmayan bir dersi alması, derse kayıtlı bir öğrencinin okuldan kaydının silinmesi, olmayan bir ürünün sipariş edilmesi vs. önlenir.
-



Bütünlük Kuralları (Integrity Rules)

- **NOT NULL (Boş Olamaz):** Alan içerisine değer yazılmasının zorlanması isteniyorsa «NOT NULL» yapılabilir. Örneğin, kişinin adının zorunlu olarak girilmesi bu şekilde sağlatılabilir.
 - **UNIQUE (Eşsiz):** Alan içerisine yazılan değerlerin her satır için eşsiz olması isteniyorsa kullanılır. Örneğin, kişilerin TC Kimlik numaralarının aynı yazılamaması bu şekilde garanti edilir.
-

Bütünlük Kuralları (Integrity Rules)

FIGURE 3.3 AN ILLUSTRATION OF INTEGRITY RULES

Table name: CUSTOMER Database name: Ch03_InsureCo
Primary key: CUS_CODE
Foreign key: AGENT_CODE

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramos	Alfred	A	05-Apr-2016	502
10011	Dunne	Leona	K	16-Jun-2016	501
10012	Smith	Kathy	W	29-Jan-2017	502
10013	Olowski	Paul	F	14-Oct-2016	
10014	Orlando	Myron		28-Dec-2016	501
10015	O'Brian	Amy	B	22-Sep-2016	503
10016	Brown	James	G	25-Mar-2017	502
10017	Williams	George		17-Jul-2016	503
10018	Farriss	Anne	O	03-Dec-2016	501
10019	Smith	Clotte	K	14-Mar-2017	503

Table name: AGENT (only five selected fields are shown)
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	132735.75
502	615	882-1244	Hahn	138367.35
503	615	123-5583	Okon	127093.45

Database: OğrenciBilgiSistemi Schema: public Table: Oğrenci

	id	ogrenciNo	adi	soyadi	bolum	danisman
3	3	B151212322	Ayşe	Güler	1	4
4	4	B111212134	Ayşen	Şahin	1	3
5	5	B101212320	Ahmet	Mert	1	3
6	6	B101212322	Ayşe	Mert	1	3
7	7	B111212323	Mehmet	Aydın	1	3
8	8	B131212122	Buse	Yılmaz	2	3
9	9	B121212134	Ayşen	Aydın	1	3
10	10	B111212321	Berkay	Aydın	1	3
11	11	B131212123	Berat	Yılmaz	2	3
12	12	B121212135	Aylin	Aydın	1	4
13	13	B151212349	Mehmet	Gen.	<NULL>	<NULL>
14	14	B151212350	Ali	Katı	<NULL>	<NULL>
15	15	B151212351	Alican	Konyalı	<NULL>	<NULL>

15 from 43

Related Table: AkademikPersonel : Ink_Oğrenci_AkademikPersonel (M:1) Mode: show all Set: OR (unic

	id	personelNo	adi	soyadi	bolum	sifre
1	3	A1	Ahmet	Yılmaz	1	c4ca4238a0b923820dcc509a6f75849b
2	4	A2	Umut	Işık	1	c81e728d9d4c2f636f067f89cc14862c

Bütünlük Kuralları (Integrity Rules)

- **Varlık Bütünlüğü Uygulama Notu**

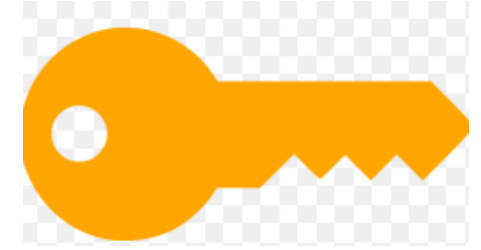
- Birincil ahatar (CUS_CODE ve id alanları) Null değer içermemektedir. Ayrıca tüm değerleri Unique özelliğe sahiptir.
- Aynı şekilde bağlantı kurulan tablonun birincil anahtarları (AGENT_CODE ve id alanları) null değer içermemektedir.

- **Referans Bütünlüğü Uygulama Notu**

- Öğrenciler tablosu «danisman» adında yabancı anahtar içermektedir, böylece Öğrenci Tablosu – Danışman Tablosu bağlantısı sağlanır.
- Müşteri tablosu «Agent_Code» adında yabancı anahtar içermektedir, böylece CUSTOMER-AGENT bağlantısı sağlanır.
- Referans bütünlüğünün sağlanması için yabancı anahtar alanı NULL olabilir.
- NULL değer sorgularda sorunlara neden olabileceğinden bazı tasarımcılar tarafından genellikle NULL değer yerine özel değerler,kodlar tercih edilir. (Flags-Bayrak)
 - Örneğin; il bilgisi olarak NULL yerine “00: Bilinmiyor” verilebilir. Bu durumda İller tablosunda 00, Bilinmiyor değerlerine sahip bir satır bulunmalıdır. Kisiler tablosundaki il alanının varsayılan değeri de 00 yapılabilir. Böylece, kişinin il bilgisi girilmediği zaman varsayılan olarak bilinmiyor bilgisi atanmış olur.

- **Flags:** Tasarımcılar tarafından gerekli bir yanıtı tetiklemek, son kullanıcıları belirtilen koşullar konusunda uyarmak veya değerleri kodlamak için uygulanan özel kodlar. Bayraklar, bir tablodaki değerin yokluğuna dikkat çekerek null değerleri önlemek için kullanılabilir.

Birincil Anahtar Özellikleri



UNIQUE (kayıtları eşsiz olarak tanımlayabilmesi) ve NOT NULL (boş olmaması) bütünlük kurallarını sağlamalı.



Zamanla değişme ihtimali olan alanlar tercih edilmemeli.



Mümkün olduğunca tek bir alandan oluşmalı.

Çok alandan oluşursa bağıntıları kurmak, sorguları yazmak karmaşıklaşır. Gecikme ve kaynak kullanımı artar.



Sayısal alanlar tercih edilmeli.

İşlem hızı ve otomatik artma (auto increment) desteği sağlanabilir.



Güvenlik riski bulunan alanlar tercih edilmemeli.

Yazılımlarda yoğun olarak birincil anahtar değeri kullanılır.

Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

- Her varlık için bir tablo (ilişki/relation) oluşturulur.

Ogrenci	
PK	ogrenciNo
FK	bolumNo
	ad
	soyad

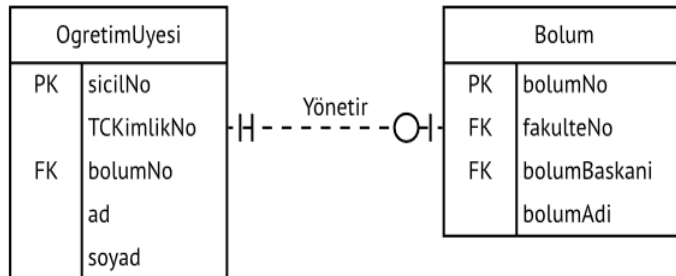
İlişkisel Şema (Metinsel Gösterim)

- Ogrenci(ogrenciNo: String, bolumNo: Integer, ad

Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

- **Bir-Bir Bağntısı**
 - 1 öğretim üyesi **en çok 1** bölüm yönetebilir.
 - 1 bölüm yalnızca **en çok 1** öğretim üyesi tarafından yönetilebilir.

Yalnızca 1 tarafının (en az 1 - en çok 1) varlığının (OgretimUyesi) birincil anahtarı (sicilNo) diğer varlığa yabancı anahtar alan (bolumBaskani) olarak eklenir.



- **İlişkisel Şema (Metinsel Gösterim)**

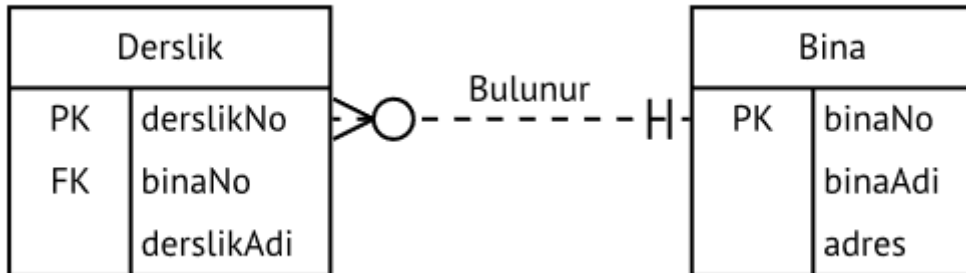
- OgretimUyesi(sicilNo: String, TCKimlikNo: Char(11), bolumNo: Integer, ad: String, soyad: String)
- Bolum(bolumNo: Integer, fakulteNo: Integer, bolumBaskani: String, bolumAdi: String)

Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

- **Bir-Çok Bağntısı**

- 1 derslik **en çok 1** binada bulunur.
- 1 binada **en çok çok** sayıda derslik bulunur.

1 tarafının varlığının (Bina) birincil anahtarı (binaNo) diğer varlığa yabancı anahtar alan (binaNo) olarak eklenir



- **İlişkisel Şema (Metinsel Gösterim)**

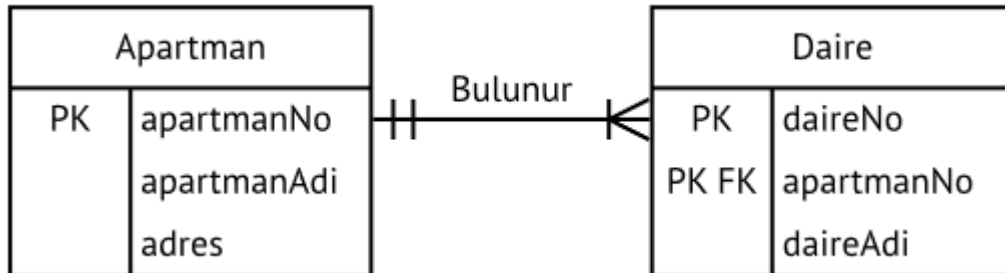
- Derslik(derslikNo: Integer, binaNo: Integer, derslikAdi: String)
- Bina(binaNo: Integer, binaAdi: String, adres: String)

Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

• Bir-Çok Bağntısı / Varolma Bağımlılığı, Tanımlama Bağntısı

- 1 apartmanda **en çok çok** sayıda daire bulunur. En az bir daire bulunmak zorundadır.
- 1 daire **en çok 1** apartmanda (**en az 1 – en çok 1**) bulunmak zorundadır.

1 tarafının (Apartman) birincil anahtarı (apartmanNo) diğer varlığa hem yabancı anahtar hem de birincil anahtarın parçası (apartmanNo) olarak eklenir.



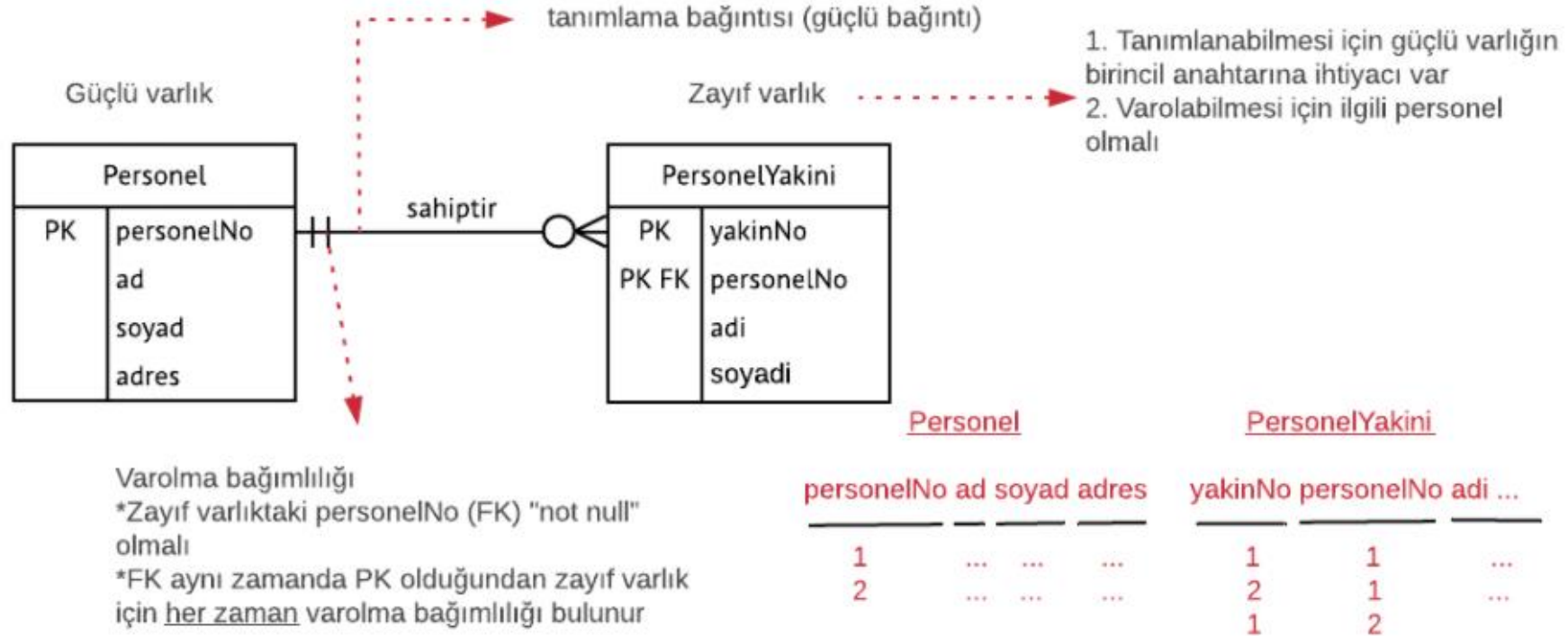
• İlişkisel Şema (Metinsel Gösterim)

- Apartman(apartmanNo: Integer, apartmanAdi: String, adres: String)
- Daire(daireNo: Integer, apartmanNo: Integer, daireAdi: String)

Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

• Bir-Çok Bağntısı / Varolma Bağımlılığı, Tanımlama Bağntısı

- Var olma Bağımlılığı, Tanımlama Bağntısı



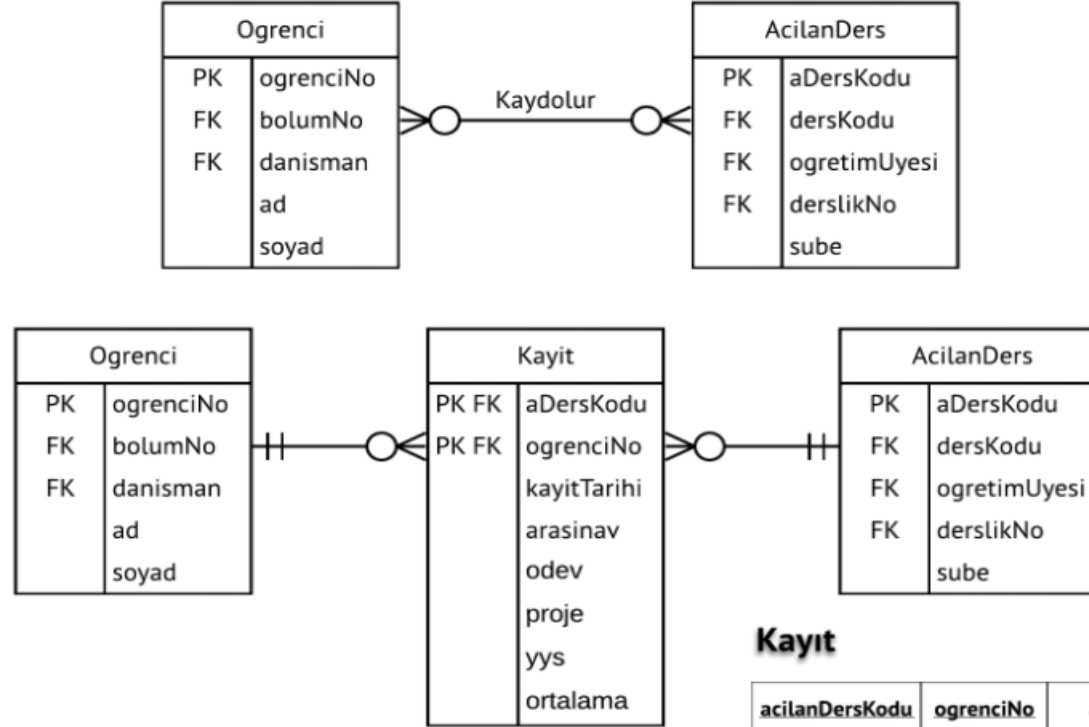
Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

• Çok-Çok Bağntısı

- 1 öğrenci **en çok çok** sayıda derse kayıt yaptırabilir.
- 1 ders **en çok çok** sayıda öğrenci tarafından alınabilir.

• İlişkisel Şema (Metinsel Gösterim)

- Öğrenci(ogrenciNo: String, bolumNo: Integer, danisman: String, ad: String, soyad: String)
- Kayit(aDersKodu: String, ogrenciNo: String, kayıtTarihi: Date, not: Float)
- AcilanDers(aDersKodu: String, dersKodu: String, ogretimUyesi: String, derslikNo: Integer, sube: Char(1))



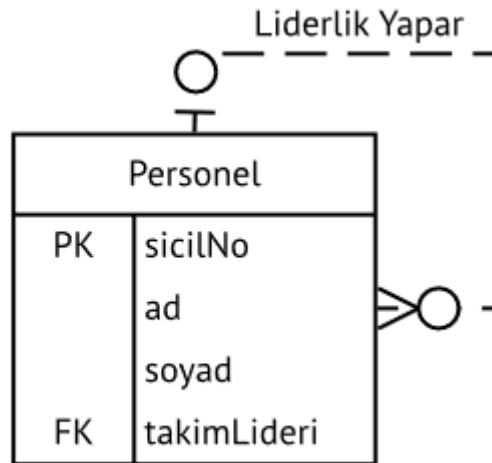
Kayıt

acilanDersKodu	ogrenciNo	arasinav	odev	proje ...
1	1	80	21	100
1	2	90	56	90
3	1	90	75	40
3	2	50	67	95
3	3	40	75	100

Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

- **Tekli Bağntı**

- 1 personele **en çok 1** personel liderlik yapar.
- 1 personel **en çok çok** sayıda personele liderlik yapar.



- İlişkisel Şema (Metinsel Gösterim)

- Personel(sicilNo: String, ad: String, soyad: String, takimLideri: String)

Varlık Bağını Modelinin İlişkisel Modele Dönüştürülmesi

- 1:M ve 1:1
- Örnek Tablo

FIGURE 3.22 THE IMPLEMENTED 1:1 RELATIONSHIP BETWEEN PROFESSOR AND DEPARTMENT

Table name: PROFESSOR

Primary key: EMP_NUM

Foreign key: DEPT_CODE

Database name: Ch03_TinyCollege

EMP_NUM	DEPT_CODE	PROF_OFFICE	PROF_EXTENSION	PROF_HIGH_DEGREE
103	HIST	DRE 156	8783	Ph.D.
104	ENG	DRE 102	5561	MA
105	ACCT	KLR 229D	8665	Ph.D.
106	MKT/MGT	KLR 126	3099	Ph.D.
110	BIOL	AAK 160	3412	Ph.D.
114	ACCT	KLR 211	4438	Ph.D.
155	MATH	AAK 201	4440	Ph.D.
160	ENG	DRE 102	2248	Ph.D.
162	CIS	KLR 203E	2359	Ph.D.
191	MKT/MGT	KLR 409B	4018	DBA
195	PSYCH	AAK 297	3550	Ph.D.
209	CIS	KLR 333	3421	Ph.D.
229	CIS	KLR 300	3000	Ph.D.
297	MATH	AAK 194	1145	Ph.D.
299	ECON/FIN	KLR 284	2851	Ph.D.
301	ACCT	KLR 244	4683	Ph.D.
335	ENG	DRE 208	2000	Ph.D.
342	SOC	BBG 208	5514	Ph.D.
387	BIOL	AAK 230	8665	Ph.D.
401	HIST	DRE 156	8783	MA
425	ECON/FIN	KLR 284	2851	MBA
435	ART	BBG 105	2270	Ph.D.

The 1:M DEPARTMENT employs PROFESSOR relationship is implemented through the placement of the DEPT_CODE foreign key in the PROFESSOR table.

Table name: DEPARTMENT

Primary key: DEPT_CODE

Foreign key: EMP_NUM

The 1:1 PROFESSOR chairs DEPARTMENT relationship is implemented through the placement of the EMP_NUM foreign key in the DEPARTMENT table.

DEPT_CODE	DEPT_NAME	SCHOOL_CODE	EMP_NUM	DEPT_ADDRESS	DEPT_EXTENSION
ACCT	Accounting	BUS	114	KLR 211, Box 52	3119
ART	Fine Arts	A&SCI	435	BBG 185, Box 128	2278
BIOL	Biology	A&SCI	367	AAK 230, Box 415	4117
CIS	Computer Info. Systems	BUS	209	KLR 333, Box 56	3245
ECON/FIN	Economics/Finance	BUS	299	KLR 284, Box 83	3126
ENG	English	A&SCI	160	DRE 102, Box 223	1004
HIST	History	A&SCI	103	DRE 156, Box 264	1067
MATH	Mathematics	A&SCI	297	AAK 194, Box 422	4234
MKT/MGT	Marketing/Management	BUS	106	KLR 126, Box 55	3342
PSYCH	Psychology	A&SCI	195	AAK 297, Box 438	4110
SOC	Sociology	A&SCI	342	BBG 208, Box 132	2000

Varlık Bağntı Modelinin İlişkisel Modele Dönüştürülmesi

- M:N Örnek Tablo

FIGURE 3.24 THE WRONG IMPLEMENTATION OF THE M:N RELATIONSHIP BETWEEN STUDENT AND CLASS

Table name: STUDENT
Primary key: STU_NUM
Foreign key: none

Database name: Ch03_CollegeTry

STU_NUM	STU_LNAME	CLASS_CODE
321452	Bowser	10014
321452	Bowser	10018
321452	Bowser	10021
324257	Smithson	10014
324257	Smithson	10018
324257	Smithson	10021

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: STU_NUM

CLASS_CODE	STU_NUM	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	321452	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10014	324257	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	321452	CIS-220	2	MMWF 9:00-9:50 a.m.	KLR211	114
10018	324257	CIS-220	2	MMWF 9:00-9:50 a.m.	KLR211	114
10021	321452	GM-261	1	MMWF 8:00-8:50 a.m.	KLR200	114
10021	324257	GM-261	1	MMWF 8:00-8:50 a.m.	KLR200	114



FIGURE 3.25 CONVERTING THE M:N RELATIONSHIP INTO TWO 1:M RELATIONSHIPS

Table name: STUDENT
Primary key: STU_NUM
Foreign key: none

Database name: Ch03_CollegeTry2

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Table name: ENROLL
Primary key: CLASS_CODE + STU_NUM
Foreign key: CLASS_CODE, STU_NUM

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MMWF 9:00-9:50 a.m.	KLR211	114
10021	GM-261	1	MMWF 8:00-8:50 a.m.	KLR200	114



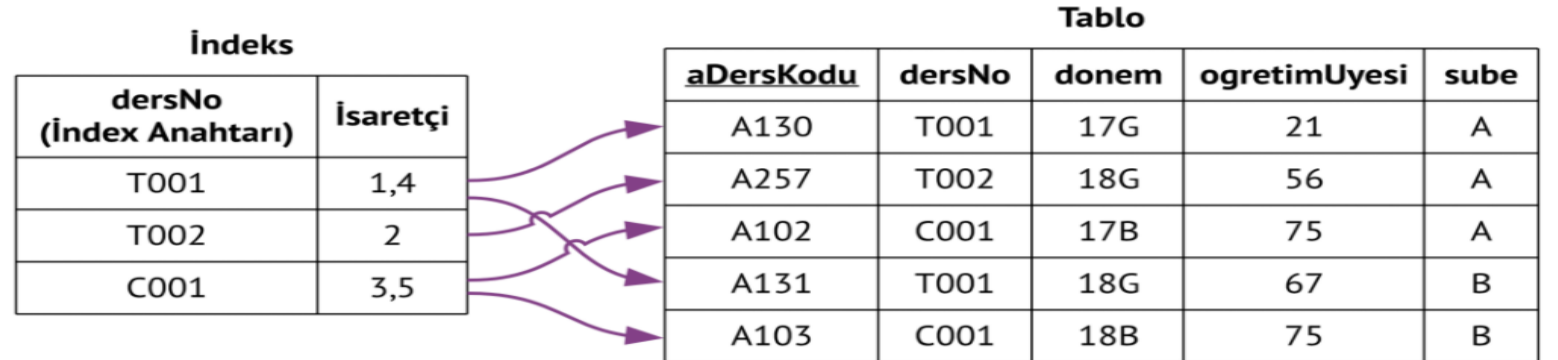
İndeks



- Bir kütüphanede bir kitap bulmak istediğinizi varsayalım. İstedığınızı bulana kadar her kitaba bakmak yerine Kitaplığın başlık, konu ve yazara göre indekslenmiş kataloğunu kullanırsınız. İndeks (manuel veya bilgisayar kütüphanesi kataloğu) sizi kitabın konumuna yönlendirerek almayı hızlı ve basit bir konu haline getirir!
- Bir kitapta bir konu bulmak istediğinizi varsayalım. Konuyla karşılaşana kadar her sayfayı okumak yerine kitabın dizinine gitmek, istediğiniz konuyu aramak ve sizi yönlendiren referansları okumak çok daha kolaydır!
- **İndeks**, Arama, listeleme işlemlerini hızlandırmak için kullanılan nesnelerdir.

İndeks

- Aranan bilginin konumunun, anahtar değerinden yararlanılarak bulunabilmesini sağlar.
- Bir indeks, diğer indeksleri etkilemeksizin yok edilebilir ya da oluşturulabilir.
- Genelde aramaların sık yapıldığı özellikler indeks yapılıdır.
- İndeks alanlarla yapılan aramalar çok hızlı olurken kayıt ekleme ve silme (INSERT, DELETE) işlemleri nispeten yavaş olur.
- İndeks alanlar sıralı olarak listelenir.
- Birincil anahtar (primary key) alanlar otomatik olarak eşsiz indeks (unique index) olurlar.
- Bir tabloda çok sayıda indeks tanımlanabilir. Her indeks yalnızca bir tabloya ait olabilir.
- Birden fazla alan beraber indeks olarak tanımlanabilir.



Veri Sözlüğü ve Sistem Kataloğu (The Data Dictionary and the System Catalog)

- **Veri Sözlüğü:** Üst verilerin (metadata) saklandığı veritabanıdır.
- **Sistem Kataloğu:** Ayrıntılı bir sistem veri sözlüğü
 - Veri sözlüğü gibi, sistem kataloğu da meta verileri içerir.
 - Sistem kataloğu, tablo adları, tablonun oluşturucusu ve oluşturulma tarihi, her tablodaki sütun sayısı, her bir sütuna karşılık gelen veri türü, indeks dosya adları, indeks oluşturucular, yetkili kullanıcılar ve erişim ayrıcalıkları dahil olmak üzere veritabanındaki tüm nesneleri açıklayan ayrıntılı bir sistem veri sözlüğü olarak tanımlanabilir.
- Veritabanı Yönetim Sistemi tarafından yönetilir.
- Oluşturulan tüm veritabanlarına ait üst veriler (tablolar, tablo alanları, alan tipleri, değer aralıkları, anahtarlar, indeksler, bağıntılar, kısıtlar vb.) burada saklanır.
- Kullanıcı tarafından sorgulanabilir.
- **Sistem kataloğu gerekli tüm veri sözlüğü bilgilerini içerdiğinden, sistem kataloğu ve veri sözlüğü terimleri genellikle birbirinin yerine kullanılır.**

TABLE 3.6

A SAMPLE DATA DICTIONARY

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
CUSTOMER	CUS_CODE	Customer account code	CHAR(5)	99999	10000-99999	Y	PK	
	CUS_LNAME	Customer last name	VARCHAR(20)	Xxxxxxxx		Y		
	CUS_FNAME	Customer first name	VARCHAR(20)	Xxxxxxxx		Y		
	CUS_INITIAL	Customer initial	CHAR(1)	X				
	CUS_RENEW_DATE	Customer insurance renewal date	DATE	dd-mmm-yyyy				
	AGENT_CODE	Agent code	CHAR(3)	999			FK	AGENT
AGENT	AGENT_CODE	Agent code	CHAR(3)	999		Y	PK	
	AGENT_AREACODE	Agent area code	CHAR(3)	999		Y		
	AGENT_PHONE	Agent telephone number	CHAR(8)	999-9999		Y		
	AGENT_LNAME	Agent last name	VARCHAR(20)	Xxxxxxxx		Y		
	AGENT_YTD_SLS	Agent year-to-date sales	NUMBER(9,2)	9,999,999.99				

FK	= Foreign key
PK	= Primary key
CHAR	= Fixed character length data (1 – 255 characters)
VARCHAR	= Variable character length data (1 – 2,000 characters)
NUMBER	= Numeric data. NUMBER (9,2) is used to specify numbers with up to nine digits, including two digits to the right of the decimal place. Some RDBMS permit the use of a MONEY or CURRENCY data type.

Veri Sözlüğü ve Sistem Kataloğu
(The Data Dictionary and the System Catalog)

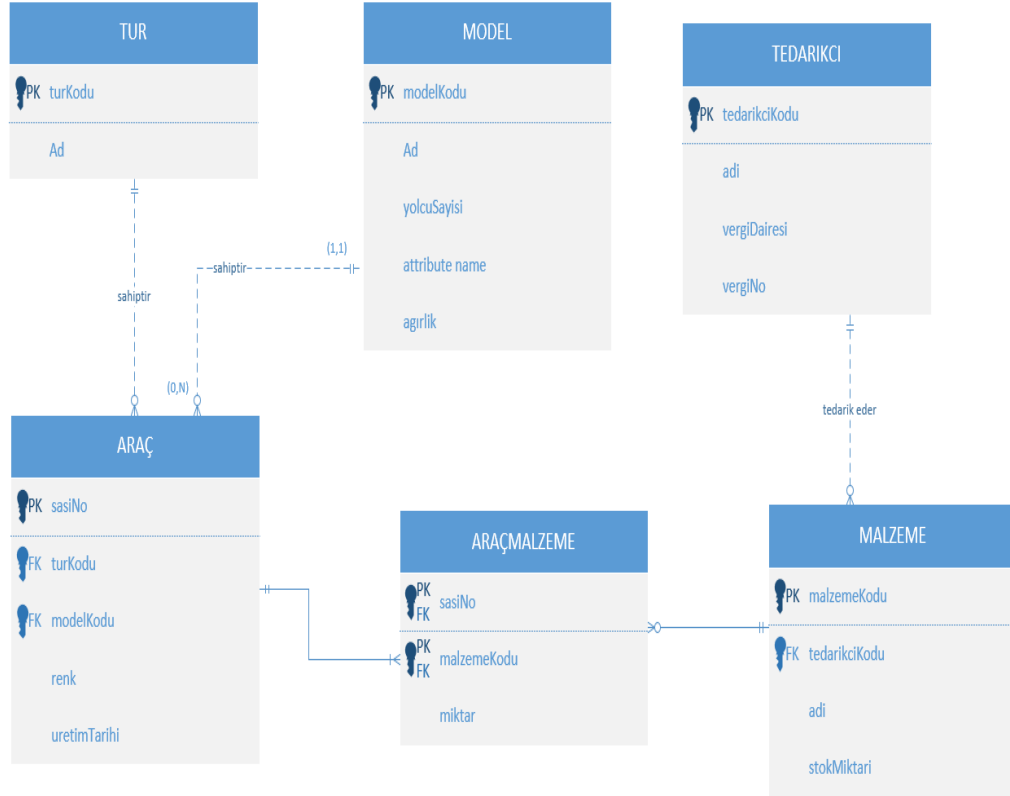
Örnek Uygulama

Araç Üreticisi

Senaryo

Bir araç üreticisi, farklı modellerde araçlar üretmektedir. Ürettiği araçlar, araçlarda kullanılan malzemeler, tedarikçiler vb. bilgilerini saklamak ve yönetmek istediği bir yazılım sistemi talep etmektedir.

- **İş Kuralları**
- Üretilen her bir aracın şasi numarası, renk, tür (otomobil, kamyon, minibüs vb.), model ve üretim tarihi bilgileri mevcuttur. Araçlar birbirinden şasi numarası ile ayırt edilir.
- Araç türlerinin kodu ve adı bilgileri mevcuttur.
- Araç modellerinin kodu, adı, ağırlık ve yolcu sayısı bilgileri mevcuttur.
- Malzemelerin kodu, adı, stok miktarı bilgileri mevcuttur.
- Malzemeler tedarikçi şirketlerden satın alınmaktadır.
- Tedarikçi şirketlerin kodu, adı, vergi dairesi, vergi numarası bilgileri mevcuttur.
- Bir tedarikçi hiçbir malzeme tedarik etmeyebileceği gibi bir veya daha fazla malzeme de tedarik edebilir. Bir malzeme yalnızca bir tedarikçi tarafından temin edilir.
- Bir araç çok sayıda malzemedен oluşur. En az bir adet malzemedен oluşmak zorundadır. Bir malzeme birden fazla araçta kullanılabilir. Ancak henüz hiçbir araçta kullanılmamış da olabilir.
- Araçlar farklı miktarda ve farklı türlerde malzemelerden üretilir. Bu bilgiler de kaydedilmelidir.
- Bir araç yalnızca bir türe sahip olur. Bir türe ait hiçbir araç olmayabileceği gibi çok sayıda araç da olabilir.
- Bir araç yalnızca bir modele sahip olur. Bir modele ait hiçbir araç olmayabileceği gibi çok sayıda araç da olabilir.



İlişkisel Şema (Metinsel Gösterim)

- Arac(sasiNo:char, renk: varchar, modelKodu: char, turKodu: char, üretimTarihi: date)
- AracMalzeme(sasiNo: char, malzemeKodu: char, miktar: int)
- Malzeme(malzemeKodu: char, adi: varchar, stokMiktari: int, tedarikciKodu: char)
- Tur(turKodu: char, ad: varchar)
- Model(modelKodu: char, ad: varchar, ağırlık: real, yolcuSayisi: int)
- Tedarikci(tedarikciKodu: char, ad: varchar, vergiDairesi: varchar, vergiNo: char)

Bölüm Özeti

- ilişkisel veri tabanının temel bloklarıdır. Varlık kümesi olarak da bilinen ilişkili varlıkların bir gruplaması içerisinde tutulur. Kavramsal olarak söz edilirse ilişkisel bir tablo kesişen oluşur. Her satır tek bir varlığı temsil eder ve her sütun varlığın karakteristiklerini (niteliklerini) temsil eder.
- ilişkisel tablo kullanımının merkezidir ve fonksiyonel bağımlılıkları tanımlar; böylece diğer nitelikler anahtara bağlıdır ve bu nedenle eğer anahtar değer biliniyorsa bulunabilirler. Anahtar çeşitleri:
.....
.....
- Her tablo satırında kendi dışındaki nitelikleri benzersiz şekilde tanımlayan bir nitelik veya nitelik grubu olabilen bulunmalıdır. Varlık bütünlüğünün sağlanması için Unique olmalı ve null değerlere müsaade edilmemelidir.
- Tablolar bağımsız olsalar da, ortak niteliklerle bağlanabilirler. Böylece, bir tablonun birincil anahtarı bağlı olduğu başka bir tabloda olarak görünebilir. Referans bütünlüğü, yabancı anahtarın ilgili tablodaki birincil anahtarla eşleşen değerler içermesi veya boş değerler içermesi gerektiğini belirtir.
- İlişkisel bir veritabanı, veri işleme işinin çoğunu perde arkasında gerçekleştirir. Örneğin, bir veritabanı oluşturduğunuzda, RDBMS otomatik olarak veritabanınız için bir veri sözlüğü barındıran bir yapı oluşturur. Veritabanında her yeni tablo oluşturduğunuzda, RDBMS veri sözlüğünü günceller ve böylece veritabanının dokümantasyonunu sağlar.
- İyi tasarım, uygun varlıkları, niteliklerini ve ardından varlıklar arasındaki ilişkileri tanımlayarak başlar. Bu ilişkiler (1: 1, 1: M ve M: N) ERD'ler kullanılarak temsil edilebilir. ERD'lerin kullanımı, basit mantıksal tasarım oluşturmanıza ve değerlendirmenize olanak tanır. 1: M ilişkisi en kolay şekilde iyi bir tasarıma dahil edilir; "1" in birincil anahtarının "çok" tablosunda yer aldığından emin olun.

Referanslar

- Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management, Cengage Learning.
- <https://github.com/celalceken/DatabaseManagementSystems>
- Microsoft Visio Professional
- <http://sr.photos3.fotosearch.com/bthumb/CSP/CSP992/k13040845.jpg>
- http://pngimg.com/uploads/notebook/notebook_PNG19198.png