

İLIŞKİSEL VERİTABANI MODELI

Verilerin Mantıksal Görünümü



İlişkisel model

- Verileri fiziksel olarak değil mantıksal olarak görüntüleme

Tablo

- Yapısal ve veri bağımsızlığı vardır.
- Çünkü ilgili kayıtların bağımsız tablolarda depolandığını düşünebilirsiniz

Kavramsal olarak bir dosyaya benzemektedir.

İlişkisel veritabanı modelinin anlaşılması **hiyerarşik** ve **ağ modellerinden** daha kolaydır.

Tablolar ve Özellikleri

Tablo: satır ve sütunlardan oluşan iki boyutlu yapıdır.

- Mantıksal ilişkinin kalıcı gösterimidir.

İlgili varlıklar grubunu içerir (varlık kümesi)

- Örneğin, bir STUDENT tablosu, her biri bir öğrenciyi temsil eden varlık oluşumları koleksiyonu içerir.

İlişki Adı,
Varlık
Kümesi



STUDENT

Öznitelikler(Attributes), Sütun(Column)

Tuples,
Rows,
Records

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1985	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1991	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1979	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1968	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1996	15	Fr	2.92	No	ACCT	2315	230

STU_NUM = Student number
STU_LNAME = Student last name
STU_FNAME = Student first name
STU_INIT = Student middle initial
STU_DOB = Student date of birth
STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_GPA = Grade point average
STU_TRANSFER = Student transferred from another institution
DEPT_CODE = Department code
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

- Sütunun izin verilen değerler aralığı etki alanı olarak bilinir.
- Örneğin, STU_GPA değerleri 0-4 (dahil) aralığıyla sınırlıysa, etki alanı [0,4] 'dir.
- Satır ve sütunların sırası kullanıcı için önemsizdir (önemsiz).
- Her tablonun bir birincil anahtarı olmalıdır.
-

Sütundaki tüm değerler özniteliğin özellikleriyle eşleşir:

a.Sayısal: STU_HRS ve STU_GPA

b.Karakter: STU_CLASS ve STU_PHONE

c.Tarih: STU_DOB

d. Mantıksal. STU_TRANSFER

Mantıksal veriler yalnızca doğru veya yanlış (evet veya hayır) değerlere sahip olabilir.

Anahtarlar (Keys)

Tablodaki her satır benzersiz olarak tanımlanabilir olmalıdır.

Anahtar, diğer öznitelikleri belirleyen bir veya daha fazla özniteliktir.

Anahtarın rolü kararlılığa dayanır.

A özniteliğinin değerini biliyorsanız, B özniteliğinin değeri saptayabilirsiniz.

İşlevsel bağımlılık (Functional dependence)

- ❑ Anahtarın rolü kararlılık olarak bilinen bir kavrama dayanmaktadır.
- ❑ Bir özneliliğin değeri diğerini tam olarak belirlerse, aralarında işlevsel bağımlılık vardır.

Örneğin, ÖĞRENCİ tablosundaki **STU_NUM** bilinmesi, öğrencinin *soyadını, not ortalamasını, telefon numarasını* vb. **arayabileceğiniz (belirleyebileceğiniz) anlamına gelir.**

- ❑ "A, B'yi belirler" ifadesinin kısa gösterimi : " $A \rightarrow B$.

İşlevsel bağımlılık (Functional dependence)

❑ **A**; *B, C ve D*'yi belirlerse, $A \rightarrow B, C, D$.

❑ Aslında **ÖĞRENCİ** tablosundaki **STU_NUM**, öğrencinin tüm öznitelik değerlerini belirler.

Örneğin, şunları yazabilirsiniz:

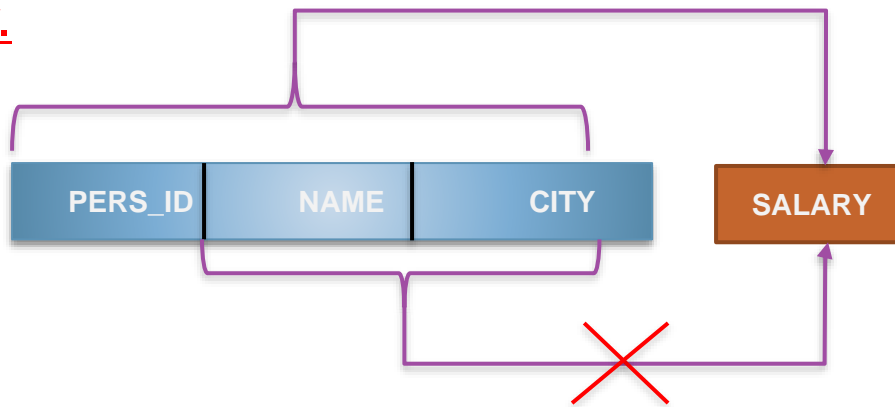
STU_NUM \rightarrow **STU_LNAME, STU_FNAME, STU_INIT, STU_DOB,**
STU_TRANSFER

Ancak, **STU_NUM, STU_LNAME** tarafından belirlenmez, çünkü birkaç öğrencinin soyadı Smith olması oldukça mümkündür.

İşlevsel bağımlılık (Functional dependence)

❖ Tam işlevsel bağımlılık

Özniteliğin işlevsel olarak bileşik anahtara bağımlı olduğu, ancak anahtarın herhangi bir alt kümesine bağlı olmadığı koşul ile oluşur.



Anahtar Türleri

❖ Anahtar özniteliği

- Anahtarın parçası olan herhangi bir öznitelik.
- Örneğin, **STU_NUMS** → **STU_GPA**

❖ Bileşik anahtar (Composite key)

- Birden fazla anahtardan oluşan bir anahtar.
- Örneğin;
(**STU_LNAME**, **STU_FNAME**, **STU_INIT**, **STU_PHONE**) → **STU_HRS**

Anahtar Türleri

❖ Süper anahtar

- Her satırı benzersiz olarak tanımlayan herhangi bir anahtar
- Başka bir deyişle, satırdaki her özniteliği işlevsel olarak belirler

Örneğin, STU_NUM ek öznitelikler olsun veya olmasın tanımlama işlemini yerine getirir.

Anahtarlar

❖ Aday anahtarı

- Aday anahtarı minimal bir süper anahtardır.

STU_NUM, STU_LNAME bir süper anahtardır.

Ancak bu bir aday anahtarı değildir, çünkü STU_LNAME kaldırılabilir ve STU_NUM anahtar yine de bir süper anahtar olacaktır.

STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE

aday anahtarı da olabilir.

Çünkü iki öğrencinin aynı soyadını, adını, baş harfini ve telefon numarasını paylaşma olasılığı azalır.

❑ Birincil anahtar:

Tablo satırlarının benzersiz olarak tanımlanacağı birincil yol olarak seçilen aday anahtardır.

Örneğin, STU_PHONE

Varlık bütünlüğü

❖ **Varlık bütünlüğü**, tablodaki her satırın (varlık örneği) kendi benzersiz kimliğine sahip olduğu koşuldur.

□ Varlık bütünlüğünü sağlamak için birincil anahtarın iki gereksinimi vardır:

- Birincil anahtardaki tüm değerler benzersiz olmalıdır.
- Birincil anahtardaki hiçbir anahtar özniteliği null içeremez.

Anahtarlar

❖ Nulls:

- Veri girişi yok
- Birincil anahtarda izin verilmez
- Diğer özniteliklerde kaçınılmalıdır
- BİR DEĞERİ SAYMA, ORTALAMA ve TOPLA gibi işlevler kullanıldığında sorun yaratabilir
- İlişkisel tablolar bağlandığında mantıksal sorunlar oluşturabilir.
- Temsil edebilir
 - Bilinmeyen bir öznitelik değeri
 - Bilinen, ancak eksik bir öznitelik değeri
 - "Uygulanamaz" koşulu

Anahtarlar

❖ Kontrol edilmiş fazlalık/artıklık: Controlled redundancy:

- ❑ İlişkisel veritabanının çalışmasını sağlar.
 - Veritabanındaki tablolar ortak öznitelikleri paylaşır.
 - Tabloların birbirine bağlanmasını sağlar.
- ❑ İlişkinin çalışması için gerektiğinde birden çok değer tekrarı olabilir.
- ❑ Artıklık yalnızca öznitelik değerlerinin çoğaltması durumunda ortaya çıkabilir.

Anahtarlar

FIGURE 3.2 AN EXAMPLE OF A SIMPLE RELATIONAL DATABASE

Table name: PRODUCT

Primary key: PROD_CODE

Foreign key: VEND_CODE

Database name: Ch03_SaleCo

PROD_CODE	PROD_DESCRIPTOR	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Household chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

link

Table name: VENDOR

Primary key: VEND_CODE

Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

İlişkisel şema, veritabanı tablolarının metinsel gösterimidir.

Burada her tablo kendi adına göre listelenir ve ardından özniteliklerinin listesi parantez içine alınır.

Birincil anahtar özniteliklerinin altı çizili olarak gösterilir.

Örneğin, Yukarıdaki şeklin ilişki şeması şöyle gösterilir:

PRODUCT (PROD_CODE, PROD_DESCRIPTOR, PROD_PRICE,
PROD_ON_HAND, VEND_CODE)

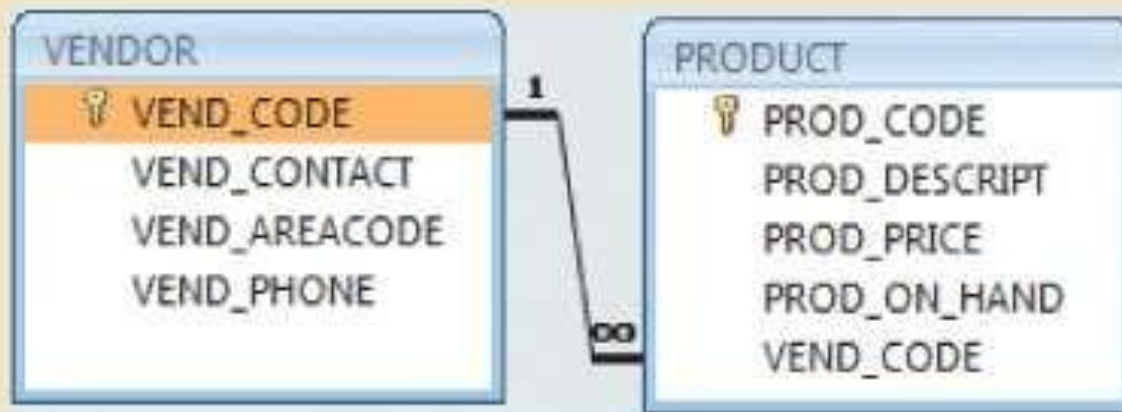
VENDOR (VEND_CODE, VEND_CONTACT, VEND_AREACODE,
VEND_PHONE)

Anahtarlar

Aşağıda ÜRÜN (PRODUCT) ve SATICI(VENDOR) tabloları arasındaki bağlantı, Aağıdaki şekilde gösterilen ilişkiyel diyagramla da temsil edilebilir.

**FIGURE
3.3**

The relational diagram for
the Ch03_SaleCo database



Anahtarlar

FIGURE 3.2 AN EXAMPLE OF A SIMPLE RELATIONAL DATABASE

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

Database name: C

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

link

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

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Yabancı anahtar (FK)

- Değerleri ilgili tablodaki birincil anahtar değerleriyle eşleşen özniteliktir.

İkincil anahtar (Diğer anahtar)

- Anahtar kesinlikle veri alma amacıyla kullanılır.
- Örneğin, müşterinin soyadı ve telefon numarası ikincil anahtardır.

Bilgi tutarlılığı (Referential integrity)

- FK, başka bir ilişkide varolan geçerli bir diziye (satır) başvuran bir değer içerir.
- Örneğin, ÜRÜN tablosundaki bir satırla başvurulanan (VEN_CODE) her satıcı geçerli bir satıcıdır.

Bütünlük Kuralları (Integrity Rules)

- ❑ İlişkisel veritabanı bütünlüğü kuralları, veritabanının iyi tasarlanmasını sağlar. Bu nedenle çok önemlidir.
- ❑ İlişkisel veritabanı yönetim sistemleri (RDBMSs), bütünlük kurallarını otomatik olarak zorlar.
- ❑ Ancak, uygulama tasarımınızın varlık ve bilgi tutarlılığı kurallarına uyması gerekir.

TABLE 3.4

INTEGRITY RULES

ENTITY INTEGRITY	DESCRIPTION
Requirement	All primary key entries are unique, and no part of a primary key may be null.
Purpose	Each row will have a unique identity, and foreign key values can properly reference primary key values.
Example	No invoice can have a duplicate number, nor can it be null; in short, all invoices are uniquely identified by their invoice number.
REFERENTIAL INTEGRITY	DESCRIPTION
Requirement	A foreign key may have either a null entry, as long as it is not a part of its table's primary key, or an entry that matches the primary key value in a table to which it is related (every non-null foreign key value <i>must</i> reference an <i>existing</i> primary key value).
Purpose	It is possible for an attribute <i>not</i> to have a corresponding value, but it will be impossible to have an invalid entry; the enforcement of the referential integrity rule makes it impossible to delete a row in one table whose primary key has mandatory matching foreign key values in another table.
Example	A customer might not yet have an assigned sales representative (number), but it will be impossible to have an invalid sales representative (number).

Tablo 3.4'te özetlenen bütünlük kuralları Şekil 3.3'te gösterilmiştir.

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	Ramas	Alfred	A	05-Apr-2018	502
10011	Dunne	Leona	K	16-Jun-2018	501
10012	Smith	Kathy	W	29-Jan-2019	502
10013	Olowski	Paul	F	14-Oct-2018	
10014	Orlando	Myron		28-Dec-2018	501
10015	O'Brian	Amy	B	22-Sep-2018	503
10016	Brown	James	G	25-Mar-2019	502
10017	Williams	George		17-Jul-2018	503
10018	Farriss	Anne	G	03-Dec-2018	501
10019	Smith	Olette	K	14-Mar-2019	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	139967.35
503	615	123-5589	Olson	127093.45

- ❑ **Varlık bütünlüğü**
- ❑ MÜŞTERİ tablosunun birincil anahtarı CUS_CODE.
- ❑ MÜŞTERİ birincil anahtar sütununda null girişi olmaz ve tüm girişler benzersizdir.

❑ **Bilgi tutarlılığı**

- ❑ MÜŞTERİ tablosu, MÜŞTERİ tablosundaki girişleri AGENT tablosuna bağlayan yabancı bir anahtar AGENT_CODE içerir.
- ❑ (Birincil anahtar) numarası 10013 ile tanımlanan CUS_CODE satırı, Paul F. Olowski'nin kendisine atanmış bir satış temsilcisi olmadığından, AGENT_CODE yabancı anahtarında boş bir giriş içerir.
- ❑ MÜŞTERİ tablosundaki kalan AGENT_CODE girişlerin tümü AGENT tablosundaki AGENT_CODE girişleri ile eşleşir.

Bütünlük Kuralları bayrakları

- Null önlemek için, bazı tasarımcılar bazı değerin yokluğunu belirtmek için **bayraklar** olarak bilinen özel kodlar kullanır.
- Örnek olarak Şekil 3.3'ü kullanarak, -99 kodu, müşteri Paul Olowski'nin kendisine atanmış bir aracısı olmadığını belirtmek için MÜŞTERİ tablosunun dördüncü satırındaki AGENT_CODE girdisi olarak kullanılabilir.

■ TABLE 3.5

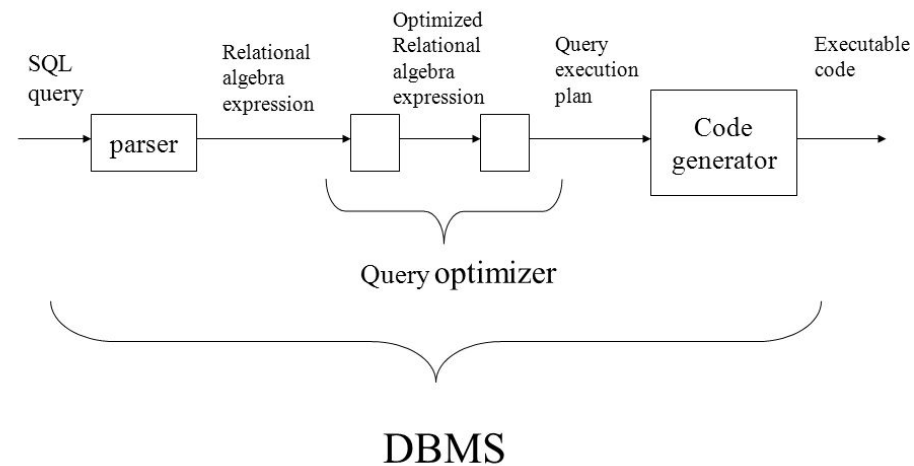
A DUMMY VARIABLE VALUE USED AS A FLAG

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
-99	000	000-0000	None	\$0.00

İlişkisel Cebir

- **İlişkisel cebir**, ilişkisel veritabanlarında sorgu işlemlerinin matematiksel modellemesi ve çözümü için kullanılır.

Relational Algebra in a DBMS



Biçimsel Tanımlar ve Terminoloji

- Belirli bir terim ilişkisini kullanmanın önemli bir yönü, ilişki ve ilişki değişkeni (relvar) arasındaki ayrımı kabul ediyor olmasıdır.
- **İlişki**, tablolarımızda gördüğümüz verilerdir.
- **Relvar**, ilişki tutan bir değişkendir.
- **Bir relvarın** iki bölümü vardır:
 - **Başlık** ve **gövde**.

Biçimsel Tanımlar ve Terminoloji

- **Relvar başlığı**, özniteliklerin adlarını içerir.
- **Relvar gövdesi**, ilişkiyi içerir.
- **Yapıya** düzgün bir relvar denir.
- **Yapıdaki** veriler bir ilişki olacaktır.
- **Notasyon:** Relvar R gösterilir. İlişki r gösterilir.

İlişkisel Küme Operatörleri

- İlişkisel işleçler kapalılık özelliğine sahiptir.
- Bazı operatörler temeldir, diğerleri ise temel operatörler kullanılarak türetilebilir.

TEMEL OPERATÖRLER

- SELECT (or RESTRICT)
- PROJECT
- UNION
- INTERSECT,
- DIFFERENCE
- PRODUCT
- JOIN
- DIVIDE

Select (Restrict)

FIGURE 3.4 SELECT

Original table				New table		
P_CODE	P_DESCRIPTION	PRICE		P_CODE	P_DESCRIPTION	PRICE
123456	Flashlight	5.26	SELECT ALL yields	123456	Flashlight	5.26
123457	Lamp	25.15		123457	Lamp	25.15
123458	Box Fan	10.99		123458	Box Fan	10.99
213345	9v battery	1.92		213345	9v battery	1.92
254467	100W bulb	1.47		254467	100W bulb	1.47
311452	Powerdrill	34.99		311452	Powerdrill	34.99

SELECT only PRICE less than \$2.00 yields		
P_CODE	P_DESCRIPTION	PRICE
213345	9v battery	1.92
254467	100W bulb	1.47

SELECT only P_CODE = 311452 yields		
P_CODE	P_DESCRIPTION	PRICE
311452	Powerdrill	34.99

- SELECT, belirli bir koşulu karşılayan bir tablodaki tüm satırlar için tüm değerleri verir.
- Tablodaki tüm satırları listelemek için de kullanılabilir.
- Tablonun yatay bir alt kümesini verir.

Resmi olarak, SELECT küçük Yunanca sigma harfi (σ) ile gösterilir.

Örneğin, P_CODE özniteliğinde "311452" değerine sahip ÜRÜN tablosundaki tüm satırları SEÇMEK için aşağıdakileri yazarsınız:

$\sigma_{p_code = 311452}$ (PRODUCT)

Not:

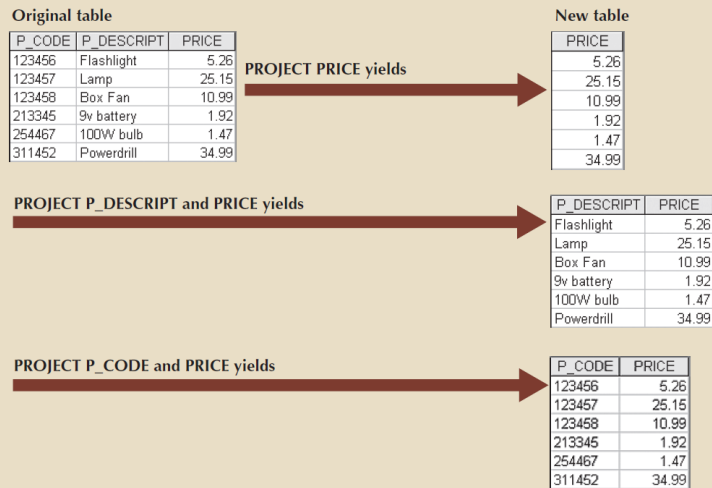
Select işleci karşılaştırma operatörleri ile de yazılır (=, <, >, ≠, ≥, ≤).

Ayrıca, \vee (veya) \wedge (ve) işleçler için kullanılabilir.

" σ " _ "p_code" = 311452 \vee fiyatı > 6 (ÜRÜN)

Project

FIGURE 3.5 PROJECT



- Project seçili öz nitelikler için tüm değerleri verir.
- Başka bir deyişle, PROJECT bir tablonun dikey bir alt kümesini verir.

Örneğin, CUS_FNAME ve CUS_LNAME özniteliklerini MÜŞTERİ tablosuna aşağıdakileri yazarız:

$\pi_{\text{cus_fname, cus_lname}}(\text{customer})$

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	05-Apr-2018	502
10011	Dunne	Leona	K	16-Jun-2018	501
10012	Smith	Kathy	W	29-Jan-2019	502
10013	Olowski	Paul	F	14-Oct-2018	
10014	Orlando	Myron		28-Dec-2018	501
10015	O'Brian	Amy	B	22-Sep-2018	503
10016	Brown	James	G	25-Mar-2019	502
10017	Williams	George		17-Jul-2018	503
10018	Farriss	Anne	G	03-Dec-2018	501
10019	Smith	Olette	K	14-Mar-2019	503

İlişkisel operatörler, ilişkileri girdi olarak kabul etmek ve ilişkileri çıktı olarak üretmek, birleştirme işleçlerini kullanmak mümkündür. Örneğin, müşterinin adını ve soyadını bulmak için önceki iki operatörü customer_code 10010 ile birleştirebilirsiniz:

$\pi_{\text{cus_fname, cus_lname}}(\sigma_{\text{cus_code}=10010}(\text{customer}))$

Union

- Yinelenen satırlar hariç, iki tablodaki tüm satırları birleştirir.
- Tablolar aynı sayıda sütuna sahip olmalıdır ve **karşılık gelen sütunları aynı veya uyumlu etki alanlarını paylaşmalıdır:** birleşim uyumlu.

FIGURE 3.6 UNION

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	160.00
345679	Dishwasher	500.00
123458	Box Fan	10.99

yields



P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

Union

- **UNION**, \cup sembolüyle gösterilir.
- TEDARİKÇİ (SUPPLIER) ve SATICI (VENDOR) ilişkileri birliğe uyumluysa, aralarındaki bir UNION aşağıdaki gibi gösterilir:
 - **supplier \cup vendor**
- Örneğin, TEDARİKÇİ ve SATICI tablolarının birleşim uyumlu olmadığını varsayalım.
- Tüm SATICI ve TEDARİKÇİ adlarının bir listesini oluşturmak isterseniz, her tablodaki adları PROJECT ve ardından onlarla bir UNION gerçekleştirebilirsiniz.
- $\pi_{\text{supplier_name}}(\text{supplier}) \cup \pi_{\text{vendor_name}}(\text{vendor})$

Intersect

- INTERSECT (KESİŞİM) yalnızca her iki tabloda da görünen satırları verir.
- UNION'da olduğu gibi, tabloların geçerli sonuçlar verebilmesi için birleşim uyumlu olması gerekir.

FIGURE 3.7 INTERSECT

STU_FNAME	STU_LNAME	INTERSECT	EMP_FNAME	EMP_LNAME	yields	STU_FNAME	STU_LNAME
George	Jones		Franklin	Lopez		Franklin	Johnson
Jane	Smith		William	Turner			
Peter	Robinson		Franklin	Johnson			
Franklin	Johnson		Susan	Rogers			
Martin	Lopez						

INTERSECT, \cap sembolüyle gösterilir.

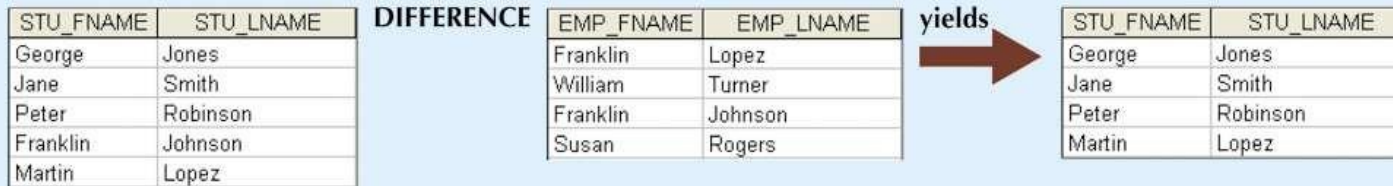
$\pi_{\text{supplier_name}} (\text{supplier}) \cap \pi_{\text{vendor_name}} (\text{vendor})$

Difference (FARKLILIK)

- Diğer tabloda bulunmayan bir tablodaki tüm satırları verir.
- Bir tabloyu diğerinden çıkarır.
- Tabloların sırası önemlidir.
- Tablolar birleşim uyumludur.

FIGURE
3.8

DIFFERENCE



SOURCE: Course Technology/Cengage Learning

DIFFERENCE eksi simgesiyle gösterilir –.

supplier – vendor

supplier ve **vendor** tablolarının birbirleri ile uyumlu olmadığını varsayarsak, bir DIFFERENCE işleci kullanabilirsiniz.

$\pi_{\text{supplier_name}}(\text{supplier}) - \pi_{\text{vendor_name}}(\text{vendor})$

Product

- PRODUCT, Kartezyen olarak da bilinen iki tablodan olası tüm satır çiftlerini verir.
- Bu nedenle, bir tabloda 6 satır ve diğer tabloda 3 satır varsa, ÜRÜN 6 'dan oluşan bir liste verir. $3 = 18$ satır.

FIGURE
3.9

PRODUCT

P_CODE	P_DESCRIPTION	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PRODUCT

STORE	aisle	Shelf
23	W	5
24	K	9
25	Z	6

yields

P_CODE	P_DESCRIPTION	PRICE	STORE	aisle	Shelf
123456	Flashlight	5.26	23	W	5
123456	Flashlight	5.26	24	K	9
123456	Flashlight	5.26	25	Z	6
123457	Lamp	25.15	23	W	5
123457	Lamp	25.15	24	K	9
123457	Lamp	25.15	25	Z	6
123458	Box Fan	10.99	23	W	5
123458	Box Fan	10.99	24	K	9
123458	Box Fan	10.99	25	Z	6
213345	9v battery	1.92	23	W	5
213345	9v battery	1.92	24	K	9
213345	9v battery	1.92	25	Z	6
311452	Powerdrill	34.99	23	W	5
311452	Powerdrill	34.99	24	K	9
311452	Powerdrill	34.99	25	Z	6
254467	100W bulb	1.47	23	W	5
254467	100W bulb	1.47	24	K	9
254467	100W bulb	1.47	25	Z	6

SOURCE: Course Technology/Cengage Learning

ÜRÜN çarpma simgesi × ile gösterilir.

MÜŞTERİ (COSTOMER) ve TEMSİLCİ (AGENT) ilişkilerinin PRODUCT işlemi aşağıdaki gibi yazılacaktır:

customer × agent

Join

- JOIN, bilgilerin iki veya daha fazla tablodan birleştirilmesine izin verir
- İlişkisel veritabanının arkasındaki gerçek güç, ortak özniteliklerle bağlantılı tabloların kullanılmasına izin verir.

FIGURE 3.10 TWO TABLES THAT WILL BE USED IN JOIN ILLUSTRATIONS

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Table name: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

Ortak öznitelikler

Natural Join (1)

- Yalnızca ortak özniteliklerinde ortak değerlere sahip satırlardan oluşan yeni bir tablo veren ilişkisel bir işlemidir.
- İlişkileri (tabloları) birleştirmek için herhangi bir koşul yazmaya gerek yoktur.
- Natural Join, üç aşamalı bir sürecin sonucudur:

Natural Join (2)

1. İlk olarak, Şekil 3.11'de gösterilen sonuçları vererek tabloların bir ÜRÜNÜ oluşturulur.

FIGURE 3.10 TWO TABLES THAT WILL BE USED IN JOIN ILLUSTRATIONS

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Table name: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445



FIGURE 3.11 NATURAL JOIN, STEP 1: PRODUCT

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1132445	Walker	32145	231	125	6152439887
1132445	Walker	32145	231	167	6153426778
1132445	Walker	32145	231	231	6152431124
1132445	Walker	32145	231	333	9041234445
1217782	Adares	32145	125	125	6152439887
1217782	Adares	32145	125	167	6153426778
1217782	Adares	32145	125	231	6152431124
1217782	Adares	32145	125	333	9041234445
1312243	Rakowski	34129	167	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1312243	Rakowski	34129	167	231	6152431124
1312243	Rakowski	34129	167	333	9041234445
1321242	Rodriguez	37134	125	125	6152439887
1321242	Rodriguez	37134	125	167	6153426778
1321242	Rodriguez	37134	125	231	6152431124
1321242	Rodriguez	37134	125	333	9041234445
1542311	Smithson	37134	421	125	6152439887
1542311	Smithson	37134	421	167	6153426778
1542311	Smithson	37134	421	231	6152431124
1542311	Smithson	37134	421	333	9041234445
1657399	Vanloo	32145	231	125	6152439887
1657399	Vanloo	32145	231	167	6153426778
1657399	Vanloo	32145	231	231	6152431124
1657399	Vanloo	32145	231	333	9041234445

Natural Join (3)

2. Adım 1'in çıktısı üzerinde yalnızca AGENT_CODE değerlerinin eşit olduğu satırları vermek için bir SELECT gerçekleştirilir.

Ortak sütunlar birleştirme sütunları olarak adlandırılır.

FIGURE 3.12 NATURAL JOIN, STEP 2: SELECT

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	125	6152439887
1321242	Rodriguez	37134	125	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1132445	Walker	32145	231	231	6152431124
1657399	Vanloo	32145	231	231	6152431124

Natural Join (4)

3. A PROJECT is performed on the results of Step 2 to yield a single copy of each attribute, thereby eliminating duplicate columns. Step 3 yields the output shown in Figure 3.13.

FIGURE 3.13 NATURAL JOIN, STEP 3: PROJECT

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124

Natural Join (4)

- Natural Join, normalde sadece resmi işlemlerde JOIN olarak adlandırılır.
- JOIN, \bowtie sembolüyle gösterilir.
- CUSTOMER ve AGENT ilişkilerinin JOIN işlemi aşağıdaki gibi yazılacaktır:
- **customer \bowtie agent**
- JOIN temel bir ilişkisel cebir operatörü değildir.
Diğer operatörlerden aşağıdaki gibi türetilebilir:

$$\pi_{\text{cus_code, cus_lname, cus_fname, cus_initial, cus_renew_date, agent_code, agent_areacode, agent_phone, agent_lname, agent_ytd_sls}}(\sigma_{\text{customer.agent_code = agent.agent_code}}(\text{customer} \times \text{agent}))$$

Diğer Join operatörleri (1)

❑ Equijoin

- Tabloları, belirtilen sütunları karşılaştıran eşitlik (=) koşuluna göre bağlar.
- Yinelenen sütunları ortadan kaldırmaz.

❑ Theta join

- Birleştirme koşulunda eşitsizlik karşılaştırma işleci (<, >, <=, >=) kullanarak tabloları bağlayan birleştirme işleci.

❑ Inner join

- Yalnızca tablolardaki eşleşen kayıtları döndürür
- Natural Join; equijoin ve theta join birleşimlerin kesişim birleşimleridir.

Diğer Join operatörleri (2)

- **Outer join**

- Eşleşen tüm kayıtları (inner join olarak) döndürür, ancak tablolardan birinden eşleşmeyen kayıtları döndürür.
- İlgili tablolardaki hangi değerlerin bilgi tutarlılığı sorunlarına neden olduğunu belirlemede yararlıdır.

Diğer Join operators (3)

- **Left outer join**
- Diğer tabloda eşleşen değeri olmayanlar da dahil olmak üzere, sol tablodaki tüm satırları (tüm değerler) veren birleştirme işlemi.

Table name: CUSTOMER				Table name: AGENT	
CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_CODE	AGENT_PHONE
1132445	Walker	32145	231	125	6152439887
1217782	Adares	32145	125	167	6153426778
1312243	Rakowski	34129	167	231	6152431124
1321242	Rodriguez	37134	125	333	9041234445
1542311	Smithson	37134	421		
1657399	Vanloo	32145	231		

LEFT JOIN


CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	125	6152439887
1321242	Rodriguez	37134	125	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1132445	Walker	32145	231	231	6152431124
1657399	Vanloo	32145	231	231	6152431124
1542311	Smithson	37134	421		

Diğer Join operators (3)

Right outer join

- Diğer tabloda eşleşen değeri olmayanlar da dahil olmak üzere, sağ tablodaki tüm satırları (tüm değerler) veren birleştirme işlemidir.

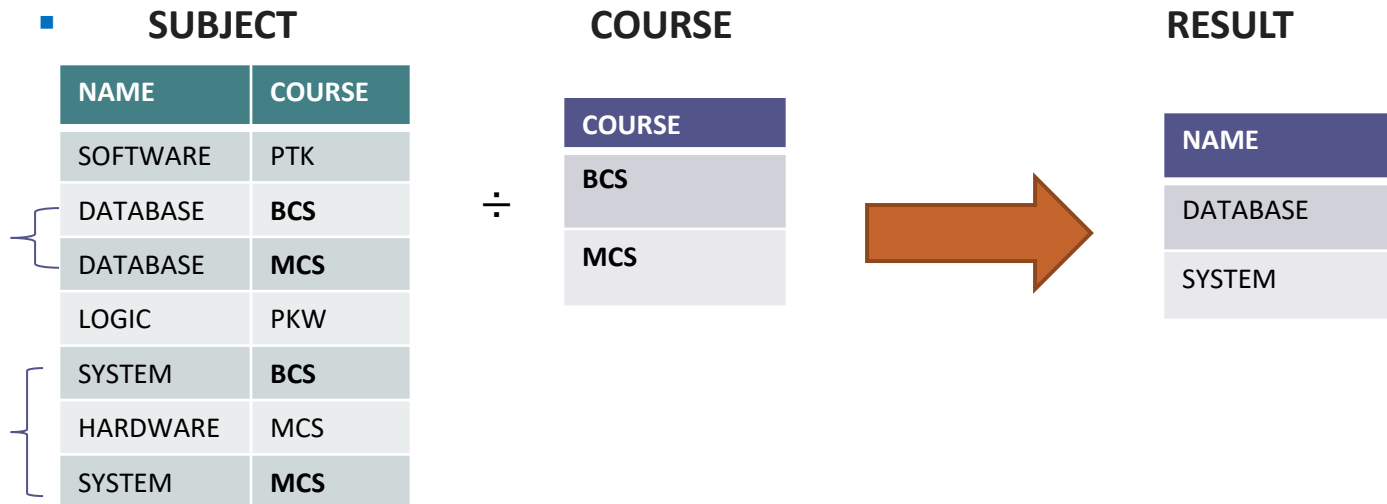
Table name: CUSTOMER				Table name: AGENT	
CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_CODE	AGENT_PHONE
1132445	Walker	32145	231	125	6152439887
1217782	Adares	32145	125	167	6153426778
1312243	Rakowski	34129	167	231	6152431124
1321242	Rodriguez	37134	125	333	9041234445
1542311	Smithson	37134	421		
1657399	Vanloo	32145	231		

 **RIGHT JOIN**

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	125	6152439887
1321242	Rodriguez	37134	125	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1132445	Walker	32145	231	231	6152431124
1657399	Vanloo	32145	231	231	6152431124
				333	9041234445

Divide

- Başka bir veri kümesindeki tüm veri değerleriyle ilişkili bir veri kümesiyle ilgili sorguları yanıtlayan işleç.



Note: DIVIDE işleci, \div bölme simgesiyle gösterilir.

R ve **S** olmak üzere iki ilişki göz önüne alındığında, bunların BÖLÜMÜ $r \div s$ olarak yazılacaktır.

Veri Sözlüğü ve Sistem Kataloğu

❖ Veri sözlüğü

- Kullanıcı/tasarımcı tarafından oluşturulan veritabanında bulunan tüm tabloların ayrıntılı muhasebesini sağlar.
- (En azından) tüm öznitelik adlarını ve sistemdeki her tablo için özellikleri barındırır.
- Meta veriler içerir: verilerle ilgili veriler.

Veri Sözlüğü ve Sistem Kataloğu

❖ Sistem kataloğu

- Meta veriler içerir
- Veritabanındaki tüm nesneleri açıklayan ayrıntılı sistem veri sözlüğü

Veri Sözlüğü ve Sistem Kataloğu (2)

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.

İlişkisellik

Veritabanı

❖ 1:M ilişki

- İlişkisel modelleme ideali
- Herhangi bir ilişkisel veritabanı tasarımında norm olmalıdır.

❖ 1:1 relationship

- Should be rare in any relational database design

● M:N ilişki

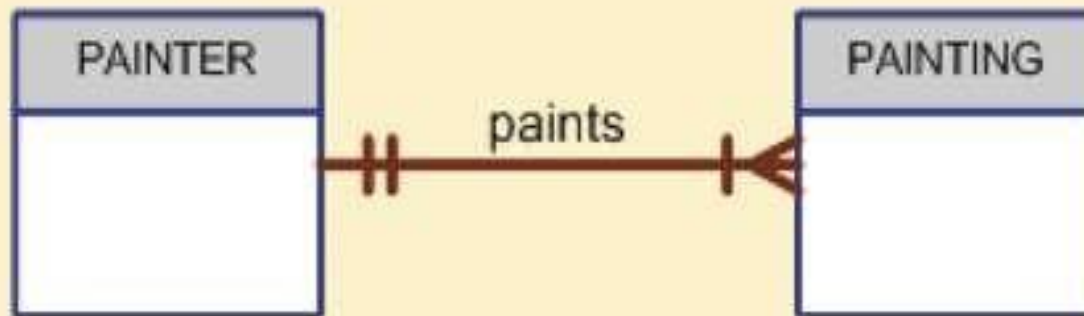
- İlişkisel modelde bu şekilde uygulanamaz.
- M:N ilişkileri 1:M ilişkilerine dönüştürülebilir.

1:M İlişki

- İlişkisel veritabanı normu
- Herhangi bir veritabanı ortamında bulunur

**FIGURE
3.17**

**The 1:M relationship between
PAINTER and PAINTING**



SOURCE: Course Technology/Cengage Learning

**FIGURE
3.18**

The implemented 1:M relationship between PAINTER and PAINTING

Table name: PAINTER

Primary key: PAINTER_NUM

Foreign key: none

Database name: Ch03_Museum

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Ittero	Julio	G

Table name: PAINTING

Primary key: PAINTING_NUM

Foreign key: PAINTER_NUM

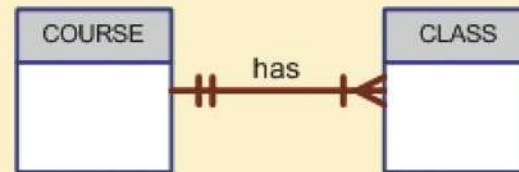
PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

PK of the "1"
side is put into
the "many"
side as a
column

SOURCE: Course Technology/Cengage Learning

FIGURE 3.19

The 1:M relationship between COURSE and CLASS



SOURCE: Course Technology/Cengage Learning

FIGURE 3.20

The implemented 1:M relationship between COURSE and CLASS

Table name: COURSE

Primary key: CRS_CODE

Foreign key: none

Database name: Ch03_TinyCollege

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

Primary key: CLASS_CODE

Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	vV 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162

SOURCE: Course Technology/Cengage Learning

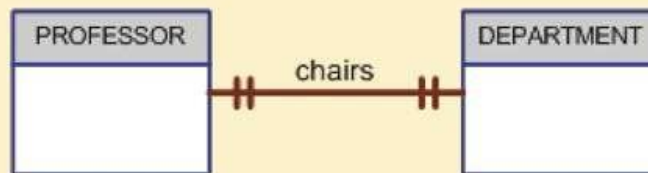
The composite key CRS_CODE and CLASS_SECTION is a candidate key as together they uniquely identify each row

1:1 İlişki

- Yalnızca bir varlıkla ilgili bir varlık ve bunun tersi de geçerlidir.
- Bazen varlık bileşenlerinin düzgün tanımlanmamış olduğu anlamına gelir.
- İki varlığın aslında aynı tabloya ait olduğunu gösterebilir.
- Bazı koşullar kesinlikle kullanımlarını gerektirir.

FIGURE
3.21

The 1:1 relationship between PROFESSOR and DEPARTMENT



SOURCE: Course Technology/Cengage Learning

FIGURE 3.22

The implemented 1:1 relationship between PROFESSOR and DEPARTMENT

Database name: Ch03_TinyCollege

Table name: PROFESSOR

Primary key: EMP_NUM

Foreign key: DEPT_CODE

EMP_NUM	DEPT_CODE	PROF_OFFICE	PROF_EXTENSION	PROF_HIGH_DEGREE
103	HIST	DRE 156	6783	Ph.D.
104	ENG	DRE 102	5561	MA
105	ACCT	KLR 229D	8665	Ph.D.
106	MKT/MGT	KLR 126	3899	Ph.D.
110	BIOL	AAK 160	3412	Ph.D.
114	ACCT	KLR 211	4436	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	4016	DBA
195	PSYCH	AAK 297	3550	Ph.D.
209	CIS	KLR 333	3421	Ph.D.
228	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	4883	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	6783	MA
425	ECON/FIN	KLR 284	2851	MBA
435	ART	BBG 185	2278	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

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	387	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 63	3126
ENG	English	A&SCI	160	DRE 102, Box 223	1004
HIST	History	A&SCI	103	DRE 156, Box 284	1867
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	2008



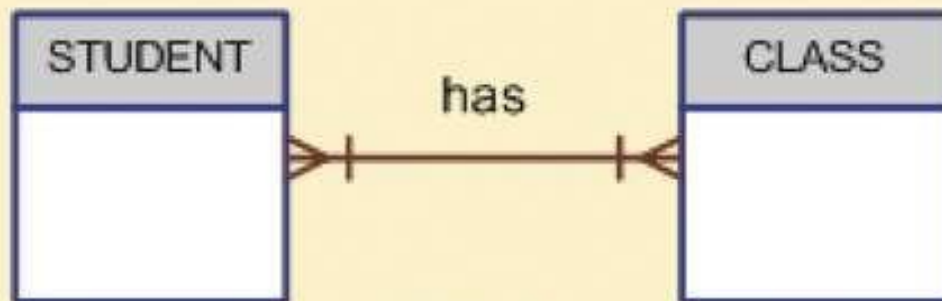
The 1:1 PROFESSOR chairs DEPARTMENT relationship is implemented through the placement of the EMP_NUM foreign key in the DEPARTMENT table.

M:N İlişki

- ❖ 1:M ilişkileri kümesi oluşturmak için parçalanma işlemi uygulanır.
- ❖ Bileşik varlık oluşturarak M:N ilişkisinin doğasında bulunan sorunlardan kaçınmak gerekir.
- ❖ Yabancı anahtarlar olarak bağlanacak tabloların birincil anahtarlarını içerir.

**FIGURE
3.23**

**The ERM's M:N relationship
between STUDENT and CLASS**



SOURCE: Course Technology/Cengage Learning

M:N İlişki (2)

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	MWTF 9:00-9:50 a.m.	KLR211	114
10018	324257	CIS-220	2	MWTF 9:00-9:50 a.m.	KLR211	114
10021	321452	QM-261	1	MWTF 8:00-8:50 a.m.	KLR200	114
10021	324257	QM-261	1	MWTF 8:00-8:50 a.m.	KLR200	114

SOURCE: Course Technology/Cengage Learning

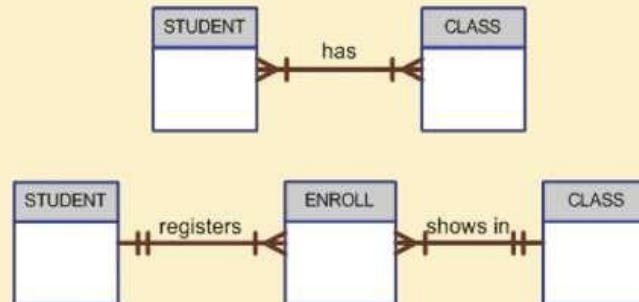
- **Fazlalıklar:**
- *STU_NUM* değerler *STUDENT* tablosunda birden çok kez oluşur.
- Gerçek dünyada, tekrarlanacak daha fazla öğrenci bilgisi olurdu (adres, telefon vb.)
- *CLASS* tablosunda da yedekli *CLASS_CODE*

M:N İlişki

- Bunun yerine, hem STUDENT hem de CLASS'ın PK'larını en az düzeyde içeren veya PK olarak yeni, tek öznitelikli bir anahtar kullanan bileşik bir varlık oluşturulması iyidir.
- Varlık köprüsü veya bağlama tablosu olarak ENROLL (KAYIT)
- Genellikle aşağıdakiler gibi diğer ilgili bilgileri içerir.

FIGURE
3.26

Changing the M:N relationship to two 1:M relationships



SOURCE: Course Technology/Cengage Learning

**FIGURE
3.25**

Converting the M:N relationship into two 1:M relationships

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

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Database name: Ch03_CollegeTry2

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

ENROLL contains multiple occurrences of the FK values, but those controlled redundancies won't cause anomalies as long as referential integrity is enforced

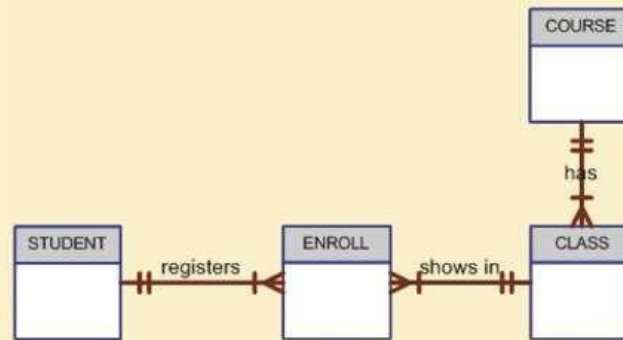
Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

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	MWTF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWTF 8:00-8:50 a.m.	KLR200	114

SOURCE: Course Technology/Cengage Learning

**FIGURE
3.27**

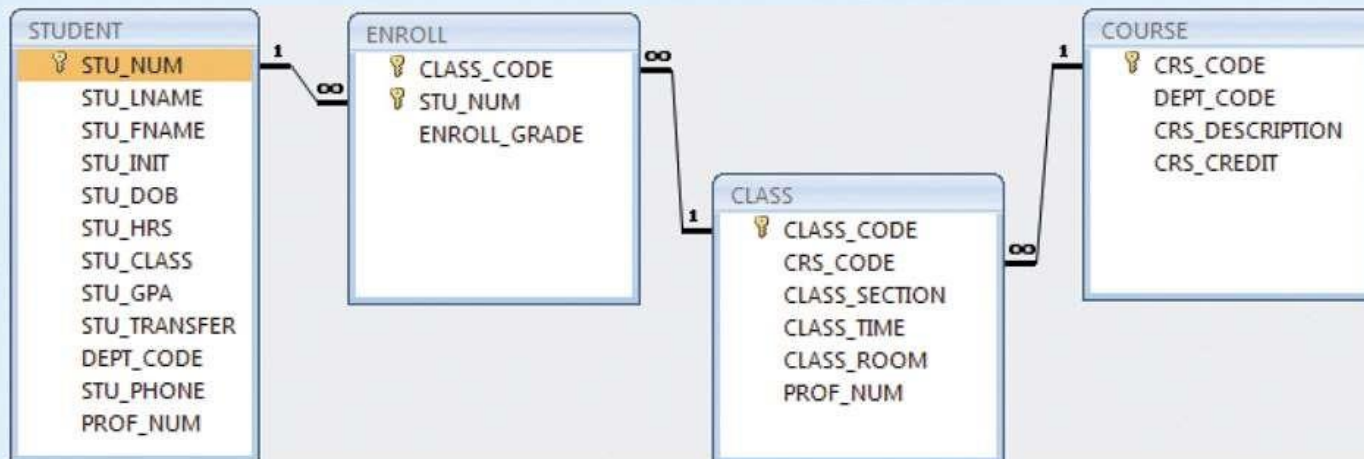
The expanded entity relationship model



SOURCE: Course Technology/Cengage Learning

**FIGURE
3.28**

The relational diagram for the Ch03_TinyCollege database



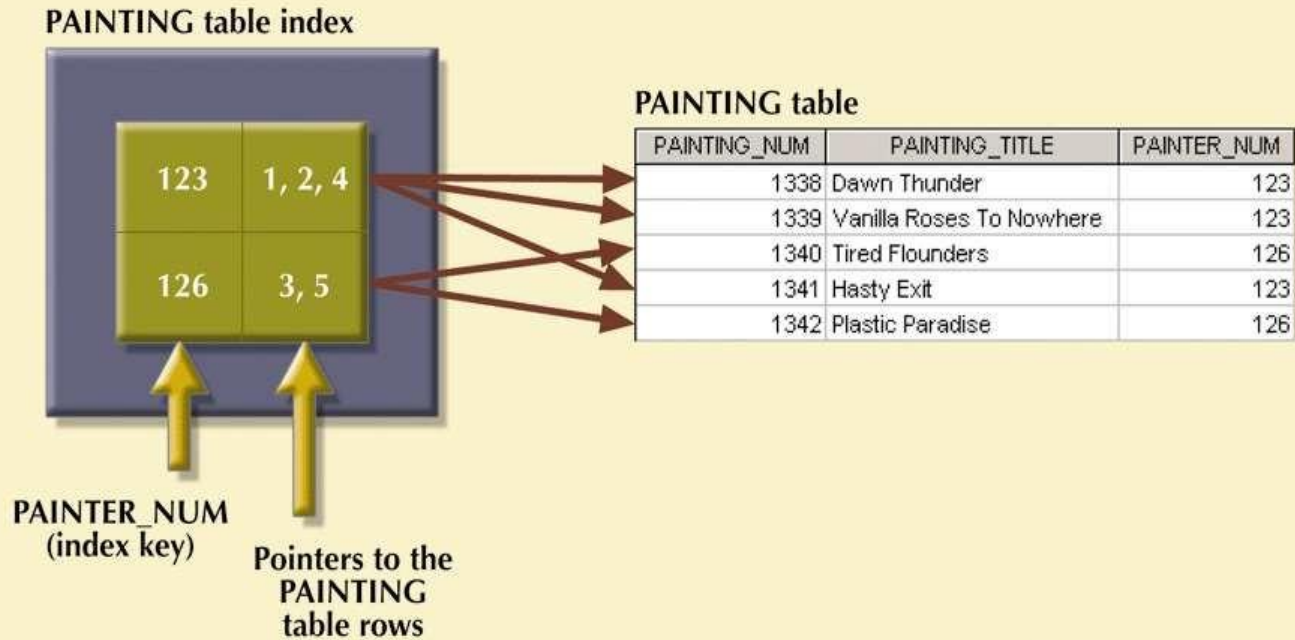
SOURCE: Course Technology/Cengage Learning

İndeksler (Dizinler)

- Bir tablodaki satırlara mantıksal olarak erişmek için bir düzenleme yapılarak aradığınız kaydı bulmak için tüm kayıtlar aranmaz.
- **Dizin anahtarı**
 - **Dizinin referans noktasıdır.**
 - Anahtarla tanımlanan veri konumunu işaret eder.
- **Benzersiz dizin**
 - Dizin anahtarının onunla ilişkilendirilmiş yalnızca bir işaretçi değerine (satır) sahip olabilir.
- **Her dizin yalnızca bir tablo ile ilişkilidir.**

**FIGURE
3.31**

Components of an index



SOURCE: Course Technology/Cengage Learning

Belirli bir PAINTER_NUM için tüm resimlere bakmak için, dizin tam olarak hangi kayıtlara bakacağınızı gösterir.

Codd'un İlişkisel Veritabanı Kuralları

- 1985'te Codd, ilişkisel bir veritabanı sistemi tanımlamak için 12 kuralın bir listesini yayınladı.
 - Minimum ilişkisel standartlara uymayan "ilişkisel" olarak pazarlanan veritabanı uygulaması vardır.
 - Baskın veritabanı satıcıları, bile tüm kuralları tam olarak desteklemez.

TABLE 13.8

DR. CODD'S 12 RELATIONAL DATABASE RULES

RULE	RULE NAME	DESCRIPTION
1	Information	All information in a relational database must be logically represented as column values in rows within tables.
2	Guaranteed access	Every value in a table is guaranteed to be accessible through a combination of table name, primary key value, and column name.
3	Systematic treatment of nulls	Nulls must be represented and treated in a systematic way, independent of data type.
4	Dynamic online catalog based on the relational model	The metadata must be stored and managed as ordinary data—that is, in tables within the database; such data must be available to authorized users using the standard database relational language.
5	Comprehensive data sublanguage	The relational database may support many languages; however, it must support one well-defined, declarative language as well as data definition, view definition, data manipulation (interactive and by program), integrity constraints, authorization, and transaction management (begin, commit, and rollback).
6	View updating	Any view that is theoretically updatable must be updatable through the system.
7	High-level insert, update, and delete	The database must support set-level inserts, updates, and deletes.
8	Physical data independence	Application programs and ad hoc facilities are logically unaffected when physical access methods or storage structures are changed.
9	Logical data independence	Application programs and ad hoc facilities are logically unaffected when changes are made to the table structures that preserve the original table values (changing order of columns or inserting columns).
10	Integrity independence	All relational integrity constraints must be definable in the relational language and stored in the system catalog, not at the application level.
11	Distribution independence	The end users and application programs are unaware of and unaffected by the data location (distributed vs. local databases).
12	Nonsubversion	If the system supports low-level access to the data, users must not be allowed to bypass the integrity rules of the database.
13	Rule zero	All preceding rules are based on the notion that to be considered relational, a database must use its relational facilities exclusively for management.

KAYNAKLAR

- Carlos Coronel, Steven Morris, DATABASE SYSTEMS, Design, Implementation, and Management, Cengage Learning, 13. edition