

isteğe bağlı katılım ER modellemesinde, bir varlık oluşumunun belirli bir ilişkide karşılık gelen bir varlık oluşumunu gerektirmediği bir durumdur.

zorunlu katılım Bir varlık oluşumunun başka bir varlıkta karşılık gelen bir oluşuma sahip olması gereken ilişki. Örneğin, bir ÇALIŞAN bir BÖLÜM'de çalışır. (Bir kişi bir şirketin bölümüne atanmadan çalışan olamaz).

SINIF tanımı gereği zayıf bir varlıktır. Her durumda, SINIF, *zayıf olarak tanımlansın ya da tanımlanmasın*, her zaman varoluş açısından DAVA'ya bağlıdır.

4-1h İlişki Katılımı

Bir varlık ilişkisine katılım ya isteğe bağlıdır ya da zorunludur. İlişkilerin çift yönlü olduğunu hatırlayın; yani her iki yönde de çalışırlar. Eğer KURS, SINIF ile ilişkili ise, tanım gereği, SINIF da KURS ile ilişkilidir. İlişkilerin çift yönlü olması nedeniyle, KURS'tan SINIF'a olan ilişkinin bağlanabilirliğini ve SINIF'tan KURS'a olan ilişkinin bağlanabilirliğini belirlemek gerekir. Benzer şekilde, ilişki için her yönde belirli maksimum ve minimum kardinaliteler belirlenmelidir. Bir kez daha, katılımı belirlerken ilişkinin çift yönlü doğasını göz önünde bulundurmalısınız.

İsteğe bağlı katılım, bir varlık oluşumunun belirli bir ilişkide karşılık gelen bir varlık oluşumunu *gerektirmediği* anlamına gelir. Örneğin, "COURSE generates CLASS" ilişkisinde, en azından bazı kursların bir sınıf oluşturmadığını belirttiniz. Başka bir deyişle, COURSE tablosundaki bir varlık oluşumu (satır), CLASS tablosunda karşılık gelen bir varlık oluşumunun varlığını gerektirmez. (Her bir varlığın bir tablo olarak uygulandığını unutmayın.) Bu nedenle, CLASS varlığının COURSE varlığı için *isteğe bağlı* olduğu kabul edilir. Crow's Foot gösteriminde, varlıklar arasındaki isteğe bağlı bir ilişki, Şekil 4.10'da gösterildiği gibi, isteğe bağlı varlığın yanına küçük bir daire (O) çizilerek gösterilir. *Opsiyonel* bir varlığın varlığı, minimum kardinalitesinin 0 olduğunu gösterir. (*Opsiyonellik* terimi, bir veya daha fazla opsiyonel ilişkinin mevcut olduğu herhangi bir durumu etiketlemek için kullanılır).

Not

İlişki kurma yükünün her zaman yabancı anahtarı içeren varlığın üzerinde olduğunu unutmayın. Çoğu durumda, bu varlık ilişkinin "birçok" tarafındadır.

Zorunlu katılım, bir varlık oluşumunun belirli bir ilişkide karşılık gelen bir varlık oluşumunu *gerektirdiği* anlamına gelir. Varlıkla birlikte herhangi bir opsiyonellik sembolü gösterilmiyorsa, varlığın ilgili zorunlu bir ilişki içinde olduğu varsayılır. Zorunlu katılım grafiksel olarak gösterilirse, tipik olarak, 1'lik bir bağlantının Karga Ayağı tasvirine benzer şekilde, ilişki çizgisi boyunca küçük bir karma işareti olarak gösterilir. Zorunlu bir ilişkinin varlığı, zorunlu varlık için minimum kardinalitenin en az 1 olduğunu gösterir.

Not

İsteğe bağlı bir ilişkide varlıklar arasında meydana geldiklerinde ilişkilerin zayıf olduğu ve zorunlu bir ilişkide varlıklar arasında meydana geldiklerinde ilişkilerin güçlü olduğu sonucuna varmak cazip gelebilir. Ancak bu sonuca varmak doğru değildir. İlişki katılımı ve ilişki gücünün aynı şeyi unutmayın. Bir varlık diğeri için isteğe bağlı olduğunda güçlü bir ilişkiyle karşılaşmanız muhtemeldir. Örneğin, ÇALIŞAN ile BAĞIMLI arasındaki ilişki açıkça güçlü ilişkidir, ancak BAĞIMLI açıkça ÇALIŞAN için isteğe bağlıdır. Sonuçta, çalışanların bakmakla yükümlü oldukları kişiler olmasını zorunlu tutamazsınız. Ayrıca, bir varlık diğeri için zorunlu olduğunda zayıf bir ilişki kurulması da mümkündür. İlişkinin gücü, ilgili varlığın PK'sının nasıl formüle edildiğine bağlıyken, ilişkiye katılım iş kuralının nasıl yazıldığına bağlıdır. Örneğin, "Her parça bir satıcı tarafından tedarik edilmelidir" ve "Bir parça bir satıcı tarafından tedarik edilebilir veya edilmeyebilir" iş kuralları aynı için farklı opsiyonellikler yaratır! Bu ayrımın anlaşılması, tablo satırları eklendiğinde veya silindiğinde büyük sorunlara neden olan kötü tasarım kararlarına yol açabilir.

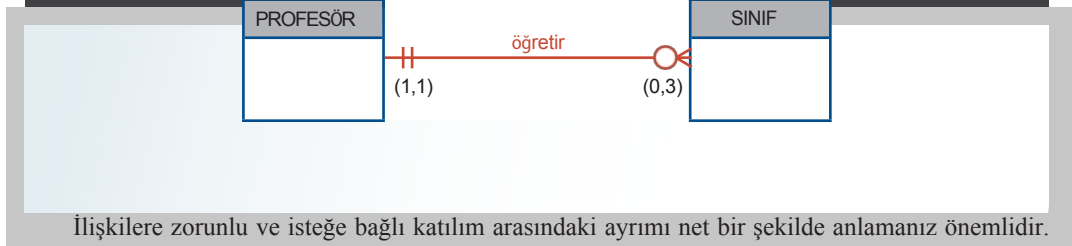
Tablo 4.3, Crow's Foot notasyonu tarafından desteklenen çeşitli bağlantı ve katılım kombinasyonlarını göstermektedir. Belirli kardinaliteler kullanılmadığında bu kombinasyonların Crow's Foot gösteriminde genellikle kardinalite olarak adlandırıldığını hatırlayın.

Tablo 4.3 Karga Ayağı Sembolleri

Sembol	Kardinalite	Yorum
	(0,N)	Sıfır veya çok; "çok" tarafı isteğe bağlıdır.
	(1,N)	Bir veya birçok; "birçok" tarafı zorunludur.
	(1,1)	Bir ve sadece bir tane; "1" tarafı zorunludur.
	(0,1)	Sıfır veya bir; "1" tarafı isteğe bağlıdır.

İlişki katılımı veritabanı tasarımının önemli bir bileşeni olduğundan, birkaç senaryoyu daha incelemelisiniz. Tiny College'da ders vermeksizin araştırma yapan bazı profesörlerin çalıştığını varsayalım. "PROFESÖR SINIF öğretir" ilişkisini incellerseniz, bir PROFESÖRÜN SINIF öğretmemesi oldukça mümkündür. Bu nedenle, SINIF PROFESÖR için *isteğe bağlıdır*. Öte yandan, bir SINIF bir PROFESÖR tarafından öğretilmelidir. Bu nedenle PROFESÖR, SINIF için *zorunludur*. Şekil 4.13'teki ERD modelinin CLASS'ın yanındaki kardinaliteyi (0,3) olarak gösterdiğine dikkat edin, bu da bir profesörün hiç ders vermeyebileceğini veya en fazla üç ders verebileceğini gösterir. Ayrıca, her CLASS tablo satırı, PROFESSOR tablosunun yanındaki (1,1) kardinalitesi ile temsil edilen bir ve yalnızca bir PROFESSOR satırına referans verir - her sınıfın bir ve yalnızca bir profesör tarafından verildiği varsayılır.

Şekil 4.13 "PROFESÖR SINIF ÖĞRETİR" İlişkisinde İsteğe Bağlı Bir SINIF Varlığı

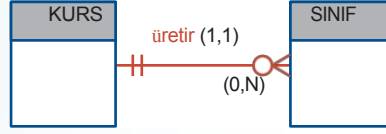


İlişkilere zorunlu ve isteğe bağlı katılım arasındaki ayrımı net bir şekilde anlamamız önemlidir. Aksi takdirde, sadece gerekli varlıkların oluşturulmasını sağlamak için garip ve gereksiz geçici satırların (varlık örnekleri) oluşturulması gereken tasarımlar geliştirebilirsiniz.

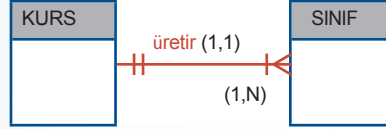
Bir sorunun anlamının bir ilişkiye katılım türünü belirleyebileceğini anlamak da önemlidir. Örneğin, Tiny College'ın çeşitli kurslar sunduğunu varsayalım; her kursun çeşitli sınıfları vardır. Bu tartışmada *sınıf* ve *kurs* arasındaki ayrıma tekrar dikkat edin: bir SINIF, bir KURSUN belirli bir sunumunu (veya bölümünü) oluşturur. Tipik olarak, kurslar üniversitenin kurs kataloğunda listelenirken, sınıflar öğrencilerin derslerine kaydolmak için kullandıkları sınıf programlarında listelenir.

CLASS varlığının "COURSE generates CLASS" ilişkisine katkısını analiz ederek, bir COURSE olmadan bir CLASS'ın var olamayacağını görmek kolaydır. Bu nedenle, COURSE varlığının ilişkide *zorunlu* olduğu sonucuna varabilirsiniz. Ancak, Şekil 4.14 ve 4.15'te gösterildiği gibi, CLASS varlığı için iki senaryo yazılabilir.

Şekil 4.14 SINIF, KURS için Opsiyoneldir



Şekil 4.15 Zorunlu Bir İlişki İçinde KURS ve SINIF



Farklı senaryolar problemin semantiğinin bir fonksiyonudur; yani ilişkinin nasıl tanımlandığına .

1. *SINIF isteğe bağlıdır.* Departmanın önce COURSE varlığını oluşturması ve ardından öğretim atamalarını yaptıktan sonra CLASS varlığını oluşturması mümkündür. Gerçek dünyada böyle bir senaryo çok olasıdır; bölümleri (sınıfları) henüz tanımlanmamış dersler olabilir. Aslında, bazı dersler yılda sadece bir kez okutulur ve her dönem sınıf oluşturmaz.
2. *SINIF zorunludur.* Bu koşul, "Her KURS bir veya daha fazla SINIF üretir" ifadesinin semantiği tarafından dayatılan kısıtlama tarafından oluşturulur. ER terimleriyle, "üretir" ilişkisindeki her KURS en az bir SINIF'a sahip olmalıdır. Bu nedenle, problemin semantiğine uymak için KURS oluşturulurken bir SINIF oluşturulmalıdır.

Şekil 4.15'te sunulan senaryonun pratik yönlerini aklınızda bulundurun. Bu ilişkinin semantiği göz önüne alındığında, sistem en az bir sınıf bölümüyle ilişkilendirilmemiş bir kursu kabul etmemelidir. Operasyonel açıdan bu kadar katı bir ortam arzu edilir mi? Örneğin, yeni bir KURS oluşturulduğunda, veritabanı önce KURS tablosunu günceller ve böylece henüz ilişkilendirilmiş bir SINIF bulunmayan bir KURS varlığı ekler. Doğal olarak, CLASS varlıkları ilgili CLASS tablosuna eklendiğinde görünürdeki sorun çözülmüş gibi görünmektedir. Ancak, zorunlu ilişki nedeniyle, sistem geçici olarak iş kuralı kısıtlamasını ihlal edecektir. Pratik amaçlar doğrultusunda, daha esnek bir tasarım üretmek için CLASS'ın isteğe bağlı olarak sınıflandırılması arzu edilir.

Son olarak, Şekil 4.14 ve 4.15' senaryoları incelerken, VTYS'nin rolünü aklınızda tutun. Veri bütünlüğünü korumak için VTYS, "çok" tarafın (SINIF) yabancı anahtar kuralları aracılığıyla bir KURS ile ilişkilendirilmesini sağlamalıdır.

41 i İlişki Derecesi

Bir **ilişki derecesi**, bir ilişkiyle ilişkili varlıkların veya katılımcıların sayısını gösterir. Bir **ilişki** tek bir varlık içinde sürdürüldüğünde **tekli** bir ilişki mevcuttur. İki varlık ilişkilendirildiğinde **ikili** bir **ilişki** mevcuttur. Üç varlık ilişkilendirildiğinde **üçlü bir** ilişki mevcuttur. Daha yüksek dereceler mevcut olsa da, bunlar nadirdir ve özel olarak adlandırılmazlar. (Örneğin, dört varlıktan oluşan bir ilişki basitçe *dört dereceli* bir ilişki olarak tanımlanır). Şekil 4.16'da bu tür ilişki dereceleri gösterilmektedir.

ilişki derecesi

Bir ilişkiyle ilişkili varlıkların veya katılımcıların sayısı. Bir ilişki derecesi tekli, ikili, üçlü veya daha yüksek olabilir.

tekli ilişki

Bir varlık içindeki bir ilişkiyi tanımlamak için kullanılan bir ER terimi. Örneğin, bir EMPLOYEE başka bir EMPLOYEE'yi yönetebilir.

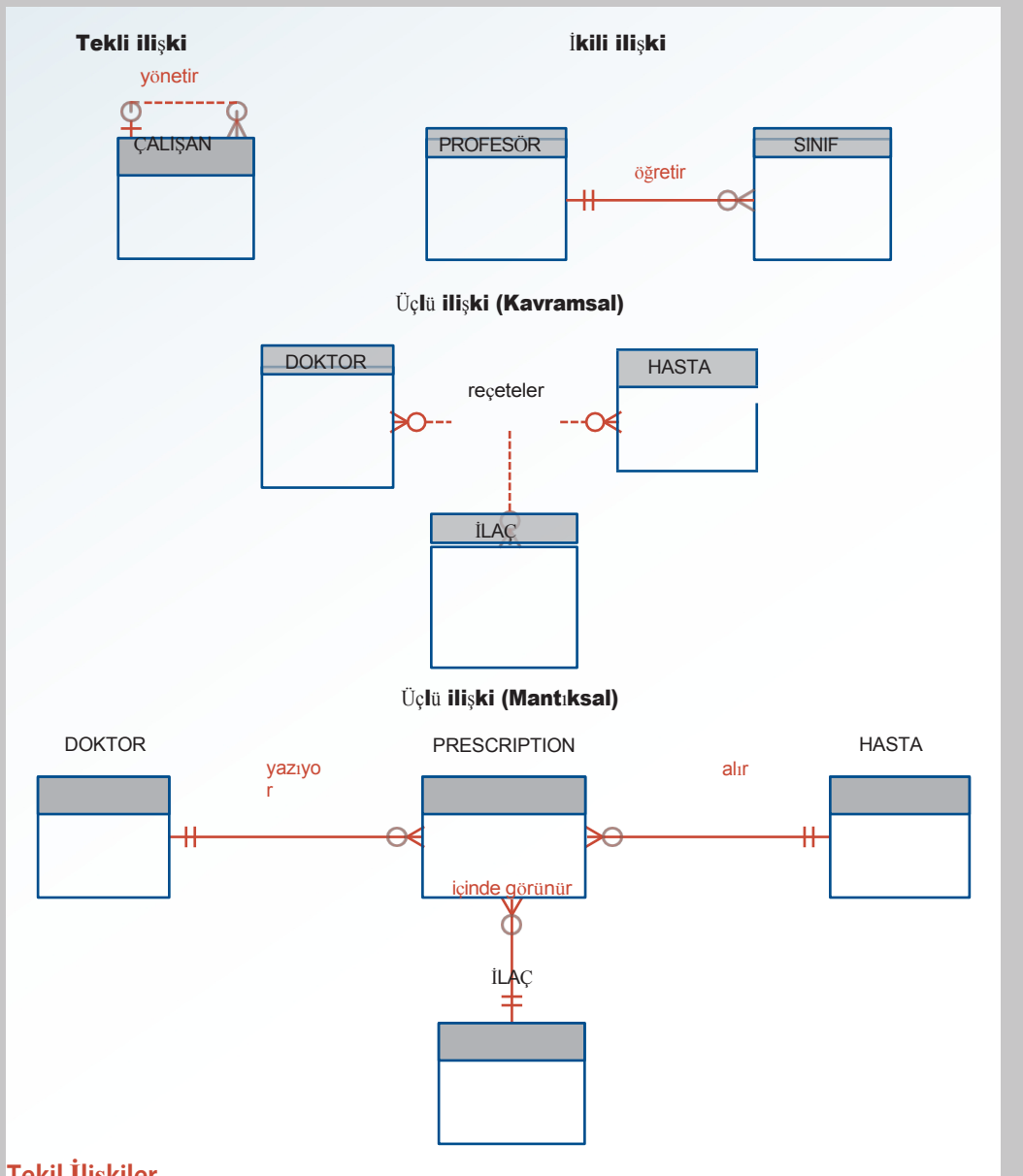
ikili ilişki

Arasındaki bir ilişki (association) için bir ER terimi iki varlık. Örneğin, PROFESÖR SINIF'a ders verir.

üçlü ilişki

Üç varlık arasındaki ilişkiyi (association) tanımlamak için kullanılan bir ER terimi. Örneğin, bir DOKTOR bir HASTA için bir İLAÇ reçete eder.

Şekil 4.16 Üç Tür İlişki Derecesi



Tekil İlişkiler

Şekil 4.16'da gösterilen tekli ilişki durumunda, EMPLOYEE varlığı içindeki bir çalışan, bu varlık içindeki bir veya daha fazla çalışanın yöneticisidir. Bu durumda, "yönetir" ilişkisinin varlığı, EMPLOYEE'nin yönetici olmak için başka bir EMPLOYEE'ye ihtiyaç duyduğu, yani EMPLOYEE'nin kendisiyle bir ilişkisi olduğu anlamına gelir. Böyle bir ilişki özyinelemeli ilişki olarak bilinir. Özyinelemeli ilişkilerin çeşitli durumları Bölüm 4-1j'de açıklanmıştır.

İkili İlişkiler

İki varlık bir ilişki içinde ilişkilendirildiğinde ikili bir ilişki ortaya çıkar. İkili ilişkiler en yaygın türüdür. Aslında, kavramsal tasarımı basitleştirmek için, çoğu yüksek dereceli (üçlü ve daha yüksek) ilişki mümkün olduğunda uygun eşdeğer ikili ilişkilere ayrıştırılır. Şekil 4.16'da, "bir PROFESÖR bir veya daha fazla SINIF öğretir" ikili bir ilişkiyi temsil etmektedir.

özyinelemeli ilişki Tek bir varlık türü içinde bulunan bir ilişki. Örneğin, bir EMPLOYEE bir EMPLOYEE ile evlidir veya bir PART başka bir PART'ın bileşenidir.

Üçlü ve Daha Yüksek Dereceli İlişkiler

Çoğu ilişki ikili olmasına rağmen, üçlü ve daha yüksek dereceli ilişkilerin kullanımı tasarımcıya bir problemin semantiği konusunda biraz serbestlik sağlar. Üçlü bir ilişki, üç farklı varlık arasında bir ilişki olduğu anlamına gelir. Örneğin, Şekil 4.17'de, aşağıdaki iş kuralları ile temsil edilen ilişkilere ve bunların sonuçlarına dikkat edin:

- Bir DOKTOR bir veya daha fazla REÇETE yazar.
- Bir HASTA bir veya daha fazla REÇETE alabilir.
- Bir İLAÇ bir veya daha fazla REÇETEDEN görünebilir. (Bu örneği basitleştirmek için, iş kuralının her reçetenin yalnızca bir ilaç içerdiğini belirttiğini varsayın. Kısacası, bir doktor birden fazla ilaç reçete ederse, her ilaç için ayrı bir reçete yazılmalıdır).

Şekil 4.17 Üçlü İlişkinin Uygulanması

Veritabanı adı: Ch04_Clinic

Tablo adı: DRUG

DRUG_CODE	DRUG_NAME	DRUG_PRICE
AF15	Afgapan-15	25.00
AF25	Afgapan-25	35.00
DRO	Droalene Chloride	111.89
DRZ	Druzochlor Cryptolene	18.99
KO15	Koliabar Oxyhexalene	65.75
OLE	Oleander-Drizapan	123.95
TRYP	Tryptolac Heptadimetric	79.45

Tablo adı: HASTA

PAT_NUM	PAT_TITLE	PAT_LNAME	PAT_FNAME	PAT_INITIAL	PAT_DOB	PAT_AREACODE	PAT_PHONE
100	Mr.	Kolmycz	George	D	15-Jun-1942	615	324-5456
101	Ms.	Lewis	Rhonda	G	19-Mar-2005	615	324-4472
102	Mr.	Vandam	Rhett		14-Nov-1958	901	675-8993
103	Ms.	Jones	Anne	M	16-Oct-1974	615	898-3456
104	Mr.	Lange	John	P	08-Nov-1971	901	504-4430
105	Mr.	vWilliams	Robert	D	14-Mar-1975	615	890-3220
106	Mrs.	Smith	Jeanine	K	12-Feb-2003	615	324-7883
107	Mr.	Diante	Jorge	D	21-Aug-1974	615	890-4567
108	Mr.	vWiesenbach	Paul	R	14-Feb-1966	615	897-4358
109	Mr.	Smith	George	K	18-Jun-1961	901	504-3339
110	Mrs.	Genkazi	Leighla	vW	19-May-1970	901	569-0093
111	Mr.	vWashington	Rupert	E	03-Jan-1966	615	890-4925
112	Mr.	Johnson	Edward	E	14-May-1961	615	898-4387
113	Ms.	Smythe	Melanie	P	15-Sep-1970	615	324-9006
114	Ms.	Brandon	Marie	G	02-Nov-1932	901	882-0845
115	Mrs.	Saranda	Hermine	R	25-Jul-1972	615	324-5505
116	Mr.	Smith	George	A	08-Nov-1965	615	890-2984

Tablo adı: DOKTOR

DOC_ID	DOC_LNAME	DOC_FNAME	DOC_INITIAL	DOC_SPECIALTY
29827	Sanchez	Julio	J	Dermatology
32445	Jorgensen	Annelise	G	Neurology
33456	Korenski	Anatoly	A	Urology
33989	LeGrande	George		Pediatrics
34409	vWashington	Dennis	F	Orthopaedics
36221	McPherson	Katye	H	Dermatology
36712	Dreifag	Herman	G	Psychiatry
38995	Minh	Tran		Neurology
40004	Chin	Ming	D	Orthopaedics
40028	Feinstein	Denise	L	Gynecology

Tablo adı: PRESCRIPTION

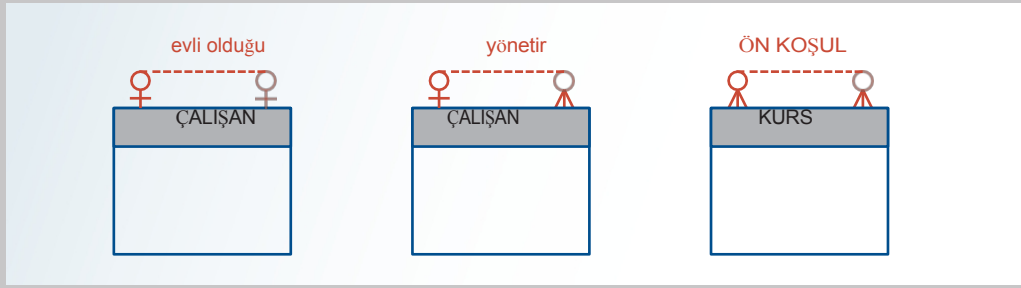
DOC_ID	PAT_NUM	DRUG_CODE	PRES_DOSAGE	PRES_DATE
32445	102	DRZ	2 tablets every four hours -- 50 tablets total	12-Nov-22
32445	113	OLE	1 teaspoon with each meal -- 250 ml total	14-Nov-22
34409	101	KO15	1 tablet every six hours -- 30 tablets total	14-Nov-22
36221	109	DRO	2 tablets with every meal -- 60 tablets total	14-Nov-22
38995	107	KO15	1 tablet every six hours -- 30 tablets total	14-Nov-22

Şekil 4.17'deki tablo içeriğini incelerken, tüm trans-aksiyonları izlemenin mümkün olduğuna dikkat edin. Örneğin, ilk reçetenin doktor 32445 tarafından hasta 102 için DRZ ilacı kullanılarak yazıldığını söyleyebilirsiniz.

4-1j Özyinelemeli İlişkiler

Az önce öğrendiğiniz gibi, özyinelemeli bir ilişki, aynı varlık kümesinin oluşumları arasında bir ilişkinin var olabileceği bir ilişkidir. (Doğal olarak, böyle bir koşul bir tekli ilişki içinde bulunur.) Örneğin, 1:1 tekli ilişki "bir ÇALIŞAN bir ve yalnızca bir başka ÇALIŞAN ile evli olabilir" şeklinde ifade edilebilir. Ayrıca, 1:M tekli ilişki "bir ÇALIŞAN birçok ÇALIŞANI yönetebilir ve her ÇALIŞAN bir ÇALIŞAN tarafından yönetilir" şeklinde ifade edilebilir. Son olarak, M:N tekil ilişkisi "bir KURS diğer birçok KURS için ön koşul olabilir ve her KURS'un ön koşul olarak birçok başka KURS'u olabilir" şeklinde ifade edilebilir. Bu ilişkiler Şekil 4.18'de gösterilmektedir.

Şekil 4.18 Özyinelemeli İlişkilerin ER Temsili



Şekil 4.18'de gösterilen 1:1 ilişkisi Şekil 4.19'da gösterilen tek tabloda uygulanabilir. Tekli ilişkilerde yabancı anahtarların yerleştirilmesi, ikili ilişkilerde yabancı anahtarların yerleştirilmesiyle aynı kuralları izler. Tek fark, bir eşanlamli kullanımının gerekli olmasıdır. İkili 1:1 ilişkisinde olduğu gibi, ilişki satırının bir tarafındaki varlıktan birincil anahtar alır ve bunu ilişki satırının diğer tarafındaki varlığa yabancı anahtar olarak yerleştirirsiniz. Ancak tekli ilişkilerde bunlar aynı varlıktır! Varlığın birincil anahtarını yabancı anahtar olarak hareket etmesi için ikinci kez varlığa yerleştirirsiniz. Bölüm 3'ten bildiğiniz gibi, bir tablodaki iki öznitelik aynı ada sahip olamaz, bu nedenle yabancı anahtar yeniden adlandırılır. Seçilen ismin temsil ettiği şeyi açıklayıcı olması gerekir. Şekil 4.19', EMP_NUM yabancı anahtar olarak hareket etmek üzere EMPLOYEE varlığına ikinci kez yerleştirildiğinde, herhangi bir satır için bu sütundaki çalışan numarasının o satırda temsil edilen çalışanın eşinin çalışan numarası olduğunu belirtmek üzere EMP_SPOUSE olarak yeniden adlandırılmıştır. James Ramirez'in, James Ramirez ile evli olan Louise Ramirez ile evli olduğunu belirleyebilirsiniz. Ayrıca Anne Jones, Anne Jones ile evli olan Anton Shapiro ile evlidir. Tekil ilişkilerle çalışırken sık karşılaşılan bir tuzak, katılım ile referans bütünlüğünü birbirine karıştırmaktır. Teoride, katılım ve referans bütünlüğü çok farklı kavramlardır ve normalde ikili ilişkilerde ayırt edilmeleri kolaydır. Pratikte ise tam tersine, katılım ve referans bütünlüğü birbirine çok benzer çünkü her ikisi de aynı nitelikler kümesi üzerindeki kısıtlamalar aracılığıyla uygulanır. Bu benzerlik, kavramlar tekli bir ilişkinin sınırlı yapısı içinde uygulandığında genellikle karışıklığa yol açar. Çalışanlar arasında Şekil 4.19'da tarif edilen tek taraflı 1:1 eş ilişkisini göz önünde bulundurun. Katılım, daha önce açıklandığı gibi, çift yönlüdür, yani her iki yönde de ele alınmalıdır

ilişki boyunca. Şekil 4.19'daki katılım aşağıdaki soruları ele almaktadır:

- Her çalışanın eşi de çalışan olmak zorunda mıdır?
- Her çalışan başka bir çalışanın eşi olmak zorunda mıdır?

Şekil 4.19'da gösterilen veriler için her iki sorunun da doğru cevabı "Hayır"dır. Hem çalışan olup hem de eş olarak başka bir çalışana sahip olmamak mümkündür. Ayrıca, hem çalışan olup hem de başka bir çalışanın eşi olmamak da mümkündür.

Referans bütünlüğü, yabancı anahtardaki değerlerin ilgili birincil anahtardaki değerlerle uyumu ile ilgilendir. Referans bütünlüğü çift yönlü değildir ve bu nedenle yalnızca tek bir soruya yanıt verir:

- Her çalışan eşi geçerli bir çalışan olmak zorunda mıdır?

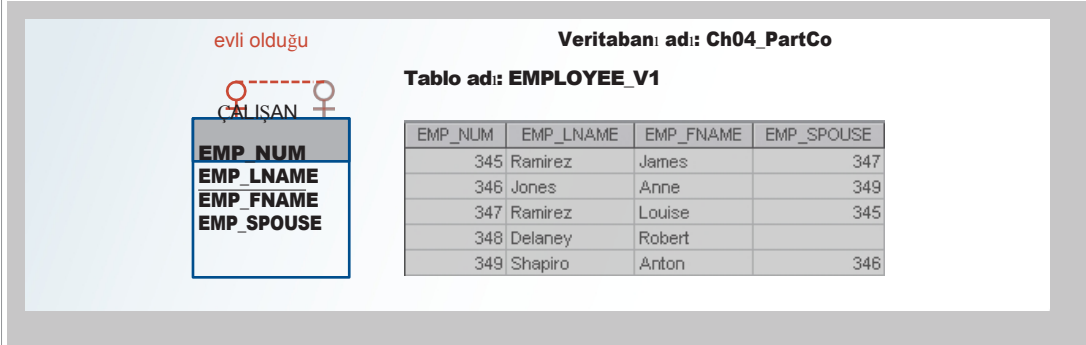
Şekil 4.19'da gösterilen veriler için doğru cevap "Evet"tir. Bu soruyu çerçevelemenin bir başka yolu da EMP_SPOUSE özniteliği için sağlanan her değer EMP_NUM özniteliğindeki bazı değerlerle eşleşmesi gerekip gerekmediğini düşünmektir.

Pratik anlamda, hem katılım hem de referans bütünlüğü, ilişkiyi uygulamak için birincil anahtarlar ve yabancı anahtarlar olarak kullanılan değerleri içerir. Referans bütünlüğü, yabancı anahtardaki değerlerin birincil anahtardaki değerlere karşılık gelmesini gerektirir. Bir yönde, katılım yabancı anahtarın bir null içerip içermediğini dikkate alır. Örneğin Şekil 4.19'da,

Robert Delaney çalışanın EMP_SPOUSE içinde bir değere sahip olması gerekmez. Diğer yönde, katılım birincil anahtardaki her değer yabancı anahtarda bir değer olarak görünmesi gerekip gerekmediğini dikkate alır. Örneğin Şekil 4.19'da, çalışan Robert Delaney'nin EMP_NUM değeri

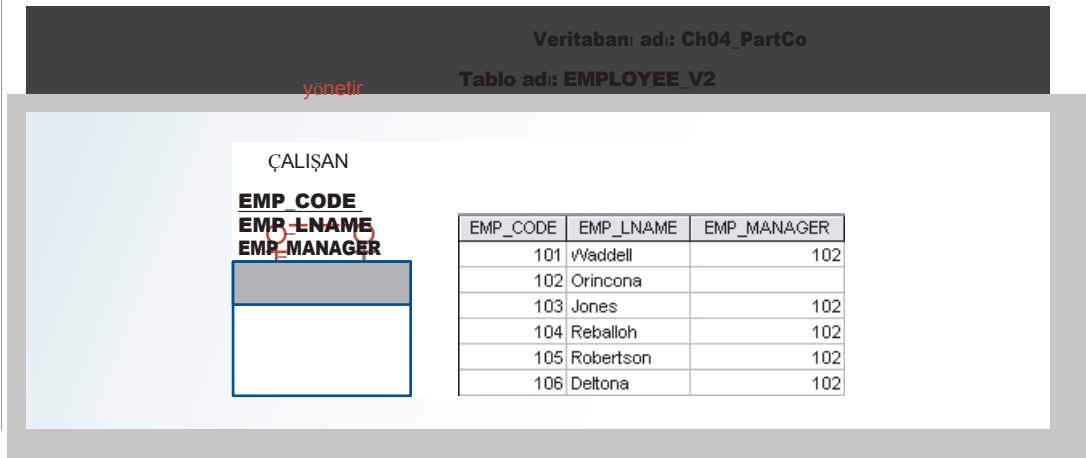
(348) değerinin başka herhangi bir çalışan için EMP_SPOUSE içinde bir değer olarak görünmesi gerekmez.

Şekil 4.19 1:1 Özyinelemeli İlişki "ÇALIŞAN, ÇALIŞAN ile evlidir"



Şekil 4.18'de gösterilen 1:M özyinelemeli ilişki "EMPLOYEE manages EMPLOYEE", Şekil 4.20'de uygulanmıştır. Yine, yabancı anahtar yerleştirme kuralları aynıdır. 1:M ilişkisi ile, "1" tarafındaki varlığın anahtarı "M" tarafındaki varlığa yabancı anahtar olarak yerleştirilir. Yine tekli bir ilişkide, aynı varlık ilişki satırının her iki tarafında da yer alır, bu nedenle birincil anahtar yabancı anahtar olarak yine varlığın içine yerleştirilir. 1:M ilişkisi ile eşanlamlı seçerken, öznitelik adının açıklayıcı olmasına ekstra özen gösterilmelidir. Bölüm 3'ten 1:1 ilişkisinde yabancı anahtarı her iki yönde de yerleştirebileceğimizi hatırlayın, ancak 1:M ilişkisinde yabancı anahtarı doğru yönde yerleştirmek ("birin anahtarı çoğa gider") çok önemlidir. Şekil 4.20'de, yabancı anahtar EMP_CODE'un yeniden adlandırılmış ikinci bir kopyasıdır. EMP_CODE'un bu ikinci kopyası, bu satırda açıklanan çalışan tarafından yönetilen kişinin çalışan kodunu mu yoksa bu satırda açıklanan çalışmanı yöneten kişinin çalışan kodunu mu içermelidir? Bir yöneticinin çok sayıda çalışmanı vardır ve verilen senaryoda her çalışanın yalnızca bir yöneticisi vardır. Bu nedenle, yabancı anahtar, satırda açıklanan çalışanın yöneticisinin çalışan kodu olmalıdır. Bu nedenle EMP_MANAGER eşanlamlısı açıklayıcı bir öznitelik adı olarak seçilmiştir. Şekil 4.20'de gösterilen verilere bakıldığında, Waddell çalışanın Orinconca çalışmanı tarafından yönetildiği açıktır. Yabancı anahtarı yanlış yerleştirmiş olsaydınız, Orinconca'nın yabancı anahtarda beş değer saklaması gerekirdi ve bu da ilişkisel tablo kurallarını ihlal ederdi.

Şekil 4.20

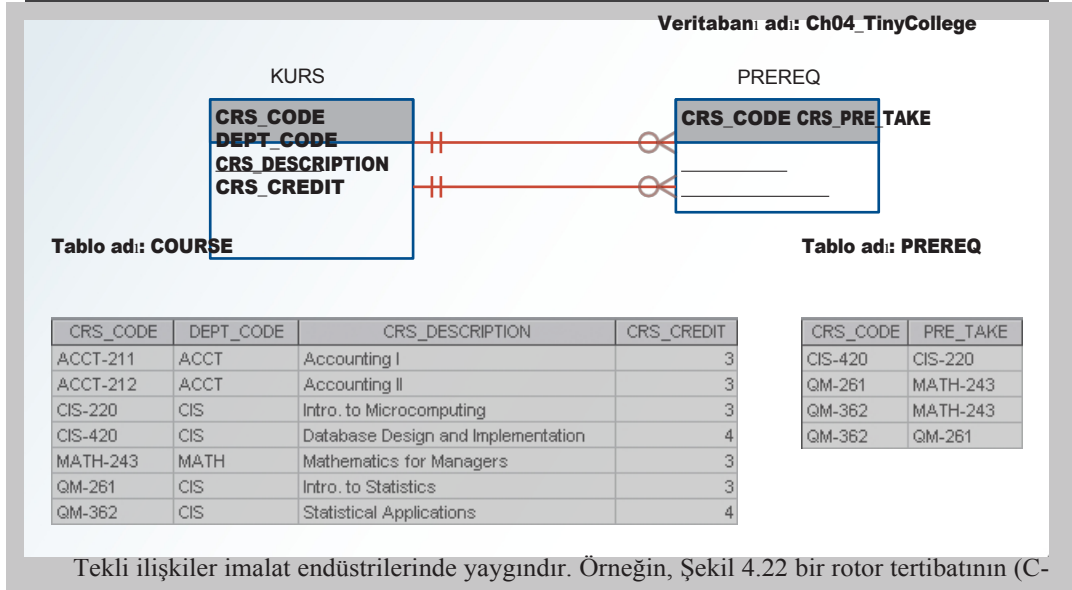


M:N özyinelemeli ilişki size okul ortamından tanıdık gelebilir. Örneğin, Şekilde gösterilen M:N "DERS, DERS gerektirir" ilişkisinin nasıl olduğuna dikkat edin

4.18 Şekil 4.21'de uygulanmıştır. M:N tekli ilişkiler, Bölüm 3'te sunulduğu ve bu bölümde daha ayrıntılı olarak ele alındığı gibi köprülerin veya bağlantı tablolarının oluşturulmasını gerektirir. Süreç ikili M:N ilişkileriyle aynıdır. Karmaşık ilişki köprü haline . Orijinal ana varlıkların birincil anahtarları köprüde bileşik bir birincil anahtar olarak birleştirilir. Tekli bir ilişkide, her iki ebeveyn olarak hareket eden yalnızca bir varlık olduğuna dikkat edin. Bu örnekte, COURSE tablosu ilişki çizgisinin her iki ucundadır, bu nedenle her iki tarafta da ebeveynidir. Bu da CRS_CODE'un PREREQ adlı köprü tablosuna iki kez yerleştirildiği anlamına gelir. Diğer tekli ilişki uygulamalarında olduğu gibi, yinelenen öznitelik adlarından kaçınmak için özniteliğin en az bir kopyasının yeniden adlandırılması gerekir. Bu durumda, CRS_CODE'un bir kopyası CRS_PRE_TAKE olarak yeniden adlandırılmıştır. Bu, COURSE ve PREREQ arasında iki adet 1:M ilişkisi oluşturur. COURSE içindeki CRS_CODE'un birincil anahtar olduğu ve PREREQ içindeki CRS_CODE'un ilgili yabancı anahtar olduğu bir ilişki vardır. Diğer ilişkide de COURSE içindeki CRS_CODE birincil ve PREREQ içindeki CRS_PRE_TAKE ilgili yabancı anahtardır. Bu örnekte, MATH-243, CRS_CODE-to-CRS_PRE_TAKE ilişkisi kullanılarak QM-261 ve QM-362 için bir ön koşuldur. Hem MATH-243 hem de QM-261, CRS_CODE-to-CRS_CODE ilişkisi kullanılarak QM-362 için ön koşuldur.

Şekil 4.21

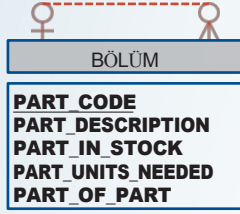
gerektirir"



Tekli ilişkiler imalat endüstrilerinde yaygındır. Örneğin, Şekil 4.22 bir rotor tertibatının (C-130) birçok parçadan oluştuğunu, ancak her bir parçanın yalnızca bir rotor tertibatı oluşturmak için kullanıldığını göstermektedir. Şekil 4.22 bir rotor tertibatının dört adet 2,5 cm'lik pul, iki adet çatal pim, bir adet 2,5 cm'lik çelik shaft, dört adet 10,25 cm'lik rotor kanadı ve iki adet 2,5 cm'lik altıgen somundan oluştuğunu göstermektedir. Şekil 4.22'de uygulanan ilişki, böylece her bir rotor tertibatı içindeki her bir parçayı izlemenizi sağlar. Tekli ilişkilerin, ikili ilişkilerde olduğu PART_UNITS_NEEDED gibi ilişkiyi tanımlayan ek nitelikleri kolayca barındırabileceğini unutmayın.

Bir parça birkaç farklı türde başka parçayı birleştirmek için kullanılabiliyorsa ve kendisi de birçok parçadan oluşuyorsa, "PARÇA PARÇAYI İÇERİR" ilişkisini uygulamak için iki tablo gerekir. Şekil 4.23 böyle bir ortamı göstermektedir. Yöneticiler daha karmaşık çıktılar üretmenin yasal sonuçlarının daha fazla farkına vardıkça parça takibi giderek daha önemli hale gelmektedir. Birçok endüstride, özellikle havacılıkla ilgili olanlarda, tam parça takibi kanunen zorunludur.

Şekil 4.22 Başka Bir Tekli İlişki: "PARÇA PARÇAYI İÇERİR"

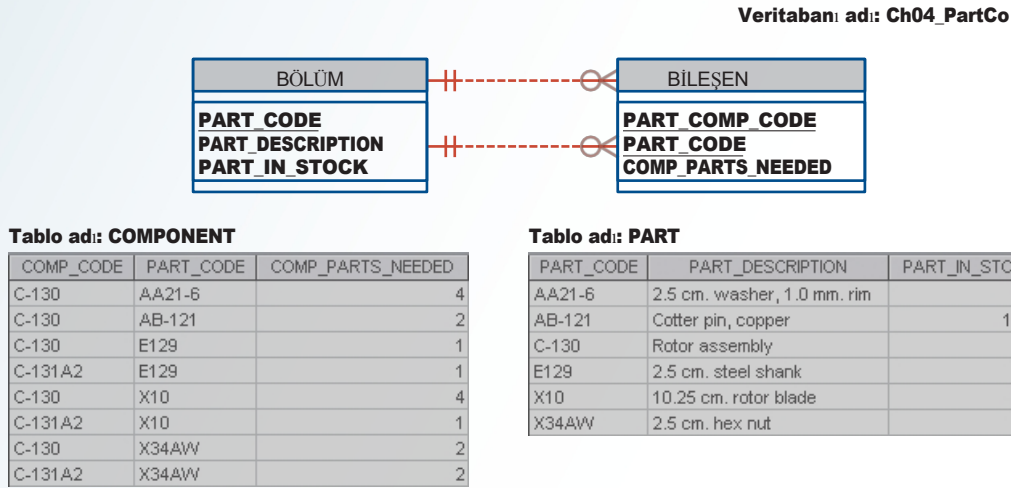


Tablo adı: PART_V1

PART_CODE	PART_DESCRIPTION	PART_IN_STOCK	PART_UNITS_NEEDED	PART_OF_PART
AA21-6	2.5 cm. washer, 1.0 mm. rim	432	4	C-130
AB-121	Cotter pin, copper	1034	2	C-130
C-130	Rotor assembly	36		
E129	2.5 cm. steel shank	128	1	C-130
X10	10.25 cm. rotor blade	345	4	C-130
X34AVV	2.5 cm. hex nut	879	2	C-130

Veritabanı adı: Ch04_PartCo

Şekil 4.23 M:N Özyinelemeli İlişkinin Uygulanması "PARÇA PARÇAYI İÇERİR"



Veritabanı adı: Ch04_PartCo

4-1k İlişkisel (Bileşik) Varlıklar

M:N ilişkileri kavramsal düzeyde geçerli bir yapıdır ve bu nedenle ER modelleme sürecinde sıkça rastlanır. Ancak, özellikle ilişkisel modelde M:N ilişkisinin uygulanması, Bölüm 2'de öğrendiğiniz gibi ek bir varlığın kullanılmasını gerektirir.

3. ER modeli, iki veya daha fazla varlık arasındaki M:N ilişkisini temsil etmek için ilişkisel varlığı kullanır. *Bileşik* veya *köprü varlık* olarak da adlandırılan bu ilişkisel varlık, ana varlıklarla 1:M ilişkisi içindedir ve her bir ana varlığın birincil anahtar niteliklerinden oluşur. Ayrıca, Şekil 4.24'teki ENROLL ilişkisel varlığında gibi, ilişkisel varlık kendi ek niteliklerine sahip olabilir. Karga Ayağı notasyonu kullanıldığında, ilişkilendirilmiş varlık, ebeveynler ve ilişkilendirilmiş varlık arasındaki düz ilişki çizgileriyle gösterildiği gibi, güçlü (tanımlayıcı) bir ilişki olarak tanımlanır.

Şekil 4.24'teki bileşik ENROLL varlığının varlığının diğer iki varlığa bağlı olduğuna dikkat edin; ENROLL varlığının bileşimi, bileşik varlık tarafından bağlanan birincil anahtarlarına dayanır. Bileşik varlık, bağlanma sürecinde hiçbir rol oynamayan ilave nitelikler de içerebilir. Örneğin, varlığın en azından STUDENT ve CLASS birincil anahtarlarından oluşması gerekmesine rağmen, notlar, devamsızlıklar ve öğrencinin belirli bir sınıftaki performansı ile benzersiz bir şekilde tanımlanan diğer veriler gibi ek nitelikler de içerebilir.

Şekil 4.24 M:N İlişkisinin İki 1:M Dönüştürülmesi

Tablo adı: ÖĞRENCİ		Veritabanı adı: Ch04_CollegeTry	
STU_NUM	STU_LNAME		
321452	Bowser		
324257	Smithson		

Tablo adı: ENROLL		
CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

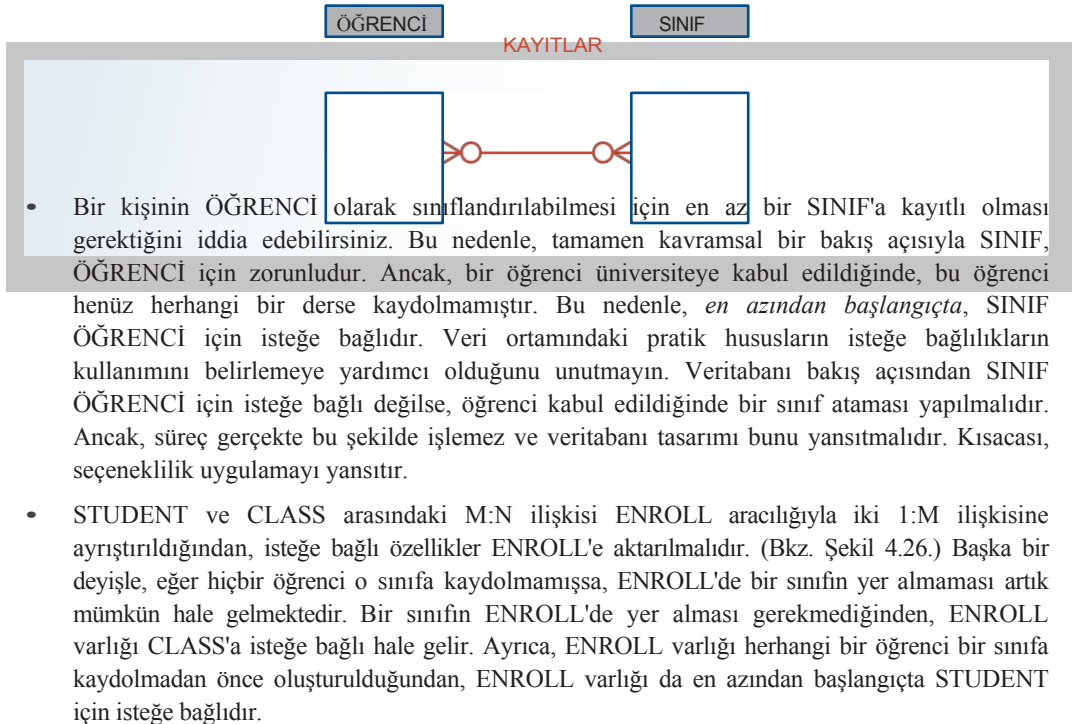
Tablo adı: SINIF					
CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWTF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWTF 8:00-8:50 a.m.	KLR200	114

Son olarak, ENROLL tablosunun anahtarının (CLASS_CODE ve STU_NUM) tamamen CLASS ve STUDENT tablolarının birincil anahtarlarından oluştuğunu unutmayın. Bu nedenle, ENROLL tablosunun anahtar niteliklerinde hiçbir boş giriş mümkün değildir.

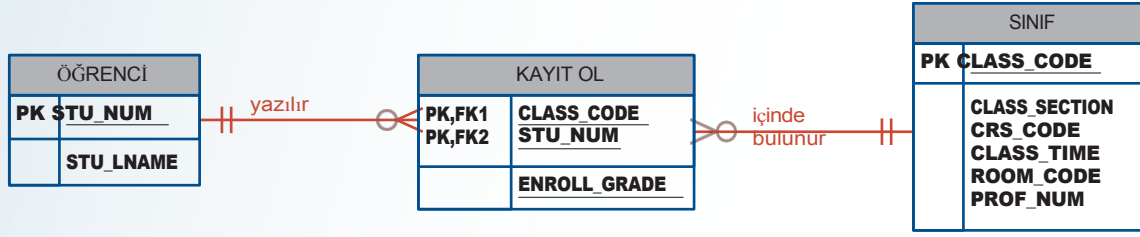
Şekil 4.24'te gösterilen küçük veritabanını uygulamak, ilişkileri net bir şekilde tanımlamanızı gerektirir. Özellikle, her bir ilişkinin "1" ve "M" taraflarını bilmeniz ve ilişkilerin zorunlu mu yoksa isteğe bağlı mı olduğunu bilmeniz gerekir. Örneğin, aşağıdaki noktalara dikkat edin:

- Bir sınıf, hiç öğrenci içermese bile (en azından kayıt başlangıcında) mevcut olabilir. Bu nedenle, Şekil 4.25'te, ÖĞRENCİ ve SINIF arasındaki M:N ilişkisinin ÖĞRENCİ tarafında isteğe bağlı bir sembol görünmelidir.

Şekil 4.25 ÖĞRENCİ ve SINIF arasındaki M:N İlişkisi



Şekil 4.26 ERD'de Bileşik Varlık



- Öğrenciler derslerine kaydolmaya başladıklarında ENROLL varlığına girileceklerdir. Doğal olarak, bir öğrenci birden fazla ders alırsa, ENROLL'de birden fazla kez yer alacaktır. Örneğin, Şekil 4.24'teki ENROLL tablosunda STU_NUM = 321452'nin üç kez geçtiğine dikkat edin. Öte yandan, her öğrenci STUDENT varlığında yalnızca bir kez yer alır. (Şekil 4.24'teki STUDENT tablosunda yalnızca bir STU_NUM = 321452 girişi). Bu nedenle, Şekil 4.26'da ÖĞRENCİ ve KAYIT arasındaki ilişki 1:M olarak gösterilmiştir ve "M" KAYIT tarafında yer almaktadır.
- Şekil 4.24'te görebileceğiniz gibi, bir sınıf ENROLL tablosunda birden fazla kez geçebilir. Örneğin, CLASS_CODE= 10014 iki kez geçer. Ancak CLASS_CODE= 10014, CLASS ve ENROLL arasındaki ilişkinin 1:M olduğunu yansıtmak için CLASS tablosunda yalnızca bir kez yer alır. Şekil 4.26'da "M" 'nin ENROLL tarafında, "1" 'in ise CLASS tarafında yer aldığına dikkat edin.

Genel bir kural olarak, bileşik bir varlığın anahtarı, üst varlıkların anahtarlarından oluşan bileşik anahtar olacaktır. Ancak, bunun bir anahtar olarak yetersiz olduğu durumlar olabilir. Bu durum, *ilişki zaman içinde tekrarlanabildiğinde* ortaya çıkar. Üçlü ilişkiyi uygulamak için Şekil 4.17'deki PRESCRIPTION tablosunda gösterilen bileşik varlığı düşünün. Bir doktorun aynı hastaya aynı ilacı birden fazla kez yazması tamamen mümkündür. Belki de ilaç, göz enfeksiyonu geçiren ve altı ay sonra kulak enfeksiyonu geçiren bir hastaya reçete edilen bir antibiyotiktir. İlişki zaman içinde tekrarlanabildiğinde, bileşik anahtarın bir parçası olarak bir zaman unsuru eklenir. Zaman ögesi, iş gereksinimlerine uygun olarak bir tarih, günün saati veya hatta bir sıralama (örneğin, birinci, ikinci, üçüncü) olabilir. Örneğin, Şekil 4.17'de, PRESCRIPTION tablosunun anahtarı dört yönlü bir bileşik olarak doktor kimliği, hasta numarası, ilaç kodu ve reçete tarihi olacaktır.

4-2 ER Diyagramı Geliştirme

yinelemeli süreç

Adımların ve prosedürlerin tekrarına dayalı bir süreç.

Veritabanı tasarım süreci doğrusal veya sıralı bir süreçten ziyade yinelemelidir. *Yinelemek* fiili "tekrar veya tekrar yapmak" anlamına gelir. Dolayısıyla, yinelemeli bir süreç, süreçlerin ve prosedürlerin tekrarlanmasına dayanır. Bir ERD oluşturmak genellikle aşağıdaki faaliyetleri içerir:

- Kuruluşun faaliyet tanımına ilişkin ayrıntılı bir anlatı oluşturun.
- İşlemlerin tanımına dayalı olarak iş kurallarını belirleyin.
- İş kurallarından ana varlıkları ve ilişkileri tanımlayın.
- İlk ERD'yi geliştirin.
- Varlıkları yeterli şekilde tanımlayan öznitelikleri ve birincil anahtarları belirleyin.
- ERD'yi revize edin ve gözden geçirin.

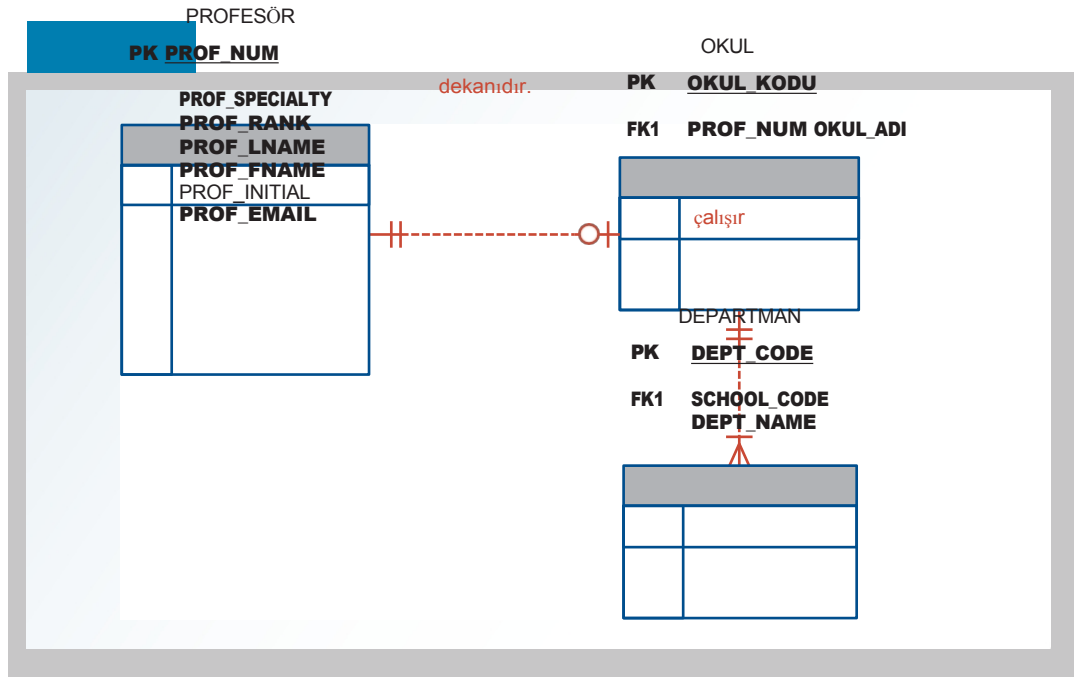
Gözden geçirme süreci sırasında muhtemelen ilave nesneler, nitelikler ve ilişkiler ortaya çıkacaktır. Bu nedenle, temel ERM yeni keşfedilen ER bileşenlerini içerecek şekilde değiştirilecektir. Daha sonra, başka bir gözden geçirme turu ek bileşenler veya mevcut diyagramın netleştirilmesini sağlayabilir. Bu süreç, son kullanıcılar ve tasarımcılar ERD'nin kurumun faaliyetlerini ve işlevlerini adil bir şekilde temsil ettiği konusunda hemfikir olana kadar tekrarlanır.

Tasarım sürecinde, veritabanı tasarımcısı varlıkları, nitelikleri ve ilişkileri tanımlamaya yardımcı olması için yalnızca görüşmelere bağlı kalmaz. Bir kuruluşun günlük faaliyetlerinde kullandığı iş formları ve raporlar incelenerek şaşırtıcı miktarda bilgi toplanabilir.

Nihayetinde uygulanabilir bir ERD ortaya çıkaran yinelemeli sürecin kullanımını göstermek için Tiny College yöneticileriyle yapılan ilk görüşmeyle başlayın. Görüşme süreci aşağıdaki iş kurallarını ortaya çıkarır:

1. Tiny College (TC) çeşitli okullara ayrılmıştır: işletme, sanat ve bilimler, eğitim ve uygulamalı bilimler. Her okul, profesör olan bir dekan tarafından yönetilir. Her profesör yalnızca bir okulun dekanı olabilir ve bir profesörün herhangi bir okulun dekanı olması gerekmez. Bu nedenle PROFESÖR ve OKUL arasında 1:1 ilişki vardır. Kardinalitenin PROFESÖR varlığının yanına (1,1) ve OKUL varlığının yanına (0,1) yazılarak ifade edilebileceğini unutmayın.
2. Her okul çeşitli bölümlerden oluşmaktadır. Örneğin, işletme okulunda bir muhasebe bölümü, bir yönetim/pazarlama bölümü, bir ekonomi/finans bölümü ve bir bilgisayar bilgi sistemleri bölümü bulunmaktadır. Kardinalite kurallarına tekrar dikkat ediniz: Bir okul tarafından işletilen en küçük departman sayısı birdir ve en büyük departman sayısı belirsizdir (N). Öte yandan, her bölüm sadece tek bir okula aittir; dolayısıyla kardinalite (1,1) ile ifade edilir. Yani, bir bölümün ait olduğu minimum okul sayısı da maksimum sayı gibi birdir. Şekil 4.27 bu ilk iki iş kuralını göstermektedir.
3. Her bölüm kurslar sunabilir. Örneğin, yönetim/pazarlama bölümü Giriş, Pazarlama İlkeleri ve Üretim Yönetimi gibi dersler sunmaktadır. Bu koşul için ERD segmenti Şekil 4.28'de gösterilmektedir. Bu ilişkinin Tiny College'ın çalışma şekline dayandığını unutmayın. Örneğin, Tiny College'ın "yalnızca araştırma" olarak sınıflandırılan bazı bölümleri olsaydı, bunlar kurs sunmazlardı; bu nedenle, KURS varlığı, DEPARTMENT isteğe bağlı olurdu.

İlk Küçük Kolej ERD Segmenti

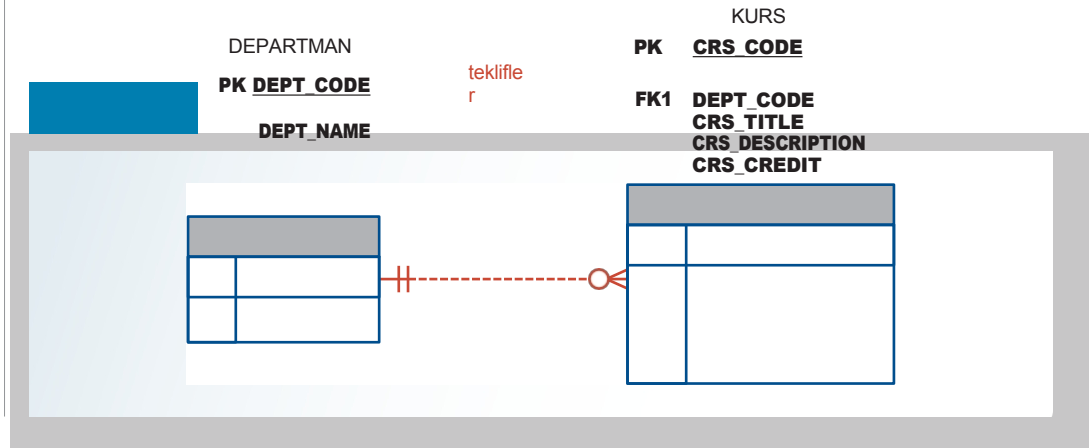


Not

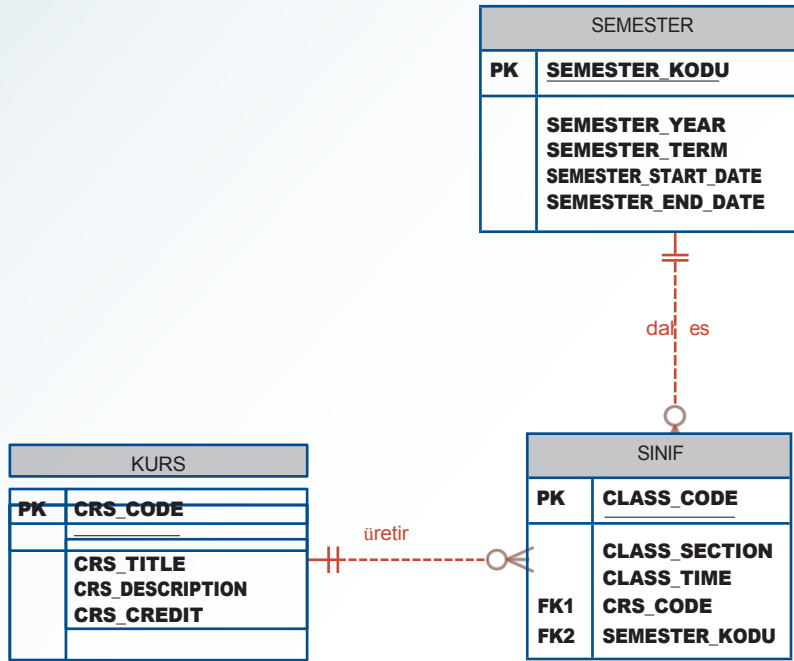
"PROFESÖR OKULUN dekanıdır" ilişkisinde PROFESÖR ve OKUL arasındaki 1:1 ilişkisinin sürdürülmesinin nedenini değerlendirmek yine yerinde olacaktır. Tekrar etmekte fayda var ki 1:1 ilişkilerin varlığı genellikle varlıkların yanlış tanımlandığına işaret eder. Bu durumda, dekanın nitelikleri OKUL varlığında saklanarak 1:1 ilişkisi kolayca ortadan kaldırılabılır. Bu çözüm aynı zamanda "Dekan kimdir?" ve "Dekanın kimlik bilgileri nelerdir?" sorgularının yanıtlanmasını da kolaylaştıracaktır. Bu çözümün dezavantajı, PROFESÖR tablosunda zaten saklanan verilerin tekrarlanması gerektirmesi ve böylece anomalilere zemin hazırlamasıdır. Ancak, her okul tek bir dekan tarafından yönetildiğinden, veri çoğaltma sorunu oldukça önemsizdir. Bir yaklaşımın diğerine tercih edilmesi genellikle bilgi gereksinimlerine, işlem hızına ve veritabanı tasarımcısının profesyonel kararına bağlıdır. Kısacası, 1:1 ilişkilerini hafife almanın ve veritabanı tasarımındaki her 1:1 ilişkisinin savunulabilir olduğundan emin olun.

4. KURS ve SINIF arasındaki ilişki Şekil 4.10'da gösterilmiştir. Bununla birlikte, bir SINIF'ın bir KURS'un bir bölümü olduğunu tekrarlamakta fayda vardır. Yani, bir departman aynı veritabanı dersinin çeşitli bölümlerini (sınıflarını) sunabilir. Bu sınıfların her biri belirli bir zamanda belirli bir yerde bir profesör tarafından verilir. Kısacası, DERS ve SINIF arasında 1:M ilişkisi vardır. Ayrıca, her sınıf belirli bir dönem boyunca sunulur. SEMESTER, sınıfın sunulacağı yılı ve dönemi tanımlar. Bunun, öğrencinin bir sınıfa fiilen kaydolduğu tarihten farklı olduğunu . Örneğin, öğrenciler yaz ve güz dönemi derslerine bahar döneminin sonuna doğru kayıt yaptırabilirler. Tiny College takviminin, sömestr sınıf programı oluşturulmadan önce sömestr başlangıç ve bitiş tarihleriyle ayarlanmış olması mümkündür, bu nedenle SINIF, SEMESTER'e isteğe bağlıdır. Bu tasarım raporlama amaçlarına da yardımcı olacaktır; örneğin, şu gibi soruları yanıtlayabilirsiniz: X döneminde hangi dersler sunuldu? Ya da Y öğrencisi X döneminde hangi dersleri aldı? Bir ders, belirli bir dönemde sınıf olarak sunulmasa bile Tiny College'ın kurs kataloğunda bulunabileceğinden, SINIF, KURS'a isteğe bağlıdır. Bu nedenle, SEMESTER, KURS ve SINIF arasındaki ilişkiler Şekil 4.29'daki gibidir.
5. Her bölüm bir veya daha fazla profesör görevlendirilmelidir. Bu profesörlerden biri ve sadece biri bölüme başkanlık eder ve hiçbir profesörün pozisyonunu kabul etmesi gerekmez. Bu nedenle, DEPARTMENT "başkanlar" ilişkisinde PROFESÖR için isteğe bağlıdır. Bu ilişkiler Şekil 4.30'da gösterilen ER segmentinde özetlenmiştir.
6. Her profesör dört sınıfa kadar ders verebilir; her sınıf bir dersin bir bölümüdür. Bir profesör aynı zamanda araştırma sözleşmeli olabilir ve hiç ders vermeyebilir. Şekil 4.31'deki ERD segmenti bu koşulları göstermektedir.

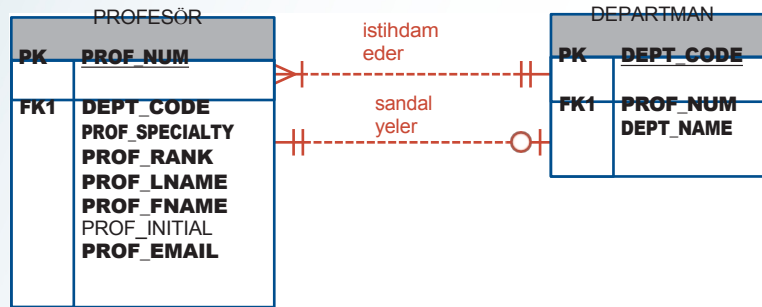
İkinci Küçük Kolej ERD Segmenti



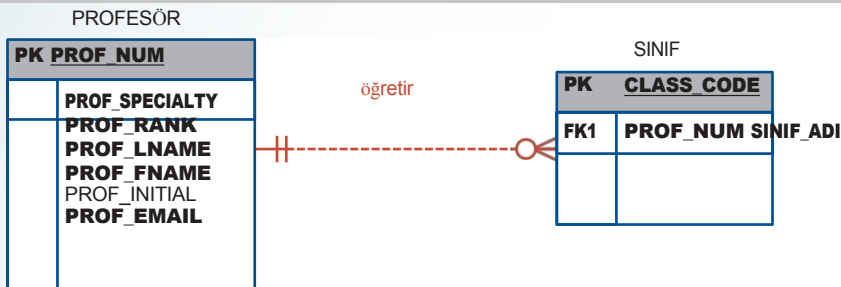
Şekil 4.29 Üçüncü Küçük Kolej ERD Segmenti



Şekil 4.30 Dördüncü Küçük Kolej ERD Segmenti

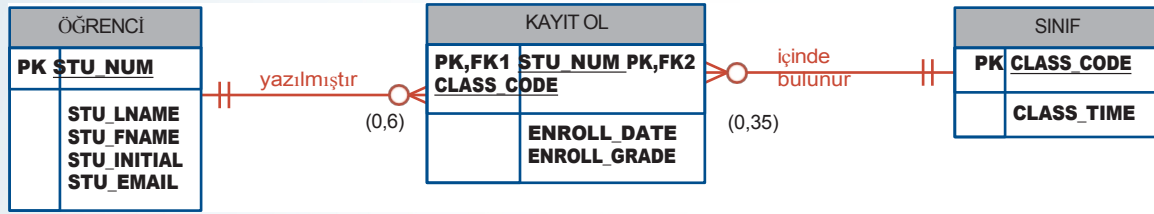


Beşinci Küçük Kolej ERD Segmenti



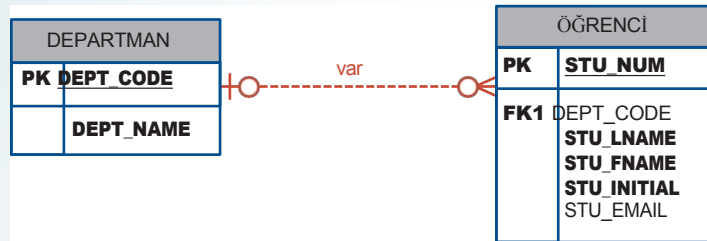
7. Bir öğrenci birden fazla sınıfa kayıt yaptırılabilir ancak her bir sınıfı herhangi bir kayıt döneminde yalnızca bir kez alabilir. Örneğin, mevcut kayıt döneminde bir öğrenci beş ders almaya karar verebilir - İstatistik, Muhasebe, İngilizce, Veritabanı ve Tarih - ancak bu öğrenci kayıt dönemi boyunca aynı İstatistik dersine beş kez kaydolmayacaktır! Her öğrenci en fazla altı sınıfa kaydolabilir ve her sınıfın en fazla 35 öğrencisi olabilir, böylece ÖĞRENCİ ve SINIF arasında bir M:N ilişkisi yaratılır. Bir SINIF, kayıt döneminin başında hiçbir öğrenci kaydolmamış olsa bile başlangıçta var olabileceğinden, ÖĞRENCİ, M:N ilişkisinde SINIF'a isteğe bağlıdır. Bu M:N ilişkisi, Şekil 4.32'deki ERD segmentinde gösterilen ENROLL varlığı kullanılarak iki 1:M ilişkisine bölünmelidir. Ancak, ENROLL ögesinin yanında isteğe bağlı sembolünün gösterildiğine dikkat edin. Bir sınıf mevcutsa ancak kayıtlı öğrencisi yoksa, bu sınıf ENROLL tablosunda yer almaz. ENROLL varlığının zayıf olduğuna da dikkat edin: varlığa bağlıdır ve (bileşik) PK'sı STUDENT ve CLASS varlıklarının PK'larından oluşur. Şekil 4.32'de gösterildiği gibi, iş kuralı kısıtlamalarını yansıtmak için ENROLL varlığının yanına (0,6) ve (0,35) kardinalitelerini ekleyebilirsiniz.

Şekil 4.32 Altıncı Minik Kolej ERD Segmenti



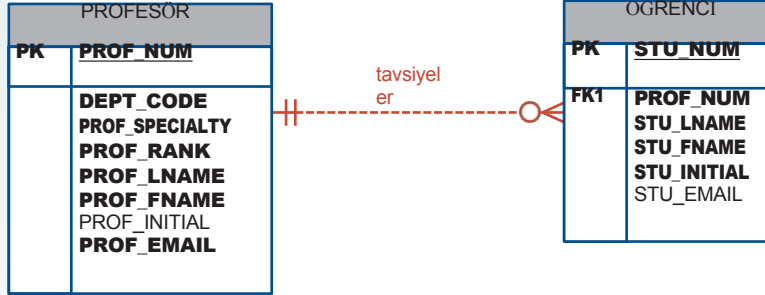
8. Her bölümün, ana dalı o bölüm tarafından sunulan birkaç (veya birçok) öğrencisi vardır. Ancak, her öğrencinin yalnızca tek bir ana dalı vardır ve bu nedenle tek bir bölümlle ilişkilendirilir. (Bkz. Şekil 4.33.) Bununla birlikte, Tiny College ortamında, bir öğrencinin ana dalını beyan -en azından bir süreliğine- mümkündür. Böyle bir öğrenci bir departmanla ilişkilendirilmeyecektir; bu nedenle DEPARTMENT, STUDENT için isteğe bağlıdır. Varlıklar arasındaki ilişkilerin ve varlıkların kendilerinin kurumun çalışma ortamını yansıttığını tekrarlamakta fayda vardır. , iş kuralları ERD bileşenlerini tanımlar.

Şekil 4.33 Yedinci Minik Kolej ERD Segmenti



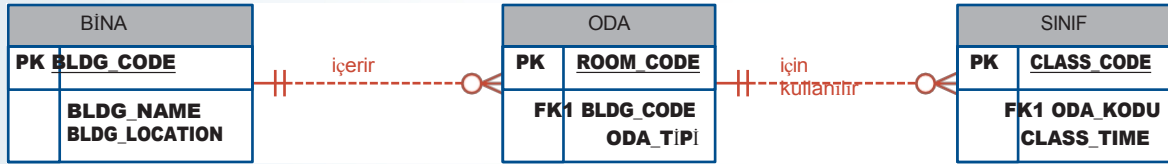
9. Her öğrencinin kendi bölümünde bir danışmanı vardır; her danışman birkaç danışmanlık yapar. Bir danışman aynı zamanda bir profesördür, ancak tüm profesörler öğrencilere danışmanlık yapmaz. Bu nedenle, ÖĞRENCİ, "PROFESÖR ÖĞRENCİYE danışmanlık yapar" ilişkisinde PROFESÖRE isteğe bağlıdır. (Bkz. Şekil 4.34.)

Şekil 4.34 Sekizinci Minik Kolej ERD Segmenti



10. Şekil 4.35'te görebileceğiniz gibi, CLASS varlığı bir ROOM_CODE niteliği içerir. Adlandırma kuralları göz önüne alındığında, ROOM_CODE'un başka bir varlığın FK'sı olduğu açıktır. Açıkçası, bir sınıf bir odada öğretildiği için, CLASS'taki ROOM_CODE'un ROOM adlı bir varlığın FK'sı olduğunu varsaymak mantıklıdır. Buna karşılık, her oda bir binada yer almaktadır. Böylece, son Tiny College ERD'si, bir BİNA'nın birçok ODA içerebileceği, ancak her ODA'nın tek bir BİNA'da bulunduğu gözlemlenerek oluşturulur. Bu ERD segmentinde, bazı binaların (sınıf) odaları içermediği açıktır. Örneğin, bir depo binası hiç adlandırılmış oda içermeyebilir.

Şekil 4.35 Dokuzuncu Minik Kolej ERD Segmenti



Yukarıdaki özeti kullanarak aşağıdaki varlıkları tanımlayabilirsiniz:

PROFESÖR	OKUL	DEPARTMAN
KURS	SINIF	SEMESTER
ÖĞRENCİ	BİNA	ODA
ENROLL (STUDENT ve CLASS arasındaki ilişkisel varlık)		

İlgili varlıkları keşfettikten sonra, ilk ilişki kümesini tanımlayabilirsiniz. Daha sonra, varlık niteliklerini tanımlarsınız. Varlıkların niteliklerinin tanımlanması, varlıklar arasındaki ilişkileri daha iyi anlamana yardımcı olur. Tablo 4.4 ERM'nin bileşenlerini özetler ve varlıklar ile ilişkilerini adlandırır.

Tablo 4.4 ERM'nin Bileşenleri

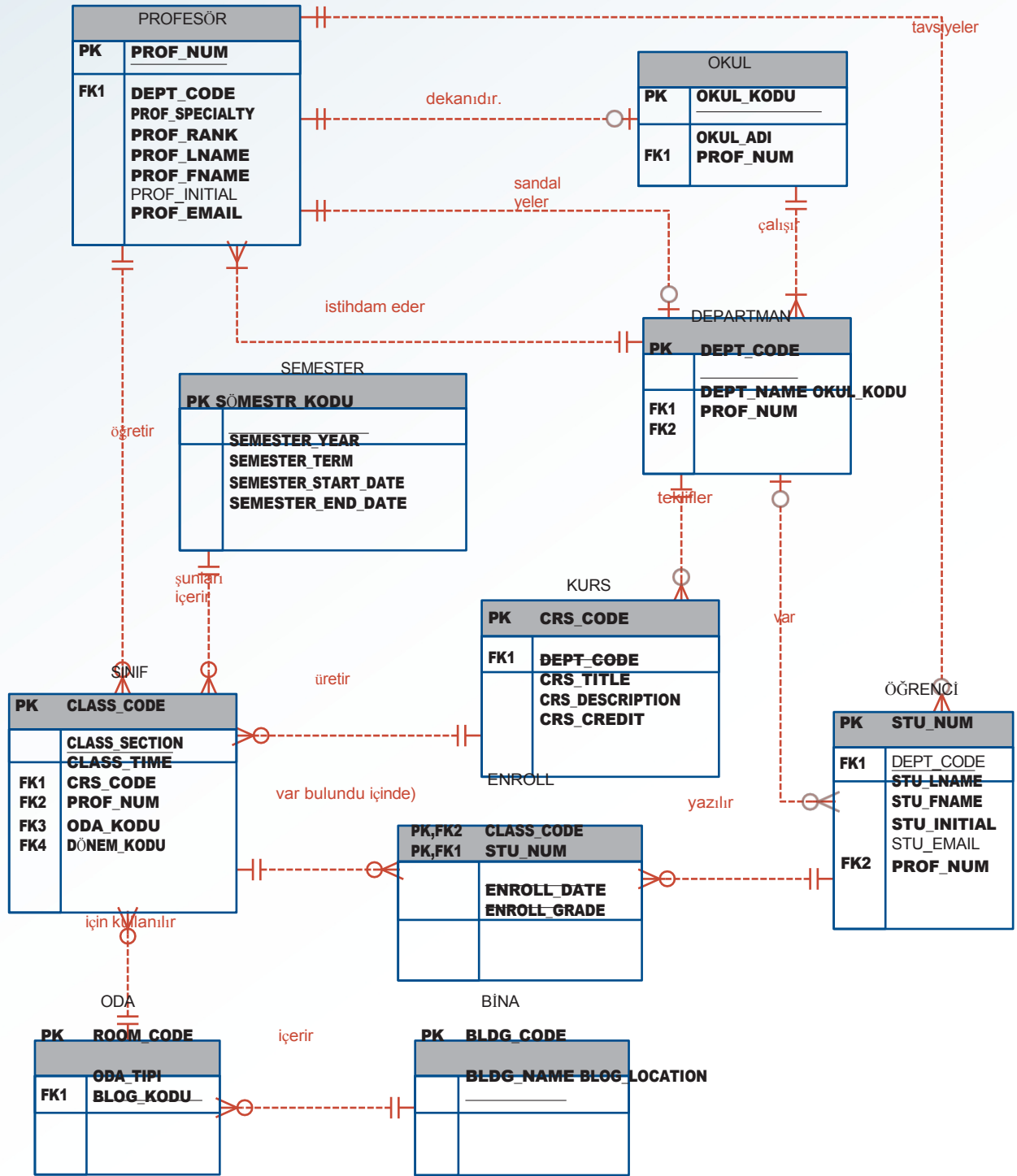
Varlık	İlişki	Bağlanabilirlik	Varlık
OKUL	çalışır	1:M	DEPARTMAN
DEPARTMAN	var	1:M	ÖĞRENCİ
DEPARTMAN	istihdam eder	1:M	PROFESÖR
DEPARTMAN	teklifler	1:M	KURS
KURS	üretir	1:M	SINIF
SEMESTER	şunları içerir	1:M	SINIF
PROFESÖR	dekanıdır.	1:1	OKUL
PROFESÖR	sandalyeler	1:1	DEPARTMAN
PROFESÖR	öğretir	1:M	SINIF
PROFESÖR	tavsiyeler	1:M	ÖĞRENCİ
ÖĞRENCİ	kaydolur	M:N	SINIF
BİNA	içerir	1:M	ODA
ODA	için kullanılır	1:M	SINIF

Not: ENROLL, "ÖĞRENCİ SINIFa kaydolar" M:N ilişkisini uygulayan bileşik varlıktır.

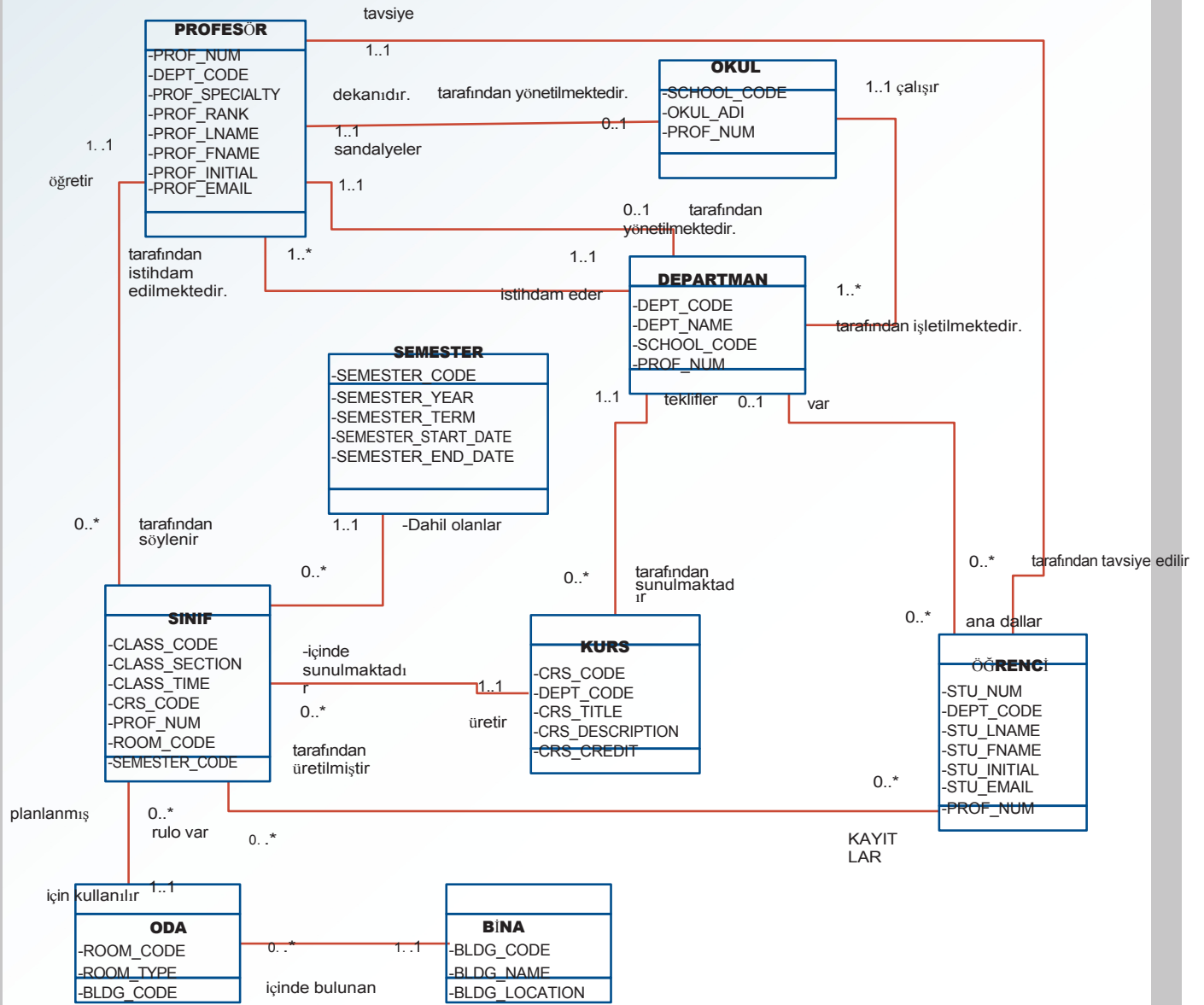
İş kurallarına dayalı olarak yeni keşfedilen ilişkiler için bağlantı ve kardinaliteyi de tanımlamanız gerekir. Ancak, diyagramın kalabalıklaşmasını önlemek için kardinaliteler gösterilmemiştir. Şekil 4.36'da Tiny College için Crow's Foot ERD gösterilmektedir. Bunun uygulamaya hazır bir model olduğuna, dolayısıyla ENROLL bileşik varlığını gösterdiğine dikkat edin.

Diyagramı geliştirmek için Crows Foot notasyonuna odaklanmamıza rağmen, bu bölümün başında da belirtildiği gibi, UML notasyonu kavramsal ve uygulama modellemesi için de popülerdir. Şekil 4.37, Tiny College için kavramsal UML sınıf diyagramını göstermektedir. Bu sınıf diyagramının STUDENT ve CLASS arasındaki M:N ilişkisini gösterdiğine dikkat ediniz. Şekil 4.38, Tiny College için uygulamaya hazır UML sınıf diyagramını göstermektedir (ENROLL bileşik varlığının bu sınıf diyagramında gösterildiğine dikkat edin). İyi bir gözlemciyseniz, Şekil 4.37 ve 4.38'deki UML sınıf diyagramlarının varlık ve öznitelik adlarını gösterdiğini ancak birincil anahtar özniteliklerini tanımlamadığını da fark edeceksiniz. Bunun nedeni UML'nin köklerine dayanmaktadır. UML sınıf diyagramları nesne yönelimli bir modelleme dilidir ve bu nedenle çoğunlukla ilişkisel dünyada bulunan "birincil veya yabancı anahtarlar" kavramını desteklemez. Bunun yerine, nesne yönelimli dünyada nesneler, oluşturulma zamanında benzersiz bir nesne tanımlayıcısını miras alır. Daha fazla bilgi için bakınız Ek G, Nesne Yönelimli Veritabanları.

Şekil 4.36 Tamamlanmış Küçük Kolej ERD



Şekil 4.37 Tiny College için Kavramsal UML Sınıf Diyagramı

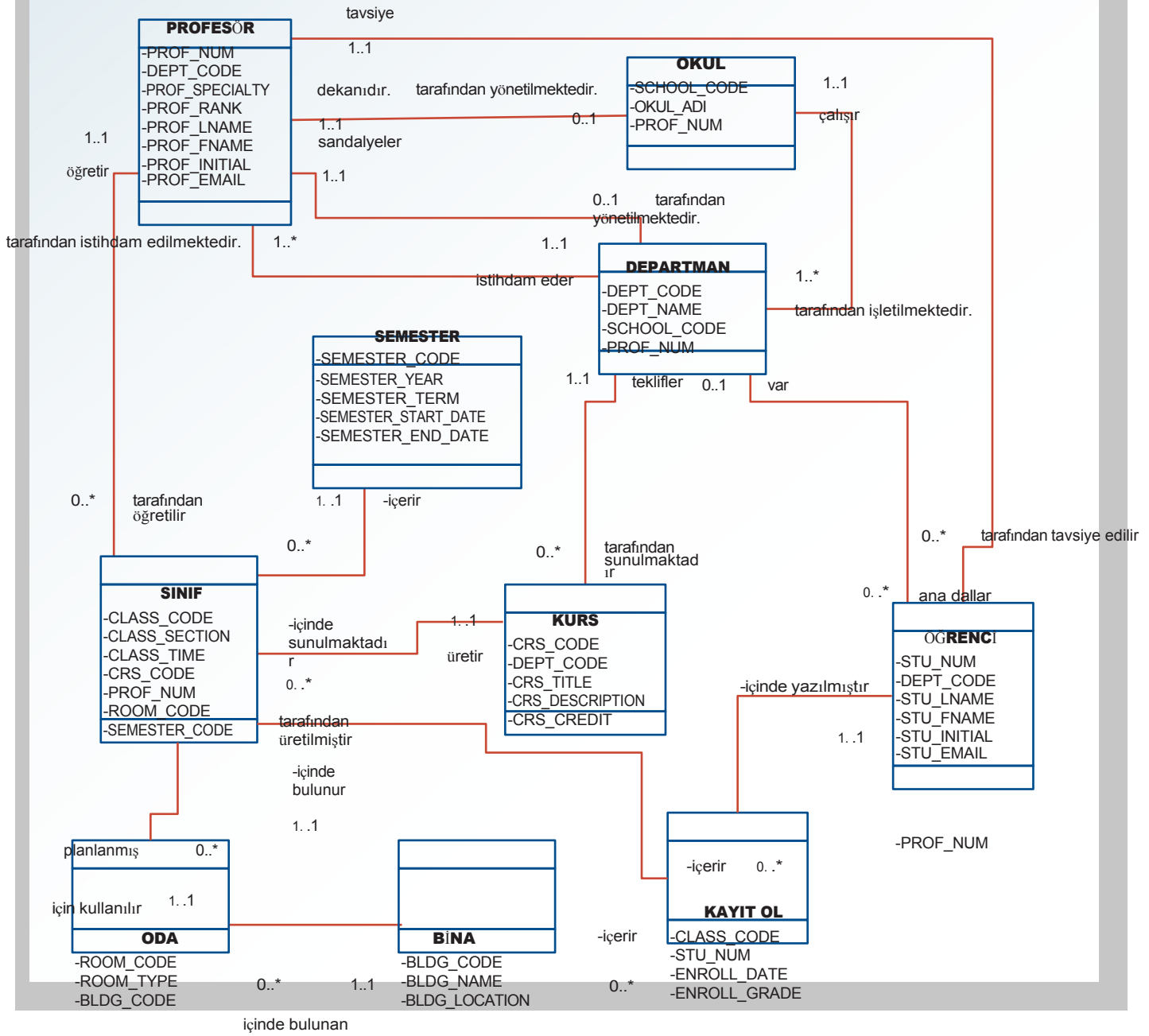


4-3 Veritabanı Tasarım Zorlukları: Çelişen Hedefler

Veritabanı tasarımcıları genellikle tasarım standartlarına bağlılık (tasarım zarafeti), işlem hızı ve bilgi gereksinimleri gibi birbirleriyle çelişen hedefler tarafından tetiklenen tasarım ödünleri vermek zorundadır.

- *Tasarım standartları.* Veritabanı tasarımı tasarım standartlarına uygun olmalıdır. Bu standartlar, veri fazlalıklarını en aza indiren mantıksal yapılar geliştirmenizde size rehberlik eder ve böylece yıkıcı veri anormalliklerinin ortaya çıkma olasılığını en aza indirir. Ayrıca, standartların mümkün olan en büyük ölçüde boşluklardan kaçınmayı nasıl öğördüğünü de öğrendiniz. Aslında, tasarım standartlarının veri tabanı tasarımı içindeki tüm bileşenlerin sunumunu yönettiğini öğrendiniz. Kısacası, tasarım standartları iyi tanımlanmış bileşenlerle çalışmanızı sağlar

Şekil 4.38 Tiny College için Uygulamaya Hazır UML Sınıf Diyagramı



ve bu bileşenlerin etkileşimini belli bir hassasiyetle değerlendirmek. Tasarım standartları olmadan, uygun bir tasarım sürecini formüle etmek, mevcut bir tasarımı değerlendirmek veya tasarımdaki değişikliklerin olası mantıksal etkilerini izlemek neredeyse imkansızdır.

- **İşlem hızı.** Birçok kuruluşta, özellikle de çok sayıda işlem gerçekleştiren kuruluşlarda, veritabanı tasarımında yüksek işlem hızları genellikle en önemli önceliktir. Yüksek işlem hızı, minimum erişim süresi anlamına gelir ve bu da mantıksal olarak istenen ilişkilerin sayısını ve karmaşıklığını en aza indirerek elde edilebilir. Örneğin, "mükemmel" bir tasarım, boşluklardan kaçınmak için 1:1 ilişki kullanabilirken, daha yüksek işlem hızını vurgulayan bir tasarım, ek bir ilişkinin kullanılmasını önlemek için iki tabloyu birleştirebilir ve boşluklardan kaçınmak için kukla girişler kullanabilir. Odak noktası veri alma hızıysa, tasarıma türetilmiş nitelikleri de dahil etmek zorunda kalabilirsiniz.