Şekil 2.7'de gösterilen ER'lere dikkat edin:

- Bir PROFESÖR çok sayıda SINIF öğretebilir ve her SINIF sadece bir PROFESÖR tarafından öğretilir; PROFESÖR ve SINIF arasında 1:M ilişkisi vardır.
- Bir SINIF birçok öğrenciyi KAYDEDEBİLİR ve her ÖĞRENCİ birçok SINIFA KAYDOLABİLİR, böylece ÖĞRENCİ ve SINIF arasında bir M:N ilişkisi yaratır. (ENROLL varlığının kesin doğası hakkında Bölüm 4'te bilgi edineceksiniz).
- Her KURS birçok SINIF oluşturabilir, ancak her SINIF tek bir KURS'a referans verir. Örneğin, bir veritabanı kursunun CIS-420 kurs koduna sahip birkaç sınıfı (bölümü) olabilir. Bu sınıflardan biri MWF'de sabah 8:00 ile 8:50 arasında, diğeri MWF'de öğleden sonra 1:00 ile 1:50 arasında, üçüncüsü ise Perşembe günleri 6:00 ile 8:40 arasında veriliyor olabilir, ancak her üç sınıf da CIS-420 kurs koduna sahiptir.
- Son olarak, bir SINIF bir ODA gerektirir, ancak bir ODA birçok SINIF için planlanabilir. , her bir sınıf birden fazla ders için kullanılabilir: örneğin biri sabah 9:00'da, biri sabah 11:00'de ve biri de öğleden sonra 1:00'de. Başka bir deyişle, ODA ve SINIF arasında 1:M ilişkisi vardır.

Veritabanının alt kümelerini temsil eden harici görünümlerin kullanımının bazı önemli avantajları vardır:

- Her bir iş biriminin faaliyetlerini desteklemek için gereken belirli verileri tanımlamak kolaydır.
- Modelin yeterliliği hakkında geri bildirim sağlayarak tasarımcının işini kolaylaştırır. Özellikle model, harici modelleri tarafından tanımlanan tüm süreçlerin yanı sıra tüm operasyonel gereksinimleri ve kısıtlamaları desteklediğinden emin olmak için kontrol edilebilir.
- Veritabanı tasarımında *güvenlik* kısıtlamalarının sağlanmasına yardımcı olur. Her bir iş birimi yalnızca bir veri alt kümesiyle çalıştığında tüm bir veritabanına zarar vermek daha zordur.
- Uygulama programı geliştirmeyi çok daha basit hale getirir.

kavramsal model
Kavramsal tasarım sürecinin çıktısı. Kavramsal model şunları sağlar tüm bir veritabanının genel bir görünümünü sunar ve ayrıntılardan kaçınarak ana veri nesnelerini açıklar.

kavramsal şema
Kavramsal modelin genellikle grafiksel olarak ifade edilen bir temsili. Ayrıca bkz. *kavramsal model*.

yazılım bağımsızlığı
Herhangi bir model veya uygulamanın, onu uygulamak için kullanılan yazılıma bağlı olmayan bir özelliği.

donanım bağımsızlığı Bir modelin, modelin uygulanmasında kullanılan donanıma bağlı olmadığı durum. Bu nedenle, donanımdaki değişiklikler kavramsal düzeyde veritabanı tasarımı üzerinde hiçbir etkisi olmayacaktır.

mantıksal tasarım
Tasarım aşamasında, kavramsal tasarımı seçilen VTYS' gereksinimleriyle eşleştiren bir aşama ve Bu nedenle yazılıma bağımlıdır. Mantıksal tasarım, kavramsal tasarımı yazılıma dönüştürmek için kullanılır.
DB2, SQL Server, Oracle, IMS, Informix, Access veya Ingress gibi seçilen bir veritabanı yönetim sistemi için dahili model.

2-6b Kavramsal Model

Kavramsal model, tüm kuruluş tarafından tüm veritabanının global bir görünümünü temsil eder. Yani kavramsal model, Şekil 2.8'de gösterildiği gibi, tüm dış görünümleri (varlıklar, ilişkiler, kısıtlamalar ve süreçler) kuruluştaki verilerin tek bir küresel görünümüne entegre eder. **Kavramsal şema** olarak da bilinen bu model, ana veri nesnelerinin tanımlanması ve üst düzey açıklaması için temel oluşturur (veritabanı modeline özgü ayrıntılardan kaçınarak).

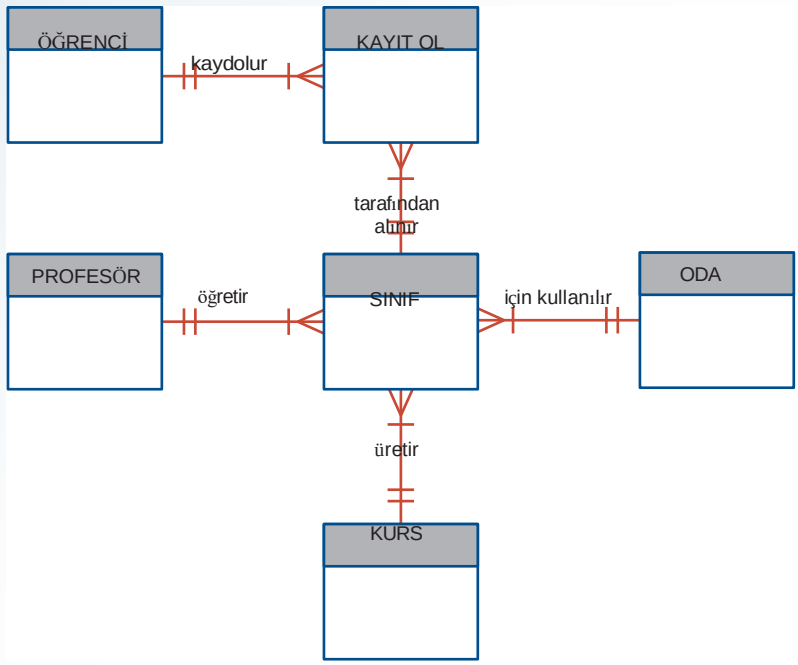
En yaygın kullanılan kavramsal model ER modelidir. ER modelinin, etkin bir şekilde temel veritabanı planı olan ERD yardımıyla gösterildiğini unutmayın. ERD, kavramsal şemayı grafiksel olarak *temsil* etmek için kullanılır.

Kavramsal model bazı önemli avantajlar sağlamaktadır. İlk olarak, veri ortamının anlaşılması nispeten kolay olan kuşbakışı (makro düzeyde) bir görünümünü sağlar. Örneğin, Şekil 2.8'deki kavramsal modeli inceleyerek Tiny College'ın veri ortamının bir özetini elde edebilirsiniz.

İkinci olarak, kavramsal model hem yazılımdan hem de donanımdan bağımsızdır. **Yazılım bağımsızlığı**, modelin, modeli uygulamak için kullanılan DBMS yazılımına bağlı olmadığı anlamına gelir. **Donanım bağımsızlığı**, modelin, modelin uygulanmasında kullanılan donanıma bağlı olmadığı anlamına gelir. Bu nedenle, donanımda ya da DBMS yazılımında yapılacak değişikliklerin kavramsal düzeydeki veritabanı tasarımı üzerinde hiçbir etkisi olmayacaktır. Genel olarak **mantıksal tasarım** terimi, herhangi bir DBMS'de uygulanabilecek kavramsal bir veri modeli oluşturma görevini ifade eder

Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir. Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.

Şekil 2.8 Küçük Kolej için Kavramsal Model



2-6c Dahili Model

Belirli bir VTYS seçildikten sonra, dahili model kavramsal modeli VTYS ile eşleştirir. **Dahili model**, veritabanının DBMS tarafından "görüldüğü" şekliyle temsilidir. Başka bir deyişle, dahili model tasarımcının kavramsal modelin özelliklerini ve kısıtlamalarını seçilen uygulama modelinin özellikleriyle eşleştirmesini gerektirir. **Dahili bir şema**, seçilen veritabanı tarafından desteklenen veritabanı yapılarını kullanarak dahili bir modelin belirli bir temsilini tasvir eder.

Bu kitap ilişkisel modele odaklandığından, dahili modeli uygulamak için ilişkisel bir veritabanı seçilmiştir. Bu nedenle, dahili şema kavramsal modeli ilişkisel model yapılarıyla eşleştirmelidir. Özellikle, kavramsal modeldeki varlıklar ilişkisel modeldeki tablolarla eşleştirilir. Aynı şekilde, ilişkisel bir veritabanı seçildiği için, iç şema ilişkisel veri tabanları için standart dil olan SQL kullanılarak ifade edilir. Şekil 2.8'de gösterilen Tiny College için kavramsal model durumunda, dahili model PROFESÖR, KURS, SINIF, ÖĞRENCİ, KAYIT ve ODA tabloları oluşturularak uygulanmıştır. Tiny College için dahili modelin basitleştirilmiş bir versiyonu Şekil 2.9'da gösterilmektedir.

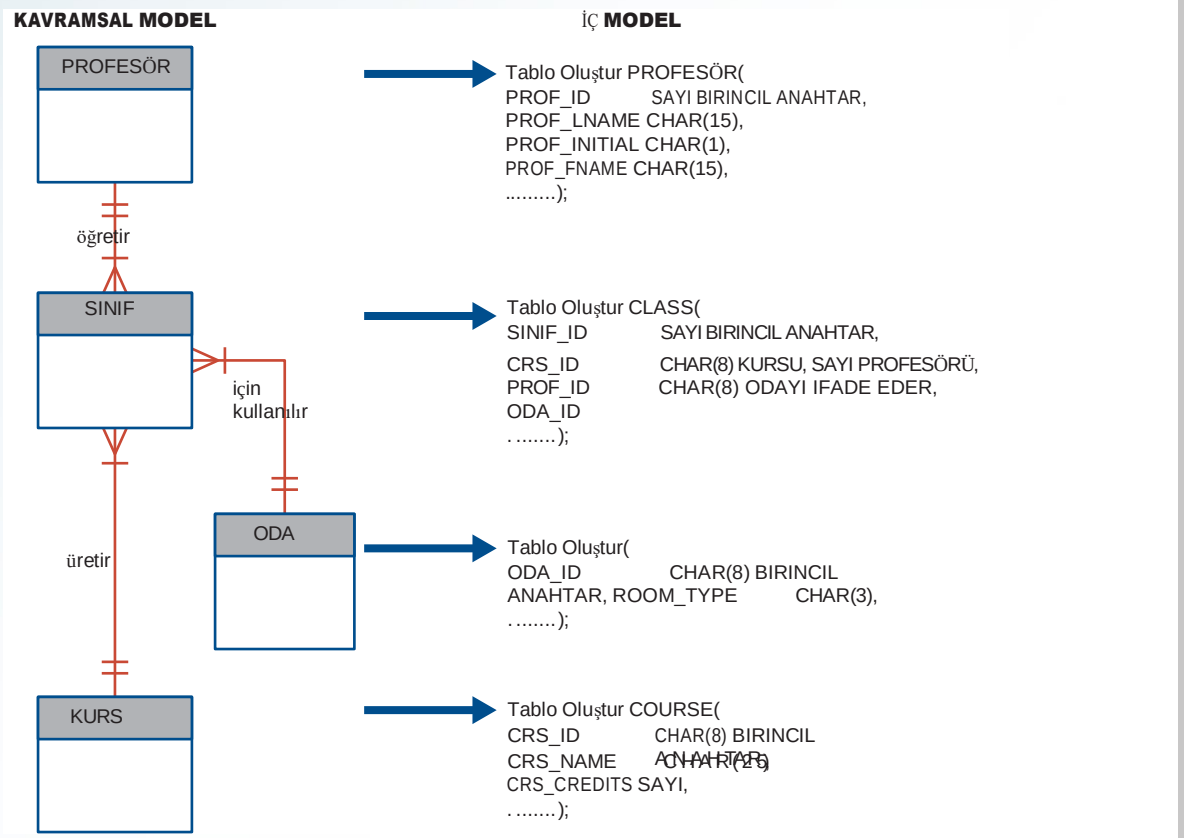
Ayrıntılı bir iç modelin geliĢtirilmesi özellikle hiyerarĢik ya da ağ modelleriyle çalıĢan veritabanı tasarımcıları için önemlidir çünkü bu modeller veri depolama konumlarının ve veri eriĢim yollarının kesin olarak belirlenmesini gerektirir. Bunun aksine, ilişkisel model iç modelinde daha az ayrıntı gerektirir çünkü çoğu RDBMS veri erişim yolu tanımını *şeffaf bir* şekilde ele alır; yani tasarımcının veri erişim yolu ayrıntılarından haberdar olması gerekmez. Bununla birlikte, ilişkisel veritabanı yazılımları bile, özellikle anabilgisayar ortamında, genellikle veri depolama konumlarının belirtilmesini gerektirir. Örneğin, DB2 veri depolama grubunu, depolama grubu içindeki veritabanının konumunu ve veritabanı içindeki tabloların konumunu belirtmenizi gerektirir.

dahili model
Veritabanı modellemesinde, Kavramsal modeli uygulama için belirli bir DBMS modeline uyarlayan bir veri soyutlama seviyesiDahili model, bir veritabanının DBMS tarafından "görüldüğü" şekliyle temsilidir. Başka bir deyişle, dahili model, tasarımcının kavramsal modelin özelliklerini ve kısıtlamalarını seçilen uygulama modelinin eşleştirmesini gerektirir.

dahili şema
Seçilen tarafından desteklenen veritabanı yapılarını kullanan dahili bir modelin temsili.

Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir. Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.

Şekil 2.9 Tiny College için Dahili Model



mantıksal bağımsızlık
Dahili modelin kavramsal modeli etkilemeden bir durum. (Dahili model donanımdan bağımsızdır çünkü etkilenmez yazılımın yüklü olduğu bilgisayar tarafından . Bu nedenle, depolama aygıtlarındaki veya işletim sistemlerindeki bir değişiklik dahili modeli etkilemeyecektir).

Dahili model belirli bir veritabanı yazılımına bağlı olduğundan, yazılım bağımlı olduğu söylenir. Bu nedenle, DBMS yazılımındaki bir değişiklik, dahili modelin uygulama veritabanı modelinin özelliklerine ve gereksinimlerine uyacak şekilde değiştirilmesini gerektirir. Kavramsal modeli etkilemeden dahili modeli değiştirebildiğinizde, **mantıksal bağımsızlığa** sahip olursunuz. Bununla birlikte, dahili model hala donanımdan bağımsızdır çünkü yazılımın yüklü olduğu bilgisayar türünden etkilenmez. Bu nedenle, depolama aygıtlarındaki bir değişiklik ya da işletim sistemlerindeki bir değişiklik dahili modeli etkilemeyecektir.

2-6d Fiziksel Model

fiziksel model
Veriler için konum, yol ve format gibi fiziksel özelliklerin tanımlandığı bir model. Fiziksel model hem donanıma hem de yazılıma bağlıdır. Ayrıca bkz. *fiziksel tasarım*.

Fiziksel model en düşük soyutlama seviyesinde çalışır ve verilerin manyetik, katı hal veya optik medya gibi depolama ortamlarına kaydedilme şeklini tanımlar. Fiziksel model, hem fiziksel depolama cihazlarının hem de bu depolama cihazlarındaki verilere ulaşmak için gereken (fiziksel) erişim yöntemlerinin tanımlanmasını gerektirir, bu da onu hem yazılım hem de donanım bağımlı hale getirir. Kullanılan depolama yapıları yazılıma (DBMS ve işletim sistemi) ve bilgisayarın kullanabileceği depolama aygıtlarının türüne bağlıdır. Fiziksel modelin tanımında gereken hassasiyet, veritabanı tasarımcılarının veritabanı tasarımını uygulamak için kullanılan donanım ve yazılım hakkında ayrıntılı bilgi sahibi olmasını gerektirir.

İlk veri modelleri, veritabanı tasarımcısını fiziksel modelin veri depolama gereksinimlerinin ayrıntılarını dikkate almaya zorlamıştır. Ancak şu anda baskın olan ilişkisel model, fiziksel düzeyden ziyade büyük ölçüde mantıksal düzeyi hedeflemektedir; bu nedenle, öncekilerde yaygın olan fiziksel düzeydeki ayrıntıları gerektirmez.

Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir. Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.

İlişkisel model, tasarımcının verilerin fiziksel depolama özellikleriyle ilgilenmesini gerektirmese de, ilişkisel bir modelin *uygulanması,* daha yüksek performans için fiziksel düzeyde ince ayar gerektirebilir. İnce ayar özellikle çok büyük veritabanları bir ana bilgisayar ortamına kurulduğunda önemlidir, ancak fiziksel düzeyde böyle bir performans ince ayarı bile fiziksel veri depolama özellikleri hakkında bilgi gerektirmez.

Daha önce de belirtildiği gibi, fiziksel model VTYS'ye, erişim yöntemlerine dosyaları ve işletim sistemi tarafından desteklenen donanım depolama aygıtı türleri. Ne zaman yapabilirsin Dahili modeli etkilemeden fiziksel modeli değiştirirseniz, **fiziksel bağımsızlığa** sahip olursunuz. Bu nedenle, depolama aygıtlarındaki veya yöntemlerindeki bir değişiklik ve hatta işletim sistemindeki bir değişiklik dahili modeli etkilemeyecektir.

fiziksel bağımsızlık
Fiziksel modelin iç modeli etkilemeden değiştirilebildiği bir durum.

Veri soyutlama seviyeleri Tablo 2.4'te özetlenmiştir.

Tablo 2.4 Veri Soyutlama Düzeyleri

Model	Soyutlama derecesi	Odaklanma	Bağımsız
Harici	Yüksek ↕ Düşük	Son kullanıcı görüşleri	Donanım ve yazılım
Kavramsal		Verilerin global görünümü (veritabanı modelinden bağımsız)	Donanım ve yazılım
Dahili		Spesifik veritabanı modeli	Donanım
Fiziksel		Depolama ve erişim yöntemleri	Ne donanım ne de yazılım

Özet

- Bir veri modeli, karmaşık bir gerçek dünya veri ortamının soyutlamasıdır. Veritabanı tasarımcıları, programcılar ve son kullanıcılarla iletişim kurmak için veri modellerini kullanır. Temel veri modelleme bileşenleri varlıklar, nitelikler, ilişkiler ve kısıtlamalardır. İş kuralları, belirli bir gerçek dünya ortamındaki temel modelleme bileşenlerini belirlemek ve tanımlamak için kullanılır.
- Hiyerarşik ve ağ veri modelleri artık kullanılmayan ilk modellerdir, ancak bazı kavramlar mevcut veri modellerinde bulunmaktadır.
- İlişkisel model mevcut veritabanı uygulama standardıdır. İlişkisel modelde, son kullanıcı verileri tablolarda depolanmış olarak algılar. Tablolar, ortak niteliklerdeki ortak değerler aracılığıyla birbirleriyle ilişkilendirilir. Varlık ilişkileri (ER) modeli, ilişkisel modeli tamamlayan veri modellemesi için popüler grafik araçtır. ER modeli, veritabanı tasarımcılarının, programcıların ve son kullanıcıların gördüğü gibi verilerin farklı görünümlerini görsel olarak sunmalarına ve verileri ortak bir çerçeveye entegre etmelerine olanak tanır.
- Nesne yönelimli veri modeli (OODM) temel modelleme yapısı olarak nesneleri kullanır. İlişkisel modelin varlığı gibi, bir nesne de olgusal içeriği ile tanımlanır. Ancak bir varlıktan farklı olarak nesne, gerçekler arasındaki ilişkilerin yanı sıra diğer nesnelerle olan ilişkiler hakkında da bilgi içerir ve böylece verilerine daha fazla anlam kazandırır.
- İlişkisel model, genişletilmiş ilişkisel veri modeli (ERDM) haline gelmek için birçok nesne yönelimli (OO) uzantıyı benimsemiştir. Nesne/ilişkisel veritabanı yönetim sistemleri (O/R DBMS) ERDM'yi uygulamak için geliştirilmiştir. Bu noktada, OODM büyük ölçüde özel mühendislik ve bilimsel uygulamalarda kullanılırken, ERDM öncelikle iş uygulamalarına yöneliktir.
- Hadoop ve NoSQL gibi Büyük Veri teknolojileri, Büyük Veri analitiği için dağıtık, hataya dayanıklı ve uygun maliyetli destek sağlamaktadır. NoSQL veritabanları, ilişkisel modeli kullanmayan ve Büyük Veri kuruluşlarının çok özel ihtiyaçlarını desteklemeye yönelik yeni bir veritabanı neslidir. NoSQL veritabanları, veri tutarlılığından ödün vererek ve ilişkileri ve veri bütünlüğünü koruma yükünü program koduna kaydırarak yüksek ölçeklenebilirlik, kullanılabilirlik ve hata toleransı sağlayan dağıtılmış veri depoları sunar.
- Veri modelleme gereksinimleri, farklı veri görünümlerinin (küresel ve yerel) ve veri soyutlama seviyesinin bir fonksiyonudur. Amerikan Ulusal Standartlar Enstitüsü Standart Planlama ve Gereksinimler Komitesi (ANSI/SPARC) üç veri soyutlama seviyesi tanımlamaktadır: harici, kavramsal ve dahili. Fiziksel seviye olarak adlandırılan dördüncü ve en düşük veri soyutlama seviyesi, yalnızca fiziksel depolama yöntemleriyle ilgilidir.

Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir. Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.

Anahtar Terimler

3 Vs
Amerikan Ulusal Standartlar Enstitüsü (ANSI)
Büyük Veri
niteliği
iş kuralı Chen
notasyon sınıfı
class diyagramı gösterimi
class diyagramı
class hiyerarşisi
istemci düğümü
kavramsal model
kavramsal şema
bağlantı kısıtlaması
Karga Ayağı notasyonu
veri tanımlama dili (DDL) veri manipülasyon dili (DML)
veri modeli
veri
modelleme
veri düğümü
varlık
varlık örneği
varlık oluşumu
varlık ilişki diyagramı (ERD)
varlık ilişki (ER) modeli (ERM)
varlık seti
genişletilmiş ilişkisel veri modeli (ERDM)
Genişletilebilir Biçimlendirme Dili (XML)
harici modeli
harici şema
Hadoop
Hadoop Dağıtılmış Dosya Sistemi (HDFS)
donanım bağımsızlığı
hiyerarşik model kalıtımı
dahili model
dahili şema
Nesnelerin İnterneti (IoT)
mantıksal tasarımı
mantıksal bağımsızlık
çoktan çoka (M:N veya *..*) ilişki
MapReduce
yöntem adı
düğüm
ağ modeli
NoSQL
nesne
nesne yönelimli veri modeli (OODM)
nesne yönelimli veritabanı yönetim sistemi (OODBMS)
nesne/ilişkisel veritabanı yönetim sistemi (O/R DBMS)
bir-çok (1:M veya 1..*) ilişkisi
bire bir (1:1 veya 1..1) ilişki fiziksel bağımsızlık
fiziksel model
ilişkisi
ilişkisel veritabanı yönetim sistemi (RDBMS)
ilişkisel diyagram
ilişkisel model
ilişki şeması
segment
anlamsal veri modeli yazılım bağımsızlığı alt şeması
tablo
tuple
Birleşik Modelleme Dili (UML)

İnceleme Soruları

1. Veri modellerinin önemini tartışabilecektir.
2. İş kuralı nedir ve veri modellemedeki amacı nedir?
3. İş kurallarını veri modeli bileşenlerine nasıl dönüştürürsünüz?
4. İlişkisel veri modelinin temel özelliklerini tanımlayabilecek ve bunların son kullanıcı ve tasarımcı için önemini tartışabilecektir.
5. Varlık ilişkileri (ER) modelinin daha yapılandırılmış bir ilişkisel veritabanı tasarım ortamının oluşturulmasına nasıl yardımcı olduğunu açıklayınız.
6. "Bir müşteri çok sayıda ödeme yapabilir, ancak her ödeme sadece bir müşteri tarafından yapılır." Bu senaryoyu bir varlık ilişki diyagramı (ERD) gösterimi için temel olarak kullanın.
7. Neden bir nesnenin bir varlıktan daha fazla anlamsal içeriğe sahip olduğu söylenir?
8. Nesne yönelimli veri modelinde (OODM) bir nesne ile bir sınıf arasındaki fark nedir?
9. Soru 6'yı bir OODM ile nasıl modellersiniz? (Kılavuz olarak Şekil 2.4'ü kullanın.)
10. ERDM nedir ve modern (üretim) veritabanı ortamında nasıl bir rol oynar?
11. İlişki nedir ve hangi üç tür vardır?

Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir. Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.

12. Üç ilişki türünün her birine bir örnek veriniz.
13. Tablo nedir ve ilişkisel modelde nasıl bir rol oynar?
14. İlişkisel diyagram nedir? Bir örnek veriniz.
15. Bağlanabilirlik nedir? (Bağlanabilirliği göstermek için bir Karga Ayağı ERD'si kullanın).
16. Büyük Veri olgusunu tanımlayın.
17. *3 Vs* terimi neyi ifade ?
18. Hadoop nedir ve temel bileşenleri nelerdir?
19. NoSQL veritabanının temel özellikleri nelerdir?
20. Hastaları ve testleri olan bir tıp kliniği örneğini kullanarak, ilişkisel modeli kullanarak bu örneğin nasıl modelleneceğine dair basit bir gösterim sununuz.
21. Mantıksal bağımsızlık nedir?
22. Fiziksel bağımsızlık nedir?

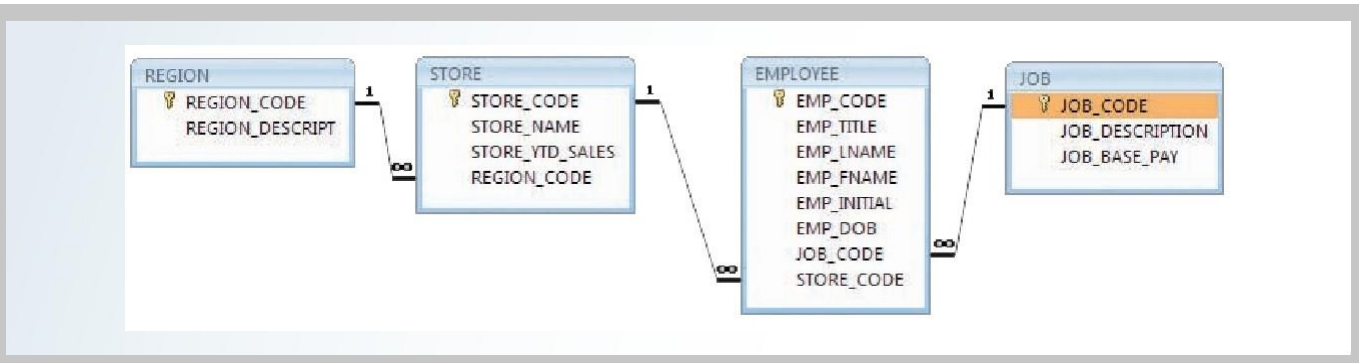
Problemler

Problem 1-3 üzerinde çalışmak için Şekil 2.1'in içeriğini kullanın.

1. ACENTE ve MÜŞTERİ arasındaki ilişkiyi yöneten iş kural(lar)ını yazın.
2. Problem 1'de yazdığınız iş kural(lar)ını göz önünde bulundurarak temel Karga Ayağı ERD'sini oluşturun.
3. Problem 2'de çizdiğiniz ERD'yi kullanarak eşdeğer nesne gösterimini ve UML sınıf diyagramını oluşturun. (Kılavuz olarak Şekil 2.4'ü kullanın.)

Şekil P2.4'ü rehber olarak kullanarak Problem 4-5'i çalışın. DealCo ilişkisel diyagramı, ülkenin iki bölgesinde bulunan DealCo mağazaları için başlangıç varlıklarını ve niteliklerini göstermektedir.

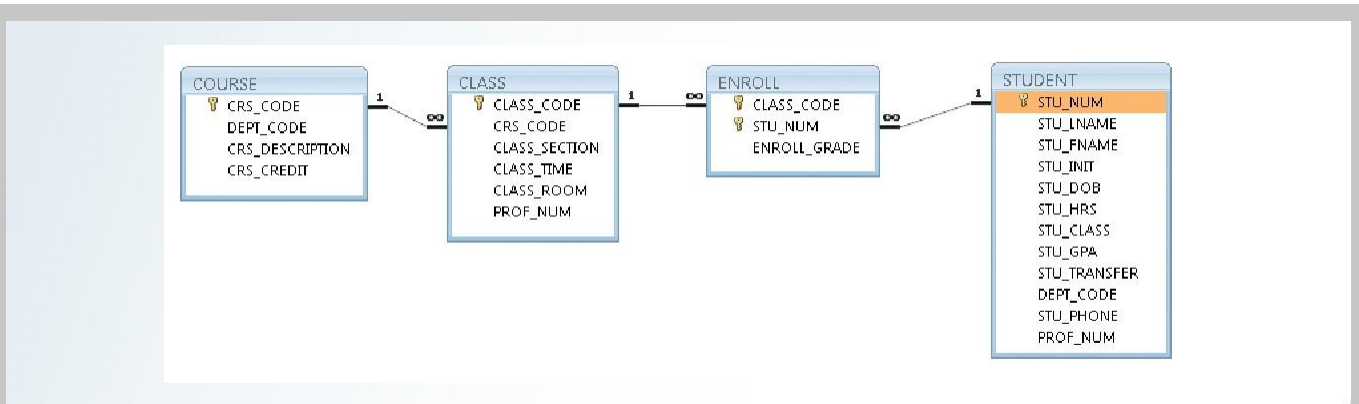
Şekil P2.4 DealCo İlişkisel Diyagramı



4. Her bir ilişki türünü tanımlayın ve tüm iş kurallarını yazın.
5. DealCo için temel Crow's Foot ERD'sini oluşturun.

Şekil P2.6'yı rehber olarak kullanarak Problem 6-8 üzerinde çalışın. Tiny College ilişkisel diyagramı, kolej için başlangıç varlıklarını ve niteliklerini göstermektedir.

Şekil P2.6 Küçük Kolej İlişkisel Diyagramı

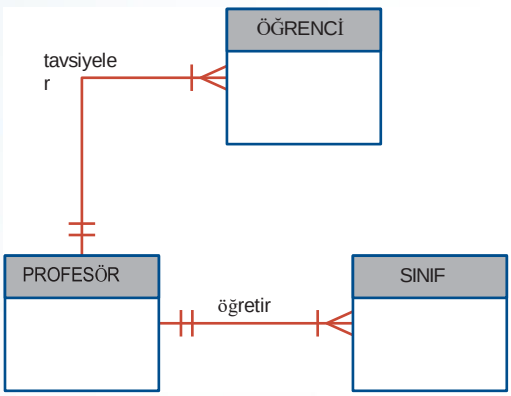


Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir.

Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.

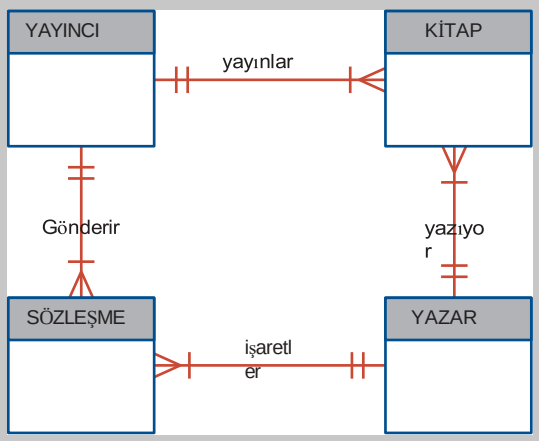
6. Her bir ilişki türünü tanımlayın ve tüm iş kurallarını yazın.
7. Tiny College için temel Karga Ayağı ERD'sini oluşturun.
8. İlişkisel diyagramda tanımladığınız varlıkları ve ilişkileri yansıtan UML sınıf diyagramını oluşturun.
9. Tipik olarak, bir hastane hastası belirli bir doktor tarafından sipariş edilen ilaçları alır. Hasta genellikle günde birkaç ilaç aldığından, HASTA ve EMİR arasında 1:M ilişkisi vardır. Benzer şekilde, her sipariş birkaç ilaç içerebilir ve bu da SİPARİŞ ile İLAÇ arasında 1:M ilişkisi yaratır.
   a. HASTA, SİPARİŞ ve için iş kurallarını tanımlayın.
   b. Bu iş kurallarını yakalamak için ilişkisel bir veritabanı modelini tasvir eden bir Crow's Foot ERD oluşturun.
10. United Broke Artists (UBA) çok ünlü olmayan sanatçılar için bir aracıdır. UBA ressamları, tabloları ve galerileri takip etmek için küçük bir veri tabanı tutmaktadır. Bir resim belirli bir sanatçı tarafından yaratılır ve daha sonra belirli bir galeride sergilenir. Bir galeride birçok resim sergilenebilir, ancak her resim yalnızca bir galeride sergilenebilir. Benzer şekilde, bir resim tek bir ressam tarafından oluşturulur, ancak her ressam birçok resim oluşturabilir. PAINTER, PAINTING ve GALLERY'yi ilişkisel bir veritabanı açısından kullanın:
   a. Hangi tabloları oluştururdunuz ve tablo bileşenleri neler ?
   b. (Bağımsız) tablolar birbirleriyle nasıl ilişkilendirilebilir?
11. Problem 10'daki ERD'yi kullanarak ilişkisel şemayı oluşturun. (Her bir varlık için uygun bir öznitelik koleksiyonu oluşturun. Öznitelikleri adlandırmak için uygun adlandırma kurallarını kullandığınızdan emin olun).
12. Problem 10'daki ERD'yi karşılık gelen bir UML sınıf diyagramına dönüştürün.
13. Şekil P2.13'te gösterilen Crow's Foot ERD'de gösterilen ilişkileri tanımlayın (iş kurallarını tanımlayın).

Şekil P2.13 Problem 13 için Karga Ayağı ERD



14. ProdCo şirketi için aşağıdaki iş kurallarını içerecek şekilde bir Crow's Foot ERD oluşturun:
   a. Her satış temsilcisi çok sayıda fatura yazar.
   b. Her fatura bir satış temsilcisi tarafından yazılır.
   c. Her satış temsilcisi bir departmana atanır.
   d. Her departmanın çok sayıda satış temsilcisi vardır.
   e. Her müşteri çok sayıda fatura oluşturabilir.
   f. Her fatura bir müşteri tarafından oluşturulur.
15. Şekil P2.15'te gösterilen ERD'de yansıtılan iş kurallarını yazın. (ERD'nin bazı basitleştirici varsayımları yansıttığına dikkat edin. Örneğin, her kitap yalnızca bir yazar tarafından yazılmıştır. Ayrıca, ERD bileşenlerinin yönü ne olursa olsun ERD'nin her zaman "1" tarafından "M" tarafına doğru okunduğunu unutmayın).

Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir. Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.



16. Aşağıdaki açıklamaların her biri için bir Karga Ayağı ERD'si oluşturun. (Birçok kelimesinin veritabanı modelleme Ortamında yalnızca birden anlamına geldiğini unutmayın).

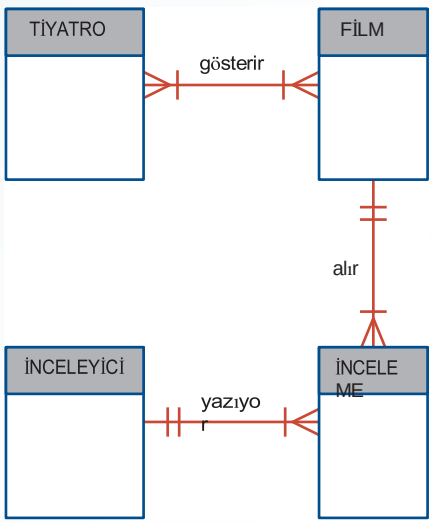
a. MegaCo Corporation'ın bölümlerinin her biri birçok departmandan oluşmaktadır.Her departmanın kendisine atanmış birçok çalışanı vardır, ancak her çalışan yalnızca bir için çalışır. Her departman bir çalışan tarafından yönetilir ve bu yöneticilerin her biri aynı anda yalnızca bir departmanı yönetebilir.

b. Belirli bir süre boyunca, bir müşteri BooksOnline'dan birçok e-kitap indirebilir. E-kitapların her biri, o süre zarfında birçok müşteri tarafından indirilebilir.

c. Bir havayolu şirketi birçok uçuşa atanabilir, ancak her uçuş sadece bir şirketi tarafından gerçekleştirilir.

d. KwikTite şirketi birçok fabrika işletmektedir. Her fabrika bir bölgede yer alır ve her bölge KwikTite'in birçok fabrikasına "ev sahipliği" yapabilir. Her fabrikanın çok sayıda çalışanı vardır, ancak her çalışan yalnızca bir fabrika tarafından istihdam edilmektedir.

e. Bir çalışan birçok derece kazanmış olabilir ve her bir derece birçok çalışan tarafından kazanılmış olabilir.



Telif Hakkı 2023 Cengage Learning. Tüm hakları saklıdır. Tamamen veya kısmen kopyalanamaz, taranamaz veya çoğaltılamaz. Elektronik haklar nedeniyle, bazı üçüncü taraf içerikleri eKitap ve/veya eBölüm(ler)den çıkarılmış olabilir. Editoryal inceleme, bastırılan herhangi bir içeriğin genel öğrenme deneyimini önemli ölçüde etkilemediğine karar vermiştir. Cengage Learning, sonraki hak kısıtlamalarının gerektirmesi halinde ek içeriği istediği zaman kaldırma hakkını saklı tutar.