

XML hakkında). Aynı zamanda, O/R DBMSS, ilişkisel veri yapılarında XML tabanlı belgeler için destek ekledi. Geniş şekilde uygulanabilir ilkelerdeki sağlam temeli nedeniyle, ilişkisel model nesneler ve XML gibi yeni yetenek sınıflarını içerecek şekilde kolayca genişletilir.

İlişkisel ve nesne/ilişkisel veritabanları mevcut veri işleme ihtiyaçlarının çoğunu ele alsa da, bazı İnternet döneminde kuruluşlarda bulunan bazı çok özel zorlukları ele almak için yeni nesil veritabanları ortaya çıkmıştır.

2-5F Ortaya Çıkan Veri Modelleri: Büyük Veri ve NoSQL

Kuruluşların yıllar boyunca biriktirdiği web verilerinin dağlarından kullanılabilir iş bilgilerinin elde edilmesi zorunlu bir ihtiyaç haline gelmiştir. Facebook, Twitter ve LinkedIn gibi kaynaklardan tarama kalıpları, satın alma geçmişleri, müşteri tercihleri, davranış kalıpları ve sosyal medya verileri şeklinde web verileri, yapılandırılmış ve yapılandırılmamış verilerin kombinasyonlarına sahip kuruluşlara sahiptir. Buna ek olarak, akıllı telefonlar ve tabletler gibi mobil teknolojiler, artı her tür GPS, RFID sistemleri, hava sensörleri, biyomedikal cihazlar, uzay araştırması problemleri, araba ve havacılık kara kutuların yanı sıra diğer arası ve hücreli bağlantılı cihazlar oluşturdu. Birden fazla formatta (metin, resim, ses, video vb.) Maddi miktarda veri toplamanın yeni yolları. Veri değiştirme ve toplama ve toplayan bu ağlar arası cihazların bu ağı Nesnelerin İnterneti (LOT) olarak bilinir.

Toplanan veri miktarı her gün katlanarak büyür. Lot, veri büyümesi oranını hızlandırdı, böylece şu anda günlük yaklaşık 2,5 quintillion bayt veri oluşturuldu. Veri büyümesinin hızlı temposu, bir sonraki en büyük zorluklar olarak sistem performansı ve ölçeklenebilirlik ile kuruluşlar için büyük bir zorluk olabilir. Bugünün bilgi teknolojisi (BT) yöneticileri, bu hızla büyüyen verileri küçülen bütçelerle yönetme ihtiyacını sürekli olarak dengeliyor. Tüm bu yakınsak eğilimleri (hızlı veri büyümesi, performans, ölçeklenebilirlik ve daha düşük maliyetler) bulma ve kullanma ihtiyacı, büyük veri adı verilen bir fenomeni tetikledi. - ve sensör oluşturulmuş veriler ve iş içgörü elde ederken, aynı zamanda yüksek performans ve ölçeklenebilirlik sağlarken makul bir maliyetle.

Büyük veriler terimi, birçok çerçevede yasadaki istatistiklere, ekonomiye, hesaplama kadar kullanılmıştır. Terim ilk olarak 1990'larda bir Silikon Grafik Bilimcisi John Mashey tarafından bir bilgi işlem çerçevesinde kullanılmış gibi görünür. Bununla birlikte, Gartner Group'tan bir veri analisti olan Douglas Laney gibi görünür, ilk önce büyük veri veritabanlarının temel özelliklerini tanımladı: cilt, hız ve çeşit veya 3 Vs.

Hacim, depolanan veri miktarlarını ifade eder. İnternet ve sosyal medyanın benimsenmesi ve büyümesi ile şirketler müşterilere ulaşma yollarını çoğaltmıştır. Yıllar geçtikçe ve teknolojik ilerlemelerden yararlanarak, milyonlarca e-işlem verileri için günlük olarak şirket veritabanlarında saklanıyordu. Ayrıca, kuruluşlar son kullanıcılarla etkileşim kurmak için çok Tiple Technologies kullanıyor ve bu teknolojiler veri üretiyor. Bu sürekli büyüyen veri hacmi hızla petabaytlara ulaştı ve hala büyüyor.

Velocity, sadece verilerin büyüdüğü hıza değil, aynı zamanda bilgi ve içgörü oluşturmak için bu verileri hızlı bir şekilde işleme ihtiyacını da ifade eder. İnternet ve sosyal medyanın ortaya çıkmasıyla, iş yanıt süreleri önemli ölçüde küçüldü.

Kuruluşların sadece büyük miktarlarda hızlı bir şekilde veri biriktirmesi gerekmez, aynı zamanda bu tür verileri hızlı bir şekilde işlemelidir. Veri büyümesinin hızı, verilerin kuruluşa borulandığı farklı veri akışlarının sayısındaki artıştan da kaynaklanmaktadır (web, e-ticaret, tweetler, Facebook gönderileri, e-postalar, sensörler, GPS vb.).

Nesnelerin İnterneti (lot)
İnternet üzerinden veri
alışverişi ve
toplama internete bağlı
cihazlardan oluşan bir ağ.

Lot cihazları, veri
toplamak ve İnternet'teki diğer
cihazlarla etkileşim kurmak için
uzaktan yönetilebilir ve

Büyük veriler
Büyük miktarlarda web
tarafından oluşturulan verileri
yönetmenin ve ondan iş
içgörü elde etmenin yeni ve
daha iyi yollarını bulma
hareketi, aynı zamanda
yüksek performans ve
ölçeklenebilirliği
makul bir şekilde sağlar.

3 vs
Büyük veri veritabanlarının üç
temel özelliği: hacim, hız ve
çeşitlilik.

*Steve Lohr, "Büyük Verilerin Kökenleri": Etimolojik Dedektif Hikayesi," New York Times, 1 Şubat 2013.

PDOUGLAS LANEY, "Veri Hacmi, Hız ve Çeşitlilik Kontrol Etme 3LD Veri Yönetimi," Meta Group, 6 Şubat 2011.

Çeşitlilik, toplanan verilerin birden fazla veri formatında gelmesi gerçeğini ifade eder. Bu verilerin büyük bir kısmı, ilişkisel modele dayanan tipik çalışma veritabanları tarafından işlenemeyecek formatlar halinde gelir.

3 vs çerçevesi, şirketlerin şu anda bildiklerini, veritabanlarında toplanan veri miktarının katlanarak boyut ve karmaşıklık olarak arttığını göstermektedir. Geleneksel ilişkisel veritabanları yapılandırılmış verileri yönetmede iyidir, ancak bugünün içinde toplanan veri miktarlarını ve türlerini yönetmek ve işlemek için çok uygun değildir.

çevre.

Sorun şu ki, ilişkisel yaklaşımın organizasyonların ihtiyaçlarını her zaman büyük veri zorluklarıyla eşleştirmemesidir.

Yapılandırılmamış, sosyal medya ve sensör tarafından oluşturulan verileri, satırların ve sütunların geleneksel ilişkisel yapısına sığdırmak her zaman mümkün değildir.

Günlük olarak milyonlarca satır multiformat (yapılandırılmış ve yapılandırılmamış) veri eklemek, kaçınılmaz olarak ilişkisel ortamda bulunmayan daha fazla depolama, işleme gücü ve sofistike veri analizi araçlarına ihtiyaç duyacaktır. Büyük veri sorunu için RDBMS ortamında gereken yüksek hacimli uygulamaların türü, donanım, depolama ve yazılım lisanslarını genişletmek için ağır bir fiyat etiketi ile birlikte gelir.

OLAP araçlarına dayanan veri analizinin, yüksek yapılandırılmış verilerle ilişkisel ortamlarda çok başarılı olduğu kanıtlanmıştır. Bununla birlikte, web kaynaklarından toplanan çok miktarda yapılandırılmamış verilerde kullanılabilir veriler için madencilik farklı bir yaklaşım gerektirir.

Veri yönetimi ihtiyaçlarına yönelik "tek bedene uyan" bir tedavi yoktur (birçok yerleşik veritabanı satıcısı muhtemelen sizi bu fikir üzerinde satmaya çalışacaktır). Bazı kuruluşlar için, büyük veri analizi için son derece ölçeklenebilir, hataya dayanıklı bir altyapı oluşturmak, iş hayatta kalma meselesi olabilir. İş dünyası, rekabet avantajı elde etmek için teknolojiden yararlanan birçok şirket örneğine ve onu kaçıran diğerlerine sahiptir. Sadece iş ortamının nasıl farklı olacağını kendinize sorun:

BlackBerry, ortaya çıkan Apple akıllı telefon teknolojisine hızla cevap vermişti.

MySpace, Facebook'un zamanında meydan okumasına cevap vermişti.

Blockbuster Netflix iş modeline daha erken tepki vermişti.

Barnes & Noble, Amazon'dan önce uygun bir internet stratejisi geliştirmişti.

Yayın televizyon ağları Hulu, Appletv ve Roku gibi akış hizmetlerine uyum sağlamada başarılı olacak mı? Ortaklıklar ve birleşmeler, endüstri değişen teknolojik olasılıklara yanıt verirken şüphesiz ev eğlencesi manzarasını değiştirecek.

Geleneksel haber kuruluşları, bin yıllık kuşağın değişen haber tüketim modellerine uyum sağlayabilecek mi?

Her tür şirket tarafından yeni hizmet türleri oluşturmak için büyük veri analizi kullanılmaktadır. Örneğin, Amazon başlangıçta "Big Box" mağazalarıyla düşük maliyetli bir sağlayıcı olarak yarıştı. Amazon sonunda film ve müzik hizmetinde yarışmaya başlamak için depolama ve işleme teknolojilerinden yararlandı ve daha yakın zamanda, öngörücü nakliye gibi yenilikçi hizmetler oluşturmak için büyük verilerden yararlandı, öngörücü nakliye, bir ürünün ne zaman ne zaman olacağını tahmin etmek için bir müşterinin satın alma patlayı kullanır. İhtiyaç duyduğunu fark etmeden önce müşteriye ihtiyaç duyulur ve müşteriye gönderir! Amazon ayrıca, doğal dil işlemeyi gerçekleştirmek için Alexa Serice'i kullanan Amazon Echo gibi ürünlerin satışlarıyla da başarılı oldu. Bu 'sürekli dinleme' cihazları, dünyanın dört bir yanındaki evlere gömüldü ve Amazon'a mevcut hizmetleri geliştirmek ve gelecekteki hizmetlerde yeniliği desteklemek için analiz edebileceği eş görülmemiş seviyeler ve veri türleri sağlar.

Daha önce kullanılmayan büyük veri mağazalarından değer yaratmak için şirketler yeni büyük veri teknolojileri kullanıyor. Bu gelişmekte olan teknolojiler, kuruluşların işlemesine izin verir

Hadoop

Java tabanlı, açık kaynaklı, yüksek hızlı, hataya toleranslı dağıtılmış depolama ve hesaplama

Hadoop, verileri depolamak ve işlemek için binlerce bilgisayar düğümünün kümelerini oluşturmak için düşük maliyetli donanım kullanır.

Hadoop Dağıtılmış Dosya Sistemi (HDFS)

Yüksek hızlarda büyük miktarda veriyi yönetmek için tasarlanmış, yüksek dağıtılmış, hataya toleranslı bir dosya depolama sistemi.

isim düğümü

Hadoop Dağıtılmış Dosya Sisteminde (HDFS) kullanılan üç tür düğümden biri. Düğüm adı, dosya sistemi hakkındaki tüm meta verileri saklar. Ayrıca bkz. İstemci düğümü ve veri

Veri düğümü

Üç türden biri Hadoop dağıtılmış dosya sisteminde (HDFS) kullanılan düğümler. Veri düğümü sabit boyuttaki veri bloklarını depolar (diğer veri düğümlerine çoğaltılabilir). Ayrıca bkz. İstemci düğümü ve ad

istemci düğümü

Hadoop Dağıtılmış Dosya Sisteminde (HDFS) kullanılan üç tür düğümden biri. İstemci düğümü, kullanıcı uygulaması ve HDFS arasındaki arayüz olarak işlev görür. Ayrıca bkz. Ad düğümü ve veri

Harita

Hızlı veri analizi hizmetleri sunan açık kaynaklı bir uygulama programlama arayüzü (API); Kuruluşların büyük veri depolarını işlemesine izin veren ana büyük veri teknolojilerinden

Nosql

Geleneksel ilişkisel veritabanı modeline dayanan yeni nesil veritabanı yönetim sistemleri.

Maliyet etkin bir şekilde birden çok formatta büyük veri depoları. En sık kullanılan büyük veri teknolojilerinden bazıları Hadoop ve NoSQL veritabanlarıdır.

Hadoop, Java tabanlı, açık kaynaklı, yüksek hızlı, hataya toleranslı dağıtılmış depolama ve hesaplama çerçevesidir. Hadoop, verileri depolamak ve işlemek için binlerce bilgisayar düğümünün kümelerini oluşturmak için düşük maliyetli donanım kullanır. Hadoop, Google'ın dağıtılmış dosya sistemleri ve paralel işleme üzerindeki çalışmalarından kaynaklanmıştır ve şu anda Apache Software Foundation tarafından desteklenmektedir. Hadoop'un birkaç modülü vardır, ancak iki ana bileşen Hadoop Dağıtılmış Dosya Sistemi (HDES) ve MapReduce'dur.

Hadoop Dağıtılmış Dosya Sistemi (HDFS), yüksek miktarda veriyi yüksek hızlarda yönetmek için tasarlanmış yüksek oranda dağıtılmış, hataya dayanıklı bir dosya depolama sistemidir. Yüksek verim elde etmek için HDFS, bir kez yazma, birçok modeli okuyun. Bu, veriler yazıldıktan sonra değiştirilemeyeceği anlamına gelir. HDFS üç tür düğüm kullanır: dosya sistemi hakkında tüm meta verileri depolayan bir ad düğümü, sabit boyuttaki veri bloklarını (diğer veri düğümlerine çoğaltılabilen) depolayan bir veri düğümü ve arayüz olarak hareket eden bir istemci düğümü Kullanıcı uygulaması ve HDFS arasında.

MapReduce, hızlı veri analizi hizmetleri sunan açık kaynaklı bir uygulama programlama arayüzüdür (API). MapReduce, verilerin işlenmesini binlerce düğüm arasında paralel olarak dağıtır. MapReduce yapılandırılmış ve yapılandırılmış verilerle çalışır. MapReduce çerçevesi iki ana işlev sağlar: harita ve azaltmak. Genel olarak, harita işlevi bir iş alır ve daha küçük çalışma birimlerine ayırır ve azaltma işlevi düğümlerden oluşturulan tüm çıktı sonuçlarını toplar ve bunları tek bir sonuç kümesine entegre eder. MapReduce'un kendisi bugün oldukça sınırlı olarak görülmesine rağmen, büyük verilerin nasıl işlendiğine ilişkin paradigmayı tanımladı.

NoSQL, yapılandırılmış ve yapılandırılmamış verileri etkili şekillerde depolayan büyük ölçekli dağıtılmış bir veritabanı sistemidir. NoSQL veritabanları Bölüm 1L4, Big Veri ve NoSQL'de daha ayrıntılı olarak ele alınmaktadır.

Hadoop Technologies, verilerin (yapılandırılmış veya yapılandırılmamış) düşük maliyetli emtia donanımı ağı kullanılarak paralel olarak dağıtıldığı, çoğaltıldığı ve işlendiği büyük veri analizi için bir çerçeve sağlar. Hadoop, verileri depolamak ve yönetmek için yeni yollar tanıttı. Kafa karışmayın: Hadoop ve NoSQL veritabanları genellikle birlikte tartışılır, çünkü her ikisi de büyük veri sorunlarını ele almadaki bileşenlerdir. Ancak, Hadoop ne bir veritabanı ne de bir veri modelidir. Dağıtılmış bir dosya depolama ve işleme modelidir. Hadoop DBM'leri yok. NoSQL veritabanları veritabanlarıdır ve NoSQL modeli, verilerin ilişkisel olmayan bir şekilde depolanmasına ve işlenmesine yaklaşmanın farklı bir yolunu temsil eder. NoSQL veritabanları, yapılandırılmış verilerin işlenmesi için dağıtılmış, hataya toleranslı veritabanları sağlar.

Büyük veri analitiğinden türetilen büyük kazanımların potansiyeli ile, bazı kuruluşların Web verileri dağlarında gizlenmiş bilgilerin zenginliğini incelemek ve rekabet avantajı elde etmek için noSQL veritabanları gibi ortaya çıkan büyük veri teknolojilerine yönelmesi şaşırtıcı değildir.

Not

Bu, ilişkisel veritabanlarının büyük veri zorlukları olan kuruluşlarda bir yeri olmadığı anlamına mı geliyor? Hayır, ilişkisel veritabanları çoğu günlük işlemleri ve yapılandırılmış veri analizi ihtiyaçlarını desteklemek için tercih edilen ve baskın veritabanları olmaya devam etmektedir. Her DBMS teknolojisinin uygulama alanları vardır ve en iyi yaklaşım iş için en iyi aracı kullanmaktır. Perspektif olarak, nesne/ilişkisel veritabanları operasyonel pazar ihtiyaçlarının yüzde 98'ine hizmet eder. Büyük veri ihtiyaçları için Hadoop ve NoSQL veritabanları seçenekler arasındadır.

Bölüm 14, Big Veri ve NoSQL, bu seçenekleri daha ayrıntılı olarak tartışmaktadır.

Hadoop hakkında daha fazla bilgi için hadoop.apache.org adresini ziyaret edin.

NoSQL veritabanları

Amazon'da bir ürün aradığınızda, Facebook'taki arkadaşlara mesaj gönderin, YouTube'da bir video izleyin veya Google Haritalar'da yol tarifi arayın, NoSQL veritabanı kullanıyorsunuz.

Herhangi bir yeni teknolojiye olduğu gibi, NoSQL terimi birçok farklı teknolojiye gevşek bir şekilde uygulanabilir. Bununla birlikte, bu bölüm, büyük veri döneminin belirli zorluklarını ele alan ve aşağıdaki genel özelliklere sahip yeni nesil veritabanlarına atıfta bulunmak için NoSQL kullanıyor:

İlişkisel modele ve SQL'e dayanmazlar; Dolayısıyla NoSQL adı.

Yüksek dağıtılmış veritabanı mimarilerini desteklerler.

Yüksek ölçeklenebilirlik, yüksek mevcudiyet ve hata toleransı sağlarlar.

Çok büyük miktarda seyrek veriyi desteklerler (çok sayıda öznitelik içeren ancak gerçek veri örneği sayısının düşük olduğu veriler).

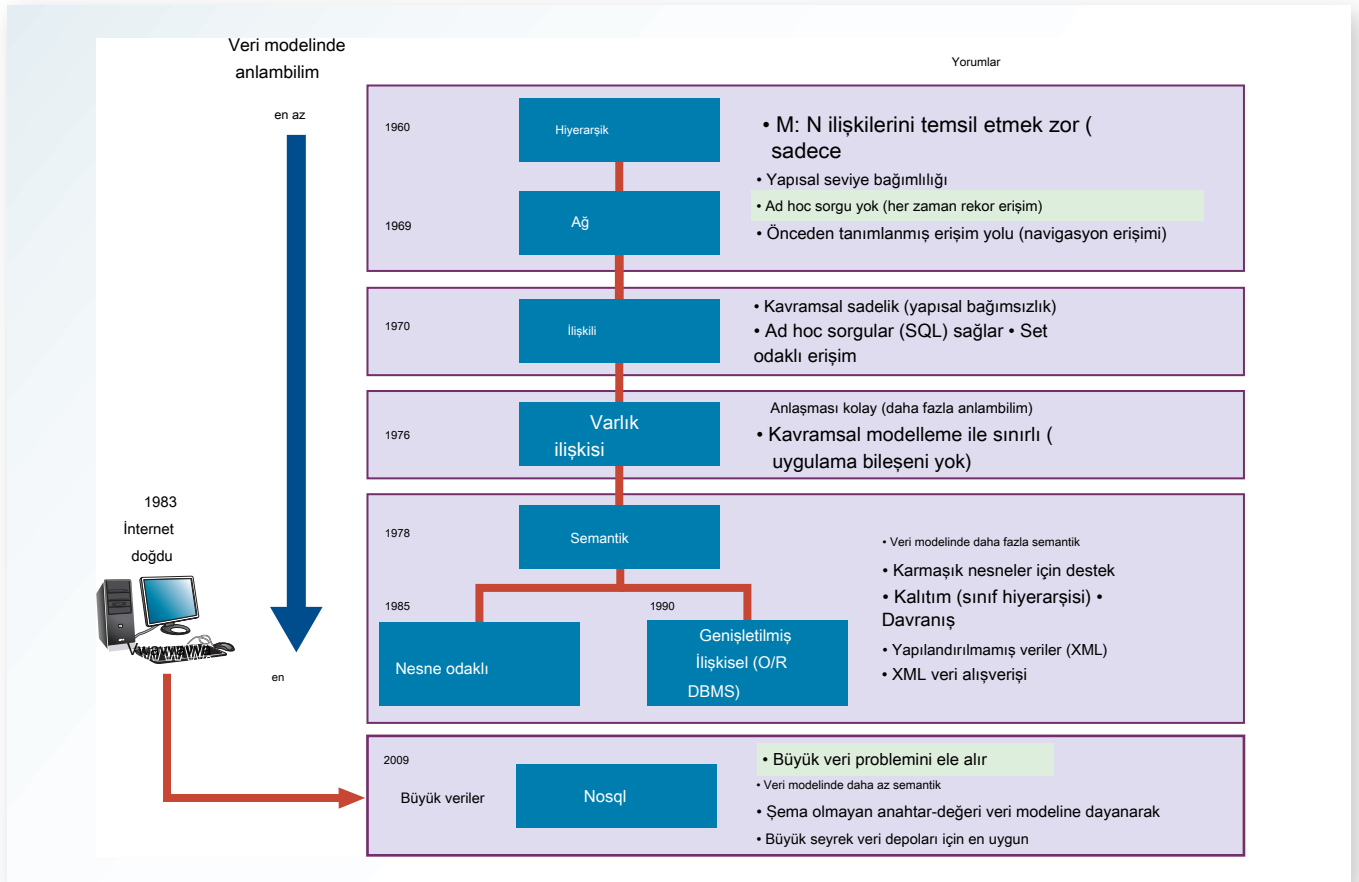
İşlem tutarlılığı yerine performansa yöneliktirler.

Veri depolama ve manipülasyon için çok kapsamlı ve uyumlu bir yaklaşım sağlayan ilişkisel modelin aksine, NoSQL modeli veri depolama ve manipülasyonuna çeşitli yaklaşımlar için geniş bir şemsiye. Bu yaklaşımların en yaygın olanı, Bölüm 14'te ayrıntılı olarak tartışıldığı gibi anahtar değer depoları, belge veritabanları, sütun veritabanları ve grafik veritabanlarıdır.

2-5G veri modelleri: bir özet

DBMS'lerin evrimi her zaman giderek daha karmaşılaşan gerçek dünya verilerini modelleme ve yönetmenin yeni yollarını arayarak yönlendirilmiştir. En yaygın olarak tanınan veri modellerinin bir özeti Şekil 2.5'teki Showm'dur.

Şekil 2.5 Veri modellerinin evrimi



Veri modellerinin evriminde, bazı yaygın özellikler onları geniş ölçüde kabul etti:

Bir veri modeli, veritabanının anlamsal eksiksizliğinden ödün vermeden bir dereceye kadar kavramsal sadelik göstermelidir. Kavramsallaştırılması gerçek dünyadan daha zor olan bir veri modeline sahip olmak mantıklı değildir. Aynı zamanda, model netlik ve alaka düzeyi göstermelidir; Yani, veri modeli açık olmalı ve problem alanı için geçerli olmalıdır. Bir veri modeli gerçek dünyayı olabildiğince yakından temsil etmelidir.

Bu amaç, modelin veri temsiline daha fazla anlambilim ekleyerek daha kolay gerçekleştirilir. (Anlambilim dinamik veri davranışı ile ilgilidir, ancak veri temsili gerçek dünya senaryosunun statik yönünü oluştururken.) Başka bir deyişle, model doğru olmalı ve tam olarak gerekli veriler dahil edilmeli ve uygun şekilde tanımlanmalıdır.

Gerçek dünya dönüşümlerinin (davranış) temsili, veri modunun amaçlanan kullanımının gerektirdiği tutarlılık ve dürüstlük özelliklerine uygun olmalıdır.

Her yeni veri modeli önceki modellerin eksikliklerini ele alır. Ağ modeli hiyerarşik modelin yerini aldı, çünkü birincisi karmaşık (çok fazla) ilişkileri temsil etmeyi çok daha kolay hale getirdi. Buna karşılık, ilişkisel model, daha basit veri temsili, üstün veri bağımlılığı ve kullanımı kolay sorgu dili aracılığıyla hiyerarşik ve ağ modellerine göre çeşitli avantajlar sunmaktadır: Bu özellikler onu iş uygulamaları için tercih edilen veri modeli haline getirmiştir. OO veri modeli, zengin bir semantik çerçeve içinde karmaşık veriler için destek getirdi. ERDM, ilişkisel modele birçok OO özelliği ekledi ve iş ortamında güçlü pazar payını korumasına izin verdi. Son yıllarda, büyük veri olgusu, geleneksel veri yönetimi ile bir molayı temsil eden verileri modellemek, depolamak ve yönetmek için alternatif yolların geliştirilmesini teşvik etmiştir.

Tüm veri modellerinin eşit yaratılmadığını unutmayın; Bazı veri modelleri bazı görevler için diğerlerinden daha uygundur. Örneğin, kavramsal modeller üst düzey veri modellemesi için daha uygundur, uygulama modelleri ise depolanan verileri uygulama amacıyla yönetmek için daha iyidir. ER modeli kavramsal bir modelin örneğidir, hiyerarşik ve ağ modelleri uygulama modellerine örnektir. Aynı zamanda, ilişkisel model ve OODM gibi bazı modeller hem kavramsal hem de uygulama modelleri olarak kullanılabilir. Tablo 2.2, çeşitli veritabanı modellerinin avantajlarını ve dezavantajlarını özetlemektedir.

Not

Tüm veritabanları, veritabanı içinde ortak bir veri havuzu kullanıldığını varsayar. Bu nedenle, tüm veritabanı modelleri veri paylaşımını teşvik eder, böylece bilgi adalarının potansiyel sorununu azaltır.

Şimdiye kadar, daha belirgin veri modellerinin temel yapılarına tanıtıldınız. Her model, gerçek dünyadaki veri ortamının anlamını yakalamak için bu yapıları kullanır. Tablo 2.3, çeşitli veri modelleri tarafından kullanılan temel terminolojiyi göstermektedir.

2-6 derece veri soyutlaması

10 veritabanı tasarımcısına bir veri modelinin ne olduğunu sorarsanız, veri soyutlama derecesine dayanan 10 farklı cevapla sonuçlanırsınız. Veri soyutlamasının anlamını göstermek için otomotiv tasarımı örneğini düşünün. Bir otomobil tasarımcısı, üretilecek araba konseptini çizerek başlar. Daha sonra, mühendisler temel konsepti üretilebilecek bir yapıya aktarmaya yardımcı olan detayları tasarlar. Son olarak, mühendislik çizimleri fabrika katında kullanılacak üretim özelliklerine çevrilir. Gördüğünüz gibi, arabayı üretme süreci yüksek bir soyutlama seviyesinde başlar ve bir

Tablo 2.3 Veri Modeli Temel Terminoloji Karşılaştırması

Gerçek dünya	Örnek	Dosya İşleme	Hiyerarşik model	Ağ modeli	İlişkisel model	Model	00 Model
Bir grup satıcı	Satıcı dosya kabini	Dosya	Segment türü	Kayıt türü	Masa	Varlık seti	Sınıf
Tek bir satıcı	Global Malzemeler	Kayıt	Segment Ortaya çıkma	Mevcut Kayıt	Sıra (Tuple)	Varlık ortaya çıkma	Nesne örneği
İletişim adı	Johnny Ventura	Alan	Segment	Kayıt Alanı	Tablo Öznitelik	Varlık özniteliği	Nesne özniteliği
Satıcı tanımlayıcısı	G12987	İndeks	alanı dizisi	Kayıt	Anahtar	varlık tanımlayıcısı	nesne tanımlayıcısı

Not: Bu tabloda kullanılan terimler hakkında ek bilgi için, bu kitaba eşlik eden ilgili bölümlere ve çevrimiçi eklere başvurun. Örneğin, 00 modeli hakkında daha fazla bilgi edinmek istiyorsanız, Ek G'ye, nesne yönelimli veritabanlarına bakın.

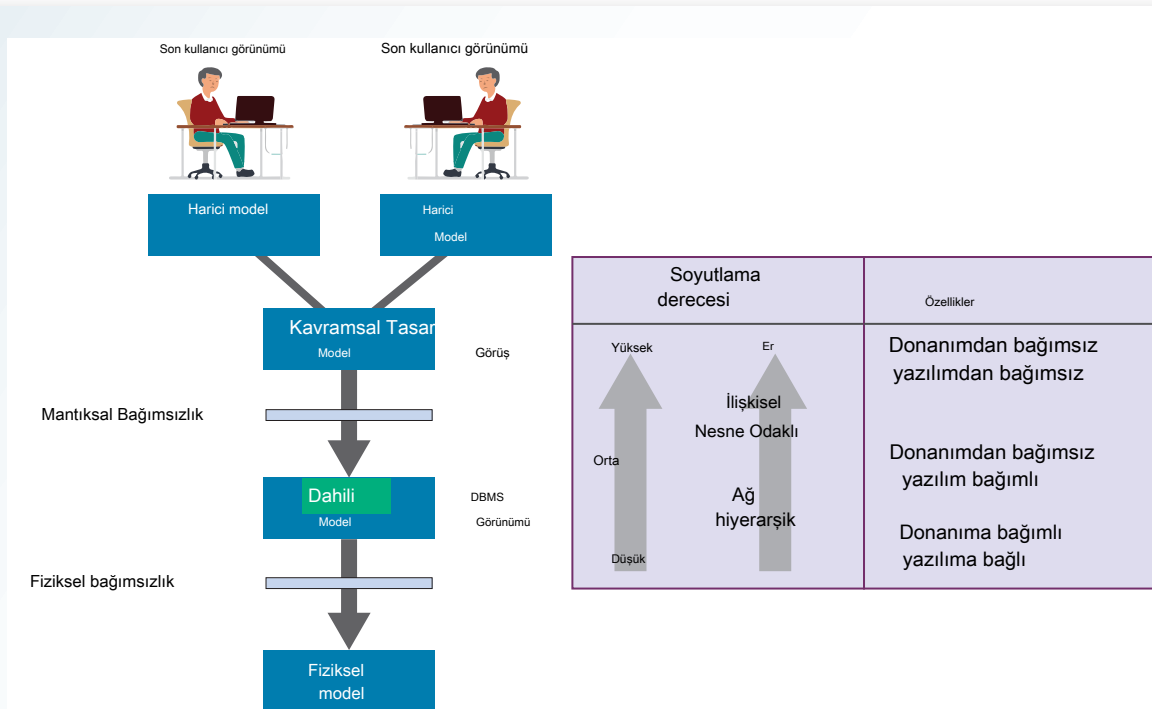
Amerikan Ulusal
Standartlar Enstitüsü (ANSI)
1975 yılında SPARC
komitesi aracılığıyla DBTG
önerilerini ve artırılmış
veritabanı standartlarını kabul
eden grup.

sürekli artan ayrıntı seviyesi. Mühendislik detayları uygun şekilde belirtilmedikçe fabrika zemin süreci devam edemez ve mühendislik detayları tasarımcı tarafından oluşturulan temel çerçeve olmadan var olamaz. Kullanılabilir bir veritabanı tasarlamak aynı temel işlemi izler.

Yani, bir veritabanı tasarımcısı genel veri ortamının soyut bir görünümüyle başlar ve tasarım uygulamaya yaklaştıkça ayrıntıları ekler. Soyutlama seviyelerinin kullanılması, bir kuruluşun farklı seviyelerinde çoklu (ve bazen çelişkili) veri görünümünün entegre edilmesinde de çok yardımcı olabilir.

1970'lerin başında, Amerikan Ulusal Standartlar Enstitüsü (ANSI) Standartları Planı ve Gereksinimler Komitesi (SPARC), veri soyutlamasına dayalı veri modellemesi için bir çerçeve tanımlamıştır. Ortaya çıkan ANSI/SPARC mimarisi üç düzey veri soyutlama tanımlar: dış, kavramsal ve iç. Bu çerçeveyi, Şekil 2.6'da gösterildiği gibi, daha iyi durma veritabanı modellerini kullanabilirsiniz. Şekilde, ANSI/SPARC çerçevesi

Şekil 2.6 Veri soyutlama seviyeleri



Dahili modelin fiziksel düzeyde uygulama ayrıntılarını açıkça ele almak için fiziksel bir model eklenmesi ile genişletilmiştir.

2-6A Harici Model

Harici model, son kullanıcıların veri ortamına bakış açısıdır. Son kullanıcılar terimi, verileri manipüle etmek ve bilgi oluşturmak için uygulama programlarını kullanan kişileri ifade eder.

Son kullanıcılar genellikle bir uygulamanın belirli bir iş birimi odağına sahip olduğu bir ortamda çalışır. Şirketler genellikle satış, finans ve pazarlama gibi çeşitli iş birimlerine ayrılır, her iş birimi belirli kısıtlamalara ve gereksinimlere tabidir ve her biri kuruluştaki genel verilerin bir alt kümesini kullanır. Bu nedenle, bu iş birimleri içindeki son kullanıcılar veri alt kümelerini kuruluş içindeki diğer birimlerden ayrı veya dışsal olarak görür.

Veriler modellendiğinden, ER diyagramları dış görünüşleri temsil etmek için kullanılacaktır.

Harici bir görünümün spesifik bir temsili harici bir şema olarak bilinir. Dış modelin görüşünü göstermek için Tiny College'ın veri ortamını inceleyin.

Şekil 2.7, iki küçük kolej iş birimi için dış şemaları sunmaktadır: öğrenci kaydı ve sınıf planlaması. Her dış şema, iş birimi tarafından getirilen uygun varlıkları, ilişkileri, süreçleri ve kısıtlamaları içerir. Ayrıca, uygulama views birbirinden izole edilmesine rağmen, her görünümün ortak bir varlığı diğer görünümle paylaştığını unutmayın. Örneğin, kayıt ve planlama harici şemaların varlıklar sınıfını ve kursunu paylaşır.

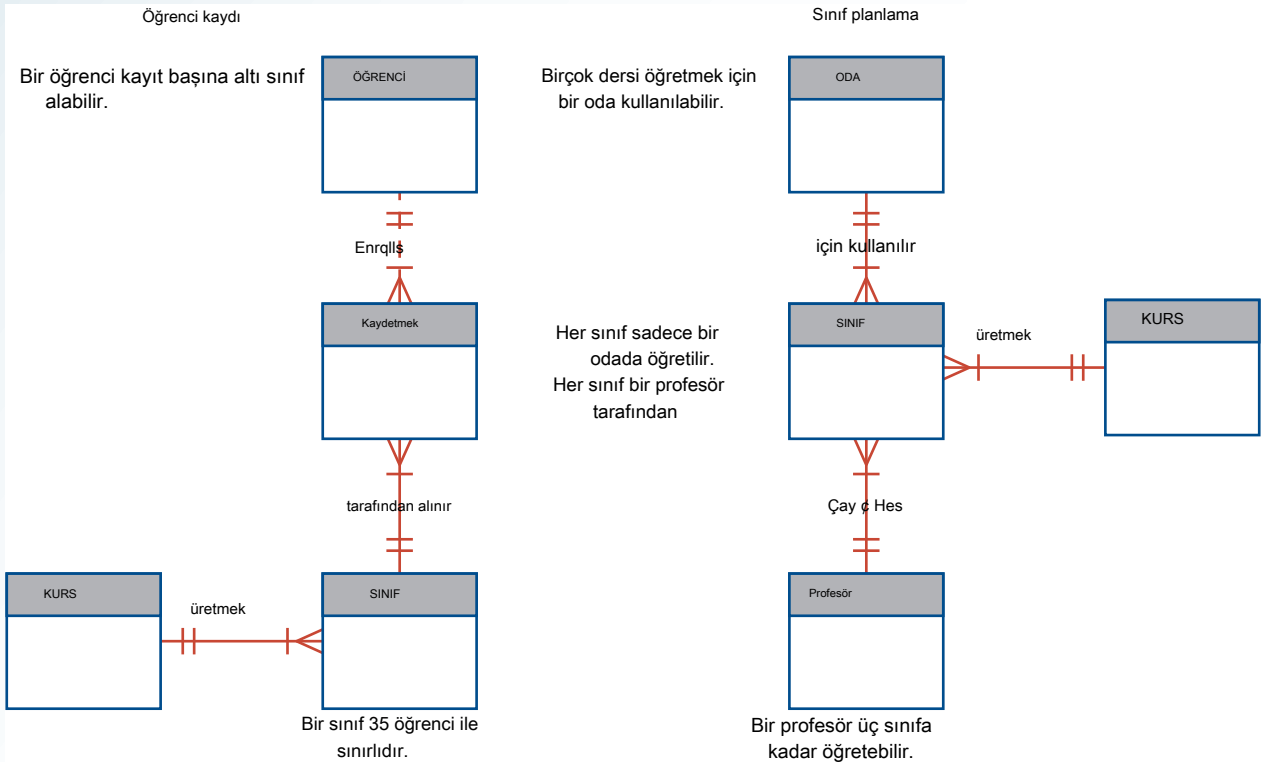
harici model

Son kullanıcının veri ortamına bakışı. İş odağı göz önüne alındığında, harici bir model küresel veritabanı şemasının bir veri alt kümesiyle çalışır.

harici şema

Dış görünüşün özel temsili; son kullanıcının veri ortamına bakışı.

Şekil 2.7 Tiny College için Harici Modeller



kavramsal model

Kavramsal tasarım sürecinin çıktısı. Kavramsal model, tüm bir veritabanının küresel görünümünü sağlar ve ana veri nesnelerini açıklar ve ayrıntılardan kaçınır.

Kavramsal Şema Genellikle grafiksel olarak ifade edilen kavramsal modelin bir kavramsal şemasıdır.

Yazılım Bağımsızlığı Herhangi bir modelin veya uygulamanın, onu uygulamak için kullanılan yazılıma bağımlı

donanım bağımsızlığı Bir modelin modelin uygulamasında kullanılan donanıma bağımlı olmadığı bir durum. Bu nedenle, donanımdaki değişikliklerin kavramsal düzeyde veritabanı tasarımını etkilememesini sağlar.

mantıksal tasarım Tasarım aşamasında, kavramsal tasarımı seçilen DBM'lerin gereksinimleriyle eşleştiren bir aşama ve bu nedenle yazılıma bağımlıdır. Mantıksal tasarım, kavramsal tasarımı DB2, Sol Server, Oracle, IMS, Informix, Access veya Ingress gibi seçilmiş bir veritabanı yönetim sistemi için dahili model olarak kabul edilir.

Şekil 2.7'de temsil edilen ER'lere dikkat edin:

Bir profesör birçok ders verebilir ve her sınıf sadece bir profesör tarafından öğretilir; Profesör ve sınıf arasında bir L: M ilişkisi vardır.

Bir sınıf birçok öğrenciyi kaydedebilir ve her öğrenci birçok sınıfa kaydolabilir, böylece öğrenci ve sınıf arasında bir M: N ilişkisi yaratabilir. (Bölüm 4'teki kayıt varlığının kesin doğası hakkında bilgi edineceksiniz.)

Her ders birçok sınıf oluşturabilir, ancak her sınıf tek bir kursa atıfta bulunur. Örneğin, bir veritabanı dersinin CIS-420 ders koduna sahip birkaç sınıf (bölüm) olabilir. Bu sınıflardan biri MWF'de sabah 8: 00-20: 00 saatleri arasında sunulabilir, diğeri MWF'de 13: 00'dan itibaren sunulabilir. 13: 50'ye kadar, üçte biri Perşembe günleri 18: 00'dan itibaren sunulabilir. 20:40 Yine de, her üç sınıfın da kurs kodu CIS-420 var.

Son olarak, bir sınıf bir oda gerektirir, ancak birçok sınıf için bir oda planlanabilir.

Yani, her sınıf birkaç sınıf için kullanılabilir: biri sabah 9: 00'da, biri LL: 00'da ve biri 13: 00'da.

Başka bir deyişle, oda ve sınıf arasında bir L: M ilişkisi vardır.

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

Her iş biriminin operasyonlarını desteklemek için gereken belirli verileri tanımlamak kolaydır.

Modelin yeterliliği hakkında geri bildirim sağlayarak tasarımcının işini kolaylaştırır.

Özellikle, model, dış modelleri tarafından tanımlanan tüm işlemleri ve tüm operasyonel gereksinimleri ve kısıtlamaları desteklediğinden emin olmak için kontrol edilebilir.

Veritabanı tasarımında güvenlik kısıtlamalarının sağlanmasına yardımcı olur. Tüm bir veritabanına zarar vermek Her iş birimi yalnızca bir veri alt kümesiyle çalıştığından daha zordur.

Uygulama programı geliştirmeyi çok daha basit hale getirir.

2-6B Kavramsal Model

Kavramsal model, tüm kuruluş tarafından tüm veritabanının küresel bir görünümünü temsil eder. Yani, kavramsal model, tüm dış görüşleri (varlıklar, ilişkiler, kısıtlamalar ve süreçler) Şekil 2.8'de gösterildiği gibi işletmedeki verilerin tek bir küresel görünümüne entegre eder. Kavramsal bir şema olarak da bilinir, ana veri nesnelerinin tanımlanması ve üst düzey açıklaması için temeldir (veritabanı modeline özgü ayrıntılardan kaçınır).

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

Kavramsal model bazı önemli avantajlar sağlar. İlk olarak, veri ortamının anlaşılması nispeten kolay olan bir kuş gözü (makro seviyesi) görünümünü sağlar. Örneğin, Şekil 2.8'deki kavramsal modeli inceleyerek Tiny College'ın veri ortamının bir özetini alabilirsiniz.

İkincisi, kavramsal model hem yazılım hem de donanımdan bağımsızdır. Yazılım bağımsızlığı, modelin modeli uygulamak için kullanılan DBMS yazılımına bağımlı olmadığı anlamına gelir. Donanım bağımsızlığı, modelin modelin uygulanmasında kullanılan donanıma bağımlı olmadığı anlamına gelir. Bu nedenle, donanım veya DBMS yazılımındaki değişikliklerin kavramsal düzeyde veritabanı tasarımı üzerinde hiçbir etkisi olmayacaktır.

Genel olarak, mantıksal tasarım terimi, herhangi bir DBM'de uygulanabilecek kavramsal bir veri modeli oluşturma görevini ifade eder.