

İLİŞKİSEL ve İLİŞKİSEL OLMAYAN (NOSQL) VERİ TABANI SİSTEMLERİ

*GİRİŞ

*BİLİŞİM SİSTEMLERİ VE YÖNETİMİ

*VERİ TABANI VE VERİ
TABANI YÖNETİM SİSTEMLERİ

*VERİ TABANI TASARIM

*İLİŞKİSEL VE İLİŞKİSEL OLMAYAN
(NoSQL) VERİ TABANI SİSTEMLERİ

*VERİTABANI MİMARİLERİNİN
PERFORMANS KARŞILAŞTIRMASI

*SONUÇ VE DEĞERLENDİRME

GİRİŞ

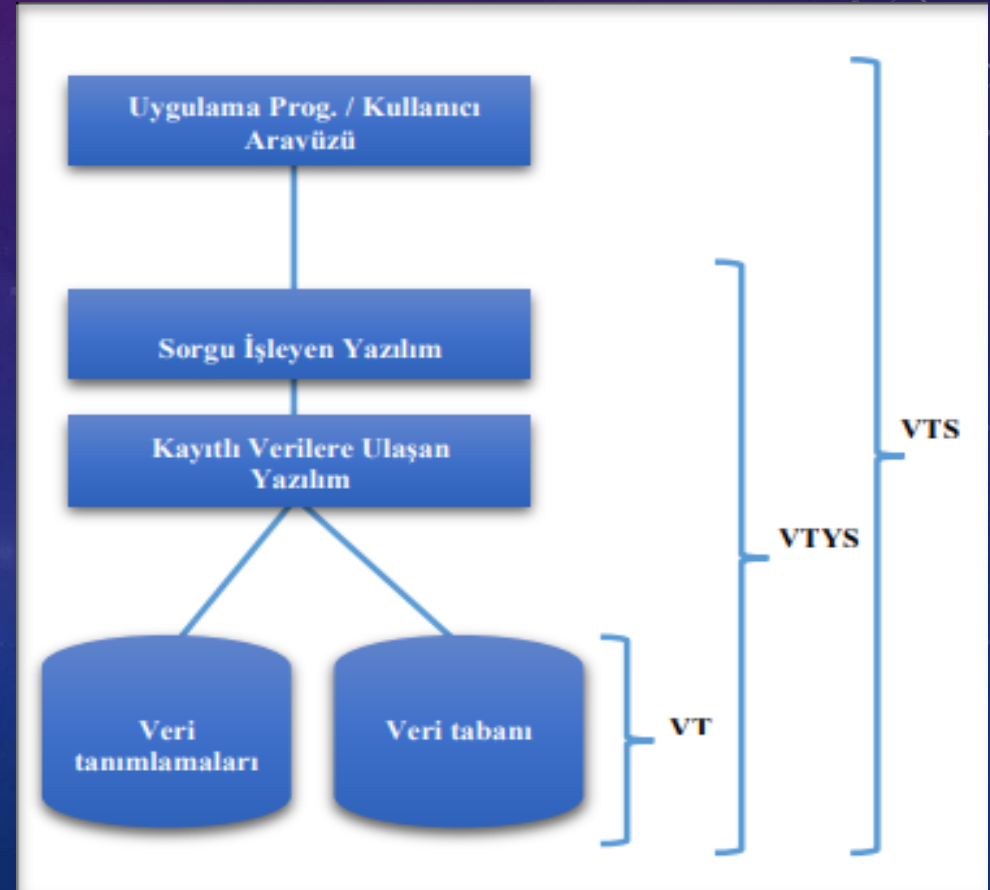
- Bilgisayar ve iletişim teknolojilerinde yaşanan hızlı gelişim her geçen gün daha fazla organizasyonu etkileyerek farklı çözümler üretmeye zorlamaktadır.
- Günümüzde yaşanan bu değişim ve gelişim, verilerin modellenerek saklanmasını ve dolayısıyla veri tabanı kullanımını zorunlu kılmaktadır
- Performans ve esneklik özellikleri ile ilişkisel olmayan veri tabanı yönetim sistemleri (NoSQL) eBay ve Amazon gibi dünyaca ünlü şirketler tarafından tercih edilebilir hale gelmiştir
- “bilişim sistemleri” ve “veri tabanı” kavramları incelenerek ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemleri mimari performansının detaylı karşılaştırılması yapılmıştır

BİLİŞİM SİSTEMLERİ VE YÖNETİMİ

- Bilişim sistemlerinde üç aktivite bilgiyi üretmek için gereklidir. Bu aktiviteler: girdi, işlem ve çıktıdır. Girdi, organizasyonun içinden veya dış çevresinden, ham bilgileri (veriyi) toplamaktır. İşlem, bu ham veriyi daha anlamlı biçime çevirir. Çıktı, işlenmiş bilgiyi (enformasyon), insanlara veya kullanılacak olan aktivitelere aktarır.

VERİ TABANI VE VERİ TABANI YÖNETİM SİSTEMLERİ

- Veri tabanları gerçekte var olan ve birbirleriyle ilişkisi olan nesneleri ve ilişkileri modeller
- Veri tabanı yönetim sistemleri veri tabanı yönetiminin bir parçası olarak, verinin nasıl depolanacağı, kullanılacağı ve erişileceğini mantıksal olarak yönlendiren bir kurallar sistemidir.



DÜZ VERİ MODELİ

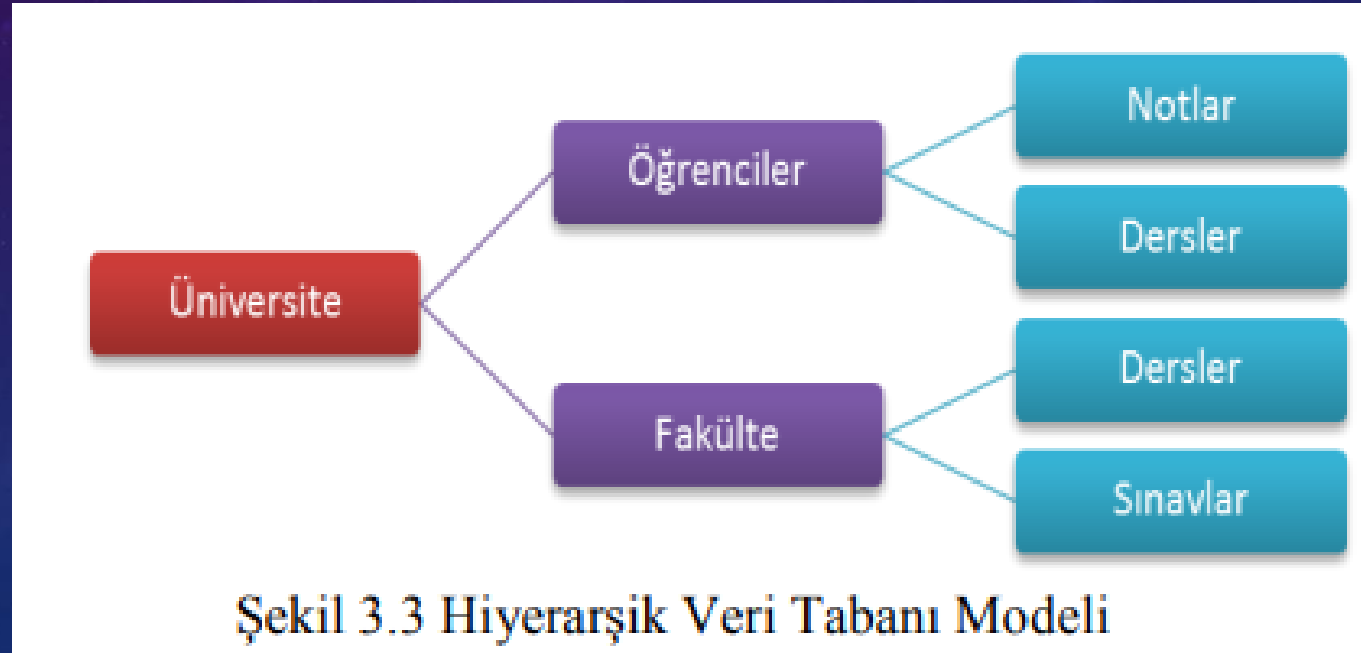
- Düz model veya tablo modeli: İki boyutlu veri grubundan oluşur. Sütunlarda verilerin benzer özellikleri, satırlarda ise veri grupları yer alır. Kullanıcı adlarının ve şifrelerinin tutulduğu veri tabanı buna örnek olarak verilebilir

	Ad Soyad	Kullanıcı Adı	Parola
Kayıt 1	Murat ERGİN	Mergin	kjVdb125
Kayıt 2	Ayşe YILMAZ	Ayılmaz	Bks46db7
Kayıt 3	Can TÜRK	Cturk	fhG8dbt9

Şekil 3.2 Düz veri modeli örneği
(Instance of flat data model)

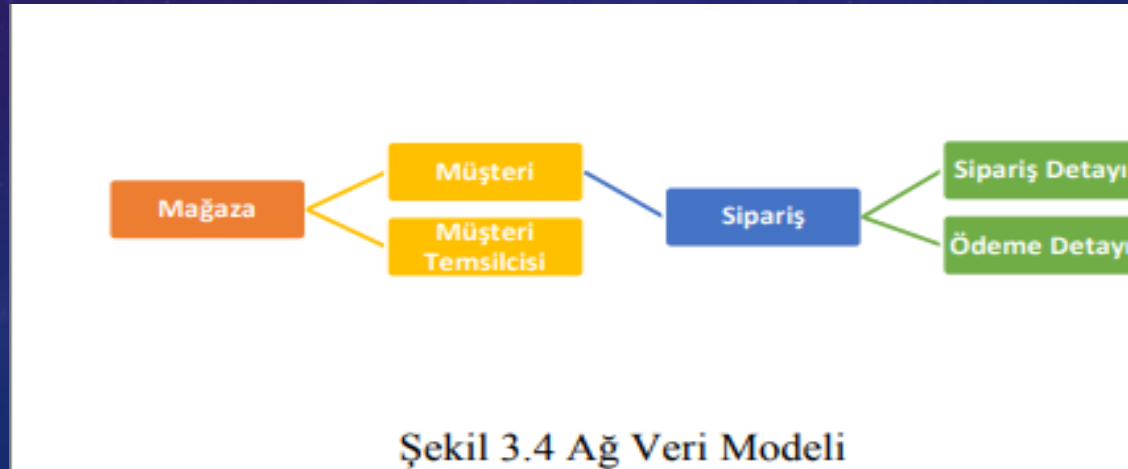
HİYERARŞİK VERİ TABANI MODELİ

Kayıtlar ağaç mimarisi şeklinde yukarıdan aşağı sıralanmaktadır. Kök adı verilen ilk kaydın bir veya daha çok çocuk kayıtları vardır. Çocuk kayıtlarında kendi çocuk kayıtları olabilir. Kök haricinde bütün kayıtların bir ebeveyni vardır



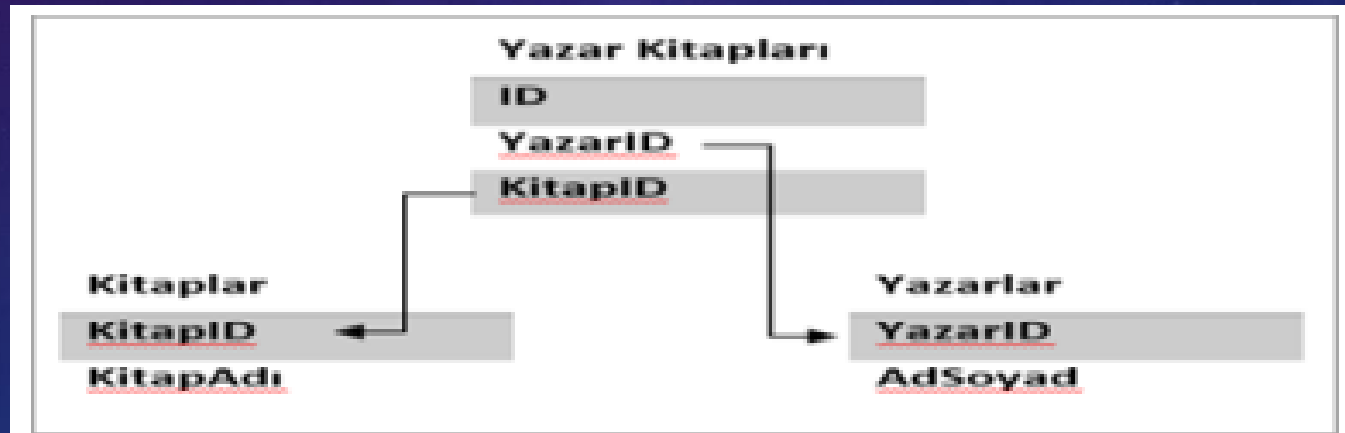
AĞ VERİ MODELİ

- Hiyerarşik veri modelinin geliştirilmiş halidir. Hızlıca kabul görmesinin nedeni bir verinin doğal olarak başka veriler ile ilişkili olmasıdır. Ağ modelinin hiyerarşik modelden en önemli farkı, uçdugum pozisyonundaki verinin iç-dügüme işaret edebilmesidir. Böylelikle ağ modelinde bire-çok ilişkiler yanında, çokaçok ilişkiler de modellenenebilir. Bu veri tekrarını önemli ölçüde azaltır



İLİŞKİSEL VERİ MODELİ

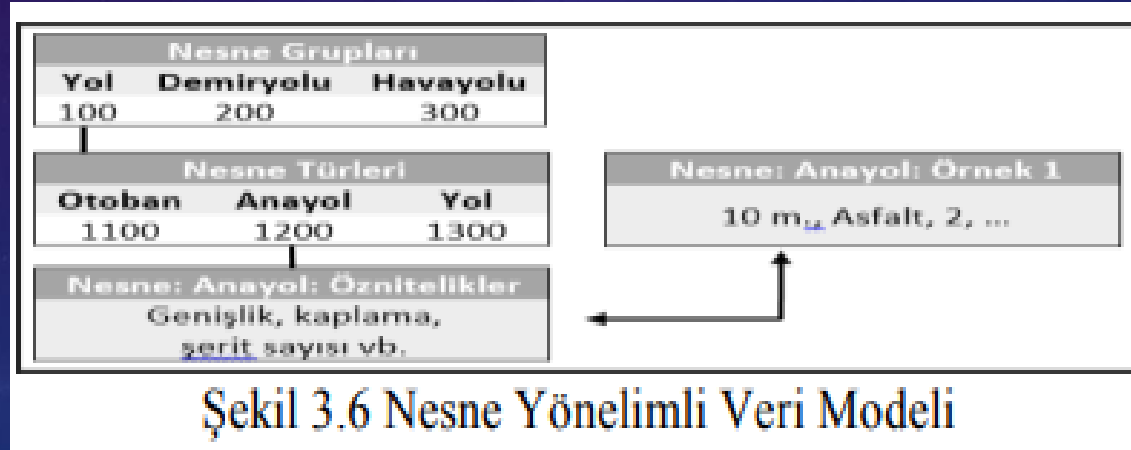
İlişkiler yardımıyla, veri içerisindeki ilişkiler modellenir. Dolayısıyla, ilişkisel bir veri tabanı, çeşitli ilişki örneklerinden oluşur. Kavramsal olarak ilişkiler, satır ve sütunlardan oluşan iki boyutlu tablolarla karakterize edilir.



Şekil 3.5 İlişkisel Veri Modeli

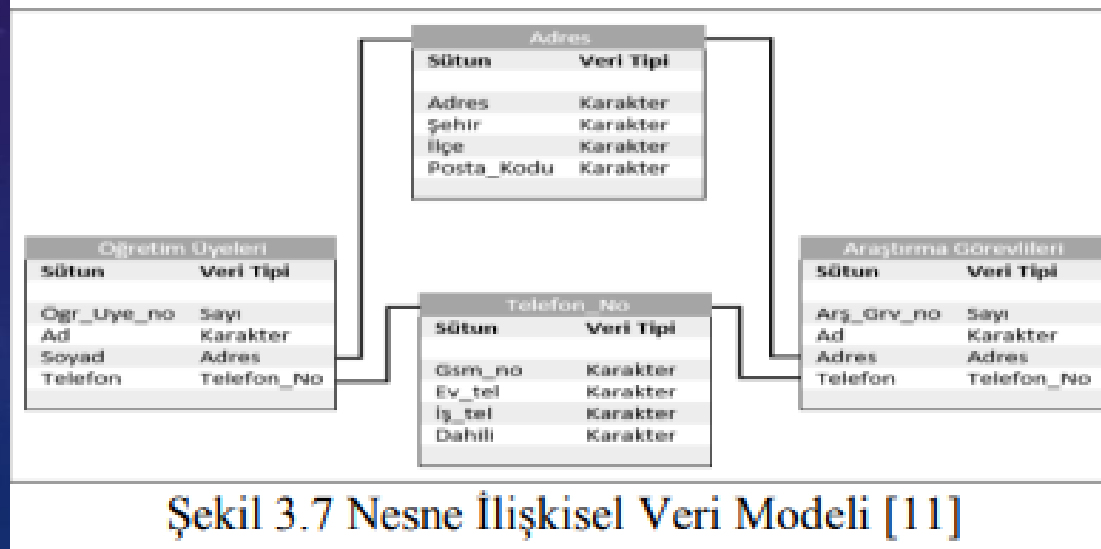
NESNE YÖNELİMLİ VERİ MODELİ

- Daha sonraları ortaya çıkmış ve başarısını kanıtlamıştır. Nesne yönelimli programlamaya dayanan veri modelidir [



NESNE İLİŞKİSEL VERİ MODELİ

- Nesne ilişkisel veri tabanı, ilişkisel işlevselliğin üzerine nesne yönelimli özellikler içerir. İlişkisel veri tabanları içinde nesne yönelimli karakteristikler içeren ilk veri tabanı 1997 yılında piyasaya sunulan Oracle8' dir



VERİ TABANI TASARIMI

- Veri tabanı tasarımında ilk olarak, olası veri tabanı kullanıcı gereksinimlerinin belirlenmesi gerekir. Söz konusu gereksinimler, veri tabanında yer alacak veri gruplarını, verilerin tiplerini ve verinin fiziksel olarak depolanması için kullanılacak olan veri yapılarını belirler.
- Geleneksel veri tabanı tasarımı, kullanıcı düzeyinden fiziksel düzeye doğrudur. Kavramsal tasarımda, gereksinimlere göre kavramsal şema belirlenir
- . Kavramsal şema, ortalama veri tabanı kullanıcısı için, veri tabanının yapısını genel olarak tanımlar.
- Kavramsal veri modelleri oldukça yüksek düzeyli olduklarından, kavramsal bir veri modelinde tanımlı bir şema genellikle doğrudan gerçekleştirilemez. Bu nedenle, geleneksel veri tabanı tasarımında, kavramsal tasarımdan sonraki adım, çoğunlukla, gerçekleştirim için kullanılacak bir veri tabanı yönetim sisteminin seçimidir.

İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NOSQL) VERİ TABANI SİSTEMLERİ

- Günümüzde en yaygın kullanılan veri tabanı sistemlerinden biridir. Satır ve sütunların meydana getirdiği tablolardan oluşur
- İlişkisel olmayan veri tabanları yatay olarak ölçeklendirilen bir veri depolama sistemidir
- NoSQL veri tabanları göreceli olarak yeni bir gelişmedir. Fakat e-ticaret, internet arama motorları ve sosyal ağlar gibi büyük ölçekli internet uygulamaları için güvenilirliğini kanıtlamıştır.

VERİTABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRMASI

- İlişkisel veri tabanı sistemlerine alternatif bir çözüm olarak ortaya çıkan, yatay olarak ölçeklendirilen bir veri depolama sistemi olan MongoDB veri tabanı sistemi kullanılmıştır.
- Yapılan çalışmada; MySQL ve MongoDB veri tabanı sistemlerinin performans ve yatay ölçeklenebilirlik incelemesi için aşağıdaki işlemlerin uygulanması ve sonuçlarının ortaya çıkarılması hedeflenmiştir.
- Veri silme işleminde ise MongoDB' nin MySQL ile benzer bir performansa sahip olduğu fakat veri silme komut sayılarının artışı ile MySQL veri tabanı sisteminin silme işleminde iyi bir performans sergilediği gözlemlenmiştir.

SONUÇ VE DEĞERLENDİRME

- Yönetim bilişim sistemleri kapsamında veri tabanlarının modellenmesi ve niteliklerinin belirlenmesi, en uygun performans ölçümleri, sürecin uygun hale getirilmesi ve en uygun veri tabanı seçiminde kullanıcılara ışık tutulması hedeflenmiştir
- Artan yazılım rekabetinde kullanıcının en önemli tercih kıstaslarından biri olan çalışma hızı dikkate alınarak, oldukça yaygın kullanım alanına sahip veri tabanı yönetim sistemlerinden MongoDB ve MySQL' in mümkün olduğunca eşit koşullarda işlem süreleri hesaplanarak performansları karşılaştırılmıştır
- Yapılan analizlerde NoSQL ağırlıklı bir veri tabanının büyük miktarda veri çiftleri içerebildiği, veri çoğaltmada da basit şeması nedeniyle MongoDB kullanılarak daha hızlı daha karmaşık sorgu tiplerinin çalıştırılabildiği izlenmiştir.
- Yapılan son performans testleri ise yazma ve silme işlemleridir. Basit arama sorgularında mantıksal olarak veri silme işlemi dikkate alınarak yapılan karşılaştırmalar sonucunda MySQL veri tabanı sistemi iyi bir performans göstermiştir
- Sonuç olarak, farklı kriterler ile bu veri tabanlarını incelediğimizde iki veri tabanının da avantaj ve dezavantajları olduğu görülmüştür