

## Data Warehouse (Veri Ambarı) Nedir?

Veri ambarı, daha iyi iş zekası için karşılaştırılıp analiz edilebilmesi veya daha fazla kaynaktan yapılandırılmış verileri toplayan bir teknolojidir. Mevcut ve geçmiş verileri, kuruluş genelinde çalışanlar için analitik raporlar oluşturur ve kullanılan tek bir yerde depolarlar. Verileri bilgiye dönüştürme ve fark yaratmak için zamanında kullanıcılara ulaştırma sürecidir.

Bir envanter sistemi için 3NF tasarımlı bir veritabanının çoğunun birbiriyle ilişkili tablolara sahip olduğunu birçoğunuz biliyorsunuz. Örneğin, mevcut envanter bilgileriyle ilgili bir rapor 12'den fazla birleştirilmiş koşulu içerebilir. Bu, sorgu ve raporun yanıt süresini hızla yavaşlatabilir. Bir veri ambarı, yanıt süresini azaltmaya yardımcı olabilecek ve raporlar ve analitik için sorguların performansını artırmaya yardımcı olabilecek yeni bir tasarım sağlar.



## Veri Ambarı Nasıl Çalışır?

Bir veri ambarı birden çok veritabanı içerebilir. Her veritabanı içinde, veriler tablolar ve sütunlar halinde düzenlenir. Her sütunun içinde, tamsayı, veri alanı veya dize gibi verilerin bir açıklamasını tanımlayabilirsiniz. Tablolar, klasörler olarak düşünebileceğiniz şemaların içinde düzenlenebilir. Veriler alındığında, şema tarafından açıklanan çeşitli tablolarda saklanır. Sorgu araçları, hangi veri tablolarına erişilip analiz edileceğini belirlemek için şemayı kullanır.

Veriler şunlar olabilir:

1. Yapılandırılmış
2. Yarı yapılandırılmış
3. Yapılandırılmamış veriler

Veri ambarı, veri madenciliğini mümkün kılar. Veri madenciliği, verilerde daha yüksek satış ve karlara yol açabilecek kalıplar arıyor.

## Veri ambarına kimin ihtiyacı var?

Aşağıdakiler gibi tüm kullanıcı türleri için veri ambarı gereklidir:

- Büyük miktarda veriye güvenen karar vericiler
- Birden çok veri kaynağından bilgi almak için özelleştirilmiş, karmaşık süreçler kullanan kullanıcılar.
- Ayrıca verilere erişmek için basit teknoloji isteyen kişiler tarafından da kullanılmaktadır.
- Karar vermek için sistematik bir yaklaşım isteyen kişiler için de gereklidir.
- Kullanıcı, raporlar, ızgaralar veya grafikler için zorunlu olan büyük miktarda veride hızlı performans istiyorsa, Veri ambarı yararlıdır.
- Veri ambarı, veri akışlarının ve gruplamaların 'gizli kalıplarını' keşfetmek istiyorsanız ilk adımdır.

## Veri Ambarı Ne İçin Kullanılır?

Veri ambarının kullanıldığı en yaygın sektörler şunlardır:

### Havayolu:

Havayolu sisteminde mürettebat atama, rota karlılık analizleri, sık uçan yolcu programı promosyonları vb. Operasyon amaçlı kullanılmaktadır.

**Bankacılık:**

Bankacılık sektöründe masa başı mevcut kaynakları etkin bir şekilde yönetmek için yaygın olarak kullanılmaktadır. Pazar araştırması, ürün ve işlemlerin performans analizi için çok az banka da kullandı.

**Sağlık hizmeti:**

Sağlık sektörü ayrıca, sonuçları strateji oluşturmak ve tahmin etmek, hastanın tedavi raporlarını oluşturmak, bağlantılı sigorta şirketleri, tıbbi yardım hizmetleri vb. ile veri paylaşmak için Veri ambarını kullandı.

**Kamu sektörü:**

Kamu sektöründe istihbarat toplama için veri ambarı kullanılmaktadır. Devlet kurumlarının her birey için vergi kayıtlarını, sağlık politikası kayıtlarını tutmasına ve analiz etmesine yardımcı olur.

**Yatırım ve Sigorta sektörü:**

Bu sektörde depolar öncelikle veri modellerini, müşteri eğilimlerini analiz etmek ve piyasa hareketlerini izlemek için kullanılır.

**Zincir tutma:**

Perakende zincirlerinde Veri ambarı, dağıtım ve pazarlama için yaygın olarak kullanılmaktadır. Ayrıca öğeleri, müşteri satın alma modelini, promosyonları takip etmeye yardımcı olur ve ayrıca fiyatlandırma politikasını belirlemek için kullanılır.

**Telekomünikasyon:**

Bu sektörde ürün promosyonları, satış kararları ve dağıtım kararları vermek için bir veri ambarı kullanılmaktadır.

**Konaklama Sektörü:**

Bu Sektör, müşterileri geri bildirimlerine ve seyahat modellerine göre hedeflemek istedikleri reklam ve promosyon kampanyalarını tasarlamak ve tahmin etmek için depo hizmetlerini kullanır.

## Veri Ambarını Uygulama Adımları

Bir Datawarehouse uygulamasıyla ilişkili iş riskini ele almanın en iyi yolu, aşağıdaki gibi üç uçlu bir strateji uygulamaktır

1. **Kurumsal strateji:** Burada, mevcut mimari ve araçlar dahil olmak üzere teknik özellikleri belirleriz. Ayrıca gerçekleri, boyutları ve nitelikleri de belirleriz. Veri eşleme ve dönüştürme de geçerli.
2. **Aşamalı teslimat:** Veri ambarı uygulaması konu alanlarına göre aşamalı olmalıdır. Rezervasyon ve faturalama gibi ilgili ticari varlıklar önce uygulanmalı ve ardından birbirleriyle entegre edilmelidir.
3. **Yinelemeli Prototipleşme:** Uygulamaya büyük patlama yaklaşımı yerine, Datawarehouse yinelemeli olarak geliştirilmeli ve test edilmelidir.

Burada, Datawarehouse uygulamasının çıktıları ile birlikte önemli adımlar.

Adım	Görevler	Teslimat
1	Proje kapsamını tanımlamanız gerekiyor	Kapsam tanımı
2	İş ihtiyaçlarını belirleme ihtiyacı	Mantıksal Veri Modeli
3	Operasyonel Veri Deposu gereksinimlerini tanımlayın	Operasyonel Veri Deposu Modeli
4	Çıkarma araçları edinin veya geliştirin	Araçları ve Yazılımları Çıkarın
5	Veri Ambarı Veri gereksinimlerini tanımlayın	Geçiş Verisi Modeli
6	Eksik verileri belgeleyin	Yapılacaklar Listesi
7	Operasyonel Veri Deposunu Veri Ambarı ile Eşler	D / W Veri Entegrasyon Haritası
8	Veri Ambarı Veritabanı tasarımı geliştirin	D / W Veritabanı Tasarımı
9	Operasyonel Veri Deposundan Veri Çıkarın	Entegre D / W Veri Özütləri
10	Veri Ambarını Yükle	İlk Veri Yükleme
11	Veri Ambarı Bakımı	Devam Eden Veri Erişimi ve Sonraki Yükleme

## Veri Ambarı uygulamak için en iyi uygulamalar

- Verilerin tutarlılığını, doğruluğunu ve bütünlüğünü test etmek için bir plan belirleyin.
- Veri ambarı iyi entegre edilmiş, iyi tanımlanmış ve zaman damgalı olmalıdır.
- Datawarehouse tasarlarken doğru aracı kullandığınızdan, yaşam döngüsüne bağlı kaldığınızdan, veri uyumsuzluklarına dikkat ettiğinizden ve hatalarınızı öğrenmeye hazır olduğunuzdan emin olun.
- Operasyonel sistemleri ve raporları asla değiştirmeyin
- Verileri çıkarmak, temizlemek ve yüklemek için çok fazla zaman harcamayın.
- Datawarehouse uygulama sürecine iş personeli dahil tüm paydaşların dahil edilmesini sağlayın. Veri ambarının bir ortak / ekip projesi olduğunu belirleyin. Son kullanıcılar için yararlı olmayan Veri ambarı oluşturmak istemezsiniz.
- Son kullanıcılar için bir eğitim planı hazırlayın.

# Veri Ambarına Neden İhtiyacımız Var? Avantajlar dezavantajlar

## Veri Ambarı Avantajları:

- Veri ambarı, iş kullanıcılarının bazı kaynaklardaki kritik verilere tek bir yerden hızla erişmesine olanak tanır.
- Veri ambarı, çeşitli işlevler arası etkinlikler hakkında tutarlı bilgiler sağlar. Ayrıca anlık raporlama ve sorgulamayı da destekliyor.
- Veri Ambarı, üretim sistemindeki stresi azaltmak için birçok veri kaynağını entegre etmeye yardımcı olur.
- Veri ambarı, analiz ve raporlama için toplam geri dönüş süresini azaltmaya yardımcı olur.
- Yeniden Yapılandırma ve Entegrasyon, kullanıcının raporlama ve analiz için kullanmasını kolaylaştırır.
- Veri ambarı, kullanıcıların çok sayıda kaynaktan kritik verilere tek bir yerden erişmesine olanak tanır. Bu nedenle, kullanıcının birden çok kaynaktan veri alma zamanından tasarruf sağlar.
- Veri ambarı, büyük miktarda geçmiş veriyi depolar. Bu, kullanıcıların gelecekteki tahminlerde bulunmak için farklı zaman dönemlerini ve eğilimleri analiz etmelerine yardımcı olur.

## Veri Ambarının dezavantajları:

- Yapılandırılmamış veriler için ideal bir seçenek değil.
- Veri Ambarının Oluşturulması ve Uygulanması kesinlikle zaman kafa karıştırıcı bir olaydır.
- Veri Ambarı nispeten hızlı bir şekilde eski hale gelebilir
- Veri türleri ve aralıklarında, veri kaynağı şemasında, izinlerde ve sorgularda değişiklik yapmak zordur.
- Veri ambarı kolay görünebilir, ancak aslında ortalama kullanıcılar için çok karmaşık.
- Proje yönetimindeki en iyi çabalara rağmen, veri ambarı proje kapsamı her zaman artacaktır.
- Bazen depo kullanıcıları farklı iş kuralları geliştirecektir.
- Kuruluşlar, eğitim ve Uygulama amacıyla kaynaklarının çoğunu harcamalıdır.

## Veri Ambar Iamanın Geleceęi

- **Düzenleyici kısıtlamalardaki** deęişiklik, farklı verilerin kaynaęını birleştirme yeteneęini sınırlandırabilir. Bu farklı kaynaklar, saklanması zor olan yapılandırılmamış verileri içerebilir.
- As **boyut** veritabanlarının büyür, çok büyük bir veritabanı içerięine ilişkin tahminler büyümeye devam ediyor. Her zaman boyutu artan veri ambarı sistemleri oluşturmak ve çalıştırmak karmaşıktır. Günümüzde mevcut olan donanım ve yazılım kaynakları, büyük miktarda verinin çevrimiçi tutulmasına izin vermemektedir.
- **Multimedya verileri**, metin verileri olarak kolayca manipüle edilemezken, metin bilgileri bugün mevcut olan ilişkisel yazılımla alınabilir. Bu bir araştırma konusu olabilir.

## Veri Ambarı Araçları

Piyasada birçok Veri Ambarı aracı bulunmaktadır. İşte en öne çıkanlardan bazıları:

### 1. MarkLogic:

MarkLogic, bir dizi kurumsal özellięi kullanarak veri entegrasyonunu daha kolay ve daha hızlı hale getiren kullanışlı bir veri ambarı çözümüdür. Bu araç, çok karmaşık arama işlemlerinin gerçekleştirilmesine yardımcı olur. Belgeler, ilişkiler ve meta veriler gibi farklı veri türlerini sorgulayabilir.

<https://developer.marklogic.com/products/>

### 2. Oracle:

Oracle, sektör lideri veri tabanıdır. Hem şirket içi hem de bulut için çok çeşitli veri ambarı çözümleri sunar. Operasyonel verimlilięi artırarak müşteri deneyimlerini optimize etmeye yardımcı olur.

<https://www.oracle.com/index.html>

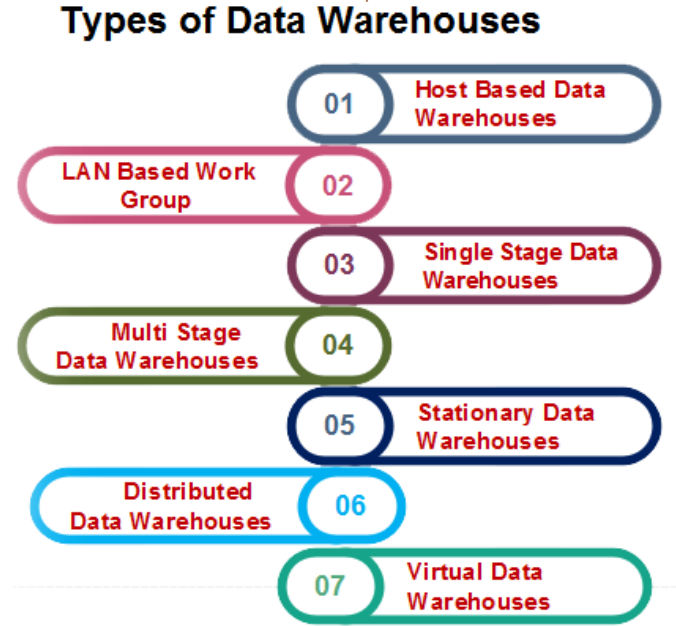
### 3. Amazon RedShift:

Amazon Redshift, Veri ambarı aracıdır. Standart SQL ve mevcut BI araçlarını kullanarak tüm veri türlerini analiz etmek için basit ve uygun maliyetli bir araçtır. Ayrıca, sorgu optimizasyonu teknięini kullanarak pet baytlarca yapılandırılmış veriye karşı karmaşık sorguların çalıştırılmasına izin verir.

[https://aws.amazon.com/redshift/?nc2=h\\_m1](https://aws.amazon.com/redshift/?nc2=h_m1)

İşte yararlı [Datawarehouse Araçlarının](#) tam listesi .

## Veri Ambarı Türleri



### Host-Based Data Warehouses

#### (Ana Bilgisayar Tabanlı Veri Ambarları)

Uygulanabilecek iki tür ana bilgisayar tabanlı veri ambarı vardır:

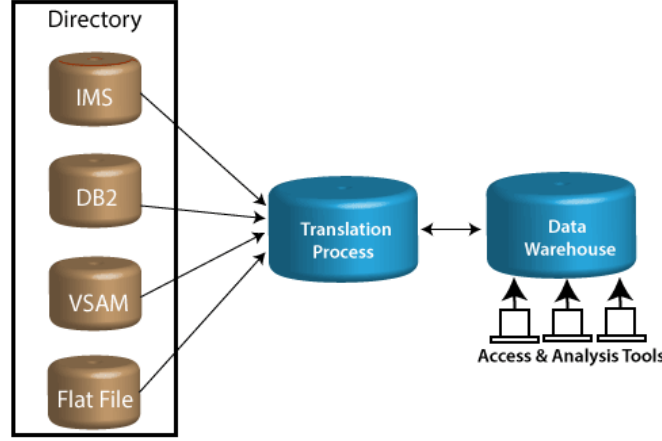
- Yüksek hacimli bir veritabanında bulunan ana bilgisayar tabanlı ana bilgisayar depoları. IBM system / 390, UNISYS ve Data General Sequent gibi sağlam ve güvenilir yüksek kapasiteli yapılardır. Sybase, Oracle, Informix ve DB2 gibi veritabanları tarafından desteklenir.
- Veri dağıtımının merkezi olarak veya çalışma grubu ortamından yapılabildiği veri ambarlarıdır. Veritabanının veri ambarlarının boyutu platforma bağlıdır.

Veri Çıkarma ve dönüştürme araçları, üretim sistemlerinden verilerin otomatik olarak çıkarılmasına ve temizlenmesine olanak tanır. Aşağıdaki nedenlerden ötürü, bu yöntem kategorilerine sorgu araçlarıyla doğrudan erişim sağlamak geçerli değildir:

1. Büyük bir karmaşık ambarlama sorguları yükü, görev açısından kritik işlem işleme (TP) odaklı uygulama üzerinde muhtemelen çok fazla zararlı etkiye sahip olacaktır.
2. Bu TP sistemleri, işlem hacmi için veritabanı tasarımlarında geliştirilmektedir. Tüm yöntemlerde, bir veritabanı, optimum sorgu veya işlem işleme için tasarlanmıştır. Karmaşık bir iş sorgusu, birçok normalleştirilmiş tablonun birleştirilmesini gerektirdi ve sonuç olarak performans genellikle zayıf olacak ve sorgu büyük ölçüde karmaşık olacaktır.
3. İki veya daha fazla üretim yöntemindeki verilerin tutarlı olacağının garantisi yoktur.

## Host-Based (MVS) Data Warehouses

### (Ana Bilgisayar Tabanlı (MVS) Veri Ambarları)



**Host Based (MVS) Data Warehouse**

MVS üzerindeki büyük hacimli veritabanlarında bulunan bu veri ambarı, ana bilgisayar tabanlı veri ambarları türleridir. Genellikle DBMS, VSAM, DB2, düz dosyalar ve Bilgi Yönetim Sistemi (IMS) dahil olmak üzere eski bilgiler için çok çeşitli orijinal kaynaklara sahip DB2'dir.

Böyle bir ambar tasarlamaya, inşa etmeye ve uygulamaya başlamadan önce, bazı ek hususlar verilmelidir, çünkü

1. Bu tür veritabanları genellikle çok yüksek hacimli veri depolamasına sahiptir.
2. Bu tür ambarlar hem MVS hem de müşteri bazlı rapor ve sorgu olanakları için destek gerektirebilir.
3. Bu depoların karmaşık kaynak sistemleri vardır.
4. Bu tür sistemler, görev açısından kritik hedefler için de kullanılmaları gerektiğinden, sürekli bakıma ihtiyaç duyuyordu.

Bu tür veri ambarlarının başarılı bir şekilde kurulmasını sağlamak için genellikle aşağıdaki aşamalar izlenir:

1. **Unload Phase:** İşlem verilerinin seçilmesini ve temizlenmesini içerir.
2. **Transform Phase:** Bunu uygun bir forma çevirmek ve ona erişim ve saklama kurallarını açıklamak için kullanılır.
3. **Load Phase:** Kaydı, başka bir veritabanına ya da MVS olmayan ambarlara taşımak üzere doğrudan DB2 tablolarına veya belirli bir dosyaya taşımak içindir.

Entegre bir Metadata havuzu, herhangi bir veri ambarı ortamının merkezidir. Böyle bir tesis, veri kaynaklarını, veri çeviri kurallarını ve kullanıcı alanlarını depoya belgelemek için gereklidir. Birden çok veri kaynağı veritabanı ile koşullu veri ambarlarının DB2'si arasında dinamik bir ağ sağlar.



## ***DB2 Nedir?***

DB2, IBM tarafından geliştirilen ve pazarlanan ilişkisel veritabanı sunucuları da dahil olmak üzere ilgili veri yönetimi ürünleri ailesidir.

IBM, 1970'lerden bu yana, Sanal Makine (VM), Genişletilmiş Sanal Depolama (VSE) ve Çoklu Sanal Depolama (MVS) gibi ana bilgisayar platformlarında başlatılan eksiksiz bir veritabanı sunucusu ailesi geliştirmiştir.

Kaynak: <https://www.db2tutorial.com/getting-started/what-is-db2/>

## ***Host-Based (UNIX) Data Warehouses***

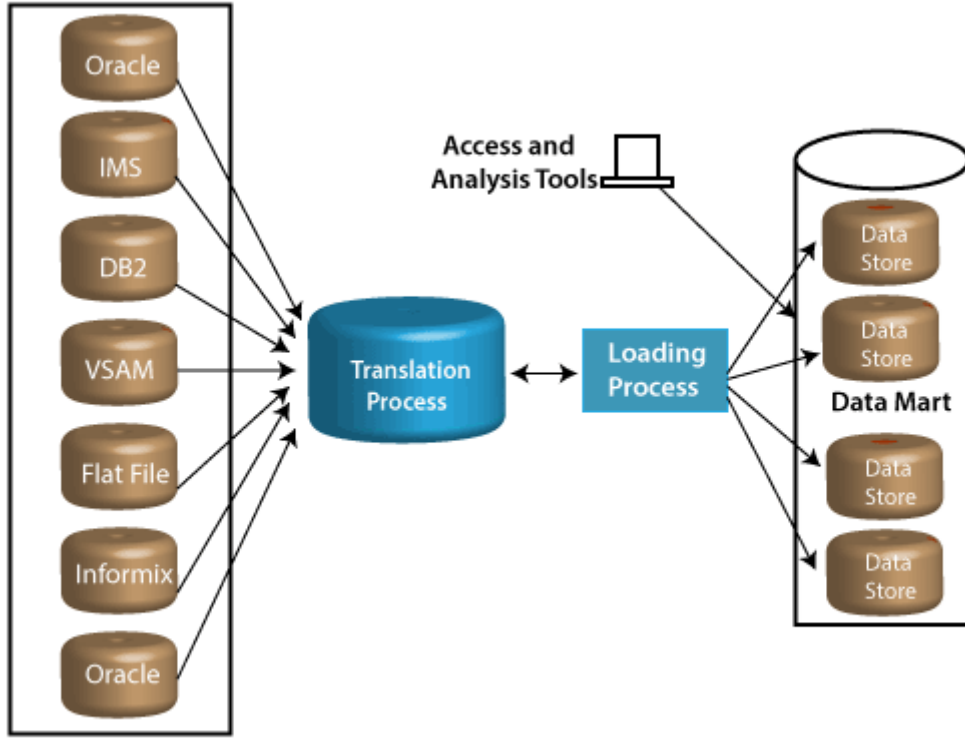
### ***(Ana Bilgisayar Tabanlı (UNIX) Veri Ambarları)***

Oracle ve Informix RDBMS'ler, bu tür veri ambarları için tesisleri destekler. Bu veri tabanlarının her ikisi de MVS- tabanlı veri tabanlarından ve daha yüksek sayıda diğer UNIX- tabanlı veri tabanlarından bilgi çıkarabilir. Bu tür ambarlar, ana bilgisayar tabanlı MVS veri ambarlarıyla aynı aşamayı izler. Ayrıca farklı ağ sunucularından gelen veriler de oluşturulabilir.

## ***LAN-Based Workgroup Data Warehouses***

### ***(LAN Tabanlı Çalışma Grubu Veri Ambarları)***

LAN tabanlı bir çalışma grubu deposu, bir LAN ortamında bir veri ambarı oluşturmak ve sürdürmek için entegre bir yapıdır. Bu depoda, çeşitli kaynaklardan bilgi alabilir ve birden çok LAN tabanlı depoyu, genellikle DB2 ailesi, Oracle, Sybase ve Informix'i içerecek şekilde seçilen depo veritabanlarını destekleyebiliriz. Seyrek olarak içerilen diğer veritabanları IMS, VSAM, Düz Dosya, MVS ve VH'dir.



### LAN Based Work Group Warehouse

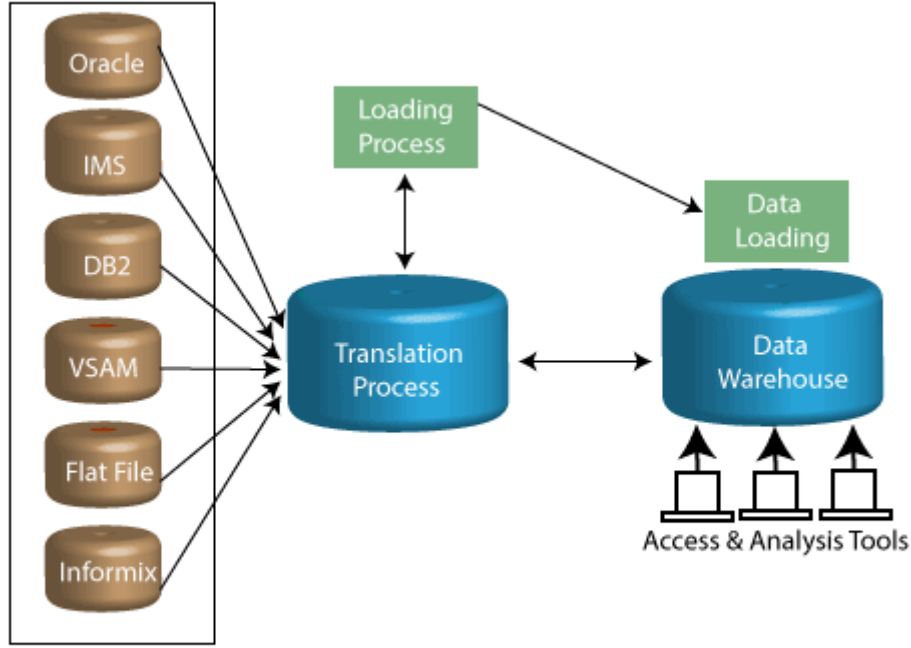
Çalışma grubu ortamı için tasarlanmış, LAN tabanlı bir çalışma grubu deposu, genellikle veri pazarı olarak adlandırılan bir veri ambarı oluşturmak isteyen tüm iş organizasyonları için idealdir. Bu tür bir veri ambarı genellikle minimum bir ilk yatırım ve teknik eğitim gerektirir.

**Data Delivery (Veri Sağlama):** LAN tabanlı bir çalışma grubu deposu ile müşteri, departman, iş birimi veya çalışma grubu düzeyinde kullanım için özelleştirilmiş bir veri deposu oluşturmak ve sürdürmek için minimum teknik bilgiye ihtiyaç duyar. LAN tabanlı bir çalışma grubu deposu, depodaki verilere taşıma erişimi sağlayarak kurumsal kaynaklardan bilgi teslimini sağlar.

### *Host-Based Single Stage (LAN) Data Warehouses*

#### *(Ana Bilgisayar Tabanlı Tek Aşamalı (LAN) Veri Ambarları)*

LAN tabanlı bir veri ambarı içinde, veri teslimi merkezi olarak veya çalışma grubu ortamından gerçekleştirilebilir, böylece iş grupları merkezi BT kaynaklarını zorlamadan ihtiyaç duydukları verileri işlemeyi karşılayabilir ve genel veri bütünlüğü ve güvenlikten ödün vermeden veri pazarlarının özerkliğinin keyfini çıkarabilir.



## LAN Based Single Stage Warehouse

### Sınırlamalar

Hem DBMS hem de donanım ölçeklenebilirlik yöntemleri genellikle LAN tabanlı ambarlama çözümlerini sınırlar.

Birçok LAN tabanlı işletme, sağlam depolama çözümleri sağlamak için yeterli iş planlama, kurtarma yönetimi, organize bakım ve performans izleme yöntemleri uygulamamıştır.

Genellikle bu depolar, kaynak kaydı için diğer platformlara bağımlıdır. Veri bütünlüğü, kurtarılabirlik ve güvenliğe sahip bir ortam oluşturmak dikkatli tasarım, planlama ve uygulama gerektirir. Aksi takdirde, kaynaklardan sunucuya dönüşüm ve yüklerin senkronizasyonu sayısız soruna neden olabilir.

Bir **LAN tabanlı depo** en az bir ilk yatırım ve teknik bilgi gerektiren birçok kaynaktan gelen verileri sağlar. LAN tabanlı bir depo, veri ambarını doldurmak ve güncellemek için replikasyon araçlarını da çalıştırabilir. Bu tür bir ambar, işletme görünümünü, geçmişleri, toplama, sürümleri ve heterojen kaynak desteğini içerebilir.

- DB2 Ailesi
- IMS, VSAM, Düz Dosya [MVS ve VM]

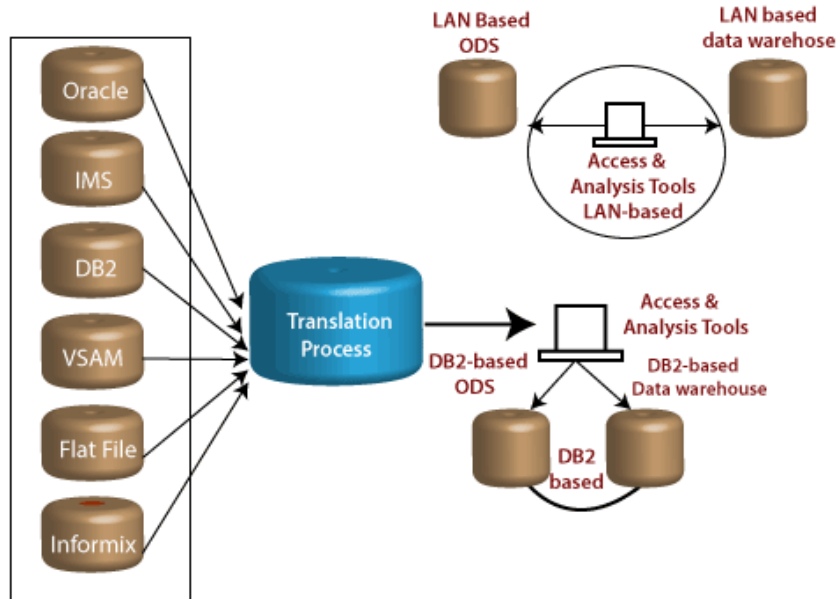
Tek bir mağaza genellikle LAN tabanlı bir depoyu çalıştırır ve mevcut DSS uygulamalarını sağlayarak iş kullanıcısının veri ambarındaki verileri bulmasını sağlar. LAN tabanlı depo, iş kullanıcılarını eksiksiz veriden bilgiye çözüm çözümüyle destekleyebilir. LAN tabanlı depo, iş verilerini kataloglama ve ihtiyaç duyan herkes için uygun hale getirme yeteneği ile meta verileri de paylaşabilir.

## Multi-Stage Data Warehouses

### (Çok Aşamalı Veri Ambarları)

Toplama yoluyla verileri analiz etmek için dönüştürme yöntemlerinde birden çok aşamayı ifade eder. Başka bir deyişle, verilerin veri ambarına yüklenmesinden önce birden çok kez aşamalandırılması. Veriler ilk olarak kaynak sistemlerinden aşamalandırma alanına çıkarılır, ardından değişiklikten sonra veri ambarına ve son olarak departmanlaştırılmış veri reyonlarına yüklenir.

Bu konfigürasyon, çeşitli kapasitelerdeki son müşterilerin hem dakikaya kadar taktik kararlar için özetlenmiş bilgilere hem de uzun vadeli stratejik kararlar için değişmeli bir kayıt olan özetlenmiş bilgilere erişime ihtiyaç duyduğu ortamlar için çok uygundur. Hem Operasyonel Veri Deposunda (ODS) hem de hacme ve özel gereksinimlere bağlı olarak ana bilgisayar tabanlı veya LAN Tabanlı veritabanlarında bulunabilir. Bunlar DB2, Oracle, Informix, IMS, Düz Dosyalar ve Sybase içerir.

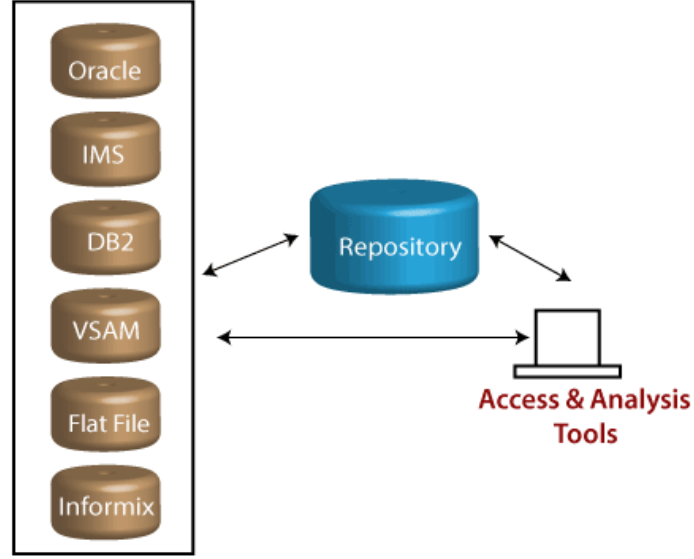


**Multistage Data Warehouse**

ODS yalnızca en güncel kayıtları saklar. Veri ambarı, dosyaların geçmiş hesaplamalarını depolar. İlk başta, her iki veritabanındaki bilgiler çok benzer olacaktır. Örneğin, yeni bir müşterinin kayıtları aynı görünecektir. Kullanıcı kaydında değişiklikler meydana geldikçe, OD'ler yalnızca en güncel verileri yansıtacak şekilde yenilenirken, veri ambarı hem geçmiş verileri hem de yeni bilgileri içerecektir. Bu nedenle, veri ambarının hacim gereksinimi ODS fazla mesainin hacim gereksinimlerini aşacaktır. Pratikte 4'e 1 oranına ulaşmak aşına değildir.

## Stationary Data Warehouses

### (Sabit Veri Ambarları)



**Stationary Data Warehouse**

Bu tür veri ambarlarında, veriler şekilde gösterildiği gibi kaynaklardan değiştirilmez:

Bunun yerine müşteriye verilere doğrudan erişim hakkı verilir. Pek çok kuruluş için, seyrek erişim, hacim sorunları veya kurumsal gereklilikler gibi yaklaşımları belirler. Bu şema, müşteri için aşağıdakiler gibi birkaç sorun yaratır:

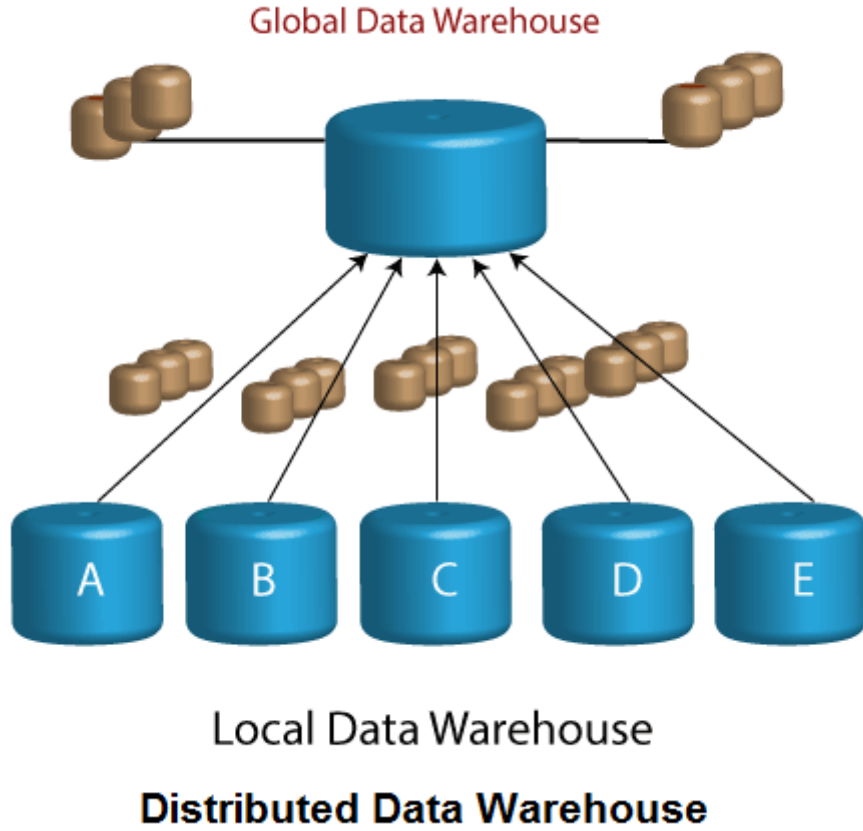
- Kullanıcılar için bilgilerin yerinin belirlenmesi
- Müşterilere, hepsi tek bir API ile tek bir DBMS olduğu gibi farklı DBMS'leri sorgulama yeteneği sağlar.
- Müşteri, üretim veri depoları ile rekabet edeceği için performansı etkiler.

Böyle bir ambar, muhtemelen müşteriyle tek bir etkileşimle son derece uzmanlaşmış ve sofistike 'ara katman yazılımına' ihtiyaç duyacaktır. Bu aynı zamanda bir tesisin, rapor oluşturmadan önce kullanıcı için çıkarılan kaydı görüntülemesi için gerekli olabilir. Entegre bir meta veri havuzu, bu ortamda mutlak bir gereklilik haline gelir.

## *Distributed Data Warehouses*

### *(Dağıtılmış Veri Ambarları)*

Dağıtılmış bir veri ambarı kavramı, iki tür dağıtılmış veri ambarı ve bunların kuruluş genelinde dağıtılan yerel işletme ambarları ve şekilde gösterildiği gibi küresel bir ambarlar için modifikasyonları olduğunu göstermektedir:



### *Yerel veri ambarlarının özellikleri*

- Etkinlik yerel düzeyde görünür
- Operasyonel işleminin toplu
- Yerel site özektir
- Her yerel veri ambarının kendine özgü mimarisi ve veri içerikleri vardır
- Veriler benzersizdir ve yalnızca bu yerellik için çok önemlidir
- Kaydın çoğunluğu yereldir ve kopyalanmamıştır
- Yerel veri ambarları arasındaki herhangi bir veri kesişimi durumsaldır
- Yerel depo farklı teknik topluluklara hizmet eder
- Yerel veri ambarlarının kapsamı yerel siteyle sınırlıdır
- Yerel depolar ayrıca geçmiş verileri içerir ve yalnızca yerel site içinde entegre edilir.

## Virtual Data Warehouses

### (Sanal Veri Ambarları)

Sanal Veri Ambarları aşağıdaki aşamalarda oluşturulur:

1. Bir dizi veri yaklaşımı, veri sözlüğü ve süreç yönetimi olanağı kurma.
2. Son müşterileri eğitmek.
3. DW tesislerinin nasıl kullanılacağını izlenmesi
4. Gerçek kullanıma dayalı olarak, yüksek frekanslı sonuçlar sağlamak için fiziksel olarak Veri Ambarı oluşturulur

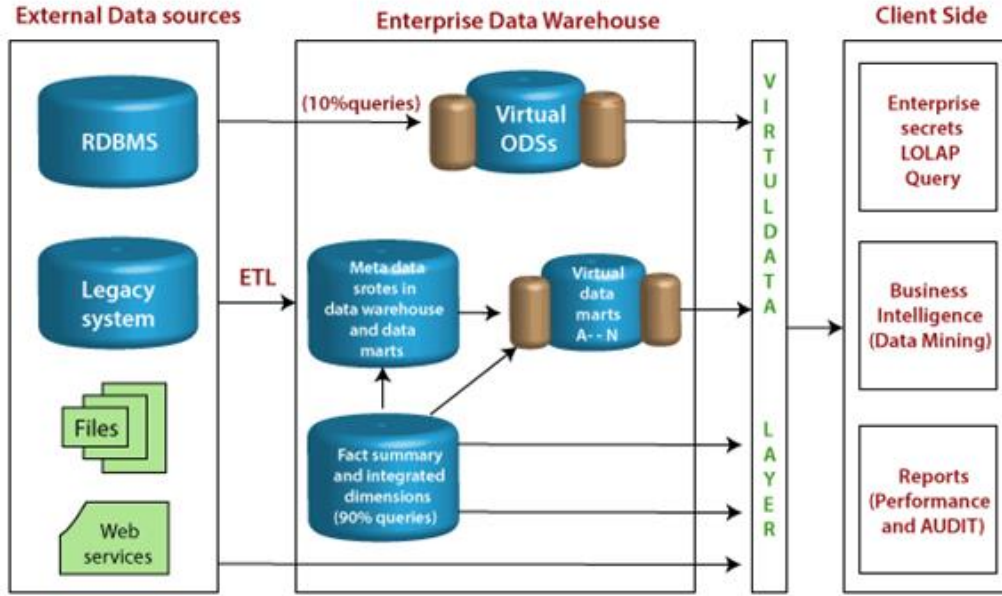
Bu strateji, son kullanıcıların, veri erişim ağına uygulanan araçları kullanarak doğrudan operasyonel veritabanlarına girmelerine izin verildiğini tanımlar. Bu yöntem, nihai esnekliğin yanı sıra yüklenmesi ve bakımı yapılması gereken minimum miktarda fazlalık bilgi sağlar. Veri ambarı harika bir fikirdir, ancak inşa edilmesi zordur ve yatırım gerektirir. Neden meta veri ve başka bir veritabanı için depoların dönüşüm aşamasını ortadan kaldırarak ucuz ve hızlı bir yöntem kullanmıyorsunuz? Bu yönteme ' **Virtual Data Warehouse** ' denir.

Bunu başarmak için dört tür veriyi tanımlamaya ihtiyaç vardır:

1. Çeşitli veritabanlarının tanımlarını içeren bir veri sözlüğü.
2. Veri bileşenleri arasındaki ilişkinin açıklaması.
3. Yöntemin açıklaması, kullanıcı sistem ile arayüz oluşturacaktır.
4. Ne yapılacağını ve nasıl yapılacağını açıklayan algoritmalar ve iş kuralları.

### Dezavantajları

1. Sorgular, üretim kaydı işlemleriyle rekabet ettiğinden, performans düşebilir.
2. Meta veri, özet kayıt veya bireysel **DSS** (Karar Destek Sistemi) entegrasyonu veya geçmişi yoktur. Tüm sorgular kopyalanmalıdır ve sisteme ek bir yük getirir.
3. Sorguların çok karmaşık olmasına neden olan yenileme süreci yoktur.



10% of user queries are fired on fact summary & 90% of user queries are fired on ODSs

## Virtual Data Warehouse

## Veri Ambarının Genel Aşamaları

Önceden kuruluşlar veri ambarını nispeten basit bir şekilde kullanıyorlardı. Zamanla şirketler daha aktif kullanmaya başladıkları için veri ambarları daha karmaşık bir hal aldı.

Veri ambarlarının genel kullanım aşamaları:

### Offline Operational Database

#### (Çevrimdışı Operasyonel Veritabanı):

Bu aşamada, veriler sadece bir işletim sisteminden başka bir sunucuya kopyalanır. Bu şekilde, kopyalanan verilerin yüklenmesi, işlenmesi ve raporlanması operasyonel sistemin performansını etkilemez.

### Offline Data Warehouse

#### (Çevrimdışı Veri Ambarı):

Veri ambarındaki veriler Operasyonel Veritabanından düzenli olarak güncellenir. Veriler, hedeflerini karşılamak için haritalanır ve dönüştürülür.



## **Real time Data Warehouse**

### **(Gerçek Zamanlı Veri Ambarı):**

Bu aşamada, operasyonel veri tabanında herhangi bir işlem gerçekleştiğinde Veri ambarları güncellenir. Örneğin, Havayolu veya demiryolu rezervasyon sistemi.

## **Integrated Data Warehouse**

### **(Entegre Veri Ambarı):**

Bu aşamada Veri Ambarları, operasyonel bir işlem gerçekleştirdiğinde sürekli güncellenir. Veri Ambarı daha sonra operasyonel sisteme geri aktarılan işlemleri üretir.

## **Veri Ambarının Bileşenleri**

Veri ambarının dört bileşeni şunlardır:

**Load Manager (Yük yöneticisi):** Yük yöneticisi, ön bileşen olarak da adlandırılır. Verilerin çıkarılması ve depoya yüklenmesi ile ilgili tüm işlemleri gerçekleştirir. Bu işlemler, verileri Veri ambarına girmek üzere hazırlamak için dönüşümleri içerir.

**Warehouse Manager (Depo Yöneticisi):** Depo yöneticisi, depodaki verilerin yönetimi ile ilgili işlemleri gerçekleştirir. Tutarlılığı sağlamak için veri analizi, dizinlerin ve görünümünün oluşturulması, normalden arındırma ve toplamaların oluşturulması, kaynak verilerin dönüştürülmesi ve birleştirilmesi, verilerin arşivlenmesi ve pişirilmesi gibi işlemleri gerçekleştirir.

**Query Manager (Sorgu Yöneticisi):** Sorgu yöneticisi, arka uç bileşeni olarak da bilinir. Kullanıcı sorgularının yönetimi ile ilgili tüm operasyon işlemlerini gerçekleştirir. Bu Veri ambarı bileşenlerinin işlemleri, sorguların yürütülmesini planlamak için uygun tablolara doğrudan sorgulardır.

### **Son kullanıcı erişim araçları:**

Araçlar beş farklı gruba ayrılmıştır:

1. Veri Raporlama
2. Sorgu Araçları
3. Uygulama geliştirme araçları
4. EIS araçları
5. OLAP araçları ve veri madenciliği araçları

# Veritabanı VS Veri Ambarı: Temel Farklar

## Veritabanı Nedir?

Veritabanı, gerçek dünyanın bazı unsurlarını temsil eden ilgili verilerin bir koleksiyonudur. Belirli bir görev için verilerle oluşturulacak ve doldurulacak şekilde tasarlanmıştır. Aynı zamanda veri çözümünüzün yapı taşıdır.

## Veri Ambarı Nedir?

Bir veri ambarı, tekli veya çoklu kaynaklardan geçmiş ve değişmeli verileri depolayan bir bilgi sistemidir. Farklı kaynaklardan gelen işlem verilerini analiz etmek, raporlamak, entegre etmek için tasarlanmıştır.

## Temel Farklar

- Veritabanı, gerçek dünyanın bazı unsurlarını temsil eden ilgili verilerin bir koleksiyonudur, oysa Veri ambarı, tekli veya çoklu kaynaklardan geçmiş ve değişmeli verileri depolayan bir bilgi sistemidir.
- Veritabanı verileri kaydetmek için tasarlanırken, Veri ambarı verileri analiz etmek için tasarlanmıştır.
- Veritabanı, uygulamaya yönelik veri toplama, Veri Ambarı ise konu odaklı veri toplama yöntemidir.
- Veritabanı Çevrimiçi İşlemse İşleme (OLTP) kullanırken Veri ambarı Çevrimiçi Analitik İşleme (OLAP) kullanır.
- Veritabanı tabloları ve birleştirmeleri, normalleştirildikleri için karmaşıktır, oysa Veri Ambarı tabloları ve birleştirmeleri normalleştirildikleri için kolaydır.
- Veri ambarını tasarlamak için ER modelleme teknikleri kullanılırken, Veri Ambarını tasarlamak için veri modelleme teknikleri kullanılır.

## Neden Veritabanı Kullanılmalı?

Veritabanı sistemini kullanmanın başlıca nedenleri şunlardır:

- Verilerin güvenliğini ve erişimini sunar
- Bir veritabanı, verileri depolamak ve almak için çeşitli teknikler sunar.
- Veritabanı, aynı verileri kullanan birden çok uygulamanın gereksinimlerini dengelemek için verimli bir işleyici görevi görür

- Bir DBMS, yasaklanmış verilere erişimi önlemek için yüksek düzeyde koruma elde etmek için bütünlük kısıtlamaları sunar.
- Bir veritabanı, aynı anda yalnızca tek bir kullanıcının aynı verilere erişebileceği şekilde eşzamanlı verilere erişmenizi sağlar.

## Neden Veri Ambarı Kullanılmalı?

Veri Ambarı kullanmanın önemli nedenleri şunlardır:

- Veri ambarı, iş kullanıcılarının bazı kaynaklardaki kritik verilere tek bir yerden erişmesine yardımcı olur.
- Çeşitli işlevler arası etkinlikler hakkında tutarlı bilgi sağlar
- Üretim sistemi üzerindeki stresi azaltmak için birçok veri kaynağını entegre etmenize yardımcı olur.
- Veri ambarı, analiz ve raporlama için TT'yi (toplam geri dönüş süresi) azaltmanıza yardımcı olur.
- Veri ambarı, kullanıcıların farklı kaynaklardan kritik verilere tek bir yerden erişmesine yardımcı olur, böylece kullanıcının birden çok kaynaktan veri bilgilerini alma zamanından tasarruf sağlar. Ayrıca buluttaki verilere kolayca erişebilirsiniz.
- Veri ambarı, gelecekteki tahminlerde bulunmak için farklı dönemleri ve eğilimleri analiz etmek için büyük miktarda geçmiş veriyi depolamanıza olanak tanır.
- Operasyonel iş uygulamalarının ve müşteri ilişkileri yönetimi sistemlerinin değerini artırır
- Analitik işlemeyi işlem veritabanlarından ayırarak her iki sistemin performansını iyileştirir
- Paydaşlar ve kullanıcılar, kaynak sistemlerdeki verilerin kalitesini olduğundan fazla tahmin ediyor olabilir. Veri ambarı daha doğru raporlar sağlar.

## Veritabanının Özellikleri?

- Güvenlik sunar ve fazlalığı ortadan kaldırır
- Verilerin birden çok görünümüne izin ver
- Veritabanı sistemi ACID uyumluluğunu (Atomiklik, Tutarlılık, İzolasyon ve Dayanıklılık) izler.
- Programlar ve veriler arasında yalıtım sağlar
- Veri paylaşımı ve çok kullanıcıli işlem işleme
- İlişkisel Veritabanı çok kullanıcıli ortamı destekler

## Veri Ambarının Özellikleri?

- Bir veri ambarı, şirketlerin devam eden operasyonları yerine temayla ilgili bilgiler sunduğu için konu odaklıdır.
- Ayrıca verilerin Datawarehouse'da ortak ve uyumluluğuyla kabul edilebilir bir şekilde depolanması gerekir.
- Veri ambarı için zaman ufku, diğer operasyonel sistemlere kıyasla nispeten geniştir.
- Bir veri ambarı geçici değildir, bu da önceki verilerin içine yeni bilgiler girildiğinde silinmediği anlamına gelir.

## Veritabanı ve Veri Ambarı Arasındaki Fark

Bu bölümdeki verileri panoply.io sitesinden aldım.

### İşleme Türleri: OLAP vs OLTP

Veritabanları ile veri ambarları arasındaki en önemli fark, verileri nasıl işledikleridir.

Veritabanları, çok sayıda kısa çevrimiçi işlemi hızla silmek, eklemek, değiştirmek ve güncellemek için Çevrimiçi İşlem İşlemini (OLTP) kullanır. Bu tür bir işlem, kullanıcı isteklerine anında yanıt verir ve bu nedenle bir işletmenin günlük işlemlerini gerçek zamanlı olarak işlemek için kullanılır. Örneğin, bir kullanıcı çevrimiçi bir rezervasyon formu kullanarak bir otel odası rezerve etmek isterse, işlem OLTP ile yürütülür.

Veri ambarları, büyük hacimli verileri hızla analiz etmek için OnLine Analitik İşlem (OLAP) kullanır. Bu süreç, analistlere verilerinize farklı bakış açılarından bakma gücü verir. Örneğin, veritabanınız her günün her dakikası için satış verilerini kaydetse bile, sadece her gün satılan toplam miktarı bilmek isteyebilirsiniz. Bunu yapmak için, her gün için satış verilerini toplamamız ve toplamamız gerekir. OLAP, bunu yapmak ve aynı hesaplamayı gerçekleştirmek için OLTP kullanmanıza göre 1000 kat daha hızlı veri depolamak için kullanmak üzere özel olarak tasarlanmıştır .

## Optimizasyon

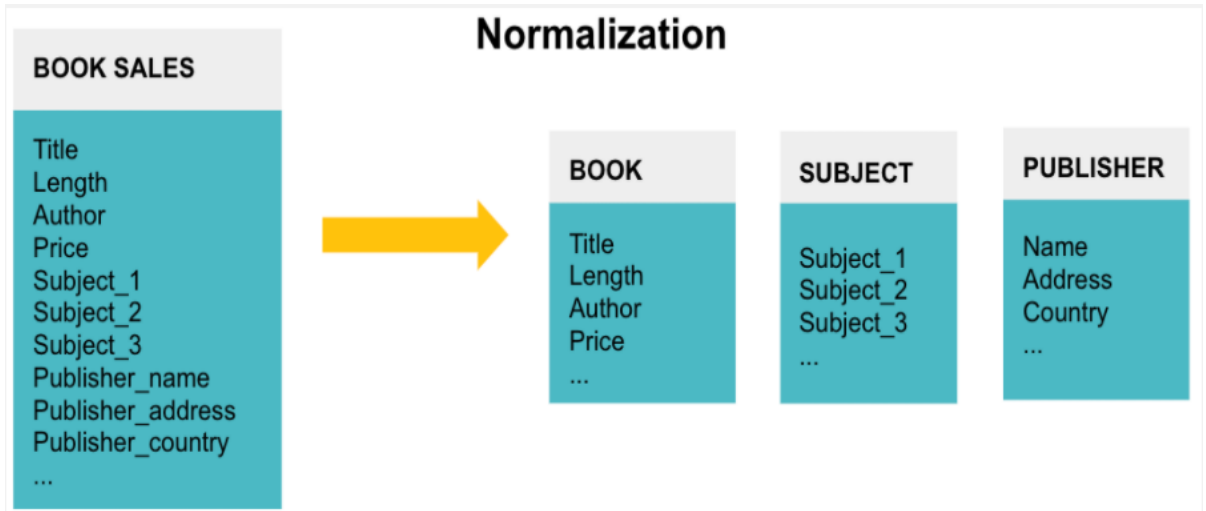
Bir veritabanı, verileri maksimum hız ve verimlilikle güncellemek (eklemek, değiştirmek veya silmek) için optimize edilmiştir. Verimli işlem işlem için veritabanlarından gelen yanıt sürelerinin son derece hızlı olması gerekir. Bir veritabanının en önemli yönü, yazma işlemini sisteme kaydetmesidir; Veritabanı her satın alma işleminin kaydını tutmadıysa, bir şirket çok uzun süre faaliyette olmayacaktır!

Veri ambarları, büyük çok boyutlu veri kümelerinde az sayıda karmaşık sorguyu hızla yürütmek için optimize edilmiştir.

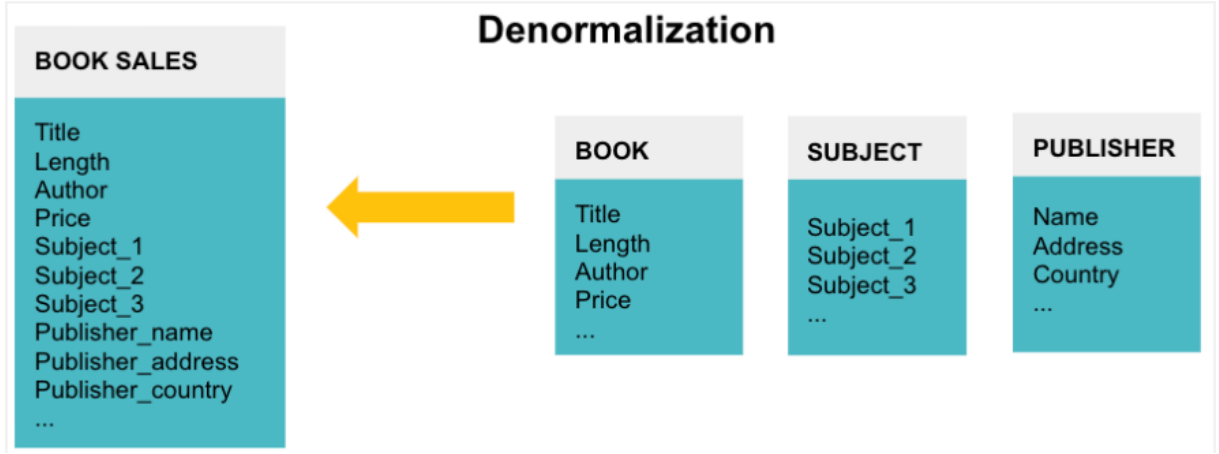
## Veri yapısı

Veritabanlarındaki veriler normalleştirilir. Normalleştirmenin amacı, veri fazlalığını azaltmak ve hatta ortadan kaldırmaktır, yani aynı veri parçasını birden fazla kez depolamaktır. Yinelenen verilerin bu şekilde azalması, tutarlılığın artmasına ve dolayısıyla, veritabanı verileri yalnızca tek bir yerde depoladığından daha doğru verilere yol açar.

Verileri normalleştirmek, verileri birçok farklı tabloya böler. Her tablo, verilerin ayrı bir varlığını temsil eder. Örneğin, KİTAP SATIŞLARI'nı kaydeden bir veritabanı, KİTAP bilgilerini, kitapta kapsanan KONU ve YAYINCI'yı belirtmek için üç tabloya sahip olabilir.



Verileri normalleştirmek, veritabanının minimum disk alanı kaplamasını ve dolayısıyla bellek açısından verimli olmasını sağlar. Ancak, sorgulama açısından verimli değildir. Normalleştirilmiş bir veritabanını sorgulamak yavaş ve külfetli olabilir. İşletmeler, veri ambarlarındaki veriler üzerinde karmaşık sorgular gerçekleştirmek istediklerinden, bu veriler genellikle normalden uzaklaştırılır ve daha kolay erişim için tekrarlanan veriler içerir.



## Veri analizi

Veritabanları genellikle sadece işlemleri işler, ancak onlarla veri analizi yapmak da mümkündür. Ancak, derinlemesine keşif, normalleştirilmiş veri yapısı ve gerçekleştirmeniz gereken çok sayıda tablo birleştirme nedeniyle hem kullanıcı hem de bilgisayar için zordur. Çok fazla zaman ve bilgi işlem kaynağı alan bir DataBase Management System (DBSM) üzerinde karmaşık sorgular oluşturmak ve yürütmek için yetenekli bir geliştirici veya analist gerektirir. Dahası, analiz derinlemesine gitmez elde edebileceğiniz en iyi şey tek seferlik statik bir rapordur, çünkü veritabanları yalnızca belirli bir zamanda verilerin anlık görüntüsünü verir.

Veri ambarları, büyük çok boyutlu veri kümeleri üzerinde karmaşık analitik sorgular basit bir şekilde gerçekleştirmek için tasarlanmıştır. Gelişmiş teori öğrenmeye veya sofistike DBMS yazılımının nasıl kullanılacağını öğrenmeye gerek yoktur. Analizin gerçekleştirilmesi daha basit olmakla kalmaz, sonuçlar çok daha kullanışlıdır; Veritabanlarının sağladığı anlık görüntü yerine, derinlemesine dalebilir ve verilerinizin zaman içinde nasıl değiştiğini görebilirsiniz.

## Veri Zaman Çizelgesi

Veritabanları, işin bir yönü için günlük işlemleri işler. Bu nedenle, genellikle tek bir iş süreciyle ilgili geçmiş veriler yerine güncel veriler içerirler.

Veri ambarları, analitik amaçlar ve iş raporlaması için kullanılır. Veri ambarları genellikle farklı kaynaklardan gelen işlem verilerinin kopyalarını entegre ederek geçmiş verileri depolar. Veri ambarları, en güncel, entegre bilgileri kullanan raporlar için gerçek zamanlı veri akışlarını da kullanabilir.

## Eşzamanlı kullanıcılar

Veritabanları , işletmenin işlemlerini yansıtmak için gerçek zamanlı olarak güncellendiğinden [binlerce eşzamanlı kullanıcıyı](#) destekler . Bu nedenle, birçok kullanıcının performansını etkilemeden veritabanıyla eşzamanlı olarak etkileşime girmesi gerekir.

Bununla birlikte, bir seferde yalnızca bir kullanıcı bir veri parçasını değiştirebilir - iki kullanıcının aynı bilgileri aynı anda farklı şekillerde yazması felaket olur!

Buna karşılık, veri ambarları sınırlı sayıda eşzamanlı kullanıcıyı destekler. Bir veri ambarı, ön uç uygulamalardan ayrılmıştır ve onu kullanmak, karmaşık sorguların yazılmasını ve yürütülmesini içerir. Bu sorgular hesaplama açısından pahalıdır ve bu nedenle sistemi aynı anda yalnızca az sayıda kişi kullanabilir.

## ACID Uyumluluğu

Veritabanı işlemleri genellikle ACID (Atomik, Tutarlı, İzole ve Dayanıklı) uyumlu bir şekilde yürütülür. Bu uyumluluk, verilerin güvenilir ve yüksek bütünlüklü bir şekilde değişmesini sağlar. Bu nedenle, hatalar veya güç kesintileri durumunda bile güvenilir olabilir. Veritabanı ticari işlemlerin bir kaydı olduğundan, her birini en üst düzeyde bütünlükle kaydetmelidir.

Veri ambarları, birçok farklı kaynaktan alınan geçmiş verileri değiştirmek yerine okumaya odaklandığından, ACID uyumluluğu daha az sıkı bir şekilde uygulanır. Bununla birlikte, Redshift ve Panoply gibi en iyi bulut sağlayıcıları, sorgularının mümkün olduğunda ACID uyumlu olmasını sağlar. Örneğin, MySQL ve PostgreSQL kullanırken her zaman durum böyledir.

## Veritabanı ve Veri Ambarı SLA'ları

Veritabanlarının çoğu SLA'sı, herhangi bir sistem arızası gelir kaybına ve davalara neden olabileceğinden %99,99 çalışma süresini karşılamaları gerektiğini belirtir.

Bazı gerçekten büyük veri ambarlarına yönelik SLA'ları, yeni verilerin periyodik olarak yüklenmesini sağlamak için genellikle kapalı kalma süresine sahiptir. Bu, modern veri ambarlama için daha az yaygındır.

## Veritabanı Kullanım Durumları

Veritabanları, bir organizasyondaki günlük işlemleri işler. Bazı veritabanı uygulamaları örnekleri şunları içerir:

- Sattığı bir ürün için sipariş oluşturan bir e-ticaret web sitesi
- Çevrimiçi rezervasyon sistemi kullanan bir havayolu
- Bir hastayı kaydeden bir hastane
- Bir hesaba ATM para çekme işlemi ekleyen banka

## Veri Ambarı Kullanım Durumları

Veri ambarları, işletmeleri daha bilinçli işler yapma konusunda güçlendiren yüksek düzeyde raporlama ve analiz sağlar. Kullanım örnekleri şunları içerir:

- Müşterilere daha özel içerik sağlamak için geçmiş satın aldıklarına göre farklı gruplara ayırmak
- Son on yıllık satış verilerini kullanarak müşteri kaybını tahmin etme
- Önümüzdeki çeyrekte hangi alanlara odaklanılacağına karar vermek için talep ve satış tahminleri oluşturmak

## Veritabanı ve Veri Ambarı Karşılaştırması

Emlak	Veri tabanı	Veri deposu
İşleme metodu	Çevrimiçi İşlem İşleme (OLTP)	OnLine Analitik İşleme (OLAP)
Optimizasyon	Çok sayıda kısa çevrimiçi işlemi hızla siler, ekler, değiştirir ve günceller.	Büyük hacimli verileri hızla analiz edin ve analistler için farklı bakış açıları sağlayın.
Veri yapısı	Gereksiz veri içermeyen birçok farklı tabloya sahip oldukça normalleştirilmiş veri yapısı.  Bu nedenle, veriler daha doğrudur ancak alınması yavaştır.	Yinelenen verileri içeren birkaç tablo içeren normal olmayan veri yapısı.  Bu nedenle, veriler potansiyel olarak daha az doğrudur ancak geri getirilmesi hızlıdır.



Veri analizi	Çok sayıda tablo birleşimine ihtiyaç duyulması ve mevcut verilerin küçük bir zaman çerçevesi nedeniyle analiz yavaş ve zahmetlidir.	Az sayıda tablo birleşimi ve mevcut geniş zaman çerçevesi nedeniyle analiz hızlı ve kolaydır.
Eşzamanlı kullanıcılar	Binlerce eşzamanlı kullanıcı destekleniyor. Bununla birlikte, her veri parçasını aynı anda yalnızca bir kullanıcı değiştirebilir.	Az sayıda eşzamanlı kullanıcı.
ACID uyumluluğu	En yüksek düzeyde bütünlük sağlamak için verileri ACID uyumlu bir şekilde kaydeder.	Bazı şirketler sunsa da her zaman ACID uyumlu değildir.
Uptime	% 99,99 çalışma süresi	Kapalı kalma süresi, yeni verilerin periyodik olarak yüklenmesine uyum sağlamak için yerleşiktir
Depolama	Belirli bir işletme fonksiyonundan tek bir veri kaynağıyla sınırlıdır	Tüm işletme fonksiyonlarından tüm veri kaynakları
Sorgu Türü	Basit işlem sorguları	Derinlemesine analiz için karmaşık sorgular
Veri özeti	Oldukça ayrıntılı ve hassas	Olmasını istediğiniz kadar ayrıntılı ve hassas

## Veritabanının Dezavantajları

- Uygulayıcı bir Veritabanı sisteminin Donanım ve Yazılım maliyeti yüksektir ve bu da kuruluşunuzun bütçesini artırabilir.
- Çoğu DBMS sistemi genellikle karmaşık sistemlerdir, bu nedenle kullanıcıların DBMS'i kullanmaları için eğitim gereklidir.
- DBMS karmaşık hesaplamalar yapamaz
- Zaten mevcut olan sistemlerle uyumlulukla ilgili sorunlar
- Veri sahipleri verileri üzerindeki kontrollerini kaybederek güvenlik, sahiplik ve gizlilik sorunlarını artırabilir.

## Veri Ambarının Dezavantajları

- Yeni veri kaynakları eklemek zaman alır ve yüksek maliyetle ilişkilendirilir.
- Bazen veri ambarıyla ilgili sorunlar yıllarca tespit edilemeyebilir.
- Veri ambarları yüksek bakım gerektiren sistemlerdir. Verilerin çıkarılması, yüklenmesi ve temizlenmesi zaman alıcı olabilir.
- Veri ambarı basit görünebilir, ancak aslında ortalama kullanıcılar için çok karmaşık. Veri madenciliği ve deposu kullanmayan son kullanıcılara eğitim vermeniz gerekiyor.
- Proje yönetimindeki en iyi çabalara rağmen, veri ambarının kapsamı her zaman artacaktır.

## ETL (Ayıklama, Dönüştürme ve Yükleme) İşlemi

### ETL Nedir?

**ETL**, verileri farklı kaynak sistemlerden çıkarır, ardından verileri dönüştürür (hesaplamalar, birleştirmeler vb.) gibi işlemleri yapar ve son olarak verileri Veri Ambarı sistemine yükler. ETL'in tam biçimi Çıkar, Dönüştür ve Yüklemdir.

Bir Veri ambarı oluşturmanın, basitçe birden çok kaynaktan veri ayıklamak ve bir Veri ambarının veritabanına yüklemek olduğunu düşünmek cazip geliyor. Bu gerçeklerden uzaktır ve karmaşık bir ETL süreci gerektirir. ETL süreci, geliştiriciler, analistler, test uzmanları, üst düzey yöneticiler gibi çeşitli paydaşlardan aktif girdiler gerektirir ve teknik olarak zordur.

Karar vericiler için bir araç olarak değerini korumak için, Veri ambarı sisteminin iş değişiklikleri ile değişmesi gerekiyor. ETL, bir Veri ambarı sisteminin yinelenen bir etkinliğidir (günlük, haftalık, aylık) ve çevik, otomatikleştirilmiş ve iyi belgelenmiş olması gerekir.

### Neden ETL'e ihtiyacınız var?

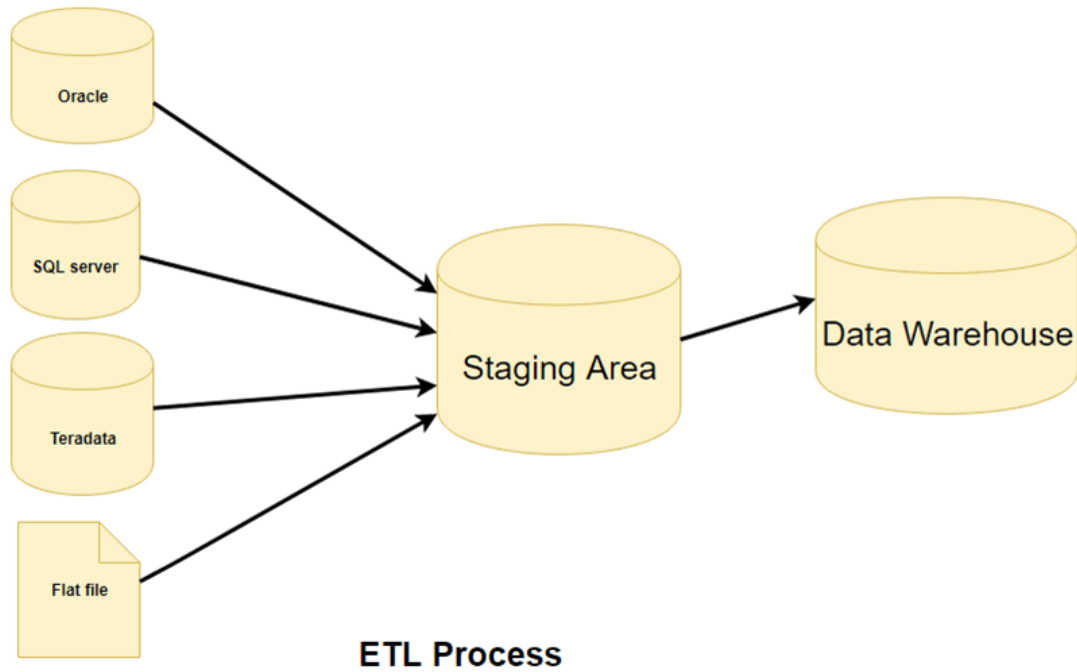
Organizasyonda ETL'i benimsemenin birçok nedeni vardır:

- Şirketlerin, kritik iş kararları almak için iş verilerini analiz etmelerine yardımcı olur.
- İşlem veritabanları, ETL tarafından yanıtlanabilecek karmaşık iş sorularını yanıtlayamaz.
- Veri Ambarı, ortak bir veri deposu sağlar
- ETL, verileri çeşitli kaynaklardan bir veri ambarına taşımak için bir yöntem sağlar.
- Veri kaynakları değiştikçe, Veri Ambarı otomatik olarak güncellenecektir.
- İyi tasarlanmış ve belgelenmiş ETL sistemi, bir Veri Ambarı projesinin başarısı için kesinlikle gereklidir.

- Veri dönüştürme, toplama ve hesaplama kurallarının doğrulanmasına izin verin.
- ETL süreci, kaynak ve hedef sistem arasında örnek veri karşılaştırmasına izin verir.
- ETL süreci, karmaşık dönüşümler gerçekleştirebilir ve verileri depolamak için fazladan alan gerektirir.
- ETL, verilerin Veri Ambarına taşınmasına yardımcı olur. Tek bir tutarlı sisteme bağlı kalmak için çeşitli biçimlere ve türlere dönüştürün.
- ETL, kaynak verilere erişmek ve hedef veritabanında bunları değiştirmek için önceden tanımlanmış bir süreçtir.
- ETL, işletme için derin bir tarihsel bağlam sunar.
- Teknik becerilere ihtiyaç duymadan kodladığı ve yeniden kullandığı için üretkenliği artırmaya yardımcı olur.

## Veri Ambarlarında ETL Süreci

ETL 3 adımlı bir süreçtir.



### Adım 1) Çıkarma

Bu adımda, veriler kaynak sistemden evreleme alanına çıkarılır. Kaynak sistemin performansında herhangi bir bozulma olmaması için aşamalandırma alanında varsa dönüşümler yapılır. Ayrıca, bozuk veriler doğrudan kaynaktan Veri ambarı veritabanına kopyalanırsa, geri alma zor olacaktır. Hazırlama alanı, çıkarılan verileri Veri ambarına taşınmadan önce doğrulama fırsatı verir.

Veri ambarı, farklı özelliklere sahip sistemleri entegre etmelidir.

DBMS, Donanım, İşletim Sistemleri ve İletişim Protokolleri. Kaynaklar, Mainframes gibi eski uygulamaları, özelleştirilmiş uygulamaları, ATM gibi temas noktası aygıtlarını, Çağrı anahtarlarını, metin dosyalarını, elektronik tabloları, ERP'yi, satıcılardan gelen verileri ve diğerlerinin yanı sıra ortakları içerebilir.

Bu nedenle, veriler çıkarılmadan ve fiziksel olarak yüklenmeden önce mantıksal bir veri haritasına ihtiyaç vardır. Bu veri haritası, kaynaklar ve hedef veriler arasındaki ilişkiyi açıklar.

### Üç Veri Çıkarma yöntemi:

1. Tam Çıkarma
2. Kısmi Çıkarma - güncelleme bildirimi olmadan.
3. Kısmi Çıkarma - güncelleme bildirimi ile

Kullanılan yöntemden bağımsız olarak, çıkarma, kaynak sistemlerin performansını ve yanıt süresini etkilememelidir. Bu kaynak sistemler, canlı üretim veritabanlarıdır. Herhangi bir yavaşlama veya kilitlenme şirketin kârını etkileyebilir.

### Çıkarma sırasında bazı doğrulamalar yapılır:

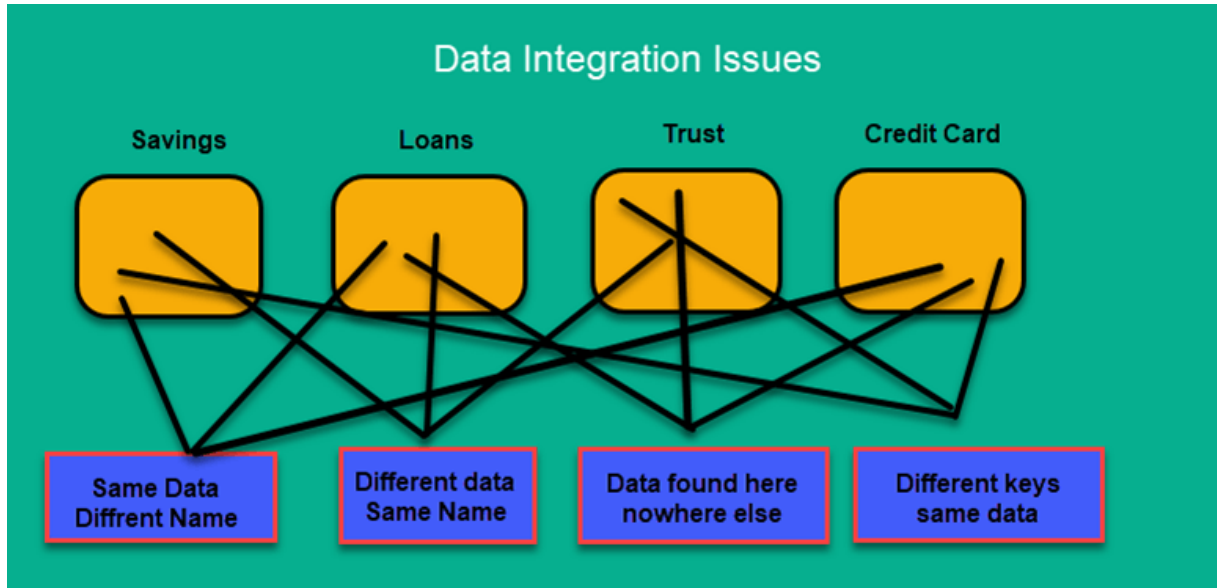
- Kayıtları kaynak verilerle uzlaştıran
- Spam / istenmeyen verilerin yüklenmediğinden emin olun
- Veri türü kontrolü
- Tüm yinelenen / parçalanmış verileri kaldırın
- Tüm anahtarların yerinde olup olmadığını kontrol edin

## Adım 2) Dönüşüm

Kaynak sunucudan çıkarılan veriler hamdır ve orijinal haliyle kullanılamaz. Bu nedenle temizlenmesi, haritası çıkarılması ve dönüştürülmesi gerekiyor. Aslında bu, ETL sürecinin değer kattığı ve verileri, anlayışlı BI raporlarının oluşturulabilmesi için değiştirdiği temel adımdır.

Bu adımda, çıkarılan verilere bir dizi işlev uygularsınız. Herhangi bir dönüşüm gerektirmeyen **verilere doğrudan taşıma** veya **veri üzerinden geçiş** denir.

Dönüşüm adımında, veriler üzerinde özelleştirilmiş işlemler gerçekleştirebilirsiniz. Örneğin, kullanıcı veritabanında olmayan satış toplamı geliri istiyorsa. Veya bir tablodaki ad ve soyad farklı sütunlarda ise. Yüklemeden önce bunları birleştirmek mümkündür.



#### Veri Bütünlüğü Sorunları aşağıdadır:

1. Aynı kişinin Jon, John vb. Gibi farklı yazılışları.
2. Google, Google Inc. gibi şirket adını belirtmenin birçok yolu vardır.
3. Cleaveland, Cleveland gibi farklı isimlerin kullanılması.
4. Aynı müşteri için çeşitli uygulamalar tarafından farklı hesap numaralarının üretildiği bir durum olabilir.
5. Bazı verilerde gerekli dosyalar boş kalır
6. POS'ta manuel giriş olarak alınan geçersiz ürün hatalara neden olabilir.

#### Doğrulamalar bu aşamada yapılır

- Filtreleme - Yükleme için yalnızca belirli sütunları seçin
- Veri standardizasyonu için kuralları ve arama tablolarını kullanma
- Karakter Kümesi Dönüştürme ve kodlama yönetimi
- Tarih Saat Dönüşümü, para birimi dönüştürmeleri, sayısal dönüştürmeler vb. Gibi Ölçü Birimlerinin Dönüştürülmesi
- Veri eşiği doğrulama kontrolü. Örneğin, yaş iki haneden fazla olamaz.
- Aşama alanından ara tablolara veri akışı doğrulaması.
- Gerekli alanlar boş bırakılmamalıdır.
- Temizleme (örneğin, NULL ile 0 veya Cinsiyet Erkeği "E" ve Kadını "K" ile eşleme vb.)
- Bir sütunu katlara bölün ve birden çok sütunu tek bir sütunda birleştirin.
- Satırların ve sütunların yerini değiştirme,
- Verileri birleştirmek için aramaları kullanın
- Herhangi bir karmaşık veri doğrulamasını kullanma (örneğin, bir satırdaki ilk iki sütun boşsa, satırın işlenmesini otomatik olarak reddeder)

### Adım 3) Yükleme

Verilerin hedef veri ambar veritabanına yüklenmesi, ETL sürecinin son adımıdır. Tipik bir Veri ambarında, nispeten kısa bir sürede (geceler) büyük miktarda veri yüklenmesi gerekir. Bu nedenle, yükleme süreci performans için optimize edilmelidir.

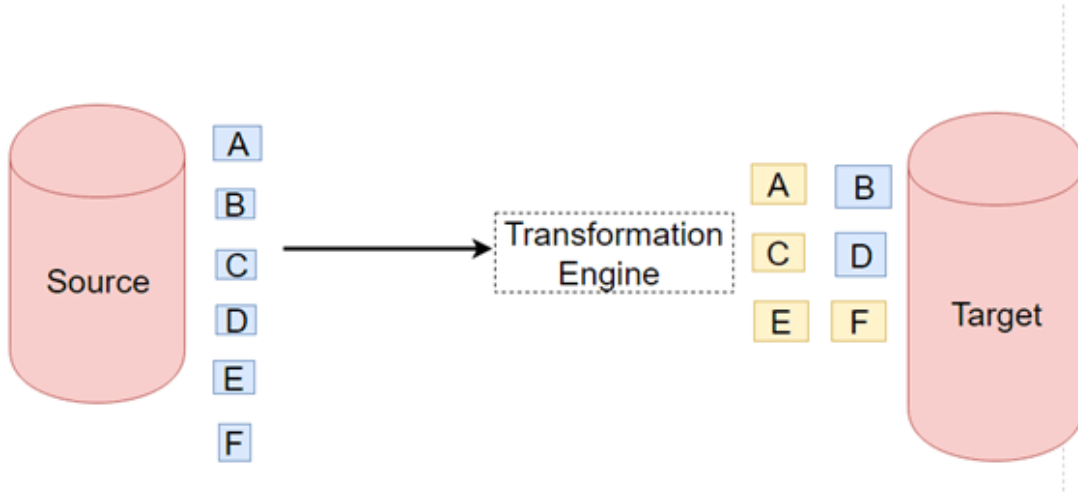
Yük arızası durumunda, kurtarma mekanizmaları veri bütünlüğü kaybı olmadan hata noktasından yeniden başlayacak şekilde yapılandırılmalıdır. Veri Ambarı yöneticilerinin geçerli sunucu performansına göre yükleri izlemesi, devam ettirmesi ve iptal etmesi gerekir.

#### Yükleme Türleri:

- **İlk Yükleme** - tüm Veri Ambarı tablolarını doldurma
- **Artımlı Yük** - periyodik olarak ihtiyaç duyulduğunda devam eden değişiklikleri uygulama.
- **Tam Yenileme** - bir veya daha fazla tablonun içeriğini silip yeni verilerle yeniden yükleme.

#### Yükleme doğrulaması

- Anahtar alan verilerinin eksik veya boş olmadığından emin olun.
- Hedef tablolara dayalı modelleme görünümünü test edin.
- Birleştirilmiş değerleri ve hesaplanan ölçüleri kontrol edin.
- Boyut tablosundaki ve geçmiş tablosundaki veri kontrolleri.
- Yüklenen olgu ve boyut tablosundaki BI raporlarını kontrol edin.



### ETL araçları

Piyasada birçok Veri Ambarı aracı bulunmaktadır. İşte en öne çıkanlardan bazıları:

### 1. MarkLogic:

MarkLogic, bir dizi kurumsal özelliği kullanarak veri entegrasyonunu daha kolay ve daha hızlı hale getiren bir veri ambarı çözümüdür. Belgeler, ilişkiler ve meta veriler gibi farklı veri türlerini sorgulayabilir.

<https://developer.marklogic.com/products/>

### 2. Oracle:

Oracle, sektör lideri veritabanıdır. Hem şirket içi hem de bulut için çok çeşitli Veri Ambarı çözümleri sunar. Operasyonel verimliliği artırarak müşteri deneyimlerini optimize etmeye yardımcı olur.

<https://www.oracle.com/index.html>

### 3. Amazon RedShift:

Amazon Redshift, Datawarehouse aracıdır. Standart SQL ve mevcut BI araçlarını kullanarak tüm veri türlerini analiz etmek için basit ve uygun maliyetli bir araçtır. Ayrıca petabaytlarca yapılandırılmış veriye karşı karmaşık sorguların çalıştırılmasına izin verir.

[https://aws.amazon.com/redshift/?nc2=h\\_m1](https://aws.amazon.com/redshift/?nc2=h_m1)

İşte yararlı [Veri ambarı Araçlarının](#) tam listesi .

## En iyi uygulamalar ETL süreci

### Asla tüm verileri temizlemeye çalışmayın:

Her kuruluş tüm verilerin temiz olmasını ister, ancak bunların çoğu beklemek için ödeme yapmaya veya beklemeye hazır değildir. Hepsini temizlemek çok uzun sürer, bu nedenle tüm verileri temizlemeye çalışmamak daha iyidir.

### Asla Hiçbir Şeyi Temizlemeyin:

Her zaman bir şeyi temizlemeyi planlayın çünkü Veri Ambarı oluşturma en büyük nedeni daha temiz ve daha güvenilir veriler sunmaktır.

### Verileri temizlemenin maliyetini belirleyin:

Tüm kirli verileri temizlemeden önce, her kirli veri ögesi için temizleme maliyetini belirlemeniz önemlidir.

**Sorgu işlemeyi hızlandırmak için yardımcı görünümlere ve dizinlere sahip olun:**

Depolama maliyetlerini azaltmak için özetlenmiş verileri disk bantlarında saklayın. Ayrıca, depolanacak veri hacmi ile ayrıntılı kullanımı arasındaki denge gereklidir. Depolama maliyetlerini düşürmek için veri ayrıntı düzeyinde değiş tokuş yapın.

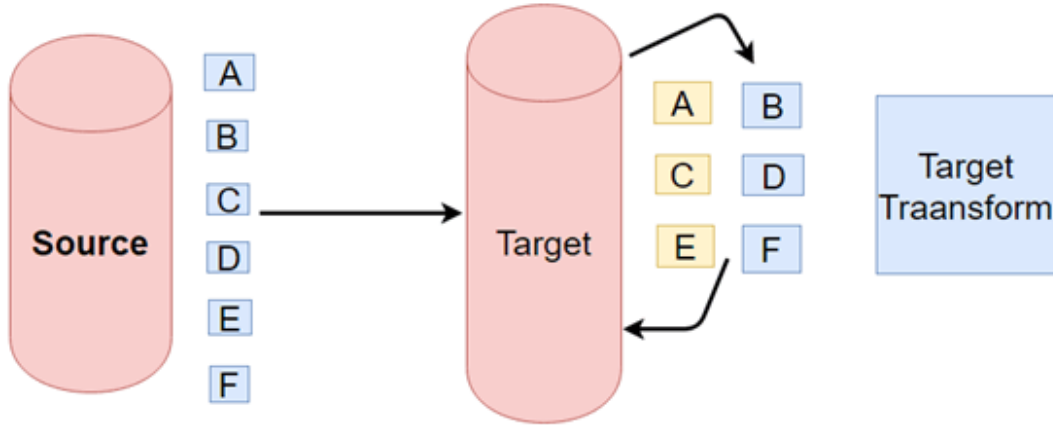
## Özet:

- ETL Ayıklama, Dönüştürme ve Yükleme anlamına gelir.
- ETL, verileri çeşitli kaynaklardan bir veri ambarına taşımak için bir yöntem sağlar.
- İlk adımda çıkarımda, veriler kaynak sistemden evreleme alanına çıkarılır.
- Dönüştürme adımı, kaynaktan çıkarılan veriler temizlenir ve dönüştürülür.
- Verilerin hedef veri ambarına yüklenmesi, ETL sürecinin son adımıdır.

## ELT nedir?

ELT, veri hareketine araç yaklaşımına bakmanın farklı bir yöntemidir. ELT, verileri yazılmadan önce dönüştürmek yerine hedef sistemin dönüşümü yapmasına izin verir. Veriler önce hedefe kopyalanır ve sonra yerine dönüştürülür.

ELT genellikle Hadoop kümesi, veri cihazı veya bulut kurulumu gibi Sql içermeyen veritabanları ile kullanılır.



## ANAHTAR FARK

- ETL, Çıkartma, Dönüştürme ve Yükleme anlamına gelirken ELT, Çıkarma, Yükleme, Dönüştürme anlamına gelir.
- ETL verileri önce hazırlama sunucusuna ve ardından hedef sisteme yüklerken ELT verileri doğrudan hedef sisteme yükler.



- ETL modeli şirket içi, ilişkisel ve yapılandırılmış veriler için kullanılırken ELT, ölçeklenebilir bulut yapılı ve yapılandırılmamış veri kaynakları için kullanılır.
- ETL temel olarak az miktarda veri için kullanılırken ELT büyük miktarlarda veri için kullanılır.
- ELT veri gölü desteği sağlarken, ETL veri gölü desteği sağlamaz.
- ETL'nin uygulanması kolaydır, ELT'nin uygulanması ve sürdürülmesi için niş beceriler gerekir.

## ETL vs ELT

Parametreler	ETL	ELT
<b>İşlem</b>	Veriler, hazırlık sunucusunda dönüştürülür ve ardından Datawarehouse DB'ye aktarılır.	Veriler, Datawarehouse'un DB'sinde kalır.
<b>Kod Kullanımı</b>	İçin kullanılır <ul style="list-style-type: none"> <li>• Bilgi İşlem Yoğun Dönüşümler</li> <li>• Küçük miktarda veri</li> </ul>	Yüksek miktarda veri için kullanılır
<b>dönüşüm</b>	Dönüşümler ETL server / staging alanında yapılır.	Hedef sistemde dönüşümler gerçekleştirilir
<b>Zaman Yüğü</b>	Veriler önce evrelemeye ve daha sonra hedef sisteme yüklenir. Yoğun zaman.	Veriler hedef sisteme yalnızca bir kez yüklendi. Daha hızlı.
<b>Zaman Dönüşümü</b>	ETL işleminin dönüşümün tamamlanmasını beklemesi gerekir. Veri boyutu büyüdükçe dönüştürme süresi artar.	ELT sürecinde hız hiçbir zaman verinin boyutuna bağlı değildir.
<b>Zaman - Bakım</b>	Yüklenecek ve dönüştürülecek verileri seçmeniz gerektiğinden yüksek düzeyde bakım gerektirir.	Veriler her zaman mevcut olduğundan az bakım gerektirir.
<b>Uygulama Karmaşıklığı</b>	Erken bir aşamada, uygulaması daha kolay.	ELT sürecini uygulamak için organizasyon, araçlar hakkında derin bilgiye ve uzman becerilere sahip olmalıdır.
<b>Veri ambarı desteği</b>	Şirket içi, ilişkisel ve yapılandırılmış veriler için kullanılan ETL modeli.	Yapılandırılmış, yapılandırılmamış veri kaynaklarını destekleyen ölçeklenebilir bulut altyapısında kullanılır.
<b>Data Lake Desteği</b>	Desteklemiyor.	Yapılandırılmamış verilerle Data lake kullanımına izin verir.
<b>Karmaşıklık</b>	ETL işlemi, tasarım zamanında tanımlanan yalnızca önemli verileri yükler.	Bu süreç, çıktıdan geriye doğru geliştirme ve yalnızca ilgili verilerin yüklenmesini içerir.
<b>Maliyet</b>	Küçük ve orta ölçekli işletmeler için yüksek maliyetler.	Çevrimiçi Yazılımları Hizmet Platformları olarak kullanarak düşük giriş maliyetleri.
<b>Aramalar</b>	ETL sürecinde, hem gerçekler hem de boyutların hazırlık alanında mevcut olması gerekir.	Çıkarma ve yükleme tek bir eylemde gerçekleştiği için tüm veriler kullanılabilir olacaktır.
<b>Toplamalar</b>	Veri kümesindeki ek veri miktarı ile karmaşıklık artar.	Hedef platformun gücü, önemli miktarda veriyi hızlı bir şekilde işleyebilir.
<b>Hesaplamalar</b>	Mevcut sütunun üzerine yazar veya Veri kümesini eklemeniz gerekir ve hedef platforma aktarır.	Hesaplanan sütunu mevcut tabloya kolayca ekleyin.
<b>Olgunluk</b>	Süreç yirmi yıldan fazla bir süredir kullanılmaktadır. İyi belgelenmiştir ve en iyi uygulamalara kolayca erişilebilir.	Nispeten yeni konsept ve uygulanması karmaşık.
<b>Donanım</b>	Çoğu aracın pahalı olan benzersiz donanım gereksinimleri vardır.	SaaS donanım maliyeti olmak bir sorun değildir.
<b>Yapılandırılmamış Veriler için Destek</b>	Çoğunlukla ilişkisel verileri destekler	Yapılandırılmamış veriler için destek hazır.

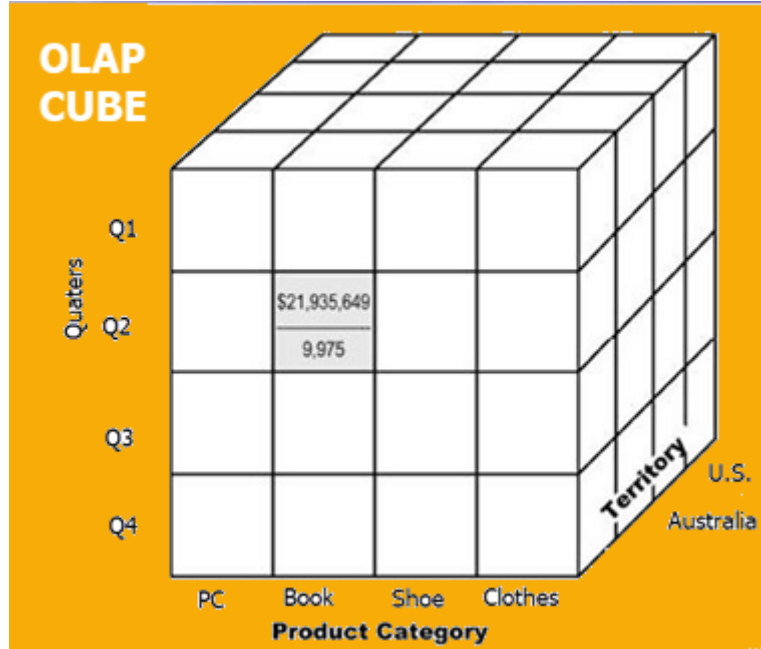
## OLAP nedir?

**Çevrimiçi Analitik İşleme (OLAP)**, kullanıcıların aynı anda birden fazla veritabanı sisteminden gelen bilgileri analiz etmesine olanak tanıyan bir yazılım kategorisidir. Analistlerin iş verilerini farklı bakış açılarından çıkarmasını ve görüntülemesini sağlayan bir teknolojidir.

Analistlerin sık sık verileri gruplaması, bir araya getirmesi ve birleştirmesi gerekir. İlişkisel veritabanlarındaki bu işlemler yoğun kaynak gerektirir. OLAP ile veriler önceden hesaplanabilir ve önceden toplanabilir, bu da analizi daha hızlı hale getirir.

OLAP veritabanları bir veya daha fazla küplere bölünmüştür. Küpler, raporların oluşturulması ve görüntülenmesi kolaylaşacak şekilde tasarlanmıştır.

## OLAP Küpü



OLAP konseptinin temelinde bir OLAP Küpü var. OLAP küpü, çok hızlı veri analizi için optimize edilmiş bir veri yapısıdır.

OLAP Küpü, boyutlara göre kategorize edilen ölçü adı verilen sayısal gerçeklerden oluşur. OLAP Küpü, **hiperküp** olarak da adlandırılır.

Genellikle, veri işlemleri ve analizi, veri değerlerinin satır ve sütun biçiminde düzenlendiği basit elektronik tablo kullanılarak gerçekleştirilir. Bu, iki boyutlu veriler için idealdir. Bununla birlikte OLAP, genellikle farklı ve ilgisiz bir kaynaktan elde edilen verilerle çok boyutlu veriler içerir. Elektronik tablo kullanmak en uygun seçenek değildir. Küp, çok boyutlu verileri mantıksal ve düzenli bir şekilde depolayabilir ve analiz edebilir.

## OLAP Nasıl Çalışır?

Bir Veri ambarı, metin dosyaları, excel sayfası, multimedya dosyaları vb.

Çıkarılan veriler temizlenir ve dönüştürülür. Veriler, bilgilerin daha fazla analiz için önceden hesaplandığı bir OLAP sunucusuna (veya OLAP küpüne) yüklenir.

## OLAP'in temel analitik işlemleri

OLAP'ta dört tür analitik işlem vardır:

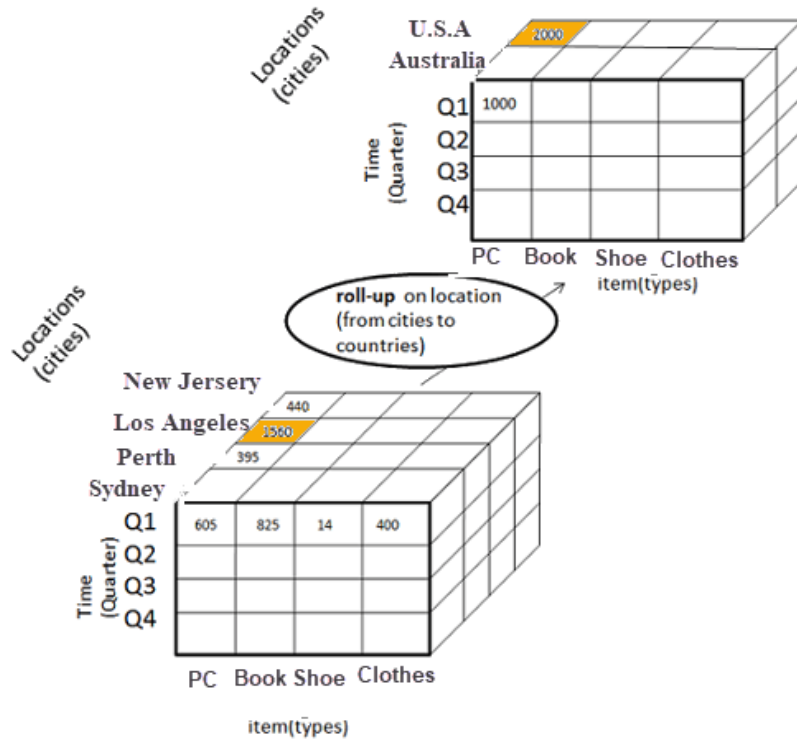
1. Toplama
2. Ayrıntılı inceleme
3. Dilim ve zar
4. Pivot (döndür)

### 1) Toplama:

Toplama, "konsolidasyon" veya "toplama" olarak da bilinir. Toplama işlemi 2 şekilde gerçekleştirilebilir

1. Boyutları küçültmek
2. Kavram hiyerarşisinde tırmanma. Kavram hiyerarşisi, nesneleri sıralarına veya düzeylerine göre gruplama sistemidir.

Aşağıdaki şemayı düşünün

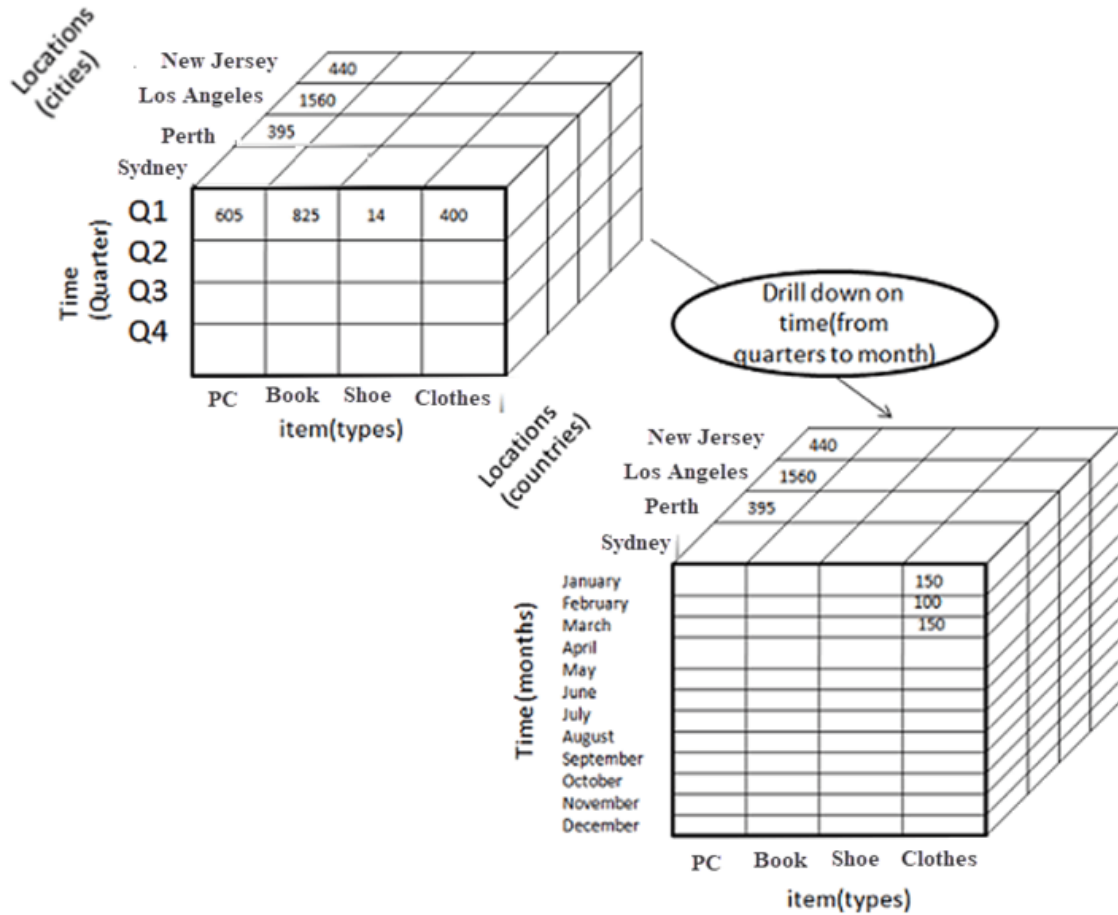


- Bu örnekte, New Jersey ve Los Angeles şehirleri, ABD ülkesine dahil edildi.
- New Jersey ve Los Angeles'ın satış rakamları sırasıyla 440 ve 1560'tır. Toplandıktan sonra 2000 olurlar
- Bu toplama sürecinde veriler, konum hiyerarşisinin şehirden ülkeye doğru hareket etmesidir.
- Toplama işleminde en az bir veya daha fazla boyutun kaldırılması gerekir. Bu örnekte Quater boyutu kaldırılmıştır.

## 2) Ayrıntılı inceleme

Detaya inmede veriler daha küçük parçalara bölünür. Toplama işleminin tam tersidir. Aracılığıyla yapılabilir

- Kavram hiyerarşisinde aşağıya inme
- Bir boyutu büyütme



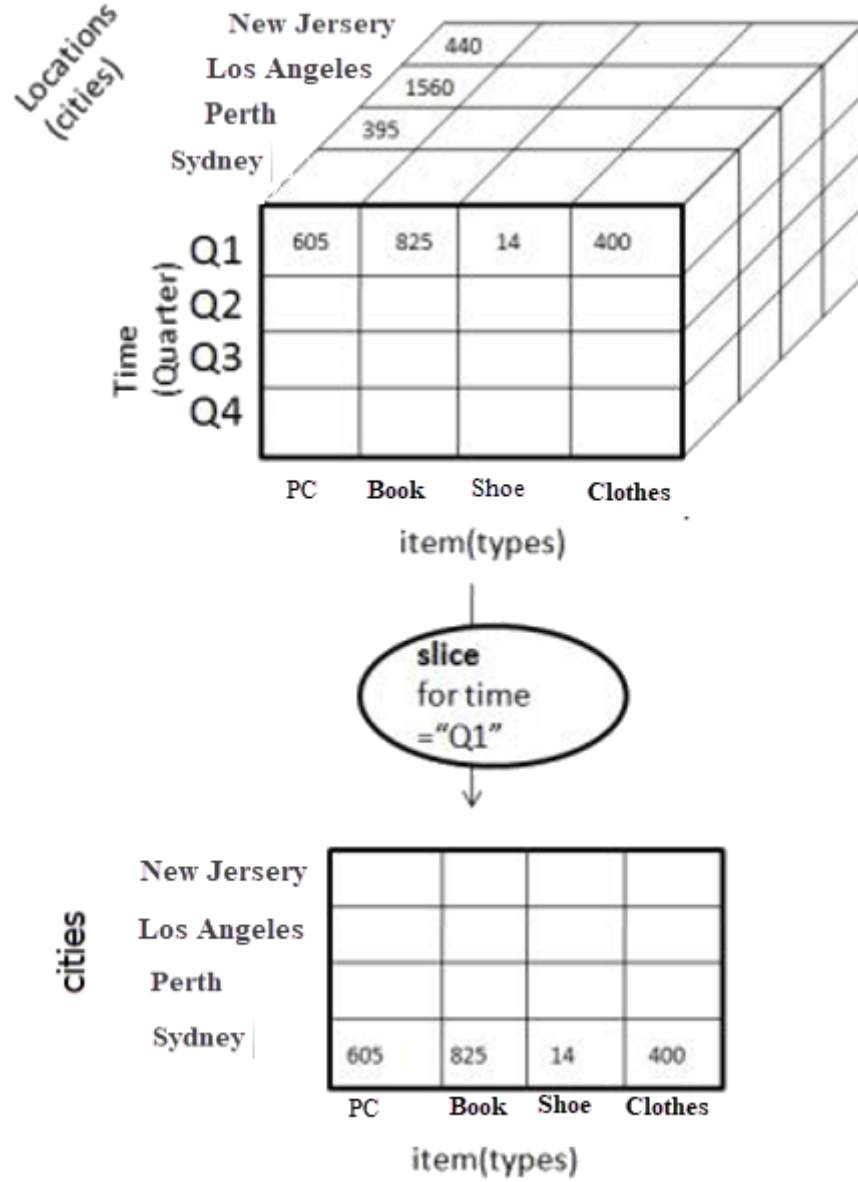
Yukarıdaki diyagramı düşünün

- Çeyrek Q1, Ocak, Şubat ve mart aylarına indirildi. İlgili satışlar da kayıtlardır.
- Bu örnekte, boyut ayları eklenmiştir.

### 3) Dilim:

Burada bir boyut seçilir ve yeni bir alt küp oluşturulur.

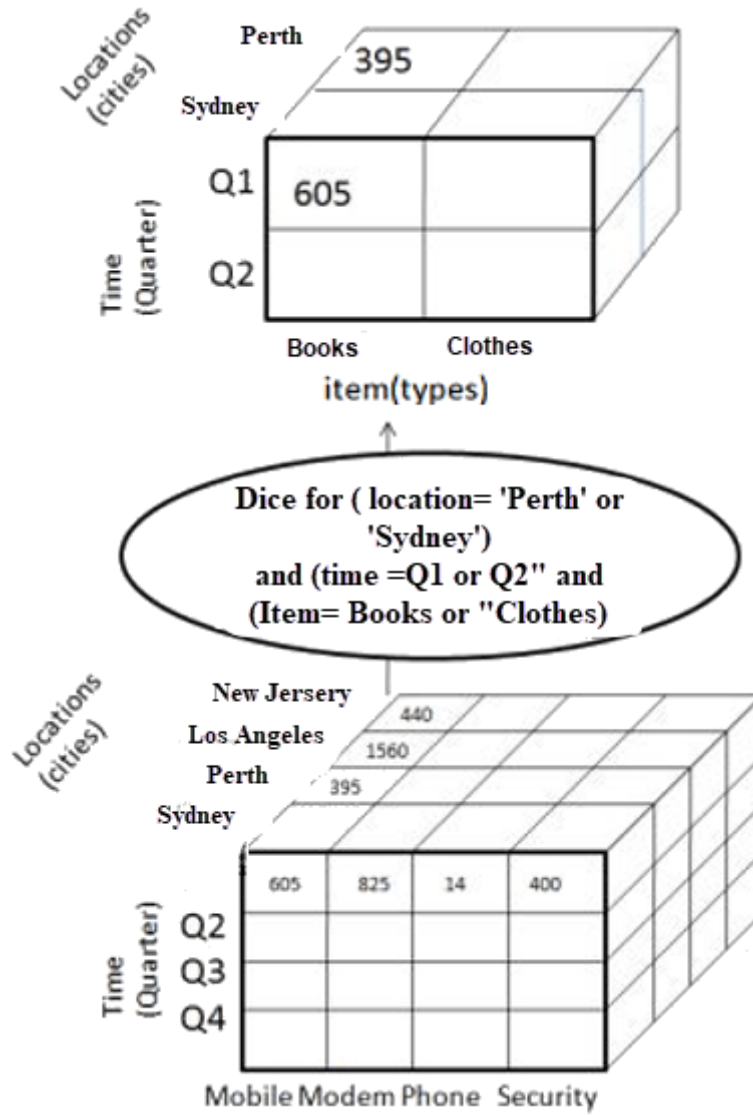
Aşağıdaki diyagram, dilim işleminin nasıl gerçekleştirildiğini açıklamaktadır:



- Boyut Zamanı, filtre olarak Q1 ile Dilimlenir.
- Tamamen yeni bir küp oluşturulur.

## Zar:

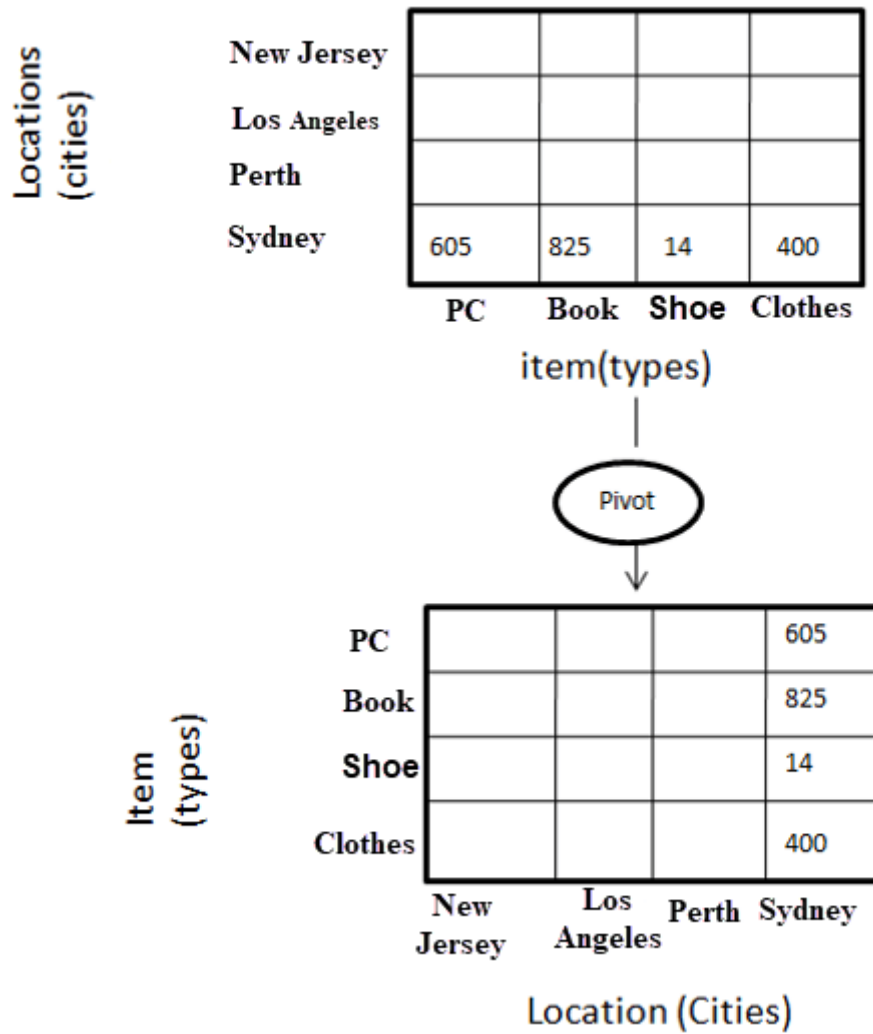
Bu işlem bir dilime benzer. Zardaki fark, bir alt küpün oluşturulmasıyla sonuçlanan 2 veya daha fazla boyut seçmenizdir.



## 4) Pivot

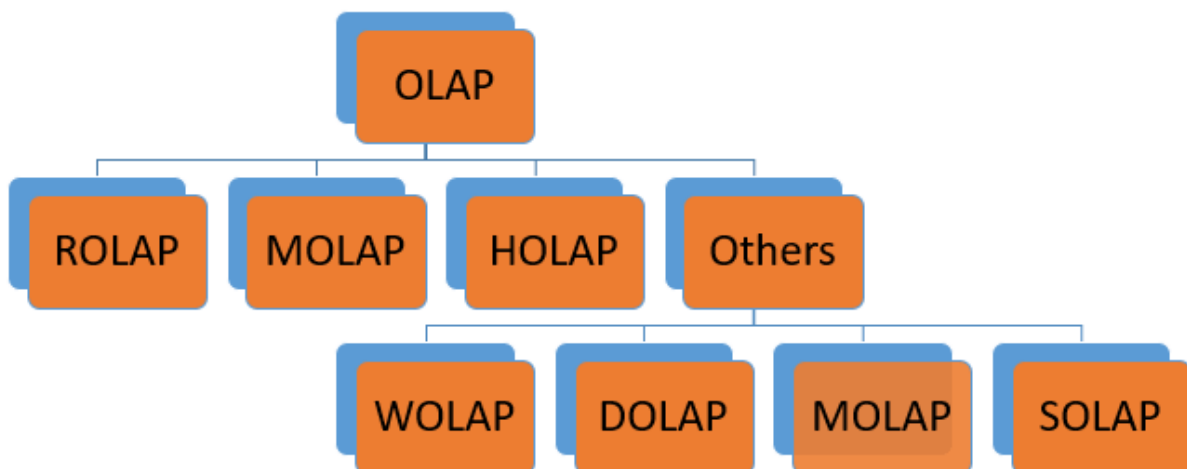
Pivotta, verilerin yedek bir sunumunu sağlamak için veri eksenlerini döndürürsünüz.

Aşağıdaki örnekte, pivot öge türlerine dayanmaktadır.



## OLAP sistemlerinin türleri

OLAP Hiyerarşik Yapısı



OLAP türü	Açıklama
İlişkisel OLAP (ROLAP):	ROLAP, standart ilişkisel işlemi gerçekleştirmek için çok boyutlu veri eşlemenin yanı sıra genişletilmiş bir RDBMS'dir.
Çok Boyutlu OLAP (MOLAP)	MOLAP, çok boyutlu verilerde işlem gerçekleştirir.
Hibrit Çevrimiçi Analitik İşleme (HOLAP)	HOLAP yaklaşımında, toplu toplamalar çok boyutlu bir veritabanında saklanırken, ayrıntılı veriler ilişkisel veritabanında saklanır. Bu, hem ROLAP modelinin veri verimliliğini hem de MOLAP modelinin performansını sunar.
Masaüstü OLAP (DOLAP)	Masaüstü OLAP'ta, bir kullanıcı verilerin bir bölümünü veritabanından yerel olarak veya masaüstüne indirir ve analiz eder.  DOLAP, diğer OLAP sistemlerine kıyasla çok az işlevsellik sunduğundan kullanımı nispeten daha ucuzdur.
Web OLAP (WOLAP)	Web tarayıcısı aracılığıyla erişilebilen OLAP sistemi olan Web OLAP. WOLAP, üç katmanlı bir mimaridir. Üç bileşenden oluşur: istemci, ara katman yazılımı ve bir veritabanı sunucusu.
Mobil OLAP:	Mobil OLAP, kullanıcıların mobil cihazlarını kullanarak OLAP verilerine erişmelerine ve bunları analiz etmelerine yardımcı olur
Uzamsal OLAP:	SOLAP, bir Coğrafi Bilgi sisteminde (GIS) hem mekansal hem de mekansal olmayan verilerin yönetimini kolaylaştırmak için oluşturulmuştur.

## ROLAP

ROLAP, ilişkisel bir veritabanında bulunan verilerle çalışır. Gerçekler ve boyut tabloları ilişkisel tablolar olarak saklanır. Ayrıca, verilerin çok boyutlu analizine izin verir ve en hızlı büyüyen OLAP'dir.

### ROLAP modelinin avantajları:

- **Yüksek veri verimliliği.** Sorgu performansı ve erişim dili özellikle çok boyutlu veri analizi için optimize edildiği için yüksek veri verimliliği sunar.
- **Ölçeklenebilirlik.** Bu tür OLAP sistemi, büyük hacimli verileri yönetmek için ve hatta veriler sürekli olarak artarken bile ölçeklenebilirlik sunar.

### ROLAP modelinin dezavantajları:

- **Daha yüksek kaynaklara talep:** ROLAP, yüksek insan gücü, yazılım ve donanım kaynaklarının kullanımına ihtiyaç duyar.
- **Toplu olarak veri sınırlamaları.** ROLAP araçları, toplu verilerin tüm hesaplamaları için SQL kullanır. Ancak, hesaplamaları işlemek için belirlenmiş bir sınır yoktur.
- **Yavaş sorgu performansı.** Bu modelde sorgu performansı, MOLAP ile karşılaştırıldığında yavaştır



## MOLAP

MOLAP, verilerin çok boyutlu görünümünü görüntülemek için dizi tabanlı çok boyutlu depolama motorlarını kullanır. Temel olarak, bir OLAP küpü kullanırlar.

OLAP hakkında [buradan](#) daha fazla bilgi edinin

## Hibrit OLAP

Hibrit OLAP hem ROLAP hem de MOLAP'ın bir karışımıdır. Hızlı MOLAP hesaplaması ve daha yüksek ROLAP ölçeklenebilirliği sunar. HOLAP, iki veritabanı kullanır.

1. Birleştirilmiş veya hesaplanmış veriler çok boyutlu bir OLAP küpünde saklanır
2. Ayrıntılı bilgi ilişkisel bir veritabanında saklanır.

### Hibrit OLAP'ın Faydaları:

- Bu tür bir OLAP, disk alanını tasarruf etmeye yardımcı olur ve aynı zamanda, erişim hızı ve rahatlığı ile ilgili sorunları önlemeye yardımcı olan kompakt kalır.
- Hibrit HOLAP'lar, her tür veri için daha hızlı performans sağlayan küp teknolojisini kullanır.
- ROLAP anında güncellenir ve HOLAP kullanıcıları bu gerçek zamanlı anında güncellenen verilere erişebilir. MOLAP, verilerin temizlenmesini ve dönüştürülmesini sağlayarak verilerin alaka düzeyini artırır. Bu, her iki dünyanın en iyisini getirir.

### Hibrit OLAP'ın Dezavantajları:

- Daha fazla karmaşıklık seviyesi: HOLAP sistemlerinde en büyük dezavantaj hem ROLAP hem de MOLAP araçlarını ve uygulamalarını desteklemesidir. Bu nedenle çok karmaşıktır.
- Olası örtüşmeler: Özellikle işlevleriyle örtüşme şansı daha yüksektir.

## OLAP'ın Avantajları

- OLAP, planlama, bütçeleme, raporlama ve analiz içeren her tür iş için bir platformdur.
- Bir OLAP küpünde bilgi ve hesaplamalar tutarlıdır. Bu çok önemli bir avantajdır.
- "Olursa ne olur" senaryolarını hızla oluşturun ve analiz edin
- OLAP veritabanında kolayca geniş veya belirli terimler için arama yapın.
- OLAP, iş modelleme araçları, Veri madenciliği araçları, performans raporlama araçları için yapı taşları sağlar.
- Kullanıcıların küp verilerini çeşitli boyutlara, ölçülere ve filtrelere göre dilimlemesine ve kesmesine olanak tanır.
- Zaman serilerini analiz etmek için iyidir.
- OLAP ile bazı kümeleri ve aykırı değerleri bulmak kolaydır.

- Daha hızlı yanıt süreleri sağlayan güçlü bir görselleştirme çevrimiçi analitik süreç sistemidir.

## OLAP'ın dezavantajları

- OLAP, verilerin bir yıldız veya kar tanesi şemasında düzenlenmesini gerektirir. Bu şemaların uygulanması ve yönetilmesi karmaşıktır
- Tek bir OLAP küpünde çok sayıda boyuta sahip olamazsınız
- OLAP sistemi ile işlem verilerine erişilemez.
- Bir OLAP küpündeki herhangi bir değişiklik, küpün tam bir güncellemesine ihtiyaç duyar. Bu, zaman alan bir süreçtir

### Özet:

- OLAP, analistlerin iş verilerini farklı bakış açılarından çıkarmasını ve görüntülemesini sağlayan bir teknolojidir.
- OLAP konseptinin temelinde bir OLAP Küpü var.
- Çeşitli iş uygulamaları ve diğer veri işlemleri OLAP Cube kullanımını gerektirir.
- OLAP 1) Toplama 2) Detaya inme 3) Dilim 4) Zar ve 5) Pivotta birincil beş tür analitik işlem vardır
- Yaygın olarak kullanılan üç tür OLAP sistemi MOLAP, ROLAP ve Hybrid OLAP'dir.
- Masaüstü OLAP, Web OLAP ve Mobil OLAP, diğer bazı OLAP sistem türleridir.

## MOLAP: Veri Ambarında Çok Boyutlu OLAP

### MOLAP nedir?

**Çok boyutlu OLAP (MOLAP)**, çok boyutlu bir veri küpü kullanarak veri analizini kolaylaştıran klasik bir OLAP'dir. Veriler önceden hesaplanır, yeniden özetlenir ve bir MOLAP'ta saklanır (ROLAP'tan önemli bir fark). Bir MOLAP kullanarak, bir kullanıcı farklı yönlerde sahip çok boyutlu görünüm verilerini kullanabilir.

İlişkisel bir veritabanı kullanılıyorsa çok boyutlu veri analizi de mümkündür. Bu, birden çok tablodan veri sorgulama gerektirir. Aksine, MOLAP, çok boyutlu bir dizide depolanan tüm olası veri kombinasyonlarına sahiptir. MOLAP bu verilere doğrudan erişebilir. Bu nedenle, MOLAP, ilişkisel Çevrimiçi Analitik İşleme (ROLAP) ile karşılaştırıldığında daha hızlıdır.

### Anahtar noktaları

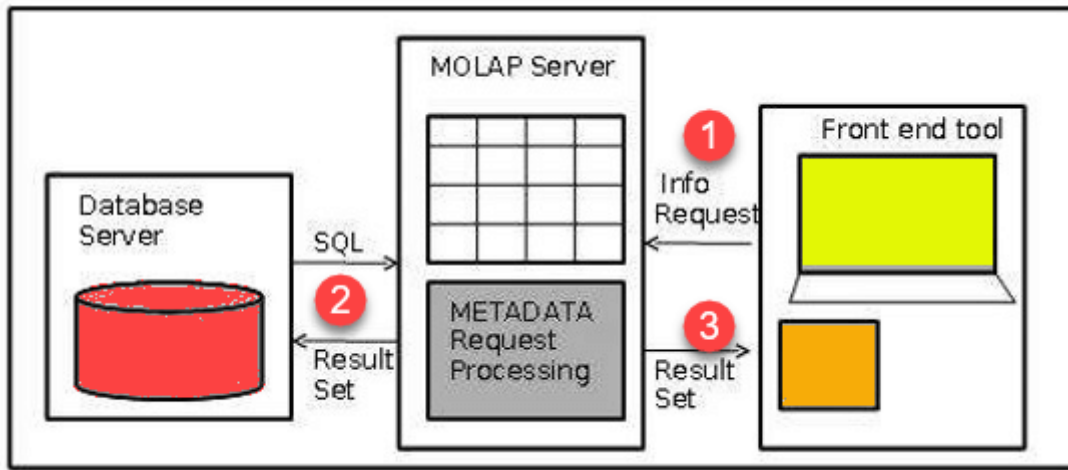
- MOLAP'ta işlemlere işleme adı verilir.
- MOLAP araçları, özetleme düzeyine bakılmaksızın bilgileri aynı miktarda yanıt süresi ile işler.

- MOLAP araçları, analiz için verileri depolamak üzere ilişkisel bir veritabanı tasarılmanın karmaşıklığını ortadan kaldırır.
- MOLAP sunucusu, yoğun ve seyrek veri kümelerini yönetmek için iki düzeyli depolama gösterimi uygular.
- Veri seti seyrekse depolama kullanımı düşük olabilir.
- Gerçekler, çok boyutlu dizilerde ve bunları sorgulamak için kullanılan boyutlarda saklanır.

## MOLAP Mimarisi

MOLAP Mimarisi aşağıdaki bileşenleri içerir

- Veritabanı sunucusu.
- MOLAP sunucusu.
- Ön uç aracı.



Yukarıda MOLAP Mimarilerini düşünün:

1. Arayüz üzerinden kullanıcı talep raporları
2. MDDB'nin uygulama mantığı katmanı, depolanan verileri Veritabanından alır
3. Uygulama mantık katmanı, sonucu istemciye / kullanıcıya iletir.

MOLAP mimarisi esas olarak önceden derlenmiş verileri okur. MOLAP mimarisi, dinamik olarak toplamalar oluşturmak veya önceden hesaplanıp depolanmamış sonuçları hesaplamak için sınırlı yeteneklere sahiptir.

Örneğin, bir muhasebe müdürü, belirli bir yıl kuruluş için kurumsal K / K hesabını veya K / K hesabını gösteren bir rapor çalıştırabilir. MDDB, önceden derlenmiş Kar ve Zarar rakamlarını alır ve bu sonucu kullanıcıya gösterir.

## MOLAP Uygulama konuları

MOLAP'ta, küpler oluşturmak için strateji oluştururken hem bakım hem de depolama etkilerini dikkate almak çok önemlidir.

- MOLAP'ı sorgulamak için kullanılan tescilli diller. Ancak, örneğin Microsoft'un MDX'i gibi kapsamlı tıklama ve sürükleme desteği içerir.
- Boyutlar arttığında gereken küp sayısı ve boyutu gerektiğinden ölçeklendirilmesi zordur.
- API'ler küplerin araştırılmasını sağlamalıdır.
- Veri analizlerinin birden çok konu alanını desteklemek için veri yapısı, hangi verilerin gezinebileceği ve analiz edilebileceği. Gezinme değiştiğinde, veri yapısının fiziksel olarak yeniden düzenlenmesi gerekir.
- Veritabanı yöneticisinin veritabanını oluşturması ve sürdürmesi için farklı beceri setlerine ve araçlara ihtiyaç vardır.

## MOLAP Avantajları

- MOLAP, önemli miktarda çok boyutlu veriyi yönetebilir, analiz edebilir ve depolayabilir.
- Optimize edilmiş depolama, indeksleme ve önbelleğe alma sayesinde Hızlı Sorgu Performansı.
- İlişkisel veritabanına kıyasla daha küçük veri boyutları.
- Daha yüksek düzeyde toplu verilerin otomatik olarak hesaplanması.
- Kullanıcıların daha büyük, daha az tanımlı verileri analiz etmesine yardımcı olun.
- MOLAP kullanıcı için daha kolaydır, bu nedenle deneyimsiz kullanıcılar için uygun bir modeldir.
- MOLAP küpleri, hızlı veri alımı için oluşturulmuştur ve dilimleme ve dilimleme işlemleri için idealdir.
- Küp oluşturulduğunda tüm hesaplamalar önceden oluşturulur.

## MOLAP Dezavantajları

- MOLAP'ın önemli bir zayıflığı, yalnızca sınırlı miktarda veriyi işlediği için ROLAP'tan daha az ölçeklenebilir olmasıdır.
- MOLAP ayrıca kaynak yoğun olduğu için veri yedekliliği sunar
- MOLAP Çözümleri, özellikle büyük veri hacimlerinde uzun olabilir.
- MOLAP ürünleri, boyutlar ondan fazla olduğunda modelleri güncellerken ve sorgularken sorunlarla karşılaşabilir.
- MOLAP ayrıntılı veri içeremez.
- Veri seti çok dağınıksa depolama kullanımı düşük olabilir.
- Yalnızca sınırlı miktarda veriyi işleyebilir, bu nedenle küpün kendisine büyük miktarda veri dahil etmek imkansızdır.

## MOLAP Araçları

- [Essbase](#) - Oracle'dan çok boyutlu bir veritabanına sahip araçlar.
- [Express Server](#) - Oracle veritabanı üzerinde çalışan web tabanlı ortam.
- [Yellowfin](#) - Raporlar ve gösterge tabloları oluşturmak için iş analizi araçları.

- [Clear Analytics](#) - Clear analytics, Excel tabanlı bir iş çözümüdür.
- [SAP Business Intelligence](#) - [SAP'den iş](#) analitiği çözümleri

## Özet:

- Çok boyutlu OLAP (MOLAP), çok boyutlu bir veri küpü kullanarak veri analizini kolaylaştıran klasik bir OLAP'dir.
- MOLAP araçları, özetleme düzeyine bakılmaksızın bilgileri aynı miktarda yanıt süresi ile işler.
- MOLAP sunucusu, yoğun ve seyrek veri kümelerini yönetmek için iki düzeyli depolama uygular.
- MOLAP, önemli miktarda çok boyutlu veriyi yönetebilir, analiz edebilir ve depolayabilir.
- Daha yüksek düzeyde toplu verilerin hesaplanmasını otomatikleştirmeye yardımcı olur
- Yalnızca sınırlı miktarda veriyi işlediği için ROLAP'tan daha az ölçeklenebilir.

## OLTP nedir?

**OLTP**, 3 katmanlı bir mimaride işlem odaklı uygulamaları destekleyen bir işletim sistemidir. Bir organizasyonun günlük işlemlerini yönetir. OLTP temel olarak sorgu işleme, çoklu erişim ortamlarında veri bütünlüğünü korumanın yanı sıra saniyedeki toplam işlem sayısı ile ölçülen etkililiğe odaklanır. OLTP'nin tam biçimi Çevrimiçi İşlem İşlemedir.

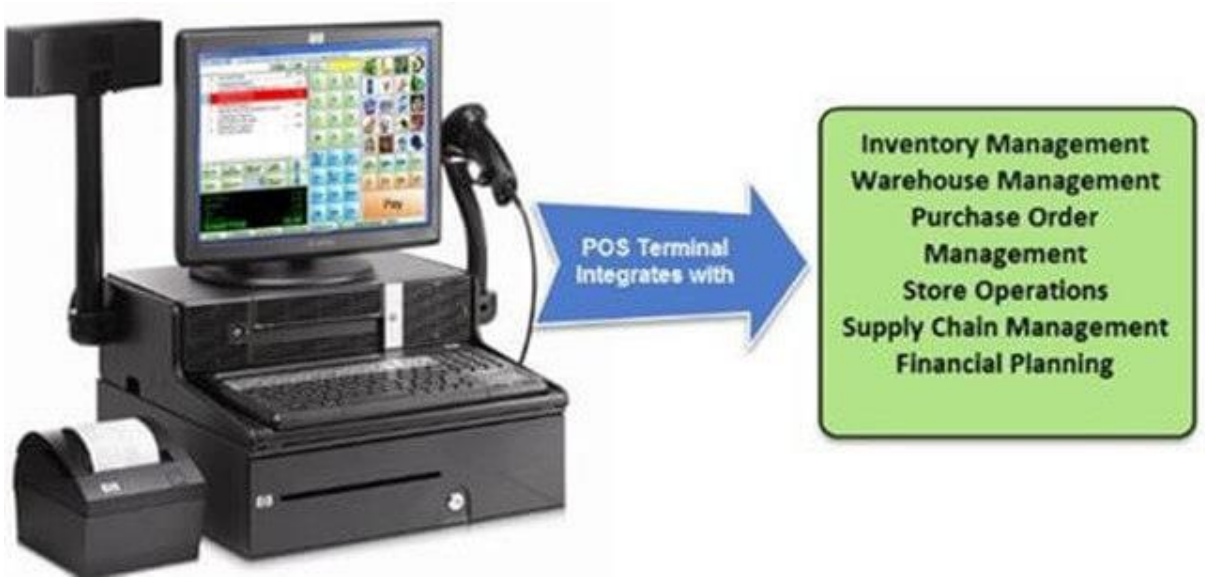
## OLTP'nin özellikleri

OLTP'nin önemli özellikleri şunlardır:

- OLTP, küçük miktarlarda veri içeren işlemleri kullanır.
- Veritabanındaki indekslenmiş verilere kolaylıkla erişilebilir.
- OLTP'nin çok sayıda kullanıcısı vardır.
- Hızlı tepki sürelerine sahiptir
- Veritabanlarına doğrudan son kullanıcılar erişebilir
- OLTP, veritabanı tutarlılığı için tamamen normalleştirilmiş bir şema kullanır.
- OLTP sisteminin yanıt süresi kısadır.
- Yalnızca az sayıda kayıt üzerinde önceden tanımlanmış işlemleri kesinlikle gerçekleştirir.
- OLTP, son birkaç günün veya bir haftanın kayıtlarını depolar.
- Karmaşık veri modellerini ve tabloları destekler.

## Bir OLTP sisteminin işleyebileceği sorgu türleri:

OLTP sistemi, çevrimiçi bir veritabanı değiştirme sistemidir. Bu nedenle, veritabanından bilgi ekleme, güncelleme ve silme gibi veritabanı sorgusunu destekler.



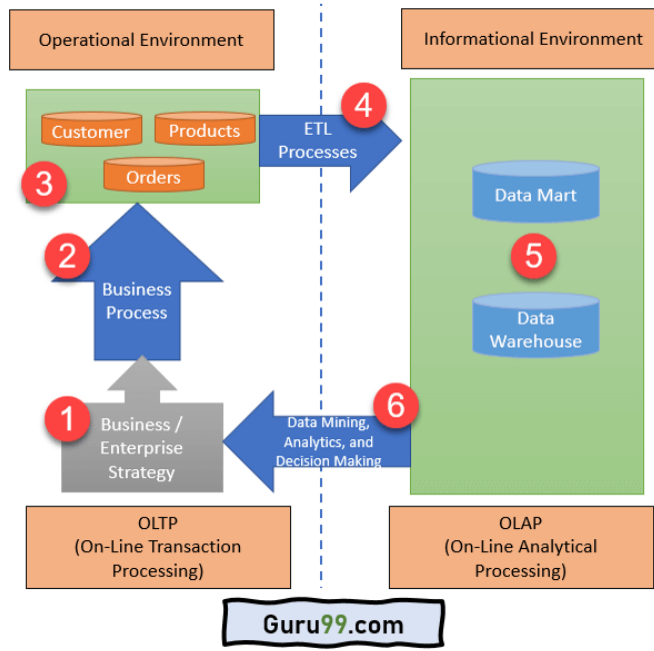
OLTP için POS sistemi

Bir süpermarketin satış noktası sistemini düşünün, bu sistemin işleyebileceği örnek sorgular şunlardır:

- Belirli bir ürünün açıklamasını almak.
- Tedarikçi ile ilgili tüm ürünleri filtrelemek.
- Müşterinin kaydının aranması.
- Beklenen miktardan daha düşük bir fiyata sahip ürünleri listelemek.

## OLTP mimarisi

İşte OLTP'nin mimarisi:



OLTP Mimarisi

- İşletme / İşletme Stratejisi:** İşletme stratejisi, organizasyonu bir bütün olarak etkileyen konularla ilgilenir. OLTP'de, genellikle yönetim kurulu veya üst yönetim tarafından firma içinde üst düzeyde geliştirilir.
- İş Süreci:** OLTP iş süreci, tamamlandıktan sonra organizasyonel bir hedefi gerçekleştirecek bir dizi faaliyet ve görevdir.
- Müşteriler, Siparişler ve Ürünler:** OLTP veritabanı, ürünler, siparişler (işlemler), müşteriler (alıcılar), tedarikçiler (satıcılar) ve çalışanlar hakkındaki bilgileri depolar.
- ETL İşlemleri:** Çeşitli RDBMS kaynak sistemlerinden verileri ayırır, ardından verileri dönüştürür (birleştirme, hesaplamalar vb.) Ve işlenen verileri Veri Ambarı sistemine yükler.
- Veri Mart ve Veri ambarı:** Veri pazarı, veri ambarı ortamlarına özgü bir yapı / erişim modelidir. OLAP tarafından işlenmiş verileri depolamak için kullanılır.
- Veri Madenciliği, Analitik ve Karar Verme:** Veri pazarı ve veri ambarında depolanan veriler, veri madenciliği, analitik ve karar verme için kullanılabilir.

Bu veriler, veri modellerini keşfetmenize, ham verileri analiz etmenize ve kuruluşunuzun büyümesi için analitik kararlar almanıza yardımcı olur.

## OLTP İřlemi rneęi

OLTP sistemine bir rnek ATM merkezidir. Bir iftin bir banka ile ortak bir hesabı olduęunu varsayalım. Bir gn her ikisi de aynı anda farklı ATM merkezlerine tam olarak aynı anda ulařıyor ve banka hesaplarında bulunan toplam tutarı ekmek istiyor.



ATM grnts iin OLTOP

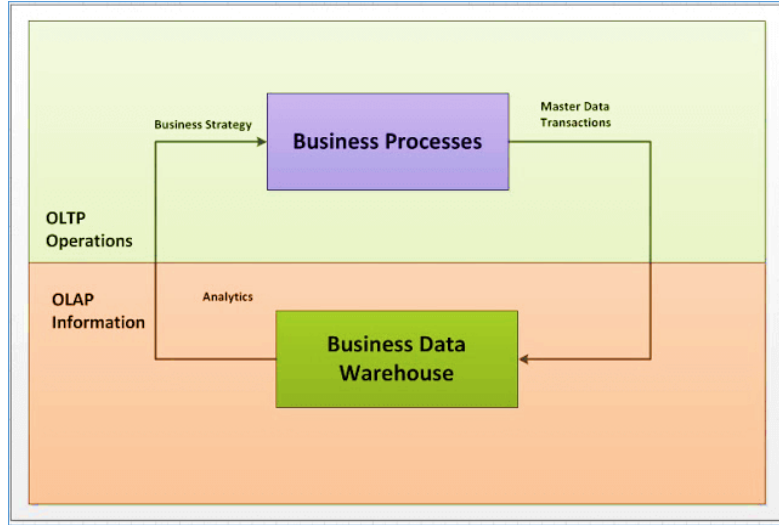
Ancak, kimlik doęrulama srecini ilk tamamlayan kiři para kazanabilecektir. Bu durumda OLTP sistemi, ekilen miktarın asla bankada bulunan miktardan fazla olmamasını saęlar. Burada dikkat edilmesi gereken anahtar nokta, OLTP sistemlerinin veri analizi yerine iřlem stnlę iin optimize edilmiř olmasıdır.

OLTP sisteminin dięer rnekleri:

- Online bankacılık
- Online uak bileti rezervasyonu
- Bir metin mesajı gnderme
- Sipariř giriři
- Alıřveriř sepetine kitap ekle



## OLTP ve OLAP



OLTP ve OLAP arasındaki önemli fark şu şekildedir:

OLTP	OLAP
OLTP, çevrimiçi bir işlem sistemidir.	OLAP, çevrimiçi bir analiz ve veri alma işlemidir.
Çok sayıda kısa çevrimiçi işlem ile karakterizedir.	Büyük miktarda veri ile karakterizedir.
OLTP, çevrimiçi bir veritabanı değiştirme sistemidir.	OLAP, çevrimiçi bir veritabanı sorgu yönetim sistemidir.
OLTP, geleneksel DBMS kullanır.	OLAP, veri ambarını kullanır.
Veritabanından bilgi Ekleme, Güncelleme ve Silme.	Çoğunlukla operasyonları seçin
OLTP ve işlemleri veri kaynaklarıdır.	OLAP için veri kaynağı, farklı OLTP veritabanları haline gelir.
OLTP veritabanı, veri bütünlüğü kısıtlamalarını korumalıdır.	OLAP veritabanı sık sık değiştirilmez. Dolayısıyla, veri bütünlüğü bir sorun değildir.
Yanıt süresi bir milisaniyedir.	Saniyelerden dakikalara yanıt süresi.
OLTP veritabanındaki veriler her zaman ayrıntılı ve düzenlidir.	OLAP sürecindeki veriler organize edilmemiş olabilir.
Okuma / yazma işlemlerine izin verin.	Sadece okuyun ve nadiren yazın.

Pazar odaklı bir süreçtir.	Müşteri odaklı bir süreçtir.
Bu süreçteki sorgular standartlaştırılmış ve basittir.	Toplamaları içeren karmaşık sorgular.
Artımlı yedeklemelerle birleştirilmiş verilerin tam yedeklemesi.	OLAP yalnızca zaman zaman yedeklemeye ihtiyaç duyar. OLTP ile karşılaştırıldığında yedekleme önemli değil
DB tasarımı uygulama odaklı bir örnektir: Veritabanı tasarımı perakende, havayolu, bankacılık vb. Sektörle birlikte değişir.	DB tasarımı konu odaklıdır. Örnek: Veritabanı tasarımı satış, pazarlama, satın alma vb. Konularla değişir.
Katip, DBA ve Veri Tabanı uzmanları gibi Veri açısından kritik olan kullanıcılar tarafından kullanılır.	Çalışanlar, yöneticiler ve CEO gibi Veri bilgisi kullanıcıları tarafından kullanılır.
Gerçek zamanlı iş operasyonları için tasarlanmıştır.	Kategori ve niteliklere göre iş önlemlerinin analizi için tasarlanmıştır.
İşlem hacmi, performans metriğidir	Sorgu işleme hızı, performans metriğidir.
Bu tür bir Veritabanı kullanıcısı binlerce kullanıcıya izin verir.	Bu tür bir Veritabanı yalnızca yüzlerce kullanıcıya izin verir.
Kullanıcının self servisini ve üretkenliğini artırmaya yardımcı olur	İş analistlerinin üretkenliğini artırmaya yardımcı olun.
Bu süreçteki sorgular standartlaştırılmış ve basittir.	Toplamaları içeren karmaşık sorgular.
Artımlı yedeklemelerle birleştirilmiş verilerin tam yedeklemesi.	OLAP yalnızca zaman zaman yedeklemeye ihtiyaç duyar. OLTP ile karşılaştırıldığında yedekleme önemli değil
DB tasarımı uygulama odaklı bir örnektir: Veritabanı tasarımı perakende, havayolu, bankacılık vb. Sektörle birlikte değişir.	DB tasarımı konu odaklıdır. Örnek: Veritabanı tasarımı satış, pazarlama, satın alma vb. Konularla değişir.
Katip, DBA ve Veri Tabanı uzmanları gibi Veri açısından kritik olan kullanıcılar tarafından kullanılır.	Çalışanlar, yöneticiler ve CEO gibi Veri bilgisi kullanıcıları tarafından kullanılır.
Gerçek zamanlı iş operasyonları için tasarlanmıştır.	Kategori ve niteliklere göre iş önlemlerinin analizi için tasarlanmıştır.
İşlem hacmi, performans metriğidir	Sorgu işleme hızı, performans metriğidir.
Bu tür bir Veritabanı kullanıcısı binlerce kullanıcıya izin verir.	Bu tür bir Veritabanı yalnızca yüzlerce kullanıcıya izin verir.
Kullanıcının self servisini ve üretkenliğini artırmaya yardımcı olur	İş analistlerinin üretkenliğini artırmaya yardımcı olun.
Veri Ambarları, tarihsel olarak, inşa edilmesi maliyetli olabilecek bir geliştirme projesi olmuştur.	OLAP küpü, açık bir SQL sunucusu veri ambarı değildir. Bu nedenle, OLAP sunucusunu yönetmek için teknik bilgi ve deneyim gereklidir.
Günlük kullanılan veriler için hızlı sonuç sağlar.	Sorguya verilen yanıtın tutarlı bir şekilde daha hızlı olmasını sağlar.
Oluşturması ve bakımı kolaydır.	Kullanıcının bir elektronik tablo yardımıyla bir görünüm oluşturmalarını sağlar.
OLTP, hızlı yanıt süresine, düşük veri yedekliliğine sahip olacak şekilde tasarlanmıştır ve normalleştirilmiştir.	Bir veri ambarı, konsolide bir veritabanı oluşturmak için farklı veri kaynaklarını entegre edebilmesi için benzersiz bir şekilde oluşturulur.

## OLTP'nin Avantajları:

OLTP sisteminin avantajları / faydaları şunlardır:

- OLTP, gelir ve gider için doğru tahmin sunar.
- Tüm işlemlerin zamanında değiştirilmesi nedeniyle istikrarlı bir işletme / organizasyon için sağlam bir temel sağlar.
- OLTP, müşteriler adına işlemleri çok daha kolaylaştırır.
- Bireysel süreçleri hızlandırarak ve basitleştirerek bir organizasyon için müşteri tabanını genişletir.
- OLTP, daha büyük veritabanları için destek sağlar.
- Veri işleme için verilerin bölümlenmesi kolaydır.

- Sistem tarafından sıklıkla gerçekleştirilen görevleri kullanmak için OLTP'ye ihtiyacımız var.
- Sadece az sayıda kayda ihtiyacımız olduğunda.
- Verilerin eklenmesini, güncellenmesini veya silinmesini içeren görevler.
- Daha fazla kullanılabilirlik sağlayan görevleri gerçekleştirmek için tutarlılık ve eşzamanlılığa ihtiyaç duyduğunuzda kullanılır.

## OLTP'nin dezavantajları

OLTP sisteminin eksileri / dezavantajları:

- OLTP sistemi donanım arızalarıyla karşılaşır, çevrimiçi işlemler ciddi şekilde etkilenir.
- OLTP sistemleri, birden çok kullanıcının aynı verilere aynı anda erişmesine ve değiştirmesine izin verir, bu da çoğu zaman benzeri görülmemiş bir durum yaratır.
- Sunucu saniyeler boyunca askıda kalırsa, çok sayıda işlemi etkileyebilir.
- OLTP, envanteri korumak için çok sayıda personelin gruplar halinde çalışmasını gerektiriyordu.
- Çevrimiçi İşlem İşleme Sistemleri, ürünleri alıcılara kendi başlarına aktarmak için uygun yöntemlere sahip değildir.
- OLTP, veritabanını bilgisayar korsanlarına ve davetsiz misafirlere karşı çok daha duyarlı hale getirir.
- B2B işlemlerinde, hem alıcıların hem de tedarikçilerin sistemin sunduğu verimlilik avantajlarını gözden kaçırma ihtimali vardır.
- Sunucu hatası, veritabanından büyük miktarda verinin silinmesine neden olabilir.
- Sınırlı sayıda sorgu ve güncelleme yapabilirsiniz.

## OLTP Sisteminin Zorlukları

- Birden fazla kullanıcının aynı verilere aynı anda erişmesine ve değiştirmesine izin verir. Bu nedenle, benzeri görülmemiş durumlardan kaçınmak için eşzamanlılık kontrolü ve kurtarma tekniği gerektirir.
- OLTP sistem verileri karar vermek için uygun değildir. OLAP sistemlerinin verilerini "ne olursa" analizi veya karar verme için kullanmanız gerekir.

## Özet

- OLTP, 3 katmanlı bir mimaride işlem odaklı uygulamaları destekleyen bir işletim sistemi olarak tanımlanır.
- OLTP, küçük miktarlarda veri içeren işlemleri kullanır.
- OLTP sistemi, çevrimiçi bir veritabanı değiştirme sistemidir.
- OLTP mimarisi

1) İşletme / İşletme Stratejisi,

2) İş Süreci,

3) Müşteriler, Siparişler ve Ürünler,

4) ETL Süreçleri,

5) Veri Mart ve Veri ambarı,

6) Veri Madenciliği, Analitik ve Karar Verme içerir.

- OLTP çevrimiçi bir işlem sistemidir, OLAP ise çevrimiçi bir analiz ve veri alma sürecidir.
- OLTP, tüm işlemlerin zamanında değiştirilmesi nedeniyle istikrarlı bir işletme / organizasyon için sağlam bir temel sağlar.
- OLTP sistemleri, birden çok kullanıcının aynı verilere aynı anda erişmesine ve değiştirmesine izin verir, bu da çoğu zaman benzeri görülmemiş bir durum yaratır.

## Veri Ambarında Boyutsal Modelleme Nedir?

### Boyutsal Modelleme nedir?

**BOYUTLU MODELLEME (DM)**, bir Veri ambarında veri depolaması için optimize edilmiş bir veri yapısı tekniğidir. Boyutsal modelin amacı, verilere hızlı erişim için veritabanını optimize etmektir. Boyutsal Modelleme kavramı, Ralph Kimball tarafından geliştirilmiştir ve "olgu" ve "boyut" tablolarından oluşmaktadır.

Boyutsal bir model, bir veri ambarındaki değerler, teraziler, sayımlar, ağırlıklar vb. Gibi sayısal bilgileri okumak, özetlemek ve analiz etmek için tasarlanmıştır. Bunun aksine ilişki modelleri, gerçek zamanlı bir Çevrimiçi İşlem Sisteminde verilerin eklenmesi, güncellenmesi ve silinmesi için optimize edilmiştir.

Bu boyutsal ve ilişkisel modeller, belirli avantajları olan benzersiz veri depolama yöntemlerine sahiptir.

Örneğin, ilişkisel modda, normalleştirme ve ER modelleri verilerdeki fazlalığı azaltır. Aksine boyutsal model, veriyi, bilgiye erişilmesi ve raporların üretilmesi daha kolay olacak şekilde düzenler.

Bu nedenle, Boyutsal modeller veri ambarı sistemlerinde kullanılır ve ilişkisel sistemler için uygun değildir.

## Boyutlu Veri Modelinin Öğeleri

### Gerçek

Gerçekler, iş sürecinize ait ölçümler / metrikler veya gerçeklerdir. Bir Satış iş süreci için bir ölçüm, üç aylık satış sayısı olacaktır

### Boyut

Dimension, bir iş süreci olayını çevreleyen bağlamı sağlar. Basit bir ifadeyle, kim, ne, nerede bir gerçeği veriyorlar. Satış iş sürecinde, üç aylık satış sayısı için boyutlar şu şekilde olacaktır:

- Kim - Müşteri Adları
- Nerede - Konum
- Ne - Ürün Adı

Başka bir deyişle, boyut, gerçeklerdeki bilgileri görüntülemek için bir penceredir.

### Öznitelikler

Nitelikler, boyutun çeşitli özellikleridir.

Konum boyutunda öznitelikler olabilir

- Durum
- Ülke
- Posta kodu vb.

Nitelikler, gerçekleri aramak, filtrelemek veya sınıflandırmak için kullanılır. Boyut Tabloları Öznitelikleri içerir

### Olgu Tablosu

Olgu tablosu, boyutlu bir modeldeki birincil tablodur.

Bir Bilgi Tablosu şunları içerir:

1. Ölçümler / gerçekler
2. Boyut tablosunun yabancı anahtarı

## Boyut tablosu

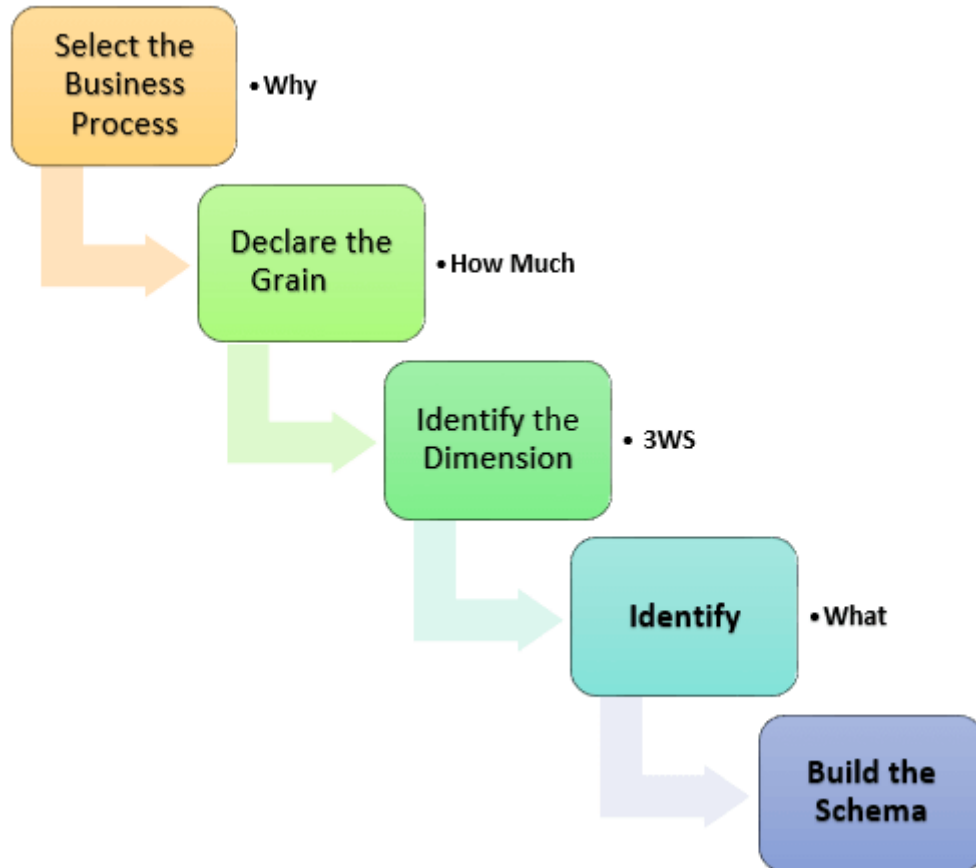
- Bir boyut tablosu bir olgunun boyutlarını içerir.
- Bir yabancı anahtar aracılığıyla olgu tablosuna birleştirilirler.
- Boyut tabloları normalleştirilmiş tablolardır.
- Boyut Özellikleri, bir boyut tablosundaki çeşitli sütunlardır
- Boyutlar, niteliklerinin yardımıyla olguların tanımlayıcı özelliklerini sunar
- Boyut sayısı için belirlenmiş bir sınır belirlenmedi
- Boyut ayrıca bir veya daha fazla hiyerarşik ilişki içerebilir

## Boyutsal Modelleme Adımları

Boyut modellemenizi oluşturmanın doğruluğu, veri ambarı uygulamanızın başarısını belirler. İşte Boyut Modeli oluşturma adımları

1. İş Sürecini Tanımlayın
2. Tahıl Tanımlayın (ayrıntı düzeyi)
3. Boyutları Tanımlayın
4. Gerçekleri Belirleyin
5. Yıldız İnşa Et

Model, iş sürecinizin Neden, Ne Kadar, Ne Zaman / Nerede / Kim ve Ne olduğunu açıklamalıdır.



## Adım 1) İş sürecini tanımlayın

Bir veri merkezinin kapsaması gereken gerçek iş sürecini tanımlama. Bu, organizasyonun veri analizi ihtiyaçlarına göre Pazarlama, Satış, İK vb. Olabilir. İş sürecinin seçimi aynı zamanda o süreç için mevcut verilerin kalitesine de bağlıdır. Veri Modelleme sürecinin en önemli adımıdır ve buradaki bir başarısızlık, kademeli ve onarılamaz kusurlara sahip olacaktır.

İş sürecini açıklamak için düz metin kullanabilir veya temel İş Süreci Modelleme Gösterimi (BPMN) veya Birleşik Modelleme Dili (UML) kullanabilirsiniz.

## Adım 2) Tahıl tanımlayın

Tahıl, iş problemi / çözümü için ayrıntı düzeyini tanımlar. Veri ambarınızdaki herhangi bir tablo için en düşük bilgi seviyesini belirleme işlemidir. Bir tablo her gün için satış verilerini içeriyorsa, günlük ayrıntı düzeyi olmalıdır. Bir tablo her ay için toplam satış verilerini içeriyorsa, aylık ayrıntı düzeyine sahiptir.

Bu aşamada şu soruları cevaplarsınız:

1. Mevcut tüm ürünleri mi yoksa sadece birkaç tür ürünü mi depolamamız gerekiyor? Bu karar, Datawarehouse için seçilen iş süreçlerine dayanmaktadır.
2. Ürün satış bilgilerini aylık, haftalık, günlük veya saatlik olarak saklıyor muyuz? Bu karar, yöneticiler tarafından talep edilen raporların niteliğine bağlıdır.
3. Yukarıdaki iki seçenek veritabanı boyutunu nasıl etkiler?

### Tahıl Örneği:

Bir çokuluslu şirketin CEO'su, belirli ürünlerin satışlarını farklı lokasyonlarda günlük olarak bulmak istiyor.

Yani tahıl, "konuma göre güne göre ürün satış bilgisi" dir.

## Adım 3) Boyutları belirleyin

Boyutlar tarih, mağaza, envanter vb. Gibi isimlerdir. Bu boyutlar, tüm verilerin depolanması gereken yerdir. Örneğin, tarih boyutu yıl, ay ve haftanın günü gibi verileri içerebilir.

### Boyut Örneği:

Bir çokuluslu şirketin CEO'su, belirli ürünlerin satışlarını farklı lokasyonlarda günlük olarak bulmak istiyor.

Boyutlar: Ürün, Konum ve Zaman

Özellikler: Ürün için: Ürün anahtarı (Yabancı Anahtar), Ad, Tür, Özellikler

Hiyerarşiler: Konum için: Ülke, Eyalet, Şehir, Sokak Adresi, Ad

## Adım 4) Gerçeği Belirleyin

Bu adım, sistemin iş kullanıcıları ile ortak ilişkilidir, çünkü burası veri ambarında depolanan verilere erişirler. Olgu tablosu satırlarının çoğu, fiyat veya birim başına maliyet gibi sayısal değerlerdir.

### Gerçek Örnekleri:

Bir çokuluslu şirketin CEO'su, belirli ürünlerin satışlarını farklı lokasyonlarda günlük olarak bulmak istiyor.

Buradaki gerçek, konuma göre ürüne göre Satışların Toplamıdır.

## Adım 5) Şema Oluşturun

Bu adımda, Boyut Modelini uygularsınız. Şema, veritabanı yapısından başka bir şey değildir (tabloların düzenlenmesi). İki popüler şema var

### 1. Yıldız Şeması

Yıldız şema mimarisinin tasarımı kolaydır. Yıldız şeması olarak adlandırılır çünkü diyagram, merkezden yayılan noktalarla bir yıldıza benzer. Yıldızın merkezi olgu tablosundan oluşur ve yıldızın noktaları boyut tablolarıdır.

Üçüncü normal biçim olan yıldız şemasındaki olgu tabloları, boyutsal tablolar ise normalleştirilmemiştir.

### 2. Kar Tanesi Şeması

Kar tanesi şeması, yıldız şemasının bir uzantısıdır. Bir kar tanesi şemasında, her boyut normalleştirilir ve daha fazla boyut tablosuna bağlanır.

## Boyutsal Modelleme Kuralları

- Atomik verileri boyutlu yapılara yükleyin.
- İş süreçleri etrafında boyutsal modeller oluşturun.
- Her olgu tablosunun ilişkili bir tarih boyut tablosuna sahip olduğundan emin olmanız gerekir.
- Tek bir olgu tablosundaki tüm bilgilerin aynı ayrıntıda veya ayrıntı düzeyinde olmasını sağlayın.
- Rapor etiketlerini depolamak ve alan değerlerini boyut tablolarında filtrelemek önemlidir
- Boyut tablolarının bir vekil anahtar kullandığından emin olmanız gerekir
- Karar verme süreçlerini desteklemek için iş çözümü sunmak için gereksinimleri ve gerçekleri sürekli olarak dengeleyin



## Boyutsal modellemenin faydaları

- Boyutların standardizasyonu, işin tüm alanlarında kolay raporlama sağlar.
- Boyut tabloları, boyutsal bilgilerin geçmişini depolar.
- Olgu tablosuna büyük aksamalar olmadan tamamen yeni bir boyut getirilmesine izin verir.
- Ayrıca, verileri veri tabanında depolandıktan sonra verilerden geri almak daha kolay olacak şekilde verileri depolamak için boyutsal.
- Normalleştirilmiş model boyut tablosu ile karşılaştırıldığında anlaşılması daha kolaydır.
- Bilgi, açık ve basit iş kategorileri halinde gruplandırılmıştır.
- Boyutsal model işletme tarafından çok anlaşılabilir. Bu model iş şartlarına dayalıdır, böylece işletme her gerçeğin, boyutun veya özelliğin ne anlama geldiğini bilir.
- Hızlı veri sorgulama için boyutsal modeller deforme edilmiş ve optimize edilmiştir. Birçok ilişkisel veritabanı platformu bu modeli tanır ve performansa yardımcı olmak için sorgu yürütme planlarını optimize eder.
- Boyutsal modelleme, yüksek performans için optimize edilmiş bir şema oluşturur. Daha az birleşme anlamına gelir ve en aza indirilmiş veri yedekliliğine yardımcı olur.
- Boyutsal model ayrıca sorgu performansını artırmaya yardımcı olur. Daha denormalize edilmiş olduğundan sorgulama için optimize edilmiştir.
- Boyutlu modeller değişime rahatlıkla uyum sağlayabilir. Bu tabloları kullanan mevcut iş zekası uygulamalarını etkilemeden boyut tablolarına daha fazla sütun eklenebilir.

## Özet:

- Boyutsal model, Veri ambarlama araçları için optimize edilmiş bir veri yapısı tekniğidir.
- Gerçekler, iş sürecinize ait ölçümler / metrikler veya gerçeklerdir.
- Dimension, bir iş süreci olayını çevreleyen bağlamı sağlar.
- Nitelikler, boyutun çeşitli özellikleridir.
- Olgu tablosu, boyutsal bir modeldeki birincil tablodur.
- Bir boyut tablosu bir olgunun boyutlarını içerir.
- Üç tür gerçek vardır 1. Katkı maddesi 2. Katkısız 3. Yarı katkı maddesi.
- Boyut Türleri Uygun, Payanda, Daraltılmış, Rol Oynama, Boyut Tablosu, Önemsiz, Dejenere, Değiştirilebilir ve Adım Boyutlarıdır.
- Boyutsal modellemenin beş adımı şunlardır: 1. İş Sürecini Belirleyin 2. Tahılları Belirleyin (ayrıntı düzeyi) 3. Boyutları Belirleyin 4. Gerçekleri Belirleyin 5. Yıldızı Oluşturun
- Boyutsal modellemede, her olgu tablosunun ilişkili bir tarih boyut tablosuna sahip olmasını sağlamaya ihtiyaç vardır.

## Çok Boyutlu şema nedir?

**Çok Boyutlu Şema**, özellikle veri ambarı sistemlerini modellemek için tasarlanmıştır. Şemalar, analitik amaç (OLAP) için tasarlanmış çok büyük veritabanlarının benzersiz ihtiyaçlarını karşılamak için tasarlanmıştır.

Veri Ambarı Şeması Türleri:

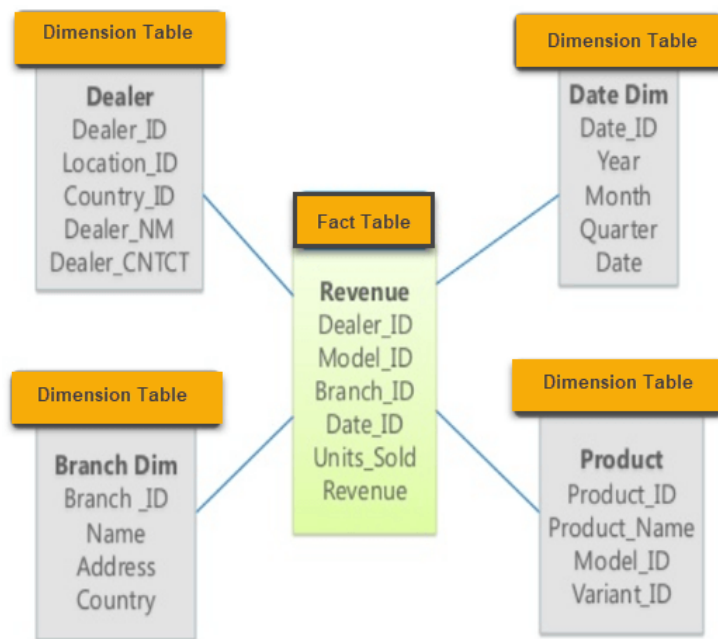
Aşağıda, her birinin kendine özgü avantajları olan 3 ana çok boyutlu şema türü yer almaktadır.

- Star (Yıldız) Şeması
- Snowflake (Kar Tanesi) Şeması
- Galaxy Şeması

## Star Schema (Yıldız Şeması) nedir?

**YILDIZ Şeması**, yıldızın merkezi bir gerçek tablo ve ilgili ölçü tabloları sayı olabilir. Yapısı bir yıldıza benzediği için yıldız şeması olarak bilinir. Yıldız şeması, Veri Ambarı şemasının en basit türüdür. Star Join Schema olarak da bilinir ve büyük veri kümelerini sorgulamak için optimize edilmiştir.

Aşağıdaki örnekte, Dealer\_ID, Model ID, Date\_ID, Product\_ID, Branch\_ID gibi her boyut tablosunun anahtarlarını ve Satılan birimler ve gelir gibi diğer öznitelikleri içeren olgu tablosu merkezdedir.



Yıldız Şeması Örneği

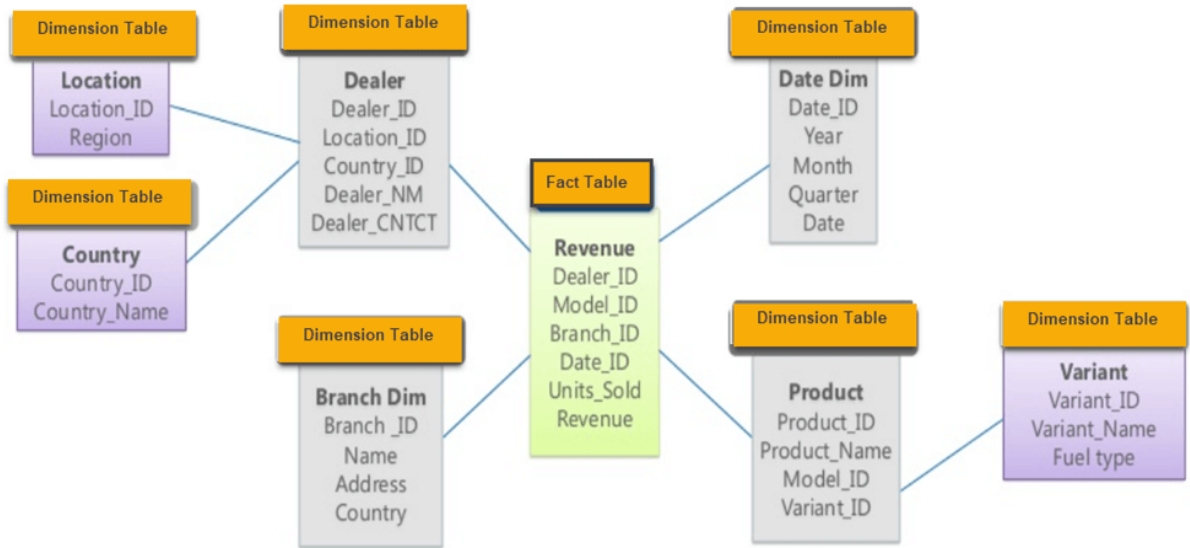
### Yıldız Şemasının Özellikleri:

- Bir yıldız şemasındaki her boyut, tek boyutlu tablo ile temsil edilir.
- Boyut tablosu, özellik kümesini içermelidir.
- Boyut tablosu, bir yabancı anahtar kullanılarak olgu tablosuna birleştirilir
- Boyut tablosu birbirine bağlı değil
- Olgu tablosu anahtar ve ölçü içerecektir
- Star şemasının anlaşılması kolaydır ve optimum disk kullanımı sağlar.
- Boyut tabloları **normalleştirilmemiştir**. Örneğin, yukarıdaki şekilde, Country\_ID, bir OLTP tasarımında olduğu gibi Ülke arama tablosuna sahip değildir.
- Şema, BI Tools tarafından yaygın olarak desteklenmektedir

### Snowflake (Kar Tanesi) Şeması nedir?

**SNOWFLAKE SCHEMA**, ER diyagramı bir kar tanesi şekline benzeyecek şekilde çok boyutlu bir veri tabanındaki tabloların mantıksal bir düzenlemesidir. Bir Kar Tanesi Şeması, bir Yıldız Şemasının bir uzantısıdır ve ek boyutlar ekler. Verileri ek tablolara bölen boyut tabloları **normalleştirilir**.

Aşağıdaki örnekte, Ülke ayrıca ayrı bir tablo halinde normalleştirilmiştir.



Kar Tanesi Şeması Örneği

### Kar Tanesi Şemasının Özellikleri:

- Kar tanesi şemasının ana yararı, daha küçük disk alanı kullanır.
- Şemaya bir boyutun uygulanması daha kolay eklenir
- Birden çok tablo nedeniyle sorgu performansı azalır

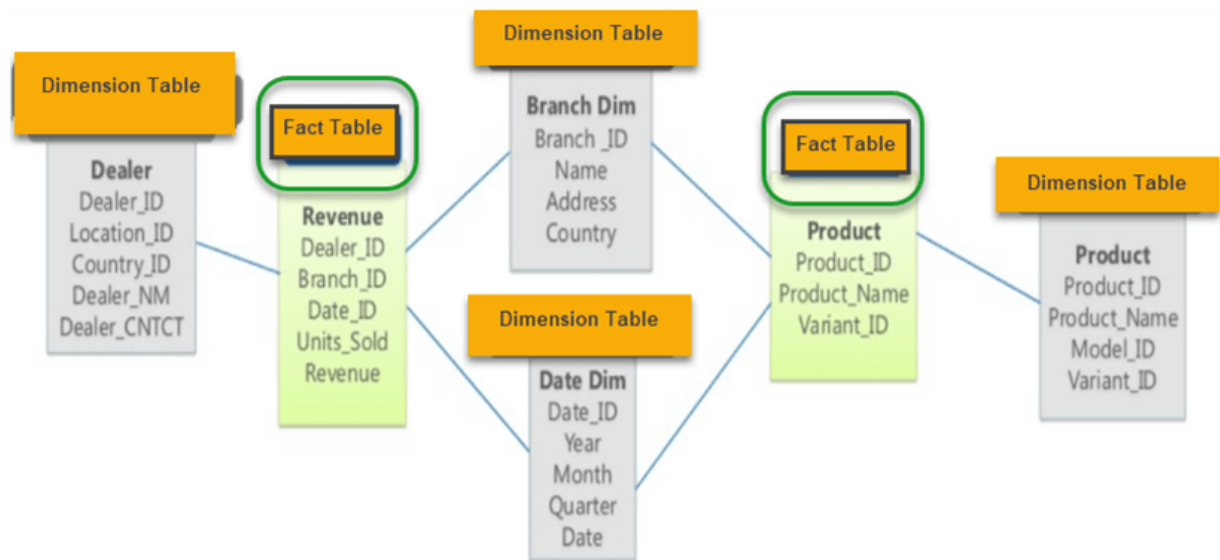
- Kar tanesi Şemasını kullanırken karşılaşacağınız başlıca zorluk, daha fazla arama tablosu nedeniyle daha fazla bakım çalışması yapmanız gerektirir.

## Star Vs Snowflake Şeması: Temel Farklılıklar

Yıldız Şeması	Kar Tanesi Şeması
Boyutlar için hiyerarşiler boyut tablosunda saklanır.	Hiyerarşiler ayrı tablolara bölünmüştür.
Boyut tablolarıyla çevrili bir olgu tablosu içerir.	Boyut tablosu ile çevrelenmiş bir olgu tablosu
Bir yıldız şemasında, yalnızca tek birleştirme olgu tablosu ile herhangi bir boyut tablosu arasındaki ilişkiyi oluşturur.	Bir kar tanesi şeması, verileri almak için birçok birleştirme gerektirir.
Basit DB Tasarımı.	Çok Karmaşık DB Tasarımı.
Normalleştirilmiş Veri yapısı ve sorgu da daha hızlı çalışır.	Normalleştirilmiş Veri Yapısı.
Yüksek düzeyde Veri yedekliliği	Çok düşük seviyeli veri yedekliliği
Tek Boyut tablosu, toplu verileri içerir.	Veriler farklı Boyut Tablolarına bölünür.
Küp işleme daha hızlıdır.	Küp işleme, karmaşık birleştirme nedeniyle yavaş olabilir.
Star Join Sorgu Optimizasyonunu kullanarak daha yüksek performanslı sorgular sunar. Tablolar birden çok boyutla birleştirilebilir.	Snow Flake Şeması, muhtemelen birden çok boyutla bağlantılı olmayan merkezi olgu tablosu ile temsil edilir.

## Galaxy şeması nedir?

Bir **GALAXY SCHEMA**, aralarında boyut tablolarını paylaşan iki olgu tablosu içerir. Fact Constellation Schema olarak da adlandırılır. Şema, bir yıldız koleksiyonu olarak görülüyor, dolayısıyla Galaxy Schema adını alıyor.



Galaxy Şeması Örneği

Yukarıdaki örnekte görebileceğiniz gibi, iki gerçekler tablosu var

1. Gelir
2. Ürün.

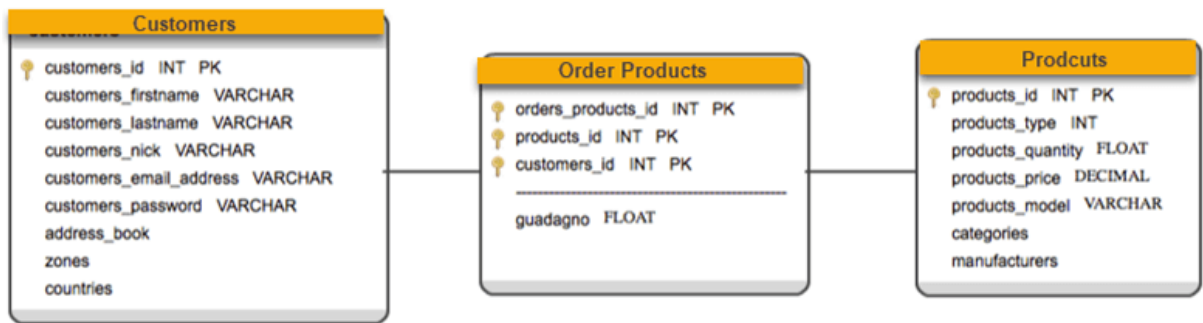
Galaxy şemasında paylaşım boyutları, Uyumlu Boyutlar olarak adlandırılır.

### Galaxy Şemasının Özellikleri:

- Bu şemadaki boyutlar, çeşitli hiyerarşi düzeylerine göre ayrı boyutlara ayrılmıştır.
- Örneğin, coğrafyada bölge, ülke, eyalet ve şehir gibi dört hiyerarşi düzeyi varsa, Galaxy şemasının dört boyutu olmalıdır.
- Dahası, tek yıldızlı şemayı daha fazla Yıldız şemasına bölerek bu tür bir şemayı oluşturmak mümkündür.
- Bu şemada, hiyerarşi düzeylerine dayalı olarak oluşturulması gereken boyutlar büyüktür.
- Bu şema, daha iyi anlaşılması için olgu tablolarının toplanması için yararlıdır.

## Star Cluster (Yıldız Küme) Şeması nedir?

Snowflake şeması, tamamen genişletilmiş hiyerarşileri içerir. Ancak bu, Şemaya karmaşıklık katabilir ve fazladan birleştirmeler gerektirir. Öte yandan, yıldız şeması tamamen daraltılmış hiyerarşiler içerir ve bu da fazlalıklara yol açabilir. Dolayısıyla, en iyi çözüm, **STAR CLUSTER SCHEMA** tasarımı olan bu iki şema arasında bir denge olabilir.



Yıldız Kümesi Şeması Örneği

Çakışan boyutlar hiyerarşilerde çatal olarak bulunabilir. Bir varlık, iki farklı boyutlu hiyerarşide bir üst öge olarak davrandığında çatal oluşur. Çatal varlıkları daha sonra bire çok ilişkileriyle sınıflandırma olarak tanımlandı.

## Özet:

- Çok boyutlu şema, özellikle veri ambarı sistemlerini modellemek için tasarlanmıştır
- Yıldız şeması, Veri Ambarı şemasının en basit türüdür. Yapısı bir yıldıza benzediği için yıldız şeması olarak bilinir.
- Bir Kar Tanesi Şeması, bir Yıldız Şemasının bir uzantısıdır ve ek boyutlar ekler. Şeması bir Kar Tanesine benzediği için buna kar tanesi denir.
- Bir yıldız şemasında, yalnızca tek birleştirme olgu tablosu ile herhangi bir boyut tablosu arasındaki ilişkiyi tanımlar.
- Yıldız şeması, boyut tablolarıyla çevrili bir olgu tablosu içerir.
- Kar tanesi şeması, boyut tablosu ile çevrelenen boyut tablosu ile çevrilidir.
- Bir kar tanesi şeması, verileri almak için birçok birleştirme gerektirir.
- Bir Galaxy Şeması, boyut tablolarını paylaşan iki olgu tablosu içerir. Fact Constellation Schema olarak da adlandırılır.
- Yıldız kümesi şeması, Başlangıç şeması ve Yavaş flake şemasının özniteliklerini içerir.

## Data Mart nedir?

Bir **DATA MART** bir organizasyonun tek bir işlevsel alanda odaklanmış ve bir veri ambarına depolanan veri alt kümesi içerir. Bir Data Mart, Veri Ambarının yoğunlaştırılmış bir sürümüdür ve bir organizasyondaki belirli bir departman, birim veya kullanıcı grubu tarafından kullanılmak üzere tasarlanmıştır. Örneğin, Pazarlama, Satış, İK veya finans. Genellikle bir organizasyondaki tek bir departman tarafından kontrol edilir.

Data Mart, genellikle bir Veri ambarına kıyasla yalnızca birkaç kaynaktan veri alır. Veri reyonları küçük boyutludur ve bir Datawarehouse'a kıyasla daha esnektir.

## Data Mart'a neden ihtiyacımız var?

- Data Mart, veri hacmindeki azalma nedeniyle kullanıcının yanıt süresini artırmaya yardımcı olur
- Sık istenen verilere kolay erişim sağlar.
- Data mart, kurumsal Datawarehouse ile karşılaştırıldığında uygulaması daha kolaydır. Aynı zamanda, Data Mart'ı uygulamanın maliyeti, tam bir veri ambarının uygulanmasına kıyasla kesinlikle daha düşüktür.
- Veri Ambarı ile karşılaştırıldığında bir veri parçası çeviktir. Modelde değişiklik olması durumunda, daha küçük olması nedeniyle datamart daha hızlı oluşturulabilir.

- Bir Datamart, tek bir Konu Datamart Uzmanı tarafından tanımlanır. Aksine veri ambarı, çeşitli alanlardan disiplinler arası KOBİ tarafından tanımlanır. Bu nedenle Data mart, Datawarehouse'a kıyasla değişime daha açıktır.
- Veriler bölümlenir ve çok ayrıntılı erişim denetimi ayrıcalıklarına izin verir.
- Veriler, farklı donanım / yazılım platformlarında bölümlere ayrılabilir ve depolanabilir.

## Data Mart Türü

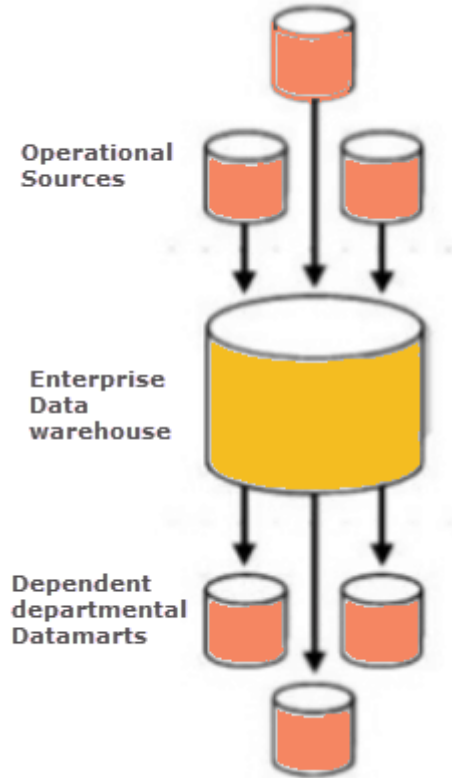
Üç ana veri pazarı türü vardır:

1. **Bağımlı** : Bağımlı veri reyonları, doğrudan operasyonel, harici veya her iki kaynaktan veri çekilerek oluşturulur.
2. **Bağımsız** : Merkezi bir veri ambarı kullanılmadan bağımsız veri pazarı oluşturulur.
3. **Karma** : Bu tür veri reyonları, veri ambarlarından veya işletim sistemlerinden veri alabilir.

## Bağımlı Data Mart

Bağımlı bir veri pazarı, kuruluşun verilerini tek bir Veri Ambarından tedarik etmeye olanak tanır. Merkezileştirmenin faydasını sunar. Bir veya daha fazla fiziksel veri reyonu geliştirmeniz gerekiyorsa, bunları bağımlı veri reyonları olarak yapılandırmanız gerekir.

Bağımlı veri reyonları iki farklı şekilde oluşturulabilir. Ya bir kullanıcının ihtiyaca bağlı olarak hem veri pazarına hem de veri ambarına erişebileceği veya erişimin yalnızca veri pazarı ile sınırlı olduğu yerlerde. İkinci yaklaşım, bazen veri hurdalığı olarak adlandırıldığı için optimal değildir. Veri hurdalığında, tüm veriler ortak bir kaynakla başlar, ancak hurdaya çıkarılır ve çoğunlukla hurdaya çıkarılır.



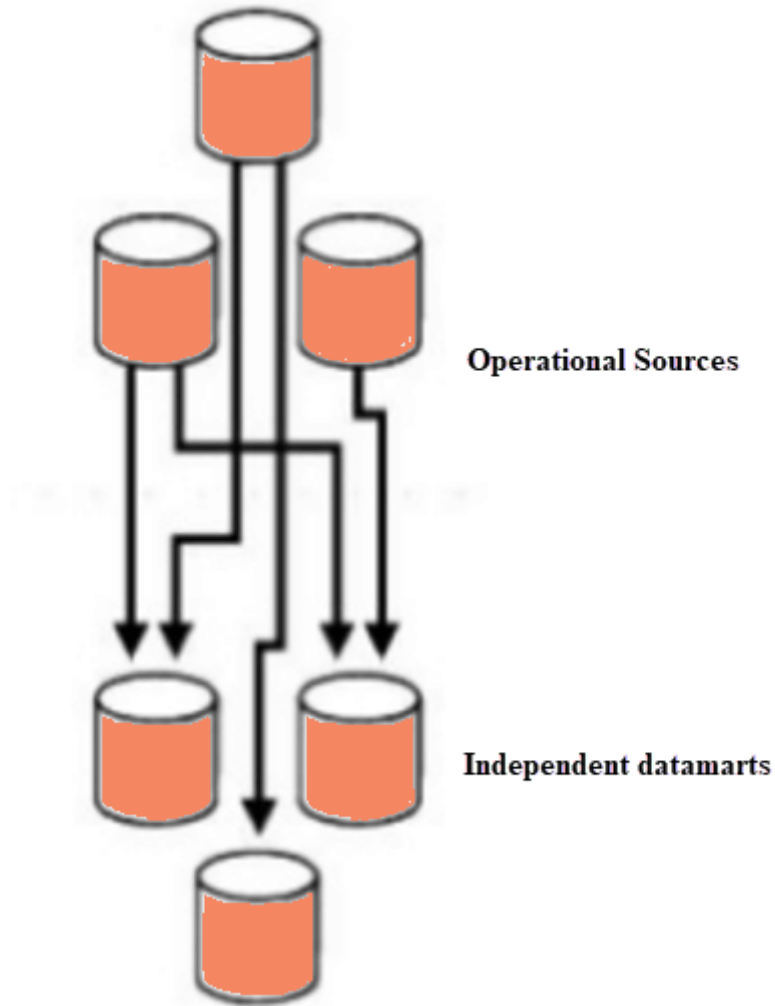


## Bağımsız Veri Mart

Merkezi Veri ambarı kullanılmadan bağımsız bir veri pazarı oluşturulur. Bu tür bir Data Mart, bir organizasyon içindeki daha küçük gruplar için ideal bir seçenektir.

Bağımsız bir veri reyonunun ne kurumsal veri ambarıyla ne de başka bir veri mağazasıyla ilişkisi yoktur. Bağımsız veri pazarında veriler ayrı ayrı girilir ve analizleri de bağımsız olarak gerçekleştirilir.

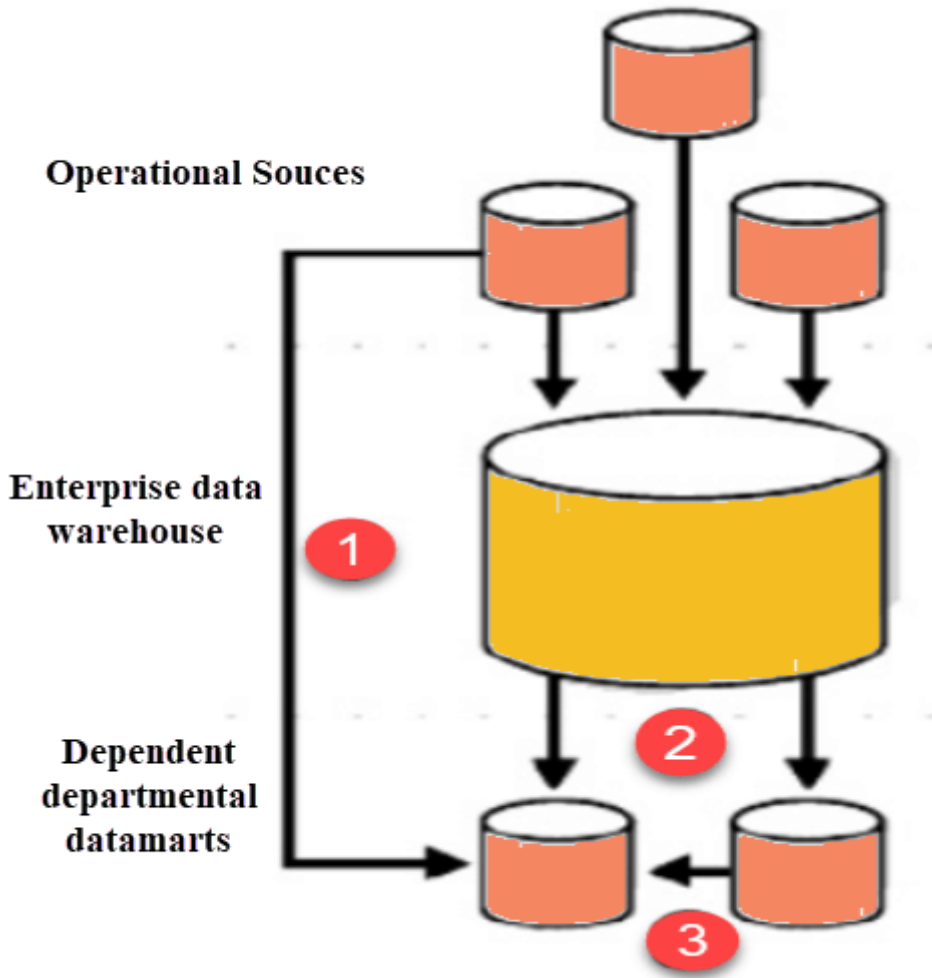
Bağımsız veri reyonlarının uygulanması, bir veri ambarı oluşturma motivasyonuna aykırıdır. Her şeyden önce, çok çeşitli bilgiler isteyen farklı ilgi alanlarına sahip birden çok kullanıcı tarafından analiz edilebilen tutarlı, merkezi bir kurumsal veri deposuna ihtiyacınız vardır.



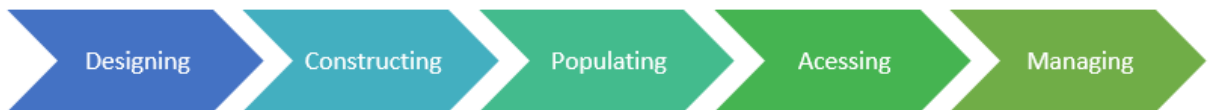
## Hibrit Data Mart:

Karma bir veri pazarı, Veri ambarı dışındaki kaynaklardan gelen girdileri birleştirir. Bu, kuruluşa yeni bir grup veya ürün eklendikten sonra geçici entegrasyon istediğinizde yardımcı olabilir.

Birden çok veritabanı ortamı ve herhangi bir kuruluş için hızlı uygulama dönüşümü için en uygun yöntemdir. Ayrıca en az veri temizleme çabası gerektirir. Hybrid Data mart aynı zamanda büyük depolama yapılarını da destekler ve daha küçük veri merkezli uygulamalar için en uygun olanıdır.



## Datamart Uygulamasında Adımlar



Bir Data Mart uygulamak, ödüllendirici ama karmaşık bir prosedürdür. Bir Data Mart uygulamak için ayrıntılı adımlar şunlardır:

## Designing (Tasarım)

Tasarım, Data Mart uygulamasının ilk aşamasıdır. Gereksinimler hakkında bilgi toplamaya kadar bir veri pazarı talebini başlatmak arasındaki tüm görevleri kapsar. Son olarak, veri pazarının mantıksal ve fiziksel tasarımını oluşturuyoruz.

**Tasarım aşaması aşağıdaki görevleri içerir:**

- İş ve teknik gereksinimleri toplamak ve veri kaynaklarını belirlemek.
- Uygun veri alt kümesini seçme.
- Veri pazarının mantıksal ve fiziksel yapısının tasarlanması.

Veriler aşağıdaki kriterlere göre bölümlere ayrılabilir:

- Tarih
- İş veya Fonksiyonel Birim
- Coğrafya
- Yukarıdakilerin herhangi bir kombinasyonu

Veriler, uygulama veya DBMS düzeyinde bölümlere ayrılabilir. İş ortamındaki değişikliklerle her yıl farklı veri modellerine izin verdiği için Uygulama düzeyinde bölümlere ayrılması önerilir.

**Hangi Ürün ve Teknolojilere İhtiyacınız Var?**

Basit bir kalem ve kağıt yeterli olurdu. UML veya ER diyagramları oluşturmanıza yardımcı olan araçlar, meta verileri mantıksal ve fiziksel tasarımlarınıza da ekler.

## Constructing (İnşaat)

Bu, uygulamanın ikinci aşamasıdır. Fiziksel veri tabanını ve mantıksal yapıları oluşturmayı içerir.

**Bu adım aşağıdaki görevleri içerir:**

- Önceki aşamada tasarlanan fiziksel veritabanının uygulanması. Örneğin tablo, indeksler, görünüm gibi veritabanı şema nesneleri oluşturulur.

**Hangi Ürün ve Teknolojilere İhtiyacınız Var?**

Bir veri pazarı oluşturmak için ilişkisel bir veritabanı yönetim sistemine ihtiyacınız var. RDBMS, bir Data Mart'ın başarısı için gerekli olan çeşitli özelliklere sahiptir.

- **Depolama yönetimi:** Bir RDBMS, verileri oluşturmak, eklemek ve silmek için verileri depolar ve yönetir.
- **Hızlı veri erişimi:** Bir SQL sorgusu ile, belirli koşullara / filtrelerle göre verilere kolayca erişebilirsiniz.

- **Veri koruma:** RDBMS sistemi ayrıca güç kesintileri gibi sistem arızalarından kurtulmanın bir yolunu sunar. Ayrıca, diskin başarısız olması durumunda bu yedeklemelerden verilerin geri yüklenmesine izin verir.
- **Çok kullanıcı desteği:** Veri yönetim sistemi, eşzamanlı erişim, birden çok kullanıcının başka bir kullanıcı tarafından yapılan değişikliklere müdahale etmeden veya üzerine yazmadan verilere erişme ve bunları değiştirme yeteneği sunar.
- **Güvenlik:** RDMS sistemi ayrıca kullanıcıların nesnelere ve belirli işlem türlerine erişimini düzenlemenin bir yolunu sağlar.

## Populating (Doldurma):

Üçüncü aşamada, veri martındaki veriler doldurulur.

Doldurma adımı aşağıdaki görevleri içerir:

- Veri Eşlemeyi hedeflemek için kaynak veriler
- Kaynak verilerin çıkarılması
- Veriler üzerinde temizleme ve dönüştürme işlemleri
- Data mart'a veri yükleme
- Meta verileri oluşturma ve depolama

## Hangi Ürün ve Teknolojilere İhtiyacınız Var?

Bu doldurma görevlerini bir ETL (Dönüştürme Yükünü Çıkarma) Aracı kullanarak gerçekleştirirsiniz. Bu araç, veri kaynaklarına bakmanıza, kaynaktan hedefe eşleme yapmanıza, verileri çıkarmanıza, dönüştürmenize, temizlemenize ve veri mağazasına geri yüklemenize olanak tanır.

Bu süreçte araç, verilerin nereden geldiği, ne kadar yeni olduğu, verilerde ne tür değişiklikler yapıldığı ve hangi düzeyde özetleme yapıldığı gibi şeylerle ilgili bazı meta veriler de oluşturur.

## Accessing (Erişim):

Erişim, verilerin kullanılmasını içeren dördüncü bir adımdır: verileri sorgulama, raporlar, grafikler oluşturma ve bunları yayınlama. Son kullanıcı veritabanına sorgu gönderir ve sorguların sonuçlarını görüntüler.

**Erişim adımının aşağıdaki görevleri gerçekleştirmesi gerekir:**

- Veritabanı yapılarını ve nesne adlarını iş terimlerine çeviren bir meta katman oluşturun. Bu, teknik bilgisi olmayan kullanıcıların Data mart'a kolayca erişmesine yardımcı olur.
- Veritabanı yapılarını kurun ve koruyun.
- Gerekirse API ve arayüzleri ayarlayın

## Hangi Ürün ve Teknolojilere İhtiyacınız Var?

Data Marta komut satırını veya GUI'yi kullanarak erişebilirsiniz. GUI, kolayca grafik oluşturabildiği ve komut satırına göre kullanıcı dostu olduğu için tercih edilmektedir.

## Managing (Yönetme):

Bu, Data Mart Uygulama sürecinin son adımıdır. Bu adım, aşağıdakiler gibi yönetim görevlerini kapsar:

- Devam eden kullanıcı erişim yönetimi.
- Gelişmiş performansa ulaşmak için sistem optimizasyonları ve ince ayar.
- Data Marta yeni veriler eklemek ve yönetmek.
- Kurtarma senaryolarını planlamak ve sistemin arızalanması durumunda sistem kullanılabilirliğini sağlamak.

## Hangi Ürün ve Teknolojilere İhtiyacınız Var?

Data mart yönetimi için GUI'yi veya komut satırını kullanabilirsiniz.

## Veri Martlarını Uygulama için en iyi uygulamalar

Data Mart Uygulama sürecindeyken izlemeniz gereken en iyi uygulamalar şunlardır:

- Bir Data Mart'ın kaynağı departman bazında yapılandırılmalıdır
- Bir Data Mart'ın uygulama döngüsü kısa zaman aralıklarında, yani aylar veya yıllar yerine haftalar halinde ölçülmelidir.
- Veri pazarı uygulaması karmaşık olabileceğinden, tüm paydaşları planlama ve tasarım aşamasına dahil etmek önemlidir.
- Data Mart Donanım / Yazılım, Ağ Oluşturma ve Uygulama maliyetleri, planınızda doğru bir şekilde bütçelendirilmelidir
- Data mart aynı donanım üzerinde oluşturulsa bile, kullanıcı sorgularını işlemek için bazı farklı yazılımlara ihtiyaç duyabilirler. Hızlı kullanıcı yanıtı için ek işlem gücü ve disk depolama gereksinimleri değerlendirilmelidir
- Bir veri pazarı, veri ambarından farklı bir konumda olabilir. Bu nedenle, veri pazarına veri aktarmak için gereken Veri hacimlerini işlemek için yeterli ağ kapasitesine sahip olduklarından emin olmak önemlidir.
- Uygulama maliyeti, Datamart yükleme işlemi için harcanan zamanı bütçelemelidir. Yükleme süresi, dönüşümlerin karmaşıklığının artmasıyla artar.

## Veri Martının Avantajları ve Dezavantajları

### Avantajlar

- Veri reyonları, kuruluş çapında verilerin bir alt kümesini içerir. Bu Veriler, bir kuruluştaki belirli bir grup insan için değerlidir.

- Oluşturulması yüksek maliyetler alabilen bir veri ambarına uygun maliyetli alternatiflerdir.
- Data Mart, Verilere daha hızlı erişim sağlar.
- Data Mart, kullanıcılarının ihtiyaçları için özel olarak tasarlandığından kullanımı kolaydır. Böylece bir veri pazarı iş süreçlerini hızlandırabilir.
- Data Marts, Veri Ambarı sistemlerine kıyasla daha az uygulama süresine ihtiyaç duyar. Yalnızca verilerin tek alt kümesine konsantre olmanız gerektiğinden Data Mart'ı uygulamak daha hızlıdır.
- Analistin veri eğilimlerini belirlemesini sağlayan geçmiş verileri içerir.

## Dezavantajları

- Çoğu zaman işletmeler, çok fazla fayda sağlamadan çok fazla farklı ve ilgisiz veri pazarı oluşturur. Korumak için büyük bir engel olabilir.
- Data Mart, veri kümeleri sınırlı olduğu için şirket çapında veri analizi sağlayamaz.

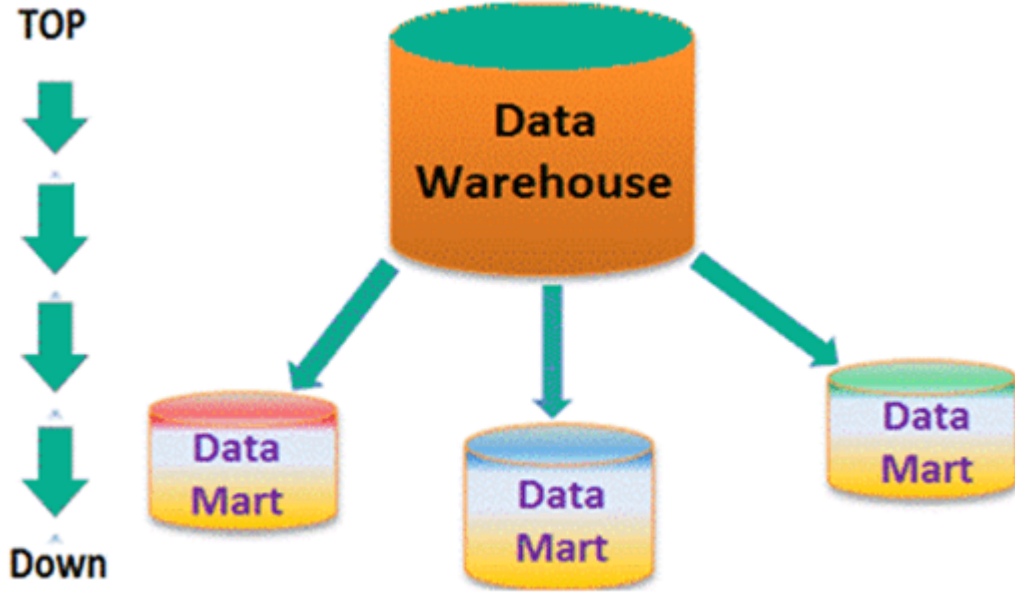
## Özet:

- Bir Veri Pazarı, bir kuruluşun tek bir işlevsel alanına odaklanan bir Veri Ambarı alt kümesi olarak tanımlanır.
- Data Mart, veri hacmindeki azalma nedeniyle kullanıcının yanıt süresini artırmaya yardımcı olur.
- Üç tür veri pazarı 1) Bağımlı 2) Bağımsız 3) Karma
- Data Mart'ın önemli uygulama adımları şunlardır: 1) Tasarlama 2) Oluşturma 3) Yerleştirme 4) Erişim ve 5) Yönetme
- Bir Data Mart'ın uygulama döngüsü kısa zaman aralıklarında, yani aylar veya yıllar yerine haftalar halinde ölçülmelidir.
- Data mart, oluşturmak için yüksek maliyetler alabilen bir veri ambarına uygun maliyetli alternatiflerdir.
- Data Mart, veri seti sınırlı olduğu için şirket çapında veri analizi sağlayamaz.

## Data Mart vs Veri Ambarı

### ANAHTAR FARK

- Veri Ambarı, farklı kaynaklardan toplanan büyük bir veri deposudur, oysa Data Mart, bir veri ambarının yalnızca alt tipidir.
- Veri Ambarı bir organizasyondaki tüm departmanlara odaklanırken, Data Mart belirli bir gruba odaklanır.
- Veri Ambarı tasarım süreci karmaşıktır, Data Mart sürecinin tasarımı ise kolaydır.
- Veri Ambarı, veri işleme için uzun zaman alırken, Data Mart veri işleme için kısa bir zaman alır.
- Veri Ambarı boyut aralığı 100 GB - 1 TB +, Data Mart boyutu 100 GB'den azdır.
- Veri Ambarı uygulama süreci 1 aydan 1 yıla kadar sürerken, Data Mart'ın uygulama sürecini tamamlaması birkaç ay sürer.



### Veri Ambarı ve Veri Martı arasındaki farklar

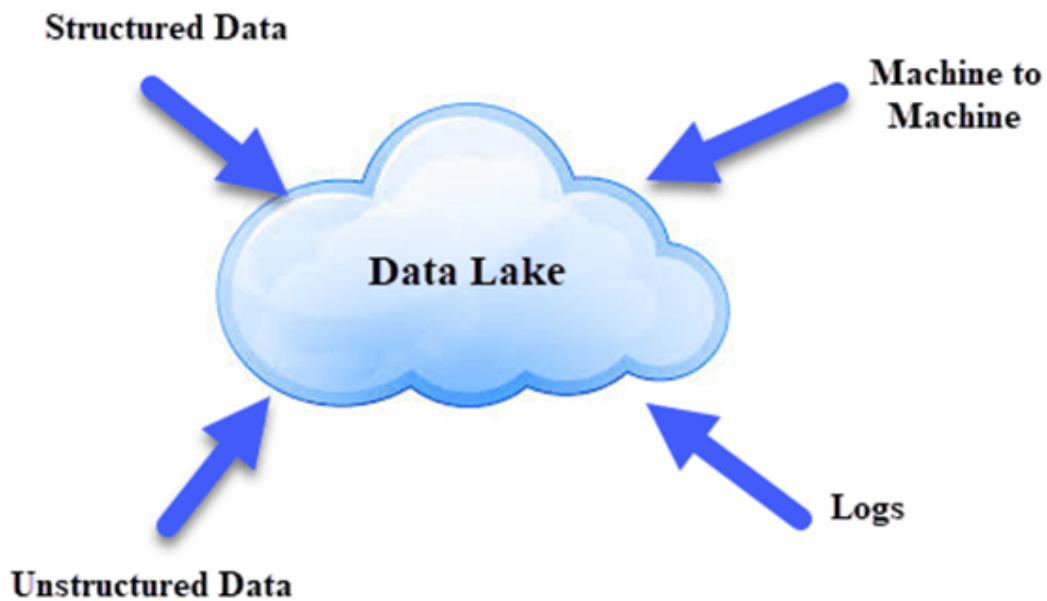
Parametre	Veri deposu	Data Mart
Tanım	Veri Ambarı, bir şirket içindeki farklı organizasyonlardan veya departmanlardan toplanan büyük bir veri havuzudur.	Bir veri pazarı, Veri Ambarının tek bir alt türüdür. Belirli bir kullanıcı grubunun ihtiyacını karşılayacak şekilde tasarlanmıştır.
Kullanım	Stratejik bir karar almaya yardımcı olur.	İş için taktiksel kararlar almaya yardımcı olur.
Amaç	Veri Ambarı'nın temel amacı, bir noktada entegre bir ortam ve işletmenin tutarlı bir resmini sağlamaktır.	Çoğunlukla departman düzeyinde bir iş bölümünde kullanılan bir veri pazarı.
Tasarım	Veri Ambarı tasarım süreci oldukça zordur.	Data Mart'ın tasarım süreci kolaydır.
	Boyutsal bir modelde kullanabilir veya kullanmayabilir. Ancak boyutsal modelleri besleyebilir.	Bir başlangıç şeması kullanarak boyutlu bir modele odaklanmıştır.
Veri işleme	Veri ambarı, kurumun geniş bir alanını kapsadığından, işlenmesi uzun zaman alır.	Veri reyonlarının kullanımı, tasarımı ve uygulaması kolaydır, çünkü yalnızca küçük miktarda veriyi işleyebilir.
Dürbün	Veri ambarlama, herhangi bir departmandan bilgi getirebileceği için daha faydalıdır.	Data mart, bir şirketin belirli bir departmanının verilerini içerir. Satış, finans, pazarlama vb. için ayrı veri reyonları olabilir. Sınırlı kullanıma sahiptir.
Kaynak	Veri Ambarında Veriler birçok kaynaktan gelir.	Data Mart'ta veriler çok az kaynaktan gelir.
Boyut	Veri Ambarı boyutu 100 GB ile 1 TB + arasında değişebilir.	Data Mart'ın Boyutu 100 GB'den az.
Uygulama süresi	Veri Ambarı uygulama süreci aylardan yıllara uzatılabilir.	Data Mart'ın uygulama süreci birkaç ay ile sınırlıdır.

Odaklanma	Veri ambarlama genel olarak tüm departmanlara odaklanmıştır. Hatta tüm şirketi temsil etmesi bile mümkündür.	Data Mart konu odaklıdır ve departman düzeyinde kullanılır.
Veri tipi	Data Warehouse'da depolanan veriler, data mart ile karşılaştırıldığında her zaman ayrıntılıdır.	Veri Martları belirli kullanıcı grupları için oluşturulmuştur. Bu nedenle, veriler kısa ve sınırlıdır.
Konu alanı	Veri Ambarı'nın temel amacı, bir noktada entegre bir ortam ve işletmenin tutarlı bir resmini sağlamaktır.	Çoğunlukla yalnızca bir konu alanını tutun - örneğin Satış rakamı.
Veri depolama	Yalnızca pazarlama verilerini değil, kurumsal çapta karar verilerini depolamak için tasarlanmıştır.	Erişim katmanının performansını optimize etmek için kullanılan boyutsal modelleme ve yıldız şeması tasarımı.
Veri tipi	Zaman farkı ve kalıcı tasarım kesinlikle uygulanır.	Çoğunlukla konu alanının sorgu ve raporlama ihtiyaçlarını karşılamak için konsolidasyon veri yapılarını içerir.
Veri değeri	Son kullanıcılar açısından Salt Okunur.	Doğrudan Veri Ambarı'ndan beslenen tahıldan bağımsız olarak işlem verileri.

## Data Lake nedir?

Veri Gölü, büyük miktarda yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış veriyi depolayabilen bir depolama havuzudur. Hesap boyutu veya dosyasında sabit sınırlar olmaksızın her tür veriyi yerel biçiminde depolayabileceğiniz bir yerdir. Analitik performansı ve yerel entegrasyonu artırmak için yüksek veri miktarı sunar.

Data Lake, gerçek göl ve nehlere çok benzeyen büyük bir konteynır gibidir. Tıpkı bir gölde olduğu gibi, gelen birden fazla kolunuz var, bir veri gölünde yapılandırılmış veriler, yapılandırılmamış veriler, makineden makineye, gerçek zamanlı olarak akan günlükler vardır.





Data Lake, verileri demokratikleştirir ve bir kuruluşun tüm verilerini daha sonra işlenmek üzere depolamanın uygun maliyetli bir yoludur. Araştırma Analisti, verilerin kendisinde değil, verilerdeki anlam kalıplarını bulmaya odaklanabilir.

Verilerin Dosyalar ve Klasörde depolandığı hiyerarşik bir Datawarehouse aksine, Data lake düz bir mimariye sahiptir.

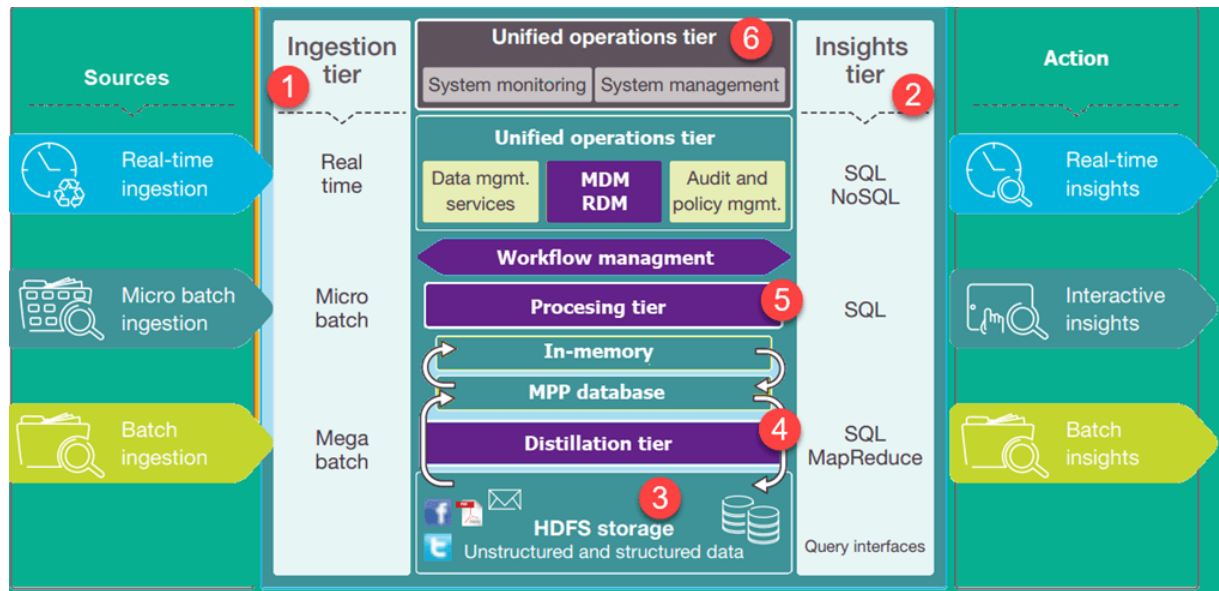
## Neden Data Lake?

Bir veri gölü oluşturma temel amacı, veri bilimcilerine verilerin rafine edilmemiş bir görünümünü sunmaktır.

Data Lake'i kullanma nedenleri şunlardır:

- Hadoop gibi depolama motorlarının başlamasıyla, farklı bilgileri depolamak kolay hale geldi. Veri Gölü ile kurumsal çapta bir şemaya veri modellemeye gerek yoktur.
- Veri hacmi, veri kalitesi ve meta verilerdeki artışla birlikte analizlerin kalitesi de artar.
- Data Lake iş çevikliği sunar
- Makine Öğrenimi ve Yapay Zeka, karlı tahminlerde bulunmak için kullanılabilir.
- Uygulayıcı kuruluşa rekabet avantajı sağlar.
- Veri silosu yapısı yoktur. Data Lake, 360 derece müşteri görüşü sağlar ve analizi daha sağlam hale getirir.

## Data Lake Mimarisi



Şekil, bir Business Data Lake mimarisini göstermektedir. Alt düzeyler, çoğunlukla hareketsiz durumda olan verileri temsil ederken, üst düzeyler gerçek zamanlı işlem verilerini gösterir. Bu veriler, hiç gecikme olmadan veya çok az gecikme ile sistem boyunca akar. Data Lake Mimarisindeki önemli katmanlar aşağıdadır:

1. **Besleme Katmanı** : Sol taraftaki katmanlar, veri kaynaklarını gösterir. Veriler veri gölüne gruplar halinde veya gerçek zamanlı olarak yüklenebilir
2. **İçgörüler Katmanı**: Sağdaki katmanlar, sistemden alınan içgörülerin kullanıldığı araştırma tarafını temsil eder. Veri analizi için SQL, NoSQL sorguları ve hatta excel kullanılabilir.
3. **HDFS**, hem yapılandırılmış hem de yapılandırılmamış veriler için uygun maliyetli bir çözümdür. Sistemde durmakta olan tüm veriler için bir iniş bölgesidir.
4. **Damıtma katmanı**, verileri depolama lastiğinden alır ve daha kolay analiz için yapılandırılmış verilere dönüştürür.
5. **İşleme katmanı**, daha kolay analiz için yapılandırılmış veriler oluşturmak üzere analitik algoritmaları ve kullanıcı sorgularını değişen gerçek zamanlı, etkileşimli, toplu olarak çalıştırır.
6. **Birleştirilmiş işlemler katmanı**, sistem yönetimini ve izlemeyi yönetir. Denetim ve yeterlilik yönetimi, veri yönetimi, iş akışı yönetimi içerir.

## Temel Veri Gölü Kavramları

Aşağıda Data Lake Mimarisini tamamen anlamak için anlaşılması gereken Anahtar Data Lake kavramları verilmiştir.



## Veri Alımı

Veri Besleme, bağlayıcıların farklı veri kaynaklarından veri almasına ve Veri gölüne yüklemesine olanak tanır.

Veri Besleme şunları destekler:

- Her türlü Yapılandırılmış, Yarı Yapılandırılmış ve Yapılandırılmamış veri.
- Toplu, Gerçek Zamanlı, Tek seferlik yükleme gibi birden çok besleme.
- Veritabanları, Web Sunucuları, E-postalar, IoT ve FTP gibi birçok veri kaynağı türü.

## Veri depolama

Veri depolama ölçeklenebilir olmalı, uygun maliyetli depolama sunmalı ve veri keşiflerine hızlı erişim sağlamalıdır. Çeşitli veri formatlarını desteklemelidir.

## Veri yönetimi

Veri yönetimi, bir kuruluştaki kullanılan verilerin kullanılabilirliğini, kullanılabilirliğini, güvenliğini ve bütünlüğünü yönetme sürecidir.

## Güvenlik

Veri gölünün her katmanında güvenliğin uygulanması gerekir. Depolama, Açığa Çıkarma ve Tüketim ile başlar. Temel ihtiyaç, yetkisiz kullanıcılar için erişimi durdurma ve Gezinmesi kolay GUI ve Dashboard'larla verilere erişmek için farklı araçları desteklemelidir.

Kimlik Doğrulama, Muhasebe, Yetkilendirme ve Veri Koruma, veri gölü güvenliğinin bazı önemli özellikleridir.

## Veri kalitesi:

Veri kalitesi, Data Lake mimarisinin önemli bir bileşenidir. Veriler, iş değerini kesinleştirmek için kullanılır. Düşük kaliteli verilerden içgörüler çıkarmak, düşük kaliteli içgörülere yol açacaktır.

## Veri Keşfi

Veri Keşfi, veri veya analiz hazırlamaya başlamadan önceki bir başka önemli aşamadır. Bu aşamada Veri gölüne alınan verilerin düzenlenmesi ve yorumlanması yoluyla verilerin anlaşılmasını ifade etmek için etiketleme tekniği kullanılır.

## Veri Denetimi

İki ana Veri denetim görevi, anahtar veri kümesindeki değişiklikleri izlemektir.

1. Önemli veri kümesi öğelerindeki değişiklikleri izleme
2. Bu öğelere nasıl / ne zaman / ve kimin değiştiğini yakalar.

Veri denetimi, risk ve uyumluluğun değerlendirilmesine yardımcı olur.

## Veri Köken

Bu bileşen, verilerin kökenleriyle ilgilenir. Esas olarak zaman içinde nereye hareket ettiği ve ona ne olduğu ile ilgilenir. Başlangıçtan varış noktasına bir veri analizi sürecindeki hata düzeltmelerini kolaylaştırır.

## Veri Keşfi

Veri analizinin başlangıç aşamasıdır. Veri Keşfetmeye başlamadan önce doğru veri kümesinin belirlenmesine yardımcı olur.

Verili tüm bileşenlerin, Data lake yapısında önemli bir rol oynamak için birlikte çalışması gerekir. Çevreyi kolayca geliştirip keşfedebilirsiniz.

## Data Lake'in olgunluk aşamaları

Data Lake Olgunluk aşamalarının tanımı ders kitabından diğerine farklılık gösterir. İşin özü aynı kalsa da. Olgunluğu takiben, aşama tanımı sıradan bir bakış açısidir.



### 1. Aşama: Verileri geniş ölçekte işleme ve kullanma

Veri Olgunluğunun bu ilk aşaması, verileri dönüştürme ve analiz etme yeteneğini geliştirmeyi içerir. Burada, işletme sahiplerinin daha fazla veri elde etmek ve analitik uygulamalar oluşturmak için becerilerine göre araçları bulması gerekir.

### Aşama 2: Analitik kasın oluşturulması

Bu, verileri dönüştürme ve analiz etme yeteneğini geliştirmeyi içeren ikinci bir aşamadır. Bu aşamada şirketler, beceri setlerine en uygun aracı kullanır. Daha fazla veri almaya ve uygulama oluşturmaya başlarlar. Burada kurumsal veri ambarının ve veri gölünün yetenekleri birlikte kullanılır.

## Aşama 3: EDW ve Data Lake birlikte çalışır

Bu adım, verileri ve analitiği olabildiğince çok kişinin eline almayı içerir. Bu aşamada, veri gölü ve kurumsal veri ambarı bir arada çalışmaya başlar. Her ikisi de analitikte rollerini oynuyor

## Aşama 4: Göldeki işletme yeteneği

Veri gölünün bu olgunluk aşamasında, Veri Gölüne kurumsal yetenekler eklenir. Bilgi yönetişiminin, bilgi yaşam döngüsü yönetimi yeteneklerinin ve Meta veri yönetiminin benimsenmesi. Ancak, çok az kuruluş bu olgunluk düzeyine ulaşabilir, ancak bu çetele gelecekte artacaktır.

### Data Lake Uygulaması için en iyi uygulamalar:

- Mimari bileşenler, etkileşimleri ve tanımlanmış ürünler yerel veri türlerini desteklemelidir
- Data Lake'in tasarımı, gerekli olanın yerine mevcut olanla yönlendirilmelidir. Şema ve veri gereksinimi sorgulanana kadar tanımlanmaz
- Tasarım, hizmet API'si ile entegre tek kullanımlık bileşenler tarafından yönlendirilmelidir.
- Veri keşfi, alma, depolama, yönetim, kalite, dönüştürme ve görselleştirme bağımsız olarak yönetilmelidir.
- Veri Gölü mimarisi belirli bir sektöre göre uyarlanmalıdır. Bu alan için gerekli yeteneklerin tasarımın doğal bir parçası olmasını sağlamalıdır.
- Yeni keşfedilen veri kaynaklarının daha hızlı devreye alınması önemlidir
- Data Lake, özelleştirilmiş yönetimin maksimum değeri elde etmesine yardımcı olur
- Veri Gölü, mevcut kurumsal veri yönetimi tekniklerini ve yöntemlerini desteklemelidir

### Bir veri gölü oluşturma zorlukları:

- Data Lake'te Veri hacmi daha yüksektir, bu nedenle işlemin programatik yönetime daha fazla bağımlı olması gerekir
- Seyrek, eksik, değişken verilerle uğraşmak zordur
- Daha geniş veri kümesi ve kaynak kapsamı, daha büyük veri yönetimi ve desteği gerektirir

### Data Lake kullanmanın Yararları ve Riskleri:

Veri Gölü kullanmanın bazı önemli avantajları şunlardır:

- Ürün iyonlaştırıcı ve gelişmiş analitik ile tamamen yardımcı olur
- Uygun maliyetli ölçeklenebilirlik ve esneklik sunar
- Sınırsız veri türlerinden değer sunar
- Uzun vadeli sahip olma maliyetini düşürür
- Dosyaların ekonomik olarak depolanmasına izin verir

- Değişikliklere hızla uyaranabilir
- Veri gölünün temel avantajı, farklı içerik kaynaklarının **merkezeleştirilmesidir**.
- Çeşitli departmanlardan kullanıcılar dünyanın dört bir yanına dağılmış olabilir, verilere **esnek erişim** sağlayabilir

#### **Data Lake Kullanma Riski:**

- Bir süre sonra Data Lake alaka düzeyini ve ivmesini kaybedebilir
- Data Lake'i tasarlarlarken daha büyük miktarda risk söz konusudur
- Yapılandırılmamış Veriler, Yönetilmeyen Chao, Kullanılmayan Veriler, Farklı ve Karmaşık Araçlar, Kurumsal Çapta İş birliği, Birleştirilmiş, Tutarlı ve Ortak olabilir
- Ayrıca depolama ve hesaplama maliyetlerini artırır
- Verilerle çalışan diğer kişilerden içgörür almanın bir yolu yoktur çünkü önceki analistlerin bulgularının soyunun bir açıklaması yoktur.
- Veri göllerinin en büyük riski güvenlik ve erişim kontrolüdür. Verilerin bir kısmının mahremiyet ve yasal düzenleme ihtiyacı olabileceğinden, bazen veriler herhangi bir gözetim olmaksızın bir göle yerleştirilebilir.

#### **Özet:**

- Veri Gölü, büyük miktarda yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış veriyi depolayabilen bir depolama havuzudur.
- Bir veri gölü oluşturmanın temel amacı, veri bilimcilerine verilerin rafine edilmemiş bir görünümünü sunmaktır.
- Birleştirilmiş işlem katmanı, İşleme katmanı, Damıtma katmanı ve HDFS, Data Lake Mimarisinin önemli katmanlarıdır
- Veri Besleme, Veri depolama, Veri kalitesi, Veri Denetleme, Veri keşfi, Veri keşfi, Data Lake Mimarisinin bazı önemli bileşenleridir
- Data Lake'in tasarımı, gerekli olanın yerine mevcut olanla yönlendirilmelidir.
- Data Lake, uzun vadeli sahip olma maliyetini düşürür ve dosyaların ekonomik olarak depolanmasına izin verir
- Veri göllerinin en büyük riski güvenlik ve erişim kontrolüdür. Bazı verilerin mahremiyet ve yasal düzenleme ihtiyacı olabileceğinden, bazen veriler herhangi bir gözetim olmaksızın bir göle yerleştirilebilir.

# Data Lake VS Data Warehouse

Belirtilen yönlerde veriyle ilişkili iki terim arasındaki temel farklar şunlardır:

Parametreler	Veri Gölü	Veri deposu
Depolama	Veri gölünde, kaynak ve yapısından bağımsız olarak tüm veriler tutulur. Veriler ham biçiminde tutulur. Yalnızca kullanıma hazır olduğunda dönüştürülür.	Bir veri ambarı, işlem sistemlerinden çıkarılan verilerden veya nitelikleriyle birlikte nicel metriklerden oluşan verilerden oluşacaktır. Veriler temizlenir ve dönüştürülür
Tarih	Veri göllerinde kullanılan büyük veri teknolojileri nispeten yenidir.	Veri ambarı konsepti, büyük veriden farklı olarak onlarca yıldır kullanılıyordu.
Veri Yakalama	Yarı yapılandırılmış ve yapılandırılmamış her türlü veri ve yapıyı orijinal haliyle kaynak sistemlerden yakalar.	Yapılandırılmış bilgileri yakalar ve bunları veri ambarı amaçları için tanımlandığı şekilde şemalar halinde düzenler
Veri Zaman Çizelgesi	Veri gölleri tüm verileri tutabilir. Bu, yalnızca kullanımda olan verileri değil, gelecekte kullanılabileceği verileri de içerir. Ayrıca, geçmişe gitmek ve analiz yapmak için veriler her zaman saklanır.	Veri ambarı geliştirme sürecinde, çeşitli veri kaynaklarını analiz etmek için önemli bir zaman harcanır.
Kullanıcılar	Veri gölü, derin analize düşkün kullanıcılar için idealdir. Bu tür kullanıcılar, öngörücü modelleme ve istatistiksel analiz gibi yeteneklere sahip gelişmiş analitik araçlara ihtiyaç duyan veri bilimcilerini içerir.	Veri ambarı, iyi yapılandırılmış olması, kullanımı ve anlaşılması kolay olması nedeniyle operasyonel kullanıcılar için idealdir.
Depolama Maliyetleri	Büyük veri teknolojilerinde veri depolamak nispeten ucuzdur ve daha sonra verileri bir veri ambarında depolamaktır.	Veri ambarında veri depolamak maliyetlidir ve zaman alıcıdır.
Görev	Veri gölleri tüm veri ve veri türlerini içerebilir; kullanıcıların, dönüştürülmüş, temizlenmiş ve yapılandırılmış süreçlerden önce verilere erişmesini sağlar.	Veri ambarları, önceden tanımlanmış veri türleri için önceden tanımlanmış sorulara ilişkin içgörüler sağlayabilir.
İşlem süresi	Veri gölleri, kullanıcıların verilere dönüştürülmeden, temizlenmeden ve yapılandırılmadan önce erişmelerini sağlar. Böylece, geleneksel veri ambarına kıyasla kullanıcıların sonuca daha hızlı ulaşmasını sağlar.	Veri ambarları, önceden tanımlanmış veri türleri için önceden tanımlanmış sorulara ilişkin içgörüler sunar. Bu nedenle, veri ambarında yapılacak herhangi bir değişiklik daha fazla zaman gerektiriyordu.
Şemanın Konumu	Tipik olarak, şema veriler depolandıktan sonra tanımlanır. Bu, yüksek çeviklik ve veri yakalama kolaylığı sunar ancak sürecin sonunda çalışma gerektirir	Tipik olarak şema, veriler depolanmadan önce tanımlanır. Sürecin başlangıcında çalışma gerektirir, ancak performans, güvenlik ve entegrasyon sunar.

<b>Veri işleme</b>	Data Lakes ELT (Yükü Çıkarma) işleminin kullanımı.	Veri ambarı, geleneksel bir ETL (Extract Transform Load) işlemi kullanır.
<b>Şikayet</b>	Veriler ham biçiminde tutulur. Yalnızca kullanıma hazır olduğunda dönüştürülür.	Veri ambarlarına yönelik başlıca şikayet, yetersizlik veya bunlarda değişiklik yapmaya çalışırken karşılaşılan sorundur.
<b>Temel Avantajlar</b>	Tamamen yeni sorular ortaya çıkarmak için farklı veri türlerini entegre ediyorlar, çünkü bu kullanıcılar veri ambarlarını kullanma olasılıkları düşük çünkü veri ambarlarının kapasitelerinin ötesine geçmeleri gerekebiliyor.	Bir organizasyondaki çoğu kullanıcı operasyoneldir. Bu tür kullanıcılar yalnızca raporları ve temel performans ölçümlerini önemsemektedir.

Umarım dökümanım faydalı olmuştur 😊

- Alperen KEZAY

Source:

<https://www.guru99.com/data-warehousing.html>

[https://en.wikipedia.org/wiki/Data\\_warehouse](https://en.wikipedia.org/wiki/Data_warehouse)

<https://www.javatpoint.com/types-of-data-warehouses>

<https://panoply.io/data-warehouse-guide/the-difference-between-a-database-and-a-data-warehouse/>

<https://aws.amazon.com/tr/data-warehouse/#:~:text=How%20does%20a%20data%20warehouse>