

# VERİTABANI KAVRAMINA GİRİŞ

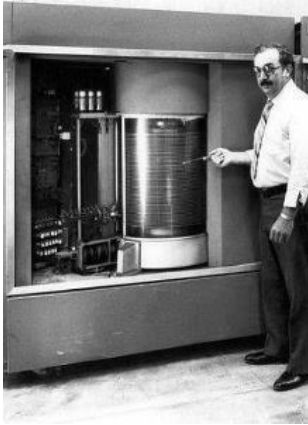
1

## VERİTABANI NEDİR?

Bilgisayar teknolojilerinin icadı ve gelişimiyle birlikte dijital veri deyimi sözcüklerimize eklendi. Dijital veri kavramı beraberinde depolama terimini de zorunlu hale getirdi. Bir şey var ise sanal ya da gerçek bir yer-alan kaplıyor demektir. Nasıl ki bir Hard Disk (*HDD*) masanın üzerinde dururken o masada yer kaplıyor ise, içerisindeki verinin de bir şekilde yer kaplaması kaçınılmazdır. Buna da sanal, yani dijital veri depolama alanı denir.

Bilgisayar sistemleri geliştikçe dijital veri kavramı da gelişti.

5mb 50yıl önce, 160.000\$



2gb,2012, 10TL



Bilgisayarın ilk yıllarında oda büyüklüğünde olan bilgisayarlar, artık cebimize sığar hale geldi. Tabi ki depolama alanları da aynı şekilde. İlk zamanlarda 5 MB'lık bir depolama alanı devasa boyutlarda iken, artık kibrit kutusunun içine bir kaç tane sığabilecek kadar küçüldü. Bu veri depolama alanları büyüdükçe verinin yönetilmesi diye bir kavramın da oluşması kaçınılmaz hale geldi.

Bir dosya, doküman, yazı, kayıt bir şekilde depolanıyor. Peki bu kayıtlara tekrar ulaşmak istediğimizde ne yapacağız?

Ticari ya da ticari olmayan sistemler, yazılımlar ve ihtiyaçları geliştikçe bunların yönetimi de zorlaştı ve bir programsal alt yapıya ihtiyaç duyulur hale geldi. Bu ihtiyaca binaen geliştirilen ilk veritabanlarına örnek olarak, metinsel veri tutulabilen Notepad gibi doküman yönetimi yapılan programları örnek verebiliriz. Herhangi bir programlama dili ile dosyaya yazma-okuma yöntemleriyle kısıtlı ölçüde sorgulamalar yapıp, ekleme ve çıkarma işlemleri, sınırlı ölçüde düzenleme işlemleri yapılabilen metinsel veri yönetim araçları geliştirildi.

Ancak bilgisayar ve yazılım teknolojileri geliştikçe daha fazla veriyi, daha hızlı, güvenli ve mantıksal olarak depolama ihtiyacı doğdu ve böylelikle ilk veritabanı mimarisinin temelleri atılmış oldu.

Veriler veritabanında mantıksal bir düzen içerisinde saklanıp, çeşitli sorgular ve veri kayıt modelleriyle yönetilir hale geldi. Bu yazılımlar beraberinde yeni fikirleri ve ihtiyaçları getirdikçe veritabanı mimarisi de gelişti.

Bu veritabanında kayıtların sorgulama ve yönetimi işlemleri yapabilmek için veritabanı programlama geliştirildi.

## SQL, T-SQL VE VERİTABANI PROGRAMLAMA

Veritabanı programlama tekniği, ilk olarak matematiksel bir sözdizimine sahip olan **SQUARE** adlı bir sorgu dili ile geliştirildi. Bu teknolojinin daha kullanışlı olması ve yaygınlaşabilmesi için matematiksel söz dizimli **SQUARE** dilinden vazgeçilerek, İngilizce'ye benzer sözdizimine sahip bir dil oluşturulmuş ve **SEQUEL** (*Structured English Query Language*) olarak adlandırılmıştır. Daha sonra da bu **SEQUEL** dili, global bir sorgu dili olması nedeniyle English kelimesi çıkarılıp, İngilizce söylenişine paralel olarak **SQL** ile adlandırılmıştır.

İşte bir standart haline gelen **SQL** (*Structured Query Language*) Türkçesi, **Yapılandırılmış Sorgu Dili** olan veritabanı sorgu dilinin geliştirilmesinin sebebi de bu ihtiyaçları ortak bir mimari ile yapabilmektir. Veritabanını sorgulama işlemleri yapan genel çalışmalara da **veritabanı programlama** denilmektedir.

SQL sorgulama dili bir **ANSI** (*American National Standards Institute*) standardıdır. Ve tüm veritabanı yönetim sistemleri bu standardı destekler. Ancak SQL sorgu dili yapı ve yetenek olarak basittir. Çok gelişmiş işlemler ve ileri yetenekler yapabilecek kapasitede değildir. Bu nedenle veritabanı motoru üreticileri, SQL dili standartlarını desteklemek şartıyla, kendi SQL sorgu motorlarını hazırlamaktadır.

Bu sorgu motorlarıyla ileri seviye işlemleri hızlı ve kolay şekilde yapacak sorgu cümlecikleri oluştururlar. Ve her sorgu motoru, kendine özgü geliştiricisi firmanın alt yapısında kullanılabilir. Örneğin; Oracle sorgu dili olan PL/SQL'i, SQL Server sorgu motorunda çalıştıramayız. PL/SQL sorgu cümleleri SQL Server veritabanı sorgu motoru için sıradan cümleler anlamına gelir. Bu nedenle hata fırlatacaktır. İşte Microsoft da SQL Server için Transact-SQL açılımına sahip T-SQL adında yeni bir sorgu dili geliştirmiştir.

Bu yeni sorgu diliyle prosedür, trigger, hata yönetimi, sistem fonksiyonları, cursor gibi yüzlerce çeşit yeni yetenek kazandırmıştır. Bir veritabanı motorunun gücü sahip olduğu sorgu hızı, güvenlik ve diğer mimari yetenekleri haricinde sorgu dili yetenekleriyle de ölçülür diyebiliriz.

## İLİŞKİSEL VERİTABANI YÖNETİM SİSTEMİ

Verilerin tablolarda satır ve sütunlar halinde tutulduğu, yüksek veri tutarlılığına sahip, mimari yapısı ve index gibi özellikleri nedeniyle, sorgulama hızı yüksek veritabanı yönetimi için tasarlanan sistemdir. İlişkisel veritabanı bir veritabanı modelidir. Amacı veriye erişim ve hızı artırmanın yanında, veri tutarlılığını sağlayıp gereksiz veri tekrarını da engellemektir.

**RDBMS** (*Relational Database Management Systems*), yani İlişkisel Veritabanı Yönetim Sistemleri olarak adlandırılan kavram, genel bir yazılım alanını belirtir. Bu yazılımlar arasında ücretli, ücretsiz, açık kaynak ya da kapalı kaynak olarak birçok farklı veritabanı yönetim sistemleri mevcuttur.

## SQL SERVER NEDİR?

Dünya'da bireysel ve kurumsal anlamda en çok tercih edilen, düşük maliyetli, yüksek performanslı, Windows işletim sistemi üzerinde çalışan, SQL standardını destekleyen, kendine özgü T-SQL sorgu diline sahip olan ve paket hizmet çözümleriyle, tam bir veritabanı yönetim sistemi çözümü sunan gelişmiş bir ilişkisel veritabanı yönetim sistemidir.

Gerçek anlamda yükselişi SQL Server 2005 ile birlikte gerçekleşen, SQL Server 2008 ve 2012 ile çok hızlı gelişim gösteren ve kısa sürede büyük firmalar tarafından tercih edilen bir **İVYS** olmuştur.

Bu gelişim ve yükselişinde Windows Server ailesi, Office çözümleri, Microsoft'un geliştiricisi olduğu C#, VB.NET ve .NET teknolojisiyle birlikte tabi ki Microsoft'un **Cloud Computing** hizmeti olan **SQL Azure** ile **Windows Azure** gibi çözümlerin büyük payı vardır. Çünkü kurumlar tüm sorunlarını çözecek hizmetleri tek merkezden almayı tercih eder.

SQL Server hizmet paketinde bulunan **Reporting Services, Business Intelligence, Analysis Services** gibi birçok servis ve yönetim aracıyla birlikte gerçek bir veritabanı çözümüdür.

## NEDEN SQL SERVER?

İlişkisel Veritabanı Yönetim Sistemleri (**İVYS**) arasında kurumsal ve geliştirici dünyasında kabul gören ve kullanılan veritabanı yazılımlarının sayısı sınırlıdır. Açık kaynak ve ücretsizlere en çok kullanılan kategorisinde Oracle firması tarafından satın alınan MySQL ve açık kaynak olarak gelişmiş, güçlü bir alt yapı sunan PostgreSQL'i örnek gösterebiliriz.

Açık kaynak ve ücretsiz yazılım camiasında veritabanı kullanıcıları yani geliştirici ya da veritabanı yöneticileri genel olarak çok büyük hizmetlerde kullanmamak ile birlikte kendi çözümlerini kendileri üretmek zorundadırlar.

Örneğin; raporlama, sorgulama arayüzleri, IDE'ler, veritabanı çözümleri geliştirme ve istatistik gibi birçok alanda çeşitli üçüncü parti araçlar geliştirirler. Bu araçlar da ücretli ya da ücretsiz olarak dağıtılmaktadır.

Bu örnekten de anlaşılacağı gibi veritabanı sadece sorgu çalıştırılan ve kayıt eklenen bir veri depolama aracı değil, aksine tamamen bu alandaki tüm araç

ve hizmetleri kapsayan bir hizmet bütünüdür. Ve böyle de olmak zorunluluğu vardır.

Veritabanı yazılımı kullanarak kurumsal, gizliliği yüksek, güvenlik öncelikli ve veri kaybı yaşamayı göze almak istemeyen kurumlar bu nedenle ücretsiz ya da açık kaynak veritabanı yazılımları kullanıp, profesyonel destek olmadan ve verilerini bir güvence altında tutmadan çalışarak böyle bir riske girmek istemezler. Bunun yerine ücretli ama maliyeti çok yüksek olmayan ve profesyonel destek alabilecekleri bir çözüm isterler.

Bu şekilde ücretli ve Dünya'da bu hizmeti en iyi şekilde verecek **İVYS (İlişkisel Veritabanı Yönetim Sistemleri)** yazılımlarından bazılarını SQL Server, Oracle, DB/2 şeklinde sıralayabiliriz.

Bu yazılımların da kendi içerisinde bazı ayrımları ve farklı özellikleri mevcuttur. Örneğin, farklı platformlarda çalışması istenen, büyük veri hacimlerini işleyip yönetebilecek bir veritabanı yazılımı isteniyor ise çözüm Oracle gibi farklı platformlarda çalışan güçlü bir **İVYS** olacaktır. Ancak Oracle kullanıcıları genel olarak Linux tabanlı sistemlerde geliştirme ve yönetim sağladıkları için bu sefer güçlü bir IT departmanı ihtiyacı doğabilir. Bu da çoğu firma için büyük maliyet anlamına gelir ve Oracle lisans bedeli olarak da maliyeti yüksek bir veritabanı yazılımıdır. Bu maliyetinin yanında kolay yönetilebilir, hata olasılığı düşük, geliştirme ve yönetim araçları kullanıcı dostu (*user friendly*) arayüzlere sahip bir çözümü sunma konusunda bazı sıkıntılar yaşanacağı aşikardır. Bu durumda Oracle doğru bir seçim olmayabilir.

Ancak riski en aza indirmek, büyük verileri kolay ve hızlı yönetebilecek, yönetimi ve geliştirmesi kolay, geliştirme ve yönetim araçları kullanıcı dostu ve hata olasılığı düşük, kurulum ve konfigürasyonu kolay, lisans bedeli düşük, bir çok programlama dili ve platformuyla sorunsuz çalışabilen, geliştirme ve yönetimde kalifiye eleman bulma konusunda sıkıntı yaşatmayacak, IT masrafı düşük, profesyonel destek alınabilecek, işletim sistemi mimarisi üzerinde tam uyumlu bir veritabanı yazılımı isteniyorsa, bunun için SQL Server doğru seçimdir.

SQL Server 2000 sürümünde şu anki gibi güçlü ve profesyonel hizmet sunamamaktaydı. Ancak 2005, sonrasında 2008 ve 2012 ile Oracle'da bulunan birçok özellik, güçlü alt yapı ve mimari yetenek ile birlikte birçok yönetim

aracını hizmet paketine ekleyerek büyük bir gelişim gösterdi. SQL Server'ın geliştiricisi olan Microsoft firmasının işletim sistemi geliştirmesi ve gene SQL Server gibi geliştiricisinin Microsoft olduğu Windows Server ailesiyle çalışıyor olması, mimari açıdan %100 tam uyumlu ve ortak çalışabilen servis ve özellikleri, mimari olarak bütünleşik olması, veritabanı uygulamaları geliştirmek için Microsoft' un geliştirdiği ve .NET destekli tüm diller ile sorunsuz ve hızlı geliştirme yapabilmesi, .NET alt yapısı ve C# ile VB.NET gibi programlama dillerinin geliştiricisinin de Microsoft olması ve Java, C++ gibi diğer .NET dışındaki diller ile de sorunsuz çalışabilmesi gibi daha birçok özellik nedeniyle genel bir hizmet paketi isteyen kurumlar için doğru bir seçim olmaktadır.

Hem sunucu ve istemci işletim sistemini, hem veritabanı yazılımını, hem programlama dilini, hem tasarım hem yönetim araçlarını, hem de programlama yapılabilen ofis araçlarını (MS Office) tek bir firmadan alarak destek masrafını azaltıp, tek mimari üzerinden çalışmayı kim istemez ki? İşte bu nedenle, Windows tabanlı çalışan kurumların tercihi SQL Server olmayı sürdürecektir.

# SQL SERVER 2012 SÜRÜMLERİ VE ÖZELLİKLERİ

## SQL SERVER 2012 SÜRÜMLERİ

SQL Server, kurumların ihtiyaçlarını karşılayacak şekilde mantıksal olarak birden fazla sürüme ayrılmıştır. Bir geliştirici ile orta ölçekli bir firmanın, orta ölçekli firma ile de büyük bir bankanın farklı veritabanı ihtiyaçları olacaktır. Microsoft, bu ihtiyaçları göz önünde bulundurarak farklı sürümler oluşturmuştur.

Bu sürümler SQL Server'ın gelişimiyle birlikte farklılık göstermiştir. Yeni yayınlanan SQL Server sürümünde kullanılan bir sürüm, sonraki versiyonda kaldırılabilir. Ancak ana işlemler için hazırlanan sürümler değişmemektedir.

Şuan SQL Server 2012'de Standard Edition, Business Intelligence Edition ve Enterprise Edition olmak üzere üç ana sürüm var. Bu sürümlerin yanında ücretsiz olarak geliştiricilere sunulan Developer Edition ve kısıtlı özelliklere sahip Express Edition da SQL Server 2012 sürümünde yerini aldı.

SQL Server 2008'de var olan Datacenter Edition ve Workgroup Edition sürümleri SQL Server 2012'den kaldırılmıştır.

Bu sürümler içerisinde en gelişmiş olanı SQL Server 2008'de adı Datacenter Edition olan, SQL Server 2012'de ise Enterprise olarak değişen sürümdür. Business Intelligence Edition sürümü SQL Server 2012 ile tanıştığımız, iş zekası (business intelligence) uygulamalarına ihtiyaç duyulan, gene orta ve büyük işletmeler için hazırlanan sürümdür. SQL Server'ın temel sürümü olan Standard Edition ise temel ihtiyaçlara yönelik özellikleri içerir.

### **SQL SERVER 2012 EXPRESS EDITION**

SQL Server'ın kullanımı kolay, işlemci, ram ve veritabanı büyüklüğü gibi birçok mimari kısıtlamaları olan, küçük uygulamalar için kullanılabilecek ücretsiz sürümüdür. Bu sürümü ticari uygulamalarınızda da kullanmanızda herhangi bir sakınca yoktur. Genel olarak, küçük veritabanı kurulumu ve yönetimi gerçekleştirecek yazılım uygulamalarında istemcilere kurularak yazılımın veritabanı ihtiyacını karşılamak için kullanılır.

### **SQL SERVER 2012 DEVELOPER EDITION**

Geliştiriciler için gelişmiş test ve uygulama alt yapısı olarak hazırlanan bu sürüm, SQL Server 2012'nin tüm özelliklerine sahiptir. Ancak sadece test ortamlarında kullanılmak üzere lisanslanan bir Enterprise Edition kopyasıdır. Test ortamı dışılarına çıkarılacağında Enterprise Edition sürümüne yükseltmek için yeniden lisanslanmalıdır.

### **SQL SERVER 2012 WEB EDITION**

Web ortamında kullanılacak özelliklere sahip şekilde hazırlanan bu sürüm, SQL Server 2012'nin hosting firmaları ya da kurum ve kuruluşların web sunucularında SQL Server ihtiyacını karşılamak için hazırlanmıştır.

### **SQL SERVER 2012 BUSINESS INTELLIGENCE EDITION**

SQL Server 2012 ve Visual Studio yeteneklerinin kullanılarak hazırlanan bir kurumsal iş zekası geliştirme sürümüdür. Tarayıcı tabanlı veri arama ve görüntüleme gibi daha birçok özelliğe sahip bu sürümün Web, Developer ve Express sürümleri de bulunmaktadır.

### **SQL SERVER 2012 STANDARD EDITION**

SQL Server'ın değişmeyen ve ana sürümü niteliğindeki sürümdür. Küçük ve orta ölçekli kurum ve kuruluşlar için hazırlanan SQL Server'ın tüm ana özelliklerine sahip sürümüdür.

## SQL SERVER 2012 ENTERPRISE EDITION

Büyük kurum ve kuruluşlar için büyük verilerin yönetim, depolama, işleme ve analiz etme gibi bir çok görevi en iyi ve performans ile gerçekleştirebileceğiniz SQL Server sürümüdür. Bu sürüm, büyük verilerin hızlı ve performanslı yönetilebilmesi için mimari olarak da en yüksek donanım ihtiyaçlarını destekleyecek şekilde tasarlanmıştır.

## SQL SERVER 2012 COMPACT EDITION

SQL Server'ın gömülü sistemler, mobil cihazlar için geliştiği sürümüdür. Genel olarak düşük kapasiteli sistemlerde kullanılır.

## SQL SERVER LİSANSLAMA

SQL Server birden fazla olan sürümü ile farklı çözümler sunuyor. Bu çözümler gibi lisanslama yöntemi de farklı şekillerde olabilmektedir.

SQL Server için temel olarak 3 lisanslama yöntemi bulunmaktadır.

- **Sunucu artı aygıt istemci erişim lisansı (Server plus device client access license - CAL)**

Bu modelde SQL Server'ın çalıştığı makine ile SQL Server'a erişen her cihazın (PC, PDA, mobil telefon, terminal vs.) lisanslanması gerekmektedir.

- **Sunucu artı kullanıcı istemci erişim lisansı (Server plus user client access license - CAL)**

Bu modelde SQL Server'ın çalıştığı makine ile SQL Server'a erişen her kullanıcının lisanslanması gerekmektedir.

- **İşlemci lisansı (Processor license)**

Bu modelde SQL Server'ın üzerinde çalıştığı işletim sistemi ortamındaki her bir işlemci için bir lisans gerekmektedir.

SQL Server 2012 ile aşağıdaki lisanslama seçenekleri sunulacak:

- **Enterprise(EE) sürüm** için çekirdek tabanlı lisanslama (**processor license**).
- **İş Zekası(BI) sürüm** için sunucu + CAL lisanslama.
- **Standart(SE) sürüm** için çekirdek tabanlı lisanslama (**processor license**) ya da sunucu + CAL lisanslama seçenekleri.



SQL Server 2012'nin çıkması ile aşağıdaki üç sürüm devreden çıkacaktır:

- **Datacenter:** Özellikleri Enterprise Edition sürümünde yer alacaktır.
- **Workgroup:** Temel veritabanı ihtiyaçları için Standard Edition sürümü kullanılacaktır.
- **Standart for Small Business:** Temel veri tabanı ihtiyaçları için Standard Edition sürümü tek sürüm olacaktır.

Bu lisanslama modelinde tek bir SQL Server CAL ile tüm sürümlere erişim imkanı olacak.

Microsoft tarafından değiştirilen ve sadeleştirilen SQL Server 2012 sürüm özelliklerini inceleyerek, hangi SQL Server 2012 sürümünün size uygun olduğunu inceleyerek bulabilirsiniz.

SQL Server satın alma ve lisanslama işlemleriyle ilgili gerekli bilgilere aşağıdaki bağlantılardan ulaşabilirsiniz.

<http://msdn.microsoft.com/en-us/library/cc645993.aspx>

<http://www.microsoft.com/sqlserver/en/us/get-sql-server/how-to-buy.aspx>

## SQL SERVER KURULUMU & KURULUMU KALDIRMA

### SQL SERVER 2012 KURULUM GEREKSİNİMLERİ

SQL Server 2012'nin doğru çalışabilmesi için öncelikle doğru kurulum ve eksiksiz bir alt yapı üzerine kurulum gerçekleştirilmelidir.

SQL Server 2012 kurulumuna başlamadan önce işletim sistemi uyumluluğunu kontrol etmemiz gerekir. SQL Server 2012'nin çalışması için aşağıdaki işletim sistemlerinden birine ihtiyacı vardır.

- Windows Vista SP2 (*Ultimate, Enterprise, Business*)
- Windows 7 SP1 (*Ultimate, Enterprise, Professional*)
- Windows Server 2008 SP1 & SP2 (*Datacenter, Enterprise, Standard, Foundation*)

Eminim işletim sistemi listesinde klasikleşen Windows XP işletim sistemini aramışsınızdır. Ancak uzun süre önce Windows XP'ye destek vermeyeceğini açıklamıştı. Bu durumda tabi ki SQL Server 2012 için Windows XP desteği vermesi beklenemezdi.

İşletim sistemi seçimimizden sonra şu önemli alt yapıların sistemde kurulu olması gerekir. Bunlar;

- Microsoft .NET Framework 3.5 SP1
- Microsoft .NET Framework 4.0
- Windows PowerShell 2.0

Windows PowerShell 2.0 Microsoft'un sistem yöneticileri için sunduğu komut satırı tabanlı yönetim aracıdır. Bu teknoloji kısaca PowerShell olarak bilinir. PowerShell, Windows Server 2008 R2 ve Windows 7 işletim sistemlerinde varsayılan olarak kurulu gelirken Windows Server 2008 ve Windows Vista işletim sistemlerinde manuel olarak kurulumu yapılmalıdır.

Microsoft .NET Framework 3.5 SP1 SQL Server kurulumu sırasında kurulmaz. Bu nedenle SQL Server kurulumuna başlamadan önce makineye kurulup, daha sonra bilgisayar yeniden başlatılarak, Microsoft .NET Framework 4.0 makineye kurulmalı (dilerseniz SQL Server kurulumu sırasında otomatik kurulabilir) ve bilgisayarı yeniden başlattıktan sonra, SQL Server 2012 kurulumu için alt yapınız hazır hale gelmiş olacaktır.

## **SQL SERVER 2012 KURULUM & KURULUMU KALDIRMA İŞLEMLERİ**

SQL Server 2012 kurulum süreci oldukça basit bir araçtır. Kurulum sırasında diğer bir çok veritabanı yönetim sisteminde olduğu gibi herhangi bir parametre dosyası, path ayarı gibi kurulum ve bilgisayardan kaldırma işlemi sırasında ek yapılandırma işlemi gerektirmez.

Tabi ki SQL Server kurulumun bu kadar kolay olmasında en büyük etken, tek bir işletim sistemi platformunda çalışıyor olmasıdır. SQL Server sadece Windows platformunda çalışır. Windows ve SQL Server'ın geliştiricisi de Microsoft'tur. Bu da iki yazılımın birbiriyle sorunsuz ve %100 uyumlu çalışması anlamına gelir. İşte bu nedenle kurulum ve kaldırma işlemleri gayet basittir.



Bu bölümde sayfalarca ekran görüntüsü ile kitabın gereksiz şekilde doldurulması taraftarı değilim. Bu nedenle SQL Server 2012 kurulum ve kurulumu kaldırma işlemlerini, size verilen DVD eki içerisinde video anlatım olarak bulacaksınız.

## SQL SERVER ARAÇLARI

### BOOKS ONLINE

SQL Server'da birçok faydalı araç vardır. Bu araçların görevleri de kendilerine özgüdür. SQL Server Books Online, bir araç gibi görünmeyebilir. Ancak SQL Server'ı doğru ve tüm hatlarıyla anlatma görevi de Books Online aracına aittir.

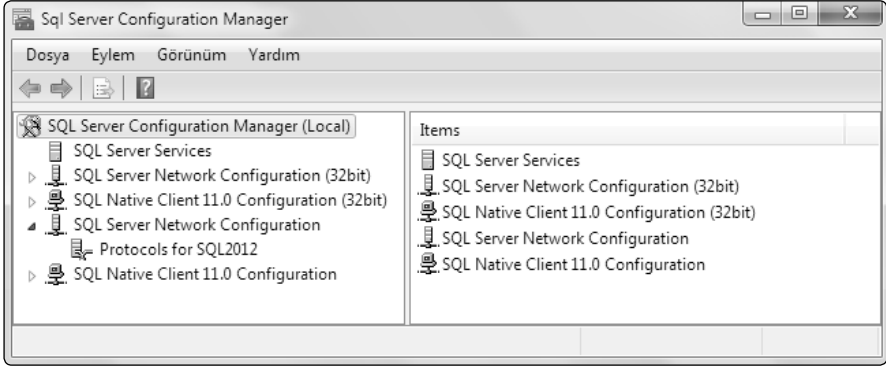
Bu nedenle hazırladığımız bu kitap ne kadar detaylı içeriğe sahip olsa da Books Online aracını mutlaka incelemenizi ve sürekli kullanmanızı öneririm.

SQL Server alt yapı çok geniş bir mimariye ve programlama gücüne sahiptir. SQL Server'ı ne kadar sık ve yoğun kullanırsanız, kullanın tamamını öğrenmeniz neredeyse mümkün değildir. Siz, rutin ve projelerinizde kullandığınız özellik ve güçlü noktalarını çok iyi kavrayın. SQL Server'ın diğer özelliklerine ihtiyaç duyduğunuzda ise araştırarak, bu kitabı ya da Books Online aracını kullanarak eksiklerinizi giderebilirsiniz.

İhtiyacınız olmayan bilgileri hafızanızda tutmak ya da tutmaya çalışmak sizi çok yorabilir ve asıl gerekli bilgileri öğrenebilmenizi daha zor hale getirebilir. Bu nedenle öğrendiğiniz bilgileri uygulayıp sonuçlarını gözlemledikten sonra unutmanız bir sorun teşkil etmeyecektir. Hatırlamanız gerektiğinde ufak bir araştırma ile tekrar aynı bilgileri kullanabilir hale geleceksinizdir. Bu kitabın ve Books Online'ın amacı öğretmek ve gerektiğinde unuttuğunuz bilgileri size hatırlatmaktır.

### SQL SERVER CONFIGURATION MANAGER

SQL Server Configuration Manager (*Yapılandırma Yöneticisi*), bilgisayar ve sistem yöneticileri için hazırlanmış hızlı, servis ve ağ yapılandırmasını gerçekleştirmeyi sağlayan bir yapılandırma aracıdır.



Bu isim ile SQL Server 2005 sürümünde duyuruldu ve bir çok farklı ayarın tek merkezden yönetilip yapılandırılmasını sağlar.

**Configuration Manager** iki ana yönetim sağlar. Bunlar;

- SQL Server Servis Yönetimi
- Ağ Yapılandırması

## SERVIS YÖNETİMİ

Servisler, **Configuration Manager -> SQL Server Services** içerisindeki sıralamaya göre şöyledir.

- **Integration Services:** Aynı ya da farklı platformlardaki verilerin taşınması ve dönüştürülmesi işlemlerini yapabildiğimiz Integration Service'lerini çalıştırır.
- **Analysis Services:** Analiz servislerini çalıştırır.
- **Full-Text Filter Daemon Launcher:** "Full-Text Search" için gerekli servisi çalıştırır.
- **Reporting Services:** SQL Server raporlama servisini çalıştırır.
- **SQL Server:** Veritabanının depolama, sorgulamalar ve diğer tüm veritabanı motorunun yapması gereken işlemleri gerçekleştirmek için kullanılan servistir. SQL Server'ın çalışması için bu servisin çalışıyor olması gerekmektedir.
- **SQL Server Agent:** SQL Server'da zamanlanmış görevleri yöneten servistir. Bu servisi kullanarak birden fazla ve farklı görevlere sahip zamanlanmış görevler oluşturabilirsiniz. Sizin belirleyeceğinizi gün ve saatte otomatik olarak bir veritabanının yedeğini almak bu görevlere bir örnek olabilir.

- **SQL Server Browser:** SQL Server'ın kurulu olduğu sistem tarafından tanınıp bulunabilmesi için gerekli servistir. Yerel ağ da server'ın bulunabilmesini sağlar.

## AĞ YAPILANDIRMASI

Sistem ya da bilgisayar yöneticisi olmayan, SQL Server'ın programsal geliştirmesini gerçekleştiren geliştiriciler için en önemli sistem sorunlarından biri SQL Server'a bağlanamama, ağ yapılandırmasında oluşan çeşitli sorunlar nedeniyle SQL Server ve bilgisayar ağı arasındaki uyumsuzluklar gibi sistem yönetimi taraflı hatalardır.

Bu hataları daha iyi kavrayıp gerektiği zaman müdahale edebilmeniz için öncelikle SQL Server ağ yapılandırmasını ve kurallarını öğrenmeniz gerekmektedir.

SQL Server ağ yönetimini sağlayabilmek için NetLibs (*Net-Libraries*) adında ağ kütüphanelerine sahiptir. NetLibs'ler istemci uygulama ile ağ protokolü arasındaki iletişimi sağlar ve bağdaştırıcı gibi çalışır.

SQL Server 2012 ile sağlanan NetLibs'ler şunlardır;

- Shared Memory
- TCP/IP
- Named Pipes
- VIA

İstemci ve server'ın birbiriyle ağ protokolü üzerinden iletişim sağlayabilmesi için her iki bilgisayarda da aynı NetLibs'lerin bulunması gerekir. İstemci bilgisayarda server'da bulunmayan bir NetLibs seçilmesi, hataya sebep olacaktır. ('Specified SQL Server Not Found' hatası)

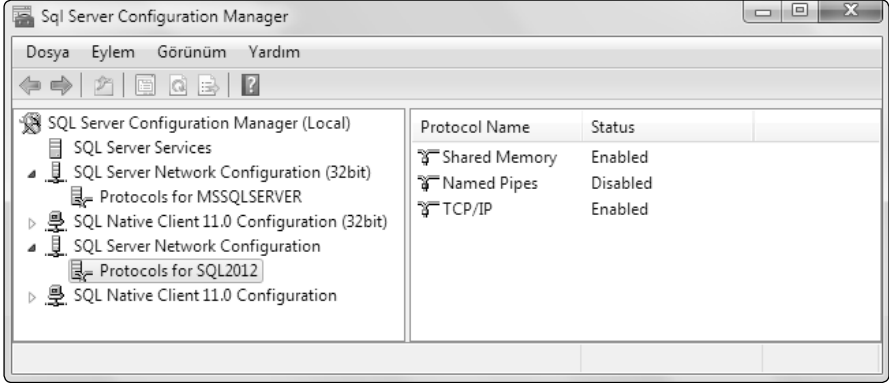
İstemci uygulamada kullanılan sürücü türü ODBC, OLE DB ya da SQL Native Client ne olursa olsun NetLib ile konuşacaktır.

Bir SQL Server sürücüsü ile NetLib'in haberleşme sırası aşağıdaki gibidir;

- İstemci uygulama sürücüsü (ODBC, OLE DB, SQL Native Client vs.) ile konuşur.
- Sürücü, istemci NetLib'i çağırır.
- NetLib, uygun ağ protokolünü çağırır ve veriyi server NetLib'ine iletir.
- Server NetLib'i, istekleri istemciden server'a iletir.

## PROTOKOLLER

SQL Server'ın hangi protokolleri dinleyebildiğini öğrenmek için SQL Server Configuration Manager aracımızdaki SQL Server Network Configuration sekmesini açarak, sağ bölümdeki protokoller listesinden tüm protokolleri görebilirsiniz.



SQL Server 2012 öncesi bir sürüm kurulu makinede protokollerden varsayılan olarak aktif olan Shared Memory'dir. Bu makine üzerine SQL Server 2012'de kurduğunuzda 2012 protokollerinde TCP/IP'nin de aktif olduğunu görebilirsiniz.

İstemcinin server ile iletişime geçip bağlantı kurabilmesi için server'ın, istemcinin kullandığı aynı port ile istemcinin kullandığı protokolü dinliyor olması gerekir. Bu nedenle Named Pipes ortamındaysak, yeni bir kütüphane eklememiz gerekebilir. Bunun için, **Protocols** sekmesinde **Named Pipes** protokolü sağ tıklanarak **enable** seçeneğiyle aktif hale getirilmelidir.

*"Neden tüm NetLib'leri aktif etmiyoruz?" diye sorduğunuzu duyar gibiyim. Tüm NetLib'leri aktif etmek bilgisayarda çalışan gerekli gereksiz tüm servisleri başlatmak gibidir. Yani gerek olmadığı halde SQL Server'ı yavaşlatırsınız. Hem de güvenliğin üst düzey olması gereken SQL Server için gereksiz yere ağınlığı açarak, bu şekilde güvenlik zafiyeti oluşmasına sebep olursunuz.*

SQL Server'da desteklenen protokoller aşağıdaki açıklanmıştır;

## TCP/IP

TCP/IP standart olan, en çok kullanılan ve mimari olarak ihtiyaç duyulan protokollerden biridir. SQL Server'a internet üzerinden erişmek için TCP/IP kullanılır. Çünkü IP kullanan tek protokol TCP/IP'dir.

SQL Server'ın internete açılması her ne sebepten olursa olsun güvenlik riskidir. Bu nedenle kritik veriler içeren bir SQL Server'ı internete açarken bir kez daha düşünün. Farklı çözüm yolları var ise, o yolları denemeniz daha az sıkıntılı olacaktır. Eğer tek seçeneğiniz SQL Server'ı internete açmak ise, bu durumda güvenlik adımlarını iyi hesaplayarak, detaylı bir güvenlik çalışması yapmanız önerilir.

## NAMED PIPES

Named Pipes protokolü az kullanılan bir protokol olmasına rağmen TCP/IP'nin olmadığı, TCP/IP altındaki server'ları isimlendirmeye izin veren **DNS (Domain Name Service)** olmadığı durumlarda kullanışlı olabilir.

SQL Server'a bilgisayar ya da sunucu adını vererek çalışabilirsiniz. Ancak DNS servisinin olmadığı durumda da sunucu adı, bilgisayar adı yerine **IP (Internet Protocol)** adresi vererek de çalışabilirsiniz.

## SHARED MEMORY

İstemci ile server'ın aynı makinede kurulu olduğu durumlarda aradaki veri iletişimini hızlandırmak, performansı artırmak için kullanılır. İstemci, server'ın veriyi sakladığı aynı eşlenmiş-bellek dosyasına doğrudan erişir. Bu yöntem ile veri gecikmesinin önüne geçerek iletişimi hızlandırır.

## VIA

Virtual Interface Adapter'in kısaltması olan **VIA**; iki sistem arasındaki bağlantıya özel, bir ağ arayüzüdür. Ve çok yüksek performansa sahiptir. Yüksek performanslı olmasının sebebi, özel olarak atanmış donanımdır.

## SQL SERVER MANAGEMENT STUDIO

SQL Server veritabanının programsal ve yönetsel olarak kolay ve hızlı olmasında en büyük paya sahip aracı tüm bu işleri ve daha fazlasını gerçekleştiren SQL Server Management Studio aracıdır.

SQL Server'ı kullanabilmeniz için tabi ki bu aracı kullanmak zorunda değilsiniz. Farklı üçüncü parti araçlar mevcuttur. Ancak size önerebileceğim en iyi araç gene Management Studio'nun kendisi olacaktır.

Bu aracın detaylarıyla bu kitabın sayfalarını doldurmayı doğru bulmuyorum. Bu tür araçların özellik ve arayüzleri önemlidir. Ancak her versiyonda farklılık gösterebilen bu tür araçların ekran görüntüleriyle kitabın sayfalarını meşgul etmektense kitap için hazırladığım DVD ekinde bulunan videolardan daha etkin ve hızlı şekilde öğrenip bu konuyu kavrayabilirsiniz.

SQL Server Management Studio aracının özellikleri DVD içerisinde video eğitim serisi şeklinde anlatılmıştır.

## SQL SERVER PROFİLER

SQL Server veritabanı motoru biz sorgu çalıştırmak da sürekli çok sayıda sorgu ile karşılaşır. Bunun sebebi; veritabanının yönetim ve istemcilerden, geliştirme araçlarından gelen isteklerin T-SQL diliyle veritabanına iletilmesidir. Biz araçları kullanırız ancak veritabanı motoruna giden yine T-SQL sorgularıdır.

Bu sorguları izlemenizi sağlayan aracın adı SQL Server Profiler'dır. Bu araç gerçek-zamanla sorgu izleme aracıdır. **Performance Monitor** ,sistem yapılandırma konusunda meydana gelen olayların tamamını izlese de, SQL Server Profiler özel durumları izlemekle ilgilenir. SQL Server Profiler, nasıl yapılandırıdığınıza bağlı olmakla birlikte, server'ınızda çalışan her sorgu cümlesinin özel T-SQL söz dizimini verir.

SQL Server Profiler, performans takibi, sorgu performansı, sorgu analizleri için veritabanı yöneticilerinin vazgeçilmez izleme aracıdır.

## SQL SERVER INTEGRATION SERVICES (SSIS)

Önceki sürümlerde **DTS** (*Data Transformation Services*) olarak bilinen **SSIS** bu araç SQL Server için karmaşık veri import ve export işlemleri için kullanılan



gerekli, güçlü ve gelişmiş bir entegrasyon aracıdır. **SSIS**, OLE DB ya da .NET veri sağlayıcıya sahip herhangi bir veri kaynağından veriyi alarak, SQL Server veritabanındaki tabloya gönderir.

## SQL SERVER REPORTING SERVICES

Reporting Services; SQL Server'ın rapor üretme, düzenleme ve çalışma platformudur. Web ortamında raporlar sunmak için yerleşik bir Windows Web Server ile birlikte çalışır. Raporlar, **Report Definition Language (RDL)** adı verilen XML tabanlı bir tanımlama dili kullanılarak hazırlanır. Business Intelligence Development Studio, hem basit hem de karmaşık raporlar için şablonlar kümesi sağlar. Raporlar, Reporting Services tarafından talep edildiğinde, **RDL** dosyasına yazılır.

## SQL SERVER BUSINESS INTELLIGENCE DEVELOPMENT STUDIO

Business Intelligence Development Studio, SQL Server Business Intelligence'a özgü ek proje türleri içeren Microsoft Visual Studio uygulamasıdır. Business Intelligence Development Studio, Analysis Services, Integration Services ve Reporting Services projelerini içeren iş çözümleri geliştirmek için kullanacağınız birincil ortamdır. Her proje türü, karar destek çözümleri için gerekli nesnelerin oluşturulması için şablonlar sağlar ve nesnelerle çalışmaya yönelik çok çeşitli tasarımcılar, araçlar ve sihirbazlar sunar.

Business Intelligence Development Studio'nun menü ve özellikleri Visual Studio versiyonu ve kurulu olup olmamasıyla belirlenir. Visual Studio kurulu ise Visual Studio menüleri ve SQL Server Analysis Services ile ilişkili menüler ve şablonların karışımı olacaktır.

Özetlemek gerekirse; bu araç bize analiz (*Analysis Services*), raporlama (*Reporting Services*), Integration Services paketleri gibi birçok iş geliştirme araçlarını kullanabilmemiz için çeşitli araç, şablon ve menülerden oluşan bir geliştirme aracıdır.

## BULK COPY PROGRAM (BCP)

**Bulk Copy Program (BCP)** komut satırı aracıdır. Bu araç, SSIS geliştirilmeden önce SQL Server'ın import/export işlemlerini gerçekleştirirdi. **BCP**, halen

kullanılır ve bu aracı kullananların başında SQL Server'ın kendisi gelmektedir. Kurulum sırasındaki işlemlerin komut satırından hızlı olarak gerçekleşmesini sağlar.

## SQLCMD

Microsoft'un komut satırı kullanmayı seven geliştiriciler ve sistem yöneticileri için geliştirdiği komut satırı aracıdır. Bu aracın diğer bir adı da OSQL olarak bilinir.

Script'lerin çalıştırılması için kullanılan bu araç işini çok iyi yapabilir. Ancak genel olarak bu aracın yaptığı işlemleri yapan birçok SQL Server aracı vardır.

## ÖZET

Bu bölümde anlatılan araçlar SQL Server için faydalı ve her biri kendi alanında uzmanlık seviyesinde gerekli araçlardır. Ancak genel olarak en çok kullanacağınız araç SQL Server Management Studio'dur.

Tabi ki tüm araçları kullanmayabilirsiniz. Ancak gerçek bir uzman öncelikle takım çantası (geliştirme araçları) geniş ve seçenekleri çok olmalıdır. Bu nedenle bu araçlardan haberdar olmanız, nasıl kullanıldığını bilmeniz ve burada anlatılmayan diğer araçları da zaman içerisinde araştırıp öğrenmeniz fayda vardır.

## SQL SERVER VERİTABANI NESNELERİ

SQL Server üzerinde her şey birer nesnedir demek hiç de yanlış olmaz. Bu nesneler veritabanı mimarisini oluşturur. Her nesneyi yap-boz'un bir parçası olarak düşünürsek bunlar bir araya getirildiğinde tam bir veritabanı oluşmaktadır.

Bu bölümde veritabanını oluşturan mimarinin parçalarını inceleyerek işe başlıyoruz. Bu parçaları öğrendiğinizde veritabanı yapısını programatik olarak daha iyi anlayacak ve ileriki konularda iyi bir alt yapıya sahip olacaksınız.

## BU BÖLÜMDE NELER ÖĞRENECEĞİZ?

Veritabanının yap-boz gibi nesnel parçalardan oluştuğunu belirtmiştik. Bu mimarinin oluşmasının temel nedeni veritabanının gerekliliğini zorunlu hale getiren veri kavramıdır.

SQL Server sadece bir veritabanı değildir. Yani sadece güvenli şekilde verilerin saklandığı, ekleme, güncelleme, silme gibi işlemlerin yapıldığı bir mimari değildir. SQL Server bir hizmet çözümü paketidir.

SQL Server'da amaç; sadece depolama işini yapmak değil geliştirilen birçok servis ile bu verinin içeri ve dışarı aktarılmasını sağlamak, güvenlik, raporlama, istatistik, analiz, performans, iş zekası ve yönetimi gibi birçok farklı görevi tek bir çözüm olarak sunmaktır.

Bu bölümde SQL Server veritabanı mimarisini oluşturan nesneleri inceleyerek mimarinin bütününe görmeye çalışacağız.

## VERİTABANI NESNELERİNE GENEL BAKIŞ

SQL Server ilişkisel veritabanı yönetim sistemi birçok nesne içerir. Bu nesnelerden bazıları şunlardır;

Database	CLR assembly	User-Defined Function
Index	Filegroup	User-Defined Data Type
Table	Diagram	View
Transaction Log	Full-Text Catalog	Trigger
Rule	User	Stored Procedure

## DATABASE NESNESİ

Veritabanı nesnesi SQL Server ile ilişkili en önemli, temel ve en yüksek seviyeli nesnedir. Bu bölüm sonunda anlayacağınız bir gerçek şu ki; diğer nesnelerin birçoğu veritabanı nesnesinin elemanlarıdır. Yani veritabanı nesnesi olmadan bahsedeceğimiz nesnelerin birçoğu anlamsız kalacaktır. Çünkü bu nesnelerin çoğu bir veritabanı nesnesi üzerinde çalışmak için tasarlanmıştır.

Veritabanı, nesneler içerisinde en temel nesnelerin başında gelen tablo ve view, stored procedure, function, trigger gibi birçok nesneyi içinde barındıran bir nesnedir.

SQL Server ilk yüklendiğinde dört sistem veritabanı ile birlikte kurulur.

- master
- model
- msdb
- tempdb

Bu veritabanları SQL Server'ın doğru çalışması için gereklidir. Eksikliklerinde SQL Server doğru çalışmayacaktır.

## MASTER

**master** veritabanı, sistemin bütünü izlemeyi sağlayan, sistem tabloları kümesini içine alır. Sahip olduğu sistem prosedürleri ve view'ler ile veritabanı işlemlerinin, tablo ve diğer nesneler ile ilgili bilgi alma trafiğinin beynini oluşturuyor da diyebiliriz.

Örneğin;

**CREATE DATABASE DIJIBIL** kodunu çalıştırarak oluşturacağınız;

	name
1	master
2	tempdb
3	model
4	msdb
5	ReportServer\$SQL2012
6	ReportServer\$SQL2012TempDB
7	AdventureWorks2008R2
8	DIJIBIL

**DIJIBIL** isimli veritabanı **master** veritabanında **Views -> System Views** içerisinde **sys.databases** view'ini çalıştırarak tüm veritabanlarımızı listeleyecektir.

---

```
SELECT * FROM sys.databases
```

---

Server'ı tanımlayan her şey **master** veritabanında saklanır. Bu nedenle içerdiği veriler kritiktir ve silinemez.

**master** veritabanındaki sistem tabloları, view ve prosedürler genel olarak veritabanı yöneticileri tarafından kullanılsa ya da onların ihtiyaçlarını görecek

şekilde hazırlanmış olsa da geliştiriciler için de performans analizi ve bilgi almak için gereklidir.



Microsoft sistem tablolarını kullanmanın tehlikeli olduğunu bildirir ve sadece son üç versiyon için sistem tablolarının kullanımını önerir. Unutmayalım ki, SQL Server geliştikçe veritabanı yapısı, nesneler ve `master` veritabanı yapısında da değişiklikler olabilir. Hatta Microsoft, `master` veritabanının her versiyonda değişikliğe uğradığının garantisini veriyor da diyebiliriz. Bu nedenle bu tablolarda değişiklik yapmamanız kesinlikle önerilir. Küçük bir fayda sağlamak için yapacağınız değişiklik SQL Server'ın çalışmamasına sebep olabilir.

## MODEL

`model`, veritabanı oluşturma işleminde referans alınmak için hazırlanmış bir model veritabanıdır. Yeni bir veritabanı oluştururken `model` veritabanı yeni bir şablon oluşturur. Yani isterseniz yeni oluşturacağınız veritabanlarının nasıl olacağını, `model` veritabanında değişiklik yaparak belirleyebilirsiniz. Örneğin; sistem üzerinde oluşturulan her yeni veritabanına, kopyalanan kullanıcı grupları oluşturabilirsiniz. `model` veritabanı, diğer veritabanları için şablon olduğu için gerekli bir veritabanıdır ve silinemez.

Oluşturduğunuz herhangi bir veritabanı `model` veritabanının büyüklüğünde olmalıdır. Yani `model` veritabanı 100 MB ise oluşturacağınız veritabanı bu büyüklükten küçük olamaz. Bu nedenle varsayılan ayarları değiştirirken her zaman dikkatli olmanız önerilir.

## MSDB

`msdb` veritabanı, SQL Agent'in herhangi bir sistem görevini sakladığı yerdir. Veritabanı üzerinde herhangi bir zamanlayıcı, örneğin yedekleme zamanlayıcısı gibi ya da stored procedure'un ilk çalışması ile ilgili kayıtlar da `msdb` de tutulur.

## TEMPDB

`tempdb`, SQL Server'da gerçekleştirilen sorgulama işlemlerine geçici hafıza görevi gören `temporary database`'dir. Önemli bir veritabanıdır. Karmaşık ya da geniş bir sorgulama yaptığınızda, SQL Server geçici tablo oluşturmaya gerektiğinde bu işlemleri `tempdb` ile yapar. Aynı zamanda T-SQL ile geçici tablo oluşturduğunuzda ya da geçici veri depolama işlemlerinde bu verilerin tutulduğu yer `tempdb`'dir.

**tempdb** veritabanının sadece içerdiği veri ya da nesneler geçici değil, veritabanının kendisi de geçicidir. SQL Server her başlatıldığında **tempdb** sıfırdan yeniden oluşturulur.

## ADVENTUREWORKS

Tüm veritabanı yönetim sistemlerinde geliştiricilerin kullanacağı örnek veritabanı tasarımlarının olması örnek çalışmaları açısından faydalı olmaktadır. SQL Server eski sürümlerinde **Northwind** veritabanı örnek şeması vardı. Ancak SQL Server özelliklerinin büyük yenilikler içermesiyle bu tür eski örnek şemalar yeterli olmamaya başladı. Microsoft, SQL Server'ın gücünü temel özellikleriyle tam olarak yansıtabilecek bir veritabanı şeması hazırlamak istiyordu. **AdventureWorks** bu düşüncenin bir ürünüdür. Mimari olarak detaylı ve büyük bir alt yapı olarak hazırlanmıştır.

Aşağıdaki adresten bu veritabanının eski ve yeni sürümlerine ulaşabilirsiniz.

<http://msftdbprodsamples.codeplex.com>

## ADVENTUREWORKSDW

**AdventureWorksDW**, (*DW, Data Warehouse = Veri Ambarı*) Analysis Service için hazırlanan örnek veritabanı şemasıdır. Microsoft'un transaction veritabanı örneği ile analiz örneğinin birlikte çalışmasını gösteren bir örnektir.

## TRANSACTION LOGS (İŞLEM GÜNLÜKLERİ)

Veritabanı dosyası verilerin kaydolduğu ve okunduğu dosyalardır. Ancak bu işlemin daha doğru ve düzgün çalışabilmesi için farklı bir mimari geliştirilmiştir. Öncelikle veritabanında gerçekleşecek herhangi bir işlem direkt olarak veritabanı dosyasına yazılmaz. Önce değişiklikler transaction log dosyasına yazılır. Zaman olarak bir sonraki adımda, veritabanı bir checkpoint yaparak, log dosyasına yazılan değişikliklerin gerçek veritabanı dosyasında üretildiği zaman düzenler.

Veritabanı dosyası rastgele, fakat log dosyası sırasıyla yazılmıştır. Veritabanı dosyasının rastgele yapısı hızlı erişim imkanı kazandırırken, log dosyasının sıralı yapısı değişiklikleri sırasıyla izleyebilme imkanı sağlar. Veritabanı üzerinde çalıştırılan ve gerçekleştiği varsayılan işlemler log dosyasında biriktirilir ve sonra server bu değişiklikleri fiziksel olarak veritabanı dosyasına gerçekleştiren zaman dilimiyle yazar.

## FILEGROUPS (DOSYA GRUPLARI)

SQL Server'da tüm tablolar ve veritabanı ile ilgili her şey tek bir veritabanı dosyasında saklanır. Bu dosya **Primary Filegroup** adı verilen birincil dosya grubu üyesidir. Bununla birlikte bu düzenlemeye bağlı olmak zorunda değilsiniz.

SQL Server pek sınır sorunu olmamakla birlikte 32.000'in üzerinde **Secondary Files** denilen ikincil dosyalar tanımlayabilirsiniz.

Sadece bir tek **Primary Filegroup**, yani birincil dosya grubu olmakla birlikte, 255 adet ikincil dosya grubu oluşturabilirsiniz.

İkincil dosya grubu `CREATE DATABASE` ya da `ALTER DATABASE` komut seçenekleriyle oluşur.

## DIAGRAMS (DİYAGRAMLAR)

Tablolar aslında SQL kodları ile oluşturulan yapılandır. Ancak görsel olarak da daha iyi anlayıp yönetebilmemiz adına tablo tasarımı denen bir olgu ile çalışırız. Diyagram'lar da bu tablo tasarımı olgusunun bir adım daha ötesidir diyebiliriz. Bir tablonun içeriğini ve var ise başka tablolar ile ilişkilendirmelerini gösteren ve yönetebileceğimiz bir veritabanı görselleştirme aracıdır.

Veritabanı programlama alanında uğraşacak geliştiricilerin sık göreceği yapı Entity-Relationship kısa adıyla ERD'dir. ERD diyagramda veritabanı iki parçaya ayrılmıştır. Varlıklar (üretici, ürün gibi) ve ilişkiler(işlenecek malzemeler ve satılacaklar) gibi.

SQL Server geliştikçe geliştirme ve modelleme araçları da gelişmektedir. Bu nedenle her yeni sürümde SQL Server Diagram Designer'a yeni özellikler eklenip kullanımı geliştirilmektedir. Bu tasarım aracı varlık-ilişkili modeli tam olarak yansıtamasa da veritabanı modellenmesinde etkili bir araçtır.

## SCHEMAS (ŞEMALAR)

Veritabanı mimari olan çokça nesneye ve katmana sahiptir. Bunların nesnesel olarak isimlendirilebilmesi için mantıksal bir isimlendirme yapısına ihtiyaç vardır. Schema'lar veritabanı ile içerisindeki nesnelerin mantıksal isimlendirmesini yapar.

SQL Server'da varsayılan Schema adı **dbo**, yani **Database Owner = Veritabanı Sahibi** anlamına gelir. Her kullanıcı varsayılan Schema'ya sahiptir. SQL Server tüm nesneleri otomatik olarak varsayılan Schema içerisinde arar.

Eğer ulaşılmak istenen nesne varsayılan Schema içerisinde değilse, `schema_ad.nesne_ad` şeklinde nesnenin içinde bulunduğu Schema adını da nesnenin başında belirterek yazmalıyız.

Schema yapısı doğru ve yerinde kullanılmadığı takdirde karmaşa ve birçok sorun getirir. Bu nedenle özel durumlar haricinde kullanılması gerekli değildir.

## TABLES (TABLOLAR)

Bir veritabanının en temel yapısı tablolardır. Veritabanı nesnelerinin neredeyse tamamı bir tabloya ya da içerisindeki veriyle alakalı işlem yapmak için tasarlanmıştır. Tablolar satır ve sütunlardan oluşur. Alan veri(sütunlar), tabloda saklanacak verilerin her biri için bir alan oluşturmak ve bu alanlara yazılacak verinin tipini belirtmeye yarar. Kayıt veri (satırlar), belirlenen alanlara verilerin girilmesiyle oluşur. Satırlar girilirken sütunların veri tipleri ve kısıtlamaları kontrol edilir. Bu kontrol işlemini gerçekleştirmek için tabloların yapısal bilgilerini tutan metadata'lar kullanılır.

## INDEXES (İNDEKSLER)

Index, sadece tablo ve view için bulunan bir nesnedir. Index sıralanma işlemini daha disiplinli yaparak performans artırmayı ve hızlı sorgu sonucu döndürmeyi amaçlar.

Index kavramı denince akla ilk gelen örnek kütüphane sistemleridir. Bir kütüphane bulunması gereken bir kitabı içerisindeki on binlerce kitabı tek tek inceleyerek bulmak yerine, bir index hazırlanarak belli bir arama ve sıralama algoritması üzerinden daha kısa sürede bulunması sağlanır.

En basit sıralama algoritması hepimizin bildiği, eski telefon rehberlerindeki A' dan Z'ye sıralama yaparak arama işlemini, bir nebze kolaylaştırıp hızlandıran index'leme yöntemidir.

Index'ler iki gruba ayrılır.

- **Clustered:** Tablo başına sadece bir clustered index kullanılabilir. Clustered index ile index'lenen tablo, fiziksel olarak bu index ile sıralanır.



Örneğin; bir ansiklopedi dizini hazırlıyorsanız, ansiklopedideki bilgiler sayfa numarasına göre sıralanacaktır.

- **Non-Clustered:** Tablo başına birden fazla non-clustered index tanımlanabilir. Non-clustered index mimarisinde fiziksel bir sıralama söz konusu değildir. Yani, clustered index gibi verinin kendisini değil pointer'ını tutmaya yarar.

## CONSTRAINTS (KISITLAMALAR)

Constraint'ler tablo sınırları içinde bir nesnedir ve adından da anlaşılacağı gibi tablo üzerinde bazı kısıtlamalar ile veri bütünlüğünü sağlamayı amaçlar.

## VIEWS (GÖRÜNÜMLER)

View bir tablonun sanal görüntüsüdür. Gerçekte içerisinde bir veri olmayan, ancak içeriğinde belirtilen SQL sorgu bloğunun getireceği verileri kendisine verilen kısa bir ad ile kullanıcıya tablo olarak göstermek için tasarlanmıştır. Bir ya da daha fazla tablodan veri çekerek tek sorgu sonucu olarak döndürebilir. Bu işlemi yapacak sorgu planı view içerisinde saklanır.

View'ler genel olarak birkaç nedenle kullanılır. Bunlar; güvenlik, performans ve kullanım kolaylığıdır. Hazırlanacak sorguların sonuçlarının tamamının görünmesini istemeyebiliriz. Örneğin; tablonun tamamının görünmesi veritabanı güvenliği açısından bir risk olabilir. Ancak hazırlanacak bir view' in kullanımı sağlayabilir ve tablonun sadece belirlenen sütunlarını view içerisinde çağırarak bir kısıtlama yapabiliriz. Bu da veri erişim güvenliğini artıracaktır. Aynı zamanda performans ve kullanım kolaylığı gibi tüm faydaları ileride anlatılacak view bölümünde tüm detaylarıyla anlatılacaktır.

## STORED PROCEDURES (SAKLI YORDAMLAR)

Stored procedure, diğer adıyla sproc'lar SQL Server programlama (T-SQL) alt yapısının en önemli ve çok kullanılan özelliklerinden biridir. Sproc'lar veritabanı içerisinde yapılacak işlemlerin bir program düzeninde gerçekleşmesini sağlayan T-SQL programcılarıdır.

İçerisinde T-SQL gücünü kullanarak değişkenler, girdi ve çıktı parametreleri, hata yakalama, klasik sql sorguları gibi birçok özelliği kullanarak işlemler yapılabilir. Sproc'lar güvenlik, performans ve hızlı işlem gerçekleştirebilme gibi birçok faydaya sahiptir.

- Sproc'lar, server başlatıldığında derlenir (*compiling*) ve derlenmiş olduğu için yeniden kullanımında zamandan tasarruf sağlar.
- Sproc'lar kısa bir isim ile adlandırılırlar. Ve içerisindeki veriyi şifreleme özelliği de bulunduğu için dilerseniz T-SQL sorgularınızı sproc'un kullanıcılarından gizleyerek veritabanı güvenliği açısından ek fayda sağlayabilirsiniz.
- Bir sproc başka bir sproc tarafından çağrılabilir. Yani iç içe kullanılabilir. Bu da daha az kod ile daha çok işlem yapmayı ve fonksiyonel bir veritabanı programlamayı sağlar.
- .NET tabanlı diller ile SQL Server içerisinde sproc geliştirilebilir.
- Sproc'lar yazılım projesi içerisinde ADO.NET ya da benzeri veri katmanları tarafından direk olarak kullanılabilir. Bu da yazılım içerisinde herhangi bir T-SQL kodu yazmamanızı sağladığı gibi, sproc kod bloğu içerisinde herhangi bir değişiklik yapılacak ise, veritabanında bulunan sproc'un **ALTER PROCEDURE** ile güncelleme yapılarak gerçekleştirilebilmesini sağlar.

## TRIGGERS (TETİKLEYİCİLER)

Trigger'lar (*tetikleyici*) tablolar üzerinde gerçekleştirilen ekleme, güncelleme, silme ve kopyalama gibi işlemler gerçekleştikten sonra ya da önce gerçekleşmesi istenen işlemlerin otomatik olarak gerçekleşmesini sağlayan T-SQL programcıklarıdır.

SQL Server ve diğer veritabanı yönetim sistemlerinde önemli bir yere sahip olan trigger'lar, hızlı, yönetilebilir ve profesyonel veritabanları için sproc'lar kadar olmasa da sık kullanılan nesnelerdir.

## USER-DEFINED FUNCTIONS (KULLANICI TANIMLI FONKSİYONLAR)

UDF, Türkçe anlamı Kullanıcı-Tanımlı Fonksiyonlar sproc'lara çok benzer. Hatta arasında çok az fark var diyebiliriz.

UDF'lerin Sproc'lara göre bazı farklılıkları;

- UDF'ler yardımcı programcıklardır ve sproc'lar gibi güçlü ve veritabanı üzerinde sistematik bir değişiklik, e-mail gönderimi, tabloların değişimi, sistem ya da veritabanı parametrelerinin değiştirilmesi gibi bir işlem gerçekleştiremez.

- UDF'ler image, cursor, timestamp, text ve ntext tipleri haricinde SQL Server veri tiplerinin bir çoğuna değer döndürür.

UDF'leri programlama dillerindeki fonksiyonlar gibi düşünebiliriz. Dışarıdan birden fazla parametre ile değer alır ve bu değerleri belirlenen işlemlere tabi tutulur ve sonucunda da geriye bir değer döndürür. Bu şekilde veritabanında fonksiyonel programlama yapmamıza yardımcı olur.

## **USER-DEFINED DATA TYPES (KULLANICI-TANIMLI VERİ TİPLERİ)**

Kullanıcı-Tanımlı veri tipleri SQL Server içerisinde özel veri tipi oluşturup kullanmanızı sağlar. Bu veri tipleri temel olarak sistem veri tipleri gibidir.

Bu özellik teknik olarak sunulsa da kullanırken dikkatli olmanız gerekmektedir. Bir nevi verinizi, veritabanınızı SQL Server'ın sisteminde olmayan veri tipiyle geliştiriyor olmanız, geliştirici olarak sorumluluğunuzu artırdığı gibi hata ve olası riskleri de artırmaktadır.

## **DEFAULTS**

Default'lar bir sütuna değer girilmemesi halinde varsayılan olarak veritabanı tarafından yazılması, istenen değeri temsil eder. Yani default değere sahip bir sütuna kayıt girmeyebilirsiniz ve bu takdirde otomatik bir değer atama gerçekleşir.

## **USERS & ROLES**

User'lar SQL Server sistemine bağlanmak isteyen, erişim hakkı tanınmak istenen kullanıcılar için verilecek veritabanına erişim yetkilendirme nesnesidir.

Role ve User'lar birbirinin tamamlayıcısıdır. Bir User'ın SQL Server sisteminde hangi göreve, yetkiye, erişim hakkına sahip olduğunu Role'ler ile belirleyebiliriz. Bir User'a birden fazla Role atanabilir.

## **RULES**

Rule'ler Constraint'ler gibi veri sınırlama işlemlerinde kullanılır. Tabloya eklenecek ya da güncellenecek veri, kural dışı ise bu işlem reddedilir.

Rule'ler Microsoft tarafından SQL Server'ın sonraki versiyonlarında kaldırılacağı için bu nesneyi kullanmakta dikkatli olmanız ve büyük çaplı projelerde kullanmamanız önerilir.

Default için en çok kullanılan örneklerden biri veri girişi sırasında tarih, saat sütununa `GETDATE()` gibi bir otomatik 'o anki zaman' bilgisini veren değerin girilmesidir.

## TANIMLAYICILAR VE İSİMLENDİRME KURALLARI

SQL Server içerisinde her şey bir nesne ya da tanımlayıcıdır ve bunların programsal olarak kullanılabilmesi için tabi ki isimlendirilmiş olması gerekir.

Aşağıdakiler isimlendirilen nesnelere örnek olarak verilebilir.

View	Login	Server
Stored Procedure	Index	Role
Trigger	User	Constraint
User-Defined Function	Database	Column
User-Defined Type	Table	File
Schema	File	Full-Text Catalog

Ve bu isimlendirilebilir nesnelere daha birçok örnek verilebilir.

## İSİMLENDİRME KURALLARI

SQL Server'da boşluklar ve anahtar kelimelerle aynı olan isimlendirmeler yapılabilir. İsimlendirme konusunda SQL Server esnektir. Ancak bazı kurallara uymak ve esnekliğin şartlarına uymak gerekir.

Uyulması gereken ana isimlendirme kuralları şunlardır;

- Normal nesneler için 128 karakter, geçici nesneler için 116 karakter uzunluğunda isimler verilebilir.
- Tanımlama için kullanılacak ismin ilk harfi Unicode 2.0'a uygun bir harf olmak zorundadır. Türkçe karakter kullanılmamalı, büyük ya da küçük harf olarak A-Z arasında Türkçe olmayan karakterler ile başlamalıdır. İlk harften sonra bu kural geçerli değildir ve farklı karakterler kullanılarak isimlendirme yapılabilir.

- SQL Server anahtar kelimeler ile aynı olan ya da boşluk içeren isimler, köşeli parantez ( [ ] ) ya da tırnak ( " " ) içerisinde gösterilmelidir. Bunun amacı, veritabanınızı kullanmak istediğiniz esnek kurallara göre bilgilendirmektir.

Çift tırnak özelliğini kullanabilmeniz için `QUOTED_IDENTIFIER` özelliği `SET QUOTED_IDENTIFIER ON` şeklinde `SET` etmeniz gerekmektedir.

Köşeli parantez kullanımı da [boşluklu tanımlayıcı] şeklinde boşluklar için kullanılabilir. Eğer boşluk kullanılacak ise bu şekilde köşeli parantez kullanılması önerilmektedir.

Bu kuralların bir kısmı genel olarak programlama kurallarıdır. Her platformda olduğu gibi yukarıda bahsettiğimiz kurallar da SQL Server platformu ve tüm tanımlamalar için geçerlidir.



Bu esnek kurallarının SQL Server mimarisinde var olması, kullanılmasını tavsiye niteliğinde algılanmamalıdır. Aksine bu esneklik kurallarını çok gereklilik duyulmadığı takdirde kullanılmamalıdır.

