**Bircan SATI**

**130757020**

**Bilgisayar Bilgi Güvenliği ve Yönetimi**

1. **Konu**

**Single Sign-On (SSO)**

**Section 1.3.3**

1. **Konu**

**Relational Data Base Access Controls Using SQL**

**Section 1.4.1**

1. **Konu**

**Makro Virüsler**

**İçindekiler**

1.Single Sign-On (SSO)

1.1 Tarihçe

1.2 SSO Nedir ?

1.3 SSO Senaryoları

1.3.1 SSO Uygulanmayan Login Senaryosu

1.3.2 Temel SSO Uygulanan Login Senaryosu

1.3.3 Merkez Domainli SSO Uygulaması

1.3.4 SSO Yaygın Kullanım

1.4 Token Nedir

2. Relational Data Base Access Controls Using SQL

2.1 Primary Key

2.2 SQL İfadeleri

2.2.1 Create

2.2.2 Insert ve Delete

2.2.3 Select

2.2.4 Update

2.3 SQL İzinleri

2.4 SQL İzin Bileşenleri

2.4.1 İlkeler

2.4.2 Sağlanabilirler

2.4.3 İzinler

2.5 İzinler

2.6 Sunucu ve Veritabanı Rolleri

2.7 SQL Server İzinleri

3.Makro Virüsler

3.1 Makro Virüs Nedir ?

3.2 Makro Virüsler Nasıl Yayılır ?

3.3 Riskler

3.4 Makro Virüs Türleri

3.4.1 Concept Virüsü

3.4.2 Melissa Virüsü

3.5 Makro Virüsleri Önleme

**1.Single Sign-On (SSO)**

**1.1 Tarihçe**

Tüm şirketler fonksiyonel bir şekilde ana bilgisayar merkezli (Maninframe-Centered) veri işleme ortamlarını sunucu/istemci tabanlı konfigürasyonlar ile değiştirdi.

Bu değişim sonucunda yeni ekonomiler, çok çeşitli operasyonel seçenekler ve bir dizi zorluklar meydana geldi.

Ana bilgisayar merkezli kurulumda sistem yönetimi genellikle bilgisayar sisteminin yönetimini yapanlar ile belirli bir alana sınırlıydı. Dağıtılmış işlemci/sunucu mimarisinde bu sistem komplike ve dağıtık bir yapıya dönüştü. En basit giriş yapma işlemi bile zor bir hale geldi.

Başlangıçta yaklaşık 50 yıl önce sistem tasarımcıları kullanıcıların bilgisayar sistemleri ile etkileşimini izlemenin bir yolu olduğunu farketti.

Kullanıcılar tek bir alanda olduğu için her kullanıcıya bir kullanıcı adı ve şifre verilerek ve bu kullanıcı ile ilgili işlemlerin kayıtları tutularak güvelik sağlayabiliyorlardı. Basit anlaşılır ve yönetimi kolaydı.

Bugünün ortamı coğrafya açısından daha dağıtık bir ortam. Bu ortam birden çok sistemin ve uygulamanın birbiri ile ağ üzerinden etkileşimi ile oluşuyor. Tüm bu çeşitli sistemlerin kullanıcıyı tanımlamak için farklı mekanizmaları var. Bu sistemler her birine özgü şifresi olan muhtemelen eşsiz derecede eşsiz kurallar tarafından yönetiliyor. Bu durum sık sık problemler yaratıyordu. Bunu kolaylaştırmak için tekli oturum açma kavramı tasarlandı.

Başlangıçta yaklaşık 50 yıl önce sistem tasarımcıları kullanıcıların bilgisayar sistemleri ile etkileşimini izlemenin bir yolu olduğunu farketti.

Kullanıcılar tek bir alanda olduğu için her kullanıcıya bir kullanıcı adı ve şifre verilerek ve bu kullanıcı ile ilgili işlemlerin kayıtları tutularak güvelik sağlayabiliyorlardı. Basit anlaşılır ve yönetimi kolaydı.

Bugünün ortamı coğrafya açısından daha dağıtık bir ortam. Bu ortam birden çok sistemin ve uygulamanın birbiri ile ağ üzerinden etkileşimi ile oluşuyor. Tüm bu çeşitli sistemlerin kullanıcıyı tanımlamak için farklı mekanizmaları var. Bu sistemler her birine özgü şifresi olan muhtemelen eşsiz derecede eşsiz kurallar tarafından yönetiliyor. Bu durum sık sık problemler yaratıyordu. Bunu kolaylaştırmak için tekli oturum açma kavramı tasarlandı.

**1.2 SSO Nedir ?**

Türkçe'ye "Tek Oturum Açma" olarak tercüme edilebilen SSO yani single sign-on, birbiriyle ilişkili fakat birbirinden bağımsız uygulamalar için kullanılabilen çoklu erişim kontrolüdür.

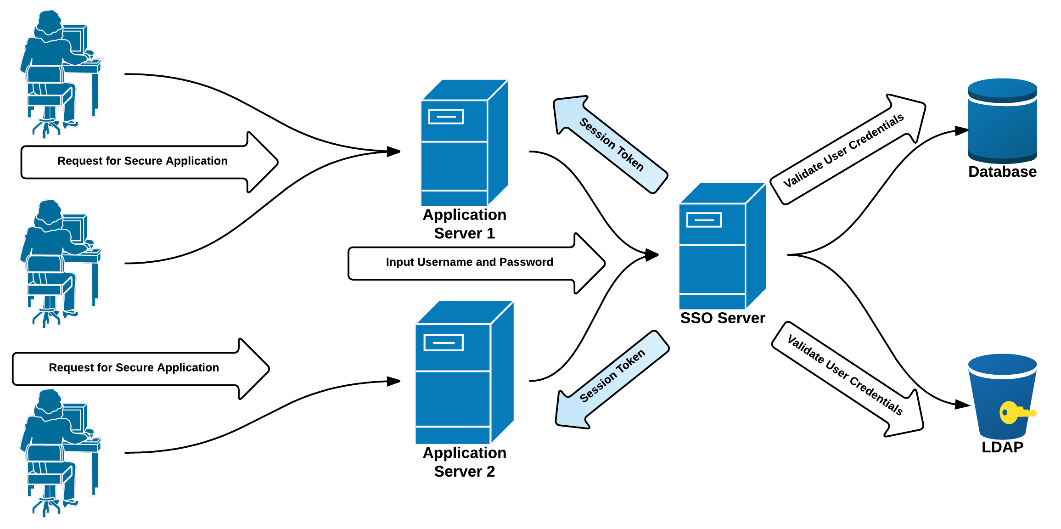
SSO tek bir kullanıcı ile oturum açarak çok sayıda uygulama/site’ye erişime olanak sağlar.

SSO’nun merkezinde girişin kontrolünü yapacak olan SSO sunucusu yer alır.

Kullanıcılar herhangi bir uygulama/site ye giriş yapmak istediklerinde bu SSO sunucusuna yönlendirilirler. Burada kullanıcılar giriş bilgileriyle sisteme giriş yaptıktan sonra bir Token oluşturulur ve SSO sunucusu kullanıcıyı bu Token ile birlikte giriş yapmış olduğu uygulamaya geri döndürür.

Kullanıcılar herhangi bir uygulama/site ye giriş yapmak istediklerinde bu SSO sunucusuna yönlendirilirler. Burada kullanıcılar giriş bilgileriyle sisteme giriş yaptıktan sonra bir Token oluşturulur ve SSO sunucusu kullanıcıyı bu Token ile birlikte giriş yapmış olduğu uygulamaya geri döndürür.

Giriş yapılan uygulama kendisine gelen Tokenı kontrol eder ve eğer Token doğru ise kullanıcı çerezler yardımıyla uygulamaya giriş hakkına erişir. Bu kullanıcı farklı bir uygulamaya istek yaptığı zaman çerez ve Token kontrolü ile oturum açma işlemi gerçekleşmiş bir şekilde uygulamayı açacaktır.



Bu sistemde uygulamaların kullanıcı bilgileri tek bir database’de birleştirilir. Bu birleşme sayesinde giriş yaparken kullanılan şifreleme algoritmaları tek bir algoritmaya iner. Bu sayede birçok farklı şifreleme algoritması kullanmak yerine tek bir algoritmayla bu sıkıntı çözülür.

Ayrıca SSO Server olarak tanımlanan uygulamaya erişim engeli konularak başka uygulamalardan veya url lerden yapılacak istekler engellenerek bir güvenlik önlemi de alınmış olur.

Bu sisteme tanımlanan uygulamalara izin verilip diğer uygulamalar engellenerek bir güvenlik katmanı oluşturulur.

**1.3 SSO Senaryoları**

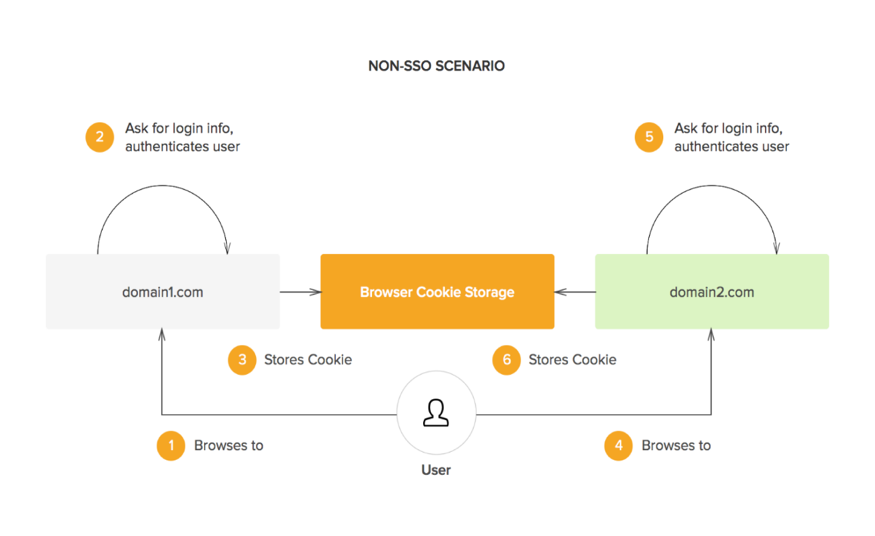
**1.3.1 SSO Uygulanmayan Login Senaryosu**

SSO uygulanmadan yapılan giriş işlemleri şu şekilde işler.

Kullanıcı “A“ uygulamasına giriş yapar ve bu uygulamaya ait olan çerez(cookie) tarayıcıda tutulur.

Kullanıcı “B” uygulamasına girmek istediğinde bu uygulamaya ait çerez olmadığı için “B” uygulaması kendi bilgilerini barındıran çerezi oluşturur ve tarayıcıya ekler.

Kullanıcı her iki sisteme aynı bilgileriyle girmesine rağmen, ikisi için de ayrı ayrı giriş yapması gerekmektedir.

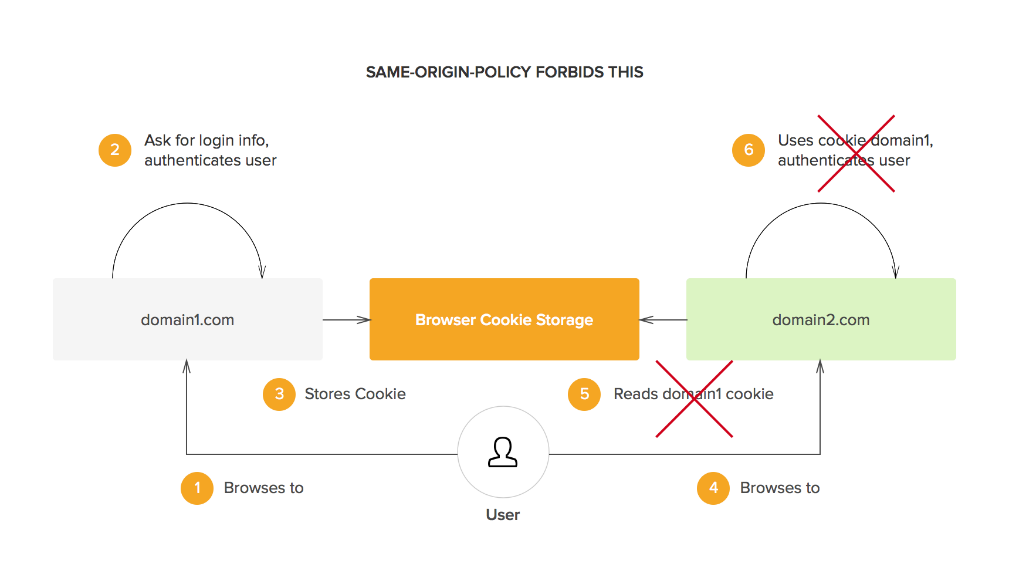


**1.3.2 Temel SSO Uygulanan Login Senaryosu**

Aslında SSO senaryosu için yapılmak istenen şey bu çerezleri farklı domainlerde paylaşabilmek. Fakat tarayıcı bu işlemi yapmamıza izin vermez.

Gerekçesi de güvenlik. “A” domaininde yaratılan bir çerez sadece bu domainden erişilebilir.

Yani “B” uygulaması “A” uygulamasına ait olan çereze erişip bu çerezde tutulan bilgileri okuyamaz. Bu tam tersi için de geçerlidir.



**1.3.3 Merkez Domainli SSO Uygulaması**

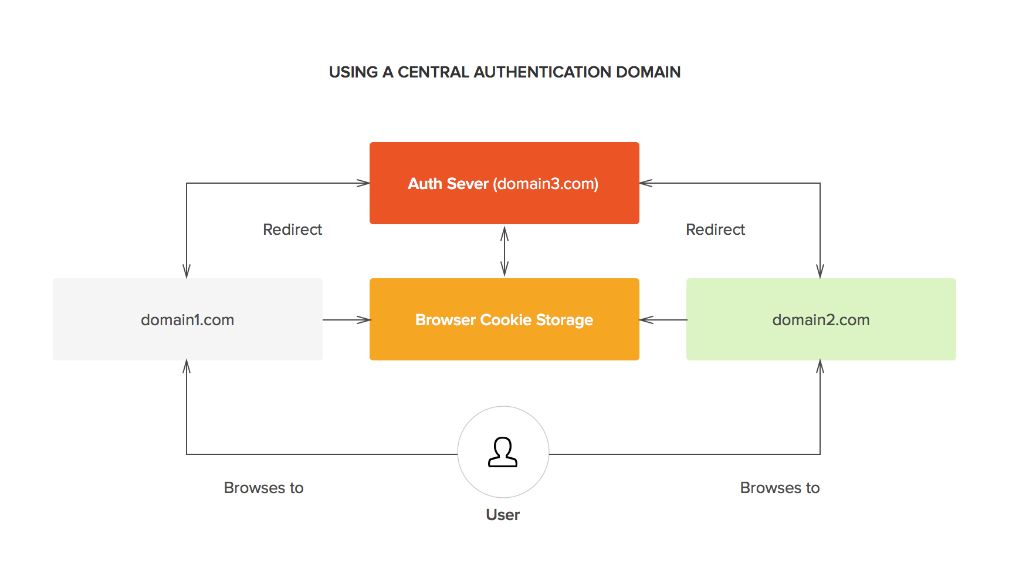
Birçok farklı SSO uygulaması ve yöntemi bulunmaktadır. Fakat etkili olan yöntemde bir adet merkez domain bulunur.

Bu merkez domain üzerinden tüm authentication işlemleri gerçekleştirilir. Oluşturulacak olan çerez bu domain üzerinde oluşturulur ve diğer domainler ile bu oluşturulan bilgi paylaşılır.

Bu paylaşılan bilgi için Token kullanılabilir.

Bu token ve giriş yapan kullanıcıyı belirleyecek bilgileri içeren bir dönüş merkez domaine istek yapan diğer domainlere gönderilebilir.

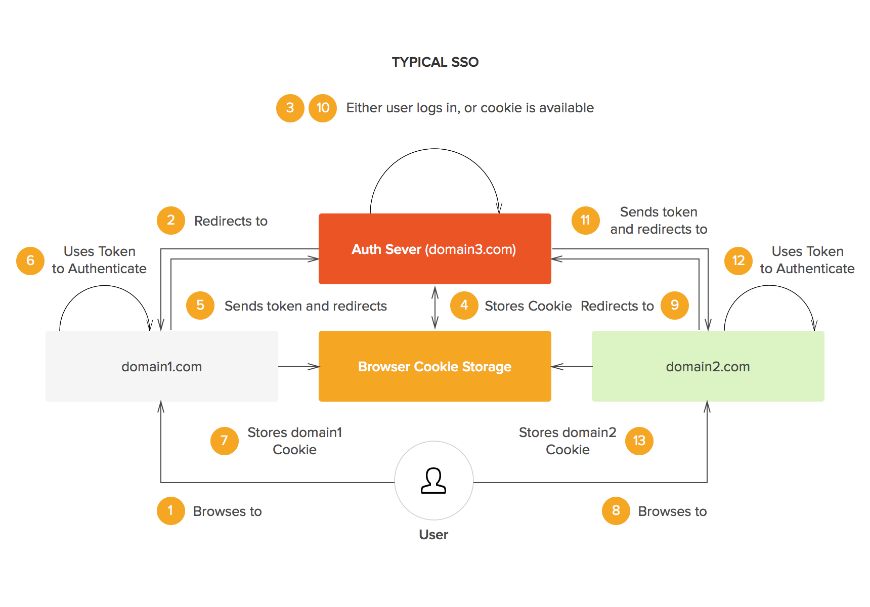
Bu Token’i alan diğer domainler bu verinin doğruluğunu sağlayıp giriş işlemini tamamlarlar.



**1.3.4 SSO Yaygın Kullanım**

Eğer kullanıcı giriş isteyen bir sayfaya gitmek istediğinde veya giriş yapmak istediğinde bu merkez domaine yönlendirilir.

Eğer kullanıcı zaten giriş yapmışsa bu domainde kayıtlı olan çerezde yer alan bilgi yani “Token” geri döndürülür.



Basit olarak anlatılmak gerekirse kullanıcı A sitesine girer.

Burada giriş olmak istediğinde M(Main) uygulamaya yönlendirilir.

Bu uygulama kendi adına kayıtlı olan çerez var mı diye kontrol eder.

Eğer yoksa A uygulamasının giriş sayfasına yönlendirir.

Bu sayfada kullanıcı verilerini girdikten sonra M uygulamasında bulunan servis sayesinde giriş bilgilerine karşılık gelen kullanıcı objesini ve bu kullanıcıya ait Token’ı gönderir.

A uygulaması bu Token’ı M uygulamasına gönderir ve M uygulaması bu Token’ı çerez olarak kaydeder ve A uygulamasına yönlendirir. A uygulaması gelen istekteki Token’ı bulur ve Token giriş yapılmış olan kullanıcı ile eşleşirse başarılı bir giriş yapılmış olur.

 Aynı kullanıcı başka bir uygulama olan B uygulamasına giriş yapmak istediğinde M uygulamasına yönlendirilir. Burada M uygulaması kayıtlı olan çerez var mı diye kontrol eder ve çerezi bulur. Bulduğu çerezin içerisinde bulunan Token’ı B uygulamasına gönderir ve B uygulaması da M de bulunan servis sayesinde bu Token’a ait olan kullanıcı çekerek giriş işlemini gerçekleştirir.

**1.4 SSO’nun Faydaları**

Farklı kullanıcı ve şifrelerden oluşabilecek şifre külfetini azaltır. Tek bir algoritma, tek bir database.

Kullanıcıların aynı bilgilerle girebileceği uygulamalara tekrar kullanıcı adı ve şifre girmesini engeller.

Giriş yapan kullanıcılar bir çatı altına toplanarak daha düzenli ve sistematik bir sistem oluşur.

Uygulamaların oturum açma işlemleri için ekstra işlemler gerçekleştirmesine gerek kalmaz. Tüm işlemler Identity Provider olarak adlandırılan uygulamada gerçekleştirilir.

**1.5 Token Nedir**

Oluşturulacak olan Token’ leri yaratma yöntemlerinden biri JWT (JSON Web Tokens) dir. JWT toplam 3 bölümden oluşur.

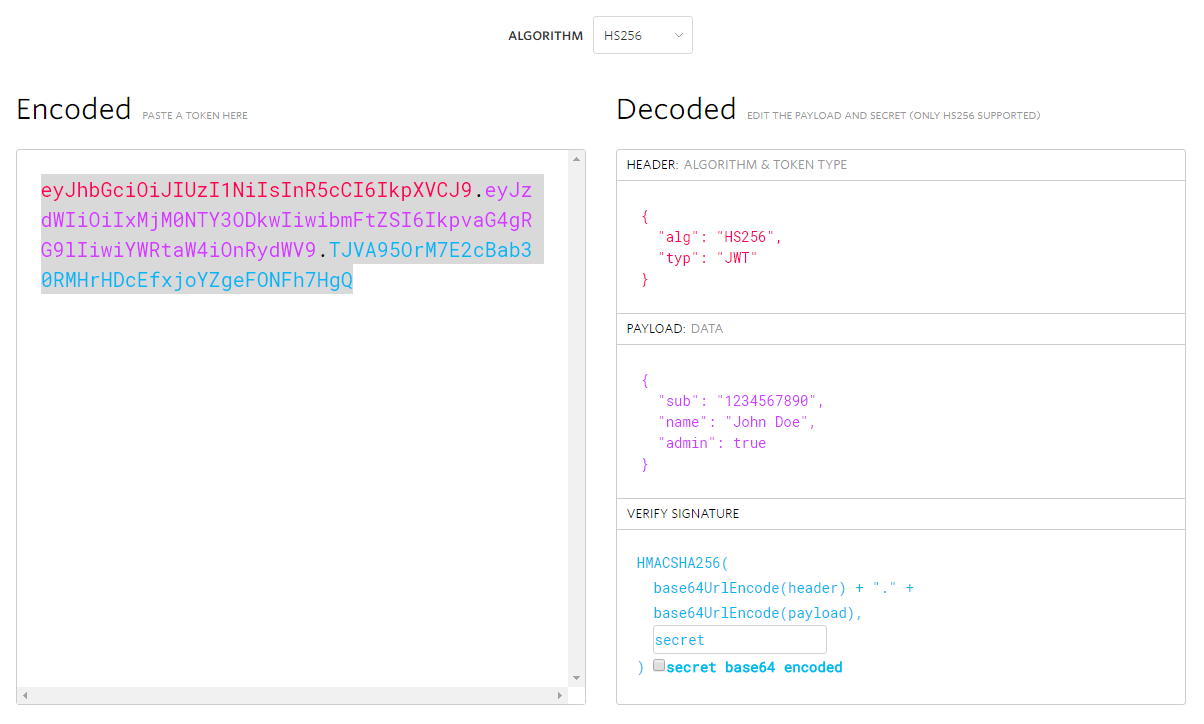
* Header
* Payload
* Signature

Header bölümü içerisinde iki adet tanımlama bulunur. Hangi hash algoritması kullanılacak ve oluşturulacak olan Token’in tipi.

 Payload bölümü claims leri içeren bölümdür. Bu bölümü tanımlamak zorunlu değil fakat istenilen veriler bu bölümde saklanmaktadır. Claimsler genelde kullanıcı bilgisi içeren bölümdür.

Signature bölümü imzanın oluşturulduğu yerdir.

 Bu üç bölüm “.” karakteriyle birleşerek Token’i oluşturur.



**2. Relational Data Base Access Controls Using SQL**

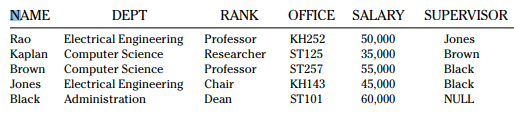
Günümüzde sıkça kullandığımız bir çok veritabanı sistemi mimari olarak ilişkisel veritabanı modelini kullanmaktadır.

İlişkisel Veritabanı Yönetim Sistemi (Relational Database Management System kısaca RDBMS) olarak adlandırılan bu sistemler veritabanının sağlıklı şekilde çalışması ,veri tutarlılığı ve veri bütünlüğü gibi konularda bize oldukça büyük faydalar sağlar.

SQL Örneği

Örneğin, aşağıdakiler EMPLOYEE ilişkisi için bir ilişki şemasıdır

EMPLOYEE (NAME, DEPT, RANK, OFFICE, SALARY, SUPERVISOR)



**2.1 Primary Key**

İlişkilerimizi tabloların anahtar sütunları arasında bağlantılar kurarak yaparız. Peki nedir bu anahtar sütunlar?

Tablomuzda tuttuğumuz her bir kayıta ait verilere ulaşmak için bir yada birden fazla sütunu seçerek o sütun üzerinden verilerimize ulaşırız.Yani belirlediğimiz sütun bizim kayıta ulaşmamız için bir nevi aracı görevi görür.



Yukarıdaki Tablodan yola çıkalım. Murat’ın Bilgilerine ulaşmak istediğimizde veritabanımız dan adı “Murat” olan kişinin bilgilerini getirmesini isteriz. Veritabanı Adı sütununa giderek buradaki veriler içerisinde “Murat” verisini arar. Arama sonucu olumlu ise bulduğu sonucu bize gösterir. Bu şekilde yaptığımız aramalar bazı sorunları da beraberinde getirmektedir.

Aynı isimde yada soy isimde farklı çalışanlarımız olabilir. Böyle bir durumda adı “Murat” olarak kayıtlı tüm kişilerin bilgileri bize gösterilir. Bu sorunu önlemek için her bir kayıtı eşsiz ve benzersiz şekilde tanımlayan (Yani aynı ve boş (NULL) veri içermeyen) sütun belirler. İşlemlerimizi o sütun üzerinden yaparız.

Yukarıdaki tablomuzda “Personel No” sütunu bu görevi görmektedir.

“Personel No = 3” olan kayıtı görmek istediğimizde karşımıza Yeşim Güler’in kayıtı çıkacaktır. Bu şekilde her bir kayıtı eşsiz ve benzersiz şekilde tanımlayan sütunlar Anahtar Sütunlardır.

**2.2 SQL İfadeleri**

**2.2.1 Create**

Employee örneği için SQL’de ilişki şemasını tanımlayalım.

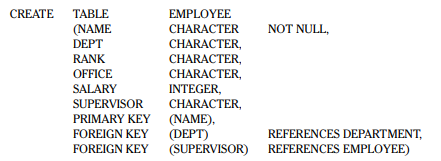
Bu ifade, altı sütunlu Employee adlı bir tablo oluşturur. AD, DEPT, RANK, OFİS ve SUPERVISOR sütunları karakter içeriyor.

SALARY sütunu, değer olarak (belirtilmemiş uzunlukta)tamsayı değerleri.

NAME birincil anahtardır.

DEPT, DEPARTMENT tablosunun birincil anahtarına başvuran yabancı bir anahtardır.

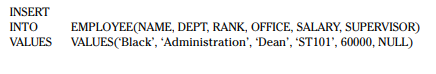
SUPERVISOR yabancı bir anahtardır.



**2.2.2 Insert ve Delete**

Oluşturduğumuz Employee tablosu başlangıçta boştur. SQL Insert komutu kullanılarak içine tuple’lar eklenir.

Eklenen Tuple’lar Delete komutu kullanılarak silinebilir.



**2.2.3 Select**

SQL’de verilerin alınması Select komutu ile gerçekleştirilir. Örneğin bilgisayar bilimleri bölümündeki çalışanlar için NAME, SALARU ve SUPERVISOR verileri aşağıdaki komutla getirilir.



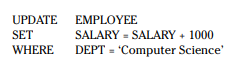
Getirilen veriler;



**2.2.4 Update**

Update ifadesi varolan bir verinin değiştirilmesi için kullanılır.

Örneğin aşağıdaki ifade Bilgisayar Bilimleri bölümündeki tüm çalışanlara 1000 dolar zam verir.



Bu ifade Employee tablosundan istenilen Tuple’ı çekerek Salary sütunundaki verilerin 1000 dolar artmasını sağladı.

**2.3 SQL İzinleri**

SQL Server, verilere, meta verilere ve diğer SQL Server kaynaklarına erişimi kontrol etmek için bir dizi ayar sağlar. SQL Server güvenliği için yeni olan veya güvenliği düzenli olarak yönetmeyenler için, bu ayarların yapılandırılması karmaşık ve kafa karıştırıcı bir işlem gibi görünebilir.

Ayarlar doğru yapılırsa sistem güvenli bir şekilde işler fakat yanlış yapılırsa, kullanıcılar ihtiyaç duydukları verilere erişemeyebilir veya daha da kötüsü yetkisiz kullanıcılar hassas bilgilere erişebilir veya kötü niyetli saldırılar yapabilir .

SQL Server'ı etkin bir şekilde korumak için, onaylı kullanıcılara, belirli SQL Server kaynaklarına erişebilmelerini , bu kaynaklardan veya diğer kaynakları etkilemeden, üç önemli bileşen türünün kullanımını içeren bir işlem sunabilmemiz gerekir.

**2.4 SQL İzin Bileşenleri**

**2.4.1 İlkeler**

SQL Server kaynaklarına erişmek için kimliği doğrulanabilen varlıklardır.

Örneğin, Windows giriş bilgileriniz bir SQL Server veritabanına bağlanmanıza izin veren bir ilke olarak yapılandırılabilir.

SQL Server, üç prensip türünü destekler: girişler, kullanıcılar ve roller. Girişler sunucu düzeyinde, kullanıcılar veritabanı düzeyinde bulunur ve roller her iki düzeyde de olabilir.

**2.4.2 Sağlanabilirler**

Bir ilke tarafından erişilebilen SQL Server kaynaklarıdır.

Güvenceler, sunucu düzeyinde (örneğin kullanılabilirlik grupları), veritabanı düzeyinde (örneğin tam metin kataloğu) veya şema düzeyinde (ör. Tablo veya işlev) korumaya çalıştığınız gerçek kaynaklardır.

**2.4.3 İzinler**

Belirli bir ilkeye güvenli bir şekilde verilen erişim türleridir.

Örneğin, bir Windows oturum açma bilgisine (anapara) belirli bir veritabanı şemasında (güvenli) verileri görüntüleme (izin) yeteneği verilebilir.

**2.5 İzinler**

SQL Server, üç farklı kapsamda güvenlik altına alınabilir:

Sunucu kapsamındaki güvenceler, girişler, sunucu rolleri, kullanılabilirlik grupları, bitiş noktaları ve bir bütün olarak veritabanları gibi kaynakları içerir.

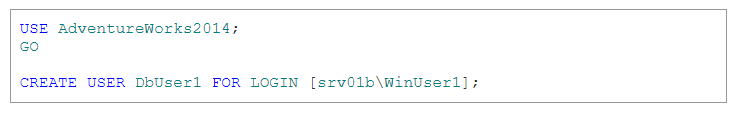
Veritabanı kapsamındaki güvenceler, kullanıcılar, veritabanı rolleri, sertifikalar, asimetrik anahtarlar, derlemeler, tam metin kataloglar ve bir bütün olarak şemalar gibi kaynakları içerir.

Şema kapsamındaki güvenceler, tablolar, görünümler, türler, işlevler, prosedürler ve XML şema koleksiyonları gibi kaynakları içerir.

Veritabanı kullanıcı prensipleri

Bir oturum açma kapsamı bir bütün olarak veritabanı motoruna uygulanır, ancak bireysel veritabanı bileşenlerine erişim sağlamaz.

Bu girişlerle ilişkili kullanıcıların veritabanı kaynaklarına erişebilmesi için, hedef veritabanlarında kullanıcılar oluşturmalıdır.



**2.6 Sunucu ve Veritabanı Rolleri**

Bir oturum açma veya kullanıcı sorumlusu oluşturduktan sonra, sunucu veya veritabanı düzeyinde erişimi kontrol etmek için izinler yapılandırılabilir.

SQL Server, üç tür rolü destekler: sunucu, veritabanı ve uygulama.

Sunucu rolleri girişlerle aynı kapsamı paylaşır; bu, sunucu düzeyinde çalıştıkları ve bir bütün olarak veritabanı motoruna ait oldukları anlamına gelir. Sonuç olarak, rollere yalnızca sunucu düzeyinde ilkeler ekleyebilir ve rolleri izinleri olan izinlerle veritabanı düzeyinde güvence altına alınabilecek şekilde yapılandırabililiz.

SQL Server, her biri değiştirilemeyecek izinlerle önceden yapılandırılmış bir dizi sabit sunucu rolü sağlar. Sabit roller, bu rollere atanan yöneticilerin belirli görevleri yerine getirmelerine izin verir.

Örneğin, sysadmin sabit sunucu rolünün üyeleri, sunucu üzerinde herhangi bir eylem gerçekleştirebilir, ancak serveradmin rolünün üyeleri yalnızca sunucu genelindeki yapılandırma seçeneklerini değiştirebilir ve sunucuyu kapatabilir.

Sabit sunucu rollerinin listesini görüntülemek için , aşağıdaki örnekte gösterildiği gibi sp\_helpsrvrole system komutunu çalıştırabiliriz.



Bir sunucu rolü oluşturmak, başka bir ilke türü oluşturmak kadar kolaydır. Örneğin , aşağıdaki örnekte gösterildiği gibi, rol adından başka bir şey belirtmeyen bir CREATE SERVER ROLE ifadesi tanımlayabiliriz.



Bu durumda, SrvRole1 bir sunucu rolü adı yaratıyoruz . Ancak, rol yalnızca SQL Server örneğinde sunucu düzeyinde bir nesne olarak, hiçbir izni olmadan var olur. Buradan, role dahil etmek istediğimiz oturum açma ilkelerini eklemeli ve sonunda rolü gerekli izinlerle yapılandırmalıyız.

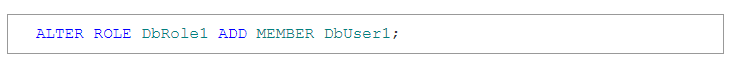
SrvRole1 sunucu rolüne bir ilke eklemek için , aşağıdaki örnekte gösterildiği gibi ALTER SERVER ROLE deyimini kullanabiliriz



Veritabanı rolü oluşturmak, sunucu rolü oluşturmak kadar kolaydır. Daha önce olduğu gibi, SSMS kullanabilir veya bir T-SQL ifadesi yayınlayabiliriz, bu durumda CREATE ROLE ifadesi örneğin, aşağıdaki örnek DbRole1 veritabanı rolü oluşturur.



DbRole1 veritabanı rolüne bir ilke eklemek için , aşağıdaki örnekte gösterildiği gibi ALTER ROLE deyimini kullanabiliriz



**2.7 SQL Server İzinleri**

* GRANT: Bir yöneticiye güvenli bir şekilde izinler verir.
* DENY: Bir yöneticiye güvenli bir şekilde izin verilmeyen izinleri reddeder. Bu önemli olabilir çünkü yöneticinin kendisine verilen diğer izinleri devralmasını engeller.
* REVOKE: Bir yöneticiye güvenli bir şekilde önceden verilmiş veya reddedilen izinleri kaldırır.
* Bu ifadelerin her biri, izin yapısının karmaşıklığının yanı sıra oldukça karmaşık bir sözdizimini de destekler.

DbUser1 veritabanı kullanıcısını içeren DbRole1 veritabanı rolüne uspUpdateEmployeeHireInfo saklı yordamına EXECUTE izni vermek için bir GRANT deyimi kullanalım

GRANT EXECUTE ON OBJECT::HumanResources.uspUpdateEmployeeHireInfo TO DbRole1;

SQL Server kaynaklarına düzenleme erişimi zaman alır ve dikkatle değerlendirilir. Sistem ne kadar karmaşıksa, o kadar fazla iş içerir.

**3.Makro Virüsler**

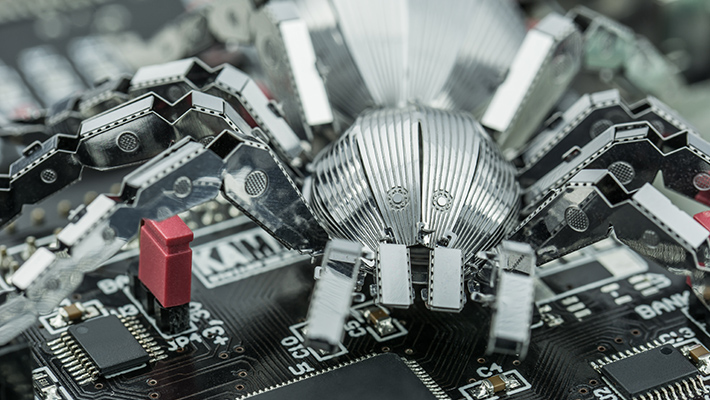
**3.1 Makro Virüs Nedir ?**

Makro virüsler kendi kodlarını belgeler, elektronik tablolar ve diğer veri dosyalarıyla ilişkili makrolara ekler.

Concept adlı ilk makro virüsü Temmuz 1995'te ortaya çıktı ve ardından makro virüsleri (çoğunlukla Word belgelerine bulaşan) yeni yüzyılın başlangıcına kadar baskın virüs türü haline geldi.

O dönemde Microsoft, Office'te (Office 2000'den sonraki sürümler) makroları varsayılan olarak devre dışı bıraktı.

O zamandan bu yana siber suçlular kendi virüslü makrolarının çalışması için kurbanları makroları etkinleştirmeleri için kandırmaya çalışmak zorunda kaldı.



**3.2 Makro Virüsler Nasıl Yayılır ?**

Makro virüsleri sıklıkla belgelere yerleştirilmiş olarak bulunur veya kelime işleme programlarına kötü amaçlı kod şeklinde eklenir.

E-postalara eklenmiş belgelerden gelebilir veya kod, reklam şeritlerindeki veya URL'lerdeki "kimlik avı" bağlantıları tıklatıldıktan sonra indirilebilir.

Virüs bulaşan bir makro çalıştırılıp bir dizi komut gerçekleştirdiği zamana kadar faaliyete geçmediklerinden algılanmaları zordur.

Bir makro virüsü, yararlı gibi görünebilmesi ve kullanıcıların kötü etkilerini hemen fark edememesi açısından bir Truva atı virüsüne benzer. Ancak, Trojanlardan farklı olarak makro virüsleri kendilerini çoğaltabilir ve diğer bilgisayarlara bulaşabilir.



**3.3 Riskler**

Makro virüslerinin ana riski hızla yayılabilmeleridir.

Virüs bulaşmış bir makro çalıştırıldıktan sonra, kullanıcının bilgisayarındaki diğer belgelere virüs bulaşır.

Bu virüslerin bazıları metin belgelerinde eksik veya eklenmiş sözcükler gibi anormalliklere neden olurken, bazıları da e-posta hesaplarına erişip virüs bulaşmış dosyaların kopyalarını bir kullanıcının kişi listesindeki herkese gönderir; daha sonra, güvenilen bir kaynaktan geldiği için bu dosyalar açılır ve dosyalara erişilir.

Bu virüsler, depolanan verileri silmek veya bu verilerin güvenliklerini ihlal etmek üzere tasarlanmış da olabilir.

Buna ek olarak, makro virüslerinin platformlar arası çalıştığına dikkat edilmesi de önemlidir; aynı kodu kullanarak hem Windows hem de Mac bilgisayarlara bulaşabilirler.

Makroları kullanan programlar ana bilgisayar olarak çalışabilir ve e-posta yoluyla gönderilen, diskte veya USB sürücüde saklanan virüs bulaşmış programın kopyası virüs içerir.

 Bu virüsleri kaldırmak için kullanıcılar, özel makro virüs algılama ve kaldırma araçları sağlayan güvenlik yazılımları kullanmalıdır. Düzenli taramalar, virüs bulaşmış belgeleri temizler ve yeni bilgisayar virüslerinin indirilmemesini sağlar.

**3.4 Makro Virüs Türleri**

Makro virüsleri birkaç farklı biçimde görülebilir. Bazıları bunların 1990'ların sonundan kalma olduğunu düşünse de son yıllarda yeniden dirilen bu virüs, kullanıcıların daha uyanık olmalarını gerektiriyor.

**3.4.1 Concept Virüsü**

Temmuz 1995'te ortaya çıkan ve Microsoft Word'ü hedefleyen Concept, ilk makro virüsüydü. Ardından makro virüsleri baskın virüs türü haline geldi.

**3.4.2 Melissa Virüsü**

Melissa, e-posta solucanı özelliği olan ilk makro virüsü olarak tarihe geçti. 26 Mart 1999'da e-posta yoluyla yayılmaya başladı ve saatler içinde on binlerce bilgisayara virüs bulaştırdı. İnternet tarihinin en ciddi salgınlarından biriydi.

**3.5 Makro Virüsleri Önleme**

Makro virüsler genellikle İnternet'te e-postayla paylaşılan uygulama dosyalarına, özellikle de kimlik avı e-postalarına yayıldığı için, makro virüs savunmaları, gelen e-posta eklerini taramak, kullanıcıların şüpheli dosyaları açmasını engellemek ve belgelerin tümünde makroların çalışmasını engellemek için doğru yoldur.



Makro virüslerin yayılmasını önlemek için bazı teknikler şunlardır:

 Bir spam filtresi kullanın. Kimlik avı yapan e-posta mesajları kullanıcıları gelen kutularına ne kadar az girerse, bilgisayarlarına kötü amaçlı yazılım bulaştırabilir.

Güçlü bir virüsten koruma programı kullanın. Antivirüs yazılımı, şüpheli dosyaları indirmeye veya zararlı linkleri açmaya çalıştığında kullanıcıları uyarır.

Bilgisayarların mevcut yazılım sürümlerini çalıştırdığından ve tüm güvenlik düzeltme eklerinin yüklendiğinden emin olun.

Bilinmeyen gönderenlerden ekleri açmayın.

Ekleri, şüpheli e-posta iletilerinde, bilinen gönderenlerden gelmiş gibi görmelerine rağmen açmayın.

Microsoft Word ve Excel'de makro güvenliği işlevini etkinleştirin ve makroları etkinleştirme konusunda çok dikkatli olun.

Makro komut dosyalarını tamamen devre dışı bırakın.

**3.6 Bir makro virüsünü kaldırma**

Yayılmasını önlemek için bir makro virüsüyle enfekte olan tüm dosyaları kaldırmak önemlidir.

 Makro kötü amaçlı yazılımları temizlemenin ilk adımı, virüslü bilgisayarı Güvenli Mod'da yeniden başlatmaktır .

 Tüm geçici dosyaları silmek, virüs taramasını hızlandırmaya, aynı zamanda disk alanını boşaltmaya ve kötü amaçlı yazılım bulaşmış geçici dosyaları silmeye yardımcı olacaktır.

