LOVE LETTER

**Analitik SQL Nedir ?**

SQL bir “DataBase” yönetim yazılımıdır. Aslında donanımlar seviyesine inecek olursa donanımı önemli kılan “veri” denen ifadeyi Programlarımızla buluşturmaya ve buna göre işlemeye yarayan yazılımdır. Analitik SQL’de aslında Bilgisayar Bilimleri yıllarca geliştikçe veri miktarı artmıştır. Veriye olan açlık doyum noktasına gelmiş ve verilerin filtrelenmesi verinin faydalı bilgi haline alması için bir ihtiyaç doğmuştur. Analitik SQL burada ortaya çıkmış ve 2012’de hayatımıza girmiştir.

**Analitik SQL Nerelerde Kullanılır ?**

Sirketler, analitik SQL sorgulari kullanarak işletme performansini degerlendirebilir, trendleri analiz edebilir ve karar destek sistemleri için veri sağlayabilir.

Örneğin , bir e ticaret şirketi , satış verilerini analiz ederek en çok satılan ürünleri belirleyebilir, pazarlama analitiği yapabilir.

Bir banka, müşteri hareketlerini analiz ederek kredi

riskini degerlendirebilir ve müşteri terkini önlemek için önlemler alınabilir.

Bir şirket gelir tablosu ve tablosunu analiz ederek gelir ve gider trendlerini çalıştırır, mali rasyoları hesaplayabilir ve bütçe tahminleri yapabilir.

Bir lojistik şirketi, nakliye hizmetlerini analiz ederek rotalama optimizasyonunu yapabilir, envanter yönetimini optimize edebilir ve teslimat yolunu iyileştirebilir.

Bir sigorta şirketi, polis teşkilatını analiz ederek potansiyel dolandırıcılık vakalarını işletmer ve müşteri profillerini tanımlayabilir.

Büyük veri kümelerindeki desenleri, ilişkileri ve trendleri kullanmak için analitik SQL sorguları kullanılabilir.

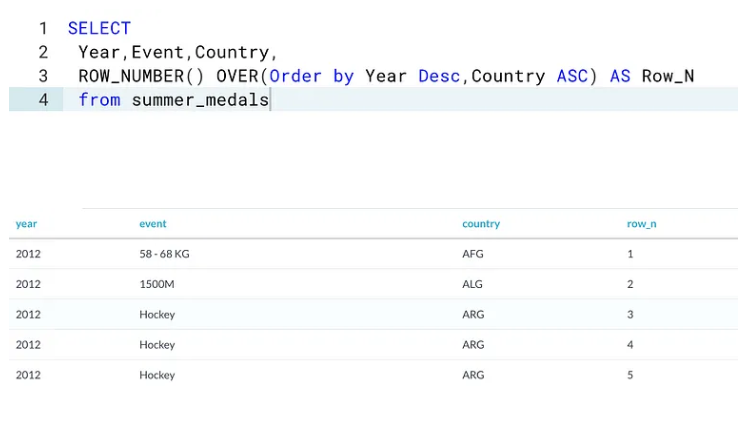
Analitik SQL, karmaşık koruyucular ve kullandıkları modelleri kullanmak için kullanılabilir. Örneğin, tahminleme, optimizasyon, simülasyon ve mekanizma analizi gibi işlemleri gerçekleştirebilir.

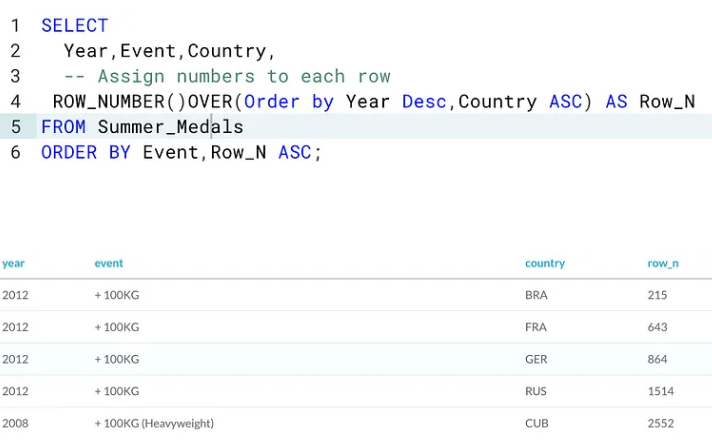
Hadoop (Açık kaynak kodlu bir kütüphane) gibi büyük veri platformları üzerinde çalışan veri analitiği için analitik SQL kullanılabilir.

Sorgu Örnekleri (her grup üyesine 1 sorgu örneği olacak şekilde)

**OVER()**

Over fonksiyonu daha çok ROW\_NUMBER ORDER BY fonksiyonları gibileriyle birlikte kullanılır. Belirli bir veri grubunu işaretlemek için eşik değeri belirlemeyi kolaylaştıran bir fonksiyondur. Mesela elimizde “yıl,etkinlik, bölge verileri ve row” şeklinde bir tablomuz var. Bu tablo üzerinde eşik değeri kullanarak ve kullanmadan analiz etmeye kalkarsak sonuçlar şu şekilde olmaktadır.



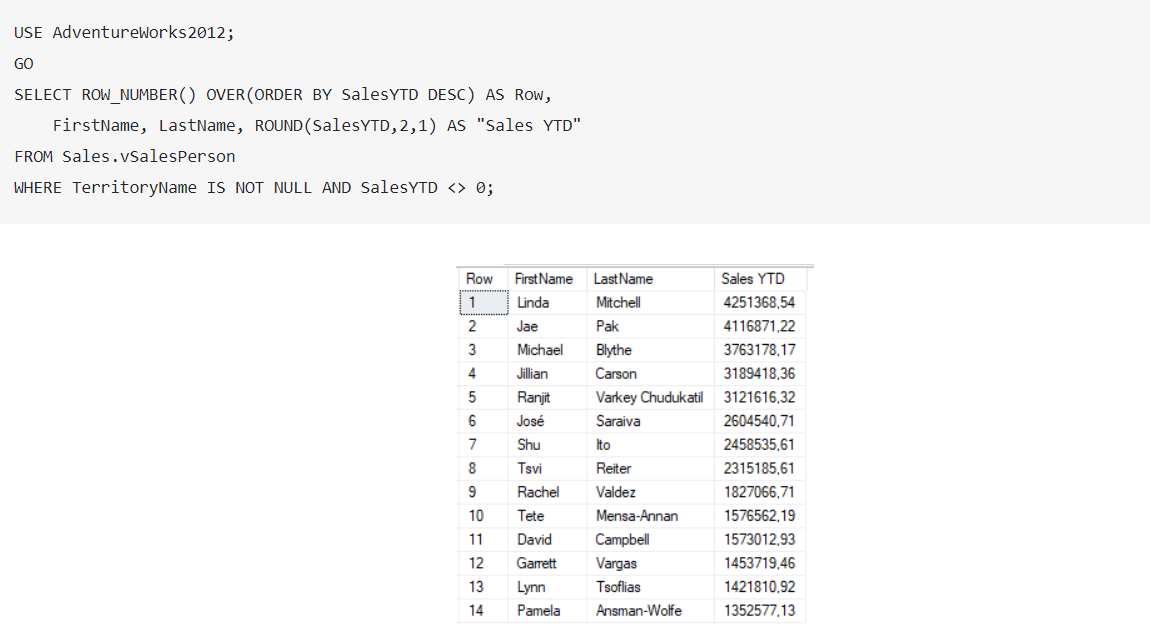


(Yukarıda yapmak istenilen örnek yıllara göre azalan ülke ismine göre artan bir sıralama hedeflenmiştir.)

**ROW\_NUMBER() ve ORDER BY()**

Tablodaki güvenlik sıralamasını belirleme ve her satıra numara atamak için kullanılırlar. ORDER BY() sorgu sonucu dönen verilerin sıra düzenini belirler.(Artan azalan gibi )

ROW NUMBER() ise belirli bir sıralama düzenine göre her satıra bir numara atar.Örnek verecek olursak satış temsilcilerinin yıla göre satış miktarlarını sorgulama işlemini yapabiliriz.



Burada “ROW\_NUMBER() OVER(ORDER BY SalesYTD DESC)” kısmı ile satış miktarlarını azalan sırada sıralayıp numaralandırmaktadır.

**RANK() - DENSE\_RANK()**

RANK() fonksiyonu ile veri setlerindeki tekrar eden satırlara aynı sıralama değeri döndürülür ve bu sayede benzersiz değerler tespit edilir. Eğer öğeler eşit sıralama değerlerine sahip olur ise bir sonraki sıralama değeri atlanır.

DENSE\_RANK() fonksiyonu ise RANK() fonksiyonundan farklı olarak numara atama işlemini yapmadan numaralam işlemine kaldığı yerden devam eder.

Burada anlatılanlara bir örnek ile anlatmak gerekirse;

**SELECT** CustomerID, COUNT(\*) **AS** "Satış Sayısı",

**RANK()** OVER(**ORDER** **BY** COUNT(\*) **DESC**) **AS** "Rank ile Sıralama",

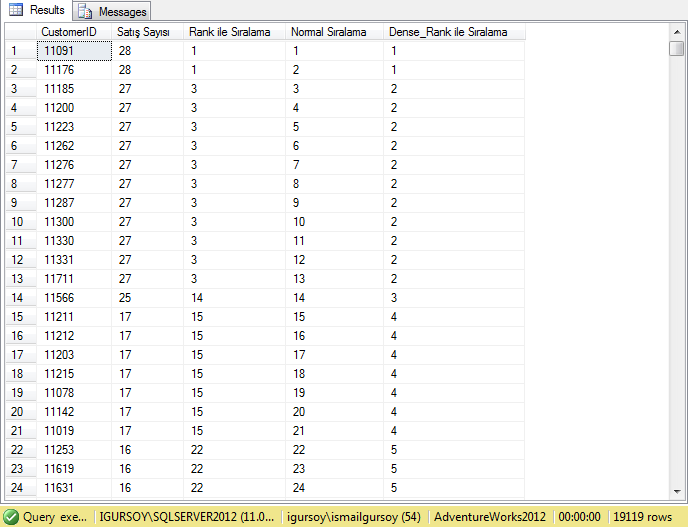
ROW\_NUMBER() OVER(**ORDER** **BY** COUNT(\*) **DESC**) **AS** "Normal Sıralama",

**DENSE\_RANK()** OVER(**ORDER** **BY** COUNT(\*) **DESC**) **AS** "Dense\_Rank ile Sıralama"

**FROM** Sales.SalesOrderHeader

**GROUP** **BY** CustomerID

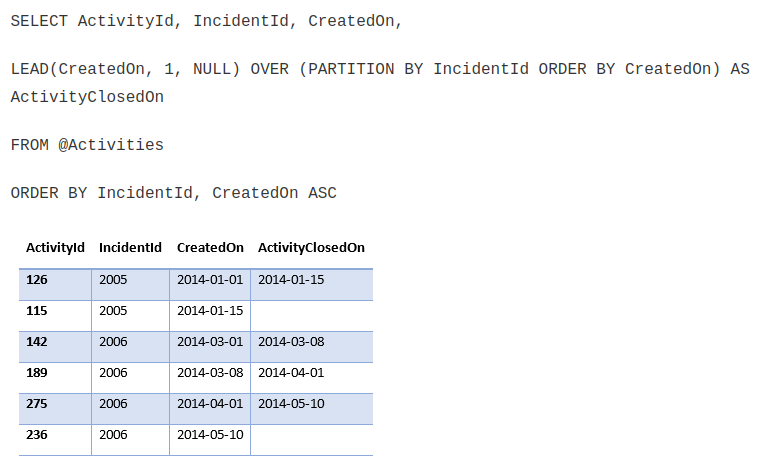
**ORDER** **BY** COUNT(\*) **DESC**;



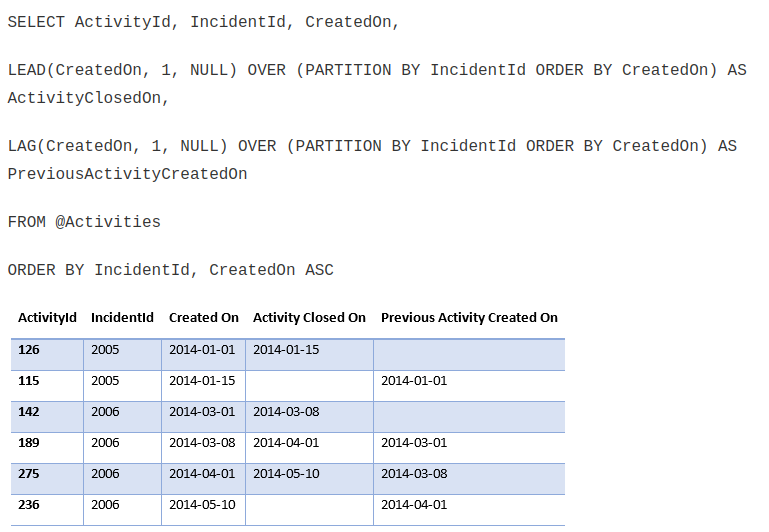
**LEAD() LAG()**

LAG ile LEAD fonksiyonu aslında bize aynı formatta veri içinde arama döndürür. Ama verinin farklı şekilde yorumlanmasında yardımcı olur. LAG fonksiyonu veri seti içerisindeki hedef veriden bir önceki veriyi hedef alarak işlem yapmamızı sağlarken LEAD fonksiyonu hedef veri’den sonraki herhangi bir veri’yi hedef alarak bize analiz yapma imkanı sunar. Metin olarak örnekleyecek olursak, bir robotu x tane odalı alanda hareket sayısı, oda metrekaresini tutan veriye sahibiz, bu metrekare alanda birim başına düşen hareket sayısının en fazla olduğu değeri bulmak istiyoruz. Burada yapmamız gerekenler yeni tablo açıp hesaplamak yerine robotun hareket sayısını OVER fonksiyonu ile belirli eşik değeri belirleyip oradan dönen sonucu odaların metrekaresine göre sıralayıp hareket sayısını LAG fonksiyonu ile önceki satır ile karşılaştırmasını sağlayarak (hatalı işlem) oranı hakkında analiz yapabiliriz. Veya LEAD fonksiyonu ile bir sonraki satıra göre Doğru işlem oranı hakkında analiz yapabiliriz.

Daha başka bir örnekle belirtecek olursak eğer;



(LEAD FONKSİYONU)

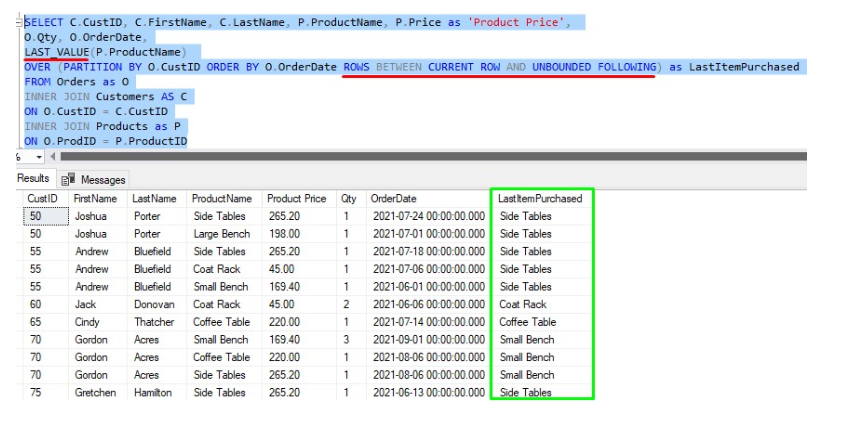


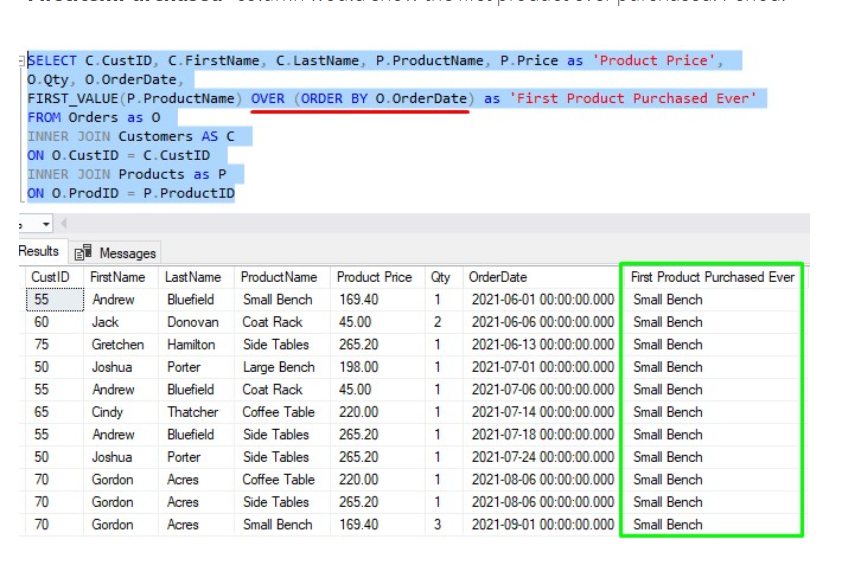
(LAG FONKSİYONU)

**FIRST\_VALUE() & LAST\_VALUE()**

LAST\_VALUE() işlemi , veri kümesi içinde bir sütundaki son değer geçişi sağlar. Yani son değeri yürütür.Bu işlev belirli bir sıralama düzenine (örneğin , bir zaman sırası veya sıra numarası ) dayalı olarak son değeri yürütür. FIRST\_VALUE() ise sütundaki ilk değeri yürütür.

Sütundaki ilk değer geçişini sağlar.





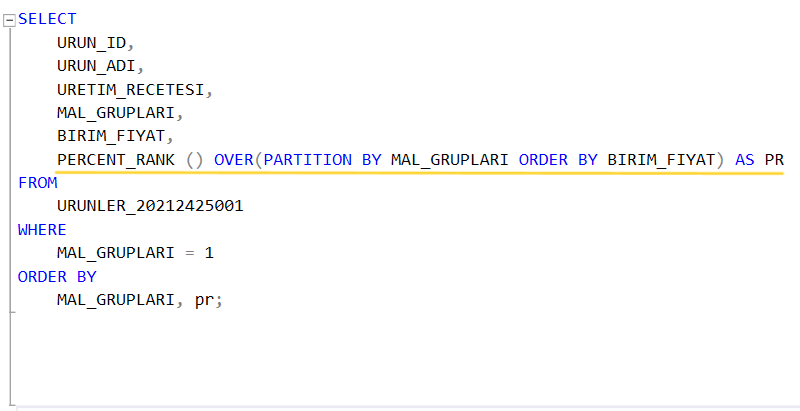
**PERCENT\_RANK**

Verileri yüzdelik olarak sıralamak için, genelde matematiksel işlemler yapmak üzere ihtiyaç duyulan fonksiyondur. 0’dan başlayıp 1’e doğru giden değerler üzerinden dağılım yapar. İlk satır her zaman 0, son satır her zaman 1 kabul edilir.

Her satır için uygulanan formül:

(satır sırası - 1) / (toplam satır sayısı - 1)

Aşağıda yer alan kod bloğunda; mal grupları 1 olan ürünlerin birim fiyatına göre küçükten büyüğe listelenmesi ve PR isimli sütunda percent\_rank değerinin tutulması beklenmiştir.



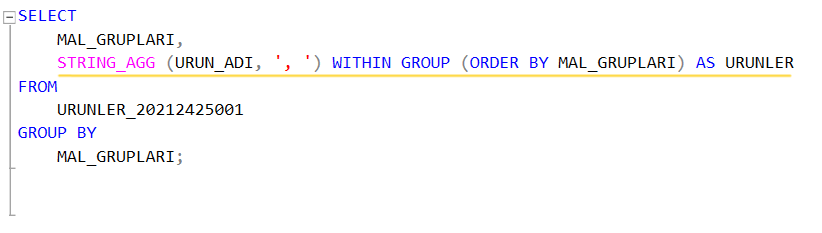
Kod bloğunun çıktısı aşağıdaki şekilde olacaktır.



**STRINGAGG() - LISTAGG()**

2 veya daha üst derece normalizasyonlarda Foreign Key yapısının kullanılmasıyla; belirli bir gruba veya kategoriye ait veriler tek satırda gösterilmek istenebilir. Bu işlemi yapabilmek için, kullanılan SQL sunucusuna bağlı olarak “stringagg” veya “listagg” fonksiyonları kullanılmalıdır.

Aşağıda yer alan kod bloğunda, belirli mal gruplarına göre kategorize edilmiş ürünler arasından hangi ürünün hangi mal grubunda yer aldığını tek satırda görebilmek istenmiştir.



Kod bloğunun çıktısı aşağıdaki gibi olacaktır.

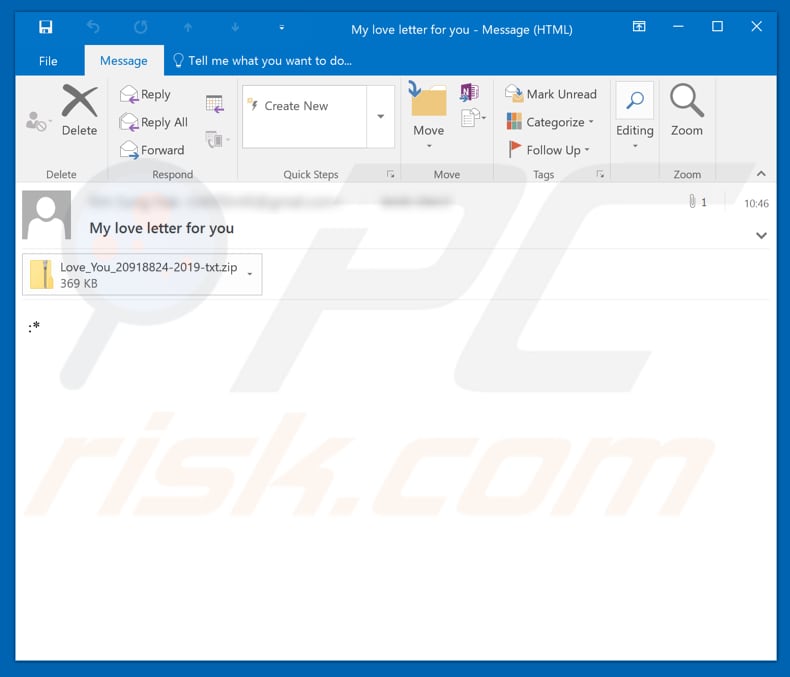


BONUS:

**LoveLetter nedir ?**

"**Love Letter Email Virus**", sahtekarların (siber suçluların) sahte e-postalar aracılığıyla alıcıları kandırmak için kullandığı bir spam kampanyasıdır. Bu sahte e-postaları alan kişileri, ekleri açmaları ve içerdikleri kötü amaçlı programları yüklemeleri konusunda kandırmaya çalışırlar.

Sunulan ek, GandCrab 5.0.4 adlı yüksek riskli bir fidye yazılımı türü olan ve XMRig Miner adlı kripto para madencisini indirip yükleyen bir JavaScript dosyası içermektedir. Ekli dosyayı açmaktan kaçınmanızı ve "Love Letter Email Virus" e-postasını görmezden gelmenizi kesinlikle öneririz.



Ek, "Love\_You\_20918824-2019-txt.zip" adında bir ZIP arşiv dosyasıdır ve içinde "Love\_You\_20918824-2019-txt.js" adlı JavaScript dosyasını içerir. Bu dosyayı çalıştırmak, yukarıda bahsedilen GandCrab 5.0.4 virüsünü ve XMRig Miner kripto para madencisini indirip yüklemenize neden olur.

GandCrab 5.0.4, verileri şifreleyen ve fidye taleplerinde bulunan bir fidye yazılımı türüdür. Bu nedenle, kurbanların dosyalara erişimini engeller ve verilerin şifresinin çözülmesi için bir şifre çözme aracı veya anahtarın satın alınması yönünde bir fidye talebinde bulunur.

Bu özel fidye yazılımı türü enfeksiyonu hakkında daha fazla bilgiye, yukarıda verilen bağlantıyı tıklayarak ulaşabilirsiniz. XMRig, enfekte bir bilgisayarın sistem kaynaklarını kullanarak Monero madenciliği yapan bir kripto para madencisidir.

Ayrıca, "Love\_You\_20918824-2019-txt.js" JavaScript dosyasını çalıştırmak, Görev Yöneticisi'nde çalışan çeşitli rastgele ve şüpheli işlemlere (örneğin "wincfg32scv.exe", "1119713827.exe", "2987227227.exe" vb.) yol açar.

Özetlemek gerekirse, "Love Letter Email Virus" içinde sunulan arşiv dosyasını çıkarmak ve içerilen JavaScript dosyasını çalıştırmak, dosyalara erişiminizin engellenmesine (veri şifreleme) ve bir kripto para madencisinin yüklenmesine neden olur. Bu ciddi sorunlara yol açabilir, bu nedenle bu e-postayı görmezden gelmeli ve ekini açmamalısınız.

**“Love Letter Email Virus” bilgisayarımıza nasıl bulaşır ?**

Çoğu spam e-posta kampanyasında olduğu gibi, "Love Letter Email Virus" da kötü amaçlı ekler aracılığıyla bilgisayarlara bulaşır (bu durumda "Love\_You\_20918824-2019-txt.zip" ZIP arşiv dosyası içinde bulunan "Love\_You\_20918824-2019-txt.js" adlı JavaScript dosyası).

Diğer durumlarda, e-postalar, enfekte dosyalara veya kötü amaçlı eklerine (örneğin, .exe uzantılı yürütülebilir dosyalar, Microsoft Office belgeleri, PDF belgeleri vb.) yönlendiren web bağlantıları içerebilir. Örneğin, açılan kötü amaçlı bir MS Office belgesi, makro komutlarını etkinleştirmek için izin talep eder. Bunları etkinleştirerek, kullanıcılar bilgisayara enfeksiyon indirilmesine ve yüklenmesine izin vermiş olurlar.

**“Love Letter Email Virus” ‘ e karşı bilgisayarımızı nasıl koruyabiliriz ?**

Şüpheli, bilinmeyen veya güvenilmeyen e-posta adreslerinden gelen e-postaları (ve içlerinde sunulan web bağlantılarını veya ekleri) görmezden gelin. Çoğu zaman, bunlar tanınmış şirket adlarını kullanarak meşru mesajlar gibi sunulur. Ayrıca, dosyaları (yazılım) resmi ve güvenilir kaynaklardan indirin.

Güvenilir bir casus yazılım veya antivirüs programının bilgisayarınızda yüklü olması da virüs veya diğer tehditler tarafından bilgisayar enfeksiyonunu önlemeye yardımcı olabilir. Eğer zaten "Love Letter Email Virus" ekini açtıysanız, infiltrasyona uğramış kötü amaçlı yazılımları otomatik olarak temizlemek için Windows için Combo Cleaner Antivirus ile tarama yapmanızı öneririz.