



**T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



Yüksek Lisans Tezi

UYGULAMA TEST VERİSİ OLUŞTURMA ve VERİ MASKELEME

Fatih Enes EKREN

Enformatik Anabilim Dalı

Enformatik Programı

**DANIŞMAN
Doç. Dr. Çiğdem EROL**

Mart, 2021

İSTANBUL

Bu alıřma, 12.03.2021 tarihinde ařağıdaki jüri tarafından Enformatik Anabilim Dalı, Enformatik Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Do. Dr. iğdem EROL(Danışman)
İstanbul Üniversitesi Enformatik Bölümü
Fakülte

Prof. Dr. Sevin GÜLSEÇEN
İstanbul Üniversitesi Enformatik Bölümü
Fakülte

Prof. Dr. Oya KALIPSIZ
Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi’nin aboneli olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Yüksek lisans tez çalışmam boyunca gösterdiği her türlü destek ve yardımlarından dolayı tez danışmanım Doç. Dr. Çiğdem Erol'a,

Üniversitemizde ders aldığım ve üzerimde emeği olan bütün hocalarıma,

Tez hazırlama sürecinde yardımlarını esirgemeyen şirket yöneticilerim Selçuk Mısırlıoğlu, Osman Meral, Fatih Sarı, Fatih Ayvaz, Bahattin Çetin, Hacı Mehmet Atılgan ve çalışma arkadaşlarım Ayhan Çelik ve İsmail Yılmaz'a,

Her zaman beni destekleyerek, her anımda yanımda duran ve beni hiçbir zaman yalnız bırakmayan canım aileme,

Bana olan inancıyla, her zaman yanımda olduğunu gösteren, bunu en içten duygularla hissetmemi sağlayan ve destekleriyle beni ayakta tutan sevgili eşime teşekkürü bir borç bilirim.

Mart 2021

Fatih Enes EKREN

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ.....	ix
SİMGE VE KISALTMA LİSTESİ	x
ÖZET	xi
SUMMARY	xiii
1. GİRİŞ	1
2. GENEL KISIMLAR.....	4
2.1. VERİ.....	4
2.1.1. Kişisel Veri	6
2.1.2. Özel Nitelikli Kişisel Veri	7
2.1.3. Kişisel Verilerin Kullanımı	7
2.2. VERİ MASKELEME	9
2.2.1. Veri Maskeleye ve Güvenlik	13
2.2.2. Veri Maskeleye Türleri.....	14
2.2.2.1. Statik Veri Maskeleye.....	14
2.2.2.2. Dinamik Veri Maskeleye	15
2.2.2.3. Anında Veri Maskeleye	15
2.2.2.4. Gizleme-Aşırı Maskeleye	16
2.2.3. Veri Maskeleye Teknikleri	16
2.2.3.1. İkame (Substitution)	17
2.2.3.2. Karıştırma (Shuffling)	17
2.2.3.3. Şifreleme (Encryption)	18
2.2.3.4. Sayısal değışiklik (Number and Date Variance).....	18
2.2.3.5. Geçersiz kılma veya silme (Nulling out or Deletion)	19
2.2.3.6. Maskeleye (Masking Out)	19
2.2.4. Veri Maskeleyemenin Olumlu ve Olumsuz Yönleri	19
2.3. TEST VERİSİ.....	20
2.3.1. Yazılım Projelerinde Test Sürecinin Konumu.....	20

2.3.2. Test Verisi Oluřturma.....	22
2.3.3. Test Verisi Yönetimi	23
2.3.4. Test Verisi Yönetimi Yařam Döngüsü.....	24
3. MALZEME VE YÖNTEM.....	26
3.1. ÇALIřMANIN AMACI VE HEDEFİ.....	26
3.2. ARAřTIRMANIN MODELİ.....	26
3.3. ARAřTIRMADA KULLANILAN YAZILIM ARACI SEÇİMİ.....	26
3.3.1. Arařtırmada Kullanılan Yazılım Aracı.....	27
3.4. VERİNİN ELDE EDİLMESİ VE HAZIRLANMASI.....	30
3.5. VERİ MASKELEME YÖNTEMİNİN SEÇİLMESİ	34
3.6. UYGULANACAK MASKELEME TEKNİKLERİNİN BELİRLENMESİ.....	35
3.7. MASKELENEN VERİ SETLERİNİN AKTARIMI	40
4. BULGULAR.....	44
4.1. MASKELEME İřLEMİNİN UYGULANMASI VE VERİ SETLERİNİN KARřILAřTIRILMASI	44
4.1.1. E-posta Maskeleme Teknięi Sonuçları.....	47
4.1.2. Dijital Maskeleme Teknięi Sonuçları.....	48
4.1.3. Tarih Maskeleme Teknięi Sonuçları	49
4.1.4. Hash Algoritmasıyla Maskeleme Teknięi Sonuçları.....	49
4.1.5. Basit Yıldız Maskeleme Teknięi Sonuçları.....	50
4.1.6. Sütun Gizleme Teknięi Sonuçları	51
4.1.7. Karıřtırma Teknięi Sonuçları	51
4.1.8. Yıldız Maskeleme Teknięi Sonuçları.....	52
4.2. MASKELEME İřLEMİNİN TEKLİ VE ÇOKLU İř PARÇACIKLARIYLA UYGULANMASI VE ZAMANSAL PERFORMANS KARřILAřTIRILMASI	53
5. TARTIřMA VE SONUÇ	55
KAYNAKLAR.....	58
ÖZGEÇMİř	64

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: VEBB Hiyerarşisi [11]	4
Şekil 2.2: Kişisel Veri	6
Şekil 2.3: Özel Nitelikli Kişisel Veri	7
Şekil 2.4: Genel Veri Maskeleye Süreci [32]	10
Şekil 2.5: Veri Maskeleye Sistem Mimarisi [33].....	11
Şekil 2.6: Statik Veri Maskeleye [40].....	14
Şekil 2.7: Dinamik Veri Maskeleye [40]	15
Şekil 2.8: Anında Veri Maskeleye Mimarisi [42].....	16
Şekil 2.9: Veri Maskeleye Teknikleri [46].....	17
Şekil 2.10: Yazılım Hata Maliyetinin Zamansal Gösterimi [55]	22
Şekil 2.11: Yazılım Test Süreci Adımları [60]	24
Şekil 3.1: CopyCat Uygulama Arayüzü.....	27
Şekil 3.2: MUSTERILER Tablo İçeriği	31
Şekil 3.3: SOZLESME Tablo İçeriği.....	31
Şekil 3.4: Varlık İlişki Diyagramı.....	32
Şekil 3.5: MUSTERILER Tablosu Veri Seti 1.Kısım	33
Şekil 3.6: MUSTERILER Tablosu Veri Seti 2.Kısım	33
Şekil 3.7: SOZLESME Tablosu Veri Seti.....	34
Şekil 3.8: Çalışma Akış Şeması	35
Şekil 3.9: E-posta Maskeleye Tekniği	35
Şekil 3.10: Dijital Maskeleye Tekniği	36
Şekil 3.11: Tarih Maskeleye Tekniği.....	36
Şekil 3.12: Hash Algoritmasıyla Maskeleye Tekniği	37

Şekil 3.13: Basit Yıldız Maskeleye Tekniği	37
Şekil 3.14: Sütun Gizleme Tekniği	38
Şekil 3.15: Karıştırma Tekniği.....	38
Şekil 3.16: Yıldız Maskeleye Tekniği.....	39
Şekil 3.17: Veri Alanlarına Tanımlanan Maskeleye Tekniklerinin Listesi	39
Şekil 3.18: Veri Seti Aktarımı Yapılan Ortam Bilgilerinin Seçilmesi.....	40
Şekil 3.19: Veri Seti Aktarım İsteği Bilgilerinin Detayı.....	41
Şekil 3.20: Veri Seti Aktarım İsteği Oluşturma	41
Şekil 3.21: Veri Seti Aktarımını Gerçekleştirme	42
Şekil 3.22: Çoklu İş Parçacıklarıyla Paralel Maskeleye İşlemi	42
Şekil 3.23: Tekli İş Parçacığıyla Paralel Olmayan Maskeleye İşlemi	43
Şekil 4.1: Maskelenmeden Önce MUSTERILER Tablosu Veri Seti 1.Kısım.....	44
Şekil 4.2: Maskelendikten Sonra MUSTERILER_TEST Tablosu Veri Seti 1.Kısım.	45
Şekil 4.3: Maskelenmeden Önce MUSTERILER Tablosu Veri Seti 2.Kısım.....	45
Şekil 4.4: Maskelendikten Sonra MUSTERILER_TEST Tablosu Veri Seti 2.Kısım.....	46
Şekil 4.5: Maskelenmeden Önce SOZLESME Tablosu Veri Seti.....	46
Şekil 4.6: Maskelendikten Sonra SOZLESME_TEST Tablosu Veri Seti	47
Şekil 4.7: Çoklu İş Parçacıklarıyla Maskeleye İşleminin Log Kayıtları.....	53
Şekil 4.8: Tekli İş Parçacığıyla Maskeleye İşleminin Log Kayıtları	53

TABLO LİSTESİ

	Sayfa No
Tablo 3.1: Maskeleye Uygulamalarının Karşılaştırılması.....	27
Tablo 4.1: E-posta Maskeleye Tekniğı Karşılaştırma	48
Tablo 4.2: Dijital Maskeleye Tekniğı Karşılaştırma	48
Tablo 4.3: Tarih Maskeleye Tekniğı Karşılaştırma.....	49
Tablo 4.4: Hash Algoritmasıyla Maskeleye Tekniğı Karşılaştırma	50
Tablo 4.5: Basit Yıldız Maskeleye Tekniğı Karşılaştırma	50
Tablo 4.6: Sütun Gizleme Tekniğı Karşılaştırma	51
Tablo 4.7: Karıştırma Tekniğı Karşılaştırma.....	52
Tablo 4.8: Yıldız Maskeleye Tekniğı Karşılaştırma	52
Tablo 4.9: Tekli ve Çoklu İş Parçacıklarıyla Maskeleye Sonuçları Karşılaştırma.....	54

SİMGE VE KISALTMA LİSTESİ

Simgeler Açıklama

% : Yüzde İşareti

Kısaltmalar Açıklama

AB : Avrupa Birliği

DES : Veri Şifreleme Standardı (*Data Encryption Standart*)

DIKW : Veri, Enformasyon, Bilgi, Bilgelik (*Data, Information, Knowledge, Wisdom*)

DNA : Deoksiribo Nükleik Asit

DVM : Dinamik Veri Maskeleye

ETDHK : Elektronik Ticaretin Düzenlenmesi Hakkında Kanun

GDPR : Genel Veri Koruma Yönetmeliği (*General Data Protection Regulation*)

HDFS : Hadoop Dağıtılmış Dosya Sistemi (*Hadoop Distributed File System*)

IP : İnternet Protokolü (*Internet Protocol*)

KVKK : Kişisel Verilerin Korunması Kanunu

LDAP : Basit İndeks Erişim Protokolü (*Lightweight Directory Access Protocol*)

MD5 : İleti Derleme Algoritması (*Message Digest Algorithm 5*)

SHA : Güvenli Karma Algoritma (*Secure Hash Algorithm*)

SQL : Yapılandırılmış Sorgu Dili (*Structured Query Language*)

SVM : Statik Veri Maskeleye

TSQR : Türkiye Yazılım Kalite Raporu (*Turkey Software Quality Report*)

VEBB : Veri, Enformasyon, Bilgi, Bilgelik

VT : Veri Tabanı

ÖZET

YÜKSEK LİSANS TEZİ

UYGULAMA TEST VERİSİ OLUŞTURMA ve VERİ MASKELEME

Fatih Enes EKREN

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman : Doç. Dr. Çiğdem EROL

Yazılım uygulamalarında test verisinin yönetilmesi, test sürecinin önemli alt başlıklarından bir tanesidir. Test verisinin hazırlanması ciddi bir çaba gerektirmekte ve bundan dolayı test süreçlerinde büyük zaman kayıpları meydana gelebilmektedir. Test verisinin oluşturulması ve aktarılmasında dikkat edilmesi gereken noktalar bulunmaktadır. Kişisel verinin bilişim sistemleri üzerinden kullanımı, kamu ve özel sektör fark etmeksizin çok yaygındır. Bu şekilde veri kullanımının sağladığı bazı avantajlar ve kolaylıklar olmasına karşın, bu durum veri güvenliği riskinin artmasına da neden olabilmektedir. Kuruluşların birçoğu veri güvenliği maliyetlerini minimize etmek, verisini korumak için veri maskeleme kullanımını tercih etmektedir.

Bu tez çalışmasında, canlı ortam veri tabanındaki verinin uygun veri maskeleme yöntemi ve maskeleme teknikleriyle maskelenerek test ortam veri tabanında oluşturulması hedeflenmiştir. Ayrıca maskeleme işleminde performansın nasıl artırılabilirliği yani zamansal maliyetinin nasıl azaltılabileceği konusunda örnek bir karşılaştırma yapılmıştır. Çalışmada materyal olarak, canlı ortam veri tabanına benzer şekilde oluşturulmuş örnek veri seti kullanılmıştır. Çalışma içeriğine uygun belirlenen veri maskeleme uygulamasını örnek veri seti üzerinde kullanabilmek için veri maskeleme yöntemi ve teknikleri belirlenmiştir. Seçilen veri maskeleme uygulaması, statik veri maskeleme yöntemi ve maskeleme teknikleriyle oluşturulan veri setine uygulanmış ve sonrasında maskelenmiş veri, test ortam veri tabanına aktarılmıştır. Uygulanan veri maskeleme teknikleri ayrı başlıklar altında, örnek veri seti ve maskelenmiş veri seti olarak

karşılaştırılmıştır. Maskeleme ve maskelenmiş verinin aktarımı işlemi ilk olarak çoklu iş parçacıklarıyla yani birbirinden bağımsız ve paralel çalıştırma olarak yapılmış, daha sonra bu işlem tekli iş parçacığıyla yani paralel çalıştırma olmadan tekrardan yapılmıştır. Çoklu iş parçacıklarıyla paralel olarak yapılan veri maskeleme ve maskelenmiş verinin aktarım işleminin, tekli iş parçacığıyla paralel çalıştırma olmadan yapılan maskeleme ve maskelenmiş verinin aktarım işlemine göre %32 oranında zamansal performansı arttırdığı yani %32 oranında zaman tasarrufu sağladığı gözlemlenmiştir.

Mart 2021, 78 sayfa.

Anahtar kelimeler: Veri Maskeleme, Test Verisi Yönetimi, Statik Veri Maskeleme



SUMMARY

M.Sc. THESIS

APPLICATION TEST DATA CREATION and DATA MASKING

Fatih Enes EKREN

İstanbul University

Institute of Graduate Studies in Sciences

Department of Informatics

Supervisor : Assoc. Prof. Dr. Çiğdem EROL

Managing test data in software applications is one of the important subtitles of the testing process. The preparation of test data requires a serious effort and therefore a great amount of time can be lost in test processes. There are points to be considered in the creation and transfer of test data. The use of personal data through information systems is very common, regardless of the public and private sector. Although there are some advantages and conveniences provided by data usage in this way, this may increase the risk of data security. Many organizations prefer to use data masking to minimize data security costs and protect their data.

In this thesis, it is aimed to create the data in the live environment database in the test environment database by masking it with appropriate data masking method and masking techniques. In addition, an exemplary comparison has been made on how the performance can be increased in the masking process, that is, how the time cost can be reduced. As a material in the study, a sample data set created similar to a living environment database was used. Data masking method and techniques have been determined in order to use the data masking application determined in accordance with the study content on the sample data set. The selected data masking application was applied to the data set created with the static data masking method and masking techniques, and then the masked data was transferred to the test environment database. Applied data masking techniques were compared under separate headings, as sample data set and masked data set. The masking and the transfer of masked data

were first performed with multiple threads, independent and parallel operation, then this process was repeated with a single thread without parallel operation. It has been observed that data masking and transfer of masked data performed in parallel with multiple threads increase the temporal performance by 32% compared to the transfer process of masking and masked data without parallel operation with a single thread, in other words, it saves time by 32%.

March 2021, 78 pages.

Keywords: Data Masking, Test Data Management, Static Data Masking



1. GİRİŞ

Günümüz dünyasında veri ihlallerinin neden olduğu yaklaşık olarak 3,86 milyon dolar bir maliyetten bahsedilmektedir [1]. Veri ihlalleri bazen şirketlerin işleyiş ve mevzuata uygun olma durumlarını etkileyerek finansal olarak zarara uğramalarına sebep olabilir. Haberlere çıkan geniş çaplı bir veri ihlali, daha sonraki yıllarda şirketin itibarına zarar verebilir ve diğer şirketlerle rekabetinde dezavantaja neden olabilir. Bu nedenle iş ve müşteri bilgilerinin gizliliğini sağlamak ve korumak günümüzde çok önemli bir gereklilik haline dönüşmüştür.

Şirketler için veri ihlali sadece bir güvenlik sorunu değil, yasal yükümlülükler açısından ele alınması gereken bir konudur. Çıkarılan önemli yasalarla veri güvenliği işletme çatısı altındaki herkesin ortak sorumluluğu haline gelmiştir [2].

Avrupa Birliği Genel Veri Koruma Yönetmeliği (*General Data Protection Regulation - GDPR*) [3], 25 Mayıs 2018’de yürürlüğe girmiştir. Bu yönetmeliğin bir amacı, şirketlerin kişisel verisini koruma yükümlülüklerin olduğunu bilmelerini sağlayan bir yasa olmasıdır. Yönetmelikle beraber vatandaşların kişisel verisi üzerinde ciddi manada kontrol sağlayabilmeleri hedeflenmektedir. GDPR’a göre şirketlerin kişisel veriyi doğru ve hatasız olarak, verileri koruyarak kullanmaları ve veri sahiplerinin haklarına riayette bulunmaları beklenecektir.

6698 sayılı Kişisel Verilerin Korunması Kanunu (KVKK) [4], 7 Nisan 2016 tarihinde yürürlüğe girmiştir. Bu kanun birçok konuda düzenleme içeren detaylı bir çalışmadır. Kişilerin temel hak ve özgürlükleri için koruma sağlamak, kişisel veri işlenirken özel hayatın gizliliğini sağlamak ve kişi temel hak ve özgürlüklerini korumayı amaçlayan kurallar barındırmaktadır. Bu kanunla beraber kişisel verinin korunması daha fazla önem arz etmektedir.

KVKK ve GDPR, veri güvenliğini sağlamak için ciddi yaptırımlara sahip iki farklı düzenleme olup, bu düzenlemeler aynı zamanda kişisel veri ihlallerine karşı güven vermektedir [5].

Bununla birlikte, artan sayıda birçok kuruluş, verisini proaktif olarak korumak, güvenlik ihlallerinin maliyetinden kaçınmak ve uyumu sağlamak için veri maskeleyesine güvenmektedir. Genel olarak, veriyi maskelemenin birincil işlevi, bilgilerin açıklığı olmadan

birisinin görebileceği durumlarda hassas, özel bilgileri korumaktır [2]. Şirketlerin kullandıkları ya da ürettikleri yazılım uygulamalarının test süreçleri içerisinde, test verisinin yönetilmesi bu açıdan çok önemlidir.

Test verisinin istenilen boyutta, çeşitlilikte ve en önemlisi uygulamanın test edilirken bütün ihtiyaçlarını karşılayabilecek şekilde olması beklenmektedir. Bir uygulama test sürecinde uygulamanın canlı ortam veri tabanından test ortam veri tabanına veri aktarımının yapılmasının gerektiği durumlar bulunmaktadır. Bu durumlarda orijinal verinin korunması ve veri ihlallerinin önüne geçilmesi gerekmektedir [6].

Bu tez çalışmasında canlı ortam veri tabanında bulunan orijinal verinin uygulamaların test süreçlerinde orijinal halleriyle kullanılmalarının önüne nasıl geçilebileceği, canlı ortam veri tabanındaki verinin maskelenerek test ortam veri tabanında nasıl oluşturulacağının gösterilmesi hedeflenmektedir. Ayrıca veri tabanında bulunan veri boyutuna göre maskeleme işleminin zamansal maliyeti artacağı için bu kapsamda performansın nasıl arttırılabileceği yani zamansal maliyetin nasıl düşürülebileceği konusunda bilgi verilmesi de hedeflenmektedir.

Çalışmanın genel kısımlar bölümünde, literatür araştırması yapılarak çalışmanın içeriğinde geçen bilgiler hakkında açıklamalar yapılmıştır.

Çalışmanın malzeme ve yöntem bölümünde, çalışmanın amacı ve hedefi, araştırmanın modeli, araştırmada kullanılacak yazılım aracının seçimi, çalışmada kullanılacak verinin elde edilmesi ve hazırlanması, çalışmada kullanılacak veri maskeleme yönteminin seçilmesi, uygulamada kullanılacak olan maskeleme tekniklerinin belirlenmesi ve maskelenen veri setlerinin aktarımı konularında bilgiler verilmiştir. Yine bu bölüm içerisinde, canlı ortam veri tabanına benzer şekilde oluşturulan örnek veri seti kullanılmıştır. Çalışma içeriğine uygun veri maskeleme uygulaması, örnek veri setine uygulamak için veri maskeleme yöntemi ve teknikleri belirlenmiştir. Seçilen veri maskeleme uygulaması, statik veri maskeleme yöntemi ve maskeleme teknikleriyle örnek veri setine uygulanmış ve sonrasında maskelenmiş veri test ortam veri tabanına aktarılmıştır. Bu işlem öncelikle çoklu iş parçacıklarıyla yapılmış, daha sonrasında tekli iş parçacığıyla maskeleme ve veri aktarımı tekrardan yapılmıştır.

Çalışmanın bulgular bölümünde, uygulanan veri maskeleme teknikleri ayrı başlıklar altında, örnek veri seti ve maskelenmiş veri seti olarak karşılaştırılmıştır. Çalışmada maskeleme ve

maskelenmiş verinin aktarımı işleminin çoklu iş parçacıklarıyla ve tekli iş parçacığıyla yapılması bu bölümde karşılaştırılmış ve uygulama performans çıktıları gösterilmiştir.

Çalışmanın son bölümünde, literatürde araştırılarak bulunan bilgiler ışığında çalışmanın bulgularıyla beraber varılan sonuçları karşılaştırılmıştır. Açıklanan sonuçlarla beraber ileride yapılması beklenen araştırmalara yönelik önerilerde bulunulmuştur.

Bu tez çalışmasıyla KVKK kapsamında Türkiye’de yasal zorunluluk haline gelen kişisel verinin korunmasına ilişkin olarak, veri maskeleme yöntem ve tekniklerinin kullanılmasının kişisel verinin korunmasına yönelik kullanılabilir bir yaklaşım olduğunun gösterilmesi amacıyla canlı ortam veri tabanındaki verinin uygun veri maskeleme yöntemi ve maskeleme teknikleriyle maskelenerek test ortam veri tabanında oluşturulması hedeflenmiştir. Ayrıca maskeleme işleminde performansın nasıl arttırılabileceği yani zamansal maliyetinin nasıl düşürülebileceği konusunda örnek bir çalışma yapıp, yapılan bu çalışmanın sonuçları karşılaştırılmıştır. Tezin konusuna benzer çalışmaların Türkçe literatürde nadir olarak bulunması nedeniyle çalışmanın literatüre katkı yapacağı düşünülmektedir.

2. GENEL KISIMLAR

2.1. VERİ

Herhangi bir problemin çözümüne yardımcı olabilecek her çeşit ölçüm, olgu, bilgi ve değer olarak tanımlanabilen veri sözcüğü, yazılı ve sözlü ifadeler olabileceği gibi modeller, resimler, şekiller, eşya, sembol ve rakamlarla ifade edilebilir [7]. Yılmaz [8], veri için, bilgi ve enformasyona temel, yalnız başına kullanımı olmayan ve bir anlam ifade edemeyen ham bilgi olarak; anlamlandırılmaya, yorumlanmaya, ilişkilendirilmeye, analiz edilmeye ve gruplandırılmaya gereksinim duyulan olarak tanımlamıştır. Prytherch [9], veri hakkında, enformasyon için kullanılan ve veri tabanında yer alan bir terim olarak ifade etmiştir. Şeker [10] ise veriyi, herhangi bir işlem olmaksızın, ölçüm yöntemleri veya gözlem ile ortamdan edinilen her türlü değer olarak belirtmektedir.

Veri tanımı araştırılırken bilgi ve enformasyon kavramları ortaya çıkabilmektedir.

Literatürde “Bilgi Hiyerarşisi”, “Enformasyon Hiyerarşisi” ve “Bilgi Piramidi” isimleriyle anılan tanımsal sıralama grafiği için ayrıca “veri” (*data*), “enformasyon” (*information*), “bilgi” (*knowledge*) ve “bilgelik” (*wisdom*) kelimelerinin baş harflerinden oluşan “VEBB Hiyerarşisi” (*DIKW Hierarchy*) ismi kullanılmaktadır [11](Şekil 2.1).



Şekil 2.1: VEBB Hiyerarşisi [11].

Şekil 2.1'deki piramide göre en altta yer alan veri çeşitli analiz, biçimlendirme, birleştirmelerle enformasyona yani piramidin bir üst aşamasına dönüşüyor. Veri, *Merriam-Webster* sözlüğünde hem kullanışlı hem de konuyla alakasız veya gereksiz bilgiyi de içeren, bilgisayarlardan

edinilmiş, sayısal biçimde iletilip işlenebilen şeklinde tanımlanırken aynı kaynakta muhakeme veya hesaplamada kullanılan (ölçü veya istatistik) gerçek bilgiler anlamıyla yer almaktadır [12].

Donald E.Hawkins'e göre ise veri ölçülebilen, sayılabilen, bir araya getirilebilen bir olgu ve istatistiklerken enformasyon sınıflandırılan ve bir anlamı ilişkisi veya amacı olan veridir [13].

Bilgi sistemleri ile verinin üretimi, alınması ve işlenmesi sonrasında veriden enformasyon çıkarılır. Enformasyon, metalaştırılmış, nesnel, aktarılabilir, dönüştürülebilir, şeffaf ve ölçülebilirdir [14].

Veri, enformasyon ve bilgi kavramlarını tam olarak anlayabilmek ve aralarındaki ilişkiyi çözebilmek için bu kavramlara bütüncül olarak bakmak gereklidir; çünkü birbirleriyle doğrudan ilişkili olan bu kavramların birinin açıklanması çoğu zaman diğer kavramların açıklanmasını da gerektirmektedir. Bu değer yargısıyla açıklanacak olursa, veriyi bir formül içerisinde düzenleyerek enformasyon yaratılabilir. Benzer şekilde bilgi yaratmaysa, enformasyonun belli bir detaya sahip üretken veya verimli bir amaç için kullanılmasıdır. Bu doğrultuda üç farklı kavramla karşılaşılır bunlar; veri, bilgi ve enformasyon kavramlarıdır. Bunlara ek olarak iki dönüşüm süreci vardır; birincisi, verinin belli bir formülle düzenlenerek enformasyona dönüştürülmesi; ikincisi ise, bilginin enformasyona dönüştürüldüğü dönüşüm süreci olarak açıklanabilir, ayrıca enformasyonun bilgiye dönüşüm süreci olarak ifade edilebilir [15].

Bir süreç sonunda elde edilebilen kavrama bilgi denir [7]. Verinin işlenmiş hali olarak belirtilir. Bu bağlamda bilgiye ulaşmada en temel kavram 'Veri'dir. Veri, "işlenmemiş kanıtlar" olarak araştırmanın temel dayanağını oluşturmaktadır ve problemin çözümünü sağlayacak sonuçlara varmada, işlenerek anlam çıkarmada kullanılır. Her bilimsel araştırmanın geçmişteki bilgi birikime dayandırılmış olması gerekliliği vardır. Geçmişten günümüze aktarılan bilgi birikimi, araştırmacının doğrudan kullanacağı veri olmayıp araştırma probleminin çözümüne yardım edici vasıfta olduğunda veri olarak kullanılabilir. Veri, herhangi bir sonuca ulaşabilmek için gerekli olan "ilk bilgi" ya da "...anlam çıkartmada veya sonuca ulaşmak için kullanılan kayıtlar, nicelikler, sayı kümeleri veya olaylar" olarak tanımlanabilmektedir.

Kişisel veri, KVKK'ya göre iki başlık altında tanımlanmaktadır [4]. Bunlar:

- Kişisel Veri
- Özel Nitelikli Kişisel Veri

2.1.1. Kişisel Veri

24.03.2016 tarihinde kabul edilen “6698 sayılı Kişisel Verilerin Korunması Kanunu” (KVKK) gereğince, “kimliği belirli yahut belirlenebilir olan bir gerçek kişiye ilişkin her türlü bilgi kişisel veridir” [4]. Bunun dışında kişinin sadece şahsi değil, ailesine ait olan (ailevi) bilgileri de ona ait kişisel veri olarak değerlendirilmektedir [16]. Bu yönde kanunda kişisel veri “bireyin şahsi, ailevi ve mesleki özelliklerini gösteren, o bireyi diğerlerinden ayırmayı ve niteliklerini ortaya koymayı sağlayan her türlü bilgi” şeklinde tanımlanabilir [17]. “Kişisel veriler ilgili kişinin açık rızası olmaksızın işlenemez” [4].



Şekil 2.2: Kişisel Veri.

Şekil 2.2’de örnekleri verilen kişisel veriyi daha detaylandırmak gerekirse; kişinin adı, soyadı, doğum yeri ve tarihi, kimlik numarası, nüfusa kayıtlı olduğu yer, medeni hâli gibi kimlik bilgilerinin yanı sıra, bedenindeki ben, yara izi, dövme gibi belirleyici işaretler, telefon numarası, kredi kartı numarası, banka hesap numarası, fotoğrafı, kişisel görüntüsü, emeklilik, sosyal sigorta, kurum sicil numarası, sosyal medya hesap bilgileri, bunlara ilişkin şifreler, internet protokolü (IP) adresi, elektronik posta adresi, güvenlik kodları, ıslak ve/veya elektronik imzası, matbu veya dijital, örneğin e-posta yolu ile veya sosyal ağlarda yaptığı yazışma ve paylaşımlar hep ait olduğu kişisel verilerdendir [16,18–21].

2.1.2. Özel Nitelikli Kişisel Veri

“Özel nitelikli kişisel veriler, işlenmeleri halinde ilgili kişilerin mağdur olmasına veya ayrımcılığa maruz kalmasına neden olma riski taşıyan verilerdir. Özel nitelikli kişisel verilerin, ilgilinin açık rızası olmaksızın işlenmesi yasaktır” [4].



Şekil 2.3: Özel Nitelikli Kişisel Veri.

Şekil 2.3’de örnekleri verilen; kişilerin etnik kökeni, ırkı, siyasi düşüncesi, dini, mezhebi veya diğer inançları, felsefi inancı, dernek, sendika ya da vakıf üyeliği, kılık ve kıyafeti, cinsel hayatı, sağlığı, ceza mahkûmiyeti ve güvenlik tedbirleriyle ilgili verisi (kayıt ve bilgileri) ile parmak izi, retina görüntüsü, Deoksiribo Nükleik Asit (DNA) özellikleri gibi genetik ve biyometrik verisi özel nitelikli kişisel veri olarak kabul edilmiştir [4,19,20,22].

2.1.3. Kişisel Verilerin Kullanımı

Kişisel veri, internet teknolojilerindeki gelişmelerle beraber, yoğun olarak ticari kurumlar tarafından kullanılmaya başlanmıştır. Ticari kurumlar müşterilerinin davranışlarını ve internetteki hareketlerini gözlemleyerek bireye özel reklamlar ayarlayabilmeyi amaçlamaktadır. Bireyin şahsına münhasır reklam gönderme uygulaması ile beraber kişisel verinin işlenmesine yönelik yeni iş biçimleri yaygınlaşmıştır. Örneğin e-ticaret sitelerinde yapılan kişiye özel sipariş verisi işlenerek, e-ticaret sitesi üzerinden veri sahibi kullanıcıya tercihlerine bağlı olarak reklam gönderilebilmektedir. Elektronik Ticaretin Düzenlenmesi Hakkında Kanun (ETDHK) m. 6/1’e göre “ticari elektronik iletiler, alıcılara ancak önceden

onayları alınmak kaydıyla gönderilebilir”. Ticari elektronik iletilerin tanımını yapan ETDHK m. 2/c’ye ve Yönetmelik’in 2. maddesine göre *“elektronik ticari iletiler, bilgilendirme amacına (örneğin kargo takibi, ürün iadesi, ödemeye vb. konulara ilişkin bilgilendirme) sahip olabileceği gibi belirli bir mal ve hizmeti tanıtmak, beğendirmek ve böylelikle sürümünü sağlamak, bir başka ifadeyle reklam amacına da sahip olabilir”* [23–25].

Kişisel verinin işlenerek kullanılmasına yönelik iş biçimlerine sahip olan kurumlar, vermiş oldukları hizmetleri belirli bir ücret karşılığında değil, kullanıcılarının kişisel verisini isteyerek bunu yapmaktadırlar. Bu kurumlar, sahip oldukları kişisel veriyi anlamlı veri haline dönüştürerek sonrasında bunları reklam olarak kullanmayı amaçlamaktadırlar [26]. Bu işletmelerin sunmuş oldukları hizmetleri kullanan kişiler, “üreten-tüketiciler” şeklinde nitelendirilebilmektedirler. Ayrıca kendilerine sunulan hizmetleri kullanan bu kişiler, hem tüketici, hem de kişisel verisini paylaşmış oldukları için bu kurumların ortaya koyduğu hizmetin hammaddesini oluşturan verinin de üreticisi tanımına sahip olmaktadır [27,28]. Belirtilen işleri yapan bu kurumların olabildiğince fazla kişisel veriye sahip olabilmeleri, bu veriyle mümkün olabilecek en fazla çeşitte kurum ve analizlerini yapabilmeleri, ticari faaliyetlerini sürekli hale getirebilmek ve kazançlarını artırmak açısından bir zorunluluk olarak düşünülebilmektedir. Bunun aksine bu hizmetlerden fayda sağlayan kişiler ise kişisel verisinin ve özel hayatının ciddi manada korunmasını istemektedirler. Bu durum, söz konusu tarafların menfaatlerinin çatışmasına neden olabilmektedir [28].

Temel hak ve özgürlüklerin korunması ve kişisel verinin işlenirken kontrol altında tutulabilmesi kişisel verinin korunabilmesiyle olur. Aslında verinin değil, veriye sahip olan kişilerin korunması amaçlanmaktadır. Diğer taraftan verinin korunması; kişiler ve haklarındaki verinin kısmi ya da tamamen otomatik veya otomatik olmayan yollarla işlenmesinden kaynaklanacak zararlardan kurtulma hedefiyle ortaya çıkmış ve kişisel verinin korunabilmesiyle alakalı olan ilkelere somutlaştırılmış teknik, hukuki ve idari önlemleri belirtmektedir. Kişisel verinin korunabilmesi bu ifadeyle, kişilere ilişkin verinin toplanarak saklanması, kullanılması ve aktarılması gibi veri işleme süreçlerinin tüm aşamalarını barındıracak biçimde kişilere kontrol haklarını eskisi gibi tekrardan kazandırmayı hedeflediği söylenebilir. Bu hedefe bağlı olarak kişisel verinin korunabilmesi, bireyin verisinin geleceğini sadece kendisinin tayin etme hakkını ifade eder. Buna ek olarak insanlık onuru ve kişilik hakkı açısından kişisel verinin korunması bir gerekliliktir [4].

Veri maskeleye ihtiyacı, 6698 sayılı KVKK kanununun gerekliliklerden bir tanesi olarak ortaya çıkmıştır. Bu bağlamda kurumsal ağlarda saklanan hassas ve gizli olarak bölümlendirilmiş veriye yetkisi olmayan kişilerin erişebilmesini engellemek gayesiyle verinin üçüncü taraflarca anlaşılamayacak bir halde dönüştürülmesi işlemine veri maskeleye denilmektedir [29]. Art niyetli kişiler tarafından hassas ve gizli olan verilerin şirket içerisine veya şirket dışına çıkarılmasını engellemek amacıyla bu yöntemin kullanılması tercih edilmektedir. Genel olarak maskeleye gerekliliği olan veri çeşitleri şu şekilde sıralanabilir:

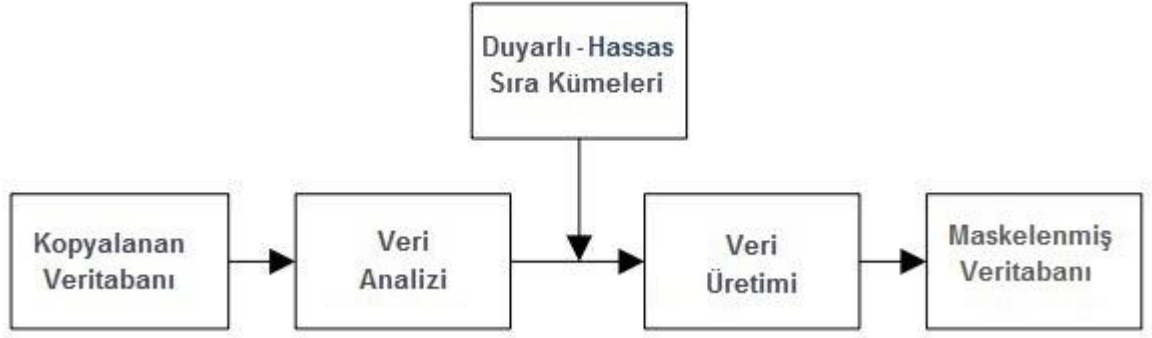
- Müşteri verisi
- Sağlık verisi
- Kişiyi özel diğer veri
- İnsan Kaynakları verisi
- Finans - Muhasebe verisi
- Bankacılık verisi

Belirtilen bu veri tipleri çeşitli alanlarda ve formatlarda olabilir. Veri ve bilgi güvenliğinin sağlanmasını noktasında şirketlerin en önemli işlem adımı, hangi bilgilerin hassas ve gizli nitelikte olduğu ve kimler tarafından görüntülenme ve değiştirilme yetkisinin olduğu bilgisinin belirgin olmasıdır. Doğru bir şekilde uygulanan her yöntem şirketlerdeki kişisel verinin korunması açısından yeterli olacaktır [30,31].

2.2. VERİ MASKELEME

Günümüz dünyasında veri ihlallerinin neden olduğu yaklaşık olarak 3,86 milyon dolar bir maliyetten bahsedilmektedir [1]. Bu nedenle iş ve müşteri bilgilerinin gizliliğini sağlamak ve korumak günümüzde çok önemli bir gereklilik haline dönüşmüştür. Çıkarılan önemli yasalarla veri güvenliği işletme çatısı altındaki herkesin ortak sorumluluğu haline gelmiştir. Bununla birlikte, artan sayıda birçok kuruluş, verisini proaktif olarak korumak, güvenlik ihlallerinin maliyetinden kaçınmak ve uyumu sağlamak için veri maskeleyesine güvenmektedir. Veri maskeleyesini ve kurumsal potansiyelini tam olarak açıklayabilmek için öncelikle nasıl çalıştığını anlamak gerekir. Adından da anlaşılacağı gibi, veri gizleme olarak da bilinen veri maskeleye, işletmelerin veriyi gizlemek için kullandıkları bir süreçtir. Genellikle, gerçek veri rastgele karakterler veya diğer veriyle gizlenir. Veri bir bakıma görüntüleme iznine göre sınırlandırılmış kapsama alınmıştır. Genel olarak, veriyi maskeleyenin birincil işlevi, bilgilerin

açıklığı olmadan birisinin görebileceği durumlarda hassas, özel bilgileri korumaktır. Kuruluşların bir veri tabanı oluşturmak için yüklenici, taşıyon bir firmayla çalıştığı bir senaryo düşünüldüğünde; verinin maskelenmesi, yüklenici firmanın gerçek, hassas müşteri bilgilerine erişmeden veri tabanı ortamını test etmesini sağlayacaktır [2].

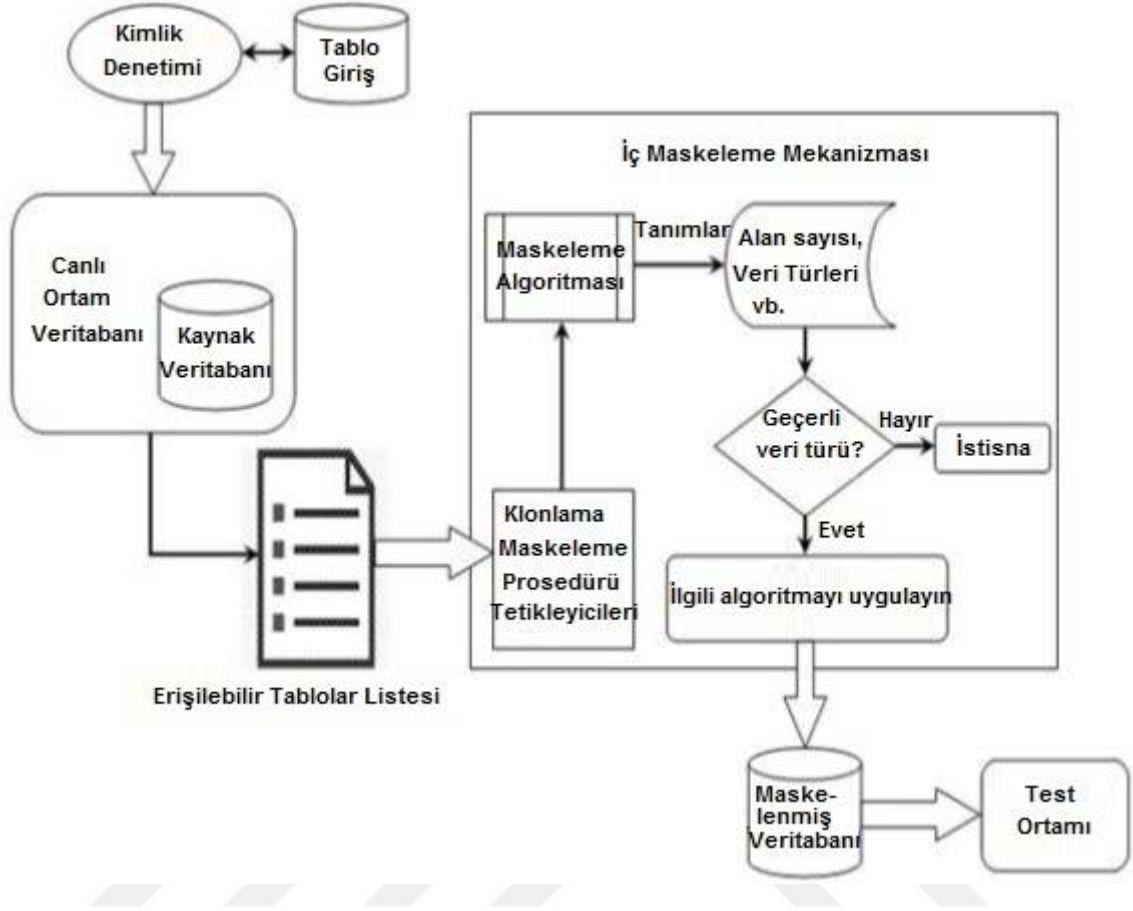


Şekil 2.4: Genel Veri Maskeleye Süreci [32].

Veri maskeleye, orijinal veriyi yok etmemek için genellikle kopyalanan veri tabanına dayanır. Genel veri maskeleye süreci Şekil 2.4'de gösterilmiştir. Veri maskeleye şu şekilde tanımlanabilir: Veri kısıtlamalarını karşılama öncülüğünde, orijinal verinin analizi yoluyla, yeni benzer veri oluşturmak için çeşitli temel maskeleye yönteminin ve birleşim yönteminin kullanımıdır [32].

Şekil 2.5'te gösterilen, veri maskeleye sistem mimarisinin arkasında çeşitli aşamalar vardır. Bunlar:

- Kimlik doğrulama, bir bilgi parçasına erişim talebinde bulunan kişi veya sistemin buna erişim için geçerli yetkiye sahip olmasının sağlandığı süreçtir. Özel ve genel bilgisayar ağlarında (İnternet dâhil) bu yöntem kullanılır. Kullanıcının kimliğini doğrulamak ve sisteme girmesine izin vermek için temel bir kullanıcı adı şifre sistemi sağlanmıştır.
- Başarılı bir kimlik doğrulama, veri tabanındaki tüm tabloların alınmasının yolunu açar.
- Kullanıcı belirtilen tabloyu ve hatta satırları / sütunları karıştırdığında / maskeleydiğinde, sisteme beslenen algoritma tetiklenir ve veri daha sonra kullanılmak üzere manipüle edilir.
- Son olarak, maskelenmiş veri tabanı test ortamına gönderilir ve bu ortam daha sonra güvenlikten ödün vermeden orijinal veri kadar iyi veriyle (maskelenmiş) çeşitli yeni geliştirme ve analitik süreçleri uygulayabilir [33].



Şekil 2.5: Veri Maskeleye Sistem Mimarisi [33].

Veri maskeleye, kuruluşların hassas veya gizli verisini değiştirmek için orijinal veriyle benzer özelliklere sahip gerçekçi ve tamamen işlevsel veri oluşturmaya olanak tanır. Veri maskeleye, orijinal hassas verinin alınmasına veya erişilmesine izin vermez. İsimler, adresler, telefon numaraları, sosyal güvenlik numarası ve kredi kartı bilgileri, bilgi içeriğinin uygunsuz ve yetkisiz görünürlükten korunmasını gerektiren veriye örneklerdir. Canlı ortam veri tabanları, değerli ve gizli bilgiler içerir, bu bilgilere erişim sıkı bir şekilde kontrol edilir ve büyük ölçüde kısıtlanır. Bununla birlikte, her canlı ortam genellikle klonlanmış geliştirme kopyalarına sahiptir ve bu tür test ortamları üzerindeki kısıtlamalar daha azdır. Bu, verinin kişisel kazançlar için uygunsuz bir şekilde kullanılması olasılığını büyük ölçüde artırır. Veri maskeleye, hassas veri tabanı kayıtlarını değiştirebilir, böylece test için kullanılabilir kalırlar, ancak gizlilik endişelerine neden olan önemli gizli veya kişisel olarak tanımlanabilir bilgiler içermezler. Yine de, test için kullanılan maskelenmiş veri, uygulamanın bütünlüğünü sağlamak için görünüşte orijinalini andırmaktadır [34].

Her türden şirket ve ajans büyük miktarda veri toplayabilir, depolayabilir ve işleyebilir. Veriden bilgi almak için analitik ve veri madenciliği araçlarını kullanarak iş sürecini iyileştirmek, bu tür veriyi toplamanın birincil amacıdır. Kuruluşlar, uygulama geliştirme, test etme veya veri analizi amacıyla canlı ortam verisini canlı ortam dışına kopyalarken hassas bilgilerden ödün verme riskiyle karşılaşır [35].

Veri Maskeleye, orijinal hassas veriyi geri alınamaz bir şekilde hayali veriyle değiştirerek bu riski azaltmaya yardımcı olur, böylece canlı ortam verisi bu ortam dışında kullanım için güvenli bir şekilde paylaşılabilir. Giderek artan sayıda işletme, kurumsal veriyi etkin bir şekilde güvence altına almak, veri güvenliği önlemlerini iyileştirmek ve veri ihlaliyle ilişkili maliyetleri düşürmek için veri maskeleye güvenmektedir. Veri maskeleye, canlı ortam dışında bulunan hassas bilgileri tanımlanamayacak hale getirerek veriyi korur ve kuruluşların geleneksel güvenlik platformlarını genişletmelerine olanak tanır. Hassas veya gizli veriyi maskeleyerek canlı ortam dışındaki sistemlere güvenle kopyalanmasını sağlar. Önceden oluşturulmuş veya özelleştirilmiş karmaşık maskeleye tekniklerinin kullanılması; kuruluşların orijinal bilgi özelliklerini (veri türleri, formatlar, vb.) koruyabilir, ayrıca veriyi ve referans bütünlüğünü koruyabilir. Uygulama geliştirme ve test etme için gerçekçi veri gereklidir. Genellikle geliştirme ekiplerine, dâhili olarak geliştirilen komut dosyası kullanarak oluşturulan canlı ortam veri tabanlarının kopyaları verilir. Ancak bu yöntem tam olarak güvenli değildir çünkü hassas bilgiler içeren gerçek veri yanlış kişilerin eline geçebilir [36].

Örneğin, bir çevrimiçi bankacılık sistemi test edilirken, uygulama testçisi müşteri kayıtlarını değiştirebilir ve sonuç olarak, güvenlik ve gizlilik endişesine neden olabilecek şekilde bireylerin adlarını, adreslerini, sosyal güvenlik numaralarını, telefon numaralarını ve diğer özel bilgilerini görüntüleyebilir. Buna ek olarak, test uzmanları veriyi kendi iyilikleri için kullanmak, veriyi kopyalamak ve organizasyondan çıkarmakla ilgileniyorlarsa ve bu durum sonucunda verinin yanlışlıkla sızdırılması, şirketin müşterilerine karşı olan güvenin kaybolmasına, ayrıca kuruma ciddi cezaların gelmesine neden olabilir. Müşterilerin saygısını ve güvenini kaybetmek, veri ihlali durumunda muhtemelen en büyük kayıptır [37].

2018 yılında, kurumsal işletmeler, özellikle GDPR [3] gereklilikleri ışığında, güvenlik stratejilerine veri maskeleye eklemeleri gerektiğini öğrenmişlerdir. GDPR, AB vatandaşlarından veri kabul eden tüm işletmelerin Mayıs 2018'e kadar yönetim ilkelerine uygun olmalarını zorunlu kılmıştır.

Maskeleye kullanılarak korunabilen birçok veri türü vardır, ancak iş dünyasında yaygın olarak kullanılanlardan bazıları şunlardır [2]:

- Kişisel olarak tanımlanabilir bilgiler
- Korumalı sağlık bilgileri
- Ödeme kartı bilgileri
- Fikri mülkiyet

2.2.1. Veri Maskeleye ve Güvenlik

Çoğu kuruluş, hassas veya düzenlenmiş canlı ortam verisini rutin olarak bu ortamın dışına kopyalarken, yanlışlıkla bilgi güvenliğini ihlal edebilir. Sonuç olarak, canlı ortam dışındaki veri giderek daha fazla siber suçluların hedefi haline gelir ve kaybolabilir veya çalınabilir. Canlı ortamlardaki veri ihlalleri gibi, canlı ortam dışındaki veri ihlalleri de milyonlarca doların harcanmasına, kuruluşun itibarını kaybetmesine ve markaya onarılamaz zararlar vermesine neden olabilir [38].

2012 Consumer Study on Data Breach Notification [39] araştırmasına göre müşterilerin % 40'ına kadar olan kesiminin kişisel verisini ifşa eden bir şirketle ilişkilerini sonlandırmayı düşündüğü bilinmektedir.

Veri maskeleye bir dizi güvenlik senaryosunda yararlıdır. Kurumsal işletmelerin veri maskeleyesini kullanmasının en önemli nedenlerinden bazıları şunlardır:

- Veriyi üçüncü taraf satıcılardan korumak için: Bazı verilerin üçüncü taraf pazarlamacılar, danışmanlar ve diğerleriyle paylaşılması gerekebilmektedir, ancak bazı bilgiler gizli tutulmalıdır.
- Operatör hatasından dolayı veri ihlallerinin oluşması: İşletmeler, şirket içerisindeki karar verme mekanizmasına ve çalışanlarına güvenir, ancak veri ihlalleri operatör hatasının bir sonucu olarak da oluşabilmektedir ve işletmeler veri maskeleye ile kendilerini korumalıdır.
- Tüm işlemler tamamen gerçek ve doğru verinin kullanılmasını gerektirmez: Bilgi Teknolojileri biriminde gerçek veriye ihtiyaç duymayan birçok iş süreci bulunmaktadır. Örneğin, bazı testler ve uygulama kullanımı.

Veri maskelemeyi tanımlamak, şirketinizin genel veri güvenliği stratejisinde oynadığı önemli rolü anlamak anlamına gelir [2].

2.2.2. Veri Maskeleme Türleri

Veri maskeleme yapılırken kullanılabilen çeşitli yaklaşımlar bulunmaktadır. Karıştırma, geçersizleştirme, bulanıklaştırma, şifreleme ve simge kullanma en fazla bilinen ve kullanılan türleridir. Veri maskelemede amaç veri değerlerinin herhangi bir yolla anlaşılabilmesi için değerlerin farklılaştırılması ve geri döndürülmesinin mümkün olmayacağı şekilde bunun yapılmasıdır. Ancak veri formatında herhangi bir değişiklik olmayacaktır. Kişisel verinin korunması noktasında doğru bir şekilde uygulanan herhangi bir yöntem yeterli olacaktır [2].

Genel olarak, bir veri maskeleme stratejisi seçilirken, kuruluşun boyutu, korumak istediği verinin konumu ve verinin karmaşıklığı dikkate alınmalıdır. Veri maskeleme, statik veri maskeleme (SVM), dinamik veri maskeleme (DVM), anında veri maskeleme (*on a fly*) ve gizleme-aşırı maskeleme olmak üzere 4 şekilde yapılabilir [2].

2.2.2.1. Statik Veri Maskeleme

Statik veri maskeleme, önemli verinin orijinal veri tabanı ortamında maskelenmesi sürecini ifade eder. İçerik bir test ortamına kopyalanır ve daha sonra üçüncü taraf satıcılar veya diğer gerekli taraflar arasında paylaşılabilir.



Şekil 2.6: Statik Veri Maskeleme [40].

Şekil 2.6'daki gibi veri maskelenir ve canlı ortam veri tabanından çıkarılır, daha sonra test ortam veri tabanına taşınır. Bu, üçüncü taraf danışmanlarla çalışmak için gerekli bir süreç olsa da, ideal değildir. Bunun nedeni, yinelenen bir veri tabanı için veri maskeleme işlemi boyunca, veri ihlallerini teşvik eden bir arka kapıyı açık bırakabilecek gerçek verinin ortaya konulmasıdır [2].

2.2.2.2. Dinamik Veri Maskeleye

Dinamik veri maskeleye (DVM), Şekil 2.7’de görüldüğü gibi üretilen verinin gerçek zamanlı maskelenmesini amaçlayan bir teknolojidir. DVM, istemcinin hassas veriye erişememesi için veri akışını değiştirir; orijinal canlı ortam verisinde herhangi bir değişiklik yapılmaz. Bu, canlı ortam veri tabanının asla terk edilmediği ve tehditlere karşı daha az hassas olunduğu anlamına gelir.



Şekil 2.7: Dinamik Veri Maskeleye [40].

Veri hiçbir zaman veri tabanına erişen kişiler tarafından çözümlenemez, çünkü içerikler gerçek zamanlı olarak karıştırılır ve bu da içerikleri doğrulanmamış hale getirir.

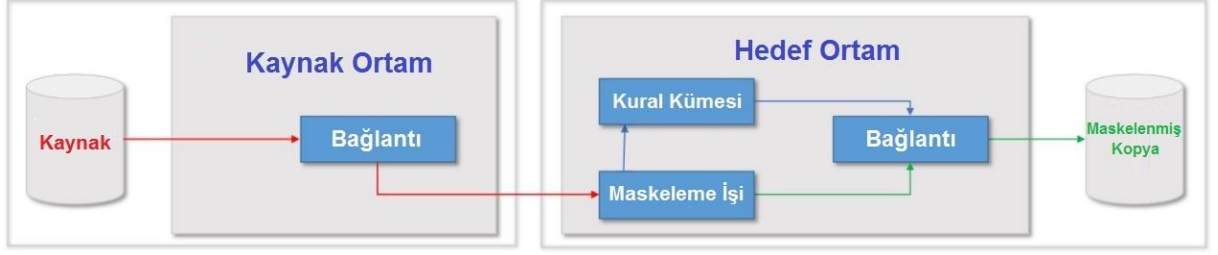
Dinamik maskeleye aracı adı verilen bir kaynak, ters proxy kullanarak belirli türde hassas veriyi bulur ve maskeler. Yalnızca yetkili kullanıcılar orijinal veriyi görebilir.

Dinamik veri maskeleyesinden kaynaklanan endişeler çoğunlukla veri tabanı performansından kaynaklanmaktadır. Kurumsal bir ortamda zaman para demektir ve milisaniyelerin bile değeri vardır. Böyle bir proxy'yi çalıştırmanın zamanla ilgili düşüncelerine ek olarak, proxy'nin güvenli olup olmadığı endişesi ortaya çıkabilir [2].

2.2.2.3. Anında Veri Maskeleye

Dinamik veri maskeleyesine benzer şekilde, istek üzerine anında veri maskeleyesi de gerçekleşir. Bu tür veri maskeleyesinde, veri, maskeleye öncesinde hedef veri tabanında mevcut değildir. Anonimleştirme kuralları, verinin kaynaktan hedefe taşınması sürecinin bir parçası olarak uygulanır. Çoğunlukla bu tür maskeleye, hedef veri tabanını oluşturan klonlama sürecine entegre edilir [41].

Anında (On-The-Fly) Veri Maskeleye



Şekil 2.8: Anında Veri Maskeleye Mimarisi [42].

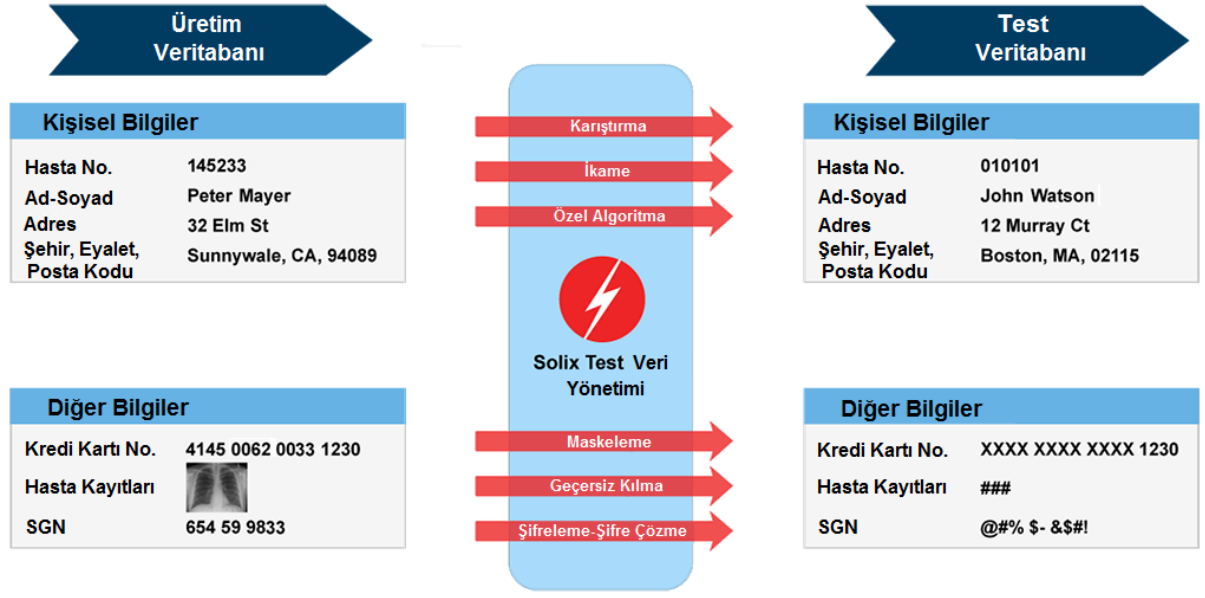
Şekil 2.8’de Anında Veri Maskeleye Mimarisi gösterilmektedir. Bu mimari avantajdır, çünkü veri asla hedef veri tabanında maskelenmemiş bir biçimde bulunmaz. Bu, bazı uygulamalar için tavsiye edilmez çünkü işlemdeki herhangi bir hata, verinin aktarımını zorunlu olarak kesintiye uğrattır. Aktarım tamamlandıktan sonra veriyi maskeleye yeteneği zahmetli olabilir. Bu, maskelenmiş hedef veri tabanının oluşturulduğu ve daha sonra belirli bir bilgi sütununun gerçekten maskelenmesi gerektiğine karar verildiği durumlarda olabilir [34].

2.2.2.4. Gizleme-Aşırı Maskeleye

Hassas değerlerin dosyalardan ve veri tabanlarından hep birlikte kaldırılması, veriyi maskeleye için çok güvenli ve verimli bir yoldur, ancak daha sonra yapay veri kümesinin oluşturulmasını gerektirdiğinden kullanım açısından çok zahmetlidir. Bu veri seti, referans bütünlüğünü bozmayacak şekilde yapılmalıdır. Bu yöntem başlangıçta maskeleye yapmanın ekonomik ve acil bir yolu olsa da, maliyet açısından çok pahalı bir yöntem olacaktır [43].

2.2.3. Veri Maskeleye Teknikleri

Çalışmanın bu kısmında veri maskeleye tekniklerinin altında yatan nasıl sorusunun cevabını oluşturan teknik detaylara yer verilmiştir. Bir mikro veri setinin orijinal veriyi, istatistiksel analiz için geçerli olan ve katılımcıların gizliliğini koruyacak şekilde yeni veri oluşturmak için veri maskeleye teknikleriyle değiştirilir [44]. Birçok veri maskeleye tekniği mevcuttur. Bunlardan en çok kullanılanları İkame, Karıştırma, Şifreleme, Sayısal değişiklik, Geçersiz kılma veya silme ve Maskeleyedir [45]. Şekil 2.9’da tüm veri maskeleye teknikleri örneklenmiştir. Bu veri maskeleye teknikleri ayrı başlıklar altında anlatılmıştır.



Şekil 2.9: Veri Maskeleye Teknikleri [46].

2.2.3.1. İkame (Substitution)

Veri maskeleymesini uygulamanın ve veri kayıtlarının gerçek görünümünü koruyabilmenin en etkili yöntemlerinden biridir. İkame, orijinal değerleri tam bir ikameyle örtmek veya maskelemek için kullanılan tekniktir. İkame, kullanıcıya gerçek görünen ancak onun için yararlı olmayan bu tür veriyi sağlamaya ihtiyaç duyulduğunda verinin şifrenmesi için kullanılır. İkame kullanarak çeşitli veri ve veri türleri değiştirilebilir. Sağlanan koşullara göre kullanılabilecek çeşitli ikame yöntemleri vardır. Bu teknik, büyük müşteri bilgilerine sahip yerlerde veya müşterinin işlemlerine ilişkin değerleri depolayan bir bankacılık sisteminde kullanılabilir [34].

Böyle bir sisteme örnek olarak, müşterinin adı, soyadı ve cinsiyeti ile müşteri kayıtlarının bulunduğu veri tabanındaki bir tablo olabilir, ikame kullanarak erkek müşterilerin adlarının ilk birkaç karakterini belirli bir karakterle değiştirebilir veya aynısı kadın müşteri adının ilk birkaç karakteri için yapılabilir. Posta kodu, maaşlar, sosyal güvenlik numarası ve hatta adresler gibi çeşitli veri tabanı alanlarında ikame yapılabilir [47].

2.2.3.2. Karıştırma (Shuffling)

Karıştırma, geçerli değer kümesi için sütunda maskelenen değerlerle değiştirilerek yeni bir değer türeten bir veri gizleme tekniğidir. Veri karıştırma, ikame etmeye biraz benzer olsa da,

veriyi maskelemek için farklı bir yaklaşıma sahiptir. Ancak veri setine belirli bir senaryo yerleştirilerek orijinal bilgiler hakkında bilgi edinmek çok daha kolaydır. Verinin karıştırılmasını şifrelemek için orijinal algoritmanın deşifre edilmesi de bir endişe kaynağı olacaktır. Karıştırma tekniğinin, maskenin yalnızca gerekli veri alanlarını etkileyecek şekilde sağlanması için belirli gereksinimlerin olduğu belirli senaryolarda, ikameye göre bazı gerçek avantajları vardır [35].

Örneğin; karıştırmadan önce 333-555-666 biçiminde bir sosyal güvenlik numarası olabilir ve bu numara karıştırdıktan sonra 666-333-555 olabilir.

2.2.3.3. Şifreleme (Encryption)

Genel olarak, üç tür şifreleme algoritması vardır: gizli anahtar algoritmaları, açık anahtar algoritmaları ve karma algoritma. Gizli anahtar algoritmaları, iletişimdeki her iki katılımcının da tek bir anahtarı paylaşması anlamında simetriktir. Başka hiç kimseyle paylaşılmayan özel bir anahtara sahip bir çift katılımcının aksine, herkesin bildiği bir açık anahtar yayınlanır. Veri maskeleme durumunda, mevcut şifreleme tekniklerinden herhangi birini kullanarak alanlardaki veriye şifreleme uygulanabilir. Maskeleme işlemi uygulanırken kullanılabilecek çeşitli standartlaştırılmış algoritmalar vardır. Çeşitli şifreleme algoritmaları DES, MD5, SHA, 3DES, vs.'dir. Bu algoritmaların kullanılması, bu algoritmalar veri üzerinde birçok şifreleme yinelemesi uyguladığından, verinin oldukça güvenli bir şekilde maskelenmesine yardımcı olur [48].

2.2.3.4. Sayısal değişiklik (Number and Date Variance)

Tabloda önemli olan çeşitli sayısal ve tarihle ilgili alanları maskelemek için sayısal ve tarih varyans tekniği kullanılabilir. Sayısal ve tarih farkı, tarihin veya sayısal bir değer orijinal değerinden farklı görünmesini sağlamak için kullanılabilir [34]. Bir maaş sütununa + 20 / -20 varyansı için sayısal varyans uygulanırsa, elde edilen veri çalışanlara ödenen maaş aralığı için hala yararlıdır ve yine de uygulamaları test etmek için kullanılabilir. Aynısı, maskeleme için geçerli tarihin + 40 / -40 gün artırılabilirdiği veya azaltılabilirdiği tarihlere sahip bir alanda da yapılabilir. Bu, uygunsuz yetkiye sahip kişilerin diğer alıcıların doğru maaşlarını veya doğum tarihlerini görmesini engelleyebilir [33].

2.2.3.5. Geçersiz kılma veya silme (Nulling out or Deletion)

Veriyi maskelemek için bir başka teknik, geçersiz kılma ve silme işleminin kullanılmasıdır. Geçersiz kılma yöntemi, herhangi birinin önemli veriyi görüntülemesini engellese de, test için kullanılan verinin gereksinimlerini kesinlikle karşılayamaz. Geçersiz kılma yöntemi genellikle çok önemli olan veya maskelemek için çok karmaşık olan alanlarda kullanılır. Bu yöntemle ilgili bir diğer sorun, veri setinde maskeleme uygulanmış olan sisteme girmek isteyen herkesin maruz kalmasıdır. Silme, çok nadiren uygulanmasına rağmen diğer maskeleme yöntemidir [33].

2.2.3.6. Maskeleme (Masking Out)

Verinin gizlenmesi, veri anonimleştirme işlemi için genel bir terim olmanın yanı sıra, belirli alanların bir maske karakteriyle (örneğin bir X) değiştirilmesi anlamına gelir. Bu, ön uç ekranlarda ve raporlarda aynı biçimlendirmeyi korurken veri içeriğini etkili bir şekilde gizler [49].

Örneğin, bir kredi kartı numarası sütunu şöyle görünebilir:

- 1594 6159 3021 2915
- 4811 2218 9311 3118
- 0987 1100 1326 0119

Ve maskeleme işleminden sonra bilgiler şu şekilde görünecektir:

- 1594 XXXX XXXX 2915
- 4811 XXXX XXXX 3118
- 0987 XXXX XXXX 0119

2.2.4. Veri Maskelemenin Olumlu ve Olumsuz Yönleri

Veri maskelemenin sistem üzerinde olumlu ve olumsuz etkileri olabilmektedir. Veri maskelemenin olumlu yönleri aşağıdaki gibidir;

- Kişisel verinin yasalara uygun olarak kullanımını sağlar.
- Test ve geliştirme ortamları için yapısal olarak benzer ancak kişisel olarak tanımlanamayan veri oluşturulmasını sağlar.
- Veri ihlali olasılığına karşı orijinal verinin korunmasını sağlar.

Veri maskelemenin olumsuz yönleri aşağıdaki gibidir;

- Geliştirmek ve sürdürmek için ek kaynağa ihtiyaç duyulabilir.
- Mevcut süreçte ek zaman ihtiyacı doğurur.
- Performansa olumsuz etki sağlayabilir [50].

2.3. TEST VERİSİ

Bir yazılımın canlı ortam sürecinde kullanılmak üzere oluşturulan veriye test verisi denir. Canlı ortamda gerçek kullanıcıların oluşturdukları verinin, test ortamında da oluşturulması bazen çok kolay olmayacaktır. Test verisi gerçek tecrübeyi yaşatabilmesi adına canlı ortam verisi değerinde olacak ancak; canlı ortamda da çalışmayacaktır [51].

2.3.1. Yazılım Projelerinde Test Sürecinin Konumu

Bir ürün için beklenen düzeyde bulunup bulunmadığını belirlemek, eğer bu düzeyde değilse istenilen düzeye getirebilmek amacıyla kullanılan sürece yazılım testi denilmektedir. Ürünün kalitesinden emin olarak müşteriye teslim edebilmek, geliştirme ya da yeniden düzeltme maliyetlerini azaltabilmek, geliştirme yapılan zaman aralığında oluşabilecek hataların bulunması ve ileride bu hataları almamak için bu aşamada çözüm üretilebilmesi, bu anlamda maliyet ve zaman kaybının önüne geçilmesi, hataların yol açabileceği itibar kaybının engellenmesi, müşterilerin memnun olma oranlarını arttırarak ilerisi için zemin oluşturmak, ürün performansını maksimuma çekebilmek ve ürünü kesintisiz şekilde oluşturabilmek için test yapılır [52].

Yazılım testi; yazılım uygulamalarının risk faktörlerini belirleyebilmek için sınırsız sayıda bulunan çalışma alanından sınırlı ve uygun olan senaryoların seçildiği ve bu senaryolarla gereksinimlerin giderilmeye çalışıldığı bağımsız inceleme faaliyetlerini kapsamaktadır [53].

Yazılım testi türleri statik test ve dinamik test olmak üzere 2'ye ayrılır. Statik test, yazılım kodunun gerçekleştirilmeden önceki safhada proje dokümanlarının kontrolüne dayanan ve projenin gözden geçirilmesini kapsayan süreçtir. Bu süreçte yürütülen testler; proje planlarının incelenmesi, gereksinim dokümanlarının elden geçirilmesi ve hazırlık/yönetim test planlarının incelenmesini kapsamaktadır. Projenin kodlama sürecine geçilmesinden sonra dinamik test türünün kullanımı başlayacaktır. Dinamik test; fonksiyonel test ve fonksiyonel olmayan test

olmak üzere 2 çeşittir [54]. Dinamik test süreçleri içerisinde uygulama test verisinin kullanıldığı her test türünde veri güvenliğinin sağlanması için veri maskeleyen yapılması bir gereklilik ve ihtiyaç olarak ortaya çıkacaktır.

Yazılım testi sürecinin projenin yaşam döngüsüne eklenmesiyle beraber üründe ortaya çıkabilecek hatalar minimum seviyeye indirgenecektir. Bu noktada yazılımın %100 testinin mümkün olmadığı unutulmaması gereken bir noktadır. Belirtilen durumun aksine bir yazılımın %100 testini yapmaya çalışmak, pratik ve maddi olarak mümkün görünmemektedir. Bu açıdan yazılımın istenildiği gibi çalıştığı durumu görüntüleyebilecek az sayıda, fakat bunu gerçekleştirmede yeterli olacak test kümesi dikkatli bir şekilde belirlenmeli ve test süreci bu küme üzerinden yürütülmelidir [52].

Bu kapsamda dikkat edilmesi gereken bazı önemli noktalar bulunmaktadır. Bunlar: Test yapılırken yazılım kesinlikle dinamik olarak çalıştırılmalıdır. Yazılım kritik düzeylerine göre sıralanmalı ve kritik durumda olan, az ve yeterli sayıda test yapılmalıdır. Bunun dışında kalan çalışma alanlarının tamamının testi imkânsız ve gereksiz olacaktır. Test edilecek yazılım için muhtemel riskler dikkate alınarak uygun şekilde seçilen testler gerçekleştirilmesi önemli olacaktır. Yazılım test sürecinde, yazılımın, mantıklı beklentilere ve gereksinimlere karşılık verebilmesi test edilebilmelidir. Profesyonel yazılım projeleri son halleriyle yani canlı ortamda kullanılmaya başlandıkları halleri gibi kalmayarak sürekli bir gelişim ve değişim süreci içinde bulunmaktadır. Yazılımlar içerisinde güncellemelerde bulunulması gerekliliği, ortaya çıkan hataların giderilmesi ve müşterilerin ürün üzerinde ek olarak yapılmasını istedikleri özel geliştirmeler için yazılım güncellemelerinin yapılması ihtimali ortaya çıkmaktadır. Yazılım bu noktada yapılacak güncellemelerden dolayı olumsuz olarak etkilenebilmektedir. Yazılıma eklenen parçaların testlerinin yapılması, bir bütün olarak sistemde oluşabilecek hataların gözden kaçırılmasına sebep olabilmektedir. Bu noktada projelerde, hataları düzeltmek için yapılacak geliştirmelerin işgücü maliyetinin en az üçte biri oranında olacak şekilde işgücü planlaması yapılması önerilmektedir. Yapılan değişikliklerin hızlı bir şekilde cevap verebilmesi ve işgücü maliyetinin düşürülebilmesi açısından testlere programlı bir şekilde devam edilmesi gerekmektedir. Bir yazılım geliştirilirken sürecin en başında karşılaşılan hatanın maliyetiyle, müşteriye teslim edilmesinden sonra bu üründe çıkabilecek hatanın maliyeti arasında Şekil 2.10'da görüldüğü gibi çok büyük fark vardır [55].

Proje Aşaması

Yazılım kullanımdayken (üretimde) tüm maliyetler artar.



Şekil 2.10: Yazılım Hata Maliyetinin Zamansal Gösterimi [55].

Günümüz dünyasında sistemin tümünün kullanış ve performans olarak sorunsuz çalıştığını görebilmek için yapılan manuel testler ciddi manada maliyetli olduklarından, yapılan herhangi bir değişiklikte bu testlerin tekrar edilmesi uygun bir yöntem olmayacaktır. Alternatif test araçlarının kullanımı ve bunun yanında test kodlarının yazılması, yazılım kod kalitesini korumak ve sistemin ileriye dönük olabilmesi için daha doğru bir yol olacaktır. Sistemde yapılan herhangi bir değişikliğin etkilerinin anlaşılabilmesi ve sistemin her zaman çalışır durumda olduğunu kontrol edebilmek için otomatik testler ile ölçüm yapmak doğru bir yöntem olacaktır [56].

2.3.2. Test Verisi Oluşturma

Test çalışması yapılırken harcanan eforun yarısı test verisinin oluşturulabilmesi için harcanır [57]. Test ortamlarında beklentiler dâhilinde uygun test verisi ihtiyacı karşılanmazsa ve yeni test verisi oluşturulmazsa test için harcanan efor miktarı daha fazla artacaktır. Daha önceki test çalışmalarında kullanılan test verisi daha sonra yapılacak testler için kullanılamayacak hale gelecektir. Daha sonrasında yeniden test verisi oluşturulmak istenirse uygun verinin dönüştürülmesi esnasında veri tutarlılığı, bütünlüğü, güvenliği noktasında kısıtlayıcı bazı

faktörler ortaya çıkabilecektir. Buna ek olarak, bu konuda yapılan çalışmalarla test verisinin kalitesi son ürün kalitesini etkileyebilmektedir [58].

2.3.3. Test Verisi Yönetimi

Test veri yönetimi yazılım test sürecinin içerisinde bulunan önemli alt bileşenlerinden bir tanesidir. Test verisinin hazırlanması test içerisinde harcanan zaman ve işgücü anlamında önemli bir paya sahiptir. Test veri yönetiminde, ihtiyaç olan test verisinin az miktarda ve çeşitliliğinin az olduğu durumlarda test veri yönetiminde ciddi sorunlar olmayacaktır, fakat test verisi miktarı ve çeşitliliğinin ciddi manada arttığı durumlarda test verisinin tutarlılığı, bütünlüğü ve güvenliği veri yönetiminde ciddi zorluklar yaşanmasına ve bu durum nedeniyle süreçte olumsuzluklar yaşanmasına sebep olacaktır [6].

Büyük sistemler veya uygulamalar geliştirilirken, şüphesiz test zorluklarıyla karşılaşılır. İş açısından kritik uygulamaları zamanında ve yüksek kalitede sunmak için, pazara sunma süresini hızlandırmak ve maliyetleri düşürmek için güvenilir ve tekrarlanabilir bir test sürecine ihtiyaç vardır. Bu sistemler ve uygulamalar aynı zamanda yüksek düzeyde veri odaklı olma eğilimindedir ve etkin bir şekilde test etmek için büyük miktarda veri gerektirir. Test zorlukları, test planlamasından veri hazırlamaya ve veri korumaya kadar birçok yönü kapsar. Veriye dayalı test zorlukları, artan maliyetlere neden olabilir ve ortaya çıkan uygulamanın kalitesi için önemli bir risk oluşturabilir. Depolama alanlarının hızla kontrolden çıktığı ortamları test etmek için büyük canlı ortam veri tabanlarını kopyalamak maliyet getirecek işlemlerdir. Yani, ihtiyaç duyulacak şey aslında iyi bir test veri yönetimi olacaktır [57].

Proje içerisinde test sürecini, zaman kısıtlılığını yönetebilmek ve koşturulan test aktivitelerinin doğruluğunu sağlamak adına test veri yönetimi büyük önem kazanmaktadır. Ayrıca yüksek kalitede, güvenilir ve tekrarlanabilir test verisiyle bu süreci yönetmek proje sürecini hızlandırdığı gibi maliyetleri de azaltacaktır. Test edenlerin testleri yaparken titiz bir şekilde çalışması ve geçerli testler gerçekleştirmeleri test veri yönetiminin en önemli amaçlarından biridir.

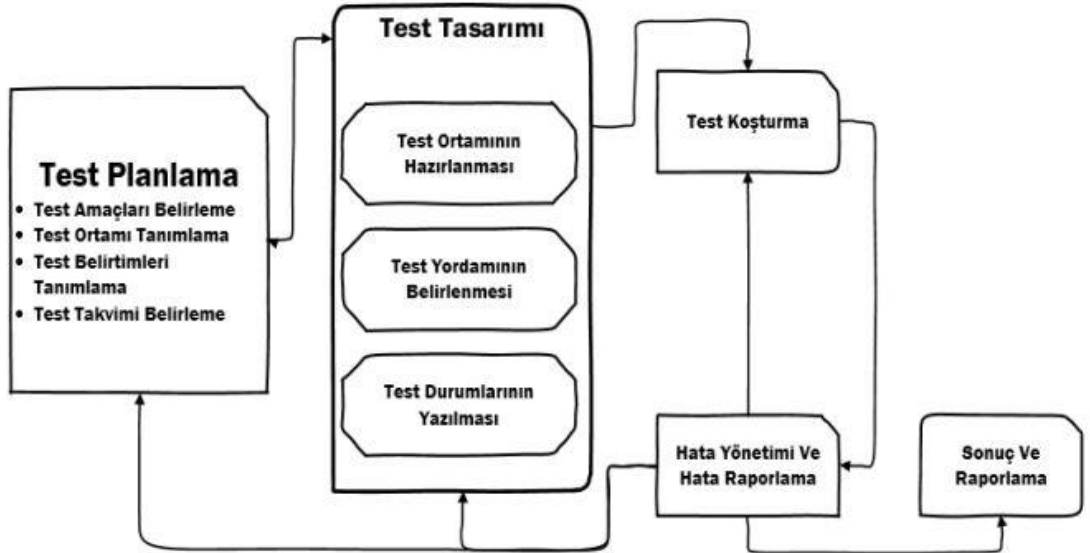
Test verisi yönetiminde özellikle üzerinde durulması gereken konulardan biri de bilgi ve veri güvenliğidir. Test ortamları, saldırılması canlı ortamlarına göre nispeten daha kolay ortamlardır. Çünkü bu ortamlarda yazılımcılar geliştirme yaptıkları için bu ortamlar daha az korunaklı olabilmektedir. Projede test verisinin direk olarak canlı ortamından alındığı

düşünülürse, test ortamına yapılacak herhangi bir sızmada canlı ortam verisi korsanların eline geçecek ve bu durum kötü sonuçlar doğurabilecektir.

Türkiye Yazılım Kalite Raporu [59]'na göre, kaliteli yazılım testi ihtiyacıyla yazılım projelerindeki artış miktarının birbirine paralel bir şekilde değiştiğine yönelik vurgu yapılmıştır. Buna rağmen sektör projeleri incelendiğinde, projelerin %80'inin test veri yönetimi faaliyetlerinin çoğunlukla manuel olarak yapıldığı önemli bir nokta olarak vurgulanmıştır. Bunun aksi yönde test veri yönetimi süreçlerinde otomatik olarak veri üretimi yapan şirketlerin yalnızca %4 olarak kaldığı bildirilmiştir. İlerleyen dönemlerde belirtilen bu %4'lük oranın, test süreçlerinde harcanan eforun azaltılması ve testlerde kalitenin artırılması amaçlarına yönelik olarak ciddi oranda yükseleceğine yönelik bir beklenti olup, bu beklenti rapor içerisinde ön plana çıkartılmıştır.

2.3.4. Test Verisi Yönetimi Yaşam Döngüsü

Test veri yönetimi yaşam döngüsü çoğunluğun yaklaşımına bağlı olarak sırasıyla gereksinimlerin analiz edilmesi, planlama ve tasarım, test verisinin oluşturulması ve test verisinin bakımı aşamalarından oluşmaktadır [57].



Şekil 2.11: Yazılım Test Süreci Adımları [60].

Genel test süreci Şekil 2.11’de gösterilmiştir [60]. Bu sürece göre planlama yapıldıktan sonraki sürecinin devamında sırasıyla tasarım yapılması, testlerin koşturulması, hataların rapor edilmesi, testin sonlandırılması ve belgelendirilmesiyle beraber test sürecinin tüm adımları tamamlanmış olacaktır. Diğer bir söyleyiş şekliyle planlama yapıldıktan sonra testlerin icra edildiği ve elde edilen sonuçların belgelendirilerek rapor oluşturulduğu bir yapıdır. Genel olarak bu sürecin amacı yazılım hatalarının bulunmasıdır.

- Sürecin en başında test için ihtiyaçlar belirlenir ve gereksinimler toplanarak analiz edilir. Bu aşamada kullanılacak veri kaynaklarının seçilmesi, maskeleme yapılacak verinin güvenliği için çok önemlidir. Ayrıca belirlenen test verisi için ihtiyaçlar çerçevesinde depolama ve arşivleme gereksinimleri bu aşama içerisinde belirlenir.
- Analiz yapıldıktan sonra, bu aşama içerisinde çözüm planı ve tasarımı yapılır. Planlama ve tasarım aşaması olarak geçen bu adımda çözümün nasıl bir çalışmayla gerçekleştirileceği yani herhangi bir araç ya da dâhili bir çalışmayla mı yapılacağı belirlenir.
- Bu aşamada, belirlenen çözüm tasarımı devreye girerek ilgili gereksinimler özelinde test verisi oluşturulur.
- Bakım aşamasında oluşturulmuş test verilerinde ortaya çıkabilecek farklı isteklere ve ihtiyaçlara yönelik olarak test yaşam döngüsü kuralları işletilerek test verilerinde değişiklik ya da yenileme yapılır [57].

3. MALZEME VE YÖNTEM

Bu bölümde tez çalışmasında kullanılan araştırma modeli, çalışma sırasında kullanılan verinin oluşturulması, verinin maskelenmesinde kullanılabilecek uygulamalar, bu çalışmada kullanılan veri maskeleyme uygulaması, çalışmaya uygun veri maskeleyme türünün seçimi ve çalışmada kullanılan veri maskeleyme teknikleriyle verinin maskelenmesi konularında bilgiler verilmiştir.

3.1. ÇALIŞMANIN AMACI VE HEDEFİ

Bu çalışmanın amacı, canlı ortam veri tabanlarında bulunan orijinal verinin uygulamaların test süreçlerinde orijinal halleriyle kullanılmalarını engellemek için canlı ortam veri tabanındaki verinin statik veri maskeleyme yöntemi ve veri tiplerine uygun olarak kullanılan veri maskeleyme teknikleriyle maskelenerek test ortam veri tabanında oluşturulmasını göstermektir. Ayrıca veri tabanında bulunan veri boyutuna göre maskeleyme işleminin zamansal maliyeti artacağı için bu kapsamda performansın nasıl artırılabilirliği yani zamansal maliyetin nasıl düşürülebileceği konusunda bilgi verilecektir. Çalışmanın sonucunda veri maskeleyme işleminin kişisel verinin korunmasına yönelik kullanılabilir bir yöntem olduğunun gösterilebilmesi hedeflenmektedir.

3.2. ARAŞTIRMANIN MODELİ

Bu araştırma kapsamında canlı ortam veri tabanında, tablolarda tutulan kişisel veri setlerinin örnek olarak maskelenmemiş, ham şekilde oluşturulması yani canlı ortam veri tabanına benzeyen bir veri tabanının oluşturulması, daha sonrasında oluşturulan kayıtların lisanslı bir veri maskeleyme yazılımı olan *CopyCat* uygulaması üzerinden statik veri maskeleyme yöntemiyle ve tablolarda oluşturulan veri tiplerine göre maskeleyme teknikleri kullanılarak test ortam veri tabanına aktarılması gösterilmiştir.

3.3. ARAŞTIRMADA KULLANILAN YAZILIM ARACI SEÇİMİ

Bu çalışmada yapılması planlanan canlı ortam veri tabanında bulunan veri setlerinin statik veri maskeleyme yöntemi kullanılarak maskelenen verinin test ortam veri tabanında oluşturulması işlemi için birçok farklı uygulama ve veri tabanı şirketlerinin maskeleyme için kendi veri tabanları özelinde oluşturdukları uygulama yöntemleri bulunmaktadır. Araştırmada kullanılacak yazılım seçiminde özellikle platform bağımsızlığının olması, çoklu kurulum, iş ve

iş parçacığı desteklenmesi gibi özellikler uygulamalar arasında ayırt edici olmuş ve uygulamanın belirlenmesinde etkili olmuştur. Araştırma kapsamında Veriskop şirketinin lisanslı bir ürünü olan *CopyCat* programı kullanılmıştır [61]. Tablo 3.1’de araştırma kapsamında kullanılabilecek maskeleye uygulamalarının bazıları özellikleri bakımından karşılaştırılmıştır.

Tablo 3.1: Maskeleye Uygulamalarının Karşılaştırılması.

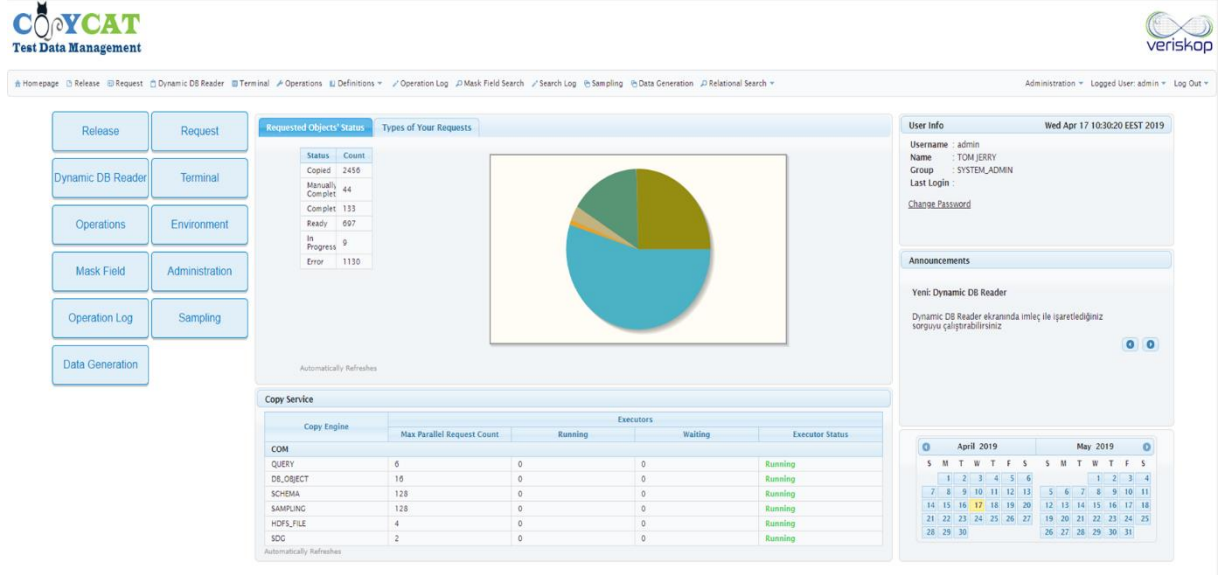
Özellikler/Uygulamalar	Veriskop - CopyCat	Informatica - TDM	IBM - Optim	Mayatech - Maya Veri Maskeleye Çözümü
Maskeleye algoritması ve kuralları için verdiği destek	Sınırsız	Kısıtlı	Kısıtlı	Kısıtlı
Kullanıcı rol yönetimi desteği	Var	Var	Var	Yok
Platform bağımsızlığı	Var	Sınırlı	Sınırlı	Yok
Çoklu kurulum, iş ve iş parçacığı desteği	Var	Yok	Var	Yok

Araştırmada kullanılabilecek yazılım araçları incelendiğinde özellikle veri tabanı şirketlerinin kendi platformlarına özel olarak, yani ortam bağımlı olarak oluşturdukları veri maskeleye araçları ortaya çıkmaktadır. Bunlar; Oracle’ın veri maskeleye aracı [62], PostgreSQL’in veri maskeleye aracı [63], Microsoft’un veri maskeleye aracı [64] vb. gibi veri tabanı şirketlerinin oluşturmuş oldukları veri maskeleye araçlarında platform bağımlı bir yaklaşım tercih edilmiştir. IBM’in veri maskeleye aracı olan Optim[65] aracındaysa platform bağımlı bir yaklaşım tercih edilmemiştir. Ayrıca birçok şirket tarafından kullanılmaktadır. Bunlar haricinde Türkiye’de Informatica şirketinin test veri yönetimi aracı [66], Mayatech firmasının Maya veri maskeleye çözümü [67], Infobox şirketinin Infobox statik veri maskeleye yazılımı [68], Krontech firmasının maskeleye çözümleri [69], Çalışkan Group’un veri maskeleye yazılımı [70] vb. yazılım araçları bulunmaktadır. Bu alternatif olarak kullanılabilecek yazılım araçlarının hepsi lisanslı ve kullanımları ücretlidir.

3.3.1. Araştırmada Kullanılan Yazılım Aracı

Araştırma kapsamında kullanılan *CopyCat*, yapısal (veri tabanı sistemleri) ve yapısal olmayan (HDFS, AB initio dosya sistemi) veri sistemleri arasında maskeli olarak veri taşıma ve test

verisi oluşturmaya yarayan bir veri maskeleyesi çözümüdür. Pazarın önde gelen ve çok kullanılan veri tabanı sistemlerinde maskeli olarak veri kopyalama, arşivleme, silme ve örnekleme imkânı sunar. *CopyCat*, Java programlama dilinde yazılmış bir uygulamadır [61]. Şekil 3.1’de uygulamanın arayüzü gösterilmiştir. Uygulama arayüzü İngilizce olup Türkçe arayüzü bulunmamaktadır.



Şekil 3.1: CopyCat Uygulama Arayüzü.

Uygulamanın fonksiyonel özellikleri şunlardır [61]:

- Kolay ve hızlı kurulum.
- Web tabanlı yetki bazlı kolay kullanım arayüzüne sahiptir.
- Yapısal sistemlerde ve yapısal olmayan dosyalarda veri maskeleye (*on the fly*) yapılabilir.
- Esnek, özelleştirilebilir kural ve maskeleye yöntemi geliştirme altyapısına sahiptir.
- Özelleştirilmiş hassas veri keşfi modülü içerir.
- Kopyalama/Maskeleye işlemleri anlık veya zamanlamalı olarak yapılabilir.
- Kopyalama/Maskeleye kapsamı belirleme özelliğine sahiptir.
- Yapısal veri sistemlerinde örnekleme yaparak büyük veri setlerinden maskeli küçük setler oluşturulmasına olanak sağlar.
- Yapısal veri sistemlerinde örneklenen verinin arşivlenme ve silinmesine olanak sağlar.
- Yapısal veri sistemlerinde sentetik veri üretimi desteği sunar.

- Yapısal veri sistemlerinde yerinde maskeleye (inline) desteği sunar.
- Test Veri Yönetimi ile kaynak ve hedef sistemler arası:
 - Metadata senkronizasyonu gerçekleştirir.
 - Sorgu bazlı maskeli veri taşıma yapar.
 - Tablo verisini maskeli taşımaya imkân sağlar.
 - Şema (Schema) altındaki tüm objelerin taşınmasını yapar.
 - Örnekleme (Sampling) ile maskeli küçük veri setleri oluşturulmasına imkân verir.
- Sentetik Veri Oluşturma ile hedef yapısal sistemlerde:
 - Var olan tabloya, belirlenen sayıda rastgele veri oluşturur.
 - Yeni tabloya, belirlenen sayıda rastgele veri oluşturur.
 - İlişkisel tablolara, belirlenen sayıda rastgele ilişkili veri oluşturur.
- Hassas Veri Keşfi ile hedef yapısal sistemlerde:
 - Kolon adı benzerliğine göre hassas veri keşfi gerçekleştirir.
 - Veri içeriğini analiz ederek hassas veri keşfi gerçekleştirir.
- Platformlar Arası Veri Taşıma ile istenilen kaynak ve hedef platformlar arası veri taşıma gerçekleştirebilir:
 - Oracle → SQL Server
 - Oracle → Hadoop
 - Oracle → Hive
 - SQL Server → Oracle
 - Hadoop → Oracle
- Dinamik Veri Sorgulama ile yapısal sistemlerinizde anlık veri sorgulama yapabilir.
 - Sorgu bloklama özelliği ile sadece istenilen türde SQL ifadelerinin çalıştırılmasına imkân sunar.
 - Tüm sorgu sonuçları belirlenen kurallar çerçevesinde maskeli olarak gösterilir.

Uygulamanın teknik özellikleri şunlardır [61]:

- Çoklu kurulum, iş ve iş parçacığı desteklenmektedir.
- Yüksek erişilebilirlik (*high availability*) ve yük paylaşımı (*load balancing*) için yatay mimariye sahiptir.
- LDAP entegrasyonu sayesinde kurumsal kimlik doğrulama ile giriş imkânı sunar.
- Desteklenen ilişkisel veri tabanları:

- Oracle
- SQL Server
- PostgreSQL
- Teradata
- Apache HIVE
- Desteklenen dosya türleri:
 - AB initio dosya sistemi: dat, dat.gz
 - HDFS: basit metin dosya formatı
 - Hive → Oracle

3.4.VERİNİN ELDE EDİLMESİ VE HAZIRLANMASI

Araştırma kapsamında kullanılan uygulama içeriğinde test verisini oluşturmak için örnek veri havuzları bulunmaktadır. Bu veri havuzlarından, canlı ortam veri tabanında bulunan orijinal veri yerine format ve tip olarak benzer şekilde test verisi oluşturularak maskeleye işlemi bu veriyle yapılmıştır. Maskeleye işlemi canlı ortam veri tabanında bulunan, şemalar altındaki tabloların içerisinde bulunan veri alanlarına uygulanacaktır. Çalışmada ilişkisel bir veri tabanı kullanılmış olup birbiriyle ilişkili MUSTERILER ve SOZLESME adında tablolar oluşturulmuştur. Bu kapsamda maskelenecek olan örnek tabloların içerisinde yer alan veri alanları aşağıda gösterilmiştir.

MUSTERILER tablosu içeriği:

- MUSTERI_NO
- ADRES1
- ADRES2
- ADRES3
- ANNE_ADI
- AD
- SOYAD
- ANNE_KIZLIK_SOYADI
- BABA_ADI
- TELEFON_NUMARASI
- MUSTERI_TEMEL_KIMLIK_NO
- EPOSTA
- DURUM
- DOGUM_TARIHI

- BASLANGIC_TARIHI
- AD_ILK_HARF
- SOYAD_ILK_HARF
- AD_SOYAD_ILK_HARF

SOZLESME tablosu içeriği:

- SOZLESME_NO
- MUSTERI_NO
- BASLANGIC_TARIHI
- BITIS_TARIHI
- DURUM
- EPOSTA
- TELEFON_NUMARASI

CopyCat uygulamasının mevcut sürümü Oracle, SQL Server, PostgreSQL, Teradata, Apache HIVE ilişkisel veri tabanlarını desteklemektedir [61]. Canlı ortam veri tabanını örneklemek için Oracle veri tabanı kullanılmıştır.

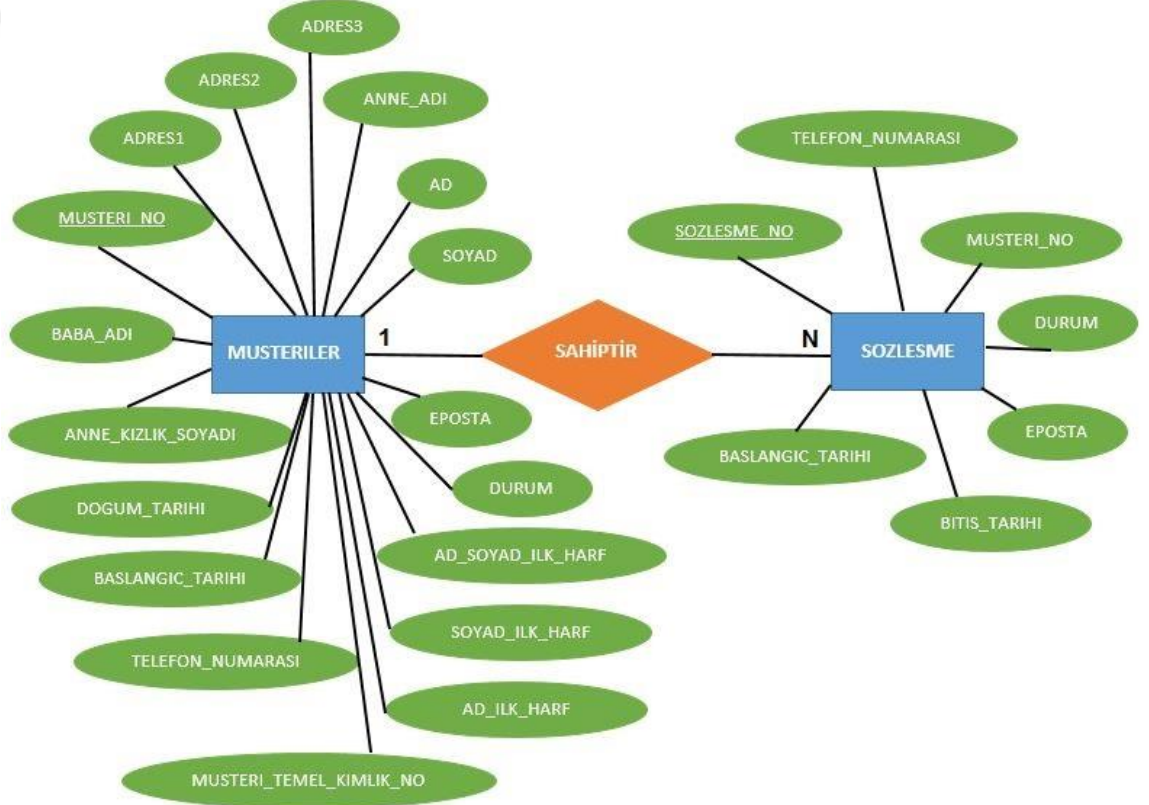
Column Name	#	Type	Type Mod	Not Null	Default	Comment
ABC ADRES1	1	VARCHAR2(200)		☐		
ABC ADRES2	2	VARCHAR2(200)		☐		
ABC ADRES3	3	VARCHAR2(200)		☐		
ABC ANNE_ADI	4	VARCHAR2(100)		☐		
ABC AD	5	VARCHAR2(100)		☐		
ABC SOYAD	6	VARCHAR2(100)		☐		
ABC ANNE_KIZLIK_SOYADI	7	VARCHAR2(100)		☐		
ABC BABA_ADI	8	VARCHAR2(100)		☐		
123 TELEFON_NUMARASI	9	NUMBER		☐		
123 MUSTERI_NO	10	NUMBER		☒		
123 MUSTERI_TEMEL_KIMLIK_NO	11	NUMBER		☐		
ABC EPOSTA	12	VARCHAR2(200)		☐		
ABC DURUM	13	VARCHAR2(20)		☐		
DOGUM_TARIHI	14	DATE		☐		
BASLANGIC_TARIHI	15	DATE		☐		
ABC AD_ILK_HARF	16	VARCHAR2(1)		☐		
ABC SOYAD_ILK_HARF	17	VARCHAR2(1)		☐		
ABC AD_SOYAD_ILK_HARF	18	VARCHAR2(2)		☐		

Şekil 3.2: MUSTERILER Tablo İçeriği.

Column Name	#	Type	Type Mod	Not Null	Default	Comment
123 SOZLESME_NO	1	NUMBER(22,0)		☒		
123 MUSTERI_NO	2	NUMBER(22,0)		☐		
BASLANGIC_TARIHI	3	DATE		☐		
BITIS_TARIHI	4	DATE		☐		
ABC DURUM	5	VARCHAR2(100)		☐		
ABC EPOSTA	6	VARCHAR2(100)		☐		
123 TELEFON_NUMARASI	7	NUMBER(38,0)		☐		

Şekil 3.3: SOZLESME Tablo İçeriği.

Şekil 3.2’de örnek tablo olarak kullanılan MUSTERILER tablosunun veri tabanındaki içeriği gösterilmiştir. Şekil 3.3’te örnek tablo olarak kullanılan SOZLESME tablosunun veri tabanındaki içeriği gösterilmiştir. Çalışmada canlı ortam veri tabanı olarak “1.7” isimindeki veri tabanı kullanılmıştır. Veri tabanı olarak ilişkisel veri tabanı kullanılmış olup, COPYCATLIVE şeması altında oluşturulmuş olan MUSTERILER tablosunun birincil anahtarı MUSTERI_NO alanıdır. Yine aynı şekilde COPYCATLIVE şeması altında oluşturulmuş olan SOZLESME tablosunun birincil anahtarı SOZLESME_NO alanı olup, ikincil anahtarı MUSTERI_NO alanıdır.



Şekil 3.4: Varlık İlişki Diyagramı.

MUSTERILER ve SOZLESME tabloları birbirleriyle ilişkili tablolar olup, MUSTERI_NO alanıyla birbirlerine bağlı tablolardır. Tablolar arasında 1’e N ilişki vardır yani MUSTERILER tablosundaki her müşteri kaydının SOZLESME tablosunda 1 ya da 1’den fazla ilişkili olduğu kayıt bulunur. Şekil 3.4’te bu tablolar için Varlık İlişki Diyagramı (Entity Relationship Diagram - ERD) gösterilmiştir.

	ABC ADRES1	ABC ADRES2	ABC ADRES3	ABC ANNE_AD1	ABC AD	ABC SOYAD	ABC ANNE_KIZLIK_SOYADI	ABC BABA_AD1	123 TELEFON_NUMARASI	123 MUSTERI_NO
1	A sokak	B caddesi	Lüleburgaz	Maral	Kadir	Tansen	Lamiye	Gizem	5,320,000,181	181
2	A sokak	B caddesi	Yalıhüyük	Janseli	Şebnem	Abir	Latife	Ilgaz	5,320,000,182	182
3	A sokak	B caddesi	Hopa	Erna	Gökhan	Tokgöz	Rayihan	Emin	5,320,000,183	183
4	A sokak	B caddesi	Tavşanlı	Gulnisa	Burcu	Atalan	Lacerem	Yağmur	5,420,000,184	184
5	A sokak	B caddesi	Merzifon	Sinem	Adnan	Taştan	Rebab	Burcu	5,350,000,185	185
6	A sokak	B caddesi	Bitlis Merkez	Lila	Mehmet furkan	Abdülvel	Lema	Candemir	5,320,000,186	186
7	A sokak	B caddesi	Pazar / Rize	Nazgül	Seda	Temcit	Rasime	Ziya	5,550,000,187	187
8	A sokak	B caddesi	Kırıkkale Merkez	Güz	Cem atakan	Taciser	Revzen	Vahdet	5,320,000,188	188
9	A sokak	B caddesi	Eruh	Ilm	Barış	Asile	Rıfki	Deran	5,550,000,189	189
10	A sokak	B caddesi	Çermik	Bilge	Esra	Tunçaslan	Lisan	Vedat	5,550,000,190	190
11	A sokak	B caddesi	Mihalgazi	Afife	Gökrem	Azadi	Rafet	Raif	5,050,000,191	191
12	A sokak	B caddesi	Alaplı	Efil	Ayşe gül	Ammar	Rical	Rukiye	5,420,000,192	192
13	A sokak	B caddesi	Amasra	Ruhgöl	Muhammed yusuf	Aymutlu	Radiye	Şengül	5,550,000,193	193
14	A sokak	B caddesi	Yakakent	Esim	Deniz	Akman	Risale	Suat	5,050,000,194	194
15	A sokak	B caddesi	Acıgöl	Meral	Ayşe	Tesliye	Ruhsar	Yusuf	5,350,000,195	195
16	A sokak	B caddesi	Sumbas	Nurten	Serkan	Alahan	Rebi	Zikri	5,350,000,196	196
17	A sokak	B caddesi	Pertek	Beliz	Ülkühan	Asfer	Lalegun	Hamit	5,050,000,198	198
18	A sokak	B caddesi	Babadağ	Canan	Umut can	Akata	Rabih	Alp	5,050,000,199	199
19	A sokak	B caddesi	Saimbeyli	Caıdan	Öznur	Taylaner	Rakime	Gökalp	5,320,000,200	200
20	A sokak	B caddesi	Gümüşhacıköy	Bike	Hanife tuğçe	Asılgül	Riayet	Fethi	5,050,000,201	201
21	A sokak	B caddesi	Arsin	Gülñihal	Emre	Asif	Rakip	Özalp	5,350,000,202	202
22	A sokak	B caddesi	Çermik	Dilan	Tuba	Aykün	Rezan	Paşa	5,420,000,203	203
23	A sokak	B caddesi	Altıntaş	Asya	Sena	Alişin	Rumeyşa	Yusuf	5,350,000,204	204
24	A sokak	B caddesi	Göıdes	Aydonat	Mehmet hilmi	Akcan	Revan	Halit	5,320,000,205	205
25	A sokak	B caddesi	Kargı	Soneda	Mehmet	Akkerman	Rekanet	Ilyas	5,550,000,206	206
26	A sokak	B caddesi	Tunceli Merkez	Anil	Taner	Andak	Reşad	Demirkan	5,550,000,207	207
27	A sokak	B caddesi	Polatlı	Birgen	Veysel	Aysan	Layık	Mesut	5,350,000,208	208
28	A sokak	B caddesi	Orhangazi	Çiler	Semra	Tokuztuğ	Reşit	Kamer	5,550,000,209	209
29	A sokak	B caddesi	Bismil	Havva	Neslihan	Ahavi	Rügan	Sancar	5,550,000,210	210
30	A sokak	B caddesi	Ayaş	Gökçe	Metin	Abdulhalim	Ruhsel	Vural	5,550,000,211	211

Şekil 3.5: MUSTERILER Tablosu Veri Seti 1.Kısım.

	123 MUSTERI_TEMEL_KIMLIK_NO	ABC EPOSTA	ABC DURUM	DOGUM_TARIHI	BASLANGIC_TARIHI	ABC AD_ILK_HARF	ABC SOYAD_ILK_HARF	ABC AD_SOYAD_ILK_HARF
1	1,000,000,181	kadir.tansen@gmail.com	ACTIVE	1992-12-26 01:40:23	2005-05-31 01:40:23	K	T	KT
2	1,000,000,182	şebnem.abir@hotmail.com	ACTIVE	2001-12-21 01:40:23	2009-11-27 01:40:23	Ş	A	ŞA
3	1,000,000,183	gtokgoz@gmail.com	ACTIVE	2006-10-07 01:40:23	2012-04-20 13:40:23	G	T	GT
4	1,000,000,184	burcu@msn.com	ACTIVE	2012-05-20 01:40:23	2015-02-10 13:40:23	B	A	BA
5	1,000,000,185	adnan.tashtan@i2i-systems.com	ACTIVE	2007-03-31 01:40:23	2012-07-17 01:40:23	A	T	AT
6	1,000,000,186	mehmet.furkan.abdullevele@i2i-systems.com	ACTIVE	1979-05-21 01:40:23	1998-08-12 01:40:23	M	A	MA
7	1,000,000,187	seda@gmail.com	ACTIVE	1991-12-08 01:40:23	2004-11-20 01:40:23	S	T	ST
8	1,000,000,188	cem.atakan.t@outlook.com	ACTIVE	2016-12-29 01:40:23	2017-06-01 13:40:23	C	T	CT
9	1,000,000,189	barış@i2i-systems.com	ACTIVE	1974-03-28 01:40:23	1996-01-15 01:40:23	B	A	BA
10	1,000,000,190	esra.tuncaslan@outlook.com	ACTIVE	1978-11-02 01:40:23	1998-05-04 01:40:23	E	T	ET
11	1,000,000,191	g.azadi@hotmail.com	ACTIVE	1968-01-07 01:40:23	1992-12-05 01:40:23	G	A	GA
12	1,000,000,192	ayşe.gul.ammar@msn.com	ACTIVE	2005-08-01 01:40:23	2011-09-17 13:40:23	A	A	AA
13	1,000,000,193	muhammed.yusuf.aymutlu@i2i-systems.com	ACTIVE	1989-04-13 01:40:23	2003-07-24 13:40:23	M	A	MA
14	1,000,000,194	deniza@outlook.com	ACTIVE	1977-06-13 01:40:23	1997-08-23 13:40:23	D	A	DA
15	1,000,000,195	ayşe@hotmail.com	ACTIVE	1987-08-31 01:40:23	2002-10-02 01:40:23	A	T	AT
16	1,000,000,196	s.alahana@i2i-systems.com	ACTIVE	1997-01-30 01:40:23	2007-06-18 01:40:23	S	A	SA
17	1,000,000,198	ulkuhana.a@msn.com	ACTIVE	1970-04-05 01:40:23	1994-01-18 13:40:23	U	A	UA
18	1,000,000,199	umut.can.a@gmail.com	ACTIVE	1999-04-08 01:40:23	2008-07-21 01:40:23	U	A	UA
19	1,000,000,200	oznur.t@msn.com	ACTIVE	1991-11-30 01:40:23	2004-11-16 01:40:23	Ö	T	ÖT
20	1,000,000,201	hanife.tuğçe@outlook.com	ACTIVE	1981-08-21 01:40:23	1999-09-27 13:40:23	H	A	HA
21	1,000,000,202	emre.asif@msn.com	ACTIVE	1986-01-09 01:40:23	2001-12-06 13:40:23	E	A	EA
22	1,000,000,203	taykun@gmail.com	ACTIVE	1988-08-30 01:40:23	2003-04-02 13:40:23	T	A	TA
23	1,000,000,204	s.alışin@msn.com	ACTIVE	1978-03-11 01:40:23	1998-01-06 01:40:23	S	A	SA
24	1,000,000,205	mehmet.hilmi@i2i-systems.com	ACTIVE	1963-06-10 01:40:23	1990-08-22 01:40:23	M	A	MA
25	1,000,000,206	mehmet@msn.com	ACTIVE	1977-08-12 01:40:23	1997-09-22 13:40:23	M	A	MA
26	1,000,000,207	taner@gmail.com	ACTIVE	2002-12-07 01:40:23	2010-05-21 13:40:23	T	A	TA
27	1,000,000,208	veyse.laysan@gmail.com	ACTIVE	1989-02-21 01:40:23	2003-06-29 01:40:23	V	A	VA
28	1,000,000,209	semra@msn.com	ACTIVE	1966-07-06 01:40:23	1992-03-05 01:40:23	S	T	ST
29	1,000,000,210	neslihan@gmail.com	ACTIVE	1994-06-04 01:40:23	2006-02-17 13:40:23	N	A	NA
30	1,000,000,211	metin.abdulhalim@outlook.com	ACTIVE	1993-05-03 01:40:23	2005-08-03 01:40:23	M	A	MA

Şekil 3.6: MUSTERILER Tablosu Veri Seti 2.Kısım.

Şekil 3.5 ve Şekil 3.6’da “1.7” isimindeki veri tabanı, COPYCATLIVE şeması altında oluşturulan ve veri maskeleme işleminde örnek olarak kullanılacak olan MUSTERILER tablosunun içerisinde bulunan test verisinden 30 kayıt temsili olarak gösterilmiştir. MUSTERILER tablosunun içerisinde 1 milyon müşteri için kayıt oluşturulmuştur. Tablo içeriğinde bulunan çoğu alanda müşterilerin kişisel verisi bulunmaktadır. Geri kalan alanlarda ise sistemde kullanılacak olan müşteriye özel veri tanımlanmıştır.

	123 SOZLESME_NO	123 MUSTERI_NO	BASLANGIC_TARIHI	BITIS_TARIHI	ABD DURUM	ABD EPOSTA	123 TELEFON_NUMARASI
1	3,773,947	2,384	2016-02-12 01:40:23	2016-02-12 01:40:23	ACTIVE	pinar.a@gmail.com	5,050,002,384
2	3,773,948	2,385	2014-09-21 13:40:23	2014-09-21 13:40:23	ACTIVE	tuba.akin@outlook.com	5,320,002,385
3	3,773,949	2,386	1996-03-29 01:40:23	1996-03-29 01:40:23	ACTIVE	tolga.akbek@gmail.com	5,050,002,386
4	3,773,950	2,388	2006-08-17 01:40:23	2006-08-17 01:40:23	ACTIVE	d.tohum@msn.com	5,050,002,388
5	3,773,951	2,389	2012-05-10 13:40:23	2012-05-10 13:40:23	ACTIVE	senan@outlook.com	5,550,002,389
6	3,773,952	2,390	1995-05-10 01:40:23	1995-05-10 01:40:23	ACTIVE	seyhan.turhan@gmail.com	5,420,002,390
7	3,773,953	2,391	2015-11-01 13:40:23	2015-11-01 13:40:23	ACTIVE	b.arif@msn.com	5,050,002,391
8	3,773,954	2,393	1996-07-26 13:40:23	1996-07-26 13:40:23	ACTIVE	e.akgoze@msn.com	5,320,002,393
9	3,773,955	2,394	2003-04-19 01:40:23	2003-04-19 01:40:23	ACTIVE	o.aydinbay@hotmail.com	5,350,002,394
10	3,773,956	2,395	1993-04-06 01:40:23	1993-04-06 01:40:23	ACTIVE	f.aramdil@i2i-systems.com	5,420,002,395
11	3,773,957	2,396	1994-04-03 13:40:23	1994-04-03 13:40:23	ACTIVE	mustafa.akal@i2i-systems.com	5,320,002,396
12	3,773,958	2,397	1994-05-25 13:40:23	1994-05-25 13:40:23	ACTIVE	ferhan.akcali@msn.com	5,320,002,397
13	3,773,959	2,398	2002-03-30 01:40:23	2002-03-30 01:40:23	ACTIVE	salih@hotmail.com	5,550,002,398
14	3,773,960	2,399	2000-12-06 01:40:23	2000-12-06 01:40:23	ACTIVE	nurdan.aybar@i2i-systems.com	5,050,002,399
15	3,773,961	2,401	2014-01-09 01:40:23	2014-01-09 01:40:23	ACTIVE	a.abdulkebir@i2i-systems.com	5,550,002,401
16	3,773,962	2,402	1999-11-25 01:40:23	1999-11-25 01:40:23	ACTIVE	f.aydogmus@i2i-systems.com	5,420,002,402
17	3,773,963	2,403	2004-04-21 13:40:23	2004-04-21 13:40:23	ACTIVE	p.aysan@hotmail.com	5,550,002,403
18	3,773,964	2,405	2013-07-10 13:40:23	2013-07-10 13:40:23	ACTIVE	mustafa.asim.tevil@gmail.com	5,550,002,405
19	3,773,965	2,406	1994-02-13 01:40:23	1994-02-13 01:40:23	ACTIVE	eren.aygule@outlook.com	5,320,002,406
20	3,773,966	2,407	2000-03-27 01:40:23	2000-03-27 01:40:23	ACTIVE	zeynep@outlook.com	5,320,002,407
21	3,773,967	2,408	2000-12-10 01:40:23	2000-12-10 01:40:23	ACTIVE	e.asiye@outlook.com	5,320,002,408
22	3,773,968	2,409	2005-10-20 01:40:23	2005-10-20 01:40:23	ACTIVE	cansu@i2i-systems.com	5,050,002,409
23	3,773,969	2,410	2000-10-01 01:40:23	2000-10-01 01:40:23	ACTIVE	irem.akdamar@i2i-systems.com	5,550,002,410
24	3,773,970	2,411	2006-07-20 01:40:23	2006-07-20 01:40:23	ACTIVE	jale.t@outlook.com	5,550,002,411
25	3,773,971	2,412	2016-04-27 13:40:23	2016-04-27 13:40:23	ACTIVE	h.acunalp@msn.com	5,350,002,412
26	3,773,972	2,413	1996-06-18 01:40:23	1996-06-18 01:40:23	ACTIVE	kubra.a@gmail.com	5,350,002,413
27	3,773,973	2,414	2012-10-28 01:40:23	2012-10-28 01:40:23	ACTIVE	m.turmerdem@i2i-systems.com	5,550,002,414
28	3,773,974	2,415	2007-12-03 13:40:23	2007-12-03 13:40:23	ACTIVE	huseyin.atasoy@i2i-systems.com	5,550,002,415
29	3,773,975	2,416	1999-02-04 01:40:23	1999-02-04 01:40:23	ACTIVE	serhat.arica@msn.com	5,320,002,416
30	3,773,976	2,418	2000-03-29 01:40:23	2000-03-29 01:40:23	ACTIVE	ozlem.a@msn.com	5,320,002,418

Şekil 3.7: SOZLESME Tablosu Veri Seti.

Şekil 3.7’de “1.7” isimdeki veri tabanı, COPYCATLIVE şeması altında oluşturulan, MUSTERILER tablosuyla ilişkili olan ve örnek olarak kullanılacak olan SOZLESME tablosunun içerisinde bulunan test verisinden 30 kayıt temsili olarak gösterilmiştir. Bu iki tabloda oluşturulan veri setleri MUSTERI_NO alanı ortak olacak şekilde tablolar ilişkilendirilerek oluşturulmuştur. MUSTERILER tablosunda olduğu gibi SOZLESME tablosunun içerisinde de 1 milyon kayıt bulunmaktadır.

3.5. VERİ MASKELEME YÖNTEMİNİN SEÇİLMESİ

Çalışma kapsamında canlı ortam veri tabanını örneklemek için ilişkisel bir veri tabanı içerisinde oluşturulan COPYCATLIVE şeması altında MUSTERILER ve SOZLESME isminde birbiriyle ilişkili örnek tablolar oluşturulmuştur. Bu tablolar arasında MUSTERI_NO alanıyla ilişki kurulmuştur.

Canlı ortam veri tabanında bulunan veriyi maskelemek ve test ortam veri tabanına aktarımını sağlamak için statik veri maskeleme yöntemi kullanılacaktır. Statik veri maskeleme, önemli verinin orijinal veri tabanı ortamında maskelenmesi sürecini ifade eder. İçerik bir test ortamına kopyalanır ve daha sonra üçüncü taraf satıcılar veya diğer gerekli taraflar arasında paylaşılabılır [2]. Veri uygun maskeleme teknikleriyle maskelendikten sonra test ortam veri tabanına taşınarak maskelenmiş halleriyle gösterilmiştir. Çalışmanın akış şeması Şekil 3.8’deki gibidir.



Şekil 3.8: Çalışma Akış Şeması.

Veri setleri hazırlandıktan sonra Oracle veri tabanında tutulan bu veri setleri içerisindeki alanlara özel olarak uygulamada mevcut olan uygun maskeleye teknikleri seçilmiştir.

3.6. UYGULANACAK MASKELEME TEKNİKLERİNİN BELİRLENMESİ

Çalışma kapsamında oluşturulan veri setleri içerisindeki alanlara *CopyCat* uygulamasında yer alan maskeleye teknikleri tanımlanmıştır.

The screenshot shows the 'Mask Field (ESC to Cancel)' dialog box. It contains the following fields and options:

- Mask Type:** Email Masking
- Field Name:** eposta
- Table:** All
- Environment:** All
- Important information about mask definitions:** Click here to read!
- Parameters:**
 - LOCAL MASK TYPE:** STAR_MASKING
 - DOMAIN MASK TYPE:** COMPOSIT_MASKING
 - Mask Local?:** ☒
 - Mask Domain?:** ☐
 - Local Percent:** 50
- Save:**

Şekil 3.9: E-posta Maskeleye Tekniği.

Şekil 3.9’da MUSTERILER ve SOZLESME tablolarında bulunan EPOSTA alanı için uygulamada bulunan E-posta Maskeleye (*Email Masking*) tekniği tanımlanmıştır. Bu maskeleye tekniğinde e-posta verisinde olması zorunlu olan “@” işaretinden önce gelen karakterlerin %50 oranındaki kısmı “*” karakteriyle maskelenmiştir.

Mask Field (ESC to Cancel)

Mask Type: Digital Masking

Field Name: telefon_numarasi

Table: All

Environment: All

Important information about mask definitions.
[Click here to read!](#)

Parameters

Number of digits to mask ? 3

Keep number unmasked? ☒

Save

Şekil 3.10: Dijital Maskeleye Tekniği.

Şekil 3.10’da MUSTERILER ve SOZLESME tablolarında bulunan TELEFON_NUMARASI alanı için uygulamada bulunan Dijital Maskeleye (*Digital Masking*) tekniği tanımlanmıştır. Bu maskeleye tekniğinde TELEFON_NUMARASI alanında bulunan verinin sağdan 3 hanesi sayısal olarak başka bir rakamla değiştirilerek maskelenmiştir.

Mask Field (ESC to Cancel)

Mask Type: Date Masking

Field Name: dogum_tarihi

Table: All

Environment: All

Important information about mask definitions.
[Click here to read!](#)

Parameters

Start date: 01 01 1990

End date: 01 01 2021

Mask Hours? ☒

Mask Year? ☐

Mask Month? ☒

Mask Day? ☒

Save

Şekil 3.11: Tarih Maskeleye Tekniği.

Şekil 3.11’de MUSTERILER tablosunda bulunan DOGUM_TARIHI alanı için uygulamada bulunan Tarih Maskeleye (*Date Masking*) tekniği tanımlanmıştır. Bu maskeleye tekniğinde

DOGUM_TARIHI alanında bulunan verinin 01.01.1990'dan sonra ve 01.01.2021'den önce olan tarih değerleriyle; saat, gün ve ay kısımları uygun değerlerle değiştirilerek maskelenmiştir. DOGUM_TARIHI verisinin yıl kısmında maskeleye yapılmamıştır.

The screenshot shows a 'Mask Field (ESC to Cancel)' dialog box. It has the following fields: 'Mask Type' set to 'Masking By Hash Algorithm', 'Field Name' set to 'anne_kizlik_soyadi', 'Table' set to 'All', and 'Environment' set to 'All'. Below these fields is a red text warning: 'Important information about mask definitions. Click here to read!'. At the bottom, there is a 'Parameters' section with 'Hashing Algorithm' set to 'SHA-256'. A 'Save' button is at the bottom right.

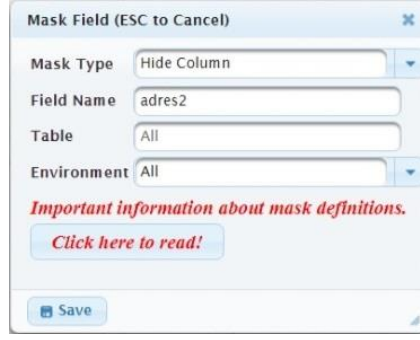
Şekil 3.12: Hash Algoritmasıyla Maskeleye Tekniği.

Şekil 3.12'de MUSTERILER tablosunda bulunan ANNE_KIZLIK_SOYADI alanı için uygulamada bulunan Hash Algoritmasıyla Maskeleye (Masking By Hash Algorithm) tekniği tanımlanmıştır. Bu maskeleye tekniğinde ANNE_KIZLIK_SOYADI alanında bulunan veri SHA-256 Hashing Algoritmasıyla maskelenmiştir.

The screenshot shows a 'Mask Field (ESC to Cancel)' dialog box. It has the following fields: 'Mask Type' set to 'Simple Star Masking', 'Field Name' set to 'adres3', 'Table' set to 'All', and 'Environment' set to 'All'. Below these fields is a red text warning: 'Important information about mask definitions. Click here to read!'. At the bottom, there is a 'Parameters' section with 'Number of star to mask ?' set to '5'. A 'Save' button is at the bottom right.

Şekil 3.13: Basit Yıldız Maskeleye Tekniği.

Şekil 3.13'te MUSTERILER tablosunda bulunan ADRES3 alanı için uygulamada bulunan Basit Yıldız Maskeleye (Simple Star Masking) tekniği tanımlanmıştır. Bu maskeleye tekniğinde ADRES3 alanında bulunan verinin yerine 5 adet "*" karakteri koyularak maskelenmiştir.



Şekil 3.14: Sütun Gizleme Tekniği.

Şekil 3.14’te MUSTERILER tablosunda bulunan ADRES2 alanı için uygulamada bulunan Sütun Gizleme (*Hide Column*) tekniği tanımlanmıştır. Bu teknik aynı zamanda Geçersiz Kılma ismi ile bilinmektedir. Bu maskeleme tekniğinde ADRES2 alanında bulunan veri gizlenerek maskeleme yapılmıştır.



Şekil 3.15: Karıştırma Tekniği.

Şekil 3.15’te MUSTERILER tablosunda bulunan BABA_ADI alanı için uygulamada bulunan Karıştırma (*Scrambling*) tekniği tanımlanmıştır. Bu maskeleme tekniğinde BABA_ADI alanında bulunan veri MUSTERILER tablosunda bulunan diğer kayıtlardaki BABA_ADI alanında bulunan veriyle değiştirilerek maskeleme yapılmıştır.

Mask Field (ESC to Cancel)

Mask Type: Star Masking

Field Name: soyad

Table: All

Environment: All

Important information about mask definitions.
[Click here to read!](#)

Parameters

Mask Percent: 50

Mask Length: 5

[Save](#)

Şekil 3.16: Yıldız Maskeleye Tekniği.

Şekil 3.16’da MUSTERILER tablosunda bulunan SOYAD alanı için uygulamada bulunan Yıldız Maskeleye (Star Masking) tekniği tanımlanmıştır. Bu maskeleye tekniğinde SOYAD alanında bulunan veri %50 oranında “*” karakteriyle maskelenmiştir.



Homepage | Release | Request | Dynamic DB Reader | Terminal | Operations | Definitions | Operation Log | Mask Field Search | Search Log | Sampling | Data Generation | Relational Search | Logged User: admin | Log Out | Administration

Mask Fields

[Create New](#)

Field Name	Environment	Table	Mask Type	Created By	Status	Actions
eposta	ALL	ALL	Email Masking	admin	ACTIVE	Actions
telefon_numarasi	ALL	ALL	Digital Masking	admin	ACTIVE	Actions
dogum_tarihi	ALL	ALL	Date Masking	admin	ACTIVE	Actions
anne_kizlik_soyadi	ALL	ALL	Masking By Hash Algorithm	admin	ACTIVE	Actions
adres3	ALL	ALL	Simple Star Masking	admin	ACTIVE	Actions
adres2	ALL	ALL	Hide Column	admin	ACTIVE	Actions
baba_adi	ALL	ALL	Scrambling	admin	ACTIVE	Actions
soyad	ALL	ALL	Composit Masking	admin	DEACTIVE	Actions
soyad	ALL	ALL	Star Masking	admin	ACTIVE	Actions

(1 of 1) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

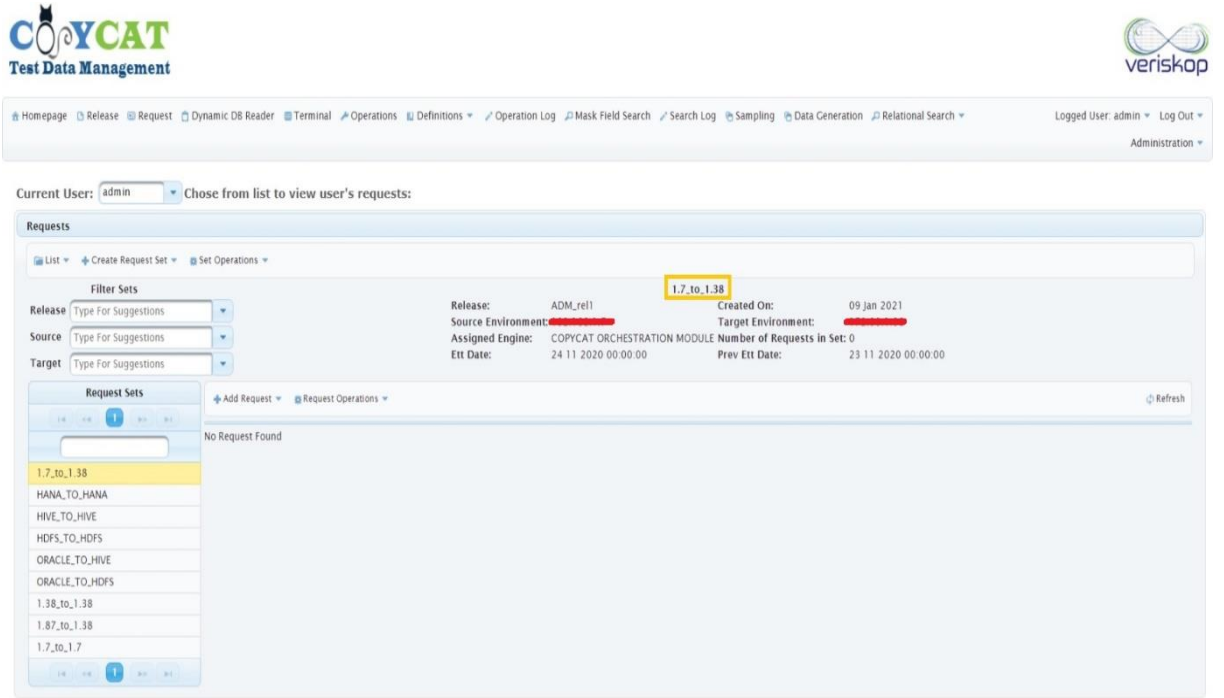
Şekil 3.17: Veri Alanlarına Tanımlanan Maskeleye Tekniklerinin Listesi.

Şekil 3.17’de MUSTERILER ve SOZLESME tablosunda bulunan alanlar için uygulanmış olan maskeleye tekniklerinin tamamı gösterilmiştir. MUSTERILER tablosunda bulunan SOYAD

alanı için uygulamada bulunan Karışık Maskeleye (*Composit Masking*) tekniği tanımlanmıştır. Fakat tanımlanan bu teknik devre dışı bırakılmıştır.

3.7. MASKELENEN VERİ SETLERİNİN AKTARIMI

Çalışma yapılırken uygulamada yer alan maskeleye teknikleri tanımlandıktan sonra maskelenecek veri setlerinin bulundukları veri tabanı ortamından nereye aktarılacağını belirlenmesi gerekmektedir. Uygulamada veri setlerinin maskelenerek aktarımının nereye yapılacağını belirlemek için uygulama içerisinde işlem isteği oluşturulur.



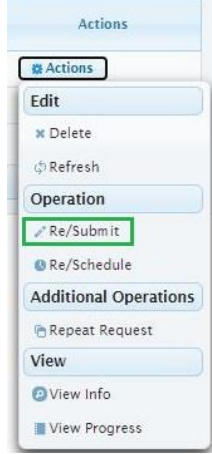
Şekil 3.18: Veri Seti Aktarımı Yapılan Ortam Bilgilerinin Seçilmesi.

Şekil 3.18’de uygulamada veri seti aktarımı yapabilmek için oluşturulan istek ekranı gösterilmektedir. Bu kısımda veri setinin aktarımının yapılacağı ortam bilgileri seçilmektedir. Çalışmada canlı ortam veri tabanı olarak “1.7” ismindeki veri tabanı kullanılmıştır. Aktarım yapılacak olan test ortam veri tabanı olaraksa “1.38” ismindeki veri tabanı seçilmiştir.

Şekil 3.19: Veri Seti Aktarım İsteği Bilgilerinin Detayı.

Şekil 3.19’da uygulamada yapılan veri seti aktarımı detay bilgilerinin seçildiği ekran gösterilmiştir. Bu ekranda veri seti aktarım isteğinin tipi, öncelik değeri, istek içeriği, hedef veri tabanı şema ismi, hedef veri tabanında oluşturulacak tablo ismi vb. bilgiler seçilerek istek oluşturulmaktadır. İstek bilgilerinde MUSTERILER ve ilişkili olduğu SOZLESME tablolarında bulunan tüm veri setlerinin test ortamı veri tabanı olarak seçilen “1.38” veri tabanı altındaki COPYCAT şemasının içerisinde MUSTERILER_TEST ve SOZLESME_TEST isimlerindeki tablolar olarak oluşturulacağı belirlenmiştir. Şekil 3.20’de detay bilgileri belirlenen istek oluşturulmuştur.

Şekil 3.20: Veri Seti Aktarım İsteği Oluşturma.



Şekil 3.21: Veri Seti Aktarımını Gerçekleştirme.

Şekil 3.21’de daha önce oluşturulmuş olan istek, onay verilerek çalıştırılır. Çalışmada kullanılan veri setleri için belirlenen teknikler kullanılarak maskelenen veri aktarım için belirlenen ortama taşınacaktır. Bu şekilde veri setlerinin maskeli bir şekilde aktarımı tamamlanmış olacaktır. İsteğin onay verilerek çalıştırılması dışında bu ekran içerisinde aktarım isteğinin detayı değiştirilebilir, istek silinebilir, istek tekrar çalıştırılabilir, isteğin çalışması için zaman planlaması yapılabilir, isteğin çalıştırılması sonucu anlık olarak, belirli sayıdaki maskelenmiş kayıt uygulama ekranından görüntülenebilir.

Operations									
Executors Requests									
Copy Engine					Executors				
					Parallel Request Count	Running Count	Waiting Count	Active Threads	Executor Status
COM									
QUERY	→ MON → TUE → WED → THU → FRI → SAT → SUN	16	0	0	0	0	0	0	RUNNING II
DB_OBJECT	→ MON → TUE → WED → THU → FRI → SAT → SUN	2	0	0	0	0	0	0	RUNNING II
SCHEMA	→ MON → TUE → WED → THU → FRI → SAT → SUN	1	0	0	0	0	0	0	RUNNING II
SAMPLING	→ MON → TUE → WED → THU → FRI → SAT → SUN	8	0	0	0	0	0	0	RUNNING II
HADOOP	→ MON → TUE → WED → THU → FRI → SAT → SUN	4	0	0	0	0	0	0	RUNNING II
SDG	→ MON → TUE → WED → THU → FRI → SAT → SUN	10	0	0	0	0	0	0	RUNNING II
ABINITIO									
MFS	→ MON → TUE → WED → THU → FRI → SAT → SUN	3	0	0	0	0	0	0	RUNNING II
SERIAL	→ MON → TUE → WED → THU → FRI → SAT → SUN	2	0	0	0	0	0	0	RUNNING II
OBJECT	→ MON → TUE → WED → THU → FRI → SAT → SUN	2	0	0	0	0	0	0	RUNNING II
CREATE_DETAIL	→ MON → TUE → WED → THU → FRI → SAT → SUN	1	0	0	0	0	0	0	RUNNING II
PUBLISHER	→ MON → TUE → WED → THU → FRI → SAT → SUN	1	0	0	0	0	0	0	RUNNING II

Şekil 3.22: Çoklu İş Parçacıklarıyla Paralel Maskeleye İşlemi.

Operations									
Executors Requests									
Copy Engine						Executors			
						Parallel Request Count	Running Count	Waiting Count	Active Threads
COM									
QUERY	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	1	0
DB_OBJECT	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	2	0
SCHEMA	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	1	0
SAMPLING	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	8	0
HADOOP	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	4	0
SDG	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	10	0
ABINITIO									
MFS	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	3	0
SERIAL	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	2	0
OBJECT	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	2	0
CREATE_DETAIL	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	1	0
PUBLISHER	→ MON	→ TUE	→ WED	→ THU	→ FRI	→ SAT	→ SUN	1	0

Şekil 3.23: Tekli İş Parçacığıyla Paralel Olmayan Maskeleye İşlemi.

Uygulamada yapılan maskeleye ve maskelenen verinin aktarımı işlemi tekli iş parçacıklarıyla yapılabildiği gibi seçilen iş parçacığı miktarı artırılarak çoklu olarak da yapılabilmektedir. Çalışma kapsamında performans çıktılarını karşılaştırmak, uygulamayla yapılan maskeleye ve maskelenen verinin aktarılması işlemi Şekil 3.22'deki gibi çoklu(16 adet) iş parçacığıyla paralel olarak çalışacak şekilde ayarlanmıştır. Hazırlanan veri setlerine uygulanan maskeleye ve maskelenen verinin aktarılması işlemi çoklu(16 adet) iş parçacığıyla paralel olarak yapılmıştır. Daha sonrasında Şekil 3.23'teki gibi tekli iş parçacığıyla paralel olmadan yapılacak şekilde uygulama ayarlanmıştır. Hazırlanan veri setlerine uygulanan maskeleye ve maskelenen verinin aktarılması işlemi tekli iş parçacığıyla paralel olmadan tekrar yapılmıştır. Yapılan bu işlemlerin sonuçları birbiriyle karşılaştırılacaktır.

4. BULGULAR

Bu bölümde, çalışma için hazırlanan veri setleri ve kullanılan uygulamayla oluşturulan veri maskeleyme tekniklerinin bu veri setlerine uygulanması sonucunda oluşan maskelenmiş veri setlerinin karşılaştırılmasına yer verilmiştir. Ayrıca uygulamada uygulanan veri maskeleyme işlemi iki farklı şekilde gerçekleştirilmiştir. Çalışmada veri maskeleyme işlemi ilk olarak tekli iş parçacığıyla paralel çalışma olmadan yapılmış olup, ikinci olarak 16 iş parçacığıyla paralel olarak çalıştırılarak tekrardan yapılmıştır. Tekli ve çoklu iş parçacıklarıyla veri maskeleyme ve maskelenmiş verinin aktarımı yine bu bölümde karşılaştırılmıştır.

4.1. MASKELEME İŞLEMİNİN UYGULANMASI VE VERİ SETLERİNİN KARŞILAŞTIRILMASI

Çalışma kapsamında oluşturulan veri setleri içerisindeki alanlara uygulamada yer alan maskeleyme teknikleri tanımlanmış ve sonrasında uygulama üzerinden oluşturulan istekle maskelenmiş veri setleri belirlenen test ortamı veri tabanına aktarılmıştır. Maskelenerek aktarılan tablo isimlerinin sonuna “_TEST” eki eklenmiştir.

SELECT * FROM COPYCATLIVE.MUSTERILER WHERE MUSTERI_NO BETWEEN 150 AND 181;

Result

SELECT * FROM COPYCATLIVE.MUSTERILER WHERE MUSTERI_NO B

	ABC ADRES1	ABC ADRES2	ABC ADRES3	ABC ANNE_ADI	ABC AD	ABC SOYAD	ABC ANNE_KIZLIK_SOYADI	ABC BABA_ADI	123 TELEFON_NUMARASI	123 MUSTERI_NO
1	A sokak	B caddesi	Bolvadin	İlkgöz	Levent	Abdülbasit	Radiye	Memduh	5,420,000,150	150
2	A sokak	B caddesi	Kuyucak	Hoşseda	Nuray	Tannıkorur	Rahile	Tayfun	5,550,000,151	151
3	A sokak	B caddesi	Keşan	Efsun	Ayşe	Tanmer	Ramize	Cafer	5,050,000,152	152
4	A sokak	B caddesi	Kartal	Seyyal	Gamze	Atak	Ruhnevaz	Öner	5,320,000,153	153
5	A sokak	B caddesi	Çeşme	Arzucan	Ebru	Atadan	Rükneddin	Zişan	5,420,000,154	154
6	A sokak	B caddesi	Gökçeada	Lamia	Sevil	Tiğın	Rabia	Cemil	5,550,000,155	155
7	A sokak	B caddesi	Demirozü	Niran	Özge	Tamkurt	Lemehat	Kemal	5,550,000,156	156
8	A sokak	B caddesi	Baykan	Candan	Erdem	Atatug	Rasi	Ogun	5,050,000,157	157
9	A sokak	B caddesi	Elazığ Merkez	Yağmur	Gözde	Akbörü	Lütfi	Arat	5,050,000,159	159
10	A sokak	B caddesi	Gülağaç	Zühre	Ferda	Aydar	Rami	Öner	5,550,000,160	160
11	A sokak	B caddesi	Abana	Serdar	Nurdan	Tolay	Rekine	Haydar	5,050,000,161	161
12	A sokak	B caddesi	Esenler	Dilber	Burçın	Altıntaş	Rücum	Yüksel	5,350,000,162	162
13	A sokak	B caddesi	Erzurum Merkez	Cavidan	Berker	Alışın	Lema	Ferhat	5,350,000,163	163
14	A sokak	B caddesi	Gümüşova	Buse	Ümit	Atadan	Rafi	Vural	5,050,000,164	164
15	A sokak	B caddesi	Eskipazar	Nevbahar	Abdul azim	Talay	Resmi	Nejat	5,050,000,166	166
16	A sokak	B caddesi	Domanıç	Aktan	Seda	Aydın	Rindan	Övünç	5,050,000,167	167
17	A sokak	B caddesi	Sinop Merkez	Erem	Elif	Tekbek	Reyan	Yaşar	5,320,000,168	168
18	A sokak	B caddesi	Sarayönü	Büküm	Pınar	Tüzeman	Rugen	Okan	5,350,000,169	169
19	A sokak	B caddesi	Bucak	Olgaç	Hasan yavuzhan	Tanyolu	Raide	Haydar	5,320,000,170	170
20	A sokak	B caddesi	Damal	Balin	Berrin	Avni	Ratip	Yalçın	5,550,000,171	171
21	A sokak	B caddesi	Danca	Mine	Hasan	Acunman	Rifat	Haydar	5,320,000,172	172
22	A sokak	B caddesi	Çaybaşı	Ferahnur	Zeynalabidin	Tuyan	Rabbani	Suat	5,350,000,173	173
23	A sokak	B caddesi	Karaburun	Semanur	Hüseyin	Tuğsavun	Lokman	Barın	5,550,000,174	174
24	A sokak	B caddesi	Dicle	Ilma	Emine	Arda	Rebia	Önay	5,320,000,175	175
25	A sokak	B caddesi	Biga	Gülbin	Mesut	Arhan	Lasif	Arda	5,420,000,176	176
26	A sokak	B caddesi	Abana	Sera	Leyla	Tüzüner	Lara	İbrahim	5,550,000,177	177
27	A sokak	B caddesi	Çatalpınar	Arzugül	İbrahim	Ayhatun	Leylak	Tümer	5,550,000,178	178
28	A sokak	B caddesi	Niksar	Belfu	Gamze	Abdünnasır	Resmiye	Yıldırım	5,320,000,179	179
29	A sokak	B caddesi	Derince	Aytu	Nevroz	Takrin	Ramize	Ramiz	5,320,000,180	180
30	A sokak	B caddesi	Lüleburgaz	Maral	Kadir	Tansen	Lamiye	Gizem	5,320,000,181	181

Şekil 4.1: Maskelenmeden Önce MUSTERILER Tablosu Veri Seti 1.Kısım.

SELECT * FROM COPYCAT.MUSTERILER_TEST WHERE MUSTERI_NO BETWEEN 150 AND 181;

Result

Enter a SQL expression to filter results (use Ctrl+Space)

Grid	ABC ADRES1	ABC ADRES2	ABC ADRES3	ABC ANNE_ADI	ABC AD	ABC SOYAD	ABC ANNE_KIZLIK_SOYADI	ABC BABA_ADI	123 TELEFON_NUMARASI	123 MUSTERI_NO
1	A sokak	[NULL]	*****	Ilkgüz	Levent	*b*ü*b*s*t	6218b1291f03734e2fe24e4b3fc3cd256ed0e483de8ab662	Talu	5,420,000,885	150
2	A sokak	[NULL]	*****	Hoşpoda	Nuray	*a**k*t*r	aced3ba941d3639a27d8fa56c8815d60a435d0baf467736	Ulu	5,550,000,883	151
3	A sokak	[NULL]	*****	Efsun	Ayşe	*a**r	d53391c1e311eb7f9d55c7cc624627e13211f0da537fb0c2	Alp	5,050,000,886	152
4	A sokak	[NULL]	*****	Seyyal	Gamze	*k*k	2d56606871c9a79cae57244bc9ade47e92ec005aa5719d07	Abdulkadir	5,320,000,887	153
5	A sokak	[NULL]	*****	Aruzcan	Ebru	*t*d'n	6de592cc9e9e9110b4ca481d6af3a0aec52acc2a9e65b5	Kemal	5,420,000,882	154
6	A sokak	[NULL]	*****	Lamia	Sevil	*t*n	23f97ad0834831ea8b5c6a77d07901d5758b6804cc709f7f	Ölmen	5,550,000,888	155
7	A sokak	[NULL]	*****	Niran	Özge	*a*k*t	ba497a5f0b930ed1a8ea26dba2149bd9c5ca12d22c7808	İdris	5,550,000,889	156
8	A sokak	[NULL]	*****	Candan	Erdem	*t*t'g	bb7c99f91c8c2777c73b6cfe0828ba2a7886cb6343914dc1f	Mecnun	5,050,000,881	157
9	A sokak	[NULL]	*****	Yağmur	Gözde	*k*o'u	472833b6af95bd7a59d1d48606bdc6b8ade0b872597c0b	Serdar	5,050,000,884	159
10	A sokak	[NULL]	*****	Zühre	Ferda	*y**r	cad0f51e24c7ec0b056c3295bb1f873741c97a4d3dc24094	Ali	5,550,000,899	160
11	A sokak	[NULL]	*****	Serdar	Nurdan	*o**y	787bdc31f9b769cea389a532a3485556682097817bf3d8ex	Rafet	5,050,000,897	161
12	A sokak	[NULL]	*****	Dilber	Burçin	*t'u*t's	38601e0bfdd86c8063120b3c4c4e30eaf179f38cf2eddd984	Ferhat	5,350,000,890	162
13	A sokak	[NULL]	*****	Cavidan	Berker	*t's'n	e6c0ca9c32b1bf56c229e7dccc13afca965717431690eb42	Necdet	5,350,000,891	163
14	A sokak	[NULL]	*****	Buse	Ümit	*t*d'n	3f0f8059a94084c212cd655a1f3184b069844f3c5e44273b	Şansal	5,050,000,896	164
15	A sokak	[NULL]	*****	Nevbahar	Abdul azim	*a*a**r	67d214f2fcc0bd7426b4b2a9485673a771914ad2705e52c05	Faik	5,050,000,893	166
16	A sokak	[NULL]	*****	Aktan	Seda	*y**n	11c31e84cc303114dd5a5981e5c6b182f967f012f2cf73f	Harun	5,050,000,895	167
17	A sokak	[NULL]	*****	Erem	Elif	*e*b*k	770164d62bf85c9004e6f865c9dbd5f5c749c9469c2f4f59a	Kamer	5,320,000,894	168
18	A sokak	[NULL]	*****	Büküm	Pinar	*ü*e**n	ff24badcd9b3557cee1f6b20456c7b262aab7dbcedc2c54	Ali	5,350,000,899	169
19	A sokak	[NULL]	*****	Olgaç	Hasan yavuzhan	*a**u	85450069beb803bf5c304ce97c485bfecc47e1a1ab13f7f1	Edip	5,320,000,818	170
20	A sokak	[NULL]	*****	Balin	Berrin	*c*y*	9704b2b46517355e5bc26a8947a9ee9d895e9af98e587cf66	Gökçe	5,550,000,816	171
21	A sokak	[NULL]	*****	Mine	Hasan	*c*n**n	f4d2bf34d8118fec0723bd6c6ad0b9a9c9bfe6a98c80f9dfc9	Tanik	5,320,000,819	172
22	A sokak	[NULL]	*****	Ferahnur	Zeynalabidin	*c**n	f5f32f96a448576dbac642da4e864ab391a2f282bc190e9	Uğur	5,350,000,810	173
23	A sokak	[NULL]	*****	Semanur	Hüseyin	*u*s*y'n	978caf0423558694f773dada1f8b6543be50b091ebb4865f	Semih	5,550,000,815	174
24	A sokak	[NULL]	*****	İlma	Emine	*a*	63c3e750bde50eb035167d5db2ed1edc5c87733036093f73	i-i	5,320,000,811	175
25	A sokak	[NULL]	*****	Gülbin	Mesut	*t**n	f700d7e02973a9d96834f76192b4c01641c3fe252a89e5	Gökbay	5,420,000,812	176
26	A sokak	[NULL]	*****	Sera	Leyla	*u**r	2ca74c8786c7cf47e0205f43f302a595f3a106231d6fb6aa	Naci	5,550,000,814	177
27	A sokak	[NULL]	*****	Azzugül	İbrahim	*y*a**n	c3eaac9c900ce99dc758d539e4b01a9d8ac9450012a0ba1	Kandemir	5,550,000,813	178
28	A sokak	[NULL]	*****	Belfu	Gamze	*b*u**n*s*r	4042f142666461213c49aea16cdc54fb5d393a392568e68ae	Tanik	5,320,000,817	179
29	A sokak	[NULL]	*****	Aytu	Nevroz	*a**n	d53391c1e311eb7f9d55c7cc624627e13211f0da537fb0c2	Unal	5,320,000,800	180
30	A sokak	[NULL]	*****	Maral	Kadir	*a*s*n	2af061fcd208c6cf93ca2660cf2d311794a856dba82689c97f	Fahrettin / fahri	5,320,000,808	181

Şekil 4.2: Maskelendikten Sonra MUSTERILER_TEST Tablosu Veri Seti 1.Kısım.

Şekil 4.1’de maskelenmemiş MUSTERILER tablosu veri setinin birinci kısmı 30 kayıt için gösterilmiştir. Şekil 4.2’de ise maskelenmiş şekilde MUSTERILER tablosu veri setinin birinci kısmı 30 kayıt için gösterilmiştir.

SELECT * FROM COPYCAT.MUSTERILER WHERE MUSTERI_NO BETWEEN 150 AND 181;

Result

Enter a SQL expression to filter results (use Ctrl+Space)

Grid	123 MUSTERI_NO	123 MUSTERI_TEMEL_KIMLIK_NO	ABC EPOSTA	ABC DURUM	DOGUM_TARIHI	BASLANGIC_TARIHI	ABC AD_ILK_HARF	ABC SOYAD_ILK_HARF	ABC AD_SOYAD_ILK_HARF
1	150	1,000,000,150	levant.a@msn.com	ACTIVE	1979-11-06 01:40:23	1998-11-04 13:40:23	L	A	LA
2	151	1,000,000,151	nuray.t@msn.com	ACTIVE	2016-03-23 01:40:23	2017-01-12 01:40:23	N	T	NT
3	152	1,000,000,152	ayse@gmail.com	ACTIVE	1964-10-08 01:40:23	1991-04-22 01:40:23	A	T	AT
4	153	1,000,000,153	gamze@hotmail.com	ACTIVE	1965-09-11 01:40:23	1991-10-08 01:40:23	G	A	GA
5	154	1,000,000,154	ebru@gmail.com	ACTIVE	1992-12-31 01:40:23	2005-06-02 13:40:23	E	A	EA
6	155	1,000,000,155	sevil.tigin@gmail.com	ACTIVE	1978-12-23 01:40:23	1998-05-29 13:40:23	S	T	ST
7	156	1,000,000,156	ozge@i2i-systems.com	ACTIVE	2009-11-20 01:40:23	2013-11-11 13:40:23	Ö	T	ÖT
8	157	1,000,000,157	erdem@msn.com	ACTIVE	1989-05-16 01:40:23	2003-08-10 01:40:23	E	A	EA
9	159	1,000,000,159	gozde.akboru@i2i-systems.com	ACTIVE	1990-11-17 01:40:23	2004-05-11 01:40:23	G	A	GA
10	160	1,000,000,160	ferda@hotmail.com	ACTIVE	1964-01-23 01:40:23	1990-12-13 13:40:23	F	A	FA
11	161	1,000,000,161	n.tolay@gmail.com	ACTIVE	1993-06-27 01:40:23	2005-08-30 13:40:23	N	T	NT
12	162	1,000,000,162	burcin@i2i-systems.com	ACTIVE	1996-03-24 01:40:23	2007-01-13 01:40:23	B	A	BA
13	163	1,000,000,163	berker.alkın@gmail.com	ACTIVE	1980-11-28 01:40:23	1999-05-17 13:40:23	B	A	BA
14	164	1,000,000,164	umit.atadan@outlook.com	ACTIVE	2005-07-27 01:40:23	2011-09-15 01:40:23	Ü	A	ÜA
15	166	1,000,000,166	abdul azim@msn.com	ACTIVE	1995-06-04 01:40:23	2006-08-19 01:40:23	A	T	AT
16	167	1,000,000,167	s.aydin@hotmail.com	ACTIVE	1971-02-16 01:40:23	1994-06-26 01:40:23	S	A	SA
17	168	1,000,000,168	elif.tekkek@gmail.com	ACTIVE	1978-09-24 01:40:23	1998-04-14 13:40:23	E	T	ET
18	169	1,000,000,169	pinart@msn.com	ACTIVE	1965-09-30 01:40:23	1991-10-17 13:40:23	P	T	PT
19	170	1,000,000,170	h.tanyolu@msn.com	ACTIVE	2011-03-28 01:40:23	2014-07-16 01:40:23	H	T	HT
20	171	1,000,000,171	berin.avni@msn.com	ACTIVE	1980-08-09 01:40:23	1999-03-23 01:40:23	B	A	BA
21	172	1,000,000,172	hasan.a@i2i-systems.com	ACTIVE	2005-05-15 01:40:23	2011-08-09 13:40:23	H	A	HA
22	173	1,000,000,173	z.tuyan@outlook.com	ACTIVE	1996-11-02 01:40:23	2007-05-04 13:40:23	Z	T	ZT
23	174	1,000,000,174	huseyin.t@hotmail.com	ACTIVE	2001-06-07 01:40:23	2009-08-20 13:40:23	H	T	HT
24	175	1,000,000,175	e.arda@i2i-systems.com	ACTIVE	1979-03-25 01:40:23	1998-07-14 13:40:23	E	A	EA
25	176	1,000,000,176	mesut@outlook.com	ACTIVE	2001-03-31 01:40:23	2009-07-17 13:40:23	M	A	MA
26	177	1,000,000,177	ltuzuner@hotmail.com	ACTIVE	1986-07-13 01:40:23	2002-03-09 01:40:23	L	T	LT
27	178	1,000,000,178	ibrahim.ayhatun@gmail.com	ACTIVE	1991-05-09 01:40:23	2004-08-05 13:40:23	I	A	IA
28	179	1,000,000,179	g.abdunnasir@gmail.com	ACTIVE	1978-07-13 01:40:23	1998-03-09 01:40:23	G	A	GA
29	180	1,000,000,180	nevroz.takrin@outlook.com	ACTIVE	2001-04-16 01:40:23	2009-07-25 13:40:23	N	T	NT
30	181	1,000,000,181	kadir.tansen@gmail.com	ACTIVE	1992-12-26 01:40:23	2005-05-31 01:40:23	K	T	KT

Şekil 4.3: Maskelenmeden Önce MUSTERILER Tablosu Veri Seti 2.Kısım.

SELECT * FROM COPYCAT.MUSTERILER_TEST WHERE MUSTERI_NO BETWEEN 150 AND 181;

Result

Enter a SQL expression to filter results (use Ctrl+Space)

	123 MUSTERI_NO	123 MUSTERI_TEMEL_KIMLIK_NO	ASC EPOSTA	ASC DURUM	DOGUM_TARIHI	BASLANGIC_TARIHI	ASC AD_ILK_HARF	ASC SOYAD_ILK_HARF	ASC AD_SOYAD_ILK_HARF
1	150	1,000,000,150	*e*e*t*a@msn.com	ACTIVE	1979-08-19 11:51:17	1998-11-04 13:40:23	L	A	LA
2	151	1,000,000,151	*u*a*t@msn.com	ACTIVE	2016-04-04 14:51:17	2017-01-12 01:40:23	N	T	NT
3	152	1,000,000,152	*y*e@gmail.com	ACTIVE	1964-10-17 11:08:43	1991-04-22 01:40:23	A	T	AT
4	153	1,000,000,153	*a**e@hotmail.com	ACTIVE	1965-02-17 11:08:43	1991-10-08 01:40:23	G	A	GA
5	154	1,000,000,154	*b*u@gmail.com	ACTIVE	1992-11-30 14:51:17	2005-06-02 13:40:23	E	A	EA
6	155	1,000,000,155	*e*i**g*n@gmail.com	ACTIVE	1978-12-04 10:51:17	1998-05-29 13:40:23	S	T	ST
7	156	1,000,000,156	*e@i2i-systems.com	ACTIVE	2009-08-08 14:51:17	2013-11-11 13:40:23	Ö	T	ÖT
8	157	1,000,000,157	*t*m@msn.com	ACTIVE	1989-05-10 12:51:17	2003-08-10 01:40:23	E	A	EA
9	159	1,000,000,159	*o*d*,k*o*u@i2i-systems.com	ACTIVE	1990-08-29 14:51:17	2004-05-11 01:40:23	G	A	GA
10	160	1,000,000,160	*e*a@hotmail.com	ACTIVE	1964-04-29 13:08:43	1990-12-13 13:40:23	F	A	FA
11	161	1,000,000,161	*o*y@gmail.com	ACTIVE	1993-01-03 14:51:17	2005-08-30 13:40:23	N	T	NT
12	162	1,000,000,162	*u*c*n@i2i-systems.com	ACTIVE	1996-04-27 14:51:17	2007-01-13 01:40:23	B	A	BA
13	163	1,000,000,163	*e*k*t**i*g*n@gmail.com	ACTIVE	1980-08-25 11:51:17	1999-05-17 13:40:23	B	A	BA
14	164	1,000,000,164	*m*t*a*d*n@outlook.com	ACTIVE	2005-06-03 14:51:17	2011-09-15 01:40:23	Ü	A	ÜA
15	166	1,000,000,166	*b*u*t*z*m@msn.com	ACTIVE	1995-01-21 14:51:17	2006-08-19 01:40:23	A	T	AT
16	167	1,000,000,167	*y**n@hotmail.com	ACTIVE	1971-07-10 12:51:17	1994-06-26 01:40:23	S	A	SA
17	168	1,000,000,168	*f*f*t*b*k@gmail.com	ACTIVE	1978-11-27 10:51:17	1998-04-14 13:40:23	E	T	ET
18	169	1,000,000,169	*r*a*t@msn.com	ACTIVE	1965-03-02 11:08:43	1991-10-17 13:40:23	P	T	PT
19	170	1,000,000,170	*a*t*o*u@msn.com	ACTIVE	2011-04-25 14:51:17	2014-07-16 01:40:23	H	T	HT
20	171	1,000,000,171	*e*r*n**v*i@msn.com	ACTIVE	1980-01-16 11:51:17	1999-03-23 01:40:23	B	A	BA
21	172	1,000,000,172	*a*a*a@i2i-systems.com	ACTIVE	2005-05-11 14:51:17	2011-08-09 13:40:23	H	A	HA
22	173	1,000,000,173	*u*t*n@outlook.com	ACTIVE	1996-08-23 14:51:17	2007-05-04 13:40:23	Z	T	ZT
23	174	1,000,000,174	*u*e*n*t@hotmail.com	ACTIVE	2001-01-18 14:51:17	2009-08-20 13:40:23	H	T	HT
24	175	1,000,000,175	*r*a@i2i-systems.com	ACTIVE	1979-04-26 11:51:17	1998-07-14 13:40:23	E	A	EA
25	176	1,000,000,176	*e*t@outlook.com	ACTIVE	2001-03-31 14:51:17	2009-07-17 13:40:23	M	A	MA
26	177	1,000,000,177	*u*t*n@hotmail.com	ACTIVE	1986-06-12 12:51:17	2002-03-09 01:40:23	L	T	LT
27	178	1,000,000,178	*b*a*i**h*t*n@gmail.com	ACTIVE	1991-05-16 14:51:17	2004-08-05 13:40:23	I	A	İA
28	179	1,000,000,179	*b*u*n*s*n@gmail.com	ACTIVE	1978-06-12 12:51:17	1998-03-09 01:40:23	G	A	GA
29	180	1,000,000,180	*e*r*t*a*r*n@outlook.com	ACTIVE	2001-02-10 14:51:17	2009-07-25 13:40:23	N	T	NT
30	181	1,000,000,181	*a*i**a*s*n@gmail.com	ACTIVE	1992-12-05 14:51:17	2005-05-31 01:40:23	K	T	KT

Şekil 4.4: Maskelendikten Sonra MUSTERILER_TEST Tablosu Veri Seti 2.Kısım.

Şekil 4.3'te maskelenmemiş MUSTERILER tablosu veri setinin ikinci kısmı 30 kayıt için gösterilmiştir. Şekil 4.4'te ise maskelenmiş şekilde MUSTERILER tablosu veri setinin ikinci kısmı 30 kayıt için gösterilmiştir.

SELECT * FROM COPYCATLIVE.SOZLESME WHERE MUSTERI_NO BETWEEN 150 AND 181;

Result

Enter a SQL expression to filter results (use Ctrl+Space)

	123 SOZLESME_NO	123 MUSTERI_NO	BASLANGIC_TARIHI	BITIS_TARIHI	ASC DURUM	ASC EPOSTA	123 TELEFON_NUMARASI
1	985,792	150	1996-11-09 13:40:23	1998-11-04 13:40:23	ACTIVE	levent.a@msn.com	5,420,000,150
2	985,793	151	2015-01-18 01:40:23	2017-01-12 01:40:23	ACTIVE	nuray.t@msn.com	5,550,000,151
3	985,794	152	1989-04-27 01:40:23	1991-04-22 01:40:23	ACTIVE	ayşe@gmail.com	5,050,000,152
4	985,795	153	1989-10-13 01:40:23	1991-10-08 01:40:23	ACTIVE	gamze@hotmail.com	5,320,000,153
5	985,796	154	2003-06-08 13:40:23	2005-06-02 13:40:23	ACTIVE	ebru@gmail.com	5,420,000,154
6	985,797	155	1996-06-03 13:40:23	1998-05-29 13:40:23	ACTIVE	sevil.tigin@gmail.com	5,550,000,155
7	985,798	156	2011-11-17 13:40:23	2013-11-11 13:40:23	ACTIVE	ozge@i2i-systems.com	5,550,000,156
8	985,799	157	2001-08-15 01:40:23	2003-08-10 01:40:23	ACTIVE	erdem@msn.com	5,050,000,157
9	985,801	159	2002-05-17 01:40:23	2004-05-11 01:40:23	ACTIVE	gozde.akboru@i2i-systems.com	5,050,000,159
10	985,802	160	1988-12-18 13:40:23	1990-12-13 13:40:23	ACTIVE	ferda@hotmail.com	5,550,000,160
11	985,803	161	2003-09-05 13:40:23	2005-08-30 13:40:23	ACTIVE	n.tolay@gmail.com	5,050,000,161
12	985,804	162	2005-01-18 01:40:23	2007-01-13 01:40:23	ACTIVE	burcin@i2i-systems.com	5,350,000,162
13	985,805	163	1997-05-22 13:40:23	1999-05-17 13:40:23	ACTIVE	berker.alışın@gmail.com	5,350,000,163
14	985,806	164	2009-09-20 01:40:23	2011-09-15 01:40:23	ACTIVE	umit.atadan@outlook.com	5,050,000,164
15	985,808	166	2004-08-24 01:40:23	2006-08-19 01:40:23	ACTIVE	abdul azim@msn.com	5,050,000,166
16	985,809	167	1992-07-01 01:40:23	1994-06-26 01:40:23	ACTIVE	s.aydın@hotmail.com	5,050,000,167
17	985,810	168	1996-04-19 13:40:23	1998-04-14 13:40:23	ACTIVE	elif.tekbek@gmail.com	5,320,000,168
18	985,811	169	1989-10-22 13:40:23	1991-10-17 13:40:23	ACTIVE	pinar.t@msn.com	5,350,000,169
19	985,812	170	2012-07-21 01:40:23	2014-07-16 01:40:23	ACTIVE	h.tanyolu@msn.com	5,320,000,170
20	985,813	171	1997-03-28 01:40:23	1999-03-23 01:40:23	ACTIVE	berrin.avni@msn.com	5,550,000,171
21	985,814	172	2009-08-14 13:40:23	2011-08-09 13:40:23	ACTIVE	hasana@i2i-systems.com	5,320,000,172
22	985,815	173	2005-05-09 13:40:23	2007-05-04 13:40:23	ACTIVE	z.tuyan@outlook.com	5,350,000,173
23	985,816	174	2007-08-26 13:40:23	2009-08-20 13:40:23	ACTIVE	huseyin.t@hotmail.com	5,550,000,174
24	985,817	175	1996-07-19 13:40:23	1998-07-14 13:40:23	ACTIVE	e.arda@i2i-systems.com	5,320,000,175
25	985,818	176	2007-07-23 13:40:23	2009-07-17 13:40:23	ACTIVE	mesut@outlook.com	5,420,000,176
26	985,819	177	2000-03-14 01:40:23	2002-03-09 01:40:23	ACTIVE	ltuzuner@hotmail.com	5,550,000,177
27	985,820	178	2002-08-11 13:40:23	2004-08-05 13:40:23	ACTIVE	ibrahim.ayhatun@gmail.com	5,550,000,178
28	985,821	179	1996-03-14 01:40:23	1998-03-09 01:40:23	ACTIVE	g.abdunnasır@gmail.com	5,320,000,179
29	985,822	180	2007-07-31 13:40:23	2009-07-25 13:40:23	ACTIVE	nevroz.takrın@outlook.com	5,320,000,180
30	985,823	181	2003-06-06 01:40:23	2005-05-31 01:40:23	ACTIVE	kadir.tansen@gmail.com	5,320,000,181

Şekil 4.5: Maskelenmeden Önce SOZLESME Tablosu Veri Seti.

SELECT * FROM COPYCAT.SOZLESME_TEST WHERE MUSTERI_NO BETWEEN 150 AND 181;

Result

SELECT * FROM COPYCAT.SOZLESME_TEST WHERE MUSTERI_NO BETWEEN 150 AND 181; Enter a SQL expression to filter results (use Ctrl+Space)

	123 SOZLESME_NO	123 MUSTERI_NO	BASLANGIC_TARIHI	BITIS_TARIHI	DURUM	EPOSTA	123 TELEFON_NUMARASI
1	3,771,906	150	1998-11-04 13:40:23	1998-11-04 13:40:23	ACTIVE	*e*e*t*a@msn.com	5,420,000,885
2	3,771,907	151	2017-01-12 01:40:23	2017-01-12 01:40:23	ACTIVE	*u*a**t@msn.com	5,550,000,883
3	3,771,908	152	1991-04-22 01:40:23	1991-04-22 01:40:23	ACTIVE	*y*e@gmail.com	5,050,000,886
4	3,771,909	153	1991-10-08 01:40:23	1991-10-08 01:40:23	ACTIVE	*a**e@hotmail.com	5,320,000,887
5	3,771,910	154	2005-06-02 13:40:23	2005-06-02 13:40:23	ACTIVE	*b*u@gmail.com	5,420,000,882
6	3,771,911	155	1998-05-29 13:40:23	1998-05-29 13:40:23	ACTIVE	*e*i*,**g*n@gmail.com	5,550,000,888
7	3,771,912	156	2013-11-11 13:40:23	2013-11-11 13:40:23	ACTIVE	*z*e@izi-systems.com	5,550,000,889
8	3,771,913	157	2003-08-10 01:40:23	2003-08-10 01:40:23	ACTIVE	*r**m@msn.com	5,050,000,881
9	3,771,914	159	2004-05-11 01:40:23	2004-05-11 01:40:23	ACTIVE	*o*d*,*k*o*u@izi-systems.com	5,050,000,884
10	3,771,915	160	1990-12-13 13:40:23	1990-12-13 13:40:23	ACTIVE	*e**a@hotmail.com	5,550,000,899
11	3,771,916	161	2005-08-30 13:40:23	2005-08-30 13:40:23	ACTIVE	*.o**y@gmail.com	5,050,000,897
12	3,771,917	162	2007-01-13 01:40:23	2007-01-13 01:40:23	ACTIVE	*u*c*n@izi-systems.com	5,350,000,890
13	3,771,918	163	1999-05-17 13:40:23	1999-05-17 13:40:23	ACTIVE	*e*k*r**l*i*n@gmail.com	5,350,000,891
14	3,771,919	164	2011-09-15 01:40:23	2011-09-15 01:40:23	ACTIVE	*m**a**d*n@outlook.com	5,050,000,896
15	3,771,920	166	2006-08-19 01:40:23	2006-08-19 01:40:23	ACTIVE	*b*u* *z*m@msn.com	5,050,000,893
16	3,771,921	167	1994-06-26 01:40:23	1994-06-26 01:40:23	ACTIVE	*.y**n@hotmail.com	5,050,000,895
17	3,771,922	168	1998-04-14 13:40:23	1998-04-14 13:40:23	ACTIVE	*l*f*t**b*k@gmail.com	5,320,000,894
18	3,771,923	169	1991-10-17 13:40:23	1991-10-17 13:40:23	ACTIVE	*i*a**t@msn.com	5,350,000,898
19	3,771,924	170	2014-07-16 01:40:23	2014-07-16 01:40:23	ACTIVE	*.a**o*u@msn.com	5,320,000,818
20	3,771,925	171	1999-03-23 01:40:23	1999-03-23 01:40:23	ACTIVE	*e*r*n**y*i@msn.com	5,550,000,816
21	3,771,926	172	2011-08-09 13:40:23	2011-08-09 13:40:23	ACTIVE	*a*a**a@izi-systems.com	5,320,000,819
22	3,771,927	173	2007-05-04 13:40:23	2007-05-04 13:40:23	ACTIVE	*.u**n@outlook.com	5,350,000,810
23	3,771,928	174	2009-08-20 13:40:23	2009-08-20 13:40:23	ACTIVE	*u*e**n*t@hotmail.com	5,550,000,815
24	3,771,929	175	1998-07-14 13:40:23	1998-07-14 13:40:23	ACTIVE	*.r*a@izi-systems.com	5,320,000,811
25	3,771,930	176	2009-07-17 13:40:23	2009-07-17 13:40:23	ACTIVE	*e**t@outlook.com	5,420,000,812
26	3,771,931	177	2002-03-09 01:40:23	2002-03-09 01:40:23	ACTIVE	*.u**n*r@hotmail.com	5,550,000,814
27	3,771,932	178	2004-08-05 13:40:23	2004-08-05 13:40:23	ACTIVE	*b*a*i*,**h*t*n@gmail.com	5,550,000,813
28	3,771,933	179	1998-03-09 01:40:23	1998-03-09 01:40:23	ACTIVE	*.b*u*n*s*r@gmail.com	5,320,000,817
29	3,771,934	180	2009-07-25 13:40:23	2009-07-25 13:40:23	ACTIVE	*e*r*z**a*r*n@outlook.com	5,320,000,800
30	3,771,935	181	2005-05-31 01:40:23	2005-05-31 01:40:23	ACTIVE	*a*i*,*a*s*n@gmail.com	5,320,000,808

Şekil 4.6: Maskelendikten Sonra SOZLESME_TEST Tablosu Veri Seti.

Şekil 4.5'te maskelenmemiş SOZLESME tablosu veri seti 30 kayıt için gösterilmiştir. Şekil 4.6'da ise maskelenmiş şekilde SOZLESME tablosu veri seti 30 kayıt için gösterilmiştir. Çalışma kapsamında oluşturulan veri setleri içerisindeki alanlara uygulamada yer alan maskeleme teknikleri uygulanmıştır. Maskelenmeden önce ve maskelendikten sonra veri setlerinin durumu, maskeleme tekniklerine göre ayrı ayrı karşılaştırılmıştır.

4.1.1. E-posta Maskeleme Tekniği Sonuçları

Bu maskeleme tekniği MUSTERILER ve SOZLESME veri setlerinde bulunan EPOSTA alanına uygulanmıştır. Bu maskeleme tekniğiyle e-posta verisinde olması zorunlu olan "@" işaretinden önce gelen karakterlerin %50 oranındaki kısmı "*" karakteriyle maskelenmiştir. Tablo 4.1'de MUSTERILER ve SOZLESME veri setlerinde bulunan EPOSTA veri alanına maskeleme işlemi uygulanmadan önceki hali ve maskeleme işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir. Çalışmada kullanılan veri setlerindeki EPOSTA alanları, veri setlerindeki MUSTERI_NO alanları aynı olan kayıtlarda aynıdır. Maskeleme işlemi 2 veri setine de uygulanmış olup Tablo 4.1'deki sonuç veri setlerindeki ortak sonuçtur.

Tablo 4.1: E-posta Maskeleye Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	EPOSTA	EPOSTA
150	levent.a@msn.com	*e*e*t*a@msn.com
151	nuray.t@msn.com	*u*a*t@msn.com
152	ayşe@gmail.com	*y*e@gmail.com
153	gamze@hotmail.com	*a**e@hotmail.com
154	ebru@gmail.com	*b*u@gmail.com
155	sevil.tiğin@gmail.com	*e*i*. **ğ*n@gmail.com
156	ozge@i2i-systems.com	*z*e@i2i-systems.com
157	erdem@msn.com	*i**m@msn.com
159	gozde.akboru@i2i-systems.com	*o*d*. *k*o*u@i2i-systems.com
160	ferda@hotmail.com	*e**a@hotmail.com

4.1.2. Dijital Maskeleye Tekniği Sonuçları

Bu maskeleye tekniği MUSTERILER ve SOZLESME veri setlerinde bulunan TELEFON_NUMARASI alanına uygulanmıştır. Bu maskeleye tekniğinde TELEFON_NUMARASI alanında bulunan verinin sağdan 3 hanesi sayısal olarak başka bir rakamla değiştirilerek maskelenmiştir. Tablo 4.2’de MUSTERILER ve SOZLESME veri setlerinde bulunan TELEFON_NUMARASI veri alanına maskeleye işlemi uygulanmadan önceki hali ve maskeleye işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir. Çalışmada kullanılan veri setlerindeki TELEFON_NUMARASI alanları, veri setlerindeki MUSTERI_NO alanları aynı olan kayıtlarda aynıdır. Maskeleye işlemi 2 veri setine de uygulanmış olup Tablo 4.2’deki sonuç veri setlerindeki ortak sonuçtur.

Tablo 4.2: Dijital Maskeleye Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	TELEFON_NUMARASI	TELEFON_NUMARASI
150	5420000150	5420000885
151	5550000151	5550000883
152	5050000152	5050000886
153	5320000153	5320000887
154	5420000154	5420000882
155	5550000155	5550000888
156	5550000156	5550000889
157	5050000157	5050000881
159	5050000159	5050000884
160	5550000160	5550000899

4.1.3. Tarih Maskeleye Tekniği Sonuçları

Bu maskeleye tekniği MUSTERILER veri setinde bulunan DOGUM_TARIHI alanına uygulanmıştır. Bu maskeleye tekniğinde DOGUM_TARIHI alanında bulunan verinin 01.01.1990'dan sonra ve 01.01.2021'den önce olan tarih değerleriyle; saat, gün ve ay kısımları uygun değerlerle değiştirilerek maskelenmiştir. DOGUM_TARIHI verisinin yıl kısmında maskeleye yapılmamıştır. Tablo 4.3'te MUSTERILER veri setinde bulunan DOGUM_TARIHI veri alanına maskeleye işlemi uygulanmadan önceki hali ve maskeleye işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir.

Tablo 4.3: Tarih Maskeleye Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	DOGUM_TARIHI	DOGUM_TARIHI
150	1979-11-06 01:40:23	1979-08-19 11:51:17
151	2016-03-23 01:40:23	2016-04-04 14:51:17
152	1964-10-08 01:40:23	1964-10-17 11:08:43
153	1965-09-11 01:40:23	1965-02-17 11:08:43
154	1992-12-31 01:40:23	1992-11-30 14:51:17
155	1978-12-23 01:40:23	1978-12-04 10:51:17
156	2009-11-20 01:40:23	2009-08-08 14:51:17
157	1989-05-16 01:40:23	1989-05-10 12:51:17
159	1990-11-17 01:40:23	1990-08-29 14:51:17
160	1964-01-23 01:40:23	1964-04-29 13:08:43

4.1.4. Hash Algoritmasıyla Maskeleye Tekniği Sonuçları

Bu maskeleye tekniği MUSTERILER veri setinde bulunan ANNE_KIZLIK_SOYADI alanına uygulanmıştır. Bu maskeleye tekniğinde ANNE_KIZLIK_SOYADI alanında bulunan veri SHA-256 Hashing Algoritmasıyla maskelenmiştir. Tablo 4.4'te MUSTERILER veri setinde bulunan ANNE_KIZLIK_SOYADI veri alanına maskeleye işlemi uygulanmadan önceki hali ve maskeleye işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir.

Tablo 4.4: Hash Algoritmasıyla Maskeleye Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	ANNE_KIZLIK_SOYADI	ANNE_KIZLIK_SOYADI
150	Radiye	6218b1291f03734e2fe24e4eb3fc3cd256ed0e483de8ab662706ec4dcec356b5
151	Rahile	aced3ba941d3639a27d8fa56c8815d60a435d0baf4677736798f7989e4d98670
152	Ramize	d53391c1e311eb7ff9d55c7cc624627e13211f0da537fb0c228d5c7dad582df5
153	Ruhnevaz	2d56606871c9a79cae57244bc9ade47e92ec005aa5719d075f7d201d67860532
154	Rükneddin	6de5592cc9e9e9110b4ca481d6a3a0aec52acc2a9e65b532a2883c44aca9c1
155	Rabia	235f79ad0834831ea8b5c8a77d07901d5758b880dcc705f7b35630e581ec7b2f
156	Lemehat	ba497a5f0b9303ed1a8ea26dba2149bd9c5c5a12d22c780811dc74dc493bd770
157	Rasi	bb7c99f91c8c2777c73b6cef0828ba2a7886cb6343914dc164d749dcc37a6cca
159	Lütfi	472833b6af95bd7a59d1d48606bdc6b8aade00b872597c0c71a2efeea6d1b562
160	Rami	cadf0f51e24c7ec0b056c3295bb1f873741c97a4d3dc24094045c91e9695e589

4.1.5. Basit Yıldız Maskeleye Tekniği Sonuçları

Bu maskeleye tekniği MUSTERILER veri setinde bulunan ADRES3 alanına uygulanmıştır. Bu maskeleye tekniğinde ADRES3 alanında bulunan verinin yerine 5 adet “*” karakteri koyularak maskelenmiştir. Tablo 4.5’te MUSTERILER veri setinde bulunan ADRES3 veri alanına maskeleye işlemi uygulanmadan önceki hali ve maskeleye işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir.

Tablo 4.5: Basit Yıldız Maskeleye Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	ADRES3	ADRES3
150	Bolvadin	*****
151	Kuyucak	*****
152	Keşan	*****
153	Kartal	*****
154	Çeşme	*****
155	Gökçeada	*****
156	Demirözü	*****
157	Baykan	*****
159	Elazığ Merkez	*****
160	Gülağaç	*****

4.1.6. Sütun Gizleme Tekniği Sonuçları

Bu maskeleye tekniği MUSTERILER veri setinde bulunan ADRES2 alanına uygulanmıştır. Bu teknik aynı zamanda Geçersiz Kılma ismi ile bilinmektedir. Bu maskeleye tekniğinde ADRES2 alanında bulunan veri gizlenerek maskelenmiştir. Tablo 4.6’da MUSTERILER veri setinde bulunan ADRES2 veri alanına maskeleye işlemi uygulanmadan önceki hali ve maskeleye işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir.

Tablo 4.6: Sütun Gizleme Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	ADRES2	ADRES2
150	B caddesi	
151	B caddesi	
152	B caddesi	
153	B caddesi	
154	B caddesi	
155	B caddesi	
156	B caddesi	
157	B caddesi	
159	B caddesi	
160	B caddesi	

4.1.7. Karıştırma Tekniği Sonuçları

Bu maskeleye tekniği MUSTERILER veri setinde bulunan BABA_ADI alanına uygulanmıştır. Bu maskeleye tekniğinde BABA_ADI alanında bulunan veri MUSTERILER tablosunda bulunan diğer kayıtlardaki BABA_ADI alanında bulunan veriyle değiştirilerek maskelenmiştir. Tablo 4.7’de MUSTERILER veri setinde bulunan BABA_ADI veri alanına maskeleye işlemi uygulanmadan önceki hali ve maskeleye işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir.

Tablo 4.7: Karıştırma Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	BABA_ADI	BABA_ADI
150	Memduh	Talu
151	Tayfun	Ulu
152	Cafer	Alp
153	Öner	Abdulkadir
154	Zişan	Kemal
155	Cemil	Ökmen
156	Kemal	İdris
157	Ogün	Mecnun
159	Arat	Serdar
160	Öner	Ali

4.1.8. Yıldız Maskeleye Tekniği Sonuçları

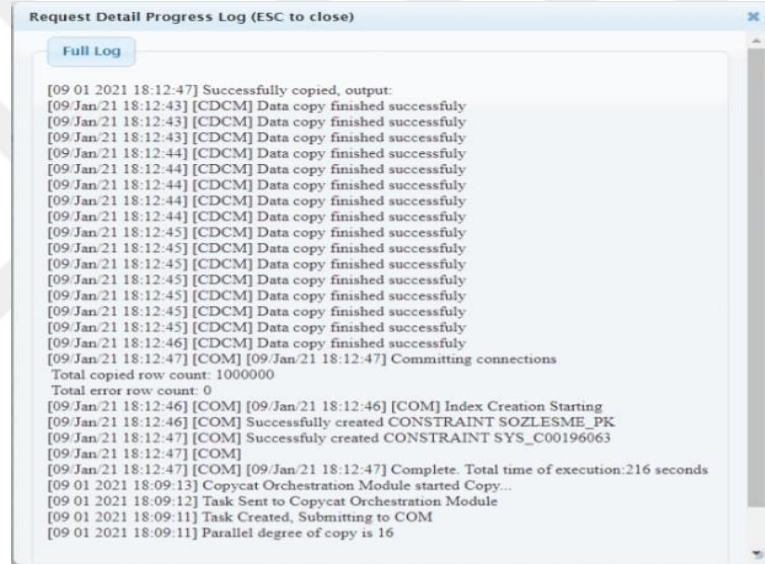
Bu maskeleye tekniği MUSTERILER veri setinde bulunan SOYAD alanına uygulanmıştır. Bu maskeleye tekniğinde SOYAD alanında bulunan veri %50 oranında “*” karakteriyle maskelenmiştir. Tablo 4.8’de MUSTERILER veri setinde bulunan SOYAD veri alanına maskeleye işlemi uygulanmadan önceki hali ve maskeleye işlemi uygulandıktan sonraki hali için 10 kayıt gösterilmiştir.

Tablo 4.8: Yıldız Maskeleye Tekniği Karşılaştırma.

	Maskelenmemiş Veri Seti	Maskelenmiş Veri Seti
MUSTERI_NO	SOYAD	SOYAD
150	Abdülbasit	*b*ü*b*s*t
151	Tanrıkörür	*a*r*k*r*r
152	Tarımer	*a*i**r
153	Atak	*t*k*
154	Atadan	*t*d*n
155	Tiğın	*i**n
156	Tamkurt	*a*k**t
157	Atatuğ	*t*t*ğ
159	Akbörü	*k*ö*ü
160	Aydar	*y**r

4.2.MASKELEME İŞLEMİNİN TEKLİ VE ÇOKLU İŞ PARÇACIKLARIYLA UYGULANMASI VE ZAMANSAL PERFORMANS KARŞILAŞTIRILMASI

Çalışma kapsamında uygulamada yapılan maskeleme ve maskelenen verinin aktarımı işlemi ilk olarak çoklu yani 16 iş parçacığıyla paralel olarak çalıştırılmıştır. Şekil 4.7’de çoklu iş parçacıklarıyla yapılan bu çalışma çıktılarının özetinin bulunduğu kayıt gösterilmiştir. Yapılan bu çalışmadan sonra performans karşılaştırması yapabilmek için uygulamada yapılan maskeleme ve maskelenen verinin aktarımı işlemi tekli iş parçacığıyla paralel çalışma olmadan yapılmıştır. Şekil 4.8’de tekli iş parçacığıyla paralel çalışma olmadan yapılan bu çalışma çıktılarının bulunduğu kayıt gösterilmiştir.



Şekil 4.7: Çoklu İş Parçacıklarıyla Maskeleme İşleminin Log Kayıtları.



Şekil 4.8: Tekli İş Parçacığıyla Maskeleme İşleminin Log Kayıtları.

Tablo 4.9: Tekli ve Çoklu İş Parçacıklarıyla Maskeleme Sonuçları Karşılaştırma.

Maskeleme ve Aktarım Tipi	Maskelenen Kayıt Sayısı	Maskelemede Hata Alan Kayıt Sayısı	Toplam İşlem Çalışma Süresi
Tekli İş Parçacığı - 1 Adet	1000000	0	317 saniye
Çoklu İş Parçacığı - 16 Adet	1000000	0	216 saniye

Tablo 4.9’da Uygulamada tekli ve çoklu iş parçacıklarıyla yapılmış olan çalışma çıktıları karşılaştırılmıştır. Çalışma kapsamında 1 milyon veri içeren veri setleri kullanılmış olup, 1 milyon veri içeren maskelenmiş veri setleri elde edilmiştir. Tekli ve çoklu iş parçacıklarıyla yapılan çalışmaların her birinde 1 milyon veri içeren aynı veri setleri kullanılmıştır. Uygulamada yapılmış olan her iki çalışma sonucunda herhangi bir hata alınmamış olup, maskeleme ve maskelenen verinin aktarımı işlemleri her iki çalışmada da başarıyla sonuçlandırılmıştır. Tekli iş parçacığıyla yani paralel çalıştırma olmadan yapılan maskeleme ve maskelenen verinin aktarımı işlemi 317 saniyede tamamlanmıştır. Diğer taraftan çoklu iş parçacıklarıyla yani 16 paralel iş parçacığının birbirinden bağımsız olarak çalıştırılmasıyla tamamlanan, maskeleme ve maskelenen verinin aktarımı işlemi 216 saniyede tamamlanmıştır.

5. TARTIŞMA VE SONUÇ

Bu çalışma kapsamında canlı ortam veri tabanında bulunan orijinal verinin, uygulamaların test süreçlerinde orijinal halleriyle kullanılmalarını önlemek ve canlı ortam veri tabanındaki verinin statik veri maskeleye yöntemi ve veri tiplerine uygun olarak kullanılan veri maskeleye teknikleriyle maskelenerek test ortam veri tabanında oluşturulması amaçlanmıştır.

Uygulamalarda test verisinin yönetilebilmesi, uygulama test süreci içerisinde önemli bir alt bileşeni oluşturmaktadır. İhtiyaç duyulan test verisinin oluşturulması aşamasında kişisel verinin korunması bir gereklilik olmuştur. KVKK [4] ile kişisel verinin korunmasına yönelik detaylı bir çalışma başlatılmıştır. Başlatılan bu çalışmayla şirketlerin kişisel verinin güvenliği noktasında veri maskeleye yöntemini kullanması zorunlu hale getirilmiştir.

Jalakas [50]'ın belirttiği gibi, veri maskeleye kişisel verinin yasalara uygun olarak kullanımını, test ve geliştirme ortamları için yapısal olarak benzer ancak kişisel olarak tanımlanamayan veri oluşturulmasını sağlar. Ayrıca veri ihlali olasılığına karşı orijinal verinin korunmasını sağlar. Bu noktada veri maskeleye işleminin kişisel veri güvenliğinin sağlanması açısından önemi aşikardır.

Yapılan çalışma sonucunda oluşturulan test verisi, verinin maskelenmemiş, ham halleriyle karşılaştırıldığında kişisel verinin bulundurulduğu veri tabanından statik veri maskeleye yöntemi kullanılarak test verisi üretilmesinin, kişisel verinin korunması amacıyla kullanılabilecek etkili bir yöntem olduğu gözlemlenmiştir.

Bu çalışmayla kişisel verinin korunması amaçlanarak uygulama testlerinde kullanılacak verinin maskelenmesi örneklendirilmiştir. Ayrıca veri tabanında bulunan veri boyutuna göre maskeleye işleminin zamansal maliyeti değişmektedir, bu kapsamda performansın nasıl artırılacağı yani zamansal maliyetin nasıl düşürülebileceği konusu önem kazanmaktadır.

Araştırmada kullanılacak yazılım seçiminde özellikle ortam bağımsızlığının olması gerekliliği, çoklu kurulum, iş ve iş parçacığı desteklenmesi özellikleri belirleyici rol oynamıştır. Veri maskeleye işlemlerinde öncelikle ortam bağımsızlığı çok önemlidir. Şirketler oluşturmuş oldukları veya kullandıkları uygulamalar için özel verisini birçok farklı veri tabanı platformu kullanarak saklayabilmektedir. Bu noktada veri tabanı şirketleri kendi platformları içerisinde, ortam bağımlı olarak maskeleye işlemini yapabilmektedir. Oracle [62], PostgreSQL [63],

Microsoft [64] vb. veri tabanı şirketlerinin oluşturmuş oldukları veri maskeleye araçları platform bağımlı bir yaklaşım tercih edilmesi ve bundan daha da önemlisi çoklu kurulum, iş ve iş parçacı desteği sağlama konusunda bu araçların kullanımı bulunmadığı için tercih edilmemiştir. IBM şirketinin veri maskeleye aracı olan Optim [65], platform bağımlılığı olmayan, ayrıca çoklu kurulum, iş ve iş parçacı desteği sağlayan lisanslı bir üründür. Birçok şirket tarafından kullanılan bir ürün olmasına karşın lisans ücreti bulunduğu için tercih edilmemiştir. Çoklu kurulum, iş ve iş parçacı desteği maskelenecek verinin boyutunun artması durumunda veri maskeleye işleminin zamansal performans olarak çok daha iyi ve kabul edilebilir oranlarda olması gerekliliği için çok önemlidir. Zira dünya genelinde büyük veri kavramının ortaya çıkması ve şirketlerde saklanan verinin ciddi boyutlarda artması bu özelliğin önemini ortaya koymaktadır. Türkiye’de Informatica şirketinin test veri yönetimi aracı [66], Mayatech firmasının Maya veri maskeleye çözümü [67], Infobox şirketinin, Infobox statik veri maskeleye yazılımı [68], Krontech firmasının maskeleye çözümleri [69], Çalışkan Group’un veri maskeleye yazılımı [70] vb. uygulamalar bulunmaktadır. Dünya genelinde veri maskeleye çözümleri için üretilmiş birçok yazılım ürünü bulunmasına karşın veri güvenliği için kullanılacak bu yazılım aracının yerli ve milli olması tercih edilmiştir. Türkiye’de üretilmiş olan yazılım araçları içerisinde uygulanabilirlik, uygulamaya erişim, ürün kullanımı hakkında tecrübe sahibi olunması, şirket çalışanı olarak lisans ücreti ödemedi kullanabileceğim bir ürün olması ve tez çalışmasında kullanılacak yapı açısından en uygun yazılım aracı olarak *CopyCat* yazılım aracı [61] tercih edilmiştir.

Bu çalışmada seçilen uygulamanın iş ve iş parçacı desteği sağlayabilmesi özelliği kullanılarak bu özelliğin veri maskeleye işleminde ne kadar önemli olduğu sonuçlar çerçevesinde gösterilmiştir. Literatür içerisinde bu çalışmayla veri maskeleye performansı konusunda karşılaştırabilecek benzer bir çalışma bulunmamaktadır. Yapılmış olan çalışmalarda veri güvenliğini sağlama konusunda veri maskeleyenin diğer yöntemler gibi uygulanabilir bir yöntem olduğu ortaya konulmuştur.

Çalışmada statik veri maskeleye yöntemi kullanılmıştır. Bu noktada literatürde belirtildiği gibi; dinamik veri maskeleye yöntemi, verinin gerçek zamanlı maskelenmesini amaçlayan bir teknoloji [2] olduğu için test verisi üretiminde kullanılması uygun görülmemiştir. Anında veri maskeleye yönteminde ise, veri maskeleye ve aktarım işlemi yapıldıktan sonra oluşabilecek herhangi bir veri değişikliği ihtimali veri maskeleye yeteneğini azaltabileceği gibi işlem

maliyetini de arttıracak için [34] bu yöntemde çalışma kapsamında uygun görülmemiştir. Gizleme-Aşırı maskeleme ise, çok güvenli ve verimli bir yol olmasının yanında maliyet açısından çok pahalı bir yöntem [43] olduğu için tercih edilmemiştir.

Jalakas [50], veri maskelemenin olumsuz yönleri arasında; mevcut süreçte ek zaman ihtiyacı doğurabileceğini ve performansı olumsuz etkileyebileceğini belirtmiştir. Bu noktada veri maskeleme performansının nasıl artırılabilirliği konusunda çalışma yapılmıştır.

Çalışmada çoklu (16 adet) iş parçacıklarıyla paralel olarak yapılan veri maskeleme ve maskelenmiş verinin aktarım işleminin, tekli iş parçacığıyla paralel çalıştırma olmadan yapılan maskeleme ve maskelenmiş verinin aktarım işlemine göre %32 oranında zamansal performansı arttırdığı yani %32 oranında zaman tasarrufu sağladığı gözlemlenmiştir. Bu sonuçla günümüz standartlarındaki uygulamalarda bulunan veri miktarlarının gün geçtikçe artmasının, uygulamalar için orijinal veriden maskelenmiş test verisi oluşturma ve maskelenmiş verinin başka ortamlara aktarımı konusunda zamansal performansın ne kadar önemli olduğunu göstermektedir.

Bu tez çalışmasından yola çıkarak ileride yapılabilecek benzer çalışmalar için aşağıdaki önerilere yer verilmiştir;

- İlerleyen zamanlarda artan veri boyutlarına yönelik olarak, zamansal performansın artırılabilirliğini görmek için farklı yazılım araçlarıyla çalışmalar yapılabilir.
- Uygulamalarda tutulan test veri boyutunun artışı ilerisi için yeni maskeleme yöntemlerinin araştırılmasını gerektirebilir. Bu noktada bulut veri maskeleme yöntemi ve içeriğinde bulunduracağı maskeleme teknikleri araştırılabilir.
- Gelecekte ortaya çıkabilecek yeni maskeleme yöntemleriyle beraber kullanılacak veri maskeleme teknikleri de genişletilebilmelidir.
- Benzer olarak yapılacak çalışmalarda kullanılacak veri maskeleme tekniklerinin performansa etkileri araştırılabilir.
- Büyük ve küçük veri setleriyle performans karşılaştırması yapılabileceği gibi, farklı veri türlerinde yapılan veri maskeleme işleminin performansa etkileri de araştırılabilir.
- Veri güvenliğinin sağlanması ve ciddi maliyetlere sebep olan veri ihlallerinin önüne geçebilmek için veri maskeleme ve benzeri teknolojiler üzerine yapılan araştırma sayısının artırılması gerekmektedir.

KAYNAKLAR

- [1]. IBM Security, 2020, *Veri İhlali Maliyet Raporu 2020*, <https://www.ibm.com/tr-tr/security/data-breach>, [Ziyaret Tarihi: 9 Ekim 2020].
- [2]. Watts, S., 2018, *Data Masking: An Introduction – BMC Blogs*, <https://www.bmc.com/blogs/data-masking>, [Ziyaret Tarihi: 10 Ekim 2020].
- [3]. General Data Protection Regulation (GDPR) – Official Legal Text, 2018, *General Data Protection Regulation (GDPR)*, <https://gdpr-info.eu/>.
- [4]. Kişisel Verilerin Korunması Kanunu, 2016, *6698 Kişisel Verilerin Korunması Kanunu*, <https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5>.
- [5]. Kron, 2021, *GDPR ve KVKK döneminde Veri Güvenliğine Hazır mısınız?*, <https://kron.com.tr/gdpr-ve-kvkk-doneminde-veri-guvenligine-hazir-misiniz>, [Ziyaret tarihi: 16 Ocak 2021].
- [6]. Kayhan, O. and Yakupoglu, B., 2017, *Test Veri Yönetiminde Alternatif Bir Yaklaşım, Test Veri Yönetimi Otomasyonu*, http://ceur-ws.org/Vol-1980/YTM_2017_paper_5.pdf, [Ziyaret Tarihi: 15 Kasım 2020].
- [7]. Kazan, H., 2016, *Bilimsel Araştırma Teknikleri*, https://cdn-acikogretim.istanbul.edu.tr/auzefcontent/19_20_Guz/bilimsel_arastirma_teknikleri/9/index.html#konu-2, [Ziyaret Tarihi: 21 Kasım 2020].
- [8]. Yılmaz, M., 2009, Enformasyon Ve Bilgi Kavramları Bağlamında Enformasyon Yönetimi Ve Bilgi Yönetimi, *Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Dergisi*, 095-118, https://doi.org/10.1501/Dtcfder_0000001205.
- [9]. Prytherch, R. J., 2005, Harrod's librarians' glossary and reference book: A directory of over 10,200 terms, organizations, projects and acronyms in the areas of information management, library science, publishing and archive management, 10th ed, *Ashgate*.
- [10]. Şeker, Ş. E., 2013, *İş Zekası ve Veri Madenciliği*, Cinius, İstanbul, ISBN: 978-605-127-671-7
- [11]. Rowley, J., 2007, The wisdom hierarchy: Representations of the DIKW hierarchy, *Journal of Information Science*, 33(2), 163-180, <https://doi.org/10.1177/0165551506070706>.
- [12]. Merriam-Webster, 1828, *Definition of Data*, <https://www.merriam-webster.com/dictionary/data>, [Ziyaret Tarihi: 07 Aralık 2020].
- [13]. Zins, C., 2007, Conceptual approaches for defining data, information, and knowledge, *Journal of the American Society for Information Science and Technology*, 58(4), 479-493, <https://doi.org/10.1002/asi.20508>.

- [14]. Hey, J., 2004, The data, information, knowledge, wisdom chain: The metaphorical link, *Intergovernmental Oceanographic Commission*, 26, 1-18, <http://www.dataschemata.com/uploads/7/4/8/7/7487334/dikwchain.pdf>.
- [15]. Jensen, P., 2005, *A contextual theory of learning and the learning organization*, Knowledge and Process Management, 12, 53-64, <https://doi.org/10.1002/kpm.217>.
- [16]. Akgül, A., 2014, *Danıştay ve Avrupa İnsan Hakları Mahkemesi Kararları Işığında Kişisel Verilerin Korunması*, Beta Yayıncılık.
- [17]. Akkurt, S. S., 2020, Kişisel Veri Kavramının Hukukî Niteliğine İlişkin Yaklaşımlara Mukayeseli Bir Bakış, *Kişisel Verileri Koruma Dergisi*, 2(1), 20-32, <https://dergipark.org.tr/tr/pub/kvkd/726372>.
- [18]. Er, C., 2007, *Biyometrik yöntemler ve özel hayatın gizliliği hakkı: Parmak izi, göz ve DNA tarama gibi teknolojik kimlik denetleme usullerinin hukuki statüsü*, Yetkin.
- [19]. Henkoğlu, T., 2015, *Bilgi güvenliği ve kişisel verilerin korunması*, Yetkin Yayınları.
- [20]. Özdemir, H., 2009, *Elektronik haberleşme alanında kişisel verilerin özel hukuk hükümlerine göre korunması*, Seçkin.
- [21]. Şimşek, O., 2008, *Anayasa hukukunda kişisel verilerin korunması*, 1. baskı, Beta.
- [22]. Akkurt, S. S., 2019, *Sosyal medyada gerçekleşen ihlaller karşısında kişilik hakkının korunması*, Seçkin, ISBN: 978-975-02-5534-2.
- [23]. Resmî Gazete, 2014, 6563 Elektronik Ticaretin Düzenlenmesi Hakkında Kanun, <https://www.resmigazete.gov.tr/eskiler/2014/11/20141105-1.htm>, [Ziyaret Tarihi: 11 Aralık 2020].
- [24]. Resmî Gazete, 2015, *Ticari İletişim Ve Ticari Elektronik İletiler Hakkında Yönetmelik*, <https://www.resmigazete.gov.tr/eskiler/2015/07/20150715-4.htm>, [Ziyaret Tarihi: 12 Aralık 2020].
- [25]. Dülger, D. D. M. V., 2020, *Kişisel Verileri Koruma Kurulunun “Mesaj ve Aramalara İlişkin” 16/10/2018 Tarihli ve 2018/119 Sayılı Kararına İlişkin Değerlendirme*, https://www.academia.edu/37691477/Kişisel_Verileri_Koruma_Kurulunun_Mesaj_ve_Aramalara_İlişkin_16_10_2018_Tarihli_ve_2018_119_Sayılı_Kararına_İlişkin_Değerlendirme, [Ziyaret Tarihi: 07 Aralık 2020].
- [26]. Keser Berber, L., 2014, *Çevrimiçi Davranışsal Reklamcılık (Online Behavioral Advertising) Uygulamaları Özelinde Kişisel Verilerin Korunması*, İstanbul, XII Levha Yayıncılık.
- [27]. Fuchs, C., 2011, *An Alternative View of Privacy on Facebook*, Information, 2(1), 140-165, <https://doi.org/10.3390/info2010140>.
- [28]. Karlıdağ, S., 2013, *Personal Data Protection from the Political Economy Perspective*, Amme İdaresi Dergisi, 46, 127-152.

- [29]. Dhir, P. and Garg, S., 2017, Survey on Cloud Computing and Data Masking Techniques, *International Journal of Innovation & Advancement in Computer Science*, 6(4), 1-7.
- [30]. Beyaz.net, 2020, *Veri maskeleye nedir? Türleri nelerdir?*, https://www.beyaz.net/tr/guvenlik/makaleler/veri_maskeleye_nedir_turleri_nelerdir.html, [Ziyaret Tarihi: 16 Eylül 2020].
- [31]. Colak, F., 2020, *Veri maskeleye nedir? Türleri nelerdir?*, <https://www.fatihcolak.com.tr/veri-maskeleye-nedir-turleri-nelerdir.html>, [Ziyaret Tarihi: 17 Eylül 2020].
- [32]. Li, M., Liu, Z., Jia, C. and Dong, Z., 2013, *Data Masking Generic Model*, 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies, 724-727, <https://doi.org/10.1109/EIDWT.2013.131>.
- [33]. Thengdi, S., Nair, S., Deshmukh, N. R., Kawley, K. and Narnaware, M., 2015, *Masking of Data For ERP Test Environment*, <https://doi.org/10.1.1.736.1335>.
- [34]. G K, R., Rabi, B. J., T N, M., Hegadi, R. and A., Archana. R., 2011, A Survey on Recent Trends, Process and Development in Data Masking for Testing, *International Journal of Computer Science Issues (IJCSI)*, 8(2), 535-544.
- [35]. G K, R., Rabi, B. J., Hegadi, R., T N, M. and Archana, R., 2011, *Experimental Study of Various Data Masking Techniques with Random Replacement using Data Volume*.
- [36]. G K, R., T N, M. and Hegadi, R., 2011, Design of Data Masking Architecture and Analysis of Data Masking Techniques for Testing, *International Journal of Engineering Science*, 3.
- [37]. PRNEWS, 2012, *Hacking at LinkedIn Highlights PR's Critical Role in Data Breaches*, <https://www.prnewsonline.com/hacking-at-linkedin-highlights-prs-critical-role-in-data-breaches>, [Ziyaret Tarihi: 24 Ekim 2020].
- [38]. Oracle, 2013, *Data Masking Best Practice*, An Oracle White Paper, <http://www.oracle.com/us/products/database/data-masking-best-practices-161213.pdf>, [Ziyaret Tarihi: 05 Kasım 2020].
- [39]. Ponemon Institute, 2012, *2012 Consumer Study on Data Breach Notification*, <https://www.experian.com/assets/data-breach/brochures/ponemon-notification-study-2012.pdf>, [Ziyaret Tarihi: 18 Ekim 2020].
- [40]. Catherine, K. S. S. B. A. and Sreeja, V., 2018, A comprehensive Study of Data Masking Techniques on cloud, *International Journal of Pure and Applied Mathematics*, 119, 3719-3727.
- [41]. A Net 2000 Ltd., 2000, *Data Masking: What You Need to Know*, White Paper, http://www.datamasker.com/DataMasking_WhatYouNeedToKnow.pdf, [Ziyaret Tarihi: 08 Kasım 2020].

- [42]. Delphix, 2020, *Best Practice: File Masking Job Configuration (KBA1821)*, [https://support.delphix.com/Delphix_Masking_Engine/Best_Practice%3A_File_Masking_Job_Configuration_\(KBA1821\)](https://support.delphix.com/Delphix_Masking_Engine/Best_Practice%3A_File_Masking_Job_Configuration_(KBA1821)), [Ziyaret Tarihi: 14 Kasım 2020].
- [43]. Hush-Hush, 2015, *Data Masking Types*, <https://mask-me.net/datamaskingwiki/wiki/27/data-masking-types>, [Ziyaret Tarihi: 24 Kasım 2020].
- [44]. Vijayarani, S. and Tamilarasi, A., 2011, An efficient masking technique for sensitive data protection, *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, 1245-1249, <https://doi.org/10.1109/ICRTIT.2011.5972275>.
- [45]. Cuzzocrea, A. and Shahriar, H., 2017, Data masking techniques for NoSQL database security: A systematic review, *2017 IEEE International Conference on Big Data (Big Data)*, 4467-4473, <https://doi.org/10.1109/BigData.2017.8258486>.
- [46]. AI Multiple, 2020, *Data Masking: What it is, how it works, types & best practices*, <https://research.aimultiple.com/data-masking>, [Ziyaret Tarihi: 10 Aralık 2020].
- [47]. Kumargk, R., B Justus Rabi, D. and Tn, M., 2012, A Study on Dynamic Data Masking with its Trends and Implications, *International Journal of Computer Applications*, 38(6), 19-24, <https://doi.org/10.5120/4612-6828>.
- [48]. Mattsson, U. T., 2005, Database Encryption - How to Balance Security with Performance, *SSRN Electronic Journal*, <https://doi.org/10.2139/ssrn.670561>.
- [49]. Olusola Olajide, A., 2014, Application of Data Masking in Achieving Information Privacy, *IOSR Journal of Engineering*, 4, 13-21, <https://doi.org/10.9790/3021-04211321>.
- [50]. Jalakas, O., 2016, *Data Masking and User Rights in Data Warehouse to Protect Data*, PhD Thesis, Tallinn University of Technology, Tallinn.
- [51]. Kavaklıoğlu, T., 2016, *Test Veri Yönetiminin Önemi ve Çözüm Getirebileceği Konular*, LinkedIn, <https://www.linkedin.com/pulse/test-veri-yönetiminin-önemi-ve-çözüm-getirebileceği-tunç-kavaklıoğlu/?originalSubdomain=tr>, [Ziyaret Tarihi: 17 Aralık 2020].
- [52]. Bourque, P., Fairley, R. E. and IEEE Computer Society., 2014, *Guide to the software engineering body of knowledge*, ISBN: 978-0-7695-5166-1.
- [53]. Vikipedi, 2020, *Yazılım testi*, https://tr.wikipedia.org/wiki/Yazılım_testi, [Ziyaret Tarihi: 25 Aralık 2020].
- [54]. Güleç, G., 2020, *Yazılım Test Türleri*, <http://gizemgulec.com/2020/04/yazilim-test-turleri/>, [Ziyaret Tarihi: 25 Aralık 2020].
- [55]. Hillenius, G., Eveleens, L., Kampstra, P., Kwiatkowski, L. and Sman, S., 2007, EQUITY 2007 Report, *2007 IEEE International Conference on Exploring Quantifiable IT Yields*, 1-8, <https://doi.org/10.1109/EQUITY.2007.9>.

- [56]. Şentürk, T., Aydemir, İ. and Eker, B., 2007, *Ölçümlemeli Test Platformu*, 11.
- [57]. Khurana, P. and Bindal, P., 2014, Test Data Management, *International Journal of Computer Trends and Technology*, 15(14), 162.
- [58]. Keytorc Yazılım Test Hizmetleri, 2017, *Test Manager Insights Report 2016-17*, <https://keytorc.com/en/test-publications/test-manager-insights-report-2016-17/>, [Ziyaret Tarihi: 18 Aralık 2020].
- [59]. Turkish Testing Board, 2016, *Türkiye Yazılım Kalite Raporu/ Turkey Software Quality Report (TSQR) 2016-17*, <https://www.turkishtestingboard.org/files/Turkey-Software-Quality-Report-2016-17.pdf>, [Ziyaret Tarihi: 18 Aralık 2020].
- [60]. Kудay, G., 2014, *Yazılım mühendisliği yöntemleriyle yazılım test süreci*, PhD Thesis, Fen Bilimleri Enstitüsü.
- [61]. Veriskop Bilişim Ve Danışmanlık Hizmetleri A.Ş., 2020, *CopyCat*, <https://www.veriskop.com/copycat>, [Ziyaret Tarihi: 02 Ocak 2021].
- [62]. Oracle Türkiye, 2020, *Oracle Data Masking and Subsetting Pack*, <https://www.oracle.com/tr/database/technologies/security/data-masking-subsetting.html>, [Ziyaret Tarihi: 10 Ocak 2021].
- [63]. PostgreSQL, 2020, *PostgreSQL Anonymizer*, <https://postgresql-anonymizer.readthedocs.io/en/latest>, [Ziyaret Tarihi: 28 Aralık 2020].
- [64]. Microsoft Azure, 2020, *Static Data Masking for Azure SQL Database and SQL Server*, <https://azure.microsoft.com/tr-tr/blog/static-data-masking-preview>, [Ziyaret Tarihi: 28 Aralık 2020].
- [65]. IBM, 2020, *InfoSphere Optim Data Privacy-Overview*, <https://www.ibm.com/products/infosphere-optim-data-privacy>, [Ziyaret Tarihi: 23 Aralık 2020].
- [66]. Informatica, 2020, *Test Data Management Tool*, <https://www.informatica.com/products/data-security/test-data-management.html>, [Ziyaret Tarihi: 25 Aralık 2020].
- [67]. Mayatech, 2020, *Maya Data Masking Technologies*, <http://www.mayatech.com.tr/maya-data-masking-solution/tdmtechnologies>, [Ziyaret Tarihi: 26 Aralık 2020].
- [68]. Infobox, 2020, *Infobox - Dinamik ve Statik Veri Maskeleyme*, LinkedIn, <https://www.linkedin.com/pulse/infobox-dinamik-ve-statik-veri-maskeleyme-hüseyin-ümit-ünsoy/?originalSubdomain=tr>, [Ziyaret Tarihi: 28 Aralık 2020].
- [69]. Krontech, 2020, *Database Access Manager & Dynamic Data Masking*, <https://krontech.com/database-access-manager-dynamic-data-masking>, [Ziyaret Tarihi: 28 Aralık 2020].

- [70]. Çalışkan Group, 2020, *Veri Maskeleye Uygulaması*,
<https://www.caliskangroup.com.tr/veri-maskeleye-yazilimi>, [Ziyaret Tarihi: 28 Aralık 2020].



ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Fatih Enes Ekren
Doğum Yeri	
Doğum Tarihi	Tarih girmek için tıklayın veya dokununuz.
Uyruğu	☒ T.C. ☐ Diğer:
Telefon	
E-Posta Adresi	
Web Adresi	

Eğitim Bilgileri	
Lisans	
Üniversite	Yıldız Teknik Üniversitesi
Fakülte	Elektrik Elektronik Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	2015

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Enformatik
Programı	Enformatik