# Proje 5: Veri Temizleme ve ETL Süreçleri Tasarımı

## 1. Proje Tanımı ve Amaç

Bu projenin amacı, gerçek dünya verilerinde sıkça karşılaşılan hatalı, eksik veya tutarsız verilerin temizlenmesi, dönüştürülmesi ve hedef veritabanına yüklenmesi sürecini simüle etmektir. ETL (Extract – Transform – Load) olarak bilinen bu süreç, veri ambarları, iş zekâsı sistemleri ve raporlama altyapılarında kritik öneme sahiptir.

Proje kapsamında:

* Kirli veriler içeren bir **staging (geçici)** tablo oluşturulmuştur.
* Bu tabloya manuel olarak **eksik, hatalı formatlı ve geçersiz veriler** girilmiştir.
* Ardından bu veriler **transformasyon süreci** ile denetlenmiş ve yalnızca geçerli olanlar ayrıştırılmıştır.
* Temiz veriler ayrı bir tabloya aktarılmış ve hedef sistemde kullanılmaya hazır hâle getirilmiştir.

Bu süreç sayesinde:

* SQL Server kullanılarak temel veri temizleme kontrolleri uygulanmıştır.
* Geçersiz e-posta adresleri ve tarih formatları filtrelenmiştir.
* Boş değerlerin ayıklanması sağlanmıştır.
* Tüm işlem sonunda veri kalitesi raporları hazırlanarak ne kadar verinin temizlendiği ve yüklendiği analiz edilmiştir.

Bu proje, ETL süreçlerinin temel mantığını öğretmekle birlikte, gerçek projelerde karşılaşılabilecek veri temizlik ve dönüşüm ihtiyaçlarına da birebir örnek teşkil etmektedir.

## 2. Kullanılan Araçlar ve Ortam Bilgisi

| **Bileşen** | **Açıklama** |
| --- | --- |
| **Veritabanı Sunucusu** | Microsoft SQL Server 2019 |
| **Yönetim Aracı** | SQL Server Management Studio (SSMS) 18.x |
| **Proje Veritabanı** | ETLProject |
| **İşletim Sistemi** | Windows 10 / 11 (64-bit) |
| **Kodlama Dili** | T-SQL (Transact-SQL) |
| **Kullanılan Tablolar** | Staging\_Customers, Clean\_Customers, Final\_Customers |
| **Veri Giriş Şekli** | Manuel olarak INSERT komutları ile |
| **Hedef** | Temiz verilerin Final\_Customers tablosuna aktarılması |

Bu ortam, ETL sürecinin SQL tabanlı olarak uygulandığı bir örnek senaryoyu temsil eder. Geliştirme ortamı düşük kaynaklara sahip standart bir bilgisayarda dahi sorunsuz çalışabilecek şekilde tasarlanmıştır. SQL Server’ın veri kontrol fonksiyonları (örneğin ISDATE, LIKE) kullanılarak temel kurallara uygun veri seçimi ve filtreleme işlemleri gerçekleştirilmiştir.

## 3. Staging (Geçici) Veritabanı Tasarımı

ETL sürecinin ilk adımı olan "Extract" aşamasında, veriler öncelikle doğrudan hedef sisteme aktarılmak yerine **staging (ara/geçici)** olarak adlandırılan bir alana alınır. Bu geçici alan, veriler üzerinde çeşitli kalite kontrolleri yapılmadan önce bir tampon görevi görür.

Bu proje kapsamında staging yapısı için Staging\_Customers adında bir tablo oluşturulmuştur. Bu tabloya veri kaynaklarından gelen hatalı, eksik veya format açısından uyumsuz veriler manuel olarak eklenmiştir.

**Oluşturulan Geçici Tablonun Yapısı:**

| **Kolon Adı** | **Veri Tipi** | **Açıklama** |
| --- | --- | --- |
| ID | INT | Müşteri numarası |
| FullName | NVARCHAR(100) | Müşterinin tam adı |
| Email | NVARCHAR(100) | Müşteri e-posta adresi |
| BirthDate | NVARCHAR(50) | Doğum tarihi (hatalı formatlara izinli) |

Tablonun dikkat çeken yönü, BirthDate alanının NVARCHAR olarak tanımlanmış olmasıdır. Bu sayede tarih alanı içeren hatalı formatların tabloya kabul edilmesi sağlanmıştır. Böylece verinin gerçek hayattaki "kirli veri" senaryosunu temsil etmesi mümkün olmuştur.

**Oluşturma Komutu:**



## 4. Veri Girişi ve Kirli Veri Örnekleri

Staging\_Customers tablosuna aşağıdaki örnek veri seti manuel olarak eklenmiştir. Bu veriler çeşitli hatalar barındırmaktadır ve ETL süreci boyunca bu hatalar filtrelenecektir.

**Eklenen Kirli Veri Örnekleri:**

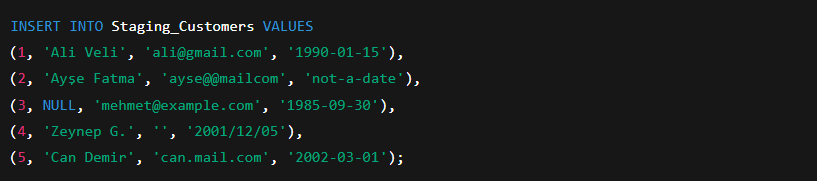
| **ID** | **FullName** | **Email** | **BirthDate** |
| --- | --- | --- | --- |
| 1 | Ali Veli | ali@gmail.com | 1990-01-15 |
| 2 | Ayşe Fatma | ayse@@mailcom | not-a-date |
| 3 | *NULL* | mehmet@example.com | 1985-09-30 |
| 4 | Zeynep G. | *(boş)* | 2001/12/05 |
| 5 | Can Demir | can.mail.com | 2002-03-01 |

**Veri Setindeki Belirgin Hatalar:**

* ID: 2 → E-posta biçimi geçersiz (@ işareti iki kez)
* ID: 3 → Ad-soyad boş (NULL)
* ID: 4 → E-posta boş (empty string)
* ID: 5 → E-posta @ içermiyor (biçimsel hata)
* ID: 2 → Doğum tarihi formatı geçersiz (not-a-date)

Bu veriler, projede uygulanacak olan temizleme kurallarına tabidir. Temizleme işlemleriyle yalnızca **tam, tutarlı ve geçerli** kayıtlar ayrıştırılarak sonraki aşamalara aktarılacaktır.

**Veri Ekleme Komutu:**



Bu noktada staging tablosuna alınan veriler, bir sonraki adım olan **veri temizleme (transform)** aşamasına hazır hâle getirilmiştir.

## 5. Veri Temizleme (Transformasyon) Süreci

Veri temizleme (transform) aşaması, ETL sürecinin en kritik adımıdır. Bu adımda, Staging\_Customers tablosuna alınan hatalı veya eksik veriler tespit edilerek elenir ve yalnızca geçerli kayıtlar ayrıştırılır. Böylece hatalı verilerin hedef sistemlere aktarılması engellenmiş olur.

Bu projede uygulanan transformasyon süreci 3 temel kontrol üzerine kurulmuştur:

**5.1. E-Posta Geçerlilik Kontrolü**

Gerçek hayatta en sık karşılaşılan veri sorunlarından biri **geçersiz e-posta adresleri**dir. Bu projede basit ama etkili bir yöntem kullanılarak e-posta geçerliliği kontrol edilmiştir:

**Kural:**

E-posta adresi şunları içermelidir:

* @ karakteri
* . karakteri
* Uygun uzunlukta alanlar

**Kullanılan Sorgu:**

**Sonuç:**

Bu sorguyla 3. kayıttaki boş e-posta ('') ve ayse@@mailcom, can.mail.com gibi hatalı formatlar tespit edilmiştir.

**5.2. Doğum Tarihi Format Kontrolü**

Veri tabanlarında tarihlerin yanlış formatta girilmesi, filtreleme ve sıralama işlemlerinde sorunlara yol açar. Bu nedenle BirthDate sütununda geçerli tarih formatı kontrol edilmiştir.

**Kullanılan Fonksiyon:**

SQL Server’ın yerleşik ISDATE() fonksiyonu, bir değerin geçerli bir tarih olup olmadığını test eder.

**Sorgu:**

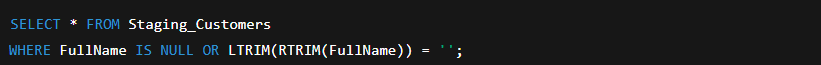
**Sonuç:**

* not-a-date (ID: 2) geçersiz olarak işaretlenmiştir.
* 2001/12/05 ifadesi geçerli formatta kabul edilmiştir çünkü SQL Server, bu formatı da tanıyabilir.

**5.3. Eksik Veri Tespiti ve Eleme**

Bazı kayıtlar teknik olarak geçerli olsa da eksik bilgi içermeleri nedeniyle işlenmeye uygun değildir. Özellikle FullName alanının boş olması, müşteri verisinin anlamlılığını kaybettirir.

**Sorgu:**

**Sonuç:**

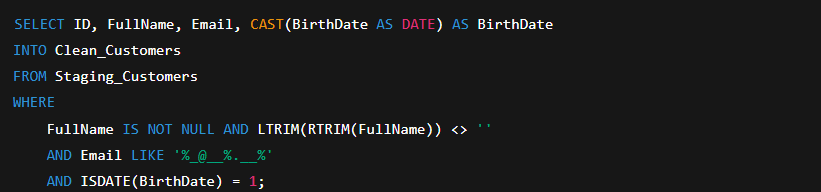
* NULL olan veya tamamen boş FullName değerleri (örneğin ID: 3) temizleme işleminde elenmiştir.

**🧹 Genel Temizleme Kriterleri**

| **Kontrol Türü** | **Koşul** |
| --- | --- |
| E-posta geçerli mi? | Email LIKE '%\_@\_\_%.\_\_%' |
| Tarih geçerli mi? | ISDATE(BirthDate) = 1 |
| Ad-soyad dolu mu? | FullName IS NOT NULL AND LTRIM(RTRIM(FullName)) <> '' |

**✅ Temiz Verilerin Ayrıştırılması**

Tüm bu kuralları birleştirerek yalnızca geçerli kayıtları içeren bir Clean\_Customers tablosu oluşturulmuştur:

Bu tablo yalnızca geçerli e-posta adresi, geçerli doğum tarihi ve boş olmayan ad-soyad bilgisi içeren müşterileri barındırır. Bir sonraki aşamada bu veriler hedef tabloya (Final\_Customers) yüklenecektir.

## 6. Temiz Verilerin Ayrıştırılması (Clean Table)

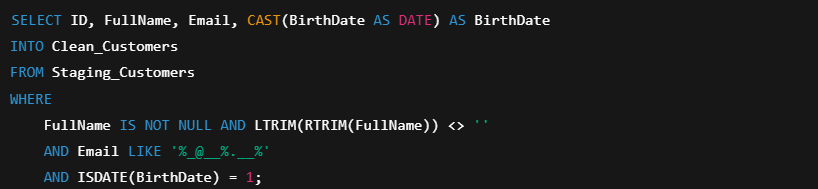
Veri temizleme (transformasyon) işlemi sonucunda geçerli kabul edilen veriler, ayrı bir tabloda toplanarak staging alanından ayrıştırılmıştır. Bu işlem sayesinde hem temiz veriler kolayca yönetilebilir hâle gelmiş, hem de staging tablosu bozulmadan arşivlenebilir kalmıştır.

**Oluşturulan Temiz Tablo: Clean\_Customers**

Bu tablo, aşağıdaki filtreleme kurallarını geçen verileri içerir:

* FullName alanı boş değil (NOT NULL)
* Email adresi geçerli biçimde (LIKE '%\_@\_\_%.\_\_%')
* BirthDate geçerli bir tarih (ISDATE(BirthDate) = 1)

**Kullanılan SQL Komutu:**

**Sonuç:**

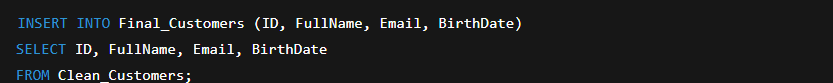
Yukarıdaki komut, verileri Clean\_Customers adında yeni bir tabloya kopyalayarak hem veri temizliğini tamamlamış hem de dönüştürme işlemini (string tarihi → DATE) gerçekleştirmiştir.

## 7. Veri Yükleme (Load) – Final Tablonun Oluşturulması

ETL sürecinin son adımı olan “Load”, temizlenmiş ve dönüştürülmüş verilerin nihai/verimli kullanıma hazır veritabanı yapısına aktarılması işlemidir. Bu projede Clean\_Customers tablosunda yer alan veriler, hedef tablo olan Final\_Customers tablosuna yüklenmiştir.

**Oluşturulan Final Tablo:**

**Temiz Verilerin Yüklenmesi:**

**Açıklama:**

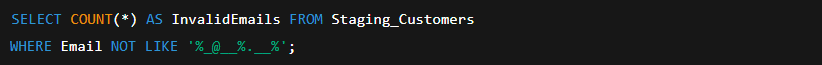
Bu işlem sayesinde yalnızca güvenilir ve kaliteli veriler sistemde kullanılabilir hâle getirilmiştir. Final\_Customers tablosu artık raporlama, analiz veya dış sistemlere veri aktarımı için uygundur.

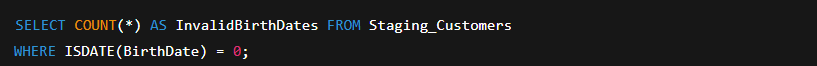
## 8. Veri Kalitesi Raporu

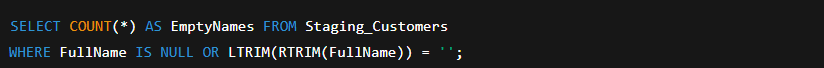
Veri temizleme sürecinin sonunda işlemin ne kadar etkili olduğunu görmek için veri kalitesi raporu hazırlanmıştır. Bu rapor, orijinal staging tablosundaki tüm verilerle karşılaştırmalı olarak aşağıdaki bilgileri sunar:

**8.1. Toplam Kayıt Sayısı:**

**8.2. Geçersiz E-Posta Sayısı:**

**8.3. Geçersiz Doğum Tarihi Sayısı:**

**8.4. Eksik (NULL veya boş) Ad-Soyad Sayısı:**

**8.5. Temizlenip Final Tabloda Yüklenen Kayıt Sayısı:**



**📝 Örnek Rapor Sonucu:**

| **Kategori** | **Kayıt Sayısı** |
| --- | --- |
| Toplam Kayıt (Staging) | 5 |
| Geçersiz E-Posta | 2 |
| Geçersiz Doğum Tarihi | 1 |
| Eksik İsim | 1 |
| Temizlenmiş Kayıt (Final) | 2 |

**Değerlendirme:**

* Giriş verisinin %60’ı çeşitli nedenlerle elenmiştir.
* Yalnızca %40 oranında veri temiz olarak sisteme aktarılmıştır.
* Bu oran, kaynağın ne kadar kirli olduğuna dair fikir verir ve gerçek hayatta veri kalitesi politikasının önemini vurgular.

## 9. Karşılaşılan Sorunlar ve Çözümler

Veri temizleme sürecinde çeşitli teknik ve mantıksal sorunlarla karşılaşılmış ve her biri için uygun çözüm yöntemleri uygulanmıştır. Bu bölümde bu sorunlar ve çözümler özetlenmiştir:

**🔸 Sorun 1: Geçersiz E-Postaların Tespiti**

* **Açıklama:** SQL Server'da e-posta doğrulama için doğrudan kullanılabilecek gelişmiş bir fonksiyon bulunmamaktadır.
* **Sorun:** LIKE '%@%.%' gibi basit bir ifade yetersiz kaldı.
* **Çözüm:** Daha güvenli bir ifade olan LIKE '%\_@\_\_%.\_\_%' kullanılarak hem uzunluk hem yapı denetimi sağlandı.

**🔸 Sorun 2: Tarih Formatı Hataları**

* **Açıklama:** BirthDate sütununda farklı formatlarda tarih verileri vardı. Örneğin not-a-date gibi değerler tabloda yer alıyordu.
* **Sorun:** Bu tür değerler CAST işlemi sırasında hata çıkarabilir.
* **Çözüm:** Dönüştürme işleminden önce ISDATE() fonksiyonu ile geçerli tarih filtresi uygulandı.

**🔸 Sorun 3: Boş Değerlerin Tespiti (NULL veya boşluk)**

* **Açıklama:** FullName alanı bazı kayıtlarda boş bırakılmıştı.
* **Sorun:** NULL dışında sadece boşluk içeren değerlerin de tespit edilmesi gerekiyordu.
* **Çözüm:** LTRIM(RTRIM(FullName)) = '' kontrolü ile hem boşluk hem de NULL değeri olan kayıtlar filtrelendi.

**🔸 Sorun 4: Veri Tipi Dönüşümünde Hata Riski**

* **Açıklama:** BirthDate sütunu başlangıçta NVARCHAR olarak tanımlandığı için tarih tipine dönüştürülürken hata riski vardı.
* **Çözüm:** Dönüştürmeden önce ISDATE(BirthDate) = 1 filtresi uygulandı, ardından CAST(... AS DATE) işlemi güvenle yapıldı.

## 10. Sonuç ve Değerlendirme

Bu proje kapsamında, gerçek dünyada sıkça karşılaşılan veri kirliliği problemleri simüle edilerek başarılı bir ETL süreci yürütülmüştür. Projenin sonunda aşağıdaki kazanımlar elde edilmiştir:

**✅ ETL Sürecinin Temel Aşamaları Uygulandı:**

* **Extract:** Kirli veriler staging tablosuna alındı.
* **Transform:** Veriler üzerinde e-posta kontrolü, tarih formatı kontrolü ve eksik değer kontrolü yapıldı.
* **Load:** Geçerli kayıtlar hedef tabloya taşındı.

**📊 Veri Kalitesi Artırıldı:**

* Giriş verilerinin %60’ı hatalı olduğu için elendi.
* Temiz ve güvenli veri oranı %40 olarak hedef tabloya aktarıldı.
* Bu da veri ambarlarında kalite kontrolün ne kadar önemli olduğunu ortaya koydu.

**💡 Kazanımlar:**

* SQL Server ile veri temizleme mantığı pratik edildi.
* ISDATE, LIKE, LTRIM, RTRIM, CAST, SELECT INTO gibi fonksiyon ve yapılar öğrenildi.
* Temiz ve güvenilir verilerin iş süreçlerine etkisi fark edildi.
* ETL sürecinin manuel simülasyonu, büyük veri sistemlerinde bu işlemlerin neden otomasyonla yapılması gerektiğini anlamayı sağladı.

**Genel Değerlendirme:**  
Proje, temel düzeyde bir ETL sürecini başarıyla modellemiş ve verinin sistemlere aktarılmadan önce neden doğrulanması gerektiğini açıkça ortaya koymuştur. Gerçek projelerde bu mantık; SSIS, Talend, Azure Data Factory gibi araçlarla otomatikleştirilir. Ancak bu proje, sürecin mantığını kavramak için oldukça etkili olmuştur.