**Proje 5: Veri Temizleme ve ETL Süreçleri Tasarımı**

**Kullanılan veri tabanı:** Wide World Importers sample database v1.0

**Link:** <https://github.com/Microsoft/sql-server-samples/releases/tag/wide-world-importers-v1.0>

**Video Linki:** [**https://drive.google.com/file/d/17ovd3MWnzschdjL8cv2JjVr\_zxQ\_6Hbt/view?usp=sharing**](https://drive.google.com/file/d/17ovd3MWnzschdjL8cv2JjVr_zxQ_6Hbt/view?usp=sharing)

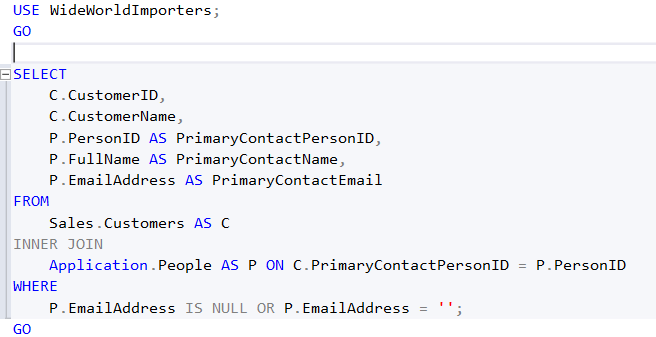
Bu projede istenen adımlar 4 temel başlık altında değerlendirilmiş. 1- Veri temizleme 2- Veri Dönüştürme, 3- Veri Yükleme, 4- Veri Kalitesi Raporları. Biz de yaptığımız adımları bu başlıklar üzerinde değerlendirecek olursak.

**Bölüm 1: Veri Temizleme**

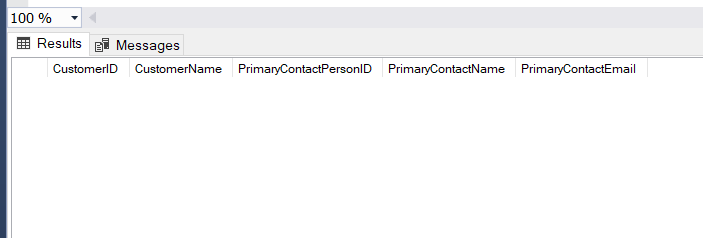
Bu bölümde, Wide World Importers veritabanındaki verilerin kalitesini artırmak amacıyla hatalı, eksik veya tutarsız verileri tespit edip SQL kullanarak temizleme işlemleri gerçekleştireceğiz.

**Adım 1.1: Eksik Birincil Kontak E-posta Adresine Sahip Müşterilerin Tespiti**

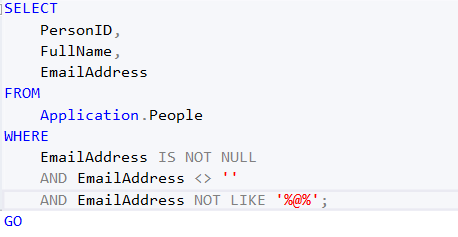
Bu adımda, Sales.Customers tablosu ile Application.People tablosu birleştirilerek, müşterilerin birincil iletişim kişilerine ait e-posta adreslerinin eksik (NULL veya boş) olup olmadığı kontrol edilmiştir.



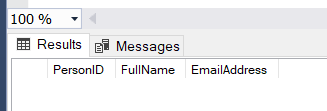
Yukarıdaki sorgunun sonucunun boş olması Wide World Importers veritabanındaki müşterilerin birincil iletişim kişilerine ait e-posta adresi alanında eksik veri (NULL veya boş) bulunmadığını gösteriyor.

****

**Adım 1.2: Geçersiz E-posta Formatı Kontrolü**

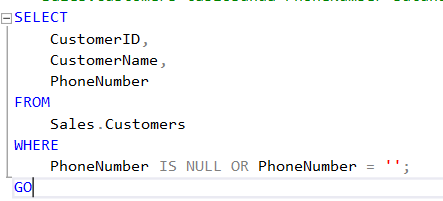
Bu adımda, Application.People tablosundaki EmailAddress sütununda '@' karakteri içermeyen, yani geçersiz formatta olabilecek e-posta adresleri kontrol edilecektir.

Yukarıdaki sorgunun sonucunda, '@' karakteri içermeyen ve potansiyel olarak geçersiz formattaki e-posta adresleri listelenir. Eğer sorgu sonuç döndürürse, bu kayıtların incelenmesi ve düzeltilmesi gerekir. Eğer sorgu sonuç döndürmezse, **Wide World Importers** veritabanındaki tüm e-posta adreslerinin en azından '@' karakteri içerdiği anlaşılır. Aşağıdaki durumda da bu olay mevcuttur.

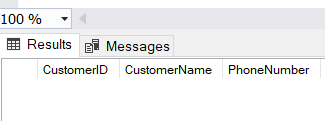


**Adım 1.3: Eksik Telefon Numarası Kontrolü**

Bu adımda, Sales.Customers tablosundaki PhoneNumber sütununda eksik (NULL veya boş) değer olup olmadığı kontrol edilecektir.



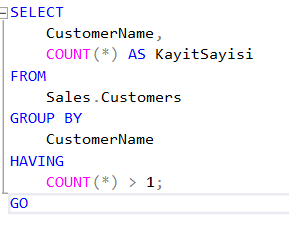
Yukarıdaki sorgunun sonucunda, telefon numarası bilgisi eksik olan müşteriler listelenir. Eğer sorgu sonuç döndürürse, bu kayıtların iletişim bilgileri eksiktir ve güncellenmesi gerekebilir. Eğer sorgu sonuç döndürmezse, **Wide World Importers** veritabanındaki tüm müşterilerin PhoneNumber alanının dolu olduğu anlaşılır. Aşağıda bunu doğrulayan sorgu sonucunu görüntüleyebiliriz.

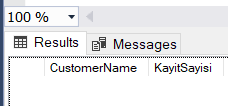


Şu ana kadar yaptığımız kontrollerde (eksik e-posta, geçersiz e-posta formatı, eksik telefon numarası) herhangi bir veri kalitesi sorununa rastlamadık. Bu, Wide World Importers örnek veritabanının bu alanlar açısından oldukça temiz olduğunu gösteriyor.

**Adım 1.4: Potansiyel Tekrarlı Müşteri Kayıtları Kontrolü**

Bu adımda, **Sales.Customers** tablosunda aynı CustomerName değerine sahip, yani potansiyel olarak tekrarlı olabilecek müşteri kayıtları sorgulanacaktır.



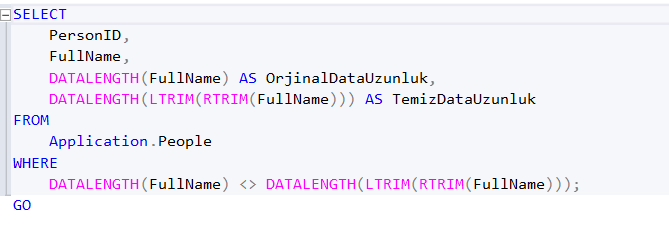


Sonuç kümesinin boş olması (0 satır), **Wide World Importers** veritabanındaki **Sales.Customers** tablosunda aynı CustomerName değerine sahip birden fazla müşteri kaydı bulunmadığını göstermektedir. Bu kontrol sonucunda potansiyel tekrarlı kayıtlara rastlanmamıştır.

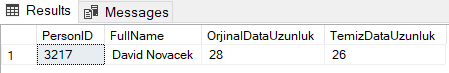
**Adım 1.5: Kişi İsimlerindeki Baş/Son Boşlukları Kontrol Etme ve Temizleme**

**Adım 1.5.1: Başında veya Sonunda Boşluk Olan İsimleri Tespit Etme**

Bu adımda, **Application.People** tablosundaki FullName sütununda başında veya sonunda gereksiz boşluk karakteri (whitespace) içeren kayıtlar olup olmadığı kontrol edilecektir. LEN() fonksiyonu sondaki boşlukları göz ardı edebildiğinden, kontrol için daha güvenilir olan DATALENGTH() fonksiyonu kullanılarak orijinal değerin byte uzunluğu ile LTRIM(RTRIM()) sonrası byte uzunluğu karşılaştırılacaktır.

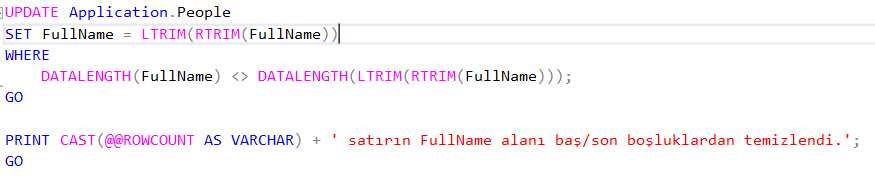


Yukarıdaki sorgu çalıştırıldığında sonuçları aşağıda görüldüğü gibidir. Sorgu, PersonID 3217 için OrjinalDataUzunluk (28) ve TemizDataUzunluk (26) değerlerinin farklı olduğunu göstermiştir. Bu, ilgili kaydın FullName alanında başında veya sonunda (bu durumda muhtemelen sonda) temizlenmesi gereken boşluk karakterleri bulunduğunu kesin olarak teyit etmektedir.



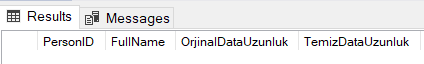
**Adım 1.5.2: Tespit Edilen İsimlerdeki Baş/Son Boşlukları Temizleme**

Bir önceki adımda DATALENGTH karşılaştırması ile tespit edilen, başında veya sonunda boşluk bulunan FullName değerleri temizlenecektir. UPDATE işlemi, DATALENGTH koşulunu sağlayan tüm satırları hedefleyecektir.



Yukarıdaki UPDATE sorgusu çalıştırıldığında, başında veya sonunda boşluk bulunan kayıtların FullName alanları temizlenir ve etkilenen satır sayısı mesaj olarak gösterilir.

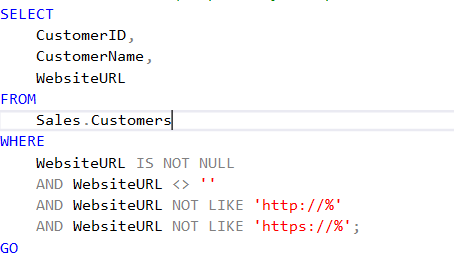
Temizleme işleminin başarılı olduğunu ve artık başında/sonunda boşluk olan kayıt kalmadığını doğrulamak için, Bir önceki adımdaki DATALENGTH kontrol sorgusu tekrar çalıştırılacaktır. Bu sorgunun artık hiçbir kayıt döndürmemesi beklenir.



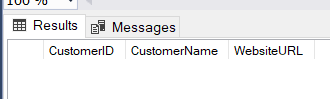
Yukarıdaki sonuçtan da görüldüğü gibi bir önceki adımdaki DATALENGHT kontrol sorgusu çalıştırıldığında bir sonuç dönmediğini görüyoruz. Bu da işlemimizin başarılı olduğu anlamına gelir.

**Adım 1.6 - WebsiteURL Protokol Kontrolü ve Düzeltme**

Bu adımda, **Sales.Customers** tablosundaki WebsiteURL sütununda 'http://' veya 'https://' protokolü ile başlamayan, yani eksik protokole sahip URL'ler kontrol edilecektir

****

Yukarıdaki sorgu çalıştırıldığında sonuçları aşağıda görüldüğü gibidir. Sorgunun hiçbir kayıt döndürmemesi, Sales.Customers tablosundaki tüm geçerli WebsiteURL değerlerinin zaten bir protokol (http:// veya https://) içerdiğini göstermektedir. Bu kontrol sonucunda düzeltilmesi gereken bir URL bulunmamıştır.

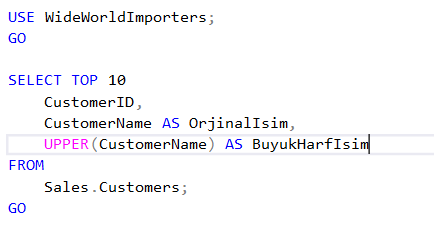


**Bölüm 2: Veri Dönüştürme**

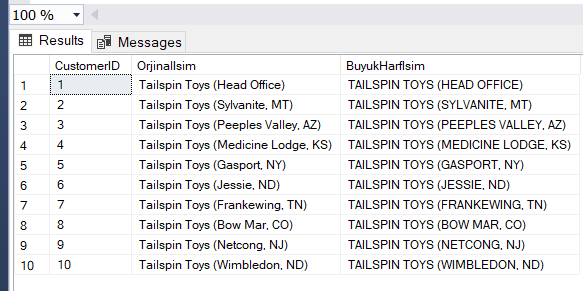
Veri temizleme adımlarının ardından, bu bölümde verilerin analiz ve raporlama için daha kullanışlı ve standart bir formata getirilmesi amacıyla veri dönüştürme işlemleri yapılacaktır. Dönüştürme, mevcut verilerin formatını, yapısını veya değerlerini değiştirmeyi içerir.

**Adım 2.1: Müşteri İsimlerini Büyük Harfe Dönüştürme**

Bu ilk dönüştürme adımında, **Sales.Customers** tablosundaki CustomerName sütunundaki değerlerin tutarlılık amacıyla tümüyle büyük harfe dönüştürülmüş hali gösterilecektir.

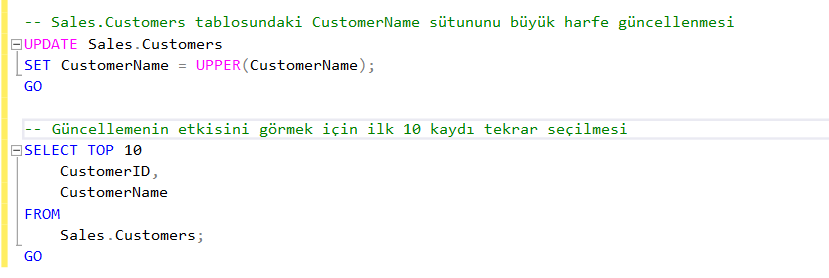


Yukarıdaki sorgu çalıştırıldı ve sonuçları ekran görüntüsünde görüldüğü gibidir. Sorgu, ilk 10 müşteri isminin orijinal halini (OrjinalIsim) ve UPPER() fonksiyonu kullanılarak tamamen büyük harfe çevrilmiş halini (BuyukHarfIsim) yan yana listeler. Bu çıktı, dönüşümün beklendiği gibi çalıştığını görsel olarak doğrulamaktadır ve tüm isimlerin büyük harfe çevrilmesinin veri setine nasıl yansıyacağını göstermektedir.



**Adım 2.1.1: Müşteri İsimlerini Kalıcı Olarak Büyük Harfe Güncelleme**

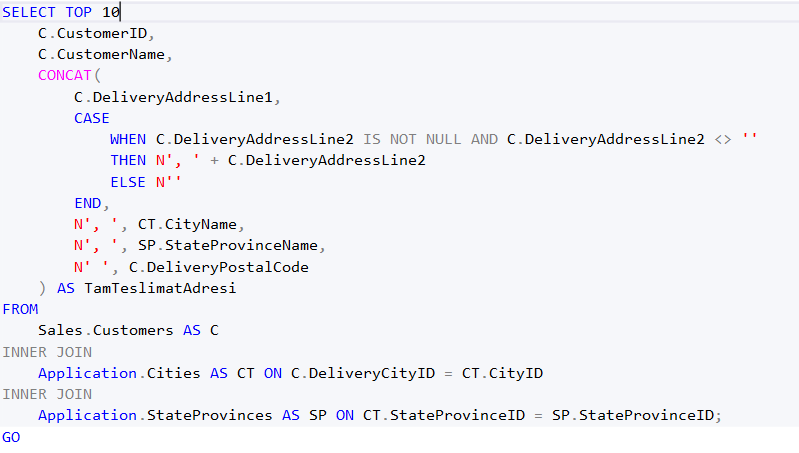
Bu adımda, bir önceki adımda gösterilen dönüşüm kalıcı hale getirilecektir. Sales.Customers tablosundaki tüm CustomerName değerleri UPPER() fonksiyonu kullanılarak büyük harfli halleriyle güncellenecektir.



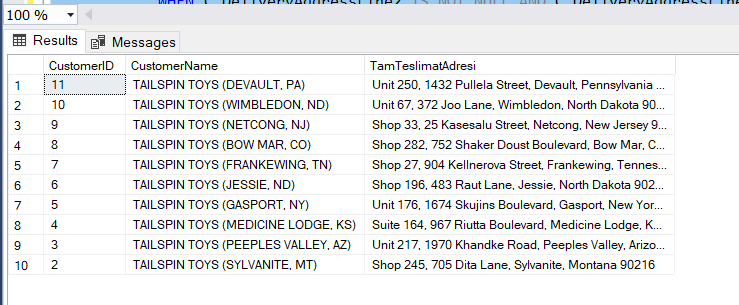
Doğrulama sorgusunun sonucunda, CustomerName sütunundaki değerlerin artık tamamen büyük harf olduğu görülmektedir. Bu, müşteri isimlerini büyük harfe dönüştürme işleminin kalıcı olarak başarıyla uygulandığını teyit eder.

**Adım 2.2: Tam Teslimat Adresi Oluşturma**

Bu adımda, müşterilerin teslimat adres bilgileri (Sales.Customers, Application.Cities, Application.StateProvinces tablolarından alınarak) tek bir TamTeslimatAdresi sütununda birleştirilecektir.

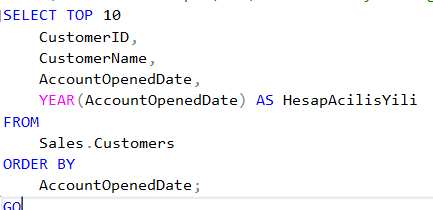


Yukarıdaki sorgu çalıştırıldı ve sonuçları aşağıda görüldüğü gibidir. Sorgu, ilgili tablolardan alınan adres bileşenlerini (DeliveryAddressLine1, DeliveryAddressLine2, CityName, StateProvinceName, DeliveryPostalCode) başarılı bir şekilde birleştirerek ilk 10 müşteri için formatlanmış TamTeslimatAdresi'ni oluşturmuştur.

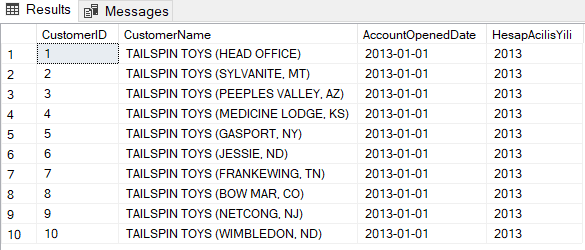


**Adım 2.3: Hesap Açılış Yılını Çıkarma**

Bu adımda, Sales.Customers tablosundaki AccountOpenedDate sütunundan yıl bilgisi YEAR() fonksiyonu kullanılarak ayrı bir sütun olarak alınacaktır.



Yukarıdaki sorgu çalıştırıldı ve sonuçları aşağıda görüldüğü gibidir. Sorgu, ilk 10 müşterinin hesap açılış tarihini (AccountOpenedDate) ve bu tarihten **YEAR()** fonksiyonu ile başarılı bir şekilde çıkarılan yıl bilgisini (HesapAcilisYili) gösterir.



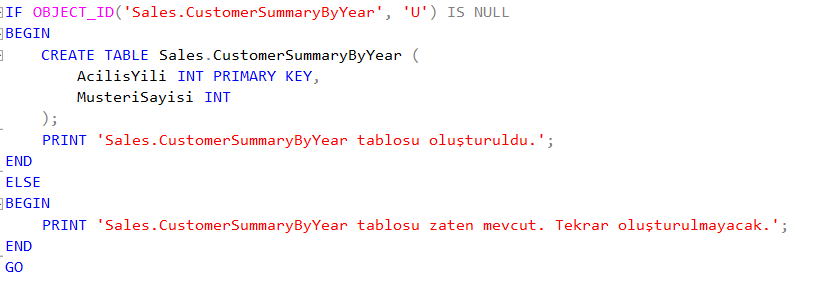
**Bölüm 3: Veri Yükleme**

**Adım 3.1: Yıllık Müşteri Özeti Tablosunu Oluşturma ve Yükleme**

Bu adım 3 alt adımdan oluşmaktadır: Hedef tablonun oluşturulması, hesaplanan verinin yüklenmesi ve yüklenen verinin doğrulanması.

**Adım 3.1.1: Hedef Tabloyu Oluşturma**

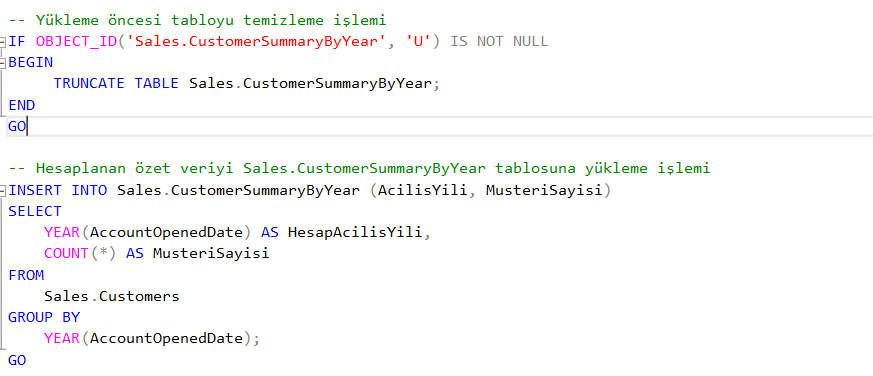
Veri yükleme işleminin ilk adımı olarak, yıllık müşteri sayısı özetini saklayacak olan Sales.CustomerSummaryByYear adında yeni bir tablo oluşturulacaktır. Tablonun zaten var olup olmadığını kontrol eden bir mekanizma eklenecektir.

****

Yukarıdaki sorgu çalıştırıldığında tablo ya oluşturulur ya da zaten var olduğu mesajı alınır.

**Adım 3.1.2: Hesaplanan Özeti Hedef Tabloya Yükleme**

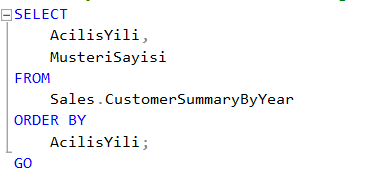
Şimdi, Sales.Customers tablosundan hesaplanan yıllık müşteri sayısı özeti, bir önceki adımda oluşturulan veya var olan Sales.CustomerSummaryByYear tablosuna INSERT INTO ... SELECT ifadesi kullanılarak yüklenecektir. Tekrarlı yüklemelerde sorun olmaması için yükleme öncesinde tablo içeriği TRUNCATE TABLE komutu ile temizlenmektedir.



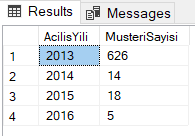
Yukarıdaki sorgu çalıştırıldığında, hesaplanan özet veriler hedef tabloya yüklenir ve kaç satırın yüklendiği mesaj olarak gösterilir.

**Adım 3.1.3: Yüklenen Veriyi Doğrulama**

Verilerin Sales.CustomerSummaryByYear tablosuna doğru şekilde yüklendiğini kontrol etmek için yeni oluşturulan ve yüklenen bu tablonun içeriği sorgulanacaktır.



Yukarıdaki sorgu çalıştırıldığında sonuçları aşağıda görüldüğü gibidir. Bu sonuçlar, yıllık müşteri sayısı özetinin başarıyla oluşturulup kalıcı bir tabloya yüklendiğini teyit eder.

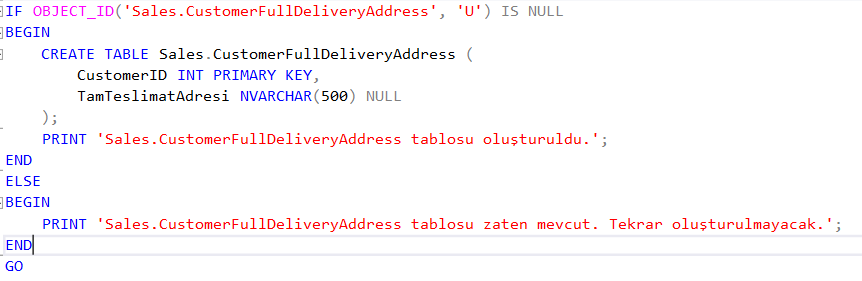


**Adım 3.2: Tam Teslimat Adreslerini Yeni Tabloya Yükleme**

Bu adım da 3 alt adımdan oluşmaktadır: Hedef tablonun oluşturulması, birleştirilmiş adres verisinin yüklenmesi ve yüklenen verinin doğrulanması.

**Adım 3.2.1: Hedef Tabloyu Oluşturma**

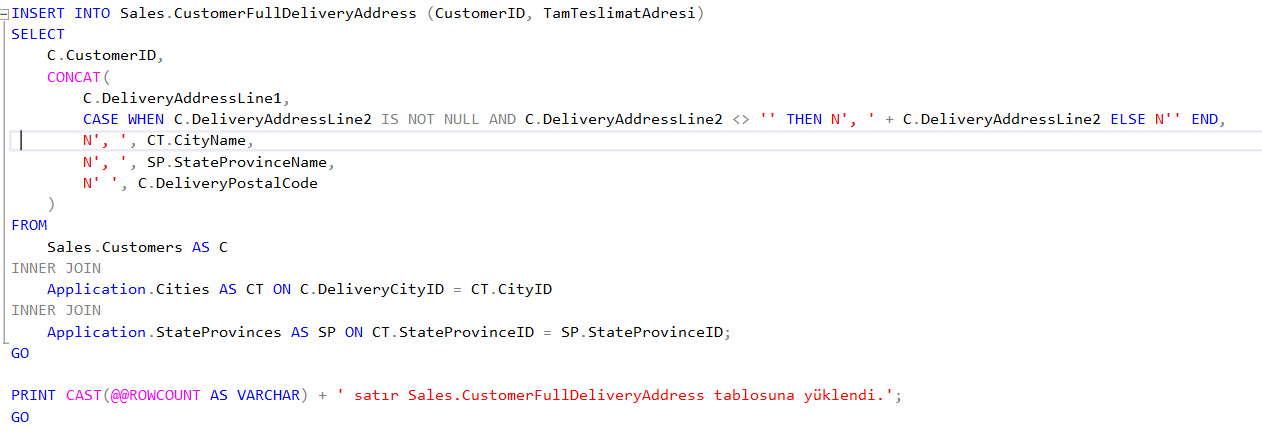
Bu adımda, Bölüm 2'de oluşturulan tam teslimat adreslerini kalıcı olarak saklamak üzere Sales.CustomerFullDeliveryAddress adında yeni bir tablo oluşturulacaktır.



Yukarıdaki sorgu çalıştırıldığında tablo ya oluşturulur ya da zaten var olduğu mesajı alınır.

**Adım 3.2.2: Birleştirilmiş Adresleri Hedef Tabloya Yükleme**

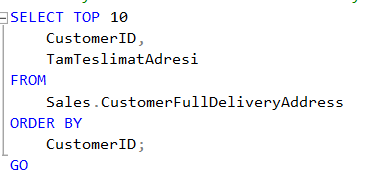
Oluşturulan (veya zaten var olan) Sales.CustomerFullDeliveryAddress tablosuna, ilgili tablolardan birleştirilerek elde edilen tam teslimat adresleri yüklenecektir. Yükleme öncesinde tablo içeriği TRUNCATE TABLE komutu ile temizlenmektedir.



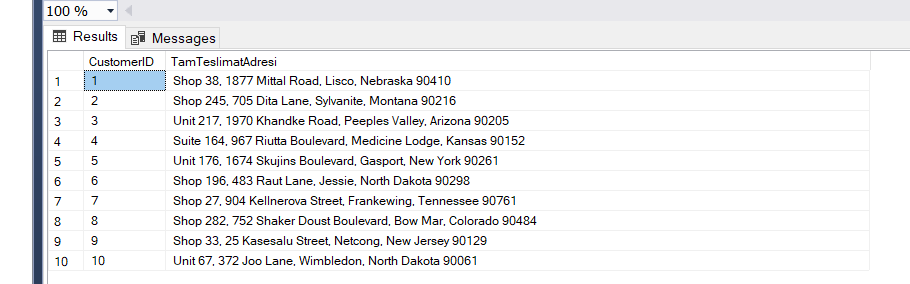
Yukarıdaki sorgu çalıştırıldığında, birleştirilmiş adres verileri hedef tabloya yüklenir ve kaç satırın yüklendiği mesaj olarak gösterilir.

**Adım 3.2.3: Yüklenen Veriyi Doğrulama**

Tam teslimat adreslerinin Sales.CustomerFullDeliveryAddress tablosuna doğru bir şekilde yüklendiğini kontrol etmek için tablodan ilk 10 kayıt sorgulanacaktır.



Yukarıdaki sorgu çalıştırıldığında sonuçları aşağıda görüldüğü gibidir. Bu, Bölüm 2'de dönüştürülerek elde edilen birleştirilmiş adres verisinin yeni tabloya başarıyla yüklendiğini gösterir.



**Bölüm 4: Veri Kalitesi Raporları**

Bu bölümde, projenin ilk aşamasında gerçekleştirilen veri kalitesi kontrollerinin sonuçları özetlenmektedir. Veri kalitesi raporlaması, veritabanındaki olası sorunları belgelemek, veri setinin güvenilirliği hakkında bilgi vermek ve gelecekteki veri yönetimi stratejileri için temel oluşturmak açısından önemlidir.

**Adım 4.1: Veri Temizleme Kontrolleri Bulgularının Özeti**

Gerçekleştirilen proje kapsamında Wide World Importers veritabanında aşağıdaki temel veri kalitesi kontrolleri gerçekleştirilmiş ve şu sonuçlar elde edilmiştir:

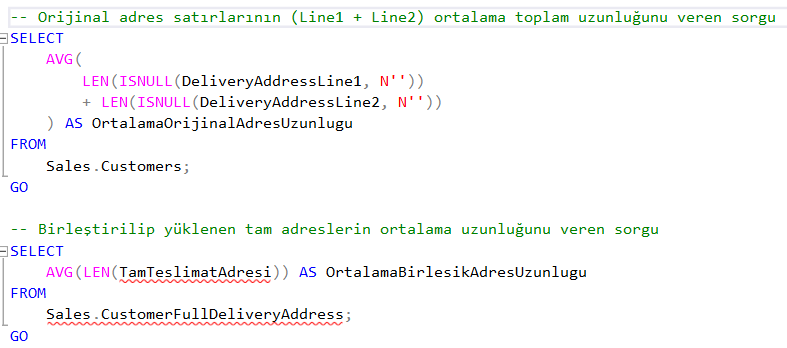
1. Eksik Birincil Kontak E-posta Adresi Kontrolü (Adım 1.1):
   * Amaç: Sales.Customers tablosundaki müşterilerin Application.People tablosunda kayıtlı birincil iletişim kişilerine ait e-posta adreslerinin eksik (NULL veya boş) olup olmadığını kontrol etmek.
   * Bulgu: Yapılan sorgulama sonucunda, birincil iletişim kişisine ait e-posta adresi eksik olan herhangi bir müşteri kaydına rastlanmamıştır.
2. Geçersiz E-posta Formatı Kontrolü (Adım 1.2):
   * Amaç: Application.People tablosundaki e-posta adreslerinin temel formatını (içinde '@' karakteri bulunup bulunmadığını) kontrol etmek.
   * Bulgu: Yapılan sorgulama sonucunda, '@' karakteri içermeyen ve potansiyel olarak geçersiz formattaki bir e-posta adresine rastlanmamıştır.
3. Eksik Telefon Numarası Kontrolü (Adım 1.3):
   * Amaç: Sales.Customers tablosundaki PhoneNumber sütununda eksik (NULL veya boş) değer olup olmadığını kontrol etmek.
   * Bulgu: Yapılan sorgulama sonucunda, telefon numarası bilgisi eksik olan herhangi bir müşteri kaydına rastlanmamıştır.
4. Potansiyel Tekrarlı Müşteri Kayıtları Kontrolü (Adım 1.4):
   * Amaç: Sales.Customers tablosunda aynı CustomerName değerine sahip birden fazla kayıt olup olmadığını kontrol etmek.
   * Bulgu: Yapılan sorgulama sonucunda, aynı müşteri ismine sahip birden fazla kayda rastlanmamıştır.

**Genel Değerlendirme:**

Gerçekleştirilen bu temel veri kalitesi kontrolleri sonucunda, Wide World Importers örnek veritabanının müşteri iletişim bilgileri (e-posta, telefon) ve müşteri isimlerinin tekilliği açısından herhangi bir sorun içermediği görülmüştür. İncelenen alanlarda eksik veya temel formatı bozuk verilere ve tekrarlı kayıtlara rastlanmamıştır. Bu durum, kullanılan örnek veri setinin bu spesifik kontroller açısından temiz ve tutarlı olduğunu göstermektedir. Daha kapsamlı veri kalitesi analizleri için farklı kontroller (adres tutarlılığı, geçersiz karakterler, aykırı değer analizi vb.) de yapılabilir, ancak projenin bu aşamasında yapılan temel kontroller olumlu sonuçlanmıştır.

**Adım 4.2: SQL ile Süreç Etkisi Raporlama**

SQL sorguları kullanılarak süreç hakkında raporlama yapmak için, adres birleştirme ve yükleme adımının (Adım 3.2) bir etkisi incelenecektir. Aşağıdaki ilk sorgu, **Sales.Customers** tablosundaki orijinal teslimat adresi satırlarının (DeliveryAddressLine1 ve DeliveryAddressLine2) toplam ortalama karakter uzunluğunu hesaplar. İkinci sorgu ise Adım 3.2'de oluşturulan ve yüklenen **Sales.CustomerFullDeliveryAddress** tablosundaki birleştirilmiş tam adreslerin (TamTeslimatAdresi) ortalama karakter uzunluğunu hesaplar.



Yukarıdaki sorgular çalıştırıldığında sonuçları aşağıda görüldüğü gibidir. İlk sonuç, orijinal adres satırlarının (DeliveryAddressLine1 + DeliveryAddressLine2) ortalama toplam uzunluğunun **25 karakter** olduğunu göstermektedir. İkinci sonuç ise, birleştirilip **Sales.CustomerFullDeliveryAddress** tablosuna yüklenen tam adreslerin ortalama uzunluğunun **55 karakter** olduğunu göstermektedir. Bu metrikler, adres birleştirme ve yükleme işleminin sonucunda adres bilgisinin ortalama olarak daha uzun (daha fazla detay içerir şekilde) hale geldiğini sayısal olarak ortaya koymaktadır.

