### ŞEKİLLER DİZİNİ

[Figure 1 Excel Veri Tabanından Görüntü 3](#_Toc124950402)

[Figure 2 Data Modeller İlişkisel Sınıf Tasarımı 3](#_Toc124950403)

[Figure 3 Data Modeller Fiziksel Sınıf Tasarımı 4](#_Toc124950404)

[Figure 4 Data Modeller DDL Kodundan Bir Kesit 4](https://sduedutr-my.sharepoint.com/personal/l1811012068_ogr_sdu_edu_tr/Documents/aras_ateser_1811012068_tasarım2_Finalll%20(1)%20(1).docx#_Toc124950405)

[Figure 5 Sınıf ve Özellikleri 6](#_Toc124950406)

[Figure 6 APEX Oracle ile SQL Komutarı 10](#_Toc124950407)

[Figure 7 İlk Dönem Değişimleri Sonrası Sınıf Tablosu 10](#_Toc124950408)

[Figure 8 Şubelerin Kod Şeması 11](https://sduedutr-my.sharepoint.com/personal/l1811012068_ogr_sdu_edu_tr/Documents/aras_ateser_1811012068_tasarım2_Finalll%20(1)%20(1).docx#_Toc124950409)

[Figure 9 İkinci Dönem Değişimleri Sonrası Sınıf Tablosu 12](#_Toc124950410)

[Figure 10 APEX Oracle "Home" Sayfası 12](https://sduedutr-my.sharepoint.com/personal/l1811012068_ogr_sdu_edu_tr/Documents/aras_ateser_1811012068_tasarım2_Finalll%20(1)%20(1).docx#_Toc124950411)

[Figure 11 APEX Oracle "Online Services" Sayfası 13](https://sduedutr-my.sharepoint.com/personal/l1811012068_ogr_sdu_edu_tr/Documents/aras_ateser_1811012068_tasarım2_Finalll%20(1)%20(1).docx#_Toc124950412)

[Figure 12 APEX Oracle "Contact" Sayfası 13](#_Toc124950413)

[Figure 13 APEX Oracle "Home Delivery" Sayfası 14](#_Toc124950414)

[Figure 14 APEX Oracle Sayfa Düzenleme Ekranı 15](#_Toc124950415)

[Figure 15 İkinci Dönem "Home" Sayfası 15](#_Toc124950416)

[Figure 16 "Calculate Price" Sayfası 16](#_Toc124950417)

[Figure 17 "Cargo Update Information" Sayfası 17](#_Toc124950418)

[Figure 18"Cargo Management" Sayfası 17](#_Toc124950419)

[Figure 19 "Cargo" Sınıfı 20](#_Toc124950420)

[Figure 20 "Courier" Sınıfı 20](#_Toc124950421)

[Figure 21"Courier" Sınıfı Bilgisi ile Gösterge Paneli 20](https://sduedutr-my.sharepoint.com/personal/l1811012068_ogr_sdu_edu_tr/Documents/aras_ateser_1811012068_tasarım2_Finalll%20(1)%20(1).docx#_Toc124950422)

[Figure 22 "Cargo" Sınıfı Bilgisi ile Gösterge Paneli 21](https://sduedutr-my.sharepoint.com/personal/l1811012068_ogr_sdu_edu_tr/Documents/aras_ateser_1811012068_tasarım2_Finalll%20(1)%20(1).docx#_Toc124950423)

### ÇİZELGELER DİZİNİ

[Table 1 Programların Üreticileri ve Sürümleri 2](#_Toc124950032)

[Table 2 Primary Key Göstergesi 5](#_Toc124950033)

### SİMGELER VE KISALTMALAR DİZİNİ

DDL Data Definatin Language (Veri Tanımlama Dili)

SQL Structured Query Language (Yapılandırılmış Sorgu Dili)

# MATERYAL VE YÖNTEM

## Kullanılan Uygulama ve Ortamlar

SQL Developer

Oracle veri tabanlarında SQL ile çalışmak için bütünleşmiş bir geliştirme ortamıdır.

Veri tabanını test etmek, yazılacak kodlar uygun ortamın sağlanabileceğini görmek ve veri tabanını görüntülemek için kullanılacak ortam.

APEX  
 Oracle APEX, Oracle veri tabanlarında web uygulamaları geliştirmek için kullanılan Oracle Corporation’un kurumsal, düşük kodlu geliştirme platformudur.

Tam fonksiyonel bir uygulama ya da bir internet sitesi tasarlamak için vakit harcamayarak, analiz, istatistik, analiz ve veri tabanı kısımlarına daha iyi yoğunlaşmak için projede yer verildi. APEX platformuna, veri tabanı, sınıflar ve ilişki bilgileri eklendikten sonra bu bilgileri kullanmak için web arayüzü kullanılacak.

APEX Sitesi (Kullanıcı Arayüzü)

Göndericisinden kargocuya kadar tüm muhtemel kullanıcıların göreceği ve iletişime geçeceği kısım.

Kullanıcılar kimliklerine göre sadece görmesi gereken kısımlara erişebilecekler. Kargo alıcısı, kargoyu kimin gönderdiği, kargonun nerede olduğu ve ne zaman ulaşabileceği gibi bilgilere ulaşabilirken kargocu, sorumlusu olduğu, teslim etmesi gereken kargoları ve alıcının iletişim bilgileri gibi kısımlara erişimi olacak.

Data Modeller

Üretkenliği artıran ve veri modelleme görevlerini basitleştiren bir grafik aracıdır.

Tasarlanan sistemde sınıfların, özelliklerinin ve aralarındaki bağların ilişkisel seviyelerinin tasarlandığı araç.

Data Modeller sayesinde ilişkisel seviyeden fiziksel seviyeye kolayca geçiş yapıp muhtemel hataların ayıklanması daha da kolaylaşıyor. Uygulama içerisinde üretilecek DDL dosyası ile APEX platformuna geçiş daha kolay ve temelli olmuş olacak. Bundan sonrasında APEX sisteminden devam edildi.

diagrams.net

Diyagram tasarlama uygulaması.

Projenin tasarlama ve raporlama kısmında görselliği arttırmak ve adımları göz önüne dökmek adına kullanılan diyagram yazılımıdır.

Excel

Microsoft tarafından yazılan bir tablolama programıdır.

Projede kullanılan Oracle veri tabanına tam geçiş yapılana kadar yaşanabilecek problemleri ve zorluğu azaltmak adına verilerin tutulacağı sistem olacak.

|  |  |  |
| --- | --- | --- |
| Üretici | Program / Ortam İsmi | Versiyon |
| Oracle | SQL Developer | 21.4.3 |
| Oracle | APEX | 22.1 |
| Oracle | Oracle Database | Express Edition 21c |
| Oracle | Data Modeller | 21.4.2 |
| JGraph | Diagrams.net | 19.0.3 |
| Microsoft | Excel | 2205 |

Tablo Kullanılan Programların Üreticileri ve Sürümleri

## Uygulama ve Ortamların İlişkileri

### Hazırlık

İlk aşamada “diagrams.net” üzerinden muhtemel sınıfların ve bağlantılarının ilişkileri Sınıf ilişkileri hazırlandıktan sonra “Oracle Database” sistemine tam geçiş yapana kadar mevcut problemlerin gözlenmesi ve projenin devam edebilmesi için geçici olarak “Excel”den geçici veri tabanları oluşturuldu (Bu sayede veri tabanı “APEX” platformuna direkt aktarılabilecek)

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure Excel Veri Tabanından Görüntü

### Başlangıç

Sınıf ilişkileri kurulduktan sonra bağlantıların yapılabilmesi ve muhtemel hataların azaltılması adına “Data Modeller” uygulamasına geçildi. “Data Modeller”da sınıflar, özellikleri ve bağlantıların ilişkisel seviyeleri, “primary key”leri, veri tipleri ve kapasiteleri hazırlandı.

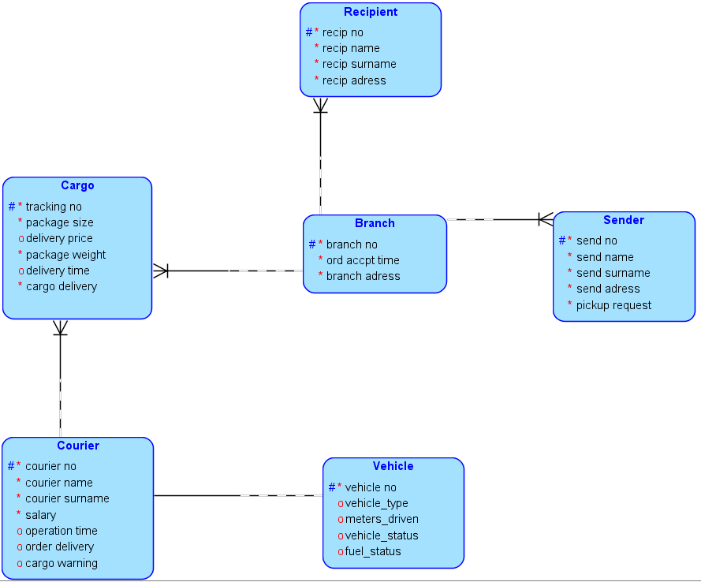


Figure Data Modeller İlişkisel Sınıf Tasarımı

İlişkisel seviyedeki hatalar ayıklandıktan sonra aynı yöntem ile ilişkisel seviye kullanılıp fiziksel seviye oluşturulup hatalar tekrar ayıklandı.

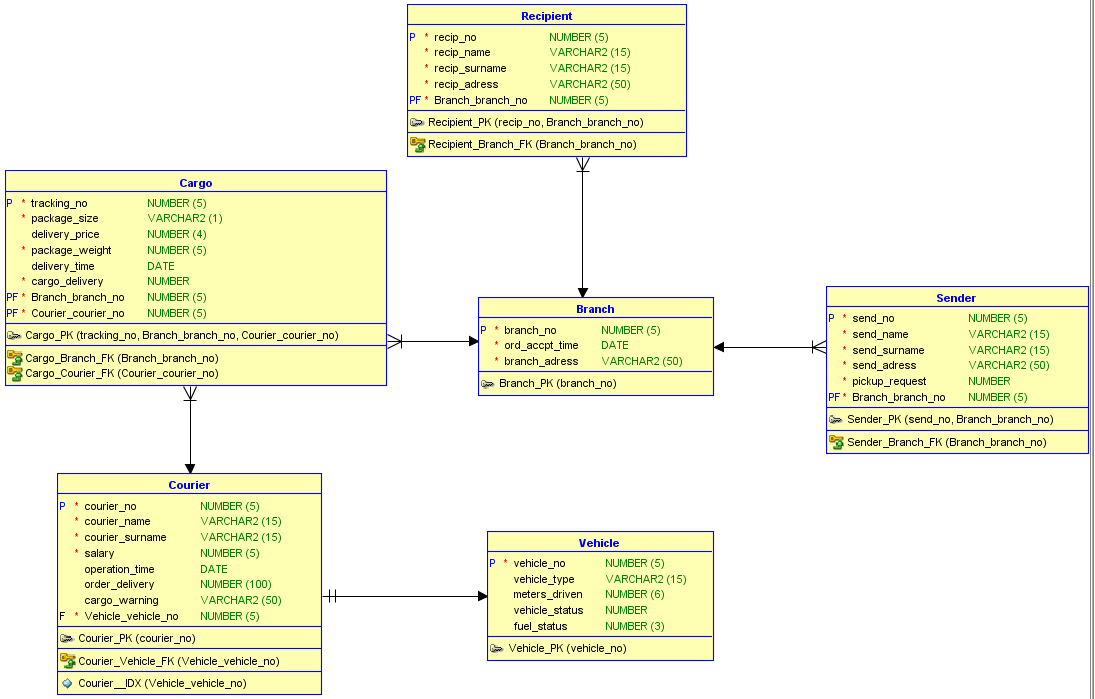


Figure Data Modeller Fiziksel Sınıf Tasarımı

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu Fiziksel seviyede son aşamaya gelindiğinde DDL dosyası oluşturuldu ve APEX ortamında geçişe hazır aşamaya gelindi.

Figure Data Modeller DDL Kodundan Bir Kesit

# BULGULAR VE TARTIŞMA

## Sınıf Haritasının Gelişimi

Sistemde bulunan sınıf ve özelliklerinin veri boyutları aksi belirtilmedikçe VARCHAR için 15 karakter; adresler için VARCHAR için 50 karakter, NUMBER için 5 rakamdı alarak ayarlandı. Mevcut boyutlar, kendilerine verilen alan en kötü senaryoda bile yeteceği düşünüldüğü için atanmıştır. Ayrıca farklı boyut verilen özelliklerin ise atanmış boyutları dışında değer almaları mümkün olmadığı ya da normal değeri rahatça aşabilecekleri için değiştirilmiştir (adres değerleri arttırılmış, benzin kapasitesi değerleri azaltılmıştır).

Primary keyler ayarlanırken “Sender” sınıfındaki “send\_no” 1001, “Recipient” sınıfındaki “recip\_no” 2001, “Cargo” sınıfındaki “tracking\_no” 3001, “Courier” sınıfındaki “courier\_no” 4001, “Vehicle” sınıfındaki “vehicle\_no” 5001, “Branch” sınıfındaki “branch\_no” 6001 sayıları ile başlamaktadır. Ve alabilecekleri en büyük değer “\_999” olarak ayarlanmıştır. Proje küçük bir yönetim sistemi olacağı için her sınıfın en azla 999 üyesi olabileceği öngörülerek tasarlanmış, sadece ilk rakamlar farklı tutularak ise ileride yaşanabilecek problemlerde projenin okunabilirliği arttırılması çalışılmıştır.

|  |  |
| --- | --- |
|  | Primary Key Başlangıç No |
| send\_no | 1001 |
| recip\_no | 2001 |
| tracking\_no | 3001 |
| courier\_no | 4001 |
| vehicle\_no | 5001 |
| branch\_no | 6001 |

Tablo Primary Key Göstergesi

## Sınıflar Arasındaki Bağlar

Şu ana kadar 6 sınıf arasında 5 farklı bağ kurulmuş her sınıf en fazla 3 en az 1 bağ kurulmuştur:

Sender – Branch Bağı: Bire çok ilişkisi kurulmuştur. Aktör ile merkez arasında bağ kurularak teslimatın yapılabilmesi için optimum bilgilerin alınması sağlanır.

Recipient – Branch Bağı: Bire çok ilişkisi kurulmuştur. Aktör ile merkez arasında bağ kurularak teslimatın yapılabilmesi için optimum bilgilerin alınması sağlanır.

Branch – Cargo Bağı: Bire çok ilişkisi kurulmuştur. Alıcı ve göndericiden gerekli bilgiler alındıktan matematiğin yapıldığı aşamadır.

Cargo – Courier Bağı: Bire çok ilişkisi kurulmuştur. Dağıtılacak kargo ya da kargolar bir kuryeye atanarak kargo karışıklığının önüne geçilir, verimliliğin arttırılması amaçlanır.

Courier – Vehicle Bağı: Bire çok ilişkisi kurulmuştur. Her kuryeye uygun bir araç atanır. Sürülebilir araçların kargoculara bildirilmesi ve her kargocuya sadece 1 araç atamayarak verimliliğin arttırılması amaçlanır.

## Sınıf Haritasının Gelişim Öncesi Hali:

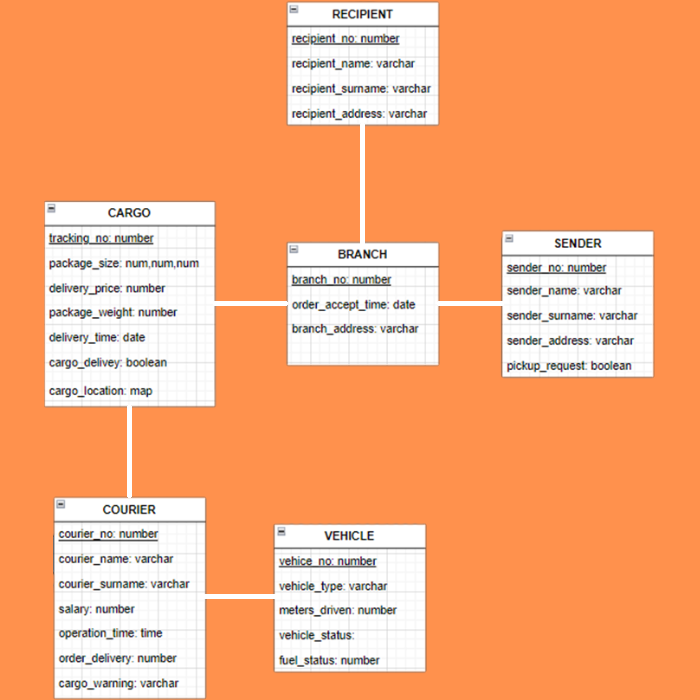


Figure Sınıf ve Özellikleri

Recipient sınıfı, kargonun alıcısı olup teslim edilecek noktanın belirlendiği aktördür ve temel gerekli bilgileri tutar.

recipient\_no: NUMBER formatında, sistem tarafından, her kullanıcısına “2” ile başlayacak özgün numaraları tutacak.

recipient\_name: VARCHAR formatında alıcının adını tutacak.

recipient\_surname: VARCHAR formatında alıcının soyadını tutacak.

recipient\_adress: VARCHAR formatında kargonun teslim edileceği konumun bilgilerini tutar.

Sender sınıfı, kargonun göndericisi olup nereden teslim alınacağına karar veren aktördür ve temel gerekli bilgileri tutar.

sender\_no: NUMBER formatında, sistem tarafından, her kullanıcısına “1” ile başlayacak özgün numaraları tutacak.

sender\_name: VARCHAR formatında göndericinin adını tutacak.

sender\_surname: VARCHAR formatında göndericinin soyadını tutacak.

sender\_address: VARCHAR formatında kargonun teslim alınacağı konumun bilgilerini tutar.

pickup\_request: BOOLEAN formatında, göndericinin kargosunun şube yerine istediği noktadan teslim alınıp alınmak istemediğinin bilgisini tutar. Gönderici kargosunun şube dışında bir noktadan alınmasını istiyorsa TRUE olarak işaretlenip adres bilgilerini girmek zorunda olacak.

Branch sınıfı, fiziksel şubeyi ve ana bağlantıların kargocuya aktarılması görevini üstlenir.

branch\_no: NUMBER formatında, sistem tarafından, her şubeye “6” ile başlayacak özgün numaraları tutacak.

order\_accept\_time: DATE formatında, göndericinin kargosunu ne zaman teslim ettiği bilgisini tutacak.

branch\_address: VARCHAR formatında, ilgili kargonun hangi şubeden dağıtıma çıkacağı ya da teslim alınacağı bilgisi tutulur.

Cargo sınıfı, göndericinin teslim ettiği kargonun fiyatlandırılması, paket bilgileri gibi bilgileri tutar.

tracking\_no: NUMBER formatında, sistem tarafından, her kargoya “3” ile başlayacak özgün numaraları tutacak.

package\_size: VARCHAR formatında, kargo kutusunun ölçülerini tutar(?).

delivery\_price: VARCHAR formatında, kargonun ağrılığı ve ölçülerini hesaba katarak kargonun ücretini ödeyecek kişinin maliyetini tutar.

package\_weight: VARCHAR formatında, kargonun ağırlığını tutar.

delivery\_time: DATE formatında, kargonun şubeye tesliminden alıcıya teslimine kadar geçen süreyi tutar.

cargo\_delivery: BOOLEAN formatında, kargonun alıcıya teslim edilip edilmediğini tutar. Kargo teslim edildiğinde TRUE, edilmediğinde FALSE değerlerini tutarken teslim aşamasındaysa NULL olarak kalır.

Courier sınıfı, kargoların taşıma işini üstlenen kargocuların bilgilerini tutar ve kargolarla olan bağlarını sağlar.

courier\_no: NUMBER formatında, sistem tarafından, her aktöre “4” ile başlayacak özgün numaraları tutacak.

courier\_name: VARCHAR formatında kargocunun adını tutacak.

courier\_surname: VARCHAR formatında kargocunun soyadını tutacak.

salary: NUMBER formatında, her kargocunun mevcut maaşını tutacak.

operation\_time: TIME formatında, kargocunun iş başında ne kadar zaman harcadığını gösterir. Belirli sınırları geçtikten sonra maaş artışını sağlar.

order\_delivery: NUMBER formatında, kuryenin dağıtıma çıkaracağı kargoları gösterir.

cargo\_warning: VARCHAR formatında, kargo teslim edilemediyse nedeninin kargocu tarafından girildiği metni tutar. cargo\_delivery özelliği FALSE işaretlendiğinde zorunlu hale gelir.

Vehicle sınıfı, kargocunun kullandığı araca dair bilgiler tutulur.

vehicle\_no: NUMBER formatında, sistem tarafından, her araca “5” ile başlayacak özgün numaraları tutacak.

vehicle\_type: VARCHAR formatında, şirketin dağıtım için sahip olduğu araçlardan kargocunun kullanabileceği aracın ismini tutar.

meters\_driven: NUMBER formatında 6 rakam tutar, kargocunun mevcut araç ile kaç kilometre yol sürdüğünü gösterir.

vehicle\_status: BOOLEAN formatında, mevcut aracın boşta olup olmadığını gösterir. TRUE ise araç kullanımda, FALSE ise bir kargocu tarafından kullanılabilir olduğunu gösterir.

fuel\_status: NUMBER formatında 3 rakam tutar, aracın mevcut benzin durumunu gösterir.

## Sınıf Haritasının Dönem İçerisindeki Değişimleri:

### Birinci Dönem Değişiklikleri

APEX Oracle sistemine geçilme sürecinde ve projenin normalleştirilme zamanında fark edilen önemli kısımlardan biri de tahmini kargo ücretinin nasıl hesaplanacağı ve veri tabanında tutulacağı. İlk tasarlanan biçiminde olduğu gibi 3 boyutlu (en, boy, yükseklik) değerler vermek yerine her bir değerin ayrı özellikler olması için değiştirildi.

cargo\_type: BOOLEAN formatında, kargonun koli (package) ya da evrak (document) olarak taşınacağı bilgisi.

cargo\_width: NUMBER formatında, kargonun genişlik bilgisi.

cargo\_height: NUMBER formatında, kargonun yükseklik bilgisi.

cargo\_length: NUMBER formatında, kargonun boy bilgisi.

İnternet sitesi aracılığı ile kullanıcıdan mesajlar alabilmek üzere “Ticket” sınıfı oluşturuldu. Göndericinin, adı, soyadı, e-posta adresi ve göndermek istediği mesajı bu sınıf altında depolanabilecek.

ticket\_no: NUMBER formatında, veri tabanına, site aracılığı ile kullanıcı tarafından gönderilecek iletileri özgünleştirecek numara dizini.

ticket\_name: VARCHAR formatında, mesajı gönderen kullanıcının ismi.

ticket\_surname: VARCHAR formatında, mesajı gönderen kullanıcının soyismi.

ticket\_email: VARCHAR formatında, mesajı gönderen kullanıcının e posta adresi.

ticket\_message: VARCHAR formatında, kullanıcının iletisini tutan özellik.

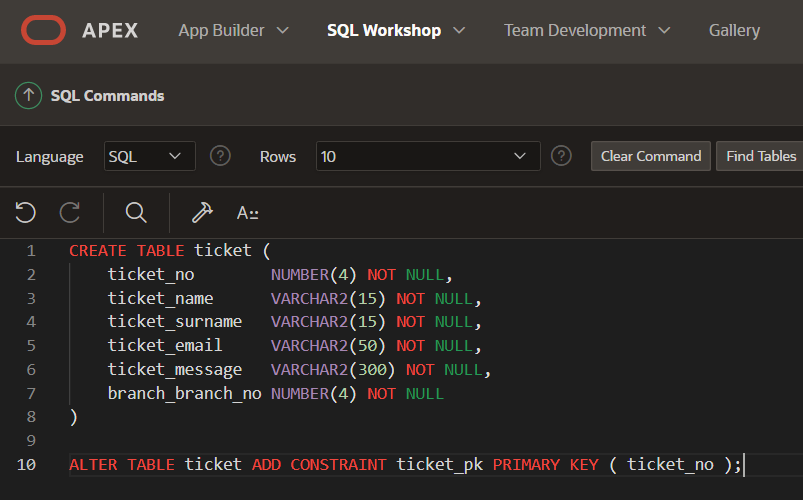


Figure APEX Oracle ile SQL Komutarı

“Ticket” sınıfının eklenmesinde diğer sınıflar için kullanılan DDL dosyası yerine APEX Oracle’ın SQL desteği kullanarak yazıldı.

Kuryeler ve araçlar arasında teke tek bağa çevrildi. Bu şekilde sistemde araçların takibi ve yönetimi kolaylaştırıldı.

Diagram

Description automatically generated

Figure İlk Dönem Değişimleri Sonrası Sınıf Tablosu

### İkinci Dönem Değişiklikleri

APEX Oracle üzerinden web sitesinin tamamlanması sırasında sınıflarda ve özelliklerde projenin mantığından uzaklaşılmaması ve verimliliğin arttırılabilmesi adına ikinci dönem değişiklikleri yapıldı.

“Sender”, “Recipient” ve “Cargo” sınıflarının hepsinde Figür 7’de görüleceği üzere adres ya da konum bilgileri yer almakta. Ama ilgili sınıfların hepsi branch adreslerini de tuttuğu için ek bir adres özelliği barındırmak bilgi kirliliğini arttırıp verimliliği azaltacaktır. Bu fikirden yola çıkarak sırasıyla “send\_adress”, “recip\_adress” ve “cargo\_location” özellikleri sınıflarından kaldırıldı.

APEX sisteminde “Home Delivery” sayfasının kodlanması sırasında “pickup\_request” özelliğinin işaretlenmesinin zorunlu olması mantık hatasına yol açtığı fark edildiği için sistemden kaldırıldı. “Home Delivery” formunu dolduran bir gönderici halihazırda evden teslim etmek istediği anlamına gelmektedir ve bu yüzden ayrıca onaylanmasına gerek yoktur.

Projenin ilerleyen safhalarında “Vehicle” sınıfının sistemin tasarlandığı yönde analiz yapılamayacağı ve veri kalabalığı oluşturacağı gerekçesiyle verit banından tamamen çıkarıldı. “Vehicle sınıfının kaldırılmasından sonra ilişkili bir şekilde “Courier” sınıfında kullanılmak üzere oluşturulmuş olan “operation\_time” özelliği de işlevini göremeyeceği için verimliliği arttırmak üzere veri tabanından kaldırıldı.

“Courier” sınıfının bir özelliği olan “cargo\_warning” kargo hakkında alakalı bilgi verdiğinden dolayı “Courier” sınıfından “Cargo” sınıfına aktarıldı. Böylece kargo hakkında bir durum kayıt altına alınması gerektiği durumda kargolar hakkında diğer bilgilerin bulunduğu gibi “Cargo” sınıfından değiştirilip ulaştırılabilecek şekilde çalışmasına karar verildi.

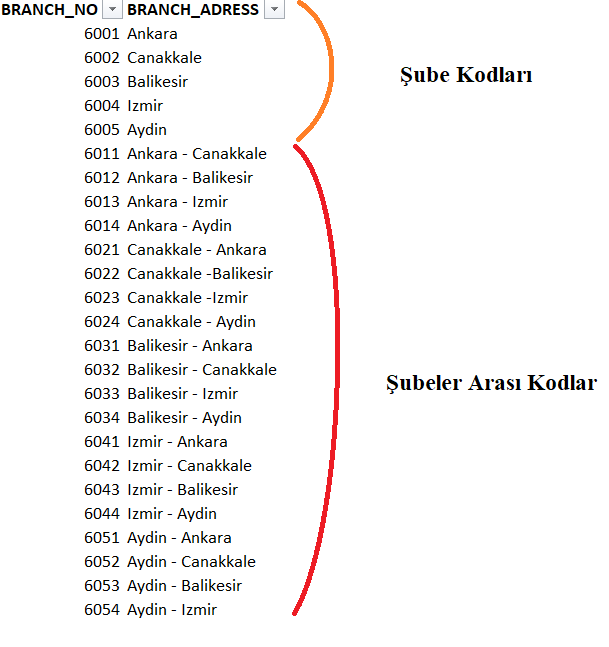


Figure Şubelerin Kod Şeması

Kargoların hangi şehirlerin arasında gönderileceğinin sistemde işaretlenebilmesi için “Branch” sınıfında “branch\_address” özelliği altında şubeler arası bağlantılar da eklendi.

Diagram

Description automatically generated

Figure İkinci Dönem Değişimleri Sonrası Sınıf Tablosu

## APEX Oracle Sistemi

APEX Oracle sistemine “Data Modeller” uygulaması oluşturulan DDL dosyası (Şekil 4) ile geçiş aşaması kolaylaştırıldı. Veri tabanının tasarıma geçilebilmesi için de planlanan sitenin sayfa tasarımları:

Graphical user interface, application

Description automatically generated “Home” sayfası kullanıcıların ulaşmak istedikleri bilgilere ve en çok kullanılması talep edilebilecek olan “kargom nerede” uygulamasını rahatça kullanabilmeleri üzerine tasarlandı. Sayfa üzerindeki butonlar sayfaların arasında rahatça geçiş yapabilmek için rahatça okunabilir ve seçilebilir tasarlandı.

Figure APEX Oracle "Home" Sayfası

Kullanıcı tarafından “Tracking Number”in aratılması ile veri tabanında ilgili kargonun gönderici, alıcı bilgileri ve konumu listelenecek.

Graphical user interface, application

Description automatically generated “Online Services“ sayfası an itibari ile kargounun desisi, gideceği mesafe ve tür bilgilerini göz önünde bulundurarak tahmini kargo ücretini hesaplamak ile kargonun göndericinin evinden teslim alınıp dağıtıma çıkması taleplerin için yöneltici bir sayfa olarak yer alıyor.

Figure APEX Oracle "Online Services" Sayfası

Graphical user interface, application

Description automatically generated

Figure APEX Oracle "Contact" Sayfası

“Contact” sayfası kullanıcının gerekli bilgilerini girerek sisteme göndermek istediği mesajı iletecek bir sayfa olarak tasarlandı. “Create” butonuna tıklandıktan sonra sisteme gönderilen tüm bilgiler veri tabanında “Ticket” sınıfı altında depolanacak. Her yeni veri kendine özgün bir sayı ile sisteme kaydedilmesi için “Sequence” fonksiyonu ile oluşturuldu.

{  
 CREATE SEQUENCE SEQ\_TICKET

MINVALUE 7000  
 MAXVALUE 7999  
 START WITH 7000  
 INCREMENT BY 1;

}

Graphical user interface, application

Description automatically generated

Figure APEX Oracle "Home Delivery" Sayfası

“Home Delivery” sayfası, eğer gönderici, kargosunun evinden teslim alınıp dağıtıma çıkmasını talep ediyor ise bu sayfadan isim, soy isim ve adres bilgileri girilerek sistem yöneticisine istek ve veri tabanına ilgili bilgiler gönderilir. Her yeni veri kendine özgün bir sayı ile sisteme kaydedilmesi için “Sequence” fonksiyonu ile oluşturuldu.

{  
 CREATE SEQUENCE SEQ\_SENDNO

MINVALUE 6000  
 MAXVALUE 6999  
 START WITH 6000  
 INCREMENT BY 1;

}

Graphical user interface, application

Description automatically generated

Figure APEX Oracle Sayfa Düzenleme Ekranı

Tasarımlar arasından Home Delivery sayfasının, APEX Oracle tasarım ara yüzünde tasarım görüntüsü.

Graphical user interface, application

Description automatically generated

Figure İkinci Dönem "Home" Sayfası

İkinci dönemde “Home” sayfasının işlevi aynı tutularak sadece görünüşü değiştirilip elverişliliği arttırıldı.

Sayfanı en üstüne sayı alanı yerleştirildi. Sayı alanının içerisine bir değer girilip sayfaya ileti gönderildiğinde girilen sayı “Cargo” sınıfı içerisinde birincil anahtarları arayarak otomatik olarak aşağısında yer alan klasik rapora ilgili bilgiler getiriliyor. Raporda tüm sınıf bilgileri kullanılmak istenmediği için bir SQL sorgusu ile “Cargo” sınıfından istenen bilgileri çekmektedir.

{  
SELECT cargo\_type,delivery\_price, cargo\_delivery, delivery\_time, cargo\_warning  
FROM cargo  
WHERE TRACKING\_NO  
=NVL(:P1\_NUMBER, TRACKING\_NO)  
}

Geçerli bir “tracking number” girildiğinde sadece izlenebilecek şekilde değerler gösterilecektir. Geçersiz girildiğinde ise boş bir form göstererek değerin veri tabanında olmadığını anlatmaktadır.

Graphical user interface, application

Description automatically generated

Figure 16 "Calculate Price" Sayfası

“Calculate Price” sayfası işlevsel hale getirildi. Cargo türünün seçilmesinin ardından gerekli alanların bilgileri girilerek yapılacak hesapların kolaylaştırılması sağlandı. Ücret hesabında en, boy ve yükseklik verilerini 3000 ile çarpıp desi değeri alınıyor ve her desi katsayısı için 10 ile çarparak tahmini ücret hesaplanıyor. Kargo ücretlerinin hesaplanmasında yaklaşık değerler alınmakla beraber gerçek hayatı yansıtmamaktadırlar.

Sayfanın en üzerine bir seçme listesi ile ne tür bir kargonun fiyatı hesaplanacağı seçildikten hemen sonra aktif kalan sayı alanlarına gerekli birimler girilerek desi ve fiyat hesaplandırılması yapılır.

Graphical user interface, application

Description automatically generated

Figure "Cargo Update Information" Sayfası

“Cargo Update Information” sayfası eklenerek kargonun alıcıya kadar ulaşması süresince değişecek tüm değer ve durumların; sistem ve kargocu tarafından gerekli olmayan ve özgün değerler dışında manuel olarak güncellenebilmesini sağlar.

Sayfanın en üzerinde yer alan sayı alanına girilen değer tıpkı “Home” sayfasında olduğu gibi SQL sorgusu ile altında yer alan etkileşimli form yapısına istenen tüm değerleri ile listelenir. Alandaki değerler sadece kuryenin değiştirmesi gereken kısımlar aktif olacak şekilde düzenlenebilir.

Table

Description automatically generated

Figure "Cargo Management" Sayfası

“Cargo Management” sayfası, gönderilmek istenen kargoların bilgisini ve gönderici bilgilerini sisteme kaydederek veri tabanına aktarılmasını sağlar. Yeni veriler kendilerine özgün bir sayı ile sisteme kaydedilmesi için “Sequence” fonksiyonu ile oluşturulmuştur.

Alt alta bulunan formlar sayfanın en altındaki “Create” butonuna basıldığında “Cargo” ve “Recipient” sınıfları aralarında bağ olan “Branch” sınıfının bağı ile veri tabanına aktarılır.

# Veri tabanının Oluşturulması

Kargo sisteminin veri tabanı, herhangi bir referans kullanılmadan Excel sayesinde hücre doldurma, formüller ve rastgele parametreler kullanılarak oluşturuldu. Ve gösterge panellerinin tasarlanmasının ardından APEX ortamına aktarıldı.

Veri tabanının merkezi olan “Branch” sınıfı, birincil anahtar değerlerini ve değerlerin işaret ettiği ilimizin isim bilgilerini tutuyor.

Veri tabanında tutulan şehir isimleri rastgele seçilmiş olup analizde önem arz etmemektedir.

6001-6005 arası değerler aktif çalışma yapılan illeri gösterirken 6011-6054 arası değerler şehirler arası transfer edilmesi gereken kargoların gideceği konumlar arası bilgisini veriyor. Bir kargonun 6011-6054 arası değer alması şehirler arası aktarımda olduğu anlamına gelir.

|  |  |
| --- | --- |
| **BRANCH\_NO** | **BRANCH\_ADRESS** |
| 6001 | Ankara |
| 6002 | Canakkale |
| 6003 | Balikesir |
| 6004 | Izmir |
| 6005 | Aydin |
| 6011 | Ankara - Canakkale |
| 6012 | Ankara - Balikesir |
| 6013 | Ankara - Izmir |
| 6014 | Ankara - Aydin |
| 6021 | Canakkale - Ankara |
| 6022 | Canakkale -Balikesir |
| 6023 | Canakkale -Izmir |
| 6024 | Canakkale - Aydin |
| 6031 | Balikesir - Ankara |
| 6032 | Balikesir - Canakkale |
| 6033 | Balikesir - Izmir |
| 6034 | Balikesir - Aydin |
| 6041 | Izmir - Ankara |
| 6042 | Izmir - Canakkale |
| 6043 | Izmir - Balikesir |
| 6044 | Izmir - Aydin |
| 6051 | Aydin - Ankara |
| 6052 | Aydin - Canakkale |
| 6053 | Aydin - Balikesir |
| 6054 | Aydin – Izmir |

Tablo “Branch” Sınıfı

“Ticket” sınıfı herhangi bir parametre ya da formüle bağlı değildir. İşlevi sistemden gönderilen mesajları tutmak olduğu için sadece gerekli başlıkları içermektedir. Kullanıcılar, sistem üzerinden mesaj gönderdiği sürece “Ticket” sınıfına kaydedilmeye devam edecektir.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TICKET\_NO | TICKET\_NAME | TICKET\_SURNAME | TICKET\_EMAIL | | TICKET\_MESSAGE | |
|  |  |  |  |  | |

Tablo “Ticket” Sınıfı

“Sender” ve “Recipient” sınıfları birbirlerine benzeyen çalışma biçimine sahipler İlk sütun her alıcı ve göndericilerin özgün numaraları olurken, sonraki iki sütun isim ve soyisim bilgilerini barındırır. “BRANCH\_NO” foreign key olarak yer alır ve göndericilerin kargosunu teslim ettiği şubeyi, alıcılar için ise kendilerine en yakın şubeyi işaret eder.

Her bir kullanıcının isimleri [“Fantasy Name Generators”](https://www.fantasynamegenerators.com/human-wow-names.php) internet sitesi üzerinden rastgele oluşturularak kullanılmıştır ve gerçek kişiler ile alakaları yoktur.

|  |  |  |  |
| --- | --- | --- | --- |
| SEND\_NO | SEND\_NAME | SEND\_SURNAME | BRANCH\_BRANCH\_NO |
| 1001 | Thierry | Shirley | 6001 |
| 1002 | Leodegrance | Beasant | 6001 |
| 1003 | Gavin | Earle | 6001 |
| 1004 | Ywan | Swett | 6001 |
| 1005 | Ywan | Breeden | 6001 |

|  |  |  |  |
| --- | --- | --- | --- |
| RECIP\_NO | RECIP\_NAME | RECIP\_SURNAME | BRANCH\_BRANCH\_NO |
| 2001 | Emery | Davenpurt | 6001 |
| 2002 | Thaddeus | Watt | 6001 |
| 2003 | Eliott | Kirby | 6001 |
| 2004 | Fitz | Foy | 6001 |
| 2005 | Hardey | Kendall | 6001 |

*Table 6 “Recipient” Sınıfı*

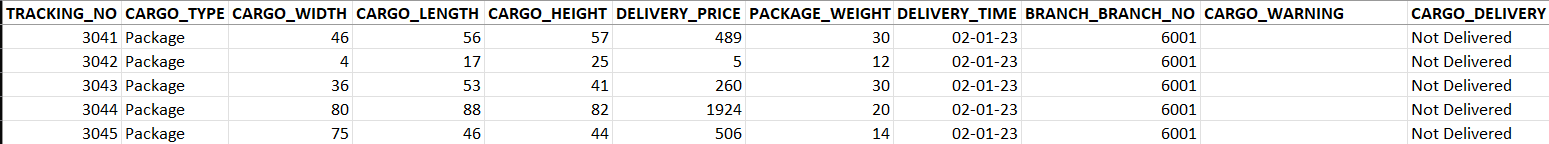
“Cargo” sınıfında, kargonun boyutları ve ağırlığı hesaba katılarak desisi hesaplanır ve 1 desi birimi 10 ile çarpılarak kargo gönderiminin ücreti hesaplanır. Örnek olarak kullanılan Excel formüllerinden biri: { =((C48\*D48\*E48)/3000)\*10 }  
“Cargo Warning” özelliği barındırarak kuryenin kargo hakkında not bırakması gerekmesi durumunda kullanıma hazır bulunacaktır.  
 Veri tabanı, sistemin 1 Ocak – 4 Ocak arası çalışması süresinde kaydedilen verileri tutmaktadır. Dinamik bir verit banı yaratılmadığından dolayı sadece tanımlanan aralıklar arasında teslim edilmesi gereken kargoların değer ve işlemleri bulunmaktadır.

Figure 19 "Cargo" Sınıfı

“Courier” Sınıfı, tüm sınıfların sahip olduğu gibi kullanıcıları tanımlamak üzere özgün değerler barındırmasının yanında kuryelerin isim, soy isim, maaş, teslim edecekleri kargolar ve bulundukları şubeleri gösteren özellikler içerir.

Maaş bilgileri Excel üzerinden { =RANDBETWEEN(5200,7900) } formülü kullanılarak 5200 ve 7900 birimleri arasında rastgele tam sayılar üreterek elde edilmiştir.

Graphical user interface

Description automatically generated with medium confidenceKuryelerin dağıtacağı kargolar ise  
{ =CONCATENATE(TRANSPOSE (I13:I16)) } formülü kullanılarak yerleştirilmiştir. Bu formülde CONCATENATE () kullanılarak bir sütun halinde olan kargo numaralarının tek bir dizede birleştirmek için kullanılmış, değerlerimiz tek bir dizinde olsa bile dikey konumda oldukları için TRANSPOSE() komutunu kullanıp dizini yatay hale getirerek “Courier” sınıfına kolayca yerleştirebilmemizi sağlıyor.

Figure 20 "Courier" Sınıfı

# İSTATİSTİK VE SONUÇ

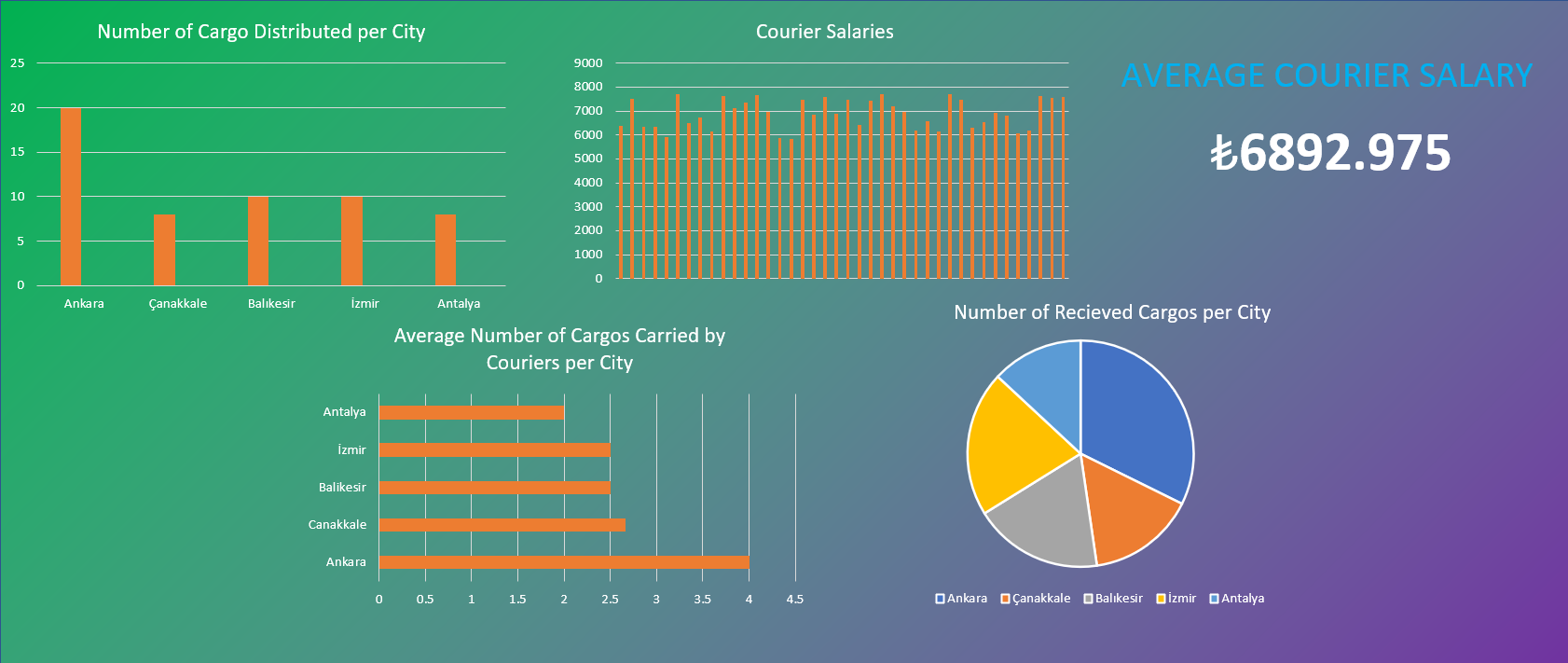
Manuel olarak oluşturulan veri tabanında istatistik için “Cargo”, “Courier”, “Recipient” ve “Sender” sınıfları kullanılarak gösterge panelleri oluşturuldu ve olabildiğince yoruma açık tablolar yaratılarak projenin içinde istatistik dahil edildi.

Figure "Courier" Sınıfı Bilgisi ile Gösterge Paneli

“Number of Cargo Distibuted per City” grafiği, şehir içi kargo gönderimlerini gösteren sütun grafiği olacak şekilde tasarlandı. Verilerden anlaşıldığı gibi en çok şehir içi gönderim yapan şehir Ankara olurken çalışma yürütülen diğer konumlarda alıcı ve göndericilerinin aynı şehirlerde olduğu anlaşılıyor.

“Courier Salaries” grafiği, sistemde çalışan tüm kuryelerin maaşlarını gösteren bir sütun çizelgesi ve yanında “Courier Salaries” grafiği ile alakalı “Average Courier Salary” bölümünde sistemdeki tüm kuryelerin maaşlarının ortalaması yer alıyor. Sütun grafiği incelendiğinde sistemdeki kuryelerin ₺6200 ve ₺7900 arasında ücret aldığı ve ortalamalarının ₺6892.98 olduğu anlaşılıyor.

“Average Number of Cargos by Couriers per City” grafiği, kargocuların taşıma yaptığı şehirlerde tek seferde ortalama kaç kargo taşımak zorunda olduklarını gösteriyor. Bar grafiğinden anlaşılacağı üzere kargocular Ankara’da en çok kargoyu taşımak zorundayken tüm şehirler arasında Antalya’da en hafif iş yükü olduğu okunabilmektedir. Sistemdeki diğer şehirlerin kargo yükünün ise birbirine yakın olduğu anlaşılmaktadır.

“Number of Recieved Cargos per City” grafiği, dairesel bir grafik olarak şehirlerde toplam ne kadar kargo teslim edilmesi gerektiğini göstermektedir. Bütün incelendiğinde en çok Ankara ve İzmir’de teslim edilmesi gereken kargoların olduğu, sistemdeki diğer şehirlerin göreceli olarak daha da az kargo teslim edilmesi gerektiği okunmaktadır.

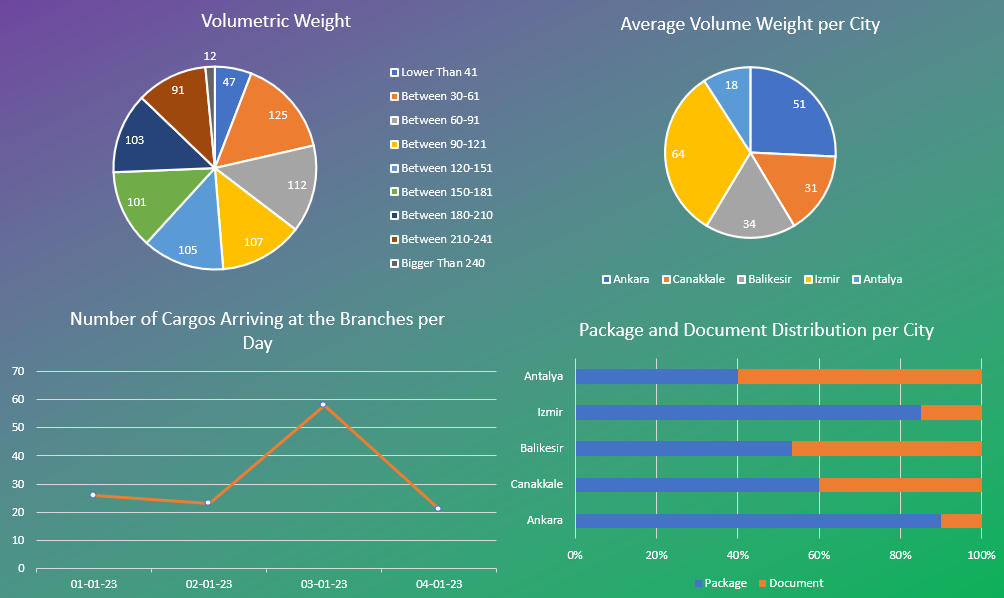
 “Volumetric Weight” grafiği, sistemdeki tüm kargoların desi aralıklarını göstermektedir. Dairesel grafikten okunacağı üzere en çok 30 ve 61 desi aralığında kargo bulunurken en az 240’tan fazla desiler sitemde yer almaktadır. Geri kalan desi aralıklarının gösterdiği üzere 210-241 ve 120-151 desi aralıkları en az kargo barındıran gruplar olurken geri kalan tüm aralıklar birbirine yaklaşık sayılarda kargo barındırmaktadır.

Figure "Cargo" Sınıfı Bilgisi ile Gösterge Paneli

“Average Volume Weight per City” grafiği, tüm şehirlerin aldığı kargoların ortalama desilerini göstermektedir. Dairesel grafikten okunduğu üzere İzmir’e ulaşan kargolar en çok desi ortalamasına sahipken Antalya’da kargo desilerinin düştüğü okunuyor.

“Number of Cargos Arriving at the Branches per Day” grafiği, sistemin temsil ettiği 4 günlük süre içerisinde her gün ne kadar kargo alındığını gösterir. Çizgi grafiğinden okunduğu üzere üçüncü gün en çok kargo alımı yapılırken diğer günler alınan kargo sayılarının birbirlerine oranla daha yakın olduğu okunuyor.

“Package and Document Disturbation per City” grafiği, Sistemdeki her şehrin aldığı paket ve belge kargolarının oranlarını göstermektedir. %100 çubuk grafiğinden okunduğu üzere İzmir ve Ankara’da paket kargo yoğunluğu yaşanırken sadece Antalya’da belge kargo oranının daha fazla olduğu ve sistemdeki diğer şehirlerin paket kargo teslimatı daha fazla olacak şekilde benzer olduğu gözlemleniyor.