1. **pl-healtcheck resource group içine Azure SQL Database Setup ve Storage Account Açılacak**

Azure SQL database içinde oluşturulması gereken tablolar:

1. Depo Lokasyon Table (Attributes: Depot ID, Lokasyon)
2. Depo Product Table (Attributes: Depot ID, Product ID, Weight, Max Quantity, Ürün Çeşidi)
3. Placement Table (Attributes: Depot ID, Lokasyon ID, Product ID, Lokasyon Ağırlık Grubu, Ürün Ağırlık Grubu, Max Quantity, Koli Miktarı, Inserted\_At)
4. Depo Lokasyon History For Slotting (Attributes: Depot ID, Type(Slotting veya Replenishment veya Manuel) Lokasyon, Product ID, Koli Miktarı, Lokasyon Ağırlık Grubu, Ürün Ağırlık Grubu, Max Quantity, Inserted\_At (WMS'den gelmeyecek))
5. Depo Rotalama Table (Attributes: depot id, orderId, original\_length, found\_length, difference (km), difference (perc), orig\_order, found\_order, orig\_path, found\_path,Weight\_Rule, Current\_Time (WMS'den gelmeyecek. DB’ye yüklenen Zaman eklenecek.))
6. Storage Account Blob Storage içinde her bir Depoya özel klasörler olacak (02579e13-8d4f-4463-968d-7affe4b9919f). Bu klasörlerin içinde olacak dosyalar:

* Distance\_Matrix Pickle Dosyası
* Slotting Klasörü 🡪 Eski ve Yeni bütün slotting sonuçları JSON formatında buraya eklenecek.
* Orders Klasörü 🡪 Rotalamadan çıkan result csvler buraya gelicek. CSV yüklenince Depo Rotalama Tablosuna veri yazılmak üzere Function app trigger olacak.
* Replenishment Klasörü 🡪 Depodan toplanacak ürünler ve koli miktarı gelince Replenishment Script trigger edilecek.
* Pointslookup (Cartezyen koordinatlar 0’a 0)
* Segments

1. **pl-healtcheck resource group içine Azure Cache For Redis Setup**

Python Scriptlerimiz için Status durumu konulacak.

* Slotting (Status: "Idle", "In Progress")
* Replenishment (Status: "Idle", "In Progress")

Replenishment scripti çalışmaya başlayınca Redis tarafında status "In Progress" olacak.

Slotting scripti çalışmaya başlayınca Redis tarafında status “In Progress” olacak.

Slotting scripti çalışmaya başlanırsa Replenishment yapılamaz. Tam terside geçerli (Replenishment çalışırken Slotting layout update yapılamaz. )

1. **Her bir order Replenishment İşlenmesinden sonra (check\_stock --> loc\_pic --> replenishment) SQL Database Update yapılacak.**

Replenishment Script’inde bulunan check\_stock, loc\_pic ve replenishment fonksiyonlarına giren her bir orderın sonunda “Replenishment Tablosu” güncellenecek. Aynı zamanda Depo doluluk oranı güncellenecek. Ve de yeni değişiklikler “Depo Lokasyon History For Slotting” tablosuna current time alınarak yazılacak.

1. **Ağırlık kategorisine, Ürün Kategorisine ve Ürün Ağırlık Grubu- Lokasyon Ağırlık Grubu Uymayan Lokasyonlar filtleleri için Function App (HTTP Trigger GET)**

HTTP Get Request ile birlikte gelen depo id ve filtreleme çeşidine göre (Ağırlık kategorisi, Ürün kategorisi, Ürün Ağırlık Grubu – Lokasyon Ağırlık Grubu Uymayan Lokasyonlar vs.) SQL sorgusu çalışacak. Bu filtreler birleştirebilir. SQL sorgusunun çıktısı JSON formatında return edilecek. (HTTP Post Request)

**Gelmesi Gereken JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Filtreler”: [1,2]

“Parametre- Ağırlık”: 5,

“Parametre-Category”: “tatlı”

}

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Filtreler”: [1],

“Parametre-Ağırlık”: “tatlı”

}

**Gönderilecek JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“DepoDoluluk”: 84.56, (Yüzde 84.56 dolu diye göndereceğiz.)

“Info” : [ { “Product ID”: 1234578,

“Lokasyon”: “4A001K1”,

“Lokasyon Ağırlık Grubu”: 5,

“Ürün Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

},

{“Product ID”: 7777888,

“Lokasyon”: “4A002K1”,

“Lokasyon Ağırlık Grubu”: 5,

“Ürün Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

}, etc…

]

}

1. **Lokasyonda bulunan ürüne ait bilgilerin ve Depo doluluk oranının Refresh butonu sayesinde Server Side'dan çekilmesi (HTTP Trigger Get)**

Koşul: Replenishment ve Slotting Script Status "Idle" olmalı. Eğer Idle değilse Server Side’tan Client tarafına hata mesajı gönderilecek.

HTTP Get Request ile birlikte gelen depo id ile SQL sorgusu çalışacak. SQL sorgusunun çıktısı JSON formatında return edilecek. (HTTP Post Request)

Return parametreleri (Placement Table):

* Depot ID
* Product ID
* Lokasyon
* Lokasyon Ağırlık Grubu
* Ürün Ağırlık Grubu
* Max Quantity
* Koli Miktarı
* İnserted\_At

Return parametreleri (Placement Table’dan depo doluluk oranını hesaplama ve return et):

* Depot ID
* Doluluk Oranı

**Gelmesi Gereken JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

}

**Gönderilecek JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“DepoDoluluk”: 84.56, (Yüzde 84.56 dolu diye göndereceğiz.)

“Info” : [ { “Product ID”: 1234578,

“Lokasyon”: “4A001K1”,

“Lokasyon Ağırlık Grubu”: 5,

“Ürün Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

},

{“Product ID”: 7777888,

“Lokasyon”: “4A002K1”,

“Lokasyon Ağırlık Grubu”: 5,

“Ürün Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

}, etc…

]

}

1. **Slotting Historysinin Web Tarafında GetAllLocationHistory Butonu ile çekilmesi için Function App (HTTP Trigger Get)**

* HTTP Get Request ile birlikte gelen depo id yi ve geçmiş tarih aralığını kullanarak GetAll history listesini “Depo Lokasyon History For Slotting” tablosundan SQL sorgusuyla çekeceğiz.
* “Depo Lokasyon History For Slotting” tablosundan gelen SQL sorgusunun cevabı bir tabloda gösterilecek.
* SQL sorgusunun çıktısı JSON formatında return edilecek (HTTP Post Request).

Return parametreleri (“Depo Lokasyon History For Slotting” Tablosundan):

* Depot ID
* Type (Slotting veya Replenishment)
* Lokasyon
* Product ID
* Koli Miktarı
* Lokasyon Ağırlık Grubu
* Ürün Ağırlık Grubu
* Max Quantity
* Inserted\_At

**Gelmesi Gereken JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Tarih (Lower-Bound)”: Datetime(12-03-2021),

“Tarih (Upper-Bound)”: Datetime(23-03-2021)

}

**Gönderilecek JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Info” : [ { “Product ID”: 1234578,

“Type”: “Slotting or Replenishment or Manuel”, (String formatında)

“Lokasyon”: 5,

“Ürün Ağırlık Grubu”: 5,

“Lokasyon Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

},

{ “Product ID”: 1234578,

“Type”: “Slotting or Replenishment or Manuel”, (String formatında)

“Lokasyon”: 5,

“Ürün Ağırlık Grubu”: 5,

“Lokasyon Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

}, etc…

]

}

1. **Slotting Historysinin Web Tarafında GetLocationHistory Butonu ile çekilmesi için Function App (HTTP Trigger GET)**

* HTTP Get Request ile birlikte gelen depo id yi, geçmiş tarih aralığını ve lokasyon bilgisini kullanarak spesifik lokasyondaki history listesini “Depo Lokasyon History For Slotting” tablosundan SQL sorgusuyla çekeceğiz.
* “Depo Lokasyon History For Slotting” tablosundan gelen SQL sorgusunun cevabı bir tabloda gösterilecek.
* SQL sorgusunun çıktısı JSON formatında return edilecek (HTTP Post Request).

Return parametreleri (“Depo Lokasyon History For Slotting” Tablosundan):

* Depot ID
* Type (Slotting veya Replenishment)
* Lokasyon
* Product ID
* Koli Miktarı
* Lokasyon Ağırlık Grubu
* Ürün Ağırlık Grubu
* Max Quantity
* Inserted\_At

**Gelmesi Gereken JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Lokasyon”: “4A001K1”,

“Tarih (Lower-Bound)”: Datetime(12-03-2021),

“Tarih (Upper-Bound)”: Datetime(23-03-2021)

}

**Gönderilecek JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Info” : [ { “Product ID”: 1234578,

“Type”: “Slotting or Replenishment or Manuel”, (String formatında)

“Lokasyon”: 5,

“Ürün Ağırlık Grubu”: 5,

“Lokasyon Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

},

{ “Product ID”: 1234578,

“Type”: “Slotting or Replenishment or Manuel”, (String formatında)

“Lokasyon”: 5,

“Ürün Ağırlık Grubu”: 5,

“Lokasyon Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

}, etc…

]

}

1. **Rotalama Historysini Web Tarafında Tarih aralığı ile GetRotalamaHistory butonu ile çekilmesi için Function App (HTTP Trigger GET)**

* HTTP Get Request ile birlikte gelen depo id yi ve geçmiş tarih aralığı bilgisini kullanarak “Depo Rotalama Table” tablosundan SQL sorgusuyla siparişleri çekeceğiz.
* “Depo Rotalama Table” tablosundan gelen SQL sorgusunun cevabı Mockuptaki orders tablosundan gözükecek. İstenilen siparişin görseli simulasyon üzerinde yapılacak.
* SQL sorgusunun çıktısı JSON formatında return edilecek (HTTP Post Request).

Return parametreleri (“Depo Rotalama Table” Tablosundan):

* Order ID
* Orig Distance
* Found Distance
* Verimlilik (Metre)
* Verimlilik (%)
* Orig Full Path
* Found Full Path
* Weight Kuralı (True veya False)
* Time

**JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Info” : [ { “OrderID”: “131 – Okan Demirel-123456”,

“Orig Distance”: 260, (Metre)

“Found Distance”: 200, (Metre)

“Verimlilik (Metre)”: 60,

“Verimlilik (%)”: 23,

“Orig Full Path”: [“start”, “yol-8”, “yol-7”, …., “start”],

“Found Full Path”: [“start”, “yol-8”, “yol-7”, …., “start”],

“Weight Kuralı”: “True veya False”,

“Time”: Datetime(23-03-2021 14:55:40),

“Points”: [ { “Location”: “4A00K1”,

“Lat”: 12355.65464,

“Lon”: 4363.12121

}, { “Location”: “4B030K1”,

“Lat”: 12355.65464,

“Lon”: 4363.12121

}, etc.]

},

{ “OrderID”: “131 – Okan Demirel-888888”,

“Orig Distance”: 260, (Metre)

“Found Distance”: 200, (Metre)

“Verimlilik (Metre)”: 60,

“Verimlilik (%)”: 23,

“Orig Full Path”: [“start”, “yol-8”, “yol-7”, …., “start”],

“Found Full Path”: [“start”, “yol-8”, “yol-7”, …., “start”],

“Weight Kuralı”: “True veya False”

“Time”: Datetime(23-03-2021 14:55:40)

“Points”: [ { “Location”: “4A00K1”,

“Lat”: 12355.65464,

“Lon”: 4363.12121

}, { “Location”: “4B030K1”,

“Lat”: 12355.65464,

“Lon”: 4363.12121

}, etc.]

},

},

]

}

1. **Manuel olarak depo üzerinde değişiklik servisi (HTTP Trigger POST)**

* Koşul: Replenishment ve Slotting status "Idle" olması gerekiyor yoksa manuel değişikliğe izin verme. Eğer “Idle” değilse Server Side’dan Client Side’a hata mesajı dönecek.
* Alınacak parametrelerle database güncellenecek ve güncellenen layout görselleştirilmesi yapılması üzerine Serhata gönderilecek (HTTP Trigger Post)

Client tarafından Aksiyonlar:

Dropdown list 🡪 Add, Delete

Required Parameters From Client:

* Depot ID
* Product ID
* Operation Type(Add or Delete)
* Lokasyon
* Koli Miktarı (Client bu bilgiyi doldurmak zorunda değil. Eğer boş gönderirse Lokasyona Tam palet yerleştirmek istediği anlaşılacak. Eğer miktar girerse sadece girdiği miktar kadar lokasyondaki stok değişecek.)
* Current\_Time

NOT: Lokasyon Ağırlık Grubu, Ürün Ağırlık Grubu, Koli Miktarı ve Max Quantity gibi parametreler Databaseden çekilebilir. Client tarafından almaya gerek yok.

**SERHAT NOT:** Add fonkiyonuyla user iki farklı aksiyon alabilir.

1. Koli Miktarı input’u girmeden boş lokasyona tam palet yerleştirmek
2. Koli Miktarı girerek boş lokasyona veya bir lokasyon üzerine koli eklemek

Delete fonkiyonuyla user iki farklı aksiyon alabilir.

1. Koli Miktarı input’u girmeden lokasyonu boşaltmak
2. Koli Miktarı girerek lokasyondan belirli bir koli eksiltmek.

**Gelmesi Gereken JSON Formatı:**

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“Operation Type”: “Add or Delete”(String),

“ProductID”: 12345,

“Lokasyon”: “4A001K1”,

“Koli Miktarı”: (Null or İnteger),

“Current Time”: Datetime(23-03-2021 14:55:40)

}

**Gönderilecek JSON Formatı:**

**IMPORTANT:** Bütün layout verisini göndermek yerine spesifik lokasyondan gerçekleşen updateleri göndersek yeterli olacak.

{

“DepotID” : “02579e13-8d4f-4463-968d-7affe4b9919f”,

“DepoDoluluk”: 84.56, (Yüzde 84.56 dolu diye göndereceğiz.)

“Info” : [ { “Product ID”: 1234578,

“Lokasyon”: “4A001K1”,

“Lokasyon Ağırlık Grubu”: 5,

“Ürün Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

},

{“Product ID”: 7777888,

“Lokasyon”: “4A002K1”,

“Lokasyon Ağırlık Grubu”: 5,

“Ürün Ağırlık Grubu”: 5,

“Max Quantity”: 100,

“Koli Miktarı”: 60,

“Inserted\_At”: Datetime(23-03-2021 14:55:40)

}, etc…

]

}

True False Yada Error

1. **New Slotting Layout to Database**

* Slotting Layout JSON formatında Client tarafından Upload edilecek. Upload edilecek dosya depot id ile birlikte “Slotting” klasörüne yüklenecek.

1. **Get SlottingLayout (HTTP Trigger POST)**

* Koşul: Replenishment Script status “Idle” olmalı. Eğer “Idle” değilse Server Side’dan Client Side’a hata mesajı dönecek.
* Client tarafından gönderilen JSON formatında data Blob Storage’a gelince Function App trigger edilecek(Blob Storage Trigger). JSON içindeki veri “Placement Table” tablosuna yazılacak.
* Blob Storage trigger edildikten sonra çalışacak olan Function App’te Hem Replenishment status “In Progress” hemde Slotting status “In Progress” olacak.
* New Slotting “Placement Table” a aktarıldıktan sonra güncel layout simulasyona dönülecek (HTTP Trigger Post).Refresh butonuyla gönderilen JSON formatıyla aynı formatta dönüş yapılacak.
* Depo doluluk oranı güncellenecek.
* Son olarak Replenishment Status: “Idle” ve Slotting Status: “Idle” olucak.

1. **Entegrasyon Rotalama**

* Rotalamadan yeni çıkmış olan result csv dosyaları Client tarafından Upload edilecek.
* Upload edilecek dosya depot id ile birlikte “Orders” klasörüne yüklenecek.
* Orders Blob Storage’a gelince Function App trigger edilecek(Blob Storage Trigger). Orders içindeki veri “Depo Rotalama Table” tablosuna yazılacak. Current time kullanılarak yazılacak.

1. **New Replenishment Orders to Database**

* Toplanacak OrderID, Product ID ve Koli miktarı içeren Replenishment csv dosyaları Client tarafından Upload edilecek.
* Upload edilecek dosya depot id ile birlikte “Replenishment” klasörüne yüklenecek.
* Orders Blob Storage’a gelince Function App trigger edilecek(Blob Storage Trigger). Replenishment Scripti çalışmaya başlayacak. Current time alınacak. Her bir order sonunda Replenishment sonuçları “Placement Tablosu”na eklenecek.