## Achtergrond

Dochterbedrijf Share 4PL vult *inslagen* in binnen een Warehouse Management System (WMS - door onze IT geschreven). De situatie is als volgt. Share 4PL gebruikt de diensten van Ludoma Services B.V. om de goederen voor haar klanten op te slaan in een warenhuis (Tilburg). Share 4PL weet vaak al van haar klanten dat er een zending aan zit te komen. Op dit punt vullen zijn binnen het WMS de informatie van deze zending in, dit noemen wij een inslag in het WMS.

De IT afdeling van Share Logistics/Share 4PL werkt samen met de IT van Ludoma om hier via webservices JSON berichten over uit te wisselen. Echter, dit zal pas later dit jaar live gaan. Tot die tijd zou Ludoma toch graag de informatie van een bovengenoemde inslag bijtijds willen ontvangen, bij wijze van ‘pre alert’/vooraanmelding. Dit dient te gebeuren in een email via Microsoft Outlook.

## Opdracht

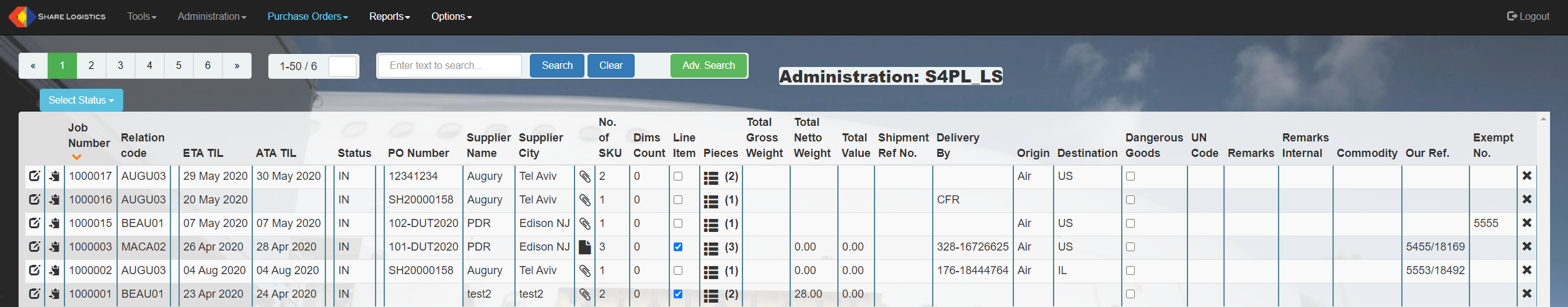
Creëer in Visual Studio een Windows Forms C# applicatie, welke de volgende taken uitvoert:

1. Voer het meegeleverde InboundScript.sql uit in bv. SQL Management Studio (gratis). Maak hierbij eerst een lege database in, voordat je binnen deze database (use <dbnaam>;) het script uitvoert. Hiermee worden 4 tables incl. (test)data aangemaakt.
2. Lees in de C# applicatie de inslagdata uit de meegeleverde tabellen m.b.v. een Microsoft SQL database verbinding. Hiermee dient er dus een database verbinding te komen met jouw aangemaakte lokale database (welke gevuld is met het meegeleverde script).
3. Voeg een veld aan een relevante tabel toe genaamd ‘PREALERT’ van het type bit (boolean).
4. Creëer in de C# applicatie een timer, welke elke 5 minuten controleert of van de huidige inslagen er een PREALERT is gestuurd. Bij het lezen van de inslagdata, dient er dus een extra conditie te komen in de query welke kijkt of er eerder een PREALERT is gestuurd. (om te voorkomen dat Ludoma elke 5 minuten een mail ontvangt).
5. Verken de mogelijkheid om op basis van de uitgelezen inslagdata een email op te laten stellen met alle relevante informatie. Als voorbeeld van welke informatie relevant is, kunnen de screenshot in de volgende sectie gebruikt worden. Iets soortgelijks zou als email body opgesteld dienen te worden welke dan direct opent in Microsoft Outlook.  
   Refereer naar de Microsoft Visual Studio documentatie zoals:  
   <https://docs.microsoft.com/en-us/visualstudio/vsto/how-to-programmatically-create-an-e-mail-item?view=vs-2019>  
     
   Hierin is een mate van vrijheid aanwezig. Het mooist zou zijn als er in de code een HTML template kan worden opgesteld op basis van de uitgelezen inslagdata. Waarna de HTML template wordt getoond als email body in een nieuwe email.
6. Maak in het Windows Form een textbox mogelijkheid om het emailadres van de te ontvangen persoon in te vullen.
7. Indien een email body is opgesteld o.b.v. de uitgelezen inslagdata, dan dient dit record geüpdatet te worden, zodat het veld PREALERT op 1/true staat (en er in het vervolg niet nog een PREALERT mail wordt opgesteld voor deze inslag).

### Tip(s)

* Probeer als laatste stap pas *alle* velden te tonen in een email (gebaseerd op de screenshots). Het makkelijkst is om bv. eerst alleen de velden INBOUNDID, RELATIONCODE en PONUMBER op te halen en te tonen in een email. Bouw het stap voor stap op.
* Er zal een situatie ontstaan waar je door de uitgelezen inslagen gaat loopen (6 records zijn meegeleverd). Waarin je gedurende het loopen op ‘goederen/colli’ niveau zal gaan loopen, om zo de goederen per inslag uit te kunnen uitlezen. Er zal dus een *nested loop* plaatsvinden.

### Screenshots en overige informatie

In het WMS wordt er onderscheid gemaakt tussen 2 niveau’s in het WMS: het hoofdniveau en het ‘colli’ (goederen) niveau. Dit is een screenshot (hoofdniveau) welke overeenkomt met de meegeleverde database informatie:  


Hier gebruiken we de volgende query, waar ook een aantal velden niet worden gebruikt in deze versie van het WMS. Deze query geeft wel een idee hoe de gebruikte tabellen met elkaar verbonden zijn middels JOINs (en wat de primary/foreign keys zijn):

WITH ResultSet AS (SELECT distinct(TRITPurchaseOrder.ID), TRITPurchaseOrder.PO\_NUMBER, TRITPurchaseOrder.INBOUNDID, convert(varchar, TRITShipment.[DATE], 106) as INBOUNDDATE, TRITShipment.SHIPMENT\_REF\_NO, TRITItem.PO\_NUMBERS\_NO, TRITItem.COLLIES, TRITItem.DIMENSIONS\_NO AS DIMENSIONS\_NO\_PERITEM, TRITPurchaseOrder.DIMENSIONS\_NO, TRITItem.KGS, TRITPurchaseOrder.LOCATION, TRITPurchaseOrder.STATUS\_DEFAULT, TRITPurchaseOrder.CREATED\_BY, TRITShipment.ITEMS\_TOTAL, TRITShipment.POL, TRITShipment.POD, TRITShipment.VESSEL, TRITShipment.REMARKS, convert(varchar, TRITShipment.ETA\_DATE, 106) as ETA\_DATE, TRITShipment.ETA\_TIME, TRITPurchaseOrder.JOBNUMBER, TRITItem.COMMODITY, TRITPurchaseOrder.CUSTOMSSTATUS, UPPER(TRITPurchaseOrder.MRN) AS MRN, convert(varchar, TRITPurchaseOrder.MRNVALIDTO, 106) MRNVALIDTO, TRITShipment.SUPPLIER\_NAME, TRITShipment.SUPPLIER\_REF, TRITShipment.SUPPLIER\_COUNTRY, TRITShipment.SUPPLIER\_CITY, TRITShipment.CUSTOMER\_NAME, TRITPurchaseOrder.RELATIONCODE, TRITPurchaseOrder.DEPOT, TRITPurchaseOrder.TTL, TRITPurchaseOrder.CURRENCY, TRITPurchaseOrder.CUSTOMSTYPE, TRITPurchaseOrder.INCOTERM, TRITPurchaseOrder.COUNTRYCODE, TRITPurchaseOrder.COUNTRYCODEDISPATCH, TRITItem.CRR, TRITItem.TOTALNETTOWEIGHT, TRITItem.TOTALVALUE, TRITItem.ITEM\_REF, TRITPurchaseOrder.PREDEPARTURE, (SELECT ISNULL(SUM([WEIGHT]), 0) FROM TRITItemDimensions WHERE TRITPurchaseOrder.INBOUNDID = TRITItemDimensions.INBOUNDID) AS DIMWEIGHTPERPO, TRITPurchaseOrder.MARKEDASSINGLEPO, TRITShipment.REMARKS\_INTERNAL, TRITShipment.AWB\_NO\_INCOMING, TRITPurchaseOrder.EDI\_STATUS, TRITShipment.SHIPMENT\_REF\_NO\_MASTER, convert(varchar, TRITShipment.SHIPMENT\_DEADLINE, 106) as SHIPMENT\_DEADLINE, TRITShipment.SHIPMENT\_DEADLINE\_TEXT, TRITItem.DANGEROUSGOODS, TRITItem.UNCODE, TRITPurchaseOrder.COURIERCHARGES, TRITPurchaseOrder.CONDITIONOFGOODS, TRITPurchaseOrder.TRACKINGNUMBER, case when len(TRITPurchaseOrder.TRACKINGNUMBER)>=20 then left(TRITPurchaseOrder.TRACKINGNUMBER, 20) + '..' else TRITPurchaseOrder.TRACKINGNUMBER end AS TRACKINGNUMBERFORMATTED, TRITPurchaseOrder.DELIVERYBY, TRITPurchaseOrder.ORIGIN, TRITShipment.DELIVERY\_CITY, convert(varchar, TRITShipment.COLLECTIONDATE, 106) as COLLECTIONDATE, convert(varchar, TRITShipment.ACTUALCOLLECTIONDATE, 106) as ACTUALCOLLECTIONDATE, TRITShipment.PICKUPCOSTS, TRITPurchaseOrder.DESTINATION, TRITPurchaseOrder.MINIMUMTEMP, TRITPurchaseOrder.MAXIMUMTEMP, TRITPurchaseOrder.LINEITEM, TRITPurchaseOrder.EXEMPTNO, TRITPurchaseOrder.TOUPLOAD, TRITPurchaseOrder.UPLOADED, convert(varchar, TRITPurchaseOrder.UPLOADLASTSENT\_DATE, 106) as UPLOADLASTSENT\_DATE, CONVERT(VARCHAR(8), TRITPurchaseOrder.UPLOADLASTSENT\_TIME, 108) AS UPLOADLASTSENT\_TIME, TRITShipment.TRANSPORT\_MODE, TRITShipment.TRANSPORT\_MODE\_TYPE, TRITShipment.HAZARDOUS,

DENSE\_RANK() OVER (ORDER BY CASE WHEN TRITPurchaseOrder.INBOUNDID is null THEN 0 ELSE 1 END, TRITPurchaseOrder.INBOUNDID DESC) AS RowNumber

FROM TRITPurchaseOrder

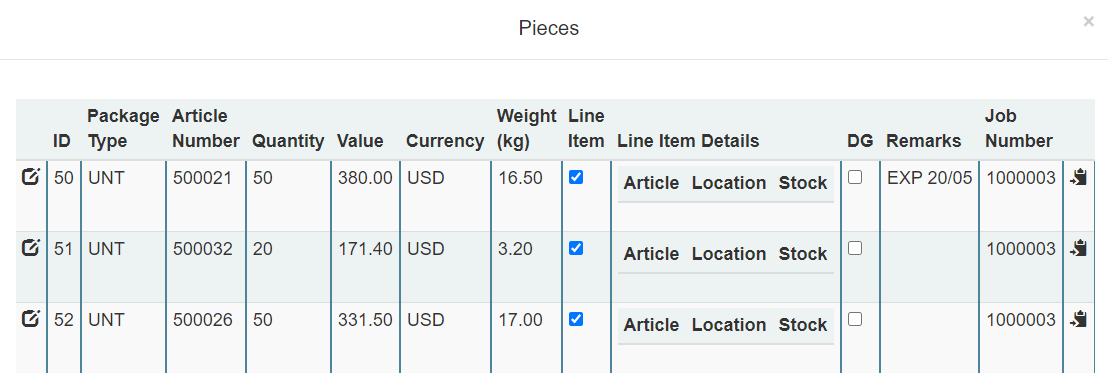
LEFT OUTER JOIN TRITItem ON TRITItem.ID = TRITPurchaseOrder.JOBNUMBER

LEFT OUTER JOIN TRITShipment ON TRITItem.ID = TRITShipment.JOBNUMBER

WHERE TRITPurchaseOrder.DEPOT = 'S4PL\_LS' AND TRITPurchaseOrder.STATUS\_DEFAULT = 'IN')

SELECT \* FROM ResultSet WHERE RowNumber BETWEEN 0 AND 50 ORDER BY INBOUNDID DESC

Hier volgt een screenshot van het colli/goederen niveau:



Bijbehorende query (ook met niet gebruikte velden er tussen):

SELECT TRITItemDimensions.ID, TRITItemDimensions.DIMENSIONS\_NO, TRITItemDimensions.ITEM\_REF, TRITItemDimensions.INBOUNDID, HSCODE, PACKAGETYPE, GOODSDESCRIPTION, HEIGHT, WIDTH, LENGTH, WEIGHT, NETTOWEIGHT, VALUE, TRITItemDimensions.JOBNUMBER, TRITPurchaseOrder.CUSTOMSSTATUS, TRITPurchaseOrder.CUSTOMSTYPE, TRITPurchaseOrder.STATUS\_DEFAULT, TRITItemDimensions.VALUEEUR, TRITItemDimensions.RATE, TRITItemDimensions.CURRENCYCODE, TRITItemDimensions.VATPERCENTAGE, TRITItemDimensions.VATAMOUNT, TRITItemDimensions.IMPORTDUTIESPERCENTAGE, TRITItemDimensions.IMPORTDUTIESAMOUNT, TRITItemDimensions.OTHERDESCRIPTION, TRITItemDimensions.OTHERPERCENTAGE, TRITItemDimensions.OTHERAMOUNT, TRITItemDimensions.LINEITEM, TRITItemDimensions.LINESTOCKAMOUNT, TRITItemDimensions.LINETOTALAMOUNT, TRITItemDimensions.LINECOUNTAMOUNT, TRITItemDimensions.[LOCATION], TRITItemDimensions.DANGEROUSGOODS, TRITItemDimensions.GOODSCONDITION, TRITItemDimensions.REMARKS, TRITItemDimensions.ARTICLE, TRITItemDimensions.QUANTITY, WEIGHT\_DIV, LENGTH\_DIV, WIDTH\_DIV, HEIGHT\_DIV, FLAG, FLAGREASON, REMARKS2 FROM TRITItemDimensions inner join TRITPurchaseOrder on TRITItemDimensions.INBOUNDID = TRITPurchaseOrder.INBOUNDID WHERE TRITItemDimensions.INBOUNDID = <inboundid>