M&S **Food Analytics**

UoD 1b: Tray utilisation/optimization by altering the UPT to better fit a layer.

SQL code and Excel method

August 9

2016

This document gives an overview of the SQL code and method for the tool that Identifies opportunities for improving tray utilization/optimization by altering the UPT to better fit a layer.

Contents

1. Purpose

This document provides the reader an overview of the SQL code and calculations used to produce the output that demonstrates the methodology and SQL code used to generate Tray utilization lists when reducing the UPT value of a product in order to fit better layers and to also effect a bar height reduction as a result identifying the products with a few "loose" units on the Top layer. The idea is to procure these product lists and then determine cost/benefits of reducing the UPT taking into account that such changes will allows allow bar height reductions which will increase the number of tray's and increases products, but also taking account of the reduction of UPTs and any effects this may have on implicated product lines.

Summary: Identify opportunities to improving tray utilisation by altering the UPT to better fit a metric and half tray layer.

2. Pre-requisites for Bar Height reduction tool

2.1. Software Required and Installation

- SAS Enterprise Guide (v.6.1 or higher) or SAS for Windows 9.4 or higher
- Microsoft Excel
  In case of any software installation/issues contact M&S IT Department at 185999

2.2. Accessing the SAs Enterprise Guide Project File and UoD SAs Program

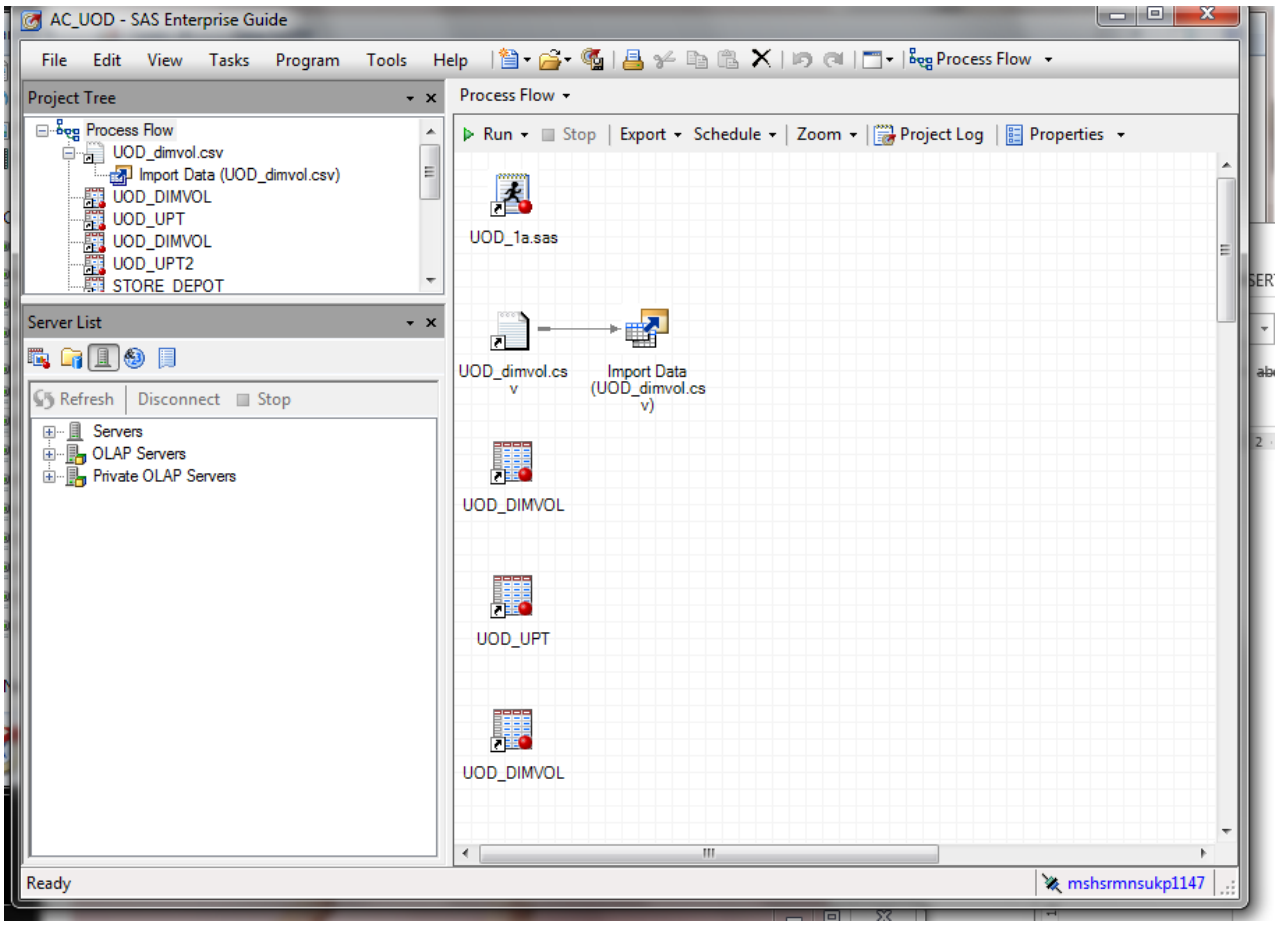
The SAS enterprise Guide Project file and UoD Program file are situated in the Handover folder on the S Drive:

S:\Supply Chain Sharepoint Site\Food Central Support\Data Team\AC\AC Store\Handover\Uod\AC_UOD.egp

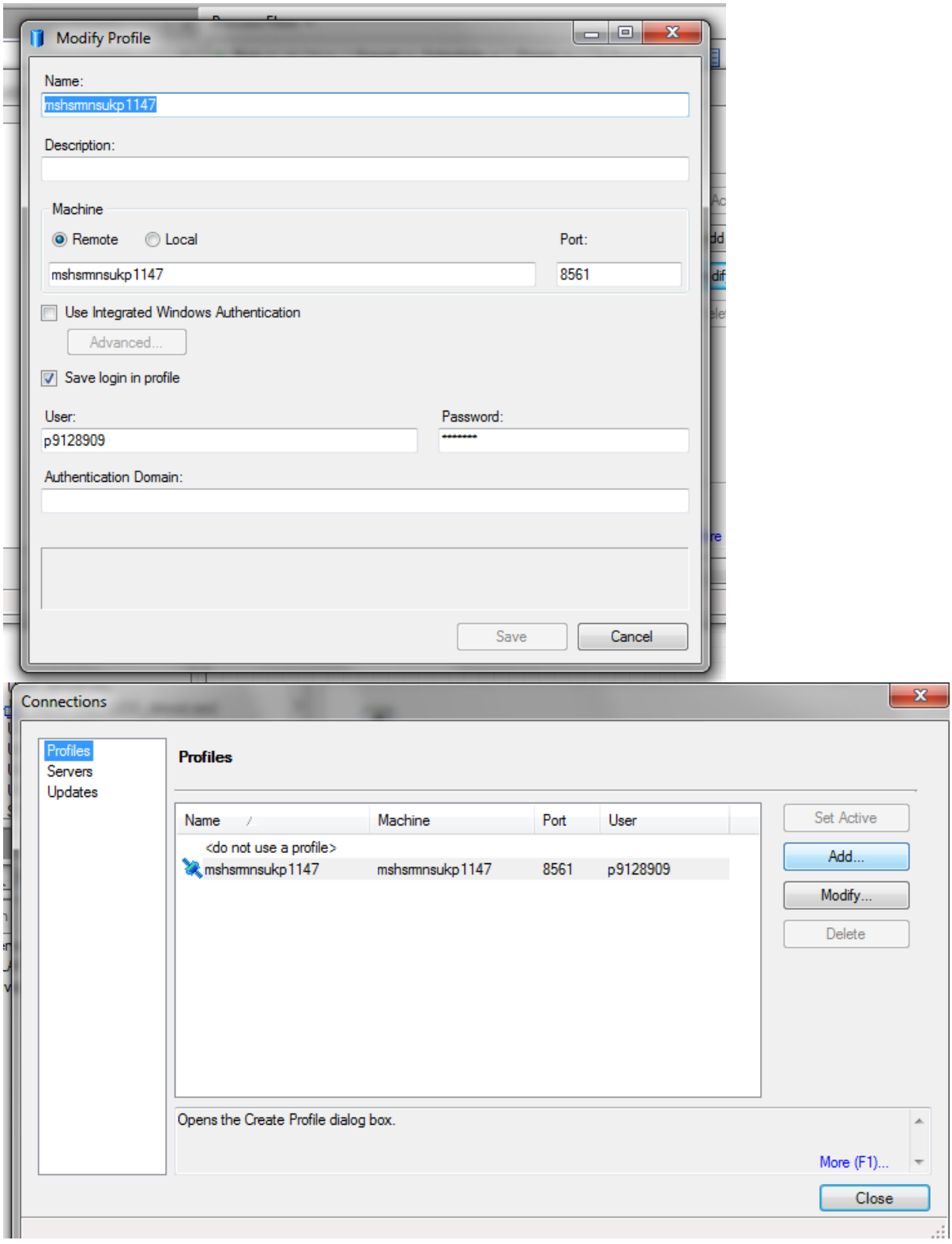
- Before working on the UoD SAS Program it will be best to setup a SAS Enterprise Guide Profile.
- Open the AC_UOD.egp file, and in the bottom right hand corner of the SAS Enterprise Guide window click the given link.
- The profile window appears and you should type in the following details:

1. Open the SAS Guide project file called ‘AC_UOD.egp’ stored at:
S:\Supply Chain Sharepoint Site\Food Central Support\Data Team\AC\AC Store\Handover\Uod

2. Click the profile link at bottom of SAS Enterprise Guide



3. Click the Add button to create a new SAS user profile



4. Click Add button to create a new SAS user profile and use the following details:

name: mshsrmnsukp1147
Machine: mshsrmnsukp1147
Port: 8561
User: use your MNS windows user id and password

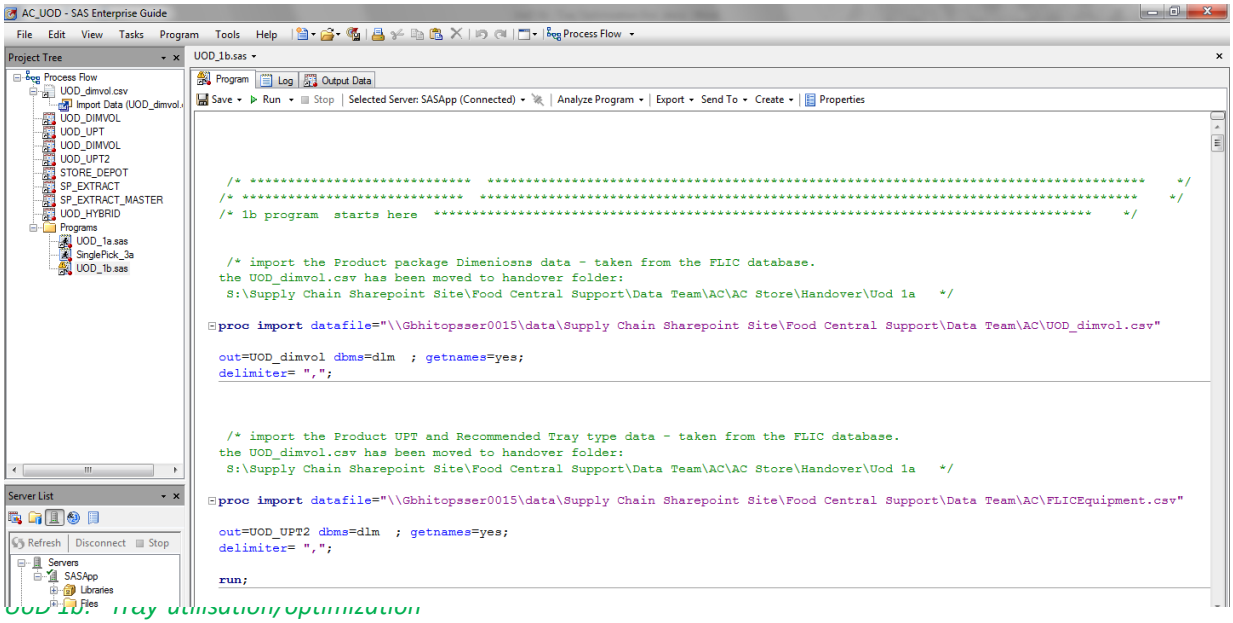
- In the event that you cannot successfully set up a SAS profile please contact Robert Cockburn (Robert.Cockburn@marks-and-spencer.com) of the MNS Big Data team or Sam Swift, analyst for the Food Support team (Sam.Swift@marks-and-spencer.com).

- You can run the UoD program locally (ie, without a SAS profile), but this would mean moving the input datasets to a C drive location altering the file paths to reflect the local C drive locations.

- So, assuming that you now have either a successful SAS profile or you have decided to work locally then you are now ready to open and work with the Uod Bar Height SAS Program, which generates the product lists for the Bar Height reductions.

3.Uod Bar Height SAS Program and running the code

- From the 'Project tree' panel on the left hand side, double click the 'UOD_1a.sas' icon situated under 'Programs'. This will open the Uod Tray Optimization program.



- The two input files 'UOD_dimvol.csv' and 'FLICEquipment.csv' are stored on the server: \\Gbhitopsser0015 and the code references as folder on that server at the start of the Program. However, both these files have also been saved in the Uod Handover folder at: S:\Supply Chain Sharepoint Site\Food Central Support\Data Team\AC\AC Store\Handover\Uod\Uod 1b

- A library folder on the '\\mshsrmnsukp1251' server has been created where all the SAS datasets for this UoD project are stored. This is sensible since this server should have enough capacity to store the datasets needed for this tool.

- The entire Program code for the SAS Program is listed below. It is well commented so that each code block and each all variables and calculations are explained:

```
/* ********************************************************************** */

/* import the Product package Dimensions data - taken from the FLIC
database.
the UOD_dimvol.csv has been moved to handover folder:
 S:\Supply Chain Sharepoint Site\Food Central Support\Data
Team\AC\AC Store\Handover\Uod 1a    */

proc import datafile="\\Gbhitopsser0015\data\Supply Chain Sharepoint
Site\Food Central Support\Data Team\AC\UOD_dimvol.csv"

out=UOD_dimvol dbms=dlm  ; getnames=yes;
delimiter= ",";



 /* import the Product UPT and Recommended Tray type data - taken
from the FLIC database.
the UOD_dimvol.csv has been moved to handover folder:
 S:\Supply Chain Sharepoint Site\Food Central Support\Data
Team\AC\AC Store\Handover\Uod 1a    */

proc import datafile="\\Gbhitopsser0015\data\Supply Chain Sharepoint
Site\Food Central Support\Data Team\AC\FLICEquipment.csv"
```

```
out=UOD_UPT2 dbms=dlm  ; getnames=yes;
delimiter= ",";

run;



/* For Half trays identify UPCs with Exess or surefiet of Units per
tray  */



/* Constants:

11466.414 /  Volume (ml) occupied  for  METRIC_TRAY 2 inch bar
height
22155.444 /  Volume (ml) occupied  for  METRIC_TRAY 5 inch bar
height
33427.512 /  Volume (ml) occupied  for  METRIC_TRAY 7 inch bar
height


4836.483  / Volume (ml) occupied  for  Half_TRAY 2 inch bar height
9957.465  / Volume (ml) occupied  for  Half_TRAY 5 inch bar height
14604.282 / Volume (ml) occupied  for  Half_TRAY 7 inch bar height


 94.83  / Area (mm square) of internal base of Half_TRAY

 51   / Height (mm) of Half-tray with 2 Inch Bar Height
 105  / Height (mm) of Half-tray with 5 Inch Bar Height
 154  / Height (mm) of Half-tray with 7 Inch Bar Height

*/




proc sql ;
/*  create table UOD.NewMetricUod as   */
 create table UOD.Half_UOD_HYBRID as

 Select

/* use the variables  from 'UPC' to 'UPT  - ( UPT -
TopLayerUnits_7in ) as     Units_Remove'  for 7-inch to 5-inch
reduction
```

```
 list, casued by removing excess units from top layer where the
product is packed in a 7 inch tray and changing to x complete layers
of
 a tray with a 5-inch bar height*/

/* UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
 ,Max_UPL_PERLAYER_7in
,Total_Layers_used_7in
,Full_Layers_used_7in
,Excess_TLayUnits_7in
,TopLayerUnits_7in
,Max_UPT_Poss_7in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
,Fin_Bar_Desig
, UPT
, ( UPT - TopLayerUnits_7in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_7in ) as     Units_Remove
*/



 /* use the variables  from 'UPC' to 'UPT  - ( UPT -
TopLayerUnits_5in ) as     Units_Remove'  for 5-inch to 2-inch
reduction
 list, casued by removing excess units from top layer where the
product is packed in a 5 inch tray and changing to x complete layers
of
 a tray with a 2-inch bar height*/

  UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
 ,Max_UPL_PERLAYER_5in
,Total_Layers_used_5in
,Full_Layers_used_5in
,Excess_TLayUnits_5in
,TopLayerUnits_5in
,Max_UPT_Poss_5in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
```

```
,Fin_Bar_Desig
, UPT
, ( UPT - TopLayerUnits_5in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_5in ) as     Units_Remove


from  (

SELECT *

 ,Case when Bar_Desig_2in = 'Yes' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'Yes' then '2'
       when Bar_Desig_2in = 'No' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'Yes' then '5'
        when Bar_Desig_2in = 'No' AND   Bar_Desig_5in =  'No' AND
Bar_Desig_7in =  'Yes' then '7'
        when Bar_Desig_2in = 'Yes' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'No' then '2'
        when Bar_Desig_2in = 'Yes' AND   Bar_Desig_5in =  'No' AND
Bar_Desig_7in =  'No' then '2'
        when Bar_Desig_2in = 'No' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'No' then '5'
End as Fin_Bar_Desig   /* Final Bar Height Designation for the
Product with its  given dimensions and UPT values. THere are 6
possible possibilites.
*/

FROM (

 Select UPC, Display_width_mm , Display_depth_mm  ,
Display_Height_mm, UPT

/* calc 1 */

,  Floor ( 94.83 / ((Display_width_mm * Display_depth_mm)/1000)  )
as  Max_UPL_PERLAYER_2in   /* Maximum Units Per Layer possible
assuming number of units  = UPT for 2-inch bar height
*/
, Floor ( 94.83    / ((Display_width_mm * Display_depth_mm)/1000) )
as Max_UPL_PERLAYER_5in  /* Maximum Units Per Layer possible
assuming number of units  = UPT for 5-inch bar height
*/

/* calc 2 */

 ,Total_Layers_used_2in
,Total_Layers_used_5in
,Total_Layers_used_7in


/*  (   =IF($N$12>ROUNDUP(I16*I15,0),0,$N$12-(I15*I21))   */
```

```
 ,Floor ( UPT / (Floor ( 94.83 / ( (Display_width_mm *
Display_depth_mm)/1000)  )  )  ) as Full_Layers_used_2in  /*
complete Layers used in 2 inch bar height
*/
,Floor ( UPT / (Floor ( 94.83 / ( (Display_width_mm *
Display_depth_mm)/1000)  )  ) ) as Full_Layers_used_5in  /*
complete Layers used in 5 inch bar height
*/
,Floor ( UPT / (Floor ( 94.83 / ( (Display_width_mm *
Display_depth_mm)/1000)  )  ) ) as Full_Layers_used_7in  /*
complete Layers used in 7 inch bar height
*/


,   Layers_used_7in
,Layers_used_5in
,   cond1_5in
,   cond2_5in
, Raw_cond2_5in
,nraw_Layers_used_5in

/* Calc 3    */

/* =IF($N$12 >  ROUNDUP(B16*B15,0),   0, $N$12-(B15*B16) )  */

,case when UPT   >  Ceil( (Total_Layers_used_2in *  Floor ( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else UPT  -((Total_Layers_used_2in *  Floor ( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) )) End as
Excess_TLayUnits_2in /*  Number of Excess Units for 2-inch bar
height when top layer is incomplete
*/

,case when UPT   >  Ceil( (Total_Layers_used_5in *  Floor ( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else UPT  -((Total_Layers_used_5in *  Floor ( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) )) End as
Excess_TLayUnits_5in /*  Number of Excess Units for 5-inch bar
height when top layer is incomplete
*/


,case when UPT   >  Ceil( (Total_Layers_used_7in *  Floor ( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else UPT  -((Total_Layers_used_7in *  Floor ( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) )) End as
Excess_TLayUnits_7in /*  Number of Excess Units for 7-inch bar
height when top layer is incomplete
*/
```

```
/* calc 4 */

,case when UPT   >  Ceil( (Total_Layers_used_2in *  Floor( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else  UPT - ( Floor ( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Floor ( UPT / (Floor ( 94.83 / (
(Display_width_mm * Display_depth_mm)/1000)  )  )  ))
End as  TopLayerUnits_2in /*  Exact Number of Units on top layer
when packed in a 2-inch bar height       */

,case when UPT   >  Ceil( (Total_Layers_used_5in *  Floor( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else  UPT - ( Floor ( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Floor ( UPT / (Floor ( 94.83 / (
(Display_width_mm * Display_depth_mm)/1000)  )  )  ))
End as  TopLayerUnits_5in  /* Exact Number of Units on top layer
when packed in a 5-inch bar height
*/


,case when UPT   >  Ceil( (Total_Layers_used_7in *  Floor( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else  UPT - ( Floor ( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Floor ( UPT / (Floor ( 94.83 / (
(Display_width_mm * Display_depth_mm)/1000)  )  )  ))
End as  TopLayerUnits_7in  /* Exact Number of Units on top layer
when packed in a 7-inch bar height
*/


/* calc 5 */


,  Floor ( 94.83 / ((Display_width_mm * Display_depth_mm)/1000)  ) *
Total_Layers_used_2in   as  Max_UPT_Poss_2in   /*  Maximum UPT value
possibel for 2-inch Bar Height
*/
, Floor ( 94.83    / ((Display_width_mm * Display_depth_mm)/1000) )
* Total_Layers_used_5in as  Max_UPT_Poss_5in  /*    Maximum UPT
value possibel for 5-inch Bar Height
*/
, Floor ( 94.83   / ((Display_width_mm * Display_depth_mm)/1000)  )
* Total_Layers_used_7in as  Max_UPT_Poss_7in  /*    Maximum UPT
value possibel for 7-inch Bar Height
*/


/* calc 6 */
```

```
,Case when UPT >  Floor ( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Total_Layers_used_2in
then 'No' ELSE 'Yes' End as Bar_Desig_2in    /*    2-inch Bar Height
Designation (yes or No)
*/

,Case when UPT >  FLoor ( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Total_Layers_used_5in
then 'No' ELSE 'Yes' End as Bar_Desig_5in   /*   5-inch Bar Height
Designation (yes or No)
*/

,Case when UPT >  Floor ( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Total_Layers_used_7in
then 'No' ELSE 'Yes' End as Bar_Desig_7in  /*  7-inch Bar Height
Designation (yes or No)
*/

,ARTICLE_DESCRIPTION
,Display_Orientation

from (

Select *
, CEIL( 94.83 / ((Display_width_mm * Display_depth_mm)/1000)  ) as
MaxUPLperLayer   /*  Maximum Number of Units per complete layer
*/
, Floor(  UPT /  Ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  )   ) as  FullLayers   /*  Number of
complete Layers
*/


,case when Floor(  UPT /  Ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  )   ) > 0
then UPT  - Floor( Ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  Floor(  UPT /  Ceil( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) )
when Floor(  UPT /  Ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  )   ) < 1
then  UPT  - ceil( ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  ceil(  UPT /  Ceil( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) )
End as Final_ExcessTopLayerUnits
/*
*/

, UPT  - Floor( Ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  Floor(  UPT /  Ceil( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) ) as
```

```
ExcessTopLayerUnits    /*
*/
, UPT  - ceil( ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  ceil(  UPT /  Ceil( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) ) as
NewExcessTopLayerUnits   /*   Number of Units that are on the top
layer, if it is an incomplete layer
*/

, ( UPT  - Floor( Ceil( 94.83 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  Floor(  UPT /  Floor( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) )  )
/   CEIL( 94.83 / ((Display_width_mm * Display_depth_mm)/1000)  )
as  Layerunits_Perc_excess   /* % of Layer Units that are on the top
layer, if it is an incomplete layer
*/


/*
,  4836.483 /  (   ((Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000 )  * UPT )
, Floor( 51 / ( Display_Height_mm ) )   *  Floor(94.83 /
((Display_width_mm * Display_depth_mm)/1000) )  as B8
, Floor( 51 / ( Display_Height_mm ) )     *     94.83 /
(Display_width_mm * Display_depth_mm)/1000   as B9
, Floor(94.83 / ((Display_width_mm * Display_depth_mm)/1000) ) as
B10        */

, case when  UPT  > Layers_used_2in then cond1_2in  Else cond2_2in
End as  Total_Layers_used_2in  /* For given UPT, the final numnber
of layers used for a 2 inch bar height
*/
, case when  UPT  > Layers_used_5in then cond1_5in  Else  cond2_5in
End as  Total_Layers_used_5in /*   For given UPT, the final numnber
of layers used for a 5 inch bar height
*/
, case when  UPT  > Layers_used_7in then cond1_7in  Else cond2_7in
End as  Total_Layers_used_7in  /*  For given UPT, the final numnber
of layers used for a 7 inch bar height
*/


,  Layers_used_7in
,Layers_used_5in
,   cond1_5in
,   cond2_5in
, Raw_cond2_5in
,nraw_Layers_used_5in

from (
```

```
Select *

, Floor (4836.483   / ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT )    ) as  Max_Layers_2in    /*
Maximum number of layers for a 2 inch bar height
*/
, CEIL (    ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT )    /  9957.465  ) as
Max_Layers_5in /*   Maximum number of layers for a 5 inch bar height
*/
, CEIL (14604.282   / ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT )    ) as  Max_Layers_7in   /*
Maximum number of layers for a 7 inch bar height
*/


,Display_width_mm
,Display_depth_mm

,ARTICLE_DESCRIPTION
,Display_Orientation


,   case when ( 4836.483 /  (   ((Display_width_mm *
Display_depth_mm  *  Display_Height_mm )/1000 )  * UPT )  < 1
AND ( Display_Height_mm > 51   )  )
 then  0
ELSE  Floor(   (51 / ( Display_Height_mm ) )   *  Floor( 51 / (
Display_Height_mm ) ) )  * ( Floor(94.83 / ((Display_width_mm *
Display_depth_mm)/1000) ) )   END as Layers_used_2in  /* third
possibility for Number of actual  Tray Layers for 2-inch   Bar
Height    */
,  Case when
4836.483   /   ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT ) < 0 then  0
ELSE   Floor (  ( 51 / Display_Height_mm ) )  End as cond1_2in  /*
Second possibility for Number of possible  Tray Layers for 2-inch
Bar Height
*/

,  Ceil ( 94.83 / ((Display_width_mm * Display_depth_mm)/1000)  ) as
cond2_2in /*  first possibility for Number of possible  Tray Layers
for 2-inch   Bar Height
*/



,case when (Floor( 9957.465
 /  (   ((Display_width_mm * Display_depth_mm  *  Display_Height_mm
)/1000 )  * UPT ))  < 1
```

```
AND ( Floor(Display_Height_mm > 105 )  )  )
 then  0
ELSE  (  Floor( 105 / ( Display_Height_mm ))  )    * ( Floor ( 94.83
/ ((Display_width_mm * Display_depth_mm)/1000)  )  )
END as Layers_used_5in  /* second possibility for Number of actual
possible Raw Tray Layers for 5-inch   Bar Height
*/
,  (  Floor( 105 / ( Display_Height_mm ))  )    * ( Floor ( 94.83 /
((Display_width_mm * Display_depth_mm)/1000)  )  )
as nraw_Layers_used_5in  /*  first possibility for Number of
possible Raw Tray Layers for 5-inch   Bar Height
*/



,  Case when
9957.465   /   ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT ) < 0 then  0
ELSE     Floor (  ( 105 / Display_Height_mm ) )  End as cond1_5in
/*     third possibility for Number of possible  Tray Layers for 5-
inch   Bar Height
*/

,  Ceil ( UPT / Floor ( 94.83   / ((Display_width_mm *
Display_depth_mm)/1000)  )  ) as cond2_5in  /*   Second possibility
for Number of possible  Tray Layers for 5-inch   Bar Height
*/

,  UPT / Floor ( 94.83   / ((Display_width_mm *
Display_depth_mm)/1000)  )   as Raw_cond2_5in      /*   first
possibility for Number of possible  Tray Layers for 5-inch   Bar
Height
*/



,case when ( 14604.282
 /  (   ((Display_width_mm * Display_depth_mm  *  Display_Height_mm
)/1000 )  * UPT )  < 1
AND ( Display_Height_mm > 154   )  )
 then  0
ELSE  Floor( 154 / ( Display_Height_mm ) )   *  Floor( 105 / (
Display_Height_mm ) )   *  Floor(94.83 / ((Display_width_mm *
Display_depth_mm)/1000) )
END as Layers_used_7in  /*  Third possibility for number of possible
Tray Layers for 7-inch   Bar Height
*/
```

```
,  Case when   14604.282   /   ( ( (Display_width_mm *
Display_depth_mm  *  Display_Height_mm )/1000)  *  UPT ) < 0 then  0
ELSE   Floor (  ( 154 / Display_Height_mm ) )  End as cond1_7in  /*
Second possibility for number of possible  Tray Layers for 7-inch
Bar Height
*/

/* ,  Ceil ( 94.83 / ((Display_width_mm * Display_depth_mm)/1000)  )
as cond2_7in  */
,  Ceil ( UPT / Floor ( 94.83   / ((Display_width_mm *
Display_depth_mm)/1000)  )  ) as cond2_7in  /* first possibility for
Number of possible  Tray Layers for 7-inch   Bar
Height
*/


From (

Select *
,Display_Height_mm -  Metric_7_to5   as  Metric_7_to5_drop   /*
Difference between Product height and of minumum height of product
fitting in the 5-inch bar height
*/
,Display_Height_mm -  Metric_5_to2   as  Metric_5_to2_drop  /*
Difference between Product height and of minumum height of product
fitting in the 2-inch bar height
*/
,Display_Height_mm
,Display_width_mm
,Display_depth_mm

from (

Select *,
case when Layers_Metric_TRAY_7 > 0  then      floor(
(Display_Height_mm * 9957.465)/(Cub_TotalProdUPTVol) )  END as
Metric_7_to5     /*   SImple Proportion calculation to determine the
new product height required for maximum UPT fora Bar Height
reduction from a 7-inch tray to a 5-inch tray. The UPC height is
multiplied by the Volume (ml) occupied  for  Half_TRAY 5 inch bar
height and then divided by the  Total cuboid volume for product when
number of units = UPT   */
, case when Layers_Metric_TRAY_5 > 0  then      floor(
(Display_Height_mm * 4836.483)/(Cub_TotalProdUPTVol)  )  END as
Metric_5_to2  /*  SImple Proportion calculation to determine the new
product height required for maximum UPT for a Bar Height reduction
from a 5-inch tray to a 2-inch tray. The UPC height is multiplied by
the Volume (ml) occupied  for  Half_TRAY 2 inch bar height and then
divided by the  Total cuboid volume for product when number of units
= UPT    */
```

```
 , Display_Height_mm

from (

Select *

,case when EQUIPMENT_TYPE =  'HALF TRAY' then floor(11466.414 /
Cub_TotalProdUPTVol)  end as Layers_HALF_TRAY_2  /* For Half tray 2-
inch Bar height number of layers occupied by product for UPT
*/
,case when EQUIPMENT_TYPE = 'HALF TRAY' then     floor(22155.444
/Cub_TotalProdUPTVol) end as Layers_HALF_TRAY_5  /* For Half tray 5-
inch Bar height number of layers occupied by product for UPT
*/
,case when EQUIPMENT_TYPE = 'HALF TRAY' then     floor(33427.512
/Cub_TotalProdUPTVol) end as Layers_HALF_TRAY_7  /* For Half tray 7-
inch Bar height number of layers occupied by product for UPT
*/


,case when EQUIPMENT_TYPE = 'METRIC TRAY' then   floor(11466.414
/Cub_TotalProdUPTVol)      end as Layers_METRIC_TRAY_2   /* For
Metric tray 2-inch Bar height number of layers occupied by product
for UPT         */
,case when EQUIPMENT_TYPE = 'METRIC TRAY' then   floor(22155.444
/Cub_TotalProdUPTVol) end as Layers_METRIC_TRAY_5  /* For Metric
tray 5-inch Bar height number of layers occupied by product for UPT
*/
,case when EQUIPMENT_TYPE = 'METRIC TRAY' then   floor(33427.592
/Cub_TotalProdUPTVol) end as Layers_METRIC_TRAY_7  /* For Metric
tray 7-inch Bar height number of layers occupied by product for UPT
*/

from

(
Select ud.*
,upt.AREA
,upt.CATEGORY_CODE
,upt.CATEGORY_NAME
,ud.Display_Orientation
,upt.ARTICLE_NO
,upt.ARTICLE_DESCRIPTION
,upt.EQUIPMENT_TYPE
,upt.UPT
, (upt.UPT  * cuboid_vol_ml)/1000 as Cub_TotalProdUPTVol    /*
Total cuboid volume for product when number of units = UPT  */
, (upt.UPT  * cylinder_vol_ml)/1000 as cyl_TotalProdUPTVol  /*
Total cylindrical volume for product when number of units = UPT  */

from UOD.UOD_DIMVOL as ud
```

```
Inner Join UOD.UOD_UPT2 as upt on ud.UPC  = upt.UPC
    where          upt.EQUIPMENT_TYPE = 'HALF TRAY'

) as t

) as u

) as v

) as x

) as y

) as z

) as a

) as b

  /*  where Fin_Bar_Desig = ''     */


/* where (  Fin_Bar_Desig = '7'    AND  TopLayerUnits_7in  > 0
)
AND  (  (UPT  - ( UPT - TopLayerUnits_7in ))    < UPT     )  AND
Excess_TLayUnits_7in <> 0   */


   where (  Fin_Bar_Desig = '5'    AND  TopLayerUnits_5in  > 0
)
AND  (  (UPT  - ( UPT - TopLayerUnits_5in ))    < UPT     )  AND
Excess_TLayUnits_5in <> 0


  order by   Fin_Bar_Desig desc, TopLayerUnits_7in,
TopLayerUnits_5in, TopLayerUnits_2in
;
run;




/* METRIC METRIC  METRIC  METRIC  METRIC  METRIC  METRIC  METRIC
METRIC  METRIC
```

```
METRIC  METRIC  METRIC  METRIC  METRIC  METRIC  METRIC  METRIC
METRIC  METRIC   */
/* For Metric trays identify UPCs with Exess or surefiet of Units
per tray  */



/* Constants:

11466.414 /  Volume (ml) occupied  for  METRIC_TRAY 2 inch bar
height
22155.444 /  Volume (ml) occupied  for  METRIC_TRAY 5 inch bar
height
33427.512 /  Volume (ml) occupied  for  METRIC_TRAY 7 inch bar
height


4836.483  / Volume (ml) occupied  for  Half_TRAY 2 inch bar height
9957.465  / Volume (ml) occupied  for  Half_TRAY 5 inch bar height
14604.282 / Volume (ml) occupied  for  Half_TRAY 7 inch bar height


 194.346  / Area (mm square) of internal base of METRIC_TRAY

 59   / Height (mm) of METRIC_TRAY with 2 Inch Bar Height
 114  / Height (mm) of METRIC_TRAY with 5 Inch Bar Height
 172  / Height (mm) of METRIC_TRAY with 7 Inch Bar Height

*/



proc sql ;

 create table UOD.Metric_UOD_HYBRID as
/* create table UOD.Metric5_UOD_HYBRID as */


 /* use the variables  from 'UPC' to 'UPT  - ( UPT -
TopLayerUnits_7in ) as     Units_Remove'  for 7-inch to 5-inch
reduction
 list, casued by removing excess units from top layer where the
product is packed in a 7 inch tray and changing to x complete layers
of
 a tray with a 5-inch bar height*/

 Select
 /* UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
```

```
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
 ,Max_UPL_PERLAYER_7in
,Total_Layers_used_7in
,Full_Layers_used_7in
,Excess_TLayUnits_7in
,TopLayerUnits_7in
,Max_UPT_Poss_7in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
,Fin_Bar_Desig
, UPT
, ( UPT - TopLayerUnits_7in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_7in ) as     Units_Remove
*/


  UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
 ,Max_UPL_PERLAYER_5in
,Total_Layers_used_5in
,Full_Layers_used_5in
,Excess_TLayUnits_5in
,TopLayerUnits_5in
,Max_UPT_Poss_5in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
,Fin_Bar_Desig
, UPT
, ( UPT - TopLayerUnits_5in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_5in ) as     Units_Remove

 /* use the variables  from 'UPC' to 'UPT  - ( UPT -
TopLayerUnits_5in ) as     Units_Remove'  for 5-inch to 2-inch
reduction
 list, casued by removing excess units from top layer where the
product is packed in a 5 inch tray and changing to x complete layers
of
 a tray with a 2-inch bar height*/


from  (

SELECT *
```

```
 ,Case when Bar_Desig_2in = 'Yes' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'Yes' then '2'
       when Bar_Desig_2in = 'No' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'Yes' then '5'
        when Bar_Desig_2in = 'No' AND   Bar_Desig_5in =  'No' AND
Bar_Desig_7in =  'Yes' then '7'
        when Bar_Desig_2in = 'Yes' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'No' then '2'
        when Bar_Desig_2in = 'Yes' AND   Bar_Desig_5in =  'No' AND
Bar_Desig_7in =  'No' then '2'
        when Bar_Desig_2in = 'No' AND   Bar_Desig_5in =  'Yes' AND
Bar_Desig_7in =  'No' then '5'
End as Fin_Bar_Desig  /* Final Bar Height Designation for the
Product with its  given dimensions and UPT values. THere are 6
possible possibilites.
*/

FROM (

 Select UPC, Display_width_mm , Display_depth_mm  ,
Display_Height_mm, UPT

/* calc 1 */

,  Floor ( 194.346 / ((Display_width_mm * Display_depth_mm)/1000)  )
as  Max_UPL_PERLAYER_2in  /* Maximum Units Per Layer possible
assuming number of units  = UPT for 2-inch bar height
*/
, Floor ( 194.346    / ((Display_width_mm * Display_depth_mm)/1000)
)            as Max_UPL_PERLAYER_5in /* Maximum Units Per Layer
possible assuming number of units  = UPT for 5-inch bar height
*/
, Floor ( 194.346   / ((Display_width_mm * Display_depth_mm)/1000)
)            as  Max_UPL_PERLAYER_7in /* Maximum Units Per Layer
possible assuming number of units  = UPT for 7-inch bar height
*/

/* calc 2 */

 ,Total_Layers_used_2in
,Total_Layers_used_5in
,Total_Layers_used_7in

/*  (   =IF($N$12>ROUNDUP(I16*I15,0),0,$N$12-(I15*I21))   */

 ,Floor ( UPT / (Floor ( 194.346 / ( (Display_width_mm *
Display_depth_mm)/1000)  )  )  )  as Full_Layers_used_2in  /*
complete Layers used in 2 inch bar height
*/
```

```
,Floor ( UPT / (Floor ( 194.346 / ( (Display_width_mm *
Display_depth_mm)/1000)  )  ) ) as Full_Layers_used_5in    /*
complete Layers used in 5 inch bar height
*/
,Floor ( UPT / (Floor ( 194.346 / ( (Display_width_mm *
Display_depth_mm)/1000)  )  ) ) as Full_Layers_used_7in   /*
complete Layers used in 7 inch bar
height
*/

,   Layers_used_7in
,Layers_used_5in
,   cond1_5in
,   cond2_5in
, Raw_cond2_5in
,nraw_Layers_used_5in


/* Calc 3    */

/* =IF($N$12 >  ROUNDUP(B16*B15,0),   0, $N$12-(B15*B16) )  */

,case when UPT   >  Ceil( (Total_Layers_used_2in *  Floor ( 194.346
/ ((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else UPT  -((Total_Layers_used_2in *  Floor ( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  ) )) End as
Excess_TLayUnits_2in  /*  Number of Excess Units for 2-inch bar
height when top layer is incomplete
*/

,case when UPT   >  Ceil( (Total_Layers_used_5in *  Floor ( 194.346
/ ((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else UPT  -((Total_Layers_used_5in *  Floor ( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  ) )) End as
Excess_TLayUnits_5in /*  Number of Excess Units for 5-inch bar
height when top layer is incomplete
*/


,case when UPT   >  Ceil( (Total_Layers_used_7in *  Floor ( 194.346
/ ((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else UPT  -((Total_Layers_used_7in *  Floor ( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  ) )) End as
Excess_TLayUnits_7in  /*  Number of Excess Units for 7-inch bar
height when top layer is incomplete
*/


/* calc 4 */
```

```
,case when UPT   >  Ceil( (Total_Layers_used_2in *  Floor( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else  UPT - ( Floor ( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Floor ( UPT / (Floor ( 194.346 / (
(Display_width_mm * Display_depth_mm)/1000)  )  ))
End as  TopLayerUnits_2in /*  Exact Number of Units on top layer
when packed in a 2-inch bar height       */

,case when UPT   >  Ceil( (Total_Layers_used_5in *  Floor( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else  UPT - ( Floor ( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Floor ( UPT / (Floor ( 194.346 / (
(Display_width_mm * Display_depth_mm)/1000)  )  ))
End as  TopLayerUnits_5in /*  Exact Number of Units on top layer
when packed in a 5-inch bar height       */


,case when UPT   >  Ceil( (Total_Layers_used_7in *  Floor( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  ) ) )
then 0 else  UPT - ( Floor ( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Floor ( UPT / (Floor ( 194.346 / (
(Display_width_mm * Display_depth_mm)/1000)  )  ))
End as  TopLayerUnits_7in  /*  Exact Number of Units on top layer
when packed in a 7-inch bar height       */

/* calc 5 */

,  Floor ( 194.346 / ((Display_width_mm * Display_depth_mm)/1000)  )
* Total_Layers_used_2in   as  Max_UPT_Poss_2in  /*  Maximum UPT
value possibel for 2-inch Bar Height
*/
, Floor ( 194.346    / ((Display_width_mm * Display_depth_mm)/1000)
)  * Total_Layers_used_5in as  Max_UPT_Poss_5in  /*  Maximum UPT
value possibel for 5-inch Bar Height
*/
, Floor ( 194.346   / ((Display_width_mm * Display_depth_mm)/1000)
)  * Total_Layers_used_7in as  Max_UPT_Poss_7in  /*  Maximum UPT
value possibel for 7-inch Bar Height
*/

/* calc 6 */

,Case when UPT >  Floor ( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Total_Layers_used_2in
then 'No' ELSE 'Yes' End as Bar_Desig_2in /*    2-inch Bar Height
Designation (yes or No)
*/

,Case when UPT >  FLoor ( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Total_Layers_used_5in
```

```
then 'No' ELSE 'Yes' End as Bar_Desig_5in /*    5-inch Bar Height
Designation (yes or No)
*/

,Case when UPT >  Floor ( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) * Total_Layers_used_7in
then 'No' ELSE 'Yes' End as Bar_Desig_7in /*    7-inch Bar Height
Designation (yes or No)
*/

,ARTICLE_DESCRIPTION
,Display_Orientation

from (

Select *
, CEIL( 194.346 / ((Display_width_mm * Display_depth_mm)/1000)  ) as
MaxUPLperLayer                 /*  Maximum Number of Units per
complete layer
*/
, Floor(  UPT /  Ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  )   ) as  FullLayers /*  Number of complete
Layers
*/


,case when Floor(  UPT /  Ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  )   ) > 0
then UPT  - Floor( Ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  Floor(  UPT /  Ceil( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) )
when Floor(  UPT /  Ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  )   ) < 1
then  UPT  - ceil( ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  ceil(  UPT /  Ceil( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) )
End as Final_ExcessTopLayerUnits

, UPT  - Floor( Ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  Floor(  UPT /  Ceil( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) ) as
ExcessTopLayerUnits
, UPT  - ceil( ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  ceil(  UPT /  Ceil( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) ) as
NewExcessTopLayerUnits   /*   Number of Units that are on the top
layer, if it is an incomplete layer
*/
```

```
, ( UPT  - Floor( Ceil( 194.346 / ((Display_width_mm *
Display_depth_mm)/1000)  ) *  Floor(  UPT /  Floor( 194.346 /
((Display_width_mm * Display_depth_mm)/1000)  )   ) )  )
/   CEIL( 194.346 / ((Display_width_mm * Display_depth_mm)/1000)  )
as  Layerunits_Perc_excess  /* % of Layer Units that are on the top
layer, if it is an incomplete layer
*/


/*
,  11466.414 /  (   ((Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000 )  * UPT )
, Floor( 59 / ( Display_Height_mm ) )   *  Floor(194.346 /
((Display_width_mm * Display_depth_mm)/1000) )  as B8
, Floor( 59 / ( Display_Height_mm ) )     *     194.346 /
(Display_width_mm * Display_depth_mm)/1000   as B9
, Floor(194.346 / ((Display_width_mm * Display_depth_mm)/1000) ) as
B10  */


, case when  UPT  > Layers_used_2in then cond1_2in  Else cond2_2in
End as  Total_Layers_used_2in    /* For given UPT, the final numnber
of layers used for a 2 inch bar height
*/
, case when  UPT  > Layers_used_5in then cond1_5in  Else  cond2_5in
End as  Total_Layers_used_5in  /* For given UPT, the final numnber
of layers used for a 5 inch bar height
*/
, case when  UPT  > Layers_used_7in then cond1_7in  Else cond2_7in
End as  Total_Layers_used_7in   /* For given UPT, the final numnber
of layers used for a 7 inch bar height
*/


,  Layers_used_7in
,Layers_used_5in
,   cond1_5in
,   cond2_5in
, Raw_cond2_5in
,nraw_Layers_used_5in

from (

Select *

, Floor (11466.414   / ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT )    ) as  Max_Layers_2in    /*
Maximum number of layers for a 2 inch bar height    */
, CEIL (    ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT )    /  22155.444  ) as
```

```
Max_Layers_5in /*  Maximum number of layers for a 5 inch bar height
*/
, CEIL (33427.512   / ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT )    ) as  Max_Layers_7in    /*
Maximum number of layers for a 7 inch bar height    */


,Display_width_mm
,Display_depth_mm

,ARTICLE_DESCRIPTION
,Display_Orientation

/*
$N$12>ROUNDUP(ROUNDDOWN(IF(AND(F8<1,$E$12>F4),0,F4/$E$12),0)*H15,0)
*/

,case when ( 11466.414 /  (   ((Display_width_mm * Display_depth_mm
*  Display_Height_mm )/1000 )  * UPT )  < 1
AND ( Display_Height_mm > 59   )  )   then  0  ELSE  Floor(  (59 / (
Display_Height_mm ) )   *  Floor( 59 / ( Display_Height_mm ) ) )  *
( Floor(194.346 / ((Display_width_mm * Display_depth_mm)/1000) ))
END as Layers_used_2in  /* third possibility for Number of actual
Tray Layers for 2-inch   Bar Height    */
,  Case when
11466.414   /   ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT ) < 0 then  0
ELSE   Floor (  ( 59 / Display_Height_mm ) )  End as cond1_2in   /*
Second possibility for Number of possible  Tray Layers for 2-inch
Bar Height
*/

,  Ceil ( 194.346 / ((Display_width_mm * Display_depth_mm)/1000)  )
as cond2_2in  /*  first possibility for Number of possible  Tray
Layers for 2-inch   Bar Height
*/


,case when (Floor( 22155.444
 /  (   ((Display_width_mm * Display_depth_mm  *  Display_Height_mm
)/1000 )  * UPT ))  < 1
AND ( Floor(Display_Height_mm > 114 )  )  )
 then  0
ELSE  (  Floor( 114 / ( Display_Height_mm ))  )    * ( Floor (
194.346 / ((Display_width_mm * Display_depth_mm)/1000)  )  )
END as Layers_used_5in  /*  first possibility for Number of possible
Raw Tray Layers for 5-inch   Bar Height
*/
```

```
,   (  Floor( 114 / ( Display_Height_mm ))  )     * ( Floor ( 194.346
/ ((Display_width_mm * Display_depth_mm)/1000)  )  )
as nraw_Layers_used_5in

,  Case when
22155.444   /   ( ( (Display_width_mm * Display_depth_mm  *
Display_Height_mm )/1000)  *  UPT ) < 0 then  0
ELSE     Floor (  ( 114 / Display_Height_mm ) )  End as cond1_5in
/*     third possibility for Number of possible  Tray Layers for 5-
inch   Bar Height
*/

,  Ceil ( UPT / Floor ( 194.346   / ((Display_width_mm *
Display_depth_mm)/1000)  )  ) as cond2_5in  /*   Second possibility
for Number of possible  Tray Layers for 5-inch   Bar Height
*/

,  UPT / Floor ( 194.346   / ((Display_width_mm *
Display_depth_mm)/1000)  )   as Raw_cond2_5in  /*   first
possibility for Number of possible  Tray Layers for 5-inch   Bar
Height
*/



,case when ( 33427.512
 /  (   ((Display_width_mm * Display_depth_mm  *  Display_Height_mm
)/1000 )  * UPT )  < 1
AND ( Display_Height_mm > 172   )  )
 then  0
ELSE  Floor( 172 / ( Display_Height_mm ) )   *  Floor( 114 / (
Display_Height_mm ) )   *  Floor(194.346 / ((Display_width_mm *
Display_depth_mm)/1000) )
END as Layers_used_7in  /*  Third possibility for number of possible
Tray Layers for 7-inch   Bar Height
*/


,  Case when   33427.512   /   ( ( (Display_width_mm *
Display_depth_mm  *  Display_Height_mm )/1000)  *  UPT ) < 0 then  0
ELSE   Floor (  ( 172 / Display_Height_mm ) )  End as cond1_7in  /*
Second possibility for number of possible  Tray Layers for 7-inch
Bar Height
*/

/* ,  Ceil ( 194.346 / ((Display_width_mm * Display_depth_mm)/1000)
) as cond2_7in  */
,  Ceil ( UPT / Floor ( 194.346   / ((Display_width_mm *
Display_depth_mm)/1000)  )  ) as cond2_7in    /* first possibility
for Number of possible  Tray Layers for 7-inch   Bar
```

```
Height
*/


From (


Select *
,Display_Height_mm -  Metric_7_to5   as  Metric_7_to5_drop
,Display_Height_mm -  Metric_5_to2   as  Metric_5_to2_drop
,Display_Height_mm
,Display_width_mm
,Display_depth_mm

from (

Select *,
case when Layers_Metric_TRAY_7 > 0  then      floor(
(Display_Height_mm * 22155.444)/(Cub_TotalProdUPTVol) )  END as
Metric_7_to5   /*   SImple Proportion calculation to determine the
new product height required for maximum UPT fora Bar Height
reduction from a 7-inch tray to a 5-inch tray. The UPC height is
multiplied by the Volume (ml) occupied  for  metric 5 inch bar
height and then divided by the  Total cuboid volume for product when
number of units = UPT   */
, case when Layers_Metric_TRAY_5 > 0  then      floor(
(Display_Height_mm * 11466.414)/(Cub_TotalProdUPTVol)  )  END as
Metric_5_to2  /*  SImple Proportion calculation to determine the new
product height required for maximum UPT for a Bar Height reduction
from a 5-inch tray to a 2-inch tray. The UPC height is multiplied by
the Volume (ml) occupied  for  metric 2 inch bar height and then
divided by the  Total cuboid volume for product when number of units
= UPT    */

   , Display_Height_mm

from (

Select *

,case when EQUIPMENT_TYPE =  'HALF TRAY' then floor(11466.414
/Cub_TotalProdUPTVol)      end as Layers_HALF_TRAY_2  /* For Half
tray 2-inch Bar height number of layers occupied by product for UPT
*/
,case when EQUIPMENT_TYPE = 'HALF TRAY' then     floor(22155.444
/Cub_TotalProdUPTVol) end as Layers_HALF_TRAY_5   /* For Half tray
5-inch Bar height number of layers occupied by product for UPT
*/
,case when EQUIPMENT_TYPE = 'HALF TRAY' then     floor(33427.512
/Cub_TotalProdUPTVol) end as Layers_HALF_TRAY_7    /* For Half tray
```

```
7-inch Bar height number of layers occupied by product for UPT
*/


,case when EQUIPMENT_TYPE = 'METRIC TRAY' then   floor(11466.414
/Cub_TotalProdUPTVol)      end as Layers_METRIC_TRAY_2   /* For
Metric tray 2-inch Bar height number of layers occupied by product
for UPT         */
,case when EQUIPMENT_TYPE = 'METRIC TRAY' then   floor(22155.444
/Cub_TotalProdUPTVol) end as Layers_METRIC_TRAY_5   /* For Metric
tray 5-inch Bar height number of layers occupied by product for UPT
*/
,case when EQUIPMENT_TYPE = 'METRIC TRAY' then   floor(33427.592
/Cub_TotalProdUPTVol) end as Layers_METRIC_TRAY_7   /* For Metric
tray 7-inch Bar height number of layers occupied by product for UPT
*/


from

(
Select ud.*
,upt.AREA
,upt.CATEGORY_CODE
,upt.CATEGORY_NAME
,ud.Display_Orientation
,upt.ARTICLE_NO
,upt.ARTICLE_DESCRIPTION
,upt.EQUIPMENT_TYPE
,upt.UPT
, (upt.UPT  * cuboid_vol_ml)/1000 as Cub_TotalProdUPTVol   /*  Total
cuboid volume for product when number of units = UPT  */
, (upt.UPT  * cylinder_vol_ml)/1000 as cyl_TotalProdUPTVol  /*
Total cylindrical volume for product when number of units = UPT  */


/* use table for wrongly assigned Half tray products */
from UOD.NewMetricUod as nmu
Inner Join  UOD.UOD_DIMVOL as ud on ud.UPC  = nmu.UPC
      Inner Join UOD.UOD_UPT2 as upt on ud.UPC  = upt.UPC

       /*
from UOD.UOD_DIMVOL as ud
Inner Join UOD.UOD_UPT2 as upt on ud.UPC  = upt.UPC
    where         upt.EQUIPMENT_TYPE = 'METRIC TRAY'    */

) as t

) as u

) as v
```

```
) as x

) as y

) as z

) as a

) as b


   where

  /*  Toggle this code for the UPT reduction that cause  a change
from a  7 inch  to a 5 inch  bar height                      */
 /* (  Fin_Bar_Desig = '7'    AND  TopLayerUnits_7in  > 0        )
AND  (  (UPT  - ( UPT - TopLayerUnits_7in ))    < UPT     )  AND
Excess_TLayUnits_7in <> 0    */


   /*  Toggle this code for the UPT reduction that cause  a change
from a  5 inch  to a 2 inch  bar height                      */
 (  Fin_Bar_Desig = '5'    AND  TopLayerUnits_5in  > 0        )
AND  (  (UPT  - ( UPT - TopLayerUnits_5in ))    < UPT     )  AND
Excess_TLayUnits_5in <> 0


  order by   Fin_Bar_Desig desc, TopLayerUnits_7in,
TopLayerUnits_5in, TopLayerUnits_2in
;
run;
```

/********** End of SAs Program **********/

4. Excel presentation of final result

4.1. Final Bar Height Reduction Lists

After Stage 1 of the code has been run then the input datasets will be in place. Stage 2 Code (for Half trays) is essentially one very large block of SQL code that is structured with many nested sub-queries. There are many calculations, which were derived in order to 1. Identify those products that have loose units on their top layer within the Half-tray and 2. For this sub-set products, calculate the number of units that can be removed from the top layer in order to pack this product into a lower bar height from either a 7-inch to 5-inch and 5-inch to 2-inch. Each constant and all calculations have been explained in general terms. Most of the calculations are concerned with (assuming the product is packed at its maximum UPT) the number of layers, the number of complete layers, the number of units packed in a layer, all of which leads us to the number of units on the top layer that can be removed in order to pack into a lower bar height. The calculations take into account both internal volumes (volume of Product per unit and total volume at max UPT, and volumes of the tray for 2-inch, 5-inch, and 7-inch), and crucially the squared internal area of the tray. Both volume and area are required to derive the best possible recommendation list since a product with x number of units may fit inside the total volume at a particular bar height but the area of the tray may not permit x units.

Stages 2 and 3 from the UOD_1b.sas' program will each generate two datasets.

Stage 2 will generate 2 separate datasets for the Half-tray tray optimization, of which are stored at: S:\Supply Chain Sharepoint Site\Food Central Support\Data Team\AC\AC Store\Handover\Uod\Uod 1b\Final Sent Datasets

The first file ('HALF_tray_5in_Excess.xlsx') stores the half-tray optimization list where a change from 5-inch to a 2-inch packing is identified and is generated by the stage 2 code in the SAS program, and the second file is ('HALF_tray_7in_Excess.xlsx) stores the half-tray optimization list where a change from 7-inch to a 5-inch packing is identified:


HALF_tray_5in_Excess.xlsx


HALF_tray_7in_Excess.xlsx

All these output files have a cover sheet with all the columns defined, however they are also listed below:

Data Dictionary of Fields:

UPC	Product code
ARTICLE_DESCRIPTION	Product name
DISPLAY_ORIENTATION	The orientation describing how the product is physically packed in the trays. Assumes that height dimension of product is orientated by height in the tray

DISPLAY_WIDTH_mm	Product width in millimetres
DISPLAY_DEPTH_mm	Product Depth in millimetres
DISPLAY_HEIGHT_mm	Product Height in millimetres
Max_UPL_PERLAYER_5in	Maximum number of Units or UPCs that can be packed in 1 single layer
Total_Layers_used_5in	Total Number of Layers used in the for a Half tray with bar height 5 inch
Full_Layers_used_5in	Number of Full Layers used for a Half tray with bar height 5 inch
Excess_TLayUnits_5in	In top layer, number of Units or Products space available (ie, number of units under utilized on top layer)
TopLayerUnits_5in	Number of UPCs packed on top layer
Max_UPT_Poss_5in	Maximum available UPT Possible in the Half tray 5 inch bar height
Bar_Desig_2in	Flag: Recommended Bar Height for 2 Inch Bar Height (Yes; No)
Bar_Desig_5in	Flag: Recommended Bar Height for 5 Inch Bar Height (Yes; No)
Bar_Desig_7in	Flag: Recommended Bar Height for 7 Inch Bar Height (Yes; No)
Fin_Bar_Desig	Final Bar Height Designation (2,5, or 7)
UPT	Units Per Tray value for the UPC . The UPT is taken from the 'FLIC logistics database, and the UPT is calculated but the UoD Model process'
New_UPT_for2In_Bar	Recommended new UPT for the lower bar height of 2 inches
Units_Remove	Number of units or UPCs to remove from top layer to fit in lower bar height

The data for the **('HALF_tray_5in_Excess.xlsx')** list is generated by running the following code from the top Select clause of the query in Stage 2 of the SAS program:

```
/* ************************************************************************** */

UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
 ,Max_UPL_PERLAYER_5in
,Total_Layers_used_5in
,Full_Layers_used_5in
,Excess_TLayUnits_5in
```

```
,TopLayerUnits_5in
,Max_UPT_Poss_5in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
,Fin_Bar_Desig
, UPT
, ( UPT - TopLayerUnits_5in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_5in ) as     Units_Remove

/* ********************************************************************** */
```

In addition, the very last 'where' clause at the bottom of the query will use the following filter to restrict the list to products with a 5-inch to 2-inch bar height change:

```
/* ********************************************************************** */

   where (  Fin_Bar_Desig = '5'    AND  TopLayerUnits_5in  > 0
)
AND  (  (UPT  - ( UPT - TopLayerUnits_5in ))    < UPT     )
AND Excess_TLayUnits_5in <> 0

/* ********************************************************************** */
```

The data for the **('HALF_tray_7in_Excess.xlsx')** list is generated by running the following code from the top Select clause of the query in Stage 2 of the SAS program:

```
/* ********************************************************************** */

UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
 ,Max_UPL_PERLAYER_7in
,Total_Layers_used_7in
,Full_Layers_used_7in
,Excess_TLayUnits_7in
,TopLayerUnits_7in
,Max_UPT_Poss_7in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
,Fin_Bar_Desig
, UPT
, ( UPT - TopLayerUnits_7in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_7in ) as     Units_Remove

/* ********************************************************************** */
```

In addition, the very last 'where' clause at the bottom of the query will use the following filter to restrict the list to products with a 7-inch to 5-inch bar height change:

```
/* ------------------------------------------------------------------------ */

   where (  Fin_Bar_Desig = '7'    AND  TopLayerUnits_7in  > 0
)
AND  (  (UPT  - ( UPT - TopLayerUnits_7in ))    < UPT     )  AND
Excess_TLayUnits_7in <> 0

/* ------------------------------------------------------------------------ */
```

Stage 3

Stage 3 will generate 2 separate datasets for the Metric tray optimization list, of which are stored at: S:\Supply Chain Sharepoint Site\Food Central Support\Data Team\AC\AC Store\Handover\Uod\Uod 1b\Final Sent Datasets

The first file ('METRIC_tray_5in_Excess.xlsx) stores the metric-tray optimization list where a change from 5-inch to a 2-inch packing is identified and is generated by the stage 2 code in the SAS program, and the second file is ('METRIC_tray_7in_Excess.xlsx') which stores the metric -tray optimization list where a change from 7-inch to a 5-inch packing is identified:

The metric-tray output files are arranged in exactly the same way as the Half tray output file, so no further explanation is required as it has been explained. However, the data for the ('Metric_tray_5in_Excess.xlsx') list is generated by running the following code from the top Select clause of the query in Stage 2 of the SAS program:

The data for the **('Metric_tray_5in_Excess.xlsx')** list is generated by running the following code from the top Select clause of the query in Stage 2 of the SAS program:

```
/* ------------------------------------------------------------------------ */

UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
```

```
 ,Max_UPL_PERLAYER_5in
,Total_Layers_used_5in
,Full_Layers_used_5in
,Excess_TLayUnits_5in
,TopLayerUnits_5in
,Max_UPT_Poss_5in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
,Fin_Bar_Desig
, UPT
, ( UPT - TopLayerUnits_5in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_5in ) as     Units_Remove

/* ************************************************************************** */
```

In addition, the very last 'where' clause at the bottom of the query will use the following filter to restrict the list to products with a 5-inch to 2-inch bar height change:

```
/* ************************************************************************** */

   where (  Fin_Bar_Desig = '5'    AND  TopLayerUnits_5in  > 0
)
AND  (  (UPT  - ( UPT - TopLayerUnits_5in ))    < UPT     )
AND Excess_TLayUnits_5in <> 0

/* ************************************************************************** */
```

The data for the **('HALF_tray_7in_Excess.xlsx')** list is generated by running the following code from the top Select clause of the query in Stage 2 of the SAS program:

```
/* ************************************************************************** */

UPC
,ARTICLE_DESCRIPTION
,DISPLAY_ORIENTATION
,DISPLAY_WIDTH_mm
,DISPLAY_DEPTH_mm
,DISPLAY_HEIGHT_mm
 ,Max_UPL_PERLAYER_7in
,Total_Layers_used_7in
,Full_Layers_used_7in
,Excess_TLayUnits_7in
,TopLayerUnits_7in
,Max_UPT_Poss_7in
,Bar_Desig_2in
,Bar_Desig_5in
,Bar_Desig_7in
,Fin_Bar_Desig
, UPT
```

```
, ( UPT - TopLayerUnits_7in ) as New_UPT_for5In_Bar
, UPT  - ( UPT - TopLayerUnits_7in ) as     Units_Remove

/* ************************************************************************** */
```

In addition, the very last 'where' clause at the bottom of the query will use the following filter to restrict the list to products with a 7-inch to 5-inch bar height change:

```
/* ************************************************************************** */

   where (  Fin_Bar_Desig = '7'    AND  TopLayerUnits_7in  > 0
)
AND  (  (UPT  - ( UPT - TopLayerUnits_7in ))    < UPT     )  AND
Excess_TLayUnits_7in <> 0

/* ************************************************************************** */
```

4.2 Tray Optimization Calculator

The use of SAS or any other SQL orientated tool facilitates a higher level of calculations for many UPCs and can process large datasets more efficiently. However, the basic calculations were worked out using Excel for single UPCS and then scaled up in SAS to enable calculations for multiple UPCs. The Excel tool was developed to help demonstrate the tool for end-users, and this tool is available at:

S:\Supply Chain Sharepoint Site\Food Central Support\Data Team\AC\AC Store\Handover\Uod\Uod 1b\ UoD 1b Tray Optimization Calculator .xlsx

The layout is very simple: tray dimensions millimetres (fixed), Product Dimensions, Layers, and Bar Height Reduction. Essentially, you can insert the product dimensions and the product Tray assignment (in Red) for any given product in the cells coloured yellow and then the remaining cells that are formula driven will change.

The 'Tray Optimizaion Output' output box at the bottom of the tool will output all the output parameters based on the tray dimensions that are entered. The example in the current tool is for 'Albacore tuna' (UPC: 998376). Based on the fact that it had been designated as a Half-Tray product (Source: FLIC database extract) then all of the packaging parameters for all 6 tray and bar height combinations are calculated. The output is shown in the table below. Now, the half-try column with the ''Half-tray_5' is the operative columns for this product, because the 'Bar_Desig' row flags the possible bar heights with the current UPT (4) and in this case both the 7-inch and 5-inch half-tray's have a "Yes2 in the 'Bar_Desig' row, however the 2-inch column has a "no" indicating that with a UPT of 4 this product cannot fit into a half-tray with a 2-inch bar height.

Tray Optimization Output

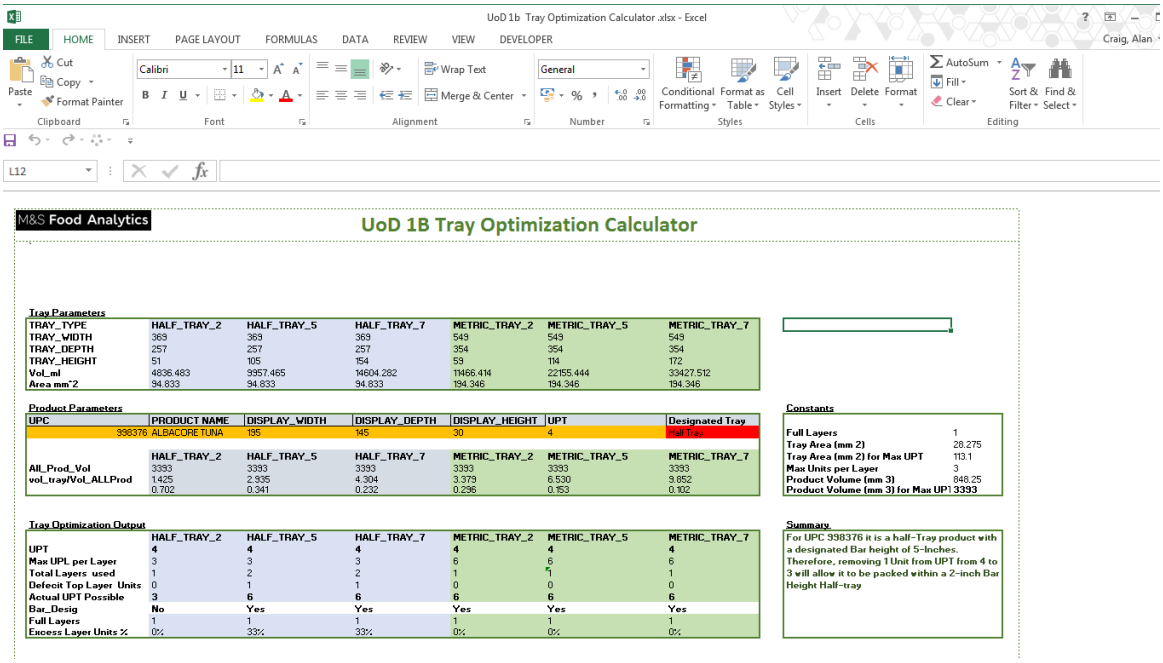
	HALF_TRAY_2	HALF_TRAY_5	HALF_TRAY_7	METRIC_TRAY_2	METRIC_TRAY_5	METRIC_TRAY_7
UPT	**4**	**4**	**4**	**4**	**4**	**4**
Max UPL per Layer	3	3	3	6	6	6
Total Layers used	1	2	2	1	1	1
Deficit Top Layer Units	0	1	1	0	0	0
Actual UPT Possible	**3**	**6**	**6**	**6**	**6**	**6**
Bar_Desig	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Full Layers	1	1	1	1	1	1
Excess Layer Units %	0%	33%	33%	0%	0%	0%

However , 'Half-Tray_2' tray is telling us that the Actual UPT possible is '3' from the 'Actual UPT Possible' row so if we go back to the UPT column in the 'Product Parameters' box and change the UPT from 4 to 3 then we will then observer that the 'Half-Tray_2' column now indicates that with a new UPT of 3 (a deduction of 1 unit) the product 'Albacore tuna' (UPC: 998376) will now fit in a bar height of 2 inches. Therefore, the 'Albacore tuna' (UPC: 998376) product will be present in the first Half-tray output file ('HALF_tray_5in_Excess.xlsx') which. Stores the half-tray optimization list where a change from 5-inch to a 2-inch packing is identified.

	HALF_TRAY_2	HALF_TRAY_5	HALF_TRAY_7	METRIC_TRAY_2	METRIC_TRAY_5	METRIC_TRAY_7
UPT	**4**	**4**	**4**	**4**	**4**	**4**
Max UPL per Layer	3	3	3	6	6	6
Total Layers used	1	1	1	1	1	1
Defecit Top Layer Units	0	0	0	0	0	0
Actual UPT Possible	**3**	**3**	**3**	**6**	**6**	**6**
Bar_Desig	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Full Layers	1	1	1	1	1	1

Excess Layer Units %	0%	0%	0%	0%	0%	0%

The Tray Optimization calculator is shown below and it can be used one UPC at a time to identify



what number of units to reduce UPT by in order to affect a bar height change that will allow optimal tray packing. Of course the SAS programmes will generate lists for multiple UPCs that can then be used in the logistics team to consider the cost/benefits of reducing a UPC by x units thus facilitating an increase of y trays that can be packed.

END of Document