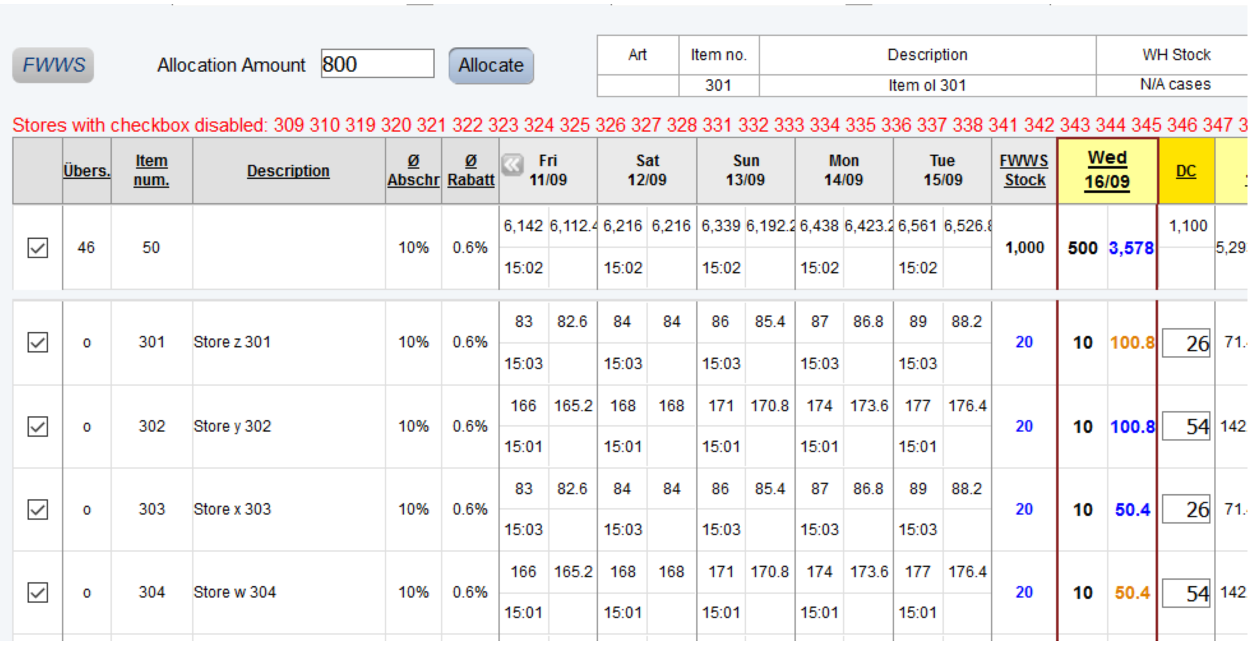
BKIT – LESSON12 – BAI TAP

(Không nộp bài nghỉ học)

Task: In many countries, there is the need to allocate assortment items from the area “fresh items” on a daily basis from WH (warehouse) to stores instead of being ordered by the stores.

To Do

With clicking on **Allocate** button in **Edit Mode** of **Controlling Cockpit** the value of field **Allocation Amount** should be distributed to stores with checkbox selected in first row of **Controlling Table** (distribution logic is described in following steps).
Results should be added to fields of **DC** column for these stores.



Allocation Amount 800

Art	Item no.	Description	WH Stock
	301	Item ol 301	N/A cases

Stores with checkbox disabled: 309 310 319 320 321 322 323 324 325 326 327 328 331 332 333 334 335 336 337 338 341 342 343 344 345 346 347 3

	Übers.	Item num.	Description	Ø Abschr	Ø Rabatt	Fri 11/09		Sat 12/09		Sun 13/09		Mon 14/09		Tue 15/09		FWWS Stock	Wed 16/09		DC	
☑	46	50		10%	0.6%	6,142 15:02	6,112.4	6,216 15:02	6,216	6,339 15:02	6,192.2	6,438 15:02	6,423.2	6,561 15:02	6,526.8	1,000	500	3,578	1,100	5,29
☑	o	301	Store z 301	10%	0.6%	83 15:03	82.6	84 15:03	84	86 15:03	85.4	87 15:03	86.8	89 15:03	88.2	20	10	100.8	26	71.
☑	o	302	Store y 302	10%	0.6%	166 15:01	165.2	168 15:01	168	171 15:01	170.8	174 15:01	173.6	177 15:01	176.4	20	10	100.8	54	142
☑	o	303	Store x 303	10%	0.6%	83 15:03	82.6	84 15:03	84	86 15:03	85.4	87 15:03	86.8	89 15:03	88.2	20	10	50.4	26	71.
☑	o	304	Store w 304	10%	0.6%	166 15:01	165.2	168 15:01	168	171 15:01	170.8	174 15:01	173.6	177 15:01	176.4	20	10	50.4	54	142

~~WFM-12091~~

Input: Warehouse Allocation Amount

Output: Allocated Amount for Selected Stores with 4 steps in calculation

Step One: filling in missing “Expected Sales” values

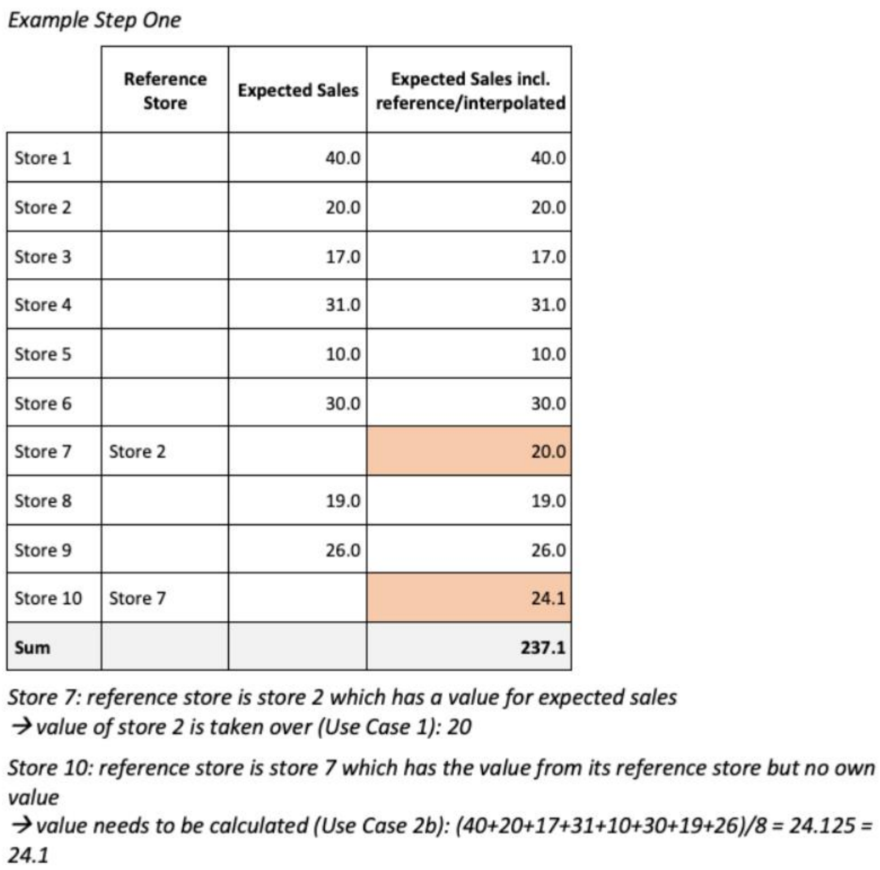
Use Case 1:
If a store selected is missing expected sales values and has a reference store maintained
--> use expected sales of reference store

Use Case 2:
If a store selected is missing expected sales values and has
a) no reference store maintained OR
b) a reference store maintained which is also missing expected sales itself
only values of reference store itself count here. It is not considered if reference store would have a reference store itself.
--> take the average of all stores with “own” data (stores with data from reference store or calculated data are skipped for average calculation); commercially rounded to one decimal digit
--> expected sales calculated with average should only be calculated and used for further calculation but should not be displayed in UI

Example Step One.png

If ALL selected stores have no expected sales and no reference store maintained (and thus a calculation of expected sales is not possible):

- stop the calculation
- display a localizable error message in red: "Expected sales cannot be calculated. Please add a reference store or include stores with expected sales for interpolation"



Example Step One

	Reference Store	Expected Sales	Expected Sales incl. reference/interpolated
Store 1		40.0	40.0
Store 2		20.0	20.0
Store 3		17.0	17.0
Store 4		31.0	31.0
Store 5		10.0	10.0
Store 6		30.0	30.0
Store 7	Store 2		20.0
Store 8		19.0	19.0
Store 9		26.0	26.0
Store 10	Store 7		24.1
Sum			**237.1**

Store 7: reference store is store 2 which has a value for expected sales
→value of store 2 is taken over (Use Case 1): 20

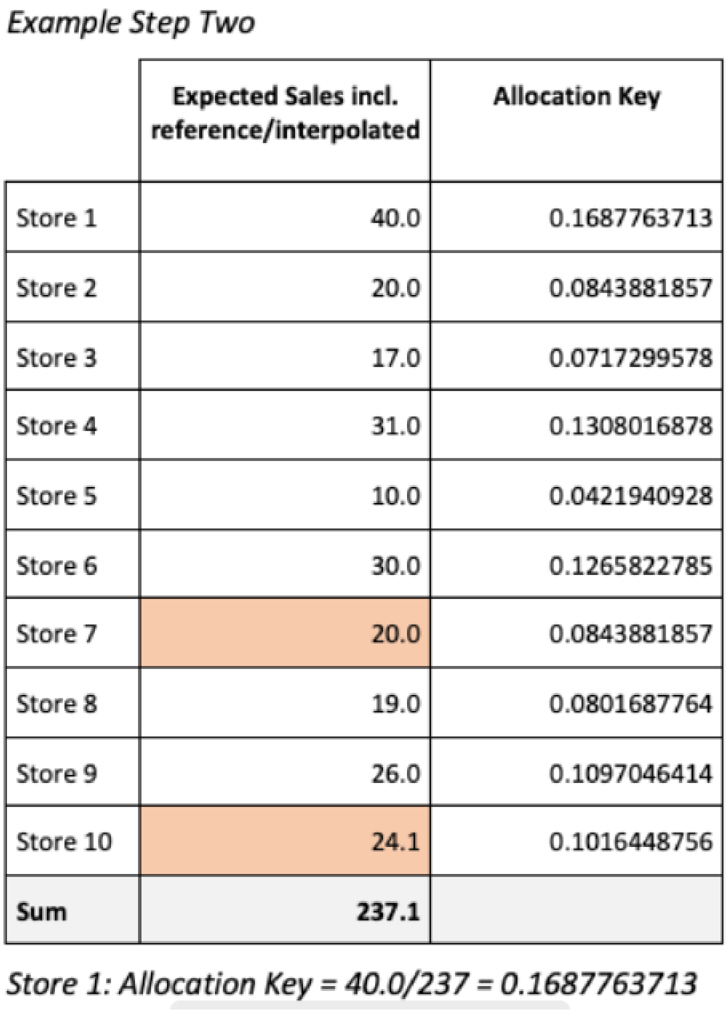
Store 10: reference store is store 7 which has the value from its reference store but no own value
→value needs to be calculated (Use Case 2b): (40+20+17+31+10+30+19+26)/8 = 24.125 = 24.1

Step Two: Calculation of “Allocation Key”

$\text{Allocation Key}_{\text{Store}} = (\text{Expected Sales incl. Reference Interpolated}_{\text{Store}}) / (\sum \text{Expected Sales incl. Reference Interpolated}_{\text{Store}})$

- for further calculation results should be commercially rounded to max. 10 decimal digits.

Example Step Two.png



Example Step Two

	Expected Sales incl. reference/interpolated	Allocation Key
Store 1	40.0	0.1687763713
Store 2	20.0	0.0843881857
Store 3	17.0	0.0717299578
Store 4	31.0	0.1308016878
Store 5	10.0	0.0421940928
Store 6	30.0	0.1265822785
Store 7	20.0	0.0843881857
Store 8	19.0	0.0801687764
Store 9	26.0	0.1097046414
Store 10	24.1	0.1016448756
Sum	**237.1**	

Store 1: Allocation Key = 40.0/237 = 0.1687763713

Step Three: Calculation of "Amount Allocated"

Amount Allocated $_{\text{Store}}$ = Allocation Key $_{\text{Store}}$ * (Allocation Amount + $\sum$ Stock Previous Day $_{\text{Store}}$) - Stock Previous Day $_{\text{Store}}$

- Allocation Amount is the amount the user added to the newly created field on top of the table
- Stock Previous Day $_{\text{Store}}$ is the value from respective column in **Controlling Table** (same logic for displaying in UI should be used here as well)
- $\sum$Stock of Previous Day $_{\text{Store}}$ is the sum over all stores in **Controlling Table** that are selected via checkbox in **Checkbox Column**

- results should be commercially rounded to an integer
- negative results should be replaced with 0 and also considered as 0 when calculating further steps

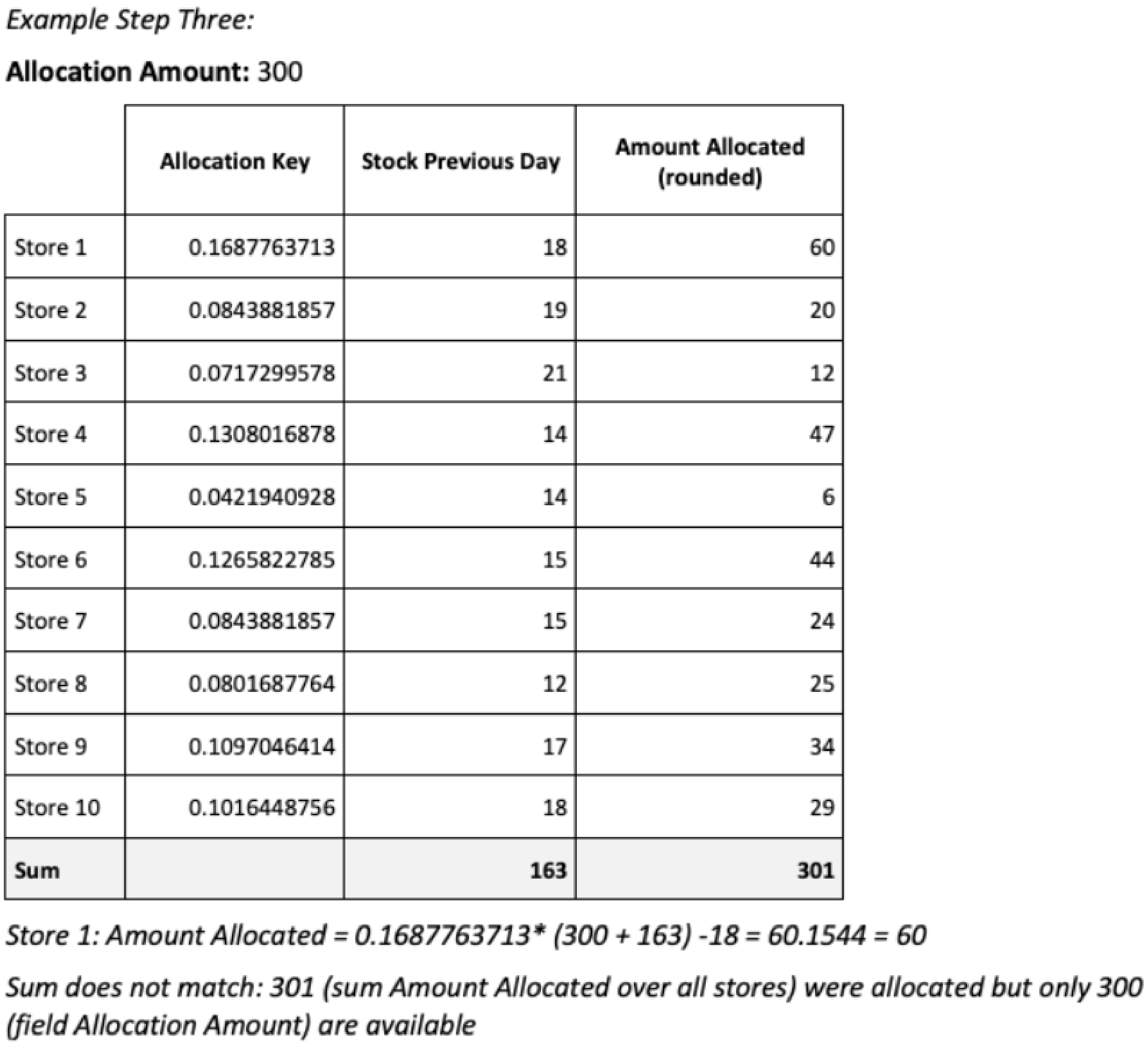
--> If sum of rounded **Amounts Allocated** over all stores checked via checkbox matches **Allocation Amount** added in field above the table:

- add values to **DC** column of respective stores and trigger real-time calculation as well as recalculation of any fields dependent on values of **DC** column

  --> If sum does not match:

- calculation needs to be continued with "Step Four".

Example Step Three.png



Example Step Three:

Allocation Amount: 300

	Allocation Key	Stock Previous Day	Amount Allocated (rounded)
Store 1	0.1687763713	18	60
Store 2	0.0843881857	19	20
Store 3	0.0717299578	21	12
Store 4	0.1308016878	14	47
Store 5	0.0421940928	14	6
Store 6	0.1265822785	15	44
Store 7	0.0843881857	15	24
Store 8	0.0801687764	12	25
Store 9	0.1097046414	17	34
Store 10	0.1016448756	18	29
Sum		**163**	**301**

Store 1: Amount Allocated = 0.1687763713 (300 + 163) -18 = 60.1544 = 60*

Sum does not match: 301 (sum Amount Allocated over all stores) were allocated but only 300 (field Allocation Amount) are available

Step Four: Fix Rounding Issues

Amounts Allocated need to be corrected until the sum over all selected stores matches value in field **Allocation Amount**.

First **Demand** per store needs to be identified via following formula:

$\text{Demand}_{\text{Store}} = \text{Expected Sales incl. Reference Interpolated}_{\text{Store}} - \text{Stock Previous Day}_{\text{Store}}$

- results should be commercially rounded to an integer
- negative results should be replaced by 0 and also considered as 0 when calculating further steps

Example Step Four - Demand.png

Afterwards it needs to be determined if sum of **Amounts Allocated** over all stores selected via checkbox is bigger or smaller than value in field **Allocation Amount**

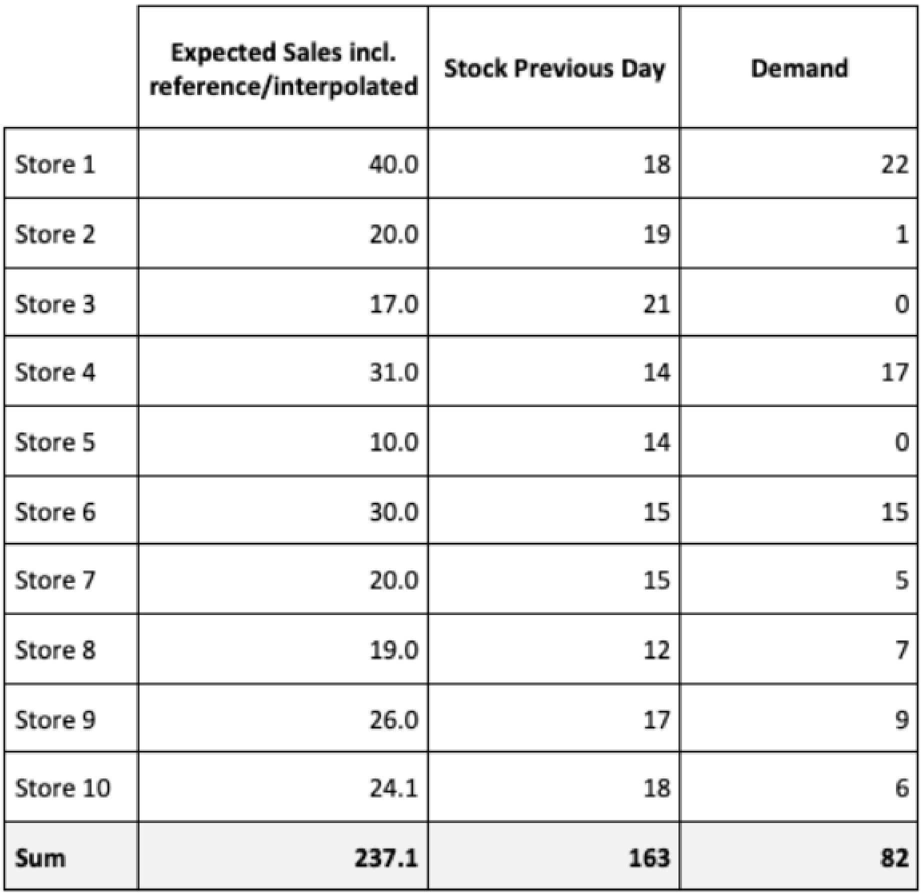
- $\sum \text{Amounts Allocated}_{\text{Store}}$ **>** Allocation Amount
- Iterative: Subtract one allocation unit (one case) from the store which's difference between **Amount Allocated** and **Demand** is biggest
- if a store has "Amount Allocated" = 0, it is skipped and no difference is calculated
- only positive values should be considered for difference; if difference of every store results into negative values, the one closest to 0 should be taken into account
- If two or more stores have the same difference: take the one with smallest **Demand**
- If two or more stores also have same **Demand**: take the one with smallest **Expected Sales incl. reference/interpolated**
- If two or more stores also have same **Expected Sales incl. reference/interpolated**: take the one with smallest store number
- After subtracting one allocation unit from a store, the difference between **Amount Allocated** and **Demand** is calculated again for this store and the iterative approach is continued until sum of **Amounts Allocated** over all stores selected via checkbox equals value of field **Allocated Amount**

- $\sum \text{Amounts Allocated}_{\text{Store}}$ **<** Allocation Amount
- Iterative: Add one allocation unit (one case) from the store which's difference between **Amount Allocated** and **Demand** is smallest.
- the smallest value, also negative value, should count as smallest difference
- If two or more stores have the same difference: take the one with biggest **Demand**
- If two or more stores also have same **Demand**: take the one with biggest **Expected Sales incl. reference/interpolated**
- If two or more stores also have same **Expected Sales incl. reference/interpolated**: take the one with smallest store number
- After adding one allocation unit to a store, the difference between **Amount Allocated** and **Demand** is calculated again for this store and the iterative approach is continued until sum of **Amounts Allocated** over all stores selected via checkbox equals value of field **Allocated Amount**

- final results should be commercially rounded to an integer
- negative results should be replaced with 0
  ---> add values to **DC** column of respective store and trigger real-time calculation as well as recalculation of any fields dependent on values of **DC** column

Example Step Four – adapting final Amounts Allocated.png

Example Step Four - Demand:



	Expected Sales incl. reference/interpolated	Stock Previous Day	Demand
Store 1	40.0	18	22
Store 2	20.0	19	1
Store 3	17.0	21	0
Store 4	31.0	14	17
Store 5	10.0	14	0
Store 6	30.0	15	15
Store 7	20.0	15	5
Store 8	19.0	12	7
Store 9	26.0	17	9
Store 10	24.1	18	6
Sum	**237.1**	**163**	**82**

Store 1: Demand = 40.0 – 18 = 22