Tips & Tricks # 48 – Non-financial Data in GL and Key Performance Indicators

One of the import themes in business data processing is to extract and process data that tells you about your “normal” and to help you anticipate changes from the normal so you can respond with the right kind of change. The process of providing a mathematical way of smoothing out the variations in the data to get at the norm allows you to get beyond e.g., the number of selling days in a month, to get “smoothed” numbers that try to eliminate the effect of different lengths of the months and variations from week to week with a selling season or, the effects of hiring on a number of new people. These non-financial numbers (store visits per week, selling days in the month, employees on the selling floor, office space required for various departments) come from outside the PBS system but can be used to normalize the data. The number of employees on the sales floor can normalize the data to average weekly sales volume per sales person. This is much like the quoting of national statistics where you might see infection rates “per thousand” people to compare various states where the states vary substantially in population. The ability to store and generate non-financial data within the reports generated by the Financial Statement processor (F/S) in PBS can be a big help in this normalizing of data and can be used to make tactical and strategic changes to a company’s processes and direction.

As indicated here, these Key Performance Indicators (KPI’s) are not usually just “raw” data like sales dollars in the month but require some processing, some scaling, to relate them to factors such as seasons, market trends, numbers of customers or internal factors controlling expenditures such as number of employees and office or plant sizes. The raw data is scaled to some non-financial number that allows month to month or location to location comparisons to be made that are actually comparable. This becomes much more important in circumstances where the business is rapidly changing either in expansion, as the company grows, or in recession, as in what we have all witnessed and participated in with Covid.

A store who would like to know on a weekly basis, the number of customers that come into the store would greatly benefit from integration with their Point-of-Sale System and PBS accounting since the POS data can flow easily into the accounting system. With this flow you should be able to easily gather the number of invoices (usually the same as the number of shoppers) and the amount spent on each register ticket. Aggregating that over a week gives you 52 data points per year; confirming when your busy seasons are and, if there are good weeks and bad weeks and the general trend in your sales volume.

Both Passport’s own POS system and the Passport interface to the NCR Counterpoint POS system offer this kind of integration/functionality.

If you were to collect daily numbers you could distinguish weekdays from weekends. This kind of KPI might point out that on Sunday while there are a fair number of shoppers, the purchases are small and only necessities – i.e., there are few people doing a big weekly or semi-monthly shop. That can lead to a decision not open Sunday, have fewer employees available on Sunday, or curtail certain in-store services e.g., Sundays are self-bagging days. The opposite is going to be true of building centers and hardware stores (Saturdays and Sundays are the DIY-er days).

Even if our cash registers don't automatically feed the accounting system, the register receipts can be counted and tallied, and that summary information manually entered – see below.

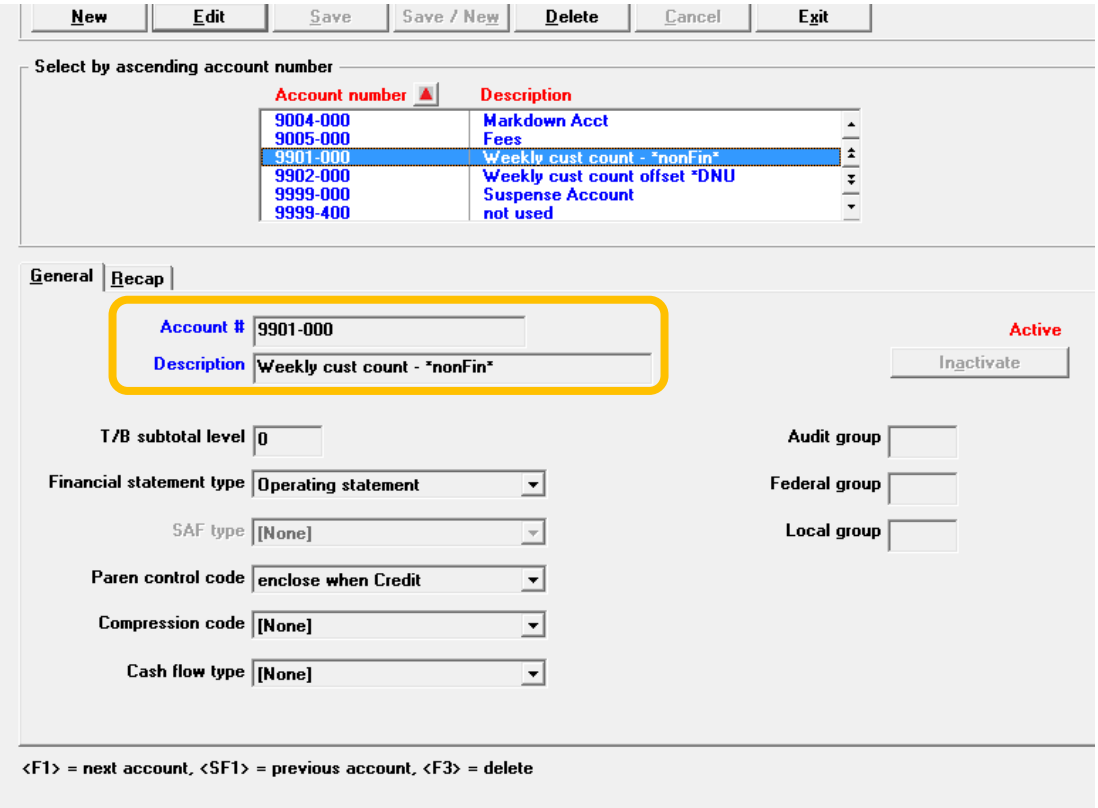
PBS does not automatically handle non-financial information at present, but we are actively exploring the addition of this feature and welcome suggestions. What follows is a way of implementing this without some of the conveniences that would be available with this automation.

So, what we can do to achieve this? The fundamental principle here is that anything numeric can be entered into the GL/Trial Balance; but only balanced transaction sets can be posted, i.e., debits = credits. (The non-financial automation we mentioned above would allow transactions that don't need offsets.)

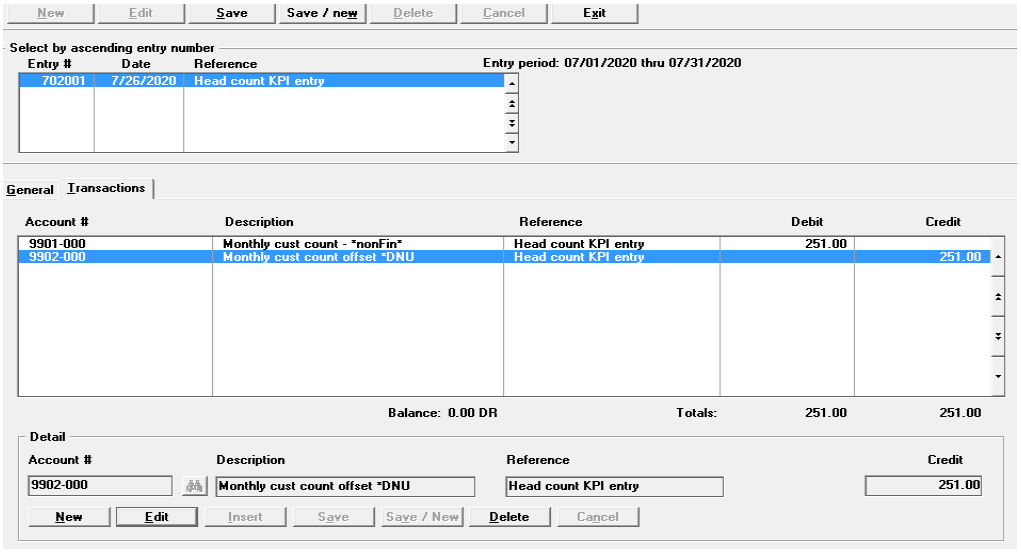
Using our example, we will set aside (assuming a four-digit account) a range of say 9900 through 9990 for the non-financial data and designate 9901 as weekly headcount of shoppers. To allow any entries into this account to balance we will also setup 9902 as its offset knowing that this is a dummy to get the 9901-transaction posted. Clearly if we enter 251 DB for the headcount for one week into 9901, we will also enter 251 CR into 9902. Since GL accounts are indexed by account number and date, we can have a separate headcount for each day, week or month of the year. If we are entering daily values weekly and monthly, totals for the headcount can be done by producing a regular F/S subtotal – see below. If there were other non-financial numbers that we want to track we could use 9903 and 9904, 99905/06 as pairs of offsets and so on. For example, there could also be an entry for the number of selling days in the month or total poundage shipped etc. etc. These will be a function of your business and your imagination to find the best suited KPI's.

In our example - by entering the headcount per week, we can use it with the arithmetic functions in the F/S generator to compute sales per customer visit per week.

The Chart entry for this would look like this:



The entry to add this week's total would look like this:



Note the "in and out" nature of the transaction with the two accounts paired but opposite in sign. We use debit for the main statistic just a a matter of convenience with not having to worry about signs.

With a posting like this:

Starting entry : "First" Ending entry : MLE9999999

Entry date	Account Number	Description	Reference	Debit amount	Credit amount
Entry #: MLE702001	Reverse next period? No	Correcting entry? No			
07/26/2020	9901-000	Monthly cust count - *nonFin*	Head count KPI entry	251.00	
07/26/2020	9902-000	Monthly cust count offset *DNU	Head count KPI entry		251.00
			Entry totals:	251.00	251.00
2 detail lines printed		1 journal entries.	Report totals:	251.00	251.00

-- End of report --

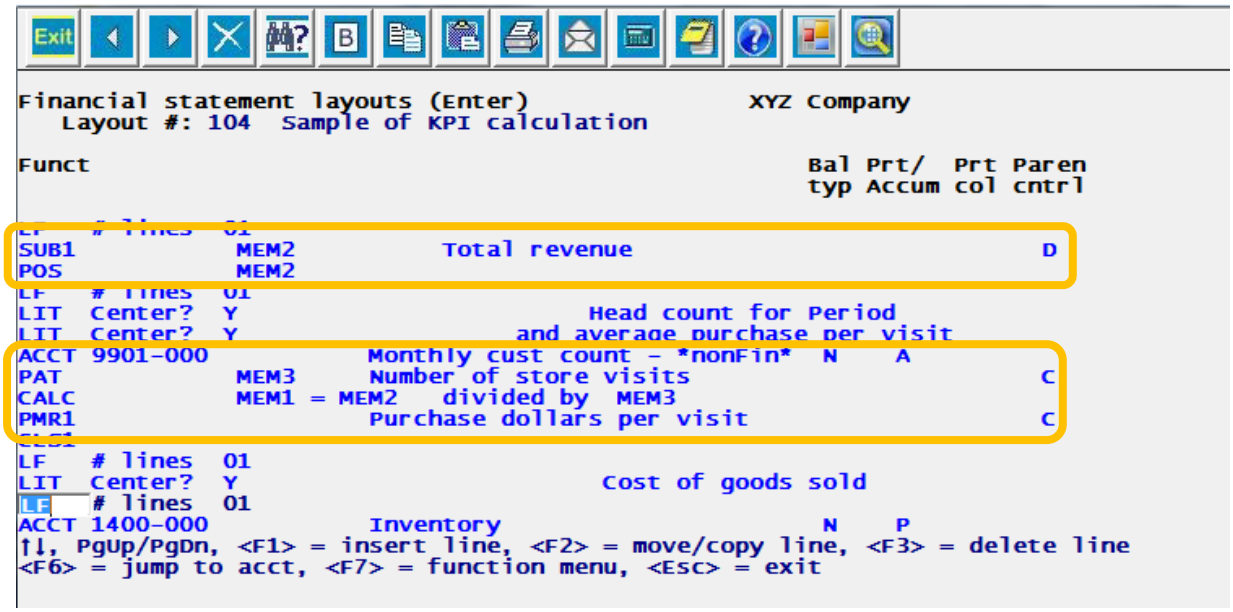
With this as setup, a sample (and simple) Financial Statement was constructed using Sales and Cost of Sales to calculate Gross Profit. After the Revenue/Sales section, the number of store visits from account 9901 is printed and then it is used to calculate an average purchase per visit. To show that these calculations can be distributed throughout the report, a similar calculation is done after the display of the Gross Profit showing the profit per visit – which is the Gross Profit divided by the number of visits. Here is the F/S:

XYZ Company
Profit & loss statement
Period: 07/01/2020 to 07/31/2020

	2020 CURRENT PERIOD ACTUALS Sub Account 000 Amount	2020 YEAR-TO-DATE ACTUALS Sub Account 000 Amount
Revenue		
Sales	$ 24,728.00	$ 24,728.00
Total revenue	$ 24,728.00	$ 24,728.00
Head count for Period and average purchase per visit		
Number of store visits	$ 251.00	$ 251.00
Purchase dollars per visit	98.52	98.52
Cost of goods sold		
Inventory cost	18,400.68	18,400.68
Total cost of goods sold	$ 18,400.68	$ 18,400.68
Gross profit	$ 6,076.32	$ 6,076.32
Profit per visit	24.21	24.21

The example here is to gather weekly not monthly statistics and the naming should be consistently “weekly”.

The F/S code to do this is shown below with explanation:

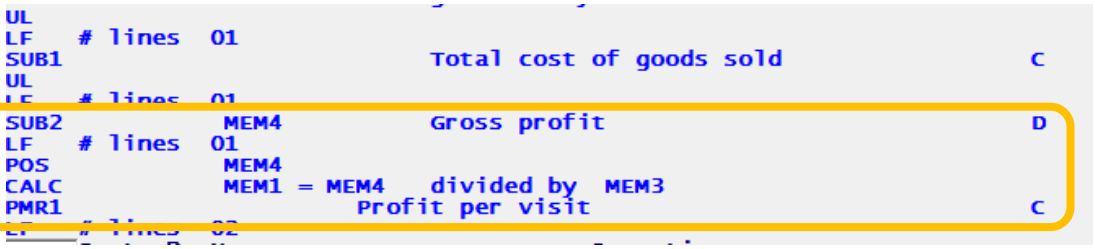


The first box shows the code that does the sub-total over all the (non-zero) revenue accounts in the data. Simultaneous to the sub-totaling, the sub-total is moved into a memory register (#2) to hold this value for later use in the calculation. Since the normal revenue value is a credit, we use the print code, at the far right, to display the credit value without a sign. The next command takes the Credit value in the MEM2 and changes it to be Positive since we want all the numbers printed here to be displayed as positive.

The second box shows the calculation and display of our simple KPI's - store visits, and average purchase per visit. Here we identify the 9901 to be included in, and use the PAT (Print Accumulated Total) as an easy way to move this value into register #3 (MEM3) so we can use it for calculation. The PAT command also prints the value with the tag "Number of store visits".

Next, we do the calculation which basically divides the value in MEM2 by the value we just put into MEM3 and store the result in MEM1. This value is then printed with its own tag line "**Purchase dollars per visit"** using the PMR1 command (Print Memory Register#1).

Essentially the same techniques are used later to compute the profit per visit:



Here Profit is computed by Sub-Totaling the Sub-totals for Revenue and Cost of Sales (the SUB2 command). Notice the sub-totaled value is again copied, this time, into the MEM4 register. Since the revenue (a credit value – see above) is larger than the cost of sales, so the signed sum of these, the gross profit, will also be a credit and will therefore need a "D" to be displayed without a sign and for the same reason we have to change the sign of the gross profit value to a plus (POS command) for the purposes of out next calculation. This divides the MEM4 (gross profit) by out store visit counter (MEM3) from previous calculation to arrive at the profit per visit. Notice we are re-using the MEM1 again to receive the calculated value since we have

already printed the value from the previous calculation. Once again, the calculated value is displayed using the PMR1 command (Print Memory Register#1).

So, there it is - the beginnings of adding in KPI's into your financial statements.

This is our initial foray into the realm of coding non-financials into financial statements. If this is of interest, please let us know if you would like to see more via email to psi@pass-port.com.