Description: Revision in the computation and presentation of standard and actual direct material costing in the reports. Report should compute and display the cost of the raw material, resin cost, PI-FG processing cost, PI hidden profit and landed cost. Labor cost should display as SF added value (labor cost of the SF material) and FG added value (labor cost of the product). Affected reports are: (1) Finished Goods report; (2) Sales Report; (3) Sales Report for Non FG Items; (4) Sample JO Report; (5) Direct Material and Labor Percentage Report; (6) Inventory Turnover Report; (7) WIP Shop Floor Report; and (8) Miscellaneous Transaction Report.

I. STORED PROCEDURES

1. Name: LSP_Rpt_DM_FinishedGoodsReportSp
Description: Stored procedure that returns the finished goods report based on filtered date.

Parameters: (1) Start Date; and **(2)** End Date
Procedure:
- Set the initial value of the two parameters. If their value is null, set their value to the current date.
- Execute the stored procedure "LSP_DM_GetAllFinishedTransactionSp" passing the value of start and end date parameter to the executing stored procedure.
- Use a cursor to fetch one by one the rows returned by the executed stored procedure in order to get and compute the standard and actual cost of each items.
- For each fetched items per row, execute the following stored procedures:
  - LSP_DM_ComputeStandardFGAddedValueSp - passing the fetched item as input parameter and returning the value of Standard FG added value cost.
  - LSP_DM_ComputeActualFGAddedValueSp - passing the fetched job order number and suffix as input parameters and returning the value of actual FG added value cost.
  - LSP_DM_ComputeStandardMaterialCostSp - passing the fetched item and transaction date as input parameters, returning the value of standard material, landed, resin, PI-FG processing and SF value added cost.
  - LSP_DM_ComputeActualMaterialCostSp - passing the fetched JO number, suffix and item as input parameters and returning the value of actual material, landed, resin, PI-FG processing, SF value added and PI vendor cost.
- Compute the standard and actual PI hidden profit:
  - Standard PI Hidden Profit= PI Vendor Cost - (Standard Resin Cost + Standard PI-FG Processing cost)
  - Actual PI Hidden Profit= PI Vendor Cost - (Actual Resin Cost + Actual PI-FG Processing cost)
- Insert values to the report table sort by transaction date.

2. Name: LSP_DM_GetAllFinishedGoodsTransactionSp
Description: Stored procedure responsible for getting all "Finished" (or job receipt) transaction type from the material transactions based on filtered date.

Parameters: (1) Start Date; and **(2)** End Date
Procedure:
- Set the initial value of the two parameters. If their value is null, set their value to the current date.
- Insert values to the report table.

Primary Source Table: matltran
Filter conditions: (1) trans_date BETWEEN parameters start date and end date; **(2)** qty is greater than zero; **(3)** trans_type is equal to "F"; **(4)** ref_type is equal to "J"; and **(5)** item starts with "FG-".

Joined Source Tables:

Table Name	Join Type	Join Condition
item	INNER JOIN	matltran.item = item.item
famcode	INNER JOIN	item.family_code = famcode.family_code
job	INNER JOIN	matltran.ref_num = job.job AND matltran.ref_line_suf = job.suffix
coitem	LEFT OUTER JOIN	job.ord_num = coitem.co_num AND job.ord_line = coitem.co_line
co	LEFT OUTER JOIN	coitem.co_num = co.co_num
custaddr	LEFT OUTER JOIN	co.cust_num = custaddr.cust_num AND co.cust_seq = custaddr.cust_seq

- Select data from report set table getting the last transaction date and sum of quantity completed grouped by Job Order number, suffix, PO number, item, item description, product code, family code and by family code description.

3. Name: LSP_Rpt_DM_SalesSampleJOReportSp

Description: Stored procedure that returns the sales and sample JO report based on filtered date.

Parameters: (1) Start Date; and **(2)** End Date

Procedure:

- Set the initial value of the two parameters. If their value is null, set their value to the current date.
- Execute the stored procedure "LSP_DM_GetAllShipTransactionSp" passing the value of start and end date parameter to the executing stored procedure.
- Use a cursor to fetch one by one the rows returned by the executed stored procedure in order to get and compute the standard and actual cost of each items. Note: Get all transactions where the "Shipped Quantity" is greater than zero (0).
- For each fetched items per row, perform the following procedures:
  - Get the job remarks and determine if the job is recoverable or not.
  - Execute stored procedure LSP_ConvertUsdToPhpCurrencySp - passing the fetched transaction date as input parameter and the exchange rate value as the output.
  - Determine if the job order exist by getting the count of row
  - If the job order count is not equal to zero(0) and the item code starts with "FG-", do the following:
    - □ Execute "LSP_DM_ComputeStandardFGAddedValueSp" passing item code as input parameter, returning the standard FG added value cost.
    - □ Execute "LSP_DM_ComputeActualFGAddedValueSp" passing job order number and suffix as input parameter, returning the actual FG added value cost.
    - □ Execute "LSP_DM_ComputeStandardMaterialCostsSp" passing item code and transaction date as input parameters, returning the standard material, landed, resin, PI-FG processing and SF value added cost.
    - □ Execute "LSP_DM_ComputeActualMaterialCostsSp" passing job order number, suffix and item code as input parameters, returning the actul material, landed, resin, PI-FG processing, SF value added cost and. PI vendor cost.
    - □ Compute the standard and actual PI Hidden Profit:
      - ▪ Standard PI Hidden Profit= PI Vendor Cost - (Standard Resin Cost + Standard PI-FG Processing cost)
      - ▪ Actual PI Hidden Profit= PI Vendor Cost - (Actual Resin Cost + Actual PI-FG Processing cost)
  - Otherwise, do the following:
    - □ Set all standard and actual cost to zero(0).
    - □ If item code starts with "FG-", execute "LSP_DM_GetItemStandardCostItemPricingSp" passing the item and transaction date as input parameters, returning the actual material cost.

□ If item code starts with "SF-", execute the following stored procedures:

- LSP_DM_GetItemStandardCostItemPricingSp - passing the fetched item and transaction date as input parameter, returning the actual material cost.
- LSP_DM_ComputeStandardSFAddedValueSp - passing the fetched item as input parameter, returning the standard SF job added value cost.
- LSP_DM_ComputeStandardMaterialCostsSp - passing the fetched item and transaction date as input parameters, returning the standard material, landed, resin, PI-FG processing and SF added value cost.

□ If item code starts with "PI-FG-", perform the following:

- Execute "LSP_DM_GetItemStandardCostItemPricingSp" passing the fetched item and transaction date as input parameter, returning the standard material cost.
- Execute "LSP_DM_GetPIResinProcessingCostSp" passing the fetched item and transaction date as input parameter, returning the actual resin cost, actual PI-FG processing cost and PI vendor cost.
- Compute for the actual PI Hidden Profit (PI vendor cost - (Actual Resin Cost + PI-FG Processing)

□ Otherwise, execute the following stored procedures:

- LSP_DM_GetItemStandardCostItemPricingSp - passing the fetched item and transaction date as input parameters, returning the standard material, landed, resin, PI-FG processing and SF added value cost.
- LSP_DM_GetUnitMaterialLandedCostSp - passing the fetched item, lot number and transaction date as input parameters, returning the actual material and landed cost.

• Insert values to the report set table. If the customer name is "LSP* Section", it should be included in the sample JO group, otherwise should be in the sales report group.

4. Name: LSP_DM_GetAllShipTransactionSp

Description: Stored procedure responsible for getting all "Ship" transaction type from the material transactions based on filtered date.

Parameters: (1) Start Date; and **(2)** End Date

Procedure:

- Set the initial value of the two parameters. If their value is null, set their value to the current date.
- Insert values to the report table.

Primary Source Table: matltran

Filter conditions: (1) trans_date BETWEEN parameters start date and end date; **(2)** trans_type is equal to "S"; and **(3)** ref_type is equal to "O"

Joined Source Tables:

Table Name	Join Type	Join Condition
item	INNER JOIN	matltran.item = item.item
famcode	INNER JOIN	item.family_code = famcode.family_code
coitem	LEFT OUTER JOIN	matltran.ref_num = coitem.co_num AND matltran.ref_line_suf = coitem.co_line
co	LEFT OUTER JOIN	coitem.co_num = co.co_num
custaddr	LEFT OUTER JOIN	co.cust_num = custaddr.cust_num AND co.cust_seq = custaddr.cust_seq

- Select data from report set table getting the last transaction date and sum of quantity shipped multiplied by negative one (-1), data grouped by PO number, CO number, CO line, item, item description, product code, JO number, JO suffix, lot number, family code, family code description, customer number, customer ship to, customer name and sales unit price.

5. Name: LSP_Rpt_DM_DirectMaterialLaborPercentageReportSp

Description: Stored procedure that returns the direct material and labor percentage report based on filtered date, product code and product model (item code).

Parameters: (1) Start Date; **(2)** End Date; **(3)** Start Product Code; **(4)** End Product Code; **(5)** Start Model; and **(6)** End Model

Procedure:

- Set the initial value of the report parameters, if their value is null, set for a default value
  - Start and End Date = current date.
  - Start Product Code = first product code value where product code starts with "FG-"
  - EndProduct Code = last product code value where product code starts with "FG-"
  - Start Model = first item model filtered by product code (start and end product code)
  - End Model = last item model filtered by product code (start and end product code)
- Execute the stored procedure "LSP_DM_GetFilteredFinishedGoodsTransactionSp" passing the value of start date, end date, start product code, end product code, start model and end model as parameters to the executing stored procedure.
- Use a cursor to fetch one by one the rows returned by the executed stored procedure in order to get and compute the standard and actual cost of each items.
- For each fetched items per row, execute the following stored procedures:
  - LSP_ConvertUsdToPhpCurrencySp - passing the fetched transaction date as input parameter and returning the exchange rate value.
  - LSP_DM_GetItemStandardCostItemPricingSp - passing the fetched item and transaction date as input parameters and returning the EX Works price.
  - LSP_DM_ComputeStandardFGAddedValueSp - passing the fetched item as input parameter and returning the standard FG added value cost.
  - LSP_DM_ComputeActualFGAddedValueSp - passing the fetched JO number, suffix as input parameters and returning the actual FG added value cost.
  - LSP_DM_ComputeStandardMaterialCostsSp - passing the fetched item and transaction date as input parameters and returning the standard material, landed, resin PI-FG processing and SF added value cost.
  - LSP_DM_ComputeActualMaterialCostsSp - passing the fetched JO number, suffix, and item as input parameters and returning the actual material, landed, resin PI-FG processing, SF added value and PI vendor cost.
- Compute for the standard and actual PI hidden profit cost.
  - Standard PI Hidden Profit= PI Vendor Cost - (Standard Resin Cost + Standard PI-FG Processing cost)
  - Actual PI Hidden Profit= PI Vendor Cost - (Actual Resin Cost + Actual PI-FG Processing cost)
- Insert values to the report table

6. Name: LSP_DM_GetFilteredFinishedGoodsTransactionSp

Description: Stored procedure responsible for getting all "Finished" (or job receipt) transaction type from the material transactions based on filtered date, product code and product model.

Parameters: (1) Start Date; **(2)** End Date; **(3)** Start Product Code; **(4)** End Product Code; **(5)** Start Model; and **(6)** End Model

Procedure:

- Set the initial value of the report parameters, if their value is null, set for a default value
  - Start and End Date = current date.
  - Start Product Code = first product code value where product code starts with "FG-"
  - EndProduct Code = last product code value where product code starts with "FG-"
  - Start Model = first item model filtered by product code (start and end product code)
  - End Model = last item model filtered by product code (start and end product code)

- Insert values to the report table.

Primary Source Table: matltran

Filter conditions: (1) trans_date BETWEEN parameters start date and end date; **(2)** qty is greater than zero; **(3)** trans_type is equal to "F"; **(4)** ref_type is equal to "J"; **(5)** item starts with "FG-"; (7) product code is between the start and end product code parameter; and **(8)** item is between the start and end model parameter.

Joined Source Tables:

Table Name	Join Type	Join Condition
item	INNER JOIN	matltran.item = item.item
famcode	INNER JOIN	item.family_code = famcode.family_code

- Select data from report set table getting the last transaction date and sum of quantity completed grouped by Job Order number, suffix, item, item description, product code, family code and family code description.

7. Name: LSP_Rpt_DM_WIPShopFloorReportSp

Description: Stored procedure responsible for getting the WIP Shop Floor Report.

Procedure:

- Get all jobs where the status is "Released" and the sum of quantity completed and quantity scrapped is not equal to the job quantity released.
- For each job order, get the material by executing the stored procedure "LSP_GetJobMaterialIssueWithdrawSp" passing the job order, suffix and item as parameter. Store the return data of the stored procedure in a temporary table variable.
- Declare a cursor to fetch the materials of the job. For each fetched rows, compute the material cost, landed cost, resin cost, PI-FG processing cost, PI vendor cost and the quantity wip based on the issued materials, quantity completed, quantity scrapped versus the material quantity of the job.
- Compute the PI hidden profit (PI vendor cost - sum of PI-FG processing and resin cost)
- Insert values to the report table.

8. Name: LSP_Rpt_DM_InventoryTurnOverReportSp

Description: Stored procedure responsible for getting the Inventory Turnover Report.

Parameter: Details (show details or not)

Procedure:

- Set the global variables start date and end date:
  - Start Date = First day of the previous year for the current month (based on generation date)
  - End Date = Last day of the previous month (based on generation date)
- Execute the stored procedure "LSP_DM_InventoryTurnoverTransactionSp" passing the start date and end date as the input parameter. Insert the returned data into a temporary variable table.
- Use a cursor to get the accumulated material and landed cost grouped per year and month then by the product code .
- For each fetched data, execute stored procedure "LSP_DM_GetInventorySafetyStockMaterialLandedCostSp" passing the fetch product code as input parameter, returning the current inventory material and landed cost and the safety stock material cost.
- Use a cursor to get all the product codes from the report table. Update the three months usage for material and landed cost, and the maximum three months material and landed cost usage.
- If show details parameter is true, insert values to the report set table, get from the temporary variable table.

9. Name: LSP_DM_InventoryTurnoverTransactionSp

Description: Stored procedure that get and returned the transactions for inventory turnover report.

Parameter: (1) Start Date; and **(2)** End Date

Procedure:

- Set the initial value of the two parameters. If their value is null, set their value to the current date.
- Use a cursor to get the transactions needed in Inventory Turnover report.

Source Tables:

Table Name	Join Type	Join Condition
matltran	PRIMARY TABLE	N/A
item	LEFT OUTER JOIN	matltran.item = item.item
Conditions:	- trans_date BETWEEN StartDate and EndDate parameter - trans_type is equal to "S" - ref_num is like: "*PDN*", "*PIS*", "*QAC*", "*PDE*" or "*MDE*" - product_code does not starts with "FG-", "OS-"; does not ends in "-SUP"	
UNION ALL		
matltran	PRIMARY TABLE	N/A
item	LEFT OUTER JOIN	matltran.item = item.item
Conditions:	- trans_date BETWEEN StartDate and EndDate parameter - trans_type is equal to "I" or "W" - ref_num is like "1#*" or "1#RM-*" - product_code does not starts with "FG-", "OS-"; does not ends in "-SUP"	
UNION ALL		
matltran	PRIMARY TABLE	N/A
item	LEFT OUTER JOIN	matltran.item = item.item
reason	LEFT OUTER JOIN	matltran.reason_code = reason.reason_code AND reason.reason_class = "MISC ISSUE"
Conditions:	- trans_date BETWEEN StartDate and EndDate parameter - trans_type is equal to "G" - reason_code are: "ARM", "ETS", "PIU" and "URE" - product_code does not starts with "FG-", "OS-"; does not ends in "-SUP"	

- For each fetched rows, set the material, landed, resin, PI-FG processing and PI vendor cost to zero(0)
- If the item starts with "SF-", execute stored procedure "LSP_DM_GetItemStandardCostItemPricingSp" passing the item and transaction date as input parameters, returning the material cost value.
- If the item starts with "PI-FG-", execute stored procedure "LSP_DM_GetPIResinProcessingCostSp" passing item and transaction date as input parameters, returning the resin, PI-FG processing and PI vendor cost values.
- Otherwise, execute stored procedure "LSP_DM_GetUnitMaterialLandedCostSp" passing item, lot number and transaction date as input parameters, returning the material and landed cost values.
- Compute the PI hidden profit cost (PI Vendor cost less the sum of Resin cost and PI-FG processing cost).
- If lot number contains "stock", execute the stored procedure "LSP_DM_GetItemStandardCostItemPricingSp" passing the item and transaction date as input parameters returning the material cost value.
- Insert values to the report set table.

10. Name: LSP_DM_GetInventorySafetyStockMaterialLandedCostSp

Description: Stored procedure that returns the inventory material and landed cost, and safety stock material cost.

Parameter: (1) Product Code; **(2)** Inventory Material Cost; **(3)** Inventory Landed Cost; and **(4)** Safety Stock Material Cost

Procedure:

- Set values for RM product code and SF product code.
  - RM Product Code = "RM-" + product code parameter value
  - SF Product Code = If product code parameter value is "PACKNG" then value is "SA-PACK"; if value is "CS-AT" then value is "SA-CSAT"; otherwise "SA-" + product code parameter value
- Use a cursor to get the quantity on hand of each item.

Source Tables:

Table Name	Join Type	Join Condition
item	PRIMARY TABLE	N/A
lot_loc	LEFT OUTER JOIN	item.item = lot_loc.item
Conditions: product_code is RM Product Code or SF Product Code		

- For each fetched rows, get the safety stock quantity of the item.
- Execute stored procedure "LSP_DM_GetItemStandardCostItemPricingSp" passing the item and today date as the input parameter, returning the item pricing cost.
- If item code starts with "SF-", execute stored procedure "LSP_DM_GetItemStandardCostItemPricingSp", passing the item and today's date as input parameter, returning the material cost.
- If item code starts with "PI-FG-", execute stored procedure "LSP_DM_GetPIResinProcessingCostSp", passing the item and today's date as input parameter, returning the material and landed cost.
- Otherwise, execute "LSP_DM_GetUnitMaterialLandedCostSp" passing the item, lot number and today's date as input parameter, returning the material and landed cost values.
- Compute the PI hidden profit cost (PI Vendor cost less the sum of Resin cost and PI-FG processing cost).
- If lot number contains "stock", execute the stored procedure "LSP_DM_GetItemStandardCostItemPricingSp" passing the item and transaction date as input parameters returning the material cost value.
- Insert values to the item cost table, setting cost multiplied by the lot quantity.
- Set values for the output parameters:
  - Inventory Material Cost = Sum of material cost
  - Inventory Landed Cost = Sum of landed cost
  - Safety Stock Material Cost = Sum of safety stock cost

11. Name: LSP_Rpt_DM_MiscellaneousTransactionReportSp

Description: Stored procedure that returns the Miscellaneous Transaction report based on filtered date.

Parameters: (1) Start Date; and **(2)** End Date

Procedure:

- Set the initial value of the two parameters. If their value is null, set their value to the current date.
- Set up an SQL Command to execute the store procedure "LSP_DM_GetScrapDataForMiscTransactionReportSp" passing the start date and end date as input parameters.
- Execute the SQL command, insert the returned values to the report set table.
- Get the total cost of all SF Scrap (material, landed, resin, PI-FG processing, PI hidden profit, sf added and fg added value cost)
- Declare a cursor that will get and hold of the miscellaneous transactions.

  Primary Source Table: matltran

  Filter conditions:(1)trans_date BETWEEN parameters start date and end date; **(2)**trans_type is "B", "G" or "H".

Joined Source Tables:

Table Name	Table Alias	Join Type	Join Condition
item	item	INNER JOIN	matltran.item = item.item
reason	r1	LEFT OUTER JOIN	matltran.reason_code = r1.reason_code AND r1.reason_class = "MISC ISSUE"
reason	r2	LEFT OUTER JOIN	matltran.reason_code = r2.reason_code AND r2.reason_class = "MISC RCPT"

- For each fetched rows, do the following:
  - If item starts with "FG-", execute the following stored procedures:
    - □ LSP_DM_GetItemStandardCostItemPricingSp = item and transaction date as input parameters, returns material cost.
    - □ LSP_DM_ComputeStandardFGAddedValueSp = item input parameter, returns FG added value cost.
  - If item starts with "SF-", execute the following stored procedures:
    - □ LSP_DM_GetSFIssueWithdrawMaterialSp = lot, item and transaction date as input parameters, returns material, landed, resin, PI-FG processing and PI vendor cost.
    - □ LSP_DM_ComputeStandardSFAddedValueSp = item input parameter, returns SF added value cost.
  - If item starts with "PI-FG", execute stored procedure "LSP_DM_GetPIResinProcessingCostSp" passing item, and transaction date as input parameter, returns resin, PI-FG processing and PI vendor cost.
  - Otherwise, execute "LSP_DM_GetRMForMicellaneousTransactionSp" passing item, lot number, transaction date and material transaction cost as input parameters and returns material and landed cost.
  - Compute for PI Hidden Profit cost = PI Vendor Cost - (Resin Cost + PI-FG Processing cost)
  - If reason description contains "Scrap", add the cost in scrap costs (material, landed, resin, PI-FG processing, PI hidden profit, sf added value and fg added value cost).
  - If reason description contains "Request", add the cost in request costs (material, landed, resin, PI-FG processing, PI hidden profit, sf added value and fg added value cost).
  - Insert value to the report set table. (Transaction description is based from the transaction type value: **(1) B =** Cycle Count; **(2) G =** Miscellaneous Issue; and **(3) H =** Miscellaneous Receipt)
- Add total scrap and section request cost to the report set table (material, landed, resin, PI-FG processing, PI Hidden profit, sf added and value added cost).

12. Name: LSP_DM_GetScrapDataForMiscTransactionReportSp

Description: Stored procedure that get returns the SF scrap data for Miscellaneous Transaction report based on filtered date.

Parameters: (1) Start Date; and **(2)** End Date

Procedure:

- Set the initial value of the two parameters. If their value is null, set their value to the current date.
- Declare a cursor that will get and hold of the miscellaneous transactions.

Primary Source Table: jobtran

Filter conditions: (1) trans_date BETWEEN parameters start date and end date; **(2)** item starts with "SF-"; and qty_scrapped is greater than zero (0).

Joined Source Tables:

Table Name	Join Type	Join Condition
job	INNER JOIN	jobtran.job = job.job AND jobtran.suffix = job.suffix
item	INNER JOIN	job.item = item.item

- For each fetched row, execute "LSP_DM_ComputeActualMaterialCostsSp" passing job, suffix and item as input parameters, returns material, landed, resin, PI-FG processing, sf added value and PI vendor cost. Then compute PI Hidden Profit cost = PI Vendor Cost - (Resin Cost + PI-FG Processing cost). Insert values to the report table.

13. Name: LSP_DM_GetCurrentMaterialCostSp

Description: Stored procedure that get and return the current material of an item together with its standard costs (material, PI-FG processing and resin cost).

Parameters: (1) Item; and **(2)** Transaction Date

Procedure:

- Set the parameter values, if item value is null, set it to blank; if transaction date is null, set it to the current date.
- Declare a cursor that will get and hold of the materials of the item.

  Primary Source Table: item

  Joined Table: jobmatl (INNER JOIN)

  Join Conditions: item.job = jobmatl.job AND item.suffix = jobmatl.suffix

  Filter condition: (1) item from job table is equal to item parameter
- For each fetched rows, do the following:
  - Initialize value of material, PI-FG processing and resin cost variable values to zero (0).
  - If item starts with "FG-", execute "LSP_DM_GetItemStandardCostItemPricingSp" passing material and transaction date as input parameters, returns material cost.
  - If item starts with "SF-", execute "LSP_DM_GetSFMaterialStandardCostSp" passing material and transaction date as input parameters, returns material, resin and PI-FG processing cost.
  - If item starts with "PI-FG-", execute "LSP_DM_GetPIResinProcessingCostSp" passing material and transaction date as input parameters, returns resin, PI-FG processing and PI vendor cost.
  - Otherwise, execute "LSP_DM_GetItemStandardCostItemPricingSp" passing material and transaction date as input parameters, returns material cost.
  - Insert values to the report set table.

14. Name: LSP_DM_GetJobMaterialIssueWithdrawSp

Description: Stored procedure that get and return the materials used in a job (Issued material less withdrawals). This stored procedure will be used to compute the actual cost of the job.

Parameters: (1) Job; **(2)** Suffix; and **(2)** Item

Procedure:

- Declare a cursor that will get and hold of the issued and/or withdrawn materials of the job. Get the total quantity and the latest transaction date.

  Primary Source Table: job

  Joined Table: matltran (INNER JOIN)

  Join Conditions: job.job = matltran.ref_num AND job.suffix = matltran.ref_line_suf AND matltran.ref_type = 'J'

  Filter conditions: (1) job equal to job parameter; **(2)** suffix is equal to suffix parameter; and **(3)** trans_type is equal to "I" or "W".

  Group by: job, suffix, item, quantity released, material item, material location, material lot
- For each fetched rows, do the following:
  - Initialize value of material, landed, PI vendor, PI-FG processing and resin cost variable values to zero (0).
  - If item starts with "FG-", execute "LSP_DM_GetItemStandardCostItemPricingSp" passing material and transaction date as input parameters, returns material cost.
  - If item starts with "SF-", execute "LSP_DM_GetSFIssueWithdrawMaterialSp" passing lot, material and transaction date as input parameters, returns material, landed, resin, PI-FG processing and PI vendor cost.
  - If item starts with "PI-FG-", execute "LSP_DM_GetPIResinProcessingCostSp" passing material and transaction date as input parameters, returns resin, PI-FG processing and PI vendor cost.
  - Otherwise, execute "LSP_DM_GetUnitMaterialLandedCostSp" passing material, lot and transaction date as input parameters, returns material and landed cost.
  - Insert values to the report set table.

15. Name: LSP_DM_GetSFIssueWithdrawMaterialSp

Description: Stored procedure that get the actual issued/withdrawn materials of an SF item (from job) and returns the material, landed, resin, PI-FG processing and PI vendor cost of that SF.

Parameters: (1) SF Job; **(2)** SF material item; **(3)** Transaction Date; **(4)** SF material cost; **(5)** SF landed cost; **(6)** SF resin cost; **(7)** SF processing cost of PI-FG; and **(8)** SF vendor cost of PI.

Procedure:

- Set value of parameters if it is null: Set job to zero, item to blank, date to current date and all costs to zero(0).
- Get the currency rate value by executing the store procedure "LSP_ConvertUsdToPhpCurrencySp" passing the transaction date as input parameter.
- Check if the job number exists in the job table. Get count of rows from job table.
- If row is zero get the standard cost of the SF item by executing "LSP_DM_GetItemStandardCostItemPricingSp" passing the SF material item and transaction date as input parameter.
- Otherwise, use a cursor to get the issuance and withdrawal transactions of that SF job. Get the total quantity and latest transaction date.

  Primary Source Table: job

  Joined Table: matltran (INNER JOIN)

  Join Conditions: job.job = matltran.ref_num AND job.suffix = matltran.ref_line_suf AND matltran.ref_type = 'J'

  Filter conditions: (1) job equal to SF job parameter; **(2)** suffix is equal to zero (0); and **(3)** trans_type is equal to "I" or "W"

  Group by: job, suffix, item, quantity released, material item, material location, material lot

  - For each fetched rows, do the following:
    - □ Initialize value of SF material, landed, PI vendor, PI-FG processing and resin cost variable values to zero.
    - □ If item starts with "FG-", execute "LSP_DM_GetItemStandardCostItemPricingSp" passing SF item and transaction date as input parameters, returns SF material cost.
    - □ If item starts with "SF-", execute LSP_DM_GetItemStandardCostItemPricingSp SF and transaction date as input parameters, returns SF material cost.
    - □ If item starts with "PI-FG-", execute "LSP_DM_GetPIResinProcessingCostSp" passing SF item and transaction date as input parameters, returns resin, PI-FG processing and PI vendor cost.
    - □ Otherwise, execute "LSP_DM_GetUnitMaterialLandedCostSp" passing SF item, lot and transaction date as input parameters, returns material and landed cost.
    - □ If lot number contains "STOCK", get the standard item cost from the item pricing by executing the stored procedure "LSP_DM_GetItemStandardCostItemPricingSp" passing the SF item and transaction date as input parameters then set the landed, PI-FG processing, resin and PI vendor cost to zero(0).
    - □ Insert values to the report set table.
  - Compute for the SF cost by getting the sum of the product of cost and quantity then divided by the job quantity released. (Applied to SF material cost, SF landed cost, SF resin cost, SF processing cost of PI-FG and SF vendor cost of PI parts.)

16. Name: LSP_DM_GetPIResinProcessingCostSp

Description: Stored procedure that get and return the resin cost, processing cost and vendor cost of PI parts.

Parameters: (1) PI Item; **(2)** Transaction Date; **(3)** Resin cost; **(4)** Processing cost; and **(5)** Vendor cost

Procedure:

- Set value of parameters if it is null: Set item to blank, date to current date and all costs to zero(0).
- Get the currency rate by executing "LSP_ConvertUsdToPhpCurrencySp" passing transaction date parameter.
- Compute for the resin, processing and vendor cost, get column values from table.

Primary Source Table: itemvend (alias = itv)
Joined Table: itemvendprice (alias = itvp; INNER JOIN)
Join Conditions: itv.item = itvp.item AND itv.rank = 1
Filter conditions: (1) item = PI item parameter; and **(2)** effect_date is less than or equal the transaction date
Order by: effect_date DESC

Resin Cost = Uf_resin_cost * Exchange Rate
Processing Cost = Uf_process_cost * Exchange Rate
Vendor Cost = brk_cost##1 * Exchange Rate

17. Name: LSP_DM_GetItemStandardCostItemPricingSp
Description: Stored procedure that return the standard unit cost of an item.

Parameters: (1) Item; **(2)** Divisor/Multiplier; **(3)** Transaction Date; and **(4)** Standard Unit Cost.
Procedure:
- Set value of parameters if it is null: Set item to blank, DivMult to 1.2, date to current date and cost to zero(0).
- Get the currency rate by executing "LSP_ConvertUsdToPhpCurrencySp" passing transaction date parameter.
- Get the latest effective date of the item from "itemprice" table.
- If the month difference of latest effective date and transaction date is greater than or equal to six (6), set the standard unit cost to zero (0).
- Otherwise, compute for the standard unit cost. Get the unit price from "itemprice" table where effective date is less than the transaction date order by effective date in descending order.
  - If Divisor/Multiplier is equal to **1.2:**

    Standard Unit Cost = (unit_price1 column value / DivMult) * Exchange Rate
  - Otherwise**:**

    Standard Unit Cost = (unit_price1 column value * DivMult) * Exchange Rate

18. Name: LSP_DM_GetSFMaterialStandardCostSp
Description: Stored procedure that return the standard unit cost of an item.

Parameters: (1) SF Item; **(2)** Transaction Date; **(3)** SF Material Unit Cost; **(4)** SF Material Resin Cost of PI parts; and **(5)** SF Material Processing Cost of PI-FG parts.
Procedure:
- Set value of parameters if it is null: Set item to blank, date to current date and all costs to zero(0).
- Get the currency rate by executing "LSP_ConvertUsdToPhpCurrencySp" passing transaction date parameter.
- Count the material of the SF item.
- If material count is equal to zero (0), set the SF material cost as standard cost by executing store procedure "LSP_DM_GetItemStandardCostItemPricingSp" passing SF item, and transaction date as input parameters.
- Otherwise, declare a cursor that will get and hold of the materials of the SF item.

  Primary Source Table: item
  Joined Table: jobmatl (INNER JOIN)
  Join Conditions: item.job = jobmatl.job AND item.suffix = jobmatl.suffix
  Filter condition: (1) item from job table is equal to item parameter
  - For each fetched rows, do the following:
    - □ Initialize value of material, PI-FG processing and resin cost variable values to zero (0).
    - □ If item starts with "FG-", execute "LSP_DM_GetItemStandardCostItemPricingSp" passing material and transaction date as input parameters, returns material cost.

□ If item starts with "PI-FG-", execute "LSP_DM_GetPIResinProcessingCostSp" passing material and transaction date as input parameters, returns resin, PI-FG processing and PI vendor cost.
□ Otherwise, execute "LSP_DM_GetItemStandardCostItemPricingSp" passing material and transaction date as input parameters, returns material cost.
□ Insert values to the report set table.

• Compute for the SF material, resin cost of SF, and PI-FG processing cost of SF. Get from report set table.
□ SF Material Unit Cost = Sum of the product of matl_cost and matl_qty
□ SF Material Resin Cost = Sum of the product of matl_pi_resin and matl_qty
□ SF Material PI-FG Processing Cost = Sum of the product of matl_pi_process and matl_qty

19. Name: LSP_DM_ComputeStandardSFAddedValueSp
Description: Stored procedure that computes and return the standard added value cost of an SF item.

Parameters: (1) Item; **(2)** SF added value cost.
Procedure:
- Set value of item to blank it is null.
- Compute for the labor and overhead cost. (Primary source table: item)
  Labor cost = Sum of Labor hours per piece * 60 * 1.75
  Overhead cost = Labor cost * 2.5
- Set SF added value cost as the sum of labor and overhead cost.

20. Name: LSP_DM_ComputeStandardFGAddedValueSp
Description: Stored procedure that computes and return the standard added value cost of an FG item.

Parameters: (1) Item; **(2)** FG added value cost.
Procedure:
- Set value of item to blank it is null.
- Compute for the labor and overhead cost. (Primary source table: item)
  Labor cost = Sum of Labor hours per piece * 60 * 1.75
  Overhead cost = Labor cost * 2.5
- Set FG added value cost as the sum of labor and overhead cost.

21. Name: LSP_DM_ComputeStandardMaterialCostsSp
Description: Stored procedure that compute and return standard costs (material, landed, resin, PI-FG processing, SF material added value costs)

Parameters: (1) Item; **(2)** Transaction Date; **(3)** Item Material Cost; **(4)** Item Landed Cost; **(5)** Item Resin Cost; **(6)** Item PI-FG Material Processing Cost; and **(7)** Current Material SF Added Value Cost.
Procedure:
- Set value of parameters if it is null: Set item to blank, date to current date and all costs to zero(0).
- Get the current materials of the item by executing "LSP_DM_GetCurrentMaterialCostSp" passing the item and transaction date as input parameters.
- If the current material count of item is equal to zero, set item material cost as the standard cost by executing "LSP_DM_GetItemStandardCostItemPricingSp" passing the item and transaction date as input parameters.
- Otherwise, declare a cursor that will get and hold of the costs from item current materials.

- For each fetched rows, do the following:
  - □ Initialize value of SF Added value material cost to zero (0).
  - □ If the material starts with "SF-*", get the standard SF material added value cost by executing the stored procedure "LSP_DM_ComputeStandardSFAddedValueSp" passing the material as input parameter.
  - □ Insert values to the report set table, cost should be multiplied by the material quantity.
- Set values for the output parameters:
  - □ Item Material Cost = Sum of material cost
  - □ Item Resin Cost = Sum of material resin cost
  - □ Item PI-FG Processing Cost = Sum of PI-FG material processing cost
  - □ Current Material SF Added Value Cost = Sum of SF material added value cost

22. Name: LSP_DM_ComputeActualFGAddedValueSp
Description: Stored procedure that computes and return the actual added value cost of an FG item.

Parameters: (1) Job; **(2)** Suffix; and **(3)** FG added value cost.
Procedure:
- Set value of job to blank and suffix to zero(0) if they are null.
- Compute for the labor and overhead cost. (Primary source table: job_sch)
  Labor cost = Sum of Labor hours per piece * 60 * 1.75
  Overhead cost = Labor cost * 2.5
- Set FG added value cost as the sum of labor and overhead cost.

23. Name: LSP_DM_GetUnitMaterialLandedCostSp
Description: Stored procedure that computes and return the actual material and landed cost.

Parameters: (1) Material Item; **(2)** Material Lot; **(3)** Transaction Date; **(4)** Material Unit Cost; and **(5)** Material Landed Cost.
Procedure:
- Get the vendors currency code of the material from the purchase of the item.

Primary Source Table: matltran
Filter conditions: (1) item is equal to Material Item parameter; **(2)** lot is equal to Material Lot parameter; and **(3)** trans_type is equal to "R"

Joined Source Tables:

Table Name	Join Type	Join Condition
poitem	INNER JOIN	matltran.ref_num = poitem.po_num AND matltran.ref_line_suf = poitem.po_line
po	INNER JOIN	poitem.po_num = po.po_num
vendor	INNER JOIN	po.vend_num = vendor.vend_num

- If currency code is "USD", set the exchange rate by executing "LSP_ConvertUsdToPhpCurrencySp" passing the transaction date as input parameter.
- If currency code is "JPY", set the exchange rate by executing "LSP_ConvertJpyToPhpCurrencySp" passing the transaction date as input parameter.
- Otherwise, set the exchange rate to one (1).
- If material item starts with "PI-FG", set the material unit cost and material landed cost to zero (0).
- Otherwise do the following:
  - Get the receipt count of item filtered by material and lot parameters.

- If receipt count is equals to zero:
  - □ If material starts with "PDN-", set the material unit cost as matl_cost of the item with trans_type of "H" from matltran filtered by item and lot.
  - □ Otherwise, get standard cost of item by executing "LSP_DM_GetItemStandardCostItemPricingSp" passing the material item and transaction date as input parameters.
  - □ Set the material landed cost value to zero (0).
- Otherwise, get and compute the material unit cost and material landed cost:

Primary Source Table: matltran
Joined Table: po_rcpt (INNER JOIN)
Join Conditions: matltran.ref_num = po_rcpt.po_num AND matltran.ref_line_suf = po_rcpt.po_line AND matltran.trans_date = po_rcpt.rcvd_date
Filter condition: (1) item is equal to material item parameter; **(2)** lot is equal to material lot parameter; and **(3)** trans_type is equal to "R"

24. Name: LSP_DM_ComputeActualMaterialCostsSp
Description: Stored procedure that computes and return the actual costs of the job (material, landed, resin, PI-FG processing, SF Added Value and PI vendor costs)

Parameters: (1) Job; **(2)** Suffix; **(3)** Transaction Date; **(4)** Job Material Cost; **(5)** Job Landed Cost; **(6)** Job Resin Cost; **(7)** Job PI-FG Processing Cost; **(8)** Job SF Added Value Cost and **(9)** Job PI Vendor Cost.
Procedure:
- Set value of job and item to blank and suffix to zero(0) if they are null.
- Get the issuance and withdrawal transactions of the material of the job by executing the stored procedure "LSP_DM_GetJobMaterialIssueWithdrawSp" passing the job, suffix and item as input parameters. Insert the return data into a temporary variable table.
- Use a cursor to get the sum of material quantity of the jobs and the job costs (material, landed, resin, PI-Fg processing and PI vendor cost). Grouped by job, suffix, job item, quantity released and then by material item.
- For each fetch data, do the following:
  - If the fetched material item starts with "SF-", execute "LSP_DM_ComputeStandardSFAddedValueSp" to get the SF added value cost, passing the fetched material as input parameter.
  - Otherwise, set the SF added value cost to zero (0).
  - Insert values to the report set table.
- Compute for the output parameters, get values from the report set table.
  - Job Material Cost = Sum of material cost divided by Job Quantity Released
  - Job Landed Cost = Sum of landed cost divided by Job Quantity Released
  - Job Resin Cost = Sum of PI resin cost divided by Job Quantity Released
  - Job PI-FG Processing Cost = Sum of PI processing cost divided by Job Quantity Released
  - Job SF Added Value Cost = Sum of sf added value cost divided by Job Quantity Released
  - Job PI Vendor Cost = Sum of PI vendor cost divided by Job Quantity Released

25. Name: LSP_DM_GetRMForMicellaneousTransactionSp
Description: Stored procedure that gets the item, lot, date and cost of the miscellaneous transactions and return the material and landed unit cost.

Parameters: (1) Material Item; **(2)** Material Lot Number; **(3)** Transaction Date; **(4)** Material Transaction Cost; **(5)** Material Unit Cost; and **(6)** Landed Cost.

Procedure:

- Get the receipt count of item filtered by material and lot parameters.
- If the receipt count is zero, do the following:
  - If material starts with "PDN-", set the material unit cost as matl_cost of the item with trans_type of "H" from matltran filtered by item and lot parameters.
  - Otherwise, set the material unit cost as the material transaction cost parameter.
  - Set the landed cost value to zero (0).
- Otherwise, get and compute the material unit cost and material landed cost:

Primary Source Table: matltran
Joined Table: po_rcpt (INNER JOIN)
Join Conditions: matltran.ref_num = po_rcpt.po_num AND matltran.ref_line_suf = po_rcpt.po_line AND matltran.trans_date = po_rcpt.rcvd_date
Filter condition: (1) item is equal to material item parameter; **(2)** lot is equal to material lot parameter; and **(3)** trans_type is equal to "R"

II. REPORT FILES

1. Name: LSP_Rpt_DM_FinishedGoodsSalesReport.rdl
Description: Report file that can print the Finished Goods, Sales and Sample JO Reports.

Datasource
Name: FG_Datasource
Embedded Connection:
Type: Miscrosoft SQL Server
Connection String: [@pConnectionString]

Datasets
Name: DS_FGReport
Data source: FG_Datasource
Query Type: Stored Procedure
Stored Procedure Name: LSP_Rpt_DM_FinishedGoodsReportSp
Parameters: (1) Start Date; (2) End Date

Name: DS_SalesReport
Data source: FG_Datasource
Query Type: Stored Procedure
Stored Procedure Name: LSP_Rpt_DM_SalesSampleJOReportSp
Parameters: (1) Start Date; (2) End Date

Report Layout Description

- Add table for the Finished Goods part. Dataset source of the table is "DS_FGReport" and set the Page Name to "FINISHED GOODS".
  - Display details of the Job showing the following columns: date, PO Number, destination, JO#, JO Suffix, item, item description, product code, family description, completed quantity, Standard & Actual Cost in PHP currency (material cost, resin cost, PI-FG processing, PI Hidden Profit,SF Added Value, FG Added Value, Total Standard or Actual Cost), Cost Difference in PHP currency (unit difference, unit difference without landed cost, and Total Difference of Actual & Standard Cost).

- Add row group adjacent after the details. Group it per family code. Display the family code and family description, then the sum of quantity completed and costs per family.
- Add row group adjacent after the family group. Group it per product code. Display the product code, then the sum of quantity completed and costs per product codes.

- Add table for the Sales and Sample JO part. Dataset source of the table is "DS_SalesReport"
  - Add parent group for the details. Group the details by ship category (if it is under sales or sample JO sheet). set the page name of the group to whatever the value of "ship category" is.
  - Display details of the sales showing the following columns: date, PO Number, destination, JO#, JO Suffix, item, item description, product code, family description, shipped quantity, Sales Price in USD and PHP currency, Standard & Actual Cost in PHP currency (material cost, resin cost, PI-FG processing, PI Hidden Profit, SF Added Value, FG Added Value, Total Standard or Actual Cost), Cost Difference in PHP currency (unit difference, unit difference without landed cost, and Total Difference of Actual & Standard Cost). For sales, display also if the job is recoverable or not and the job remarks.
  - Add row group adjacent after the details. Group it per family code. Display the family code and family description, then the total sales quantity, sales price and costs per family.
  - Add row group adjacent after the family group. Group it per product code. Display the product code, then the total sales quantity, sales price and costs per product code.

2. Name: LSP_Rpt_DM_DirectMaterialLaborPercentageReport.rdl
Description: Report file that can print the Direct Material and Labor Percentage Report.

Datasource
Name: DMLabor_Datasource
Embedded Connection:
Type: Miscrosoft SQL Server
Connection String: [@pConnectionString]

Datasets
Name: DS_DMLabor
Data source: DMLabor_Datasource
Query Type: Stored Procedure
Stored Procedure Name: LSP_Rpt_DM_DirectMaterialLaborPercentageReportSp
Parameters: (1) Start Date; (2) End Date; (3) Start Product Code; (4) End Product Code; (5) Start Model; (6) End Model

Report Layout Description
- Add table for the display of the report for DM and Labor Percentage per Product Model. Set the table page name to "ProductModel".
  - Add parent row group, grouped by "item" code, displaying the product model and family code columns.
  - Add child row group, grouped by year and then my month of transaction, displaying the month and year of the transaction.
  - Display details showing the sum of the following columns per model: quantity completed, produced amount, RM & Sales Percentage, total RM cost (actual & standard), percentage of actual and standard cost, actual RM cost per piece, total assembly cost (actual & standard), percentage of total actual and standard cost.
- Add table for the display of the report for DM and Labor Percentage per Product Code. Set the table page name to "ProductCode".
  - Add parent row group, grouped by "product code", displaying the product code and family code columns.

- Add child row group, grouped by year and then my month of transaction, displaying the month and year of the transaction.
- Display details showing the sum of the following columns per model: quantity completed, produced amount, RM & Sales Percentage, total RM cost (actual & standard), percentage of actual and standard cost, actual RM cost per piece, total assembly cost (actual & standard), percentage of total actual and standard cost.

- **NOTE** that the "FG-" prefix should remove in the display of product model and product code values.

3. Name: LSP_Rpt_DM_WIPShopFloorReport.rdl
Description: Report file that can print the WIP shop floor report.

Datasource
Name: WIP_Datasource
Embedded Connection:
Type: Miscrosoft SQL Server
Connection String: [@pConnectionString]

Datasets
Name: DS_WIPShopFloor
Data source: WIP_Datasource
Query Type: Stored Procedure
Stored Procedure Name: LSP_Rpt_DM_WIPShopFloorReportSp

Report Layout Description
- Add table for the display of the report for DM and Labor Percentage per Product Model.
  - Display the details with the following columns: transaction date, item issued (material and material description), reference job (JO# and suffix), quantity WIP, WIP unit cost in PHP currency (material cost, landed cost, resin cost, PI-FG processing, PI Hidden Profit) and the total WIP cost.

4. Name: LSP_Rpt_DM_InventoryTurnOverReport.rdl
Description: Report file that can print the Inventory Turnover report.

Datasource
Name: InvTurn_Datasource
Embedded Connection:
Type: Miscrosoft SQL Server
Connection String: [@pConnectionString]

Datasets
Name: DS_InvTurn
Data source: InvTurn_Datasource
Query Type: Stored Procedure
Stored Procedure Name: LSP_Rpt_DM_InventoryTurnOverReportSp
Parameter: Details (with detailed transaction)

Report Layout Description
- Add table for the display of the report for Inventory turnover report usage sheet. Set the page name to "Usage". Add filter in the table, should only show row where report group is "USAGE".

- Add parent row group, group by product code and display the column value.
- Add parent column group, grouped by month of transaction date, displaying the total material cost and landed cost per month and per product code.
- Outside the column group, display the following with material cost and landed cost columns: Average Usage, Maximum usage, 3 months usages, maximum 3 months usage, and current inventory. The following columns doesn't need to display separately the material and landed cost: Safety Stock Cost, Ideal Inventory, Age of Inventory, and Excess Inventory.
- If the value of "Age of Inventory" is greater than four(4), change the font color of "Age of Inventory" and "Excess Inventory" values into red.

- Add table for the display of the report for Inventory turnover report usage details sheet. Set the page name to "DETAILS". Add filter in the table, should only show row where report group is "DETAILED". This table will ony visible if the "Details" parameter value is equal to one (1) or True.
  - Add parent row group, group by RowNumber over 60,000 to split the details 60,000 rows per sheet. Do this to avoid problem in exporting the report in MS Excel file format.
  - Display the following column details: transaction date, transaction type, reason code, reason description, item, item description, product code, reference (reference number and suffix), quantity and total amount in PHP currency (showing material cost and landed cost).

5. Name: LSP_Rpt_DM_MiscellaneousTransactionReport.rdl
Description: Report file that can print the Miscellaneous Transaction report.

Datasource
Name: MiscTrans_Datasource
Embedded Connection:
Type: Miscrosoft SQL Server
Connection String: [@pConnectionString]

Datasets
Name: DS_MiscTrans
Data source: MiscTrans_Datasource
Query Type: Stored Procedure
Stored Procedure Name: LSP_Rpt_DM_MiscellaneousTransactionReportSp
Parameters: (1) Start Date; (2) End Date

Report Layout Description
- Add table for the display of the summary of transactions. Set the page name to "Summary".
  - Add row groups, for the display of the transaction summary. Group data per transaction type, then by miscellaneous transaction class, reason description and lastly by work center.
  - Display the sum of the following columns per reason description: material cost, landed cost, resin cost, PI-FG processing cost, PI Hidden Profit, Added Value Cost and Total Cost in PHP currency.
  - For SF scrap reason, should also display the summary per work center.
- Add table for the display of the transaction details.
  - Add parent row group for transaction details. Set page break between each instance of a group and set the page name to whatever the value of transaction description. Sort the group by transaction description and then by transaction date.
- Add table for the display of the SF Scrap summary sheet. Filter row data, where transaction description is equal to "SF Scrap Data".

- Add parent row group, grouped by work center. Show the sum of the following columns per work center: material cost, landed cost, resin cost, PI-FG processing, PI Hidden Profit, Added Value and Total Cost in PHP currency.
- Add row group adjacent, grouped by work center and then by item code. Show details per item and sum per work center of the following columns: scrapped quantity, unit cost in PHP currency (material cost, landed cost, resin cost, PI-FG processing, PI hidden profit and added value), and total cost in PHP currency. Highlight in color yellow the total per work center.

- Add table for the display of the SF Scrap Data. Filter row data, where transaction description is
  - Show details with the following columns: JO#, transaction date, item, item description, completed quantity, scrapped quantity, employee number, work center, unit cost in PHP curreny (material cost, landed cost, resin cost, PI-FG processing, PI hidden profit and added value), and total cost in PHP currency.

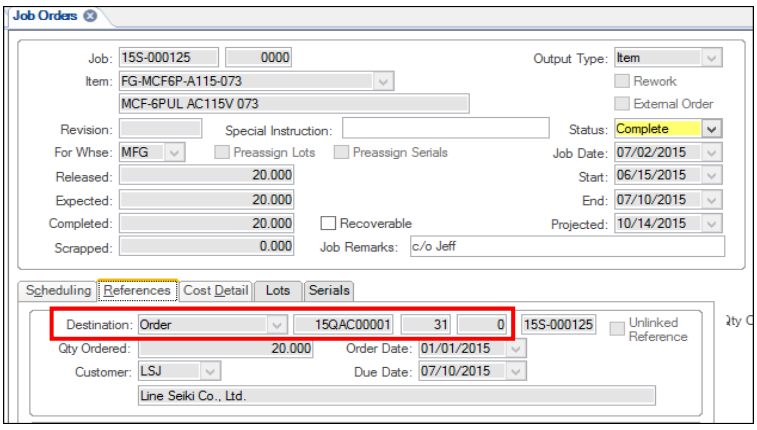
CONSLUSIONS AND RECOMMENDATIONS

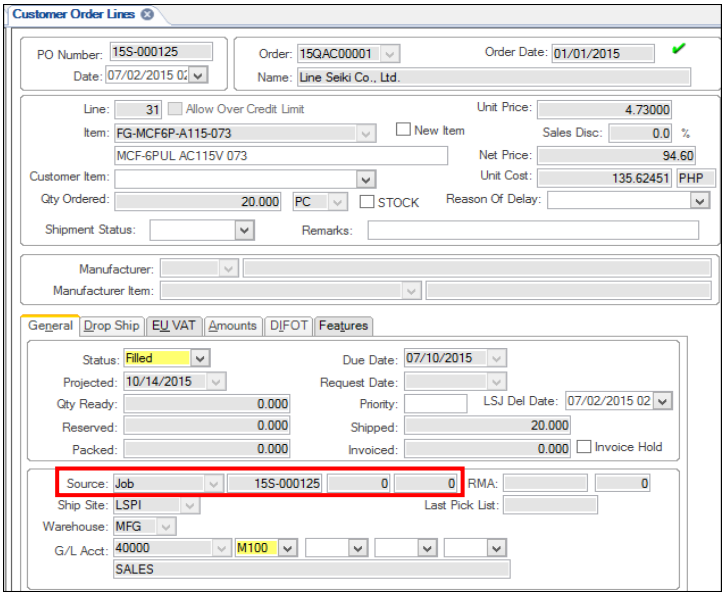
- All items (RM, SC, SF, PI-FG, FG, PDN, etc.) should have cost in "Item Pricing" that changes every six months. Direct Material reporting will use the cost in Item Pricing for the standard material costs.

- Determining the standard cost from item pricing is based on effective date compared to the transaction date from material transactions (for finish, ship, issue and withdrawal transactions). But limit to six months difference only between the effective date and transaction date. If the difference of the two dates are more than six months, standard cost will become zero (0). For example, transaction date is August, but the latest effective date of item pricing is January, the standard cost will be zero (0) because standard of EX Works price should change semi annualy (every January and July). Therefore, for transactions made from January to June, the standard cost will be based on Item Pricing where effective date is January, then July for all transactions from July to December.

- If the report was already generated before the update in Item Pricing, user should need to re-generate or reprint the reports in order to update the standard cost of items. Likewise, if there were updates in the Item Pricing.

- Finished goods are based from job orders, to get its PO numbers, it is linked to the customer order lines based on the destination reference from job orders form. Sales report is from material transactions of the customer order lines, then the source reference in customer order is link to job orders. "Destination" reference in job orders form and "Source" reference in customer order lines should not be change or "Unlinked" if not needed to avoid mismatch of the job to its PO. The source and destination reference is used to get and compute the actual and standard costs.





- Added value cost is based on the labor hours per piece from the current operations for the standard cost and job operations for the actual cost. Standard and actual cost may differ if the planner or user change the labor hour per piece in job operations. Upon creating the job, the default values in job operations are the values from current operations of the item. **NOTE:** The standard cost may also differ based on when the report was generated. If user generate the report, then later on there were changes in current operations, when another user generated the reports and compare it to the previously generated report, there were really variances.

- When receiving SF jobs, the lot number value should be the source Job Order number. The actual cost of SF materials when issued to FG jobs is based on the lot number of the issued SF material. If the source lot of the issued SF materials were not found as JO number in jobs, it will use the standard cost from Item Pricing.

- To get the actual material and landed cost of RM items, it based on the lot number as the source of issued materials. It will then get the PO receipt transaction based on the lot number and item code of the material. The landed cost that will get is the estimated cost allocated to the item upon performing the PO receiving. If the source lot does not have PO receipt transaction and/or if the source lot is "0-STOCK" (this lot happens during physical inventory), the actual cost will become the standard cost of the material from the Item Pricing form.

- Accuracy of the Finished Goods report is based on the Job Receiving of the FG warehouse. Ex. a job was received in FG warehouse physically and was transacted the following day in ERP, the report may not be accurate if the month of receiving in the system and in physical is different.

- Accuracy of the Sales report is based on the transaction of the impex officer. There might be an issue with the sales report if there were returned shipment transactions due to user error, especially if the month of the transaction of shipping and return are different, the cost of those transactions will be posted in different month also.

Prepared by:

Dan Olivier V. Arca
Name/Signature/Date

Noted by:

Mildred Vendiola	Luz Cornejo	Julius Colega
Name/Signature/Date	Name/Signature/Date	Name/Signature/Date
Raquel Jose	Gina Gregorio	
Name/Signature/Date	Name/Signature/Date	