Description: This documentation contains the specifications of the LSP Customized Report – Direct Material Reports that includes: (1) Finished Goods and Sales Report; (2) Direct Material and Labor Percentage Report; (3) RM Breakdown of FG and Sales report; (4) RM Beginning Balance Report; (5) RM Breakdown for Specific JO Report; (6) Slow Moving Analysis Report; (7) WIP Shop Floor Report; (8) Inventory Turnover Report and (9) Miscellaneous Transaction Report

1. **Stored Procedure Name:** LSP_ERPReport_GetFGItemListPerProdCodeWihtNullSp
   Definition: Returns the list of FG item code based on Product Code filter
   Parameters: (1) @StartProdCode; (2) @EndProdCode; (3) @Search
   Procedures:
   1. Set default value for all parameters if values are NULL. @StartProdCode = dbo.LowCharacter(); @EndProdCode = dbo.Character(); @Search = empty character
   2. Select values:

Table Name:	item
SELECT Columns:	item, description
WHERE Conditions:	item LIKE 'FG-%'
AND	product_code BETWEEN @StartProdCode AND @EndProdCode
AND	product_code LIKE '%'+@Search+'%'

   UNION Empty strings for item and description

2. **Stored Procedure Name:** LSP_fn_GetCurrencyConversion
   Definition: Returns the converted amount based on inputted currencies and transaction date
   Parameters: (1) @TransDate; (2) @FromCurrency; (3) @ToCurrency; (4) @Amount;
   Procedures:
   1. Set default value for all parameters if values are NULL. @TransDate = Current Date; @FromCurrency = 'PHP'; @ToCurrency = 'PHP'; @Amount = 0; @ConvertedAmount = 0; @ExchangeRate = 0
   2. Check if FromCurrency and ToCurrency code are "Divisor" type currencies
   3. If FromCurrency and ToCurrency code are equal, set the ExchangeRate = 1 and ConvertedAmount = Amount parameter; otherwise get the the "sell_rate" from **currate** table filtering out the from_curr_code, to_curr_code and eff_date
   4. Set value for the Converted Amount based on below conditions:
      - IF @ToIsDivisor=1 AND @FromCurrency='PHP' ➔ (@Amount / NULLIF(@ExchangeRate, 0))
      - ELSE IF @FromIsDivisor = 1 AND @ToCurrency = 'PHP' ➔ @Amount * NULLIF(@ExchangeRate, 0)
      - ELSE IF @ToIsDivisor = 1 AND @FromCurrency <> 'PHP' ➔@Amount * @ExchangeRate)
      - ELSE IF @FromIsDivisor = 1 AND @ToCurrency <> 'PHP' ➔@Amount / @ExchangeRate
      - ELSE IF @ToIsDivisor <> 1 AND @FromCurrency = 'PHP' ➔@Amount * NULLIF(@ExchangeRate, 0)
      - ELSE IF @ToIsDivisor<>1 AND @FromCurrency <> 'PHP' ➔@Amount / NULLIF(@ExchangeRate, 0)
      - ELSE ➔ @ConvertedAmount = @Amount
   5. RETURN value of the @ConvertedAmount

3. **Stored Procedure Name:** LSP_NewDM_GetAllRMProductCodesGroupedSp
   Definition: Returns the list of RM product code to be used in the selection of product code dropdown
   Procedures:
   1. Get product code values, removing the prefix values of "RM-" and "SA-" adding "ALL" options

Table Name:	prodcode
SELECT Columns:	REPLACE(REPLACE(product_code, 'RM-', ''), 'SA-', '') AS ProductCode , REPLACE(REPLACE(product_code, 'RM-', ''), 'SA-', '') AS [Description]
WHERE Conditions:	product_code starts with "RM-", "SA-", "FG-PI", "PS-RM"

4. **Stored Procedure Name:** LSP_StdCost_GetMatlCostingSp
   Definition: Stored procedure to returns the standard cost of the specified material.
   Input Parameters: (1) @Matl = item code of the material; (2) @TransDate = transaction date of the material
   Output Parameters: (1) @MatlUnitCost; (2) @PIProcessCost; (3) @PIResinCost; (4) @PIHiddenProfit; (5) @SFLbrCst; (6) @SFOvhdCst; (7) @FGLbrCst; and (8) @FGOvhdCst
   Procedures:
   1. If material is PI parts (item vendor is equal to "LPI0001"), set PI Resin Cost, PI Process Cost and PI Vendor cost from "Vendor Contract Prices". Compute for the PI Hidden Profit (PI Vendor Cost – (Process Cost + Resin Cost))
   2. If material is not SF, FG or PI parts, get material cost from item pricing filtered by effectivity date and input parameter transaction date. If item price currency is "USD", divide it by 1.2 otherwise convert the price into USD.
   3. If material is SF or FG, get first the labor rate and overhead rate from "LSP Labor and Overhead Rates" based on effectivity date filtered by the input transaction date. Get the total "Labor hours per piece" of the item from "Current Operations" multiple by 60 multiply by the labor rate to get the total labor cost. Multiply the labor cost to overhead rate to get the total overhead cost. If material is SF, set it to "SF Labor Cost" and "SF Overhead Cost" otherwise set it to "FG Labor Cost" and "FG overhead cost"

5. **Stored Procedure Name:** LSP_DM_StdCost_GetCurrentMatlCostingSp
   Definition: Get the standard BOM costing of the item. It gets the current material of the item, compute and returns it standard cost values.
   Parameters: (1) @Item; (2) @TransDate
   Procedures:
   1. Insert first the detail of parent item and the first level BOM of the item. Get it from "Current Materials"
   2. Breakdown each SF or FG materials of the first level BOM, get the BOM of the said item from "Current Materials", repeat the same step until all SF and FG parts were breakdown.
   3. For each material (up to the last RM), execute the stored procedure LSP_StdCost_GetMatlCostingSp to get the standard cost of each material populating the correct cost field (material costs, PI costs, SF costs or FG costs)
   4. To get the current material standard cost of SF, get the material unit cost of that SF multiply by the quantity used based on BOM (current material). If the SF material has a SF sub material, get first the cost (unit cost x quantity) of the last RM adding it up to the highest SF material.

6. **Stored Procedure Name:** LSP_ActlCost_GetMatlCostingSp
   Definition: Stored procedure to returns the actual cost of the specified material.
   Input Parameters: (1) @matl_item = item code of the material; (2) @matl_lot = lot number of the material where it was issued from; and (3) @matlTransDate = transaction date of the material
   Output Parameters: (1) @JobQty; (2) @matl_unit_cost_usd; (3) @matl_landed_cost_usd; (4) @pi_fg_process_usd; (5) @pi_resin_usd; (6) @pi_vend_cost_usd; (7) @pi_hidden_profit_usd; (8) @sf_lbr_cost_usd; (9) @sf_ovhd_cost_usd; (10) @fg_lbr_cost_usd; (11) @fg_ovhd_cost_php; (12) @matl_unit_cost_php; (13) @matl_landed_cost_php; (14) @pi_fg_process_php; (15) @pi_resin_php; (16) @pi_vend_cost_php; (17) @pi_hidden_profit_php; (18) @sf_lbr_cost_php; (19) @sf_ovhd_cost_php; (20) @fg_lbr_cost_php; (21) @fg_ovhd_cost_php;
   Procedures:
   1. If material is PI parts (item vendor is equal to "LPI0001"), set PI Resin Cost and PI Process Cost from "Vendor Contract Prices". Set material landed cost to zero (0). Perform currency conversion if PI resin cost and PI process cost from USD to PHP. Get the PI vendor cost from material transaction receipts based on the lot number and item code (amount is in PHP). Perform currency conversion from PHP to USD. Compute for the PI Hidden Profit (PI Vendor Cost – (Process Cost + Resin Cost)) for both USD and PHP currency costs.

2. If material is not SF, FG or PI parts, get material cost and landed cost from material transaction receipts based on the lot number and item code (if currency is USD, set amount to USD costs then convert to PHP for PHP costs; if currency is PHP, set amount to PHP costs then convert to USD for USD costs; if currency is JPY, convert and set amount into PHP and USD). If there were no PO receipt transactions, check if it has miscellaneous transactions (Misc. Receipts, Cycle Count or Physical Inventory), then get its material cost in PHP. Perform currency conversion from PHP to USD for material cost, set landed cost into zero (0). If there were no PO receipt and miscellaneous transactions, get the standard price: Set material unit price USD get from item price divide it by 1.2, perform currency conversion to PHP to set the material unit price in PHP.

3. If material is FG, check if lot number exists as Job Order number. If yes, get first the labor rate and overhead rate from "LSP Labor and Overhead Rates" based on effectivity date filtered by the input transaction date. Get the total "Labor hours per piece" of the item from "Job Operations" of the multiple by 60 multiply by the labor rate to get the total labor cost. Multiply the labor cost to overhead rate to get the total overhead cost. Set value for FG labor cost and overhead cost in USD currency. Identify the item lot creation date, get it from material transactions (FG receipts) or from miscellaneous transactions (Misc. Trans., Cycle Count or Physical Inventory). Perform currency conversion for labor cost and overhead cost into PHP currency using the lot creation date to get the currency rate. If lot number is not a JO number, identify if it has PO receipt transactions. If yes, get the receipt cost as material cost in PHP, then perform currency conversion into USD. Otherwise, use the standard cost of FG (get it from item price divided by 1.2 to get the unit cost in USD) check if it has miscellaneous transactions receipts, if yes set it as the material transaction date, then perform currency conversion into PHP currency.

4. If material is SF, check if lot number exists as Job Order number. If yes, get first the labor rate and overhead rate from "LSP Labor and Overhead Rates" based on effectivity date filtered by the input transaction date. Get the total "Labor hours per piece" of the item from "Job Operations" of the multiple by 60 multiply by the labor rate to get the total labor cost. Multiply the labor cost to overhead rate to get the total overhead cost. Set value for SF labor cost and overhead cost in USD currency. Identify the item lot creation date, get it from material transactions (Job receipts) or from miscellaneous transactions (Misc. Trans., Cycle Count or Physical Inventory). Perform currency conversion for labor cost and overhead cost into PHP currency using the lot creation date to get the currency rate. If lot number is not a JO number, identify if it has miscellaneous receipt transactions. If yes, get the receipt cost as material cost in PHP, then perform currency conversion into USD. Otherwise, use the standard cost of SF (get it from item price divided by 1.2 to get the unit cost in USD) check if it has miscellaneous transactions receipts, if yes set it as the material transaction date, then perform currency conversion into PHP currency.

7. **Stored Procedure Name:** LSP_DM_ActlCost_GetJobMatlTransCostingSp
   Definition: Get and returns the actual BOM costing of the item and JO from job material transactions (issuance and withdrawal transactions. It gets the job material of the item, compute and returns it actual cost values.
   Parameters: (1) @Job; (2) @Suffix; (3) @Item; (4) @JobTransDate; and (5) QtyTrans
   Procedures:
   1. Insert first the detail of parent item and the first level BOM of the item. Get it from "Material Transactions" where transaction type are "Issuance" and "Withdrawal", get the sum per item.
   2. Breakdown each SF or FG materials of the first level BOM, get the BOM of the said item from "Material Transactions" where transaction type are "Issuance" and "Withdrawal", get the sum per item. Repeat the same step until all SF and FG parts were breakdown.
   3. For each material (up to the last RM), execute the stored procedure LSP_ActlCost_GetMatlCostingSp to get the actual cost of each material populating the correct cost field (material costs, PI costs, SF costs or FG costs)

4. To get the actual material cost of SF, get the material unit cost of that SF multiply by the quantity used based on BOM (current material). If the SF material has a SF sub material, get first the cost (unit cost x quantity) of the last RM adding it up to the highest SF material.

FINISHED GOODS AND SALES REPORT

1. **Stored Procedure Name:** LSP_Rpt_NewDM_FinishedGoodsReportSp
   Definition: Get and returns the finished goods report with the actual and standard cost based on the filtered start and end date transactions. Actual cost and standard cost are computed up to the last RM of the finished goods item.
   Parameters: (1) @StartDate; and (2) @EndDate
   Procedures:
   1. Declare and create temporary tables for the storage of the item standard cost, item actual cost, FG receipt transactions, shipment transactions and finished goods report final table.
   2. Set default value for @StartDate and @EndDate if "NULL" to current date.
   3. Insert data into variable table @ship_tran, get "Ship" transactions from material transaction where transaction date is between the @StartDate and @EndDate, transaction type is "Ship", reference type is "Customer Order" and item starts with "FG-"
   4. Insert data into FG receipts table, get from material transactions where transaction date is between @StartDate and @EndDate parameters, transaction type is "FG Receipts", reference type is "Job", quantity is greater than zero and item starts with "FG".
   5. For each item in the FG receipts transaction table, execute the following:
      i. Execute stored procedure LSP_DM_StdCost_GetCurrentMatlCostingSp passing the item and transaction date as parameters, insert into standard cost temporary table.
      ii. Set the FG item standard cost, get from temporary table where "Level" is zero (0). This will be the item standard cost in USD currency. Multiply it to the exchange rate based on transaction date, this will be the standard cost in PHP currency.
      iii. Execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the job order number, JO suffix, item, transaction date and quantity completed as parameters, insert into actual cost temporary table
      iv. Set the FG item actual cost, get from temporary table where "Level" is zero (0) divided by the job order qty. This will be the actual unit cost in PHP currency, needs to multiply it to FG receipt quantity to get the actual cost based on the FG receipt transaction.
      v. Set for the EXWorks unit price, get it from item price divided by 1.2. If the item price currency code is not PHP, perform currency conversion from EXworks price currency into "PHP"
      vi. Insert into FG report final table.

2. **Stored Procedure Name:** LSP_Rpt_NewDM_SalesAndSampleJOReportSp
   Definition: Get and returns the sales and sample JO report with the actual and standard cost based on the filtered start and end date transactions. Actual cost and standard cost are computed up to the last RM of the finished goods item.
   Parameters: (1) @StartDate; and (2) @EndDate
   Procedures:
   1. Declare and create temporary tables for the storage of the item standard cost, item actual cost, shipment transactions and sales report final table.
   2. Set default value for @StartDate and @EndDate if "NULL" to current date.
   3. Insert data into variable table @ShipTrans, get "Ship" transactions from material transaction where transaction date is between the @StartDate and @EndDate, transaction type is "Ship", reference type is "Customer Order" and delivery order status is "Approved". Union data for sample JO data, get from material transactions where transaction date is between the @StartDate and @EndDate, transaction type is "Ship", reference type is "Customer Order" and customer name contains "LSP*SECTION"

4. For each item in the shipment transaction table, perform the following:
   i. Check if reference number from material transaction exists as JO number from Job Orders. Set IsRecoverable and JobRemarks from the Job Orders data.
   ii. If reference number exists as JO number:
      a. Execute stored procedure LSP_DM_StdCost_GetCurrentMatlCostingSp passing the item and transaction date as parameters, insert into standard cost temporary table.
      b. Set the FG item standard cost, get from temporary table where "Level" is zero (0). This will be the item standard cost in USD currency. Multiply it to the exchange rate based on transaction date, this will be the standard cost in PHP currency.
      c. Execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the job order number, JO suffix, item, transaction date and quantity completed as parameters, insert into actual cost temporary table
      d. Set the FG item actual cost, get from temporary table where "Level" is zero (0) divided by the job order qty. This will be the actual unit cost in PHP currency, needs to multiply it to FG receipt quantity to get the actual cost based on the FG receipt transaction.
   iii. If reference number not exists as JO number (possible RM or if FG or SF, there is no JO reference), execute stored procedure to get the standard and actual material costs:
      a. Execute stored procedure LSP_StdCost_GetMatlCostingSp to get the standard material cost based on the shipped item
      b. Execute stored procedure LSP_ActlCost_GetMatlCostingSp to get the actual material cost based on the shipped item
      c. Compute for the Standard and actual SF Added and FG added values (this will just be in case that the item shipped was SF or FG and there's no reference JO number in material transactions)
   iv. Set for the EXWorks unit price, get it from item price divided by 1.2. If the item price currency code is not PHP, perform currency conversion from EXworks price currency into "PHP"
   v. Insert into FG report final table.

3. **Stored Procedure Name:** LSP_Rpt_NewDM_SalesSummaryReportSp
   Definition: Get and returns the sales summary report from A/R transactions.
   Parameters: (1) @StartDate; and (2) @EndDate
   Procedures:
   1. Set default value for @StartDate and @EndDate if "NULL" to current date.
   2. Insert value for sales summary report, get from A/R posted transaction data where invoice date is between @StartDate and @EndDate and type is not equal to "Payment"
      * Invoice Description values: C = Credit Memo, D = Debit Memo, with DO number reference = Products, otherwise "Product Eng' Design"
   3. Select sales summary report order by invoice date in ascending order.

 DIRECT MATERIAL AND LABOR PERCENTAGE REPORT

1. **Stored Procedure Name:** LSP_NewDM_GetFilteredFinishedGoodsTransactionSp
   Definition: Get and returns the finished good receipts transaction based on the filtering options.
   Parameters: (1) @StartDate; (2) @EndDate; (3) @StartProdCode; (4) @EndProdCode; (5) @StartModel; and (6) @EndModel
   Procedures:
   1. Set default values for parameters: @StartDate and @EndDate if "NULL" to current date; @StartProdCode and @EndProdCode to first and last product code options from Product Code lists; @StartModel and @EndModel to the first and last item from Item lists based on @StartProdCode and @EndProdCode filter.

2. Insert data into temporary table, get from material transactions where transaction date is between @StartDate, @EndDate, quantity is greater than zero (0), "Finish" transaction type (FG Receipts), reference type is "Job", item starts with "FG-", product code between @StartProdCode and @EndProdCode, and item is between @StartModel and @EndModel.
   * If there were multiple transactions in one month, get the total transactions to show one line only getting the latest as it's transaction date.

2. **Stored Procedure Name:** LSP_Rpt_NewDM_DirectMaterialLaborPercentageReportSp
   Definition: Get and returns the direct material and labor percentage report based on filtering options of date, product code and product model.
   Parameters: (1) @StartDate; (2) @EndDate; (3) @StartProdCode; (4) @EndProdCode; (5) @StartModel; and (6) @EndModel
   Procedures:
   1. Declare and create temporary tables for the storage of the item standard cost, item actual cost and report final table.
   2. Set default values for parameters: @StartDate and @EndDate if "NULL" to current date; @StartProdCode and @EndProdCode to first and last product code options from Product Code lists; @StartModel and @EndModel to the first and last item from Item lists based on @StartProdCode and @EndProdCode filter.
   3. Execute stored procedure LSP_NewDM_GetFilteredFinishedGoodsTransactionSp passing up the parameters needed.
   4. For each item in the FG receipts transaction table, execute the following:
      a. Execute stored procedure LSP_DM_StdCost_GetCurrentMatlCostingSp passing the item and transaction date as parameters, insert into standard cost temporary table.
      b. Set the FG item standard cost, get from temporary table where "Level" is zero (0). This will be the item standard cost in USD currency. Multiply it to the exchange rate based on transaction date, this will be the standard cost in PHP currency.
      c. Execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the job order number, JO suffix, item, transaction date and quantity completed as parameters, insert into actual cost temporary table
      d. Set the FG item actual cost, get from temporary table where "Level" is zero (0) divided by the job order qty. This will be the actual unit cost in PHP currency, needs to multiply it to FG receipt quantity to get the actual cost based on the FG receipt transaction.
      e. Set for the EXWorks unit price, get it from item price divided by 1.2. If the item price currency code is not PHP, perform currency conversion from EXworks price currency into "PHP"
      f. Insert into report final table.

   RM BREAKDOWN REPORTS

1. **Stored Procedure Name:** LSP_Rpt_NewDM_RMBreakdownPerJOSp
   Definition: Get and returns the raw material (RM) breakdown of a specific JO or PO number. This will get the breakdown up to the last RM and the material cost based on the completed or job order quantity.
   Parameters: (1) @JobOrder; and (2) @PONumber; and (3) Quantity
   Procedures:
   1. Declare and create temporary tables for the storage of the item standard cost, item actual cost and RM breakdown report final table.
   2. Set and get values for job order number, PO number, job suffix, job item and job quantity from job order list using the filter @JobOrder or @PONumber parameters. Set @JobDate as the latest transaction date from job material transactions where transaction type is "Job Finish".
   3. Set FG standard and actual labor hours as the total labor hours per piece of the job order multiply by the job quantity released.

4. Insert into item standard cost, get values by executing the stored procedure LSP_DM_StdCost_GetCurrentMatlCostingSp passing the job item and job date as input parameters.
5. Insert into item actual cost, get values by executing the stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the job order, job suffix, job item, job date and job quantity released as input parameters.
6. Insert values into the RM breakdown final table. (1) For level 0 (parent items), directly insert it from standard and actual item costs temporary tables. (2) For level 1 (immediate child), insert it from standard and actual item costs temporary tables, getting the quotient of material cost and actual material quantity. (3) For level 2 and lower (child RM of parent item e.g., SF or FG materials), insert it from standard and actual item costs temporary tables, getting the ratio and proportion of the issued material and the immediate parent item issued material and job quantity.

2. **Stored Procedure Name:** LSP_Rpt_NewDM_RMBreakdownFinishedGoodsSp
   Definition: Get and returns the raw material (RM) breakdown of finished goods report based on the date range. This will get the breakdown up to the last RM and the material cost based on the completed or job order quantity.
   Parameters: (1) @StartDate; and (2) @EndDate
   Procedures:
   1. Declare and create temporary tables for the storage of ship_tran, FG receipts and RM breakdown report final table.
   2. Insert data into variable table @ship_tran, get "Ship" transactions from material transaction where transaction date is between the @StartDate and @EndDate, transaction type is "Ship", reference type is "Customer Order" and item starts with "FG-".
   3. Insert data into FG receipts table, get from material transactions where transaction date is between @StartDate and @EndDate parameters, transaction type is "FG Receipts", reference type is "Job", quantity is greater than zero and item starts with "FG".
   4. For each job order number in the FG receipts transaction table, perform the following:
      a. Execute stored procedure LSP_Rpt_NewDM_RMBreakdownPerJOSp passing the job order number, PO number and quantity completed as input parameters.
      b. Insert into RM breakdown report final table.
   5. Select data from RM breakdown report final table where [Level] not equal to zero (0).

3. **Stored Procedure Name:** LSP_Rpt_NewDM_RMBreakdownSalesSp
   Definition: Get and returns the raw material (RM) breakdown of sales report based on the date range. This will get the breakdown up to the last RM and the material cost based on the completed or job order quantity.
   Parameters: (1) @StartDate; and (2) @EndDate
   Procedures:
   1. Declare and create temporary tables for the storage of the shipment transactions and RM breakdown sales report final table.
   2. Set default value for @StartDate and @EndDate if "NULL" to current date.
   3. Insert data into variable table @ShipTrans, get "Ship" transactions from material transaction where transaction date is between the @StartDate and @EndDate, transaction type is "Ship", reference type is "Customer Order" and delivery order status is "Approved". Union data for sample JO data, get from material transactions where transaction date is between the @StartDate and @EndDate, transaction type is "Ship", reference type is "Customer Order" and customer name contains "LSP*SECTION"
   4. For each job order in the shipment transaction table, perform the following:
      a. Execute stored procedure LSP_Rpt_NewDM_RMBreakdownPerJOSp passing the job order number, PO number and quantity shipped as input parameters.
      b. Insert into RM breakdown report final table.
   5. Select data from RM breakdown report final table where [Level] not equal to zero (0).

RM BEGINNING BALANCE REPORT

1. **Stored Procedure Name:** LSP_Rpt_NewDM_RMBeginningBalanceReportSp
   Definition: Get and returns the raw material (RM) beginning balance of each item as of the specified transaction date and product code filters. This will get actual cost of the RM based on the PO receipts and correct conversion rate, for SF items it will get the actual cost up to the last RM of the SF job if reference is from SF job order while standard cost for non-SF job order reference.
   Parameters: (1) @TransDate; and (2) @ProdCode
   Procedures:
   1. Declare and create temporary tables for the storage item actual costs and rm beginning balance report table result set.
   2. Get total quantity from material transactions where transaction type is not "Split/Merge" and not "Labor Next Operation" (means all transaction type including cycle count, issuance, withdrawal, cycle count, receipts, etc.) grouped by per item, item vendor and per lot number where item product code is @ProdCode parameter and transaction date is less before the @TransDate parameter.
   3. For each item and lot number, perform the following:
       a. For SF item where lot number exists as job order number: (1) insert values into actual cost temporary table, execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the lot number as job order number, 0 as suffix, item, transaction date and job qty as input parameters. (2) Then insert into rm beginning balance report table, get values form actual cost temporary table where [Level] = 0 (Level 0 means the parent item, in this case the SF material is the parent item). Divide the actual cost to the SF job quantity to get the unit cost then multiply it by the total qty on hand to get the current actual sf cost.
       b. For SF item where lot number does not exist as job order number: (1) get the item standard cost, get it from Item Pricing where effectivity date is before the @TransDate parameter; (2) perform currency conversion to USD and PHP (from currency code is from the currency code of the item in item pricing); (3) insert values into rm beginning balance report table, set the material unit cost in PHP and USD based on the currency conversion execution, other costs set to zero (0) (these includes PI-FG costs, landed cost and sf labor costs)
       c. For non SF items (RM, PI-FG, SC, etc): (1) execute the stored procedure LSP_ActlCost_GetMatlCostingSp passing the item, lot number, transaction date as input parameters, job qty, actual costs in PHP and USD values as output parameters (these includes material cost, landed cost, PI costs, labor and overhead costs). (2) Insert values into rm beginning balance report table, set the item costs get it from the output parameter values upon executing the stored procedure.
       d. Initialize all values related to rm beginning balance report fields to zero (0).
   4. Return the consolidated values from rm beginning balance report temporary tables.

SLOW MOVING ANALYSIS REPORT

1. **Stored Procedure Name:** LSP_GetSlowMovingAnalysisReportRemarksSp
   Definition: Returns the slow moving analysis remarks based on the item.
   Input Parameter: (1) @Material
   Output Parameter: (1) @Remarks
   Procedures:
   1. Declare and create temporary table for the storage of level 1 parent item code
   2. Insert values into the temporary table, get it from current materials. Get the parent item code of the said material.
   3. Set value for @FGParentCount, get it from temporary table where count of item starts with "FG-"; set value @SFParentCount where item starts with "SF-"
   4. Set value for @SFSubParentCount, get count of parent items who are using the first level SF item.

5. Set remarks to "Not being used by any FG" if @FGParentCount and @SFParentCount is equal to zero(0); "Being used by SF but SF is not being used by any FG" if @SFParentCount is greater than zero and @SFSubParentCount is equal to zero (0). Otherwise set remarks to blank.

2. **Stored Procedure Name:** LSP_Rpt_NewDM_SlowMovingAnalysisReportSp
   Definition: Get and returns the slow-moving analysis report based on the past 12 or 24 months.
   Parameters: (1) @Months
   Procedures:
   1. Declare and create temporary tables for slow moving item lists, latest PO received dates, latest issue dates, item lot location costs and direct material actual costs.
   2. Set value for @StartDate and @EndDate. @StartDate = current date minus @Months parameter value (values can be 12 or 24); @EndDate = current date.
   3. Insert values into slow moving item list, get it from item lot locations where qty on hand is greater than zero (0) and item does not have "Issuance" material transactions between @StartDate and @EndDate date range and item status is not "OBSOLETE" (excluding supplies item, product code of "OS-" and "-SUP").
   4. Insert values into temporary table latest issue dates, get items only exists in slow moving item lists and transaction type is "Issuance" and reference type = "Job"
   5. Insert values into temporary table latest PO received dates, get items only exists in slow moving item lists and transaction type is "Receipts" and reference type = "Purchase Order"
   6. For each slow-moving item per lot number, perform the following:
      a. For SF item where lot number exists as job order number: (1) insert values into actual cost temporary table, execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the lot number as job order number, 0 as suffix, item, transaction date and job qty as input parameters. (2) Then insert into item lot location cost temporary table, get values form actual cost temporary table where [Level] = 0 (Level 0 means the parent item, in this case the SF material is the parent item). Divide the actual cost to the SF job quantity to get the unit cost then multiply it by the total qty on hand to get the current actual sf cost.
      b. For SF item where lot number does not exist as job order number: (1) get the item standard cost, get it from Item Pricing where effectivity date is before the @TransDate parameter; (2) perform currency conversion to PHP (source currency code is from the currency code of the item in item pricing); (3) insert values into item lot location cost temporary table, set the material unit cost in PHP based on the currency conversion execution, other costs set to zero (0) (these includes PI-FG costs, landed cost and sf labor costs)
      c. For non-SF items (RM, PI-FG, SC, etc): (1) execute the stored procedure LSP_ActlCost_GetMatlCostingSp passing the item, lot number, transaction date as input parameters, job qty, actual costs in PHP and USD values as output parameters (these includes material cost, landed cost, PI costs, labor and overhead costs). (2) Insert values into item lot location cost temporary table, set the item costs get it from the output parameter values upon executing the stored procedure (insert into PHP unit costs).
      d. Initialize all values related to rm beginning balance report fields to zero (0).
   7. Select report output, get from slow moving item lists joining temporary tables latest PO received date, latest issue date and item lot location actual costs. *Material status: A = Active; O = Obsolete and S = Slow Moving

   INVENTORY TURNOVER REPORT

1. **Stored Procedure Name:** LSP_NewDM_GetInventorySafetyStockMaterialLandedCostSp
   Definition: Stored procedure that gets and return the inventory material cost, landed cost and safety stock material unit cost
   Input Parameter: (1) @ProdCode

Output Parameter: (1) @InvtyMaterialCost; (2) @InvtyLandedCost; and (3) @SafetyMaterialCost

Procedures:

1. Declare and create temporary table for item costs, item lot location cost and item actual material costs.

2. Get all list of items from item lot locations (quantity on hand is greater than zero (0)) based on the product code filter parameter.

3. For each item perform the following:

   a. Set safety stock quantity, get from "Item/Warehouse"; set standard item price to unit_price1 times 1.2 and item pricing currency code from the "Item Pricing" (price conversion into PHP if currency code is not "PHP")

   b. If item code is "SF-" and lot number exist as job order number: insert into actual cost temporary tables, execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the parameters @LotNumber as @JobOrder, 0 as @Suffix, @Item, @LotCreateDate as @TransDate and @LotQty as @JobQty for input parameters. Insert into @ItemCost table, get values from #DMActualCost where [Level] = 0 (means the parent SF item) computing the material costs as the sum of material and PI material costs and landed costs divided by the SF job quantity to get the SF item unit material costs.

   c. If item code is "SF-" and lot number does not exists as job order number: get the standard price (unit_price1 divided by 0.9 from "Item Pricing") where the effectivity date is beyond the lot create date performing currency conversion into PHP and/or USD based on the item pricing currency code. Insert into @ItemCost tables the computed costs setting up the landed cost into zero (0).

   d. If item code is not "SF-", execute stored procedure LSP_ActlCost_GetMatlCostingSp passing up @Item, @LotNumber, @LotCreateDate as input parameters; @JobQty, @matl_unit_cost_usd, @matl_landed_cost_usd, @pi_fg_process_usd, @pi_resin_usd, @pi_vend_cost_usd, @pi_hidden_profit_usd, @sf_lbr_cost_usd, @sf_ovhd_cost_usd, @fg_lbr_cost_usd, @fg_ovhd_cost_usd, @matl_unit_cost_php, @matl_landed_cost_php, @pi_fg_process_php, @pi_resin_php, @pi_vend_cost_php, @pi_hidden_profit_php, @sf_lbr_cost_php, @sf_ovhd_cost_php, @fg_lbr_cost_php , @fg_ovhd_cost_php as output parameters. Insert into @ItemCosts, setting up the material cost as the sum of material and PI material costs from the output parameters and @matl_landed_cost_php output parameter as the material landed cost.

   e. Reset all output parameters into zero (0).

4. Insert into @ItemLotCost temporary table, get the total lot material cost and lot landed costs per item.

5. Set values for output parameters: @InvtyMaterialCost = total all lot material costs; @InvtyLandedCost = total all lot landed costs; @SafetyMaterialCost = total safety stock costs.

2. **Stored Procedure Name:** LSP_Rpt_NewDM_InventoryTurnOverReportSP

   Definition: Stored procedure that gets and return the inventory turnover report.

   Parameters: (1) @IsShowDetail; (2) @StartDate; (3) @EndDate

   Procedures:

   1. Declare and create temporary table for inventory turnover detail, direct material actual costs and report set final table.

   2. Set @StartDate as the first day of the past 12 month, @EndDate as the last date of the previous month

   3. Get all material transactions based on the following filters: (1) Ship transactions where reference number is not LSP section (PDN, PIS, QAC, PDE, MDE) and product code of the item are not FG and supplies (OS-* and *-SUP); (2) Issuance and withdrawal transactions where reference number is "**-" and "**RM-*" (e.g. 19-, 20-, 19RM-, 20RM-, etc) and product code of the item are not FG and supplies (OS-* and *-SUP); and (3) Miscellaneous Issuance transactions where reason code are ARM

(Additional RM Usage), ETS (Endorsed to Supplier (for Billing)), PIU (Plastic Injection Usage) and URE (Urethane Usage) and product code of the item are not FG and supplies (OS-* and *-SUP).

4. For each item perform the following:

   a. If item code is "SF-" and lot number exists as job order number: insert into actual cost temporary tables, execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the parameters @LotNumber as @JobOrder, 0 as @Suffix, @Item, @LotCreateDate as @TransDate and @LotQty as @JobQty for input parameters. Insert into inventory turnover detail temporary table (#InvtyTurnOverDtl), get values from #DMActualCost where [Level] = 0 (means the parent SF item) computing the material costs as the sum of material and PI material costs and landed costs divided by the SF job quantity to get the SF item unit material costs.

   b. If item code is "SF-" and lot number does not exist as job order number: get the standard price (unit_price1 divided by 0.9 from "Item Pricing") where the effectivity date is beyond the lot create date performing currency conversion into PHP and/or USD based on the item pricing currency code. Insert into Insert into inventory turnover detail temporary table (#InvtyTurnOverDtl) the computed costs setting up the landed cost into zero (0).

   c. If item code is not "SF-", execute stored procedure LSP_ActlCost_GetMatlCostingSp passing up @Item, @LotNumber, @LotCreateDate as input parameters; @JobQty, @matl_unit_cost_usd, @matl_landed_cost_usd, @pi_fg_process_usd, @pi_resin_usd, @pi_vend_cost_usd, @pi_hidden_profit_usd, @sf_lbr_cost_usd, @sf_ovhd_cost_usd, @fg_lbr_cost_usd, @fg_ovhd_cost_usd, @matl_unit_cost_php, @matl_landed_cost_php, @pi_fg_process_php, @pi_resin_php, @pi_vend_cost_php, @pi_hidden_profit_php, @sf_lbr_cost_php, @sf_ovhd_cost_php, @fg_lbr_cost_php , @fg_ovhd_cost_php as output parameters. Insert into Insert into inventory turnover detail temporary table (#InvtyTurnOverDtl), setting up the material cost as the sum of material and PI material costs from the output parameters and @matl_landed_cost_php output parameter as the material landed cost.

   d. Reset all output parameters into zero (0).

5. Perform material usage computation, get total material cost and landed cost grouped by product code (discarding SA- and RM- for SF and RM product codes).

6. For each product code usage per month, perform the following:

   a. Execute stored procedure LSP_NewDM_GetInventorySafetyStockMaterialLandedCostSp passing up @ProdCodeCrsr as @ProdCode as input parameter, @InvtyMatlCost, @InvtyLandedCost and @SafetyMatlCost as output parameters

   b. Insert into temporary table @DMInvty, inserting values for the total usage of material and landed costs, output parameter results of the stored procedure

7. Compute for the material usage, use pivot function for the total material and landed cost usage per month of each product code, insert into @report_set final report for inventory turnover report.

8. Update @report_set table, adding up the per 3 months usage based on @StartDate and @EndDate. Then identify the maximum and average data based on monthly material and landed cost usage and the max material and landed cost usage on 3 months total usage

9. If @IsShowDetail = 1, insert values into @report_set get values from #InvtyTurnOverDtl temporary table tagging the rows as "DETAILED" data.

INVENTORY TURNOVER REPORT

1. **Stored Procedure Name:** LSP_Rpt_NewDM_WIPShopFloorReportSp
   Definition: Stored procedure that gets and return the WIP Shop Floor Report. Actual cost of material is up to the last RM of the issued materials
   Procedures:
   1. Declare and create temporary table for item actual material costs and WIP shop floor report table.

2. Get all job order list where job status = "Release" and the total quantity completed plus quantity scrapped is not equal to job quantity released (all incomplete job orders)
3. For each job order and item perform the following:
   a. Insert values into item actual material costs by executing the stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing @Job, @Suffx, @Item, @JobStartDate, @QtyWip as input parameters.
   b. Insert into WIP shop floor report, get values form actual costs temporary table where Level = 1 (immediate child of the issued job material).
4. Select data from WIP shop floor report showing up the unit cost and total costs in PHP and USD currencies.

INVENTORY TURNOVER REPORT

1. **Stored Procedure Name:** LSP_Rpt_NewDM_MiscellaneousTransactionReportSp
   Definition: Stored procedure that gets and return the miscellaneous transaction report based on the filtered start and end dates.
   Parameters: (1) @StartDate; and (2) @EndDate
   Procedures:
   1. Declare and create temporary table for item actual material costs and miscellaneous transaction report table.
   2. Get all miscellaneous transactions data: (1) from job transaction where item is "SF-", quantity scrapped is greater than zero (0) and job transaction is from @StartDate and @EndDate; (2) from material transaction where transaction types are "Misc. Issue", "Misc. Receipt" and "Cycle Count" and material transaction is from @StartDate and @EndDate.
   3. For each transaction get, perform the following:
      a. If item code is "SF-" and lot number exists as job order number: insert into actual cost temporary tables, execute stored procedure LSP_DM_ActlCost_GetJobMatlTransCostingSp passing the parameters @LotNumber as @JobOrLot, 0 as @Suffix, @Item, @TransDate as @TransDate and @LotQty as @ABSTransQty for input parameters. Insert into material transaction temporary table (#MiscTransReport), get values from #DMActualCost where [Level] = 0 (means the parent SF item) computing the material costs as the sum of material and PI material costs and landed costs divided by the SF job quantity to get the SF item unit material costs.
      b. If item code is "SF-" and lot number does not exist as job order number: get the standard price (unit_price1 divided by 0.9 from "Item Pricing") where the effectivity date is beyond the lot create date performing currency conversion into PHP and/or USD based on the item pricing currency code. Insert into Insert into material transaction temporary table (#MiscTransReport) the computed costs setting up the landed cost into zero (0).
      c. If item code is not "SF-", execute stored procedure LSP_ActlCost_GetMatlCostingSp passing up @Item, @LotNumber, @LotCreateDate as input parameters; @JobQty, @matl_unit_cost_usd, @matl_landed_cost_usd, @pi_fg_process_usd, @pi_resin_usd, @pi_vend_cost_usd, @pi_hidden_profit_usd, @sf_lbr_cost_usd, @sf_ovhd_cost_usd, @fg_lbr_cost_usd, @fg_ovhd_cost_usd, @matl_unit_cost_php, @matl_landed_cost_php, @pi_fg_process_php, @pi_resin_php, @pi_vend_cost_php, @pi_hidden_profit_php, @sf_lbr_cost_php, @sf_ovhd_cost_php, @fg_lbr_cost_php , @fg_ovhd_cost_php as output parameters. Insert into Insert into material transaction temporary table (#MiscTransReport), setting up the material cost as the sum of material and PI material costs from the output parameters and @matl_landed_cost_php output parameter as the material landed cost.
      d. Reset all output parameters into zero (0).
   4. Select all computed data from material transaction temporary table (#MiscTransReport), adding up new rows for (1) "SCRAP" (total misc. issue and SF Scrap data where reason description is "*Scrap*") and (2) "Section Request" (total misc. issue where reason description is "*Request*")