**\*Omiscope Process\***  
  
 **Data Import**

* First, we bring in data from *Pcard* and *Spend data* in Omniscope, as shown in the image below.  
    
  A screenshot of a computer program

  Description automatically generated

**Append Data**

* Append the data from both sources.

A screenshot of a computer program

Description automatically generated  
**Formatting and Merging**

* Apply formatting, such as renaming and removing unnecessary fields.
* Merge the resulting data with *Coupa* data, matching on *PO Number* and *Supplier Name*. This process pulls in the *Purchase ID* from the Coupa dataset. A screenshot of a computer

  Description automatically generated

**Publish to tblRefresh**

* Publish the final merged data to tblRefresh.  
     
  A screenshot of a computer

  Description automatically generated

**Execute Script 1 for Updates**

* Run *Script 1*, which applies further updates to tblRefresh as specified.

\***Microsoft SQL Management Studio\***

1. **Data Normalization**  
   We update the tblRefresh table with normalized supplier names from the kbNormalisation table. If the supplier name is missing, it will be labeled as 'UNKNOWN SUPPLIER'.
2. **Company Structure Update**  
   Then we update Company Region and Company Country based on luCompanyRegions. For specific entries like "PCARD-US%", it assigns the country as "United States".
3. **Supplier Type Update**  
   All records with "PCARD" system type are updated to have the supplier type as "PCARD". Unwanted entries like those with "AMERICAN EXPRESS" and department code '0000' are deleted from the table.
4. **Month & Year Update from Reporting Date**  
   Then the Reporting Month and Reporting Year fields in the tblRefresh table based on the Reporting Date field.
5. **Payment Terms Update**  
   Payment terms in tblRefresh are updated using values from the kbPaymentTerms table.
6. **Financial Year Update**  
   the Financial Year, Financial Quarter, and Financial Period for each record based on data from either luFinancialYear or an external data source.
7. **Preferred Supplier Identification**  
   This step identifies preferred suppliers based on luPreferredSuppliersList by matching the normalized supplier and region. If no match is found, the value is set to 'No'.
8. **Country Name Update**  
   Vendor and company country names are updated based on ISO country codes found in kbLatLong. The script maps 2-character and 3-character ISO country codes to full country names.
9. **Currency Name and Prefix Update**  
   Currency name and prefix are updated based on the kbCurrencyName table for the respective reporting currency codes in the tblRefresh table.

**10. Supplier Information:**

* This updates the Normalized Supplier in the tblRefresh table using data from the kbNormalisation table.

UPDATE a

SET a.[Normalized Supplier] = b.[Normalized Supplier]

FROM tblRefresh a

INNER JOIN kbNormalisation b

ON a.[Supplier Name] = b.[Supplier Name];

**11. Insert Missing Normalizations:**

* Any suppliers that do not exist in the kbNormalisation table are added from the tblRefresh table.

INSERT INTO kbNormalisation

SELECT DISTINCT [Supplier Name], [Normalized Supplier]

FROM tblRefresh

WHERE [Supplier Name] NOT IN (

SELECT DISTINCT [Supplier Name]

FROM kbNormalisation

WHERE [Supplier Name] IS NOT NULL

);

**12. Reset Categories in tblRefresh:**

* Clears existing category data (Category 1, Category 2, Category 3, and Category Source) in the tblRefresh table.

UPDATE tblREFRESH

SET [Category 1] = NULL,

[Category 2] = NULL,

[Category 3] = NULL,

[Category Source] = NULL;

**13. Temporary Table Creation for Unique Classifications:**

* Create a temporary table tmpUnique by selecting unique suppliers from tblHistorical that have a single classification for each category.

SELECT [Supplier Name], [Category 1], [Category 2], [Category 3], [Category 4]

INTO tmpUnique

FROM tblHistorical

WHERE [Category 1] IS NOT NULL AND [Supplier Name] IS NOT NULL

GROUP BY [Supplier Name], [Category 1], [Category 2], [Category 3], [Category 4]

HAVING COUNT(DISTINCT [Category 1]) = 1

AND COUNT(DISTINCT [Category 2]) = 1

AND COUNT(DISTINCT [Category 3]) = 1;

**14. Add Classification Count to Temporary Table:**

* Adds a Count column to the tmpUnique table to count how many times each supplier appears.

ALTER TABLE tmpUnique ADD Count INT;

UPDATE tmpUnique

SET COUNT = (

SELECT COUNT(\*)

FROM tmpUnique AS S

WHERE s.[Supplier Name] = tmpUnique.[Supplier Name]

);

**15. Update tblRefresh with Unique Supplier Classification:**

* Updates classification information for suppliers in tblRefresh based on unique entries from tmpUnique.

UPDATE a

SET a.[Category 1] = b.[Category 1],

a.[Category 2] = b.[Category 2],

a.[Category 3] = b.[Category 3],

a.[Category Source] = 'UNIQUE'

FROM tblRefresh a

INNER JOIN tmpUnique b

ON a.[Supplier Name] = b.[Supplier Name]

WHERE b.[Count] = 1 AND [Supplier Type] <> 'EMPLOYEE';

**16. Update Based on Source Information:**

* Several updates are performed on tblRefresh based on various sources:
  + Update where Category 1 is not NULL using 'Oracle' as the source.
  + Map suppliers from luExpenseTypeMapping, luTaxonomyMapping, luAccountMapping, luSaasMapping, and other classification mappings.
  + Use keyword-based classification for specific invoice line descriptions (e.g., internet services, contingent workforce outsourcing).
  + Handle non-addressable categories such as tax authorities and other non-classifiable entities.

**17. Special Supplier Classification Updates:**

* Special cases for suppliers like CDW, American Express, Cushman & Wakefield, and others are handled using specific rules and mappings.

**18. Final Updates:**

* Reset classifications for specific suppliers, especially where certain descriptions or categories don’t apply or need to be set to 'Non-Addressable'.
* Assign final classifications for suppliers where necessary, such as using LinkedIn data for certain professional and managed services.

**19. Category Updates**

* Updates Category 1, Category 2, Category 3, and Category Source fields in the tblRefresh table from various lookup tables based on different matching conditions, such as:  
  + Supplier Number, System, Supplier Type, Supplier Region.
  + Supplier name when supplier number isn’t available.
* Each section has a conditional check to ensure updates only apply where Category 1 is not equal to 'Non-Addressable' or is NULL.
* Final classifications come from different sources (LINKEDIN, SEGMENTATION, AIC, etc.).

**20. Classification from Knowledge Base (KB)**

* **Update Classification from KB:**
  + Updates categories from kbClassification for suppliers with no assigned categories.
  + This update introduces several additional category fields (Category 4 to Category 7), expanding the classification system.
* **Add New Entries to KB:**
  + Adds new classification data from tblRefresh into kbClassification if entries do not already exist.

**21. Classification Delta Creation**

* **Create Table for Classification Deltas:**
  + Creates a temporary table qaDeltas containing classification deltas for all records in tblRefresh without an existing classification.
* **Delta Log Table Creation:**
  + Inserts qaDeltas data into logQADeltas for audit and tracking purposes.

**23. Contract Information Update**

* Updates contract details from the luApptusContracts table into tblRefresh, utilizing fields such as:
  + Contract Number or PO Number as the matching criteria to ensure accuracy in assigning contract details.

**24. Payment Status Calculation**

* **Calculate Payment Terms:**
  + Using indices on key date columns, calculates Actual Payment Terms, Days Before Due, and other metrics.
  + Determines Payment Status (Late, Early, On Time) based on Days Before Due.
* **Weighted Calculation:**
  + Calculates Weighted Calculation based on the product of Payment Terms and Spend.

**25. Purchase ID and PR Number Updates**

* Updates various custom fields in tblRefresh from the luPURCHID3 table, including purchase ID, PR number, and details related to purchase category and project type.

**26. Flag Updates**

* **Flags for Retro PO, Contract, and PO Presence:**
  + Retro PO flags are set to ‘Yes’ or ‘No’ based on whether Invoice Date precedes PO Date.
  + Additional flags (With Contract, With PO) identify records with contract or PO numbers.
* After normalization, we send the file to QA.
* Once the classification process is completed by the QA team, we merge the data with the historical records and perform the following updates.

 **Initial Addressable Update**

* Set all rows in the [Addressable] column to 'Addressable'.

sql

Copy code

UPDATE tblHistorical

SET [Addressable] = 'Addressable'

GO

 **Clear Custom Field**

* Reset the [Custom Yes/No 5] field to NULL.

sql

Copy code

UPDATE tblHistorical

SET [Custom Yes/No 5] = NULL

GO

 **Non-Addressable Classification**

* For records with [Category 1] as 'Non-Addressable', update specific fields to reflect non-addressable status.

sql

Copy code

UPDATE tblHistorical

SET [Addressable] = 'Non-Addressable',

[Managed Spend Type] = 'Non-Addressable',

[Category Manager] = 'Non-Addressable',

[Managed Spend] = 'No',

[Entity Description] = 'Non-Addressable',

[Custom 22] = 'Non-Addressable',

[Custom 20] = 'No',

[Custom 21] = 'Not Managed by GSS or IBE',

[Custom Yes/No 5] = 'YES'

WHERE [Category 1] = 'Non-Addressable'

GO

 **Update from luSegmentation**

* For non-addressable suppliers, update multiple fields by joining tblHistorical with luSegmentation using Supplier Name, Supplier Number, Company Region, and other category fields.

sql

Copy code

UPDATE a

SET a.[Normalized Supplier] = b.[Normalized Supplier],

a.[Category 1] = b.[Category 1],

a.[Category 2] = b.[Category 2],

a.[Category 3] = b.[Category 3],

a.[Category Manager] = b.[Category Manager],

a.[Custom 21] = b.[Managed Supplier/Custom 21],

a.[Entity Description] = b.[Addressable Owner/Entity Description],

a.[Addressable] = b.[Addressable],

a.[Custom 20] = b.[Pass-Thru/Custom 20],

a.[Managed Spend Type] = b.[Segment/Managed Spend Type],

a.[Managed Spend] = b.[Managed Spend],

a.[Custom 22] = b.[Category Team/Custom 22],

a.[Custom Yes/No 5] = 'YES'

FROM tblHistorical a INNER JOIN luSegmentation b

ON a.[Supplier Name] = b.[Supplier Name]

WHERE a.[Category 1] != 'Non-addressable'

 **Update with luSegmentationsupno**

* Match data based on Supplier Number, Company Region, and other categories for updating addressable fields.

sql

Copy code

UPDATE a

SET a.[Normalized Supplier] = b.[Normalized Supplier],

a.[Category 1] = b.[Category 1],

a.[Category 2] = b.[Category 2],

a.[Category 3] = b.[Category 3],

a.[Category Manager] = b.[Category Manager],

a.[Custom 21] = b.[Managed Supplier/Custom 21],

a.[Entity Description] = b.[Addressable Owner/Entity Description],

a.[Addressable] = b.[Addressable],

a.[Custom 20] = b.[Pass-Thru/Custom 20],

a.[Managed Spend Type] = b.[Segment/Managed Spend Type],

a.[Managed Spend] = b.[Managed Spend],

a.[Custom 22] = b.[Category Team/Custom 22],

a.[Custom Yes/No 5] = 'YES'

FROM tblHistorical a INNER JOIN luSegmentationsupno b

ON a.[Supplier Number] = b.[Supplier Number]

WHERE a.[Category 1] != 'Non-addressable'

GO

 **luSegmentationnoclass Update**

* Update using Supplier Name where [Category 1] is NULL.

sql

Copy code

UPDATE a

SET a.[Normalized Supplier] = b.[Normalized Supplier],

a.[Category 1] = b.[Category 1],

a.[Category 2] = b.[Category 2],

a.[Category 3] = b.[Category 3],

a.[Category Manager] = b.[Category Manager],

a.[Custom 21] = b.[Managed Supplier/Custom 21],

a.[Entity Description] = b.[Addressable Owner/Entity Description],

a.[Addressable] = b.[Addressable],

a.[Custom 20] = b.[Pass-Thru/Custom 20],

a.[Managed Spend Type] = b.[Segment/Managed Spend Type],

a.[Managed Spend] = b.[Managed Spend],

a.[Custom 22] = b.[Category Team/Custom 22],

a.[Custom Yes/No 5] = 'YES'

FROM tblHistorical a INNER JOIN luSegmentationnoclass b

WHERE a.[Category 1] IS NULL

GO

 **luEmployeeAMEXSEGmapping Update**

* Update for American Express suppliers where [Managed Spend] is NULL.

sql

Copy code

UPDATE a

SET a.[Category Manager] = b.[Category Manager],

a.[Custom 21] = b.[Managed Supplier/Custom 21],

a.[Entity Description] = b.[Addressable Owner/Entity Description],

a.[Addressable] = b.[Addressable],

a.[Custom 20] = b.[Pass-Thru/Custom 20],

a.[Managed Spend Type] = b.[Segment/Managed Spend Type],

a.[Managed Spend] = b.[Managed Spend],

a.[Custom 22] = b.[Category Team/Custom 22],

a.[Custom Yes/No 5] = 'YES'

FROM tblHistorical a INNER JOIN luEmployeeAMEXSEGmapping b

WHERE a.[Managed Spend] IS NULL AND b.[Normalized Supplier] = 'American Express'

GO

 **Non-Addressable LinkedIn and American Express**

* Update LinkedIn and American Express suppliers as non-addressable.

sql

Copy code

UPDATE a

SET a.[Addressable] = b.[Addressable],

a.[Entity Description] = 'Non-Addressable'

FROM tblHistorical a INNER JOIN luNonAddressable b

WHERE a.[Normalized Supplier] = 'LINKEDIN EMPLOYEE' OR a.[Normalized Supplier] = 'AMERICAN EXPRESS'

GO

 **Final Updates for Supplier 66549**

* Apply specific updates to supplier 66549 in the AMER region.

sql

Copy code

UPDATE tblHistorical

SET [Category Manager] = 'Corporate Services, Technology & Workplace',

[Custom 21] = 'Managed by IBE',

[Entity Description] = 'Addressable ES',

[Addressable] = 'Addressable',

[Custom 20] = 'Yes',

[Managed Spend Type] = 'Critical',

[Managed Spend] = 'Yes',

[Custom 22] = 'Karttikeya Shah',

[Custom Yes/No 5] = 'YES',

[Category 1] = 'Technology',

[Category 2] = 'Hardware',

[Category 3] = 'Computers & Peripheral Equipment'

WHERE [Supplier Number] = '66549'

AND [Company Region] = 'AMER'

GO

 **Set Intercompany Designations**

* Update Intercompany status based on supplier.

sql

Copy code

UPDATE tblHistorical

SET [Intercompany] = 'NO'

WHERE [Normalized Supplier] <> 'LINKEDIN'

GO

UPDATE tblHistorical

SET [Intercompany] = 'YES'

WHERE [Normalized Supplier] = 'LINKEDIN'

GO

 **Rank Classification Index Creation and Table Initialization**:

* It starts by creating an index idxRK1 to optimize searches in tblHistorical.
* A temporary table #tmpTopCat is populated with supplier spend information, grouped by categories, and includes a column to count the number of lines per supplier.

 **Vendor Total Spend Calculation**:

* Another temporary table, #tmpVenTot, is created to store the total spend per supplier.
* The total vendor spend is then joined back into #tmpTopCat to calculate the percentage of spend per category.

 **Filtering and Deduplication**:

* Vendors are classified for update based on whether they exceed a 55% spend threshold (or are single-line records).
* A deduplication step identifies vendors with multiple entries to help focus updates on the most significant records.

 **Updating tblHistorical Table**:

* The script applies the ranking categories back to tblHistorical where category fields are blank, effectively filling in missing data based on the classification logic.

 **Manual Updates to Specific Suppliers**:

* For specific suppliers like UPSTAK, WILLIS TOWERS WATSON, and PHS WEST, categories are directly set in tblHistorical.

 **Contract Information Update**:

* The contract information, including contract names, currency, values, and renewal dates, is updated by joining tblHistorical with luApptusContracts.
* This helps align any blank contract fields with values from the contracts table, providing richer, complete contract data for existing records.

 **Additional Custom Fields Update**:

* For records that align with luPURCHID3, additional fields (e.g., custom categories, assignment groups, statuses, and negotiation types) are populated in tblHistorical.

 **Preferred Supplier**:

* Sets all Preferred Supplier entries to 'NO' and updates it to 'YES' if there's a match in the luPreferredSuppliersList table.

 **Cost Center Mapping Lookup**:

* Updates fields [Custom 2], [Custom 3], and [Custom 4] based on a lookup table luCostCentermapping.

 **Diversity**:

* Resets fields related to Diversity to NULL for specific records, then updates them by matching tblHistorical with tables luDiversity1, luDiversity2, and luDiversity3 based on multiple criteria.
* Sets Diversity to 'NOT DIVERSE' when certain criteria aren't met.

 **SSA Numbers**:

* Updates [Risk Factor 1] and sets a flag in [Risk Indicator] based on data in the luSSANumbers table.

 **Invoice Totals and Ranges**:

* Creates a temporary table for calculating invoice totals and assigns them to specific range labels.
* Updates tblHistorical with new invoice totals and range labels.

 **Vendor Spend by Year**:

* Generates a temporary table for calculating total yearly spend per vendor, assigning range labels, and updating tblHistorical.

 **Classification Status**:

* Sets the [Classified Flag] to 'Yes' or 'No' based on whether [Category 1] is populated.

 **Vendors Main Category**:

* Determines the main category per vendor based on highest spend in [Category 1] and updates tblHistorical.

 **Businesses Supplied**:

* Calculates the number of distinct businesses supplied by each vendor and updates tblHistorical.

 **Last 12 Months**:

* Labels data by previous years based on [Reporting Date].

Lastly we push the data to looker which updates the AIC instance.