# Mapping

No QVD files are referenced directly in this specific part of the script.

* Creates a mapping table MapFlags to convert binary flags (1 or 0) into human-readable values (Yes or blank).
* Defines a mapping table MapReg\_PDT\_Strength\_Status to map various project status descriptions to a value of 0.
* Defines a mapping table MapReg\_PJT\_Status to map project statuses "Completed" and "Cancelled" to a value of 0.
* Using the Mapping

Summary

This part of the script focuses on setting up mapping tables for different status flags and project statuses to standardize and simplify the data transformation process. Each mapping table converts specific text values into numeric or other text values.

Islands

* Generates a temporary calendar table tCalendar with date numbers from 1 to twice the value of vToday.
* Creates a detailed calendar table iCalendar with various date-related fields such as year, quarter, month, week, day, and different date differences relative to vToday. The temporary tCalendar table is dropped after use.
* Loads a predefined set of error types from an inline table, which categorizes various errors with codes and descriptions.

# Country Region

* Loads data from the DIM\_PLM\_COUNTRY\_REGION.QVD file.
* Renames fields for clarity and usability, such as converting [COUNTRY\_KEY] to [\_CNTRY.REGION\_COUNTRY\_KEY].
* Uses nested If and match functions to set the CNTRY.Market field based on the values of L3\_CODE, REGION\_NAME, and L2\_CODE.

This script section focuses on loading and transforming country region data from a QVD file, renaming fields, and applying conditional logic to derive new values.

# ORGANIZATION

#### Loading Organization Data

This section loads data from the DIM\_PLM\_COUNTRY\_ORGANIZATION.QVD file, renaming fields for clarity and consistency with the data model. The fields include various market and country identifiers and descriptions.

#### Joining Organization Data to Country Region Data

The script performs a left join to merge the COUNTRY\_REGION data into the ORGANIZATION table using the keys \_CNTRY.REGION\_COUNTRY\_KEY and \_CNTRY.ORGANIZATION\_KEY. This join enriches the organization data with additional regional details.

# Product

This section loads product-related data from the DIM\_PLM\_PRODUCT.QVD file, renaming fields for clarity and consistency. The fields include various product identifiers, descriptions, pharmaceutical details, regulatory statuses, and additional flags indicating missing or specific attributes.

#### Field Renaming and Transformation

Fields are renamed to provide clear and consistent naming conventions. This includes product codes, descriptions, internal numbers, administration routes, packaging forms, product technologies, dosage forms, and other relevant product information. Flags are used to indicate missing or specific attributes.

# Indication

This section loads indication-related data from the DIM\_PLM\_INDICATION.QVD file, renaming fields for clarity and consistency. The fields include various keys, product characteristics, dates, and status indicators relevant to pharmaceutical indications.

#### Field Renaming and Transformation

Fields are renamed to provide clear and consistent naming conventions. This includes product keys, region country keys, start dates, submission dates, approval dates, and other relevant information about the pharmaceutical indications.

# Project

* loads project-related data from the DIM\_PLM\_PROJECT.QVD file, renaming fields for clarity and consistency. The fields include various project identifiers, names, descriptions, statuses, dates, and other relevant project details.
* Fields are renamed to provide clear and consistent naming conventions. This includes project keys, names, statuses, creation dates, modification dates, and other relevant information about the projects.
* loads project history data from the DIM\_PLM\_PROJECT\_HIST.QVD file. It includes conditions to filter records based on activity indicators, version, dates, and status. The script then performs a left join to add a flag to the main PROJECT table for projects moved to next year, canceled, or on hold.
* loads and transforms data related to different types of project sites (Manufacturing, Development, Packaging, Release, Testing) from the PROJECT table. The fields are renamed for clarity, and specific values are derived using functions like Trim and Combining Site Data
* Combines the site data into a unified format for easy reference and analysis, using concatenation to handle multiple entries.
* Handles specific data for PLP sites and IP owners, including additional transformation and joining steps to integrate all necessary information.

TGO

* loads data related to TGO segments from the DIM\_TGO\_SEGMENT.QVD file. The fields include manufacturing sites and TGO segment details.
* Fields are renamed to provide clear and consistent naming conventions. This includes manufacturing sites and TGO segment information.

# Medical Writing Monthly Milestone

* loads data related to medical milestones from the DIM\_MEDICAL\_MILESTONE.QVD file. The fields include project keys, milestone categories, clinical study phases, activity names, and milestone dates.
* processes the medical milestone data for each month from the current month to 16 months ahead. It joins the milestone data to create monthly milestone records, transforming and preparing milestone descriptions and categories for each month.
* After processing, the script consolidates the final milestone data into DIM\_MEDICAL\_MILESTONE\_FINAL, ensuring that only records with non-null milestones are included. It then drops the intermediate tables used during processing.

# Api

* loads API supplier-related data from the API\_SUPPLIER\_N\_N\_BI.QVD file, renaming fields for clarity and consistency. The fields include various keys, lead times, shelf life, supplier information, and statuses relevant to API (Active Pharmaceutical Ingredient) suppliers.
* Fields are renamed to provide clear and consistent naming conventions. This includes project keys, API lead times, shelf life, supplier details, prices, and other relevant information about API suppliers. Additional fields such as API\_AGREEMENT\_STATUS and FACILITY\_ADDRESS have been added for more detailed information.

# Packaging

* loads packaging-related data from the DIM\_PR\_PACKAGING.QVD file. The fields include project keys, packaging agreement statuses, comments, material names, supplier details, manufacturing facility information, and prices.
* Fields are renamed to provide clear and consistent naming conventions. This includes packaging keys, project identifiers, agreement statuses, material names, supplier names, manufacturing facilities, prices, lead times, and risk assessments.

# Excipients

* loads excipients-related data from the DIM\_PR\_EXCIPIENTS.QVD file. The fields include project keys, excipient agreement statuses, comments, supplier details, manufacturing facility information, and prices.
* Fields are renamed to provide clear and consistent naming conventions. This includes excipient keys, project identifiers, agreement statuses, material names, supplier names, manufacturing facilities, prices, shelf life, lead times, and risk assessments.

# Calendar Creation

* defines a subroutine makeCalendar to create calendar tables. It loads date-related fields from an existing iCalendar table, transforming and renaming fields for specific use cases. The subroutine is then called for different project milestones:
  + Loads date fields from iCalendar.
  + Renames fields to include the provided name prefix.
  + Filters data to include only dates that exist in iCalendar.
* Creates calendar tables for various project milestones such as PJT.1st Submission, PJT.1st Launch, PJT.Selection, and PJT.Endorsement.

# SKU

* loads SKU-related data from the DIM\_SKU.QVD file. The fields include various product codes, descriptions, strengths, inventory types, packaging details, manufacturing sites, suppliers, and market information.
* Fields are renamed to provide clear and consistent naming conventions. This includes SKU keys, product identifiers, descriptions, dosage forms, technologies, packaging details, prices, and other relevant information about the SKUs.

# Pack

* loads pack-related data from the DIM\_PLM\_PACK.QVD file. The fields include project keys, product keys, country codes, manufacturing facilities, packaging details, and various project-specific attributes.
* Fields are renamed to provide clear and consistent naming conventions. This includes pack keys, project identifiers, activity links, creation and modification dates, packaging sizes, and statuses.

# Workpackage

* loads workpackage-related data from the DIM\_PLM\_WORKPACKAGE.QVD file. The fields include keys, project details, planned and actual dates, statuses, probabilities, and various other attributes related to work packages.
* Fields are renamed to provide clear and consistent naming conventions. This includes workpackage keys, project identifiers, planned and actual dates, probabilities, statuses, and other relevant information about work packages.
* The script includes logic to handle duplicates by counting distinct keys and flagging records where necessary.
* The script performs joins to integrate workpackage data into the main PROJECT table, aggregating certain flags and ensuring comprehensive project data.

# TASK\_WP

* Loads task-related data from DIM\_TASK\_OF\_WP.QVD.
* Several fields are renamed for clarity.
* Joins the workpackage data to add clinical work package keys to sub-work package tasks.
* Aggregates dates related to clinical review and design review meetings.
* Joins the aggregated data with the project table to ensure comprehensive date tracking for projects.
* Final join operations to integrate task-related information with project-level data.
* Drops intermediate fields post-aggregation to maintain data integrity and relevance.

# Country Milestone

The script loads the COUNTRY\_MILESTONE table from the DIM\_COUNTRY\_MILESTONE.QVD file.  The script performs a left join on the TASK\_WP table with the COUNTRY\_MILESTONE table. The join is based on MILESTONE\_C\_NAME, aligning it with TASK\_WP.Activity Name.

# For HiH

1. **Launch Dates Extraction**:
   * Loads and processes data for project launch dates, focusing on "Country Launch" activities.
   * Determines the earliest launch date and assigns it to each project, while also extracting the country launch type and name.
2. **Submission Dates Extraction**:
   * Processes submission dates for projects, focusing on activities related to "Country Submission".
   * Filters submissions matching the launch short name and determines the earliest submission date for each project.
3. **Target Approval Dates Extraction**:
   * Loads data for target approval dates, focusing on "Country Target Approval".
   * Filters approvals matching the launch short name and determines the earliest approval date for each project.
4. **Data Cleanup**:
   * Drops intermediate tables to maintain a clean data model.

# Task Indication

* Loads task indication data from the DIM\_TASK\_OF\_IND.QVD file into the TASK\_IND table.
* Selects and renames various fields related to task indication, including keys, activity names, types, start and finish dates, product information, and other relevant details.

# Milestone Data

* Loads milestone data from the TASK\_WP table into the MILESTONES table.
* renames fields to create a new structure:
* Calculates and formats dates:
  + MLSTN.First Planned Date and MLSTN.First Actual Date using the minimum planned and actual start dates.
  + MLSTN.Last Planned Date and MLSTN.Last Actual Date using the maximum planned and actual start dates.

# Key\_Table:

* Loads distinct keys for \_SKU.KEY, \_PACK.KEY, \_WP.KEY, \_SKU\_Lync.PROJECT\_KEY, and \_SKU\_Lync.REGION\_COUNTRY\_KEY from SKU\_Lync.
* Loads distinct project-related keys and region-country keys from the PROJECT table into a temporary table Temp.
* Categorizes projects into various groups (Combination Projects, BD Projects, TAPI Projects, etc.) based on specific conditions using wildmatch and match functions.
* Joins these groups back to the Temp table.
* Assigns specific region-country keys based on conditions related to Europe.
* Joins these to the Temp table.
* Assigns manufacturing and development site categories based on specific conditions.
* Joins these categories back to the Temp table.
* Joins the prepared data from Temp into KeyTableTemp.
* Adds project keys and flags for PLP projects.
* Ensures non-null product keys are assigned correctly.
* Joins additional data from various tables (FACT\_REG\_PDT, INDICATION, PRODUCT, etc.) to KeyTableTemp.
* Ensures all relevant keys and fields are included.
* Joins region and organization data into KeyTableTemp.
* Loads final data into KeyTable1, ensuring that PLP project flags are correctly assigned.
* Prepares the final KeyTableTemp, combining project types and regions into specific groups.
* Loads all combined data into the final KeyTable, creating a unique \_AuthID.
* Drops all intermediary and temporary tables to clean up the data model.

# Bridge

* Loads distinct combinations of \_AuthID and concatenated fields (\_SA\_PJT\_GROUP, \_SA\_DEV\_SITE, \_SA\_MFG\_SITE, \_Region\_Country, \_Region, \_SA\_PLP\_PROJECT) into the bridge\_table.
* The concatenation uses UPPER to ensure consistency in casing and <ANY> placeholders for wildcard entries.
* Multiple variations of the concatenated fields are loaded using different combinations of these fields, where each combination checks for specific conditions to ensure the correct entries are concatenated.
* Creates the \_bridge table by loading \_AuthID and \_AUTHP from the bridge\_table.
* Loads user data from the GLB\_PLM\_USERS.QVD file.
  + Constructs \_AUTHP using a combination of user-specific fields (SA.PJT\_GROUP, SA.DEV\_SITE, SA.MFG\_SITE, SA.CNTRY, SA.REGION, SA.PLP\_PROJECT).
  + Includes additional fields for user details like UserFirstName, UserName, Login, and USERID.
  + Drops intermediary fields \_SA\_PJT\_GROUP, \_SA\_DEV\_SITE, \_SA\_MFG\_SITE, \_Region\_Country, and \_Region from the data model to clean up the final dataset.