Operator Volume Report

This report had information about Each operator/teams counts and details of ref_ids, container_ids, and customers.
Data: From CW and IE, only year 2024
SQL to get CW data:

```
SELECT
    '', --operation_id populated as hash in python
    'CW', --as [System],
    'A' as RecStatus,
    '',   --as [LOB],
    (SELECT TOP 1 staff.GS_FullName
     FROM CustomsReporting.dbo.Cargowise_OrgStaffAssignments_PROD_REPORTING sa
     JOIN CustomsReporting.dbo.Cargowise_GLBStaff_PROD_REPORTING staff
     ON staff.GS_Code = (
         SELECT TOP 1 O8_GS_NKPersonResponsible
         FROM CustomsReporting.dbo.Cargowise_OrgStaffAssignments_PROD_REPORTING sa_inner
         WHERE sa_inner.O8_OH = sa.O8_OH
         AND sa_inner.O8_Role =
           CASE
             WHEN EXISTS (
               SELECT 1
               FROM CustomsReporting.dbo.Cargowise_OrgStaffAssignments_PROD_REPORTING sa2
               WHERE sa2.O8_OH = sa.O8_OH
               AND sa2.O8_Role = 'CUS'
             ) THEN 'CUS'
             ELSE 'ACT'
           END
         ORDER BY sa_inner.O8_SystemCreateTimeUTC DESC  -- Ensure only one value is selected
       )
     WHERE sa.O8_OH = OH.OH_PK
     ORDER BY sa.O8_SystemCreateTimeUTC DESC) AS operator, --Operator
    '', --customercsr
    CASE WHEN JS_RS_NKServiceLevel = 'AWS' THEN 'All Water'
         WHEN JS_RS_NKServiceLevel = 'MLB' THEN 'MLB'
         ELSE NULL END AS ShipMode, --case statement to return either 'MLB, All Water'
    '', -- as  [DeliveryType]
    jk.JK_UniqueConsignRef as ref_id, --as [REFID], --C OR S USED IN JOINS
    jk.JK_BookingReference, -- as [BookingNo],
    jk.JK_MasterBillNum, --MBL
    JS.JS_HouseBill,
    '', --SONumber
    CAST(OH.OH_PK AS VARCHAR(100)) AS to_company_id,
    OH.OH_FullName as to_company, --as ToC,
    OH.OH_FullName as consignee, -- as Consignee,
    '', --notify
    (SELECT OH_FullName FROM CustomsReporting.[dbo].[Cargowise_ORGHEADER_PROD_REPORTING] OH
WHERE OH.OH_PK = JS.JS_OH_ImportBroker) AS Broker, --broker
    '', --NVO
    '', --shipline
    (SELECT oh.OH_CODE
     FROM CustomsReporting.[dbo].[Cargowise_OrgAddress_PROD_REPORTING] oa
     JOIN CustomsReporting.[dbo].[Cargowise_ORGHEADER_PROD_REPORTING] oh ON oh.OH_PK = oa.OA_OH
     WHERE oa.OA_PK = jk.JK_OA_ShippingLineAddress) AS [SCAC],
    jwConsolMother.JW_Vessel, -- as [Vessel],
    jwConsolMother.JW_VoyageFlight, -- as [Voyage],
```

```sql
    jwConsolETD.JW_Vessel, -- as [Feeder],
    jwConsolETD.JW_VoyageFlight, -- as [FVOY],
    (SELECT RL_PortName + ', ' + RL_RN_NKCountryCode FROM
CustomsReporting.[dbo].[Cargowise_RefUNLOCO_PROD_REPORTING] rl WHERE rl.RL_Code =
jk.JK_RL_NKLoadPort) AS LoadingPort,
    (SELECT RL_PortName + ', ' + RL_RN_NKCountryCode FROM
CustomsReporting.[dbo].[Cargowise_RefUNLOCO_PROD_REPORTING] rl WHERE rl.RL_Code =
JS.JS_RL_NKOrigin), --CW origin
    (SELECT RL_PortName + ', ' + RL_RN_NKCountryCode FROM
CustomsReporting.[dbo].[Cargowise_RefUNLOCO_PROD_REPORTING] rl WHERE rl.RL_Code =
JK_RL_NKDischargePort) AS Destination, --CW dest
    (SELECT RL_PortName + ', ' + RL_RN_NKCountryCode FROM
CustomsReporting.[dbo].[Cargowise_RefUNLOCO_PROD_REPORTING] rl WHERE rl.RL_Code =
jwConsolMother.JW_RL_NKDiscPort), -- as [InPort],
    (SELECT TOP 1 CAST(CE_EntryNum AS VARCHAR(1000)) FROM
CustomsReporting.[dbo].[Cargowise_CUSENTRYNUM_PROD_REPORTING] WHERE CE_EntryType = 'IT' AND
CE_ParentID = JS_PK), -- as [InITNo],
    (SELECT RL_PortName + ', ' + RL_RN_NKCountryCode FROM
CustomsReporting.[dbo].[Cargowise_RefUNLOCO_PROD_REPORTING] rl WHERE rl.RL_Code =
JS.JS_RL_NKDestination) AS FinalDest,
    '', --incoterms
    '', -- as [CstmsRlsdBy],
    '',-- as [EntryNo],
    '0',-- as [CntrRls],
    '', -- as [DetailComment],
    '', -- as [AN_Remark],
    '', -- as [TradecardFCR],
    JK_RL_NKLOadPort AS LoadingPortID,
    JS.JS_RL_NKOrigin AS OriginPort,
    JK_RL_NKDischargePort AS DestinationPort,
    --Dates--
    NULL AS [CarrierBookingSubmitted],
    NULL AS [CargoReadyDate],
    NULL AS [CarrierBookingConfirmed],
    NULL AS [ERD],
    NULL AS [MBLDate], --eDocs tab
    NULL AS [DocCutoffDate],
    NULL AS [CutOfDate],
    NULL AS [PORETA],
    NULL AS [PORATA],
    jwConsolETD.JW_ETD AS loading_etd,
    jwConsolETD.JW_ATD AS [ActDeptDate],
    NULL AS [DocCompleted],
    jwConsolMother.JW_ETA AS [DischargeETA],
    jwConsolMother.JW_ATA AS [ActDischargePort],
    (SELECT MIN(CH_EntrySubmittedDate)
     FROM CustomsReporting.[dbo].[Cargowise_CUSENTRYHEADER_PROD_REPORTING]
     WHERE CH_JE IN (
            SELECT CH_JE
            FROM CustomsReporting.[dbo].[Cargowise_CUSENTRYHEADER_PROD_REPORTING] ch
            JOIN CustomsReporting.[dbo].[Cargowise_JobDeclaration_PROD_REPORTING] je
            ON je.JE_PK = ch.CH_JE
            WHERE je.JE_JS = js.JS_PK
            )
    ) AS [CstmsRlsdDate],
    JK_SystemCreateTimeUtc AS CreateDate
```

```
FROM CustomsReporting.[dbo].[Cargowise_JobConsol_PROD_REPORTING] jk
JOIN CustomsReporting.[dbo].[Cargowise_JobConShipLink_PROD_REPORTING] l ON l.JN_JK = jk.JK_PK
JOIN CustomsReporting.[dbo].[Cargowise_JobShipment_PROD_REPORTING] js ON js.JS_PK = l.JN_JS
JOIN CustomsReporting.[dbo].[Cargowise_JobDocAddress_PROD_REPORTING] E2 ON E2.E2_ParentID =
JS.JS_PK AND E2_AddressType = 'CED'
JOIN CustomsReporting.[dbo].[Cargowise_OrgAddress_PROD_REPORTING] OA ON OA.OA_PK =
E2.E2_OA_Address
JOIN CustomsReporting.[dbo].[Cargowise_ORGHEADER_PROD_REPORTING] OH ON OH.OH_PK = OA.OA_OH
LEFT OUTER JOIN CustomsReporting.[dbo].[cwJWConsolETDViewCustomsReporting] jwConsolETD ON
jwConsolETD.JW_ParentGuid = jk.JK_PK
LEFT OUTER JOIN CustomsReporting.[dbo].[cwJWConsolMotherViewCustomsReporting] jwConsolMother ON
jwConsolMother.JW_ParentGuid = jk.JK_PK
WHERE js.JS_TransportMode = 'SEA'
AND jk.JK_SystemCreateTimeUtc >= '2021-01-01 00:00:00.000'
AND YEAR(jwConsolETD.JW_ETD) = 2024 -- Filter for records where the LoadingETD is in the year 2024
ORDER BY jwConsolETD.JW_ETD ASC;
```

Above SQL is used to map the current working operator name with the account just for CW data.

Python Script:
Execuitng the "operator_volume.ipynb" file will generate a few different files used to fill the data to get the report

Existing Report file:
Existing report file "operator_volume_report.xlsm" is used, alter data in it to get the updated report.
NOTE: the file is the macro-enabled file, to enable it follow the steps in the file "Macro_enabling"
Data to be altered in it: sheet-Data, Details
Update the pivot table data

Steps to get the report:
1. Execute the SQL to get CW data
2. Extract that data to a file named "main_cw_op.xlsx"
3. Start executing the Python script(Overview below)
    a. It will connect to the DB
    b. Read the CW data
    c. Fetch IE data and merge both IE and CW
    d. Columns needed: ['ref_id', 'operator','loading_etd','to_company_id','to_company','consignee']
    e. # Replace operator name where to_company
        'AMERICAN SIGNATURE INC (ASI)', 'operator'] = 'ASI Team'
        'SVP Sewing Brands, LLC', 'operator'] = 'David McKeen'
        'ITG VOMA', 'operator'] = 'Michelle Ford'
        'Igloo Products Corp.', 'operator'] = 'Gage Foster'
    f. Create a data frame to get the count of container and ref_id, this will be saved to a diff file:
        'operator_volume_report.xlsx'
    g. Make a separate data frame to get details of the customer/consignee and details and save it to
        'output_acc.xlsx'
4. Open the existing report file "operator_volume_report.xlsm"
    a. Copy the data from file 'operator_volume_report.xlsx' to Sheet "Data"
    b. Copy the data from file 'output_acc.xlsx' to Sheet "Details"
    c. It will contain few missing data,
        i. Connect with Misti to get details about missing information and update
    d. Refresh and Change the data source for all the pivot tables in both sheets: Count, Customer Volume
5. To get details of ref_ids based on any information, the Python script will generate another file
    "'output_merged_df.xlsx'