**Documentation for SQL Query: NRA Detailed**

The NRA Detailed report is designed to provide a comprehensive view of client activities, their associated business entities, and user-defined fields. This report integrates data from multiple tables and uses advanced DAX logic in Power BI to present insights in a clear and actionable format.

**1. Overview**

The SQL query is constructed to:

* **Purpose of the report:** Combine information from multiple tables to create a detailed report that includes active clients, their associated business activities.

**2. Query Breakdown**

This query is combination of multiple subqueries, each performing a specific task:

* **1.Net Asset Value Query**

**Purpose**: This query calculates the net asset value for clients by aggregating relevant general ledger entries and joining them with client account and status data. The net asset value is a crucial metric for understanding the financial health of each client.

* **Steps**:

1. **GL CTE**:
   * **Filter and Aggregate Data**: The GL CTE filters general ledger entries where the account number starts with ‘4’ or ends with ‘9’. It excludes hidden entries and includes only account numbers between 40000 and 79998. The entries are grouped by client\_no\_ and G\_L\_ACCOUNT\_NO\_, and the amounts are summed.
   * **Convert Data Types**: The account number is cast to an integer to facilitate numerical comparisons.
   * **Exclude Hidden Entries**: Entries marked as hidden (hidden\_entry = 0) are excluded to ensure only visible transactions are considered.
   * **Range Filter**: The query includes only account numbers between 40000 and 79998, further refining the selection to relevant accounts.
   * **Group and Sum**: The entries are grouped by client number (client\_no\_) and account number (G\_L\_ACCOUNT\_NO\_), and the amounts are summed to get the total amount for each account.

* **SQL**

WITH GL AS (

SELECT

client\_no\_,

G\_L\_ACCOUNT\_NO\_,

SUM(Sum\_Amount) AS Sum\_Amount

FROM (

SELECT DISTINCT

Amount AS Sum\_Amount,

client\_no\_,

TRY\_CAST(G\_L\_ACCOUNT\_NO\_ AS INTEGER) AS G\_L\_ACCOUNT\_NO\_,

HIDDEN\_ENTRY

FROM CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_G\_L\_ENTRY

WHERE G\_L\_ACCOUNT\_NO\_ LIKE '4%' OR G\_L\_ACCOUNT\_NO\_ LIKE '%9%'

) AS alpha

WHERE hidden\_entry = 0

AND G\_L\_ACCOUNT\_NO\_ > 40000 AND G\_L\_ACCOUNT\_NO\_ < 79998

GROUP BY client\_no\_, G\_L\_ACCOUNT\_NO\_

)

* **A CTE**:
  + **Join with Client Account Data**: The A CTE joins the aggregated GL data with client account data where the account type is 0. It groups by client and account details to ensure unique records.
  + **Steps**:
    1. **Join Tables**: The aggregated data from the GL CTE is joined with the JERSEY\_CLIENT\_ACCOUNT table on the client number (client\_no\_).
    2. **Filter Data**: Only accounts where the account type (account\_type) is 0 are included, ensuring that only active accounts are considered.
    3. **Group Data**: The data is grouped by client and account details to ensure unique records for each client-account combination.
* **SQL**

A AS (

SELECT

Sum\_Amount,

CA.client\_no\_,

G\_L\_ACCOUNT\_NO\_,

NO\_,

ACCOUNT\_TYPE

FROM GL

JOIN CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_ACCOUNT CA

ON GL.client\_no\_ = CA.client\_no\_

WHERE CA.account\_type = 0

GROUP BY CA.client\_no\_, G\_L\_ACCOUNT\_NO\_, NO\_, ACCOUNT\_TYPE, Sum\_Amount

)

* **B CTE**:
  + **Join with Client Status Data**: The B CTE joins client data with general ledger setup data. It filters out clients with status 3 and temporary clients, and adds descriptive columns for client status and dissolution status using CASE statements.
  + **Steps**:
    1. **Join Tables**: The JERSEY\_CLIENT table is joined with the JERSEY\_CLIENT\_GENERAL\_LEDGER\_SETUP table on the client number (client\_no\_).
    2. **Filter Data**: Clients with status 3 (Closed) and temporary clients are excluded to focus on active and relevant clients.
    3. **Add Descriptive Columns**: CASE statements are used to add columns for client status (Client\_Status) and dissolution status (DissolvedORStruck\_Off), providing a clear description of each client’s current state.
* **SQL**

B AS (

SELECT

JC.client\_no\_,

clientanalysisa,

clientanalysisb,

clientanalysisc,

clientanalysisd,

clientanalysise,

clientanalysisf,

dissolved\_struck\_off,

status,

GLS.client\_no\_ AS Right\_Client\_No,

CASE

WHEN status = 0 THEN 'Prospective'

WHEN status = 1 THEN 'Live'

WHEN status = 2 THEN 'Terminal'

WHEN status = 3 THEN 'Closed'

ELSE ''

END AS Client\_Status,

CASE

WHEN dissolved\_struck\_off = 0 THEN 'N/A'

WHEN dissolved\_struck\_off = 1 THEN 'Dissolved'

WHEN dissolved\_struck\_off = 2 THEN 'Struck off'

ELSE ''

END AS DissolvedORStruck\_Off,

lcy\_code

FROM CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT JC

LEFT JOIN CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_GENERAL\_LEDGER\_SETUP GLS

ON JC.client\_no\_ = GLS.client\_no\_

WHERE status <> 3 AND NOT CONTAINS(JC.client\_no\_, 'TEMP')

)

* **Final Select**:
  + **Combine Results**: The final select statement joins the results of CTE A and B. It selects distinct client details, net assets, and other client analysis fields.
  + **Steps**:
    1. **Join CTEs**: The results of CTE A and B are joined on the client number (client\_no\_).
    2. **Select Columns**: The query selects distinct client details, net assets, and other client analysis fields, ensuring that each client is represented uniquely
* **SQL**

SELECT distinct

A.Client\_No\_,

Sum\_Amount AS Net\_Assets,

Client\_Status,

DissolvedorStruck\_off,

ClientAnalysisA AS Administrator,

ClientAnalysisB AS Manager,

ClientAnalysisC AS Director,

ClientAnalysisD AS Accountant,

ClientAnalysisE AS Client\_Group,

ClientAnalysisF AS Solution\_Line,

Right\_Client\_No,

LCY\_Code

FROM A

LEFT JOIN B ON A.CLIENT\_NO\_ = B.CLIENT\_NO\_

* **Client Extract Terminal Query**

**Purpose**: This query extracts details of terminal clients, including their status and analysis fields. It provides a focused view of clients who are in the terminal phase, which is crucial for understanding their current state and planning future actions.

* **Steps**:

1. **Select Client Details**:
   * **Columns**: Select client number, type, name, date of exit, and client analysis fields. The query selects client number (client\_no\_), type (type), name (name), date of exit (DATE\_OF\_EXIT), and client analysis fields (clientanalysisb, clientanalysisc, clientanalysisd, clientanalysise, clientanalysisf).
   * **Format Date**: Convert DATE\_OF\_EXIT to NULL if it is ‘1753-01-01’. The DATE\_OF\_EXIT is converted to NULL if it is ‘1753-01-01’, ensuring that only valid dates are displayed.
     1. **Add Descriptive Columns**: Add a column for client status using a CASE statement. A CASE statement is used to add a column for client status (Client\_Status), providing a clear description of each client’s current state.

* **SQL**

SELECT distinct

JC.CLIENT\_NO\_,

JC.type,

JC.name,

CASE WHEN CAST(DATE\_OF\_EXIT AS DATE) = '1753-01-01' THEN NULL ELSE CAST(DATE\_OF\_EXIT AS DATE) END AS DATE\_OF\_EXIT,

clientanalysisb AS Manager,

clientanalysisc AS Director,

clientanalysisd AS Account,

clientanalysise AS Client\_Grp,

clientanalysisf AS Solution\_line,

CASE

WHEN Status = 0 THEN 'Prospective'

WHEN Status = 1 THEN 'Live'

WHEN Status = 2 THEN 'Terminal'

WHEN Status = 3 THEN 'Closed'

ELSE 'Null'

END AS Client\_Status

FROM CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT JC

WHERE JC.client\_no\_ NOT IN ('JEC03200','JEC03204','JEC01817','JEC01823','JEC04353')

AND status = 2

ORDER BY JC.client\_no\_ ASC

* **Business Activity Query**

**Purpose**: This query links business activities with client details, providing insights into the risk ratings and descriptions of business activities. It helps in understanding the business activities each client is involved in and their associated risks.

* **Steps**:
  + **Subquery A**: The subquery joins business activity types with client business activity data to provide detailed information about each business activity.
  + **Join Tables**: Join JERSEY\_BUSINESS\_ACTIVITY\_TYPES with JERSEY\_CLIENT\_BUS\_\_ACTIVITY\_TYPES on BUSINESS\_ACTIVITY.
  + **Select Columns**: Select business activity code, description, risk rating, and client number.
  + **SQL Code**:
* **SQL**

(SELECT

BAT.code,

BAT.description,

BAT.risk\_rating,

CAT.CLIENT\_NO\_,

CAT.BUSINESS\_ACTIVITY

FROM CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_BUSINESS\_ACTIVITY\_TYPES BAT

JOIN CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_BUS\_\_ACTIVITY\_TYPES CAT

ON CAT.BUSINESS\_ACTIVITY = BAT.code) AS A

1. **Subquery B**:
   * **Select Client Details**: Select client details from JERSEY\_CLIENT.
   * **Add Descriptive Columns**: Add columns for client status and analysis fields using CASE statements.
   * **SQL Code**:

* **SQL**

(SELECT

JC.Client\_No\_,

JC.Type,

JC.Name,

JC.Status,

JC.ClientAnalysisA AS Admin,

JC.ClientAnalysisB AS Manager,

JC.ClientAnalysisC AS Director,

JC.ClientAnalysisD AS Account,

JC.ClientAnalysisE AS Client\_Group,

JC.ClientAnalysisF AS Solution\_Line,

CASE

WHEN JC.Status = 0 THEN 'Prospective'

WHEN JC.Status = 1 THEN 'Live'

WHEN JC.Status = 2 THEN 'Terminal'

WHEN JC.Status = 3 THEN 'Closed'

ELSE 'Null'

END AS Client\_Status

FROM CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT JC) AS B

1. **Final Select**:
   * **Join Subqueries**: Join the results of Subquery A and B on client\_no\_.
   * **Select Columns**: Select business activity details and client details.
   * **SQL Code**:

* **SQL**

SELECT

A.code,

A.description,

A.risk\_rating,

A.CLIENT\_NO\_ AS BusinessActivityClientNo,

A.BUSINESS\_ACTIVITY,

B.Client\_No\_,

B.Type,

B.Name,

B.Status,

B.Admin,

B.Manager,

B.Director,

B.Account,

B.Client\_Group,

B.Solution\_Line,

B.Client\_Status

FROM A

JOIN B ON A.CLIENT\_NO\_ = B.Client\_No\_

* **4. Total Assets Value Query**

**Purpose**: This query calculates the total assets for clients by aggregating relevant general ledger entries and joining them with client account and status data.

1. **Steps**:
2. Filters and aggregates general ledger entries where G\_L\_ACCOUNT\_NO\_ starts with ‘4’ or ends with ‘9’.
3. Ensures hidden\_entry is 0 and G\_L\_ACCOUNT\_NO\_ is between 40000 and 59997.
4. Groups by client\_no\_ and G\_L\_ACCOUNT\_NO\_ to get the sum of amounts.

* **A CTE:**

1. Joins the aggregated GL data with client account data where account\_type is 0.
2. Groups by client and account details.

* **B CTE**:

1. Joins client data with general ledger setup data.
2. Filters out clients with status 3 and temporary clients.
3. Adds descriptive columns for client status and dissolution status.

* **Final Select**:

1. Joins the results of CTE A and B.
2. Selects distinct client details, total assets, and other client analysis fields.
3. **Power BI Logic Overview:**

* Rename Columns: Ensure all columns have meaningful names.
* Change Data Types: Convert columns to appropriate data types (e.g., dates, numbers).
* Remove Unnecessary Columns: Remove columns that are not needed for analysis.
* Filter Rows: Apply any necessary filters to clean the data.
* **Data Modeling**
* **Steps:**

1. **Define Relationships:**
   * Go to the Model view.
   * Create relationships between tables based on key columns (e.g., client\_no\_).
2. **Create Calculated Columns and Measures:**
   * Calculated Columns: Add columns for derived data (e.g., Client\_Status, DissolvedORStruck\_Off).
   * Measures: Create measures for aggregations (e.g., Total Assets, Net Assets).

Example Calculated Column:

Client\_Status = SWITCH(

[status],

0, "Prospective",

1, "Live",

2, "Terminal",

3, "Closed",

"Unknown"

)

**Example Measure:**

Total\_Assets = SUM(GL[Sum\_Amount])

1. **Page 1: Overview**
   * Title: “Client Financial Overview”
   * Visualizations: Summary of net assets, total assets, and client status distribution.
2. **Page 2: Net Asset Value**
   * Title: “Net Asset Value Analysis”
   * Visualizations: Bar chart of net assets by client, slicers for client status and dissolution status.
3. **Page 3: Client Extract Terminal**
   * Title: “Terminal Clients”
   * Visualizations: Table of terminal client details, filters for specific client groups.
4. **Page 4: Business Activity**
   * Title: “Business Activity Analysis”
   * Visualizations: Pie chart of business activity risk ratings, table of business activity and client details.
5. **Page 5: Total Assets Value**
   * Title: “Total Assets Analysis”
   * Visualizations: Bar chart of total assets by client, slicers for client status and dissolution status.

**Conclusion**

This SQL query provides a comprehensive and detailed report on the Detailed of specific clients. It integrates multiple sources, cleanses data, handles exceptions, and formats the final output to meet regulatory requirements.