**Documentation for SQL Query: NRA TCB**

The NRA TCB 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:

* **Client Subquery:**Focuses on active clients, pulling essential details like client number, name, status, and type.  
  **Purpose:** Establishes a foundation of active clients as the starting point for the analysis.
* **Entity Subquery (ent):**Retrieves information on active entities associated with each client.  
  **Purpose:** Links clients with their corresponding entities to provide a broader view of the client's operations.
* **User-Defined Fields Subquery (udf):**Extracts custom fields related to each client that are currently active.  
  **Purpose:** Adds additional, customizable data points that might be critical for specific business needs.
* **Responsibility Subquery (res):**Gathers data on the types of responsibilities associated with each client.  
  **Purpose:** Provides insights into the responsibilities held by or for the client, giving a more comprehensive understanding of the client's obligations.
* **Business Activity Subquery (bus):**Captures details of completed business activities related to each client.  
  **Purpose:** Adds historical context by showing what business activities have been completed, which could impact current client status or future strategies.
* **Final Query:**Combines all the above subqueries using LEFT JOINs, ensuring that all clients are included even if they do not have related entities, user-defined fields, responsibilities, or business activities.  
  **Purpose:** Creates a detailed and comprehensive dataset that includes all relevant client information, providing a complete view for analysis.

**3. Power BI Logic Overview:**

* **First Logic (Question\_Legal):**
  + **Formula:** CONCATENATE("sa1", sa1[legal\_entity])
  + **Purpose:** This concatenation creates a unique identifier for legal entities associated with the question "sa1." It helps in associating the legal entities with specific questions, facilitating easier filtering and analysis in reports.
* **Second Logic (Questionnaire Legal):**
  + **Formula:** CONCATENATE(questionnaire[questions\_no], questionnaire[legal\_entity])
  + **Purpose:** Similar to the first logic, but applied to the questionnaire table. This helps in linking legal entities to their respective questions, ensuring that the data is accurately represented in the analysis.
* **Third Logic (sa1 Calculation):**
  + **Formula:**
    - The logic checks if "sa1" is the selected question.
    - It then counts the legal entities under various conditions:
      * **Trusts**
      * **Jersey Companies**
      * **Non-Jersey Companies**
      * **Other Jersey Companies**
      * **Other Non-Jersey Companies**
    - **Purpose:** This logic provides a detailed breakdown of the legal entities associated with the question "sa1", categorized by their jurisdiction and type. This allows for a deeper understanding of the data, revealing patterns and trends based on location and entity type.
* **Final Logic (Answer Calculation):**
  + **Formula:** IF(questionnaire[questions\_no] = "sa1", sa1[sa1])
  + **Purpose:** This logic populates the answer field based on the selected question. If "sa1" is the question, it displays the results from the third logic. This provides a direct, user-friendly output for stakeholders to interpret the data.
* **Dynamic Interaction:**
  + The dashboard allows users to filter and view data based on specific questions (like 'sa1'). As users interact with the report, the visuals dynamically update to show relevant legal entity counts and corresponding answers.
  + Power BI logic categorizes legal entities by jurisdiction and type, offering granular insights that can inform business decisions.

**4.Query:**

with CLIENT AS

(

select

JC.CLIENT\_NO\_,

JC.type,

[JC.name](http://jc.name/),

JC.global\_dimension\_1\_code as Legal\_entity,

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,

JC.jurisdiction,

JC.STATUS,

JC.ClientAnalysisE as   Client\_Group,

JC.ClientAnalysisF as Solution\_Line,

JC.last\_date\_modified,

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

Legal\_Entity is not null and Status in (1,2) and date\_of\_exit ='1753-01-01 00:00:00.000'

and Legal\_Entity != 'LASFJ' and Legal\_Entity != 'LCSJ' and not contains(JC.client\_no\_,'TEMP')

),

ENT AS

(

select

JCE.ENTITY\_NO\_, JCE1.ENTITY\_NO\_ as Resource\_Entity\_No\_ ,

JCE1.ENTITY\_TYPE AS Resource\_ENTITY\_TYPE,

CASE WHEN CAST(JCE1.date\_appointed AS DATE) = '1753-01-01' THEN NULL ELSE CAST(JCE1.date\_appointed AS DATE) END AS Resource\_DATE\_APPOINTED,

CASE WHEN CAST(JCE1.date\_resigned AS DATE) = '1753-01-01' THEN NULL ELSE CAST(JCE1.date\_resigned AS DATE) END AS Resource\_Date\_Resigned,

JCE.client\_no\_ ,

JCE.ENTITY\_TYPE,

JCE.ENTITY\_NAME,

JCE.ENTITY\_TYPE\_DESCRIPTION,

CASE WHEN CAST(JCE.date\_appointed AS DATE) = '1753-01-01' THEN NULL ELSE CAST(JCE.date\_appointed AS DATE) END AS DATE\_APPOINTED,

CASE WHEN CAST(JCE.date\_resigned AS DATE) = '1753-01-01' THEN NULL ELSE CAST(JCE.date\_resigned AS DATE) END AS Date\_Resigned

from CLIENT JC

join CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_ENTITY JCE on JCE.client\_no\_ = JC.client\_no\_

LEFT join CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_ENTITY JCE1 on JCE.client\_no\_ = JCE1.ENTITY\_NO\_

and CAST(JCE.date\_appointed AS DATE) <= '2019-01-31'

),

UDF AS(

select

UDF.CODE,

UDF.description as DNAV\_Code,

[UDFD.DATA](http://udfd.data/),

udfd.linked\_to\_no\_,

udf.status

from ENT C

JOIN CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_ZURICH\_USER\_DEFINED\_FIELD\_DATA UDFD ON udfd.linked\_to\_no\_ = C.CLIENT\_NO\_

JOIN CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_USER\_DEFINED\_FIELD UDF ON UDFD.code = UDF.code -- where UDF.code = 'RGO'

),

RES AS (

select DISTINCT

CR.CLIENT\_NO\_,

CR.type,

CR.template\_code,

CR.TEMPLATE,

CR.SERVICE\_DIMENSION\_1,

CR.SERVICE\_DIMENSION\_2,

CR.SERVICE\_DIMENSION\_3,

CR.SERVICE\_DIMENSION\_4,

CR.SERVICE\_DIMENSION\_5,

CASE WHEN CAST(CR.Start\_Date AS DATE) = '1753-01-01' THEN NULL ELSE CAST(CR.Start\_Date AS DATE) END AS Start\_Date,

CASE WHEN CAST(CR.End\_date AS DATE) = '1753-01-01' THEN NULL ELSE CAST(CR.End\_date AS DATE) END AS End\_date,

RT.AEOI\_SERVICE,

from CLIENT C JOIN

CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_RESPONSIBILITIES CR ON CR.CLIENT\_NO\_ = C.CLIENT\_NO\_

join CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_RESPONSIBILITY\_TYPE RT on CR.responsibility\_type = RT.code),

BUS as (

select

CBA.Client\_No\_,

CBA.BUSINESS\_ACTIVITY,

BAT.Description

from CLIENT C

JOIN CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_BUS\_\_ACTIVITY\_TYPES CBA ON C.CLIENT\_NO\_ = CBA.CLIENT\_NO\_

join CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_BUSINESS\_ACTIVITY\_TYPES BAT on CBA.BUSINESS\_ACTIVITY = BAT.Code

where CBA.BUSINESS\_ACTIVITY is not NULL and Len(CBA.BUSINESS\_ACTIVITY) <> 0),

RRH AS

(select

a.Client\_No\_,

a.last\_rating,

a.last\_risk\_assesment\_date,

a.risk\_rating

from CLIENT C

JOIN (select

no\_ as Client\_No\_,risk\_weighting,DATE as last\_risk\_assesment\_date,risk\_rating, current\_rating as last\_rating

from CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_RISK\_RATING\_HISTORY

where current\_rating =1 and date <= '2023-12-31' and risk\_rating is not null)a ON C.CLIENT\_NO\_ = A.CLIENT\_NO\_

join

(select no\_ as Client\_No\_, Max(date) as curr\_risk\_assesment\_date, current\_rating from CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_RISK\_RATING\_HISTORY

where current\_rating != 1 Group by no\_,current\_rating)b on a.Client\_No\_=b.Client\_No\_),

GL as (

Select distinct SUM(Amount) as Sum\_Amount , client\_no\_ ,max(G\_L\_ACCOUNT\_NO\_) as G\_L\_ACCOUNT\_NO\_

from (

SELECT CLIENT\_NO\_,try\_cast(G\_L\_ACCOUNT\_NO\_ as integer) AS G\_L\_ACCOUNT\_NO\_,aMOUNT 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 '5%') and POSTING\_DATE <= '31-Dec-2023'

and aMOUNT <> 0) as alpha

where G\_L\_ACCOUNT\_NO\_> 40000 AND G\_L\_ACCOUNT\_NO\_ <= 59998 and G\_L\_ACCOUNT\_NO\_ not Like '%[a-z]%'

group by client\_no\_ --,G\_L\_ACCOUNT\_NO\_

),

LE AS(

select DISTINCT CRE.STARTING\_DATE,

E.Client\_No\_,E.Sum\_Amount as Sum\_Amount,L.LCY\_Code as Currency,CRE.EXCHANGE\_RATE\_AMOUNT, CRE.CURRENCY\_CODE,E.G\_L\_ACCOUNT\_NO\_

from GL E

LEFT OUTER join CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CLIENT\_GENERAL\_LEDGER\_SETUP L on L.CLIENT\_NO\_ = E.client\_no\_

join CLEANSING\_ZONE.JER\_NAVONE\_DBO.CZ\_JER\_NAVONE\_DBO\_\_JERSEY\_CURRENCY\_EXCHANGE\_RATE CRE on L.LCY\_Code = CRE.currency\_code AND CRE.STARTING\_DATE = '2019-12-31'

)

--SELECT \* FROM le

select distinct

A.Legal\_entity,A.CLIENT\_NO\_, A.type, [A.name](http://a.name/),A.Client\_Status,A.Client\_Group,  A.Solution\_Line, A.DATE\_OF\_EXIT, A.jurisdiction,A.STATUS, ENT.Resource\_Entity\_No\_,ENT.rESOURCE\_DATE\_APPOINTED,     ENT.rESOURCE\_DATE\_RESIGNED,ENT.Resource\_ENTITY\_TYPE,  ENT.DATE\_APPOINTED,     ENT.DATE\_RESIGNED,A.last\_date\_modified,ENT.ENTITY\_NO\_,   ENT.ENTITY\_TYPE,    ENT.ENTITY\_NAME,    ENT.ENTITY\_TYPE\_DESCRIPTION,

B.CODE, B.DNAV\_Code, [B.DATA](http://b.data/),B.linked\_to\_no\_ as UDF\_Client\_no\_,

C.type as Res\_Type,C.template\_code,C.TEMPLATE,C.SERVICE\_DIMENSION\_1,C.SERVICE\_DIMENSION\_2,C.SERVICE\_DIMENSION\_3,C.SERVICE\_DIMENSION\_4,C.SERVICE\_DIMENSION\_5,C.AEOI\_SERVICE,C.start\_date, C.End\_date ,

D.Client\_No\_ AS RIGHT\_Client\_No\_, D.BUSINESS\_ACTIVITY, D.Description,

E.last\_risk\_assesment\_date, E.risk\_rating,

LE.Sum\_Amount as account\_balance, LE.Currency as Acc\_Currency, LE.EXCHANGE\_RATE\_AMOUNT, LE.CURRENCY\_CODE,LE.Sum\_Amount / LE.EXCHANGE\_RATE\_AMOUNT as GBP\_Sum\_Amount,LE.G\_L\_ACCOUNT\_NO\_

from

Client A

left join ENT on A.Client\_No\_ = ENT.Client\_No\_

left join UDF B on A.Client\_No\_ = B.linked\_to\_no\_

left join RES C on A.Client\_No\_ = C.Client\_No\_

LEFT join BUS D on A.Client\_No\_ = D.Client\_No\_

left join RRH E on A.Client\_No\_ = E.Client\_No\_

left join LE on A.Client\_No\_ = LE.Client\_No\_

--WHERE A.Client\_No\_ in( 'JEC00195','JEC00203')

**Conclusion**

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