|  |
| --- |
| Inland Revenue |

**Data Intelligence Platform**

**Refined Layer - Account Model**

**Transformation Rules**

**Prepared by:** Ray Duncan

**Version:** v3.17

**Version Date:** 19/02/2020

**IN REVIEW**

**About this document**

This document is designed to provide a set of outline principles, key information, standards and guidance for the clear and unambiguous definition of transmission and transformation, of operational data from core IR data and intelligence sources, into the DIP physical Party model.

It assumes the transformation logic will be that data will be transmitted from EDW and START as the data ‘**sources**’ to the DIP data model as a ‘**target**’.

This is a living document for the duration of the iterative development and will be used to guide the build and transformation logic of the full DIP data model. Therefore, this document will be reviewed and updated with each data model element update.

**Document control**

|  |  |
| --- | --- |
| **File name and path** | <https://irnz.sharepoint.com/:w:/s/DataIntelligenceTeam/EX-ae7nWwugxFr7fImioV_uwBVK8-HFUmfkTT5mee7h2Ekw?e=2Eirge> |
|  |  |
| **Contact person** | Ally Rutherford |
| **Status** | FINAL |

**Document review history**

| **No.** | **Date** | **Change Description** | **Contact** |
| --- | --- | --- | --- |
| 0.01 | 29/07/2019 | Document creation from PMO template. | Ray Duncan |
| 1.00 | 13/08/2019 | Initial version for Start – EDW to follow | Ray Duncan |
| 2.00 | 02/10/2019 | Miscellaneous changes | Ray Duncan |
| 3.00 | 08/10/2019 | Various updates to START sections | Scott Walkinshaw |
| 3.01 | 09/10/2019 | Retain inactive flngver=0 records until step#6. Rewrite NameRecord joins for hierarchy ‘level’. | Ray Duncan |
| 3.02 | 16/10/2019 | Derive name records via join to accounts in step#1 before de-duplicating in step#2. | Ray Duncan |
| 3.03 | 17/10/2019 | Derive indicators via account/profile in step#1 before de-duplicating in step#2. | Ray Duncan |
| 3.04 | 18/10/2019 | Specify cut-off boundaries for single customer timeline. New Step#6b to compress the post-merge timelines. New logic for Step#5 and filters for Step#6a. | Ray Duncan |
| 3.05 | 21/10/2019 | EDW SQL and Steps 1-7. Waiting on business input on EDW-Start code mappings for Account (Tax) Types, filing frequencies, cessation reasons and exemption reasons. | Richard Holley |
| 3.06 | 31/10/2019 | Changed Step#6a filtering rules for business active date ranges - also avoid high-date issue when timestamp offset is > minute. No change for “Re-opened” GST Accounts. | Ray Duncan |
| 3.07 | 01/11/2019 | Updated EDW-Start code mappings for filing frequencies. | Ray Duncan |
| 3.08 | 04/11/2019 | Use filing frequency period instead of filing frequencies for Step#7 transform rules and EDW-Start code mappings. Remove column filing\_frequency\_period. | Ray Duncan |
| 3.09 | 05/11/2019 | Profile EDW-Start code mappings to complete ‘missing’ cessation and exemption reasons | Ray Duncan |
| 3.10 | 06/11/2019 | GST Accounts for EDW with code mappings for accounting basis and registration circumstance. | Ray Duncan |
| 3.11 | 08/11/2019 | First cut of EDW Accounts complete. In Review. | Ray Duncan |
| 3.12 | 11/11/2019 | Changes in response to first round of business review. | Ray Duncan |
| 3.13 | 12/11/2019 | Five new derived tables for EDW for CPR/NCP/CSE, SLE, KSE, KSR, SSC tax\_types for cross-account checks. Corrected DIP\_ACCOUNT\_ID and tax\_code mappings (see notes in Appendix C and I for details) | Ray Duncan |
| 3.14 | 13/11/2019 | Changes to EDW Step#7 rules in response to business review part 2. Included heritage\_location\_number in account SK mapping. Build changes. | Ray Duncan  Scott Walkinshaw |
| 3.15 | 14/01/2020 | Post-build review changes. | Ray Duncan |
| 3.16 | 30/01/2020 | Small changes to Legal\_Name derivation. | Marc Pearce |
| 10/02/2020 | Use derived field “number\_of\_valid\_days” to rank ‘duplicate’ records when relevant. | Ray Duncan |
| 3.17 | 19/02/2020 | Include standard Unknown and N/A rows for all dimensions to join to Fact tables. Replace “number\_of\_valid\_days” with “number\_of\_valid\_seconds” to rank ‘duplicate’ records. | Ray Duncan |

Contents

[1 Introduction 6](#_Toc33002183)

[1.1 Background 6](#_Toc33002184)

[1.2 Document Purpose 6](#_Toc33002185)

[1.3 Audience 6](#_Toc33002186)

[1.4 Reference Documents 6](#_Toc33002187)

[2 Data Model 7](#_Toc33002188)

[2.1 Account Model – Account Type Layers. 7](#_Toc33002189)

[2.2 Account Model - Conformed Dimensions 7](#_Toc33002190)

[2.3 DIP Account Level Table view 8](#_Toc33002191)

[3 Processing Overview 9](#_Toc33002192)

[4 Processing Sequence 10](#_Toc33002193)

[4.1 Processing Sequence 10](#_Toc33002194)

[4.2 EDW-START Cut-Off 10](#_Toc33002195)

[4.3 Raw Layer tables from START 10](#_Toc33002196)

[4.4 Raw Layer tables from EDW (FIRST) 11](#_Toc33002197)

[5 Mapping Tables Required for Processing EDW & START 11](#_Toc33002198)

[5.1 EDW-START Code Map 11](#_Toc33002199)

[5.2 Start Account Key – IRD Number Map 12](#_Toc33002200)

[6 Transform Rules: EDW 14](#_Toc33002201)

[Step 1 – Selecting Valid Values from Source Tables 14](#_Toc33002202)

[6.1 Step 2 – De-duplication of Records 18](#_Toc33002203)

[6.2 Step 3 – Compressing the timeline 20](#_Toc33002204)

[6.3 Step 4 – Combining all change timeline 22](#_Toc33002205)

[6.4 Step 5 – Create a single timeline for each Account 23](#_Toc33002206)

[6.5 Step 6a – Merge change timelines with Account timeline 24](#_Toc33002207)

[6.6 Step 6b – Compress the Post-Merge Timelines 26](#_Toc33002208)

[6.7 Step 7 – Transform Rule Logic 27](#_Toc33002209)

[7 Transform Rules: START 52](#_Toc33002210)

[7.1 Step 1 – Selecting Valid Values from Source Tables 52](#_Toc33002211)

[7.2 Step 2 – De-duplication of Records 56](#_Toc33002212)

[7.3 Step 3 – Compressing the timeline 58](#_Toc33002213)

[7.4 Step 4 – Combining all change timeline 63](#_Toc33002214)

[7.5 Step 5 – Create a single timeline for each Account 65](#_Toc33002215)

[7.6 Step 6a – Merge change timelines with Account timeline 65](#_Toc33002216)

[7.7 Step 6b – Compress the Post-Merge Timelines 68](#_Toc33002217)

[7.8 Step 7 – Transform Rule Logic 77](#_Toc33002218)

[8 Appendix A – Acronyms, Abbreviations and Key Terms 101](#_Toc33002219)

[9 Appendix B – EDW-Start Cut-off 103](#_Toc33002220)

[10 Appendix C – Common Natural Key for DIP Accounts 107](#_Toc33002221)

[11 Appendix D – Standard De-Duplication Process 108](#_Toc33002222)

[12 Appendix E – Example SQL for Context – EDW 111](#_Toc33002223)

[13 Appendix F – Example SQL for Context – Start 134](#_Toc33002224)

[14 Appendix G – Conceptual Data Model – Start Accounts 165](#_Toc33002225)

[15 Appendix H - Conceptual Data Model – EDW Accounts 166](#_Toc33002226)

[16 Appendix I – EDW-Start Code Mappings 167](#_Toc33002227)

# Introduction

## Background

Inland Revenue (IR) is delivering a new Data and Intelligence Platform (DIP) as part of the Business Transformation Programme and a move to becoming an Intelligence-led organisation.

The DIP will be pivotal in enabling IR to make better decisions faster, and to improve outcomes for customers.

The DIP will have a pivotal role in exposing consistent, cleansed and integrated data to IR staff to enable the concept of being “intel-led” to become embedded throughout the organisation.

## Document Purpose

To detail the information and processing required to populate the new IR DIP Data Model, starting at the Account level.

The core is a set of transformation instructions determining how to convert the structure and content of raw data layer generated by regular extracts from START and EDW (FIRST) to the required structure and content needed in the DIP (Target) refined data layer.

## Audience

This document is primarily intended for to be used by data integration developers within the Platform & Data team to develop the SAS Data Integration Studio packages that will process the data from the raw layer to the refined layer.

A secondary purpose is to allow any user including data scientists and data engineers to understand how the raw source system data was cleansed, integrated and transformed into the refined layer model.

## Reference Documents

The following deliverables, reviews and open source information can be considered when reading this document:

* IRD DIP - [Understanding and Navigating the START database](https://irnz.sharepoint.com/:w:/s/DataIntelligenceTeam/EfW98mcCJXBKrv5PhWKvmVcBJ7445kWMbwtqah-aKLcGIw?e=FcBCls)
* IRD DIP – Data Modelling Standards – ***To be confirmed***
* IRD DIP - DIP\_Account (DIP\_Data\_Model) Oracle SQL Developer | Data Modeller (Figure 2)
* IRD DIP – Standard transform and formatting rules for dates, numeric’s, categorical; treatment of missing values etc
* IRD DIP - Slowly Changing Dimension (SCD) Management Rules.
* IRD DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487)

# Data Model

The Refined Layer Data Model has been designed specifically for use with the DIP. It is aligned with the concept of Dimensional Modelling in that some tables are clearly dimensions and others clearly fact tables. However, where dimension to dimension relationships are held over time, the model generally uses a relationship table holding the natural keys rather than the dimensional surrogate keys to reduce processing needs and make the data more accessible.

The Data model is designed around an ‘Account Concept’. Though many account types are held in Start /EDW, the data model is only concerned with 4 generic account types i.e. PAY (Payroll – employment activities), PIE (Portfolio Investment Entity), INC (Income Tax) and GST.

## Account Model – Account Type Layers.

Account

PAY

PIE

INC

GST

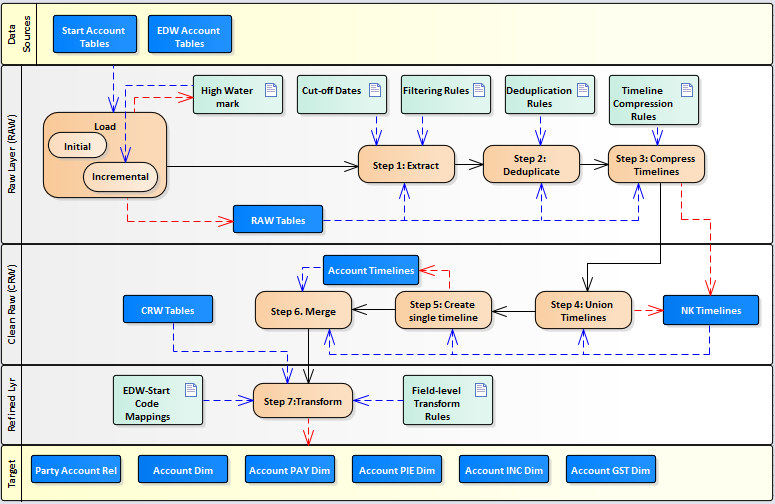
Figure 1: Initial Level - Account Model Design

## Account Model - Conformed Dimensions

Tables ACCOUNT\_DIM, ACCOUNT\_PAY\_DIM, ACCOUNT\_INC\_DIM, ACCOUNT\_GST\_DIM AND ACCOUNT\_PIE\_DIM are “conformed dimensions” with ‘common’ attributes having identical column names and data type domains. Information may be combined in a single report by using conformed dimension attributes that are associated with each fact table. When a ‘conformed’ attribute is used as the row header (that is, the grouping column in the SQL query), the results can be aligned on the same rows in a drill-across report.

## DIP Account Level Table view

# Processing Overview



**Step 1**: Extract each source table/view and apply filtering criteria to exclude:

1. Invalid records or records types not required
2. Changes before high-water-mark
3. START records expiring before Cut-Off (initial load only)
4. EDW records effective after Cut-Off (initial load only)

**Step 2**: De-duplicate each source to produce a serialized timeline for each natural key (NK).

**Step 3**: Compress each NK timeline from (2) to 'merge' contiguous duplicates of the specified columns within the specified time stamps for each natural key

**Step 4**: Union the NK timelines from (3) for each account to provide a basis for creating change records

**Step 5**: Create a single timeline for each account from (4)

**Step 6a**: Merge (4) with (5) for each account to ensure correctly versioned for step (5)

**Step 6b:** Compress the post-merge timelines from (6a)

**Step 7**: Apply the field-level transformation rules and populate the Refined Layer

# Processing Sequence

## Processing Sequence

The Account level model is one of the foundation layers of the DIP data model. It is generated by <40 common fields found in various Account Type layers and so processing is required to follow in the below order:

| **#** | **Source** | **Ingestion Details** |
| --- | --- | --- |
| 1 | START | Initial ingestion of Account Types (PAY, PIE, INC, GST) with cut-off at EDW-START Cut-Off (see Section 4.2 for details). This must run first to create the Start Account Key lookup (se Sevtion 5.2) |
| 2 | EDW | One-off ingestion of Account Types (PAY, PIE, INC, GST) with cut-off at EDW-START Cut-Off (see Section 4.2 for details) |
| 3 | START | Ongoing incremental ingestion of Account domain after initial load |

## EDW-START Cut-Off

EDW records which correspond to Start records which have record\_expiry\_date after EDW-START Cut-Off, set the record\_active\_flag to ‘N’ and set the record\_expiry\_date to $$TSDIFF prior to the EDW-START Cut-Off (see Appendix A for definitions). Otherwise, leave the record unchanged.

Start accounts were migrated by releases R1, R2 and R3. For each release an EDW-Start Cut-off date is specified to avoid intra-day gaps between contiguous EDW and Start data. See Appendix B for details.

|  |  |  |  |
| --- | --- | --- | --- |
| **Start Release** | **Start Release Dates** | **Cut-Off Placeholder** | **Cut-Off Date** |
| R1 | 3-Feb-2017 to 4-Feb-2017 | $$EDWTS1 | 5-Feb-2017 |
| R2 | 13-Apr-2018 | $$EDWTS2 | 14-Apr-2018 |
| R3 | 20-Apr-2019 | $$EDWTS3 | 21-Apr-2019 |

Prior to loading the tables into the **Account** area, the following raw tables are required:

## Raw Layer tables from START

|  |
| --- |
| app\_tblAccount |
| app\_tblCustomer |
| app\_tblcustomerlevel |
| app\_tblid |
| app\_tblIndicator |
| app\_tblProfile |
| app\_tblNameRecord |
| app\_tblNZ\_AccGSTInfo |
| app\_tblNZ\_AccIncome |
| app\_tblNZ\_AccIPE |
| app\_tblNZ\_AccITN |
| app\_tblNZ\_AccPieInfo |
| app\_tblNZ\_AccPSO |
| app\_tblNZ\_AccountStd |
| app\_tblNZ\_GSTReg |
| ref\_lanAccountType |
| ref\_lanCloseReason |
| ref\_lancustomerlevel |
| ref\_lanFiling |
| ref\_lanListItem |
| ref\_lanNZ\_EmployerGroup |
| ref\_lanNZ\_EmployerType |
| ref\_lanNZ\_IncTaxExemptionRsn |
| ref\_lanprofiletype |
| ref\_lanindicator |

## Raw Layer tables from EDW (FIRST)

|  |
| --- |
| dss\_tax\_registrations |
| dss\_account\_halts |
| dss\_cm\_audit\_periods |
| dss\_clients |
| dss\_tax\_gst |
| dss\_tax\_pay |
| dss\_tax\_pie |
| dss\_tax\_cessation\_reasons |
| dss\_tax\_filing\_frequencies |
| dss\_tax\_types |
| dss\_tax\_exemption\_reasons |
| dss\_client\_names |
| dss\_customers |
| dss\_cm\_audit\_cases |
| dss\_cm\_audit\_periods |

# Mapping Tables Required for Processing EDW & START

Some mapping tables are required to be built to allow the correct assignment of “START-like” codes to legacy data.

## EDW-START Code Map

Because columns in EDW tables have different sets of codes and/or descriptions when compared to the equivalent columns in START we need a mapping table to allow processing to lookup the correct values.

Table mp\_source\_code (mp is an abbreviation for mapping) will follow a pattern 1 design and will only function as an insert table using pre-set string values input by the user. The table itself will only be populated once for party and used initially for only mapping START to EDW value precedence where appropriate for consistency of data across EDW and START. e.g. where a value in EDW is ‘I’, START may have a value of ‘INDVL’ which will overwrite I in the load to the party domain.

The EDW-Start code and description mappings are contained in table mp\_source\_code and documented in Appendix I:

1. Account (Tax) Types
2. Cessation Reasons
3. Filing Frequency Periods
4. Exemption Reasons
5. Accounting Basis
6. Registration Circumstance
7. PIE Filing Option
8. PIE Attribution Period
9. PIE FDR Calculation Period
10. PAY Filing Option

## Start Account Key – IRD Number Map

Mappings between start account keys and IR numbers are required to populate the start\_account\_key and dip\_account\_id fields in the Account table. This information is available in the rfn\_account\_sk\_map table which holds the generated dip account surrogate key for each version of the account changes for both Start and EDW.

Table rfn\_account\_sk\_map should contain at least the following columns and source from Start using the specified Step7 Transform Rule.

| **Column Name** | **Data Type** | **Keys** | **See Start Step 7 Transform Rule**  **(Target Column)** |
| --- | --- | --- | --- |
| dip\_account\_sk | bigint |  | dip\_account\_sk |
| dip\_account\_id | varchar(30) |  | dip\_account\_id |
| start\_account\_key | int | UK | account\_key |
| record\_effective\_timestamp | timestamp | UK | record\_effective\_timestamp |
| ird\_no | int |  | account\_ird\_number |
| edw\_customer\_type | varchar(12) |  | for START records derive from app\_tblCustomer.fstrCustomerType  where flngCustomerKey = app\_tblAccount.flngCustomerKey |
| edw\_location\_number | int |  | edw\_ location\_number |
| edw\_tax\_type | varchar(6) |  |  |
| source | varchar(6) |  | EDW or START  required because a small number of records in sk\_map will have the same record\_effective\_dates for both START and EDW e.g. start\_account\_key 1585895744 |

During the Account load for Start, the map table is populated with each unique version of start *account key* + *record\_effective\_timestamp* with an allocated DIP account surrogate key, the active associated ird\_number and transformed dip\_account\_id.

The following SQL used in the ETL process performs the following tasks:

1. Identify all unique versions of account changes from the timeline table for START.
2. Check if the start *account key* has been loaded into the key map table before.
3. If the start *account key* already exists in the key map, use the *account IRD number* from the key map for this version
4. If the start key doesn’t exist, look up the active *account IRD number* in the clean raw tblid table to find a valid *account IRD number* (see Step 7 Transform Rules for Start).

After the Start Account load is completed, the EDW Account load uses this table to find the start account key for its combinatons of tax type and IRD number and populates the unique version to table rfn\_account\_sk\_map. The following SQL used by the ETL process performs the following tasks:

1. Identify all the unique version of account changes from the all\_dates table for EDW
2. Check if the start *account key* has been loaded into the key map table
3. Look up the tax\_type + IRD\_Number + location\_number in the key map to find the associated start account key number.

| **Column Name**  **(rfn\_account\_sk\_map)** | **Lookup using** |
| --- | --- |
| ird\_no | dss\_tax\_registrations.ird\_number |
| start\_account\_type | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_code\_name = ‘TAX\_TYPE’  and source\_code\_value = dss\_tax\_registrations.tax\_type  and source\_cust\_type in (‘’, dss\_clients.cust\_type) if dip\_code\_value is returnedc.  then dip\_code\_value else tax\_type |
| heritage\_location\_number | dss\_tax\_registrations.location\_number |

If the process is not able to find a matched account key from the Start side, it sets the ignore indicator to 0 which is used to determine the application of the interleave operation for this EDW client.

# Transform Rules: EDW

## Step 1 – Selecting Valid Values from Source Tables

The purpose of this step is to reduce the number of records from the source object to those that are relevant to the processing. This may include only extracting a subset of rows as well as excluding records that are not deemed to be valid.

| **Source Table** | **Filtering Criteria** | **Transformation Rule / Logic** | **Change Capture Column** | **Business Active**  **Timestamps** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| dss\_clients | select cust\_type in ('ORG', 'IND')  and ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents |  | timestamp |  |  |
| dss\_special\_clients\_all | select ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers |  | timestamp |  |  |
| dss\_tax\_registrations | ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents | Derive ActiveFlag:  if treg\_date\_start  = treg\_date\_end  then ActiveFlag = 0  else ActiveFlag = 1 | timestamp | treg\_date\_start treg\_date\_end | See note 4 |
| dss\_tax\_gst | Select all records where ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents |  | timestamp | treg\_date\_start | See note 4 |
| dss\_tax\_pay | Select all records where ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents |  | timestamp | treg\_date\_start | See note 4 |
| dss\_tax\_pie | Select all records ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents |  | timestamp | treg\_date\_start | See note 4 |
| dss\_client\_names | Select all records where client\_name\_type in (‘P’, ’T’)  and ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents |  | date\_applied |  | See notes 3,11 |
| dss\_customers | select location\_number = 1  and ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents |  | timestamp |  |  |
| dss\_account\_halts | Select all records where ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents  and halt\_level\_indicator in ('C', 'L', 'T')  and date\_ceased = cast('9000-12-31 00:00:00' as timestamp) |  | timestamp | date\_halt\_start  date\_halt\_end | See notes 5-8 |
| dss\_cm\_audit\_cases | Select all records where ird\_number not equal to old\_system\_number within dss\_old\_system\_numbers  and ird\_number not equal to old\_system\_number within dss\_agents  and ird\_number is not null  and location\_number is not null  and date\_ceased = cast('9000-12-31 00:00:00' as timestamp) |  | timestamp | case\_opened\_date  case\_closed\_date |  |
| dss\_tax\_cessation\_reasons | Select all records where validated = ‘Y’ |  | timestamp |  | See note 1 |
| dss\_tax\_filing\_frequencies | Select all records where validated = ‘Y’ |  | timestamp |  | See note 1 |
| dss\_tax\_types | Select all records where validated = ‘Y’ |  | timestamp |  | See note 1 |
| dss\_tax\_exemption\_reasons | Select all records where validated = ‘Y’ |  | timestamp |  | See note 1 |
| dss\_tax\_reg\_status\_codes | Select all records where validated = ‘Y’ |  | timestamp |  | See note 1 |
| dss\_old\_system\_numbers | Select distinct old\_system\_number where old\_system\_number is not null |  |  |  | See note 2 |
| dss\_agents | Select distinct old\_system\_number where old\_system\_number is not null |  |  |  | See note 2 |

**Bridging Tables:**

These tables do not require de-duplicating and do not contribute to the account’s timeline.

| **Source Table** | **Filtering Criteria** | **Change Capture Column** | **Business Active**  **Timestamps** | **Comments** |
| --- | --- | --- | --- | --- |
| dss\_cm\_audit\_periods | Select all records where date\_ceased = cast('9000-12-31 00:00:00' as timestamp) | Timestamp | date\_allocated  date\_completed | See note 9 |

**Derived Tables:**

| **Derived Table** | **Source Table(s)** | **Derivation Rules** | **Change Capture Column** | **Comments** |
| --- | --- | --- | --- | --- |
| crw\_ae\_account\_halts | dss\_account\_halts  crw\_ae\_tax\_registrations (post clean raw processing) | **Derive table crw\_ae\_account\_halts:**  **LEFT JOIN** crw\_ae\_tax\_registrations (tr) todss\_account\_halts (ah)  on tr.ird\_number = ah.ird\_number  and tr.location\_number = (case when ah.halt\_level\_indicator = ‘C’  then 0 else ah.location\_number end)  and tr.tax\_type = (case when ah.halt\_level\_indicator in (‘C’,’L’)   then ‘XXX’ else ah.tax\_type end)  and ah.date\_halt\_start between tr.record\_effective\_timestamp and tr.record\_expiry\_timestamp  and ah.halt\_level\_indicator in ('C','L','T')  **Derive columns**:  ***derived\_location\_number*** from ifnull(tr.location\_number, -2)  ***derived\_tax\_type*** from ifnull(tr.tax\_type, -3)  ***derived\_level*** based on halt\_level\_indicator(C=3, L=2, T=1) |  | See Appendix E for sample SQL  See note 10 |
| crw\_ae\_cm\_audit\_cases | dss\_cm\_audit\_cases dss\_cm\_audit\_periods | **Derive table** crw\_ae\_cm\_audit\_cases:  **LEFT JOIN** dss\_cm\_audit\_periods (ap) to dss\_cm\_audit\_cases (ac)  on ap.case\_identifier = ac.case\_identifier  **Derive column(s):**  **derived\_tax\_type** from distinct ap.tax\_type |  | See Appendix E for sample SQL |

**Notes:**

|  |  |
| --- | --- |
| 1 | This is a lookup value table. |
| 2 | This is a lookup value table used to exclude old system numbers which are not to be processed. |
| 3 | For preferred names, location\_number=0. For trading names, location\_number=1. Unlike Start, there is no hierarchy for client names in EDW. |
| 4 | Each tax registration is effective between the Tax registration start and end dates |
| 5 | The account halts may be placed at one of 4 levels in a hierarchy i.e. 'C' Customer, 'T' Tax type, 'R' Return Period date and 'L' location (see HALT\_LEVEL\_INDICATOR). The scope of a customer-level halt includes all its locations. The scope of a location-level halt is all its tax types. The scope of a tax-type-level halt is only that tax type - all return-period-level account halts are to be ignored. |
| 6 | Though account halt records specify a taxable period (RETURN\_PERIOD\_DATE is the last date of the taxable period), this can be ignored since for 'C' Customer, 'T' Tax type, and 'L' location level account halts, the taxable period defaults to 31-DEC-2025. |
| 7 | The account halt period is between DATE\_HALT\_START and DATE\_HALT\_END. There may be successive ‘active’ account halt records for an account halt period – the last record may be ‘inactive’ (HALT\_STATUS\_INDICATOR = ‘I’) and all other records will be ‘active’ (HALT\_STATUS\_INDICATOR = ‘A’). Though successive records for an account halt period may change the value of the DATE\_HALT\_END, the value of DATE\_HALT\_START will not change. |
| 8 | Because there is only the initial load for EDW accounts, only the current record for each account halt period is required. Normally this record is ‘inactive’ (HALT\_STATUS\_INDICATOR = ‘I’) though the current record may be ‘active’ (HALT\_STATUS\_INDICATOR = ‘A’). The current record is identifiable by a date\_ceased of 31-DEC-9000. When the current record is ‘active’, the DATE\_HALT\_END is null. |
| 9 | This is a bridging table required to restrict dss\_cm\_audit\_cases to only those tax (account) types which are being audited. This may be a subset of those tax (account) types related to the specified ird\_number and location\_number. These records do not require de-duplicating and do not contribute to the account’s timeline. |
| 10 | Account halts may be placed at one of 4 levels in a hierarchy i.e. 'C' Customer, 'T' Tax type, 'R' Return Period date and 'L' location (see HALT\_LEVEL\_INDICATOR) - all return-period-level account halts are to be ignored. This hierarchy must be ‘unpacked’ prior to Step#2 de-duplication to ensure that duplicates across hierarchy levels are correctly dealt with. |
| 11 | Clean raw table has aleady been built for party with the same criteria so will be re-used rather than build a separate accounts clean raw table |

## Step 2 – De-duplication of Records

The purpose of this step is to remove duplicates over time so that the resulting rows for each natural key are contiguous and non-overlapping in time. The default de-duplication process (see [Appendix D](#_Appendix_D_–)) is to be following unless explicitly stated otherwise.

NB. The derived field “number\_of\_valid\_days” used to rank ‘duplicate’ records is to be calculated as the number of days that the business active timestamps ‘overlap’ the system active timestamps. See [Appendix D](#_Appendix_D_–) for details.

| **Source Table** | **De-duplication Rule** | **Natural Key** | **Business Active**  **Timestamps** | **Tie-Breaker(s)** |
| --- | --- | --- | --- | --- |
| dss\_clients |  | ird\_number |  |  |
| dss\_special\_clients\_all |  | ird\_number |  |  |
| dss\_tax\_registrations | Use the default de-duplication process (see Appendix D) | ird\_number  location\_number  tax\_type | treg\_date\_start treg\_date\_end | Rank:   1. ActiveFlag (1 before 0) 2. ~~record\_expiry\_timestamp (latest first)~~ 3. Number\_of\_valid\_days (highest first) 4. treg\_date\_end (latest first) 5. treg\_date\_start (earliest first first) |
| dss\_tax\_gst | Use the default de-duplication process (see Appendix D) | ird\_number  location\_number  tax\_type | treg\_date\_start | Rank:   1. record\_expiry\_timestamp (latest first) 2. treg\_date\_start (earliest first first) |
| dss\_tax\_pay | Use the default de-duplication process (see Appendix D) | ird\_number  location\_number  tax\_type | treg\_date\_start | Rank:   1. record\_expiry\_timestamp (latest first) 2. treg\_date\_start (earliest first first) |
| dss\_tax\_pie | Use the default de-duplication process (see Appendix D) | ird\_number  location\_number  tax\_type | treg\_date\_start | Rank:   1. record\_expiry\_timestamp (latest first) 2. treg\_date\_start (earliest first first) |
| dss\_client\_names | Use the default de-duplication process (see Appendix D) | ird\_number  location\_number  client\_name\_type |  | Rank:   1. record\_expiry\_timetamp (latest first) 2. date\_applied (latest first) 3. sequence\_number (highest first – final tie-breaker if needed) |
| dss\_customers |  | ird\_number  location\_number |  |  |
| dss\_account\_halts | Use the default de-duplication process (see Appendix D) | ird\_number  ***derived\_location\_number***  ***derived\_tax\_type*** | date\_halt\_start  date\_halt\_end | Rank:   1. Number\_of\_valid\_days (highest first) 2. derived\_level (asc) –priortise across hierarchy 3. date\_applied (latest first) 4. ~~record\_expiry\_timetamp (latest first)~~ 5. ~~record\_effective\_timetamp (latest first)~~ |
| dss\_cm\_audit\_cases | Use the default de-duplication process (see Appendix D) | ird\_number  location\_number  ***derived\_tax\_type*** | case\_opened\_date  case\_closed\_date | Rank:   1. Number\_of\_valid\_days (highest first) 2. date\_applied (latest first) 3. ~~record\_expiry\_timetamp (latest first)~~ 4. ~~record\_effective\_timetamp (latest first)~~ |
| dss\_tax\_cessation\_reasons |  | cessation\_reason\_code |  |  |
| dss\_tax\_filing\_frequencies |  | filing\_frequency |  |  |
| dss\_tax\_types |  | tax\_type |  |  |
| dss\_tax\_exemption\_reasons |  | exemption\_reason\_code |  |  |
| dss\_tax\_reg\_status\_codes |  | treg\_status\_code |  |  |
| dss\_old\_system\_numbers |  | old\_system\_number |  |  |
| dss\_agents |  | old\_system\_number |  |  |

## Step 3 – Compressing the timeline

After de-duplicating records in Step 2, one or more contiguous records sharing the same Natural Key may have the same values for all columns used for comparison. This step will ‘compress’ (i.e. merge) any such contiguous records using the minimum Record\_Effective\_Date and maximum Record\_Expiry\_Date for the resulting timeline.

NB. Business active timestamps/dates should always be included in the “columns to compare”.

| **Source Table** | **Natural Key** | **Business Active**  **Timestamps** | **Columns to Compare** | **Dates for Timelines** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| dss\_clients | ird\_number |  | balance\_date  cust\_type | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_special\_clients\_all | ird\_number |  | special\_classification |  |  |
| dss\_tax\_registrations | ird\_number  location\_number  tax\_type | treg\_date\_start treg\_date\_end | treg\_status  cessation\_reason\_code  filing\_frequency  exemption\_reason\_code  treg\_date\_start treg\_date\_end | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_gst | ird\_number  location\_number  tax\_type | treg\_date\_start | accounting\_basis\_code  nat\_of\_rgn  exempt\_supp\_ind  exp\_ind  self\_inv\_ind  treg\_date\_start | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_pay | ird\_number  location\_number  tax\_type | treg\_date\_start | filing\_frequency  paye\_employer\_class\_code treg\_date\_start | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_pie | ird\_number  location\_number  tax\_type | treg\_date\_start | pie\_type  pie\_report\_option  income\_allocation\_period  treg\_date\_start | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_client\_names | ird\_number  location\_number  client\_name\_type |  | title first\_names  surname  organisation\_name | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_customers | ird\_number  location\_number |  | entity\_class | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_account\_halts | ird\_number  ***derived\_location\_number***  ***derived\_tax\_type***  date\_halt\_start | date\_halt\_start  date\_halt\_end | halt\_status\_indicator  ~~date\_halt\_start~~  date\_halt\_end | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_cm\_audit\_cases | ird\_number  location\_number  ***derived\_tax\_type*** | case\_opened\_date  case\_closed\_date | case\_open\_reason\_desc case\_opened\_date  case\_closed\_date | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_cessation\_reasons | cessation\_reason\_code |  | description | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_filing\_frequencies | filing\_frequency |  | description | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_types | tax\_type |  | description | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_exemption\_reasons | exemption\_reason\_code |  | description | record\_effective\_timestamp record\_expiry\_timestamp |  |
| dss\_tax\_reg\_status\_codes | treg\_status\_code |  | description | record\_effective\_timestamp record\_expiry\_timestamp |  |

## Step 4 – Combining all change timeline

The purpose of this step is to create a UNION of all change timestamps for the Account data that will become the basis for all change records created. These timelines are stored in the Refined Layer in table rfn\_ae\_all\_dates. Each record in the timeline is bound in time from record\_effective\_timestamp to record\_expiry\_timestamp.

A change timestamp is generated for each Record\_Effective\_Timestamp at a 1-minute granularity, additionally where a Business Start Date and/or Business End date are listed below, a change timestamp should be created for each at 1-minute granularity. Where a Business\_Start\_Date contains sub-minute data the time should be **rounded up** to the next minute. A Business\_End\_Date should be rounded up to the next unit of time. (If an end date preseted at the day grain is shown as ’01-10-1989’ rounding up to the next unit of time means creating a change timestamp of ’02-01-1989’. This is done as we need to create a change timestamp immediately after the business end date, so that we show a Party record ***not*** having the “ceased” value.)

When there are 2 or more change timestamps each 1 minute apart, select the only the latest one. This is to prevent issues with the setting of Record\_Expiry\_Timestamp at a 1-minute grain and effectively takes the final “accumulated” record when multiple changes come through at the same time (or within a minute of each other).

e.g. A set of records having change timestamps of:

01-Mar-2019 10:52 A B NULL NULL NULL  
01-Mar-2019 10:53 A B X NULL NULL  
01-Mar-2019 10:54 A B X Y NULL  
01-Mar-2019 10:55 A B X Y Z

will be treated as:

01-Mar-2019 10:55 A B X Y Z

This may result in additional change records being created, but they will be compressed in the final step of the Refined Layer build for each target table.

| **Raw Table** | **Clean Raw Table** | **Business Start date** | **Business End date** | **Comments** |
| --- | --- | --- | --- | --- |
| dss\_clients | crw\_ae\_clients |  |  |  |
| dss\_special\_clients\_all | crw\_pe\_special\_clients\_all |  |  | This table is reused from DIP Party |
| dss\_tax\_registrations | crw\_ae\_tax\_registrations | treg\_date\_start | if treg\_date\_end is not = $$HDT  then days\_add(treg\_date\_end,1) |  |
| dss\_tax\_gst | crw\_ae\_tax\_gst | treg\_date\_start |  |  |
| dss\_tax\_pay | crw\_ae\_tax\_pay | treg\_date\_start |  |  |
| dss\_tax\_pie | crw\_ae\_tax\_pie | treg\_date\_start |  |  |
| dss\_client\_names | crw\_ae\_client\_names |  |  |  |
| dss\_customers | crw\_pe\_customers |  |  | This table is reused from DIP Party |
| dss\_account\_halts | crw\_ae\_account\_halts |  |  |  |
| dss\_cm\_audit\_cases | crw\_ae\_cm\_audit\_cases | case\_opened\_date | if case\_closed\_date is not = $$HDT  then days\_add(case\_closed\_date,1) |  |

## Step 5 – Create a single timeline for each Account

Create a single timeline for each Account. The account timeline is stored the Refined Layer in table rfn\_ae\_timeline. Each record in the timeline is bound in time from record\_effective\_timestamp to record\_expiry\_timestamp.

| **Condition** | **Cut-off Boundaries** | **Rule Logic** | **Comments** |
| --- | --- | --- | --- |
| EDW Account has been migrated to Start | Lower cut-off date | Use the earliest EDW Account start date | The account start date is sourced from dss\_tax\_registrations.treg\_date\_start |
| Upper cut-off date | ~~Interleave date - 1 minute~~  If ***R1 account type***  then $$EDWDT1 – 1 minute else if ***R2 account type***  then $$EDWDT2 – 1 minute else if ***R3 account type***  then $$EDWDT3 – 1 minute | ***R1 account types***: ‘GST’  ***R2 account types***:  ‘AIL’, ,CSE’, ‘DWT’, ‘FBA’, ‘FBI’, ‘FBT’, ‘GMD’, ‘IPE’, ‘IPS’, ‘NRT’, ‘PAY’, ‘PIE’, ‘RLT’, ‘RWT’, ‘SLE’, ‘SSC’, ‘WPE’, ‘WPN’  ***R3 account types***:  ‘CPR’, ‘FAM’, ‘ICA’, ‘INC’, ‘KSS’, ‘MAC’, ‘NCP’, ‘PPL’, ‘QCT’, ‘REB’, ‘SLS’ |
| EDW Account has not been migrated to Start | Lower cut-off date | Use the earliest EDW Account start date | The account start date is sourced from dss\_tax\_registrations.treg\_date\_start |
| Upper cut-off date | $$HIDATE | No cut-off is required for the upper boundary |

## Step 6a – Merge change timelines with Account timeline

Merge the change timelines from Step#4 with the single Account timeline from Step#5. Each record in the timeline has a natural key of ird\_number, location\_number, tax\_type, record\_effective\_timestamp and record\_expiry\_timestamp.

NB. When the offset for the business active timestamps (or dates) is greater than one minute, one day must be added to the ‘end’ timestamp when comparing to the timeline.record\_effective\_date.

| **Clean Raw Table** | **Filtering Criteria** | **Business Active**  **Timestamps** | **Time-stamp Offset** | **Comments** |
| --- | --- | --- | --- | --- |
| crw\_ae\_tax\_registrations | ***timeline.ird\_no*** = ird\_number  and ***timeline.location\_number*** = location\_number  and ***timeline.tax\_type*** = tax\_type  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_effective\_timestamp*** >= treg\_date\_start | treg\_date\_start treg\_date\_end | Day  Day | See Note 1 |
| crw\_ae\_tax\_gst | ***timeline.ird\_ no*** = ird\_number  and ***timeline.location\_number*** = location\_number  and ***timeline.tax\_type*** = tax\_type  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_effective\_timestamp*** >= treg\_date\_start | treg\_date\_start | Day | See Note 1 |
| crw\_ae\_tax\_pay | ***timeline.ird\_ no*** = ird\_number  and ***timeline.location\_number*** = location\_number  and ***timeline.tax\_type*** = tax\_type  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_effective\_timestamp*** >= treg\_date\_start | treg\_date\_start | Day | See Note 1 |
| crw\_ae\_tax\_pie | ***timeline.ird\_ no*** = ird\_number  and ***timeline.location\_number*** = location\_number  and ***timeline.tax\_type*** = tax\_type  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_effective\_timestamp*** >= treg\_date\_start | treg\_date\_start | Day | See Note 1 |
| crw\_pe\_client\_names | ***timeline.ird\_ no*** = ird\_number  and ***timeline.location\_number*** = location\_number  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp |  |  |  |
| crw\_ae\_account\_halts | ***timeline.ird\_ no*** = ird\_number  and ***timeline.location\_number*** = derived\_location\_number  and ***timeline.tax\_type*** = derived\_tax\_type  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_effective\_timestamp*** >= date\_halt\_start  and days\_add(***timeline.record\_effective\_timestamp***,-1) <= date\_halt\_end | date\_halt\_start  date\_halt\_end | Day  Day | See Note 1 |
| crw\_ae\_cm\_audit\_cases | ***timeline.ird\_ no*** = ird\_number  and ***timeline.location\_number*** = location\_number  and ***timeline.tax\_type*** = derived\_tax\_type  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_efective\_timestamp*** >= case\_opened\_date  and days\_add(***timeline.record\_effective\_timestamp***,-1) <= case\_closed\_date | case\_opened\_date  case\_closed\_date | Day  Day | See Note 1 |

**Notes:**

|  |  |
| --- | --- |
| 1 | When the ‘offset’ for the business active timestamps (or dates) is greater than one minute, one unit of the ‘offset’ value must be subtracted from the timeline.record\_effective\_date when compared to the record’s ‘end’ timestamp. |

## Step 6b – Compress the Post-Merge Timelines

After timelines are merged in Step#6a, one or more contiguous records sharing the same Natural Key may have the same values for all columns used for comparison. This step will ‘compress’ (i.e. merge) any such contiguous records using the minimum Record\_Effective\_Date and maximum Record\_Expiry\_Date for the resulting timeline.

See Step#6b for Start.

## Step 7 – Transform Rule Logic

### Example SQL for Context

Appendix E contains the example SQL statements for Start Accounts. The “Select” column names in these SQL statements will be refered to in the “Source Table/Column” fields of the transformation rules below and are intended to give context to those rules as well as a guide on how to merge all the source objects into the final Account Type data streams.

### Overarching Rules/logic

1. Account Type records are to be versioned to share the same keys. Therefore, any change to the versioning for an Account will result in an identical change to the versioning for either the corresponding Account Type records.
2. Required date validation functions are:

| **Function Name** | **Purpose** | **Logic** |
| --- | --- | --- |
| $$is\_date\_valid$$ | Determine if the specified date (parameter: $in\_date$) is valid | FUNCTION $$check\_date\_valid$$ (in $in\_date$) return boolean  If $in\_date$ is not between ‘1880-JAN-01’ and $today$  then return FALSE  else if $in\_date$ = ‘1900-JAN-01’  then return FALSE else return TRUE |

1. Relevant Account types are:

|  |  |  |
| --- | --- | --- |
| **DIP Account Type Table** | **Account Type** | **Description** |
| ACCOUNT | All | All Account types |
| PARTY\_ACCOUNT | All | All Account types |
| ACCOUNT\_PAY | PSO | Employment Activities (Payroll) |
| ACCOUNT\_PIE | PIE | Portfolio Investment Entity |
| ACCOUNT\_INC | IIT | Income Tax Individuals |
| ITN | Income Tax Non-Individuals |
| ACCOUNT\_GST | GST | Goods and Services Tax |

### EDW Party Account

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| dss\_tax\_registrations.ird\_number | lookup rfn\_party\_sk\_map on ird\_number  if dip\_party\_id is returned  then dip\_party\_id  else concatenate (‘DIP-I-’, cast (ird\_number as varchar)) | dip\_party\_id | PK |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_id | PK |  |
|  | *Standard processing field* | record\_effective\_timestamp | PK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
|  | ‘Owned’ | relationship\_type |  |  |
| dss\_tax\_registrations.treg\_date\_start | Copy | relationship\_effective\_from |  |  |
| dss\_tax\_registrations.treg\_date\_end | Copy | relationship\_effective\_to |  | ‘9999-12-31‘ |
|  | ‘FIRST’ | relationship\_origin |  |  |
|  | ‘HIGH’ | dip\_confidence\_level |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### EDW Account

No Filter i.e. populate for all Account Types.

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
|  | Increment by 1 for each record. This key is pervasive across the Account table and all Account Type tables. | dip\_account\_sk | SK |  |
| dss\_tax\_registrations.ird\_number  dss\_tax\_registrations.location\_number  dss\_tax\_registrations.tax\_type  rfn\_account\_sk\_map.dip\_account\_id (see Section 5.2) | lookup rfn\_account\_sk\_map   on ird\_number, location\_number, tax\_type  if dip\_account\_id is returned  then dip\_account\_id  else CONCAT('DIP-', LPAD(CAST(ird\_number AS VARCHAR(20)), 9, '0'), '-', tr.tax\_type, LPAD(CAST(location\_number AS VARCHAR(3)), 3, '0'))) | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
|  | ‘FIRST’ | account\_origin |  |  |
| dss\_tax\_registrations.ird\_number  dss\_tax\_registrations.location\_number  dss\_tax\_registrations.tax\_type  rfn\_account\_sk\_map.dip\_account\_id (see Section 5.2) | lookup rfn\_account\_sk\_map   on ird\_number, location\_number and tax\_type  if start\_account\_key is returned  then start\_account\_key  else ‘Unknown’ | account\_key |  | Unknown |
| dss\_tax\_registrations.ird\_number | Copy | account\_ird\_number |  | N/A |
|  | ‘N/A’ | start\_account\_id |  | N/A |
|  | ‘N/A’ | start\_profile\_number |  | N/A |
|  | ‘N/A’ | start\_parent\_profile\_type |  | N/A |
|  | ‘N/A’ | start\_parent\_profile\_number |  | N/A |
| dss\_tax\_registrations.location\_number | Copy | heritage\_location\_number |  | Unknown |
| dss\_client\_names.title  dss\_client\_names.first\_names  dss\_client\_names.surname  dss\_client\_names.organisation\_name  dss\_client\_names.client\_name\_type  dss\_client\_names.location\_number  dss\_clients.cust\_type | if cust\_type = ‘ORG’  and client\_name\_type = ‘P’  and location\_number = 0  then organisation\_name  else if cust\_type = ‘IND  and client\_name\_type = ‘P’  and location\_number = 0  then concatenate(title,’ ‘,first\_names, ‘ ’, surname)  *Eliminating any spurious spaces caused by one of the fields being NULL*  else        ‘Unknown’ | legal\_name |  | Unknown |
| dss\_client\_names.organisation\_name  dss\_client\_names.client\_name\_type  dss\_client\_names.location\_number | if client\_name\_type = ‘T’  and location\_number = 1  and organisation\_name is not null  then organisation\_name  else same as *Target Column legal\_name*  NB. *Trading name for individuals is stored in organisation\_name* | trading\_name |  | Unknown |
| dss\_tax\_types.description  dss\_tax\_registrations.tax\_type  mp\_source\_code.dip\_description (see Section 5.1)  dss\_clients.cust\_type | lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_code\_name = ‘TAX\_TYPE’  and source\_code\_value = tax\_type  and source\_cust\_type in (‘’, dss\_clients.cust\_type)  if dip\_description is returned  then dip\_description else  lookup dss\_tax\_types on tax\_type  if description is returned  then description | account\_type |  | Unknown |
| dss\_tax\_registrations.tax\_type  mp\_source\_code.dip\_description (see Section 5.1)  dss\_clients.cust\_type | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_code\_name = ‘TAX\_TYPE’  and source\_code\_value = tax\_type  and source\_cust\_type in (‘’, dss\_clients.cust\_type) if dip\_code\_value is returned  then dip\_code\_value else tax\_type | account\_type\_code |  | Unknown |
| dss\_tax\_registrations.treg\_date\_start | if $$is\_date\_valid$$(treg\_date\_start)  then treg\_date\_start | account\_start\_date |  | $$LDT |
| dss\_tax\_registrations.treg\_date\_end | if $$is\_date\_valid$$(treg\_date\_end)  then treg\_date\_end | account\_ceased\_date |  | $$HDT |
| dss\_tax\_reg\_status\_codes.description  dss\_tax\_registrations.treg\_status | if treg\_status = ‘A’  then ‘Active’ else if treg\_status = ‘C’  then ‘Closed’ else Lookup dss\_tax\_reg\_status\_codes  on treg\_status\_code = treg\_status if description is returned  then description | account\_status |  | Unknown |
| dss\_tax\_registrations.treg\_status | if treg\_status = ‘A’  then ‘ACT’ else if treg\_status = ‘C’  then ‘CLS’ else Copy | account\_status\_code |  | Unknown |
| ~~dss\_special\_clients\_all.special\_classification~~  ref\_lancustomerlevel.fstrCustomerLevel  ref\_lancustomerlevel.fstrDecode2  *target column: security\_level\_code* | ~~if special\_classification = ‘STD’  then ‘Unrestricted Standard Customer’ else if special\_classification = ‘SPC’  then ‘Special File’~~  lookup ref\_lancustomerlevel   on fstrCustomerLevel = *security\_level\_code* if fstrDecode2 is returned then  update all records for this DIP\_ACCOUNT\_ID  Set to fstrDecode2 **over all time** | security\_level |  | Unknown |
| dss\_special\_clients\_all.special\_classification  dss\_tax\_registrations.ird\_number  rfn\_party\_sk\_map.ird\_no  rfn\_party\_sk\_map.start\_customer\_key  app\_tblCustomer.flngcustomerkey  app\_tblCustomer.flngDocKey  app\_tblcustomerlevel.fstrCustomerLevel | ~~Copy~~  Lookup rfn\_party\_sk\_map on ird\_no = ird\_number  if start\_customer\_key is returned then  Lookup app\_tblcustomer   on flngcustomerkey = start\_customer\_key if flngDocKey is returned then  Lookup app\_tblcustomerlevel on flngDocKey  if fstrCustomerLevel is returned then  update all records for this DIP\_ACCOUNT\_ID   Set to fstrCustomerLevel **over all time**  else if special\_classification is populated then  update all records for this DIP\_ACCOUNT\_ID   set to special\_classification **over all time**  else if special\_classification is populated then  update all records for this DIP\_ACCOUNT\_ID   set to special\_classification **over all time**  else if special\_classification is populated then  update all records for this DIP\_ACCOUNT\_ID   set to special\_classification **over all time**  ***NB. In the START load this field will be updated over all time with the latest value of Customer Level (becomes a SCD1 type operation over all time)*** | security\_level\_code |  | N/A |
| dss\_tax\_registrations.cessation\_reason\_code dss\_tax\_cessation\_reasons.description  dss\_tax\_registrations.tax\_type  mp\_source\_code.dip\_description (see Section 5.1) | if tax\_registrations.treg\_status = ‘C’  then Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘CESSATION\_REASON’  and source\_code\_value = cessation\_reason\_code  and source\_tax\_type = tax\_type   if dip\_description is returned  then dip\_description  else Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘CESSATION\_REASON’  and source\_code\_value = cessation\_reason\_code  and source\_tax\_type = ‘’  if dip\_description is returned  then dip\_description  else   lookup dss\_tax\_cessation\_reasons  on cessation\_reason\_code  if description is returned  then description  else ‘Unknown’  else ‘N/A’ | cessation\_reason |  | Unknown |
| dss\_tax\_registrations.cessation\_reason\_code  dss\_tax\_registrations.tax\_type  mp\_source\_code. dip\_code\_value (see Section 5.1) | if tax\_registrations.treg\_status = ‘C’   then Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘CESSATION\_REASON’  and source\_code\_value = cessation\_reason\_code  and source\_tax\_type = tax\_type   if dip\_code\_value is returned  then dip\_code\_value  else Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘CESSATION\_REASON’  and source\_code\_value = cessation\_reason\_code  and source\_tax\_type = ‘’  if dip\_code\_value is returned  then dip\_code\_value  else cessation\_reason\_code else ‘N/A’ | cessation\_reason\_code |  | Unknown |
| dss\_tax\_filing\_frequencies.description  mp\_source\_code.dip\_description (see Section 5.1)  dss\_tax\_registrations.filing\_frequency | Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘FILING\_FREQUENCY’  and source\_code\_value = filing\_frequency  and dip\_code\_name = ‘FILING\_FREQ\_PERIOD’ if dip\_description is returned  then initcap(dip\_description) else lookup dss\_tax\_filing\_frequencies on filing\_frequency  if description is returned  then initcap(description)  else ‘Unknown’  else ‘N/A’ | filing\_frequency |  | Unknown |
| dss\_tax\_registrations.filing\_frequency | Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘FILING\_FREQUENCY’  and source\_code\_value = filing\_frequency  and dip\_code\_name = ‘FILING\_FREQUENCY’ if dip\_code\_value is returned  then dip\_code\_value else filing\_frequency | filing\_frequency\_code |  | Unknown |
| dss\_tax\_exemption\_reasons.description  mp\_source\_code.dip\_description (see Section 5.1)  dss\_tax\_registrations.exemption\_reason\_code  dss\_tax\_registrations.tax\_type | Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘EXEMPTION\_REASON’  and source\_code\_value = exemption\_reason\_code  and source\_tax\_type = tax\_type if dip\_description is returned  then dip\_description else lookup dss\_tax\_exemption\_reasons   on exemption\_reason\_code  if description is returned  then description  else ‘Unknown’ | exemption\_reason |  | Unknown |
| dss\_tax\_registrations.exemption\_reason\_code  dss\_tax\_registrations.tax\_type  mp\_source\_code.dip\_code\_value (see Section 5.1) | Lookup mp\_source\_code   for source\_system = ‘FIRST’  and source\_code\_name = ‘EXEMPTION\_REASON’  and source\_code\_value = exemption\_reason\_code  and source\_tax\_type = tax\_type if dip\_code\_value is returned  then dip\_code\_value else exemption\_reason\_code | exemption\_reason\_code |  | Unknown |
| dss\_tax\_registrations.exemption\_reason\_code | if dss\_tax\_registrations.exemption\_reason\_code is populated   then ‘Y’  else ‘N’ | exempted\_flag |  |  |
|  | ‘N’ | green\_listed\_flag |  |  |
|  | ‘N’ | red\_listed\_flag |  |  |
| dss\_account\_halts.halt\_status\_indicator  check for desc | if halt\_status\_indicator is populated  then ‘Y’ else ‘N’ | account\_halt\_flag |  |  |
| dss\_cm\_audit\_cases.case\_open\_reason\_desc | Copy | audit\_case\_active\_desc |  | Unknown |
|  | ‘Unknown’ | audit\_case\_active\_code |  | Unknown |
| dss\_cm\_audit\_cases.derived\_tax\_type | if derived\_tax\_type is populated  then ‘Y’ else ‘N’ | audit\_case\_active\_flag |  |  |
|  | ‘Unknown’ | stop\_mail\_desc |  | Unknown |
|  | ‘Unknown’ | stop\_mail\_code |  | Unknown |
|  | ‘N’ | stop\_mail\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### EDW Account PAY

Populate when tax\_type = ‘PAY’

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
|  | Increment by 1 for each record. This key is pervasive across the Account table and all Account Type tables. | dip\_account\_pay\_sk | SK |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_origin |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_key |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ird\_number |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_account\_id |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | heritage\_location\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | legal\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | trading\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | security\_level |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | security\_level\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exempted\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | green\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | red\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_halt\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_desc |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_code |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_flag |  |  |
| dss\_tax\_pay.paye\_employer\_class\_code | if paye\_employer\_class\_code = ‘I’  then ‘Y’ else ‘N’ | ir56\_flag |  |  |
| dss\_tax\_pay.filing\_frequency  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PAY’  and source\_code\_name = ‘FILING\_FREQUENCY’  and dip\_code\_name = ‘FILING\_OPTION’  and source\_code\_value = filing\_frequency if dip\_description is returned  then dip\_description else ‘Unknown’ | filing\_option |  | Unknown |
| dss\_tax\_pay.filing\_frequency  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PAY’  and source\_code\_name = ‘FILING\_FREQUENCY’  and dip\_code\_name = ‘FILING\_OPTION’  and source\_code\_value = filing\_frequency if dip\_code\_value is returned  then dip\_code\_value else ‘Unknown’ | filing\_option\_code |  | Unknown |
|  | ‘N’ | payday\_filer\_flag |  |  |
|  | $$HDT | payday\_filer\_start\_date |  |  |
| dss\_tax\_registrations.ird\_number [TR]  dss\_tax\_registrations.location\_number [TR]  dss\_tax\_registrations\_cse.ird\_number [CSE]  dss\_tax\_registrations\_cse.location\_number [CSE] | Lookup dss\_tax\_registrations\_cse   on [TR].ird\_number, [TR].location\_number  if [CSE] record is found  then ‘Y’ else ‘N’  NB. *Check is for a CPR, CSE or NCP TaxType for the same IRD# and location* | employer\_child\_support\_flag |  |  |
| dss\_tax\_registrations.ird\_number [TR]  dss\_tax\_registrations.location\_number [TR]  dss\_tax\_registrations\_sle.ird\_number [SLE]  dss\_tax\_registrations\_sle.location\_number [SLE] | Lookup dss\_tax\_registrations\_sle   on [TR].ird\_number, [TR].location\_number  if [SLE] record is found  then ‘Y’ else ‘N’  NB. *Check is for a SLE TaxType for the same IRD# and location* | employer\_student\_loan\_flag |  |  |
| dss\_tax\_registrations.ird\_number [TR]  dss\_tax\_registrations.location\_number [TR]  dss\_tax\_registrations\_kse.ird\_number [KSE]  dss\_tax\_registrations\_kse.location\_number [KSE] | Lookup dss\_tax\_registrations\_kse   on [TR].ird\_number, [TR].location\_number  if [KSE] record is found  then ‘Y’ else ‘N’  NB. *Check is for a KSE TaxType for the same IRD# and location* | kiwisaver\_employee\_dedn\_flag |  |  |
| dss\_tax\_registrations.ird\_number [TR]  dss\_tax\_registrations.location\_number [TR]  dss\_tax\_registrations\_ksr.ird\_number [KSR]  dss\_tax\_registrations\_ksr.location\_number [KSR] | Lookup dss\_tax\_registrations\_ksr   on [TR].ird\_number, [TR].location\_number  if [KSR] record is found  then ‘Y’ else ‘N’  NB. *Check is for a KSR TaxType for the same IRD# and location* | kiwisaver\_employer\_cntrb\_flag |  |  |
| dss\_tax\_registrations.ird\_number [TR]  dss\_tax\_registrations.location\_number [TR]  dss\_tax\_registrations\_ssc.ird\_number [SSC]  dss\_tax\_registrations\_ssc.location\_number [SSC] | Lookup dss\_tax\_registrations\_ssc   on [TR].ird\_number, [TR].location\_number  if [SSC] record is found  then ‘Y’ else ‘N’  NB. *Check is for a SSC TaxType for the same IRD# and location* | ESCT\_flag |  |  |
|  | ‘Unknown’ | employer\_group |  | Unknown |
|  | ‘Unknown’ | employer\_group\_code |  | Unknown |
|  | ‘N’ | payday\_variation\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### EDW Account PIE

Populate when tax\_type = ‘PIE’.

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_pie\_sk | SK |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_origin |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_key |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ird\_number |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_account\_id |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | heritage\_location\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | legal\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | trading\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | security\_level |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | security\_level\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exempted\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | green\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | red\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_halt\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_flag |  |  |
| dss\_tax\_pie.pie\_type  ref\_lanlistitem.fstrdecode1 | if pie\_type = ‘DB’  then ‘Defined Benefit’  else lookup ref\_lanlistitem on fstrfield = 'NZ.ACCPIE.PIEType'  and fstrlistitem = pie\_type if fstrdecode1 is returned  then fstrdecode1  else  ‘Unknown’ | pie\_type |  | Unknown |
| dss\_tax\_pie.pie\_type | Copy | pie\_type\_code |  | Unknown |
| dss\_tax\_pie.pie\_report\_option  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PIE’  and source\_code\_name = ‘PIE\_REPORT\_OPTION’  and source\_code\_value = pie\_report\_option if dip\_description is returned  then dip\_description else ‘Unknown’ | filing\_option |  | Unknown |
| dss\_tax\_pie.pie\_report\_option  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PIE’  and source\_code\_name = ‘PIE\_REPORT\_OPTION’  and source\_code\_value = pie\_report\_option if dip\_code\_value is returned  then dip\_code\_value else ‘Unknown’ | filing\_option\_code |  | Unknown |
| dss\_tax\_pie.income\_allocation\_period  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PIE’  and source\_code\_name = ‘INCOME\_ALLOCATION\_PERIOD’  and source\_code\_value = income\_allocation\_period if dip\_description is returned  then dip\_description else ‘Unknown’ | attribution\_period |  | Unknown |
| dss\_tax\_pie.fdr\_calculation\_period  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PIE’  and source\_code\_name = ‘FDR\_CALCULATION\_PERIOD’  and source\_code\_value = fdr\_calculation\_period if dip\_description is returned  then dip\_description else ‘Unknown’ | fdr\_calc\_period |  | Unknown |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### EDW Account INC

Populate when tax\_type = ‘INC’

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_inc\_sk | SK |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_origin |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_key |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ird\_number |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_account\_id |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | heritage\_location\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | legal\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | trading\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | security\_level |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | security\_level\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exempted\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | green\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | red\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_halt\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_desc |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_code |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_flag |  |  |
| dss\_clients.balance\_date | If balance\_date is null  then ‘Unknown’ else if balance\_date = 0  then ‘Unknown’  else  CAST(CONCAT(MONTHNAME(to\_timestamp(concat('2000',lpad(cast(balance\_date as varchar(4)),4,'0')), 'yyyyMMdd')), ' ',CAST(DAY(to\_timestamp(concat('2000',lpad(cast(balance\_date as varchar(4)),4,'0')), 'yyyyMMdd')) AS VARCHAR(2))) AS VARCHAR(30)))  N.B example of ‘331’ is formatted to “March 31” | balance\_date |  | Unknown |
|  | ‘N’ | zero\_tailored\_rate\_flag |  |  |
|  | ‘N’ | trans\_tasman\_flag |  |  |
| dss\_customers.entity\_class | if entity\_class = ‘AI’  then ‘Y’ else ‘N’ | agent\_non\_res\_insurer\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### EDW Account GST

Populate when tax\_type = ‘GST’.

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_gst\_sk | SK |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_origin |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_key |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ird\_number |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_account\_id |  | N/A |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | heritage\_location\_number |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | legal\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | trading\_name |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exemption\_reason\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | exempted\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | green\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | red\_listed\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | account\_halt\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_desc |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_code |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See EDW ACCOUNT for details* | *See EDW ACCOUNT for details* | stop\_mail\_flag |  |  |
| dss\_tax\_gst.accounting\_basis\_code  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_code\_name = ‘ACCOUNTING\_BASIS\_CODE’  and source\_code\_value = accounting\_basis\_code if dip\_description is returned   then dip\_description else ‘Unknown’  NB. A value of ‘X’ will result in ‘Unknown’ | accounting\_basis |  | Unknown |
| dss\_tax\_gst.nat\_of\_rgn  mp\_source\_code.dip\_description (see Section 5.1) | if nat\_of\_rgn = ‘S’  then ‘Special’  else Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_code\_name = ‘NAT\_OF\_REGN’  and source\_code\_value = nat\_of\_rgn if dip\_description is returned   then dip\_description else ‘Unknown’  NB. A value of ‘X’ will result in ‘Unknown’ | registration\_circumstance |  | Unknown |
|  | ‘N’ | included\_in\_prices\_flag |  |  |
| dss\_tax\_gst.exempt\_supp\_ind | if exempt\_supp\_ind = ‘Y’  then ‘Y’ else ‘N’ | exempt\_supplies\_flag |  |  |
| dss\_tax\_gst.exp\_ind | if exp\_ind in (‘E’,’B’)  then ‘Y’ else ‘N’ | exporter\_flag |  |  |
| dss\_tax\_gst.exp\_ind | if exp\_ind in (‘I’,’B’)  then ‘Y’ else ‘N’ | importer\_flag |  |  |
|  | ‘N’ | unpoliced\_filer\_flag |  |  |
| dss\_tax\_gst.self\_inv\_ind | if self\_inv\_ind = ‘Y’  then ‘Y’ else ‘N’ | self\_invoicer\_flag |  |  |
|  | ‘N’ | hospice\_filer\_flag |  |  |
|  | ‘N/A’ | hospice\_type |  |  |
|  | ‘N/A’ | hospice\_type\_desc |  |  |
|  | ‘N’ | gst\_on\_remote\_srvcs\_flag |  |  |
|  | ‘N/A’ | gst\_on\_remote\_srvcs\_type |  |  |
|  | ‘N/A’ | gst\_on\_remote\_srvcs\_code |  |  |
|  | ‘N’ | gst\_zero\_rated\_fin\_srv\_flag |  |  |
|  | ‘N’ | gst\_non\_resident\_claimant\_flag |  |  |
|  | ‘N’ | taxable\_activity\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

# Transform Rules: START

## Step 1 – Selecting Valid Values from Source Tables

The purpose of this step is to reduce the number of records from the source object to those that are relevant to the processing. This may include only extracting a subset of rows as well as excluding records that are not deemed to be valid.

The Filtering Criteria will apply to both the initial raw ingestion and subsequent raw ingestion for incremental loads. For an incremental load, changes for each source table will be identified by the corresponding change capture column.

| **Source Table** | **Filtering Criteria** | **Transform Rule/Logic** | **Change Capture Column** | **Business Active**  **Timestamps** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| app\_tblAccount |  |  | fdtmwhen\_info |  | See Derived Tables  See notes 1, 2, 3 |
| app\_tblCustomer | record\_expiry\_timestamp > $$EDWTS1 and fstrcustomertype in (‘IND’, 'CHD',’COM’) |  | fdtmwhen\_info |  | See notes 1, 4 |
| app\_tblcustomerlevel | record\_expiry\_timestamp > $$EDWDT |  | fdtmwhen |  | See note 1 |
| app\_tblProfile | record\_expiry\_timestamp > $$EDWTS1 |  | fdtmwhen |  | See note 1 |
| app\_tblid | record\_expiry\_timestamp > $$EDWTS1 and fstrIdType in (‘IRD’, ’ACCIRD’, ‘ACC’) | if fdtmcommence  = ftdmcease  then fblnActive = 0 | fdtmwhen | fdtmcommence  ftdmcease | See note 5 |
| app\_tblIndicator | record\_expiry\_timestamp > $$EDWTS1 and fintLevel in (0,1,2) | if fdtmcommence  = ftdmcease  then fblnActive = 0 | fdtmwhen | fdtmcommence  ftdmcease | See note 6  See Derived Tables |
| app\_tblNameRecord | record\_expiry\_timestamp > $$EDWTS1 and fstrNameType in (‘LGL’, ’DBA’, ’DBACST’) | if fdtmeffectivefrom  = fdtmeffectiveto  then fblnActive = 0 | fdtmwhen | fdtmeffectivefrom  fdtmeffectiveto | See notes 1, 8  See Derived Tables |
| app\_tblNZ\_AccGSTInfo |  |  | fdtmwhen |  | See note 7 |
| app\_tblNZ\_AccIncome |  |  | fdtmwhen |  | See note 7 |
| app\_tblNZ\_AccIPE |  |  | fdtmwhen |  | See note 7 |
| app\_tblNZ\_AccITN |  |  | fdtmwhen |  | See note 7 |
| app\_tblnz\_AccPieInfo |  |  | fdtmwhen |  | See note 7 |
| app\_tblNZ\_AccPSO |  |  | fdtmwhen |  | See note 7 |
| app\_tblNZ\_AccountStd |  |  | fdtmwhen |  | See note 7 |
| app\_tblNZ\_GSTReg |  |  | fdtmwhen |  | See note 7 |
| ref\_lanAccountType | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lanCloseReason | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lancustomerlevel | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lanFiling | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lanListItem | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lanNZ\_EmployerGroup | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lanNZ\_EmployerType | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lanprofiletype | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lannz\_inctaxexemptionrsn | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_lanindicator | fstrLanguage = ‘ENG’ |  | record\_effective\_timestamp |  | See note 7 |
| ref\_rfrNZ\_Filing |  |  |  |  | See note 9 |

**Notes:**

|  |  |
| --- | --- |
| 1 | This table only holds the current version (i.e. flngVer always 0) |
| 2 | Start accounts were migrated by releases R1, R2 and R3. For each release an EDW-Start Cut-off date is specified to avoid intra-day gaps between contiguous EDW and Start data. |
| 3 | Because the Accounts table (i.e. app\_tblAccount) joins to all other tables in this sub-domain, only the filtering criteria based on EDW-Start Cut-off dates need to be specified for the Accounts table. |
| 4 | This table is already being built for party (same filters, NK, dedupe rules) so it is not required to be built again |
| 5 | IRD is at the customer level (i.e. fintProfileNumber=0). ACCIRD is at the account level (i.e. fintProfileNumber=2). |
| 6 | Indicators may be stored at the account (fintLevel=2), profile (fintLevel=1) or customer (fintLevel=0) levels. |
| 7 | This table is not versioned (i.e. there is no flngVer column) |
| 8 | Name record can belong to the customer’s primary profile (fintProfileNumber =1), the account’s profile or the account’s parent profile. Accounts do not have a ‘preferred’ name - only legal and trading names. |
| 9 | Because this table is a full replace, there are no change capture columns. |

**Derived Tables:**

| **Derived Table** | **Source Table(s)** | **Derivation Rules** | **Change Capture Column** | **Comments** |
| --- | --- | --- | --- | --- |
| crw\_as\_tblindicator | app\_tblindicator (post filter)  crw\_as\_tblaccount (post clean raw processing) | **Derive table crw\_as\_tblindicator:**  **LEFT JOIN** crw\_as\_tblaccount (a) to app\_tblindicator (i)  on i.flngCustomerKey = a.flngCustomerKey  and i.flngAccountKey = (case when i.fintlevel = 2 then a.flngAccountKey  else 0 end)  and i.fintprofilenumber = (case when i.fintlevel = 2 then a.fintProfileNumber  when i.fintlevel = 1 then a.fintParentProfileNumber  when i.fintlevel = 0 then 0 end)  **Return**:  ***derived\_Account\_Key*** from ifnull(a.flngAccountKey, -1)  ***derived\_Profile\_Number*** from ifnull(a.fintProfileNumber, -2)  all columns from app\_tblIndicator |  | See Apendix F for sample SQL  See note 2 |
| crw\_as\_tblindicator\_misc | crw\_as\_tblindicator | **Filter from table crw\_as\_tblindicator:**  where fstrIndicator in ('ACCHLT', 'GRNLST', 'FRDRED', 'ZROTTR', 'PDEXMP') | fdtmwhen | See note 1 |
| crw\_as\_tblindicator\_exmpt | crw\_as\_tblindicator | **Filter from table crw\_as\_tblindicator:**  fstrIndicator in ('CRTEXM', 'SLSRDE', 'ESSEXM', 'RREXM', 'FTXEXM', 'ELCEXM', 'RWTEXM') | fdtmwhen | See note 1 |
| crw\_as\_tblindicator\_audact | crw\_as\_tblindicator | **Filter from table crw\_as\_tblindicator:**  fstrIndicator in ('AUDIT', 'AUDPRA', 'AUDSFP')  and fdtmcommence <= fdtmcease | fdtmwhen | See note 1 |
| crw\_as\_tblindicator\_dontmail | crw\_as\_tblindicator | **Filter from table crw\_as\_tblindicator:**  fstrIndicator in ('INVADD', 'STPMAL') | fdtmwhen | See note 1 |
| crw\_as\_tblnamerecord | app\_tblNameRecord (post filter)  crw\_as\_tblaccount (post clean raw processing) |  |  | See Appendix F for sample SQL  See note 2 |

**Notes:**

|  |  |
| --- | --- |
| 1 | Indicators may be stored at the account (fintLevel=2), profile (fintLevel=1) or customer (fintLevel=0) levels. |
| 2 | This will create a dependency on the account commit job running before the cleanse jobs for names and indicators, the remaining indicator jobs will also be dependent on this commit job (SW). |
|  |  |

## Step 2 – De-duplication of Records

The purpose of this step is to remove duplicates over time so that the resulting rows for each natural key are contiguous and non-overlapping in time. The default de-duplication process (see [Appendix D](#_Appendix_D_–)) is to be following unless explicitly stated otherwise.

NB. For Start, active flag should always be the first tie-breaker for de-duplication.

The derived field “number\_of\_valid\_days” used to rank ‘duplicate’ records is to be calculated as the number of days that the business active timestamps ‘overlap’ the system active timestamps. See [Appendix D](#_Appendix_D_–) for details.

| **Source Table** | **De-duping Rule** | **Natural Key** | **Business Active**  **Timestamps** | **Tie-Breaker(s)** |
| --- | --- | --- | --- | --- |
| app\_tblAccount | No de-duplication required | flngAccountKey |  |  |
| app\_tblCustomer | No de-duplication required | flngCustomerKey |  |  |
| app\_tblcustomerlevel | No de-duplication required | flngDocKey |  |  |
| app\_tblProfile | No de-duplication required | flngCustomerKey  fintProfileNumber |  |  |
| app\_tblid | Use the default de-duplication process (see Appendix D) | flngCustomerKey  flngAccountKey fstridtype | fdtmcommence  ftdmcease | Rank:   1. fblnactive (1 before 0) 2. number\_of\_valid\_seconds (highest first) 3. ~~record\_expiry\_timestamp (latest first)~~ 4. ~~ftdmcease (latest first)~~ 5. ~~fdtmcommence (latest first)~~ 6. fdtmWhen (latest first) 7. fblnDefault (1 before 0) 8. flngidkey (highest first) 9. flngverlast (highest first) |
| app\_tblIndicator | Use the default de-duplication process (see Appendix D) | flngCustomerKey  derived\_account\_key fstrindicator | fdtmcommence  ftdmcease | Rank:   1. fblnactive (1 before 0) 2. ~~record\_expiry\_timestamp (latest first)~~ 3. ~~fintLevel (desc) – required to priortise across hierarchy~~ 4. ~~ftdmcease (latest first)~~ 5. ~~fdtmcommence (latest first)~~ 6. number\_of\_valid\_seconds (highest first) 7. fintLevel (desc) – required to priortise across hierarchy 8. fdtmWhen (latest first) 9. flngindicatorkey (highest first) 10. flngverlast (highest first) |
| app\_tblIndicator\_exmpt | Use the default de-duplication process (see Appendix D) | flngCustomerKey  derived\_account\_key | fdtmcommence  ftdmcease | Same as app\_tblIndicator |
| app\_tblIndicator\_audact | Use the default de-duplication process (see Appendix D) | flngCustomerKey  derived\_account\_key | fdtmcommence  ftdmcease | Same as app\_tblIndicator |
| app\_tblIndicator\_dontmail | Use the default de-duplication process (see Appendix D) | flngCustomerKey  derived\_account\_key | fdtmcommence  ftdmcease | Same as app\_tblIndicator |
| app\_tblIndicator\_misc | Use the default de-duplication process (see Appendix D) | flngCustomerKey  derived\_account\_key | fdtmcommence  ftdmcease | Same as app\_tblIndicator |
| app\_tblNameRecord | Use the default de-duplication process (see Appendix D) | flngCustomerKey  derived\_Account\_Key  derived\_Name\_Type | fdtmeffectivefrom  fdtmeffectiveto | Rank:   1. fblnactive (1 before 0) 2. record\_expiry\_timestamp (latest first) 3. derived\_Level (highest first) 4. fdtmeffectiveto (latest first) 5. fdtmeffectivefrom (latest first) 6. fl64namekey (highest first) 7. flngverlast (highest first) |
| app\_tblNZ\_AccGSTInfo | No de-duplication required | flngdockey |  |  |
| app\_tblNZ\_AccIncome | No de-duplication required | flngdockey |  |  |
| app\_tblNZ\_AccIPE | No de-duplication required | flngdockey |  |  |
| app\_tblNZ\_AccITN | No de-duplication required | flngdockey |  |  |
| app\_tblnz\_AccPieInfo | No de-duplication required | flngdockey |  |  |
| app\_tblNZ\_AccPSO | No de-duplication required | flngdockey |  |  |
| app\_tblNZ\_AccountStd | No de-duplication required | flngdockey |  |  |
| app\_tblNZ\_GSTReg | No de-duplication required | flngdockey |  |  |
| ref\_lanAccountType | No de-duplication required | fstrAccountType |  |  |
| ref\_lanCloseReason | No de-duplication required | fstrReason |  |  |
| ref\_lancustomerlevel | No de-duplication required | fstrLanguage fstrCustomerLevel |  |  |
| ref\_lanFiling | No de-duplication required | fstrFiling |  |  |
| ref\_lanListItem | No de-duplication required | fstrField fstrListItem |  |  |
| ref\_lanNZ\_EmployerGroup | No de-duplication required | fstrEmployerGroup |  |  |
| ref\_lanNZ\_EmployerType | No de-duplication required | fstrEmployerType |  |  |
| ref\_lanprofiletype | No de-duplication required | fstrprofiletype |  |  |
| ref\_lannz\_inctaxexemptionrsn | No de-duplication required | fstrExemptionReason |  |  |
| ref\_lanindicator | No de-duplication required | fstrindicator |  |  |
| ref\_rfrNZ\_Filing | No de-duplication required | fstrfiling |  |  |

**Notes:**

|  |  |
| --- | --- |
| 1 | Must include profile because an account can be transferred back to its customer with a different profile (see A/c 861510912) |

## Step 3 – Compressing the timeline

After de-duplicating records in Step 2, one or more contiguous records sharing the same Natural Key may have the same values for all columns used for comparison. This step will ‘compress’ (i.e. merge) any such contiguous records using the minimum Record\_Effective\_Date and maximum Record\_Expiry\_Date for the resulting timeline.

NB. Business active timestamps/dates should always be included in the “columns to compare”.

| **Source Table** | **Natural Key** | **Business Active**  **Timestamps** | **Columns to Compare** | **Dates for Timelines** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| app\_tblAccount | flngAccountKey |  | flngCustomerKey fstrAccountType fstrFiling fdtmCommence fdtmCease fstrStatus fstrClosureReason fdtmCreated | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblCustomer | flngCustomerKey |  | fstrCustomerType  fdtmcommence  fdtmcease  fdtmcreated | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblcustomerlevel | flngDocKey |  | fstrCustomerLevel | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblProfile | flngCustomerKey  fintProfileNumber |  | fstrProfileType fintParentProfileNumber | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblid | flngCustomerKey  flngaccountkey  fstridtype | fdtmcommence  ftdmcease | fstrid fblnactive  fdtmcommence  fdtmcease | record\_effective\_timestamp record\_expiry\_timestamp | See note 1 |
| app\_tblIndicator | flngcustomerkey  derived\_account\_key fstrindicator | fdtmcommence  ftdmcease | fstrindicator fblnactive  fdtmcommence  ftdmcease | record\_effective\_timestamp record\_expiry\_timestamp | See note 1 |
| app\_tblIndicator\_exmpt | flngcustomerkey  derived\_account\_key | fdtmcommence  ftdmcease | fstrindicator fblnactive  fdtmcommence  fdtmcease | record\_effective\_timestamp record\_expiry\_timestamp | See note 1 |
| app\_tblIndicator\_audact | flngcustomerkey  derived\_account\_key | fdtmcommence  ftdmcease | fstrindicator fblnactive  fdtmcommence  fdtmcease | record\_effective\_timestamp record\_expiry\_timestamp | See note 1 |
| app\_tblIndicator\_dontmail | flngcustomerkey  derived\_account\_key | fdtmcommence  ftdmcease | fstrindicator fblnactive  fdtmcommence  fdtmcease | record\_effective\_timestamp record\_expiry\_timestamp | See note 1 |
| app\_tblIndicator\_misc | flngcustomerkey  derived\_account\_key | fdtmcommence  ftdmcease | fstrindicator fblnactive  fdtmcommence  fdtmcease | record\_effective\_timestamp record\_expiry\_timestamp | See note 1 |
| app\_tblNameRecord | flngCustomerKey derived\_Account\_Key derived\_Name\_Type | fdtmeffectivefrom  fdtmeffectiveto | fstrlistformatname  fstrtitle  fstrfirstname  fstrmiddlename  fstrlastname fblnactive  fdtmeffectivefrom  fdtmeffectiveto | record\_effective\_timestamp record\_expiry\_timestamp | See note 1 |
| app\_tblNZ\_AccGSTInfo | flngdockey |  | fblnPricesIncGST  fblnExemptSupplies  fblnImporter  fblnExporter  fstrRegCircumstance  fblnUnpolicedFiler  fblnSelfInvoicing  fblnHospiceFiler  fstrHospiceType  fstrLastAccountingBasis  fblnGSTOnElectronicServices  fstrGORSType  fblnZeroRatedFinancialServices  fblnNonResBusinessClaimant | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblNZ\_AccIncome | flngdockey |  | fdtmBalanceDate  fstrExemptionReason | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblNZ\_AccIPE | flngdockey |  | fstrExemptionReason | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblNZ\_AccPSO | flngdockey |  | fblnIR56  fstrFilingOption  fblnPayDayFiler  fdtmPayDayFilerStart  fblnCSE  fblnSLE  fblnKSE  fblnKSR  fblnSSC  fstrEmployerGroup | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblNZ\_AccITN | flngdockey |  | fblnTransTasman  fblnAgentNonResInsurer | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblnz\_AccPieInfo | flngdockey |  | fstrPIEType  fstrFilingOption  fstrAttributionPeriod  fstrFDRCalcPeriod | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblNZ\_AccountStd | flngdockey |  | fintHeritageLocationNumber | record\_effective\_timestamp record\_expiry\_timestamp |  |
| app\_tblNZ\_GSTReg | flngdockey |  | fblnTaxableActivity | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanAccountType | fstrAccountType |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanCloseReason | fstrReason |  | fstrdecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lancustomerlevel | fstrCustomerLevel |  | fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanFiling | fstrFiling |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanListItem | fstrField fstrListItem |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanNZ\_EmployerGroup | fstrEmployerGroup |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanNZ\_EmployerType | fstrEmployerType |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanNZ\_IncTaxExemptionRsn | fstrExemptionReason |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanprofiletype | fstrprofiletype |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_lanindicator | fstrindicator |  | fstrDecode1  fstrDecode2 | record\_effective\_timestamp record\_expiry\_timestamp |  |
| ref\_rfrNZ\_Filing | fstrfiling |  |  |  | See note 2 |

**Notes:**

|  |  |
| --- | --- |
| 1 | Because accounts are transferrable, the combination of CustomerKey, ProfileNumber and AccountKey is required for the Natural Key. |
| 2 | Because this table is a full replace, there are no columns to compare or dates for timelines. |

## Step 4 – Combining all change timeline

The purpose of this step is to create a UNION of all change timestamps for the Account data that will become the basis for all change records created. These timelines are stored in the Refined Layer in table rfn\_as\_all\_dates. Each record in the timeline is bound in time from record\_effective\_timestamp to record\_expiry\_timestamp.

A change timestamp is generated for each Record\_Effective\_Timestamp at a 1-minute granularity, additionally where a Business Start Date and/or Business End date are listed below, a change timestamp should be created for each at 1-minute granularity. Where a Business\_Start\_Date contains sub-minute data the time should be **rounded up** to the next minute. A Business\_End\_Date should be rounded up to the next unit of time. (If an end date preseted at the day grain is shown as ’01-10-1989’ rounding up to the next unit of time means creating a change timestamp of ’02-01-1989’. This is done as we need to create a change timestamp immediately after the business end date, so that we show a Party record ***not*** having the “ceased” value.)

When there are 2 or more change timestamps each 1 minute apart, select the only the latest one. This is to prevent issues with the setting of Record\_Expiry\_Timestamp at a 1-minute grain and effectively takes the final “accumulated” record when multiple changes come through at the same time (or within a minute of each other).

e.g. A set of records having change timestamps of:

01-Mar-2019 10:52 A B NULL NULL NULL  
01-Mar-2019 10:53 A B X NULL NULL  
01-Mar-2019 10:54 A B X Y NULL  
01-Mar-2019 10:55 A B X Y Z

will be treated as:

01-Mar-2019 10:55 A B X Y Z

This may result in additional change records being created, but they will be compressed in the final step of the Refined Layer build for each target table.

NB. Only tables which are combined to the main timeline are shown here, reference data (lookup) are ignored.

| **Raw Table** | **Clean Raw Table** | **Business Start date** | **Business End date** | **Comments** |
| --- | --- | --- | --- | --- |
| app\_tblAccount | crw\_as\_tblaccount | fdtmCreated |  |  |
| app\_tblCustomer | crw\_ps\_tblcustomer |  |  | This table is reused from DIP Party |
| app\_tblcustomerlevel | crw\_ps\_tblcustomerlevel |  |  | This table is reused from DIP Party |
| app\_tblProfile | crw\_as\_tblprofile |  |  |  |
| app\_tblid | crw\_as\_tblid | fdtmcommence | if ftdmcease is not = $$HDT  then days\_add(ftdmcease,1) |  |
| app\_tblIndicator | crw\_as\_tblindicator | fdtmcommence | if ftdmcease is not = $$HDT   then minutes\_add(ftdmcease,1) |  |
| app\_tblIndicator | crw\_as\_tblindicator\_audact | fdtmcommence | if ftdmcease is not = $$HDT   then minutes\_add(ftdmcease,1) |  |
| app\_tblIndicator | crw\_as\_tblindicator\_dontmail | fdtmcommence | if ftdmcease is not = $$HDT   then minutes\_add(ftdmcease,1) |  |
| app\_tblIndicator | crw\_as\_tblindicator\_exmpt | fdtmcommence | if ftdmcease is not = $$HDT   then minutes\_add(ftdmcease,1) |  |
| app\_tblIndicator | crw\_as\_tblindicator\_misc | fdtmcommence | if ftdmcease is not = $$HDT   then minutes\_add(ftdmcease,1) |  |
| app\_tblNameRecord | crw\_as\_tblnamerecord | fdtmeffectivefrom | if fdtmeffectiveto is not = $$HDT  then days\_add(fdtmeffectiveto,1) | fdtmEffectiveFrom and fdtmEffectiveTo are already driving the Record\_Effective\_Timestamp and Record\_Expiry\_Timestamp |
| app\_tblNZ\_AccGSTInfo | crw\_as\_tblnz\_acc\_gst\_info |  |  |  |
| app\_tblNZ\_AccIncome | crw\_as\_tblnz\_acc\_income |  |  |  |
| app\_tblNZ\_AccIPE | crw\_as\_tblnz\_acc\_ipe |  |  |  |
| app\_tblNZ\_AccITN | crw\_as\_tblnz\_acc\_itn |  |  |  |
| app\_tblnz\_AccPieInfo | crw\_as\_tblnz\_acc\_pie\_info |  |  |  |
| app\_tblNZ\_AccPSO | crw\_as\_tblnz\_acc\_pso |  |  |  |
| app\_tblNZ\_AccountStd | crw\_as\_tblnz\_account\_std |  |  |  |
| app\_tblNZ\_GSTReg | crw\_as\_tblnz\_gst\_reg |  |  |  |
| ref\_lanAccountType | crw\_as\_lan\_account\_type |  |  |  |
| ref\_lanCloseReason | crw\_as\_lan\_close\_reason |  |  |  |
| ref\_lancustomerlevel | crw\_ps\_lancustomerlevel |  |  |  |
| ref\_lanFiling | crw\_as\_lan\_filing |  |  |  |
| ref\_lanListItem | crw\_as\_lan\_list\_item |  |  |  |
| ref\_lanNZ\_EmployerGroup | crw\_as\_lannz\_employer\_group |  |  |  |
| ref\_lanNZ\_EmployerType | crw\_as\_lannz\_employer\_type |  |  |  |
| ref\_lanprofiletype | crw\_as\_lan\_profile\_type |  |  |  |
| ref\_lannz\_inctaxexemptionrsn | crw\_as\_lannz\_inctax\_exmpt\_rsn |  |  |  |
| ref\_lanindicator | crw\_ps\_lan\_indicator |  |  | This table is reused from DIP Party |
| ref\_rfrNZ\_Filing | crw\_as\_rfrNZ\_Filing |  |  |  |

## Step 5 – Create a single timeline for each Account

Create a single timeline for each Account. The account timeline is stored the Refined Layer in table rfn\_as\_timeline. Each record in the timeline is bound in time from record\_effective\_timestamp to record\_expiry\_timestamp.

| **Condition** | **Cut-off Boundaries** | **Rule Logic** | **Comments** |
| --- | --- | --- | --- |
| All Start accounts | Lower cut-off date | Use the later of:   1. the earliest Start account\_start\_date 2. default interleaving date |  |
| Upper cut-off date | $$HIDATE | No cut-off is required for the upper boundary |

## Step 6a – Merge change timelines with Account timeline

Merge the change timelines from Step#4 with the single Account timeline from Step#5. Each record in the timeline has a natural key of flngaccountkey, record\_effective\_timestamp and record\_expiry\_timestamp.

NB. When the offset for the business active timestamps (or dates) is greater than one minute, one day must be added to the ‘end’ timestamp when comparing to the timeline.record\_effective\_date.

| **Clean Raw Table** | **Filtering Criteria** | **Business Active**  **Timestamps** | **Time-stamp Offset** | **Comments** |
| --- | --- | --- | --- | --- |
| crw\_as\_tblaccount |  |  |  |  |
| crw\_ps\_tblcustomer |  |  |  |  |
| crw\_ps\_tblcustomerlevel |  |  |  |  |
| crw\_as\_tblprofile |  |  |  |  |
| crw\_as\_tblid | ***timeline.flngaccountkey*** = flngaccountkey  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_effective\_timestamp*** >= fdtmcommence  and days\_add(***timeline.record\_effective\_timestamp***,-1) <= fdtmcease  and fblnactive = 1 | fdtmcommence  ftdmcease | Day | See notes 1, 2 |
| crw\_as\_tblindicator | ***timeline.flngaccountkey*** = flngaccountkey  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp ***timeline.record\_effective\_timestamp***   between fdtmcommence and fdtmcease  and fblnactive = 1 | fdtmcommence  ftdmcease | Millisecond | See note 1 |
| crw\_as\_tblindicator\_audact | ***timeline.flngaccountkey*** = flngaccountkey  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp ***timeline.record\_effective\_timestamp***   between fdtmcommence and fdtmcease  and fblnactive = 1 | fdtmcommence  ftdmcease | Millisecond | See note 1 |
| crw\_as\_tblindicator\_dontmail | ***timeline.flngaccountkey*** = flngaccountkey  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp ***timeline.record\_effective\_timestamp***   between fdtmcommence and fdtmcease  and fblnactive = 1 | fdtmcommence  ftdmcease | Millisecond | See note 1 |
| crw\_as\_tblindicator\_exmpt | ***timeline.flngaccountkey*** = flngaccountkey  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp ***timeline.record\_effective\_timestamp***   between fdtmcommence and fdtmcease  and fblnactive = 1 | fdtmcommence  ftdmcease | Millisecond | See note 1 |
| crw\_as\_tblindicator\_misc | ***timeline.flngaccountkey*** = flngaccountkey  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp ***timeline.record\_effective\_timestamp***   between fdtmcommence and fdtmcease  and fblnactive = 1 | fdtmcommence  ftdmcease | Millisecond | See note 1 |
| crw\_as\_tblnamerecord | ***timeline.flngaccountkey*** = flngaccountkey  and ***timeline.record\_effective\_timestamp*** between record\_effective\_timestamp and record\_expiry\_timestamp  and ***timeline.record\_effective\_timestamp*** >= fdtmeffectivefrom  and days\_add(***timeline.record\_effective\_timestamp***,-1) <= fdtmeffectiveto | fdtmeffectivefrom  fdtmeffectiveto | Day | See note 2 |
| crw\_as\_tblnz\_acc\_gst\_info |  |  |  |  |
| crw\_as\_tblnz\_acc\_income |  |  |  |  |
| crw\_as\_tblnz\_acc\_ipe |  |  |  |  |
| crw\_as\_tblnz\_acc\_itn |  |  |  |  |
| crw\_as\_tblnz\_acc\_pie\_info |  |  |  |  |
| crw\_as\_tblnz\_acc\_pso |  |  |  |  |
| crw\_as\_tblnz\_account\_std |  |  |  |  |
| crw\_as\_tblnz\_gst\_reg |  |  |  |  |
| crw\_as\_lan\_account\_type |  |  |  |  |
| crw\_as\_lan\_close\_reason |  |  |  |  |
| crw\_ps\_lancustomerlevel |  |  |  |  |
| crw\_as\_lan\_filing |  |  |  |  |
| crw\_as\_lan\_list\_item |  |  |  |  |
| crw\_as\_lannz\_employer\_group |  |  |  |  |
| crw\_as\_lannz\_employer\_type |  |  |  |  |
| crw\_as\_lan\_profile\_type |  |  |  |  |
| crw\_as\_lannz\_inctax\_exmpt\_rsn |  |  |  |  |
| crw\_ps\_lan\_indicator |  |  |  |  |
| crw\_as\_rfrNZ\_Filing |  |  |  |  |

**Notes:**

|  |  |
| --- | --- |
| 1 | ‘Inactive’ (i.e. fblnActive=0) records with flngver=0 are to be ignored in this step. They are required for Step#5 to ensure that the record\_expiry\_timestamp is set correctly for contiguous records and must now be removed. |
| 2 | When the ‘offset’ for the business active timestamps (or dates) is greater than one minute, one unit of the ‘offset’ value must be subtracted from the timeline.record\_effective\_date when compared to the record’s ‘end’ timestamp. |

## Step 6b – Compress the Post-Merge Timelines

After timelines are merged in Step#6a, one or more contiguous records sharing the same Natural Key may have the same values for all columns used for comparison. This step will ‘compress’ (i.e. merge) any such contiguous records using the minimum Record\_Effective\_Date and maximum Record\_Expiry\_Date for the resulting timeline.

| **Source Table** | **Natural Key** | **Columns to Compare** | **Columns for Timelines** | **Time-stamp Offset** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| rfn\_as\_account | dip\_account\_id | dip\_account\_id  customer\_key  account\_key  ird\_number  account\_ird\_number  start\_profile\_number  start\_parent\_profile\_type  start\_parent\_profile\_number  heritage\_location\_no  legal\_name  trading\_name  title  given\_name  other\_given\_name  family\_name  account\_type  account\_type\_code  account\_start\_date  account\_ceased\_date  account\_status  account\_status\_code  security\_level  security\_level\_code  cessation\_reason  cessation\_reason\_code  filing\_frequency\_period  filing\_frequency  filing\_frequency\_code  exemption\_reason  exemption\_reason\_code  exempted\_flag  green\_listed\_flag  red\_listed\_flag  account\_halt\_flag  audit\_case\_active\_reason  audit\_case\_active\_flag  stop\_mail\_reason  stop\_mail\_flag | Record\_Effective\_Timestamp Record\_Expiry\_Timestamp | Minute |  |
| rfn\_as\_account\_gst | dip\_account\_id | dip\_account\_id  customer\_key  account\_key  ird\_number  account\_ird\_number  start\_profile\_number  start\_parent\_profile\_type  start\_parent\_profile\_number  heritage\_location\_no  legal\_name  trading\_name  title  given\_name  other\_given\_name  family\_name  account\_type  account\_type\_code  account\_start\_date  account\_ceased\_date  account\_status  account\_status\_code  security\_level  security\_level\_code  cessation\_reason  cessation\_reason\_code  filing\_frequency\_period  filing\_frequency  filing\_frequency\_code  exemption\_reason  exemption\_reason\_code  exempted\_flag  green\_listed\_flag  red\_listed\_flag  account\_halt\_flag  audit\_case\_active\_reason  audit\_case\_active\_flag  stop\_mail\_reason  stop\_mail\_flag  accounting\_basis  registration\_circumstance  included\_in\_prices\_flag  exempt\_supplies\_flag  exporter\_flag  importer\_flag  unpoliced\_filer\_flag  self\_invoicer\_flag  hospice\_filer\_flag  hospice\_type  hospice\_type\_desc  gst\_on\_electronic\_srvcs\_flag  gst\_on\_remote\_services\_type  gst\_on\_remote\_srvcs\_code  gst\_zero\_rated\_financial\_services\_flag  gst\_non\_resident\_claimant\_flag  taxable\_activity\_flag | Record\_Effective\_Timestamp Record\_Expiry\_Timestamp | Minute |  |
| rfn\_as\_account\_inc | dip\_account\_id | dip\_account\_id  customer\_key  account\_key  ird\_number  account\_ird\_number  start\_profile\_number  start\_parent\_profile\_type  start\_parent\_profile\_number  heritage\_location\_no  legal\_name  trading\_name  title  given\_name  other\_given\_name  family\_name  account\_type  account\_type\_code  account\_start\_date  account\_ceased\_date  account\_status  account\_status\_code  security\_level  security\_level\_code  cessation\_reason  cessation\_reason\_code  filing\_frequency\_period  filing\_frequency  filing\_frequency\_code  exemption\_reason  exemption\_reason\_code  exempted\_flag  green\_listed\_flag  red\_listed\_flag  account\_halt\_flag  audit\_case\_active\_reason  audit\_case\_active\_flag  stop\_mail\_reason  stop\_mail\_flag  balance\_date  zero\_tailored\_rate\_flag  transtasman\_flag  agent\_non\_res\_insurer\_flag | Record\_Effective\_Timestamp Record\_Expiry\_Timestamp | Minute |  |
| rfn\_as\_account\_pay | dip\_account\_id | dip\_account\_id  customer\_key  account\_key  account\_ird\_number  start\_profile\_number  start\_parent\_profile\_type  start\_parent\_profile\_number  heritage\_location\_no  legal\_name  trading\_name  title  given\_name  other\_given\_name  family\_name  account\_type  account\_type\_code  account\_start\_date  account\_ceased\_date  account\_status  account\_status\_code  security\_level  security\_level\_code  cessation\_reason  cessation\_reason\_code  filing\_frequency\_period  filing\_frequency  filing\_frequency\_code  exemption\_reason  exemption\_reason\_code  exempted\_flag  green\_listed\_flag  red\_listed\_flag  account\_halt\_flag  audit\_case\_active\_reason  audit\_case\_active\_flag  stop\_mail\_reason  stop\_mail\_flag  ir56\_flag  filing\_option  filing\_option\_code  payday\_filer\_flag  dayfiler\_start\_date  child\_support\_flag  student\_loan\_flag  pay\_kiwisaver\_member\_flag  kiwisaver\_employer\_flag  superannuation\_flag  employer\_group  employer\_group\_code  payday\_variation\_flag | Record\_Effective\_Timestamp Record\_Expiry\_Timestamp | Minute |  |
| rfn\_as\_account\_pie | dip\_account\_id | dip\_account\_id  customer\_key  account\_key  ird\_number  account\_ird\_number  start\_profile\_number  start\_parent\_profile\_type  start\_parent\_profile\_number  heritage\_location\_no  legal\_name  trading\_name  title  given\_name  other\_given\_name  family\_name  account\_type  account\_type\_code  account\_start\_date  account\_ceased\_date  account\_status  account\_status\_code  security\_level  security\_level\_code  cessation\_reason  cessation\_reason\_code  filing\_frequency\_period  filing\_frequency  iling\_frequency\_code  exemption\_reason  exemption\_reason\_code  exempted\_flag  green\_listed\_flag  red\_listed\_flag  account\_halt\_flag  audit\_case\_active\_reason  audit\_case\_active\_flag  stop\_mail\_reason  stop\_mail\_flag  pie\_type  pie\_type\_code  filing\_option  filing\_option\_code  attribution\_period  fdr\_cal\_period | Record\_Effective\_Timestamp Record\_Expiry\_Timestamp | Minute |  |
| rfn\_as\_party\_account | dip\_party\_id  dip\_account\_id | dip\_party\_id  dip\_account\_id  relationship\_type  dip\_confidence\_level | Record\_Effective\_Timestamp Record\_Expiry\_Timestamp | Minute |  |

## Step 7 – Transform Rule Logic

### Example SQL for Context

Appendix F contains the example SQL statements for Start Accounts. The “Select” column names in these SQL statements will be refered to in the “Source Table/Column” fields of the transformation rules below and are intended to give context to those rules as well as a guide on how to merge all the source objects into the final Account Type data streams.

### Overarching Rules/logic

1. Account Type records are to be versioned to share the same keys. Therefore, any change to the versioning for an Account will result in an identical change to the versioning for either the corresponding Account Type records.
2. Required date validation functions are:

| **Function Name** | **Purpose** | **Logic** |
| --- | --- | --- |
| $$is\_date\_valid$$ | Determine if the specified date (parameter: $in\_date$) is valid | FUNCTION $$check\_date\_valid$$ (in $in\_date$) return boolean  If $in\_date$ is not between ‘1880-JAN-01’ and $today$  then return FALSE else if $in\_date$ is ‘1900-JAN-01’  then return FALSE else return TRUE |

1. Relevant Account types are:

|  |  |  |
| --- | --- | --- |
| **DIP Account Type Table** | **Account Type** | **Description** |
| ACCOUNT | All | All Account types |
| PARTY\_ACCOUNT | All | All Account types |
| ACCOUNT\_PAY | PSO | Employment Activities (Payroll) |
| ACCOUNT\_PIE | PIE | Portfolio Investment Entity |
| ACCOUNT\_INC | IIT | Income Tax Individuals |
| ITN | Income Tax Non-Individuals |
| ACCOUNT\_GST | GST | Goods and Services Tax |

### Start Party Account

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| app\_tblAccount.flngCustomerKey | concatenate(‘DIP-C-’, flngCustomerKey) | dip\_party\_id | PK |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_id | PK |  |
|  | *Standard processing field* | record\_effective\_timestamp | PK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
|  | ‘Owned’ | relationship\_type |  |  |
| app\_tblAccount.fdtmCommence | copy | relationship\_effective\_from |  |  |
| app\_tblAccount.fdtmCease | copy | relationship\_effective\_to |  | ‘9999-12-31‘ |
|  | ‘FIRST’ | relationship\_origin |  |  |
|  | ‘HIGH’ | dip\_confidence\_level |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### Start Account

No Filter i.e. for all values of app\_tblAccount.fstrAccountType.

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
|  | Increment by 1 for each record. Starting value must exceed the highest corresponding value for EDW. This key is pervasive across the Account table and all Account Type tables. | dip\_account\_sk | SK |  |
| app\_tblAccount.flngAccountKey | concatenate(‘DIP-A-’, flngaccountkey) | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| app\_tblAccount.flngAccountKey | Copy | account\_key |  |  |
| app\_tblid.fstrId  app\_tblid.fstrIdType | If fstrIdType = ‘ACCIRD’  then lpad(fstrId, 9, ’0’) else ‘Unknown’ | account\_ird\_number |  | Unknown |
| app\_tblid.fstrId  app\_tblid.fstrIdType | If fstrIdType = ‘ACC’  then fstrId  else ‘N/A’ | start\_account\_id |  | N/A |
| app\_tblAccount.fintProfileNumber | Copy | start\_profile\_number |  |  |
| app\_tblProfile.fstrProfileType  ref\_lanprofiletype.fstrdecode2 | Lookup ref\_lanprofiletype on fstrprofiletype if fstrdecode2 is returned  then fstrdecode2 | start\_parent\_profile\_type |  | N/A |
| app\_tblAccount.fintParentProfileNumber | Copy | start\_parent\_profile\_number |  | N/A |
| app\_tblNZ\_AccountStd.fintHeritageLocationNumber | Copy | heritage\_location\_number |  | Unknown |
| app\_tblCustomer.fstrCustomerType app\_tblNameRecord\_app.fstrNameType app\_tblNameRecord\_app.fstrtitle  app\_tblNameRecord\_app.fstrfirstname  app\_tblNameRecord\_app.fstrmiddlename  app\_tblNameRecord\_app.fstrlastname app\_tblNameRecord\_app.fstrListFormatName | if fstrCustomerType = ‘COM’ and fstrNameType = ‘LGL’  then fstrListFormatName  else if fstrCustomerType in (‘IND’, ‘CHD’) and fstrNameType = ‘LGL’  then concatenate(fstrtitle, ‘ ’, fstrfirstname, ‘ ‘,fstrmiddlename, ‘ ‘,fstrlastname)  *Eliminating any spurious spaces caused by one of the fields being NULL*  else ‘Unknown’ | legal\_name |  | Unknown |
| app\_tblNameRecord\_app.fstrListFormatName  app\_tblNameRecord\_app.fstrNameType | if fstrNameType in (‘DBA’, 'DBACST')   then fstrListFormatName  else same as *Target Column legal\_name* | trading\_name |  | Unknown |
| ref\_lanAccountType.fstrdecode2  app\_tblAccount.fstrAccountType | Lookup ref\_lanAccountType on fstrAccountType if fstrdecode2 is returned  then fstrdecode2 | account\_type |  | Unknown |
| app\_tblAccount.fstrAccountType | Copy | account\_type\_code |  | Unknown |
| app\_tblAccount.fdtmCreated  app\_tblAccount.fdtmCommence | if $$is\_date\_valid$$ (fdtmcommence)  then fdtmcommence  else if $$is\_date\_valid$$ (fdtmcreated)   then fdtmcreated  else $$LDT | account\_start\_date |  | $$LDT |
| app\_tblAccount.fdtmCease | if $$is\_date\_valid$$ (fdtmcease)   then fdtmcease  else $$HDT | account\_ceased\_date |  | $$HDT |
| app\_tblAccount.fstrStatus | If fstrStatus = ‘ACT’  then ‘Active’ else if fstrStatus = ‘CLS’  then ‘Closed’ else if fstrStatus = ‘SUS’  then ‘Suspended’ | account\_status |  | Unknown |
| app\_tblAccount.fstrStatus | Copy | account\_status\_code |  | Unknown |
| ~~app\_tblCustomer.flngDocKey~~  app\_tblcustomerlevel.fstrCustomerLevel  ref\_lancustomerlevel.fstrDecode2  *target column: security\_level\_code* | ~~Lookup app\_tblcustomerlevel on flngDocKey if fstrCustomerLevel is returned then~~  lookup ref\_lancustomerlevel   on fstrCustomerLevel = *security\_level\_code*  if fstrDecode2 is returned then  update all records for this DIP\_ACCOUNT\_ID   set to fstrDecode2 **over all time** | security\_level |  | Unknown |
| app\_tblCustomer.flngDocKey  app\_tblcustomerlevel.fstrCustomerLevel  app\_tblAccount.flngCustomerKey  rfn\_party\_sk\_map.start\_customer\_key  rfn\_party\_sk\_map.ird\_no  dss\_special\_clients\_all.ird\_number  dss\_special\_clients\_all.special\_classification | Lookup app\_tblcustomerlevel on flngDocKey if fstrCustomerLevel is returned then  update all records for this DIP\_ACCOUNT\_ID   set to fstrCustomerLevel **over all time**  else lookup rfn\_party\_sk\_map  on start\_customer\_key = flngCustomerKey  if ird\_no is returned then  lookup dss\_special\_clients\_all on ird\_no  if special\_classification is returned then  update all records for this DIP\_ACCOUNT\_ID   set to special\_classification **over all time** | security\_level\_code |  | Unknown |
| ref\_lanCloseReason.fstrdecode1  app\_tblAccount.fstrClosureReason | if fstrStatus = ‘CLS’ then lookup ref\_lanCloseReason on fstrreason = fstrClosureReason  if fstrdecode1 is returned  then fstrdecode1  else ‘Unknown’ else ‘N/A’ | cessation\_reason |  | Unknown |
| app\_tblAccount.fstrClosureReason | if fstrStatus = ‘CLS’  then Copy  else ‘N/A’ | cessation\_reason\_code |  | Unknown |
| ref\_rfrNZ\_Filing.fstrRegularFrequency app\_tblAccount.fstrFiling | Lookup ref\_rfrNZ\_Filing on fstrFiling  if fstrRegularFrequency is returned  then initcap(fstrRegularFrequency)  NB check return level - Tony | filing\_frequency |  | Unknown |
| app\_tblAccount.fstrFiling | Copy | filing\_frequency\_code |  | Unknown |
| app\_tblAccount.fstrAccountType app\_tblNZ\_AccIncome.fstrExemptionReason [accinc]  app\_tblNZ\_AccIPE.fstrExemptionReason [accipe]  ref\_lanListItem.fstrdecode1 [accipe]  ref\_lannz\_inctaxexemptionrsn.fstrdecode1 [incxmp]  app\_tblindicator\_exempt.fstrIndicator  ref\_lanindicator.fstrdecode2 | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  if fstrAccountType = ‘IPE’  and [accipe].fstrExemptionReason is not null  then lookup ref\_lanListItem   on fstrfield = 'NZ.ACCIPE.ExemptionReason'  and fstrlistitem = [accipe].fstrExemptionReason   if [accipe].fstrdecode1 is returned  then fstrdecode1   else ‘Unknown’  else if fstrAccountType in (‘ITN’,’IIT’)  and [accinc].fstrExemptionReason is not null  then lookup ref\_lannz\_inctaxexemptionrsn   on [accinc].fstrExemptionReason   if [incxmp].fstrdecode1 is returned  then fstrdecode1  else ‘Unknown’  else lookup ref\_lanindicator on fstrIndicator  if fstrdecode2 is returned  then cast(fstrdecode2 as varchar(30))  else ‘Unknown’  else ‘Unknown’ | exemption\_reason |  | Unknown |
| app\_tblAccount.fstrAccountType app\_tblNZ\_AccIncome.fstrExemptionReason [accinc]  app\_tblNZ\_AccIPE.fstrExemptionReason [accipe]  app\_tblindicator\_exempt.fstrIndicator | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  if fstrAccountType = ‘IPE’   and [accipe].fstrExemptionReason is not null  then [accipe].fstrExemptionReason   else if fstrAccountType in (‘ITN’,’IIT’)   and [accinc].fstrExemptionReason is not null  then [accinc].fstrExemptionReason else fstrIndicator else ‘Unknown’ | exemption\_reason\_code |  | Unknown |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  then ‘Y’  else ‘N’ | exempted\_flag |  |  |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator = ‘GRNLST’   then ‘Y’  else ‘N’ | green\_listed\_flag |  |  |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator = ‘FRDRED’   then ‘Y’  else ‘N’ | red\_listed\_flag |  |  |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator = ‘ACCHLT’   then ‘Y’  else ‘N’ | account\_halt\_flag |  |  |
| ref\_lanindicator.fstrdecode2 | if fstrIndicator in ('AUDIT', 'AUDPRA', 'AUDSFP')  then lookup ref\_lanindicator on fstrIndicator  if fstrdecode2 is returned  then cast(fstrdecode2 as varchar(30)) | audit\_case\_active\_desc |  | Unknown |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator in ('AUDIT', 'AUDPRA', 'AUDSFP')  then fstrIndicator | audit\_case\_active\_code |  | Unknown |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator in ('AUDIT', 'AUDPRA', 'AUDSFP')  then ‘Y’  else ‘N’ | audit\_case\_active\_flag |  |  |
| ref\_lanindicator.fstrdecode2 | if fstrIndicator in ('INVADD', 'STPMAL')  then lookup ref\_lanindicator on fstrIndicator  if fstrdecode2 is returned  then cast(fstrdecode2 as varchar(30)) | ~~stop\_mail\_reason~~  stop\_mail\_desc |  | Unknown |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator in ('INVADD', 'STPMAL')  then fstrIndicator else ‘N’ | stop\_mail\_code |  | Unknown |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator in ('INVADD', 'STPMAL')  then ‘Y’  else ‘N’ | stop\_mail\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### Start Account PAY

Populate only for PAY Accounts i.e. app\_tblAccount.fstrAccountType = ‘PSO’.

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_pay\_sk | SK |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_key |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ird\_number |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_account\_id |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | heritage\_location\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | legal\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | trading\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| app\_tblindicator\_exempt.fstrIndicator  ref\_lanindicator.fstrdecode2 | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  thenlookup ref\_lanindicator on fstrIndicator  if fstrdecode2 is returned  then cast(fstrdecode2 as varchar(30))  else ‘Unknown’ else ‘Unknown’ | exemption\_reason |  |  |
| app\_tblindicator\_exempt.fstrIndicator | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  then fstrIndicator  else ‘Unknown’ | exemption\_reason\_code |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | exempted\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | green\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | red\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_halt\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_code |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_flag |  |  |
| app\_tblNZ\_AccPSO.fblnIR56 | if fblnIR56 = 1  then ‘Y’  else ‘N’ | ir56\_flag |  |  |
| ref\_lanNZ\_EmployerType.fstrdecode2  app\_tblNZ\_AccPSO.fstrfilingoption | Lookup ref\_lanNZ\_EmployerType   on fstrEmployerType = fstrFilingOption if fstrdecode2 is returned  then fstrdecode2 | filing\_option |  | Unknown |
| app\_tblNZ\_AccPSO.fstrfilingoption | Copy | filing\_option\_code |  | Unknown |
| app\_tblNZ\_AccPSO.fblnpaydayfiler | if fblnpaydayfiler = 1  then ‘Y’  else ‘N’ | payday\_filer\_flag |  |  |
| app\_tblNZ\_AccPSO.fdtmpaydayfilerstart | if $$is\_date\_valid$$ (fdtmpaydayfilerstart)   then fdtmpaydayfilerstart  else $$HDT | payday\_filer\_start\_date |  | $$HDT |
| app\_tblNZ\_AccPSO.fblnCSE | if fblnCSE = 1  then ‘Y’  else ‘N’ | employer\_child\_support\_flag |  |  |
| app\_tblNZ\_AccPSO.fblnSLE | if fblnSLE = 1  then ‘Y’  else ‘N’ | employer\_student\_loan\_flag |  |  |
| app\_tblNZ\_AccPSO.fblnKSE | if fblnKSE = 1  then ‘Y’  else ‘N’ | kiwisaver\_employee\_dedn\_flag |  |  |
| app\_tblNZ\_AccPSO.fblnKSR | if fblnKSR = 1  then ‘Y’  else ‘N’ | kiwisaver\_employer\_cntrb\_flag |  |  |
| app\_tblNZ\_AccPSO.fblnSSC | if fblnSSC = 1   then ‘Y’  else ‘N’ | ESCT\_flag |  |  |
| ref\_lanNZ\_EmployerGroup.fstrDecode2  app\_tblNZ\_AccPSO.fstrEmployerGroup | Lookup ref\_lanNZ\_EmployerGroup on fstrEmployerGroup if fstrDecode2 is returned   then fstrDecode2 | employer\_group |  | Unknown |
| app\_tblNZ\_AccPSO.fstrEmployerGroup | Copy | employer\_group\_code |  | Unknown |
| app\_tblindicator\_app.fstrIndicator | if fstrIndicator = 'PDEXMP'  then ‘Y’  else ‘N’ | payday\_variation\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### Start Account PIE

Populate only for PIE Accounts i.e. app\_tblAccount.fstrAccountType = ‘PIE’.

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_pie\_sk | SK |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_key |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ird\_number |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_account\_id |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | heritage\_location\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | legal\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | trading\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| app\_tblindicator\_exempt.fstrIndicator  ref\_lanindicator.fstrdecode2 | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  thenlookup ref\_lanindicator on fstrIndicator  if fstrdecode2 is returned  then cast(fstrdecode2 as varchar(30))  else ‘Unknown’ else ‘Unknown’ | exemption\_reason |  |  |
| app\_tblindicator\_exempt.fstrIndicator | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  then fstrIndicator  else ‘Unknown’ | exemption\_reason\_code |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | exempted\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | green\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | red\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_halt\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_code |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_flag |  |  |
| ref\_lanlistitem.fstrdecode1  app\_tblNZ\_AccPIEInfo.fstrPIEType | Lookup ref\_lanlistitem on fstrfield = 'NZ.ACCPIE.PIEType'  and fstrlistitem = fstrPIEType if fstrdecode1 is returned  then fstrdecode1 | pie\_type |  | Unknown |
| app\_tblNZ\_AccPIEInfo.fstrPIEType | Copy | pie\_type\_code |  | Unknown |
| ref\_lanlistitem.fstrdecode1  app\_tblNZ\_AccPIEInfo.fstrFilingOption | Lookup ref\_lanlistitem on fstrfield = 'NZ.ACCPIE.FilingOption’  and fstrlistitem = fstrFilingOption if fstrdecode1 is returned  then fstrdecode1 | filing\_option |  | Unknown |
| app\_tblNZ\_AccPIEInfo.fstrFilingOption | Copy | filing\_option\_code |  | Unknown |
| app\_tblNZ\_AccPIEInfo.fstrattributionPeriod  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PIE’  and source\_code\_name = ‘INCOME\_ALLOCATION\_PERIOD’  and **dip\_code\_value** = fstrattributionPeriod if dip\_description is returned  then dip\_description else ‘Unknown’  NB. *Use the FIRST mappings to provide a meaningful description* | attribution\_period |  | Unknown |
| app\_tblNZ\_AccPIEInfo.fstrFDRCalcPeriod  mp\_source\_code.dip\_description (see Section 5.1) | Lookup mp\_source\_code  on source\_system = ‘FIRST’  and source\_tax\_type = ‘PIE’  and source\_code\_name = ‘FDR\_CALCULATION\_PEROD’  and **dip\_code\_value** = fstrFDRCalcPeriod if dip\_description is returned  then dip\_description else ‘Unknown’  NB. *Use the FIRST mappings to provide a meaningful description* | fdr\_calc\_period |  | Unknown |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### Start Account INC

Populate only for INC Accounts i.e. app\_tblAccount.fstrAccountType in (‘IIT’,’ITN’).

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_inc\_sk | SK |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_key |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ird\_number |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_account\_id |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | heritage\_location\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | legal\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | trading\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| app\_tblNZ\_AccIncome.fstrExemptionReason  ref\_lannz\_inctaxexemptionrsn.fstrdecode1  app\_tblindicator\_exempt.fstrIndicator  ref\_lanindicator.fstrdecode2 | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  if fstrExemptionReason is not null  then lookup ref\_lannz\_inctaxexemptionrsn on   fstrExemptionReason   if fstrdecode1 is returned  then fstrdecode1  else ‘Unknown’  else lookup ref\_lanindicator on fstrIndicator  if fstrdecode2 is returned  then cast(fstrdecode2 as varchar(30))  else ‘Unknown’ else ‘Unknown’ | exemption\_reason |  | Unknown |
| app\_tblNZ\_AccIncome.fstrExemptionReason  app\_tblindicator\_exempt.fstrIndicator | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  if fstrExemptionReason is not null   Copy  else fstrIndicator else ‘Unknown’ | exemption\_reason\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | exempted\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | green\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | red\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_halt\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_code |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_flag |  |  |
| app\_tblNZ\_AccIncome.fdtmBalanceDate | CAST(CONCAT(MONTHNAME(accinc.fdtmBalanceDate), ' ',CAST(DAY(accinc.fdtmBalanceDate) AS VARCHAR(2))) AS VARCHAR(30)) [e.g. November 10] | balance\_date |  |  |
| app\_tblIndicator\_acnt.fstrIndicator | if fstrIndicator = 'ZROTTR'  then ‘Y’  else ‘N’ | zero\_tailored\_rate\_flag |  |  |
| app\_tblNZ\_AccITN.fblnTransTasman | if fblnTransTasman = 1  then ‘Y’  else ‘N’ | trans\_tasman\_flag |  |  |
| app\_tblNZ\_AccITN.fblnAgentNonResInsurer | if fblnAgentNonResInsurer = 1  then ‘Y’  else ‘N’ | agent\_non\_res\_insurer\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

### Start Account GST

Populate only for GST Accounts i.e. app\_tblAccount.fstrAccountType = ‘GST’.

See DIP - [Dimension Standards](https://teams.microsoft.com/l/file/C43DC83E-9B7D-4863-99A6-A1E2A5CFA3E1?tenantId=fb39e3e9-23a9-404e-93a2-b42a87d94f35&fileType=docx&objectUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam%2FShared%20Documents%2FRefined%20Layer%20Design%20and%20Build%2F9.%20%20Standards%2FDIP_Dimension_Standards.docx&baseUrl=https%3A%2F%2Firnz.sharepoint.com%2Fsites%2FDataPlatformTeam&serviceName=teams&threadId=19:c9ee9980de684e74badc5fe1d414bc56@thread.skype&groupId=7100c035-72cd-4625-aa78-807e71980487) for standard rows for Unknown and N/A DIP Ids.

| **Source Table/Column** | **Transform Rule/Logic** | **Target Column** | **Key** | **Default value for Null** |
| --- | --- | --- | --- | --- |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_gst\_sk | SK |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | dip\_account\_id | UK |  |
|  | *Standard processing field* | record\_effective\_timestamp | UK |  |
|  | *Standard processing field* | record\_expiry\_timestamp |  |  |
|  | *Standard processing field* | record\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_key |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ird\_number |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_account\_id |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | start\_parent\_profile\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | heritage\_location\_number |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | legal\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | trading\_name |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_type\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_start\_date |  | $$LDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_ceased\_date |  | $$HDT |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_status\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | security\_level\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | cessation\_reason\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | filing\_frequency\_code |  | Unknown |
| app\_tblindicator\_exempt.fstrIndicator  ref\_lanindicator.fstrdecode2 | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  thenlookup ref\_lanindicator on fstrIndicator  if fstrdecode2 is returned  then cast(fstrdecode2 as varchar(30))  else ‘Unknown’ else ‘Unknown’ | exemption\_reason |  |  |
| app\_tblindicator\_exempt.fstrIndicator | if fstrIndicator in ('CRTEXM’, ’SLSRDE’, ’RWTEXM’, ’RREXM’, ’FTXEXM’, ’ESSEXM’, ’ELCEXM')  then fstrIndicator  else ‘Unknown’ | exemption\_reason\_code |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | exempted\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | green\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | red\_listed\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | account\_halt\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_code |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | audit\_case\_active\_flag |  |  |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_desc |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_code |  | Unknown |
| *See Start ACCOUNT for details* | *See Start ACCOUNT for details* | stop\_mail\_flag |  |  |
| app\_tblnz\_accgstinfo.fstrLastAccountingBasis | if fstrLastAccountingBasis = ‘HYB’  then ‘Hybrid’ else if fstrLastAccountingBasis = ‘INV’  then ‘Invoice’ else if fstrLastAccountingBasis = ‘PYM’  then ‘Payments’ | accounting\_basis |  | Unknown |
| app\_tblnz\_accgstinfo.fstrRegCircumstance  ref\_lanlistitem.fstrdecode1 | Lookup ref\_lanlistitem   on fstrField = ‘NZ.ACCGST.RegCircumstance’   and fstrlistitem = fstrRegCircumstance if fstrdecode1 is returned  then fstrdecode1 | registration\_circumstance |  | Unknown |
| app\_tblnz\_accgstinfo.fblnPricesIncGST | if fblnPricesIncGST = 1  then ‘Y’  else ‘N’ | included\_in\_prices\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnExemptSupplies | if fblnExemptSupplies = 1  then ‘Y’  else ‘N’ | exempt\_supplies\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnExporter | if fblnExporter = 1  then ‘Y’  else ‘N’ | exporter\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnImporter | if fblnImporter = 1  then ‘Y’  else ‘N’ | importer\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnUnpolicedFiler | if fblnUnpolicedFiler = 1  then ‘Y’  else ‘N’ | unpoliced\_filer\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnSelfInvoicing | if fblnSelfInvoicing = 1  then ‘Y’  else ‘N’ | self\_invoicer\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnHospiceFiler | if fblnHospiceFiler = 1  then ‘Y’  else ‘N’ | hospice\_filer\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnHospiceFiler app\_tblnz\_accgstinfo.fstrHospiceType | if fblnHospiceFiler = 1  if fstrHospiceType is populated  then copy  else ‘Unknown’ else ‘Unknown’ | hospice\_type |  | Unknown |
| app\_tblnz\_accgstinfo.fblnHospiceFiler ref\_lanListItem.fstrDecode1  app\_tblnz\_accgstinfo.fstrHospiceType | if fblnHospiceFiler = 1  lookup ref\_lanListItem   on fstrfield = 'NZ.RTNGST.Hospice'  and fstrlistitem = fstrHospiceType  if fstrDecode1 is returned  then fstrDecode1  else ‘Unknown’ else ‘Unknown’ | hospice\_type\_desc |  | Unknown |
| app\_tblnz\_accgstinfo.fblnGSTOnElectronicServices | if fblnGSTOnElectronicServices = 1  then ‘Y’  else ‘N’ | gst\_on\_remote\_srvcs\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnGSTOnElectronicServices ref\_lanlistitem.fstrListItem  ref\_lanlistitem.fstrField  ref\_lanListItem.fstrDecode1  app\_tblnz\_accgstinfo.fstrGORSType | if fblnGSTOnElectronicServices = 1  lookup ref\_lanlistitem   on fstrListItem = fstrGORSType  and fstrField = 'NZ.ACCGST.GORSType'  if fstrDecode1 is returned  then fstrDecode1  else ‘Unknown’ else‘Unknown’  Scott: this is being cast to varchar(10) instead of varchar(30) | gst\_on\_remote\_srvcs\_type |  |  |
| app\_tblnz\_accgstinfo.fblnGSTOnElectronicServices app\_tblnz\_accgstinfo.fstrGORSType | if fblnGSTOnElectronicServices = 1  if fstrGORSType is populated  then copy  else ‘Unknown’ else ‘Unknown’ | gst\_on\_remote\_srvcs\_code |  | Unknown |
| app\_tblnz\_accgstinfo.fblnZeroRatedFinancialServices | if fblnZeroRatedFinancialServices = 1  then ‘Y’  else ‘N’ | gst\_zero\_rated\_fin\_srv\_flag |  |  |
| app\_tblnz\_accgstinfo.fblnNonResBusinessClaimant | if fblnNonResBusinessClaimant = 1  then ‘Y’  else ‘N’ | gst\_non\_resident\_claimant\_flag |  |  |
| app\_tblNZ\_GSTreg.fblnTaxableActivity | if fblnTaxableActivity = 1  then ‘Y’  else ‘N’ | taxable\_activity\_flag |  |  |
|  | *Standard processing field* | insert\_object\_run\_key |  |  |
|  | *Standard processing field* | update\_object\_run\_key |  |  |

# Appendix A – Acronyms, Abbreviations and Key Terms

Acronyms

| **Acronym** | **Long Name** |
| --- | --- |
| START | IR’s new OLTP system |
| TDW | Transitional Data Warehouse |
| EDW | Enterprise Data Warehouse, based on FIRST data |
| DIP | Data & Intelligence Platform, ultimately replacing EDW & TDW |
| FIRST | IR’s old OLTP system |

Abbreviations

| **Abbreviation** | **Meaning** | **Definition** |
| --- | --- | --- |
| $$HWM | High-water mark | This is the timestamp of the point in time to which a DIP table has been populated. Just as a dam has a high-water mark which measure how much water it contains; each DIP table has a high-water mark that shows timestamp of the last record inserted in chronological order. |
| $$LDT | System low date | 01-Jan-1800 |
| $$HDT | System high date | 31-Dec-9000 |
| $$EDWTS1 | EDW Cut-off timestamp for Parties and GST (R1) accounts | 05-Feb-2017  This is the cut-off timestamp between EDW and Start used for the initial load of R1 accounts (GST accounts for Goods and Services Tax were populated on 3-4 Feb-2017 by user CNVR1) |
| $$EDWTS2 | EDW Cut-off timestamp for R2 accounts (e.g. PSO/PIE) | 14-Apr-2018  This is cut-off timestamp between EDW and Start used for the initial load of R2 accounts (PSO accounts for Portfolio Investment Entity were populated on 13-Apr-2018 by user CNVR2) |
| $$EDWTS3 | EDW Cut-off timestamp for R3 accounts (e.g. IIT/ITN) | 21-Apr-2019  This is the cut-off timestamp between EDW and Start used for the initial load of R3 accounts (e.g.IIT/ITN accounts for Income Tax Individuals/non-individuals were populated on 20-Apr-2019 by user CNVR3) |
| $$TSDIFF | The difference between contiguous timestamps | Logically contiguous timestamp ranges start/end at the same point in time. However, for design reasons these timestamps will be stored with a difference measured by $$TSDIFF e.g. 1 minute ~~second~~ (specified in design). |

Key Terms

| **Term** | **Definition** |
| --- | --- |
| Production quality data | An instance or subset of production data, or, an offline version of production data that is representative of the quality levels of current production data. The data must be sufficiently representative that it can be effectively used for data profiling or quality assessment. |
| Master Data | Any information that is relatively static and is considered to play a key role in the core operation of the business (i.e. reference data); includes data about customers, products, employees, partners etc. Master data is typically shared by multiple users and groups across the organisation. |
| Dimensional or Star Schema | Dimensional data modelling standard is most often used in data warehousing systems. This is different from the Relational or 3rd normal form, commonly used for transactional (OLTP) type systems. Dimensional modelling uses denormalisation steps to improve performance, which introduces redundancy of data |
| Subject Area | A logical grouping of Entities relating to a sub domain of interest within a data model. A subject area allows focus attention on key entities of interest. |
| Staging Layer | The table structures in staging layer reflect the table or file structure of the source extract without any transformation applied. |

# Appendix B – EDW-Start Cut-off

Start accounts were migrated by releases R1, R2 and R3. For each release an EDW-Start Cut-off date is specified to avoid intra-day gaps between contiguous EDW and Start data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Start Release** | **Conversion User** | **Conversion Dates** | **Cut-Off Placeholder** | **Cut-Off Date** |
| R1 | cnvr1 | 3-Feb-2017 to 4-Feb-2017 | $$EDWTS1 | 5-Feb-2017 |
| R2 | cnvr2 | 13-Apr-2018 | $$EDWTS2 | 14-Apr-2018 |
| R3 | cnvr3 | 20-Apr-2019 | $$EDWTS3 | 21-Apr-2019 |

Because the Accounts table (i.e. app\_tblAccount) joins to all other tables in this sub-domain, only filtering criteria based on EDW-Start Cut-off date need to be specified for the Accounts table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Account Types which exist in EDW** | $$LDT | R1 | R2 | R3 | SSHDT |
| R1 Account Types |  | EDW | Start | Start | Start |
| R2 Account Types |  | EDW | EDW | Start | Start |
| R3 Account Types |  | EDW | EDW | EDW | Start |

Account (Tax) Types by START Release

| **EDW** | | **Start** | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tax Type** | **Description** | **Account Type** | **Description** | **Comments** | **R1** | **R2** | **R3** |
| ACC | Emp r ACC Levy | n/a | n/a | Deprecated |  |  |  |
| AIL | Ap Issuer Levy | AIL | Approved Issuer Levy |  |  | Y |  |
| CAR | CLEARING ACCT | n/a | n/a | Not in Start (pseudo a/c) |  |  |  |
| CPR | Cust Parent | CPR | Receiving Carer |  |  |  | Y |
| CSE | CSA Employer | PSO | Employment Activities | Consolidated into PSO |  | Y |  |
| DWT | Ord Div PAYE | DWT | Dividend Withholding Tax |  |  | Y |  |
| FAM | Family Support / (WFFTC) | FAM | Working for Families |  |  |  | Y |
| FBA | FBT Ann Liable | FBT | Fringe Benefit Tax |  |  | Y |  |
| FBI | FBT Inc Yr | FBT | Fringe Benefit Tax |  |  | Y |  |
| FBT | Fringe Benefit | FBT | Fringe Benefit Tax |  |  | Y |  |
| GMD | Gaming Duty | GMD | Gaming Machine Duty |  |  | Y |  |
| GST | Goods&Services | GST | Goods and Services Tax |  | Y |  |  |
| ICA | Imp Credit A/C | IIT | Income Tax Individuals | For individuals only |  |  | Y |
|  |  | ITN | Income Tax Non-Individuals | For organisations only |  |  | Y |
| INC | INCOME TAX | IIT | Income Tax Individuals | For individuals only |  |  | Y |
| ITN | Income Tax Non-Individuals | For organisations only |  |  | Y |
| IPE | IP Exempt Recp | IPE | Withholding Disbursements Account |  |  | Y |  |
| IPS | Interest PAYE | IPS | RWT on Interest |  |  | Y |  |
| KSE | EMPLOYEE DED | KSF | KiwiSaver Scheme |  |  |  |  |
| KSR | EMPLOYER CONT | KSF | KiwiSaver Scheme |  |  |  |  |
| KSS | MEMBER ACCOUNT | KSS | KiwiSaver Member |  |  |  | Y |
| MAC | MA CREDIT A/C | IIT | Income Tax Individuals | For individuals only |  |  | Y |
|  |  | ITN | Income Tax Non-Individuals | For organisations only |  |  | Y |
| NCP | Non-Cust Parnt | NCP | Liable Parent |  |  |  | Y |
| NRT | Non Res W/Tax | NRT | Non-Resident Withholding Tax |  |  | Y |  |
| PAY | Tax Deductions | PSO | Employment Activities |  |  | Y |  |
| PIE | PIE TAX | PIE | Portfolio Investment Entity |  |  | Y |  |
| PPL | PdParental Lve | PPL | Paid Parental Leave |  |  |  | Y |
| QCT | QC Elect Tax | IIT | Income Tax Individuals | For individuals only |  |  | Y |
|  |  | ITN | Income Tax Non-Individuals | For organisations only |  |  | Y |
| REB | Rebate Claim | REB | Donation Tax Credit |  |  |  | Y |
| RLT | RES LAND W/H | RLT | Residential Land Withholding Tax |  |  | Y |  |
| RWT | Res W/H (Spec) | RWT | Resident Withholding Tax |  |  | Y |  |
| SEA | Self Empl ACC | n/a | n/a | Deprecated |  |  |  |
| SHR | S/Hold Emp ACC | n/a | n/a | Not in Start |  |  |  |
| SLE | Repayment Ded | PSO | Employment Activities | Consolidated into PSO |  | Y |  |
| SLS | Student Loan | SLS | Student Loans |  |  |  | Y |
| SSC | Spec Super W/H | PSO | Employment Activities | Consolidated into PSO |  | Y |  |
| WPE | W/H Pay Elect | PSO | Employment Activities | Consolidated into PSO |  | Y |  |
| WPN | W/H Pay. No.El | PSO | Employment Activities | Consolidated into PSO |  | Y |  |

Notes:

1. Because there must only be one instance of an Account (Tax) Type at a location for a Party, multiple EDW tax types must not be mapped to a Start account type. In other words, though data conversion has merged CSE, SSC, SLE, WPE, WPN and PAY Tax Types from EDW into PSO account types in Start, it is not possible to ‘merge’ these tax types over time. Therefore, all ‘secondary’ EDW tax types must retain their original code value and description.
2. Tax types KSE and KSR are expected to be migrated to Start in R4.
3. Tax Types ACC, CAR, SEA and SHR have been deprecated and will not be in Start. Therefore, there is no need for a cut-off date for these.

START R1 Account Types

Conversion (cnvr1) on 3-Feb-2017 to 4-Feb-2017. Cut-off on 5-Feb-2017. The following counts are of the records created by the corresonding combination of user and date.

| **Account Type** | **Account Type Description** | **cnvr1 03-Feb** | **cnvr1 04-Feb** | **batch 07-Feb** | **batch 08-Feb** | **batch 09-Feb** |
| --- | --- | --- | --- | --- | --- | --- |
| GST | Goods and Services Tax | 523,059 | 359,319 | 42 | 460 | 156 |

START R2 Account Types

Conversion (cnvr2) on 13-Apr-2018. Cut-off on 14-Apr-2018 ~~2019~~. The following counts are of the records created by the corresonding combination of user and date.

| **Account Type** | **Account Type Description** | **cnvr2 13-Apr** | **batch 14-Apr** | **batch 15-Apr** | **batch 16-Apr** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| AIL | Approved Issuer Levy | 3,025 |  |  |  |  |
| CRS | Common Reporting Standard |  |  |  | 1 |  |
| DWT | Dividend Withholding Tax | 614,531 | 1 |  | 421 |  |
| FAT | Foreign Account Tax Compliance Act | 139 |  |  |  |  |
| FBT | Fringe Benefit Tax | 39,165 |  |  |  |  |
| GMD | Gaming Machine Duty | 656 |  |  |  |  |
| IPE | Withholding Disbursements Account | 95,300 |  |  |  |  |
| IPS | RWT on Interest | 40,984 |  |  |  |  |
| NRT | Non-Resident Withholding Tax | 7,598 |  |  |  |  |
| PIE | Portfolio Investment Entity | 811 |  |  |  |  |
| PSO | Employment Activities (Payroll) | 266,903 | 7 | 10 | 237 |  |
| RLT | Residential Land Withholding Tax | 159 |  |  |  |  |
| RWT | Resident Withholding Tax | 3,344 |  |  | 1 |  |

START R3 Account Types

Conversion (cnvr3) on 20-Apr-2019. Cut-off on 21-Apr-2019. The following counts are of the records created by the corresonding combination of user and date.

| **Account Type** | **Account Type Description** | **cnv3 20-Apr** | **batch 22-Apr** | **batch 23-Apr** | **batch 24-Apr** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| CPR | Receiving Carer | 657,264 |  | 2 |  |  |
| DSB | Client Disbursements |  | 176 |  |  |  |
| FAM | Working for Families Tax Credits | 779,581 |  |  | 31 |  |
| IIT | Income Tax Individuals | 7,305,937 |  |  | 792 |  |
| ITN | Income Tax Non-Individuals | 1,508,247 |  |  |  |  |
| KSS | KiwiSaver Member | 5,894,946 |  |  | 8 |  |
| NCP | Liable Parent | 553,095 |  |  |  |  |
| PPL | Paid Parental Leave | 315,302 |  |  | 109 |  |
| RDI | R&D Tax Incentive |  |  |  |  | First on 15-Jul-2019 |
| REB | Donation Tax Credit | 1,308,501 | 88 | 1 | 126 |  |
| SLS | Student Loans | 1,366,217 |  |  |  |  |
| TPA | Tax Pooling Account | 5 |  |  |  |  |

START Tables across Multiple Releases

The following counts are of the records created by the corresonding combination of user and table.

| **Start Table** | **Type** | **cnvr1** | **cnvr2** | **cnvr3** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| tblAccount |  | 882,378 | 1,072,615 | 19,689,095 | Same as for tblId |
| tblAccountTransfer |  |  |  | 1,635 |  |
| tblCollectPaymentPlan |  | 14,605 |  | 66,300 |  |
| tblCollectPaymentPlanBillitem |  | 732 | 186 | 234,505 |  |
| tblCustomer |  | 8,549,254 |  | 302,050 |  |
| tblid | ACCIRD | 882,378 | 1,072,615 | 19,689,095 | Same as for tblAccount |
| tblIndicator (Account) | ACCHLT | 1317 | 64 | 7,655 |  |
| CRTEXM |  |  | 31,743 |  |
| INVADD |  | 36 | 33,865 |  |
| SLSRDE |  |  | 1,007 |  |
| STPMAL | 22 |  |  |  |
| ZROTTR |  |  | 4,688 |  |
| tblIndicator (Customer) | FRDRED | 226 |  |  |  |
| INVADD |  |  | 95,411 |  |
| PAYPLN | 9,852 |  | 32,481 |  |
| STPMAL | 947,516 |  |  |  |
| tblProfile |  | 9,494,584 | 1,058,041 | 19,993,610 |  |
| tblNameProfile |  | 9,072,518 |  | 304,541 |  |
| tblNameRecord (Account) | DBA | 12,683 |  |  |  |
| LGL |  |  | 304,079 |  |
| tblNameRecord (Customer) | DBACST | 383,455 |  |  |  |
| LGL | 4,585,347 |  |  |  |
| tblNZ\_AccGSTInfo |  | 607,175 |  |  |  |
| tblNZ\_AccIncome |  |  |  | 8,028,838 |  |
| tblNZ\_AccIPE |  |  | 91,751 |  |  |
| tblNZ\_AccITN |  |  |  | 1,056,002 |  |
| tblNZ\_AccPieInfo |  |  | 345 |  |  |
| tblNZ\_AccPSO |  |  |  |  | 336,451 by batch |
| tblNZ\_AccountStd |  | 607,175 | 600,355 | 17,877,777 |  |
| tblNZ\_GSTReg |  |  |  |  | 2,666 by batch |

# Appendix C – Common Natural Key for DIP Accounts

| **Candidates for DIP\_ACCOUNT\_ID** | **Comments** |
| --- | --- |
| ACCIRD | Not suitable for the following reasons:   1. ACCIRD may be created after Start Account (e.g. flngaccountkey = 1620689408; fdtmWhen = 14-Apr-2018 20:05:04 for tblAccount but fdtmWhen = 07-Nov-2018 18:21:28 for ACCIRD). There are many examples for online users with several minutes difference (e.g. flngaccountkey 90346816 (FAM), 848560640 (GST), 1851236864 (GST)). 2. A Start Account may have several ACCIRDs (e.g. flngaccountkey = 397014016) |
| flngAccountKey | Does not exist in EDW |
| Other | Solution: populate DIP\_ACCOUNT\_ID as follows:   * When sourced from Start: concatenate(‘DIP-A-’, flngaccountkey) * When sourced from EDW:   if the combination of (ird\_number + location\_number + tax\_type) from EDW   exists in Start (at cut-off date/time)  then use DIP\_ACCOUNT\_ID (from Start)  else concatenate(‘DIP-’, <tax\_type>, ‘-‘, <ird\_number>, ’-‘, <location\_number>)   NB. *Because the same ird\_number may have multiple locations for a Tax Type, and/or multiple Tax Types at a location, the combination of (ird\_number + location\_number + tax\_type) is required.* |

# Appendix D – Standard De-Duplication Process

The purpose of step 2 is to remove duplicates over time so that the resulting rows for each natural key are contiguous and non-overlapping in time. The default de-duplication process is to be following unless explicitly stated otherwise.

**De-duplication Process where Business Dates Exist**

We use a slightly different process where the data has business start and end dates which are often completely independent of the RecordEffective and RecordExpiry dates. The former shows a history of when something was valid, and the latter are showing what the source system looked like at a point in time.

We can take both timelines into account when looking to resolve duplicates, by creating some derived columns during processing and then using those columns in our tie-breaker logic.

For each record we create the following derived columns:

| **Derived Column** | **Definition** |
| --- | --- |
| **timeslice\_from** | Business\_Start\_Date or Record\_Effective\_Timestamp |
| **timeslice\_to** | Business\_End\_Date or Record\_Expiry\_Timestamp (if the Business\_End\_Date is null, substitute with 31-12-9000) |
| **~~number\_of\_valid\_days~~**  **number\_of\_valid\_seconds** | Min(Business\_End\_Date or Record\_Expiry\_Timestamp)  – Max(Business\_Start\_Date or Record\_Effective\_Timestamp) |

The derived column number\_of\_valid\_seconds~~Days~~ can then be used as a tie-breaker field in determining the “correct” record for each timeslice.

**Default De-duplication Process**

De-duplicate each source table over its ***Natural Key*** as follows:

1. Establish a single timeline from all unique ***business active timestamps***. Gaps may occur but there must be no overlaps.
2. Determine all time-periods between two contiguous timestamps on the timeline from (1) regardless of whether they are ‘start’ or ‘end’ timestamps
3. For each time-period from (2):
   1. Determine the record(s) which ‘exist’ during this time-period
   2. Identify the record in (3a) with the highest priority using the specified ***Tie-breakers***
4. Concatenate the ‘priority’ records from 3b. Gaps may occur but there must be no overlaps. The following ‘compress’ step with ‘merge’ contiguous duplicate values.

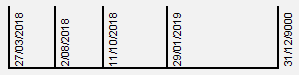
**Example**

This example uses the following source data.



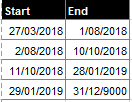
Step 2.1: Establish a single timeline

Establish a single timeline from all unique ***business active timestamps***. Gaps may occur but there must be no overlaps.



Step 2.2: Determine all time-periods

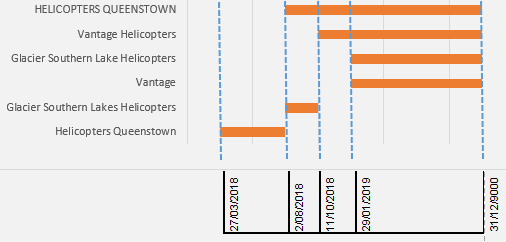
Determine all time-periods between two contiguous timestamps on the timeline from Step 2.1 regardless of whether they are ‘start’ or ‘end’ timestamps.



Step 2.3: Identify highest-priority records

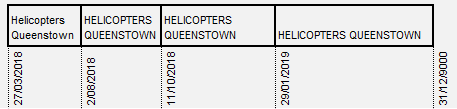
For each time-period from Step 2.2:

1. Determine the record(s) which ‘exist’ during this time-period.
   1. For 27/03/2018-02/08/2018, there is only one record.
   2. For 02/08/2018-11/10/2018, there are 2 records.
   3. For 11/10/2018-29/01/2019, there are also 2 records. Again, the top-most record has the highest priority.
   4. For 29/01/2019-31/12/9000, there are 4 records and the top-most record has the highest priority.
2. Identify the record in Step2.3a with the highest priority using the specified ***Tie-breakers***
   1. For 27/03/2018-02/08/2018, there is only one record, therefore, it is automatically the highest priority.
   2. For 02/08/2018-11/10/2018, the top-most record has the highest priority.
   3. For 11/10/2018-29/01/2019, the top-most record has the highest priority.
   4. For 29/01/2019-31/12/9000, the top-most record has the highest priority.



Step 2.4: Concatenate

Concatenate the ‘priority’ records from 3b. Gaps may occur but there must be no overlaps.



Step 3: Compress

After de-duplicating records in Step 2, one or more contiguous records sharing the same Natural Key may have the same values for all columns used for comparison. This step will ‘compress’ (i.e. merge) any such contiguous records using the minimum Record\_Effective\_Date and maximum Record\_Expiry\_Date for the resulting timeline.

In this example, records 2-4 from Step 2.4 are merged into record 5. Record 1 remains unchanged.



# Appendix E – Example SQL for Context – EDW

The ‘Select’ column names in these SQL statements will be referred to in the “Source Table/Column” fields of the transformation rules and are intended to give context to those rules as well as a guide on how to merge all the source objects into the final Account Type data streams.

### EDW Party Account

select

CAST(IFNULL(party.dip\_party\_id, CONCAT('DIP-I-', LPAD(CAST(tr.ird\_number AS VARCHAR), 9, '0'))) AS VARCHAR(30)) as dip\_party\_id,

CAST(IFNULL(CAST(account.dip\_account\_id AS VARCHAR(20)),

CONCAT('DIP-', LPAD(CAST(tr.ird\_number AS VARCHAR(10)), 9, '0'), '-',

tr.tax\_type, LPAD(CAST(tr.location\_number AS VARCHAR(3)), 3, '0'))) AS VARCHAR(30)) as dip\_account\_id,

tr.record\_effective\_timestamp,

tr.record\_expiry\_timestamp,

tr.record\_active\_flag,

CAST('OWNER' AS VARCHAR(10)) as relationship\_type,

tr.treg\_date\_start as relationship\_effective\_from,

IFNULL(tr.treg\_date\_end, '9999-12-31') as relationship\_effective\_to,

CAST('FIRST' AS VARCHAR(10)) as relationship\_origin,

tr.insert\_object\_run\_key,

tr.update\_object\_run\_key

from

crw\_ae\_tax\_registrations as tr left join

(

select distinct

account.dip\_account\_id,

account.ird\_no,

account.edw\_tax\_type,

account.edw\_location\_number

from

rfn\_account\_sk\_map as account

where

source = 'EDW'

) as account

on

(

tr.ird\_number = IFNULL(account.ird\_no, -1)

AND tr.tax\_type = IFNULL(account.edw\_tax\_type, '-1')

AND tr.location\_number = IFNULL(account.edw\_location\_number, -1)

) left join

(

select distinct

rfn\_party\_sk\_map.dip\_party\_id,

rfn\_party\_sk\_map.ird\_no

from

rfn\_party\_sk\_map as rfn\_party\_sk\_map

) as party

on

(

tr.ird\_number = IFNULL(party.ird\_no, -1)

)

### EDW Account

select

sk\_map.dip\_account\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

CAST(timeline.record\_active\_flag AS CHAR(1)) as record\_active\_flag,

CAST('FIRST' AS VARCHAR(10)) as account\_origin,

CAST(IFNULL(CAST(sk\_map.start\_account\_key AS VARCHAR), 'Unknown') AS VARCHAR(10)) as account\_key,

CAST(IFNULL(LPAD(CAST(tr.ird\_number AS VARCHAR(10)), 9, '0'),'Unknown') AS VARCHAR(10)) as account\_ird\_number,

CAST('N/A' AS VARCHAR(10)) as start\_profile\_number,

CAST('N/A' AS VARCHAR(50)) as start\_parent\_profile\_type,

CAST('N/A' AS VARCHAR(10)) as start\_parent\_profile\_number,

CAST(IFNULL(CAST(tr.location\_number AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END AS VARCHAR(255)) as legal\_name,

CAST(IFNULL(tradn.organisation\_name,

CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END) AS VARCHAR(255)) as trading\_name,

CAST(IFNULL(mp\_tt.dip\_description, IFNULL(tt.description, 'Unknown')) AS VARCHAR(30)) as account\_type,

CAST(IFNULL(mp\_tt.dip\_code\_value, IFNULL(tr.tax\_type, 'Unknown')) AS VARCHAR(30)) as account\_type\_code,

CASE WHEN tr.treg\_date\_start BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_start

ELSE '1900-01-01'

END as account\_start\_date,

CASE WHEN tr.treg\_date\_end BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_end

ELSE '9000-12-31'

END as account\_ceased\_date,

CAST(IFNULL(trsc.description, 'Unknown') AS VARCHAR(30)) as account\_status,

CAST(IFNULL(tr.treg\_status, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

CAST('TBC' AS VARCHAR(255)) as security\_level,

CAST('TBC' AS VARCHAR(10)) as security\_level\_code,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_description, IFNULL(cr.description, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(30)) as cessation\_reason,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_code\_value, IFNULL(tr.cessation\_reason\_code, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(10)) as cessation\_reason\_code,

CAST(IFNULL(mp\_ff.dip\_description, IFNULL(ff.description, 'Unknown'))

AS VARCHAR(30)) as filing\_frequency,

CAST(IFNULL(mp\_ff.dip\_code\_value, IFNULL(tr.filing\_frequency, 'Unknown'))

AS VARCHAR(10)) as filing\_frequency\_code,

CAST(IFNULL(mp\_er.dip\_description, IFNULL(er.description, 'Unknown'))

AS VARCHAR(30)) as exemption\_reason,

CAST(IFNULL(mp\_er.dip\_code\_value, IFNULL(tr.exemption\_reason\_code, 'Unknown'))

AS VARCHAR(10)) as exemption\_reason\_code,

CAST(IF(tr.exemption\_reason\_code IS NOT NULL, 'Y', 'N') AS CHAR(1)) as exempted\_flag,

CAST('N' AS CHAR(1)) as green\_listed\_flag,

CAST('N' AS CHAR(1)) as red\_listed\_flag,

CAST(IF(ah.halt\_status\_indicator IS NOT NULL, 'Y', 'N') AS CHAR(1)) as account\_halt\_flag,

CAST(IFNULL(ac.case\_open\_reason\_desc, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST('Unknown' AS VARCHAR(30)) as audit\_case\_active\_code,

CAST(IF(ac.derived\_tax\_type IS NOT NULL, 'Y', 'N') AS CHAR(1)) as audit\_case\_active\_flag,

CAST('Unknown' AS VARCHAR(30)) as stop\_mail\_desc,

CAST('Unknown' AS VARCHAR(10)) as stop\_mail\_code,

CAST('N' AS CHAR(1)) as stop\_mail\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* ac \*/

/\* ah \*/

/\* er \*/

/\* mp\_er \*/

/\* ff \*/

/\* mp\_ff \*/

/\* cr \*/

/\* mp\_cr \*/

/\* trsc \*/

/\* tt \*/

/\* mp\_tt \*/

/\* c \*/

/\* tradn \*/

/\* legn \*/

/\* tr \*/

/\* sk\_map \*/

rfn\_ae\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.ird\_no = sk\_map.ird\_no

AND timeline.edw\_tax\_type = sk\_map.edw\_tax\_type

AND timeline.edw\_location\_number = sk\_map.edw\_location\_number

AND timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_ae\_tax\_registrations as tr

on

(

timeline.ird\_no = tr.ird\_number

AND timeline.edw\_location\_number = tr.location\_number

AND timeline.edw\_tax\_type = tr.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tr.record\_effective\_timestamp AND tr.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr.treg\_date\_start

) left join

crw\_pe\_client\_names as legn

on

(

timeline.ird\_no = legn.ird\_number

AND legn.client\_name\_type = 'P'

AND legn.location\_number = 0

AND timeline.record\_effective\_timestamp BETWEEN legn.record\_effective\_timestamp AND legn.record\_expiry\_timestamp

) left join

crw\_pe\_client\_names as tradn

on

(

timeline.ird\_no = tradn.ird\_number

AND tradn.client\_name\_type = 'T'

AND tradn.location\_number = 1

AND timeline.record\_effective\_timestamp BETWEEN tradn.record\_effective\_timestamp AND tradn.record\_expiry\_timestamp

) left join

crw\_ae\_clients as c

on

(

timeline.ird\_no = c.ird\_number

AND timeline.record\_effective\_timestamp BETWEEN c.record\_effective\_timestamp AND c.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_tt

on

(

tr.tax\_type = mp\_tt.source\_code\_value

AND mp\_tt.source\_system = 'FIRST'

AND mp\_tt.source\_code\_name = 'TAX\_TYPE'

AND mp\_tt.source\_cust\_type IN ('', c.cust\_type)

AND timeline.record\_effective\_timestamp BETWEEN mp\_tt.record\_effective\_timestamp

AND mp\_tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_types as tt

on

(

tr.tax\_type = tt.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tt.record\_effective\_timestamp

AND tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_reg\_status\_codes as trsc

on

(

trsc.treg\_status\_code = tr.treg\_status

AND timeline.record\_effective\_timestamp BETWEEN trsc.record\_effective\_timestamp

AND trsc.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_cr

on

(

tr.cessation\_reason\_code = mp\_cr.source\_code\_value

AND tr.tax\_type IN ('', mp\_cr.source\_tax\_type)

AND mp\_cr.source\_system = 'FIRST'

AND mp\_cr.source\_code\_name = 'CESSATION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_cr.record\_effective\_timestamp

AND mp\_cr.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_cessation\_reasons as cr

on

(

cr.cessation\_reason\_code = tr.cessation\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN cr.record\_effective\_timestamp

AND cr.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_ff

on

(

tr.filing\_frequency = mp\_ff.source\_code\_value

AND mp\_ff.source\_system = 'FIRST'

AND mp\_ff.source\_code\_name = 'FILING\_FREQUENCY'

AND mp\_ff.dip\_code\_name = 'FILING\_FREQ\_PERIOD'

AND timeline.record\_effective\_timestamp BETWEEN mp\_ff.record\_effective\_timestamp

AND mp\_ff.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_filing\_frequencies as ff

on

(

ff.filing\_frequency = tr.filing\_frequency

AND timeline.record\_effective\_timestamp BETWEEN ff.record\_effective\_timestamp

AND ff.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_er

on

(

tr.exemption\_reason\_code = mp\_er.source\_code\_value

AND tr.tax\_type IN ('', mp\_er.source\_tax\_type)

AND mp\_er.source\_system = 'FIRST'

AND mp\_er.source\_code\_name = 'EXEMPTION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_er.record\_effective\_timestamp

AND mp\_er.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_exemption\_reasons as er

on

(

er.exemption\_reason\_code = tr.exemption\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN er.record\_effective\_timestamp

AND er.record\_expiry\_timestamp

) left join

crw\_ae\_account\_halts as ah

on

(

ah.ird\_number = tr.ird\_number

AND ah.derived\_location\_number = tr.location\_number

AND ah.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ah.record\_effective\_timestamp

AND ah.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ah.date\_halt\_start

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ah.date\_halt\_end

) left join

crw\_ae\_cm\_audit\_cases as ac

on

(

ac.ird\_number = tr.ird\_number

AND ac.location\_number = tr.location\_number

AND ac.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ac.record\_effective\_timestamp

AND ac.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ac.case\_opened\_date

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ac.case\_closed\_date

)

where

sk\_map.source = 'EDW'

AND tr.ird\_number IS NOT NULL

### EDW Account PAY

select

sk\_map.dip\_account\_sk as dip\_account\_pay\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

CAST(timeline.record\_active\_flag AS CHAR(1)) as record\_active\_flag,

CAST('FIRST' AS VARCHAR(10)) as account\_origin,

CAST(IFNULL(CAST(sk\_map.start\_account\_key AS VARCHAR), 'Unknown') AS VARCHAR(10)) as account\_key,

CAST(IFNULL(LPAD(CAST(tr.ird\_number AS VARCHAR(10)), 9, '0'),'Unknown') AS VARCHAR(10)) as account\_ird\_number,

CAST('N/A' AS VARCHAR(10)) as start\_profile\_number,

CAST('N/A' AS VARCHAR(50)) as start\_parent\_profile\_type,

CAST('N/A' AS VARCHAR(10)) as start\_parent\_profile\_number,

CAST(IFNULL(CAST(tr.location\_number AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END AS VARCHAR(255)) as legal\_name,

CAST(IFNULL(tradn.organisation\_name,

CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END) AS VARCHAR(255)) as trading\_name,

CAST(IFNULL(mp\_tt.dip\_description, IFNULL(tt.description, 'Unknown')) AS VARCHAR(30)) as account\_type,

CAST(IFNULL(mp\_tt.dip\_code\_value, IFNULL(tr.tax\_type, 'Unknown')) AS VARCHAR(30)) as account\_type\_code,

CASE WHEN tr.treg\_date\_start BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_start

ELSE '1900-01-01'

END as account\_start\_date,

CASE WHEN tr.treg\_date\_end BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_end

ELSE '9000-12-31'

END as account\_ceased\_date,

CAST(IFNULL(trsc.description, 'Unknown') AS VARCHAR(30)) as account\_status,

CAST(IFNULL(tr.treg\_status, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

CAST('TBC' AS VARCHAR(255)) as security\_level,

CAST('TBC' AS VARCHAR(10)) as security\_level\_code,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_description, IFNULL(cr.description, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(30)) as cessation\_reason,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_code\_value, IFNULL(tr.cessation\_reason\_code, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(10)) as cessation\_reason\_code,

CAST(IFNULL(mp\_ff.dip\_description, IFNULL(ff.description, 'Unknown'))

AS VARCHAR(30)) as filing\_frequency,

CAST(IFNULL(mp\_ff.dip\_code\_value, IFNULL(tr.filing\_frequency, 'Unknown'))

AS VARCHAR(10)) as filing\_frequency\_code,

CAST(IFNULL(mp\_er.dip\_description, IFNULL(er.description, 'Unknown'))

AS VARCHAR(30)) as exemption\_reason,

CAST(IFNULL(mp\_er.dip\_code\_value, IFNULL(tr.exemption\_reason\_code, 'Unknown'))

AS VARCHAR(10)) as exemption\_reason\_code,

CAST(IF(tr.exemption\_reason\_code IS NOT NULL, 'Y', 'N') AS CHAR(1)) as exempted\_flag,

CAST('N' AS CHAR(1)) as green\_listed\_flag,

CAST('N' AS CHAR(1)) as red\_listed\_flag,

CAST(IF(ah.halt\_status\_indicator IS NOT NULL, 'Y', 'N') AS CHAR(1)) as account\_halt\_flag,

CAST(IFNULL(ac.case\_open\_reason\_desc, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST('Unknown' AS VARCHAR(30)) as audit\_case\_active\_code,

CAST(IF(ac.derived\_tax\_type IS NOT NULL, 'Y', 'N') AS CHAR(1)) as audit\_case\_active\_flag,

CAST('Unknown' AS VARCHAR(30)) as stop\_mail\_desc,

CAST('Unknown' AS VARCHAR(10)) as stop\_mail\_code,

CAST('N' AS CHAR(1)) as stop\_mail\_flag,

CAST(IF(pay.paye\_employer\_class\_code = 'I', 'Y', 'N') AS CHAR(1)) as ir56\_flag,

CAST(IFNULL(mp\_fo.dip\_description, 'Unknown') AS VARCHAR(30)) as filing\_option,

CAST(IFNULL(mp\_fo.dip\_code\_value, 'Unknown') AS VARCHAR(10)) as filing\_option\_code,

CAST('N' AS CHAR(1)) as payday\_filer\_flag,

NULL as payday\_filer\_start\_date,

CAST(IF(tr\_cse.ird\_number IS NOT NULL, 'Y', 'N') AS CHAR(1)) as employer\_child\_support\_flag,

CAST(IF(tr\_sle.ird\_number IS NOT NULL, 'Y', 'N') AS CHAR(1)) as employer\_student\_loan\_flag,

CAST(IF(tr\_kse.ird\_number IS NOT NULL, 'Y', 'N') AS CHAR(1)) as kiwisaver\_employee\_dedn\_flag,

CAST(IF(tr\_ksr.ird\_number IS NOT NULL, 'Y', 'N') AS CHAR(1)) as kiwisaver\_employer\_cntrbflag,

CAST(IF(tr\_ssc.ird\_number IS NOT NULL, 'Y', 'N') AS CHAR(1)) as esct\_flag,

CAST('Unknown' AS VARCHAR(255)) as employer\_group,

CAST('Unknown' AS VARCHAR(10)) as employer\_group\_code,

CAST('N' AS CHAR(1)) as payday\_variation\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* tr\_ssc \*/

/\* tr\_ksr \*/

/\* tr\_kse \*/

/\* tr\_sle \*/

/\* tr\_cse \*/

/\* mp\_fo \*/

/\* pay \*/

/\* ac \*/

/\* ah \*/

/\* er \*/

/\* mp\_er \*/

/\* ff \*/

/\* mp\_ff \*/

/\* cr \*/

/\* mp\_cr \*/

/\* trsc \*/

/\* tt \*/

/\* mp\_tt \*/

/\* c \*/

/\* tradn \*/

/\* legn \*/

/\* tr \*/

/\* sk\_map \*/

rfn\_ae\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.ird\_no = sk\_map.ird\_no

AND timeline.edw\_tax\_type = sk\_map.edw\_tax\_type

AND timeline.edw\_location\_number = sk\_map.edw\_location\_number

AND timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_ae\_tax\_registrations as tr

on

(

timeline.ird\_no = tr.ird\_number

AND timeline.edw\_location\_number = tr.location\_number

AND timeline.edw\_tax\_type = tr.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tr.record\_effective\_timestamp AND tr.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr.treg\_date\_start

AND tr.tax\_type = 'PAY'

) left join

crw\_pe\_client\_names as legn

on

(

timeline.ird\_no = legn.ird\_number

AND legn.client\_name\_type = 'P'

AND legn.location\_number = 0

AND timeline.record\_effective\_timestamp BETWEEN legn.record\_effective\_timestamp AND legn.record\_expiry\_timestamp

) left join

crw\_pe\_client\_names as tradn

on

(

timeline.ird\_no = tradn.ird\_number

AND tradn.client\_name\_type = 'T'

AND tradn.location\_number = 1

AND timeline.record\_effective\_timestamp BETWEEN tradn.record\_effective\_timestamp AND tradn.record\_expiry\_timestamp

) left join

crw\_ae\_clients as c

on

(

timeline.ird\_no = c.ird\_number

AND timeline.record\_effective\_timestamp BETWEEN c.record\_effective\_timestamp AND c.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_tt

on

(

tr.tax\_type = mp\_tt.source\_code\_value

AND mp\_tt.source\_system = 'FIRST'

AND mp\_tt.source\_code\_name = 'TAX\_TYPE'

AND mp\_tt.source\_cust\_type IN ('', c.cust\_type)

AND timeline.record\_effective\_timestamp BETWEEN mp\_tt.record\_effective\_timestamp

AND mp\_tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_types as tt

on

(

tr.tax\_type = tt.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tt.record\_effective\_timestamp

AND tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_reg\_status\_codes as trsc

on

(

trsc.treg\_status\_code = tr.treg\_status

AND timeline.record\_effective\_timestamp BETWEEN trsc.record\_effective\_timestamp

AND trsc.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_cr

on

(

tr.cessation\_reason\_code = mp\_cr.source\_code\_value

AND tr.tax\_type IN ('', mp\_cr.source\_tax\_type)

AND mp\_cr.source\_system = 'FIRST'

AND mp\_cr.source\_code\_name = 'CESSATION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_cr.record\_effective\_timestamp

AND mp\_cr.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_cessation\_reasons as cr

on

(

cr.cessation\_reason\_code = tr.cessation\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN cr.record\_effective\_timestamp

AND cr.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_ff

on

(

tr.filing\_frequency = mp\_ff.source\_code\_value

AND mp\_ff.source\_system = 'FIRST'

AND mp\_ff.source\_code\_name = 'FILING\_FREQUENCY'

AND mp\_ff.dip\_code\_name = 'FILING\_FREQ\_PERIOD'

AND timeline.record\_effective\_timestamp BETWEEN mp\_ff.record\_effective\_timestamp

AND mp\_ff.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_filing\_frequencies as ff

on

(

ff.filing\_frequency = tr.filing\_frequency

AND timeline.record\_effective\_timestamp BETWEEN ff.record\_effective\_timestamp

AND ff.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_er

on

(

tr.exemption\_reason\_code = mp\_er.source\_code\_value

AND tr.tax\_type IN ('', mp\_er.source\_tax\_type)

AND mp\_er.source\_system = 'FIRST'

AND mp\_er.source\_code\_name = 'EXEMPTION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_er.record\_effective\_timestamp

AND mp\_er.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_exemption\_reasons as er

on

(

er.exemption\_reason\_code = tr.exemption\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN er.record\_effective\_timestamp

AND er.record\_expiry\_timestamp

) left join

crw\_ae\_account\_halts as ah

on

(

ah.ird\_number = tr.ird\_number

AND ah.derived\_location\_number = tr.location\_number

AND ah.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ah.record\_effective\_timestamp

AND ah.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ah.date\_halt\_start

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ah.date\_halt\_end

) left join

crw\_ae\_cm\_audit\_cases as ac

on

(

ac.ird\_number = tr.ird\_number

AND ac.location\_number = tr.location\_number

AND ac.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ac.record\_effective\_timestamp

AND ac.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ac.case\_opened\_date

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ac.case\_closed\_date

) left join

crw\_ae\_tax\_pay as pay

on

(

pay.ird\_number = tr.ird\_number

AND pay.location\_number = tr.location\_number

AND pay.tax\_type = tr.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN pay.record\_effective\_timestamp AND pay.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= pay.treg\_date\_start

) left join

mp\_source\_code as mp\_fo

on

(

tr.filing\_frequency = mp\_fo.source\_code\_value

AND mp\_fo.source\_system = 'FIRST'

AND mp\_fo.source\_tax\_type = 'PAY'

AND mp\_fo.source\_code\_name = 'FILING\_FREQUENCY'

AND mp\_fo.dip\_code\_name = 'FILING\_OPTION'

AND timeline.record\_effective\_timestamp BETWEEN mp\_fo.record\_effective\_timestamp

AND mp\_fo.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_registrations as tr\_cse

on

(

timeline.ird\_no = tr\_cse.ird\_number

AND timeline.edw\_location\_number = tr\_cse.location\_number

AND tr\_cse.tax\_type IN ('CSE', 'CPR', 'NCP')

AND timeline.record\_effective\_timestamp BETWEEN tr\_cse.record\_effective\_timestamp AND tr\_cse.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr\_cse.treg\_date\_start

) left join

crw\_ae\_tax\_registrations as tr\_sle

on

(

timeline.ird\_no = tr\_sle.ird\_number

AND timeline.edw\_location\_number = tr\_sle.location\_number

AND tr\_sle.tax\_type = 'SLE'

AND timeline.record\_effective\_timestamp BETWEEN tr\_sle.record\_effective\_timestamp AND tr\_sle.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr\_sle.treg\_date\_start

) left join

crw\_ae\_tax\_registrations as tr\_kse

on

(

timeline.ird\_no = tr\_kse.ird\_number

AND timeline.edw\_location\_number = tr\_kse.location\_number

AND tr\_kse.tax\_type = 'KSE'

AND timeline.record\_effective\_timestamp BETWEEN tr\_kse.record\_effective\_timestamp AND tr\_kse.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr\_kse.treg\_date\_start

) left join

crw\_ae\_tax\_registrations as tr\_ksr

on

(

timeline.ird\_no = tr\_ksr.ird\_number

AND timeline.edw\_location\_number = tr\_ksr.location\_number

AND tr\_ksr.tax\_type = 'KSR'

AND timeline.record\_effective\_timestamp BETWEEN tr\_ksr.record\_effective\_timestamp AND tr\_ksr.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr\_ksr.treg\_date\_start

) left join

crw\_ae\_tax\_registrations as tr\_ssc

on

(

timeline.ird\_no = tr\_ssc.ird\_number

AND timeline.edw\_location\_number = tr\_ssc.location\_number

AND tr\_ssc.tax\_type = 'SSC'

AND timeline.record\_effective\_timestamp BETWEEN tr\_ssc.record\_effective\_timestamp AND tr\_ssc.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr\_ssc.treg\_date\_start

)

where

sk\_map.source = 'EDW'

AND tr.ird\_number IS NOT NULL

### EDW Account PIE

select

sk\_map.dip\_account\_sk as dip\_account\_pie\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

CAST(timeline.record\_active\_flag AS CHAR(1)) as record\_active\_flag,

CAST('FIRST' AS VARCHAR(10)) as account\_origin,

CAST(IFNULL(CAST(sk\_map.start\_account\_key AS VARCHAR), 'Unknown') AS VARCHAR(10)) as account\_key,

CAST(IFNULL(LPAD(CAST(tr.ird\_number AS VARCHAR(10)), 9, '0'),'Unknown') AS VARCHAR(10)) as account\_ird\_number,

CAST('N/A' AS VARCHAR(10)) as start\_profile\_number,

CAST('N/A' AS VARCHAR(50)) as start\_parent\_profile\_type,

CAST('N/A' AS VARCHAR(10)) as start\_parent\_profile\_number,

CAST(IFNULL(CAST(tr.location\_number AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END AS VARCHAR(255)) as legal\_name,

CAST(IFNULL(tradn.organisation\_name,

CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END) AS VARCHAR(255)) as trading\_name,

CAST(IFNULL(mp\_tt.dip\_description, IFNULL(tt.description, 'Unknown')) AS VARCHAR(30)) as account\_type,

CAST(IFNULL(mp\_tt.dip\_code\_value, IFNULL(tr.tax\_type, 'Unknown')) AS VARCHAR(30)) as account\_type\_code,

CASE WHEN tr.treg\_date\_start BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_start

ELSE '1900-01-01'

END as account\_start\_date,

CASE WHEN tr.treg\_date\_end BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_end

ELSE '9000-12-31'

END as account\_ceased\_date,

CAST(IFNULL(trsc.description, 'Unknown') AS VARCHAR(30)) as account\_status,

CAST(IFNULL(tr.treg\_status, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

CAST('TBC' AS VARCHAR(255)) as security\_level,

CAST('TBC' AS VARCHAR(10)) as security\_level\_code,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_description, IFNULL(cr.description, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(30)) as cessation\_reason,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_code\_value, IFNULL(tr.cessation\_reason\_code, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(10)) as cessation\_reason\_code,

CAST(IFNULL(mp\_ff.dip\_description, IFNULL(ff.description, 'Unknown'))

AS VARCHAR(30)) as filing\_frequency,

CAST(IFNULL(mp\_ff.dip\_code\_value, IFNULL(tr.filing\_frequency, 'Unknown'))

AS VARCHAR(10)) as filing\_frequency\_code,

CAST(IFNULL(mp\_er.dip\_description, IFNULL(er.description, 'Unknown'))

AS VARCHAR(30)) as exemption\_reason,

CAST(IFNULL(mp\_er.dip\_code\_value, IFNULL(tr.exemption\_reason\_code, 'Unknown'))

AS VARCHAR(10)) as exemption\_reason\_code,

CAST(IF(tr.exemption\_reason\_code IS NOT NULL, 'Y', 'N') AS CHAR(1)) as exempted\_flag,

CAST('N' AS CHAR(1)) as green\_listed\_flag,

CAST('N' AS CHAR(1)) as red\_listed\_flag,

CAST(IF(ah.halt\_status\_indicator IS NOT NULL, 'Y', 'N') AS CHAR(1)) as account\_halt\_flag,

CAST(IFNULL(ac.case\_open\_reason\_desc, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST('Unknown' AS VARCHAR(30)) as audit\_case\_active\_code,

CAST(IF(ac.derived\_tax\_type IS NOT NULL, 'Y', 'N') AS CHAR(1)) as audit\_case\_active\_flag,

CAST('Unknown' AS VARCHAR(30)) as stop\_mail\_desc,

CAST('Unknown' AS VARCHAR(10)) as stop\_mail\_code,

CAST('N' AS CHAR(1)) as stop\_mail\_flag,

CAST(IF(pie.pie\_type = 'DB', 'Defined Benefit', IFNULL(piet.fstrdecode1, 'Unknown')) AS VARCHAR(30)) as pie\_type,

CAST(IFNULL(pie.pie\_type, 'Unknown') AS VARCHAR(30)) as pie\_type\_code,

CAST(IFNULL(mp\_pro.dip\_description, 'Unknown') AS VARCHAR(255)) as filing\_option,

CAST(IFNULL(mp\_pro.dip\_code\_value, 'Unknown') AS VARCHAR(10)) as filing\_option\_code,

CAST(IFNULL(mp\_iap.dip\_description, 'Unknown') AS VARCHAR(50)) as attribution\_period,

CAST(IFNULL(mp\_fdr.dip\_description, 'Unknown') AS VARCHAR(50)) as fdr\_calc\_period,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* mp\_fdr \*/

/\* mp\_iap \*/

/\* mp\_pro \*/

/\* piet \*/

/\* pie \*/

/\* ac \*/

/\* ah \*/

/\* er \*/

/\* mp\_er \*/

/\* ff \*/

/\* mp\_ff \*/

/\* cr \*/

/\* mp\_cr \*/

/\* trsc \*/

/\* tt \*/

/\* mp\_tt \*/

/\* c \*/

/\* tradn \*/

/\* legn \*/

/\* tr \*/

/\* sk\_map \*/

rfn\_ae\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.ird\_no = sk\_map.ird\_no

AND timeline.edw\_tax\_type = sk\_map.edw\_tax\_type

AND timeline.edw\_location\_number = sk\_map.edw\_location\_number

AND timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_ae\_tax\_registrations as tr

on

(

timeline.ird\_no = tr.ird\_number

AND timeline.edw\_location\_number = tr.location\_number

AND timeline.edw\_tax\_type = tr.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tr.record\_effective\_timestamp AND tr.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr.treg\_date\_start

AND tr.tax\_type = 'PIE'

) left join

crw\_pe\_client\_names as legn

on

(

timeline.ird\_no = legn.ird\_number

AND legn.client\_name\_type = 'P'

AND legn.location\_number = 0

AND timeline.record\_effective\_timestamp BETWEEN legn.record\_effective\_timestamp AND legn.record\_expiry\_timestamp

) left join

crw\_pe\_client\_names as tradn

on

(

timeline.ird\_no = tradn.ird\_number

AND tradn.client\_name\_type = 'T'

AND tradn.location\_number = 1

AND timeline.record\_effective\_timestamp BETWEEN tradn.record\_effective\_timestamp AND tradn.record\_expiry\_timestamp

) left join

crw\_ae\_clients as c

on

(

timeline.ird\_no = c.ird\_number

AND timeline.record\_effective\_timestamp BETWEEN c.record\_effective\_timestamp AND c.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_tt

on

(

tr.tax\_type = mp\_tt.source\_code\_value

AND mp\_tt.source\_system = 'FIRST'

AND mp\_tt.source\_code\_name = 'TAX\_TYPE'

AND mp\_tt.source\_cust\_type IN ('', c.cust\_type)

AND timeline.record\_effective\_timestamp BETWEEN mp\_tt.record\_effective\_timestamp

AND mp\_tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_types as tt

on

(

tr.tax\_type = tt.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tt.record\_effective\_timestamp

AND tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_reg\_status\_codes as trsc

on

(

trsc.treg\_status\_code = tr.treg\_status

AND timeline.record\_effective\_timestamp BETWEEN trsc.record\_effective\_timestamp

AND trsc.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_cr

on

(

tr.cessation\_reason\_code = mp\_cr.source\_code\_value

AND tr.tax\_type IN ('', mp\_cr.source\_tax\_type)

AND mp\_cr.source\_system = 'FIRST'

AND mp\_cr.source\_code\_name = 'CESSATION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_cr.record\_effective\_timestamp

AND mp\_cr.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_cessation\_reasons as cr

on

(

cr.cessation\_reason\_code = tr.cessation\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN cr.record\_effective\_timestamp

AND cr.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_ff

on

(

tr.filing\_frequency = mp\_ff.source\_code\_value

AND mp\_ff.source\_system = 'FIRST'

AND mp\_ff.source\_code\_name = 'FILING\_FREQUENCY'

AND mp\_ff.dip\_code\_name = 'FILING\_FREQ\_PERIOD'

AND timeline.record\_effective\_timestamp BETWEEN mp\_ff.record\_effective\_timestamp

AND mp\_ff.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_filing\_frequencies as ff

on

(

ff.filing\_frequency = tr.filing\_frequency

AND timeline.record\_effective\_timestamp BETWEEN ff.record\_effective\_timestamp

AND ff.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_er

on

(

tr.exemption\_reason\_code = mp\_er.source\_code\_value

AND tr.tax\_type IN ('', mp\_er.source\_tax\_type)

AND mp\_er.source\_system = 'FIRST'

AND mp\_er.source\_code\_name = 'EXEMPTION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_er.record\_effective\_timestamp

AND mp\_er.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_exemption\_reasons as er

on

(

er.exemption\_reason\_code = tr.exemption\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN er.record\_effective\_timestamp

AND er.record\_expiry\_timestamp

) left join

crw\_ae\_account\_halts as ah

on

(

ah.ird\_number = tr.ird\_number

AND ah.derived\_location\_number = tr.location\_number

AND ah.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ah.record\_effective\_timestamp

AND ah.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ah.date\_halt\_start

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ah.date\_halt\_end

) left join

crw\_ae\_cm\_audit\_cases as ac

on

(

ac.ird\_number = tr.ird\_number

AND ac.location\_number = tr.location\_number

AND ac.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ac.record\_effective\_timestamp

AND ac.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ac.case\_opened\_date

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ac.case\_closed\_date

) left join

crw\_ae\_tax\_pie as pie

on

(

pie.ird\_number = tr.ird\_number

AND pie.location\_number = tr.location\_number

AND pie.tax\_type = tr.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN pie.record\_effective\_timestamp

AND pie.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= pie.treg\_date\_start

) left join

crw\_as\_lan\_list\_item as piet

on

(

piet.fstrlistitem = pie.pie\_type

AND piet.fstrfield = 'NZ.ACCPIE.PIEType'

AND timeline.record\_effective\_timestamp BETWEEN piet.record\_effective\_timestamp

AND piet.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_pro

on

(

pie.pie\_report\_option = mp\_pro.source\_code\_value

AND mp\_pro.source\_system = 'FIRST'

AND mp\_pro.source\_code\_name = 'PIE\_REPORT\_OPTION'

AND mp\_pro.source\_tax\_type = 'PIE'

AND timeline.record\_effective\_timestamp BETWEEN mp\_pro.record\_effective\_timestamp

AND mp\_pro.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_iap

on

(

pie.income\_allocation\_period = mp\_iap.source\_code\_value

AND mp\_iap.source\_system = 'FIRST'

AND mp\_iap.source\_code\_name = 'INCOME\_ALLOCATION\_PERIOD'

AND mp\_iap.source\_tax\_type = 'PIE'

AND timeline.record\_effective\_timestamp BETWEEN mp\_iap.record\_effective\_timestamp

AND mp\_iap.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_fdr

on

(

pie.fdr\_calculation\_period = mp\_fdr.source\_code\_value

AND mp\_fdr.source\_system = 'FIRST'

AND mp\_fdr.source\_code\_name = 'FDR\_CALCULATION\_PERIOD'

AND mp\_fdr.source\_tax\_type = 'PIE'

AND timeline.record\_effective\_timestamp BETWEEN mp\_fdr.record\_effective\_timestamp

AND mp\_fdr.record\_expiry\_timestamp

)

where

sk\_map.source = 'EDW'

AND tr.ird\_number IS NOT NULL

### EDW Account INC

select

sk\_map.dip\_account\_sk as dip\_account\_inc\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

CAST(timeline.record\_active\_flag AS CHAR(1)) as record\_active\_flag,

CAST('FIRST' AS VARCHAR(10)) as account\_origin,

CAST(IFNULL(CAST(sk\_map.start\_account\_key AS VARCHAR), 'Unknown') AS VARCHAR(10)) as account\_key,

CAST(IFNULL(LPAD(CAST(tr.ird\_number AS VARCHAR(10)), 9, '0'),'Unknown') AS VARCHAR(10)) as account\_ird\_number,

CAST('N/A' AS VARCHAR(10)) as start\_profile\_number,

CAST('N/A' AS VARCHAR(50)) as start\_parent\_profile\_type,

CAST('N/A' AS VARCHAR(10)) as start\_parent\_profile\_number,

CAST(IFNULL(CAST(tr.location\_number AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END AS VARCHAR(255)) as legal\_name,

CAST(IFNULL(tradn.organisation\_name,

CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END) AS VARCHAR(255)) as trading\_name,

CAST(IFNULL(mp\_tt.dip\_description, IFNULL(tt.description, 'Unknown')) AS VARCHAR(30)) as account\_type,

CAST(IFNULL(mp\_tt.dip\_code\_value, IFNULL(tr.tax\_type, 'Unknown')) AS VARCHAR(30)) as account\_type\_code,

CASE WHEN tr.treg\_date\_start BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_start

ELSE '1900-01-01'

END as account\_start\_date,

CASE WHEN tr.treg\_date\_end BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_end

ELSE '9000-12-31'

END as account\_ceased\_date,

CAST(IFNULL(trsc.description, 'Unknown') AS VARCHAR(30)) as account\_status,

CAST(IFNULL(tr.treg\_status, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

CAST('TBC' AS VARCHAR(255)) as security\_level,

CAST('TBC' AS VARCHAR(10)) as security\_level\_code,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_description, IFNULL(cr.description, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(30)) as cessation\_reason,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_code\_value, IFNULL(tr.cessation\_reason\_code, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(10)) as cessation\_reason\_code,

CAST(IFNULL(mp\_ff.dip\_description, IFNULL(ff.description, 'Unknown'))

AS VARCHAR(30)) as filing\_frequency,

CAST(IFNULL(mp\_ff.dip\_code\_value, IFNULL(tr.filing\_frequency, 'Unknown'))

AS VARCHAR(10)) as filing\_frequency\_code,

CAST(IFNULL(mp\_er.dip\_description, IFNULL(er.description, 'Unknown'))

AS VARCHAR(30)) as exemption\_reason,

CAST(IFNULL(mp\_er.dip\_code\_value, IFNULL(tr.exemption\_reason\_code, 'Unknown'))

AS VARCHAR(10)) as exemption\_reason\_code,

CAST(IF(tr.exemption\_reason\_code IS NOT NULL, 'Y', 'N') AS CHAR(1)) as exempted\_flag,

CAST('N' AS CHAR(1)) as green\_listed\_flag,

CAST('N' AS CHAR(1)) as red\_listed\_flag,

CAST(IF(ah.halt\_status\_indicator IS NOT NULL, 'Y', 'N') AS CHAR(1)) as account\_halt\_flag,

CAST(IFNULL(ac.case\_open\_reason\_desc, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST('Unknown' AS VARCHAR(30)) as audit\_case\_active\_code,

CAST(IF(ac.derived\_tax\_type IS NOT NULL, 'Y', 'N') AS CHAR(1)) as audit\_case\_active\_flag,

CAST('Unknown' AS VARCHAR(30)) as stop\_mail\_desc,

CAST('Unknown' AS VARCHAR(10)) as stop\_mail\_code,

CAST('N' AS CHAR(1)) as stop\_mail\_flag,

CAST(CASE WHEN c.balance\_date IN (NULL, 0) THEN 'Unknown'

ELSE CONCAT(MONTHNAME(to\_timestamp(concat('2000',lpad(cast(c.balance\_date as varchar(4)),4,'0')), 'yyyyMMdd')), ' ',CAST(DAY(to\_timestamp(concat('2000',lpad(cast(c.balance\_date as varchar(4)),4,'0')), 'yyyyMMdd')) AS VARCHAR(2)))

END AS VARCHAR(30)) as balance\_date,

CAST('N' AS CHAR(1)) as zero\_tailored\_rate\_flag,

CAST('N' AS CHAR(1)) as trans\_tasman\_flag,

CAST(IF(cu.entity\_class = 'AI', 'Y', 'N') AS CHAR(1)) as agent\_non\_res\_insurer\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* cu \*/

/\* ac \*/

/\* ah \*/

/\* er \*/

/\* mp\_er \*/

/\* ff \*/

/\* mp\_ff \*/

/\* cr \*/

/\* mp\_cr \*/

/\* trsc \*/

/\* tt \*/

/\* mp\_tt \*/

/\* c \*/

/\* tradn \*/

/\* legn \*/

/\* tr \*/

/\* sk\_map \*/

rfn\_ae\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.ird\_no = sk\_map.ird\_no

AND timeline.edw\_tax\_type = sk\_map.edw\_tax\_type

AND timeline.edw\_location\_number = sk\_map.edw\_location\_number

AND timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_ae\_tax\_registrations as tr

on

(

timeline.ird\_no = tr.ird\_number

AND timeline.edw\_location\_number = tr.location\_number

AND timeline.edw\_tax\_type = tr.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tr.record\_effective\_timestamp AND tr.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr.treg\_date\_start

AND tr.tax\_type = 'INC'

) left join

crw\_pe\_client\_names as legn

on

(

timeline.ird\_no = legn.ird\_number

AND legn.client\_name\_type = 'P'

AND legn.location\_number = 0

AND timeline.record\_effective\_timestamp BETWEEN legn.record\_effective\_timestamp AND legn.record\_expiry\_timestamp

) left join

crw\_pe\_client\_names as tradn

on

(

timeline.ird\_no = tradn.ird\_number

AND tradn.client\_name\_type = 'T'

AND tradn.location\_number = 1

AND timeline.record\_effective\_timestamp BETWEEN tradn.record\_effective\_timestamp AND tradn.record\_expiry\_timestamp

) left join

crw\_ae\_clients as c

on

(

timeline.ird\_no = c.ird\_number

AND timeline.record\_effective\_timestamp BETWEEN c.record\_effective\_timestamp AND c.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_tt

on

(

tr.tax\_type = mp\_tt.source\_code\_value

AND mp\_tt.source\_system = 'FIRST'

AND mp\_tt.source\_code\_name = 'TAX\_TYPE'

AND mp\_tt.source\_cust\_type IN ('', c.cust\_type)

AND timeline.record\_effective\_timestamp BETWEEN mp\_tt.record\_effective\_timestamp

AND mp\_tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_types as tt

on

(

tr.tax\_type = tt.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tt.record\_effective\_timestamp

AND tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_reg\_status\_codes as trsc

on

(

trsc.treg\_status\_code = tr.treg\_status

AND timeline.record\_effective\_timestamp BETWEEN trsc.record\_effective\_timestamp

AND trsc.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_cr

on

(

tr.cessation\_reason\_code = mp\_cr.source\_code\_value

AND tr.tax\_type IN ('', mp\_cr.source\_tax\_type)

AND mp\_cr.source\_system = 'FIRST'

AND mp\_cr.source\_code\_name = 'CESSATION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_cr.record\_effective\_timestamp

AND mp\_cr.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_cessation\_reasons as cr

on

(

cr.cessation\_reason\_code = tr.cessation\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN cr.record\_effective\_timestamp

AND cr.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_ff

on

(

tr.filing\_frequency = mp\_ff.source\_code\_value

AND mp\_ff.source\_system = 'FIRST'

AND mp\_ff.source\_code\_name = 'FILING\_FREQUENCY'

AND mp\_ff.dip\_code\_name = 'FILING\_FREQ\_PERIOD'

AND timeline.record\_effective\_timestamp BETWEEN mp\_ff.record\_effective\_timestamp

AND mp\_ff.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_filing\_frequencies as ff

on

(

ff.filing\_frequency = tr.filing\_frequency

AND timeline.record\_effective\_timestamp BETWEEN ff.record\_effective\_timestamp

AND ff.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_er

on

(

tr.exemption\_reason\_code = mp\_er.source\_code\_value

AND tr.tax\_type IN ('', mp\_er.source\_tax\_type)

AND mp\_er.source\_system = 'FIRST'

AND mp\_er.source\_code\_name = 'EXEMPTION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_er.record\_effective\_timestamp

AND mp\_er.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_exemption\_reasons as er

on

(

er.exemption\_reason\_code = tr.exemption\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN er.record\_effective\_timestamp

AND er.record\_expiry\_timestamp

) left join

crw\_ae\_account\_halts as ah

on

(

ah.ird\_number = tr.ird\_number

AND ah.derived\_location\_number = tr.location\_number

AND ah.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ah.record\_effective\_timestamp

AND ah.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ah.date\_halt\_start

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ah.date\_halt\_end

) left join

crw\_ae\_cm\_audit\_cases as ac

on

(

ac.ird\_number = tr.ird\_number

AND ac.location\_number = tr.location\_number

AND ac.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ac.record\_effective\_timestamp

AND ac.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ac.case\_opened\_date

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ac.case\_closed\_date

) left join

crw\_pe\_customers as cu

on

(

timeline.ird\_no = cu.ird\_number

AND cu.location\_number = 1

AND timeline.record\_effective\_timestamp BETWEEN cu.record\_effective\_timestamp AND cu.record\_expiry\_timestamp

)

where

sk\_map.source = 'EDW'

AND tr.ird\_number IS NOT NULL;

### EDW Account GST

select

sk\_map.dip\_account\_sk as dip\_account\_gst\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

CAST(timeline.record\_active\_flag AS CHAR(1)) as record\_active\_flag,

CAST('FIRST' AS VARCHAR(10)) as account\_origin,

CAST(IFNULL(CAST(sk\_map.start\_account\_key AS VARCHAR), 'Unknown') AS VARCHAR(10)) as account\_key,

CAST(IFNULL(LPAD(CAST(tr.ird\_number AS VARCHAR(10)), 9, '0'),'Unknown') AS VARCHAR(10)) as account\_ird\_number,

CAST('N/A' AS VARCHAR(10)) as start\_profile\_number,

CAST('N/A' AS VARCHAR(50)) as start\_parent\_profile\_type,

CAST('N/A' AS VARCHAR(10)) as start\_parent\_profile\_number,

CAST(IFNULL(CAST(tr.location\_number AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END AS VARCHAR(255)) as legal\_name,

CAST(IFNULL(tradn.organisation\_name,

CASE WHEN c.cust\_type = 'ORG' THEN IFNULL(legn.organisation\_name, 'Unknown')

WHEN c.cust\_type = 'IND' AND legn.first\_names IS NOT NULL

AND legn.surname IS NOT NULL THEN CONCAT(legn.first\_names, ' ', legn.surname)

ELSE 'Unknown'

END) AS VARCHAR(255)) as trading\_name,

CAST(IFNULL(mp\_tt.dip\_description, IFNULL(tt.description, 'Unknown')) AS VARCHAR(30)) as account\_type,

CAST(IFNULL(mp\_tt.dip\_code\_value, IFNULL(tr.tax\_type, 'Unknown')) AS VARCHAR(30)) as account\_type\_code,

CASE WHEN tr.treg\_date\_start BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_start

ELSE '1900-01-01'

END as account\_start\_date,

CASE WHEN tr.treg\_date\_end BETWEEN '1880-01-01' and now() THEN tr.treg\_date\_end

ELSE '9000-12-31'

END as account\_ceased\_date,

CAST(IFNULL(trsc.description, 'Unknown') AS VARCHAR(30)) as account\_status,

CAST(IFNULL(tr.treg\_status, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

CAST('TBC' AS VARCHAR(255)) as security\_level,

CAST('TBC' AS VARCHAR(10)) as security\_level\_code,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_description, IFNULL(cr.description, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(30)) as cessation\_reason,

CAST(CASE WHEN tr.treg\_status = 'C' THEN IFNULL(mp\_cr.dip\_code\_value, IFNULL(tr.cessation\_reason\_code, 'Unknown'))

ELSE 'N/A'

END

AS VARCHAR(10)) as cessation\_reason\_code,

CAST(IFNULL(mp\_ff.dip\_description, IFNULL(ff.description, 'Unknown'))

AS VARCHAR(30)) as filing\_frequency,

CAST(IFNULL(mp\_ff.dip\_code\_value, IFNULL(tr.filing\_frequency, 'Unknown'))

AS VARCHAR(10)) as filing\_frequency\_code,

CAST(IFNULL(mp\_er.dip\_description, IFNULL(er.description, 'Unknown'))

AS VARCHAR(30)) as exemption\_reason,

CAST(IFNULL(mp\_er.dip\_code\_value, IFNULL(tr.exemption\_reason\_code, 'Unknown'))

AS VARCHAR(10)) as exemption\_reason\_code,

CAST(IF(tr.exemption\_reason\_code IS NOT NULL, 'Y', 'N') AS CHAR(1)) as exempted\_flag,

CAST('N' AS CHAR(1)) as green\_listed\_flag,

CAST('N' AS CHAR(1)) as red\_listed\_flag,

CAST(IF(ah.halt\_status\_indicator IS NOT NULL, 'Y', 'N') AS CHAR(1)) as account\_halt\_flag,

CAST(IFNULL(ac.case\_open\_reason\_desc, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST('Unknown' AS VARCHAR(30)) as audit\_case\_active\_code,

CAST(IF(ac.derived\_tax\_type IS NOT NULL, 'Y', 'N') AS CHAR(1)) as audit\_case\_active\_flag,

CAST('Unknown' AS VARCHAR(30)) as stop\_mail\_desc,

CAST('Unknown' AS VARCHAR(10)) as stop\_mail\_code,

CAST('N' AS CHAR(1)) as stop\_mail\_flag,

CAST(IFNULL(mp\_ab.dip\_description, 'Unknown') AS VARCHAR(10)) as accounting\_basis,

CAST(IF(gst.nat\_of\_rgn = 'S', 'Special', IFNULL(mp\_nor.dip\_description, 'Unknown')) AS VARCHAR(10)) as registration\_circumstance,

CAST('N' AS CHAR(1)) as included\_in\_prices\_flag,

CAST(IF(gst.exempt\_supp\_ind = 'Y', 'Y', 'N') AS CHAR(1)) as exempt\_supplies\_flag,

CAST(IF(gst.exp\_ind IN ('E', 'B'), 'Y', 'N') AS CHAR(1)) as exporter\_flag,

CAST(IF(gst.exp\_ind IN ('I', 'B'), 'Y', 'N') AS CHAR(1)) as importer\_flag,

CAST('N' AS CHAR(1)) as unpoliced\_filer\_flag,

CAST(IF(gst.self\_inv\_ind ='Y', 'Y', 'N') AS CHAR(1)) as self\_invoicer\_flag,

CAST('N' AS CHAR(1)) as hospice\_filer\_flag,

CAST('Unknown' AS VARCHAR(10)) as hospice\_type,

CAST('Unknown' AS VARCHAR(30)) as hospice\_type\_desc,

CAST('N' AS CHAR(1)) as gst\_on\_remote\_srvcs\_flag,

CAST('Unknown' AS VARCHAR(30)) as gst\_on\_remote\_srvcs\_type,

CAST('Unknown' AS VARCHAR(30)) as gst\_on\_remote\_srvcs\_code,

CAST('N' AS CHAR(1)) as gst\_zero\_rated\_fin\_srv\_flag,

CAST('N' AS CHAR(1)) as gst\_non\_resident\_claimant\_flag,

CAST('N' AS CHAR(1)) as taxable\_activity\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* mp\_nor \*/

/\* mp\_ab \*/

/\* gst \*/

/\* ac \*/

/\* ah \*/

/\* er \*/

/\* mp\_er \*/

/\* ff \*/

/\* mp\_ff \*/

/\* cr \*/

/\* mp\_cr \*/

/\* trsc \*/

/\* tt \*/

/\* mp\_tt \*/

/\* c \*/

/\* tradn \*/

/\* legn \*/

/\* tr \*/

/\* sk\_map \*/

rfn\_ae\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.ird\_no = sk\_map.ird\_no

AND timeline.edw\_tax\_type = sk\_map.edw\_tax\_type

AND timeline.edw\_location\_number = sk\_map.edw\_location\_number

AND timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_ae\_tax\_registrations as tr

on

(

timeline.ird\_no = tr.ird\_number

AND timeline.edw\_location\_number = tr.location\_number

AND timeline.edw\_tax\_type = tr.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tr.record\_effective\_timestamp AND tr.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= tr.treg\_date\_start

AND tr.tax\_type = 'GST'

) left join

crw\_pe\_client\_names as legn

on

(

timeline.ird\_no = legn.ird\_number

AND legn.client\_name\_type = 'P'

AND legn.location\_number = 0

AND timeline.record\_effective\_timestamp BETWEEN legn.record\_effective\_timestamp AND legn.record\_expiry\_timestamp

) left join

crw\_pe\_client\_names as tradn

on

(

timeline.ird\_no = tradn.ird\_number

AND tradn.client\_name\_type = 'T'

AND tradn.location\_number = 1

AND timeline.record\_effective\_timestamp BETWEEN tradn.record\_effective\_timestamp AND tradn.record\_expiry\_timestamp

) left join

crw\_ae\_clients as c

on

(

timeline.ird\_no = c.ird\_number

AND timeline.record\_effective\_timestamp BETWEEN c.record\_effective\_timestamp AND c.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_tt

on

(

tr.tax\_type = mp\_tt.source\_code\_value

AND mp\_tt.source\_system = 'FIRST'

AND mp\_tt.source\_code\_name = 'TAX\_TYPE'

AND mp\_tt.source\_cust\_type IN ('', c.cust\_type)

AND timeline.record\_effective\_timestamp BETWEEN mp\_tt.record\_effective\_timestamp

AND mp\_tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_types as tt

on

(

tr.tax\_type = tt.tax\_type

AND timeline.record\_effective\_timestamp BETWEEN tt.record\_effective\_timestamp

AND tt.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_reg\_status\_codes as trsc

on

(

trsc.treg\_status\_code = tr.treg\_status

AND timeline.record\_effective\_timestamp BETWEEN trsc.record\_effective\_timestamp

AND trsc.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_cr

on

(

tr.cessation\_reason\_code = mp\_cr.source\_code\_value

AND tr.tax\_type IN ('', mp\_cr.source\_tax\_type)

AND mp\_cr.source\_system = 'FIRST'

AND mp\_cr.source\_code\_name = 'CESSATION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_cr.record\_effective\_timestamp

AND mp\_cr.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_cessation\_reasons as cr

on

(

cr.cessation\_reason\_code = tr.cessation\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN cr.record\_effective\_timestamp

AND cr.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_ff

on

(

tr.filing\_frequency = mp\_ff.source\_code\_value

AND mp\_ff.source\_system = 'FIRST'

AND mp\_ff.source\_code\_name = 'FILING\_FREQUENCY'

AND mp\_ff.dip\_code\_name = 'FILING\_FREQ\_PERIOD'

AND timeline.record\_effective\_timestamp BETWEEN mp\_ff.record\_effective\_timestamp

AND mp\_ff.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_filing\_frequencies as ff

on

(

ff.filing\_frequency = tr.filing\_frequency

AND timeline.record\_effective\_timestamp BETWEEN ff.record\_effective\_timestamp

AND ff.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_er

on

(

tr.exemption\_reason\_code = mp\_er.source\_code\_value

AND tr.tax\_type IN ('', mp\_er.source\_tax\_type)

AND mp\_er.source\_system = 'FIRST'

AND mp\_er.source\_code\_name = 'EXEMPTION\_REASON'

AND timeline.record\_effective\_timestamp BETWEEN mp\_er.record\_effective\_timestamp

AND mp\_er.record\_expiry\_timestamp

) left join

crw\_ae\_tax\_exemption\_reasons as er

on

(

er.exemption\_reason\_code = tr.exemption\_reason\_code

AND timeline.record\_effective\_timestamp BETWEEN er.record\_effective\_timestamp

AND er.record\_expiry\_timestamp

) left join

crw\_ae\_account\_halts as ah

on

(

ah.ird\_number = tr.ird\_number

AND ah.derived\_location\_number = tr.location\_number

AND ah.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ah.record\_effective\_timestamp

AND ah.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ah.date\_halt\_start

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ah.date\_halt\_end

) left join

crw\_ae\_cm\_audit\_cases as ac

on

(

ac.ird\_number = tr.ird\_number

AND ac.location\_number = tr.location\_number

AND ac.derived\_tax\_type = tr. tax\_type

AND timeline.record\_effective\_timestamp BETWEEN ac.record\_effective\_timestamp

AND ac.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= ac.case\_opened\_date

AND DAYS\_ADD(timeline.record\_effective\_timestamp,-1) <= ac.case\_closed\_date

) left join

crw\_ae\_tax\_gst as gst

on

(

gst.ird\_number = tr.ird\_number

AND gst.tax\_type = tr.tax\_type

AND gst.location\_number = tr.location\_number

AND timeline.record\_effective\_timestamp BETWEEN gst.record\_effective\_timestamp AND gst.record\_expiry\_timestamp

AND timeline.record\_effective\_timestamp >= gst.treg\_date\_start

) left join

mp\_source\_code as mp\_ab

on

(

gst.accounting\_basis\_code = mp\_ab.source\_code\_value

AND mp\_ab.source\_system = 'FIRST'

AND mp\_ab.source\_code\_name = 'ACCOUNTING\_BASIS\_CODE'

AND timeline.record\_effective\_timestamp BETWEEN mp\_ab.record\_effective\_timestamp

AND mp\_ab.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_nor

on

(

gst.nat\_of\_rgn = mp\_nor.source\_code\_value

AND mp\_nor.source\_system = 'FIRST'

AND mp\_nor.source\_code\_name = 'NAT\_OF\_REGN'

AND timeline.record\_effective\_timestamp BETWEEN mp\_nor.record\_effective\_timestamp

AND mp\_nor.record\_expiry\_timestamp

)

where

sk\_map.source = 'EDW'

AND tr.ird\_number IS NOT NULL

### Derived table: crw\_ae\_account\_halts

select

tr.location\_number, -- derived\_location\_number

tr.tax\_type, -- derived\_tax\_type

case when ah.halt\_level\_indicator = ‘C’ then 3

when ah.halt\_level\_indicator = ‘L’ then 2

when ah.halt\_level\_indicator = ‘T’ then 1

end, -- derived\_level

ah.\*

from dss\_account\_halts ah

LEFT OUTER JOIN crw\_ae\_tax\_registrations tr

on tr.ird\_number = ah.ird\_number

and tr.location\_number = (case when ah.halt\_level\_indicator = ‘C’ then 0 else ah.location\_number end)

and tr.tax\_type = (case when ah.halt\_level\_indicator in (‘C’,’L’) then ‘XXX’ else ah.tax\_type end)

and ah.date\_halt\_start between tr.record\_effective\_timestamp and tr.record\_expiry\_timestamp

and ah.halt\_level\_indicator in ('C','L','T')

### Derived table: crw\_ae\_cm\_audit\_cases

select

ac.ird\_number,

ac.location\_number,

ac.case\_opened\_date,

ac.case\_closed\_date,

ap.tax\_type --derived tax type

ap.case\_open\_reason\_desc

from dss\_cm\_audit\_cases ac

LEFT OUTER JOIN (select distinct case\_identifier, tax\_type

from dss\_cm\_audit\_periods

where tax\_type is not null) ap

on ap.case\_identifier = ac.case\_identifier

NB. tax\_type may be null for audit periods.

# Appendix F – Example SQL for Context – Start

The ‘Select’ column names in these SQL statements will be referred to in the “Source Table/Column” fields of the transformation rules and are intended to give context to those rules as well as a guide on how to merge all the source objects into the final Account Type data streams.

### Start Party Account

select

CAST(CONCAT('DIP-C-', CAST(crw\_as\_tblaccount.flngcustomerkey AS VARCHAR)) AS VARCHAR(30)) as dip\_party\_id,

CAST(CONCAT('DIP-A-', CAST(crw\_as\_tblaccount.flngaccountkey AS VARCHAR)) AS VARCHAR(30)) as dip\_account\_id,

crw\_as\_tblaccount.record\_effective\_timestamp,

crw\_as\_tblaccount.record\_expiry\_timestamp,

crw\_as\_tblaccount.record\_active\_flag,

CAST('OWNER' AS VARCHAR(10)) as relationship\_type,

crw\_as\_tblaccount.fdtmcommence as relationship\_effective\_from,

crw\_as\_tblaccount.fdtmcease as relationship\_effective\_to,

CAST('START' AS VARCHAR(10)) as relationship\_origin,

crw\_as\_tblaccount.insert\_object\_run\_key,

crw\_as\_tblaccount.update\_object\_run\_key

from

crw\_as\_tblaccount as crw\_as\_tblaccount

### Start Account

select

sk\_map.dip\_account\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

cast(timeline.record\_active\_flag as char(1)) as record\_active\_flag,

CAST('START' AS VARCHAR(10)) as account\_origin,

CAST(a.flngaccountkey AS VARCHAR(10)) as account\_key,

cast(ifnull(lpad(cast(sk\_map.ird\_no as varchar(10)),9,'0'),'Unknown') as varchar(10)) as account\_ird\_number,

cast(ifnull(CAST(a.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_profile\_number,

cast(ifnull(ppt.fstrdecode2, 'Unknown') AS VARCHAR(50)) as start\_parent\_profile\_type,

cast(ifnull(CAST(pp.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_number,

cast(ifnull(CAST(nz\_std.fintheritagelocationnumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE

WHEN c.fstrcustomertype = 'COM' THEN ifnull(lgl.fstrlistformatname, 'Unknown')

WHEN c.fstrcustomertype IN ('IND', 'CHD') THEN

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end

end AS VARCHAR(255)) as legal\_name,

CAST(ifnull(dba.fstrlistformatname,

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end)

AS VARCHAR(255)) as trading\_name,

cast(ifnull(actyp.fstrdecode2, 'Unknown') AS VARCHAR(30)) as account\_type,

cast(ifnull(a.fstraccounttype, 'Unknown') AS VARCHAR(30)) as account\_type\_code,

case when a.fdtmcommence between '1880-01-01' and now() then date\_trunc('day', a.fdtmcommence)

when a.fdtmcreated between '1880-01-01' and now() then date\_trunc('day', a.fdtmcreated)

else '1900-01-01' end as account\_start\_date,

if(a.fdtmcease between '1880-01-01' and now(), date\_trunc('day', a.fdtmcease), '9000-12-31') as account\_ceased\_date,

CAST(CASE a.fstrstatus

WHEN 'ACT' THEN 'Active'

WHEN 'CLS' THEN 'Closed'

WHEN 'SUS' THEN 'Suspended'

ELSE 'Unknown'

END AS VARCHAR(30)) as account\_status,

cast(ifnull(a.fstrstatus, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

cast(ifnull(cld.fstrdecode2, 'Unknown') AS VARCHAR(255)) as security\_level,

cast(ifnull(cl.fstrcustomerlevel, 'Unknown') AS VARCHAR(10)) as security\_level\_code,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN ifnull(cr.fstrdecode1, 'Unknown')

ELSE 'N/A' END AS VARCHAR(30)) as cessation\_reason,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN a.fstrClosureReason

ELSE 'N/A' END AS VARCHAR(10)) as cessation\_reason\_code,

cast(ifnull(ffp.fstrregularfrequency, 'Unknown') AS VARCHAR(30)) as filing\_frequency,

cast(ifnull(a.fstrfiling, 'Unknown') AS VARCHAR(10)) as filing\_frequency\_code,

CAST(

CASE WHEN exmpt.fstrindicator is not null

AND a.fstraccounttype = 'IPE'

AND accipe.fstrexemptionreason is not null

THEN ifnull(ipexmp.fstrdecode1, 'Unknown')

WHEN exmpt.fstrindicator is not null

AND a.fstraccounttype IN ('ITN', 'IIT')

AND accinc.fstrexemptionreason is not null

THEN ifnull(incxmp.fstrdecode1, 'Unknown')

WHEN exmpt.fstrindicator is not null

AND lind.fstrdecode2 is not null

THEN lind.fstrdecode2

ELSE 'Unknown' END AS VARCHAR(30)) as exemption\_reason,

CAST(

CASE WHEN exmpt.fstrindicator is not null

AND a.fstraccounttype = 'IPE'

AND accipe.fstrexemptionreason is not null

THEN accipe.fstrexemptionreason

WHEN exmpt.fstrindicator is not null

AND a.fstraccounttype IN ('ITN', 'IIT')

AND accinc.fstrexemptionreason is not null

THEN accinc.fstrexemptionreason

ELSE exmpt.fstrindicator END AS VARCHAR(30)) as exemption\_reason\_code,

CAST(if(exmpt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as exempted\_flag,

CAST(if(grnlst.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as green\_listed\_flag,

CAST(if(frdred.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as red\_listed\_flag,

CAST(if(acchlt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as account\_halt\_flag,

CAST (IF(audact.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST(ifnull(audact.fstrindicator, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_code,

CAST(if(audact.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as audit\_case\_active\_flag,

CAST (IF(dontmail.fstrindicator IS NULL, 'Unknown', ifnull(mlind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as stop\_mail\_desc,

CAST(IFNULL(dontmail.fstrindicator, 'Unknown') AS VARCHAR(10)) as stop\_mail\_code,

cast(if(mlind.fstrdecode2 IS NULL, 'N', 'Y') AS CHAR(1)) as stop\_mail\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* ffp \*/

/\* cld \*/

/\* cl \*/

/\* mlind \*/

/\* dontmail \*/

/\* alind \*/

/\* audact \*/

/\* acchlt \*/

/\* frdred \*/

/\* grnlst \*/

/\* lind \*/

/\* exmpt \*/

/\* incxmp \*/

/\* accinc \*/

/\* ipexmp \*/

/\* accipe \*/

/\* ff \*/

/\* cr \*/

/\* actyp \*/

/\* c \*/

/\* dba \*/

/\* lgl \*/

/\* nz\_std \*/

/\* ppt \*/

/\* pp \*/

/\* p \*/

/\* a \*/

/\* sk\_map \*/

rfn\_as\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.flngaccountkey = sk\_map.start\_account\_key

and timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_as\_tblaccount as a

on

(

timeline.flngaccountkey = a.flngaccountkey

and timeline.record\_effective\_timestamp between a.record\_effective\_timestamp and a.record\_expiry\_timestamp

) left join

crw\_as\_tblprofile as pp

on

(

pp.flngCustomerKey = a.flngCustomerKey

and pp.fintProfileNumber = a.fintParentProfileNumber

and timeline.record\_effective\_timestamp between pp.record\_effective\_timestamp and pp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_profile\_type as ppt

on

(

pp.fstrprofiletype = ppt.fstrprofiletype

and timeline.record\_effective\_timestamp between ppt.record\_effective\_timestamp and ppt.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_account\_std as nz\_std

on

(

a.flngdockey = nz\_std.flngdockey

and timeline.record\_effective\_timestamp between nz\_std.record\_effective\_timestamp and nz\_std.record\_expiry\_timestamp

) left join

crw\_as\_tblnamerecord as lgl

on

(

lgl.flngCustomerKey = a.flngCustomerKey

and lgl.derived\_Profile\_Number = a.fintProfileNumber

and lgl.derived\_Account\_Key = a.flngAccountKey

and lgl.derived\_name\_type= 'LGL'

and timeline.record\_effective\_timestamp between lgl.record\_effective\_timestamp and lgl.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= lgl.fdtmeffectivefrom

and days\_add(timeline.record\_effective\_timestamp ,-1)<= lgl.fdtmeffectiveto

and lgl.fblnactive = 1

) left join

crw\_as\_tblnamerecord as dba

on

(

dba.flngCustomerKey = a.flngCustomerKey

and dba.derived\_Profile\_Number = a.fintProfileNumber

and dba.derived\_Account\_Key = a.flngAccountKey

and dba.derived\_name\_type = 'DBA'

and timeline.record\_effective\_timestamp between dba.record\_effective\_timestamp and dba.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= dba.fdtmeffectivefrom

and days\_add(timeline.record\_effective\_timestamp ,-1)<= dba.fdtmeffectiveto

and dba.fblnactive = 1

) left join

crw\_ps\_tblcustomer as c

on

(

a.flngcustomerkey = c.flngcustomerkey

and timeline.record\_effective\_timestamp between c.record\_effective\_timestamp and c.record\_expiry\_timestamp

) left join

crw\_as\_lan\_account\_type as actyp

on

(

actyp.fstrAccountType = a.fstrAccountType

and timeline.record\_effective\_timestamp between actyp.record\_effective\_timestamp and actyp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_close\_reason as cr

on

(

cr.fstrreason = a.fstrClosureReason

and timeline.record\_effective\_timestamp between cr.record\_effective\_timestamp and cr.record\_expiry\_timestamp

) left join

crw\_as\_lan\_filing as ff

on

(

ff.fstrFiling = a.fstrFiling

and timeline.record\_effective\_timestamp between ff.record\_effective\_timestamp and ff.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_ipe as accipe

on

(

accipe.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accipe.record\_effective\_timestamp and accipe.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as ipexmp

on

(

ipexmp.fstrfield = 'NZ.ACCIPE.ExemptionReason'

and ipexmp.fstrlistitem = accipe.fstrExemptionReason

and timeline.record\_effective\_timestamp between ipexmp.record\_effective\_timestamp and ipexmp.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_income as accinc

on

(

accinc.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accinc.record\_effective\_timestamp and accinc.record\_expiry\_timestamp

) left join

crw\_as\_lannz\_inctax\_exmpt\_rsn as incxmp

on

(

incxmp.fstrExemptionReason = accinc.fstrExemptionReason

and timeline.record\_effective\_timestamp between incxmp.record\_effective\_timestamp and incxmp.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_exmpt as exmpt

on

(

exmpt.flngCustomerKey = a.flngCustomerKey

and exmpt.derived\_profile\_number= a.fintProfileNumber

and exmpt.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between exmpt.record\_effective\_timestamp and exmpt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between exmpt.fdtmcommence and exmpt.fdtmcease

and exmpt.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as lind

on

(

exmpt.fstrindicator = lind.fstrindicator

and timeline.record\_effective\_timestamp between lind.record\_effective\_timestamp and lind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as grnlst

on

(

grnlst.flngCustomerKey = a.flngCustomerKey

and grnlst.derived\_profile\_number= a.fintProfileNumber

and grnlst.derived\_account\_key= a.flngAccountKey

and grnlst.fstrIndicator = 'GRNLST'

and timeline.record\_effective\_timestamp between grnlst.record\_effective\_timestamp and grnlst.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between grnlst.fdtmcommence and grnlst.fdtmcease

and grnlst.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as frdred

on

(

frdred.flngCustomerKey = a.flngCustomerKey

and frdred.derived\_profile\_number = a.fintProfileNumber

and frdred.derived\_account\_key = a.flngAccountKey

and frdred.fstrIndicator = 'FRDRED'

and timeline.record\_effective\_timestamp between frdred.record\_effective\_timestamp and frdred.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between frdred.fdtmcommence and frdred.fdtmcease

and frdred.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as acchlt

on

(

acchlt.flngCustomerKey = a.flngCustomerKey

and acchlt.derived\_profile\_number= a.fintProfileNumber

and acchlt.derived\_account\_key= a.flngAccountKey

and acchlt.fstrIndicator = 'ACCHLT'

and timeline.record\_effective\_timestamp between acchlt.record\_effective\_timestamp and acchlt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between acchlt.fdtmcommence and acchlt.fdtmcease

and acchlt.fblnactive = 1

) left join

crw\_as\_tblindicator\_audact as audact

on

(

audact.flngCustomerKey = a.flngCustomerKey

and audact.derived\_profile\_number= a.fintProfileNumber

and audact.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between audact.record\_effective\_timestamp and audact.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between audact.fdtmcommence and audact.fdtmcease

and audact.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as alind

on

(

audact.fstrindicator = alind.fstrindicator

and timeline.record\_effective\_timestamp between alind.record\_effective\_timestamp and alind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_dontmail as dontmail

on

(

dontmail.flngCustomerKey = a.flngCustomerKey

and dontmail.derived\_profile\_number= a.fintProfileNumber

and dontmail.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between dontmail.record\_effective\_timestamp and dontmail.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between dontmail.fdtmcommence and dontmail.fdtmcease

and dontmail.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as mlind

on

(

dontmail.fstrindicator = mlind.fstrindicator

and timeline.record\_effective\_timestamp between mlind.record\_effective\_timestamp and mlind.record\_expiry\_timestamp

) left join

crw\_ps\_tblcustomerlevel as cl

on

(

cl.flngdockey = c.flngdockey

and timeline.record\_effective\_timestamp between cl.record\_effective\_timestamp and cl.record\_expiry\_timestamp

) left join

crw\_ps\_lancustomerlevel as cld

on

(

cld.fstrcustomerlevel = cl.fstrcustomerlevel

and timeline.record\_effective\_timestamp between cld.record\_effective\_timestamp and cld.record\_expiry\_timestamp

) left join

crw\_as\_rfrnz\_filing as ffp

on

(

a.fstrfiling = ffp.fstrfiling

)

where

a.flngaccountkey is not null

AND sk\_map.source = 'START'

### Start Account PAY

select

sk\_map.dip\_account\_sk as dip\_account\_pay\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

cast(timeline.record\_active\_flag as char(1)) as record\_active\_flag,

CAST('START' AS VARCHAR(10)) as account\_origin,

CAST(a.flngaccountkey AS VARCHAR(10)) as account\_key,

cast(ifnull(lpad(cast(sk\_map.ird\_no as varchar(10)),9,'0'),'Unknown') as varchar(10)) as account\_ird\_number,

cast(ifnull(CAST(p.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_profile\_number,

cast(ifnull(ppt.fstrdecode2, 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_type,

cast(ifnull(CAST(pp.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_number,

cast(ifnull(CAST(nz\_std.fintheritagelocationnumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE

WHEN c.fstrcustomertype = 'COM' THEN ifnull(lgl.fstrlistformatname, 'Unknown')

WHEN c.fstrcustomertype IN ('IND', 'CHD') THEN

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end

end AS VARCHAR(255)) as legal\_name,

CAST(ifnull(dba.fstrlistformatname,

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end)

AS VARCHAR(255)) as trading\_name,

cast(ifnull(actyp.fstrdecode1, 'Unknown') AS VARCHAR(30)) as account\_type,

cast(ifnull(a.fstraccounttype, 'Unknown') AS VARCHAR(30)) as account\_type\_code,

case when a.fdtmcommence between '1880-01-01' and now() then date\_trunc('day', a.fdtmcommence)

when a.fdtmcreated between '1880-01-01' and now() then date\_trunc('day', a.fdtmcreated)

else '1900-01-01' end as account\_start\_date,

if(a.fdtmcease between '1880-01-01' and now(), date\_trunc('day', a.fdtmcease), '9000-12-31') as account\_ceased\_date,

CAST(CASE a.fstrstatus

WHEN 'ACT' THEN 'Active'

WHEN 'CLS' THEN 'Closed'

WHEN 'SUS' THEN 'Suspended'

ELSE 'Unknown'

END AS VARCHAR(30)) as account\_status,

cast(ifnull(a.fstrstatus, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

cast(ifnull(cld.fstrdecode2, 'Unknown') AS VARCHAR(255)) as security\_level,

cast(ifnull(cl.fstrcustomerlevel, 'Unknown') AS VARCHAR(10)) as security\_level\_code,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN ifnull(cr.fstrdecode1, 'Unknown')

ELSE 'N/A' END AS VARCHAR(30)) as cessation\_reason,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN 'CLS'

ELSE 'N/A' END AS VARCHAR(10)) as cessation\_reason\_code,

cast(ifnull(ff.fstrdecode1, 'Unknown') AS VARCHAR(30)) as filing\_frequency,

cast(ifnull(a.fstrfiling, 'Unknown') AS VARCHAR(10)) as filing\_frequency\_code,

CAST('N/A' AS VARCHAR(30)) as exemption\_reason,

CAST('N/A' AS VARCHAR(30)) as exemption\_reason\_code,

CAST( if(exmpt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as exempted\_flag,

CAST(if(grnlst.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as green\_listed\_flag,

CAST(if(frdred.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as red\_listed\_flag,

CAST(if(acchlt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as account\_halt\_flag,

CAST (IF(audact.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST(ifnull(audact.fstrindicator, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_code,

cast(if(audact.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as audit\_case\_active\_flag,

CAST (IF(dontmail.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as stop\_mail\_desc,

CAST(IFNULL(dontmail.fstrindicator, 'Unknown') AS VARCHAR(10)) as stop\_mail\_code,

cast(if(dontmail.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as stop\_mail\_flag,

cast(if(accpso.fblnir56 = 1, 'Y', 'N') AS CHAR(1)) as ir56\_flag,

CAST(ifnull(et.fstrdecode2, 'Unknown') AS VARCHAR(30)) as filing\_option,

CAST(ifnull(accpso.fstrfilingoption, 'Unknown') AS VARCHAR(10)) as filing\_option\_code,

cast(if(accpso.fblnpaydayfiler = 1, 'Y', 'N') AS CHAR(1)) as payday\_filer\_flag,

IF(accpso.fdtmpaydayfilerstart BETWEEN '1880-01-01' AND NOW(), accpso.fdtmpaydayfilerstart, NULL) as payday\_filer\_start\_date,

CAST(IF(accpso.fblncse = 1, 'Y', 'N') AS CHAR(1)) as employer\_child\_support\_flag,

CAST(IF(accpso.fblnsle = 1, 'Y', 'N') AS CHAR(1)) as employer\_student\_loan\_flag,

CAST(IF(accpso.fblnkse = 1, 'Y', 'N') AS CHAR(1)) as kiwisaver\_employee\_dedn\_flag,

CAST(IF(accpso.fblnksr = 1, 'Y', 'N') AS CHAR(1)) as kiwisaver\_employer\_cntrbflag,

CAST(IF(accpso.fblnssc = 1, 'Y', 'N') AS CHAR(1)) as esct\_flag,

CAST(ifnull(eg.fstrdecode2, 'Unknown') AS VARCHAR(255)) as employer\_group,

CAST(ifnull(accpso.fstremployergroup, 'Unknown') AS VARCHAR(10)) as employer\_group\_code,

CAST( if(pdexmp.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as payday\_variation\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* ffp \*/

/\* cld \*/

/\* cl \*/

/\* pdexmp \*/

/\* eg \*/

/\* et \*/

/\* accpso \*/

/\* mlind \*/

/\* dontmail \*/

/\* alind \*/

/\* audact \*/

/\* acchlt \*/

/\* frdred \*/

/\* grnlst \*/

/\* lind \*/

/\* exmpt \*/

/\* incxmp \*/

/\* accinc \*/

/\* ipexmp \*/

/\* accipe \*/

/\* ff \*/

/\* cr \*/

/\* actyp \*/

/\* c \*/

/\* dba \*/

/\* lgl \*/

/\* nz\_std \*/

/\* ppt \*/

/\* pp \*/

/\* p \*/

/\* ird \*/

/\* a \*/

/\* sk\_map \*/

rfn\_as\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.flngaccountkey = sk\_map.start\_account\_key

and timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_as\_tblaccount as a

on

(

timeline.flngaccountkey = a.flngaccountkey

and a.fstrAccountType = 'PSO'

and timeline.record\_effective\_timestamp between a.record\_effective\_timestamp and a.record\_expiry\_timestamp

) left join

crw\_as\_tblid as ird

on

(

ird.flngCustomerKey = a.flngCustomerKey

and ird.fstridtype = 'IRD'

and timeline.record\_effective\_timestamp between ird.record\_effective\_timestamp and ird.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= ird.fdtmcommence and timeline.record\_effective\_timestamp < days\_add(ird.fdtmcease,-1)

and ird.fblnactive = 1

) left join

crw\_as\_tblprofile as p

on

(

a.flngcustomerkey = p.flngcustomerkey

and a.fintProfileNumber = p.fintProfileNumber

and timeline.record\_effective\_timestamp between p.record\_effective\_timestamp and p.record\_expiry\_timestamp

) left join

crw\_as\_tblprofile as pp

on

(

pp.flngCustomerKey = p.flngCustomerKey

and pp.fintProfileNumber = p.fintParentProfileNumber

and timeline.record\_effective\_timestamp between pp.record\_effective\_timestamp and pp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_profile\_type as ppt

on

(

pp.fstrprofiletype = ppt.fstrprofiletype

and timeline.record\_effective\_timestamp between ppt.record\_effective\_timestamp and ppt.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_account\_std as nz\_std

on

(

a.flngdockey = nz\_std.flngdockey

and timeline.record\_effective\_timestamp between nz\_std.record\_effective\_timestamp and nz\_std.record\_expiry\_timestamp

) left join

crw\_as\_tblnamerecord as lgl

on

(

lgl.flngCustomerKey = a.flngCustomerKey

and lgl.derived\_Profile\_Number = a.fintProfileNumber

and lgl.derived\_Account\_Key = a.flngAccountKey

and lgl.derived\_name\_type= 'LGL'

and timeline.record\_effective\_timestamp between lgl.record\_effective\_timestamp and lgl.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= lgl.fdtmeffectivefrom and timeline.record\_effective\_timestamp < days\_add(lgl.fdtmeffectiveto,-1)

and lgl.fblnactive = 1

) left join

crw\_as\_tblnamerecord as dba

on

(

dba.flngCustomerKey = a.flngCustomerKey

and dba.derived\_Profile\_Number = a.fintProfileNumber

and dba.derived\_Account\_Key = a.flngAccountKey

and dba.derived\_name\_type = 'DBA'

and timeline.record\_effective\_timestamp between dba.record\_effective\_timestamp and dba.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= dba.fdtmeffectivefrom and timeline.record\_effective\_timestamp < days\_add(dba.fdtmeffectiveto,-1)

and dba.fblnactive = 1

) left join

crw\_ps\_tblcustomer as c

on

(

a.flngcustomerkey = c.flngcustomerkey

and timeline.record\_effective\_timestamp between c.record\_effective\_timestamp and c.record\_expiry\_timestamp

) left join

crw\_as\_lan\_account\_type as actyp

on

(

actyp.fstrAccountType = a.fstrAccountType

and timeline.record\_effective\_timestamp between actyp.record\_effective\_timestamp and actyp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_close\_reason as cr

on

(

cr.fstrreason = a.fstrClosureReason

and timeline.record\_effective\_timestamp between cr.record\_effective\_timestamp and cr.record\_expiry\_timestamp

) left join

crw\_as\_lan\_filing as ff

on

(

ff.fstrFiling = a.fstrFiling

and timeline.record\_effective\_timestamp between ff.record\_effective\_timestamp and ff.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_ipe as accipe

on

(

accipe.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accipe.record\_effective\_timestamp and accipe.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as ipexmp

on

(

ipexmp.fstrfield = 'NZ.ACCIPE.ExemptionReason'

and ipexmp.fstrlistitem = accipe.fstrExemptionReason

and timeline.record\_effective\_timestamp between ipexmp.record\_effective\_timestamp and ipexmp.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_income as accinc

on

(

accinc.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accinc.record\_effective\_timestamp and accinc.record\_expiry\_timestamp

) left join

crw\_as\_lannz\_inctax\_exmpt\_rsn as incxmp

on

(

incxmp.fstrExemptionReason = accinc.fstrExemptionReason

and timeline.record\_effective\_timestamp between incxmp.record\_effective\_timestamp and incxmp.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_exmpt as exmpt

on

(

exmpt.flngCustomerKey = a.flngCustomerKey

and exmpt.derived\_profile\_number= a.fintProfileNumber

and exmpt.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between exmpt.record\_effective\_timestamp and exmpt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between exmpt.fdtmcommence and exmpt.fdtmcease

and exmpt.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as lind

on

(

exmpt.fstrindicator = lind.fstrindicator

and timeline.record\_effective\_timestamp between lind.record\_effective\_timestamp and lind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as grnlst

on

(

grnlst.flngCustomerKey = a.flngCustomerKey

and grnlst.derived\_profile\_number= a.fintProfileNumber

and grnlst.derived\_account\_key= a.flngAccountKey

and grnlst.fstrIndicator = 'GRNLST'

and timeline.record\_effective\_timestamp between grnlst.record\_effective\_timestamp and grnlst.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between grnlst.fdtmcommence and grnlst.fdtmcease

and grnlst.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as frdred

on

(

frdred.flngCustomerKey = a.flngCustomerKey

and frdred.derived\_profile\_number= a.fintProfileNumber

and frdred.derived\_account\_key= a.flngAccountKey

and frdred.fstrIndicator = 'FRDRED'

and timeline.record\_effective\_timestamp between frdred.record\_effective\_timestamp and frdred.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between frdred.fdtmcommence and frdred.fdtmcease

and frdred.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as acchlt

on

(

acchlt.flngCustomerKey = a.flngCustomerKey

and acchlt.derived\_profile\_number= a.fintProfileNumber

and acchlt.derived\_account\_key= a.flngAccountKey

and acchlt.fstrIndicator = 'ACCHLT'

and timeline.record\_effective\_timestamp between acchlt.record\_effective\_timestamp and acchlt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between acchlt.fdtmcommence and acchlt.fdtmcease

and acchlt.fblnactive = 1

) left join

crw\_as\_tblindicator\_audact as audact

on

(

audact.flngCustomerKey = a.flngCustomerKey

and audact.derived\_profile\_number= a.fintProfileNumber

and audact.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between audact.record\_effective\_timestamp and audact.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between audact.fdtmcommence and audact.fdtmcease

and audact.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as alind

on

(

audact.fstrindicator = alind.fstrindicator

and timeline.record\_effective\_timestamp between alind.record\_effective\_timestamp and alind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_dontmail as dontmail

on

(

dontmail.flngCustomerKey = a.flngCustomerKey

and dontmail.derived\_profile\_number= a.fintProfileNumber

and dontmail.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between dontmail.record\_effective\_timestamp and dontmail.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between dontmail.fdtmcommence and dontmail.fdtmcease

and dontmail.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as mlind

on

(

dontmail.fstrindicator = mlind.fstrindicator

and timeline.record\_effective\_timestamp between mlind.record\_effective\_timestamp and mlind.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_pso as accpso

on

(

accpso.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accpso.record\_effective\_timestamp and accpso.record\_expiry\_timestamp

) left join

crw\_as\_lannz\_employer\_type as et

on

(

et.fstrEmployerType = accpso.fstrFilingOption

and timeline.record\_effective\_timestamp between et.record\_effective\_timestamp and et.record\_expiry\_timestamp

) left join

crw\_as\_lannz\_employer\_group as eg

on

(

eg.fstrEmployerGroup = accpso.fstrEmployerGroup

and timeline.record\_effective\_timestamp between eg.record\_effective\_timestamp and eg.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as pdexmp

on

(

pdexmp.flngAccountKey = a.flngAccountKey

and pdexmp.flngCustomerKey = a.flngCustomerKey

and pdexmp.fintProfileNumber = a.fintProfileNumber

and pdexmp.fstrIndicator = 'PDEXMP'

and timeline.record\_effective\_timestamp between pdexmp.record\_effective\_timestamp and pdexmp.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between pdexmp.fdtmcommence and pdexmp.fdtmcease

) left join

crw\_ps\_tblcustomerlevel as cl

on

(

cl.flngdockey = c.flngdockey

and timeline.record\_effective\_timestamp between cl.record\_effective\_timestamp and cl.record\_expiry\_timestamp

) left join

crw\_ps\_lancustomerlevel as cld

on

(

cld.fstrcustomerlevel = cl.fstrcustomerlevel

and timeline.record\_effective\_timestamp between cld.record\_effective\_timestamp and cld.record\_expiry\_timestamp

) left join

crw\_as\_rfrnz\_filing as ffp

on

(

a.fstrfiling = ffp.fstrfiling

)

where

a.flngaccountkey is not null

AND sk\_map.source = 'START'

### Start Account PIE

select

sk\_map.dip\_account\_sk as dip\_account\_pie\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

CAST(timeline.record\_active\_flag AS CHAR(1)) as record\_active\_flag,

CAST('START' AS VARCHAR(10)) as account\_origin,

CAST(a.flngaccountkey AS VARCHAR(10)) as account\_key,

cast(ifnull(lpad(cast(sk\_map.ird\_no as varchar(10)),9,'0'),'Unknown') as varchar(10)) as account\_ird\_number,

cast(ifnull(CAST(p.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_profile\_number,

cast(ifnull(ppt.fstrdecode2, 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_type,

cast(ifnull(CAST(pp.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_number,

cast(ifnull(CAST(nz\_std.fintheritagelocationnumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE

WHEN c.fstrcustomertype = 'COM' THEN ifnull(lgl.fstrlistformatname, 'Unknown')

WHEN c.fstrcustomertype IN ('IND', 'CHD') THEN

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end

end AS VARCHAR(255)) as legal\_name,

CAST(ifnull(dba.fstrlistformatname,

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end)

AS VARCHAR(255)) as trading\_name,

cast(ifnull(actyp.fstrdecode1, 'Unknown') AS VARCHAR(30)) as account\_type,

cast(ifnull(a.fstraccounttype, 'Unknown') AS VARCHAR(30)) as account\_type\_code,

case when a.fdtmcommence between '1880-01-01' and now() then date\_trunc('day', a.fdtmcommence)

when a.fdtmcreated between '1880-01-01' and now() then date\_trunc('day', a.fdtmcreated)

else '1900-01-01' end as account\_start\_date,

if(a.fdtmcease between '1880-01-01' and now(), date\_trunc('day', a.fdtmcease), '9000-12-31') as account\_ceased\_date,

CAST(CASE a.fstrstatus

WHEN 'ACT' THEN 'Active'

WHEN 'CLS' THEN 'Closed'

WHEN 'SUS' THEN 'Suspended'

ELSE 'Unknown'

END AS VARCHAR(30)) as account\_status,

cast(ifnull(a.fstrstatus, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

cast(ifnull(cld.fstrdecode2, 'Unknown') AS VARCHAR(255)) as security\_level,

cast(ifnull(a.fstrstatus, 'Unknown') AS VARCHAR(10)) as security\_level\_code,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN ifnull(cr.fstrdecode1, 'Unknown')

ELSE 'N/A' END AS VARCHAR(30)) as cessation\_reason,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN 'CLS'

ELSE 'N/A' END AS VARCHAR(10)) as cessation\_reason\_code,

cast(ifnull(ff.fstrdecode1, 'Unknown') AS VARCHAR(30)) as filing\_frequency,

cast(ifnull(a.fstrfiling, 'Unknown') AS VARCHAR(10)) as filing\_frequency\_code,

CAST('N/A' AS VARCHAR(30)) as exemption\_reason,

CAST('N/A' AS VARCHAR(30)) as exemption\_reason\_code,

CAST( if(exmpt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as exempted\_flag,

CAST(if(grnlst.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as green\_listed\_flag,

CAST(if(frdred.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as red\_listed\_flag,

CAST(if(acchlt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as account\_halt\_flag,

CAST (IF(audact.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST(ifnull(audact.fstrindicator, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_code,

CAST(IF(audact.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as audit\_case\_active\_flag,

CAST (IF(dontmail.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as stop\_mail\_desc,

CAST(IFNULL(dontmail.fstrindicator, 'Unknown') AS VARCHAR(10)) as stop\_mail\_code,

CAST(if(dontmail.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as stop\_mail\_flag,

CAST(ifnull(pietyp.fstrdecode1, 'Unknown') AS VARCHAR(30)) as pie\_type,

CAST(ifnull(pieinfo.fstrpietype, 'Unknown') AS VARCHAR(10)) as pie\_type\_code,

CAST(ifnull(piefop.fstrdecode1, 'Unknown') AS VARCHAR(255)) as filing\_option,

CAST(ifnull(pieinfo.fstrfilingoption, 'Unknown') AS VARCHAR(10)) as filing\_option\_code,

CAST(ifnull(mp\_iap.dip\_description, 'Unknown') AS VARCHAR(50)) as attribution\_period,

CAST(ifnull(mp\_fdr.dip\_description, 'Unknown') AS VARCHAR(50)) as fdr\_calc\_period,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* mp\_fdr \*/

/\* mp\_iap \*/

/\* ffp \*/

/\* cld \*/

/\* cl \*/

/\* piefop \*/

/\* pietyp \*/

/\* pdexmp \*/

/\* pieinfo \*/

/\* mlind \*/

/\* dontmail \*/

/\* alind \*/

/\* audact \*/

/\* acchlt \*/

/\* frdred \*/

/\* grnlst \*/

/\* lind \*/

/\* exmpt \*/

/\* incxmp \*/

/\* accinc \*/

/\* ipexmp \*/

/\* accipe \*/

/\* ff \*/

/\* cr \*/

/\* actyp \*/

/\* c \*/

/\* dba \*/

/\* lgl \*/

/\* nz\_std \*/

/\* ppt \*/

/\* pp \*/

/\* p \*/

/\* ird \*/

/\* a \*/

/\* sk\_map \*/

rfn\_as\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.flngaccountkey = sk\_map.start\_account\_key

and timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_as\_tblaccount as a

on

(

timeline.flngaccountkey = a.flngaccountkey

and a.fstrAccountType = 'PIE'

and timeline.record\_effective\_timestamp between a.record\_effective\_timestamp and a.record\_expiry\_timestamp

) left join

crw\_as\_tblid as ird

on

(

ird.flngCustomerKey = a.flngCustomerKey

and ird.fstridtype = 'IRD'

and timeline.record\_effective\_timestamp between ird.record\_effective\_timestamp and ird.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= ird.fdtmcommence and timeline.record\_effective\_timestamp < days\_add(ird.fdtmcease,-1)

and ird.fblnactive = 1

) left join

crw\_as\_tblprofile as p

on

(

a.flngcustomerkey = p.flngcustomerkey

and a.fintProfileNumber = p.fintProfileNumber

and timeline.record\_effective\_timestamp between p.record\_effective\_timestamp and p.record\_expiry\_timestamp

) left join

crw\_as\_tblprofile as pp

on

(

pp.flngCustomerKey = p.flngCustomerKey

and pp.fintProfileNumber = p.fintParentProfileNumber

and timeline.record\_effective\_timestamp between pp.record\_effective\_timestamp and pp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_profile\_type as ppt

on

(

pp.fstrprofiletype = ppt.fstrprofiletype

and timeline.record\_effective\_timestamp between ppt.record\_effective\_timestamp and ppt.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_account\_std as nz\_std

on

(

a.flngdockey = nz\_std.flngdockey

and timeline.record\_effective\_timestamp between nz\_std.record\_effective\_timestamp and nz\_std.record\_expiry\_timestamp

) left join

crw\_as\_tblnamerecord as lgl

on

(

lgl.flngCustomerKey = a.flngCustomerKey

and lgl.derived\_Profile\_Number = a.fintProfileNumber

and lgl.derived\_Account\_Key = a.flngAccountKey

and lgl.derived\_name\_type= 'LGL'

and timeline.record\_effective\_timestamp between lgl.record\_effective\_timestamp and lgl.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= lgl.fdtmeffectivefrom and timeline.record\_effective\_timestamp < days\_add(lgl.fdtmeffectiveto,-1)

and lgl.fblnactive = 1

) left join

crw\_as\_tblnamerecord as dba

on

(

dba.flngCustomerKey = a.flngCustomerKey

and dba.derived\_Profile\_Number = a.fintProfileNumber

and dba.derived\_Account\_Key = a.flngAccountKey

and dba.derived\_name\_type = 'DBA'

and timeline.record\_effective\_timestamp between dba.record\_effective\_timestamp and dba.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= dba.fdtmeffectivefrom and timeline.record\_effective\_timestamp < days\_add(dba.fdtmeffectiveto,-1)

and dba.fblnactive = 1

) left join

crw\_ps\_tblcustomer as c

on

(

a.flngcustomerkey = c.flngcustomerkey

and timeline.record\_effective\_timestamp between c.record\_effective\_timestamp and c.record\_expiry\_timestamp

) left join

crw\_as\_lan\_account\_type as actyp

on

(

actyp.fstrAccountType = a.fstrAccountType

and timeline.record\_effective\_timestamp between actyp.record\_effective\_timestamp and actyp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_close\_reason as cr

on

(

cr.fstrreason = a.fstrClosureReason

and timeline.record\_effective\_timestamp between cr.record\_effective\_timestamp and cr.record\_expiry\_timestamp

) left join

crw\_as\_lan\_filing as ff

on

(

ff.fstrFiling = a.fstrFiling

and timeline.record\_effective\_timestamp between ff.record\_effective\_timestamp and ff.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_ipe as accipe

on

(

accipe.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accipe.record\_effective\_timestamp and accipe.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as ipexmp

on

(

ipexmp.fstrfield = 'NZ.ACCIPE.ExemptionReason'

and ipexmp.fstrlistitem = accipe.fstrExemptionReason

and timeline.record\_effective\_timestamp between ipexmp.record\_effective\_timestamp and ipexmp.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_income as accinc

on

(

accinc.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accinc.record\_effective\_timestamp and accinc.record\_expiry\_timestamp

) left join

crw\_as\_lannz\_inctax\_exmpt\_rsn as incxmp

on

(

incxmp.fstrExemptionReason = accinc.fstrExemptionReason

and timeline.record\_effective\_timestamp between incxmp.record\_effective\_timestamp and incxmp.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_exmpt as exmpt

on

(

exmpt.flngCustomerKey = a.flngCustomerKey

and exmpt.derived\_profile\_number= a.fintProfileNumber

and exmpt.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between exmpt.record\_effective\_timestamp and exmpt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between exmpt.fdtmcommence and exmpt.fdtmcease

and exmpt.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as lind

on

(

exmpt.fstrindicator = lind.fstrindicator

and timeline.record\_effective\_timestamp between lind.record\_effective\_timestamp and lind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as grnlst

on

(

grnlst.flngCustomerKey = a.flngCustomerKey

and grnlst.derived\_profile\_number= a.fintProfileNumber

and grnlst.derived\_account\_key= a.flngAccountKey

and grnlst.fstrIndicator = 'GRNLST'

and timeline.record\_effective\_timestamp between grnlst.record\_effective\_timestamp and grnlst.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between grnlst.fdtmcommence and grnlst.fdtmcease

and grnlst.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as frdred

on

(

frdred.flngCustomerKey = a.flngCustomerKey

and frdred.derived\_profile\_number= a.fintProfileNumber

and frdred.derived\_account\_key= a.flngAccountKey

and frdred.fstrIndicator = 'FRDRED'

and timeline.record\_effective\_timestamp between frdred.record\_effective\_timestamp and frdred.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between frdred.fdtmcommence and frdred.fdtmcease

and frdred.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as acchlt

on

(

acchlt.flngCustomerKey = a.flngCustomerKey

and acchlt.derived\_profile\_number= a.fintProfileNumber

and acchlt.derived\_account\_key= a.flngAccountKey

and acchlt.fstrIndicator = 'ACCHLT'

and timeline.record\_effective\_timestamp between acchlt.record\_effective\_timestamp and acchlt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between acchlt.fdtmcommence and acchlt.fdtmcease

and acchlt.fblnactive = 1

) left join

crw\_as\_tblindicator\_audact as audact

on

(

audact.flngCustomerKey = a.flngCustomerKey

and audact.derived\_profile\_number= a.fintProfileNumber

and audact.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between audact.record\_effective\_timestamp and audact.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between audact.fdtmcommence and audact.fdtmcease

and audact.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as alind

on

(

audact.fstrindicator = alind.fstrindicator

and timeline.record\_effective\_timestamp between alind.record\_effective\_timestamp and alind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_dontmail as dontmail

on

(

dontmail.flngCustomerKey = a.flngCustomerKey

and dontmail.derived\_profile\_number= a.fintProfileNumber

and dontmail.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between dontmail.record\_effective\_timestamp and dontmail.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between dontmail.fdtmcommence and dontmail.fdtmcease

and dontmail.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as mlind

on

(

dontmail.fstrindicator = mlind.fstrindicator

and timeline.record\_effective\_timestamp between mlind.record\_effective\_timestamp and mlind.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_pie\_info as pieinfo

on

(

pieinfo.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between pieinfo.record\_effective\_timestamp and pieinfo.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as pdexmp

on

(

pdexmp.flngAccountKey = a.flngAccountKey

and pdexmp.flngCustomerKey = a.flngCustomerKey

and pdexmp.fintProfileNumber = a.fintProfileNumber

and pdexmp.fstrIndicator = 'PDEXMP'

and timeline.record\_effective\_timestamp between pdexmp.record\_effective\_timestamp and pdexmp.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between pdexmp.fdtmcommence and pdexmp.fdtmcease

) left join

crw\_as\_lan\_list\_item as pietyp

on

(

pietyp.fstrfield = 'NZ.ACCPIE.PIEType'

and pietyp.fstrlistitem = pieinfo.fstrPIEType

and timeline.record\_effective\_timestamp between pietyp.record\_effective\_timestamp and pietyp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as piefop

on

(

piefop.fstrfield = 'NZ.ACCPIE.FilingOption'

and piefop.fstrlistitem = pieinfo.fstrFilingOption

and timeline.record\_effective\_timestamp between piefop.record\_effective\_timestamp and piefop.record\_expiry\_timestamp

) left join

crw\_ps\_tblcustomerlevel as cl

on

(

cl.flngdockey = c.flngdockey

and timeline.record\_effective\_timestamp between cl.record\_effective\_timestamp and cl.record\_expiry\_timestamp

) left join

crw\_ps\_lancustomerlevel as cld

on

(

cld.fstrcustomerlevel = cl.fstrcustomerlevel

and timeline.record\_effective\_timestamp between cld.record\_effective\_timestamp and cld.record\_expiry\_timestamp

) left join

crw\_as\_rfrnz\_filing as ffp

on

(

a.fstrfiling = ffp.fstrfiling

) left join

mp\_source\_code as mp\_iap

on

(

pieinfo.fstrattributionPeriod = mp\_iap.dip\_code\_value

AND mp\_iap.source\_system = 'FIRST'

AND mp\_iap.source\_tax\_type = 'PIE'

AND mp\_iap.source\_code\_name = 'INCOME\_ALLOCATION\_PERIOD'

AND timeline.record\_effective\_timestamp BETWEEN mp\_iap.record\_effective\_timestamp

AND mp\_iap.record\_expiry\_timestamp

) left join

mp\_source\_code as mp\_fdr

on

(

pieinfo.fstrFDRCalcPeriod = mp\_fdr.dip\_code\_value

AND mp\_fdr.source\_system = 'FIRST'

AND mp\_fdr.source\_tax\_type = 'PIE'

AND mp\_fdr.source\_code\_name = 'FDR\_CALCULATION\_PERIOD'

AND timeline.record\_effective\_timestamp BETWEEN mp\_fdr.record\_effective\_timestamp

AND mp\_fdr.record\_expiry\_timestamp

)

where

a.flngaccountkey is not null

AND sk\_map.source = 'START'

### Start Account INC

select

sk\_map.dip\_account\_sk as dip\_account\_inc\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

cast(timeline.record\_active\_flag as char(1)) as record\_active\_flag,

CAST('START' AS VARCHAR(10)) as account\_origin,

CAST(a.flngaccountkey AS VARCHAR(10)) as account\_key,

cast(ifnull(lpad(cast(sk\_map.ird\_no as varchar(10)),9,'0'),'Unknown') as varchar(10)) as account\_ird\_number,

cast(ifnull(CAST(p.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_profile\_number,

cast(ifnull(ppt.fstrdecode2, 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_type,

cast(ifnull(CAST(pp.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_number,

cast(ifnull(CAST(nz\_std.fintheritagelocationnumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE

WHEN c.fstrcustomertype = 'COM' THEN ifnull(lgl.fstrlistformatname, 'Unknown')

WHEN c.fstrcustomertype IN ('IND', 'CHD') THEN

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end

end AS VARCHAR(255)) as legal\_name,

CAST(ifnull(dba.fstrlistformatname,

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end)

AS VARCHAR(255)) as trading\_name,

cast(ifnull(actyp.fstrdecode1, 'Unknown') AS VARCHAR(30)) as account\_type,

cast(ifnull(a.fstraccounttype, 'Unknown') AS VARCHAR(30)) as account\_type\_code,

case when a.fdtmcommence between '1880-01-01' and now() then date\_trunc('day', a.fdtmcommence)

when a.fdtmcreated between '1880-01-01' and now() then date\_trunc('day', a.fdtmcreated)

else '1900-01-01' end as account\_start\_date,

if(a.fdtmcease between '1880-01-01' and now(), date\_trunc('day', a.fdtmcease), '9000-12-31') as account\_ceased\_date,

CAST(CASE a.fstrstatus

WHEN 'ACT' THEN 'Active'

WHEN 'CLS' THEN 'Closed'

WHEN 'SUS' THEN 'Suspended'

ELSE 'Unknown'

END AS VARCHAR(30)) as account\_status,

cast(ifnull(a.fstrstatus, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

cast(ifnull(cld.fstrdecode2, 'Unknown') AS VARCHAR(255)) as security\_level,

cast(ifnull(cl.fstrcustomerlevel, 'Unknown') AS VARCHAR(10)) as security\_level\_code,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN ifnull(cr.fstrdecode1, 'Unknown')

ELSE 'N/A' END AS VARCHAR(30)) as cessation\_reason,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN 'CLS'

ELSE 'N/A' END AS VARCHAR(10)) as cessation\_reason\_code,

cast(ifnull(ff.fstrdecode1, 'Unknown') AS VARCHAR(30)) as filing\_frequency,

cast(ifnull(a.fstrfiling, 'Unknown') AS VARCHAR(10)) as filing\_frequency\_code,

CAST( ifnull(incxmp.fstrdecode1, 'Unknown') AS VARCHAR(30)) as exemption\_reason,

CAST( ifnull(accinc.fstrexemptionreason, 'Unknown') AS VARCHAR(30)) as exemption\_reason\_code,

CAST( if(exmpt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as exempted\_flag,

CAST(if(grnlst.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as green\_listed\_flag,

CAST(if(frdred.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as red\_listed\_flag,

CAST(if(acchlt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as account\_halt\_flag,

CAST (IF(audact.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST(ifnull(audact.fstrindicator, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_code,

cast(if(audact.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as audit\_case\_active\_flag,

CAST (IF(dontmail.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as stop\_mail\_desc,

CAST(IFNULL(dontmail.fstrindicator, 'Unknown') AS VARCHAR(10)) as stop\_mail\_code,

cast(if(dontmail.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as stop\_mail\_flag,

CAST(CONCAT(MONTHNAME(accinc.fdtmbalancedate), ' ', CAST(DAY(accinc.fdtmbalancedate) AS VARCHAR(2))) AS VARCHAR(30)) as balance\_date,

cast(if(zrottr.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as zero\_tailored\_rate\_flag,

CAST(IF(accitn.fblntranstasman = 1, 'Y', 'N') AS CHAR(1)) as trans\_tasman\_flag,

cast(if(accitn.fblnagentnonresinsurer = 1, 'Y', 'N') AS CHAR(1)) as agent\_non\_res\_insurer\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* ffp \*/

/\* cld \*/

/\* cl \*/

/\* accitn \*/

/\* zrottr \*/

/\* pdexmp \*/

/\* mlind \*/

/\* dontmail \*/

/\* alind \*/

/\* audact \*/

/\* acchlt \*/

/\* frdred \*/

/\* grnlst \*/

/\* lind \*/

/\* exmpt \*/

/\* incxmp \*/

/\* accinc \*/

/\* ipexmp \*/

/\* accipe \*/

/\* ff \*/

/\* cr \*/

/\* actyp \*/

/\* c \*/

/\* dba \*/

/\* lgl \*/

/\* nz\_std \*/

/\* ppt \*/

/\* pp \*/

/\* p \*/

/\* ird \*/

/\* a \*/

/\* sk\_map \*/

rfn\_as\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.flngaccountkey = sk\_map.start\_account\_key

and timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_as\_tblaccount as a

on

(

timeline.flngaccountkey = a.flngaccountkey

and a.fstrAccountType IN ('IIT', 'ITN')

and timeline.record\_effective\_timestamp between a.record\_effective\_timestamp and a.record\_expiry\_timestamp

) left join

crw\_as\_tblid as ird

on

(

ird.flngCustomerKey = a.flngCustomerKey

and ird.fstridtype = 'IRD'

and timeline.record\_effective\_timestamp between ird.record\_effective\_timestamp and ird.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= ird.fdtmcommence and timeline.record\_effective\_timestamp < days\_add(ird.fdtmcease,-1)

and ird.fblnactive = 1

) left join

crw\_as\_tblprofile as p

on

(

a.flngcustomerkey = p.flngcustomerkey

and a.fintProfileNumber = p.fintProfileNumber

and timeline.record\_effective\_timestamp between p.record\_effective\_timestamp and p.record\_expiry\_timestamp

) left join

crw\_as\_tblprofile as pp

on

(

pp.flngCustomerKey = p.flngCustomerKey

and pp.fintProfileNumber = p.fintParentProfileNumber

and timeline.record\_effective\_timestamp between pp.record\_effective\_timestamp and pp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_profile\_type as ppt

on

(

pp.fstrprofiletype = ppt.fstrprofiletype

and timeline.record\_effective\_timestamp between ppt.record\_effective\_timestamp and ppt.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_account\_std as nz\_std

on

(

a.flngdockey = nz\_std.flngdockey

and timeline.record\_effective\_timestamp between nz\_std.record\_effective\_timestamp and nz\_std.record\_expiry\_timestamp

) left join

crw\_as\_tblnamerecord as lgl

on

(

lgl.flngCustomerKey = a.flngCustomerKey

and lgl.derived\_Profile\_Number = a.fintProfileNumber

and lgl.derived\_Account\_Key = a.flngAccountKey

and lgl.derived\_name\_type= 'LGL'

and timeline.record\_effective\_timestamp between lgl.record\_effective\_timestamp and lgl.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= lgl.fdtmeffectivefrom and timeline.record\_effective\_timestamp < days\_add(lgl.fdtmeffectiveto,-1)

and lgl.fblnactive = 1

) left join

crw\_as\_tblnamerecord as dba

on

(

dba.flngCustomerKey = a.flngCustomerKey

and dba.derived\_Profile\_Number = a.fintProfileNumber

and dba.derived\_Account\_Key = a.flngAccountKey

and dba.derived\_name\_type = 'DBA'

and timeline.record\_effective\_timestamp between dba.record\_effective\_timestamp and dba.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= dba.fdtmeffectivefrom and timeline.record\_effective\_timestamp < days\_add(dba.fdtmeffectiveto,-1)

and dba.fblnactive = 1

) left join

crw\_ps\_tblcustomer as c

on

(

a.flngcustomerkey = c.flngcustomerkey

and timeline.record\_effective\_timestamp between c.record\_effective\_timestamp and c.record\_expiry\_timestamp

) left join

crw\_as\_lan\_account\_type as actyp

on

(

actyp.fstrAccountType = a.fstrAccountType

and timeline.record\_effective\_timestamp between actyp.record\_effective\_timestamp and actyp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_close\_reason as cr

on

(

cr.fstrreason = a.fstrClosureReason

and timeline.record\_effective\_timestamp between cr.record\_effective\_timestamp and cr.record\_expiry\_timestamp

) left join

crw\_as\_lan\_filing as ff

on

(

ff.fstrFiling = a.fstrFiling

and timeline.record\_effective\_timestamp between ff.record\_effective\_timestamp and ff.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_ipe as accipe

on

(

accipe.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accipe.record\_effective\_timestamp and accipe.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as ipexmp

on

(

ipexmp.fstrfield = 'NZ.ACCIPE.ExemptionReason'

and ipexmp.fstrlistitem = accipe.fstrExemptionReason

and timeline.record\_effective\_timestamp between ipexmp.record\_effective\_timestamp and ipexmp.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_income as accinc

on

(

accinc.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accinc.record\_effective\_timestamp and accinc.record\_expiry\_timestamp

) left join

crw\_as\_lannz\_inctax\_exmpt\_rsn as incxmp

on

(

incxmp.fstrExemptionReason = accinc.fstrExemptionReason

and timeline.record\_effective\_timestamp between incxmp.record\_effective\_timestamp and incxmp.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_exmpt as exmpt

on

(

exmpt.flngCustomerKey = a.flngCustomerKey

and exmpt.derived\_profile\_number= a.fintProfileNumber

and exmpt.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between exmpt.record\_effective\_timestamp and exmpt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between exmpt.fdtmcommence and exmpt.fdtmcease

and exmpt.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as lind

on

(

exmpt.fstrindicator = lind.fstrindicator

and timeline.record\_effective\_timestamp between lind.record\_effective\_timestamp and lind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as grnlst

on

(

grnlst.flngCustomerKey = a.flngCustomerKey

and grnlst.derived\_profile\_number= a.fintProfileNumber

and grnlst.derived\_account\_key= a.flngAccountKey

and grnlst.fstrIndicator = 'GRNLST'

and timeline.record\_effective\_timestamp between grnlst.record\_effective\_timestamp and grnlst.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between grnlst.fdtmcommence and grnlst.fdtmcease

and grnlst.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as frdred

on

(

frdred.flngCustomerKey = a.flngCustomerKey

and frdred.derived\_profile\_number= a.fintProfileNumber

and frdred.derived\_account\_key= a.flngAccountKey

and frdred.fstrIndicator = 'FRDRED'

and timeline.record\_effective\_timestamp between frdred.record\_effective\_timestamp and frdred.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between frdred.fdtmcommence and frdred.fdtmcease

and frdred.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as acchlt

on

(

acchlt.flngCustomerKey = a.flngCustomerKey

and acchlt.derived\_profile\_number= a.fintProfileNumber

and acchlt.derived\_account\_key= a.flngAccountKey

and acchlt.fstrIndicator = 'ACCHLT'

and timeline.record\_effective\_timestamp between acchlt.record\_effective\_timestamp and acchlt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between acchlt.fdtmcommence and acchlt.fdtmcease

and acchlt.fblnactive = 1

) left join

crw\_as\_tblindicator\_audact as audact

on

(

audact.flngCustomerKey = a.flngCustomerKey

and audact.derived\_profile\_number= a.fintProfileNumber

and audact.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between audact.record\_effective\_timestamp and audact.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between audact.fdtmcommence and audact.fdtmcease

and audact.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as alind

on

(

audact.fstrindicator = alind.fstrindicator

and timeline.record\_effective\_timestamp between alind.record\_effective\_timestamp and alind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_dontmail as dontmail

on

(

dontmail.flngCustomerKey = a.flngCustomerKey

and dontmail.derived\_profile\_number= a.fintProfileNumber

and dontmail.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between dontmail.record\_effective\_timestamp and dontmail.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between dontmail.fdtmcommence and dontmail.fdtmcease

and dontmail.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as mlind

on

(

dontmail.fstrindicator = mlind.fstrindicator

and timeline.record\_effective\_timestamp between mlind.record\_effective\_timestamp and mlind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as pdexmp

on

(

pdexmp.flngAccountKey = a.flngAccountKey

and pdexmp.flngCustomerKey = a.flngCustomerKey

and pdexmp.fintProfileNumber = a.fintProfileNumber

and pdexmp.fstrIndicator = 'PDEXMP'

and timeline.record\_effective\_timestamp between pdexmp.record\_effective\_timestamp and pdexmp.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between pdexmp.fdtmcommence and pdexmp.fdtmcease

) left join

crw\_as\_tblindicator\_misc as zrottr

on

(

zrottr.flngAccountKey = a.flngAccountKey

and zrottr.flngCustomerKey = a.flngCustomerKey

and zrottr.fintProfileNumber = a.fintProfileNumber

and zrottr.fstrIndicator = 'ZROTTR'

and timeline.record\_effective\_timestamp between zrottr.record\_effective\_timestamp and zrottr.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between zrottr.fdtmcommence and zrottr.fdtmcease

) left join

crw\_as\_tblnz\_acc\_itn as accitn

on

(

accitn.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accitn.record\_effective\_timestamp and accitn.record\_expiry\_timestamp

) left join

crw\_ps\_tblcustomerlevel as cl

on

(

cl.flngdockey = c.flngdockey

and timeline.record\_effective\_timestamp between cl.record\_effective\_timestamp and cl.record\_expiry\_timestamp

) left join

crw\_ps\_lancustomerlevel as cld

on

(

cld.fstrcustomerlevel = cl.fstrcustomerlevel

and timeline.record\_effective\_timestamp between cld.record\_effective\_timestamp and cld.record\_expiry\_timestamp

) left join

crw\_as\_rfrnz\_filing as ffp

on

(

a.fstrfiling = ffp.fstrfiling

)

where

a.flngaccountkey is not null

AND sk\_map.source = 'START'

### Start Account GST

select

sk\_map.dip\_account\_sk as dip\_account\_gst\_sk,

sk\_map.dip\_account\_id,

timeline.record\_effective\_timestamp,

timeline.record\_expiry\_timestamp,

cast(timeline.record\_active\_flag as char(1)) as record\_active\_flag,

CAST('START' AS VARCHAR(10)) as account\_origin,

CAST(a.flngaccountkey AS VARCHAR(10)) as account\_key,

cast(ifnull(lpad(cast(sk\_map.ird\_no as varchar(10)),9,'0'),'Unknown') as varchar(10)) as account\_ird\_number,

cast(ifnull(CAST(p.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_profile\_number,

cast(ifnull(ppt.fstrdecode2, 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_type,

cast(ifnull(CAST(pp.fintprofilenumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as start\_parent\_profile\_number,

cast(ifnull(CAST(nz\_std.fintheritagelocationnumber AS VARCHAR), 'Unknown') AS VARCHAR(10)) as heritage\_location\_number,

CAST(CASE

WHEN c.fstrcustomertype = 'COM' THEN ifnull(lgl.fstrlistformatname, 'Unknown')

WHEN c.fstrcustomertype IN ('IND', 'CHD') THEN

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end

end AS VARCHAR(255)) as legal\_name,

CAST(ifnull(dba.fstrlistformatname,

case when lgl.fstrlastname is null THEN 'Unknown'

when lgl.fstrmiddlename is null THEN concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrlastname)

else concat(lgl.fstrtitle, ' ', lgl.fstrfirstname, ' ', lgl.fstrmiddlename, ' ', lgl.fstrlastname)

end)

AS VARCHAR(255)) as trading\_name,

cast(ifnull(actyp.fstrdecode1, 'Unknown') AS VARCHAR(30)) as account\_type,

cast(ifnull(a.fstraccounttype, 'Unknown') AS VARCHAR(30)) as account\_type\_code,

case when a.fdtmcommence between '1880-01-01' and now() then date\_trunc('day', a.fdtmcommence)

when a.fdtmcreated between '1880-01-01' and now() then date\_trunc('day', a.fdtmcreated)

else '1900-01-01' end as account\_start\_date,

if(a.fdtmcease between '1880-01-01' and now(), date\_trunc('day', a.fdtmcease), '9000-12-31') as account\_ceased\_date,

CAST(CASE a.fstrstatus

WHEN 'ACT' THEN 'Active'

WHEN 'CLS' THEN 'Closed'

WHEN 'SUS' THEN 'Suspended'

ELSE 'Unknown'

END AS VARCHAR(30)) as account\_status,

cast(ifnull(a.fstrstatus, 'Unknown') AS VARCHAR(10)) as account\_status\_code,

cast(ifnull(cld.fstrdecode2, 'Unknown') AS VARCHAR(255)) as security\_level,

cast(ifnull(cl.fstrcustomerlevel, 'Unknown') AS VARCHAR(10)) as security\_level\_code,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN ifnull(cr.fstrdecode1, 'Unknown')

ELSE 'N/A' END AS VARCHAR(30)) as cessation\_reason,

CAST(

CASE WHEN a.fstrstatus = 'CLS' THEN 'CLS'

ELSE 'N/A' END AS VARCHAR(10)) as cessation\_reason\_code,

cast(ifnull(ff.fstrdecode1, 'Unknown') AS VARCHAR(30)) as filing\_frequency,

cast(ifnull(a.fstrfiling, 'Unknown') AS VARCHAR(10)) as filing\_frequency\_code,

CAST('N/A' AS VARCHAR(30)) as exemption\_reason,

CAST('N/A' AS VARCHAR(30)) as exemption\_reason\_code,

CAST( if(exmpt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as exempted\_flag,

CAST(if(grnlst.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as green\_listed\_flag,

CAST(if(frdred.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as red\_listed\_flag,

CAST(if(acchlt.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as account\_halt\_flag,

CAST (IF(audact.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as audit\_case\_active\_desc,

CAST(ifnull(audact.fstrindicator, 'Unknown') AS VARCHAR(30)) as audit\_case\_active\_code,

cast(if(audact.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as audit\_case\_active\_flag,

CAST (IF(dontmail.fstrindicator IS NULL, 'Unknown', ifnull(alind.fstrdecode2, 'Unknown')) AS VARCHAR(30)) as stop\_mail\_desc,

CAST(IFNULL(dontmail.fstrindicator, 'Unknown') AS VARCHAR(10)) as stop\_mail\_code,

cast(if(dontmail.fstrindicator IS NULL, 'N', 'Y') AS CHAR(1)) as stop\_mail\_flag,

CAST(

CASE gsti.fstrlastaccountingbasis

WHEN 'HYB' THEN 'Hybrid'

WHEN 'INV' THEN 'Invoice'

WHEN 'PYM' THEN 'Payments'

END AS VARCHAR(10)) as accounting\_basis,

CAST(ifnull(gstregc.fstrdecode1, 'Unknown') AS VARCHAR(10)) as registration\_circumstance,

cast(if(gsti.fblnpricesincgst = 1, 'Y', 'N') AS CHAR(1)) as included\_in\_prices\_flag,

cast(if(gsti.fblnexemptsupplies = 1, 'Y', 'N') AS CHAR(1)) as exempt\_supplies\_flag,

cast(if(gsti.fblnexporter = 1, 'Y', 'N') AS CHAR(1)) as exporter\_flag,

cast(if(gsti.fblnimporter = 1, 'Y', 'N') AS CHAR(1)) as importer\_flag,

cast(if(gsti.fblnunpolicedfiler= 1, 'Y', 'N') AS CHAR(1)) as unpoliced\_filer\_flag,

cast(if(gsti.fblnselfinvoicing= 1, 'Y', 'N') AS CHAR(1)) as self\_invoicer\_flag,

cast(if(gsti.fblnhospicefiler= 1, 'Y', 'N') AS CHAR(1)) as hospice\_filer\_flag,

CAST(IF(gsti.fblnhospicefiler = 1, ifnull(gsti.fstrhospicetype, 'Unknown'), 'N/A') AS VARCHAR(10)) as hospice\_type,

CAST(IF(gsti.fblnhospicefiler = 1, ifnull(gsthspc.fstrdecode1, 'Unknown'), 'N/A') AS VARCHAR(30)) as hospice\_type\_desc,

cast(if(gsti.fblngstonelectronicservices= 1, 'Y', 'N') AS CHAR(1)) as

CAST(IF(gsti.fblngstonelectronicservices = 1, ifnull(gstgor.fstrdecode1, 'Unknown'), 'N/A') AS VARCHAR(30)) as gst\_on\_remote\_srvcs\_type,

CAST(IF(gsti.fblngstonelectronicservices = 1, ifnull(gsti.fstrgorstype, 'Unknown'), 'N/A') AS VARCHAR(30)) as gst\_on\_remote\_srvcs\_code,

CAST(IF(gsti.fblnzeroratedfinancialservices = 1, 'Y', 'N') AS CHAR(1)) as gst\_zero\_rated\_fin\_srv\_flag,

cast(if(gsti.fblnnonresbusinessclaimant = 1, 'Y', 'N') AS CHAR(1)) as gst\_non\_resident\_claimant\_flag,

cast(if(gstreg.fblntaxableactivity = 1, 'Y', 'N') AS CHAR(1)) as taxable\_activity\_flag,

timeline.insert\_object\_run\_key,

timeline.update\_object\_run\_key

from

/\* ffp \*/

/\* cld \*/

/\* cl \*/

/\* gstreg \*/

/\* gstgor \*/

/\* gstregc \*/

/\* gsthspc \*/

/\* gsti \*/

/\* mlind \*/

/\* dontmail \*/

/\* alind \*/

/\* audact \*/

/\* acchlt \*/

/\* frdred \*/

/\* grnlst \*/

/\* lind \*/

/\* exmpt \*/

/\* ipexmp \*/

/\* accipe \*/

/\* ff \*/

/\* cr \*/

/\* actyp \*/

/\* c \*/

/\* dba \*/

/\* lgl \*/

/\* nz\_std \*/

/\* ppt \*/

/\* pp \*/

/\* p \*/

/\* a \*/

/\* sk\_map \*/

rfn\_as\_timeline\_f as timeline left join

rfn\_account\_sk\_map as sk\_map

on

(

timeline.flngaccountkey = sk\_map.start\_account\_key

and timeline.record\_effective\_timestamp = sk\_map.record\_effective\_timestamp

) left join

crw\_as\_tblaccount as a

on

(

timeline.flngaccountkey = a.flngaccountkey

and a.fstrAccountType = 'GST'

and timeline.record\_effective\_timestamp between a.record\_effective\_timestamp and a.record\_expiry\_timestamp

) left join

crw\_as\_tblprofile as p

on

(

a.flngcustomerkey = p.flngcustomerkey

and a.fintProfileNumber = p.fintProfileNumber

and timeline.record\_effective\_timestamp between p.record\_effective\_timestamp and p.record\_expiry\_timestamp

) left join

crw\_as\_tblprofile as pp

on

(

pp.flngCustomerKey = p.flngCustomerKey

and pp.fintProfileNumber = p.fintParentProfileNumber

and timeline.record\_effective\_timestamp between pp.record\_effective\_timestamp and pp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_profile\_type as ppt

on

(

pp.fstrprofiletype = ppt.fstrprofiletype

and timeline.record\_effective\_timestamp between ppt.record\_effective\_timestamp and ppt.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_account\_std as nz\_std

on

(

a.flngdockey = nz\_std.flngdockey

and timeline.record\_effective\_timestamp between nz\_std.record\_effective\_timestamp and nz\_std.record\_expiry\_timestamp

) left join

crw\_as\_tblnamerecord as lgl

on

(

lgl.flngCustomerKey = a.flngCustomerKey

and lgl.derived\_Profile\_Number = a.fintProfileNumber

and lgl.derived\_Account\_Key = a.flngAccountKey

and lgl.derived\_name\_type= 'LGL'

and timeline.record\_effective\_timestamp between lgl.record\_effective\_timestamp and lgl.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= lgl.fdtmeffectivefrom

and days\_add(timeline.record\_effective\_timestamp ,-1)<= lgl.fdtmeffectiveto

and lgl.fblnactive = 1

) left join

crw\_as\_tblnamerecord as dba

on

(

dba.flngCustomerKey = a.flngCustomerKey

and dba.derived\_Profile\_Number = a.fintProfileNumber

and dba.derived\_Account\_Key = a.flngAccountKey

and dba.derived\_name\_type = 'DBA'

and timeline.record\_effective\_timestamp between dba.record\_effective\_timestamp and dba.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp >= dba.fdtmeffectivefrom

and days\_add(timeline.record\_effective\_timestamp ,-1)<= dba.fdtmeffectiveto

and dba.fblnactive = 1

) left join

crw\_ps\_tblcustomer as c

on

(

a.flngcustomerkey = c.flngcustomerkey

and timeline.record\_effective\_timestamp between c.record\_effective\_timestamp and c.record\_expiry\_timestamp

) left join

crw\_as\_lan\_account\_type as actyp

on

(

actyp.fstrAccountType = a.fstrAccountType

and timeline.record\_effective\_timestamp between actyp.record\_effective\_timestamp and actyp.record\_expiry\_timestamp

) left join

crw\_as\_lan\_close\_reason as cr

on

(

cr.fstrreason = a.fstrClosureReason

and timeline.record\_effective\_timestamp between cr.record\_effective\_timestamp and cr.record\_expiry\_timestamp

) left join

crw\_as\_lan\_filing as ff

on

(

ff.fstrFiling = a.fstrFiling

and timeline.record\_effective\_timestamp between ff.record\_effective\_timestamp and ff.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_ipe as accipe

on

(

accipe.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between accipe.record\_effective\_timestamp and accipe.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as ipexmp

on

(

ipexmp.fstrfield = 'NZ.ACCIPE.ExemptionReason'

and ipexmp.fstrlistitem = accipe.fstrExemptionReason

and timeline.record\_effective\_timestamp between ipexmp.record\_effective\_timestamp and ipexmp.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_exmpt as exmpt

on

(

exmpt.flngCustomerKey = a.flngCustomerKey

and exmpt.derived\_profile\_number= a.fintProfileNumber

and exmpt.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between exmpt.record\_effective\_timestamp and exmpt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between exmpt.fdtmcommence and exmpt.fdtmcease

and exmpt.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as lind

on

(

exmpt.fstrindicator = lind.fstrindicator

and timeline.record\_effective\_timestamp between lind.record\_effective\_timestamp and lind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_misc as grnlst

on

(

grnlst.flngCustomerKey = a.flngCustomerKey

and grnlst.derived\_profile\_number= a.fintProfileNumber

and grnlst.derived\_account\_key= a.flngAccountKey

and grnlst.fstrIndicator = 'GRNLST'

and timeline.record\_effective\_timestamp between grnlst.record\_effective\_timestamp and grnlst.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between grnlst.fdtmcommence and grnlst.fdtmcease

and grnlst.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as frdred

on

(

frdred.flngCustomerKey = a.flngCustomerKey

and frdred.derived\_profile\_number= a.fintProfileNumber

and frdred.derived\_account\_key= a.flngAccountKey

and frdred.fstrIndicator = 'FRDRED'

and timeline.record\_effective\_timestamp between frdred.record\_effective\_timestamp and frdred.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between frdred.fdtmcommence and frdred.fdtmcease

and frdred.fblnactive = 1

) left join

crw\_as\_tblindicator\_misc as acchlt

on

(

acchlt.flngCustomerKey = a.flngCustomerKey

and acchlt.derived\_profile\_number= a.fintProfileNumber

and acchlt.derived\_account\_key= a.flngAccountKey

and acchlt.fstrIndicator = 'ACCHLT'

and timeline.record\_effective\_timestamp between acchlt.record\_effective\_timestamp and acchlt.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between acchlt.fdtmcommence and acchlt.fdtmcease

and acchlt.fblnactive = 1

) left join

crw\_as\_tblindicator\_audact as audact

on

(

audact.flngCustomerKey = a.flngCustomerKey

and audact.derived\_profile\_number= a.fintProfileNumber

and audact.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between audact.record\_effective\_timestamp and audact.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between audact.fdtmcommence and audact.fdtmcease

and audact.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as alind

on

(

audact.fstrindicator = alind.fstrindicator

and timeline.record\_effective\_timestamp between alind.record\_effective\_timestamp and alind.record\_expiry\_timestamp

) left join

crw\_as\_tblindicator\_dontmail as dontmail

on

(

dontmail.flngCustomerKey = a.flngCustomerKey

and dontmail.derived\_profile\_number= a.fintProfileNumber

and dontmail.derived\_account\_key= a.flngAccountKey

and timeline.record\_effective\_timestamp between dontmail.record\_effective\_timestamp and dontmail.record\_expiry\_timestamp

and timeline.record\_effective\_timestamp between dontmail.fdtmcommence and dontmail.fdtmcease

and dontmail.fblnactive = 1

) left join

crw\_ps\_lan\_indicator as mlind

on

(

dontmail.fstrindicator = mlind.fstrindicator

and timeline.record\_effective\_timestamp between mlind.record\_effective\_timestamp and mlind.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_acc\_gst\_info as gsti

on

(

gsti.flngdockey = a.flngdockey

and timeline.record\_effective\_timestamp between gsti.record\_effective\_timestamp and gsti.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as gsthspc

on

(

gsthspc.fstrfield = 'NZ.RTNGST.Hospice'

and gsthspc.fstrlistitem = gsti.fstrHospiceType

and timeline.record\_effective\_timestamp between gsthspc.record\_effective\_timestamp and gsthspc.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as gstregc

on

(

gstregc.fstrfield = 'NZ.ACCGST.RegCircumstance'

and gstregc.fstrlistitem = gsti.fstrRegCircumstance

and timeline.record\_effective\_timestamp between gstregc.record\_effective\_timestamp and gstregc.record\_expiry\_timestamp

) left join

crw\_as\_lan\_list\_item as gstgor

on

(

gstgor.fstrfield = 'NZ.ACCGST.GORSType'

and gstgor.fstrlistitem = gsti.fstrGORSType

and timeline.record\_effective\_timestamp between gstgor.record\_effective\_timestamp and gstgor.record\_expiry\_timestamp

) left join

crw\_as\_tblnz\_gst\_reg as gstreg

on

(

gstreg.flngDocKey = a.flngDocKey

and timeline.record\_effective\_timestamp between gstreg.record\_effective\_timestamp and gstreg.record\_expiry\_timestamp

) left join

crw\_ps\_tblcustomerlevel as cl

on

(

cl.flngdockey = c.flngdockey

and timeline.record\_effective\_timestamp between cl.record\_effective\_timestamp and cl.record\_expiry\_timestamp

) left join

crw\_ps\_lancustomerlevel as cld

on

(

cld.fstrcustomerlevel = cl.fstrcustomerlevel

and timeline.record\_effective\_timestamp between cld.record\_effective\_timestamp and cld.record\_expiry\_timestamp

) left join

crw\_as\_rfrnz\_filing as ffp

on

(

a.fstrfiling = ffp.fstrfiling

)

where

a.flngaccountkey is not null

AND sk\_map.source = 'START'

### Derived Table: crw\_as\_tblnamerecord

select

nr.flngCustomerKey,

a.fintprofilenumber, -- derived\_profile\_number

a.flngaccountkey, -- derived\_account\_key

(case when nr.fintprofilenumber = 1 then 0

when nr.fintprofilenumber = a.fintparentprofilenumber then 1

when nr.fintprofilenumber = a.fintprofilenumber then 2

end) derived\_Level, -- derived\_level

(case when nr.fstrNameType = ‘DBACST’ then ‘DBA’

else nr.fstrNameType

end) derived\_name\_type, -- derived\_name\_type

nr.\*

from app\_tblnamerecord nr

LEFT OUTER JOIN app\_tblaccount a

on nr.flngCustomerKey = a.flngCustomerKey

and nr.fintprofilenumber in (1,a.fintprofilenumber,a.fintparentprofilenumber)

### Derived Table: crw\_as\_tblindicator

select

a.fintprofilenumber, -- derived\_profile\_number

a.flngaccountkey, -- derived\_account\_key

i.\*

from app\_tblindicator i

INNER JOIN crw\_as\_tblaccount a

on i.flngCustomerKey = a.flngCustomerKey

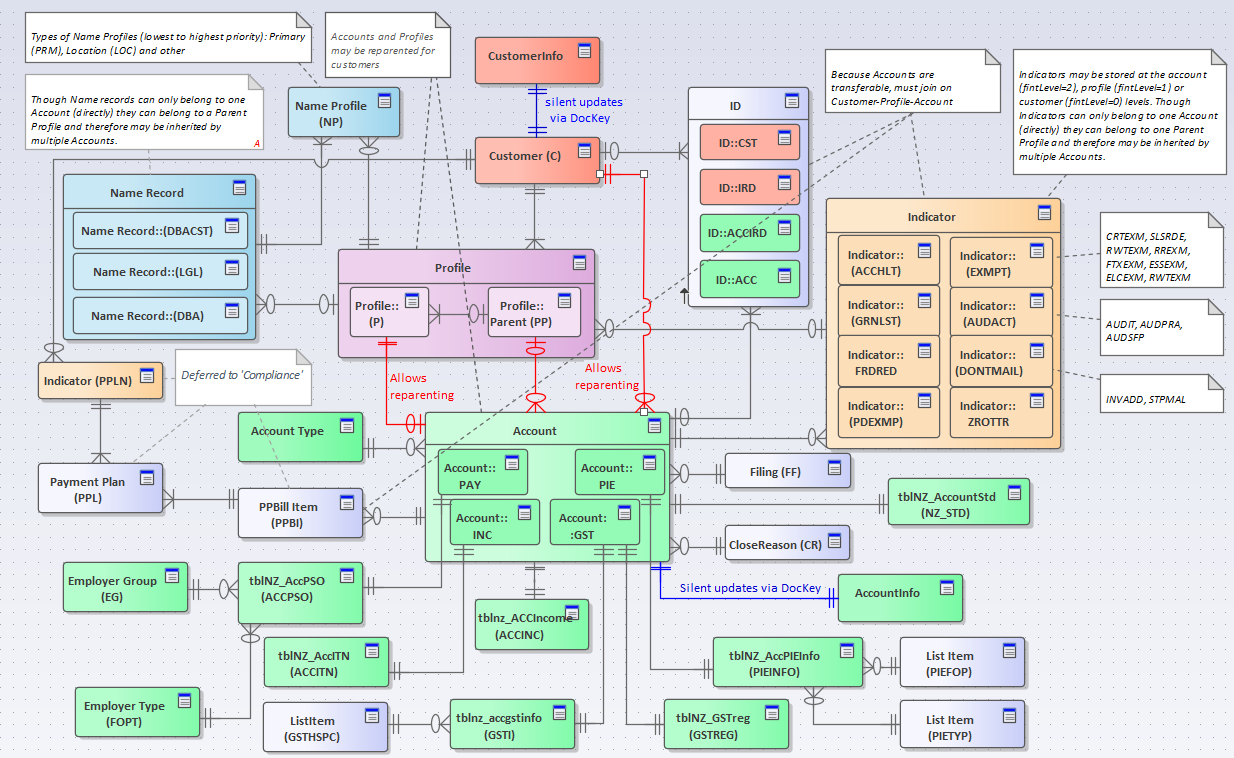
and i.flngAccountKey = (case when i.fintlevel = 2 then a.flngAccountKey else 0 end)

and i.fintprofilenumber = (case when i.fintlevel = 2 then a.fintProfileNumber

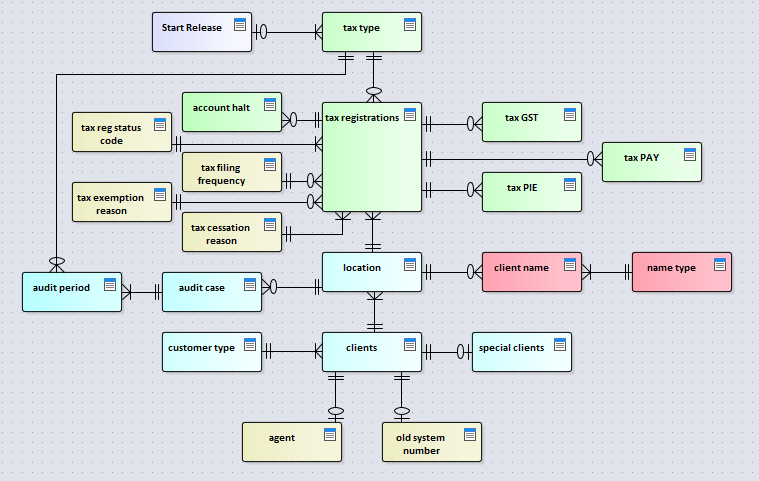
when i.fintlevel = 1 then a.fintParentProfileNumber

when i.fintlevel = 0 then 0 end)

# Appendix G – Conceptual Data Model – Start Accounts



# Appendix H - Conceptual Data Model – EDW Accounts



# Appendix I – EDW-Start Code Mappings

### Account (Tax) Types

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | TAX\_TYPE | TAX\_TYPE\_CODE |

| **Source** | | | | **DIP** | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| AIL |  | Ap Issuer Levy |  | AIL | Approved Issuer Levy |
| CPR |  | Cust Parent |  | CPR | Receiving Carer |
| DWT |  | Ord Div PAYE |  | DWT | Dividend Withholding Tax |
| FAM |  | (WFFTC) |  | FAM | Working for Families |
| FBT |  | Fringe Benefit |  | FBT | Fringe Benefit Tax |
| GMD |  | Gaming Duty |  | GMD | Gaming Machine Duty |
| GST |  | GOODS&SERVICES |  | GST | Goods and Services Tax |
| INC |  | INCOME TAX | IND | IIT | Income Tax Individuals |
| INC |  | INCOME TAX | ORG | ITN | Income Tax Non-Individuals |
| IPE |  | IP Exempt Recp |  | IPE | Withholding Disbursements Account |
| IPS |  | Interest PAYE |  | IPS | RWT on Interest |
| KSS |  | MEMBER ACCOUNT |  | KSS | KiwiSaver Member |
| NCP |  | Non-Cust Parnt |  | NCP | Liable Parent |
| NRT |  | Non Res W/Tax |  | NRT | Non-Resident Withholding Tax |
| PAY |  | Tax Deductions |  | PSO | Employment Activities |
| PIE |  | PIE TAX |  | PIE | Portfolio Investment Entity |
| PPL |  | PdParental Lve |  | PPL | Paid Parental Leave |
| REB |  | REBATE CLAIM |  | REB | Donation Tax Credit |
| RLT |  | RES LAND W/H |  | RLT | Residential Land Withholding Tax |
| RWT |  | Res W/H (Spec) |  | RWT | Resident Withholding Tax |
| SLS |  | Student Loan |  | SLS | Student Loans |

Notes

1. Because there must only be one instance of an Account (Tax) Type at a location for a Party, multiple EDW tax types must not be mapped to a Start account type. In other words, though data conversion has merged CSE, SSC, SLE, WPE, WPN and PAY Tax Types from EDW into PSO accounts (employment activities) in Start, it is not possible to ‘merge’ these tax types over time. Therefore, all ‘secondary’ EDW tax types must retain their original code value and description.
2. Tax types KSE and KSR are expected to be migrated to Start in R4. These will also be mapped to PSO accounts (employment activities) in Start.
3. Tax Types ACC, CAR, SEA and SHR have been deprecated and will not be in Start.

### Cessation Reasons

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | CESSATION\_REASON | CESSATION\_REASON |

| **Source** | | | | **DIP** | |
| --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| 01 |  | Exempt status change |  | EXMCHG | Exempt Status Change |
| 02 |  | $2m t/o not achieved |  | NOTURN | Turnover Not Achieved |
| 03 |  | Cert issued in error |  | NOELIG | No Longer Eligible |
| 05 |  | Business/Org ceased |  | NOELIG | No Longer Eligible |
| 06 |  | Limited exemption |  | NOELIG | No Longer Eligible |
| AC | GST | Amalamating Company |  | AMGENT | Amalgamating Entity |
| AC | CSE | Amalamating Company |  | NOPSO | No longer employing staff |
| AC | SLE | Amalamating Company |  | NOPSO | No longer employing staff |
| AC | SSC | Amalamating Company |  | NOPSO | No longer employing staff |
| AC | WPE | Amalamating Company |  | NOPSO | No longer employing staff |
| AC | WPN | Amalamating Company |  | NOPSO | No longer employing staff |
| AC | DWT | Amalamating Company |  | NOLGLI | No Longer Liable |
| AC | FBA | Amalamating Company |  | NOFBT | Not providing fringe benefits |
| AC | FBI | Amalamating Company |  | NOFBT | Not providing fringe benefits |
| AC | FBT | Amalamating Company |  | NOFBT | Not providing fringe benefits |
| AC |  | Amalamating Company |  | NOTXAC | Activity ceased |
| BC |  | Business/Org. Ceased |  | NOTXAC | Activity ceased |
| BN |  | Bankruptcy |  | BNKRPT | Insolvency |
| BR |  | BREACH |  | BREACH | Breach |
| CA |  | CAS-SYSTEM ONLY |  | NOTXAC | Activity ceased |
| CB |  | Change of business |  | NOTXAC | Activity ceased |
| CE | FBA | Ceased employing |  | NOFBT | Not providing fringe benefits |
| CE | FBI | Ceased employing |  | NOFBT | Not providing fringe benefits |
| CE | FBT | Ceased employing |  | NOFBT | Not providing fringe benefits |
| CE |  | Ceased employing |  | NOPSO | No longer employing staff |
| CL | GST | Closed |  | CLOSED | Closed |
| CL | PAY | Closed |  | NOPSO | No longer employing staff |
| CL | SLE | Closed |  | NOPSO | No longer employing staff |
| CL | SSC | Closed |  | NOPSO | No longer employing staff |
| CL |  | Closed |  | NOTXAC | Activity ceased |
| CX |  | Conversion Error |  | NOTXAC | Activity ceased |
| DC |  | Deceased |  | DECEAS | Deceased |
| DL |  | DEC D/LOSS GRDNSHIP |  | DL | Dead or Loss of Guardianship |
| DP |  | Duplicate IRD |  | CSTDUP | Duplicate IRD number |
| EC |  | ELECTION CANCELLED |  | NOTXAC | Activity ceased |
| EE |  | Employee |  | NOTXAC | Activity ceased |
| EM |  | Emigration |  | EMIG | Emigrated |
| ER |  | Loaded in Error |  | ERROR | Registered in Error |
| FC |  | Filing Option Chg d |  | NOLGLI | No Longer Liable |
| FR | GST | Fraud |  | FRAUD | Fraud |
| FR | FAM | Fraud |  | NOENTL | No Remaining Entitlements |
| FR |  | Fraud |  | NOTXAC | Activity ceased |
| GC |  | Consol Grp Ceased |  | NOTXAC | Activity ceased |
| IN |  | Incapacitated |  | NOLGLI | No Longer Liable |
| JG |  | JOINED GROUP |  | NOTXAC | Activity ceased |
| LC |  | Licence Cancelled |  | NOTXAC | Activity ceased |
| LP |  | Loan Repaid |  | NOTXAC | Activity ceased |
| LT |  | Low taxable supplies |  | NOTXAC | Activity ceased |
| MS |  | Missing |  | MISSNG | Missing |
| NE | DWT | No Longer Eligible |  | NOLGLI | No Longer Liable |
| NE | QCT | No Longer Eligible |  | STROFF | Removed from the register |
| NE |  | No Longer Eligible |  | NOELIG | No Longer Eligible |
| NL | NCP | No longer liable |  | NOTXAC | Activity ceased |
| NL | AIL | No longer liable |  | NOTXAC | Activity ceased |
| NL | GMD | No longer liable |  | NOTXAC | Activity ceased |
| NL | ICA | No longer liable |  | NOTXAC | Activity ceased |
| NL | INC | No longer liable |  | NOTXAC | Activity ceased |
| NL | NRT | No longer liable |  | NOTXAC | Activity ceased |
| NL | WPN | No longer liable |  | NOPSO | No longer employing staff |
| NL | CSE | No longer liable |  | NOPSO | No longer employing staff |
| NL | SLE | No longer liable |  | NOPSO | No longer employing staff |
| NL | PAY | No longer liable |  | NOPSO | No longer employing staff |
| NL | WPE | No longer liable |  | NOPSO | No longer employing staff |
| NL | QCT | No longer liable |  | STROFF | Removed from the register |
| NL |  | No longer liable |  | NOLGLI | No Longer Liable |
| NT |  | No taxable activity |  | NOTXAC | Activity ceased |
| OO |  | OPT-OUT |  | OPTOUT | Opt Out |
| OP |  | OPT OUT-SYSTEM ONLY |  | OPTOUT | Opt Out |
| OS | PAY | Multiple OSN numbers |  | NOPSO | No longer employing staff |
| OS | ICA | Multiple OSN numbers |  | STROFF | Removed from the register |
| OS | DWT | Multiple OSN numbers |  | NOLGLI | No Longer Liable |
| OS | FBT | Multiple OSN numbers |  | NOLGLI | No Longer Liable |
| OS | WPN | Multiple OSN numbers |  | NOPSO | No longer employing staff |
| OS | IPE | Multiple OSN numbers |  | NOELIG | No Longer Eligible |
| OS | SSC | Multiple OSN numbers |  | NOPSO | No longer employing staff |
| OS | IPS | Multiple OSN numbers |  | NOLGLI | No Longer Liable |
| OS |  | Multiple OSN numbers |  | NOTXAC | Activity ceased |
| OV | FAM | Overseas |  | NOENTL | No Remaining Entitlements |
| OV |  | Overseas |  | NOTXAC | Activity ceased |
| PA |  | PROVIDER-SYSTEM ONLY |  | NOTXAC | Activity ceased |
| PC |  | Payment Completed |  | PC | Payment Period Completed |
| RC |  | UNKNOWN |  | NOTXAC | Activity ceased |
| RD |  | Retired |  | NOTXAC | Activity ceased |
| SO | GST | Struck Off/Dissolved |  | STRUCK | Removed/Dissolved |
| SO | WPN | Struck Off/Dissolved |  | NOPSO | No longer employing staff |
| SO | DWT | Struck Off/Dissolved |  | NOLGLI | No Longer Liable |
| SO | FBT | Struck Off/Dissolved |  | NOFBT | Not providing fringe benefits |
| SO | FBI | Struck Off/Dissolved |  | NOFBT | Not providing fringe benefits |
| SO | FBA | Struck Off/Dissolved |  | NOFBT | Not providing fringe benefits |
| SO | SSC | Struck Off/Dissolved |  | NOPSO | No longer employing staff |
| SO | PAY | Struck Off/Dissolved |  | NOPSO | No longer employing staff |
| SO | CSE | Struck Off/Dissolved |  | NOPSO | No longer employing staff |
| SO | SLE | Struck Off/Dissolved |  | NOPSO | No longer employing staff |
| SO | WPE | Struck Off/Dissolved |  | NOPSO | No longer employing staff |
| SO | IPS | Struck Off/Dissolved |  | NOTXAC | Activity ceased |
| SO | NRT | Struck Off/Dissolved |  | NOTXAC | Activity ceased |
| SO | ICA | Struck Off/Dissolved |  | NOTXAC | Activity ceased |
| SO | AIL | Struck Off/Dissolved |  | NOTXAC | Activity ceased |
| SO | MAC | Struck Off/Dissolved |  | NOTXAC | Activity ceased |
| SO | RWT | Struck Off/Dissolved |  | NOTXAC | Activity ceased |
| SO |  | Struck Off/Dissolved |  | STROFF | Removed from the register |
| SP |  | Suspence Clearance |  | NOTXAC | Activity ceased |
| TF |  | Transferred |  | TF | Payments Transferred |
| TS | FAM | TRANSFERRED TO START |  | FAMNIL | 2 Year Nil programme |
| TS | DWT | TRANSFERRED TO START |  | NOLGLI | No Longer Liable |
| TS | REB | TRANSFERRED TO START |  | DECEAS | Deceased |
| TS | ICA | TRANSFERRED TO START |  | STROFF | Removed from the register |
| TS | IPS | TRANSFERRED TO START |  | NOLGLI | No Longer Liable |
| TS | IPE | TRANSFERRED TO START |  | NOELIG | No Longer Eligible |
| TS | FBT | TRANSFERRED TO START |  | NOFBT | Not providing fringe benefits |
| TS | FBI | TRANSFERRED TO START |  | NOFBT | Not providing fringe benefits |
| TS | FBA | TRANSFERRED TO START |  | NOFBT | Not providing fringe benefits |
| TS | MAC | TRANSFERRED TO START |  | EXEMPT | Tax Exemption |
| TS | PIE | TRANSFERRED TO START |  | WOUP | Wound Up |
| TS | NRT | TRANSFERRED TO START |  | ERROR | Registered in Error |
| TS |  | TRANSFERRED TO START |  | NOTXAC | Activity ceased |
| WU | PIE | Wound up |  | WOUP | Wound Up |
| WU |  | Wound up |  | NOTXAC | Activity ceased |
| X5 |  | CROSS BORDER 5YR BAN |  | CRSBAN | Cross Border 5 yr Ban |
| XC | GST | CROSS BORDER CEASED |  | CRSCES | Cross Border Ceased |
| XC |  | CROSS BORDER CEASED |  | NOTXAC | Activity ceased |

### Filing Frequency Periods

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | FILING\_FREQUENCY | FILING\_FREQ\_PERIOD |

| **Source** | | | | **DIP** | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| D |  | D, Twice monthly, 00 |  | twicemonthly | twicemonthly |
| M |  | M, Monthly, 01 |  | monthly | monthly |
| M1 |  | M, Monthly (1st - 15th), 00 |  | monthly | monthly |
| M2 |  | M, Monthly (15th-31st), 00 |  | monthly | monthly |
| MC |  | M, Monthly(Curr Year), 00 |  | monthly | monthly |
| MD |  | M, Monthly(Not P C/Yr), 00 |  | monthly | monthly |
| ME |  | M, MONTHLY (ELECTRONIC), 1 |  | monthly | monthly |
| MN |  | M, Monthly(not policed), 00 |  | monthly | monthly |
| Q |  | Q, Quarterly, 00 |  | quarterly | quarterly |
| SA |  | S, Six Month(Jan & Jul), 06 |  | sixmonthly | sixmonthly |
| SB |  | S, Six Month(Feb & Aug), 06 |  | sixmonthly | sixmonthly |
| SC |  | S, Six Month(Mar & Sep), 06 |  | sixmonthly | sixmonthly |
| SD |  | S, Six Month(Apr & Oct), 06 |  | sixmonthly | sixmonthly |
| SE |  | S, Six Month(May & Nov), 06 |  | sixmonthly | sixmonthly |
| SF |  | S, Six Month(Jun & Dec), 06 |  | sixmonthly | sixmonthly |
| TA |  | T, Two monthly (Odd), 02 |  | twomonthly | twomonthly |
| TB |  | T, Two monthly (Even), 02 |  | twomonthly | twomonthly |
| Y |  | Y, Yearly, 00 |  | annual | annual |
| YC |  | Y, Yearly(current Year), 00 |  | annual | annual |

### Exemption Reasons

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | EXEMPTION\_REASON | EXEMPTION\_REASON |

| **Source** | | | | **DIP** | |
| --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| 01 | IPE | Bank or Building Soc |  | 01 | Reg bank or building society |
| 02 | IPE | Trustees |  | 02 | Trustee company |
| 03 | IPE | Income > $2 million |  | 03 | Income > $2 million |
| 04 | IPE | Est income > $2m |  | 04 | Estimated income > $2 million |
| 05 | IPE | Finance Houses |  | 05 | Financiers and brokers |
| 06 | IPE | Charitable Orgs |  | 06 | Charitable org or estate |
| 07 | IPE | Sports/Racing Clubs |  | 07 | Amat. sport promoter/race club |
| 08 | IPE | Promoter |  | 12 | Improvement or research promo. |
| 09 | IPE | Local/Public Auth. |  | 08 | Local or public authority |
| 10 | IPE | Friend Soc/Cr Unions |  | 09 | Friendly soc. or credit union |
| 11 | IPE | Non-profit Orgs |  | 10 | Non-profit organisation |
| 12 | IPE | T/p's accum losses |  | 11 | Losses or RWT refund of >=$500 |
| 13 | IPE | Non Resident |  | 20 | Non-resident |
| 13 | INC | Non Resident |  | NonRes | NonRes |
| AD | INC | Soc/Club income<1000 |  | SocClubInc | Soc/Club Income < 1000 |
| AS | INC | Amateur Sportsclubs |  | Amateur | Amateur Sportsclub |
| CP | IPE | Consulate Post |  | 21 | Embassy/Consulate |
| CT | INC | Charity |  | CharOrg | Charitable Organisation |
| DH | INC | Dairy Herd Society |  | DairyHerd | Dairy Herd Society |
| DS | INC | District Societies |  | DistrictSoc | DistrictSoc |
| EB | IPE | Embassy |  | 21 | Embassy/Consulate |
| EB | INC | Embassy |  | Embassy | Embassy/High Commission |
| FS | INC | Friendly Societies |  | FriendlySoc | Friendly Society |
| GM | INC | GAMING MACHINE |  | GamingMach | GamingMach |
| edw\_JV | INC | Joint Venture |  | Joint | Joint Venture |
| LA | INC | Local Authority |  | LocPubAuth | LocPubAuth |
| NC | INC | Nil Company |  | NilCompany | Nil Company |
| NR | INC | NON-RES/NO INC RTRN |  | NonRes | NonRes |
| OT | INC | Other |  | Other | Other - Tax Exemption |
| PA | INC | Public Authority |  | LocPubAuth | LocPubAuth |
| PE | INC | PIE Exempt |  | PieExempt | PieExempt |
| PI | INC | PAYE INTERMEDIARY |  | Other | Other - Tax Exemption |
| SR | INC | Scient. & Ind. Res. |  | SciIndRes | SciIndRes |
| VT | INC | Vet. service clubs |  | VetService | Vet. service club |
| XB | INC | CROSS BORDER GST |  | CrossBrdGst | Cross Border GST |

### Accounting Basis

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | ACCOUNTING\_BASIS\_CODE | ACCOUNTING\_BASIS |

| **Source** | | | | **DIP** | |
| --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| H |  | Hybrid |  | HYB | Hybrid |
| I |  | Invoice |  | INV | Invoice |
| P |  | Payments |  | PYM | Payments |

### Registration Circumstance

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | NAT\_OF\_REGN | REGISTRATION\_CIRCUMSTANCE |

| **Source** | | | | **DIP** | |
| --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| F |  | Forced |  | FRC | Forced |
| R |  | Required |  | REQ | Required |
| V |  | Voluntary |  | VOL | Voluntary |

### PIE Filing Option

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | PIE\_REPORT\_OPTION | FILING\_OPTION |

| **Source** | | | | **DIP** | |
| --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| AE | PIE | Annual Exitor |  | PIEMO | Exit MRP |
| AI | PIE | Annual Income Tax |  | PIEINC | Annual income tax |
| AP | PIE | Annual Prov Tax |  | PIE1A | Provisional tax |
| QZ | PIE | Quarterly Zero Rates |  | PIE3M | Quarterly MRP |

### PIE Attribution Period

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | INCOME\_ALLOCATION\_PERIOD | ATTRIBUTION\_PERIOD |

| **Source** | | | | **DIP** | |
| --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| A | PIE | Annual |  | ANNUAL | Annual |
| D | PIE | Daily |  | DAILY | Daily |
| M | PIE | Monthly |  | MONTH | Monthly |
| N | PIE | Not Applicable |  | NA | Not Applicable |
| Q | PIE | Quarterly |  | QUART | Quarterly |

NB. EDW also contains the code ‘X’ which has no Start equivalent.

### PIE FDR Calculation Period

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | FDR\_CALCULATION\_PEROD | FDR\_CAL\_PERIOD |

| **Source** | | | | **DIP** | |
| --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| A | PIE | Annual |  | ANNUAL | Annual |
| D | PIE | Daily |  | DAILY | Daily |
| M | PIE | Monthly |  | MONTH | Monthly |
| N | PIE | Not Applicable |  | NA | Not Applicable |
| O | PIE | Other |  | OTHER | Other |
| Q | PIE | Quarterly |  | QUART | Quarterly |
| W | PIE | Weekly |  | WEEK | Weekly |

NB. EDW also contains the code ‘X’ which has no Start equivalent.

### PAY Filing Option

| **Source System** | **Source Code Name** | **DIP Code Name** |
| --- | --- | --- |
| FIRST | FILING\_FREQUENCY | FILING\_OPTION |

| **Source** | | | | **DIP** | | **Comments** |
| --- | --- | --- | --- | --- | --- | --- |
| **Code Value** | **Tax Type** | **Description** | **Cust Type** | **Code Value** | **Description** |
| D | PAY | Twice Monthly |  | D | Twice Monthly |  |
| I | PAY | Irregular (No IR 348) |  | I | Irregular |  |
| M | PAY | Monthly |  | M | Monthly |  |
| M1 | PAY | Monthly 1st-15th |  | M1 | Monthly First Half |  |
| M2 | PAY | Monthly 15th-31st |  | M2 | Monthly Second Half |  |
| ME | PAY |  |  | ME | Monthly Electronic |  |
| MN | PAY | Monthly (not policed) |  | MN | Monthly Non-Policed |  |
| N | PAY | No Return Required |  | N | No Return Required | Does not exist in Start as at 7-11-2019. Included for backwards compatability with EDW. |
| Y | PAY | Yearly |  | Y | Yearly |