**Etl Usage Guide**

Guidance for use the Etl

Bradford Teaching Hospital Foundation Trust

EHDEN OMOP Project

Contents

[1 Introduction 0](#_Toc152339009)

[2 Setup 0](#_Toc152339010)

[3 Running the Etl 0](#_Toc152339011)

[4 Testing the Etl counts 0](#_Toc152339012)

[5 Etl details 2](#_Toc152339013)

[5.1 Transfer 2](#_Toc152339014)

[5.2 Counts 4](#_Toc152339015)

[6 Understanding the ETL and mapping structure 5](#_Toc152339016)

[6.1 Example 5](#_Toc152339017)

[6.1.1 Corresponding ETL code 6](#_Toc152339018)

# Introduction

This guide provides details on the use and implementation of the Etl to transform the source data to the OMOP CDM format.

# Setup

If the Etl has not been setup yet, then this will need to be called. It isn’t required when running the Etl. The setup process creates the log mappings tables.

CALL CY\_IMOSPHERE\_WORKSPACE.ETL\_SETUP()

# Running the Etl

The transfer process will copy and transform the data from the source to the OMOP CDM.

CALL CY\_IMOSPHERE\_WORKSPACE.ETL\_TRANSFER()

# Testing the Etl counts

The counts process will return any records where the expected counts from the source don’t match counts from the destination.

CALL CY\_IMOSPHERE\_WORKSPACE.ETL\_COUNTS()

# Etl details

This is quick reference guide for the Etl components.

## Transfer

|  |  |
| --- | --- |
| **Procedure** | **Source table, notes** |
| POPULATE\_SPELLNUMBER\_LOOKUPS | Used for visit occurrence and visit detail   * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 * SUS\_BRI\_APC\_010415\_to\_300619\_P1 |
| TRANSFER\_LOCATION | * tbl\_srorganisation * tbl\_SRPatientAddressHistory * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 * SUS\_BRI\_AE\_010415\_to\_300619 * SUS\_BRI\_OP\_010415\_to\_300619 * SUS\_BRI\_APC\_010415\_to\_300619\_P1 |
| TRANSFER\_CARE\_SITE | * tbl\_srorganisation |
| TRANSFER\_PERSON | Date of birth and person id are required.   * tbl\_SRPatient * SUS\_BRI\_AE\_010415\_to\_300619 * SUS\_BRI\_APC\_010415\_to\_300619\_P1 * SUS\_BRI\_OP\_010415\_to\_300619 * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 |
| TRANSFER\_DEATH | Date of death is required.   * tbl\_SRPatient |
| TRANSFER\_PROVIDER | See procedure for transformations.   * tbl\_SRCode * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 * SUS\_BRI\_APC\_010415\_to\_300619\_P1 |
| TRANSFER\_VISIT\_OCCURRENCE | Uses mapping tables to create unique ids.   * tbl\_SRVisit * SUS\_BRI\_OP\_010415\_to\_300619 * SUS\_BRI\_APC\_010415\_to\_300619\_P1 * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 * SUS\_BRI\_AE\_010415\_to\_300619 |
| TRANSFER\_OBSERVATION\_PERIOD | Uses visit\_occurrence. |
| TRANSFER\_VISIT\_DETAIL | Uses mapping tables to create unique ids.   * SUS\_BRI\_APC\_010415\_to\_300619\_P1 * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 |
| TRANSFER\_CONDITION\_OCCURRENCE | Multiple columns are mapped, duplicate codes are excluded.   * tbl\_SRPatient * tbl\_SRCode * SUS\_BRI\_APC\_010415\_to\_300619\_P1 * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 |
| TRANSFER\_PROCEDURE\_OCCURRENCE | Multiple columns are mapped, duplicate codes are excluded.   * SUS\_BRI\_AE\_010415\_to\_300619 * SUS\_BRI\_APC\_010415\_to\_300619\_P1 * SUS\_BRI\_OP\_010415\_to\_300619 |
| TRANSFER\_OBSERVATION | Multiple columns are mapped, duplicate codes are excluded.   * tbl\_SRVisit * tbl\_SRPatient * tbl\_SRCode * SUS\_BRI\_OP\_010415\_to\_300619 * SUS\_BRI\_APC\_010415\_to\_300619\_P1 * src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 * SUS\_BRI\_AE\_010415\_to\_300619 |
| TRANSFER\_MEASUREMENT | Multiple columns are mapped, duplicate codes are excluded.   * SUS\_BRI\_AE\_010415\_to\_300619 * tbl\_SRCode |
| TRANSFER\_DRUG\_EXPOSURE | * tbl\_SRImmunisation * tbl\_SRPrimaryCareMedication |
| TRANSFER\_CDM\_SOURCE | Static data. |

## Counts

|  |  |
| --- | --- |
| **Procedure** | **Count method** |
| COUNT\_LOCATION | Counts overall number of locations. |
| COUNT\_CARE\_SITE | Counts overall number of care sites. |
| COUNT\_PERSON | Counts overall number of people. |
| COUNT\_DEATH | Counts overall number of deaths. |
| COUNT\_PROVIDER | Counts overall number of providers. |
| COUNT\_VISIT\_OCCURRENCE | Counts overall number of visit occurrences. |
| COUNT\_OBSERVATION\_PERIOD | Counts number of people with missing observation periods. |
| COUNT\_VISIT\_DETAIL | Counts overall number of visit details. |
| COUNT\_CONDITION\_OCCURRENCE | Counts per source value. |
| COUNT\_OBSERVATION | Counts per source value. |
| COUNT\_MEASUREMENT | Counts per source value. |
| COUNT\_DRUG\_EXPOSURE | Counts per source value. |

# Understanding the ETL and mapping structure

If more source data fields are required to be added to the ETL, the relevant mappings and logic need to be added to the components list above.

Imosphere manages the mappings in an xlsx file ‘ConnectedBradford Mappings 11-10-22’. This file represents the logic and mappings that are present in the stored procedures delivered in the ETL. Each tab in the spreadsheet will correspond to both an OMOP target table and a stored procedure in the ETL.

## Example

If looking at the mapping file ‘ConnectedBradford Mappings 11-10-22’, the ‘OBSERVATION’ tab corresponds to the ‘ObservationTransfer’ SQL file in the ETL package.

The columns in file are described briefly below:

A Field Name: The filed name present in the OBSERVATION table in the target OMOP database

B Source Table: The source table where the relevant source data to be mapped is located

C Source Column: The source column within the relevant table where the data is located to be mapped to the corresponding filed in the OMOP table.

D Mapping logic notes: details of how the SQL code will define the mapping logic from source to target. This will typically specify one or more of the following:

1. Logic to conform to OMOP standards or to consolidate, rationalise or filter source data, sometimes form multiple source fields. E.g. observation\_datetime will be defined by CONCAT(Appointment\_Date,Appointment\_Time)
2. References or lookups to apply. This might be referencing a specific vocabulary, such as CTV3, which are already mapped to standard OMOP concepts and are available for the OMOP reference library: <https://athena.ohdsi.org/search-terms/start>. Often, the term ‘See Lookup Mappings’ is used. This is in cases where extensive lists of concepts are captured within a single field. These are listed in a separate tab called ‘Lookup Mappings’.
3. Specific OMOP concepts. This is when a code has been manually mapped to a corresponding OMOP concept. E.g. responses from Discharge\_Destination\_Hospital\_Provider\_Spell in the src\_BDCT\_CDS130InpatientFCEdata010419\_310320Final\_V2 table are mapped as follows:
   1. If source = '19': 4140634 - Discharge to home
   2. If source='79': 4194377 - Patient discharge, deceased, no autopsy

E Notes: This column is occasionally used to add any additional notes that will be relevant to constructing the transfers.

### Corresponding ETL code

Staying with the OBSERVATION tab in the mappings spreadsheet, if we look at row 22:

* The OMOP destination field is observation\_concept\_id
* The source Table is SUS\_BRI\_AE\_010415\_to\_300619
* The Source Column is Ethnic\_Category
* The mapping applied will be a specific OMOP concept: 44803968 - Ethnicity

This is represented in the ObservationTransfer stored procedure as:

/\*

--[SUS\_BRI\_AE\_010415\_to\_300619 - Ethnic\_Category] Transfer

\*/

SET max\_current\_id = COALESCE((SELECT observation\_id FROM `CY\_IMOSPHERE\_CDM\_531.observation` ORDER BY 1 DESC LIMIT 1), 0);

INSERT INTO `CY\_IMOSPHERE\_CDM\_531.observation`

(

observation\_id,

person\_id,

observation\_concept\_id,

observation\_source\_value,

observation\_date,

observation\_datetime,

observation\_type\_concept\_id,

value\_as\_concept\_id,

visit\_occurrence\_id

)

SELECT DISTINCT

max\_current\_id + ROW\_NUMBER() OVER() as observation\_id,

o.person\_id,

o.observation\_concept\_id,

o.observation\_source\_value,

o.observation\_date,

o.observation\_datetime,

o.observation\_type\_concept\_id,

o.value\_as\_concept\_id,

o.visit\_occurrence\_id

FROM

(

SELECT DISTINCT

o.person\_id person\_id,

44803968 observation\_concept\_id, --Ethnicity

o.Ethnic\_Category observation\_source\_value,

PARSE\_DATE("%Y%m%d", CONCAT(Date\_of\_Birth ,'01')) observation\_date,

PARSE\_DATETIME('%Y%m%d%H%M%S', CONCAT(Date\_of\_Birth ,'01000000')) observation\_datetime,

32817 observation\_type\_concept\_id, --EHR

oc.concept\_id value\_as\_concept\_id,

vl.id visit\_occurrence\_id

FROM `CY\_IMOSPHERE\_WORKSPACE.SUS\_BRI\_AE\_010415\_to\_300619` o

JOIN `CY\_IMOSPHERE\_WORKSPACE.tbl\_etl\_mappings` oc on oc.source\_value = o.Ethnic\_Category AND oc.destination\_table = 'OBSERVATION' AND oc.source\_table = 'SUS\_BRI\_AE\_010415\_to\_300619' AND oc.source\_column = 'Ethnic\_Category'

LEFT JOIN (SELECT id, source\_value FROM `CY\_IMOSPHERE\_WORKSPACE.tbl\_visit\_occurrence\_lookups` l WHERE l.source\_table = 'SUS\_BRI\_AE\_010415\_to\_300619') vl On vl.source\_value = CONCAT(CAST(o.AandE\_Attendance\_Number as STRING),'\_', o.person\_id)

WHERE (o.Date\_of\_Birth IS NOT NULL AND o.Date\_of\_Birth NOT LIKE '% %')

AND o.person\_id IS NOT NULL

) o

LEFT JOIN `CY\_IMOSPHERE\_CDM\_531.observation` oo ON

oo.person\_id = o.person\_id

AND oo.observation\_concept\_id = o.observation\_concept\_id

AND oo.observation\_source\_value = o.observation\_source\_value

AND COALESCE(oo.observation\_source\_concept\_id, -1) = -1

AND oo.observation\_datetime = o.observation\_datetime

AND oo.observation\_type\_concept\_id = o.observation\_type\_concept\_id

AND COALESCE(oo.provider\_id, -1) = -1

AND COALESCE(oo.visit\_occurrence\_id, -1) = COALESCE(o.visit\_occurrence\_id, -1)

AND COALESCE(oo.visit\_detail\_id, -1) = -1

AND COALESCE(oo.value\_as\_concept\_id, -1) = COALESCE(o.value\_as\_concept\_id, -1)

AND COALESCE(CAST(oo.value\_as\_number AS STRING), '') = ''

AND COALESCE(oo.qualifier\_concept\_id, -1) = -1

AND COALESCE(oo.unit\_concept\_id, -1) = -1

AND COALESCE(oo.qualifier\_source\_value, '') = ''

AND COALESCE(oo.unit\_source\_value, '') = ''

WHERE oo.observation\_id IS NULL;

INSERT INTO `CY\_IMOSPHERE\_WORKSPACE.tbl\_etl\_log` (log\_message, run\_id, related\_table, related\_count, logged\_at)

SELECT 'Added observation', run\_id, 'SUS\_BRI\_AE\_010415\_to\_300619 - Ethnic\_Category -> observation', @@row\_count, CURRENT\_DATETIME();