



Australian Government
**Australian Institute of
Health and Welfare**

Residential Mental Health Care

National Minimum Data Set

2026-27 version 7.01

The metadata for RMHC 7.01 can be found on the [Online Validator metadata page](#)

As at 27 February, 2026

Table of Contents

- 1. 2026-27 RMHC NMDS.....3
 - 1.1. Essential definitions3
 - 1.2. Changes for 2026-27.....3
 - 1.3. Reporting service provider entities.....3
 - 1.4. Data model of the RMHC Extract.....6
 - 1.5. Data integrity8
 - 1.6. Dataset specification (DSS)8
 - 1.7. Data elements.....22
 - 1.8. Virtual elements39
 - 1.9. Rules.....63
- 2. Submission and delivery of RMHC NMDS data files.....165
 - 2.1. Timelines165
 - 2.2. File type and naming convention165
 - 2.3. Validation166
 - 2.4. Submission report.....167
 - 2.5. Additional information and queries167

1. 2026-27 RMHC NMDS

1.1. Essential definitions

Scope - Residential mental health care NMDS (METEOR ID [807050](#))

Statistical Unit - Episodes of residential care (METEOR ID [723170](#))

1.2. Changes for 2026-27

The specific detailed changes to the 2026-27 specifications (version 7.01), compared to the 2025-26 (version 7.00) specifications, are detailed below.

1.2.1. Changes to the data model

No changes were made to the data model.

1.2.2. Changes to definitions

No changes were made to the definitions.

1.3. Reporting service provider entities

The reporting of service entities aims to create relationships between the mental health NMDSs, and where possible, the National Outcomes and Casemix Collection (NOCC), Local Hospital Networks/Public Hospital Establishments (PHE) NMDS and Admitted Patient Care (APC) NMDS (see [Table 1.1](#)).

The identifiers used in the RMHC NMDS are:

- State or territory (1 character)
- Region (2 characters)
- Specialised mental health service organisation (9 characters)
- Service unit cluster (9 characters)
- Service unit (9 characters)

Table 1.1 Reporting requirements for mental health and related NMDSs

Identifier element names	METEOR identifier	Community mental health care NMDS	Residential mental health care NMDS	National Outcomes and Casemix Collection NMDS	Mental health establishments NMDS	Public hospital establishments NMDS	Admitted patient care NMDS
Australian State or Territory identifier	790405	Yes	Yes	Yes	Yes	Yes	Yes
Region identifier	269940	Yes	Yes	Yes	Yes	Yes	Yes
Specialised mental health service organisation identifier	795837	Yes	Yes	Yes	Yes	No	No
Hospital/ Service unit cluster identifier	795844 (MHE, NOCC) / 795848 (MHE, CMHC, RMHC, NOCC)	Yes	Yes	Yes	Yes	Yes	Yes
Service unit identifier	795850 (MHE, NOCC) / 795855 (MHE, CMHC, NOCC) / 795859 (MHE, RMHC, NOCC)	Yes	Yes	Yes	Yes	No	No

The use of identical identifiers between the various mental health data sets is tested via the Mental Health Establishments Skeleton file (SKL), handled by the Online Validator. The reports section of the RMHC submission will highlight any mismatches which should be rectified either through re-supply of the Skeleton file, or adjustment to the RMHC submission.

The following section explores in more detail the reporting levels used in the RMHC NMDS.

1.3.1. State or territory

This level refers to the state or territory and should be reported using the *State/Territory identifier* data element.

1.3.2. Region

The region refers to an administrative concept not a geographical one. States and territories may have one or more regions into which the state or territory is divided and to which its mental health service organisations belong. Region would be reported using the *Region identifier* (RegId) data element. In the smaller states or in the territories there may only be one or no region. In these cases, the *Region identifier* is to be reported as '00' and the *Region name* (RegName) would repeat the name of the State or Territory.

1.3.3. Organisation

The organisation is defined according to the Object class *Specialised mental health service organisation* (METEOR identifier 795837) and reported using the *Organisation Identifier* (OrgId) data element. An organisation is a separately constituted specialised mental health service that is responsible for the clinical governance, administration and financial management of service units providing specialised mental health care. An organisation may consist of one or more service units based in different locations.

1.3.4. Service unit cluster

A specialised mental health service organisation may consist of one or more clusters of service units providing services in admitted patient, residential and/or ambulatory settings. For example, a specialised mental health service organisation may consist of several hospitals (clusters of admitted patient service units) and/or two or more ambulatory or residential service clusters (for example, a cluster of child and adolescent ambulatory service units, and a cluster of aged residential service units).

To allow service units (as defined below) to be reported individually, but still to be identified as part of a cluster, a separate reporting level has been created called 'Service unit cluster'.

Residential service units will not necessarily belong to a 'cluster'. However, groups of residential service units could be usefully reported as clusters. For example, clusters may exist of groups of residential services for aged persons in particular geographical areas. Where there is no service unit cluster then all service units within the organisation should be identified under a *Service unit cluster identifier* reported as '00000' and the *Service unit cluster name* would use the relevant organisation name.

1.3.5. Service units

Service units in-scope for reporting to the RMHC NMDS must be identical to those reported to the MHE NMDS—that is, at a minimum disaggregated by operator (government-operated and non government operated), hours staffed (24-hour staffed and non-24-hour staffed), and Target population, even though hours staffed and target population are not reporting requirements for the RMHC NMDS.

1.3.6. Sector

Sector is not considered part of the identifier, however, within this NMDS, sector is an important attribute for each service unit.

1.3.7. Consistency of identifiers across reference periods

In the specifications from 2025–26 onwards, establishment identifiers (OrgId, ClusId, SUIId) have increased the maximum character length to 9 characters.

Where no major service reorganisations have occurred, the region, organisation, service unit cluster and service unit identifiers (RegId, OrgId, ClusId, SUIId) used by a jurisdiction should be preferably identical to the previous year, and space-fill as required. However, given that all jurisdictions have committed to aligning ID numbers between the different NMDSs, changes in ID numbers due to this process will be accepted, as will be the case for those jurisdictions that have undergone significant reorganisation of service delivery that warrant new service entity identifiers. In these cases, jurisdictions should provide a supplementary mapping document that clearly illustrates the changes in ID numbers between collection periods, at all levels.

Region, organisation, service unit cluster and service unit name changes are acceptable, especially if the new name is more locally relevant. These will be identified as a change in the Online Validator, however will not affect the generation of the historical trends reports for RMHC in future.

1.4. Data model of the RMHC Extract

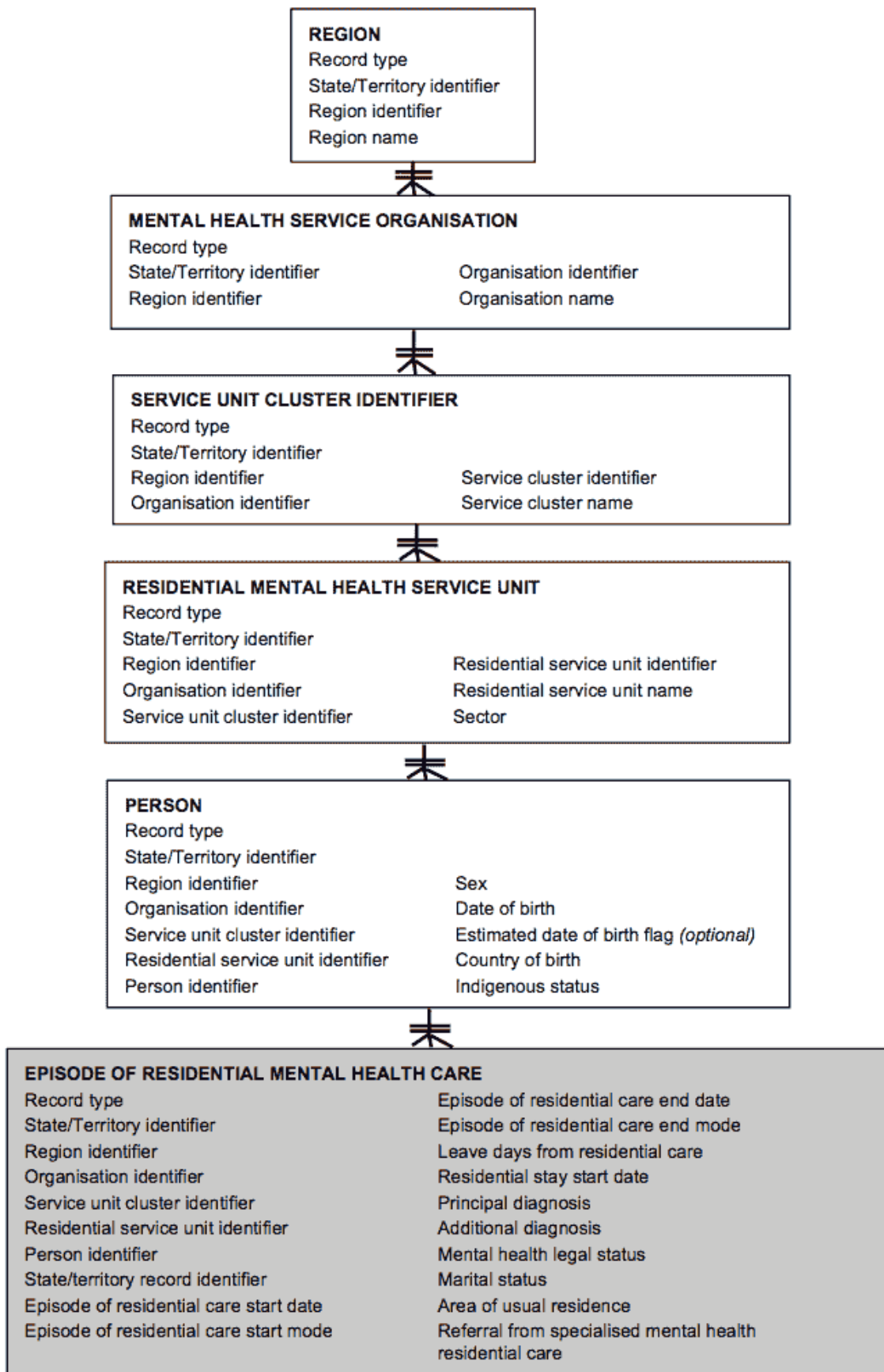
The basic design of the extract consists of a single data record for each *Episode of residential care*. Each *Episode of residential care* 'belongs' to a *Person* (the patient or consumer of services), who in turn is linked to a *Residential mental health service unit* (the provider of services), which may be linked to a *Residential service unit cluster*, which is linked to a *Specialised mental health service organisation*, which is linked to a *Region* which is linked to a *State/Territory*.

The structure of the data to be reported is represented in the data model shown in [Fig. 1.1](#). In the model, a single *Residential mental health service unit* has associated records for one or more *Persons*, who each may have one or more *Episodes of residential care*. Each of the six data model building blocks (region, organisation, service unit cluster, service unit, person, episodes) has a unique set of attributes which comprise the NMDS data elements and additional supplementary information.

Two features of the model should be noted:

1. First, details of the residential service units reporting episode of residential care data are incorporated as part of the data extract, allowing linkage to related data sets provided by states and territories (in particular, the MHE NMDS).

2. Second, non-volatile person-level data in respect of patients (Date of Birth, Sex, Country of Birth, Indigenous Status) are separated from person-level data items that may change between episodes. This is designed to remove the redundancy that would occur if all patient-level data items were reported with each episode record and allow more straightforward counts/analysis at client level.



1.5. Data integrity

For cases of missing data (that is, unknown, not stated or not available):

- For **Numeric [Num] fields**, the data should be reported as zero, using leading zeros when necessary to pad out the field to the required length. The principle here is that all numeric fields require a valid value.
- For **Text [Char] fields**, the data should be space-filled to the required length. For single character fields where a 'missing/not stated' value has been specified for a particular data element (for example, '9' has been specified for missing data), use the stated value for 'missing/not stated' rather than simply space filling.

Values in **Date [Date]** fields must be recorded in compliance with the standard format used across the *National health data dictionary*; specifically, dates must be of fixed 8 column width in the format DDMMYYYY, with leading zeros used when necessary to pad out a value. For instance, 13 March 2027 would appear as 13032027.

Values in **Numeric [Num]** fields must be zero-filled and right-justified. These should consist only of the numerals 0 to 9 and the decimal ('.') point if applicable to the data element.

Note: Fields defined as 'Numeric' are those that have numeric properties—that is, the values, for example, can be added or subtracted in a manner that is valid. Where a field uses numeric characters that do not have these properties (for example, the use of numbers for *Patient identifier*), the field is defined as 'Character'.

Values in **Character [Char]** fields must be left justified and space-filled. These should consist of any of the printable ASCII character set (that is, excluding control codes such as newline, bell and linefeed).

1.6. Dataset specification (DSS)

The file structure for the transmission of data from jurisdictions to the AIHW is a single Fixed Format data file. The following tables specify the order in which the data items should be provided to the AIHW.

The extract format consists of a set of hierarchically ordered *Data records*, of which there are six types (see [Table 1.2](#)):

- Region details records
- Organisation details records
- Service unit cluster details records
- Service unit details records
- Person details records
- Episode of residential care details records

In each extract file for any given period, the *Data records* must be preceded by a single *File Header Record* having the structure outlined below in [Table 1.3](#).

All records presented in the extract file should be grouped in the following order: Header Record, Region details records, Organisation details records, Service unit details records, Person details records, followed by Episode of residential care records.

With the exception of Region, Organisation and Service unit cluster details records, all *Data records* should include the following elements in the order shown:

- Record Type
- Establishment identifier (comprising: *State/Territory identifier*, *Region identifier*, *Organisation identifier*, *Service unit cluster identifier* and *Service unit identifier*)
- Specific data in the format specified for the given record type.

The order of fields in a record must be the same as the order they are specified in the Record Layouts below. Field values should be formatted as shown in the Record Layouts.

The first field in each record must be *Record Type*. Valid values are shown in [Table 1.2](#).

Table 1.2 Valid values for Record Type

Record Type	Description
HR	File Header Record
REG	Region details
ORG	Organisation details
CLUS	Service unit cluster details
SERV	Service unit details
PER	Person details
EPI	Episode of residential care details

1.6.1. File header record

The first record of the extract file must be a File Header Record (Record Type = 'HR'), and it must be the only such record in the file.

The File Header Record is a quality control mechanism, which uniquely identifies each file that is sent to the AIHW (that is, who sent the file, what date the file was sent, batch number of file, etc). The information contained in the header fields will be checked against the actual details of the file to ensure that the file received has not been corrupted.

The layout of the File Header Record is shown in [Table 1.3](#).

Table 1.3 Record Layout for File Header Record within the data extract

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type RecType	Char[8]	1	—	Value = HR
State/Territory Identifier 1 State	Char[1]	9	790405	1: New South Wales 2: Victoria 3: Queensland 4: South Australia 5: Western Australia 6: Tasmania 7: Northern Territory 8: Australian Capital Territory
Batch Number BatchNo	Char[9]	10	—	Represents the YYYYNNNNN component of the extract file name.
Report Period Start Date RepStart	Date[8]	19	—	Report period start date
Report Period End Date RepEnd	Date[8]	27	—	Report period end date
Data File Generation Date GenDt	Date[8]	35	—	Data file generation date
Data File Type FileType	Char[4]	43	—	Value = RMHC
RMHC Specification Version Number SpecVer	Char[5]	47	—	Value = 07.01

Record length = 51

Notes

[1]

METEOR includes code 9, but that is not applicable to the RMHC NMDS

1.6.2. Region data record

The extract format for the *Data records* is specified in detail in tables [Table 1.3](#) to [Table 1.9](#). The order of fields in each record must be the same as the order they are shown below. Field values should be formatted as specified.

Table 1.4 Data record layout - Region Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type RecType	Char[8]	1	—	Value = REG
State/ Territory Identifier 2 State	Char[1]	9	790405	1: New South Wales 2: Victoria 3: Queensland 4: South Australia 5: Western Australia 6: Tasmania 7: Northern Territory 8: Australian Capital Territory
Region Identifier RegId	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction) Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Region Name RegName	Char[60]	12	407187	Common name used to identify the Region.

Record length = 71

Notes

[2]

METEOR includes code 9, but that is not applicable to the RMHC NMDS

1.6.3. Organisation data record

Table 1.5 Data record layout - Organisation Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type RecType	Char[8]	1	—	Value = ORG

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier ³ State	Char[1]	9	790405	1: New South Wales 2: Victoria 3: Queensland 4: South Australia 5: Western Australia 6: Tasmania 7: Northern Territory 8: Australian Capital Territory
Region Identifier RegId	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction) Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Organisation Identifier OrgId	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDS for Mental Health Establishments.
Organisation Name OrgName	Char[100]	21	405767	Common name used to identify the Organisation

Record length = 120

Notes

[3]

METEOR includes code 9, but that is not applicable to the RMHC NMDS

1.6.4. Service unit cluster data record

Table 1.6 Data record layout - Service Unit Cluster Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type RecType	Char[8]	1	—	Value = CLUS

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier ⁴ State	Char[1]	9	790405	1: New South Wales 2: Victoria 3: Queensland 4: South Australia 5: Western Australia 6: Tasmania 7: Northern Territory 8: Australian Capital Territory
Region Identifier RegId	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction) Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Organisation Identifier OrgId	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDS for Mental Health Establishments.
Service Unit Cluster Identifier ClusId	Char[9]	21	795848	A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDS for Mental Health Establishments, the identifiers used in this collection should be the same.
Service Unit Cluster Name ClusName	Char[100]	30	409209	If no cluster applies, enter organisation name as appears in previous line.

Record length = 129

Notes

[4]

METEOR includes code 9, but that is not applicable to the RMHC NMDS

1.6.5. Residential service unit data record

Table 1.7 Data record layout — Residential Service Unit Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type RecType	Char[8]	1	—	Value = SERV

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier ⁵ <small>State</small>	Char[1]	9	790405	1: New South Wales 2: Victoria 3: Queensland 4: South Australia 5: Western Australia 6: Tasmania 7: Northern Territory 8: Australian Capital Territory
Region Identifier <small>RegId</small>	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction) Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Organisation Identifier <small>OrgId</small>	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDS for Mental Health Establishments.
Service Unit Cluster Identifier <small>ClusId</small>	Char[9]	21	795848	A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDS for Mental Health Establishments, the identifiers used in this collection should be the same.
Residential Service Unit Identifier <small>SUId</small>	Char[9]	30	795859	A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Residential Service Unit Name <small>SUName</small>	Char[100]	39	722715	Common name used to identify the service unit
Sector <small>Sector</small>	Char[1]	139	269977	1: Public 2: Private

Record length = 139

Notes

[5]
METEOR includes code 9, but that is not applicable to the RMHC NMDS

1.6.6. Residential person data record

Attention

Where multiple values of Sex, Gender, Date of birth, Country of birth, or Indigenous status are recorded for different service contacts for one PersId, data providers should adopt the value recorded for the last valid service contact.

Table 1.8 Data record layout — Person Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type RecType	Char[8]	1	—	Value = PER
State/ Territory Identifier 6 State	Char[1]	9	790405	1: New South Wales 2: Victoria 3: Queensland 4: South Australia 5: Western Australia 6: Tasmania 7: Northern Territory 8: Australian Capital Territory
Region Identifier RegId	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction) Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Organisation Identifier OrgId	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDS for Mental Health Establishments.
Service Unit Cluster Identifier ClusId	Char[9]	21	795848	A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDS for Mental Health Establishments, the identifiers used in this collection should be the same.
Residential Service Unit Identifier SUIId	Char[9]	30	795859	A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Person Identifier PersId	Char[64]	39	799014	Person identifier is unique and stable for each individual patient within each service unit. Individual service units or collection authorities may use their own alphabetic, numeric or alphanumeric coding systems. The following special characters can be used: +, =, /

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Sex ⁷ Sex	Char[1]	103	741686	<ul style="list-style-type: none"> 1: Male 2: Female 3: Another term 9: Not stated / inadequately described
Date of Birth DoB	Date[8]	104	287007	The date of birth of the person.
Estimated Date of Birth Flag ⁸ DoBFlag	Char[1]	112	—	<ul style="list-style-type: none"> 1: Date of birth is accurate 2: Date of birth is an estimate 8: Date of birth is a “dummy” date (ie, 09099999) 9: Accuracy of stated date of birth is not known
Country of Birth CoB	Char[4]	113	659454	The country in which the person was born. To be provided in accordance with the Standard Australian Classification of Countries (SACC). ABS catalogue no. 1269.0 (2016). Values from 1601-1607, inclusive, are not permitted in this NMDS (Antarctica).
Indigenous Status IndigSt	Char[1]	117	602543	<ul style="list-style-type: none"> 1: Aboriginal but not Torres Strait Islander origin 2: Torres Strait Islander but not Aboriginal origin 3: Both Aboriginal and Torres Strait Islander origin 4: Neither Aboriginal and Torres Strait Islander origin 9: Not stated/inadequately described
Gender ⁹ Gender	Char[1]	118	741842	<ul style="list-style-type: none"> 1: Man, or boy, or male 2: Woman, or girl, or female 3: Non-binary 4: Different term 5: Prefer not to answer 9: Not stated / inadequately described

Record length = 118

Notes

[6]

METEOR includes code 9, but that is not applicable to the RMHC NMDS

[7]

Sex and Gender are conditional data elements. Data must be reported for at least one of the two elements, either Sex or Gender. Data may be reported for both. Where data is not collected for one of the data elements, this field should be reported as 'null' through space-filling.

[8]

Optional data element providing additional information regarding the quality of date of birth data. Code 1 should be used when it is known that the reported date of birth is accurate, code 2 when it is known that one or more parts of the date of birth is an estimate, code 8 when birth date is unknown and a 'dummy' date of birth has been used (that is, 09099999), and code 9 when it is not known whether the date of birth is accurate or an estimate.

[9]

Sex and Gender are conditional data elements. Data must be reported for at least one of the two elements, either Sex or Gender. Data may be reported for both. Where data is not collected for one of the data elements, this field should be reported as 'null' through space-filling.

1.6.7. Episode of residential care data record

Table 1.9 Data record layout — Episode of Residential Care Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type RecType	Char[8]	1	—	Value = EPI
State/ Territory Identifier 10 State	Char[1]	9	790405	1: New South Wales 2: Victoria 3: Queensland 4: South Australia 5: Western Australia 6: Tasmania 7: Northern Territory 8: Australian Capital Territory
Region Identifier RegId	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction) Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Organisation Identifier OrgId	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDS for Mental Health Establishments.

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Service Unit Cluster Identifier ClusId	Char[9]	21	795848	A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDS for Mental Health Establishments, the identifiers used in this collection should be the same.
Residential Service Unit Identifier SUId	Char[9]	30	795859	A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.
Person Identifier PersId	Char[64]	39	799014	Person identifier is unique and stable for each individual patient within each service unit. Individual service units or collection authorities may use their own alphabetic, numeric or alphanumeric coding systems. The following special characters can be used: +, =, /
State/Territory Record Identifier RecordId	Char[128]	103	—	This can be formed using alphabetic, numeric or alphanumeric coding systems. The following special characters can be used: +, =, /
Episode of Residential Care Start Date EpiStartDt	Date[8]	231	723194	Valid date expressed as DDMMYYYY Date on which resident formally or statistically starts an episode of residential care.
Episode of Residential Care Start Mode EpiStartMode	Char[1]	239	723196	2: Start of new residential stay 3: Start of a new reference period 4: Start of expected short concurrent residential stay 9: Unknown/not stated/inadequately described
Episode of Residential Care End Date EpiEndDt	Date[8]	240	723189	Date on which resident formally or statistically ends an episode of residential care.
Episode of Residential Care End Mode EpiEndMode	Char[1]	248	723192	1: Died 2: Left against clinical advice/at own risk 3: Did not return from leave 4: Formal discharge from residential care at this establishment 5: End of reference period 6: Return to other residential health service 9: Unknown/not stated/inadequately described

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Leave Days From Residential Care NLeaveDays	Number[3]	249	723200	Represented as NNN.
Residential Stay Start Date ResStartDt	Date[8]	252	723256	Date on which resident formally started a residential stay.
Principal Diagnosis DxPrinc	Char[6]	260	793125	<p>Represented as ANN{.N[N]}</p> <p>The diagnosis established after study to be chiefly responsible for occasioning an episode of residential care, as represented by a code. The principal diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).</p>
Additional Diagnosis 1 DxAdd01	Char[6]	266	793130	<p>Represented as ANN{.N[N]}</p> <p>Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).</p>
Additional Diagnosis 2 DxAdd02	Char[6]	272	793130	<p>Represented as ANN{.N[N]}</p> <p>Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).</p>
Additional Diagnosis 3 DxAdd03	Char[6]	278	793130	<p>Represented as ANN{.N[N]}</p> <p>Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).</p>

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Additional Diagnosis 4 DxAdd04	Char[6]	284	793130	Represented as ANN{.N[N]} Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).
Additional Diagnosis 5 DxAdd05	Char[6]	290	793130	Represented as ANN{.N[N]} Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).
Additional Diagnosis 6 DxAdd06	Char[6]	296	793130	Represented as ANN{.N[N]} Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).
Additional Diagnosis 7 DxAdd07	Char[6]	302	793130	Represented as ANN{.N[N]} Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).
Additional Diagnosis 8 DxAdd08	Char[6]	308	793130	Represented as ANN{.N[N]} Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Additional Diagnosis 9 DxAdd09	Char[6]	314	793130	<p>Represented as ANN{.N[N]}</p> <p>Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).</p>
Additional Diagnosis 10 DxAdd10	Char[6]	320	793130	<p>Represented as ANN{.N[N]}</p> <p>Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).</p>
Mental Health Legal Status LegalSt	Char[1]	326	727343	<p>1: Involuntary patient</p> <p>2: Voluntary patient</p> <p>9: Not reported/Unknown</p>
Marital Status MaritalSt	Char[1]	327	766507	<p>1: Never married</p> <p>2: Widowed</p> <p>3: Divorced</p> <p>4: Separated</p> <p>5: Married (registered and de facto)</p> <p>6: Not stated/inadequately described</p>
Area of Usual Residence ResArea	Char[9]	328	747315	<p>Statistical Area Level 2 (SA2) code (ASGS Edition 3) N(9)</p>

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Mental Health Care Referral Destination ReferralMHRC	Char[1]	337	723198	1: Specialised mental health admitted patient care 2: Specialised mental health residential care 3: Specialised mental health ambulatory care 4: Private psychiatrist care 5: General practitioner care 6: Other care 7: Not referred 8: Not applicable (i.e. end of reference period or died) 9: Unknown/not stated/inadequately described

Record length = 337

Notes

[10]

METEOR includes code 9, but that is not applicable to the RMHC NMDS

1.7. Data elements

1.7.1. Additional Diagnosis 1

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd01

METEOR Identifier:

793130

1.7.2. Additional Diagnosis 2

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd02

METEOR Identifier:

793130

1.7.3. Additional Diagnosis 3

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd03

METEOR Identifier:

793130

1.7.4. Additional Diagnosis 4

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd04

METEOR Identifier:

793130

1.7.5. Additional Diagnosis 5

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd05

METEOR Identifier:

793130

1.7.6. Additional Diagnosis 6

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd06

METEOR Identifier:

793130

1.7.7. Additional Diagnosis 7

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd07

METEOR Identifier:

793130

1.7.8. Additional Diagnosis 8

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd08

METEOR Identifier:

793130

1.7.9. Additional Diagnosis 9

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd09

METEOR Identifier:

793130

1.7.10. Additional Diagnosis 10

Definition:

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

Refers to a condition or complaint either coexisting with the principal diagnosis or arising during the episode of residential care. The additional diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxAdd10

METEOR Identifier:

793130

1.7.11. Area of Usual Residence

Definition:

Geographical region in which a person or group of people usually reside.

Domain:

Statistical Area Level 2 (SA2) code (ASGS Edition 3) N(9)

Field Name:

ResArea

METEOR Identifier:

747315

1.7.12. Batch Number

Definition:

Represents the YYYYNNNNN component of the extract file name.

Field Name:

BatchNo

1.7.13. Country of Birth

Definition:

The country in which the person was born.

Domain:

The country in which the person was born. To be provided in accordance with the Standard Australian Classification of Countries (SACC). ABS catalogue no. 1269.0 (2016). Values from 1601-1607, inclusive, are not permitted in this NMDS (Antarctica).

Field Name:

CoB

METEOR Identifier:

659454

1.7.14. Data File Generation Date

Definition:

Data file generation date

Domain:

Data file generation date

Valid date expressed as DDMMYYYY

Field Name:

GenDt

1.7.15. Data File Type

Definition:

Data file type

Domain:

Value = RMHC

Field Name:

FileType

1.7.16. Date of Birth

Definition:

The date of birth of the person.

Domain:

The date of birth of the person.

Valid date expressed as DDMMYYYY. For an estimated date of birth supply the estimated date in a valid format and if possible include the optional estimated date of birth flag. For an unknown date of birth use 09099999 when supplying data.

Field Name:

DoB

METEOR Identifier:

287007

1.7.17. Episode of Residential Care End Date

Definition:

Date on which resident formally or statistically ends an episode of residential care.

Domain:

Date on which resident formally or statistically ends an episode of residential care.

Valid date expressed as DDMMYYYY

Field Name:

EpiEndDt

METEOR Identifier:

723189

1.7.18. Episode of Residential Care End Mode

Definition:

Reason for end of episode of residential care.

Domain:

1:

Died

2:

Left against clinical advice/at own risk

3:

Did not return from leave

4:

Formal discharge from residential care at this establishment

5:

End of reference period

6:

Return to other residential health service

9:

Unknown/not stated/inadequately described

Field Name:

EpiEndMode

METEOR Identifier:

723192

1.7.19. Episode of Residential Care Start Date

Definition:

Valid date expressed as DDMMYYYY Date on which resident formally or statistically starts an episode of residential care.

Field Name:

EpiStartDt

METEOR Identifier:

723194

1.7.20. Episode of Residential Care Start Mode

Definition:

Reason for start of episode

Domain:**2:**

Start of new residential stay

3:

Start of a new reference period

4:

Start of expected short concurrent residential stay

9:

Unknown/not stated/inadequately described

Field Name:`EpiStartMode`**METEOR Identifier:**

723196

1.7.21. Estimated Date of Birth Flag**Definition:**

Flag to indicate whether the person's date of birth is accurate or an estimate

Domain:**1:**

Date of birth is accurate

2:

Date of birth is an estimate

8:

Date of birth is a "dummy" date (ie, 09099999)

9:

Accuracy of stated date of birth is not known

Optional data element providing additional information regarding the quality of date of birth data. Code 1 should be used when it is known that the reported date of birth is accurate, code 2 when it is known that one or more parts of the date of birth is an estimate, code 8 when birth date is unknown and a 'dummy' date of birth has been used (that is, 09099999), and code 9 when it is not known whether the date of birth is accurate or an estimate.

Field Name:`DoBFlag`**1.7.22. Gender****Domain:****1:**

Man, or boy, or male

2:

Woman, or girl, or female

3:

Non-binary

4:

Different term

5:

Prefer not to answer

9:

Not stated / inadequately described

Sex and Gender are conditional data elements. Data must be reported for at least one of the two elements, either Sex or Gender. Data may be reported for both. Where data is not collected for one of the data elements, this field should be reported as 'null' through space-filling.

Field Name:

Gender

METEOR Identifier:

741842

1.7.23. Indigenous Status

Definition:

Indigenous status

Domain:

1:

Aboriginal but not Torres Strait Islander origin

2:

Torres Strait Islander but not Aboriginal origin

3:

Both Aboriginal and Torres Strait Islander origin

4:

Neither Aboriginal and Torres Strait Islander origin

9:

Not stated/inadequately described

Field Name:

IndigSt

METEOR Identifier:

602543

1.7.24. Leave Days From Residential Care

Definition:

The total number of days (from midnight to midnight) spent on leave from a residential care service during an episode of residential care.

Domain:

Represented as NNN.

Field Name:

NLeaveDays

METEOR Identifier:

723200

1.7.25. Marital Status

Definition:

A person's current relationship status in terms of a couple relationship or, for those not in a couple relationship, the existence of a current or previous registered marriage, as represented by a code.

Domain:

1:

Never married

2:

Widowed

3:

Divorced

4:

Separated

5:

Married (registered and de facto)

6:

Not stated/inadequately described

Field Name:

MaritalSt

METEOR Identifier:

766507

1.7.26. Mental Health Care Referral Destination

Definition:

The type of health care the resident is referred to by the residential health care service for further care at the end of the residential stay.

Domain:

1:

Specialised mental health admitted patient care

2:

Specialised mental health residential care

3:

Specialised mental health ambulatory care

4:

Private psychiatrist care

5:

General practitioner care

6:

Other care

7:

Not referred

8:

Not applicable (i.e. end of reference period or died)

9:

Unknown/not stated/inadequately described

Field Name:

ReferralMHRC

METEOR Identifier:

723198

1.7.27. Mental Health Legal Status

Definition:

Whether a person is treated on an involuntary basis.

Domain:

1:

Involuntary patient

2:

Voluntary patient

9:

Not reported/Unknown

Field Name:

LegalSt

METEOR Identifier:

727343

1.7.28. Organisation Identifier

Definition:

Mental health service organisation identifier.

Domain:

A(9): Mental health service organisation identifier.

Identifiers used in this collection should map to the identifiers used in data for the NMDS for Mental Health Establishments.

Field Name:

OrgId

METEOR Identifier:

795837

1.7.29. Organisation Name

Definition:

Common name used to identify the Organisation

Field Name:

OrgName

METEOR Identifier:

405767

1.7.30. Person Identifier

Definition:

Person identifier is unique and stable for each individual patient within each service unit. Individual service units or collection authorities may use their own alphabetic, numeric or alphanumeric coding systems. The following special characters can be used: +, =, /

Field Name:

PersId

METEOR Identifier:

799014

1.7.31. Principal Diagnosis

Definition:

The diagnosis established after study to be chiefly responsible for occasioning an episode of residential care, as represented by a code. The principal diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Domain:

Represented as ANN{.N[N]}

The diagnosis established after study to be chiefly responsible for occasioning an episode of residential care, as represented by a code. The principal diagnosis must be a valid code from the International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM) (13th Edition) or from the ICD-10-AM Mental Health Manual: An integrated classification and diagnostic tool for community based mental health services (1st Edition).

Field Name:

DxPrinc

METEOR Identifier:

793125

1.7.32. Record Type

Definition:

A code indicating the type of each record included in a RMHC data file.

Domain:

CLUS:

Service unit cluster details

EPI:

Episode of residential care details

HR:

File header record

ORG:

Organisation details

PER:

Person details

REG:

Region details

SERV:

Service unit details

Field Name:

RecType

1.7.33. Region Identifier

Domain:

AA: Region (values as specified by individual jurisdiction)

Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.

Field Name:

RegId

METEOR Identifier:

269940

1.7.34. Region Name

Definition:

Common name used to identify the Region.

Field Name:

RegName

METEOR Identifier:

407187

1.7.35. Report Period End Date

Definition:

Report period end date

Domain:

Report period end date

Must be the end of the financial year, expressed as DDMMYYYY

Field Name:

RepEnd

1.7.36. Report Period Start Date

Definition:

Report period start date

Domain:

Report period start date

Must be the start of the financial year, expressed as DDMMYYYY

Field Name:

RepStart

1.7.37. Residential Service Unit Identifier

Domain:

A(9): Service unit identifier.

Identifiers used in this collection should map to the identifiers used in the NMDS for Mental Health Establishments.

Field Name:

SUIId

METEOR Identifier:

795859

1.7.38. Residential Service Unit Name

Definition:

Common name used to identify the service unit

Field Name:

SUName

METEOR Identifier:

722715

1.7.39. Residential Stay Start Date

Definition:

Date on which resident formally started a residential stay.

Domain:

Date on which resident formally started a residential stay.

Valid date expressed as DDMMYYYY

Field Name:

ResStartDt

METEOR Identifier:

723256

1.7.40. RMHC Specification Version Number

Definition:

The version number of the RMHC specification document used

Domain:

Value = 07.01

Field Name:

SpecVer

1.7.41. Sector

Definition:

Service unit sector

Domain:

1:

Public

2:

Private

Field Name:

Sector

METEOR Identifier:

269977

1.7.42. Service Unit Cluster Identifier

Domain:

A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDS for Mental Health Establishments, the identifiers used in this collection should be the same.

Field Name:

ClusId

METEOR Identifier:

795848

1.7.43. Service Unit Cluster Name

Definition:

Common name used to identify the service unit cluster.

Domain:

If no cluster applies, enter organisation name as appears in previous line.

Field Name:

ClusName

METEOR Identifier:

409209

1.7.44. Sex

Definition:

The sex of the person.

Domain:

1:

Male

2:

Female

3:

Another term

9:

Not stated / inadequately described

Sex and Gender are conditional data elements. Data must be reported for at least one of the two elements, either Sex or Gender. Data may be reported for both. Where data is not collected for one of the data elements, this field should be reported as 'null' through space-filling.

Field Name:

Sex

METEOR Identifier:

741686

1.7.45. State/Territory Identifier

Definition:

An identifier indicating the State or Territory responsible for the collection and submission of the RMHC data file.

Domain:

1:

New South Wales

2:

Victoria

3:

Queensland

4:

South Australia

5:

Western Australia

6:

Tasmania

7:

Northern Territory

8:

Australian Capital Territory

METEOR includes code 9, but that is not applicable to the RMHC NMDS

Field Name:

State

METEOR Identifier:

790405

1.7.46. State/Territory Record Identifier

Definition:

Unique Record Identifier used to identify records in discussion with jurisdictions (i.e. if problems are found).

This should be stable in the data collection of the jurisdiction.

Domain:

This can be formed using alphabetic, numeric or alphanumeric coding systems.

The following special characters can be used: +, =, /

Field Name:

RecordId

1.8. Virtual elements

1.8.1. EpiAge

Base:

EPI

Title:

Age

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       FLOOR((EpiEndDt - DoB) / 365.25) as Age
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DoB != '9999-09-09'
and DoBFlag in ('1', '2')
```

Rules:

- [BadDxAdd01Ad](#)
- [BadDxAdd01LowAge](#)
- [BadDxAdd01Ppm](#)
- [BadDxAdd02Ad](#)
- [BadDxAdd02LowAge](#)
- [BadDxAdd02Ppm](#)
- [BadDxAdd03Ad](#)

- [BadDxAdd03LowAge](#)
- [BadDxAdd03Ppm](#)
- [BadDxAdd04Ad](#)
- [BadDxAdd04LowAge](#)
- [BadDxAdd04Ppm](#)
- [BadDxAdd05Ad](#)
- [BadDxAdd05LowAge](#)
- [BadDxAdd05Ppm](#)
- [BadDxAdd06Ad](#)
- [BadDxAdd06LowAge](#)
- [BadDxAdd06Ppm](#)
- [BadDxAdd07Ad](#)
- [BadDxAdd07LowAge](#)
- [BadDxAdd07Ppm](#)
- [BadDxAdd08Ad](#)
- [BadDxAdd08LowAge](#)
- [BadDxAdd08Ppm](#)
- [BadDxAdd09Ad](#)
- [BadDxAdd09LowAge](#)
- [BadDxAdd09Ppm](#)
- [BadDxAdd10Ad](#)
- [BadDxAdd10LowAge](#)
- [BadDxAdd10Ppm](#)
- [BadDxPrincAd](#)
- [BadDxPrincLowAge](#)
- [BadDxPrincPpm](#)
- [HighAge](#)
- [LowAge](#)
- [LowAgeMarriage](#)

1.8.2. EpiDays

Base:

EPI

Title:

Episode Days

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       (EpiEndDt - EpiStartDt) - NLeaveDays as Days
from EPI
```

Rules:

- [LowEpiDays](#)

1.8.3. EpiPrevGap

Base:

EPI

Title:

Days Since End of Previous Episode

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiStartDt,
       lag(EpiEndDt) over PerEpi as prevEpiEndDt,
       EpiStartDt - lag(EpiEndDt) over PerEpi
from EPI window PerEpi as (partition by State, RegId, OrgId, ClusId, SUIId, PersId
order by EpiStartDt, EpiEndDt, RecordId)
```

Rules:

- [EpiAdjacentDate](#)
- [EpiDateOverlap](#)

1.8.4. HrCoBPropNotAus

Base:

HR

Title:

State Birth Country not Australia Proportion

SQL:

```
select State,
       sum((CoB not in ('1100', '1101', '1102', '1199'))::INT) / count(*)::FLOAT as prop
from PER
where CoB is not null
group by State
```

Rules:

- [HighExtCoBProp](#)

1.8.5. HrCoBPropSupp

Base:

HR

Title:

State Birth Country is Supplementary

SQL:

```
select State,
       sum(((CoB ~ '^d+ *' AND CoB::INT < 1000) OR CoB LIKE '%00'))::INT) /
count(*)::FLOAT as prop
from PER
where CoB is not null
group by State
```

Rules:

- [HighSuppCoBProp](#)

1.8.6. HrDoBFlagProp

Base:

HR

Title:

State Estimated DoB Proportion

SQL:

```
select State,
       sum((DoBFlag in ('2', '8', '9'))::INT) / count(*)::FLOAT as prop
from PER
where DoBFlag is not null
group by State
```

Rules:

- [HighEstDoBFlagProp](#)

1.8.7. HrIndigSt1Count

Base:

HR

Title:

HR Count for IndigSt 1

SQL:

```
select State,
       coalesce(Count, 0) as Count
from HR
left join (
  select State,
         count(*) as Count
  from PER
  where IndigSt = '1'
  group by State
) counts using (State)
```

1.8.8. HrIndigSt2Count

Base:

HR

Title:

HR Count for IndigSt 2

SQL:

```
select State,
       coalesce(Count, 0) as Count
from HR
left join (
  select State,
         count(*) as Count
  from PER
  where IndigSt = '2'
  group by State
) counts using (State)
```

1.8.9. HrIndigSt3Count

Base:

HR

Title:

HR Count for IndigSt 3

SQL:

```
select State,
       coalesce(Count, 0) as Count
from HR
left join (
  select State,
         count(*) as Count
  from PER
  where IndigSt = '3'
  group by State
) counts using (State)
```

1.8.10. HrIndigStPropNonAus

Base:

HR

Title:

State Indigenous Born Outside Australia Proportion

SQL:

```
select State,
       sum((CoB not in ('1100', '1101', '1102', '1199'))::INT) / count(*)::FLOAT as prop
from PER
where IndigSt in ('1', '2', '3')
       and CoB is not null
group by State
```

Rules:

- [HighIndigNonAustProp](#)

1.8.11. HrLegalSt1Count

Base:

HR

Title:

HR Count for LegalSt 1

SQL:

```
select State,
       coalesce(Count, 0) as Count
from HR
left join (
  select State,
         count(*) as Count
  from EPI
  where LegalSt = '1'
  group by State
) counts using (State)
```

1.8.12. HrLegalSt1Prop

Base:

HR

Title:

State Involuntary Legal Status Proportion

SQL:

```
select State,
       sum((LegalSt = '1')::INT) / count(*)::FLOAT as prop
from EPI
where LegalSt is not null
group by State
```

1.8.13. HrLegalSt2Count

Base:

HR

Title:

HR Count for LegalSt 2

SQL:

```
select State,
       coalesce(Count, 0) as Count
from HR
left join (
  select State,
         count(*) as Count
  from EPI
  where LegalSt = '2'
  group by State
) counts using (State)
```

1.8.14. HrLegalStProp

Base:

HR

Title:

State Missing Legal Status Proportion

SQL:

```
select State,
       sum((LegalSt = '9')::INT) / count(*)::FLOAT as prop
from EPI
where LegalSt is not null
group by State
```

1.8.15. HrRefProp9Mode4

Base:

HR

Title:

State Referral Unknown Code 9 Used with End Mode 4

SQL:

```
select State,
       sum((ReferralMHRC = '9')::INT) / count(*)::FLOAT as prop
from EPI
where EpiEndMode = '4'
group by State
```

1.8.16. HrResAreaProp

Base:

HR

Title:

State in-state ResArea Proportion

SQL:

```
select State,
       sum((State != substr(ResArea, 1, 1))::INT) / count(*)::FLOAT as prop
  from EPI
 where ResArea is not null
 group by State
```

Rules:

- [HighExtSA2Prop](#)

1.8.17. HrSector1Count

Base:

HR

Title:

HR Count for Sector 1

SQL:

```
select State,
       coalesce(Count, 0) as Count
  from HR
 left join (
    select State,
           count(*) as Count
      from SERV
     where Sector = '1'
    group by State
  ) counts using (State)
```

1.8.18. HrSector2Count

Base:

HR

Title:

HR Count for Sector 2

SQL:

```
select State,
       coalesce(Count, 0) as Count
  from HR
 left join (
    select State,
           count(*) as Count
      from SERV
     where Sector = '2'
    group by State
  ) counts using (State)
```

1.8.19. OrgCoBPropNotAus

Base:

ORG

Title:

Organisation Birth Country not Australia Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       sum((CoB not in ('1100', '1101', '1102', '1199'))::INT) / count(*)::FLOAT as prop
from PER
where CoB is not null
group by State,
       RegId,
       OrgId
```

1.8.20. OrgCoBPropSupp

Base:

ORG

Title:

Organisation Birth Country is Supplementary

SQL:

```
select State,
       RegId,
       OrgId,
       sum(((CoB ~ '^\\d+ *' AND CoB::INT < 1000) OR CoB LIKE '%00'))::INT) /
count(*)::FLOAT as prop
from PER
where CoB is not null
group by State,
       RegId,
       OrgId
```

1.8.21. OrgDoBFlagProp

Base:

ORG

Title:

Organisation Estimated DoB Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       sum((DoBFlag in ('2', '8', '9'))::INT) / count(*)::FLOAT as prop
from PER
where DoBFlag is not null
group by State,
       RegId,
       OrgId
```

1.8.22. OrgIndigSt1Count

Base:

ORG

Title:

ORG Count for IndigSt 1

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
from ORG
left join (
  select State,
         RegId,
         OrgId,
         count(*) as Count
  from PER
  where IndigSt = '1'
  group by State,
         RegId,
         OrgId
) counts using (State, RegId, OrgId)
```

1.8.23. OrgIndigSt2Count

Base:

ORG

Title:

ORG Count for IndigSt 2

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
from ORG
left join (
  select State,
         RegId,
         OrgId,
         count(*) as Count
  from PER
  where IndigSt = '2'
  group by State,
         RegId,
         OrgId
) counts using (State, RegId, OrgId)
```

1.8.24. OrgIndigSt3Count

Base:

ORG

Title:

ORG Count for IndigSt 3

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
from ORG
left join (
  select State,
         RegId,
         OrgId,
         count(*) as Count
  from PER
  where IndigSt = '3'
  group by State,
         RegId,
         OrgId
) counts using (State, RegId, OrgId)
```

1.8.25. OrgIndigStPropNonAus

Base:

ORG

Title:

Organisation Indigenous Born Outside Australia Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       sum((CoB not in ('1100', '1101', '1102', '1199'))::INT) / count(*)::FLOAT as prop
from PER
where IndigSt in ('1', '2', '3')
and CoB is not null
group by State,
       RegId,
       OrgId
```

1.8.26. OrgLegalSt1Count

Base:

ORG

Title:

ORG Count for LegalSt 1

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
from ORG
left join (
  select State,
         RegId,
         OrgId,
         count(*) as Count
  from EPI
  where LegalSt = '1'
  group by State,
         RegId,
         OrgId
) counts using (State, RegId, OrgId)
```

1.8.27. OrgLegalSt1Prop

Base:

ORG

Title:

Organisation Involuntary Legal Status Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       sum((LegalSt = '1')::INT) / count(*)::FLOAT as prop
from EPI
where LegalSt is not null
group by State,
       RegId,
       OrgId
```

Rules:

- [OrgInvolPropVaries](#)

1.8.28. OrgLegalSt2Count

Base:

ORG

Title:

ORG Count for LegalSt 2

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
from ORG
left join (
  select State,
         RegId,
         OrgId,
         count(*) as Count
  from EPI
  where LegalSt = '2'
  group by State,
         RegId,
         OrgId
) counts using (State, RegId, OrgId)
```

1.8.29. OrgLegalStProp

Base:

ORG

Title:

Organisation Missing Legal Status Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       sum((LegalSt = '9')::INT) / count(*)::FLOAT as prop
from EPI
where LegalSt is not null
group by State,
       RegId,
       OrgId
```

1.8.30. OrgRefProp9Mode4

Base:

ORG

Title:

Organisation Referral Unknown Code 9 Used with End Mode 4

SQL:

```
select State,
       RegId,
       OrgId,
       sum((ReferralMHRC = '9')::INT) / count(*)::FLOAT as prop
from EPI
where EpiEndMode = '4'
group by State,
       RegId,
       OrgId
```

1.8.31. OrgResAreaProp

Base:

ORG

Title:

Organisation in-state ResArea Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       sum((State != substr(ResArea, 1, 1))::INT) / count(*)::FLOAT as prop
from EPI
where ResArea is not null
group by State,
       RegId,
       OrgId
```

1.8.32. OrgSector1Count

Base:

ORG

Title:

ORG Count for Sector 1

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
from ORG
left join (
  select State,
         RegId,
         OrgId,
         count(*) as Count
  from SERV
  where Sector = '1'
  group by State,
         RegId,
         OrgId
) counts using (State, RegId, OrgId)
```

1.8.33. OrgSector2Count

Base:

ORG

Title:

ORG Count for Sector 2

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
from ORG
left join (
  select State,
         RegId,
         OrgId,
         count(*) as Count
  from SERV
  where Sector = '2'
  group by State,
         RegId,
         OrgId
) counts using (State, RegId, OrgId)
```

1.8.34. RegCoBPropNotAus

Base:

REG

Title:

Region Birth Country not Australia Proportion

SQL:

```
select State,
       RegId,
       sum((CoB not in ('1100', '1101', '1102', '1199'))::INT) / count(*)::FLOAT as prop
from PER
where CoB is not null
group by State,
       RegId
```

1.8.35. RegCoBPropSupp

Base:

REG

Title:

Region Birth Country is Supplementary

SQL:

```
select State,
       RegId,
       sum(((CoB ~ '^\\d+ *' AND CoB::INT < 1000) OR CoB LIKE '%00'))::INT) /
count(*)::FLOAT as prop
from PER
where CoB is not null
group by State,
       RegId
```

1.8.36. RegDoBFlagProp

Base:

REG

Title:

Region Estimated DoB Proportion

SQL:

```
select State,
       RegId,
       sum((DoBFlag in ('2','8','9'))::INT) / count(*)::FLOAT as prop
  from PER
 where DoBFlag is not null
 group by State,
        RegId
```

1.8.37. RegIndigSt1Count

Base:

REG

Title:

REG Count for IndigSt 1

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
  from REG
 left join (
  select State,
         RegId,
         count(*) as Count
    from PER
   where IndigSt = '1'
   group by State,
          RegId
 ) counts using (State, RegId)
```

1.8.38. RegIndigSt2Count

Base:

REG

Title:

REG Count for IndigSt 2

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
from REG
left join (
  select State,
         RegId,
         count(*) as Count
  from PER
  where IndigSt = '2'
  group by State,
         RegId
) counts using (State, RegId)
```

1.8.39. RegIndigSt3Count

Base:

REG

Title:

REG Count for IndigSt 3

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
from REG
left join (
  select State,
         RegId,
         count(*) as Count
  from PER
  where IndigSt = '3'
  group by State,
         RegId
) counts using (State, RegId)
```

1.8.40. RegIndigStPropNonAus

Base:

REG

Title:

Region Indigenous Born Outside Australia Proportion

SQL:

```
select State,
       RegId,
       sum((CoB not in ('1100', '1101', '1102', '1199'))::INT) / count(*)::FLOAT as prop
from PER
where IndigSt in ('1', '2', '3')
and CoB is not null
group by State,
       RegId
```

1.8.41. RegLegalSt1Count

Base:

REG

Title:

REG Count for LegalSt 1

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
from REG
left join (
  select State,
         RegId,
         count(*) as Count
  from EPI
  where LegalSt = '1'
  group by State,
         RegId
) counts using (State, RegId)
```

1.8.42. RegLegalSt1Prop

Base:

REG

Title:

Region Involuntary Legal Status Proportion

SQL:

```
select State,
       RegId,
       sum((LegalSt = '1')::INT) / count(*)::FLOAT as prop
from EPI
where LegalSt is not null
group by State,
       RegId
```

1.8.43. RegLegalSt2Count

Base:

REG

Title:

REG Count for LegalSt 2

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
from REG
left join (
  select State,
         RegId,
         count(*) as Count
  from EPI
  where LegalSt = '2'
  group by State,
         RegId
) counts using (State, RegId)
```

1.8.44. RegLegalStProp

Base:

REG

Title:

Region Missing Legal Status Proportion

SQL:

```
select State,
       RegId,
       sum((LegalSt = '9')::INT) / count(*)::FLOAT as prop
from EPI
where LegalSt is not null
group by State,
       RegId
```

1.8.45. RegRefProp9Mode4

Base:

REG

Title:

Region Referral Unknown Code 9 Used with End Mode 4

SQL:

```
select State,
       RegId,
       sum((ReferralMHRC = '9')::INT) / count(*)::FLOAT as prop
from EPI
where EpiEndMode = '4'
group by State,
       RegId
```

1.8.46. RegResAreaProp

Base:

REG

Title:

Region in-state ResArea Proportion

SQL:

```
select State,
       RegId,
       sum((State != substr(ResArea, 1, 1))::INT) / count(*)::FLOAT as prop
from EPI
where ResArea is not null
group by State,
       RegId
```

1.8.47. RegSector1Count**Base:**

REG

Title:

REG Count for Sector 1

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
from REG
left join (
  select State,
         RegId,
         count(*) as Count
  from SERV
  where Sector = '1'
  group by State,
         RegId
) counts using (State, RegId)
```

1.8.48. RegSector2Count**Base:**

REG

Title:

REG Count for Sector 2

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
from REG
left join (
  select State,
         RegId,
         count(*) as Count
  from SERV
  where Sector = '2'
  group by State,
         RegId
) counts using (State, RegId)
```

1.8.49. ResiStayDays

Base:

EPI

Title:

Residential Stay Days

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiEndDt - ResStartDt - NLeaveDays as Days
from EPI
```

Rules:

- [ResiStayLessThanZero](#)

1.8.50. ServCoBPropNotAus

Base:

SERV

Title:

Service Unit Birth Country not Australia Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum((CoB not in ('1100', '1101', '1102', '1199'))::INT) / count(*)::FLOAT as prop
from PER
where CoB is not null
group by State,
       RegId,
       OrgId,
       ClusId,
       SUIId
```

1.8.51. ServCoBPropSupp

Base:

SERV

Title:

Service Unit Birth Country is Supplementary

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum(((CoB ~ '^\\d+ *' AND CoB::INT < 1000) OR CoB LIKE '%00'))::INT) /
count(*)::FLOAT as prop
from PER
where CoB is not null
group by State,
       RegId,
       OrgId,
       ClusId,
       SUIId
```

1.8.52. ServDoBFlagProp

Base:

SERV

Title:

Service Unit Estimated DoB Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum((DoBFlag in ('2','8','9'))::INT) / count(*)::FLOAT as prop
from PER
where DoBFlag is not null
group by State,
       RegId,
       OrgId,
       ClusId,
       SUIId
```

1.8.53. ServIndigStPropNonAus

Base:

SERV

Title:

Service Unit Indigenous Born Outside Australia Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum((CoB not in ('1100','1101','1102','1199'))::INT) / count(*)::FLOAT as prop
from PER
where IndigSt in ('1','2','3')
       and CoB is not null
group by State,
       RegId,
       OrgId,
       ClusId,
       SUIId
```

1.8.54. ServLegalSt1Prop

Base:

SERV

Title:

Service Unit Involuntary Legal Status Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum((LegalSt = '1')::INT) / count(*)::FLOAT as prop
  from EPI
 where LegalSt is not null
 group by State,
         RegId,
         OrgId,
         ClusId,
         SUIId
```

1.8.55. ServLegalStProp

Base:

SERV

Title:

Service Unit Missing Legal Status Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum((LegalSt = '9')::INT) / count(*)::FLOAT as prop
  from EPI
 where LegalSt is not null
 group by State,
         RegId,
         OrgId,
         ClusId,
         SUIId
```

1.8.56. ServRefProp9Mode4

Base:

SERV

Title:

Service Unit Referral Unknown Code 9 Used with End Mode 4

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum((ReferralMHRC = '9')::INT) / count(*)::FLOAT as prop
  from EPI
 where EpiEndMode = '4'
 group by State,
        RegId,
        OrgId,
        ClusId,
        SUIId
```

Rules:

- [HighRefProp9Mode4](#)

1.8.57. ServResAreaProp

Base:

SERV

Title:

Service Unit in-state ResArea Proportion

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       sum((State != substr(ResArea, 1, 1))::INT) / count(*)::FLOAT as prop
  from EPI
 where ResArea is not null
 group by State,
        RegId,
        OrgId,
        ClusId,
        SUIId
```

1.9. Rules

1.9.1. BadDxAdd01Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () less than 15

Mark:

EPI.DxAdd01

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd01,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd01 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9 ')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd01
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.2. BadDxAdd01F**Class:**

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd01

Description:

The following diagnosis codes should only apply to females: 'F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3 ', 'F52.5 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd01,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd01 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd01
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.3. BadDxAdd01LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () less than 1

Mark:

EPI.DxAdd01

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd01,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd01 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0
', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd01
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.4. BadDxAdd01M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not male

Mark:

EPI.DxAdd01

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd01,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd01 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd01
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.5. BadDxAdd01Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd01

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd01,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd01 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd01
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.6. BadDxAdd02Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd02**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd02

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd02,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd02 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd02
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.7. BadDxAdd02F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd02

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd02,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd02 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd02
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.8. BadDxAdd02LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (`$DxAdd02`) and Age (`$Age`) less than 1

Mark:

EPI.DxAdd02

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd02,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd02 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0 ',
                 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
                 'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd02
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.9. BadDxAdd02M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not male

Mark:

EPI.DxAdd02

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd02,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd02 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd02
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.10. BadDxAdd02Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd02

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd02,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd02 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd02
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.11. BadDxAdd03Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd03**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd03

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd03,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd03 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd03
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.12. BadDxAdd03F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd03

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd03,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd03 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd03
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.13. BadDxAdd03LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd03**) and Age (**\$Age**) less than 1

Mark:

EPI.DxAdd03

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd03,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd03 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0
', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd03
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.14. BadDxAdd03M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not male

Mark:

EPI.DxAdd03

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd03,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd03 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd03
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.15. BadDxAdd03Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd03

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd03,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd03 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd03
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.16. BadDxAdd04Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd04**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd04

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd04,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd04 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd04
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.17. BadDxAdd04F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd04

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd04,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd04 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd04
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.18. BadDxAdd04LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (`$DxAdd04`) and Age (`$Age`) less than 1

Mark:

EPI.DxAdd04

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd04,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd04 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0
', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd04
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.19. BadDxAdd04M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not male

Mark:

EPI.DxAdd04

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd04,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd04 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd04
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.20. BadDxAdd04Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd04

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd04,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd04 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd04
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.21. BadDxAdd05Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd05**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd05

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd05,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd05 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd05
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.22. BadDxAdd05F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd05

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd05,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd05 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd05
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.23. BadDxAdd05LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (`$DxAdd05`) and Age (`$Age`) less than 1

Mark:

EPI.DxAdd05

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd05,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd05 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0 ',
                 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
                 'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd05
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.24. BadDxAdd05M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis (`$DxAdd05`) and Sex (`$Sex`) is not male

Mark:

EPI.DxAdd05

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd05,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd05 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd05
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.25. BadDxAdd05Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd05

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd05,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd05 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd05
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.26. BadDxAdd06Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd06**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd06

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd06,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd06 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd06
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.27. BadDxAdd06F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd06

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd06,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd06 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd06
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.28. BadDxAdd06LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd06**) and Age (**\$Age**) less than 1

Mark:

EPI.DxAdd06

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd06,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd06 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0
', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd06
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.29. BadDxAdd06M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not male

Mark:

EPI.DxAdd06

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd06,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd06 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd06
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.30. BadDxAdd06Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd06

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd06,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd06 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd06
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.31. BadDxAdd07Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd07**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd07

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd07,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd07 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd07
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.32. BadDxAdd07F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd07

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd07,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd07 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd07
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.33. BadDxAdd07LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (\$DxAdd07) and Age (\$Age) less than 1

Mark:

EPI.DxAdd07

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd07,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd07 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0
', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd07
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.34. BadDxAdd07M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not male

Mark:

EPI.DxAdd07

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd07,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd07 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd07
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.35. BadDxAdd07Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd07

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd07,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd07 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd07
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.36. BadDxAdd08Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd08**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd08

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd08,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd08 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd08
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.37. BadDxAdd08F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd08

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd08,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd08 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd08
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.38. BadDxAdd08LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd08**) and Age (**\$Age**) less than 1

Mark:

EPI.DxAdd08

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd08,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd08 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0
', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd08
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.39. BadDxAdd08M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis (`$DxAdd08`) and Sex (`$Sex`) is not male

Mark:

EPI.DxAdd08

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd08,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd08 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd08
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.40. BadDxAdd08Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd08

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd08,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd08 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd08
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.41. BadDxAdd09Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd09**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd09

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd09,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd09 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd09
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.42. BadDxAdd09F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd09

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd09,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd09 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd09
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.43. BadDxAdd09LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd09**) and Age (**\$Age**) less than 1

Mark:

EPI.DxAdd09

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0 ', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82 ', 'F83 ', 'F88 ', 'F89 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd09,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd09 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0
', 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd09
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.44. BadDxAdd09M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not male

Mark:

EPI.DxAdd09

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd09,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd09 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd09
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.45. BadDxAdd09Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd09

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd09,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd09 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd09
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.46. BadDxAdd10Ad

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd10**) and Age (**\$Age**) less than 15

Mark:

EPI.DxAdd10

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd10,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd10 in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxAdd10
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.47. BadDxAdd10F

Class:

Inconsistent

Priority:

High

Message:

Diagnosis () and Sex () is not female

Mark:

EPI.DxAdd10

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd10,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd10 in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd10
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.48. BadDxAdd10LowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxAdd10**) and Age (**\$Age**) less than 1

Mark:

EPI.DxAdd10

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd10,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd10 in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0 ',
                 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
                 'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxAdd10
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.49. BadDxAdd10M

Class:

Inconsistent

Priority:

High

Message:

Diagnosis (`$DxAdd10`) and Sex (`$Sex`) is not male

Mark:

EPI.DxAdd10

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd10,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxAdd10 in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxAdd10
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.50. BadDxAdd10Ppm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxAdd10

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxAdd10,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxAdd10 in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxAdd10
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.51. BadDxPrincAd

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxPrinc**) and Age (**\$Age**) less than 15

Mark:

EPI.DxPrinc

Description:

The following diagnosis codes should not apply to ages less than 15: 'F03', 'F01.0', 'F01.1', 'F01.2', 'F01.3', 'F01.8', 'F01.9'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxPrinc,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxPrinc in ('F03 ', 'F01.0 ', 'F01.1 ', 'F01.2 ', 'F01.3 ', 'F01.8 ', 'F01.9
')
and Age < 15
```

Data Elements:

- ClusId
- DxPrinc
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.52. BadDxPrincF

Class:

Inconsistent

Priority:

High

Message:

Diagnosis (`$DxPrinc`) and Sex (`$Sex`) is not female

Mark:

EPI.DxPrinc

Description:

The following diagnosis codes should only apply to females: 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3', 'F52.5'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxPrinc,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxPrinc in ('F53.0 ', 'F53.1 ', 'F53.8 ', 'F53.9 ', 'F32.01', 'F32.11',
                 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3 ', 'F52.5 ')
and Sex != '2'
and Sex is not null
```

Data Elements:

- ClusId
- DxPrinc
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.53. BadDxPrincLowAge

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis (**\$DxPrinc**) and Age (**\$Age**) less than 1

Mark:

EPI.DxPrinc

Description:

The following diagnosis codes should not apply to ages less than 1: 'F80.0', 'F80.1', 'F80.2', 'F80.3', 'F80.8', 'F80.9', 'F81.0', 'F81.1', 'F81.2', 'F81.3', 'F81.8', 'F81.9', 'F82', 'F83', 'F88', 'F89'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxPrinc,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxPrinc in ('F80.0 ', 'F80.1 ', 'F80.2 ', 'F80.3 ', 'F80.8 ', 'F80.9 ', 'F81.0 ',
                 'F81.1 ', 'F81.2 ', 'F81.3 ', 'F81.8 ', 'F81.9 ', 'F82  ', 'F83  ', 'F88  ',
                 'F89  ')
and Age < 1
```

Data Elements:

- ClusId
- DxPrinc
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.54. BadDxPrincM

Class:

Inconsistent

Priority:

High

Message:

Diagnosis (`$DxPrinc`) and Sex (`$Sex`) is not male

Mark:

EPI.DxPrinc

Description:

The following diagnosis codes should only apply to males: 'F52.4 '

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxPrinc,
       Sex
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DxPrinc in ('F52.4 ')
and Sex != '1'
and Sex is not null
```

Data Elements:

- ClusId
- DxPrinc
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- Sex
- State

1.9.55. BadDxPrincPpm

Class:

Inconsistent

Priority:

Low

Message:

Diagnosis () and Age () not between 10 and 60

Mark:

EPI.DxPrinc

Description:

The following diagnosis codes should only apply to ages 10-60: 'F53', 'F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01', 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', 'O99.3'

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxPrinc,
       Age
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where DxPrinc in ('F53.0', 'F53.1', 'F53.8', 'F53.9', 'F32.01',
                 'F32.11', 'F32.21', 'F32.31', 'F32.81', 'F32.91', '099.3')
and Age not between 10 and 60
```

Data Elements:

- ClusId
- DxPrinc
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.56. BadHrLegalSt12Counts

Class:

Anomaly

Priority:

High

Message:

LegalSt '1' count (\$LowCount) exceeds '2' count (\$HighCount)

Mark:

HR

Description:

Mental Health Legal Status (LegalSt) '1' (Involuntary patient) count exceeds '2' (Voluntary patient) count

SQL:

```
select State,
       sum((LegalSt = '1')::INT) LowCount,
       sum((LegalSt = '2')::INT) HighCount
from EPI
group by State
having sum((LegalSt = '1')::INT) > sum((LegalSt = '2')::INT)
```

Data Elements:

- [LegalSt](#)
- [State](#)

1.9.57. BadModelRef

Class:

Inconsistent

Priority:

High

Message:

EpiEndMode is `{EpiEndMode}` and ReferralMHRC is not 8 (`{ReferralMHRC.qt}`)

Mark:

EPI.EpiEndDt

Description:

EpiEndMode is 1 (Died) and ReferralMHRC is not 8 (Not applicable)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiEndDt,
       ReferralMHRC
from EPI
where EpiEndMode = '1'
and ReferralMHRC != '8'
```

Data Elements:

- [ClusId](#)
- [EpiEndDt](#)
- [EpiEndMode](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [ReferralMHRC](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.58. BadMode3StartDt

Class:

Inconsistent

Priority:

High

Message:

EpiStartMode is `$EpiStartMode` and EpiStartDt is not 1/July (`$EpiStartDt.dmy`)

Mark:

EPI.EpiStartDt

Description:

EpiStartMode is 3 (End of reference period) and EpiStartDt is not 1/July

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiStartDt,
       EpiStartMode
from EPI
where EpiStartMode = '3'
      and not ((extract(month FROM EpiStartDt) = 7) and (extract(day FROM EpiStartDt) =
1))
```

Data Elements:

- ClusId
- EpiStartDt
- EpiStartMode
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.59. BadMode5EndDt**Class:**

Inconsistent

Priority:

High

Message:

EndMode is `$EpiEndMode` and EpiEndDt is not 30/June (`$EpiEndDt.dmy`)

Mark:

EPI.EpiEndDt

Description:

EndMode is 5 (End of reference period) and EpiEndDt is not 30/June

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiEndDt,
       EpiEndMode
from EPI
where EpiEndMode = '5'
      and not ((extract(month FROM EpiEndDt) = 6) and (extract(day FROM EpiEndDt) = 30))
```

Data Elements:

- ClusId
- EpiEndDt
- EpiEndMode
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.60. BadMode5Ref

Class:

Inconsistent

Priority:

High

Message:

EndMode is 5 and ReferralMHRC is not 8 (\$ReferralMHRC)

Mark:

EPI.ReferralMHRC

Description:

EndMode is 5 (End of reference period) and ReferralMHRC is not 8

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       ReferralMHRC
from EPI
where EpiEndMode = '5'
      and ReferralMHRC != '8'
```

Data Elements:

- ClusId
- EpiEndMode
- OrgId
- PersId
- RecordId
- ReferralMHRC
- RegId
- SUIId
- State

1.9.61. BadSA2Prop**Class:**

Anomaly

Priority:

Low

Message:

Greater than 5% (`$prop.perc`) of SA2s are invalid (`$BadCount` in total)

Mark:

HR.State

Description:

Greater than 5% of SA2s (ResArea) are invalid

SQL:

```
select State,
       count(*) as AllCount,
       bad.BadCount,
       (bad.BadCount) / count(*)::FLOAT as prop
  from EPI
 cross join (
   select count(*) as BadCount
   from error
   join rule
   on (rule.id = error.rule)
   where rule.name = 'Domain'
   and error.field = 'ResArea'
 ) as bad
 group by State,
        bad.BadCount
 having (bad.BadCount) / count(*)::FLOAT > 0.05
```

Data Elements:

- ResArea
- State

1.9.62. ClusClusNameMissing**Class:**

Missing

Priority:

High

Message:

Missing data - ClusName `$ClusName.q`

Mark:

CLUS.ClusName

Description:

Missing data - Service Unit Cluster Name (ClusName)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ClusName
from CLUS
where ClusName is null
```

Data Elements:

- ClusId
- ClusName
- OrgId
- RegId
- State

1.9.63. ClusInSkIOnly

Class:

Skeleton

Priority:

High

Message:

Clus `$name` expected from SKL is missing

Description:

Service Unit Cluster (CLUS) appears in skeleton data only - A Service Unit Cluster (CLUS) with matching Ids is expected based on the SKL data but is not present in this file

1.9.64. ClusNotInSkI

Class:

Skeleton

Priority:

High

Message:

Clus `$name` not in SKL data

Description:

Service Unit Cluster not in skeleton reference data - A matching Residential Service Unit Cluster (CLUS) was not found in the skeleton data

1.9.65. EpiAdjacentDate

Class:

Anomaly

Priority:

High

Message:

EpiStartDt `$EpiStartDt.dmy` adjacent to EpiEndDt `$prevEpiEndDt.dmy` of earlier EPI

Mark:

EPI.StartDt

Description:

Start date for and episode of care is immediately adjacent to the end date of the preceding episode of care for a unique PersId (within a unit).

SQL:

```
select *
  from EpiPrevGap
 where EpiStartDt = 1 + prevEpiEndDt
```

Data Elements:

- [EpiStartDt](#)

Virtual Elements:

- [EpiPrevGap](#)

1.9.66. EpiDateOverlap

Class:

Inconsistent

Priority:

High

Message:

EpiStartDt `$EpiStartDt.dmy` on/before EpiEndDt `$prevEpiEndDt.dmy` of earlier EPI

Mark:

EPI.StartDt

Description:

Start date for and episode of care is before or equal to the end date for another episode of care for a unique PersId (within a unit).

SQL:

```
select *
  from EpiPrevGap
 where EpiStartDt <= prevEpiEndDt
```

Data Elements:

- [EpiStartDt](#)

Virtual Elements:

- [EpiPrevGap](#)

1.9.67. EpiDxPrincMissing

Class:

Missing

Priority:

High

Message:

Missing data - DxPrinc `$DxPrinc.q`

Mark:

EPI.DxPrinc

Description:

Missing data - Principal Diagnosis (DxPrinc)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       DxPrinc
from EPI
where DxPrinc is null
```

Data Elements:

- [ClusId](#)
- [DxPrinc](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.68. EpiEndBeforeResStart

Class:

Inconsistent

Priority:

High

Message:

EpiEndDt (`$EpiEndDt`) is before ResStartDt (`$ResStartDt`)

Mark:

EPI.EpiEndDt

Description:

The EpiStartDt cannot be after the ResEndDt

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiEndDt,
       ResStartDt
from EPI
where EpiEndDt < ResStartDt
```

Data Elements:

- ClusId
- EpiEndDt
- OrgId
- PersId
- RecordId
- RegId
- ResStartDt
- SUIId
- State

1.9.69. EpiEndDtBeforeDoB

Class:

Inconsistent

Priority:

High

Message:

EpiEndDt (`$EpiEndDt.dmy`) is before DoB (`$DoB.dmy`)

Mark:

EPI.EpiEndDt

Description:

EpiEndDt is before DoB

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiEndDt,
       DoB
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DoB != '9999-09-09'
and EpiEndDt < DoB
```

Data Elements:

- ClusId
- DoB
- EpiEndDt
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.70. EpiEndModeNot5**Class:**

Anomaly

Priority:

High

Message:

EpiEndDt is 30/June and EpiEndMode is not 5

Mark:

EPI.EpiEndMode

Description:

EpiEndDt is 30/June EpiEndMode is not 5 (End of reference period)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId
from EPI
where EpiEndMode != '5'
      and ((extract(month FROM EpiEndDt) = 6) and (extract(day FROM EpiEndDt) = 30))
```

Data Elements:

- ClusId
- EpiEndDt
- EpiEndMode
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.71. EpiEpiEndDtMissing

Class:

Missing

Priority:

High

Message:

Missing data - EpiEndDt `$EpiEndDt.q`

Mark:

EPI.EpiEndDt

Description:

Missing data - Episode of Residential Care End Date (EpiEndDt)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiEndDt
from EPI
where EpiEndDt is null
```

Data Elements:

- [ClusId](#)
- [EpiEndDt](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.72. EpiEpiEndModeMissing

Class:

Missing

Priority:

High

Message:

Missing data - EpiEndMode `$EpiEndMode.q`

Mark:

EPI.EpiEndMode

Description:

Missing data - Episode of Residential Care End Mode (EpiEndMode)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiEndMode
from EPI
where EpiEndMode is null
```

Data Elements:

- [ClusId](#)
- [EpiEndMode](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.73. EpiEpiStartDtMissing

Class:

Missing

Priority:

High

Message:

Missing data - EpiStartDt `$EpiStartDt.q`

Mark:

EPI.EpiStartDt

Description:

Missing data - Episode of Residential Care Start Date (EpiStartDt)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiStartDt
from EPI
where EpiStartDt is null
```

Data Elements:

- [ClusId](#)
- [EpiStartDt](#)

- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.74. EpiEpiStartModeMissing

Class:

Missing

Priority:

High

Message:

Missing data - EpiStartMode `$EpiStartMode.q`

Mark:

EPI.EpiStartMode

Description:

Missing data - Episode of Residential Care Start Mode (EpiStartMode)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiStartMode
from EPI
where EpiStartMode is null
```

Data Elements:

- ClusId
- EpiStartMode
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.75. EpiLegalStMissing

Class:

Missing

Priority:

High

Message:

Missing data - LegalSt `$LegalSt.q`

Mark:

EPI.LegalSt

Description:

Missing data - Mental Health Legal Status (LegalSt)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       LegalSt
from EPI
where LegalSt is null
```

Data Elements:

- ClusId
- LegalSt
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.76. EpiMaritalStMissing

Class:

Missing

Priority:

High

Message:

Missing data - MaritalSt `$MaritalSt.q`

Mark:

EPI.MaritalSt

Description:

Missing data - Marital Status (MaritalSt)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       MaritalSt
from EPI
where MaritalSt is null
```

Data Elements:

- [ClusId](#)
- [MaritalSt](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.77. EpiNLeaveDaysMissing

Class:

Missing

Priority:

High

Message:

Missing data - NLeaveDays `$NLeaveDays.q`

Mark:

EPI.NLeaveDays

Description:

Missing data - Leave Days From Residential Care (NLeaveDays)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       NLeaveDays
from EPI
where NLeaveDays is null
```

Data Elements:

- [ClusId](#)
- [NLeaveDays](#)

- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.78. EpiReferralMHRCMissing

Class:

Missing

Priority:

High

Message:

Missing data - ReferralMHRC `$ReferralMHRC.q`

Mark:

EPI.ReferralMHRC

Description:

Missing data - Mental Health Care Referral Destination (ReferralMHRC)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       ReferralMHRC
from EPI
where ReferralMHRC is null
```

Data Elements:

- ClusId
- OrgId
- PersId
- RecordId
- ReferralMHRC
- RegId
- SUIId
- State

1.9.79. EpiResAreaMissing

Class:

Missing

Priority:

High

Message:

Missing data - ResArea `$ResArea.q`

Mark:

EPI.ResArea

Description:

Missing data - Area of Usual Residence (ResArea)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       ResArea
from EPI
where ResArea is null
```

Data Elements:

- ClusId
- OrgId
- PersId
- RecordId
- RegId
- ResArea
- SUIId
- State

1.9.80. EpiResStartDtMissing

Class:

Missing

Priority:

High

Message:

Missing data - ResStartDt `$ResStartDt.q`

Mark:

EPI.ResStartDt

Description:

Missing data - Residential Stay Start Date (ResStartDt)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       ResStartDt
from EPI
where ResStartDt is null
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [ResStartDt](#)
- [SUIId](#)
- [State](#)

1.9.81. EpiStartDtBeforeDoB

Class:

Inconsistent

Priority:

High

Message:

EpiStartDt ([\\$EpiStartDt.dmy](#)) is before DoB ([\\$DoB.dmy](#))

Mark:

EPI.EpiStartDt

Description:

EpiStartDt is before DoB

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiStartDt,
       DoB
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DoB != '9999-09-09'
and EpiStartDt < DoB
```

Data Elements:

- [ClusId](#)
- [DoB](#)
- [EpiStartDt](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.82. EpiStartModeMismatch

Class:

Anomaly

Priority:

High

Message:

EpiEndMode 6 but EpiStartMode is not 4 (`$EpiStartMode`).

Mark:

EPI.EpiStartMode

Description:

EpiEndMode 6 but EpiStartMode is not 4.

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiStartMode
from EPI
where EpiEndMode='6'
      and EpiStartMode != '4';
```

Data Elements:

- [ClusId](#)
- [EpiEndMode](#)
- [EpiStartMode](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.83. EpiStartModeNot3

Class:

Anomaly

Priority:

High

Message:

EpiStartDt is 1/July and EpiStartMode is not 3

Mark:

EPI.EpiStartMode

Description:

EpiStartDt is 1/July and EpiStartMode is not 3 (Start of a new reference period)

SQL:

```
select State,
        RegId,
        OrgId,
        ClusId,
        SUIId,
        PersId,
        RecordId
from EPI
where EpiStartMode != '3'
      and ((extract(month FROM EpiStartDt) = 7) and (extract(day FROM EpiStartDt) = 1))
```

Data Elements:

- ClusId
- EpiStartDt
- EpiStartMode
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

1.9.84. HighAge

Class:

Anomaly

Priority:

Low

Message:

Age is greater than 124 years (\$Age)

Mark:

EPI

Description:

Age is greater than 124 years

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       Age
from EpiAge
where Age > 124
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

Virtual Elements:

- [EpiAge](#)

1.9.85. HighEstDoBFlagProp

Class:

Anomaly

Priority:

Low

Message:

Greater than 5% ([\\$prop.perc](#)) of Persons have an Estimated Date of Birth

Mark:

HR.State

Description:

Greater than 5% of Persons have an Estimated Date of Birth

SQL:

```
select State,
       prop
from HrDoBFlagProp
where prop > 0.05
```

Data Elements:

- [State](#)

Virtual Elements:

- [HrDoBFlagProp](#)

1.9.86. HighExtCoBProp

Class:

Anomaly

Priority:

High

Message:

Greater than 50% (`$prop.perc`) of Persons have a CoB other than Australia

Mark:

HR.State

Description:

Greater than 50% of Persons have a CoB other than Australia

SQL:

```
select State,
        prop
  from HrCoBPropNotAus
 where prop > 0.5
```

Data Elements:

- [State](#)

Virtual Elements:

- [HrCoBPropNotAus](#)

1.9.87. HighExtSA2Prop

Class:

Anomaly

Priority:

Low

Message:

Greater than 5% (`$prop.perc`) of Episodes have a ResArea out of state

Mark:

HR.State

Description:

Greater than 5% of Episodes a ResArea out of state. (This check does not apply to ACT)

SQL:

```
select State,
        prop
  from HrResAreaProp
 where State != '8'
        and prop > 0.05
```

Data Elements:

- [State](#)

Virtual Elements:

- [HrResAreaProp](#)

1.9.88. HighIndigNonAustProp

Class:

Anomaly

Priority:

High

Message:

Proportion of IndigSt not born in Australia is greater than 5% (`$prop.perc`)

Mark:

HR.State

Description:

Proportion of Indigenous Australians not born in Australia is greater than 5%

SQL:

```
select State,
        prop
  from HrIndigStPropNonAus
 where prop > 0.05
```

Data Elements:

- [State](#)

Virtual Elements:

- [HrIndigStPropNonAus](#)

1.9.89. HighRefProp9Mode4

Class:

Anomaly

Priority:

High

Message:

Proportion of EpiEndMode=4 with ReferralMHRC=9 is over than 50% (`$prop.perc`)

Mark:

SERV

Description:

Proportion of Episodes with EpiEndMode 4 (Formal discharge from residential care at this establishm...) having ReferralMHRC 9 (Unknown/not stated/inadequately described) at Service unit level is greater than 50%

SQL:

```
select State,
        RegId,
        OrgId,
        ClusId,
        SUIId,
        prop
  from ServRefProp9Mode4
 where prop > 0.5
```

Data Elements:

- [ClusId](#)
- [OrgId](#)

- [RegId](#)
- [SUIId](#)
- [State](#)

Virtual Elements:

- [ServRefProp9Mode4](#)

1.9.90. HighSuppCoBProp

Class:

Anomaly

Priority:

Low

Message:

Greater than 10% (`$prop.perc`) of Persons have a CoB with Supplementary code

Mark:

HR.State

Description:

Greater than 10% of Persons have a CoB with Supplementary code

SQL:

```
select State,
        prop
  from HrCoBPropSupp
 where prop > 0.10
```

Data Elements:

- [State](#)

Virtual Elements:

- [HrCoBPropSupp](#)

1.9.91. HrGenDtMissing

Class:

Missing

Priority:

High

Message:

Missing data - GenDt `$GenDt.q`

Mark:

HR.GenDt

Description:

Missing data - Data File Generation Date (GenDt)

SQL:

```
select State,
        GenDt
  from HR
 where GenDt is null
```

Data Elements:

- [GenDt](#)
- [State](#)

1.9.92. InvalidZeroEpiDays**Class:**

Inconsistent

Priority:

Low

Message:

Episode length is equal to zero without valid EpiStartMode and EpiEndMode.

Mark:

EPI

Description:

Episode length is equal to zero, with EpiStartMode!=3 and if EpiEndMode=5 then the start date is not 30 June.

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId
  from EPI
 where ((EpiEndDt - EpiStartDt - NLeaveDays = 0) and EpiStartMode != '3' and
        (EpiEndMode != '5' or (EpiEndMode='5' and not (extract(month FROM
        EpiStartDt)::INTEGER = 6 and extract(day FROM EpiStartDt)::INTEGER = 30))));
```

Data Elements:

- [ClusId](#)
- [EpiEndDt](#)
- [EpiEndMode](#)
- [EpiStartDt](#)
- [EpiStartMode](#)
- [NLeaveDays](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.93. LowAge**Class:**

Anomaly

Priority:

Low

Message:

Age is less than 10 years (**\$Age**)

Mark:

EPI

Description:

Age at Episode End is less than 10 years

SQL:

```
select State,
        RegId,
        OrgId,
        ClusId,
        SUIId,
        PersId,
        RecordId,
        Age
from EpiAge
where Age < 10
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

Virtual Elements:

- [EpiAge](#)

1.9.94. LowAgeMarriage

Class:

Anomaly

Priority:

Low

Message:

Age is less than 16 years (**\$Age**) and MaritalSt is **\$MaritalSt**

Mark:

EPI.MaritalSt

Description:

Age is less than 16 years and Marital Status is not 1 (Never married)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       Age,
       MaritalSt
from EPI
join EpiAge using (State, RegId, OrgId, ClusId, SUIId, PersId, RecordId)
where Age < 16
and MaritalSt != '1'
```

Data Elements:

- ClusId
- MaritalSt
- OrgId
- PersId
- RecordId
- RegId
- SUIId
- State

Virtual Elements:

- EpiAge

1.9.95. LowEpiDays

Class:

Inconsistent

Priority:

High

Message:

Episode Days is less than 0 days ()

Mark:

EPI.NLeaveDays

Description:

Episode Days is less than 0 days

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       Days
from EpiDays
where Days < 0
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

Virtual Elements:

- [EpiDays](#)

1.9.96. OrgInSkIOnly

Class:

Skeleton

Priority:

High

Message:

Org [\\$name](#) expected from SKL is missing

Description:

Organisation (ORG) appears in skeleton data only - A Organisation (ORG) with matching Ids is expected based on the SKL data but is not present in this file

1.9.97. OrgInvolPropVaries

Class:

Historical

Priority:

Medium

Message:

OrgLegalSt1Prop variation of [\\$PercChange](#) percent.

Mark:

ORG

Description:

Variation over 10 percentage points in organisation-wide ratio of Involuntary legal status.

SQL:

```
select State,
       RegId,
       OrgId,
       round(100::float * abs(New.prop - Old.prop)) as PercChange
from OrgLegalSt1Prop as New
join hist.OrgLegalSt1Prop as Old using(State, RegId, OrgId)
where abs(New.prop - Old.prop) > 0.10;
```

Data Elements:

- [OrgId](#)
- [RegId](#)

- [State](#)

Virtual Elements:

- [OrgLegalSt1Prop](#)

1.9.98. OrgNotInSkl

Class:

Skeleton

Priority:

High

Message:

Org \$name not in SKL data

Description:

Organisation not in skeleton reference data - A matching Residential Organisation (ORG) was not found in the skeleton data

1.9.99. OrgOrgNameMissing

Class:

Missing

Priority:

High

Message:

Missing data - OrgName \$OrgName.q

Mark:

ORG.OrgName

Description:

Missing data - Organisation Name (OrgName)

SQL:

```
select State,
       RegId,
       OrgId,
       OrgName
from   ORG
where  OrgName is null
```

Data Elements:

- [OrgId](#)
- [OrgName](#)
- [RegId](#)
- [State](#)

1.9.100. PerCoBDiffers

Class:

Inconsistent

Priority:

High

Message:

Person has `$attr_count` values for CoB (`$attr_vals`)

Mark:

PER.CoB

Description:

Person has multiple values for CoB (Country of Birth) across one organisation

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       attr_count,
       attr_vals
from PER
join (
  select State,
         RegId,
         OrgId,
         PersId,
         count(DISTINCT CoB) as attr_count,
         string_agg(DISTINCT CoB::TEXT, ',') as attr_vals
  from PER
  group by State,
           RegId,
           OrgId,
           PersId
  having count(DISTINCT CoB) > 1
) counts using (State, RegId, OrgId, PersId)
```

Data Elements:

- [ClusId](#)
- [CoB](#)
- [OrgId](#)
- [PersId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.101. PerCoBMissing**Class:**

Missing

Priority:

High

Message:

Missing data - CoB `$CoB.q`

Mark:

PER.CoB

Description:

Missing data - Country of Birth (CoB)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       CoB
from PER
where CoB is null
```

Data Elements:

- ClusId
- CoB
- OrgId
- PersId
- RegId
- SUIId
- State

1.9.102. PerDoBCount**SQL:**

```
select sum(DoBCount) as PerDoBCountTotal,
       avg(DoBCount) as PerDoBCountAvg
from (
  select count(*) as DoBCount
  from PER
  where DoB != '9999-09-09'
  group by DoB
) as tmpcounts
```

Data Elements:

- DoB

1.9.103. PerDoBDiffers**Class:**

Inconsistent

Priority:

High

Message:

Person has `$attr_count` values for DoB (`$attr_vals`)

Mark:

PER.DoB

Description:

Person has multiple values for DoB (Date of Birth) across one organisation

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       attr_count,
       attr_vals
from PER
join (
  select State,
         RegId,
         OrgId,
         PersId,
         count(DISTINCT DoB) as attr_count,
         string_agg(DISTINCT DoB::TEXT, ',') as attr_vals
  from PER
  group by State,
           RegId,
           OrgId,
           PersId
  having count(DISTINCT DoB) > 1
) counts using (State, RegId, OrgId, PersId)
```

Data Elements:

- [ClusId](#)
- [DoB](#)
- [OrgId](#)
- [PersId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.104. PerDoBFlagDiffers**Class:**

Inconsistent

Priority:

High

Message:

Person has `$attr_count` values for DoBFlag (`$attr_vals`)

Mark:

PER.DoBFlag

Description:

Person has multiple values for DoBFlag (Estimated Date of Birth Flag) across one organisation

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       attr_count,
       attr_vals
from PER
join (
  select State,
         RegId,
         OrgId,
         PersId,
         count(DISTINCT DoBFlag) as attr_count,
         string_agg(DISTINCT DoBFlag::TEXT, ',') as attr_vals
  from PER
  group by State,
           RegId,
           OrgId,
           PersId
  having count(DISTINCT DoBFlag) > 1
) counts using (State, RegId, OrgId, PersId)
```

Data Elements:

- ClusId
- DoBFlag
- OrgId
- PersId
- RegId
- SUIId
- State

1.9.105. PerDoBFlagMissing

Class:

Missing

Priority:

High

Message:

Missing data - DoBFlag `DoBFlag.q`

Mark:

PER.DoBFlag

Description:

Missing data - Estimated Date of Birth Flag (DoBFlag)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       DoBFlag
from PER
where DoBFlag is null
```

Data Elements:

- [ClusId](#)
- [DoBFlag](#)
- [OrgId](#)
- [PersId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.106. PerDoBMissing

Class:

Missing

Priority:

High

Message:

Missing data - DoB

Mark:

PER.DoB

Description:

Missing data - Date of Birth (DoB)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       DoB
from PER
where DoB is null
```

Data Elements:

- [ClusId](#)
- [DoB](#)
- [OrgId](#)
- [PersId](#)

- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.107. PerIndigStDiffers

Class:

Inconsistent

Priority:

High

Message:

Person has `$attr_count` values for IndigSt (`$attr_vals`)

Mark:

PER.IndigSt

Description:

Person has multiple values for IndigSt (Indigenous Status) across one organisation

SQL:

```

select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       attr_count,
       attr_vals
from PER
join (
  select State,
         RegId,
         OrgId,
         PersId,
         count(DISTINCT IndigSt) as attr_count,
         string_agg(DISTINCT IndigSt::TEXT, ',') as attr_vals
  from PER
  group by State,
           RegId,
           OrgId,
           PersId
  having count(DISTINCT IndigSt) > 1
) counts using (State, RegId, OrgId, PersId)

```

Data Elements:

- [ClusId](#)
- [IndigSt](#)
- [OrgId](#)
- [PersId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.108. PerIndigStMissing

Class:

Missing

Priority:

High

Message:

Missing data - IndigSt `$IndigSt.q`

Mark:

PER.IndigSt

Description:

Missing data - Indigenous Status (IndigSt)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       IndigSt
from PER
where IndigSt is null
```

Data Elements:

- [ClusId](#)
- [IndigSt](#)
- [OrgId](#)
- [PersId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

1.9.109. PerSexDiffers

Class:

Inconsistent

Priority:

High

Message:

Person has `$attr_count` values for Sex (`$attr_vals`)

Mark:

PER.Sex

Description:

Person has multiple values for Sex (Sex) across one organisation

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       attr_count,
       attr_vals
from PER
join (
  select State,
         RegId,
         OrgId,
         PersId,
         count(DISTINCT Sex) as attr_count,
         string_agg(DISTINCT Sex::TEXT, ',') as attr_vals
  from PER
  group by State,
           RegId,
           OrgId,
           PersId
  having count(DISTINCT Sex) > 1
) counts using (State, RegId, OrgId, PersId)
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [PersId](#)
- [RegId](#)
- [SUIId](#)
- [Sex](#)
- [State](#)

1.9.110. PerSexGenderMissing

Class:

Missing

Priority:

High

Message:

Missing data - at least one of Sex or Gender required

Mark:

PER

Description:

Missing data - at least one of Sex (Sex) or Gender (Gender) required

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       Sex,
       Gender
  from PER
 where (coalesce(cast(Sex in ('1','2','3','9') as integer),0) + coalesce(cast(Gender
 in ('1','2','3','4','5','9') as integer),0) + 0) = 0
```

Data Elements:

- ClusId
- Gender
- OrgId
- PersId
- RegId
- SUIId
- Sex
- State

1.9.111. PropResAreaUnknown

Class:

Anomaly

Priority:

Low

Message:

Greater than 10% of episodes within a service unit have a ResArea that is coded as unknown or no usual address (of).

Mark:

SERV

Description:

Greater than 10% of episodes within a service unit have a ResArea that is coded as unknown or no usual address.

SQL:

```
with epi_counts as(
  select State,
         RegId,
         OrgId,
         ClusId,
         SUIId,
         count(*) as total,
         sum(CASE WHEN ResArea ~ '^\.99999(497|498|499|999)$' THEN 1 ELSE 0 END)
as unknowns
  from EPI
  group by State,
         RegId,
         OrgId,
         ClusId,
         SUIId
) select State,
RegId,
OrgId,
ClusId,
SUIId,
unknowns,
total
from epi_counts
where unknowns > 0
and total > 0
and sd_div(unknowns, total, 3) > 0.10;
```

Data Elements:

- ClusId
- OrgId
- RegId
- ResArea
- SUIId
- State

1.9.112. RegInSkIOnly

Class:

Skeleton

Priority:

High

Message:

Reg \$name expected from SKL is missing

Description:

Region (REG) appears in skeleton data only - A Region (REG) with matching Ids is expected based on the SKL data but is not present in this file

1.9.113. RegNotInSkl

Class:

Skeleton

Priority:

High

Message:

Reg `$name` not in SKL data

Description:

Region not in skeleton reference data - A matching Residential Region (REG) was not found in the skeleton data

1.9.114. RegRegNameMissing

Class:

Missing

Priority:

High

Message:

Missing data - RegName `$RegName.q`

Mark:

REG.RegName

Description:

Missing data - Region Name (RegName)

SQL:

```
select State,
        RegId,
        RegName
from REG
where RegName is null
```

Data Elements:

- [RegId](#)
- [RegName](#)
- [State](#)

1.9.115. ResStartBeforeEpiStart

Class:

Inconsistent

Priority:

High

Message:

ResStartDt (`$ResStartDt`) must not be later than EpiStartDt (`$EpiStartDt`)

Mark:

EPI.ResStartDt

Description:

The ResStartDt cannot be after the EpiStartDt

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       EpiStartDt,
       ResStartDt
from EPI
where ResStartDt > EpiStartDt
```

Data Elements:

- ClusId
- EpiStartDt
- OrgId
- PersId
- RecordId
- RegId
- ResStartDt
- SUIId
- State

1.9.116. ResStartDtBeforeDoB

Class:

Inconsistent

Priority:

High

Message:

ResStartDt (`$ResStartDt.dmy`) is before DoB (`$DoB.dmy`)

Mark:

EPI.ResStartDt

Description:

ResStartDt is before DoB

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       ResStartDt,
       DoB
from EPI
join PER using (State, RegId, OrgId, ClusId, SUIId, PersId)
where DoB != '9999-09-09'
and ResStartDt < DoB
```

Data Elements:

- [ClusId](#)
- [DoB](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [ResStartDt](#)
- [SUIId](#)
- [State](#)

1.9.117. ResiStayLessThanZero**Class:**

Anomaly

Priority:

Low

Message:

Residential stay is less than zero (days).

Mark:

EPI

Description:

Residential stay is less than zero.

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       PersId,
       RecordId,
       days
from ResiStayDays
where days < 0;
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [PersId](#)
- [RecordId](#)
- [RegId](#)
- [SUIId](#)
- [State](#)

Virtual Elements:

- [ResiStayDays](#)

1.9.118. ServClosed

Class:

Historical

Priority:

High

Message:

Serv closed, historical `$hist_name.qt` (SUIId: `$hist_SUIId`) no longer exists

Mark:

HR.State

Description:

Service Unit Closed - A historical Service Unit was not found in current data

SQL:

```
select State,
       hist_entity.State as hist_State,
       hist_entity.RegId as hist_RegId,
       hist_entity.OrgId as hist_OrgId,
       hist_entity.ClusId as hist_ClusId,
       hist_entity.SUIId as hist_SUIId,
       hist_entity.SUName as hist_name
from hist.SERV as hist_entity
left join main.SERV using (State, RegId, OrgId, ClusId, SUIId)
where SERV.SUIId is null
```

Data Elements:

- ClusId
- OrgId
- RegId
- SUIId
- SUName
- State

1.9.119. ServInSkOnly

Class:

Skeleton

Priority:

High

Message:

Serv `$name` expected from SKL is missing

Description:

Service Unit (RESI) with sector 1 (public) appears in skeleton data only - A Service Unit (SERV) with matching Ids is expected based on the SKL data but is not present in this file

1.9.120. ServNotInSk

Class:

Skeleton

Priority:

High

Message:Serv \$name not in SKL data**Description:**

Service Unit not in skeleton reference data - A matching Residential Service Unit (RESI) was not found in the skeleton data

1.9.121. ServOpened

Class:

Historical

Priority:

High

Message:Serv opened, \$\$SUName.qt (SUIId: \$SUIId) not in historical data**Mark:**

SERV

Description:

Service Unit Opened - A matching Service Unit was not found in the historical data

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       SERV.SUName
from   main.SERV
left join hist.SERV as hist_entity using (State, RegId, OrgId, ClusId, SUIId)
where hist_entity.SUIId is null
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [RegId](#)
- [SUIId](#)
- [SUName](#)
- [State](#)

1.9.122. ServRenamed

Class:

Historical

Priority:

High

Message:Serv renamed from \$hist_name.qt to \$SUName.qt

Mark:

SERV.SUName

Description:

Service Unit Renamed - Service Unit Name differs between historical and current data

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       SERV.SUName,
       hist_entity.SUName as hist_name
from SERV
join hist.SERV as hist_entity using(State, RegId, OrgId, ClusId, SUIId)
where not sloppy_match(SERV.SUName, hist_entity.SUName)
```

Data Elements:

- [ClusId](#)
- [OrgId](#)
- [RegId](#)
- [SUIId](#)
- [SUName](#)
- [State](#)

1.9.123. ServSUNameMissing

Class:

Missing

Priority:

High

Message:

Missing data - SUName

Mark:

SERV.SUName

Description:

Missing data - Residential Service Unit Name (SUName)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       SUName
from SERV
where SUName is null
```

Data Elements:

- [ClusId](#)
- [OrgId](#)

- RegId
- SUIId
- SUName
- State

1.9.124. ServSectorMissing

Class:

Missing

Priority:

High

Message:

Missing data - Sector `$Sector.q`

Mark:

SERV.Sector

Description:

Missing data - Sector (Sector)

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       SUIId,
       Sector
from SERV
where Sector is null
```

Data Elements:

- ClusId
- OrgId
- RegId
- SUIId
- Sector
- State

1.9.125. SussHrDoBCount

Class:

Anomaly

Priority:

Low

Message:

More than double the average birthrate (`$DoBCount` vs `$PerDoBCountAvg`) on a suspicious date (`$DoB`)

Mark:

HR.State

Description:

There are greater than double the average number of births for a date which is likely to be a default or erroneous, selected from 9/9/9, 9/9/99, 1/1/1970, 1/1/11

SQL:

```
select State,
       DoB,
       count(*) AS DoBCount,
       round(PerDoBCountAvg,1) as PerDoBCountAvg
  from PER,
       PerDoBCount
 where DoB in ('1970-01-01', '1911-01-01', '2011-01-01', '1909-09-09', '1999-09-09')
       and DoBFlag = '1'
 group by State,
         DoB,
         PerDoBCountAvg
 having count(*) > PerDoBCountAvg * 2
```

Data Elements:

- DoB
- DoBFlag
- State

2. Submission and delivery of RMHC NMDS data files

Submission, delivery and validation of the 2026-27 RMHC data (version 7.01) will occur through the [Online Validator](#). AIHW and the Department of Health, Disability and Ageing will obtain jurisdictional DAT files directly from the Online Validator using the download functionality available to reviewers.

2.1. Timelines

Jurisdictions are requested to propose a file using the Online Validator by **3 January 2028**. The Australian Institute of Health and Welfare (AIHW) and the Department of Health, Disability and Ageing are aiming to have fully reviewed and validated the file by **14 April 2028** to facilitate timelier reporting of RMHC data, in accordance with the schedule in [Table 2.1](#).

Table 2.1 CMHC and RMHC NMDS 2026-27 data validation

Progress point description	Responsibility	Completion Time	Completion date
Compliant file proposed for review	Jurisdictions		3 January 2028
Submitter comment on issues within the issue list	Jurisdictions	7 weeks	18 February 2028
Reviewer reply to issues within the issue list and raise other issues based on historical reports	AIHW	3 weeks	10 March 2028
Resolution of any remaining issues - validation process completed and submission finalised	AIHW/Jurisdiction discussion back and forth	5 weeks	14 April 2028

2.2. File type and naming convention

DAT files should be a single Fixed Format data file, with each record in the file being terminated with Carriage Return (CR) and Line Feed (LF) characters.

The data file will have the naming convention of *RMHCSSSYYYYYNNNNN.DAT* where:

- RMHC denotes 'Residential Mental Health Care'
- SSS is the abbreviation for the State name, using the following convention:
 - New South Wales = NSW
 - Victoria = VIC
 - Queensland = QLD
 - Western Australia = WAU
 - South Australia = SAU
 - Tasmania = TAS
 - Australian Capital Territory = ACT
 - Northern Territory = NTE

- YYYY indicates the reporting year covered in the file, using the convention where financial years are abbreviated by referring to the last calendar year of the pair (for example, 2026-27 is identified as 2027)
- NNNNN represents an incremental batch number (leading zeros present).

Any resubmitted files should have a batch number greater than the file they replace. For example, the first RMHC data file submitted by the Australian Capital Territory covering the 2026-27 year would be named 'RMHCACT202700001.DAT'.

2.3. Validation

Mental Health National Minimum Dataset (NMDS) validation is the process of reviewing and cleaning the mental health service data received from state and territory governments using the [Online Validator](#) a web based validation tool. The process includes:

- Checks on file upload, before the submitter chooses to propose the file for review
- The option for reviewers to accept or reject the proposed file for review
- Collaborative review and validation
 - Review and accept all issues in the issues list in consultation with jurisdictions as necessary.
 - Review data set reports (CMHC and RMHC) and raise additional issues with the jurisdiction as necessary.
- A finalisation step, indicating that the submission process for that reporting cycle is complete.

2.3.1. Checks on file upload

The pre-submission checks ensure that the submitted data file structure is correct: that the data is in the correct layout, that there no disallowed blank fields and no invalid characters. These checks ensures that each line of data is correctly formatted and aligns to the specifications.

2.3.2. Collaborative review and validation

Collaborative review and validation is the process of reviewing unusual trends in the files. Each collection is different and has its own structure, but the general process of validating is the same. In this process:

1. Submitters work though the issues list, either providing comments on known issues or updating and resubmitting the file. It is expected that most jurisdictions will need to submit updated files multiple times before validation is finalised. Note that jurisdictions must submit comments on issues before the AIHW is able to accept.
2. Submitters and reviewers collaborate, via the Online Validator, when more information is required on an issue. Each jurisdiction will have an AIHW staff member assigned as the lead validator, who will be in touch early in the validation period. Please ensure that any email communication is also Cc'd to the Mentalhealth@aihw.gov.au inbox in case of staff absences or changes in AIHW staffing.

3. The AIHW will review the 'data set reports' and will raise additional queries about any large fluctuations evident in these reports after investigating which regions, organisations and service units are causing them. The AIHW will endeavour to ensure that additional queries are limited to essential issues that impact the state-wide result.

2.4. Submission report

States and territories must complete the submission report, formerly the 'data quality survey', in the Online Validator for the RMHC submission. Details will be used to inform the Quality statements published on METEOR for each NMDS collection period.

2.5. Additional information and queries

AIHW and Logicly staff are available to answer any queries in respect of changes to the Validator's RMHC module and validation rules. In order to obtain a coordinated response, requests should be sent to the following parties simultaneously:

AIHW

Email: mentalhealth@aihw.gov.au

Logicly

Email: support@validator.com.au