Loading RIF data

Database Requirements

Version: 0.1

Peter Hambly

SAHSU

7th December 2015

**Change Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 0.1 | 7th December 2016 | Peter Hambly | Initial version |

Contents

[Overview 3](#_Toc468882453)

[Restrictions 3](#_Toc468882454)

[Conventions 3](#_Toc468882455)

[Pre Data Loader Build 4](#_Toc468882456)

[Postgres 4](#_Toc468882457)

[SQL Server 4](#_Toc468882458)

[Loading Data 4](#_Toc468882459)

[Geospatial Data 4](#_Toc468882460)

[Health Study Themes 5](#_Toc468882461)

[Numerator data 6](#_Toc468882462)

[Numerator mandatory columns 6](#_Toc468882463)

[Numerator meta data 7](#_Toc468882464)

[Numerator RIF40\_TABLES column definitions 7](#_Toc468882465)

[Age sex group 8](#_Toc468882466)

[Ontology data 8](#_Toc468882467)

[Denominator data 10](#_Toc468882468)

[Denominator mandatory columns 11](#_Toc468882469)

[Denominator RIF40\_TABLES column definitions 11](#_Toc468882470)

[Covariate Data 11](#_Toc468882471)

[Covariate mandatory columns 12](#_Toc468882472)

[RIF40\_COVARIATES column definitions 13](#_Toc468882473)

# Overview

This document defines the database requirements for loading RIF data using the data load tool.

RIF database data is of three types:

1. Study data. These tables always contain a *study\_id* column. To allow for multi user support the tables start t\_<view> with an DML capable view to the users to view and update only the data they can access.
2. Configuration data. This data is of two types – meta data for load data and other configuration data
3. Load data. This is data imported into the RIF by the data loader. Principally it is of four types:
   1. Numerator data
   2. Denominator data
   3. Covariate data
   4. Geospatial data

The geospatial data must be setup first. Additionally, numerators are allocated to health study themes to allow numbers of numerators to be supported and sorted for ease of access.

# Restrictions

* The data loader only supports one denominator per numerator. This is to allow the user to choose a numerator and the denominator can then be automatically pre-populated. These tables have the automatic flag set to Y in RIF40\_TABLES. This limits a numerator denominator pair to one geography.
* The table T\_RIF40\_NUM\_DENOM is a system and per user table for extra numerator and denominator pairs not added automatically. Typically, these are the standard SAHSU tables with three geographies (ED91, OA2001, OA2011) that break the one denominator per numerator rule. This will not be supported by the data loader and will require further work in the front end and middleware.
* ICD version 10 is the only ontology currently supported, with a single ICD field.
* Only very limited support for dependency management is required in the first version of the data load tool, namely for:
  + Geography support. Limited to one per numerator denominator pair
  + Only support the standard 21 year bands AGE\_SEX\_GROUP. Do not support AGE or SEX fields in the production table (obviously, they would be expected to be present in the source data).
  + Only support a single ICD field.
  + Only one ontology is supported: ICD10.

These restrictions will be progressively removed in future RIF releases.

# Conventions

* All table and column names are stored in the database in upper case. The database and middleware converts the names into the default case of the database (lowercase for both Postgres and SQL server).
* Do not use spaces in table or column names, only use A to Z, 0 to 9 and \_ (underscore).
* All table and column names restricted to 30 characters.

# Pre Data Loader Build

## Postgres

Currently data is loaded by: *v4\_0\_postgres\_sahsuland\_imports.sql* and the empty data loader test database created by *v4\_0\_postgres\_sahsuland\_empty\_imports.sql*. Both scripts are in the directory *..rapidInquiryFacility\rifDatabase\Postgres\sahsuland*.

The script: *v4\_0\_postgres\_sahsuland\_imports.sql* will eventually call the data loader directly. It will then only load the standard RIF data (i.e. the contents of RIF40\_REFERENCE\_TABLES):

* Rif40\_PREDEFINED\_GROUPS
* RIF40\_AGE\_GROUP\_NAMES
* RIF40\_AGE\_GROUPS
* RIF40\_PROJECTS
* RIF40\_OUTCOMES
* RIF40\_OUTCOME\_GROUPS
* RIF40\_REFERENCE\_TABLES
* RIF40\_VERSION
* T\_RIF40\_PARAMETERS

This also includes schema metadata only used by the build process and not by the RIF itself:

* RIF40\_ERROR\_MESSAGES
* RIF40\_COLUMNS
* RIF40\_TAVBLES\_AND\_VIEWS
* RIF40\_TRIGGERS

The following are obsolete and will be removed:

* RIF40\_ICD10
* RIF40\_ICD9
* RIF40\_ICD\_O\_3
* RIF40\_OPCS4
* RIF40\_A\_AND\_E
* RIF40\_CHI2
* RIF40\_POIS\_DISTRIBUTION
* RIF40\_POPULATION\_EUROPE
* RIF40\_POPULATION\_US
* RIF40\_POPULATION\_WORLD
* T\_RIF40\_FDW\_TABLES
* RIF40\_DUAL

## SQL Server

To be added.

# Loading Data

## Geospatial Data

The following geospatial data are required by the data loader:

* Geography, in RIF40\_GEOGRAPHIES. The columns GEOGRAPHY and DESCRIPTION name and describe the geography respectively. The other columns are either related to the load process (e.g. MAX\_GEOJSON\_DIGITS) or set default behaviour in the runtime (e.g. DEFAULTSTUDYAREA) or provide the location for looking tables (e.g. TILETABLE). The SRID spatial reference identifier is used by load process and will in future be used by the runtime as it defines the map projection.
* Two or more geolevels in RIF40\_GEOLEVELS. The column GEOLEVEL\_NAME is the name of geolevel. This will be a column name in the numerator/denominator tables. GEOLEVEL\_ID is an identifier for ordering (1=lowest resolution). Up to 99 supported. DESCRIPTION is self evident. As above the other columns are either related to the load process (e.g. SHAPEFILE) or set default behaviour in the runtime (e.g. RESOLUTION) or provide the location for looking tables (e.g. LOOKUP\_TABLE). The spatial control fields are defined as follows:
  + RESTRICTED: Is geolevel access restricted by Information Governance restrictions (0/1). If 1 (Yes) then a) students cannot access this geolevel and b) if the system parameter ExtractControl=1 then the user must be granted permission by a RIF\_MANAGER to extract from the database the results, data extract and maps tables. This will be enforced by the RIF application.
  + RESOLUTION: Can use a map for selection at this resolution (0/1)
  + COMPAREA: Able to be used as a comparison area (0/1)
  + LISTING: Able to be used in a disease map listing (0/1)

The data loader is not concerned with the spatial control fields.

The following is the RIF 3.0 setup for SAHSU using the 2001 census administrative geography:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Geolevel Name | Geolevel ID | Description | Restricted | Resolution | Comparison area | Listing |
| SCNTRY2001 | 1 | 2001 super country (UK) | 0 | 1 | 1 | 1 |
| CNTRY2001 | 2 | 2001 country (e.g. England) | 0 | 1 | 1 | 1 |
| GOR2001 | 3 | 2001 Government office region (e.g. London) | 0 | 1 | 1 | 1 |
| LADUA2001 | 4 | 2001 local area district or unitary authority | 0 | 1 | 1 | 1 |
| WARD2001 | 5 | 2001 Census statistical ward | 1 | 1 | 1 | 0 |
| SOA2001 | 6 | 2001 Super output areas lower level | 1 | 0 | 1 | 0 |
| OA2001 | 7 | 2001 Census output areas | 1 | 0 | 1 | 0 |

Geospatial data must be loaded first.

How geospatial data is loaded will be covered in a later version.

## Health Study Themes

The data loader needs to either create a new theme or re-use a pre-exiting theme:

DELETE FROM rif40\_health\_study\_themes WHERE theme = 'SAHSULAND';

INSERT INTO rif40\_health\_study\_themes(theme, description) VALUES('SAHSULAND', 'SAHSU land cancer incidence example data');

Themes are intended to group numerator tables together for ease of use; e.g. “England Hospital Episode Statistics”, “England Cancer Data”. The THEME must be unique.

## Numerator data

The numerator table must be created, indexed, commented and access granted:

CREATE TABLE rif\_data.sahsuland\_cancer (

year smallint NOT NULL,

age\_sex\_group smallint NOT NULL,

level1 varchar(20) NOT NULL,

level2 varchar(20) NOT NULL,

level3 varchar(20) NOT NULL,

level4 varchar(20) NOT NULL,

icd varchar(4) NOT NULL,

total double precision NOT NULL

);

CREATE INDEX sahsuland\_cancer\_age\_sex\_group ON rif\_data.sahsuland\_cancer (age\_sex\_group);

CREATE INDEX sahsuland\_cancer\_icd ON rif\_data.sahsuland\_cancer (icd);

CREATE INDEX sahsuland\_cancer\_sex ON rif\_data.sahsuland\_cancer (trunc(age\_sex\_group/100));

CREATE UNIQUE INDEX sahsuland\_cancer\_pk ON rif\_data.sahsuland\_cancer (year,level4,age\_sex\_group,icd);

CREATE INDEX sahsuland\_cancer\_year ON rif\_data.sahsuland\_cancer (year);

CREATE INDEX sahsuland\_cancer\_age\_group ON rif\_data.sahsuland\_cancer (mod(age\_sex\_group,100));

CREATE INDEX sahsuland\_cancer\_level4 ON rif\_data.sahsuland\_cancer (level4);

CREATE INDEX sahsuland\_cancer\_level2 ON rif\_data.sahsuland\_cancer (level2);

CREATE INDEX sahsuland\_cancer\_level3 ON rif\_data.sahsuland\_cancer (level3);

CREATE INDEX sahsuland\_cancer\_level1 ON rif\_data.sahsuland\_cancer (level1);

COMMENT ON TABLE rif\_data.sahsuland\_cancer IS 'SAHSU land Population';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.age\_sex\_group IS 'Age sex group';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.icd IS 'ICD';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.level1 IS 'level1';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.level2 IS 'level2';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.level3 IS 'level3';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.level4 IS 'level4';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.total IS 'Total';

COMMENT ON COLUMN rif\_data.sahsuland\_cancer.year IS 'Year';

GRANT SELECT ON rif\_data.sahsuland\_cancer TO rif\_user, rif\_manager;

User may wish to have project and dataset specific grants. These will be supported by the information governance tool in a later release, access is currently granted to the standard roles: *rif\_user* and *rif\_manager*.

Scripts exist to partition numerator data on Postgres. These will be supported by the data loader in a later release.

### Numerator mandatory columns

These columns must be present in the numerator table. They do not have to be populated in the data.

* YEAR
* AGE\_SEX\_GROUP
* All <geolevel name> columns
* <outcome column> e.g. ICD
* <total field>, e.g. TOTAL. The RIF aggregates the data total using either COUNT(<primary key column>) or SUM(<total field>) if RIF40\_TABLES.TOTAL\_FIELD is defined.

If the primary key column is not defined then *COUNT(\*)* will be used. This is inefficient, especially on Postgres, *always define a primary key (unlike in the above example).*

The table data must be loaded:

\COPY sahsuland\_cancer FROM '../sahsuland/data/sahsuland\_cancer.csv' WITH (FORMAT csv, QUOTE '"', ESCAPE '\');

### Numerator meta data

Setup the numerator meta data:

DELETE FROM rif40\_tables WHERE table\_name = 'SAHSULAND\_CANCER';

INSERT INTO rif40\_tables(table\_name, description, theme, year\_start, year\_stop, total\_field, isindirectdenominator, isdirectdenominator, isnumerator, automatic, age\_group\_id)

VALUES (

'SAHSULAND\_CANCER','Cancer cases in SAHSU land', 'SAHSULAND', 1989,1996,'TOTAL',0,0,1,1,1);

### Numerator RIF40\_TABLES column definitions

* THEME: Health study theme (RIF40\_HEALTH\_STUDY\_THEMES.THEME)
* TABLE\_NAME: RIF table name. Normally the schema owner will not be able to see the health data tables, so no error is raised if the table cannot be resolved to an accessible object. The schema owner must have access to automatic indirect standardisation denominators. If the table is not accessible to the user, it will not appear in the RIF40\_NUM\_DENOM view. The TABLE\_NAME must be unique.
* DESCRIPTION: Description of table.
* YEAR\_START: Year table data starts.
* YEAR\_STOP: Year table data stops.
* TOTAL\_FIELD: Total field (when used aggregated tables). If set must exist in TABLE\_NAME.
* ISNUMERATOR: Is table a numerator (0/1): Set to 1.
* ISINDIRECTDENOMINATOR: table a denominator to be used in indirect standardisation (0/1). This is always the opposite of ISDIRECTDENOMINATOR. 0 if the table is a denominator as only indirect standardisation is supported at present. Set to 0.
* ISDIRECTDENOMINATOR: Is table a denominator to be used in direct standardisation (0/1). E.g. POP\_WORLD, POP\_EUROPE. Always 0 (unless the denominator is a standard population). Direct standardisation using standard population is not currently supported. Set to 0.
* AUTOMATIC: Able to be used in automatic RIF40\_NUM\_DENOM (0/1, default 0). Cannot be applied to direct standardisation denominator. Restricted to 1 denominator per geography to prevent the automatic RIF40\_NUM\_DENOM having >1 pair per numerator. This restriction is enforced in RIF40\_NUM\_DENOM because of the “ORA-04091: table RIF40.RIF40\_TABLES is mutating, trigger/function may not see it”; error. A user specific T\_RIF40\_NUM\_DENOM is supplied for other combinations. Set to 1.
* SEX\_FIELD\_NAME: Name of SEX field. No default. AGE\_GROUP\_FIELD\_NAME must be set, AGE\_SEX\_GROUP\_FIELD\_NAME must not be set. Always NULL as only an AGE\_SEX\_GROUP\_FIELD\_NAME of AGE\_SEX\_GROUP is currently supported. Set to NULL.
* AGE\_GROUP\_FIELD\_NAME: Name of AGE\_GROUP field. No default. SEX\_FIELD\_NAME must be set, AGE\_SEX\_GROUP\_FIELD\_NAME must not be set. Always NULL as only an AGE\_SEX\_GROUP\_FIELD\_NAME of AGE\_SEX\_GROUP is currently supported. Set to NULL
* AGE\_SEX\_GROUP\_FIELD\_NAME: Name of AGE\_SEX\_GROUP field. Default: AGE\_SEX\_GROUP; AGE\_GROUP\_FIELD\_NAME and SEX\_FIELD\_NAME must not be set. Always ‘AGE\_SEX\_GROUP’ as only an AGE\_SEX\_GROUP\_FIELD\_NAME of AGE\_SEX\_GROUP is currently supported. Use: AGE\_SEX\_GROUP
* AGE\_GROUP\_ID: Type of RIF age group in use. Link to RIF40\_AGE\_GROUP\_NAMES. Use 1 (RIF Default)
* VALIDATION\_DATE: Date table contents were validated OK. Leave NULL.

### Age sex group

The RIF supports the following types of age sex group defined in RIF40\_AGE\_GROUP\_NAMES:

* 1: RIF Default
* 2: RIF Birth defects
* 3: RIF Childhood infectious diseases

Always uses an AGE\_GROUP\_ID of 1. This is defined as:

|  |  |  |  |
| --- | --- | --- | --- |
| Offset | Low age | High age | Field name |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 |
| 5 | 5 | 9 | 5\_9 |
| 6 | 10 | 14 | 10\_14 |
| 7 | 15 | 19 | 15\_19 |
| 8 | 20 | 24 | 20\_24 |
| 9 | 25 | 29 | 25\_29 |
| 10 | 30 | 34 | 30\_34 |
| 11 | 35 | 39 | 35\_39 |
| 12 | 40 | 44 | 40\_44 |
| 13 | 45 | 49 | 45\_49 |
| 14 | 50 | 54 | 50\_54 |
| 15 | 55 | 59 | 55\_59 |
| 16 | 60 | 64 | 60\_64 |
| 17 | 65 | 69 | 65\_69 |
| 18 | 70 | 74 | 70\_74 |
| 19 | 75 | 79 | 75\_79 |
| 20 | 80 | 84 | 80\_84 |
| 21 | 85 | 255 | 85PLUS |

The SEX is always defined as males: 1 and females: 2. AGE\_SEX\_GROUP is defined as SEX\*100+OFFSET; e.g. female 35 to 39 is an AGE\_SEX\_GROUP of 211.

The FIELD\_NAME column is an historic artefact from the old, unnormalized population tables where the population data for a given geolevel/year was held in 42 columns; e.g. female 35 to 39 would be the column F35\_39. This is very inefficient for data extraction and was removed in the migration from RIF2.0 to RIF 3.0, RIF 3.0 could use both types of table. It is possible to create a query or view to allow the data to be queried in this form should the Epidemiologists require it.

### Ontology data

Integrate ICD data:

ICD columns must be added to the table RIF40\_TABLE\_OUTCOMES so that the data extract can create the correct outcome filter. There are two control tables: RIF40\_OUTCOMES for the differing types of ontology (e.g. ICD) and RIF40\_OUTCOME\_GROUPS for sets of one or more outcome filter column names.

#### RIF40\_OUTCOMES

The table RIF40\_OUTCOMES defines the ontologies to be supported by the RIF. Only ICD in version 10 is currently supported, with a single ICD field.

|  |  |  |
| --- | --- | --- |
| Outcome type | Outcome Description | Current version |
| A&E | A&E clinical diagnosis (3 char) | Unknown |
| BIRTHWEIGHT | Birthweight (e.g. low <2500g) | 1 |
| ICD | International Classification of Disease | 10 |
| ICD-O | International Classification of Disease for Oncology | 3 |
| OPCS | Office of Population Censuses and Surveys [OPCS] Classification of Interventions and Procedures | 4 |

It is expected that the data loader will eventually added support for ontologies, and so will manage RIF40\_OUTCOMES.

#### RIF40\_OUTCOME\_GROUPS

Users either:

1. create a new outcome group, e.g.

INSERT INTO rif40\_outcome\_groups

(

outcome\_type, /\* Outcome type: ICD, ICD-0 or OPCS \*/

outcome\_group\_name, /\* Outcome Group Name. e.g. SINGLE\_ICD \*/

outcome\_group\_description, /\* Outcome Group Description. E.g. “Single variable ICD” \*/

field\_name, /\* Outcome field name, e.g. ICD\_SAHSU\_01 \*/

multiple\_field\_count /\* Outcome Group multiple field count (0-99) \*/

)

VALUES (‘<my type>’, ‘<my outcome group name>’, ‘<my field description>’, ‘<my field>’, 0);

* The OUTCOME\_GROUP\_NAME must be unique.
* The outcome Group multiple field count (0-99) must be 0. If 0, <my field> is the actual database field. If >0 the actual database field is <my field>\_<n> where <n> is between 12 and MULTIPLE\_FIELD\_COUNT.

Or:

1. use a pre-existing one, e.g. '*SINGLE\_ICD'.*

The pre-existing outcome groups defined in RIF40\_OUTCOME\_GROUPS are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outcome type | Outcome group name | Outcome description | Field name | Multiple field count |
| BIRTHWEIGHT | BIRTHWEIGHT | Birthweight (e.g. low <2500g) | BIRTHWEIGHT | 0 |
| ICD-O | CANCER\_ICD\_O | Cancer type of growth histology coded to Classification of Neoplasms | TYPE\_GROWTH | 0 |
| A&E | HES\_MULTIPLE\_A&E | HES multiple A+E | DIAG | 20 |
| ICD | HES\_MULTIPLE\_ICD | HES multiple ICD | ICD\_SAHSU | 20 |
| OPCS | HES\_MULTIPLE\_OPCS | HES multiple OPCS | OPCS\_SAHSU | 20 |
| A&E | HES\_SINGLE\_A&E | HES single A+E | DIAG | 0 |
| OPCS | HES\_SINGLE\_OPCS | HES single OPCS | OPCS\_SAHSU | 0 |
| ICD | MORTALITY\_MULTIPLE\_ICD | ONS Mortality multiple ICD | ICD\_SAHSU | 16 |
| ICD | MORTALITY\_SECONDARY\_ICD | Single ICD - Secondary cause of death | ICD\_SAHSU\_01S | 0 |
| ICD | SAHSULAND\_ICD | Single ICD | ICD | 0 |
| ICD | SINGLE\_ICD | Single ICD | ICD\_SAHSU\_01 | 0 |

The FIELD\_NAME defined in RIF40\_OUTCOME\_GROUPS must match the numerator table ICD column.

#### RIF40\_TABLE\_OUTCOMES

In this case the expected column name is ICD and *CURRENT\_VERSION\_START\_YEAR* in RIF40\_TABLE\_OUTCOMES is the first year of ICD10 data.

DELETE FROM rif40\_table\_outcomes WHERE numer\_tab = 'SAHSULAND\_CANCER';

INSERT INTO rif40\_table\_outcomes(outcome\_group\_name, numer\_tab, current\_version\_start\_year)

VALUES (‘SAHSULAND\_ICD', 'SAHSULAND\_CANCER', 1993);

The OUTCOME\_GROUP\_NAME and NUMER\_TAB combination must be unique.

It is appreciated that that Java data loader uses a different definition for multiple ICD columns. The RIF40\_OUTCOME\_GROUPS table does not define the actual field columns, so one possible solution would be to rename RIF40\_OUTCOME\_GROUPS to T\_ RIF40\_OUTCOME\_GROUPS and create a RIF40\_OUTCOME\_GROUPS view with the real column names. For example, the OUTCOME\_GROUP\_NAME of “MORTALITY\_MULTIPLE\_ICD” has a MULTIPLE\_FIELD\_COUNT of 16 so the columns would be names ICD\_SAHSU\_01 to ICD\_SAHSU\_16. The view could only be queried.

The table RIF40\_TABLE\_OUTCOMES is used to create the ontology filter as part of the data extraction.

## Denominator data

The denominator table must be created, indexed, commented and access granted:

CREATE TABLE pop.sahsuland\_pop (

year smallint NOT NULL,

age\_sex\_group smallint NOT NULL,

level1 varchar(20) NOT NULL,

level2 varchar(20) NOT NULL,

level3 varchar(20) NOT NULL,

level4 varchar(20) NOT NULL,

total double precision NOT NULL

);

CREATE INDEX sahsuland\_pop\_level4 ON pop.sahsuland\_pop (level4);

CREATE INDEX sahsuland\_pop\_age\_sex\_group ON pop.sahsuland\_pop (age\_sex\_group);

CREATE INDEX sahsuland\_pop\_level1 ON pop.sahsuland\_pop (level1);

CREATE INDEX sahsuland\_pop\_age\_group ON pop.sahsuland\_pop (mod(age\_sex\_group,100));

CREATE UNIQUE INDEX sahsuland\_pop\_pk ON pop.sahsuland\_pop (year,level4,age\_sex\_group);

CREATE INDEX sahsuland\_pop\_year ON pop.sahsuland\_pop (year);

CREATE INDEX sahsuland\_pop\_level2 ON pop.sahsuland\_pop (level2);

CREATE INDEX sahsuland\_pop\_sex ON pop.sahsuland\_pop (trunc(age\_sex\_group/100));

CREATE INDEX sahsuland\_pop\_level3 ON pop.sahsuland\_pop (level3);

COMMENT ON TABLE pop.sahsuland\_pop IS 'SAHSU land Population';

COMMENT ON COLUMN pop.sahsuland\_pop.age\_sex\_group IS 'Age sex group';

COMMENT ON COLUMN pop.sahsuland\_pop.level1 IS 'level1';

COMMENT ON COLUMN pop.sahsuland\_pop.level2 IS 'level2';

COMMENT ON COLUMN pop.sahsuland\_pop.level3 IS 'level3';

COMMENT ON COLUMN pop.sahsuland\_pop.level4 IS 'level4';

COMMENT ON COLUMN pop.sahsuland\_pop.total IS 'Total';

COMMENT ON COLUMN pop.sahsuland\_pop.year IS 'Year';

GRANT SELECT ON pop.sahsuland\_pop TO rif\_user, rif\_manager;

Scripts exist to partition denominator data on Postgres. These will be supports by the data loader in a later release.

### Denominator mandatory columns

These columns must be present. They do have to be populated in the data, i.e. all years, AGE\_SEX\_GROUP and geolevel column present. If this is not so, cases may be excluded from the study even if the population is zero. The years present should equal or exceed the numerator data.

* YEAR
* AGE\_SEX\_GROUP
* All <geolevel name> columns
* TOTAL. The RIF denominator using SUM(total). RIF40\_TABLES.TOTAL\_FIELD should be defined as “TOTAL”.

The table data must be loaded:

\COPY sahsuland\_pop FROM '../sahsuland/data/sahsuland\_pop.csv' WITH (FORMAT csv, QUOTE '"', ESCAPE '\');

Setup the denominator meta data:

DELETE FROM rif40\_tables WHERE table\_name = 'SAHSULAND\_POP';

INSERT INTO rif40\_tables(table\_name, description, theme, year\_start, year\_stop, total\_field, isindirectdenominator, isdirectdenominator, isnumerator, automatic, age\_group\_id)

VALUES (

'SAHSULAND\_POP','SAHSU land population', 'SAHSULAND', 1989,1996,NULL,1,0,0,1,1);

### Denominator RIF40\_TABLES column definitions

For other RIF40\_TABLES column definitions see numerator data:

* ISNUMERATOR: Is table a numerator (0/1): Set to 0.
* ISINDIRECTDENOMINATOR: table a denominator to be used in indirect standardisation (0/1). This is always the opposite of ISDIRECTDENOMINATOR. 0 if the table is a denominator as only indirect standardisation is supported at present. Set to 1.
* ISDIRECTDENOMINATOR: Is table a denominator to be used in direct standardisation (0/1). E.g. POP\_WORLD, POP\_EUROPE. Always 0 (unless the denominator is a standard population). Direct standardisation using standard population is not currently supported. Set to 0.

## Covariate Data

Covariate data is required if *COVARIATE\_TABLE\_NAME* is defined for a geolevel in the table RIF40\_COVARIATES. Covariates are normally only defined for the higher resolution geoelvels and must not be defined for the first level (low resolution, e.g. country).

The covariate table(s) must be created, indexed, commented and access granted:

CREATE TABLE rif\_data.sahsuland\_covariates\_level3 (

year smallint NOT NULL,

level3 varchar(20) NOT NULL,

ses smallint NOT NULL,

ethnicity smallint NOT NULL

);

CREATE UNIQUE INDEX sahsuland\_covariates\_level3\_pk ON rif\_data.sahsuland\_covariates\_level3 (year,level3);

CREATE TABLE rif\_data.sahsuland\_covariates\_level4 (

year smallint NOT NULL,

level4 varchar(20) NOT NULL,

ses smallint NOT NULL,

areatri1km smallint NOT NULL,

near\_dist double precision NOT NULL,

tri\_1km smallint NOT NULL

);

CREATE UNIQUE INDEX sahsuland\_covariates\_level4\_pk ON rif\_data.sahsuland\_covariates\_level4 (year,level4);

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level3.ethnicity IS 'Ethnicity % non white - 1: <5%, 2: 5 to 10%, 3: >= 10%';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level3.level3 IS 'Level3';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level3.ses IS 'Social Economic Status (quintiles)';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level3.year IS 'Year';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level4.areatri1km IS 'Toxic Release Inventory within 1km of area (0=no/1=yes)';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level4.level4 IS 'Level4';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level4.near\_dist IS 'Distance (m) from area centroid to nearest TRI site';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level4.ses IS 'Social Economic Status (quintiles)';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level4.tri\_1km IS 'Toxic Release Inventory within 1km of areai centroid (0=no/1=yes)';

COMMENT ON COLUMN rif\_data.sahsuland\_covariates\_level4.year IS 'Year';

GRANT SELECT ON rif\_data.sahsuland\_covariates\_level3 TO PUBLIC;

GRANT SELECT ON rif\_data.sahsuland\_covariates\_level4 TO PUBLIC;

### Covariate mandatory columns

These columns must be present. They do have to be populated in the data, i.e. all years and geolevel columns present and the values must be within the specified ranges. If this is not so, cases may be excluded from the study if the covariate is NULL, zero out outside of the defined maximum and minimum values.

* YEAR
* <geolevel name>
* <covariate fields>, e.g. SES

Covariate never have AGE\_SEX\_GROUP, AGE or SEX columns.

Annualised data (e.g. incomes) and non-annualised data is supported. For non-annualised data an entry must be present for all years covered by the denominator data.

The table data must be loaded:

\COPY sahsuland\_covariates\_level3 FROM '../sahsuland/data/sahsuland\_covariates\_level3.csv' WITH (FORMAT csv, QUOTE '"', ESCAPE '\');

\COPY sahsuland\_covariates\_level4 FROM '../sahsuland/data/sahsuland\_covariates\_level4.csv' WITH (FORMAT csv, QUOTE '"', ESCAPE '\');

Covariate data is already partly setup in rif40\_geolevels. To complete the setup the individual covariate columns must be added to RIF40\_COVARIATES

--

-- RIF40\_COVARIATES

--

DELETE FROM rif40\_covariates WHERE geography = 'SAHSU';

INSERT INTO rif40\_covariates(geography, geolevel\_name, covariate\_name, min, max, type)

SELECT 'SAHSU', 'LEVEL3', 'SES', MIN(ses), MAX(ses), 1 FROM sahsuland\_covariates\_level3;

INSERT INTO rif40\_covariates(geography, geolevel\_name, covariate\_name, min, max, type)

SELECT 'SAHSU', 'LEVEL3', 'ETHNICITY', MIN(ethnicity), MAX(ethnicity), 1 FROM sahsuland\_covariates\_level3;

INSERT INTO rif40\_covariates(geography, geolevel\_name, covariate\_name, min, max, type)

SELECT 'SAHSU', 'LEVEL4', 'SES', MIN(ses), MAX(ses), 1 FROM sahsuland\_covariates\_level4;

INSERT INTO rif40\_covariates(geography, geolevel\_name, covariate\_name, min, max, type)

SELECT 'SAHSU', 'LEVEL4', 'AREATRI1KM', MIN(areatri1km), MAX(areatri1km), 1 FROM sahsuland\_covariates\_level4;

INSERT INTO rif40\_covariates(geography, geolevel\_name, covariate\_name, min, max, type)

SELECT 'SAHSU', 'LEVEL4', 'TRI\_1KM', MIN(tri\_1km), MAX(tri\_1km), 1 FROM sahsuland\_covariates\_level4;

INSERT INTO rif40\_covariates(geography, geolevel\_name, covariate\_name, min, max, type)

SELECT 'SAHSU', 'LEVEL4', 'NEAR\_DIST', MIN(near\_dist), MAX(near\_dist), 2 FROM sahsuland\_covariates\_level4;

### RIF40\_COVARIATES column definitions

* GEOGRAPHY: Geography (e.g. EW2001)
* GEOLEVEL\_NAME: Name of geolevel. This will be a column name in the numerator/denominator tables
* COVARIATE\_NAME: Covariate name. This will be a column name in RIF40\_GEOLEVELS.COVARIATE\_TABLE
* MIN: Minimum value
* MAX: Maximum value
* TYPE: TYPE of covariate (1=integer score/2=continuous numeric variable). Min must be less than max. Max and min precision is appropriate to type. Continuous variables are not currently supported. Integer scores can be a binary variable 0/1 or an NTILE e.g. 1 to 5 for a quintile.