Population Explorer Datamart - user perspective

Version number	Date	Who	What
0.1	29 November 2017	Peter Ellis	Initial version
0.2	11 December 2017	Peter Ellis	First substantively complete version
0.3	20 December 2017	Michael and Dominik	Clean up + better links
0.4	9 May 2018	Aaron Lefebre	Further clean up before publishing on IDI Wiki

Purpose and background

This document outlines the Population Explorer Datamart from the perspective of a user seeking to understand what tables have been built, why, and how they are to be used.

Separate documentation explains the build process and the software behind the front end prototype. This document focuses on the tables in the Datamart.

The Population Explorer is being developed by Stats NZ as part of the Integrated Data Infrastructure 2 project. IDI 2 includes four workstreams:

- Access pathways
- Information layers and confidentialisation solution
- Fundamental redesign
- Scalable infrastructure in the cloud

The Population Explorer is the main deliverable under the "Information layers and confidentialisation solution" part of the work programme. Development work began in September 2017 and a releasable product is scheduled to be available (although not necessarily deployed) by December 2017. A target date of late April has been set to deploy the first iteration of the Datamart to all IDI users in the secure Data Lab environment.

Overall approach

The "rollup" idea

Microdata is data about individual people, households or organisations. Analysis that uses microdata needs to include steps (such as counting, averaging, or other aggregation) that remove any chance of attributing sensitive values to known individuals. The Population Explorer is being built on the assumption that the microdata in the IDI, and the careful controls

on access to it via the Data Lab, remain essentially the same. The Population Explorer is about improving existing usability of the data.

Most of the data in the IDI is in the form of *events* (such as "person X purchased pharmaceutical Y on 17 June 2012") and *spells* (such as "person A attended the year at school B from 12 February to 17 November 2012). A significant part of researcher time in any analysis involving the IDI is spent "rolling up" such data into regular observations (eg quarterly or annual), such as "number and value of pharmaceutical purchases per year". In order for researchers to do this, they may often have to work to become familiar with variables and concepts that they are only indirectly interested in (for example, to control for ethnicity in a statistical model).

The fundamental idea of the Population Explorer is to perform this "roll up" for around 20 to 50 variables, at annual and (if possible) quarterly[1] intervals, so researchers who are already in the Data Lab can save many days of work. This version of the data, which we describe as the Population Explorer "Datamart", will be available as a number of datasets in the Data Lab similar to existing IDI datasets. Access is available to all IDI users. The Datamart can be found under the database 'IDI_pop_explorer' on the usual SQL server connection.

Data model

The Datamart is being built with a "dimensionally modelled" data model along the lines proposed by Ralph Kimball and now standard in the presentation layer of data warehouses around the world. The data model is illustrated on the next page

The data model has been chosen to be stable regardless of how many variables are "rolled up" into it. When a new variable is added, it becomes a new row in the *dim_explorer_variable* table, several new rows in the *dim_explorer_value_year* and the *dim_explorer_value_qtr* tables, and thousands or millions of new rows in the *fact_rollup_year* and *fact_rollup_qtr* tables.

Some general conventions

- Each "fact" is a unique combination of person, rollup period (eg year) and variable. The value for that person at that time on that variable is represented in both a categorical fashion (the *fact_rollup_year.fk_value_code* column) and, for most facts, a numerical value (*fact_rollup_year.value*). Some variables (eg "region most lived in") do not have numeric values.
- all table names and schema names are in lower case and do not contain macrons, spaces or other illegal characters. However, the word Māori always includes a macron when it is a value in a table eg in *dim_explorer_variable.short_name* where it is indicating a variable included in the data mart
- table names beginning with *dim_* are dimension tables
- table names beginning with *fact_* are core fact tables
- table names beginning with *vw_* are alternative, redundant, partial and analytically convenient versions of the fact tables, which were originally developed as views (hence the naming convention) but have been materialized as tables for performance so they can have columnstore indexes on them
- columns that might need to contain macrons are of data type NVARCHAR or NCHAR so they can contain values such as 'Māori'
- all tables have columnstore indexes and at least one clustered index on them

[1] Please note, the quarterly tables will not be available in the first version of the Datamart. They will be a priority build and made available mid-2018

- attributes in the dimension tables are in plain English (not fully normalised codes linked to "snowflaking" lookup tables), to make queries readable and facilitate the use of the data model in a query-building front end
- variable names beginning with *fk_* are columns joined to a dimension table by a foreign key.

Simplified view of core layer

We recommend using SQL Server Management Studio to investigate the tables in the Population Explorer, even if you intend to mostly query them from R, Stata or SAS. Management Studio is also the best environment for developing SQL queries which you then can use from another application.

Overview

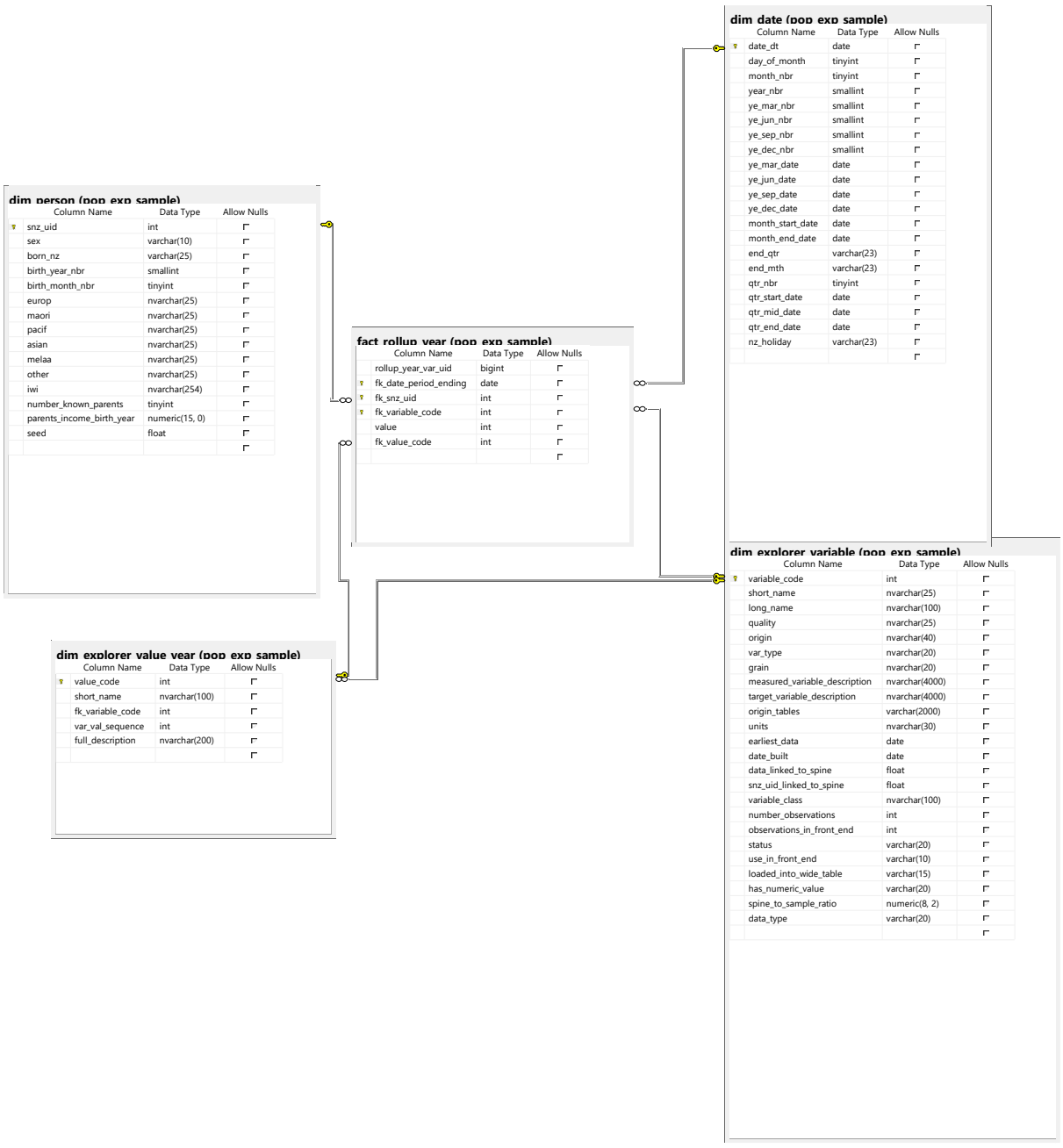


Table by table description

dim_date

This is a standard part of a dimensionally modelled data warehouse that makes it easy by doing joins to relate days to important information about that day. In the case of the Population Explorer, it is used much more during the build than it is likely to be used by researchers.

The attributes of the *dim_date* table such as 'ye_mar_nbr', 'ye_mar_date', 'month_end_date' should be self-explanatory; if not, a quick look at this should make it clear:

```
SELECT TOP 100 * FROM IDI_Pop_explorer.pop_exp.dim_date
```

date_dt	day_of_month	month_nbr	year_nbr	ye_mar_nbr	ye_jun_nbr	ye_sep_nbr	ye_dec_nbr	ye_mar_date	ye_jun_date	ye_sep_date	ye_dec_date	month_start_date
1900-07-01	1	7	1900	1901	1901	1900	1900	1901-03-31	1901-06-30	1900-09-30	1900-12-31	1900-07-01
1901-07-01	1	7	1901	1902	1902	1901	1901	1902-03-31	1902-06-30	1901-09-30	1901-12-31	1901-07-01
1902-07-01	1	7	1902	1903	1903	1902	1902	1903-03-31	1903-06-30	1902-09-30	1902-12-31	1902-07-01
1903-07-01	1	7	1903	1904	1904	1903	1903	1904-03-31	1904-06-30	1903-09-30	1903-12-31	1903-07-01
1904-07-01	1	7	1904	1905	1905	1904	1904	1905-03-31	1905-06-30	1904-09-30	1904-12-31	1904-07-01
1905-07-01	1	7	1905	1906	1906	1905	1905	1906-03-31	1906-06-30	1905-09-30	1905-12-31	1905-07-01
1906-07-01	1	7	1906	1907	1907	1906	1906	1907-03-31	1907-06-30	1906-09-30	1906-12-31	1906-07-01
1907-07-01	1	7	1907	1908	1908	1907	1907	1908-03-31	1908-06-30	1907-09-30	1907-12-31	1907-07-01
1908-07-01	1	7	1908	1909	1909	1908	1908	1909-03-31	1909-06-30	1908-09-30	1908-12-31	1908-07-01
1909-07-01	1	7	1909	1910	1910	1909	1909	1910-03-31	1910-06-30	1909-09-30	1909-12-31	1909-07-01

(some columns not shown)

dim_person

The *dim_person* table holds information on the enduring aspects of people – such as sex, ethnicity, parents' income in the year of their birth, etc.

The attributes are presented in text, not codes, in order to make querying simpler and resemble English as much as possible. For example, the query below:

```
SELECT
       COUNT(1) AS freq,
       sex
FROM IDI_Pop_explorer.pop_exp.dim_person
GROUP BY sex
```

returns a meaningful cross tab immediately, with numbers for 'Male', 'Female' and 'No data'. Notice that NULL is not used as a value in textual attributes (although it is for numeric attributes such as parents_income_at_birth when the value is not known) but they are explicitly written as 'No data'.

The version of the Datamart in the IDI_Pop_explorer database on WPRDSQL36, *dim_person* is a complete collection of everyone on the IDI spine. In other versions used during development it can be a simple random sample from the IDI spine, with a "spine_to_sample_ratio" greater than 1. This query from the variable dimension table (described later) will let you know if this is the case:

```
SELECT spine_to_sample_ratio
FROM IDI_Pop_explorer.pop_exp.dim_explorer_variable
WHERE short_name = 'Generic'
```

fact_rollup_year

The *fact_rollup_year* table contains the bulk of the data. It has one row for each combination of person, year, and variable. Each row has two columns for actual facts; one a numeric value and one a categorical. Most variables are meaningfully measured against both eg income can be "$45,726" as well as "$40,001 to $50,000". Variables that are only meaningful in a categorical sense (eg "region most lived in this year") have zero in the value column

Column	Explanation
rollup_year_var_uid	Unique identifier integer
fk_date_period_ending	The date (data type DATE) of the day the yearly period finishes. Should always be 31 December of some year. Joins to dim_data.date_dt.
fk_snz_uid	The snz_uid of a person on the IDI spine. Joins to dim_person.snz_uid.
fk_variable_code	The variable code (data type INT). Joins to dim_explorer_variable.variable_code.
value	The numeric value of this particular fact eg an actual dollar income such as "$45,726".
fk_value_code	The value code (an integer) for the categorical version of this particular fact eg "$40,001 to $50,000". Joins to dim_explorer_value_year.value_code.

dim_explorer_value_year

The value dimension table exists to give meaningful English names for the coded categories for each variable. Note that all variables have their categories and codes in this one table; this is a key design feature to avoid proliferation of lookup tables in the database.

Column	Explanation
value_code	Unique identifier (data type INT) of the particular classification. Is linked to from fact_rollup_year.fk_value_code.
short_name	Short name of the particular value of the classification eg "$40,001 to $50,000".
fk_variable_code	Which variable is this classification code used for? Joins to dim_explorer_variable.variable_code.
var_val_sequence	If the categorical values for this variable are ordinal, what is the ranking of this particular value? Used in queries in the front end prototype to preserve meaningful ordering for categories that are not meaningfully ordered alphabetically (which is all of them).
description	Verbose description, if needed for the particular value of this classification. Not currently used.

This query illustrates the relationship of the value and variable dimension tables (see *dim_explorer_variable*, next section), by showing the value codes and categories for a particular variable (in this case, Income).

```
SELECT *
FROM IDI_Pop_explorer.pop_exp.dim_explorer_value_year AS val
INNER JOIN IDI_Pop_explorer.pop_exp.dim_explorer_variable AS vr
ON val.fk_variable_code = vr.variable_code
WHERE vr.short_name = 'Income'
```

value_code	short_name	fk_variable_code	var_val_sequence	full_description	variable_code	short_name	long_name	quality	origin	var_type	grain
3	loss	2	1	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
4	$0 - $5,000	2	2	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
5	$5,001 - $10,000	2	3	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
6	$10,001 - $20,000	2	4	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
7	$20,001 - $30,000	2	5	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
8	$30,001 - $40,000	2	6	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
9	$40,001 - $50,000	2	7	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
10	$50,001 - $70,000	2	8	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
11	$70,001 - $100,000	2	9	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
12	$100,001 - $150,000	2	10	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio
13	$150,001+	2	11	NULL	2	Income	Income all sources	Good	IRD	continuous	person-perio

dim_explorer_variable

The variable dimension table contains attributes for each of the variables in the Datamart. Some of these are aimed at helping researchers understand each variable; some are used primarily during the build process but still have information of possible use:

Column	**Explanation**
variable_code	Unique identifier integer
short_name	Unique short name, for the variable, suitable for being a column name if necessary (eg no spaces or other illegal characters). short_name is NVARCHAR data type so it supports macrons eg 'Māori' is a short_name in the database. Macrons are removed before using short_name as a column later.
long_name	Long name of the variable, suitable for use in drop down boxes etc.
quality	'Poor', 'medium' or 'good' - arbitrary judgements made by the developer, indicating a very rough assessment of the quality of the data, linkage rates, and business rules that have been made in constructing the variable.
origin	Brief English description of the origin of the data eg DIA, MoH, etc
var_type	English description of the type of variable - continuous, categorical, ordinal, etc
grain	Either 'person' (for enduring variables used in dim_person) or 'person-period' (ie changing over time)
measured_variable_description	Detailed English description of how the variable was created.

target_variable_description	What the variable is really trying to measure, or being used as a proxy for.
origin_tables	Comma-separated string of all the tables that this variable depended on in the build process. This list of tables is used in the build process to construct the data_linked_to_spine and snz_uid_linked_to_spine columns, and also is intended as a useful guide for researchers seeking to understand the origin of a variable. The actual code used to create each variable is published on GitHub. (Please note this needs to be updated to reflect the latest changes to the Datamart).
units	The units that the numeric version of each variable is in eg "dollars", "number of discharges", "number of claims"
earliest_data	The earliest data that has been included in the fact_rollup_year table. Note that not all data in fact_rollup_year transfers to vw_year_wide. vw_year_wide only goes back to 1990.
date_built	The date this variable was added to this particular build of the Datamart. Usually this is the same for all variables in any particular iteration of the Datamart, as it is built in one end-to-end process. So date_built really indicates the date the Datamart was last refreshed.
data_linked_to_spine	The *minimum* linkage rate of rows in the origin_tables used for this variable. Linkage rates for each table in the IDI were estimated by the number of rows in the table that were linked to an snz_uid that is on the IDI spine, divided by the number of rows in the table that were linked to an snz_uid that is *not* on the spine.
snz_uid_linked_to_spine	The *minimum* linkage rate of individual people in the origin_tables used for this variable. Linkage rates for each table in the IDI were estimated by the number of snz_uid values in the table that were linked to the IDI spine, divided by the number of values of snz_uid that are *not* on the spine. This number will vary from data_linked_to_spine to the degree that the people linked to the spine have a different average number of values in this table than people who are not linked to the spine.
variable_class	A rough classification of variables into categories such as "Income and employment", "Health and wellbeing", "Family and childhood". Used in the front end prototype to make it easier for a user to navigate through a drop-down list of variables.
number_observations	The number of observations (ie person-year combinations) in fact_rollup_year for this variable.
observations_in_front_end	The number of observations in vw_year_wide for this variable. This is less than number_observations because vw_year_wide is only for 1990 onwards and for people estimated to have spent one day or more in New Zealand.
status	Whether or not the inclusion and definition of this variable has been approved by the Integrated Data Advisory Group. The intention is to use this column to indicate which variables are endorsed as the standard, default way of using that variable. Currently all values are

	"Not approved" (and one "Not applicable", for the "Generic" variable).
use_in_front_end	Whether or not this variable is included in the pivoted, wide version of the data ie the vw_year_wide table. This column is used during the build process, but also provides useful information to researchers.
loaded_into_wide_table	Whether or not this variable was successfully loaded into vw_year_wide.
has_numeric_value	Whether or not this variable has a numeric version of its facts in addition to a categorical version. For example, income "Has numeric value" whereas as region "No numeric value". This column is used during the build process. Variables with "No numeric value" appear as columns in vw_year_wide only in the form region_code, whereas variables with "Has numeric value" appear as two columns eg income and income_code.
spine_to_sample_ratio	Only populated for the "Generic" variable, this single number indicates the proportion of the spine that was included in this build of the database. 1 means the entire spine was included; 20 means a simple random sample of 1 twentieth of the spine was included. Some versions of the Datamart in development and test include less than the full spine for faster performance during development and test, but the version deployed for the datalab on WPRDSQL36 has the full spine.
data_type	Used during the build process; which data type to use for the numeric column for this variable in vw_year_wide. Currently nearly all columns are INT, in order to save space; the developers judged that the extra precision below a dollar in the case of monetary variables was not worth the cost in disk space.

Example queries combining the "core" tables

SQL - Total and mean income by year:

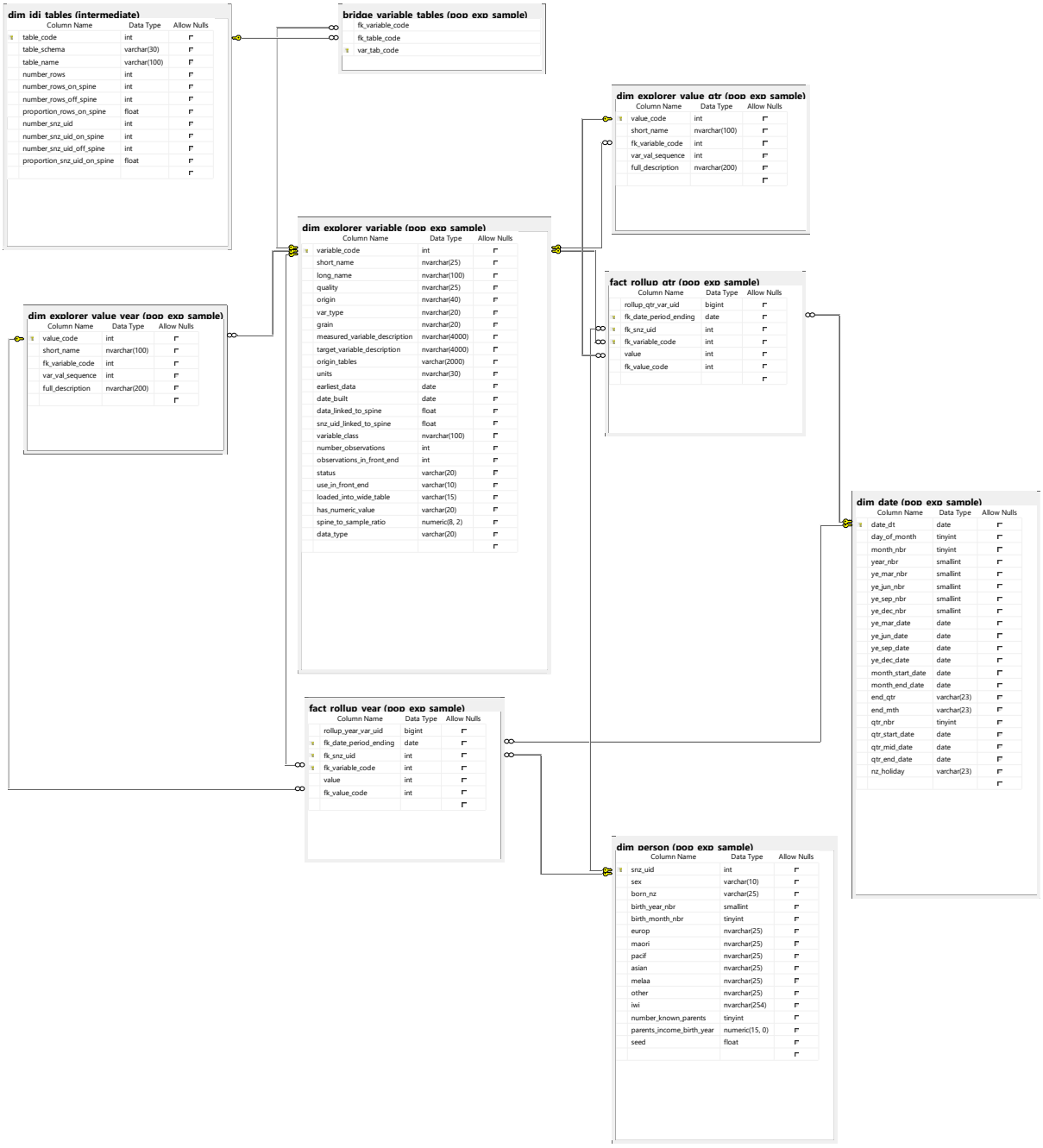
```
USE IDI_Pop_explorer

SELECT
      COUNT(1)     AS people_with_income,
      SUM(value)   AS income,
      AVG(value)   AS mean_income_of_those_with_any,
      fk_date_period_ending
FROM pop_exp.fact_rollup_year                    AS f
INNER JOIN pop_exp.dim_explorer_variable    AS vr
      ON f.fk_variable_code = vr.variable_code
WHERE vr.short_name = 'Income'
GROUP BY fk_date_period_ending
order by fk_date_period_ending
```

SQL - Number of people in each benefits category by Māori ethnicity:

```
SELECT
      COUNT(1)          AS people_with_income,
      val.short_name    AS benefits_category,
      val.var_val_sequence,
      AVG(value)        AS mean_benefits_in_this_category,
      maori,
      fk_date_period_ending
FROM pop_exp.fact_rollup_year                    AS f
INNER JOIN pop_exp.dim_explorer_variable         AS vr
      ON f.fk_variable_code = vr.variable_code
INNER JOIN pop_exp.dim_explorer_value_year  AS val
      ON f.fk_value_code = val.value_code
INNER JOIN pop_exp.dim_person                    AS p
      ON f.fk_snz_uid = p.snz_uid
WHERE vr.short_name = 'Benefits'
GROUP BY fk_date_period_ending, val.short_name, maori, var_val_sequence
ORDER BY fk_date_period_ending, maori, var_val_sequence
```

Full view of core layer

Overview



Additional tables

Some tables in what could be regarded as the core data model of the Population Explorer were not described in the previous, simplified section. Those tables are listed here.

fact_rollup_qtr and dim_explorer_value_qtr

The quarterly version of the database is a work in progress. The *fact_rollup_qtr* has the same shape as *fact_rollup_year* and, as the name implies, it contains quarterly data. Not all variables included in fact_rollup_year yet have quarterly versions. *dim_explorer_value_qtr* is the quarterly equivalent of *dim_explorer_value_year*.

fact_rollup_qtr and *fact_rollup_yr* use the same variable codes (*dim_explorer_variable.variable_code*), but they have different value codes. This is because value classifications useful for annual data are unlikely to work with quarterly data (eg the bands for `Income` in *dim_explorer_value_qtr* have cut-offs at 1/4 the value of those in *dim_explorer_value_year*.

Pop_exp.dim_idi_tables

This table has a row for each table in IDI_Clean that includes a column *snz_uid*. This is not all tables in IDI_Clean, only those that link directly to individuals.

The column names of *dim_idi_tables* should be self-explanatory:

- table_code
- table_schema
- table_name
- number_rows
- number_rows_on_spine
- number_rows_off_spine
- proportion_rows_on_spine
- number_snz_uid
- number_snz_uid_on_spine
- number_snz_uid_off_spine
- proportion_snz_uid_on_spine

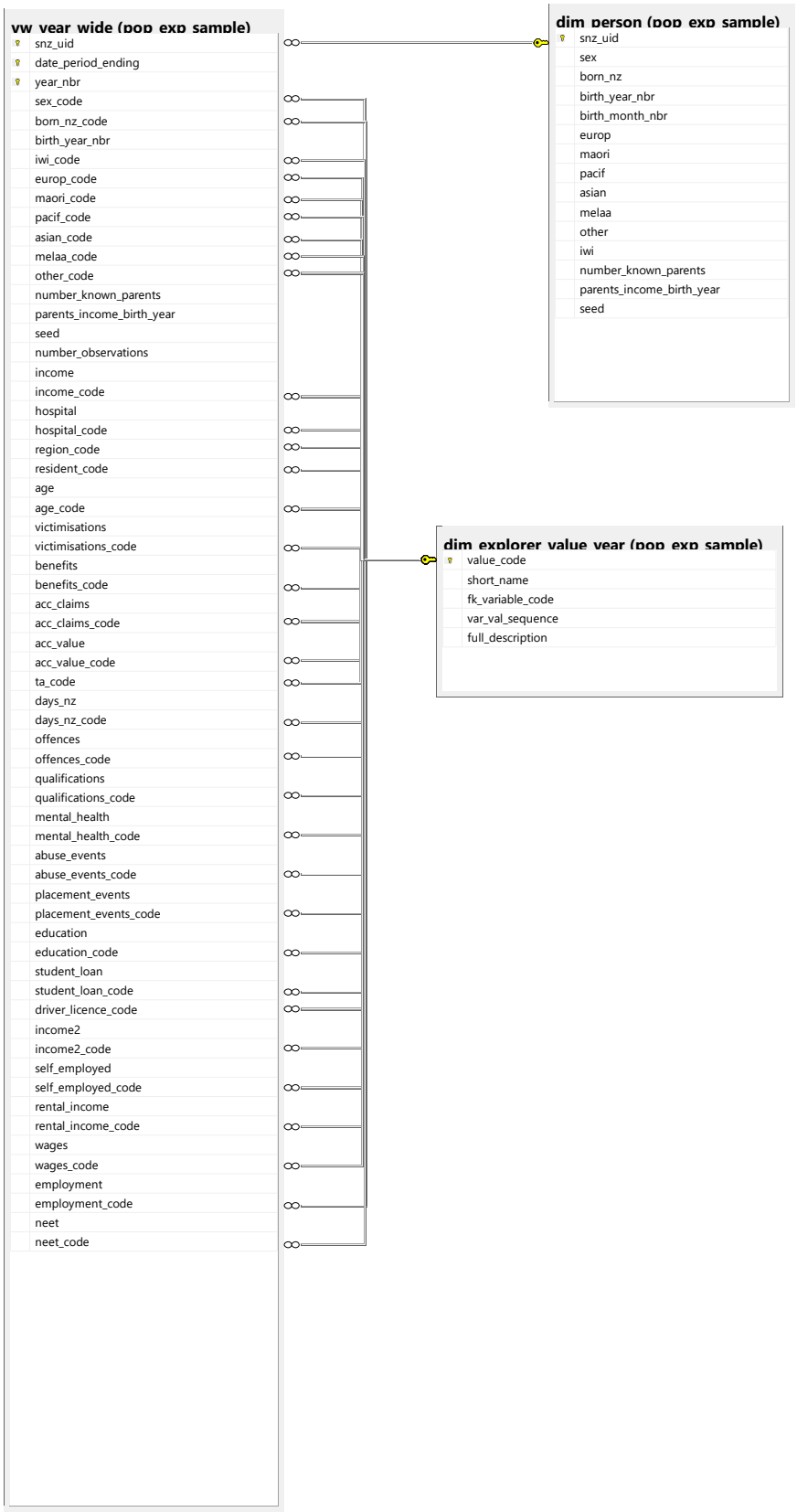
The "on spine" concept is as described above for the *dim_explorer_variable* table.

The intermediate.dim_idi_tables table is not a core part of the Population Explorer, but might be of interest to researchers. Some of the key information from it has been incorporated into *dim_explorer_variable*.

"Wide table"

Overview

In addition to the core layer, an additional wide table is provide for analysts' convenience



Use

This table does not have all the data in the more efficiently shaped fact_rollup_year but is easier to write queries against. For example, the 20+ lines of SQL used earlier in the query to create the scatter plot of income and student loans could be replaced with this six line version:

```
SELECT TOP 10000
       income,
       student_loan
FROM IDI_Pop_Explorer.pop_exp.vw_year_wide
WHERE year_nbr = 2012
ORDER BY NEWID()
```

(actually, the result is not exactly the same, mostly because the above query includes people with NULL ie no data against both items - this could be changed with adding `AND income IS NOT NULL AND student_loan IS NOT NULL` to the `WHERE` clause).

The SQL generated by the Population Explorer front end prototype uses the vw_year_wide table as its main source. The SQL it generates can be used directly in Management Studio, or integrated into R, Stata or SAS programs.

Known issues

- Individuals that should have qualification code "3.5" as their "value" for qualification have been truncated to "3". However, they have the correct categorical value_code, which equates to "Other tertiary qualification", in between NCEA3 and a degree. Most users would use the categories rather than a pseudo-continuous number for qualification; there are only a relatively small number of these individuals; and the cost of accommodating 3.5 would be to convert the data type of fact_rollup_year.value from INT to a type that takes up more space and has poorer performance.