"Totalnum" Patient Counting Scripts



These scripts will count the total number of patients for every item in your ontology, and fill these in the c_totalnum column of your ontology tables. This can be used for internal quality checks and will display in the i2b2 query tool (not SHRINE), to help researchers pick ontology items that have data behind them. The obfuscated reports of sites' totalnums can also be collected, aggregated, and analyzed by quality checking / exploring tools (forthcoming). Current technical documentation will be available in the latest release notes as new versions are released.

Installing the Scripts

- Load the stored procedures
  - Run the procedures manually, or in Release_1-7/NewInstall/Metadata/ run the ant script to create the stored procedures.
    ant -f data_build.xml create_metadata_procedures_release_1-7
- Add the totalnum and totalnum_report tables to i2b2

i2b2 Configurations Supported

1. All data mapping must be in the ontology tables, not (exclusively) the concept_dimension or Adapter Mappings files.
2. Multiple fact tables cannot coexist in the same hierarchy. (i.e. any single ontology can reference a fact table other than observation_fact, but it cannot reference multiple fact tables)

Running the scripts

See database-specific instructions below. After running the scripts, results are placed in: c_totalnum column of all ontology tables, the totalnum table (keeps a historical record), and the totalnum_report table (most recent run, obfuscated). These total counts will also be visible in the ontology browser in the web client.

MSSQL Version, February 2021

By Mike Mendis and Jeff Klann, PhD based on code by Griffin Weber, MD, PhD

Run with:

```
exec RunTotalnum or exec RunTotalnum 'observation_fact','dbo','@'
```

The optional parameters are:

1. Observation table name (for multi-fact-table setups)
2. Schema name
3. A single ontology table name (specify to to run on a single ontology table - otherwise (or if '@' is specified) runs on all tables in table_access)
4. A wildcard flag that will ignore multifact references in the ontology if 'Y'. (See below for the use case.)

Note that visit and patient dimension will only be counted in conjunction with the default (observation_fact) tablename!

To use with multi-fact-table setups:

Option 1) If you have at most one fact table per ontology, run this once with each fact table specified!
e.g., to use on a fact table called derived_fact with just the act_covid ontology: exec RunTotalnum 'derived_fact','dbo','act_covid'

Option 2) Create a fact table view as the union of all your fact tables. (This is essentially going back to a single fact table, but it is only used for totalnum counting. This is needed to correctly count patients that mention multiple fact tables within a hierarchy.)
e.g.,

Example 1: Counting using OMOP tables

```
create view observation_fact_view as
select * from CONDITION_VIEW
union all
select * from drug_view
```

And then run the totalnum counter with the wildcard flag, to ignore multifact references in the ontology, e.g.,

```
exec RunTotalnum 'observation_fact_view','dbo','@','Y'
```

Example 2: Counting using a derived fact table and the regular fact table, using a single ontology

```
create view observation_fact_view as
select * from observation_fact
union all
select * from derived_fact
```

Run the totalnum counter with the wildcard flag, to ignore multifact references in the ontology, and specify an ontology table, e.g.,

```
exec RunTotalnum 'observation_fact_view','dbo','act_covid_v4','Y'
```

Note this approach does not work if you have conflicting concept_cds across fact tables.

Oracle Version, March 2020

By Mike Mendis, based on SQL Server code by Griffin Weber, MD, PhD
Performance improvements by Jeff Green and Jeff Klann, PhD 03-20

Run the procedure like this (but with your schema name instead of i2b2demodata):

```
begin
runtotalnum('observation_fact','i2b2demodata');
end;
```

You can optionally include a table named if you only want to count one ontology table (this IS case sensitive):

```
begin
runtotalnum('observation_fact','i2b2demodata','I2B2');
end;
```

Note: If you get the error as: ERROR at line 1: ORA-01031: insufficient privilege, then run the command:
grant create table to (DB USER)

Postgres Version, March 2020

Original PostgreSQL code by Dan Vianello, Center for Biomedical Informatics, Washington University in St. Louis
2019 - Modified for i2b2 1.7.12 release by Mike Mendis, Partners Healthcare
2020 - Updated to support reporting and single-table runs by Jeff Klann, Massachusetts General Hospital

Usage example:

```
select runtotalnum('observation_fact','public')
```

- Replace 'public' by the schema name for the fact table.
- If using a schema other than public for metadata, you might need to run "set search_path to 'i2b2metadata','public' " first
- You can optionally specify a single table name, to count using only one ontology table. This is case sensitive.

Running using ANT

Run the ant command to execute the data_build.xml file with below specified target

- POSTGRESQL : ant -f data_build.xml db_metadata_run_total_count_postgresql
- ORACLE : ant -f data_build.xml db_metadata_run_total_count_oracle
- SQL SERVER : ant -f data_build.xml db_metadata_run_total_count_sqlserver

Script status 9/14/21

- **Postgres and Oracle:** totalnum counting through ontology tables with obfuscated report and performance improvements for ACT and large ontologies
- **SqlServer:** the above plus support for multiple fact tables