Building QA into the i2b2 ETL lifecycle

While we have had basic QA in the ETL pipeline for some time, it has lacked the granularity necessary to make truly informed decisions about ETL development directions. Recently, we joined the SCILHS shrine network which has some pretty granular data reporting requirements. Kun Wei at Wake Forest did some great work developing an initial Python script to meet the reporting requirements. I used this script as a launchpad and added some extensions that will allow us to use this against any i2b2 database. The output of this script is incredibly granular, summing up a count of distinct patients associated with every searchable node in the i2b2 ontology. It leverages the natural i2b2 ontology hierarchy to roll these counts up from child to parent concepts.

How it works, stepwise

1. The script connects to the i2b2 metadata schema and discovers all of your ontology entries by scanning the table_access table.
2. The script iterates through the ontology tables and uses the data within them to construct queries against your CRC schema
3. These queries return sets of patient_nums
4. The sets of patient_nums are unioned up from child to parent nodes in the ontology, creating a representation of unique patients at each level in the ontology
5. These sets are used to create a variety of useful output

Output, Explained

The script produces csv files with the distinct count information, optionally writes this same data to audit database, and creates sql files that can directly executed against your database to update the c_totalnum columns in your ontology.

CSV output

The csv output creates a 3 column file:

c_fullname	concept_cd	mycount
The c_fullname from the ontology table	concept_cd(the c_basecode from the ontology table. This >should< match the concept_cd in the concept_dimension and your observation_fact table)	Count of distinct patients. This includes all patients that have at least one instance of this concept OR any of its children

The CSV output is organize in two ways. There is one file for each of the ontology files plus one file that combines all of the entries into one master file.

> **Master File limitation**
> Due to the way the script functions, it does not create patient sets at a 'higher' level than the individual ontology tables such that there will be output produced at the true root or '\'

SQL output

The sql output creates one file per ontology file and contains one sql statement per line that updates the c_totalnum columns for every c_fullname. Example:

update_c_totalnum.sql

```
UPDATE <ontology_table> SET c_totalnum = N where c_fullname='<c_fullname>'
```

I2B2 Audit Database

This script will also optionally write to an i2b2 audit database. If you choose to take advantage of this you will need to have a table titled i2b2_data_audit instantiated. Here is the script to create it:

create_i2b2_data_audit.sql

```
CREATE TABLE I2B2_DATA_AUDIT
        (
             AUDIT_DATE DATE,
          C_FULLNAME VARCHAR2(3000 BYTE),
          CONCEPT_CD VARCHAR2(1000 BYTE),
             MYCOUNT NUMBER(8,0)
         )
```

This writes the same data as is written to the csv file, described above, plus and audit_date is added to allow for tracking of data over time. By loading this into a database, we gain access to programmatic analysis, sql and otherwise.