OpenEyes - SnoMed CT

Editors: G W Aylward

Version: 0.9.1:

Date issued: 9 December 2009

Target Audience

General Interest	
Heathcare managers	
Ophthalmologists	✔
Developers	✔

Amendment Record

Issue	Description	Author	Date
0.9	Draft	G W Aylward	22/04/2010
0.9.1	Update to discuss the addition of disorder trees	M C Smith	07/02/2013

Table of Contents

Introduction 4

Database setup 4

Creating SNOMED core tables 4

Create and populate the Concepts Table 4

Create and populate the Descriptions Table 5

Create and populate the Relationships Table 6

Extract current concepts and preferred descriptions 6

Using subsets 7

Creating a subset table 7

Creating a subset members table 8

Get subset of descriptions table 8

Disorder Subset 9

Extract data from concept & description tables 9

Create disorders table 9

Setting the Specialty ID 10

Defining the Required Trees 11

Procedures Subset 11

Create a new SNOMED database for procedures 11

Create and populate the Concepts Table 11

Create and populate the Descriptions Table 12

Create and populate the Relationships Table 13

Create procedures table 14

Populate procedures table 14

Introduction

SNOMED CT (Systematized Nomenclature of Medicine -- Clinical Terms), is a systematically organized collection of medical terminology covering most areas of clinical information such as diseases, findings, procedures, microorganisms, and drugs.

OpenEyes uses SNOMED for coding of diagnoses and procedures, with the option of adding additional areas in the future.

This document describes how to make use of a SNOMED release within OpenEyes. It assumes that the initial tables for OpenEyes have been created in your database, and data needs to be setup for SNOMED. This process relies on the RF1 format for the SNOMED data.

N.B. *Work has yet to be carried out to allow for updates to SNOMED after the system is live. Specifically the removal of no longer "current" concepts is a problem that has yet to be resolved.*

Database setup

For the purpose of loading data, you will need your mysql server to be configured to support loading data from files. In the server configuration (/etc/mysql/my.cnf), add the line:

```
local-infile=1
```

and when running the client, use the following option:

```
--local-infile
```

Creating SNOMED core tables

The three SNOMED core tables are first created from the text files provided in the release. Each text file contains tab separated fields with one line for each row of the table. The first row of each file contains column headings. The following is a step by step guide to creating and populating the tables from the text files.

This creates a large set of data, and should be used as reference data to populate OpenEyes specific data tables such as disorder, procedure and related tree tables. Sections detailing the recommended process for this are laid out in later sections.

1. Create and populate the Concepts Table

The following SQL statement creates the core Concepts table.

```
CREATE TABLE snomed_concepts (

ConceptId BIGINT UNSIGNED NOT NULL,
ConceptStatus INT UNSIGNED NOT NULL,
FullySpecifiedName CHAR(255) NOT NULL,
CTV3ID CHAR(5) NOT NULL,
SNOMEDID CHAR(8) NOT NULL,
IsPrimitive BOOL,
PRIMARY KEY (Conceptid)
);
```

Now populate the table reading from the provided text file. It is assumed that the files have been downloaded to a suitable location and are to be found in a directory at the path ‘/tmp/snomed/’. In the commands that follow replace the given path with the actual path on your machine, and replace yyyymmdd with the date stamp on the files.

The following command will load all the concepts. For use in OpenEyes it is recommended that only a subset is used.

```
LOAD DATA LOCAL INFILE '/tmp/snomed/RF1Release/Terminology/Content/
sct1_Concepts_Core_INT_yyyymmdd.txt'
INTO TABLE snomed_concepts
IGNORE 1 LINES;
```

2. Create and populate the Descriptions Table

The following SQL statement creates the core Descriptions table.

```
CREATE TABLE snomed_descriptions (
DescriptionId BIGINT UNSIGNED NOT NULL,
DescriptionStatus INT UNSIGNED NOT NULL,
ConceptId BIGINT UNSIGNED NOT NULL,
Term CHAR(255) NOT NULL,
InitialCapitalStatus BOOL,
DescriptionType INT UNSIGNED NOT NULL,
LanguageCode CHAR(8),
PRIMARY KEY (DescriptionId)
);
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL
```

```
INFILE '/tmp/snomed/RF1Release/Terminology/Content/
sct1_Descriptions_en_INT_yyyymmdd.txt'

INTO TABLE snomed_descriptions

IGNORE 1 LINES;
```

3. Create and populate the Relationships Table

```
CREATE TABLE snomed_relationships (

RelationshipId BIGINT UNSIGNED NOT NULL,

ConceptId1 BIGINT UNSIGNED NOT NULL,

RelationshipType BIGINT UNSIGNED NOT NULL,

ConceptId2 BIGINT UNSIGNED NOT NULL,

CharacteristicType INT UNSIGNED NOT NULL,

RelationshipGroup SMALLINT UNSIGNED,

PRIMARY KEY (RelationshipId)

);
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL

INFILE '/tmp/snomed/RF1Release/Terminology/Content/
sct1_Relationships_Core_INT_yyyymmdd.txt'

INTO TABLE snomed_relationships

IGNORE 1 LINES;
```

4. Extract current concepts and preferred descriptions

```
SELECT c.ConceptId, c.FullySpecifiedName, d.Term, d.DescriptionType,
d.InitialCapitalStatus

FROM snomed_concepts AS c

INNER JOIN snomed_descriptions AS d USING(ConceptId)
```

```
WHERE c.ConceptStatus = 1
AND d.DescriptionStatus = 0
AND d.DescriptionType = 'Preferred'
AND c.ConceptId = 3318611000001103
```

Note:

- ConceptStatus value of 1 is current
- DescriptionStatus value of 0 is Current
- DescriptionType of 1 is ‘Preferred’

Using subsets

SNOMED CT is a comprehensive and therefore very large terminology, intended to cover all possible healthcare purposes. The core files can be unwieldy, so smaller subsets of concepts are often useful. SNOMED CT has a subset mechanism allows the creation of subsets of any of the core tables. Lists of members of a particular subset are provided with the release, and the following steps will create tables in MySQL to make use of the subset files. These statements load the UK english subset.

1. Creating a subset table

The following SQL statement creates the subsets table

```
CREATE TABLE subsets (
SubsetId BIGINT UNSIGNED NOT NULL,
SubsetOriginalId BIGINT UNSIGNED NOT NULL,
SubsetVersion INT UNSIGNED NOT NULL,
SubsetName VARCHAR(255),
SubsetType SMALLINT UNSIGNED,
LanguageCode VARCHAR(8),
RealmId VARCHAR(24),
ContextId VARCHAR(18),
PRIMARY KEY (SubsetId)
);
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL INFILE
'/tmp/snomed/RF1Release/Subsets/Language-en-GB/der1_Subsets_en-GB_INT_yyyymmd.txt'
```

```
INTO TABLE subsets
IGNORE 1 LINES
```

2. Creating a subset members table

The following SQL statement creates the subsets members table

```
CREATE TABLE subsetmembers (
SubsetId BIGINT UNSIGNED NOT NULL,
MemberId BIGINT UNSIGNED NOT NULL,
MemberStatus INT UNSIGNED,
LinkedId BIGINT UNSIGNED NOT NULL,
PRIMARY KEY(MemberId)
)
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL INFILE
'/tmp/snomed/RF1Release/Subsets/Language-en-GB/der1_SubsetMembers_en-
GB_INT_yyyymmdd.txt'
INTO TABLE subsetmembers
IGNORE 1 LINES
```

3. Get subset of descriptions table

```
SELECT d.* from snomed_descriptions AS d INNER JOIN subsetmembers AS s ON
d.DescriptionId = s.MemberId
```

Search for a preferred description starting with given text

SELECT CONCAT(last_name, ', ', first_name, ' - ', address1) AS details, patient_id AS value

```
SELECT c.ConceptId, c.ConceptStatus, d.Term, d.DescriptionStatus,
d.DescriptionType
FROM snomed_concepts AS c
INNER JOIN snomed_descriptions AS d USING(ConceptId)
WHERE c.ConceptStatus = 1
AND d.DescriptionStatus = 0
AND d.DescriptionType = 1
AND d.Term RLIKE '^retinal'
```

Disorder Subset

In order to use SNOMED CT for recording of diagnoses in OpenEyes, it is preferable to use a subset of concepts. Analysis of the Core tables in the July 2009 release shows the reduction in rows in each of the core tables;

Table	Full Release	OpenEyes Subset
Concepts	388,289	63,840
Descriptions	1,149,406	63,840
Relationships	1,387,930	124,080

Having loaded the SnoMed concepts into the database as described above, the disorder model should be populated with the appropriate SnoMed concepts.

4. Extract data from concept & description tables

Use the following SQL to restrict table to current, preferred, descriptions that refer to the remaining concepts in the concepts table;

```
SELECT c.ConceptId, c.FullySpecifiedName, d.Term

FROM snomed_concepts AS c

INNER JOIN snomed_descriptions AS d USING(ConceptId)

INNER JOIN subsetmembers AS s ON d.DescriptionId = s.MemberId

WHERE d.DescriptionStatus = 'Current'

AND d.DescriptionType = 'Preferred'

AND d.term rlike '(disorder)'

AND c.ConceptStatus = 1

INTO OUTFILE '/tmp/disorders.txt';
```

5. Create disorders table

The disorder table should already be defined in the OpenEyes database, but for reference, the sql to create it is as follows.

```
CREATE TABLE `disorder` (

  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,

  `fully_specified_name` varchar(255) COLLATE utf8_bin NOT NULL,
  `term` varchar(255) COLLATE utf8_bin NOT NULL,

  `last_modified_user_id` int(10) unsigned NOT NULL DEFAULT '1',

  `last_modified_date` datetime NOT NULL DEFAULT '1900-01-01 00:00:00',

  `created_user_id` int(10) unsigned NOT NULL DEFAULT '1',

  `created_date` datetime NOT NULL DEFAULT '1900-01-01 00:00:00',

  `specialty_id` int(10) unsigned DEFAULT NULL,

  PRIMARY KEY (`id`),

  KEY `term` (`term`),

  KEY `disorder_last_modified_user_id_fk` (`last_modified_user_id`),

  KEY `disorder_created_user_id_fk` (`created_user_id`),

  KEY `disorder_specialty_fk` (`specialty_id`),

  CONSTRAINT `disorder_created_user_id_fk` FOREIGN KEY (`created_user_id`)
REFERENCES `user` (`id`),

  CONSTRAINT `disorder_last_modified_user_id_fk` FOREIGN KEY
(`last_modified_user_id`) REFERENCES `user` (`id`),

  CONSTRAINT `disorder_specialty_fk` FOREIGN KEY (`specialty_id`) REFERENCES
`specialty` (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=442782009 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

Load data;

```
LOAD DATA LOCAL

INFILE '/tmp/disorders.txt'
```

6. Setting the Specialty ID

The disorder table has a specialty_id, which is used to determine whether a disorder is part of a particular specialty or not. The following example is for Opthamology, but a similar approach could be taken for any other given specialty. The goal is to come up with a set of Snomed concept ids that apply to a given specialty, and set the appropriate id in the disorder table.

Note the specialty_id of opthamology is 109 in this example. This should be verified in your database, and altered accordingly.

On the command line, switch to the OpenEyes/protected directory, and run the following command:

```
‣ ./yiic buildtree Disorder 128127008
```

128127008 is the concept id for “Visual System Disorder”. This command will build the SNOMED concept tree for this, populating the disorder_tree table. *(Note the buildtree command empties the tree table for the specified model. The command should be run again as laid out below to define the full set of required Disorder trees)*

```
UPDATE disorder set specialty_id = 109 WHERE id in (SELECT distinct(id) from
disorder_tree)
```

7. Defining the Required Trees

It’s possible to defined various Disorder trees in the system. These are used by the Patient model to provide various details on the patient. The diabetes functions require that the diabetes tree is defined, for example.

Diabetes Mellitus - 73211009

```
‣ ./yiic buildtree Disorder 73211009
```

The list of required trees can be determined by looking at the Disorder model constants

Procedures Subset

This section of the document may be out of date and is subject to future review.

In order to use SNOMED CT for recording of procedures in OpenEyes, it is preferable to use a subset of concepts.

1. Create a new SNOMED database for procedures

The following SQL commands creates a new database to contain the core tables.

```
CREATE DATABASE snomed_procedures

USE snomed_procedures
```

2. Create and populate the Concepts Table

The following SQL statement creates the core Concepts table.

```
CREATE TABLE concepts (

  ConceptId bigint(20) unsigned NOT NULL,

  ConceptStatus
enum('Current','Retired','Duplicate','Outdated','Ambiguous','Erroneous','Limited
','Moved elsewhere','Pending Move','UNDEFINED') DEFAULT NULL,

  FullySpecifiedName char(255) NOT NULL,

  CTV3ID char(5) NOT NULL,

  SNOMEDID char(8) NOT NULL,
  IsPrimitive tinyint(1) DEFAULT NULL,
  PRIMARY KEY (ConceptId)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Now populate the table reading from the provided text file. It is assumed that the files have been downloaded to a suitable location and are to be found in a directory at the path '/Users/bill/Desktop/'. In the commands that follow replace the given path with the actual path on your machine. The following command will load all the concepts. For use in OpenEyes it is recommended that only a subset is used.

```
LOAD DATA LOCAL INFILE '/Users/bill/Databases/OpenEyes/SnoMed/September2011/
SnomedCT_GB1000000_20111003/Terminology/Content/
sct1_Concepts_National_GB1000000_20111003.txt' INTO TABLE concepts IGNORE 1
LINES SET ConceptStatus = if(ConceptStatus = 0, 'Current', if(ConceptStatus = 1,
'Retired', if(ConceptStatus = 2, 'Duplicate', if(ConceptStatus = 3, 'Outdated',
if(ConceptStatus = 4, 'Ambiguous', if(ConceptStatus = 5, 'Erroneous',
if(ConceptStatus = 6, 'Limited', if(ConceptStatus = 10, 'Moved elsewhere',
if(ConceptStatus = 11, 'Pending move', 'UNDEFINED')))))))));
```

3. Create and populate the Descriptions Table

The following SQL statement creates the core Descriptions table.

```
CREATE TABLE descriptions (
  DescriptionId bigint(20) unsigned NOT NULL,
  DescriptionStatus enum('Current','Non-
Current','Duplicate','Outdated','Erroneous','Limited','Inappropriate','Concept
non-current','Moved elsewhere','Pending Move','UNDEFINED') DEFAULT NULL,
  ConceptId bigint(20) unsigned NOT NULL,
  Term char(255) NOT NULL,
  InitialCapitalStatus tinyint(1) DEFAULT NULL,
  DescriptionType enum('Unspecified','Preferred','Synonym','FullySpecifiedName')
DEFAULT NULL,
  LanguageCode char(8) DEFAULT NULL,
  PRIMARY KEY (DescriptionId)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL INFILE '/Users/bill/Databases/OpenEyes/SnoMed/September2011/
SnomedCT_GB1000000_20111003/Terminology/Content/sct1_Descriptions_en-
GB_GB1000000_20111003.txt'
INTO TABLE descriptions
IGNORE 1 LINES
```

```
SET DescriptionStatus = if(DescriptionStatus = 0, 'Current',
if(DescriptionStatus = 1, 'Non-Current', if(DescriptionStatus = 2, 'Duplicate',
if(DescriptionStatus = 3, 'Outdated', if(DescriptionStatus = 5, 'Erroneous',
if(DescriptionStatus = 6, 'Limited', if(DescriptionStatus = 7, 'Inappropriate',
if(DescriptionStatus = 8, 'Concept non-current', if(DescriptionStatus = 10,
'Moved elsewhere', if(DescriptionStatus = 11, 'Pending move',
'UNDEFINED')))))))))), DescriptionType = if(DescriptionType = 0, 'Unspecified',
if(DescriptionType = 1, 'Preferred', if(DescriptionType = 2, 'Synonym',
if(DescriptionType = 3, 'FullySpecifiedName', 'UNDEFINED'))));
```

4. Create and populate the Relationships Table

```
CREATE TABLE relationships (
  RelationshipId bigint(20) unsigned NOT NULL,
  ConceptId1 bigint(20) unsigned NOT NULL,
  RelationshipType bigint(20) unsigned NOT NULL,
  ConceptId2 bigint(20) unsigned NOT NULL,
  CharacteristicType enum('Defining','Qualifier','Historical','Additional')
DEFAULT NULL,
  Refinability enum('Not refinable','Optional','Mandatory') DEFAULT NULL,
  RelationshipGroup smallint(5) unsigned DEFAULT NULL,
  PRIMARY KEY (RelationshipId)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL INFILE '/Users/bill/Databases/OpenEyes/SnoMed/September2011/
SnomedCT_GB1000000_20111003/Terminology/Content/
sct1_Relationships_National_GB1000000_20111003.txt'
INTO TABLE relationships
IGNORE 1 LINES
SET CharacteristicType = if(CharacteristicType = 0, 'Defining',
if(CharacteristicType = 1, 'Qualifier', if(CharacteristicType = 2, 'Historical',
if(CharacteristicType = 3, 'Additional', 'UNDEFINED')))), Refinability =
if(Refinability = 0, 'Not refinable', if(Refinability = 1, 'Optional',
if(Refinability = 2, 'Mandatory', 'UNDEFINED')));
```

5. Create procedures table

The procedures table combines the ConceptID, fully specified name, and preferred term, long with an enumerated field indicating whether the term is an ophthalmological disorder

```
CREATE TABLE procedures (

  id bigint(20) unsigned NOT NULL AUTO_INCREMENT,

  snomed_code bigint(20) unsigned NOT NULL,

  term char(255) CHARACTER SET latin1 NOT NULL,

  type enum('Other','Ophthalmic') CHARACTER SET latin1 DEFAULT 'Other',

  PRIMARY KEY (id),

  KEY (term)

)
```

6. Populate procedures table

The following SQl will take all the entries from the SnoMed master tables which are procedures, and load them into the procedures table

```
INSERT INTO procedures

SELECT 'NULL', c.ConceptId AS snomed_code, d.Term, 'Other' AS term FROM concepts AS c

INNER JOIN descriptions AS d USING(ConceptId)

WHERE c.ConceptStatus = 'Current'

AND d.DescriptionStatus = 'Current'

AND d.DescriptionType = 'Preferred'

AND c.FullySpecifiedName LIKE '%procedure%'
```

Oops, SnoMed procedures not fit for purpose!

User following SQL to extract a range of codes to choose from in creating custom table

```
SELECT c.ConceptId AS snomed_code, c.ConceptStatus, c.FullySpecifiedName, d.Term, d.DescriptionStatus, d.DescriptionType FROM concepts AS c

INNER JOIN descriptions AS d USING(ConceptId)
```

```
WHERE c.FullySpecifiedName LIKE '%procedure%'

AND d.Term LIKE '%phako%'
```