OpenEyes - SnoMed CT

Editors: G W Aylward
Version: 0.9:
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

Table of Contents

Introduction 4

Core Tables 4

Description 4

Fields 4

Creating SNOMED core tables 5

Create a new SNOMED database 5

Create and populate the Concepts Table 5

Create and populate the Descriptions Table 5

Create and populate the Relationships Table 6

Extract current concepts and preferred or unspecified descriptions 7

Using subsets 7

Creating a subset table 7

Creating a subset members table 8

Get subset of descriptions table 8

Finding Chidren 9

Subsets 10

Disorders Subset 11

Extract disorders from the concepts content file 12

Load truncated file into a MySQL table 12

Remove non-current concepts 12

Trim descriptions table 12

Create disorders table 13

Trim relationships table 13

SQL Statements for Disorders 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 drugs, with the option of adding additional areas in the future.

This document describes how to convert a new SNOMED release into a form useable by OpenEyes.

Core Tables

Description

The patients table contains personal details of a particular patient. It has relationships to most of the other tables in the database

Fields

Field	Type	Comments
patient_id	INT UNSIGNED NOT NULL AUTO_INCREMENT	Primary Key, 4 billion records
title	VARCHAR(8)	
first_name	VARCHAR(20) NOT NULL	
last_name	VARCHAR(40) NOT NULL	
dofb	DATE	
sex	ENUM('Male', 'Female')	
hosnum	VARCHAR(40)	
nhsnum	VARCHAR(40)	
address1	VARCHAR(40)	
address2	VARCHAR(40)	
city	VARCHAR(24)	
postcode	VARCHAR(8)	
country	VARCHAR(16)	
telephone	VARCHAR(24)	
mobile	VARCHAR(24)	
email	VARCHAR(60)	
comments	TINYTEXT	Maximum 255 characters

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.

1. Create a new SNOMED database

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

```
CREATE DATABASE snomed

USE snomed
```

2. Create and populate the Concepts Table

The following SQL statement creates the core Concepts table.

```
CREATE TABLE concepts (
     ConceptId BIGINT UNSIGNED NOT NULL,
     ConceptStatus ENUM
     ('Current','Retired','Duplicate','Outdated','Ambiguous','Erroneous','Limite
     d','Moved elsewhere','Pending Move','UNDEFINED'),
     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 '/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 '/Content/sct_concepts_20090731.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 UNSIGNED NOT NULL,
     DescriptionStatus ENUM('Current','Non-
     Current','Duplicate','Outdated','Erroneous','Limited','Inappropriate','Conc
     ept non-current','Moved elsewhere','Pending Move','UNDEFINED'),
     ConceptId BIGINT UNSIGNED NOT NULL,
     Term CHAR(255) NOT NULL,
     InitialCapitalStatus BOOL,
     DescriptionType ENUM
     ('Unspecified','Preferred','Synonym','FullySpecifiedName'),
     LanguageCode CHAR(8),
     PRIMARY KEY (DescriptionId)
     );
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL
     INFILE '/Content/sct_descriptions_20090731.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 UNSIGNED NOT NULL,
     ConceptId1 BIGINT UNSIGNED NOT NULL,
     RelationshipType BIGINT UNSIGNED NOT NULL,
     ConceptId2 BIGINT UNSIGNED NOT NULL,
     CharacteristicType ENUM('Defining','Qualifier','Historical','Additional'),
     Refinability ENUM('Not refinable','Optional','Mandatory'),
     RelationshipGroup SMALLINT UNSIGNED,
     PRIMARY KEY (RelationshipId)
     );
```

Now populate the table reading from the provided text file.

```
LOAD DATA LOCAL
     INFILE '/Content/sct_relationships_20090731.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. Extract current concepts and preferred or unspecified descriptions

SELECT c.ConceptId, c.FullySpecifiedName, d.Term, d.DescriptionType, d.InitialCapitalStatus
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.ConceptId = 22298006

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.

NB. The subset data will probably be found stored in a different directory, so change the symbolic link;

```
ln -s /Users/bill/Databases/OpenEyes/SnoMed/IHTSDO_SNOMED/
Essential\ Resources/Subsets/UK\ Language\ Subset /Content
```

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
    '/Content/sct_subsets_uk_20090731.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
    '/Content/sct_subsetmembers_uk_20090731.txt'
    INTO TABLE subsetmembers
    IGNORE 1 LINES
```

3. Get subset of descriptions table

```
SELECT d.* from 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 concepts AS c
    INNER JOIN descriptions AS d USING(ConceptId)
```

```
    WHERE c.ConceptStatus = 'Current'
    AND d.DescriptionStatus = 'Current'
    AND d.DescriptionType = 'Preferred'
    AND d.Term RLIKE '^retinal'
```

4. Finding Chidren

Use SQL to find children (138875005 is the ConceptID for the Root Concept, and 116680003 is the ConceptID for 'Is a',

```
SELECT r.ConceptId1, r.ConceptId2, c.FullySpecifiedName, r.CharacteristicType
    FROM concepts AS c, relationships AS r
    WHERE c.ConceptId = r.ConceptId1 AND r.ConceptId2 = 138875005
    AND r.RelationshipType = 116680003;
```

Parents of retinal detachment:

SELECT r.ConceptId1, r.ConceptId2, c.FullySpecifiedName, r.CharacteristicType FROM concepts AS c, relationships AS r WHERE c.ConceptId = r.ConceptId1 AND r.ConceptId2 = 42059000 AND r.RelationshipType = 116680003;

Eye disorders

SELECT r.ConceptId1, r.ConceptId2, c.FullySpecifiedName, r.CharacteristicType FROM concepts AS c, relationships AS r WHERE c.ConceptId = r.ConceptId1 AND r.ConceptId2 =118934005 AND r.RelationshipType = 116680003;

Getting descriptions for a given ConceptId

SELECT c.ConceptId, c.ConceptStatus, c.FullySpecifiedName, d.Term, d.DescriptionStatus, d.DescriptionType From concepts AS c INNER JOIN descriptions AS d USING(ConceptId) WHERE c.ConceptId = 4855003

Extract current concepts and preferred or unspecified descriptions

SELECT c.ConceptId, c.FullySpecifiedName, d.Term, d.DescriptionType, d.InitialCapitalStatus
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.ConceptId = 22298006

Subsets

The SnoMed CT subset mechanism allows the creation of subsets of any of the core tables.

Creating the subset members 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)
     )
```

Load Data

```
LOAD DATA LOCAL INFILE
     '/Users/bill/Databases/ProMed3/SnoMed/IHTSDO_SNOMED/Essential\ Resources/
     Subsets/UK\ Language\ Subset/sct_subsets_uk_20090731.txt'
     INTO TABLE subsets
     IGNORE 1 LINES
```

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

```
LOAD DATA LOCAL INFILE
     '/Users/bill/Databases/ProMed3/SnoMed/IHTSDO_SNOMED/Essential\ Resources/
     Subsets/UK\ Language\ Subset/sct_subsetmembers_uk_20090731.txt'
     INTO TABLE subsetmembers
     IGNORE 1 LINES
```

Get subset of descriptions table

```
SELECT d.* from 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 concepts AS c
     INNER JOIN descriptions AS d USING(ConceptId)
     WHERE c.ConceptStatus = 'Current'
     AND d.DescriptionStatus = 'Current'
     AND d.DescriptionType = 'Preferred'
     AND d.Term RLIKE '^retinal'
```

Disorders 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

In due course, OpenEyes will release and maintain subsets using the SNOMED CT subset mechanism(ref), but in the meantime, the following steps can be used to produce working tables

1. Extract disorders from the concepts content file

Use a text editor or grep to remove all lines without the suffix '(disorder)' in the fully specified name. Using the command line, navigate to the directory containing the content file and type the following command:

```
grep '(disorder)' sct_concepts_20090731.txt >> sct_concepts_disorders.txt
```

2. Load truncated file into a MySQL table

```
LOAD DATA LOCAL
     INFILE '/Users/bill/Databases/ProMed3/SnoMed/IHTSDO_SNOMED/Essential\
     Resources/SNOMED\ CT\ Terminology/Content/sct_concepts_disorders.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. Remove non-current concepts

```
DELETE FROM concepts WHERE ConceptStatus != 'Current';

OPTIMIZE TABLE concepts;
```

4. Trim descriptions table

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 concepts AS c
     INNER JOIN descriptions AS d USING(ConceptId)
     INNER JOIN subsetmembers AS s ON d.DescriptionId = s.MemberId
     WHERE d.DescriptionStatus = 'Current'
```

```
    AND d.DescriptionType = 'Preferred'
    INTO OUTFILE '/tmp/disorders.txt';
```

5. Create disorders table

The disorders 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 disorders (
    disorder_id BIGINT UNSIGNED NOT NULL,
    fully_specified_name CHAR(255) NOT NULL,
    term CHAR(255) NOT NULL,
    type ENUM('Systemic', 'Ophthalmic') DEFAULT 'Systemic',
    PRIMARY KEY (disorder_id)
    )
```

Load data;

```
LOAD DATA LOCAL
    INFILE '/tmp/disorders.txt'
    INTO TABLE disorders
    (disorder_id, fully_specified_name, term)
    SET type = 'systemic'
```

Add indices

```
ALTER TABLE disorders ADD INDEX (term);
```

6. Trim relationships table

Restrict entries in the relationships table to RelationshipType of 'is a' (`116680003`) with either a parent or a child in the disorders table.

```
DELETE FROM relationships WHERE RelationshipType != 116680003;

DELETE FROM relationships WHERE ConceptId1 NOT IN (SELECT disorder_id FROM
    disorders) AND ConceptId2 NOT IN (SELECT disorder_id FROM disorders);
```

Pack table

```
OPTIMIZE TABLE relationships;
```

SQL Statements for Disorders table

Finding Children;
retinal detachment = 42059000

```
SELECT d.disorder_id, d.term, r.RelationshipID
     FROM disorders AS d, relationships AS r
     WHERE d.disorder_id = r.ConceptId1
     AND r.ConceptId2 = 42059000
```

Finding Parents;

```
SELECT d.disorder_id, d.term, r.RelationshipID
     FROM disorders AS d, relationships AS r
     WHERE d.disorder_id = r.ConceptId2
     AND r.ConceptId1 = 42059000
```

NB “AND r.RelationshipType = 116680003;” is not required using trimmed version of relationships table, since all rows are of this type.