

9.2 Patient Record Queries

The query language used to query a set of patient records is usually dependent on the type of database used to store the patient records. For example:

- Relational databases may be queried using SQL (Structured Query Language)
- Object-oriented databases may be queried using OQL (Object Query Language)
- RDF databases may be queried using SPARQL (SPARQL Protocol And RDF Query Language)
- XML databases may be queried using XQuery (XML Query Language)
- OLAP databases may be queried using MDX (Multidimensional Expressions)

However, some query languages support logical queries that are independent of the application, programming languages, system environment and storage models - for example, AQL (Archetype Query Language) and EQL (EHR Query Language). These languages instead focus on queries based on the relevant information models (called 'archetypes').

To get the most benefit from using SNOMED CT in patient records, however, one must be able to not only query the records themselves, but also query SNOMED CT.

One way of achieving this is to include a list of all possible SNOMED CT codes that are required within the query. For example, to find the patients with a Respiratory system disorder, one could include every individual code that is a descendant of 50043002 |disorder of respiratory system| (around 3000 codes) within the patient record query. Using SQL, this would look like:

```
SELECT DISTINCT PatientID FROM ProblemList
```

```
WHERE Code IN (140004, 181007, 222008, 490008, 517007, 599006, 652005, 663008, etc)
```

However, this creates a lengthy query that is difficult to both validate and maintain. In some cases, it may also be too long to be accepted by the query engine.

Another approach would be to use a subset of respiratory system disorders, and load these into a separate table – for example:

```
SELECT DISTINCT PatientID FROM ProblemList
```

```
WHERE Code IN (SELECT * FROM RespiratorySystemDisorders)
```

However, it may not be scalable to create a new table for each terminology query that is required.

A third approach would be to use a transitive closure table to test the hierarchical relationship between each SNOMED CT code and 50043002 |disorder of respiratory system|. For example,

```
SELECT DISTINCT PatientID FROM ProblemList PL
```

```
INNER JOIN SNOMEDTransitiveClosure TC ON TC.SourceId = PL.Code
```

```
WHERE TC.TargetId = 50043002
```

However, to support a more advanced style of query that utilizes the full capabilities of SNOMED CT, SNOMED CT query languages or API calls must be embedded within the patient record query languages. For example, the following queries use the SNOMED CT Expression Constraint Language embedded within a SQL query.

```
SELECT DISTINCT PatientID FROM ProblemList
```

```
WHERE Code IN (< 50043002 |disorder of respiratory system| )
```

```
SELECT DISTINCT PatientID FROM ProblemList
```

```
WHERE Code IN (<< 404684003 |clinical finding|:
```

```
363698007 |finding site| = << 39057004 |pulmonary valve structure|,
```

```
116676008 |associated morphology| = << 415582006 |stenosis|)
```