# S2 Listing - SEARCH query pseudo-code implementation

**Couchbase**

bucket.query(ViewQuery.from("tests", "by\_icd").stale(Stale.FALSE).key(icd\_code))

view := "by\_icd"

function (doc, meta) {

 if(doc.versions.data.type == "COMPOSITION") {

 for(var i=0; i<doc.versions.data.content.length; i++) {

 if(doc.versions.data.content[i].archetype\_node\_id == "openEHR-EHR-EVALUATION.problem\_diagnosis-sus.v1") {

 for(var j=0; j<doc.versions.data.content[i].data.items.length; j++) {

 if(doc.versions.data.content[i].data.items[j].archetype\_node\_id == "at0002.1") { emit(doc.versions.data.content[i].data.items[j].value.defining\_code.code\_string,doc.owner\_id.id.value);

} } } } } }

**ElasticSearch**

client.prepareSearch(index)

 .setTypes(type).setScroll(new TimeValue(60000))

 .setFetchSource(new String[]{"owner\_id.id.value"}, null)

 .setQuery(QueryBuilders.boolQuery() .filter(QueryBuilders.termQuery("versions.data.content.data.items.value.defining\_code.code\_string", icd\_code))

 .filter(QueryBuilders.termQuery("versions.data.content.archetype\_node\_id", "openEHR-EHR-EVALUATION.problem\_diagnosis-sus.v1")))

 .setSize(scrollSize).execute().actionGet()

**eXist-db**

runQuery(“xquery version '3.0'

declare namespace v1 = 'http://schemas.openehr.org/v1'

declare default element namespace 'http://schemas.openehr.org/v1'

declare namespace xsi = 'http://www.w3.org/2001/XMLSchema-instance'

for $match in collection('/db/ORBDA')//versioned\_composition//\*[code\_string = '" + icd\_code + "']

 return <value>{$match/ancestor::\*/owner\_id/id/value/text()}</value>”

)