

# DATA STORE FOR SHARP N TO STORE CEM DATA

# REQUIREMENTS (1/2)

Store disparate CEM types

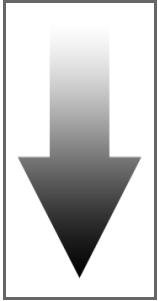
- Labs
- Medications
- Administrative Diagnosis
- Demographics
- Disease/Disorder

*All these types are available on SharpN web page*

## REQUIREMENTS (2/2)

Ability to easily store new CEMs quickly

Easily query to retrieve datasets as needed



**(#/2/1)**

# TRIDITIONAL METHODOLOGY WE HAVE USED SO FAR

- Relational Schema
  - CEMS were modeled first
  - We took a subset of the CEM to model the relational schema
  - Manually identified fields to create an index table from CEM
- We began to add more and more CEMs and use cases, we were having difficulty in deciding what fields were to be picked

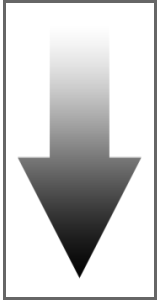
# THIS PRESENTS A PROBLEM:

- New types of CEMs
- Had to hand pick the fields which could be queried
- New indices to be created based on the algorithm to be executed
- We had to guess the use cases to achieve these steps

# WE NEEDED A NEW APPROACH

- All fields are present
- All fields can be queried
- All fields can be indexed
- Any use case can be supported

# COUCH DB



**(#/2/1)**

# WHAT IS COUCHDB?

- It is an open source Apache foundation project
- CouchDB is a JSON document-oriented database written in **Erlang**  
(<http://erlang.org/>)
- It is a highly concurrent database designed to be easily replicable, horizontally, across numerous devices and be fault tolerant
- A RESTful HTTP/JSON API accessible from many programming libraries and tools

Sources: **CouchDB**,  
(<http://wiki.apache.org/couchdb/>)  
**nettuts**,  
(<http://net.tutsplus.com>)



The screenshot shows the CouchDB Futon interface. At the top, there's a navigation bar with 'Most Visited', 'CouchDB', and 'mayo' tabs. Below that is the 'Overview' section with a '+ Create Database ...' button. A table lists several databases with their names, sizes, document counts, and update sequences. A red box highlights the first three rows of the table, and a red arrow points from the text 'Databases' below to this box. To the right is a sidebar with the CouchDB logo and a 'Tools' menu. The 'Tools' menu is highlighted with a red box, and a red arrow points from the text 'In-built tools' below to it. The 'Recent Databases' section in the sidebar lists several database names, including 'secondaryusenoteddrug', 'secondaryusepatient', 'sharpn\_meds', 'sharpn\_patient', and 'sharpn\_test'. At the bottom right, there are links for 'Signup or Login' and the version 'Futon on Apache CouchDB 1.0.1'.

Name	Size	Number of Documents	Update Seq
secondaryusestandardlab	79 bytes	0	0
secondaryusepatient	268.1 KB	41	42
secondaryusenoteddrug	308.1 KB	42	42
_users	12.1 KB	3	3
sharpn_labs	79 bytes	0	0
sharpn_meds	280.1 KB	43	43
sharpn_patient	228.1 KB	43	43

Showing 1-7 of 7 databases | ← Previous Page | Rows per page: 10 | Next Page →

**Databases**

**In-built tools**

**Tools**

- Overview
- Configuration
- Replicator
- Status
- Test Suite

**Recent Databases**

- secondaryusenoteddrug
- secondaryusepatient
- sharpn\_meds
- sharpn\_patient
- sharpn\_test

Signup or Login

Futon on Apache CouchDB 1.0.1

Overview > secondaryusenoteddrug

+ New Document | Security... | Compact & Cleanup... | Delete Database... | Jump to: Document ID | View: All documents | Stale views

Key	Value
"55541e10-09b6-11e2-9e43-89deb817d48a" ID: 55541e10-09b6-11e2-9e43-89deb817d48a	{rev: "1-6b21f92ca1a99edacbe7fcec0843b3b5"}
"55047766-09b6-11e2-9e43-89deb817d48a" ID: 55047766-09b6-11e2-9e43-89deb817d48a	{rev: "1-098e1d0cf65b4fac70cf92b26eff9fbb"}
"54c1f01c-09b6-11e2-9e43-89deb817d48a" ID: 54c1f01c-09b6-11e2-9e43-89deb817d48a	{rev: "1-f13390021ca0e87cac52286f6a494997"}
"547863f2-09b6-11e2-9e43-89deb817d48a" ID: 547863f2-09b6-11e2-9e43-89deb817d48a	{rev: "1-8d76ad3a53c5cc7fa392c36ea7803c26"}
"5431e508-09b6-11e2-9e43-89deb817d48a" ID: 5431e508-09b6-11e2-9e43-89deb817d48a	{rev: "1-60cc1985ec433a434db6c7a36363e8c3"}
"53e5e7de-09b6-11e2-9e43-89deb817d48a" ID: 53e5e7de-09b6-11e2-9e43-89deb817d48a	{rev: "1-cf85117bfb58f786d40cf4e498c7e4a3"}
"53a18bd4-09b6-11e2-9e43-89deb817d48a" ID: 53a18bd4-09b6-11e2-9e43-89deb817d48a	{rev: "1-76c17f62576396a3dc5b7f34b219974c"}
"5357154a-09b6-11e2-9e43-89deb817d48a" ID: 5357154a-09b6-11e2-9e43-89deb817d48a	{rev: "1-d419d8c96f4eebfd064a51a4132564e4"}
"53096a70-09b6-11e2-9e43-89deb817d48a" ID: 53096a70-09b6-11e2-9e43-89deb817d48a	{rev: "1-8e641a943e92220dccccfaa3ddc83a43"}
"52c61fd6-09b6-11e2-9e43-89deb817d48a" ID: 52c61fd6-09b6-11e2-9e43-89deb817d48a	{rev: "1-f695e7a2d4f145a389d72ca6e9b017bc"}

Showing 1-10 of 42 rows | << Previous Page | Rows per page: 10 | Next Page >>



The screenshot shows a web interface for editing a CouchDB document. The document ID is 55541e10-09b6-11e2-9e43-89deb817d48a. The document content is a JSON object with the following structure:

```
{
  "_id": "55541e10-09b6-11e2-9e43-89deb817d48a",
  "_rev": "1-6b21f92ca1a99edacbe7fcec0043b3b5",
  "SecondaryUseNotedDrug": {
    "key": {
      "isNull": {
        "value": false
      },
      "code": {
        "value": {
          "value": "3d2c5cbe-cafc-4977-ac7e-c0e819e251aa"
        }
      },
      "codeSystem": {
        "value": {
          "value": "b79119e6-cc04-40e2-afba-588277feb73b"
        }
      },
      "originalText": {
        "value": "NOTEDDRUG"
      },
      "type": "CD",
      "clazz": "DATA"
    },
    "cd": {
      "isNull": {
        "value": false
      }
    }
  }
}
```

# HOW DO WE RETIEVE THE DATA?

# ELASTIC SEARCH

# WHAT IS ELASTICSEARCH?

- Open Source, Apache 2 license
- Search engine built on Apache Lucene
- Support RESTful HTTP/JSON API
- Painless setup, completely free search schema (mapping)
- Search servers always available
- Fast, real-time, scalable search infrastructure
- It is built for the cloud

The screenshot shows the ElasticSearch browser interface. At the top, the URL is `http://172.24.156.147:9200/_plugin/head/`. The cluster name is `sharpn-147-master` and the cluster health is `yellow (1, 11)`. The interface includes tabs for Overview, Browser, Structured Query, and Any Request. Below the navigation, there are sections for All Indices, Indices, Types, and Fields. The main area displays search results in a table with columns for `_index`, `_type`, `_id`, `_score`, and `_rev`. The results list various documents from the `secondaryusenoteddrugidx` index, including fields like `source`, `originalText`, `usage`, and `translation`.

_index	_type	_id	_score	_rev
secondaryusenoteddrugidx	secondaryusenoteddrug	5105d2fa-09b6-11e2-9e43-89d817d48a	1	1-2f8c440719afd85ee
secondaryusenoteddrugidx	secondaryusenoteddrug	51e81bc8-09b6-11e2-9e43-89d817d48a	1	1-dc0c0397f1f9631f8
secondaryusenoteddrugidx	secondaryusenoteddrug	53a18bd4-09b6-11e2-9e43-89d817d48a	1	1-76c17f62576396a3
secondaryusenoteddrugidx	secondaryusenoteddrug	53e5e7de-09b6-11e2-9e43-89d817d48a	1	1-cf05117bf58f786d
secondaryusenoteddrugidx	secondaryusenoteddrug	55541e10-09b6-11e2-9e43-89d817d48a	1	1-6b21f92ca1a99eda
secondaryusenoteddrugidx	secondaryusenoteddrug	4a0f4acc-09b6-11e2-9e43-89d817d48a	1	1-eaebfa0dda9cb770f
secondaryusenoteddrugidx	secondaryusenoteddrug	4d0531be-09b6-11e2-9e43-89d817d48a	1	1-8834d59882ea34ac
secondaryusenoteddrugidx	secondaryusenoteddrug	4d8f4962-09b6-11e2-9e43-89d817d48a	1	1-d51d7579233d75ef
secondaryusenoteddrugidx	secondaryusenoteddrug	4eb3090a-09b6-11e2-9e43-89d817d48a	1	1-3b9a6922a461cb43
secondaryusenoteddrugidx	secondaryusenoteddrug	4f4dc27e-09b6-11e2-9e43-89d817d48a	1	1-4f70c2de880750e9
secondaryusenoteddrugidx	secondaryusenoteddrug	51597144-09b6-11e2-9e43-89d817d48a	1	1-091fee35a5722cb7f
secondaryusenoteddrugidx	secondaryusenoteddrug	523689f2-09b6-11e2-9e43-89d817d48a	1	1-f328cbc97656f28a
secondaryusenoteddrugidx	secondaryusenoteddrug	52c61fd6-09b6-11e2-9e43-89d817d48a	1	1-f695e7a2d4f145a3f
secondaryusenoteddrugidx	secondaryusenoteddrug	53096a70-09b6-11e2-9e43-89d817d48a	1	1-8e641a943e92220c
secondaryusenoteddrugidx	secondaryusenoteddrug	5357154a-09b6-11e2-9e43-89d817d48a	1	1-d419d8c96f4eefbdc
secondaryusenoteddrugidx	secondaryusenoteddrug	547863f2-09b6-11e2-9e43-89d817d48a	1	1-8d76ad3a53c5cc7fe
secondaryusenoteddrugidx	secondaryusenoteddrug	4b9dd95c-09b6-11e2-9e43-89d817d48a	1	1-c3444b9554ec552
secondaryusenoteddrugidx	secondaryusenoteddrug	4be1c036-09b6-11e2-9e43-89d817d48a	1	1-d23bc6510d5db595
secondaryusenoteddrugidx	secondaryusenoteddrug	4dd8123c-09b6-11e2-9e43-89d817d48a	1	1-ea46133886ec4598
secondaryusenoteddrugidx	secondaryusenoteddrug	51a6a6ee-09b6-11e2-9e43-89d817d48a	1	1-1420a0459934823c
secondaryusenoteddrugidx	secondaryusenoteddrug	49778dd8-09b6-11e2-9e43-89d817d48a	1	1-c1c7e9a939e00cdb
secondaryusenoteddrugidx	secondaryusenoteddrug	49c374b2-09b6-11e2-9e43-89d817d48a	1	1-982d77315adde0c
secondaryusenoteddrugidx	secondaryusenoteddrug	4af760fa-09b6-11e2-9e43-89d817d48a	1	1-6956cff559535ecf
secondaryusenoteddrugidx	secondaryusenoteddrug	4cc23544-09b6-11e2-9e43-89d817d48a	1	1-24413a3ea8ee5b65
secondaryusenoteddrugidx	secondaryusenoteddrug	4e2017c6-09b6-11e2-9e43-89d817d48a	1	1-1d7a3fd5574f9a0fc
secondaryusenoteddrugidx	secondaryusenoteddrug	4e67cf30-09b6-11e2-9e43-89d817d48a	1	1-32a68513e06b6faa
secondaryusenoteddrugidx	secondaryusenoteddrug	4f9022b8-09b6-11e2-9e43-89d817d48a	1	1-f9e3ba943cb6cd28
secondaryusenoteddrugidx	secondaryusenoteddrug	506d6376-09b6-11e2-9e43-89d817d48a	1	1-7cff5c0f2d8ffaeacdd

The screenshot shows the ElasticSearch web interface. At the top, the browser address bar shows 'http://172.24.156.147:9200/\_plugin/head/'. The page title is 'ElasticSearch' and the cluster status is 'sharprn-147-master cluster health: yellow (1, 11)'. The search bar contains the query 'secondaryusenoteddrugidx (42 docs) for documents where: must secondaryusenoteddrug.SecondaryUseNotedDrug.cd.originalText.value query\_string Aspirin'. The search results are displayed in a table with the following data:

_index	_type	_id	_score	_rev	clazz	type	usage	value
secondaryusenoteddrugidx	secondaryusenoteddrug	55541e10-09b6-11e2-9e43-89deb817d48a	2.5040774	1-6b21f92ca1a99edacbe7fcec0043b3b5	DATA	CD	QUALIFIER	1191

Red annotations include:

- A red arrow pointing to the 'query\_string' tab with the text: *Tabs: build query, inspect data*
- A red arrow pointing to the 'secondaryusenoteddrugidx' index selection with the text: *pick an index*
- A red box around the query construction area with the text: *select fields to build query*



# DEFINING A VIEW - SIMPLE MAP FUNCTION

```
function(doc) {  
    if(doc.patient.type == "header"){  
        emit(doc.patient.join_id, doc);  
    }  
}
```

## QUERYING DATA(1/3)

```

public class ExampleSearch5
    extends BaseSearch{
    public void example5() throws JAXBException,
        IllegalArgumentException, IllegalAccessException,
        InvocationTargetException{
        SearchResponse sr = searchField(
            MEDS_CLINICAL_DRUG_ORIG_TEXT,
            "Aspirin", SEC_USE_MEDICATION_IDX,
            SearchType.DEFAULT);
        SearchHits hits = sr.getHits();

        for(SearchHit hit:hits)
            System.out.println(hit.getSourceAsString());
    }

    @SuppressWarnings("restriction")
    public static void main(String[] args) throws Exception{
        ExampleSearch5 es = new ExampleSearch5();
        es.init();
        es.example5();
        es.cleanup();
    }
}

```

## QUERYING DATA (2/3)

```

public class ExampleSearch2
    extends BaseSearch{
    private void example2() throws JAXBException{
        List patIds = new ArrayList();
        SearchResponse sr = searchField(PATIENT_GENDER_FIELD, "F",
            SEC_USE_PATIENT_IDX, SearchType.DEFAULT);
        SearchHits hits = sr.getHits();
        for(SearchHit hit:hits){
            SecondaryUsePatient sp = Helpers.getSecondaryUsePatientInstance(hit.getSourceAsString());
            String patientId = ((PatientExternalId)sp.getPatientExternalId()
                .get(0)).getII().getExtension().getValue();

            if(!patIds.contains(patientId)) patIds.add(patientId);
        }
        for(String patid:patIds){
            sr = searchField(MEDS_PAT_EXT_ID_FIELD, String.valueOf(patid), SEC_USE_MEDICATION_IDX, SearchType.DEFAULT);
            hits = sr.getHits();
            for(SearchHit hit:hits){
                SecondaryUseNotedDrug sund = Helpers.getSecondaryUseNotedDrugInstance(hit.getSourceAsString());
            }
        }
    }
}

```

```
        System.out.println("Medication Original Text:"+
            sund.getClinicalDrug().getCD().getOriginalText().getValue());
    }
}
}
//-- main method not shown - see previous example
}
```

## QUERYING DATA (3/3)

SQL-like query which is mapped on to elasticsearch query:

```
SELECT
  personName.familyName.st.value.value,
  personName.givenName.st.value.value,
  administrativeGender.cd.code.value.value,
  patientExternalId.ii.correlationId.value
FROM
  SecondaryUsePatient
WHERE
  SHOULD secondaryusepatient.SecondaryUsePatient
    .administrativeGender.cd.code.value.value=F
  MUST secondaryusepatient.SecondaryUsePatient
    .patientExternalId.ii.correlationId.value=*
```

This is a preliminary implementation - a lot more to do here

## CHALLENGES

- Storing XML data which is not native to CouchDB
- Retrieving JSON documents when then needs to be transformed back to XML
- Still working on: Multiple instances of Patient records



