Full text search->

Select \* from hjt where word like ‘%word%’

Drawback is no scalability

Technical way is break word in tokens , algorithm for phrases and proximity search

What are the basic search features?

Search keywords phrases

Positive and negative terms

Sort relevancy newness

Pagination

Compact summary in results

SPEED

Advanced search features:

1. Facets
2. Language specific processing
3. Domain specific( Gray , grey , GREY)
4. Geographic search
5. Did you mean(Spellcheck) , more like this(related searches) , autocomplete(typeahead)
6. Scaling/clustering

Configuration files:

1. Schema.xml
2. Solr.xml
3. Solrconfig.xml
4. **Core.properties** − This file contains the configurations specific to the core. It is referred for **core discovery**, as it contains the name of the core and path of the data directory. It can be used in any directory, which will then be treated as the **core directory**.

If necessary, two cores in a Solr application can communicate with each other.

What is shards?

Shards is a logical indexing of documents. While indexing the document ,uses hash key and puts it in shards

Diff b/w schema n schema less?

Has schema.xml eg Solr .doesnot have schema ex Elastic search

Steps:

1. Set up solr in local
2. Create/delete cores
3. Add data( xml, csv, pdf) called as indexing

Other way includes using post command to add data

How to add XML documents in solr?

Example:

<add>

<doc>

<field name = "id">006</field>

<field name = "first name">Archana</field>

<field name = "last name">Mishra</field>

<field name = "phone">9848022335</field>

<field name = "city">Chennai</field>

</doc>

</add>

* **add** − This is the root tag for adding documents to the index. It contains one or more documents that are to be added.
* **doc** − The documents we add should be wrapped within the <doc></doc> tags. This document contains the data in the form of fields.
* **field** − The field tag holds the name and value of the fields of the document.

[Hadoop@localhost bin]$ ./post -c my\_core sample.xml

From bin directory in solr to my\_core

UPDATING the document:

Can be done in XML or using Java api client

DELETING a field:

<delete>

<id>003</id>

<id>005</id>

<id>004</id>

<id>002</id>

</delete>

[Hadoop@localhost bin]$ ./post -c my\_core delete.xml

Save the delete ids in a particular file and make a post command

Documents can also be deleted as a whole

QUERYING:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| q | This is the main query parameter of Apache Solr, documents are scored by their similarity to terms in this parameter. |
| fq | This parameter represents the filter query of Apache Solr the restricts the result set to documents matching this filter. |
| start | The start parameter represents the starting offsets for a page results the default value of this parameter is 0. |
| rows | This parameter represents the number of the documents that are to be retrieved per page. The default value of this parameter is 10. |
| sort | This parameter specifies the list of fields, separated by commas, based on which the results of the query is to be sorted. |
| fl | This parameter specifies the list of the fields to return for each document in the result set. |
| wt | This parameter represents the type of the response writer we wanted to view the result. |

FACETING:

Combines the result and shows its distribution based on a field.

Eg books data

Facet is author name

It will show distinct author names with no.of books by each author.

Comes under the section facet counts -🡪 facet fields and so on

There are two types of facets available in Apache Solr −

* **Query faceting** − It returns the number of documents in the current search results that also match the given query.
* **Date faceting** − It returns the number of documents that fall within certain date ranges.