

## Monica Santana – Exercise 8

### Mynewcollection in Solr web interface

The screenshot displays the Solr web interface for the 'mynewcollection' core. The left sidebar contains navigation links: Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, mynewcollection (selected), Overview, Analysis, Documents, Parameters, Files, Ping, Plugins / Stats, Query, Replication, Schema, and Segments info. The main content area is divided into three sections: Statistics, Instance, and Replication (Leader). The Statistics section shows: Last Modified: -, Num Docs: 0, Max Doc: 0, Deleted Docs: 0, Version: 2, Segment Count: 0, and Current: ✓. The Instance section shows: CWD: /opt/solr-9.6.1/server, Instance: /var/solr/data/mynewcollection, Data: /var/solr/data/mynewcollection/data, Index: /var/solr/data/mynewcollection/data/index, and Impl: org.apache.solr.core.NRTCacheDirectoryFactory. The Replication (Leader) section shows a table with columns: Version, Gen, and Size. The table has two rows: Leader (Searching) with values 0, 1, and 69 bytes; and Leader (Replicable) with values -, -, and -. A Healthcheck section at the bottom indicates: Ping request handler is not configured with a healthcheck file. At the bottom right, there are links for Documentation, Solr Query Syntax, Community, Issue Tracker, Slack, and IRC.

### Added json data in the mynewcollection and the queries

```
santanamonica@dscbigdata: ~/dsc650-infra/believe-bigdata/solr
Posting files to [base] url http://localhost:8983/solr/mynewcollection/update...
Entering auto mode. File endings considered are xml,json,jsonl,csv,pdf,doc,docx,
ppt,pptx,xls,xlsx,odt,odp,ods,ott,otp,ots,rtf,htm,html,txt,log
POSTing file products.json (application/json) to [base]/json/docs
1 files indexed.
COMMITting Solr index changes to http://localhost:8983/solr/mynewcollection/upda
te...
Time spent: 0:00:00.645
solr@ea89058bdc35:/opt/solr-9.6.1$ curl "http://localhost:8983/solr/mynewcollect
ion/select?q=*:*"
{
  "responseHeader":{
    "status":0,
    "QTime":76,
    "params":{
      "q":"*:*"
    }
  },
  "response":{
    "numFound":2,
    "start":0,
    "numFoundExact":true,
    "docs":[
      {
        "id":"1",
        "name":["Product A"],
        "category":["Electronics"],
        "price":100,
        "version_":1805850296162713600
      },
      {
        "id":"2",
        "name":["Product B"],
        "category":["Books"],
        "price":20,
        "version_":1805850296296931328
      }
    ]
  }
}
solr@ea89058bdc35:/opt/solr-9.6.1$ curl "http://localhost:8983/solr/mynewcollect
ion/select?q=category:Electronics"
{
  "responseHeader":{
    "status":0,
    "QTime":122,
    "params":{
      "q":"category:Electronics"
    }
  },
  "response":{
    "numFound":1,
    "start":0,
    "numFoundExact":true,
    "docs":[
      {
        "id":"1",
        "name":["Product A"],
        "category":["Electronics"],
        "price":100,
        "version_":1805850296162713600
      }
    ]
  }
}
solr@ea89058bdc35:/opt/solr-9.6.1$
```

## Counting the number of products in each category by filtering them separately, first electronics

The screenshot shows the Solr Admin interface for the 'mynewcollection' core. The left sidebar contains navigation links: Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, mynewcollection (selected), Overview, Analysis, Documents, Parameters, Files, Ping, Plugins / Stats, Query (selected), Replication, Schema, and Segments info. The main panel is titled 'Request-Handler (qt)' and shows the following configuration:

- q: /select
- common: category:Electronics
- q.op: OR
- fq: (empty)
- sort: (empty)
- start, rows: 0, 10
- fl: (empty)
- df: (empty)
- paramset(s): (empty)
- wt: (empty)
- indent on: ☒
- debugQuery: ☐
- defType: (empty)
- hl: ☐
- facet: ☐
- spatial: ☐
- spellcheck: ☐
- Raw Query Parameters: (empty)
- JSON Query: (empty)

The 'Execute Query' button is at the bottom. The right pane shows the HTTP request and the JSON response:

```
http://localhost:8983/solr/mynewcollection/select?indent=true&q.op=OR&q.category%3AElectronics&useParams=

{
  "responseHeader": {
    "status": 0,
    "qtime": 1,
    "params": {
      "q": "category:Electronics",
      "indent": "true",
      "q.op": "OR",
      "useParams": "",
      "_: "1722193450808"
    }
  },
  "response": {
    "numfound": 1,
    "start": 0,
    "numfoundexact": true,
    "docs": [
      {
        "id": "1",
        "name": "Product A",
        "category": ["Electronics"],
        "price": 100,
        "version": "180505029629692715600"
      }
    ]
  }
}
```

## Now books category

The screenshot shows the Solr Admin interface for the 'mynewcollection' core. The left sidebar contains navigation links: Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, mynewcollection (selected), Overview, Analysis, Documents, Parameters, Files, Ping, Plugins / Stats, Query (selected), Replication, Schema, and Segments info. The main panel is titled 'Request-Handler (qt)' and shows the following configuration:

- q: /select
- common: category:Books
- q.op: OR
- fq: (empty)
- sort: (empty)
- start, rows: 0, 10
- fl: (empty)
- df: (empty)
- paramset(s): (empty)
- wt: (empty)
- indent on: ☒
- debugQuery: ☐
- defType: (empty)
- hl: ☐
- facet: ☐
- spatial: ☐
- spellcheck: ☐
- Raw Query Parameters: (empty)
- JSON Query: (empty)

The 'Execute Query' button is at the bottom. The right pane shows the HTTP request and the JSON response:

```
http://localhost:8983/solr/mynewcollection/select?indent=true&q.op=OR&q.category%3ABooks&useParams=

{
  "responseHeader": {
    "status": 0,
    "qtime": 1,
    "params": {
      "q": "category:Books",
      "indent": "true",
      "q.op": "OR",
      "useParams": "",
      "_: "1722193450808"
    }
  },
  "response": {
    "numfound": 1,
    "start": 0,
    "numfoundexact": true,
    "docs": [
      {
        "id": "2",
        "name": "Product B",
        "category": ["Books"],
        "price": 20,
        "version": "1805050296296921328"
      }
    ]
  }
}
```

First query is selecting products with a price up to \$25, which would be the book

The screenshot shows the Solr Admin interface at `localhost:8983/solr/#/mynewcollection/query?q=*:*&q.op=OR&indent=true&fq=price:%5B*%20TO%2025%5D&useParams=`. The left sidebar contains navigation links: Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, mynewcollection (selected), Overview, Analysts, Documents, Params, Files, Ping, Plugins / Stats, Query (selected), Replication, Schema, and Segments info. The main panel displays the query configuration for the 'mynewcollection' core. The 'q' field is set to '\*:\*'. The 'q.op' is set to 'OR'. The 'fq' field is set to 'price:[\* TO 25]'. The 'start' is 0 and 'rows' is 10. The 'wt' is set to 'json'. The 'indent on' checkbox is checked. The 'JSON Query' field is empty. The 'Execute Query' button is at the bottom. The right panel shows the JSON response:

```
{
  "responseHeader": {
    "status": 0,
    "qtime": 16,
    "params": {
      "q": "*:*",
      "indent": "true",
      "q.op": "OR",
      "fq": "price:[* TO 25]",
      "useParams": "true",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 1,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "id": "2",
        "name": "Product 2",
        "category": "books",
        "price": 20
      }
    ]
  }
}
```

Second query is selecting products with a price up to \$50 in the electronics category, which is 0

The screenshot shows the Solr Admin interface at `localhost:8983/solr/#/mynewcollection/query?q=category:Electronics&q.op=OR&indent=true&fq=price:%5B*%20TO%2050%5D&useParams=`. The left sidebar is the same as the first screenshot. The main panel displays the query configuration for the 'mynewcollection' core. The 'q' field is set to 'category:Electronics'. The 'q.op' is set to 'OR'. The 'fq' field is set to 'price:[\* TO 50]'. The 'start' is 0 and 'rows' is 10. The 'wt' is set to 'json'. The 'indent on' checkbox is checked. The 'JSON Query' field is empty. The 'Execute Query' button is at the bottom. The right panel shows the JSON response:

```
{
  "responseHeader": {
    "status": 0,
    "qtime": 16,
    "params": {
      "q": "category:Electronics",
      "indent": "true",
      "q.op": "OR",
      "fq": "price:[* TO 50]",
      "useParams": "true",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 0,
    "start": 0,
    "numFoundExact": true,
    "docs": []
  }
}
```

Last query is sorting all the data in a descending order so now product B, books, is at top

The screenshot displays the Solr Admin UI at localhost:8983/solr/#/mynewcollection. The left sidebar shows the 'Query' tab selected. The main interface is split into two sections. The left section, 'Request-Handler (qt)', contains a form for query parameters: 'q' is set to '\*:\*' (common), 'q.op' is 'OR', 'fq' is 'id%20desc', 'sort' is 'id desc', 'start' is 0, 'rows' is 10, and 'indent on' is checked. The right section displays the JSON response of the query. The response includes a 'responseHeader' with status, time, and params, and a 'response' object with 'numfound', 'start', 'numfoundExact', 'docs' (a list of two product documents), and 'debugInfo'.

Created myowncollection in Solr and the queries

echo '['

```
{ "id": "1", "name": "banana", "category": "fruit", "shoppinglist": "yes" },
{ "id": "2", "name": "carrot", "category": "vegetable", "shoppinglist": "no" },
{ "id": "3", "name": "watermelon", "category": "fruit", "shoppinglist": "yes" },
{ "id": "4", "name": "chips", "category": "salty", "shoppinglist": "yes" },
{ "id": "5", "name": "onion", "category": "vegetable", "shoppinglist": "no" },
{ "id": "6", "name": "cupcake", "category": "sweet", "shoppinglist": "yes" },
{ "id": "7", "name": "chocolate", "category": "sweet", "shoppinglist": "yes" },
{ "id": "8", "name": "seaweed", "category": "salty", "shoppinglist": "yes" },
{ "id": "9", "name": "fish", "category": "seafood", "shoppinglist": "no" },
{ "id": "10", "name": "mango", "category": "fruit", "shoppinglist": "yes" },
{ "id": "11", "name": "egg", "category": "dairy", "shoppinglist": "no" },
{ "id": "12", "name": "cheese", "category": "dairy", "shoppinglist": "no" },
```

]' > /tmp/products.json

## Myowncollection and the generated data

```
santanamonica@dscbigdata: ~/dsc650-infra/bellevue-bigdata/solr
solr@ea89058bde35:/opt/solr-9.6.1$ /opt/solr/bin/solr create -c myowncollection
WARNING: Using default configset with data driven schema functionality. NOT RECOMMENDED for production use.
To turn off: bin/solr config -c myowncollection -p 8983 -action set-use
r-property -property update.autoCreateFields -value false

Created new core 'myowncollection'
solr@ea89058bde35:/opt/solr-9.6.1$ echo '[{"id":"1", "name":"banana", "category":"fruit", "shoppinglist":["yes"]}, {"id":"2", "name":"carrot", "category":"vegetable", "shoppinglist":["no"]}, {"id":"3", "name":"watermelon", "category":"fruit", "shoppinglist":["yes"]}, {"id":"4", "name":"chips", "category":"salty", "shoppinglist":["yes"]}, {"id":"5", "name":"onion", "category":"vegetable", "shoppinglist":["no"]}, {"id":"6", "name":"cupcake", "category":"sweet", "shoppinglist":["yes"]}, {"id":"7", "name":"chocolate", "category":"sweet", "shoppinglist":["yes"]}, {"id":"8", "name":"seaweed", "category":"salty", "shoppinglist":["yes"]}, {"id":"9", "name":"fish", "category":"seafood", "shoppinglist":["no"]}, {"id":"10", "name":"mango", "category":"fruit", "shoppinglist":["yes"]}, {"id":"11", "name":"egg", "category":"dairy", "shoppinglist":["no"]}, {"id":"12", "name":"cheese", "category":"dairy", "shoppinglist":["no"]} ]' > /tmp/products.json
solr@ea89058bde35:/opt/solr-9.6.1$ /opt/solr/bin/post -c myowncollection /tmp/products.json
The bin/post script is deprecated in favour of the bin/solr post command. Please update your scripts.
/opt/java/openjdk/bin/java -classpath /opt/solr/server/solr-webapp/webapp/WEB-INF/lib/solr-core-9.6.1.jar -Dauto=yes -Dc=myowncollection -Ddata-files org.apache.solr.cli.SimplePostTool /tmp/products.json
SimplePostTool version 9.6.1
Posting files to [base] url http://localhost:8983/solr/myowncollection/update...
Entering auto mode. File endings considered are xml,json,xml,jsonl,doc,docx,txt,log
Posting file products.json (application/json) to [base]/json/docs
1 files indexed.
COMMITTING Solr index changes to http://localhost:8983/solr/myowncollection/update...
Time spent: 0:00:00.557
solr@ea89058bde35:/opt/solr-9.6.1$ curl "http://localhost:8983/solr/myowncollection/select?q=*:*"
{"responseHeader":{"status":0,"QTime":1,"params":{"q":"*:*"}}, "response":{"numFound":12,"start":0,"numFoundExact":true,"docs":[{"id":"1","name":["banana"], "category":["fruit"], "shoppinglist":["yes"], "version_":1805858216499740672}, {"id":"2",
```

## Queries of entire data and categories of vegetable and fruit

```
santanamonica@dscbigdata: ~/dsc650-infra/bellevue-bigdata/solr
COMMITTING Solr index changes to http://localhost:8983/solr/myowncollection/update...
Time spent: 0:00:00.557
solr@ea89058bde35:/opt/solr-9.6.1$ curl "http://localhost:8983/solr/myowncollection/select?q=*:*"
{"responseHeader":{"status":0,"QTime":1,"params":{"q":"*:*"}}, "response":{"numFound":12,"start":0,"numFoundExact":true,"docs":[{"id":"1","name":["banana"], "category":["fruit"], "shoppinglist":["yes"], "version_":1805858216499740672}, {"id":"2","name":["carrot"], "category":["vegetable"], "shoppinglist":["no"], "version_":1805858216508129280}, {"id":"3","name":["watermelon"], "category":["fruit"], "shoppinglist":["yes"], "version_":1805858216509177856}, {"id":"4","name":["chips"], "category":["salty"], "shoppinglist":["yes"], "version_":1805858216510226432}, {"id":"5","name":["onion"], "category":["vegetable"], "shoppinglist":["no"], "version_":1805858216511275008}, {"id":"6","name":["cupcake"], "category":["sweet"], "shoppinglist":["yes"], "version_":1805858216512323584}, {"id":"7","name":["chocolate"], "category":["sweet"], "shoppinglist":["yes"], "version_":1805858216513372160}, {"id":"8","name":["seaweed"], "category":["salty"], "shoppinglist":["yes"], "version_":1805858216514418816}, {"id":"9","name":["fish"], "category":["seafood"], "shoppinglist":["no"], "version_":1805858216515465344}, {"id":"10","name":["mango"], "category":["fruit"], "shoppinglist":["yes"], "version_":1805858216516511872}, {"id":"11","name":["egg"], "category":["dairy"], "shoppinglist":["no"], "version_":18058582165175584}, {"id":"12","name":["cheese"], "category":["dairy"], "shoppinglist":["no"], "version_":18058582165186048} ]}
```

```

santanamonica@dscbigdata: ~/dsc650-infra/bellevue-bigdata/solr
solr@ea89058bde35: /opt/solr-9.6.18curl "http://localhost:8983/solr/myowncollection/select?q=category:vegetable"

{
  "responseHeader": {
    "status": 0,
    "QTime": 2,
    "params": {
      "q": "category:vegetable"
    }
  },
  "response": {
    "numFound": 2,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "id": "2",
        "name": ["carrot"],
        "category": ["vegetable"],
        "shoppinglist": ["no"],
        "_version_": 1805858216508129280
      },
      {
        "id": "5",
        "name": ["onion"],
        "category": ["vegetable"],
        "shoppinglist": ["no"],
        "_version_": 1805858216511275008
      }
    ]
  }
}

solr@ea89058bde35: /opt/solr-9.6.18curl "http://localhost:8983/solr/myowncollection/select?q=category:fruit"

{
  "responseHeader": {
    "status": 0,
    "QTime": 2,
    "params": {
      "q": "category:fruit"
    }
  },
  "response": {
    "numFound": 3,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "id": "1",
        "name": ["banana"],
        "category": ["fruit"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216499740672
      },
      {
        "id": "3",
        "name": ["watermelon"],
        "category": ["fruit"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216509177856
      },
      {
        "id": "10",
        "name": ["mango"],
        "category": ["fruit"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216517566464
      }
    ]
  }
}

```

## Queries in Solr Web Interface

First query is selecting all the items that are on the shopping list “yes” out of the 12

The screenshot shows the Solr Web Interface with a query executed. The left sidebar contains navigation links: Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, myowncollection (selected), Overview, Analysis, Documents, Parameters, Files, Ping, Plugins / Stats, Query (selected), Replication, Schema, and Segments info.

The main query area shows the following configuration:

- Common: q
- q: shoppinglist:yes
- q.op: OR
- fq:
- sort:
- start, rows: 0, 10
- df:
- params(s):
- wt: -----
- indent on: ☒
- debugQuery: ☐
- defType: -----
- hl: ☐
- facet: ☐
- spatial: ☐
- spellcheck: ☐
- Raw Query Parameters:
- JSON Query:
- Execute Query button

The response is displayed in JSON format on the right:

```

{
  "responseHeader": {
    "status": 0,
    "QTime": 3,
    "params": {
      "q": "shoppinglist:yes",
      "indent": "true",
      "q.op": "OR",
      "useParams": "",
      "_": "172281738989"
    }
  },
  "response": {
    "numFound": 7,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "id": "1",
        "name": ["banana"],
        "category": ["fruit"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216499740672
      },
      {
        "id": "3",
        "name": ["watermelon"],
        "category": ["fruit"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216509177856
      },
      {
        "id": "4",
        "name": ["chips"],
        "category": ["salty"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216510226432
      },
      {
        "id": "6",
        "name": ["cupsake"],
        "category": ["sweet"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216513225584
      },
      {
        "id": "7",
        "name": ["chocolate"],
        "category": ["sweet"],
        "shoppinglist": ["yes"],
        "_version_": 180585821651372180
      },
      {
        "id": "8",
        "name": ["sacaweed"],
        "category": ["salty"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216515468192
      },
      {
        "id": "10",
        "name": ["mango"],
        "category": ["fruit"],
        "shoppinglist": ["yes"],
        "_version_": 1805858216517566464
      }
    ]
  }
}

```

Second query is selecting all the items that are on the shopping list and in the salty category

The screenshot shows the Solr Admin interface at localhost:8983. The left sidebar contains navigation links: Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, myowncollection (selected), Overview, Analysis, Documents, Params, Files, Ping, Plugins / Stats, Query (selected), Replication, Schema, and Segments info. The main panel is titled 'Request-Handler (qt)' and shows the following configuration: q: /select, category: salty, q.op: OR, fq: shoppinglist:yes, sort: (empty), start: 0, rows: 10, df: (empty), paramset(s): (empty), wt: (empty), indent on: checked, debugQuery: unchecked, defType: (empty), hl: unchecked, facet: unchecked, spatial: unchecked, spellcheck: unchecked, Raw Query Parameters: (empty), and JSON Query: (empty). The 'Execute Query' button is at the bottom. The right pane displays the JSON response for the query: http://localhost:8983/solr/myowncollection/select?fq=shoppinglist%3Ayes&indent=true&q.op=OR&q.category%3Asalty&useParams=. The response is a JSON object with status, time, params, and response fields. The response contains one document with id '1', name 'cheese', category 'dairy', and shoppinglist 'no'.

Third query is selecting all the items that are in the dairy category and descending the order

The screenshot shows the Solr Admin interface at localhost:8983. The left sidebar is the same as the previous screenshot. The main panel is titled 'Request-Handler (qt)' and shows the following configuration: q: /select, category: dairy, q.op: OR, fq: (empty), sort: id desc, start: 0, rows: 10, df: (empty), paramset(s): (empty), wt: (empty), indent on: checked, debugQuery: unchecked, defType: (empty), hl: unchecked, facet: unchecked, spatial: unchecked, spellcheck: unchecked, Raw Query Parameters: (empty), and JSON Query: (empty). The 'Execute Query' button is at the bottom. The right pane displays the JSON response for the query: http://localhost:8983/solr/myowncollection/select?indent=true&q.op=OR&q.category%3Adairy&sort=id%20desc&useParams=. The response is a JSON object with status, time, params, and response fields. The response contains one document with id '1', name 'cheese', category 'dairy', and shoppinglist 'no'.