1. Download the [Amazon fine foods dataset.](https://snap.stanford.edu/data/web-FineFoods.html)
2. Design a Solr schema for the dataset.
3. Write a script to import the data using one of the DataStax CQL drivers.
4. Produce a sample query to find all reviews with “cat.\*” in either the summary or text.
   1. [http://localhost:8983/solr/bootcamp.review/select?q=summary%3Acat\*+OR+text%3Acat\*&wt=xml&indent=true](http://localhost:8983/solr/bootcamp.review/select?q=summary%3Acat*+OR+text%3Acat*&wt=xml&indent=true)
5. Produce a sample query which highlights the searched term.
   1. <http://localhost:8983/solr/bootcamp.review/select?q=text%3Aoatmeal&wt=xml&indent=true&hl=true&hl.fl=text&hl.simple.pre=%3Cem%3E&hl.simple.post=%3C%2Fem%3E>
6. Produce a sample query which finds the most prolific reviewer.
   1. [http://localhost:8983/solr/bootcamp.review/select?q=\*%3A\*&wt=xml&indent=true&hl=true&hl.fl=text&hl.simple.pre=%3Cem%3E&hl.simple.post=%3C%2Fem%3E&facet=true&facet.query=user\_id%3A\*&facet.field=user\_id](http://localhost:8983/solr/bootcamp.review/select?q=*%3A*&wt=xml&indent=true&hl=true&hl.fl=text&hl.simple.pre=%3Cem%3E&hl.simple.post=%3C%2Fem%3E&facet=true&facet.query=user_id%3A*&facet.field=user_id)
7. Produce a sample query that finds all reviews for an item, and a distribution of review scores.
   1. [http://localhost:8983/solr/bootcamp.review/select?q=product\_id%3AB001EO5QW8&wt=xml&indent=true&facet=true&facet.query=score%3A\*&facet.field=score](http://localhost:8983/solr/bootcamp.review/select?q=product_id%3AB001EO5QW8&wt=xml&indent=true&facet=true&facet.query=score%3A*&facet.field=score)
8. Configure the MoreLikeThis component to provide “similar items” recommendations.
   1. [http://localhost:8983/solr/bootcamp.review/select?q=\*%3A\*&fq=unique\_id%3AA2GHZ2UTV2B0CD-B001EO5QW8&mlt=true&mlt.fl=text&wt=xml&indent=true](http://localhost:8983/solr/bootcamp.review/select?q=*%3A*&fq=unique_id%3AA2GHZ2UTV2B0CD-B001EO5QW8&mlt=true&mlt.fl=text&wt=xml&indent=true)
9. Configure the SpellCheck component to provide typeahead for user profile names.
10. Produce a hive query to find the most/least highly rated item.
    1. SELECT product\_id, avg(score) AS score FROM bootcamp.review GROUP BY product\_id SORT BY score DESC LIMIT 10;
    2. SELECT product\_id, avg(score) AS score FROM bootcamp.review GROUP BY product\_id SORT BY score ASC LIMIT 10;
11. Produce a hive query to find the the most/least helpful reviewer.
    1. SELECT user\_id, (SUM(helpful) / SUM(out\_of)) \* 10 AS Helpful FROM bootcamp.review WHERE Helpful > 0 GROUP BY user\_id SORT BY Helpful DESC LIMIT 10;
    2. SELECT user\_id, (SUM(helpful) / SUM(out\_of)) \* 10 AS Helpful FROM bootcamp.review WHERE Helpful > 0 GROUP BY user\_id SORY BY Helpful ASC LIMIT 10;
12. Produce a word cloud using review for all items.
13. Produce a word cloud using review for a specific item.
14. What can be done to increase the quality of the word clouds and MoreLikeThis results?
    1. conceptual, solrconfig.xml enhancements, process to increase solr results.
15. **Bonus:** Use Mahout to generate recommendations for a user by clustering users by their review history.