**Lab 2**

**Youtube Video Link:** [**https://youtu.be/rtNj2qeklD4**](https://youtu.be/rtNj2qeklD4)

[**https://youtu.be/s5rFWDHS0c0**](https://youtu.be/s5rFWDHS0c0)

**Team Mates: Dave**

**Vinay Maturi**

**HIVE**

**Objectives**

Process Seinfeld episode data into a HIVE table, utilizing a complex data type, and perform queries on the data.

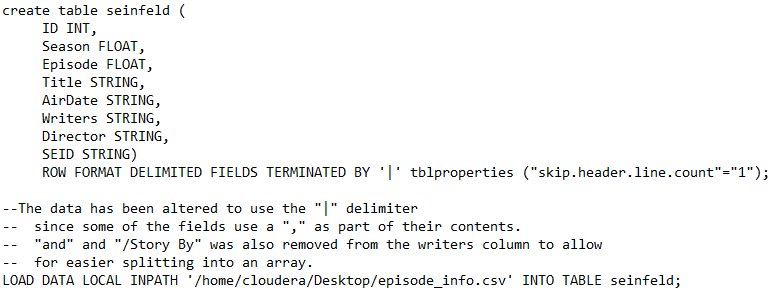
**Workflow**

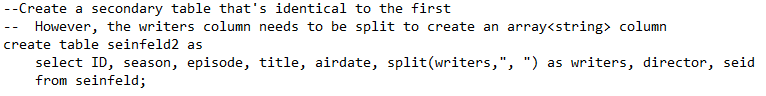
Responsibility for this section was split, with each partner designing 5 queries for the data. The partners imported data differently, both methods are demonstrated.

**Data import**

**Dave:**

The Seinfeld dataset contains fields for unique ID, season number, episode number (of that season), the episode title, the original date the episode aired, the writers of the episode, the director, and the season/episode identifier. Both the air date and the writers fields contain commas as a part of their contents. This causes an import problem since the fields are delimited by commas. To resolve this issue, all commas were replaced with pipes, and those pipes occurring with a space afterwards were returned to being commas. Similarly, all "and"s and "\ Story By "s were replaced with commas. The lone non-breaking space character was also removed.

To get the data into hive, first a table was created to store the base information. Then the data was loaded in from the csv. 

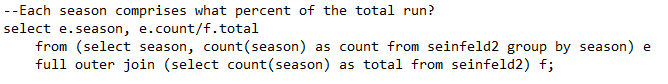
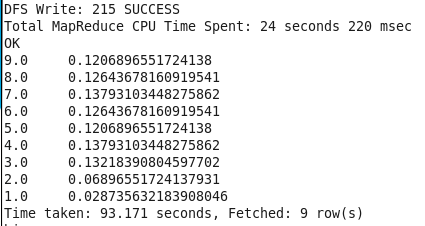
This table was then duplicated using a select statement to split the writers string into an Array. 

**Vinay:**

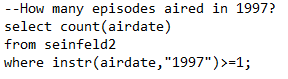
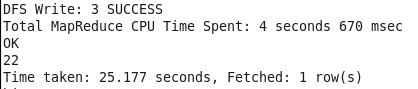
**Querries:**

**Dave:**

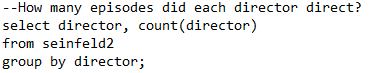
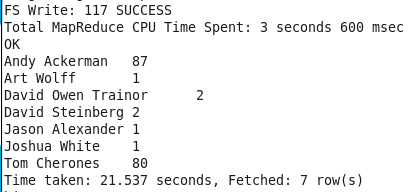
What percentage of the total run of the show does each season account for?

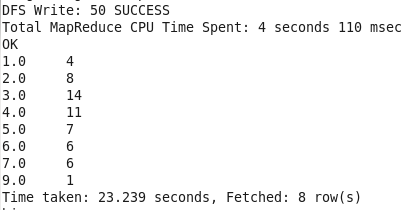
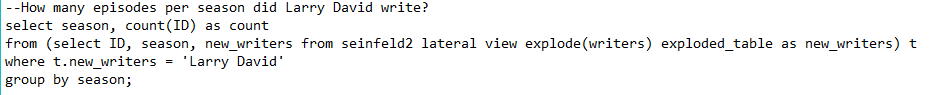
How many episodes aired in 1997?

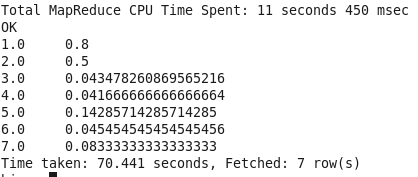
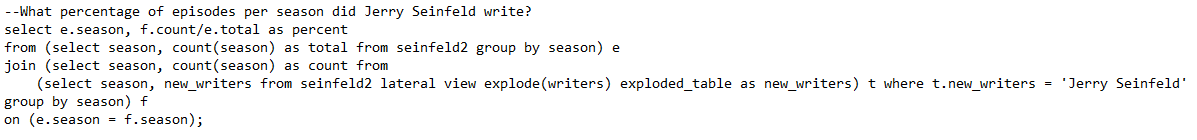
How many episodes did each director direct?

How many episodes per season were written, at least in part, by Larry David?

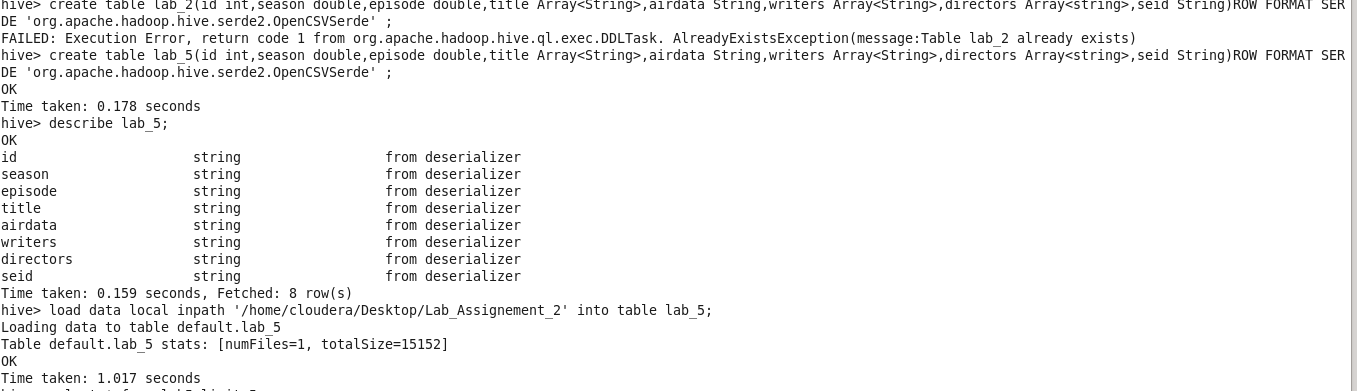


What percent of episodes per season were written, at least in part, by Jerry Seinfeld?



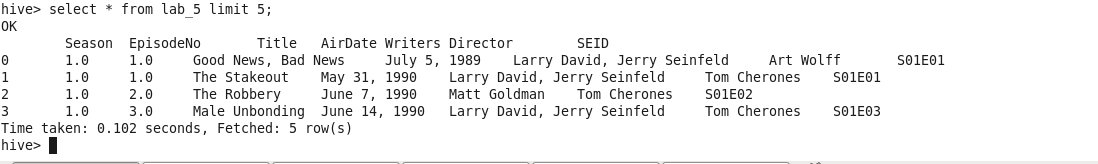
**Vinay:**

A table is created in the following way.



The queries are executed on the hive table as follows.

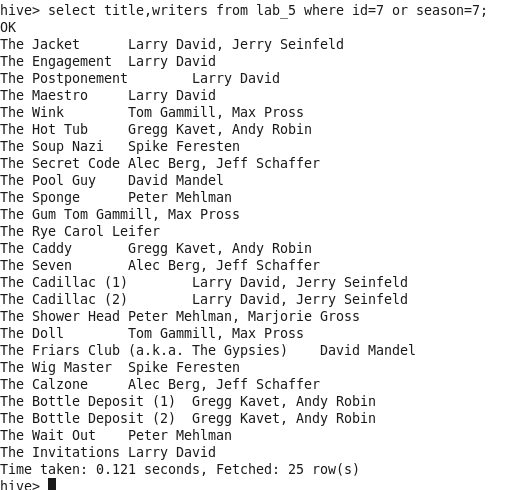
**Query 1:**

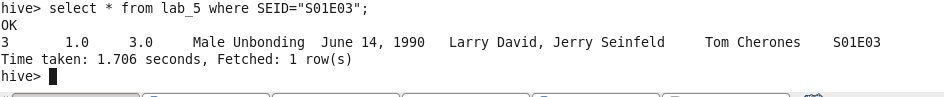


**Query\_2:**

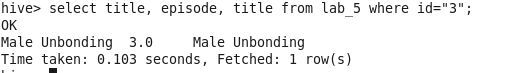


Query 3:



**Query 4:**

**Query 5:**



**Solr**

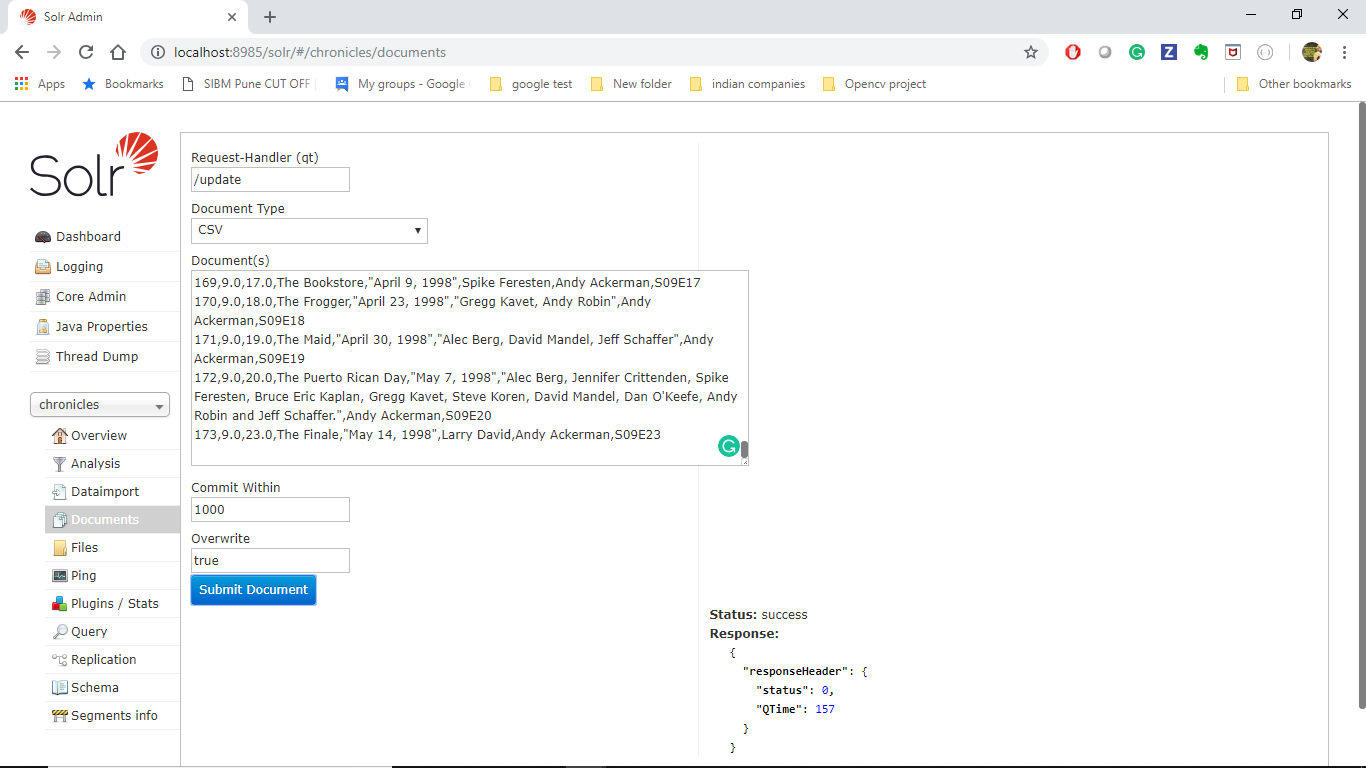
**Objectives:**

TO execute the search on the selected datasets with different queries.

**Workflow:**

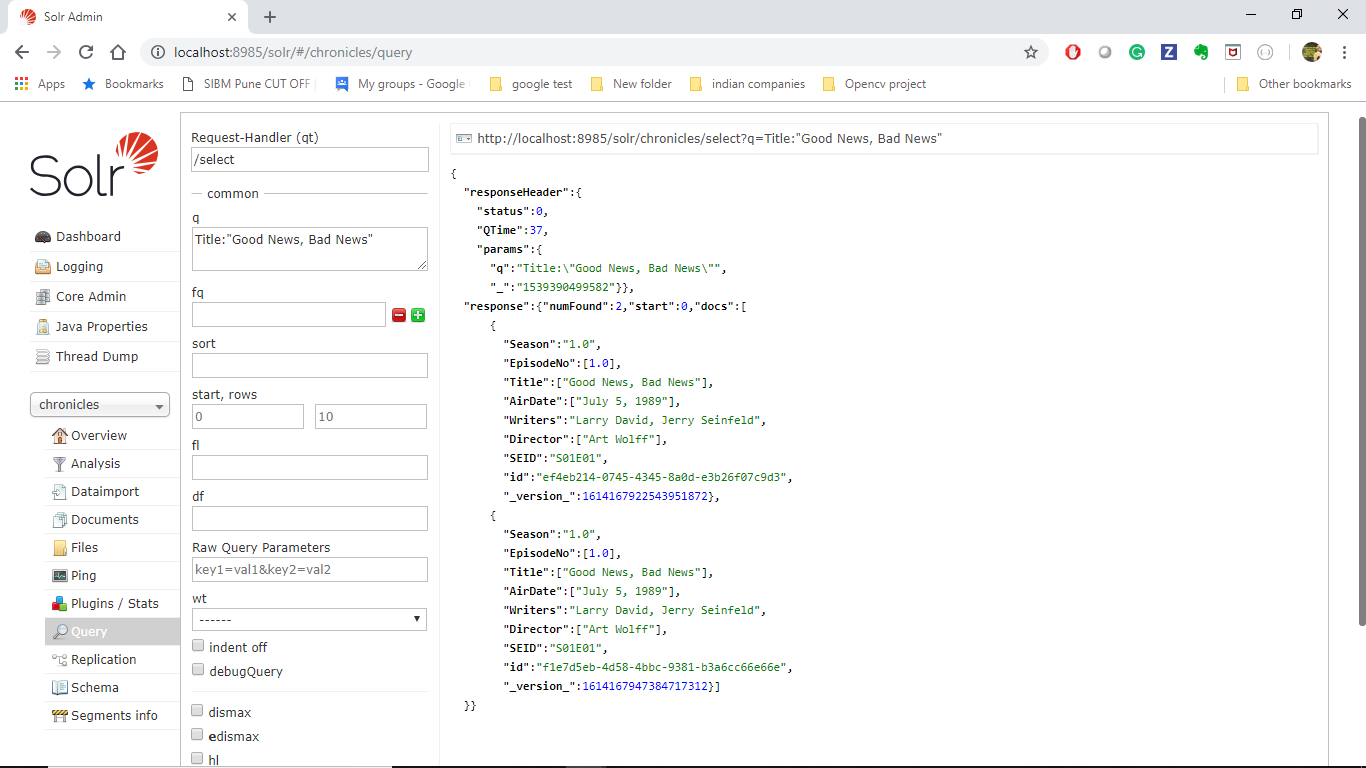
Solr processing was completed by Vinay maturi

**Loading the Dataset and Checking the success**

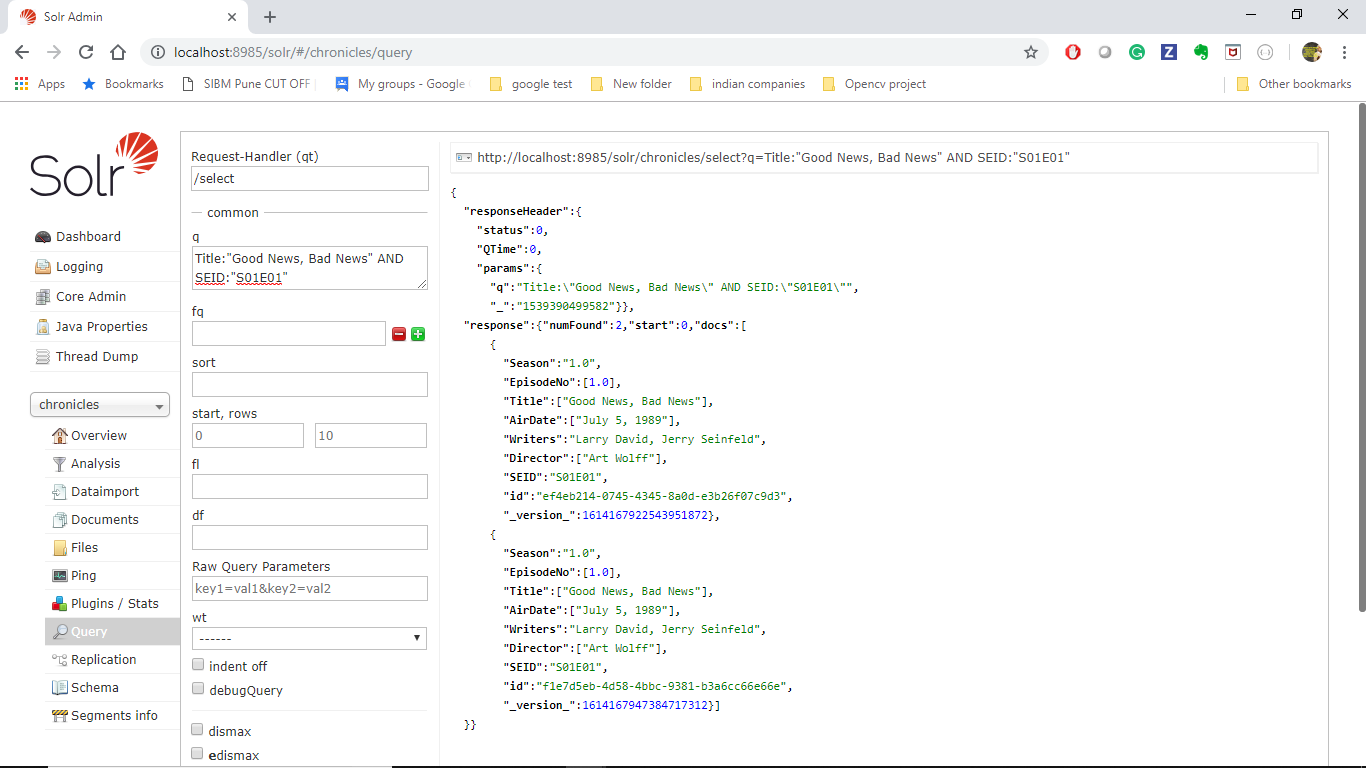
****

**Query 1:**

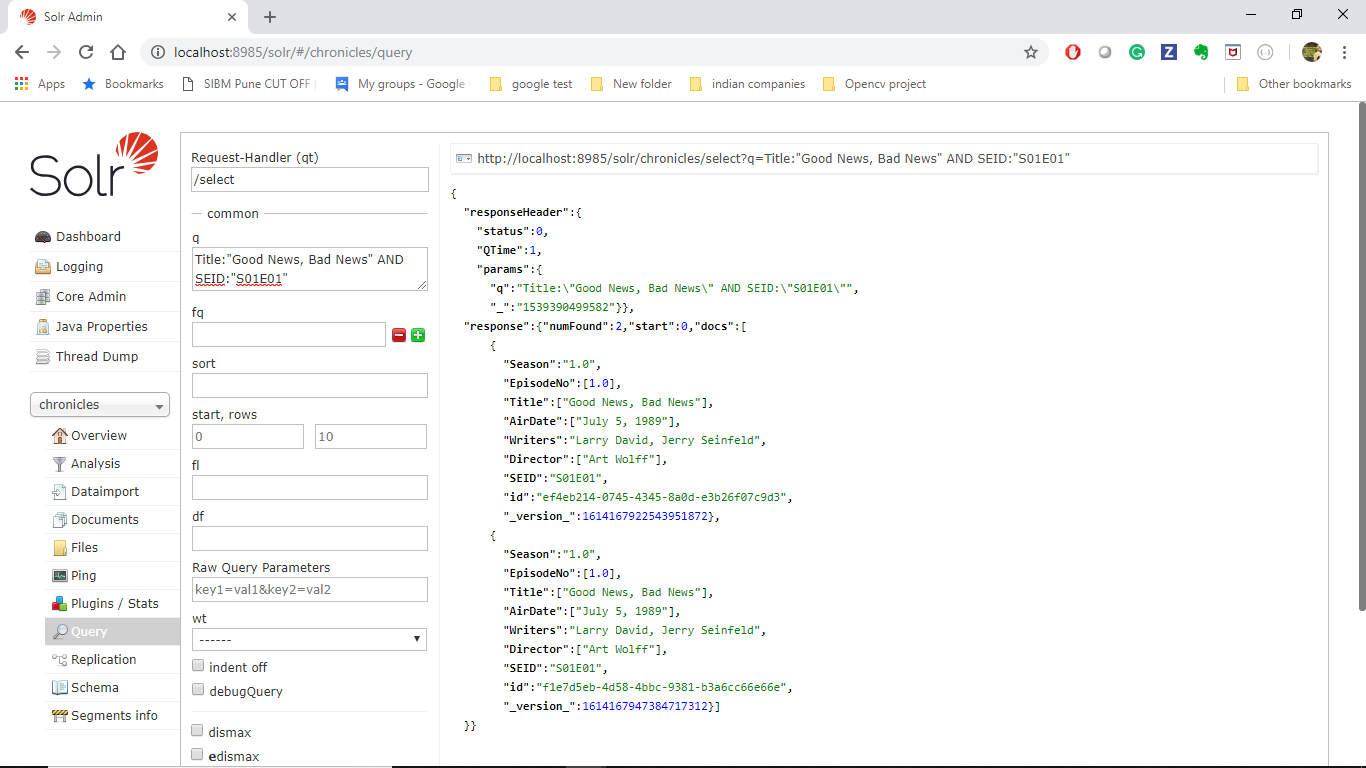
Display the info about the whole info whose title name is “Good News , Bad news”



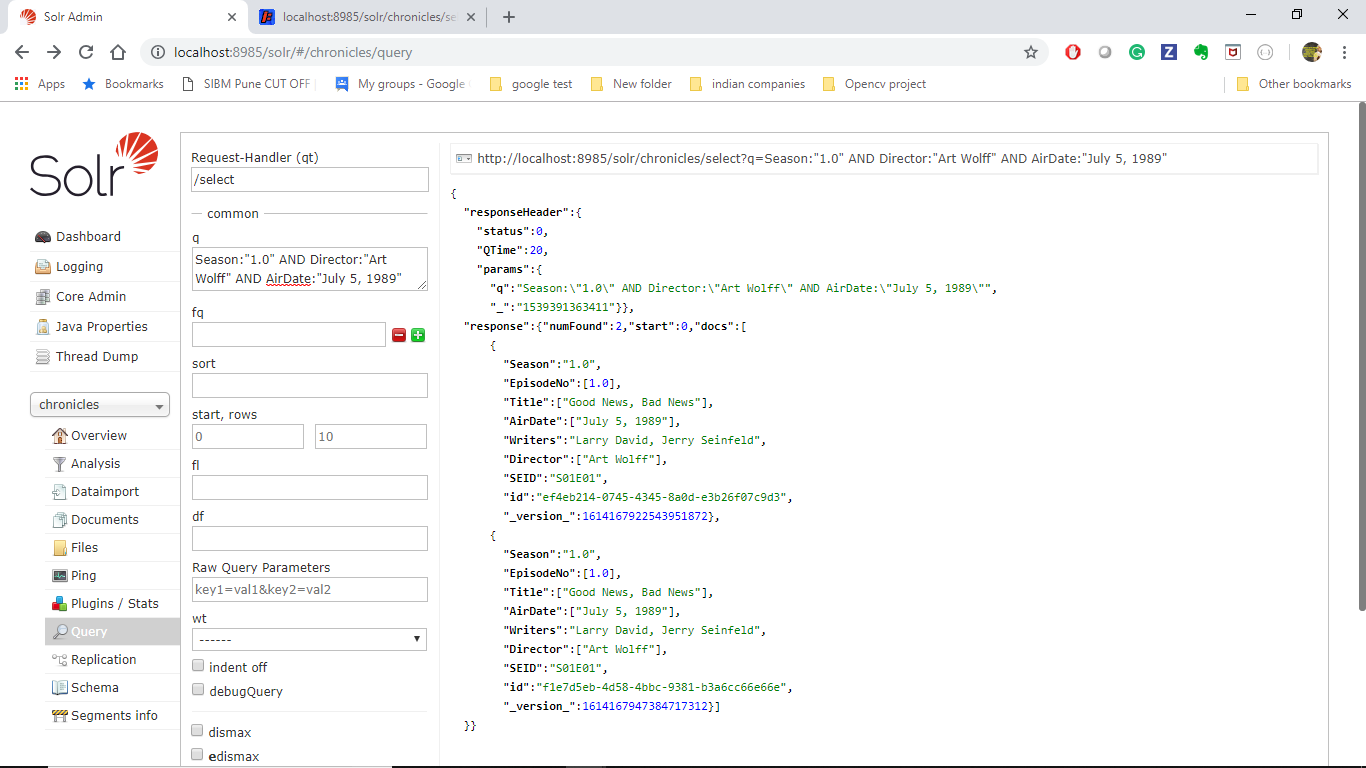
Query\_2



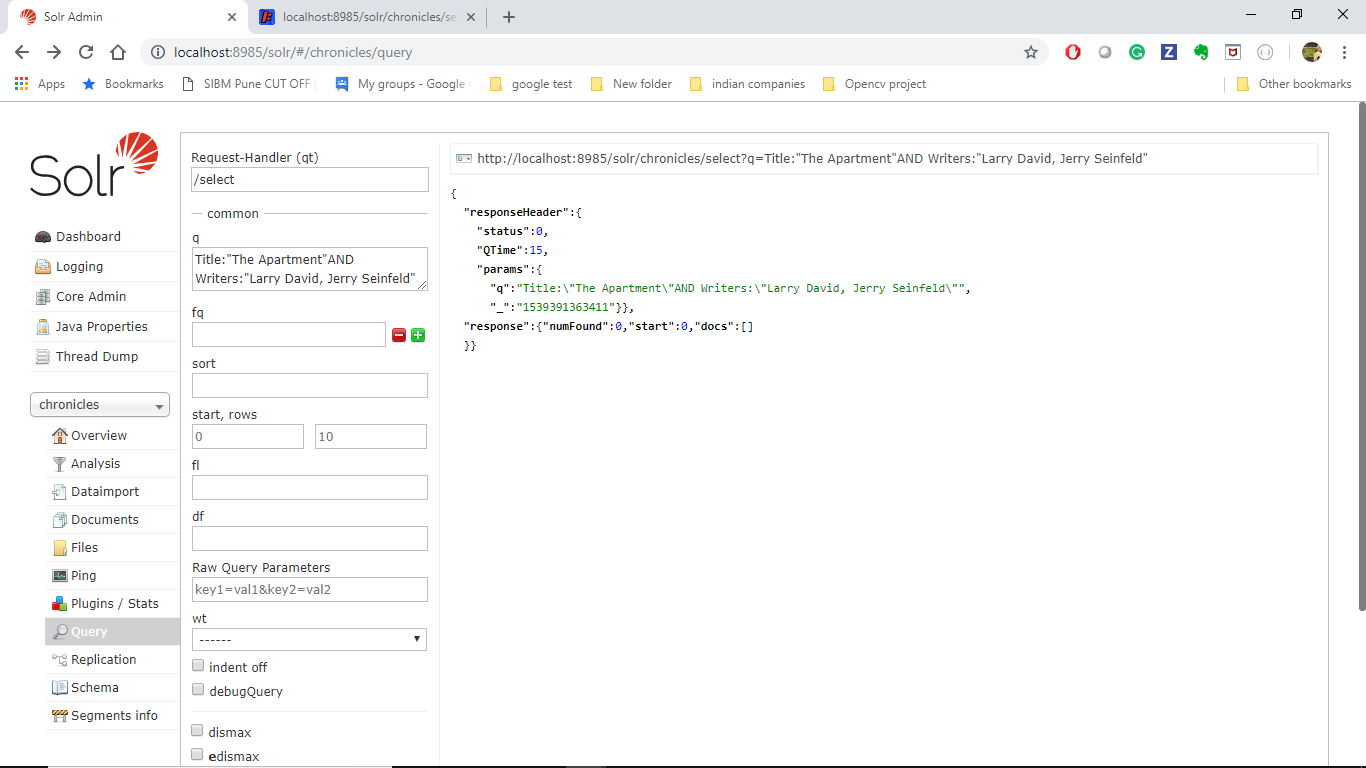
Query\_3



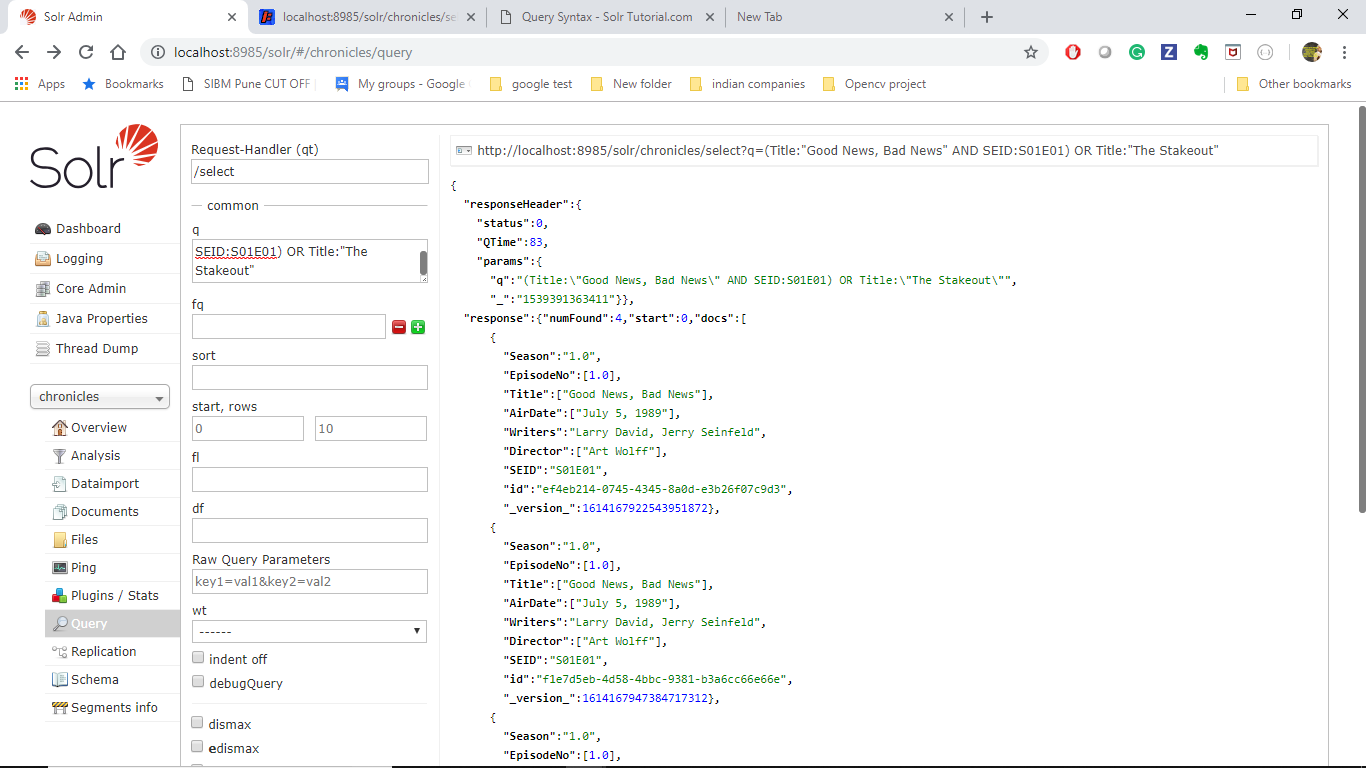
Query\_4



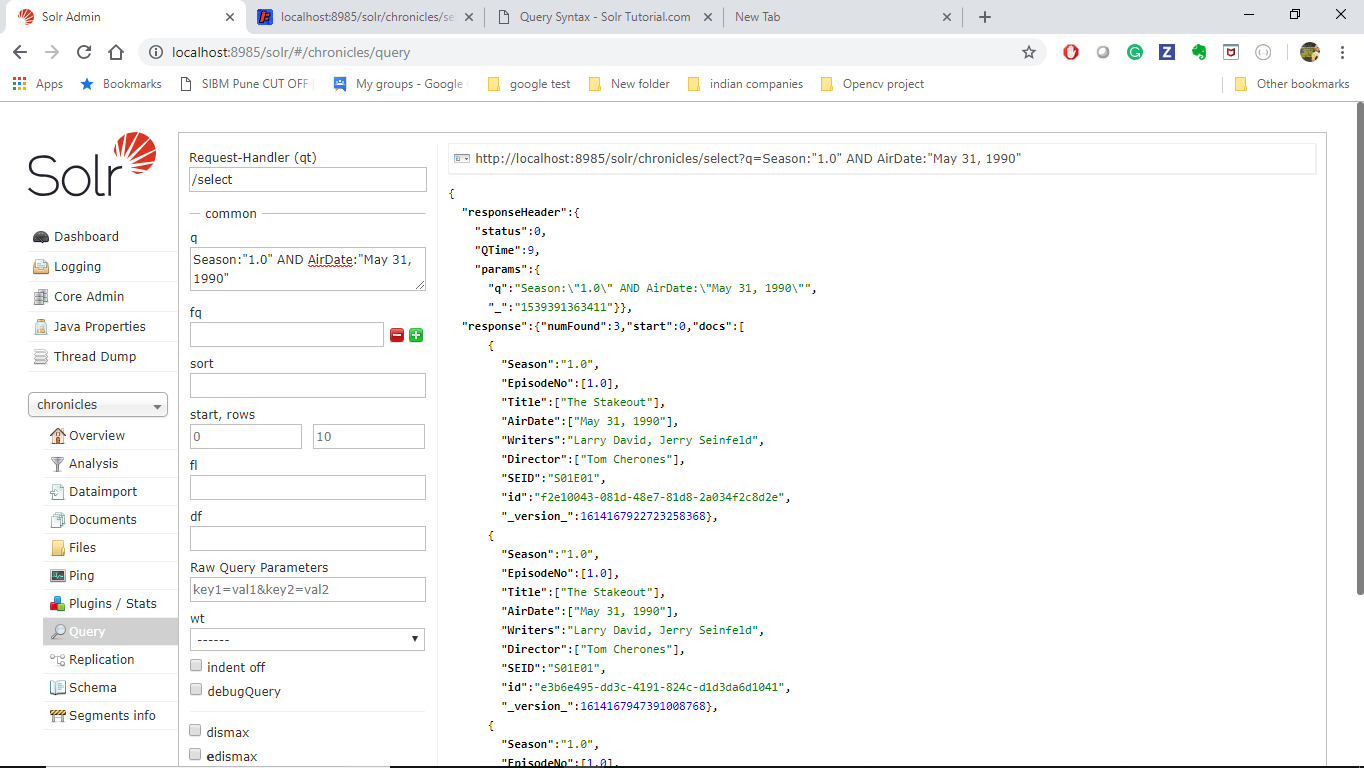
Query\_5



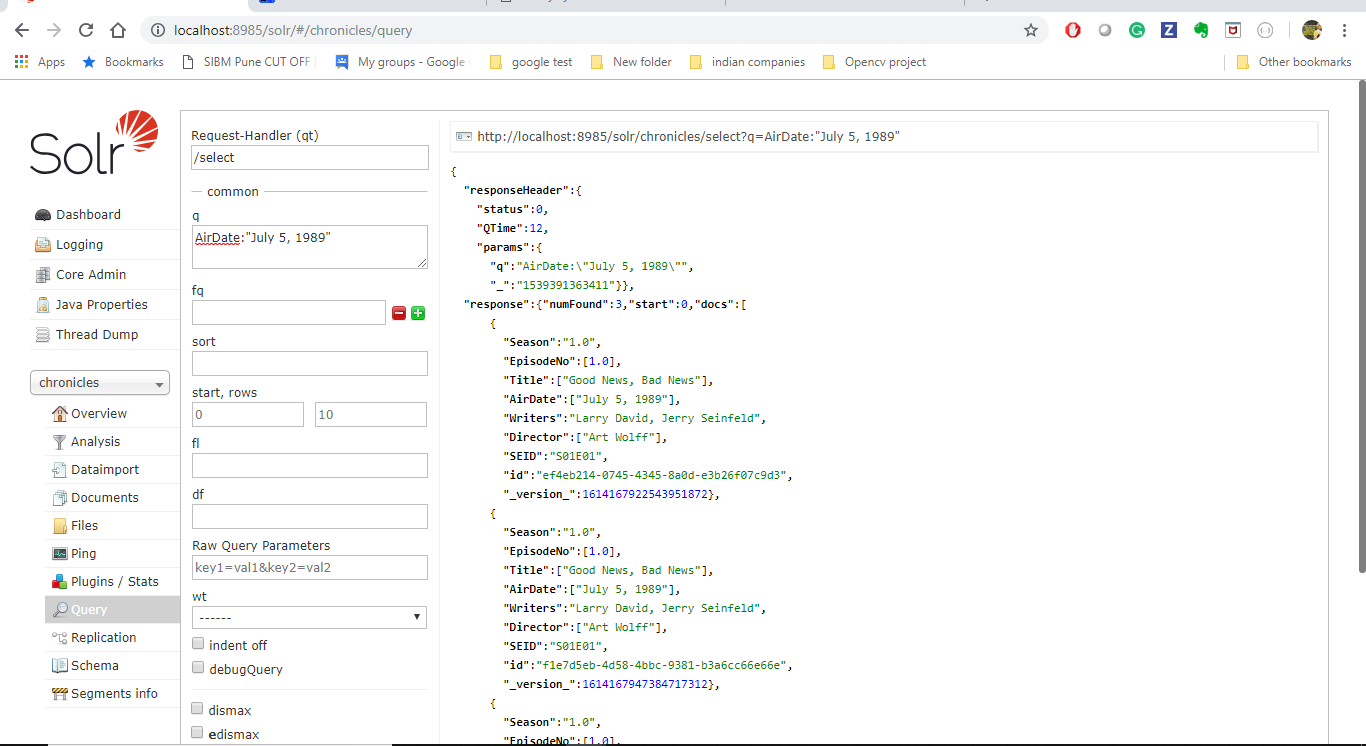
Query\_6



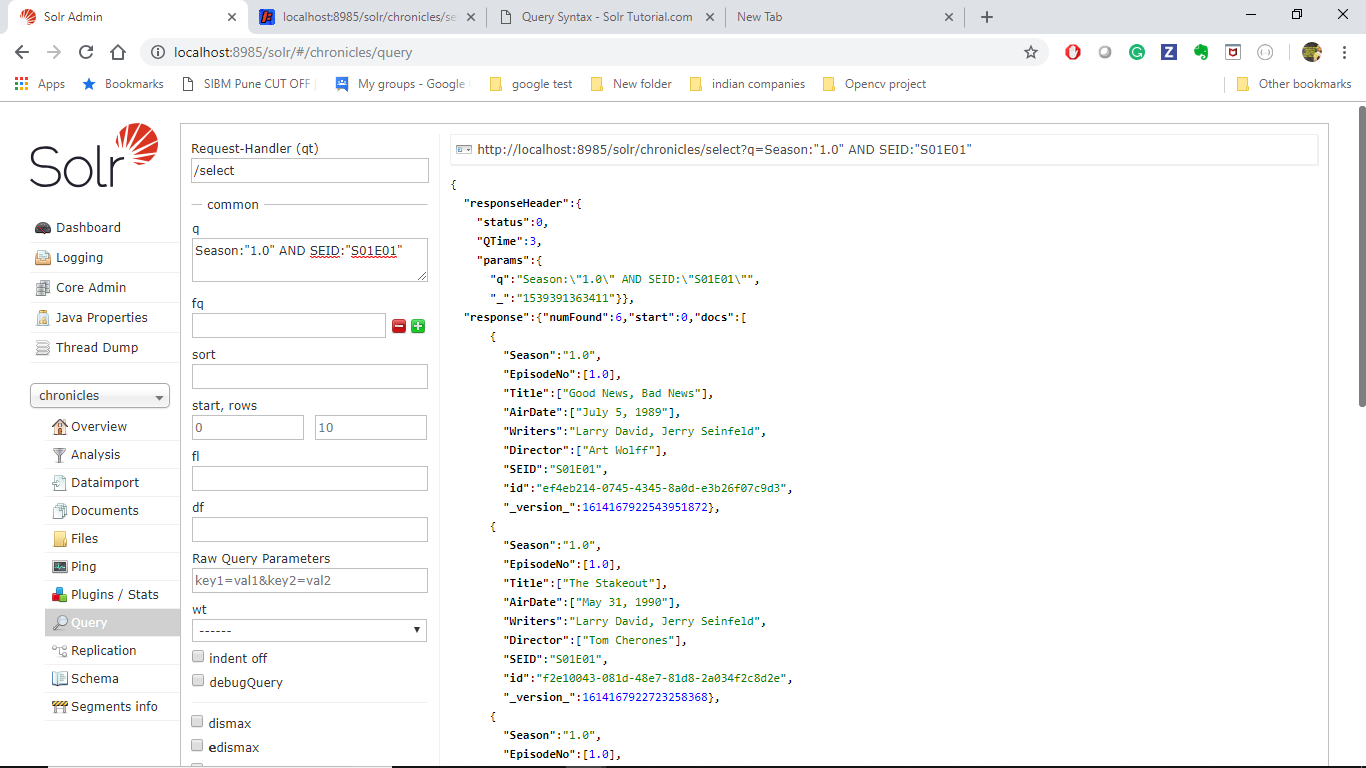
Query\_7



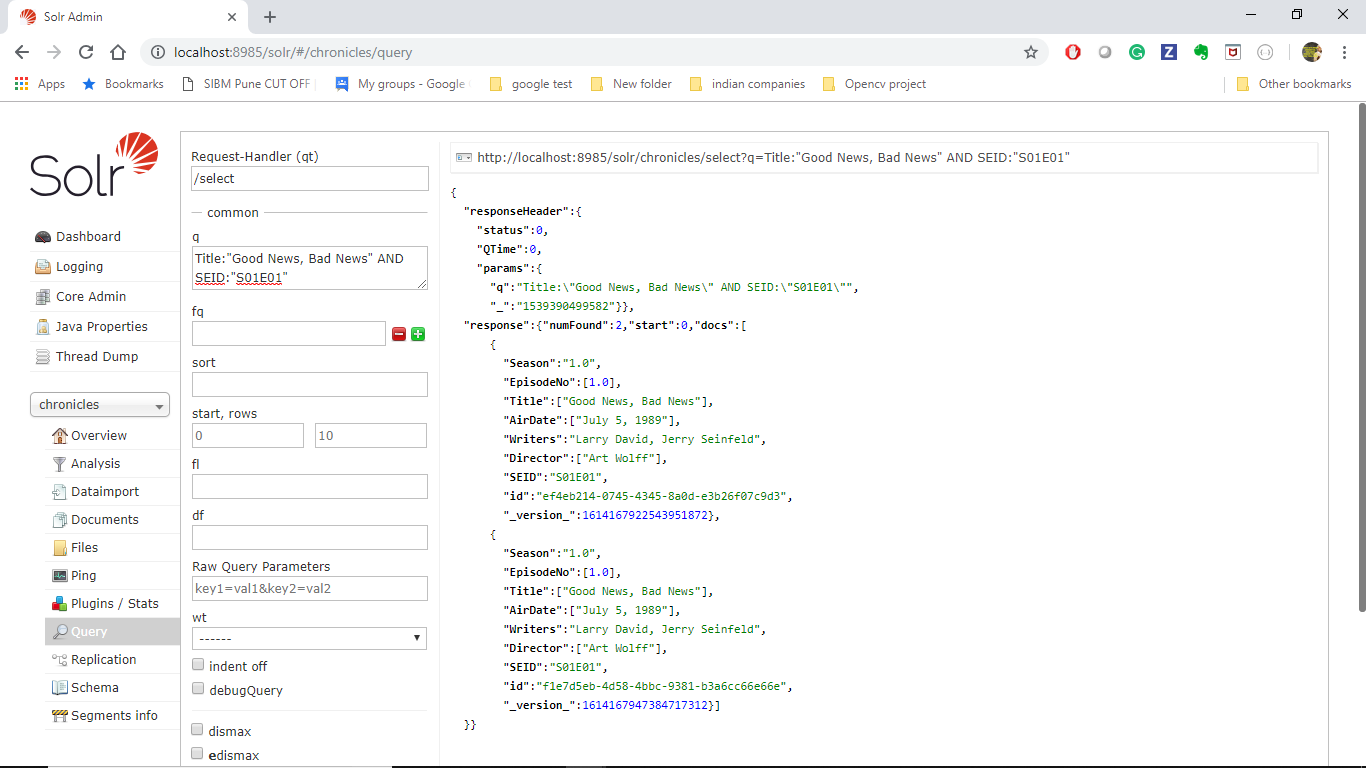
Query\_8:



Query\_9:



Query\_10:



**Cassandra**

**Objectives:**

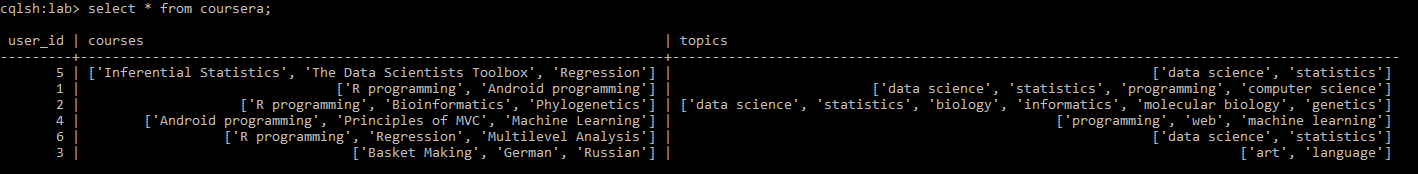
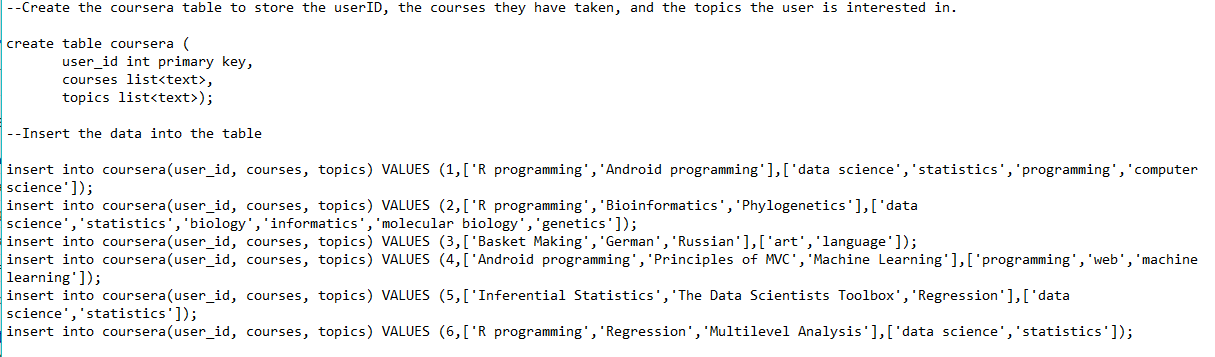
Consider one the the provided use cases, develop a small data set, and perform a few queries on the data that address the use case.

**Workflow:**

Cassandra processing was completed by Dave Walsh.

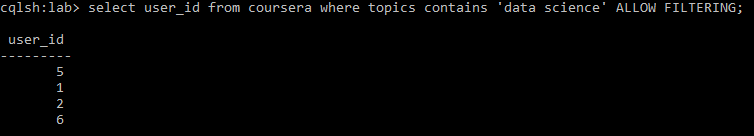
**Dataset:**

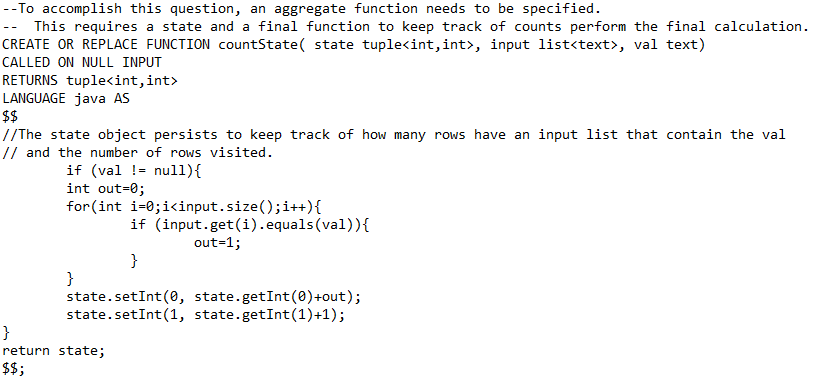
A small data set was constructed consisting of user ID's, a list of classes taken or enrolled in, and a list of topics of interest selected by the user.

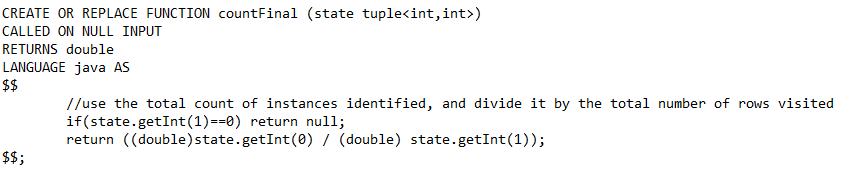


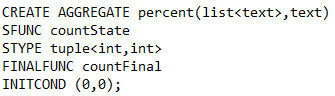
**Use case:**

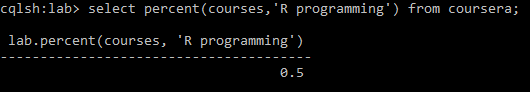
*A new course is being offered that is tagged by the interest 'data science', a list of users needs to be identified to that they can be notified of the new course availability.*

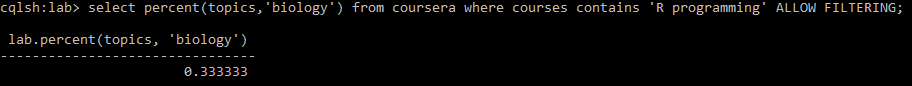
https://github.com/Viral1101/big_data/raw/master/Lab%202/Documentation/cass-c1.PNG

*How successfully is a particular course reaching students? This could play a role in deciding whether to continue offering the course or not.* In order to address this scenario, an aggregate function had to be created. This comprises three separate functions: the state function, the final function, and the aggregate itself. The state function in this example looks through a list to identify a given text within that list. If that text is identified, the first field of the state tuple is incremented. The second field of the state tuple is always incremented to keep track of the total fields. 

The final function takes the state tuple and performs the final arithmetic, dividing the first field by the second to obtain a percent. 

The aggregate portion pieces the first two functions together and sets the initial state of the tuple to (0,0). 



*What percentage of students interested in Biology have taken the R programming course? Perhaps other topics should be offered for them to like.* https://github.com/Viral1101/big_data/raw/master/Lab%202/Documentation/cass-c3.PNG

**Conclusion**

The query languages worked with in this exercise provide a familiar environment to those exposed to SQL previously. The freedom of Cassandra, for example, to not be limited to a relational style database through the use of column families provides an interesting platform where the schema doesn't need to be updated if more information needs to be added. The advantage of these technologies are that they can exist in a cluster environment and they are open source. This provides monetary advantages in that if more power is required, a node simply needs to be added, and there are no licensing requirements to use the software.

Cassandra provided an interesting challenge in that it has a fairly limited selection of pre-defined functions. This does provide the flexibility of customized programming, but also requires a bit of a learning curve to initially get the hang of.