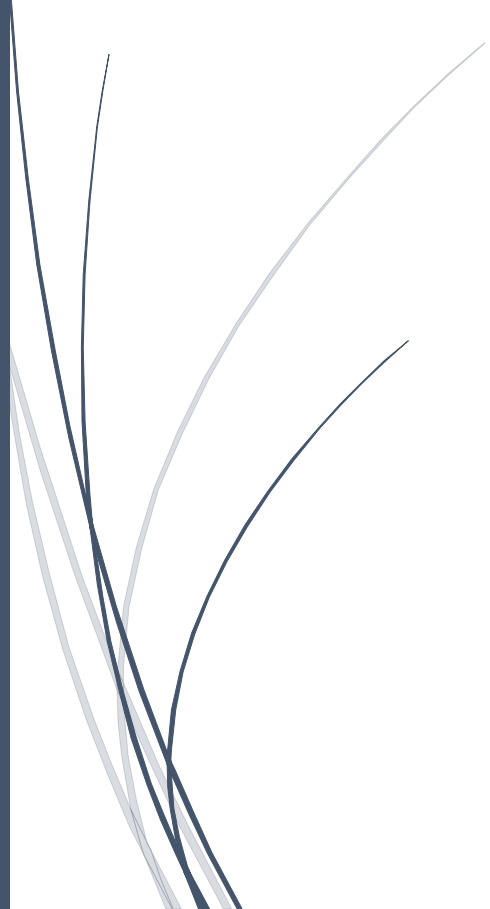


A thick dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

12/7/2017

Yelp Review Analytics

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

**Subhasree Murugan
Priyanka Shukla
Supreet Kaur
Ruhani Chawla
Priyanka Chandore**

YELP!!!

About

Yelp, a publicly traded company (**YELP**) was founded by Jeremy Stoppelman and Russel Simmons in 2004. Yelp emerged from San Francisco incubator MRL Ventures with a name that contracted the phrase “Yellow Pages” into four letters. Initially an email recommendation service, it launched in its existing form in October 2004, catering to the San Francisco market. Now, the company has its headquarters in San Francisco, California and has is now available in 219 cities.

The AHA Moment!

Back in 2004, Stoppelman searched Google for a new doctor and realized there was no way to determine if “doctor A” was better than “doctor B.” Traditionally, when you have a problem, you ask a trustworthy friend for advice. Although this process may sound easy, it’s not. People are busy and want to make their own decisions quickly. With the growing usage of the internet, Stoppelmen knew he couldn’t be the only person struggling with this common task. Yelp’s first product didn’t include social networking, it was just an email service where friends could exchange local businesses. Simply put, it’s word of mouth – amplified.

What’s in the name?

One of its founder, Stoppelman initially wanted to call it “Yocal” as in local yocal, however he couldn’t obtain the domain name. Later, an early employee, suggested the name “Yelp”. Stoopleman’s first reaction was it sounded like a cry for help. After that he realized, he liked that it was short, memorable and how it tied into what Yelp was all about: helping consumers make better choices. Being crunched for time in the decision, Yelp was born.

What Yelp is all about?

Yelp aims to help consumers connect with local businesses. The company’s rating system (5-star reviews) and filtering options make it easy for consumers to find the perfect place to eat or visit. Yelp focused on building a network of reviewers with profiles, friends, and accolades. Profiles gave anonymous reviews a name and a face, making them more trustworthy. This was key to Yelp’s growth, because users are more likely to trust reviews from real people than anonymous internet strangers. People love the chance to share their opinions with others.

Yelp Review Analytics

Goals

The objective is to propose new webpage design for businesses and reviewers on Yelp which will use existing yelp data to provide insightful analytics and help existing users and business owners, to make important decisions regarding businesses. The new design that we are proposing will be driven by analytics metrics.

Problem Statement

Yelp is a website which publishes crowd sourced reviews about local businesses (Restaurants, Department Stores, Bars, Home-Local Services, Cafes, Automotive, etc.). It provides opportunity to users to choose best business amongst available and business owners to improve their services. Although Yelp has covered major aspects of being a successful “review-driven” website, currently Yelp doesn’t have some important metrics that could help a user to carve out important information out of such huge number of reviews. Also, it does not provide advanced analytics to business owners to grow their business and improve their services. Our solution will focus on providing advance analytics from yelp data to help existing users and business owners.

Proposed Solution Statement

Here are the main points we are including in our solution:

- We aim on improving a user’s experience on Yelp by providing them a way of surfing through the reviews that matter to them.
- We want to provide some enhancements to the existing website by adding certain features that can be helpful to a user.
- We want to enable existing business owners to improve their services using analytics so that they can make important decisions regarding business expansion in new cities, countries.
- We want to enable users to figure out few interesting trends and most happening places using yelp data.

Dataset Source

Yelp dataset is available at: https://www.yelp.com/dataset_challenge

INTRODUCTION

Data Preparation

We have downloaded the Yelp dataset and have taken 5 JSON files namely Business, Review, User, Checkin and Tip for our analysis. We are then taking these JSON files and have explored each one of them to determine the required fields for our solution.

Since the business, review and user collections were too large, we have restricted our analysis to two businesses and have presented our solutions based on these selections. To reduce the dataset, we have followed following chain steps:

1. Selecting two cities out of business collection.

```
db.business.find({city:{$in:["Las Vegas","Madison"]}}).forEach( function(x) {  
db.sample_business.insert(x) } );
```

```
MongoDB Enterprise >  
MongoDB Enterprise >  
MongoDB Enterprise > db.business.find({city:{$in:["Las Vegas","Madison"]}}).forEach( function(x) { db.sample_business.insert(x) } );
```

Explanation:

Here we have taken businesses for 2 cities that are Las Vegas and Madison and created a smaller dataset which we will use in further steps.

2. Creating the reduced review collection.

```
db.review.find({business_id:{$in:db.sample_business.distinct("business_id")}}).forEach(  
function(x){db.sample_review.insert(x) } );
```

```
MongoDB Enterprise >  
MongoDB Enterprise > db.review.find({business_id:{$in:db.sample_business.distinct("business_id")}}).forEach( function(x){db.sample_review.insert(x) } );
```

Explanation:

By using the sample business collection, we have created a sample review collection which will contain the reviews of businesses in Las Vegas and Madison.

Note: We have used the original collections of Checkin, Tip and User to run our queries.

DESIGN

We have used two databases which are MongoDB for running JSON queries and Neo4j for running graph queries for our project. We will now elicit the design that has been used in our project.

MongoDB

MongoDB pseudo schema:

➤ **Business:**

```
{
  "_id" : "5a20c068ac81a9068c6eeb87",
  "bus_id" : "YDf95gJZaq05wvo7hTQbbQ",
  "name" : "Richmond Town Square",
  "address" : "691 Richmond Rd",
  "city" : "Richmond Heights",
  "state" : "OH",
  "postal code" : 44143,
  "latitude" : "41.5417162",
  "longitude" : "-81.4931165",
  "stars" : 2.0,
  "review_count" : 17,
  "attributes" : [{ "DogsAllowed" : "No"
                  "Caters" : "Yes"
                  "HasTV" : "Yes"
                  "WiFi" : "No"
                }],
  "categories" : [{ 0: "Restaurants"
                   1: "Greek"
                }],
  "neighborhood": "sunset",
  "is_open" : 1,
  hours : [{ "Monday" : "12:00-21:00"
            "Tuesday" : "12:00-21:00"
            "Wednesday" : "12:00-21:00"
            "Thursday" : "12:00-21:00"
            "Friday" : "12:00-21:00"
            "Saturday" : "12:00-21:00"
            }]
}
```

➤ Review:

```
{
  "_id" : "5a20c068ac81a9068c6effa4",
  "review_id" : "VfBHSwC5Vz_pbFluy07i9Q",
  "user_id" : "cjpgDjZyprfyDG3RlkVG3w",
  "bus_id" : "uYHaNptLzDLoV_JZ_MuzUA",
  "stars" : 5.0,
  "date" : "2016-07-12",
  "text" : "My girlfriend and I stayed here for 3 nights and loved it....I would highly
recommend this hotel          to friends, and when I return to Edinburgh (which I
most definitely will) I will be staying here without any hesitation.",
  "useful" : 0,
  "funny" : 0,
  "cool" : 0
}
```

User:

```
{
  "_id" : "5a20c07aac81a9068c7ad427",
  "user_id" : "lsSiIjAKVl-QRxKjRErBeg",
  "name" : "Cin",
  "review_count" : 272,
  "yelping_since" : "2010-07-13",
  "friends" : [{ 0 : "M19NwFwAXKRZzt8koF11hQ"
                  1 : "QRcMZ8pJJBBZaKubHOoMDQ"
                  2 : "uimsjcHoBnXz1MAKGvB26w"
                  3 : "vP5ajc1oGURsNvCXewsnDw"
                }],
  "useful" : 17019,
  "funny" : 16605,
  "cool" : 16856,
  "fans" : 209,
  "elite" : [{ 0 : 2014
                1 : 2016
              }],
  "average_stars" : 3.8
}
```

➤ Checkin:

```
{
  "_id" : "5a20c068ac81a9068c6eec22",
  "time" : [{ "Thursday" : "22:00" : 1
              "18:00" : 1
              "Wednesday" : "23:00" : 2
              "19:00" : 1
            }],
  "bus_id" : "7KPBkxAOEtb3QeIL9PEErg"
}
```

Tip:

```
{
  "_id" : "5a20c068ac81a9068c6f0e00",
  "user_id" : "zcTZk7OG8ovAmh_fenH21g" ,
  "text" : "Get here early enough to have dinner.",
  "date" : "2012-07-15",
  "bus_id" : "tJRDlI5yqpZwehenzE2cSg"
}
```

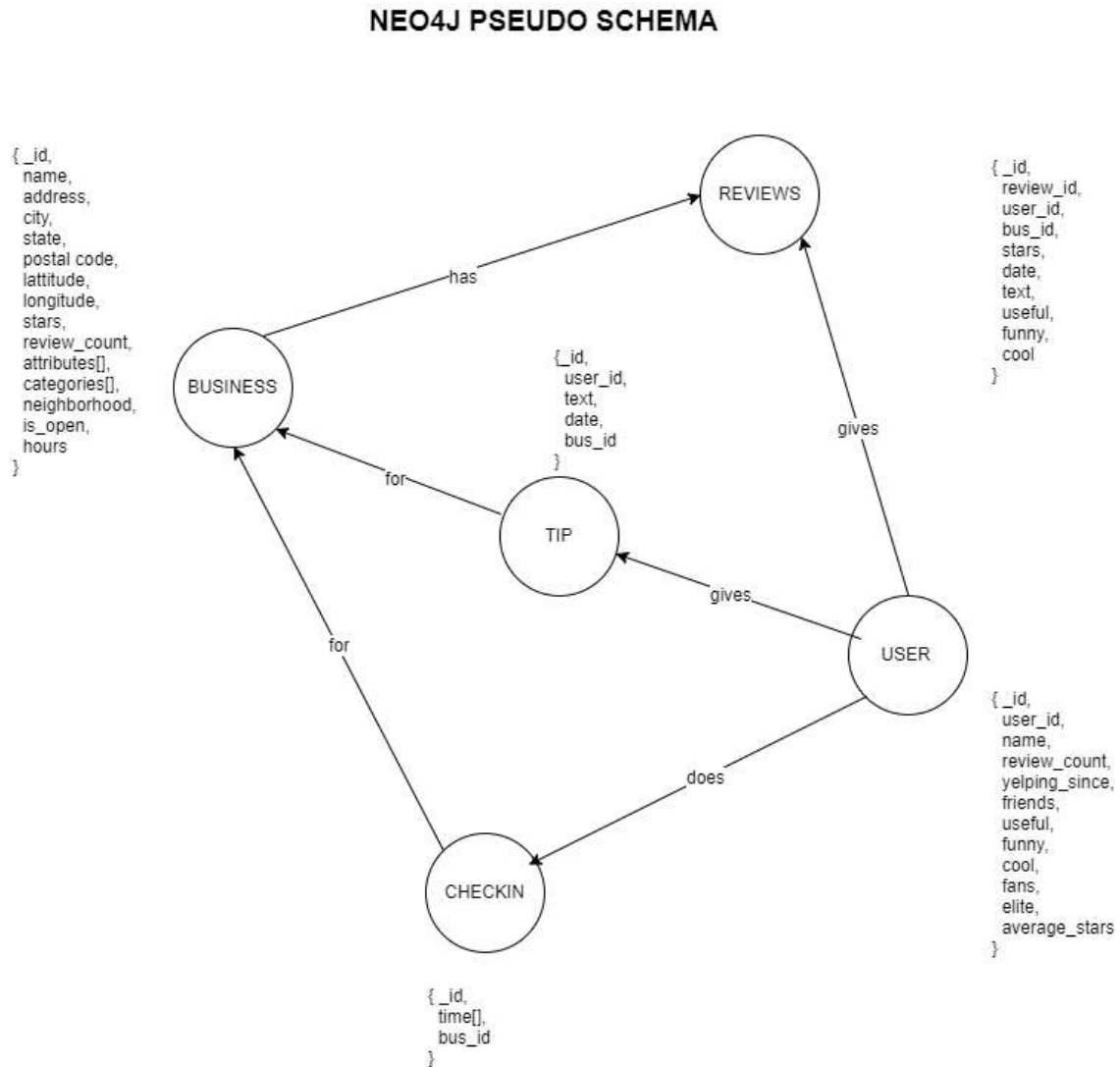
BUSINESS		
Attributes	Datatype	Constraint
bus_id	varchar(50)	PK
Name	varchar(50)	not null
Address	varchar(100)	not null
City	varchar(50)	not null
State	varchar(50)	not null
postal_code	number	not null
Latitude	double	
Longitude	double	
Stars	double	CHECK([stars] > 0)
review_count	number	not null
Attributes	varchar(50)	
categories	varchar(50)	
neighborhood	varchar(50)	
is_open	boolean	not null
Hours	varchar(50)	not null

REVIEW		
Attributes	Datatype	Constraint
review_id	varchar(100)	PK
user_id	varchar(100)	FK
bus_id	varchar(100)	FK
Stars	double	CHECK([stars] > 0)
Date	date	not null
Text	varchar(1000)	not null
Useful	number	
Funny	number	
Cool	number	

USER		
Attributes	Datatype	Constraint
user_id	varchar(100)	PK
Name	varchar(50)	not null
review_count	number	not null
yelping_since	date	not null
Friends	varchar(100)	not null
Useful	number	
Funny	number	
Cool	number	
Fans	number	
Elite	number	
average_stars	double	CHECK([average_stars] > 0)
CHECKIN		
Attributes	Datatype	Constraint
Time	varchar(50)	not null
bus_id	varchar(100)	FK

TIP		
Attributes	Datatype	Constraint
user_id	varchar(100)	FK
Text	varchar(300)	not null
Date	date	not null
bus_id	varchar(100)	FK

Neo4j pseudo schema:



Since for our Neo4j queries, we have used only 3 nodes-User, Review and business and 2 relationships- User gives Review and Business has review, we created only the required nodes and relationship.

🔧 Queries to create nodes:

- **User:**

load csv from "file:///user.csv" as u

```
create(:User{_id:u[0],user_id:u[21],name:u[18],review_count:u[19],
yelping_since:u[22],friends:u[16],useful:u[20],funny:u[17],cool:u[13],fans:u[15],
elite:u[14],average_stars:u[1]})
```

- **Review:**

```
load csv from "file:///review.csv" as sr
create(:Review{_id:sr[0],business_id:sr[1],review_id:sr[5],stars:sr[6],
text:sr[7],useful:sr[8],user_id:sr[9]})
```

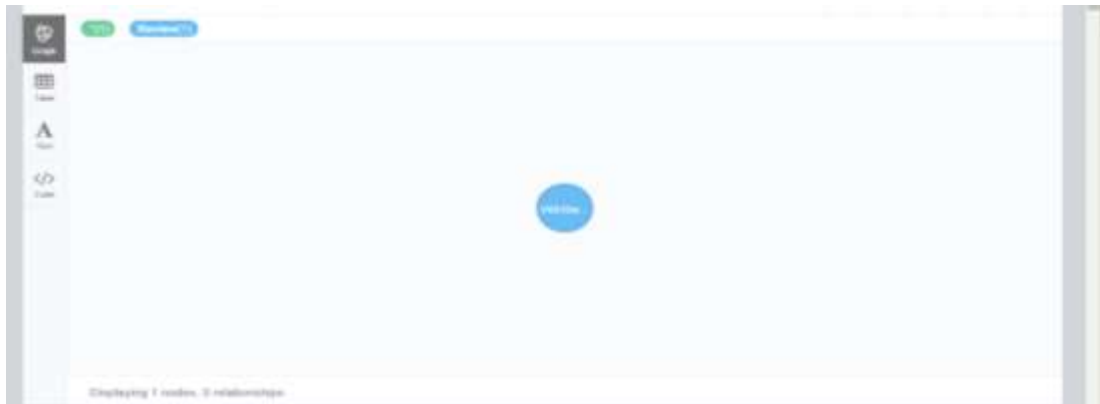
- **Business:**

```
load csv from "file:///business.csv" as b
create(:business{_id:b[0],address:b[1],bus_id:b[46],cat:b[47],cat_gen:b[50],city:b[55],is_open:b[
63],lat:b[64],long:b[65],name:b[66],neighbourhood:b[67],postal:b[68],rev_count:b[69],stars:b[7
0],state:b[71]})
```

Nodes:

Review:

```
MATCH (n:Review) RETURN n limit 1
```



Business:

```
MATCH (n:business) RETURN n limit 1
```



User:

`MATCH (n:User) RETURN n limit 1`



Relationships:

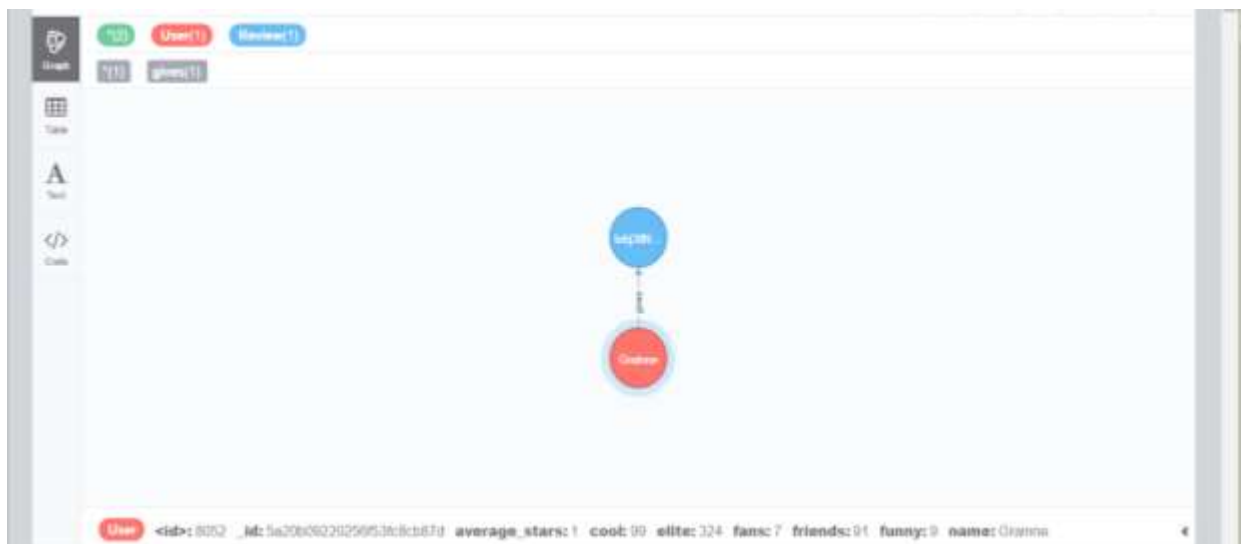
- **Between Review and Business**

```
MATCH (a:Review),(b:business)
WHERE a.business_id = b.bus_id
CREATE (b)-[h:Has]->(a);
MATCH p=()-[r:Has]->() RETURN p LIMIT 1
```



- **Between User and Review**

```
MATCH (a:User),(b:Review)
WHERE a.user_id = b.user_id
CREATE (a)-[h:gives]->(b);
MATCH p=()-[r:gives]->() RETURN p limit 1
```



BASIC UNDERSTANDING OF THE DATA

We will try to have a basic understanding of the data with the help of the following queries:

1. What is the overall number of businesses?

Dataset: Business

Query:

```
db.business.aggregate({$group:{_id:"$business_id"}},{ $count:"TotalBusiness"})
```

```
> db.business.aggregate({$group:{_id:"$business_id"}},{ $count:"TotalBusiness"})
{ "TotalBusiness" : 156639 }
```

```
db.business.distinct("business_id").length
```

```
> db.business.distinct("business_id").length
156639
```

Explanation:

There were two ways to approach this problem of finding the total number of distinct businesses in the complete dataset. First, we can group the data using business id and find the count of all the entries. Second, we can use distinct to find all different business ids and length of the same.

2. What is the overall number of reviews?

Dataset: Review

Query:

```
db.sample_review.aggregate([{$group:{_id:"$review_id"}},{ $count:"Total"}],{allowDiskUse:true})
```

```
> db.sample_review.aggregate([{$group:{_id:"$review_id"}},{ $count:"Total"}],{allowDiskUse:true})
{ "Total" : 1199468 }
```

Explanation:

In this query, we try to find the total number of reviews in the review dataset. We found that there are total 1199468 reviews in this set. We group the reviews with the review id to find all distinct reviews.

3. What is the overall number of reviewers?

Dataset: User

Query:

```
db.sample_user.aggregate({$group:{_id:"$user_id"}},{ $count:"TotalReviewers"})
```

```
>  
> Query: db.sample_user.aggregate({$group:{_id:"$user_id"}},{ $count:"TotalReviewers"})  
{ "TotalReviewers" : 339508 }  
>  
>
```

Explanation:

In this query, we try to find the total number of users in the user dataset. We found that there are total 339508 users in this set. We find the distinct users by grouping them on the basis of user id and finding their total number.

4. What is the overall number of reviews with ratings less than 3?

Dataset: Review

Query:

```
db.sample_review.aggregate([{$match:{stars:{ $lt:3 }}},{ $group:{_id:"$review_id"}},{ $count:"  
Total Reviews"}],{allowDiskUse:true})
```

```
> db.sample_review.aggregate([{$match:{stars:{ $lt:3 }}},{ $group:{_id:"$review_id"}},{ $count:"Total Reviews"}],{allowDiskU  
se:true})  
{ "Total Reviews" : 265760 }  
>  
>
```

Explanation:

To analyze the number of review less than 3, we use the review dataset.

First, we take all the reviews which have the stars less than 3 and grouped them using the review id to get values for all distinct reviews and calculated the total number of reviews. The result shows that there are total 265760 reviews who have the rating less than 3.

5. What is the overall number of reviews with ratings more than 3?

Dataset: Review

Query:

```
db.sample_review.aggregate([{$match:{stars:{ $gt:3 }}},{ $group:{_id:"$review_id"}},{ $count:"  
Total Reviews"}],{allowDiskUse:true})
```

```
> db.sample_review.aggregate([{$match:{stars:{$gt:3}}},{$group:{_id:"$review_id"},{$count:"Total Reviews"}},{allowDiskUse:true})
{ "Total Reviews" : 786719 }
```

Explanation:

We are analyzing the review database to find the total number of reviews with rating greater than 3. So, we group the data with the review id and use the ratings greater than 3 to find the total reviews.

6. What is the average number of reviews per business?

Dataset: Business

Query: `db.sample_business.aggregate([{$group:{"_id":"$business_id", avg:{$avg:"$review_count"}}})`

```
db.sample_business.aggregate([{$group:{"_id":"$business_id", avg:{$avg:"$review_count"}}})
{"_id": "yDRt1K6dLvrR_DZMubSJnQ", "avg": 3 }
{"_id": "Oq5Jv6cMe5wuy_Clwqh0Uu", "avg": 4 }
{"_id": "VIRyha5jnDXDntsPQ_SRNQ", "avg": 5 }
{"_id": "2B684j08PLAUpN6Svly_lQ", "avg": 3 }
{"_id": "FEH0175nTH81k451ZadKIQ", "avg": 256 }
{"_id": "awJC3VnaVwtBg0eXGnUj5g", "avg": 8 }
{"_id": "ekqC7Q2UrtN5EzRv4Rpztu", "avg": 9 }
{"_id": "9J9NkyI1bfnOY_nY6qeHzg", "avg": 12 }
{"_id": "Jt1fSVNbszJn_A3DtcyJqg", "avg": 3 }
{"_id": "PtqKpZWuvG7mF4hMx-9Sug", "avg": 3 }
{"_id": "U--dPMXyMaks6dV9A3nEw", "avg": 54 }
{"_id": "X3E_gq0I1Na0kkrGEgxq9A", "avg": 13 }
{"_id": "QOmR4Z53knbIDwRrVdN7ZA", "avg": 3 }
{"_id": "STBMrqx1IHuBz6h_qQ1Xbg", "avg": 12 }
{"_id": "L772e612Yd8DJEyCBx8Nng", "avg": 167 }
{"_id": "eUcOSFmNvXVkyB8QF4SOPEQ", "avg": 4 }
{"_id": "v0-OfmJhdCGnhrPjG9aVzA", "avg": 5 }
{"_id": "6sARBqQp1IEBKsALJ0zVA", "avg": 15 }
{"_id": "wGCM32sRvy6010LHVnHPDw", "avg": 8 }
{"_id": "rsVSc6hVYCbo61tnH2u5_u", "avg": 3 }
```

Explanation:

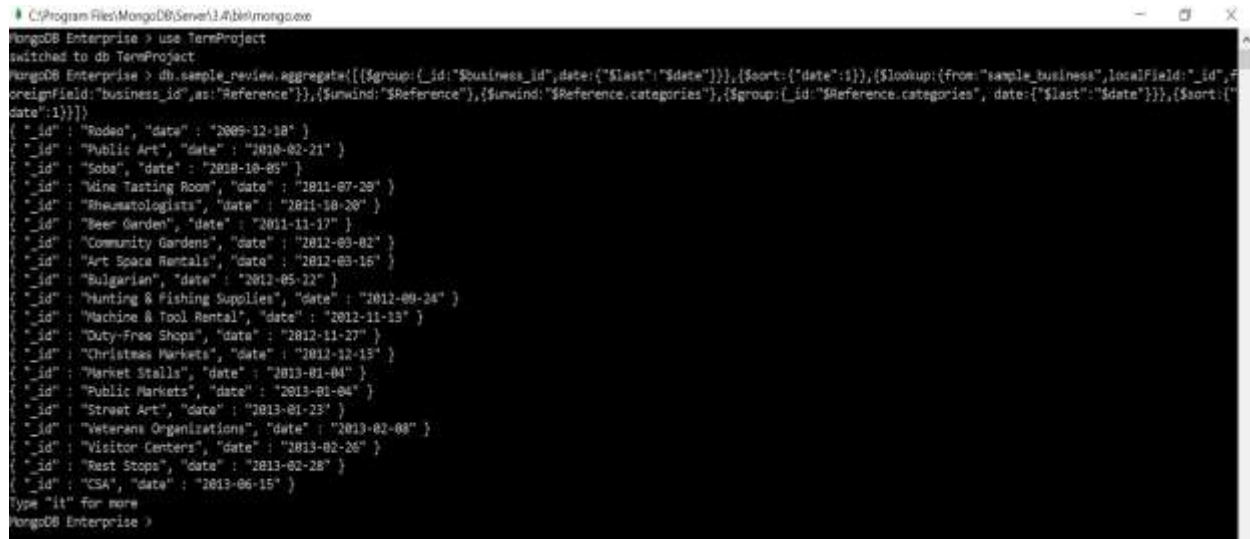
We are trying to find the average number of reviews in a business. For this query, we used business database and grouped them with the business id to calculate the average of the review count. We found that there is an average of 3 reviews for a business "yDRt1K6dLvrR_DZMubSJnQ". We can use this data to show it on the business metrics page and infer if the business needs to improve or maintain the rating.

7. What are the dates of first review per category?

Dataset: Review

Query:

```
db.sample_review.aggregate([{$group:{_id:"$business_id",date:{"$last":"$date"}}},{ $sort:{"date":1}},{ $lookup:{from:"sample_business",localField:"_id",foreignField:"business_id",as:"Reference"}},{ $unwind:"$Reference"},{$unwind:"$Reference.categories"},{$group:{_id:"$Reference.categories", date:{"$last":"$date"}}},{ $sort:{"date":1}}])
```



```
C:\Program Files\MongoDB\Server\3.4\bin\mongo.exe
MongoDB Enterprise > use TernProject
switched to db TernProject
MongoDB Enterprise > db.sample_review.aggregate([{$group:{_id:"$business_id",date:{"$last":"$date"}}},{ $sort:{"date":1}},{ $lookup:{from:"sample_business",localField:"_id",foreignField:"business_id",as:"Reference"}},{ $unwind:"$Reference"},{$unwind:"$Reference.categories"},{$group:{_id:"$Reference.categories", date:{"$last":"$date"}}},{ $sort:{"date":1}}])
{"_id":"Rodeo","date":"2009-12-18"}
{"_id":"Public Art","date":"2010-02-21"}
{"_id":"Soba","date":"2010-10-05"}
{"_id":"Wine Tasting Room","date":"2011-07-29"}
{"_id":"Rheumatologists","date":"2011-10-20"}
{"_id":"Beer Garden","date":"2011-11-17"}
{"_id":"Community Gardens","date":"2012-03-02"}
{"_id":"Art Space Rentals","date":"2012-03-16"}
{"_id":"Bulgarian","date":"2012-05-22"}
{"_id":"Hunting & Fishing Supplies","date":"2012-09-24"}
{"_id":"Machine & Tool Rental","date":"2012-11-13"}
{"_id":"Duty-Free Shops","date":"2012-11-27"}
{"_id":"Christmas Markets","date":"2012-12-13"}
{"_id":"Market Stalls","date":"2013-01-04"}
{"_id":"Public Markets","date":"2013-01-04"}
{"_id":"Street Art","date":"2013-01-23"}
{"_id":"Veterans Organizations","date":"2013-02-08"}
{"_id":"Visitor Centers","date":"2013-02-26"}
{"_id":"Rest Stops","date":"2013-02-28"}
{"_id":"CSA","date":"2013-06-15"}
Type "it" for more
MongoDB Enterprise >
```

Explanation:

We have combined two datasets- business and review as the category field is defined in business dataset and review texts are in the review dataset. In this query, we group the reviews based on business id and sorted them on the basis of date. The business was looked in the business database using lookup and the keys used were business id in review database and id in business database. We used unwind to go through all the business and categories and grouped them again to find the category for that business.

8. Which are the top 10 most verbose reviewers?

Dataset: Review

Query:

```
db.sample_review.aggregate([{$project:{"review_id":1,"user_id":1,_id:0,"reviewTextLength":{"$strLenCP:"$text"}}},{ $sort:{"reviewTextLength":-1}},{ $project:{"reviewTextLength":0}},{ $limit:10}])
```



```
> db.sample_review.aggregate([{$project:{review_id:1,user_id:1,_id:0,reviewTextLength:{$strLenCP:$text}}},{$sort:{reviewTextLength:-1}},{$project:{reviewTextLength:0}},{$limit:10}])
{ "review_id" : "nZzCynDXMygfEU1Yobk1g", "user_id" : "Rp-c5IHm-EK8xg8573PHNg" }
{ "review_id" : "JQvRIEq9_AOU9uSMGUA-w", "user_id" : "3GNk1KBr_1v-tywV84sE5Q" }
{ "review_id" : "rdQvA1KNI21VwAkgR3wTsg", "user_id" : "DXBR8BU2XoKvgSohqR8TZg" }
{ "review_id" : "ey23R1xndFkP66s5FwV7w", "user_id" : "CLh48KGGb1AFCHVaalbU4g" }
{ "review_id" : "Kz5K9eu-9QYAFrv13Fivlg", "user_id" : "ddrg5x188B9fcfXVViscQA" }
{ "review_id" : "kShZ27MOMUKvUE4C6S8wPQ", "user_id" : "nj3sy1Ls31ABLTPVyl0ivA" }
{ "review_id" : "K28WVzcq7DFNFQJVS1j8A", "user_id" : "LRxRMV1GEAFK2UVJ56MRg" }
{ "review_id" : "W2d1Ca_2FR8_DbgvAdayCg", "user_id" : "5412e_28dTjmgT4t1cgSA" }
{ "review_id" : "DF3bqhRstr36h1KXsWdQpw", "user_id" : "Hf1p6Jr61bfYU58HmDg_fu" }
{ "review_id" : "XESCH2N76p3NH2p_2n_A2Q", "user_id" : "2c35m1w_9jdLANJpRve21Q" }
```

Explanation:

This query analyzes the top 10 most verbose reviewers. In this query, we use review database to calculate the length of all the reviews and sort this text length in the descending order. We also displayed that the user id of the user who wrote that review.

(b) Analysis on positive and negative reviews

Dataset: Review

Query:

```
db.reduced_review.aggregate([{$match:{stars:{$gt:3}}},{$group:{_id:{bussid:"$business_id"},rating:{$push:"$stars"}}},{$limit:10},{$project:{rating:{$sum:"$rating"}}},{$allowDiskUse:true}).pretty()
```

```
> db.reduced_review.aggregate([{$match:{stars:{$gt:3}}},{$group:{_id:{bussid:"$business_id"},rating:{$push:"$stars"}}},{$limit:10},{$project:{rating:{$sum:"$rating"}}},{$allowDiskUse:true}).pretty()
{ "_id" : { "bussid" : "BII12Kgh-3FJMqDQv1H-Ug" }, "rating" : 58 }
{ "_id" : { "bussid" : "oWEmuRdJpt8Xrh6Mzd8KFA" }, "rating" : 29 }
{ "_id" : { "bussid" : "cl8Tx8m7N1K3mJ52CKbMQA" }, "rating" : 4 }
{ "_id" : { "bussid" : "LdFj0iE02okpp0cWBr04A" }, "rating" : 29 }
{ "_id" : { "bussid" : "7I4TUpu38kLnkuQkoe0QuQ" }, "rating" : 9 }
{ "_id" : { "bussid" : "A_8fGxpCFUVD0byu7DihjQ" }, "rating" : 38 }
{ "_id" : { "bussid" : "sYUsL6Qg4axZduPrJpd10g" }, "rating" : 55 }
{ "_id" : { "bussid" : "JxC-_DMSGtfgsgD6guAcQ" }, "rating" : 5 }
{ "_id" : { "bussid" : "982ki3k4t72mr-04Ry_soQ" }, "rating" : 24 }
{ "_id" : { "bussid" : "77aZ2bz8S1zRYYS_Hs1o8A" }, "rating" : 97 }
```

Explanation:

This query analyzes the total number of positive reviews for a particular business, which is identified using parameter stars. We group the reviews on the basis of the business id and add all the ratings to a list and calculated the size of that list. The result show that the buss_id "BII12Kgh-3FJMqDQv1H-Ug" has total 58 positive ratings.

Dataset: Review

Query:

```
db.reduced_review.aggregate([{$match:{"stars":{$gt:3}}},{ $group: {_id:{$business_id"},rating:{$push:"$stars"}}},{ $limit:10},{ $project: {rating:{$sum:"$rating"}}}],{allowDiskUse:true}).pretty()
```

Explanation:

From the above 2 queries, it is evident that “Bil12Kgh-3FJMqDQv1H-Ug” has more positive reviews than negative reviews, and this information can be used to display on the business page to know how well the business is performing.

Query:

[illegible]

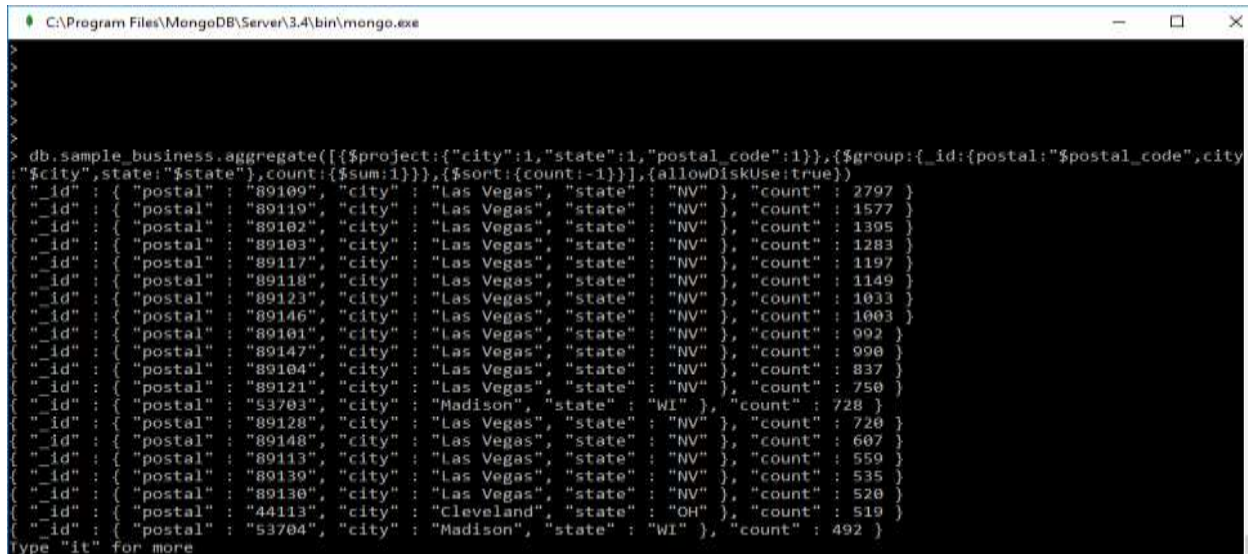
ANALYTICS

➤ To show the most happening area in a city and state

We are analyzing the business database. We are taking the total business in an area on the basis of postal code in a particular region. We see that the postal code 89109 has the largest number of businesses, i.e., 2797. From the above results, we can thus imply that “Las Vegas” is the most happening city as most of the businesses are set there. This “Most happening city” tag can be used on Yelp to attract more users to businesses in a particular region.

Query:

```
db.sample_business.aggregate([{$project:{"city":1,"state":1,"postal_code":1}},{$group: {_id:{"postal":"$postal_code",city:"$city",state:"$state"},count:{$sum:1}}},{ $sort: {count:-1}}],{allowDiskUse:true})
```



```
C:\Program Files\MongoDB\Server\3.4\bin\mongo.exe
>
>
>
> db.sample_business.aggregate([{$project:{"city":1,"state":1,"postal_code":1}},{$group: {_id:{"postal":"$postal_code",city:"$city",state:"$state"},count:{$sum:1}}},{ $sort: {count:-1}}],{allowDiskUse:true})
{ "_id" : { "postal" : "89109", "city" : "Las Vegas", "state" : "NV" }, "count" : 2797 }
{ "_id" : { "postal" : "89119", "city" : "Las Vegas", "state" : "NV" }, "count" : 1577 }
{ "_id" : { "postal" : "89182", "city" : "Las Vegas", "state" : "NV" }, "count" : 1395 }
{ "_id" : { "postal" : "89183", "city" : "Las Vegas", "state" : "NV" }, "count" : 1283 }
{ "_id" : { "postal" : "89117", "city" : "Las Vegas", "state" : "NV" }, "count" : 1197 }
{ "_id" : { "postal" : "89118", "city" : "Las Vegas", "state" : "NV" }, "count" : 1149 }
{ "_id" : { "postal" : "89123", "city" : "Las Vegas", "state" : "NV" }, "count" : 1033 }
{ "_id" : { "postal" : "89146", "city" : "Las Vegas", "state" : "NV" }, "count" : 1003 }
{ "_id" : { "postal" : "89101", "city" : "Las Vegas", "state" : "NV" }, "count" : 992 }
{ "_id" : { "postal" : "89147", "city" : "Las Vegas", "state" : "NV" }, "count" : 990 }
{ "_id" : { "postal" : "89104", "city" : "Las Vegas", "state" : "NV" }, "count" : 837 }
{ "_id" : { "postal" : "89121", "city" : "Las Vegas", "state" : "NV" }, "count" : 750 }
{ "_id" : { "postal" : "53703", "city" : "Madison", "state" : "WI" }, "count" : 728 }
{ "_id" : { "postal" : "89128", "city" : "Las Vegas", "state" : "NV" }, "count" : 720 }
{ "_id" : { "postal" : "89148", "city" : "Las Vegas", "state" : "NV" }, "count" : 607 }
{ "_id" : { "postal" : "89113", "city" : "Las Vegas", "state" : "NV" }, "count" : 559 }
{ "_id" : { "postal" : "89139", "city" : "Las Vegas", "state" : "NV" }, "count" : 535 }
{ "_id" : { "postal" : "89130", "city" : "Las Vegas", "state" : "NV" }, "count" : 520 }
{ "_id" : { "postal" : "44113", "city" : "Cleveland", "state" : "OH" }, "count" : 519 }
{ "_id" : { "postal" : "53704", "city" : "Madison", "state" : "WI" }, "count" : 492 }
Type "it" for more
```

➤ To find the list of top 10 most recent reviews

We analyze the review database to sort the reviews written by users on the basis of the date on which they were written. The latest reviews are the most recent and helpful reviews. This metrics can be helpful for the users to look for a business.

Query:

```
db.sample_review.aggregate([{$group: {_id:{"buss_id":"$business_id"},reviews:{$push:{"reviewtext":"$text"}}}},{$limit:10},{ $sort: {product:1,date:1 }},{ $project: {review:{$slice:["$reviews",10]}}}],{allowDiskUse:true}).pretty()
```

```

Select C:\Program Files\MongoDB\Server\3.2\bin\mongo.exe
> db.sample_review.aggregate([{$group: {_id: {business_id: "$business_id"}, reviews: {$push: {reviewtext: "$text"} }}, {$limit: 10}, {$sort: {product1.date: -1}}, {$project: {review: {$$
list: ["review", "_id"]}}, {allowDiskUse: true}}).pretty()
{
  "_id" : {
    "business_id" : "--0d4a81678b31550de9f0"
  },
  "review" : [
    {
      "reviewtext" : "This will be the last year this event will be held in Las Vegas:( I was fortunate enough to enjoy and be a food judge for all 3
years. It had moved from Bally's to Downtown Las Vegas. There was always plenty to do, see and sample. There were vendors around every turn giving out samples and sang.
Demonstrations and chefs cooking could be watched through out the days.\nI agree with a fellow Velper that the Bourbon, BBQ, and BBQs event was poorly run (not surpris
ed). Line to get in went horribly slow. They ran out of the bourbon they were giving samples of and alot of the BBQ booths ran out of ribs before the event was over. Fav
orites were The Shed and Lucky 13! There were no tables or chairs to set your ribs, who was in charge of this? On top of everything else there was 1 security guard the
t kept yelling at people even though they weren't doing anything wrong, preaching to them about crossing the street correctly, etc. We just wanna eat and drink! Please
leave us alone!\nOver all its always a good time. It will be in Florida next year:("
    },
    {
      "reviewtext" : "Disclaimer: I attended the World Food Championship (WFC) as a judge and media member. I also won free admission to the Burger Ex
travaganza event, courtesy of Yelp.\nWhen I first heard about the inaugural WFC, I was immediately intrigued. It's a food competition featuring the best of the best.
Categories like chili, burgers, sandwiches, and BBQ are represented. It's all American food, so it doesn't exactly represent the "world." Adam Richman was the perfect
host for this event, as everyone knows him from Man vs. Food.\nI had a great time judging food. For categories like sandwiches and burgers, the competition went thro
ugh several rounds of judging. The finalists would then have their food judged by a panel of celebrities and chefs, most notably Adam Richman. As for the BBQ, I thought
the samples I had were very good -- all of the BBQ teams are previous grand champions, so they brought their game.\nFrom my point of view, I thought that the event w
as reasonably organized. Something of this size with so many people and logistics involved, I'm sure this took a lot of work to set up. There were a lot of booths from
sponsors set up at the front of Bally's right by the Strip, and that attracted the majority of the crowd. The stage up front was perfect to show off demos, music, and a
nouncements. The sandwich and burger competitions were also held at the front of the hotel.\nHowever, the energy at the back of the Bally's hotel was dead. It was th
e area for the BBQ and chili competition. Maybe because it covered so much ground, but most of the time it seemed really quiet. If I was a casual foodie or passer-by, I
wouldn't have known about the area, so they probably could've done a better job with putting up signs. Many of the eating and VIP experiences were very expensive -- so
much that the casual foodie would probably not fork over the money.\nPersonally, I had a great time. At the Burger Extravaganza, I got to sample multiple gourmet b
urgers (wilder sized), along with drinks and dessert. I even got to chat with Adam for a few minutes. He's just a normal, cool guy who loves his food and BBQ. Safe to s
ay, his shows on Travel Channel has skyrocketed him to celebrity status. It remains to be seen if they will organize this event next year. If they do, you know where I'
ll be!"
    },
    {
      "reviewtext" : "I was really disappointed in this event. I bought the so-called V.I.F. (Very Important Foodie) pass for $275 for myself and a fr
iend because it said it covered all but one major event. Upon arrival, none of the information staff (volunteers) knew anything about it. I had also signed up to judge
and was sent to the D Hotel, 12th floor WFC Command Center. No one there knew anything about scanning my receipt in order to receive my passes. I waited at a minimu
m to be given an itinerary of events, times and places. Nope. Those were only provided to official staff. \n\nThe opening reception was ridiculous. It was held on the roo
ftop of the OLV and there were probably close to 1000 people there. There were lines to get to the bar for the "Free" but awful wine, $8 or beer in a plastic cup cost
$7. Mixed drinks were more. The servers were muddled as they came into the crowd so if you weren't poised to strike as they came out, no fun! for you. I was able to snag
a chicken stick (no satay sauce), a meat/mushroom puff, a mini-eggroll and a cocktail pig in a blanket. Wow! I did mention that this the World Food Championships we're
at, right? Pinta I was told that the only judges meeting would be just before my event. I asked exactly where and when my judging event was and was told to report to th
e judges tent an hour ahead of time. When I did, I was told to come back an hour later. Coming back at that time, I damn near missed my event. The judging itself was fu
"
    }
  ]
}

```

➤ Top 10 prolific reviewers

As we need to show the most prolific reviewer, we will be grouping the collection by the business id and calculating the total number of reviews written by all the reviewers for that business. We will be sorting the number of reviews written in the descending order and limiting the number of reviews to only 10. The result shows the number of reviews written by a particular reviewer.

Query:

```

db.sample_review.aggregate([{$group: {_id: {Reviewer: "$user_id"}, NoOfReviewsWritten: {$sum: 1}}}, {$project: {user_id: 1, NoOfReviewsWritten: 1}}, {$sort: {NoOfReviewsWritten: -1}}, {$limit: 10}], {allowDiskUse: true}).pretty()

```

```

Select C:\Program Files\MongoDB\Server\3.2\bin\mongo.exe
> db.sample_review.aggregate([{$group: {_id: {Reviewer: "$user_id"}, NoOfReviewsWritten: {$sum: 1}}}, {$project: {user_id: 1, NoOfReviewsWritten: 1}}, {$sort: {NoOfReviewsWritten: -1}}, {$limit: 10}], {allowDiskUse: true}).pretty()
{
  "_id" : {
    "Reviewer" : "P6C4x8bL_Fk1h2ed0F2u0BQ"
  },
  "NoOfReviewsWritten" : 128
},
{
  "_id" : {
    "Reviewer" : "h188881ggf9u0R12Q-2ju"
  },
  "NoOfReviewsWritten" : 126
},
{
  "_id" : {
    "Reviewer" : "lvc0ndw12h8q63q11nguu"
  },
  "NoOfReviewsWritten" : 89
},
{
  "_id" : {
    "Reviewer" : "h888718x1288k11x_sauu"
  },
  "NoOfReviewsWritten" : 84
},
{
  "_id" : {
    "Reviewer" : "x121Q8_jkYy0r-luVWw3Chg"
  },
  "NoOfReviewsWritten" : 81
},
{
  "_id" : {
    "Reviewer" : "F31q07p1Gex104H4qd.Fpg"
  },
  "NoOfReviewsWritten" : 56
},
{
  "_id" : {
    "Reviewer" : "B8CvVyp1n0c54x3hvvt1g"
  },
  "NoOfReviewsWritten" : 53
}

```

➤ Review Histogram

We are using review database for this analysis and we find the review histogram for ratings. The review ratings of number of 5's, 4's, 3's, 2's and 1's are counted, and the count is shown. We group on the basis of business_id and rating and represent the total number of reviews that the business has got. This analysis can help the business to know their performance and can take the appropriate measures to improve the same.

Query:

```
db.sample_review.aggregate([{$group: {_id: {business_id: "$business_id", stars: "$stars"}, count: {$sum: 1}}}], {allowDiskUse: true})
```

```
> db.sample_review.aggregate([{$group: {_id: {business_id: "$business_id", stars: "$stars"}, count: {$sum: 1}}}], {allowDiskUse: true})
{"_id": {"business_id": "Dm9p-2DX-3ID9B_DslHauQ", "stars": 5 }, "count": 2 }
{"_id": {"business_id": "Dm9p-2DX-3ID9B_DslHauQ", "stars": 1 }, "count": 1 }
{"_id": {"business_id": "V4M9X-wK_6H-1ELbqUmQvvg", "stars": 5 }, "count": 8 }
{"_id": {"business_id": "2muD3YfbRQo1f6H5p3jlnA", "stars": 5 }, "count": 3 }
{"_id": {"business_id": "2muD3YfbRQo1f6H5p3jlnA", "stars": 4 }, "count": 1 }
{"_id": {"business_id": "H6kP7sKjIhTKMo55jer1Kg", "stars": 3 }, "count": 5 }
{"_id": {"business_id": "H6kP7sKjIhTKMo55jer1Kg", "stars": 5 }, "count": 6 }
{"_id": {"business_id": "H6kP7sKjIhTKMo55jer1Kg", "stars": 1 }, "count": 14 }
{"_id": {"business_id": "4V2tiI7RXt60jo_k1go2-A", "stars": 5 }, "count": 42 }
{"_id": {"business_id": "m5_9gtI0Yw6m9DYrzgC8hQ", "stars": 4 }, "count": 2 }
{"_id": {"business_id": "m5_9gtI0Yw6m9DYrzgC8hQ", "stars": 5 }, "count": 2 }
{"_id": {"business_id": "D47sxEtW7DZHfUnwKnrjRsg", "stars": 4 }, "count": 1 }
{"_id": {"business_id": "D47sxEtW7DZHfUnwKnrjRsg", "stars": 2 }, "count": 2 }
{"_id": {"business_id": "D47sxEtW7DZHfUnwKnrjRsg", "stars": 5 }, "count": 20 }
{"_id": {"business_id": "7k3BHUGfx3aUqNudYvcjgw", "stars": 5 }, "count": 2 }
{"_id": {"business_id": "7k3BHUGfx3aUqNudYvcjgw", "stars": 1 }, "count": 1 }
{"_id": {"business_id": "-8d4L4U3vXnT18MEgyX_WA", "stars": 5 }, "count": 6 }
{"_id": {"business_id": "qdrXZHTwW-XC_RVqIhN7QA", "stars": 1 }, "count": 3 }
{"_id": {"business_id": "pa3q5sCfjgbo3TxPwpQNLtW", "stars": 3 }, "count": 4 }
{"_id": {"business_id": "-1Z9wSHdSYlmqcTdnzztVA", "stars": 4 }, "count": 5 }
Type "it" for more
```

➤ Oldest user with Maximum number of reviews and total number of friends

We are using user dataset to find the oldest user with maximum number of reviews. We will analyze when yelp reviews have started and how many reviews are written by oldest user and can display this information in the leaderboard. We see that the oldest user is in year 2004 when yelp started and has the number of reviews as 1317. This in turn can help the business to understand the trends of reviews with the most loyal and oldest customer.

Query:

```
db.reduced_user.aggregate([{$project: {_id: 0, user_id: "$_id", "yelping_since": 1, "review_count": 1, "friends": {$size: "$friends"} }}, {$sort: {"yelping_since": 1, "review_count": -1}}, {$limit: 10}], {allowDiskUse: true}).pretty()
```

```

> db.review.aggregate([{$project:{_id:0, user_id:"$id","yelping_since":1,"review_count":1,"friends":{$size:"$friends"}}},{$sort:{"yelping_since":1,"review_count":-1}},{$limit:10}],{allowDiskUse:true}),pretty()
{
  "review_count" : 1317,
  "yelping_since" : "2004-10-12",
  "user_id" : ObjectId("5a162d80a5b5b5b5b5b5b5b5"),
  "friends" : 6736
}
{
  "review_count" : 991,
  "yelping_since" : "2004-10-12",
  "user_id" : ObjectId("5a162d80a5b5b5b5b5b5b5b5"),
  "friends" : 625
}
{
  "review_count" : 105,
  "yelping_since" : "2004-10-12",
  "user_id" : ObjectId("5a162d80a5b5b5b5b5b5b5b5"),
  "friends" : 388
}
{
  "review_count" : 104,
  "yelping_since" : "2004-10-12",
  "user_id" : ObjectId("5a162d80a5b5b5b5b5b5b5b5"),
  "friends" : 24
}
{
  "review_count" : 105,
  "yelping_since" : "2004-10-15",
  "user_id" : ObjectId("5a162d80a5b5b5b5b5b5b5b5"),
  "friends" : 11
}
{
  "review_count" : 7,
  "yelping_since" : "2004-10-15",
  "user_id" : ObjectId("5a162d80a5b5b5b5b5b5b5b5"),
  "friends" : 0
}

```

➤ Filter By feature

Motive: Here we added a new feature on yelp page to select reviews based on a feature of a business. Since we worked on restaurants reviews. We selected the features – Price, ambience and quality. We implemented a text based query to sort the reviews based on the relevance score (most relevant ones on top). Our dictionary included synonyms and words close to the feature that the customer is looking for.

Value Addition: It will be easier for users to find the reviews for a business that are relevant to the features they are looking for. For example, a couple going for dinner would be more interested in the ambience of the restaurant, so that they can sort the reviews with ambience to get information about that.

Index creation for text search:

```
db.review.createIndex( { text: "text" } )
```

1. Price

```

db.review.aggregate([{$match:{$text:{$search:"price cost expense charge penalty value money dollars costly fee tariff fare worth $ buck expenditure amount pay payment costed expensive loss penny"}}},{$group: {_id:{business_id:"$business_id",text:"$text"}}},{$project:{business_id:1,text:1,score:{$meta:"textScore"}}},{$sort:{score:-1}},{$limit:10}],{allowDiskUse:true})

```



```

1 db.review.aggregate([{$match:{$text:{$search:"price cost expense charge penalty value money dollars costs fee tariff fare worth $ loss expenditure amount pay payment costed expensive loss penny"}}}]
2

```

```

154 }
155 {
156   "_id" : ObjectId("5a20b36428504f33f0c3add"),
157   "review_id" : "Y10dwbVv8H13uq11u0g",
158   "user_id" : "53Fvz7yphdp-0kC5ghC11g",
159   "business_id" : "8aP1UD1s4QD6W9t7117ev",
160   "stars" : NumberInt(5),
161   "date" : "2016-02-01",
162   "text" : "The Pig's Nose stands out as one of the most unique and unexpectedly-enjoyable little bars I've ever had the fortune of walking into. Unfortunately, I can't",
163   "useful" : NumberInt(13),
164   "funny" : NumberInt(5),
165   "cool" : NumberInt(10)
166 }
167 {
168   "_id" : ObjectId("5a20b40c3d350f33f0c39269"),
169   "review_id" : "5a13Aa8H978601-1848b",
170   "user_id" : "5uTf1aayX0wuhR0rsk(b)",

```

2. Ambience

```

db.review.aggregate([{$match:{$text:{$search:"atmosphere ambience sit sitting light comfortable comfort cleanliness sanitation pure hygiene sterility purity disinfection air aura climate mood feel feeling character quality impression complexion flavor look tone tenor setting milieu background backdrop element environment conditions situation atmosphere"}}},
{$group:{$_id:{$business_id:"$business_id",text:"$text"}}},{$project:{$business_id:1,text:1,score:{$meta:"textScore"}}},{$sort:{$score:-1}},{$limit:10}],{$allowDiskUse:true})

```

```

1 db.review.aggregate([{$match:{$text:{$search:"atmosphere ambience sit sitting light comfortable comfort cleanliness sanitation pure hygiene sterility purity disinfection air aura climate mood feel feeling the
2
3 {$project:{$business_id:1,text:1,score:{$meta:"textScore"}}},{$sort:{$score:-1}},{$limit:10}],{$allowDiskUse:true})

```

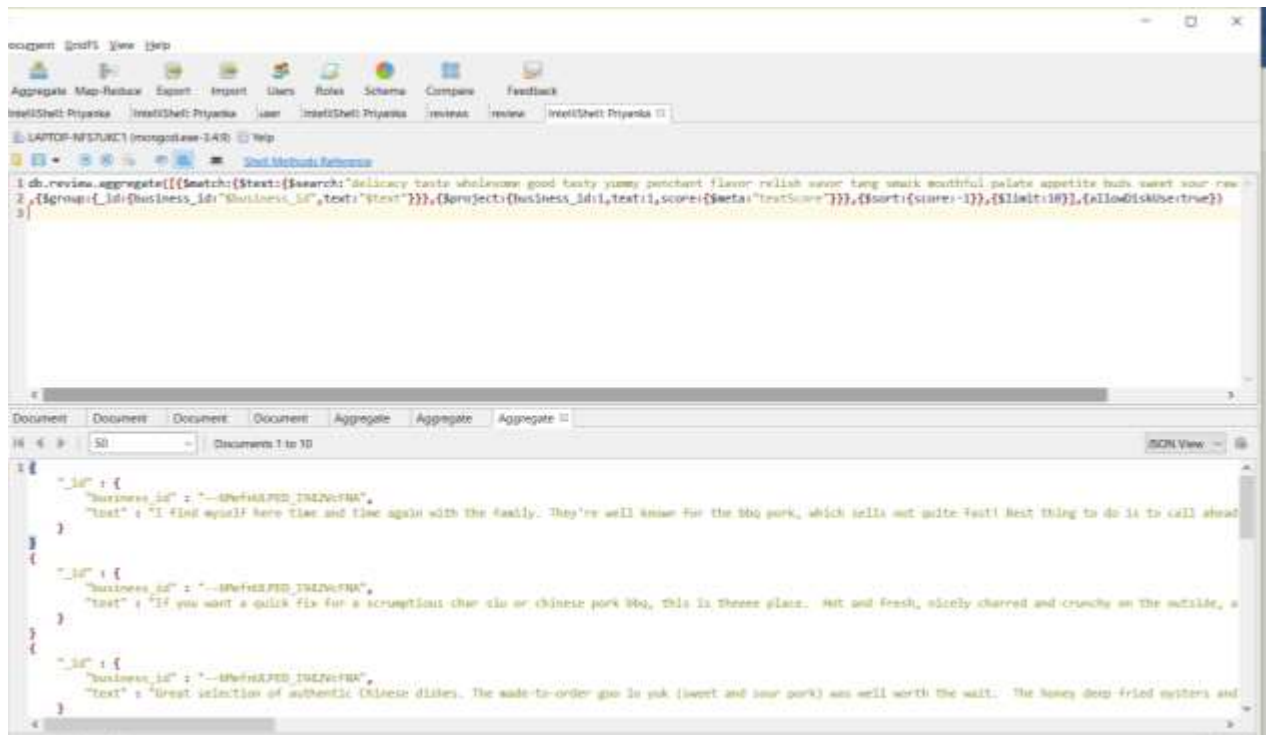
```

1 {
2   "_id" : {
3     "business_id" : "--6bfe80f2D-18426cF8A",
4     "text" : "John's Chinese BBQ Restaurant is one of those restaurants that you will feel so out of place if you come as a couple or don't even think about it alone \
5   }
6 }
7 {
8   "_id" : {
9     "business_id" : "--6bfe80f2D-18426cF8A",
10    "text" : "John's Chinese BBQ ruined Char Broiled Pork for me... forever. I cannot eat it at din um, or a Chinese food court, or anywhere for that matter, without thinki
11  }
12 }
13 {
14   "_id" : {
15     "business_id" : "--6bfe80f2D-18426cF8A",
16     "text" : "I come here with my gf and her family often and they never disappoint. The food is always served fresh and with typical Chinese restaurants the service i
17   }
18 }

```

3. Taste

```
db.review.aggregate([{$match:{$text:{$search:"delicacy taste wholesome good tasty yummy  
penchant flavor relish savor tang smack mouthful palate appetite buds sweet sour raw cooked  
swallow"}}}  
,{$group:{$_id:{business_id:"$business_id",text:"$text"}}},{$project:{business_id:1,text:1,score:  
{$meta:"textScore"}}},{$sort:{score:-1}},{$limit:10}],{allowDiskUse:true})
```



➤ Trending business based on the number of checkins

Currently, there is no metrics present to show trending business in a week. We will be showing the check-ins for a particular business based on the timings in a day. So, we will be grouping the dataset on the basis of business and timings on all the days in a week. This metrics can be helpful in analyzing that the timings during which the business is going to be busiest, which in-turn can help in inferring the trending business.

Query:

```
db.checkin.aggregate([{$group:{$_id:{business_id:"$business_id",mon:"$time.Monday",tue: "$time.Tuesday",  
wed:"$time.Wednesday",thur:"$time.Thursday",fri:"$time.Friday",  
sat:"$time.Saturday",sun:"$time.Sunday"}}},{$limit:2}],{allowDiskUse:true}).pretty()
```



```

C:\Program Files\MongoDB\Server\3.4\bin\mongo.exe
> db.checkin.aggregate([{$group: {_id: {Business_ID: "$business_id", mon: "$time.Monday", tue: "$time.Tuesday", wed: "$time.Wednesday", thur: "$time.Thursday", fri: "$time.Friday", sat: "$time.Saturday", sun: "$time.Sunday"}}, {$limit: 2}}, {$allowDiskUse: true}).pretty()
{
  "_id" : {
    "business_id" : "...6MeFNUlPED_1942vcFMA",
    "mon" : {
      "17:00" : 1,
      "18:00" : 2,
      "19:00" : 1,
      "20:00" : 4,
      "21:00" : 3,
      "22:00" : 1,
      "23:00" : 1,
      "24:00" : 4
    },
    "tue" : {
      "17:00" : 2,
      "18:00" : 2,
      "23:00" : 1
    },
    "wed" : {
      "18:00" : 1,
      "19:00" : 1,
      "20:00" : 3,
      "21:00" : 3,
      "22:00" : 2
    }
  },
  "count" : 12,
  "list" : [
    {
      "Tip" : "Combo A: Roast duck, roast pork, Singapore noodle and sauteed veggies! Cash only."
    },
    {
      "Tip" : "Make reservation on weekend"
    },
    {
      "Tip" : "Great place for couple has $7.99 dish"
    },
    {
      "Tip" : "King of bbq pork for $22"
    },
    {
      "Tip" : "Their lunch combos for small groups is a decent deal"
    },
    {
      "Tip" : "Make sure to request the delicious house soup. It's on the house if you ask for it!"
    },
    {
      "Tip" : "$7.50 lunch special, dish of rice/noodles with soup"
    },
    {
      "Tip" : "$5 lunch special"
    },
    {
      "Tip" : "BBQ pork is sold out early on Saturday"
    },
    {
      "Tip" : "$6 lunch special. A lot of selection on the lunch menu."
    },
    {
      "Tip" : "At lunch, you can ask for dinner menu if you feel the lunch menu is bit lacking in taste"
    }
  ]
}

```

➤ Tips for a particular business

There are no tips present on the webpage of the yelp. The reviews are generally lengthy and it is a time consuming task to check for a particular thing in a review. Also, a user has to scroll down to the bottom to check for the reviews. So, we are trying to find the list of the tips and total number of tips for a particular business using the tip dataset. We grouped the data using business id and made a list of all the tips and calculated the total number of tips. This metrics is also helpful for a business to keep track if the business is performing well.

Query:

```

db.tip.aggregate([{$group: {_id: {Business_ID: "$business_id"}, count: {$sum: 1}, list: {$push: {Tip: "$text"} } } }], {$allowDiskUse: true}).pretty()

```

```

Select C:\Program Files\MongoDB\Server\3.4\bin\mongo.exe
> db.tip.aggregate([{$group: {_id: {Business_ID: "$business_id"}, count: {$sum: 1}, list: {$push: {Tip: "$text"} } } }], {$allowDiskUse: true}).pretty()
{
  "_id" : {
    "Business_ID" : "...6MeFNUlPED_1942vcFMA"
  },
  "count" : 12,
  "list" : [
    {
      "Tip" : "Combo A: Roast duck, roast pork, Singapore noodle and sauteed veggies! Cash only."
    },
    {
      "Tip" : "Make reservation on weekend"
    },
    {
      "Tip" : "Great place for couple has $7.99 dish"
    },
    {
      "Tip" : "King of bbq pork for $22"
    },
    {
      "Tip" : "Their lunch combos for small groups is a decent deal"
    },
    {
      "Tip" : "Make sure to request the delicious house soup. It's on the house if you ask for it!"
    },
    {
      "Tip" : "$7.50 lunch special, dish of rice/noodles with soup"
    },
    {
      "Tip" : "$5 lunch special"
    },
    {
      "Tip" : "BBQ pork is sold out early on Saturday"
    },
    {
      "Tip" : "$6 lunch special. A lot of selection on the lunch menu."
    },
    {
      "Tip" : "At lunch, you can ask for dinner menu if you feel the lunch menu is bit lacking in taste"
    }
  ]
}

```

➤ Helpfulness of the review

(a) Text Based Definition

Create Index:

```
db.sample_review.createIndex(text:"$text")
```

Query:

```
db.sample_review.aggregate([{$match:{$text:{$search:"astonishing awesome ugly amazing wonderful stunning impressive nice marvelous pleasant lovely great splendid spectacular excellent bad terrible good poor disappointed miserable awful dreadful horrible unpleasant healthy unhealthy nutritious beneficial harmful sick "}}},{ $group:{$_id:{$business_id:"$business_id",text:"$text"}}},{ $project:{$business_id:1,text:1,score:{$meta:"textScore"}}},{ $sort:{$score:-1}},{ $limit:10}},{allowDiskUse:true}).pretty()
```



```
C:\Program Files\MongoDB\Server\3.6\bin>mongo
> use sample_review
switched to db sample_review
> db.sample_review.aggregate([{$match:{$text:{$search:"astonishing awesome ugly amazing wonderful stunning impressive nice marvelous pleasant lovely great splendid spec
tacular excellent bad terrible good poor disappointed miserable awful dreadful horrible unpleasant healthy unhealthy nutritious beneficial harmful sick "}}},{ $group:{$_id:{$business_id:"$business_id",text:"$text"}}},{ $project:{$business_id:1,text:1,score:{$meta:"textScore"}}},{ $sort:{$score:-1}},{ $limit:10}},{allowDiskUse:true}).pretty()
{
  "_id" : {
    "business_id" : "--Rzj7lPBz1hGVjUs6A",
    "text" : "We love this store. We will always drive out of the way to go here!! A huge reason is Mike aka Gubby! He is awesome, very friendly and always smiling. All of the staff is friendly and really remember who we are after seeing us. The only down fall I think is that sometimes the syrup of cherry or vanilla is out. But that really doesn't change my mind! We love this store!! And love Mike!! Hire more people like him!!"
  },
  "score" : 1
},
{
  "_id" : {
    "business_id" : "--SnpzF1u0_VFwB_Cet0w",
    "text" : "A great place to go for the afternoon tea special, which starts at 3pm. Order anything from the special menu and get a choice of any drink for free. Not just limited to tea and coffee, you can order the fancy crushed ice drinks with ice cream at no extra charge. Untypical Hong Kong diner style food, baked rice and spaghetti, large selection of food on both the regular menu and the afternoon tea menu. Been here a few times for dinner and afternoon tea, the portions are decent. The consistency of the food ordered each time is practically the same. The taste of the food is always good and never disappoints. The rice noodle roll comes out steaming hot and is covered with sauces, served with wood skewers. It's soft and smooth but have to eat it while it's hot. The fried meats were on point for both dishes, crispy and well seasoned. In family style restaurant suitable for small or large groups. The service was mediocre but no complaints as it was expected. Overall, really enjoy the afternoon tea special and definitely a place to keep coming back to."
  },
  "score" : 1
},
{
  "_id" : {
    "business_id" : "--GdmwK30TB3aJG0eJrQ",
    "text" : "This will be the last year this event will be held in Las Vegas! I was fortunate enough to enjoy and be a food judge for all 3 years. It had moved from Bally's to Downtown Las Vegas. There was always plenty to do, see and sample. There were vendors around every turn giving out samples and swag. Demonstrations and chefs cooking could be satished through out the day. I agree with a fellow Valper that the Bourbon, BBQ, and Banquet event was poorly run (not surprised). Line to get in went horribly slow, they ran out of the bourbon they were giving samples of and alot of the BBQ booths ran out of ribs before the event was over. Favorites were The Shed and lucky 13! There were no tables or chairs to eat your ribs, who was in charge of this!? On top of everything else there was 1 security guard that kept yelling at people even though they weren't doing anything wrong, preaching to them about crossing the street correctly, etc. We just wanna eat and drink! Please leave us alone! Over all its always a good time. It will be in Florida next year!"
  },
  "score" : 1
},
{
  "_id" : {
    "business_id" : "--Rzj7lPBz1hGVjUs6A",
    "text" : "This is my favorite Circle K! The employees are so nice and friendly. They also have fresh brewed iced tea...which rocks my world!"
  },
  "score" : 1
}
```

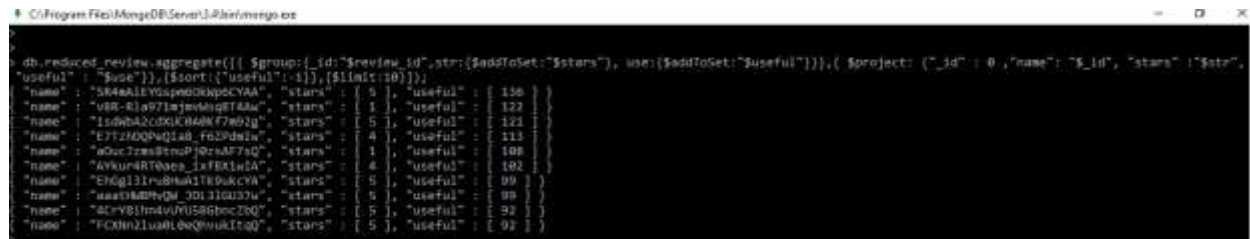
Explanation:

We are using review dataset to find the text based definition for helpfulness of the review. We grouped the data with a business id and a defined a certain set of words mentioned in the above query to understand if the reviews are helpful for a user, which is passed to match the texts in reviews. For matching to a certain set of words in the query, we first need to create an index for the text field. The query shows the result for only 10 reviews.

(b) Non Text Based Definition

Query:

```
db.reduced_review.aggregate([{$group:{$_id:"$review_id",str:{$addToSet:"$stars"},
use:{$addToSet:"$useful"}}},{ $project: {"_id": 0,"name": "$_id", "stars": "$str", "useful" :
"$use"}},$sort:{"useful":-1}},{$limit:10}]);
```



```
C:\Program Files\MongoDB\Server\3.2\bin>mongo.exe
>use review
>db.reduced_review.aggregate([{$group:{$_id:"$review_id",str:{$addToSet:"$stars"},
use:{$addToSet:"$useful"}}},{ $project: {"_id": 0,"name": "$_id", "stars": "$str", "useful" :
"$use"}},$sort:{"useful":-1}},{$limit:10}]);
{"name": "584a41e7a971a971a971a971", "stars": [ 5 ], "useful": [ 136 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 1 ], "useful": [ 122 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 5 ], "useful": [ 122 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 5 ], "useful": [ 122 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 4 ], "useful": [ 113 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 1 ], "useful": [ 108 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 4 ], "useful": [ 102 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 5 ], "useful": [ 99 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 5 ], "useful": [ 92 ] }
{"name": "584a41e7a971a971a971a971", "stars": [ 5 ], "useful": [ 92 ] }
```

Explanation:

We are using the review dataset to understand the helpfulness of the review on the basis of useful field in the review database. We group all the reviews and calculated the total number of reviews and displayed the result using useful field in the decreasing order. The results show that the first review id as the maximum number of useful field with the maximum rating. But, the second entry in the result shows useful reviews for less rating. We can infer that the reviews with lesser rating and more useful comments will also be counted as helpful review. We can further visualize these values to show the metrics for helpfulness on the web page.

➤ Identifying mob reviewers

Problem: A group of people can review on a business and give them either bad or good reviews. This might affect the business and change the overall rating of the business.

- ❖ 1st analysis – Whether any user has reviewed same business multiple times? This can implicate a fake review scenario where a single user is targeting a particular set of business.

Implementation: Grouped business_ids and user_ids and checked the count of such groups.

Finding: No such pattern found. Yelp doesn't allow that.

Query:

```
MATCH(r:Review)
with r.user_id as u1,r.business_id as b1,collect(r) as r1
unwind r1 as r2
with count(r2) as c,u1,b1
where c>1
return c,u1,b1
```



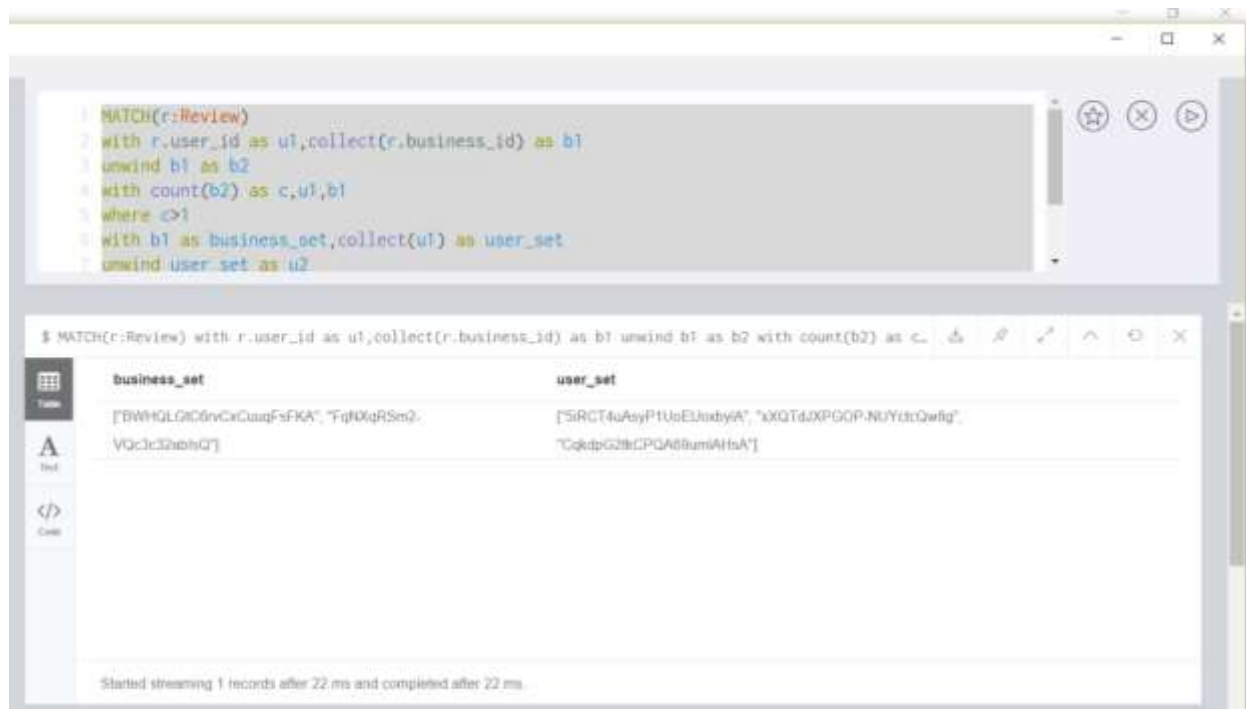
- ❖ 2nd Analysis - Are there patterns of mob reviews in data (whether a set of users have reviewed a common set of products. This can implicate a fake review scenario where a single user is targeting a set of business.

Implementation: To find this, we identified if a common set of business has been reviewed by multiple reviewers.

Finding: Such patterns found in data which can depicts a possibility of fake reviews (businesses targeted).

```
MATCH(r:Review)
with r.user_id as u1, collect(r.business_id) as b1
unwind b1 as b2
with count(b2) as c, u1, b1
where c > 1
with b1 as business_set, collect(u1) as user_set
unwind user_set as u2
with count(u2) as c2, business_set, user_set
where c2 > 1
return business_set, user_set
```

Note: Due to space issues we used a very limited data sample, but even in that sample we found a set of 3 users who reviewed a common set of 2 businesses.



Value addition: This analysis can find the mob review pattern and find out the fake reviewers who are targeting a set of businesses.

➤ Uniformity or divergence in business ratings in close proximity

We grouped restaurants with less than 10 geodesic distance, calculated the maximum, minimum, mean and standard deviation of their ratings.

Assumptions:

1. Standard deviation above 1 is assumed as threshold for divergence.
2. Proximity is assumed to within 100 geodesic distance.
3. Only 1 city has been considered in the query.

Findings: Although some difference of ratings between some restaurants were high but overall deviation for most of those groups were below 1.

Query:

```
MATCH (a:business{city:"Esslingen"})
MATCH(b:business)
WHERE 2 * 6371 * asin(sqrt(haversin(radians(toint(a.lat) -toint(b.lat)))+
cos(radians(toint(a.lat))) *
cos(radians(toint(a.lat))) * haversin(radians(toint(a.long) - toint(b.long))))) < 100.0 and
a.name<>b.name
with a.name as n1,collect(tofloat(b.stars)) as s1,collect(b.name) as n2
```

```

unwind s1 as s2
return n2,s1,stDev(s2) as standard_deviation,AVG(s2) as mean,max(s2) as
maximum_rating,min(s2) as minimum_rating

```

```

$ MATCH (a:business{city:"Esslingen"}) MATCH(b:business) WHERE 2 * 6371 * asin(sqrt(haversin(radians(to
cos(radians(toint(a.lat))) + haversin(radians(toint(a.long) - toint(b.long))))) < 100.0 and
a.name<>b.name
with a,name as n1,collect(tofloat(b.stars)) as s1,collect(b.name) as n2
unwind s1 as s2
return n2,s1,stDev(s2) as standard_deviation,AVG(s2) as mean,max(s2) as maximum_rating,min(s2)
as minimum_rating

```

n2	s1	standard_deviation	mean	maximum_rating	minimum_rating
["Royal Pizza Service", "Hotel Pfefferburg"]	[1, 3.5]	0.7718479277897064	3.711180134223603	5	1
"Cafe Fruchdee", "Valeo Pizzai", "Valeo",	4, 3.5]				
"Mezzai", "1 Sindelfinger Kaffeehaus",	4, 4,				
"Galerie", "Amadeus Restaurant & Bar",	3.5,				
"Gitzinger Wirtsh", "Imbiss Bräu", "Mezzai",	4.5, 4,				
"Leri W&K", "Schweiner", "super pizzaservice",	3, 4,				
"Organi", "Cafébräu", "Café - Konditorei Karl	4.5,				
W&H Nachf.", "Lakoni's Restaurant Hirsch",	3.5,				
"Amazzai", "Restaurant Corbis", "Steffen",	3.5,				
"New Asia Hofbr", "Brauhaus Am	4.5,				
Sollideplatz", "Leri's", "Le P&H", "Ratskeller	3.5,				
De Villiers", "Weber", "F&H", "Hotel am Park",	4.5, 2,				
"Restaurant Madagasc", "Radissonparkhotel	5, 2,				
Edelweiss", "Trollinger", "Theng Long",	3.5, 4,				
"Bäckerei & Konditorei Hahndörfer",	4.5, 4,				
"Baumstübe Chez Valere", "McDonald's",	4.5, 3,				
"Neckarberggarten", "Chipsie Kumpi",	4.5, 3,				

➤ Filter Review by friend

Problem: There is no filtering of reviews by friends reviews on yelp.

Motivation: A person relies more on a friend's review that any unknown person.

Implementation: Our proposal is to add a filter by friend feature on the review page so that a user can directly filter and see the reviews by his friends. Since, we are also using Neo4j for our analysis, we are trying to illustrate this idea through Neo4j.

Query:

Match (a:user)-[]->(b:review)

Where a.friend_id1 = b.user_id

Return b.text



Explanation:

We are taking just two users where one is a friend of another user. We are trying to find the reviews given by the user's friend which can thus be shown on the webpage.

➤ Identifying Fake Reviews

Approach: SAS

In the age of the web, on-line reviews will create or break a business — that is why some businesses supply free food or alternative rewards in exchange for positive reviews, or perhaps pay shady firms to come up with reviews for them. A CBS News report sheds light on how Yelp is continuing the fight against fraudulent reviews. 90% users feel that reviews impact their decisions. Roughly 16% of the restaurants reviews on Yelp are fake, according to a 2013 study. And upto 15% of all online reviews are fake. Consumer confidence in reviews has taken a major hit, which has put the marketers in a tough position.

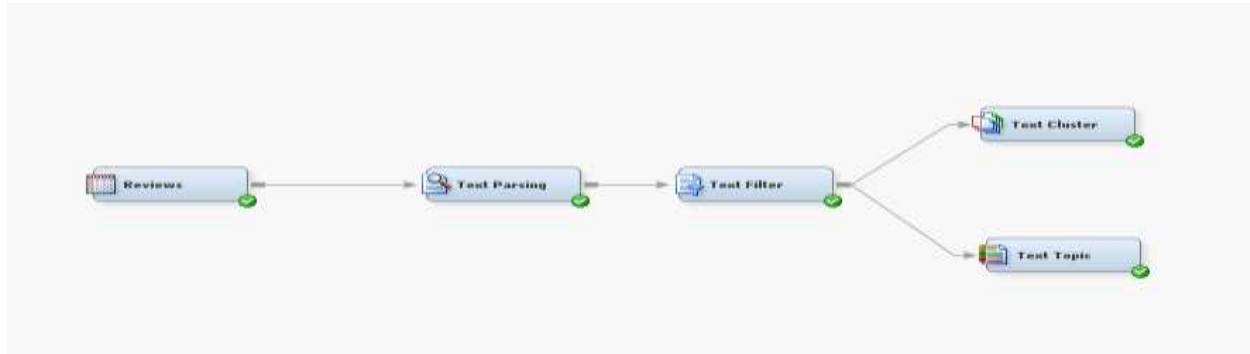
In SaS based analysis, we have proposed and derived a text-based definition to identify the fake reviews. For our analysis we have chosen the reviews dataset.

▪ Steps

1. Firstly, we converted the dataset that was in JSON format to CSV format. After that, we exported a subset of the dataset available in CSV format into sas7bdat (SaS) dataset format for our analysis.
2. Secondly, we came up with 2 approaches for our proposed solution:

Approach 1: Uncheck the words that seem relevant which will contribute to real reviews. This means that the words that we keep will help us find out fake reviews based on the clusters and topics of documents that we get.

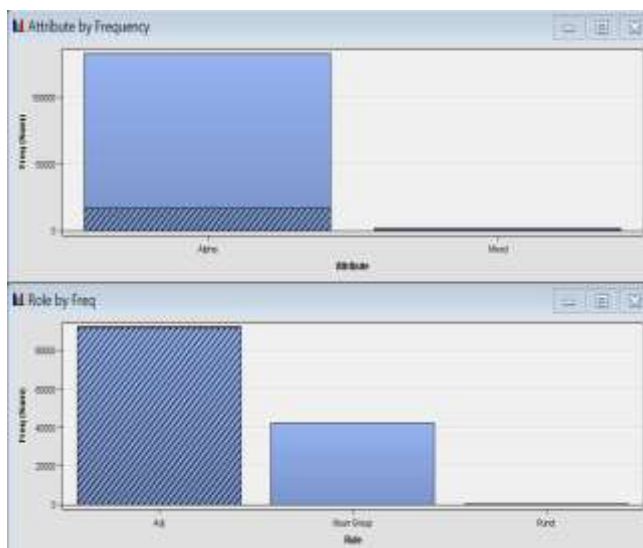
Approach 2: Just select the adjectives from the dataset and do analysis on those words. Words that show hyper emotions and if they appear frequently in a given document, then those reviews will be considered as fake reviews.



We followed steps such as:

- a) Text Parsing
- b) Text Filter
- c) Text Cluster
- d) Text Topic

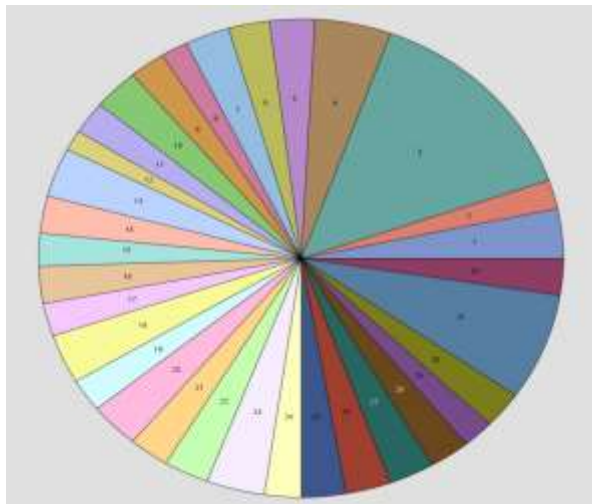
- **Text Parsing:** In this part, we took only the adjectives for our fake reviews analysis.



- **Text Filter:** In this section, we decided on the words that according to us will contribute to the fake reviews on Yelp. Our analysis was that words like excellent, pretty darn good, terrible, horrible etc. could be a part of fake reviews (in short words representing extreme or hyper emotions).



- **Text Cluster:** In the clustering part, we analyzed the clusters along with the cluster frequency of the words appearing in the documents. It gives different number of clusters for words with similar meaning. In our analysis, we got 32 different clusters.

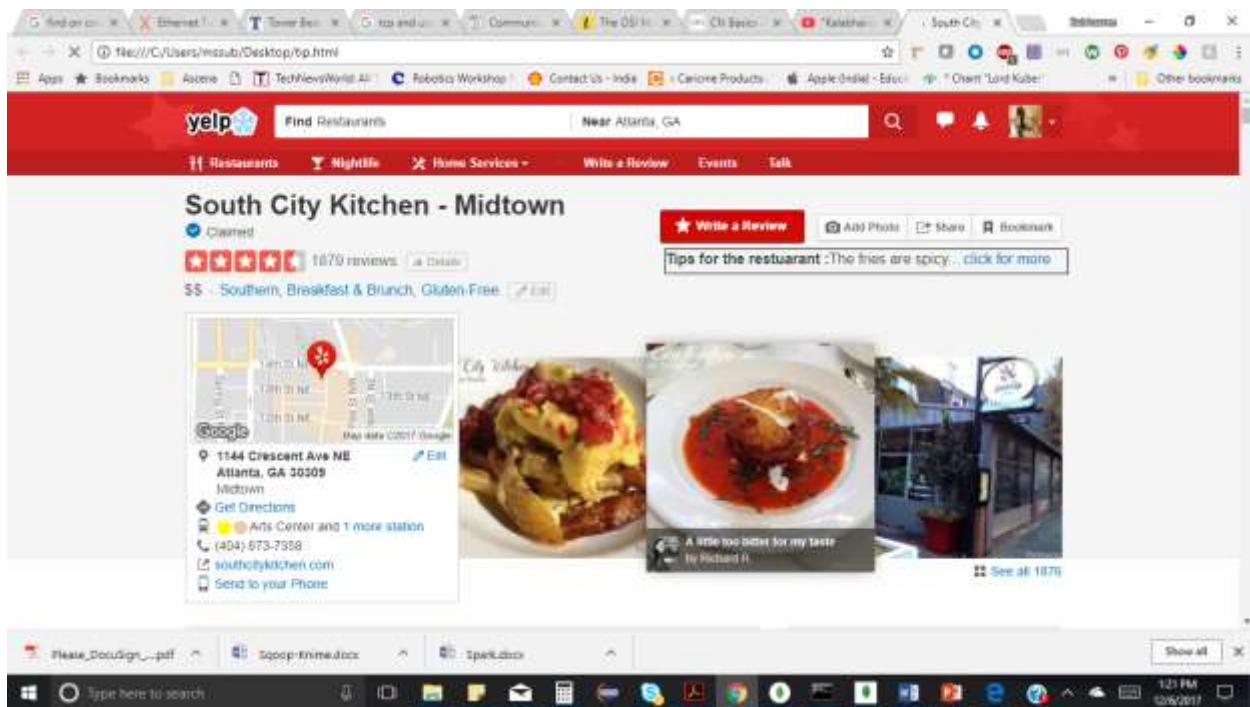


WEBPAGE DESIGN

We have used static web pages and made some enhancements to showcase some of the problems listed in the previous sections.

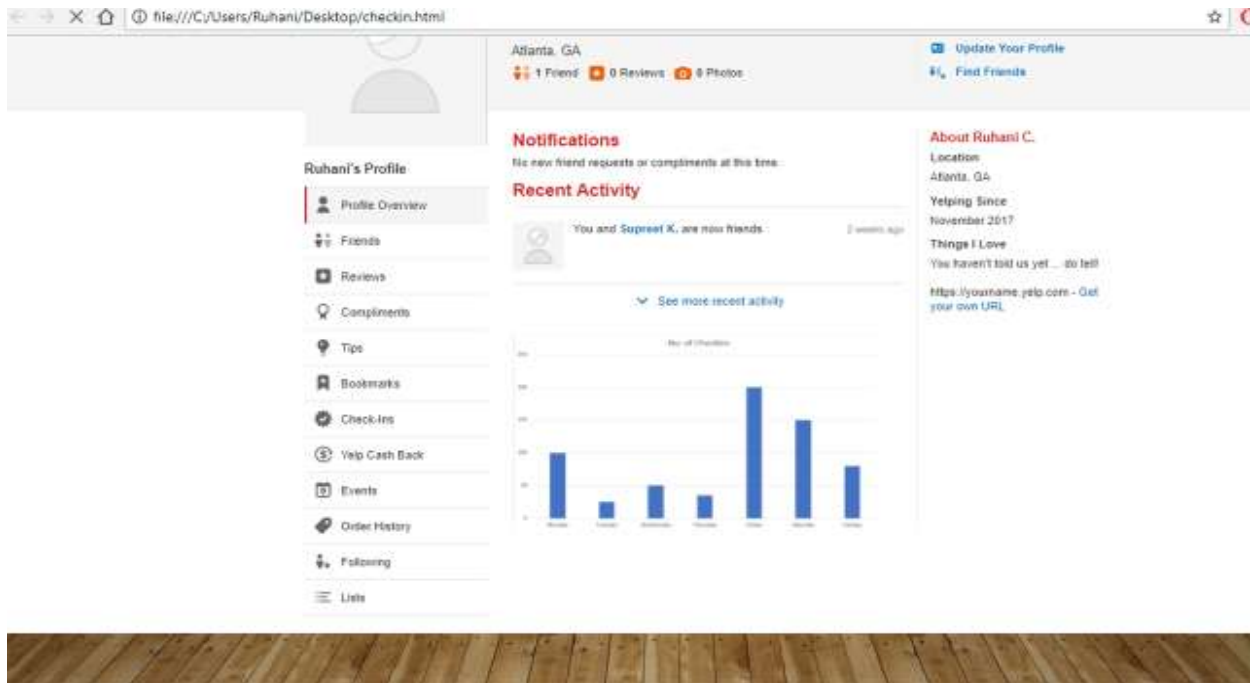
(a) Tips for a business

We will be displaying tips section for the restaurant just below “Write a Review”. A single tip in the list will be displayed by slicing the tips we got in the previous query. There is a link “click for more” added in the section to navigate to a new screen to check for all tips. In this way, it will be easier for a user to just go through a limited and concise text to gather brief information about the business.



(b) Metrics based on check-ins

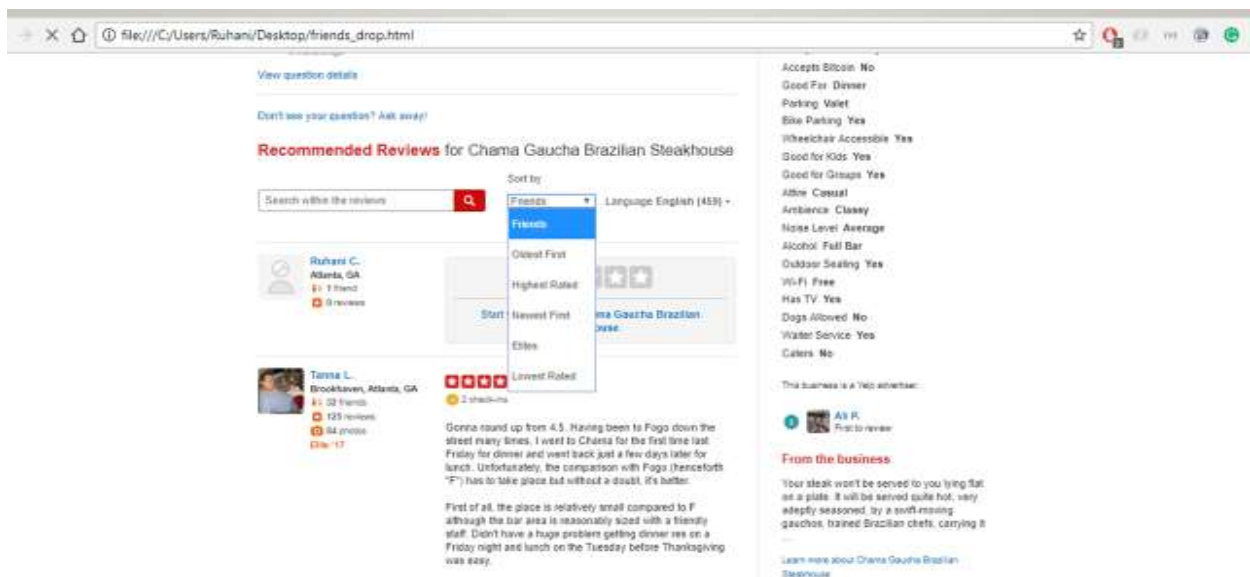
In this page, we will be creating a new web page for a business and will display the check-ins based on the days to determine the traffic during a particular hour in a day.



We have displayed only one metrics in this web page design but we can also include other metrics in this page to analyze the top 10 prolific reviewers, top 10 recent reviewers to analyze the growth of the business.

(c) Sort by friends

Some of the reviews in the review list for a business can be fake. So generally, Users trust the reviews given by friends or some known person. We have added a new field in the sort by drop down list. Only the reviews given by the friends of users will be displayed once it is clicked.



(d) Filter by feature

In this design, we have added the filters in the reviews section of the business. Filters can be selected such as: ambience, price, quality, taste or whatever feature a user is interested in. In this way, if a user is looking for reviews on the basis of price, only that reviews will be displayed.

