

YELP Dataset Based Review Webpage Proposal for Businesses and Consumers CIS8045 – Term Project

CIS8045 – TERM PROJECT

Shiva Vaidyanathan, Gopika Gopi Krishnan, Pavan C. Moganti,
Mythri Gopluni, Santhana Krishnan Pattabiraman

J. Mack Robinson College of Business | Georgia State University

YELP Dataset Based Review Webpage Proposal for Businesses and Consumers

CIS8045 – Term Project

Introduction

“The local search service, YELP, is looking to implement improvements and changes to its review dashboards for its consumers and business owners that provide deeper insight into their respective performance and activity on the YELP website.”

The above statement is taken as the premise in order to complete a set of tasks for the term project in the course CIS 8045 Unstructured Data Management. This project focuses on implementing various concepts of unstructured data management that were covered in the course to give students the practical applications of different data technologies.

For the purpose of this project, the YELP dataset provided online by YELP for its yearly challenge is utilized. This dataset provided by Yelp includes information about the member, business (registered on yelp), reviews, photos, tips and check-in. Some of the tasks included in the project are preparing pseudo database schemas for MongoDB and Neo4j taking into consideration any factors that could affect the schema, understanding the dataset by running certain queries on MongoDB, use these findings to decide certain metrics that could be useful while designing a webpage and running queries on those metric in both MongoDB and Neo4j, and finally designing a webpage for the business owner and consumer where there would be able to find review related information in Yelp.

Database Design

In any database system, the design or a basic schema of how data is going to be stored in the database is important. The schema provides information on what type of information can be stored in what field so that there is a uniformity that is maintained in the database. Even in unstructured data it is important that data is stored uniformly across collections which is especially important when querying data from the database.

A pseudo schema gives the database engineer a pattern to follow while creating a database. This schema also includes the different constraints and indexing that may be involved in the database to ensure the uniformity.

MongoDB - Pseudo Schema

Here, according to the team's understanding of how a review system database could be best implemented in MongoDB, the following schema is what was finalized for the this purpose.

On Yelp, broadly there are two types of participants involved with a review, the business owner and the consumer/user. These are the two collections that were chosen to be created to store information pertaining to the review system.

Collection creation with constraints

```
db.createCollection("YelpBusiness",{
    validator:{
        $and:[
            { "stars": { $gte : 1 ,$lte: 5}},
            { "is_open": { $in: [ 0,1 ] }},
            {"attributes.RestaurantsPriceRange2": {$in:[1,2,3,4]}}
        ]
    }
})
```

Firstly, the collection “YelpBusiness” is created and a validator is included to ensure that the field ‘stars’ is between 1 and 5 inclusive to ensure no invalid ratings are entered, has a ‘is_open’ binary value of either 0 or 1, where zero indicates not open and one indicates open, and the business attribute ‘RestaurantsPriceRange2’ which should have a value in the range 1 to 4, where 1 indicates least expensive and 4 indicates most expensive.

```
db.createCollection("YelpUsers",{
    validator:{
        "$and" : [
            { "average_stars": { $gte : 1 , $lte : 5},
            { "name": {
                "$type" : "string",
                "$exists" : "true"
            }},
            {"emailid":{
                "$type" : "string",
                "$exists" : "true"
            }},
            {"password" : {
```

```
        "$type" : "string",
        "$exists" : "true"
    }},
    {"zipcode" : {
        "$type" : "string",
        "$exists" : "true"
    }}}
]}))
```

Now, the collection “YelpUsers” is created and the constraints applied using the validator are to limit the ‘average_starts’ field to be either greater than or equal to 1 or less than or equal to 5. The fields ‘name’, ‘emailid’, ‘password’, ‘zipcode’ are all mandatory fields and their data types have also been specified, as they play an important part in creation of a user account and identification in the future.

Indexing

Indexing is a useful technique especially in databases it improves the efficiency of query from the database. Below is the indexing command for the pseudo schema created.

```
db.YelpBusiness.ensureIndex({ business_id: 1},  
{ background: true });  
  
db.YelpUsers.ensureIndex({user_id: 1},  
{ background: true });
```

Document presentation

To understand how data would be stored in the database, below are two examples, one each for each of the created collection, to give a better understanding of what data can be stored in the collections.

YelpBusiness

```
db.YelpBusiness.insert
(
[
{
  "_id": "5bf4978b37605d2facc4acbb",
  "business_id": "Aprn5Q_b6Nz61Tq4XzPdf9A",
  "name": "Minhas Micro Brewery",
  "neighborhood": "",
  "address": "1314 44 Avenue NE",
  "city": "Calgary",
  "state": "AB",
  "postal_code": "T2E 6L6",
  "latitude": 51.0918130155,
  "longitude": -114.031674872,
  "stars": 4,
  "review_count": 24.
```

```

"is_open": 1,
"attributes": {
    "BikeParking": "False",
    "BusinessAcceptsCreditCards": "True",
    "BusinessParking": "{garage': False, 'street': True, 'validated': False, 'lot': False, 'valet': False}",
    "GoodForKids": "True",
    "HasTV": "True",
    "NoiseLevel": "average",
    "OutdoorSeating": "False",
    "RestaurantsAttire": "casual",
    "RestaurantsDelivery": "False",
    "RestaurantsGoodForGroups": "True",
    "RestaurantsPriceRange2": "2",
    "RestaurantsReservations": "True",
    "RestaurantsTakeOut": "True"
},
"categories": "Tours, Breweries, Pizza, Restaurants, Food, Hotels & Travel",
"hours": {
    "Monday": "8:30-17:0",
    "Tuesday": "11:0-21:0",
    "Wednesday": "11:0-21:0",
    "Thursday": "11:0-21:0",
    "Friday": "11:0-21:0",
    "Saturday": "11:0-21:0"
},
"checkinTime": {
    "Fri-0": 2,
    "Sat-0": 1,
    "Sun-0": 1,
    "Wed-0": 2
},
"photos": [
    {
        "photo_id": "luXwafFH3fZlTyXA-poz0w",
        "caption": "",
        "label": "food"
    }
]
}
)

```

YelpUsers

```

db.YelpUsers.insert
(

```

```
[
  {
    "_id": "5bf49c3a37605d2fac230f46",
    "user_id": "IzlZwlpuSWXEnNS91wxjHw",
    "name": "Susan",
    "emailid": "susan@hotmail.com"
    "password": "#####
    "zipcode" :"40005AB"
    "review_count": 1,
    "yelping_since": "2015-09-28",
    "friends": "None",
    "useful": 0,
    "funny": 0,
    "cool": 0,
    "fans": 0,
    "elite": "None",
    "average_stars": 2,
    "compliment_hot": 0,
    "compliment_more": 0,
    "compliment_profile": 0,
    "compliment_cute": 0,
    "compliment_list": 0,
    "compliment_note": 0,
    "compliment_plain": 0,
    "compliment_cool": 0,
    "compliment_funny": 0,
    "compliment_writer": 0,
    "compliment_photos": 0,
    "reviews": [
      {
        "review_id": "x7mDliDB3jEiPGPHOmDzyw",
        "user_id": "msQe1u7Z_XuqjGoqhB0J5g",
        "business_id": "iCQpiavjjPzJ5_3gPD5Ebg",
        "stars": 2,
        "date": "2011-02-25",
        "text": "The pizza was okay. ",
        "useful": 0,
        "funny": 0,
        "cool": 0
      },
      {},
      {}
    ],
    "tips": [
      {

```

```

        "text": "Family style dining. Known for their super burger and strawberry pie.",
        "date": "2013-05-28",
        "likes": 0,
        "business_id": "siHpyAfAnoBMPpl9d_ib3A",
        "user_id": "fmzIm7RxEdii5Jz44PtO7g"
    }
]
}
]
)

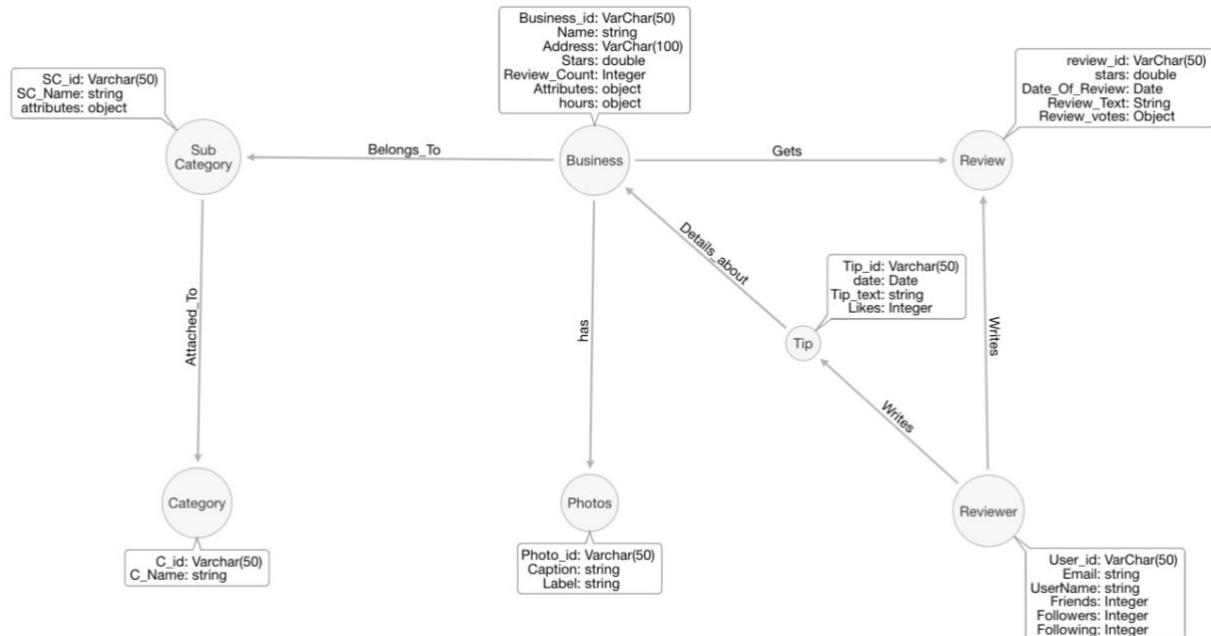
```

Neo4j – Pseudo Schema

This section shall cover the preparation of a pseudo schema for a graph database system for the Yelp review system. A graph data model is slightly different from the data models that have been covered so far and has nodes. These nodes have a relationship among each other and the relationships have properties which make up the characteristics of the model.

For the purpose of designing this model, we are identifying the following nodes.

Nodes identified are: Business, Review, Reviewer, Photos, Member, Tip, Category and Sub category. The nodes each have certain properties which identify them.



The nodes have certain relationships along with properties which are identified in bold as follows:

1. Business **Gets** a Review

2. Reviewer **Writes** a Review.
3. Reviewer also **Writes** a Tip
4. A Tips **details_about** Business
5. A Business **has** Photos.
6. A Business also **belongs_to** a Sub-Category
7. A Sub-category is **attached_to** a Category

Characteristics of the Graph Model

The order of a graph model is defined by the total number of nodes in the model, which in this case is seven. The size of the graph model designed is seven, this is characterized by the total number of relationships in the model.

Every node has a certain number relationships attached to it which is known as its degree. Following are the degree of the different nodes in the model prepared.

- Business node has a degree of 4.
- Review node has a degree of 2.
- Tip node has a degree of 2.
- Reviewer node has a degree of 2.
- Photos node has a degree of 1.
- Sub-Category node has a degree of 2.
- Category node has a degree of 1.

Diagram description (TBC)

The designed graph data model helps us identify each Business and the different relationships it has. The Reviewer and the Business are connected but not directly. They are connected in two ways, one via the Review node and the other via the Tip node. In a review system, every establishment does also have Photos of the business. Also a business just existing without being classified would mean that is less organized, for that the business is attached to a Sub-category of business which are attached to a Category. In this manner, the business is also connected to a category. This pseudo data model gives an overall idea of the Review System in the Yelp database. The graph model in the document is a directed graph where the relationship between each node is directed towards another node.

Data Evaluation

The dataset provided by Yelp that is going to be used for the purpose of this project consists of six collections. The collections have data about the following.

1. Business

This has information about the businesses registered on yelp. The data includes the business id, name, address, ratings, number of reviews, the attributes (wheelchair access, dogs allowed, Wi-fi, etc), categories the business belongs to, hours of operation of the business which are the prominent fields.

2. User

Information about the users registered on Yelp can be found in this set. Some of the data fields include user id, name, number of reviews, start date of reviewing, friends, average stars and many compliment fields (writer, plain, note, cool, etc.) among others.

3. Review

Here all the information about reviews given to a business can be found. The data has the fields review id, user id (User who has written the review), business id (business being reviewed), stars, date of review, text content and some other features like useful, cool and funny.

4. Tip

Users have the option to provide tips about a certain businesses which is stored in this set. Here the fields include the user and business ids, the tip text, date the tip was written and likes.

5. Check-in

This a relatively simpler set of data consisting of the business id and a time object with details of the time checked in.

6. Photo

Here the information on photos attached to a business are stored. The set includes the photo id, business id, a caption and label.

To understand the data, it would be impossible to go through all of the dataset and extract an overview. There to understand the data better, loading the datasets in MongoDB and running a few queries would help us understand what data we are dealing with and comprehend the results the queries are providing and if there are useful or not. Below are some of the questions that were compiled and to get the results, the queries were run on MongoDB after the datasets were loaded into the database.

1. What is the overall number of businesses in Cleveland and Belmont

Query for Cleveland:

Query Description: This query returns number of unique businesses for city Cleveland. Since Cleveland can have multiple units of same business, we are grouping based on city and adding unique business_id to set which is being used in projection to fetch the size.

```
db.yelpBusiness.aggregate([
  {$match:{city:{$eq:"Cleveland}}},
  {$group:{_id:"$city",uniquebusiness:{$addToSet:"$business_id}}},
  {$project:{_id:"$_id",businessCount:{$size:"$uniquebusiness}}},
  {$sort:{businessCount:-1}}
])
```

Key	Value	Type
(1) Cleveland	<code>_id</code> : Cleveland <code>businessCount</code> : 3506	<code>_id</code> : String <code>businessCount</code> : Int32

Summary

On running the query we understand that Cleveland has a total of 3506 businesses

Query for Belmont:

Description: This query returns number of unique businesses for city Belmont. Since Belmont can have multiple units of same business, we are grouping based on city and adding unique business_id to set which is being used in projection to fetch the size.

```
db.yelpBusiness.aggregate([
  {$match:{city:{$eq:"Belmont}}},
  {$group:{_id:"$city",uniquebusiness:{$addToSet:"$business_id}}},
  {$project:{_id:"$_id",businessCount:{$size:"$uniquebusiness}}},
  {$sort:{businessCount:-1}}
])
```

```

db.yelpBusiness.aggregate([
  {$match: {city: {$eq: "Belmont"} } },
  {$group: {_id: "$city", uniquebusiness: {$addToSet: "$business_id"} } },
  {$project: {_id: "$_id", businessCount: {$size: "uniquebusiness"} } },
  {$sort: {businessCount: -1} }
])

```

Key	Value	Type
(1) Belmont	{ _id: Belmont, businessCount: 158 }	Object
_id	Belmont	String
businessCount	158	Int32

Summary Results:

On running the query we understand that Belmont has a total of 158 businesses .

2. The overall number of reviews per business

Query for Cleveland

Description: This Query returns overall number of reviews per business in Cleveland . Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. As one business_id may have multiple reviews, \$group is performed on business_id and count is calculated for each business_id

```

db.yelpBusiness.aggregate([
  {
    $lookup: {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    }
  },
  {
    $unwind: "$yelpReview"
  },
  {
    $match:
  }
])

```

```

        {"city":{$eq:"Cleveland"}}
    }, {
        $group:{
            "_id" : "$business_id",
            count:{$sum:1}
        }
    })
])

```

The screenshot shows the Robo 3T interface with the following details:

- File**, **View**, **Options**, **Window**, **Help** menu at the top.
- LocalDB (5)** sidebar with collections: System, config, handson, termProject, yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, yelpUser, Functions, Users (0).
- localhost:27017** and **termProject** tabs open.
- Term Project** tab content shows the aggregate query:

```

        foreignField: "business_id",
        as: "yelpReview"
    },
    $unwind:"$yelpReview" ),
    {
        $match:
            {"city":{$eq:"Cleveland"}}
    },
    {
        $group:{
            "_id" : "$business_id",
            count:{$sum:1}
        }
    }
])

```
- yelpBusiness** table below the query shows one document:

Key	Value	Type
_id	n2lge2of0PFdQ54v1a9TtG	String
count	8.0	Double
(2) 70uY7NAB1KuC9cYn8He3Pw	{ 2 fields }	Object
- Logs** tab at the bottom left.
- Windows taskbar at the bottom with various icons and the date/time: 7:07 AM 12/5/2018.

Summary of Results:

The result shows the count of the total reviews per business as displayed. A sample output is included for the city of Cleveland.

Query for Belmont

Description: This Query returns overall number of reviews per business in Belmont . Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. As one business_id may have multiple reviews, \$group is performed on business_id and count is calculated for each business_id.

```

db.yelpBusiness.aggregate(
{
    $lookup:{
        from: "yelpReview",
        localField: "business_id",

```

```

        foreignField: "business_id",
        as: "yelpReview"
    }
},
{ $unwind:"$yelpReview" },
{
$match:
  {"city":{$eq: "Belmont"}}
},
{
  $group:{
    "_id" : "$business_id",
    count:{$sum:1}
  }
}
])

```

The screenshot shows the Robo 3T interface with the following details:

- File**, **View**, **Options**, **Window**, **Help** menu.
- LocalDB (5)** database list on the left, including System, config, handson, termProject, and collections: yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, yelpUser.
- Welcome** tab showing the aggregate query: `db.yelpBusiness.aggregate([...])`.
- localhost:27017** and **termProject** connection status.
- Result Preview** pane showing the query results:


```

from: "yelpReview",
localField: "business_id",
foreignField: "business_id",
as: "yelpReview"
},
{
  $unwind:"$yelpReview" ,
{
$match:
  {"city":{$eq : "Belmont"}}
},
{
  $group:{
    "_id" : "$business_id",
    count:{$sum:1}
  }
}
])

```
- Logs** tab at the bottom-left.
- Windows taskbar at the bottom-right with icons for File Explorer, Google Chrome, Microsoft Word, Microsoft Excel, and others.

Summary Results:

The result shows the count of the total reviews per business as displayed. A sample output is included for the city of Belmont.

3. The overall number of reviewers per business.

Description: This Query returns overall number of unique reviewers per business in Cleveland. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness

and yelpReview using the common attribute business_id. As one business_id may have multiple reviewers, \$group is performed on business_id and unique user_id(reviewers id) is collected in a set. The size of the set will give us unique reviewer count.

Query for Cleveland

```
db.yelpBusiness.aggregate([
{   $lookup:{ from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
},
{   $unwind:"$yelpReview" },
{
  $match:
  {"city":{$eq:"Cleveland"}}
},
{
  $group:{
    "_id" : "$business_id",
    "uniqueUserID":{$addToSet:"$yelpReview.user_id"}
  }
},
{
  $project:{
    _id:"$_id",
    "reviewerCountPerBusiness":{$size:"$uniqueUserID"}
  }
}
])
```

The screenshot shows the Robo 3T interface. On the left, the sidebar displays a database named 'LocalDB' with a collection named 'yelpBusiness' selected. The main pane shows an aggregation pipeline:

```

db.yelpBusiness.aggregate([
  {
    $unwind: "$yelpReview",
    $match: {
      "city": {$eq: "Cleveland"}
    },
    $group: {
      "_id": "$business_id",
      "uniqueUserID": {$addToSet: "$yelpReview.user_id"}
    },
    $project: {
      _id: "$_id",
      "reviewerCountPerBusiness": {$size: "$uniqueUserID"}
    }
  }
])
  
```

The results table below the pipeline shows one document with the key '_id' and value 'n2lge2oFoPfdQS4v1a9TTg'. The 'reviewerCountPerBusiness' field has a value of 8.

Key	Value	Type
_id	{ 2 fields }	Object
_id	n2lge2oFoPfdQS4v1a9TTg	String
reviewerCountPerBusiness	8	Int32

Summary of Results:

On running the query we see the overall count of reviewers per business for the city of Cleveland.

Query for Belmont

Description: This Query returns overall number of unique reviewers per business in Belmont. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. As one business_id may have multiple reviewers, \$group is performed on business_id and unique user_id (reviewer id) is collected in a set. The size of the set will give us unique reviewer count.

```

db.yelpBusiness.aggregate([
  {
    $lookup: {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    },
    {
      $unwind: "$yelpReview"
    },
    {
      $match: {
        "city": {$eq: "Belmont"}
      }
    },
    {
      $group: {
        "_id": "$business_id",
        "uniqueUserID": {$addToSet: "$yelpReview.user_id"}
      }
    },
    {
      $project: {
        _id: "$_id",
        "reviewerCountPerBusiness": {$size: "$uniqueUserID"}
      }
    }
])
  
```

```

        "uniqueUserID":{$addToSet:"$yelpReview.user_id"}
    }
},
{
    $project:{  

        _id:"$_id",  

        "reviewerCountPerBusiness":{$size:"$uniqueUserID"}  

    }
}
])

```

The screenshot shows the Robo 3T interface with the following details:

- File Menu:** File, View, Options, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and Database selection.
- Left Sidebar:** Shows the database structure under LocalDB (5). It includes System, config, handson, termProject, and its collections: yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, yelpUser, Functions, and Users (0).
- Central Query Editor:** Displays the MongoDB aggregate query:

```

db.yelpBusiness.aggregate([
    {
        $lookup: {
            from: "yelpReview",
            localField: "business_id",
            foreignField: "business_id",
            as: "yelpReview"
        }
    },
    {
        $unwind: "$yelpReview" ,
    },
    {
        $match:
            {"city":{$eq:"Belmont"}}
    },
    {
    }
])

```
- Bottom Results Table:** Shows the results of the query for the document with _id A0FLFsqkGK59l2WPhi93mg.

Key	Value	Type
(1) A0FLFsqkGK59l2WPhi93mg	{ 2 fields }	Object
_id	A0FLFsqkGK59l2WPhi93mg	String
reviewerCountPerBusiness	8	Int32
- System Tray:** Shows the date and time (7:11 AM, 12/5/2018) and system icons.

Summary Results:

On running the query we see the overall count of reviewers per business for the city of Belmont.

5.a) Overall count of reviewers.

Query for Cleveland

Description: This Query returns overall number of unique reviewers in Cleveland. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. As one user_id can give multiple reviews, we are collecting user_id in set and count of the set will give the number of unique reviewers.

```

db.yelpBusiness.aggregate([
  {
    $lookup: {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    }
  },
  {
    $unwind: "$yelpReview",
    {
      $match:
        {"city":{$eq: "Cleveland"}}
    },
    {
      $group: {
        "_id":null,
        "uniqueUserID":{$addToSet:"$yelpReview.user_id"}
      }
    },
    {
      $project: {
        _id:"count",
        "reviewerCountPerBusiness":{$size:"$uniqueUserID"}
      }
    }
  }
])

```

The screenshot shows the Robo 3T interface. On the left, the sidebar displays the database structure: LocalDB (5) with System, config, handson, and termProject collections. Under termProject, there are Collections (6): yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, and yelpUser. Below these are Functions (0) and Users (0). The main pane shows an aggregation query:

```

        as: "yelpReview"
    },
    {
        $unwind: "$yelpReview" ,
    {
        $match:
            {"city": {$eq:"Cleveland"}}
    },
    {
        $group:
            {"_id":null,
            "uniqueUserID":{$addToSet:"$yelpReview.user_id"}
            }
    },
    {
        $project:
    }

```

The results pane shows the output of the query:

Key	Value	Type
count	{ 2 fields }	Object
_id	count	String
reviewerCountPerBusiness	39858	Int32

The status bar at the bottom indicates the operation took 1.95 seconds.

Summary Results:

On running the query we get the overall number of unique reviewers in the whole city of Cleveland.

Query for Belmont:

Description: This Query returns overall number of unique reviewers in Belmont. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. As one user_id can give multiple reviews, we are collecting user_id in set and count of the set will give the number of unique reviewers.

```

db.yelpBusiness.aggregate([
{
    $lookup:{
        from: "yelpReview",
        localField: "business_id",
        foreignField: "business_id",
        as: "yelpReview"
    }
},
{ $unwind:"$yelpReview" },
{
    $match:

```

```

        {"city":{$eq: "Belmont" }}

    },
    {
        $group:{

            "_id":null,
            "uniqueUserID":{$addToSet:"$yelpReview.user_id"}
        }
    },
    {
        $project:{

            _id:"count",
            "reviewerCountPerBusiness":{$size:"$uniqueUserID"}
        }
    }
])

```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the database structure under 'LocalDB' (5 collections: System, config, handson, termProject, yelpBusiness). The 'yelpBusiness' collection is selected. The main area contains the aggregate query:

```

db.yelpBusiness.aggregate([
    {"city":{$eq: "Belmont" }},

    {
        $group:{

            "_id":null,
            "uniqueUserID":{$addToSet:"$yelpReview.user_id"}
        }
    },
    {
        $project:{

            _id:"count",
            "reviewerCountPerBusiness":{$size:"$uniqueUserID"}
        }
    }
])

```

Below the query, the results table shows one document:

Key	Value	Type
(1) count	{ _id: count, reviewerCountPerBusiness: 1991 }	Object
_id	count	String
reviewerCountPerBusiness	1991	Int32

Summary Results:

On running the query we get the overall number of unique reviewers in the whole city of Belmont.

6. The overall number of reviews with ratings more than 3

Description:

This Query returns overall number of reviews with stars greater than 3 in Cleveland. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. \$group is used to collect business_id in a set and count of the results is calculated using \$sum.

Query for Cleveland:

```
db.yelpBusiness.aggregate([
  {$lookup:
    {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    }
  },
  {$unwind:"$yelpReview" },
  { $match:
    { "$and": [{"city": "Cleveland"}, {"yelpReview.stars": {
      "$gt":3
    }}]
  }
  },
  { $group: {
    "_id": "",
    "business_idSet": {"$addToSet": "$business_id"},
    "count": {
      "$sum":1
    }
  }
  },
  {
    $project:{ 
      _id : null,
      "business_idSet": "$business_idSet",
      records:"$count"
    }
  }
])
```

The screenshot shows the Robo 3T interface. On the left, the sidebar displays a database named 'LocalDB' with several collections: System, config, handson, termProject, yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, and yelpUser. The 'yelpBusiness' collection is selected. The main pane shows an aggregation pipeline:

```

from: "yelpReview",
localField: "business_id",
foreignField: "business_id",
as: "yelpReview"
},
{
$unwind: "$yelpReview" ,
{
$match:
{
"$and": [
{"city": "Cleveland"},
{"yelpReview.stars": {
"$gt": 3
}}
]
}
},
{
$group: {
"_id": "",
"business_idSet": {"$addToSet": "$business_id"},
"records": {"$sum": 1}
}
}
]
}

```

The results table below the pipeline shows the following data:

Key	Value	Type
<code>_id</code>	null	Null
<code>business_idSet</code>	[3307 elements]	Array
<code>records</code>	69279.0	Double

The status bar at the bottom indicates the operation took 1.74 seconds.

Summary Results:

This query gives us a count of all reviews for all the businesses with a star rating more than 3 for the city of Cleveland.

Query for Belmont

Description:

This Query returns overall number of reviews with stars greater than 3 in Belmont. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. \$group is used to collect business_id in a set and count of the results is calculated using \$sum.

```

db.yelpBusiness.aggregate([
{$lookup:
{
from: "yelpReview",
localField: "business_id",
foreignField: "business_id",
as: "yelpReview"
}
}],
)

```

```

{$unwind:"$yelpReview" },
{ $match:
  { "$and" :[{"city": "Belmont"},
    {"yelpReview.stars":{
      "$gt":3
    }}]
  }
}

},
{ $group: {
  "_id": "",
  "business_idSet":{"$addToSet":"$business_id"}, 
  "count":{ 
    "$sum":1
  }
}
},
{
  $project:{ 
    _id : null,
    "business_idSet":"$business_idSet",
    records:"$count"
  }
}
])

```

The screenshot shows the Robo 3T interface with the following details:

- File Menu:** File, View, Options, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and a search bar labeled "Type here to search".
- Left Sidebar:** Shows the database structure under "LocalDB (5)". It includes collections: yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, yelpUser, Functions, and Users (0).
- Middle Panel:** Displays the MongoDB query in the "Welcome" tab:

```

db.yelpBusiness.aggregate([
  {$lookup:
    {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    }
  },
  {$unwind:"$yelpReview" },
  { $match:
    { "$and" :[{"city": "Belmont"},
      {"yelpReview.stars":{
        "$gt":3
      }}]
    }
  }
])

```
- Bottom Panel:** Shows the execution results for the "yelpBusiness" collection. The results table has columns: Key, Value, and Type. The data is:

Key	Value	Type
_id	null	Null
business_idSet	[152 elements]	Array
records	1955.0	Double
- System Bar:** Shows the date and time (7:16 AM 12/5/2018) and other system icons.

Summary Results:

This query gives us a count of all reviews for all the businesses with a star rating more than 3 for the city of Belmont.

7. The overall number of reviews with ratings less than or equal 3

Query for Cleveland

Description:

This Query returns overall number of reviews with stars less than or equals 3 in Cleveland. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. \$group is used to collect business_id in a set and count of the results is calculated using \$sum.

```
db.yelpBusiness.aggregate([
{$lookup:
  {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  }
},
{$unwind:"$yelpReview" },
{$match:
  { "$and": [{"city": "Cleveland"}, {"yelpReview.stars": {
      "$lte":3
    }}]
}
},
{$group: {
  "_id": "",
  "business_idSet":{"$addToSet":"$business_id"},
  "count": {
    "$sum":1
  }
},
{
  "sum": {
    "$sum":1
  }
}
]);
```

```

$project:{  

    _id : null,  

    "business_idSet":"$business_idSet",  

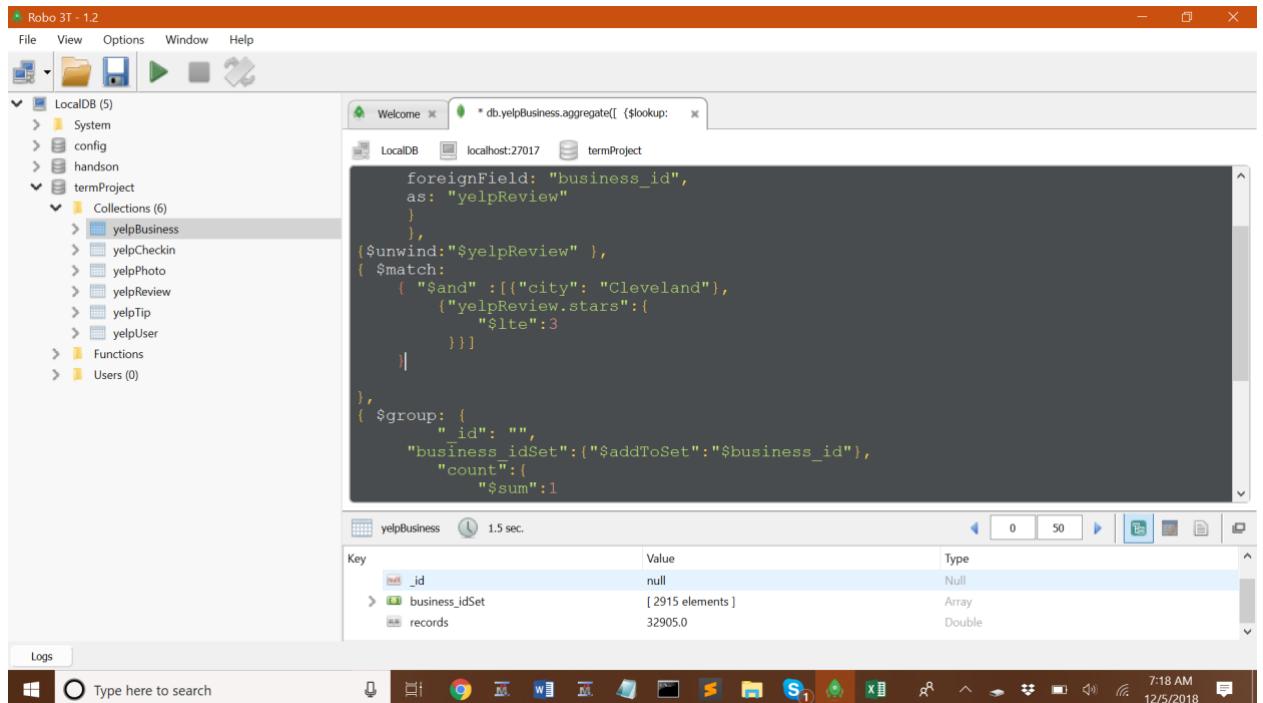
    records:"$count"  

}  

}  

])

```



The screenshot shows the Robo 3T interface with the following details:

- File Menu:** File, View, Options, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and Project.
- Left Sidebar:** Shows the database structure under LocalDB (S):
 - System
 - config
 - handson
 - termProject (selected)
 - Collections (6):
 - yelpBusiness
 - yelpCheckin
 - yelpPhoto
 - yelpReview
 - yelpTip
 - yelpUser
 - Functions
 - Users (0)
- Central Panel:** Displays the MongoDB aggregation query:


```

foreignField: "business_id",
as: "yelpReview"
},
{$unwind: "$yelpReview" },
{$match:
{
"$and": [
{"city": "Cleveland"},
{"yelpReview.stars": {
"$lte": 3
}}
]
}
},
{$group: {
"_id": "",
"business_idSet": {"$addToSet": "$business_id"},  

"count": {
"$sum": 1
}
}
}
      
```
- Bottom Panel:** Shows the results of the aggregation query in a table:

Key	Value	Type
_id	null	Null
business_idSet	[2915 elements]	Array
records	32905.0	Double
- System Tray:** Shows the date and time (7:18 AM, 12/5/2018) and other system icons.

Summary Results:

This query gives us a count of all reviews for all the businesses with a star rating less than 3 for the city of Cleveland.

Query for Belmont:

Description:

This Query returns overall number of reviews with stars less than or equals 3 in Belmont. Since, city attribute is not part of yelpReview collection, \$lookup is used to join yelpBusiness and yelpReview using the common attribute business_id. \$group is used to collect business_id in a set and count of the results is calculated using \$sum.

```
db.yelpBusiness.aggregate([
```

```
  {$lookup:
```

```
  {
```

```

from: "yelpReview",
localField: "business_id",
foreignField: "business_id",
as: "yelpReview"
}
},
{$unwind:"$yelpReview" },
{ $match:
{ "$and": [{"city": "Belmont"}},
{"yelpReview.stars":{
    "$lte":3
}}]
}
},
{ $group: {
    "_id": "",
    "business_idSet": {"$addToSet": "$business_id"},
    "count":{
        "$sum":1
    }
},
{
    $project:{
        _id : null,
        "business_idSet": "$business_idSet",
        records:"$count"
    }
}
])

```

The screenshot shows the Robo 3T interface. On the left, the sidebar displays the database structure under 'LocalDB (5)'. The 'Collections' section contains 'yelpBusiness', 'yelpCheckin', 'yelpPhoto', 'yelpReview', 'yelpTip', and 'yelpUser'. The main panel shows an aggregation pipeline in the 'Welcome' tab:

```

{
  $lookup: {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  },
  $unwind: "$yelpReview",
  $match: {
    $and: [
      { city: "Belmont" },
      { "yelpReview.stars": { $lte: 3 } }
    ]
  },
  $group: {
    _id: ""
  }
}

```

The results table below the pipeline shows one document with the following fields:

Key	Value	Type
_id	null	Null
business_idSet	[135 elements]	Array
records	958.0	Double

The status bar at the bottom indicates the operation took 0.248 seconds.

Summary Results:

This query gives us a count of all reviews for all the businesses with a star rating less than 3 for the city of Belmont.

8. The average number of reviews per business

Query for Cleveland:

Query Description :

This query gives the average number of reviews per business .We are using lookup to get the city form the yelpBusiness .

```

db.yelpBusiness.aggregate([
{
  $lookup: {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  }
},
{ $unwind: "$yelpReview" },
{
  $match:

```

```

        {"city":{"$eq":"Cleveland"}}
    },
    {
        $group:{ "_id" : "$business_id",
        count:{$sum:1}
    }
},
{ "$group" :
    { "average" :
        { "$avg" : "$count" },
        "_id" : "AverageNumberOfReviews"
    }
}
])

```

The screenshot shows the Robo 3T interface with the following details:

- File Menu:** File, View, Options, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and Close, as well as a search bar.
- Left Sidebar:** Shows the database structure under LocalDB (5). It includes System, config, handson, and termProject collections. termProject contains Collections (6) with yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, and yelpUser.
- Central Area:** Displays the MongoDB aggregation pipeline. The pipeline consists of several stages:
 - \$unwind: "\$yelpReview"
 - \$match: {"city": {"\$eq": "Cleveland"}}
 - \$group: { "_id": "\$business_id", count: {\$sum: 1} }
 - { "\$group" : { "average" :
 - { "\$avg" : "\$count" },
 - _id: "AverageNumberOfReviews"
- Bottom Panel:** Shows the results of the aggregation. A table titled "yelpBusiness" displays one document with key-value pairs:

Key	Value	Type
\$_id	AverageNumberOfReviews	String
average	29.1454649172847	Double

Results Summary :

We get an overall idea of the number of reviews each business has in the city of Cleveland.

Query for Belmont:

Query Description :

This query gives the average number of reviews per business .We are using lookup to get the city form the yelpBusiness .

```

db.yelpBusiness.aggregate([
  {
    $lookup: {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    }
  },
  { $unwind:"$yelpReview" },
  {
    $match:
      {"city":{"$eq":"Belmont"}}
  },
  {
    $group:{
      "_id" : "$business_id",
      count:{$sum:1}
    }
  },
  { "$group" :
    { "average" :
      { "$avg" : "$count"},
      "_id" : "AverageNumberOfReviews"
    }
  }
])

```

The screenshot shows the Robo 3T interface. On the left, the sidebar displays a database named 'LocalDB' with several collections: System, config, handson, termProject, yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, yelpUser, Functions, and Users. The 'yelpBusiness' collection is selected. The main pane shows an aggregation pipeline. The results table below shows one document with fields '_id' and 'average'. The bottom status bar indicates the operation took 0.243 seconds.

```

db.yelpBusiness.aggregate([
  {
    $unwind: "$yelpReview"
  },
  {
    $match: {
      "city": {"$eq": "Belmont"}
    }
  },
  {
    $group: {
      "_id": "$business_id",
      count: {"$sum": 1}
    }
  },
  {
    "$group": {
      "average": {
        "average": "$count"
      }
    }
  }
])
  
```

Key	Value	Type
(1) AverageNoOfReviews	{ _id: "AverageNoOfReviews", average: 18.436708607595 }	Object
_id	AverageNoOfReviews	String
average	18.436708607595	Double

Results Summary :

We get an overall idea of the number of reviews each business has in the city of Belmont.

9. The date of the first review per business.

Query for Cleveland:

Query Description:

This query gives the first time a business was reviewed in the city of Cleveland. We used lookup to get the city from the business collection.

```

db.yelpBusiness.aggregate([
{$lookup:{
  from: "yelpReview",
  localField: "business_id",
  foreignField: "business_id",
  as: "yelpReview"
},
{$unwind:"$yelpReview" },
{$match: {"city": "Cleveland"} },
{$sort: {"yelpReview.date":1}},
{$group:{"_id": "$business_id","oldestDate":{$first:"$yelpReview.date"}}, {"allowDiskUse":true}}
])
  
```

The screenshot shows the Robo 3T interface. On the left, the sidebar displays a database named 'LocalDB' with several collections: System, config, handson, termProject, yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, yelpUser, Functions, and Users (0). The 'yelpBusiness' collection is selected. In the main pane, a query is being run against the 'yelpBusiness' collection:

```
db.yelpBusiness.aggregate([
  {$lookup: {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  },
  {$unwind: "$yelpReview" },
  { $match: { "city": "Cleveland" } },
  { $sort: { "yelpReview.date":1 } },
  { $group:{ "_id": "$business_id", "oldestDate":{$first:"$yelpReview.date"} } },
  {"allowDiskUse":true}
])
```

The results table shows the following data:

Key	Value	Type
<code>_id</code>	AXYafJW1-xo9oRgEnrP0TQ	String
<code>oldestDate</code>	2018-06-20	String
(2) <code>2tpaEOpDgI5ExAldxrSyg</code>	{ 2 fields }	Object
<code>_id</code>	2tpaEOpDgI5ExAldxrSyg	String
<code>oldestDate</code>	2018-05-27	String
(3) <code>U5DjWjQDp4H8t4WXMi3hwA</code>	{ 2 fields }	Object
(4) <code>3Fs04jkpmjB7B2ZVj9tJg</code>	{ 2 fields }	Object

At the bottom, the status bar indicates the operation took 5.03 seconds.

Summary Results:

The oldest date of when the business was reviewed is returned for the city of Cleveland.

Query for Belmont :

Query Description:

This query gives the first time a business was reviewed in the city of Cleveland. We used lookup to get the city from the business collection.

```
db.yelpBusiness.aggregate([
  {$lookup:{
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  },
  {$unwind: "$yelpReview" },
  { $match: { "city": "Belmont" } },
  { $sort: { "yelpReview.date":1 } },
  { $group:{ "_id": "$business_id", "oldestDate":{$first:"$yelpReview.date"} } }
])
```

```
{"allowDiskUse":true}
```

The screenshot shows the Robo 3T interface. On the left, the database structure is displayed under 'LocalDB'. A 'termProject' database is selected, containing a 'yelpBusiness' collection. In the main pane, a query is being run against the 'yelpBusiness' collection:

```
db.yelpBusiness.aggregate([
  {
    $lookup: {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    }
  },
  {
    $unwind: "$yelpReview",
    $match: { "city": "Belmont" },
    $sort: { "yelpReview.date": 1 },
    $group: { "_id": "$business_id", "oldestDate": { $first: "$yelpReview.date" } },
    {"allowDiskUse": true}
])
```

The results of the aggregation are shown in a table below:

Key	Value	Type
(1) WWGeCgjOPvocbYOy4_TMnw	{ _id: WWGeCgjOPvocbYOy4_TMnw, oldestDate: 2018-01-25 }	Object
(2) l2kzkT3iw10klq2nIWRbgQ	{ _id: l2kzkT3iw10klq2nIWRbgQ, oldestDate: 2017-11-16 }	Object
(3) t2oNKePput5H5iRjOGwDQ	{ _id: t2oNKePput5H5iRjOGwDQ }	Object

Summary Results:

The oldest date of when the business was reviewed is returned for the city of Cleveland.

10. The top 10 most verbose reviewers

Query description: This query is to find the length of the longest review in terms of length for a user and display such top 10 reviews for the city of Cleveland.

Summary Results :

This query gives the top 10 reviewers from highest to lowest in the order of the count of their total number of reviews.

```
db.yelpBusiness.aggregate([
{
  $lookup:
  {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  }
})
```

```

},
{$unwind: "$yelpReview" },
{$match: {"city": "Cleveland"}},
{$project : 
{"business_id":1,"review_id": "$yelpReview.review_id","user_id": "$yelpReview.user_id",
"review_length": {$strLenCP : "$yelpReview.text" }}},
{$sort:{"review_length": -1}},
{$limit:10}])

```

```

db.yelpBusiness.aggregate([
{
  $lookup:
  {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  }
},
{$unwind: "$yelpReview" },
{$match: {"city": "Cleveland"}},
{$project : {"business_id":1,"review_id": "$yelpReview.review_id","user_id": "$yelpReview.user_id",
"review_length": {$strLenCP : "$yelpReview.text" }}},
{$sort:{"review_length": -1}},
{$limit:10}])

```

Key	Value	Type
review_id	cueOsgQFEG0eEHx147eDyA	String
user_id	q26iA15EgS7B5jtWFX98nw	String
review_length	5000	Int32
(2) ObjectId("5bf4978e37605d2facc55af4")	{ 5 fields }	Object

Query description: This query is to find the length of the longest review in terms of length for a user and display such top 10 reviews for the city of Belmont.

```

db.yelpBusiness.aggregate([
{
  $lookup:
  {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  }
},
{$unwind: "$yelpReview" },

```

```

{$match: {"city": "Belmont"}},
{$project : {
    "business_id":1,"review_id": "$yelpReview.review_id","user_id": "$yelpReview.user_id",
    "review_length": {$strLenCP : "$yelpReview.text" }}},
{$sort:{"review_length": -1}},
{$limit:10}]

```

Key	Value	Type
business_id	z2Yivr-C0K0H2sLNxvnZyg	String
review_id	lAzxUY89YPduKsQWjo9pfQ	String
user_id	w0B4DF6qQW_9c5LWldC_A	String
review_length	4891	Int32

Summary Results

This query gives the top 10 reviewers from highest to lowest in the order of the count of their total number of reviews of Belmont .

11. This query lists out the top 10 categories with the highest number of reviews in the given city.

Query for Cleveland:

Query Description:

The query is to get the highest number of reviews per category sorted in descending order . We used to lookup to get the city name from the business collection.

We get the overall number

```

db.yelpBusiness.aggregate( [
    {$match:{ "city":{$eq:"Cleveland"} } },
    { $group: { _id:{ "categories": "$categories", "city": "$city" },}
]

```

```

        count: { $sum: "$review_count" }}}},
        { $sort : {count:-1} },
        { $project: { "categories": "$_id.categories", "city": "$_id.city",
            "count": "$count" }},
        { $limit : 10 }
    ])
)

```

Key	Value	Type
\$_id	{4 fields}	Object
categories	{2 fields}	Object
city	Cleveland	String
count	1264	Int32
\$_id	{4 fields}	Object
categories	{2 fields}	Object
city	Cleveland	String
count	966	Int32

Summary Results :

The query gives the overall number of reviews per category arranged from highest to lowest. This can indicate the popularity of a category based on the review count.

Query for Belmont :

Query Description:

The query is to get the highest number of reviews per category sorted in descending order . We used to lookup to get the city name from the business collection.

```

db.yelpBusiness.aggregate([
    {$match:{'city':{$eq:'Belmont'}}},
    {$group: { '_id':{ 'categories': '$categories', 'city': '$city' },
        'count': { '$sum: '$review_count' } } },
    {$sort : {count:-1} },
    {$project: { 'categories': '$_id.categories', 'city': '$_id.city',
        'count': '$count' } },
])
)

```

```
{ $limit : 10 }
])
```

The screenshot shows the Robo 3T interface with the following details:

- File Menu:** File, View, Options, Window, Help.
- LocalDB (5) Tree:**
 - System
 - config
 - handsom
 - termProject** (selected)
 - yelpBusiness** (selected)
 - yelpCheckin
 - yelpPhoto
 - yelpReview
 - yelpTip
 - yelpUser
 - Functions
 - Users (0)
- Query Bar:** db.yelpBusiness.aggregate([{ \$lookup: ... }, { \$match: ... }])
- Result Preview:** Shows the output of the aggregate query. It includes two documents:

Key	Value	Type
1 { 2 fields }	<ul style="list-style-type: none"> _id categories city count 	Object
2 { 2 fields }	<ul style="list-style-type: none"> _id categories city count 	Object

 The first document corresponds to Belmont with a count of 323. The second document corresponds to Soul Food, Arts & Entertainment, Restaurants, Souther... with a count of 289.
- Logs:** Logs tab is visible at the bottom left.
- Taskbar:** Shows the Windows taskbar with various pinned icons and the system tray indicating the date and time (7:32 AM, 12/5/2018).

Summary Results :

The query gives the overall number of reviews per category arranged from highest to lowest. This can indicate the popularity of a category based on the review count.

In the above section of Data Evaluation of the project, we were able to develop a holistic understanding of the data from the queries that we ran. We could develop an understanding of the overall number of businesses in each of the cities Cleveland and Belmont. The total number of reviewers and reviewers pertaining to a business and the whole city has given insights on the data and explore data based on categories. After the initial understanding of the queries ran on various aspects as included in section B we have been able to get a deeper understanding of the data specifically the yelpReview and yelpBusiness datasets. This developed the base for the queries on section C which are the metrics to analyze the data and make inferences on various aspects as described in the section C Queries.

Data Analytics

Based on the results and inference obtained in
Some of the metrics that have considered based on the

1. The top 10 most prolific reviewers

Query for Cleveland:

Query Description :

The query is to give the 10 top users that have the highest count of reviews .
We have used lookup to get the city from the business collection.

```
db.yelpBusiness.aggregate([
{
  $lookup:
  {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  }
},
{$unwind:"$yelpReview" },
{$match: {"city": "Cleveland"}},
{ $group:{"_id":"$yelpReview.user_id","count":{"$sum":1}} },
{$sort:{"count":-1}},
{$project : {"user_id":"$_id","count":"$count"}
},
{$limit:10}],
{"allowDiskUse":true})
```

The screenshot shows the Robo 3T interface. On the left, the sidebar displays the LocalDB (5) database with collections: System, config, handson, termProject, and Collections (6) which include yelpBusiness, yelpCheckin, yelpPhoto, yelpReview, yelpTip, and yelpUser. The termProject collection is selected. The main area shows an aggregation pipeline:

```
{
  $lookup: {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  },
  {$unwind: "$yelpReview" },
  {$match: {"city": "Cleveland"}},
  {$group: {"_id": "$yelpReview.user_id", "count": {"$sum": 1}}},
  {$sort: {"count": -1}},
  {$project: {"user_id": "$_id", "count": "$count"}}
},
{$limit: 10},
{"allowDiskUse": true}
}
```

Below the pipeline, the results are displayed in a table:

Key	Value	Type
_id	dt9IHwfuz9D9LOH7gjNew	String
user_id	dt9IHwfuz9D9LOH7gjNew	String
count	366.0	Double

Summary Results :

We get the top 10 reviewers in terms of the review count which could be considered to rank reviewers based on the count of the reviews for the city of Cleveland.

2. Retrieving the reviewers who have the most number of reviews

Query Description :

The query is to give the 10 top users that have the highest count of reviews. We have used lookup to get the city from the business collection.

```
db.yelpBusiness.aggregate([
{
  $lookup: {
    from: "yelpReview",
    localField: "business_id",
    foreignField: "business_id",
    as: "yelpReview"
  }
},
{$unwind: "$yelpReview" },
{$match: {"city": "Belmont"}},
{$group: {"_id": "$yelpReview.user_id", "count": {"$sum": 1}}},
{$sort: {"count": -1}},
{$project: {"user_id": "$_id", "count": "$count"}}
],
```

```

{$limit:10}],
{"allowDiskUse":true}

```

The screenshot shows the Robo 3T interface. On the left, the database structure for 'LocalDB' is displayed, including collections like 'System', 'config', 'handsom', 'termProject', and 'yelpBusiness'. The 'yelpBusiness' collection is selected. In the main pane, a query is being run against the 'yelpBusiness' collection:

```

db.yelpBusiness.aggregate([
  {
    $lookup: {
      from: "yelpReview",
      localField: "business_id",
      foreignField: "business_id",
      as: "yelpReview"
    }
  },
  {$unwind: "$yelpReview" },
  {$match: { "city": "Belmont"}},
  { $group: { "_id": "$yelpReview.user_id", "count": {"$sum":1} } },
  { $sort: { "count": -1 } },
  { $project : { "user_id": "$_id", "count": "$count" } },
  { $limit:10 },
  {"allowDiskUse":true}
])

```

The results table below the query shows the top 10 reviewers with their review counts:

Key	Value	Type
_id	Jm67TDhh8t1VHPACeORuOg	String
user_id	Jm67TDhh8t1VHPACeORuOg	String
count	26.0	Double

Summary Results

We get the top 10 reviewers in terms of the review count which could be considered to rank reviewers based on the count of the reviews for the city of Belmont.

3. Finding the top rated restraint in a given location based on Latitude n longitude-based:

```

db.YelpBusiness.aggregate([
{$geoNear: {near: {type: "Point", coordinates: [-81.68, 41.50]}},
distanceField: "dist.calculated",
minDistance: 0,}},
{$$match: {$and : [{"review_count":{$gte:1}}, {"stars": {$gt:4}}]}},
{$$sort: {"review_count": -1}},
{$$project: {business_id:1, stars:1, review_count:1, loc:1, city:1}},
{$$limit: 5}
]).pretty()

```

Query Explanation: This query finds the location using the Latitude and longitude by using \$geoNear, and takes the minimum distance as 0 to take the exact location. Then matches the review count greater than or equal to 1 and stars greater than 4, sorts the review count in Descending order and projects the required fields such as business id, city, stars etc. Limiting the records to 5

```

Administrator: Command Prompt - mongo
MongoDB Enterprise > db.YelpBusiness.aggregate([
... {$geoNear: {near: {type: "Point", coordinates: [-81.68, 41.50]}, 
... distanceField: "dist.calculated",
... minDistance: 0,}},
... {$match: {$and : [{"review_count":{$gte:1}}, {"stars": {$gt:4}}]}},
... {$sort: {"review_count": -1}},
... {$project: {business_id:1, stars:1, review_count:1, loc:1, city:1}},
... {$limit: 5}
... ]).pretty()
{
    "_id" : ObjectId("5bf6e74d4dc6a713b4909c34"),
    "business_id" : "n5yDcNR0OygR1wkrvsUhA",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 75,
    "loc" : {
        "longitude" : -81.6808043,
        "latitude" : 41.5014035
    }
}
{
    "_id" : ObjectId("5bf6e76f4dc6a713b4937cea"),
    "business_id" : "n5yDcNR0OygR1wkrvsUhA",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 75,
    "loc" : {
        "longitude" : -81.6808043,
        "latitude" : 41.5014035
    }
}
{
    "_id" : ObjectId("5bf6e74c4dc6a713b4905f17"),
    "business_id" : "Rt7gjrTce086sCtnS8Td0g",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 49,
    "loc" : {
        "longitude" : -81.6812,
        "latitude" : 41.500613
    }
}
{
    "longitude" : -81.6808043,
    "latitude" : 41.5014035
}
{
    "_id" : ObjectId("5bf6e74c4dc6a713b4905f17"),
    "business_id" : "Rt7gjrTce086sCtnS8Td0g",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 49,
    "loc" : {
        "longitude" : -81.6812,
        "latitude" : 41.500613
    }
}
{
    "_id" : ObjectId("5bf6e76f4dc6a713b4933fcf"),
    "business_id" : "Rt7gjrTce086sCtnS8Td0g",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 49,
    "loc" : {
        "longitude" : -81.6812,
        "latitude" : 41.500613
    }
}
{
    "_id" : ObjectId("5bf6e7534dc6a713b4929022"),
    "business_id" : "q5WnhptrzfpHPYt1RVxHQ",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 48,
    "loc" : {
        "longitude" : -81.6807262,
        "latitude" : 41.5012767
    }
}
MongoDB Enterprise >

```

```

Administrator: Command Prompt - mongo
{
    "longitude" : -81.6808043,
    "latitude" : 41.5014035
}
{
    "_id" : ObjectId("5bf6e74c4dc6a713b4905f17"),
    "business_id" : "Rt7gjrTce086sCtnS8Td0g",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 49,
    "loc" : {
        "longitude" : -81.6812,
        "latitude" : 41.500613
    }
}
{
    "_id" : ObjectId("5bf6e76f4dc6a713b4933fcf"),
    "business_id" : "Rt7gjrTce086sCtnS8Td0g",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 49,
    "loc" : {
        "longitude" : -81.6812,
        "latitude" : 41.500613
    }
}
{
    "_id" : ObjectId("5bf6e7534dc6a713b4929022"),
    "business_id" : "q5WnhptrzfpHPYt1RVxHQ",
    "city" : "Cleveland",
    "stars" : 4.5,
    "review_count" : 48,
    "loc" : {
        "longitude" : -81.6807262,
        "latitude" : 41.5012767
    }
}
MongoDB Enterprise >

```

Summary: User can find the top rated restaurant in his location. The above query is giving the top restaurants with user location as Lat: 41.50 and Long: -81.68 (Cleveland city)

4. Review Histogram of overall ratings per reviewer:

```

db.Yelpreview.aggregate([
{$group: {_id: '$review_id', 'OverAll': {$push: '$stars'}, count: {$sum:1}}},
{$sort: {_id:1}}
]

```

```
50}], {"allowDiskUse": true})
```

Query Description: This query will group by ReviewID and by Overall having stars column. Then need to count it using \$sum, sort the results with review id by ascending. Limiting the number of records to 50.

```

MongoDB Enterprise > db.Yelpreview.aggregate([
  { $group: { _id: '$review_id', 'OverAll': { $push: '$stars', count:{$sum:1}} },
  { $sort: { _id: 1} },
  { $limit: 50 }
], {"allowDiskUse":true})
{
  "_id": "--1U_KyCfPuYgza1q09PQ",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--2ZCerInkOBYS21laIg",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--2KPEupsYzgbCM4IwtG",
  "OverAll": [ { 3, 1 }, { "count": 2 } ]
},
{
  "_id": "--54U1LrC43ketkO3hB5wg",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--63b6e2zvPvPS1T1-0Hc",
  "OverAll": [ { 2, 2 }, { "count": 2 } ]
},
{
  "_id": "--9Qzx_UgvjwvJ-10p",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--aR7gwpwAVLsVT1238w",
  "OverAll": [ { 5, 2 }, { "count": 2 } ]
},
{
  "_id": "--BnqoDg2d621HnH4hM",
  "OverAll": [ { 4 }, { "count": 1 } ]
},
{
  "_id": "--o0JLxpzt20hL7RB-Chp",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--Prw4tLx-TFneqEDj3Ew",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--QWde22JneqEDj3Ew",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--QQ84kkrsgslns550dpSQ",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--R_3kjoxaypzr2rbfQG",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--Tew3x8ldcc_3wBaQag",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--VxVg-13whfPwv008g8mUA",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--VmD1voByocca1PhgHy",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--wbMfvnsa4n9uZDmZ1w",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--X6hr50THCw8v1uAme6Q",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--XkgF592Fwini5-xFRy1g",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--XsqvLXhtt20d5zSLR74g",
  "OverAll": [ { 1, 1 }, { "count": 2 } ]
}
]
Type "it" for more
MongoDB Enterprise > it
{
  "_id": "--Y361n-X-U1Ej2zdPrh_A",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--Yp2tFkoFDn1igu6ASFog",
  "OverAll": [ { 1, 1 }, { "count": 2 } ]
},
{
  "_id": "--YjCE383csOSHyFrncnsdA",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--YseMrdpEBfcPEGMwXlQA",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--_JFK-w4RELrKowIAEBZA",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--adRHCMBLfGeL-odhgw",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--bdwTmtB18D01SwmOTRA",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--f37P1rj7q72mQ1wrfIda",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--FDx_fj5CmxHyKAfbeg",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--KRTz-D8RRSoykOA2FO4w",
  "OverAll": [ { 4, 4 }, { "count": 2 } ]
},
{
  "_id": "--1-2-ZUT0m0jLAaYhvotg",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--opRwsqwdGyF6UXjyfMq",
  "OverAll": [ { 3, 3 }, { "count": 2 } ]
},
{
  "_id": "--pxCNwqA1MS5VjYtrVOOA",
  "OverAll": [ { 3, 3 }, { "count": 2 } ]
},
{
  "_id": "--rphq88LTbJAnJ88go1Qw",
  "OverAll": [ { 5 }, { "count": 1 } ]
}
]
Type "it" for more
MongoDB Enterprise > it
{
  "_id": "--Y361n-X-U1Ej2zdPrh_A",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--Yp2tFkoFDn1igu6ASFog",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--YjCE383csOSHyFrncnsdA",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--YseMrdpEBfcPEGMwXlQA",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--_JFK-w4RELrKowIAEBZA",
  "OverAll": [ { 1, 1 }, { "count": 1 } ]
},
{
  "_id": "--adRHCMBLfGeL-odhgw",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--bdwTmtB18D01SwmOTRA",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--f37P1rj7q72mQ1wrfIda",
  "OverAll": [ { 3 }, { "count": 1 } ]
},
{
  "_id": "--FDx_fj5CmxHyKAfbeg",
  "OverAll": [ { 4, 4 }, { "count": 2 } ]
},
{
  "_id": "--KRTz-D8RRSoykOA2FO4w",
  "OverAll": [ { 5, 1 }, { "count": 1 } ]
},
{
  "_id": "--1-2-ZUT0m0jLAaYhvotg",
  "OverAll": [ { 3, 3 }, { "count": 2 } ]
},
{
  "_id": "--opRwsqwdGyF6UXjyfMq",
  "OverAll": [ { 3, 3 }, { "count": 2 } ]
},
{
  "_id": "--pxCNwqA1MS5VjYtrVOOA",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--rphq88LTbJAnJ88go1Qw",
  "OverAll": [ { 5, 5 }, { "count": 1 } ]
},
{
  "_id": "--ss64nvMhOnj0Qoggdw",
  "OverAll": [ { 4, 4 }, { "count": 2 } ]
},
{
  "_id": "--ua9CEgruwysvekRBVw",
  "OverAll": [ { 1, 1 }, { "count": 2 } ]
},
{
  "_id": "--uzhZBMLHsBV7wzq5Rvw",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--w8F1whtj-7Q-ZoaZQ",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--xzYHfdteuc2_k_UwA",
  "OverAll": [ { 1 }, { "count": 1 } ]
}
]
Type "it" for more
MongoDB Enterprise > it
{
  "_id": "--0mJiRV761TmgMs7chQ",
  "OverAll": [ { 5 }, { "count": 1 } ]
},
{
  "_id": "--01Pak821wHtdam-67SkaQ",
  "OverAll": [ { 4, 4 }, { "count": 2 } ]
},
{
  "_id": "--07oFEMlfrt1k1dkbQ6qmIA",
  "OverAll": [ { 1 }, { "count": 1 } ]
},
{
  "_id": "--0AHAv168Gfhzfs5vrgA",
  "OverAll": [ { 5, 5 }, { "count": 1 } ]
},
{
  "_id": "--0E61MfAgelFbU9CEDBA",
  "OverAll": [ { 5, 5 }, { "count": 2 } ]
},
{
  "_id": "--0GQf7q4x5-zTrPQ",
  "OverAll": [ { 5, 5 }, { "count": 1 } ]
},
{
  "_id": "--0WV11gawKCKCkQ",
  "OverAll": [ { 4, 4 }, { "count": 1 } ]
},
{
  "_id": "--013gdnjLkUsh-g3R",
  "OverAll": [ { 5, 5 }, { "count": 1 } ]
},
{
  "_id": "--193EVjAE7-dAKrOrDW",
  "OverAll": [ { 9, 9 }, { "count": 1 } ]
},
{
  "_id": "--0L0ln-1q4niv1CLV2Q",
  "OverAll": [ { 1, 1 }, { "count": 1 } ]
}
MongoDB Enterprise >
```

Summary of Results: This query provides an output for the count for each star reviewer has given. We can infer the distribution of star ratings of a user.

5. Latest Review of a particular reviewer:

```

db.YelpBusiness.aggregate([
  { $lookup: {
    from: "YelpReview",
    localField: "business_id",

```

```

        foreignField: "business_id",
        as: "yelpReview"
    },
    {$unwind:"$yelpReview" },
    { $match: { "city": "Cleveland"} },
    { $sort: {"yelpReview.date":1}},
    {$group:{"_id":
        "$yelpReview.review_id","LatestReviewDate":{$last:"$yelpReview.date"}},],
    {"allowDiskUse":true})

```

Query Explanation: This query is doing an outer join between YelpBusiness and yelpreview collections and then unwinds the yelp review columns. Matches to the city named Cleveland sort the review date. Then group the review id from yelp review table and find the last review date using \$last function.

Key	Type
(1) OZKh3e1NSPYwI9-LlOoYjg	Object
_id	String
latestDate	String
(2) yN7DaDixPbxwlaStJLcK6A	Object
(3) TZBZq36Z06i_6Y-qB9muqq	Object
(4) D4N-gAMppQdJrrk68RkezQ	Object
(5) OJSNhYN0DYLa_hd0z52Wyw	Object
(6) dtCIPHO_SKKcMrFF9LxKuQ	Object

Summary: This will provide the latest review done by the reviewer. This can be used to display on the profile of the reviewer.

6. Review Histogram per business_id:

```

db.YelpBusiness.aggregate([
{$group: {_id: {Business: "$business_id", OverAll: "$stars"}, "count":{$sum: 1}}},
{$sort: { "_id.OverAll": -1,"_id.Business": 1 } },
{$limit: 50}

```

])

Query Explanation: this query uses \$group to group by the business id and stars. Then count the number using \$sum. Sort the records using overall star ratings in DESC to get the top ratings and then by business id. Limiting the number of records to 50.

```
Administrator: Command Prompt - mongo
MongoDB Enterprise > db.YelpBusiness.aggregate([
... {$group: { _id: {Business: "$business_id", OverAll: "$stars"}, "count":{$sum: 1}}},
... {$sort: { "_id.OverAll": -1, "_id.Business": 1 } },
... {$limit: 50}
... ])
{
  "_id" : { "Business" : "-Y7NhBkZLtb1l3MjUX_wfg", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-cgVkbWtiga30YTkmQ4", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-irnpXK9y_xl1XPvPAQovw", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-1pMVmkCuji0ZrphHXEA", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-27PM8AGajP0aBmGM7RA", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-00ea0fek6tqfLnrf7x0kg", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-0gUghA7z4-xotUfx1lw", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-0ImWDeqewaA3th8Fz1g", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-0Y5yQz0ufL8EcP8shG1g", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-0z0e0m2ADchyt1ekoxmg", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-0d414u3vKnTT0MEgyX_NA", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-0gEYajhAshchmvJ0vuag", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-0nEqS3_A_mnlut4DrPEEG", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-14Heo2ScisBjI9yjzle9A", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-1BPe8ujf2_1shvK-DFUjA", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-1StD61bfyHQJnjxqsuMA", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-1NptXJadSoqqxQPx3rg3Q", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-1W8v_OLVdvXHtyvnCn44A", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-1hxR87Rw9OYCljdju817dg", "OverAll" : 5 }, "count" : 2 },
  {"_id" : { "Business" : "-1cGM0x0i7T88C9g9w8LxJw", "OverAll" : 5 }, "count" : 2 }
}
Type "it" for more
MongoDB Enterprise > it
{
  "_id" : { "Business" : "1uVyyy3UXNa9WoPucNria", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-1wkz431Z509ysu6kzb5xA", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-28hE6AcAubictGivvC9kw", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-2ATYzeA-9_qC-Whi0-2w", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-2Cx1MjXkyxctJ70ah380", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-2MfmbTycnJ4up77jkfCA", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-2bLuJsmZ20lh19daurVNO", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-35MwouI07lbbstduZfcig", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-38Kck4m1k8wd60xyZRg", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-3K14kIKBhgB0Lff-XFsg", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-3QHAYlnB-VmMcg2Rf5aw", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-3VZCSFu1lispeJ3iit-RQ", "OverAll" : 5 }, "count" : 2 },
  "_id" : { "Business" : "-3ZKe67EhB3m0Mak9YouQ", "OverAll" : 5 }, "count" : 2 },
  {"_id" : { "Business" : "-3pXlL_3nLFUelsSLX1Xug", "OverAll" : 5 }, "count" : 2 }
}
```

```
Administrator: Command Prompt - mongo
MongoDB Enterprise > db.YelpBusiness.aggregate([
... {$group: { _id: {Business: "$business_id", OverAll: "$stars"}, "count":{$sum: 1}}},
... {$sort: { "_id.OverAll": 1, "_id.Business": 1 } },
... {$limit: 50}
... ])
{
  "_id" : { "Business" : "-17Uz92KtnwOpeGVvg17g", "OverAll" : 1 }, "count" : 2 },
  "_id" : { "Business" : "-2dIgo1ass-zmrESrghQw", "OverAll" : 1 }, "count" : 2 },
  "_id" : { "Business" : "-37chdR8740qrle6b6vOB9w", "OverAll" : 1 }, "count" : 2 },
  "_id" : { "Business" : "-4h-Q_DSBqScC1f6j35uPg", "OverAll" : 1 }, "count" : 2 },
  "_id" : { "Business" : "-6Xlt_F22x-pgM3IUJBQmA", "OverAll" : 1 }, "count" : 2 },
  "_id" : { "Business" : "-71WvdGETHYCNRB_V2g", "OverAll" : 1 }, "count" : 2 },
  "_id" : { "Business" : "-8PU0z4Q2Nc-sYebejj4D1Q", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-8aBkrRmrbxNwXELYxvzQO", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-CAH9wl1fu1lgPsERXKLSA", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-DPq2CaW-mBZY3hfed4pxw", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-GS7mhPszZSJTfdwSdp6QQ", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-H6_7u1-hr6wpPP-CZ7iVg", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-HfxGN4083xBP4rZc_-1Q", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-KSQaCB2D4mLiugRdUk1kw", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-Kj1hySwNCred001z-inA", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-L15vDE0acSmDybA_k6zg", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-LG_SXzrWLTrfSXKRPZQvZA", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-Lw8Ve6NLbr0djhGu2fMOA", "OverAll" : 1 }, "count" : 2 },
  {"_id" : { "Business" : "-M8Yt36x3RcCYUD3NmPQ", "OverAll" : 1 }, "count" : 2 }
}
{"_id" : { "Business" : "-MOLcYIhZAgkKJTpjtYfgQ", "OverAll" : 1 }, "count" : 2 }
Type "it" for more
MongoDB Enterprise >
```

Summary: This query will give an idea on the list of individual ratings of a particular business got.

7. List of top restaurants under selected cuisine type for each city.

Steak House

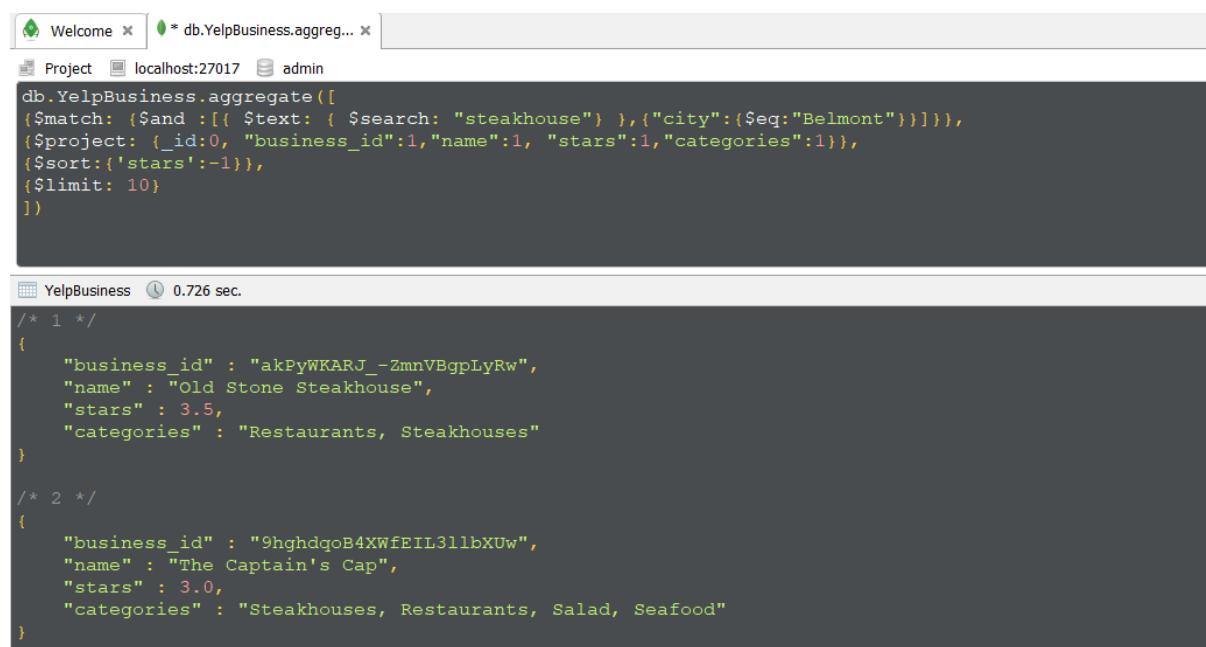
```
db.YelpBusiness.ensureIndex({categories:"text"})
```

Belmont

```
db.YelpBusiness.aggregate([
{$match: {$and :[ { $text: { $search: "steakhouse"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}},
{$limit: 10}
])
```

Explanation: The query tries to group the results by the category Steakhouse for the city Belmont and projects business id, name, stars and category sorted in descending order

Analysis: Helps the user to find the steakhouses in a given city with the highest reviews



The screenshot shows a MongoDB shell interface with the following details:

- Project: localhost:27017
- admin database
- Query: db.YelpBusiness.aggregate([{\$match: {\$and :[{ \$text: { \$search: "steakhouse"} }, {"city":{\$eq:"Belmont"} }]}}, {\$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}}, {\$sort:{'stars':-1}}, {\$limit: 10}])
- Execution time: 0.726 sec.
- Results:
 - /* 1 */

```
{
  "business_id" : "akPyWKARJ_-ZmnVBgpLyRw",
  "name" : "Old Stone Steakhouse",
  "stars" : 3.5,
  "categories" : "Restaurants, Steakhouses"
}
```
 - /* 2 */

```
{
  "business_id" : "9hghdgoB4XWfEIL3llbXUw",
  "name" : "The Captain's Cap",
  "stars" : 3.0,
  "categories" : "Steakhouses, Restaurants, Salad, Seafood"
}
```

Result: Displays the steakhouses in Belmont and also displays the business id, name, stars and category sorted in descending order

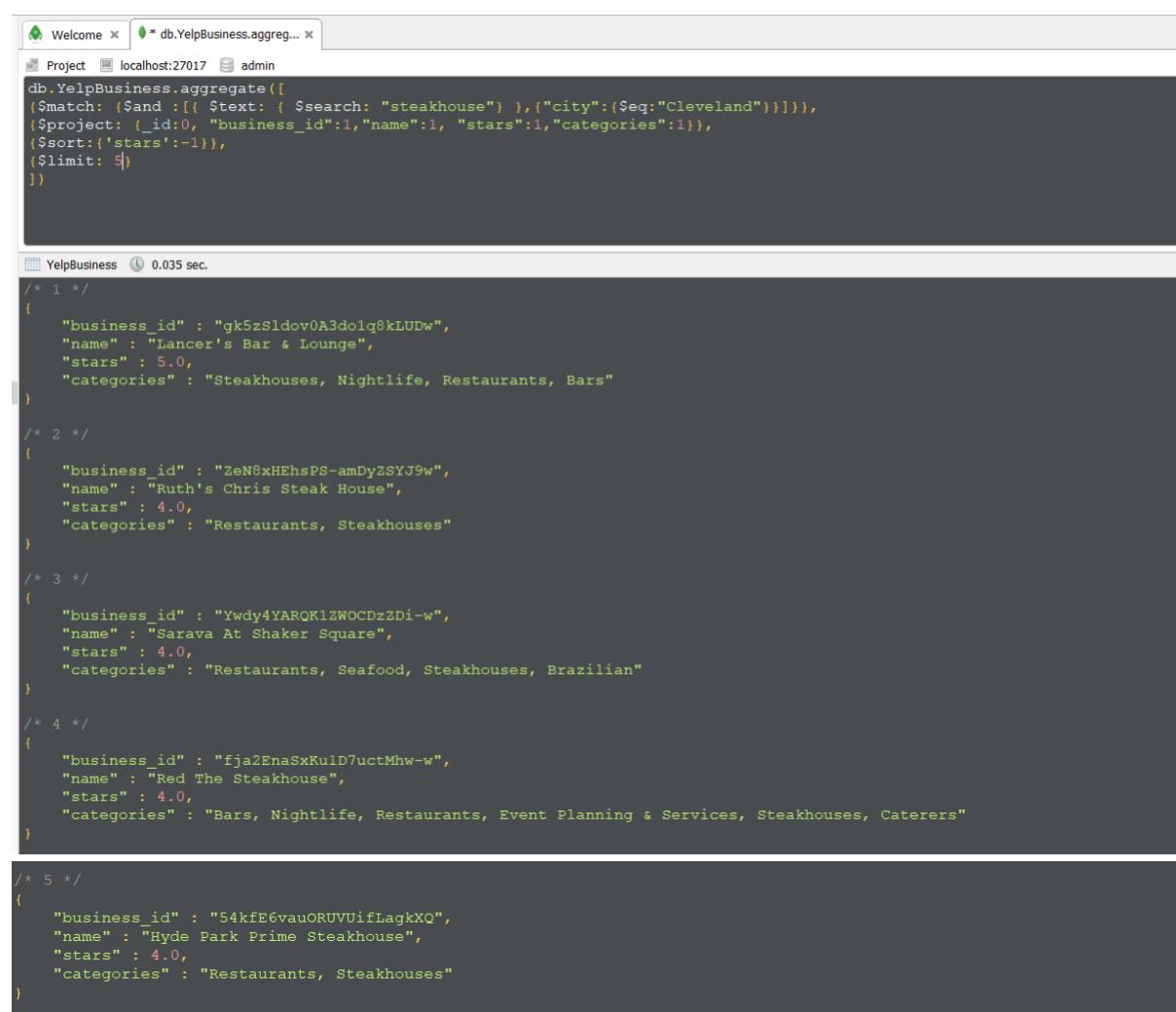
Cleveland

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "steakhouse"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 10}
])

```

Explanation: The query tries to group the results by the category Steakhouse for the city Cleveland and projects business id, name, stars and category sorted in descending order
 Analysis: Helps the user to find the steakhouses in a given city with the highest reviews



The screenshot shows a MongoDB shell interface with the following details:

- Project:** localhost:27017
- Database:** admin
- Query:** db.YelpBusiness.aggregate([{\$match: {\$and :[{\$text: { \$search: "steakhouse"} }, {"city":{\$eq:"Cleveland"} }]}}, {\$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}}, {\$sort:{'stars':-1}}, {\$limit: 10}])
- Time:** 0.035 sec.
- Results:** Displays five documents (labeled 1-5) representing steakhouses in Cleveland, ordered by stars in descending order. Each document includes business_id, name, stars, and categories.

```

/* 1 */
{
  "business_id" : "gk5zSldov0A3dolq8kLUDw",
  "name" : "Lancer's Bar & Lounge",
  "stars" : 5.0,
  "categories" : "Steakhouses, Nightlife, Restaurants, Bars"
}

/* 2 */
{
  "business_id" : "ZeN8xHEhsPS-amDyZSYJ9w",
  "name" : "Ruth's Chris Steak House",
  "stars" : 4.0,
  "categories" : "Restaurants, Steakhouses"
}

/* 3 */
{
  "business_id" : "Ywdy4YARQK1ZWOCdzZDi-w",
  "name" : "Sarava At Shaker Square",
  "stars" : 4.0,
  "categories" : "Restaurants, Seafood, Steakhouses, Brazilian"
}

/* 4 */
{
  "business_id" : "fja2EnaSxKu1D7uctMhw-w",
  "name" : "Red The Steakhouse",
  "stars" : 4.0,
  "categories" : "Bars, Nightlife, Restaurants, Event Planning & Services, Steakhouses, Caterers"
}

/* 5 */
{
  "business_id" : "54kfE6vauORUVUiLlagkQ",
  "name" : "Hyde Park Prime Steakhouse",
  "stars" : 4.0,
  "categories" : "Restaurants, Steakhouses"
}

```

Result: Displays the steakhouses in Belmont and also displays the business id, name, stars and category sorted in descending order

5.list of top bars and pubs in the cities with highest review count.

```

db.YelpBusiness.aggregate([

```

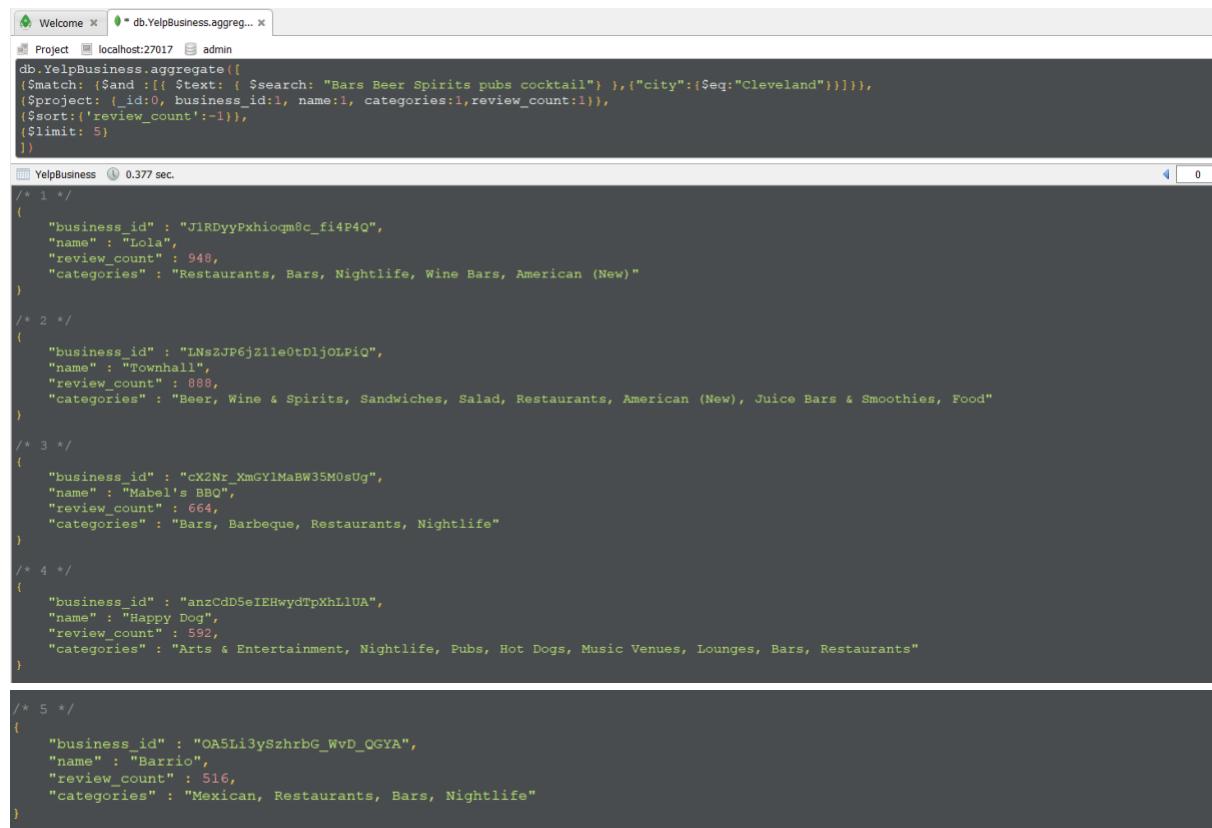
```

{$match: { $and :[{ $text: { $search: "Bars Beer Spirits pubs cocktail"}},
 {"city":{$eq:"Cleveland"}]}]},
 {$project: {_id:0, business_id:1, name:1, categories:1,review_count:1}},
 {$sort:{'review_count':-1}},
 {$limit: 5}
])

```

Explanation: The query tries to group the results by the category of Bars and Pubs using keywords Bars, Beer, Spirits and Cocktail for the city Cleveland and projects business id, name, stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants with Bars and Pubs in a given city with the highest reviews



The screenshot shows the MongoDB Compass interface with the aggregation pipeline and the resulting documents. The pipeline is:

```

db.YelpBusiness.aggregate([
  {$match: { $and :[{ $text: { $search: "Bars Beer Spirits pubs cocktail"}},
    {"city":{$eq:"Cleveland"}]}]},
  {$project: {_id:0, business_id:1, name:1, categories:1,review_count:1}},
  {$sort:{'review_count':-1}},
  {$limit: 5}
])

```

The resulting documents are:

```

/* 1 */
{
  "business_id" : "J1RDyyPxhiogm8c_fi4P4Q",
  "name" : "Lola",
  "review_count" : 948,
  "categories" : "Restaurants, Bars, Nightlife, Wine Bars, American (New)"
}

/* 2 */
{
  "business_id" : "LNszJP6jZ1le0tDljoLPiQ",
  "name" : "Townhall",
  "review_count" : 888,
  "categories" : "Beer, Wine & Spirits, Sandwiches, Salad, Restaurants, American (New), Juice Bars & Smoothies, Food"
}

/* 3 */
{
  "business_id" : "cX2Nr_XmGY1MaBW3SM0sUg",
  "name" : "Mabel's BBQ",
  "review_count" : 664,
  "categories" : "Bars, Barbecue, Restaurants, Nightlife"
}

/* 4 */
{
  "business_id" : "anzCdd5eIEHwydTpXhLlUA",
  "name" : "Happy Dog",
  "review_count" : 592,
  "categories" : "Arts & Entertainment, Nightlife, Pubs, Hot Dogs, Music Venues, Lounges, Bars, Restaurants"
}

/* 5 */
{
  "business_id" : "OA5Li3ySzhrbG_WvD_QGYA",
  "name" : "Barrio",
  "review_count" : 516,
  "categories" : "Mexican, Restaurants, Bars, Nightlife"
}

```

Result: Displays the Restaurants with Bars and Pubs in Cleveland and also displays the business id, name, stars and category sorted in descending order

```

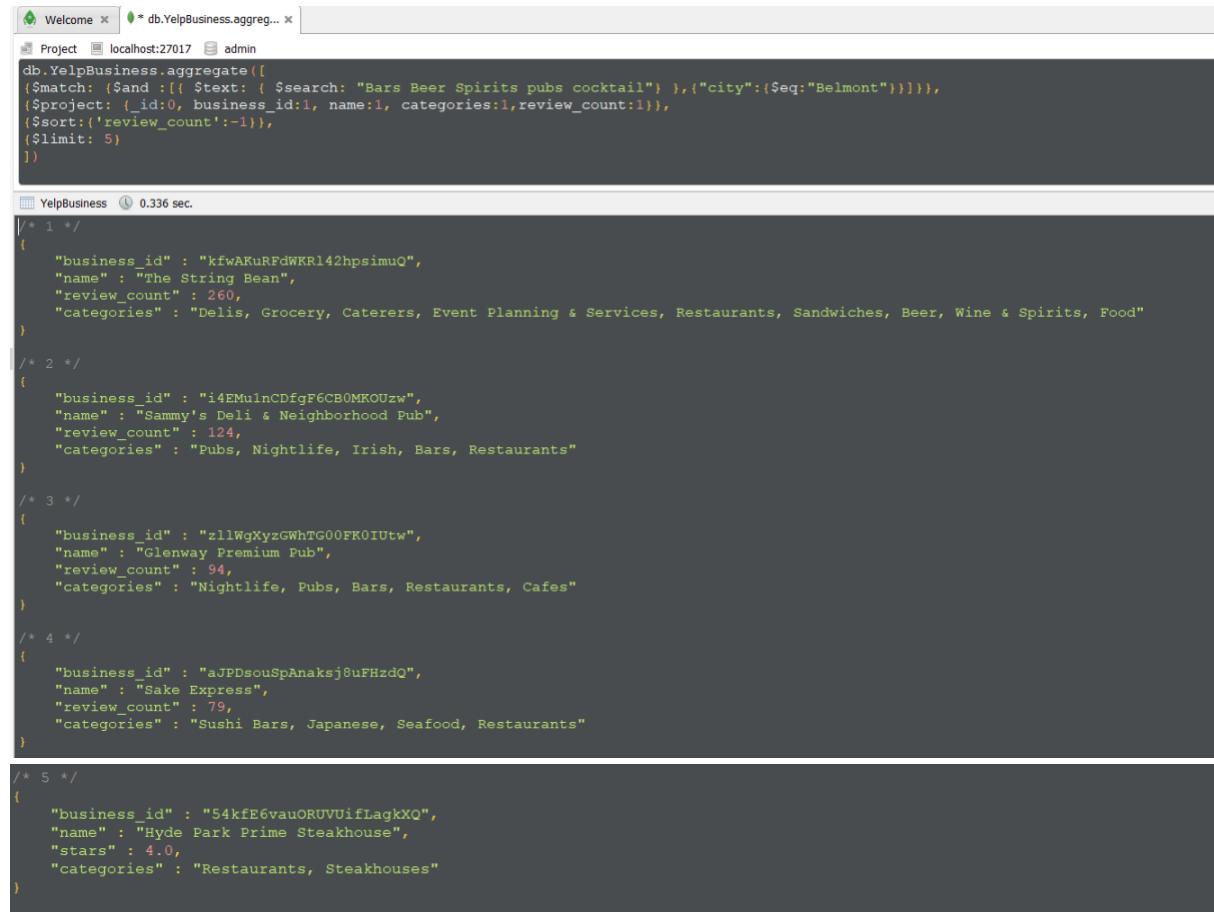
db.YelpBusiness.aggregate([
{$match: { $and :[{ $text: { $search: "Bars Beer Spirits pubs cocktail"}},
 {"city":{$eq:"Belmont"}]}]}},
 {$project: {_id:0, business_id:1, name:1, categories:1,review_count:1}},
 {$sort:{'review_count':-1}},
 {$limit: 5}
]
)

```

])

Explanation: The query tries to group the results by the category of Bars and Pubs using keywords Bars,Beer,Spirits and Cocktail for the city Belmont and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants with Bars and Pubs in a given city with the highest reviews



```
db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "Bars Beer Spirits pubs cocktail"}}, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, business_id:1, name:1, categories:1,review_count:1}},
{$sort:{'review_count':-1}},
{$limit: 5}
])

YelpBusiness 0.336 sec.

/*
 1
{
  "business_id" : "kfwAKuRFdWKRl42hpsimuQ",
  "name" : "The String Bean",
  "review_count" : 260,
  "categories" : "Delis, Grocery, Caterers, Event Planning & Services, Restaurants, Sandwiches, Beer, Wine & Spirits, Food"
}

/*
 2
{
  "business_id" : "i4EMuinCDfgF6CB0MKOUzw",
  "name" : "Sammy's Deli & Neighborhood Pub",
  "review_count" : 124,
  "categories" : "Pubs, Nightlife, Irish, Bars, Restaurants"
}

/*
 3
{
  "business_id" : "zllWgXyzGWhTG00FK0IUtw",
  "name" : "Glenway Premium Pub",
  "review_count" : 94,
  "categories" : "Nightlife, Pubs, Bars, Restaurants, Cafes"
}

/*
 4
{
  "business_id" : "aJPDsouSpAnaksj8uFHzdQ",
  "name" : "Sake Express",
  "review_count" : 79,
  "categories" : "Sushi Bars, Japanese, Seafood, Restaurants"
}

/*
 5
{
  "business_id" : "54kfE6vauORUVUiLagkXQ",
  "name" : "Hyde Park Prime Steakhouse",
  "stars" : 4.0,
  "categories" : "Restaurants, Steakhouses"
}
```

Result: Displays the Restaurants with Bars and Pubs in Belmont and also displays the business id, name, stars and category sorted in descending order

Pizza

```
db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "pizza"} }, {"city":{$eq:"Cleveland"} }]}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}},
{$limit: 5}
])
```

Explanation: The query tries to group the results by the category for the city Cleveland and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants that offers pizza in a given city with the highest reviews

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "pizza"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])

YelpBusiness 0.147 sec.

/* 1 */
{
  "business_id" : "32DUrg5ThezfileGsWK6Gdw",
  "name" : "Moe's Tavern",
  "stars" : 5.0,
  "categories" : "Dive Bars, Nightlife, Bars, Gastropubs, Restaurants, Pubs, Pizza"
}

/* 2 */
{
  "business_id" : "GtxwFRoQTittTa6u3oVgS0g",
  "name" : "Juice Up Cold Lpress",
  "stars" : 5.0,
  "categories" : "Restaurants, Food, Organic Stores, Live/Raw Food, Juice Bars & Smoothies, Pizza"
}

/* 3 */
{
  "business_id" : "Y1Nw27hk9C-xofNxO3TIRw",
  "name" : "Marco's Pizza",
  "stars" : 5.0,
  "categories" : "Restaurants, Pizza"
}

/* 4 */
{
  "business_id" : "dhKYHhp136Fzu_RASIiTeg",
  "name" : "Cha Spirits & Pizza Kitchen",
  "stars" : 5.0,
  "categories" : "Nightlife, Restaurants, Pizza, American (New), Bars"
}

/* 5 */
{
  "business_id" : "q0PH5CjXsdmpL-LLgCNdeA",
  "name" : "Ferrara's Imported Foods",
  "stars" : 4.5,
  "categories" : "Food, Grocery, Pizza, Restaurants, Delis"
}

```

Result: Displays the Restaurants that provides good pizzas in Cleveland and also displays the business id,name,stars and category sorted in descending order

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "pizza"} }, {"city":{$eq:"Belmont"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])

```

Explanation: The query tries to group the results by the category for the city Belmont and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants that offers pizza in a given city with the highest reviews

The screenshot shows the MongoDB Compass interface with two tabs: 'Welcome' and 'YelpBusiness'. The 'YelpBusiness' tab displays the results of an aggregate query. The query is:

```
db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "pizza"} }, {"city":{$eq:"Belmont"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])
```

The results are listed below, showing five documents (labeled 1 through 5) with their business id, name, stars, and categories.

```
/*
 1
{
  "business_id" : "7MwN0zxxJNBWJ6hvsBjGPg",
  "name" : "Johnny B's Pizza Pad & Watering Hole",
  "stars" : 4.0,
  "categories" : "Pizza, Restaurants"
}

/*
 2
{
  "business_id" : "7zbTRmkPaiSch7tTSy7Dw",
  "name" : "Joe's Touch of Italy",
  "stars" : 4.0,
  "categories" : "Restaurants, Italian, Sandwiches, Pizza"
}

/*
 3
{
  "business_id" : "54rjpC8YuM3QN_QbFGmLAQ",
  "name" : "Simonetti's Pizza",
  "stars" : 3.5,
  "categories" : "Sandwiches, Restaurants, Pizza, Italian"
}

/*
 4
{
  "business_id" : "hQMyVVYkPl05i2ItdHmiGw",
  "name" : "Papa John's Pizza",
  "stars" : 3.0,
  "categories" : "Restaurants, Italian, Pizza"
}

/*
 5
{
  "business_id" : "KLtrBp5jFUccZQhcbA46Fg",
  "name" : "Sub Corral Sandwich Shop",
  "stars" : 3.0,
  "categories" : "Sandwiches, Pizza, Restaurants, Burgers"
}
```

Result: Displays the Restaurants that provides good pizzas in Belmont and also displays the business id, name, stars and category sorted in descending order

Mexican

```
db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "mexican"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])
```

Explanation: The query tries to group the results by the category for the city Cleveland and projects business id, name, stars and category sorted in descending order

Analysis: Helps the user to find the Mexican Restaurants in a given city with the highest reviews

The screenshot shows the MongoDB shell interface with a query and its execution results.

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "mexican"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])

```

YelpBusiness 0.147 sec.

```

/* 1 */
{
  "business_id" : "32DUrg5ThezfileGsWK6Gdw",
  "name" : "Moe's Tavern",
  "stars" : 5.0,
  "categories" : "Dive Bars, Nightlife, Bars, Gastropubs, Restaurants, Pubs, Pizza"
}

/* 2 */
{
  "business_id" : "GtxwFRoQTItTa6u3cVgS0g",
  "name" : "Juice Up Cold Lpress",
  "stars" : 5.0,
  "categories" : "Restaurants, Food, Organic Stores, Live/Raw Food, Juice Bars & Smoothies, Pizza"
}

/* 3 */
{
  "business_id" : "Y1Nw27hk9C-xofNxO3TIRw",
  "name" : "Marco's Pizza",
  "stars" : 5.0,
  "categories" : "Restaurants, Pizza"
}

/* 4 */
{
  "business_id" : "dhKYHhp136Fzu_RASIIiTEg",
  "name" : "Cha Spirits & Pizza Kitchen",
  "stars" : 5.0,
  "categories" : "Nightlife, Restaurants, Pizza, American (New), Bars"
}

/* 5 */
{
  "business_id" : "q0PH5CjXSdmpL-LLgCNdeA",
  "name" : "Ferrara's Imported Foods",
  "stars" : 4.5,
  "categories" : "Food, Grocery, Pizza, Restaurants, Delis"
}

```

Result: Displays the Restaurants with Mexican cuisine in Cleveland and also displays the business id,name,stars and category sorted in descending order

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "mexican"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}}
])

```

Explanation: The query tries to group the results by the category for the city Belmont and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Mexican Restaurants in a given city with the highest reviews

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "mexican"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}),
{$sort:{'stars':-1}}
])

YelpBusiness 0.125 sec.

/* 1 */
{
  "business_id" : "qqIGwFEE-2My54wRM1leAw",
  "name" : "Taxco Mexican Grill",
  "stars" : 3.5,
  "categories" : "Restaurants, Mexican"
}

/* 2 */
{
  "business_id" : "bHWUajFY-fHtNXTRnXh1TA",
  "name" : "Taco Bell",
  "stars" : 2.5,
  "categories" : "Tex-Mex, Restaurants, Mexican, Fast Food"
}

```

Result: Displays the Restaurants with Mexican cuisine in Belmont and also displays the business id,name,stars and category sorted in descending order

Chinese

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "chinese"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}),
{$sort:{'stars':-1}}
])

```

Explanation: The query tries to group the results by the category for the city Belmont and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Chinese Restaurants in a given city with the highest reviews

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "chinese"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}),
{$sort:{'stars':-1}}
])

YelpBusiness 0.098 sec.

/* 1 */
{
  "business_id" : "DwauOpTROj2QsR_hz7JVoQ",
  "name" : "New China",
  "stars" : 4.0,
  "categories" : "Chinese, Restaurants"
}

/* 2 */
{
  "business_id" : "kIoKla6pdHMzwhV9YksaQ",
  "name" : "Beijing To Go",
  "stars" : 3.5,
  "categories" : "Cantonese, Szechuan, Chinese, Restaurants"
}

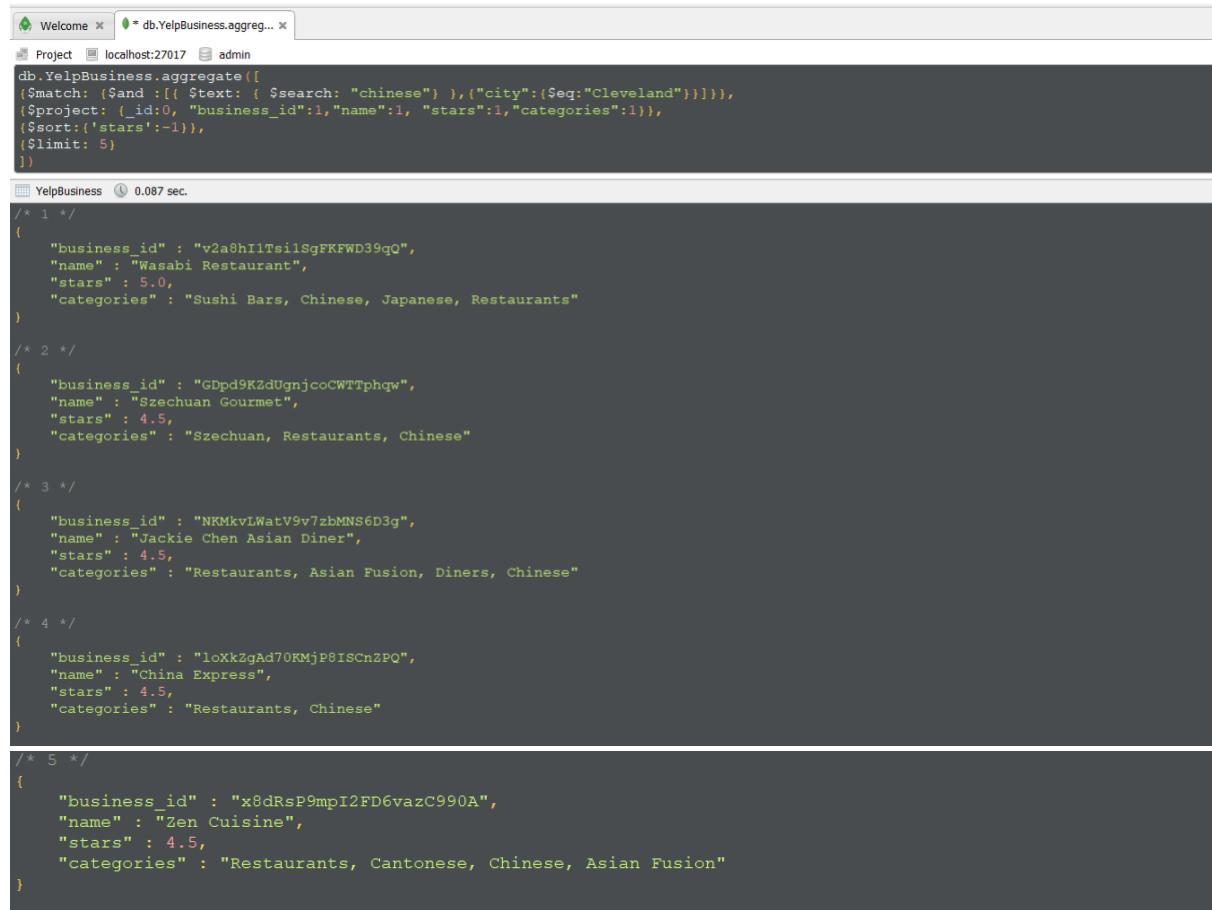
```

Result: Displays the the Restaurants with Chinese cuisine in Belmont and also displays the business id, name, stars and category sorted in descending order

```
db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "chinese"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])
```

Explanation: The query tries to group the results by the category for the city Cleveland and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Chinese Restaurants in a given city with the highest reviews



The screenshot shows the MongoDB shell interface with a single tab titled "db.YelpBusiness.aggregate...". The query is displayed in the top pane:

```
db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "chinese"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])
```

The results of the query are shown in the bottom pane, displaying five documents, each representing a Chinese restaurant in Cleveland with its business ID, name, stars, and categories:

```
/* 1 */
{
  "business_id" : "v2a8hI1Tsi1SgFKFWD39gQ",
  "name" : "Wasabi Restaurant",
  "stars" : 5.0,
  "categories" : "Sushi Bars, Chinese, Japanese, Restaurants"
}

/* 2 */
{
  "business_id" : "GDpd9KZdUgnjcoCWTTphqw",
  "name" : "Szechuan Gourmet",
  "stars" : 4.5,
  "categories" : "Szechuan, Restaurants, Chinese"
}

/* 3 */
{
  "business_id" : "NKMkvLWatV9v7zbMNS6D3g",
  "name" : "Jackie Chen Asian Diner",
  "stars" : 4.5,
  "categories" : "Restaurants, Asian Fusion, Diners, Chinese"
}

/* 4 */
{
  "business_id" : "loXkZgAd70KMjP0ISChzPQ",
  "name" : "China Express",
  "stars" : 4.5,
  "categories" : "Restaurants, Chinese"
}

/* 5 */
{
  "business_id" : "x8dRsP9mpI2FD6vazC990A",
  "name" : "Zen Cuisine",
  "stars" : 4.5,
  "categories" : "Restaurants, Cantonese, Chinese, Asian Fusion"
}
```

Result: Displays the Restaurants with Chinese cuisine in Cleveland and also displays the business id,name,stars and category sorted in descending order

Coffee

```
db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "coffee"} }, {"city":{$eq:"Cleveland"} }]}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}},
{$limit: 5}
])
```

Explanation: The query tries to group the results by the category for the city Cleveland and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants with special coffee corners in a given city with the highest reviews

```
db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "coffee"} }, {"city":{$eq:"Cleveland"} }]}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}},
{$limit: 5}
])

/* 1 */
{
  "business_id" : "7OaW_FuiE6jObsoCBdw7gg",
  "name" : "Coffee Bean Cafe & Bakery",
  "stars" : 5.0,
  "categories" : "Restaurants, Bakeries, Cafes, Coffee & Tea, Food, Sandwiches"
}

/* 2 */
{
  "business_id" : "C2jzQ78vtQyKGTKrIIEcqA",
  "name" : "High Society Boutique",
  "stars" : 5.0,
  "categories" : "Coffee & Tea, Tobacco Shops, Personal Shopping, Shopping, Food"
}

/* 3 */
{
  "business_id" : "oxx6LBBeBMr3xLkj3jq6rFw",
  "name" : "Tabu Coffee Company",
  "stars" : 5.0,
  "categories" : "Food, Cafes, Coffee & Tea, Restaurants"
}

/* 4 */
{
  "business_id" : "OKufZ-0fcNdHmLGgk62ZMQ",
  "name" : "Westown Grind Caffe",
  "stars" : 5.0,
  "categories" : "Restaurants, Sandwiches, Food, Coffee & Tea"
}

/* 5 */
{
  "business_id" : "-H1YgsXYBjH-va7cLIqXxg",
  "name" : "Foyer",
  "stars" : 5.0,
  "categories" : "Restaurants, Food, Cafes, Venues & Event Spaces, Coffee & Tea, Event Planning & Services"
}
```

Result: Displays the Restaurants that is known for coffee in Cleveland and also displays the business id,name,stars and category sorted in descending order

```
db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "coffee"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
```

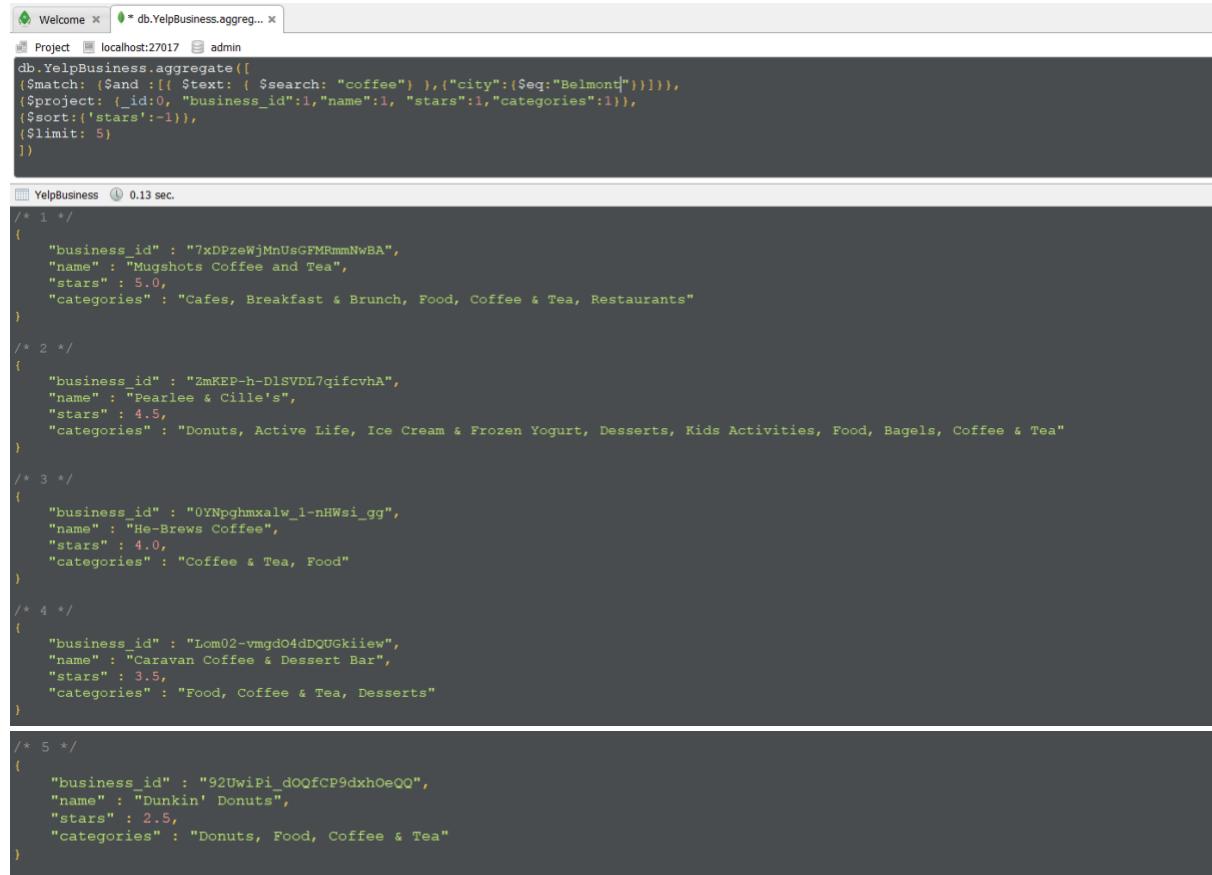
```

{$sort:{'stars':-1}},
{$limit: 5}
])

```

Explanation:The query tries to group the results by the category for the city Belmont and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants with special coffee corners in a given city with the highest reviews



The screenshot shows the MongoDB Compass interface with a query in the top panel and its results in the bottom panel.

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "coffee"} }, {"city":{$eq:"Belmont"} }]}},
  {$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])

```

YelpBusiness 0.13 sec.

```

/* 1 */
{
  "business_id" : "7xDPzeWjMnUsGFMRmmNwBA",
  "name" : "Mugshots Coffee and Tea",
  "stars" : 5.0,
  "categories" : "Cafes, Breakfast & Brunch, Food, Coffee & Tea, Restaurants"
}

/* 2 */
{
  "business_id" : "ZmKEP-h-DlSVDL7qifcvhA",
  "name" : "Pearlee & Cille's",
  "stars" : 4.5,
  "categories" : "Donuts, Active Life, Ice Cream & Frozen Yogurt, Desserts, Kids Activities, Food, Bagels, Coffee & Tea"
}

/* 3 */
{
  "business_id" : "0YNpghmxalw_1-nHwsi_gg",
  "name" : "He-Brews Coffee",
  "stars" : 4.0,
  "categories" : "Coffee & Tea, Food"
}

/* 4 */
{
  "business_id" : "Lom02-vmgdO4dDQUGkiew",
  "name" : "Caravan Coffee & Dessert Bar",
  "stars" : 3.5,
  "categories" : "Food, Coffee & Tea, Desserts"
}

/* 5 */
{
  "business_id" : "92UwiPi_dOQfCP9dxhoeQQ",
  "name" : "Dunkin' Donuts",
  "stars" : 2.5,
  "categories" : "Donuts, Food, Coffee & Tea"
}

```

Result:Displays the Restaurants that is known for coffee in Belmont and also displays the business id,name,stars and category sorted in descending order

American

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "american"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}},
{$limit: 5}
]

```

Explanation:The query tries to group the results by the category for the city Belmont and projects business id,name,stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants with American cuisine in a given city with the highest reviews

```
])
```

```
Project  localhost:27017  admin
db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "american"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}),
{$limit: 5}
])

YelpBusiness  0.245 sec.

/* 1 */
{
  "business_id" : "z2Yivr-COKOH2sLNxvnZyg",
  "name" : "Peace -N- Hominy Q",
  "stars" : 4.5,
  "categories" : "Restaurants, Barbecue, Hot Dogs, Shopping, Breakfast & Brunch, Appliances, Burgers, American (Traditional), Southern, Home & Garden"
}

/* 2 */
{
  "business_id" : "V6VBg1sexfS2cX7yyPZmFQ",
  "name" : "LunaHombre",
  "stars" : 4.5,
  "categories" : "Beer, Wine & Spirits, Tapas/Small Plates, Food, American (New), Nightlife, Wine Bars, Bars, Mediterranean, Restaurants"
}

/* 3 */
{
  "business_id" : "pCJorQE8rdmvvSNR49WS5w",
  "name" : "Ranucci's BBQ & Grill",
  "stars" : 4.0,
  "categories" : "American (Traditional), Restaurants, Barbecue"
}

/* 4 */
{
  "business_id" : "l2kzkT3iw10kIq2niWRbgQ",
  "name" : "Famous Toastery",
  "stars" : 4.0,
  "categories" : "Restaurants, Event Planning & Services, Breakfast & Brunch, American (Traditional), Caterers"
}

/* 5 */
{
  "business_id" : "fWUG5sZRLHleCR2G_dhPRw",
  "name" : "Catawba Fountain & Grill",
  "stars" : 3.5,
  "categories" : "Restaurants, Barbecue, American (Traditional)"
}
```

Result: Displays the Restaurants with American cuisine in Belmont and also displays the business id, name, stars and category sorted in descending order

```
db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "american"} }, {"city":{$eq:"Cleveland"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
{$sort:{'stars':-1}},
{$limit: 5}
])
```

Explanation: The query tries to group the results by the category for the city Cleveland and projects business id, name, stars and category sorted in descending order

Analysis: Helps the user to find the Restaurants with American cuisine in a given city with the highest reviews

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "american"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1}},
  {$sort:{'stars':-1}},
  {$limit: 5}
])

YelpBusiness 0.209 sec.

/* 1 */
{
  "business_id" : "mXaqyzYTqiTc-LrTFAm0Yw",
  "name" : "Sabor Miami Cafe & Gallery",
  "stars" : 5.0,
  "categories" : "Restaurants, Latin American, Cafes"
}

/* 2 */
{
  "business_id" : "dhKYHhp136Fzu_RASLiTEG",
  "name" : "Cha Spirits & Pizza Kitchen",
  "stars" : 5.0,
  "categories" : "Nightlife, Restaurants, Pizza, American (New), Bars"
}

/* 3 */
{
  "business_id" : "AXYaEjWl-xo9oRgEnrPOTQ",
  "name" : "Trio",
  "stars" : 5.0,
  "categories" : "Restaurants, American (New)"
}

/* 4 */
{
  "business_id" : "g8_Tf84s9Elvh5c3BC7Zzg",
  "name" : "Miramar",
  "stars" : 5.0,
  "categories" : "Restaurants, Salvadoran, Latin American"
}

/* 5 */
{
  "business_id" : "Q373Y-siDCSP63rU_F3qMw",
  "name" : "Cleveland Mofongo Latin Grill",
  "stars" : 5.0,
  "categories" : "Latin American, Caribbean, Food, Restaurants, Desserts"
}

```

Result: Displays the Restaurants with American cuisine in Cleveland and also displays the business id,name,stars and category sorted in descending order

Sorting	by	Category
----------------	-----------	-----------------

Total restaurants

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "Restaurants"} }, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
{$sort:{'stars':-1}}
])

```

Explanation:this query is trying to search for the Restaurants among the businesses in Belmont and projecting the business id,name,stars,category and city name and sorted it in descending order of stars received

Analysis:This query helps the user to find out the top restaurants in the main category in the Yelp webpage.

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "Restaurants"}}, {"city":{$eq:"Belmont"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
  {$sort:{'stars':-1}}
])

YelpBusiness ① 0.886 sec.

/* 1 */
{
  "business_id" : "7xDPzeWjMnUsGFMRmmNwBA",
  "name" : "Mugshots Coffee and Tea",
  "city" : "Belmont",
  "stars" : 5.0,
  "categories" : "Cafes, Breakfast & Brunch, Food, Coffee & Tea, Restaurants"
}

/* 2 */
{
  "business_id" : "rGjHD5Si1WoW_12-IMPa3A",
  "name" : "Ranucci's Big Butt BBQ",
  "city" : "Belmont",
  "stars" : 5.0,
  "categories" : "Food, Restaurants, Street Vendors, Food Trucks, Caterers, Barbeque, Event Planning & Services"
}

/* 3 */
{
  "business_id" : "PXrvCV3zjU60s0_hiuffvA",
  "name" : "Cherubs Cafe",
  "city" : "Belmont",
  "stars" : 5.0,
  "categories" : "Sandwiches, Cafes, Restaurants, Salad"
}

/* 4 */
{
  "business_id" : "zOyCggVF-nzhG4BGh9NLQg",
  "name" : "Purdy Sweet",
  "city" : "Belmont",
  "stars" : 5.0,
  "categories" : "Restaurants, Bakeries, Food"
}

```

Result: Displays the business_id, name, city, stars, categories for each Restaurants sorted in descending order

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "Restaurants"}}, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
  {$sort:{'stars':-1}}
])

```

Explanation: This query is trying to search for the Restaurants among the businesses in Cleveland and projecting the business id, name, stars, category and city name and sorted it in descending order of stars received

Analysis: This query helps the user to find out the top restaurants in the main category in the Yelp webpage.

```

Welcome * db.YelpBusiness.aggregate()
Project localhost:27017 admin
db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "Restaurants" } }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1, "categories":1, "city":1}},
  {$sort:{'stars':-1}}
])

YelpBusiness 0.878 sec.
/* 1 */
{
  "business_id" : "rnvKg9DtsAIHWU4eywsqmA",
  "name" : "Boney Fingers BBQ",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Barbeque, Breakfast & Brunch, Restaurants, Salad, Caterers, Event Planning & Services"
}

/* 2 */
{
  "business_id" : "F7tBX4pRGhWxAxr4ld7dzQ",
  "name" : "Damascino",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Food, Restaurants, Food Delivery Services, Halal, Bakeries, Arabian"
}

/* 3 */
{
  "business_id" : "32DUrg5ThezfileGsWK6Gdw",
  "name" : "Moe's Tavern",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Dive Bars, Nightlife, Bars, Gastropubs, Restaurants, Pubs, Pizza"
}

/* 4 */
{
  "business_id" : "0vvPP6Ir2723kjRZFJg7tg",
  "name" : "Sebastians Deli",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Food, Specialty Food, Meat Shops, Delis, Restaurants"
}

```

Result: Displays the business_id, name, city, stars, categories for the each Restaurants sorted in descending order

Home services

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "home" } }, {"city":{$eq:"Belmont"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1, "categories":1, "city":1}},
  {$sort:{'stars':-1}}
])

```

Explanation: this query is trying to search for the home services among the businesses in Belmont and projecting the business id, name, stars, category and city name and sorted it in descending order of stars received

Analysis: This query helps the user to find out the top home services in the main category in the Yelp webpage.

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "home"} }, {"city":{$eq:"Belmont"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
  {$sort:{'stars':-1}}
])

Yelpbusiness 0.348 sec.

/* 1 */
{
  "business_id" : "6mzdzKxJwoxmsIWewxmHvA",
  "name" : "Ken' Furniture Repair",
  "city" : "Belmont",
  "stars" : 5.0,
  "categories" : "Home Services, Furniture Repair, Local Services, Furniture Reupholstery"
}

/* 2 */
{
  "business_id" : "NMG5mUe55LhWbKknqejkAw",
  "name" : "Carolina ProSweep",
  "city" : "Belmont",
  "stars" : 5.0,
  "categories" : "Chimney Sweeps, Contractors, Home Services, Home Cleaning, Waterproofing, Fireplace Services, Roofing"

}

/* 3 */
{
  "business_id" : "z2Yivr-COKOH2sLNxvnZyg",
  "name" : "Peace -N- Hominy",
  "city" : "Belmont",
  "stars" : 4.5,
  "categories" : "Restaurants, Barbeque, Hot Dogs, Shopping, Breakfast & Brunch, Appliances, Burgers, American (Traditional), Southern, Home & Garden"
}

/* 4 */
{
  "business_id" : "kGpwo-rQ6gy8HKxTjrdmA",
  "name" : "D-Pack Moving",
  "city" : "Belmont",
  "stars" : 4.5,
  "categories" : "Home Services, Movers"
}

```

Result: Displays the business_id, name, city, stars, categories for each home service sorted in descending order

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "home"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
  {$sort:{'stars':-1}}
])

```

Explanation: This query is trying to search for the home services among the businesses in Cleveland and projecting the business id, name, stars, category and city name and sorted it in descending order of stars received

Analysis: This query helps the user to find out the top home services in the main category in the Yelp webpage.

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "home" }}, {"city":{$eq:"Cleveland"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
{$sort:{'stars':-1}}
])

```

YelpBusiness ① 0.36 sec.

```

/* 1 */
{
  "business_id" : "ahLYnXn0Ai6CfvU8xOIJQ",
  "name" : "J & L Country Market",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Christmas Trees, Nurseries & Gardening, Shopping, Food, Fruits & Veggies, Home & Garden, Festivals, Christmas Markets, Specialty Food, Arts & Entertainment"
}

/* 2 */
{
  "business_id" : "FYgYCLODMCbqVvAzRNBUzA",
  "name" : "Fabric Showcase",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Furniture Repair, Furniture Reupholstery, Fabric Stores, Interior Design, Home Services, Shopping, Arts & Crafts, Local Services"
}

/* 3 */
{
  "business_id" : "gFXB0AVBzj32FkWvPfeQg",
  "name" : "Certified Laptop Repair",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Internet Service Providers, Local Services, Mobile Phone Repair, Professional Services, IT Services & Computer Repair, Home Services"
}

/* 4 */
{
  "business_id" : "Ap8ol_gh3xNkndeSbT9SEw",
  "name" : "CJ Trivisonno - Trivisonno Real Estate Team",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Real Estate, Real Estate Agents, Marketing, Home Services, Professional Services, Real Estate Services"
}

```

Result: Displays the business_id,name,city,stars,categories for the each Home service sorted in descending order

Auto services

```

db.YelpBusiness.aggregate([
{$match: {$and :[{$text: { $search: "auto" }}, {"city":{$eq:"Belmont"} }]}},
{$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
{$sort:{'stars':-1}}
])

```

Explanation: This query is trying to search for the auto services among the businesses in Belmont and projecting the business id,name,stars,category and city name and sorted it in descending order of stars received

Analysis:This query helps the user to find out the top home services in the main category in the Yelp webpage.

```

Welcome db.YelpBusiness.aggregate
Project localhost:27017 admin
db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "auto"} }, {"city":{$eq:"Belmont"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
  {$sort:{'stars':-1}}
])
YelpBusiness 0.124 sec.

/* 1 */
{
  "business_id" : "g4E17SxNTjyuQIJ3g6NecA",
  "name" : "TRC Automotive",
  "city" : "Belmont",
  "stars" : 5.0,
  "categories" : "Automotive, Auto Repair, Transmission Repair, Oil Change Stations"
}

/* 2 */
{
  "business_id" : "xA9EJAQF1Vr0K1P6Udr1UA",
  "name" : "Clements Automotive",
  "city" : "Belmont",
  "stars" : 4.0,
  "categories" : "Automotive, Auto Repair"
}

/* 3 */
{
  "business_id" : "WTFDOzWPODYzRDIjzEXwBw",
  "name" : "Firestone Complete Auto Care",
  "city" : "Belmont",
  "stars" : 4.0,
  "categories" : "Auto Repair, Auto Parts & Supplies, Oil Change Stations, Tires, Automotive"
}

/* 4 */
{
  "business_id" : "TgDeUb2hCaSWm8mcEZ1I3g",
  "name" : "Advance Auto Parts",
  "city" : "Belmont",
  "stars" : 3.5,
  "categories" : "Automotive, Auto Parts & Supplies"
}

```

Result: Displays the business_id, name, city, stars, categories for each auto service sorted in descending order

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "auto"} }, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1}},
  {$sort:{'stars':-1}}
])

```

Explanation: This query is trying to search for the auto services among the businesses in Cleveland and projecting the business id, name, stars, category and city name and sorted it in descending order of stars received

Analysis: This query helps the user to find out the top home services in the main category in the Yelp webpage.

```

Welcome * db.YelpBusiness.aggregate
Project localhost:27017 admin
db.YelpBusiness.aggregate([
  {$match: {$and :[{$text: { $search: "auto" }}, {"city":{$eq:"Cleveland"} }]}},
  {$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1,"city":1}),
  {$sort:{'stars':-1}}
])
YelpBusiness 0.129 sec.

/* 1 */
{
  "business_id" : "_5FvUqPqLoNBygO2H10HfA",
  "name" : "B Safe Locksmiths",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Keys & Locksmiths, Commercial Real Estate, Home Services, Real Estate, Auto Parts & Supplies, Automotive"
}

/* 2 */
{
  "business_id" : "yMbsaA9vfHyOyeoScrdy_w",
  "name" : "White Wheel Aligning Service Inc",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Automotive, Auto Repair, Tires, Wheel & Rim Repair"
}

/* 3 */
{
  "business_id" : "l0ArHlP9kNoj1P58Eeaodw",
  "name" : "Broadview Auto Service",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Automotive, Auto Repair, Smog Check Stations, Tires"
}

/* 4 */
{
  "business_id" : "wwH2f3J8mpjXNqobMEQiuA",
  "name" : "Terry's North Coast Auto",
  "city" : "Cleveland",
  "stars" : 5.0,
  "categories" : "Tires, Automotive, Body Shops, Auto Repair"
}

```

Result: Displays the business_id, name, city, stars, categories for the each auto service sorted in descending order

Top	Restaurants	with	wheel	chair	accessibility
-----	-------------	------	-------	-------	---------------

```

db.YelpBusiness.aggregate([
  {$match: {"attributes.WheelchairAccessible":{$eq:"True"}}, {"city":{$eq:"Belmont"}}, { $text: { $search: "Restaurants" } }}),
  {$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1,"city":1,"attributes.WheelchairAccessible":1}},
  {$sort:{'stars':-1}}
])

```

Explanation: This query is trying to search for the restaurants with wheel chair accessibility among the businesses in Belmont and projecting the business id, name, stars, category , city name and wheelchairaccessibilty attribute and sorted it in descending order of stars received
 Analysis: This query helps the user to find out the top businesses in the sub category restaurants in the Yelp webpage which helps people with disabilities to find a place convenient for them.

```

Welcome * db.YelpBusiness.aggregate [Project localhost:27017 admin]
db.YelpBusiness.aggregate([
  {$match: { $and :[({ "attributes.WheelchairAccessible":{$eq:"True"}}, {"city":{$eq:"Belmont"}}, { $text: { $search: "Restaurants"} })]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1,"attributes.WheelchairAccessible":1}},
  {$sort:{'stars':-1}}
])
YelpBusiness 0.991 sec.

/* 1 */
{
  "business_id" : "7xDPzeWjMnUsGFMRmmNwBA",
  "name" : "Mugshots Coffee and Tea",
  "city" : "Belmont",
  "stars" : 5.0,
  "attributes" : {
    "WheelchairAccessible" : "True"
  },
  "categories" : "Cafes, Breakfast & Brunch, Food, Coffee & Tea, Restaurants"
}

/* 2 */
{
  "business_id" : "PXrvCV3ZjU60s0_hiuffvA",
  "name" : "Cherubs Cafe",
  "city" : "Belmont",
  "stars" : 5.0,
  "attributes" : {
    "WheelchairAccessible" : "True"
  },
  "categories" : "Sandwiches, Cafes, Restaurants, Salad"
}

/* 3 */
{
  "business_id" : "ZOyCggVF-nzhG4BGh9NLQg",
  "name" : "Purdy Sweet",
  "city" : "Belmont",
  "stars" : 5.0,
  "attributes" : {
    "WheelchairAccessible" : "True"
  },
  "categories" : "Restaurants, Bakeries, Food"
}

```

Result: Displays the business_id, name, city, stars, categories for each restaurant that have a wheel chair accessibility, sorted in descending order

```

db.YelpBusiness.aggregate([
  {$match: { $and :[({ "attributes.WheelchairAccessible":{$eq:"True"}}, {"city":{$eq:"Cleveland"}}, { $text: { $search: "Restaurants"} })]}},
  {$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1,"city":1,"attributes.WheelchairAccessible":1}},
  {$sort:{'stars':-1}}
])

```

Explanation: this query is trying to search for the restaurants with wheel chair accessibility among the businesses in Cleveland and projecting the business id, name, stars, category , city name and wheelchairaccessibilty attribute and sorted it in descending order of stars received
 Analysis: This query helps the user to find out the top businesses in the sub category restaurants in the Yelp webpage which helps people with disabilities to find a place convenient for them.

```

Welcome * db.YelpBusiness.aggregate([
  {$match: {$and :[{"attributes.WheelchairAccessible":{$eq:"True"}}, {"city":{$eq:"Cleveland"}}, {"$text: { $search: "Restaurants"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1, "categories":1, "city":1, "attributes.WheelchairAccessible":1}},
  {$sort:{'stars':-1}}
])
YelpBusiness 0.972 sec.

/* 1 */
{
  "business_id" : "rnnvKg9DtsAIHWU4eywsqmA",
  "name" : "Boney Fingers BBQ",
  "city" : "Cleveland",
  "stars" : 5.0,
  "attributes" : {
    "WheelchairAccessible" : "True"
  },
  "categories" : "Barbeque, Breakfast & Brunch, Restaurants, Salad, Caterers, Event Planning & Services"
}

/* 2 */
{
  "business_id" : "32DUrg5ThezfileGsWK6Gdw",
  "name" : "Moe's Tavern",
  "city" : "Cleveland",
  "stars" : 5.0,
  "attributes" : {
    "WheelchairAccessible" : "True"
  },
  "categories" : "Dive Bars, Nightlife, Bars, Gastropubs, Restaurants, Pubs, Pizza"
}

/* 3 */
{
  "business_id" : "dhKYHhp136Fzu_RASiTEg",
  "name" : "Cha Spirits & Pizza Kitchen",
  "city" : "Cleveland",
  "stars" : 5.0,
  "attributes" : {
    "WheelchairAccessible" : "True"
  },
  "categories" : "Nightlife, Restaurants, Pizza, American (New), Bars"
}

```

Result: Displays the business_id, name, city, stars, categories for the each restaurant that have a wheel chair accessibility, sorted in descending order

Top restaurants where dogs are allowed

```

db.YelpBusiness.aggregate([
  {$match: {$and :[{"attributes.DogsAllowed":{$eq:"True"}}, {"city":{$eq:"Belmont"}}, {"$text: { $search: "Restaurants"} }]}},
  {$project: {_id:0, "business_id":1, "name":1, "stars":1, "categories":1, "city":1, "attributes.DogsAllowed":1}},
  {$sort:{'stars':-1}}
])

```

Explanation: This query is trying to search for the restaurants where pet dogs are allowed among the businesses in Belmont and projecting the business id, name, stars, category , city name and dogs allowed attribute and sorted it in descending order of stars received

Analysis: This query helps the user to find out the top businesses in the sub category restaurants in the Yelp webpage which helps people with pet dogs to find a place convenient for them to take their pets along with them.

```

db.YelpBusiness.aggregate([
{$match: {$and :[{"attributes.DogsAllowed":{$eq:"True"}}, {"city":{$eq:"Belmont"}}, { $text: { $search: "Restaurants"} }}}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1,"city":1,"attributes.DogsAllowed":1}},
{$sort:{'stars':-1}}
])

```

Yelpbusiness 0.924 sec.

```

/* 1 */
{
  "business_id" : "z2Yivr-C0K0H2sLNxvn2yg",
  "name" : "Peace -N- Hominy Q",
  "city" : "Belmont",
  "stars" : 4.5,
  "attributes" : {
    "DogsAllowed" : "True"
  },
  "categories" : "Restaurants, Barbeque, Hot Dogs, Shopping, Breakfast & Brunch, Appliances, Burgers, American (Traditional), Southern, Home & Garden"
}

/* 2 */
{
  "business_id" : "12kzkf3iw10kiq2niWRbgQ",
  "name" : "Famous Toastery",
  "city" : "Belmont",
  "stars" : 4.0,
  "attributes" : {
    "DogsAllowed" : "True"
  },
  "categories" : "Restaurants, Event Planning & Services, Breakfast & Brunch, American (Traditional), Caterers"
}

/* 3 */
{
  "business_id" : "MRKFi_LGu6EPPrEmxKs3pRg",
  "name" : "Jive Turkey Legs",
  "city" : "Belmont",
  "stars" : 3.5,
  "attributes" : {
    "DogsAllowed" : "True"
  },
  "categories" : "American (Traditional), Restaurants, Sandwiches, American (New), Food Trucks, Food"
}

```

Result: Displays the business_id, name, city, stars, categories for the each restaurant where pets are allowed, sorted in descending order

```

db.YelpBusiness.aggregate([
{$match: {$and :[{"attributes.DogsAllowed":{$eq:"True"}}, {"city":{$eq:"Cleveland"}}, { $text: { $search: "Restaurants"} }}}},
{$project: {_id:0, "business_id":1,"name":1, "stars":1,"categories":1,"city":1,"attributes.DogsAllowed":1}},
{$sort:{'stars':-1}}
])

```

Explanation: This query is trying to search for the restaurants where pet dogs are allowed among the businesses in Cleveland and projecting the business id, name, stars, category , city name and dogs allowed attribute and sorted it in descending order of stars received

Analysis: This query helps the user to find out the top businesses in the sub category restaurants in the Yelp webpage which helps people with pet dogs to find a place convenient for them to take their pets along with them.

```

Welcome * db.YelpBusiness.aggregate...
Project localhost:27017 admin
db.YelpBusiness.aggregate([
  {$match: { $and :[ { "attributes.DogsAllowed":{ $eq:"True"}}, {"city":{$eq:"Cleveland"}}, { $text: { $search: "Restaurants"} } ]}},
  {$project: { _id:0, "business_id":1, "name":1, "stars":1,"categories":1,"city":1,"attributes.DogsAllowed":1}},
  {$sort:{'stars':-1}}
])
YelpBusiness 0.925 sec.

/* 1 */
{
  "business_id" : "32DUrg5ThezfileGsWK6Gdw",
  "name" : "Moe's Tavern",
  "city" : "Cleveland",
  "stars" : 5.0,
  "attributes" : {
    "DogsAllowed" : "True"
  },
  "categories" : "Dive Bars, Nightlife, Bars, Gastropubs, Restaurants, Pubs, Pizza"
}

/* 2 */
{
  "business_id" : "dhKYHhpI36Fzu_RASIITEGR",
  "name" : "Cha Spirits & Pizza Kitchen",
  "city" : "Cleveland",
  "stars" : 5.0,
  "attributes" : {
    "DogsAllowed" : "True"
  },
  "categories" : "Nightlife, Restaurants, Pizza, American (New), Bars"
}

/* 3 */
{
  "business_id" : "26CMx5t6_h5PWT4gcae4zw",
  "name" : "Beviamo Cafe",
  "city" : "Cleveland",
  "stars" : 4.5,
  "attributes" : {
    "DogsAllowed" : "True"
  },
  "categories" : "Lebanese, Food, Juice Bars & Smoothies, Vegan, Middle Eastern, Restaurants, Coffee & Tea"
}

```

Result: Displays the business_id, name, city, stars, categories for each restaurant where pets are allowed, sorted in descending order

Non-text based definition-helpfulness

Query for Cleveland:

Query Description:

We are considering the data in the useful and cool attributes in the yelpReview dataset. We are adding up the values of the two attributes get a total and the review with the highest number of votes can be considered the most useful review .

```

db.yelpBusiness.aggregate([
  {$lookup: { from: "yelpReview", localField: "business_id", foreignField: "business_id", as: "yelpReview" }},
  {$unwind:"$yelpReview" },
  {$match: {"city": { $eq:"Cleveland"} } },
  {$group:{"_id": {"user_id": "$yelpReview.user_id", "business_id": "$business_id"}, "projections":{$push:{ "cool": "$yelpReview.cool", "useful": "$yelpReview.useful" }}} },
  {$project:{ "userId": "$_id.user_id", "business_id": "$_id.business_id", "cool": { $arrayElemAt: [ "$projections.cool", 0]}, "useful": { $arrayElemAt: [ "$projections.useful", 0]}}, "usefulness":{$add:[{$arrayElemAt: [ "$projections.cool", 0]}, {$arrayElemAt: [ "$projections.useful", 0]}]}}
]

```

```
 }},{$sort:{"usefulness": -1}}, { $limit:20}], {"allowDiskUse":true})
```

Summary Results:

Here we get a helpfulness quotient which is the sum of useful and coolness columns as usefulness which defines how useful the review is . these are ordered form highest to lowest.

Query for Belmont

Query Description:

we are considering the data in the useful and cool attributes in the yelpReview dataset. We are adding up the values of the two attributes get a total and the review with the highest number of votes can be considered the most useful review.

```
db.yelpBusiness.aggregate([
{$lookup: { from: "yelpReview", localField: "business_id", foreignField: "business_id", as: "yelpReview" }},
{$unwind: "$yelpReview" },
{$match: {"city": {$eq: "Belmont"}}, {$group: {"_id": {"user_id": "$yelpReview.user_id", "business_id": "$business_id"}, "projections": {$push: {"cool": "$yelpReview.cool", "useful": "$yelpReview.useful"}}, {"$project: {"userId": "$_id.user_id", "business_id": "$_id.business_id", "cool": {$arrayElemAt: [ "$projections.cool", 0 ]}, "useful": {$arrayElemAt: [ "$projections.useful", 0 ]}, "usefulness": {$add: [ {$arrayElemAt: [ "$projections.cool", 0 ]}, {$arrayElemAt: [ "$projections.useful", 0 ]} ]}, {"$sort: {"usefulness": -1}}, {"$limit: 20}], {"allowDiskUse": true}})}
```

```
"usefulNess":{$add:[{$arrayElemAt:      [      "$projections.cool",      0]},{$arrayElemAt:      [
"$projections.useful", 0]}]}
}},{$sort:{"usefulNess":-1}},{ $limit:20}],{"allowDiskUse":true}
```

Key	Value	Type
1 { 2 fields }	{ 6 fields }	Object
_id	{ 2 fields }	Object
userId	GGTF7hnQi6DSW77_qiKlqg	String
business_id	I2zkT3iw10klq2nlWRbgQ	String
cool	32	Int32
useful	48	Int32
usefulNess	80	Int32
2 { 2 fields }	{ 6 fields }	Object
3 { 2 fields }	{ 6 fields }	Object
4 { 2 fields }	{ 6 fields }	Object

Summary Results:

Here we get a helpfulness quotient which is the sum of useful and coolness columns as usefulness which defines how useful the review is . these are ordered from highest to lowest.

Text based definition-helpfulness

Criteria

Review text must contain a string length of more than 300 words assuming helpful reviews will have more details and content. Must contain a meaningful words such as “Good awesome cool helpful best details reservation compulsory”. Will take the documents with test score greater than 0.8.

Query

Description:

This query will get indexed with text column indexed and search for the relevant words, then projects the required fields from yelp review collection along with text score being calculated for the search criteria and string length is measured. Then matches to the text score greater than 0.8 and string length greater than 300.

```
db.YelpReview.aggregate([
```

```

{$match: { $text: { $search: "Good awesome cool helpful best details reservation compulsory"
} } },
{$project: {_id:0, review_id:1, business_id:1,stars:1, text:1, TextScore: { $meta: "textScore" },
StringLength: {$strLenCP:"$text"} }},
{$match: {$and: [{TextScore: { $gt: 0.8 }}, {StringLength: {$gt: 300}}]} }
]).pretty()

```

Summary Results: This gives us a text based searched records having the matched criteria. Assuming helpful reviews will be very long and have some specific words.

Mob Review Behavior – Neo4j

We have performed the neo4j queries with only one city Belmont to make our analysis more simple and comprehensible. Due to limitation in loading millions of data into Neo4j, we have done some ETL tasks using Tableau Prep to Clean and process the data by joining both review and business datasets through the business_id which were separate and took only the details of Restaurant Business in the City Belmont .

Create Statement

```

LOAD CSV WITH HEADERS FROM "file:///C:/YelpDataBelmont.csv" AS YelpBusiness
WITH YelpBusiness where not YelpBusiness.ReviewCount is null
WITH YelpBusiness where not YelpBusiness.BusinessID is null
WITH YelpBusiness where not YelpBusiness.Latitude is null
WITH YelpBusiness where not YelpBusiness.Longitude is null
WITH YelpBusiness where not YelpBusiness.City is null
WITH YelpBusiness where not YelpBusiness.State is null
WITH YelpBusiness where not YelpBusiness.Categories is null
WITH YelpBusiness where not YelpBusiness.Address is null
MERGE(B:Business{Business_id:YelpBusiness.BusinessID,Name:YelpBusiness.Name,Categories:YelpBusiness.Categories,Latitude:YelpBusiness.Latitude,Longitude:YelpBusiness.Longitude,Address:YelpBusiness.Address,city:YelpBusiness.City,state:YelpBusiness.State})
MERGE(Rv:Review{Stars:YelpBusiness.Stars,Review_Count:YelpBusiness.ReviewCount,ReviewDate:YelpBusiness.Date})
MERGE(Rvr:Reviewer{Reviewer_id:YelpBusiness.UserID})
CREATE (Rvr)-[:writes]->(Rv)<-[:have]-(B)

```

neo4j@bolt://localhost:11008 - Neo4j Browser

File Edit View Window Help Developer

```

1 LOAD CSV WITH HEADERS FROM "file:///C:/YelpDataBelmont.csv" AS YelpBusiness
2 WITH YelpBusiness where not YelpBusiness.ReviewID is null
3 WITH YelpBusiness where not YelpBusiness.Latitude is null
4 WITH YelpBusiness where not YelpBusiness.Longitude is null
5 WITH YelpBusiness where not YelpBusiness.City is null
6 WITH YelpBusiness where not YelpBusiness.Categories is null
7 WITH YelpBusiness where not YelpBusiness.Address is null

```

\$ LOAD CSV WITH HEADERS FROM "file:///C:/YelpDataBelmont.csv" AS YelpBusiness WITH YelpBusiness where not YelpBusiness.ReviewID is null WITH YelpBusiness where not YelpBusiness.Latitude is null WITH YelpBusiness where not YelpBusiness.Longitude is null WITH YelpBusiness where not YelpBusiness.City is null WITH YelpBusiness where not YelpBusiness.Categories is null WITH YelpBusiness where not YelpBusiness.Address is null

Added 19551 labels, created 19551 nodes, set 49757 properties, created 29943 relationships, completed after 419679 ms.

Added 19551 labels, created 19551 nodes, set 49757 properties, created 29943 relationships, completed after 419679 ms.

\$:server switch success

Connection updated
You have switched connection.

You are connected as user `neo4j`
to `bolt://localhost:11008`
Connection credentials are stored in your web browser.

Type here to search

1. Reviewers review as a mob query

Belmont

```

MATCH (Rvr)-[:writes]->(Rv)<-[ :have ]-(B)
WITH (Rvr.Reviewer_id) as Reviewers,
COLLECT (B.Business_id) as CommonSet_Of_Business
RETURN DISTINCT CommonSet_Of_Business, COLLECT(Reviewers) as Common_reviewers
ORDER BY CommonSet_Of_Business DESC

```

Explanation

Matched the Reviewer with the Review and the Business through writes and have relations respectively.

Grouped the reviewer id as common set of businesses and collected it as CommonSet_Of_Business. The query returns the distinct common set of businesses and collected the reviewers as common reviewers and sorted in the order of common set of business in descending order.

Output

Displayed the common set of businesses and the common reviewrs who reviewed those businesses.

Geographical proximity query – Neo4j

For a better analysis we divided the dataset into 3 according to the year of review. We took 2015, 2016, 2017 as our reference and compared the overall ratings for each business for each year and tried to compare the change in average rating with time. Since we only took one city for analysis we did not use the latitude or longitude condition since we wanted to compare the restaurants in one city which are in close proximity to each other rather than comparing restaurants from different cities.

```
LOAD CSV WITH HEADERS FROM "file:///C:/YelpData2015.csv" AS YelpBusiness
WITH YelpBusiness where not YelpBusiness.ReviewCount is null
WITH YelpBusiness where not YelpBusiness.BusinessID is null
WITH YelpBusiness where not YelpBusiness.Latitude is null
WITH YelpBusiness where not YelpBusiness.Longitude is null
WITH YelpBusiness where not YelpBusiness.City is null
WITH YelpBusiness where not YelpBusiness.State is null
WITH YelpBusiness where not YelpBusiness.Categories is null
WITH YelpBusiness where not YelpBusiness.Address is null
MERGE(B:Business{Business_id:YelpBusiness.BusinessID,Name:YelpBusiness.Name,Categories:YelpBusiness.Categories,Latitude:YelpBusiness.Latitude,Longitude:YelpBusiness.Longitude,Address:YelpBusiness.Address,city:YelpBusiness.City,state:YelpBusiness.State})
MERGE(Rv:Review{Stars:YelpBusiness.Stars,Review_Count:YelpBusiness.ReviewCount,ReviewDate:YelpBusiness.Date})
MERGE(Rvr:Reviewer{Reviewer_id:YelpBusiness.UserID})
CREATE (Rvr)-[:writes]->(Rv)<-[:have]-(B)
```

```
MATCH (Rv)
SET Rv.Stars=toInteger(Rv.Stars)
```

The screenshot shows the Neo4j Browser interface. The top part displays a Cypher query:

```

1 MATCH (Rvr)-[:writes]->(Rv)<-[::have]-(B)
2 WITH (Rvr.Reviewer_id) as Reviewers,
3     COLLECT (B.Business_id) as CommonSet_Of_Business
4 RETURN DISTINCT CommonSet_Of_Business, COLLECT(Reviewers) as Common_reviewers
5 ORDER BY CommonSet_Of_Business DESC

```

The bottom part shows the results in a table format:

CommonSet_Of_Business	Common_reviewers
["zllWgXyzGWhTG00FK0IUtw", "4EMuinCDgf8C80MKOUzw"]	["KjY-4E_zjTzaYYXzGnoXLw"]
["zllWgXyzGWhTG00FK0IUtw"]	["Xe45W64qdqSN2rzvI0_0A", "LYu1X7RFsGgbDCWZ2Cqng", "qm8xUyPQwON7tjmLe3DxQ", "CoLTNpjAHMzq34aEo3IA"]
["z6ixxpJ8JUW0UF7drl65rC4w", "qbCPF1Z-DheR2MYUyFCX4Q", "gzlItaSAV/MIm60ORlgPMwA", "kfwAkURFdWKRI42hpsimUQ"]	["topsrVlj6m4IX7Z7lb2gw"]
["z6ixxpJ8JUW0UF7drl65rC4w"]	["NDE3cIZldyK60wOBjDHqg"]
["V6Bbg1sxxtS2xC7yyPZmFQ", "kfwAkURFdWKRI42hpsimUQ"]	
["z6ixxpJ8JUW0UF7drl65rC4w"]	["NBX-W2eoK4XzLiwvYJTQCA", "4YOqEhX8GK126GiczLEVgA", "gKSw0glzq36UwK76vMIWg", "Ag56Px7GKhjeQoEoSMe", "kncd-o49xrnHBmEmVC1mMQ", "AiCNL0-E-iOWmpcNB0RGg", "0iClktu60rAsX2hbHzA", "WIxE_0GBKN_3K1syqUR3cA", "7k_Plx9i0ZCRqBokYeDg", "D8mYMYzDxErpTDM8rOrVA", "HMp85fFglyYw70XqyTTtQ", "AXK7nFuJ59LyLPURSWz", "decETHC-xdIyIRibtaXv", "ZXmd5d5AfMmndyBs8XhV_g", "vvnjTSC52j81rJy7D7GCA", "kuoHSFlcME3u_ESfV5yCw", "o-atCCsScWfybuk8uVC_8A", "C9zggQ8b0yIb6sjHWiaJyg", "xAYLSpbbI0ozHcpSikhN5w", "ALBAAzUfSgCJET3ZPQ0-w", "BFJr-japW5RFYMA12Jyg", "pn7PiTOzZpzRcGO_DmC-A", "lBqOx1MUKFAK3SyENk2A", "5XgBa7lyRHmTawSOMRiZg", "GuWz9CcziZsp5TNTB4Q", "VYIL2KdgMwtrnpCM4mE3pg", "gkty-r0gc6o2ZiQ9i4GnsAA", "XUV56Edymif91-XYMgJB9Q", "jhGm8Isd1XnvBpGeOYjs_g", "Nlm-kn3avWNYzN4ray3Xig", "Xq_MovX1NhiCzDpnhfMPg", "XK2mK0LV_RGIP4abgT_QA", "DrnOglRFvghncst-B1dW1Q", "90qzD3NDHmAyDar7QPg9UA", "azmBvRmViHd80fC8rCQCw"]

Started streaming 83 records after 1074 ms and completed after 1086 ms.

Analysis

We did the query to find out reviewers who reviewed a common set of businesses. We did the analysis on a single category “Restaurants” and found out that there are groups of people who reviewed as a mob. This shows the natural tendency of people to follow a mob which has the same taste of choice.

Belmont year 2015

```

MATCH (Rvr)-[:writes]->(Rv)<-[::have]-(B)
WITH B.Name as BusinessName, AVG(Rv.Stars) as RatingCount,B.city as Cityname,B.state as
Statename,COLLECT(DISTINCT Rv.ReviewDate) as dateof_review
RETURN DISTINCT(BusinessName) as Businesses,RatingCount as
TotalRating,Cityname,Statename,dateof_review
ORDER BY TotalRating DESC

```

Output FOR THE YEAR 2015

```

MATCH (Rvr)-[:writes]->(Rv)<-[::have]-(B)
WITH B.Name as BusinessName, AVG(Rv.Stars) as RatingCount,B.city as Cityname,B.state as Statename,COLLECT(DISTINCT Rv.ReviewDate) as dateof_review
RETURN DISTINCT(BusinessName) as Businesses,RatingCount as TotalRating,Cityname,Statename,dateof_review

```

Businesses	TotalRating	Cityname	Statename	dateof_review
"Wendy's"	3.0	"Belmont"	"NC"	["01-01-2015", "01-05-2015", "01-10-2015", "01-14-2015", "01-17-2015", "01-21-2015", "01-24-2015", "01-14-2015", "02-25-2015", "03-06-2015", "02-21-2015", "03-08-2015", "03-12-2015", "03-17-2015", "03-20-2015", "03-22-2015", "03-27-2015", "03-31-2015", "04-02-2015", "04-06-2015", "04-15-2015", "04-20-2015", "04-29-2015", "05-02-2015", "05-03-2015", "05-06-2015", "05-07-2015", "05-13-2015", "05-15-2015", "05-16-2015", "05-18-2015", "06-20-2015", "06-23-2015", "05-24-2015", "05-29-2015", "05-31-2015", "06-02-2015", "06-04-2015", "06-06-2015", "06-12-2015", "06-22-2015", "06-28-2015", "06-30-2015", "07-02-2015", "07-04-2015", "07-08-2015", "07-09-2015", "07-13-2015", "07-18-2015", "07-19-2015", "07-24-2015", "08-04-2015", "08-02-2015", "08-03-2015", "08-08-2015", "08-11-2015", "08-16-2015", "08-19-2015", "08-21-2015", "08-25-2015", "08-27-2015", "08-29-2015", "08-30-2015", "09-06-2015", "09-11-2015", "09-12-2015", "09-14-2015", "09-22-2015", "09-30-2015", "10-01-2015", "10-07-2015", "10-13-2015", "10-15-2015", "10-19-2015", "10-20-2015", "10-31-2015", "11-08-2015", "11-22-2015", "11-30-2015", "12-05-2015", "12-06-2015", "12-15-2015", "12-16-2015", "12-19-2015", "12-21-2015", "12-22-2015", "02-08-2016", "03-14-2015", "03-28-2015", "04-13-2015", "04-26-2015", "04-30-2015", "05-05-2015", "05-19-2015", "05-27-2015", "05-30-2015", "06-07-2015", "07-29-2015", "08-30-2015", "09-19-2015", "11-13-2015", "11-16-2015", "11-30-2015", "01-29-2015", "01-28-2015", "05-08-2015", "07-31-2015", "08-25-2015", "07-03-2015", "12-10-2015", "06-24-2015", "01-06-2015", "05-29-2015", "04-04-2015", "05-04-2015", "07-14-2015", "05-09-2015", "12-12-2015", "05-14-2015", "07-01-2015", "09-21-2015", "08-06-2015", "03-26-2015", "04-23-2015", "11-11-2015", "08-20-2015", "12-23-2015", "11-26-2015", "01-25-2015", "08-24-2015", "02-09-2015", "10-21-2015", "02-06-2015", "01-04-2015", "01-08-2015", "01-11-2015", "01-16-2015", "01-18-2015", "01-22-2015", "02-15-2015", "03-08-2015", "03-16-2015", "03-26-2015", "04-16-2015", "04-18-2015", "04-21-2015", "04-24-2015", "06-01-2015", "05-08-2015", "05-12-2015", "06-10-2015", "06-15-2015", "06-14-2015", "06-18-2015", "06-23-2015", "06-28-2015", "06-29-2015"]

Started streaming 21 records after 216 ms and completed after 216 ms.

```

MATCH (Rvr)-[:writes]->(Rv)<-[::have]-(B)
WITH B.Name as BusinessName, AVG(Rv.Stars) as RatingCount,B.city as Cityname,B.state as Statename,COLLECT(DISTINCT Rv.ReviewDate) as dateof_review
RETURN DISTINCT(BusinessName) as Businesses,RatingCount as TotalRating,Cityname,Statename,dateof_review

```

Businesses	TotalRating	Cityname	Statename	dateof_review
"Avi's Touch of Italy"	3.0	"Belmont"	"NC"	["01-22-2015"]
"Olemya Premium Pub"	3.0	"Belmont"	"NC"	["10-06-2015"]
"Purdy's Sweet Belmont Soda Fountain"	3.0	"Belmont"	"NC"	["06-11-2015"]
"Whit's Restaurant"	3.0	"Belmont"	"NC"	["07-26-2015"]
"Belmont Foot & Beverage"	3.0	"Belmont"	"NC"	["11-20-2015"]
"Sammy's Deli & Neighborhood"	4.0000000000000007	"Belmont"	"NC"	["10-14-2015", "05-10-2015", "05-26-2015"]

Started streaming 21 records after 216 ms and completed after 216 ms.

Output:

Businesses, TotalRating, Cityname, Statename and a collection of dateof_review is displayed for each business per year.

OUTPUT QUERY for 2016

```

MATCH (Rvr)-[:writes]->(Rv)<-[::have]-(B)
WITH B.Name as BusinessName, AVG(Rv.Stars) as RatingCount,B.city as Cityname,B.state as Statename,COLLECT(DISTINCT Rv.ReviewDate) as dateof_review
RETURN DISTINCT(BusinessName) as Businesses,RatingCount as TotalRating,Cityname,Statename,dateof_review

```

ORDER BY TotalRating DESC

Output:

Businesses, TotalRating, Cityname, Statename and a collection of dateof_review is displayed for each business per year.

Output query for 2017

```
MATCH (Ryr)-[:writes]->(Ry)<-[ :have ]-(B)
```

```
WITH B.Name as BusinessName, AVG(Rv.Stars) as RatingCount,B.city as Cityname,B.state as Statename.COLLECT(DISTINCT Rv.ReviewDate) as dateof_review
```

```
RETURN          DISTINCT(BusinessName)      as      Businesses,RatingCount      as  
TotalRating,Cityname,Statename,dateof review
```

ORDER BY TotalRating DESC

The screenshot shows three separate windows of the Neo4j browser, each displaying a different part of the query and its results.

```

MATCH (Rvr)-[:writes]->(Rv)<-[have]-(B)
WITH B.Name as BusinessName, AVG(Rv.Stars) as RatingCount,B.city as Cityname,B.state as Statename,COLLECT(DISTINCT Rv.ReviewDate) as dateof_review
RETURN DISTINCT(BusinessName) as Businesses,RatingCount as TotalRating,Cityname,Statename,dateof_review
    
```

Businesses, TotalRating, Cityname, Statename, dateof_review

Businesses	TotalRating	Cityname	Statename	dateof_review
"Wendy's"	5.0	"Belmont"	"NC"	["03-26-2017", "10-07-2017", "04-02-2017", "01-04-2017", "01-05-2017", "01-07-2017", "01-19-2017", "01-25-2017", "02-29-2017", "03-17-2017", "03-28-2017", "05-29-2017", "06-09-2017", "03-02-2017", "03-03-2017", "03-08-2017", "03-15-2017", "03-20-2017", "03-22-2017", "03-25-2017", "03-26-2017", "04-02-2017", "04-06-2017", "04-13-2017", "04-12-2017", "04-13-2017", "04-15-2017", "04-19-2017", "04-20-2017", "04-23-2017", "04-25-2017", "04-26-2017", "04-28-2017", "04-30-2017", "05-01-2017", "05-02-2017", "05-03-2017", "05-13-2017", "05-19-2017", "05-22-2017", "05-29-2017", "06-01-2017", "06-04-2017", "06-06-2017", "06-08-2017", "06-10-2017", "06-11-2017", "06-15-2017", "06-14-2017", "06-16-2017", "06-22-2017", "06-25-2017", "06-24-2017", "07-03-2017", "07-15-2017", "07-17-2017", "07-19-2017", "07-22-2017", "08-02-2017", "08-08-2017", "08-13-2017", "08-17-2017", "08-19-2017", "08-20-2017", "08-26-2017", "08-08-2017", "08-15-2017", "09-21-2017", "09-06-2017", "03-15-2017", "03-25-2017", "11-05-2017", "11-15-2017", "11-18-2017", "11-22-2017", "11-23-2017", "11-28-2017", "11-29-2017", "12-06-2017", "12-13-2017", "01-19-2017", "02-08-2017", "03-27-2017", "05-15-2017", "06-20-2017", "08-11-2017", "11-16-2017", "12-12-2017", "12-16-2017", "12-23-2017", "03-10-2017", "07-22-2017", "07-31-2017", "09-09-2017", "02-06-2017", "03-17-2017", "06-24-2017", "07-09-2017", "02-04-2017", "11-16-2017", "04-27-2017", "12-28-2017", "08-21-2017", "03-30-2017", "01-29-2017", "03-03-2017", "08-21-2017", "01-03-2017", "09-12-2017", "01-24-2017", "01-27-2017", "02-04-2017", "02-07-2017", "02-12-2017", "02-15-2017", "02-16-2017", "02-25-2017", "02-22-2017", "02-27-2017", "03-14-2017", "03-18-2017", "04-05-2017", "04-07-2017", "04-14-2017", "04-16-2017", "04-21-2017", "04-22-2017", "04-24-2017", "05-08-2017", "05-11-2017", "05-12-2017", "05-17-2017", "05-18-2017", "05-20-2017", "05-21-2017", "05-23-2017", "05-25-2017", "05-30-2017", "05-01-2017", "06-03-2017", "06-07-2017", "06-15-2017", "06-25-2017", "06-29-2017"]


```

MATCH (Rvr)-[:writes]->(Rv)<-[have]-(B)
WITH B.Name as BusinessName, AVG(Rv.Stars) as RatingCount,B.city as Cityname,B.state as Statename,COLLECT(DISTINCT Rv.ReviewDate) as dateof_review
RETURN DISTINCT(BusinessName) as Businesses,RatingCount as TotalRating,Cityname,Statename,dateof_review
    
```

Businesses, TotalRating, Cityname, Statename, dateof_review

Businesses	TotalRating	Cityname	Statename	dateof_review
"Lunarmobra"	5.0	"Belmont"	"NC"	["01-26-2017", "10-15-2017", "10-01-2017", "03-27-2017"]
"Peace AI"	5.0	"Belmont"	"NC"	["01-31-2017", "04-21-2017", "06-07-2017", "08-19-2017", "09-08-2017", "09-11-2017", "10-12-2017", "04-22-2017"]
"Honey Q"				
"Ave Turkey"	5.0	"Belmont"	"NC"	["07-25-2017"]
"Legit"				
"Cherubic"	5.0	"Belmont"	"NC"	["03-25-2017"]
"Cafe"				
"Inaction"	5.0	"Belmont"	"NC"	["04-16-2017"]
"Buying To Go"	5.0	"Belmont"	"NC"	["04-03-2017"]
"Germany Premium Pub"	4.5	"Belmont"	"NC"	["12-05-2017", "04-16-2017", "07-30-2017", "03-08-2017", "03-12-2017"]
"Mugshot Coffee and Tea"	4.5	"Belmont"	"NC"	["03-18-2017", "03-30-2017", "04-19-2017", "07-05-2017"]
"New China"	4.0	"Belmont"	"NC"	["02-14-2017", "06-27-2017", "04-16-2017", "09-18-2017"]
"Joe's Touch of Asia"	4.0	"Belmont"	"NC"	["03-25-2017", "12-23-2017"]

Output:

Businesses, TotalRating, Cityname, Statename and a collection of dateof_review is displayed for each business per year.

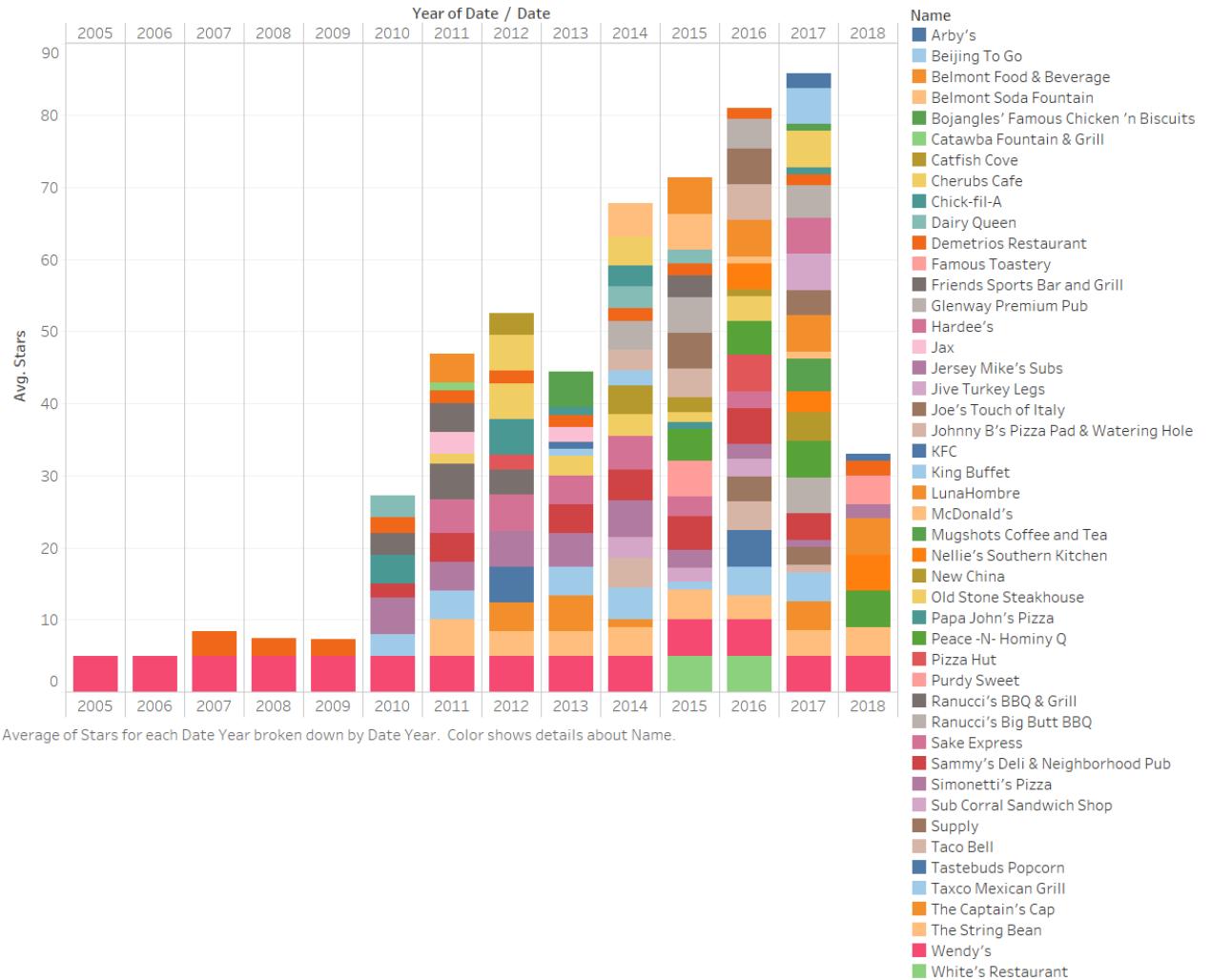
Output Explanation for year 2015, 2016, 2017

Matched the Reviewer and Review nodes through writes and have relations respectively. Grouped the fields by business name, took average of stars as rating count, city name as Cityname, state ad Statename, collected distinct values of Reviewdate as dateof_review. Returned distinct business

names as Businesses, ratingcount as TotalRating, Cityname, Statename and dateof_review and ordered the results in descending order of Totalrating.

Analysis and Visualization

Belmont-ReviewCount v/s proximity v/s time



Belmont-ReviewCount v/s proximity v/s time



Average of Stars for each Date Year broken down by Name. Color shows details about Name. The data is filtered on maximum of Review Count, which ranges from 3 to 323. The view is filtered on Name, which keeps Wendy's.

Analysis

After doing the queries we did a visualization in tableau to get a better picture of the change in metrics. We observed that the average of the rating for each business is slowly increasing from 2005 and once it reaches 2009, it increases abruptly due to addition of more businesses. In 2013 it again drops and then increases progressively till 2017. In 2018 it drops to a much lower value due to incomplete data on reviews. Out of all the Restaurants, Wendy's (marked in pink) is showing the highest average rating of 5 consistently for all the years and can be inferred that it is the restaurant which showed a uniform performance over time. The average rating for each restaurants which are located in a particular geographic radius is increasing over the years for the city Belmont.

The second graph shows the average star rating of Wendy's over a period of 14 years with a consistent value 5.

Identifying Fake Reviews – SAS (Statistical Analytical System)

Fake Review Criteria

Text based definition of fake reviews:

A fake review has certain characteristics that distinguish it from a genuine review. A review could potentially be a fake review when it has less content i.e. the number of words are limited and is not descriptive of the place/product being reviewed. The review could be limited to adjectives like good, bad, mediocre, etc. which are very generic in nature. It does not contain content pertaining to the specific location or the product and it could contain a lot of pronouns that include he, she, I, you, etc. that do not add any value to the usefulness of the review. For example we could consider a restaurant of Mexican cuisine for a review. (we are considering positive tone for this example).

A descriptive review would be: “the tacos were extremely great and the bartender made great cocktails. service was extremely quick and the staff was very caring. Our family enjoyed the live music in the evening while having dinner at the restaurant ”.

A fake review could be: “this is the best place in the city . I went with my wife and had a great time. Will go back to this place very soon”.

Also, reviews with extreme ratings (very high/very low) when compared to the average or median ratings could be more biased and not genuine. Users who flood a bunch of places with similar type of reviews could potentially be writing reviews that are not true.

Input File:

We have imported a dataset of a total of around 20,000 records with extreme low reviews (star rating less than or equal to two) and extreme high rating (star rating greater than or equal to 4).

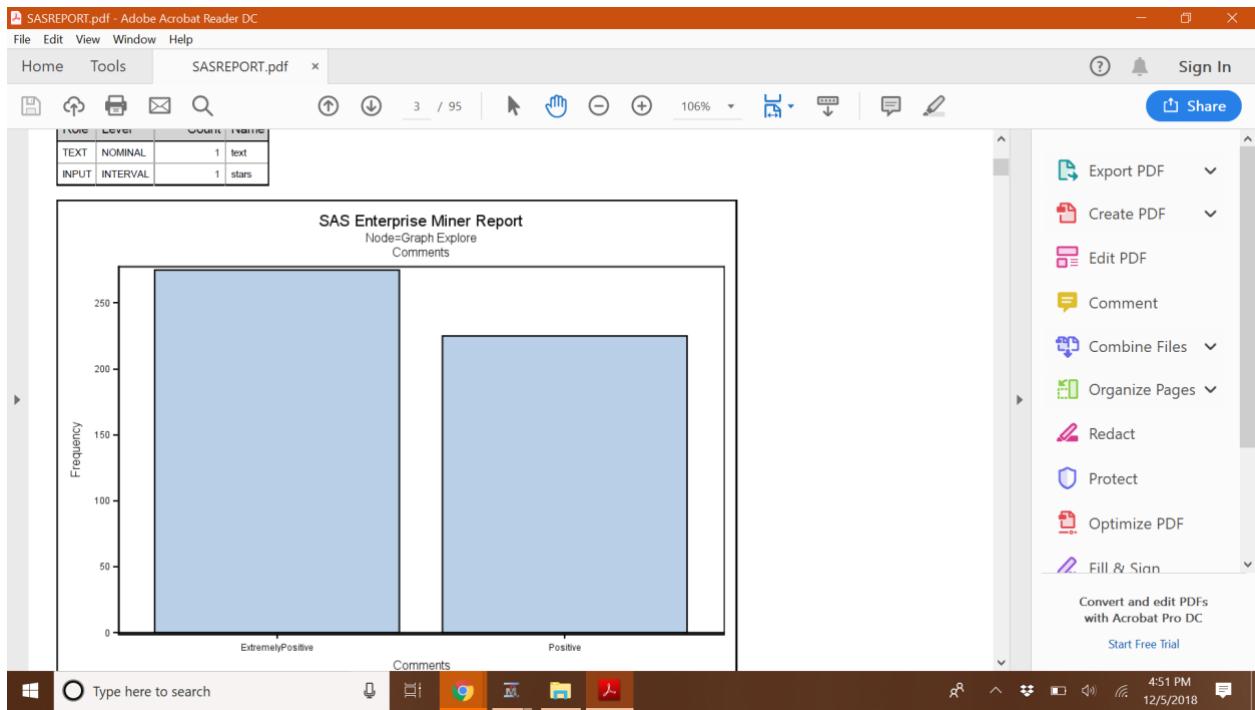
The input File consists of four variables: userID(ID role on SAS), ReviewText(Text Role on SAS), Stars(input Role on SAS), Comments(Target Role).

The review text which has the text role on SAS is the variable on which the Text mining is performed. We have taken the corresponding review text for each of the user ID. The comments variable contains the following comments: Extremely positive(rating of 5) , positive(rating 4), extremely negative(rating 1), negative(2). This will be our target variable on SAS. Our prediction will be based on this variable and we will compare the results after the analysis.

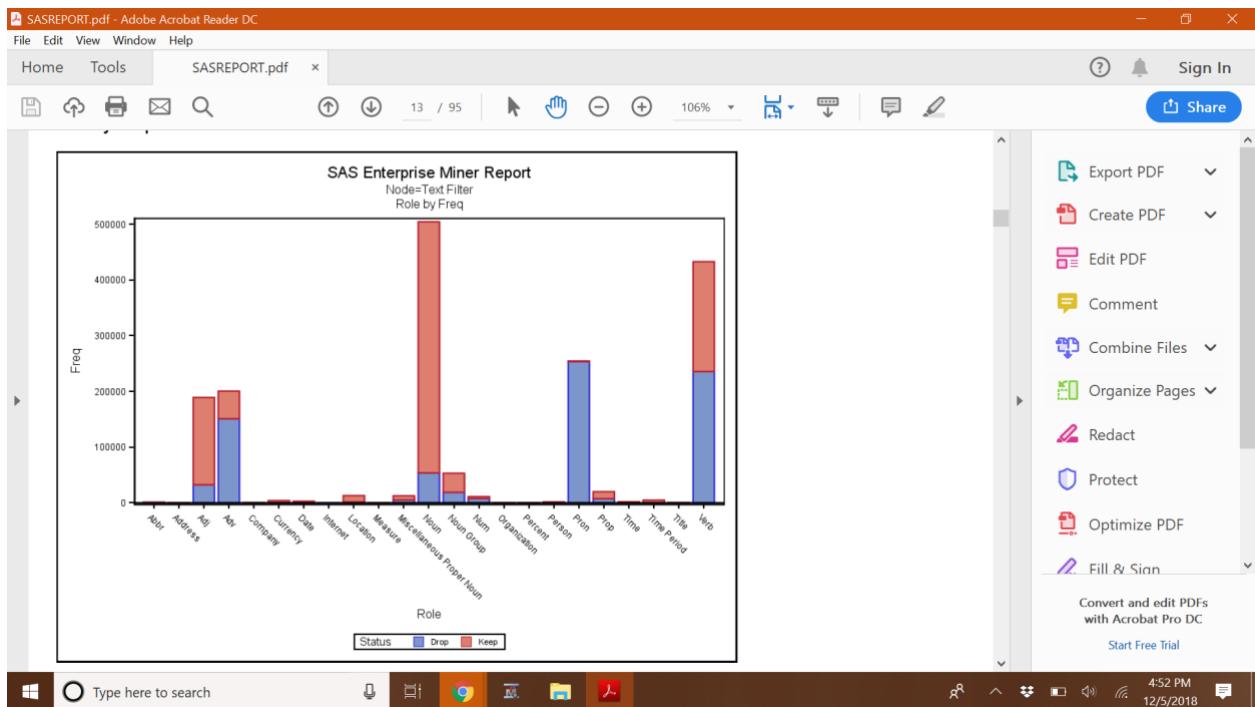
Dictionaries used:

Analysis Path and Analysis Results:

We have imported the data and converted to SAS table called IAST and used a graph explore node to explore the data graphically. We have used a text parsing node to do the tokenization of data and used a text filter to get the most important terms filtered out and we are creating a cluster node for positive reviews (rating 4 and 5) and negative reviews (rating 1 and 2) to understand the groups of data that occur most frequently. We have a text topic in parallel to the cluster Node. From the text topic we understand the topics and the number of documents those topics are found in. We have used a user created a text topic node that is user created with our topics that contain extreme positive or extreme negative words. We see that the weightage of these topics on the overall documents is less compared to the system generated topics. The documents that contain the topics with extreme words could possibly be fake reviews since they do not look balanced and are inclined to extremity and bias. Further more we have used two models random Forest and MBR to evaluate our assumptions. We also have used a model comparison and score nodes.



From the graph explore we can see the ratio of extreme positive vs positive reviews.



On applying the text filter we see that the terms marked in blue are dropped from previous text parsing. A majority of the words are grouped as nouns while the adjectives are really less number comparatively.

Node=userTopic_P

Topics

Category	Topic ID	Document Cutoff	Term Cutoff	Topic	Number of Terms	# Docs
User	1	0.001	0.001	abysmal angry banal rude impolite	1	76
User	2	0.001	0.001	awful terrible	2	477
User	3	0.001	0.001	horrible disgusting worthless too long	1	630
Multiple	4	0.007	0.013	+chicken,+order,+food,+sauce,rice	370	10937
Multiple	5	0.007	0.013	+time,first,first time,last,+year	252	10890
Multiple	6	0.008	0.013	+room,+hotel,+stay,+bed,+park	234	7544
Multiple	7	0.009	0.012	+pizza +crust +good,+order,+topping	104	6776
Multiple	8	0.008	0.013	+salad,+sandwich,+lunch,+soup,+menu	309	13128
Multiple	9	0.008	0.013	+coffee,+shop,+tea,+cup,+coffee shop	213	7775
Multiple	10	0.006	0.013	+place,+hair,+nail,+good,+steak	318	10971
Multiple	11	0.007	0.013	+beer,+selection,+bar,+wine,+patio	218	9930
Multiple	12	0.007	0.012	+cupcake,+cake,+frosting,+flavor,+cookie	167	10967
Multiple	13	0.008	0.013	+dim,+sum,+dish,+dim sum,+menu	318	12271

In the above screenshot we see the user given topics that have been given by us to indicate the fake reviews . these topics are present in approximately 1200 documents. There is a good chance that the documents containing these terms are fake, This is in the negative perspective

We have designed two web pages in the reviewer's perspective and business owner's perspective. To accomplish this task we have used the results that we could derive from queries based on various metrics as available in section Data Analysis Section which include analysis using MongoDB, Neo4j and SAS. These metrics that appear on the web pages can be used in real time to help reviewer's and business owners understand the statistics of their reviews and businesses respectively .

Webpage Design

Based on the results and inferences obtained from the analytics done in MongoDB, Neo4j and SAS, the team decided to build a webpage design using the informative metric to provide a more interactive webpage for the business owner and the user.

Business owner's webpage design

The screenshot shows the Yelp Business Owners interface for a restaurant named "Truly Delicious".

Header: Yelp Business Owners, Your business, Account Settings, Support, Inbox, User profile icon.

Business Info: Truly Delicious, 334 North Springs Blvd, Boston, MA, 37 Reviews, 312 Rank, Restaurants category.

Left Sidebar (Activity): Activity, Yelp Ads, Page Upgrades, Call to Action, Business Information, **Reviews** (selected), Photos and Videos, Inbox, Check-in Offers, Deals & Gift Certificates.

Ratings Summary: A pie chart showing review distribution: 5-Star (27%), 4-Star (21.6%), 3-Star (21.6%), 2-Star (13.5%), 1-Star (8%). Overall rating: 4.4.

Reviews (37): Shows a review by Annie L. (Elite '17) from May 14, 2018, with a link to "Continue Reading".

Your Business' Top Reviewers: Annie L, Carol Smith, David William, Chris Paul, Mike Hannigan.

Your Business' Best Review: David Smith (Elite '17) from Nov 15, 2017, with a link to "Continue Reading".

Similar Businesses Near by: 1. Poor Calvin's (510 Piedmont Avenue, Downtown, (404) 254-4051, 5 stars), 2. Aviva (276 Peachtree, Downtown, (420) 314-1231, Juice & Smoothies, Bar, 5 stars).

Map: A map showing the location of Truly Delicious and nearby businesses marked with pins.

The current web design of Yelp review webpage for Business owners only provides information of the reviews that have been written for that business. But according to our analysis, we felt there are many items that could be added to make it informative for the owner. Firstly, a rating breakdown of the what has been received in the form of a pie chart has been placed. After which, the Reviews can be viewed, one review is placed to show the owner that he can view others. Below that we see that the business owner can also view businesses that are near to his/her own. On the right hand side, the owner can view the top

reviewer's for the business and below that block, a block with the best review for the business has been placed and the owner has the option to personally thank that reviewer.

User's webpage design

The screenshot shows a Yelp user profile for "John Doe" located in Pittsburgh, PA. He has 42 reviews and is ranked 134. The profile includes a placeholder icon for a photo, a list of 230 friends, and links to profile overview, friends, reviews, and other account features.

My Reviews (42)

A review for "Pizza Place" at 20 Cumming St, Alpharetta, GA, dated Oct 12, 2014, is highlighted. It reads: "Rated 1/5 This is the worst Pizza Hut I have been to yet. Bad service and terrible food". It was reviewed by 5 others. A "See More ..." link is present.

Helpful Review of the Day

Annie L. (734 reviews, Elite '17) wrote a review for "Truly Delicious" on May 14, 2018. The review text: "I came here last night to give this place another try; as I had mentioned before, we truly wanted to love this place after hearing so many rave reviews. So, why not give this place another go? Continue Reading". A "See More ..." link is present.

Review Restaurants Near by?

Two nearby restaurants are listed: "Poor Calvin's" (510 Piedmont Avenue, Downtown, (404) 254-4051) and "Aviva" (276 Peachtree, Downtown, (420) 314-1231). Each has a star rating and a "See More ..." link.

Reviewer Rankings

A list of top reviewers: 1. Annie L., 2. Carol Smith, 3. David William, 4. Chris Paul, 5. Mike Hannigan.

My Ratings Breakdown

Total Reviews: 42

Rating	Count
5 stars	25
4 stars	7
3 stars	5
2 stars	0
1 star	7

Map

A map of the area shows several location markers corresponding to the nearby restaurants listed.

Here again, in the current iteration Yelp's reviewer page, the reviewer can only view the reviews he/she has written. But from the queries run and analysis done, the following features have been added. On the top of the page, the reviewer can see his summary as a reviewer, which include his/her rank. Along with the review he or she has written, the reviewer can view how many others have reviewed a particular business, something that isn't there currently. Below that the reviewer can view a Helpful query of the day, which might help or influence them to check out that business. Below that, the reviewer is encouraged to review businesses that are close in the vicinity. On the right hand side, a reviewer can view the top ranked reviewers and blow that there is a block summarizing all the ratings a reviewer has given in the form of a histogram.

Conclusion

In this term project first we have created the database schemas for both mongodb and Neo4j as an initial step. Indexes and constraints are created for the mongoDB pseudo. Then data is evaluated and separate collections are created for each dataset: Business, Review, users, Tip, Photos, Check In. We narrowed down our analysis on two cities: Cleveland and Belmont and ran the initial set of queries for collecting data pertaining to various metrics to get an overview of the datasets. Then we did a detailed analysis of metrics that we collected in the prior queries in both Mongodb and Neo4j to draw insights from them to add value to the business. As the last step of analysis we did a text mining using SAS Enterprise Miner to determine the fake reviews in the review text. As the last step we created a modified design of Yelp Business page as well as Yelp Review page with added features which we derived from the analysis.