**Big Data Engineering Tutorial 2일차**

3조 신행철 백규진 이소연

1. **Business TABLE**

* Create **EXPLODED** table with flattened categories  
  Since our file contains an array of categories we need to flatten those categories in order to be able to query them easily. We use LATERAL VIEW explode function for the column **categories** and alias it as **cat\_exploded**

|  |
| --- |
| CREATE TABLE EXPLODED AS  SELECT \* FROM business  LATERAL VIEW explode(categories) business AS cat; |

* **Q1.1** List the values of the columns business\_id, categories, cat\_exploded from the table EXPLODED with 10 records selected:

|  |
| --- |
| SELECT business\_id, categories, cat FROM EXPLODED limit 10; |

* **Q1.2** Find total number of businesses in the data set:

|  |
| --- |
| SELECT count(business\_id) from business; |

1. **Restaurant TABLE**

* Create restaurant table from the table EXPLODED with the column cat\_exploded="Restaurants":

|  |
| --- |
| CREATE TABLE restaurant\_exploded AS  SELECT \* FROM EXPLODED  Where cat = “Restaurants”; |

* **Q2.1.** List 5 records with the columns **name, review\_count, stars, cat\_exploded category** from your restaurants table created - **cat\_exploded** column is renamed as **category** :

|  |
| --- |
| SELECT name, review\_count, stars, cat  FROM restaurant\_exploded limit 5; |

* **Q2.2.** query columns [name and nested object (attributes > ambience > romantic) of restaurants table:   
  a) To select nested columns it worselect \* from review;ks as following: parent.child.grandchild

|  |
| --- |
| SELECT name, attributes.ambience.romantic  FROM restaurant\_exploded limit 5; |

* **Q2.3** find the number of restaurants in the dataset

|  |
| --- |
| SELECT count(business\_id) FROM restaurant\_exploded; |

**3. Review TABLE**

* Create Review\_Filtered Table
  + using JOIN for **review** and **restaurants** with the same business\_id
  + NOTE: In order to improve the performance, do you prefer **review** JOIN **restaurant** or the other way?

|  |
| --- |
| create table review\_filtered as  select A.review\_id, count(\*) from review\_p A join restaurants B on  (A.business\_id=B.business\_id) group by A.review\_id; |

* **Q3.1.** find the number of restaurant reviews from Review\_Filtered Table

|  |
| --- |
|  |

**4. Users TABLE**

* User table 에서 find the number of all distinct users  in dataset:

|  |
| --- |
| SELECT COUNT(DISTINCT user\_id) FROM users;  //Hadoop 부하를 줄이려면 아래 쿼리문 사용  SELECT count(\*) FROM (SELECT DISTINCT user\_id FROM users) a; |

**5. Users/Elite TABLE**

**Q5.1.** select 4 records with the COLUMN **elite** from **USERS** table, which is generated from user.json to see elite users: NOTE: table should be USERS not USER

|  |
| --- |
| Select \* from users limit 4; |

**Q5.2.** Create **Elite\_Users** Table by exploding **elite** column as **elite\_year** column using LATERAL VIEW in order to separate elite users: For example, [2016, 2017, 2018] => 3 exploded records as follows:

|  |
| --- |
| CREATE TABLE elite\_users AS  SELECT \* FROM users\_p  LATERAL VIEW explode(elite) users\_p AS yr;  SELECT \* from elite\_users where yr in(2016,2017,2018); |

**6. Create Tip Table**

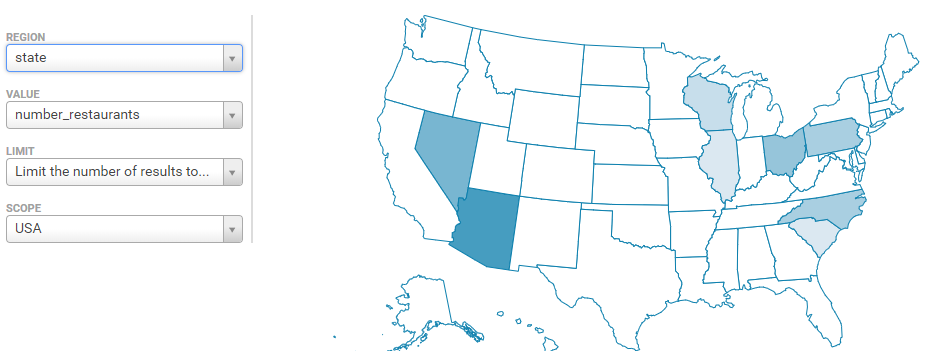
|  |
| --- |
| DROP table `yelp\_db`.`tip` ;  CREATE external TABLE IF NOT EXISTS `yelp\_db`.`tip` (  `user\_id` STRING,  `business\_id` STRING,  `text` STRING,  `date` STRING,  `likes` INT)  ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  LOCATION 's3://yelp-com/tip'; |

**7. 데이터 분석 심층 및 시각화**

1) In order to map restaurants across United, Select the columns (**state**, count of **business\_id** as **number\_restaurants** ) by grouping data of **restaurants** table by STATE and then order it by **number\_restaurants**

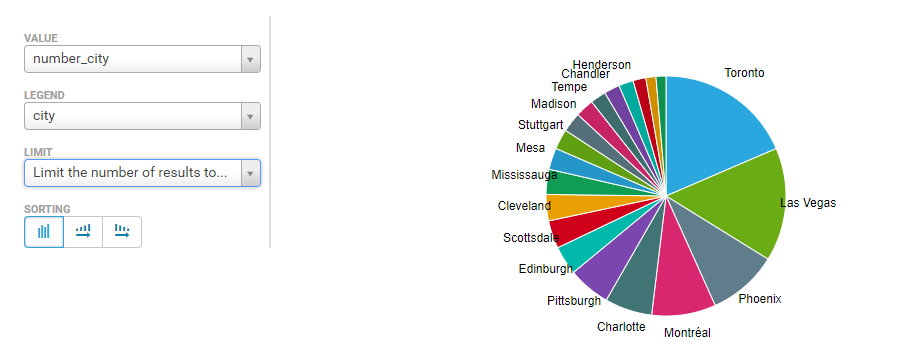
**Q7.1:** 시각화

* 시각화: marker map



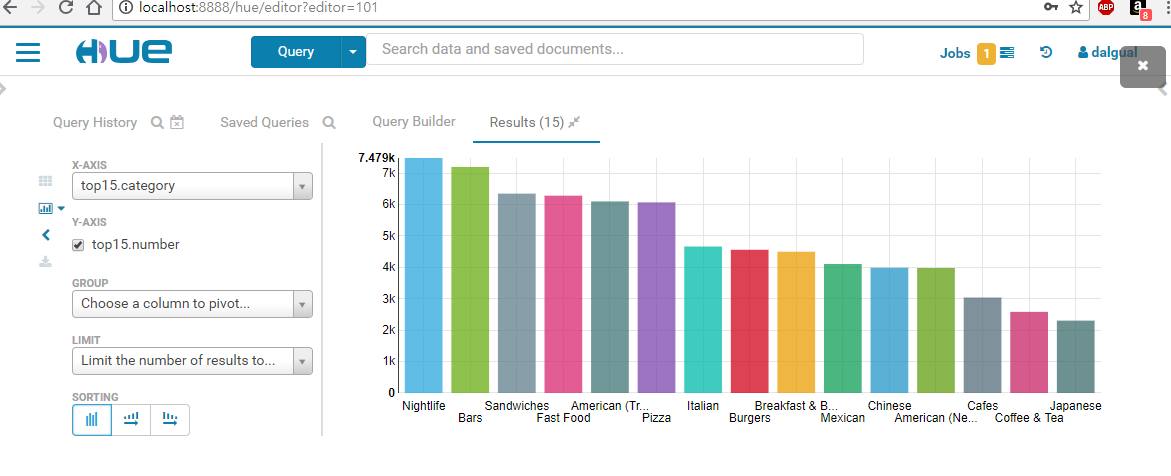
|  |
| --- |
| select state, avg(latitude) as lat, avg(longitude) as lon, count(\*) as cnt  from restaurants  group by state; |

2) Which Cities Have The Highest Number Of   Restaurants? In order to map restaurants across United, Select the columns (**city**, count of **business\_id** as **number\_city**) by grouping data of restaurants table by **city** and then order it by **number\_city**



|  |
| --- |
| select city, count(distinct business\_id ) as num  from restaurants group by city order by num desc ; |

3) Find out Top 15 Sub-Categories Of  Restaurants from tables **exploded**  and **restaurants** with **business\_id** and grouping by **cat\_exploded** column of the table **exploded,** which are not in("Restaurants","Food")



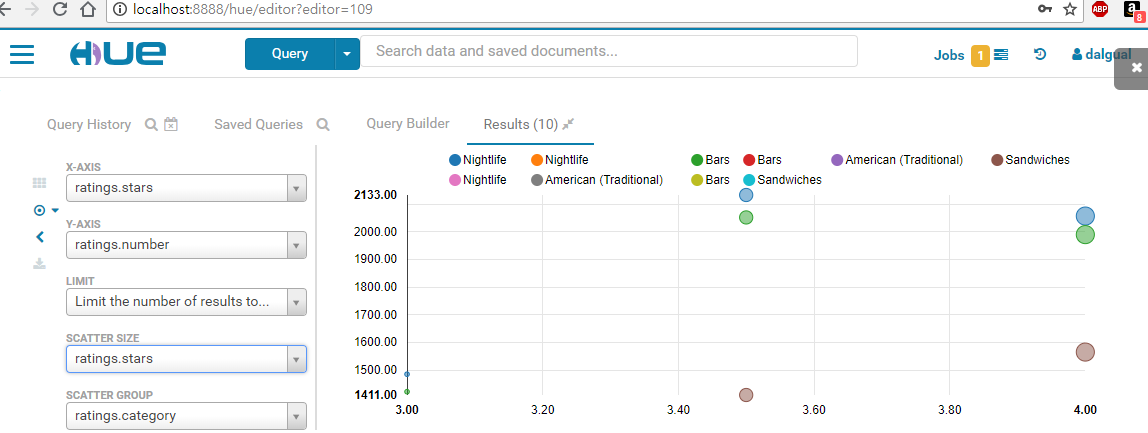
|  |
| --- |
| select \* from (  select cat, count(\*) as cnt, rank() OVER(order by count(\*) desc) as rnk  from business\_exploded a  join (select business\_id from business\_exploded where cat = 'Restaurants') b on (a.business\_id = b.business\_id)  where cat not in ('Restaurants','Food')  group by cat) x where x.rnk<16; |

4) Distribution of ratings vs  categories:

테이블 ratings를 앞서 만든 테이블 exploded 와 restaurant을 가지고 같은 business\_id로 조인하여 다음의 조건을 만족하여 만드세요:

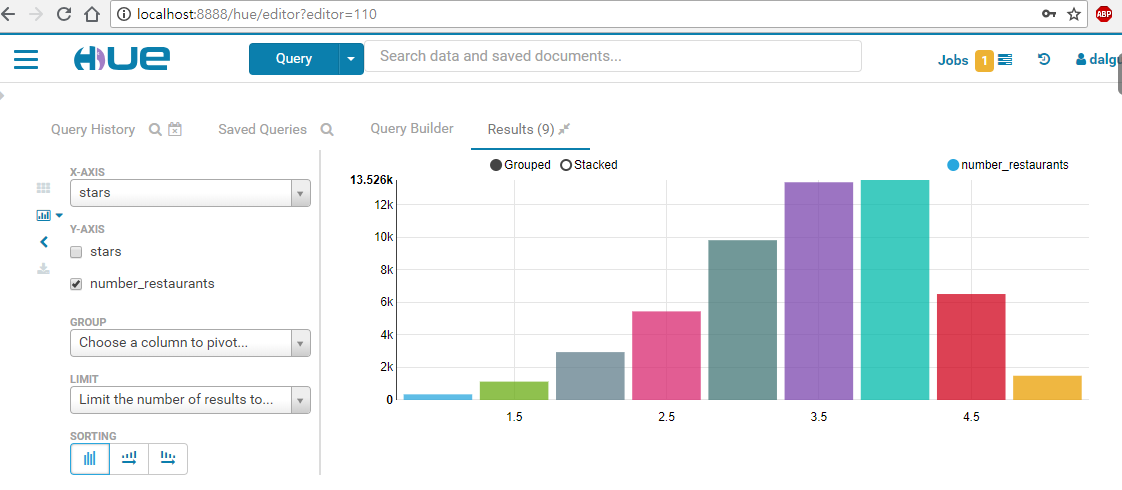
* exploded 의 cat\_exploded 값은 다음의 값중 하나:  "Nightlife","Bars", "Sandwiches", "Fast Food","American (Traditional)"
* exploded 의 cat\_exploded 와 stars 컬럼을 가지고 Group By  하고 stars 컬럼으로 ASC 하여 정렬하세여;

다음의 HiveQL로 Hue에서 그래프를 다음과 같이 만듬: select \* from ratings order by number DESC;



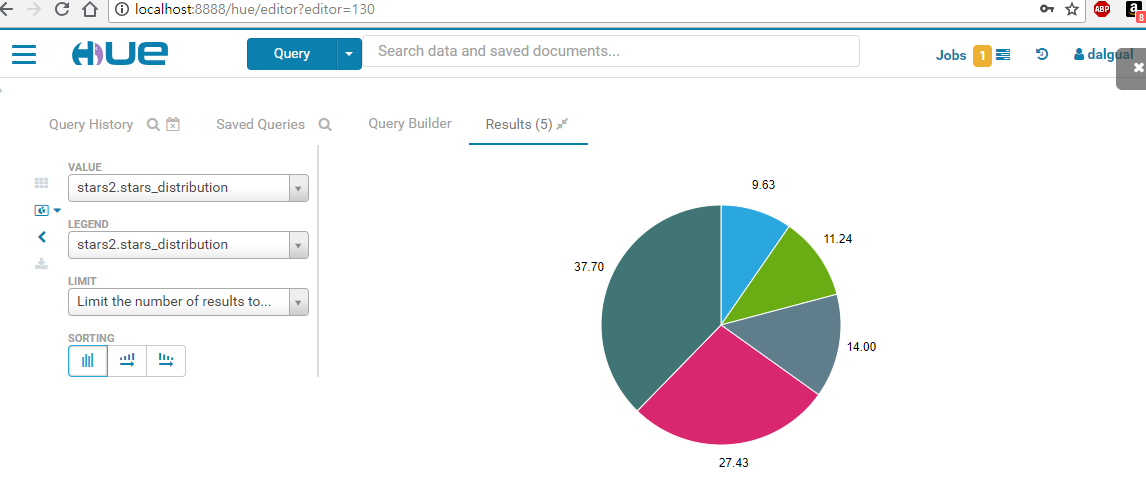
|  |
| --- |
| create table ratings AS  select a.cat, a.stars, count(\*) as cnt  from business\_exploded a  join (select business\_id from business\_exploded where cat = 'Restaurants') b on (a.business\_id = b.business\_id)  where cat in ("Nightlife","Bars", "Sandwiches", "Fast Food","American (Traditional)")  group by cat, stars  order by cat, stars asc;  select \* from ratings order by cnt desc limit 10; |

(5) What ratings do the majority of restaurants   have?



|  |
| --- |
| select stars, count(\*)  from restaurants  group by stars  order by stars asc; |

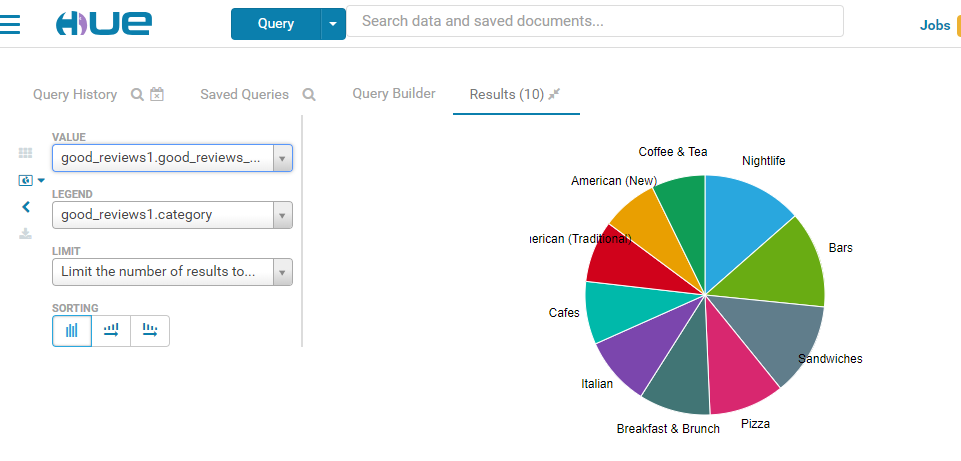
6) Rating distribution in restaurant  reviews



|  |
| --- |
| select a.stars, round( (count(a.stars)\*100.0/sum(count(a.stars)) over()), 2) as star\_distribution  from review\_p a  join restaurants b on (a.business\_id = b.business\_id)  group by a.stars; |

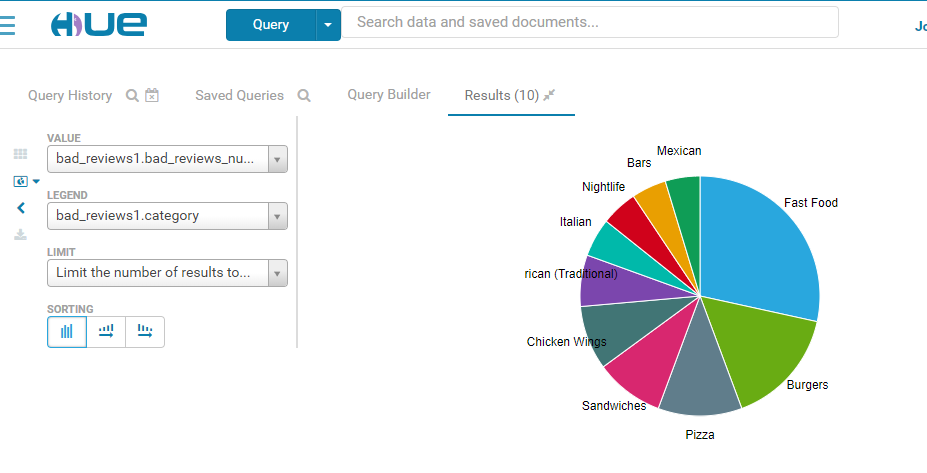
7) Which restaurants get bad vs good   reviews?

1. Good Review



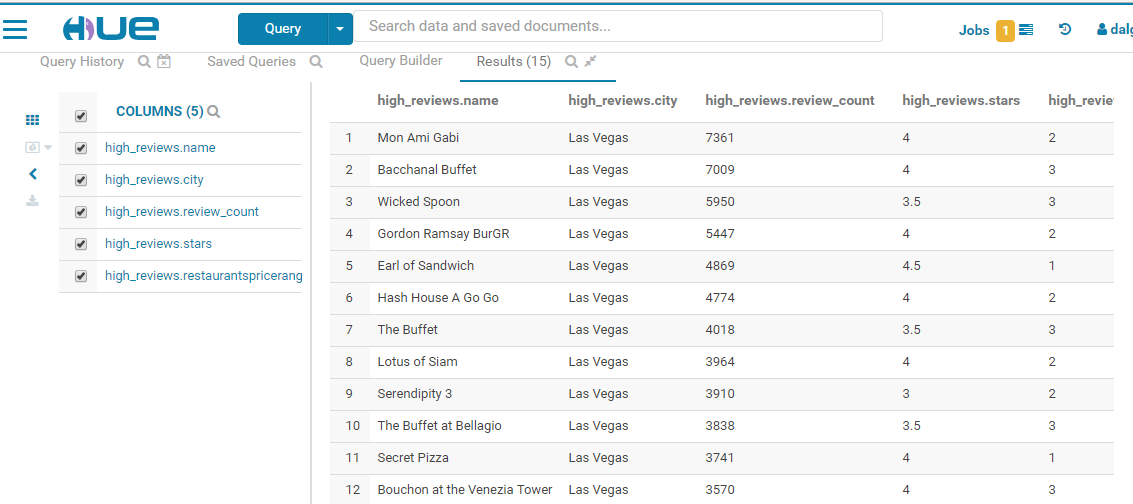
|  |
| --- |
| select a.cat, count(\*) as cnt from business\_exploded a join restaurants b on(a.business\_id=b.business\_id)  where a.cat NOT IN ("Restaurants","Food")  and b.stars>=4 group by a.cat order by cnt desc; |

b. Bad Review



|  |
| --- |
| select a.cat, count(\*) as cnt from business\_exploded a join restaurants b on(a.business\_id=b.business\_id)  where a.cat NOT IN ("Restaurants","Food")  and b.stars<=2 group by a.cat order by cnt desc; |

8) Which restaurants have the most  reviews?



|  |
| --- |
| select \* from restaurants  where review\_count >= 1000  order by review\_count desc limit 15; |

9) What number of yelp users are   elite?  
Do they rate differently than non -elite  users?

1. Average rating by all users:

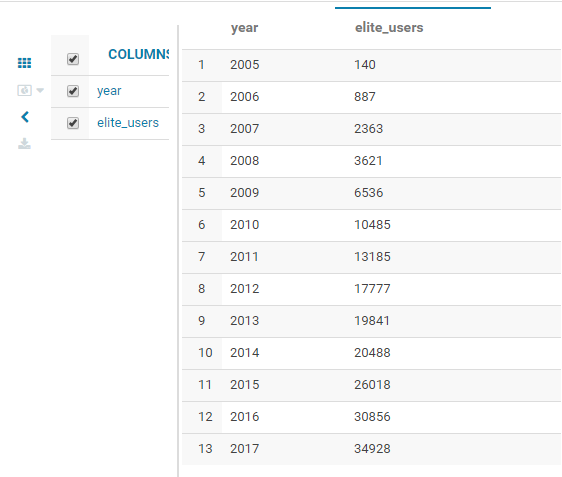
|  |
| --- |
|  |

1. Average rating by elite users:

|  |
| --- |
|  |

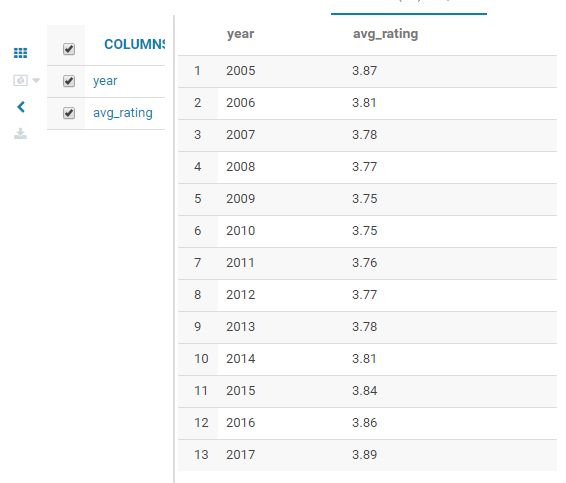
1. Count number of elite users by year:

|  |
| --- |
|  |



d. Count average reviews by elite users by year

|  |
| --- |
|  |



2) Which Cities Have The Highest Number Of   Restaurants?

|  |
| --- |
|  |