**Hive queries:**

***For business table creation:***

create table business\_tab(business\_id string, name string, address string, hours array<string>, is\_open int, categories array<string>, city string, state string, postal\_code string, latitude float, longitude float, stars int, review\_count int, attributes array<string>)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\n'

STORED AS TEXTFILE;

***For Loading the business table from hdfs directory – DemoDir1:***

LOAD DATA INPATH '/DemoDir1/businessdata.tsv' OVERWRITE INTO TABLE business\_tab;

***For Reviewing the data after table creation:***

select \* from business\_tab limit 10;

***For Review table creation:***

CREATE TABLE review\_tab(

review\_id string, user\_id string, business\_id string, stars int, date1 date, text string, useful int, funny int, cool int, type string

)

COMMENT 'row data csv'

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

"separatorChar" = "\,",

"quoteChar" = "\""

)

STORED AS TEXTFILE;

***For Loading the review table from hdfs directory – DemoDir1:***

LOAD DATA INPATH '/DemoDir1/review.csv' OVERWRITE INTO TABLE review\_tab;

***For Reviewing the data after table creation:***

select \* from review\_tab limit 10;

***Lateral view for the table with ‘Health and Medical’ category:***

SELECT business\_id, name, address, hours, is\_open, categories, city, state, postal\_code, latitude, longitude, stars, review\_count, attributes

FROM business\_tab LATERAL VIEW explode(categories) business\_cat AS cat where cat

RLIKE "Health & Medical" and is\_open = 1;

***Integration of business and review tables.***

Select b.business\_id, b.name, b.address, b.hours, b.is\_open, b.categories, b.city, b.state, b.postal\_code, b.review\_count, b.stars, b.latitude, b.longitude, b.attributes, r.business\_id, r.review\_id, r.user\_id, r.text, r,date1, r.stars as reviewStars from business\_tab b join review\_tab r ON (b.business\_id == r.business\_id) where r.text <> ‘ ’;

***Creating table cat\_explode1(category explode) in order split the categories across row:***

Create External table IF NOT EXISTS cat\_explode1(business\_id string, name string, city string, state string, address string, latitude float, longitude float, stars float, review\_count int, cat string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘\t’

collection items terminated by ‘,’

LINES TERMINATED BY ‘\n’

STORED AS TEXTFILE

tblproperties (“skip.header.line.count”=”1”);

***Maximum review\_count of health center is derived based on specific cities.***

Select DISTINCT review\_id, city, MAX(review\_count) as maxcount FROM review\_business1 LATERAL VIEW explode(categories) business\_cat AS cat group by city, review\_id limit 100;

***Best doctor in a specific city, by considering positive reviews.***

Select distinct rb1.business\_id, city, rb1.review\_count, rb1.name, rb1.address, rb1.categories, rb1.state, rb1.postal\_code, rb1.latitude, rb1.longitude, rb1.overall\_review, rb1.stars from review\_business1 as rb1 where city in (select rb2.city from review\_business1 as rb2 group by rb2 city) and rb1 overall\_review = ‘P’

***Best doctor in a specific city, by considering positive reviews.***

Select distinct rb1.business\_id, city, rb1.review\_count, rb1.name, rb1.address, rb1.categories, rb1.state, rb1.postal\_code, rb1.latitude, rb1.longitude, rb1.overall\_review, rb1.stars from review\_business1 as rb1 where city in (select rb2.city from review\_business1 as rb2 group by rb2 city) and rb1 overall\_review = ‘P’