Tasks : 12/09/2020

**Task 1: Perform queries for the following:**

// 1. Show the record where name="Amit"

db.Registration.find**({**Name**:**"Amit"**})**

// 2. Show all records where location="mumbai"

db.Profile.find**({**Location**:**"mumbai"**})**

// 3. Display people whose age is less than 25. Use $lt to filter

db.Profile.find**({**Age**:{**$lt**:**25**}})**

// 4. Display the document where fees < 20000 & course is JAVA

db.Registration.find**({**$and**:[{**Fees**:{**$lt**:**20000**}},{**Course**:**"JAVA"**}]})**

// 5. Display the document where fees between 20000 and 50000 and course is MEANSTACK

db.Registration.find**({**$and**:[{**Fees**:{**$lte**:**50000**}},{**Fees**:{**$gte**:**20000**}},{**Course**:**"MEANSTACK"**}]})**

**Task 2: Perform group by aggregation:**

// Grouping docs by Location

db.Profile.aggregate**({**$group**:{**\_id**:**"$Location"**,**count**:{**$sum**:**1**}}})**

**Task 3: Perform sort() and limit():**

// Sort in increasing order of fees

db.Registration.find**({})**.sort**({**Fees**:**1**})**

//Show top 4 docs of registration details

db.Registration.find**()**.limit**(**4**)**

**Lab Task : Perform following queries on given collection:**



// 1. Write a MongoDB query to display all the documents in the collection restaurants.

db.restaurants.find**()**

// 2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant

db.restaurants.find**({},{**restaurant\_id**:**1**,**name**:**1**,**borough**:**1**,**cuisine**:**1**})**

// 3. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine, but exclude the field \_id for all the documents in the collection restaurant

db.restaurants.find**({},{**\_id**:**0**,**restaurant\_id**:**1**,**name**:**1**,**borough**:**1**,**cuisine**:**1**})**

// 4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant

db.restaurants.find**({},{**\_id**:**0**,**restaurant\_id**:**1**,**name**:**1**,**borough**:**1**,**"address.zipcode"**:**1**})**

// 5. Write a MongoDB query to display all the restaurant which is in the borough Bronx.

db.restaurants.find**({**borough**:**"Bronx"**})**

// 6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx

db.restaurants.find**({**borough**:**"Bronx"**})**.limit**(**5**)**

// 7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx

db.restaurants.find**({**borough**:**"Bronx"**})**.skip**(**5**)**.limit**(**5**)**

// 8. Write a MongoDB query to find the restaurants who achieved a score more than 90

db.restaurants.find**({**grades**:{**score**:{**$gt**:**90**}}})**

// 9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100

db.restaurants.find**({**grades**:{**score**:{**$gt**:**80**,**$lt**:**100**}}})**

// 10. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168

db.restaurants.find**({**"address.coord" **:** **{**$lt **:** -72.754168**}});**

// 11. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

db.restaurants.find**({**$and**:[{**cuisine**:{**$ne**:**"American"**}},{**grades.score**:{**$gt**:**90**}},{**address.coord**:{**$lt**:**-65.744168**}}]})**

// 12. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168. Note : Do this query without using $and operator.

db.restaurants.find**({**"cuisine" **:** **{**$ne **:** "American "**},**"grades.score" **:{**$gt**:** 70**},**"address.coord" **:** **{**$lt **:** -65.754168**}});**

// 13. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.

db.restaurants.find**(** **{**"cuisine" **:** **{**$ne **:** "American "**},**"grades.grade" **:**"A"**,**"borough"**:** **{**$ne **:** "Brooklyn"**}})**.sort**({**"cuisine"**:**-1**});**

// 14. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

db.restaurants.find**({**name**:** /^Wil/**},{**"restaurant\_id" **:** 1**,**"name"**:**1**,**"borough"**:**1**,**"cuisine" **:**1**});**

// 15. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

db.restaurants.find**({**name**:** /ces$/**},{**"restaurant\_id" **:** 1**,**"name"**:**1**,**"borough"**:**1**,**"cuisine" **:**1**});**