|  |  |
| --- | --- |
|  |  |
|  | Create Order Management System using MongoDB and Implement Following Statements   1. Display the name of customers who have maximum orders. 2. Display the Mob No of customers who have highest Buying Total. 3. Display how many customers are there in customer collection. 4. Using collection of customer, and $exists, tell me how many customers belongs from pune city. 5. Find the customer who purchased shoes and cloth product. 6. Find the top 10 buyers. 7. Display all the orders where total amount is >1000. 8. Display All the customers with corresponding buying price. 9. write a MapReduce function which will return the Total Price per Customer. |
|  | Consider Customer database for collection  1. Use index on collection  2. Calculate amount per customer using aggregation /Map reduce operation using  MongoDB. |
|  | * 1. Create **restaurants Management System** using MongoDB   {  "address": {  "building": "1007",  "coord": [ -73.856077, 40.848447],  "street": "Tilak Road",  "zipcode": "411046"  },  "borough": "Bronx",  "cuisine": "Bakery",  "grades": [  { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },  { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },  { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },  { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },  { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }  ],  "name": "Morris Park Bake Shop",  "restaurant\_id": "30075445"  }  Implement Following Statements   1. Write a MongoDB query to display all the documents in the collection restaurants 2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant. 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. 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. 5. Write a MongoDB query to display all the restaurant which is in the borough Bronx. 6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx. 7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx. 8. Write a MongoDB query to find the restaurants who achieved a score more than 90. 9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100. 10. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double. 11. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.   12. write a MapReduce function which will return the Total rating score per restaurant. |
|  | Create Teacher Database in Mongodb Database. Which contain the information of  Teacher\_id, name of a teacher, department of a teacher, salary and status of a teacher? Here  status is whether teacher is approved by the university or not.   * Implement all the basic queries on the Teacher Database using mongodb commands. * Find out the teacher who are having salary greater than 25000 * Modify the E &TC department teacher salary with 10% growth * Display the teacher information according to status * Implement map reduce function to display average salary of teacher in each * department. |
|  | Design the DB for ”Pets” in Mongodb and perform following operations:  Pet(pet\_name, owner,sex, birth\_date, death\_date)  1)Insert, Update, Delete  2) Display all the pets information.  3) Display all pets in ascending order according to birth date.  4) Display all pets in descending order according birth date.  5) Display the pet information of specific owner.  6) Validate the data wherever require |
|  | Create a collection Tutorial( tutorial\_id,tut\_title,author,submission\_date)  Implement following operation on above collection  1)Insert 10 tutorial in collection  2)Update the date of specific tutorial  3)Delete tutorial from collection  4)Display all the tutorial details  5) Find specific tutorial\_id and author of respective tutorial.  6)Display all the records of tutorial where author name starts with “a”  7) Display all tutorials in Ascending and Descending order according to tutorial\_id or author name.  8) validate the data before entering into collection |
|  | Create a NOSQL DB on “Movie”(Movie\_id, name, type, budget ,date\_of\_release) using  MongoDB and implement following operations on document.   Insert (batch insert, insert validation) the single and multiple documents   Remove the documents   Update the type of movie   Upserts   Use aggregate function to retrieve the big budget, small budget movie  name.   Find the average budget of year “2017” |
|  | Create a NOSQL DB on “student”(roll\_no, name, marks, residential\_address,college\_address  ) using MongoDB .Execute at least 10 queries on above MongoDB database that  demonstrates following:   Find the student who has live in same city as college.   Retrieve the 4, 5 documents from collection.   Update the marks array for SE, TE, BE marks.   Use index, explain it and drop the index   Find the average of each year for every student. |
|  | Implement the aggregation and indexing with suitable example on above MongoDB database.   Aggregation   Create and drop different types of indexes and explain () to show the advantage of the indexes. |
|  | Consider following database:  Student (Roll\_no, Name, Address)  Subject (Sub\_code, Sub\_name)  Marks (Roll\_no, Sub\_code, marks)  Write following queries in MongoDB:  Consider wherever embedded collection is required   Find average marks of each student, along with the name of student.   Find how many students have failed in the subject “DBMS”.   Find the students who get marks greater than 75 and and also find student who get  less than 40.   Find the student whose addresses are ‘PUNE’. |