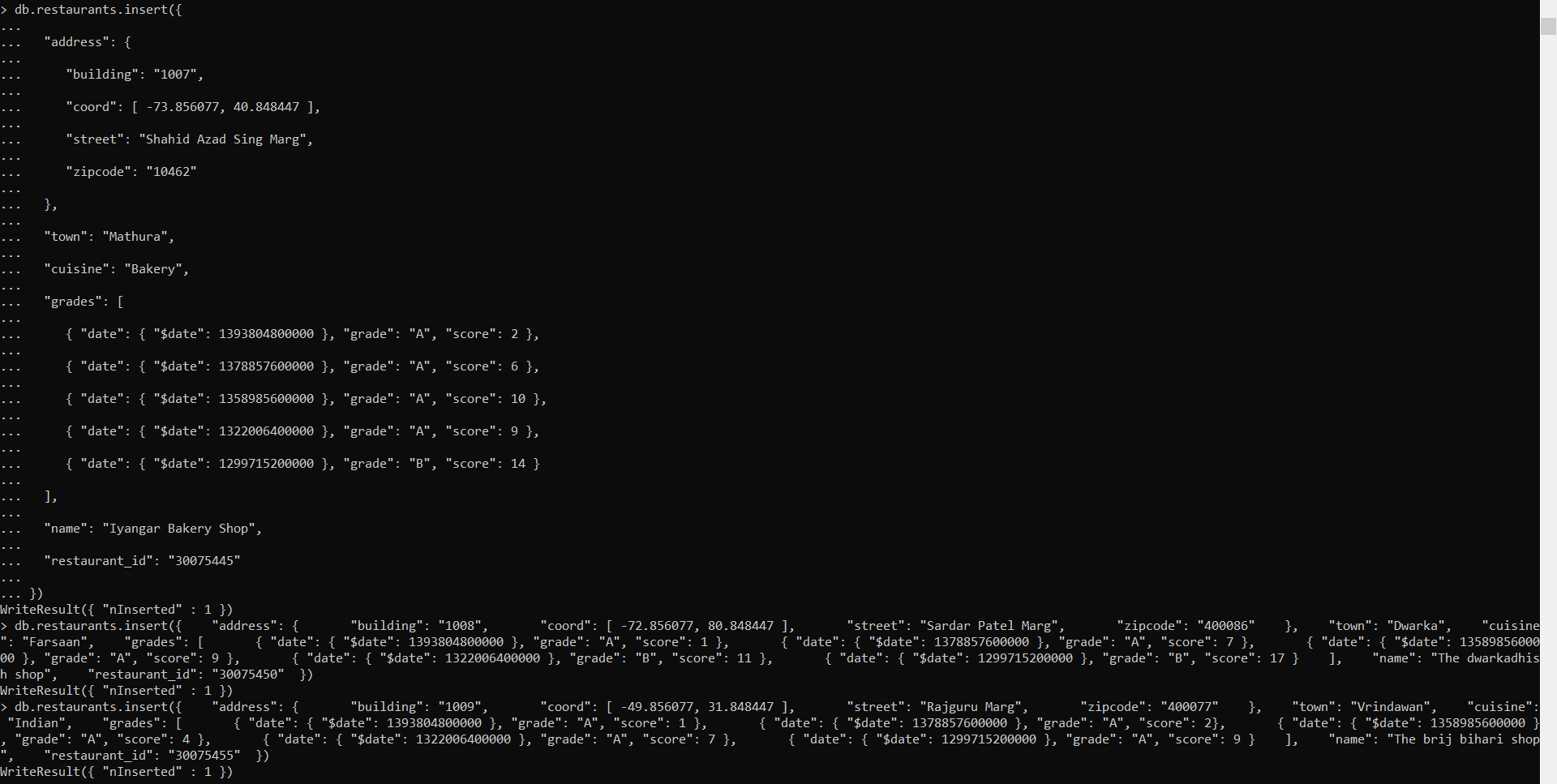
**Name: Nayan Mandliya  
Roll No: 1911027**

**Division: A**

**Mern Batch: 1**

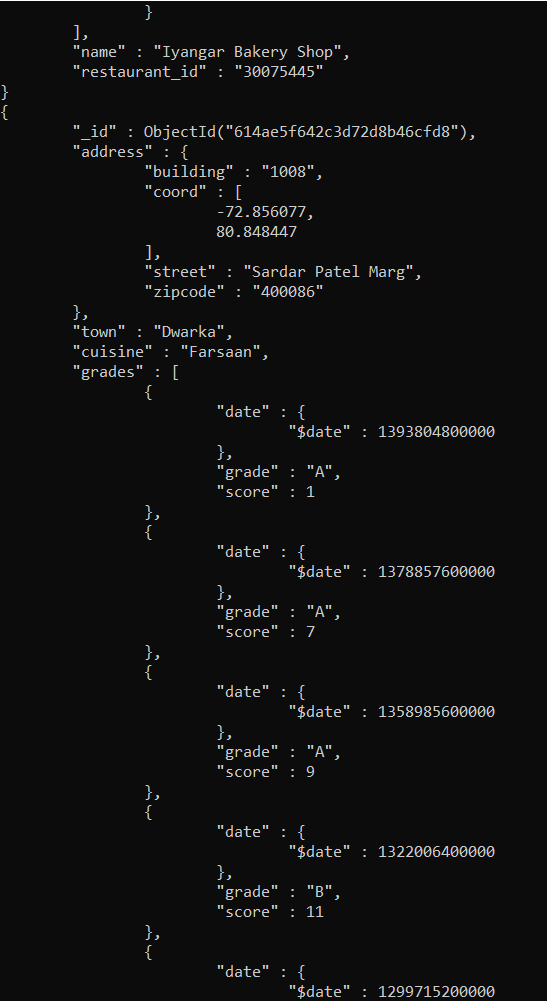
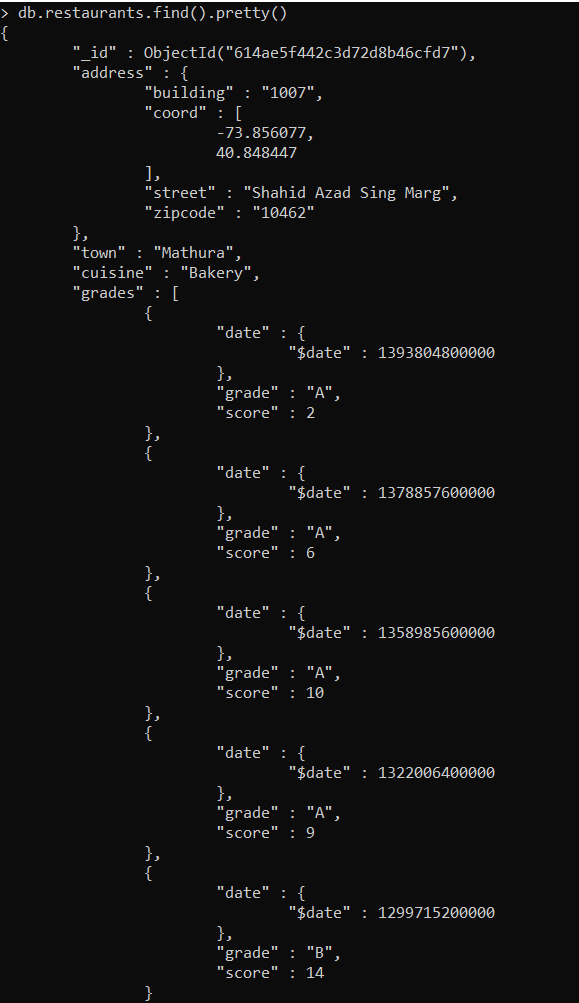
**Q 1) Write a MongoDB query to display all the documents in the collection restaurants.**

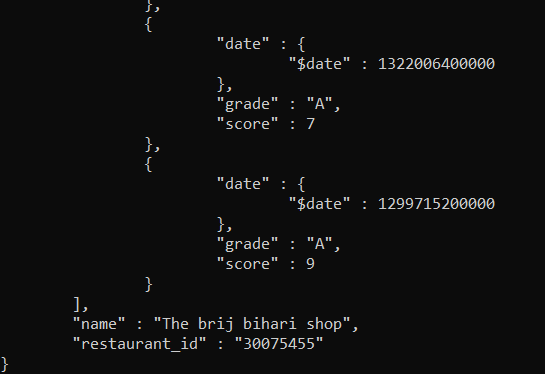
Collections inserted:

****

Query: db.restaurants.find().pretty()

Output:

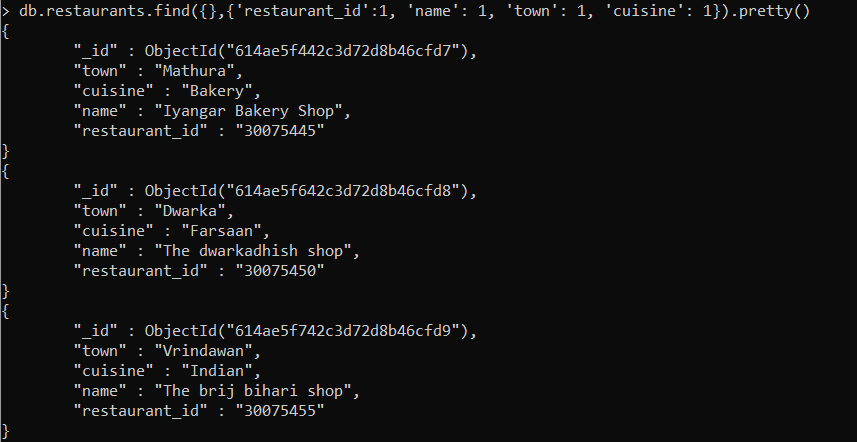




**Q 2) Write a MongoDB query to display the fields restaurant\_id, name, town and cuisine for all the documents in the collection restaurant.**

Query: db.restaurants.find({},{'restaurant\_id':1, 'name': 1, 'town': 1, 'cuisine': 1}).pretty()

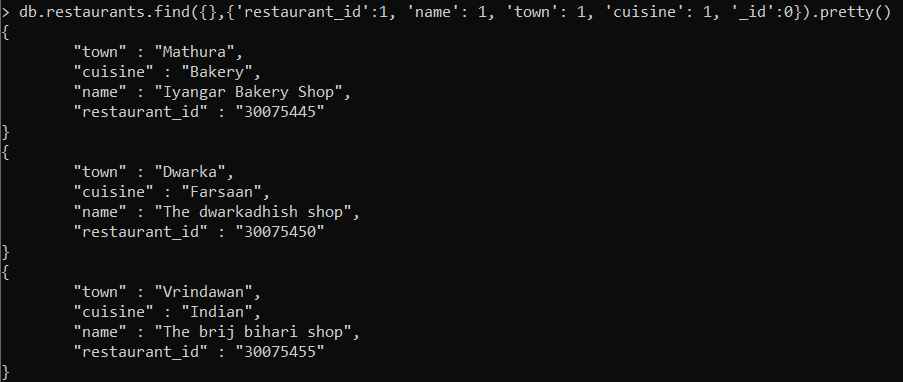
Output:



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

Query: Query: db.restaurants.find({},{'restaurant\_id':1, 'name': 1, 'town': 1, 'cuisine': 1, '\_id':0}).pretty()

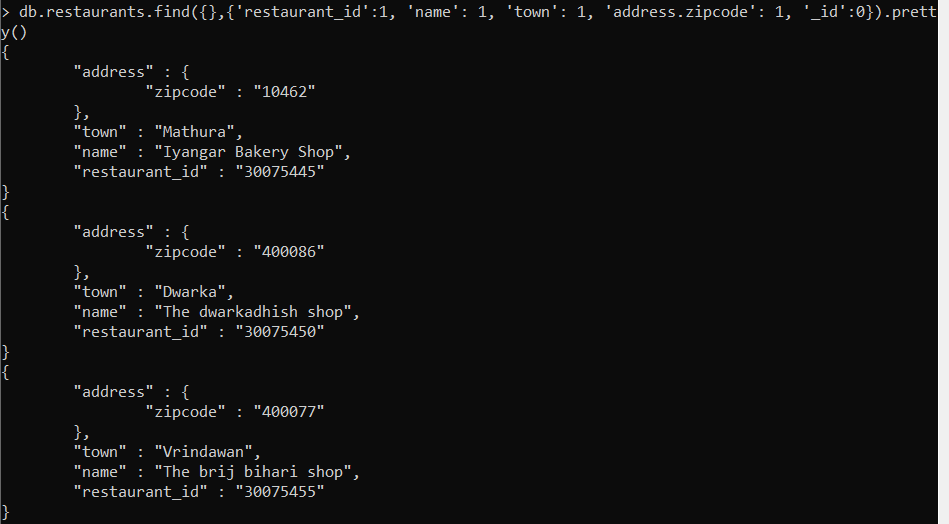
Output:



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

Query: db.restaurants.find({},{'restaurant\_id':1, 'name': 1, 'town': 1, 'address.zipcode': 1, '\_id':0}).pretty()

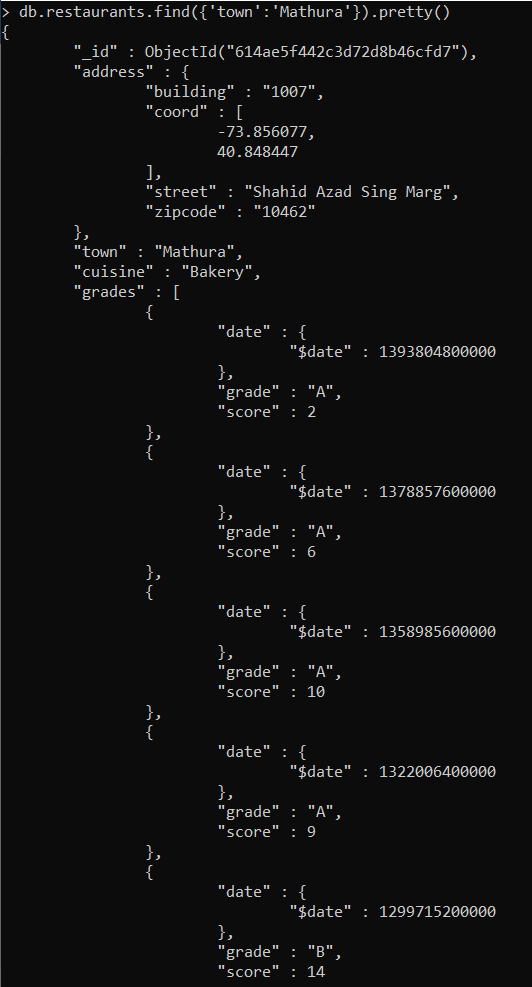
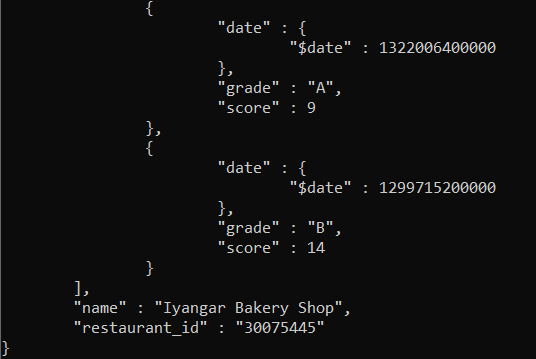
Output:



**Q 5) Write a MongoDB query to display all the restaurant which is in the town Mathura.**

Query: db.restaurants.find({'town':'Mathura'}).pretty()

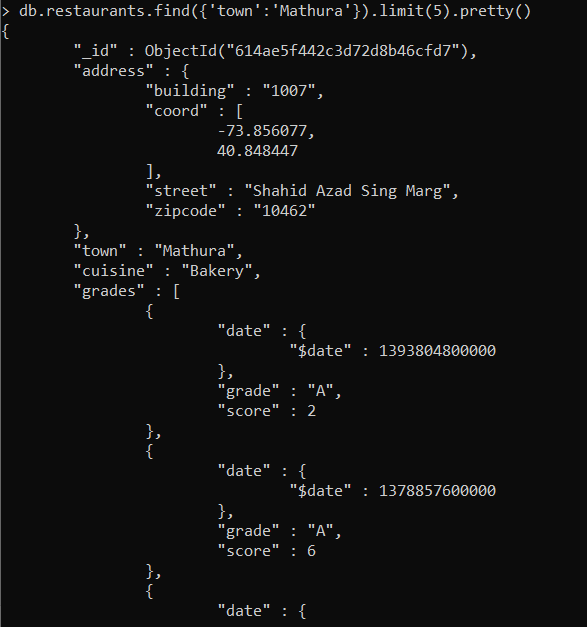
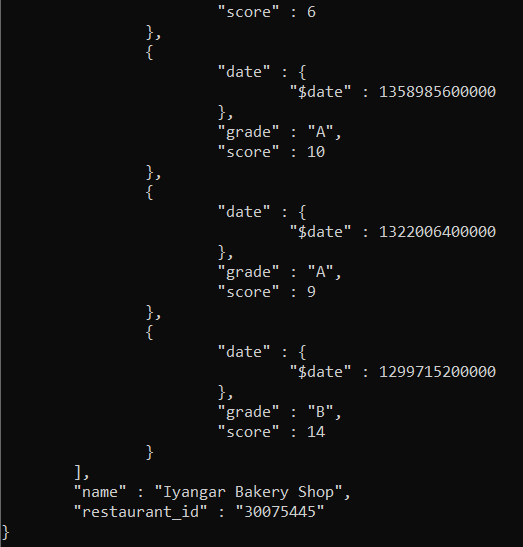
Output:

**Q 6) Write a MongoDB query to display the first 5 restaurant which is in the town Mathura.**

Query: db.restaurants.find({'town':'Mathura'}).limit(5).pretty()

Output:

**Q 7) Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the town Mathura.**

Query: db.restaurants.find({'town':'Mathura'}).skip(5).limit(5).pretty()

Output:

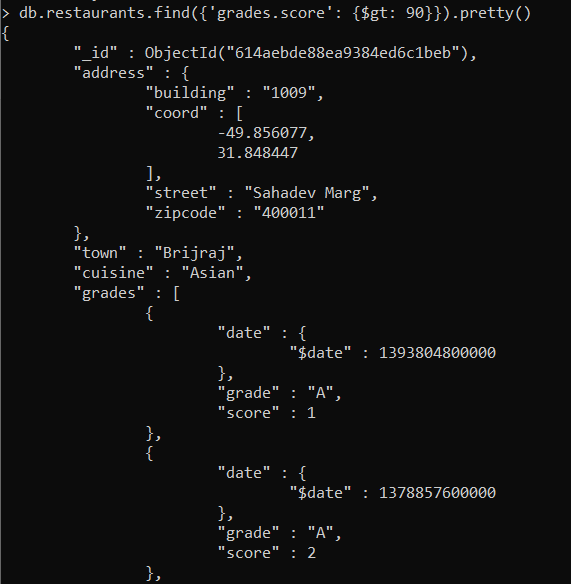
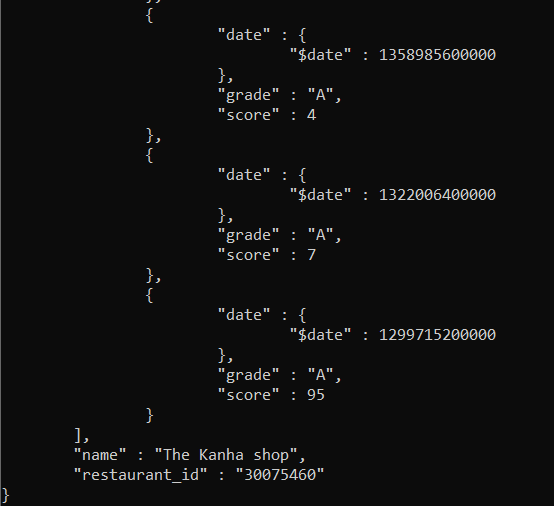


As there is only one restaurant which is in Mathura so nothing will be displayed.

**Q 8) Write a MongoDB query to find the restaurants who achieved a score more than 90. (Additional document is inserted for this)**

Query: db.restaurants.find({'grades.score': {$gt: 90}}).pretty()

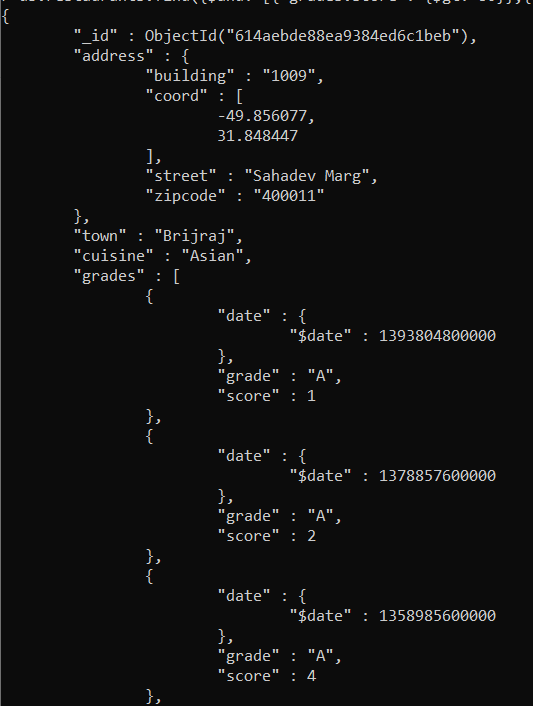
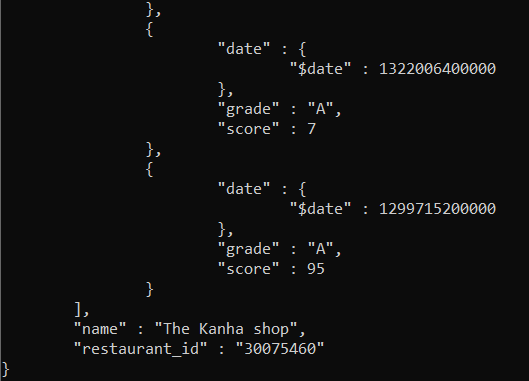
Output:

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

Query: db.restaurants.find({$and: [{'grades.score': {$gt: 80}},{'grades.score': {$lt: 100}}]}).pretty()

Output:

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

Query: db.restaurants.find({'address.coord': {$lt: -95.754168}}).pretty()

Output:

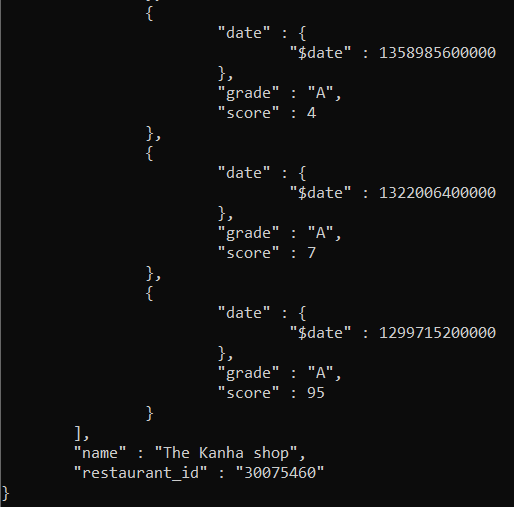
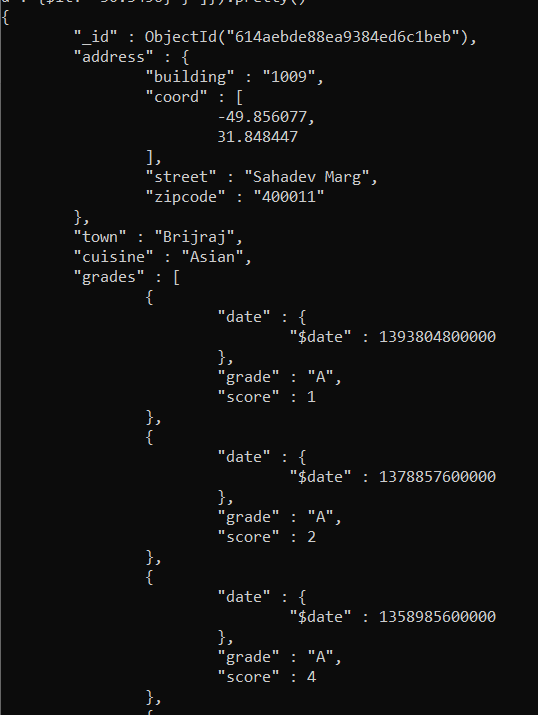


As there are no restaurants locate in latitude less than specified value.

**Q 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 -30.3456. ( latitude is changed to check whether output is correct or not)**

Query: db.restaurants.find({$and: [{'cuisine': {$ne: 'American'}},{'grades.score': {$gt: 70}}, {'address.coord': {$lt: -30.3456} } ]}).pretty()

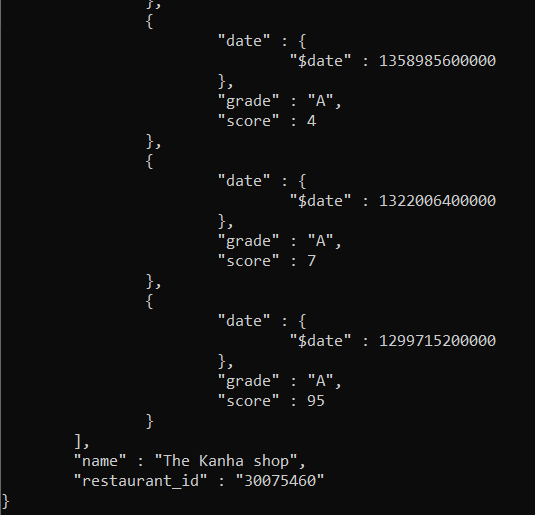
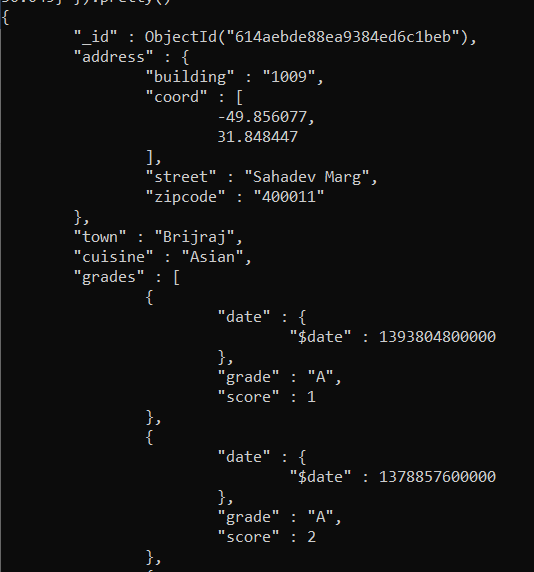
Output:



**Q 12) Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'Bengali' and achieved a score more than 70 and located in the longitude less than -30.045. Note : Do this query without using $and operator. (To verify output latitude is changed)**

Query: db.restaurants.find({ 'cuisine': {$ne: 'Bengali'}, 'grades.score': {$gt: 70}, 'address.coord': {$lt: -30.045} }).pretty()

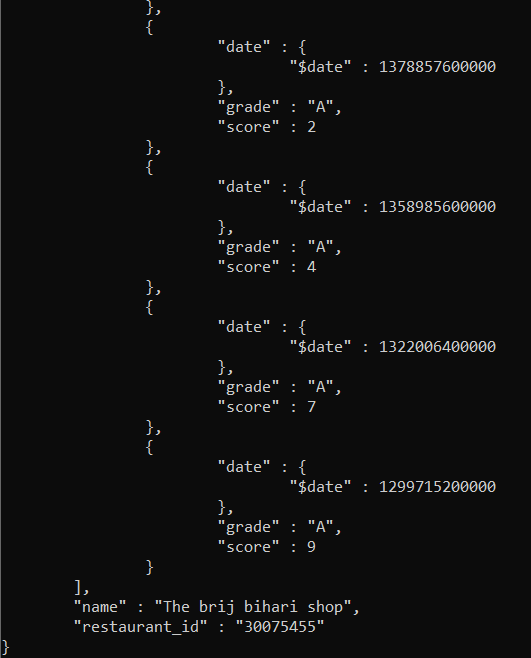
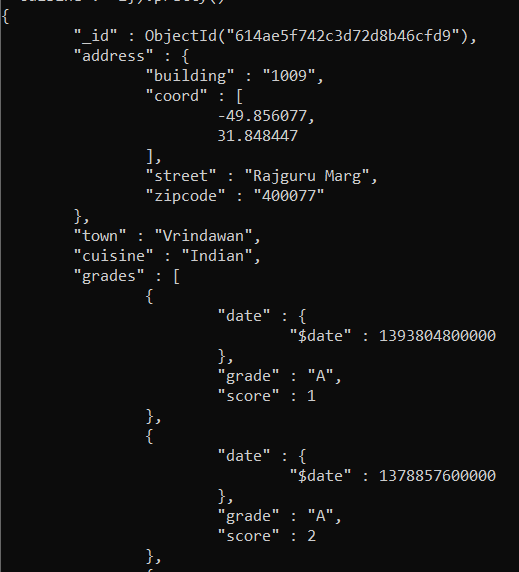
Output:

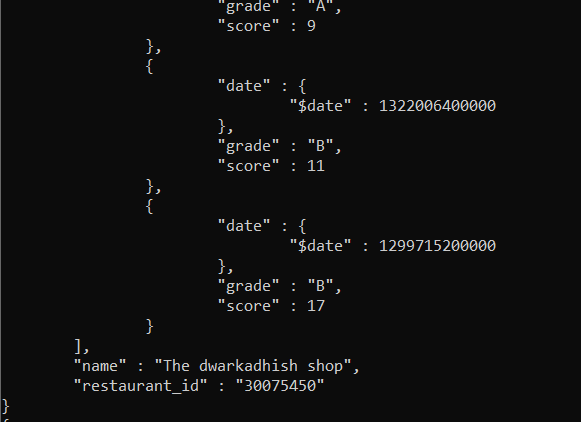
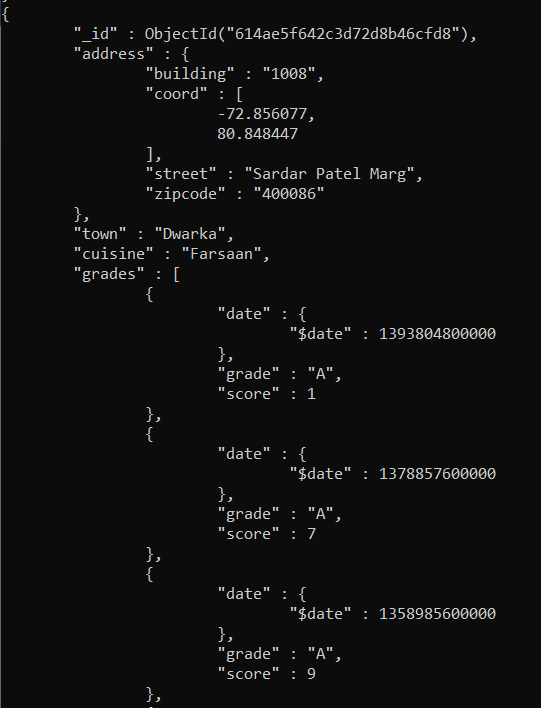


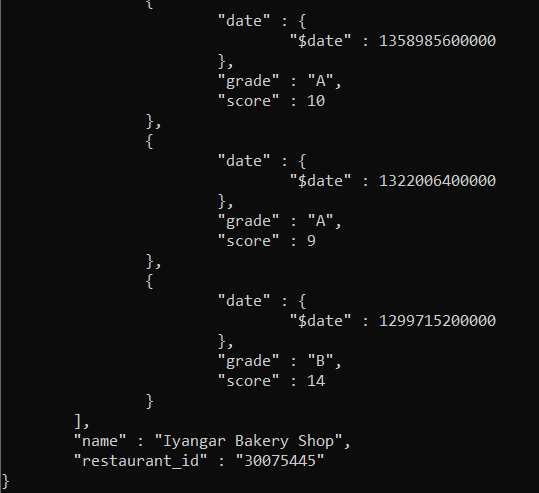
**Q 13) Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'South' and achieved a grade point 'A' not belongs to the town Delhi. The document must be displayed according to the cuisine in descending order.**

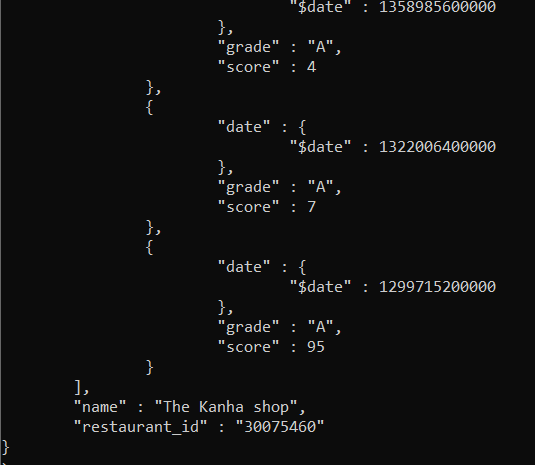
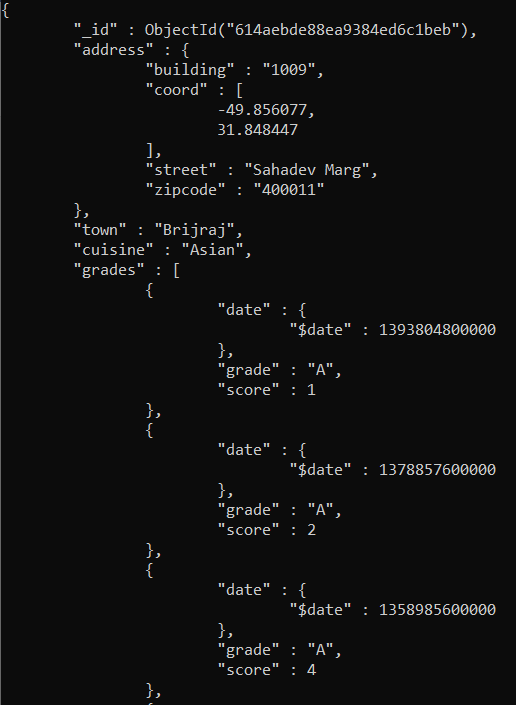
Query: db.restaurants.find({ 'cuisine': {$ne: 'South'}, 'grades.grade': 'A', 'town': {$ne: 'Delhi'} }).sort({ 'cuisine': -1}).pretty()

Output:









**Q 14) Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which contain 'The' as first three letters for its name.( For checking output ‘Vad’ is replaced by ‘The’)**

Query: db.restaurants.find({'name': /^The/},{'restaurant\_id':1,'name': 1, 'town': 1, 'cuisine': 1, '\_id':0}).pretty()

Output:



**Q 15) Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which contain 'shop' as last three letters for its name.( for verification purpose ‘shop’ is considered)**

Query: db.restaurants.find({'name': /shop$/},{'restaurant\_id':1,'name': 1, 'town': 1, 'cuisine': 1, '\_id':0}).pretty()

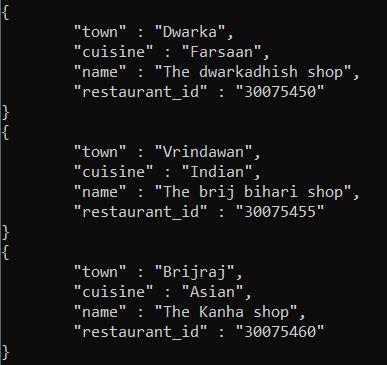
Output:



**Q 16) Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which contain 'he ' as three letters somewhere in its name.**

Query: db.restaurants.find({'name': /.\*he .\*/},{'restaurant\_id':1,'name': 1, 'town': 1, 'cuisine': 1, '\_id':0}).pretty()

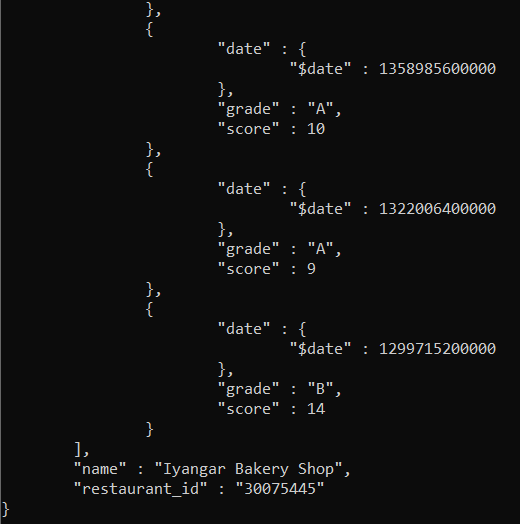
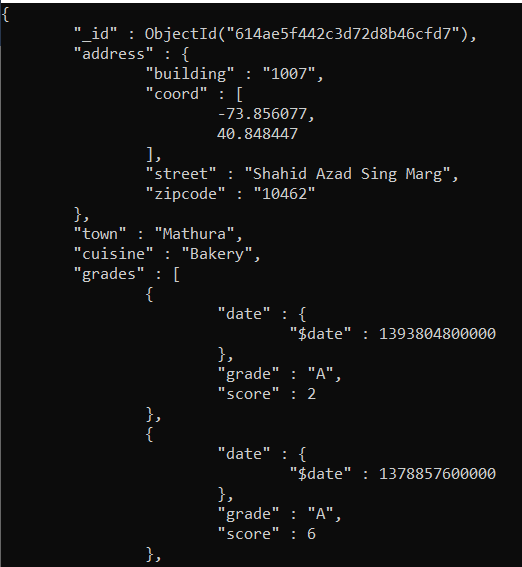
Output:



**Q 17) Write a MongoDB query to find the restaurants which belong to the town Mathura and prepared either south or Bakery dish.**

Query: db.restaurants.find({'town': 'Mathura', $or: [{'cuisine': 'south'},{'cuisine': 'Bakery'}]},).pretty()

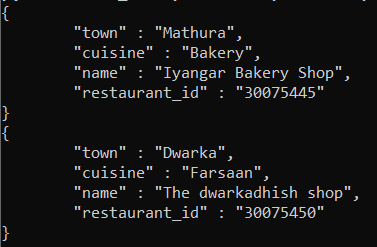
Output:



**Q 18) Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which belong to the town Pune or Nagpur or Mathura or Dwarka.**

Query: db.restaurants.find({$or: [{'town': 'Pune'},{'town': 'Nagpur'},{'town': 'Mathura'},{'town':'Dwarka'}]},{'restaurant\_id':1, 'name': 1, 'town':1, 'cuisine':1, '\_id':0}).pretty()

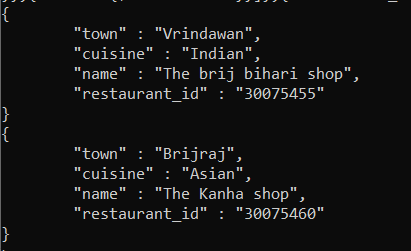
Output:



**Q 19) Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which are not belonging to the town Pune or Nagpur or Mathura or Dwarka.**

Query: db.restaurants.find({$and: [{'town': {$ne: 'Pune'}},{'town': {$ne: 'Nagpur'}},{'town': {$ne: 'Mathura'}},{'town': {$ne: 'Dwarka'}}]},{'restaurant\_id':1, 'name': 1, 'town':1, 'cuisine':1, '\_id':0}).pretty()

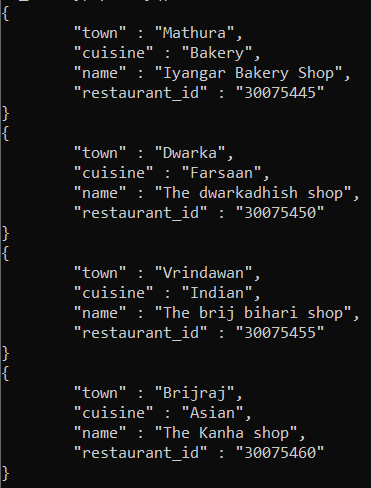
Output:



**Q 20) Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which achieved a score which is not more than 10.**

Query: db.restaurants.find({'grades.score': {$lte: 10}},{'restaurant\_id':1, 'name': 1, 'town':1, 'cuisine':1, '\_id':0}).pretty()

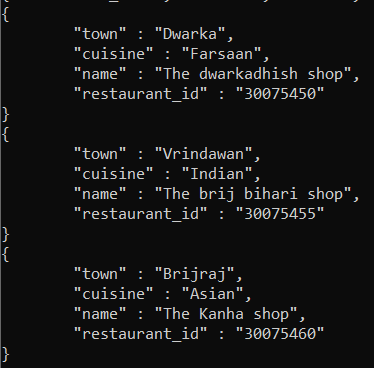
Output:



**Q 21) Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which prepared dish except 'Bengali' and 'Chinees' or restaurant's name begins with letter 'kri'.**

Query: db.restaurants.find({$or: [{$and: [{'cuisine': 'Bengali'},{'cuisine': 'Chinees'}]},{'name': /^The/}]},{'restaurant\_id':1, 'name': 1, 'town':1, 'cuisine': 1, '\_id': 0}).pretty()

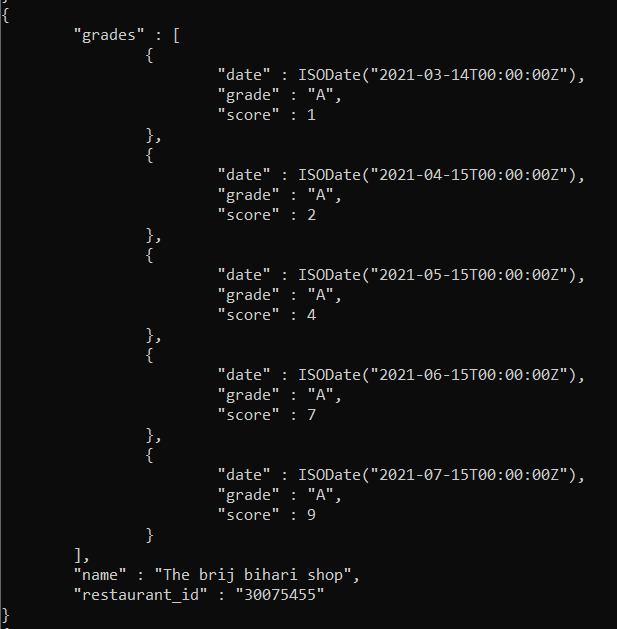
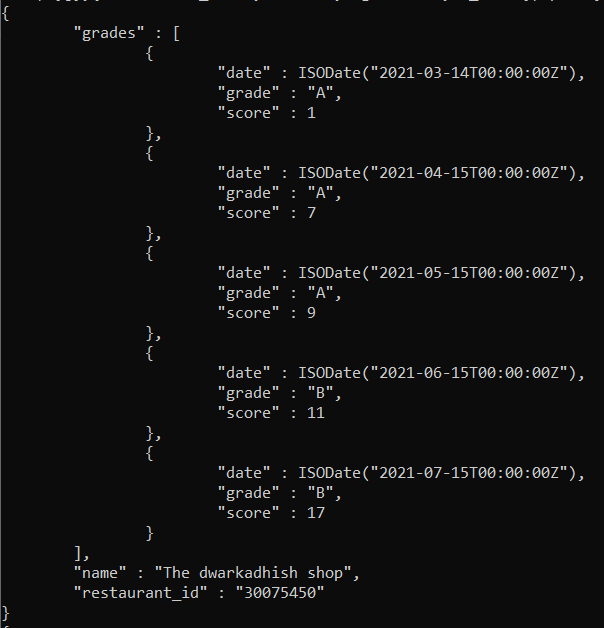
Output:

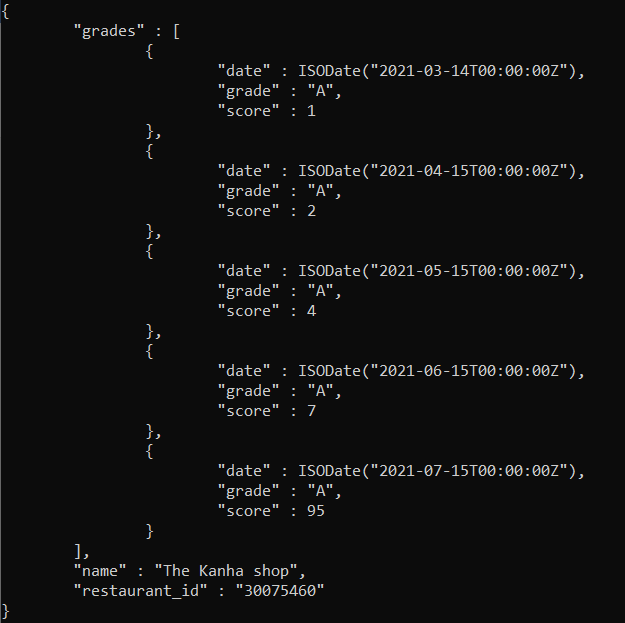


**Q 22) Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 1 on an ISODate "2021-03-14" among many of survey dates.**

Query: db.restaurants.find({$and: [{'grades.grade':'A'}, {'grades.score':1}, {'grades.date': ISODate('2021-03-14') }]},{'restaurant\_id':1,'name':1, 'grades':1, '\_id':0}).pretty()

Output:

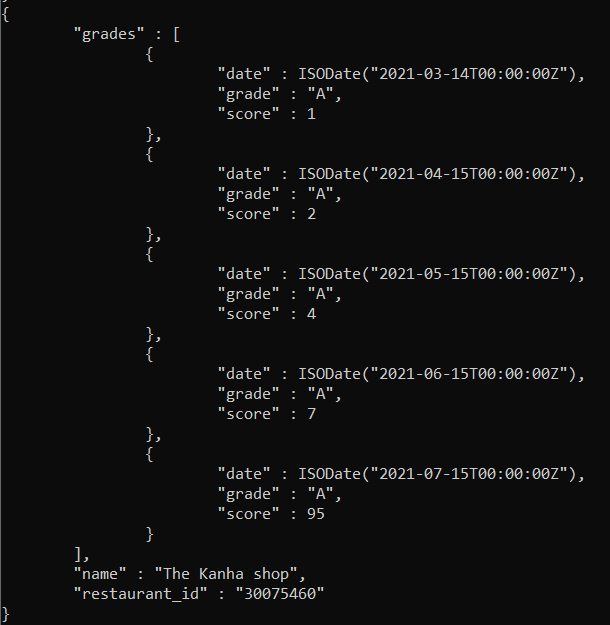
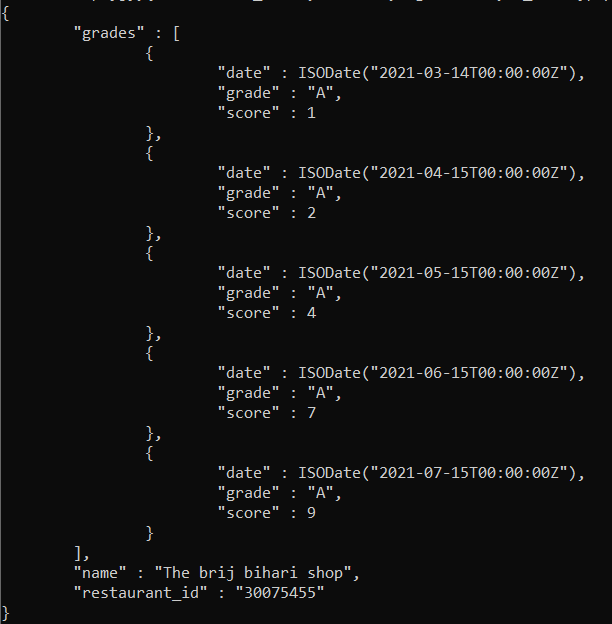




**Q 23) Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2021-04-15".**

Query: db.restaurants.find({$and: [{'grades.1.grade':'A'}, {'grades.1.score':2}, {'grades.1.date': ISODate('2021-04-15') }]},{'restaurant\_id':1,'name':1, 'grades':1, '\_id':0}).pretty()

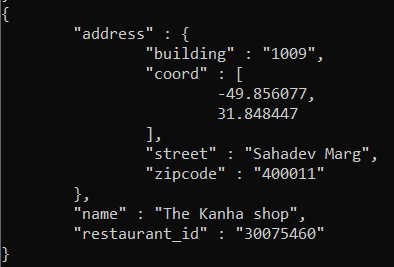
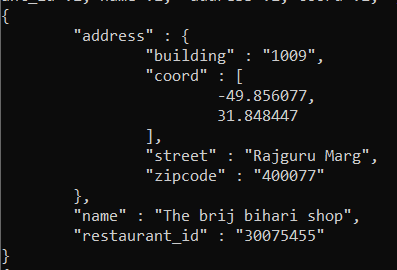
Output:



**Q 24) Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 30 and upto 40.**

Query: db.restaurants.find({$and: [{'address.coord.1': {$gt: 30}}, {'address.coord.1': {$lt: 40}}]},{'restaurant\_id':1,'name':1, 'address':1,'coord':1, '\_id':0}).pretty()

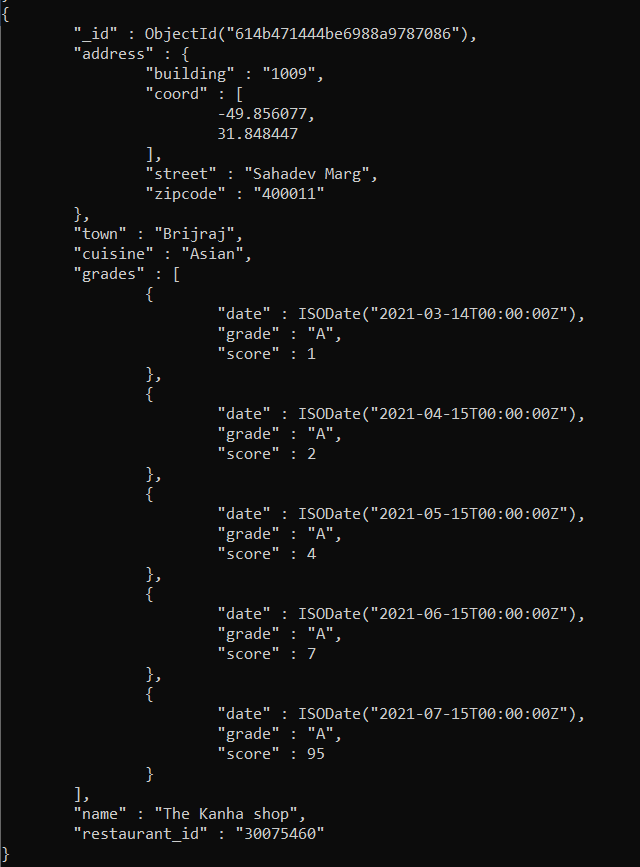
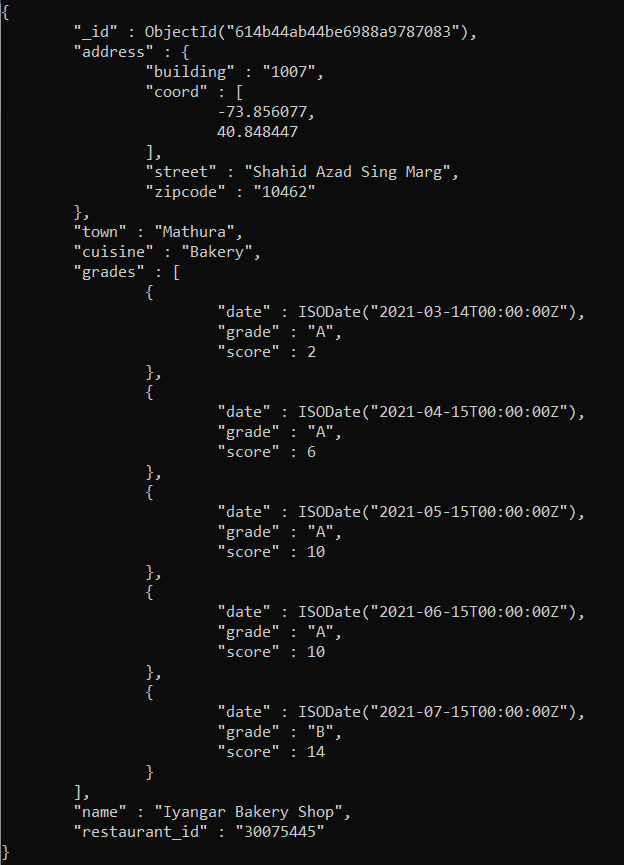
Output:

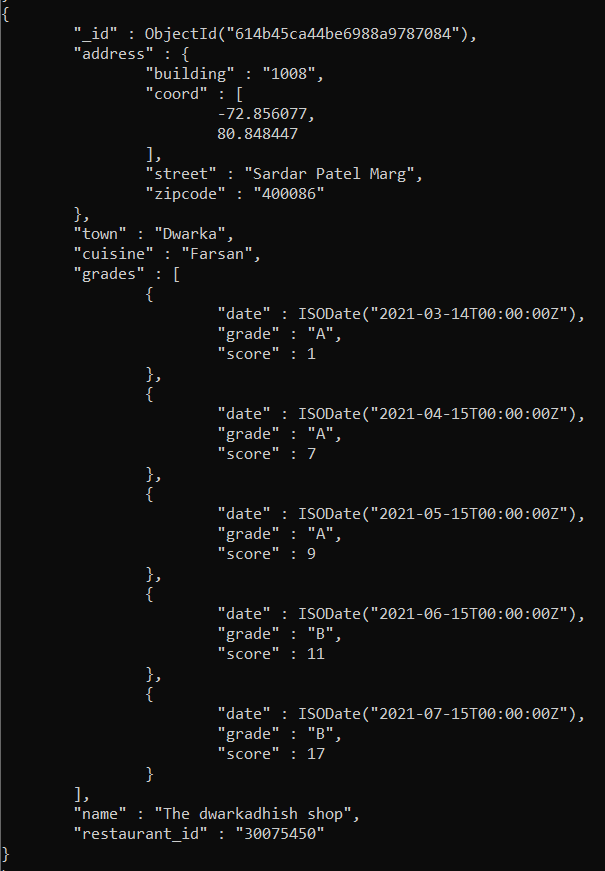
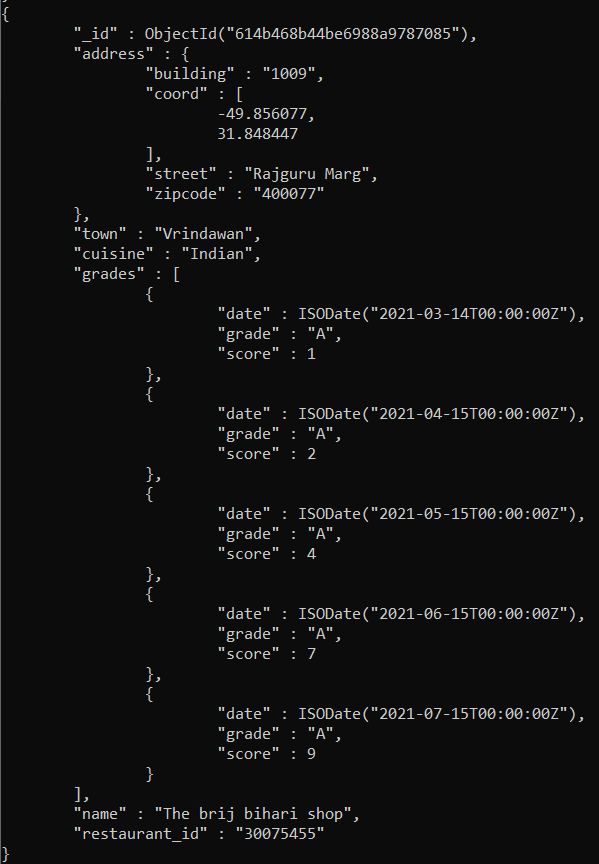


**Q 25) Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.**

Query: db.restaurants.find().sort({'name':1}).pretty()

Output:

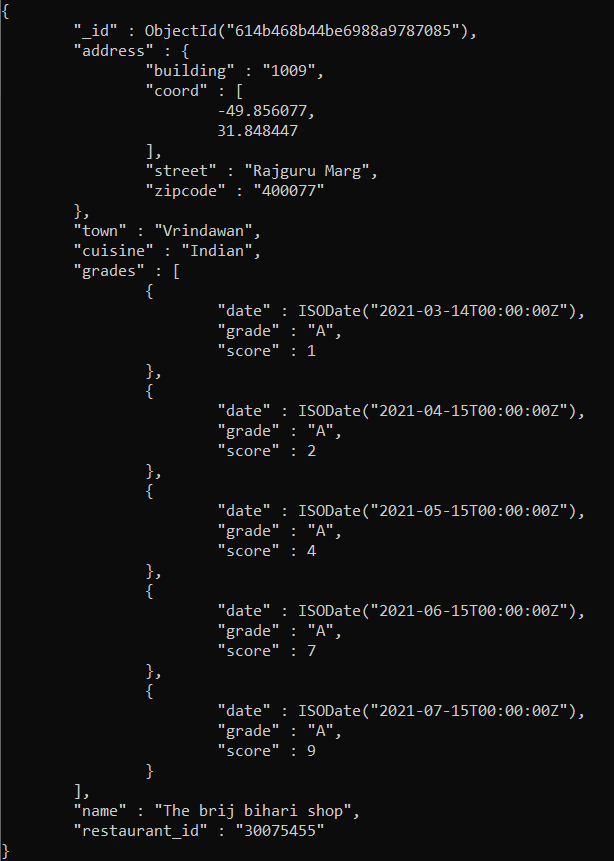
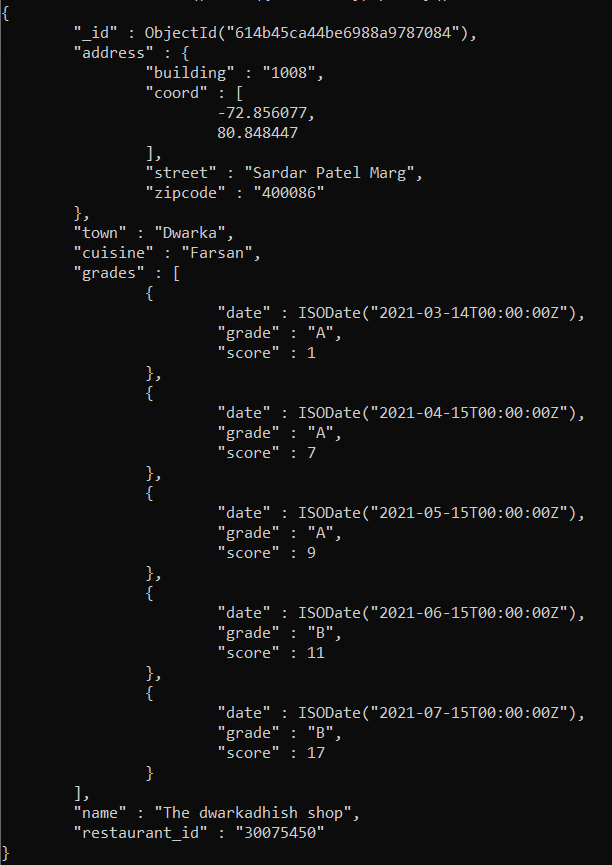


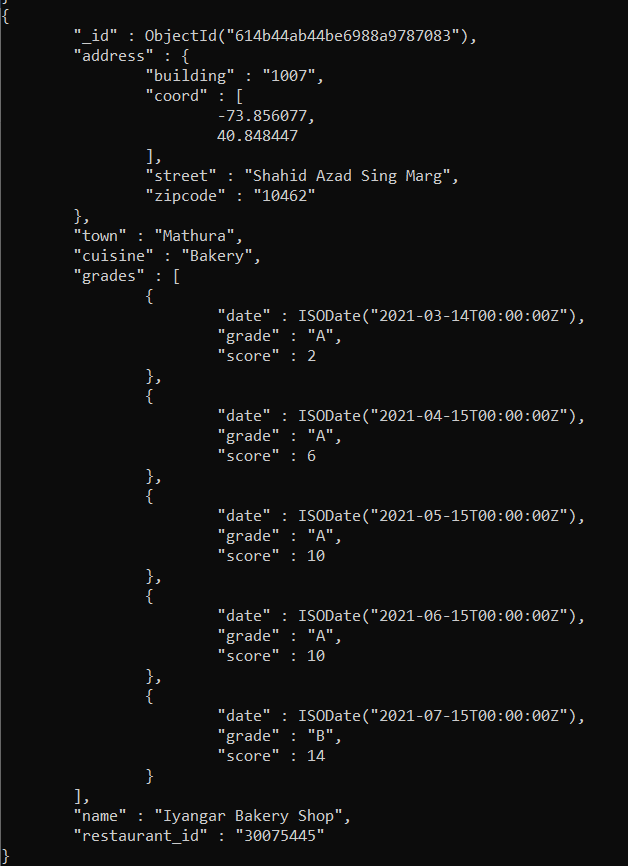
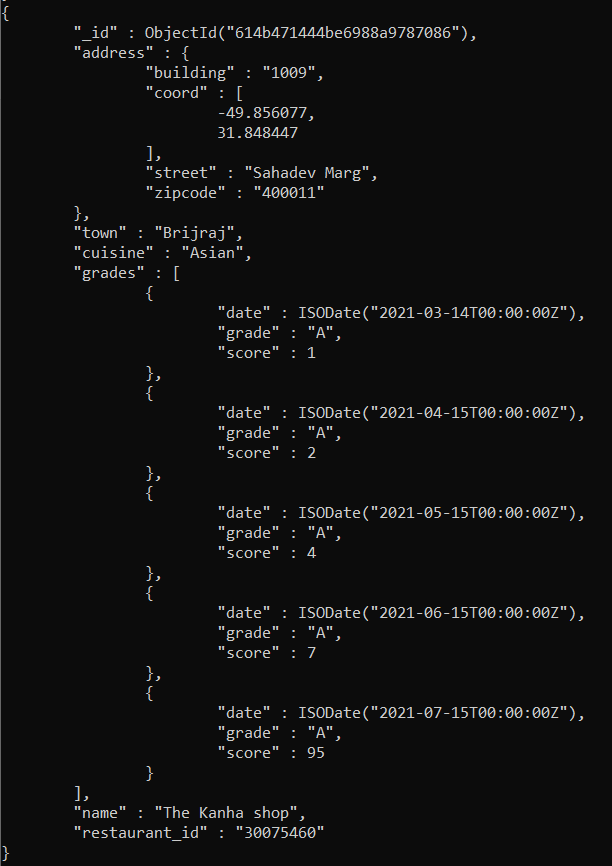


**Q 26) Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.**

Query: db.restaurants.find().sort({'name':-1}).pretty()

Output:

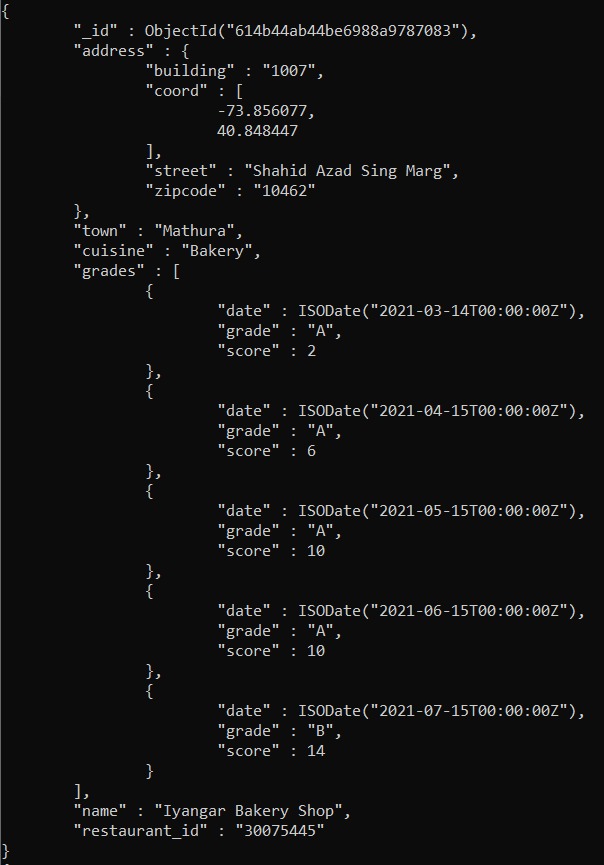
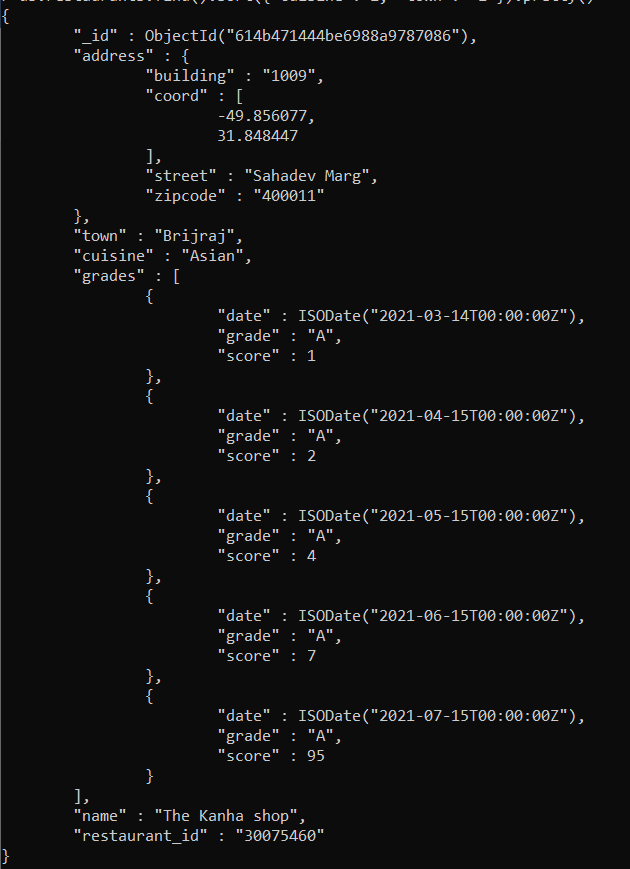


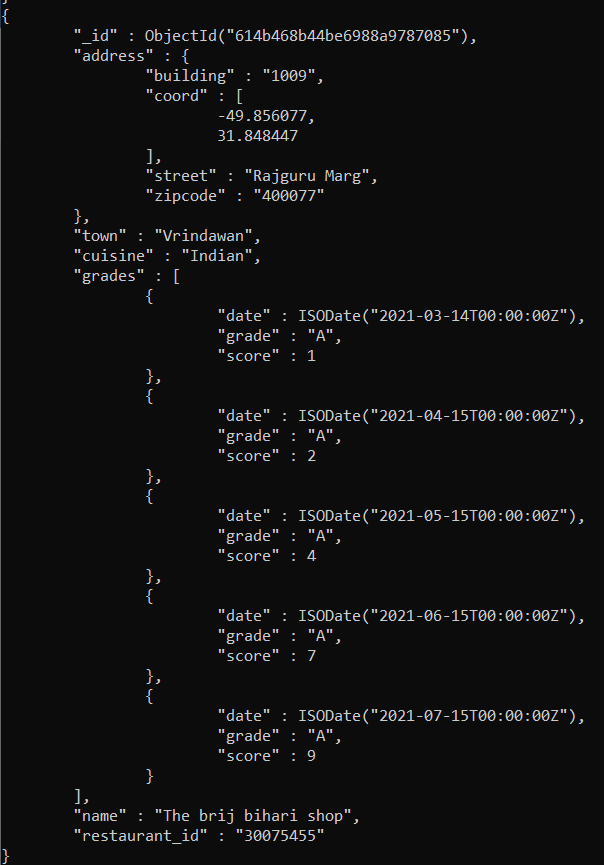
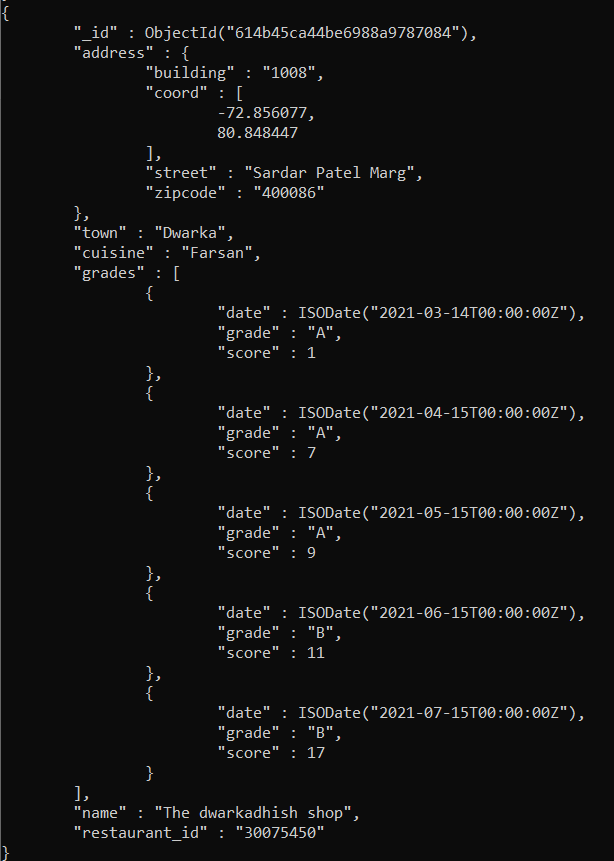


**Q 27) Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine town should be in descending order.**

Query: db.restaurants.find().sort({'cuisine': 1, 'town': -1 }).pretty()

Output:

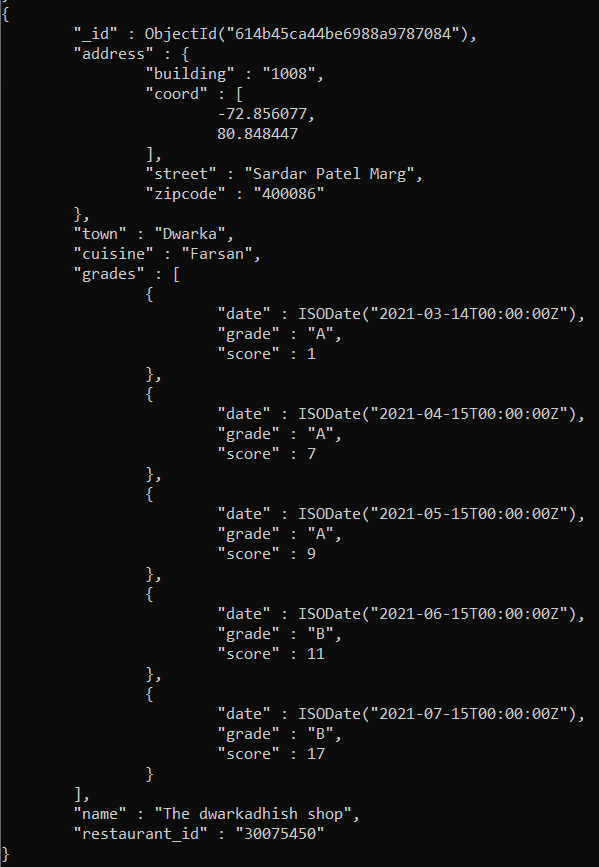
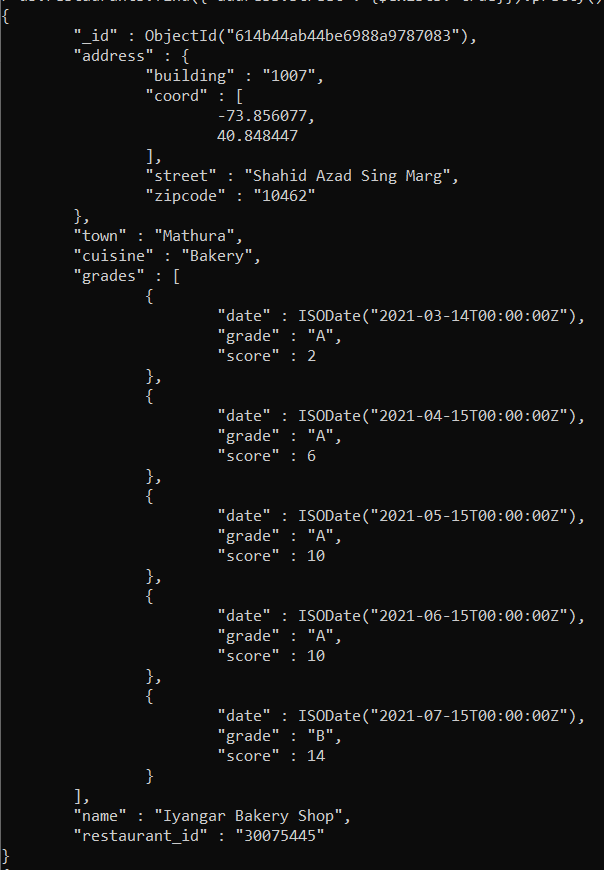


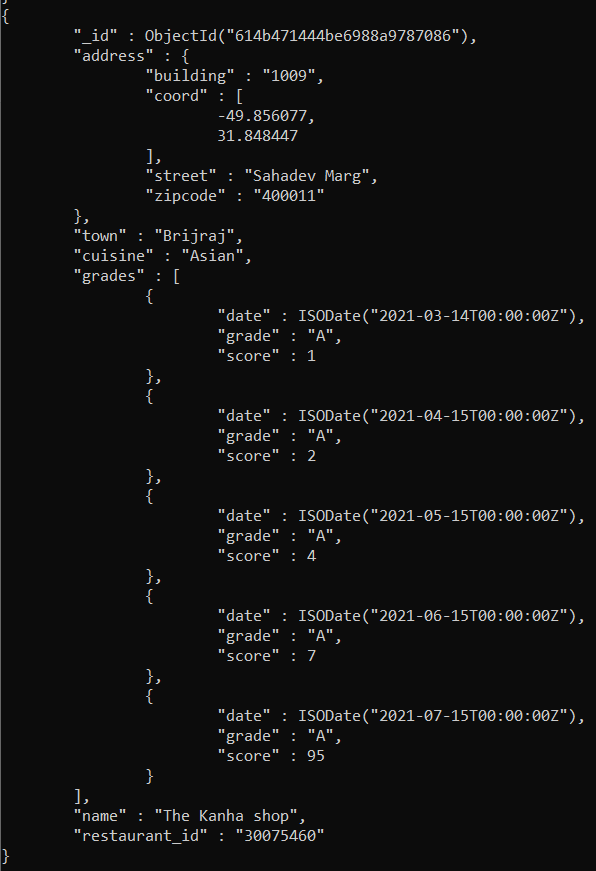
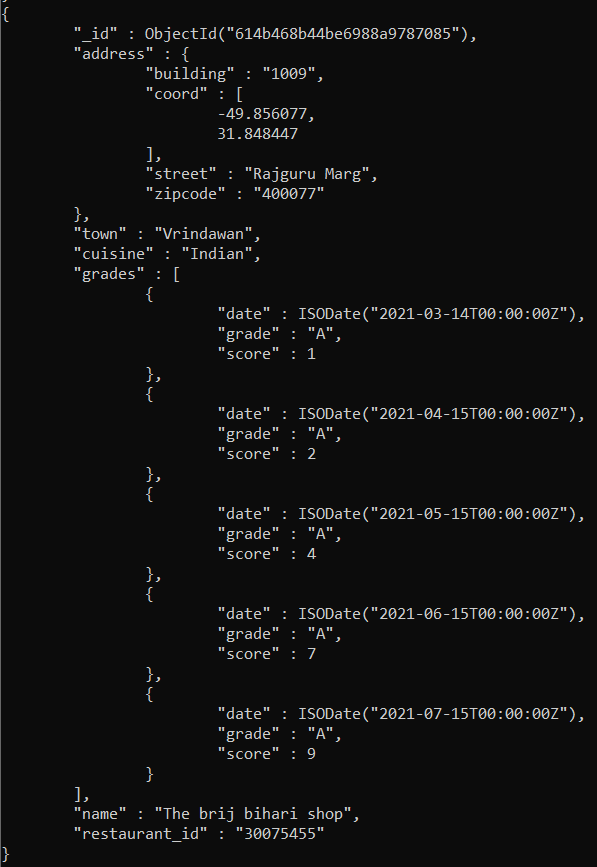


**Q 28) Write a MongoDB query to know whether all the addresses contains the street or not.**

Query: db.restaurants.find({'address.street': {$exists: true}}).pretty()

Output:

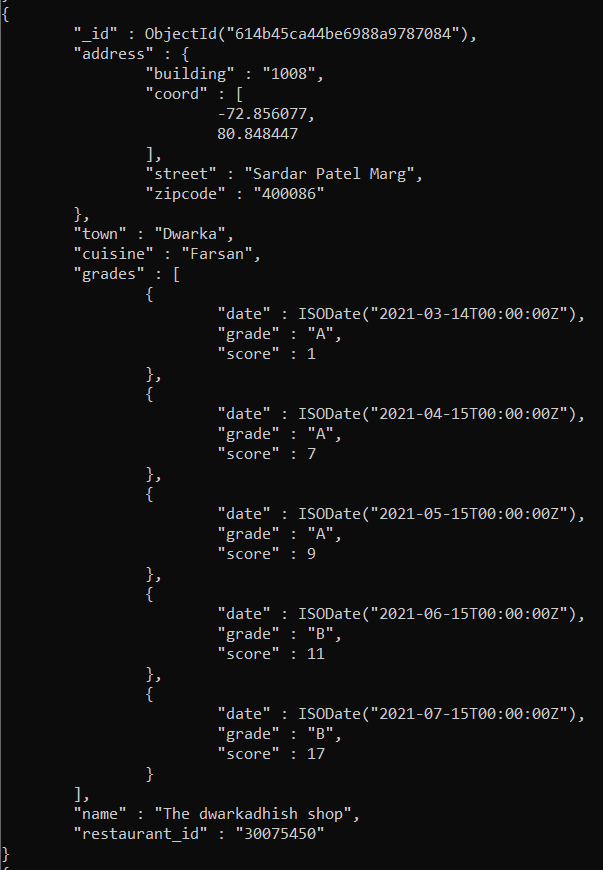
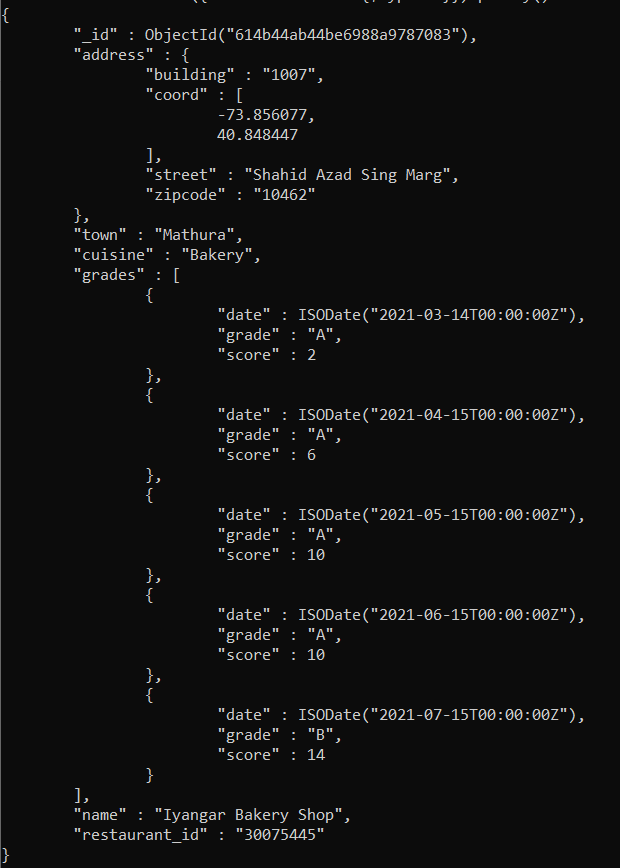


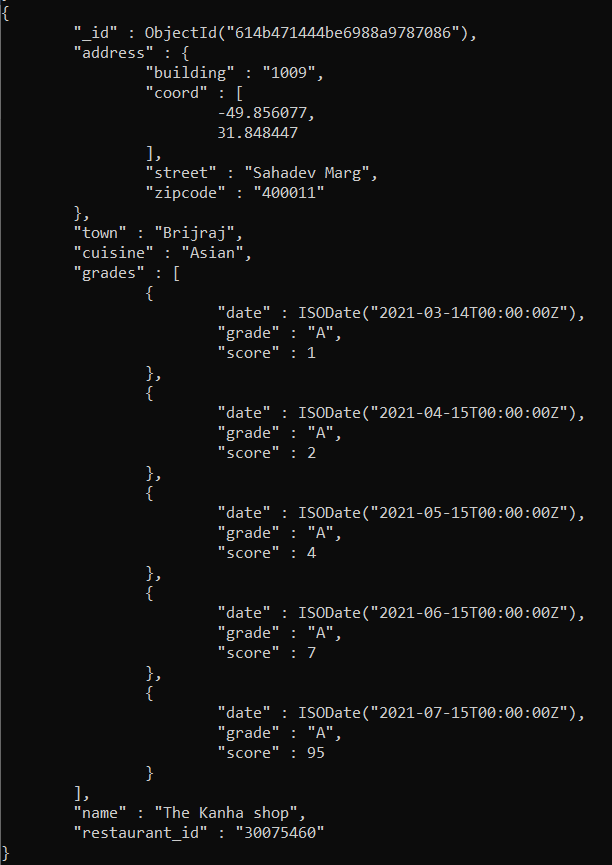
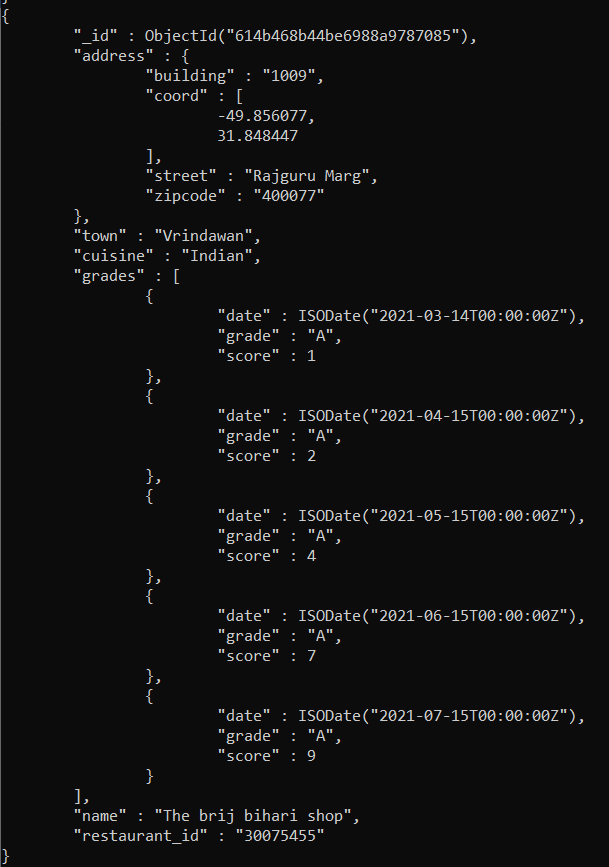


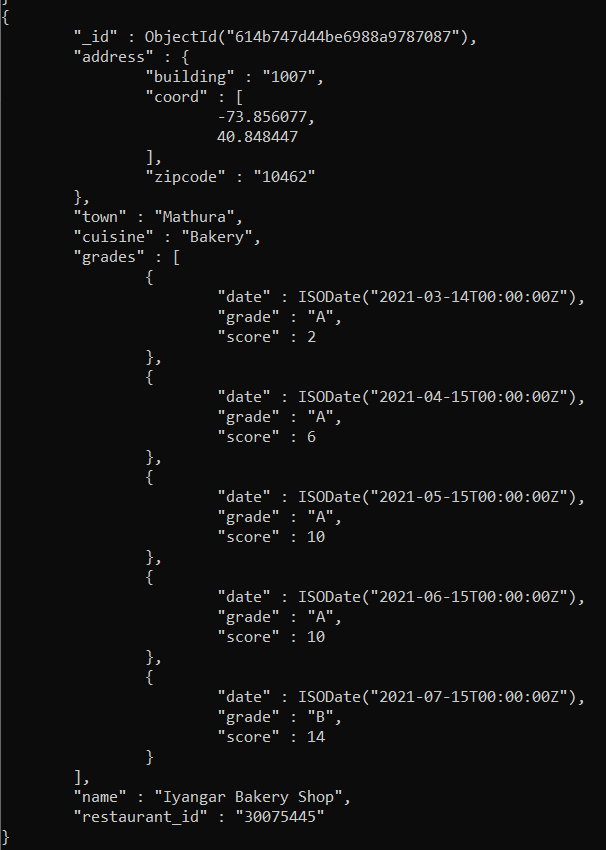
**Q 29) Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.**

Query: db.restaurants.find({'address.coord': {$type: 1}}).pretty()

Output:



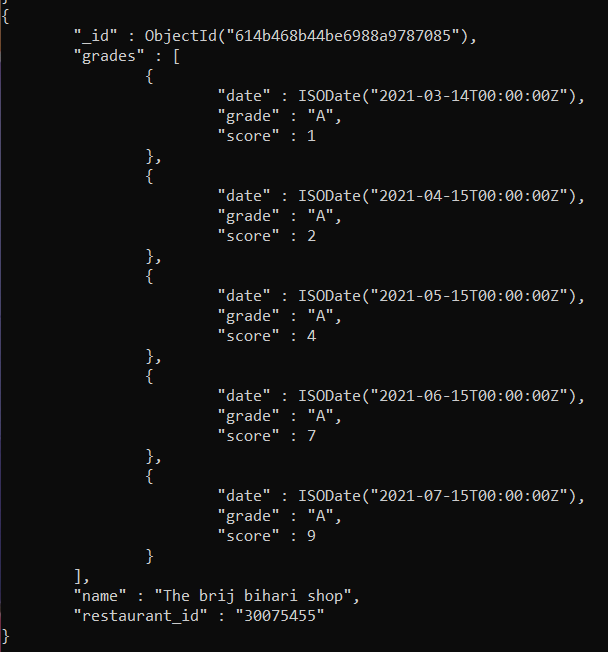
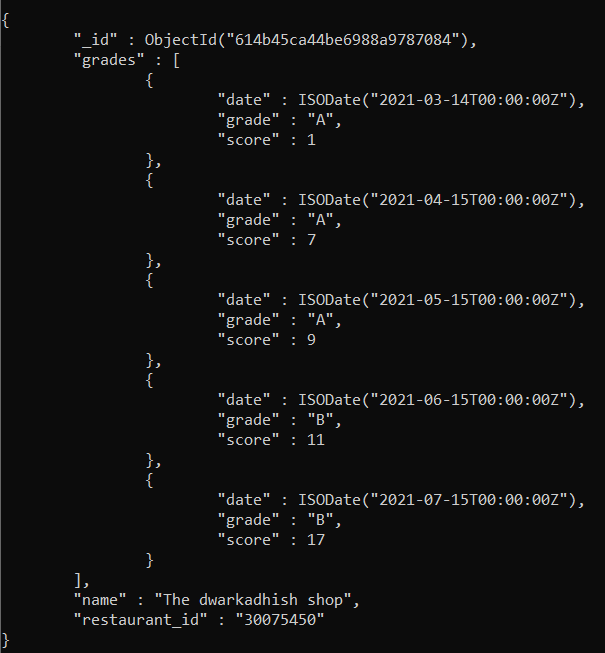




**Q 30) Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 9.**

Query: db.restaurants.find({'grades.score': {$mod: [9,0]}},{'restaurant\_id':1,'name':1, 'grades':1}).pretty()

Output:



**Q 31) Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'sho as three letters somewhere in its name.**

Query: db.restaurants.find({'name': /.\*sho.\*/},{'name':1,'address.coord':1,'cuisine':1,'\_id':0}).pretty()

Output:



**Q 32) Write a MongoDB query to find the restaurant name, town, longitude and latitude and cuisine for those restaurants which contain 'sta' as first three letters of its name.**

Query: db.restaurants.find({'name': /^The/},{'name':1,'town':1,'address.coord':1,'cuisine':1,'\_id':0}).pretty()

Output:

