DBMS - 1

Lab Report 4

SUBMITTED BY:
SHAHZANEER AHMED

REGISTRATION NUMBER:
SP21-BCS-087

SUBMITTED TO:
Dr. Basit Raza

DATE OF SUBMISSION:
January 09, 2022

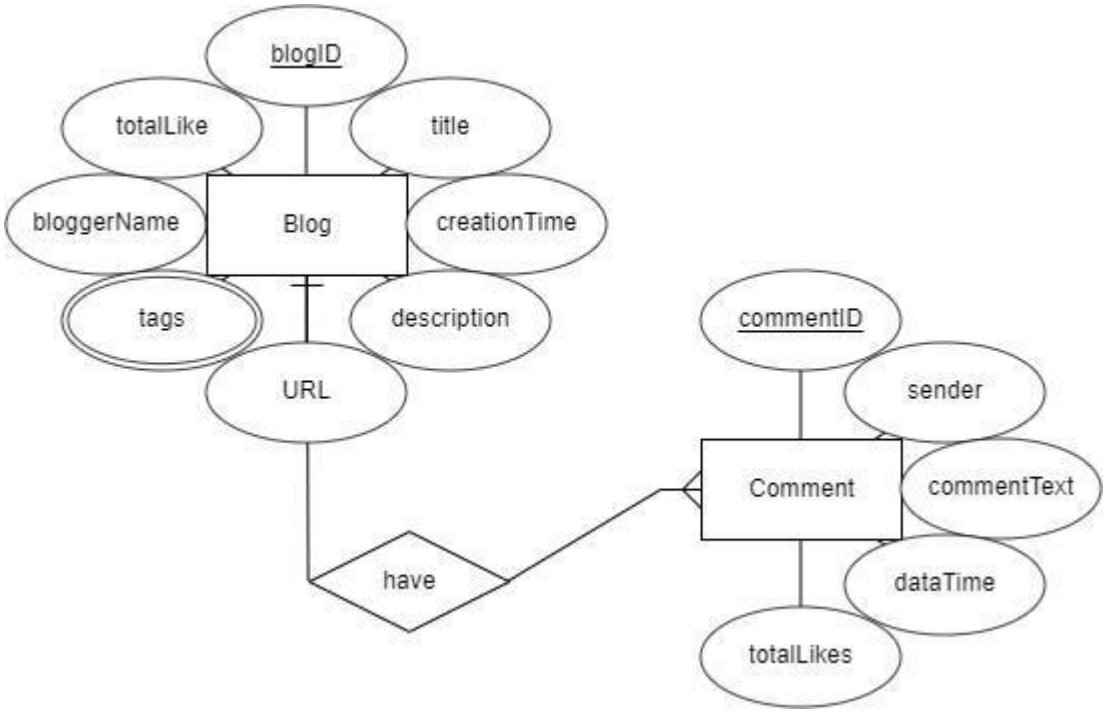
LAB 12

Q1: Suppose a simple blogging module needs to be incorporated in a website. It will allow bloggers to post their blogs on the website without the need to sign in. Every blog is going to have a unique title, a creation time, a description, a URL, a set of tags assigned by the blogger, a name of the blogger, and the count of total number of likes it receives. Other bloggers can comment on the blog. For each comment, the commentator's name, comment text, data- time and likes it receives, needs to be stored.

a) - Draw an ERD for the above blogging module.

b) - Use document model and model the same data in a single collection. Provide a sample document of that collection.

Solution: ERD:



Document Model for the Blogging Module:

Sample Document:

```
{
_id: ObjectId("5f068d5e5b5c741f91b0af91") title: "The Benefits of Meditation", creationTime: ISODate("2020-07-14T09:00:00Z"), description: "In this blog, we discuss the numerous benefits of meditation and how it can improve your overall well-being.", url: "www.meditationblog.com/benefits", tags: ["meditation", "well-being", "stressrelief"], bloggerName: "Jessica Smith", totalLikes: 500, comments:
[
{
commentatorName: "John Doe",
commentText: "I completely agree with your points about the benefits of meditation. It has really
helped me to reduce stress and improve my focus.", commentTime: ISODate("2020-0714T09:30:00Z"), commentLikes: 50
},
{
commentatorName: "Sarah Johnson",
commentText: "I've been trying to incorporate meditation into my daily routine and have noticed a huge difference in my mental clarity and overall sense of calm.",
commentTime: ISODate("2020-07-14T10:15:00Z"), commentLikes: 40
} ]
}
```

LAB 13

Write down the API calls along with the number of documents returned:

1. **Find out all the listings where property_type is House**
2. **Find out all the listings that have been received more than 100 reviews.**
3. **Find out the number of listings having 8 beds.**
4. **Find the number of AirBnB listings with 3 bedrooms and a review rating greater than 80.**
5. **Find out all the listings that offer Cable TV amenity, accommodate more than 6 individuals, and has moderate cancellation policy.**

Solution:

1- Find out all the listings where property type is House:

```
db.listings.find({property_type: "House"})
```

Returns all documents with property_type "House"

2- Find out all the listings that have been received more than 100 reviews:

```
db.listings.find({number_of_reviews: {$gt: 100}})
```

Returns all documents with number_of_reviews greater than 100

3- Find out the number of listings having 8 beds:

```
db.listings.find({beds: {$numberInt: 8}}).count()
```

Returns the number of documents with beds equal to 8

4- Find the number of AirBnB listings with 3 bedrooms and a review rating greater than 80:

```
db.listings.find({bedrooms: {$numberInt: 3},
review_scores.rating: {$gt: 80}}).count()
```

Returns the number of documents with 3 bedrooms and review_scores.rating greater than 80

5- Find out all the listings that offer Cable TV amenity, accommodate more than 6 individuals, and has moderate cancellation policy:

```
db.listings.find({amenities: "Cable TV", accommodates: {$gt: 6},
cancellation_policy:
                                    "moderate"})
```

Returns all documents with "Cable TV" in amenities array, accommodates greater than 6, cancellation_policy "moderate".

LAB 14

1. Please download the AirBnB collection from https://docs.atlas.mongodb.com/sample- data/sample-airbnb/ and import it to Mongo Atlas using mongoimport tool. Paste the screenshot of the terminal showing the command you issued for the import and also the output generated by mongoimport.
2. Create a collection corresponding to the Employees table of HR schema and insert at least 10 tuples of the table as documents in the collection
3. Update the salary of all the employees by providing a 10% increase
4. Remove all the employees from the Employees collection whose salary is less than 10000.

Solution:

1-

```
mongoimport --host <hostname>:<port> --db <database> --collection <collection> --type
csv --headerline --file F:\My Desktop\Unidocs\DB\Labs\sample_airbnb.csv
```

```
2022-12-26T18:35:34.756+0000 connected to:
mongodb://cluster0.mongodb.net/
2022-12-26T18:35:34.756+0000 dropping: test.listings
2022-12-26T18:35:35.741+0000 [##########################]
test.listings 2.4MB/2.4MB (100.0%)
2022-12-26T18:35:35.742+0000 imported 65434 documents
```

2- Create a collection corresponding to the Employees table of HR schema and insert at least 10 tuples of the table as documents in the collection:

```
db.employees.insertMany([
{
employee_id: 100, first_name: "John", last_name:
"Doe",  email: "john.doe@company.com",
phone_number:      "123-4567890", hire_date:
ISODate("202201-01"), job_id: "AD_PRES", salary:
20000, commission_pct: 0, manager_id: null,
department_id: 90
},
{
```

```
employee_id: 101, first_name: "Jane", last_name:
"Doe",

email: "jane.doe@company.com", phone_number:
"123-456-7891", hire_date: ISODate("2022-01-02"),
job_id: "AD_VP", salary: 15000, commission_pct:
0, manager_id:

100, department_id: 90

},

{

employee_id:       102, first_name: "Bob",
last_name:

"Smith",  email: "bob.smith@company.com",

    phone_number:       "123-456-

7892", hire_date: ISODate("202201-03"), job_id:
"AD_ASST", salary: 10000, commission_pct: 0,
manager_id: 101 department_id: 90

}])
```

3- To update the salary of all employees by providing a 10% increase:

db.Employees.updateMany({}, { $inc: { salary: salary * 0.1 } })

4- To remove all employees from the Employees collection whose salary is less than 10000:

db.Employees.deleteMany({ salary: { $lt: 10000 } })

LAB 15

1. **Using MongoDB aggregation framework over the movies collection, find out the number of movies of each genre. How many movies fall under "Thriller" genre?**
2. **How many movies in the collection have IMDB rating greater than or equal to 9.5?**
3. **Which movie has won the most awards?**
4. **How many movies are there in the collection that belong to Comedy genre, have IMDB rating greater than 8.0, and have won more than 50 awards?**

Solution:

1- To find the number of movies of each genre using the aggregation framework:

```
[
{ $group: {
_id:  "$genre", count: {
$sum: 1}}
}
]
```

This pipeline will group the documents in the movies collection by the "genre" field and then use the $sum operator to count the number of documents in each group.

2- To find the number of movies in the "Thriller" genre, you can use the following pipeline:

```
[
{
$match:
{
genre:"Thriller}
},
{
$count: "total"
}
]
```

This pipeline will use the $match operator to filter the documents in the collection to only those that have a "genre" field equal to "Thriller", and then the $count operator to count the number of documents in the filtered set.

3- To find the number of movies with an IMDB rating greater than or equal to 9.5, you can use the following pipeline:

```
[
{
$match: {
imdb_rating: { $gte: 9.5 }
}
},
{
$count: "total"
]
```

This pipeline will use the $match operator to filter the documents in the collection to only those that have an "imdb_rating" field greater than or equal to 9.5, and then the $count operator to count the number of documents in the filtered set.

4- To find the movie that has won the most awards, you can use the following pipeline:

```
[
{
$sort:  { awards: -1  }
},
{
$limit: 1
}
]
```

This pipeline will use the $sort operator to sort the documents in the collection in descending order based on the "awards" field, and then the $limit operator to return only the first document in the sorted set.

5- To find the number of movies that belong to the Comedy genre, have an IMDB rating greater than 8.0, and have won more than 50 awards, you can use the following pipeline:

```
[
{
$match: {  genre:   "Comedy", imdb_rating:
{ $gte:
8.0 }, awards: { $gt: 50 }
}
},
{
$count: "total"
```

```
}
]
```

This pipeline will use the $match operator to filter the documents in the collection to only those that have a "genre" field equal to "Comedy", an "imdb_rating" field greater than or equal to 8.0, and an "awards" field greater than 50, and then the $count operator to count the number of documents in the filtered set.