

Homework # 2: MongoDB Query Language (MQL)

Tuesday, November 26th, 2024

Contribution

Jesus Salomon : * Q1, Q2, Q3
* Review Q4, Q5, Q6
* Technical advising

Deasy Pasaribu : * Q4, Q5, Q6
* Review Q1, Q2, Q3
* Documentation and submission

1. (JESUS) List properties that include a garden amenity or have both a fire extinguisher and a smoke detector. Sort by the number of beds in descending order. Return the first 10 records.
(Fields): _id, name, property_type, beds
Hint: "Garden," "Fire extinguisher," and "Smoke detector" are case-sensitive in the dataset.

```
# Access my database and collection
db = client['sample_airbnb']
collection = db['listingsAndReviews']
# Query1
query1 = {
    "$or": [
        { "amenities": "Garden" },
        {
            "$and": [
                { "amenities": "Fire extinguisher" },
                { "amenities": "Smoke detector" }
            ]
        }
    ]
}

# Define my projection
projection = {
    "_id": 1,
    "name": 1,
    "property_type": 1,
    "beds": 1
}

# Execute the query with sorting and limiting
results = collection.find(query1, projection).sort("beds", -1).limit(10)

# Print the results aka we done!
print("Query1 results:")
for document in results:
    print(document)

Query1 results:
{'_id': '20701559', 'name': 'Venue Hotel Old City', 'property_type': 'Boutique hotel', 'beds': 25}
{'_id': '2271702', 'name': 'LUXURY HOUSE IN BARRA DA TIJUCA', 'property_type': 'House', 'beds': 18}
{'_id': '3410502', 'name': 'Excellent Location and price !!', 'property_type': 'Bed and breakfast', 'beds': 16}
{'_id': '4742607', 'name': 'Casa de Vilela', 'property_type': 'Farm stay', 'beds': 16}
{'_id': '21929645', 'name': 'Good location Four Triple room One Quadruple room', 'property_type': 'Aparthotel', 'beds': 15}
{'_id': '73683', 'name': 'Sagrada Familia area for 12 people', 'property_type': 'Apartment', 'beds': 14}
{'_id': '21865684', 'name': 'Le Bed & Cocktail: Discover Montreal like a local', 'property_type': 'Hostel', 'beds': 12}
{'_id': '32280408', 'name': 'Kai uli Beachfront Estate Laumilo', 'property_type': 'House', 'beds': 11}
{'_id': '5307830', 'name': '★ Ocean/City Views★Van Included★Hot Tub★3 Bath', 'property_type': 'House', 'beds': 10}
{'_id': '2161945', 'name': 'Sand Sun Surf w Parking. City 9km', 'property_type': 'House', 'beds': 10}
```

2. (JESUS) List the 5 most expensive properties based on listing price.
(Fields): _id, name, price

```
# Access my database and collection
db = client['sample_airbnb']
collection = db['listingsAndReviews']

# Define my projection ☐
projection = {
    "_id": 1,
    "name": 1,
    "price": 1
}

# Execute the query with sorting and limiting ☰
results = collection.find({}, projection).sort("price", -1).limit(5)

# Print the results aka we done! 🎉
print("Query2 results:")
for document in results:
    print(document)

Query2 results:
{'_id': '20275354', 'name': 'İstanbul un kalbi sisli. Center of istanbul sisli', 'price': Decimal128('48842.00')}
{'_id': '14644562', 'name': '良德街3号温馨住宅', 'price': Decimal128('11681.00')}
{'_id': '27593455', 'name': 'HS1-2人大床房+丰泽、苏宁、百脑汇+女人街+美食中心', 'price': Decimal128('11681.00')}
{'_id': '13997910', 'name': 'Apartamento de luxo em Copacabana - 4 quartos', 'price': Decimal128('11190.00')}
{'_id': '22200454', 'name': 'LM 三個睡房的整间公寓', 'price': Decimal128('10001.00')}
```

3. (JESUS) List properties where the cleanliness rating is higher than the accuracy rating. Sort by the number of reviews in ascending order. Return the first 5 records.

(Fields): _id, name, number_of_reviews, review_scores_cleanliness, review_scores_accuracy

```
# Access my database and collection
db = client['sample_airbnb']
collection = db['listingsAndReviews']

# Query3
query3 = {
    "$expr": {
        "$gt": [
            "$review_scores.review_scores_cleanliness",
            "$review_scores.review_scores_accuracy"
        ]
    }
}

# Define my projection ☐
projection = {
    "_id": 1,
    "name": 1,
    "number_of_reviews": 1,
    "review_scores_cleanliness": "$review_scores.review_scores_cleanliness",
    "review_scores_accuracy": "$review_scores.review_scores_accuracy"
}

# Execute the query with sorting and limiting 🕵
results = collection.find(query3, projection).sort("number_of_reviews", 1).limit(5)

# Print the results aka we done! 🎉
print("Query results:")
for document in results:
    print(document)
```

```
Query results:
{'_id': '11007658', 'name': 'Alcam Colón 42 Apartment', 'number_of_reviews': 1, 'review_scores_cleanliness': 8, 'review_scores_accuracy': 6}
{'_id': '12157453', 'name': 'Honolulu 1 BR/1Bath Condo - Hilton Hawaiian', 'number_of_reviews': 1, 'review_scores_cleanliness': 10, 'review_scores_accuracy': 8}
{'_id': '12591225', 'name': 'Cosy penthouse close to Barra beach', 'number_of_reviews': 1, 'review_scores_cleanliness': 10, 'review_scores_accuracy': 8}
{'_id': '12354373', 'name': '2017 , férias no Rio', 'number_of_reviews': 1, 'review_scores_cleanliness': 10, 'review_scores_accuracy': 6}
{'_id': '10228731', 'name': 'Quarto inteiro na Tijuca', 'number_of_reviews': 1, 'review_scores_cleanliness': 10, 'review_scores_accuracy': 8}
```

4. (DEASY) List properties located in Portugal with 3 or more bedrooms. Sort by review score rating in descending order and, if ratings are the same, sort by the number of reviews in descending order. List the first 5 records.

(Fields): _id, name, address.country, review_scores_rating, bedrooms, number_of_reviews

```
# Access to database and collection
db = client['sample_airbnb']
collection = db['listingsAndReviews']

# Define the query
query4 = {
    "address.country": "Portugal",
    "bedrooms": {"$gte": 3},
}

# Define the projection
projection = {
    "_id": 1,
    "name": 1,
    "address.country": 1,
    "review_scores_rating": 1,
    "bedrooms": 1,
    "number_of_reviews": 1,
    "review_scores_rating": "$review_scores.review_scores_rating"
}

# Execute the query
results = list(collection.find(query4, projection).sort("review_scores_rating", -1).sort("number_of_reviews", -1).limit(5))

# Print the result
print("Query 4 Result:")
if results:
    for results in results:
        print(results)
else:
    print("No results found.")
```

```
Query 4 Result:
{'_id': '6836536', 'name': 'Lulapartment', 'bedrooms': 3, 'number_of_reviews': 239, 'address': {'country': 'Portugal'}, 'review_scores_rating': 92}
{'_id': '6868273', 'name': 'Casa da Hera - The Ivy House', 'bedrooms': 3, 'number_of_reviews': 236, 'address': {'country': 'Portugal'}, 'review_scores_rating': 94}
{'_id': '16147485', 'name': 'SANTA CATARINA HOUSE', 'bedrooms': 3, 'number_of_reviews': 221, 'address': {'country': 'Portugal'}, 'review_scores_rating': 97}
{'_id': '12400621', 'name': 'Elegant suite in historic center', 'bedrooms': 5, 'number_of_reviews': 203, 'address': {'country': 'Portugal'}, 'review_scores_rating': 94}
{'_id': '186655', 'name': 'Charming Family House with Garden (6981/AL)', 'bedrooms': 3, 'number_of_reviews': 171, 'address': {'country': 'Portugal'}, 'review_scores_rating': 95}
```

5. (DEASY) List properties with fewer than 500 reviews that have a review score of 9 or above for communication. Sort by review score for checkin in descending order, and if scores are the same, sort by the number of reviews in descending order. Display the first 5 records.

(Fields): _id, name, number_of_reviews, review_scores_communication, review_scores_checkin

```
# Access to database and collection
db = client['sample_airbnb']
collection = db['listingsAndReviews']

# Define the query
query5 = {
    "number_of_reviews": {"$lt": 500},
    "review_scores_communication": {"$gte": 9}
}

# Define the projection
projection = {
    "_id": 1,
    "name": 1,
    "number_of_reviews": 1,
    "review_scores_communication": 1,
    "review_scores_checkin": 1
}

# Execute the query
results = list(collection.find(query5, projection).sort("review_scores_checkin", -1).sort("number_of_reviews", -1).limit(5))

# Print the result
print("Query 5 Result:")
if results:
    for results in results:
        print(results)
else:
    print("No results found.")
```

```
Query 5 Result:
No results found.
```

6. (DEASY) List the 5 hosts with the most bedrooms across all of their property listings as type 'House'. Group by host to calculate the total number of bedrooms for each host. Sort the results by the total number of bedrooms in descending order and display each host's ID, name, and total bedrooms. (Fields): host_id, host_name, total_bedrooms

Hint: Consider using an aggregation pipeline with \$match, \$group, and \$sort stages to filter, group, and sort the results. The \$sum operator in the \$group stage can be used to calculate the total number of bedrooms across a host's listings.

```
# Access to database and collection
db = client['sample_airbnb']
collection = db['listingsAndReviews']

# Define the query
query6 = {
    "property_type": "House",
    "bedrooms": {"$gte": 1}
}

# Define the projection
projection = {
    "_id": 1,
    "host.host_id": 1,
    "host.host_name": 1,
    "bedrooms": 1
}

# Define aggregation pipeline then execute
pipeline = [
    {"$match": query6},
    {"$group": {
        "_id": {"host_id": "$host.host_id", "host_name": "$host.host_name"},
        "total_bedrooms": {"$sum": "$bedrooms"}}
    },
    {"$sort": {"total_bedrooms": -1}},
    {"$limit": 5}
]
results = list(collection.aggregate(pipeline))

# Print the result
print("Query 6 Results:")
if results:
    for host in results:
        print({
            "host_id": host["_id"]["host_id"],
            "host_name": host["_id"]["host_name"],
            "total_bedrooms": host["total_bedrooms"]})
else:
    print("No results found.")
```

```
Query 6 Results:
{'host_id': '10496350', 'host_name': 'Elite', 'total_bedrooms': 13}
{'host_id': '67742378', 'host_name': 'Fernando', 'total_bedrooms': 9}
{'host_id': '11914644', 'host_name': 'Luxico Holiday Homes', 'total_bedrooms': 9}
{'host_id': '1191923', 'host_name': 'Hawaii', 'total_bedrooms': 7}
{'host_id': '5167360', 'host_name': 'Bailey', 'total_bedrooms': 7}
```