Data Science Homework 5

By: Rania Siddiqui 07494
rs07494@st.habib.edu.pk

1. the count of total number of records in the collection.

Query:

```
db.airbnb.countDocuments()
```

Output:

```
DS
The find query will be run with Query Assist.

5555
```

Interpretation: This query returns the total number of records in the `airbnb` collection. It gives an overall idea of how much data we are working with. For example, we can see that there are 5555 numbers of records, so we know there are 5555 airbnb listings or entries in the dataset.

2. the count of records for property_type “House”.

Query:

```
db.airbnb.countDocuments({ property_type: "House" })
```

Output:

```
DS
The find query will be run with Query Assist.

606
```

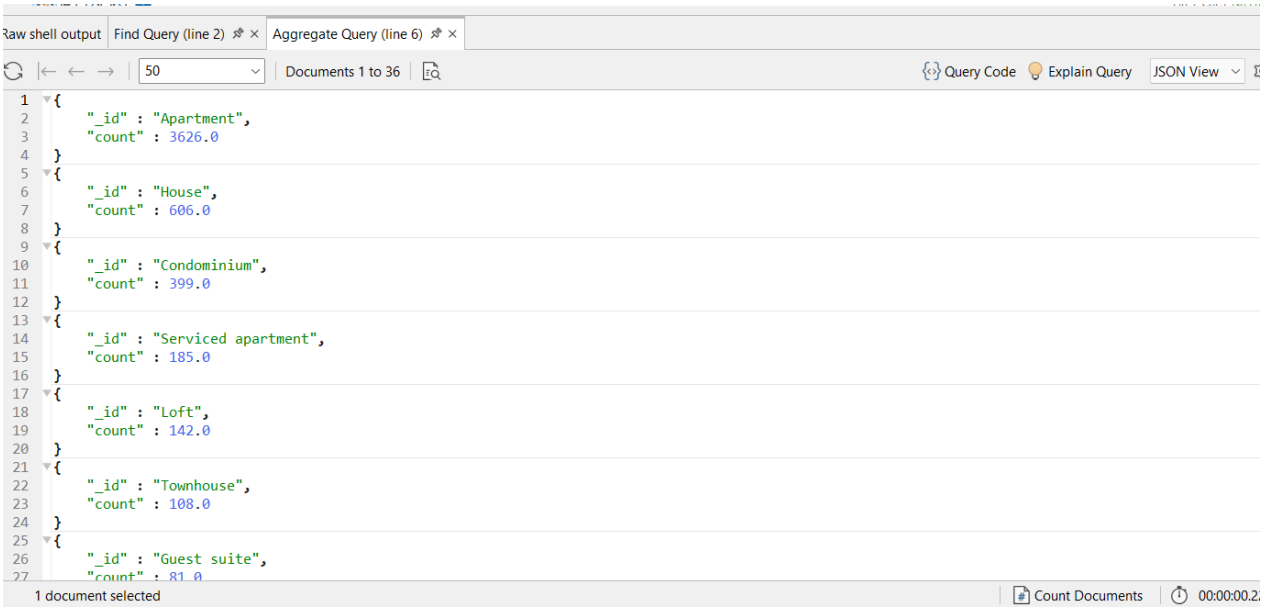
Interpretation: This counts how many properties in the dataset are specifically of the type "House." There are 606 records for property type ‘house’ in the collection of airbnb. This can indicate the popularity or availability of houses compared to other property types (e.g., apartments or condos).

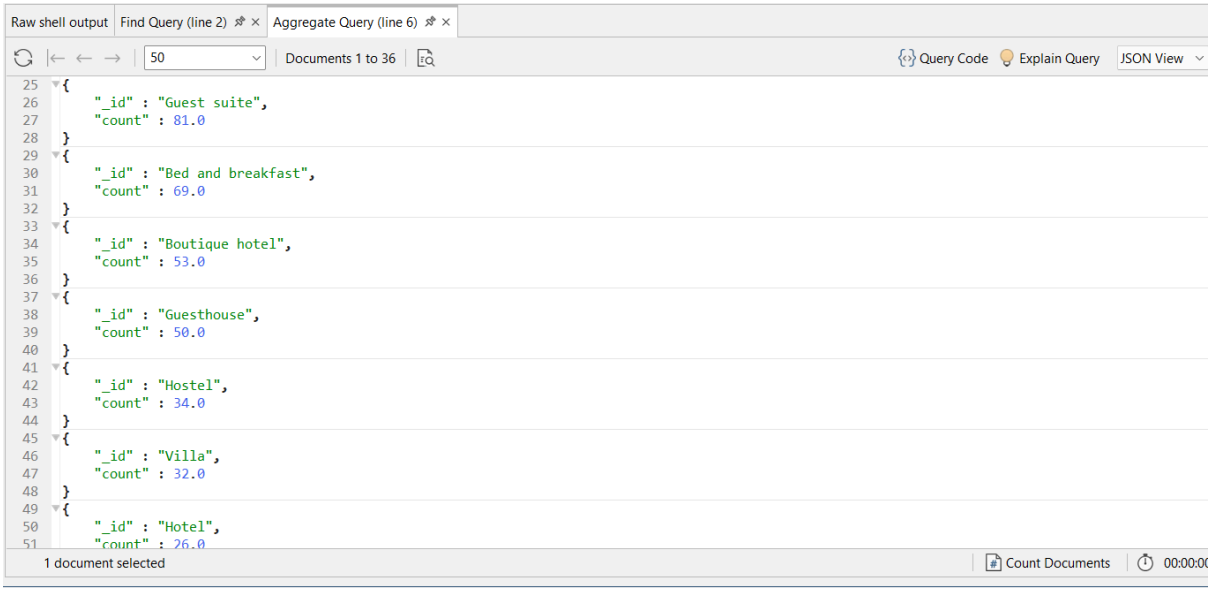
3. counts for each property_type in descending order of count (use group).

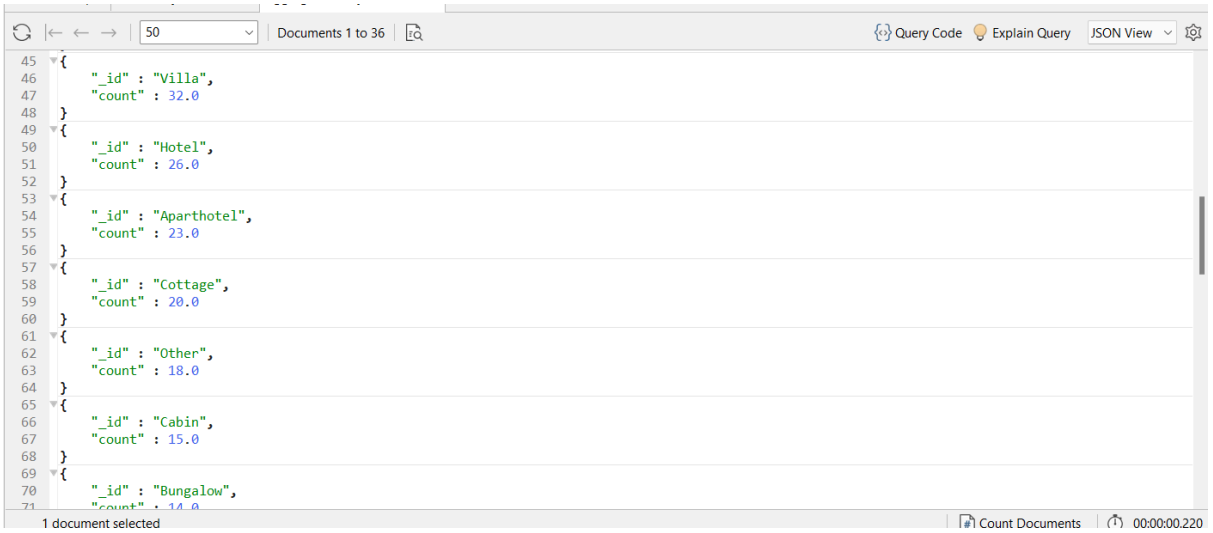
Query:

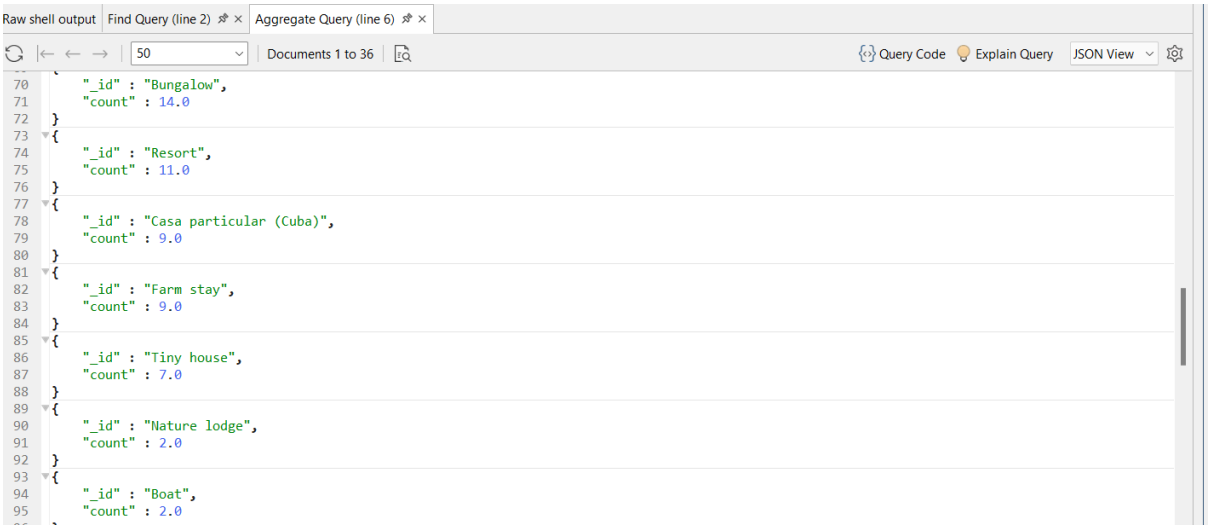
```
db.airbnb.aggregate([
 { $group: { _id: "$property_type", count: { $sum: 1 } } },
 { $sort: { count: -1 } }
])
```

Output(Raw shell output in JSON)


Raw shell output
Find Query (line 2)
Aggregate Query (line 6)
50
Documents 1 to 36
Query Code
Explain Query
JSON View
{
"_id" : "Apartment",
"count" : 3626.0
}
{
"_id" : "House",
"count" : 606.0
}
{
"_id" : "Condominium",
"count" : 399.0
}
{
"_id" : "Serviced apartment",
"count" : 185.0
}
{
"_id" : "Loft",
"count" : 142.0
}
{
"_id" : "Townhouse",
"count" : 108.0
}
{
"_id" : "Guest suite",
1 document selected
Count Documents


Raw shell output
Find Query (line 2)
Aggregate Query (line 6)
50
Documents 1 to 36
Query Code
Explain Query
JSON View
{
"_id" : "Guest suite",
"count" : 81.0
}
{
"_id" : "Bed and breakfast",
"count" : 69.0
}
{
"_id" : "Boutique hotel",
"count" : 53.0
}
{
"_id" : "Guesthouse",
"count" : 50.0
}
{
"_id" : "Hostel",
"count" : 34.0
}
{
"_id" : "Villa",
"count" : 32.0
}
{
"_id" : "Hotel",
1 document selected
Count Documents


50
Documents 1 to 36
Query Code
Explain Query
JSON View
{
"_id" : "Villa",
"count" : 32.0
}
{
"_id" : "Hotel",
"count" : 26.0
}
{
"_id" : "Aparthotel",
"count" : 23.0
}
{
"_id" : "Cottage",
"count" : 20.0
}
{
"_id" : "Other",
"count" : 18.0
}
{
"_id" : "Cabin",
"count" : 15.0
}
{
"_id" : "Bungalow",
1 document selected
Count Documents
00:00:00.220



Note: The output was too long, but above are a few snippets of how my output looks like in raw json format.

Interpretation:
This shows how many records exist for each `property_type`, such as houses, apartments, etc., sorted from the most to the least common. From this, we can easily identify the most popular property types in the dataset, helping us understand the distribution of property listings.

4. the count of records for review_scores_cleanliness less than 5.

Query:

```
db.airbnb.find({ "review_scores.review_scores_cleanliness": { $lt: 5 } }).count()
```

Output:

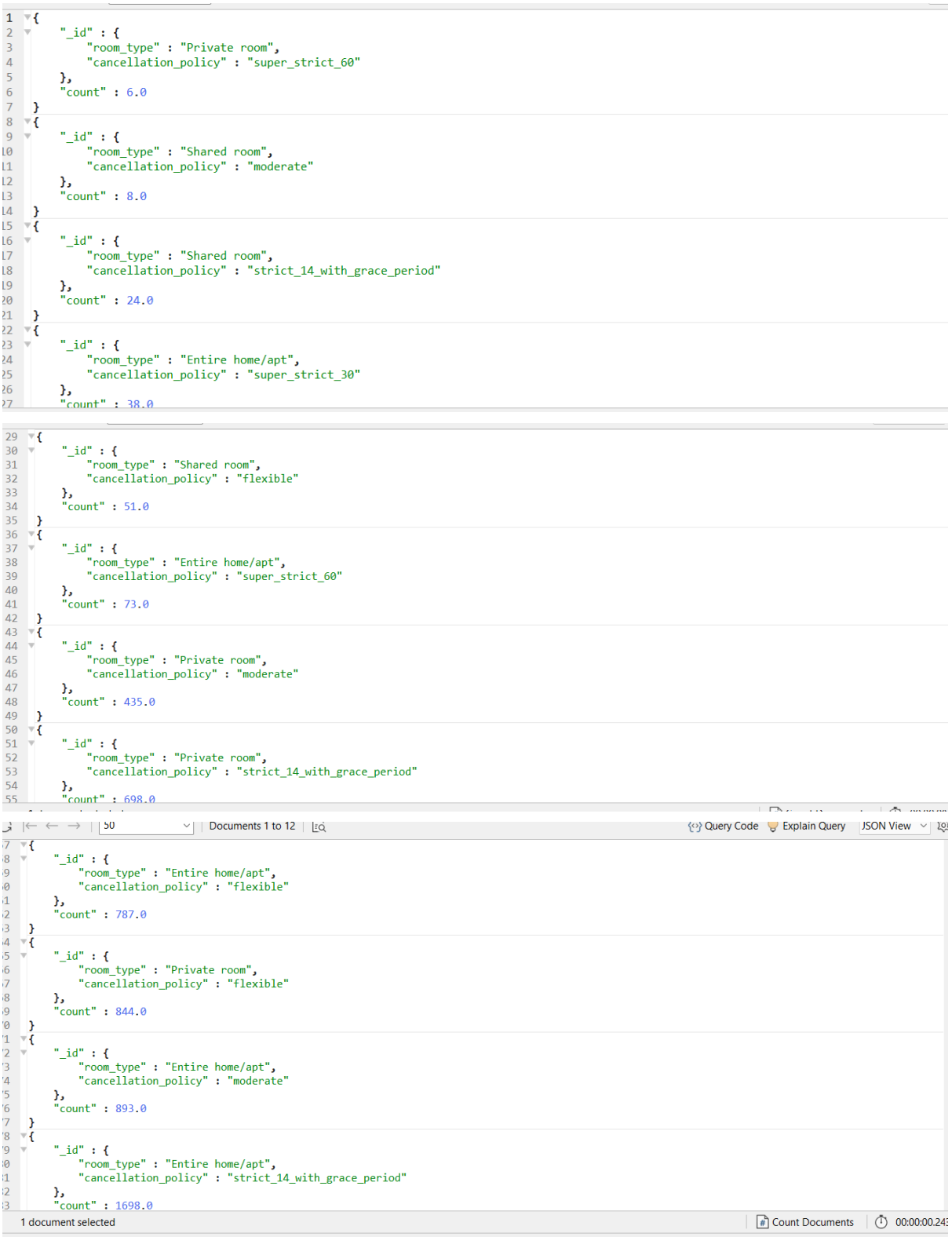
```
DS
The find query will be run with Query Assist.

38
```

Interpretation: This counts the number of records with a cleanliness review score of less than 5. A high count, in our case which is 38, indicates cleanliness issues. This can help focus on improving properties with low cleanliness scores.

5. the count of records for each room_type, cancellation_policy. Order the output by count in ascending order.

Query:

```
db.airbnb.aggregate([ { $group: { _id: { room_type: "$room_type", cancellation_policy: "$cancellation_policy" }, count: { $sum: 1 } } }, { $sort: { count: 1 } } ])
```



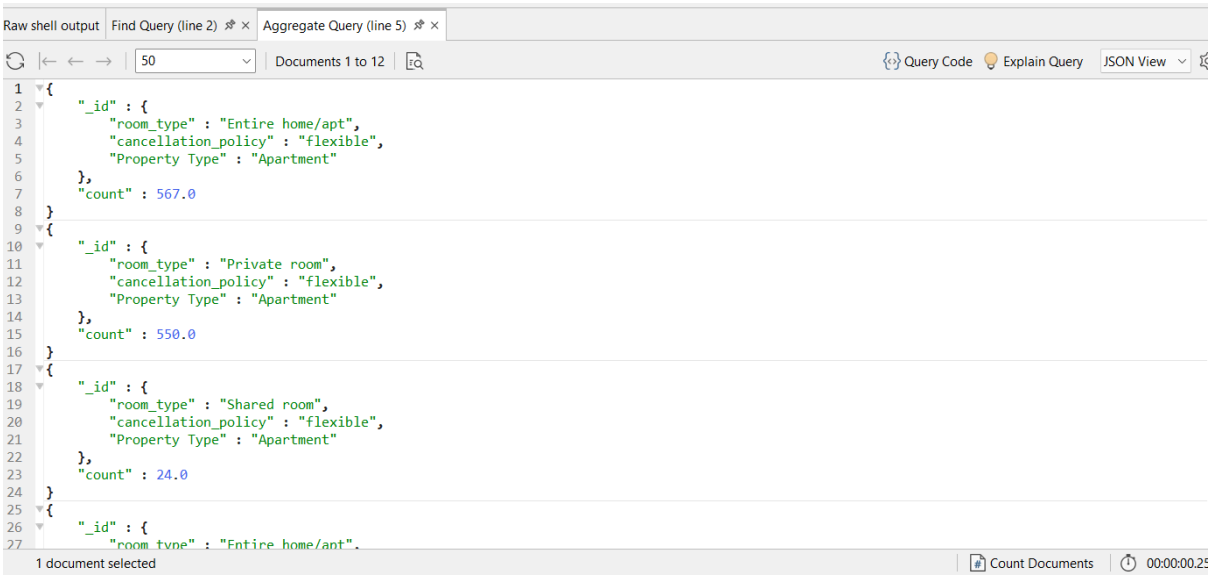
Interpretation: This groups the data by `room_type` (e.g., entire home, shared room) and `cancellation_policy` (e.g., strict, flexible) and counts the number of records for each combination. Sorting by count in ascending order helps identify us less common room type and cancellation policy combinations, which could reveal niche areas.

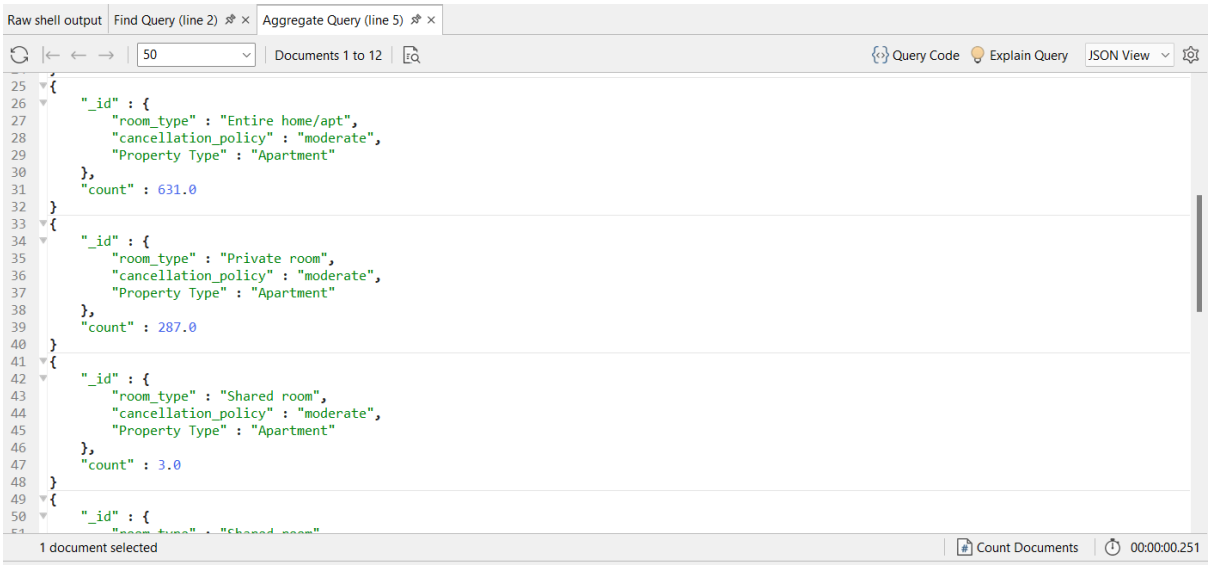
6. Similar to question 5, the count of records for each room_type, cancellation_policy when property_type is “Apartment”. Order the output by cancellation_policy in ascending order.

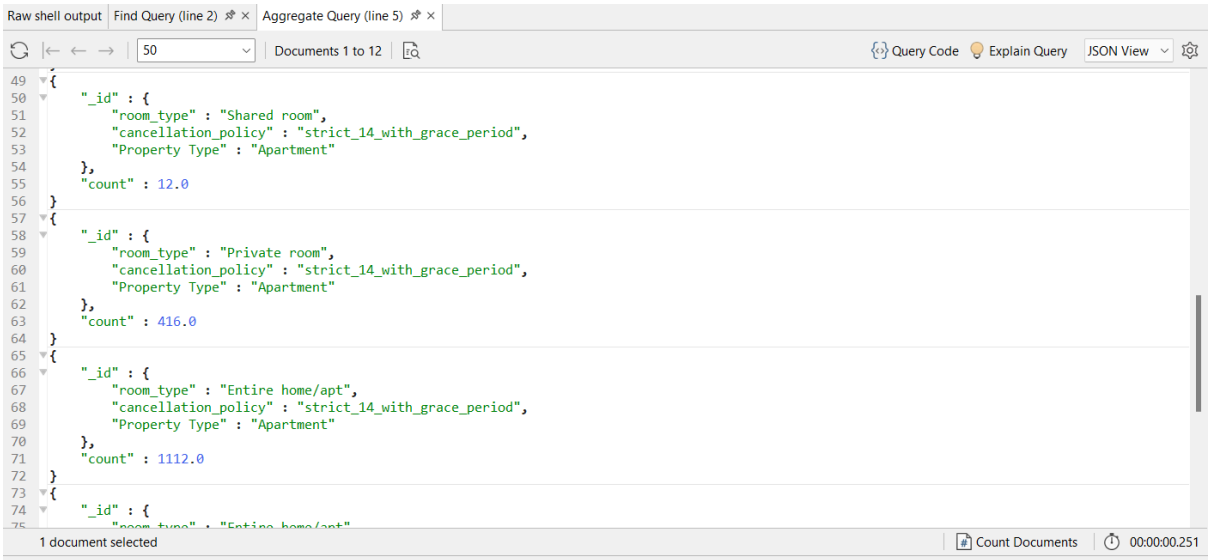
Query:

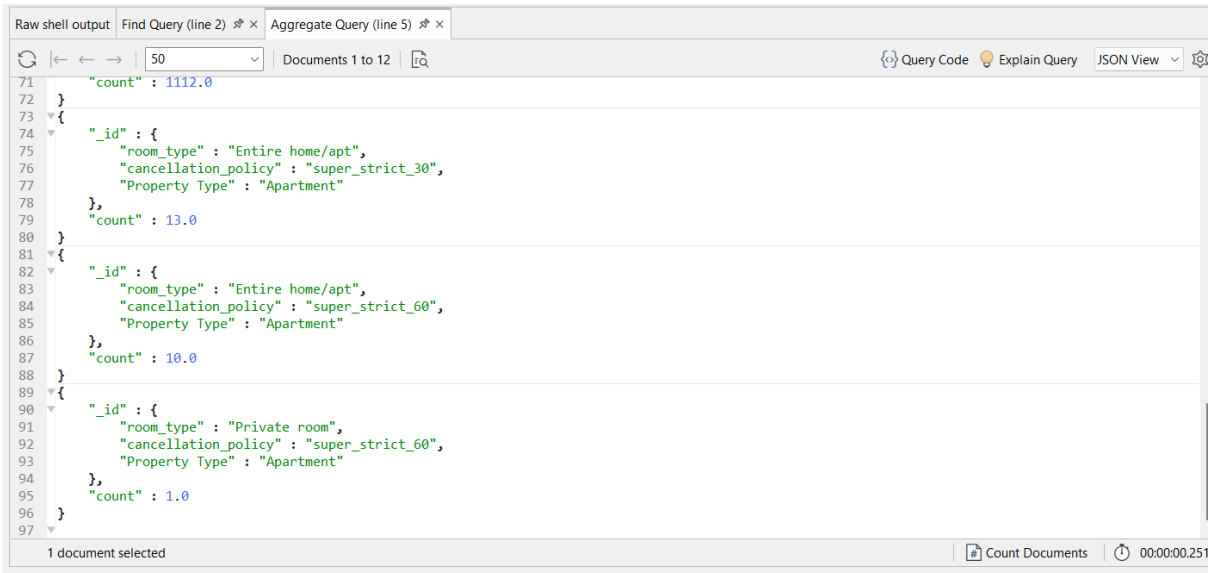
```
db.airbnb.aggregate([
          { $match: { property_type: "Apartment" } },
  {
     $group: {
        _id: {
           room_type: "$room_type",
           cancellation_policy: "$cancellation_policy",
           "Property Type":"$property_type"
        },
        count: { $sum: 1 }
     }
  },
  {
     $sort: { "_id.cancellation_policy": 1 }
   }
]);
```

Output:









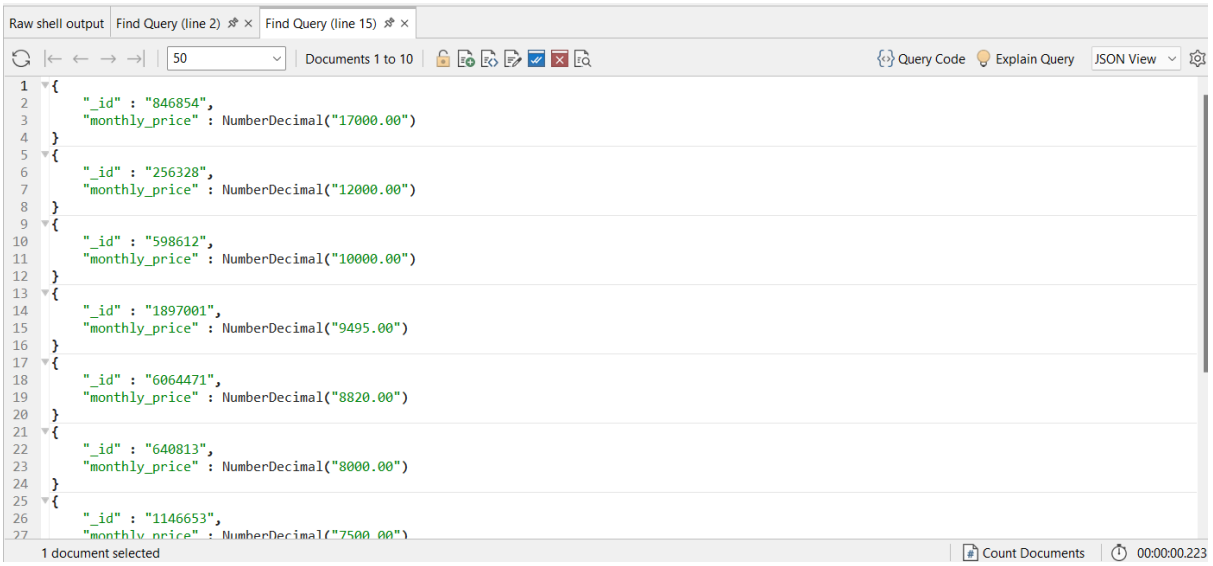
Interpretation:
This can help us understand the flexibility or strictness of cancellation policies specifically for apartments.

7. Find all the records where address.market is “New York” and monthly_price is greater than 5000. Display only id and monthly_price and sort the output by monthly_price in descending order.

Query:

```
db.airbnb.find(
 { "address.market": "New York", monthly_price: { $gt: 5000 } },
 { _id: 1, monthly_price: 1 }
).sort({ monthly_price: -1 })
```

Output:




```
{
    "_id" : "1155475",
    "monthly_price" : NumberDecimal("6999.00")
}
{
    "_id" : "2253500",
    "monthly_price" : NumberDecimal("6250.00")
}
{
    "_id" : "102995",
    "monthly_price" : NumberDecimal("6000.00")
}
```


Interpretation:
This gives insight into the high-end rental market in New York, showing the most expensive properties.

8. Display the records with cleaning_fee in descending order. Display name, property_type and cleaning_fee. Limit records to 10 (easy)

```
db.airbnb.find(
  { cleaning_fee: { $exists: true } },
  { name: 1, property_type: 1, cleaning_fee: 1 }
).sort({ cleaning_fee: -1 }).limit(10)
```


```
{
    "_id" : "1176693",
    "name" : "BEST REVIEWS*BEST MALLS*SAFE STAY*DIMSUM*CWB*MTR",
    "property_type" : "Apartment",
    "cleaning_fee" : NumberDecimal("942.00")
}
{
    "_id" : "20362690",
    "name" : "WORLD CLASS MALLS*LUXURY 3BED2BATH*CLEAN*MTR*SAFE",
    "property_type" : "Apartment",
    "cleaning_fee" : NumberDecimal("942.00")
}
{
    "_id" : "15488401",
    "name" : "INSTAGRAM HOME *MTR*ELEMENT MALL*NEW TRAIN KOWLOON",
    "property_type" : "Apartment",
    "cleaning_fee" : NumberDecimal("942.00")
}
{
    "_id" : "16215566",
    "name" : "Kahala Ali'i",
    "property_type" : "House",
    "cleaning_fee" : NumberDecimal("910.00")
}
```



```
{
    "_id" : "5640127",
    "name" : "Laulea Kailani Villa (Kauai), Peaceful Luxury",
    "property_type" : "House",
    "cleaning_fee" : NumberDecimal("850.00")
}
{
    "_id" : "25065550",
    "name" : "Lanikai La'i Hale with Oceanview 3BD plus Loft/3BA",
    "property_type" : "House",
    "cleaning_fee" : NumberDecimal("800.00")
}
```


Interpretation:
This can be useful for identifying properties that charge a high fee for cleaning, which may be correlated with higher-end or luxury listings.

9. Come up with your own query to show any interesting insight. Use atleast two fields for match and two fields for group.

Query:

```
db.airbnb.aggregate([
 {
  $match: {
    accommodates: { $gte: 4 },
    bathrooms: { $gt: 1 }
  }
```

```
},
{
  $group: {
    _id: { room_type: "$room_type", property_type: "$property_type" },
    avg_price: { $avg: "$price" },
    total_reviews: { $sum: "$number_of_reviews" }
  }
},
{
  $sort: { avg_price: -1 }
}
])
```

The output was too long so I have included just a few snippets.

Interpretation:

This query helps us explore pricing trends and the popularity of different types of large accommodations with multiple bathrooms. It provides a way to compare higher-end properties against each other based on price and guest feedback. Moreover, the `total_reviews` metric

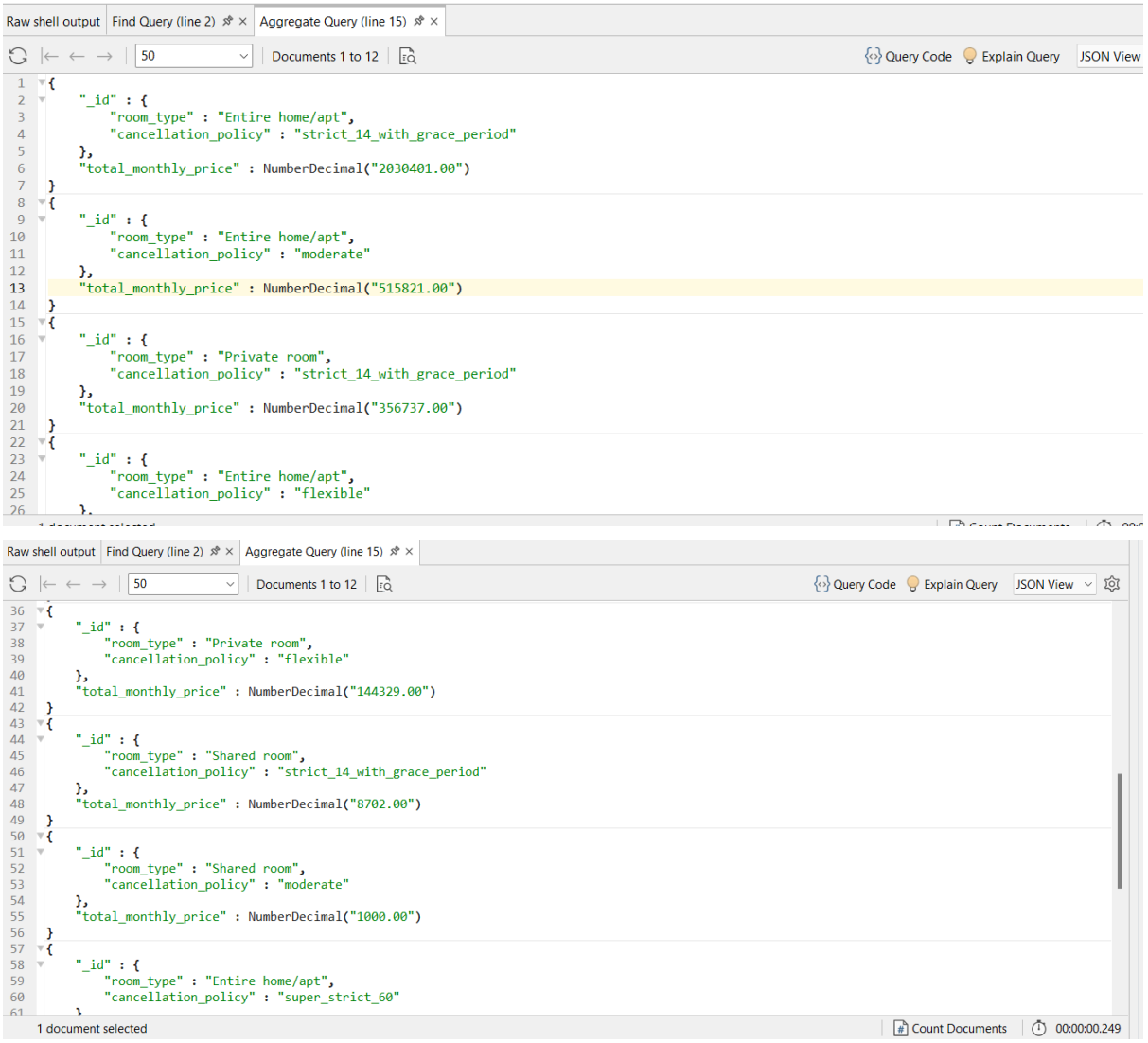
allows us to assess which room/property types are attracting the most attention or bookings. If a property type has a high price but low reviews, it suggests that it is less popular, or perhaps it's new to the market.

10. . Come up with your own query to show any interesting insight. Use atleast two fields for group.

Show the total monthly_price for each combination of room_type and cancellation_policy:

```
db.airbnb.aggregate([
 { $group: { _id: { room_type: "$room_type", cancellation_policy: "$cancellation_policy" },
total_monthly_price: { $sum: "$monthly_price" } } },
 { $sort: { total_monthly_price: -1 } }
])
```

Output:



Interpretation:
The result shows which room type and cancellation policy combinations contribute the most revenue or listing prices, helping us identify the most financially impactful configurations.