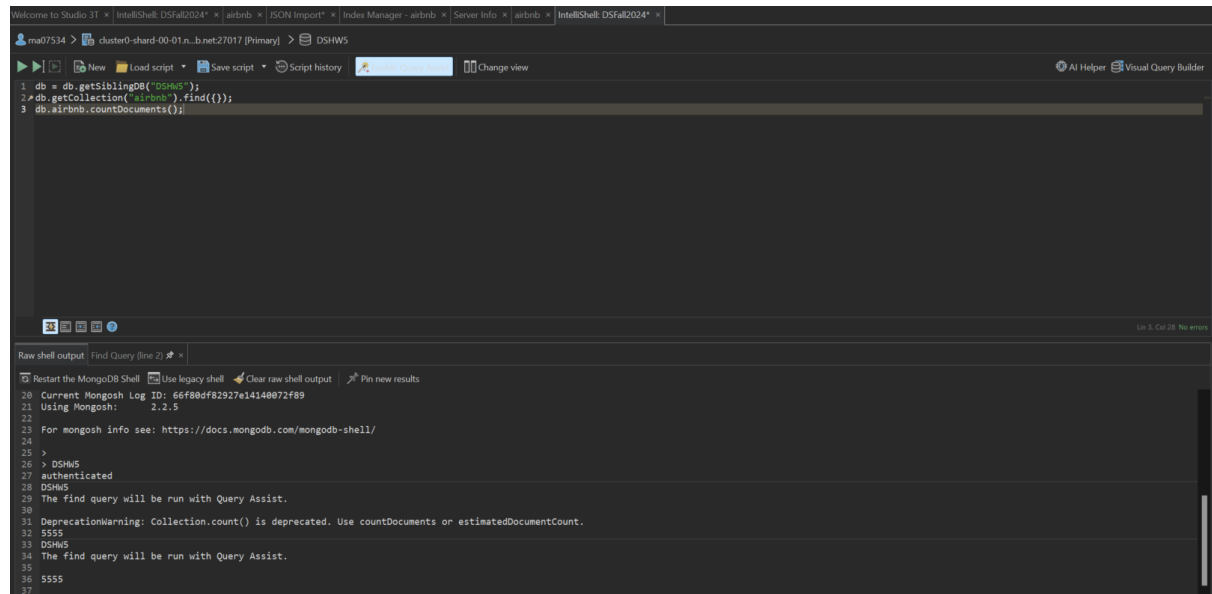


Data Science HW5:

M.Abdullah (ma07534)

Query the collection and interpret the results, displaying:

Q1. the count of total number of records in the collection



The screenshot shows the MongoDB Studio interface. The top toolbar includes buttons for 'New', 'Load script', 'Save script', 'Script history', and 'Change view'. The main editor area contains the following JavaScript code:

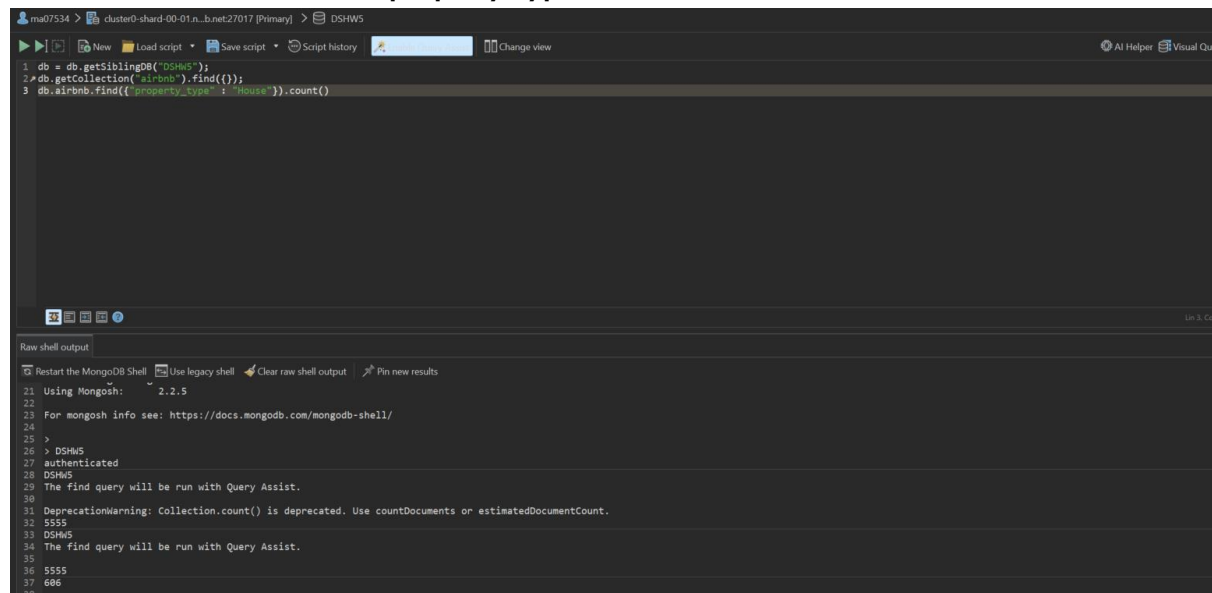
```
1 db = db.getSiblingDB('DSHW5');
2 db.getCollection('airbnb').find({});
3 db.airbnb.countDocuments();
```

The 'Raw shell output' pane at the bottom shows the execution results:

```
Restart the MongoDB Shell | Use legacy shell | Clear raw shell output | Pin new results
20 Current Mongosh Log ID: 66f88df82927e14140872f89
21 Using Mongosh: 2.2.5
22
23 For mongosh info see: https://docs.mongodb.com/mongodb-shell/
24
25 >
26 > DSHW5
27 authenticated
28 DSHW5
29 The find query will be run with Query Assist.
30
31 DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
32 5555
33 DSHW5
34 The find query will be run with Query Assist.
35
36 5555
37
```

Interpretation: The query output displays that there are a total of 5555 records in the collection.

Q2. the count of records for property_type "House"



The screenshot shows the MongoDB Studio interface. The top toolbar includes buttons for 'New', 'Load script', 'Save script', 'Script history', and 'Change view'. The main editor area contains the following JavaScript code:

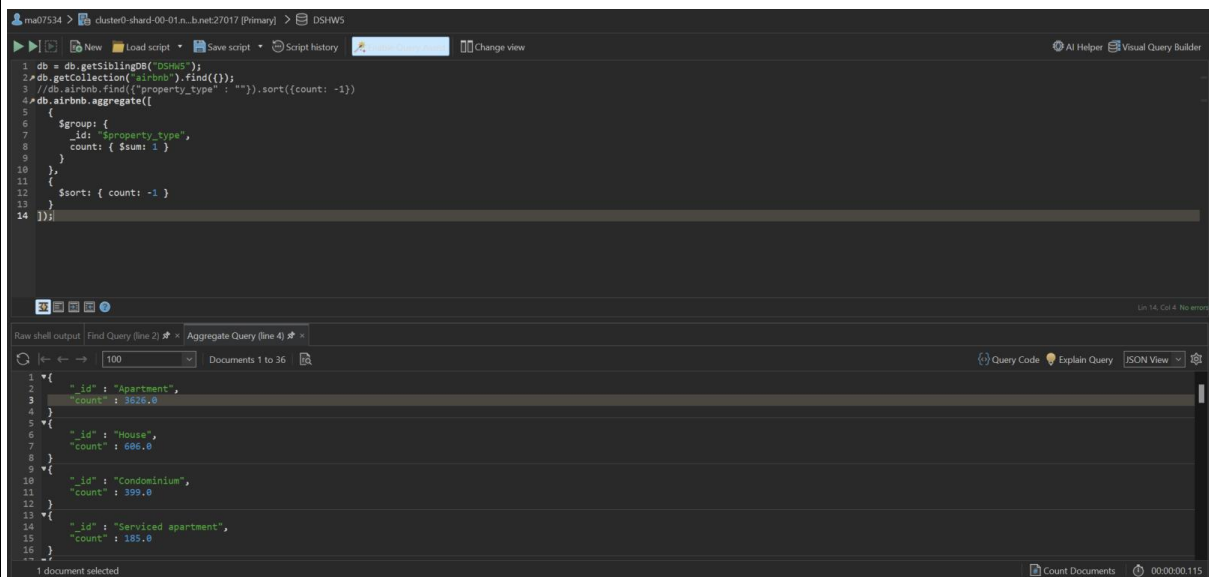
```
1 db = db.getSiblingDB('DSHW5');
2 db.getCollection('airbnb').find({});
3 db.airbnb.find({'property_type': 'House'}).count();
```

The 'Raw shell output' pane at the bottom shows the execution results:

```
Restart the MongoDB Shell | Use legacy shell | Clear raw shell output | Pin new results
21 Using Mongosh: 2.2.5
22
23 For mongosh info see: https://docs.mongodb.com/mongodb-shell/
24
25 >
26 > DSHW5
27 authenticated
28 DSHW5
29 The find query will be run with Query Assist.
30
31 DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
32 5555
33 DSHW5
34 The find query will be run with Query Assist.
35
36 5555
37 606
38
```

Interpretation: The query output displays that there are a total of 606 records of property_type "House"

Q3. counts for each property_type in descending order of count (use group)



The screenshot shows a MongoDB query editor with the following aggregate query:

```
1 db = db.getSiblingDB('DSHW5');
2 db.getCollection('airbnb').find({});
3 //db.airbnb.find({"property_type": ""}).sort({"count": -1})
4 db.airbnb.aggregate([
5   {
6     $group: {
7       _id: "$property_type",
8       count: { $sum: 1 }
9     }
10  },
11  {
12    $sort: { count: -1 }
13  }
14 ])
```

The results are displayed in JSON format, showing the count for each property type in descending order:

```
1 {
2   "_id": "Apartment",
3   "count": 3626.0
4 }
5 {
6   "_id": "House",
7   "count": 606.0
8 }
9 {
10  "_id": "Condominium",
11  "count": 399.0
12 }
13 {
14  "_id": "Serviced apartment",
15  "count": 185.0
16 }
17 ..
```

```
{
  "_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",  
  "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",  
  "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",
    "count" : 14.0
  }
  {
    "_id" : "Resort",
    "count" : 11.0
  }
  {
    "_id" : "Farm stay",
    "count" : 9.0
  }
  {
    "_id" : "Casa particular (Cuba)",
    "count" : 9.0
  }
  {
    "_id" : "Tiny house",
    "count" : 7.0
  }
  {
    "_id" : "Camper/RV",
    "count" : 2.0
  }
  {
    "_id" : "Nature lodge",
    "count" : 2.0
  }
  {
    "_id" : "Chalet",
    "count" : 2.0
  }
```

```
}  
  
{  
  "_id" : "Boat",  
  "count" : 2.0  
}  
  
{  
  "_id" : "Earth house",  
  "count" : 1.0  
}  
  
{  
  "_id" : "Castle",  
  "count" : 1.0  
}  
  
{  
  "_id" : "Barn",  
  "count" : 1.0  
}  
  
{  
  "_id" : "Pension (South Korea)",  
  "count" : 1.0  
}  
  
{  
  "_id" : "Campsite",  
  "count" : 1.0  
}  
  
{  
  "_id" : "Train",  
  "count" : 1.0  
}  
  
{  
  "_id" : "Heritage hotel (India)",  
  "count" : 1.0  
}  
  
{  
  "_id" : "Hut",  
  "count" : 1.0  
}  
  
{
```

```

    "_id" : "Treehouse",

    "count" : 1.0
}

{

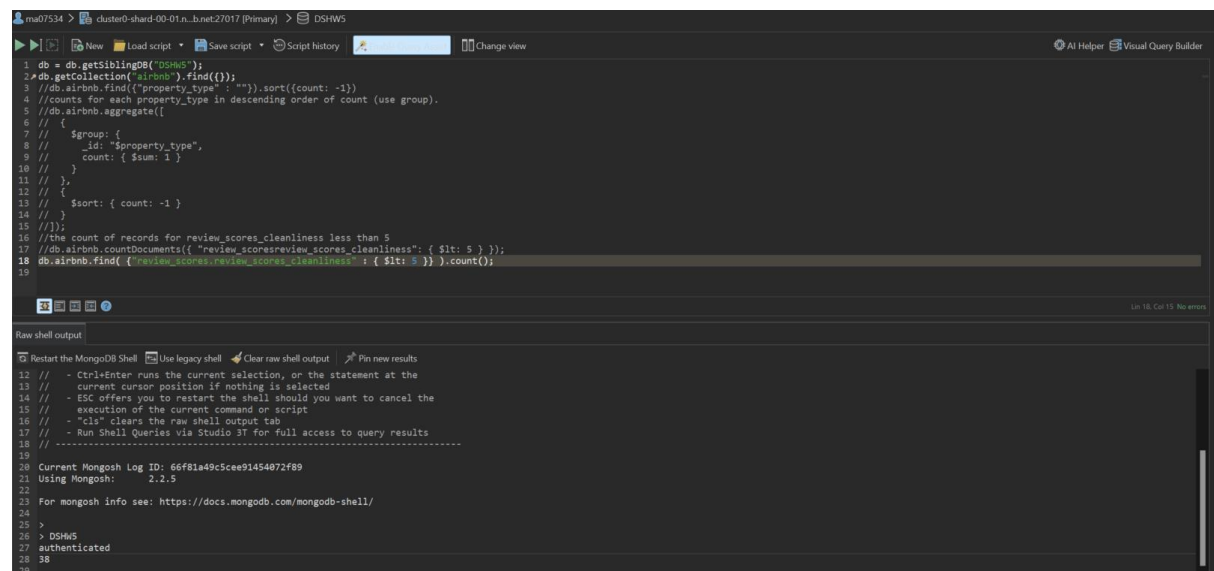
    "_id" : "Houseboat",

    "count" : 1.0
}

```

Interpretation: the query indicates that Apartment's count is the highest at 3626, the count of "House" is 606, the count of "Condominium" is 399. These are the top 3, in this long list of property type counts in descending order.

Q4. the count of records for review_scores_cleanliness less than 5.



```

1 db = db.getSiblingDB("DSHW5");
2 db.getCollection("airbnb").find({});
3 //db.airbnb.find({"property_type" : ""}).sort({count: -1})
4 //counts for each property_type in descending order of count (use group).
5 //db.airbnb.aggregate([
6 // {
7 //   $group: {
8 //     _id: "$property_type",
9 //     count: { $sum: 1 }
10 //   }
11 // },
12 // {
13 //   $sort: { count: -1 }
14 // }
15 //]);
16 //the count of records for review_scores_cleanliness less than 5
17 //db.airbnb.countDocuments({ "review_scores.review_scores_cleanliness": { $lt: 5 } });
18 db.airbnb.find( { "review_scores.review_scores_cleanliness" : { $lt: 5 } }).count();
19

```

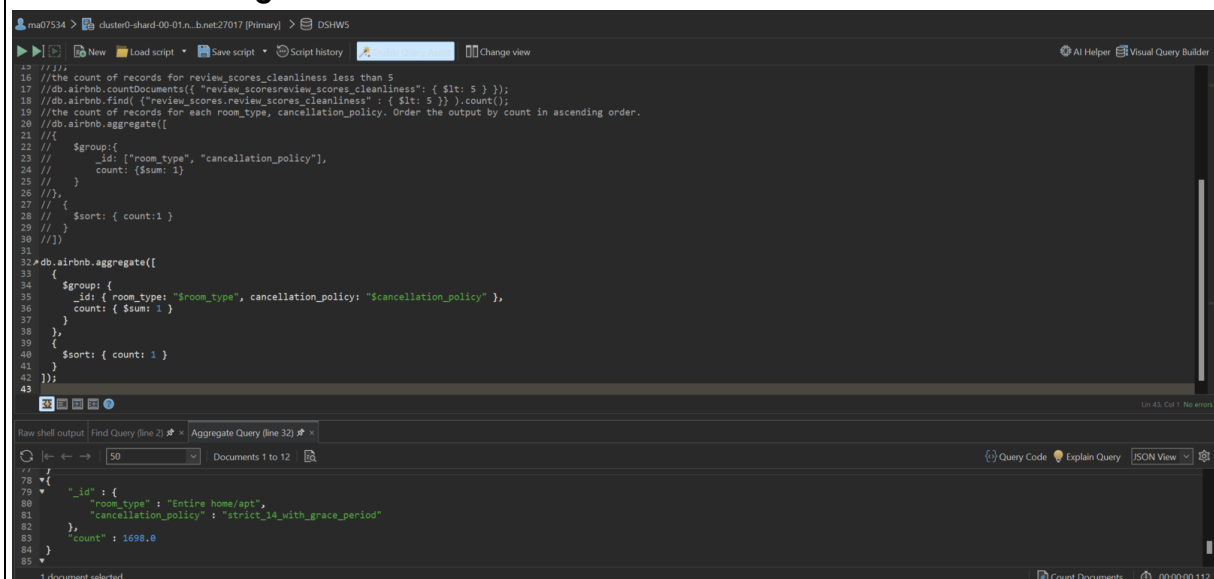
```

12 // - Ctrl+Enter runs the current selection, or the statement at the
13 //   current cursor position if nothing is selected
14 // - ESC offers you to restart the shell should you want to cancel the
15 //   execution of the current command or script
16 // - "cls" clears the raw shell output tab
17 // - Run Shell Queries via Studio 3T for full access to query results
18 // -----
19
20 Current Mongosh Log ID: 66f81a49c5cee91454072f89
21 Using Mongosh: 2.2.5
22
23 For mongosh info see: https://docs.mongodb.com/mongodb-shell/
24
25 >
26 > DSHW5
27 authenticated
28

```

Interpretation: the query indicates that the count of records for review_score_cleanliness less than 5 is 38.

Q5. the count of records for each room_type, cancellation_policy. Order the output by count in ascending order



```
12 //the count of records for review_scores_cleanliness less than 5
13 //db.airbnb.countDocuments({ "review_scores.review_scores_cleanliness": { $lt: 5 } });
14 //db.airbnb.find( { "review_scores.review_scores_cleanliness" : { $lt: 5 } } ).count();
15 //the count of records for each room_type, cancellation_policy. Order the output by count in ascending order.
16 //db.airbnb.aggregate([
17 //  {
18 //    $group: {
19 //      _id: { "room_type", "cancellation_policy" },
20 //      count: { $sum: 1 }
21 //    }
22 //  },
23 //  { $sort: { count: 1 } }
24 // ]);
25
26 db.airbnb.aggregate([
27   {
28     $group: {
29       _id: { room_type: "$room_type", cancellation_policy: "$cancellation_policy" },
30       count: { $sum: 1 }
31     }
32   },
33   { $sort: { count: 1 } }
34 ]);
```

Raw shell output: Find Query (line 2) x Aggregate Query (line 32) x

50 Documents 1 to 12

```
78 {
79   "_id" : {
80     "room_type" : "Entire home/apt",
81     "cancellation_policy" : "strict_14_with_grace_period"
82   },
83   "count" : 1998.0
84 }
85
1 document selected
```

```
{
  "_id" : {
    "room_type" : "Private room",
    "cancellation_policy" : "super_strict_60"
  },
  "count" : 6.0
}
{
  "_id" : {
    "room_type" : "Shared room",
    "cancellation_policy" : "moderate"
  },
  "count" : 8.0
}
{
  "_id" : {
    "room_type" : "Shared room",
    "cancellation_policy" : "strict_14_with_grace_period"
  },
  "count" : 24.0
}
{
  "_id" : {
    "room_type" : "Entire home/apt",
```

```
    "cancellation_policy": "super_strict_30"
  },
  "count": 38.0
}
{
  "_id": {
    "room_type": "Shared room",
    "cancellation_policy": "flexible"
  },
  "count": 51.0
}
{
  "_id": {
    "room_type": "Entire home/apt",
    "cancellation_policy": "super_strict_60"
  },
  "count": 73.0
}
{
  "_id": {
    "room_type": "Private room",
    "cancellation_policy": "moderate"
  },
  "count": 435.0
}
{
  "_id": {
    "room_type": "Private room",
    "cancellation_policy": "strict_14_with_grace_period"
  },
  "count": 698.0
}
{
  "_id": {
    "room_type": "Entire home/apt",
    "cancellation_policy": "flexible"
  },
  "count": 787.0
}
```



```
}  
{  
  "_id" : {  
    "room_type" : "Private room",  
    "cancellation_policy" : "flexible"  
  },  
  "count" : 844.0  
}  
{  
  "_id" : {  
    "room_type" : "Entire home/apt",  
    "cancellation_policy" : "moderate"  
  },  
  "count" : 893.0  
}  
{  
  "_id" : {  
    "room_type" : "Entire home/apt",  
    "cancellation_policy" : "strict_14_with_grace_period"  
  },  
  "count" : 1698.0  
}
```

Interpretation: The count of records for each room_type, cancellation_policy. Ordered the output by count in ascending order indicates that the count is between 6 and 1698.

Q6. 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.

The screenshot shows a MongoDB query editor with the following aggregate query:

```

33 // {
34 //   $group: {
35 //     _id: { room_type: "$room_type", cancellation_policy: "$cancellation_policy" },
36 //     count: { $sum: 1 }
37 //   },
38 // },
39 // {
40 //   $sort: { count: 1 }
41 // }
42 // });
43
44 db.airbnb.aggregate([
45 {
46   $match: {
47     property_type: "Apartment"
48   },
49 },
50 {
51   $group: {
52     _id: { room_type: "$room_type", cancellation_policy: "$cancellation_policy" },
53     count: { $sum: 1 }
54   },
55 },
56 {
57   $sort: { "_id.cancellation_policy": 1 }
58 }
59 ]]);
60

```

The output shows the first document selected:

```

77 {
78   "_id" : {
79     "room_type" : "Entire home/apt",
80     "cancellation_policy" : "super_strict_60"
81   },
82   "count" : 10.0
83 }
84
85

```

```

{
  "_id" : {
    "room_type" : "Private room",
    "cancellation_policy" : "flexible"
  },
  "count" : 550.0
}
{
  "_id" : {
    "room_type" : "Shared room",
    "cancellation_policy" : "flexible"
  },
  "count" : 24.0
}
{
  "_id" : {
    "room_type" : "Entire home/apt",
    "cancellation_policy" : "flexible"
  },
  "count" : 567.0
}

```

```
{
  "_id" : {
    "room_type" : "Entire home/apt",
    "cancellation_policy" : "moderate"
  },
  "count" : 631.0
}
{
  "_id" : {
    "room_type" : "Private room",
    "cancellation_policy" : "moderate"
  },
  "count" : 287.0
}
{
  "_id" : {
    "room_type" : "Shared room",
    "cancellation_policy" : "moderate"
  },
  "count" : 3.0
}
{
  "_id" : {
    "room_type" : "Entire home/apt",
    "cancellation_policy" : "strict_14_with_grace_period"
  },
  "count" : 1112.0
}
{
  "_id" : {
    "room_type" : "Private room",
    "cancellation_policy" : "strict_14_with_grace_period"
  },
  "count" : 416.0
}
{
  "_id" : {
    "room_type" : "Shared room",
```

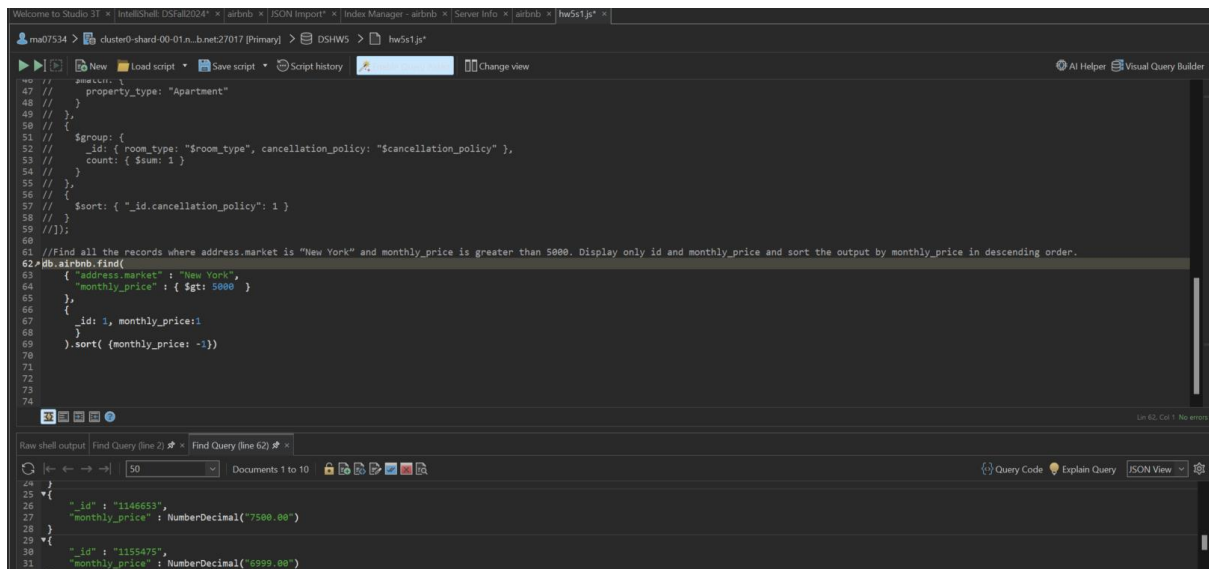
```

    "cancellation_policy": "strict_14_with_grace_period"
  },
  "count": 12.0
}
{
  "_id": {
    "room_type": "Entire home/apt",
    "cancellation_policy": "super_strict_30"
  },
  "count": 13.0
}
{
  "_id": {
    "room_type": "Private room",
    "cancellation_policy": "super_strict_60"
  },
  "count": 1.0
}
{
  "_id": {
    "room_type": "Entire home/apt",
    "cancellation_policy": "super_strict_60"
  },
  "count": 10.0
}

```

Interpretation: The count of records for each room_type, cancellation_policy when property_type is "apartment", ordered by cancellation_policy in ascending order shows room types and cancellation policies.

Q7. 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



The screenshot shows the MongoDB Studio 2.15.0 interface. The top toolbar includes buttons for 'New', 'Load script', 'Save script', 'Script history', and 'Change view'. The main editor displays a JSON document and a query. The query is as follows:

```
62 db.airbnb.find(
63   { "address.market": "New York",
64     "monthly_price": { $gt: 5000 }
65 },
66   { "_id": 1, monthly_price: 1
67   })
68   .sort( {monthly_price: -1})
69
70
71
72
73
74
```

The bottom panel shows the 'Raw shell output' with the following results:

```
24 {
25   "_id": "1146653",
26   "monthly_price": NumberDecimal("7500.00")
27 }
28 {
29   "_id": "1155475",
30   "monthly_price": NumberDecimal("6999.00")
31 }
```

```
{
  "_id": "846854",
  "monthly_price": NumberDecimal("17000.00")
}
{
  "_id": "256328",
  "monthly_price": NumberDecimal("12000.00")
}
{
  "_id": "598612",
  "monthly_price": NumberDecimal("10000.00")
}
{
  "_id": "1897001",
  "monthly_price": NumberDecimal("9495.00")
}
{
  "_id": "6064471",
  "monthly_price": NumberDecimal("8820.00")
}
{
  "_id": "640813",
```

```

    "monthly_price": NumberDecimal("8000.00")
  }
  {
    "_id": "1146653",
    "monthly_price": NumberDecimal("7500.00")
  }
  {
    "_id": "1155475",
    "monthly_price": NumberDecimal("6999.00")
  }
  {
    "_id": "2253500",
    "monthly_price": NumberDecimal("6250.00")
  }
  {
    "_id": "102995",
    "monthly_price": NumberDecimal("6000.00")
  }

```

Interpretation: Displayed are the records where address.market is "New York" and monthly_price is greater than 500, displayed by id and monthly_price and sorted by monthly_price in descending order. These prices range from 17000 to 6000.

Q8. Display the records with cleaning_fee in descending order. Display name, property_type and cleaning_fee. Limit records to 10 (easy)

The screenshot displays the MongoDB Compass application. At the top, the breadcrumb navigation shows the path: `cluster0-shard-00-01.n.b.net:27017 [Primary] > DSHWS > hw5s1js*`. The main editor area contains a MongoDB query:

```

60 //
61 //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.
62 //db.airbnb.find(
63 //  { "address.market" : "New York",
64 //    "monthly_price" : { $gt: 5000 }
65 //  },
66 //  {
67 //    _id: 1, monthly_price:1
68 //  }
69 //  ).sort( {monthly_price: -1})
70
71 //db.airbnb.find(
72 //  {},
73 //  {
74 //    name: 1,
75 //    property_type: 1,
76 //    cleaning_fee : -1
77 //  }
78 //  ).sort({cleaning_fee: -1}).limit(10);
79
80
81

```

Below the query editor, the 'Find Query (line 2)' tab is active, showing the results of the query. The results are displayed in a table with columns for the document's fields. The first document shown is:

```

7 {
8   "_id" : "5725151",
9   "name" : "service apartment with terrace",
10  "property_type" : "apartment",
11  "cleaning_fee" : NumberDecimal("1200.00")
12 }
13
14 {
15   "_id" : "6147746",
16   "name" : "Apartment in Manhattan",
17   "property_type" : "apartment",
18   "cleaning_fee" : NumberDecimal("1200.00")
19 }

```

The interface also includes a 'Law shell output' tab, a 'Find Query (line 2)' tab, and a 'Find Query (line 7)' tab. The bottom status bar indicates 'Query Code', 'Explain Query', and 'JSON View' options.

```
{
  "_id" : "13927230",
  "name" : "Casa completa p olimpíadas com serviços incluído",
  "property_type" : "House",
  "cleaning_fee" : NumberDecimal("2000.00")
}
{
  "_id" : "5725151",
  "name" : "service apartment with terrace",
  "property_type" : "Apartment",
  "cleaning_fee" : NumberDecimal("1200.00")
}
{
  "_id" : "6147746",
  "name" : "Stunning Waterfront Marina bay house in Sai Kung",
  "property_type" : "House",
  "cleaning_fee" : NumberDecimal("1000.00")
}
{
  "_id" : "28884716",
  "name" : "两房一厅,出租单人床房间(该房间不足 99 尺,房客不能带超过 2 个行李箱。另外,双人床房已有 1 人长住)",
  "property_type" : "Serviced apartment",
  "cleaning_fee" : NumberDecimal("1000.00")
}
{
  "_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 Alii",
  "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: Displayed are the records with `cleaning_fee` in descending order. We display the name, `property_type` and `cleaning_fee`, limiting the records to 10. We can see that the least cleaning fee is 800, with the highest fee at 2000.


```
{
  "_id" : {
    "cancellation_policy" : "flexible",
    "bedrooms" : NumberInt(1)
  },
  "average_price" : NumberDecimal("289.1621621621621621621621621622"),
  "total_listings" : 74.0
}
{
  "_id" : {
    "cancellation_policy" : "strict_14_with_grace_period",
    "bedrooms" : NumberInt(0)
  },
  "average_price" : NumberDecimal("310.50"),
  "total_listings" : 8.0
}
{
  "_id" : {
    "cancellation_policy" : "strict_14_with_grace_period",
    "bedrooms" : NumberInt(1)
  },
  "average_price" : NumberDecimal("310.52"),
  "total_listings" : 25.0
}
{
  "_id" : {
    "cancellation_policy" : "flexible",
    "bedrooms" : NumberInt(2)
  },
  "average_price" : NumberDecimal("358.4255319148936170212765957446809"),
  "total_listings" : 47.0
}
{
  "_id" : {
    "cancellation_policy" : "moderate",
    "bedrooms" : NumberInt(2)
  },
  "average_price" : NumberDecimal("397.5384615384615384615384615384615"),
```

```
"total_listings" : 13.0
}
{
  "_id" : {
    "cancellation_policy" : "moderate",
    "bedrooms" : NumberInt(0)
  },
  "average_price" : NumberDecimal("403.50"),
  "total_listings" : 4.0
}
{
  "_id" : {
    "cancellation_policy" : "moderate",
    "bedrooms" : NumberInt(4)
  },
  "average_price" : NumberDecimal("430.00"),
  "total_listings" : 2.0
}
{
  "_id" : {
    "cancellation_policy" : "moderate",
    "bedrooms" : NumberInt(3)
  },
  "average_price" : NumberDecimal("446.1428571428571428571428571428571"),
  "total_listings" : 7.0
}
{
  "_id" : {
    "cancellation_policy" : "strict_14_with_grace_period",
    "bedrooms" : NumberInt(2)
  },
  "average_price" : NumberDecimal("471.90909090909090909090909090909"),
  "total_listings" : 22.0
}
{
  "_id" : {
    "cancellation_policy" : "strict_14_with_grace_period",
    "bedrooms" : NumberInt(3)
```

```
    },
    "average_price" : NumberDecimal("556.1428571428571428571428571428571"),
    "total_listings" : 7.0
  }
{
  "_id" : {
    "cancellation_policy" : "flexible",
    "bedrooms" : NumberInt(3)
  },
  "average_price" : NumberDecimal("654.30"),
  "total_listings" : 10.0
}
{
  "_id" : {
    "cancellation_policy" : "strict_14_with_grace_period",
    "bedrooms" : NumberInt(4)
  },
  "average_price" : NumberDecimal("914.50"),
  "total_listings" : 2.0
}
{
  "_id" : {
    "cancellation_policy" : "flexible",
    "bedrooms" : NumberInt(5)
  },
  "average_price" : NumberDecimal("1055.00"),
  "total_listings" : 1.0
}
{
  "_id" : {
    "cancellation_policy" : "super_strict_30",
    "bedrooms" : NumberInt(3)
  },
  "average_price" : NumberDecimal("1698.00"),
  "total_listings" : 1.0
}
{
  "_id" : {
```

```

      "cancellation_policy": "flexible",
      "bedrooms": NumberInt(10)
    },
    "average_price": NumberDecimal("2083.00"),
    "total_listings": 1.0
  }
  {
    "_id": {
      "cancellation_policy": "moderate",
      "bedrooms": NumberInt(8)
    },
    "average_price": NumberDecimal("2974.00"),
    "total_listings": 1.0
  }
  {
    "_id": {
      "cancellation_policy": "moderate",
      "bedrooms": NumberInt(7)
    },
    "average_price": NumberDecimal("4745.00"),
    "total_listings": 1.0
  }

```

This query filters Airbnb listings in Istanbul to focus on entire homes or apartments. It ensures that the listings have a cancellation policy and a specified number of bedrooms. Then, it groups the results by both the cancellation policy and the number of bedrooms, calculating the average price for each group and counting the total number of listings. Finally, the results are sorted by average price in ascending order, allowing us to see how prices vary based on different cancellation policies and bedroom counts in Istanbul.

We can see that the listings range from a price of 4745 to 246, and we can see the varying cancellation policies as well.

Q10. Come up with your own query to show any interesting insight. Use atleast two fields for group.

```
104 db.airbnb.aggregate([
105   {
106     $match: {
107       "amenities": { $exists: true, $ne: [] },
108       "price": { $exists: true }
109     }
110   },
111   {
112     $unwind: "$amenities"
113   },
114   {
115     $group: {
116       _id: "$amenities",
117       average_price: { $avg: { $toDouble: "$price" } },
118       total_listings: { $sum: 1 }
119     }
120   },
121   {
122     $sort: { average_price: -1 }
123   },
124   {
125     $limit: 10
126   }
127 ]]);
128
```

Raw shell output Find Query (line 2) Aggregate Query (line 104)

50 Documents 1 to 10

Query Code Explain Query JSON View

```
1 *{
2   "_id" : "Home theater",
3   "average_price" : NumberDecimal("2356.00"),
4   "total_listings" : 1.0
5 }
6 *{
7   "_id" : "Ice Machine",
8   "average_price" : NumberDecimal("2356.00"),
9   "total_listings" : 1.0
10 }
```

```
{
  "_id" : "Home theater",
  "average_price" : NumberDecimal("2356.00"),
  "total_listings" : 1.0
}
{
  "_id" : "Ice Machine",
  "average_price" : NumberDecimal("2356.00"),
  "total_listings" : 1.0
}
{
  "_id" : "Parking",
  "average_price" : NumberDecimal("2356.00"),
  "total_listings" : 1.0
}
{
  "_id" : "Dining area",
  "average_price" : NumberDecimal("2356.00"),
  "total_listings" : 1.0
}
{
  "_id" : "Sonos sound system",
```

```

    "average_price" : NumberDecimal("2356.00"),
    "total_listings" : 1.0
  }
  {
    "_id" : "Ironing Board",
    "average_price" : NumberDecimal("2356.00"),
    "total_listings" : 1.0
  }
  {
    "_id" : "Chef's kitchen",
    "average_price" : NumberDecimal("2356.00"),
    "total_listings" : 1.0
  }
  {
    "_id" : "Breakfast bar",
    "average_price" : NumberDecimal("2356.00"),
    "total_listings" : 1.0
  }
  {
    "_id" : "Permit parking",
    "average_price" : NumberDecimal("2356.00"),
    "total_listings" : 1.0
  }
  {
    "_id" : "Alfresco shower",
    "average_price" : NumberDecimal("2356.00"),
    "total_listings" : 1.0
  }
}

```

The query begins by filtering listings to include only those that have non-empty amenities and an existing price field. It then unwinds the amenities array, creating separate documents for each amenity to allow for detailed analysis. After that, the results are grouped by individual amenities, calculating the average price and counting how many listings feature each one. The results are sorted in descending order by average price to highlight the most valuable amenities. Finally, the output is limited to the top 10 amenities based on average price. This query provides insight into how specific amenities impact listing prices, helping hosts determine which features might enhance their pricing strategy.

We can see that the top 10 amenities all nearly have the same price at 2356, and all of these can enhance their pricing strategy.