

CS/CE 457/464 - Homework Assignment 5: NoSQL

Breeha Qasim

bq08283@st.habib.edu.pk

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

```
14 db.airbnb_listings.count()
Raw shell output
Restart the MongoDB Shell Use legacy shell Clear raw shell output Pin new results
31 0
32 0
33 hw5.airbnb_listings
34 5555
```

2. the count of records for property_type "House"

```
15 db.airbnb_listings.countDocuments({ property_type: "House" })
Raw shell output
Restart the MongoDB Shell Use legacy shell Clear raw shell output Pin new results
39 606
```

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

```
16 db.airbnb_listings.aggregate([{$group: { _id: "$property_type", count: { $sum: 1 } }},{ $sort: { count: -1 } }])
Raw shell output Aggregate Query (line 16)
Documents 1 to 36
airbnb_listings > count
_id count
Apartment 3626.0
House 606.0
Condominium 399.0
Serviced apartm... 185.0
Loft 142.0
Townhouse 108.0
Guest suite 81.0
Bed and breakfast 69.0
```

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

```
17 db.airbnb_listings.countDocuments({ review_scores_cleanliness: { $lt: 5 } })
Raw shell output
Restart the MongoDB Shell Use legacy shell Clear raw shell output Pin new results
27 authenticated
28 0
```

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

```
16 db.airbnb_listings.aggregate([
17   { $group: { _id: { room_type: "$room_type", cancellation_policy: "$cancellation_policy" }, count: { $sum: 1 } } },
18   { $sort: { count: 1 } }
19 ])
```

Raw shell output | Aggregate Query (line 16) ✕

50 Documents 1 to 12

Query Code Explain Query Table View

airbnb_listings > count

_id	count
{ "room_type": "Entire home/apt", "cancellation_policy": "Flexible" }	6.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Moderate" }	8.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Strict" }	24.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Super Strict" }	38.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Super Strict (30-day)" }	51.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Super Strict (60-day)" }	73.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Super Strict (14-day)" }	435.0

1 document selected

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.

```
21 db.airbnb_listings.aggregate([
22   { $match: { property_type: "Apartment" } },
23   { $group: { _id: { room_type: "$room_type", cancellation_policy: "$cancellation_policy" }, count: { $sum: 1 } } },
24   { $sort: { "_id.cancellation_policy": 1 } }
25 ])
```

Raw shell output | Aggregate Query (line 21) ✕

50 Documents 1 to 12

Query Code Explain Query Table View

airbnb_listings > count

_id	count
{ "room_type": "Entire home/apt", "cancellation_policy": "Flexible" }	24.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Moderate" }	550.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Strict" }	567.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Super Strict" }	631.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Super Strict (30-day)" }	3.0
{ "room_type": "Entire home/apt", "cancellation_policy": "Super Strict (60-day)" }	207.0

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.

```
27 db.airbnb_listings.find(
28   { "address.market": "New York", monthly_price: { $gt: 5000 } },
29   { _id: 1, monthly_price: 1 }).sort({ monthly_price: -1 })
30
```

Raw shell output Find Query (line 27) ✖

50 Documents 1 to 10

airbnb_listings > monthly_price

_id	monthly_price
846854	17000.0
256328	12000.0
598612	10000.0
1897001	9495.0
6064471	8820.0
400013	8000.0

1 document selected

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

```
31 db.airbnb_listings.find({}, { name: 1, property_type: 1, cleaning_fee: 1 }).sort({ cleaning_fee: -1 }).limit(10)
32
```

Raw shell output Find Query (line 31) ✖

50 Documents 1 to 10

airbnb_listings > name

_id	name	property_type	cleaning_fee
13927230	Casa completa ...	House	2000.0
5725151	service apartm...	Apartment	1200.0
6147746	Stunning Waterf...	House	1000.0
28884716	两房一厅,出租...	Serviced apartm...	1000.0
20362690	WORLD CLASS ...	Apartment	942.0
1176693	BEST REVIEWS*...	Apartment	942.0
15488401	INSTAGRAM H...	Apartment	942.0
46215566	Kabala Aki...	House	910.0

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

```
35 db.airbnb_listings.aggregate([
36   { $group: { _id: "$address.suburb", average_cleanliness: { $avg: "$review_scores.review_scores_cleanliness" },
37     average_rating: { $avg: "$review_scores.review_scores_rating" } } }, { $sort: { average_cleanliness: -1, average_rating: -1 } }
38 ])
39
```

Raw shell output | Aggregate Query (line 35) ✖

50 Documents 1 to 50

airbnb_listings > average_cleanliness

_id	average_cleanliness	average_rating
Camperdown	10.0	100.0
Deaconsfield	10.0	100.0
Ridgewood	10.0	100.0
Porta	10.0	100.0
Artarmon	10.0	100.0
Little Burgundy	10.0	100.0

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

```
41 db.airbnb_listings.aggregate([
42   { $group: {
43     _id: { bedrooms: "$bedrooms", property_type: "$property_type" },
44     average_price: { $avg: "$price" },
45     max_price: { $max: "$price" },
46     min_price: { $min: "$price" } } }, { $sort: { "_id.bedrooms": 1, "_id.property_type": 1 } }
47 ])
48
```

Raw shell output | Aggregate Query (line 41) ✖

50 Documents 1 to 50

airbnb_listings > average_price

_id	average_price	max_price	min_price
{ 1 fields }	244.666666666...	105.0	70.0
{ 1 fields }	50.0	50.0	50.0
{ 1 fields }	79.0	79.0	79.0
{ 2 fields }	200.0	200.0	200.0
{ 2 fields }	252.754098360...	2238.0	20.0