POLITECNICO
MILANO 1863

MongoDB - Exercise Session

Andrea Tocchetti
andrea.tocchetti@polimi.it

MongoDB - Exercise Session

Write one or more queries to return a single housing, and each single one of their attributes that are sub-documents or array. Inspect their content to understand their structure.

```
db.air_bnb_collection.findOne()
db.air_bnb_collection.findOne({ }, {amenities: 1})
db.air_bnb_collection.findOne({ }, {images: 1})
db.air_bnb_collection.findOne({ }, {host: 1})
db.air_bnb_collection.findOne({ }, {address: 1})
db.air_bnb_collection.findOne({ }, {availability: 1})
db.air_bnb_collection.findOne({ }, {review_scores: 1})
db.air_bnb_collection.findOne({ }, {reviews: 1})
```

MongoDB - Exercise Session

Write a query to return all the housings whose minimum nights is equal to 2.

```
db.air_bnb_collection.find({"minimum_nights": {"$eq": "2"} },
                           {"minimum_nights": 1})*
```

* I return minimum_nights just to check that the answer is correct. I'll do the same in other queries to properly visualize the result.

MongoDB - Exercise Session

Write a query to return the number of housings whose number of reviews is greater than 10.

```
db.air_bnb_collection.find({"number_of_reviews": {"$gt": 10} },
                           {"number_of_reviews": 1})
                .sort({"number_of_reviews": 1})
```

MongoDB - Exercise Session

Write a query to return the top 10 housings based on their number of bedrooms.

```
db.air_bnb_collection.find({ },
                           {"bedrooms": 1})
                     .sort({"bedrooms": -1})
                     .limit(10)
```

MongoDB - Exercise Session

Write a query to return the housings whose host response rate is greater than 90.

```
db.air_bnb_collection.find({"host.host_response_rate": {"$gt": 90} }
                           {"host.host_response_rate": 1"})
```

MongoDB - Exercise Session

Write a query to return the name of the hosts whose housings includes a TV and Iron.

```
db.air_bnb_collection.find({"$and": [
                                    {"amenities": "TV"},
                                    {"amenities": "Iron"}
                                ],
                                {"name": 1, "amenities": 1})
```

MongoDB - Exercise Session

Write a query to return the name of the hosts whose housings includes a TV or Iron.

```
db.air_bnb_collection.find({"$or": [
                                    {"amenities": "TV"},
                                    {"amenities": "Iron"}
                                ],
                                {"name": 1, "amenities": 1})
```

MongoDB - Exercise Session

Write a query to count the number of housings based on their price.

```
db.air_bnb_collection.aggregate(["$group": {
                                    "_id": "$price",
                                    "housing_count": {"$sum": 1}
                                }])
```

MongoDB - Exercise Session

Write a query to compute the total number of bedrooms available across all housings.

```
db.air_bnb_collection.aggregate(["$group": {
                                    "_id": true,
                                    "bedroom_count": {"$sum": "$bedroom"}
                                }])
```

MongoDB - Exercise Session

Write a query to compute the total number of housings based on their cancellation policy, whose accommodates is greater than 6 and whose host has a profile picture.

```
db.air_bnb_collection.aggregate([{"$match": {"accommodates": {"$gt": 6}} },
                                 {"$match": {"host.host_has_profile_pic": true} },
                                 {"$group": {
                                       "_id": "$cancellation_policy",
                                       "cancellation_policy_count": {"$sum": 1}
                                 }}])
```

MongoDB - Exercise Session

Write a query to compute the total number of housings for each amenity.

```
db.air_bnb_collection.aggregate([{"$unwind": {"path": "$amenities"} },
                                 {"$group": {
                                      "_id": "$amenities",
                                      "housing_with_amenity_count": {"$sum": 1}
                                 }}])
```

MongoDB - Exercise Session

Write a query to compute the total number of housings for each amenity, returning only the amenities whose count is greater than 15.

```
db.air_bnb_collection.aggregate([{"$unwind": {"path": "$amenities"} },
                                 {"$group": {
                                       "_id": "$amenities",
                                       "housing_with_amenity_count": {"$sum": 1}
                                 }},
                                 {"$match": {"housing_with_amenity_count": {"$gt": 15}} },
                                ])
```

MongoDB - Exercise Session

Write a query to compute the average price and the average cleaning fees of the housings based on their number of bedrooms and bathrooms, returning only those whose average price is greater than 60 and average cleaning fees is greater than 40.

```
db.air_bnb_collection.aggregate([{"$group": {
        "_id": {"n_bed": "$bedrooms", "n_bath": "$bathrooms"},
        "avg_price": {"$avg": "$price"},
        "avg_cleaning_fees": {"$avg": "$cleaning_fee"}
    }},
    {"$match": {"avg_price": {"$gt": 60}} },
    {"$match": {"avg_cleaning_fees": {"$gt": 40}} },
])
```

MongoDB - Exercise Session

Write a query to return the housings for which at least one review with listing id 10006546 and reviewer id 11263097 was written.

```
db.air_bnb_collection.aggregate([{"$match": {"reviews": {
                                        "$elemMatch": {
                                                "listing_id": "10006546",
                                                "reviewer_id": "11263097"
                                        }
                                }}
}])
```

MongoDB - Exercise Session

Write a query to return the housings for which at least one review with listing id 10006546 and one with reviewer name Bridget were written.

```
db.air_bnb_collection.aggregate([{"$match": {"reviews.listing_id": "10006546"} },
                                 {"$match": {"reviews.reviewer_name": "Bridget"} },
                                ])
```

MongoDB - Exercise Session

For each country, compute the total number of unique hosts.

```
db.air_bnb_collection.aggregate([{"$group": {
        "_id": {"country": "$address.country", "host_id": "$host.host_id"},
        "duplication_count_per_host": {"$sum": 1}
    }},
    {"$group": {
        "_id": "$_id.country",
        "unique_host_count": {"$sum": 1}
    }},
])
```


ANY
Questions?