EXERCISES

MongoDB query

EXERCISE 1 – PROPERTY LISTING

A) Select all properties that have wifi and dryer as amenities sorted by increasing price. Display only the name, url and price.

SOLUTION

```
db.listingsAndReviews.find(
      { amenities:
            {$all: ["Wifi", "Dryer"] }
      },
{_id:0, name: 1, price:1, url: 1}
).sort( {price:1} )
```

```
{
  "_id": "10006546",
  "url": "https://www.airbnb.com/rooms/10006546",
  "name": "Ribeira Charming Duplex",
  "description": "Fantastic duplex apartment with . . .",
  "property_type": "House",
  "minimum_nights": 2,
  "maximum_nights": 30,
  "beds": 5,
  "number_of_reviews": 51,
  "amenities": [
    "TV",
    "Wifi",
    "Smoking allowed",
    "Pets allowed",
    "Dryer",
    "Heating"
  ],
  "price": 80,
  "host": {
    "host_id": "51399391",
    "name": "Ana&Gonçalo",
    "city": "Porto",
    "country": "Portugal"
  }
}
```

EXERCISE 1 – PROPERTY LISTING

B) For all House type properties located in the city of Turin, set the minimum number of nights to 3. If the minimum number of nights is greater than 3, it should not be updated.

SOLUTION

```
db.listingsAndReviews.updateMany(
 {"property_type": "House",
  "host.city": "Turin",
  "minimum_nights": {lt: 3 }
 },
 {$set: {"minimum_nights": 3}}
)
```

```
{
  "_id": "10006546",
  "url": "https://www.airbnb.com/rooms/10006546",
  "name": "Ribeira Charming Duplex",
  "description": "Fantastic duplex apartment with . . .",
  "property_type": "House",
  "minimum_nights": 2,
  "maximum_nights": 30,
  "beds": 5,
  "number_of_reviews": 51,
  "amenities": [
    "TV",
    "Wifi",
    "Smoking allowed",
    "Pets allowed",
    "Dryer",
    "Heating"
  ],
  "price": 80,
  "host": {
    "host_id": "51399391",
    "name": "Ana&Gonçalo",
    "city": "Porto",
    "country": "Portugal"
  }
}
```

EXERCISE 1 – PROPERTY LISTING

C) Calculate the median price of all House type properties.

```
{
  "_id": "10006546",
  "url": "https://www.airbnb.com/rooms/10006546",
  "name": "Ribeira Charming Duplex",
  "description": "Fantastic duplex apartment with . . .",
  "property_type": "House",
  "minimum_nights": 2,
  "maximum_nights": 30,
  "beds": 5,
  "number_of_reviews": 51,
  "amenities": [
    "TV",
    "Wifi",
    "Smoking allowed",
    "Pets allowed",
    "Dryer",
    "Heating"
  ],
  "price": 80,
  "host": {
    "host_id": "51399391",
    "name": "Ana&Gonçalo",
    "city": "Porto",
    "country": "Portugal"
  }
}
```

SOLUTION

```
db.listingsAndReviews.aggregate(
      [
{     $match: {property_type: "House"}},
{     $sort: {price: 1}},
{     $group: {
            '_id': null,
            'value': {'$push': '$price'}}},
{     $project: {
            _id: 1,
            "median": {$arrayElemAt:["$value", {$floor:
{$multiply: [0.50,{$size: "$value"} ]}} ]}}}
]
)
```

EXERCISE 2 – EXAMS

A) For each type of test, calculate the minimum, maximum, and average grades taken by students. Display this information only for the types with more than 50 tests taken.

SOLUTION

```
db.grades.aggregate(
[{$unwind: {path: '$scores'}},
{$group: {
  _id: '$scores.type',
  avg: {$avg: '$scores.score'},
  min: {$min: '$scores.score'},
  max: {$max: '$scores.score'},
  count: {$sum: 1}}},
{$match: { count: {$gt:50}}}]
)
```

```
{
  "_id": "56d5f7eb604eb380b0d8d8d2",
  "student_id": 0,
  "scores": [
    {
      "type": "exam",
      "score": 41.25131199553351
    },
    {
      "type": "quiz",
      "score": 91.7351500084582
    },
    {
      "type": "homework",
      "score": 24.198828271948415
    },
    {
      "type": "homework",
      "score": 79.77471812670814
    }
  ],
  "class_id": 391
}
```

EXERCISE 2 – EXAMS

B) Considering only the sufficient scores (grade equal to or higher than 60) of the exam type, calculate for each class the number of exams taken and the average of the grades obtained.

SOLUTION

```
db.grades.aggregate(
[{$unwind: {
  path: "$scores",
  }},
{$match: {
  "scores.type":"exam",
  "scores.score": {$gte: 60}
}},
{$group: {
  _id: "$class_id",
  c: {$sum:1},
  avg: {$avg:"$scores.score"}
}}]
```

```
{
  "_id": "56d5f7eb604eb380b0d8d8d2",
  "student_id": 0,
  "scores": [
    {
      "type": "exam",
      "score": 41.25131199553351
    },
    {
      "type": "quiz",
      "score": 91.7351500084582
    },
    {
      "type": "homework",
      "score": 24.198828271948415
    },
    {
      "type": "homework",
      "score": 79.77471812670814
    }
  ],
  "class_id": 391
}
```

EXERCISE 2 – EXAMS

C) Enter the certificate grade of 85 for the student with identifier of 1000 enrolled in grade 150. Also set a flag named "completed".

SOLUTION

```
db.grades.updateOne({ "student_id": 1000,
class_id:150},
{$push: {scores: {"type": "certificate", "grade": 85
}}, $set: {completed: true} })
```

```
{
  "_id": "56d5f7eb604eb380b0d8d8d2",
  "student_id": 0,
  "scores": [
    {
      "type": "exam",
      "score": 41.25131199553351
    },
    {
      "type": "quiz",
      "score": 91.7351500084582
    },
    {
      "type": "homework",
      "score": 24.198828271948415
    },
    {
      "type": "homework",
      "score": 79.77471812670814
    }
  ],
  "class_id": 391
}
```