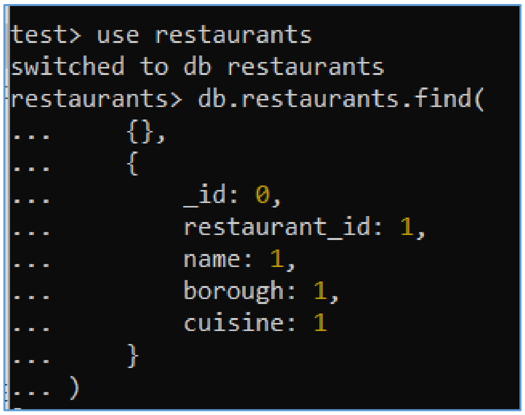
Лабораторная работа №8

Задание 1

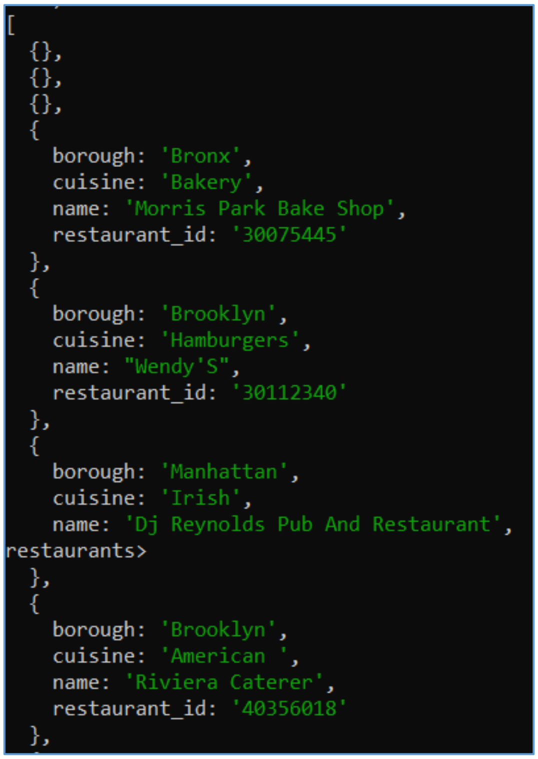
Выполните импорт коллекции из файла restaurants.json

Выполните запросы :

1. Выведите все документы коллекции Ресторан в формате: ***restaurant_id***, ***name, borough*** и ***cuisine***, вывод ***_id*** для всех документов исключить.

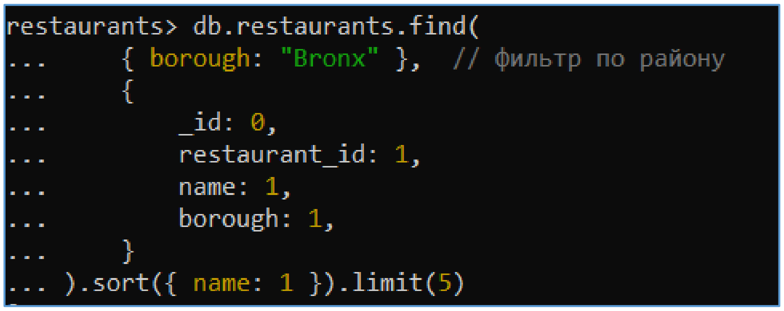


```
test> use restaurants
switched to db restaurants
restaurants> db.restaurants.find(
...     {},
...     {
...         _id: 0,
...         restaurant_id: 1,
...         name: 1,
...         borough: 1,
...         cuisine: 1
...     }
... )
```

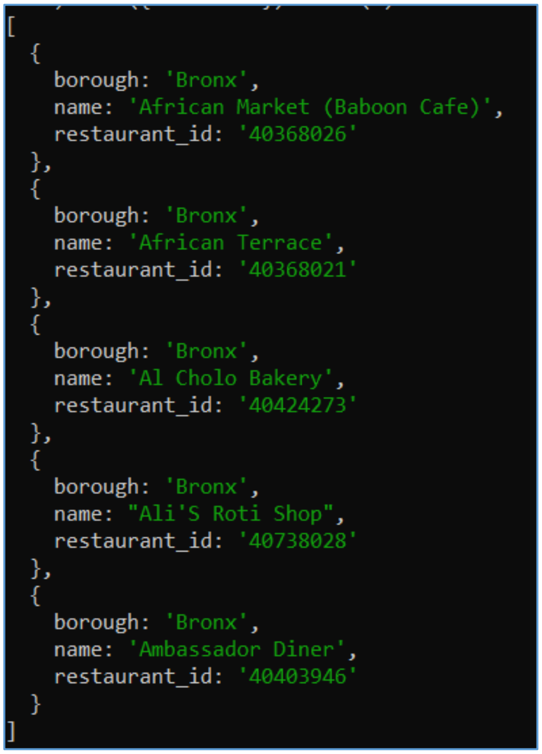


```
[
  {},
  {},
  {},
  {
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
restaurants>
  },
  {
    borough: 'Brooklyn',
    cuisine: 'American  ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
```

2. Выведите первые 5 ресторанов в алфавитном порядке, которые находятся в районе ***Bronx***.

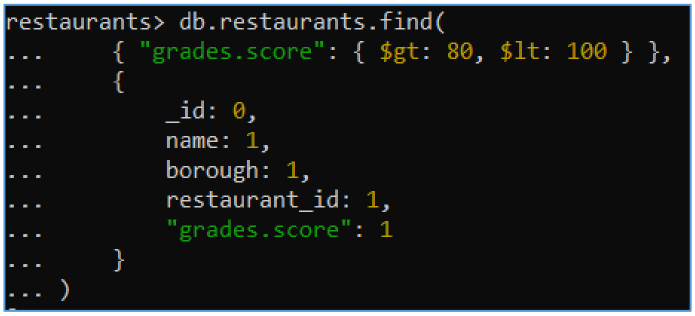


```
restaurants> db.restaurants.find(
...     { borough: "Bronx" },  // фильтр по району
...     {
...         _id: 0,
...         restaurant_id: 1,
...         name: 1,
...         borough: 1,
...     }
... ).sort({ name: 1 }).limit(5)
```

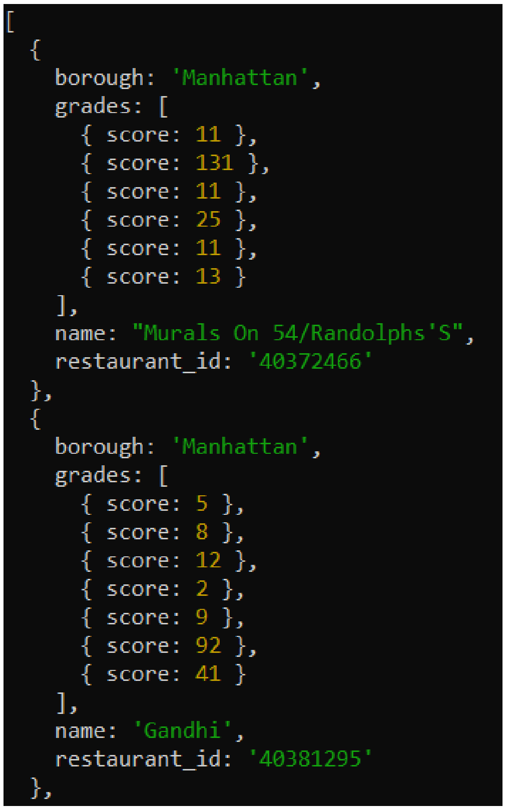


```
[
  {
    borough: 'Bronx',
    name: 'African Market (Baboon Cafe)',
    restaurant_id: '40368026'
  },
  {
    borough: 'Bronx',
    name: 'African Terrace',
    restaurant_id: '40368021'
  },
  {
    borough: 'Bronx',
    name: 'Al Cholo Bakery',
    restaurant_id: '40424273'
  },
  {
    borough: 'Bronx',
    name: "Ali'S Roti Shop",
    restaurant_id: '40738028'
  },
  {
    borough: 'Bronx',
    name: 'Ambassador Diner',
    restaurant_id: '40403946'
  }
]
```

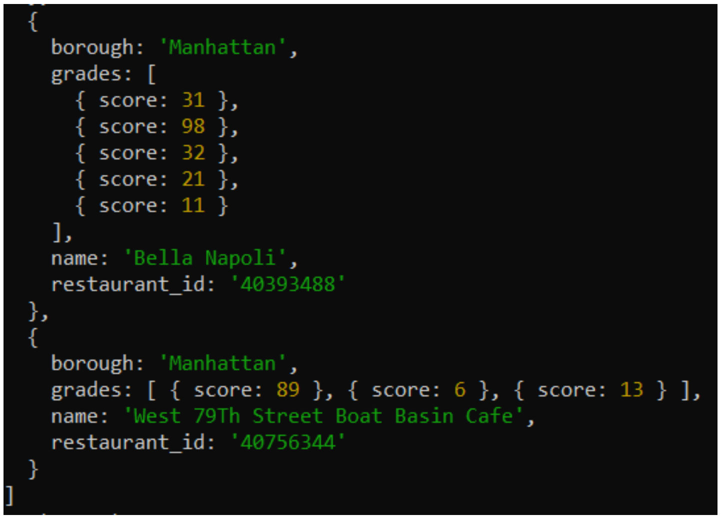
3. Найдите рестораны, которые набрали более 80, но менее 100 баллов.



```
restaurants> db.restaurants.find(
...     { "grades.score": { $gt: 80, $lt: 100 } },
...     {
...         _id: 0,
...         name: 1,
...         borough: 1,
...         restaurant_id: 1,
...         "grades.score": 1
...     }
... )
```

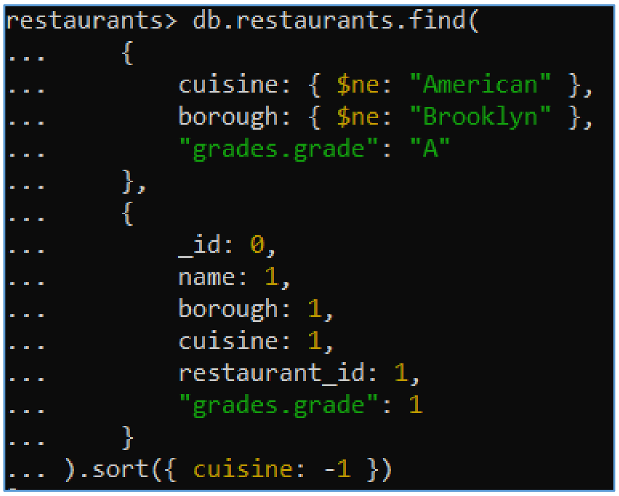


```
[
  {
    borough: 'Manhattan',
    grades: [
      { score: 11 },
      { score: 131 },
      { score: 11 },
      { score: 25 },
      { score: 11 },
      { score: 13 }
    ],
    name: "Murals On 54/Randolphs'S",
    restaurant_id: '40372466'
  },
  {
    borough: 'Manhattan',
    grades: [
      { score: 5 },
      { score: 8 },
      { score: 12 },
      { score: 2 },
      { score: 9 },
      { score: 92 },
      { score: 41 }
    ],
    name: 'Gandhi',
    restaurant_id: '40381295'
  },
```

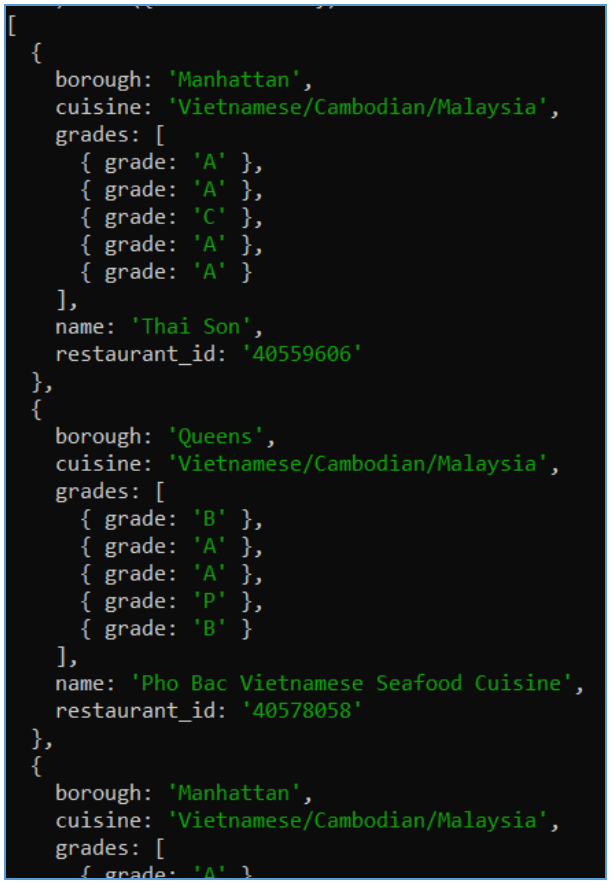


```
  {
    borough: 'Manhattan',
    grades: [
      { score: 31 },
      { score: 98 },
      { score: 32 },
      { score: 21 },
      { score: 11 }
    ],
    name: 'Bella Napoli',
    restaurant_id: '40393488'
  },
  {
    borough: 'Manhattan',
    grades: [ { score: 89 }, { score: 6 }, { score: 13 } ],
    name: 'West 79Th Street Boat Basin Cafe',
    restaurant_id: '40756344'
  }
]
```

4. Найдите рестораны, которые не относятся к типу кухни ***American***, получили оценку «***A***», не расположены в районе ***Brooklyn***. Документ должен отображаться в соответствии с кухней в порядке убывания.

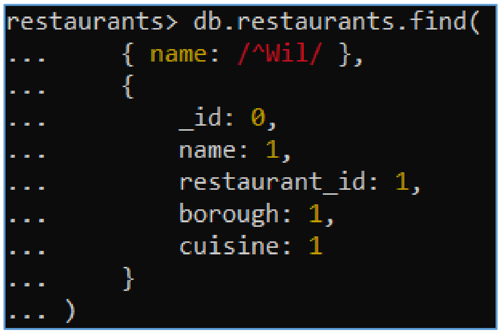


```
restaurants> db.restaurants.find(
...     {
...         cuisine: { $ne: "American" },
...         borough: { $ne: "Brooklyn" },
...         "grades.grade": "A"
...     },
...     {
...         _id: 0,
...         name: 1,
...         borough: 1,
...         cuisine: 1,
...         restaurant_id: 1,
...         "grades.grade": 1
...     }
... ).sort({ cuisine: -1 })
```

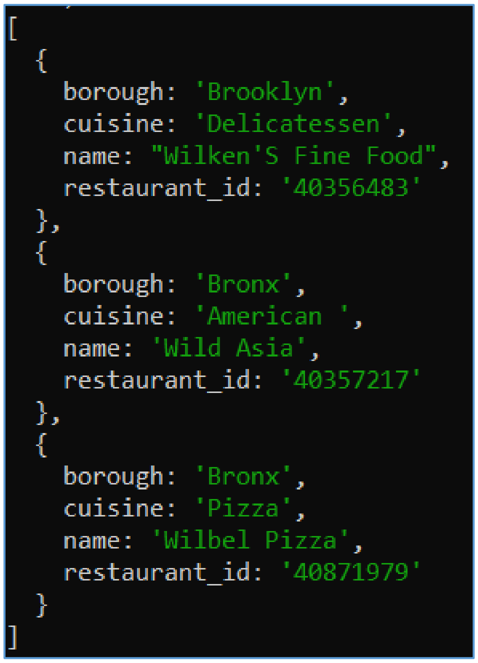


```
[
  {
    borough: 'Manhattan',
    cuisine: 'Vietnamese/Cambodian/Malaysia',
    grades: [
      { grade: 'A' },
      { grade: 'A' },
      { grade: 'C' },
      { grade: 'A' },
      { grade: 'A' }
    ],
    name: 'Thai Son',
    restaurant_id: '40559606'
  },
  {
    borough: 'Queens',
    cuisine: 'Vietnamese/Cambodian/Malaysia',
    grades: [
      { grade: 'B' },
      { grade: 'A' },
      { grade: 'A' },
      { grade: 'P' },
      { grade: 'B' }
    ],
    name: 'Pho Bac Vietnamese Seafood Cuisine',
    restaurant_id: '40578058'
  },
  {
    borough: 'Manhattan',
    cuisine: 'Vietnamese/Cambodian/Malaysia',
    grades: [
      { grade: 'A' }
```

5. Найдите идентификатор ресторана, название, район и кухню для тех ресторанов, чье название начинается с первых трех букв «***Wil***»

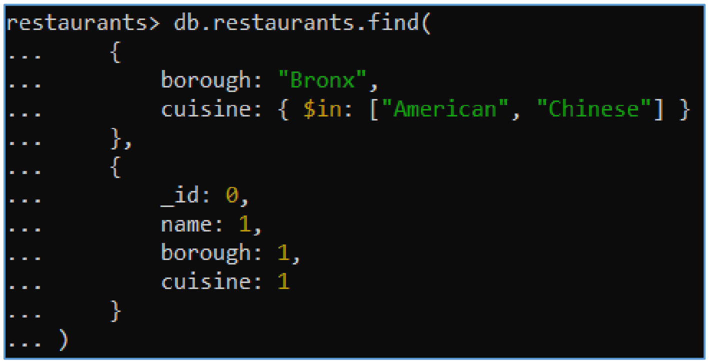


```
restaurants> db.restaurants.find(
...     { name: /^Wil/ },
...     {
...         _id: 0,
...         name: 1,
...         restaurant_id: 1,
...         borough: 1,
...         cuisine: 1
...     }
... )
```

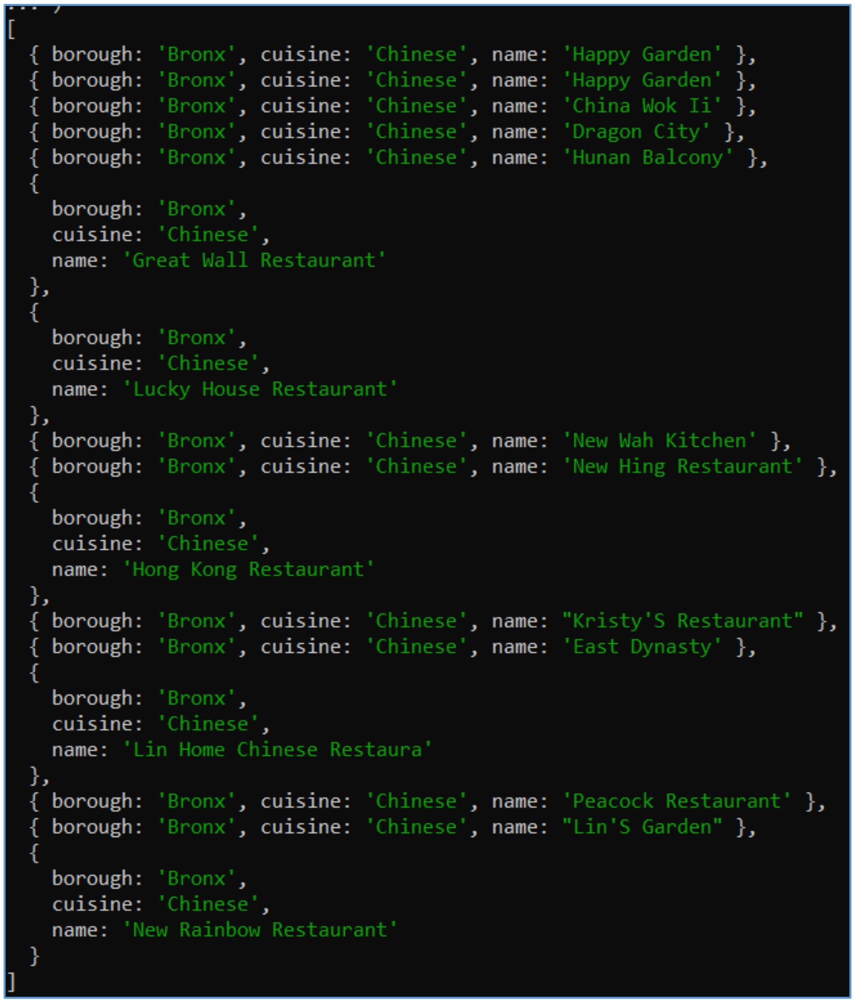


```
[
  {
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: "Wilken'S Fine Food",
    restaurant_id: '40356483'
  },
  {
    borough: 'Bronx',
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    borough: 'Bronx',
    cuisine: 'Pizza',
    name: 'Wilbel Pizza',
    restaurant_id: '40871979'
  }
]
```

6. Найдите рестораны, которые относятся к району ***Bronx*** и готовят ***American*** или ***Chinese*** блюда.

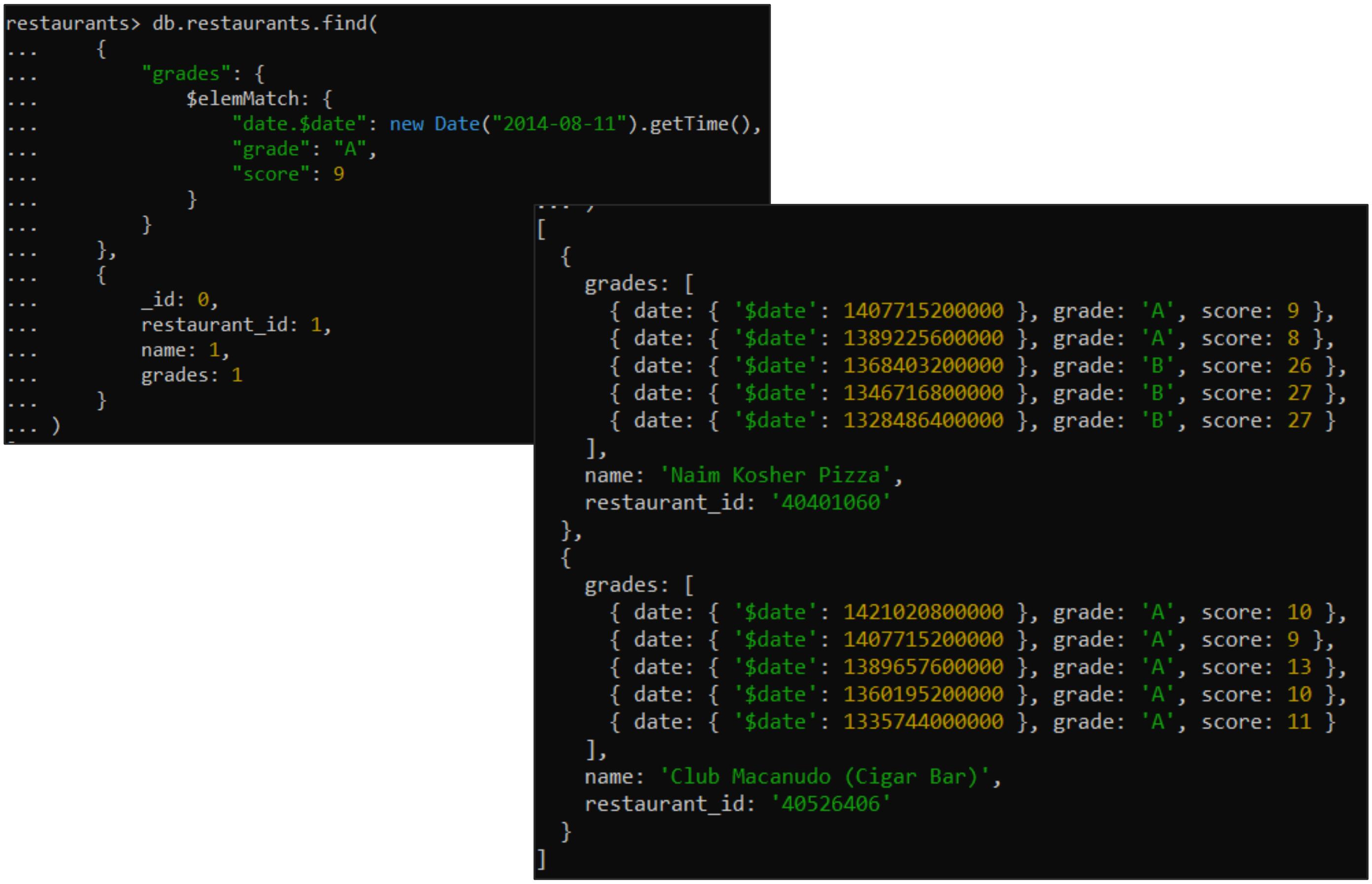


```
restaurants> db.restaurants.find(
...     {
...         borough: "Bronx",
...         cuisine: { $in: ["American", "Chinese"] }
...     },
...     {
...         _id: 0,
...         name: 1,
...         borough: 1,
...         cuisine: 1
...     }
... )
```



```
... )
[
  { borough: 'Bronx', cuisine: 'Chinese', name: 'Happy Garden' },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'Happy Garden' },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'China Wok Ii' },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'Dragon City' },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'Hunan Balcony' },
  {
    borough: 'Bronx',
    cuisine: 'Chinese',
    name: 'Great Wall Restaurant'
  },
  {
    borough: 'Bronx',
    cuisine: 'Chinese',
    name: 'Lucky House Restaurant'
  },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'New Wah Kitchen' },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'New Hing Restaurant' },
  {
    borough: 'Bronx',
    cuisine: 'Chinese',
    name: 'Hong Kong Restaurant'
  },
  { borough: 'Bronx', cuisine: 'Chinese', name: "Kristy'S Restaurant" },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'East Dynasty' },
  {
    borough: 'Bronx',
    cuisine: 'Chinese',
    name: 'Lin Home Chinese Restaura'
  },
  { borough: 'Bronx', cuisine: 'Chinese', name: 'Peacock Restaurant' },
  { borough: 'Bronx', cuisine: 'Chinese', name: "Lin'S Garden" },
  {
    borough: 'Bronx',
    cuisine: 'Chinese',
    name: 'New Rainbow Restaurant'
  }
]
```

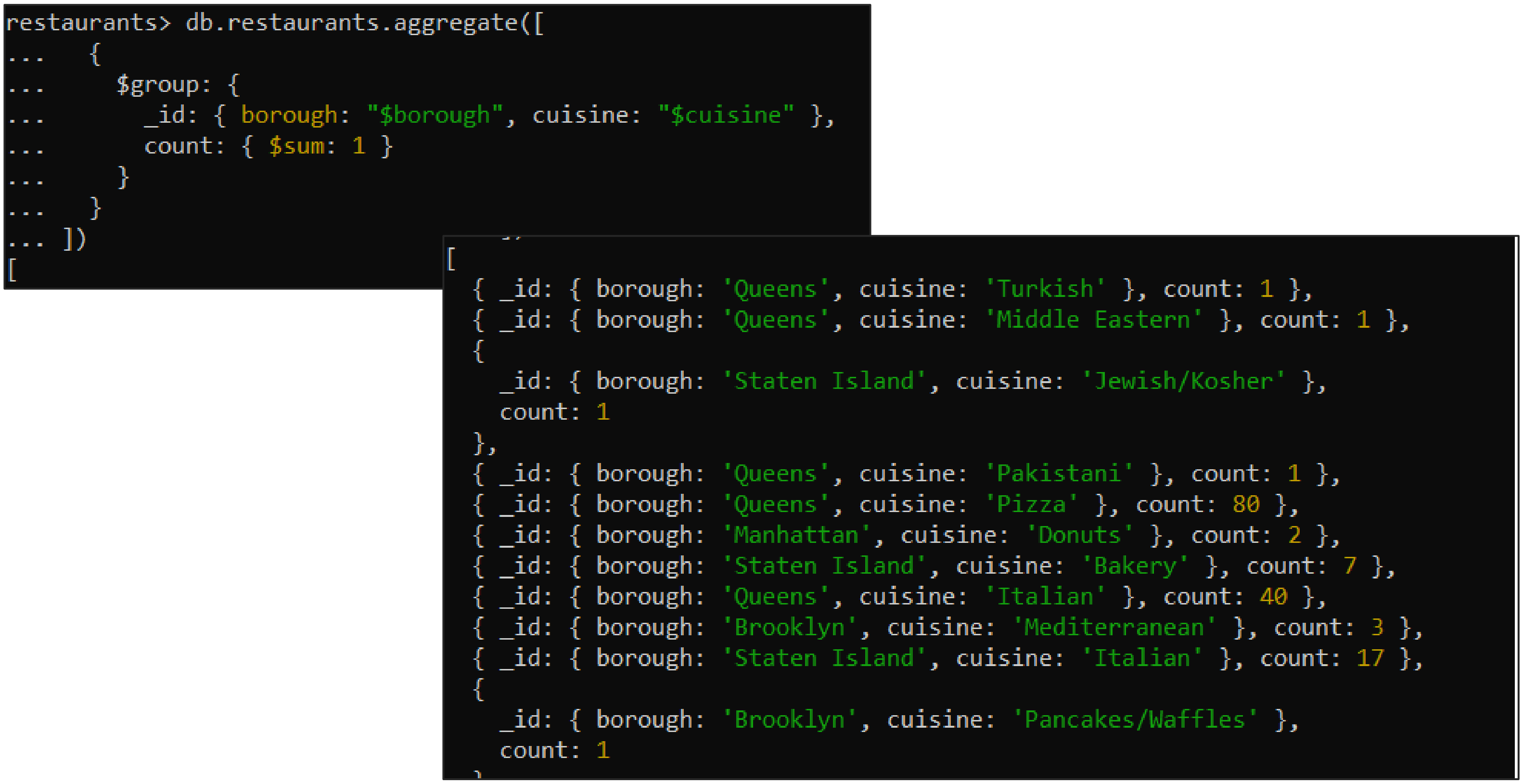
7. Найдите идентификатор ресторана, название и оценки для тех ресторанов, которые «***2014-08-11T00: 00: 00Z***» набрали ***9*** баллов за оценку ***A***



```
restaurants> db.restaurants.find(
...     {
...         "grades": {
...             $elemMatch: {
...                 "date.$date": new Date("2014-08-11").getTime(),
...                 "grade": "A",
...                 "score": 9
...             }
...         }
...     },
...     {
...         _id: 0,
...         restaurant_id: 1,
...         name: 1,
...         grades: 1
...     }
... )
```

```
[
  {
    grades: [
      { date: { '$date': 1407715200000 }, grade: 'A', score: 9 },
      { date: { '$date': 1389225600000 }, grade: 'A', score: 8 },
      { date: { '$date': 1368403200000 }, grade: 'B', score: 26 },
      { date: { '$date': 1346716800000 }, grade: 'B', score: 27 },
      { date: { '$date': 1328486400000 }, grade: 'B', score: 27 }
    ],
    name: 'Naim Kosher Pizza',
    restaurant_id: '40401060'
  },
  {
    grades: [
      { date: { '$date': 1421020800000 }, grade: 'A', score: 10 },
      { date: { '$date': 1407715200000 }, grade: 'A', score: 9 },
      { date: { '$date': 1389657600000 }, grade: 'A', score: 13 },
      { date: { '$date': 1360195200000 }, grade: 'A', score: 10 },
      { date: { '$date': 1335744000000 }, grade: 'A', score: 11 }
    ],
    name: 'Club Macanudo (Cigar Bar)',
    restaurant_id: '40526406'
  }
]
```

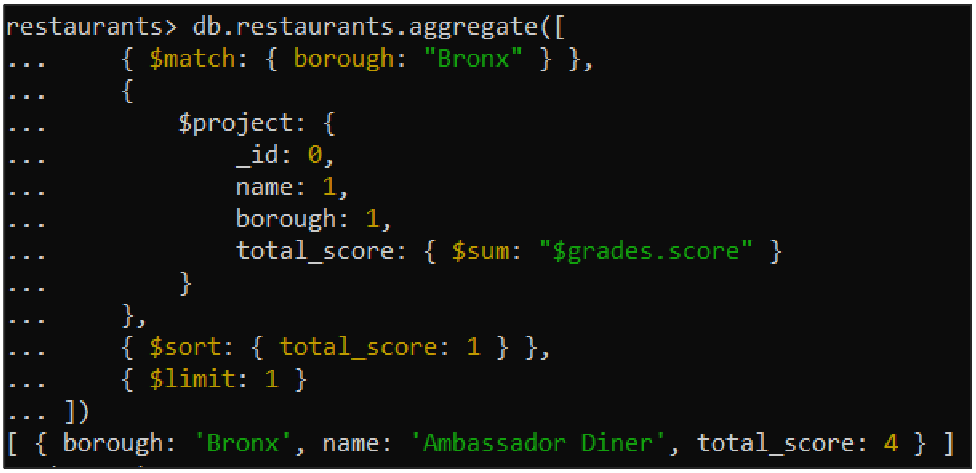
8. В каждом районе посчитайте количество ресторанов по каждому виду кухни. Документ должен иметь формат ***borough, cuisine, count***



```
restaurants> db.restaurants.aggregate([
...   {
...     $group: {
...       _id: { borough: "$borough", cuisine: "$cuisine" },
...       count: { $sum: 1 }
...     }
...   }
... ])
[
```

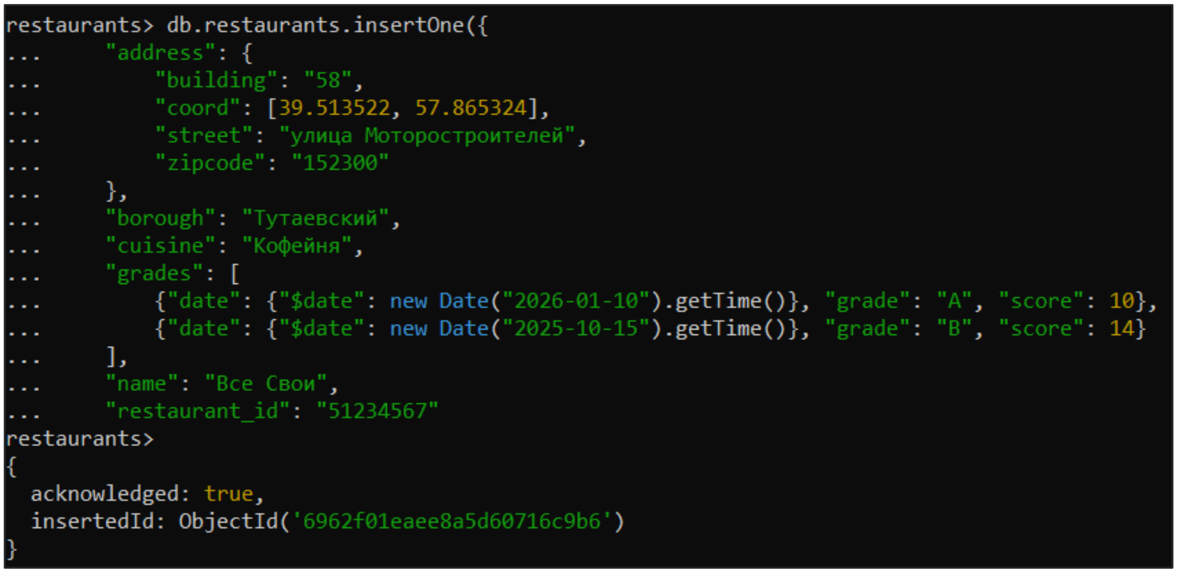
```
[
  { _id: { borough: 'Queens', cuisine: 'Turkish' }, count: 1 },
  { _id: { borough: 'Queens', cuisine: 'Middle Eastern' }, count: 1 },
  {
    _id: { borough: 'Staten Island', cuisine: 'Jewish/Kosher' },
    count: 1
  },
  { _id: { borough: 'Queens', cuisine: 'Pakistani' }, count: 1 },
  { _id: { borough: 'Queens', cuisine: 'Pizza' }, count: 80 },
  { _id: { borough: 'Manhattan', cuisine: 'Donuts' }, count: 2 },
  { _id: { borough: 'Staten Island', cuisine: 'Bakery' }, count: 7 },
  { _id: { borough: 'Queens', cuisine: 'Italian' }, count: 40 },
  { _id: { borough: 'Brooklyn', cuisine: 'Mediterranean' }, count: 3 },
  { _id: { borough: 'Staten Island', cuisine: 'Italian' }, count: 17 },
  {
    _id: { borough: 'Brooklyn', cuisine: 'Pancakes/Waffles' },
    count: 1
```

9. В районе ***Bronx*** найдите ресторан с минимальной суммой набранных баллов.



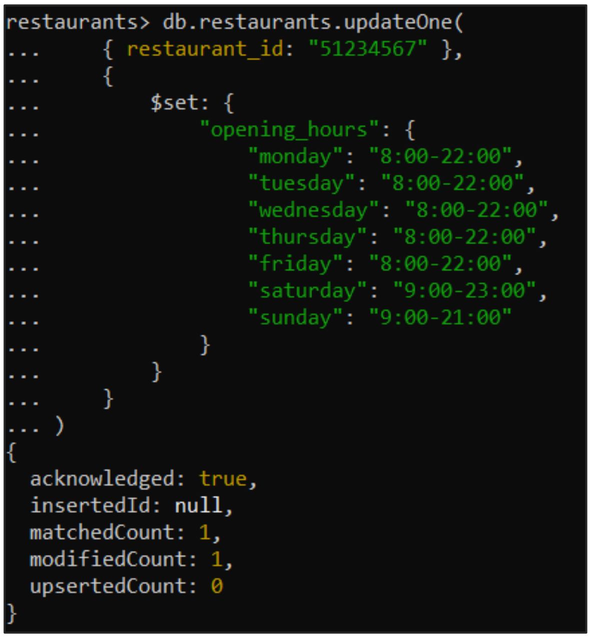
```
restaurants> db.restaurants.aggregate([
...     { $match: { borough: "Bronx" } },
...     {
...         $project: {
...             _id: 0,
...             name: 1,
...             borough: 1,
...             total_score: { $sum: "$grades.score" }
...         }
...     },
...     { $sort: { total_score: 1 } },
...     { $limit: 1 }
... ])
[ { borough: 'Bronx', name: 'Ambassador Diner', total_score: 4 } ]
```

10. Добавьте в коллекцию свой любимый ресторан.

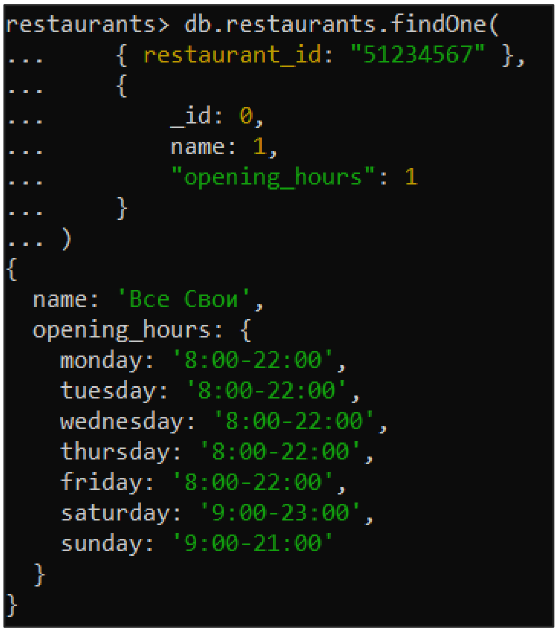


```
restaurants> db.restaurants.insertOne({
...     "address": {
...         "building": "58",
...         "coord": [39.513522, 57.865324],
...         "street": "улица Моторостроителей",
...         "zipcode": "152300"
...     },
...     "borough": "Тутаевский",
...     "cuisine": "Кофейня",
...     "grades": [
...         {"date": {"$date": new Date("2026-01-10").getTime()}, "grade": "A", "score": 10},
...         {"date": {"$date": new Date("2025-10-15").getTime()}, "grade": "B", "score": 14}
...     ],
...     "name": "Все Свои",
...     "restaurant_id": "51234567"
restaurants>
{
  acknowledged: true,
  insertedId: ObjectId('6962f01eaee8a5d60716c9b6')
}
```

11. В добавленном ресторане укажите информацию о времени его работы.

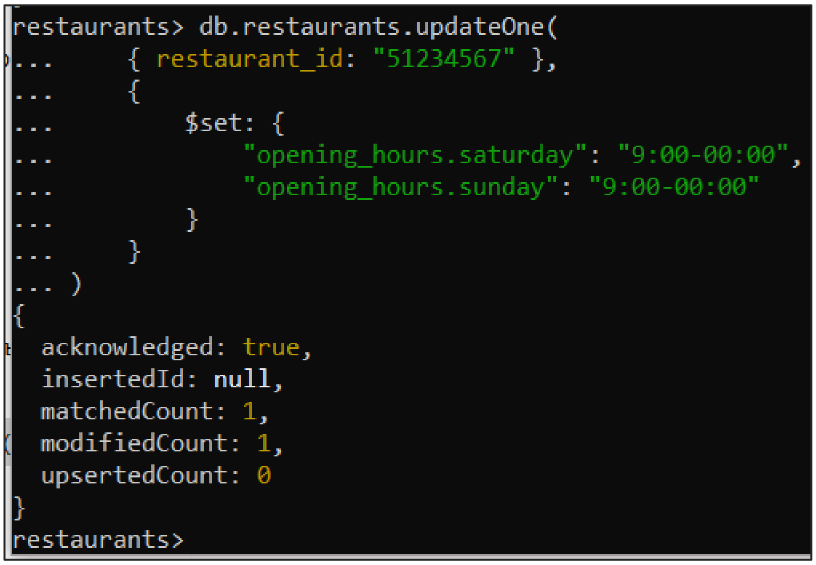


```
restaurants> db.restaurants.updateOne(
...     { restaurant_id: "51234567" },
...     {
...         $set: {
...             "opening_hours": {
...                 "monday": "8:00-22:00",
...                 "tuesday": "8:00-22:00",
...                 "wednesday": "8:00-22:00",
...                 "thursday": "8:00-22:00",
...                 "friday": "8:00-22:00",
...                 "saturday": "9:00-23:00",
...                 "sunday": "9:00-21:00"
...             }
...         }
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

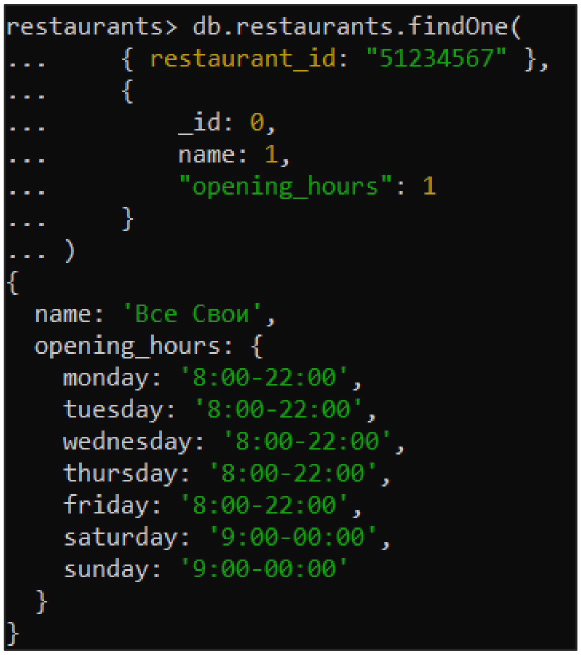


```
restaurants> db.restaurants.findOne(
...     { restaurant_id: "51234567" },
...     {
...         _id: 0,
...         name: 1,
...         "opening_hours": 1
...     }
... )
{
  name: 'Все Свои',
  opening_hours: {
    monday: '8:00-22:00',
    tuesday: '8:00-22:00',
    wednesday: '8:00-22:00',
    thursday: '8:00-22:00',
    friday: '8:00-22:00',
    saturday: '9:00-23:00',
    sunday: '9:00-21:00'
  }
}
```

12. Измените время работы вашего любимого ресторана.



```
restaurants> db.restaurants.updateOne(
...     { restaurant_id: "51234567" },
...     {
...         $set: {
...             "opening_hours.saturday": "9:00-00:00",
...             "opening_hours.sunday": "9:00-00:00"
...         }
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
restaurants>
```



```
restaurants> db.restaurants.findOne(
...     { restaurant_id: "51234567" },
...     {
...         _id: 0,
...         name: 1,
...         "opening_hours": 1
...     }
... )
{
  name: 'Все Свои',
  opening_hours: {
    monday: '8:00-22:00',
    tuesday: '8:00-22:00',
    wednesday: '8:00-22:00',
    thursday: '8:00-22:00',
    friday: '8:00-22:00',
    saturday: '9:00-00:00',
    sunday: '9:00-00:00'
  }
}
```

Структура коллекции ресторанов

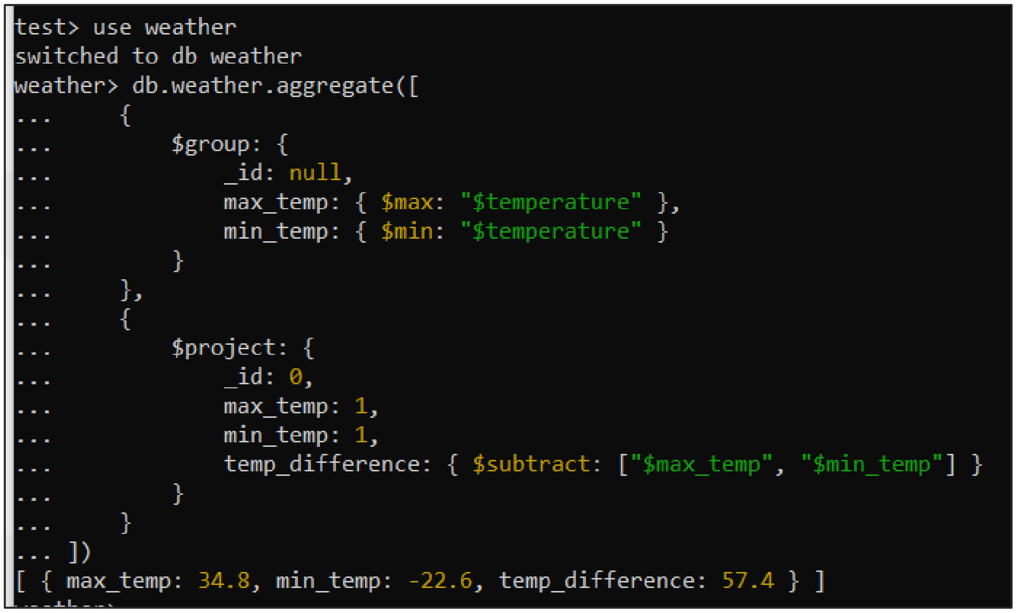
```
{
  "address": {
      «building»: «1007»,
      «coord»: [-73,856077, 40,848447],
      "street": "Моррис Парк Авеню",
      "zipcode": 10462
  },
  "borough": "Бронкс",
  "cuisine": "пекарня",
  "grades": [
      {"date": {"$ date": 1393804800000}, "grade": "A", "score": 2},
      {"date": {"$ date": 1378857600000}, "grade": "A", "score": 6},
      {"date": {"$ date": 1358985600000}, "grade": "A", "score": 10},
      {"date": {"$ date": 1322006400000}, "grade": "A", "score": 9},
      {"date": {"$ date": 1299715200000}, "grade": "B", "score": 14}
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

Задание 2

Выполните импорт коллекции из файла weather.json

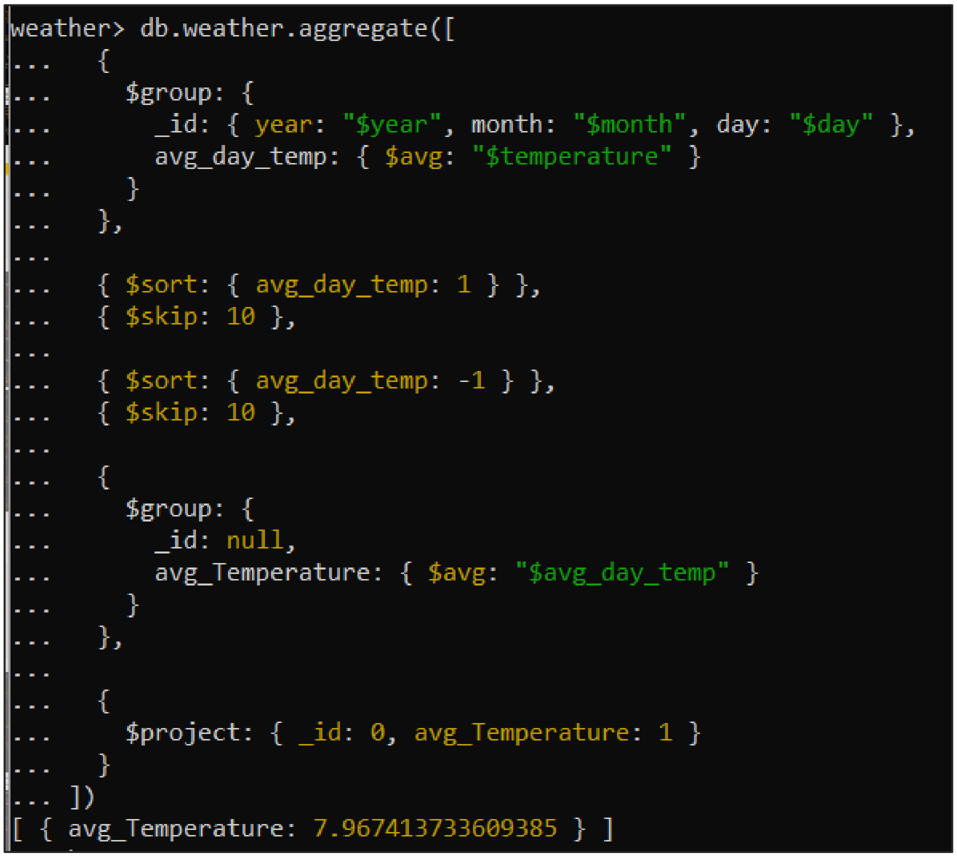
Выполните запросы с использованием Aggregate:

1. Какова разница между максимальной и минимальной температурой в течение года?



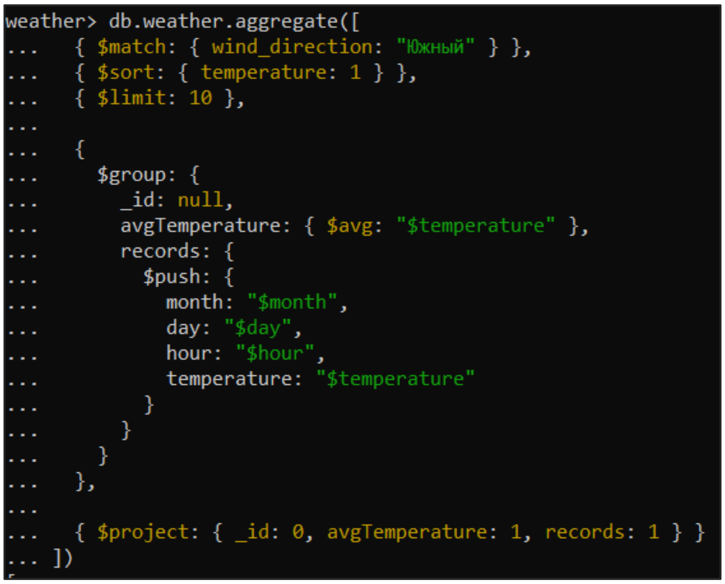
```
test> use weather
switched to db weather
weather> db.weather.aggregate([
...     {
...         $group: {
...             _id: null,
...             max_temp: { $max: "$temperature" },
...             min_temp: { $min: "$temperature" }
...         }
...     },
...     {
...         $project: {
...             _id: 0,
...             max_temp: 1,
...             min_temp: 1,
...             temp_difference: { $subtract: ["$max_temp", "$min_temp"] }
...         }
...     }
... ])
[ { max_temp: 34.8, min_temp: -22.6, temp_difference: 57.4 } ]
```

2. Какова средняя температура в году, если исключить 10 дней с самой низкой температурой и 10 дней с самой высокой?

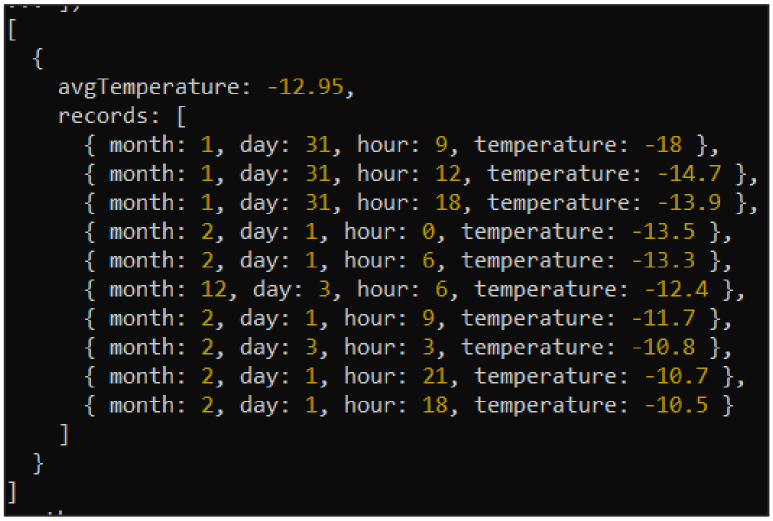


```
weather> db.weather.aggregate([
...   {
...     $group: {
...       _id: { year: "$year", month: "$month", day: "$day" },
...       avg_day_temp: { $avg: "$temperature" }
...     }
...   },
...
...   { $sort: { avg_day_temp: 1 } },
...   { $skip: 10 },
...
...   { $sort: { avg_day_temp: -1 } },
...   { $skip: 10 },
...
...   {
...     $group: {
...       _id: null,
...       avg_Temperature: { $avg: "$avg_day_temp" }
...     }
...   },
...
...   {
...     $project: { _id: 0, avg_Temperature: 1 }
...   }
... ])
[ { avg_Temperature: 7.967413733609385 } ]
```

3. Найти первые 10 записей с самой низкой погодой, когда дул ветер с юга и посчитайте среднюю температуры для этих записей

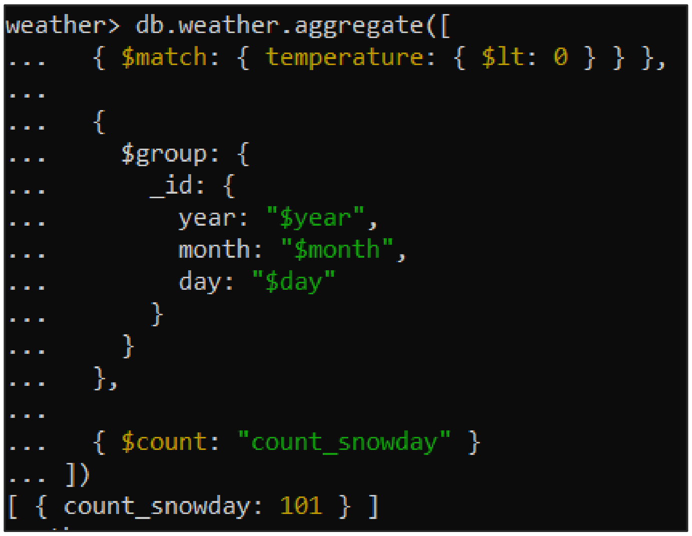


```
weather> db.weather.aggregate([
...   { $match: { wind_direction: "Южный" } },
...   { $sort: { temperature: 1 } },
...   { $limit: 10 },
...
...   {
...     $group: {
...       _id: null,
...       avgTemperature: { $avg: "$temperature" },
...       records: {
...         $push: {
...           month: "$month",
...           day: "$day",
...           hour: "$hour",
...           temperature: "$temperature"
...         }
...       }
...     }
...   },
...
...   { $project: { _id: 0, avgTemperature: 1, records: 1 } }
... ])
```



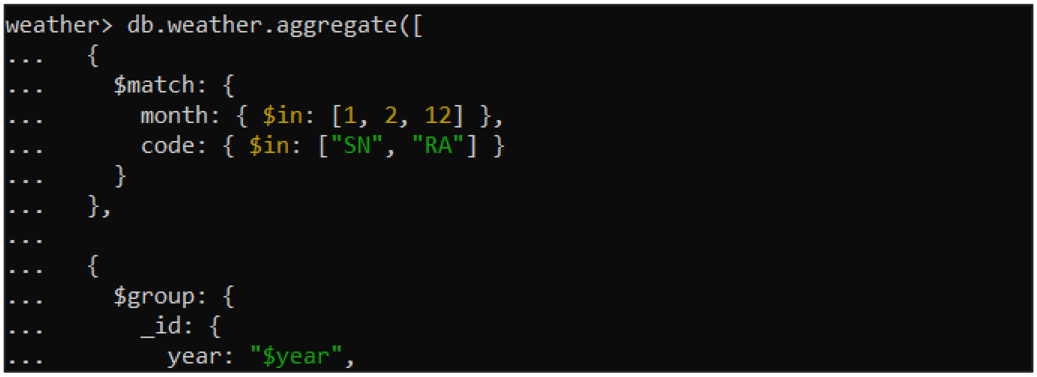
```
[
  {
    avgTemperature: -12.95,
    records: [
      { month: 1, day: 31, hour: 9, temperature: -18 },
      { month: 1, day: 31, hour: 12, temperature: -14.7 },
      { month: 1, day: 31, hour: 18, temperature: -13.9 },
      { month: 2, day: 1, hour: 0, temperature: -13.5 },
      { month: 2, day: 1, hour: 6, temperature: -13.3 },
      { month: 12, day: 3, hour: 6, temperature: -12.4 },
      { month: 2, day: 1, hour: 9, temperature: -11.7 },
      { month: 2, day: 3, hour: 3, temperature: -10.8 },
      { month: 2, day: 1, hour: 21, temperature: -10.7 },
      { month: 2, day: 1, hour: 18, temperature: -10.5 }
    ]
  }
]
```

4. Подсчитайте количество дней, когда шел снег. (Будем считать снегом осадки, которые выпали, когда температура была ниже нуля)

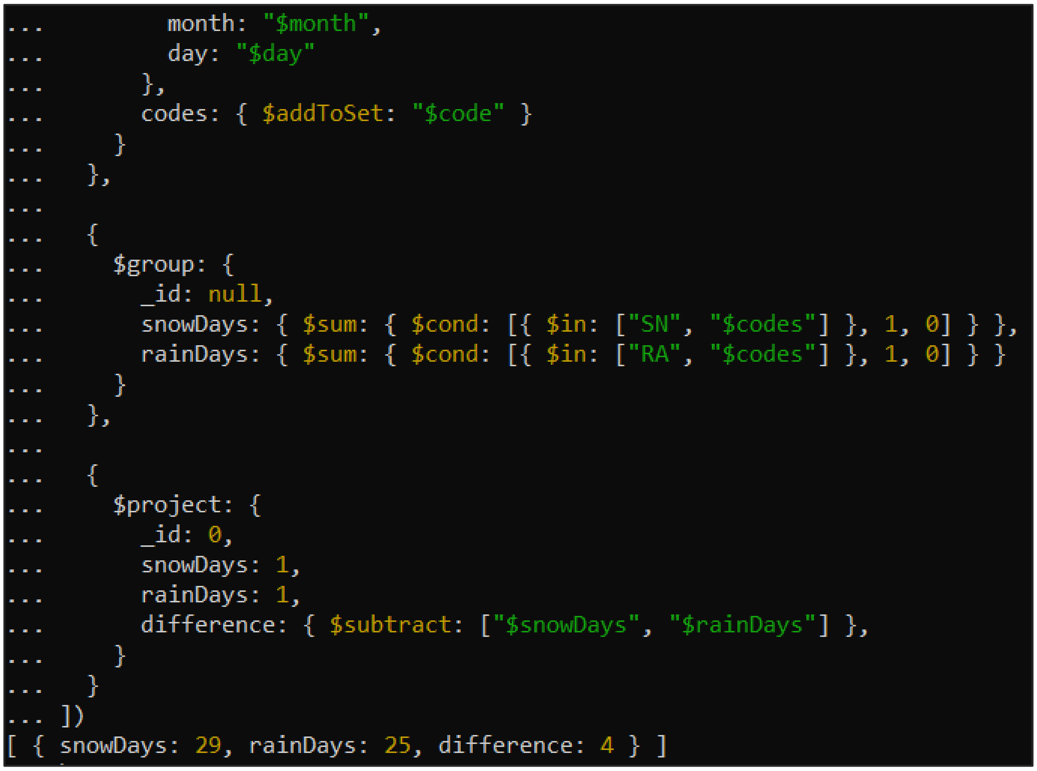


```
weather> db.weather.aggregate([
...   { $match: { temperature: { $lt: 0 } } },
...
...   {
...     $group: {
...       _id: {
...         year: "$year",
...         month: "$month",
...         day: "$day"
...       }
...     }
...   },
...
...   { $count: "count_snowday" }
... ])
[ { count_snowday: 101 } ]
```

5. В течение зимы иногда шел снег, а иногда дождь. Насколько больше (или меньше) выпало осадков в виде снега.

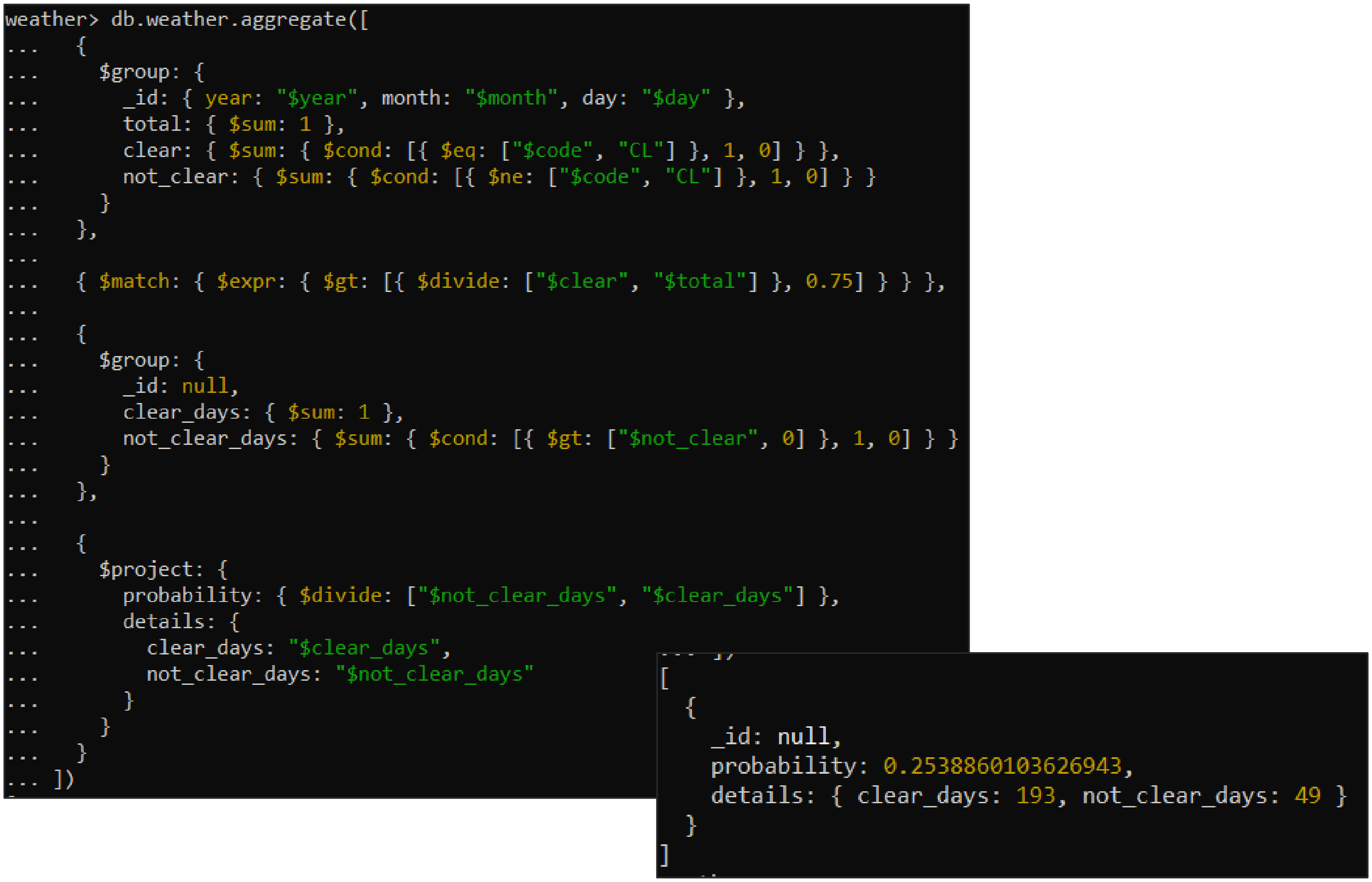


```
weather> db.weather.aggregate([
...   {
...     $match: {
...       month: { $in: [1, 2, 12] },
...       code: { $in: ["SN", "RA"] }
...     }
...   },
...
...   {
...     $group: {
...       _id: {
...         year: "$year",
```



```
...          month: "$month",
...          day: "$day"
...        },
...        codes: { $addToSet: "$code" }
...      }
...    },
...
...    {
...      $group: {
...        _id: null,
...        snowDays: { $sum: { $cond: [{ $in: ["SN", "$codes"] }, 1, 0] } },
...        rainDays: { $sum: { $cond: [{ $in: ["RA", "$codes"] }, 1, 0] } }
...      }
...    },
...
...    {
...      $project: {
...        _id: 0,
...        snowDays: 1,
...        rainDays: 1,
...        difference: { $subtract: ["$snowDays", "$rainDays"] },
...      }
...    }
... ])
[ { snowDays: 29, rainDays: 25, difference: 4 } ]
```

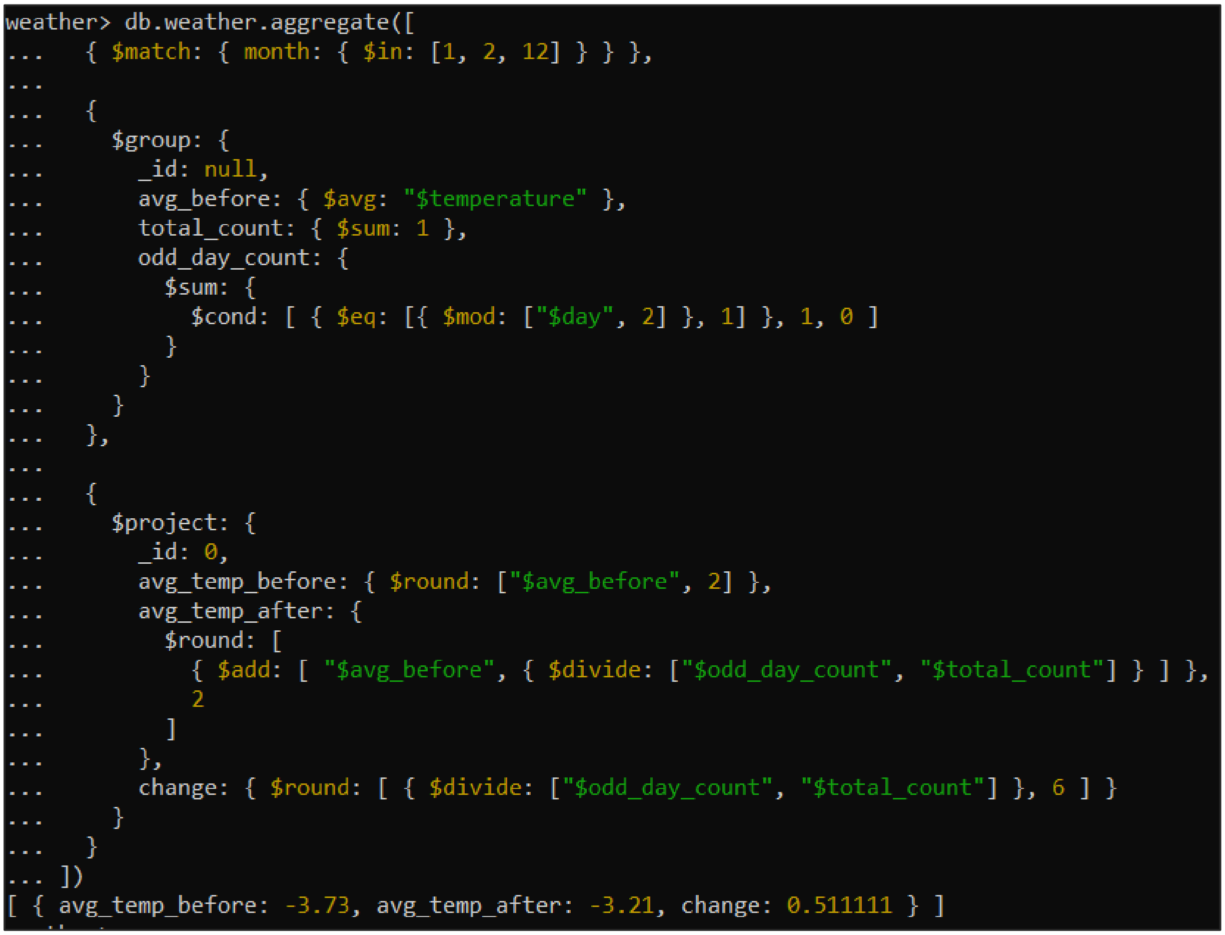
6. Какова вероятность того что в ясный день выпадут осадки? (Предположим, что день считается ясным, если ясная погода фиксируется более чем в 75% случаев)



```
weather> db.weather.aggregate([
...   {
...     $group: {
...       _id: { year: "$year", month: "$month", day: "$day" },
...       total: { $sum: 1 },
...       clear: { $sum: { $cond: [{ $eq: ["$code", "CL"] }, 1, 0] } },
...       not_clear: { $sum: { $cond: [{ $ne: ["$code", "CL"] }, 1, 0] } }
...     }
...   },
...
...   { $match: { $expr: { $gt: [{ $divide: ["$clear", "$total"] }, 0.75] } } },
...
...   {
...     $group: {
...       _id: null,
...       clear_days: { $sum: 1 },
...       not_clear_days: { $sum: { $cond: [{ $gt: ["$not_clear", 0] }, 1, 0] } }
...     }
...   },
...
...   {
...     $project: {
...       probability: { $divide: ["$not_clear_days", "$clear_days"] },
...       details: {
...         clear_days: "$clear_days",
...         not_clear_days: "$not_clear_days"
...       }
...     }
...   }
... ])
```

```
[
  {
    _id: null,
    probability: 0.2538860103626943,
    details: { clear_days: 193, not_clear_days: 49 }
  }
]
```

7. Увеличьте температуру на один градус при каждом измерении в нечетный день во время зимы.   На сколько градусов изменилась средняя температура?



```
weather> db.weather.aggregate([
...   { $match: { month: { $in: [1, 2, 12] } } },
...
...   {
...     $group: {
...       _id: null,
...       avg_before: { $avg: "$temperature" },
...       total_count: { $sum: 1 },
...       odd_day_count: {
...         $sum: {
...           $cond: [ { $eq: [{ $mod: ["$day", 2] }, 1] }, 1, 0 ]
...         }
...       }
...     }
...   },
...
...   {
...     $project: {
...       _id: 0,
...       avg_temp_before: { $round: ["$avg_before", 2] },
...       avg_temp_after: {
...         $round: [
...           { $add: [ "$avg_before", { $divide: ["$odd_day_count", "$total_count"] } ] },
...           2
...         ]
...       },
...       change: { $round: [ { $divide: ["$odd_day_count", "$total_count"] }, 6 ] }
...     }
...   }
... ])
[ { avg_temp_before: -3.73, avg_temp_after: -3.21, change: 0.511111 } ]
```

Структура коллекции Погода

```
{
      "year": 2014,
      "month": 1,
      "day": 1,
      "hour": 0,
      "temperature": -1,
      "wind_direction": "Южный",
      "wind": "3",
      "code": "CL",
      "clouds": 9,
      "visibility": 10,
      "humidity": 94,
      "pressure": 1030
  }
```