

## lab 10 – Week 14

### (MongoDB – Aggregation)

PARAS SINGH 165-114-232

#### Tasks

1. Write an aggregate statement to sort the documents in the *grades* collection based on students ID and class ID. Display only the student ID and the class ID for each document. Sort the result from high to low values for student ID and from low to high for class ID.

```
college> db.grades.aggregate([
...   {
...     $project: {
...       _id: 0,
...       student_id: 1,
...       class_id: 1
...     }
...   },
...   { $sort: { student_id: -1, class_id: 1 } }
... ])
[
  { student_id: 49, class_id: 29 },
  { student_id: 48, class_id: 2 },
  { student_id: 48, class_id: 8 },
  { student_id: 48, class_id: 10 },
  { student_id: 48, class_id: 14 },
  { student_id: 48, class_id: 15 },
  { student_id: 47, class_id: 22 },
  { student_id: 46, class_id: 3 },
  { student_id: 46, class_id: 20 },
  { student_id: 45, class_id: 5 },
  { student_id: 45, class_id: 6 },
  { student_id: 45, class_id: 13 },
  { student_id: 45, class_id: 25 },
  { student_id: 45, class_id: 29 },
  { student_id: 44, class_id: 1 },
  { student_id: 44, class_id: 7 },
  { student_id: 44, class_id: 11 },
  { student_id: 44, class_id: 21 },
  college>
```

2. Revise the previous query to show the result for students with IDs between 10 and 12.

```
college> db.grades.aggregate([
...   {
...     $match: {
...       student_id: { $gte: 10, $lte: 12 }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       student_id: 1,
...       class_id: 1
...     }
...   },
...   { $sort: { student_id: -1, class_id: 1 } }
... ])
[
  { student_id: 12, class_id: 3 },
  { student_id: 12, class_id: 4 },
  { student_id: 12, class_id: 10 },
  { student_id: 12, class_id: 11 },
  { student_id: 12, class_id: 13 },
  { student_id: 12, class_id: 21 },
  { student_id: 12, class_id: 23 },
  { student_id: 12, class_id: 27 },
  { student_id: 11, class_id: 24 },
  { student_id: 11, class_id: 25 },
  { student_id: 10, class_id: 5 }
]
```

3. Show only existing class IDs in the grades collection. (Do not show duplicates.)

```
college> db.grades.aggregate([
...   {
...     $group: {
...       _id: "$class_id"
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       class_id: "$_id"
...     }
...   }
... ])
[
  { class_id: 8 }, { class_id: 3 },
  { class_id: 18 }, { class_id: 13 },
  { class_id: 10 }, { class_id: 11 },
  { class_id: 29 }, { class_id: 16 },
  { class_id: 0 }, { class_id: 20 },
  { class_id: 19 }, { class_id: 14 },
  { class_id: 17 }, { class_id: 4 },
  { class_id: 30 }, { class_id: 27 },
  { class_id: 12 }, { class_id: 23 },
  { class_id: 22 }, { class_id: 24 }
]
```

4. Write a query to display student ID and class ID for students whose score are greater than 99.00. Sort the result based on student ID from high to low and class ID from low to high.

```

college> db.grades.aggregate([
...   {
...     $match: {
...       "scores.type": "exam",
...       "scores.score": { $gt: 99.00 }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       student_id: 1,
...       class_id: 1
...     }
...   },
...   { $sort: { student_id: -1, class_id: 1 } }
... ])
[
  { student_id: 44, class_id: 7 },
  { student_id: 44, class_id: 11 },
  { student_id: 43, class_id: 27 },
  { student_id: 42, class_id: 14 },
  { student_id: 41, class_id: 19 },
  { student_id: 21, class_id: 25 },
  { student_id: 21, class_id: 30 },
  { student_id: 14, class_id: 6 },
  { student_id: 14, class_id: 20 },
  { student_id: 12, class_id: 27 },
  { student_id: 3, class_id: 11 }
]
college>

```

5. Write a query to show the maximum and the minimum class ID for each student. *Sort the result based on student ID from low to high.* Show only the first **10** students. See the following sample output:

```

{ "_id" : 0, "max_class_id" : 30, "min_class_id" : 2 }
{ "_id" : 1, "max_class_id" : 28, "min_class_id" : 13 }
{ "_id" : 2, "max_class_id" : 27, "min_class_id" : 24 }
{ "_id" : 3, "max_class_id" : 25, "min_class_id" : 3 }
{ "_id" : 4, "max_class_id" : 26, "min_class_id" : 0 }
{ "_id" : 5, "max_class_id" : 30, "min_class_id" : 0 }
{ "_id" : 6, "max_class_id" : 29, "min_class_id" : 8 }
{ "_id" : 7, "max_class_id" : 17, "min_class_id" : 17 }
{ "_id" : 8, "max_class_id" : 29, "min_class_id" : 0 }
{ "_id" : 9, "max_class_id" : 30, "min_class_id" : 0 }

```

```
college> db.grades.aggregate([
...   {
...     $group: {
...       _id: "$student_id",
...       max_class_id: { $max: "$class_id" },
...       min_class_id: { $min: "$class_id" }
...     },
...     { $sort: { _id: 1 } }, // Sort by student ID from low to high
...     { $limit: 10 } // Show only first 10 students
...   })
... ])
[
  { _id: 0, max_class_id: 30, min_class_id: 2 },
  { _id: 1, max_class_id: 28, min_class_id: 13 },
  { _id: 2, max_class_id: 27, min_class_id: 24 },
  { _id: 3, max_class_id: 25, min_class_id: 3 },
  { _id: 4, max_class_id: 26, min_class_id: 0 },
  { _id: 5, max_class_id: 30, min_class_id: 0 },
  { _id: 6, max_class_id: 29, min_class_id: 8 },
  { _id: 7, max_class_id: 17, min_class_id: 17 },
  { _id: 8, max_class_id: 29, min_class_id: 0 },
  { _id: 9, max_class_id: 30, min_class_id: 0 }
]
college>
```

6. Write a query to find the number of failed exams for student with ID 48.

```
college> db.grades.aggregate([
...   {
...     $match: { student_id: 48 }
...   },
...   {
...     $unwind: "$scores"
...   },
...   {
...     $match: {
...       "scores.type": "exam",
...       "scores.score": { $lt: 50 }
...     }
...   },
...   { $count: "failed_exams" }
... ])
[ { failed_exams: 3 } ]
```

--

Good luck.