DBS311NEE L10 Answers and outputs

**Name: Wei-Chih Kao  
ID: 158000190  
Date: Dec. 06. 2021**

Task1  
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.

> db.grades.aggregate({$project : {\_id:0, student\_id:1, class\_id:1}}, {$sort : {"student\_id" : -1, "class\_id" : 1}})

Output:

{ "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 }  
{ "student\_id" : 43, "class\_id" : 2 }  
{ "student\_id" : 43, "class\_id" : 10 }  
Type "it" for more

Task2  
Revise the previous query to show the result for students with IDs between 10 and 12.

> db.grades.aggregate({$project : {\_id: 0, student\_id: 1, class\_id: 1}}, {$match: {student\_id: {$gte: 10, $lte: 12}}}, {$sort : {"student\_id" : 1, "class\_id" : -1}})

Output:

{ "student\_id" : 10, "class\_id" : 5 }  
{ "student\_id" : 11, "class\_id" : 25 }  
{ "student\_id" : 11, "class\_id" : 24 }  
{ "student\_id" : 12, "class\_id" : 27 }  
{ "student\_id" : 12, "class\_id" : 23 }  
{ "student\_id" : 12, "class\_id" : 21 }  
{ "student\_id" : 12, "class\_id" : 13 }  
{ "student\_id" : 12, "class\_id" : 11 }  
{ "student\_id" : 12, "class\_id" : 10 }  
{ "student\_id" : 12, "class\_id" : 4 }  
{ "student\_id" : 12, "class\_id" : 3 }

Task3  
Show only existing class IDs in the grades collection. (Do not show duplicates.)  
  
> db.grades.aggregate({$group:{\_id:0,class\_id:{$addToSet: "$class\_id"}}})  
  
Output:

{ "\_id" : 0, "class\_id" : [ 27, 16, 26, 0, 6, 25, 15, 9, 17, 12, 23, 22, 8, 20, 4, 1, 3, 14, 5, 18, 2, 10, 11, 29, 13, 7, 21, 30, 19, 28, 24 ] }

Task4  
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.  
  
> db.grades.aggregate({$unwind: "$scores"}, {$project : {\_id: 0, student\_id: 1, class\_id: 1, scores: 1}}, {$match: {"scores.score": {$gt: 99}}}, {$sort: {"student\_id": -1, "class\_id": 1}})  
Output:

{ "student\_id" : 44, "class\_id" : 7, "scores" : { "type" : "exam", "score" : 99.36623841444576 } }  
{ "student\_id" : 44, "class\_id" : 11, "scores" : { "type" : "exam", "score" : 99.40117530792308 } }  
{ "student\_id" : 43, "class\_id" : 27, "scores" : { "type" : "exam", "score" : 99.4040826043679 } }  
{ "student\_id" : 42, "class\_id" : 14, "scores" : { "type" : "homework", "score" : 99.6826006851697 } }  
{ "student\_id" : 41, "class\_id" : 19, "scores" : { "type" : "exam", "score" : 99.9822072102702 } }  
{ "student\_id" : 21, "class\_id" : 25, "scores" : { "type" : "quiz", "score" : 99.9092162705742 } }  
{ "student\_id" : 21, "class\_id" : 30, "scores" : { "type" : "homework", "score" : 99.62785870384556 } }  
{ "student\_id" : 14, "class\_id" : 6, "scores" : { "type" : "exam", "score" : 99.49380951735357 } }  
{ "student\_id" : 14, "class\_id" : 20, "scores" : { "type" : "exam", "score" : 99.70054435154259 } }  
{ "student\_id" : 12, "class\_id" : 27, "scores" : { "type" : "homework", "score" : 99.45407211904612 } }  
{ "student\_id" : 3, "class\_id" : 11, "scores" : { "type" : "quiz", "score" : 99.70947557740388 } }

Task5  
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 }

> db.grades.aggregate({$project : {\_id: 0, student\_id: 1, class\_id: 1}}, {$group: {"\_id": "$student\_id", "max\_class\_id": {$max: "$class\_id"}, "min\_class\_id":{$min: "$class\_id"}}}, {$sort: {"\_id": 1}}, {$limit : 10})

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 }

Task6  
Write a query to find the number of failed exams for student with ID 48.

> db.grades.aggregate({$unwind: "$scores"}, {$project : {student\_id: 1, class\_id: 1, scores: 1, \_id: 0}}, {$match: {student\_id: 48, "scores.type": "exam", "scores.score": {$lt: 60}}}, {$group: {"\_id": "$student\_id", "NumOfFailed": {$sum: 1}}})

Output:

{ "\_id" : 48, "NumOfFailed" : 3 }