Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра АСОІУ

**ЗВІТ**

про виконання комп’ютерного практикуму № 5

з дисципліни

“ Програмні додатки з використанням баз даних”

по темі

«MongoDB»

Варіант 6

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Cт.вик. Клименко О. М. |  | студент 3-го курсу  гр. ІП-51 ФІОТ  Зарічковий Олександр Анатолійович |

Київ – 2017

**ЗМІСТ:**

[1 Завдання 3](#_Toc495230724)

[2 Виконання Завдання 4](#_Toc495230725)

# Завдання

Завдання 1:

1. Встановити MongoDB
2. Створити базу MongoDB на основі реляційної бази предметної області

Завдання 2:

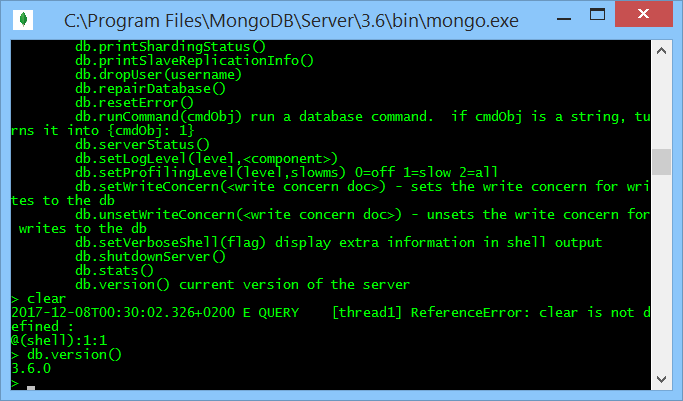
1. Перепроектувати схеми БД та доповнити даними
2. Написати до своєї предметної області 2 запити з групуванням та 2 запити з використанням агрегатних функцій.
3. Виконати запит в MongoDB, який за кодом, що зберігається в одній колекції, виводить назву, що зберігається в іншій колекції, тобто аналог JOIN в SQL.

**Преметна область**: Навчання з охорони праці

# Виконання Завдання

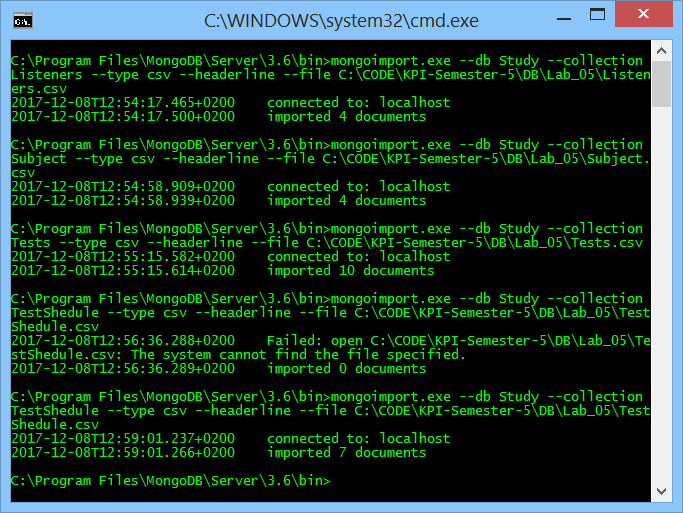
1. Завдання по базовим операціям з MongoDB:
   1. Встановлення MongoDB

*Результат виконання*:



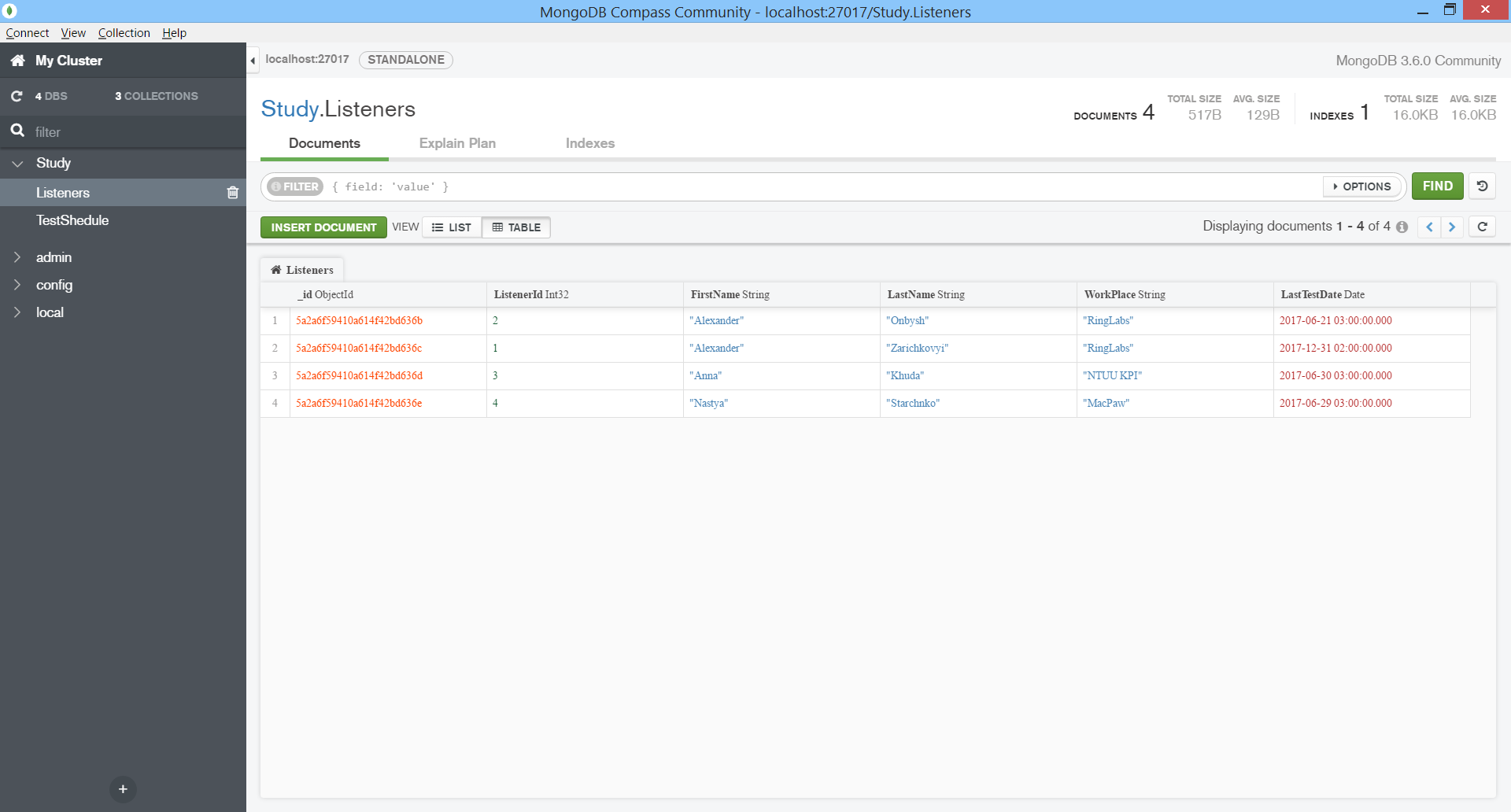
* 1. Створити базу MongoDB на основі реляційної бази предметної області:

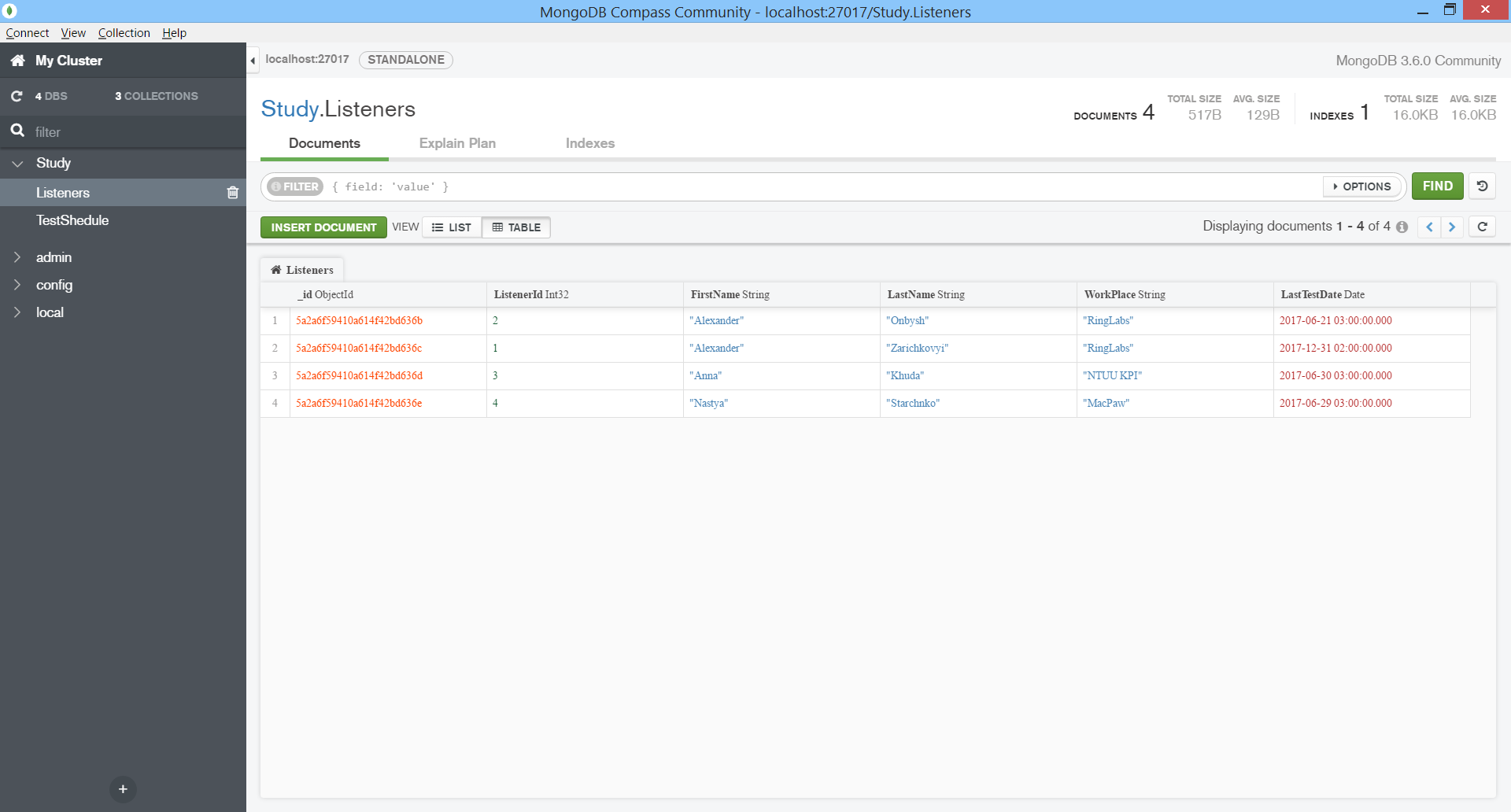
*Результат виконання*:



1. Завдання по написанню запитів в MongoDB:
   1. Перепроектувати схеми БД та доповнити даними

Опис дій: В колекції TestShedule поле SubjectId було замінено на Subject. Колекція Subject була видалена, оскільки містила дублюючу інформацію. Поле PassDate було перенесено з колекції Tests в TestShedule. Колекція Tests була видалена, оскільки містила дублюючу інформацію. В наслідок всіх цих операцій ми отримали структуру яка достатньо зрозуміла для людини, оскільки оперує об’єктами та зручний доступ до всіх необхідних полів даних. Але тепер база даних не нормалізована (відповідає лише 1 НФ), що ускладнить її подальшу модифікацію і супроводження.





* 1. Написати до своєї предметної області 2 запити з групуванням та 2 запити з використанням агрегатних функцій.
     1. *Призначення*: Підрахувати кількість документів в яких згадано поле «Математика».

*Текст запиту*:

db.TestShedule.aggregate([

{

$match:{

Subject: "Math"

}

},

{

$group:{

\_id: null,

total: {

$sum : 1

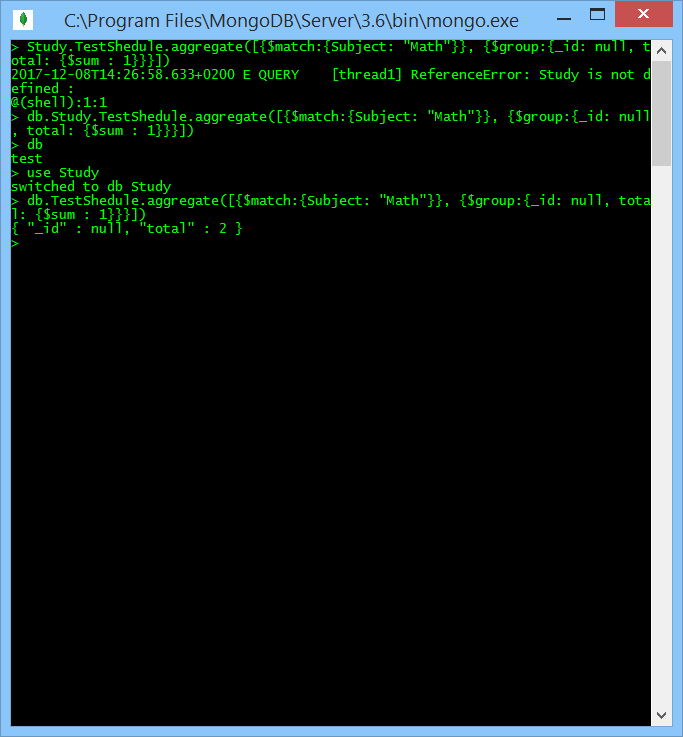
}

}

}

])

*Результат виконання*:



* + 1. *Призначення*: Для кожного слухача підрахувати кількість складених ним екзаменів.

*Текст запиту*:

db.TestShedule.aggregate([

{

$match: {

PassDate: {

$ne: null

}

}

},

{

$group:{

\_id: "$ListnerId",

passed\_exams\_count: {

$sum: 1

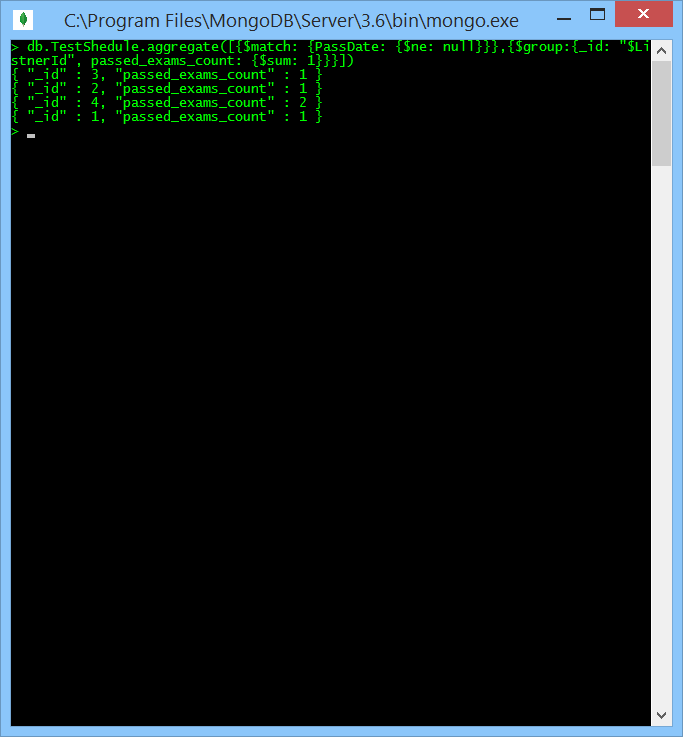
}

}

}

])

*Результат виконання*:



* + 1. *Призначення*: Вивести найновішу дату проходження екзаменів.

*Текст запиту*:

db.Listeners.aggregate(

{

$group: {

\_id: null,

newest\_pass\_date: {

$max: "$LastTestDate"

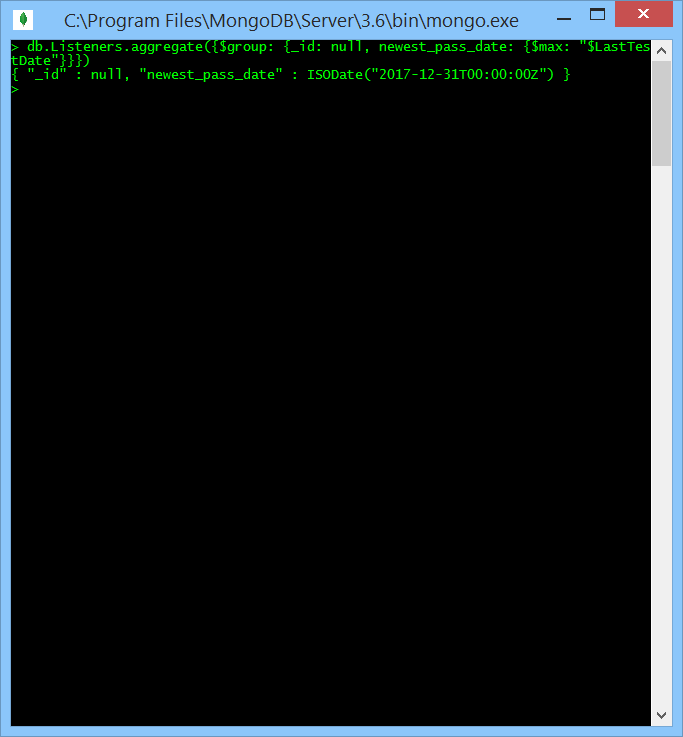
}

}

}

)

*Результат виконання*:



* + 1. *Призначення*: Вивести найпростіший предмет.

*Текст запиту*:

db.TestShedule.aggregate([

{

$match: {

PassDate: {

$ne: null

}

}

},

{

$group: {

\_id: "$Subject",

listner\_pass\_count: {

$sum : 1

}

}

},

{

$sort: {

listner\_pass\_count: -1

}

},

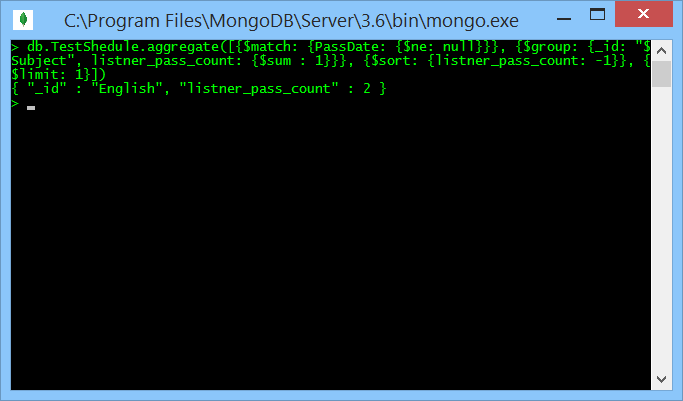
{

$limit: 1

}

])

*Результат виконання*:



* 1. Виконати запит в MongoDB, який за кодом, що зберігається в одній колекції, виводить назву, що зберігається в іншій колекції, тобто аналог JOIN в SQL.

*Призначення*: Вивести всіх слухачів та предмети, які вони складали.

*Текст запиту*:

db.TestShedule.aggregate([

{

$match: {

Subject: "Math"

}

},

{

$lookup: {

from: "Listeners",

localField: "ListnerId",

foreignField: "ListenerId",

as: "Listener"

}

},

{

$unwind: "$Listener"

}

])

// ALT JOIN

var mapData = function () {

var output= {

Id: this.ListenerId,

Subject: this.Subject,

FirstName: null,

LastName: null

};

emit(this.ListnerId, output);

};

var mapListner = function () {

var output= {

Id: this.ListenerId,

FirstName: this.FirstName,

LastName: this.LastName,

Subject: null

};

emit(this.ListenerId, output);

};

var reduceF = function(key, values) {

var result = {

Subject: [],

FirstName: null,

LastName: null

};

values.forEach(function(value) {

if(result.FirstName == null) {

result.FirstName = value.FirstName;

}

if(result.LastName == null) {

result.LastName = value.LastName;

}

if(value.Subject != null) {

result.Subject = result.Subject.concat([value.Subject]);

}

});

return result;

};

result = db.Listeners.mapReduce(

mapListner,

reduceF,

{out:

{reduce: 'joined\_collection'}

}

);

result = db.TestShedule.mapReduce(

mapData,

reduceF,

{out:

{reduce: 'joined\_collection'}

}

);

db.joined\_collection.find()

*Результат виконання*:

