LAB10: STUDENT INFORMATION SYSTEM DESIGN USING MONGODB P-||

Question1.execute the following queries in your Students collection

```
>db.students.save({_id:"arun",name:{first:"Arun",last:"kumar"},year:1992,
courses:["java","php","mongodb"]});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.students.save({_id:"sam",name:{first:"Sam",last:"Peter"},year:1995,
courses:["php","python","java"]});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.students.save({_id:"anna",name:{first:"Anna",last:"Eva"},year:1997,
courses:["java"]});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.students.save({_id:"rex",name:{first:"Rex",last:"Samuel"},year:1988,
courses:["python"]});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.students.save({_id:"olivia",name:{first:"Olivia",last:"Freda"},year:2006});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.students.save({_id:"sylvia",name:{first:"Sylvia",last:"Diana"},year:2008,
courses:["php"]});
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "sylvia" })
```

```
>db.students.save({_id:"Benita",name:{first:"Benita",last:"Sam"},year:1988,
courses:"php"});
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "Benita" })
```

Question2.Execute the following queries in your courses collection

```
> db.courses.save({_id:"java",title:{tamil:"Java Programming",en:"Java
Programming"},year:2000,rating:84,students:["arun","sam","anna"],departments:["cs","c
a"],campus:["Trichy","Dubai"]});;
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "java" })
```

```
> db.courses.save({_id:"php",title:"PHP
Programming",year:2007,professor:{first:"John",last:"Peter"},rating:53,students:["arun","
sam","sylvia","divya","hazel"],departments:["cs","ca"],campus:["Trichy","Singapore"]});
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "php" })
```

```
> db.courses.save({_id:"python",title:{tamil:"Pythonprogramming",
en:"Python Programming"},
year:2006,professor:{first:"John",last:"Santhosh"},rating:76,
students:["rex","sam","anna"],departments:"cs",campus:"Trichy"});
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "python" })
```

```
>db.courses.save({_id:"os",title:"OperatingSystems",year:2003,professor:{last:"Titus",firs
t:"Antony"},rating:81,students:[],departments:
["ds","ca"],campus:["Trichy","Sharjah","Singapore"]});
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "os" })
```

```
> db.courses.save({_id:"networks",title:"NetworkingFundamentals",
year:2005,professor:{last:"Solomon",first:"Balu"},rating:72,awards:[{type:"National
Award",year:2005}]});
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "networks" })
```

```
> db.courses.save({_id:"graphics",title:"ComputerGraphics",year:1996,
rating:86,awards:[{type:"NationalAward",year:1996},{type:"National
Award",year:1996},{type:"International Award",category:"Core CS",year:2005}]});
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "graphics" })
```

LAB 10: STUDENT INFORMATION SYSTEM DESIGN USING MONGODB

Question 3: Express the following queries, execute and find answers

1. Find students born in 1995 with first name Sam

```
> db.students.find({"name.first":"sam",year:1995})
```

```
{ "_id" : "sam", "name" : { "first" : "sam", "last" : "peter" }, "year" : 1995,
"courses" : [ "java", "php", "python" ] }
```

2. Find courses delivered by John Peter

```
> db.courses.find({"professor.first":"john","professor.last":"peter"})
```

```
{ "_id" : "PHP", "title" : "PHP programmimg", "year" : 2007, "professor" : {
"first" : "john", "last" : "peter" }, "rating" : 53, "students" : [ "arun", "sam",
"sylia", "divya", "hazel" ], "depts" : [ "cs", "ca" ], "campus" : [ "trichy",
"singapore" ] }
```

3. Find students with first name *Sam* who take the course *php.*
(<u>return name of these students only)</u>

```
> db.students.find({"name.first":"sam",courses:"php"},{name:1,_id:0})
```

```
{ "name" : { "first" : "sam", "last" : "peter" } }
```

4. Find courses delivered between years *2000* and *2005* such that they have a professor specified.
(<u>return course identifier only</u>
<u>Order of result by rating in desc and by years in aesc)</u>

```
> db.courses.find({year:{$gte:2000,$lte:2005}},{_id:1}).sort({rating:-1,year:1})
```

```
{ "_id" : "java" }
{ "_id" : "os" }

{ "_id" : "networks" }
```

5. Find students who studied courses, *java* or *php.* (return student indentifier only)

```
> db.students.find({courses:{$in:["php","java"]}},{_id:1})

{ "_id" : "arun" }

{ "_id" : "sam" }

{ "_id" : "anna" }

{ "_id" : "sylvia" }

{ "_id" : "benita" }
```

6. Find students who studied courses both *java* and *php.* (return student identifier only)

```
> db.students.find({courses:{$all:["php","java"]}},{_id:1})


{ "_id" : "arun" }

{ "_id" : "sam" }
```

7. Find courses with Tamil title equal to *Python Programming.*
(return course title only)

```
> db.courses.find({"title.tamil":"java programming"},{title:1,_id:0})

{ "title" : { "tamil" : "java programming", "en" : "java programmimg" } }
```

8. Find courses that have a *National* award from *2005* (return course identifier and all awards)

```
> db.courses.find({"awards.type":"national
awards",year:{$gte:2005}},{_id:1,"awards.type":1})
```

```
{ "_id" : "networks", "awards" : [ { "type" : "national awards" } ] }
```

9. Find courses that are offered in CS and CA departments at the same time or have a rating *80* or more. (return course identifier and at most 2 campus)

```
> db.courses.find({$or:[{depts:{$all:["cs","ca"]}} ,
{rating:{$gte:80}}]},{_id:1,campus:{$slice:2}})
```

```
{ "_id" : "java", "campus" : [ "trichy", "dubai" ] }

{ "_id" : "PHP", "campus" : [ "trichy", "singapore" ] }

{ "_id" : "graphics" }

{ "_id" : "os", "campus" : [ "trichy", "sharjah"] }
```