Lab10. Student Information System Design using MongoDB PART-II

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"]
});

db.students.save({
        _id: "sam",
        name: { first: "Sam", last:"Peter" }, year: 1995,
        courses: [ "php", "python", "java" ]
});

db.students.save({
        _id: "anna",
        name: {first: "Anna", last:"Eva" }, year: 1997,
        courses: [ "java" ]
});

db.students.save({
        _id: "rex",
        name: {first: "Rex", last:"Samuel" }, year: 1988,
        courses: [ "python" ]
});

db.students.save({
        _id: "olivia",
        name: { first: "Olivia", last:"Freda" }, year: 2006
});

db.students.save({
  _id: "sylvia",
  name: { first: "Sylvia", last: "Diana" }, year: 2008,
  courses: [ "php"]
});

db.students.save({
  _id: "Benita",
  name: { last: "Benita", first: "Sam" }, year: 1988,
  courses: "php"
```

});

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", "ca" ], campus: [ "Trichy", "Dubai" ] });

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" ] });

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

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

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

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

Question3. Express the following MongoDB 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

- Note that the order of fields for first and last names can be arbitrary

```
> db.courses.find({"professor.first": "john", "professor.last":
"peter"})
{"id": "php", title": "php programming", "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*

- Return names of these students only

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

4. Find courses delivered between years *2000* and *2005* such that they have a professor specified

- Return course identifier only
- Order the result by ratings in descending order and then by years in ascending order

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

5. Find students who studied courses, *java* or *php*

- Return student identifier only
- Propose two different approaches

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

6. Find students who studied courses both *java* and *php*

- Return student identifier only
- Propose two different approaches

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

7. Find courses with Tamil title equal to *Python Programming*

- Return course title only
- Note that there are two ways how course titles are defined

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

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})
```

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 campuses

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